

Engineer/ PLC Designer



Application Samples Inverter 8400/9400_____

Software-Handbuch

DE



Inhalt

1	Über diese Dokumentation.....	7
1.1	Dokumenthistorie.....	10
1.2	Verwendete Konventionen.....	11
1.3	Verwendete Begriffe.....	12
1.4	Verwendete Hinweise.....	13
2	Sicherheitshinweise.....	14
3	Voraussetzungen.....	16
3.1	Systemvoraussetzungen.....	16
3.2	Geräte (Controller, Inverter, Motoren) verdrahten.....	17
3.3	Kommunikation zum Controller einrichten.....	18
4	Lenze Application Samples.....	19
4.1	Übersicht der Application Samples.....	20
4.2	Wo finden Sie die Application Samples?.....	21
5	Beispielprojekte im »Engineer« verwenden.....	22
6	Beispielprojekte im »PLC Designer« verwenden.....	24
6.1	Steuerungskonfiguration (EtherCAT/CAN).....	26
6.2	Beispielprojekt mit der Applikation "Stellantrieb - Drehzahl".....	27
6.3	Beispielprojekt mit der Applikation "Tabellenpositionierung".....	28
6.4	Programmbausteine (PRG).....	29
6.5	Visualisierungen der Applikation "Stellantrieb - Drehzahl".....	30
6.5.1	Information (Startseite).....	31
6.5.2	Automatic-Modus.....	32
6.5.3	Manual-Modus (Handfahr-Betrieb).....	33
6.5.4	Service-Modus.....	34
6.5.5	Parameter.....	35
6.6	Visualisierungen der Applikation "Tabellenpositionierung".....	36
6.6.1	Information (Startseite).....	37
6.6.2	Automatic-Modus.....	38
6.6.3	Manual-Modus (Handfahr-Betrieb).....	39
6.6.4	Service-Modus.....	40
6.6.5	Homing-Modus (Referenzfahrt).....	41
6.6.6	Parameter.....	42
6.7	Beispielprojekte erweitern (Geräte hinzufügen).....	43
6.7.1	EtherCAT-Feldgeräte anhängen.....	43
6.7.2	CAN-Feldgeräte anhängen.....	46

7	Gerätebeschreibungen	50
7.1	Inverter Drive 8400 BaseLine C.....	50
7.2	Inverter Drive 8400 StateLine C	52
7.3	Inverter Drive 8400 HighLine C.....	54
7.4	Inverter Drive 8400 TopLine C.....	56
7.5	Servo Drive 9400 HighLine.....	58
7.5.1	Indizes bei Technologieapplikation "Tabellenpositionierung".....	59
7.5.2	Indizes bei den weiteren Technologieapplikationen	60
7.6	Servo Drive 9400 Regenerative Power Supply Module	61
7.7	Lenze Generic Drive	63

8	Funktionsbibliothek L_LCB_LogicControlBasic.....	65
8.1	Übersicht der Funktionen und Funktionsbausteine.....	66
8.2	Funktionsbausteine	67
8.2.1	L_LCB_AXIS_REF.....	67
8.2.2	L_LCB_AXIS_REF_CAN (interne Verwendung)	68
8.2.3	L_LCB_AXIS_REF_ETC (interne Verwendung)	68
8.2.4	L_LCB_8400Drive.....	69
8.2.5	L_LCB_9400Drive.....	70
8.2.6	L_LCB_GenericDrive	72
8.2.7	L_LCB_ActuatorSpeed	73
8.2.8	L_LCB_TablePositioning.....	77
8.2.9	L_LCB_Norm_aToNorm_n	83
8.2.10	L_LCB_Norm_nToNorm_a	83
8.2.11	L_LCB_Norm_nToSpeed_s	84
8.2.12	L_LCB_Speed_nToNorm_r	84
8.2.13	L_LCB_Speed_rToNorm_n	85
8.2.14	L_LCB_Speed_sToNorm_n	85
8.2.15	L_LCB_Speed_sToSpeed_v	86
8.2.16	L_LCB_Speed_vToSpeed_s	86
8.2.17	L_LCB_GetAxisData.....	87
8.2.18	L_LCB_GetPosition.....	88
8.2.19	L_LCB_GetSpeed	89
8.2.20	L_LCB_SetAxisData	90
8.2.21	L_LCB_SetPosition	92
8.2.22	L_LCB_SetSpeed	93
8.2.23	L_LCB_AccToUnit	94
8.2.24	L_LCB_PosToUnit	95
8.2.25	L_LCB_SpeedToUnit	96
8.2.26	L_LCB_UnitToAcc	97
8.2.27	L_LCB_UnitToPos	98
8.2.28	L_LCB_UnitToSpeed	99
8.2.29	L_LCB_TaskCycle	100
8.3	Strukturen (Structs)	101
8.3.1	L_LCB_AxisData	101
8.4	Enumerationen (Enums).....	102
8.4.1	L_LCB_AXISMODE	102
8.4.2	L_LCB_Error	102
8.4.3	L_LCB_LOGICDEVICE	102

9	Funktionsbibliothek L_DCO_DriveCommunication.....	103
9.1	Übersicht der Funktionen und Funktionsbausteine.....	104
9.2	Funktionsbausteine	105
9.2.1	L_DCO_AXIS_REF_BASE.....	105
9.2.2	L_DCO_ReadDriveParameter.....	106
9.2.3	L_DCO_ReadDriveParameterString	108
9.2.4	L_DCO_TransferDriveParameter.....	110
9.2.5	L_DCO_WriteDriveParameter	112
9.3	Strukturen (Structs)	114
9.3.1	L_DCO_SDOData	114
9.3.2	L_DCO_TransferData	114
9.4	Enumerationen (Enums).....	115
9.4.1	L_DCO_CommState	115
9.4.2	L_DCO_Error	116
9.5	Unions	119
9.5.1	L_DCO_ConvertData.....	119
10	Funktionsbibliothek L_DAC_DataConversion.....	120
10.1	Übersicht der Funktionen und Funktionsbausteine.....	120
10.2	Bit operations.....	121
10.2.1	L_DAC_GetBitOfByte	121
10.2.2	L_DAC_GetBitOfDWord.....	121
10.2.3	L_DAC_GetBitOfWord	122
10.2.4	L_DAC_ResetBitOfByte.....	122
10.2.5	L_DAC_ResetBitOfDWord.....	123
10.2.6	L_DAC_ResetBitOfWord	123
10.2.7	L_DAC_SetBitOfByte.....	124
10.2.8	L_DAC_SetBitOfDWord	124
10.2.9	L_DAC_SetBitOfWord.....	125
10.3	Bit splitter	126
10.3.1	L_DAC_BitsToByte.....	126
10.3.2	L_DAC_BitsToDWord	126
10.3.3	L_DAC_BitsToWord.....	127
10.3.4	L_DAC_ByteToBits	127
10.3.5	L_DAC_DWordToBits	128
10.3.6	L_DAC_WordToBits.....	128

10.4	Type converter	129
10.4.1	L_DAC_2BytesToWord	129
10.4.2	L_DAC_2WordsToDWord	129
10.4.3	L_DAC_4BytesToDWord	130
10.4.4	L_DAC_DWordTo2Words	130
10.4.5	L_DAC_DWordTo4Bytes	131
10.4.6	L_DAC_WordTo2Bytes	131
11	Automatisch generierte Funktionen (Projektinformationen)	132
11.1	GetBooleanProperty	132
11.2	GetCompany	133
11.3	GetNumberProperty	133
11.4	GetTextProperty	134
11.5	GetTitle	134
11.6	GetVersion	135
11.7	GetVersionProperty	135
12	Index	136
	<i>Ihre Meinung ist uns wichtig</i>	<i>139</i>

1 Über diese Dokumentation

Diese Dokumentation beschreibt die Lenze Application Samples zur einfachen Inbetriebnahme von Inverter Drives 8400 und Servo Drives 9400 im Rahmen des Lenze-Automatationssystems "Controller-based Automation".

Die Application Samples sind vordefinierte Beispielprojekte für die Lenze Engineering Tools »PLC Designer« und »Engineer«.

Je nach Anwendungsfall können die Beispielprojekte beliebig erweitert und an die jeweiligen Anforderungen angepasst werden.

Diese Dokumentation ordnet sich in die Handbuchsammlung "Controller-based Automation" ein. Diese besteht aus folgenden Dokumentationen:

Dokumentationstyp	Thema
Produktkatalog	Controller-based Automation (Systemübersicht, Beispieltopologien) Lenze-Controller (Produktinformationen, Technische Daten)
Systemhandbücher	Visualisierung (Systemübersicht/Beispieltopologien)
Kommunikationshandbücher Online-Hilfen	Bussysteme • Controller-based Automation EtherCAT® • Controller-based Automation CANopen® • Controller-based Automation PROFIBUS® • Controller-based Automation PROFINET®
Referenzhandbücher Online-Hilfen	Lenze-Controller: • Controller 3200 C • Controller c300 • Controller p300 • Controller p500
Software-Handbücher Online-Hilfen	Lenze Engineering Tools: • »PLC Designer« (Programmierung) • »Engineer« (Parametrierung, Konfigurierung, Diagnose) • »VisiWinNET® Smart« (Visualisierung) • »Backup & Restore« (Datensicherung, Wiederherstellung, Aktualisierung)

Weitere Technische Dokumentationen zu Lenze-Produkten

Weitere Informationen zu Lenze-Produkten, die in Verbindung mit der Controller-based Automation verwendbar sind, finden Sie in folgenden Dokumentationen:

Planung / Projektierung / Technische Daten	
☐	Produktkataloge • Controller-based Automation • Controller • Inverter Drives/Servo Drives
Montage und Verdrahtung	
	Montageanleitungen • Controller • Kommunikationskarten (MC-xxx) • I/O-System 1000 (EPM-Sxxx) • Inverter Drives/Servo Drives • Kommunikationsmodule
☐	Gerätehandbücher • Inverter Drives/Servo Drives
Parametrierung / Konfigurierung / Inbetriebnahme	
☐	Online-Hilfe / Referenzhandbücher • Controller • Inverter Drives/Servo Drives • I/O-System 1000 (EPM-Sxxx)
☐	Online-Hilfe / Kommunikationshandbücher • Bussysteme • Kommunikationsmodule
Beispielapplikationen und Vorlagen	
☐	Online-Hilfe / Software- und Referenzhandbücher • Application Sample i700 • Application Samples 8400/9400 • FAST Application Template Lenze/PackML • FAST Technologiemodule

Symbole:

-  Gedruckte Dokumentation
- ☐ PDF-Datei / Online-Hilfe im Lenze Engineering Tool



Tipp!

Aktuelle Dokumentationen und Software-Updates zu Lenze-Produkten finden Sie im Download-Bereich unter:

www.lenze.com

Zielgruppe

Diese Dokumentation richtet sich an alle Personen, die ein Lenze-Automationssystem projektieren, installieren, in Betrieb nehmen und warten.

Informationen zur Gültigkeit

Die Informationen in dieser Dokumentation sind gültig für:

Engineering Tool	ab Software-Version
»PLC Designer«	3.3
»Engineer«	2.13

Die Lenze Engineering Tools stehen zum Download zur Verfügung unter:

www.lenze.com → Download → Software Downloads

Screenshots/Anwendungsbeispiele

Alle Screenshots in dieser Dokumentation sind Anwendungsbeispiele. Je nach Firmware-Version der eingesetzten Lenze-Geräte und Software-Version der installierten Engineering Tools (z. B. »PLC Designer«) können die Screenshots in dieser Dokumentation von der Bildschirm-Darstellung abweichen.

1 Über diese Dokumentation

1.1 Dokumenthistorie

1.1 Dokumenthistorie

Version			Beschreibung
2.1	08/2016	TD11	Hinweise zur Verwendung ergänzt: ▶ L_LCB_SetAxisData (📖 90) ▶ L_DCO_AXIS_REF_BASE (📖 105) ▶ L_DCO_ReadDriveParameter (📖 106) ▶ L_DCO_ReadDriveParameterString (📖 108) ▶ L_DCO_TransferDriveParameter (📖 110) ▶ L_DCO_WriteDriveParameter (📖 112)
2.0	02/2016	TD11	Allgemeine Überarbeitung und Aktualisierungen
1.4	06/2015	TD11	Aktualisiert: • L_LCB_GetAxisData (📖 87)
1.3	06/2014	TD11	• Übersicht der Application Samples (📖 20) ergänzt. • Darstellung der Funktionsbausteine aktualisiert.
1.2	06/2012	TD11	Hinweise zur Verwendung ergänzt: ▶ L_LCB_ActuatorSpeed (📖 73) ▶ L_LCB_TablePositioning (📖 77)
1.1	05/2011	TD11	• Gerätebeschreibungen (📖 50) ergänzt. • Funktionsbibliothek L_LCB_LogicControlBasic (📖 65) erweitert.
1.0	04/2011	TD11	Erstausgabe

1 Über diese Dokumentation

1.2 Verwendete Konventionen

Diese Dokumentation verwendet folgende Konventionen zur Unterscheidung verschiedener Arten von Information:

Informationsart	Auszeichnung	Beispiele/Hinweise
Zahlenschreibweise		
Dezimaltrennzeichen	Punkt	Es wird generell der Dezimalpunkt verwendet. Zum Beispiel: 1234.56
Hexadezimalzahl	0x	Für Hexadezimalzahlen wird der Präfix "0x" verwendet. Beispiel: 0x60F4
Binärzahl	0b	Für Binärzahlen wird der Präfix "0b" verwendet. Beispiel: 0b00010111
Textauszeichnung		
Versionsinfo	Textfarbe blau	Alle Informationen, die nur für oder ab einem bestimmten Softwarestand des Inverters gelten, sind in dieser Dokumentation entsprechend gekennzeichnet. Beispiel: Diese Funktionserweiterung ist ab dem Softwarestand V3.0 verfügbar!
Programmname	» «	Lenze »PLC Designer«...
Fensterbereich	kursiv	Das Meldungsfenster... / Das Dialogfeld Optionen...
Variablenbezeichner		Durch Setzen von <i>bEnable</i> auf TRUE...
Steuerelement	fett	Die Schaltfläche OK ... / Der Befehl Kopieren ... / Die Registerkarte Eigenschaften ... / Das Eingabefeld Name ...
Folge von Menübefehlen		Sind zum Ausführen einer Funktion mehrere Befehle nacheinander erforderlich, sind die einzelnen Befehle durch einen Pfeil voneinander getrennt: Wählen Sie den Befehl Datei → Öffnen , um...
Tastaturbefehl	<fett>	Mit <F1> rufen Sie die Onlinehilfe auf. Ist für einen Befehl eine Tastenkombination erforderlich, ist zwischen den Tastenbezeichnern ein "+" gesetzt: Mit <Shift>+<ESC>...
Programmcode	Courier	IF var1 < var2 THEN a = a + 1 END IF
Schlüsselwort	Courier fett	
Hyperlink	<u>unterstrichen</u>	Optisch hervorgehobener Verweis auf ein anderes Thema. Wird in dieser Online-Dokumentation per Mausklick aktiviert.
Seitenverweis	( 11)	Verweis auf weiterführenden Informationen: Seitenzahl in PDF-Datei.
Symbole		
Schrittweise Anleitung		Schrittweise Anleitungen sind durch ein Piktogramm gekennzeichnet.

1 Über diese Dokumentation

1.3 Verwendete Begriffe

1.3 Verwendete Begriffe

Begriff	Bedeutung
Controller	Der Controller ist die zentrale Komponente des Lenze-Automationssystems, das mit Hilfe des Betriebssystems die Bewegungsabläufe steuert. Der Controller kommuniziert über den Feldbus mit den Feldgeräten (Inverter).
Engineering PC	Mit dem Engineering PC und den darauf installierten Engineering Tools konfigurieren und parametrieren Sie das System. Der Engineering PC kommuniziert über Ethernet mit dem Controller.
 EtherCAT	EtherCAT® (Ethernet for Controller and Automation Technology) ist ein Ethernet-basierendes Feldbussystem, welches das Anwendungsprofil für industrielle Echtzeitsysteme erfüllt. EtherCAT® ist eine eingetragene Marke und patentierte Technologie, lizenziert durch die Beckhoff Automation GmbH, Deutschland.
FAST	Lenze Application Software
Inverter	Oberbegriff für Lenze-Frequenzumrichter, Servo-Umrichter
PLC	Programmable Logic Controller (deutsche Bezeichnung: SPS - Speicherprogrammierbare Steuerung)

1.4 Verwendete Hinweise

Um auf Gefahren und wichtige Informationen hinzuweisen, werden in dieser Dokumentation folgende Signalwörter und Symbole verwendet:

Sicherheitshinweise

Aufbau der Sicherheitshinweise:



Piktogramm und Signalwort!

(kennzeichnen die Art und die Schwere der Gefahr)

Hinweistext

(beschreibt die Gefahr und gibt Hinweise, wie sie vermieden werden kann)

Piktogramm	Signalwort	Bedeutung
	Gefahr!	Gefahr von Personenschäden durch gefährliche elektrische Spannung Hinweis auf eine unmittelbar drohende Gefahr, die den Tod oder schwere Verletzungen zur Folge haben kann, wenn nicht die entsprechenden Maßnahmen getroffen werden.
	Gefahr!	Gefahr von Personenschäden durch eine allgemeine Gefahrenquelle Hinweis auf eine unmittelbar drohende Gefahr, die den Tod oder schwere Verletzungen zur Folge haben kann, wenn nicht die entsprechenden Maßnahmen getroffen werden.
	Stop!	Gefahr von Sachschäden Hinweis auf eine mögliche Gefahr, die Sachschäden zur Folge haben kann, wenn nicht die entsprechenden Maßnahmen getroffen werden.

Anwendungshinweise

Piktogramm	Signalwort	Bedeutung
	Hinweis!	Wichtiger Hinweis für die störungsfreie Funktion
	Tipp!	Nützlicher Tipp für die einfache Handhabung
		Verweis auf andere Dokumentation

2 Sicherheitshinweise

Beachten Sie die Sicherheitshinweise in dieser Dokumentation, wenn Sie ein Automationssystem oder eine Anlage mit einem Lenze-Controller in Betrieb nehmen möchten.



Die Gerätedokumentation enthält Sicherheitshinweise, die Sie beachten müssen!

Lesen Sie die mitgelieferten und zugehörigen Dokumentationen der jeweiligen Komponenten des Automationssystems sorgfältig durch, bevor Sie mit der Inbetriebnahme des Controllers und der angeschlossenen Geräte beginnen.



Gefahr!

Hohe elektrische Spannung

Personenschäden durch gefährliche elektrische Spannung

Mögliche Folgen

Tod oder schwere Verletzungen

Schutzmaßnahmen

Die Spannungsversorgung ausschalten, bevor Arbeiten an den Komponenten des Automationssystems durchgeführt werden.

Nach dem Ausschalten der Spannungsversorgung spannungsführende Geräteteile und Leistungsanschlüsse nicht sofort berühren, weil Kondensatoren aufgeladen sein können.

Die entsprechenden Hinweisschilder auf dem Gerät beachten.



Gefahr!

Personenschäden

Verletzungsgefahr besteht durch ...

- nicht vorhersehbare Motorbewegungen (z. B. ungewollte Drehrichtung, zu hohe Geschwindigkeit oder ruckhafter Lauf);
- unzulässige Betriebszustände bei der Parametrierung, während eine Online-Verbindung zum Gerät besteht.

Mögliche Folgen

Tod oder schwere Verletzungen

Schutzmaßnahmen

- Anlagen mit eingebauten Invertern ggf. mit zusätzlichen Überwachungs- und Schutzeinrichtungen nach den jeweils gültigen Sicherheitsbestimmungen ausrüsten (z. B. Gesetz über technische Arbeitsmittel, Unfallverhütungsvorschriften).
- Während der Inbetriebnahme einen ausreichenden Sicherheitsabstand zum Motor oder den vom Motor angetriebenen Maschinenteilen einhalten.



Stop!

Beschädigung oder Zerstörung von Maschinenteilen

Beschädigung oder Zerstörung von Maschinenteilen besteht durch ...

- Kurzschluss oder statische Entladungen (ESD);
- nicht vorhersehbare Motorbewegungen (z. B. ungewollte Drehrichtung, zu hohe Geschwindigkeit oder ruckhafter Lauf);
- unzulässige Betriebszustände bei der Parametrierung, während eine Online-Verbindung zum Gerät besteht.

Schutzmaßnahmen

- Vor allen Arbeiten an den Komponenten des Automationssystems immer die Spannungsversorgung ausschalten.
- Elektronische Bauelemente und Kontakte nur berühren, wenn zuvor ESD-Maßnahmen getroffen wurden.
- Anlagen mit eingebauten Invertern ggf. mit zusätzlichen Überwachungs- und Schutzeinrichtungen nach den jeweils gültigen Sicherheitsbestimmungen ausrüsten (z. B. Gesetz über technische Arbeitsmittel, Unfallverhütungsvorschriften).

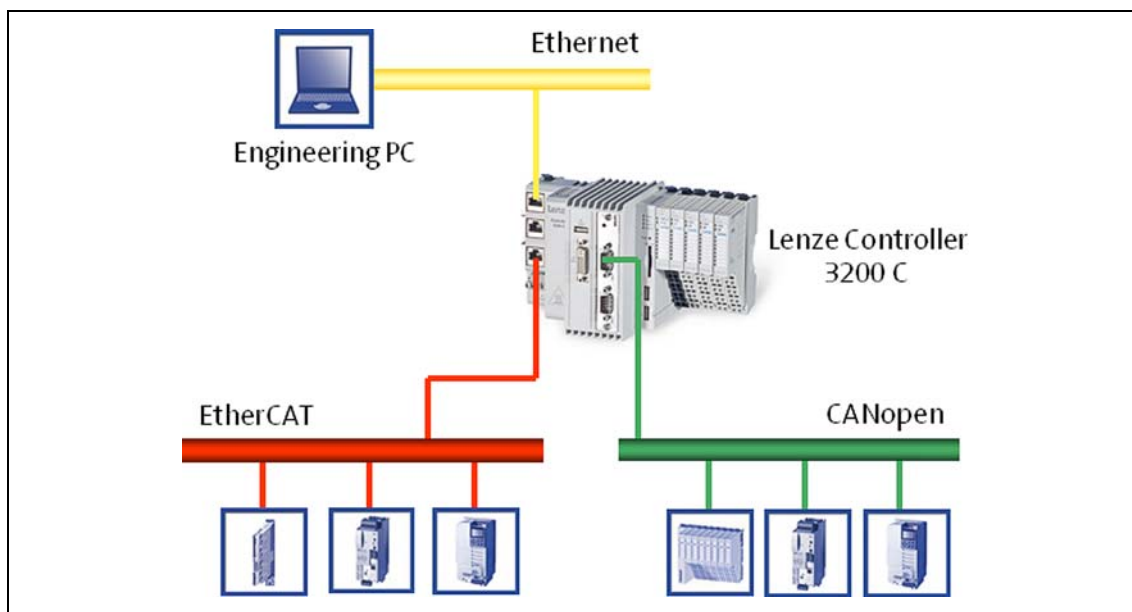
3 Voraussetzungen

3.1 Systemvoraussetzungen

3 Voraussetzungen

Die in diesem Kapitel beschriebenen Voraussetzungen müssen erfüllt sein, damit ein Application Sample genutzt werden kann.

3.1 Systemvoraussetzungen



[3-1] Beispiel: Systemaufbau mit einem Controller 3200 C

Komponenten	Engineering PC	Lenze Controller/Inverter	
Hardware	PC/Notebook	Controller: • 3200 C / p500 / c300 / p300 Inverter: • 8400 BaseLine (CAN) • 8400 StateLine (CAN, EtherCAT) • 9400 HighLine (CAN, EtherCAT)	
Betriebssystem	Windows®	Controller:	
		3200C	Windows® CE
		p500	
		c300	Windows® Embedded Compact 7
		p300	
Enginnering Tool	»PLC Designer« ab Version 3.3 »Engineer« ab Version 2.13	• Controller: Application Software "FAST Runtime" (ehem. "L-force Logic" (LPC 1000)) • Inverter: Firmware ist geräteabhängig.	
Sonstiges	-	Auswählbare Beispielapplikationen für CAN oder EtherCAT	

Ausführliche Informationen zu den Systemvoraussetzungen für die Lenze Engineering Tools finden Sie auf den Download-Seiten der Engineering Tools unter:

www.lenze.com → Download → Software Downloads

3.2

Geräte (Controller, Inverter, Motoren) verdrahten

Weiterführende Informationen zu den elektrischen Anschlüssen finden Sie in den mitgelieferten Dokumentationen. Lesen Sie zuerst die den Geräten beiliegende Montageanleitung, bevor Sie mit den Arbeiten beginnen!

Die Montageanleitung enthält Sicherheitshinweise, die Sie beachten müssen!

Bevor Sie mit einem Beispielprojekt arbeiten können, müssen die Hardware-Komponenten miteinander verbunden werden.

- Verbinden Sie den Controller mit den Feldbus-Teilnehmern (Inverter).
Dazu ein passendes Feldbuskabel verwenden.
- Verbinden Sie die Geräte (Controller, Inverter, Motoren) mit der entsprechenden Spannungsversorgung über die Leistungsanschlüsse.
- Steueranschlüsse verdrahten.
- USB-Diagnoseadapter anschließen.
- Spannungsversorgung der Feldbus-Teilnehmer (Inverter) einschalten.

Wenn die grüne LED "DRV-RDY" blinkt und die rote LED "DRV-ERR" aus ist, ist der Inverter einschaltsbereit und Sie können mit der Inbetriebnahme fortfahren.

3.3 Kommunikation zum Controller einrichten

Verbinden Sie den Engineering PC über ein Netzkabel mit dem Lenze Controller. Der »PLC Designer« greift via Ethernet auf den Controller zu.

Ausführliche Informationen zur Einrichtung der Feldbus-Kommunikation und Inbetriebnahme des Lenze-Automationssystems "Controller-based Automation" finden Sie in diesen Dokumentationen:



Kommunikationshandbuch Controller-based Automation EtherCAT

Hier finden Sie Informationen zur Inbetriebnahme von EtherCAT-Komponenten.



Kommunikationshandbuch Controller-based Automation CANopen

Hier finden Sie Informationen zur Inbetriebnahme von CANopen-Komponenten.

4 Lenze Application Samples

Die Lenze Application Samples sind vorgefertigte Beispielprojekte für die Engineering Tools »Engineer« und »PLC Designer«.

Die Beispielprojekte enthalten gängige Systemkonfigurationen, die typische Anwendungsfälle darstellen. Je nach Anwendungsfall können die Beispielprojekte beliebig erweitert und an die jeweiligen Anforderungen angepasst werden.

So können Sie mit geringem Aufwand ein lauffähiges Lenze-Automationssystem entwickeln.

Aufbau der Application Samples

Jedes Beispielprojekt besteht aus:

- Vorgefertigten Projekten im »Engineer«
- Programmcode und Visualisierungen im »PLC Designer«

Mögliche Betriebs-Modi sind:

- Automatic
- Manual (Handfahr-Betrieb)
- Service
- Homing (Referenzfahrt)

4.1 Übersicht der Application Samples

Produkt/ Applikation	Projektname	Bussystem	Lenze Engineering Tool	
			»PLC Designer« 	»Engineer« 
Inverter Drives 8400				
BaseLine				
"Stellantrieb - Drehzahl"	LAS_40_INTF_Can_84BL_Speed_ [...]	CAN	●	●
StateLine				
"Stellantrieb- Drehzahl"	LAS_40_INTF_Can_84SL_Speed_ [...]	CAN	●	●
	LAS_40_INTF_ETC_84SL_Speed_ [...]	EtherCAT	●	●
HighLine				
"Tabellen- positionierung"	LAS_40_INTF_Can_84HL_TabPos_ [...]	CAN	●	●
	LAS_40_INTF_ETC_84HL_TabPos_ [...]	EtherCAT	●	●
Servo Drives 9400				
HighLine				
"Stellantrieb - Drehzahl"	LAS_40_INTF_Can_94HL_Speed_ [...]	CAN	●	●
	LAS_40_INTF_ETC_94HL_Speed_ [...]	EtherCAT	●	●
"Tabellen- positionierung"	LAS_40_INTF_Can_94HL_TabPos_ [...]	CAN	●	●
	LAS_40_INTF_ETC_94HL_TabPos_ [...]	EtherCAT	●	●

Bezeichnung der Application Samples

LAS_<Sortierschlüssel>_<Kategorie>_<Bussystem>_<Inverter>_<Applikation>_<Version>

Schlüsselwort	Eintrag	Bedeutung
Sortierschlüssel	40	Applikationsbeispiele mit Schnittstellenfunktion (Interface-Anbindung) zur Anbindung der Lenze-Inverter.
Kategorie	INTF	
Bussystem	Can	Projekt für CAN
	ETC	Projekt für EtherCAT
Inverter	84BL	Inverter Drive 8400 BaseLine
	84SL	Inverter Drive 8400 Stateline
	84BH	Inverter Drive 8400 HighLine
	94HL	Servo Drive 9400 HighLine
Applikation	Speed	Applikation "Stellantrieb - Drehzahl"
	TabPos	Applikation "Tabellenpositionierung"
Version	xyyy	Sample-Version • Beispiel: Eintrag '0200' => Version '2.0'

4.2 Wo finden Sie die Application Samples?

Nach erfolgreicher Installation des »Engineer« und »PLC Designer« finden Sie die Application Samples im **Windows® Startmenü** unter:

Start → Alle Programme → Lenze → AppSamples → ...



Tipp!

Die »Engineer«-Beispielprojekte sind im ZIP-Format archiviert, damit Sie diese z. B. per E-Mail verschicken können.

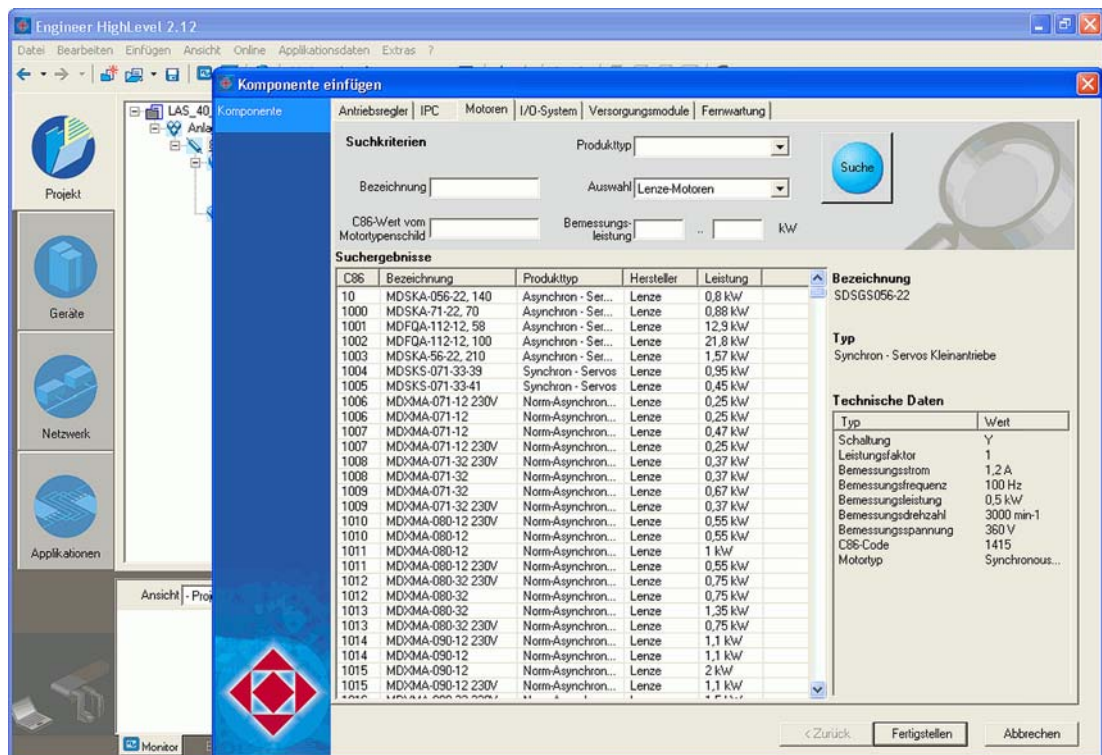
Der »Engineer« unterstützt das Speichern (**Datei → Archiv speichern**) und Öffnen (**Datei → Archiv öffnen**) von Projekten im ZIP-Format.

5 Beispielprojekte im »Engineer« verwenden



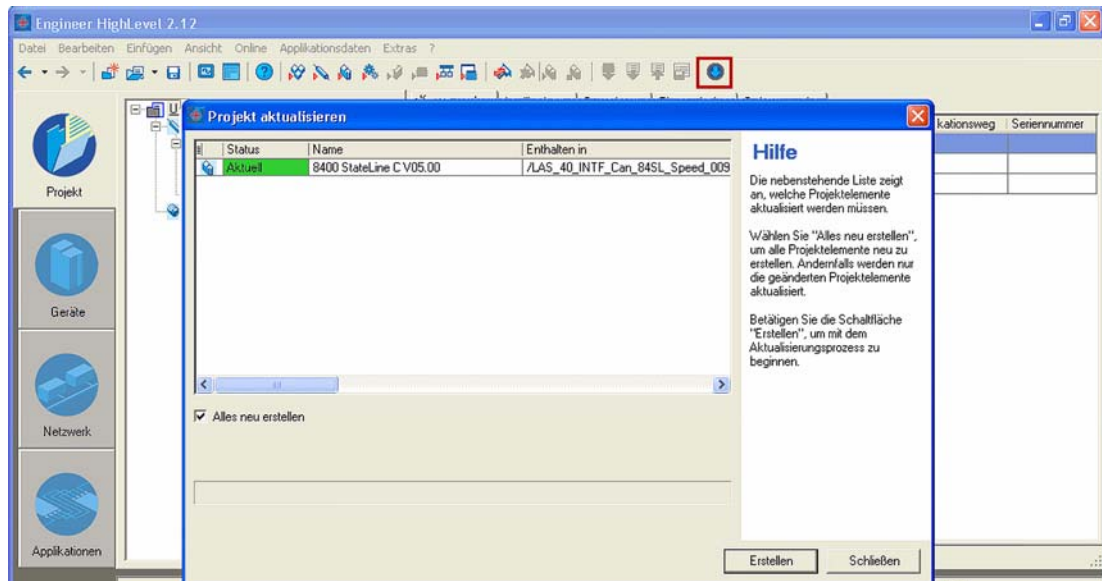
So gehen Sie vor:

1. Den »Engineer« starten.
2. Das gewünschte Beispielprojekt (ZIP-Archiv) öffnen.
 - Doppelklick auf das ZIP-Archiv im **Windows® Startmenü** unter:
Start → Alle Programme → Lenze → AppSamples → ...
- Oder:
 - Menübefehl: **Datei → Archiv öffnen**
3. Im Projektbaum die Achse (Inverter) markieren und den Menübefehl **Einfügen → Komponente** ausführen.
4. Im erscheinenden Auswahldialog unter der Registerkarte "Motoren" den gewünschten Motor auswählen.



5. Mit der Schaltfläche **Fertigstellen** den ausgewählten Motor in den Projektbaum einfügen.
6. Mit dem Menübefehl **Einfügen → Komponente** oder **Einfügen → Getriebe** gegebenenfalls weitere Geräte oder ein Getriebe einfügen.
7. Nicht benötigte Geräte im Projektbaum markieren und entfernen, entweder mit der Delete-Taste oder dem Menübefehl.

8. Mit der Schaltfläche  das Projekt aktualisieren.



- Wählen Sie die Option "Alles neu Erstellen".
- Betätigen Sie die Schaltfläche **Erstellen**, um das Projekt zu aktualisieren.

9. Mit der Schaltfläche  online gehen.

Nach erfolgreicher Verbindung mit dem Inverter wird in der Statuszeile folgender Status angezeigt:



10. Mit der Schaltfläche  den Parametersatz zum Gerät übertragen.

Dieser Befehl bewirkt, dass die Applikation und die Parametereinstellungen des »Engineer«-Projektes die Parametereinstellungen im Inverter überschreiben.

