

Dokumentation | DE

BC9020, BC9050 and BC9120

Busklemmen-Controller für Ethernet



Inhaltsverzeichnis

1	Vorwort	7
1.1	Hinweise zur Dokumentation	7
1.2	Sicherheitshinweise	8
1.3	Ausgabestände der Dokumentation	9
1.4	Hinweise zur Informationssicherheit	10
2	Produktübersicht	11
2.1	BC9020, BC9050 und BC9120 - Übersicht	11
2.1.1	BC9020 - Einführung	12
2.1.2	BC9050 - Einführung	13
2.1.3	BC9120 - Einführung	14
2.2	Technische Daten	15
2.2.1	Technische Daten - Ethernet	15
2.2.2	Technische Daten - Busklemmen-Controller	16
2.2.3	Technische Daten - SPS	17
2.3	Prinzip der Busklemme	18
2.4	Das Beckhoff Busklemmensystem	19
3	Montage und Verdrahtung	21
3.1	Hinweise zum ESD-Schutz	21
3.2	Montage	22
3.2.1	Abmessungen	22
3.2.2	Tragschienenmontage	24
3.2.3	Entsorgung	24
3.3	Verdrahtung	25
3.3.1	Potentialgruppen, Isolationsprüfung und PE	25
3.3.2	Spannungsversorgung	27
3.3.3	Ethernet-Topologien	28
3.3.4	Ethernet-Anschluss	30
3.3.5	Ethernet-Kabel	33
3.4	ATEX - Besondere Bedingungen (Standardtemperaturbereich)	35
3.5	ATEX - Besondere Bedingungen (erweiterter Temperaturbereich)	36
3.6	Weiterführende Dokumentation zu ATEX und IECEx	37
4	Parametrierung und Inbetriebnahme	38
4.1	Anlaufverhalten des Busklemmen-Controllers	38
4.2	DIP-Schalter	39
4.3	Einstellung der IP-Adresse	41
4.3.1	Adresseinstellung über den TwinCAT System Manager	42
4.3.2	Adresseinstellung über BootP-Server	42
4.3.3	Adresseinstellung über DHCP-Server	44
4.3.4	Auto-IP-Adresse	44
4.3.5	Subnetz-Maske	44
4.4	Konfiguration	45
4.4.1	Überblick	45
4.4.2	Busklemmen-Controller mit dem TwinCAT System Manager suchen	47

4.4.3	Anlegen einer TwinCAT-Konfiguration	49
4.4.4	Download einer TwinCAT-Konfiguration.....	50
4.4.5	Upload einer TwinCAT-Konfiguration	52
4.4.6	Ressourcen im Busklemmen-Controller	53
4.4.7	ADS-Verbindung über die serielle Schnittstelle	56
4.4.8	Ethernet	57
4.4.9	K-Bus	60
4.4.10	PLC	62
4.5	KS2000.....	66
4.5.1	Konfigurations-Software KS2000.....	66
5	Programmierung.....	67
5.1	PLC-Eigenschaften.....	67
5.2	TwinCAT PLC.....	68
5.3	TwinCAT PLC - Fehler-Codes.....	68
5.4	Remanente Daten	71
5.5	Persistente Daten	72
5.6	Lokierte Merker.....	72
5.7	Lokales Prozessabbild im Auslieferungszustand (Default Config)	74
5.8	Mapping der Busklemmen	75
5.9	Lokales Prozessabbild in der TwinCAT-Konfiguration	76
5.10	Erzeugen eines Boot-Projekts	77
5.11	Kommunikation zwischen TwinCAT und BX/BCxx50.....	77
5.12	Up- und Download von Programmen	79
5.13	Bibliotheken	82
5.13.1	Bibliotheken - Übersicht.....	82
5.13.2	Bibliotheken für BX9000, BC9020, BC9050, BC9120 - Übersicht.....	84
5.13.3	TcBaseBCxx50	85
5.13.4	TcBaseBX9000.....	89
5.14	Programmübertragung	115
5.14.1	Programmübertragung über Ethernet.....	115
5.14.2	Programmübertragung über die serielle Schnittstelle.....	116
5.15	Prozessabbild	118
5.15.1	Feldbusprozessabbild.....	118
6	Ethernet.....	119
6.1	Systemvorstellung	119
6.1.1	Ethernet	119
6.2	ModbusTCP.....	121
6.2.1	ModbusTCP-Protokoll.....	121
6.2.2	Modbus TCP-Interface.....	122
6.2.3	Fehlerantwort des ModbusTCP Slaves (BK9000, BX/BC9xx0, BC9191, IP/ILxxxx-B/C900, EK9000).....	123
6.2.4	ModbusTCP-Funktionen.....	124
6.3	ADS-Kommunikation	127
6.3.1	ADS-Kommunikation	127
6.3.2	ADS-Protokoll	128
6.3.3	ADS-Dienste	129

7 Fehlerbehandlung und Diagose.....	133
7.1 Diagnose	133
7.2 Diagnose-LEDs	134
8 Anhang	138
8.1 BC9xx0 - Erste Schritte	138
8.2 Umstieg zwischen den Controllern	144
8.3 Allgemeine Betriebsbedingungen	146
8.4 Prüfnormen für Geräteprüfung	148
8.5 Literaturverzeichnis	149
8.6 Abkürzungsverzeichnis.....	150
8.7 Support und Service	151

1 Vorwort

1.1 Hinweise zur Dokumentation

Zielgruppe

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

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

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

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

Disclaimer

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

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

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

Marken

Beckhoff®, TwinCAT®, TwinCAT/BSD®, TC/BSD®, EtherCAT®, EtherCAT G®, EtherCAT G10®, EtherCAT P®, Safety over EtherCAT®, TwinSAFE®, XFC®, XTS® und XPlanar® sind eingetragene und lizenzierte Marken der Beckhoff Automation GmbH. Die Verwendung anderer in dieser Dokumentation enthaltenen Marken oder Kennzeichen durch Dritte kann zu einer Verletzung von Rechten der Inhaber der entsprechenden Bezeichnungen führen.

Patente

Die EtherCAT-Technologie ist patentrechtlich geschützt, insbesondere durch folgende Anmeldungen und Patente: EP1590927, EP1789857, EP1456722, EP2137893, DE102015105702 mit den entsprechenden Anmeldungen und Eintragungen in verschiedenen anderen Ländern.



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

Copyright

© Beckhoff Automation GmbH & Co. KG, Deutschland.

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

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

1.2 Sicherheitshinweise

Sicherheitsbestimmungen

Beachten Sie die folgenden Sicherheitshinweise und Erklärungen!

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

Haftungsausschluss

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

Qualifikation des Personals

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

Erklärung der Hinweise

In der vorliegenden Dokumentation werden die folgenden Hinweise verwendet.
Diese Hinweise sind aufmerksam zu lesen und unbedingt zu befolgen!

GEFAHR

Akute Verletzungsgefahr!

Wenn dieser Sicherheitshinweis nicht beachtet wird, besteht unmittelbare Gefahr für Leben und Gesundheit von Personen!

WARNUNG

Verletzungsgefahr!

Wenn dieser Sicherheitshinweis nicht beachtet wird, besteht Gefahr für Leben und Gesundheit von Personen!

VORSICHT

Schädigung von Personen!

Wenn dieser Sicherheitshinweis nicht beachtet wird, können Personen geschädigt werden!

HINWEIS

Schädigung von Umwelt/Geräten oder Datenverlust

Wenn dieser Hinweis nicht beachtet wird, können Umweltschäden, Gerätebeschädigungen oder Datenverlust entstehen.



Tipp oder Fingerzeig

Dieses Symbol kennzeichnet Informationen, die zum besseren Verständnis beitragen.

1.3 Ausgabestände der Dokumentation

Version	Kommentar
2.4.0	<ul style="list-style-type: none"> • Kapitel <i>Technische Daten - BC</i> aktualisiert • Kapitel <i>Hinweise zum ESD-Schutz</i> hinzugefügt • Kapitel <i>Entsorgung</i> hinzugefügt • Kapitel <i>Hinweise zur Informationssicherheit</i> hinzugefügt • Neue Titelseite
2.3.0	<ul style="list-style-type: none"> • Kapitel <i>Adresseinstellung über BootP-Server</i> aktualisiert • Dokumentstruktur aktualisiert
2.2.0	<ul style="list-style-type: none"> • Download Links aktualisiert
2.1.0	<ul style="list-style-type: none"> • Technische Daten aktualisiert
2.0.0	<ul style="list-style-type: none"> • Migration • Technische Daten aktualisiert
1.0.0	<ul style="list-style-type: none"> • Erste Version

Firmware BC9050

Firmware	Kommentar
B4	Optimierungen
B0	Erste Version

Firmware BC9020

Firmware	Kommentar
B4	ADS Index Group 0x4040 implementiert
B3	Optimierung der seriellen Schnittstelle
B2	Erste Version
B0	Interne Version

Firmware BC9120

Firmware	Kommentar
B4	ADS Index Group 0x4040 implementiert
B3	Optimierung der seriellen Schnittstelle
B2	Erste Version
B1	Interne Version

1.4 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>.

2 Produktübersicht

2.1 BC9020, BC9050 und BC9120 - Übersicht

Die Busklemmen-Controller der Serie BC9xx0 sind Steuerungen mit Ethernet-Interface und mit hohem Flexibilitätsgrad. Sie können auch autark betrieben werden.

Die Busklemmen-Controller der BC-Familie bestehen aus einem programmierbaren IEC 61131-3 Controller und dem K-Bus Interface für den Anschluss der Beckhoff Busklemmen sowie eines Ethernet Interface.

Die Busklemmen können wie gewohnt direkt angeschlossen werden. Durch das umfangreiche Spektrum an verschiedenen E/As kann jedes Eingangssignal gelesen und jedes benötigte Ausgangssignal erzeugt werden. Dadurch sind mit dieser Kleinststeuerung vielfältige Automatisierungsaufgaben lösbar, von der Garagentorsteuerung bis hin zur autarken Temperaturregelung an einer Spritzgussmaschine. Auch im Hinblick auf ein modulares Maschinenkonzept ist die BC-Familie hervorragend einzusetzen. Im Verbund kann der Busklemmen-Controller über die Feldbusschnittstelle mit anderen Maschinenteilen Daten austauschen.

● Busklemmen erforderlich

i An den K-Bus der Busklemmen-Controller muss mindestens eine Busklemme mit Prozessabbild und eine Endklemme gesteckt werden.

Feldbus-Interface

Die Varianten der Busklemmen-Controller der Serie BC9xx0 unterscheiden sich durch die unterschiedlichen Feldbusschnittstellen mit und ohne Switch sowie in der maximal möglichen Programmgröße.

- [BC9050](#): 1 x Ethernet Interface, 48 kByte Programmspeicher
- [BC9020](#): 1 x Ethernet Interface, 128 kByte Programmspeicher
- [BC9120](#): 2 x Ethernet Interface, 128 kByte Programmspeicher

Programmierung

Programmiert werden die BC-Geräte nach der leistungsfähigen Norm IEC 61131-3. Wie auch bei allen anderen Beckhoff Steuerungen ist die Automatisierungssoftware TwinCAT Grundlage für die Parametrierung und Programmierung. Dem Anwender stehen also die gewohnten TwinCAT-Werkzeuge, wie z. B. SPS-Programmiersoberfläche, System Manager zur Verfügung. Der Datenaustausch erfolgt wahlweise über die serielle Schnittstelle (COM1) oder über Ethernet.

Konfiguration

Die Konfiguration erfolgt ebenfalls mit TwinCAT. Über den System Manager kann das Feldbus-Interface konfiguriert und parametrierung werden. Alle angeschlossenen Geräte und Busklemmen können vom System Manager ausgelesen werden. Die Konfiguration wird nach der Parametrierung über die serielle Schnittstelle auf den BC gespeichert. Diese erstellte Konfiguration kann auch wieder ausgelesen werden.

● Erforderliche TwinCAT-Version

i BC9020, BC9050, BC9120 werden ab TwinCAT 2.10 Build 1322 unterstützt.

2.1.1 BC9020 - Einführung

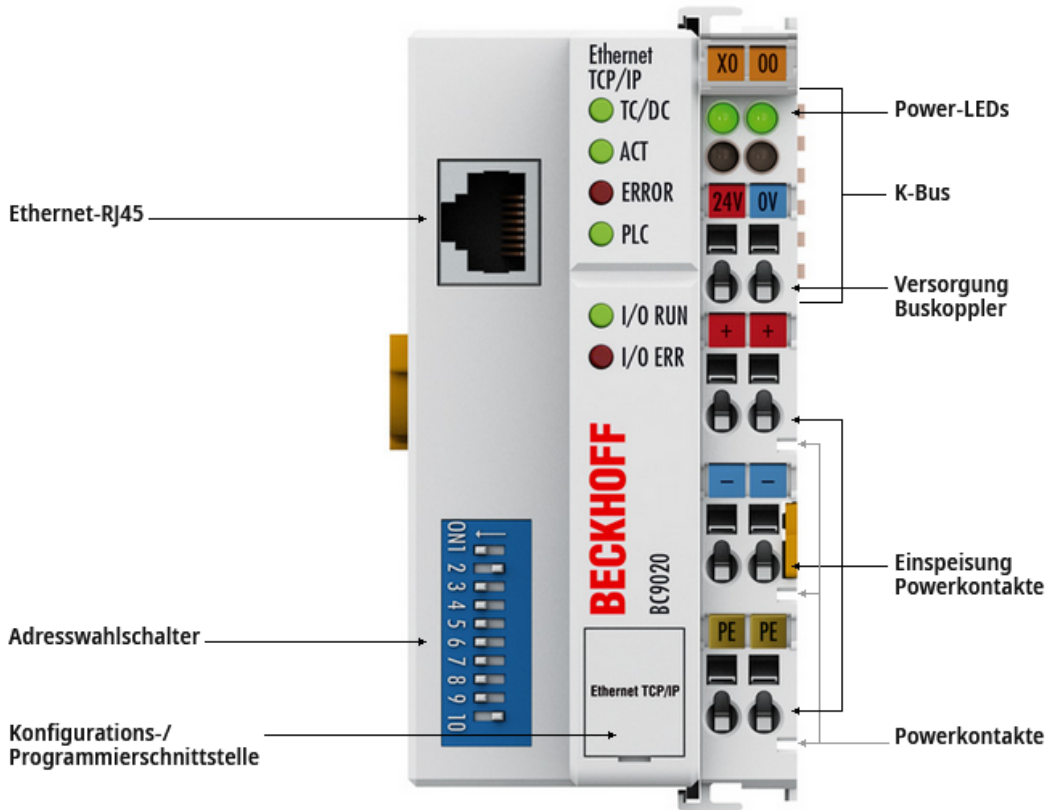


Abb. 1: BC9020

Der Busklemmen-Controller BC9020 ist ein Buskoppler mit integrierter SPS-Funktionalität und besitzt eine Feldbusschnittstelle für Ethernet. Er ist ein intelligenter Slave und kann als dezentrale Intelligenz im Ethernet-System eingesetzt werden. Der BC9020 ermöglicht, in Verbindung mit der K-Bus-Verlängerung, den Anschluss von bis zu 255 Busklemme.

In Bezug auf seine Leistung liegt der Busklemmen Controller BC9020 zwischen dem BC9050 und dem BX9000. Die Busklemmen Controller der Serie BX unterscheiden sich durch eine größere Speicherausstattung und zusätzliche Schnittstellen zur Integration weiterer Peripheriegeräte.

2.1.2 BC9050 - Einführung

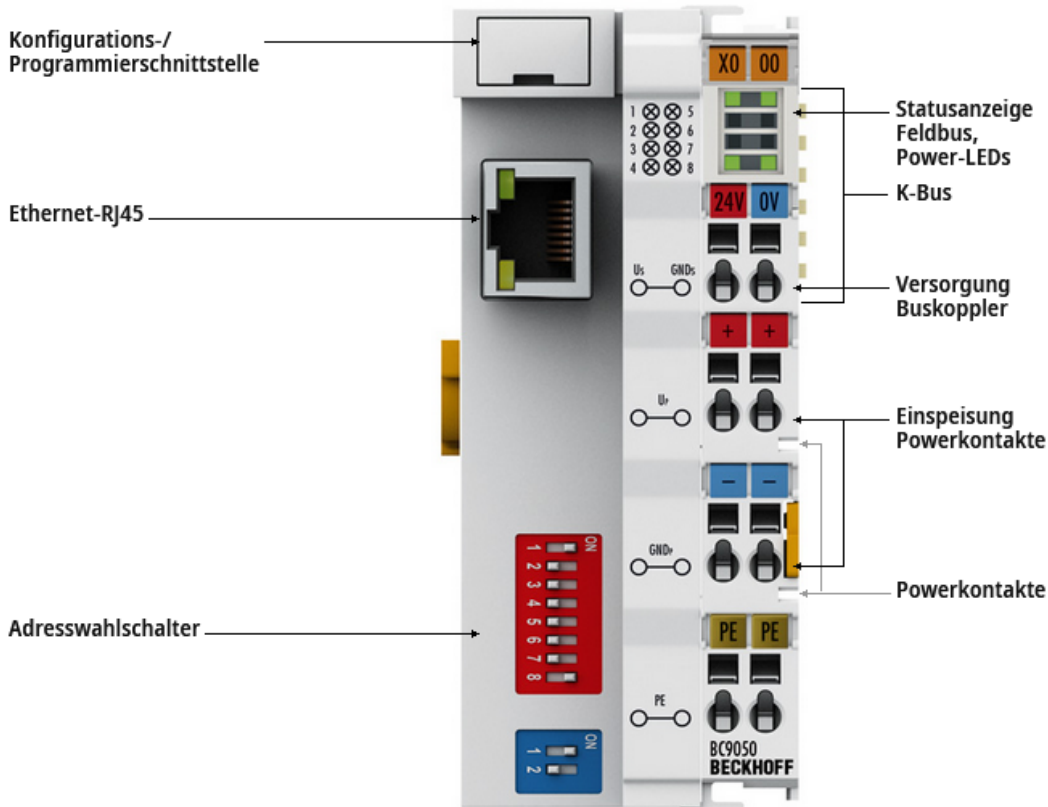


Abb. 2: BC9050

Der Busklemmen-Controller BC9050 im kompakten Gehäuse ist ein Buskoppler mit integrierter SPS-Funktionalität und einer Feldbusschnittstelle für Ethernet. Er ist ein intelligenter Slave, der als dezentrale Intelligenz im Ethernet-System eingesetzt werden kann. Eine Einheit besteht aus dem Busklemmen Controller, einer beliebigen Anzahl von 1 bis 64 Klemmen und einer Busendklemme. Mit der Klemmenbusverlängerung ist der Anschluss von bis zu 255 Busklemmen möglich.

2.1.3 BC9120 - Einführung

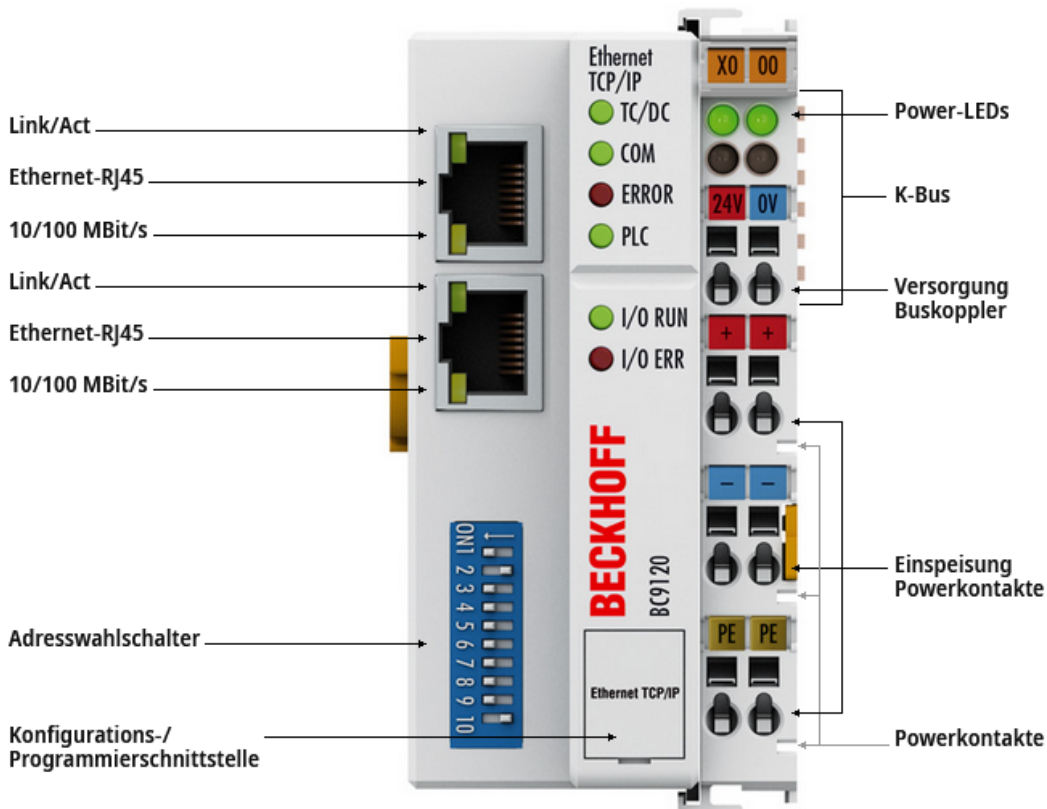


Abb. 3: BC9120

Der Busklemmen Controller BC9120 ist ein Buskoppler mit integrierter SPS-Funktionalität und besitzt eine Feldbusschnittstelle für Ethernet. Er ist ein intelligenter Slave und kann als dezentrale Intelligenz im Ethernet-System eingesetzt werden. Der BC9120 ermöglicht, in Verbindung mit der K-Bus-Verlängerung, den Anschluss von bis zu 255 Busklemmen.

In Bezug auf seine Leistung liegt der Busklemmen Controller BC9120 zwischen dem BC9050 und dem BX9000. Die Busklemmen Controller der Serie BX unterscheiden sich durch eine größere Speicherausstattung und zusätzliche Schnittstellen zur Integration weiterer Peripheriegeräte.

Der BC9120 enthält einen zusätzlichen RJ45-Port. Beide Ethernet-Ports arbeiten als Zweikanal-Switch. Damit können die I/O-Stationen, anstatt in der klassischen Sterntopologie, als Linientopologie aufgebaut werden. Der Verdrahtungsaufwand wird hierdurch bei vielen Anwendungen erheblich vereinfacht, und die Kabelkosten werden reduziert. Die maximale Entfernung zwischen zwei Kopplern/Controllern beträgt 100 m. Bis zu 20 BC9120 sind kaskadierbar, sodass man eine maximale Linie von 2 km erreichen kann.

2.2 Technische Daten

2.2.1 Technische Daten - Ethernet

Systemdaten	Ethernet (BC9050, BC9020, BC9120)
Anzahl der E/A-Module	steuerungsabhängig
Anzahl der E/A-Punkte	steuerungsabhängig
Übertragungsmedium	4 x 2 Kupferkabel (Twisted-Pair), Kategorie 5 (100 MBaud)
Leitungslänge	100 m vom Switch bis zum Busklemmen Controller
Übertragungsrate	10/100 MBaud
Topologie	sternförmige Verkabelung (beim BC9120 auch Linie möglich, maximal 20 Geräte in einer Linie)

Verbindungen	Ethernet (BC9050, BC9020, BC9120)
Anzahl der ADS-Verbindungen über TCP/IP	5
Anzahl der ADS-Verbindungen über UDP/IP	1
Anzahl der ModbusTCP-Verbindungen	3
SMTP (Simple Mail Transfer Protocol)	ja
SNTP (Simple Network Time Protocol)	ja
BootP/DHCP	ja/ja
UDP-Socket-Verbindungen	3
TCP-Socket-Verbindungen	3

● Reduzieren Sie die Anzahl der TCP/IP-Verbindungen auf ein Minimum

i Die Gesamte Anzahl der TCP/IP-Verbindungen sollte auf ein Minimum reduziert werden. Je weniger sie nutzen, desto mehr Zeit bleibt dem Controller für seine eigentliche Aufgabe. Verschenden Sie nicht unnötig TCP/IP-Verbindungen, sondern versuchen Sie mit möglichst wenigen TCP/IP-Verbindungen auszukommen. Wenn möglich nutzen Sie statt TCP/IP- besser UDP/IP-Kommunikation, zum Beispiel für ADS. Kommunizieren Sie bei der TCP/IP- und auch bei der UDP/IP-Verbindung mit "vernünftigen" Zeitintervallen. Beispiel: Die Task-Zeit auf dem Controller ist 20 ms. Hier sollten Sie dann mit minimal 40 ms Zykluszeit bei Ihrem Client arbeiten.

2.2.2 Technische Daten - Busklemmen-Controller

Technische Daten	BC9050	BC9020	BC9120
Prozessor	16 Bit Mikrocontroller		
Diagnose LEDs	2 x Spannungsversorgung, 2 x K-Bus, 4 x Bus und Status LEDs		
Konfigurations- und Programmiersoftware	TwinCAT 2.10 ab Build 1322		
Feldbus-Interface	Ethernet		
Klemmenbus (K-Bus)	64 (255 mit K-Bus-Verlängerung)		
Digitale Peripheriesignale	2040 Ein-/Ausgänge		
Analoge Peripheriesignale	512 Ein-/Ausgänge		
Konfigurationsmöglichkeit	über TwinCAT oder die Steuerung		
maximale Byte-Anzahl Feldbus	512 Byte Eingangsdaten, 512 Byte Ausgangsdaten		
maximale Byte-Anzahl SPS	2048 Byte Eingangsdaten, 2048 Byte Ausgangsdaten		
Feldbus-Anschluss	RJ45	RJ45	2 x RJ45 Switch
Spannungsversorgung (Us)	24 V _{DC} (-15%/+20%)		
Eingangsstrom (Us)	70 mA + (ges. K-Bus Strom)/4, 4,500 mA max	85 mA + (ges. K-Bus Strom)/4	90 mA + (ges. K-Bus Strom)/4
Einschaltstrom (Us)	ca. 2,5 x Dauerstrom		
K-Bus-Strom (5 V)	maximal 1750 mA		
Spannung Powerkontakt (Up)	maximal 24 V _{DC} (-15%/+20%)		
Stromlast Powerkontakt (Up)	maximal 10 A		
Spannungsfestigkeit	500 V (Powerkontakt/Versorgungsspannung/Feldbus)		
Anschlussquerschnitt	0,08 mm ² ... 2,5 mm ² AWG 28-14 8 bis 9 mm Abisolierlänge		
Abmessungen (B x H x T)	44 x 100 x 68 mm	44 x 100 x 70 mm	
Gewicht	ca. 100 g	ca. 170 g	
Montage [► 24]	auf 35 mm Tragschiene nach EN 60715		
zulässiger Umgebungstemperaturbereich im Betrieb	0°C ... +55°C	-25°C ... +60°C	
zulässiger Umgebungstemperaturbereich bei Lagerung	-25°C ... +85°C	-40°C ... +85°C	
Relative Feuchte	95% ohne Betauung		
Vibrations-/ Schockfestigkeit	gemäß EN 60068-2-6 / EN 60068-2-27		
EMV-Festigkeit / Aussendung	gemäß EN 61000-6-2 / EN 61000-6-4		
Einbaulage	beliebig		
Schutzart	IP20		
Zulassungen	CE, UKCA, cULus, EAC, ATEX [► 35]	CE, UKCA, cULus, EAC, GL, ATEX [► 36]	

*) Real zutreffende Zulassungen/Kennzeichnungen siehe seitliches Typenschild (Produktbeschriftung).

Ex-Kennzeichnung

Standard	Kennzeichnung
ATEX	II 3 G Ex nA IIC T4 Gc

2.2.3 Technische Daten - SPS

PLC Daten	BC9050	BC9020	BC9120
Programmiermöglichkeit	über Programmierschnittstelle oder über Feldbus		
Programmspeicher	48 kByte	128 kByte	
Source-Code-Speicher	128 kByte	256 kByte	
Datenspeicher	32 kByte	128 kByte	
Remanente Daten	2 kByte		
Persistente Daten	1000 Byte		
SPS-Zykluszeit	ca. 0,85 ms für 1000 AWL-Befehle (ohne E/A-Zyklus)		
Programmiersprachen	IEC 6-1131-3 (AWL, KOP, FUP, ST, AS)		
Laufzeit	1 SPS-Task		
Online Change	Ja		
Up/Down Load Code	Ja/Ja		

2.3 Prinzip der Busklemme

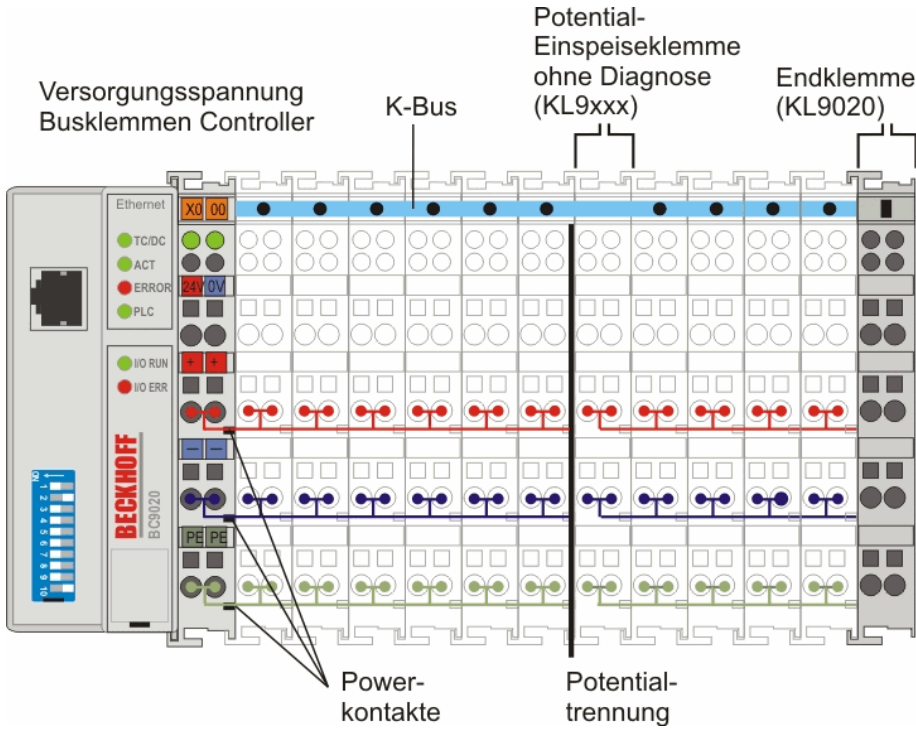


Abb. 4: BC9020, BC9120 - Prinzip der Busklemmen

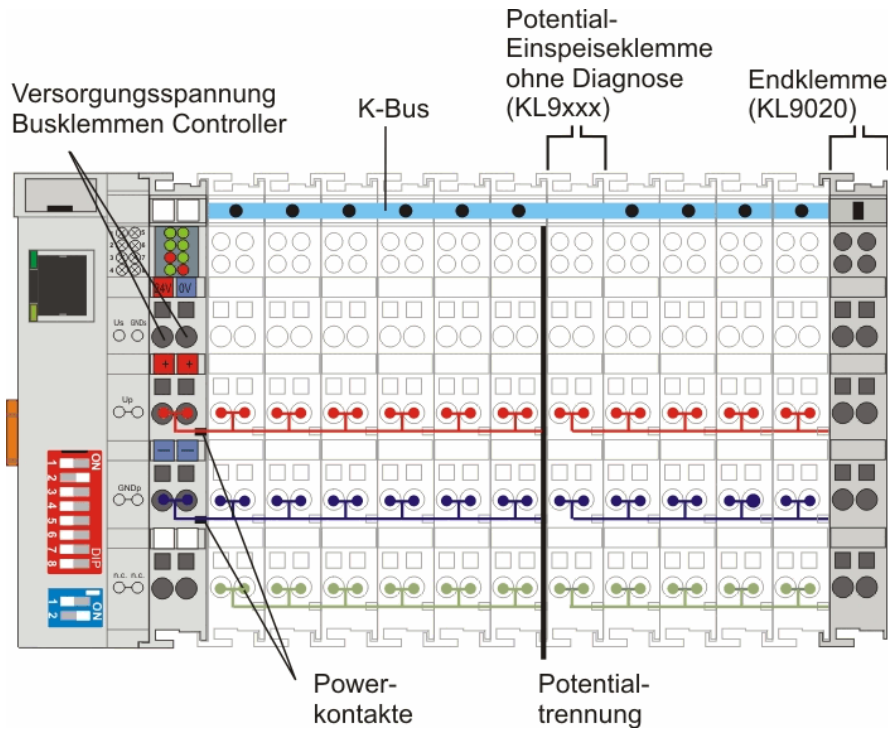


Abb. 5: BC9050 - Prinzip der Busklemme

2.4 Das Beckhoff Busklemmensystem

Bis zu 256 Busklemmen mit ein bis 16 E/A-Kanälen für jede Signalform

Das Busklemmen-System ist das universelle Bindeglied zwischen einem Feldbus-System und der Sensor / Aktuator - Ebene. Eine Einheit besteht aus einem Buskoppler als Kopfstation und bis zu 64 elektronischen Reihenklemmen, wovon die letzte eine Endklemme ist. Mit der K-Bus Erweiterung können bis zu 255 Busklemmen angeschlossen werden. Für jede technische Signalform stehen Klemmen mit ein, zwei, vier oder acht E/A-Kanälen zur Verfügung, die beliebig gemischt werden können. Dabei haben alle Klemmentypen die gleiche Bauform, wodurch der Projektierungsaufwand sehr gering gehalten wird. Bauhöhe und Tiefe sind auf kompakte Klemmenkästen abgestimmt.

Dezentrale Verdrahtung der E/A-Ebene

Die Feldbustechnik erlaubt den Einsatz kompakter Steuerungsbauformen. Die E/A-Ebene muss nicht bis zur Steuerung geführt werden. Die Verdrahtung der Sensoren und Aktuatoren ist dezentral mit minimalen Kabellängen durchführbar. Der Installationsstandort der Steuerung kann im Bereich der Anlage beliebig gewählt werden.

Industrie-PCs als Steuerung

Durch den Einsatz eines Industrie-PCs als Steuerung lässt sich das Bedien- und Beobachtungselement in der Hardware der Steuerung realisieren. Der Standort der Steuerung kann deshalb ein Bedienpult, eine Leitwarte oder ähnliches sein. Die Busklemmen stellen die dezentrale Ein-/Ausgabebene der Steuerung im Schaltschrank und untergeordneten Klemmenkästen dar. Neben der Sensor/Aktuator-Ebene wird auch der Leistungsteil der Anlage über das Bussystem gesteuert. Die Busklemme ersetzt die konventionelle Reihenklemme als Verdrahtungsebene im Schaltschrank. Der Schaltschrank kann kleiner dimensioniert werden.

Buskoppler für alle gängigen Bussysteme

Das Beckhoff Busklemmen-System vereint die Vorteile eines Bussystems mit den Möglichkeiten der kompakten Reihenklemme. Busklemmen können an allen gängigen Bussystemen betrieben werden und verringern so die Teilevielfalt in der Steuerung. Dabei verhalten sich Busklemmen wie herkömmliche Anschaltungen dieses Bussystems. Alle Leistungsmerkmale des jeweiligen Bussystems werden unterstützt.

Montage auf genormten Tragschienen

Die einfache und platzsparende Montage auf einer genormten Tragschiene (EN 60715, 35 mm) und die direkte Verdrahtung von Aktoren und Sensoren ohne Querverbindungen zwischen den Klemmen standardisiert die Installation. Dazu trägt auch das einheitliche Beschriftungskonzept bei.

Die geringe Baugröße und die große Flexibilität des Busklemmen-Systems ermöglichen den Einsatz überall dort, wo auch eine Reihenklemme zur Anwendung kommt. Jede Art von Ankopplung, wie analoge, digitale, serielle oder der Direktanschluss von Sensoren kann realisiert werden.

Modularität

Die modulare Zusammenstellung der Klemmleiste mit Busklemmen verschiedener Funktionen begrenzt die Zahl der ungenutzten Kanäle auf maximal einen pro Funktion. Die Anzahl von zwei Kanälen in einer Klemme trifft das Optimum zwischen der Zahl der ungenutzten Kanäle und den Kosten pro Kanal. Auch die Möglichkeit der Potentialtrennung durch Einspeiseklemmen hilft, die Anzahl der ungenutzten Kanäle gering zu halten.

Anzeige des Kanalzustands

Die integrierten Leuchtdioden zeigen in Sensor/Aktuator-Nähe den Zustand des entsprechenden Kanals an.

K-Bus

Der K-Bus ist der Datenweg innerhalb der Klemmleiste. Über sechs Kontakte an den Seitenwänden der Klemmen wird der K-Bus vom Buskoppler durch alle Klemmen geführt. Die Endklemme schließt den K-Bus ab. Der Benutzer muss sich keinerlei Wissen über die Funktion des K-Bus oder die interne Arbeitsweise von Klemmen und Buskoppler aneignen. Viele lieferbare Software-Tools erlauben eine komfortable Projektierung, Konfiguration und Bedienung.

Potential-Einspeiseklemmen für potentialgetrennte Gruppen

Über drei Powerkontakte wird die Betriebsspannung an die nachfolgenden Klemmen weitergegeben. Durch den Einsatz von Potential-Einspeiseklemmen, können Sie die Klemmleiste in beliebige potentialgetrennte Gruppen gliedern. Die Potential-Einspeiseklemmen werden bei der Ansteuerung der Klemmen nicht berücksichtigt, sie dürfen an beliebiger Stelle in die Klemmleiste eingereiht werden.

In einem Klemmenblock können Sie bis zu 64 Busklemmen einsetzen und diesen über die K-Busverlängerung auf bis zu 256 Busklemmen erweitern. Dabei werden Potential-Einspeiseklemmen mitgezählt, die Endklemme nicht.

Buskoppler für verschiedene Feldbus-Systeme

Verschiedene Buskoppler lassen sich einsetzen, um die elektronische Klemmleiste schnell und einfach an unterschiedliche Feldbus-Systeme anzukoppeln. Auch eine nachträgliche Umrüstung auf ein anderes Feldbus-System ist möglich. Der Buskoppler übernimmt alle Kontroll- und Steuerungsaufgaben, die für den Betrieb der angeschlossenen Busklemmen notwendig sind. Die Bedienung und Konfiguration der Busklemmen wird ausschließlich über den Buskoppler durchgeführt. Die eingestellten Parameter werden jedoch spannungsausfallsicher in den jeweiligen Busklemmen gespeichert. Feldbus, K-Bus und E/A-Ebene sind galvanisch getrennt.

Wenn der Datenaustausch über den Feldbus zeitweise gestört ist oder ausfällt, bleiben Registerinhalte (wie z. B. Zählerstände) erhalten, digitale Ausgänge werden gelöscht und analoge Ausgänge nehmen einen Wert an, der bei der Inbetriebnahme für jeden Ausgang konfigurierbar ist. Die Default-Einstellung der analogen Ausgänge ist 0 V bzw. 0 mA. Digitale Ausgänge fallen in einen inaktiven Zustand zurück. Die Timeout-Zeiten der Buskoppler entsprechen den für das Feldbus-System üblichen Zeiten. Bei der Umstellung auf ein anderes Bussystem beachten Sie im Falle großer Zykluszeiten des Bussystems die Änderung der Timeout-Zeiten.

Die Schnittstellen

Ein Buskoppler besitzt sechs unterschiedliche Anschlussmöglichkeiten. Diese Schnittstellen sind als Steckverbindungen und Federkraftklemmen ausgelegt.

3 Montage und Verdrahtung

3.1 Hinweise zum ESD-Schutz

HINWEIS

Zerstörung der Geräte durch elektrostatische Aufladung möglich!

Die Geräte enthalten elektrostatisch gefährdete Bauelemente, die durch unsachgemäße Behandlung beschädigt werden können.

- Sie müssen beim Umgang mit den Komponenten elektrostatisch entladen sein; vermeiden Sie außerdem die Federkontakte (s. Abb.) direkt zu berühren.
- Vermeiden Sie den Kontakt mit hoch isolierenden Stoffen (Kunstfaser, Kunststofffolien etc.)
- Beim Umgang mit den Komponenten ist auf gute Erdung der Umgebung zu achten (Arbeitsplatz, Verpackung und Personen)
- Jede Busstation muss auf der rechten Seite mit der Endklemme KL9010 abgeschlossen werden, um Schutzart und ESD-Schutz sicher zu stellen.

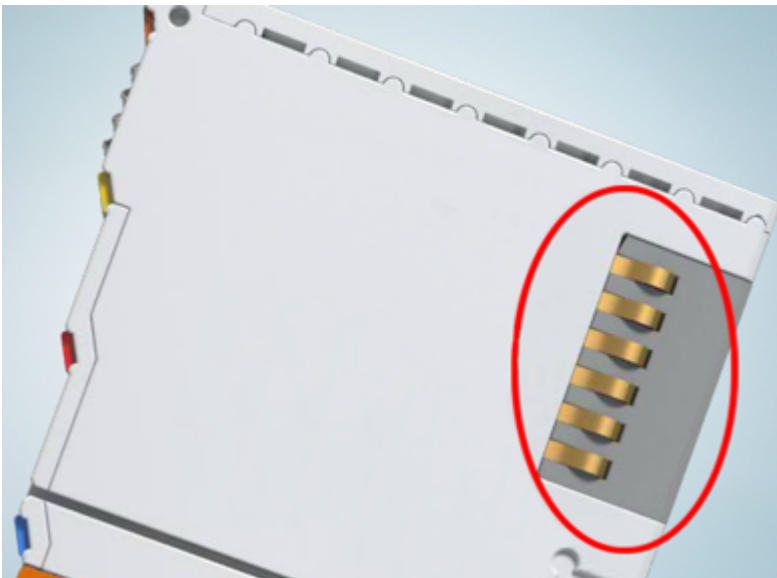


Abb. 6: Federkontakte der Beckhoff I/O-Komponenten

3.2 Montage

⚠️ WARNUNG

Verletzungsgefahr durch Stromschlag und Beschädigung des Gerätes möglich!

Setzen Sie das Busklemmen-System in einen sicheren, spannungslosen Zustand, bevor Sie mit der Montage, Demontage oder Verdrahtung der Komponenten beginnen!

3.2.1 Abmessungen

Das System der Beckhoff-Busklemmen zeichnet sich durch geringes Bauvolumen und hohe Modularität aus. Für die Projektierung muss ein Buskoppler und eine Anzahl von Busklemmen vorgesehen werden. Die Baumaße der Buskoppler sind unabhängig vom Feldbus-System.

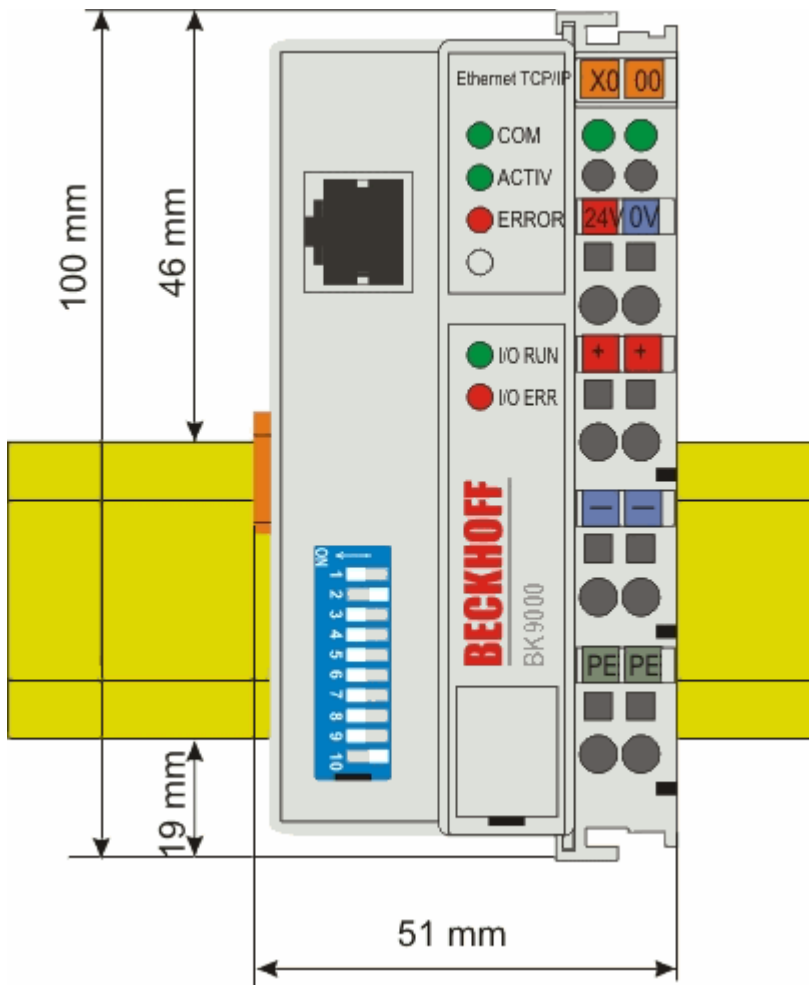


Abb. 7: BK9000, BK9100, BC9000, BC9020, BC9100, BC9120

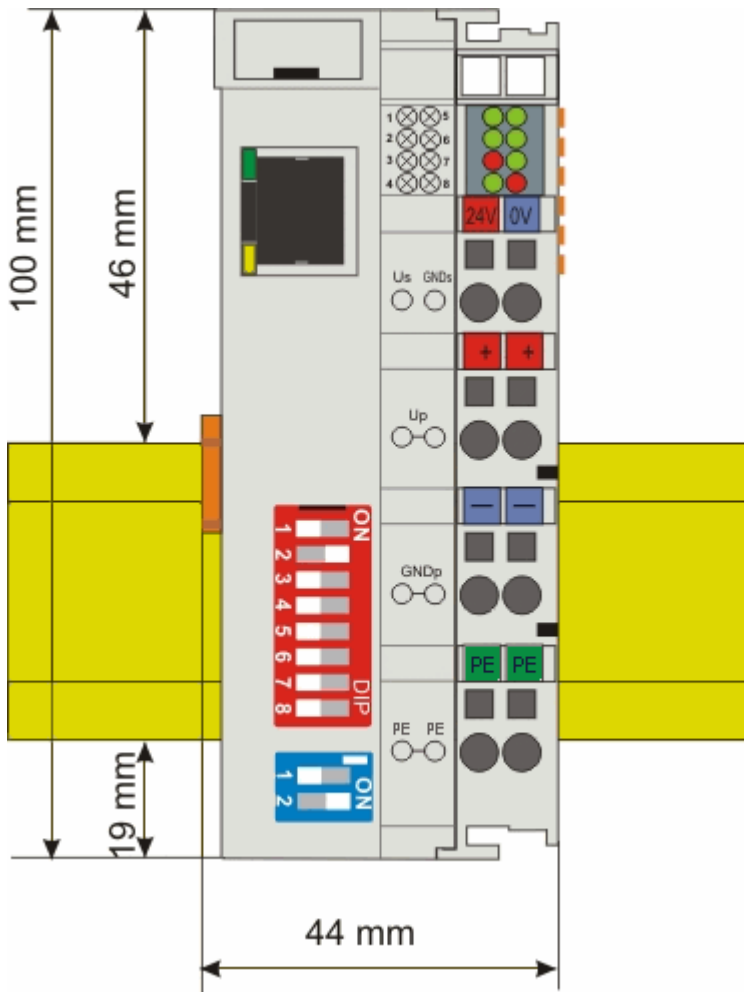


Abb. 8: BK9050, BC9050

Die Gesamtbreite der Anwendung setzt sich aus der Breite des Buskopplers, der Breite der verwendeten Busklemmen und der Busendklemme KL9010 zusammen. Die Busklemmen sind je nach Funktion 12 mm oder 24 mm breit. Die lichte Höhe von 68 mm wird durch die Frontverdrahtung in Abhängigkeit von der Drahtstärke um ca. 5 mm bis 10 mm überbaut.

3.2.2 Tragschienenmontage

Der Buskoppler und alle Busklemmen können durch leichten Druck auf eine 35 mm Tragschiene aufgerastet werden. Die einzelnen Gehäuse können durch einen Verriegelungsmechanismus nicht mehr abgezogen werden. Zum Entfernen von der Tragschiene entsichert die orangefarbene Zuglasche den Rastmechanismus und ermöglicht ein kraftloses Abziehen der Gehäuse.

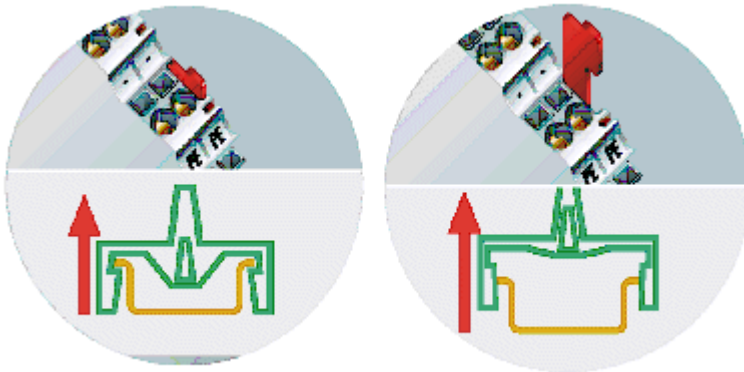


Abb. 9: Entsichern der Verriegelung durch die orangefarbene Zuglasche

An den Buskoppler können auf der rechten Seite bis zu 64 Busklemmen angereiht werden. Beim Zusammenstecken der Komponenten ist darauf zu achten, dass die Gehäuse mit Nut und Feder aneinander gesetzt zu montieren sind. Durch das Zusammenschieben der Gehäuse auf der Tragschiene kann keine funktionsfähige Verbindung hergestellt werden. Bei richtiger Montage ist kein nennenswerter Spalt zwischen den angereihten Gehäusen zu sehen.

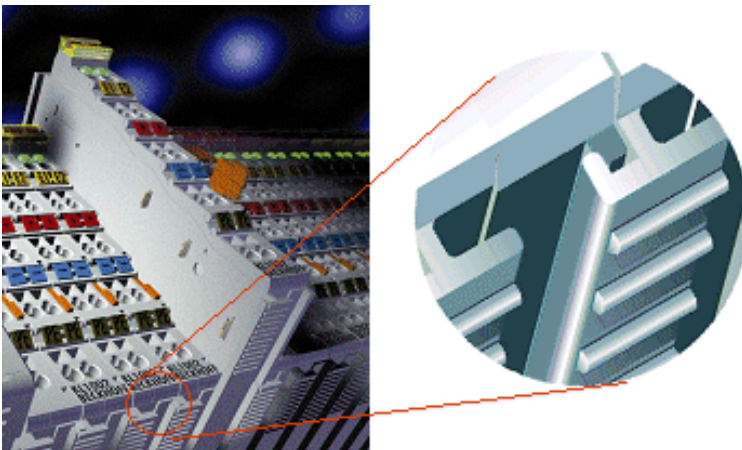


Abb. 10: Nut und Feder der Gehäuse

HINWEIS

Busklemmen nur im abgeschalteten Zustand ziehen oder stecken!

Das Ziehen und Stecken von Busklemmen aus dem Verbund ist nur im abgeschalteten Zustand zulässig. Die Elektronik der Busklemmen und des Buskopplers ist weitestgehend vor Zerstörungen geschützt, Fehlfunktionen und Schädigungen können beim Zusammenstecken unter Spannung jedoch nicht ausgeschlossen werden.

3.2.3 Entsorgung



Mit einer durchgestrichenen Abfalltonne gekennzeichnete Produkte dürfen nicht in den Hausmüll. Das Gerät gilt bei der Entsorgung als Elektro- und Elektronik-Altgerät. Die nationalen Vorgaben zur Entsorgung von Elektro- und Elektronik-Altgeräten sind zu beachten.

3.3 Verdrahtung

⚠️ WARNUNG

Verletzungsgefahr durch Stromschlag und Beschädigung des Gerätes möglich!

Setzen Sie das Busklemmen-System in einen sicheren, spannungslosen Zustand, bevor Sie mit der Montage, Demontage oder Verdrahtung der Komponenten beginnen!

3.3.1 Potentialgruppen, Isolationsprüfung und PE

Potentialgruppen

Ein Beckhoff Busklemmenblock verfügen in der Regel über drei verschiedene Potentialgruppen:

- Die Feldbusschnittstelle ist (außer bei einzelnen Low Cost Kopplern) galvanisch getrennt und bildet die erste Potentialgruppe.
- Buskoppler- / Busklemmen-Controller-Logik, K-Bus und Klemmenlogik bilden eine zweite galvanisch getrennte Potentialgruppe.
- Die Ein- und Ausgänge werden über die Powerkontakte gespeist und bilden weitere Potentialgruppen.

Gruppen von E/A-Klemmen lassen sich durch Potentialeinspeiseklemmen oder Trennklemmen zu weiteren Potentialgruppen zusammenfassen.

