

Lenze

Handbuch



Global Drive

Software Package - Cam

Funktionsbibliothek

LenzeCamControl.lib

Diese Dokumentation ist gültig für die Funktionsbibliotheken **LenzeCamControl1Vxxxx.lib** und **LenzeCamControl2Vxxxx.lib** ab Version 2.00.

Die Funktionsbibliotheken können für folgende Lenze PLC-Geräte verwendet werden:

	Typenbezeichnung	ab Hardwarestand	ab Softwarestand
9300 Servo PLC	EVS93XX-xT	6A	6.0
ECSxA	ECSxAxxx	1C	7.0

Systemvoraussetzungen **Drive PLC Developer Studio**:

DDS-P (ESP-DDS2-P) oder DDS-PD (ESP-DDS2-PD) ab Version 2.2

Wichtiger Hinweis:

Die Software wird dem Benutzer in der vorliegenden Form zur Verfügung gestellt. Alle Risiken hinsichtlich der Qualität und der durch ihren Einsatz ermittelten Ergebnisse verbleiben beim Benutzer. Entsprechende Sicherheitsvorkehrungen gegen eventuelle Fehlbedienungen sind vom Benutzer vorzusehen.

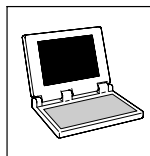
Wir übernehmen keine Verantwortung für direkt oder indirekt entstandene Schäden, z. B. Gewinnverluste, Auftragsverluste oder geschäftliche Beeinträchtigungen jeglicher Art.

© 2007 Lenze Drive Systems GmbH

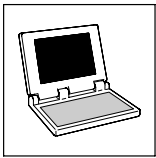
Ohne besondere schriftliche Genehmigung von Lenze Drive Systems GmbH darf kein Teil dieser Dokumentation vervielfältigt oder Dritten zugänglich gemacht werden.

Wir haben alle Angaben in dieser Dokumentation mit größter Sorgfalt zusammengestellt und auf Übereinstimmung mit der beschriebenen Hard- und Software geprüft. Trotzdem können wir Abweichungen nicht ganz ausschließen. Wir übernehmen keine juristische Verantwortung oder Haftung für Schäden, die dadurch eventuell entstehen. Notwendige Korrekturen werden wir in die nachfolgenden Auflagen einarbeiten.

Alle in dieser Dokumentation aufgeführten Markennamen sind Warenzeichen ihrer jeweiligen Besitzer.

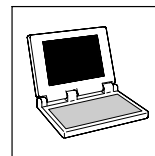


1 Vorwort und Allgemeines	1-1
1.1 Über dieses Handbuch	1-1
1.1.1 Konventionen in diesem Handbuch	1-1
1.1.2 Piktogramme in diesem Handbuch	1-1
1.1.3 Aufbau der Beschreibungen	1-2
1.1.4 Verwendete Begriffe	1-2
1.2 Versionskennung der Funktionsbibliotheken	1-3
1.3 Technologiefunktion "Kurvenscheibe"	1-4
1.3.1 Konzept	1-4
1.3.2 Allgemeine Hinweise zu den Funktionsblöcken	1-4
2 Funktionsblöcke LenzeCamControl1V02xx.lib	2-1
2.1 L_CamClutchXAxis - Virtuelle Kupplung	2-2
2.2 L_CamContactor - Nockenschaltwerk	2-7
2.3 L_CamContactorVPos - Dynamische Verstellung der Nocken/Hysteresse	2-16
2.4 L_CamControl - Sollwertaufbereitung	2-20
2.5 L_CamData - Profilverwaltung	2-28
2.6 L_CamExtrapolate - Extrapolation	2-38
2.7 L_CamGetPosition - Einheitenumrechnung	2-40
2.8 L_CamLimitVelocity - Begrenzungsglied	2-41
2.9 L_CamMachineData - Maschinenparameter	2-42
2.10 L_CamMonitFollowError - Schleppfehlerüberwachung	2-44
2.11 L_CamPosCtrlLin - Linearer Profilgenerator	2-46
2.12 L_CamProfileData - Laden wichtiger Profilpunkte	2-50
2.13 L_CamSequence - Vorgabe einer Profilsequenz	2-52
2.14 L_CamSetPosition - Einheitenumrechnung	2-58
2.15 L_CamSetXAxisVelocity - Dehnen und Stauchen des Leitwerts	2-59
2.16 L_CamSetYAxisVelocity - Dehnen und Stauchen des Sollwerts	2-64
2.17 L_CamVMasterPosition - Virtueller Master mit Positionierfunktion	2-68
2.18 L_CamVMasterVelocity - Virtueller Master	2-74
3 Funktionsblöcke LenzeCamControl2V02xx.lib	3-1
3.1 L_CamAddVelocity - Drehzahl-Addition	3-2
3.2 L_CamAdjustDfIn - Leitfrequenzanpassung	3-3
3.3 L_CamDifferentiate - Konverter/Differenzierer	3-4
3.4 L_CamEditProfileData - Online-Änderung der Profilpunkte	3-5
3.5 L_CamIntegrateVarLim - Winkelintegrator	3-11
3.6 L_CamSectionShift - Abschnittsverschiebung	3-14
3.7 L_CamSelPosition - Leitwertauswahl	3-17
3.8 L_CamSelVelocity - Leitfrequenzauswahl	3-18
3.9 L_CamVelocityFilter - Filter (PT1) für Geschwindigkeitssignale	3-19
4 Index	4-1



Funktionsbibliothek LenzeCamControl.lib

Inhalt



1 Vorwort und Allgemeines

1.1 Über dieses Handbuch

Dieses Handbuch enthält Informationen zu den Funktionsbibliotheken **LenzeCamControl1Vxxx.lib** und **LenzeCamControl2Vxxx.lib** für das **Drive PLC Developer Studio**.

- In den Funktionsbibliotheken **LenzeCamControl1Vxxx.lib** und **LenzeCamControl2Vxxx.lib** sind Funktionen zur Realisierung von Kurvenscheibenaufgaben enthalten.
- Die Funktionsbibliotheken sind Bestandteil des **Software Package - Cam**.

1.1.1 Konventionen in diesem Handbuch

Dieses Handbuch verwendet folgende Konventionen zur Unterscheidung von verschiedenen Arten von Informationen:

Variablenbezeichner

sind im erklärenden Text in Kursivschrift dargestellt:

- "Mit *bReset...*"



Tipp!

Informationen zu den Konventionen, die für die Variablenbezeichner von Lenze Systembausteinen, Funktionsblöcken sowie Funktionen verwendet werden, um eine einheitliche und durchgängige Benennung zu gewährleisten und dadurch die Lesbarkeit von SPS-Programmen zu fördern, finden Sie im Anhang der DDS-Online-Dokumentation "Einführung in die IEC 61131-3 Programmierung".

Lenze Funktionen/Funktionsblöcke

erkennen Sie am Namen. Sie beginnen immer mit "L_":

- "Mit dem FB **L_CamControl...**"

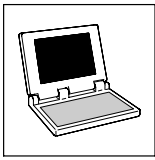
Programmlistings

sind in der Schriftart "Courier" aufgeführt, wobei Schlüsselwörter fett ausgezeichnet sind:

- "**IF** (ReturnValue < 0) **THEN...**"

1.1.2 Piktogramme in diesem Handbuch

	verwendete Piktogramme	Signalwörter	
Warnung vor Sachschäden		Stop!	Warnt vor möglichen Sachschäden . Mögliche Folgen bei Missachtung: Beschädigung des Antriebsreglers/Antriebssystems oder seiner Umgebung.
Sonstige Hinweise		Tipp! Hinweis!	Kennzeichnet einen Tipp bzw. Hinweis.



Funktionsbibliothek LenzeCamControl.lib

Vorwort und Allgemeines Über dieses Handbuch

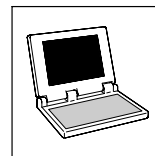
1.1.3 Aufbau der Beschreibungen

Die Beschreibungen der einzelnen Funktionen/Funktionsblöcke sowie Systembausteine in diesem Handbuch sind einheitlich nach folgendem Schema aufgebaut:

	Funktion	Funktionsblock (FB)/ Systembaustein (SB)
①	Überschrift mit Funktion und Bezeichner	
②	Deklaration der Funktion: - Datentyp des Rückgabewertes - Funktionsbezeichner - Liste der Übergabeparameter	-
③	Kurzbeschreibung mit den wichtigsten Eigenschaften	
④	Grafische Darstellung der Funktion mit den zugehörigen Variablen - Übergabeparameter - Rückgabewert	Grafische Darstellung des FB/SB mit den zugehörigen Variablen - Eingangsvariablen - Ausgangsvariablen
⑤	Tabelle mit Informationen zu den Übergabeparametern: - Bezeichner - Datentyp - Einstellmöglichkeiten - Info	Tabelle mit Informationen zu den Ein- und Ausgangsvariablen: - Bezeichner - Datentyp - Variablentyp - Einstellmöglichkeiten - Info
⑥	Tabelle mit Informationen zum Rückgabewert: - Datentyp des Rückgabewertes - Mögliche Rückgabewerte und ihre Bedeutung	-
⑦	Weitere Informationen (Anmerkungen, Tipps, Anwendungsbeispiele, etc.)	

1.1.4 Verwendete Begriffe

Begriff	Im folgenden Text verwendet für
Cxxx/y	Subcode y der Codestelle Cxxxx (z. B. C0470/3 = Subcode 3 der Codestelle C0470)
DDS	Drive PLC Developer Studio
FB	Funktionsblock
GDC	Global Drive Control (Parametrier-Programm von Lenze)
Parameter-Codestellen	Codestellen zum Einstellen der Funktionalität eines Funktionsblocks
PLC	9300 Servo PLC - Achsmodul ECSxA "Application"
SB	Systembaustein
[unit]	Platzhalter für die physikalische Einheit des verwendeten externen Maßsystems (z. B. "µm", "mm", "cm")



1.2 Versionskennung der Funktionsbibliotheken

Versionskennung **ab Version 1.05** der Funktionsbibliotheken

- LenzeCamControl1Vxxxx.lib
- LenzeCamControl2Vxxxx.lib

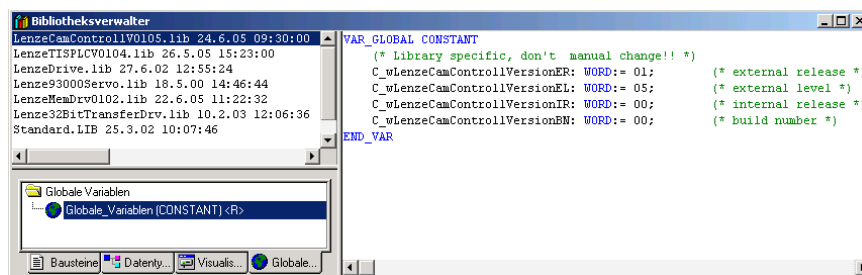
Konstante	Bedeutung	Beispiel Wert
C_w[Funktionsbibliotheksname]VersionER	Externes Release	01
C_w[Funktionsbibliotheksname]VersionEL	Externes Level	05
C_w[Funktionsbibliotheksname]VersionIR	Internes Release	00
C_w[Funktionsbibliotheksname]VersionBN	Build-Nummer	00

Version: 01 05 00 00

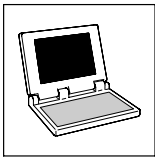
Der Wert dieser Konstanten ist hexadezimal-kodiert.

- Im Beispiel bedeutet der Wert "01050000" Version "1.05.00.00".

Im Offline-Modus können Sie sich den Wert dieser Konstanten über die **Bibliotheksverwaltung/Register Globale Variablen** anzeigen lassen:



- Die Bibliotheksverwaltung öffnen Sie im DDS über den Befehl **Fenster→Bibliotheksverwaltung**.
- Wählen Sie die Registerkarte **Globale Variablen** und öffnen Sie den Ordner **Globale Variablen**.
- Wenn Sie die *Globale_Variablen...* markieren, erscheint die Version im Bibliotheksverwalter.



Funktionsbibliothek LenzeCamControl.lib

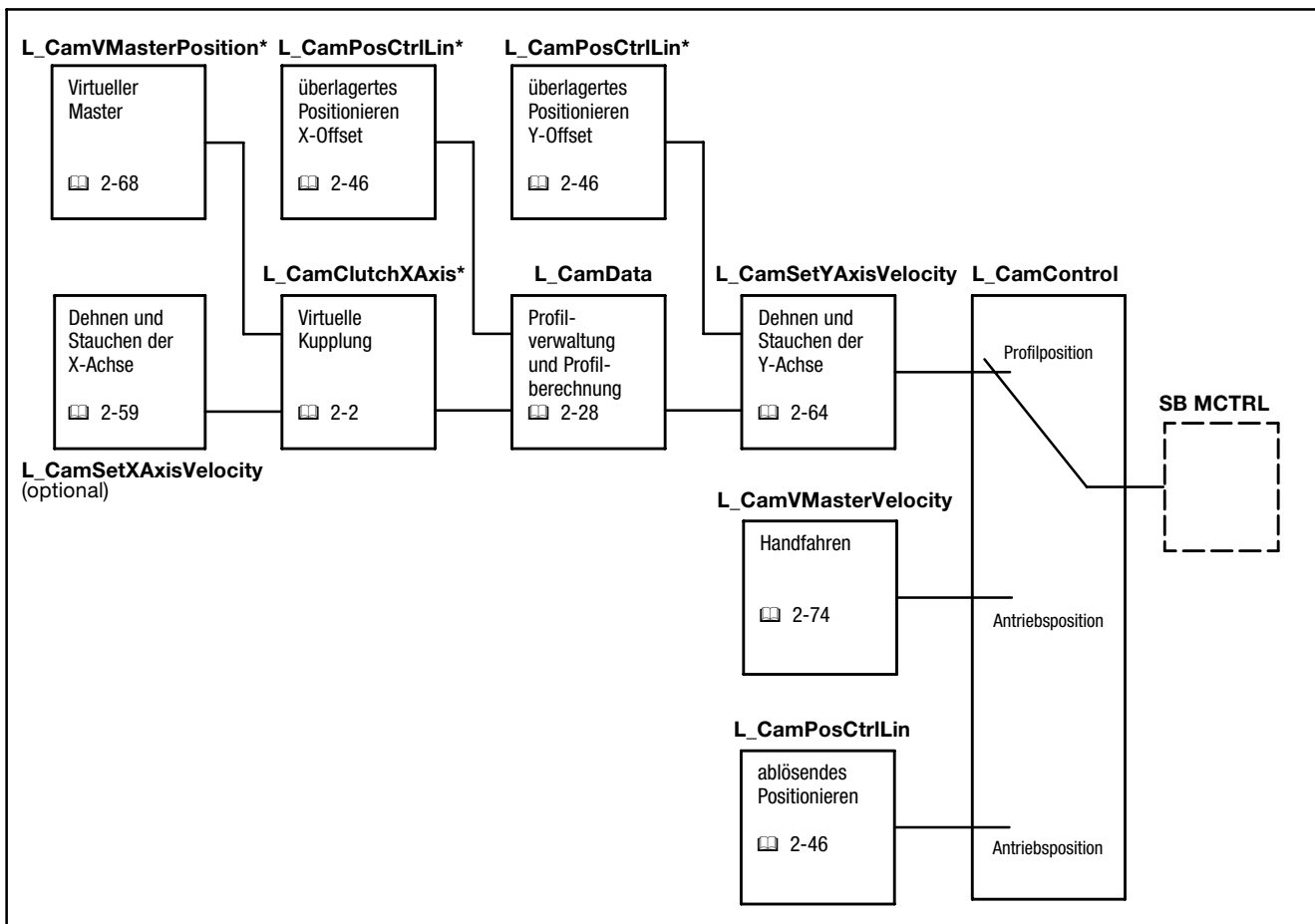
Vorwort und Allgemeines

Technologiefunktion "Kurvenscheibe"

1.3 Technologiefunktion "Kurvenscheibe"

1.3.1 Konzept

Die Funktionsblöcke für die Technologiefunktion "Kurvenscheibe" basieren auf folgendem Konzept:



1.3.2 Allgemeine Hinweise zu den Funktionsblöcken

Zeitäquidistanter Aufruf von Funktionsblöcken

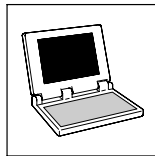
Die folgenden Funktionsblöcke müssen auf Grund ihrer internen Struktur in einer zeitäquidistanten Task aufgerufen werden (z. B. in einer 2 ms-Task):

L_CamClutchXAxis	L_CamExtrapolate	L_CamSetXAxisVelocity
L_CamContactor	L_CamIntegrateVarLim	L_CamSetYAxisVelocity
L_CamControl	L_CamPosCtrlLin	L_CamVMasterPosition
L_CamData	L_CamSectionShift	L_CamVMasterVelocity
L_CamDifferentiate		



Hinweis!

Die zyklische Task **PLC_PRG** ist nicht zeitäquidistant!



Freischalteingang "bEnable"

Funktionsblöcke, die diesen Eingang besitzen, können hierüber aktiviert (Zustand TRUE) bzw. bei Nichtverwendung aus Rechenzeitgründen deaktiviert werden (Zustand FALSE).



Stop!

Beachten Sie, dass Zustandsänderungen am Freischalteingang *bEnable* den Antrieb in Bewegung versetzen können (Ausgleichsbewegungen).

Statusausgang "nState"

Über diesen Ausgang kann bei einigen Funktionsblöcken der aktuelle Status abgefragt werden.

- Werte > "0" zeigen Warnungen an, die interne Funktion wird aber nicht unterbrochen.
- Der Wert "0" zeigt den Zustand "OK" an.
- Werte < "0" zeigen Fehler an und führen zum Abbruch der Funktion.



Tipp!

Eine angezeigte Fehlernummer wird intern nicht gespeichert und steht nur solange an, bis der Fehler behoben ist.



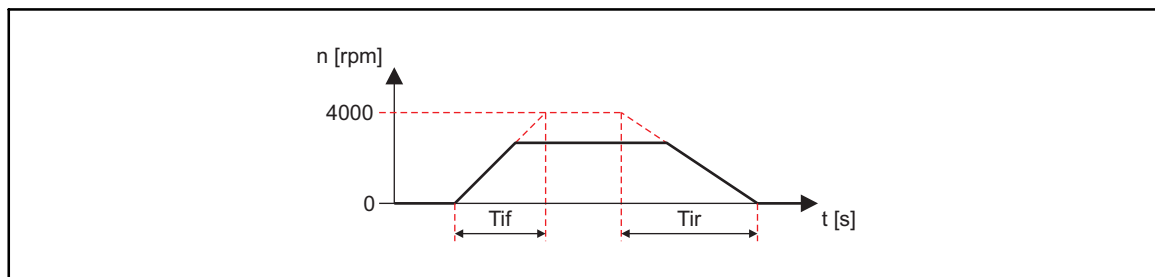
Stop!

Beachten Sie, dass auch Warnungen zu einem Fehlverhalten führen können!

Beschleunigungs-/Verzögerungsrampen

Bei Funktionsblöcken mit Positioniereigenschaften ist die Steilheit der Beschleunigungs- und Verzögerungsrampen durch sogenannte Ti-Zeiten definiert.

- Die Parametrierung dieser Ti-Zeiten können Sie über entsprechende Codestellen vornehmen.
- Die Ti-Zeiten beziehen sich immer auf eine Geschwindigkeitsänderung von 4000 rpm:



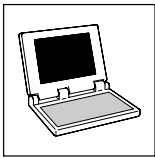
Funktionsblöcke mit *.lc9-Zugriff

Die Funktionsblöcke **L_CamContactor**, **L_CamData**, **L_CamMachineData** und **L_CamProfile-Data** greifen auf die *.lc9-Datei zu, die die Daten der Bewegungsprofile enthält und im RAM-Speicherblock 1 der PLC abgelegt ist. Die *.lc9-Datei wird über den **CamDesigner** erzeugt (siehe auch die entsprechende Dokumentation vom CamDesigner/CamLoader).



Hinweis!

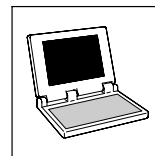
Durch die Nutzung des **RAM-Speicherblock 1** für die Ablage der *.lc9-Datei steht dieser Speicherbereich somit nicht mehr zur freien Verfügung. Beachten Sie diese Einschränkung bei der Verwendung der Speicherfunktionen der Funktionsbibliothek **LenzeMemDrv.lib** (siehe auch Handbuch zur Funktionsbibliothek **LenzeMemDrv.lib**)!



Funktionsbibliothek LenzeCamControl.lib

Vorwort und Allgemeines

Technologiefunktion "Kurvenscheibe"



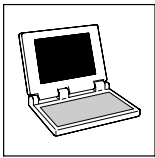
2

Funktionsblöcke *LenzeCamControl1V02xx.lib*

Die Funktionsblöcke für die Technologiefunktion "Kurvenscheibe" sind auf die zwei Funktionsbibliotheken **LenzeCamControl1V02xx.lib** und **LenzeCamControl2V02xx.lib** aufgeteilt.



In diesem Kapitel erhalten Sie Informationen zu den Funktionsblöcken, die in der Funktionsbibliothek **LenzeCamControl1V02xx.lib** enthalten sind.



Funktionsbibliothek LenzCamControl.lib

Funktionsblöcke

2.1 L_CamClutchXAxis - Virtuelle Kupplung

Über diesen FB kann ein positionsgenaues Ein- und Auskuppeln der virtuellen Königswelle (X-Achse) mit einstellbaren Beschleunigungs- und Bremsrampen realisiert werden.

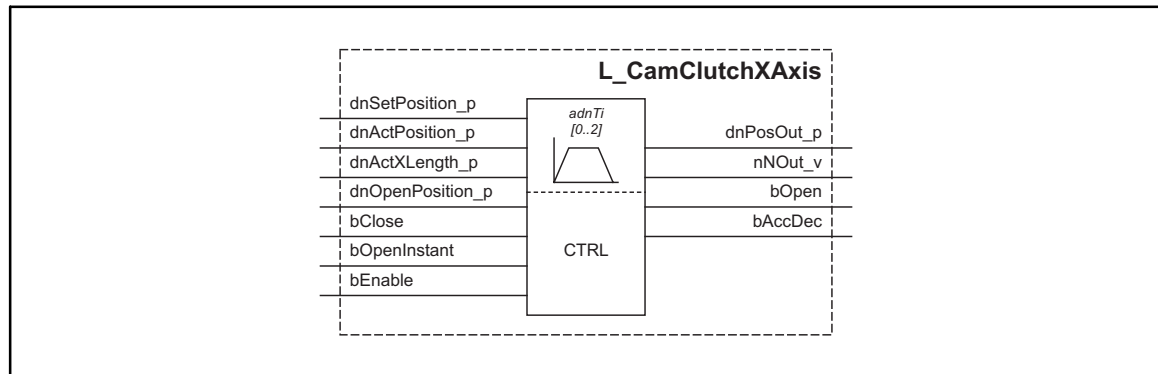
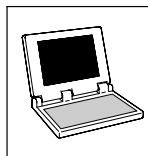


Abb. 2-1 Funktionsblock "L_CamClutchXAxis"

⇒ Eingänge (Variablentyp: VAR_INPUT)							
Bezeichner	Datentyp	Signaltyp	Info/Einstellmöglichkeiten				
dnSetPosition_p	Double Integer	position	Eingang der virtuellen Königswelle (extern)				
dnActPosition_p	Double Integer	position	Istposition der virtuellen Königswelle (intern) • z. B. Rückführung von <i>dnPosOut_p</i> (siehe Verschaltungsbeispiel)				
dnActXLength_p	Double Integer	position	Taktlänge der X-Achse • Eingang mit <i>dnActXLength_p</i> vom FB L_CamData verbinden.				
dnOpenPosition_p	Double Integer	position	Sollposition der virtuellen Königswelle nach Öffnen der Kupplung im Stillstand				
bClose	Bool	binary	Kupplung <table><tr><td>FALSE</td><td>Kupplung öffnen, über <i>adnTi[2]</i> bremsen und an <i>dnOpenPosition_p</i> anhalten.</td></tr><tr><td>TRUE</td><td>Über <i>adnTi[1]</i> auf <i>dnSetPosition_p</i> aufsynchronisieren, Kupplung schließen.</td></tr></table>	FALSE	Kupplung öffnen, über <i>adnTi[2]</i> bremsen und an <i>dnOpenPosition_p</i> anhalten.	TRUE	Über <i>adnTi[1]</i> auf <i>dnSetPosition_p</i> aufsynchronisieren, Kupplung schließen.
FALSE	Kupplung öffnen, über <i>adnTi[2]</i> bremsen und an <i>dnOpenPosition_p</i> anhalten.						
TRUE	Über <i>adnTi[1]</i> auf <i>dnSetPosition_p</i> aufsynchronisieren, Kupplung schließen.						
bOpenInstant	Bool	binary	Zwangsöffnung • Dieser Eingang hat Priorität gegenüber <i>bClose</i>. • Die Stillstandsposition der X-Achse ergibt sich aus der momentanen Geschwindigkeit der X-Achse und der eingestellten Verzögerungsrampe <i>adnTi[0]</i>. <table><tr><td>FALSE</td><td>Kupplungsfunktion aktiv</td></tr><tr><td>TRUE</td><td>Die Königswelle (intern) wird unmittelbar über <i>adnTi[0]</i> in den Stillstand geführt.</td></tr></table>	FALSE	Kupplungsfunktion aktiv	TRUE	Die Königswelle (intern) wird unmittelbar über <i>adnTi[0]</i> in den Stillstand geführt.
FALSE	Kupplungsfunktion aktiv						
TRUE	Die Königswelle (intern) wird unmittelbar über <i>adnTi[0]</i> in den Stillstand geführt.						
bEnable	Bool	binary	Kupplungsfunktion aktivieren • Dieser Eingang hat die höchste Priorität. <table><tr><td>FALSE</td><td><i>dnSetPosition_p</i> wird direkt am Ausgang <i>dnPosOut_p</i> ausgegeben. Die Rampenfunktion ist inaktiv. Die Kupplung ist geschlossen (eingekuppelt).</td></tr><tr><td>TRUE</td><td>Kupplungsfunktion aktiv</td></tr></table>	FALSE	<i>dnSetPosition_p</i> wird direkt am Ausgang <i>dnPosOut_p</i> ausgegeben. Die Rampenfunktion ist inaktiv. Die Kupplung ist geschlossen (eingekuppelt).	TRUE	Kupplungsfunktion aktiv
FALSE	<i>dnSetPosition_p</i> wird direkt am Ausgang <i>dnPosOut_p</i> ausgegeben. Die Rampenfunktion ist inaktiv. Die Kupplung ist geschlossen (eingekuppelt).						
TRUE	Kupplungsfunktion aktiv						

Ausgänge → (Variablentyp: VAR_OUTPUT)			
Bezeichner	Datentyp	Signaltyp	Wert/Bedeutung
dnPosOut_p	Double Integer	position	Istposition der virtuellen Königswelle (intern)
nNOut_v	Integer	velocity	Geschwindigkeit der virtuellen Königswelle (intern)
			Hinweis: Das Winkeldifferenzsignal berücksichtigt den inkrementellen Taktsprung von $x_{\max}$ auf 0 am Ende/Anfang eines Taktes, nicht jedoch einen eventuellen inkrementellen Nachkommaanteil in der Taktlänge. Verwenden Sie dieses Ausgangssignal folglich nicht, wenn die Taktlänge inkrementell nicht ganzzahlig darstellbar ist!
bOpen	Bool	binary	Kupplung öffnen
			TRUE Antrieb folgt nicht mehr dem Leitwinkel.



Ausgänge ⇒ (Variablentyp: VAR_OUTPUT)				
Bezeichner	Datentyp	Signaltyp	Wert/Bedeutung	
bAccDec	Bool	binary	Beschleunigung/Verzögerung (Königswelle intern)	
			FALSE	Keine Beschleunigung/Verzögerung.
			TRUE	Königswelle wird auf den Leitwinkel synchronisiert (beschleunigt bzw. verzögert).

Parameter (Variablentyp: VAR_CONSTANT_RETAIN)				
Bezeichner	Datentyp	Signaltyp	Info	
Codestelle*			Einstellmöglichkeiten	Voreinstellung
adnTi[0..2]	Array of Double Integer	-	Rampen für den Ein-/Auskuppelvorgang (bezogen auf 4000 rpm)	
C1412/1			adnTi[0] = Verzögerungsrampe für Zwangsöffnung (bOpenInstant = TRUE)	
			0.010	{0.001 s} 130.000 1.000 s
C1412/2			adnTi[1] = Beschleunigungsrampe für Einkuppeln (bClose = TRUE)	
			0.010	{0.001 s} 130.000 1.000 s
C1412/3			adnTi[2] = Verzögerungsrampe für Auskuppeln (bClose = FALSE)	
			0.010	{0.001 s} 130.000 1.000 s
* Codestelle der Erstinstanz				

* Codestelle der Erstinstanz

Verschaltung

Verbinden Sie zur Positionsbestimmung der X-Achse (Leitantrieb) den Eingang *dnActXLength_p* mit dem gleichnamigen Ausgang *dnActXLength_p* des FB **L_CamData**:

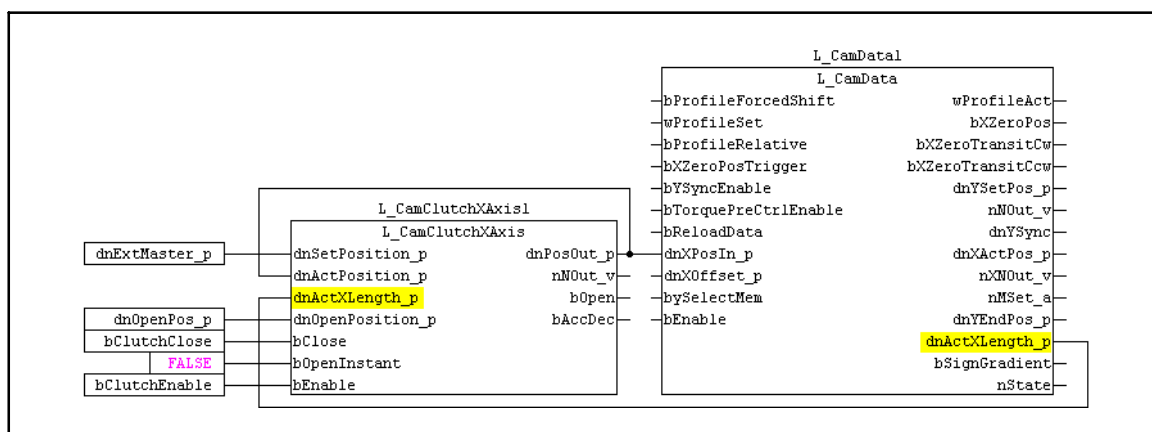


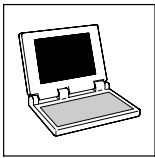
Abb. 2-2

Verschaltung zur Positionsbestimmung



Hinweis!

Bei Kurvenumschaltung zwischen Kurven mit unterschiedlicher X-Achsenlänge darf sich der Leitwinkel nur in positive Richtung verdrehen.



Funktionsbibliothek LenzeCamControl.lib

Funktionsblöcke

Kupplungsfunktion aktivieren

Die Kupplungsfunktion wird mit *bEnable* = TRUE aktiviert, das Verhalten der Kupplungsfunktion ist vom Eingang *bClose* abhängig:

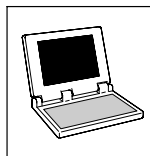
- *bClose* = TRUE: Anforderung die Kupplung zu schliessen.
- *bClose* = FALSE: Anforderung die Kupplung zu öffnen.



Tipp!