11. Optional: Gewünschte Kommunikationseinstellungen (z. B. Netzwerkadresse, Übertragungsrate) anpassen.

12. Mit der Schaltfläche  den Parametersatz speichern.

13. Netz einschalten.

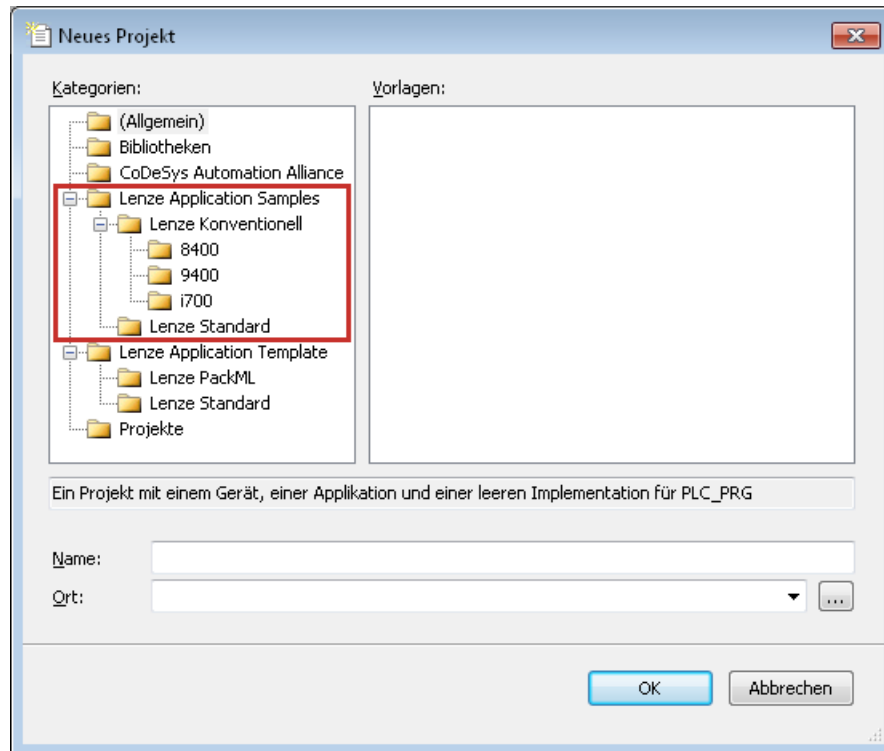
Die Kommunikationseinstellungen werden in das Gerät übertragen.

6 Beispielprojekte im »PLC Designer« verwenden



So gehen Sie vor:

1. Den »PLC Designer« starten.
2. Mit dem Menübefehl **Datei → Neues Projekt** ein neues Projekt erstellen.
3. Unter der Kategorie "Lenze Application Samples" ein Beispielprojekt aus den Ordnern '8400' oder '9400' auswählen.



4. Mit der Schaltfläche **OK** das Beispielprojekt öffnen.
 - ▶ [Übersicht der Application Samples](#) (20)
 - ▶ [Beispielprojekt mit der Applikation "Stellantrieb - Drehzahl"](#) (27)
 - ▶ [Beispielprojekt mit der Applikation "Tabellenpositionierung"](#) (28)
5. Die Beispielprojekte, je nach Anwendungsfall, erweitern und anpassen.
 - ▶ [Beispielprojekte erweitern \(Geräte hinzufügen\)](#) (43)
 - ▶ [Programmbausteine \(PRG\)](#) (29) anpassen.
6. Mit dem Menübefehl **Erstellen → Übersetzen** oder mit der Funktionstaste **<F11>** den PLC-Programmcode zu übersetzen.
 - Traten bei der Übersetzung Fehler auf, können Sie diese anhand der »PLC Designer«-Fehlermeldungen lokalisieren und entsprechend korrigieren. Übersetzen Sie danach den Programmcode erneut.
 - Wenn bei der Übersetzung keine Fehler auftraten, speichern Sie das »PLC Designer«-Projekt im Projektordner mit dem Menübefehl **Datei → Projekt speichern / Projekt speichern unter ...**

7. Mit dem Menübefehl **Online → Einloggen** oder mit **<Alt>+<F8>** in den Controller einloggen.
 - Das PLC-Programm muss dazu fehlerfrei sein.
 - Mit dem Einloggen wird die [Steuerungskonfiguration \(EtherCAT/CAN\)](#) (📖 26) und das PLC-Programm in den Controller geladen. Dabei wird eine eventuell vorhandene Konfiguration und ein eventuell vorhandenes PLC-Programm überschrieben.
8. Mit dem Menübefehl **Debug → Start** oder mit der Funktionstaste **<F5>** das PLC-Programm starten.



Tipp!

Richten Sie das PLC-Programm als "**Boot-Projekt**" ein, um es nach einem Neustart des Gerätes automatisch zu laden.

Menübefehl: **Online → Erzeuge Bootapplikation für L-force Controller**

6.1 Steuerungskonfiguration (EtherCAT/CAN)

EtherCAT

- Der EtherCAT-Master ist die Steuereinheit der Systemkonfiguration.
- Unterhalb des EtherCAT-Masters ist der Inverter (Stations-ID = 1001) angehängt.
- Unterhalb des Inverters sind die Achsdaten in Form eines zusätzlichen Knotens (LC_Drive) verfügbar und über die Funktionsbaustein-Instanz [L_LCB_AXIS_REF](#) (📖 67) abrufbar.



Kommunikationshandbuch Controller-based Automation EtherCAT

Hier finden Sie ausführliche Informationen zur Inbetriebnahme von EtherCAT-Komponenten.

CAN

- Der CAN-Master (Node-ID = 127) ist die Steuereinheit der Systemkonfiguration.
- Unterhalb des CAN-Masters ist der Inverter (Node-ID = 5) angehängt.
- Unterhalb des Inverters sind die Achsdaten in Form eines zusätzlichen Knotens (LC_Drive) verfügbar und über die Funktionsbaustein-Instanz [L_LCB_AXIS_REF](#) (📖 67) abrufbar.
- Es wird jeweils ein PDO für CAN-Input und ein PDO für CAN-Output verwendet.
- Der Übertragungsmodus ist Sync-gesteuert (10 ms).



Kommunikationshandbuch Controller-based Automation CANopen

Hier finden Sie ausführliche Informationen zur Inbetriebnahme von CANopen-Komponenten.

Erweiterung der Steuerungskonfiguration

Weiterführende Informationen zur Erweiterung der Steuerungskonfiguration finden Sie hier:

► [Beispielprojekte erweitern \(Geräte hinzufügen\)](#) (📖 43)

6.2 Beispielprojekt mit der Applikation "Stellantrieb - Drehzahl"

Hier werden folgende Funktionsbibliotheken und Funktionsbausteine verwendet:

- ▶ [Funktionsbibliothek L_LCB_LogicControlBasic](#) (📖 65)
- ▶ [Funktionsbibliothek L_DCO_DriveCommunication](#) (📖 103)
- ▶ [L_LCB_ActuatorSpeed](#) (📖 73)

Übersicht der Funktionen

Modus	Beschreibung
Automatic	Im Automatic-Modus läuft eine Sequenz über eine Endlosschleife. Vorgabe: 1. 3 Sekunden mit der Geschwindigkeit '1' fahren. 2. Dann 5 Sekunden mit der Geschwindigkeit '2' fahren. 3. Anschließend wieder mit der Geschwindigkeit '1' fahren.
Manual	Im Manual-Modus (Handfahrbetrieb) ist der Inverter durch Setzen einzelner Steuer-Bits (xJog1Set, xJog2Set, xDriveSetQsp, xResetError, ...) manuell steuerbar. Beispielsweise ist der Inverter zum Reinigen oder Wechseln des Werkzeugs im Handfahrbetrieb über die Steuer-Bits "xJog1Set" und "xJog2Set" steuerbar.
Service	Der Service-Modus dient zum Abgleich des Inverters. Vorgabe: 1. 1 Sekunde in positive Richtung fahren. 2. 1 Sekunde stoppen. 3. 1 Sekunde in negative Richtung fahren.

6.3 Beispielprojekt mit der Applikation "Tabellenpositionierung"

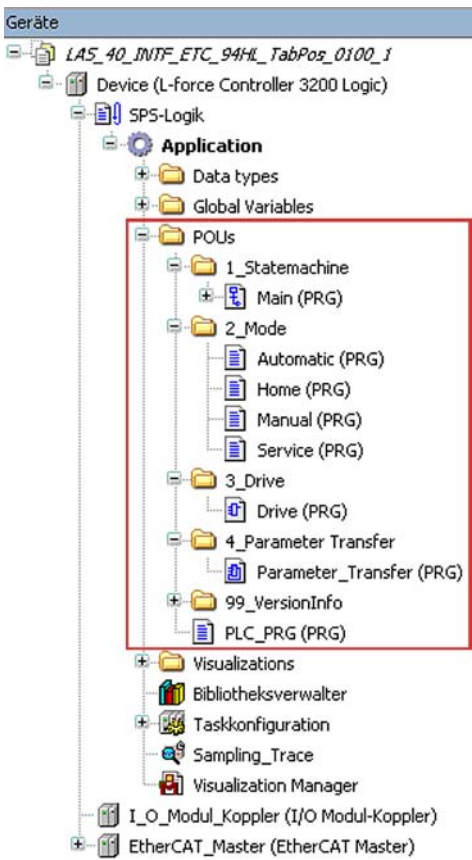
Hier werden folgende Funktionsbibliotheken und Funktionsbausteine verwendet:

- ▶ [Funktionsbibliothek L_LCB_LogicControlBasic](#) (📖 65)
- ▶ [Funktionsbibliothek L_DCO_DriveCommunication](#) (📖 103)
- ▶ [L_LCB_TablePositioning](#) (📖 77)
- ▶ [L_LCB_SetAxisData](#) (📖 90)

Übersicht der Funktionen

Modus	Beschreibung
Automatic	Im Automatic-Modus läuft eine Sequenz über eine Endlosschleife. Vorgabe: Wenn der Antrieb über den Homing-Modus (siehe unten) referenziert ist ... 1. zu Position '1' fahren, 2. zu Position '2' fahren, 3. wieder zu Position '1' fahren.
Manual	Im Manual-Modus (Handfahrbetrieb) ist der Inverter durch Setzen einzelner Steuer-Bits (xManualPos, xManualNeg, xDriveSetQsp, xResetError, ...) manuell steuerbar. Beispielsweise ist der Inverter zum Reinigen oder Wechseln des Werkzeugs im Handfahrbetrieb über die Steuer-Bits "xManualPos" und "xManualNeg" steuerbar.
Service	Der Service-Modus dient zum Abgleich des Inverters. Vorgabe: 1. 1 Sekunde in positive Richtung fahren. 2. 1 Sekunde stoppen. 3. 1 Sekunde in negative Richtung fahren.
Homing	Im Homing-Modus wird der Antrieb referenziert. Entweder wird die Referenzposition direkt gesetzt oder eine Referenzfahrt gestartet.

6.4 Programmbausteine (PRG)



Main (PRG)
Das Programm "Main (PRG)" enthält das Hauptprogramm zur Abarbeitung der Programmabfolgen: Initialisierung des Inverters und Auswahl der unterschiedlichen Modi.

Automatic (PRG) / Home (PRG) / Manual (PRG) / Service (PRG)
Hier sind die Programme zu den einzelnen Modi der Applikation enthalten.

- Home (PRG) nicht bei "Stellantrieb - Drehzahl"!

Drive (PRG)

- Das Programm "Drive (PRG)" steuert den Inverter – je nach Beispielprojekt – mit dem Funktionsbaustein [L_LCB_ActuatorSpeed](#) (□ 73) oder [L_LCB_TablePositioning](#) (□ 77) an.
- Achsdaten sind über die Funktionsbaustein-Instanz [L_LCB_AXIS_REF](#) (□ 67) abrufbar.
- Für die "Tabellenpositionierung" werden Maschinenkonstanten über den Funktionsbaustein [L_LCB_SetAxisData](#) (□ 90) aufbereitet zur Verfügung gestellt. – Nicht bei "Stellantrieb - Drehzahl"!

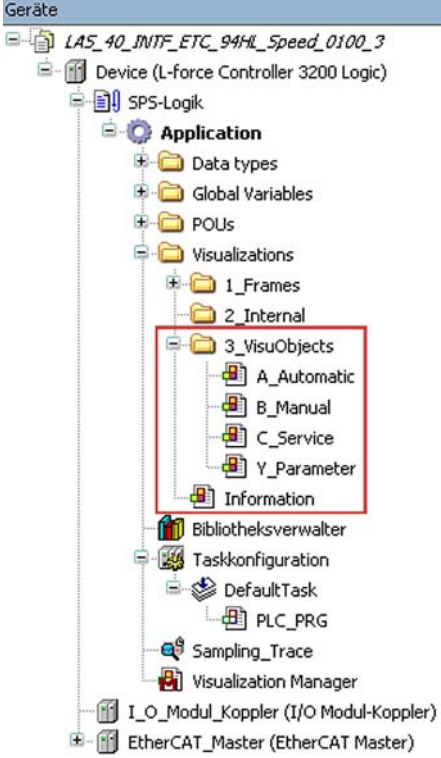
Parameter_Transfer (PRG)
Das Programm "Parameter_Transfer (PRG)" enthält Aufrufe der Parameter-Funktionsbausteine [L_DCO_ReadDriveParameter](#) (□ 106) und [L_DCO_WriteDriveParameter](#) (□ 112), mit denen einzelne Parameter gelesen oder geschrieben werden können.

PLC_PRG (PRG)
Das Programm "PLC_PRG (PRG)" enthält alle Programmaufrufe.

[6-1] Beispielprojekt: Tabellenpositionierung - Servo Drive 9400 HighLine - EtherCAT

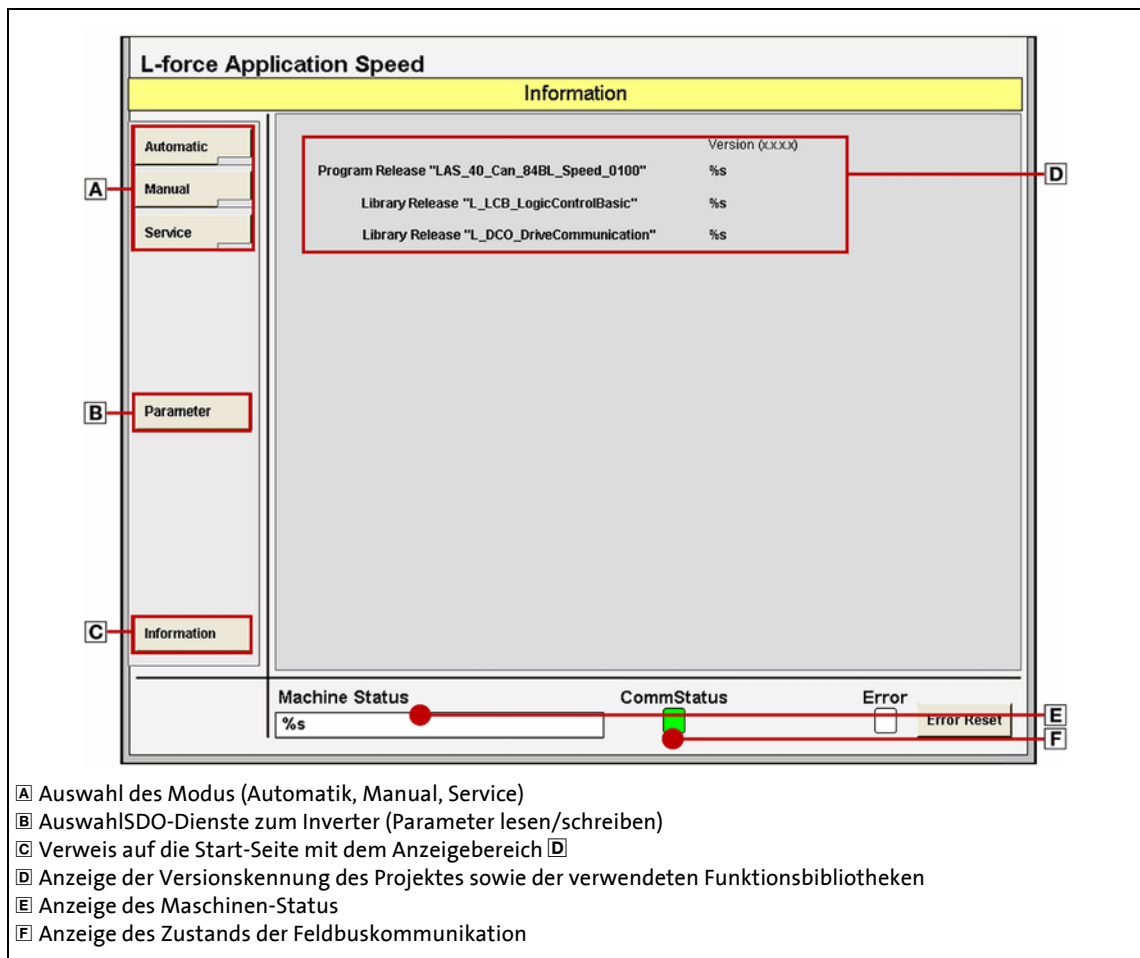
6.5 Visualisierungen der Applikation "Stellantrieb - Drehzahl"

Das Beispielprojekt mit der Applikation "Stellantrieb - Drehzahl" enthält Visualisierungen zur Bedienung, zur Einstellung und Anzeige von Parametern und zur Anzeige von Fehlern.

	"Information" (Startseite) enthält allgemeine Informationen zum Beispielprojekt. Zudem gibt es Visualisierungen zu den Betriebs-Modi: • Automatic • Manual (Handfahr-Betrieb) • Service "Y_Parameter" ist die Visualisierung der Parameter-Funktionsbausteine zum Lesen oder Schreiben einzelner Parameter.
--	--

[6-2] **Beispielprojekt:** Stellantrieb - Drehzahl - Servo Drive 9400 HighLine - EtherCAT

6.5.1 Information (Startseite)



[6-3] Startseite der Visualisierung in der Applikation "Stellantrieb - Drehzahl"

Anzeige von Fehlermeldungen

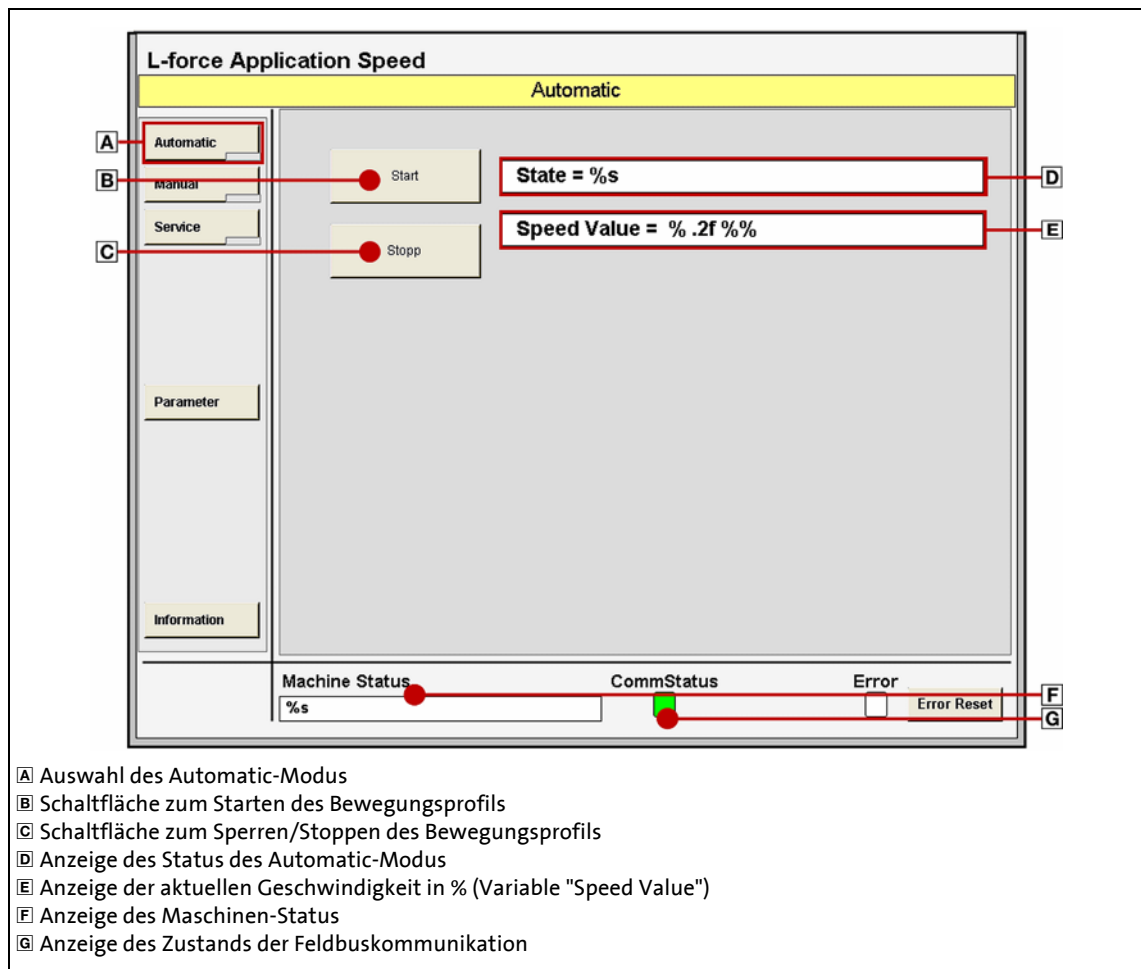
Das Kontrollfeld **CommStatus** (F) zeigt den Zustand der Feldbuskommunikation an.

Bei einem Fehler der Feldbuskommunikation müssen der Controller und die Inverter neu gestartet werden, da die Feldbuskommunikation nicht automatisch zurückgesetzt wird.

Das Kontrollfeld **Error** zeigt Fehler in der Applikation und Fehler im Inverter an.

Die Schaltfläche **Error Reset** setzt die Fehlermeldungen zurück.

6.5.2 Automatic-Modus



[6-4] Visualisierung des Automatic-Modus in der Applikation "Stellantrieb - Drehzahl"

Anzeige von Fehlermeldungen

Das Kontrollfeld **CommStatus** (G) zeigt den Zustand der Feldbuskommunikation an.

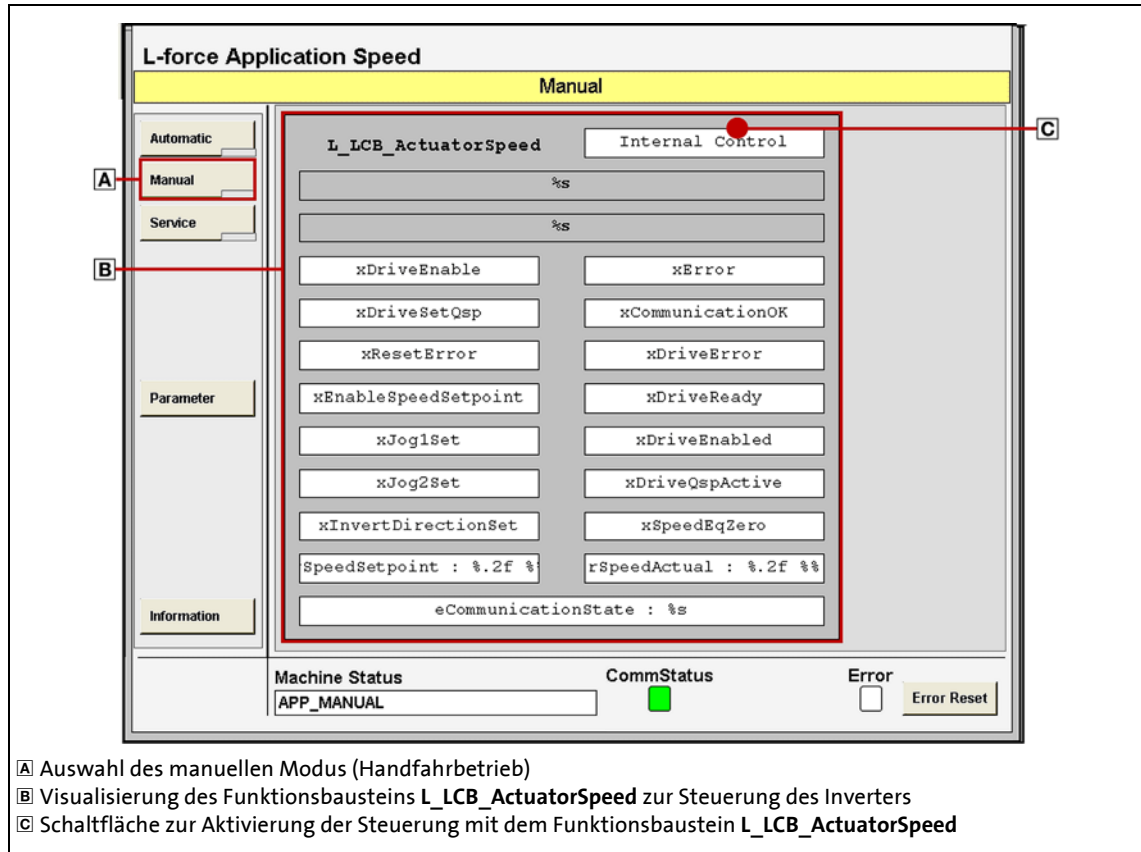
Bei einem Fehler der Feldbuskommunikation müssen der Controller und die Inverter neu gestartet werden, da die Feldbuskommunikation nicht automatisch zurückgesetzt wird.

Das Kontrollfeld **Error** zeigt Fehler in der Applikation und Fehler im Inverter an.

Die Schaltfläche **Error Reset** setzt die Fehlermeldungen zurück.

6.5.3 Manual-Modus (Handfahr-Betrieb)

Der Inverter ist über die Visualisierung des Funktionsbausteins [L_LCB_ActuatorSpeed](#) (73) im Handfahrbetrieb steuerbar.

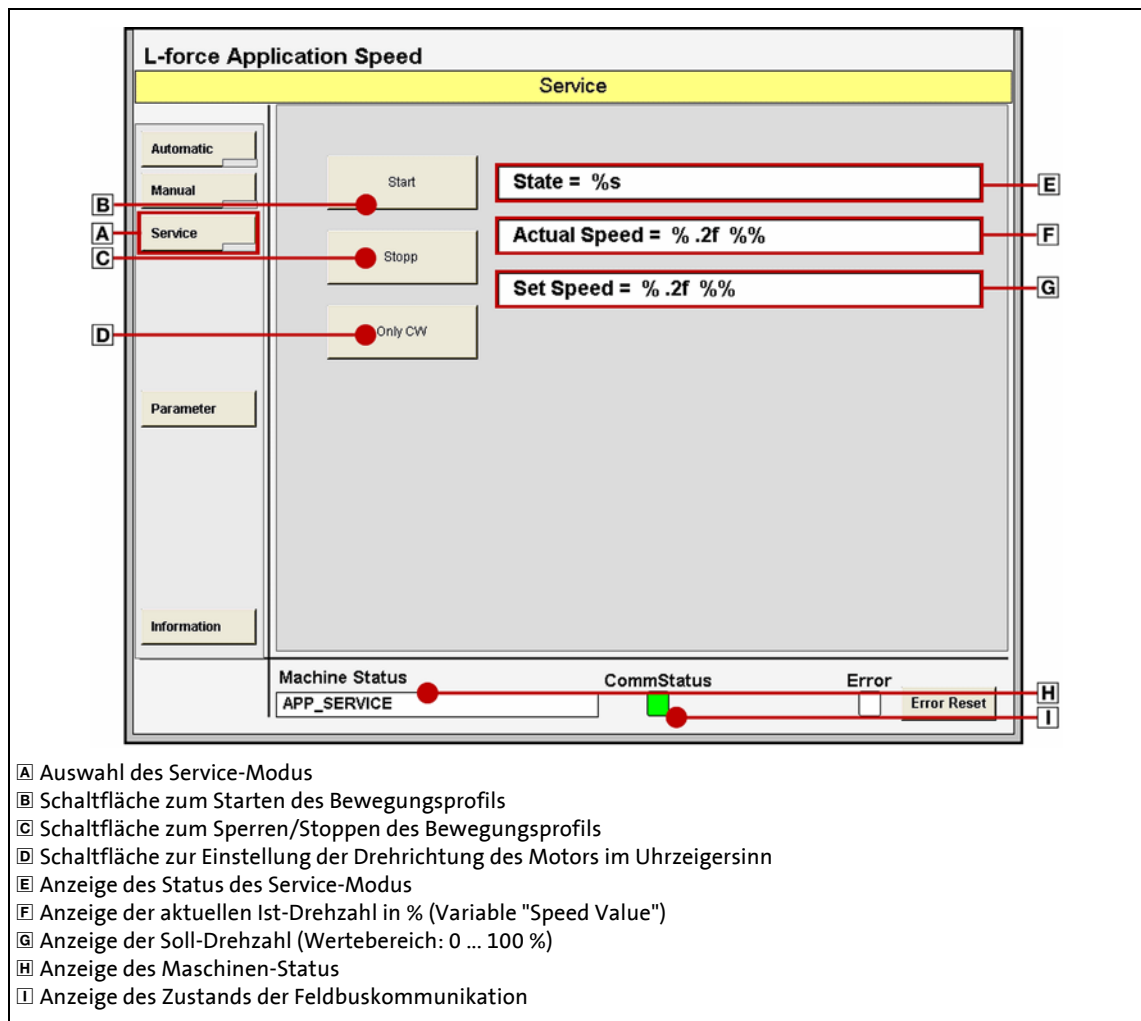


[6-5] Visualisierung des manuellen Modus in der Applikation "Stellantrieb - Drehzahl"

- Aktivieren Sie die Schaltfläche **Internal Control** (C), um den Inverter über die Visualisierung bedienen zu können.
- Geben Sie den Inverter über den Eingang **xDriveEnable** frei, um Sollwerte manuell vorgeben zu können.
- Betätigen Sie die Schaltfläche **xDriveSetQsp**, um den Schnellhalt (QSP) zu aktivieren.

6.5.4

Service-Modus



[6-6] Visualisierung des Service-Modus in der Applikation "Stellantrieb - Drehzahl"

Anzeige von Fehlermeldungen

Das Kontrollfeld **CommStatus** (I) zeigt den Zustand der Feldbuskommunikation an.

Bei einem Fehler der Feldbuskommunikation müssen der Controller und die Inverter neu gestartet werden, da die Feldbuskommunikation nicht automatisch zurückgesetzt wird.

Das Kontrollfeld **Error** zeigt Fehler in der Applikation und Fehler im Inverter an.

Die Schaltfläche **Error Reset** setzt die Fehlermeldungen zurück.

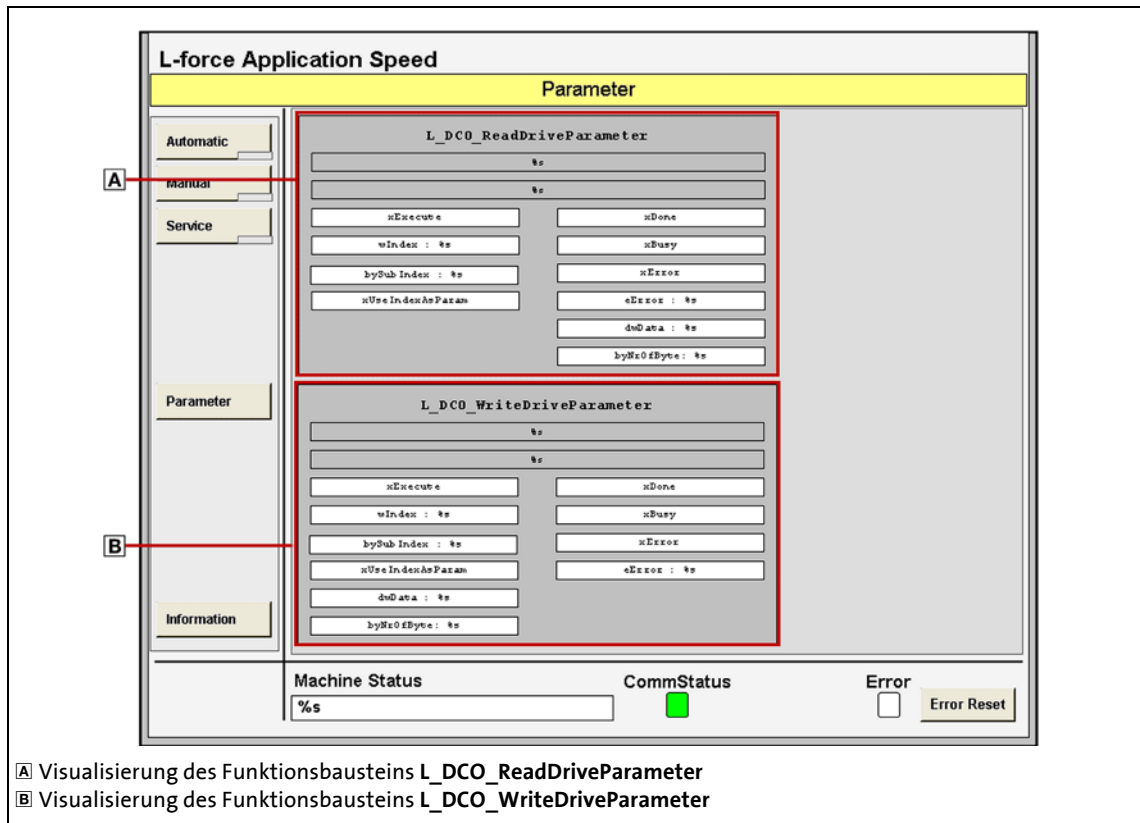
6.5.5 Parameter

Mit der Schaltfläche **Parameter** wird die Visualisierung der Parameter-Funktionsbausteine geöffnet:

▶ [L_DCO_ReadDriveParameter](#) (106)

▶ [L_DCO_WriteDriveParameter](#) (112)

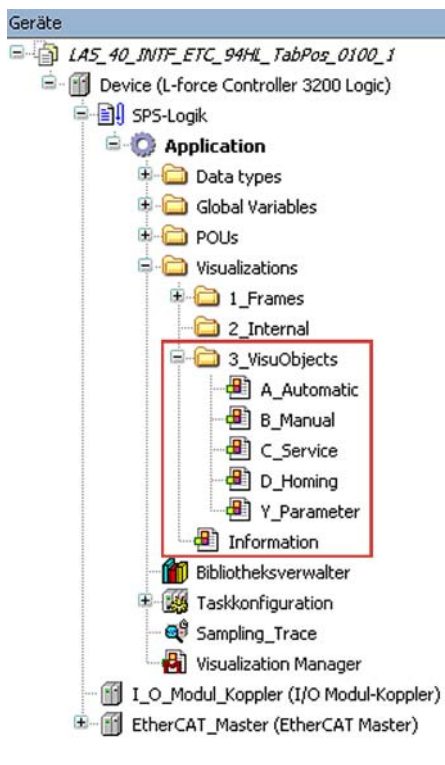
Mit diesen Funktionsbausteinen können einzelne Parameter gelesen oder geschrieben werden. Der zuletzt ausgewählte Modus bleibt aktiviert.



[6-7] Visualisierung der Parameter-Funktionsbausteine in der Applikation "Stellantrieb - Drehzahl"

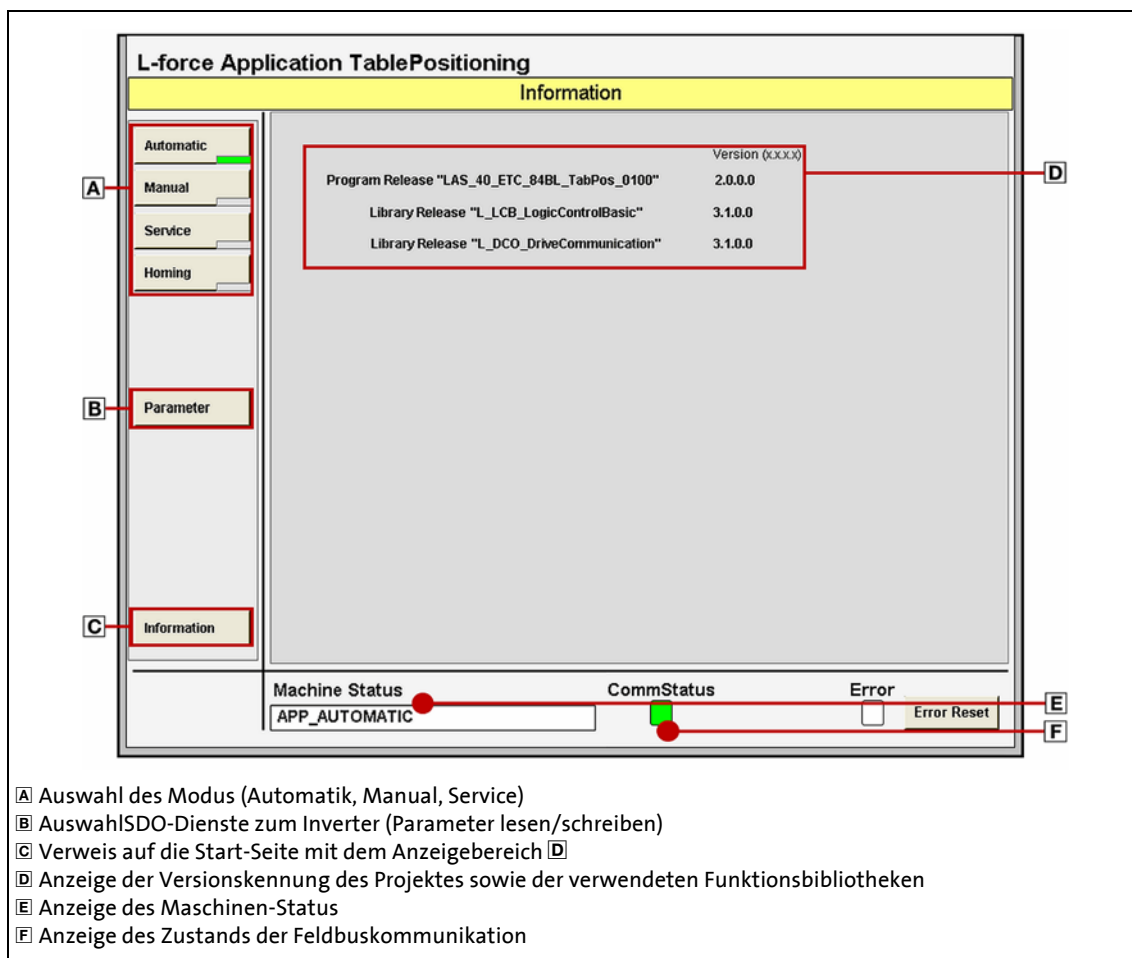
6.6 Visualisierungen der Applikation "Tabellenpositionierung"

Das Beispielprojekt mit der Applikation "Tabellenpositionierung" enthält Visualisierungen zur Bedienung, zur Einstellung und Anzeige von Parametern und zur Anzeige von Fehlern.