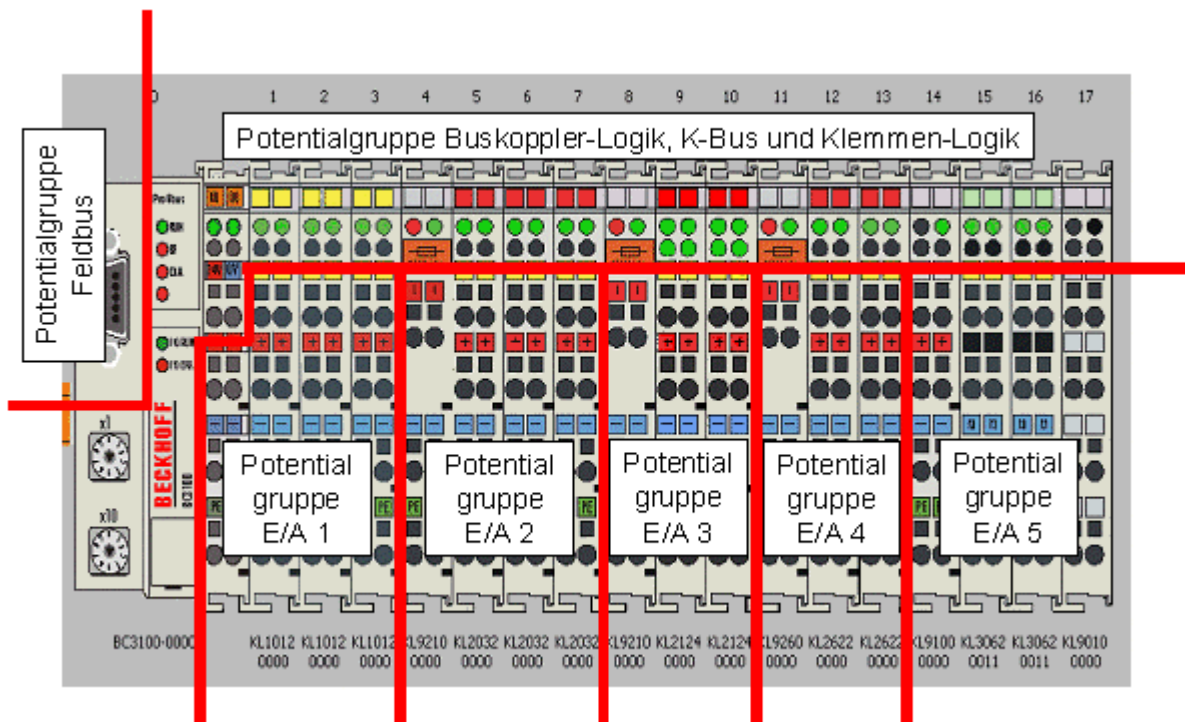


Abb. 11: Potentialgruppen eines Busklemmenblocks

Isolationsprüfung

Die Verbindung zwischen Buskoppler- / Busklemmen-Controller und Busklemmen wird durch das Zusammenstecken der Komponenten automatisch realisiert. Die Übertragung der Daten und die Versorgungsspannung der intelligenten Elektronik der Busklemmen übernimmt der K-Bus. Die Versorgung der Feldelektronik wird über die Powerkontakte durchgeführt. Die Powerkontakte stellen durch das Zusammenstecken eine Versorgungsschiene dar. Da einige Busklemmen (z. B. analoge Busklemmen oder digitale Vierkanal-Busklemmen) diese Powerkontakte nicht oder nicht vollständig durchschleifen, sind die Kontaktbelegungen der Busklemmen zu beachten.

Die Einspeiseklemmen unterbrechen die Powerkontakte und stellen den Anfang einer neuen Versorgungsschiene dar. Der Buskoppler- / Busklemmen-Controller kann auch zur Einspeisung der Powerkontakte eingesetzt werden.

PE-Powerkontakte

Der Powerkontakt mit der Bezeichnung PE kann als Schutzerde eingesetzt werden. Der Kontakt ist aus Sicherheitsgründen beim Zusammenstecken voreilend und kann Kurzschlussströme bis 125 A ableiten.

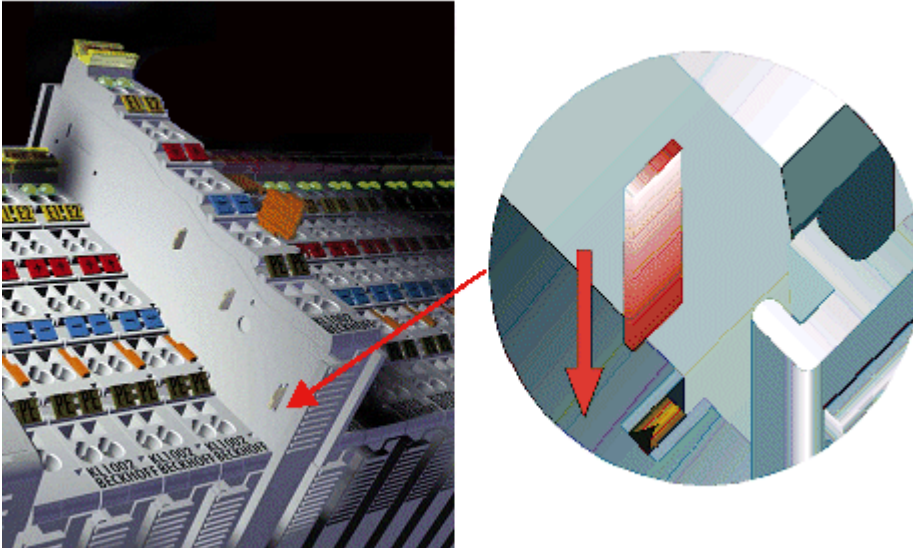


Abb. 12: Linksseitiger Powerkontakt

Es ist zu beachten, dass aus EMV-Gründen die PE-Kontakte kapazitiv mit der Tragschiene verbunden sind. Das kann zu falschen Ergebnissen und auch zur Beschädigung der Klemme bei der Isolationsprüfung führen (z. B. Isolationsdurchschlag an einem 230 V-Verbraucher zur PE-Leitung). Die PE-Zuleitung am Buskoppler- / Busklemmen-Controller muss zur Isolationsprüfung abgeklemmt werden. Um weitere Einspeisestellen für die Prüfung zu entkoppeln, können die Einspeiseklemmen aus dem Verbund der übrigen Klemmen mindestens 10 mm herausgezogen werden. Die PE-Zuleitungen müssen in diesem Fall nicht abgeklemmt werden.

Der Powerkontakt mit der Bezeichnung PE darf nicht für andere Potentiale verwendet werden.

3.3.2 Spannungsversorgung

BC9050

Versorgung von Busklemmen-Controller und Busklemmen (Us)

Der Busklemmen-Controller benötigt zum Betrieb eine Versorgungsspannung von 24 V_{DC}.

Der Anschluss findet über die oberen Federkraftklemmen mit der Bezeichnung 24 V und 0 V statt. Diese Versorgungsspannung versorgt die Elektronik der Buskoppler / Busklemmen-Controller sowie über den K-Bus die Elektronik der Busklemmen. Sie ist galvanisch von der Spannung der Feldebene getrennt.

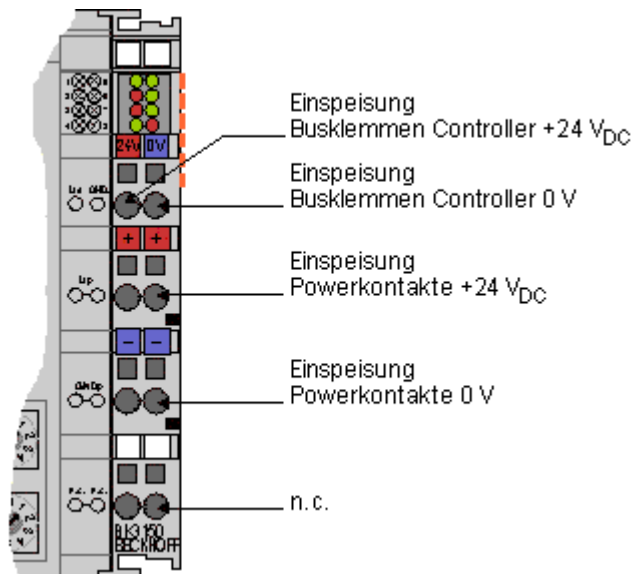


Abb. 13: Klemmstellen zur Versorgung des Busklemmen-Controllers

Versorgung der Powerkontakte (Up)

Die unteren sechs Anschlüsse mit Federkraftklemmen können zur Einspeisung der Peripherieversorgung benutzt werden. Die Federkraftklemmen sind paarweise mit einem Powerkontakt verbunden. Die Einspeisung zu den Powerkontakten besitzt keine Verbindung zur Spannungsversorgung der BC-Elektronik.

Die Federkraftklemmen sind für Drähte von 0,08 mm² bis 2,5 mm² Querschnitt ausgelegt.

Die paarweise Anordnung und die elektrische Verbindung zwischen den Speiseklemmkontakten ermöglicht das Durchschleifen der Anschlussdrähte zu unterschiedlichen Klemmpunkten. Die Strombelastung über den Powerkontakt darf 10 A nicht dauerhaft überschreiten. Die Strombelastbarkeit zwischen zwei Federkraftklemmen ist mit der Belastbarkeit der Verbindungsdrähte identisch.

Powerkontakte

An der rechten Seitenfläche des Busklemmen Controllers befinden sich drei Federkontakte der Powerkontaktverbindungen. Die Federkontakte sind in Schlitzen verborgen um einen Berührungsschutz sicher zu stellen. Durch das Anreihen einer Busklemme werden die Messerkontakte auf der linken Seite der Busklemme mit den Federkontakten verbunden. Die Nut-/Federführung an der Ober- und Unterseite der Busklemmen Controller und Busklemmen garantiert eine sichere Führung der Powerkontakte.

3.3.3 Ethernet-Topologien

BK9000, BK9050, BC9000, BC9020, BC9050

Diese Buskoppler und Busklemmen-Controller besitzen einen einfachen Ethernet-Anschluss. Dieser kann direkt an einen externen Switch angeschlossen werden. Somit ist der Aufbau der für Ethernet typischen Sterntopologie möglich.

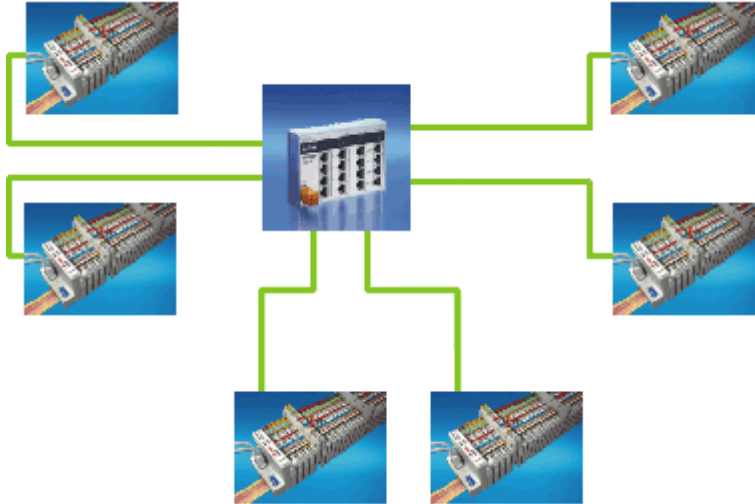


Abb. 14: Ethernet-Aufbau in Sterntopologie

BK9100, BC9100, BC9120, BC9191

Diese Buskoppler und Busklemmen-Controller besitzen einen internen Dreifach-Switch mit zwei externen und einem internen Port. Der interne Switch ermöglicht den einfachen Aufbau einer Linien-Topologie. Sie können maximal 20 BK9100/BC91x0/BC9191 zu einer physikalischen Linie hintereinander schalten. Dabei darf die Distanz zwischen zwei Ethernet-Geräten maximal 100 m betragen. Die gesamte Länge der so aufgebauten Linie darf also 2 km betragen. In diese Linie dürfen keine weiteren Switches eingebaut werden.

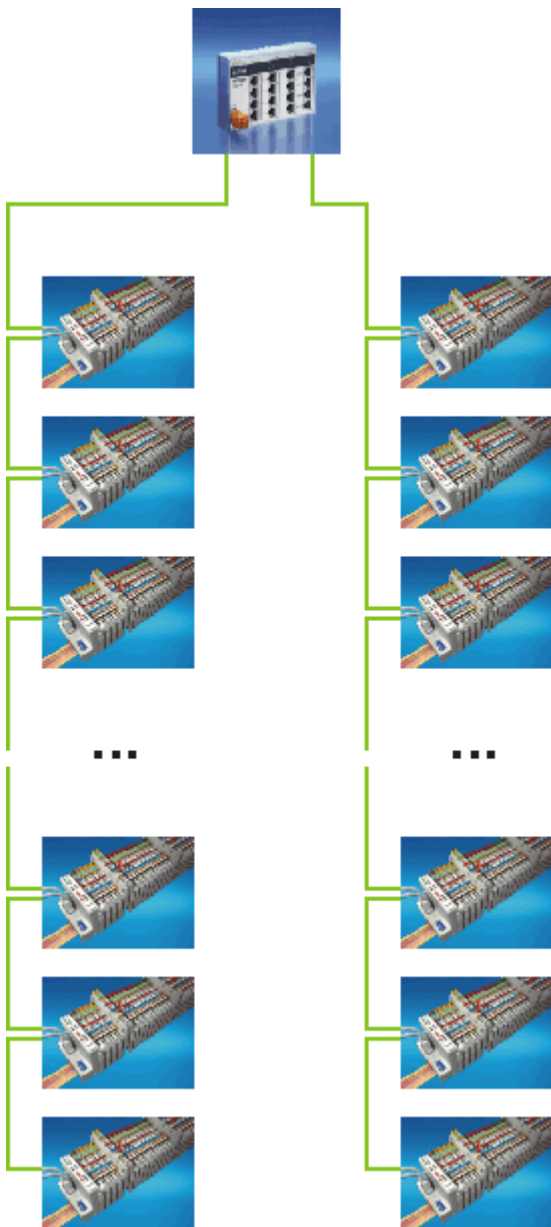


Abb. 15: Ethernet-Aufbau in Linien-Topologie

Natürlich ist auch mit diesen Buskopplern und Busklemmen-Controllern der Aufbau einer klassischen Sterntopologie möglich.

3.3.4 Ethernet-Anschluss

Der Anschluss an den Ethernet-Bus erfolgt über einen RJ45-Stecker (Westernstecker).



Abb. 16: RJ45-Stecker

Ethernet Verkabelung BC9020 oder BC9050

Verbindung über Hub oder Switch

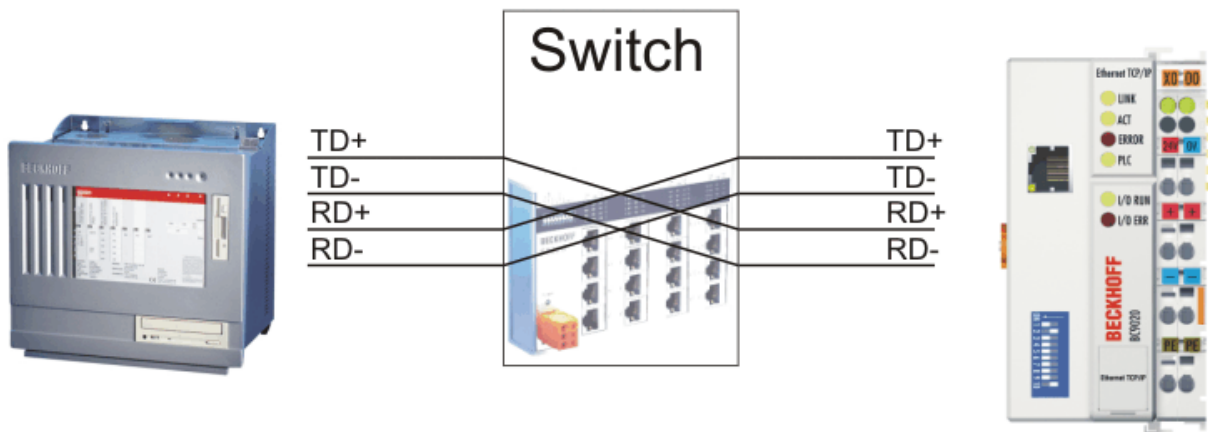


Abb. 17: Ethernet-Verbindung zwischen PC und BC9020, BC9050 über Hub oder Switch

Verbinden Sie die Netzwerkkarte des PCs über ein Standard-Ethernet-Kabel mit dem Hub/Switch und den Hub/Switch ebenfalls über ein Standard-Ethernet-Kabel mit dem Busklemmen Controller.

Direkte Verbindung zwischen PC mit Ethernet-Karte und BX9000

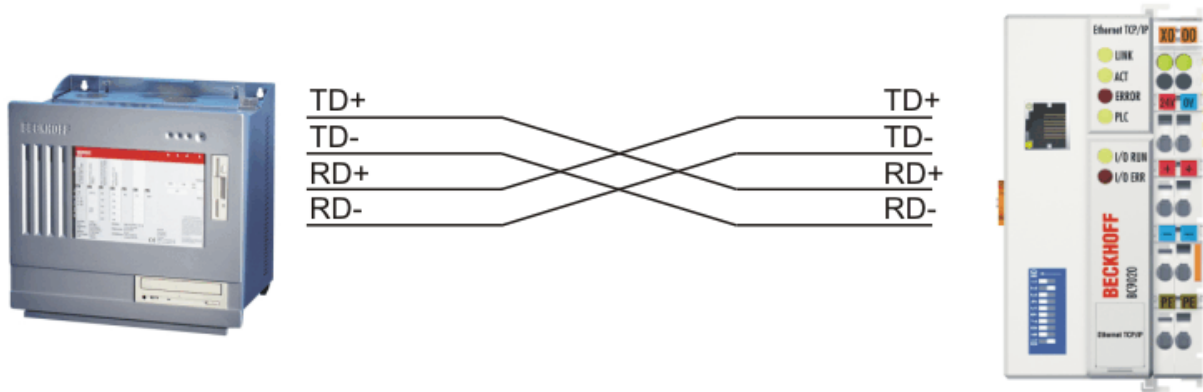


Abb. 18: Direkte Ethernet-Verbindung zwischen PC und BC9020, BC9050 über Cross-Over-Kabel

Um den PC direkt mit dem BC9020 oder BC9050 zu verbinden, müssen Sie ein Ethernet-Kabel mit gekreuzten Aderpaaren (Cross-Over-Kabel) verwenden.

Ethernet Verkabelung BC9120

Verbindung über Hub oder Switch

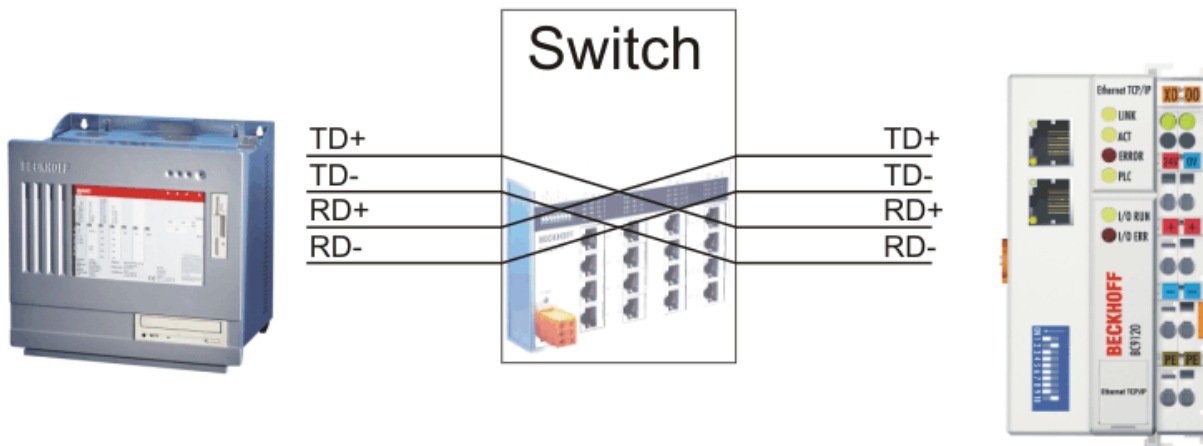


Abb. 19: Ethernet-Verbindung zwischen PC und BC9120 über Hub oder Switch

Verbinden Sie die Netzwerkkarte des PCs über ein Standard-Ethernet-Kabel mit dem Hub/Switch und den Hub/Switch ebenfalls über ein Standard-Ethernet-Kabel mit dem Busklemmen Controller.

Direkte Verbindung zwischen PC mit Ethernet-Karte und BC9120

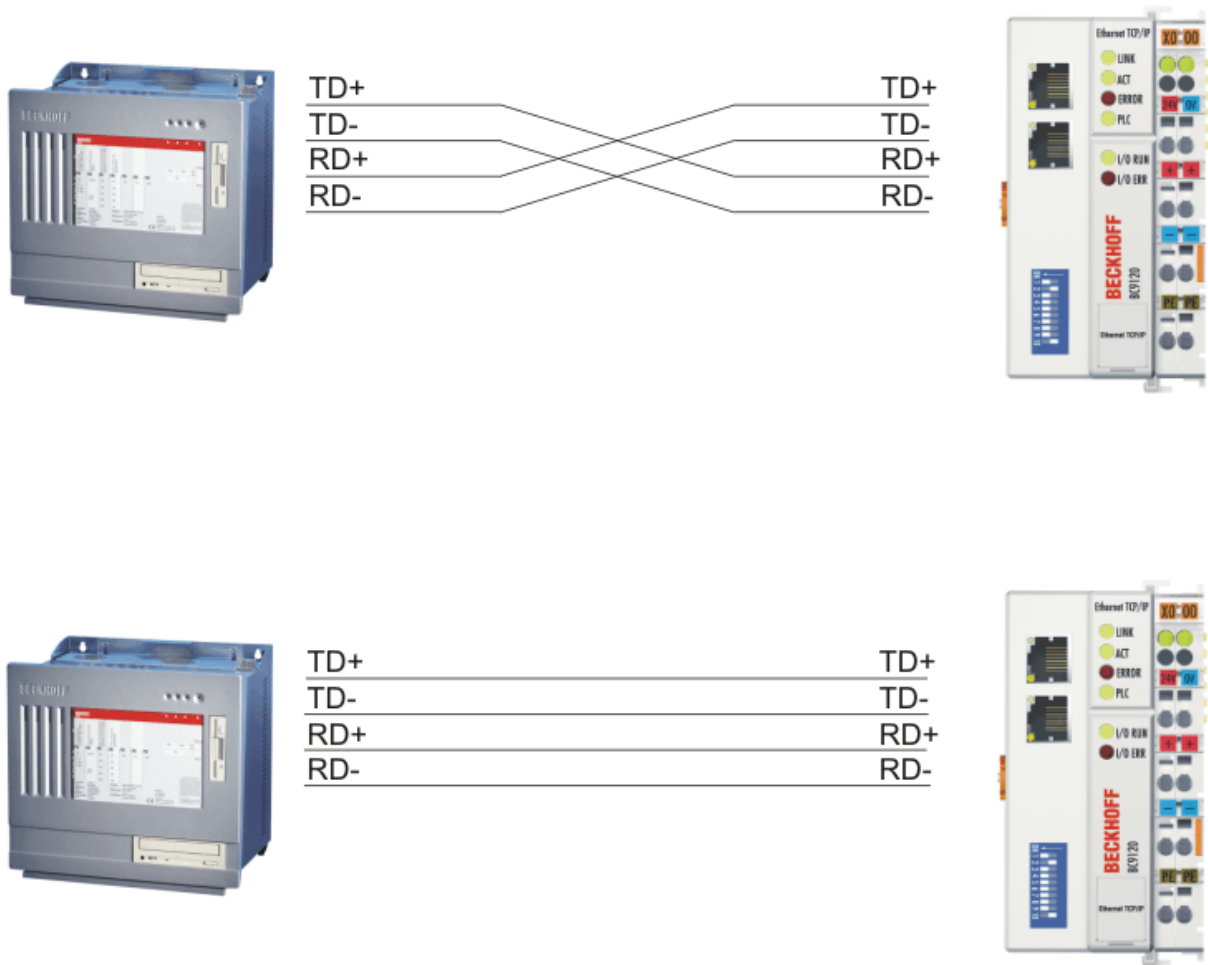


Abb. 20: Direkte Ethernet-Verbindung zwischen PC und BC9120

Um den PC direkt mit dem Busklemmen Controller zu verbinden, kann ein beliebiges Ethernet-Kabel genommen werden. Der interne Switch des BC91x0, BK9100 erkennt dies Automatisch.

Belegung des RJ45-Steckers

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

3.3.5 Ethernet-Kabel

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 dem 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 70 dB 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.

3.4 ATEX - Besondere Bedingungen (Standardtemperaturbereich)

⚠️ WARNUNG

Beachten Sie die besonderen Bedingungen für die bestimmungsgemäße Verwendung von Beckhoff-Feldbuskomponenten mit Standardtemperaturbereich in explosionsgefährdeten Bereichen (Richtlinie 2014/34/EU)!

- Die zertifizierten Komponenten sind in ein geeignetes Gehäuse zu errichten, das eine Schutzart von mindestens IP54 gemäß EN 60079-15 gewährleistet! Dabei sind die Umgebungsbedingungen bei der Verwendung zu berücksichtigen!
- Für Staub (nur die Feldbuskomponenten der Zertifikatsnummer KEMA 10ATEX0075 X Issue 9): Das Gerät ist in ein geeignetes Gehäuse einzubauen, das einen Schutzgrad von IP54 gemäß EN 60079-31 für Gruppe IIIA oder IIIB und IP6X für Gruppe IIIC bietet, wobei die Umgebungsbedingungen, unter denen das Gerät verwendet wird, zu berücksichtigen sind!
- Wenn die Temperaturen bei Nennbetrieb an den Einführungsstellen der Kabel, Leitungen oder Rohrleitungen höher als 70°C oder an den Aderverzweigungsstellen höher als 80°C ist, so müssen Kabel ausgewählt werden, deren Temperaturdaten den tatsächlich gemessenen Temperaturwerten entsprechen!
- Beachten für Beckhoff-Feldbuskomponenten mit Standardtemperaturbereich beim Einsatz in explosionsgefährdeten Bereichen den zulässigen Umgebungstemperaturbereich von 0 bis 55°C!
- Es müssen Maßnahmen zum Schutz gegen Überschreitung der Nennbetriebsspannung durch kurzzeitige Störspannungen um mehr als 40% getroffen werden!
- Die einzelnen Klemmen dürfen nur aus dem Busklemmensystem gezogen oder entfernt werden, wenn die Versorgungsspannung abgeschaltet wurde bzw. bei Sicherstellung einer nicht-explosionsfähigen Atmosphäre!
- Die Anschlüsse der zertifizierten Komponenten dürfen nur verbunden oder unterbrochen werden, wenn die Versorgungsspannung abgeschaltet wurde bzw. bei Sicherstellung einer nicht-explosionsfähigen Atmosphäre!
- Die Sicherung der Einspeiseklemmen KL92xx/EL92xx dürfen nur gewechselt werden, wenn die Versorgungsspannung abgeschaltet wurde bzw. bei Sicherstellung einer nicht-explosionsfähigen Atmosphäre!
- Adresswahlschalter und ID-Switche dürfen nur eingestellt werden, wenn die Versorgungsspannung abgeschaltet wurde bzw. bei Sicherstellung einer nicht-explosionsfähigen Atmosphäre!

Normen

Die grundlegenden Sicherheits- und Gesundheitsanforderungen werden durch Übereinstimmung mit den folgenden Normen erfüllt:

- EN 60079-0:2012+A11:2013
- EN 60079-15:2010
- EN 60079-31:2013 (nur für Zertifikatsnummer KEMA 10ATEX0075 X Issue 9)

Kennzeichnung

Die gemäß ATEX-Richtlinie für den explosionsgefährdeten Bereich zertifizierten Beckhoff-Feldbuskomponenten mit Standardtemperaturbereich tragen eine der folgenden Kennzeichnungen:



II 3G KEMA 10ATEX0075 X Ex nA IIC T4 Gc Ta: 0 ... +55°C
 II 3D KEMA 10ATEX0075 X Ex tc IIIC T135°C Dc Ta: 0 ... +55°C
 (nur für Feldbuskomponenten mit Zertifikatsnummer KEMA 10ATEX0075 X Issue 9)

oder



II 3G KEMA 10ATEX0075 X Ex nA nC IIC T4 Gc Ta: 0 ... +55°C
 II 3D KEMA 10ATEX0075 X Ex tc IIIC T135°C Dc Ta: 0 ... +55°C
 (nur für Feldbuskomponenten mit Zertifikatsnummer KEMA 10ATEX0075 X Issue 9)

3.5 ATEX - Besondere Bedingungen (erweiterter Temperaturbereich)

⚠️ WARNUNG

Beachten Sie die besonderen Bedingungen für die bestimmungsgemäße Verwendung von Beckhoff-Feldbuskomponenten mit erweitertem Temperaturbereich (ET) in explosionsgefährdeten Bereichen (Richtlinie 2014/34/EU)!

- Die zertifizierten Komponenten sind in ein geeignetes Gehäuse zu errichten, das eine Schutzart von mindestens IP54 gemäß EN 60079-15 gewährleistet! Dabei sind die Umgebungsbedingungen bei der Verwendung zu berücksichtigen!
- Für Staub (nur die Feldbuskomponenten der Zertifikatsnummer KEMA 10ATEX0075 X Issue 9): Das Gerät ist in ein geeignetes Gehäuse einzubauen, das eine Schutzart von IP54 gemäß EN 60079-31 für Gruppe IIIA oder IIIB und IP6X für Gruppe IIIC bietet, wobei die Umgebungsbedingungen, unter denen das Gerät verwendet wird, zu berücksichtigen sind!
- Wenn die Temperaturen bei Nennbetrieb an den Einführungsstellen der Kabel, Leitungen oder Rohrleitungen höher als 70°C oder an den Aderverzweigungsstellen höher als 80°C ist, so müssen Kabel ausgewählt werden, deren Temperaturdaten den tatsächlich gemessenen Temperaturwerten entsprechen!
- Beachten Sie für Beckhoff-Feldbuskomponenten mit erweitertem Temperaturbereich (ET) beim Einsatz in explosionsgefährdeten Bereichen den zulässigen Umgebungstemperaturbereich von -25 bis 60°C!
- Es müssen Maßnahmen zum Schutz gegen Überschreitung der Nennbetriebsspannung durch kurzzeitige Störspannungen um mehr als 40% getroffen werden!
- Die einzelnen Klemmen dürfen nur aus dem Busklemmensystem gezogen oder entfernt werden, wenn die Versorgungsspannung abgeschaltet wurde bzw. bei Sicherstellung einer nicht-explosionsfähigen Atmosphäre!
- Die Anschlüsse der zertifizierten Komponenten dürfen nur verbunden oder unterbrochen werden, wenn die Versorgungsspannung abgeschaltet wurde bzw. bei Sicherstellung einer nicht-explosionsfähigen Atmosphäre!
- Die Sicherung der Einspeiseklemmen KL92xx/EL92xx dürfen nur gewechselt werden, wenn die Versorgungsspannung abgeschaltet wurde bzw. bei Sicherstellung einer nicht-explosionsfähigen Atmosphäre!
- Adresswahlschalter und ID-Switche dürfen nur eingestellt werden, wenn die Versorgungsspannung abgeschaltet wurde bzw. bei Sicherstellung einer nicht-explosionsfähigen Atmosphäre!

Normen

Die grundlegenden Sicherheits- und Gesundheitsanforderungen werden durch Übereinstimmung mit den folgenden Normen erfüllt:

- EN 60079-0:2012+A11:2013
- EN 60079-15:2010
- EN 60079-31:2013 (nur für Zertifikatsnummer KEMA 10ATEX0075 X Issue 9)

Kennzeichnung

Die gemäß ATEX-Richtlinie für den explosionsgefährdeten Bereich zertifizierten Beckhoff-Feldbuskomponenten mit erweitertem Temperaturbereich (ET) tragen die folgende Kennzeichnung:



II 3G KEMA 10ATEX0075 X Ex nA IIC T4 Gc Ta: -25 ... +60°C
 II 3D KEMA 10ATEX0075 X Ex tc IIIC T135°C Dc Ta: -25 ... +60°C
 (nur für Feldbuskomponenten mit Zertifikatsnummer KEMA 10ATEX0075 X Issue 9)

oder



II 3G KEMA 10ATEX0075 X Ex nA nC IIC T4 Gc Ta: -25 ... +60°C
 II 3D KEMA 10ATEX0075 X Ex tc IIIC T135°C Dc Ta: -25 ... +60°C
 (nur für Feldbuskomponenten mit Zertifikatsnummer KEMA 10ATEX0075 X Issue 9)

3.6 Weiterführende Dokumentation zu ATEX und IECEx

HINWEIS



Weiterführende Dokumentation zum Explosionsschutz gemäß ATEX und IECEx

Beachten Sie auch die weiterführende Dokumentation

Explosionsschutz für Klemmensysteme

Hinweise zum Einsatz der Beckhoff Klemmensysteme in explosionsgefährdeten Bereichen gemäß ATEX und IECEx,

die Ihnen auf der Beckhoff-Homepage www.beckhoff.de im Download-Bereich Ihres Produktes zum Download zur Verfügung steht!

4 Parametrierung und Inbetriebnahme

4.1 Anlaufverhalten des Busklemmen-Controllers

Nach dem Einschalten prüft der Busklemmen-Controller seinen Zustand, konfiguriert den K-Bus, erstellt anhand der gesteckten Busklemmen eine Aufbauliste und startet seine lokale SPS.

Beim Hochlauf des Busklemmen-Controllers leuchten und blinken die I/O-LEDs. Im fehlerfreien Zustand sollte nach ca. 3 bis 5 Sekunden keine I/O-LED mehr blinken. Im Fehlerfall hängt es von der Fehlerart ab, welche LED blinkt (siehe Diagnose-LEDs).

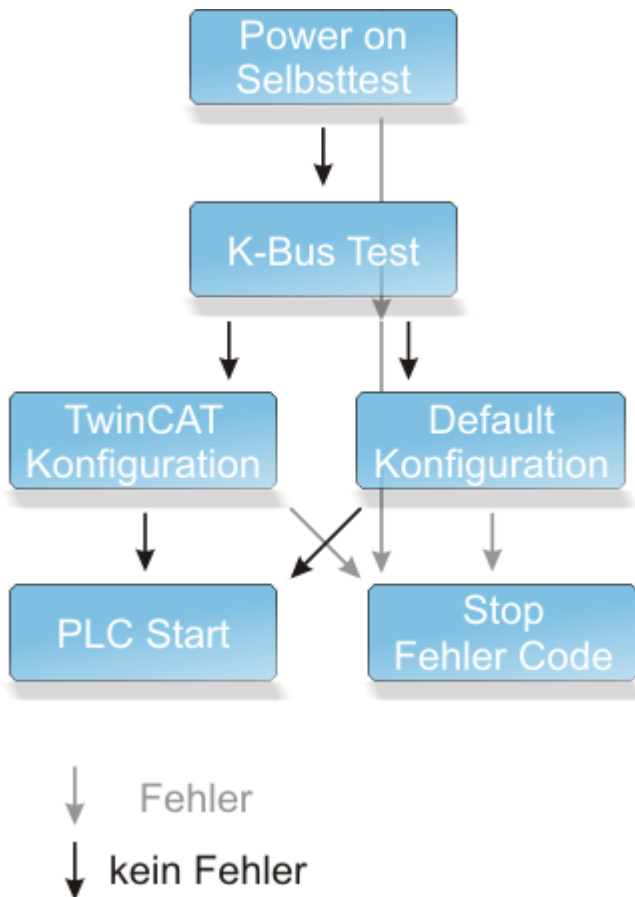


Abb. 21: Anlaufverhalten des BC9xx0

Der Busklemmen-Controller zeigt eine fehlerhafte TwinCAT-Konfiguration an (LEDs/Display). Bei den Busklemmen-Controllern der Serien BCxx50, BCxx20 und BXxxxx kann die Anzeige auch über den System Manager ausgelesen werden.

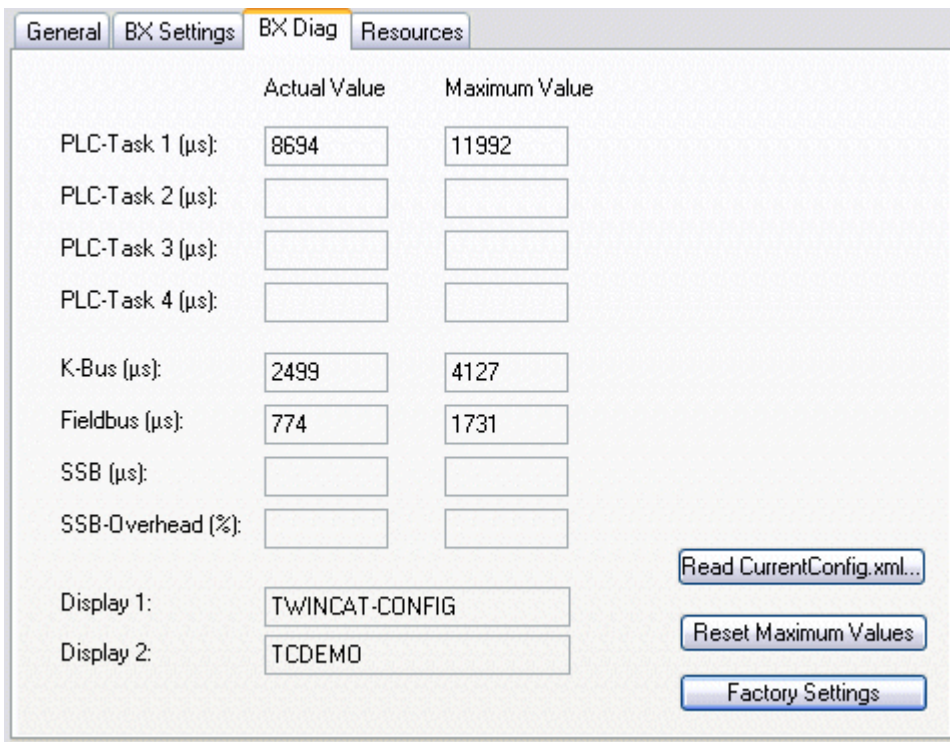


Abb. 22: Anzeige im System-Manager

4.2 DIP-Schalter

Auf den BC9050 und BC9120/BC9020 befindet sich DIP-Schalter. Diese DIP-Schalter haben zwei Funktionen.

1. Adressierungsmodus oder Einstellen der IP-Adresse
2. Konfigurationsmodus

Modus eins wird aktiviert, wenn am K-Bus Busklemmen angesteckt sind und der Busklemmen-Controller eingeschaltet wird. Modus zwei wird aktiviert, wenn nur die Endklemme beim Einschalten des Busklemmen-Controllers gesteckt ist.

Modus 1: Adressierungsmodus

DIP-Schalter des BC9050

Schalter rechte Position: ON, Schalter linke Position: OFF.

- Die DIP-Schalter 1 bis 8 editieren das letzte Byte der IP-Adresse (hier im Bild die roten DIP-Schalter).
- Die DIP-Schalter 1 bis 2 wählen den IP-Adress-Mode (hier im Bild die blauen DIP-Schalter).

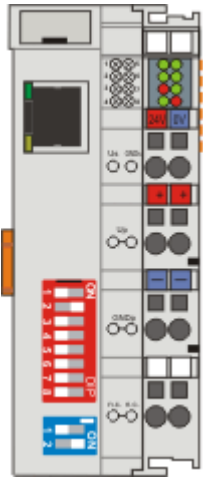


Abb. 23: DIP-Schalter des BC9050

DIP1	DIP2	Bedeutung
off	off	172.16.21.xxx (xxx Entspricht DIP Schalter 1-8), Sub Net Mask 255.255.0.0, Gefault Gateway 0.0.0.0
on	off	BootP (DIP 1-8 alle off), BootP & Safe (DIP 1-8 alle on)
off	on	DHCP
on	on	Konfiguration über den TwinCAT System Manager

DIP-Schalter des BC9020 und des BC9120

Schalter rechte Position: OFF, Schalter linke Position: ON.

- Die DIP-Schalter 1 bis 8 editieren das letzte Byte der IP-Adresse.
- Die DIP-Schalter 9 bis 10 wählen den IP-Adress-Mode.

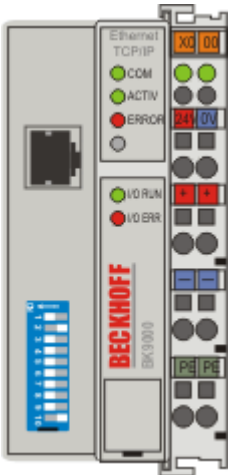


Abb. 24: DIP-Schalter des BC9020 und des BC9120

DIP9	DIP10	Bedeutung
off	off	BC9020 172.16.22.xxx (xxx Entspricht DIP Schalter 1-8), Sub Net Mask 255.255.0.0, Gefault Gateway 0.0.0.0 BC9120 172.16.23.xxx (xxx Entspricht DIP Schalter 1-8), Sub Net Mask 255.255.0.0, Gefault Gateway 0.0.0.0
on	off	BootP (DIP 1-8 alle off), BootP & Safe (DIP 1-8 alle on)
off	on	DHCP
on	on	Konfiguration über den TwinCAT System Manager

Modus 2: Konfigurationsmodus

Stecken Sie an den spannungslosen Busklemmen-Controller nur die Endklemme KL9010. Stellen Sie den DIP-Schalter auf die gewünschte Funktion und schalten dann den Busklemmen-Controller ein. Sobald der Controller hoch gelaufen ist, ist die Funktion aktiviert worden. Sie können dann wie gewohnt fortfahren. Beim BC9020 und BC9120 müssen DIP 9 und 10 off sein, sowie beim BC9050 die DIP-Schalter 1 und 2 beim 2 poligen DIP-Schalter ebenso auf off stellen.

Einstellung		Bedeutung
BC9050	BC9020 BC9120	
99	99	Hersteller Einstellungen
98	98	Bootprojekt löschen
97	97	TwinCAT Config löschen

99 DIP Schalter

DIP 1 ON	DIP 2 ON	DIP 3 OFF	DIP 4 OFF	DIP 5 OFF	DIP 6 ON	DIP 7 ON	DIP 8 OFF
-------------	-------------	--------------	--------------	--------------	-------------	-------------	--------------

98 DIP Schalter

DIP 1 OFF	DIP 2 ON	DIP 3 OFF	DIP 4 OFF	DIP 5 OFF	DIP 6 ON	DIP 7 ON	DIP 8 OFF
--------------	-------------	--------------	--------------	--------------	-------------	-------------	--------------

97 DIP Schalter

DIP 1 ON	DIP 2 OFF	DIP 3 OFF	DIP 4 OFF	DIP 5 OFF	DIP 6 ON	DIP 7 ON	DIP 8 OFF
-------------	--------------	--------------	--------------	--------------	-------------	-------------	--------------

4.3 Einstellung der IP-Adresse

Die Einstellung der IP-Adresse kann mit vier unterschiedlichen Verfahren durchgeführt werden, die im Folgenden genauer beschrieben werden.

BX9000: Das Adressierungsverfahren wird per Navigationsschalter und Display oder über die TwinCAT Config vorgegeben.

Verfahren	Erläuterung	Benötigte Komponenten
Manuell	Adressierung über den Navigationsschalter (nur BX9000)	keine
TwinCAT	<u>Adressierung über TwinCAT System Manager</u> [▶ 42]	PC mit Netzwerk und TwinCAT
BootP*	<u>Adressierung über BootP-Server</u> [▶ 42]	BootP-Server
DHCP*	<u>Adressierung über DHCP-Server</u> [▶ 44]	DHCP-Server
Lokale IP-Adresse*	Lokale IP-Adresse	Falls kein BootP- oder DHCP-Server antwortet oder vorhanden ist

*) BX9000 ab Firmware-Version 1.20

4.3.1 Adresseinstellung über den TwinCAT System Manager

Voraussetzungen

- ab BX-Firmware 1.08 (siehe Anzeige auf dem Display des BX9000 nach Einschalten der Betriebsspannung).
- alle BC9050, BC9020 und BC9120
- ab TwinCAT 2.10, Build 1322

Für die Einstellung der IP-Adresse über den System Manager ist eine funktionsfähige ADS-Verbindung notwendig. Diese kann Seriell oder über Ethernet erfolgen (siehe Kapitel Busklemmen-Controller mit dem TwinCAT System Manager suchen).

Bei den BC9x20 müssen die DIP-Schalter 9 und 10 auf ON stehen, beim BC9050 müssen die Schalter 1 und 2 des 2-poligen DIP-Schalter auf ON stehen, um die IP-Adressierung über den System Manager zuzulassen.

Dann sind die Einstellungen, die über den System Manager gemacht wurden, gültig.

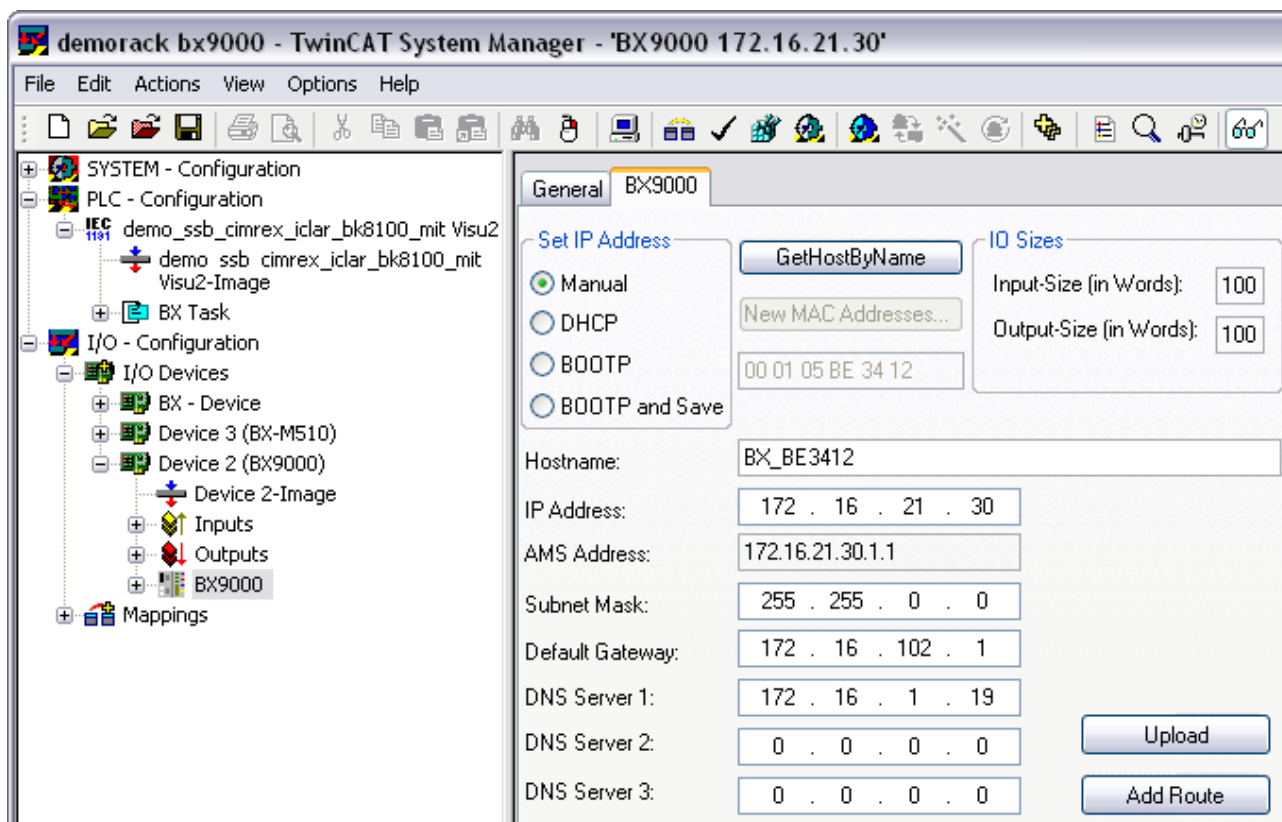



Abb. 25: Adresseinstellung über den TwinCAT System Manager

Mit Upload können Sie jetzt die aktuellen Einstellungen auslesen. Sollten Sie den Busklemmen-Controller offline Konfiguriert haben können Sie durch mit der Schaltfläche *Add Route* eine Verbindung zum Busklemmen-Controller aufbauen. Editieren Sie Ihre gewünschten Einstellungen, aktivieren Sie die

Konfiguration  und starten Sie Ihren Busklemmen-Controller mit dem das grünen TwinCAT Icon (Tastenkombination [Ctrl] [F4]) neu. Sollten Sie dem Busklemmen-Controller eine neue IP-Adresse gegeben haben so müssen Sie erneut die neue Route eintragen (siehe Kapitel Busklemmen-Controller mit dem TwinCAT System Manager suchen).

4.3.2 Adresseinstellung über BootP-Server

BX9000: Die Vergabe der IP-Adresse über einen BootP-Server wird beim BX9000 über den Navigationsschalter des Controllers aktiviert (siehe Kapitel BX-Menü, oder First Steps).

BC9xx0: Die Vergabe der IP-Adresse über einen BootP-Server wird beim BC9xx0 über den DIP-Schalter aktiviert.

Speicherverhalten der IP-Adresse

BootP & Save

Bei *BootP & Save* wird die vom BootP-Server vergebene IP-Adresse auf dem BX-Controllergespeichert und der BootP-Service wird beim nächsten Kaltstart nicht erneut angefragt. Die Adresse kann durch Reaktivierung der Herstellereinstellungen (mittels Konfigurationssoftware KS2000 oder DIP-Schalter und Endklemme) wieder gelöscht werden.

BootP

Bei BootP ist die vom BootP-Server vergebene IP-Adresse nur bis zum Ausschalten des Busklemmen-Controller gültig. Beim nächsten Neustart muss der BootP-Server dem Busklemmen-Controller eine neue IP-Adresse vergeben. Bei einem Software-Reset des Busklemmen-Controller bleibt die Adresse allerdings erhalten.

Beckhoff BootP-Server

Beckhoff bietet einen BootP-Server für Windows 98, ME, NT4.0, NT2000 und XP an. Sie finden die Installationsversion auf der Beckhoff CD *Software Products* im Ordner *Unsupported Utilities* oder im Internet unter <https://download.beckhoff.com/>

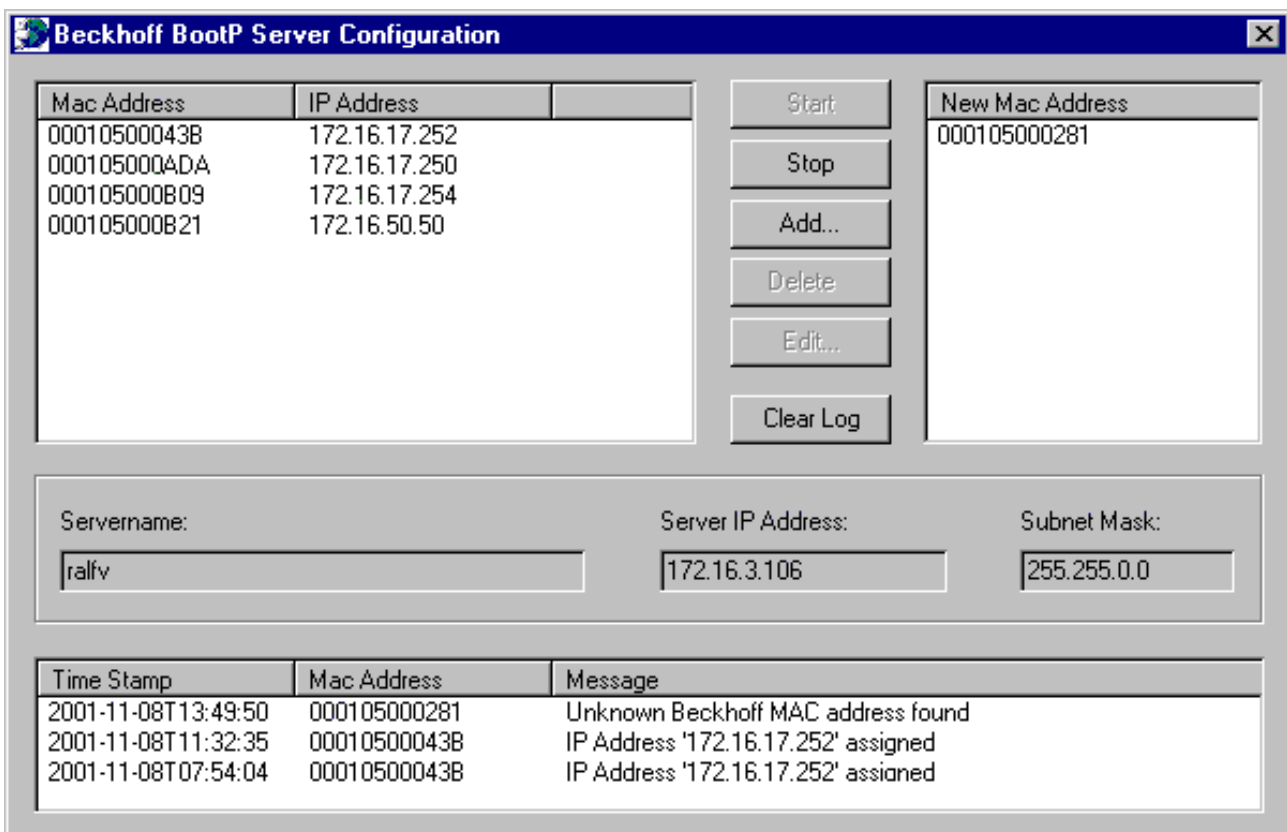


Abb. 26: Adresseinstellung über BootP-Server

Sobald der BootP-Server gestartet wird zeigt das Fenster *New MAC Address* alle Beckhoff-Knoten an, die im BootP-Betrieb arbeiten und noch keine IP-Adresse bekommen haben. Die Zuweisung der MAC-ID zur IP-Adresse erfolgt mit der Schaltfläche [**<<**]. Eine erfolgreiche Zuweisung wird im Log-Fenster angezeigt. Um den BootP-Server beim Booten Ihres PCs automatisch zu starten reicht eine Verknüpfung im Windows Autostart-Ordner. Fügen Sie hierzu im Verknüpfungspfad den Parameter */Start* an (.../TcBootPDlg.exe/start).

4.3.3 Adresseinstellung über DHCP-Server

Die Vergabe der IP-Adresse über einen DHCP-Server wird beim BX9000 über den Navigationsschalter des Busklemmen-Controllers aktiviert (siehe dazu BX-Menü, oder First Steps).

Die Vergabe der IP-Adresse über einen DHCP-Server wird beim BC9xx0 über den Dipschalter des Busklemmen-Controllers aktiviert (siehe Kapitel *DIP-Schalter*).

Ist DHCP eingeschaltet bekommt der Busklemmen-Controller automatisch eine IP-Nummer vom DHCP-Server zugewiesen. Der DHCP-Server muss hierfür die MAC-ID des Busklemmen-Controller kennen. Die IP-Adresse sollte statisch eingestellt werden.

Der DNS-Name wird aus dem Typ und den letzten 3 Byte der MAC-ID gebildet. Die MAC-ID steht auf dem Produktionsaufkleber des Busklemmen Controllers.