Mit *bEnable* = FALSE gilt die Kupplung als geschlossen.

- Der Leitwinkel der Königswelle wird direkt durchgeschaltet. Es erfolgt keine Führung über Rampen (die Ausgänge folgen dem Eingang *dnSetPosition_p*).
- Hiermit lässt sich auch die Funktion "direktes Einkuppeln" realisieren.



Einkuppeln über Rampenfunktion mit anschließendem Synchronlauf

Nach einem – zumeist prozessbedingten – Pausieren kann der Kurvenscheibenantrieb bei durchlaufender Königswelle (Leitantrieb) mit Hilfe dieses FBs wieder an den Prozess angekuppelt werden.

Zum Zeitpunkt des Einkuppelns (*bOpen* = FALSE) haben interne und externe Königswelle zueinander sowohl synchrone Drehzahlen als auch gleiche Positionen.

Der Einkuppelvorgang wird mit *bClose* = TRUE eingeleitet. Der Synchronisiervorgang erfolgt daraufhin automatisch, wie folgende Abbildung zeigt:

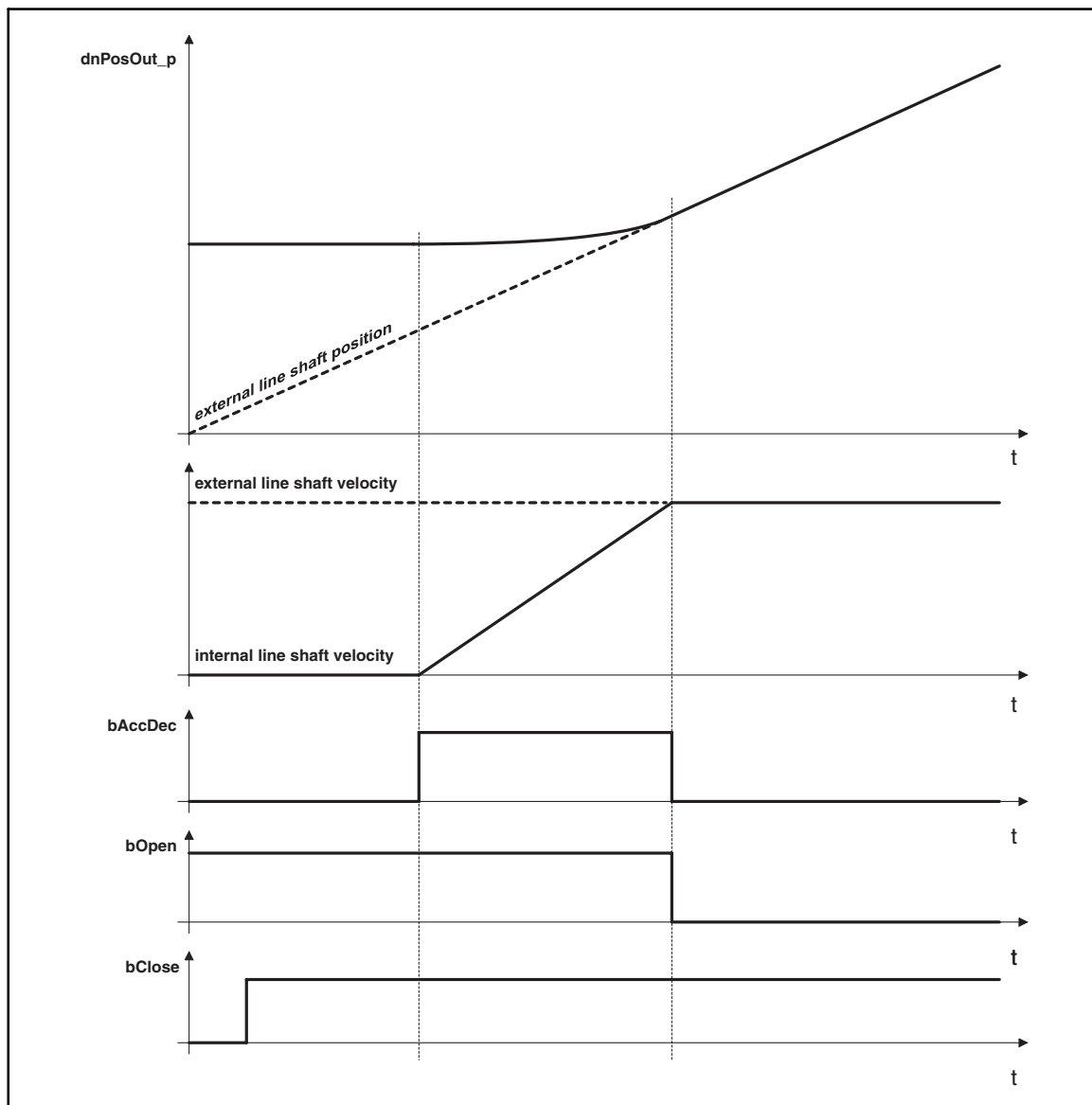
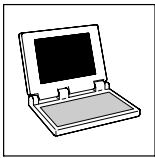


Abb. 2-3

Einkuppeln über Rampenfunktion



Funktionsbibliothek LenzeCamControl.lib

Funktionsblöcke

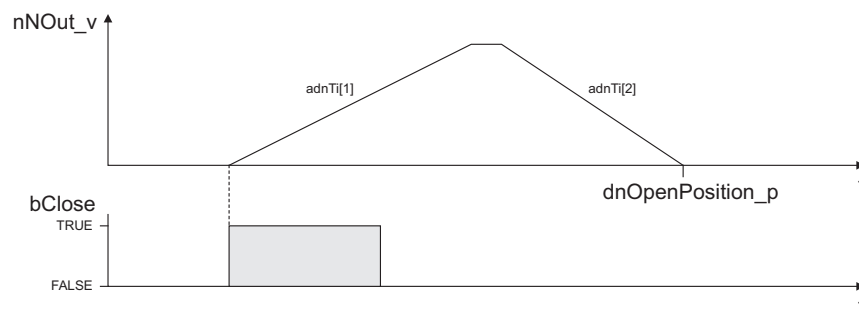
1. Der FB berechnet den Startzeitpunkt, um mit der eingestellten Rampe *adnTi[1]* in den Synchronpunkt zu laufen.
 - Abhängig von der eingestellten Rampe kann der Einkuppelvorgang auch über mehrere Takte andauern.
 - Ist die Positionsänderung pro Zyklus an *dnSetPosition_p* so gering, dass keine Beschleunigungsrampe berechnet werden kann, wird am Synchronpunkt "hart" eingekuppelt.
2. Die interne Königswelle wird beschleunigt, sobald die Startposition erreicht ist. Während des Beschleunigungsvorgangs ist *bAccDec* auf TRUE gesetzt.
3. Ist der Synchronpunkt erreicht, werden *bAccDec* und *bOpen* auf FALSE gesetzt.



Hinweis!

Der Eingang *bClose* sollte solange auf TRUE gesetzt bleiben, solange die interne Königswelle synchron zur externen Königswelle laufen soll.

Wird *bClose* vor Beendigung des Einkuppelvorgangs wieder auf FALSE zurückgesetzt, wird die interne Königswelle trotzdem vollständig eingekuppelt und erst anschließend wieder über die Rampenfunktion in die Position *dnOpenPosition_p* ausgekuppelt:



Auskuppeln über Rampenfunktion

Beim Auskuppeln wird die interne Königswelle von der externen Königswelle getrennt.

Der Auskuppelvorgang wird eingeleitet, indem *bClose* wieder auf FALSE zurückgesetzt wird:

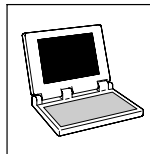
1. Die Kupplung wird geöffnet.
2. Über die Verzögerungsrampe *adnTi[2]* wird abgebremst.
 - Abhängig von der eingestellten Rampe kann der Auskuppelvorgang auch über mehrere Takte andauern.
3. An *dnOpenPosition_p* wird angehalten.

Sofortiges Auskuppeln (Zwangsöffnung)

Diese Funktion entspricht quasi der einer mechanischen Kupplung, die interne und externe Königswelle werden sofort voneinander getrennt.

Der Auskuppelvorgang wird eingeleitet, indem *bOpenInstant* auf TRUE gesetzt wird:

1. Der Ausgang *bOpen* wird auf TRUE gesetzt.
2. Die interne Königswelle wird über die Verzögerungsrampe *adnTi[0]* in den Stillstand geführt. Es erfolgt keine Positionierung auf *dnOpenPosition_p*.
3. Bis zum Stillstand der internen Königswelle ist *bAccDec* auf TRUE gesetzt.



2.2 L_CamContactor - Nockenschaltwerk

Über diesen FB kann ein Nockenschaltwerk mit folgenden Modi realisiert werden:

- Wegnocken in positiver Wirkrichtung
- Wegnocken in negativer Wirkrichtung
- Wegnocken mit beiden Wirkrichtung
- Weg-Zeitnocken in positiver Wirkrichtung
- Weg-Zeitnocken in negativer Wirkrichtung

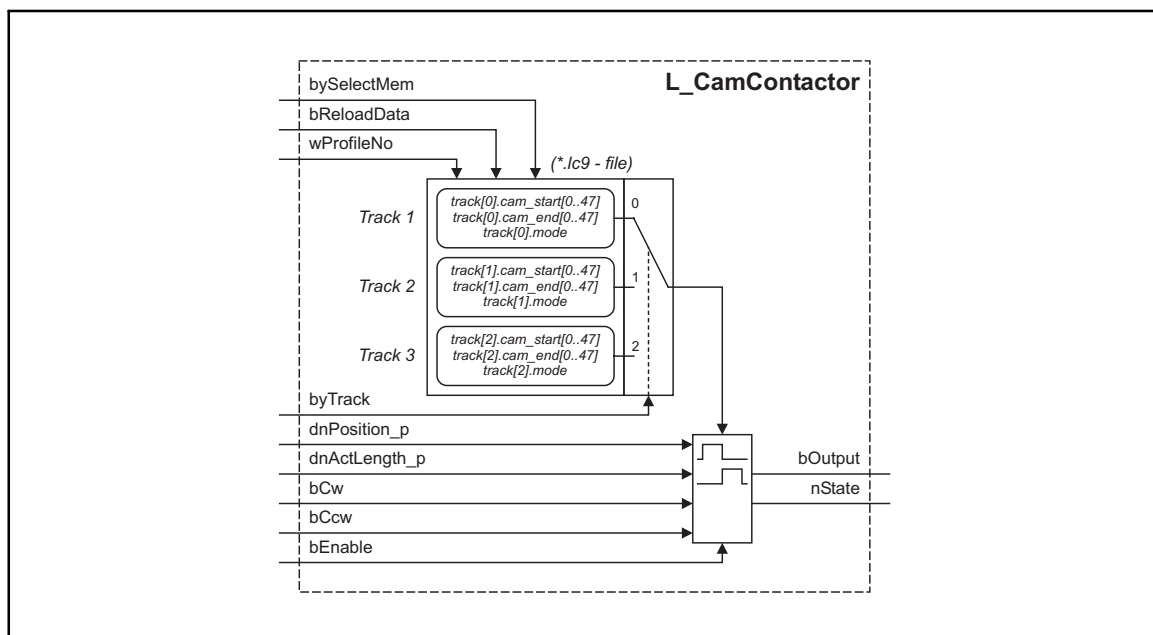
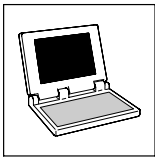


Abb. 2-4 Funktionsblock "L_CamContactor"

⇒ Eingänge (Variablentyp: VAR_INPUT)			
Bezeichner	Datentyp	Signaltyp	Info/Einstellmöglichkeiten
bySelectMem	Byte	number	Auswahl der Datenstruktur (*.lc9-Datei) im Speicher Zur Übernahme einer neuen Auswahl ist eine FALSE-TRUE-Flanke an <i>bEnable</i> oder an <i>bReloadData</i> erforderlich.
			1 RAM-Speicherblock 1 (Vordergrunddatenfeld, Standard)
			2 RAM-Speicherblock 2 (Hintergrunddatenfeld)
bReloadData	Bool	binary	Ausgewählte Datenstruktur übernehmen
			FALSE → TRUE Die über <i>bySelectMem</i> ausgewählte Datenstruktur wird übernommen und die Nocken werden initialisiert.
wProfileNo	Word	-	Auswahl des Profils
			0 ... 47 Verwendung der Daten von Profil Nr. 0 ... 47
byTrack	Byte	-	Auswahl der Spur
			0 ... 2 Spur 0 ... 2
			Anderer Wert Spur 0; <i>nFailNumber</i> wird auf "2" gesetzt.
dnPosition_p	Double Integer	position	Achsenposition
dnActLength_p	Double Integer	position	Länge der X- bzw. Y-Achse
			0 lineare Achsen
			Taktlänge zyklische Achsen



Funktionsbibliothek LenzeCamControl.lib

Funktionsblöcke

L_CamContactor - Nockenschaltwerk

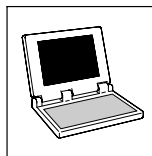
⇒ Eingänge (Variablentyp: VAR_INPUT)							
Bezeichner	Datentyp	Signaltyp	Info/Einstellmöglichkeiten				
bCw	Bool	binary	Drehrichtungsvorgabe Verbinden Sie diesen Eingang mit dem gleichnamigen Ausgang <i>bCw</i> des FB L_CamContactorVPos. <table><tr><td>TRUE</td><td>positiv (im Uhrzeigersinn)</td></tr></table>	TRUE	positiv (im Uhrzeigersinn)		
TRUE	positiv (im Uhrzeigersinn)						
bCcw	Bool	binary	Drehrichtungsvorgabe Verbinden Sie diesen Eingang mit dem gleichnamigen Ausgang <i>bCcw</i> des FB L_CamContactorVPos. <table><tr><td>TRUE</td><td>negativ (gegen den Uhrzeigersinn)</td></tr></table>	TRUE	negativ (gegen den Uhrzeigersinn)		
TRUE	negativ (gegen den Uhrzeigersinn)						
bEnable	Bool	binary	Funktionsblock freischalten <table><tr><td>FALSE</td><td>Funktionsblock gesperrt Intern werden keine Funktionen ausgeführt, beim Zeitnoken wird eine ggf. gestartete Zeit zurückgesetzt. </td></tr><tr><td>TRUE</td><td>Funktionsblock freigeschaltet (Nockenschaltwerk aktiv)</td></tr></table>	FALSE	Funktionsblock gesperrt Intern werden keine Funktionen ausgeführt, beim Zeitnoken wird eine ggf. gestartete Zeit zurückgesetzt.	TRUE	Funktionsblock freigeschaltet (Nockenschaltwerk aktiv)
FALSE	Funktionsblock gesperrt Intern werden keine Funktionen ausgeführt, beim Zeitnoken wird eine ggf. gestartete Zeit zurückgesetzt.						
TRUE	Funktionsblock freigeschaltet (Nockenschaltwerk aktiv)						

Ausgänge ⇨ (Variablentyp: VAR_OUTPUT)																	
Bezeichner	Datentyp	Signaltyp	Wert/Bedeutung														
bOutput	Bool	binary	Zustand des Nockens, abhängig vom ausgewählten Datensatz.														
nState	Integer	-	<table><tr><td colspan="2">Status</td></tr><tr><td>10</td><td>Initialisierung über zwei Taktzyklen aktiv.</td></tr><tr><td>2</td><td>Warnung: <i>byTrack</i> außerhalb des gültigen Bereiches 0 ... 2. - Es werden die Daten der ersten Spur [0] verwendet. </td></tr><tr><td>1</td><td>Warnung: <i>wProfileNo</i> außerhalb des gültigen Bereiches 0 ... 47. - Es werden die Daten des ersten Profils [0] verwendet. </td></tr><tr><td>0</td><td>OK, kein Fehler</td></tr><tr><td>-1</td><td>Fehler: Keine Daten oder ungültiger Modus in der Datenstruktur eingetragen. - Alle Ausgänge sowie gestartete Zeiten werden zurückge- setzt. </td></tr><tr><td>-2</td><td>Fehler: Keine internen Daten verfügbar. - Dieser FB kann nur in Verbindung mit der Kurvenscheiben- Technologie eingesetzt werden. </td></tr></table>	Status		10	Initialisierung über zwei Taktzyklen aktiv.	2	Warnung: <i>byTrack</i> außerhalb des gültigen Bereiches 0 ... 2. Es werden die Daten der ersten Spur [0] verwendet.	1	Warnung: <i>wProfileNo</i> außerhalb des gültigen Bereiches 0 ... 47. Es werden die Daten des ersten Profils [0] verwendet.	0	OK, kein Fehler	-1	Fehler: Keine Daten oder ungültiger Modus in der Datenstruktur eingetragen. Alle Ausgänge sowie gestartete Zeiten werden zurückge- setzt.	-2	Fehler: Keine internen Daten verfügbar. Dieser FB kann nur in Verbindung mit der Kurvenscheiben- Technologie eingesetzt werden.
Status																	
10	Initialisierung über zwei Taktzyklen aktiv.																
2	Warnung: <i>byTrack</i> außerhalb des gültigen Bereiches 0 ... 2. Es werden die Daten der ersten Spur [0] verwendet.																
1	Warnung: <i>wProfileNo</i> außerhalb des gültigen Bereiches 0 ... 47. Es werden die Daten des ersten Profils [0] verwendet.																
0	OK, kein Fehler																
-1	Fehler: Keine Daten oder ungültiger Modus in der Datenstruktur eingetragen. Alle Ausgänge sowie gestartete Zeiten werden zurückge- setzt.																
-2	Fehler: Keine internen Daten verfügbar. Dieser FB kann nur in Verbindung mit der Kurvenscheiben- Technologie eingesetzt werden.																



Hinweise!

- Der FB muss in einer zyklischen Task aufgerufen werden, ein Aufruf in einer ereignisgesteuerten Task oder in **PLC_PRG** ist nicht zulässig.
- Wird zur Drehrichtungsvorgabe nicht der FB **L_CamContactorVPos** verwendet, muss sichergestellt werden, dass die Eingangssignale an *bCw* und *bCcw* der tatsächlichen Drehrichtung entsprechen, z. B. dürfen beide Eingänge nicht auf TRUE gesetzt sein.



Drehrichtungsvorgabe

Zur Drehrichtungsvorgabe verschalten Sie die Eingänge *bCw* und *bCcw* mit den entsprechenden Ausgängen des FB **L_CamContactorVPos**.

- Beachten Sie hierbei, dass der FB **L_CamContactorVPos** zwei Zyklen für die Drehrichtungserkennung benötigt, um hierdurch ein Toggeln des Nockens zu verhindern. Somit wird z. B. der Ausgang *bCw* des FB **L_CamContactorVPos** erst auf TRUE gesetzt, nachdem mindestens zweimal hintereinander Rechtsdrehung erkannt wurde.



Tip!

Wenn sichergestellt ist, dass sich die Achse nur in die entsprechende Richtung dreht und die Nockendaten mit dem **CamDesigner** generiert wurden, können Sie die Drehrichtungseingänge *bCw* und *bCcw* auch direkt beschalten.

Eingabe im CamDesigner

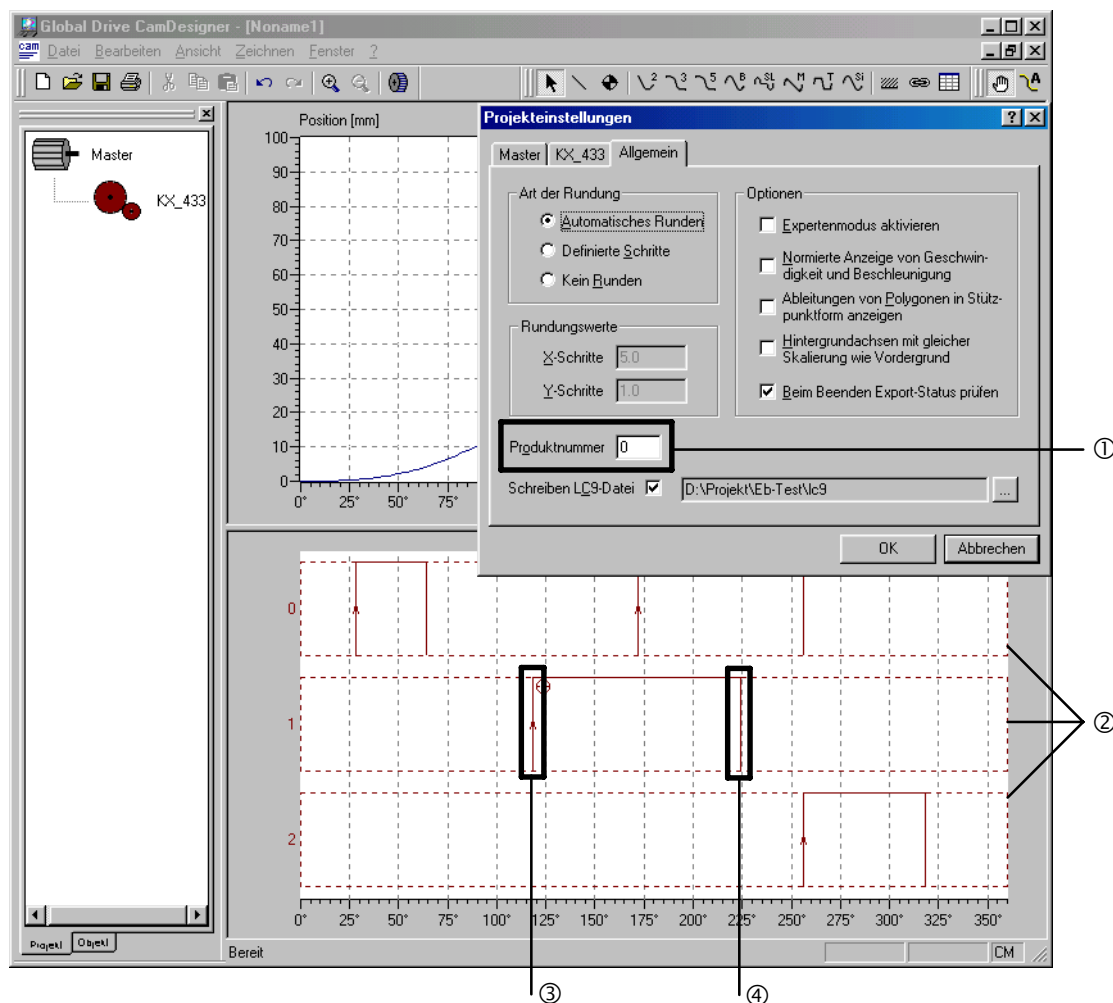
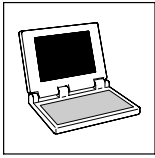


Abb. 2-5

Beispiel: Ansicht im CamDesigner

- ① Angabe der Produktnummer bzw. Profilnummer *wProductNo* (0 ... 47)
- ② Darstellung der drei Nockenspuren (*byTrack*)
- ③ Anfang eines Nockens
- ④ Ende eines Nockens



Funktionsbibliothek LenzCamControl.lib

Funktionsblöcke

L_CamContactor - Nockenschaltwerk

Modi für das Nockenschaltwerk

Für die Festlegung eines Nockens stehen fünf Modi zur Verfügung:



Hinweis!

Zwei aufeinanderfolgende Nocken werden sicher als einzelne Nocken erkannt, wenn ihr Abstand zueinander größer ist als die Positionsänderung pro Taktzyklus. Dadurch wird verhindert, dass der Zustand "Aus" zwischen den Nocken übersprungen wird.

Für Nocken mit sehr kurzer Einschaltdauer muss die Einschaltdauer größer als die Positionsänderung pro Taktzyklus sein, damit der Nocken für mindestens einen Taktzyklus geschaltet wird.

Wegnocken in positiver Wirkrichtung



Beim Erreichen des Referenzwertes *cam_start* und Veränderung in positiver Richtung des Positionseingangs *dnPosition_p* wird der Ausgang *bOutput* auf TRUE gesetzt und beim Überschreiten des Wertes von *cam_end* wieder auf FALSE zurückgesetzt.

- Führt die Achse nach Erreichen von *cam_start* wieder zurück, wird der Ausgang *bOutput* sofort auf FALSE zurückgesetzt.

Legende:

cam_start : Anfang des Nockens

cam_end : Ende des Nockens

▲ / ▼ : flankengesteuert

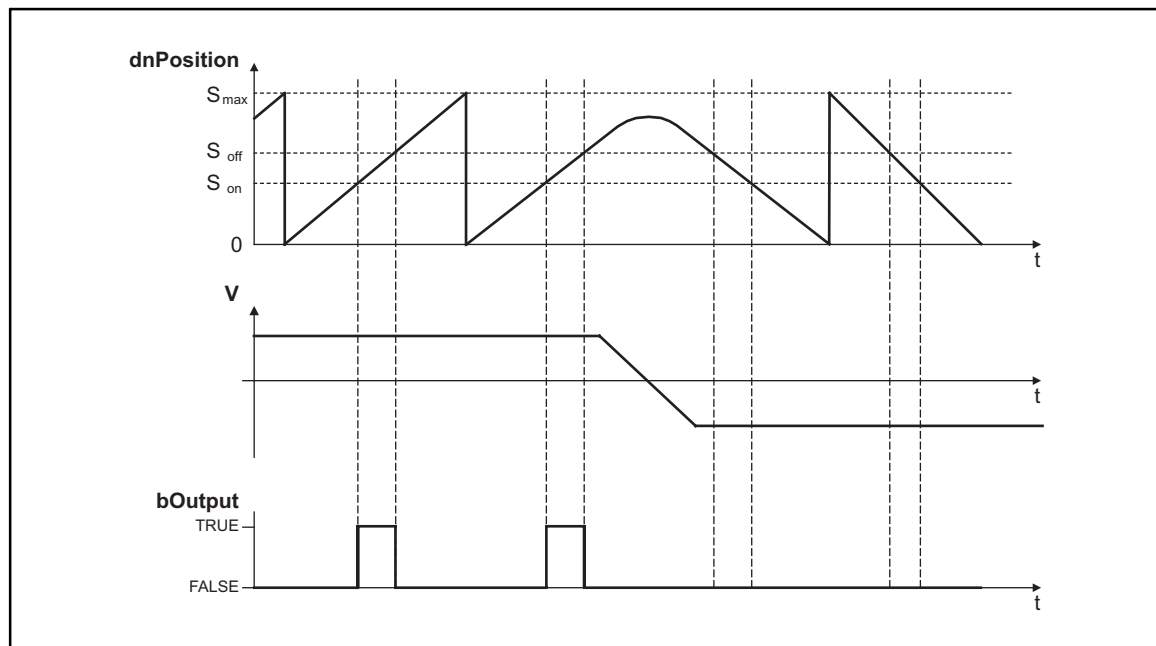
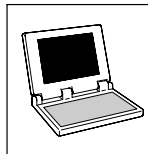
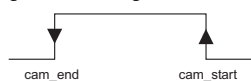


Abb. 2-6

Wegnocken in positiver Wirkrichtung



Wegnocken in negativer Wirkrichtung



Beim Erreichen des Referenzwertes *cam_end* und Veränderung in negativer Richtung des Positionseingangs *dnPosition_p* des Signals wird der Ausgang *bOutput* auf TRUE gesetzt und beim Überschreiten des Wertes von *cam_start* wieder auf FALSE zurückgesetzt.

- Führt die Achse nach Erreichen von *cam_end* wieder zurück, wird der Ausgang *bOutput* sofort auf FALSE zurückgesetzt.

Legende:

cam_start : Anfang des Nockens

cam_end : Ende des Nockens

▲ / ▼ : flankengesteuert

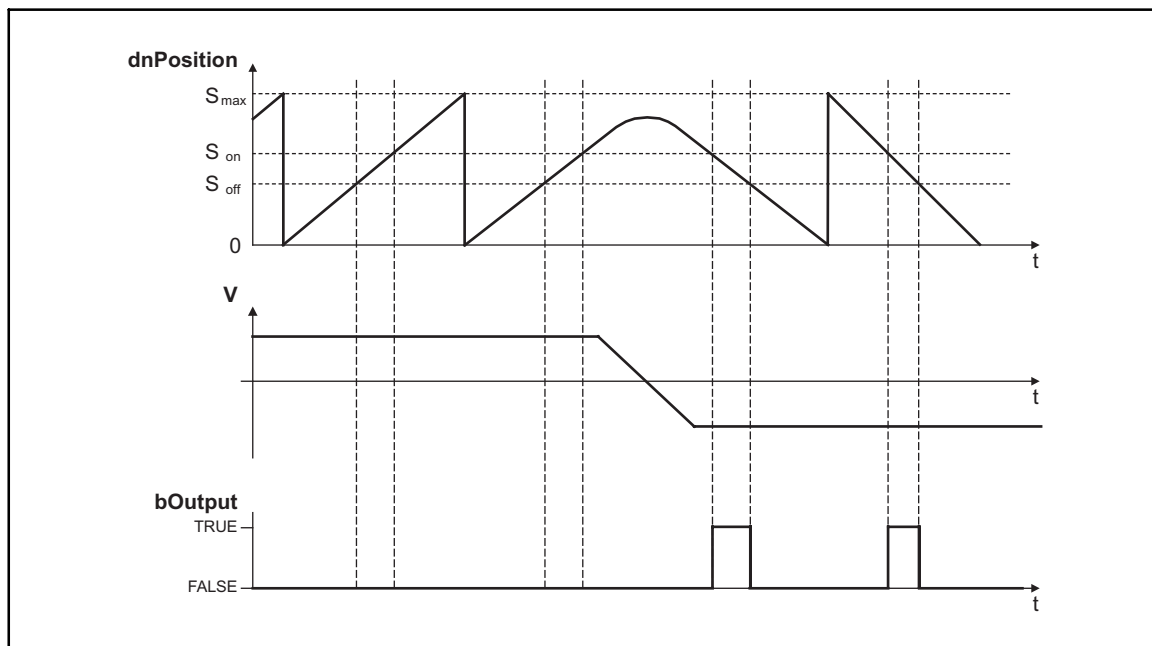
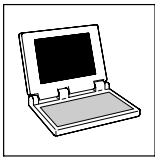


Abb. 2-7

Wegnocken in negativer Wirkrichtung



Funktionsbibliothek LenzeCamControl.lib

Funktionsblöcke

L_CamContactor - Nockenschaltwerk

Wegnocken mit bipolarer Wirkrichtung



Innerhalb der Referenzwerte *cam_start* und *cam_end* wird der Ausgang *bOutput* auf TRUE gesetzt.

- Die Durchlaufrichtung des Signals spielt in diesem Modus keine Rolle.

Legende:

cam_start : Anfang des Nockens

cam_end : Ende des Nockens

▲ / ▼ : flankengesteuert

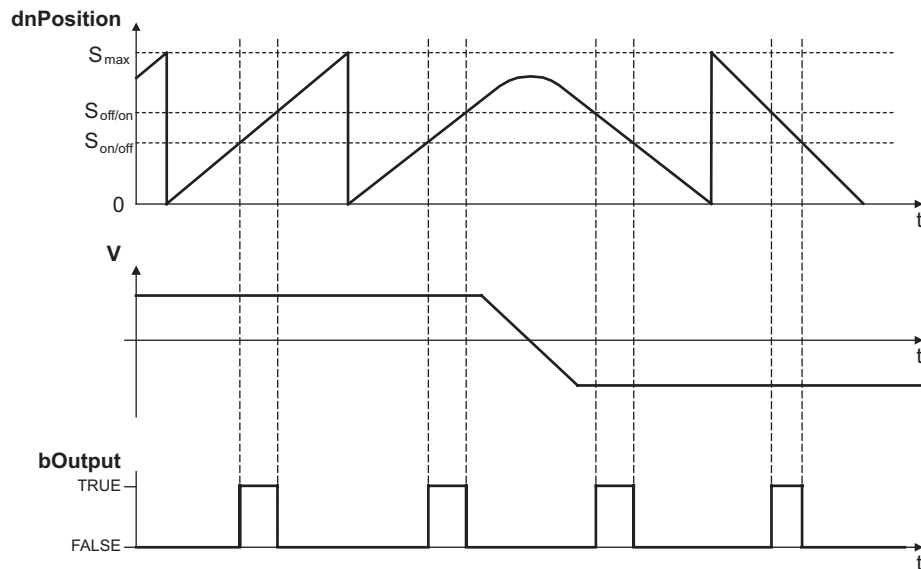
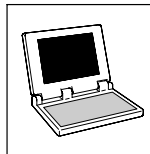


Abb. 2-8

Wegnocken in bipolarer Wirkrichtung

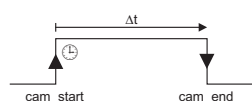


Hinweis!