	"Information" (Startseite) enthält allgemeine Informationen zum Beispielprojekt. Zudem gibt es Visualisierungen zu den Betriebs-Modi: • Automatic • Manual (Handfahr-Betrieb) • Service • Homing (Referenzfahrt) "Y_Parameter" ist die Visualisierung der Parameter-Funktionsbausteine zum Lesen oder Schreiben einzelner Parameter.
--	--

[6-8] **Beispielprojekt:** Tabellenpositionierung - Servo Drive 9400 HighLine - EtherCAT

6.6.1 Information (Startseite)



- [A] Auswahl des Modus (Automatik, Manual, Service)
- [B] Auswahl SDO-Dienste zum Inverter (Parameter lesen/schreiben)
- [C] Verweis auf die Start-Seite mit dem Anzeigebereich [D]
- [D] Anzeige der Versionskennung des Projektes sowie der verwendeten Funktionsbibliotheken
- [E] Anzeige des Maschinen-Status
- [F] Anzeige des Zustands der Feldbuskommunikation

[6-9] Startseite der Visualisierung in der Applikation "Tabellenpositionierung"

Anzeige von Fehlermeldungen

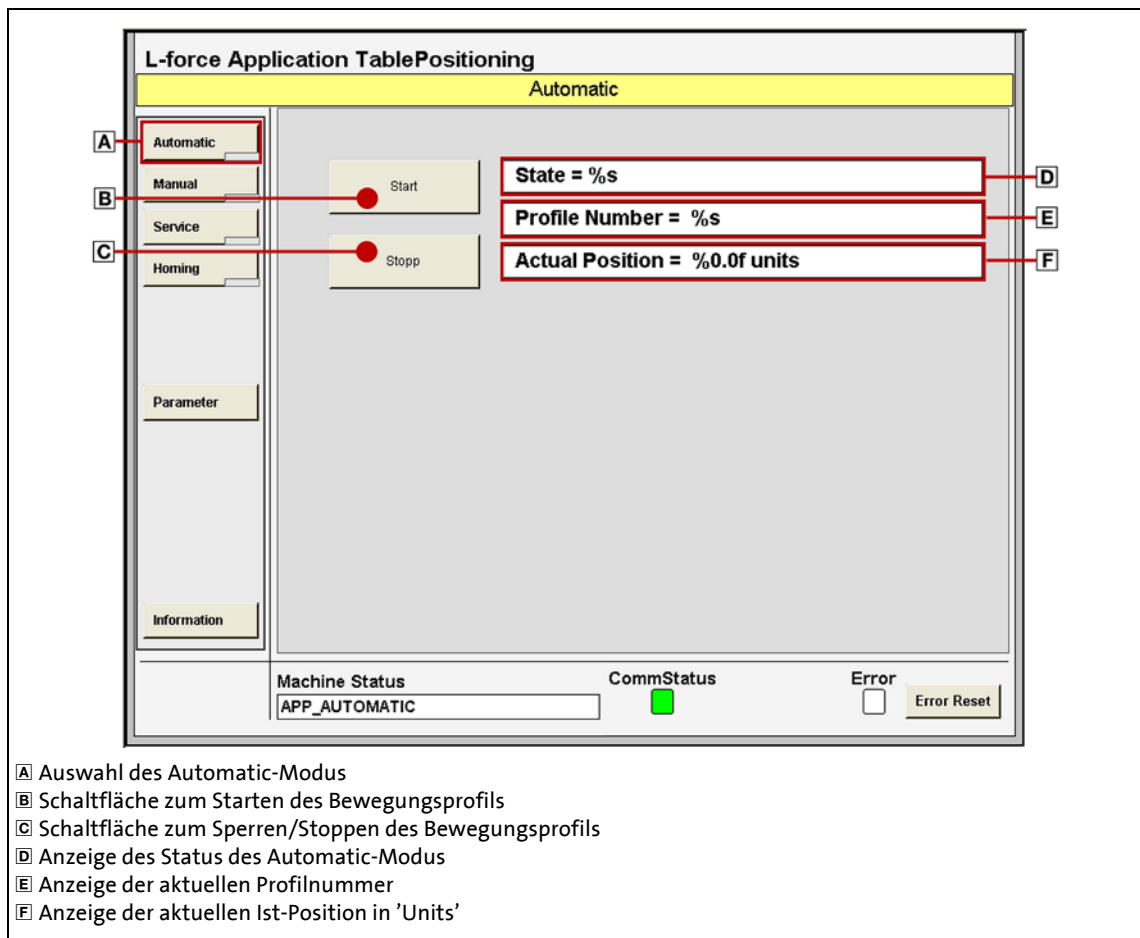
Das Kontrollfeld **CommStatus** (F) zeigt den Zustand der Feldbuskommunikation an.

Bei einem Fehler der Feldbuskommunikation müssen der Controller und die Inverter neu gestartet werden, da die Feldbuskommunikation nicht automatisch zurückgesetzt wird.

Das Kontrollfeld **Error** zeigt Fehler in der Applikation und Fehler im Inverter an.

Die Schaltfläche **Error Reset** setzt die Fehlermeldungen zurück.

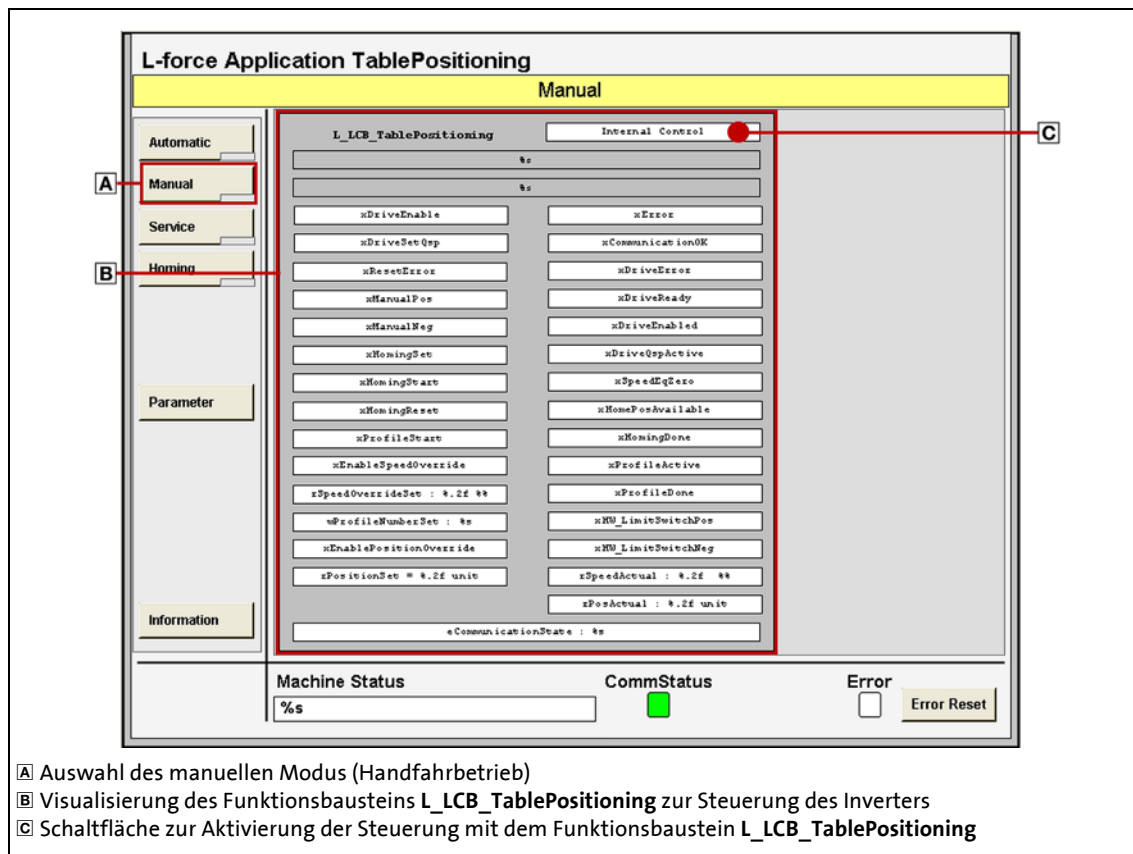
6.6.2 Automatic-Modus



[6-10] Visualisierung des Automatic-Modus in der Applikation "Tabellenpositionierung"

6.6.3 Manual-Modus (Handfahr-Betrieb)

Der Inverter ist über die Visualisierung des Funktionsbausteins **L_LCB_TablePositioning** im Handfahrbetrieb steuerbar.



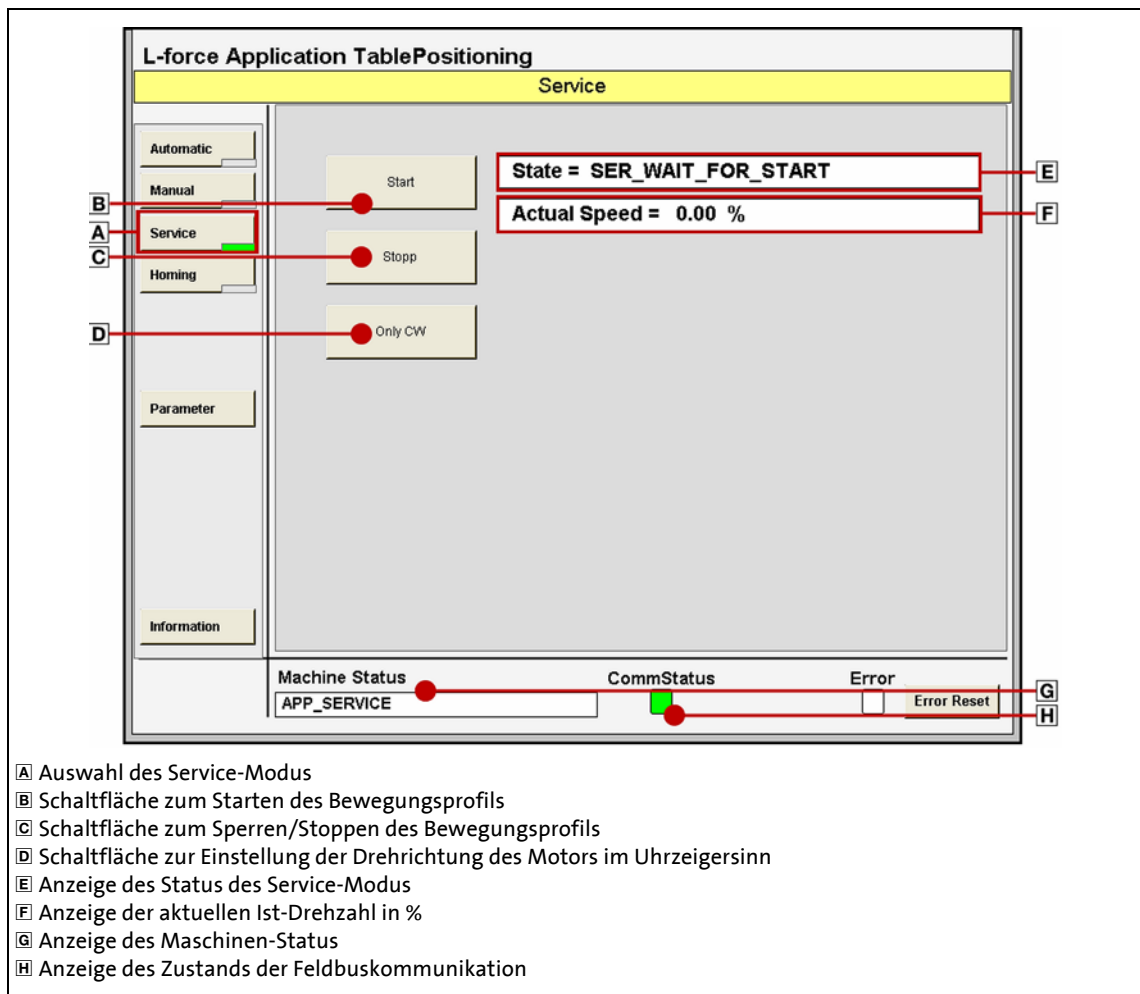
[6-11] Visualisierung des manuellen Modus in der Applikation "Tabellenpositionierung"

- Aktivieren Sie die Schaltfläche **Internal Control** (Ⓒ), um den Inverter über die Visualisierung bedienen zu können.
- Geben Sie den Inverter über den Eingang **xDriveEnable** frei, um Sollwerte manuell vorgeben zu können.
- Betätigen Sie die Schaltfläche **xDriveSetQsp**, um den Schnellhalt (QSP) zu aktivieren.

Weiterführende Informationen zum Funktionsbaustein **L_LCB_TablePositioning** finden Sie hier:

► [L_LCB_TablePositioning](#) (73)

6.6.4 Service-Modus



[6-12] Visualisierung des Service-Modus in der Applikation "Tabellenpositionierung"

Anzeige von Fehlermeldungen

Das Kontrollfeld **CommStatus** (H) zeigt den Zustand der Feldbuskommunikation an.

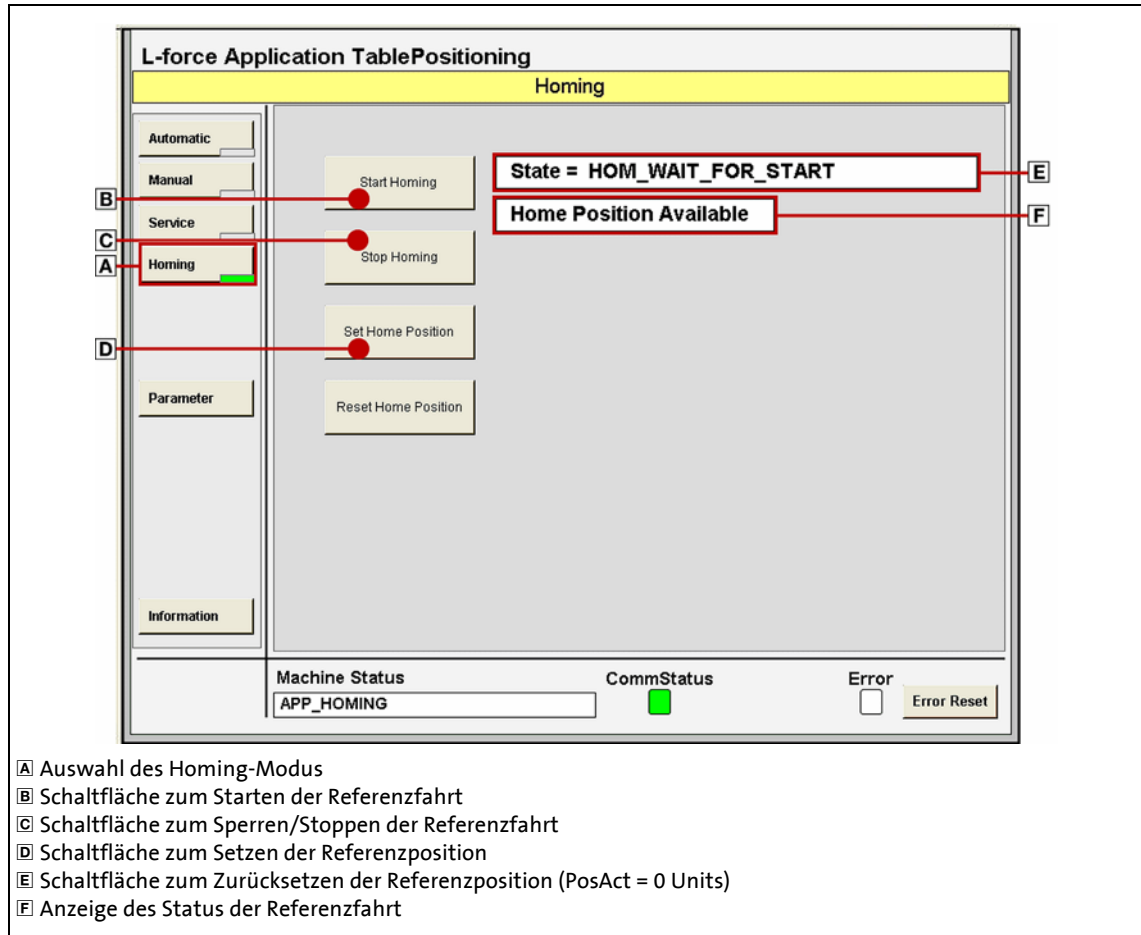
Bei einem Fehler der Feldbuskommunikation müssen der Controller und die Inverter neu gestartet werden, da die Feldbuskommunikation nicht automatisch zurückgesetzt wird.

Das Kontrollfeld **Error** zeigt Fehler in der Applikation und Fehler im Inverter an.

Die Schaltfläche **Error Reset** setzt die Fehlermeldungen zurück.

6.6.5 Homing-Modus (Referenzfahrt)

Im Inverter ist in der Lenze Standard-Einstellung der Homing-Modus '12' aktiv (auf negativen Endschalter fahren / Digitaler Eingang 3).



[6-13] Visualisierung des Homing-Modus in der Applikation "Tabellenpositionierung"

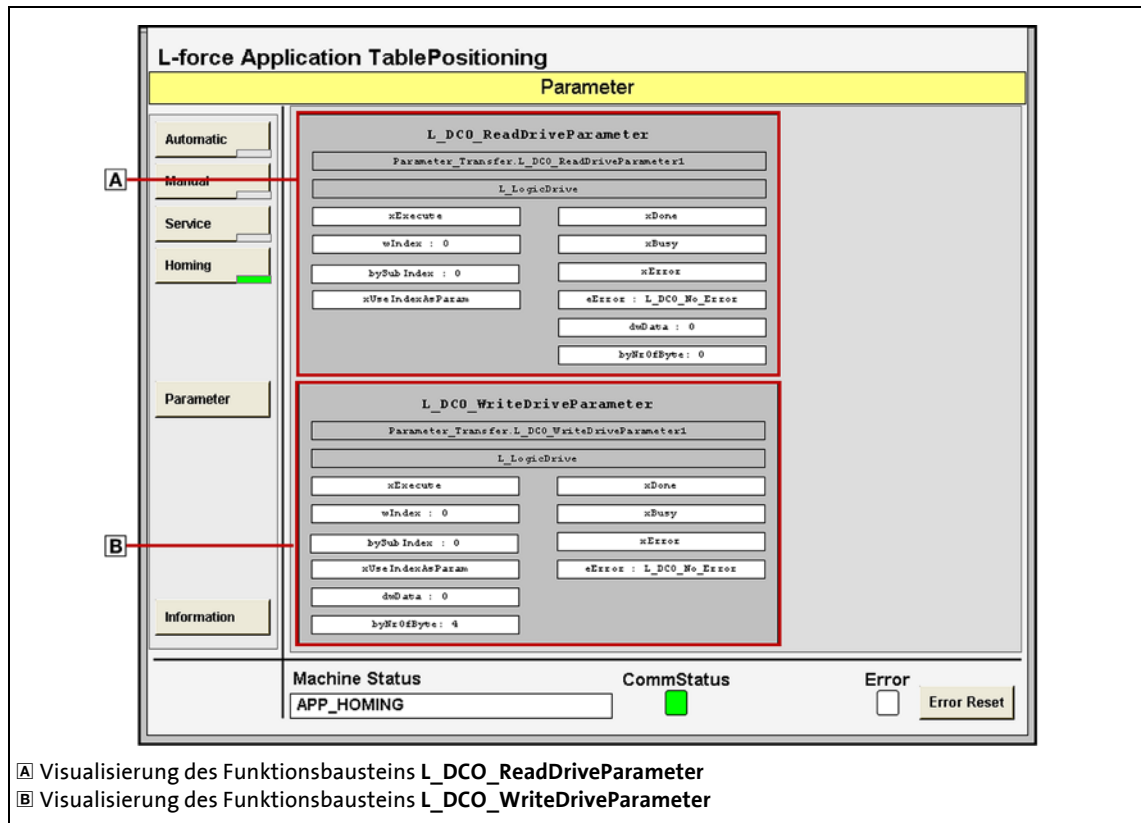
6.6.6 Parameter

Mit der Schaltfläche **Parameter** wird die Visualisierung der Parameter-Funktionsbausteine geöffnet:

▶ [L_DCO_ReadDriveParameter](#) (106)

▶ [L_DCO_WriteDriveParameter](#) (112)

Mit diesen Funktionsbausteinen können einzelne Parameter gelesen oder geschrieben werden. Der zuletzt ausgewählte Modus bleibt aktiviert.



[6-14] Visualisierung der Parameter-Funktionsbausteine in der Applikation "Tabellenpositionierung"

6.7 Beispielprojekte erweitern (Geräte hinzufügen)

In diesem Kapitel erfahren Sie, wie Sie die Steuerungskonfiguration manuell erweitern können.

Befolgen Sie die in den folgenden Abschnitten beschriebene Vorgehensweise, um ein bestehendes Beispielprojekt individuell zu erweitern.

Passen Sie für jeden weiteren Teilnehmer die Kommunikationseinstellungen an.

6.7.1 EtherCAT-Feldgeräte anhängen



Hinweis!

- Die Reihenfolge der EtherCAT-Slaves im Gerätebaum muss der physikalischen Anordnung der EtherCAT-Topologie entsprechen.
- SoftMotion-Betrieb ist nur mit EtherCAT-Slaves möglich, welche die CiA402-Applikation verwenden (SM_Drives, z. B. Servo Drives 9400 HighLine CiA402).
- Wählen Sie Zykluszeiten, gemäß den technischen Daten, von 1 ... 10 ms.



Tipp!

Feldbus-Scan

Der »PLC Designer« bietet einen "Feldbus-Scan", der die am Feldbus angeschlossenen Geräte automatisch erkennt. Führen Sie dazu im Kontextmenü des EtherCAT-Masters den Befehl **Geräte suchen** aus.

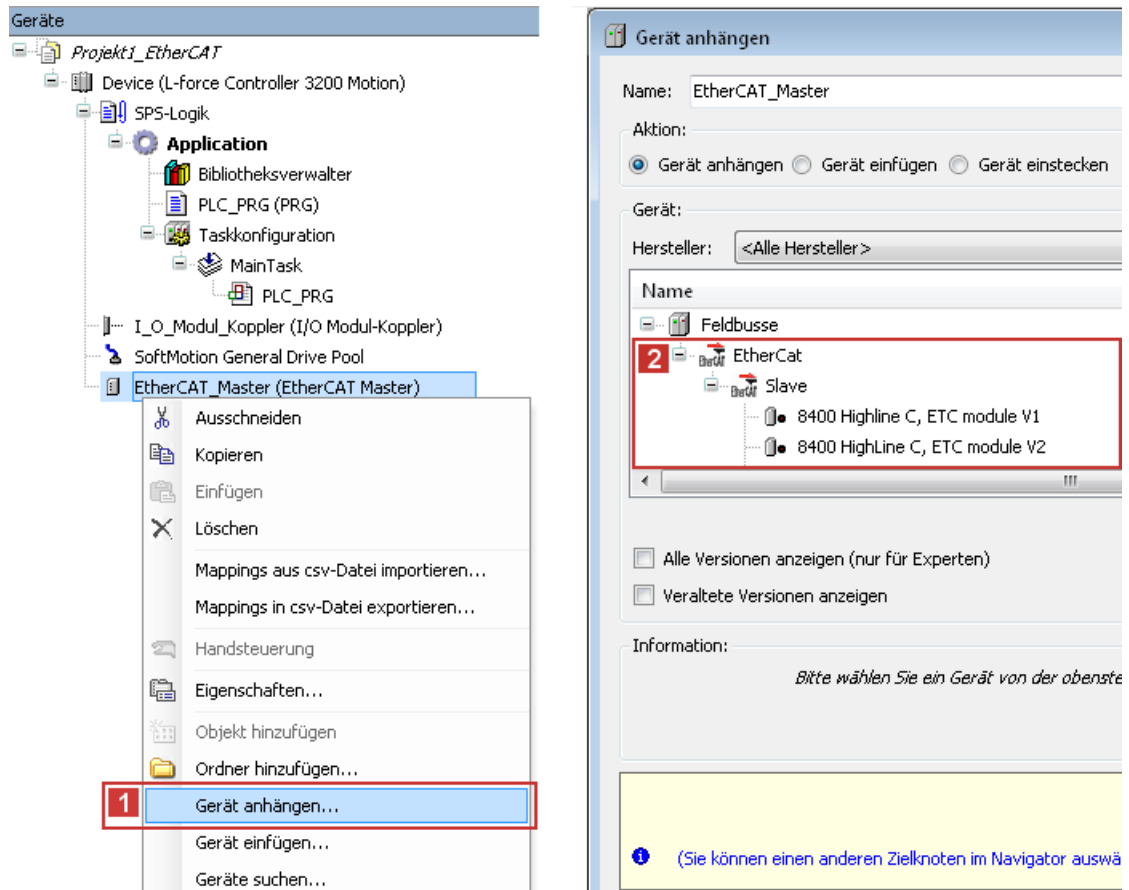
Fehlende Geräte oder Geräte anderer Hersteller

Um fehlende Geräte oder Geräte anderer Hersteller einzubinden, sind die entsprechenden Gerätebeschreibungsdateien des Herstellers erforderlich. Im »PLC Designer« können Gerätebeschreibungsdateien vom Typ *.XML, *.devdesc.XML, *.EDS, *.DCF und *.GSx über den Menübefehl **Tools → Geräte-Repository...** importiert werden.



So hängen Sie EtherCAT-Feldgeräte manuell an:

1. Im Kontextmenü des EtherCAT-Master mit dem Befehl **1** **Gerät anhängen** einen EtherCAT-Slave anhängen.



Wählen Sie aus der **2** Auswahlliste ein Feldgerät. Es sind nur Geräte auswählbar, deren EtherCAT-Gerätebeschreibungsdateien im »PLC Designer« importiert wurden.

Wiederholen Sie den Befehl **1** **Gerät anhängen** solange, bis alle am Feldbus teilnehmenden Slaves in der EtherCAT-Konfiguration eingebunden sind.

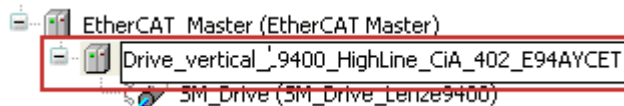
2. Den eingefügten Slaves sinnvolle Bezeichnungen geben (z. B. "Drive_vertical").

Die Bezeichnungen dürfen ...

- nur die Zeichen "A ... Z", "a ... z", "0 ... 9" oder "_" enthalten;
- nicht mit einer Ziffer beginnen.

Durch einen Mausklick auf das Element wird die Bezeichnung zur Eingabe freigegeben.

Beispiel:



Wir empfehlen bei Servo Drives 9400 und Inverter Drives 8400 die gleiche Bezeichnung zu verwenden, die unter der Geräte-Codestelle **C00199** eingetragen ist.

Unterhalb des Inverters sind die Achsdaten in Form eines zusätzlichen Knotens (LC_Drive) verfügbar und über die Funktionsbaustein-Instanz [L_LCB_AXIS_REF](#) (67) abrufbar.

3. Verbinden Sie einen Funktionsbaustein mit der Achsdaten-Instanz **LC_Drive** aus der Steuerungskonfiguration, um eine Anbindung an den Inverter herzustellen.

Übertragene Prozessdaten werden automatisch in die Achsdaten geschrieben. Dadurch ist keine manuelle Zuordnung der Prozessdaten in die Applikation erforderlich.

► [Funktionsbaustein an einen Inverter anbinden](#) (65)



Kommunikationshandbuch "Controller-based Automation EtherCAT"

Beachten Sie die weiteren Informationen zur Inbetriebnahme:

- "Feldbus-Scan" / Import von Gerätebeschreibungsdateien
- Synchronisation mit "Distributed Clocks"
- Einstellungen von Zykluszeiten und Motion-Parametern
- EtherCAT I/O-Mapping
- Modulare Maschinenkonfiguration (ab Release 3.10)
- Funktionsbibliothek **L_IODrvEtherCAT**
- Mischbetrieb EtherCAT mit anderen Bussystemen
- EtherCAT-Diagnose

6.7.2

CAN-Feldgeräte anhängen

**Hinweis!**

Die Konfiguration eines Lenze Controllers im CANopen-Netzwerk muss im »PLC Designer« angelegt werden, da beim Start eines Controllers die komplette Konfiguration in die angeschlossenen Slaves geschrieben wird. Zuvor vorgenommene Einstellungen in den Slaves werden dabei überschrieben.

**Tipp!****Fehlende Geräte oder Geräte anderer Hersteller**

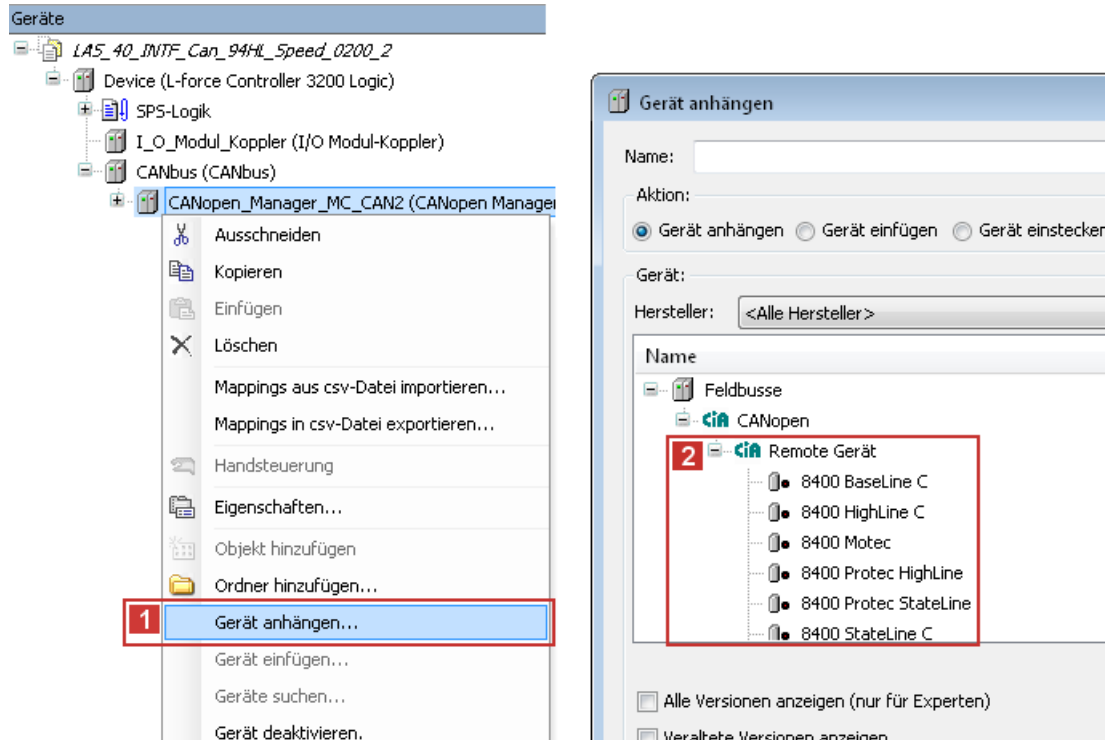
Um fehlende Geräte oder Geräte anderer Hersteller einzubinden, sind die entsprechenden Gerätebeschreibungsdateien des Herstellers erforderlich. Im »PLC Designer« können Gerätebeschreibungsdateien vom Typ *.XML, *.devdesc.XML, *.EDS, *.DCF und *.GSx über den Menübefehl **Tools → Geräte-Repository...** importiert werden.

Zudem steht die allgemeine Gerätebeschreibung "[Lenze Generic Drive](#)" (📖 63) zur Einbindung von anderen Lenze-Geräten zur Verfügung. Die Ansteuerung erfolgt über den Funktionsbaustein [L_LCB_GenericDrive](#) (📖 72).



So hängen Sie CAN-Feldgeräte an:

1. Im Kontextmenü des "CANopen Manager" mit dem Befehl **1 Gerät anhängen** ein Logic-Gerät (Slave) unter dem "CANopen Manager" (Master) einfügen.



Wählen Sie aus der **2** Auswahlliste ein Feldgerät. Es sind nur Geräte auswählbar, deren CAN-Gerätebeschreibungsdateien im »PLC Designer« importiert wurden.

Bei EDS-Dateien, die im »Engineer« erzeugt wurden, erscheint das Feldgerät in der Auswahlliste mit dem Namen, das es beim Export der EDS-Datei im »Engineer« hatte, ergänzt um den Namen der Schnittstelle und den Gerätetyp.

2. Wiederholen Sie den Befehl **1 Gerät anhängen** solange, bis alle am Feldbus teilnehmenden Slaves in der Steuerungskonfiguration eingebunden sind.

Alternativ können Sie einen bereits eingefügten Teilnehmer über das Kontextmenü des Gerätes **kopieren** und **einfügen**.

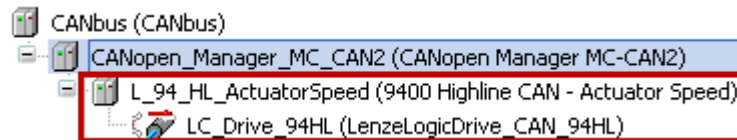
3. Den eingefügten Slaves sinnvolle Bezeichnungen geben (z. B. "Drive_vertical").

Die Bezeichnungen dürfen ...

- nur die Zeichen "A ... Z", "a ... z", "0 ... 9" oder "_" enthalten;
- nicht mit einer Ziffer beginnen.

Durch einen Mausklick auf das Element wird die Bezeichnung zur Eingabe freigegeben.

Beispiel:



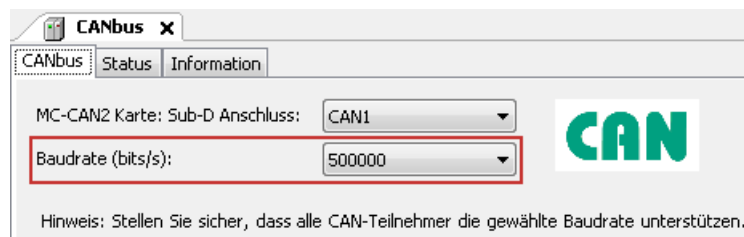
Unterhalb des Inverters sind die Achsdaten in Form eines zusätzlichen Knotens (LC_Drive) verfügbar und über die Funktionsbaustein-Instanz [L_LCB_AXIS_REF](#) ([67](#)) abrufbar.

4. Verbinden Sie einen Funktionsbaustein mit der Achsdaten-Instanz **LC_Drive** aus der Steuerungskonfiguration, um eine Anbindung an den Inverter herzustellen.

Übertragene Prozessdaten werden automatisch in die Achsdaten geschrieben. Dadurch ist keine manuelle Zuordnung der Prozessdaten in die Applikation erforderlich.

► [Funktionsbaustein an einen Inverter anbinden](#) ([65](#))

5. Über die Registerkarte **CANbus** die Übertragungsrate einstellen.



Hinweis!

Die im »PLC Designer« eingestellte Übertragungsrate überschreibt die für die Feldgeräte über »Engineer« oder »EASY Starter« eingestellte Übertragungsrate.

Stellen Sie in einem CANopen-Netzwerk generell für alle Teilnehmer dieselbe Übertragungsrate ein.

6. Über die Registerkarte **CANopen_Manager** die Parameter zur Sync-Erzeugung einstellen.

Die Sync-Erzeugung ist notwendig, wenn ...

- auf dem Bus mindestens ein PDO mit Sync-gesteuerter Verarbeitung benutzt wird;
- in mehreren Feldgeräten die Applikationen Takt-synchron laufen sollen;
- am Feldbus Motion-Geräte betrieben werden sollen.

Falls Sie die CAN-Synchronisation einsetzen wollen, setzen Sie ein Häkchen im Eingabefeld

1 Sync-Erzeugung aktivieren.

Stellen Sie im Eingabefeld die **2 Sync-Zykluszeit** ein.



Kommunikationshandbuch "Controller-based Automation CANopen"

Beachten Sie die weiteren Informationen zur Inbetriebnahme:

- Import von Gerätebeschreibungsdateien
- Einstellungen von CAN-Parametern und PDO-Mapping
- Besonderheiten bei Inverter Drives 8400, Servo Drives 9400 und I/O-System 1000
- Mischbetrieb CANopen mit anderen Bussystemen
- CAN-Diagnose

7 Gerätebeschreibungen

Dieses Kapitel enthält Informationen zu den im »PLC Designer« verfügbaren Gerätebeschreibungen der Inverter Drives 8400 und Servo Drives 9400.