Beispiel zum BX9000

- MAC ID: 00-01-05-01-02-03
- DNS-Name: BX_010203

Beispiel zum BC9050

- MAC ID: 00-01-05-03-02-01
- DNS-Name: BC_030201

4.3.4 Auto-IP-Adresse

Die Auto IP-Adresse wird aktiviert wenn ein DHCP oder BootP Server nicht gefunden wird. Das kann bis zu mehreren Minuten dauern. Die Auto IP-Adresse wird wie folgt gebildet:

Im Falle eines Timeouts wird die IP-Adresse 169.254.[MAC_05].[MAC06] erzeugt. Sollte MAC_05 0 sein wird sie auf 1 gesetzt.

Es werden dann 3 ARP Probes gesendet. Kommt keine Antwort wird ein gratuitous ARP gesendet und die IP wird übernommen.

Ist die IP-Adresse bereits vorhanden wird auf [MAC_05] und [MAC06] [MAC_04] aufaddiert und es wird noch mal versucht.

4.3.5 Subnetz-Maske

Die Subnetz-Maske unterliegt der Kontrolle des Netzwerkverwalters und legt die Struktur der Subnetze fest.

Kleine Netze ohne Router benötigen keine Subnetz-Maske. Das gleiche gilt, wenn Sie keine registrierten IP-Nummern verwenden. Sie können die Subnetz-Maske dazu verwenden, anstelle des Gebrauchs vieler Netznummern das Netz mit dieser Maske zu unterteilen.

Die Subnetz-Maske ist eine 32-Bit Ziffer:

- Einsen in der Maske kennzeichnen den Subnetz-Nummernteil eines Adressraums.
- Nullen kennzeichnen den Teil des Adressraums, der für die Host-IDs zur Verfügung steht.

Beschreibung	Binäre Darstellung	Dezimale Darstellung
IP-Adresse	10101100.00010000.00010001.11001000	172.16.17.200
Subnetz-Maske	11111111.11111111.00010100.00000000	255.255.20.0
Netz-ID	10101100.00010000.00010000.00000000	172.16.16.0
Host-ID	00000000.00000000.00000001.11001000	0.0.1.200

Standard Subnetz-Maske

Adressklasse	Standard Subnetz-Maske (dezimal)	Standard Subnetz-Maske (hex)
A	255.0.0.0	FF.00.00.00
B	255.255.0.0	FF.FF.00.00
C	255.255.255.0	FF.FF.FF.00

● Beachten Sie zu Subnetzen und zur Hostnummer

i Die Subnetze 0 und das nur aus Einsen bestehende Subnetz dürfen nicht verwendet werden!
 Die Host-Nummer 0 und die aus nur Einsen bestehende Host-Nummer dürfen nicht verwendet werden!
 Wenn die IP-Adresse über die Konfigurationssoftware KS2000 eingestellt wurde, muss auch die Subnetz-Maske mit der Konfigurationssoftware KS2000 geändert werden.
 Bei ARP-Adressierung wird anhand der IP-Adresse die dazugehörige Standard Subnetz-Maske eingetragen.
 Bei BootP und DHCP wird die Subnetz-Maske mit vom Server übertragen.

4.4 Konfiguration

4.4.1 Überblick

Konfigurationsarten

Bei den Busklemmen-Controllern der Serien BCxx50, BCxx20 und BXxx00 gibt es zwei verschiedene Arten der Konfiguration, die DEFAULT CONFIG und die TWINCAT CONFIG.

DEFAULT-CONFIG

Die Busklemmen mappen sich in der Reihenfolge wie diese gesteckt sind, erst die komplexen Busklemmen, dann die digitalen Busklemmen.

Das Mapping der komplexen Busklemmen ist:

- Word-Alignment
- komplexe Darstellung

⚠ VORSICHT

Prozessabbild ist abhängig von angesteckten Klemmen!

Das Prozessabbild verändert sich, sobald eine Klemme dazu gesteckt wird oder entfernt wird!

Die Daten der Feldbus-Slave Schnittstelle werden SPS-Variablen genannt. Die SPS-Variablen befinden sich ab der Adresse %QB1000 und %IB1000.

Weiterhin kann die DEFAULT-CONFIG ohne SPS-Programm für das Schreiben und Testen der Angeschlossenen Busklemmen verwendet werden. Dafür muss im System Manager der Busklemmen-Controller gescannt werden und die Betriebsart FreeRun aktiviert werden (um diese Funktion zu nutzen darf kein SPS-Programm auf dem Busklemmen-Controller sein).

TWINCAT-CONFIG

In der TWINCAT-CONFIG können die Busklemmen und die SPS-Variablen frei verknüpft sein (TwinCAT System Manager-File notwendig). Die Konfiguration wird mit Hilfe des System Managers per ADS zum Koppler übertragen.

Für die TwinCAT Config (TC-File) benötigen Sie folgendes:

- Über der Feldbus (PROFIBUS, CANopen, Ethernet)
 PROFIBUS: (BC3150, BX3100)

- PC mit FC310x ab Version 2.0 und TwinCAT 2.9 Build 1000
- BX3100 mit CIF60 oder CP5412
- TwinCAT 2.9 Build 946
(**ACHTUNG:** bei den PROFIBUS Karten von Hilscher ist nur eine ADS-Kommunikation erlaubt, d.h. entweder System Manager oder PLC Control)
CANopen: (BC5150, BX5100)
- PC mit FC510x ab Version 1.76 TwinCAT Build 1030
DeviceNet: (BC5250, BX5200)
- Auf Anfrage
Ethernet: (BC9050, BC9020, BC9120, BX9000)
- PC mit TwinCAT 2.10 Build 1322
- Über das serielle ADS TwinCAT 2.9 Build 1010
 - BX3100 Version 1.00
 - BX5100 Version 1.00
 - BX5200 Version 1.10
 - BX8000 Version 1.00
 - BC3150, BC5150, BC5250, BC9050, BC9020, BC9120 ab Firmware B0
 - Für BC8150 ab TwinCAT 2.10 Build 1243

BCxx50 und BXxx00 können über den System Manager des TwinCAT Programms parametrierbar werden.

- Variables I/O Mapping
- Typ gerechte PROFIBUS Daten (nur BC3150 und BX3100)
- RTC (Real Time Clock) (nur BX-Serie)
- SSB Bus (Smart System Bus) (nur BX-Serie)
- PLC Einstellungen
- K-Bus Einstellungen

Die Konfiguration kann per Feldbus ADS-Protokoll oder seriellem ADS-Protokoll zum BCxx50 oder BXxx00 übertragen werden.

Mit der TwinCAT Konfiguration kann man Variablen, I/Os und Daten verknüpfen. Folgendes ist möglich:

- PLC - K-BUS
- PLC - Feldbus (z. B. PROFIBUS Slave Schnittstelle zur PLC)
- K-Bus - Feldbus (nur bei den BX-Controllern)
- Unterstützung der TwinSAFE-Klemmen (nur BX-Controller ab Firmware 1.17)

Zusätzlich können mit der TwinCAT Konfiguration spezielle Verhalten parametrierbar werden, zum Beispiel ob bei einem Feldbus Fehler die Daten erhalten bleiben oder auf "0" gesetzt werden sollen.

Die Echtzeituhr kann man über einen Karteireiter im System Manager einstellen.

Arbeitsschritte

1. Feldbus Adresse einstellen
2. System Manager öffnen und TC-File anlegen
3. Feldbus Daten in dem TC-File konfigurieren
4. TC-File speichern
5. Neuer System Manager öffnen und PC-File anlegen und gespeichertes TC-File einlesen
6. Verknüpfen zu einer SPS-Task herstellen
7. Speichern der Konfiguration
8. Starten des TwinCAT Systems
9. System Manager des TC-File öffnen, fertig konfigurieren und zum BCxx50, BCxx20 oder BXxx00 übertragen
10. Programm zum BCxx50, BCxx20 oder BXxx00 übertragen

11. Bootprojekt erzeugen

4.4.2 Busklemmen-Controller mit dem TwinCAT System Manager suchen

Vorraussetzung BX9000:

- ab BX-Firmware 1.08 (siehe Anzeige auf dem Display des BX9000 nach Einschalten der Betriebsspannung)
- ab TwinCAT 2.10 Build 1251

Vorraussetzung BC9050, BC9020, BC9120:

- ab Firmware B1
- ab TwinCAT 2.10 Build 1322

Zur Einstellung der IP-Adresse über den System Manager ist eine funktionsfähige ADS-Verbindung notwendig. Diese kann Seriell oder über Ethernet erfolgen. Beim BC9191 und BC9191-0100 nur über Ethernet.

Für die ADS-Verbindung über Ethernet ist eine funktionierende Ethernet Verbindung notwendig. Sie können die IP-Verbindung mit dem Befehl PING testen. Per Default ist der Busklemmen-Controller auf 172.16.21.20 eingestellt mit der Sub-Netzmaske 255.255.0.0. Stellen Sie Ihren PC auf die gleiche Netzwerkkategorie ein, zum Beispiel 172.16.200.100 (Sub Net Mask 255.255.0.0) ein.

Prüfen Sie nun mit PING ob eine Verbindung existiert:

Starten Sie nun den TwinCAT System Manager und suchen Sie den Busklemmen-Controller mit Hilfe des im Bild markierten Buttons oder drücken Sie F8:

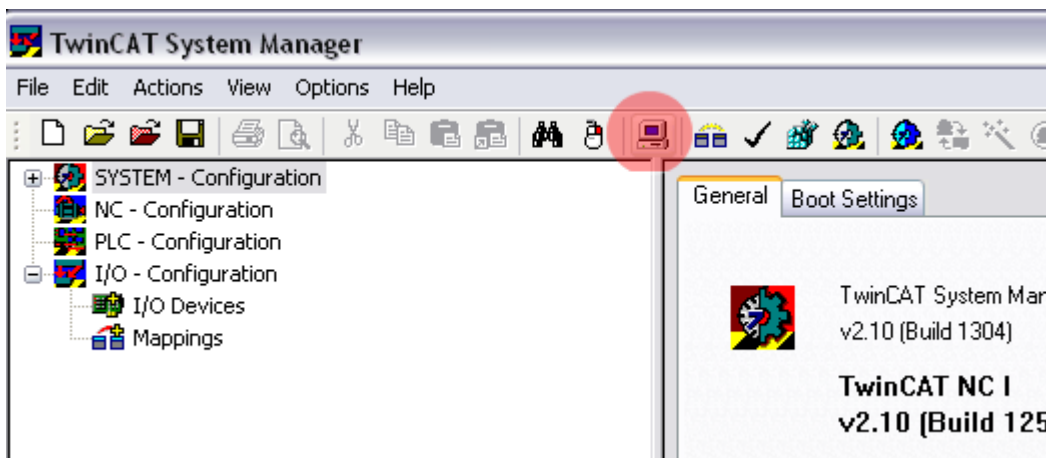


Abb. 27: Suchen des Busklemmen-Controllers

Nun suchen Sie über Ethernet den Busklemmen-Controller:

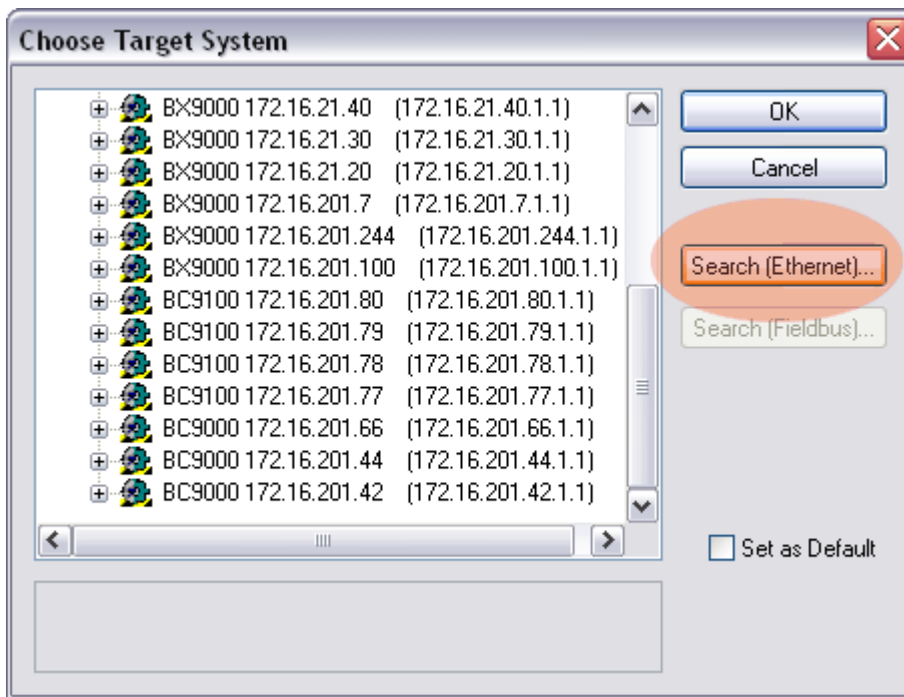


Abb. 28: Auswahl des Zielsystems

Suchen Sie nun über Broadcast Search Ihren Busklemmen-Controller. Haben Sie mehrere Busklemmen-Controller in Ihrem Netzwerk, können Sie diese mit Hilfe des Namens unterscheiden. Der Name stellt sich aus "BX_" oder "BC_" und den letzten drei Bytes der MAC-ID zusammen. Die MAC-ID finden Sie beim BX9000 auf der Seite und bei den Busklemmen-Controllern der BC-Serie auf der Unterseite des Gehäuses.

Beispiel: MAC-ID: 00-01-05-00-1D-C3, dann ist der Default-Name BX_001DC3.

Sollte über den Broadcast Search kein Gerät gefunden werden, überprüfen Sie die Ethernet-Verbindung oder die Firmware des Controllers.

Ist der Busklemmen-Controller über DHCP adressiert worden, können Sie den Busklemmen-Controller über die Schaltfläche *Add Route* im Dialog *Host Name* in ihre Verbindung aufnehmen.

Ist der Busklemmen-Controller manuell oder über BootP Adressiert worden so wählen sie vergebene *IP-Adresse* an und über die Schaltfläche *Add Route* nehmen Sie die Verbindung auf.

Quittieren bei Abfrage eines Passwortes diesen Dialog einfach ohne Eingabe. Beim Busklemmen-Controller ist kein Passwort notwendig; es wird auch nicht von den Busklemmen-Controllern unterstützt.

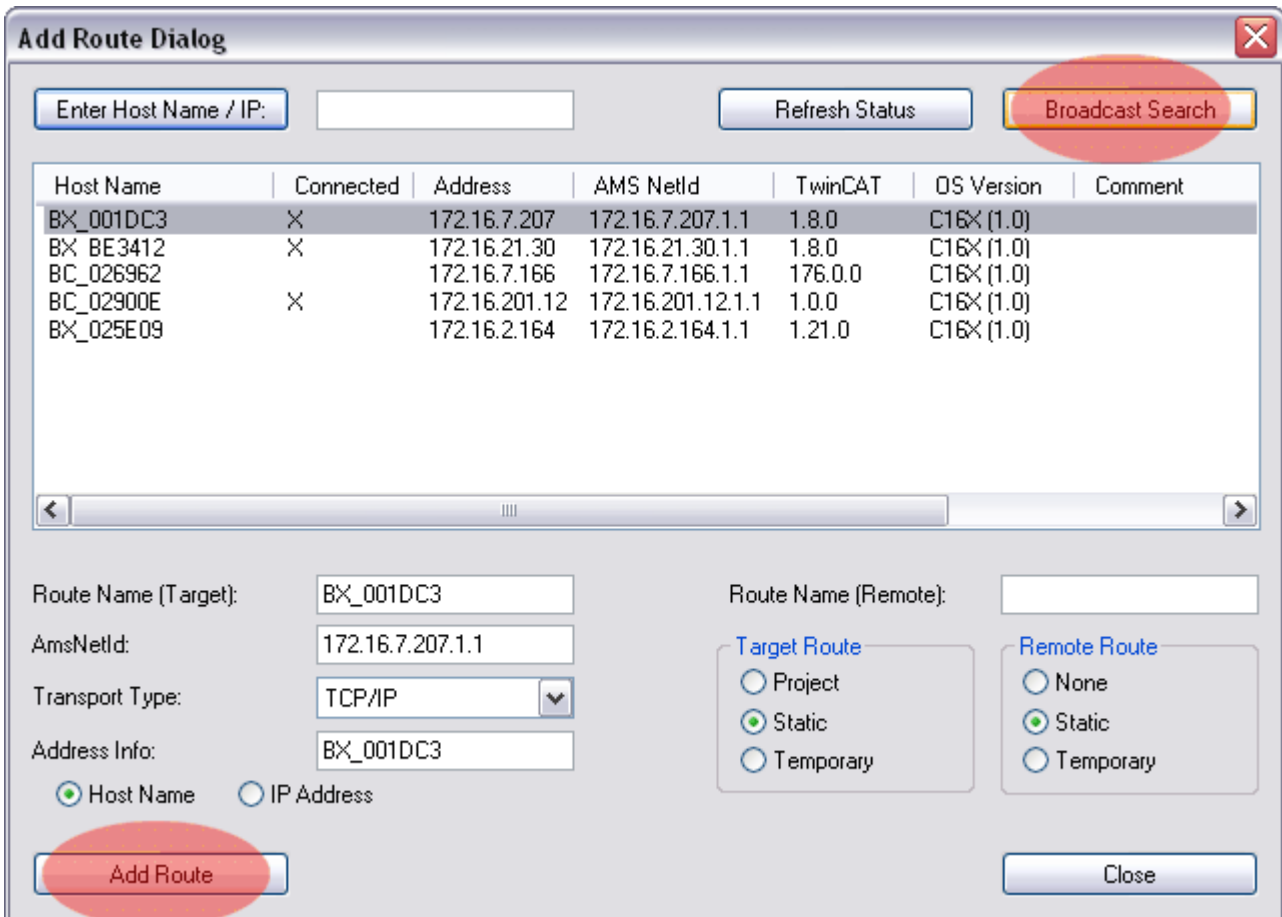


Abb. 29: Add Route Dialog

Wählen Sie nun den Busklemmen-Controller aus, mit dem Sie sich verbinden wollen und Scannen Sie die daran angeschlossenen Geräte (SSB, K-Bus). Der Busklemmen-Controller muss sich dazu im Config Mode befinden (Shift-F4).

4.4.3 Anlegen einer TwinCAT-Konfiguration

Um einen Busklemmen-Controller der Serien BCxx50, BCxx20, BXxx00 oder BC9191 zu konfigurieren muss im System-Manager ein BX-File angelegt werden. Zur Vereinfachung sind die Grundgeräte schon als File vorbereitet. Dazu öffnen Sie mit *New from Template* den entsprechenden Busklemmen-Controller.

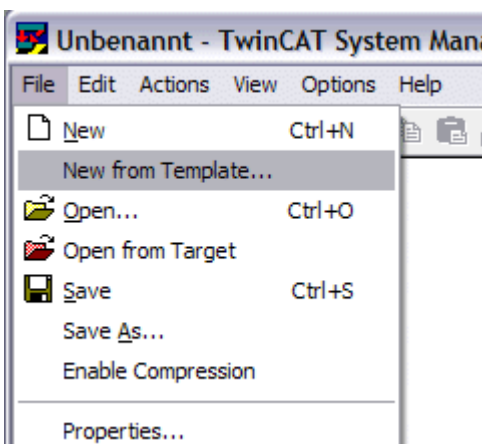


Abb. 30: Anlegen einer TwinCAT-Konfiguration

Wählen Sie den entsprechenden Busklemmen-Controller aus.

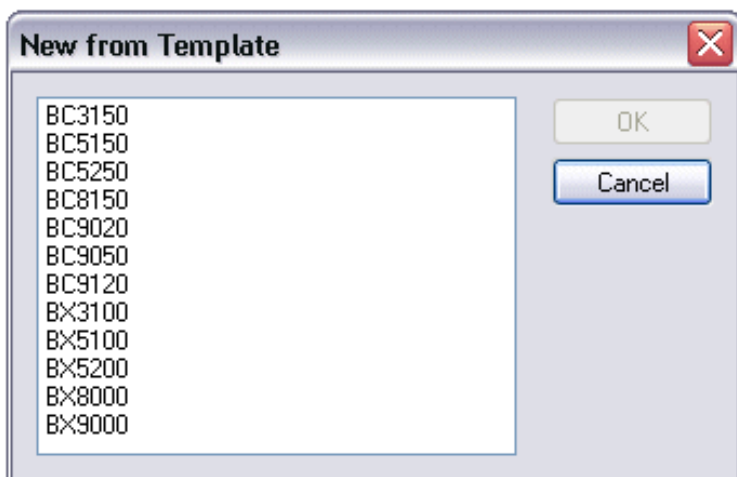


Abb. 31: Auswahl des Busklemmen-Controllers

Nun sind alle Komponenten des Busklemmen-Controllers vorhanden:

- Feldbusschnittstelle
- [K-Bus Interface \[► 60\]](#)
- [PLC Programm \[► 62\]](#)
- SSB (nur Busklemmen-Controller der BX-Serie)

Die Konfiguration der Geräte entnehmen Sie den entsprechenden Kapiteln.

4.4.4 Download einer TwinCAT-Konfiguration

Die TwinCAT-Konfiguration wird per ADS-Protokoll zum Busklemmen-Controller geladen.

Serielles ADS-Protokoll

(alle Busklemmen-Controller der Serien BXxx00 und BCxx50)

Tragen Sie die serielle ADS-Verbindung ein, wie unter dem Kapitel [Serielle ADS \[► 56\]](#) beschrieben ist.

ADS-Protokoll über den Feldbus

(nur BC3150, BC5150, BC9x20, BC9050, BX3100, BX5100, BX9000, BC9191)

Voraussetzung hierfür ist, dass TwinCAT als Master arbeitet und sich im Datenaustausch befindet, d.h. die physikalische, wie auch die Feldbus-Konfiguration muss abgeschlossen sein und der Datenaustausch vom Master (z. B. Feldbus-Master-Karte) zum Busklemmen-Controller stattfinden.

Auswahl des Zielsystems

Wählen Sie den Busklemmen-Controller aus, auf den Sie die Konfiguration laden wollen. Mit der Funktionstaste F8 öffnet sich der Dialog, mit dem Sie Ihr File auf das entsprechende Gerät herunterladen können.

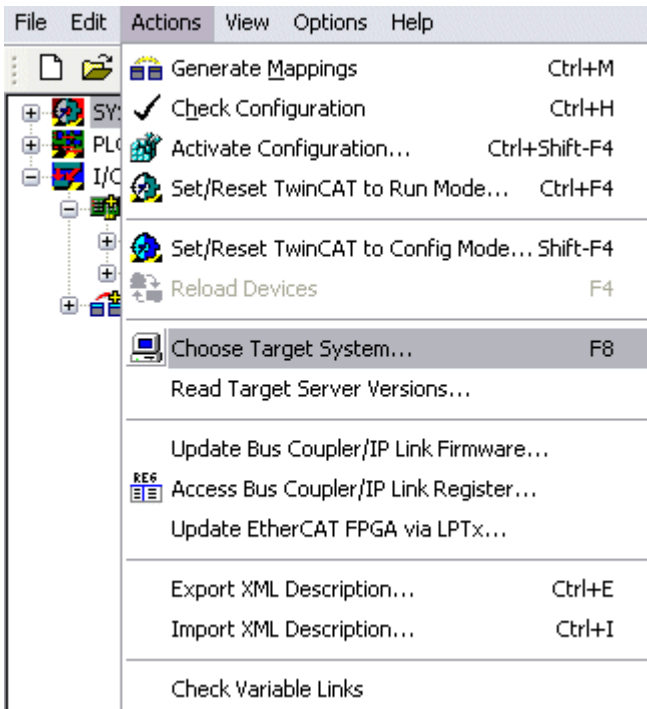


Abb. 32: Download einer TwinCAT-Konfiguration

Wählen Sie den entsprechenden Busklemmen-Controller aus.

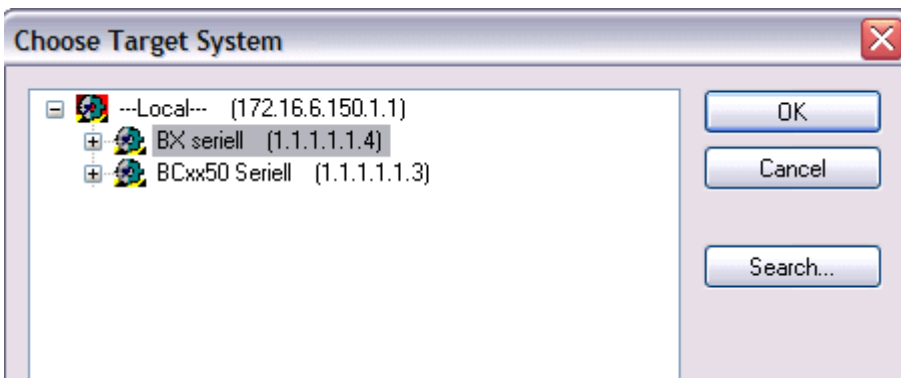


Abb. 33: Auswahl des Busklemmen-Controllers

Den Zustand des Busklemmen-Controllers wird unten rechts im System-Manager angezeigt.

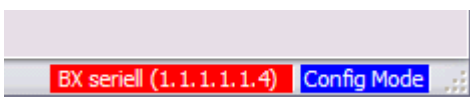


Abb. 34: Zustand des Busklemmen-Controllers

Im *Config Mode* / *FreeRun* kann man jetzt die Konfiguration zum Busklemmen-Controller herunterspielen. Wenn der Busklemmen-Controller im *Stop Mode* ist, ist die ADS-Kommunikation noch nicht aktiviert. Ein Download der Konfiguration ist dann nicht möglich.

Zum Aktivieren der TwinCAT-Konfiguration wählen Sie Ctrl+Shift+F4 oder *Activate Configuration*.

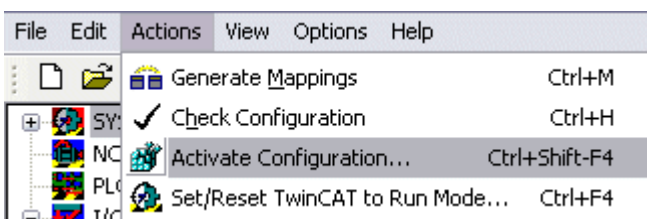


Abb. 35: Aktivieren der TwinCAT-Konfiguration

Die aktuelle Konfiguration wird in dem Busklemmen-Controller geladen. Im Display erscheint *Store Config* und die BUS und I/O LED blinken. Nachdem die Konfiguration erfolgreich im Busklemmen-Controller ist, sollte ein BXxx00 im Display *TwinCAT Config* anzeigen. Jetzt kann man das entsprechende Programm zum Busklemmen-Controller übertragen (Programm-Download über den Feldbus) [[► 115](#)].

4.4.5 Upload einer TwinCAT-Konfiguration

Die TwinCAT-Konfiguration wird per ADS-Protokoll zum Busklemmen-Controller geladen.

Seriellles ADS-Protokoll

(alle Busklemmen-Controller der Serien BCxx50, BCxx20 und BXxx00)

Tragen Sie die serielle ADS-Verbindung ein, wie unter dem Kapitel Seriellles ADS [[► 56](#)] beschrieben ist.

ADS-Protokoll über den Feldbus

(nur BC3150, BC5150, BC9x20, BC9050, BX3100, BX5100, BX9000, BC9191)

Voraussetzung hierfür ist, dass TwinCAT als Master arbeitet und sich im Datenaustausch befindet, d.h. die physikalische, wie auch die Feldbus Konfiguration muss abgeschlossen sein und der Datenaustausch vom Master (z. B. Feldbus-Karte) zum Busklemmen-Controller stattfindet.

Auswahl des Zielsystems

Wählen Sie den Busklemmen-Controller aus, auf den Sie die Konfiguration laden wollen. Mit der Funktionstaste [F8] öffnet sich der Dialog, mit indem Sie Ihr File auf das entsprechende Gerät herunterladen können.

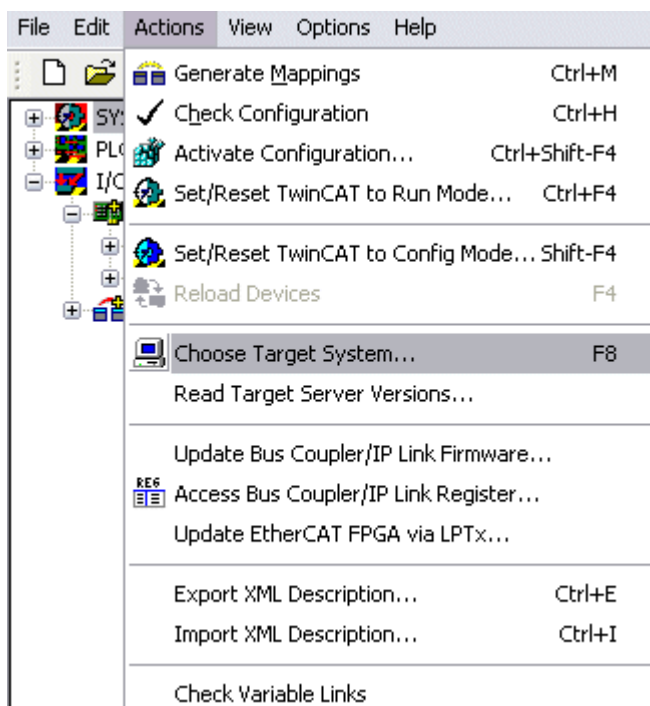


Abb. 36: Auswahl des Zielsystems

Wählen Sie den entsprechenden Busklemmen-Controller aus.

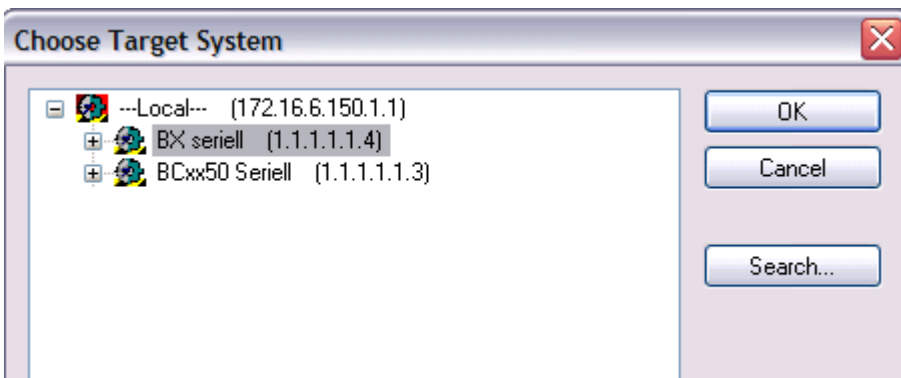


Abb. 37: Auswahl des Busklemmen-Controllers

Der Zustand des Busklemmen-Controllers wird unten rechts im System-Manager angezeigt.

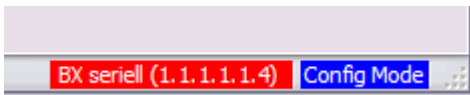


Abb. 38: Zustand des Busklemmen-Controllers

Klicken Sie den roten Ordner an. Die TwinCAT-Konfiguration wird jetzt hochgeladen.

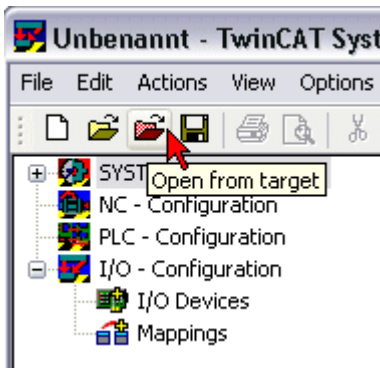


Abb. 39: Hochladen der TwinCAT-Konfiguration

4.4.6 Ressourcen im Busklemmen-Controller

Die im Busklemmen-Controller belegten Speicher-Ressourcen zeigt der System Manager auf dem Karteireiter *Resources* des Busklemmen-Controller an.

Mapping Code

Der Mapping Code wird für die Berechnung der TwinCAT Konfiguration benötigt (siehe Abb. *Speicher für das Code Mapping*). Die Prozentzahlen werden hier zusammen addiert, in dem Beispiel aus Abb. *Speicher für das Code Mapping* sind 8% des Speichers für die Mapping-Berechnung belegt.

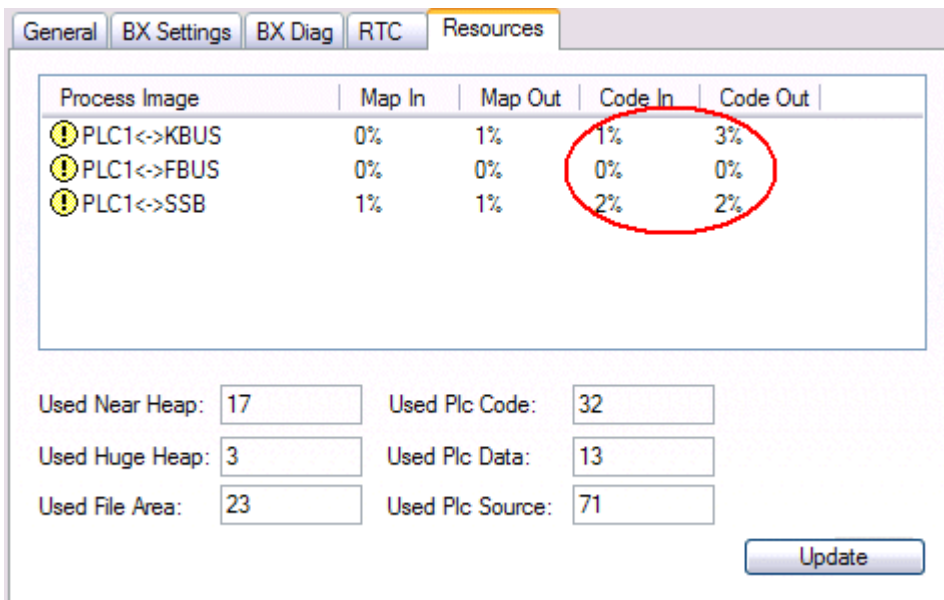


Abb. 40: Speicher für das Code Mapping

Daten Speicher Mapping

Daten Speicher für die Mappings. Die Werte sind einzeln zu betrachten, das bedeutet das jeder der Werte bis zu 100% betragen kann.

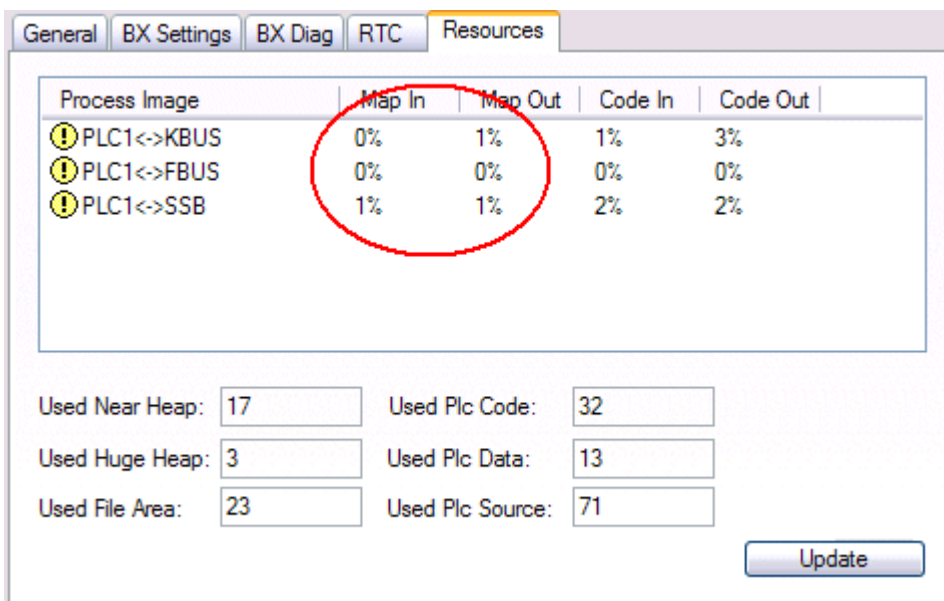


Abb. 41: Daten Speicher Mapping

Verbraucher Code und Daten Speicher

- Abb. Code und Daten Speicher (1) "Used Plc Code" verbrauchter PLC Code, Angabe in %.
- Abb. Code und Daten Speicher (2) "Used Plc Data" verbrauchter PLC Daten, Angabe Speicher in %.
- Abb. Code und Daten Speicher (3) "Used Plc Source" verbrauchter Source Code, Angabe in %.

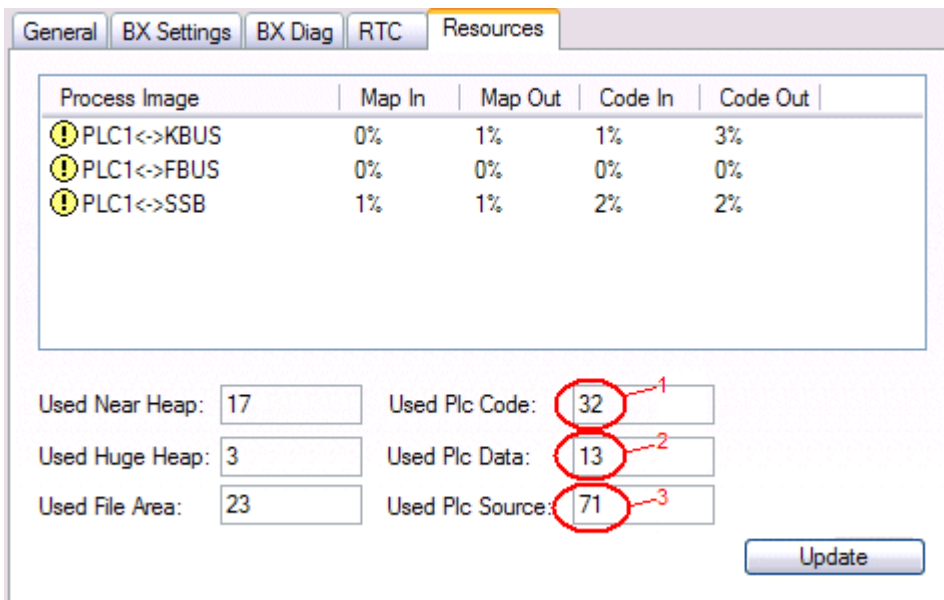


Abb. 42: Code und Daten Speicher

Sonstiger Speicher

Abb. *Sonstiger Speicher* (1) "Used Near Heap" wird für die COM Schnittstelle und SSB benötigt. Angabe in %.

Abb. *Sonstiger Speicher* (2) "Used Huge Heap" wird für die ADS Kommunikation benötigt. Angabe in %. Dieser Wert sollte kleiner 30 % betragen.

Abb. *Sonstiger Speicher* (3) "Used File Area" wird für die TwinCAT Konfiguration, dem TSM-File und dem 16 kByte Flash Zugriff benötigt. Angabe in %.

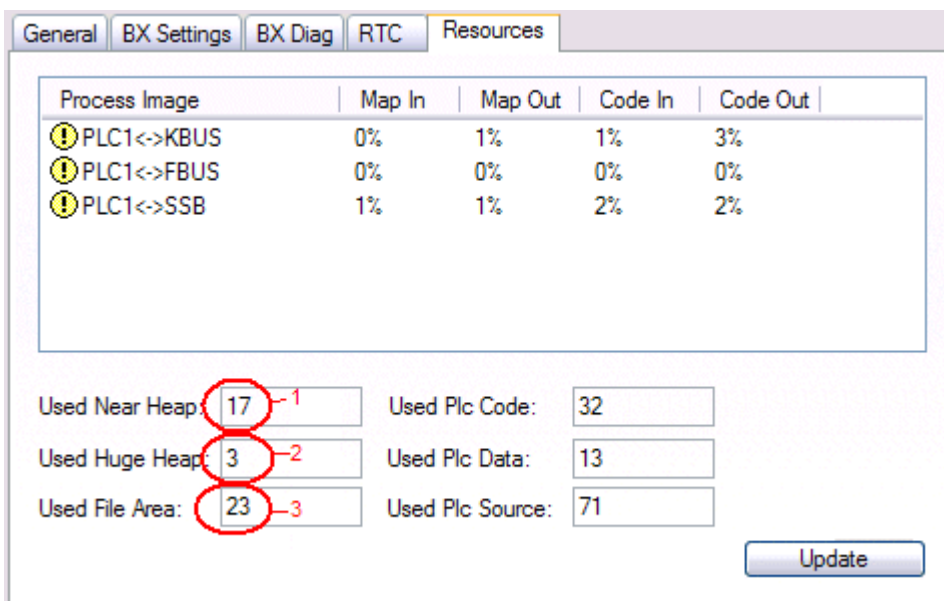


Abb. 43: Sonstiger Speicher

4.4.7 ADS-Verbindung über die serielle Schnittstelle

(ab Firmware-Version 1.xx oder 0.99x - Busklemmen-Controller der BX-Serie und bei allen BCxx50)

Ab TwinCAT 2.9 Build 1020 (TwnCAT Level PLC, NC oder NCI)

● Verwenden Sie nur eine serielle Verbindung

i Um die problemlose Funktion der ADS-Verbindung über die serielle Schnittstelle zu garantieren, ist nur dabei eine serielle Verbindung zum BX-Controller erlaubt. Nach einer erfolgreichen Konfiguration mit dem System Manager schließen Sie diesen, bevor Sie die Programmierung starten.

● AMS-Net-ID im Auslieferungszustand (Default)

i Für BX9000

Die Default AMS-Net-ID lautet 172.16.21.20.1.1. Wird die IP-Adresse des BX9000 verändert, so verändert sich auch die AMS-Net-ID des BX9000. Die aktuelle AMS-Net-ID können Sie mit Hilfe des Menüs sich auf dem Display anzeigen lassen.

Beispiel: Sie ändern die IP-Adresse auf 10.2.3.7 so ändert sich die AMS-Net-ID auf 10.2.3.7.1.1.

Für BC9050, BC9020, BC9120

Die Default AMS-Net-ID lautet 172.16.xxx.[DIP-Switch].1.1. Wird die IP-Adresse des BC9xxx verändert, so verändert sich auch die AMS-Net-ID des BC9xxx.

Beispiel: Sie ändern die IP-Adresse auf 10.2.3.7 so ändert sich die AMS-Net-ID auf 10.2.3.7.1.1.

BC9050: DEFAULT 172.16.21.[DIP-Switch].1.1

BC9020: DEFAULT 172.16.22.[DIP-Switch].1.1

BC9120: DEFAULT 172.16.23.[DIP-Switch].1.1

ADS-Verbindung initialisieren

Tragen Sie den Busklemmen-Controller unter TwinCAT in die Remote-Verbindung ein. Klicken Sie dazu auf das TwinCAT-Icon und öffnen Sie das Eigenschafts-Menü. Unter dem Karteireiter >AMS Remote< können Sie dann folgende Einträge vornehmen.

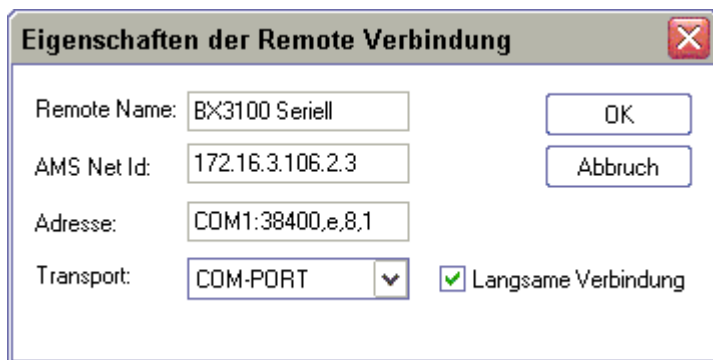


Abb. 44: Eigenschaften der Remote-Verbindung

Remote Name: Beliebig
 AMS-Net-ID: 1.1.1.1.1.1 (Default)
 Adresse: COM Port: Baudrate, Parity, Datenbits, Stop Bits
 Transport: Hier ist "COM-PORT" anzuwählen

Wird der Busklemmen-Controller eingeschaltet, so hat dieser per Default immer die AMS-Net-ID "1.1.1.1.1.1" (außer alle Ethernet Controller).

Sie können diese AMS-Net-ID beliebig ändern. Die neue AMS-Net-ID kann dann allerdings nicht noch einmal geändert werden.

Wenn Sie die AMS-Net-ID erneut ändern möchten, müssen Sie den Busklemmen-Controller zuerst neu starten, so dass die AMS Net Id wieder auf die Default AMS-Net-ID "1.1.1.1.1.1" zurückgesetzt wird. Nun können sie die AMS-Net-ID wiederum einmal ändern.

● Erst beim zweiten Aufruf können Sie Strings eingeben

i Beim ersten Aufruf des Dialogs (siehe oben) können Sie unter Adresse keine Strings eingeben. Tragen Sie Namen, AMS-Net-ID und Transport-Typ ein und schließen Sie den Dialog. Beim zweiten Aufruf können Sie dann Ihre COM-Schnittstelle eintragen.

Die Kommunikation startet sobald TwinCAT im Config- (TwinCAT-Icon ist blau) oder RUN-Modus (TwinCAT-Icon ist grün) ist. Die COM-Schnittstelle wird erst mit einem TwinCAT Stopp (TwinCAT-Icon ist rot) geschlossen und ist dann wieder für andere Programme verfügbar. Sollte die COM-Schnittstelle bei einem TwinCAT-Neustart von einem anderen Programm (z. B. von der Konfigurations-Software KS2000) verwendet werden, so wird kein Fehlerhinweis ausgegeben.

● AMS-Net-ID nach ADS-Verbindung über den Feldbus

i Sollte Sie vor oder während der Nutzung der seriellen ADS-Verbindung den Busklemmen-Controller mit einer ADS-Verbindung über den Feldbus angesprochen haben, ist die AMS-Net-ID vom System Manager automatisch verändert worden. Eine erneute serielle ADS-Verbindung ist dann nur möglich, wenn die AMS-Net-ID angepasst wird.

BX-Serie: auslesen der AMS-Net-ID

Die aktuelle AMS-Net-ID kann aus dem Menü über das Display des Busklemmen-Controller der BX-Serie ausgelesen werden.

AMS AMS-Net-ID
1.1.1.1.1.1

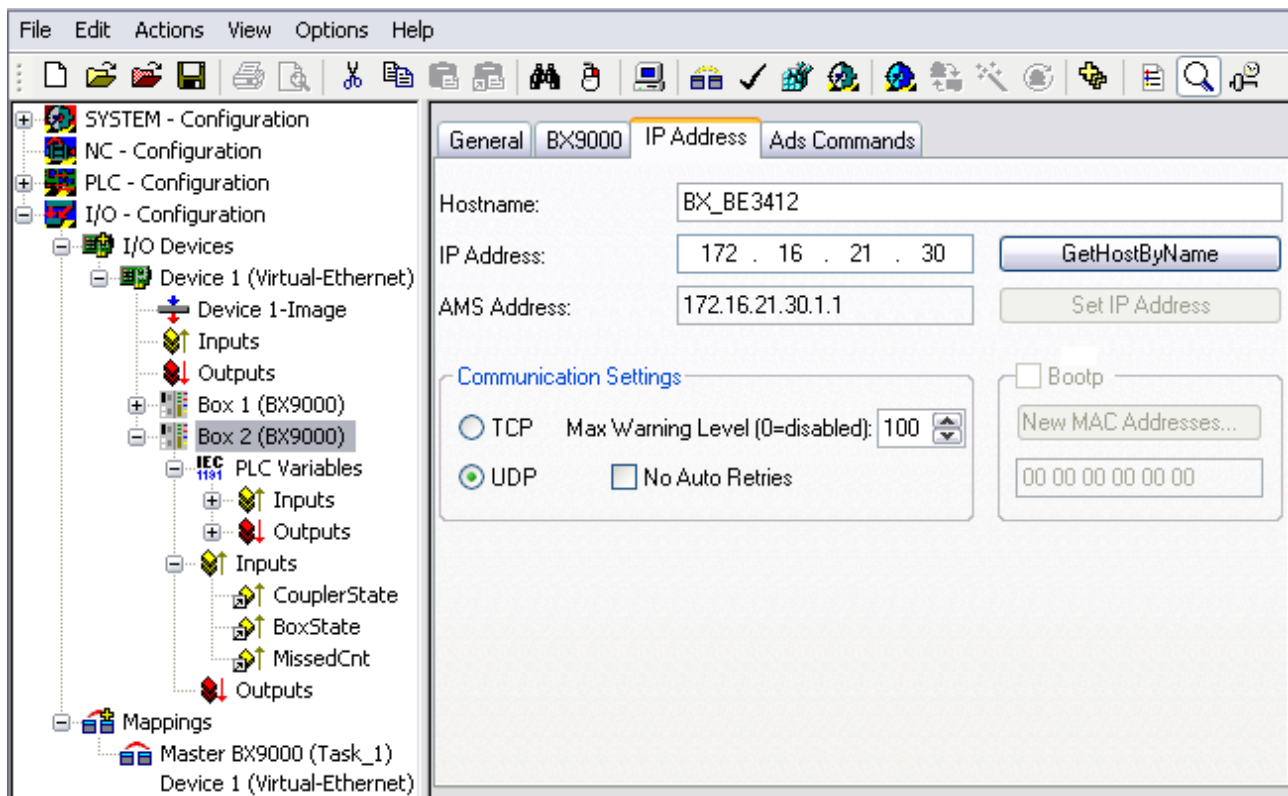
4.4.8 Ethernet

4.4.8.1 TwinCAT PLC als Master

Ein PC oder ein CX kann als Master für den BC9x20/BC9050/BX9000 arbeiten. Er pollt dabei die SPS-Variablen in Abhängigkeit der eingestellten Taskzeit. Damit ist es möglich Daten von einem Leitsystem zum BC9x20/BC9050/BX9000 zu senden wie auch zu empfangen. Folgende Kommunikationsmöglichkeiten sind erlaubt:

- ADS TCP Zyklisch oder Azyklisch aus der SPS mit den ADS READ und WRITE Bausteinen
- ADS UDP Zyklisch
- ModbusTCP (mit TC-Modbus-Client)
- Busklemmen-Controller sendet oder liest Daten aus der SPS mit Hilfe der ADS READ und WRITE Bausteinen

GetHostByName: Diese Funktion erlaubt die Suche der IP-Adresse über den Namen (Funktioniert nur wenn der Busklemmen-Controller seine IP-Adresse über DHCP erhalten hat)



GetHostByName - Suche der IP-Adresse über den Namen

PLC Variablen: Daten für die zyklische Datenverbindung. Diese müssen mit mindestens einer Task verbunden werden. Maximal 256 Worte Ein- oder Ausgänge. Sollten mehr Daten für die Übertragung notwendig sein können diese Azyklisch über den Merkerbereich des Busklemmen-Controller gelesen oder geschrieben werden.

Diagnose Daten:

Coupler State: Sollte immer null sein. "1" wird gesetzt wenn zum Beispiel der K-Bus einen Fehler meldet

BoxState: siehe Comment im Dialog

MissedCnt: Sollte möglichst nicht hochzählen. Da das TwinCAT in Echtzeit läuft, TCP oder UDP aber keine Echtzeit Protokolle sind, ist es nicht auszuschließen das unter Umständen der Zähler sich erhöht. Der Zähler zählt immer dann um eins hoch, wenn die Daten, die er bei Task-Anfang gesendet hat, beim nächsten Task-Anfang noch nicht wieder eingetroffen sind.

Die Task Zeit sollte wie folgt eingestellt werden

Bei ADS TCP Zyklisch

Messen Sie die benötigte PLC Zeit auf dem Busklemmen-Controller und addieren sie 20-30% drauf und stellen Sie so die Task-Zeit auf dem Busklemmen-Controller ein. Nun nehmen Sie die Task-Zeit mal drei und dies entspricht dann der Task-Zeit auf ihrer Mastersteuerung. Beispiel 5 ms gemessene PLC Zeit, stellen Sie 7ms Task-Zeit für den Busklemmen-Controller ein und $3 \times 7 \text{ ms} = 21 \text{ ms}$ für die Master SPS. Bei mehreren Busklemmen-Controller, ist der mit der langsamsten PLC Zeit der entscheidende, der die Task-Zeit auf der Master-Steuerung vorgibt.

Bei ADS UDP Zyklisch

Messen Sie die benötigte PLC Zeit auf dem Busklemmen-Controller und addieren sie 20-30% drauf und stellen Sie so die Task-Zeit auf dem Busklemmen-Controller ein. Nun nehmen Sie die Task-Zeit mal zwei und dies entspricht dann der Task-Zeit auf ihrer Mastersteuerung. Beispiel 5 ms gemessene PLC Zeit, stellen Sie 7ms Task-Zeit für den Busklemmen-Controller ein und $2 \times 7 \text{ ms} = 14 \text{ ms}$ für die Master SPS. Bei mehreren Busklemmen-Controller, ist der mit der langsamsten PLC Zeit der entscheidende, der die Task-Zeit auf der Master-Steuerung vorgibt.

Bei ModbusTCP Zyklisch

Messen Sie die benötigte PLC Zeit auf dem Busklemmen-Controller und addieren sie 20-30% drauf und stellen Sie so die Task-Zeit auf dem Busklemmen-Controller ein. Nun nehmen Sie die Task-Zeit mal zwei und dies entspricht dann der Task-Zeit auf ihrer Mastersteuerung. Beispiel 5 ms gemessene PLC Zeit, stellen Sie 7ms Task-Zeit für den Busklemmen-Controller ein und $2 \times 7 \text{ ms} = 14 \text{ ms}$ für die Master SPS. Bei mehreren Busklemmen-Controller , ist der mit der langsamsten PLC Zeit der entscheidende, der die Task-Zeit auf der Master-Steuerung vorgibt.

Unterschiedliche SPS-Zykluszeiten

Wenn die Busklemmen-Controller Ihrer Anlage unterschiedlich lange Zykluszeiten für ihre lokale SPS-Abarbeitung benötigen, können Sie die Zeit nach der die übergeordnete TwinCAT PLC jeden einzelnen Busklemmen-Controller abfragt individuell anpassen.