Beim Einsatz von Zeitnocken kann es zu überlappenden Einschaltzuständen von aufeinanderfolgenden Nocken kommen, wenn die Einschaltdauer des Zeitnocken größer ist als der zeitliche Abstand zur nächsten Einschaltposition des folgenden Zeitnocken. (Nachtrigger-Funktion)

Ist die Einschaltdauer eines Zeitnockens länger als die gesamte Taktzeit, bleibt das Signal des Zeitnocken dauerhaft eingeschaltet.

Weg-Zeitnocken in positiver Wirkrichtung



Legende:

Beim Erreichen des Referenzwertes cam_start und Veränderung in positiver Richtung des Positionseingangs $dnPosition_p$ wird der Ausgang $bOutput$ auf TRUE gesetzt und nach Ablauf der in cam_end festgelegten Zeit (in Millisekunden) wieder auf FALSE zurückgesetzt.

- Ist der Nocken gesetzt, hat eine darauffolgende Drehrichtungsänderung keinen Einfluss auf das Nockensignal.
- Ein gesetzter Zeitnocken wird nur nach Ablauf der für sie gesetzten Zeitdauer zurückgesetzt.

cam_start : Anfang des Nockens

cam_end : Ende des Nockens

▲ / ▼ : flankengesteuert

⌚ : zeitgesteuert (Δt = Einschaltzeit in [ms])

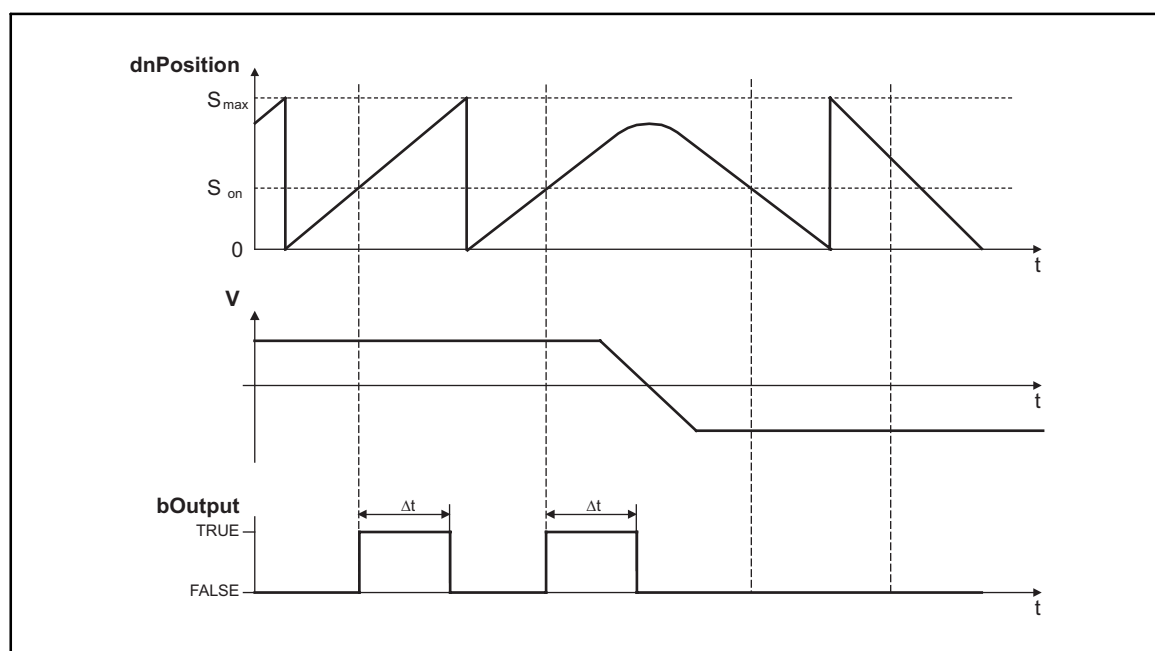
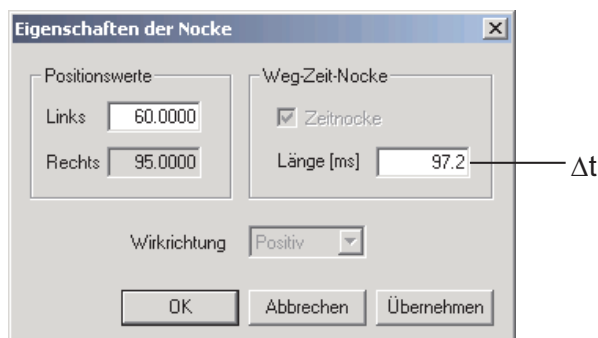
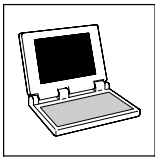


Abb. 2-9

Weg-Zeitnocken in positiver Wirkrichtung



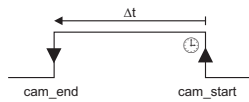


Funktionsbibliothek LenzeCamControl.lib

Funktionsblöcke

L_CamContactor - Nockenschaltwerk

Weg-Zeitnocken in negativer Wirkrichtung



Beim Erreichen des Referenzwertes *cam_start* und Veränderung in negativer Richtung des Positionseingangs *dnPosition_p* des Signals wird der Ausgang *bOutput* auf TRUE gesetzt und nach Ablauf der in *cam_end* festgelegten Zeit (in Millisekunden) wieder auf FALSE zurückgesetzt.

- Ist der Nocken gesetzt, hat eine darauffolgende Drehrichtungsänderung keinen Einfluss auf das Ausgangssignal.
- Ein gesetzter Zeitnocken wird nur nach Ablauf der für sie gesetzten Zeitdauer zurückgesetzt.

Legende:

cam_start : Anfang des Nockens

cam_end : Ende des Nockens

▲ / ▼ : flankengesteuert

⌚ : zeitgesteuert (Δt = Einschaltzeit in [ms])

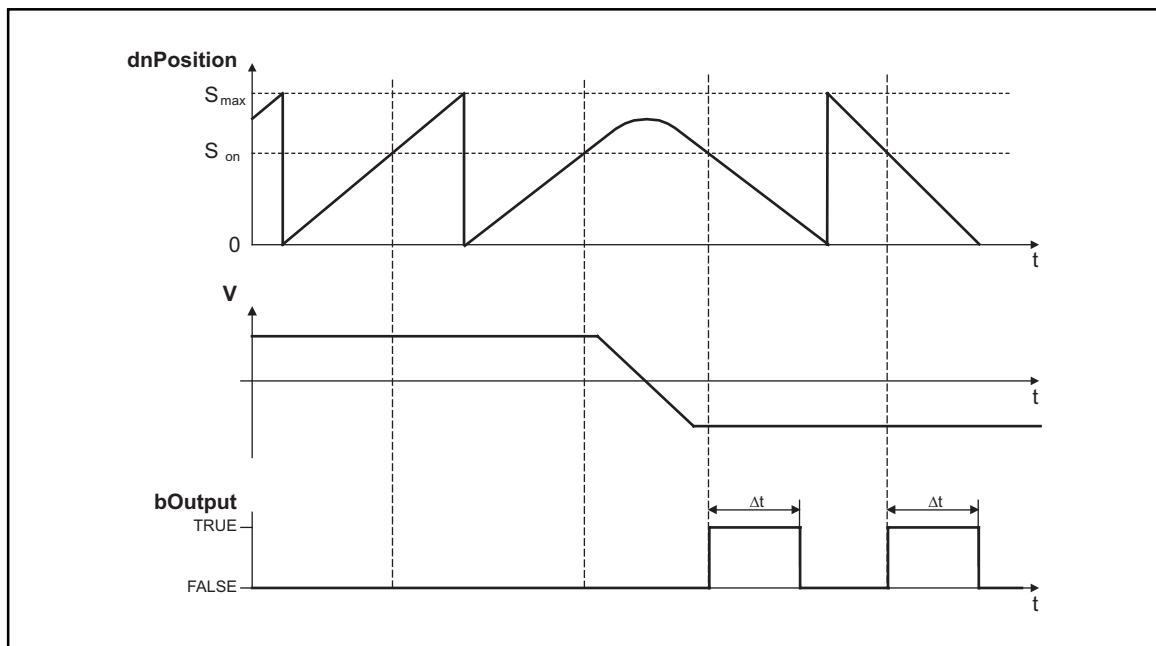
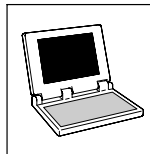
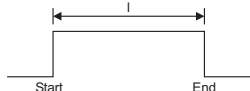
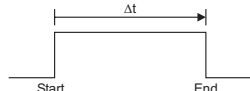


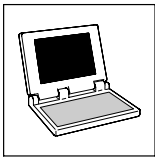
Abb. 2-10

Weg-Zeitnocken in negativer Wirkrichtung



Gegenüberstellung Wegnocken/Zeitnocken

	Wegnocken	Zeitnocken
	 l = Nockenlänge ($cam_end - cam_start$)	 Δt = Einschaltzeit (cam_end) in [ms]
Wirkrichtung	Zwei Wirkrichtungen sind möglich: • Positiv: Der Nocken wird am Nockenanfang eingeschaltet, wenn die Achse in Richtung steigender Wert des Positionseinganges $dnPosition_p$ läuft. • Negativ: Der Nocken wird am Nockenende eingeschaltet, wenn die Achse in Richtung fallender Wert des Positionseinganges $dnPosition_p$ läuft. • Es können auch beide Wirkrichtungen eingeschaltet werden.	Zwei Wirkrichtungen sind möglich: • Positiv (Cw): Der Nocken wird am Nockenanfang eingeschaltet, wenn die Achse in Richtung steigender Wert des Positionseinganges $dnPosition_p$ läuft. • Negativ (Ccw): Der Nocken wird am Nockenanfang eingeschaltet, wenn die Achse in Richtung fallender Wert des Positionseinganges $dnPosition_p$ läuft.
Einschalten	Der Nocken schaltet ein: • Am Nockenanfang, wenn die Bewegungsrichtung der Achse positiv ist und positive Wirkrichtung eingestellt ist. • Am Nockenende, wenn die Bewegungsrichtung der Achse negativ ist und negative Wirkrichtung eingestellt ist. • Wenn der Wert des Positionseinganges $dnPosition_p$ im Bereich des Nockens liegt.	Der Nocken schaltet ein: • Am Nockenanfang, wenn die Bewegungsrichtung der Achse positiv ist und positive Wirkrichtung eingestellt ist. • Am Nockenanfang, wenn die Bewegungsrichtung der Achse negativ ist und negative Wirkrichtung eingestellt ist. Nach dem Einschalten läuft die volle Nockeneinschaltzeit ab. • Die Einschaltzeit läuft auch dann ab, wenn nach dem Einschalten des Nockens die Bewegungsrichtung der Achse verändert wird. • Wenn während der Einschaltzeit eines Nockens dessen Nockenanfang erneut überfahren wird, wird der Nocken nachgetriggert.
Ausschalten	Der Nocken schaltet aus, wenn • der Wert des Positionseinganges $dnPosition_p$ nicht mehr im Bereich des Nockens liegt. • die Wirkrichtung entgegen der Bewegungsrichtung der Achse erkannt ist.	Der Nocken schaltet aus, wenn die parametrisierte Zeit abgelaufen ist.
Weglänge	Die Weglänge des Nockens ist durch Nockenanfang und Nockenende definiert.	Die Einschaltweglänge des Nockens ist abhängig von der Geschwindigkeit, mit der die Achse während der Einschaltzeit des Nockens verfährt.
Einschaltdauer	Die Einschaltdauer des Nockens ist abhängig von der Geschwindigkeit, mit der die Achse die Weglänge des Nockens überfährt.	Die Einschaltdauer des Nockens ist durch die Einschaltzeit definiert.
Hinweise	Der Start- und Endwert des Nockensbereiches wird durch den Nocken-Editor vom CamDesigner festgelegt.	Die kleinste mögliche Zeit für den Zeitnocken ist immer größer bzw. gleich der eingestellten Taskzeit.



Funktionsbibliothek LenzeCamControl.lib

Funktionsblöcke

L_CamContactorVPos - Dynamische Verstellung der Nocken/Hysteresese

2.3

L_CamContactorVPos - Dynamische Verstellung der Nocken/Hysteresese

Mit diesem FB können Sie zur Realisierung eines Nockenschaltwerkes den aktuellen Positions- wert über eine Hysteresese bewerten sowie eine Totzeitpunktberechnung durchführen.

Des Weiteren kann mit diesem FB die Totzeit angeschlossener Elemente Geschwindigkeits- bzw. Drehzahl-abhängig kompensiert werden.

Die Berechnung der Nockenschaltpunkte erfolgt im FB **L_CamContactor**.

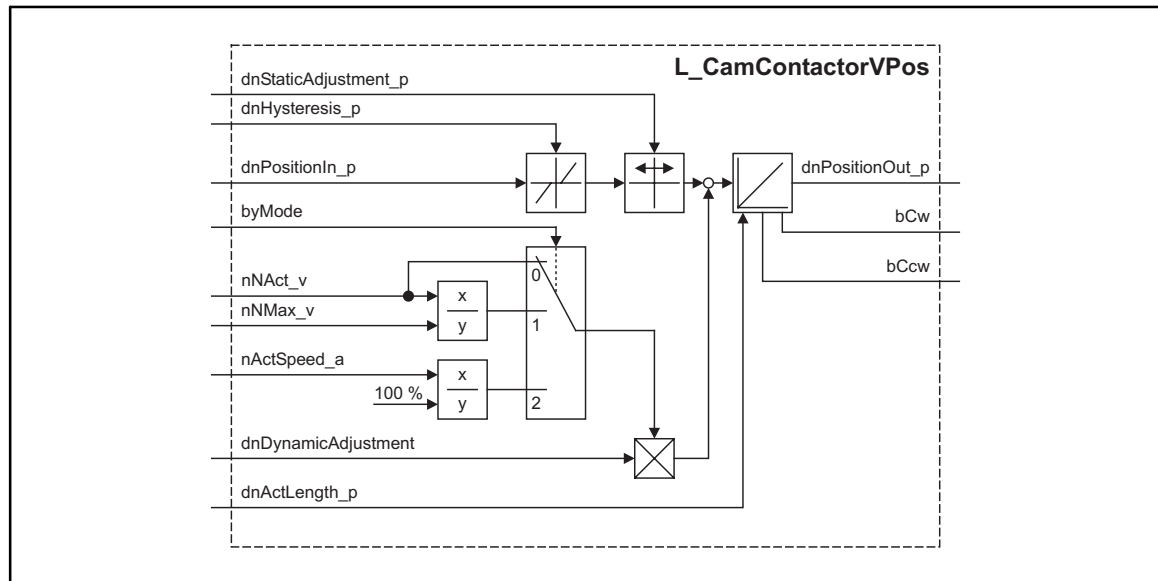
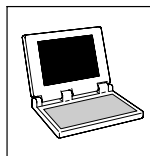


Abb. 2-11

Funktionsblock "L_CamContactorVPos"

⇒ Eingänge (Variablentyp: VAR_INPUT)									
Bezeichner	Datentyp	Signaltyp	Info/Einstellmöglichkeiten						
dnStaticAdjustment_p	Double Integer	position	Statische Verstellung (Kompensation vom Verzögerungsweg angeschlossener Schaltelemente). Dieser Wert wird zu <i>dnPositionIn_p</i> aufaddiert.						
dnHysteresis_p	Double Integer	position	Hysteresese Um diesen Wert muss sich der Eingangswert <i>dnPositionIn_p</i> minimal ändern, damit der Ausgangswert <i>dnPositionOut_p</i> aktualisiert wird.						
dnPositionIn_p	Double Integer	position	Ist-Position						
byMode	Byte	-	Auswahl des Modus zur Vorgabe der dynamischen Verschiebung <table><tr><td>0</td><td>Zeitvorgabe in [ms] an <i>dnDynamicAdjustment</i>. - Drehzahleingang <i>nNAct_v</i> verwenden. </td></tr><tr><td>1</td><td>Positionsvorgabe in [inc] an <i>dnDynamicAdjustment</i>. - Nur Drehzahleingänge <i>nNAct_v</i> und <i>nNMax_v</i> verwenden. </td></tr><tr><td>2</td><td>Positionsvorgabe in [inc] an <i>dnDynamicAdjustment</i>. - Nur Drehzahleingang <i>nActSpeed_a</i> verwenden. </td></tr></table>	0	Zeitvorgabe in [ms] an <i>dnDynamicAdjustment</i> . Drehzahleingang <i>nNAct_v</i> verwenden.	1	Positionsvorgabe in [inc] an <i>dnDynamicAdjustment</i> . Nur Drehzahleingänge <i>nNAct_v</i> und <i>nNMax_v</i> verwenden.	2	Positionsvorgabe in [inc] an <i>dnDynamicAdjustment</i> . Nur Drehzahleingang <i>nActSpeed_a</i> verwenden.
0	Zeitvorgabe in [ms] an <i>dnDynamicAdjustment</i> . Drehzahleingang <i>nNAct_v</i> verwenden.								
1	Positionsvorgabe in [inc] an <i>dnDynamicAdjustment</i> . Nur Drehzahleingänge <i>nNAct_v</i> und <i>nNMax_v</i> verwenden.								
2	Positionsvorgabe in [inc] an <i>dnDynamicAdjustment</i> . Nur Drehzahleingang <i>nActSpeed_a</i> verwenden.								
nNAct_v	Integer	velocity	Istdrehzahl der X- oder Y-Achse in [inc/ms] - Nur relevant bei <i>byMode</i> = 0 oder 1 - An diesen Eingang kann z. B. das Geschwindigkeitssignal <i>nXNOut_v</i> vom FB L_CamData angelegt werden.						
nNMax_v	Integer	velocity	Bemessungstaktzahl in [inc/ms] - Nur relevant bei <i>byMode</i> = 1 - An diesen Eingang kann z. B. das Signal <i>g_CamAxis.wXCycles</i> angelegt werden.						
nActSpeed_a	Integer	analog	Istdrehzahl der X- oder Y-Achse 16384 ≙ 100 % - Nur relevant bei <i>byMode</i> = 2 - An diesen Eingang kann das Signal <i>MCTRL_nNAct_a</i> vom SB MCTRL angelegt werden.						



⇒ Eingänge (Variablentyp: VAR_INPUT)			
Bezeichner	Datentyp	Signaltyp	Info/Einstellmöglichkeiten
dnDynamicAdjustment	Double Integer	position	Dynamische Verstellung von Verschiebungswerten angeschlossener Nocken - Bei <i>byMode</i> = 0: - Vorgabe der Verschiebung in [ms]. - Bei <i>byMode</i> = 1 oder 2: - Vorgabe der Verschiebung in [inc] bei maximaler Geschwindigkeit (Drehzahl). Der Verschiebungswert kann in allen Modi auch negativ vorgegeben werden. Dabei wird in die andere Richtung verschoben (Verschiebung nacheilend).
dnActLength_p	Double Integer	position	Anzahl Inkremente, die bei einer Rundachse für eine Umdrehung benötigt werden. - Der Ausgangswert ist dann immer kleiner als dieser Wert. - Bei Verwendung einer Längsachse tragen Sie hier den Wert "0" ein.

Ausgänge ⇒ (Variablentyp: VAR_OUTPUT)			
Bezeichner	Datentyp	Signaltyp	Wert/Bedeutung
dnPositionOut_p	Double Integer	position	Bewerteter Positionswert (nach Hysterese und Totzeitpunktberechnung) Verbinden Sie diesen Ausgang mit dem Eingang <i>dnPositionIn_p</i> des FB L_CamContactor.
bCw	Bool	-	Drehrichtungsausgabe Verbinden Sie diesen Ausgang mit dem gleichnamigen Eingang <i>bCw</i> des FB L_CamContactor. TRUE positiv (im Uhrzeigersinn)
bCcw	Bool	-	Drehrichtungsausgabe Verbinden Sie diesen Ausgang mit dem gleichnamigen Eingang <i>bCcw</i> des FB L_CamContactor. TRUE negativ (gegen den Uhrzeigersinn)

Drehrichtungserkennung

Die Berechnung der Nockenschaltpunkte erfolgt im FB **L_CamContactor**. Da dieser FB keine interne Drehrichtungserkennung beinhaltet, muss der FB **L_CamContactorVPos** dem FB **L_CamContactor** vorgeschaltet werden:

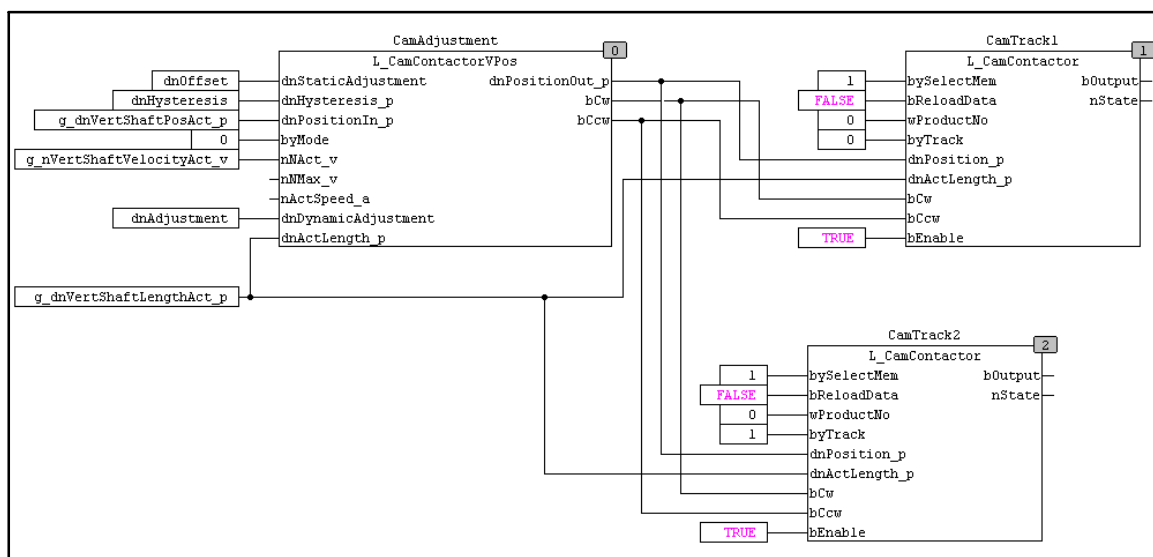
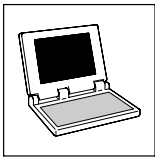


Abb. 2-12

Verschaltungsbeispiel



Funktionsbibliothek LenzeCamControl.lib

Funktionsblöcke

L_CamContactorVPos - Dynamische Verstellung der Nocken/Hysterese



Hinweis!

Bei Verwendung mehrerer Nockenschaltwerke verbinden Sie die Ausgänge *bCw* und *bCcw* mit den entsprechenden Eingängen aller beteiligten FBs **L_CamContactor**, unabhängig davon, ob die Nockenschaltwerke dynamisiert werden sollen oder nicht.

Begrenzung des Positionswertes bei einer Rundachse

Damit bei einer Rundachse keine Positionswerte außerhalb des Bereiches ausgegeben werden können, geben Sie am Eingang *dnActLength_p* als Begrenzungswert die Anzahl der Inkremente für eine Umdrehung vor.

- Für die Länge der X-Achse kann z. B. eine Verknüpfung mit dem Ausgang *dnActXLength_p* des FB **L_CamData** erfolgen.
- Für die Länge der Y-Achse kann z. B. eine Verknüpfung mit dem Ausgang *dnYEndPos_p* des FB **L_CamData** erfolgen.

Bei einer Längsachse (z. B. Y-Achse bei begrenztem Verfahrbereich) legen Sie am Eingang *dnActLength_p* eine "0" an.

Dynamische Verschiebung

Mit Hilfe der dynamischen Verschiebung (*dnDynamicAdjustment*) können Verzögerungszeiten angeschlossener Schaltelemente kompensiert werden.

Für die Vorgabe der Verschiebung stehen folgende Modi zur Verfügung:

byMode	Info	
0	Zeitvorgabe • Verschiebungswert in [ms] über <i>dnDynamicAdjustment</i> vorgeben. • Drehzahl über <i>nNAct_v</i> vorgeben.	Einfache Verschiebung
1	Positionsvorgabe 1 • Verschiebungswert bei maximaler Geschwindigkeit (<i>nNMax_v</i>) über <i>dnDynamicAdjustment</i> vorgeben. • Drehzahl über <i>nNAct_v</i> vorgeben. • Maximale Geschwindigkeit (Drehzahl) über <i>nNMax_v</i> vorgeben.	Vorgabe einer maximalen Geschwindigkeit
2	Positionsvorgabe 2 • Verschiebungswert bei 100 % über <i>dnDynamicAdjustment</i> vorgeben. • Drehzahl über <i>nNAct_v</i> vorgeben (16384 $\equiv$ 100 %).	Verschiebung mit höherer Auflösung (z. B. gut geeignet für Anbindung an <i>MCTRL_nNAct_a</i>)

- Wird ein positiver Wert an *dnDynamicAdjustment* angelegt, eilt die Position am Ausgang *dnPositionOut_p* der Position am Eingang *dnPositionIn_p* vor und die Nocken schalten entsprechend früher.
- Durch die Vorgabe eines negativen Wertes an *dnDynamicAdjustment* können Sie die Nocken dementsprechend später schalten lassen.

Die Verschiebung wirkt sich auf beide Flanken einer Nocke aus:

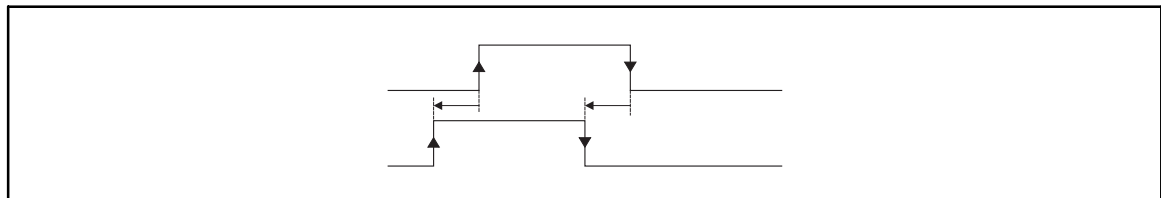
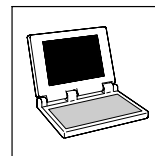


Abb. 2-13

Verschiebung beider Flanken einer Nocke



Beispiel für *bMode* = 2:

Ein angeschlossenes Ventil mit einem festen Verzögerungswert muss bei maximaler Geschwindigkeit 2000 Inkremente vor dem Schaltpunkt geschaltet werden.

- Der Eingang *nActSpeed_a* wird mit dem Ausgang *MCTRL_nNAct_a* des SB **MCTRL** verbunden.
- Die Einstellung "2000" in *dnDynamicAdjustment* hat zur Folge, dass *dnPositionOut_p* bei *nActSpeed_a* = 16384 ($\equiv 100\%$) gegenüber *dnPositionIn_p* um 2000 Inkremente voreilt und dadurch der Ausgang des nachgeschalteten FB **L_CamContactor** entsprechend früher geschaltet wird.
- Die Totzeitkompensation ist linear abhängig von der Geschwindigkeit, d. h. bei halber Geschwindigkeit wird auch nur um 1000 Inkremente früher geschaltet. Bei Stillstand tritt gar keine Verschiebung auf.

Statische Verschiebung

Durch die Vorgabe eines Offset an *dnStaticAdjustment* können zusätzliche Verzögerungsweg ausgeglichen werden.

Beispiel:

An einer Maschine ist im laufenden Betrieb eine Feineinstellung von 50 Inkrementen notwendig.

- Durch die Einstellung "50" in *dnStaticAdjustment* wird *dnPositionOut_p* genau um diesen Offset in positiver Richtung verschoben.

Hysterese

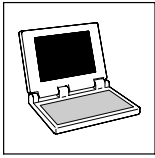
Durch Vorgabe einer Hysterese können Sie ein permanentes Ein-/Ausschalten — ein sogenanntes "Toggeln" — von Nocken unterbinden, das durch Lageistwertänderungen auf Grund einer mechanischen Unruhe an der Achse auftreten kann.

- Das permanente Ein-/Ausschalten von Nocken tritt dann auf, wenn die mechanische Unruhe so groß ist, dass der Lageistwert um eine Flanke eines Nockens oder innerhalb eines aktiven Nockens mit nur einer Wirkrichtung schwankt.
- Die Hysterese stellen Sie über *dnHysteresis_p* ein. Sie ist Istwertabhängig und gilt bei allen Nocken. Sie wirkt auch, wenn kein Nocken an der aktuellen Achsposition gesetzt ist.

Regeln für den Hysteresebereich

Für den Hysteresebereich gelten beim Zusammenschalten der Funktionsblöcke folgende Regeln:

- Innerhalb der Hysterese bleibt die Anzeige des Istwertes konstant.
- Innerhalb der Hysterese wird die Richtung nicht neu bestimmt.
- Innerhalb der Hysterese wird ein Nocken weder ein- noch ausgeschaltet.
- Innerhalb der Hysterese wird ein Zeitnocken nicht eingeschaltet, ein eingeschalteter Zeitnocken wird nach Ablauf der parametrisierten Einschaltzeit ausgeschaltet (nicht erst am Ende der Hysterese).
- Nach Verlassen des Hysteresebereichs bestimmt der FB den Lageistwert, die aktuelle Bewegungsrichtung der Achse und die aktuellen Zustände aller Nocken.
- Der Hysteresebereich ist für alle Nocken gültig.