Wird ein Inverter in die Steuerungskonfiguration eingefügt, sind unterhalb des Gerätes die Achsdaten in Form eines zusätzlichen Knotens (LC_Drive) verfügbar und über die Funktionsbaustein-Instanz [L_LCB_AXIS_REF](#) (67) abrufbar.

Informationen zur Erweiterung der [Steuerungskonfiguration \(EtherCAT/CAN\)](#) (26) im »PLC Designer« finden Sie hier:

► [Beispielprojekte erweitern \(Geräte hinzufügen\)](#) (43)

7.1 Inverter Drive 8400 BaseLine C



Die Kommunikation mit dem Inverter Drive 8400 BaseLine C ist nur über den CAN-Bus (CANopen) möglich.

Verfügbare Technologieapplikationen

- Keine

Beispielprojekt

- LAS_40_INTF_Can_84BL_Speed_0100

Übersicht der Indizes

Die empfangenen Prozessdaten werden in das Element *awReceiveData* der [L_LCB_AXIS_REF](#)-Instanz (67) übertragen.

Die zu sendenden Prozessdaten werden vom Element *awTransmitData* der [L_LCB_AXIS_REF](#)-Instanz (67) übertragen.

Die Tabelle zeigt die Abbildung der Indizes auf Variablen.

Applikationsvariable	Name CAN	Index CAN	Name EtherCAT	Index EtherCAT
awTransmitData1	LP_CanIn1_wCtrl	0x5E66/01	-	-
awTransmitData2	LP_CanIn1_wIn2	0x5E66/02	-	-
awTransmitData3	LP_CanIn1_wIn3	0x5E66/03	-	-
awTransmitData4	LP_CanIn1_wIn4	0x5E66/04	-	-
awTransmitData5	LP_CanIn2_wIn1	0x5E66/05	-	-
awTransmitData6	LP_CanIn2_wIn2	0x5E66/06	-	-
awTransmitData7	LP_CanIn2_wIn3	0x5E66/07	-	-
awTransmitData8	LP_CanIn2_wIn4	0x5E66/08	-	-
awTransmitData9	-	-	-	-
awTransmitData10	-	-	-	-
awTransmitData11	-	-	-	-
awTransmitData12	-	-	-	-
awTransmitData13	-	-	-	-
awTransmitData14	-	-	-	-
awTransmitData15	-	-	-	-

Applikationsvariable	Name CAN	Index CAN	Name EtherCAT	Index EtherCAT
awTransmitData16	-	-	-	-
awReceiveData1	LP_CanOut1 _wState	0x5C9B/01	-	-
awReceiveData2	LP_CanOut1 _wOut2	0x5C9B/02	-	-
awReceiveData3	LP_CanOut1 _wOut3	0x5C9B/03	-	-
awReceiveData4	LP_CanOut1 _wOut4	0x5C9B/04	-	-
awReceiveData5	LP_CanOut2 _wOut1	0x5C9B/05	-	-
awReceiveData6	LP_CanOut2 _wOut2	0x5C9B/06	-	-
awReceiveData7	LP_CanOut2 _wOut3	0x5C9B/07	-	-
awReceiveData8	LP_CanOut2 _wOut4	0x5C9B/08	-	-
awReceiveData9	-	-	-	-
awReceiveData10	-	-	-	-
awReceiveData11	-	-	-	-
awReceiveData12	-	-	-	-
awReceiveData13	-	-	-	-
awReceiveData14	-	-	-	-
awReceiveData15	-	-	-	-
awReceiveData16	-	-	-	-

7.2

Inverter Drive 8400 StateLine C



Die Kommunikation mit dem Inverter Drive 8400 StateLine C ist über den CAN-Bus (CANopen) und EtherCAT möglich.

Verfügbare Technologieapplikationen

- Stellantrieb Drehzahl
- Abschaltpositionierung ohne Rückführung

Beispielprojekte

- LAS_40_INTF_Can_84SL_Speed_0200
- LAS_40_INTF_ETC_84SL_Speed_0100

Übersicht der Indizes

Die empfangenen Prozessdaten werden in das Element *awReceiveData* der [L_LCB_AXIS_REF](#)-Instanz (67) übertragen.

Die zu sendenden Prozessdaten werden vom Element *awTransmitData* der [L_LCB_AXIS_REF](#)-Instanz (67) übertragen.

Die Tabelle zeigt die Abbildung der Indizes auf Variablen.

Applikationsvariable	Name CAN	Index CAN	Name EtherCAT	Index EtherCAT
awTransmitData1	LP_CanIn1_wCtrl	0x5E66/01	MciIn_wCtrl	0x5C93/01
awTransmitData2	LP_CanIn1_wIn2	0x5E66/02	MciIn_wIn2	0x5C93/02
awTransmitData3	LP_CanIn1_wIn3	0x5E66/03	MciIn_wIn3	0x5C93/03
awTransmitData4	LP_CanIn1_wIn4	0x5E66/04	MciIn_wIn4	0x5C93/04
awTransmitData5	LP_CanIn2_wIn1	0x5E66/05	MciIn_wIn5	0x5C93/05
awTransmitData6	LP_CanIn2_wIn2	0x5E66/06	MciIn_wIn6	0x5C93/06
awTransmitData7	LP_CanIn2_wIn3	0x5E66/07	MciIn_wIn7	0x5C93/07
awTransmitData8	LP_CanIn2_wIn4	0x5E66/08	MciIn_wIn8	0x5C93/08
awTransmitData9	LP_CanIn3_wIn1	0x5E66/09	MciIn_wIn9	0x5C93/09
awTransmitData10	LP_CanIn3_wIn2	0x5E66/0A	MciIn_wIn10	0x5C93/0A
awTransmitData11	LP_CanIn3_wIn3	0x5E66/0B	MciIn_wIn11	0x5C93/0B
awTransmitData12	LP_CanIn3_wIn4	0x5E66/0C	MciIn_wIn12	0x5C93/0C
awTransmitData13	-	-	MciIn_wIn13	0x5C93/0D
awTransmitData14	-	-	MciIn_wIn14	0x5C93/0E
awTransmitData15	-	-	MciIn_wIn15	0x5C93/0F
awTransmitData16	-	-	MciIn_wIn16	0x5C93/10
awReceiveData1	LP_CanOut1_wState	0x5C9B/01	MciOut_wState	0x5C92/01
awReceiveData2	LP_CanOut1_wOut2	0x5C9B/02	MciOut_wOut2	0x5C92/02
awReceiveData3	LP_CanOut1_wOut3	0x5C9B/03	MciOut_wOut3	0x5C92/03
awReceiveData4	LP_CanOut1_wOut4	0x5C9B/04	MciOut_wOut4	0x5C92/04
awReceiveData5	LP_CanOut2_wOut1	0x5C9B/05	MciOut_wOut5	0x5C92/05

Applikationsvariable	Name CAN	Index CAN	Name EtherCAT	Index EtherCAT
awReceiveData6	LP_CanOut2_wOut2	0x5C9B/06	MciOut_wOut6	0x5C92/06
awReceiveData7	LP_CanOut2_wOut3	0x5C9B/07	MciOut_wOut7	0x5C92/07
awReceiveData8	LP_CanOut2_wOut4	0x5C9B/08	MciOut_wOut8	0x5C92/08
awReceiveData9	LP_CanOut3_wOut1	0x5C9B/09	MciOut_wOut9	0x5C92/09
awReceiveData10	LP_CanOut3_wOut2	0x5C9B/0A	MciOut_wOut10	0x5C92/0A
awReceiveData11	LP_CanOut3_wOut3	0x5C9B/0B	MciOut_wOut11	0x5C92/0B
awReceiveData12	LP_CanOut3_wOut4	0x5C9B/0C	MciOut_wOut12	0x5C92/0C
awReceiveData13	-	-	MciOut_wOut13	0x5C92/0D
awReceiveData14	-	-	MciOut_wOut14	0x5C92/0E
awReceiveData15	-	-	MciOut_wOut15	0x5C92/0F
awReceiveData16	-	-	MciOut_wOut16	0x5C92/10

7.3

Inverter Drive 8400 HighLine C



Die Kommunikation mit dem Inverter Drive 8400 HighLine C ist über den CAN-Bus (CANopen) und EtherCAT möglich.

Verfügbare Technologieapplikationen

- Stellantrieb Drehzahl
- Abschaltpositionierung ohne Rückführung
- Tabellenpositionierung ohne Rückführung

Beispielprojekte

- LAS_40_INTF_Can_84HL_TabPos_0200
- LAS_40_INTF_ETC_84HL_TabPos_0100

Übersicht der Indizes

Die empfangenen Prozessdaten werden in das Element *awReceiveData* der [L_LCB_AXIS_REF](#)-Instanz (67) übertragen.

Die zu sendenden Prozessdaten werden vom Element *awTransmitData* der [L_LCB_AXIS_REF](#)-Instanz (67) übertragen.

Die Tabelle zeigt die Abbildung der Indizes auf Variablen.

Applikationsvariable	Name CAN	Index CAN	Name EtherCAT	Index EtherCAT
awTransmitData1	LP_CanIn1_wCtrl	0x5E66/01	MciIn_wCtrl	0x5C93/01
awTransmitData2	LP_CanIn1_wIn2	0x5E66/02	MciIn_wIn2	0x5C93/02
awTransmitData3	LP_CanIn1_wIn3	0x5E66/03	MciIn_wIn3	0x5C93/03
awTransmitData4	LP_CanIn1_wIn4	0x5E66/04	MciIn_wIn4	0x5C93/04
awTransmitData5	LP_CanIn2_wIn1	0x5E66/05	MciIn_wIn5	0x5C93/05
awTransmitData6	LP_CanIn2_wIn2	0x5E66/06	MciIn_wIn6	0x5C93/06
awTransmitData7	LP_CanIn2_wIn3	0x5E66/07	MciIn_wIn7	0x5C93/07
awTransmitData8	LP_CanIn2_wIn4	0x5E66/08	MciIn_wIn8	0x5C93/08
awTransmitData9	LP_CanIn3_wIn1	0x5E66/09	MciIn_wIn9	0x5C93/09
awTransmitData10	LP_CanIn3_wIn2	0x5E66/0A	MciIn_wIn10	0x5C93/0A
awTransmitData11	LP_CanIn3_wIn3	0x5E66/0B	MciIn_wIn11	0x5C93/0B
awTransmitData12	LP_CanIn3_wIn4	0x5E66/0C	MciIn_wIn12	0x5C93/0C
awTransmitData13	-	-	MciIn_wIn13	0x5C93/0D
awTransmitData14	-	-	MciIn_wIn14	0x5C93/0E
awTransmitData15	-	-	MciIn_wIn15	0x5C93/0F
awTransmitData16	-	-	MciIn_wIn16	0x5C93/10
awReceiveData1	LP_CanOut1_wState	0x5C9B/01	MciOut_wState	0x5C92/01
awReceiveData2	LP_CanOut1_wOut2	0x5C9B/02	MciOut_wOut2	0x5C92/02
awReceiveData3	LP_CanOut1_wOut3	0x5C9B/03	MciOut_wOut3	0x5C92/03
awReceiveData4	LP_CanOut1_wOut4	0x5C9B/04	MciOut_wOut4	0x5C92/04

Applikationsvariable	Name CAN	Index CAN	Name EtherCAT	Index EtherCAT
awReceiveData5	LP_CanOut2_wOut1	0x5C9B/05	MciOut_wOut5	0x5C92/05
awReceiveData6	LP_CanOut2_wOut2	0x5C9B/06	MciOut_wOut6	0x5C92/06
awReceiveData7	LP_CanOut2_wOut3	0x5C9B/07	MciOut_wOut7	0x5C92/07
awReceiveData8	LP_CanOut2_wOut4	0x5C9B/08	MciOut_wOut8	0x5C92/08
awReceiveData9	LP_CanOut3_wOut1	0x5C9B/09	MciOut_wOut9	0x5C92/09
awReceiveData10	P_CanOut3_wOut2	0x5C9B/0A	MciOut_wOut10	0x5C92/0A
awReceiveData11	P_CanOut3_wOut3	0x5C9B/0B	MciOut_wOut11	0x5C92/0B
awReceiveData12	P_CanOut3_wOut4	0x5C9B/0C	MciOut_wOut12	0x5C92/0C
awReceiveData13	-	-	MciOut_wOut13	0x5C92/0D
awReceiveData14	-	-	MciOut_wOut14	0x5C92/0E
awReceiveData15	-	-	MciOut_wOut15	0x5C92/0F
awReceiveData16	-	-	MciOut_wOut16	0x5C92/10

7.4

Inverter Drive 8400 TopLine C



Die Kommunikation mit dem Inverter Drive 8400 TopLine C ist über den CAN-Bus (CANopen) und EtherCAT möglich.

Verfügbare Technologieapplikationen

- Stellantrieb Drehzahl
- Abschaltpositionierung ohne Rückführung
- Tabellenpositionierung ohne Rückführung (mit Ablaufpositionierung)

Erweiterte Funktionen

Funktionsblöcke für ...

- Positionier-Ablaufsteuerung
- Elektrische Welle (Geschwindigkeits- und Winkelgleichlauf)
- Tänzerregelung
- Netzausfallregelung

Beispielprojekte

- Keine

Übersicht der Indizes

Die empfangenen Prozessdaten werden in das Element *awReceiveData* der [L LCB AXIS REF-](#)Instanz (67) übertragen.

Die zu sendenden Prozessdaten werden vom Element *awTransmitData* der [L LCB AXIS REF-](#)Instanz (67) übertragen.

Die Tabelle zeigt die Abbildung der Indizes auf Variablen.

Applikationsvariable	Name CAN	Index CAN	Name EtherCAT	Index EtherCAT
awTransmitData1	LP_CanIn1_wCtrl	0x5E66/01	MciIn_wCtrl	0x5C93/01
awTransmitData2	LP_CanIn1_wIn2	0x5E66/02	MciIn_wIn2	0x5C93/02
awTransmitData3	LP_CanIn1_wIn3	0x5E66/03	MciIn_wIn3	0x5C93/03
awTransmitData4	LP_CanIn1_wIn4	0x5E66/04	MciIn_wIn4	0x5C93/04
awTransmitData5	LP_CanIn2_wIn1	0x5E66/05	MciIn_wIn5	0x5C93/05
awTransmitData6	LP_CanIn2_wIn2	0x5E66/06	MciIn_wIn6	0x5C93/06
awTransmitData7	LP_CanIn2_wIn3	0x5E66/07	MciIn_wIn7	0x5C93/07
awTransmitData8	LP_CanIn2_wIn4	0x5E66/08	MciIn_wIn8	0x5C93/08
awTransmitData9	LP_CanIn3_wIn1	0x5E66/09	MciIn_wIn9	0x5C93/09
awTransmitData10	LP_CanIn3_wIn2	0x5E66/0A	MciIn_wIn10	0x5C93/0A
awTransmitData11	LP_CanIn3_wIn3	0x5E66/0B	MciIn_wIn11	0x5C93/0B
awTransmitData12	LP_CanIn3_wIn4	0x5E66/0C	MciIn_wIn12	0x5C93/0C
awTransmitData13	-	-	MciIn_wIn13	0x5C93/0D
awTransmitData14	-	-	MciIn_wIn14	0x5C93/0E
awTransmitData15	-	-	MciIn_wIn15	0x5C93/0F
awTransmitData16	-	-	MciIn_wIn16	0x5C93/10

Applikationsvariable	Name CAN	Index CAN	Name EtherCAT	Index EtherCAT
awReceiveData1	LP_CanOut1_wState	0x5C9B/01	MciOut_wState	0x5C92/01
awReceiveData2	LP_CanOut1_wOut2	0x5C9B/02	MciOut_wOut2	0x5C92/02
awReceiveData3	LP_CanOut1_wOut3	0x5C9B/03	MciOut_wOut3	0x5C92/03
awReceiveData4	LP_CanOut1_wOut4	0x5C9B/04	MciOut_wOut4	0x5C92/04
awReceiveData5	LP_CanOut2_wOut1	0x5C9B/05	MciOut_wOut5	0x5C92/05
awReceiveData6	LP_CanOut2_wOut2	0x5C9B/06	MciOut_wOut6	0x5C92/06
awReceiveData7	LP_CanOut2_wOut3	0x5C9B/07	MciOut_wOut7	0x5C92/07
awReceiveData8	LP_CanOut2_wOut4	0x5C9B/08	MciOut_wOut8	0x5C92/08
awReceiveData9	LP_CanOut3_wOut1	0x5C9B/09	MciOut_wOut9	0x5C92/09
awReceiveData10	P_CanOut3_wOut2	0x5C9B/0A	MciOut_wOut10	0x5C92/0A
awReceiveData11	P_CanOut3_wOut3	0x5C9B/0B	MciOut_wOut11	0x5C92/0B
awReceiveData12	P_CanOut3_wOut4	0x5C9B/0C	MciOut_wOut12	0x5C92/0C
awReceiveData13	-	-	MciOut_wOut13	0x5C92/0D
awReceiveData14	-	-	MciOut_wOut14	0x5C92/0E
awReceiveData15	-	-	MciOut_wOut15	0x5C92/0F
awReceiveData16	-	-	MciOut_wOut16	0x5C92/10

7.5

Servo Drive 9400 HighLine



Die Kommunikation mit dem Servo Drive 9400 HighLine ist über den CAN-Bus (CANopen) und EtherCAT möglich.

Verfügbare Technologieapplikationen

Ab Firmware-Version 1.0.1:

- Leere Applikation

Hinweis!

Nach der Erstellung der »Engineer«-Applikation müssen die Indizes manuell im »PLC Designer« konfiguriert werden, da hier keine vordefinierten Port-Variablen (Indizes) definiert sind.

Ab Firmware-Version 4.0:

- Stellantrieb - Drehzahl
- Stellantrieb - Drehmoment

Ab Firmware-Version 5.0:

- Tabellenpositionierung
- Elektronisches Getriebe
- Gleichlauf mit Markensynchronisation
- Positionier-Ablaufsteuerung

Ab Firmware-Version 8.0:

- CiA 402 (für SoftMotion-Anwendungen)

Erweiterte Funktionen

- Funktionsblöcke für Kurvenscheiben- Funktion

Beispielprojekte

- LAS_40_INTF_Can_94HL_Speed_0200
- LAS_40_INTF_Can_94HL_TabPos_0200
- LAS_40_INTF_ETC_94HL_Speed_0100
- LAS_40_INTF_ETC_94HL_TabPos_0100

7.5.1 Indizes bei Technologieapplikation "Tabellenpositionierung"

Die empfangenen Prozessdaten werden in das Element *awReceiveData* der [L_LCB_AXIS_REF](#)-Instanz (67) übertragen.

Die zu sendenden Prozessdaten werden vom Element *awTransmitData* der [L_LCB_AXIS_REF](#)-Instanz (67) übertragen.

Die Tabelle zeigt die Abbildung der Indizes auf Variablen.

Applikationsvariable	Name CAN	Index CAN	Name EtherCAT	Index EtherCAT
awTransmitData1	LPortControl1	0xA580/02	LPortAxisIn1_wControl	0xA580/01
awTransmitData2	LPort16In1	0xA540/02	LPortAxisIn1_nIn1	0xA540/01
awTransmitData3	LPort16In2	0xA540/03	LPortAxisIn1_dnIn2	0xA640/01
awTransmitData4	LPort16In3	0xA540/04		
awTransmitData5	LPort32In1	0xA640/02	LPortControl1	0xA580/02
awTransmitData6			LPortControl2	0xA580/03
awTransmitData7	LPort32In2	0xA640/03	LPort32In1	0xA640/02
awTransmitData8				
awTransmitData9	LPort32In3	0xA640/04	LPort32In2	0xA640/03
awTransmitData10				
awTransmitData11	LPortControl2	0xA580/03	LPort32In3	0xA640/04
awTransmitData12	LPortAxisIn1_wControl	0xA580/01		
awTransmitData13	LPortAxisIn1_nIn1	0xA540/01	LPort16In1	0xA540/02
awTransmitData14	LPortAxisIn1_dnIn2	0xA640/01	LPort16In2	0xA540/03
awTransmitData15			LPort16In3	0xA540/04
awTransmitData16	-	-	-	-
awReceiveData1	LPortAxisOut1_wStatus	0xA100/01	LPortAxisOut1_wStatus	0xA100/01
awReceiveData2	LPortAxisOut1_nOut1	0xA0C0/01	LPortAxisOut1_nOut1	0xA0C0/01
awReceiveData3	LPortAxisOut1_dnOut2	0xA1C0/01	LPortAxisOut1_dnOut2	0xA1C0/01
awReceiveData4				
awReceiveData5	LPortStatus1	0xA100/02	LPortStatus1	0xA100/02
awReceiveData6	LPortStatus2	0xA100/03	LPortStatus2	0xA100/03
awReceiveData7	LPort32Out1	0xA1C0/02	LPort32Out1	0xA1C0/02
awReceiveData8				
awReceiveData9	LPort32Out2	0xA1C0/03	LPort32Out2	0xA1C0/03
awReceiveData10				
awReceiveData11	LPort32Out3	0xA1C0/04	LPort32Out3	0xA1C0/04
awReceiveData12				
awReceiveData13	LPort16Out1	0xA0C0/02	LPort16Out1	0xA0C0/02
awReceiveData14	LPort16Out2	0xA0C0/03	LPort16Out2	0xA0C0/03
awReceiveData15	LPort16Out3	0xA0C0/04	LPort16Out3	0xA0C0/04
awReceiveData16	-	-	-	-

7.5.2 Indizes bei den weiteren Technologieapplikationen

Die empfangenen Prozessdaten werden in das Element *awReceiveData* der [L_LCB_AXIS_REF](#)-Instanz (67) übertragen.

Die zu sendenden Prozessdaten werden vom Element *awTransmitData* der [L_LCB_AXIS_REF](#)-Instanz (67) übertragen.

Die Tabelle zeigt die Abbildung der Indizes auf Variablen.

Applikationsvariable	Name CAN	Index CAN	Name EtherCAT	Index EtherCAT
awTransmitData1	LPortAxisIn1_wControl	0xA580/01	LPortAxisIn1_wControl	0xA580/01
awTransmitData2	LPortAxisIn1_nIn1	0xA540/01	LPortAxisIn1_nIn1	0xA540/01
awTransmitData3	LPortAxisIn1_dnIn2	0xA640/01	LPortAxisIn1_dnIn2	0xA640/01
awTransmitData4				
awTransmitData5	LPortControl1	0xA580/02	LPortControl1	0xA580/02
awTransmitData6	LPortControl2	0xA580/03	LPortControl2	0xA580/03
awTransmitData7	LPort32In1	0xA640/02	LPort32In1	0xA640/02
awTransmitData8				
awTransmitData9	LPort32In2	0xA640/03	LPort32In2	0xA640/03
awTransmitData10				
awTransmitData11	LPort32In3	0xA640/04	LPort32In3	0xA640/04
awTransmitData12				
awTransmitData13	LPort16In1	0xA540/02	LPort16In1	0xA540/02
awTransmitData14	LPort16In2	0xA540/03	LPort16In2	0xA540/03
awTransmitData15	LPort16In3	0xA540/04	LPort16In3	0xA540/04
awTransmitData16	-	-	-	-
awReceiveData1	LPortAxisOut1_wStatus	0xA100/01	LPortAxisOut1_wStatus	0xA100/01
awReceiveData2	LPortAxisOut1_nOut1	0xA0C0/01	LPortAxisOut1_nOut1	0xA0C0/01
awReceiveData3	LPortAxisOut1_dnOut2	0xA1C0/01	LPortAxisOut1_dnOut2	0xA1C0/01
awReceiveData4				
awReceiveData5	LPortStatus1	0xA100/02	LPortStatus1	0xA100/02
awReceiveData6	LPortStatus2	0xA100/03	LPortStatus2	0xA100/03
awReceiveData7	LPort32Out1	0xA1C0/02	LPort32Out1	0xA1C0/02
awReceiveData8				
awReceiveData9	LPort32Out2	0xA1C0/03	LPort32Out2	0xA1C0/03
awReceiveData10				
awReceiveData11	LPort32Out3	0xA1C0/04	LPort32Out3	0xA1C0/04
awReceiveData12				
awReceiveData13	LPort16Out1	0xA0C0/02	LPort16Out1	0xA0C0/02
awReceiveData14	LPort16Out2	0xA0C0/03	LPort16Out2	0xA0C0/03
awReceiveData15	LPort16Out3	0xA0C0/04	LPort16Out3	0xA0C0/04
awReceiveData16	-	-	-	-

7.6

Servo Drive 9400 Regenerative Power Supply Module



Die Kommunikation mit dem Servo Drive 9400 Regenerative Power Supply Module ist über den CAN-Bus (CANopen) und EtherCAT möglich.

Verfügbare Technologieapplikationen

- Applikation auf Basis der Technologieapplikation VR-Modul V3.0

Beispielprojekte

- Keine

Übersicht der Indizes

Die empfangenen Prozessdaten werden in das Element *awReceiveData* der [L_LCB_AXIS_REF](#)-Instanz (67) übertragen.

Die zu sendenden Prozessdaten werden vom Element *awTransmitData* der [L_LCB_AXIS_REF](#)-Instanz (67) übertragen.

Die Tabelle zeigt die Abbildung der Indizes auf Variablen.

Applikationsvariable	Name CAN	Index CAN	Name EtherCAT	Index EtherCAT
awTransmitData1	LPortAxisIn1_wControl	0xA580/01	LPortAxisIn1_wControl	0xA580/01
awTransmitData2	LPortAxisIn1_nIn1	0xA540/01	LPortAxisIn1_nIn1	0xA540/01
awTransmitData3	LPortAxisIn1_dnIn2	0xA640/01	LPortAxisIn1_dnIn2	0xA640/01
awTransmitData4				
awTransmitData5	LPortControl1	0xA580/02	LPortControl1	0xA580/02
awTransmitData6	LPortControl2	0xA580/03	LPortControl2	0xA580/03
awTransmitData7	LPort32In1	0xA640/02	LPort32In1	0xA640/02
awTransmitData8				
awTransmitData9	LPort32In2	0xA640/03	LPort32In2	0xA640/03
awTransmitData10				
awTransmitData11	LPort32In3	0xA640/04	LPort32In3	0xA640/04
awTransmitData12				
awTransmitData13	LPort16In1	0xA540/02	LPort16In1	0xA540/02
awTransmitData14	LPort16In2	0xA540/03	LPort16In2	0xA540/03
awTransmitData15	LPort16In3	0xA540/04	LPort16In3	0xA540/04
awTransmitData16	-	-	-	-
awReceiveData1	LPortAxisOut1_wStatus	0xA100/01	LPortAxisOut1_wStatus	0xA100/01
awReceiveData2	LPortAxisOut1_nOut1	0xA0C0/01	LPortAxisOut1_nOut1	0xA0C0/01
awReceiveData3	LPortAxisOut1_dnOut2	0xA1C0/01	LPortAxisOut1_dnOut2	0xA1C0/01
awReceiveData4				
awReceiveData5	LPortStatus1	0xA100/02	LPortStatus1	0xA100/02
awReceiveData6	LPortStatus2	0xA100/03	LPortStatus2	0xA100/03
awReceiveData7	LPort32Out1	0xA1C0/02	LPort32Out1	0xA1C0/02
awReceiveData8				

Applikationsvariable	Name CAN	Index CAN	Name EtherCAT	Index EtherCAT
awReceiveData9	LPort32Out2	0xA1C0/03	LPort32Out2	0xA1C0/03
awReceiveData10				
awReceiveData11	LPort32Out3	0xA1C0/04	LPort32Out3	0xA1C0/04
awReceiveData12				
awReceiveData13	LPort16Out1	0xA0C0/02	LPort16Out1	0xA0C0/02
awReceiveData14	LPort16Out2	0xA0C0/03	LPort16Out2	0xA0C0/03
awReceiveData15	LPort16Out3	0xA0C0/04	LPort16Out3	0xA0C0/04
awReceiveData16	-	-	-	-

7.7

Lenze Generic Drive



Das "Lenze Generic Drive" dient zur Ansteuerung eines beliebigen Lenze-Inverters an den CAN-Bus (CANopen).

Die Ansteuerung erfolgt über den Funktionsbaustein [L LCB GenericDrive](#) (72).

Beispielprojekte

- Keine

Übersicht der Indizes



Hinweis!

Das Mapping der Prozessdaten muss im Inverter erfolgen.

Über die Gerätebeschreibung des "Lenze Generic Drive" werden Platzhalter (z. B. "CAN IN1 W1") definiert, in welche die Prozessdaten übertragen werden.

Die empfangenen Prozessdaten werden in das Element *awReceiveData* der [L LCB AXIS REF](#)-Instanz (67) übertragen.

Die zu sendenden Prozessdaten werden vom Element *awTransmitData* der [L LCB AXIS REF](#)-Instanz (67) übertragen.

Die Tabelle zeigt die Abbildung der Indizes auf Variablen.

Applikationsvariable	Name CAN	Index CAN	Name EtherCAT	Index EtherCAT
awTransmitData1	CAN IN1 W1	0x5CA7/01	-	-
awTransmitData2	CAN IN1 W2	0x5CA7/02	-	-
awTransmitData3	CAN IN1 W3	0x5CA7/03	-	-
awTransmitData4	CAN IN1 W4	0x5CA7/04	-	-
awTransmitData5	CAN IN2 W1	0x5CA8/01	-	-
awTransmitData6	CAN IN2 W2	0x5CA8/02	-	-
awTransmitData7	CAN IN2 W3	0x5CA8/03	-	-
awTransmitData8	CAN IN2 W4	0x5CA8/04	-	-
awTransmitData9	CAN IN3 W1	0x5CB0/01	-	-
awTransmitData10	CAN IN3 W2	0x5CB0/02	-	-
awTransmitData11	CAN IN3 W3	0x5CB0/03	-	-
awTransmitData12	CAN IN3 W4	0x5CB0/04	-	-
awTransmitData13	-	-	-	-
awTransmitData14	-	-	-	-
awTransmitData15	-	-	-	-
awTransmitData16	-	-	-	-
awReceiveData1	CAN OUT1 W1	0x5CAD/01	-	-
awReceiveData2	CAN OUT1 W2	0x5CAD/02	-	-
awReceiveData3	CAN OUT1 W3	0x5CAD/03	-	-
awReceiveData4	CAN OUT1 W4	0x5CAD/04	-	-
awReceiveData5	CAN OUT2 W1	0x5CAE/01	-	-

Applikationsvariable	Name CAN	Index CAN	Name EtherCAT	Index EtherCAT
awReceiveData6	CAN OUT2 W2	0x5CAE/02	-	-
awReceiveData7	CAN OUT2 W3	0x5CAE/03	-	-
awReceiveData8	CAN OUT2 W4	0x5CAE/04	-	-
awReceiveData9	CAN OUT3 W1	0x5CAF/01	-	-
awReceiveData10	CAN OUT3 W2	0x5CAF/02	-	-
awReceiveData11	CAN OUT3 W3	0x5CAF/30	-	-
awReceiveData12	CAN OUT3 W4	0x5CAF/04	-	-
awReceiveData13	-	-	-	-
awReceiveData14	-	-	-	-
awReceiveData15	-	-	-	-
awReceiveData16	-	-	-	-

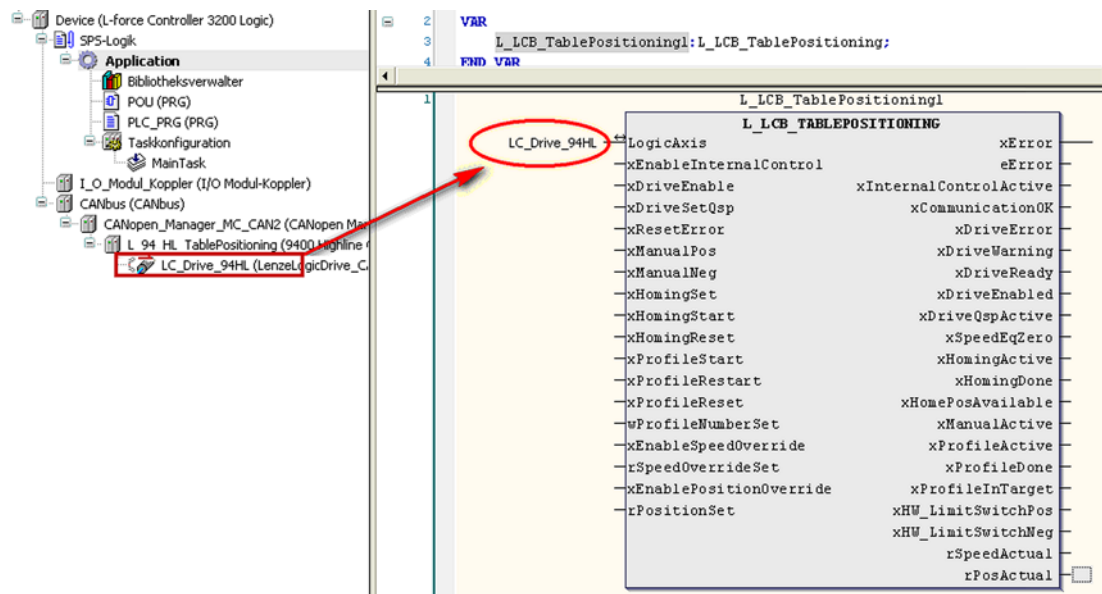
8 Funktionsbibliothek L_LCB_LogicControlBasic

Die Bibliothek **L_LCB_LogicControlBasic** enthält Funktionsbausteine zur geräteunabhängigen Ansteuerung von Invertern am Logic-Bus:

Funktionsbaustein	Beschreibung
▶ L_LCB_ActuatorSpeed	Ansteuerung eines Inverters mit dem Applikationsbeispiel "Stellantrieb - Drehzahl"
▶ L_LCB_TablePositioning	Ansteuerung eines Inverters mit dem Applikationsbeispiel "Tabellenpositionierung"
▶ L_LCB_8400Drive	Ansteuerung eines Inverter Drive 8400
▶ L_LCB_9400Drive	Ansteuerung eines Servo Drive 9400
▶ L_LCB_GenericDrive	Ansteuerung eines beliebigen Lenze Inverters mit einer beliebigen Applikation

Funktionsbaustein an einen Inverter anbinden

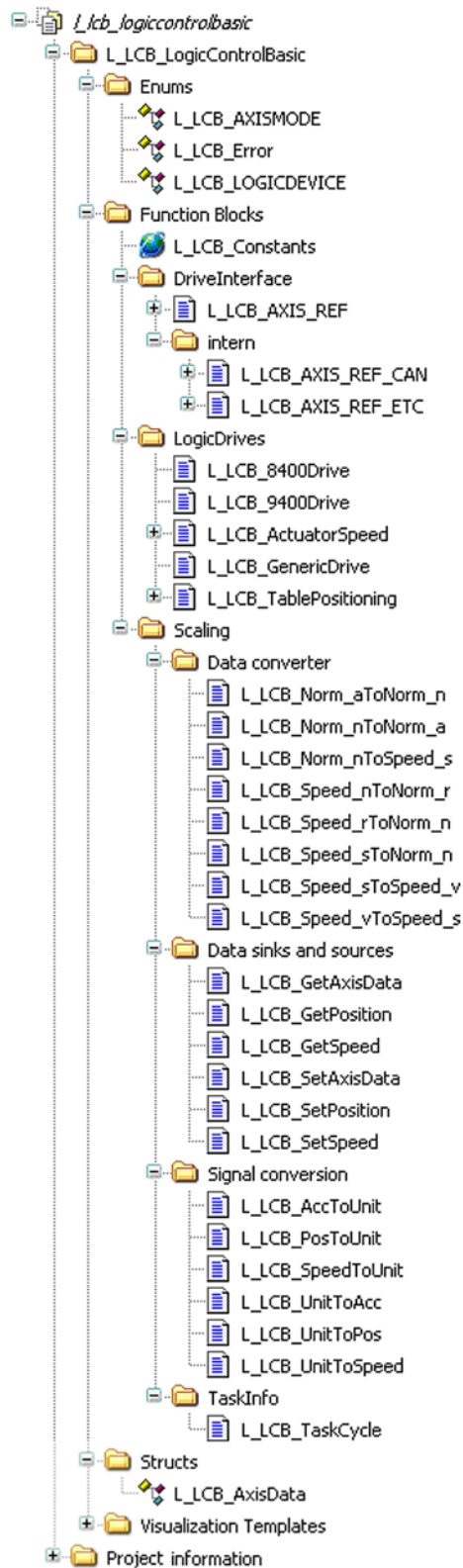
Zur Anbindung eines Funktionsbausteins an einen Inverter muss die Achsdaten-Instanz aus der Steuerungskonfiguration mit dem Funktionsbaustein verbunden werden.



[8-1] Beispielverschaltung: Achsdaten-Instanz eines 9400 HighLine mit dem Funktionsbaustein **L_LCB_TablePositioning**

8.1 Übersicht der Funktionen und Funktionsbausteine

Die Funktionen und Funktionsbausteine der Bibliothek **L_LCB_LogicControlBasic** sind in verschiedene Gruppen unterteilt.