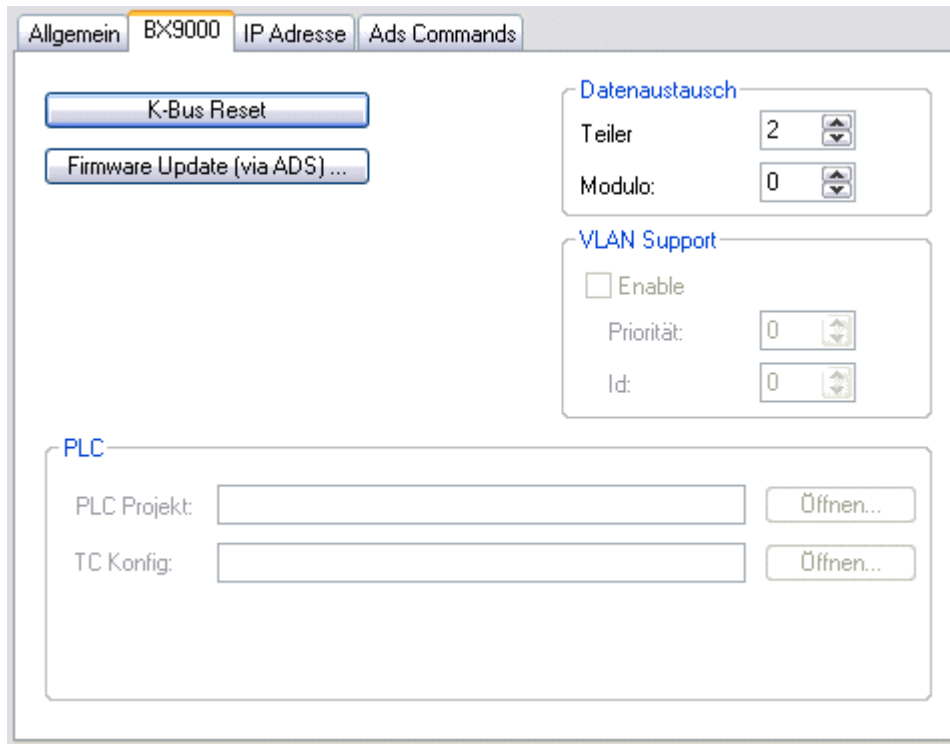


Abb. 45: Anpassen der Zeit nach der die TwinCAT PLC einen Busklemmen-Controller abfragt

Teiler

Benutzen Sie hierfür den Teiler. Dieser nimmt als Grundlage die Zykluszeit der übergeordneten TwinCAT PLC, zum Beispiel 10 ms. Wird der Teiler nun auf 2 gestellt so wird alle $2 \times 10 \text{ ms}$ also alle 20 ms ein Telegramm zum Busklemmen-Controller gesendet.

Modulo

Mit Modulo können Sie zusätzlich einstellen, wann das übergeordnete TwinCAT PLC dies machen soll. Beispiel:

Teiler 3, Modulo 0, bedeutet nach dem 1. Task-Zyklus und dann nach jedem 3. Task-Zyklus wird ein Telegramm versendet.

Steht der Modulo auf 1 wird erst nach dem 2.Task-Zyklus ein Telegramm versendet und dann nach jedem 3.Taskzyklus + 1.

So können Sie bei vielen Ethernet-Knoten die Anzahl der Ethernet-Pakete besser verteilen, haben eine gleichmäßigere Netzwerkauslastung und keine Spitzen in der Netzwerklast.

4.4.9 K-Bus

i Busklemme und Endklemme erforderlich

Zum Betrieb eines Busklemmen-Controllers der BC- oder BX-Serie müssen an dessen K-Bus mindestens eine Busklemme mit Prozessabbild und die Endklemme gesteckt sein!

Karteireiter BX Settings

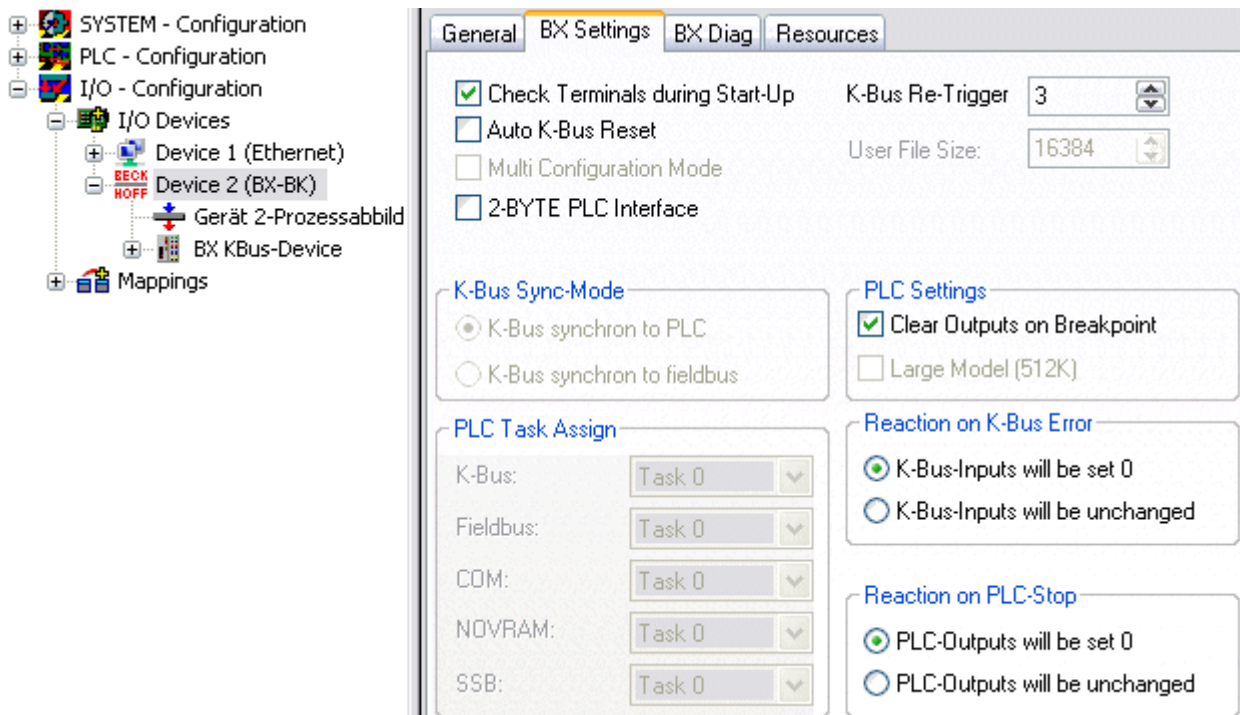


Abb. 46: Karteireiter BX Settings

Check Terminals during Start up

Beim Erzeugen eines Bootprojektes wird die aktuelle Busklemmenkonfiguration gespeichert. Beim Erneuten starten des Busklemmen-Controllers werden die angeschlossenen Busklemmen überprüft. Ist diese Option angewählt, geht der Busklemmen-Controller nicht in den Datenaustausch. Das PLC Projekt wird nicht gestartet.

Auto K-Bus Reset

Nach Behebung eines K-Bus-Fehlers geht der Busklemmen-Controller automatisch wieder in den Datenaustausch.

⚠ VORSICHT

Nach Behebung eines K-Bus Fehlers werden die Ausgänge sofort wieder aktiv!

Achten Sie darauf, dass die Ausgänge dann sofort wieder aktiv werden und analoge Ausgänge ihren programmierten Wert erhalten, wenn dies nicht in Ihrem PLC-Projekt beachtet wurde.

Clear Outputs on Breakpoint

Wenn Breakpoints im PLC Control gesetzt werden, wird der K-Bus nicht mehr bearbeitet, das heißt die Ausgänge werden in den sichern Zustand, sprich null, gesetzt.

K-Bus Sync Mode

Das Schreiben und lesen der Busklemmen kann synchron zur Task1 oder dem Feldbus stattfinden.

K-Bus Re-Trigger

Sollte das PLC Projekt oder der SSB länger den Prozessor belasten, kann der K-Bus eine Zeit nicht mehr bearbeitet werden. Das führt dazu, dass der Watchdog der Busklemmen zuschlägt und Ausgänge abfallen. Der Busklemmen-Controller ist so eingestellt, das nach 85 ms der K-Bus Watchdog nachgetriggert wird und das 3 mal. Danach würde der K-Bus Watchdog zuschlagen.

K-Bus Re-Trigger 0: 100 ms

K-Bus Re-Trigger 1: 2 x 85 ms = 170 ms

K-Bus Re-Trigger 2: 3 x 85 ms = 255 ms

K-Bus Re-Trigger 3: 4 x 85 ms = 340 ms

Reaktion auf K-Bus Fehler

Bei K-Bus Fehler werden die K-Bus Eingänge auf "0" geschrieben oder behalten Ihren letzten Stand.

Reaktion auf PLC-Stop

Wird das PLC Projekt gestoppt, kann man Einstellen, wie sich die PLC Feldbus Ausgangsdaten verhalten sollen. Im Master sind diese Daten dann Eingangsdaten. Die Daten können bei PLC Stop auf "0" geschrieben oder unverändert gelassen werden.

Karteireiter BX Diag

Anzeige der Zykluszeit für Task 1, K-Bus, Bearbeitung Feldbus und die Auslastung des SSB.

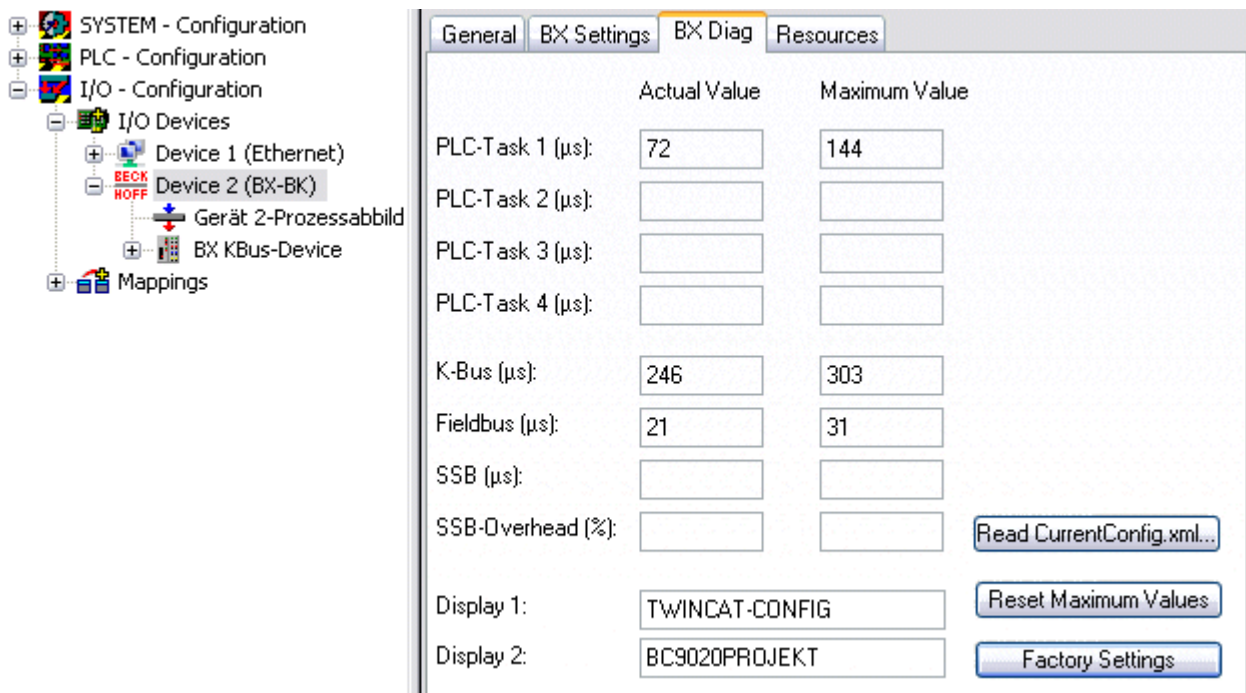
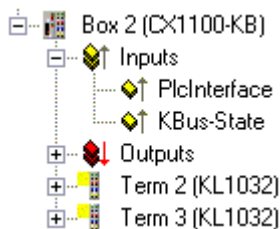


Abb. 47: Karteireiter BX Diag

Factory Settings - Der Busklemmen-Controller wird in seinen Auslieferungszustand gesetzt. Mit Restart System oder aus- und einschalten sind diese Einstellungen wieder gültig (Display DEFAULT-CONFIG).
Reset Maximum Values - löscht die Maximalen Werte

K-Bus Variablen



PLC-Interface: Wird nicht unterstützt (nur enthalten um CX oder BX Projekte zu verschieben)

K-Bus-State: Siehe Diagnose

4.4.10 PLC

4.4.10.1 Einfügen eines PLC-Projektes

Für ein variables Mapping muss im System Manager die Konfiguration festgelegt sein. Hier vereinbart man die Verknüpfung zwischen PLC und der Hardware. Die Variablen können Bit, Byte, Wort oder Datenstrukturen verarbeiten und verknüpfen. Eine automatische Adressierung durch den System Manager ist möglich, sollte aber auf ihren Offset überprüft werden.

i Word Alignment - Byte Orientierung

Achten Sie bei Datenstrukturen darauf, dass der Busklemmen-Controller die Daten in Word Alignment speichert und der System Manager Byte orientiert arbeitet (siehe [Datenstrukturen](#) | 77)

Im PLC-Control muss ein gültiges Projekt übersetzt und gespeichert sein. Diese Daten werden als *.tpy Datei abgespeichert. Um ein PLC-Projekt einzufügen klicken Sie mit der rechten Maustaste auf *PLC-Configuration*. Wählen Sie Ihr aktuelles PLC Projekt aus.

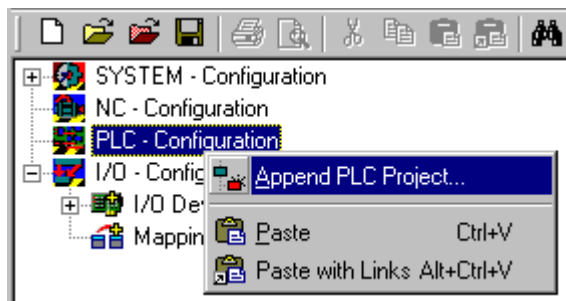


Abb. 48: Auswahl des PLC-Projekts

Verbinden Sie die PLC-Variable mit der Hardware (z. B. digitale Busklemme).

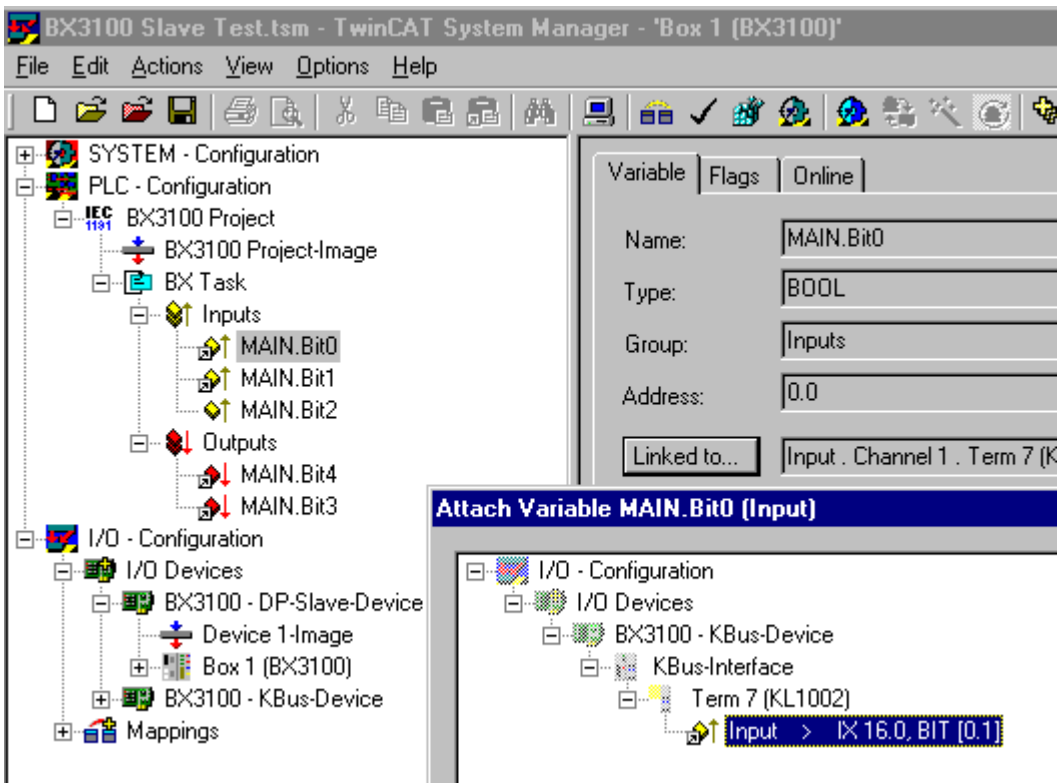


Abb. 49: Verbinden vom PLC-Variable und Hardware

Nachdem alle Verknüpfungen erstellt sind aktivieren Sie die Konfiguration *Actions/Activate Configuration* (Ctrl+Shift+F4) und Starten Sie TwinCAT *Set/Reset TwinCAT to Run Mode*. Achten Sie darauf, dass Sie das richtige Zielsystem angewählt haben (unten rechts im Fenster des System-Managers).

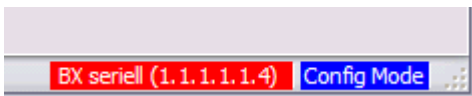


Abb. 50: Anzeige des Ziel-Systems

4.4.10.2 Messen der PLC-Zykluszeit

Die Task-Zeit wird im PLC Control eingestellt. Die Default Einstellung beträgt 20 ms.

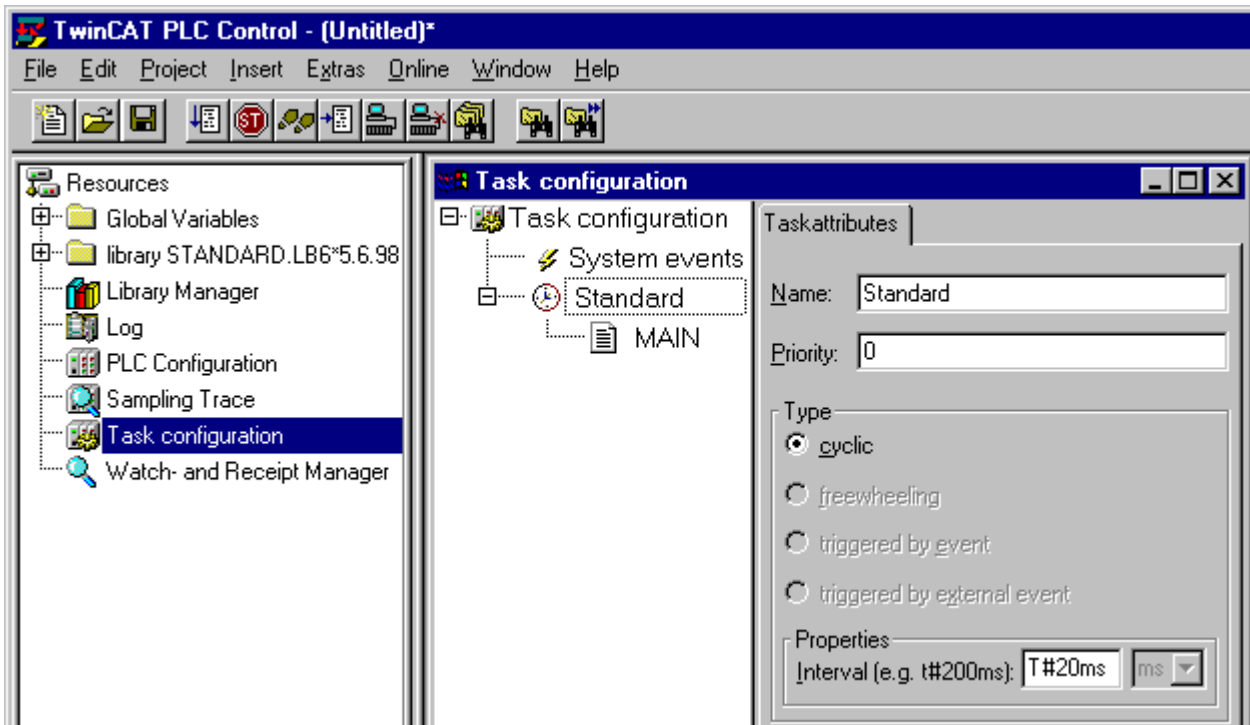


Abb. 51: Einstellen der Task-Zeit

Das PLC-Programm wird in der Default-Einstellung alle 20 ms erneut aufgerufen, solange die allgemeine Zykluszeit kleiner als 20 ms ist. Um die Auslastung ihres Systems zu ermitteln kann im System Manager die Zykluszeit der PLC gemessen werden. Um einen Problemlosen Betrieb zu garantieren, muss die eingestellte Task-Zeit 20 bis 30 % höher sein als die gemessene gesamt Zykluszeit. Eine genauere Aufschlüsselung der Zykluszeit finden Sie unter der Beschreibung [K-Bus-Reiter \[► 60\]](#). Die gesamte Zykluszeit wird mit der TcBase Bibliothek angezeigt (siehe TcBase.lbx oder TcBaseBCxx50.lbx).

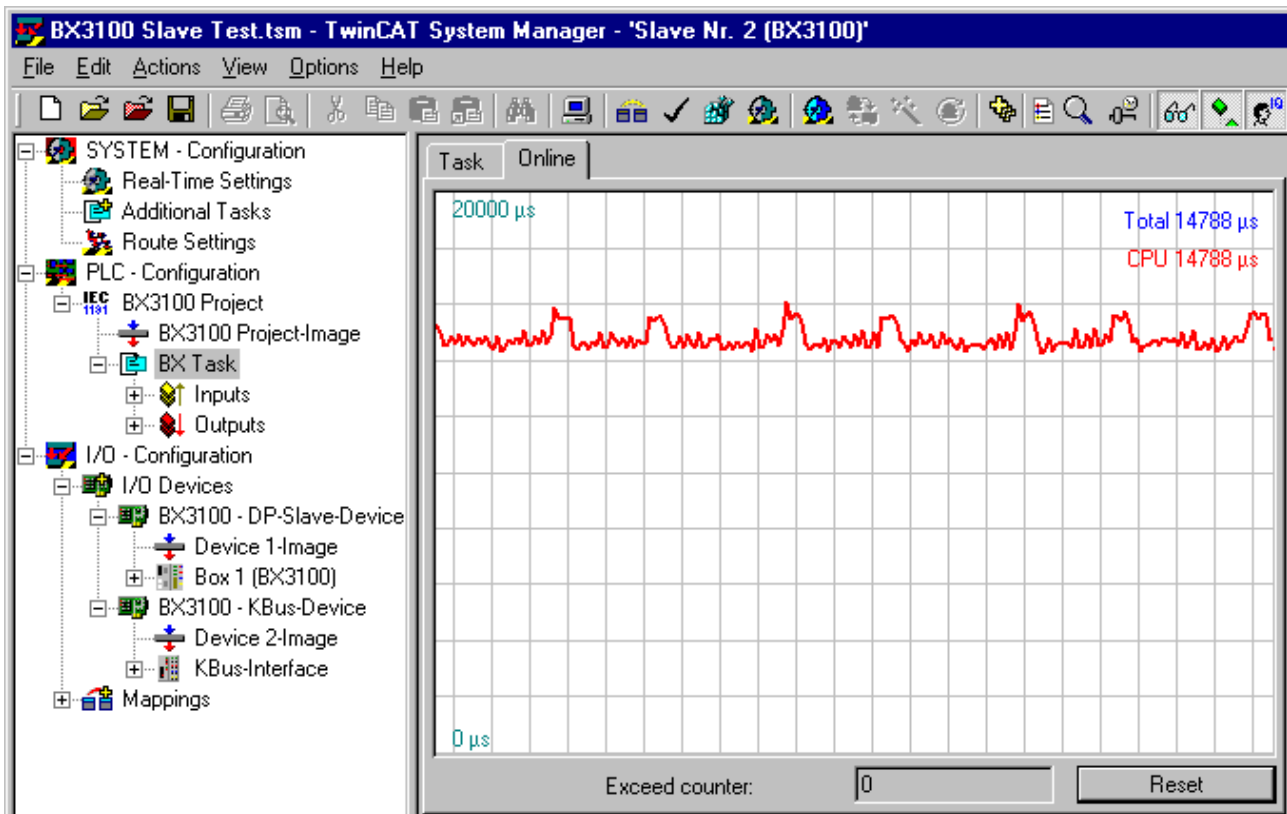


Abb. 52: Anzeige der PLC-Zykluszeit

4.5 KS2000

4.5.1 Konfigurations-Software KS2000

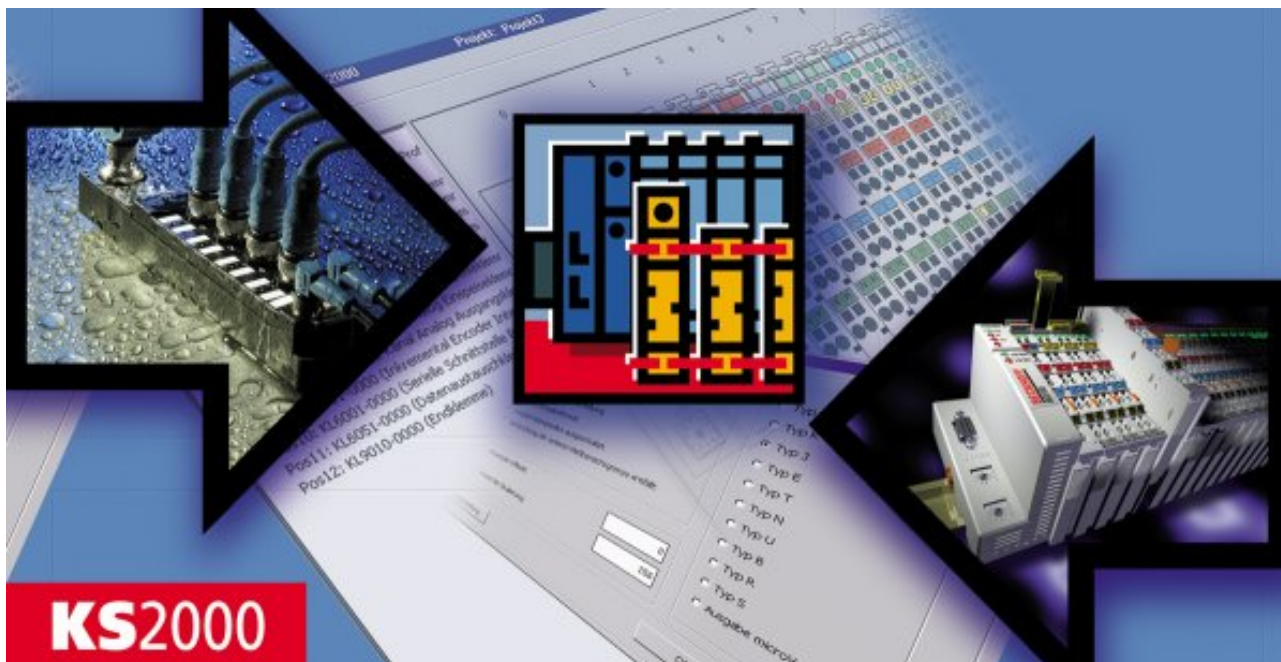


Abb. 53: Konfigurations-Software KS2000

Eine Parametrierung und Konfiguration der Busklemmen-Controller der BC9x20 und der BC9050 ist mit der Konfigurations-Software KS2000 nicht notwendig. Die Konfiguration kann mit den TwinCAT System Manager vorgenommen werden.

Die Konfigurations-Software KS2000 kann Ihnen aber bei der Konfiguration oder Diagnose der an dem Busklemmen-Controller angereihten Busklemmen helfen.

Es empfiehlt sich hierzu die Baudrate in der Konfigurations-Software KS2000, auf 38400 Baud zu stellen (8 Datenbits, Even, 1 Stopp-Bit).

● COM1 - automatische Baudraten Erkennung

i Die Schnittstelle COM 1 der Controller hat eine automatische Baudraten Erkennung von 9,6 kBaud bis 38,4 kBaud.

● Erforderliche KS2000-Version

i Die Konfiguration oder Diagnose von Busklemmen wird ab KS2000 Version 4.3.0.65 unterstützt.

Um bei einigen Busklemmen die Konfigurationsdialoge nutzen zu können (z. B. KL25xx, KL6811, KL6201, KL6401) müssen folgende Punkte eingestellt werden:

- Ein SPS-Projekt oder ein Boot-Projekt muss deaktiviert sein.
- Der Controller muss in der Default-Konfiguration sein. Setzen Sie die Herstellereinstellung oder schalten Sie im TwinCAT System Manager auf Config-Mode (blaues TwinCAT Icon).
- Der Controller muss im FreeRun-Modus sein. Aktivieren Sie dies mit dem TwinCAT System Manager.

Nun können Sie sich mit der Konfigurations-Software KS2000 über ADS (Port 100) oder das serielle Kabel einloggen und die KS2000-Dialoge der Busklemmen nutzen.

5 Programmierung

5.1 PLC-Eigenschaften

BC9050

Beschreibung	Wert
Datenspeicher	32 kByte
Programmspeicher	48 kByte minus Task-Konfiguration minus POU's beim Online Change
Source Code Speicher	128 kByte
RETAIN	2 kByte
Persistente Daten	1000 Byte
INPUT	2 kByte
OUTPUT	2 kByte
MERKER	4 kByte
max. grÖÙe einer Variable	16 kByte
max. POU's	Beschränkung durch Speicher

BC9020

Beschreibung	Wert
Datenspeicher	128 kByte
Programmspeicher	128 kByte minus Task-Konfiguration minus POU's beim Online Change
Source Code Speicher	256 kByte
RETAIN	2 kByte
Persistente Daten	1000 Byte
INPUT	2 kByte
OUTPUT	2 kByte
MERKER	4 kByte
max. grÖÙe einer Variable	16 kByte
max. POU's	Beschränkung durch Speicher

BC9120

Beschreibung	Wert
Datenspeicher	128 kByte
Programmspeicher	128 kByte minus Task-Konfiguration minus POU's beim Online Change
Source Code Speicher	256 kByte
RETAIN	2 kByte
Persistente Daten	1000 Byte
INPUT	2 kByte
OUTPUT	2 kByte
MERKER	4 kByte
max. grÖÙe einer Variable	16 kByte
max. POU's	Beschränkung durch Speicher

5.2 TwinCAT PLC

Das Beckhoff TwinCAT Software-System verwandelt jeden kompatiblen PC in eine Echtzeitsteuerung mit Multi-SPS-System, NC-Achsregelung, Programmierumgebung und Bedienstation. Die Programmierumgebung von TwinCAT wird auch für die Programmierung der BC/BX genutzt. Wenn Sie TwinCAT PLC (Windows NT4/2000/XP) installiert haben, können Sie die Feldbus-Verbindung oder die serielle Schnittstelle für Software-Download und Debugging verwenden.

TwinCAT I/O oder TwinCAT PLC können auch als Ethernet-Master (Host) genutzt werden, um Prozessdaten mit dem Busklemmen-Controller auszutauschen. TwinCAT stellt ihnen hierzu den System Manager als Konfigurationstool sowie die Treiber und das ADS-Protokoll zu Verfügung.

Busklemmen-Controller der Serien BCxx50, BCxx20 und BXxx00

Diese Busklemmen Controller der 2. Generation werden mit dem TwinCAT System Manager konfiguriert und mit TwinCAT PLC-Control programmiert. Sie müssen für diese Koppler TwinCAT PLC installieren (Windows NT4, Windows 2000, Windows XP).

Programmierung und Programmübertragung

- [über die serielle Schnittstelle \[► 116\]](#)
- [über die Feldbus-Schnittstelle \[► 115\]](#) (nur bei Busklemmen-Controllern für PROFIBUS, CANopen und Ethernet)

Online Change

Die Busklemmen-Controller Busklemmen-Controller der BX-Serie und die BCxx50 unterstützen Online Change. Das bedeutet, dass das PLC-Programm durch ein neues Programm ersetzt wird, ohne dass dadurch das Programm unterbrochen wird. Nach Beendigung der Task wird auf das neue Programm umgeschaltet. Damit verbunden ist die doppelte Haltung des SPS Programms. Es stehen 512 kByte zu Verfügung, die für die doppelte Haltung des PLC Programm durch zwei dividiert werden muss, also 256 kByte. Zusätzlich müssen noch für Task-Konfiguration etc. einige kByte abgezogen werden. Beim Online Change werden noch dynamische Daten in den Speicher abgelegt. Sollte ein Programm an die Speichergrenze stoßen (größer 240 kByte) kann es passieren das ein Online Change nicht mehr funktioniert auch wenn das Programm nach einem "Rebuild all" wieder in den BX geschrieben werden kann.

Wann geht kein Online Change?

Es gibt einige Punkte, nachdem ist ein Online Change nicht mehr möglich.

- einfügen einer neuen Bibliothek
- Änderung der Task-Einstellung
- ein "Rebuild all"
- Grenzbereich in der Speicherauslastung des Controllers (PLC Programm größer 90%)

5.3 TwinCAT PLC - Fehler-Codes

Fehlerart	Beschreibung
PLC-Kompilierfehler	Maximum number of POUs (...) exceeded
PLC-Kompilierfehler	Out of global data memory ...

Fehler POUs

Für jeden Baustein wird eine POU (Process Object Unit) angelegt. Per Default sind 256 Bausteine möglich.

Error 3612: Maximum number of POUs (100) exceeded! Compile is aborted.

Data allocation
1 Error(s), 0 Warning(s).

Abb. 54: Maximale Anzahl der POUs überschritten

Wenn man Bibliotheken einbindet, kann dieser Wert nicht mehr ausreichen. Erhöhen Sie dann die Anzahl an POUs.

Öffnen Sie dazu im PLC Control unter Projekte/Optionen...

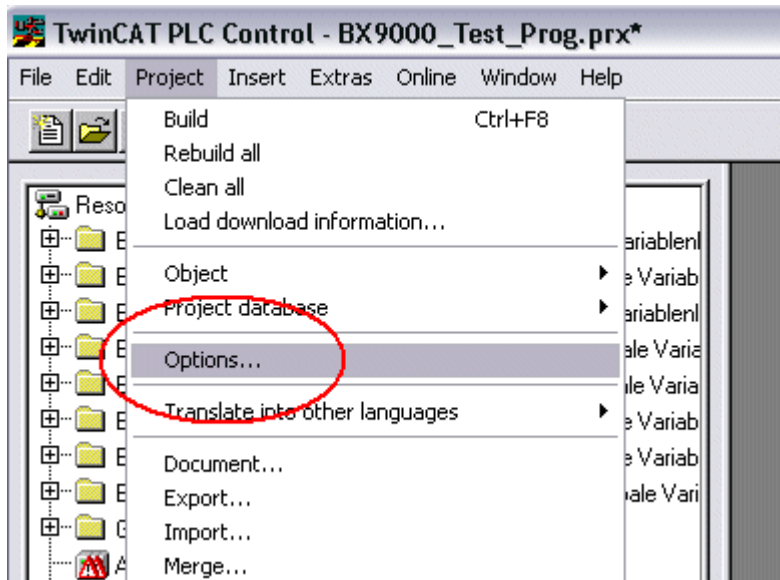


Abb. 55: Menüpfad Projekte / Optionen / Controller Settings

...die Controller Settings.

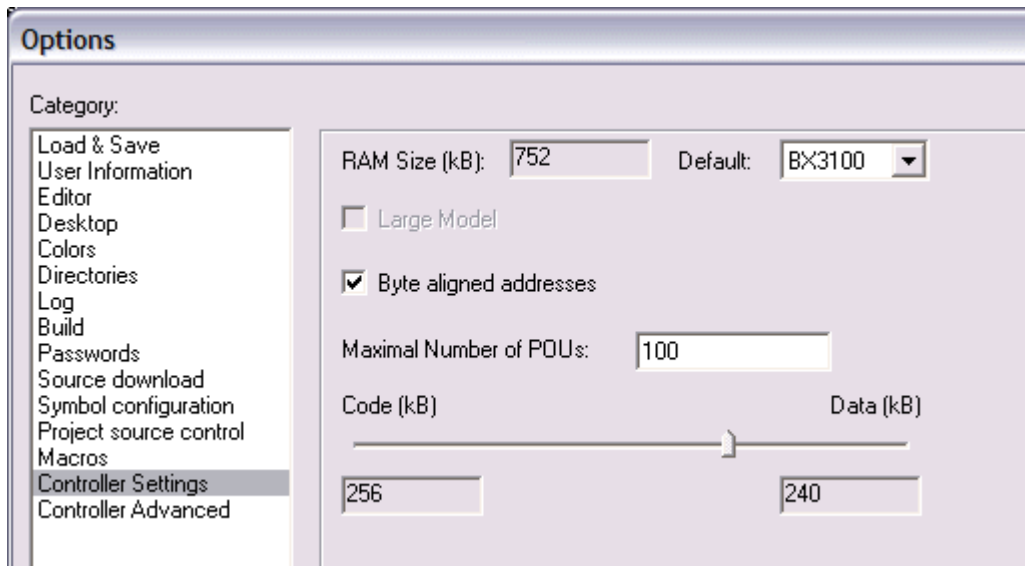


Abb. 56: Controller Settings

Eine Änderung dieser Einstellungen hat zur Folge, dass kein Online Change mehr geht.

Fehler Globaler Speicher

```
Interface of POU 'MAIN'
Data allocation
Error 3803: MAIN (7): Out of global data memory. Variable 'Test_', 16002 bytes.
1 Error(s), 0 Warning(s).
```

Abb. 57: Globaler Speicher nicht ausreichend

Per Default sind 2 x 16 kByte Daten angelegt. Wenn viele Daten benutzt werden sollen, müssen Sie diesen Bereich vergrößern. Beim BX sind maximal 14 Datensegmente möglich.

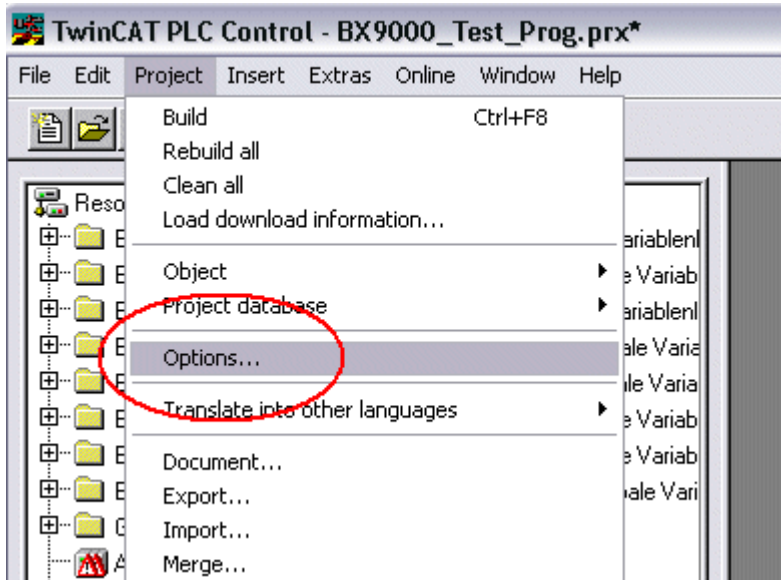


Abb. 58: Menüpfad Projekte / Optionen / Build

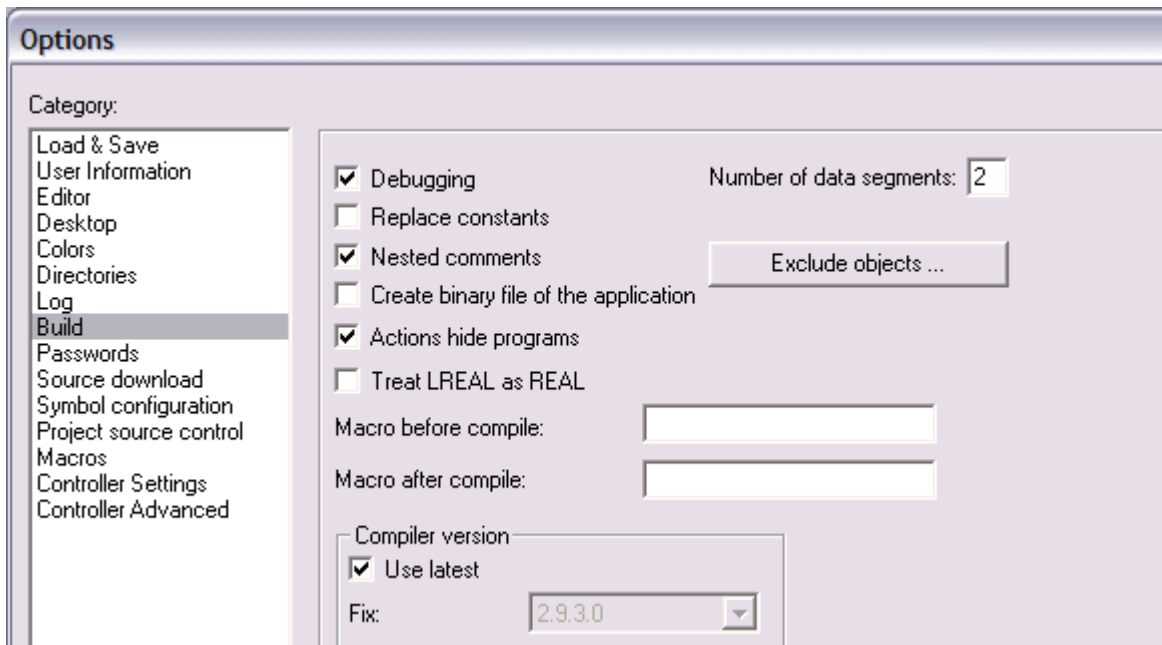


Abb. 59: Build

5.4 Remanente Daten

Auf dem BC9191 und den BX-Controller stehen 2000 Byte an remanenten Daten zur Verfügung. Diese Daten werden im PLC Control als VAR RETAIN deklariert.

Beispiel

```
VAR RETAIN
  Test      :BOOL;
  Count     :INT;
END_VAR
```

Zwischen VAR RETAIN und END_VAR stehen die Retain Daten. Diese Daten werden in einem NOVDRAM gespeichert und sind über den ganzen 2 kByte großen Bereich konsistent. Die RETAIN Daten werden nach jedem Zyklus ins NOVDRAM gespeichert. Für 2 kByte werden ca. 2 ms benötigt (für 1 kByte ca. 1 ms). Die Variablen können lokal oder global Konfiguriert sein. Variablen die lokiert sind (%MB, %QB, %IB) können nicht als Remanente Daten genutzt werden.

● VAR_RETAIN nicht in Funktionsbausteinen benutzen

i VAR_RETAIN sollte nicht in Funktionsbausteinen benutzt werden. Sämtliche Daten in einem FB werden in den Retain Speicher kopiert, damit erhöht sich unnötig die Zykluszeit und der Retain Speicher wird mit unnötigen Datenmengen gefüllt.

● Variablen mit Adresse nicht als remanente Daten verwenden

i Variablen die auf einer Adresse liegen (%MB, %QB, %IB) dürfen nicht als remanente Daten verwendet werden.

Beispiel für remanente Daten im Funktionsbaustein

Da immer alle Daten eines Funktionsbausteins, in dem auch nur ein remanentes Bit zu finden ist, gespeichert wird, sollte dies möglichst vermieden werden. Im Anschluss finden Sie ein Programmbeispiel.

Funktionsbaustein Test (Kein Programm Code notwendig - in ST reicht ein Semikolon)

```
FUNCTION_BLOCK Test
VAR_INPUT
END_VAR
VAR_OUTPUT
END_VAR
VAR
END_VAR
VAR_IN_OUT
  Counter    :INT;
END_VAR
```

Programm MAIN

```
PROGRAM MAIN
VAR
  fb_Test:Test;
END_VAR
VAR RETAIN
  iCounter1:INT;
END_VAR
fb_Test(Counter:=iCounter1);
```

5.5 Persistente Daten

Es stehen auf dem Busklemmen-Controller 1000 Byte an persistenten Daten zur Verfügung. Im Unterschied zu den Retain-Daten werden diese auch bei einem neuen Projekt, bei einem Reset der SPS sowie bei einem neuen Download nicht gelöscht.

Um die persistenten Daten zu nutzen müssen diese erst einmal mit einem Funktionsbaustein aus der SPS heraus aktiviert werden.

Zweitens müssen die Variablen auf dem lokierten Merkerbereich liegen. Hier steht es Ihnen frei wo ihre persistenten Daten liegen. Es stehen 4 kByte an lokierten Merkern zur Verfügung und sie können davon 1000 Byte als persistente Daten deklarieren.

Beispiel

```
VAR
  Test AT %MX1000 :BOOL;
  Count AT %MB1002 :INT;
END_VAR
```

Mit dem Baustein **Persistent_Data** legen sie die Anfangsadresse fest und die Länge in Byte ab der die Daten persistent sein sollen.

Mit der Eingangsvariable *WriteOffset* gibt mal den Byte Offset des Merkerbereichs an, mit *WriteSize* die Länge in Byte.

Den Baustein finden Sie in der TcSystemBX.lbx Bibliothek. Sollte diese nicht vorhanden sein, laden Sie sie sich aus dieser Dokumentation herunter (siehe Bibliotheken).

Beispielwerte

WriteOffset 1000
WriteSize 10


Alle Daten die im Bereich %MB1000 - %MB1009 sind dann persistent. Es spielt keine Rolle um welchen Variablen Typ es sich handelt.

Die Daten werden wie bei den Retain Daten in NOVRAM kopiert und sind daher in jedem Zyklus beschreibbar.

i **Persistenten Daten ab Firmware 1.17**
Die persistenten Daten werden bei allen BX-Controllern ab Firmware 1.17 oder höher unterstützt.

i **Parameter sind sofort gültig**
Das Schreiben der Parameter muss nur einmal erfolgen und ist sofort gültig. Diese Daten werden dauerhaft gespeichert.
Das aktivieren der Herstellereinstellung löscht dies wieder und auch die persistenten Daten werden gelöscht.

Beispielprogramm

Klicken Sie auf den Link  (<https://infosys.beckhoff.com/content/1031/bc9xx0/Resources/zip/3207307659.zip>) um ein Beispielprogramm aus dieser Dokumentation herunterzuladen.

5.6 Lokierte Merker

Es stehen 4 kByte an lokierten Merkern zur Verfügung. Diese können genutzt werden, um unterschiedliche Variabelentypen auf die gleiche Adresse zu legen, zum Beispiel um aus Strings Bytes zu machen. Weiter können Daten hinterlegt werden, die per ADS von der Steuerung gelesen und/oder geschrieben werden können.

i **Lokierte Variablen sind keine remanenten Daten**

Bei den Busklemmen-Controller der BX-Serie und den BCxx50 werden die lokierte Variablen **nicht** als remanente Daten gespeichert.

Read / Write lokierter Merker per ADS

Die Merker können auch über die Steuerung per ADS ausgelesen werden. Bei PROFIBUS werden dazu die DPV-1 Dienste genutzt, bei CANopen die SDO Kommunikation.

Die AmsNetId ist aus dem System Manager zu entnehmen oder man kann Sie sich in dem Menü des Busklemmen-Controllers Anzeigen lassen.

Die Port Nummer ist 800 für die PLC.

Index Group	Bedeutung	Index Offset (Wertebereich)
0x4020	Merker (nur BXxxx0)	0..4096

Beispiel

BX Programm

```
VAR
    Flag_01 AT %MB0: WORD;
END_VAR
```

TwinCAT PC/CX Master Programm

```
VAR
    fbADRSREAD: ADSREAD;
    Flag_M: WORD;
END_VAR

fbADRSREAD (
    NETID:='172.16.3.0.2.3' , (* AMSNetId BX *)
    PORT:=800 , (* 800 - PLC *)
    IDXGRP:=16#4020 , (* 0x4020hex falgs *)
    IDXOFFS:=0 , (* byte offset *)
    LEN:=2 , (* Lenght byte *)
    DESTADDR:=ADR(Merker) ,
    READ:=TRUE ,
    TMOUT:=t#1s );
IF NOT fbADRSREAD.BUSY THEN
    fbADRSREAD(READ:=FALSE);
END_IF
```

5.7 Lokales Prozessabbild im Auslieferungszustand (Default Config)

Das Prozessabbild der Busklemmen-Controller besteht aus Eingangs-, Ausgangs- und Merkerbereich. Daneben gibt es noch die unlokierten Daten. Diese Daten besitzen keine feste Adresse. Sie werden ohne Angabe einer Adresse angelegt. Für diese Art der Variablen stellt

- der BCxx50 48 kByte,
- der BC9x20, BC9191 128 kByte und
- der BXxx00 256 kByte Speicher zur Verfügung.

Eine Variable oder Struktur (Array) darf maximal 16 kByte groß sein. Für die lokierten Daten stehen 2048 Byte Ein- und 2048 Byte Ausgangsdaten bereit. Für den lokierten Merkerbereich verfügt der Busklemmen-Controller über 4 kByte Speicher.

Im Auslieferungszustand (Default Configuration) des BX/BCxx50 werden allen angeschlossenen Busklemmen feste Adressen zugewiesen. Die Daten für die Ethernet-Kommunikation beginnen ab dem Adress-Offset 1000_{dez}. Die Länge der Ethernet-Daten hängt von der Anzahl der konfigurierten Daten ab und beträgt maximal 1000 Byte beim BX9000.

Eingänge	Ausgänge
Busklemme %IB0 ...	Busklemmen %QB0 ...
Ethernet DATEN (SPS- Variablen) %IB1000 ... (ModbusTCP/ADS-TCP/ADS-UDP)	Ethernet DATEN (SPS- Variablen) %QB1000 ... (ModbusTCP/ADS-TCP/ADS-UDP)
... %IB2047 Maximal	... %QB2047 Maximal

Adressierung der Angeschlossenen Busklemmen

Alle angeschlossenen Busklemmen werden in der Default-Einstellung dem lokalen Prozessabbild zugewiesen. Das Mapping im Busklemmen Controller erfolgt nach folgender Gesetzmäßigkeit: Erst alle komplexen Busklemmen, in der Reihenfolge wie diese gesteckt sind und anschließend die digitalen Busklemmen, die zu einem Byte aufgefüllt werden. Das Default-Mapping der komplexen Busklemmen ist:

- komplette Auswertung
- Intel-Format
- Word Alignment

Beispielaufbau

Busklemmen Controller: 1 x BCxx50, BCxx20 oder BXxx00
 Position 1: 1 x KL1012
 Position 2: 1 x KL1104
 Position 3: 1 x KL2012
 Position 4: 1 x KL2034
 Position 5: 1 x KL1501
 Position 6: 1 x KL3002
 Position 7: 1 x KL4002
 Position 8: 1 x KL6001
 Position 9: 1 x KL9010

Tab. 1: Prozessabbild

Busklemme	Position	Eingangsabbild	Ausgangsabbild	Größe
KL1501	5	%IB0...%IB5	%QB0...%QB5	6 Byte
KL3002	6	%IB6...%IB13	%QB6...%QB13	8 Byte
KL4002	7	%IB14...%IB21	%QB14...%QB21	8 Byte
KL6001	8	%IB22...%IB29	%QB22...%QB29	6 Byte
KL1012	1	%IX30.0...%IX30.1	-	2 Bit
KL1104	2	%IX30.1...%IX30.5	-	4 Bit
KL2012	3	-	%QX30.0...%IX30.1	2 Bit
KL2034	4	-	%QX30.2...%IX30.5	4 Bit
KL9010	9	-	-	-

5.8 Mapping der Busklemmen

Die genaue Belegung der byteorientierten Busklemmen entnehmen Sie bitte der Konfigurations-Anleitung zur jeweiligen Busklemme. Diese Dokumentation finden Sie im Internet unter <http://www.beckhoff.de>.

byteorientierte Busklemmen	bitorientierte Busklemmen
KL15x1	KL10xx, KL11xx, KL12xx, KL17xx, KM1xxx
KL25xx	KL20xx, KL21xx, KL22xx, KL26xx, KL27xx, KM2xxx
KL3xxx	
KL4xxx	
KL5xxx	
KL6xxx	
KL7xxx	
KL8xxx	
	KL9110, KL9160, KL9210, KL9260

5.9 Lokales Prozessabbild in der TwinCAT-Konfiguration

Die TwinCAT-Konfiguration (TwinCAT-CONFIG) ermöglicht das freie Mapping zwischen Feldbus, K-Bus und SPS-Variablen. Variablen können unabhängig von ihrer Adresse mit Hilfe des System Managers verknüpft werden.