Funktionsbibliothek LenzeCamControl.lib

Funktionsblöcke

L_CamControl - Sollwertaufbereitung

2.4 L_CamControl - Sollwertaufbereitung

Über diesen FB können folgende Funktionen realisiert werden:

- Integration von Leitfrequenzsignalen zu einer Lageposition
- Laden des Hilfsintegrators unter Beibehaltung des Schleppfehlers
- Bildung eines Hilfslagesollwertes für abgelöstes Positionieren (z. B. Handfahren/Referenzieren)
- Synchronisation der Y-Achse über Touch-Probe/Geber-Nullimpuls
- Drehzahl- und Drehmomentvorsteuerung

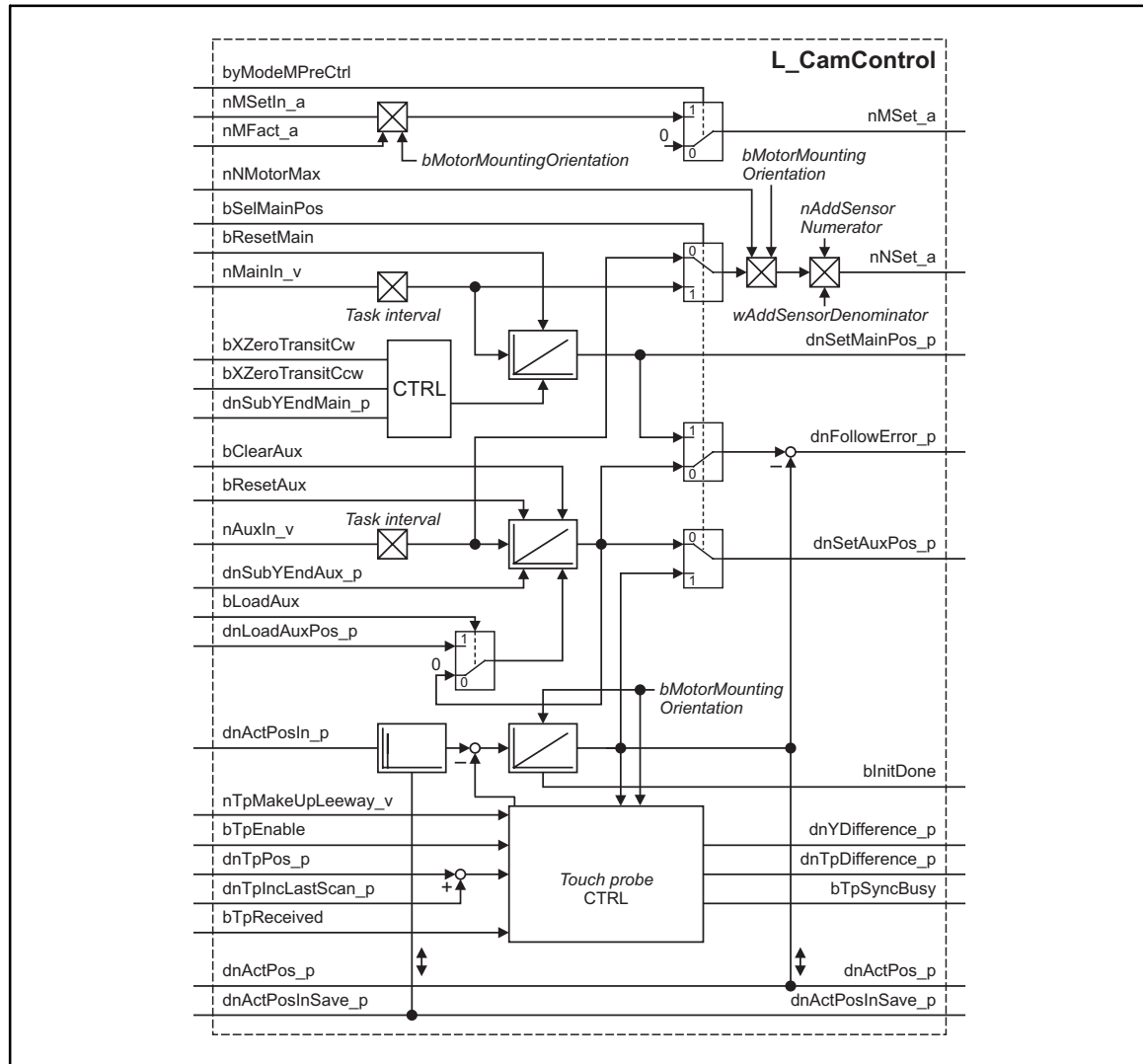
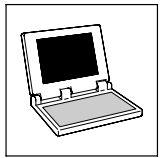


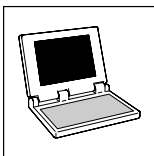
Abb. 2-14

Funktionsblock "L_CamControl"

⇒ Eingänge (Variablentyp: VAR_INPUT)			
Bezeichner	Datentyp	Signaltyp	Info/Einstellmöglichkeiten
byModeMPreCtrl	Byte	analog	Modus der Drehmomentvorsteuerung
			0 aus
			1 über entsprechendes Beschleunigungsprofil an <i>nMSetIn_a</i>
nMSetIn_a	Integer	analog	Drehmomentvorsteuerung über Kurve (FB L_CamData , normiert) Nur bei <i>byModeMPreCtrl</i> = 1
nMFact_a	Integer	analog	Anpassung der Drehmomentvorsteuerung an das Massenträgheitsmoment 16384 ≙ 100 %



⇒ Eingänge (Variablentyp: VAR_INPUT)							
Bezeichner	Datentyp	Signaltyp	Info/Einstellmöglichkeiten				
nNMotorMax	Integer	-	Maximale Drehzahl in [rpm] - Bezugswert für Drehzahlvorsteuerung. - Verbinden Sie diesen Eingang mit dem Ausgang <i>MCTRL_nNMaxC11</i> des SB MCTRL.				
bSelMainPos	Bool	binary	Auswahl des Lagesollwertes <table><tr><td>FALSE</td><td>Hilfslagesollwert</td></tr><tr><td>TRUE</td><td>Hauptlagesollwert Kurve/Bewegungsprofil Ausgang <i>dnSetAuxPos_p</i> folgt dem Istwert <i>dnActAuxPos_p</i>. </td></tr></table>	FALSE	Hilfslagesollwert	TRUE	Hauptlagesollwert Kurve/Bewegungsprofil Ausgang <i>dnSetAuxPos_p</i> folgt dem Istwert <i>dnActAuxPos_p</i>.
FALSE	Hilfslagesollwert						
TRUE	Hauptlagesollwert Kurve/Bewegungsprofil Ausgang <i>dnSetAuxPos_p</i> folgt dem Istwert <i>dnActAuxPos_p</i>.						
bResetMain	Bool	binary	Integrator für Hauptlagesollwert (Main) auf Istwert setzen. (siehe auch <i>bResetMainMode</i>) <table><tr><td>TRUE</td><td>Auf Istwert setzen.</td></tr></table>	TRUE	Auf Istwert setzen.		
TRUE	Auf Istwert setzen.						
nMainIn_v	Integer	velocity	Drehzahleingang für Hauptlagesollwert				
bXZeroTransitCw	Bool	binary	Eingang für Meldesignal "Nulldurchgang der X-Achse in positiver Richtung (Cw)" Verbinden Sie diesen Eingang mit dem gleichnamigen Ausgang des FB L_CamData. <table><tr><td>TRUE</td><td>Nulldurchgang der X-Achse in positiver Richtung (Cw), Hauptintegrator wird auf Wert an <i>dnSubYEndMain_p</i> begrenzt.</td></tr></table>	TRUE	Nulldurchgang der X-Achse in positiver Richtung (Cw), Hauptintegrator wird auf Wert an <i>dnSubYEndMain_p</i> begrenzt.		
TRUE	Nulldurchgang der X-Achse in positiver Richtung (Cw), Hauptintegrator wird auf Wert an <i>dnSubYEndMain_p</i> begrenzt.						
bXZeroTransitCcw	Bool	binary	Eingang für Meldesignal "Nulldurchgang der X-Achse in negativer Richtung (Ccw)" Verbinden Sie diesen Eingang mit dem gleichnamigen Ausgang des FB L_CamData. <table><tr><td>TRUE</td><td>Nulldurchgang der X-Achse in negativer Richtung (Ccw), Hauptintegrator wird auf Wert an <i>dnSubYEndMain_p</i> begrenzt.</td></tr></table>	TRUE	Nulldurchgang der X-Achse in negativer Richtung (Ccw), Hauptintegrator wird auf Wert an <i>dnSubYEndMain_p</i> begrenzt.		
TRUE	Nulldurchgang der X-Achse in negativer Richtung (Ccw), Hauptintegrator wird auf Wert an <i>dnSubYEndMain_p</i> begrenzt.						
dnSubYEndMain_p	Double Integer	position	Begrenzung des Hauptlagesollwerts auf Y_{max} - Verbinden Sie diesen Eingang mit dem Ausgang <i>dnYEndPos_p</i> des FB L_CamData. <i>dnSubYEndMain_p</i> = "0" ≡ Begrenzung inaktiv.				
bClearAux	Bool	binary	Hilfsintegrator zurücksetzen Ausgang <i>dnSetAuxPos_p</i> = 0 <table><tr><td>TRUE</td><td>Zurücksetzen</td></tr></table>	TRUE	Zurücksetzen		
TRUE	Zurücksetzen						
bResetAux	Bool	binary	Hilfsintegrator auf Istwert setzen - Ausgang <i>dnSetAuxPos_p</i> = <i>dnActPos_p</i> - Schleppfehler wird zurückgesetzt. - Dieser Eingang hat eine höhere Priorität als <i>bClearAux</i>. <table><tr><td>TRUE</td><td>Auf Istwert setzen</td></tr></table>	TRUE	Auf Istwert setzen		
TRUE	Auf Istwert setzen						
nAuxIn_v	Integer	velocity	Drehzahleingang für Hilfslagesollwert				
dnSubYEndAux_p	Double Integer	position	Begrenzung des Hilfslagesollwerts auf Y_{max} - Verbinden Sie diesen Eingang mit dem Ausgang <i>dnYEndPos_p</i> des FB L_CamData, wenn das Werkzeug nur innerhalb des Taktes positioniert. <i>dnSubYEndAux_p</i> = "0" ≡ Begrenzung inaktiv.				
bLoadAux	Bool	binary	Hilfsintegrator mit Wert an <i>dnLoadAuxPos_p</i> laden Der Istwertintegrator wird unter Beibehaltung des Schleppfehlers ebenfalls gesetzt. <table><tr><td>FALSE ↗ TRUE</td><td>Hilfsintegrator laden</td></tr></table>	FALSE ↗ TRUE	Hilfsintegrator laden		
FALSE ↗ TRUE	Hilfsintegrator laden						
dnLoadAuxPos_p	Double Integer	position	Wert, mit dem der Hilfsintegrator durch Setzen von <i>bLoadAux</i> auf TRUE geladen wird.				
dnActPosIn_p	Double Integer	position	Istposition der Motorsteuerung - Der Wert wird intern differenziert und als Positionsänderung pro Zyklus auf den Istwertintegrator gegeben. - Verbinden Sie diesen Eingang mit dem Ausgang <i>dnPos_p</i> des SB MCTRL.				
nTpMakeUp Leeway_v	Integer	velocity	Geschwindigkeit, mit der die Touch-Probe-Korrektur durchgeführt wird. - Bei Vorgabe einer "0" bleibt die Y-Differenz erhalten. - Für einen stoßfreien Ausgleich sollte der FB L_CamPosCtrlLin eingesetzt werden (siehe Verschaltungsbeispiel).				



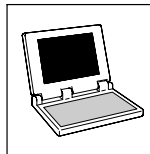
Funktionsbibliothek LenzCamControl.lib

Funktionsblöcke

L_CamControl - Sollwertaufbereitung

⇒ Eingänge (Variablentyp: VAR_INPUT)							
Bezeichner	Datentyp	Signaltyp	Info/Einstellmöglichkeiten				
bTpEnable	Bool	binary	Auswertung der Touch-Probe-Signale/Nullimpuls-Signale, bezogen auf den Hauptlagesollwert, aktivieren/deaktivieren. Nur bei <i>bSelMainPos</i> = TRUE. <table><tr><td>TRUE</td><td>Die Auswertung der Touch-Probe-Signale/Nullimpuls-Signale ist aktiviert.</td></tr><tr><td>FALSE</td><td>Die Auswertung der Touch-Probe-Signale/Nullimpuls-Signale ist deaktiviert.</td></tr></table>	TRUE	Die Auswertung der Touch-Probe-Signale/Nullimpuls-Signale ist aktiviert.	FALSE	Die Auswertung der Touch-Probe-Signale/Nullimpuls-Signale ist deaktiviert.
TRUE	Die Auswertung der Touch-Probe-Signale/Nullimpuls-Signale ist aktiviert.						
FALSE	Die Auswertung der Touch-Probe-Signale/Nullimpuls-Signale ist deaktiviert.						
dnTpPos_p	Double Integer	position	Position des Touch-Probe-Gebers				
dnTpIncLastScan_p	Double Integer	position	Inkremente seit Touch-Probe-Signal Verbinden Sie diesen Eingang mit dem Ausgang <i>MCTRL_dnActIncLastScan</i> des SB MCTRL.				
bTpReceived	Bool	binary	Steuersignal für Touch-Probe-Korrektur, manuell oder über interruptfähigen Eingang Verbinden Sie diesen Eingang mit dem Ausgang <i>MCTRL_bActTpReceived</i> des SB MCTRL. <table><tr><td>TRUE</td><td>Touch-Probe-Signal empfangen.</td></tr></table>	TRUE	Touch-Probe-Signal empfangen.		
TRUE	Touch-Probe-Signal empfangen.						

Ausgänge ⇨ (Variablentyp: VAR_OUTPUT)				
Bezeichner	Datentyp	Signaltyp	Wert/Bedeutung	
nMSet_a	Integer	analog	Drehmomentvorsteuerung Verbinden Sie diesen Ausgang mit dem Eingang <i>MCTRL_nMAdd_a</i> des SB MCTRL.	
nNSet_a	Integer	analog	Drehzahlvorsteuerung Verbinden Sie diesen Ausgang mit dem Eingang <i>MCTRL_nNSet_a</i> des SB MCTRL.	
dnSetMainPos_p	Double Integer	position	Sollwert des Bewegungsprofils Interne Begrenzung auf <i>dnYEndPos_p</i>, Position bleibt erhalten bei Verwendung des Hilfsintegrators.	
dnFollowError_p	Double Integer	position	Schleppfehler Verbinden Sie diesen Ausgang mit dem Eingang <i>MCTRL_dnSetPos_p</i> des SB MCTRL (Lageregler).	
dnSetAuxPos_p	Double Integer	position	Hilfssollwert bzw. Werkzeugposition Bei Verwendung des Hauptintegrators (<i>bSelMainPos</i> = TRUE) entspricht dieser Wert der Istposition.	
blnitDone	Bool	binary	Meldesignal "Initialisierung abgeschlossen"	
			TRUE	Initialisierung ist abgeschlossen, FB ist betriebsbereit.
			FALSE	Bleibt der Ausgang <i>blnitDone</i> auf FALSE, kann dies folgende Ursachen haben: - unvollständige/keine Kurvendaten im Arbeitsspeicher des Zielsystems: Übertragen Sie die LC9-Datei erneut in das Zielsystem. - PLC Stop: prüfen sie Codestelle C2108
dnYDifference_p	Double Integer	position	Korrekturwert nach einem Touch-Probe-Signal	
dnTpDifference_p	Double Integer	position	Differenz zwischen zwei Touch-Probe-Ereignissen	
bTpSyncBusy	Bool	binary	Meldesignal "Synchronisation ist aktiv"	
			FALSE	Momentan findet keine Korrektur des Sollwertes statt.
			TRUE	Synchronisation ist aktiv, Y-Differenz wird ausgeglichen.



⇒ Ein-/Ausgänge (Variablentyp: VAR_IN_OUT)			
Bezeichner	Datentyp	Signaltyp	Info/Einstellmöglichkeiten
dnActPos_p	Double Integer	position	Ausgang des Istwertintegrators, begrenzt auf Y_{max} . - Relative Istposition, bezogen auf den Takt des Bewegungsprofils. - Speichern Sie diesen Wert netzausfallsicher in einer Variable vom Typ PERSISTENT.
dnActPosInSave_p	Double Integer	position	Absoluter Istwert - Unabhängig vom Takt des Bewegungsprofils. - Speichern Sie diesen Wert netzausfallsicher in einer Variable vom Typ PERSISTENT.

Parameter (Variablentyp: VAR_CONSTANT_RETAIN)					
Bezeichner	Datentyp	Signaltyp	Info		
Codestelle*			Einstellmöglichkeiten		Voreinstellung
bMotorMounting Orientation C1313	Bool	-	Motoranbaulage Aus Sicht einer sich positiv bewegendes Maschine		
			FALSE	rechtsdrehend (Cw)	FALSE
			TRUE	linksdrehend (Ccw)	
bResetMainMode C1314	Bool	-	Modus für das Verhalten des <i>bResetMain</i>		
			FALSE	Hauptintegrator (Main) auf Istwert setzen.	FALSE
			TRUE	Hauptintegrator (Main) auf Null setzen.	
nAddSensor Numerator C1348	Integer	-	Faktor zur Anpassung einer zusätzlichen Lagerückführung (Zähler)		
			-32767	{1}	32767 0
wAddSensor Denominator C1349	Word	-	Faktor zur Anpassung einer zusätzlichen Lagerückführung (Nenner)		
			1	{1}	65535 1

* Codestelle der Erstinstanz

Integration von Leitfrequenzsignalen zu einer Lageposition

Zur Verfügung stehen der Hauptlagesollwert (Main) als Führungsgröße zur Regelung der Kurvenscheibe und der Hilflagesollwert (Aux) für abgelöste Positionieraufgaben oder Ähnliches.



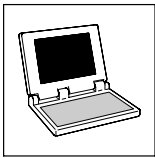
Stop!

Eine Umschaltung auf den Hauptlagesollwert mit $bSelMainPos = TRUE$ darf nur erfolgen, wenn Hauptlagesollwert und Hilflagesollwert gleich sind ($dnSetAuxPos_p = dnSetMainPos_p$), da sonst ein ungewollter Ausgleichsvorgang mit max. Drehmoment erfolgen kann.

Hauptlagesollwert

Der Integrator für den Hauptlagesollwert wird mit $bSelMainPos = TRUE$ ausgewählt.

- Der Hauptlagesollwert bleibt erhalten, wenn mit $bSelMainPos = FALSE$ auf den Hilflagesollwert umgeschaltet wird. So kann der Antrieb später seine ursprüngliche Profilposition wieder einnehmen.
- Der Integrator kann über $bResetMain = TRUE$, je nach Stellung der internen Variable $bResetMainMode$, wie folgt gesetzt werden:
 - $bResetMainMode = FALSE$:
Der Hauptlagesollwert wird auf den aktuellen Istwert ($L_CamControl.dnActPos_p$) gesetzt.
 - $bResetMainMode = TRUE$:
Der Hauptlagesollwert wird auf Null gesetzt.



Funktionsbibliothek LenzeCamControl.lib

Funktionsblöcke

L_CamControl - Sollwertaufbereitung



Stop!

Erfolgt die Schleppfehlerbildung zwischen Hauptintegrator und Istwertintegrator ($bSelMainPos = TRUE$), kann es zu einem ungewollten Ausgleichsvorgang mit max. Drehmoment kommen.

- Eine Begrenzung des Hauptlagesollwertes auf den Kurventakt findet nur statt, wenn die Eingänge $bXZeroTransmitCw$, $bXZeroTransmitCcw$ und $dnSubYEndMain_p$ mit den entsprechenden Ausgängen des FB **L_CamData** verbunden sind, zum Zeitpunkt des Nulldurchgangs wird dann einmalig Y_{max} vom Hauptlagesollwert subtrahiert bzw. addiert.

Hilfslagesollwert zum abgelösten Positionieren

Der Integrator für den Hilfslagesollwert wird mit $bSelMainPos = FALSE$ ausgewählt.

- Ist der Integrator nicht ausgewählt, liegt an dessen Ausgang $dnSetAuxPos_p$ immer der Istwert des Antriebs ($dnActPos_p$) an.
- Der Integrator kann mit $bResetAux = TRUE$ auf den Istwert gesetzt werden.



Tipp!

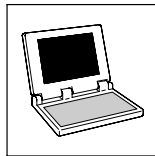
Setzen Sie $bResetAux$ bei Reglersperre und bei Schnellhalt (QSP) immer auf TRUE, um ungewollte Drehbewegungen auszuschließen.

- Der Integrator kann mit $bClearAux = TRUE$ auf Null gesetzt werden.
- Die Begrenzung des Hilfslagesollwertes findet kontinuierlich statt, wenn am Eingang $dnSubYEndAux_p$ ein Wert ungleich Null vorgegeben wird.

Laden des Hilfsintegrators

Durch Setzen von $bLoadAux$ auf TRUE kann der Hilfsintegrator mit dem Wert geladen werden, der am Positioneingang $dnLoadAuxPos_p$ anliegt (z. B. für die Funktion "dynamisches Referenz setzen").

- Das Laden des Hilfsintegrators kann nur durchgeführt werden, wenn mit $bSelMainPos = FALSE$ der Hilfsintegrator ausgewählt ist.
- Der Schleppfehler geht hierbei nicht verloren, der Istwert-Integrator wird über den Schleppfehler mitgeladen ($dnActPos_p = dnLoadAuxPos_p - dnFollowError_p$).



Synchronisation der Y-Achse über Touch-Probe (TP)



Hinweis!

Im folgenden Text wird exemplarisch die Synchronisation mittels Touch-Probe beschrieben.

Ausführliche Informationen zur Touch-Probe-Konfiguration (z. B. Aktivierung, Flankenwahl und Verzögerungszeit) finden Sie im Handbuch des entsprechenden PLC-Gerätes!

Bei der **9300 Servo PLC** kann eine Synchronisation über den Nullimpuls eines angeschlossenen Encoders bzw. Resolvers oder über die digitalen Eingänge erfolgen.

Unabhängig davon kann die Touch-Probe-Korrektur auch manuell über den Eingang *bTpReceived* gestartet werden.

- Hierbei wird kein Interrupt ausgelöst. Somit liegt kein Wert vom SB **MCTRL** am Eingang *dnTpIncLastScan_p* an, der für eine genaue Synchronisation erforderlich wäre.
- Die Synchronisationsgenauigkeit hängt somit vom Intervall der Task ab, in die der FB **L_CamControl** eingebunden ist.

Wird *bTpReceived* auf TRUE gesetzt (vom SB **MCTRL** oder manuell) und ist die Touch-Probe-Synchronisation mit *bTpEnable* = TRUE aktiviert, wird die Differenz zwischen Soll- und Istwert berechnet und an *dnYDifference_p* ausgegeben.

- Sollen eingehende Touch-Probe-Signale ausgeblendet werden, können Sie durch Setzen von *bTpEnable* auf FALSE die Touch-Probe-Synchronisation an entsprechender Stelle deaktivieren, z. B. mit Hilfe des Nockenschaltwerkes **L_CamContactor**.
- Zum Ausgleich der Differenz kann über *nTpMakeUpLeeway_v* die Geschwindigkeit zum Aufholen vorgegeben werden. Wird keine Geschwindigkeit vorgegeben, wird die berechnete Differenz nicht abgebaut, d. h. die Synchronisation bleibt inaktiv.
- Für einen stoßfreien Ausgleich der Differenz schalten Sie an *nTpMakeUpLeeway_v* den FB **L_CamPosCtrlLin** als Rampengenerator vor:

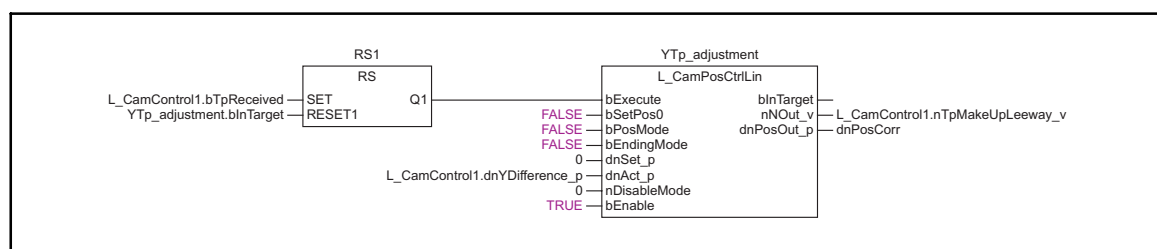
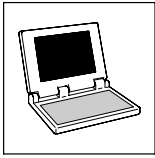


Abb. 2-15

Funktionsblock "L_CamPosCtrlLin" als Rampengenerator



Funktionsbibliothek LenzeCamControl.lib

Funktionsblöcke

L_CamControl - Sollwertaufbereitung

Getrennte Rückführung für Drehzahl- und Lageregelung

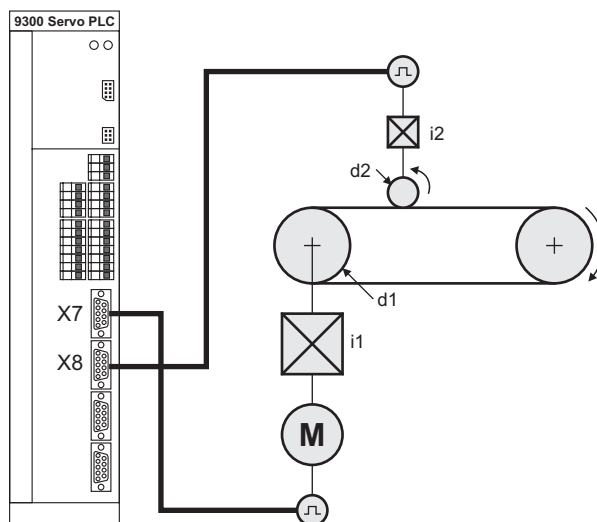
Bei schlupfenden Material kann zur Lageerfassung des Werkzeugs bzw. des Materials ein weiterer Sensor eingesetzt werden.

- Wählen Sie über C0490 das entsprechende Rückführsystem:

Code	LCD	Einstellmöglichkeiten		Info
		Lenze	Auswahl	
[C0490]	Feedback pos	0	0 Resolver an X7 1 Encoder TTL an X8 2 Sinus-/Cosinus-Geber an X8 3 Absolutwertgeber ST an X8 4 Absolutwertgeber MT an X8	Rückführsystem für den Lageregler

Über die Parameter *nAddSensorNumerator* und *wAddSensorDenominator* stellen Sie den Faktor für die Anpassung der zusätzlichen Lagerückführung ein, dieser ergibt sich aus den gegebenen Maschinenkonstanten:

$$\frac{nAddSensorNumerator}{wAddSensorDenominator} = \frac{vk2}{vk1} \cdot \frac{i1}{i2}$$



- vk1 Vorschubkonstante Drehzahlrückführung (ergibt sich aus Durchmesser d1)
- i1 Getriebefaktor Drehzahlrückführung
- vk2 Vorschubkonstante Lagerückführung (ergibt sich aus Durchmesser d2)
- i2 Getriebefaktor Lagerückführung

Abb. 2-16

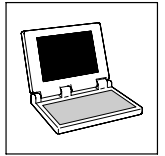
Rückführung für Drehzahl- und Lageregelung

- Der eingestellte Faktor beeinflusst nur den Drehzahlvorsteuerwert *nNSet_a*.
- Eine Anpassung erfolgt nur, wenn *nAddSensorNumerator* $\neq$ "0" ist.



Hinweis!

Bei der Erstellung des Bewegungsprofils mit dem **CamDesigner** oder **CamLoader** (als Import) werden für die getrennte Rückführung nur die Maschinenkonstanten des Lagesensors verwendet und als Maschinenkonstanten (*vk2* und *i2*) im Dialogfeld *Projekteinstellungen* für den Slave eingegeben!



Drehzahlvorsteuerung

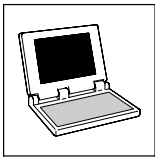
Bei der Drehzahlvorsteuerung über den Ausgang *nNSet_a* handelt es sich um die auf den Bezugs-
wert *nNMotorMax* normierte Geschwindigkeit des gewählten Integrators.

- Verbinden Sie den Vorsteuerungsausgang *nNSet_a* direkt mit dem Eingang *MCTRL_nNSet_a* des SB **MCTRL**.

Drehmomentvorsteuerung

Die Art der Drehmomentvorsteuerung über den Ausgang *nMSet_a* wird über *byModeMPreCtrl* fest-
gelegt.

- Verbinden Sie den Vorsteuerungsausgang *nMSet_a* direkt mit dem Eingang *MCTRL_nMAdd_a* des SB **MCTRL**.
- Ist *byModeMPreCtrl* auf "1" gesetzt, kann eine Vorsteuerung über ein Beschleunigungsprofil vom FB **L_CamData** erfolgen.
- Über *nMFact_a* wird eine Dehnung/Stauchung bzw. die Anpassung an das Massenträgheitsmoment des zu bewegenden Werkzeugs vorgenommen.



Funktionsbibliothek LenzeCamControl.lib

Funktionsblöcke

L_CamData - Profilverwaltung

2.5 L_CamData - Profilverwaltung

Über diesen FB können bis zu 48 verschiedene Bewegungsprofile (Kurven) verwaltet werden, deren Daten in einer internen Struktur (*.lc9-Datei) abgelegt sind.

- Die Bewegungsprofile und die daraus resultierende *.lc9-Datei können Sie auf komfortable Weise mit dem Lenze-Tool **CamDesigner** erzeugen.
- Im DDS können Sie dann mit Hilfe des **Cam Support** die erzeugte *.lc9-Datei an das Projekt binden und zusammen mit dem SPS-Programm in den Speicher der PLC laden. Der **Cam Support** befindet sich als Objekt in der Registerkarte **Ressourcen** im **Object Organizer**.