Enums

- [L_LCB_AXISMODE](#) (102)
- [L_LCB_Error](#) (102)
- [L_LCB_LOGICDEVICE](#) (102)

Function Blocks

DriveInterface

- [L_LCB_AXIS_REF](#) (67)
- [L_LCB_AXIS_REF_CAN \(interne Verwendung\)](#) (68)
- [L_LCB_AXIS_REF_ETC \(interne Verwendung\)](#) (68)

LogicDrives

- [L_LCB_8400Drive](#) (69)
- [L_LCB_9400Drive](#) (70)
- [L_LCB_ActuatorSpeed](#) (73)
- [L_LCB_GenericDrive](#) (72)
- [L_LCB_TablePositioning](#) (77)

Data converter

- [L_LCB_Norm_aToNorm_n](#) (83)
- [L_LCB_Norm_nToNorm_a](#) (83)
- [L_LCB_Norm_nToSpeed_s](#) (84)
- [L_LCB_Speed_nToNorm_r](#) (84)
- [L_LCB_Speed_rToNorm_n](#) (85)
- [L_LCB_Speed_sToNorm_n](#) (85)
- [L_LCB_Speed_sToSpeed_v](#) (86)
- [L_LCB_Speed_vToSpeed_s](#) (86)

Data sinks and sources

- [L_LCB_GetAxisData](#) (87)
- [L_LCB_GetPosition](#) (88)
- [L_LCB_GetSpeed](#) (89)
- [L_LCB_SetAxisData](#) (90)
- [L_LCB_SetPosition](#) (92)
- [L_LCB_SetSpeed](#) (93)

Signal conversion

- [L_LCB_AccToUnit](#) (94)
- [L_LCB_PosToUnit](#) (95)
- [L_LCB_SpeedToUnit](#) (96)
- [L_LCB_UnitToAcc](#) (97)
- [L_LCB_UnitToPos](#) (98)
- [L_LCB_UnitToSpeed](#) (99)

TaskInfo

- [L_LCB_TaskCycle](#) (100)

Structs

- [L_LCB_AxisData](#) (101)

8.2 Funktionsbausteine

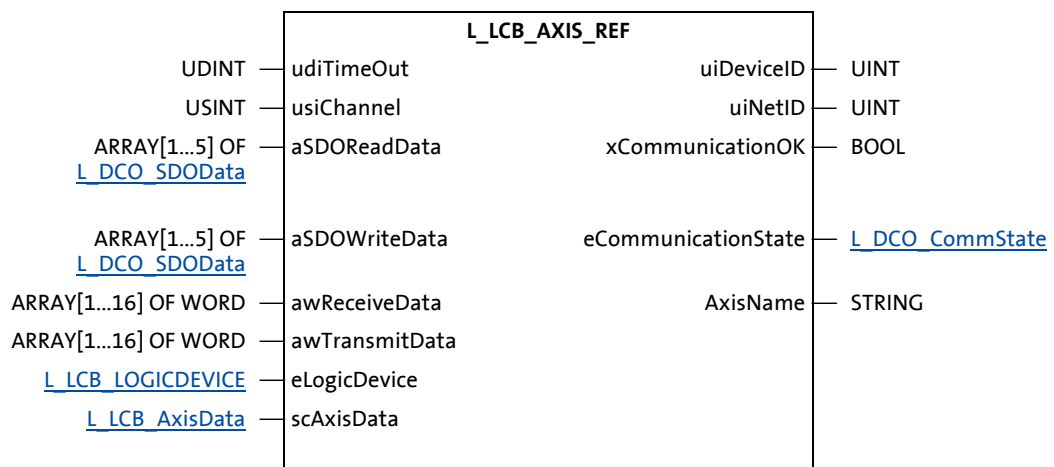
8.2.1 L_LCB_AXIS_REF

Diese Funktionsbaustein-Instanz stellt einen Logic-Antrieb in der Applikation dar.

Jede in der Steuerungskonfiguration eingebundene Logic-Achse ist eine erweiterte Instanz von **L_LCB_AXIS_REF**. Eine manuelle Instanzierung dieses Funktionsbausteins ist nicht erforderlich.

Über diesen Funktionsbaustein werden die Prozessdaten der verbundenen Logic-Achse der Applikation zur Verfügung gestellt.

L_LCB_AXIS_REF ist eine Erweiterung des Funktionsbausteins **L_DCO_AXIS_REF_BASE** (105).



Eingänge

Bezeichner/Datentyp	Info/Einstellmöglichkeiten
udiTimeOut UDINT	Timeout-Zeit für die SDO-Kommunikation • Initialwert: 10000
usiChannel USINT	Zu verwendender SDO-Kanal • Initialwert: 1
aSdoReadData ARRAY[1...5] OF L_DCO_SDOData	Austausch für die SDO-Kommunikation (interne Verwendung) • Initialwert: 0
aSdoWriteData ARRAY[1...5] OF L_DCO_SDOData	Austausch für die SDO-Kommunikation (interne Verwendung) • Initialwert: 0
awReceiveData ARRAY[1...16] OF WORD	Schnittstelle für die vom Inverter empfangenen Prozessdaten. • Initialwert: 0
awTransmitData ARRAY[1...16] OF WORD	Schnittstelle für die zum Inverter zu sendenden Prozessdaten. • Initialwert: 0
eLogicDrive L_LCB_LOGICDEVICE	Typ des angeschlossenen Inverters
scAxisData L_LCB_AxisData	Maschinenparameter des angeschlossenen Inverters zur Umrechnung von Maschineneinheiten in interne Einheiten.

Ausgänge

Bezeichner/Datentyp	Wert/Bedeutung
uiDeviceld UINT	Stations-/Knotenadresse der Achse im verwendeten Bussystem • Initialwert: 0
uiNetId UINT	Netzwerk/Master-Instanznummer des verwendeten Bussystems • Initialwert: 0
xCommunicationOK BOOL	TRUE Buskommunikation vorhanden
	FALSE Keine Buskommunikation aktiv
eCommunicationState L_DCO_CommState	Busstatus • Initialwert: 0
AxisName STRING	Bezeichnung der Achse

8.2.2 L_LCB_AXIS_REF_CAN (interne Verwendung)

Feldbusspezifische Erweiterung des Funktionsbausteins [L_LCB_AXIS_REF](#) für das Bussystem CAN.

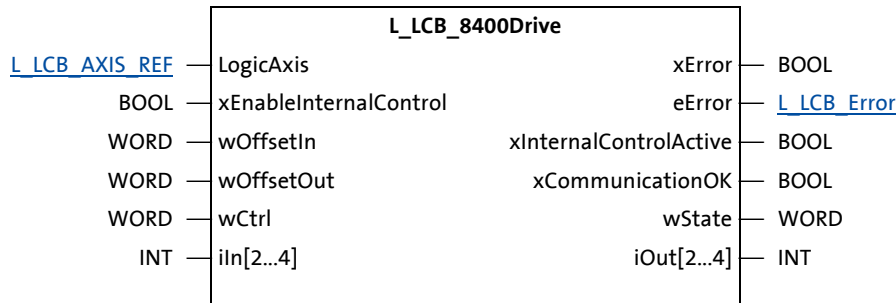
8.2.3 L_LCB_AXIS_REF_ETC (interne Verwendung)

Feldbusspezifische Erweiterung des Funktionsbausteins [L_LCB_AXIS_REF](#) für das Bussystem EtherCAT.

8.2.4 L_LCB_8400Drive

Dieser Funktionsbaustein dient zur Ansteuerung von Inverter Drives 8400.

Die Kommunikation zu den Geräten ist über die Bussysteme CAN oder EtherCAT möglich.



Eingänge

Bezeichner/Datentyp	Info/Einstellmöglichkeiten	
LogicAxis L_LCB_AXIS_REF	Strukturdaten zur Antriebssteuerung	
xEnableInternalControl BOOL	TRUE	Die interne Steuerung des Inverters ist über die Schaltfläche "Internal Control" in der Visualisierung aktivierbar.
xOffsetIn WORD	Verschiebung des Zugriffs auf Prozessdaten für Eingänge um <i>wOffsetIn</i> x 8 Byte Beispiele: • '0' für CanIn PDO 1 • '1' für CanIn PDO 2	
xOffsetOut WORD	Verschiebung des Zugriffs auf Prozessdaten für Ausgänge um <i>wOffsetOut</i> x 8 Byte Beispiele: • '0' für CanOut PDO 1 • '1' für CanOut PDO 2	
wCtrl WORD	Schnittstelle des Inverter Drive 8400	
iIn[2...4] INT	Schnittstellen des Inverter Drive 8400	

Ausgänge

Bezeichner/Datentyp	Wert/Bedeutung	
xError BOOL	TRUE	Es ist ein Fehler aufgetreten. Details siehe <i>eError</i> .
eError L_LCB_Error	Wenn <i>xError</i> = TRUE: Fehlernummer zum aufgetretenen Fehler. ► Fehlermeldungen (102)	
xInternalControlActive BOOL	TRUE	Der Eingang <i>xEnableInternalControl</i> und die Schaltfläche "Internal Control" in der Visualisierung sind aktiviert.
xCommunicationOK BOOL	TRUE	Buskommunikation vorhanden
	FALSE	Keine Buskommunikation aktiv
wState WORD	Schnittstelle des Inverter Drive 8400	
iOut[2...4] INT	Schnittstellen des Inverter Drive 8400	

8.2.5 L_LCB_9400Drive

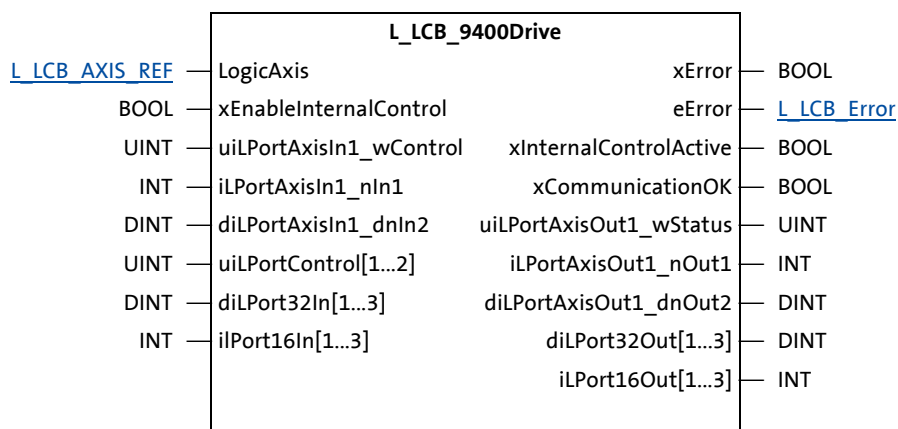
Dieser Funktionsbaustein dient zur Ansteuerung von Servo Drives 9400 mit einer beliebigen Applikation.

**Hinweis!**

Für eine fehlerfreie Funktion des Funktionsbausteins muss die Port-Konfiguration der Applikation auf dem Inverter beibehalten werden!

Es dürfen keine I/O-Ports gelöscht oder verschoben werden.

Die Kommunikation zu den Geräten ist über die Bussysteme CAN oder EtherCAT möglich.

**Eingänge**

Bezeichner/Datentyp	Info/Einstellmöglichkeiten	
LogicAxis L_LCB_AXIS_REF	Strukturdaten zur Antriebssteuerung	
xEnableInternalControl BOOL	TRUE	Die interne Steuerung des Inverters ist über die Schaltfläche "Internal Control" in der Visualisierung aktivierbar.
uiLPortAxisIn1_wControl UINT	Schnittstelle <i>LPortAxisIn1.wControl</i> des Servo Drive 9400	
iLPortAxisIn1_nIn1 INT	Schnittstelle <i>LPortAxisIn1.nIn1</i> des Servo Drive 9400	
diLPortAxisIn1_dnIn2 DINT	Schnittstelle <i>LPortAxisIn1.dnIn2</i> des Servo Drive 9400	
uiLPortControl1 UINT	Schnittstelle <i>LPortControl1</i> des Servo Drive 9400	
uiLPortControl2 UINT	Schnittstelle <i>LPortControl2</i> des Servo Drive 9400	
diLPort32In1 DINT	Schnittstelle <i>LPort32In1</i> des Servo Drive 9400	
diLPort32In2 DINT	Schnittstelle <i>LPort32In2</i> des Servo Drive 9400	
diLPort32In3 DINT	Schnittstelle <i>LPort32In3</i> des Servo Drive 9400	
iLPort16In1 INT	Schnittstelle <i>LPort16In1</i> des Servo Drive 9400	

Bezeichner/Datentyp	Info/Einstellmöglichkeiten
iLPort16In2 INT	Schnittstelle <i>LPort16In2</i> des Servo Drive 9400
iLPort16In3 INT	Schnittstelle <i>LPort16In3</i> des Servo Drive 9400

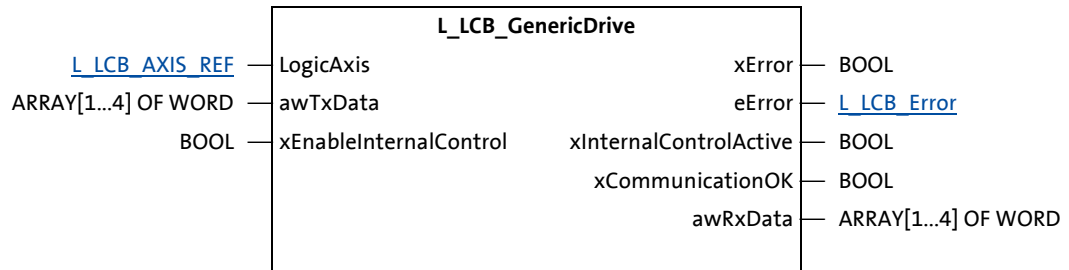
Ausgänge

Bezeichner/Datentyp	Wert/Bedeutung
xError BOOL	TRUE Es ist ein Fehler aufgetreten. Details siehe <i>eError</i> .
eError L_LCB_Error	Wenn <i>xError</i> = TRUE: Fehlernummer zum aufgetretenen Fehler. ► Fehlermeldungen (102)
xInternalControlActive BOOL	TRUE Der Eingang <i>xEnableInternalControl</i> und die Schaltfläche "Internal Control" in der Visualisierung sind aktiviert.
xCommunicationOK BOOL	TRUE Buskommunikation vorhanden
	FALSE Keine Buskommunikation aktiv
uiLPortAxisOut1_wStatus UINT	Schnittstelle <i>LPortAxisOut1.wStatus</i> des Servo Drive 9400
diLPortAxisOut1_dnOut1 INT	Schnittstelle <i>LPortAxisOut1.nOut1</i> des Servo Drive 9400
diLPortAxisOut1_dnOut2 DINT	Schnittstelle <i>LPortAxisOut1.dnOut2</i> des Servo Drive 9400
uiLPortStatus1 UINT	Schnittstelle <i>LPortStatus1</i> des Servo Drive 9400
uiLPortStatus2 UINT	Schnittstelle <i>LPortStatus2</i> des Servo Drive 9400
diLPort32Out1 DINT	Schnittstelle <i>LPort32Out1</i> des Servo Drive 9400
diLPort32Out2 DINT	Schnittstelle <i>LPort32Out2</i> des Servo Drive 9400
diLPort32Out3 DINT	Schnittstelle <i>LPort32Out3</i> des Servo Drive 9400
iLPort16Out1 INT	Schnittstelle <i>LPort16Out1</i> des Servo Drive 9400
iLPort16Out2 INT	Schnittstelle <i>LPort16Out2</i> des Servo Drive 9400
iLPort16Out3 INT	Schnittstelle <i>LPort16Out3</i> des Servo Drive 9400

8.2.6 L_LCB_GenericDrive

Dieser Funktionsbaustein dient zur Ansteuerung eines beliebigen Lenze-Inverters.

Die Kommunikation zu dem Gerät ist über CAN möglich.



Eingang

Bezeichner/Datentyp	Info/Einstellmöglichkeiten	
LogicAxis L_LCB_AXIS_REF	Strukturdaten zur Antriebssteuerung	
awTxData ARRAY[1...4] OF WORD	Zum Inverter zu übertragende Prozessdaten	
xEnableInternalControl BOOL	TRUE	Die interne Steuerung des Inverters ist über die Schaltfläche "Internal Control" in der Visualisierung aktivierbar.

Ausgänge

Bezeichner/Datentyp	Wert/Bedeutung	
xError BOOL	TRUE	Es ist ein Fehler aufgetreten. Details siehe <i>eError</i> .
eError L_LCB_Error	Wenn <i>xError</i> = TRUE: Fehlernummer zum aufgetretenen Fehler. ► Fehlermeldungen (102)	
xInternalControlActive BOOL	TRUE	Der Eingang <i>xEnableInternalControl</i> und die Schaltfläche "Internal Control" in der Visualisierung sind aktiviert.
xCommunicationOK BOOL	TRUE	Buskommunikation vorhanden
	FALSE	Keine Buskommunikation aktiv
awRxData ARRAY[1...4] OF WORD	Vom Inverter empfangene Prozessdaten	

8.2.7 L_LCB_ActuatorSpeed

Dieser Funktionsbaustein dient zur Ansteuerung von Lenze-Invertern mit der Applikation "Stellantrieb - Drehzahl".



Hinweis!

Der Funktionsbaustein funktioniert nur in Kombination mit einem Beispielprojekt "Stellantrieb - Drehzahl" (gekennzeichnet mit dem Kürzel "_Speed").

► [Übersicht der Application Samples](#) (20)

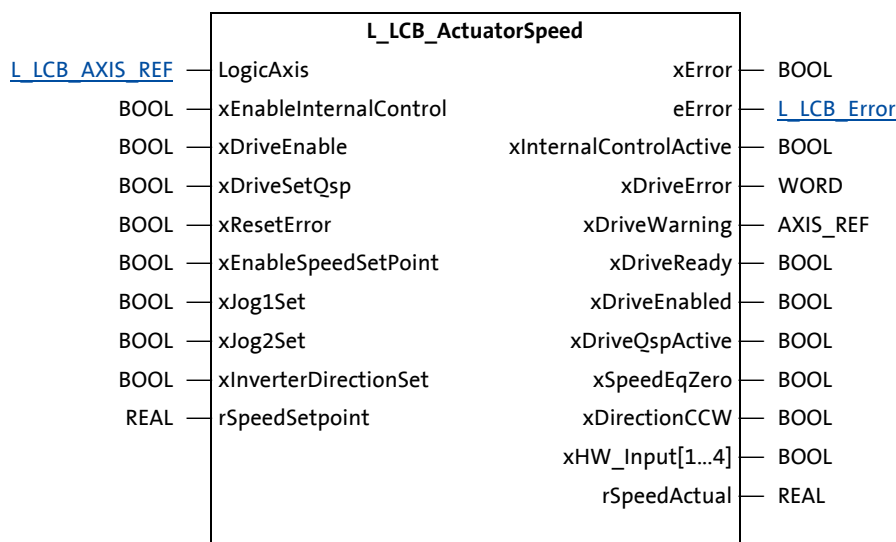
Zur Erstellung einer Funktionsbaustein-Verschaltung können folgende Bausteine verwendet werden:

► [L_LCB_8400Drive](#) (69)

► [L_LCB_9400Drive](#) (70)

► [L_LCB_GenericDrive](#) (72)

Die Kommunikation zu den Geräten ist über die Bussysteme CAN oder EtherCAT möglich.



Eingänge

Bezeichner/Datentyp	Info/Einstellmöglichkeiten	
LogicAxis L_LCB_AXIS_REF	Strukturdaten zur Antriebssteuerung	
xEnableInternalControl BOOL	TRUE	Die interne Steuerung des Inverters ist über die Schaltfläche "Internal Control" in der Visualisierung aktivierbar.
xDriveEnable BOOL	TRUE: Reglersperre aufheben (invertiert)	
xDriveSetQSP BOOL	TRUE: Schnellhalt aktivieren	
xResetError BOOL	TRUE: Fehlermeldung zurücksetzen (quittieren)	
xEnableSpeedSetpoint BOOL	TRUE: Drehzahlregelung aktiviert • Die Eingänge <i>xJogSet1</i> , <i>xJogSet2</i> und <i>rSpeedSetPoint</i> werden ausgewertet.	

Bezeichner/Datentyp	Info/Einstellmöglichkeiten
xJog1Set BOOL	TRUE: Jog 1 aktiviert • Drehzahlsollwert 1 angefordert • Aktiv, wenn <i>xEnableSpeedSetpoint</i> = TRUE
xJog2Set BOOL	TRUE: Jog 2 aktiviert • Drehzahlsollwert 2 angefordert • Aktiv, wenn <i>xEnableSpeedSetpoint</i> = TRUE
xInvertDirectionSet BOOL	TRUE: Invertierung der Drehrichtung des Motors (Rechtslauf/Linkslauf)
rSpeedSetpoint REAL	Sollwert-Vorgabe für die Drehzahl in [%] • Aktiv, wenn... • <i>xEnableSpeedSetpoint</i> = TRUE <u>und</u> <i>xJog1Set</i> = FALSE <u>und</u> <i>xJog2Set</i> = FALSE

Ausgänge

Bezeichner/Datentyp	Wert/Bedeutung
xError BOOL	TRUE Es ist ein Fehler aufgetreten. Details siehe <i>eError</i> .
eError L_LCB_Error	Wenn <i>xError</i> = TRUE: Fehlernummer zum aufgetretenen Fehler. ► Fehlermeldungen (102)
xCommunicationOK BOOL	TRUE Buskommunikation vorhanden FALSE Keine Buskommunikation aktiv
xInternalControlActive BOOL	TRUE Der Eingang <i>xEnableInternalControl</i> und die Schaltfläche "Internal Control" in der Visualisierung sind aktiviert.
xDriveError BOOL	Statussignal "Fehler im Inverter"
xDriveWarning BOOL	TRUE: Inverter befindet sich im Gerätezustand "Warning"
xDriveReady BOOL	Statussignal "Inverter ist betriebsbereit"
xDriveEnabled BOOL	TRUE: Inverter ist freigegeben
xDriveQSPActive BOOL	TRUE: Schnellhalt (QSP) aktiv
xSpeedEqZero BOOL	TRUE: Geschwindigkeit ist NULL
xDirectionCCW BOOL	Anzeige der Drehrichtung des Motors TRUE: Linkslauf aktiv
xHW_Input1 BOOL	Digitaler Eingang 1 aktiv
xHW_Input2 BOOL	Digitaler Eingang 2 aktiv
xHW_Input3 BOOL	Digitaler Eingang 3 aktiv
xHW_Input4 BOOL	Digitaler Eingang 4 aktiv
rSpeedActual REAL	Aktuelle Drehzahl der Motorwelle in [%] -199.99 ... 199.99 %

8.2.7.1 Belegung der Eingangs-Ports

Bezeichner/Datentyp	Servo Drive 9400 HighLine	Inverter Drive 8400 HighLine	WORD
xDriveEnable BOOL	NOT Bit 0	Bit 0 und 3	WORD 0
xDriveSetQSP BOOL	Bit 1	Bit 2	
xResetError BOOL	Bit 10	Bit 7	
xEnableSpeedSetpoint BOOL	Bit 2	-	
xJog1Set BOOL	Bit 3	Bit 12 und xEnableSpeedSetpoint	
xJog2Set BOOL	Bit 4	Bit 13 und xEnableSpeedSetpoint	
xInvertDirectionSet BOOL	Bit 5	Bit 15	
	xCtrl06	xCtrl01	
	xCtrl07	xCtrl04	
	xCtrl08	xCtrl05	
	xCtrl09	xCtrl06	
	-	xCtrl08	
	xCtrl11/xSetErrorSet	xCtrl09	
	xCtrl12	xCtrl10	
	xCtrl13/xEnableManual	xCtrl11	
	xCtrl14/xManualPositive	xCtrl14/xSetError	
	xCtrl15/xManualNegative	frei	
	frei	rSpeedSetpoint	WORD 1
	rSpeedSetpoint	frei	WORD 2
		frei	WORD 3

8.2.7.2 Belegung der Ausgangs-Ports

Bezeichner/Datentyp	Servo Drive 9400 HighLine	Inverter Drive 8400 HighLine	WORD
xDriveError BOOL	Bit 0	Bit 0	WORD 0
xDriveWarning BOOL	Bit 12	Bit 12	
xDriveReady BOOL	Bit 2	Bit 2	
xDriveEnabled BOOL	NOT Bit 7	Bit 7	
xDriveQSPActive BOOL	Bit 3	Bit 3	
xSpeedEqZero BOOL	Bit 6	Bit 6	
xDirectionCCW BOOL	-	-	
xHW_Input1 BOOL	Bit 4	Bit 4/Digital_In1	
xHW_Input2 BOOL	Bit 5	Bit 5/Digital_In2	
xHW_Input3 BOOL	Bit 14	Bit 14/Digital_In3	
xHW_Input4 BOOL	Bit 15	Bit 15/Digital_In4	
	xStat08/SF_active	xStat08/StatusCodeBit1	
	xStat09	xStat09/StatusCodeBit1	
	xStat10	xStat10/StatusCodeBit1	
	xStat11	xStat11/StatusCodeBit1	
	xStat13	xStat13/Message	
	frei	rSpeedActual	WORD 1
	rSpeedActual	frei	WORD 2

8.2.8 L_LCB_TablePositioning

Dieser Funktionsbaustein dient zur Ansteuerung von Inverter Drives 8400 HighLine und Servo Drives 9400 HighLine mit der Applikation "Tabellenpositionierung".



Hinweis!

Der Funktionsbaustein funktioniert nur in Kombination mit einem Beispielprojekt "Tabellenpositionierung" (gekennzeichnet mit dem Kürzel "_TabPos").

► [Übersicht der Application Samples](#) (📖 20)

Zur Erstellung einer Funktionsbaustein-Verschaltung können folgende Bausteine verwendet werden:

► [L_LCB_8400Drive](#) (📖 69)

► [L_LCB_9400Drive](#) (📖 70)

► [L_LCB_GenericDrive](#) (📖 72)

Die Kommunikation zu den Geräten ist über die Bussysteme CAN oder EtherCAT möglich.

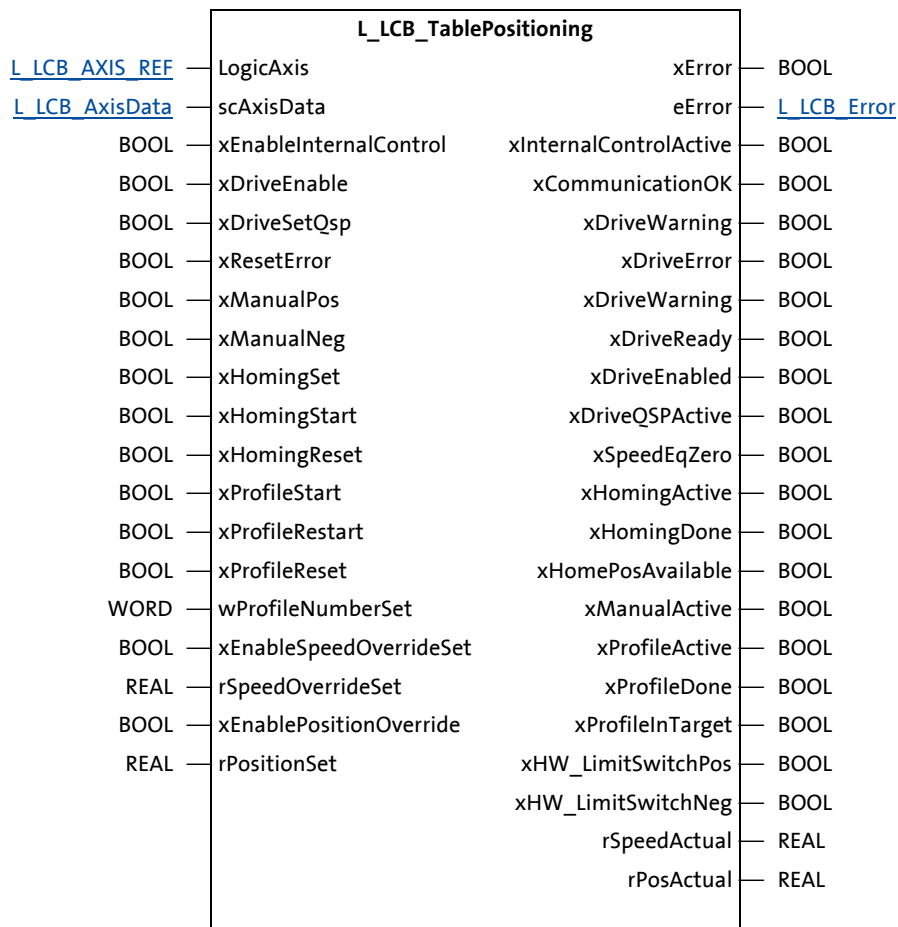
"Position Override" aktivieren

Um eine Positionierung durchführen zu können, muss der Inverter freigegeben sein (Eingang *xDriveEnable* auf TRUE setzen) und die Referenzposition gesetzt sein (z. B. über den Eingang *xHomingSet*).



So aktivieren Sie einen "Position Override":

1. Den Eingang *xEnablePositionOverride* auf TRUE setzen.
2. Die gewünschte Sollposition am Eingang *rPositionSet* setzen.
3. Am Eingang *wProfileNumberSet* die Nummer des Profildatensatzes für die Positionierung setzen.
4. Den Positioniervorgang durch Setzen des Eingangs *xProfileStart* auf TRUE starten.
 - Die Position wird in die Codestelle/den Parameter des ausgewählten Profildatensatzes geschrieben.
 - Die Positionierung erfolgt mit der Geschwindigkeit und Beschleunigung des ausgewählten Profildatensatzes.



Eingänge

Bezeichner/Datentyp	Info/Einstellmöglichkeiten	
LogicAxis L_LCB_AXIS_REF	Strukturdaten zur Antriebssteuerung	
scAxisData L_LCB_AxisData	Maschinenparameter des anzusteuernenden Inverters	
xEnableInternalControl BOOL	TRUE	Die interne Steuerung des Inverters ist über die Schaltfläche "Internal Control" in der Visualisierung aktivierbar.
xDriveEnable BOOL	TRUE	Reglersperre aufheben (invertiert)
xDriveSetQSP BOOL	TRUE	Schnellhalt (QSP) aktivieren
xResetError BOOL	TRUE	Fehlermeldung zurücksetzen (quittieren)
xManualPos BOOL	TRUE	Handfahren in positive Richtung (Rechtslauf)
xManualNeg BOOL	TRUE	Handfahren in negative Richtung (Linkslauf)
xHomingSet BOOL	TRUE	Referenzposition setzen
xHomingStart BOOL	TRUE	Referenzfahrt starten

Bezeichner/Datentyp	Info/Einstellmöglichkeiten
xHomingReset BOOL	TRUE Referenzposition zurücksetzen
xProfileStart BOOL	Positionierung starten/stoppen
	TRUE Positioniervorgang starten/Neustart • Das zu fahrende Profil am Eingang <i>wProfileNumberSet</i> vorgeben. • Während einer aktiven Positionierung kann über <i>wProfileNumberSet</i> ein anderes Profil vorgegeben werden, das nach dem Neustart (erneuter Zustandswechsel von TRUE auf FALSE) abgefahren wird. • Eine zuvor abgebrochene Positionierung wird fortgesetzt.
	FALSE Die aktive Positionierung abbrechen.
xProfileRestart BOOL	FALSE↗TRUE Aktualisierung der Profildaten eines gerade aktiven Profils unter Berücksichtigung des bereits zurückgelegten Weges.
xProfileReset BOOL	FALSE↗TRUE Ein zuvor abgebrochenes Profil wird nicht mehr fortgesetzt. • Mit <i>xProfileStart</i> = TRUE erfolgt der Neustart eines Profils.
wProfileNumberSet WORD	Profilnummer des zu fahrenden Profils aus der Profildatenverwaltung • Servo Drives 9400: 1 ... 75 • Inverter Drives 8400: 3 ... 15
xEnableSpeedOverride BOOL	TRUE Geschwindigkeits-Override einschalten
rSpeedOverrideSet REAL	Wert für den Geschwindigkeits-Override in [%] • Einheit: % • Wertebereich: 0.00 ... 199.99
xEnablePositionOverride BOOL	TRUE "Position Override" einschalten Hinweis: Für das Bussystem CAN muss das zweite Empfangs-PDO (RPDO) im »PLC Designer« freigeschaltet werden (Registerkarte "PDO Mapping"). Für das Bussystem EtherCAT ist diese Einstellung nicht erforderlich.
rPositionSet REAL	Positionsvorgabe • Einheit: Units

Ausgänge

Bezeichner/Datentyp	Wert/Bedeutung
xError BOOL	TRUE Es ist ein Fehler aufgetreten. Details siehe <i>eError</i> .
eError L_LCB_Error	Wenn <i>xError</i> = TRUE: Fehlernummer zum aufgetretenen Fehler. ► Fehlermeldungen (102)
xCommunicationOK BOOL	TRUE Buskommunikation vorhanden
	FALSE Keine Buskommunikation aktiv
xInternalControlActive BOOL	TRUE Der Eingang <i>xEnableInternalControl</i> und die Schaltfläche "Internal Control" in der Visualisierung sind aktiviert.
xDriveError BOOL	TRUE Ein Fehler ist im Inverter aktiv (Quittierung erforderlich).
xDriveWarning BOOL	TRUE Der Inverter befindet sich im Gerätezustand "Warning".
xDriveReady BOOL	TRUE Der Inverter ist betriebsbereit.
xDriveEnabled BOOL	TRUE Der Inverter ist freigegeben.
xDriveQSPActive BOOL	TRUE Ein Schnellhalt (QSP) durch Applikation ist aktiv.

Bezeichner/Datentyp		Wert/Bedeutung	
xSpeedEqZero	BOOL	TRUE	Der Inverter befindet sich im Stillstand • Drehzahlwert in Codestelle C00019 beachten!
xHomingActive	BOOL	TRUE	Die Referenzfahrt ist aktiviert.
xHomingDone	BOOL	TRUE	Die Referenzfahrt wurde beendet.
xHomePosAvailable	BOOL	TRUE	Die Referenzposition ist bekannt.
xManualActive	BOOL	TRUE	Das Handfahren ist aktiviert.
xProfileActive	BOOL	TRUE	Die Positionierung ist aktiviert.
xProfileDone	BOOL	TRUE	Die Positionierung wurde beendet.
xProfileInTarget	BOOL	TRUE	Die Zielposition wurde erreicht.
xHW_LimitSwitchPos	BOOL	TRUE	Der positive Hardware-Endschalter wurde angefahren.
xHW_LimitSwitchNeg	BOOL	TRUE	Der negative Hardware-Endschalter wurde angefahren.
rSpeedActual	REAL	Aktuelle Geschwindigkeit • Einheit: %	
rPositionActual	REAL	Aktuelle Position • Einheit: Units	

8.2.8.1 Belegung der Eingangs-Ports

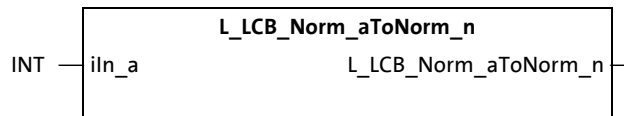
Bezeichner/Datentyp	Servo Drive 9400 HighLine	Inverter Drive 8400 HighLine	WORD
xDriveEnable BOOL	NOT Bit 0	Bit 3	WORD 0
xDriveSetQSP BOOL	Bit 1	Bit 15	
xResetError BOOL	Bit 10	Bit 11	
xManualPos BOOL	Bit 14	Bit 12	
xManualNeg BOOL	Bit 15	Bit 13	
xHomingSet BOOL	Bit 12	Bit 10	
xHomingStart BOOL	Bit 3	Bit 8	
xHomingReset BOOL	Bit 4	Bit 14	
xProfileStart BOOL	Bit 5	Bit 0	
	Bit 6	Bit 1	
	-	Bit 8	
xProfileRestart BOOL	Bit 7	Bit 0	
	-	Bit 1	
	-	Bit 8	
xProfileReset BOOL	Bit 8	Bit 0	
	-	Bit 1	
	-	Bit 8	
xEnableSpeedOverride BOOL	Bit 9	Bit 9	
xEnablePositionOverride REAL	Bit 2	Bit 2	
		wProfileNumberSet.0 Bit 4	
		wProfileNumberSet.0 Bit 5	
		wProfileNumberSet.0 Bit 6	
		wProfileNumberSet.0 Bit 7	
	rSpeedOverrideSet	rSpeedOverrideSet	WORD 1
	wProfileNumberSet	rPositionSet	WORD 2

8.2.8.2 Belegung der Ausgangs-Ports

Bezeichner/Datentyp	Servo Drive 9400 HighLine	Inverter Drive 8400 HighLine	WORD
xDriveError BOOL	Bit 0	Bit 0	WORD 0
xDriveWarning BOOL	Bit 9	Bit 15	
xDriveReady BOOL	Bit 2	Bit 4	
xDriveEnabled BOOL	NOT Bit 1	NOT Bit 3	
xDriveQSPActive BOOL	Bit 3	Bit 14	
xSpeedEqZero BOOL	Bit 4	Bit 11	
xHomingActive BOOL	Bit 5	Bit 9	
xHomingDone BOOL	Bit 6	Bit 6	
xHomePosAvailable BOOL	Bit 7	Bit 10	
xManualActive BOOL	Bit 8	Bit 9	
xProfileActive BOOL	Bit 10	Bit 9	
xProfileDone BOOL	Bit 11	Bit 7	
xProfileInTarget BOOL	Bit 12	Bit 8	
xHW_LimitSwitchPos BOOL	Bit 14	Bit 12	
xHW_LimitSwitchNeg BOOL	Bit 15	Bit 13	
rSpeedActual REAL			WORD 1
rPositionActual REAL			WORD 2

8.2.9 L_LCB_Norm_aToNorm_n

Diese Funktion konvertiert ein 16-Bit Eingangssignal in ein 32-Bit Ausgangssignal.