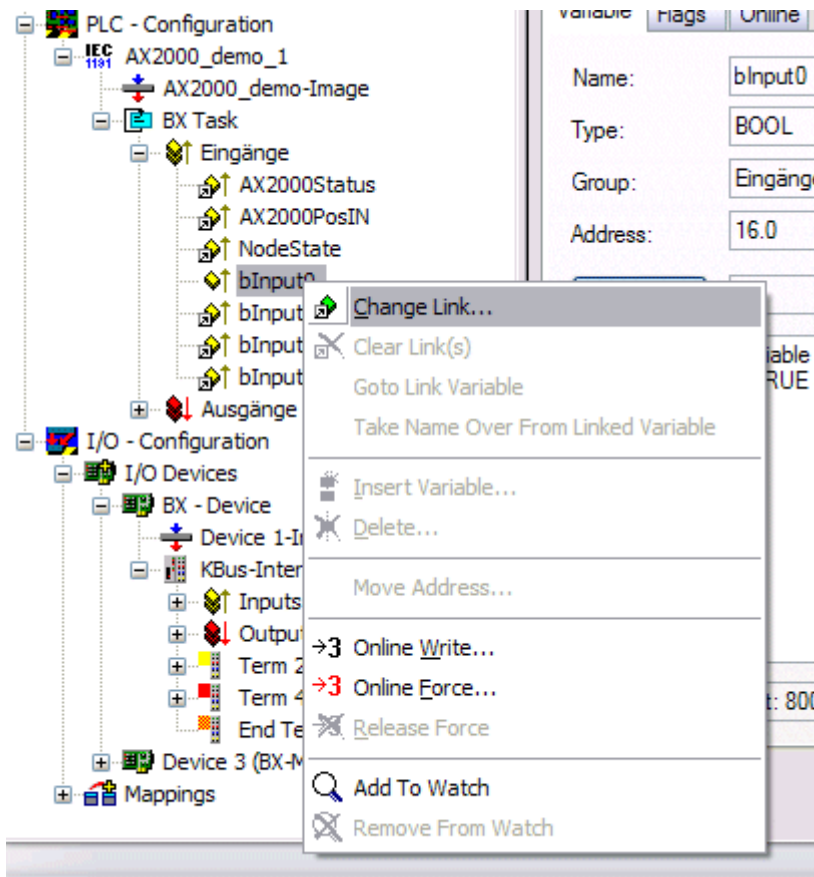


Abb. 60: Ändern der Verknüpfung von Variablen

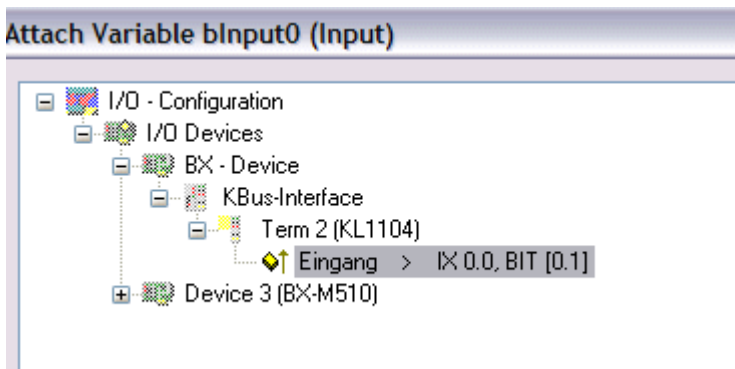


Abb. 61: Verknüpfen einer Variable mit einem Eingang

In der Default-Konfiguration liegen alle Busklemmen auf feste Adressen. Fügt man eine Busklemme ein, verschiebt sich eventuell der ganze Adressbereich. Mit der TwinCAT-Konfiguration hat man die Möglichkeit, seine lokierten Variablen, frei mit einer Busklemme zu verknüpfen. Man parametrisiert dies im System Manager und lädt dann diese erstellte Konfiguration auf den Busklemmen-Controller herunter (siehe [TwinCAT-Konfiguration](#) [▶ 45]). Ein Hochladen einer bestehenden TwinCAT-Konfiguration ist auch möglich.

5.10 Erzeugen eines Boot-Projekts

Für das Erzeugen des Boot-Projektes stehen

- auf den Busklemmen-Controllern der BX-Serie ca. 250 kByte Flash zu Verfügung.
- auf den Busklemmen-Controllern der BCxx50-Serie ca. 48 kByte Flash zu Verfügung.

PLC Control

Im TwinCAT PLC Control kann man, wenn man eingeloggt ist ein Boot-Projekt erzeugen.

- Öffnen eines PLC Projektes
- Auswahl des Zielsystem (oder Auswahl der seriellen Schnittstelle)
- Einloggen auf den BX/BCxx50
- Erzeugen des Boot-Projektes (Online\Create Bootproject)

Ist ein gültiges Boot-Projekt auf dem BX/BCxx50 leuchtet die LED PLC grün.

Bei den Busklemmen-Controllern der BX-Serie blinkt während der Erzeugung des Boot-Projektes die die PLC LED orange. Ist auf dem BX kein Boot-Projekt vorhanden, leuchtet die PLC LED orange.

Löschen eines Boot-Projektes

Sie können das Boot-Projekt auch vom Busklemmen-Controller löschen. Folgende Schritte sind einzuhalten:

- Öffnen des Projektes
- Einloggen auf den Busklemmen-Controller
- Löschen des Boot-Projektes (Online>Delete Boot Project)

Nach dem Löschen des Boot-Projektes ist die PLC LED orange.

● Übernahme des aktuellen Projektes als Boot-Projekt



Nach einem Online Change ist als Boot-Projekt noch immer das alte Projekt eingetragen. Soll das aktuelle Projekt (nach dem Online-Change) als Boot-Projekt übernommen werden, muss dieses neu erzeugt werden.

Umgehen eines Starten des Boot-Projekt*

Bei den Busklemmen-Controllern der BX-Serie kann beim Booten das Starten des Boot-Projekts durch Drücken des Navi-Schalters verhindert werden. Das Boot-Projekt ist damit nicht gelöscht und wird beim erneuten Booten des Busklemmen-Controller wieder gestartet.

* ab Version 0.85

5.11 Kommunikation zwischen TwinCAT und BX/BCxx50

Um von TwinCAT-Daten zum Busklemmen-Controller zu transportieren liegt es nahe, die Daten in einer Struktur anzulegen. Da die Datenhaltung auf beiden Systemen unterschiedlich ist sind folgende Hinweise zu beachten.

- Wenn zwei unterschiedliche Datentypen aufeinander folgen (zum Beispiel Byte und INT) wird die folgende Variable auf den nächsten graden Adress-Offset gelegt
- Boolsche Variablen sollten nie einzeln in eine Struktur gelegt werden, da sie so immer 1 Byte belegen würden. Boolsche Ausdrücke sollten immer in ein Byte oder Wort maskiert sein.

Beispiel 1: Eine Struktur auf den BX/BCxx50 und auf dem PC

Variable	Speicher des BX/BCxx50	Speicher des PC (TwinCAT)
Byte	%..B0	%..B0
INT (1)	%..B2	%..B1
INT (2)	%..B4	%..B3

Dadurch das hinter dem ersten Byte eine anderer Variable-Typ (INT) folgt ist dieser im BX/BCxx50 auf die nächste freie grade Adresse gelegt worden. Will man beide auf beiden Systemen die gleiche Datenstruktur haben, muss im PC-Projekt ein Dummy-Byte eingefügt werden (siehe Beispiel 2).

Beispiel 2: Eine Struktur auf den BX/BCxx50 und auf dem PC mit gleicher Speicherbelegung

Variable	Speicher des BX/BCxx50	Speicher des PC (TwinCAT)
Byte	%..B0	%..B0
Byte (Dummy)	%..B1 (nicht unbedingt notwendig, da dies das System selber macht, wenn diese Variabel nicht vorhanden ist)	%..B1
INT (1)	%..B2	%..B2
INT (2)	%..B4	%..B4

Daten Struktur

```
Type PB_Data
STRUCT
  wVar_1:WORD;
  iValue_1:INT;
  iValue_2:INT;
  iValue_3:INT;
END_STRUCT
END_TYPE
```

Anlegen einer Struktur in den Variablen

```
VAR_Global
  strData_Out AT %QB1000:PB_Data; (*PLC Variables *)
  bInput_01 AT %IX0.0:BOOL; (* Input from a terminal *)
END_VAR
```

Kleines Programmbeispiel

```
strData_Out.wVar_1.0:=bInput_01;
```

● Keine Real-Werte in gemischter Datenstruktur verwenden

i In einer gemischten Datenstruktur sollten keine Real-Werte enthalten sein. Wenn dies doch der Fall ist, muss zusätzlich im BX/BCxx50 oder im TwinCAT-Masterprojekt das High und Low Word vertauscht werden. Verwenden Sie besser ein Array von Real-Werten oder übertragen Sie die Real-Werte einzeln.

● Größere Feldbusdatenblöcke

i Sie können auch größere Feldbusdatenblöcke übertragen, um eine Reserve für Ihre Struktur zu haben. Nachteil: Diese Reserven werden dann mit jedem Feldbustelegramm übertragen, was eine Mehrbelastung der Feldbuskommunikation verursacht.

5.12 Up- und Download von Programmen

Der Busklemmen-Controller verfügt über einen Speicher für den Quell-Code. Hier können das Programm, die Task-Konfiguration und die Bibliotheken abgespeichert werden. Sollte Speicher für den Quell-Code nicht ausreichen, kann man auch nur den Quell-Code ohne Task-Konfiguration und die Bibliotheken ablegen. Dies benötigt wesentlich weniger Speicherplatz!

Allgemeine Einstellungen

Unter Bearbeiten/Optionen kann eingestellt werden wann der Quell-Code zum Zielsystem heruntergeladen werden soll. Öffnen Sie das Optionsmenü.

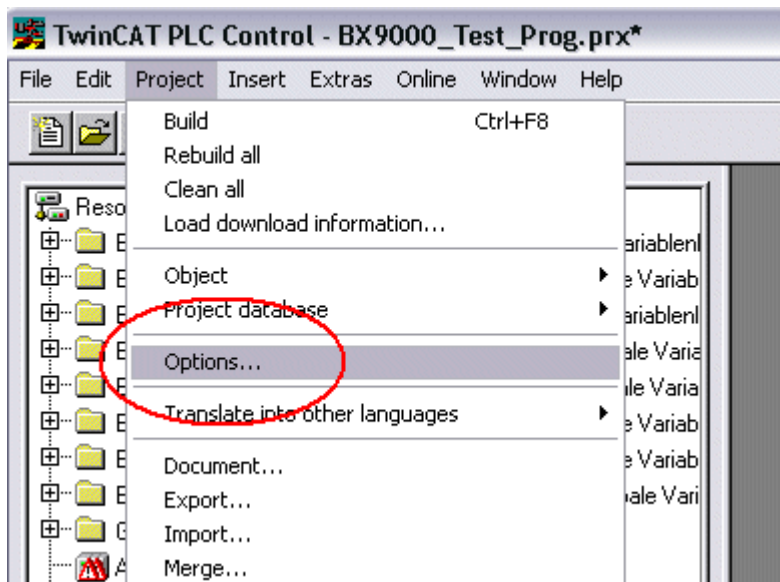


Abb. 62: Öffnen des Optionsmenüs

Wählen Sie nun den Source Download an.

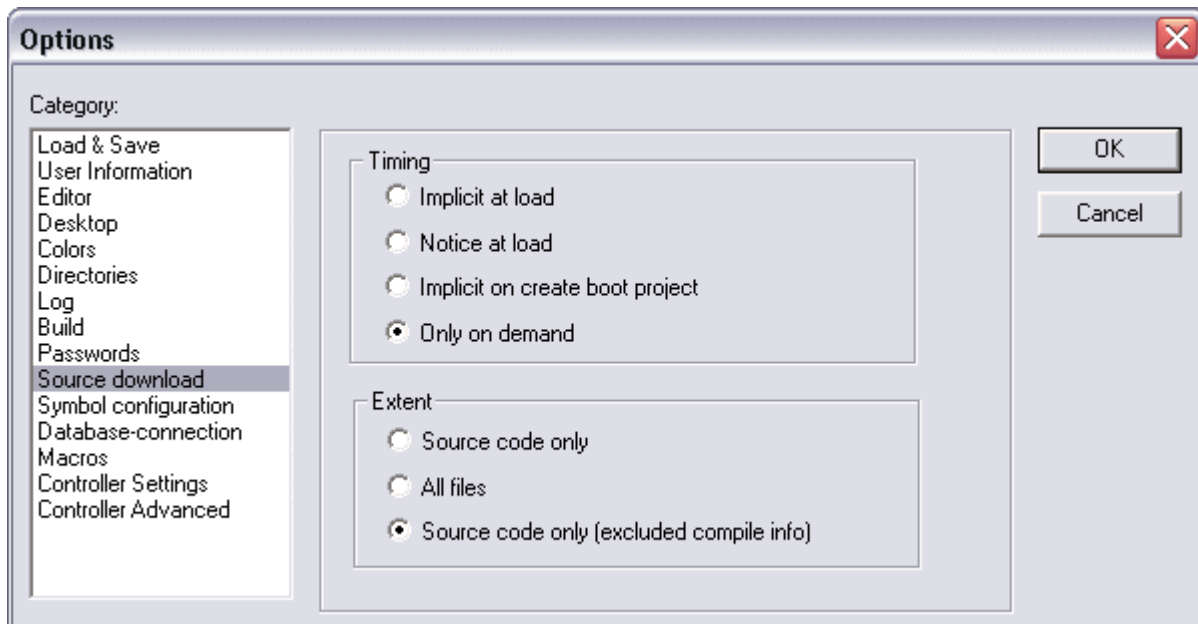


Abb. 63: Auswahl des Source Downloads

Hier könne Sie einstellen wann und was vom Source-Code zum Busklemmen-Controller runtergeladen werden soll.

Source code only: es wird das prx-File mit Informationen zum online Change übertragen. Damit ist ein einloggen per Online-Change möglich (die SPS stoppt nicht).

All files: Wie *Source code only* plus alle notwendigen Bibliotheken.

Source code only (excluded compile info): es wird nur das prx File übertragen. Ein einloggen ist nur möglich, wenn die SPS stoppt.

Welche Option Sie verwenden können hängt von der Größe Ihrer Projekte ab.

Download eines Programms

Der Quell-Code kann man auf Anforderung zum Zielsystem übertragen. Dafür muss man mit seinem Programm eingeloggt sein. Unter Online/Quell-Code Download kann jetzt der Programm-Code zum Busklemmen-Controller übertragen werden.

Online	Window	Help
Login		F11
Logout		F12
Download		
Run		F5
Stop		Shift+F8
Reset		
Reset All		
Toggle Breakpoint		
Breakpoint Dialog		F9
Step over		F10
Step in		F8
Single Cycle		Ctrl+F5
Write Values		
Force Values		Ctrl+F7
Release Force		F7
Write/Force-Dialog		Shift+F7
Write/Force-Dialog		
		Ctrl+Shift+F7
Show Call Stack...		
Display Flow Control		Ctrl+F11
Simulation Mode		
Communication Parameters...		
Sourcecode download		
Choose Run-Time System...		
Create Bootproject		
Create Bootproject (offline)		
Delete Bootproject		

Abb. 64: Download des Programm Codes

Nach einer kurzen Zeit öffnet sich ein Fenster, das den Vorschrift des Downloads anzeigt.

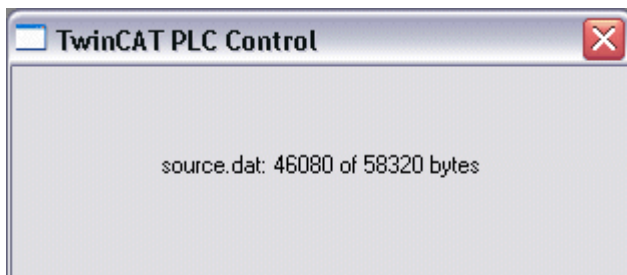


Abb. 65: Vorschrift des Downloads

Upload eines Programms

Um den Programm-Code wieder hoch zu laden, öffnen Sie im PLC Control ein neues File. Als nächstes klicken Sie auf die Schaltfläche PLC.

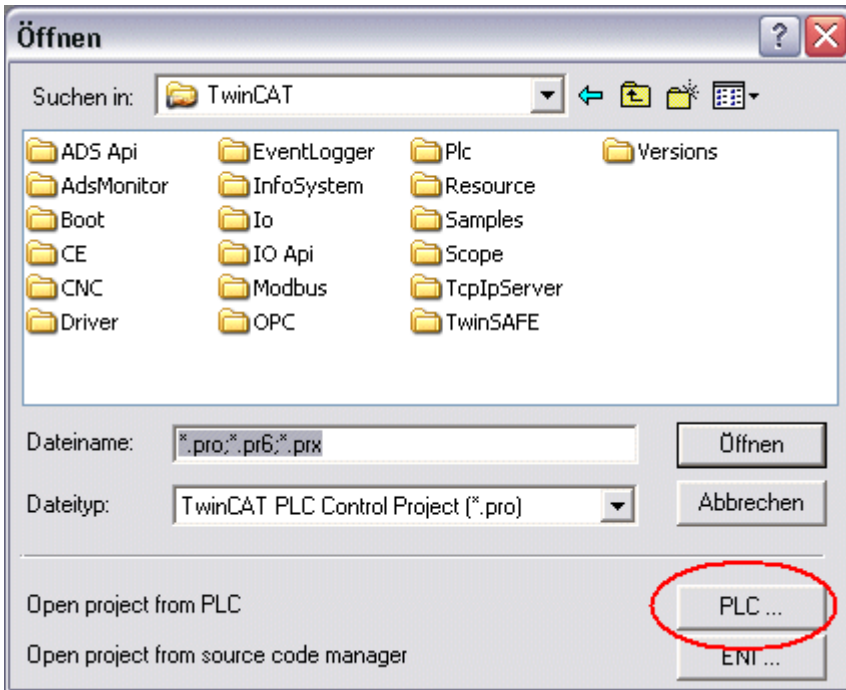


Abb. 66: Upload eines Programms

Wählen Sie den Datenübertragungsweg aus:

- *BCxx50 or BX via AMS*, wenn sie über den Feldbus mit dem Busklemmen-Controller verbunden sind oder
- *BCxx50 or BX via serial*, wenn sie über die serielle Schnittstelle mit dem Busklemmen-Controller verbunden sind.

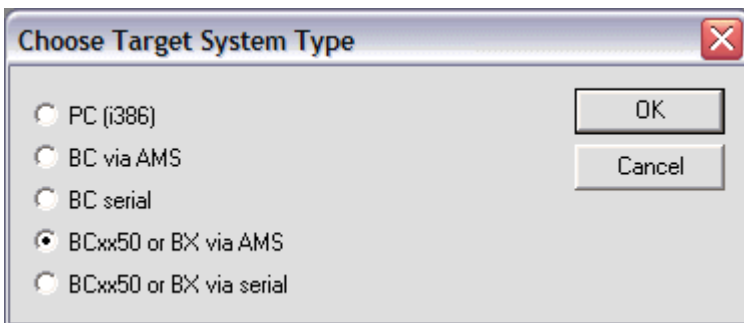


Abb. 67: Auswahl des Datenübertragungswegs

Als nächsten Schritt wählen Sie das Gerät aus und bestätigen Sie mit OK.

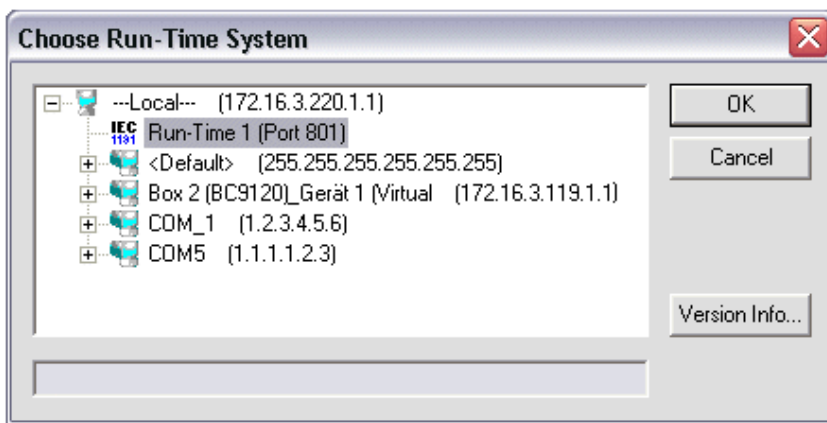


Abb. 68: Auswahl des Gerätes

Der Quell-Code wird nun hochgeladen.

Passwort

Mit einem Passwort können Sie Ihr Projekt schützen (im PLC Control Projekt/Optionen/Kennworte).




5.13 Bibliotheken

5.13.1 Bibliotheken - Übersicht

Für die Busklemmen Controller (Buskoppler mit SPS-Funktionalität) der Serie BC9050, BC9020, BC9120 stellt die Automatisierungs-Software TwinCAT verschiedene Bibliotheken (Libraries) zur Verfügung (siehe [Beckhoff Information System](#)).

Download

Die Bibliotheken sind auch in dieser Dokumentation enthalten. Klicken Sie zum Extrahieren der Bibliotheken mit der linken Maustaste auf den Link und kopieren Sie die Bibliotheken in das Verzeichnis TwinCAT\PLC\LIB.

- Standard
-  TcSystemBCxx50, Die TcSystemBCxx50 benötigt die TcBaseBCxx50 Bibliothek. (<https://infosys.beckhoff.com/content/1031/bc9xx0/Resources/zip/3740875659.zip>)
-  TcBaseBCxx50 (<https://infosys.beckhoff.com/content/1031/bc9xx0/Resources/zip/3740877835.zip>)
-  ChrAscBX.lbx (<https://infosys.beckhoff.com/content/1031/bc9xx0/Resources/zip/3740880011.zip>)
- [TcBaseBX9000.lbx](#) [[► 89](#)]

i Verwenden Sie die zur Firmware passende Bibliothek

Benutzen Sie zusammen mit der neusten BC-Firmware stets auch die neusten Bibliotheken. Sollten Sie die Firmware Ihres Busklemmen-Controller aktualisieren, aktualisieren Sie auch die Bibliotheken. Dazu kopieren Sie die neuen Bibliotheken ins LIB-Verzeichnis, entfernen Sie aus Ihrem Projekt und fügen sie dort erneut ein.

TcSystemBCxx50

ADS	Version	Firmware				
		BC9050	BC9020	BC9120	-	-
ADSREAD		B0	B0	B1	-	-
ADSWRITE		B0	B0	B1	-	-
ADSRDWRT		B0	B0	B1	-	-
ADSWRTCTL		B0	B0	B1	-	-
ADSRDSTATE		B0	B0	B1	-	-
ADSRDDEVINFO		B0	B0	B1	-	-

Bit Functions	Version	Firmware				
		BC9050	BC9020	BC9120	-	-
CLEARBIT32		B0	B0	B1	-	-
CSETBIT32		B0	B0	B1	-	-
GETBIT32		B0	B0	B1	-	-
SETBIT32		B0	B0	B1	-	-

Controller	Version	Firmware				
		BC9050	BC9020	BC9120	-	-
FB_BasicPID	-	B0	B0	B1	-	-
-	-	-	-	-	-	-

File Access	Version	Firmware				
		BC9050	BC9020	BC9120	-	-
FB_ReadFromFile		-	-	-	-	-
FB_WriteToFile		-	-	-	-	-
FB_ReadWriteFile		-	-	-	-	-

Memory Functions	Version	Firmware				
		BC9050	BC9020	BC9120	-	-
MEMCMP		B0	B0	B1	-	-
MEMCYP		B0	B0	B1	-	-
MEMMOVE		B0	B0	B1	-	-
MEMSET		B0	B0	B1	-	-

NOVRAM Functions	Version	Firmware				
		BC9050	BC9020	BC9120	-	-
-	-	-	-	-	-	-

SFC	Version	Firmware				
		BC9050	BC9020	BC9120	-	-
AnalyzeExpression		-	-	-	-	-
AppendErrorString		-	-	-	-	-
SFCActionControl		-	-	-	-	-

System / Time / TBus	Version	Firmware				
		BC9050	BC9020	BC9120	-	-
DRAND		B0	B0	B1	-	-
SYSTEMTIME_TO_DT		B0	B0	B1	-	-
DT_TO_SYSTEMTIME		B0	B0	B1	-	-
GetSysTick		B0	B0	B1	-	-
PresetSysTick		B0	B0	B1	-	-
Reboot		B0	B0	B1	-	-


Debug	Version	Firmware				
		BC9050	BC9020	BC9120	-	-
F_ReadDebugTimer		B0	B0	B1	-	-
F_StartDebugTimer		B0	B0	B1	-	-

5.13.2 Bibliotheken für BX9000, BC9020, BC9050, BC9120 - Übersicht

Spezielle Funktionsbausteine für die Busklemmen-Controller BX9000, BC9050, BC9120 und BC9020.

Download

Zum Download der Bibliotheken klicken Sie bitte mit der linken Maustaste auf das Diskettensymbol. Kopieren Sie die Bibliotheken dann in das Verzeichnis TwinCAT\PLC\LIB.

 TcBaseBX9000
 (<https://infosys.beckhoff.com/content/1031/bc9xx0/Resources/zip/3207318539.zip>)



Verwenden Sie die zur Firmware passende Bibliothek

Für die neuste Firmware ist auch die neuste Bibliothek zu benutzen. Sollten Sie Ihren BX-Controller Updaten, tauschen Sie auch die Bibliotheken. Kopieren Sie die Bibliotheken in das Verzeichnis TwinCAT\PLC\LIB. Entfernen Sie Bibliotheken dann aus Ihrem Projekt und fügen Sie sie erneut wieder neu hinzu.

TcBaseBX9000

Services	Version	Firmware			
		BX9000	BC9020	BC9050	BC9120
FB_AddDnsServer [► 109]	30.05.06	1.12	B0	B0	B1
FB_GetHostByAddr [► 110]	30.05.06	1.12	B0	B0	B1
FB_GetHostByName [► 111]	30.05.06	1.12	B0	B0	B1
FB_GetNetworkConfig [► 112]	30.05.06	1.12	B0	B0	B1
FB_SetTargetName [► 113]	30.05.06	1.12	B0	B0	B1

SMTP	Version	Firmware			
		BX9000	BC9020	BC9050	BC9120
FB_SMTP [► 106]	30.05.06	1.12	B0	B0	B1

SNTP	Version	Firmware			
		BX9000	BC9020	BC9050	BC9120
FB_SNTP [► 108]	30.05.06	1.12	B0	B0	B1

IP-TCP/IP-UDP	Version	Firmware			
		BX9000	BC9020	BC9050	BC9120
FB_AddMultiRoute [► 91]	26.09.06	1.14	-	-	-
FB_DelMultiRoute [► 91]	26.09.06	1.14	-	-	-
FB_IpClose [► 91]	26.09.06	1.14	B0	B0	B1
FB_IpEndSession [► 91]	26.09.06	1.14	B0	B0	B1
FB_IpOpen [► 91]	26.09.06	1.14	B0	B0	B1
FB_IpReceive [► 91]	26.09.06	1.14	B0	B0	B1
FB_IpSend [► 91]	26.09.06	1.14	B0	B0	B1
FB_IpStartSession [► 91]	26.09.06	1.14	B0	B0	B1

ModbusTCP	Version	Firmware			
		BX9000	BC9020	BC9050	BC9120
FB_MBClose [► 102]	26.09.06	1.14	B0	B0	B1
FB_MBConnect [► 102]	26.09.06	1.14	B0	B0	B1
FB_MBGenericReq [► 102]	26.09.06	1.14	B0	B0	B1

5.13.3 TcBaseBCxx50

5.13.3.1 System Task Info

```
VAR_GLOBAL
SystemTaskInfo : SYSTEMTASKINFOTYPE;
END_VAR
```

Systemflags sind implizit deklarierte Variablen. Mit der Eingabehilfe finden Sie unter Systemvariablen eine Variable SystemTaskInfoArr. Diese Variable ist ein Feld von vier Strukturen des Typs SYTEMTASKINFOTYPE [► 85]. Die Strukturdefinition ist in der System-Bibliothek zu finden. Der Index in dieses Feld ist die Task-Id.

Entwicklungsumgebung	Zielplattform	Einzubindende SPS Bibliotheken
TwinCAT v2.9.0	BCxx50, BC9x20 Controller	TcBaseBCxx50.lbx

5.13.3.2 System Task Info Type

```
TYPE SYSTEMTASKINFOTYPE
STRUCT
active          :      BOOL;
taskName       :      STRING(16);
firstCycle     :      BOOL;
cycleTimeExceeded :    BOOL;
cycleTime      :      UDINT;
lastExecTime  :      UDINT;
priority       :      BYTE;
cycleCount     :      UDINT;
END_STRUCT
END_TYPE
```

Legende

- active: Diese Variable zeigt an, ob die Task aktiv ist.
- taskName: Der Taskname.
- firstCycle: Diese Variable hat im ersten Zyklus der SPS-Task den Wert: TRUE.
- cycleTimeExceeded: In dieser Variablen wird ein Überschreiten der eingestellten Taskzykluszeit gemeldet.
- cycleTime :Eingestellte Taskzykluszeit in Vielfachen von 100 ns.

lastExecTime: Benötigte Zykluszeit für den letzten Zyklus in Vielfachen von 100 ns.
 priority: Eingestellte Priorität der Task.
 cycleCount: Zykluszähler.

Entwicklungsumgebung	Zielformat	Einzubindende SPS Bibliotheken
TwinCAT v2.9.0	BCxx50, BC9x20 Controller	TcBaseBCxx50.lbx

5.13.3.3 System Info

```
VAR_GLOBAL
    SystemInfo : SYSTEMINFOTYPE;
END_VAR
```

Systemflags sind implizit deklarierte Variablen. Mit der Eingabehilfe finden Sie unter Systemvariablen eine Variable Systeminfo. Der Typ `SYSTEMINFOTYPE` [▶ 86] ist in der System-Bibliothek deklariert. Um auf die Variable zugreifen zu können muss die System-Bibliothek in das Projekt eingebunden werden.

Entwicklungsumgebung	Zielformat	Einzubindende SPS Bibliotheken
TwinCAT v2.9.0	BCxx50, BC9x20 Controller	TcBaseBCxx50.lbx

5.13.3.4 System Info Type

```
TYPE SYSTEMINFOTYPE
STRUCT
    runTimeNo : BYTE;
    projectName : STRING(32);
    numberOfTasks : BYTE;
    onlineChangeCount : UINT;
    bootDataFlags : BYTE;
    systemStateFlags : WORD;
END_STRUCT
END_TYPE
```

Legende

runTimeNo: Gibt die Nummer des Laufzeitsystems (1) an.
 projectName: Name des Projekts als STRING.
 numberOfTasks: Anzahl der im Laufzeitsystem befindlichen Tasks (max. 1).
 onlineChangeCount: Anzahl der seit dem letzten Komplettdownload gemachten Online-Änderungen.
 bootDataFlags: Reserviert
 systemStateFlags: Reserviert.

Entwicklungsumgebung	Zielformat	Einzubindende SPS Bibliotheken
TwinCAT v2.9.0	BCxx50, BC9x20 Controller	TcBaseBCxx50.lbx

5.13.3.5 ADS

5.13.3.5.1 Lokale ADS-Port Nummern - Übersicht

Port Nummer	Beschreibung
100 [▶ 87] dez	Lesen und schreiben von Registern und Tabellen aus den Koppler und den komplexen Busklemmen
150 [▶ 87] dez	Lesen und schreiben der RTC (Echt-Zeit-Uhr)
153 [▶ 87] dez	SSB - auslesen der Emergency-Nachricht
800 [▶ 87] dez	Lokales Prozessabbild der PLC, siehe auch Port 801
801 [▶ 87] dez	Lokales Prozessabbild der PLC, siehe auch Port 800
0x1000 + Node ID [▶ 87]	SSB - SDO Kommunikation mit CANopen-Knoten (Slave Nummer)

5.13.3.5.2 ADS-Dienste

Lokales Prozessabbild PLC Task 1 Port 800/801

Im lokalen Prozessabbild können Daten gelesen und geschrieben werden. Sollten Ausgänge geschrieben werden muss darauf geachtet werden, das diese von der lokalen SPS nicht verwendet werden, da die lokale Steuerung diese Werte überschreibt. Die Daten sind nicht an einen Watchdog gebunden und müssen und dürfen daher nicht für Ausgänge verwendet werden, die im Fehlerfall ausgeschaltet werden müssen.

Index Group	Bedeutung	Index Offset (Wertebereich)
0xF020	Input - Eingänge	0...2047
0xF021	Input - Eingänge Bit	0...16376
0xF030	Output - Ausgänge	0...2047
0xF031	Output - Ausgänge Bit	0...16376
0x4020	Merker	0...4095
0x4021	Merker Bit	0...32760

Dienste des ADS

AdsServerAdsState

Datentyp (only Read)	Bedeutung
String	Start - die lokale PLC läuft Stop - die lokale PLC ist im Stop

AdsServerDeviceState

Datentyp (only Read)	Bedeutung
INT	0 - Start - die lokale PLC läuft 1 - Stop - die lokale PLC ist im Stop

AdsServerType

Datentyp (only Read)	Bedeutung
String	BX PLC Server

ADSWriteControl

Datentyp (write only)	Bedeutung
NetID	Net ID des Ethernet Controllers*
Port	800
ADSSTATE	5 - RUN / 6 - STOP
DEVSTATE	0
LEN	0
SRCADDR	0
WRITE	positive Flanke startet den Baustein
TMOUT	zum Beispiel: t#1000 ms

* BC9050, BC9020, BC9120, BX9000

Register Zugriff Port 100

Die ADS-Portnummer ist bei den Busklemmen-Controllern der BX-Serie und den BCxx50/xx20 für die Register-Kommunikation fest vorgegeben und beträgt 100.

Index Group	Index Offset (Wertebereich)		Bedeutung
	Hi-Word	Lo-Word	
0 [READ ONLY]	0...127	0...255	Register des Buskopplers Hi-Word Tabellennummer des Buskopplers Lo-Word Registernummer der Tabelle
1...255	0...3	1...255	Register der Busklemmen Hi-Word Kanalnummer Lo-Word Registernummer der Busklemme

● **Minimaler Time-Out**

i Beachten Sie beim Lesen der Register, dass der Time-Out beim ADS-Baustein auf eine Zeit größer 1 sec eingestellt wird.

● **Passwort setzen**

i Beachten Sie beim Schreiben auf die Register, dass das Passwort gesetzt wird (siehe Dokumentation zur entsprechenden Busklemme).

5.13.3.6 BX Debug-Funktion

Diese Funktionen können benutzt werden um Befehls-Ausführungszeiten in einem PLC-Projekt zu messen. Die Einheit ist ein Tick. Ein Tick entspricht 5.12 µs.

Funktion Start Debug Timer

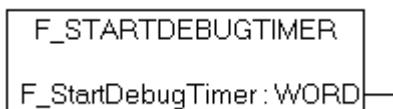


Abb. 69: Funktionsbaustein F_STARTDEBUGTIMER

Der Aufruf dieser Funktion startet den Timer. Der Rückgabe Wert ist "0".

Funktion Read Debug Timer

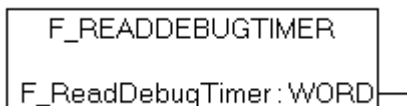


Abb. 70: Funktionsbaustein F_READDEBUGTIMER

Mit dieser Funktion wird der Timer-Wert gelesen. Der Rückgabewert muss mit 5.12 µs multipliziert werden.

Beispiel

```
VAR
    Timer_BX      :WORD;
    i              :INT;
END_VAR
```

Programm

```
F_STARTDEBUGTIMER();
For i:=0 to 1000 do
;
END_FOR
Timer_BX:=F_READDEBUGTIMER();
```



5.13.4 TcBaseBX9000

5.13.4.1 Bibliotheken für BC9020, BC9050, BC9120 - Übersicht

Spezielle Funktionsbausteine für die Busklemmen-Controller BC9050, BC9120 und BC9020.

Download

Zum Download der Bibliotheken klicken Sie bitte mit der linken Maustasten auf den Link. Kopieren Sie die Bibliotheken dann in das Verzeichnis TwinCAT\PLC\LIB.

 TcBaseBX9000
 (<https://infosys.beckhoff.com/content/1031/bc9xx0/Resources/zip/3740882187.zip>)

i Verwenden Sie die zur Firmware passende Bibliothek

Für die neuste Firmware ist auch die neuste Bibliothek zu benutzen. Sollten Sie Ihren BX-Controller Updaten, tauschen Sie auch die Bibliotheken.

Kopieren Sie die Bibliotheken in das Verzeichnis TwinCAT\PLC\LIB. Entfernen Sie Bibliotheken dann aus Ihrem Projekt und fügen Sie sie erneut wieder neu hinzu.

TcBaseBX9000

Services	Version	Firmware		
		BC9020	BC9050	BC9120
FB_AddDnsServer	30.05.06	B0	B0	B1
FB_GetHostByAddr	30.05.06	B0	B0	B1
FB_GetHostByName	30.05.06	B0	B0	B1
FB_GetNetworkConfig	30.05.06	B0	B0	B1
FB_SetTargetName	30.05.06	B0	B0	B1

SMTP	Version	Firmware		
		BC9020	BC9050	BC9120
FB_SMTP	30.05.06	B0	B0	B1

SNTP	Version	Firmware		
		BC9020	BC9050	BC9120
FB_SNTP	30.05.06	B0	B0	B1

IP-TCP/IP-UDP	Version	Firmware		
		BC9020	BC9050	BC9120
FB_AddMultiRoute	26.09.06	-	-	-
FB_DelMultiRoute	26.09.06	-	-	-
FB_IpClose	26.09.06	B0	B0	B1
FB_IpEndSession	26.09.06	B0	B0	B1
FB_IpOpen	26.09.06	B0	B0	B1
FB_IpReceive	26.09.06	B0	B0	B1
FB_IpSend	26.09.06	B0	B0	B1
FB_IpStartSession	26.09.06	B0	B0	B1

ModbusTCP	Version	Firmware		
		BC9020	BC9050	BC9120
FB_MBClose	26.09.06	B0	B0	B1
FB_MBCConnect	26.09.06	B0	B0	B1
FB_MBGenericReq	26.09.06	B0	B0	B1

5.13.4.2 Socket Interface

5.13.4.2.1 Übersicht: Socket-Interface

Das Socket-Interface dient dazu, beliebige Ethernet Telegramme zu empfangen oder auch von der Steuerung zu senden. So können beliebige Protokolle auf SPS Ebene in der IEC 61131-3 programmiert werden.

Unterstützte Ethernet Protokolle

Type	STREAM	DGRAM	RAW
IP	n.i.	n.i.	n.i.
ICMP	n.i.	n.i.	n.i.
IGMP	n.i.	n.i.	n.i.
TCP	Implementiert*	n.i.	n.i.
UDP	n.i.	Implementiert*	n.i.
RAW	n.i.	n.i.	n.i.

n.i.: nicht implementiert

Arbeitsweise einer Socket-Verbindung

Bevor man ein Socket öffnen kann, muss der Steuerung für diese Art von Verbindung Ressourcen zur Verfügung gestellt werden. Dies geschieht durch das Starten einer Session. Hier gibt man an wie und was man tun möchte. Danach kann die Session genutzt werden, um Ethernet Telegramme zu verschicken oder zu empfangen. Die implementierten Protokolle sind der obenstehenden Tabelle zu entnehmen.

Client/Server-Beziehung

Unter Client versteht man den Teilnehmer, der eine Verbindung aufbauen möchte und den aktiven Teil eines Verbindungsaufbaus initialisiert. Der Server ist erst einmal passiv und wartet auf eine Client Anfrage. Erst wenn ein Client eine Verbindung aufbaut, wird auch der Server aktiv. Ist eine Verbindung zwischen Client und Server aufgebaut, können beide Teilnehmer beliebig Daten senden und auch empfangen.

* zur Tabelle [Unterstützte Ethernet Protokolle](#) [► 90]

Entwicklungsumgebung	Zielplattform	Einzubindende SPS Bibliotheken
TwinCAT v2.10.0 und höher	BC9050 (165) Firmware-Version ≥B0	TcBaseBX9000.lbx
TwinCAT v2.10.0 und höher	BC9020 (165) Firmware-Version ≥B0	TcBaseBX9000.lbx
TwinCAT v2.10.0 und höher	BC9120 (165) Firmware-Version ≥B1	TcBaseBX9000.lbx
TwinCAT v2.10.0 und höher	BC9191 Firmware-Version ≥3.1	TcBaseBX9000.lbx
TwinCAT v2.10.0 und höher	BX9000 (165) Firmware-Version ≥1.14	TcBaseBX9000.lbx

5.13.4.2.2 IP-Bausteinübersicht

IP-TCP/IP-UDP	Firmware	Beschreibung
FB_IpStartSession [▶ 91]	1.14	Öffnen einer Session
FB_IpEndSession [▶ 92]	1.14	Schließen einer Session
FB_IpOpen [▶ 93]	1.14	Öffnen eine TCP/IP-Verbindung (wird für eine UDP-Kommunikation nicht gebraucht)
FB_IpClose [▶ 93]	1.14	Schließen einer TCP/IP-Verbindung (wird für eine UDP-Kommunikation nicht gebraucht)
FB_IpReceive [▶ 95]	1.14	Empfang von TCP- oder UDP-Telegrammen
FB_IpSend [▶ 96]	1.14	Senden von TCP- oder UDP-Telegrammen

Multicast-Bausteine	Firmware	Beschreibung
FB_AddMultiRoute [▶ 108]	1.14	Eine Muticast-Adresse anlegen
FB_DelMultiRoute [▶ 108]	1.14	Eine Muticast-Adresse löschen

FB_IpStartSession

Der Baustein reserviert Ressourcen auf der Steuerung für die Ethernet Kommunikation. Mit *bStart* wird der Baustein aktiv. Solange der Baustein arbeitet, ist das **bBusy** gesetzt. **iDevice** ist immer mit Null zu belegen. **iPort** ist für die lokale TCP oder UDP Port Nummer. **iMaxConnection** gibt die Anzahl der maximal möglichen Verbindungen an (maximal 3).

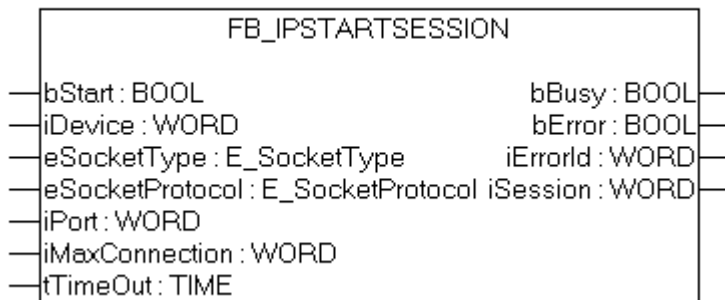


Abb. 71: Funktionsbaustein FB_IPSTARTSESSION

INPUT

```

VAR_INPUT
  bStart      : BOOL;
  iDevice     : WORD;
  eSocketType : E_SocketType;
  eSocketProtocol : E_SocketProtocol;
  iPort       : WORD;
  iMayConnection : WORD;
  tTimeout    : TIME;
END_VAR
    
```

bStart: Eine steigende Flanke aktiviert den Baustein.

iDevice: immer "0"

eSocketType: bei TCP/IP ist "SOCK_STREAM" zu benutzen. Beachten Sie, dass die Daten als Byte Stream abgelegt werden. Sie sollten möglichst die Länge der empfangenden Daten kennen oder ein Protokoll mit einer Start- und Endkennung verwenden, damit Sie im Daten-Stream eindeutig Anfang und Ende erkennen können. Bei UDP/IP ist "SOCK_DGRAM" einzustellen. Dabei wird ein UDP-Frame immer komplett in einen Speicher eingetragen. Es stehen 4 Speicher zur Verfügung. Liest das Anwenderprogramm die Daten nicht schnell genug aus dem Speicher der SPS aus, gehen weitere UDP-Frames verloren.

eSocketProtocol: für TCP/IP ist "IPPROTO_TCP" zu benutzen und für UDP/IP "IPPROTO_UDP".

iPort: Absender Port Nummer

iMaxConnection: Anzahl der maximal möglichen Verbindungen (max. ist 3)

tTimeout: Zeit nach der abgebrochen werden soll.

OUTPUT

```
VAR_OUTPUT
  bBusy      : BOOL;
  bError     : BOOL;
  iErrorId   : WORD;
  iSession   : WORD;
END_VAR
```

bBusy: Dieser Ausgang bleibt solange auf TRUE, bis die Befehlsausführung abgeschlossen wurde.

bError: Dieser Ausgang wird auf TRUE geschaltet, wenn bei der Ausführung eines Befehls ein Fehler aufgetreten ist. Der befehlspezifische Fehlercode ist in *iErrorId* enthalten.

iErrorId: Enthält den befehlspezifischen Fehler-Code des zuletzt ausgeführten Befehls (siehe Tabelle).

iSession: Gibt die Session-Nummer an alle IP-Bausteine weiter, für die diese Verbindung angelegt wurde.

FB_IpEndSession

Der Baustein schließt eine geöffnete Session. Positive Flanke von *bStart* schließt die Session und gibt Ressourcen frei.

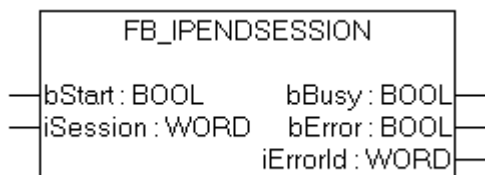


Abb. 72: Funktionsbaustein FB_IPENDSESSION

INPUT

```
VAR_INPUT
  bStart      : BOOL;
  iSession    : WORD;
END_VAR
```

bStart: Eine steigende Flanke aktiviert den Baustein.

iSession: wird mit der Session Nummer aus dem Baustein FB_IpStartSession.

OUTPUT

```
VAR_OUTPUT
  bBusy      : BOOL;
  bError     : BOOL;
  iErrorId   : WORD;
END_VAR
```

bBusy: Dieser Ausgang bleibt solange auf TRUE, bis die Befehlsausführung abgeschlossen wurde.

bError: Dieser Ausgang wird auf TRUE geschaltet, wenn bei der Ausführung eines Befehls ein Fehler aufgetreten ist. Der befehlspezifische Fehlercode ist in *iErrorId* enthalten.

iErrorId: Enthält den befehlspezifischen Fehlercode des zuletzt ausgeführten Befehls (siehe Tabelle).

FB_IpOpen

Der Baustein ist notwendig für das Öffnen einer TCP/IP Verbindung aus der SPS Steuerung heraus. In diesem Fall ist die Steuerung der Client, der aktive eine Verbindung zu einem TCP/IP Server aufbaut. Mit einer positiven Flanke von **bStart** wird eine Verbindung zu einem Server mit der IP-Adresse aus **sRemoteIPAddr** aufgebaut. Die Empfänger Port Nummer wurde durch das Starten der Session vorgegeben (siehe [FB_IpStartSession \[▶ 91\]](#)). Die Absender Port Nummer wird von der Steuerung vergeben und kann aus **iPortNo** ausgelesen werden. Die Absender Port Nummer wird für das Senden (siehe [FB_IpSend \[▶ 96\]](#)) gebraucht. Das **bBusy** wird solange gesetzt wie der Baustein aktiv ist. Wird das **bBusy** zurückgesetzt und ist das **bError** FALSE ist der Aufbau der TCP/IP Verbindung erfolgreich abgeschlossen und es können Daten gesendet oder empfangen werden.



Abb. 73: Funktionsbaustein FB_IPOPEN

INPUT

```
VAR_INPUT
  bStart      : BOOL;
  iSession    : WORD;
  sRemoteIPAddr : STRING(15);
END_VAR
```

bStart: Eine steigende Flanke aktiviert den Baustein.

iSession: wird mit der Session Nummer aus dem Baustein [FB_IpStartSession](#) verbunden.

OUTPUT

```
VAR_OUTPUT
  bBusy      : BOOL;
  bError     : BOOL;
  iErrorId   : WORD;
  iPortNo    : WORD;
END_VAR
```

bBusy: Dieser Ausgang bleibt solange auf TRUE, bis die Befehlsausführung abgeschlossen wurde.

bError: Dieser Ausgang wird auf TRUE geschaltet, wenn bei der Ausführung eines Befehls ein Fehler aufgetreten ist. Der befehlspezifische Fehlercode ist in *iErrorId* enthalten.

iErrorId: Enthält den befehlspezifischen Fehlercode des zuletzt ausgeführten Befehls (siehe Tabelle).

iPortNo: TCP Port Nummer, die beim Öffnen der TCP/IP Verbindung vergeben wurde (Lokale Port Nummer).

FB_IpClose

Der Baustein ist notwendig für das Schließen einer TCP/IP Verbindung aus der SPS Steuerung heraus. Mit einer positiven Flanke von **bStart** wird eine Verbindung mit der IP-Adresse aus **sRemoteIPAddr** abgebaut. Der Baustein sendet ein FIN und wartet auf die Bestätigung des Verbindungsabbaus. Hierfür müssen die Gegenstelle und die Verbindung existieren und funktionieren. Ist das **bResetConnection** gesetzt sendet der Baustein, dass die Verbindung beendet wurde, wartet aber nicht auf eine Bestätigung des anderen Teilnehmers. Das **bBusy** wird solange gesetzt wie der Baustein aktiv ist. Wird das **bBusy** zurückgesetzt und ist das **bError** FALSE ist der Abbau der TCP/IP Verbindung erfolgreich abgeschlossen.

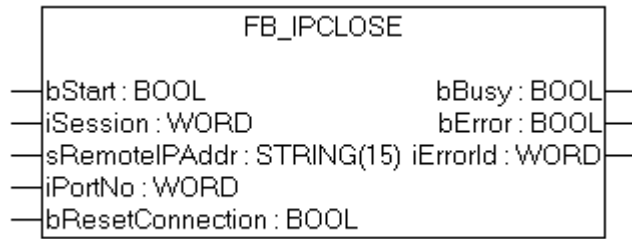


Abb. 74: Funktionsbaustein FB_IPCLOSE

INPUT

```
VAR_INPUT
  bStart      : BOOL;
  iSession    : WORD;
  sRemoteIPAddr : STRING(15);
  iPortNo     : WORD;
  bResetConnection : BOOL;
END_VAR
```

bStart: Eine steigende Flanke aktiviert den Baustein.

iSession: wird mit der Session Nummer aus dem Baustein FB_IpStartSession verbunden.

sRemoteIPAddr: IP-Adresse des Teilnehmers, mit dem die Verbindung abgebaut werden soll.

iPortNo: Port Nummer des Teilnehmers, mit dem die Verbindung abgebaut werden soll.

bResetConnection: FALSE es wird ein FIN gesendet - TRUE die TCP/IP Verbindung wird geschlossen ohne das auf ein Quittung der Gegenstelle gewartet wird, es ist in beiden Fällen ein erneutes Open notwendig um die Verbindung wieder aufzubauen.

OUTPUT

```
VAR_OUTPUT
  bBusy      : BOOL;
  bError     : BOOL;
  iErrorId   : WORD;
END_VAR
```

bBusy: Dieser Ausgang bleibt solange auf TRUE, bis die Befehlsausführung abgeschlossen wurde.

bError: Dieser Ausgang wird auf TRUE geschaltet, wenn bei der Ausführung eines Befehls ein Fehler aufgetreten ist. Der befehlspezifische Fehlercode ist in *iErrorId* enthalten.

iErrorId: Enthält den befehlspezifischen Fehlercode des zuletzt ausgeführten Befehls (siehe Tabelle).

FB_IpReceive

Der Baustein *FB_IpReceive* ermöglicht das Empfangen von UDP- oder TCP-Telegrammen. Welche der beiden Verbindungen verwendet wird, wurde im [FB_IpStartSession](#) [▶ 91] festgelegt.

Daten die empfangen werden, müssen in einer SPS-Variabel abgelegt werden. Dafür müssen Sie einen Pointer auf **pBuffAddr** legen und die Größe der Variable in *cbBuffLen* eintragen. Mit einer positiven Flanke von **bValid** wird angezeigt, dass Daten im Speicher liegen bzw. die Daten ihrer Variable jetzt gültig sind. Mit einer positiven Flanke von **bClear** ist der Baustein wieder bereit Daten zu empfangen, bzw. liegen noch Daten im Puffer werden diese als nächstes in die Variable kopiert.

Achten Sie besonders bei TCP darauf, wie viele Daten empfangen wurden bzw. wie viele Daten noch im Puffer liegen. Bei TCP/IP werden die Daten als "Stream" abgelegt, d.h. es gibt hier kein Anfang und kein Ende. Im Gegensatz dazu wird bei UDP immer der Inhalt eines UDP Frames in einen eigenen Puffer abgelegt.

Es können 4 UDP-Telegramme gepuffert werden, weitere UDP Telegramme gehen verloren. **cbReceive** zeigt Ihnen die Anzahl der Daten in Bytes an, die in Ihre Variable kopiert wurden. Sind mehr Daten in Puffer als ausgelesen wurden, steht die Anzahl, der noch verbleibenden Daten in **cbBytesInStream**.

sReceiveIPAddr gibt Ihnen die IP-Adresse des Teilnehmers an, der Daten zur Beckhoff-Steuerung gesendet hat und die entsprechende Port Nummer **iReceivePortNo**. Beide Variablen werden mit **bClear** wieder gelöscht. Sie können **sReceiveIPAddr** und *iReceivePortNo* dafür nutzen um Daten an den Teilnehmer wieder zurück zu senden mit dem Baustein [FB_IpSend](#) [▶ 96].

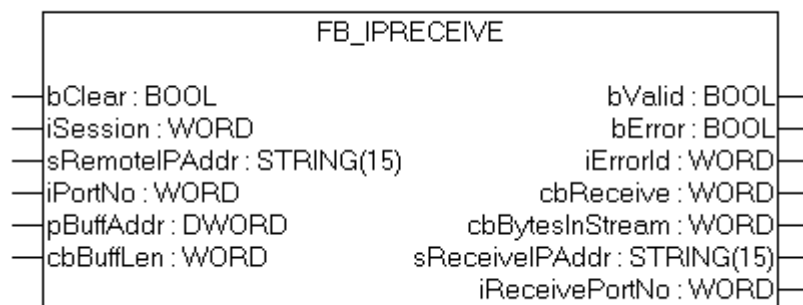


Abb. 75: Funktionsbaustein FB_IPRECEIVE

INPUT

```
VAR_INPUT
  bClear      : BOOL;
  iSession    : WORD;
  sRemoteIPAddr : STRING(15);
  iPortNo     : WORD;
  pBuffAddr   : DWORD;
  cbBuffLen   : WORD;
END_VAR
```

bClear: Eine steigende Flanke löscht den Speicher und der Baustein ist wieder bereit Daten zu empfangen.

iSession: wird mit *iSession* aus dem Baustein [FB_StartSession](#) [▶ 91] verbunden.

sRemoteIPAddr: kann als Filter verwendet werden um nur eine spezielle IP Adresse zuzulassen.

iPortNo: kann als Filter verwendet werden um nur einen speziellen Port zuzulassen.

pBuffAddr: Mit **ADR** wird hier der Pointer auf die Variable übergeben, wohin die Daten kopiert werden sollen, die empfangen wurden.

cbBuffLen: Größe der Variable, kann mit **SIZEOF** ermittelt werden.