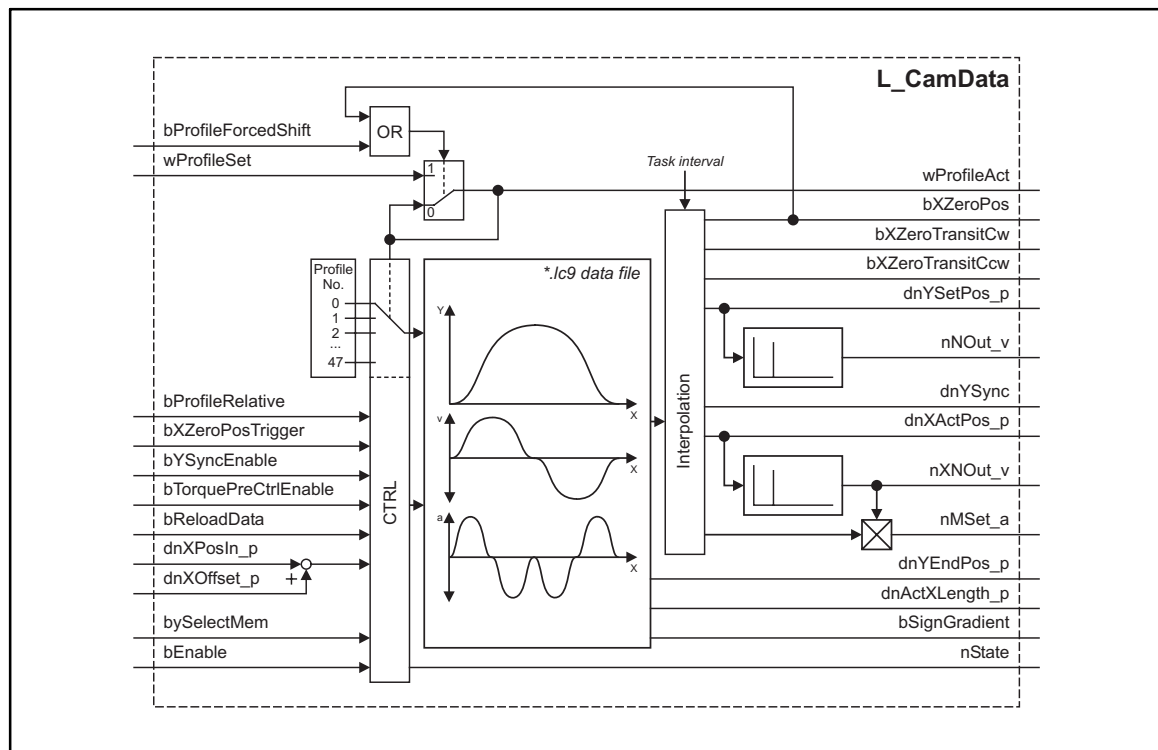
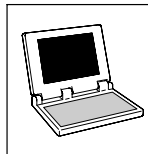
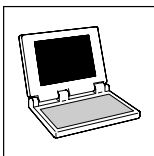


Abb. 2-17

Funktionsblock "L_CamData"



⇒ Eingänge (Variablentyp: VAR_INPUT)							
Bezeichner	Datentyp	Signaltyp	Info/Einstellmöglichkeiten				
bProfileForcedShift	Bool	binary	Zwangsumschaltung des Bewegungsprofils innerhalb des X-Takts Das neue Profil wird bzgl. der Y-Achse initialisiert, die X-Achse bleibt erhalten. <table><tr><td>FALSE ↗ TRUE</td><td>Zwangsumschaltung</td></tr></table>	FALSE ↗ TRUE	Zwangsumschaltung		
FALSE ↗ TRUE	Zwangsumschaltung						
wProfileSet	Word	number	Auswahl des Bewegungsprofils - Auswahlbereich 0 ... 47 - Der Auswahlbereich wird auf die Anzahl der in der *.lc9-Datei definierten Bewegungsprofile begrenzt. - Eine Umschaltung auf das neue Bewegungsprofil erfolgt nur im Nulldurchgang des Takts oder bei Zwangsumschaltung.				
bProfileRelative	Bool	binary	Typ bzw. Bewertung des Bewegungsprofils <table><tr><td>FALSE</td><td>Bewegungsprofil als Vor-/Rückbewegung bewerten (absolute Positionierung)</td></tr><tr><td>TRUE</td><td>Bewegungsprofil als Vorschubbewegung bewerten (relative Positionierung)</td></tr></table>	FALSE	Bewegungsprofil als Vor-/Rückbewegung bewerten (absolute Positionierung)	TRUE	Bewegungsprofil als Vorschubbewegung bewerten (relative Positionierung)
FALSE	Bewegungsprofil als Vor-/Rückbewegung bewerten (absolute Positionierung)						
TRUE	Bewegungsprofil als Vorschubbewegung bewerten (relative Positionierung)						
bXZeroPosTrigger	Bool	binary	Initialisierung der Profildaten durch vorgeschalteten FB L_CamSectionShift Wenn Sie den FB L_CamSectionShift einsetzen, verbinden Sie diesen Eingang mit dem gleichnamigen Ausgang des FB L_CamSectionShift. <table><tr><td>TRUE</td><td>Initialisierung der Profildaten.</td></tr></table>	TRUE	Initialisierung der Profildaten.		
TRUE	Initialisierung der Profildaten.						
bYSyncEnable	Bool	binary	Synchronisiertes Dehnen/Stauchen über den FB L_CamSetYAxisVelocity aktivieren. <table><tr><td>FALSE</td><td>Synchronisiertes Dehnen/Stauchen deaktiviert. (Eine deaktivierte Y-Synchronisation kann einen Inkrementenverlust zur Folge haben.)</td></tr><tr><td>TRUE</td><td>Synchronisiertes Dehnen/Stauchen aktiviert. (Diese Einstellung ist bei Verwendung des FB L_CamSetYAxisVelocity erforderlich.)</td></tr></table>	FALSE	Synchronisiertes Dehnen/Stauchen deaktiviert. (Eine deaktivierte Y-Synchronisation kann einen Inkrementenverlust zur Folge haben.)	TRUE	Synchronisiertes Dehnen/Stauchen aktiviert. (Diese Einstellung ist bei Verwendung des FB L_CamSetYAxisVelocity erforderlich.)
FALSE	Synchronisiertes Dehnen/Stauchen deaktiviert. (Eine deaktivierte Y-Synchronisation kann einen Inkrementenverlust zur Folge haben.)						
TRUE	Synchronisiertes Dehnen/Stauchen aktiviert. (Diese Einstellung ist bei Verwendung des FB L_CamSetYAxisVelocity erforderlich.)						
bTorquePreCtrl Enable	Bool	binary	Berechnung der Drehmomentvorsteuerung aktivieren Lassen Sie die Berechnung deaktiviert, wenn Sie das Drehmomentprofil nicht benötigen, um dadurch Rechenzeit zu sparen. <table><tr><td>FALSE</td><td>Berechnung deaktiviert</td></tr><tr><td>TRUE</td><td>Berechnung aktiviert</td></tr></table>	FALSE	Berechnung deaktiviert	TRUE	Berechnung aktiviert
FALSE	Berechnung deaktiviert						
TRUE	Berechnung aktiviert						
bReloadData	Bool	binary	Erneute Initialisierung der Profildaten - Hierdurch wird auch der Leitwert auf X0 zurückgesetzt. - Eine an <i>wProfileSet</i> anliegende neue Profilvernummer wird übernommen. - Eine neue Auswahl der Datenstruktur über <i>bySelectMem</i> wird übernommen. - Eine erneute Initialisierung sollte bei Verwendung des relativen Datenmodells nur bei gesetzter Reglersperre erfolgen. <table><tr><td>FALSE ↗ TRUE</td><td>Erneute Initialisierung der Profildaten.</td></tr><tr><td>TRUE</td><td>Die interne X-Achse wird auf "0" gehalten.</td></tr></table>	FALSE ↗ TRUE	Erneute Initialisierung der Profildaten.	TRUE	Die interne X-Achse wird auf "0" gehalten.
FALSE ↗ TRUE	Erneute Initialisierung der Profildaten.						
TRUE	Die interne X-Achse wird auf "0" gehalten.						
dnXPosIn_p	Double Integer	position	Leitwertvorgabe (nur als Position)				
dnXOffset_p	Double Integer	position	Offset für den Leitwert (z. B. über den FB L_CamPosCtrlLin)				
bySelectMem	Byte	number	Auswahl der Datenstruktur (*.lc9-Datei) im Speicher. Zur Übernahme einer neuen Auswahl ist eine FALSE-TRUE-Flanke an <i>bReloadData</i> erforderlich. <table><tr><td>1</td><td>RAM-Speicherblock 1 (Vordergrunddatenfeld, Standard)</td></tr><tr><td>2</td><td>RAM-Speicherblock 2 (Hintergrunddatenfeld)</td></tr></table>	1	RAM-Speicherblock 1 (Vordergrunddatenfeld, Standard)	2	RAM-Speicherblock 2 (Hintergrunddatenfeld)
1	RAM-Speicherblock 1 (Vordergrunddatenfeld, Standard)						
2	RAM-Speicherblock 2 (Hintergrunddatenfeld)						
bEnable	Bool	binary	Funktionsblock ein-/ausschalten <table><tr><td>TRUE</td><td>FB ist eingeschaltet; der Funktionsblock greift sofort auf die im Speicher befindlichen Profildaten zu, verarbeitet sie und stellt an den Ausgängen die entsprechenden Positions- und Geschwindigkeits-Signale zur Verfügung.</td></tr><tr><td>FALSE</td><td>FB ist ausgeschaltet; alle Positionsausgänge bleiben auf ihrer letzten Position stehen, alle Geschwindigkeitsausgänge werden auf Null gesetzt.</td></tr></table>	TRUE	FB ist eingeschaltet; der Funktionsblock greift sofort auf die im Speicher befindlichen Profildaten zu, verarbeitet sie und stellt an den Ausgängen die entsprechenden Positions- und Geschwindigkeits-Signale zur Verfügung.	FALSE	FB ist ausgeschaltet; alle Positionsausgänge bleiben auf ihrer letzten Position stehen, alle Geschwindigkeitsausgänge werden auf Null gesetzt.
TRUE	FB ist eingeschaltet; der Funktionsblock greift sofort auf die im Speicher befindlichen Profildaten zu, verarbeitet sie und stellt an den Ausgängen die entsprechenden Positions- und Geschwindigkeits-Signale zur Verfügung.						
FALSE	FB ist ausgeschaltet; alle Positionsausgänge bleiben auf ihrer letzten Position stehen, alle Geschwindigkeitsausgänge werden auf Null gesetzt.						

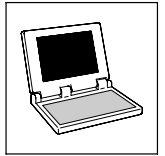


Funktionsbibliothek LenzeCamControl.lib

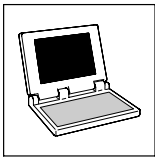
Funktionsblöcke

L_CamData - Profilverwaltung

Ausgänge ⇨ (Variablentyp: VAR_OUTPUT)							
Bezeichner	Datentyp	Signaltyp	Wert/Bedeutung				
wProfileAct	Word	number	Nummer des Bewegungsprofils, das gerade abgearbeitet wird				
bXZeroPos	Bool	binary	Meladesignal "Nulldurchgang des Leitwerts" Das Signal steht nur für einen Zyklus an, wenn Geschwindigkeit der X-Achse > 0. <table><tr><td>TRUE</td><td>Nulldurchgang (X0-Position)</td></tr></table>	TRUE	Nulldurchgang (X0-Position)		
TRUE	Nulldurchgang (X0-Position)						
bXZeroTransitCw	Bool	binary	Meladesignal "Nulldurchgang der X-Achse in positiver Richtung (Cw)" - Verbinden Sie diesen Ausgang mit dem gleichnamigen Eingang des FB L_CamControl. <table><tr><td>TRUE</td><td>Nulldurchgang der X-Achse in positiver Richtung (Cw). - Nicht bei Stillstand in X0. </td></tr></table>	TRUE	Nulldurchgang der X-Achse in positiver Richtung (Cw). Nicht bei Stillstand in X0.		
TRUE	Nulldurchgang der X-Achse in positiver Richtung (Cw). Nicht bei Stillstand in X0.						
bXZeroTransitCcw	Bool	binary	Meladesignal "Nulldurchgang der X-Achse in negativer Richtung (Ccw)" Verbinden Sie diesen Ausgang mit dem gleichnamigen Eingang des FB L_CamControl. <table><tr><td>TRUE</td><td>Nulldurchgang der X-Achse in negativer Richtung (Ccw).</td></tr></table>	TRUE	Nulldurchgang der X-Achse in negativer Richtung (Ccw).		
TRUE	Nulldurchgang der X-Achse in negativer Richtung (Ccw).						
dnYSetPos_p	Double Integer	position	Y-Sollwert - Absolute Kurvenposition, nicht begrenzt auf <i>dnYEndPos_p</i>. - Der Y-Sollwert und der Drehzahlsollwert <i>nNOut_v</i> laufen auseinander, wenn sich eine Positionsänderung ergibt, die größer ist als ±16384 [inc/ms]. Dieser Fall kann z. B. bei einer erneuten Initialisierung über <i>bReloadData</i> auftreten.				
nNOut_v	Integer	velocity	Drehzahlsollwert 16384 ≡ 15000 rpm - Wert wird intern begrenzt auf ±16384 [inc/ms] ≡ ±15000 [rpm]				
dnYSync	Double Integer	-	Synchronisation der Y-Achse (über den FB L_CamSetYAxisVelocity) - Bei <i>bYSyncEnable</i> = TRUE werden an diesem Ausgang Korrektur-Inkremente ausgegeben. - Verbinden Sie zur Synchronisation diesen Ausgang mit dem Eingang <i>dnSyncln</i> des FB L_CamSetYAxisVelocity.				
dnXActPos_p	Double Integer	position	Aktuelle X-Position				
nXNOut_v	Integer	velocity	Leitfrequenzausgang X-Achse 16384 ≡ 15000 rpm - Wird dieser Ausgang zur Erfassung der Taktfrequenz für das Nockenschaltwerk verwendet, müssen die Nocken deaktiviert werden, falls ein Reset der X-Achse durch Setzen von <i>bReloadData</i> auf TRUE durchgeführt wird. Hinweis: Das Winkeldifferenzsignal berücksichtigt den inkrementellen Taktsprung von x_{max} auf 0 am Ende/Anfang eines Taktes, nicht jedoch einen eventuellen inkrementellen Nachkommaanteil in der Taktlänge. Verwenden Sie dieses Ausgangssignal folglich nicht, wenn die Taktlänge inkrementell nicht ganzzahlig darstellbar ist!				
nMSet_a	Integer	analog	Drehmomentvorsteuerung - Ausgabe des Beschleunigungsprofils als Funktion der X-Achsen-Geschwindigkeit. - Wenn der FB L_CamSetYAxisVelocity eingesetzt wird, verbinden Sie diesen Ausgang mit dessen Eingang <i>nMIn_a</i>, ansonsten mit dem Eingang <i>nMSet_a</i> des FB L_CamControl.				
dnYEndPos_p	Double Integer	position	Takt des Bewegungsprofils 65535 ≡ 1 Motorumdrehung - Berechnung nach Profil: $dnYEndPos_p = Y_{End(Xmax)} - Y_{End(X0)}$				
dnActXLength_p	Double Integer	position	Länge der X-Achse (Taktlänge in Inkrementen)				
bSignGradient	Bool	binary	Aktuelle Steigung <table><tr><td>FALSE</td><td>Positive Steigung</td></tr><tr><td>TRUE</td><td>Negative Steigung</td></tr></table>	FALSE	Positive Steigung	TRUE	Negative Steigung
FALSE	Positive Steigung						
TRUE	Negative Steigung						



Ausgänge ⇒ (Variablentyp: VAR_OUTPUT)			
Bezeichner	Datentyp	Signaltyp	Wert/Bedeutung
nState	Integer	number	Status
			30 Initialisierung der Profildaten aktiv
			20 Warnung: Zu großer Leitwertsprung an <i>dnXPosIn_p</i> . Stützpunkte pro Zyklus konnten nicht abgearbeitet werden. Drehzahlausgang <i>nNOuT_v</i> und Positionsausgang <i>dnYSetPos_p</i> sind asynchron.
			10 Warnung: Profil nicht vorhanden (<i>wProfileSet</i> > max. Profilnummer)
			0 OK, kein Fehler.
			-2 Fehler: Es konnten keine Profildaten (*.lcl9-Datei) gelesen werden.



Funktionsbibliothek LenzCamControl.lib

Funktionsblöcke

L_CamData - Profilverwaltung

Leitwert (X-Position) für den Kurvenantrieb vorgeben

Die Vorgabe des Leitwerts (Leitwinkels) erfolgt absolut und nur positiv an $dnXPosIn_p$. Sie kann von einem virtuellen Master (FB **L_CamVMasterPosition**) generiert oder direkt von einem Positionsgeber bzw. einem Bussystem vorgegeben werden.

- Wenn Sie Leitfrequenzen als Vorgabe verwenden möchten, müssen diese über den FB **L_CamSetXAxisVelocity** zu einem Leitwinkel aufintegriert werden.
- Wird ein externer Leitwinkelgeber (programmierbarer Multiturn-/Singleturn-Absolutwertgeber) zur Vorgabe verwendet, muss dieser ggf. über einen Arithmetikblock angepasst werden.
- Das Dehnen und Stauchen der X-Achse ist nur über den FB **L_CamSetXAxisVelocity** möglich. Dieser FB beinhaltet auch die Markensynchronisierung der X-Achse (Touch-Probe) für schlupfendes Material und die Möglichkeit der Vorgabe eines Startwertes.



Hinweis!

Ergibt sich aufgrund der Vorgaben an $dnXPosIn_p$ oder $dnXOffset_p$ ein Positionssprung, der größer der halben Taktlänge ($dnActXLength_p$) ist, wird die Nulldurchgangserkennung im FB aktiviert!

Vermeiden Sie daher grundsätzlich Positionssprünge auf der X-Achse!

Offset für den Leitwinkel

Über den Eingang $dnXOffset_p$ kann ein Offset vorgegeben werden, der direkt zum Leitwinkel (X-Position) addiert wird:

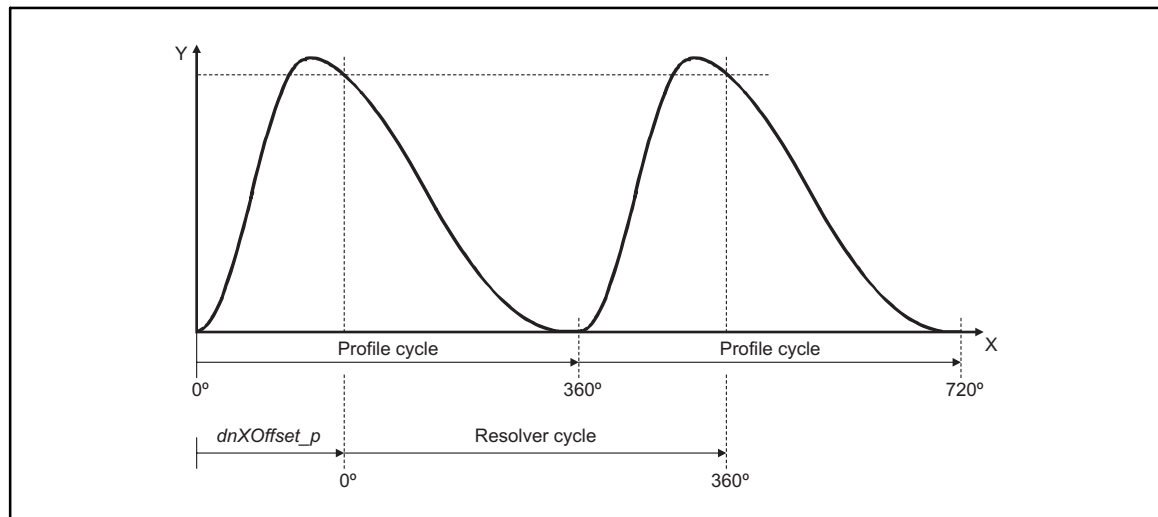


Abb. 2-18

Offsetvorgabe für den Leitwert

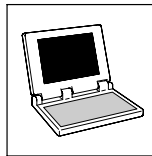
Der Offset ist nicht auf eine Taktlänge begrenzt und kann daher auch negativ vorgegeben werden.



Stop!

Der Antrieb folgt dem Offset $dnXOffset_p$ unmittelbar, d. h. eine zu große Änderung kann zu Drehmomentstößen führen.

- Steuern Sie den Eingang daher nicht direkt, sondern über einen Rampengenerator an, um den Antrieb in die gewünschte Position zu bringen oder führen Sie größere Änderungen des Offsets nur bei Reglersperre durch.
- Als Rampengenerator können Sie den FB **L_CamPosCtrlLin** verwenden.



Auswahl des Bewegungsprofils

Die Auswahl des Bewegungsprofils erfolgt durch Übergabe der entsprechenden Nummer an *wProfileSet*. Maximal 48 unterschiedlichen Bewegungsprofile (0 ... 47) können über diesen Eingang adressiert werden.

- Erfolgt eine unzulässige Auswahl, wird dies über die Statusanzeige *nState* angezeigt.
- Die Nummer des aktuell bearbeiteten Profils wird über den Ausgang *wProfileAct* angezeigt.

Umschaltung des Bewegungsprofils

Die Umschaltung auf das neue Bewegungsprofil erfolgt, wenn eine der folgenden Bedingungen erfüllt ist:

- Das Bewegungsprofil befindet sich im Nulldurchgang (Ausgang *bXZeroPos* = TRUE).
 - Durch die Umschaltung im Nulldurchgang werden unerwünschte Drehmomentstöße vermieden.
- Durch eine FALSE-TRUE-Flanke an *bProfileForcedShift* wird die Zwangsumschaltung innerhalb des Profils aktiviert.
- Durch eine FALSE-TRUE-Flanke an *bReloadData* werden die Profildaten neu initialisiert.



Hinweis!

- Bei Kurven mit unterschiedlicher X-Taktlänge darf das Bewegungsprofil nur bei einem X-Offset von "0" umgeschaltet werden!
- Eine Zwangsumschaltung mit *bProfileForcedShift* sollte grundsätzlich nur durchgeführt werden, wenn das aktuelle und das neue Profil im Umschaltpunkt den gleichen Y-Hub haben und die beiden Profile vom gleichen Typ sind.
 - Eine Zwangsumschaltung von einem Vorschub- auf ein Vor-/Rückprofil ist nicht möglich.
 - Beachten Sie, dass durch Setzen von *bProfileForcedShift* auf TRUE die Umschaltung des Profils im Nulldurchgang nicht deaktiviert wird.
 - Benutzen Sie für die Erstellung der Profile nur das absolute Datenmodell ohne Optimierung (äquidistante Verteilung der Stützpunkte).

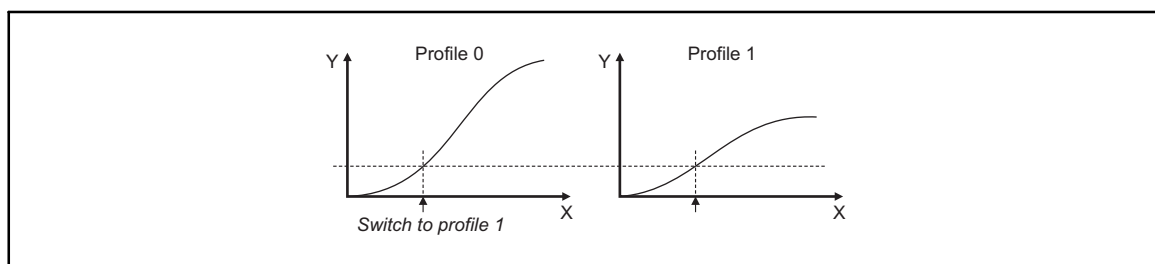
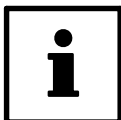


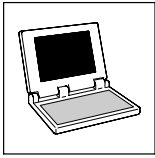
Abb. 2-19

Zwangsumschaltung von Profil 0 nach Profil 1 bei gleichem Y-Hub



Tipp!

Verwenden Sie zur Vorgabe einer Sequenz aus unterschiedlichen Bewegungsprofilen den FB **L_CamSequence!**



Funktionsbibliothek LenzCamControl.lib

Funktionsblöcke

L_CamData - Profilverwaltung

Typ des Bewegungsprofils

Über den Eingang *bProfileRelative* legen Sie fest, wie der FB das aktuelle Bewegungsprofil bewerten soll:

bProfileRelative = FALSE $\Rightarrow$ **Absolute Positionierung**

- Die absolute Positionierung verwenden Sie für Vor-/Rückprofile, bei denen der Start- und Endwert gleich ist, und wenn Bewegungsprofile absolut hintereinander geschaltet werden sollen.
- Kennzeichen: begrenzter Verfahrensweg (z. B. Spindelwerkzeuge)

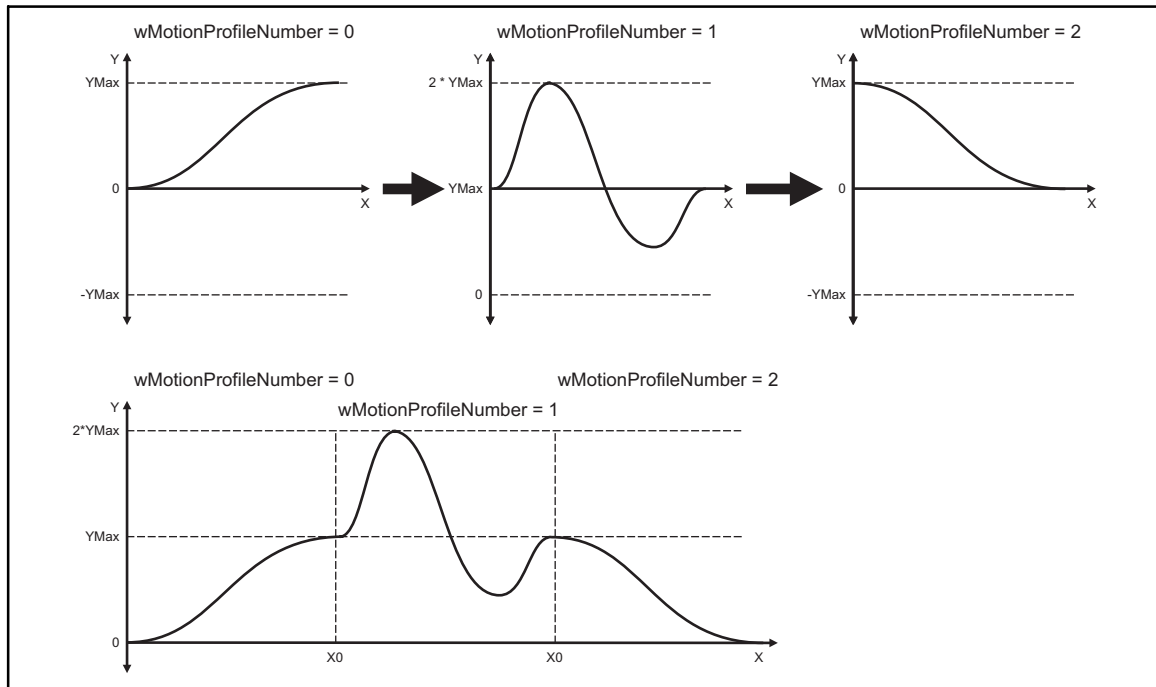


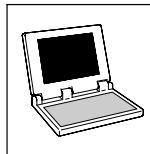
Abb. 2-20

Beispiel: Absolute Positionierung bei Vor-/Rückprofilen



Hinweis!

Stellen Sie sicher, dass bei der absoluten Positionierung und Profilschaltung die Start- und Endwerte der entsprechenden Bewegungsprofile gleich sind, da es ansonsten zu Ausgleichsbewegungen kommt!



$bProfileRelative = TRUE \Rightarrow$ **Relative Positionierung**

- Die relative Positionierung verwenden Sie für Vorschubprofile (Rundachsen), bei denen der Start- und Endwert unterschiedlich ist, und wenn Bewegungsprofile relativ hintereinander geschaltet werden sollen.
- Kennzeichen: unbegrenzter Verfahrensweg (z. B. Koppelgetriebe, Förderbänder, Rundtische)

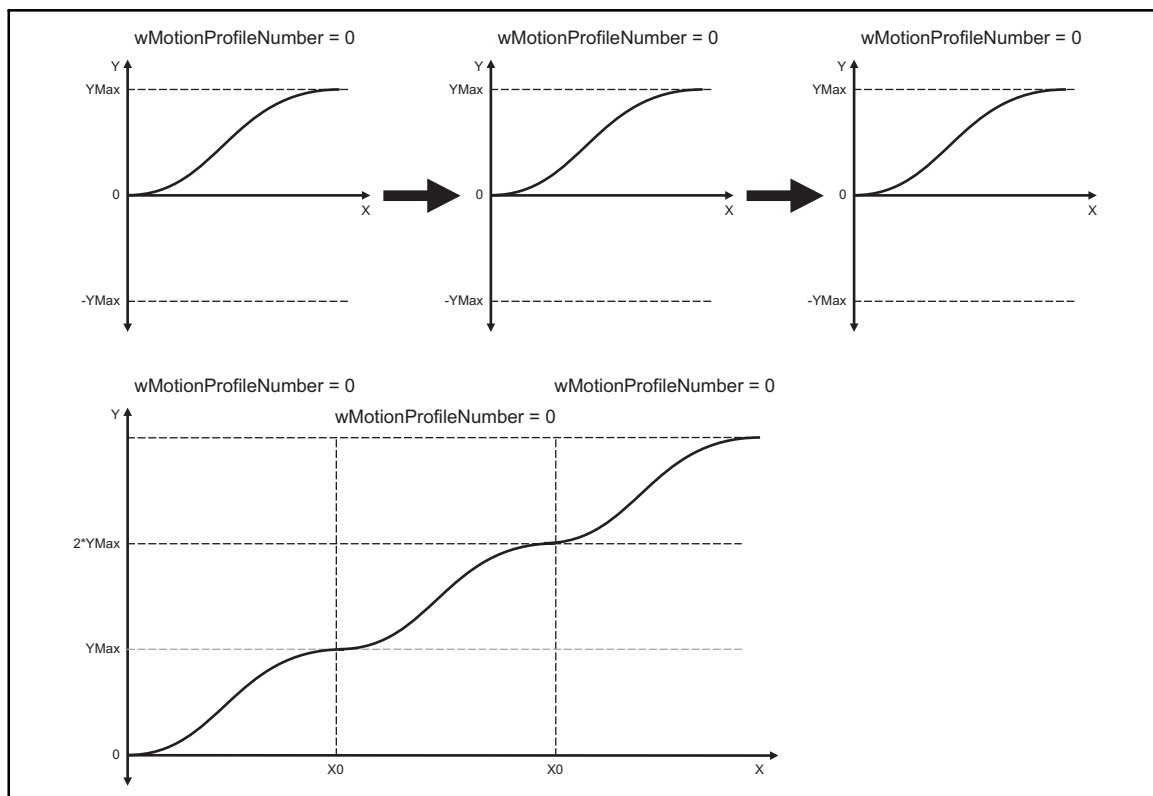


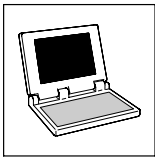
Abb. 2-21

Beispiel: Relative Positionierung bei einem Vorschubprofil



Hinweise!

- Stellen Sie sicher, dass bei der relativen Positionierung und Profilschaltung die Startwerte der entsprechenden Bewegungsprofile gleich sind!
- Bei Kurvenumschaltung zwischen Kurven mit unterschiedlicher X-Achsenlänge darf sich der Leitwinkel nur in positive Richtung verdrehen.
- Wird während des Betriebs der Dehn-/Stauchfaktor geändert, muss für diese Profilart das synchronisierte Dehnen/Stauchen aktiviert sein.
- Bei Vorschubkurven sind für die Y-Achse z. B. 0° und 360° rechnerisch nicht die gleiche Position. Eine Zwangsumschaltung innerhalb eines Vorschubprofils auf ein Vor-/Rückprofil mit $bProfileForcedShift = TRUE$ ist daher nicht möglich.



Funktionsbibliothek LenzeCamControl.lib

Funktionsblöcke

L_CamData - Profilverwaltung

Synchronisiertes Dehnen und Stauchen der Y-Achse

Ein synchronisiertes Dehnen bzw. Stauchen ist immer dann erforderlich, wenn Leitwert und Antrieb absolut synchron zueinander laufen müssen und der Dehnungs-/Stauchungsfaktor während der Produktbearbeitung umgestellt werden muss.

Die eigentliche Dehnung/Stauchung erfolgt durch den FB **L_CamSetYAxisVelocity**, der zwischen dem FB **L_CamData** und dem FB **L_CamControl** geschaltet wird:

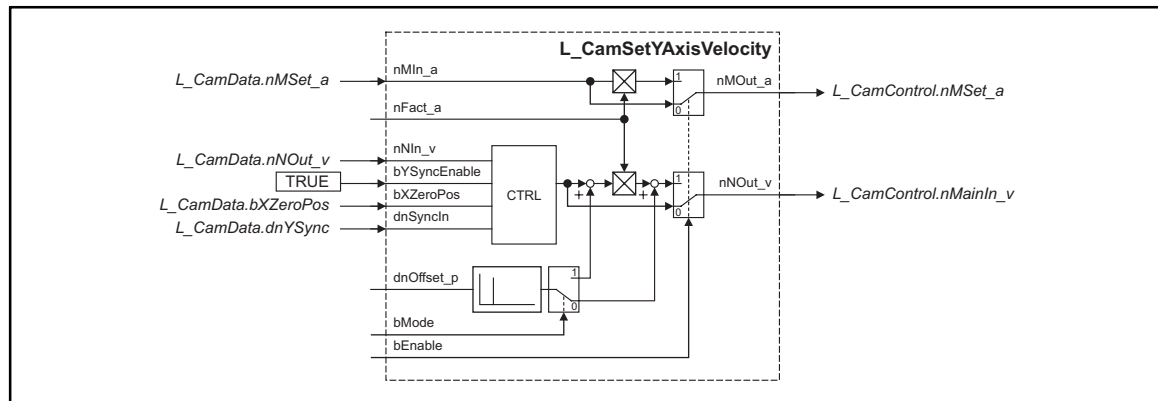


Abb. 2-22

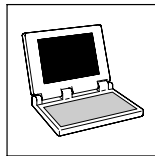
Synchronisiertes Dehnen/Stauchen mit dem FB **L_CamSetYAxisVelocity**

Die Markensynchronisierung (Touch-Probe) für die Y-Achse erfolgt durch den FB **L_CamControl**.