Eingang

Bezeichner/Datentyp	Info/Einstellmöglichkeiten
iln_a INT	Eingangssignal • $100\% \equiv 2^{14} = 16384$



Hinweis!

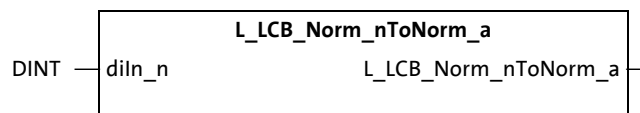
Hat die Eingangsinformation den Wert '-32768', begrenzt der Funktionsbaustein den Wert intern auf '-32767' und kopiert diesen in das High Word des Ausgangs *dnOut_n*.
Mit diesem Funktionsbaustein ist es deshalb nicht möglich, Werte vom Datentyp "Low Word" und "High Word" zu einem Wert vom Typ "DWORD" zusammenzubauen.

Ausgangssignal

L_LCB_Norm_aToNorm_n ($100\% \equiv 2^{30} = 1073741824$)

8.2.10 L_LCB_Norm_nToNorm_a

Diese Funktion konvertiert ein 32-Bit Eingangssignal in ein 16-Bit Ausgangssignal.



Eingang

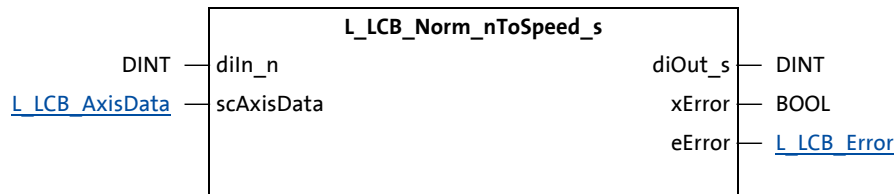
Bezeichner/Datentyp	Info/Einstellmöglichkeiten
diIn_n DINT	Eingangssignal • Einheit: % ($100\% \equiv 2^{30} = 1073741824$)

Ausgangssignal

L_LCB_Norm_nToNorm_a ($100\% \equiv 2^{14} = 16384$)

8.2.11 L_LCB_Norm_nToSpeed_s

Dieser Funktionsbaustein verwendet die übertragenen Maschinenparameter und konvertiert ein 32-Bit Eingangssignal in ein 32-Bit Speed-Signal.



Eingang

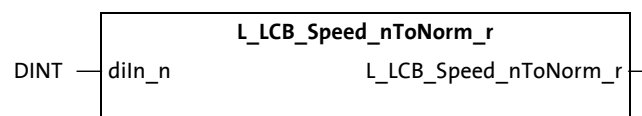
Bezeichner/Datentyp	Info/Einstellmöglichkeiten
diIn_n DINT	Eingangssignal • Einheit: % (100 % = $2^{30} = 1073741824$)
scAxisData L_LCB_AxisData	Maschinenparameter

Ausgänge

Bezeichner/Datentyp	Wert/Bedeutung
diOut_s DINT	Ausgangssignal • Einheit: min^{-1} ($15000 \text{ min}^{-1} = 2^{26} = 67108864$)
xError BOOL	TRUE Es ist ein Fehler aufgetreten. Details siehe <i>eError</i> .
eError L_LCB_Error	Wenn <i>xError</i> = TRUE: Fehlernummer zum aufgetretenen Fehler. ► Fehlermeldungen (102)

8.2.12 L_LCB_Speed_nToNorm_r

Diese Funktion konvertiert ein 32-Bit Eingangssignal in ein Speed-Signal vom Typ "REAL".



Eingang

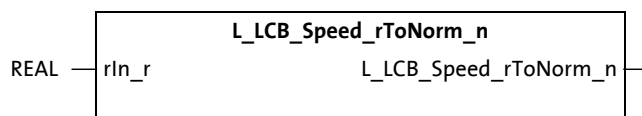
Bezeichner/Datentyp	Info/Einstellmöglichkeiten
diIn_n DINT	Eingangssignal • Einheit: % (100 % = $2^{30} = 1073741824$)

Ausgangssignal

L_LCB_Speed_nToNorm_r (REAL 0.0 ... 100.0 %)

8.2.13 L_LCB_Speed_rToNorm_n

Diese Funktion konvertiert ein Eingangssignal vom Typ "REAL" in ein 32-Bit Speed-Signal.



Eingang

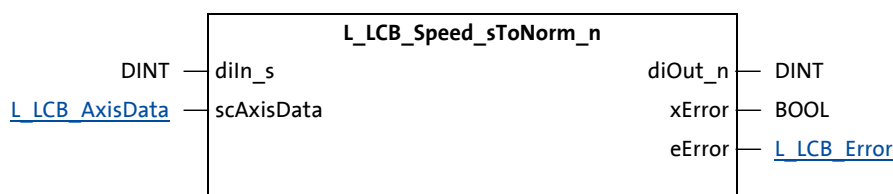
Bezeichner/Datentyp	Info/Einstellmöglichkeiten
rIn_r REAL	Eingangssignal • Einheit: % (0 ... 100 %)

Ausgangssignal

L_LCB_Speed_rToNorm_n (0 ... 2^{30} = 0 ... 1073741824)

8.2.14 L_LCB_Speed_sToNorm_n

Dieser Funktionsbaustein verwendet die übertragenen Maschinenparameter und konvertiert ein 32-Bit Speed-Signal in ein 32-Bit Ausgangssignal.



Eingänge

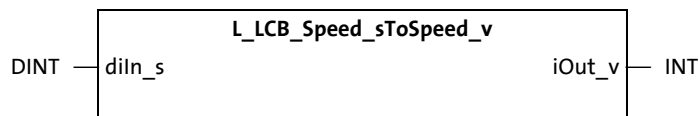
Bezeichner/Datentyp	Info/Einstellmöglichkeiten
diIn_s DINT	Eingangssigna • Einheit: min^{-1} ($15000 \text{ min}^{-1} = 2^{26} = 67108864$)
scAxisData L_LCB_AxisData	Maschinenparameter

Ausgänge

Bezeichner/Datentyp	Wert/Bedeutung
diOut_n DINT	Ausgangssignal • Einheit: % ($100 \% = 2^{30} = 1073741824$)
xError BOOL	TRUE Es ist ein Fehler aufgetreten. Details siehe <i>eError</i> .
eError L_LCB_Error	Wenn <i>xError</i> = TRUE: Fehlernummer zum aufgetretenen Fehler. ► Fehlermeldungen (102)

8.2.15 L_LCB_Speed_sToSpeed_v

Diese Funktion konvertiert ein 32-Bit Speed-Signal in ein 16-Bit Speed-Signal.



Eingang

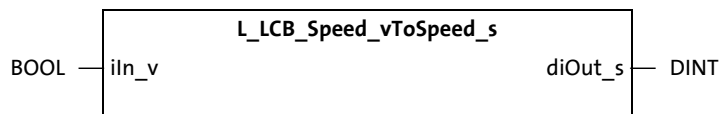
Bezeichner/Datentyp	Info/Einstellmöglichkeiten
diIn_s DINT	Eingangssigna • Einheit: min^{-1} ($15000 \text{ min}^{-1} \approx 2^{26} = 67108864$)

Ausgang

Bezeichner/Datentyp	Wert/Bedeutung
iOut_v INT	Ausgangssignal • Einheit: min^{-1} ($15000 \text{ min}^{-1} \approx 2^{14} = 16384$)

8.2.16 L_LCB_Speed_vToSpeed_s

Diese Funktion konvertiert ein 16-Bit Speed-Signal in ein 32-Bit Speed-Signal.



Eingang

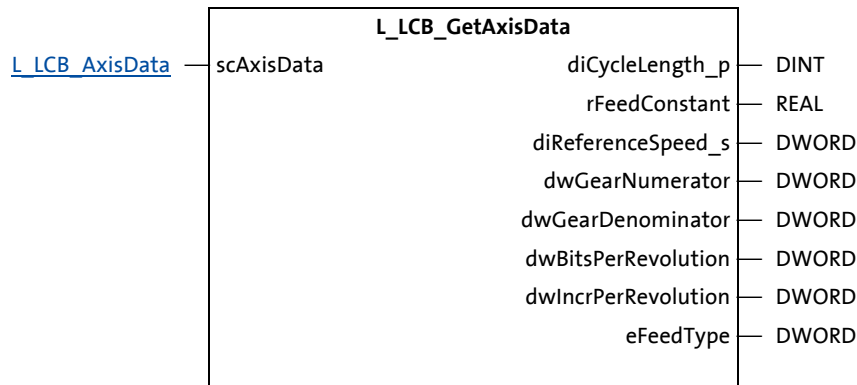
Bezeichner/Datentyp	Info/Einstellmöglichkeiten
iln_v INT	Eingangssignal • Einheit: min^{-1} ($15000 \text{ min}^{-1} \approx 2^{14} = 16384$)

Ausgang

Bezeichner/Datentyp	Wert/Bedeutung
diOut_s DINT	Ausgangssigna • Einheit: min^{-1} ($15000 \text{ min}^{-1} \approx 2^{26} = 67108864$)

8.2.17 L_LCB_GetAxisData

Dieser Funktionsbaustein liest aus den übergebenen Achsdaten ausgewählte Maschinenparameter aus und stellt sie nachfolgenden Funktionsbausteinen zur Verfügung, die diese Daten für interne Berechnungen benötigen.



Eingang

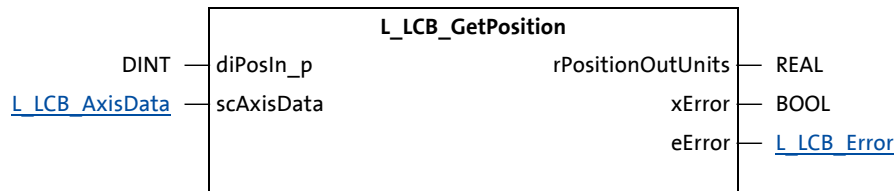
Bezeichner/Datentyp	Info/Einstellmöglichkeiten
scAxisData L_LCB_AxisData	Maschinenparameter

Ausgänge

Bezeichner/Datentyp	Wert/Bedeutung
diCycleLength_p DINT	Taktlänge • Einheit: Inkremente
rFeedConstant REAL	Taktlänge
diReferenceSpeed_s DWORD	Bezugsgeschwindigkeit als Drehzahlsignal
dwGearNumerator DWORD	Getriebefaktor-Zähler
dwGearDenominator DWORD	Getriebefaktor-Nenner
dwBitsPerRevolution DWORD	Auflösung einer Geberumdrehung • Einheit: Bit (10 ... 24 Bits)
dwIncrPerRevolution DWORD	Auflösung einer Umdrehung • Einheit: Inkremente
eFeedType L_LCB_AXISMODE	Fahrbereich
	0 Unbegrenzt
	1 Begrenzt
	2 Modulo

8.2.18 L_LCB_GetPosition

Dieser Funktionsbaustein verwendet die übertragenen Maschinenparameter und konvertiert den übermittelten Positionswert in Inkrementen zu einem Positionswert in der realen Einheit der Maschine.



Eingänge

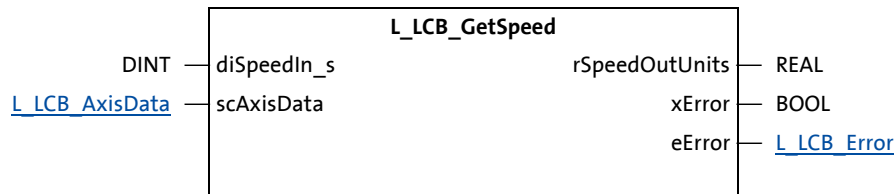
Bezeichner/Datentyp	Info/Einstellmöglichkeiten
diPosIn_p DINT	Positionswert - Einheit: Inkremente - Normierung: 1 Geberumdrehung = 2^{16} Inkremente (oder gemäß scAxisData)
scAxisData L_LCB_AxisData	Maschinenparameter

Ausgänge

Bezeichner/Datentyp	Wert/Bedeutung
rPositionOutUnits REAL	Anzeige des Positionswertes vom Eingangssignal <i>dnPosIn_p</i> in der realen Einheit der Maschine - Einheit: Units - Anzeige in Codestelle C04273
xError BOOL	TRUE Es ist ein Fehler aufgetreten. Details siehe <i>eError</i> .
eError L_LCB_Error	Wenn <i>xError</i> = TRUE: Fehlernummer zum aufgetretenen Fehler. ► Fehlermeldungen (102)

8.2.19 L_LCB_GetSpeed

Dieser Funktionsbaustein verwendet die übertragenen Maschinenparameter und konvertiert die übermittelte Drehzahl zu einem Geschwindigkeitswert.



Eingänge

Bezeichner/Datentyp	Info/Einstellmöglichkeiten
diSpeedIn_s DINT	Drehzahl • Einheit: min^{-1} ($15000 \text{ min}^{-1} \approx 2^{26} = 67108864$)
scAxisData L_LCB_AxisData	Maschinenparameter

Ausgänge

Bezeichner/Datentyp	Wert/Bedeutung
rSpeedOutUnits REAL	Geschwindigkeitswert • Anzeige in Codestelle C04283
xError BOOL	TRUE Es ist ein Fehler aufgetreten. Details siehe <i>eError</i> .
eError L_LCB_Error	Wenn <i>xError</i> = TRUE: Fehlernummer zum aufgetretenen Fehler. ► Fehlermeldungen (102)

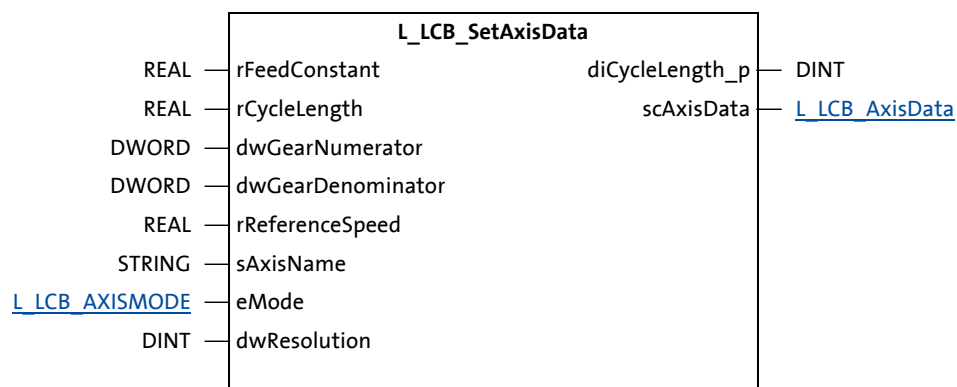
8.2.20 L_LCB_SetAxisData

Mit diesem Funktionsbaustein lassen sich die Maschinenparameter eines übergeordneten Antriebs abbilden. Der Baustein bereitet die Maschinenparameter für die interne Darstellung auf, die in den physikalischen Einheiten der Maschine über Parameter vorgegeben werden.

Am Ausgang *scAxisData* gibt der Baustein einen Pointer auf die Datenstruktur [L_LCB_AxisData](#) (101) mit den aufbereiteten Maschinenparametern aus.

Um die Maschinenparameter an einen Funktionsbaustein zu übergeben, der diese Daten für interne Berechnungen benötigt, muss der Ausgang *scAxisData* mit dem gleichnamigen Eingang des entsprechenden Funktionsbausteins verbunden werden.

Parameter, die über diesen Funktionsbaustein eingegeben werden, werden nicht direkt zu einem Antriebsregler übertragen. Diese Parameter werden nur innerhalb der PLC-Applikation für die bewegungsausführenden Funktionsbausteine verwendet (z. B. L_LCB_Tablepositioning).



Eingänge

Bezeichner/Datentyp	Info/Einstellmöglichkeiten
rFeedConstant REAL	Vorschubkonstante - Einheit: Units - Initialwert: 360.0
rCycleLength REAL	Taktlänge - Einheit: Units - Initialwert: 360.0
dwGearNumerator DWORD	Getriebefaktor-Zähler Initialwert: 1
dwGearDenominator DWORD	Getriebefaktor-Nenner - Initialwert: 1 Beispiel: Getriebeübersetzung 10:1, d. h. der Motor dreht 10-mal schneller als der Abtrieb. Die Parametrierung dazu ist: - dwGearNumerator = 1 - dwGearDenominator = 10
rReferenceSpeed REAL	Bezugsgeschwindigkeit
sAxisName STRING(16)	Achsname

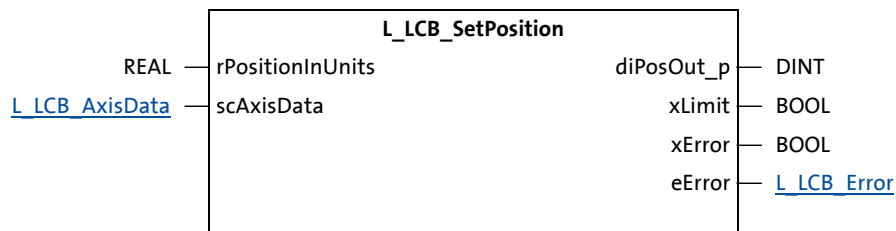
Bezeichner/Datentyp	Info/Einstellmöglichkeiten						
eMode L_LCB_AXISMODE	Fahrbereich Initialwert: 0 (Unbegrenzt) <table> <tr> <td>0</td><td>Unbegrenzt</td></tr> <tr> <td>1</td><td>Reserviert</td></tr> <tr> <td>2</td><td>Modulo</td></tr> </table>	0	Unbegrenzt	1	Reserviert	2	Modulo
0	Unbegrenzt						
1	Reserviert						
2	Modulo						
dwResolution DINT	Auflösung einer Umdrehung Beispiel: Bei einer Auflösung von 65535 Inkrementen pro Umdrehung muss folgender Wert eingestellt werden: dwResolution = 16 ($2^{16} = 65536$)						

Ausgänge

Bezeichner/Datentyp	Wert/Bedeutung
diCycleLength_p DINT	Taktlänge - Einheit: Inkremente - Die Taktlänge in der realen Einheit der Maschine ist in der entsprechenden Codestelle sichtbar.
scAxisData L_LCB_AxisData	Maschinenparameter - Pointer auf die Datenstruktur L_LCB_AxisData (101), die die Maschinenparameter im internen Maßsystem enthält. - Über diese Schnittstelle werden die Maschinenparameter anderen Funktionsbausteinen, die diese Daten für interne Berechnungen benötigen, zur Verfügung gestellt.

8.2.21 L_LCB_SetPosition

Dieser Funktionsbaustein rechnet, anhand der übergebenen Maschinenparameter, eine Position in 'Units' um in eine Position in 'Inkrementen' und stellt diese am Ausgang *dnPosOut_p* zur Weiterverarbeitung zur Verfügung.



Eingänge

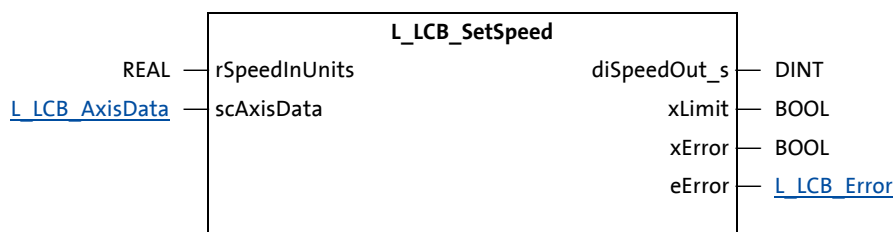
Bezeichner/Datentyp	Info/Einstellmöglichkeiten
rPositionInUnits REAL	Position • Einheit: Units
scAxisData L_LCB_AxisData	Maschinenparameter

Ausgänge

Bezeichner/Datentyp	Wert/Bedeutung
diPosOut_p DINT	Position • Einheit: Inkremente
xLimit BOOL	TRUE Das Ausgangssignal wird auf den darstellbaren Zahlenbereich ± 214748.3647 begrenzt.
xError BOOL	TRUE Es ist ein Fehler aufgetreten. Details siehe <i>eError</i> .
eError L_LCB_Error	Wenn <i>xError</i> = TRUE: Fehlernummer zum aufgetretenen Fehler. ► Fehlermeldungen (102)

8.2.22 L_LCB_SetSpeed

Dieser Funktionsbaustein rechnet, anhand der übergebenen Maschinenparameter, eine Geschwindigkeit um in eine Drehzahl und stellt diese am Ausgang *dnSpeedOut_s* zur Weiterverarbeitung zur Verfügung.



Eingänge

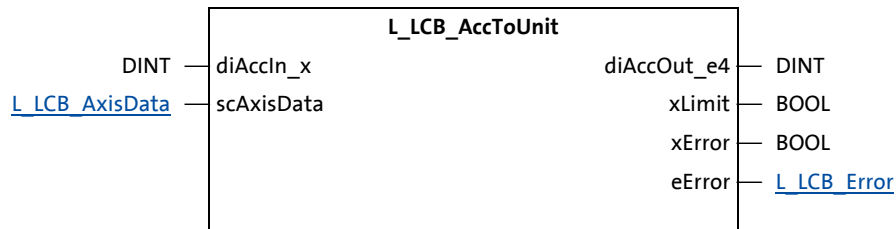
Bezeichner/Datentyp	Info/Einstellmöglichkeiten
rSpeedInUnits REAL	Geschwindigkeit • Einheit: Units
scAxisData L_LCB_AxisData	Maschinenparameter

Ausgänge

Bezeichner/Datentyp	Wert/Bedeutung
diSpeedOut_s DINT	Ausgabe der Geschwindigkeit als Drehzahl • Einheit: min^{-1} ($15000 \text{ min}^{-1} \approx 2^{26} = 67108864$)
xLimit BOOL	TRUE Das Ausgangssignal wird auf den darstellbaren Zahlenbereich ± 214748.3647 begrenzt.
xError BOOL	TRUE Es ist ein Fehler aufgetreten. Details siehe <i>eError</i> .
eError L_LCB_Error	Wenn <i>xError</i> = TRUE: Fehlernummer zum aufgetretenen Fehler. ► Fehlermeldungen (102)

8.2.23 L_LCB_AccToUnit

Dieser Funktionsbaustein rechnet, anhand der übergebenen Maschinenparameter, einen Beschleunigungswert in der internen Einheit [min^{-1}/s] um in die reale Einheit der Maschine.



Eingänge

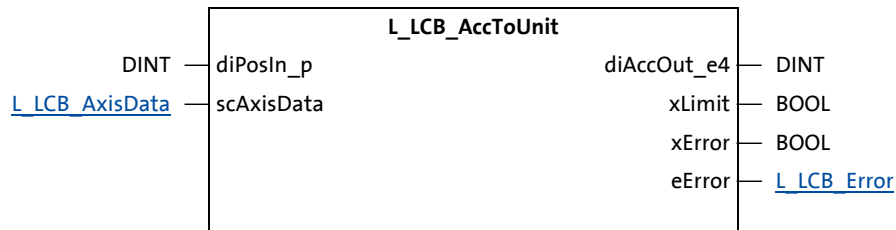
Bezeichner/Datentyp	Info/Einstellmöglichkeiten
diAccIn_x DINT	Beschleunigung als Drehzahländerung/Zeit • Einheit: min^{-1}/s ($15000 \text{ min}^{-1}/\text{s} \approx 2^{22} = 4194304$)
scAxisData L_LCB_AxisData	Maschinenparameter

Ausgänge

Bezeichner/Datentyp	Wert/Bedeutung
diAccOut_e4 DINT	Beschleunigung in der realen Einheit der Maschine • Einheit: Units/s^2 • Ausgabewert mit Fixkomma und vier Nachkommastellen
xLimit BOOL	TRUE Das Ausgangssignal wird auf den darstellbaren Zahlenbereich ± 214748.3647 begrenzt.
xError BOOL	TRUE Es ist ein Fehler aufgetreten. Details siehe <i>eError</i> .
eError L_LCB_Error	Wenn <i>xError</i> = TRUE: Fehlernummer zum aufgetretenen Fehler. ► Fehlermeldungen (102)

8.2.24 L_LCB_PosToUnit

Dieser Funktionsbaustein rechnet, anhand der übergebenen Maschinenparameter, einen Positionswert in Inkrementen um in die reale Einheit der Maschine.



Eingänge

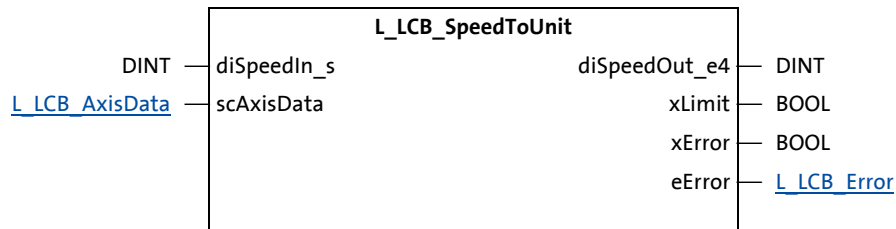
Bezeichner/Datentyp	Info/Einstellmöglichkeiten
diPosIn_p DINT	Position • Einheit: Inkremente
scAxisData L_LCB_AxisData	Maschinenparameter

Ausgänge

Bezeichner/Datentyp	Wert/Bedeutung
diPosOut_e4 DINT	Position in der realen Einheit der Maschine • Einheit: Units • Ausgabewert mit Fixkomma und vier Nachkommastellen
xLimit BOOL	TRUE Das Ausgangssignal wird auf den darstellbaren Zahlenbereich ± 214748.3647 begrenzt.
xError BOOL	TRUE Es ist ein Fehler aufgetreten. Details siehe <i>eError</i> .
eError L_LCB_Error	Wenn <i>xError</i> = TRUE: Fehlernummer zum aufgetretenen Fehler. ► Fehlermeldungen (102)

8.2.25 L_LCB_SpeedToUnit

Dieser Funktionsbaustein rechnet, anhand der übergebenen Maschinenparameter, einen Geschwindigkeitswert in der internen Einheit [min⁻¹] um in die reale Einheit der Maschine.



Eingänge

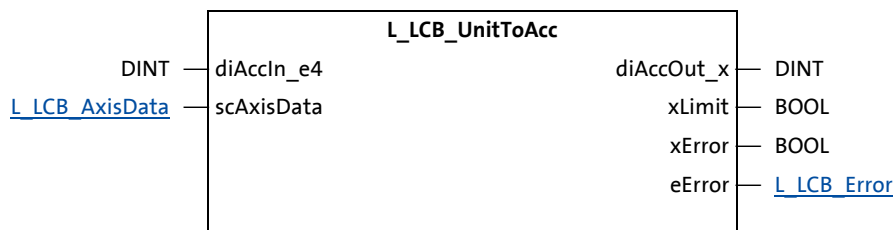
Bezeichner/Datentyp	Info/Einstellmöglichkeiten
diSpeedIn_s DINT	Geschwindigkeit als Drehzahl • Einheit: min ⁻¹ (15000 min ⁻¹ = 2 ²⁶ = 67108864)
scAxisData L_LCB_AxisData	Maschinenparameter

Ausgänge

Bezeichner/Datentyp	Wert/Bedeutung
diSpeedOut_e4 DINT	Geschwindigkeit in der realen Einheit der Maschine • Einheit: Units/s • Ausgabewert mit Fixkomma und vier Nachkommastellen
xLimit BOOL	TRUE Das Ausgangssignal wird auf den darstellbaren Zahlenbereich ±214748.3647 begrenzt.
xError BOOL	TRUE Es ist ein Fehler aufgetreten. Details siehe <i>eError</i> .
eError L_LCB_Error	Wenn <i>xError</i> = TRUE: Fehlernummer zum aufgetretenen Fehler. ► Fehlermeldungen (102)

8.2.26 L_LCB_UnitToAcc

Dieser Funktionsbaustein rechnet, anhand der übergebenen Maschinenparameter, einen Beschleunigungswert in der realen Einheit der Maschine um die interne Einheit [min^{-1}].



Eingänge

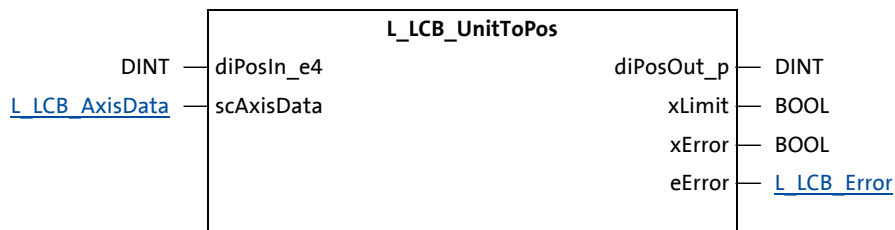
Bezeichner/Datentyp	Info/Einstellmöglichkeiten
diAccIn_e4 DINT	Beschleunigung in der realen Einheit der Maschine • Einheit: Units/s^2 • Vorgabewert mit Fixkomma und vier Nachkommastellen
scAxisData L_LCB_AxisData	Maschinenparameter

Ausgänge

Bezeichner/Datentyp	Wert/Bedeutung
diAccOut_x DINT	Beschleunigung als Drehzahländerung/Zeit • Einheit: min^{-1}/s ($15000 \text{ min}^{-1}/\text{s} \approx 2^{22} = 4194304$)
xLimit BOOL	TRUE Das Ausgangssignal wird auf den darstellbaren Zahlenbereich ± 214748.3647 begrenzt.
xError BOOL	TRUE Es ist ein Fehler aufgetreten. Details siehe <i>eError</i> .
eError L_LCB_Error	Wenn <i>xError</i> = TRUE: Fehlernummer zum aufgetretenen Fehler. ► Fehlermeldungen (102)

8.2.27 L_LCB_UnitToPos

Dieser Funktionsbaustein rechnet, anhand der übergebenen Maschinenparameter, einen Positionswert in der realen Einheit der Maschine um in Inkremente.



Eingänge

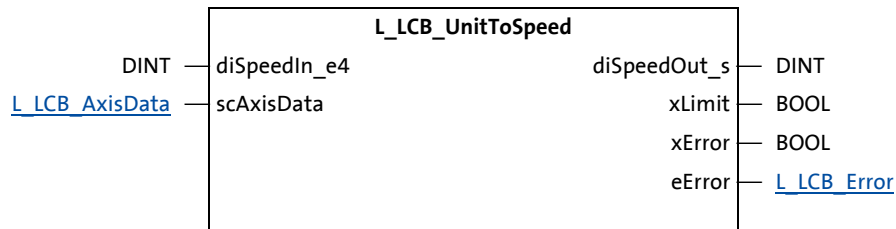
Bezeichner/Datentyp	Info/Einstellmöglichkeiten
diPosIn_e4 DINT	Position in der realen Einheit der Maschine • Einheit: Units • Vorgabewert mit Fixkomma und vier Nachkommastellen
scAxisData L_LCB_AxisData	Maschinenparameter

Ausgänge

Bezeichner/Datentyp	Wert/Bedeutung
diPosOut_p DINT	Position • Einheit: Inkremente
xLimit BOOL	TRUE Das Ausgangssignal wird auf den darstellbaren Zahlenbereich ± 214748.3647 begrenzt.
xError BOOL	TRUE Es ist ein Fehler aufgetreten. Details siehe <i>eError</i> .
eError L_LCB_Error	Wenn <i>xError</i> = TRUE: Fehlernummer zum aufgetretenen Fehler. ► Fehlermeldungen (102)

8.2.28 L_LCB_UnitToSpeed

Dieser Funktionsbaustein rechnet, anhand der übergebenen Maschinenparameter, einen Geschwindigkeitswert in der realen Einheit der Maschine um die interne Einheit [min⁻¹].



Eingänge

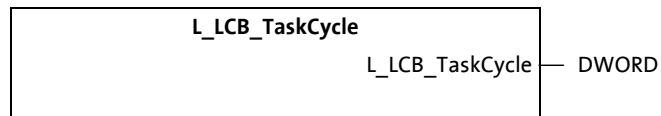
Bezeichner/Datentyp	Info/Einstellmöglichkeiten
diSpeedIn_e4 DINT	Geschwindigkeit in der realen Einheit der Maschine • Einheit: Units/s • Vorgabewert mit Fixkomma und vier Nachkommastellen
scAxisData L_LCB_AxisData	Maschinenparameter

Ausgänge

Bezeichner/Datentyp	Wert/Bedeutung
diSpeedOut_s DINT	Geschwindigkeit als Drehzahl • Einheit: min⁻¹ (15000 min⁻¹ = 2²⁶ = 67108864)
xLimit BOOL	TRUE Das Ausgangssignal wird auf den darstellbaren Zahlenbereich begrenzt.
xError BOOL	TRUE Es ist ein Fehler aufgetreten. Details siehe <i>eError</i> .
eError L_LCB_Error	Wenn <i>xError</i> = TRUE: Fehlernummer zum aufgetretenen Fehler. ► Fehlermeldungen (102)

8.2.29 L_LCB_TaskCycle

Diese Funktion ermittelt die Zykluszeit der Task.

**Ausgangssignal**

Bezeichner/Datentyp	Wert/Bedeutung
L_LCB_TaskCycle DWORD	Task-Zykluszeit • Einheit: ms

8.3 Strukturen (Structs)

8.3.1 L_LCB_AxisData

Bezeichner/Datentyp	Wert/Bedeutung
dwAxisID DWORD	Achsenzuordnung
rFeedConstant REAL	Vorschubkonstante
rInvFeedConstant REAL	Inverse Vorschubkonstante
rCycleLength REAL	Taktlänge (nur Modulo-Betrieb)
dwGearNumerator DWORD	Getriebefaktor-Zähler
dwGearDenominator DWORD	Getriebefaktor-Nenner
rMaxSpeed REAL	Maximale Geschwindigkeit
rReferenceSpeed REAL	Bezugsgeschwindigkeit
diReferenceSpeed_s DINT	Bezugsgeschwindigkeit
sAxisName STRING[16]	Achsenname
eMode L_LCB_AXISMODE	Fahrbereich
	0 Unbegrenzt
	1 Begrenzt
	2 Modulo
dwResolution DWORD	Geberauflösung
rEncoderGear REAL	Auflösung einer Geberumdrehung
rInvEncoderGear REAL	Invertierte Auflösung einer Geberumdrehung
diCycleLength_p DINT	Taktlänge
rUnitsToPos REAL	Interner Berechnungswert
rPosToUnits REAL	Interner Berechnungswert
rUnitsToSpeed REAL	Interner Berechnungswert
rSpeedToUnits REAL	Interner Berechnungswert
rTimeUnit REAL	Interner Berechnungswert

8.4 Enumerationen (Enums)

8.4.1 L_LCB_AXISMODE

Der Typ **L_LCB_AXISMODE** beschreibt den Fahrbereich des Inverters in den Maschinenparametern.

Wert	Bezeichner	Beschreibung
0	L_LCB_AM_UNLIMITED	Fahrbereich unbegrenzt
1	L_LCB_AM_LIMITED	Fahrbereich begrenzt
2	L_LCB_AM_MODULO	Fahrbereich modulo

8.4.2 L_LCB_Error

Der Typ **L_LCB_Error** beschreibt die möglichen Fehlermeldungen, die am Ausgang *eError* der Funktionsbausteine der Bibliothek **L_LCB_LogicControlBasic** generiert werden können.

Fehlermeldungen

Fehlernummer (dec)	Bezeichner	Beschreibung
0	L_LCB_NO_ERROR	Kein Fehler
20101	L_LCB_AXIS_IS_NO_AXIS_REF	Die angeschlossene Axis_REF -Instanz ist ungültig.
20102	L_LCB_CON_DRIVE_INVALID	Ungültige Antriebskennung vom Typ L_LCB_LOGICDEVICE
20110	L_LCB_AXISDATA_INVALID	Die verbundenen Maschinenparameter sind ungültig.

8.4.3 L_LCB_LOGICDEVICE

Der Typ **L_LCB_LOGICDEVICE** beschreibt die Kennung von verbundenen Logic-Geräten.

Wert	Bezeichner	Beschreibung
0	L_LCB_DeviceDefault	Unbekanntes Gerät
110	L_LCB_Device84_BL	ID für ein Inverter Drive 8400 BaseLine
111	L_LCB_Device84_SL	ID für ein Inverter Drive 8400 Stateline
112	L_LCB_Device84_HL	ID für ein Inverter Drive 8400 HighLine
120	L_LCB_Device94_HL	ID für ein Servo Drive 9400 HighLine

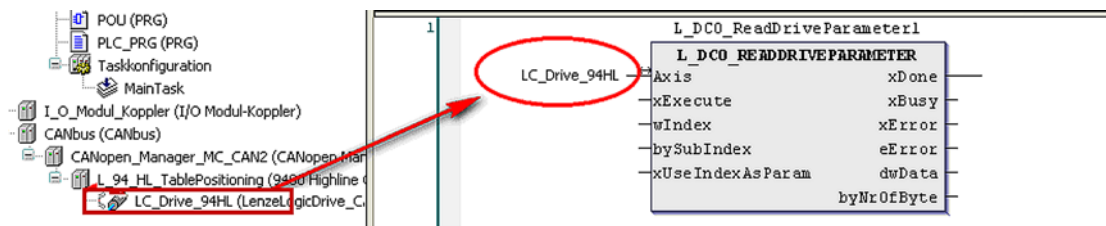
9 Funktionsbibliothek L_DCO_DriveCommunication

Die Bibliothek **L_DCO_DriveCommunication** enthält Funktionsbausteine zur busunabhängigen Parameterkommunikation.