OUTPUT

```
VAR_OUTPUT
  bValid      : BOOL;
  bError      : BOOL;
  iErrorId    : WORD;
  cbReceive   : WORD;
  cbBytesInStream : WORD;
  sReceiveIPAddr : STRING(15);
  iReceivePortNo : WORD;
END_VAR
```

bValid: Ein positiver Flankenwechsel zeigt an das neue Daten Empfangen wurden. Diese Daten stehen nun in der Variable zur Verfügung, die über den Pointer **pBuffAddr** verbunden wurde.

bError: Dieser Ausgang wird auf TRUE geschaltet, wenn bei der Ausführung eines Befehls ein Fehler aufgetreten ist. Der befehlspezifische Fehlercode ist in *iErrorId* enthalten.

iErrorId: Enthält den befehlspezifischen Fehlercode des zuletzt ausgeführten Befehls (siehe Tabelle).

cbReceive: Gibt die Anzahl an Bytes an die kopiert worden sind.

cbBytesInStream: Gibt die noch im Speicher befindlichen Daten an. Dies sollte immer Null sein. Ist der Wert >0, haben Sie die Variable die mit **pBuffAddr** verbunden wurde zu klein gewählt. Es muss dann noch einmal ausgelesen werden und den Rest der Daten zu erhalten.

sReceiveIPAddr: Zeigt Ihnen die IP-Adresse an von dem Teilnehmer, der die Daten geschickt hat. Wird mit positiver Flanke von **bClear** gelöscht.

iReceivePortNo: Zeigt Ihnen die Port-Nummer an von dem Teilnehmer, der die Daten geschickt hat. Wird mit positiver Flanke von **bClear** gelöscht.

FB_IpSend

Der Baustein sendet Daten über TCP oder UDP. Welche der Verbindung verwendet wird, ist in der [FB_IpStartSession \[▶ 91\]](#) festgelegt worden. An den **pBuffAddr** wird per "ADR" der Pointer auf die Variabel festgelegt, die Daten für das Senden beinhaltet. **cbBuffLen** gibt die Länge der Daten an. Mit **sRemoteIPAddr** wird die IP Adresse angegeben, an die die Daten gesendet werden. Mit **iPortNo** wird bei TCP dies mit der Port Nummer des [FB_IpOpen \[▶ 93\]](#) verknüpft. Bei UDP kann eine beliebige Portnummer verwendet werden. **iPortNo** ist die Absender Port Nummer. Mit einer positiven Flanke von **bStart** wird der Baustein aktiviert und die Daten werden versendet. Solange der Baustein aktiv ist, ist das **bBusy** auf TRUE. Sind die Daten versendet worden geht das **bBusy** auf FALSE und das **bError** bleibt ebenfalls auf FALSE. Bei einem Fehler wird das **bError** gesetzt.

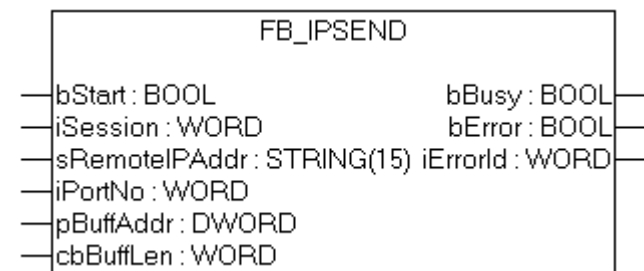


Abb. 76: Funktionsbaustein FB_IPSEND

INPUT

```
VAR_INPUT
  bStart      : BOOL;
  iSession    : WORD;
  sRemoteIPAddr : STRING(15);
  pBuffAddr   : DWORD;
  cbBuffLen   : WORD;
END_VAR
```

bStart: Eine steigende Flanke aktiviert den Baustein.

iSession: wird mit der Session Nummer aus dem Baustein `FB_IpStartSession` [► 91] verbunden.

sRemotelPAddr: IP-Adresse des Teilnehmers, an den die Daten gesendet werden sollen.

iPortNo: Absender Port Nummer. Bei TCP ist die Port Nummer aus den `FB_IpOpen` [► 93] Baustein zu verwenden, bei UDP kann eine beliebige Port Nummer verwendet werden.

pBuffAddr: Pointer auf die Daten die gesendet werden sollen (Befehl: **ADR**).

cbBuffLen: Länge der Daten, die gesendet werden sollen. Die Länge sollte immer kleiner/gleich der Variabel sein auf die der Pointer von `pBuffAddr` zeigt (Befehl: **SIZEOF**)

OUTPUT

```
VAR_OUTPUT
  bBusy      : BOOL;
  bError     : BOOL;
  iErrorId   : WORD;
END_VAR
```

bBusy: Dieser Ausgang bleibt solange auf TRUE, bis die Befehlsausführung abgeschlossen wurde.

bError: Dieser Ausgang wird auf TRUE geschaltet, wenn bei der Ausführung eines Befehls ein Fehler aufgetreten ist. Der befehlspezifische Fehlercode ist in `iErrorId` enthalten.

iErrorId: Enthält den befehlspezifischen Fehlercode des zuletzt ausgeführten Befehls (siehe Tabelle).

5.13.4.2.3 TCP/IP

TCP/IP Client

TCP/IP ist eine verbindungsorientierte Kommunikationsart, eigentlich eine peer-to-peer Verbindung. In diesem Beispiel wird ihnen gezeigt wie eine Verbindung von der Beckhoff Steuerung zu einem Server aufgebaut wird. Die benötigten Bausteine sind `FB_IpStartSession`, `FB_IpOpen`, `FB_IpSend` und optional `FB_IpReceive`, `FB_IpClose` und `FB_IpEndSession`.

Es empfiehlt sich eine Schrittkette zu programmieren, wie in dem Beispiel (siehe unten) angegeben ist.

Die Beckhoff Steuerung ist hier der Client und die Gegenstelle ein TCP Server. Der Server ist ein VB6 Programm.

Schritt 1

FB_IpStartSession

Der Baustein reserviert Ressourcen auf der Steuerung für die TCP/IP Kommunikation. Mit `bStart` wird der Baustein aktiv. Solange der Baustein arbeitet ist das **bBusy** gesetzt. **iDevice** ist immer mit null zu belegen. **iPort** ist für die lokale TCP/IP Port Nummer. die **iMaxConnection** gibt die Anzahl der maximal möglichen Verbindungen an, maximal ist dieser Wert 3.

INPUT

bStart: Eine steigende Flanke aktiviert den Baustein.

iDevice: immer "0"

eSocketType: bei TCP/IP ist "SOCK_STREAM" zu benutzen. Beachten Sie das die Daten als Byte-Stream abgelegt werden.

Sie sollten die Länge der zu empfangenden Daten wissen oder ein Protokoll mit Start- und Endkennung verwenden, damit Sie im Daten-Stream eindeutig Anfang und Ende erkennen können.

eSocketProtocol: für TCP/IP ist "IPPROTO_TCP" zu benutzen

iPort: Absender Port Nummer

iMaxConnection: Anzahl der maximal möglichen Verbindungen (max. ist 3)

tTimeout: Zeit nach der abgebrochen werden soll.

OUTPUT

bBusy: Dieser Ausgang bleibt solange auf TRUE, bis die Befehlsausführung abgeschlossen wurde.

bError: Dieser Ausgang wird auf TRUE geschaltet, wenn bei der Ausführung eines Befehls ein Fehler aufgetreten ist. Der befehlspezifische Fehlercode ist in *iErrorId* enthalten.

iErrorId: Enthält den befehlspezifischen Fehlercode des zuletzt ausgeführten Befehls (siehe Tabelle).

iSession: Gibt die Session Nummer an alle IP Bausteine weiter, für die diese Verbindung angelegt wurde.

Schritt 2

FB_IpOpen

Daten die empfangen werden, müssen in eine PLC Variable abgelegt werden. Dafür müssen Sie einen Pointer auf **pBuffAddr** legen und die Größe der Variable in **cbBuffLen** eintragen. Mit einer positiven Flanke von **bValid** wird angezeigt das Daten im Speicher liegen bzw. die Daten ihrer Variable jetzt gültig sind.

INPUT

bClear: Eine steigende Flanke löscht den Speicher und der Baustein ist wieder bereit Daten zu empfangen.

iSession: wird mit dem iSession aus dem Baustein FB_StartSession verbunden

sRemotelIPAddr: kann als Filter verwendet werden um nur eine spezielle IP-Adresse zuzulassen

eSocketProtocol: für TCP/IP ist "IPPROTO_TCP" zu benutzen

iPort: Locale Port Nummer

iMaxConnection: Anzahl der maximal möglichen Verbindungen (max. ist 3)

tTimeout: Zeit nach der abgebrochen werden soll.

OUTPUT

bBusy: Dieser Ausgang bleibt solange auf TRUE, bis die Befehlsausführung abgeschlossen wurde.

bError: Dieser Ausgang wird auf TRUE geschaltet, wenn bei der Ausführung eines Befehls ein Fehler aufgetreten ist. Der befehlspezifische Fehlercode ist in *iErrorId* enthalten.

iErrorId: Enthält den befehlspezifischen Fehlercode des zuletzt ausgeführten Befehls (siehe Tabelle).

iSession: Gibt die Session Nummer an alle IP Bausteine weiter, für die diese Verbindung angelegt wurde.

Sobald die ersten Daten empfangen wurden können auch Daten wieder zurückgeschickt werden. Dies ist optional und wird in dem Beispiel verwendet.

Beispiel



Download VB6 Programm als TCP/IP Server Zip File

(<https://infosys.beckhoff.com/content/1031/bc9xx0/Resources/zip/3207366411.zip>)



Download TwinCAT Projekt als TCP/IP Client prx File

(<https://infosys.beckhoff.com/content/1031/bc9xx0/Resources/zip/3207368587.zip>)

TCP/IP Server

TCP/IP ist eine verbindungsorientierte Kommunikationsart, eigentlich eine peer-to-peer Verbindung. In diesem Beispiel wird Ihnen gezeigt wie eine Verbindung von außen auf die Beckhoff Steuerung aufgebaut wird. Die benötigten Bausteine sind FB_IpStartSession, FB_IpReceive und optional FB_IpSend, FB_IpClose und FB_IpEndSession.

Es empfiehlt sich eine Schrittkette zu programmieren, wie in dem Beispiel (siehe unten) angegeben ist.

Die Beckhoff Steuerung ist hier Server und die Gegenstelle ein TCP Client. Der Client ist ein VB6 Programm.

Schritt 1

FB_IpStartSession

Der Baustein reserviert Ressourcen auf der Steuerung für die TCP/IP Kommunikation. Mit *bStart* wird der Baustein aktiviert. Solange der Baustein arbeitet ist das **bBusy** gesetzt. **iDevice** ist immer mit null zu belegen. **iPort** ist für die lokale TCP/IP Port Nummer. die **iMaxConnection** gibt die Anzahl der maximal möglichen Verbindungen an, maximal ist dieser Wert 3.

INPUT

bStart: Eine steigende Flanke aktiviert den Baustein.

iDevice: immer "0"

eSocketType: bei TCP/IP ist "SOCK_STREAM" zu benutzen. Beachten Sie das die Daten als Byte Stream abgelegt werden. Sie sollten möglichst die Länge der empfangenden Daten wissen oder ein Protokoll mit einer Start und Endkennung damit Sie aus den Stream von Daten eindeutig den Anfang und das Ende erkennen können.

eSocketProtocol: für TCP/IP ist "IPPROTO_TCP" zu benutzen

iPort: Empfangsport Nummer

iMaxConnection: Anzahl der maximal möglichen Verbindungen (max. ist 3)

tTimeout: Zeit nach der abgebrochen werden soll.

OUTPUT

bBusy: Dieser Ausgang bleibt solange auf TRUE, bis die Befehlsausführung abgeschlossen wurde.

bError: Dieser Ausgang wird auf TRUE geschaltet, wenn bei der Ausführung eines Befehls ein Fehler aufgetreten ist. Der befehlspezifische Fehlercode ist in *iErrorId* enthalten.

iErrorId: Enthält den befehlspezifischen Fehlercode des zuletzt ausgeführten Befehls (siehe Tabelle).

iSession: Gibt die Session Nummer an alle IP Bausteine weiter, für die diese Verbindung angelegt wurde.

Schritt 2

FB_IpReceive

Daten die empfangen werden, müssen in eine PLC Variable abgelegt werden. Dafür müssen Sie einen Pointer auf *pBuffAddr* legen und die Größe der Variable in *cbBuffLen*. Mit einer positiven Flanke von *bValid* wird angezeigt das Daten im Speicher liegen bzw. die Daten ihrer Variable jetzt gültig sind.

INPUT

bClear: Eine steigende Flanke löscht den Speicher und der Baustein ist wieder bereit Daten zu empfangen.

iSession: wird mit dem *iSession* aus dem Baustein FB_StartSession verbunden

sRemoteIPAddr: kann als Filter verwendet werden um nur eine spezielle IP Adresse

eSocketProtocol: für TCP/IP ist "IPPROTO_TCP" zu benutzen

iPort: Locale Port Nummer

iMaxConnection: Anzahl der maximal möglichen Verbindungen (max. ist 3)

tTimeout: Zeit nach der abgebrochen werden soll.

OUTPUT


bBusy: Dieser Ausgang bleibt solange auf TRUE, bis die Befehlsausführung abgeschlossen wurde.


bError: Dieser Ausgang wird auf TRUE geschaltet, wenn bei der Ausführung eines Befehls ein Fehler aufgetreten ist. Der befehlspezifische Fehlercode ist in *iErrorId* enthalten.

iErrorId: Enthält den befehlspezifischen Fehlercode des zuletzt ausgeführten Befehls (siehe Tabelle).

iSession: Gibt die Session Nummer an alle IP Bausteine weiter, für die diese Verbindung angelegt wurde.

Beispiel

 Download VB6 Programm als TCP/IP Client Zip File
 (<https://infosys.beckhoff.com/content/1031/bc9xx0/Resources/zip/3207370763.zip>)

 Download TwinCAT Projekt als TCP/IP Server prx File
 (<https://infosys.beckhoff.com/content/1031/bc9xx0/Resources/zip/3207372939.zip>)

5.13.4.2.4 UDP/IP

UDP/IP-Verbindung

UDP ist eine sehr einfache Ethernet-Verbindung. UDP-Daten werden versendet, ohne dass es einen Mechanismus gibt ob das Telegramm angekommen ist oder nicht. Für eine funktionierende UDP-Kommunikation muss die Port-Nummer auf beiden Seiten bekannt sein.

Der BX9000 sendet Daten an ein VB6-Programm, das diese Daten wieder zurück schickt.

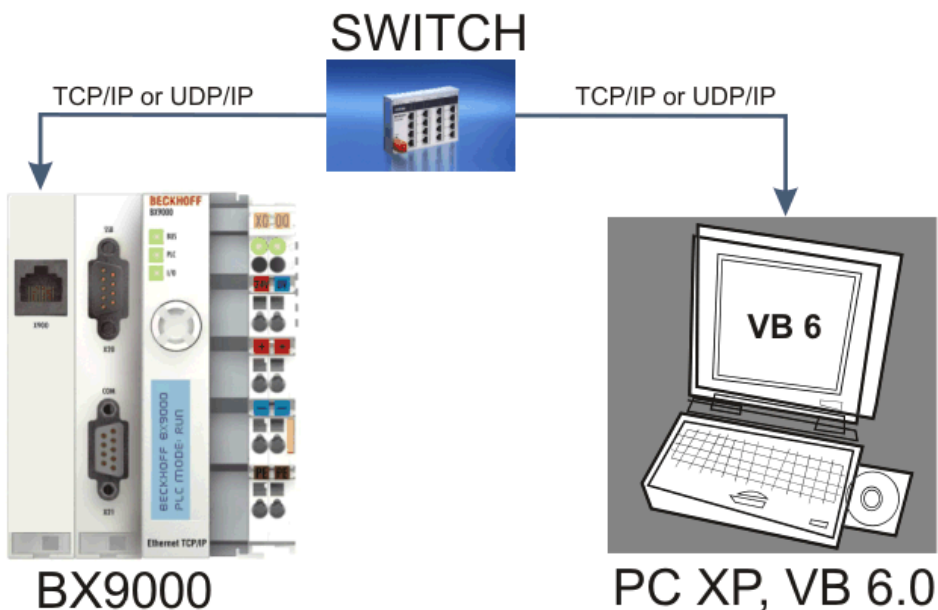


Abb. 77: UDP/IP-Verbindung

Schritt 1: Vorbereitung der UDP Kommunikation

FB_IpStartSession

Der Baustein reserviert Ressourcen auf der Steuerung für die UDP/IP Kommunikation. Mit *bStart* wird der Baustein aktiv. Solange der Baustein arbeitet ist das **bBusy** gesetzt. **iDevice** ist immer mit null zu belegen. **iPort** ist für die lokale UDP Port Nummer des BX9000. Die **iMaxConnection** gibt die Anzahl der maximal möglichen Verbindungen an, maximal ist dieser Wert 3. Der **eSocketType** ist auf "SOCK_DGRAM" zu stellen. **eSocketProtocol** muss auf "IPPROTO_UDP" stehen für die UDP Kommunikation. Der *tTimeout* wird für die UDP Kommunikation nicht verwendet. Die **iSession** muss mit den folgenden Bausteinen FB_IpSend und FB_IpReceive verknüpft sein.

Schritt 2: Senden von UDP-Frames

FB_IpSend

Mit einer positiven Flanke von **bStart** wird ein UDP Frame abgeschickt. Die IP-Adresse wird mit **sRemoteIPAddr** beschrieben und die Ziel UDP Port Nummer mit **iPortNo**. Die Absender UDP Port Nummer wurde ja schon in den **FB_IpStartSession** konfiguriert. Wird das **bBusy** vom Baustein zurückgesetzt ist der Befehl ausgeführt worden.

Schritt 3: Empfangen von UDP-Frames

FB_IpReceive

Mit diesem Baustein empfängt man die Daten. Sobald der Baustein aufgerufen wird hört der Baustein ob UDP-Frames angekommen sind. Es werden bis zu 4 UDP-Frames gepuffert, weitere UDP-Frames werden verworfen. Mit **sRemoteIPAddr** kann man einen IP-Adressfilter parametrieren um nur von einem bestimmten Teilnehmer Daten zu empfangen. Will man alle UDP-Frames empfangen gibt man ein leer String an oder lässt diese Variabel offen. Hat man einen IP-Adressfilter parametrieren kann man zusätzlich noch die Port Nummer rausfiltern. Dafür setzt man einfach in die Variabel **iPortNo** die entsprechende Portnummer ein. Will man auch hier alle UDP Daten empfangen dann lässt man die Variabel offen. Werden Daten empfangen wird die Variabel *bValid* auf TRUE gesetzt. Die Daten sind jetzt gültig. Ist der Wert der Variabel **cbBytesInStream** ungleich Null, so ist die Variabel, die an den Baustein verknüpft worden ist zu klein gewählt worden und es befinden sich noch Daten im Puffer.

Ablauf des Beispielprogramms

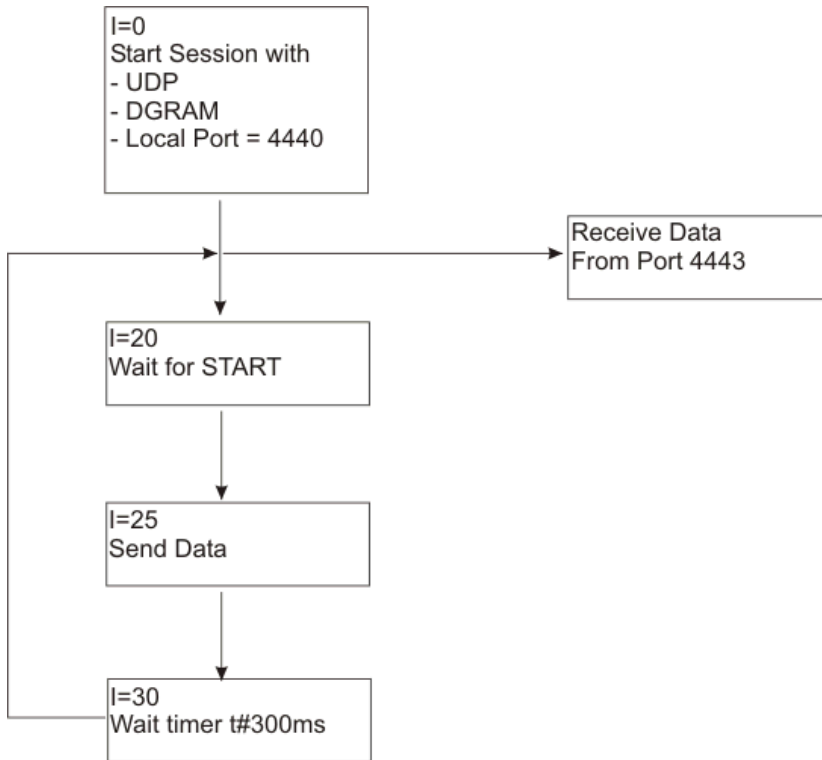


Abb. 78: Ablauf des Beispielprogramms

Beispiel



Download VB6 Programm als TCP/IP Server, Zip File
 (<https://infosys.beckhoff.com/content/1031/bc9xx0/Resources/zip/3207375115.zip>)



Download BX9000 TwinCAT Projekt als TCP/IP Client, prx File
 (<https://infosys.beckhoff.com/content/1031/bc9xx0/Resources/zip/3207377291.zip>)

5.13.4.3 ModbusTCP Client

Mit dem ModbusTCP Client (Modbus Master) kann der BX9000 eine Verbindung zu einem ModbusTCP-Server (Modbus Slave) aufnehmen. Es stehen 3 Bausteine zur Verfügung. Fb_MBConnect zum Aufbauen einer TCP/IP Verbindung, FB_MBGenericReq für das Versenden und Empfangen von beliebigen Modbus Funktionen und den FB_MBClose zum Schließen der TCP/IP Verbindung. Da es sich beim FB_MBGenericReq um einen allgemeinen Baustein handelt, müssen die entsprechenden Modbus Funktionen noch implementiert werden, dies erlaubt daher auch nicht Modbus konforme Protokolle zu versenden. Es sind maximal 3 Client Verbindungen gleichzeitig möglich. Durch die Möglichkeit eine TCP/IP Verbindung zu schließen und neu öffnen zu können, sind damit nahezu beliebig viele ModbusTCP-Server ansprechbar.

FB_MBConnect

Der FB_MBConnect verbindet den BX9000 über ModbusTCP mit einem anderen Teilnehmer. Es wird eine Verbindung aufgebaut mit steigender Flanke von *bExecute*. Die IP-Adresse wird über *sIPAddr* eingestellt. Die zu verwendete TCP-Portnummer über *nTCPPort* und ist normalerweise bei ModbusTCP auf 502 spezifiziert. Solange der Baustein aktiv ist, ist das **bBusy** gesetzt. Wurde die ModbusTCP Verbindung erfolgreich initialisiert, wird das *bBusy* auf FALSE gesetzt und das *bError* Flag ist FALSE. Sollte das *bError* Flag TRUE sein, ist der Verbindungsaufbau fehlgeschlagen und in der Variable *nErrId* ist der entsprechende Fehlercode enthalten. **nHandlemuss** mit den Bausteinen **FB_MBGenericReq** und **FB_MBClose** verbunden werden.

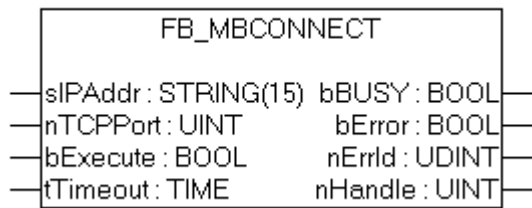


Abb. 79: Funktionsbaustein FB_MBCONNECT

INPUT

```

VAR_INPUT
  sIPAddr      : STRING(15);
  nTCPPort     : UINT;
  bExecute     : BOOL;
  tTimeout     : TIME;
END_VAR

```

sIPAddr: IP-Adresse des ModbusTCP-Servers (Slave Gerät) mit dem eine Modbus Verbindung hergestellt werden soll.

nTCPPort: ModbusTCP Port Nummer, normalerweise ist diese bei ModbusTCP 502dez.

bExecute: Eine steigende Flanke aktiviert den Baustein.

tTimeout: Zeit nach der abgebrochen werden soll.

OUTPUT

```

VAR_OUTPUT
  bBusy        : BOOL;
  bError       : BOOL;
  iErrId       : WORD;
  nHandle      : UINT;
END_VAR

```

bBusy: Dieser Ausgang bleibt solange auf TRUE, bis die Befehlsausführung abgeschlossen wurde.

bError: Dieser Ausgang wird auf TRUE geschaltet, wenn bei der Ausführung eines Befehls ein Fehler aufgetreten ist. Der befehlspezifische Fehlercode ist in *iErrorId* enthalten.

iErrorId: Enthält den befehlspezifischen Fehlercode des zuletzt ausgeführten Befehls ([siehe Tabelle](#) [▶ 106](#)).

nHandle: Handle, der mit den Bausteinen FB_MBGenericReq und FB_MBClose verbunden werden muss.

FB_MBGENERICREQ

Der Baustein FB_MBGENERICREQ ermöglicht das Senden und Empfangen beliebiger ModbusTCP Funktionen. Mit steigender Flanke von *bExecute* wird der Baustein aktive, das **bBusy** wird gesetzt und er sendet seine Daten, die im **pReqBuff** (Pointer auf die Daten die verschickt werden sollen) enthalten sind. Die Länge der zu senden Daten entnimmt der Baustein der Variable **cbReqLen**. Die Variable **nHandle** muss mit der Variable aus dem Baustein Fb_MBCconnect **nHandle** verbunden werden. Die Antwort des ModbusTCP-Servers (Slave) wird in dem **pResBuff** (Pointer auf die Empfangsdaten) abgelegt. Die Größe der Variable sollte groß genug gewählt werden, damit alle Daten eines ModbusTCP Telegramms empfangen werden. Empfängt der ModbusClient innerhalb der **tTimeout** Zeit keine Antwort, so wird das **bBusy** auf FALSE gesetzt und das **bError** gesetzt. In der Variable **nErrId** findet man den Fehlercode. In **cbResponse** ist die Anzahl der empfangenden Bytes enthalten. Diese können mit der Puffergröße der **cdResLen** verglichen werden. Ist **cbResponse** größer als **dResLen** gehen Daten verloren und Buffer muss größer gewählt werden.

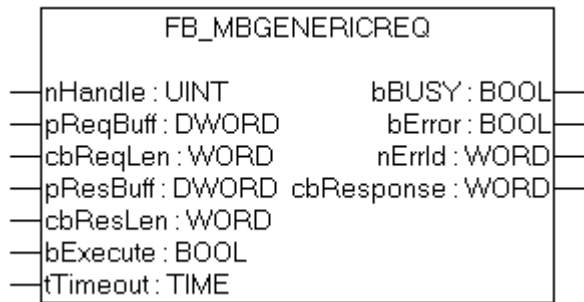


Abb. 80: Funktionsbaustein FB_MBGENERICREQ

INPUT

```
VAR_INPUT
    nHandle      : UINT;
    pReqBuff     : DWORD;
    cbReqLen     : WORD;
    pResBuff     : DWORD;
    cbResLen     : WORD;
    bExecute     : BOOL;
    tTimeout     : TIME;
END_VAR
```

nHandle: Handle, wird mit den Bausteinen Fb_MBCconnect verbunden.

pReqBuff: Pointer auf die Daten, die versendet werden sollen.

cbReqLen: Länge der Daten die versendet werden sollen.

pResBuff: Pointer auf die Daten, die empfangen werden sollen. Die Variable muss groß genug angelegt werden.

cbResLen: Länge der Daten die empfangen werden sollen. Die Länge muss groß genug sein.

bExecute: Eine steigende Flanke aktiviert den Baustein.

tTimeout: Zeit nach der abgebrochen werden soll.

OUTPUT

```
VAR_OUTPUT
    bBusy       : BOOL;
    bError      : BOOL;
    iErrId      : WORD;
    cbResponse  : WORD;
END_VAR
```

bBusy: Dieser Ausgang bleibt solange auf TRUE, bis die Befehlsausführung abgeschlossen wurde.

bError: Dieser Ausgang wird auf TRUE geschaltet, wenn bei der Ausführung eines Befehls ein Fehler aufgetreten ist. Der befehlspezifische Fehlercode ist in *iErrorId* enthalten.

iErrorId: Enthält den befehlspezifischen Fehlercode des zuletzt ausgeführten Befehls (siehe Tabelle [▶ 106]).

cbResponse: Anzahl an Bytes die empfangen wurden.

FB_MBCLOSE

Der Baustein FB_MBCLOSE ermöglicht das Schließen einer TCP/IP Verbindung. Mit einer steigenden Flanke von **bExecute** wird der Baustein aktiviert. Solange der Baustein arbeitet ist das Bit **bBusy** gesetzt. **nHandle** muss mit dem Baustein Fb_MBCconnect **nHandle** verbunden sein. Tritt ein Fehler auf wird **bError** auf TRUE gesetzt und in **nErrId** findet man den Fehlercode. Nach fehlerfreien Aufruf des Bausteins kann mit Fb_MBCconnect ein neuer Socket geöffnet werden.

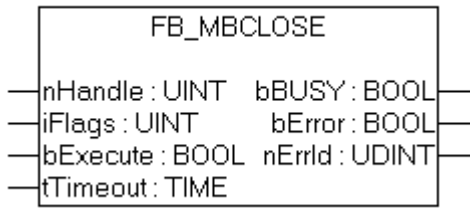


Abb. 81: Funktionsbaustein FB_MBCLOSE

INPUT

```

VAR_INPUT
  nHandle      : UINT;
  iFlags       : UINT;
  bExecute     : BOOL;
  tTimeout     : TIME;
END_VAR
  
```

nHandle: Handle, wird mit den Bausteinen Fb_MBCConnect verbunden.

iFlags: 0 - TCP/IP Verbindung wird geschlossen auch wenn die Gegenstelle nicht erreichbar ist, 1 - TCP/IP Verbindung wird mit "FIN" beendet, damit der Baustein ohne Fehler die TCP/IP Verbindung schließt muss dieser vorhanden sein und die Kommunikation funktionieren. Es empfiehlt sich die 0 zu nehmen da hier auf jedenfall unabhängig vom Zustand der Gegenstelle die TCP/IP Verbindung geschlossen wird.

bExecute: Eine steigende Flanke aktiviert den Baustein.

tTimeout: Zeit nach der abgebrochen werden soll.

OUTPUT

```

VAR_OUTPUT
  bBusy        : BOOL;
  bError       : BOOL;
  iErrId       : WORD;
END_VAR
  
```


bBusy: Dieser Ausgang bleibt solange auf TRUE, bis die Befehlsausführung abgeschlossen wurde.


bError: Dieser Ausgang wird auf TRUE geschaltet, wenn bei der Ausführung eines Befehls ein Fehler aufgetreten ist. Der befehlspezifische Fehlercode ist in *iErrorId* enthalten.

iErrorId: Enthält den befehlspezifischen Fehlercode des zuletzt ausgeführten Befehls (siehe Tabelle [▶ 106]).

Beispiel

Für das Beispiel brauchen Sie zwei BX9000 oder statt eines BX9000 einen BC9x00.

 Download Erster BX9000 als ModbusTCP Client
 (<https://infosys.beckhoff.com/content/1031/bc9xx0/Resources/zip/3207379467.zip>)

 Download Zweiter BX9000 als ModbusTCP-Server (oder einen BC9x00)
 (<https://infosys.beckhoff.com/content/1031/bc9xx0/Resources/zip/3207381643.zip>)

Entwicklungsumgebung	Zielplattform	Einzubindende SPS Bibliotheken
TwinCAT v2.10.0 und höher	BX9000 (165) firmware version ≥1.14	TcBaseBX9000.lbx

Error Codes

Fehlercode	Beschreibung
0xFB00	keine gültige IP-Adresse
0xFC00	Response Länge größer als Speicher
0xFD00	Request konnte nicht abgesetzt werden
0xFF00	TimeOut während der Kommunikation

Modbus Close

Fehlercode	Beschreibung
0xFA00	Verbindung bereits geschlossen
0xFF00	TimeOut beim Schließen der Verbindung. Es wird RST verwendet und der Remote Teilnehmer ist nicht vorhanden.

Modbus Client Connect

Fehlercode	Beschreibung
0xFA00	Interne Verbindung wird verweigert
0xFA01	Connection Timeout Event
0xFA02	Connect wurde vom Remote nicht akzeptiert
0xFA03	ungültige IP Adresse
0xFA04	Alle Modbus Verbindungen belegt
0xFA05	kein interner Socket mehr vorhanden
0xFA06	interner Socket Fehler
0xFA07	Fehler beim Connect Aufruf

5.13.4.4 SMTP

5.13.4.4.1 FB_Smtp

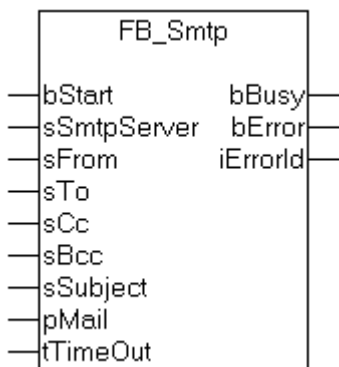


Abb. 82: Funktionsbaustein FB_Smtp

Der Funktionsbaustein nutzt das SMTP-Protokoll (Simple Mail Transfer Protocol) um E-Mails zu verschicken. Der Funktionsbaustein kann z. B. dazu benutzt werden Fehler, Diagnoseinformationen, Warnungen als E-Mail zu verschicken. Die Empfängeradressen werden als Strings an die *sTo*-, *sCc*-, *sBcc*- und *sSubject*-Eingangsvariablen übergeben. Die maximale Stringlänge der Empfängeradressen wurde auf 80 Zeichen begrenzt um die Ressourcen zu schonen. Der String mit dem eigentlichen Mail-Text darf auch länger sein.

VAR_INPUT

```

VAR_INPUT
  bStart      : BOOL;
  sSmtServer  : STRING(15);
  sFrom       : STRING;
  sTo         : STRING;
  sCc         : STRING;
  sBcc        : STRING;
  sSubject    : STRING;
  pMail       : DWORD;
  tTimeOut    : TIME;
END_VAR

```

bStart: Eine steigende Flanke an diesem Eingang aktiviert den Funktionsbaustein.

sSmtServer: IP-Adresse des SMTP-Servers als String.

sFrom: Ein String mit der E-Mail Adresse des Absenders. Wird ein Leerstring angegeben, dann erzeugt der Bus-Controller eine E-Mail Adresse aus dem Namen des Bus-Controllers und der MAC ID. Die maximale Stringlänge ist auf 80 Zeichen begrenzt. Es dürfen auch mehrere Empfängeradressen getrennt durch Semikolons angegeben werden.

sTo: Ein String mit der E-Mail Adresse des Empfängers. Es muss eine gültige E-Mail Adresse angegeben werden. Die maximale Stringlänge ist auf 80 Zeichen begrenzt. Es dürfen auch mehrere Empfängeradressen getrennt durch Semikolons angegeben werden.

sCc: Ein String mit der E-Mail Adresse eines weiteren Empfängers (Cc=Carbon Copy). Es kann auch ein Leerstring angegeben werden. Eine Kopie der E-Mail wird an diesen Empfänger gesendet. Die E-Mail Adresse dieses Empfängers wird bei anderen Empfängern **sichtbar**. Die maximale Stringlänge ist auf 80 Zeichen begrenzt. Es dürfen auch mehrere Empfängeradressen getrennt durch Semikolons angegeben werden.

sBcc: Ein String mit der E-Mail Adresse eines weiteren Empfängers (Bcc=Blind Carbon Copy). Es kann auch ein Leerstring angegeben werden. Eine Kopie der E-Mail wird an diesen Empfänger gesendet. Die E-Mail Adresse dieses Empfängers wird bei anderen Empfängern **nicht sichtbar**. Die maximale Stringlänge ist auf 80 Zeichen begrenzt. Es dürfen auch mehrere Empfängeradressen getrennt durch Semikolons angegeben werden.

sSubject: Ein String mit der Betreff-Zeile zu der E-Mail. Es kann auch ein Leerstring angegeben werden. Die maximale Stringlänge ist auf 80 Zeichen begrenzt.

pMail: Die Adresse (Pointer) eines nullterminierten Strings mit dem E-Mail Text. Es kann auch ein Leerstring angegeben werden. Die Adresse des Strings kann mit dem ADR-Operator ermittelt werden.

tTimeOut: Maximale Zeit, die bei der Ausführung des Befehls nicht überschritten werden darf.

VAR_OUTPUT

```

VAR_OUTPUT
  bBusy       : BOOL;
  bError      : BOOL;
  iErrorId    : WORD;
END_VAR

```

bBusy: Dieser Ausgang bleibt solange auf TRUE, bis der Baustein eine Befehlsanforderung ausführt, längstens aber für die Dauer der, an dem *tTimeOut*-Eingang angelegten Zeit.

bError: Dieser Ausgang wird auf TRUE geschaltet, wenn bei der Ausführung eines Befehls ein Fehler aufgetreten ist. Der befehlspezifische Fehlercode ist in *iErrorId* enthalten.

iErrorId: Enthält den befehlspezifischen Fehlercode des zuletzt ausgeführten Befehls (Tabelle).

Fehlercode (hex)	Beschreibung
0x8000	SMTP server not found.
0x8001	Resource error.
0x8002	Socket resource error.
0x8003	Connection fault.
0x8004	Communication fault.
0x8005	Rx error. Communication time exceeded.
0x8006	Rx error. Communication fault.
0x8007	Rx error. Frame error.
0x8008	Communication error. Wrong response.
0x8009	Tx error. Communication fault.
0x800A	Communication shutdown error.
0x800B	Communication timeout.
0x8010	Invalid parameter.

Beispiel für einen Aufruf in FUP

```

PROGRAM MAIN
VAR
    fbSMTP : FB_Smtp;
    bSend   : BOOL;
    sMsg    : STRING(100) := 'Test';
    bBusy   : BOOL;
    bError  : BOOL;
    nErrId  : UDINT;
END_VAR
    
```

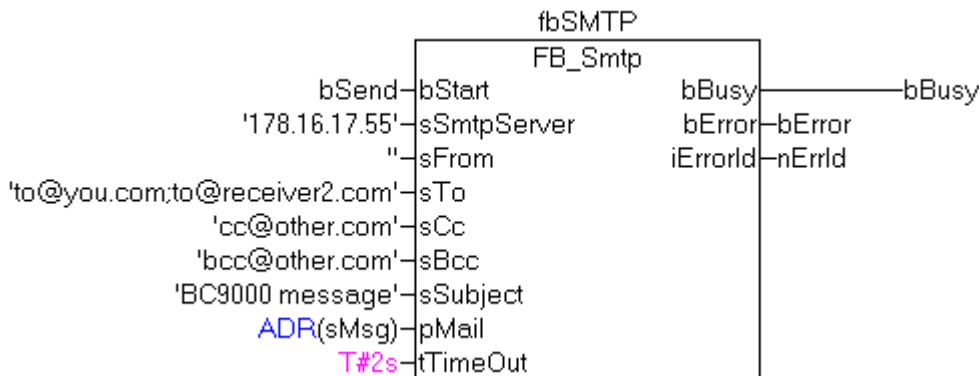


Abb. 83: Funktionsbaustein fbSMTP

Im Beispiel wird bei einer steigenden Flanke am *bStart*-Eingang eine Mail an 4 Empfänger verschickt.

Entwicklungsumgebung	Zielplattform	Einzubindende SPS Bibliotheken
TwinCAT v2.10.0 und höher	BX9000 (165) firmware version ≥1.12	TcBaseBX9000.lbx

5.13.4.5 SNTP

5.13.4.5.1 Time Protokoll (SNTP)

(BX9000 ab Firmware-Version 1.12, BC9050, BC9x20)

Das Simple Network Time Protokoll dient zu synchronisieren von Uhren über das Internet. Sie können den BX9000 zu einem Time-Server synchronisieren.

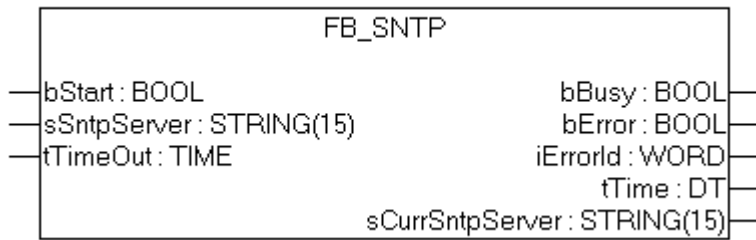


Abb. 84: Funktionsbaustein FB_Sntp

FUNCTION_BLOCK FB_Sntp

Wird eine IP-Adresse eingetragen benutzt der Busklemmen Controller das Sntp-Protokoll. Wird eine Leer-String übergeben wird das Time-Protokoll (UDP Port 37) verwendet.

VAR_INPUT

```
bStart          :BOOL;
sSntpServer     :STRING(15);
tTimeout       :TIME;
```

bOpen: Positive Flanke startet den Baustein

sSntpServer: Eintrag des Sntp Servers. Wenn ein Leerstring eingegeben wird, wird das Time-Protokoll verwendet (UDP Port 37)*.

tTimeout: TMOut nach dem abgebrochen werden soll

VAR_OUTPUT

```
bBusy          :BOOL;
bError         :BOOL;
iErrorId       :WORD;
tTime          :DT;
sCurrSntpServer :STRING(15);
```

bBusy: Solange der Baustein TRUE ist der Baustein aktiv.

bError: Fehler Bit.

iErrorId: Fehler Nummer.

tTime: Zeit und Datum.

sCurrSntpServer: IP-Adresse des Sntp-Servers

Rückgabeparameter iErrId	Bedeutung
0	kein Fehler
<> 0	Fehler Nummer [▶ 114]



Download

(<https://infosys.beckhoff.com/content/1031/bc9xx0/Resources/zip/3207383819.zip>)

5.13.4.6 Services

5.13.4.6.1 FB_AddDnsServer

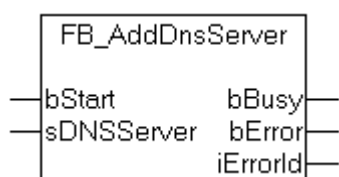


Abb. 85: Funktionsbaustein FB_AddDnsServer

Mit dem Funktionsbaustein können dem Bus-Controller maximal bis zu drei DNS-Server übergeben werden.

INPUT

```
VAR_INPUT
  bStart      : BOOL;
  sDNSServer  : STRING(15);
END_VAR
```

bStart: Eine steigende Flanke an diesem Eingang aktiviert den Funktionsbaustein.

sDNSServer: String mit der IP-Adresse des DNS-Servers.

OUTPUT

```
VAR_OUTPUT
  bBusy       : BOOL;
  bError      : BOOL;
  iErrorId    : WORD;
END_VAR
```

bBusy: Dieser Ausgang bleibt solange auf TRUE, bis die Befehlsausführung abgeschlossen wurde.

bError: Dieser Ausgang wird auf TRUE geschaltet, wenn bei der Ausführung eines Befehls ein Fehler aufgetreten ist. Der befehlspezifische Fehlercode ist in *iErrorId* enthalten.

iErrorId: Enthält den befehlspezifischen Fehlercode des zuletzt ausgeführten Befehls (siehe Tabelle [▶ 114]).

Entwicklungsumgebung	Zielpattform	Einzubindende SPS Bibliotheken
TwinCAT v2.10.0 und höher	BX9000 (165) firmware version ≥1.12 BC9191 firmware version ≥3.1	TcBaseBX9000.lbx

5.13.4.6.2 FB_GetHostByAddr



Abb. 86: Funktionsbaustein FB_GetHostByAddr

Mit dem Funktionsbaustein kann zu einer bestimmten IP-Adresse der zugehörige Hostname ermittelt werden.

INPUT

```
VAR_INPUT
  bStart      : BOOL;
  sIPAddr     : STRING(15);
  pHostName   : DWORD;
  cbMaxNameLen : WORD;
END_VAR
```

bStart: Eine steigende Flanke an diesem Eingang aktiviert den Funktionsbaustein.

sIPAddr: Ein String mit der IP-Adresse des Hosts.

pHostName: Enthält die Adresse eines String-Puffers, in den der ermittelte Hostname hineingeschrieben wird. Der Programmierer ist selbst dafür verantwortlich, den Puffer in der Größe so zu dimensionieren, dass *cbMaxNameLen*-Bytes daraus entnommen werden können. Die Adresse kann man mit dem ADR - Operator ermitteln.

cbMaxNameLen: Enthält die Bytelänge des Puffers in den der ermittelte Hostname hineingeschrieben werden soll.

OUTPUT

```
VAR_OUTPUT
  bBusy      : BOOL;
  bError     : BOOL;
  iErrorId   : WORD;
END_VAR
```

bBusy: Dieser Ausgang bleibt solange auf TRUE, bis die Befehlsausführung abgeschlossen wurde.

bError: Dieser Ausgang wird auf TRUE geschaltet, wenn bei der Ausführung eines Befehls ein Fehler aufgetreten ist. Der befehlspezifische Fehlercode ist in *iErrorId* enthalten.

iErrorId: Enthält den befehlspezifischen Fehlercode des zuletzt ausgeführten Befehls (siehe Tabelle [► 114]).

Entwicklungsumgebung	Zielplattform	Einzubindende SPS Bibliotheken
TwinCAT v2.10.0 und höher	BX9000 (165) firmware version ≥1.12 BC9191 firmware version ≥3.1	TcBaseBX9000.lbx

5.13.4.6.3 FB_GetHostByName



Abb. 87: Funktionsbaustein FB_GetHostByName

Mit dem Funktionsbaustein kann zu einem bestimmten Hostnamen die zugehörige IP-Adresse ermittelt werden.

INPUT

```
VAR_INPUT
  bStart      : BOOL;
  pHostName   : DWORD;
END_VAR
```

bStart: Eine steigende Flanke an diesem Eingang aktiviert den Funktionsbaustein.

pHostName: Enthält die Adresse einer Stringvariablen, mit dem Hostnamen. Die Adresse einer Stringvariablen kann mit dem ADR-Operator ermittelt werden.

OUTPUT

```
VAR_OUTPUT
  bBusy      : BOOL;
  bError     : BOOL;
  iErrorId   : WORD;
  sIPAddr    : STRING(15);
END_VAR
```

bBusy: Dieser Ausgang bleibt solange auf TRUE, bis die Befehlsausführung abgeschlossen wurde.

bError: Dieser Ausgang wird auf TRUE geschaltet, wenn bei der Ausführung eines Befehls ein Fehler aufgetreten ist. Der befehlspezifische Fehlercode ist in *iErrorId* enthalten.

iErrorId: Enthält den befehlspezifischen Fehlercode des zuletzt ausgeführten Befehls (siehe Tabelle [► 114]).

sIPAddr: Beim Erfolg enthält die Variable die IP-Adresse des Hosts.

Entwicklungsumgebung	Zielplattform	Einzubindende SPS Bibliotheken
TwinCAT v2.10.0 und höher	BX9000 (165) firmware version ≥1.12 BC9191 firmware version ≥3.1	TcBaseBX9000.lbx

5.13.4.6.4 FB_GetNetworkConfig

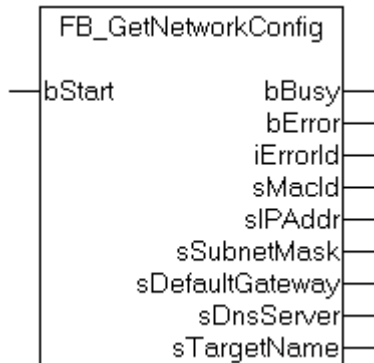


Abb. 88: Funktionsbaustein FB_GetNetworkConfig

Mit dem Funktionsbaustein können Informationen zu der aktuellen Netzwerkkonfiguration ermittelt werden.

INPUT

```
VAR_INPUT
  bStart      : BOOL;
END_VAR
```

bStart: Eine steigende Flanke an diesem Eingang aktiviert den Funktionsbaustein.

OUTPUT

```
VAR_OUTPUT
  bBusy       : BOOL;
  bError      : BOOL;
  iErrorId    : WORD;
  sMacId      : STRING(17);
  sIPAddr     : STRING(15);
  sSubnetMask : STRING(15);
  sDefaultGateway : STRING(15);
  sDnsServer  : STRING(15);
  sTargetName : STRING(20);
END_VAR
```

bBusy: Dieser Ausgang bleibt solange auf TRUE, bis die Befehlsausführung abgeschlossen wurde.

bError: Dieser Ausgang wird auf TRUE geschaltet, wenn bei der Ausführung eines Befehls ein Fehler aufgetreten ist. Der befehlspezifische Fehlercode ist in *iErrorId* enthalten.

iErrorId: Enthält den befehlspezifischen Fehlercode des zuletzt ausgeführten Befehls (siehe Tabelle [▶ 114]).

sMacId: Liefert die MAC ID des Bus-Controllers zurück, z. B. '00-01-05-13-45-63'.

sIPAddr: Liefert die IP-Adresse des Bus-Controllers zurück.

sSubnetMask: Liefert die SubNet Mask.

sDefaultGateway: Liefert den default Geteway.

sDnsServer: Liefert den default DNS-Server (vom DHCP-Server zugewiesen) zurück.

sTargetName: Liefert den aktuellen Target-Namen des Bus-Controllers zurück.

Entwicklungsumgebung	Zielplattform	Einzubindende SPS Bibliotheken
TwinCAT v2.10.0 und höher	BX9000 (165) firmware version ≥1.12 BC9191 firmware version ≥3.1	TcBaseBX9000.lbx

5.13.4.6.5 FB_SetTargetName

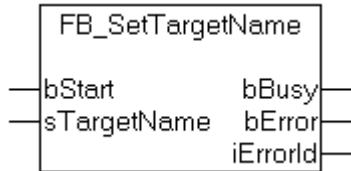


Abb. 89: Funktionsbaustein FB_SetTargetName

Mit dem Funktionsbaustein kann dem Bus-Controller ein Target-Name zugewiesen werden.

INPUT

```
VAR_INPUT
    bStart      : BOOL;
    sTargetName : STRING(20);
END_VAR
```

bStart: Eine steigende Flanke an diesem Eingang aktiviert den Funktionsbaustein.

sTargetName: Ein String, der den neuen Target-Namen des Bus-Controllers enthält.

OUTPUT

```
VAR_OUTPUT
    bBusy      : BOOL;
    bError     : BOOL;
    iErrorId   : WORD;
END_VAR
```

bBusy: Dieser Ausgang bleibt solange auf TRUE, bis die Befehlsausführung abgeschlossen wurde.

bError: Dieser Ausgang wird auf TRUE geschaltet, wenn bei der Ausführung eines Befehls ein Fehler aufgetreten ist. Der befehlspezifische Fehlercode ist in *iErrorId* enthalten.

iErrorId: Enthält den befehlspezifischen Fehlercode des zuletzt ausgeführten Befehls (siehe Tabelle).

Entwicklungsumgebung	Zielplattform	Einzubindende SPS Bibliotheken
TwinCAT v2.10.0 und höher	BX9000 (165) firmware version ≥1.12 BC9191 firmware version ≥3.1	TcBaseBX9000.lbx

5.13.4.6.6 Fehlernummern

Hex	Beschreibung
0x8010	falsche Parameter
0x8011	DNS-Server Liste voll
0x801F	Resourcen Fehler (intern)
0x8020	ungültige Hostlänge
0x8021	kein Hostname gefunden
0x802E	Heap Fehler (intern)
0x802F	Resourcen Fehler (intern)
0x8030	falsche Parameter
0x8031	ungültige Hostname
0x8032	Hostname zu lang
0x803F	Resourcen Fehler (intern)
0x804F	Resourcen Fehler (intern)
0x8050	keine gültige IP-Adresse
0x8052	keine gültige Klasse-D IP-Adresse
0x8055	Fehler im Stack (intern)
0x805F	Resourcen Fehler (intern)
0x8060	ERR_SNTP_INVALID_SERVER
0x8061	ERR_SNTP_NO_MORE_SOCKETS
0x8064	ERR_SNTP_RX_ERR
0x8065	ERR_SNTP_CONN_SHUT_DOWN
0x8066	ERR_SNTP_TIMEOUT
0x8100	falsche Parameter
0x8102	keine offene Verbindung
0x8104	keine Daten
0x8106	Pufferüberlauf (nur bei UDP)
0x8200	Verbindung wurde schon geöffnet
0x8201	Verbindungsfehler
0x8202	Zeitüberschreitung
0x8203	Resourcen Fehler (intern)
0x8300	Zeitüberschreitung
0x8301	Fernverbindung beendet
0x8302	linterner Fehler
0x8303	Resourcen Fehler (intern)
0x8304	keine Fernverbindung
0x8400	Verbindung icht geöffnet
0x8401	Zeitüberschreitung
0x8800	keine gültige Netzwerkkonfiguration (z.B. DHCP läuft noch)
0x8F09	ADS Socket steht nicht zur Verfügung (Zu viele offene ADS Verbindungen)
0x8F10	Session nicht gestartet
0x8F11	Session nicht bereit
0x8F12	ungültige Session ID
0x8F13	ungültiger Zustand (intern)
0x8F20	keine freien Slots. Es sind bereits 2 Sessions gestartet
0x8F21	Fehler bei der Taskinitialisierung
0x8F22	Session ist bereits geschlossen

5.14 Programmübertragung

5.14.1 Programmübertragung über Ethernet

TwinCAT bietet die Möglichkeit, das Anwenderprogramm über den Feldbus auf den Busklemmen-Controller zu übertragen. Im PLC Control kann nach dem Sichern in der Registry und einem Restart des TwinCAT-Systems, als Zielsystem der BC/BX angewählt werden. Erforderlich ist der TwinCAT-Level TwinCAT PLC.

Mindestanforderungen:

- TwinCAT 2.10 Build 1251

Initialisieren des Busklemmen-Controllers

Möglichkeit 1: Wenn Sie TwinCAT als pollende PLC nutzen.

Um den Koppler im PLC Control auswählen zu können, muss er dem System erst einmal bekannt gemacht werden.

Tragen Sie den Busklemmen Controller in dem System Manager ein, legen Sie Art, Anzahl und Größe der Feldbusvariablen an und verknüpfen sie diese mit einer Task. Sichern Sie die Einstellungen und aktivieren Sie die Konfiguration. Danach starten Sie das TwinCAT System und die zyklische Task.

Möglichkeit 2: Wenn Sie TwinCAT nur zum Programmieren oder Konfigurieren nutzen:

Klicken Sie auf das TwinCAT-Icon und starten Sie die Eigenschaften. Unter AMS Router können Sie den BX9000 eintragen.