Hinweis!

Wird der FB **L_CamSetYAxisVelocity** nicht verwendet bzw. soll kein synchronisiertes Dehnen/Stauchen über den FB **L_CamSetYAxisVelocity** erfolgen, setzen Sie **bYSyncEnable** auf FALSE.



Drehmomentvorsteuerung

Für hochdynamische Anwendungen kann der FB **L_CamData** gleichzeitig ein dem Bewegungsprofil entsprechendes Drehmomentprofil aus der *.lc9-Datei entnehmen und am Ausgang *nMSet_a* für eine Drehmomentvorsteuerung ausgeben.

- Das Drehmomentprofil kann z. B. über den FB **L_CamSetYAxisVelocity** und den FB **L_CamControl** an die Motorregelung (SB **MCTRL**) der **9300 Servo PLC** übergeben werden.
- Wenn Sie das Drehmomentprofil nicht benötigen, sollten Sie *bTorquePreCtrlEnable* auf FALSE setzen, um diese Funktionalität zu deaktivieren und dadurch Rechenzeit zu sparen.
- Ist *bTorquePreCtrlEnable* auf TRUE gesetzt, wird das Drehmomentprofil aus der *.lc9-Datei verwendet, wobei die Amplitude bzw. der Grad der Vorsteuerung abhängig von der Geschwindigkeit der Königswelle (X-Achse) ist.



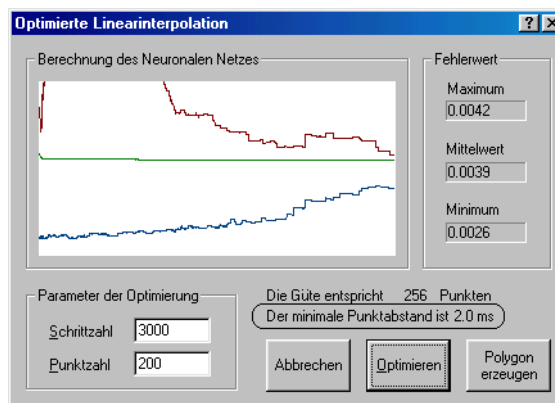
Tipp!

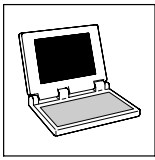
- Geben Sie beim Erstellen des Bewegungsprofils im **CamDesigner** im Dialogfeld *Eigenschaften des Masters* unbedingt die Bemessungstaktzahl mit ein, um das Signal darauf zu normieren.
- Eine Anpassung an das Massenträgheitsmoment erfolgt über den FB **L_CamControl**.

Maximale Anzahl der Stützpunkte

Die Anzahl der Stützpunkte des Bewegungsprofils sollte einen Stützpunkt pro Bearbeitungszyklus des FB **L_CamData** nicht überschreiten, da ansonsten Stützpunkte übersprungen werden.

- Werden vom FB mehr als 20 Stützpunkte übersprungen, wird am Ausgang *nState* die Warnung "20" ausgegeben. In diesem Fall sollte eine Fehlerbehandlung (z. B. Verringern der X-Achsen-Geschwindigkeit) vorgesehen werden, damit sich der Antrieb nicht unkontrolliert verhält.
- Im **CamDesigner** kann der Abstand der Stützpunkte bei gewählter Taktzahl über das Dialogfeld *Interpolation* kontrolliert werden (nur für das absolute Datenmodell):





Funktionsbibliothek LenzeCamControl.lib

Funktionsblöcke

L_CamExtrapolate - Extrapolation

2.6 L_CamExtrapolate - Extrapolation

Mit diesem FB kann eine Extrapolation von Positions- bzw. Lageinformationen durchgeführt werden, um z. B. Busübertragungszyklen größer 1 ms zu kompensieren oder Absolutwertgeber mit geringer Auflösung zu glätten. Eine Zwangsextrapolation ist ebenfalls möglich.

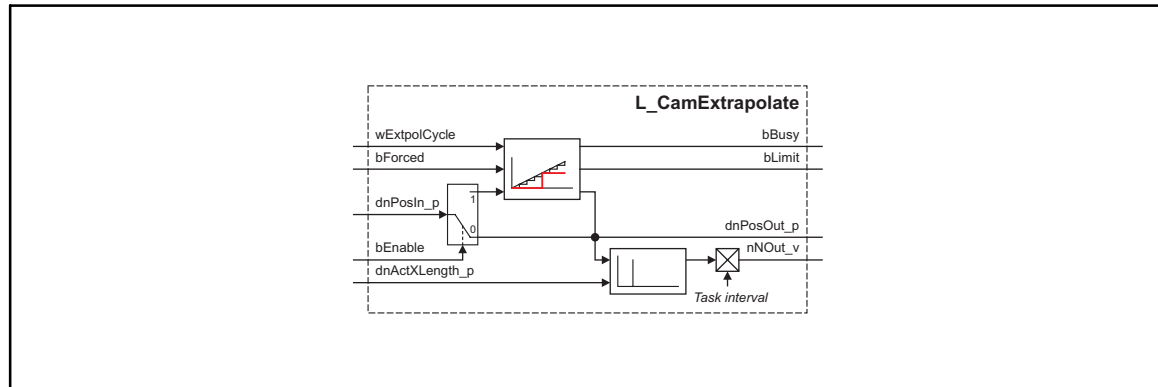
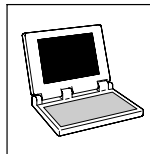


Abb. 2-23 Funktionsblock "L_CamExtrapolate"

⇒ Eingänge (Variablentyp: VAR_INPUT)							
Bezeichner	Datentyp	Signaltyp	Info/Einstellmöglichkeiten				
wExtpolCycle	Word	-	Anzahl der gewünschten Extrapolationszyklen - Wert wird intern auf 1...127 begrenzt. - Abhängig vom Taskintervall und Buszyklus.				
bForced	Bool	binary	Zwangsextrapolation <table><tr><td>FALSE</td><td>Extrapolation wird nur durchgeführt, falls sich der Eingangswert nicht geändert hat.</td></tr><tr><td>TRUE</td><td>Extrapolation wird entsprechend der Anzahl in <i>nExtpolCycle</i> durchgeführt, auch wenn sich der Eingangswert geändert hat.</td></tr></table>	FALSE	Extrapolation wird nur durchgeführt, falls sich der Eingangswert nicht geändert hat.	TRUE	Extrapolation wird entsprechend der Anzahl in <i>nExtpolCycle</i> durchgeführt, auch wenn sich der Eingangswert geändert hat.
FALSE	Extrapolation wird nur durchgeführt, falls sich der Eingangswert nicht geändert hat.						
TRUE	Extrapolation wird entsprechend der Anzahl in <i>nExtpolCycle</i> durchgeführt, auch wenn sich der Eingangswert geändert hat.						
dnPosIn_p	Double Integer	position	Positions- bzw. Lageinformation				
bEnable	Bool	binary	Extrapolation aktivieren <table><tr><td>FALSE</td><td>Der Leitwert <i>dnPosIn_p</i> wird ohne Extrapolation 1:1 am Positionsausgang <i>dnPosOut_p</i> ausgegeben. - Der Drehzahlausgang <i>nNOut_v</i> wird berechnet (Differenzierung der X-Achse). </td></tr><tr><td>TRUE</td><td>Der Leitwert <i>dnPosIn_p</i> wird extrapoliert am Positionsausgang <i>dnPosOut_p</i> ausgegeben. - Die Dauer der Extrapolation ist abhängig von der Einstellung <i>bForced</i> und <i>wExtpolCycle</i>. </td></tr></table>	FALSE	Der Leitwert <i>dnPosIn_p</i> wird ohne Extrapolation 1:1 am Positionsausgang <i>dnPosOut_p</i> ausgegeben. Der Drehzahlausgang <i>nNOut_v</i> wird berechnet (Differenzierung der X-Achse).	TRUE	Der Leitwert <i>dnPosIn_p</i> wird extrapoliert am Positionsausgang <i>dnPosOut_p</i> ausgegeben. Die Dauer der Extrapolation ist abhängig von der Einstellung <i>bForced</i> und <i>wExtpolCycle</i>.
FALSE	Der Leitwert <i>dnPosIn_p</i> wird ohne Extrapolation 1:1 am Positionsausgang <i>dnPosOut_p</i> ausgegeben. Der Drehzahlausgang <i>nNOut_v</i> wird berechnet (Differenzierung der X-Achse).						
TRUE	Der Leitwert <i>dnPosIn_p</i> wird extrapoliert am Positionsausgang <i>dnPosOut_p</i> ausgegeben. Die Dauer der Extrapolation ist abhängig von der Einstellung <i>bForced</i> und <i>wExtpolCycle</i>.						
dnActXLength_p	Double Integer	position	Länge der X-Achse				

Ausgänge → (Variablentyp: VAR_OUTPUT)			
Bezeichner	Datentyp	Signaltyp	Wert/Bedeutung
bBusy	Bool	binary	Meldesignal "Extrapolation wird durchgeführt" TRUE Extrapolation wird durchgeführt.
bLimit	Bool	binary	Meldesignal "Extrapolationsgrenze erreicht" TRUE Extrapolationsgrenze <i>wExtpolCycle</i> erreicht.
dnPosOut_p	Double Integer	phase	Ausgabe der Positions- bzw. Lageinformation
nNOut_v	Integer	velocity	Leitfrequenzausgang 16384 ≙ 15000 rpm - Begrenzung + Nachlaufspeicher Hinweis: Das Winkeldifferenzsignal berücksichtigt den inkrementellen Taktsprung von $x_{\max}$ auf 0 am Ende/Anfang eines Taktes, nicht jedoch einen eventuellen inkrementellen Nachkommaanteil in der Taktlänge. Verwenden Sie dieses Ausgangssignal folglich nicht, wenn die Taktlänge inkrementell nicht ganzzahlig darstellbar ist!



Kompensation von Busübertragungszyklen > 1ms

Werden Positions- bzw. Lageinformationen (z. B. die Lage der Königswelle) zwischen Leit- und Folgeantrieben über ein Bussystem mit einem Übertragungszyklus > 1 ms oder zwischen Tasks mit verschiedenen Zykluszeiten ausgetauscht, können die Sollwerte nicht im Regelungstakt der Antriebe verarbeitet werden. Dies hat zur Folge, dass sich bei konstanter Geschwindigkeit die Lage nicht linear, sondern treppenförmig ändert, was wiederum zu Drehmomentstößen führt.

Mit diesem FB können Sie die treppenförmige Abstufung durch Extrapolation rechnerisch glätten:

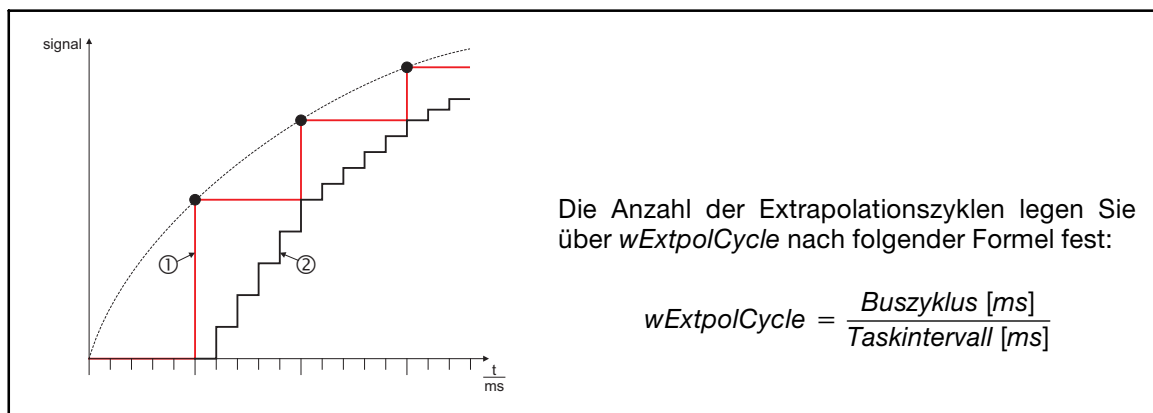


Abb. 2-24

Signalverlauf
① Sollwertsignal über Bussystem
② Interpoliertes Sollwertsignal



Tipp!

Stellen Sie sicher, dass sich für *wExtpolCycle* ein ganzzahliger Wert ergibt, damit die Lageinformation linear ansteigt und eine Drehzahlberechnung korrekt durchgeführt werden kann.

- Wird ein kleinerer Wert eingestellt, als nach der obigen Formel berechnet, wird die Extrapolationsgrenze erreicht und dies durch den Ausgang *bLimit* angezeigt. In diesem Fall wird *dnPosIn_p* direkt auf *dnPosOut_p* durchgeschaltet.
- Wird ein größerer Wert eingestellt, als nach der obigen Formel berechnet, hat dies keine Auswirkungen.

- Die Extrapolation aktivieren Sie, indem Sie *bEnable* auf TRUE setzen.
- Ist *bEnable* auf FALSE gesetzt, wird das Eingangssignal nur durchgeschliffen bzw. die Differenzbildung zur Ausgabe der Drehzahl durchgeführt.

Glättung eines gering auflösenden Absolutwertgebers

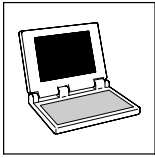
Ist die Auflösung des Absolutwertgebers so gering, dass nicht zu jedem Taskaufruf ein neuer Sollwert ansteht, kann mit Hilfe dieses FBs für einen besseren Rundlauf zwischen den Sollwerten "feininterpoliert" werden.

Geben Sie dazu in der oben aufgeführten Formel zur Berechnung von *wExtpolCycle* anstelle der Buszykluszeit die bei minimaler Drehzahl umgerechnete Auflösung ein.

Zwangsextrapolation

Zwangsextrapolation bedeutet, dass unabhängig davon, ob sich das Eingangssignal verändert hat oder nicht, bis zum maximalen Wert unter *wExtpolCycle* extrapoliert wird.

- Die Zwangsextrapolation aktivieren Sie, indem Sie *bForced* auf TRUE setzen.



Funktionsbibliothek LenzeCamControl.lib

Funktionsblöcke

L_CamGetPosition - Einheitenumrechnung

2.7

L_CamGetPosition - Einheitenumrechnung

Dieser FB rechnet den inkrementellen Wert einer Position in die physikalische Einheit um.

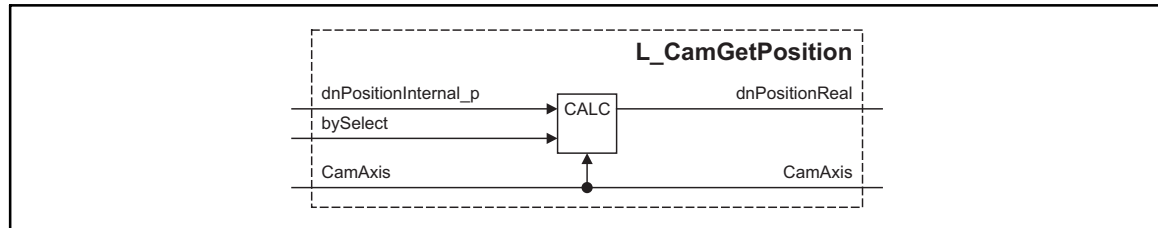


Abb. 2-25

Funktionsblock "L_CamGetPosition"

⇒ Eingänge (Variablentyp: VAR_INPUT)			
Bezeichner	Datentyp	Signaltyp	Info/Einstellmöglichkeiten
dnPositionInternal_p	Double Integer	position	Position in [inc] • 65535 ≡ 1 Motorumdrehung bzw. Geberumdrehung
bySelect	Byte	number	Auswahl, ob Master- oder Slave-Positionen umgerechnet werden sollen. 0 Master-Positionen 1 Slave-Positionen

Ausgänge ⇒ (Variablentyp: VAR_OUTPUT)			
Bezeichner	Datentyp	Signaltyp	Wert/Bedeutung
dnPositionReal	Double Integer	unit	Position in [unit] • 10000 = 1.0000 [unit]

Ein-/Ausgänge (Variablentyp: VAR_IN_OUT)			
Bezeichner	Datentyp	Signaltyp	Info
CamAxis	CAM_AXIS_REF	-	Profilparameter • Datenstruktur, deren Elemente die Maschinenparameter im internen Maßsystem beinhalten. (Beschreibung siehe FB L_CamMachineData. 2-42) • Verbinden Sie diesen Eingang mit einer globalen Variable, die vom FB L_CamMachineData beschrieben wird.

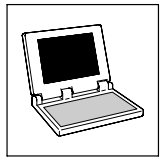
Formel

$$dnPositionReal [unit] = dnPositionInternal_p [inc] \cdot \frac{10000 \cdot \text{Vorschubkonstante [unit/Umdr.]}}{65536 [inc/Umdr.]} \cdot \frac{\text{Getriebefaktornenner}}{\text{Getriebefaktorzähler}}$$



Tipp!

- Rufen Sie diesen FB aus Rechenzeitgründen nur bei einer Änderung des Eingangssignals auf.
- Der Ausgangswert *dnPositionReal* wird im Fixkommaformat mit 4 Nachkommastellen ausgegeben.
– Beispiel: Der Wert "1205900" bedeutet 120.5900 [units].



2.8 L_CamLimitVelocity - Begrenzungsglied

Dieser FB stellt einen Drehzahlbegrenzer ohne Positionsverlust zur Verfügung.

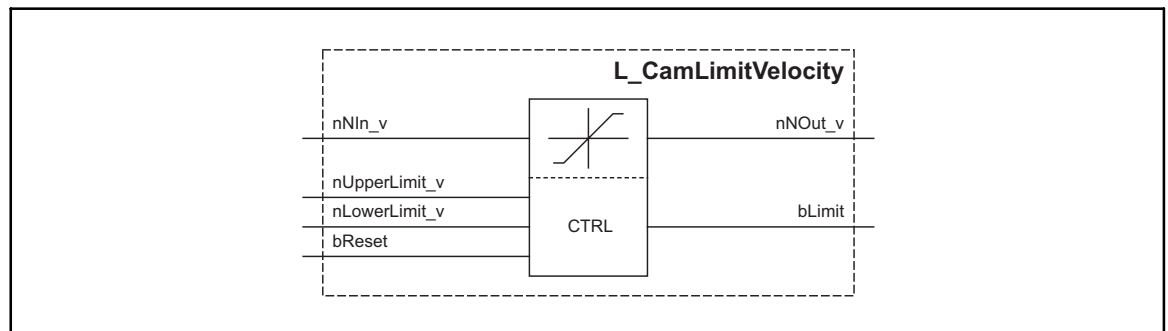


Abb. 2-26 Funktionsblock "L_CamLimitVelocity"

⇒ Eingänge (Variablentyp: VAR_INPUT)			
Bezeichner	Datentyp	Signaltyp	Info/Einstellmöglichkeiten
nNIn_v	Integer	velocity	Geschwindigkeit in [inc/ms] • 16384 [inc/ms] ≙ 15000 rpm
nUpperLimit_v	Integer	velocity	Obere Drehzahlbegrenzung
nLowerLimit_v	Integer	velocity	Untere Drehzahlbegrenzung
bReset	Bool	binary	Nachlaufspeicher löschen
			TRUE Nachlaufspeicher wird gelöscht.

Ausgänge ⇒ (Variablentyp: VAR_OUTPUT)			
Bezeichner	Datentyp	Signaltyp	Wert/Bedeutung
nNOut_v	Integer	velocity	Geschwindigkeit in [inc/ms] • 16384 [inc/ms] ≙ 15000 rpm
bLimit	Bool	binary	Meldesignal "Begrenzung aktiv"
			TRUE Obere bzw. untere Begrenzung ist aktiv.

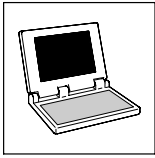
Drehzahlbegrenzung ohne Positionsverlust

Während die Drehzahlbegrenzung aktiv ist, wird ein Nachlaufspeicher gefüllt. Er sorgt dafür, dass die Position nicht verloren geht. Es handelt sich hierbei um einen asynchronen Zustand, d. h. die Kurvenscheibe läuft nicht mehr synchron zum Leitantrieb bzw. zur Königswelle, wenn das Eingangssignal *nNIn_v* außerhalb der Grenzen ist.

- Der Inhalt des Nachlaufspeichers wird geleert, wenn das Eingangssignal wieder kleiner als der obere und größer als der untere Grenzwert ist.
- Über den boolschen Eingang *bReset* besteht die Möglichkeit, z. B. für den Einrichtbetrieb den Nachlaufspeicher ohne Einfluss auf die übrige Funktionalität zu löschen.

Begrenzungseingänge

- Stellen Sie sicher, dass der Wert an *nUpperLimit_v* immer größer ist als an *nLowerLimit_v*.
- Wenn *nUpperLimit_v* = 0 UND *nLowerLimit_v* = 0 ist keine Begrenzung aktiv und das Eingangssignal wird direkt auf den Ausgang geschaltet. Der Nachlaufspeicher wird deaktiviert und gelöscht.
- Wenn *nUpperLimit_v* = +16384 UND *nLowerLimit_v* = -16384 ist ebenfalls keine Begrenzung aktiv.



Funktionsbibliothek LenzeCamControl.lib

Funktionsblöcke

L_CamMachineData - Maschinenparameter

2.9

L_CamMachineData - Maschinenparameter

Dieser FB bereitet die Maschinenparameter für das interne Maßsystem auf.

- Die Maschinenparameter geben Sie in den physikalischen Einheiten der Maschine im Lenze Tool **CamDesigner** ein, diese werden dann zusammen mit den Daten der Bewegungsprofile in einer *.lc9-Datei abgelegt.
- Im DDS können Sie daraufhin mit Hilfe des **Cam Support** die erzeugte *.lc9-Datei an das Projekt binden und zusammen mit dem SPS-Programm in den Speicher der PLC laden. Der **Cam Support** befindet sich als Objekt in der Registerkarte **Ressourcen** im **Object Organizer**.
- Die für das interne Maßsystem aufbereiteten Maschinenparameter legt der FB in einer Datenstruktur (CAM_AXIS_REF) ab.
- Über den Ausgang *CamAxis* können Sie diese Datenstruktur einer globalen Variable zuweisen, um über diese "Schnittstelle" die Maschinenparameter an andere LenzeCamControl-FBs zu übergeben.

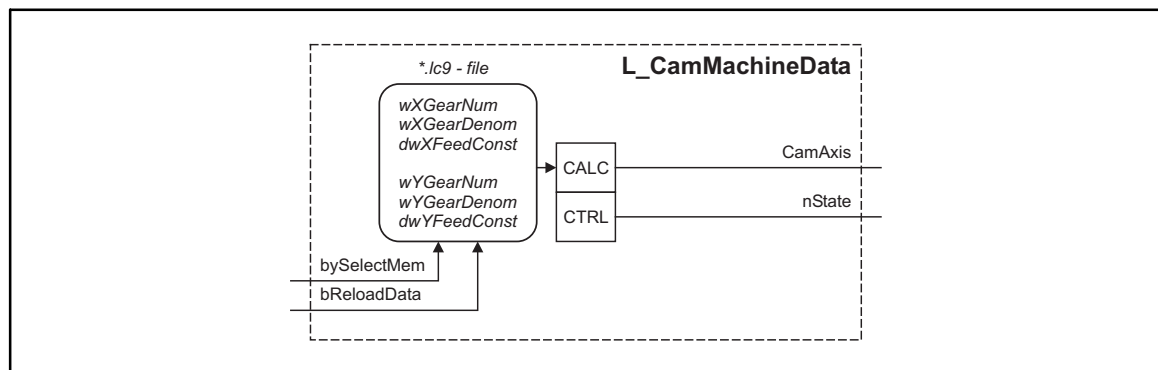
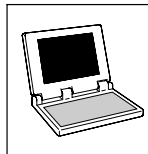


Abb. 2-27

Funktionsblock "L_CamMachineData"

Interne Parameter (Maschinenkonstanten)			
Bezeichner	Datentyp	Signaltyp	Info/Einstellmöglichkeiten
wXGearNum	Word	-	Master (X) - Getriebefaktorzähler
wXGearDenom	Word	-	Master (X) - Getriebefaktorenenner
dwXFeedConst	Double Word	-	Master (X) - Vorschubkonstante
wYGearNum	Word	-	Slave (Y) - Getriebefaktorzähler
wYGearDenom	Word	-	Slave (Y) - Getriebefaktorenenner
dwYFeedConst	Double Word	-	Slave (Y) - Vorschubkonstante

⇒ Eingänge (Variablentyp: VAR_INPUT)			
Bezeichner	Datentyp	Signaltyp	Info/Einstellmöglichkeiten
bySelectMem	Byte	number	Auswahl der Datenstruktur (*.lc9-Datei) im Speicher Zur Übernahme einer neuen Auswahl ist eine FALSE-TRUE-Flanke an <i>bReloadData</i> erforderlich.
			1RAM-Speicherblock 1 (Vordergrunddatenfeld, Standard)
			2RAM-Speicherblock 2 (Hintergrunddatenfeld)
bReloadData	Bool	binary	Aktualisierung der Ausgänge
			FALSE → TRUEErneute Initialisierung der Profildaten und Aktualisierung der Ausgänge.



Ausgänge ⇒ (Variablentyp: VAR_OUTPUT)									
Bezeichner	Datentyp	Signaltyp	Wert/Bedeutung						
CamAxis	CAM_AXIS_REF	-	Datenstruktur mit den Maschinenparametern im internen Maßsystem. Mit Hilfe einer globalen (Zwischen-)Variable können Sie diese Maschinenparameter anderen LenzeCamControl-FBs zur Verfügung stellen.						
nState	Integer	number	Status<table><tr><td>10</td><td>Initialisierung: Lesen der Profildaten</td></tr><tr><td>0</td><td>OK, kein Fehler</td></tr><tr><td>-2</td><td>Fehler: Es konnten keine Profildaten (*.lc9-Datei) gelesen werden.</td></tr></table>	10	Initialisierung: Lesen der Profildaten	0	OK, kein Fehler	-2	Fehler: Es konnten keine Profildaten (*.lc9-Datei) gelesen werden.
10	Initialisierung: Lesen der Profildaten								
0	OK, kein Fehler								
-2	Fehler: Es konnten keine Profildaten (*.lc9-Datei) gelesen werden.								

Übergabe der Maschinenparameter

Deklarieren Sie zur Übergabe der Maschinenparameter an andere FBs eine globale Variable vom Datentyp **CAM_AXIS_REF** und verbinden Sie diese Variable

- mit dem Ausgang **CamAxis** des FB **L_CamMachineData**
- mit den entsprechenden Eingängen der FBs, welche die Maschinenparameter im internen Maßsystem benötigen.

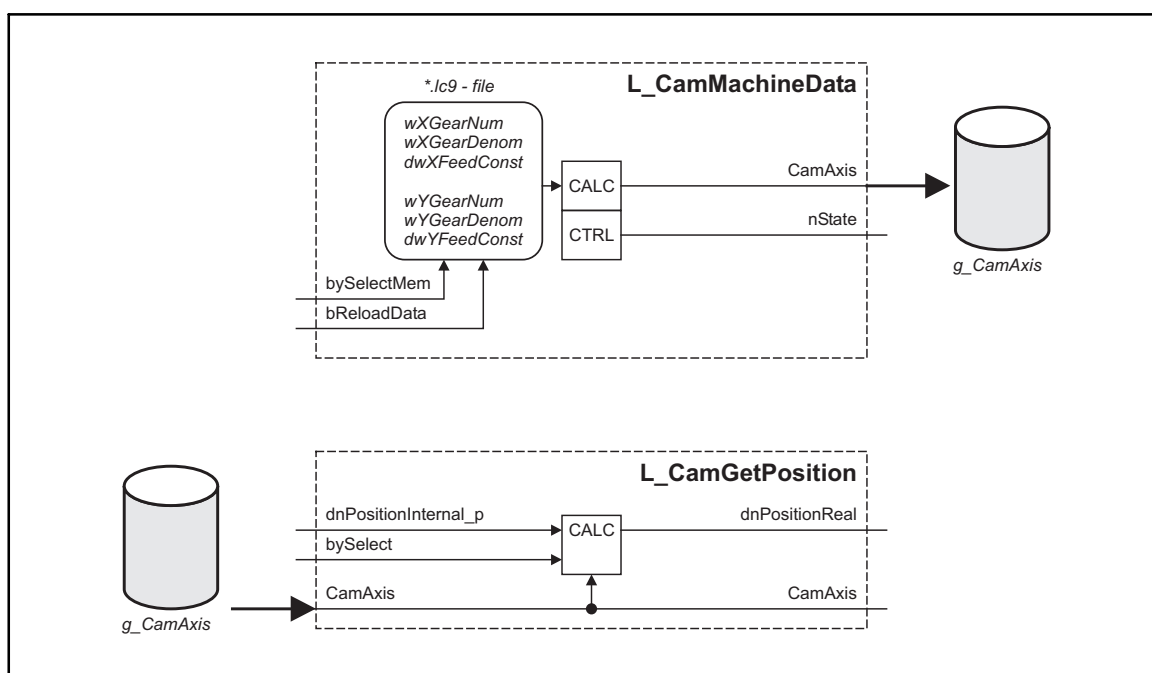


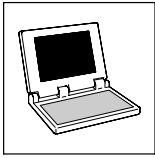
Abb. 2-28

Übergabe der Maschinenparameter mit Hilfe einer globalen Variable (hier: *g_CamAxis*)



Tipp!

Der Aufruf des FB **L_CamMachineData** kann z. B. im Hauptprogramm **PLC_PRG** erfolgen. Die internen Maschinenparameter werden aktualisiert durch eine FALSE-TRUE-Flanke an *bReloadData*.



Funktionsbibliothek LenzeCamControl.lib

Funktionsblöcke

L_CamMonitFollowError - Schleppfehlerüberwachung

2.10 L_CamMonitFollowError - Schleppfehlerüberwachung

Mit diesem FB können Sie eine Schleppfehlerüberwachung realisieren, ausgewertet wird hierbei das Schleppfehlersignal des FB **L_CamControl**.

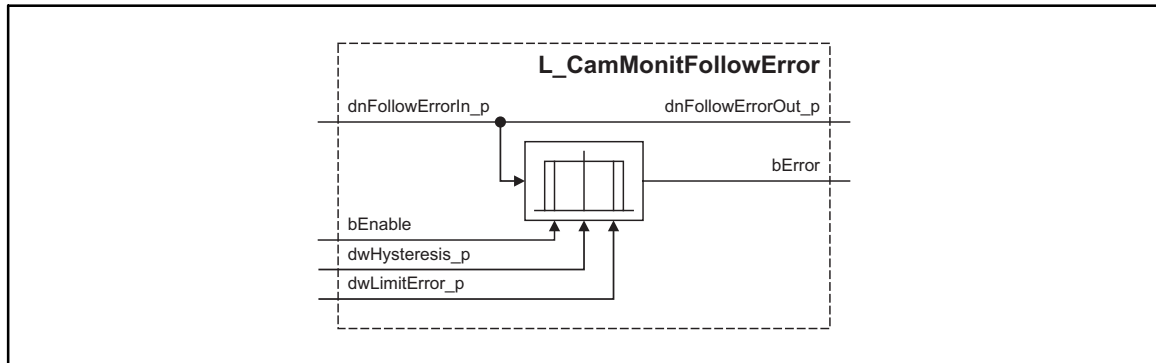
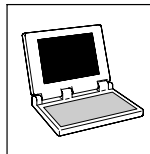


Abb. 2-29

Funktionsblock "L_CamMonitFollowError"

⇒ Eingänge (Variablentyp: VAR_INPUT)			
Bezeichner	Datentyp	Signaltyp	Info/Einstellmöglichkeiten
bEnable	Bool	binary	Überwachung aktivieren TRUE Überwachung aktiviert.
dnFollowErrorIn_p	Double Integer	position	Schleppfehlerereingang • Verbinden Sie diesen Eingang mit dem Ausgang <i>dnFollowError_p</i> des FB L_CamControl .
dwHysteresis_p	Double Word	position	Hysterese für den Fehlerausgang
dwLimitError_p	Double Word	position	Schaltschwelle für positiven und negativen Schleppfehler

Ausgänge ⇒ (Variablentyp: VAR_OUTPUT)			
Bezeichner	Datentyp	Signaltyp	Wert/Bedeutung
dnFollowErrorOut_p	Double Integer	position	Schleppfehlerausgang • Verbinden Sie diesen Ausgang z. B. mit dem Eingang <i>dnPosSet_p</i> des SB MCTRL (Abweichung von Ist- zu Sollwinkel für Winkelregler).
bError	Bool	binary	Meldesignal "Schleppfehler" TRUE Schleppfehler über Schaltschwelle



Auswertung

Ausgewertet wird das Schleppfehlersignal *dnFollowError_p* vom FB **L_CamControl**, das Sie hierzu mit dem Eingang *dnFollowErrorIn_p* verbinden müssen.