Die Funktionsbausteine können mit Logic-Achsen der Inverter-Reihen 8400 und 9400 sowie in FAST Motion-Anwendungen mit Invertern der i-Reihe (z. B. i500, i700) verwendet werden.

Funktionsbaustein an einen Inverter anbinden

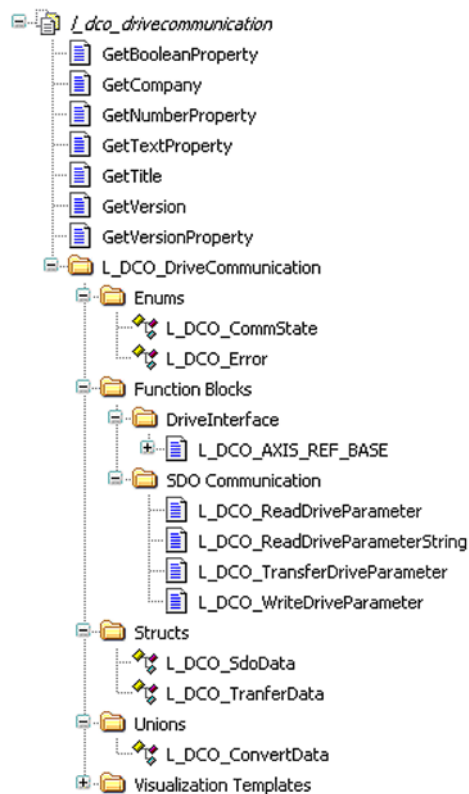
Zur Anbindung eines Funktionsbausteins an einen Inverter muss die Achsdaten-Instanz aus der Steuerungskonfiguration mit dem Funktionsbaustein verbunden werden.



[9-1] **Beispielverschaltung:** Achsdaten-Instanz eines 9400 HighLine mit dem Funktionsbaustein **L_DCO_ReadDriveParameter**

9.1 Übersicht der Funktionen und Funktionsbausteine

Die Funktionen und Funktionsbausteine der Bibliothek **L_DCO_DriveCommunication** sind in verschiedene Gruppen unterteilt.



Automatisch generierte Funktionen (Projektinformationen)

- [GetBooleanProperty](#) (132)
- [GetCompany](#) (133)
- [GetNumberProperty](#) (133)
- [GetTextProperty](#) (134)
- [GetTitle](#) (134)
- [GetVersion](#) (135)
- [GetVersionProperty](#) (135)

Enums

- [L_DCO_CommState](#) (115)
- [L_DCO_Error](#) (116)

Function Blocks

DriveInterface

- [L_DCO_AXIS_REF_BASE](#) (105)

SDO Communication

- [L_DCO_ReadDriveParameter](#) (106)
- [L_DCO_ReadDriveParameterString](#) (108)
- [L_DCO_TransferDriveParameter](#) (110)
- [L_DCO_WriteDriveParameter](#) (112)

Structs

- [L_DCO_SDOData](#) (114)
- [L_DCO_TransferData](#) (114)

Unions

- [L_DCO_ConvertData](#) (119)

9.2 Funktionsbausteine

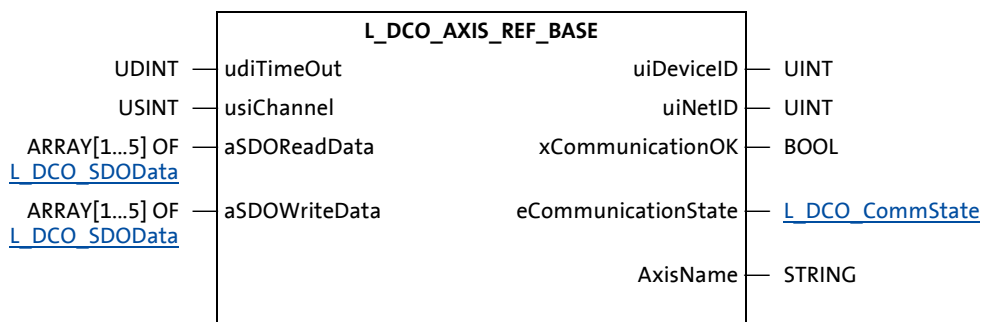
9.2.1 L_DCO_AXIS_REF_BASE

Dieser Funktionsbaustein stellt die Basis-Kommunikationsinformationen der Logic-Achse bereit.

Der Funktionsbaustein kann mit Logic-Achsen der Inverter-Reihen 8400 und 9400 sowie in FAST Motion-Anwendungen mit Invertern der i-Reihe (z. B. i500, i700) verwendet werden.

Jede in der Steuerungskonfiguration eingebundene Logic-Achse ist eine erweiterte Instanz von **L_DCO_AXIS_REF_BASE**. Eine manuelle Instanziierung dieses Funktionsbausteins ist nicht erforderlich.

Der Funktionsbaustein besitzt Routinen, die als Treiber dienen.



Eingänge

Bezeichner/Datentyp	Info/Einstellmöglichkeiten
udiTimeOut UDINT	Timeout-Zeit für die SDO-Kommunikation • Initialwert: 10000
usiChannel USINT	Zu verwendender SDO-Kanal • Initialwert: 1
aSdoReadData ARRAY[1...5] OF L_DCO_SDOData	Austausch für die SDO-Kommunikation (interne Verwendung) • Initialwert: 0
aSdoWriteData ARRAY[1...5] OF L_DCO_SDOData	Austausch für die SDO-Kommunikation (interne Verwendung) • Initialwert: 0

Ausgänge

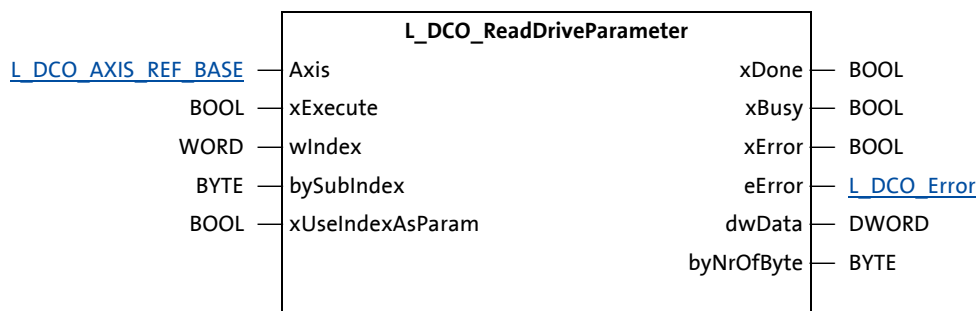
Bezeichner/Datentyp	Wert/Bedeutung
uiDeviceId UINT	Stations-/Knotenadresse am verwendeten Bussystem • Initialwert: 0
uiNetId UINT	Netzwerk/Master-Instanznummer des verwendeten Bussystems. • Initialwert: 0
xCommunicationOK BOOL	TRUE Buskommunikation vorhanden
	FALSE Keine Buskommunikation aktiv
eCommunicationState L_DCO_CommState	Buszustand • Initialwert: 0
AxisName STRING	Instanzname

9.2.2 L_DCO_ReadDriveParameter

Dieser Funktionsbaustein liest die Parameter aus einem Lenze-Inverter aus und stellt diese der Applikation zur Verfügung.

- Maximal sind 32 Bit große Parameterwerte lesbar.
- Der zu lesende Parameter kann als Index oder Codestellen-Nummer vorgegeben werden.
- Der Funktionsbaustein ist vom verwendeten Bussystem unabhängig.

Der Funktionsbaustein kann mit Logic-Achsen der Inverter-Reihen 8400 und 9400 sowie in FAST Motion-Anwendungen mit Invertern der i-Reihe (z. B. i500, i700) verwendet werden.



Eingänge

Bezeichner/Datentyp	Info/Einstellmöglichkeiten	
Axis L_DCO_AXIS_REF_BASE	Anschluss der Achsdaten der Logic-Achse	
xExecute BOOL	Aktivierung der Funktion "Parameter lesen"	
	FALSE ∇ TRUE	Parameter-Leseauftrag absetzen
	FALSE	Ausgänge werden zurückgesetzt. • Nach Beendigung des Auftrages werden die Ausgänge zurückgesetzt. • Statussignale werden mindestens für die Dauer eines Zykluses gesetzt.
wIndex WORD	Index des zu lesenden Parameters • Bei <i>xUseIndexAsParam</i> = TRUE: Angabe der Parameter-/Codestellennummer	
bySubIndex BYTE	Subindex des zu lesenden Parameters	
xUseIndexAsParam BOOL	TRUE	Vorgabe der Indizes gemäß Lenze-Parametern • Der an <i>wIndex</i> angegebene Wert wird als Parameter-/Codestellennummer interpretiert.
	FALSE	Vorgabe der Indizes gemäß CANopen

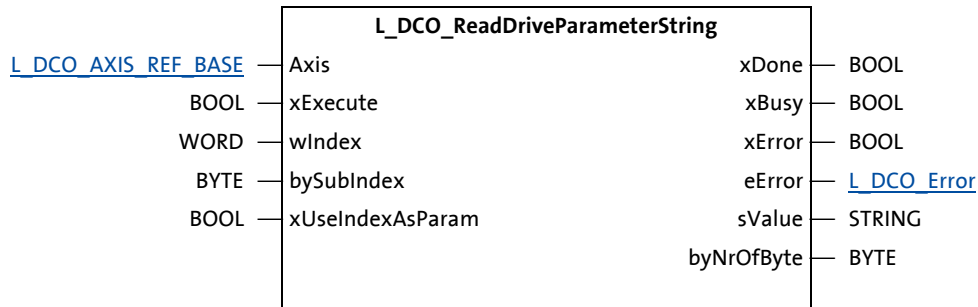
Ausgänge

Bezeichner/Datentyp	Wert/Bedeutung
xDone BOOL	TRUE Leseauftrag beendet
	FALSE Leseauftrag wird aktuell ausgeführt
xBusy BOOL	TRUE Der Leseauftrag wurde an das System übermittelt. Der Funktionsbaustein wartet auf die Antwort des Auftrages.
	FALSE Kein Leseauftrag aktiv
xError BOOL	TRUE Es ist ein Fehler aufgetreten. Details siehe <i>eError</i> .
eError L_DCO_Error	Wenn <i>xError</i> = TRUE: Fehlernummer zum aufgetretenen Fehler. ► Fehlermeldungen (116)
dwData DWORD	Gelesener Parameterwert
byNrOfByte BYTE	Länge des gelesenen Wertes

9.2.3 L_DCO_ReadDriveParameterString

Dieser Funktionsbaustein liest einen Parameter vom Datentyp "STRING" eines Lenze-Inverters, welcher über ein Bussystem mit dem Controller verbunden ist und stellt diesen der Applikation zur Verfügung.

Der Funktionsbaustein kann mit Logic-Achsen der Inverter-Reihen 8400 und 9400 sowie in FAST Motion-Anwendungen mit Invertern der i-Reihe (z. B. i500, i700) verwendet werden.



Eingänge

Bezeichner/Datentyp	Info/Einstellmöglichkeiten	
Axis L_DCO_AXIS_REF_BASE	Anschluss der Achsdaten der Logic-Achse	
xExecute BOOL	Aktivierung der Funktion "Parameter lesen"	
	FALSE → TRUE	Parameter-Leseauftrag absetzen
	FALSE	Ausgänge werden zurückgesetzt. • Nach Beendigung des Auftrages werden die Ausgänge zurückgesetzt. • Statussignale werden mindestens für die Dauer eines Zykluses gesetzt.
wIndex WORD	Index des zu lesenden Parameters • Bei <i>xUseIndexAsParam</i> = TRUE: Angabe der Parameter-/Codestellennummer	
bySubIndex BYTE	Subindex des zu lesenden Parameters	
xUseIndexAsParam BOOL	TRUE	Vorgabe der Indizes gemäß Lenze-Parametern • Der an <i>wIndex</i> angegebene Wert wird als Parameter-/Codestellennummer interpretiert.
	FALSE	Vorgabe der Indizes gemäß CANopen

Ausgänge

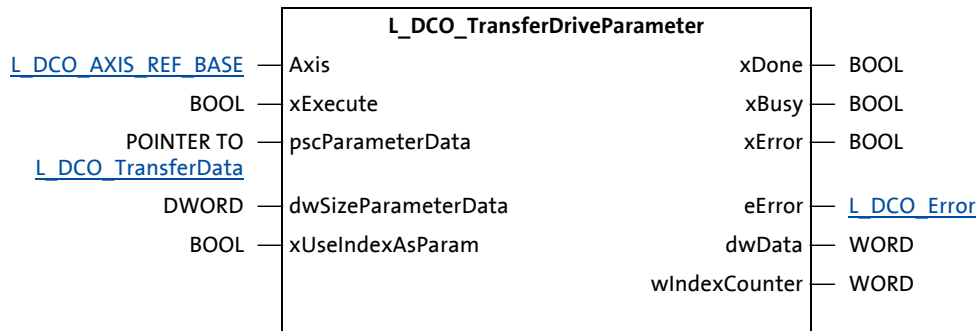
Bezeichner/Datentyp	Wert/Bedeutung
xDone BOOL	TRUE Leseauftrag beendet
	FALSE Leseauftrag wird aktuell ausgeführt
xBusy BOOL	TRUE Der Leseauftrag wurde an das System übermittelt. Der Funktionsbaustein wartet auf die Antwort des Auftrages.
	FALSE Kein Leseauftrag aktiv
xError BOOL	TRUE Es ist ein Fehler aufgetreten. Details siehe <i>eError</i> .
eError L_DCO_Error	Wenn <i>xError</i> = TRUE: Fehlernummer zum aufgetretenen Fehler. ► Fehlermeldungen (116)
dwData DWORD	Gelesener Parameterwert
byNrOfByte BYTE	Länge des gelesenen Wertes

9.2.4 L_DCO_TransferDriveParameter

Dieser Funktionsbaustein überträgt Parameter eines Lenze-Inverters, der über ein Bussystem mit dem Controller verbunden ist.

Die zu übertragenden Parameter werden über die Array-Struktur "ARRAY[1...n] OF L_DCO_TransferData" vorgegeben (siehe Programmbeispiel [\[9-2\]](#) ([111](#))).

Der Funktionsbaustein kann mit Logic-Achsen der Inverter-Reihen 8400 und 9400 sowie in FAST Motion-Anwendungen mit Invertern der i-Reihe (z. B. i500, i700) verwendet werden.



Eingänge

Bezeichner/Datentyp	Info/Einstellmöglichkeiten	
Axis L_DCO_AXIS_REF_BASE	Anschluss der Achsdaten der Logic-Achse	
xExecute BOOL	Aktivierung der Funktion "Parameter übertragen"	
	FALSE ∇ TRUE	Parameternauftrag absetzen
	FALSE	Ausgänge werden zurückgesetzt. • Nach Beendigung des Auftrages werden die Ausgänge zurückgesetzt. • Statussignale werden mindestens für die Dauer eines Zykluses gesetzt.
pscParameterData POINTER TO L_DCO_TransferData	Hier wird die Adresse der Array-Struktur vom Typ L_DCO_TransferData (114) übergeben	
dwSizeParameterData DWORD	Größe der zu übertragenden Array-Struktur, die die Codestellen beinhaltet.	
xUseIndexAsParam BOOL	TRUE	Vorgabe der Indizes gemäß Lenze-Parametern • Der an <i>wIndex</i> angegebene Wert wird als Parameter-/Codestellennummer interpretiert.
	FALSE	Vorgabe der Indizes gemäß CANopen

Ausgänge

Bezeichner/Datentyp	Wert/Bedeutung
xDone BOOL	TRUE Parameterauftrag beendet
	FALSE Parameterauftrag wird aktuell ausgeführt
xBusy BOOL	TRUE Der Parameterauftrag wurde an das System übermittelt. Der Funktionsbaustein wartet auf die Antwort des Auftrages.
	FALSE Kein Parameterauftrag aktiv
xError BOOL	TRUE Es ist ein Fehler aufgetreten. Details siehe <i>eError</i> .
eError L_DCO_Error	Wenn <i>xError</i> = TRUE: Fehlernummer zum aufgetretenen Fehler. ► Fehlermeldungen (116)
dwData DWORD	Gelesener Parameterwert
wIndexCounter WORD	Anzahl der abgearbeiteten Parameteraufträge • Im Fehlerfall wird hier die Nummer des Auftrags angegeben, der den Übertragungsfehler verursacht hat.

```

PROGRAM Demo
VAR
L_DCO_TransferDriveParameter1 : L_DCO_TransferDriveParameter;
scTransferData : ARRAY[1..5] OF L_DCO_TransferData

:= [
(wIndex := 11, bySubIndex := 0, dwData := 0, byNrOfByte := 4, xEnable := TRUE,
xWriteAccess:= FALSE ),
(wIndex := 11, bySubIndex := 0, dwData := 0, byNrOfByte := 4, xEnable := TRUE,
xWriteAccess:= FALSE ),
(wIndex := 11, bySubIndex := 0, dwData := 0, byNrOfByte := 4, xEnable := TRUE,
xWriteAccess:= FALSE ),
(wIndex := 11, bySubIndex := 0, dwData := 0, byNrOfByte := 4, xEnable := TRUE,
xWriteAccess:= FALSE ),
(wIndex := 11, bySubIndex := 0, dwData := 0, byNrOfByte := 4, xEnable := TRUE,
xWriteAccess:= FALSE ) ];

END_VAR

...

// Aufruf der Instanz
L_DCO_TransferDriveParameter1.pscParameterData := ADR(scTransferData);
L_DCO_TransferDriveParameter1.dwSizeParameterData := SIZEOF(scTransferData);
L_DCO_TransferDriveParameter1(Axis:=LC_Drive_94HL);

```

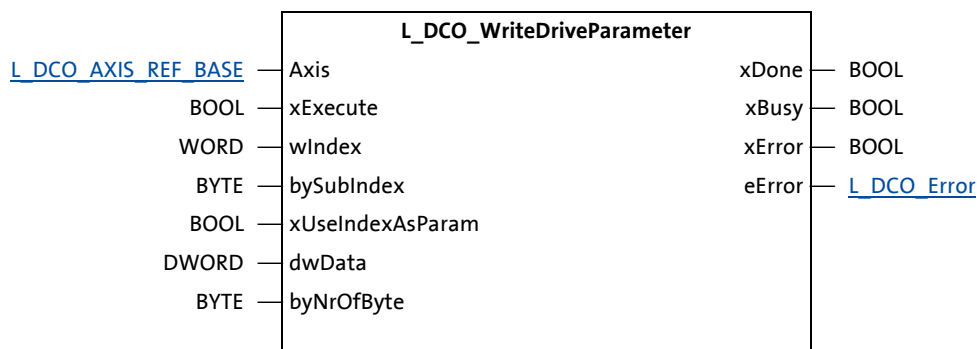
[9-2] **Programmbeispiel:** Verwendung des Funktionsbausteins **L_DCO_TransferDriveParameter**

9.2.5 L_DCO_WriteDriveParameter

Dieser Funktionsbaustein schreibt einen Parameter eines Lenze-Inverters, welcher über ein Bussystem mit dem Controller verbunden ist.

- Maximal sind 32 Bit große Parameterwerte schreibbar.
- Der zu schreibende Parameter kann als Index oder Codestellen-Nummer vorgegeben werden.
- Der Funktionsbaustein ist vom verwendeten Bussystem unabhängig.

Der Funktionsbaustein kann mit Logic-Achsen der Inverter-Reihen 8400 und 9400 sowie in FAST Motion-Anwendungen mit Invertern der i-Reihe (z. B. i500, i700) verwendet werden.



Eingänge

Bezeichner/Datentyp	Info/Einstellmöglichkeiten	
Axis <u>L_DCO_AXIS_REF_BASE</u>	Anschluss der Achsdaten der Logic-Achse	
xExecute BOOL	Aktivierung der Funktion "Parameter schreiben"	
	FALSE → TRUE	Parameter-Schreibauftrag absetzen
	FALSE	Ausgänge werden zurückgesetzt. • Nach Beendigung des Auftrages werden die Ausgänge zurückgesetzt. • Statussignale werden mindestens für die Dauer eines Zykluses gesetzt.
wIndex WORD	Index des zu schreibenden Parameters	
bySubIndex BYTE	Subindex des zu schreibenden Parameters	
xUseIndexAsParam BOOL	TRUE	Vorgabe der Indizes gemäß Lenze-Parametern • Der an <i>wIndex</i> angegebene Wert wird als Parameter-/Codestellennummer interpretiert.
	FALSE	Vorgabe der Indizes gemäß CANopen
dwData DWORD	Zu schreibender Parameterwert	
byNrOfByte BYTE	Anzahl der zu schreibenden Bytes	

Ausgänge

Bezeichner/Datentyp		Wert/Bedeutung
xDone	BOOL	TRUE Schreibauftrag beendet
		FALSE Schreibauftrag wird aktuell ausgeführt
xBusy	BOOL	TRUE Der Schreibeauftrag wurde an das System übermittelt. Der Funktionsbaustein wartet auf die Antwort des Auftrages.
		FALSE Kein Schreibeauftrag aktiv
xError	BOOL	TRUE Es ist ein Fehler aufgetreten. Details siehe <i>eError</i> .
eError	L_DCO_Error	Wenn <i>xError</i> = TRUE: Fehlernummer zum aufgetretenen Fehler. ▶ Fehlermeldungen (📖 116)

9.3 Strukturen (Structs)

9.3.1 L_DCO_SDOData

Die Struktur **L_DCO_SDOData** dient zum Austausch von Parameterschreibaufträgen und Parameterleseaufträgen.

Diese Struktur wird nur intern verwendet!

Bezeichner/Datentyp	Wert/Bedeutung
wParam WORD	Index-Nummer des zu bearbeitenden Parameters • Initialwert: 0
bySubIndex BYTE	Sub-Index des zu bearbeitenden Parameters • Initialwert: 0
wState WORD	Status des Parameterauftrags • Initialwert: 0
dwOwner DWORD	Kennung des Auftraggebers • Initialwert: 0
dwData DWORD	Übertragene/Gelesene Daten • Initialwert: 0
psData POINTER OF BYTE	Pointer auf ein Byte, in das die gelesenen Daten geschrieben werden. • Initialwert: 0
uDataLength UNIT	Datenlänge des Parameters • Initialwert: 0
wError WORD	Fehlernummer • Initialwert: 0

9.3.2 L_DCO_TransferData

Die Struktur **L_DCO_TransferData** beschreibt einen Auftrag zur Übertragung eines Parameters.

Mit Hilfe des Funktionsbausteins [L_DCO_TransferDriveParameter](#) (110) können über ein "ARRAY[1...n] OF L_DCO_TransferData" mehrere Parameter von einem Antrieb übertragen werden.

Bezeichner/Datentyp	Beschreibung
wIndex WORD	Index-Nummer des zu übertragenden Parameters • Initialwert: 0
bySubIndex BYTE	Sub-Index des zu übertragenden Parameters • Initialwert: 0
xWriteAccess BOOL	Schreib- oder Lesezugriff • Initialwert: FALSE
	TRUE Schreibzugriff
	FALSE Lesezugriff
xEnable BOOL	Aktivierung des Parameterauftrags • Initialwert: FALSE
	TRUE Parameterauftrag wird abgearbeitet
	FALSE Parameterauftrag wird ignoriert
byNrOfByte BYTE	Anzahl der übertragenen/gelesenen Bytes • Initialwert: 0
dwData DWORD	Übertragene/Gelesene Daten • Initialwert: 0

9.4 Enumerationen (Enums)

9.4.1 L_DCO_CommState

Der Typ **L_DCO_CommState** beschreibt die möglichen Zustände, die die Feldbusteilnehmer (Master und Slaves) annehmen können.

Der Kommunikationszustand ist über eine Instanz des Funktionsbausteins [L_DCO_AXIS_REF_BASE](#) (105) und dessen Erweiterungen verfügbar.

Die jeweilige Funktionsbaustein-Instanz wird im Geräte-Editor bereitgestellt.

Wert	Bezeichner	Beschreibung
0	L_DCO_STATE_BOOT	Zustand "Boot-Vorgang"
10	L_DCO_STATE_INIT	Zustand "Initialisierung"
20	L_DCO_STATE_RESET_COMM	Zustand "Reset der Kommunikation"
30	L_DCO_STATE_RESET_APP	Zustand "Reset"
40	L_DCO_STATE_PREOPERATIONAL	Zustand "Preoperational"
50	L_DCO_STATE_STOPPED	Zustand "Stopped"
70	L_DCO_STATE_OPERATIONAL	Zustand "Operational"
80	L_DCO_STATE_UNKNOWN	Zustand "unbekannt"
90	L_DCO_STATE_NOT_AVAIL	Zustand "nicht vorhanden"

9.4.2 L_DCO_Error

Der Typ **L_DCO_Error** beschreibt die möglichen Fehlermeldungen, die am Ausgang *eError* der Funktionsbausteine der Bibliothek **L_DCO_DriveCommunication** generiert werden können.

Fehlermeldungen

Fehlernummer (dec)	Bezeichner	Beschreibung
0	L_DCO_No_Error	Kein Fehler gemeldet
20001	L_DCO_OTHERERROR	Allgemeiner Fehler
20002	L_DCO_INVALIDINDEX	Ungültiger SDO-Index
20003	L_DCO_INVALIDOFFSET	Ungültiger Offset-Wert beim Zugriff auf das Prozessdaten-Abbild
20004	L_DCO_INVALIDSIZE	Ungültiger Längenwert ... • beim Zugriff auf das Prozessdaten-Abbild • beim Speichern von Daten in einen Datenbereich
20005	L_DCO_INVALIDDATA	Ungültige Daten
20006	L_DCO_NOTREADY	Interner Software-Fehler • Ursache nicht gemeldet
20007	L_DCO_BUSY	Der Master ist zur Zeit beschäftigt und kann die API-Funktion nicht bearbeiten. • Funktion zu einem späteren Zeitpunkt erneut aufrufen.
20008	L_DCO_NOMEMORY	Zu wenig Anwendungsspeicher vorhanden.
20009	L_DCO_INVALIDPARM	Fehlerhafter Aufruf einer API-Funktion durch ungültige Parameter.
20010	L_DCO_INVALIDSTATE	Ungültiger Zustand
20011	L_DCO_TIMEOUT	Zeitüberschreitung (Timeout)
20012	L_DCO_INVALID_SLAVE_STATE	Unzulässige Mailbox-Befehle beim aktuellen Slave-Zustand.
20013	L_DCO_LINK_DISCONNECTED	EtherCAT-Kabel ist nicht mit der Steuerung verbunden.
20015	L_DCO_ALL_CHANNELS_BUSY	Alle Kanäle zur Verarbeitung der Parameteranfrage belegt.
20016	L_DCO_DATA_REQUEST_ERROR	Allgemeiner Fehler bei Anfrage gemeldet
20017	L_DCO_BUS_OFF	Bus nicht initialisiert
20018	L_DCO_SLAVE_ERROR	Slave nicht adressierbar
20020	L_DCO_SDO_ABORTCODE_TOGGLE	Unveränderter Zustand des Toggle-Bits • Abort Code 0x05030000
20021	L_DCO_SDO_ABORTCODE_TIMEOUT	Zeitüberschreitung (Timeout) des SDO-Protokolls • Abort Code 0x05040000
20022	L_DCO_SDO_ABORTCODE_CCS_SCS	Ungültiges oder unbekanntes Spezifikationsymbol für den Client/Server-Befehl • Abort Code 0x05040001
20023	L_DCO_SDO_ABORTCODE_BLK_SIZE	Ungültige Blockgröße (nur im "Blockmode") • Abort Code 0x05040002
20024	L_DCO_SDO_ABORTCODE_SEQNO	Ungültige Sequenznummer (nur im "Block mode") • Abort Code 0x05040003
20025	L_DCO_SDO_ABORTCODE_CRC	CRC-Fehler (nur im "Block mode") • Abort Code 0x05040004
20026	L_DCO_SDO_ABORTCODE_MEMORY	Zu wenig Speicherplatz im Hauptspeicher frei • Abort Code 0x05040005

Fehlernummer (dec)	Bezeichner	Beschreibung
20027	L_DCO_SDO_ABORTCODE_ACCESS	Zugriff auf ein Objekt abgelehnt • Abort Code 0x06010000
20028	L_DCO_SDO_ABORTCODE_WRITEONLY	Lesezugriff auf ein schreibgeschütztes Objekt • Abort Code 0x06010001
20029	L_DCO_SDO_ABORTCODE_READONLY	Schreibzugriff auf ein schreibgeschütztes Objekt • Abort Code 0x06010002
20030	L_DCO_SDO_ABORTCODE_INDEX	Ein Objekt im Objektverzeichnis nicht gefunden. • Abort Code 0x06020000
20031	L_DCO_SDO_ABORTCODE_PDO_MAP	Ein Objekt kann nicht ins PDO gemappt werden. • Abort Code 0x06040041
20032	L_DCO_SDO_ABORTCODE_PDO_LEN	Anzahl/Länge der gemappten Objekte überschreitet die PDO-Länge. • Abort Code 0x06040042
20033	L_DCO_SDO_ABORTCODE_P_INCOMP	Allgemeine Meldung zur Inkompatibilität der Parameter • Abort Code 0x06040043
20034	L_DCO_SDO_ABORTCODE_I_INCOMP	Allgemeine Meldung zur Inkompatibilität der Parameter • Abort Code 0x06040043
20035	L_DCO_SDO_ABORTCODE_HARDWARE	Fehlgeschlagener Zugriff durch Hardwarefehler • Abort Code 0x06060000
20036	L_DCO_SDO_ABORTCODE_DATA_SIZE	Datentyp oder Parameterlänge sind unterschiedlich • Abort Code 0x06070010
20037	L_DCO_SDO_ABORTCODE_DATA_SIZE1	Falscher Datentyp (Die Parameterlänge ist zu groß.) • Abort Code 0x06070012
20038	L_DCO_SDO_ABORTCODE_DATA_SIZE2	Falscher Datentyp (Die Parameterlänge ist zu klein.) • Abort Code 0x06070013
20039	L_DCO_SDO_ABORTCODE_OFFSET	Fehlernder Subindex • Abort Code 0x06090011
20040	L_DCO_SDO_ABORTCODE_DATA_RANGE	Der Wertebereich für Parameter ist zu groß (nur bei Schreibzugriff). • Abort Code 0x06090030
20041	L_DCO_SDO_ABORTCODE_DATA_RANGE1	Der Parameterwert ist zu hoch. • Abort Code 0x06090031
20042	L_DCO_SDO_ABORTCODE_DATA_RANGE2	Der Parameterwert ist zu niedrig. • Abort Code 0x06090032
20043	L_DCO_SDO_ABORTCODE_MINMAX	Der Maximalwert ist kleiner als der Minimalwert. • Abort Code 0x06090036
20044	L_DCO_SDO_ABORTCODE_GENERAL	Allgemeiner Fehler • Abort Code 0x08000000
20045	L_DCO_SDO_ABORTCODE_TRANSFER	Fehler beim Speichern/Übertragen der Daten in die Anwendung • Abort Code 0x08000020
20046	L_DCO_SDO_ABORTCODE_TRANSFER1	Fehler beim Speichern/Übertragen der Daten in die Anwendung (verursacht durch lokale Steuerung) • Abort Code 0x08000021
20047	L_DCO_SDO_ABORTCODE_TRANSFER2	Fehler beim Speichern/Übertragen der Daten in die Anwendung (verursacht durch aktuellen Gerätezustand) • Abort Code 0x08000022
20048	L_DCO_SDO_ABORTCODE_DICTIONARY	Dynamische Objektverzeichnisgenerierung ist fehlgeschlagen/Kein Objektverzeichnis verfügbar • Abort Code 0x08000023

Fehlernummer (dec)	Bezeichner	Beschreibung
20049	L_DCO_SDO_ABORTCODE _UNKNOWN	Interner Fehler des Slaves (Ursache unbekannt)
20060	L_DCO_TRANSFERTABLE_INVALID	Ungültige Struktur L_DCO_TransferData (114) an Funktionsbaustein L_DCO_TransferDriveParameter (110) übergeben.

9.5

Unions

9.5.1

L_DCO_ConvertData

Stellt unterschiedliche Zugriffsmöglichkeiten auf einen 32-Bit-Speicherbereich bereit.

Bezeichner/Datentyp		Beschreibung
abData	ARRAY OF BYTE	Übertragene/Gelesene Daten
awData	ARRAY OF WORD	Übertragene/Gelesene Daten
dwData	DWORD	Übertragene/Gelesene Daten

10

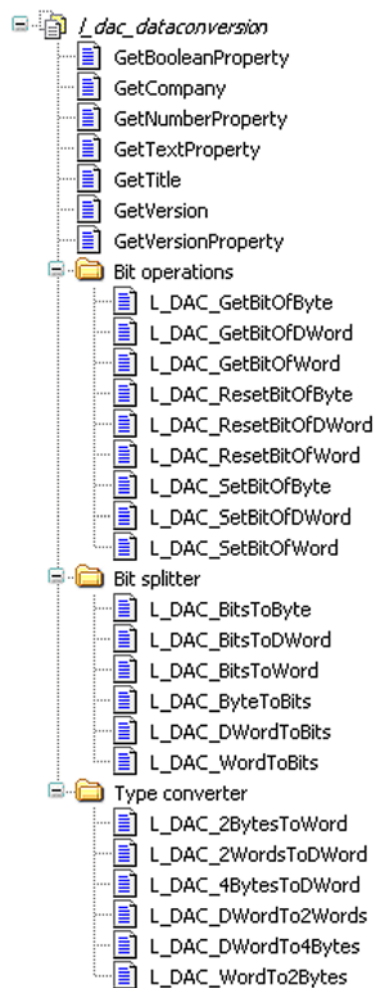
Funktionsbibliothek L_DAC_DataConversion

Die Bibliothek **L_DAC_DataConversion** enthält Funktionen und Funktionsbausteine zur Umrechnung von Daten.

10.1

Übersicht der Funktionen und Funktionsbausteine

Die Funktionen und Funktionsbausteine der Bibliothek **L_DAC_DataConversion** sind in verschiedene Gruppen unterteilt.



Automatisch generierte Funktionen (Projektinformationen)

- ▶ [GetBooleanProperty](#) (132)
- ▶ [GetCompany](#) (133)
- ▶ [GetNumberProperty](#) (133)
- ▶ [GetTextProperty](#) (134)
- ▶ [GetTitle](#) (134)
- ▶ [GetVersion](#) (135)
- ▶ [GetVersionProperty](#) (135)

Bit operations

- ▶ [L_DAC_GetBitOfByte](#) (121)
- ▶ [L_DAC_GetBitOfDWord](#) (121)
- ▶ [L_DAC_GetBitOfWord](#) (122)
- ▶ [L_DAC_ResetBitOfByte](#) (122)
- ▶ [L_DAC_ResetBitOfDWord](#) (123)
- ▶ [L_DAC_ResetBitOfWord](#) (123)
- ▶ [L_DAC_SetBitOfByte](#) (124)
- ▶ [L_DAC_SetBitOfDWord](#) (124)
- ▶ [L_DAC_SetBitOfWord](#) (125)

Bit splitter

- ▶ [L_DAC_BitsToByte](#) (126)
- ▶ [L_DAC_BitsToDWord](#) (126)
- ▶ [L_DAC_BitsToWord](#) (127)
- ▶ [L_DAC_ByteToBits](#) (127)
- ▶ [L_DAC_DWordToBits](#) (128)
- ▶ [L_DAC_WordToBits](#) (128)

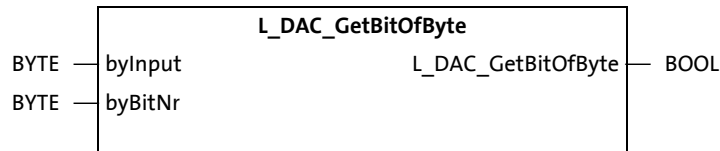
Type converter

- ▶ [L_DAC_2BytesToWord](#) (129)
- ▶ [L_DAC_2WordsToDWord](#) (129)
- ▶ [L_DAC_4BytesToDWord](#) (130)
- ▶ [L_DAC_DWordTo2Words](#) (130)
- ▶ [L_DAC_DWordTo4Bytes](#) (131)
- ▶ [L_DAC_WordTo2Bytes](#) (131)

10.2 Bit operations

10.2.1 L_DAC_GetBitOfByte

Diese Funktion liefert den Zustand ('TRUE' oder 'FALSE') eines einzelnen Bits innerhalb eines "BYTE"-Wertes zurück.