Name: beliebig

AMS Net Id: IP Adresse plus ".1.1"

IP Adresse: IP Adresse des BX9000

Transport Typ: TCP/IP

Nun starten Sie TwinCAT in den Config- (blaues TwinCAT Icon) oder RUN-Modus (Grünes TwinCAT Icon)

TwinCAT System Manager

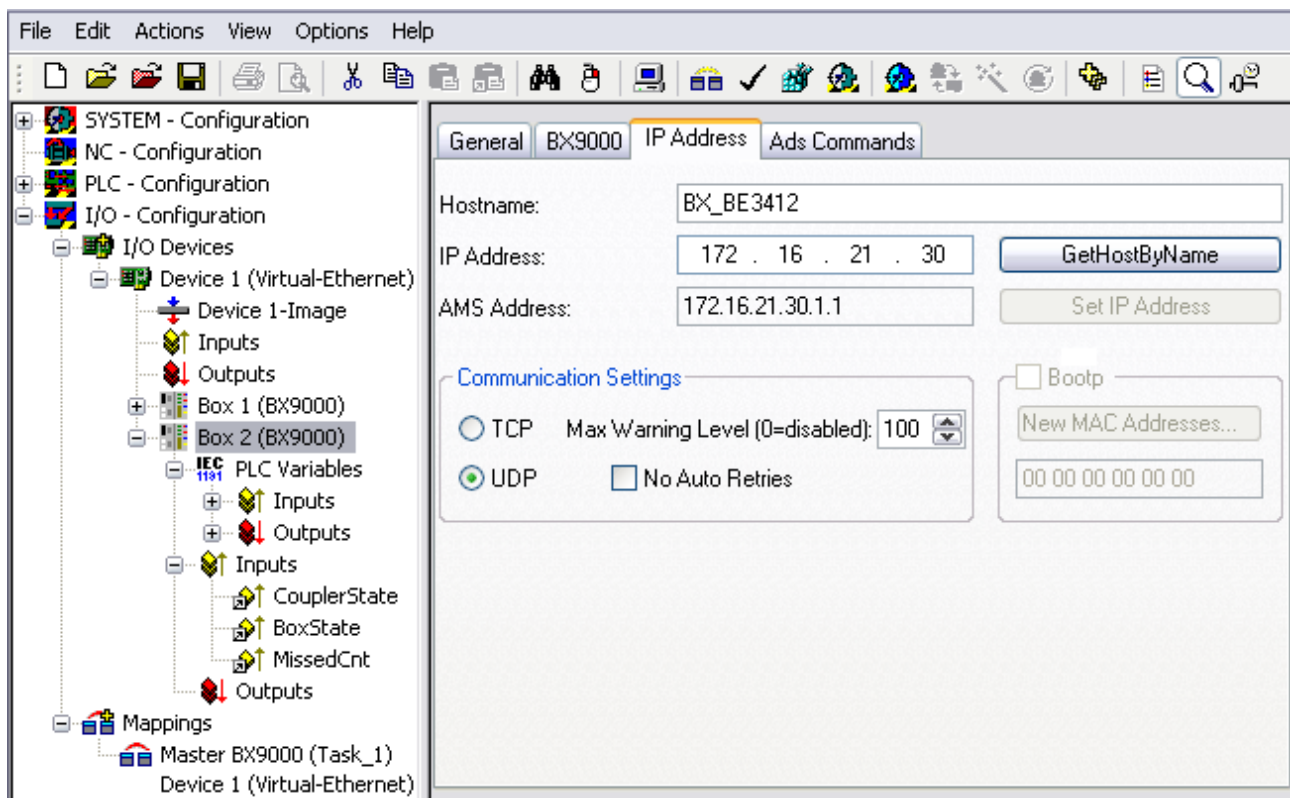


Abb. 90: IP-Adresse des BX9000 im TwinCAT System manager

PLC Control

Beim Neustart des TwinCAT PLC Control fragt TwinCAT nach der Zielplattform, d.h. auf welchem Gerät später das Anwender-Programm laufen soll. TwinCAT bietet als Steuerung zwei Zielplattformen, den PC oder den Busklemmen-Controller.

Für die Übertragung zum Busklemmen-Controller stehen Ihnen zwei Wege zur Verfügung:

- AMS für BCxx00 (Busklemmen Controller ohne Online Change)
- AMS für BCxx50 und BX (Busklemmen Controller mit Online Change)
- BC seriell, das serielle Kabel für die Kommunikation über die RS232-Schnittstelle [► 116] des PCs und die Programmierschnittstelle des Busklemmen-Controllers
- Für den BC9191 wählen Sie "BCxx50 or BX via AMS" aus

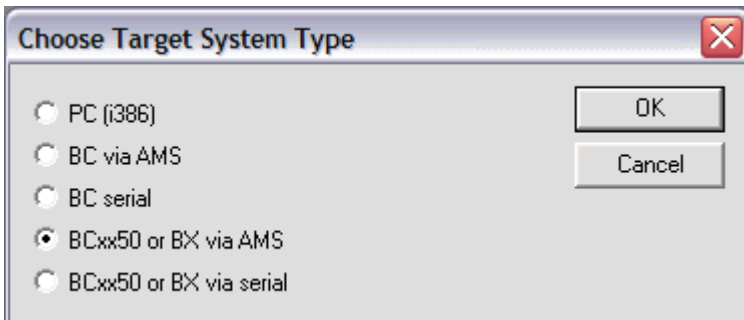


Abb. 91: Auswahl des Datenübertragungswegs - AMS

Wählen Sie nach Erstellung Ihres Programms unter der Symbolleiste *Online* das Zielsystem aus. Hierzu muss TwinCAT gestartet sein. Im Beispiel ist dies die Ethernet-Karte mit der Box 1 und die Run-Time 1 des Busklemmen-Controllers.

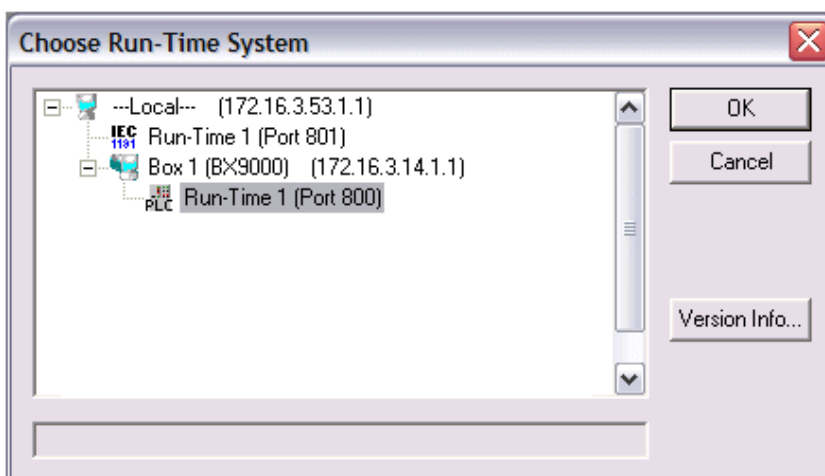


Abb. 92: Auswahl des Zielsystems

5.14.2 Programmübertragung über die serielle Schnittstelle

Jeder Busklemmen-Controller kann über die RS232-Schnittstelle des PCs programmiert werden.

Wählen Sie im TwinCAT PLC Control die serielle Schnittstelle an.

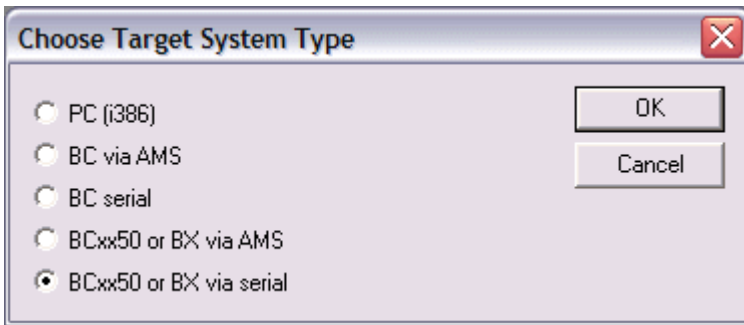


Abb. 93: Auswahl des Datenübertragungswegs - Serielle Schnittstelle

Unter Online/Kommunikationsparameter finden Sie im PLC Control die Einstellungen zur seriellen Schnittstelle, Portnummer, Baud-Rate usw.

Der Busklemmen Controller benötigt folgende Einstellung:

- Baud-Rate: 9600/19200/38400/57600 Baud (automatische Baudratenerkennung)
- Stop Bits: 1
- Parity: Gerade

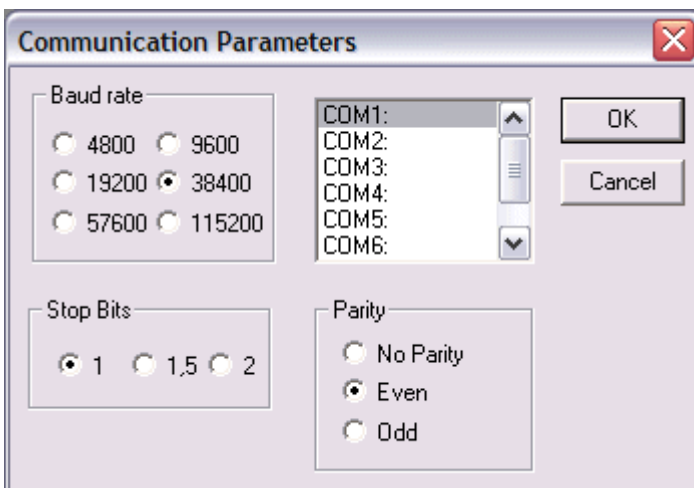


Abb. 94: Parametrierung der seriellen Schnittstelle

Programmübertragung über die serielle Schnittstelle per ADS

Der Busklemmen-Controller kann über die RS232-Schnittstelle des PC programmiert werden. Bevor Sie damit arbeiten können muss der Busklemmen-Controller dem TwinCAT bekannt gemacht werden (siehe [serielles ADS](#) [► 56]).

Wählen Sie im TwinCAT PLC-Control die ADS-Verbindung an.

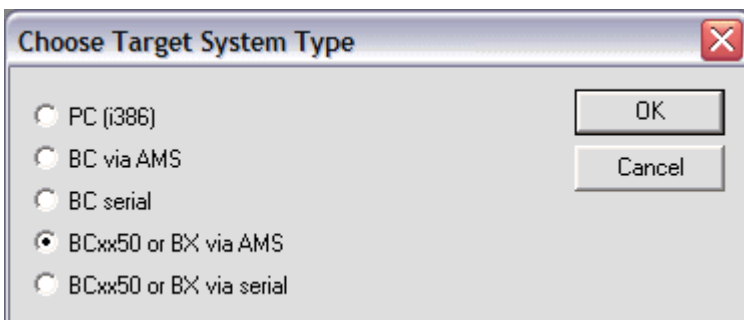


Abb. 95: Auswahl des Datenübertragungswegs - AMS

Im PLC Control kann man sich unter *Online/Communication Parameters...* einwählen.

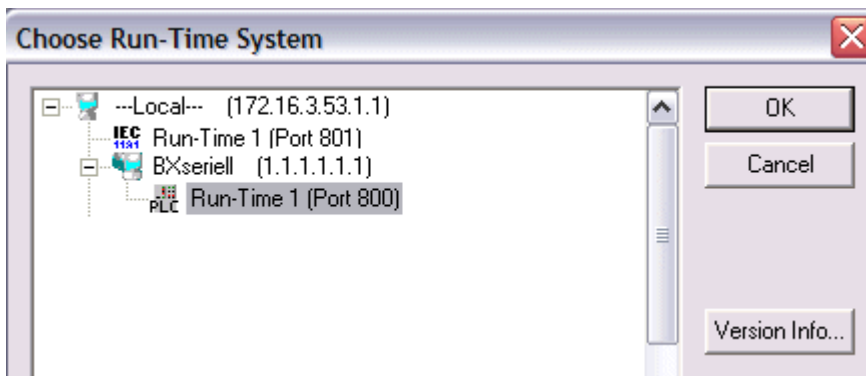


Abb. 96: Auswahl des Gerätes

5.15 Prozessabbild

5.15.1 Feldbusprozessabbild

Es gibt zwei Arten von Konfiguration auf den BX-Controller. Welche von beiden aktiv ist können sie auf den Display des BX auslesen nachdem der BX gestartet ist.

Default Config

In der Default Config liegen die Feldbusdaten ab der Adresse 1000. (%IB1000... und %QB1000).

TwinCAT Config

In der TwinCAT Config sind die Feldbusdaten nicht auf bestimmte Adressen gebunden sondern in Abhängigkeit wie diese Verknüpft worden sind.

Das Feldbusprozessabbild ist unabhängig vom verwendeten Ethernet Protokoll.

6 Ethernet

6.1 Systemvorstellung

6.1.1 Ethernet

Ethernet wurde ursprünglich von DEC, Intel und Xerox (als DIX-Standard) für die Datenübertragung zwischen Bürogeräten entwickelt. Heute versteht man darunter meist die Spezifikation *IEEE 802.3 CSMA/CD*, die 1985 veröffentlicht wurde. Diese Technologie ist durch ihren weltweiten Einsatz und die hohen Stückzahlen überall erhältlich und sehr preiswert. Eine Anbindung an vorhandene Netze kann so problemlos realisiert werden.

Mittlerweile gibt es die verschiedensten Übertragungsmedien: Koaxialkabel (10Base5), Lichtwellenleiter (10BaseF) oder verdrehte Zweidrahtleitung (10BaseT) mit Schirmung (STP) oder ohne Schirmung (UTP). Mit Ethernet lassen sich verschiedenen Topologien aufbauen wie Ring, Linie oder Stern.

Ethernet transportiert Ethernet-Pakete von einem Sender zu einem oder mehreren Empfängern. Diese Übertragung verläuft ohne Quittung und ohne Wiederholung von verlorenen Paketen. Für die sichere Daten-Kommunikation stehen Protokolle wie TCP/IP zu Verfügung, die auf Ethernet aufsetzen.

MAC-ID

Sender und Empfänger von Ethernet-Paketen werden über die MAC-ID adressiert. Die MAC-ID ist ein 6 Byte großer Identifikations-Code, der eindeutig, d.h. für jedes Ethernet-Gerät weltweit unterschiedlich ist. Die MAC-ID besteht aus zwei Teilen. Der erste Teil (d.h. die ersten 3 Byte) ist eine Herstellerkennung. Die Firma Beckhoff hat die Kennung 00 01 05. Die nächsten 3 Byte werden durch den Hersteller vergeben und entsprechen einer eindeutigen Seriennummer. Die MAC-ID kann zum Beispiel beim BootP-Protokoll zum Einstellen der TCP/IP-Nummer verwendet werden. Dafür wird ein Telegramm zum entsprechenden Knoten geschickt, das die Informationen wie Name oder TCP/IP-Nummer beinhaltet. Sie können die MAC-ID mit der Konfigurationssoftware KS2000 auslesen.

Internet-Protokoll (IP)

Die Grundlage der Datenkommunikation ist das Internet-Protokoll (IP). IP transportiert Datenpakete von einem Teilnehmer zu einem anderen, der sich im gleichen oder in einem anderen Netz befinden kann. IP kümmert sich dabei um das Adress-Management (Finden und Zuordnen der MAC-IDs), die Segmentierung und die Pfadsuche (Routing). Wie das Ethernet-Protokoll gewährleistet auch IP keinen gesicherten Transport der Daten; Datenpakete können verloren gehen oder in ihrer Reihenfolge vertauscht werden.

Für einen standardisierten, gesicherten Informationsaustausch zwischen beliebig vielen verschiedenen Netzwerken wurde TCP/IP entwickelt. Dabei ist TCP/IP weitgehend unabhängig von der verwendeten Hard- und Software. Oftmals als ein Begriff verwendet, handelt es sich hierbei um mehrere aufeinander aufgesetzte Protokolle: z. B. IP, TCP, UDP, ARP und ICMP.

Transmission Control Protocol (TCP)

Das auf IP aufsetzende Transmission Control Protocol (TCP) ist ein verbindungsorientiertes Transport-Protokoll. Es umfasst Fehlererkennungs- und Behandlungsmechanismen. Verlorene Telegramme werden wiederholt.

User Datagram Protocol (UDP)

UDP ist ein verbindungsloses Transport-Protokoll. Es gibt keine Kontrollmechanismen beim Datenaustausch zwischen Sender und Empfänger. Dadurch resultiert eine schneller Verarbeitungsgeschwindigkeit als zum Beispiel bei TCP. Eine Prüfung ob das Telegramm angekommen ist muss vom übergeordneten Protokoll durchgeführt werden.

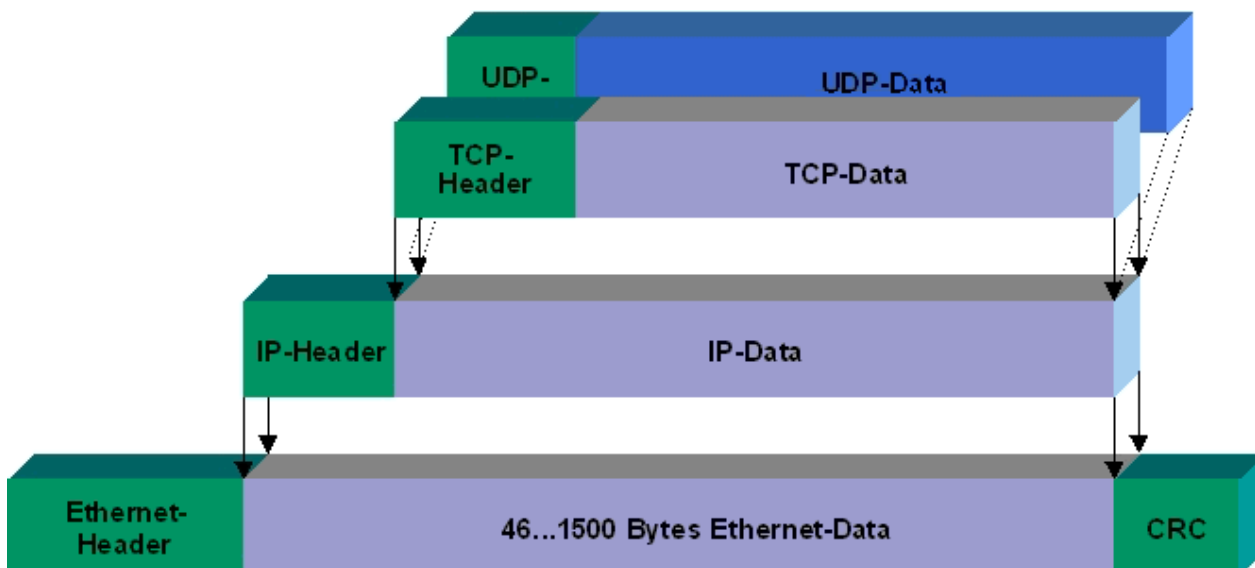


Abb. 97: User Datagram Protocol (UDP)

Auf TCP/IP und UDP/IP aufsetzende Protokolle

Auf TCP/IP bzw. UDP können folgende Protokolle aufsetzen:

- ADS
- ModbusTCP

Beide Protokolle sind parallel auf dem Buskoppler implementiert, so dass für die Aktivierung der Protokolle keine Konfiguration nötig ist.

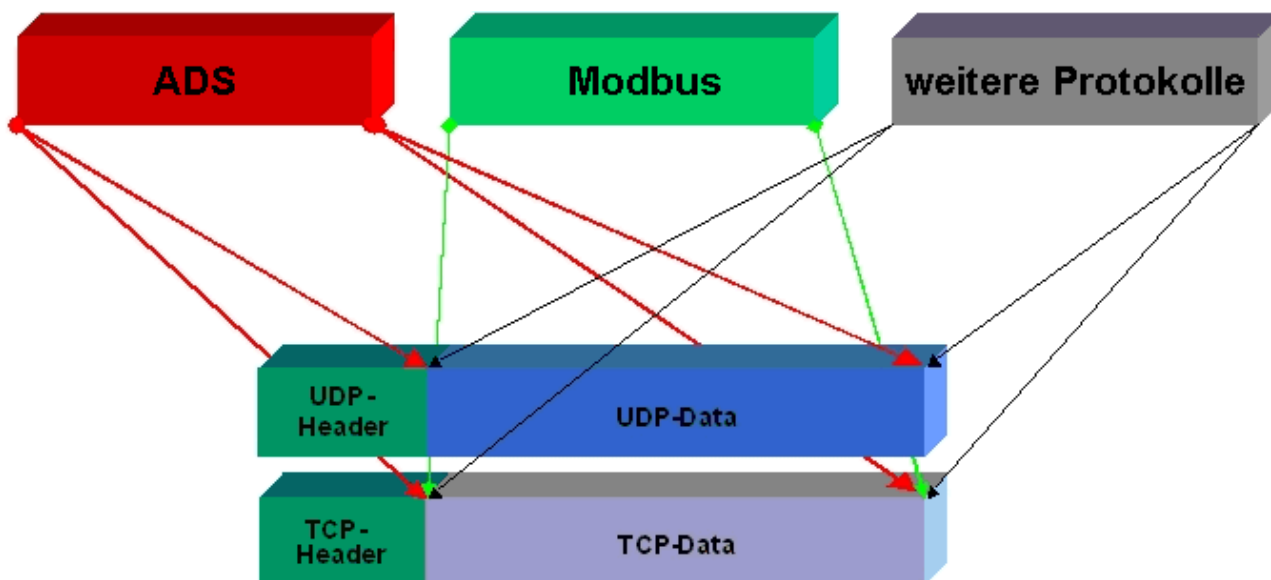


Abb. 98: Auf TCP/IP und UDP/IP aufsetzende Protokolle

ADS setzt wahlweise auf TCP oder UDP auf, während ModbusTCP stets auf TCP/IP basiert.

6.2 ModbusTCP

6.2.1 ModbusTCP-Protokoll

Das Ethernet-Protokoll wird über die MAC-ID adressiert. Der Anwender braucht sich meist um diese Adresse nicht zu kümmern. Die IP-Nummer ist 4 Byte groß und muss vom Anwender auf dem Buskoppler und in der Anwendung parametrisiert werden. Der TCP-Port ist bei ModbusTCP auf 502 festgelegt. Die UNIT ist bei ModbusTCP frei wählbar und braucht vom Anwender nicht konfiguriert werden.

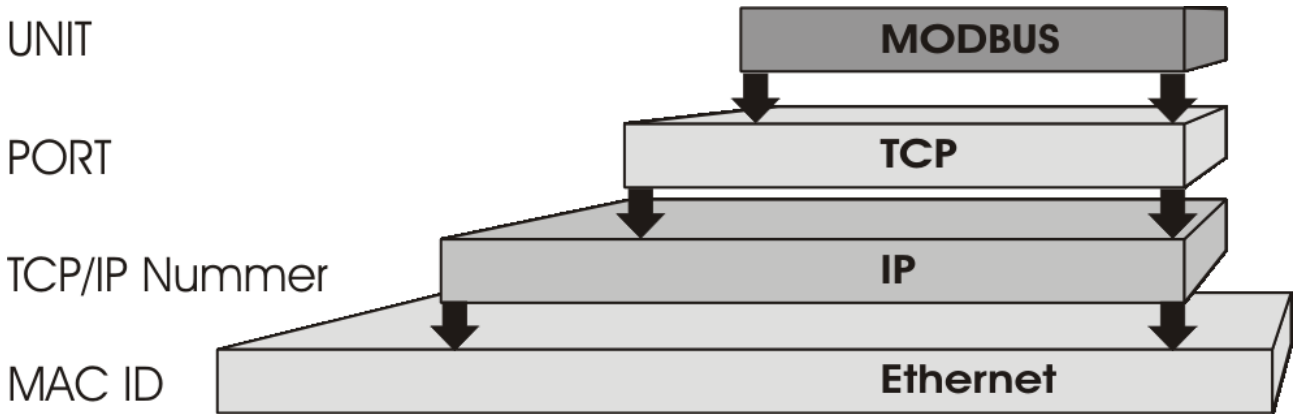


Abb. 99: ModbusTCP-Protokoll

TCP-Port-Nummer

Die TCP-Port-Nummer von ModbusTCP wurde auf den Wert 502 standardisiert.

Modbus-Unit

Die Unit wird vom Slave zurückgegeben.

ModbusTCP-Protokoll

Byte	Name	Beschreibung
0	Transaction identifier	wird vom Slave zurückgesendet
1	Transaction identifier	wird vom Slave zurückgesendet
2	Protocol identifier	immer 0
3	Protocol identifier	immer 0
4	Length field	0 (wenn die Nachricht kleiner 256 Byte ist)
5	Length field	Anzahl der folgenden Bytes
6	UNIT identifier	wird vom Slave zurückgegeben
7	Modbus	es folgt das Modbus-Protokoll beginnend mit der Funktion

6.2.2 Modbus TCP-Interface

Adresse		Beschreibung		
0x0000 0x00FF		Prozessdaten-Interface Eingänge		
0x0800 0x08FF		Prozessdaten-Interface Ausgänge		
0x1000 0x1006	Read only	Buskopplerkennung		
0x100A		2 Byte SPS-Interface		
0x100B		Busklemmendiagnose		
0x100C		Buskoppler Status		
0x1010		Prozessabbildlänge in Bit, analoge Ausgänge (ohne SPS Variablen)		
0x1011		Prozessabbildlänge in Bit, analoge Eingänge (ohne SPS Variablen)		
0x1012		Prozessabbildlänge in Bit, digitale Ausgänge		
0x1013		Prozessabbildlänge in Bit, digitale Eingänge		
0x1020		Watchdog, aktuelle Zeit in [ms]		
0x110A		Read/Write	2 Byte SPS Interface	
0x110B			Busklemmendiagnose	
0x1120			Watchdog vordefinierte Zeit in [ms] (Default: 1000)	
0x1121	Watchdog Reset Register			
0x1122	Art des Watchdogs		1	Telegramm Watchdog (Default)
			0	Schreibtelegramm Watchdog
0x1123**	ModbusTCP Mode**		1	Fast Modbus
		0	Normal Modbus (Default)	
0x4000* 0x47FF		Merkerbereich (%MB..)*		

* alle Busklemmen Controller BC9xx0 und BX9000

** für BC9x00 ab Firmware B7 und BK9000 ab Firmware B5 und alle nicht aufgeführten BK9xxx und BC/BX9xxx

Watchdog

Der Watchdog ist im Auslieferungszustand aktiviert. Nach dem ersten Schreibtelegramm wird der Watchdog scharf geschaltet und bei jedem empfangenden Telegramm dieses Teilnehmers getriggert. Andere Teilnehmer haben auf den Watchdog keinen Einfluss. Eine zweite Möglichkeit, die eine schärfere Bedingung des Watchdogs darstellt, ist, dass der Watchdog nur nach jedem Schreibtelegramm getriggert wird. Dafür Schreiben Sie in das Register 0x1122 eine Null (Default "1").

Der Watchdog kann deaktiviert werden in dem im Offset 0x1120 eine Null geschrieben wird. Das Watchdog Register darf nur dann beschrieben werden, wenn der Watchdog noch nicht aktiv ist. Die Daten in diesem Register bleiben gespeichert.

Watchdog-Register

Sollte der Watchdog auf Ihren Slave abgelaufen sein können sie diesen durch ein zweimaliges beschreiben des Registers 0x1121 zurücksetzen. Dazu muss folgendes in das Register geschrieben werden: 0xBECF 0xAFFE. Dies kann mit der Funktion 6 oder der Funktion 16 geschehen.

Statusregister des Buskopplers

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	FB	-	-	-	-	-	-	-	-	-	-	-	-	-	CNF	KB

Legende

Bit	Name	Wert	Beschreibung
15	FB	1 _{bin}	Feldbusfehler, Watchdog abgelaufen
14...2	-	-	reserviert
1	CNF	1 _{bin}	Buskoppler Konfigurationsfehler
0	KB	1 _{bin}	Busklemmenfehler

ModbusTCP Mode

Der Fast Modbus Mode sollte nur in kleinen lokalen Netzwerken eingesetzt werden. In der Default Einstellung ist der Fast ModbusTCP nicht aktiviert. Werden Probleme mit dieser Kommunikationsart festgestellt, ist der Buskoppler auf die "normale" ModbusTCP Kommunikation umzustellen. Per Modbus Interface Offset 0x1123 ist dieser Mode einzustellen. Nach der Änderung ist ein Reset (zum Beispiel mit der ModbusTCP Funktion 8) des Buskopplers notwendig. Im Fast Modbus Mode ist es nicht gestattet, mehr als ein Modbus Dienst in einem Ethernet Frame zu versenden.

2 Byte SPS-Interface

Mit dem 2 Byte SPS-Interface können Register der komplexen Klemmen sowie Register des Busklemmen-Controllers gelesen bzw. beschrieben werden. Die Register der komplexen Klemmen sind in der Dokumentation zur jeweiligen Klemme beschreiben. Über die Register des Buskopplers können z. B. K-Busdiagnosedaten, der Klemmenaufbau oder Zykluszeiten gelesen sowie die programmierte Konfiguration beschrieben werden. Weiterhin kann darüber auch ein manueller K-Bus-Reset durchgeführt werden. Das 2 Byte SPS-Interface benötigt je zwei Bytes in den Ein- und Ausgangsdaten, über die ein spezielles Protokoll abgewickelt wird. Eine Beschreibung des 2 Byte SPS-Interface, der verfügbaren Register im Buskopplers sowie Funktionsbausteine für verschiedene SPS-Systeme, die das 2 Byte SPS-Interface unterstützen, kann auf Anfrage geliefert werden.

2 Byte Diagnose-Interface

Die Fehlermeldungen der Klemmen können mit dem 2 Byte Diagnose-Interface gesendet werden. Dazu ist aber die K-Busdiagnose zu aktivieren. Das 2 Byte Diagnose-Interface belegt je zwei Bytes in den Ein- und Ausgangsdaten, über die ein spezielles Protokoll durchgeführt wird. Eine Beschreibung des 2 Byte Diagnose-Interfaces kann auf Anfrage geliefert werden.

6.2.3 Fehlerantwort des ModbusTCP Slaves (BK9000, BX/BC9xx0, BC9191, IP/ILxxxx-B/C900, EK9000)

Wenn der Anwender dem Slave eine Anforderung oder Mitteilung sendet, die der Koppler nicht versteht, antwortet der Slave mit einer Fehlermitteilung. Diese Antwort enthält die Funktion und den Fehler-Code. Der Funktionsrückgabewert wird mit 0x80 addiert.

Code	Name	Bedeutung
1	ILLEGAL FUNKTION	Nicht implementierte Modbus-Funktion
2	ILLEGAL DATA ADDRESS	Ungültige Adresse oder Länge
3	ILLEGAL DATA VALUE	Ungültige Parameter - Diagnose-Funktionen - falsches Register
4	SLAVE DEVICE ERROR	Watchdog- oder K-Bus-Fehler EK9000: E-Bus Fehler
6	SLAVE DEVICE BUSY	Es wird schon von einem anderen IP-Teilnehmer Ausgangs-Daten empfangen

6.2.4 ModbusTCP-Funktionen

6.2.4.1 Read holding register (Funktion 3)

Mit der Funktion *Read holding register* können die Ein- und Ausgangsworte und die Register gelesen werden. Eingänge ab dem Offset 0 - 0xFF und Ausgänge ab den Offset 0x800 - 0x8FF und bei den Steuerungen (BC, BX) der Merker Bereich ab den Offset 0x4000.

In diesem Beispiel werden die ersten zwei analogen Ausgänge (oder 2 Ausgangsworte) gelesen. Die analogen Ausgänge (oder Ausgangsworte) beginnen beim Offset 0x800. Die Länge bezeichnet die Anzahl, der zu lesenden Kanäle (oder der zu lesenden Worte).

Anfrage (Query)

Byte Name	Beispiel
Funktions-Code	3
Start-Adresse high	8
Start-Adresse low	0
Anzahl high	0
Anzahl low	2

Der Feldbus-Koppler antwortet mit dem Byte Count 4, d.h. es kommen 4 Byte Daten zurück. Die Anfrage waren zwei Analogkanäle, die auf zwei Worte aufgeteilt sind. Im analogen Ausgangsprozessabbild hat der erste Kanal einen Wert von 0x3FFF und der zweite Kanal einen Wert von 0x0.

Antwort (Response)

Byte Name	Beispiel
Funktions-Code	3
Byte Count	4
Daten 1 High-Byte	63
Daten 1 Low-Byte	255
Daten 2 High-Byte	0
Daten 2 Low-Byte	0

6.2.4.2 Read input register (Funktion 4)

Die Funktion *Read input register* liest wort-orientiert die Eingänge aus.

In diesem Beispiel werden die ersten zwei analogen Eingänge (oder die erste 2 Eingangsworte) gelesen. Die analogen Eingänge (oder Eingangsworte) beginnen bei einem Offset von 0x0000. Die Länge bezeichnet die Anzahl der zu lesenden Worte. Eine KL3002 z. B. hat zwei Worte Eingangsdaten, daher ist die einzugebende Länge bei *Anzahl low* zwei.

Anfrage (Query)

Byte Name	Beispiel
Funktions-Code	4
Start-Adresse high	0
Start-Adresse low	0
Anzahl high	0
Anzahl low	2

Der Feldbus-Koppler antwortet mit dem Byte Count 4, d.h. es kommen vier Byte Daten zurück. Die Anfrage waren zwei analog Kanäle, die jetzt auf zwei Worte aufgeteilt werden. Im analogen Eingangsprozessabbild hat der erste Kanal einen Wert von 0x0038 und der zweite Kanal einen Wert von 0x3F1B.

Antwort (Response)

Byte Name	Beispiel
Funktions-Code	4
Byte Count	4
Daten 1 High-Byte	0
Daten 1 Low-Byte	56
Daten 2 High-Byte	63
Daten 2 Low-Byte	11

6.2.4.3 Preset single register (Funktion 6)

Mit der Funktion *Preset single register* kann auf das Ausgangs- oder Merkerprozessabbild (nur bei den Steuerungen) und das [Modbus-TCP-Interface \[► 122\]](#) zugegriffen werden.

Bei der Funktion 6 wird das erste Ausgangswort beschrieben. Die Ausgänge beginnen bei einem Offset von 0x0800. Auch hier beschreibt der Offset immer ein Wort. Das heißt, dass der Offset 0x0803, das 4. Wort auf dem Ausgangsprozessabbild ist.

Anfrage (Query)

Byte Name	Beispiel
Funktions-Code	6
Start-Adresse high	8
Start-Adresse low	0
Daten high	63
Daten low	255

Der Feldbus-Koppler antwortet mit dem gleichen Telegramm und der Bestätigung der empfangenen Werte.

Antwort (Response)

Byte Name	Beispiel
Funktions-Code	6
Start-Adresse high	8
Start-Adresse low	0
Daten high	63
Daten low	255

6.2.4.4 Preset multiple register (Funktion 16)

Mit der Funktion *Preset multiple register* können mehrere Ausgänge beschrieben werden. In diesem Beispiel werden die ersten zwei analogen Ausgangsworte beschrieben. Die Ausgänge beginnen bei einem Offset von 0x0800. Hier beschreibt der Offset immer ein Wort. Der Offset 0x0003 schreibt ab dem vierten Wort auf das Ausgangsprozessabbild. Die Länge gibt die Anzahl der Worte an und der *Byte Count* setzt sich aus den zu schreibenden Bytes zusammen.

Beispiel: 4 Worte - entsprechen 8 Byte Count

Die Datenbytes enthalten die Werte für die analogen Ausgänge. In diesem Beispiel sind es zwei Worte, die zu beschreiben sind. Das erste Wort mit dem Wert 0x7FFF und das zweite Wort mit dem Wert 0x3FFF.

Anfrage (Query)

Byte Name	Beispiel
Funktions-Code	16
Start-Adresse high	8
Start-Adresse low	0
Länge high	0
Länge low	2
Byte Count	4
Daten 1 Byte 1	127
Daten 1 Byte 2	255
Daten 2 Byte 1	63
Daten 2 Byte 2	255

Antwort (Response)

Der Koppler antwortet mit der Start-Adresse und der Länge der gesendeten Worte.

Byte Name	Beispiel
Funktions-Code	16
Start-Adresse high	8
Start-Adresse low	0
Länge high	0
Länge low	2

6.2.4.5 Read / write registers (Funktion 23)

Mit der Funktion *Read / write registers* können mehrere analoge Ausgänge beschrieben und in einem Telegramm mehrere analoge Eingänge gelesen werden. In diesem Beispiel werden die ersten zwei analogen Ausgangsworte beschrieben und die ersten zwei analogen Eingänge gelesen. Die analogen Ausgänge beginnen beim Offset 0x0800 und die Eingänge ab dem Offset 0x0000. Hier beschreibt der Offset immer ein Wort. Der Offset 0x0003 schreibt ab dem 4. Wort auf das Ausgangsprozessabbild. Die Länge gibt die Anzahl der Worte an und der *Byte Count* setzt sich aus den zu schreibenden Bytes zusammen. Beispiel: 4 Worte - entsprechen 8 Byte Count

Die Datenbytes enthalten die Werte für die analogen Ausgänge. In diesem Beispiel sind es zwei Worte, die zu beschreiben sind. Das erste Wort mit dem Wert 0x3FFF und das zweite Wort mit dem Wert 0x7FFF.

Anfrage (Query)

Byte Name	Beispiel
Funktions-Code	23
Lesen Start-Adresse high	0
Lesen Start-Adresse low	0
Lesen Länge high	0
Lesen Länge low	2
Schreiben Start-Adresse high	8
Schreiben Start-Adresse low	0
Schreiben Länge high	0
Schreiben Länge low	2
Byte Count	4
Daten 1 high	63
Daten 1 low	255
Daten 2 high	127
Daten 2 low	255

Antwort (Response)

Der Koppler antwortet mit der Start-Adresse und der Länge der übertragenen Bytes im *Byte Count*. Es folgen die Dateninformationen. In diesem Beispiel steht im ersten Wort eine 0x0038 und im zweiten Wort eine 0x3F0B.

Byte Name	Beispiel
Funktions-Code	23
Byte Count	4
Daten 1 high	0
Daten 1 low	56
Daten 2 high	63
Daten 2 low	11

6.3 ADS-Kommunikation

6.3.1 ADS-Kommunikation

Das ADS-Protokoll (ADS: Automation Device Specification) ist eine Transportschicht innerhalb des TwinCAT-Systems. Es ist für den Datenaustausch der verschiedenen Software-Module entwickelt worden, zum Beispiel für die Kommunikation zwischen der NC und der PLC. Mit diesem Protokoll haben Sie die Freiheit von jedem Punkt im TwinCAT mit anderen Tools kommunizieren zu können. Wird die Kommunikation zu anderen PCs oder Geräten benötigt, setzt das ADS-Protokoll auf TCP/IP auf. Somit ist es in einem vernetzten System möglich, alle Daten von einem beliebigen Punkt aus zu erreichen.

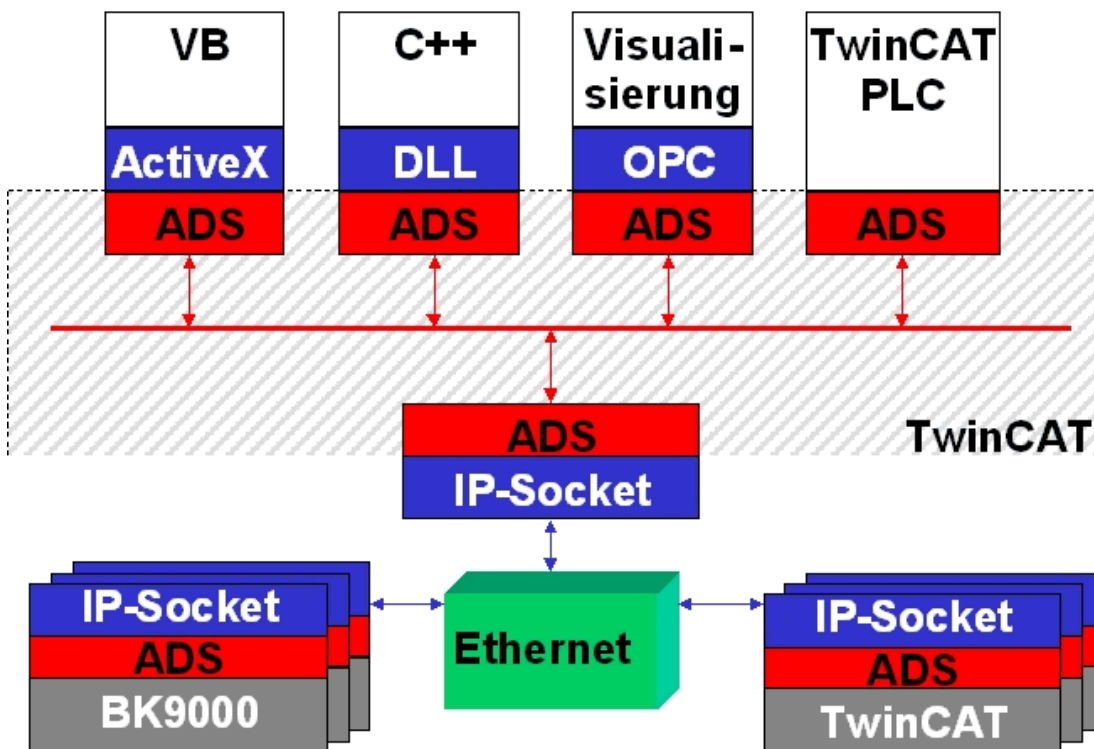


Abb. 100: Das ADS-Protokoll als Transportschicht innerhalb des TwinCAT-Systems

Das ADS-Protokoll wird auf das TCP/IP- oder UDP/IP-Protokoll aufgesetzt. Es ermöglicht dem Benutzer innerhalb des Beckhoff-Systems über nahezu beliebige Verbindungswege mit allen angeschlossenen Geräten zu kommunizieren und diese zu parametrieren. Außerhalb des Beckhoff-Systems stehen verschiedene Wege offen, um mit anderen Software-Tools Daten auszutauschen.

Software-Schnittstellen

ADS-OCX

Das ADS-OCX ist eine Active-X-Komponente und bietet eine Standardschnittstelle zum Beispiel zu Visual Basic, Delphi, usw.

ADS-DLL

Sie können die ADS-DLL (DLL: Dynamic Link Library) in Ihr C-Programm einbinden.

OPC

Die OPC-Schnittstelle ist eine genormte Standardschnittstelle für die Kommunikation in der Automatisierungstechnik. Beckhoff bietet hierfür einen OPC-Server an.

6.3.2 ADS-Protokoll

Die ADS-Funktionen bieten die Möglichkeit, direkt vom PC auf Informationen des Buskopplers zuzugreifen. Dafür können ADS-Funktionsbausteine im TwinCAT PLC Control verwendet werden. Die Funktionsbausteine sind in der Bibliothek *TcSystem.lib* enthalten. Genauso ist es möglich, die ADS-Funktionen von AdsOCX, ADSDLL oder OPC aufzurufen. Über die ADS-Portnummer 300 sind alle Daten und über die ADS-Portnummer 100 Zugriffe auf die Register des Buskopplers und der Klemmen möglich.

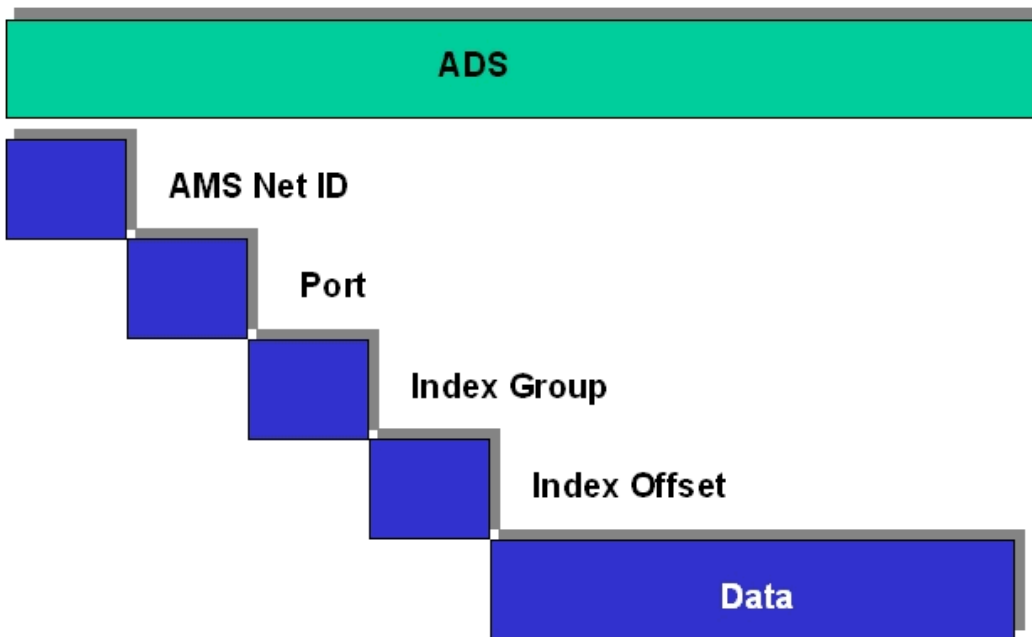


Abb. 101: Aufbau des ADS-Protokolls

AMSNNetID

Die AMSNetID beschreibt das anzusprechende Gerät. Diese wird aus der eingestellten TCP/IP-Adresse und zusätzlichen 2 Byte erstellt. Diese zusätzlichen 2 Byte bestehen aus "1.1" und sind nicht veränderbar.

Beispiel:

IP-Adresse 172.16.17.128

AMSNNetID 172.16.17.128.1.1

Port-Nummer

Die Portnummer unterscheidet im angeschlossenen Gerät Unterelemente.

Port 100: Registerzugriff

Port 300: Prozessdaten Feldbus

Port 800: lokale Prozessdaten (nur BC90x0, C900)

Index Group

Die Index Group unterscheidet innerhalb eines Ports verschiedene Daten.

Index Offset

Gibt den Offset an, ab welchem Byte gelesen oder geschrieben werden soll.

Len

Gibt die Länge der Daten in Byte an, die gelesen bzw. geschrieben werden sollen.

TCP-Port-Nummer

Die TCP-Port-Nummer beträgt für das ADS-Protokoll 48898 oder 0xBF02.

6.3.3 ADS-Dienste

● Nutzdaten eines ADS-Telegramms



Die Nutzdaten eines ADS-Telegramms betragen maximal 1900 Byte.

Prozessdaten-Port 300

Der Zugriff auf die Feldbusdaten erfolgt über die ADS-Portnummer 300. Diese Daten werden von einem Watchdog überwacht. Sollte innerhalb von 1000 ms kein weiteres Telegramm eintreffen, werden die Ausgänge in den sicheren Zustand geschaltet.

Index Group	Bedeutung	Index Offset (Wertebereich)
0xF020	Input - Eingänge	0...511
0xF030	Output - Ausgänge	0...511

Lokales Prozessabbild Port 800 (nur BC9000, C900)

Im lokalen Prozessabbild können Daten gelesen und geschrieben werden. Sollten Ausgänge geschrieben werden, muss darauf geachtet werden, dass diese von der lokalen SPS nicht verwendet werden, da die lokale Steuerung diese Werte überschreibt. Die Daten sind nicht an einen Watchdog gebunden und dürfen daher nicht für Ausgänge verwendet werden, die im Fehlerfall ausgeschaltet werden müssen.

Index Group	Bedeutung	Index Offset (Wertebereich)
0xF020	Input - Eingänge	0...511
0xF030	Output - Ausgänge	0...511
0x4020	Merker (nur BC9000, C900)	0...4096

Dienste des ADS

AdsServerAdsStateB7

Datentyp (read only)	Bedeutung
String	Start - die lokale PLC läuft Stop - die lokale PLC ist im Stop

AdsServerDeviceStateB7

Datentyp (read only)	Bedeutung
INT	0 - Start - die lokale PLC läuft 1 - Stop - die lokale PLC ist im Stop

AdsServerTypeB7

Datentyp (read only)	Bedeutung
String	Coupler_PLC

ADSWriteControlBA

Datentyp (write only)	Bedeutung
NetID	Net ID des BC9000, C900
Port	800
ADSSTATE	5 - RUN / 6 - STOP
DEVSTATE	0
LEN	0
SRCADDR	0
WRITE	positive Flanke startet den Baustein
TMOUT	zum Beispiel: t#1000 ms

Azyklische Datenkommunikation

Bei der azyklischen Datenkommunikation muss man beachten, dass ein TCP/IP-Verbindungsaufbau ca. 20 bis 30 ms benötigt. Nach erfolgreichem Verbindungsaufbau werden die ADS-Daten gesendet oder gelesen. Ein ADS Read von 1000 Byte dauert ca. 50 ms. Werden 10 sec lang keine Daten übertragen, wird die TCP/IP Verbindung vom BC/BK9000, B/C900 getrennt.

Register Port 100

Die ADS-Portnummer ist beim BK/BC9000, B/C900 für die Register-Kommunikation fest vorgegeben und beträgt 100.

Index Group	Index Offset (Wertebereich)		Bedeutung
	Hi-Word	Lo-Word	
0	0...127	0...255	Register des Buskopplers Hi-Word Tabellenummer des Buskopplers Lo-Word Registernummer der Tabelle
1...64	0...3	1...64	Register der Busklemmen Hi-Word Kanalnummer Lo-Word Registernummer der Busklemme

Time-Out des ADS-Bausteins

Beachten Sie beim Lesen der Register, dass der Time-Out des ADS-Bausteins auf eine Zeit größer 1 Sekunde eingestellt wird.

Passwort setzen

Beachten Sie beim Schreiben auf die Register, dass das Passwort gesetzt wird (siehe Dokumentation zur entsprechenden Busklemme).

Zugriffskontrolle und IP - AMS-Net-ID Zuordnung^{B6}

Die Tabelle *AMS Net-Id* ermöglicht über AMS eine Zugriffskontrolle auf den BC9000, C900. Sobald Einträge in dieser Tabelle vorhanden sind, können nur noch die eingetragenen AMS-Teilnehmer auf den BC9000 zugreifen.

Des Weiteren wird hier explizit eine Zuordnung der AMS Net ID zur IP-Adresse des Knotens gemacht.

Die Tabelle *AMS Net-Id* kann mit ADS Write Kommandos gefüllt werden:

Es sind maximal 10 Einträge möglich.

Aufbau der Struktur

AMS Net ID	Größe
AMS Net ID	6 Byte
IP-Adresse	4 Byte
Reserve	2 Byte
Reserve	4 Byte
Reserve	4 Byte

Zugriff erfolgt über Portnummer: 10.000

Indexgroup: 700

Index Offset (Write)	Bemerkung	Daten
0	Hinzufügen eines Eintrags	Datenstruktur 20 Byte
1	Löschen eines Eintrags	-
2	Löschen aller Einträge	-
10	Speichern der Tabelle im Flash	-

Index Offset (Read)	Bemerkung	Daten
0	Anzahl der Einträge	2 Byte
1..10	Eintrag n (1...10)	Datenstruktur 20 Byte

● **Erster Eintrag**

i Der erste Eintrag muss der Teilnehmer sein der in die Tabelle schreibt, da diese Einstellungen sofort gültig sind. Achten Sie darauf dass die Einstellungen alle Richtig sind. Die Tabelle kann ebenfalls gelöscht werden, wenn nur die Endklemme gesteckt ist und die DIP-Schalter 1 bis 7 auf ON gesetzt sind.

* ab Firmware B6

** ab Firmware B7

)² ab Firmware BA

Siehe Beispiel.

7 Fehlerbehandlung und Diagnose

7.1 Diagnose

Zustand des Ethernet

In vielen Fällen ist es wichtig zu wissen ob die Kommunikation mit dem übergeordneten Master noch funktioniert. Verknüpfen Sie hierfür die Variable *FieldbusState* mit Ihrem SPS-Programm.

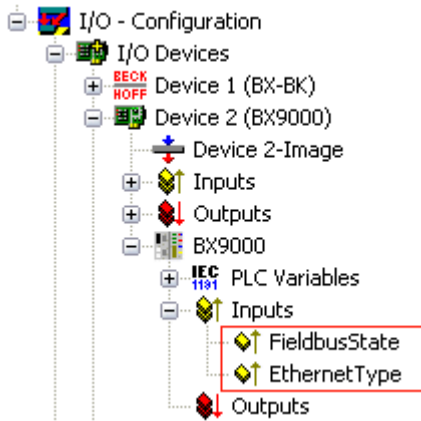


Abb. 102: Zustand des Ethernet

Fieldbus State

Fehler Nummer	Beschreibung	Abhilfe
0	kein Fehler	-
1	Watchdog Fehler	Kommunikation unterbrochen

Ethernet Typ

Hier kann man erkennen welches Ethernet Protokoll auf die SPS Variablen zugreift und damit den Watchdog aktiviert (zum Beispiel die Daten in der Default Config ab der Adresse %IB1000 und %QB1000).

Diagnose Nummer	Beschreibung	Abhilfe
0x0000	kein Protokoll greift auf die SPS Variablen zu	-
0x0001	ADS TCP	Kommunikation über ADS TCP/IP
0x0002	ADS UDP	Kommunikation über ADS UDP/IP
0x0010	ModbusTCP	Kommunikation über Modbus TCP/IP

Auslesen des Feldbusstatus per ADS

In der Default Konfiguration oder in TwinCAT Konfiguration kann der Feldbusstatus über ADSREAD ausgelesen werden (in Vorbereitung).

Parameter ADSREAD Baustein	Beschreibung
NetID	lokal - Leerstring
Port	
IndexGroup	
IndexOffset	
LEN	

Zustand des K-Bus

Sollte der interne Bus oder einer seiner Busklemmen ein Problem haben wird dies im K-Bus-State angezeigt. Eine genauere Fehlerursache kann mit einem Funktionsbaustein ausgelesen werden (in Vorbereitung). Verknüpfen Sie hierfür die Variable *K-Bus-State* mit Ihrem SPS-Programm.

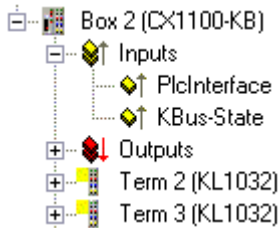


Abb. 103: Zustand des K-Bus

Fehlerbit	Beschreibung	Fehlerart
0	kein Fehler	Kein FEHLER
Bit 0	K-Bus Fehler	FEHLER
Bit 2	K-Bus wird Nachgetriggert	HINWEIS

Auslesen des K-Bus Status per ADS

In der Default Konfiguration oder in TwinCAT Konfiguration kann der Feldbusstatus über ADSREAD ausgelesen werden.