- Im FB erfolgt ein Vergleich mit der konfigurierbaren Schleppfehlergrenze *dwLimitError_p*. Ein Überschreiten dieser Grenze in positiver oder negativer Richtung wird durch den Zustand TRUE am Ausgang *bError* signalisiert.
- Für das Rücksetzen des Ausgangs *bError* kann an *dwHysteresis_p* eine Hysterese vorgegeben werden, um ein ständiges Hin- und Herschalten im Umschaltpunkt zu vermeiden.
- Das Schleppfehlereingangssignal wird zudem 1:1 an *dnFollowErrorOut_p* ausgegeben.

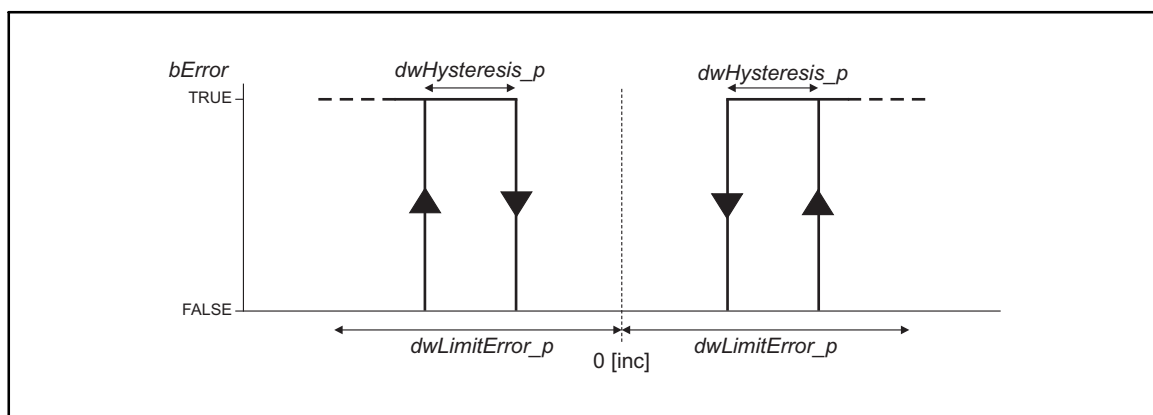


Abb. 2-30

Schaltschwelle mit Hysterese für (Vorwarnstufe-)Ausgang *bError*



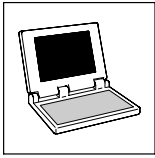
Tipp!

Bei der **9300 Servo PLC** können Sie z. B. den Ausgang *dnFollowErrorOut_p* mit dem Eingang *dnPosSet_p* des SB **MCTRL** verbinden, um der Motorsteuerung die Abweichung von Ist- zu Sollwinkel für den Winkelregler mitzuteilen.

Deaktivierung der Schleppfehlerüberwachung

Durch Setzen von *bEnable* auf FALSE kann die Schleppfehlerüberwachung deaktiviert werden.

Auch bei deaktivierter Schleppfehlerüberwachung wird das Schleppfehlereingangssignal 1:1 am Ausgang *dnFollowErrorOut_p* ausgegeben, so dass der FB zu diesem Zweck nicht ausgebaut werden muss.



Funktionsbibliothek LenzCamControl.lib

Funktionsblöcke

L_CamPosCtrlLin - Linearer Profilgenerator

2.11 L_CamPosCtrlLin - Linearer Profilgenerator

Über diesen FB können folgende Positionierfunktionen realisiert werden:

- Antrieb auf Kurvenposition bringen (z. B. nach Netzeinschalten, Handfahren, Referenzieren)
- Antrieb von der Kurve lösen und in sichere Position bringen (ablösende Positionierung)
- Positionierung der X-Achse über X-Offset (überlagerte Positionierung)
- Positionierung der Y-Achse über Y-Offset (überlagerte Positionierung)

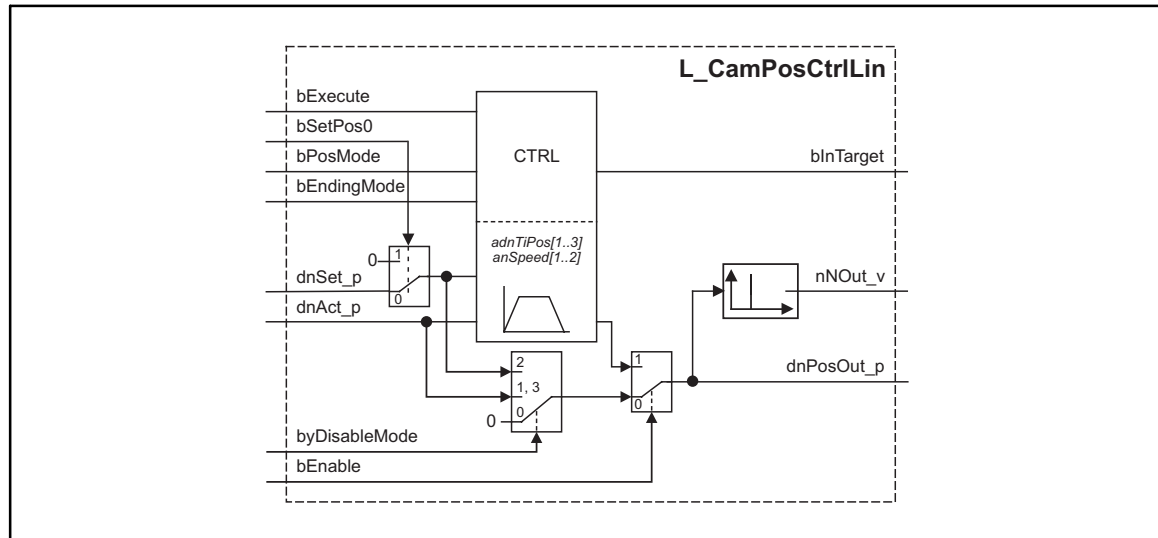
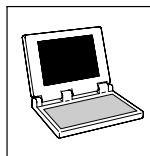


Abb. 2-31

Funktionsblock "L_CamPosCtrlLin"

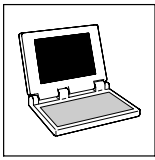
⇒ Eingänge (Variablentyp: VAR_INPUT)				
Bezeichner	Datentyp	Signaltyp	Info/Einstellmöglichkeiten	
bExecute	Bool	binary	Positionierprofil ausführen Nur möglich, wenn der Eingang <i>bEnable</i> = TRUE gesetzt ist.	
			FALSE	Keine Positionierung / Positionierung wird abgebrochen. Die Geschwindigkeit wird auf 0 rpm geführt (<i>nOut_v</i> = 0).
			TRUE	Positionierung wird durchgeführt. Bei Nutzung des Abschaltmodus <i>bEndingMode</i> = TRUE ist eine erneute positive Flanke für einen Folgepositioniervorgang erforderlich.
bSetPos0	Bool	binary	Die Sollposition wird intern auf "0" gesetzt. Nur wenn <i>bEnable</i> = TRUE.	
			FALSE	Als Sollposition wird der Wert am Eingang <i>dnSet_p</i> verwendet.
			TRUE	Der Wert am Eingang <i>dnSet_p</i> wird ignoriert. Die Sollposition wird intern auf "0" gesetzt.
bPosMode	Bool	binary	Positioniermodus	
			FALSE	Absolute Positionierung
			TRUE	Relative Positionierung - Nur bei <i>bEndingMode</i> = TRUE (mit Endabschaltung) verwenden! - Mit einer FALSE-TRUE-Flanke am Eingang <i>bExecute</i> verfahren die Ausgänge um den Wert <i>dnSet_p</i> weiter.



⇒ Eingänge (Variablentyp: VAR_INPUT)											
Bezeichner	Datentyp	Signaltyp	Info/Einstellmöglichkeiten								
bEndingMode	Bool	binary	Endabschaltung <table><tr><td>FALSE</td><td>Ohne Endabschaltung - Die Ausgänge folgen kontinuierlich dem Sollwert an <i>dnSet_p</i>. </td></tr><tr><td>TRUE</td><td>Mit Endabschaltung - Die Ausgänge folgen einmalig dem Sollwert an <i>dnSet_p</i>. - Ändert sich die Sollposition, ist für einen weiteren Positioniervorgang eine erneute FALSE-TRUE-Flanke am Eingang <i>bExecute</i> erforderlich. </td></tr></table>	FALSE	Ohne Endabschaltung Die Ausgänge folgen kontinuierlich dem Sollwert an <i>dnSet_p</i>.	TRUE	Mit Endabschaltung - Die Ausgänge folgen einmalig dem Sollwert an <i>dnSet_p</i>. - Ändert sich die Sollposition, ist für einen weiteren Positioniervorgang eine erneute FALSE-TRUE-Flanke am Eingang <i>bExecute</i> erforderlich.				
FALSE	Ohne Endabschaltung Die Ausgänge folgen kontinuierlich dem Sollwert an <i>dnSet_p</i>.										
TRUE	Mit Endabschaltung - Die Ausgänge folgen einmalig dem Sollwert an <i>dnSet_p</i>. - Ändert sich die Sollposition, ist für einen weiteren Positioniervorgang eine erneute FALSE-TRUE-Flanke am Eingang <i>bExecute</i> erforderlich.										
dnSet_p	Double Integer	position	Sollwert für die Positionierung								
dnAct_p	Double Integer	position	Istposition Verbinden Sie diesen Eingang z. B. mit dem Ausgang <i>dnSetAuxPos_p</i> des FB L_CamControl.								
byDisableMode	Byte	-	Verhalten bei deaktivierter Positionierfunktion (<i>bEnable</i> = FALSE) <table><tr><td>0</td><td>Der Ausgang <i>dnPosOut_p</i> springt auf "0". Das Geschwindigkeitssignal am Ausgang <i>nOut_v</i> bildet diesen Positionssprung als Winkelsignal ab. Hinweis: Wirkt der Ausgang <i>nOut_v</i> auf die Sollwertbildung für den Antrieb, kann es zu undefinierten Bewegungen (Stöße, Schleppfehler) kommen.</td></tr><tr><td>1</td><td>Die Ausgänge <i>dnPosOut_p</i> und <i>nOut_v</i> folgen dem Wert am Eingang <i>dnAct_p</i> (Istposition).</td></tr><tr><td>2</td><td>Die Ausgänge <i>dnPosOut_p</i> und <i>nOut_v</i> folgen dem Wert am Eingang <i>dnSet_p</i> (Sollwert).</td></tr><tr><td>3</td><td>Die Ausgänge <i>dnPosOut_p</i> und <i>nOut_v</i> folgen dem Wert am Eingang <i>dnAct_p</i> (Istposition), jedoch ohne Begrenzung.</td></tr></table>	0	Der Ausgang <i>dnPosOut_p</i> springt auf "0". Das Geschwindigkeitssignal am Ausgang <i>nOut_v</i> bildet diesen Positionssprung als Winkelsignal ab. Hinweis: Wirkt der Ausgang <i>nOut_v</i> auf die Sollwertbildung für den Antrieb, kann es zu undefinierten Bewegungen (Stöße, Schleppfehler) kommen.	1	Die Ausgänge <i>dnPosOut_p</i> und <i>nOut_v</i> folgen dem Wert am Eingang <i>dnAct_p</i> (Istposition).	2	Die Ausgänge <i>dnPosOut_p</i> und <i>nOut_v</i> folgen dem Wert am Eingang <i>dnSet_p</i> (Sollwert).	3	Die Ausgänge <i>dnPosOut_p</i> und <i>nOut_v</i> folgen dem Wert am Eingang <i>dnAct_p</i> (Istposition), jedoch ohne Begrenzung.
0	Der Ausgang <i>dnPosOut_p</i> springt auf "0". Das Geschwindigkeitssignal am Ausgang <i>nOut_v</i> bildet diesen Positionssprung als Winkelsignal ab. Hinweis: Wirkt der Ausgang <i>nOut_v</i> auf die Sollwertbildung für den Antrieb, kann es zu undefinierten Bewegungen (Stöße, Schleppfehler) kommen.										
1	Die Ausgänge <i>dnPosOut_p</i> und <i>nOut_v</i> folgen dem Wert am Eingang <i>dnAct_p</i> (Istposition).										
2	Die Ausgänge <i>dnPosOut_p</i> und <i>nOut_v</i> folgen dem Wert am Eingang <i>dnSet_p</i> (Sollwert).										
3	Die Ausgänge <i>dnPosOut_p</i> und <i>nOut_v</i> folgen dem Wert am Eingang <i>dnAct_p</i> (Istposition), jedoch ohne Begrenzung.										
bEnable	Bool	binary	Positionierfunktion aktivieren/deaktivieren - Dieser Eingang hat die höchste Priorität <table><tr><td>FALSE</td><td>Positionierfunktion deaktiviert - Verhalten siehe <i>nDisableMode</i> </td></tr><tr><td>TRUE</td><td>Positionierfunktion aktiviert</td></tr></table>	FALSE	Positionierfunktion deaktiviert Verhalten siehe <i>nDisableMode</i>	TRUE	Positionierfunktion aktiviert				
FALSE	Positionierfunktion deaktiviert Verhalten siehe <i>nDisableMode</i>										
TRUE	Positionierfunktion aktiviert										

Ausgänge → (Variablentyp: VAR_OUTPUT)					
Bezeichner	Datentyp	Signaltyp	Wert/Bedeutung		
dnPosOut_p	Double Integer	position	Sollposition		
nNOut_v	Integer	velocity	Geschwindigkeitsausgang des Profilgenerators 16384 ≙ 15000 rpm		
blnTarget	Bool	binary	Meldesignal "Lageziel erreicht" <table><tr><td>TRUE</td><td>Lageziel erreicht</td></tr></table>	TRUE	Lageziel erreicht
TRUE	Lageziel erreicht				

Parameter (Variablentyp: VAR_CONSTANT_RETAIN)																											
Bezeichner	Datentyp	Signaltyp	Info																								
Codestelle*			Einstellmöglichkeiten Voreinstellung																								
adnTiPos[1..3]	Array of Double Integer	-	Rampen für Beschleunigung/Verzögerung (bezogen auf 4000 rpm) <table> <tr> <td><i>adnTiPos[1]</i> = Verzögerungsrampe für Halt vor Erreichen der Sollposition (<i>bExecute</i> = FALSE)</td><td></td><td></td><td></td></tr> <tr> <td>0.010</td><td>{0.001 s}</td><td>130.000</td><td>1.000 s</td></tr> <tr> <td><i>adnTiPos[2]</i> = Beschleunigungsrampe für den Positioniervorgang</td><td></td><td></td><td></td></tr> <tr> <td>0.010</td><td>{0.001 s}</td><td>130.000</td><td>1.000 s</td></tr> <tr> <td><i>adnTiPos[3]</i> = Verzögerungsrampe für den Positioniervorgang</td><td></td><td></td><td></td></tr> <tr> <td>0.010</td><td>{0.001 s}</td><td>130.000</td><td>1.000 s</td></tr> </table>	<i>adnTiPos[1]</i> = Verzögerungsrampe für Halt vor Erreichen der Sollposition (<i>bExecute</i> = FALSE)				0.010	{0.001 s}	130.000	1.000 s	<i>adnTiPos[2]</i> = Beschleunigungsrampe für den Positioniervorgang				0.010	{0.001 s}	130.000	1.000 s	<i>adnTiPos[3]</i> = Verzögerungsrampe für den Positioniervorgang				0.010	{0.001 s}	130.000	1.000 s
<i>adnTiPos[1]</i> = Verzögerungsrampe für Halt vor Erreichen der Sollposition (<i>bExecute</i> = FALSE)																											
0.010	{0.001 s}	130.000	1.000 s																								
<i>adnTiPos[2]</i> = Beschleunigungsrampe für den Positioniervorgang																											
0.010	{0.001 s}	130.000	1.000 s																								
<i>adnTiPos[3]</i> = Verzögerungsrampe für den Positioniervorgang																											
0.010	{0.001 s}	130.000	1.000 s																								
anSpeed[1..2]	Array of Integer	-	Verfahrgeschwindigkeiten																								



Funktionsbibliothek LenzeCamControl.lib

Funktionsblöcke

L_CamPosCtrlLin - Linearer Profilgenerator

Parameter (Variablentyp: VAR_CONSTANT_RETAIN)				
Codestelle*			Einstellmöglichkeiten	Voreinstellung
C1078/1			<i>anSpeed[1]</i> = Vorlauf (positive Geschwindigkeit)	
			-15000 {1 rpm} 15000	200 rpm
C1078/2			<i>anSpeed[2]</i> = Rücklauf (negative Geschwindigkeit)	
			-15000 {1 rpm} 15000	200 rpm

* Codestelle der Erstinstanz

Absolute/relative Positionierung

Absolute Positionierung	Relative Positionierung
- Auswahl: <i>bPosMode</i> = FALSE - Der Sollwert an <i>dnSet_p</i> gilt als absolute Sollposition (bezogen auf die Nullposition).	- Auswahl: <i>bPosMode</i> = TRUE - Die Ausgänge werden von der aktuellen Lage weiter um den Wert <i>dnSet_p</i> verfahren. - Bedingung: <i>bEndingMode</i> = TRUE (mit Endabschaltung)

Ohne/mit Endabschaltung

Ohne Endabschaltung	Mit Endabschaltung
- Auswahl: <i>bEndingMode</i> = FALSE - Die Ausgänge folgen kontinuierlich dem Sollwert an <i>dnSet_p</i>.	- Auswahl: <i>bEndingMode</i> = TRUE - Die Ausgänge folgen einmalig dem Sollwert an <i>dnSet_p</i>. - Ändert sich die Sollposition, ist für einen weiteren Positioniervorgang eine erneute FALSE-TRUE-Flanke am Eingang <i>bExecute</i> erforderlich.

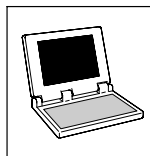
Positionierung stoppen

Mit *bExecute* = FALSE kann ein aktiver Positioniervorgang gestoppt werden.

- Über die Verzögerungsrampe *adnTiPos[1]* wird die Drehzahl am Ausgang *nNOut_v* in den Stillstand geführt.
- Gleichzeitig wird der Ausgang *dnPosOut_p* angehalten.
- Mit *bExecute* = TRUE kann die Positionierung fortgesetzt bzw. eine neuer Positioniervorgang gestartet werden. In der Betriebsart "mit Endabschaltung" (*bEndingMode* = TRUE) ist hierfür eine FALSE-TRUE-Flanke an *bExecute* erforderlich.

Auf Nullposition fahren

Mit *bSetPos0* = TRUE wird *dnSet_p* intern auf Null gesetzt und der Ausgang *dnPosOut_p* kann auf Nullposition gefahren werden.



Beispiel: Vorgabe eines Offset für die Y-Achse

Folgendes Fahrprofil soll mit dem FB **L_CamPosCtrlLin** realisiert werden:

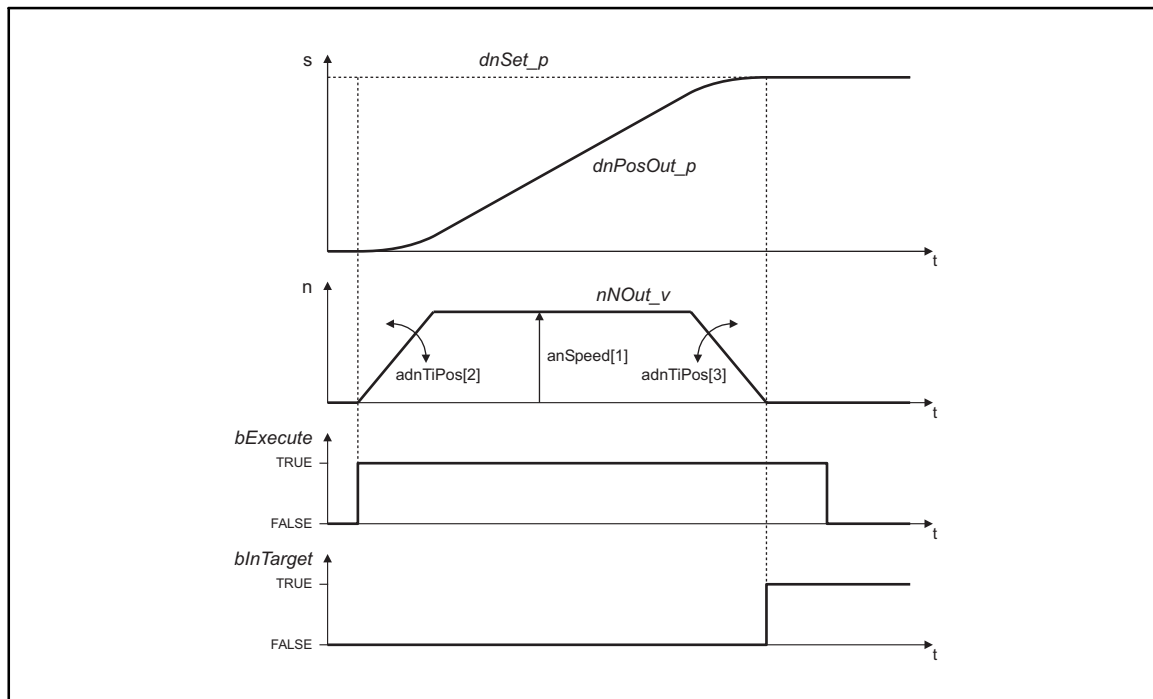


Abb. 2-32

Fahrprofil

Schaltungsvorschlag:

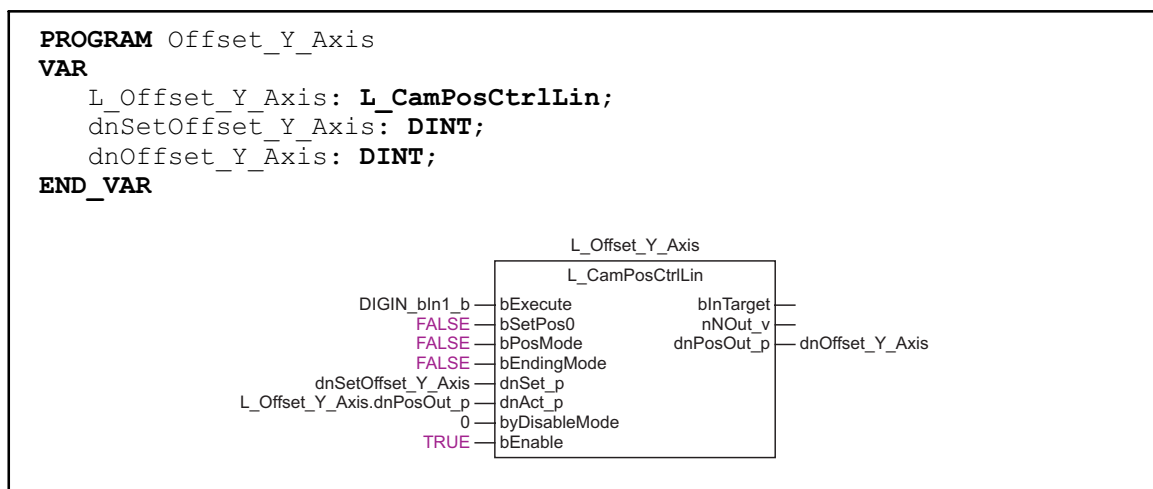
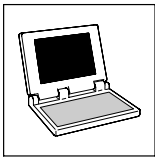


Abb. 2-33

Schaltungsvorschlag für Vorgabe eines Offset für die Y-Achse

- Die Sollposition wird hier über die Variable *dnSetOffset_Y_Axis* vorgegeben.
- Die Istposition kann direkt vom Ausgang *dnPosOut_p* abgenommen werden.
- Über den digitalen Eingang *DIGIN_bIn1_b* erfolgt die Freigabe der Positionierung.
- Der Eingang *bEnable* muss während der Positionierung auf TRUE gesetzt sein.



Funktionsbibliothek LenzeCamControl.lib

Funktionsblöcke

L_CamProfileData - Laden wichtiger Profilpunkte

2.12 L_CamProfileData - Laden wichtiger Profilpunkte

Dieser FB stellt wichtige Profilpunkte als interne inkrementelle Größen zur Verfügung.

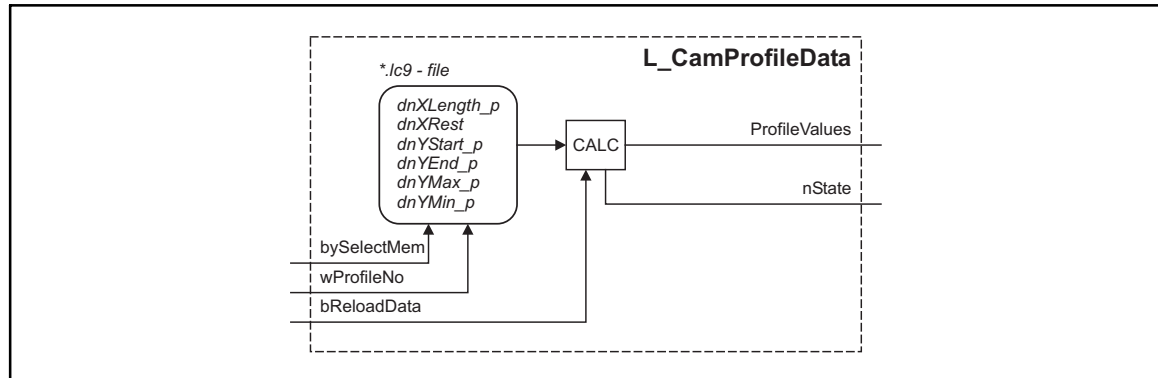
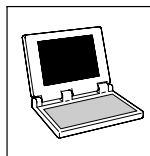


Abb. 2-34

Funktionsblock "L_CamProfileData"

⇒ Eingänge (Variablentyp: VAR_INPUT)				
Bezeichner	Datentyp	Signaltyp	Info/Einstellmöglichkeiten	
bySelectMem	Byte	number	Auswahl der Datenstruktur (*.lc9-Datei) im Speicher	
			Zur Übernahme einer neuen Auswahl ist eine FALSE-TRUE-Flanke an <i>bReloadData</i> erforderlich.	
			<table><tr><td>1</td><td>RAM-Speicherblock 1 (Vordergrunddatenfeld, Standard)</td></tr><tr><td>2</td><td>RAM-Speicherblock 2 (Hintergrunddatenfeld)</td></tr></table>	1
1	RAM-Speicherblock 1 (Vordergrunddatenfeld, Standard)			
2	RAM-Speicherblock 2 (Hintergrunddatenfeld)			
wProfileNo	Word	number	Auswahl des Bewegungsprofils - Auswahlbereich 0 ... 47 - Der Auswahlbereich wird auf die Anzahl der in der *.lc9-Datei definierten Bewegungsprofile begrenzt.	
bReloadData	Bool	binary	Aktualisierung der Ausgänge	
			Eine neue Auswahl der Datenstruktur über <i>bySelectMem</i> wird übernommen. <table><tr><td>FALSE ↗ TRUE</td><td>Die Ausgänge werden aktualisiert.</td></tr></table>	FALSE ↗ TRUE
FALSE ↗ TRUE	Die Ausgänge werden aktualisiert.			

Ausgänge ⇨ (Variablentyp: VAR_OUTPUT)				
Bezeichner	Datentyp	Signaltyp	Wert/Bedeutung	
ProfileValues	CAM_PROFILE_REF	position	Datenstruktur mit den wichtigen Punkten des ausgewählten Bewegungsprofils Positionen in [inc]	
nState	Integer	number	Status	
			15	Initialisierung bzw. Lesen der Daten aus der *.lc9-Datei
			10	Warnung: Profil nicht vorhanden (<i>wProfileNo</i> > max. Profilnummer)
			0	FB befindet sich im Zustand "StandBy"
			-2	Fehler: Es konnten keine Profildaten (*.lc9-Datei) gelesen werden.



Datenstruktur CAM_PROFILE_REF

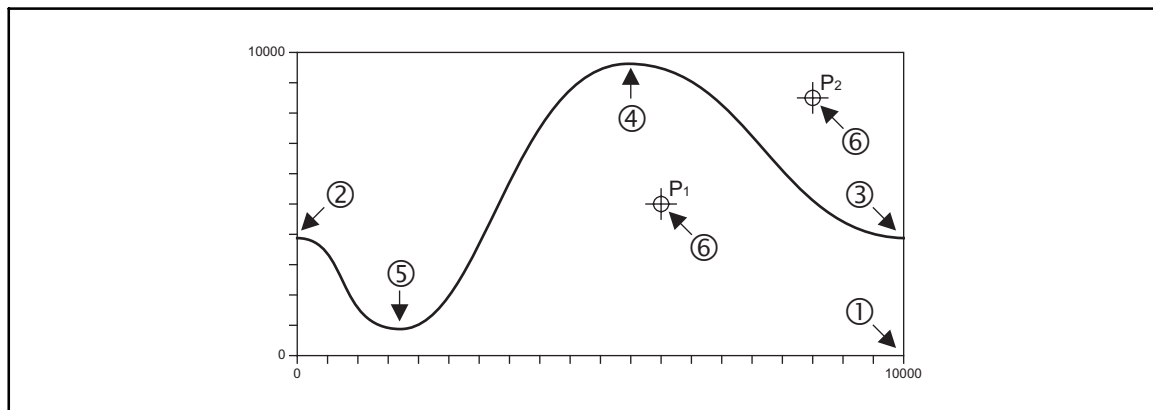


Abb. 2-35

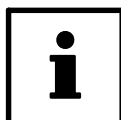
Beispielprofil

Elemente der Datenstruktur CAM_PROFILE_REF			
Bezeichner		Datentyp	Bedeutung
①	dnXLength_p	Double Integer	Taktlänge (Länge der X-Achse) in [inc]
	dnXRest	Double Integer	Nachkommateil der Taktlänge
②	dnYStart_p	Double Integer	Startposition Y-Achse in [inc] • bei X-Position = 0
③	dnYEnd_p	Double Integer	Endposition Y-Achse in [inc] • bei X-Position = <i>dnXLength_p</i>
④	dnYMax_p	Double Integer	Maximale Y-Position/maximaler Y-Hub in [inc]
⑤	dnYMin_p	Double Integer	Minimale Y-Position/minimaler Y-Hub in [inc]
⑥	Über den CamDesigner können Sie bis zu 4 zusätzliche Stützpunkte eingeben, die bzgl. der Y-Position nicht zwingend auf dem Bewegungsprofil liegen müssen. • Die Eingabe dieser Stützpunkte erfolgt über den Befehl Zeichnen→Positions-Merker. • Beim Kompilieren werden die Stützpunktdaten zusammen mit dem Bewegungsprofil und den Nocken in die aktuelle *.Ic9-Datei geschrieben.		
	dnXAddPos_p[0..3]	Array of Double Integer	X-Positionen der zusätzlichen Stützpunkte (P1 ... P4) in [inc]
	dnYAddPos_p[0..3]	Array of Double Integer	Y-Positionen der zusätzlichen Stützpunkte (P1 ... P4) in [inc]

Ausgabe/Aktualisierung der Daten

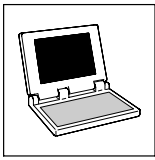
Die Positionen am Ausgang *ProfileValues* werden in den folgenden Fällen aktualisiert:

- Nach einem Profilwechsel an *wProfileNo*.
- Durch eine FALSE-TRUE-Flanke an *bReloadData*.
 - Dies ist notwendig z. B. nach einer Veränderung der Profildaten durch den FB **L_CamEditProfilData**.