Eingänge

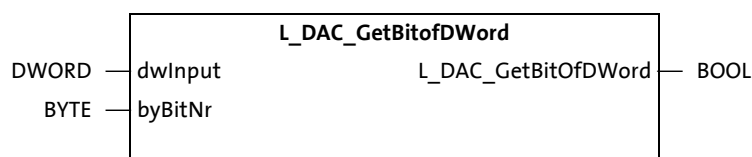
Bezeichner/Datentyp	Info/Einstellmöglichkeiten
byInput BYTE	Eingangssignal
byBitNr BYTE	Nummer (0 ... 7) des Bits von <i>byInput</i> , dessen Zustand ermittelt werden soll.

Ausgang

Bezeichner/Datentyp	Wert/Bedeutung
L_DAC_GetBitOfByte BOOL	Zustand ('TRUE' oder 'FALSE') des Bits <i>byBitNr</i> aus dem Eingangssignal <i>byInput</i> .

10.2.2 L_DAC_GetBitOfDWord

Diese Funktion liefert den Zustand ('TRUE' oder 'FALSE') eines einzelnen Bits innerhalb eines "DWORD"-Wertes zurück.



Eingänge

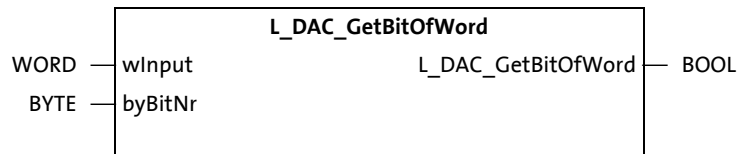
Bezeichner/Datentyp	Info/Einstellmöglichkeiten
dwInput DWORD	Eingangssignal
byBitNr BYTE	Nummer (0 ... 31) des Bits von <i>dwInput</i> , dessen Zustand ermittelt werden soll.

Ausgang

Bezeichner/Datentyp	Wert/Bedeutung
L_DAC_GetBitOfDWord BOOL	Zustand ('TRUE' oder 'FALSE') des Bits <i>byBitNr</i> aus dem Eingangssignal <i>dwInput</i> .

10.2.3 L_DAC_GetBitOfWord

Diese Funktion liefert den Zustand ('TRUE' oder 'FALSE') eines einzelnen Bits innerhalb eines "WORD"-Wertes zurück.



Eingänge

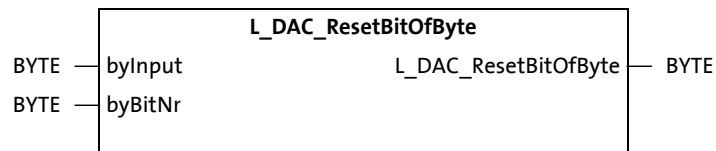
Bezeichner/Datentyp	Info/Einstellmöglichkeiten
wInput WORD	Eingangssignal
byBitNr BYTE	Nummer (0 ... 15) des Bits von dwInput, dessen Zustand ermittelt werden soll.

Ausgang

Bezeichner/Datentyp	Wert/Bedeutung
L_DAC_GetBitOfWord BOOL	Zustand ('TRUE' oder 'FALSE') des Bits <i>byBitNr</i> aus dem Eingangssignal <i>wInput</i> .

10.2.4 L_DAC_ResetBitOfByte

Diese Funktion setzt den Zustand eines einzelnen Bits innerhalb eines "BYTE"-Wertes auf 'FALSE'.



Eingänge

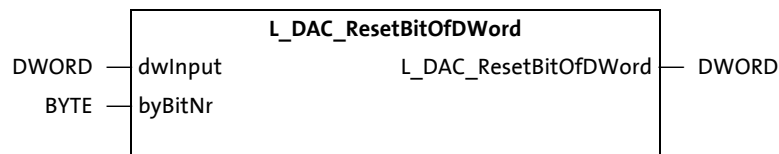
Bezeichner/Datentyp	Info/Einstellmöglichkeiten
byInput BYTE	Eingangssignal
byBitNr BYTE	Nummer (0 ... 7) des Bits, das auf 'FALSE' gesetzt werden soll.

Ausgang

Bezeichner/Datentyp	Wert/Bedeutung
L_DAC_ResetBitOfByte BYTE	Ausgangssignal • Der Wert ergibt sich durch das auf 'FALSE' gesetzte Bit <i>byBitNr</i> aus dem Eingangssignal <i>byInput</i> .

10.2.5 L_DAC_ResetBitOfDWord

Diese Funktion setzt den Zustand eines einzelnen Bits innerhalb eines "DWORD"-Wertes auf 'FALSE'.



Eingänge

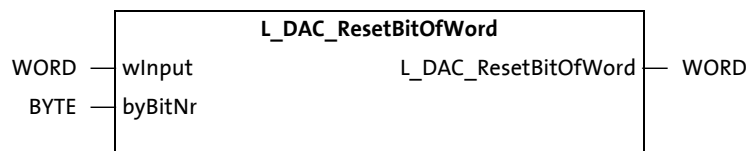
Bezeichner/Datentyp	Info/Einstellmöglichkeiten
dwInput DWORD	Eingangssignal
byBitNr BYTE	Nummer (0 ... 31) des Bits, das auf 'FALSE' gesetzt werden soll.

Ausgang

Bezeichner/Datentyp	Wert/Bedeutung
L_DAC_ResetBitOfDWord DWORD	Ausgangssignal • Der Wert ergibt sich durch das auf 'FALSE' gesetzte Bit <i>byBitNr</i> aus dem Eingangssignal <i>dwInput</i> .

10.2.6 L_DAC_ResetBitOfWord

Diese Funktion setzt den Zustand eines einzelnen Bits innerhalb eines "WORD"-Wertes auf 'FALSE'.



Eingänge

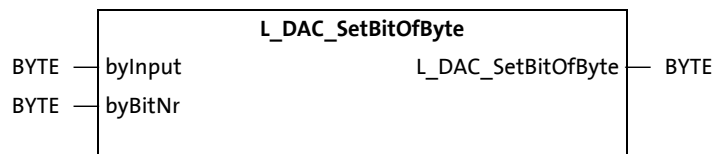
Bezeichner/Datentyp	Info/Einstellmöglichkeiten
wInput WORD	Eingangssignal
byBitNr BYTE	Nummer (0 ... 15) des Bits, das auf 'FALSE' gesetzt werden soll.

Ausgang

Bezeichner/Datentyp	Wert/Bedeutung
L_DAC_ResetBitOfWord WORD	Ausgangssignal • Der Wert ergibt sich durch das auf 'FALSE' gesetzte Bit <i>byBitNr</i> aus dem Eingangssignal <i>wInput</i> .

10.2.7 L_DAC_SetBitOfByte

Diese Funktion setzt den Zustand eines einzelnen Bits innerhalb eines "BYTE"-Wertes auf 'TRUE'.



Eingänge

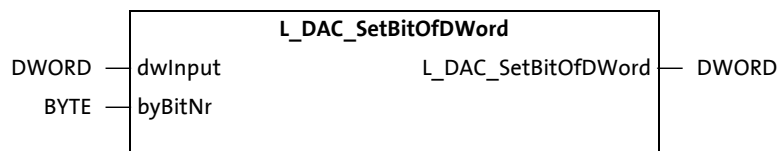
Bezeichner/Datentyp	Info/Einstellmöglichkeiten
byInput BYTE	Eingangssignal
byBitNr BYTE	Nummer (0 ... 7) des Bits, das auf 'TRUE' gesetzt werden soll.

Ausgang

Bezeichner/Datentyp	Wert/Bedeutung
L_DAC_SetBitOfByte BYTE	Ausgangssignal • Der Wert ergibt sich durch das auf 'TRUE' gesetzte Bit <i>byBitNr</i> aus dem Eingangssignal <i>byInput</i> .

10.2.8 L_DAC_SetBitOfDWord

Diese Funktion setzt den Zustand eines einzelnen Bits innerhalb eines "DWORD"-Wertes auf 'TRUE'.



Eingänge

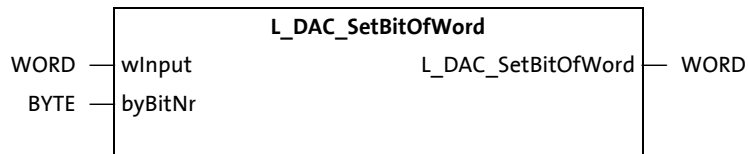
Bezeichner/Datentyp	Info/Einstellmöglichkeiten
dwInput DWORD	Eingangssignal
byBitNr BYTE	Nummer (0 ... 31) des Bits, das auf 'TRUE' gesetzt werden soll.

Ausgang

Bezeichner/Datentyp	Wert/Bedeutung
L_DAC_SetBitOfDWord BYTE	Ausgangssignal • Der Wert ergibt sich durch das auf 'TRUE' gesetzte Bit <i>byBitNr</i> aus dem Eingangssignal <i>dwInput</i> .

10.2.9 L_DAC_SetBitOfWord

Diese Funktion setzt den Zustand eines einzelnen Bits innerhalb eines "WORD"-Wertes auf 'TRUE'.



Eingänge

Bezeichner/Datentyp	Info/Einstellmöglichkeiten
wInput WORD	Eingangssignal
byBitNr BYTE	Nummer (0 ... 15) des Bits, das auf 'TRUE' gesetzt werden soll.

Ausgang

Bezeichner/Datentyp	Wert/Bedeutung
L_DAC_SetBitOfWord BYTE	Ausgangssignal Der Wert ergibt sich durch das auf 'TRUE' gesetzte Bit <i>byBitNr</i> aus dem Eingangssignal <i>wInput</i>.

10.3 Bit splitter

10.3.1 L_DAC_BitsToByte

Dieser Funktionsbaustein konvertiert 8 Eingangs-Bits in einen Wert vom Typ "BYTE".



Eingänge

Bezeichner/Datentyp	Info/Einstellmöglichkeiten
xBit[0...7] BOOL	Eingangssignal der Bits 0 ... 7

Ausgänge

Bezeichner/Datentyp	Wert/Bedeutung
byOutput BYTE	Ausgabewert, der sich durch die Eingangs-Bits <i>xBit[0...7]</i> ergibt.

10.3.2 L_DAC_BitsToDWord

Dieser Funktionsbaustein konvertiert 32 Eingangs-Bits in einen Wert vom Typ "DWORD".



Eingänge

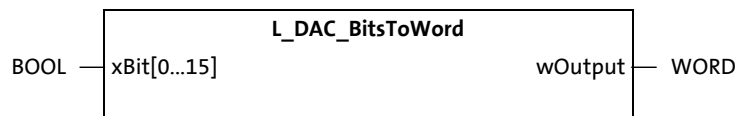
Bezeichner/Datentyp	Info/Einstellmöglichkeiten
xBit[0...31] BOOL	Eingangssignal der Bits 0 ... 31

Ausgänge

Bezeichner/Datentyp	Wert/Bedeutung
dwOutput DWORD	Ausgabewert, der sich durch die Eingangs-Bits <i>xBit[0...31]</i> ergibt.

10.3.3 L_DAC_BitsToWorld

Dieser Funktionsbaustein konvertiert 16 Eingangs-Bits in einen Wert vom Typ "WORD".



Eingänge

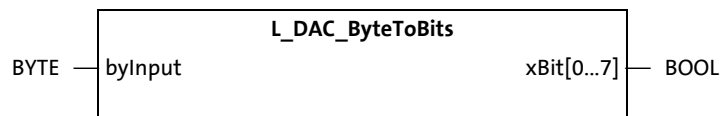
Bezeichner/Datentyp	Info/Einstellmöglichkeiten
xBit[0...15] BOOL	Eingangssignal der Bits 0 ... 15

Ausgänge

Bezeichner/Datentyp	Wert/Bedeutung
wOutput WORD	Ausgabewert, der sich durch die Eingangs-Bits <i>xBit[0...15]</i> ergibt.

10.3.4 L_DAC_ByteToBits

Dieser Funktionsbaustein gibt zu einem Eingangswert vom Typ "BYTE" die 8 entsprechenden Bit-Werte aus.



Eingänge

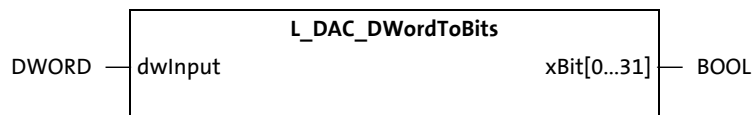
Bezeichner/Datentyp	Info/Einstellmöglichkeiten
byInput BYTE	Eingangssignal

Ausgänge

Bezeichner/Datentyp	Wert/Bedeutung
xBit[0...7] BOOL	Ausgabewerte für die Bits 0 ... 7, die sich durch den Eingangswert <i>byInput</i> ergeben. • Wertigkeit: $2^0 \dots 2^7$

10.3.5 L_DAC_DWordToBits

Dieser Funktionsbaustein gibt zu einem Eingangswert vom Typ "DWORD" die 32 entsprechenden Bit-Werte aus.



Eingänge

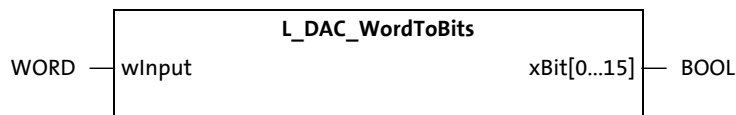
Bezeichner/Datentyp	Info/Einstellmöglichkeiten
dwInput DWORD	Eingangssignal

Ausgänge

Bezeichner/Datentyp	Wert/Bedeutung
xBit[0...31] BOOL	Ausgabewerte für die Bits 0 ... 31, die sich durch den Eingangswert <i>dwInput</i> ergeben. • Wertigkeit: $2^0 \dots 2^{31}$

10.3.6 L_DAC_WordToBits

Dieser Funktionsbaustein gibt zu einem Eingangswert vom Typ "WORD" die 16 entsprechenden Bit-Werte aus.



Eingänge

Bezeichner/Datentyp	Info/Einstellmöglichkeiten
wInput WORD	Eingangssignal

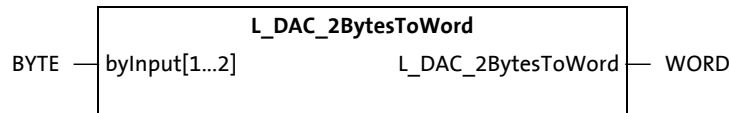
Ausgänge

Bezeichner/Datentyp	Wert/Bedeutung
xBit[0...15] BOOL	Ausgabewerte für die Bits 0 ... 15, die sich durch den Eingangswert <i>wInput</i> ergeben. • Wertigkeit: $2^0 \dots 2^{15}$

10.4 Type converter

10.4.1 L_DAC_2BytesToWord

Diese Funktion konvertiert zwei Eingangswerte vom Typ "BYTE" in einen Ausgangswert vom Typ "WORD".



Eingänge

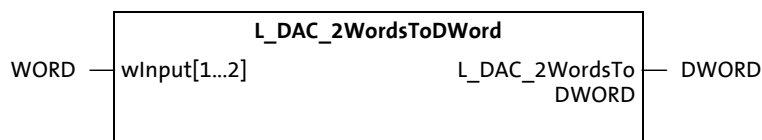
Bezeichner/Datentyp	Info/Einstellmöglichkeiten
byInput1 BYTE	Eingangswert 1 = Bit 0 ... 7 (Byte 1)
byInput2 BYTE	Eingangswert 2 = Bit 8 ... 15 (Byte 2)

Ausgänge

Bezeichner/Datentyp	Info/Einstellmöglichkeiten
L_DAC_2BytesToWord WORD	Ausgabewert, der sich durch die Eingangswerte <i>byInput1</i> (Byte 1) und <i>byInput2</i> (Byte 2) ergibt.

10.4.2 L_DAC_2WordsToDWord

Diese Funktion konvertiert zwei Eingangswerte vom Typ "WORD" in einen Ausgangswert vom Typ "DWORD".



Eingänge

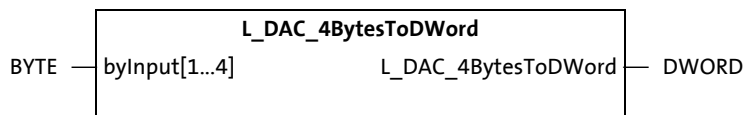
Bezeichner/Datentyp	Info/Einstellmöglichkeiten
wInput1 WORD	Eingangswert 1 = Bit 0 ... 7 (Word 1)
wInput2 WORD	Eingangswert 2 = Bit 8 ... 15 (Word 2)

Ausgänge

Bezeichner/Datentyp	Info/Einstellmöglichkeiten
L_DAC_2WordsToDWord DWORD	Ausgabewert, der sich durch die Eingangswerte <i>wInput1</i> (Word 1) und <i>wInput2</i> (Word 2) ergibt.

10.4.3 L_DAC_4BytesToDWord

Diese Funktion konvertiert vier Eingangswerte vom Typ "BYTE" in einen Ausgangswert vom Typ "DWORD".



Eingänge

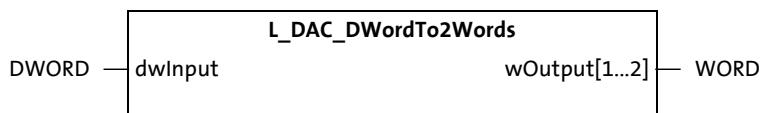
Bezeichner/Datentyp	Info/Einstellmöglichkeiten
byInput1 BYTE	Eingangswert 1 = Bit 0 ... 7 (Byte 1)
byInput2 BYTE	Eingangswert 2 = Bit 8 ... 15 (Byte 2)
byInput3 BYTE	Eingangswert 3 = Bit 16 ... 23 (Byte 3)
byInput4 BYTE	Eingangswert 4 = Bit 24 ... 31 (Byte 4)

Ausgänge

Bezeichner/Datentyp	Info/Einstellmöglichkeiten
L_DAC_4BytesToDWord DWORD	Ausgabewert, der sich durch die Eingangswerte <i>byInput[1...4]</i> (Bytes 1 ... 4) ergibt.

10.4.4 L_DAC_DWordTo2Words

Dieser Funktionsbaustein konvertiert einen Eingangswert vom Typ "DWORD" in zwei Ausgangswerte vom Typ "WORD".



Eingänge

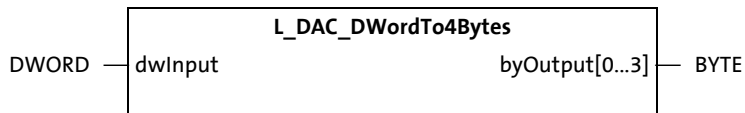
Bezeichner/Datentyp	Info/Einstellmöglichkeiten
dwInput DWORD	Eingangswert

Ausgänge

Bezeichner/Datentyp	Wert/Bedeutung
wOutput1 WORD	Ausgangswert 1 = Bit 0 ... 15 von <i>dwInput</i>
wOutput2 WORD	Ausgangswert 2 = Bit 16 ... 31 von <i>dwInput</i>

10.4.5 L_DAC_DWordTo4Bytes

Dieser Funktionsbaustein konvertiert einen Eingangswert vom Typ "DWORD" in vier Ausgangswerte vom Typ "BYTE".



Eingänge

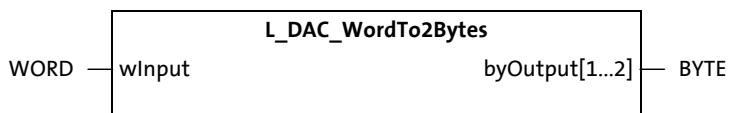
Bezeichner/Datentyp	Info/Einstellmöglichkeiten
dwInput DWORD	Eingangswert

Ausgänge

Bezeichner/Datentyp	Wert/Bedeutung
byOutput0 BYTE	Ausgangswert 1 = Bit 0 ... 7 von <i>dwInput</i>
byOutput1 BYTE	Ausgangswert 2 = Bit 8 ... 15 von <i>dwInput</i>
byOutput2 BYTE	Ausgangswert 3 = Bit 16 ... 23 von <i>dwInput</i>
byOutput3 BYTE	Ausgangswert 4 = Bit 24 ... 31 von <i>dwInput</i>

10.4.6 L_DAC_WordTo2Bytes

Dieser Funktionsbaustein konvertiert einen Eingangswert vom Typ "WORD" in zwei Ausgangswerte vom Typ "BYTE".



Eingänge

Bezeichner/Datentyp	Info/Einstellmöglichkeiten
wInput WORD	Eingangswert

Ausgänge

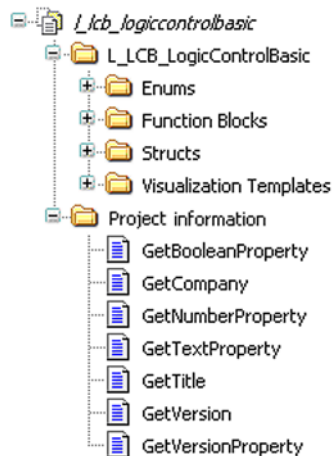
Bezeichner/Datentyp	Wert/Bedeutung
byOutput1 BYTE	Ausgangswert 1 = Bit 0 ... 7 von <i>wInput</i>
byOutput2 BYTE	Ausgangswert 2 = Bit 8 ... 15 von <i>wInput</i>

11

Automatisch generierte Funktionen (Projektinformationen)

In diesem Abschnitt finden Sie Informationen zu den Funktionen, die der »PLC Designer« automatisch generiert.

Die Ausgabewerte finden Sie unter dem Menübefehl **Projekt → Projektinformationen**.



Project information

- ▶ [GetBooleanProperty](#) (132)
- ▶ [GetCompany](#) (133)
- ▶ [GetNumberProperty](#) (133)
- ▶ [GetTextProperty](#) (134)
- ▶ [GetTitle](#) (134)
- ▶ [GetVersion](#) (135)
- ▶ [GetVersionProperty](#) (135)

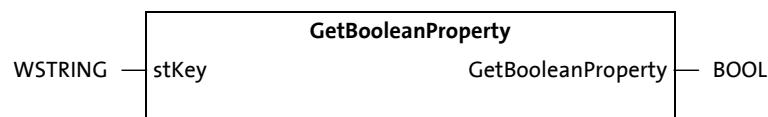
11.1

GetBooleanProperty

Diese Funktion liefert eine boolsche Projektinformationen der Bibliothek.

**Hinweis!**

Diese Funktion muss immer mit dem passenden Namensraum der zugehörigen Bibliothek verwendet werden, z. B. "**L_DAC**.GetBooleanProperty(stKey)".

**Eingänge**

Bezeichner/Datentyp	Info/Einstellmöglichkeiten
stKey WSTRING	Schlüssel der Projektinformation

Ausgänge

Bezeichner/Datentyp	Wert/Bedeutung
GetBooleanProperty BOOL	Wert der Projektinformationen

11.2

GetCompany

Diese Funktion liefert den Namen der Firma, die diese Bibliothek erstellt hat.

**Hinweis!**

Diese Funktion muss immer mit dem passenden Namensraum der zugehörigen Bibliothek verwendet werden, z. B. "**L_DAC**.GetCompany ()".

**Ausgänge**

Bezeichner/Datentyp	Wert/Bedeutung
getCompany WSTRING	Name der Firma, die die Bibliothek erstellt hat.

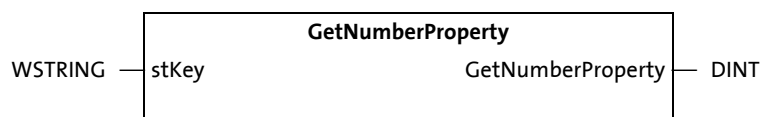
11.3

GetNumberProperty

Diese Funktion liefert eine numerische Projektinformationen der Bibliothek.

**Hinweis!**

Diese Funktion muss immer mit dem passenden Namensraum der zugehörigen Bibliothek verwendet werden, z. B. "**L_DAC**.GetNumberProperty (stKey)".

**Eingänge**

Bezeichner/Datentyp	Info/Einstellmöglichkeiten
stKey WSTRING	Schlüssel der Projektinformation

Ausgänge

Bezeichner/Datentyp	Wert/Bedeutung
GetNumberProperty DINT	Wert der Projektinformationen

11.4

GetTextProperty

Diese Funktion liefert eine textuelle Projektinformationen der Bibliothek.

**Hinweis!**

Diese Funktion muss immer mit dem passenden Namensraum der zugehörigen Bibliothek verwendet werden, z. B. "**L_DAC**.GetTextProperty(stKey)".

**Eingänge**

Bezeichner/Datentyp	Info/Einstellmöglichkeiten
stKey WSTRING	Schlüssel der Projektinformation • "Company" - Name des Bibliothekslieferanten • "Title" - Name der Bibliothek • "DefaultNamespace" - Namensraum der Bibliothek • "Project" - Name der Bibliothek • "Description" - Beschreibung der Bibliothek • "Author" - Autor der Bibliothek

Ausgänge

Bezeichner/Datentyp	Wert/Bedeutung
GetTextProperty WSTRING	Wert der Projektinformationen

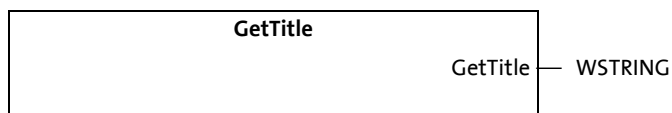
11.5

GetTitle

Diese Funktion liefert den Titel der Bibliothek.

**Hinweis!**

Diese Funktion muss immer mit dem passenden Namensraum der zugehörigen Bibliothek verwendet werden, z. B. "**L_DAC**.GetTitle".

**Ausgänge**

Bezeichner/Datentyp	Wert/Bedeutung
GetTitle WSTRING	Name der Bibliothek

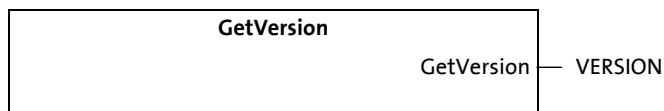
11.6

GetVersion

Diese Funktion liefert die Version der Bibliothek.

**Hinweis!**

Diese Funktion muss immer mit dem passenden Namensraum der zugehörigen Bibliothek verwendet werden, z. B. "**L_DAC**.GetVersion".

**Ausgänge**

Bezeichner/Datentyp	Wert/Bedeutung
GetVersion VERSION	Version der Bibliothek

11.7

GetVersionProperty

Diese Funktion liefert die Version der Bibliothek.

**Hinweis!**

Diese Funktion muss immer mit dem passenden Namensraum der zugehörigen Bibliothek verwendet werden, z. B. "**L_DAC**.GetVersion(stKey)".

**Eingänge**

Bezeichner/Datentyp	Info/Einstellmöglichkeiten
stKey WSTRING	Schlüssel der Projektinformation • Erweiterte Versionierungsinformation der Bibliothek

Ausgänge

Bezeichner/Datentyp	Wert/Bedeutung
GetVersionProperty VERSION	Version der Bibliothek

12 Index

A

Anbindung von Funktionsbausteinen [65](#), [103](#)
Anwendungsbeispiele [9](#)
Anwendungshinweise (Darstellung) [13](#)
Application Samples [19](#)
Application Samples - Programmbausteine [29](#)
Application Samples - Visualisierungen [30](#), [36](#)
Application Samples verwenden (Engineer) [22](#)
Application Samples verwenden (PLC Designer) [24](#)
Application Samples, Aufbau [19](#)
Application Samples, Bezeichnung [20](#)
Application Samples, Übersicht [20](#)
Aufbau der Application Samples [19](#)
Automatic (Visualisierung "Stellantrieb - Drehzahl") [32](#)
Automatic (Visualisierung "Tabellenpositionierung") [38](#)
Automatisch generierte Funktionen (Projektinformationen) [132](#)

B

Begriffe [12](#)
Beispielprojekte - Programmbausteine [29](#)
Beispielprojekte - Visualisierungen [30](#), [36](#)
Beispielprojekte erweitern (Geräte hinzufügen) [43](#)
Beispielprojekte verwenden (Engineer) [22](#)
Beispielprojekte verwenden (PLC Designer) [24](#)
Belegung der Ausgangs-Ports (L_LCB_ActuatorSpeed) [76](#)
Belegung der Ausgangs-Ports (L_LCB_TablePositioning) [82](#)
Belegung der Eingangs-Ports (L_LCB_ActuatorSpeed) [75](#)
Belegung der Eingangs-Ports (L_LCB_TablePositioning) [81](#)
Bezeichnung der Application Samples [20](#)
Bit operations [121](#)
Bit splitter [126](#)

C

CANbus [48](#)
CAN-Feldgeräte anhängen [46](#)
CANopen Manager [49](#)
CAN-Steuerungskonfiguration [26](#)

D

Dokumenthistorie [10](#)

E

E-Mail an Lenze [139](#)
Enumerationen (L_DCO_DriveCommunication) [115](#)
Enumerationen (L_LCB_LogicControlBasic) [102](#)
EtherCAT-Feldgeräte anhängen [43](#)
EtherCAT-Steuerungskonfiguration [26](#)

F

Feedback an Lenze [139](#)
Fehlende Geräte oder Geräte anderer Hersteller [43](#)
Fehlermeldungen (L_LCB_Error) [102](#), [116](#)
Feldbus-Scan (EtherCAT) [43](#)
Feldgeräte verdrahten [17](#)
Funktionen und Funktionsbausteine (Übersicht L_DAC_DataConversion) [120](#)
Funktionen und Funktionsbausteine (Übersicht L_DCO_DriveCommunication) [104](#)
Funktionen und Funktionsbausteine (Übersicht L_LCB_LogicControlBasic) [66](#)
Funktionsbaustein anbinden [65](#), [103](#)
Funktionsbausteine (L_DCO_DriveCommunication) [105](#)
Funktionsbausteine (L_LCB_LogicControlBasic) [67](#)
Funktionsbibliothek L_DAC_DataConversion [120](#)
Funktionsbibliothek L_DCO_DriveCommunication [103](#)
Funktionsbibliothek L_LCB_LogicControlBasic [65](#)

G

Generic Drive [63](#)
Geräte (Controller, Inverter, Motoren) verdrahten [17](#)
Geräte anderer Hersteller [43](#)
Geräte anhängen (EtherCAT) [43](#)
Gerätebeschreibungen [50](#)
GetBooleanProperty [132](#)
GetCompany [133](#)
GetNumberProperty [133](#)
GetTextProperty [134](#)
GetTitle [134](#)
GetVersion [135](#)
GetVersionProperty [135](#)
Gültigkeit der Dokumentation [9](#)

H

Homing (Visualisierung "Tabellenpositionierung") [41](#)

I

Information (Visualisierung "Stellantrieb - Drehzahl") [31](#)

Information (Visualisierung "Tabellenpositionierung") [37](#)

Inverter Drive 8400 BaseLine C [50](#)

Inverter Drive 8400 HighLine C [54](#)

Inverter Drive 8400 StateLine C [52](#)

Inverter Drive 8400 TopLine C [56](#)

K

Kommunikation zum Controller einrichten [18](#)

L

L_DAC_2BytesToWord [129](#)

L_DAC_2WordsToDWord [129](#)

L_DAC_4BytesToDWord [130](#)

L_DAC_BitsToByte [126](#)

L_DAC_BitsToDWord [126](#)

L_DAC_BitsToWord [127](#)

L_DAC_ByteToBits [127](#)

L_DAC_DataConversion [120](#)

L_DAC_DWordTo2Words [130](#)

L_DAC_DWordTo4Bytes [131](#)

L_DAC_DWordToBits [128](#)

L_DAC_GetBitOfByte [121](#)

L_DAC_GetBitOfDWord [121](#)

L_DAC_GetBitOfWord [122](#)

L_DAC_ResetBitOfByte [122](#)

L_DAC_ResetBitOfDWord [123](#)

L_DAC_ResetBitOfWord [123](#)

L_DAC_SetBitOfByte [124](#)

L_DAC_SetBitOfDWord [124](#)

L_DAC_SetBitOfWord [125](#)

L_DAC_WordTo2Bytes [131](#)

L_DAC_WordToBits [128](#)

L_DCO_AXIS_REF_BASE [105](#)

L_DCO_CommState [115](#)

L_DCO_ConvertData [119](#)

L_DCO_DriveCommunication [103](#)

L_DCO_Error [116](#)

L_DCO_ReadDriveParameter [106](#)

L_DCO_ReadDriveParameterString [108](#)

L_DCO_SDOData [114](#)

L_DCO_TransferData [114](#)

L_DCO_TransferDriveParameter [110](#)

L_DCO_WriteDriveParameter [112](#)

L_LCB_8400Drive [69](#)

L_LCB_9400Drive [70](#)

L_LCB_AccToUnit [94](#)

L_LCB_ActuatorSpeed [73](#)

L_LCB_AXIS_REF [67](#)

L_LCB_AXIS_REF_CAN [68](#)

L_LCB_AXIS_REF_ETC [68](#)

L_LCB_AxisData [101](#)

L_LCB_AXISMODE [102](#)

L_LCB_Error [102](#)

L_LCB_GenericDrive [72](#)

L_LCB_GetAxisData [87](#)

L_LCB_GetPosition [88](#)

L_LCB_GetSpeed [89](#)

L_LCB_LogicControlBasic [65](#)

L_LCB_LOGICDEVICE [102](#)

L_LCB_Norm_aToNorm_n [83](#)

L_LCB_Norm_nToNorm_a [83](#)

L_LCB_Norm_nToSpeed_s [84](#)

L_LCB_PosToUnit [95](#)

L_LCB_SetAxisData [90](#)

L_LCB_SetPosition [92](#)

L_LCB_SetSpeed [93](#)

L_LCB_Speed_nToNorm_r [84](#)

L_LCB_Speed_rToNorm_n [85](#)

L_LCB_Speed_sToNorm_n [85](#)

L_LCB_Speed_sToSpeed_v [86](#)

L_LCB_Speed_vToSpeed_s [86](#)

L_LCB_SpeedToUnit [96](#)

L_LCB_TablePositioning [77](#)

L_LCB_TaskCycle [100](#)

L_LCB_UnitToAcc [97](#)

L_LCB_UnitToPos [98](#)

L_LCB_UnitToSpeed [99](#)

Lenze Generic Drive [63](#)

M

Manual (Visualisierung "Stellantrieb - Drehzahl") [33](#)

Manual (Visualisierung "Tabellenpositionierung") [39](#)

P

Parameter (Visualisierung "Stellantrieb - Drehzahl") [35](#)

Parameter (Visualisierung "Tabellenpositionierung") [42](#)

Position Override aktivieren [77](#)

Programmbausteine (PRG) [29](#)

Projektinformationen [132](#)

S

Screenshots [9](#)
Service (Visualisierung "Stellantrieb - Drehzahl") [34](#)
Service (Visualisierung "Tabellenpositionierung") [40](#)
Servo Drive 9400 HighLine [58](#)
Servo Drive 9400 Regenerative Power Supply Module [61](#)
Sicherheitshinweise [14](#)
Sicherheitshinweise (Darstellung) [13](#)
Stellantrieb - Drehzahl (Beispielprojekt im Engineer) [22](#)
Stellantrieb - Drehzahl (Beispielprojekt im PLC Designer) [27](#)
Stellantrieb - Drehzahl (Belegung der Ausgangs-Ports) [76](#)
Stellantrieb - Drehzahl (Belegung der Eingangs-Ports) [75](#)
Stellantrieb - Drehzahl (Programmbausteine) [29](#)
Stellantrieb - Drehzahl (Visualisierungen) [30](#)
Steuerungskonfiguration (EtherCAT/CAN) [26](#)
Strukturen (L_DCO_DriveCommunication) [114](#)
Strukturen (L_LCB_LogicControlBasic) [101](#)
Sync-Erzeugung einstellen [49](#)
Systemvoraussetzungen [16](#)

T

Tabellenpositionierung (Beispielprojekt im Engineer) [22](#)
Tabellenpositionierung (Beispielprojekt im PLC Designer) [28](#)
Tabellenpositionierung (Belegung der Ausgangs-Ports) [82](#)
Tabellenpositionierung (Belegung der Eingangs-Ports) [81](#)
Tabellenpositionierung (Programmbausteine) [29](#)
Tabellenpositionierung (Visualisierungen) [36](#)
Type converter [129](#)

U

Übersicht der Application Samples [20](#)
Unions (L_DCO_DriveCommunication) [119](#)

V

Verwendete Hinweise [13](#)
Verwendete Konventionen [11](#)
Visualisierungen "Stellantrieb - Drehzahl" [30](#)
Visualisierungen "Tabellenpositionierung" [36](#)
Voraussetzungen [16](#)

Z

Zielgruppe [9](#)



Ihre Meinung ist uns wichtig

Wir erstellen diese Anleitung nach bestem Wissen mit dem Ziel, Sie bestmöglich beim Umgang mit unserem Produkt zu unterstützen.

Vielleicht ist uns das nicht überall gelungen. Wenn Sie das feststellen sollten, senden Sie uns Ihre Anregungen und Ihre Kritik in einer kurzen E-Mail an:

feedback-docu@lenze.com

Vielen Dank für Ihre Unterstützung.

Ihr Lenze-Dokumentationsteam

Lenze Automation GmbH
Postfach 10 13 52, 31763 Hameln
Hans-Lenze-Straße 1, 31855 Aerzen
GERMANY
HR Hannover B 205381
 +49 5154 82-0
 +49 5154 82-2800
 lenze@lenze.com
 www.lenze.com

Service

Lenze Service GmbH
Breslauer Straße 3, 32699 Extertal
GERMANY
 008000 24 46877 (24 h helpline)
 +49 5154 82-1112
 service@lenze.com