Parameter ADSREAD Baustein	Beschreibung
NetID	lokal - Leerstring
Port	1
IndexGroup	16#0006
IndexOffset	16#000C_9000
LEN	1

7.2 Diagnose-LEDs

Zur Statusanzeige besitzen die Busklemmen Controller LEDs.

BC9050

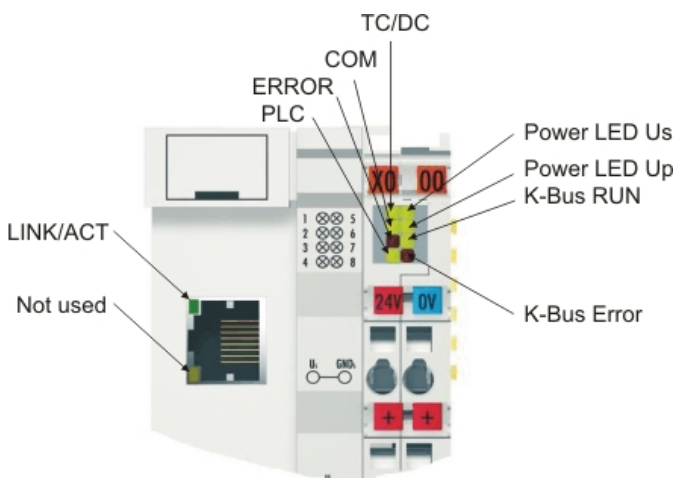


Abb. 104: BC9050 LEDs

LEDs zur Diagnose der Spannungsversorgung

LED (Power LEDs)	Bedeutung
Power LED Us aus	Buskoppler hat keine Spannung 24 V _{DC}
Power LED Up aus	Keine Spannungsversorgung 24 V _{DC} an den Powerkontakten angeschlossen

LEDs zur Diagnose des K-Busses

LED (K-Bus)	Bedeutung
K-Bus RUN	an oder blinkt - K-Bus läuft
K-Bus ERR	blinkt (siehe Fehler Code) [► 137]

LEDs zur Diagnose des Ethernets

LED (Ethernet)	Bedeutung
LINK/ACT	an - LINK vorhanden, blinkt - LINK vorhanden und Kommunikation
ERROR	blinkt - DHCP oder BootP aktiv. Es wird auf eine IP-Adresse gewartet
COM	Kommunikation mit Controller vorhanden

LEDs zur Diagnose SPS

LED (Ethernet)	Bedeutung
PLC	an - PLC läuft, blinkt - Zykluszeit wird überschritten, aus - Zykluszeit dauerhaft überschritte oder PLC im Stopp
TC/DC	an - TwinCAT Konfiguration aktiv, aus - Default Konfiguration aktiv, blinkt TwinCAT Konfiguration fehlerhaft

BC9120

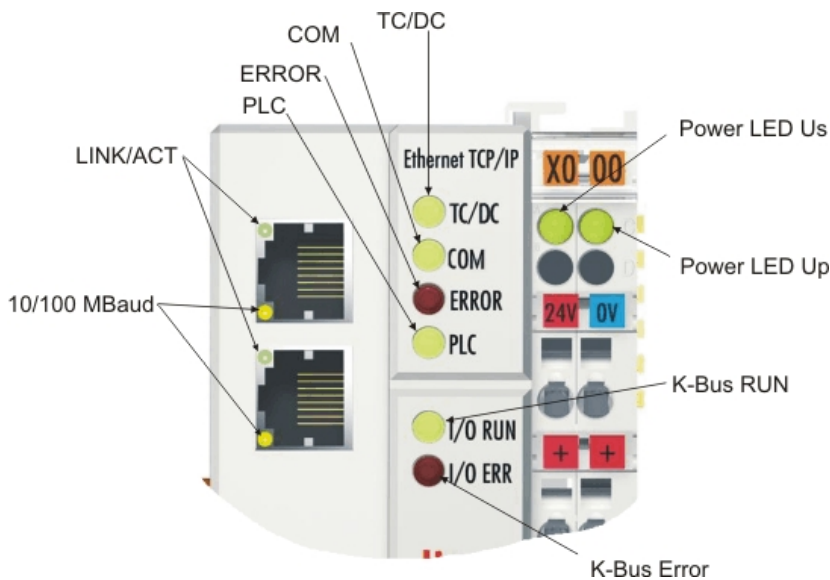


Abb. 105: LEDs BC9120

LEDs zur Diagnose der Spannungsversorgung

LED (Power LEDs)	Bedeutung
Power LED Us aus	Buskoppler hat keine Spannung 24 V _{DC}
Power LED Up aus	Keine Spannungsversorgung 24 V _{DC} an den Powerkontakten angeschlossen

LEDs zur Diagnose des K-Busses

LED (K-Bus)	Bedeutung
K-Bus RUN	an oder blinkt - K-Bus läuft
K-Bus ERR	blinkt (siehe Fehler Code [▶ 137])

LEDs zur Diagnose des Ethernets

LED (Ethernet)	Bedeutung
LINK/ACT	an - LINK vorhanden, blinkt - LINK vorhanden und Kommunikation
10/100MBaud	an - 100 MBaud, aus - 10 MBaud
ERROR	blinkt - DHCP oder BootP aktiv. Es wird auf eine IP-Adresse gewartet
COM	Kommunikation mit Controller vorhanden

LEDs zur Diagnose SPS

LED (Ethernet)	Bedeutung
PLC	an - PLC läuft, blinkt - Zykluszeit wird überschritten, aus - Zykluszeit dauerhaft überschritten oder PLC im Stopp
TC/DC	an - TwinCAT Konfiguration aktiv, aus - Default Konfiguration aktiv, blinkt TwinCAT Konfiguration fehlerhaft

Fehler-Codes zur K-Bus-Diagnose

Fehler-Code	Fehler-argument	Beschreibung	Abhilfe
-	ständiges 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 Koppler) der Fehler lokalisiert werden
1	0	EEPROM- Prüfsummenfehler	Herstellereinstellung mit der Konfigurationssoftware KS2000 setzen
	1	Überlauf im Code Buffer	Weniger Busklemmen stecken. Bei prog. Konfiguration sind zu viele Einträge in der Tabelle
	2	Unbekannter Datentyp	Software Update des Buskopplers notwendig
2	-	Reserve	-
3	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. Dies weiter durchführen, bis die defekte Busklemme lokalisiert ist.
4	0	K-Bus-Datenfehler, Bruchstelle hinter dem Buskoppler	Prüfen ob die n+1 Busklemme richtig gesteckt ist, gegebenenfalls tauschen
	n	Bruchstelle hinter Busklemme n	Kontrollieren ob die Busendklemme KL9010 gesteckt ist
5	n	K-Bus-Fehler bei Register-Kommunikation mit Busklemme n	n-te Busklemme tauschen
6	0	Fehler bei der Initialisierung	Buskoppler tauschen
	1	Interner Datenfehler	Hardware-Reset des Buskopplers (aus - und wieder einschalten)
	2	DIP-Schalter nach einem Software-Reset verändert	Hardware-Reset des Buskopplers (aus - und wieder einschalten)
	3	IP-Adresse bereits vergeben	Überprüfen Sie ob die IP Adresse bereits im Netzwerk vorhanden ist.
7	0	Hinweis: Zykluszeit wurde überschritten	Warning: Die eingestellte Zykluszeit wurde überschritten. Dieser Hinweis (blinken der LEDs) kann nur durch erneutes booten des Buskopplers gelöscht werden. Abhilfe: Zykluszeit erhöhen
9	0	Checksummenfehler im Programm-Flash	Programm erneut zum BC übertragen
	1	Falsche oder fehlerhaft Bibliothek implementiert	Entfernen Sie die fehlerhaft Bibliothek
10	n	Die Busklemme n stimmt nicht mit der Konfiguration, die beim Erstellen des Boot-Projektes existiert überein	Die n-te Busklemme überprüfen. Sollte eine n-te Busklemme gewollt eingefügt worden sein, muss das Bootprojekt gelöscht werden.
14	n	n-te Busklemme hat das falsche Format	Buskoppler erneut Starten, falls der Fehler erneut auftritt die Busklemme tauschen.
15	n	Anzahl der Busklemmen stimmt nicht mehr	Buskoppler erneut Starten, falls der Fehler erneut auftritt, Herstellereinstellung mit der Konfigurationssoftware KS2000 setzen
16	n	Länge der K-Bus-Daten stimmt nicht mehr	Buskoppler erneut Starten, falls der Fehler erneut auftritt, Herstellereinstellung mit der Konfigurationssoftware KS2000 setzen

8 Anhang

8.1 BC9xx0 - Erste Schritte

Benötigte Komponenten

- Einen BC9xx0
- KL1114
- KL2134
- KL9010
- Ein 24 V_{DC} Netzteil, Verkabelungsmaterial
- Ein Ethernet Cross Kabel bei eine 1:1 Verbindung vom PC zum BC9020 oder BC9050 (ohne Switch)
- Ein "normales" Ethernet Kabel vom PC zum BC9120

Einstellen der DIP-Schalter

Stellen Sie den DIP-Schalter auf 1 auf ON, alle anderen DIP-Schalter lassen Sie auf OFF. Es wird mit einer Festen IP-Adresse gearbeitet. Nach Einstellen der DIP-Schalter starten Sie den BC neu.

BC9050 hat dann die IP-Adresse 172.16.21.1

BC9020 hat dann die IP-Adresse 172.16.22.1

BC9120 hat dann die IP-Adresse 172.16.23.1

Stellen Sie die IP-Adresse Ihres Programmier-PCs entsprechend ein. Zum Beispiel 172.16.100.100 (SubNetMask 255.255.0.0).

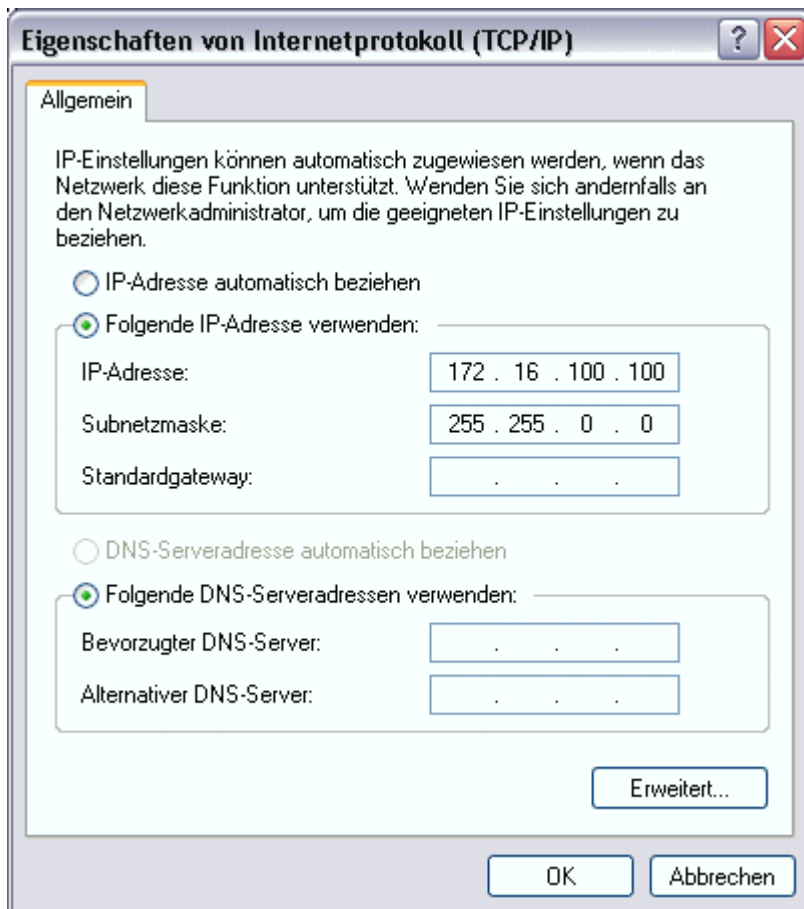


Abb. 106: Einstellen der IP-Adresse des Programmier-PCs

Prüfen Sie mit dem >Ping< Befehl ob eine TCP/IP-Kommunikation vorhanden ist. Öffnen Sie dazu die Eingabeaufforderung (DOS-Box) und tragen Sie zum Beispiel bei der Verwendung eines BC9050 folgendes ein: Ping 172.16.21.1. Ist der Ping erfolgreich können sie fortfahren. Sollte der Ping nicht beantwortet werden, ist die Kommunikation fehlerhaft. Kontrollieren sie die Ethernet Verbindung - Richtiges Kabel, ist der DIP-Schalter des Controllers richtig eingeschaltet, ist die IP-Adresse auf dem PC umgestellt.

BC mit dem System-Manager suchen

Starten Sie den System-Manager. Wählen Sie ein neues Zielsystem.

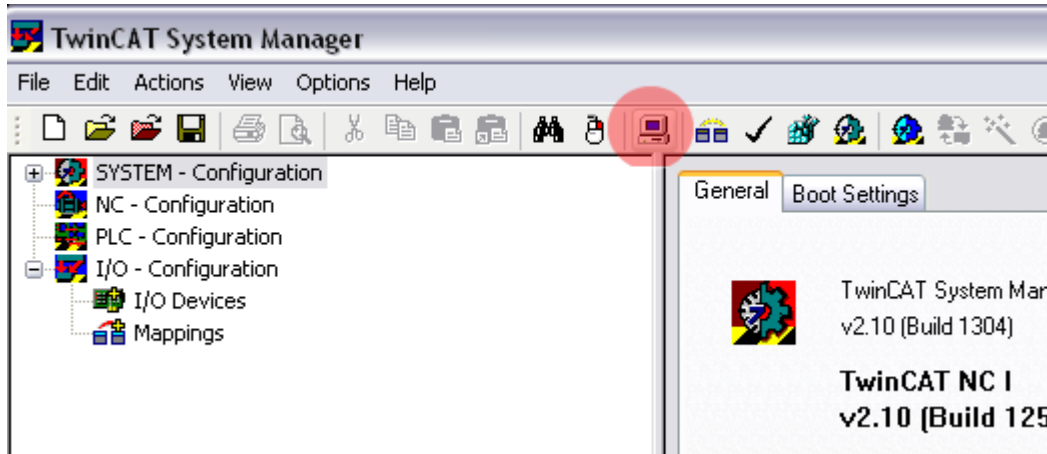


Abb. 107: BC mit dem System-Manager suchen

Klicken Sie *Search (Ethernet)...*

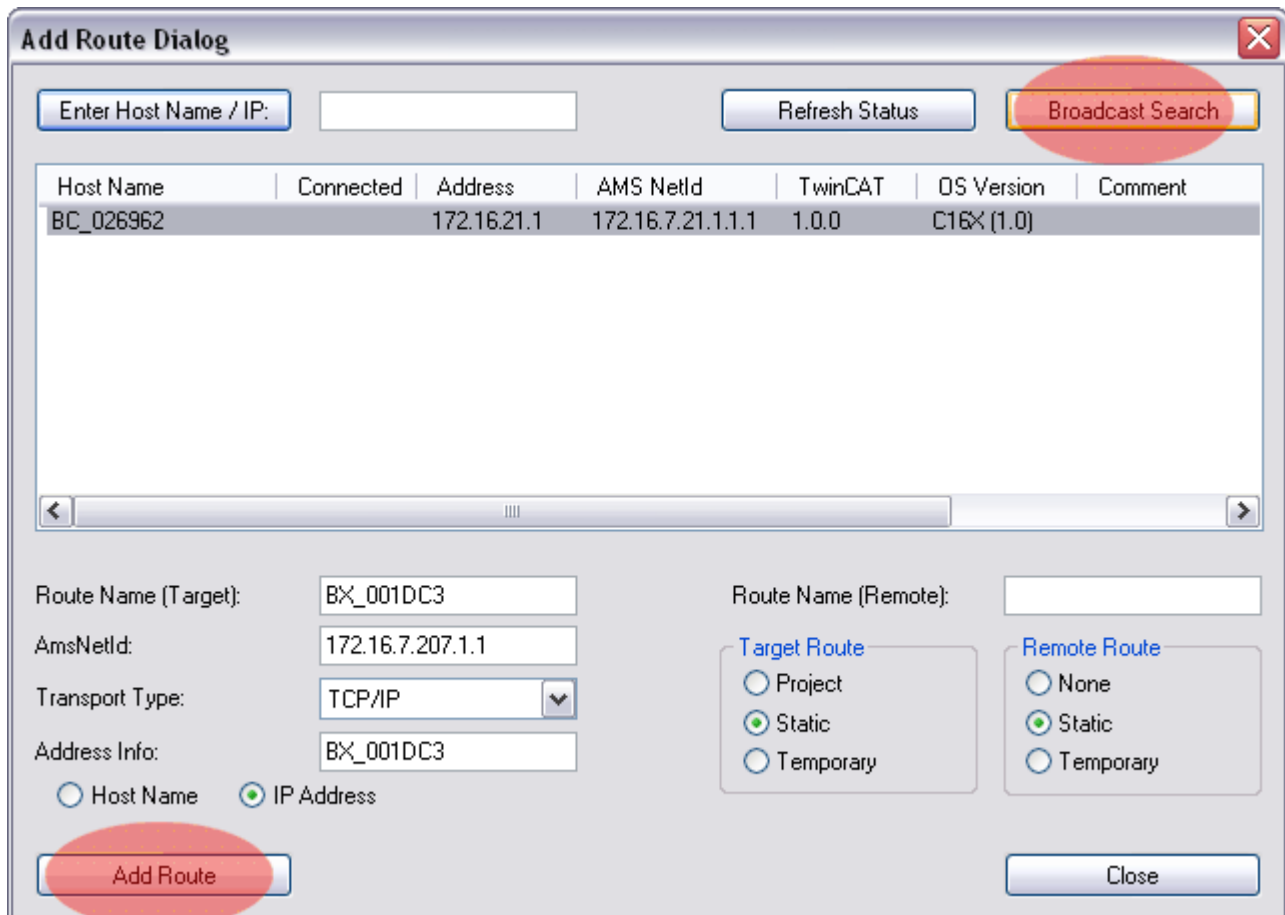


Abb. 108: System-Manager - Broadcast Search

Als nächstes klicken Sie auf **Broadcast Search**. Der BC erscheint nun im Fenster. Der Host Name wird gebildet aus BC plus den letzten 3 Byte der MAC-ID und ist somit einmalig in Ihrem System. Die MAC-ID steht auf einem Aufkleber unter dem Busklemmen-Controller. Stellen Sie nun den Options-Button auf *IP Address* und klicken Sie nun *Add Route*. Es sollte danach im Feld Connected ein Kreuz oder "X" erscheinen. Sollte eine Abfrage nach einem Passwort erscheinen, ist dies mit OK zu bestätigen ohne eine Eingabe zu machen. Die Funktion Passwort ist beim Busklemmen-Controller nicht implementiert.

Nun können Sie den Busklemmen-Controller auswählen. Klicken Sie dazu den BC an und bestätigen Sie mit OK.

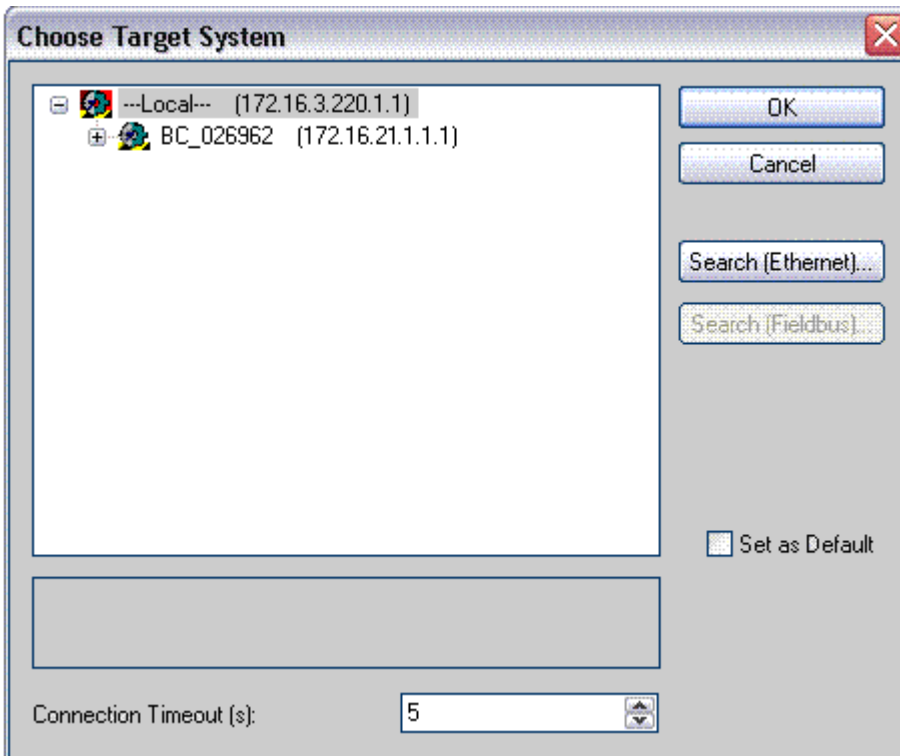


Abb. 109: Auswahl des Zielsystems

Links unten im System-Manager steht in rot Ihr BC mit Name und IP-Adresse. In dem Feld daneben sollte Config Mode stehen und das Feld blau sein. Ist das der Fall können Sie nun das Gerät Scannen. Klicken Sie dazu auf mit der rechten Maustaste auf *IO Devices* und dann **Scan Devices**

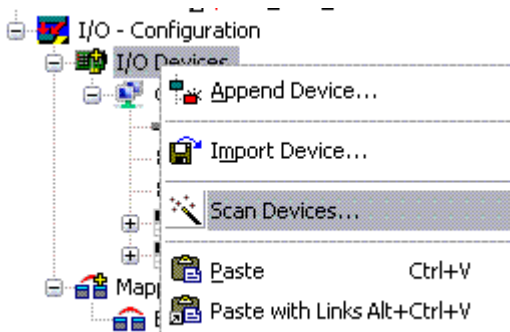


Abb. 110: System-Manager - Scan Devices

Der System-Manager sollte 2 Geräte finden. Das Ethernet Interface und das K-Bus Interface mit den angeschlossenen Busklemmen.

Ethernet Interface

Klicken sie auf **Upload** um die aktuelle IP-Konfiguration hoch zu laden.

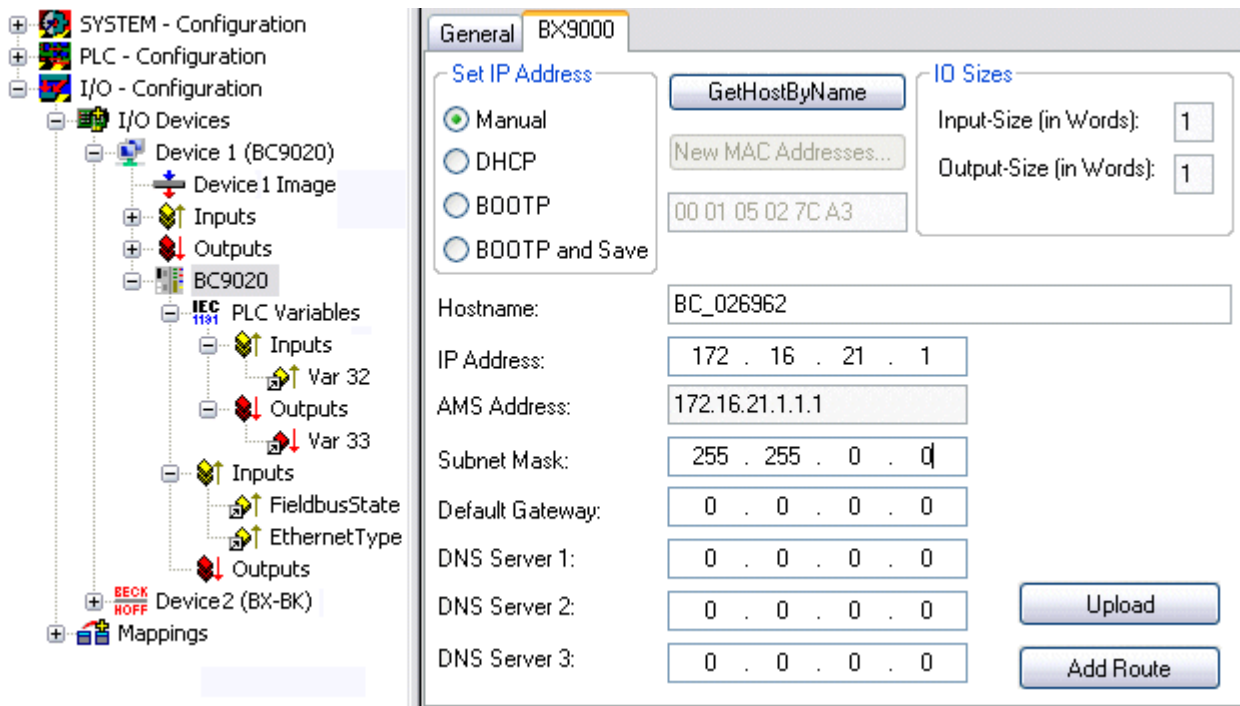



Abb. 111: Upload der die aktuellen IP-Konfiguration

PLC Configuration

Fügen Sie hier Ihr SPS-Programm an. Öffnen Sie als erstes das folgende Beispiel:

 (<https://infosys.beckhoff.com/content/1031/bc9xx0/Resources/zip/3740903947.zip>) Speichern Sie die Datei lokal auf ihrem Rechner.

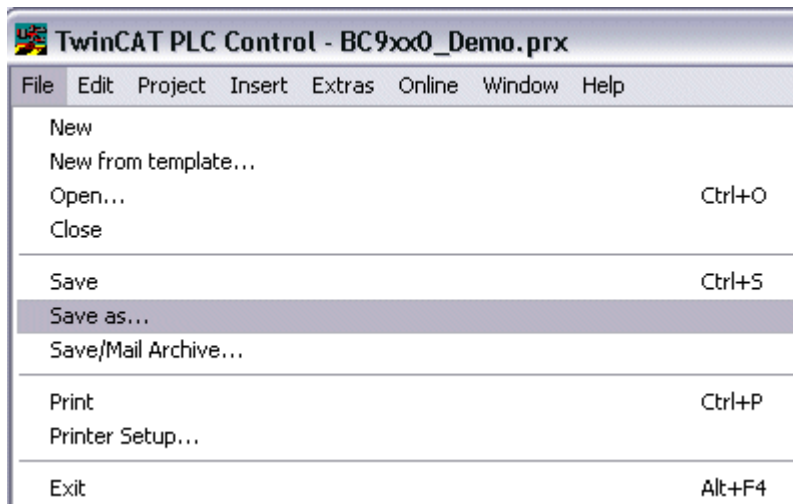


Abb. 112: Speichern der Datei lokal auf dem Rechner

Übersetzen Sie die Datei neu.

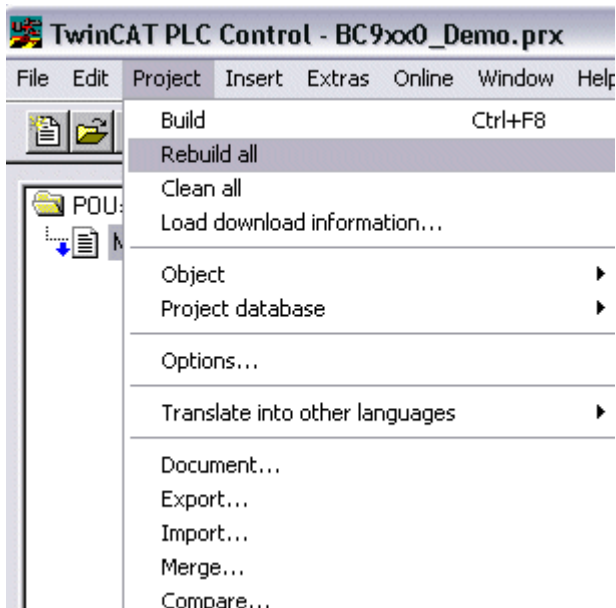


Abb. 113: Übersetzen der Datei

Speichern Sie erneut.

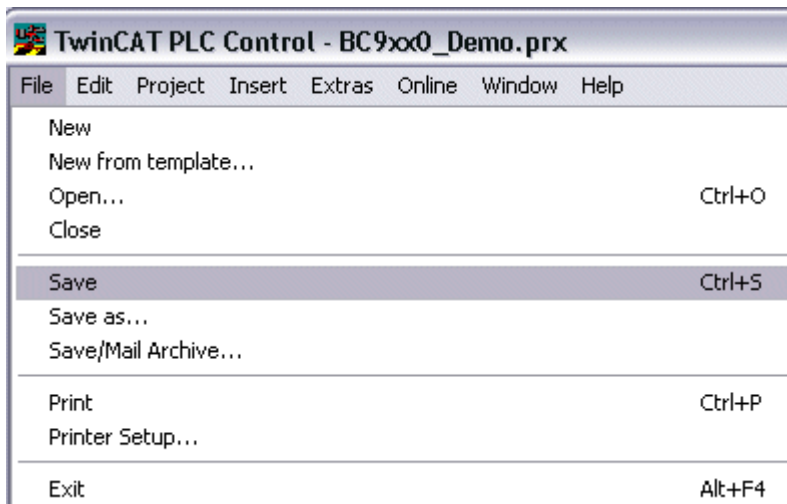


Abb. 114: Erneutes Speichern der Datei

Wechseln Sie nun wieder in den System-Manager und fügen Sie Ihr SPS-Programm an (rechte Maustaste auf PLC Configuration klicken).

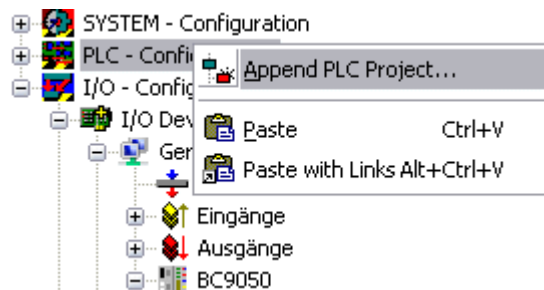


Abb. 115: Anfügen des SPS-Programms

Suchen sie nach Ihrer Datei als `tpy File`. Öffnen Sie dann den Baum. Sie sehen alle lokierten Daten. Nehmen Sie sich die Variablen und verknüpfen Sie diese mit Ihren Busklemmen.
BIN_1 mit KL1104 Kanal 1

bOUT_1 mit KL2134 Kanal 1
 bOUT_2 mit KL2134 Kanal 2
 bOUT_3 mit KL2134 Kanal 3

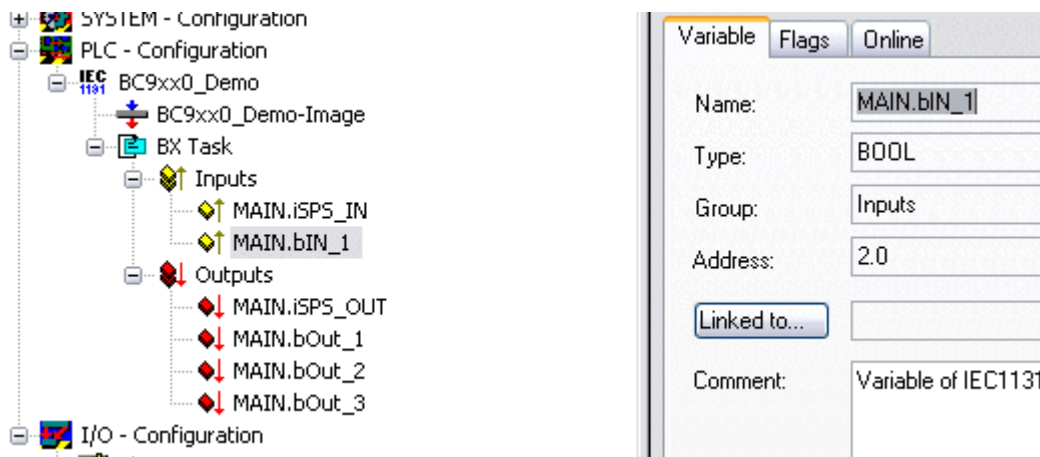


Abb. 116: Verknüpfen der lokierten Daten mit den Busklemmen

Nach der Verknüpfung muss es wie folgt aussehen.

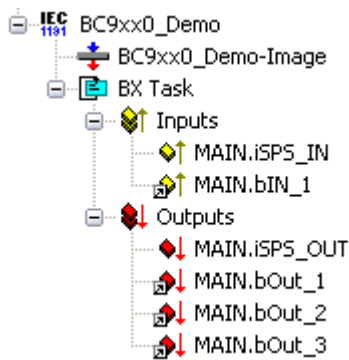




Abb. 117: Darstellung der verknüpften lokierten Daten

Aktivieren Sie nun die Konfiguration im System-Manager. Klicken Sie auf  um die Konfiguration zum Busklemmen-Controller herunterzuladen. Bestätigen Sie alle Meldungen mit OK. Zum Schluss wird nach einem Restart gefragt. Der BC bootet dann neu. In dem Augenblick verliert der System-Manager die Verbindung zum Busklemmen-Controller. Warten Sie bis der System-Manager die Verbindung wieder aufgebaut hat. Nach einigen Sekunden sollte im System-Manager links unten in der Ecke das Status Feld Grün

 sein. Ist dieser blau ist der BC erneut im Config Mode hochgelaufen. Überprüfen Sie ihre Konfiguration.

Wechseln Sie nun wieder in das PLC Control. Gehen Sie auf Online und Wählen Sie das Zielsystem aus.

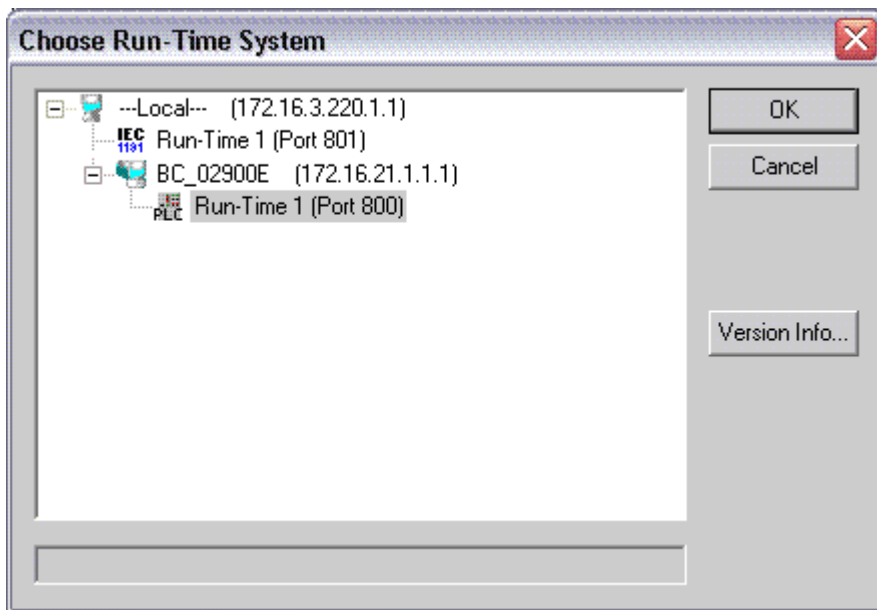


Abb. 118: Auswahl des Zielsystems

Danach loggen Sie sich ein und starten das Programm.

8.2 Umstieg zwischen den Controllern

Umstieg vom BCxx00 zum BCxx50/BCxx20

Dateinamen

Bei Busklemmen-Controllern der Serie BCxx50 und BCxx20 heißen Bibliotheken *.lbx und Programme werden als *.prx gespeichert.

Merker Variablen

Die lokierten Merker Variablen

- der BCxx00 belegen %MB0...%MB511 (außer BC9000/BC9100: %MB0...%B4095).
- der BCxx20 belegen %MB0...%MB4095
- der BCxx50 belegen %MB0...%MB4095

Zustandsinformationen wie K-Bus-/Feldbus-Status, Zyklustick werden nicht in den BCxx50/BCxx20 kopiert. Diese Informationen stehen in der TcSystemBCxx50.lbx für den BCxx50/BCxx20 als Funktion zur Verfügung.

Die lokierten Merker arbeiten **nicht** als Retain-Variablen.

Retain-Daten

Die Retain-Daten müssen als `VAR RETAIN [► 71]` deklariert werden. Es stehen bis zu 2 kByte zur Verfügung.

SPS-Variablen

In der Default-Config fangen die SPS-Variablen ab %IB1000 und %QB1000 an.

Large Model

Gibt es nicht bei BCxx50 und BCxx20.

Max. Speicher:

- BCxx50: 48 kByte
- BCxx20: 128 kByte

Task Time

Die Task-Zeit wird im PLC-Control festgelegt, diese sollte auf eine realistisch Größe eingestellt sein (messen der PLC Zykluszeit und des K-Busses). Die Background-Zeit entfällt.

Task-Konfiguration

Es steht maximal eine Task zur Verfügung. Diese Task muss konfiguriert werden.

PLC und Feldbusklemmen

Bei den Standard Busklemmen Controllern (BCxx00) gab es die Möglichkeit zu wählen, ob eine Busklemme dem Feldbus oder der lokalen PLC zugewiesen wird.

In der Default-Konfiguration des BCxx50/BCxx20 sind alle Busklemmen der lokalen PLC zugewiesen. Eine Zuordnung zum Feldbus ist hier nicht möglich.

Umstieg vom BCxx00 zum BXxx00

Dateinamen

Bei Busklemmen-Controllern der Serie BXxx00 heißen Bibliotheken *.lhx und Programme werden als *.prx gespeichert.

Merker-Variablen

Die lokierten Merker-Variablen

- der BCxx00 belegen %MB0...%MB511 (außer BC9000/BC9100: %MB0...%B4095).
- der BXxx00 belegen %MB0...%MB4095

Zustandsinformationen wie K-Bus-/Feldbus-Status, Zyklustick werden nicht in den BXxx00 kopiert. Diese Informationen stehen in der TcSystemBX.lhx für den BXxx00 als Funktion zur Verfügung.

Die lokierten Merker arbeiten **nicht** als Retain-Variablen.

Retain-Daten

Die Retain-Daten müssen als `VAR_RETAIN [► 71]` deklariert werden. Es stehen bis zu 2 kByte zur Verfügung.

SPS-Variablen

In der Default-Config fangen die SPS-Variablen ab %IB1000 und %QB1000 an.

Large Model

Gibt es nicht bei BXxx00. Max. Speicher: 256 kByte.

Task Time

Die Task-Zeit wird im PLC-Control festgelegt, diese sollte auf eine realistisch Größe eingestellt sein (messen der PLC Zykluszeit und des K-Busses). Die Background-Zeit entfällt.

Task-Konfiguration

Es steht maximal eine Task zur Verfügung. Diese Task muss konfiguriert werden.

PLC und Feldbusklemmen

Bei den Standard Busklemmen Controller (BCxx00) gab es die Möglichkeit zu wählen, ob eine Busklemme dem Feldbus oder der lokalen PLC zugewiesen wird.

In der Default-Konfiguration des BXxx00 sind alle Busklemmen der lokalen PLC zugewiesen. Eine Zuordnung zum Feldbus ist hier nicht möglich.

Umstieg vom PC zum BCxx50/BCxx20/BXxx00

Dateinamen

Bei den Busklemmen-Controllern der Serien BCxx50/BCxx20 und BXxx00 heißen Bibliotheken *.lbx und Programme werden als *.prx gespeichert.

Lokierte Variablen

Es stehen bei den Busklemmen-Controllern der Serien BCxx50/BCxx20 und BXxx00 eine begrenzte Anzahl von lokierten Daten zur Verfügung:

- Inputs 2 kByte, %IB0...2048
- Outputs 2 kByte, %QB0...2048
- Merker 4 kByte, %MB0...4095

Task-Konfiguration

Maximal steht eine Task zur Verfügung. Es ist eine sinnvolle Task-Zeit zu wählen. Passen Sie die Task-Zeit Ihrer Anwendung an, messen Sie dazu die benötigte Systemzeit (PLC + K-Bus + Feldbus + sonstiges).

Retain-Daten

Bei den Busklemmen-Controllern der Serien BCxx50, BCxx20 und BXxx00 stehen bis zu 2 kByte Retain-Daten zur Verfügung. Achten Sie daher darauf das Sie keine (oder nur sehr beschränkt) Retain-Daten in Funktionsblöcken verwenden (siehe [RETAIN Daten](#) [► 71]).

8.3 Allgemeine Betriebsbedingungen

Um einen fehlerfreien Betrieb der Feldbuskomponenten zu erreichen, müssen die nachfolgenden Bedingungen eingehalten werden.

Bedingungen an die Umgebung

Betrieb

An folgenden Orten dürfen die Komponenten nicht ohne Zusatzmaßnahmen eingesetzt werden:

- unter erschwerten Betriebsbedingungen, wie z.B. ätzende Dämpfe oder Gase, Staubbildung
- bei hoher ionisierender Strahlung

Bedingung	zulässiger Bereich
zulässige Umgebungstemperatur im Betrieb	siehe Technische Daten
Einbaulage	beliebig
Vibrationsfestigkeit	gemäß EN 60068-2-6
Schockfestigkeit	gemäß EN 60068-2-27
EMV-Festigkeit	gemäß EN 61000-6-2
Aussendung	gemäß EN 61000-6-4

Transport und Lagerung

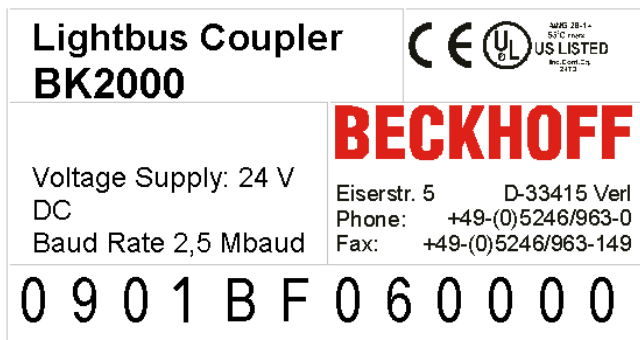
Bedingung	zulässiger Bereich
zulässige Umgebungstemperatur bei Lagerung	-25°C... +85°C
Relative Feuchte	95 %, keine Betauung
Freier Fall	originalverpackt bis 1 m

Schutzklasse und Schutzart


Bedingung	zulässiger Bereich
Schutzklasse nach IEC 536 (VDE 0106, Teil 1)	An der Profilschiene ist ein Schutzleiteranschluss erforderlich!
Schutzart nach IEC 529	IP20 (Schutz gegen Berührung mit Standard Prüffinger)
Schutz gegen Fremdkörper	kleiner 12 mm im Durchmesser
Schutz gegen Wasser	kein Schutz

Kennzeichnung der Komponenten

Jede ausgelieferte Komponente enthält einen Aufkleber, mit Informationen über die Zulassung des Produkts. Beispiel für den Buskoppler BK2000:



Auf dem Aufkleber sind folgende Informationen abzulesen:

Aufdruck	Bedeutung für diesen Aufkleber
genaue Produktbezeichnung	Lightbus Koppler BK2000
Versorgungsspannung Us	24 V _{DC} (Benutzen Sie eine 4 A Sicherung oder eine der Class 2 entsprechende Spannungsversorgung um die UL-Anforderungen zu erfüllen!)
Übertragungsrate	2,5 MBaud
Hersteller	Beckhoff Automation GmbH & Co. KG
CE-Zeichen	Konformitätskennzeichnung
UL-Zeichen 	Kennzeichen für UL-Zulassung. UL steht für Underwriters Laboratories Inc., die führende Zertifizierungsorganisation für Nordamerika mit Sitz in den USA. C = Kanada, US = USA, UL File Nummer: E172151
Produktionsbezeichnung	Die Zahlenfolge ergibt von links nach rechts die Produktionswoche (2 Ziffern), das Produktionsjahr (2 Ziffern), die Software- (2 Ziffern) sowie die Hardwareversion (2 Ziffern) und Sonderbezeichnungen (4 Ziffern). In diesem Fall handelt es sich also um einen BK2000 - produziert in der 9. Kalenderwoche - de Jahres 2001 - mit der Firmware-Version BF - in der 6. Hardwareversion - ohne Sonderbezeichnung

8.4 Prüfnormen für Geräteprüfung

EMV

EMV-Festigkeit

EN 61000-6-2

EMV-Aussendung

EN 61000-6-4

Vibrations-/ Schockfestigkeit

Vibrationsfestigkeit

EN 60068-2-6

Schockfestigkeit

EN 60068-2-27

8.5 Literaturverzeichnis

TCP/IP

TCP/IP (deutsch)
Aufbau und Betrieb eines TCP/IP Netzes
von Kevin Washburn, Jim Evans
Verlag: ADDISON-WESLEY Longmann Verlag

TCP/IP (englisch)
Illustrated, Volume1 The Protocols
von W. Richard Stevens
Verlag: ADDISON-WESLEY Longmann Verlag

Modbus/TCP

<http://www.modicon.com/>
<http://www.modbus.org>

TwinCAT

BECKHOFF Information System
<http://infosys.beckhoff.com>

8.6 Abkürzungsverzeichnis

ADS

Automation Device Specification

IP (20)

Schutzart der Busklemmen

IPC

Industrie-PC

E/A

Ein- und Ausgänge

K-Bus

Klemmen-Bus

KS2000

Konfigurationssoftware für Busklemmen, Bus Koppler, Busklemmen Controller, Feldbus Box Module usw.

PE

Der PE-Powerkontakt kann als Schutz Erde verwendet werden.

TwinCAT

The Windows Control and Automation Technology

8.7 Support und Service

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

Beckhoff Niederlassungen und Vertretungen

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

Die Adressen der weltweiten Beckhoff Niederlassungen und Vertretungen entnehmen Sie bitte unseren Internetseiten: <https://www.beckhoff.de>

Dort finden Sie auch weitere Dokumentationen zu Beckhoff Komponenten.

Beckhoff Support

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

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

Hotline: +49(0)5246 963 157
Fax: +49(0)5246 963 9157
E-Mail: support@beckhoff.com

Beckhoff Service

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

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

Hotline: +49(0)5246 963 460
Fax: +49(0)5246 963 479
E-Mail: service@beckhoff.com

Beckhoff Firmenzentrale

Beckhoff Automation GmbH & Co. KG

Hülshorstweg 20
33415 Verl
Deutschland

Telefon: +49(0)5246 963 0
Fax: +49(0)5246 963 198
E-Mail: info@beckhoff.com
Internet: <https://www.beckhoff.de>

Abbildungsverzeichnis

Abb. 1	BC9020	12
Abb. 2	BC9050	13
Abb. 3	BC9120	14
Abb. 4	BC9020, BC9120 - Prinzip der Busklemmen.....	18
Abb. 5	BC9050 - Prinzip der Busklemme.....	18
Abb. 6	Federkontakte der Beckhoff I/O-Komponenten	21
Abb. 7	BK9000, BK9100, BC9000, BC9020, BC9100, BC9120	22
Abb. 8	BK9050, BC9050	23
Abb. 9	Entsichern der Verriegelung durch die orangefarbene Zuglasche	24
Abb. 10	Nut und Feder der Gehäuse	24
Abb. 11	Potentialgruppen eines Busklemmenblocks	25
Abb. 12	Linksseitiger Powerkontakt	26
Abb. 13	Klemmstellen zur Versorgung des Busklemmen-Controllers	27
Abb. 14	Ethernet-Aufbau in Sterntopologie.....	28
Abb. 15	Ethernet-Aufbau in Linien-Topologie	29
Abb. 16	RJ45-Stecker	30
Abb. 17	Ethernet-Verbindung zwischen PC und BC9020, BC9050 über Hub oder Switch	30
Abb. 18	Direkte Ethernet-Verbindung zwischen PC und BC9020, BC9050 über Cross-Over-Kabel	31
Abb. 19	Ethernet-Verbindung zwischen PC und BC9120 über Hub oder Switch	31
Abb. 20	Direkte Ethernet-Verbindung zwischen PC und BC9120	32
Abb. 21	Anlaufverhalten des BC9xx0	38
Abb. 22	Anzeige im System-Manager.....	39
Abb. 23	DIP-Schalter des BC9050.....	40
Abb. 24	DIP-Schalter des BC9020 und des BC9120.....	40
Abb. 25	Adresseinstellung über den TwinCAT System Manager	42
Abb. 26	Adresseinstellung über BootP-Server.....	43
Abb. 27	Suchen des Busklemmen-Controllers	47
Abb. 28	Auswahl des Zielsystems	48
Abb. 29	Add Route Dialog.....	49
Abb. 30	Anlegen einer TwinCAT-Konfiguration	49
Abb. 31	Auswahl des Busklemmen-Controllers	50
Abb. 32	Download einer TwinCAT-Konfiguration.....	51
Abb. 33	Auswahl des Busklemmen-Controllers	51
Abb. 34	Zustand des Busklemmen-Controllers.....	51
Abb. 35	Aktivieren der TwinCAT-Konfiguration.....	51
Abb. 36	Auswahl des Zielsystems	52
Abb. 37	Auswahl des Busklemmen-Controllers	53
Abb. 38	Zustand des Busklemmen-Controllers.....	53
Abb. 39	Hochladen der TwinCAT-Konfiguration	53
Abb. 40	Speicher für das Code Mapping	54
Abb. 41	Daten Speicher Mapping	54
Abb. 42	Code und Daten Speicher	55
Abb. 43	Sonstiger Speicher	55
Abb. 44	Eigenschaften der Remote-Verbindung.....	56

Abb. 45	Anpassen der Zeit nach der die TwinCAT PLC einen Busklemmen-Controller abfragt	59
Abb. 46	Karteireiter BX Settings	60
Abb. 47	Karteireiter BX Diag	61
Abb. 48	Auswahl des PLC-Projekts	62
Abb. 49	Verbinden vom PLC-Variable und Hardware.....	63
Abb. 50	Anzeige des Ziel-Systems	63
Abb. 51	Einstellen der Task-Zeit.....	64
Abb. 52	Anzeige der PLC-Zykluszeit	65
Abb. 53	Konfigurations-Software KS2000.....	66
Abb. 54	Maximale Anzahl der POU's überschritten.....	69
Abb. 55	Menüpfad Projekte / Optionen / Controller Settings	69
Abb. 56	Controller Settings	69
Abb. 57	Globaler Speicher nicht ausreichend.....	70
Abb. 58	Menüpfad Projekte / Optionen / Build	70
Abb. 59	Build.....	70
Abb. 60	Ändern der Verknüpfung von Variablen.....	76
Abb. 61	Verknüpfen einer Variable mit einem Eingang	76
Abb. 62	Öffnen des Optionsmenüs	79
Abb. 63	Auswahl des Source Downloads	79
Abb. 64	Download des Programm Codes.....	80
Abb. 65	Vorschritt des Downloads	80
Abb. 66	Upload eines Programms	81
Abb. 67	Auswahl des Datenübertragungswegs	81
Abb. 68	Auswahl des Gerätes.....	82
Abb. 69	Funktionsbaustein F_STARTDEBUGTIMER.....	88
Abb. 70	Funktionsbaustein F_READDEBUGTIMER	88
Abb. 71	Funktionsbaustein FB_IPSTARTSESSION.....	91
Abb. 72	Funktionsbaustein FB_IPENDSESSION	92
Abb. 73	Funktionsbaustein FB_IPOPEN	93
Abb. 74	Funktionsbaustein FB_IPCLOSE	94
Abb. 75	Funktionsbaustein FB_IPRECEIVE.....	95
Abb. 76	Funktionsbaustein FB_IPSEND.....	96
Abb. 77	UDP/IP-Verbindung	100
Abb. 78	Ablauf des Beispielprogramms	102
Abb. 79	Funktionsbaustein FB_MBCONNECT	103
Abb. 80	Funktionsbaustein FB_MBGENERICREQ	104
Abb. 81	Funktionsbaustein FB_MBCLOSE	105
Abb. 82	Funktionsbaustein FB_Smtp.....	106
Abb. 83	Funktionsbaustein fbSMTP.....	108
Abb. 84	Funktionsbaustein FB_Sntp	109
Abb. 85	Funktionsbaustein FB_AddDnsServer.....	109
Abb. 86	Funktionsbaustein FB_GetHostByAddr	110
Abb. 87	Funktionsbaustein FB_GetHostByName	111
Abb. 88	Funktionsbaustein FB_GetNetworkConfig.....	112
Abb. 89	Funktionsbaustein FB_SetTargetName.....	113
Abb. 90	IP-Adresse des BX9000 im TwinCAT System manager.....	115

Abb. 91	Auswahl des Datenübertragungswegs - AMS	116
Abb. 92	Auswahl des Zielsystems	116
Abb. 93	Auswahl des Datenübertragungswegs - Serielle Schnittstelle.....	117
Abb. 94	Parametrierung der seriellen Schnittstelle	117
Abb. 95	Auswahl des Datenübertragungswegs - AMS	117
Abb. 96	Auswahl des Gerätes.....	118
Abb. 97	User Datagram Protocol (UDP)	120
Abb. 98	Auf TCP/IP und UDP/IP aufsetzende Protokolle	120
Abb. 99	ModbusTCP-Protokoll.....	121
Abb. 100	Das ADS-Protokoll als Transportschicht innerhalb des TwinCAT-Systems	128
Abb. 101	Aufbau des ADS-Protokolls	129
Abb. 102	Zustand des Ethernet	133
Abb. 103	Zustand des K-Bus	134
Abb. 104	BC9050 LEDs	134
Abb. 105	LEDs BC9120	135
Abb. 106	Einstellen der IP-Adresse des Programmier-PCs.....	138
Abb. 107	BC mit dem System-Manager suchen	139
Abb. 108	System-Manager - Broadcast Search.....	139
Abb. 109	Auswahl des Zielsystems	140
Abb. 110	System-Manager - Scan Devices	140
Abb. 111	Upload der die aktuellen IP-Konfiguration	141
Abb. 112	Speichern der Datei lokal auf dem Rechner	141
Abb. 113	Übersetzen der Datei.....	142
Abb. 114	Erneutes Speichern der Datei.....	142
Abb. 115	Anfügen des SPS-Programms.....	142
Abb. 116	Verknüpfen der lokierten Daten mit den Busklemmen	143
Abb. 117	Darstellung der verknüpften lokierten Daten	143
Abb. 118	Auswahl des Zielsystems	144

Mehr Informationen:
www.beckhoff.de/BCxxxx

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