Tipp!

Es reicht aus, den FB in einer zyklischen Task einzubinden, d. h. der Aufruf des FBs muss nicht zwingend in einer Task mit fester Zykluszeit (z. B. T#10ms) erfolgen.



Funktionsbibliothek LenzeCamControl.lib

Funktionsblöcke

L_CamSequence - Vorgabe einer Profilsequenz

2.13 L_CamSequence - Vorgabe einer Profilsequenz

Mit diesem FB kann dem FB **L_CamData** eine Sequenz von Profilen vorgegeben werden, hierbei werden u. a. folgende Funktionen unterstützt:

- Vorgabe einer beliebigen Sequenz
- Takt- und Dauerbetrieb
- Hold- und Reset-Funktion
- Vorgabe eines Startschritts

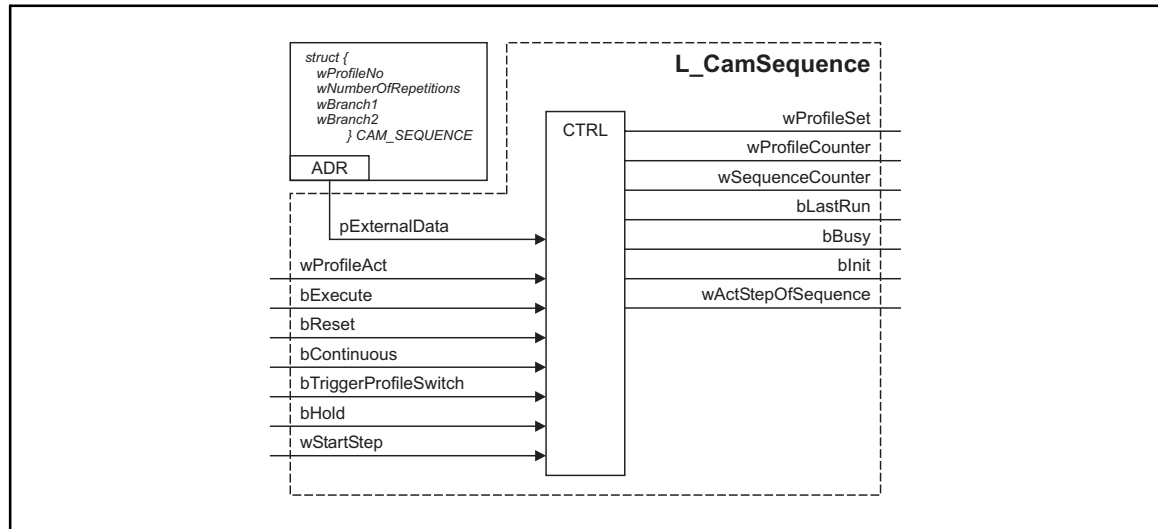
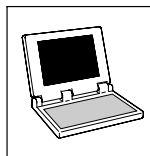


Abb. 2-36

Funktionsblock "L_CamSequence"

⇒ Eingänge (Variablentyp: VAR_INPUT)			
Bezeichner	Datentyp	Signaltyp	Info/Einstellmöglichkeiten
pExternalData	Pointer to CAM_SEQUENCE	-	Pointer auf die Anfangsadresse der Sequenz (Array vom Datentyp <i>CAM_SEQUENCE</i> im Anwenderprogramm).
wProfileAct	Word	number	Nummer des momentan abgefahrenen Profils (Istprofil). • Verbinden Sie diesen Eingang mit dem gleichnamigen Ausgang des FB L_CamData. 0 ... 47 Profil Nr. 0 ... 47
bExecute	Bool	binary	Sequenz starten/abbrechen FALSE Abarbeitung der Sequenz abbrechen • Das aktuelle Profil wird noch abgefahren. • Durch erneutes Setzen auf TRUE wird die Abarbeitung der Sequenz fortgesetzt. TRUE Abarbeitung der Sequenz starten/fortsetzen
bReset	Bool	binary	Reset durchführen • Die Durchführung des Reset benötigt 4 Aufrufzyklen des FB. • Während der Durchführung des Reset ist der Ausgang <i>bInit</i> auf TRUE gesetzt. TRUE Die Daten werden mit den Startwerten neu initialisiert und die Zählerausgänge werden zurückgesetzt.
bContinuous	Bool	binary	Auswahl Dauerbetrieb/Taktbetrieb FALSE Taktbetrieb • Die Sequenz wird genau einmal abgearbeitet. • Ist die Sequenz abgearbeitet, wird <i>bBusy</i> auf FALSE gesetzt. TRUE Dauerbetrieb • Die Sequenz wird fortlaufend erneut abgearbeitet; nach Sequenzende wird wieder mit <i>wStartStep</i> begonnen. • Über <i>wSequenceCounter</i> wird ausgegeben, wie oft die Sequenz bereits abgearbeitet wurde.



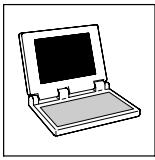
⇒ Eingänge (Variablentyp: VAR_INPUT)							
Bezeichner	Datentyp	Signaltyp	Info/Einstellmöglichkeiten				
bTriggerProfile Switch	Bool	binary	Trigger-Eingang für Profil- bzw. Schrittwechsel Verbinden Sie diesen Eingang mit dem Ausgang <i>bXZeroPos</i> des FB L_CamData. <table><tr><td>TRUE ⇒ FALSE</td><td>Es wird zum nächsten Profil bzw. Schritt der Sequenz gewechselt.</td></tr></table>	TRUE ⇒ FALSE	Es wird zum nächsten Profil bzw. Schritt der Sequenz gewechselt.		
TRUE ⇒ FALSE	Es wird zum nächsten Profil bzw. Schritt der Sequenz gewechselt.						
bHold	Bool	binary	Haltefunktion Der Ausgang <i>bBusy</i> bleibt während der Haltefunktion auf TRUE gesetzt. <table><tr><td>FALSE</td><td>Sequenz fortsetzen.</td></tr><tr><td>TRUE</td><td>Die Sequenz wird angehalten und es wird immer der gleiche Schritt abgearbeitet. Der Zählerausgang <i>wProfileCounter</i> läuft bis Null abwärts.</td></tr></table>	FALSE	Sequenz fortsetzen.	TRUE	Die Sequenz wird angehalten und es wird immer der gleiche Schritt abgearbeitet. Der Zählerausgang <i>wProfileCounter</i> läuft bis Null abwärts.
FALSE	Sequenz fortsetzen.						
TRUE	Die Sequenz wird angehalten und es wird immer der gleiche Schritt abgearbeitet. Der Zählerausgang <i>wProfileCounter</i> läuft bis Null abwärts.						
wStartStep	Word	number	Nummer des ersten Schritts der Sequenz. Ein neuer Wert wird nur bei <i>bExecute</i> = FALSE und abgeschlossener Sequenz (<i>bBusy</i> = FALSE) übernommen. <table><tr><td>0 ... 47</td><td>Schritt Nr. 0 ... 47</td></tr></table>	0 ... 47	Schritt Nr. 0 ... 47		
0 ... 47	Schritt Nr. 0 ... 47						

Ausgänge ⇨ (Variablentyp: VAR_OUTPUT)					
Bezeichner	Datentyp	Signaltyp	Wert/Bedeutung		
wProfileSet	Word	number	Nummer des Profils des aktuellen Schritts (Sollprofil). Verbinden Sie diesen Ausgang mit dem gleichnamigen Eingang des FB L_CamData .		
wProfileCounter	Word	number	Anzahl, wie oft das Profil des aktuellen Schritts noch abgefahren wird. Der Wert "0" bedeutet "letzter Durchlauf".		
wSequenceCounter	Word	number	Anzahl, wie oft die Sequenz im Dauerbetrieb bereits durchlaufen wurde. Maximaler Zählerwert = 65535		
bLastRun	Bool	binary	Meldesignal "Abarbeitung des letzten Profils" Kann verwendet werden, um den Virtuellen Master zu stoppen. <table><tr><td>TRUE</td><td>Letztes Profil innerhalb der Sequenz wird abgearbeitet.</td></tr></table>	TRUE	Letztes Profil innerhalb der Sequenz wird abgearbeitet.
TRUE	Letztes Profil innerhalb der Sequenz wird abgearbeitet.				
bBusy	Bool	binary	Meldesignal "Abarbeitung aktiv" <table><tr><td>TRUE</td><td>Sequenz wird noch abgearbeitet.</td></tr></table>	TRUE	Sequenz wird noch abgearbeitet.
TRUE	Sequenz wird noch abgearbeitet.				
bInit	Bool	binary	Meldesignal "Initialisierung aktiv" <table><tr><td>TRUE</td><td>Initialisierung über <i>bReset</i> aktiv (max. 4 Aufrufzyklen des FB).</td></tr></table>	TRUE	Initialisierung über <i>bReset</i> aktiv (max. 4 Aufrufzyklen des FB).
TRUE	Initialisierung über <i>bReset</i> aktiv (max. 4 Aufrufzyklen des FB).				
wActStepOf Sequence	Word	number	Anzeige des aktuell bearbeiteten Schrittes einer Sequenz		

Vorgabe der Sequenz

Die Sequenz wird im Anwenderprogramm anhand eines Arrays der Datenstruktur *CAM_SEQUENCE* definiert:

CAM_SEQUENCE (Struktur)		
Bezeichner	Datentyp	Info
wProfileNo	Word	Sollprofil für diesen Schritt.
wNumberOfRepetitions	Word	Anzahl, wie oft der Schritt nacheinander durchlaufen werden soll.
wBranch1	Word	Folgeschritt, wenn Anzahl festgelegter Wiederholungen noch nicht erreicht wurde.
wBranch2	Word	Folgeschritt, wenn Anzahl festgelegter Wiederholungen erreicht wurde.



Funktionsbibliothek LenzeCamControl.lib

Funktionsblöcke

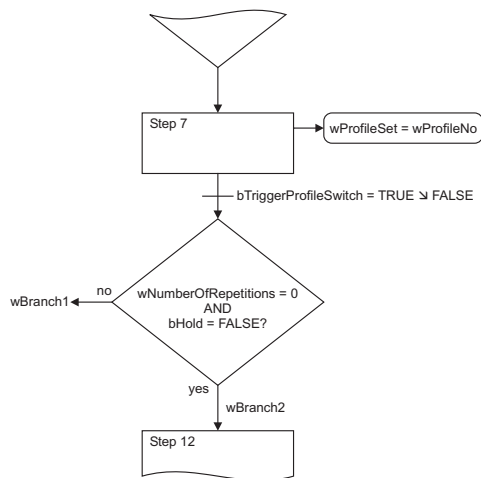
L_CamSequence - Vorgabe einer Profilsequenz

Die Anfangsadresse des Arrays muss dem FB am Eingang *pExternalData* über einen Adressoperator (**ADR**) mitgeteilt werden.

Es können max. 48 verschiedene Schritte (Array-Index 0 ... 47) für eine Sequenz nach folgendem Muster definiert werden:

```
PROGRAM SequenceInit
VAR
    MySequenceStep: ARRAY[0..47] OF CAM_SEQUENCE;
END VAR
```

```
MySequenceStep[7]
{
    wProfileNo := 17;
    wNumberOfRepetitions := 3000;
    wBranch1 := 7;
    wBranch2 := 12
}
```



Das obige Beispiel zeigt den Schritt Nr. 7 einer Sequenz.

- Das abzufahrende Sollprofil für diesen Schritt ist Profil Nr. 17 (*wProfileNo*).
- Das Profil soll insgesamt 3000 mal abgefahren werden (*wNumberOfRepetitions*).
- Die Wiederholungen erfolgen direkt hintereinander, da die Verzweigung 1 die gleiche Schrittnummer beinhaltet (*wBranch1*).
- Nach Abarbeitung der Wiederholungen folgt Schritt Nr. 12 (*wBranch2*).

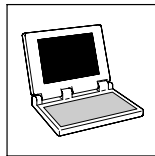


Tipp!

Änderungen an den Parametern für die Verzweigungen und Wiederholungen werden nur im Ausgangszustand (*bExecute* = FALSE) übernommen.

Sequenzende

Das Sequenzende wird durch die Verzweigung 2 (*wBranch2*) bestimmt. Ist in der Verzweigung 2 eines Schritts der Startschritt (*wStartStep*) angegeben, ist im Taktbetrieb (*bContinuous* = FALSE) die Sequenz beendet.



Taktbetrieb

Im Taktbetrieb wird die Sequenz genau einmal abgearbeitet.

- Durch Setzen von *bContinuous* auf FALSE wird der Taktbetrieb aktiviert.
- Durch Setzen von *bExecute* auf TRUE wird die Abarbeitung der Sequenz gestartet.
- Durch Rücksetzen von *bExecute* auf FALSE kann die laufende Abarbeitung unterbrochen, durch erneutes Setzen auf TRUE fortgesetzt werden (Zwischenstopp).
- Während der Abarbeitung ist *bBusy* = TRUE. Der Zustand FALSE signalisiert, dass die Abarbeitung der Sequenz beendet oder über *bExecute* unterbrochen wurde.
- Beim Start des letzten Profils der Sequenz wird *bLastRun* auf TRUE gesetzt. Über dieses Meldesignal können andere FBs (z. B. der FB **L_CamVMasterPosition**) gesteuert werden.
- Nach einem Durchlauf kann die Abarbeitung der Sequenz erneut über *bExecute* gestartet werden (Taktbetrieb).

Dauerbetrieb

Im Gegensatz zum Taktbetrieb wird im Dauerbetrieb die Sequenz fortlaufend abgearbeitet.

- Durch Setzen von *bContinuous* auf TRUE wird der Dauerbetrieb aktiviert.
- Durch Setzen von *bExecute* auf TRUE wird die Abarbeitung der Sequenz gestartet.
- Durch Rücksetzen von *bExecute* auf FALSE kann die laufende Abarbeitung unterbrochen, durch erneutes Setzen auf TRUE fortgesetzt werden (Zwischenstopp).
- Während der Abarbeitung ist *bBusy* = TRUE. Der Zustand FALSE signalisiert, dass die Abarbeitung der Sequenz beendet oder über *bExecute* unterbrochen wurde.
- Nach einem Sequenzdurchlauf wird automatisch wieder mit dem Schritt *wStartStep* begonnen.
- Über *wSequenceCounter* wird die Anzahl ausgegeben, wie oft die Sequenz bereits durchlaufen wurde.
- Durch Rücksetzen von *bContinuous* auf FALSE wird der Dauerbetrieb nach vollständigem Durchlauf der Sequenz beendet.

Hold-Funktion

Durch Setzen von *bHold* auf TRUE kann der Sequenzdurchlauf angehalten werden.

- Es wird dann immer nur das Sollprofil des aktuellen Schritts abgefahren.
- Während der Abarbeitung ist *bBusy* = TRUE.

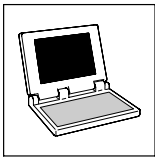
Reset-Funktion

Durch Setzen von *bReset* auf TRUE kann der Ausgangszustand (Startwerte) geladen werden.

- Sequenz- und Profilzähler werden zurückgesetzt.
- Die Reset-Funktion läuft über 4 Zyklen ab, der Ausgang *blnit* ist solange auf TRUE gesetzt.

Vorgabe des Startschritts

Im Ausgangszustand (*bExecute* = FALSE) kann über *wStartStep* jeder definierte Schritt innerhalb der Sequenz als Startschritt vorgegeben werden.



Funktionsbibliothek LenzeCamControl.lib

Funktionsblöcke

L_CamSequence - Vorgabe einer Profilsequenz

Beispiel-Sequenz

Initialisierung:

```

IF SYSTEM_b1LoopTask2 THEN
  MySequence[0].wProfileNo := 5;
  MySequence[0].wNumberOfRepetitions := 3;
  MySequence[0].wBranch1 := 1;
  MySequence[0].wBranch2 := 1;

  MySequence[1].wProfileNo := 4;
  MySequence[1].wNumberOfRepetitions := 3;
  MySequence[1].wBranch1 := 2;
  MySequence[1].wBranch2 := 2;

  MySequence[2].wProfileNo := 3;
  MySequence[2].wNumberOfRepetitions := 3;
  MySequence[2].wBranch1 := 0;
  MySequence[2].wBranch2 := 3;

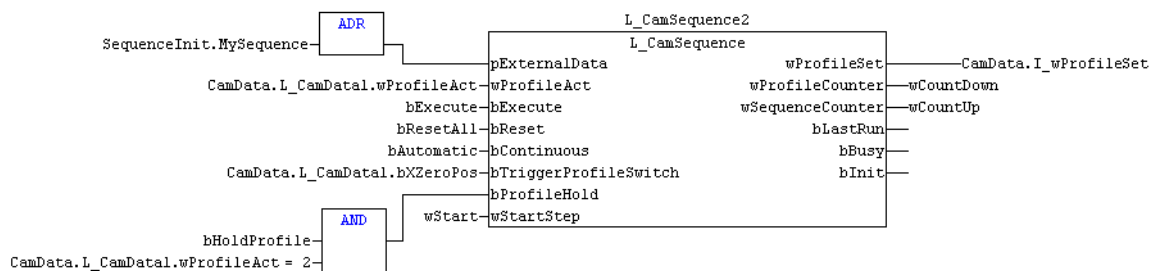
  MySequence[3].wProfileNo := 2;
  MySequence[3].wNumberOfRepetitions := 1;
  MySequence[3].wBranch1 := 3;
  MySequence[3].wBranch2 := 4;

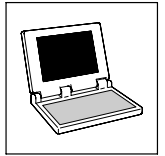
  MySequence[4].wProfileNo := 1;
  MySequence[4].wNumberOfRepetitions := 1;
  MySequence[4].wBranch1 := 4;
  MySequence[4].wBranch2 := 5;

  MySequence[5].wProfileNo := 0;
  MySequence[5].wNumberOfRepetitions := 1;
  MySequence[5].wBranch1 := 5;
  MySequence[5].wBranch2 := 0;
END IF

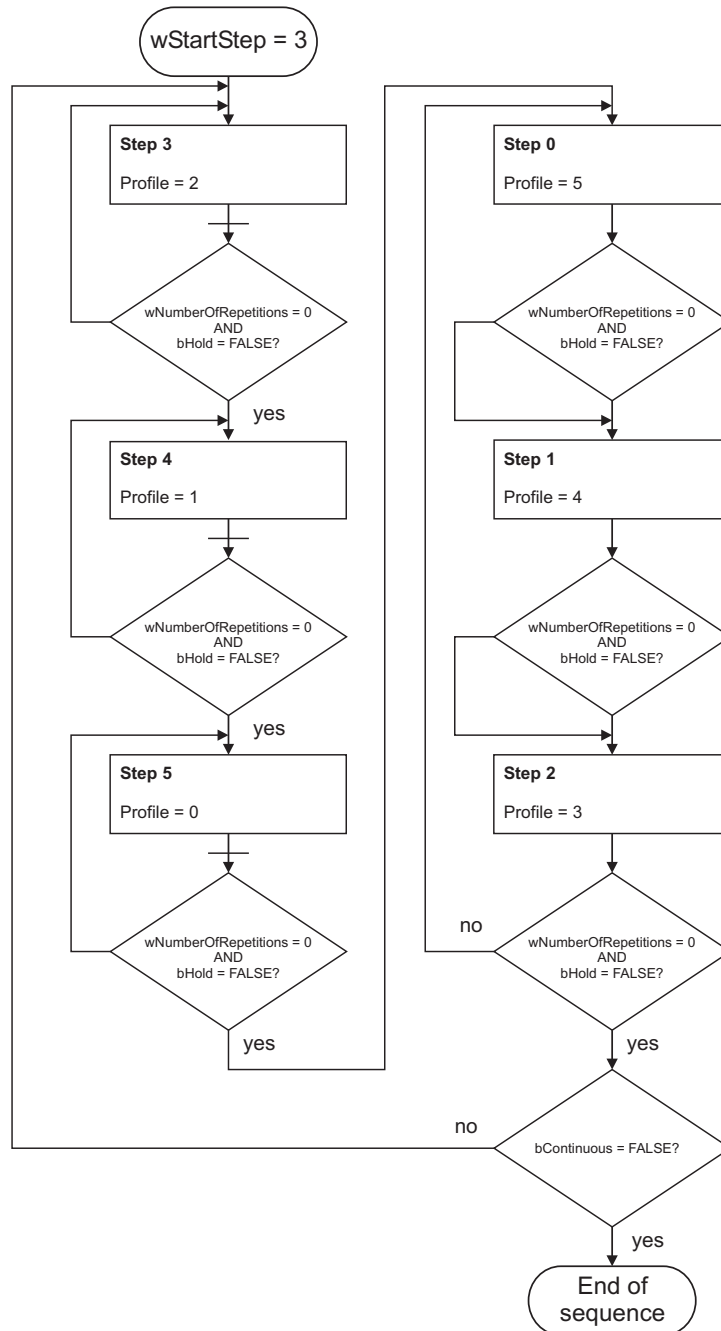
```

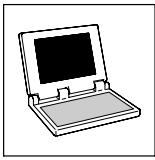
FB-Verschaltung:





Ablauf bei $wStartStep = 3$:





Funktionsbibliothek LenzeCamControl.lib

Funktionsblöcke

L_CamSetPosition - Einheitenumrechnung

2.14 L_CamSetPosition - Einheitenumrechnung

Dieser FB rechnet die physikalische Einheit einer Position in die interne Einheit um.

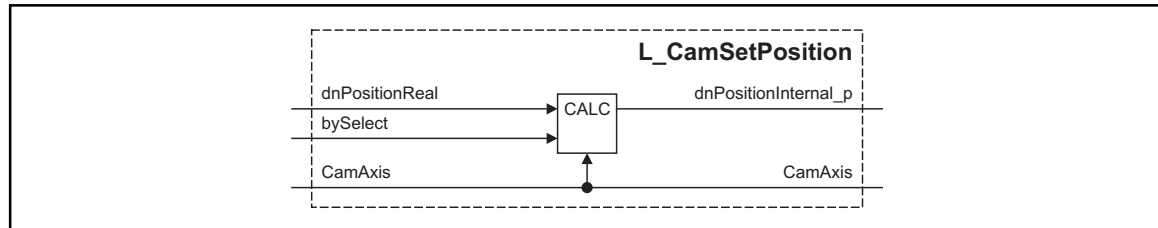


Abb. 2-37

Funktionsblock "L_CamSetPosition"

⇒ Eingänge (Variablentyp: VAR_INPUT)			
Bezeichner	Datentyp	Signaltyp	Info/Einstellmöglichkeiten
dnPositionReal	Double Integer	unit	Position in [unit] • 10000 ≡ 1.0000 [unit]
bySelect	Byte	number	Auswahl, ob Master- oder Slave-Positionen umgerechnet werden sollen
		0	Master-Positionen
		1	Slave-Positionen

Ausgänge ⇒ (Variablentyp: VAR_OUTPUT)			
Bezeichner	Datentyp	Signaltyp	Wert/Bedeutung
dnPositionInternal_p	Double Integer	position	Position in [inc] • 65536 ≡ 1 Motorumdrehung

Ein-/Ausgänge (Variablentyp: VAR_IN_OUT)			
Bezeichner	Datentyp	Signaltyp	Info
CamAxis	CAM_AXIS_REF	-	Maschinenparameter • Datenstruktur, deren Elemente die Maschinenparameter im internen Maßsystem beinhalten. (Beschreibung siehe FB L_CamMachineData. 2-42) • Verbinden Sie diesen Eingang mit einer globalen Variable, die vom FB L_CamMachineData beschrieben wird.

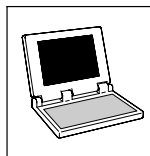
Formel

$$dnPositionInternal_p \text{ [inc]} = \frac{dnPositionReal \text{ [unit]}}{10000} \cdot \frac{65536 \text{ [inc/Umdr.]}}{\text{Vorschubkonstante [unit/Umdr.]}} \cdot \frac{\text{Getriebefaktor Zähler}}{\text{Getriebefaktornenner}}$$



Tipp!

- Rufen Sie diesen FB aus Rechenzeitgründen nur auf, wenn sich der Eingangswert *dnPositionReal* geändert hat.
- Der Eingangswert muss im Fixkommaformat mit 4 Nachkommastellen vorgegeben werden.
– Beispiel: 120.59 [unit] müssen als "1205900" vorgegeben werden.



2.15 L_CamSetXAxisVelocity - Dehnen und Stauchen des Leitwerts

Mit diesem FB können Sie die X-Achse dehnen/stauchen und eine Synchronisation über Touch-Probe durchführen. Die Leitfrequenz wird innerhalb der Taktlänge zur Position integriert.

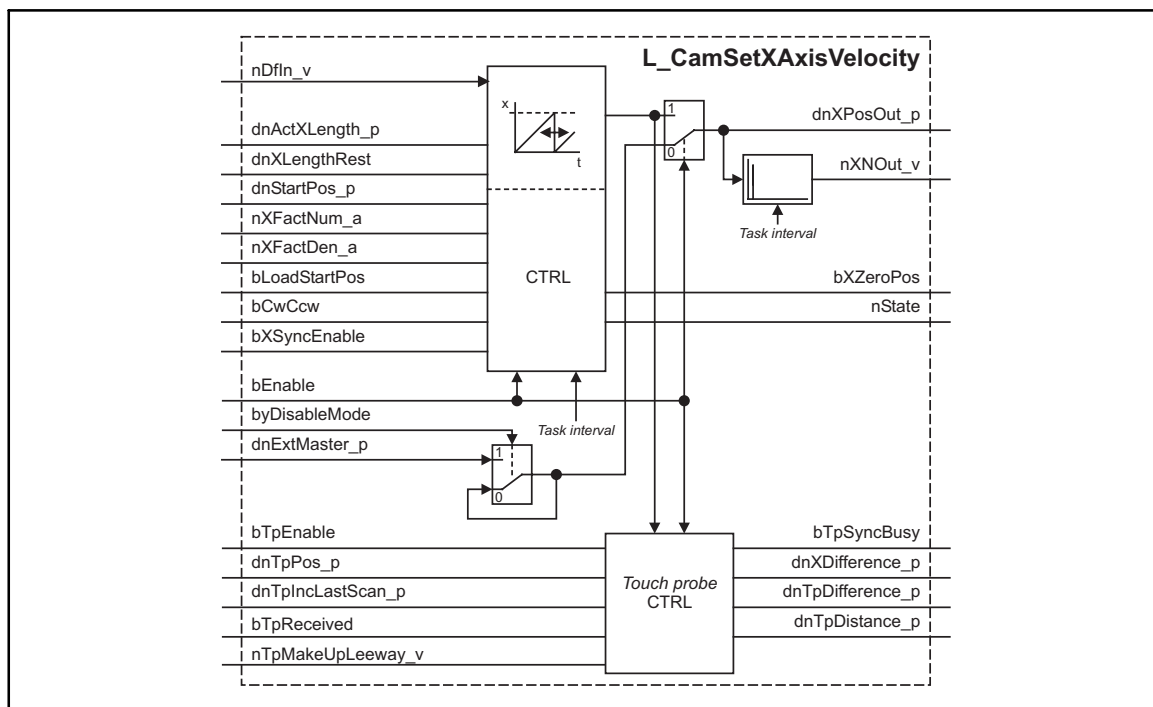
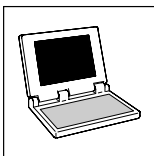


Abb. 2-38

Funktionsblock "L_CamSetXAxisVelocity"

⇒ Eingänge (Variablentyp: VAR_INPUT)							
Bezeichner	Datentyp	Signaltyp	Info/Einstellmöglichkeiten				
nDfln_v	Integer	velocity	Leitfrequenzeingang				
nTpMakeUpLeeway_v	Integer	velocity	Geschwindigkeit für Touch-Probe-Korrektur - Bei Vorgabe einer "0" bleibt die X-Differenz erhalten. - Für einen stoßfreien Ausgleich setzen Sie den FB L_CamPosCtrlLin ein.				
dnActXLength_p	Double Integer	position	Länge der X-Achse Verbinden Sie diesen Eingang mit dem Ausgang <i>dnActXLength_p</i> des FB L_CamData.				
dnXLengthRest	Double Integer	-	Nachkommenteil der X-Achse (<i>dnActXLength_p</i>) - Bei nicht ganzzahligen Getriebeverhältnis für die X-Achse - Dieser Wert wird vom FB L_CamProfileData zur Verfügung gestellt.				
dnStartPos_p	Double Integer	position	Startposition des Leitwertintegrators - Nur positive Werte zwischen 0 und <i>dnActXLength_p</i> möglich. - Übernahme erfolgt durch FALSE-TRUE-Flanke an <i>bLoadStartPos</i> (bei freigeschaltetem FB).				
nXFactNum_a	Integer	analog	Faktor für das Dehnen/Stauchen: Zähler - Das Produkt aus resultierenden Faktor, Leitfrequenz und Taskintervall (maximale Positionsänderung pro Zyklus) darf den Integer-Zahlenbereich nicht verlassen. - Änderungen werden "online" durchgeführt.				
nXFactDen_a	Integer	analog	Faktor für das Dehnen/Stauchen: Nenner - Nur Werte <> "0" vorgeben. - Änderungen werden "online" durchgeführt.				
bLoadStartPos	Bool	binary	Startposition in Leitwertintegrator laden <table><tr><td>TRUE</td><td>Startposition <i>dnStartPos_p</i> in Leitwertintegrator laden.</td></tr></table>	TRUE	Startposition <i>dnStartPos_p</i> in Leitwertintegrator laden.		
TRUE	Startposition <i>dnStartPos_p</i> in Leitwertintegrator laden.						
bCwCcw	Bool	binary	Invertierung der Königswelle (X-Achse) <table><tr><td>FALSE</td><td>Rechtslauf</td></tr><tr><td>TRUE</td><td>Linkslauf</td></tr></table>	FALSE	Rechtslauf	TRUE	Linkslauf
FALSE	Rechtslauf						
TRUE	Linkslauf						



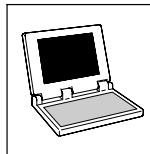
Funktionsbibliothek LenzeCamControl.lib

Funktionsblöcke

L_CamSetXAxisVelocity - Dehnen und Stauchen des Leitwerts

⇒ Eingänge (Variablentyp: VAR_INPUT)			
Bezeichner	Datentyp	Signaltyp	Info/Einstellmöglichkeiten
bXSyncEnable	Bool	binary	Synchronisiertes Dehnen/Stauchen aktivieren
			TRUE Ein neuer Dehnungs-/Stauchungsfaktor wird nur im Nulldurchgang übernommen.
			FALSE Ein neuer Dehnungs-/Stauchungsfaktor wird unmittelbar übernommen.
bEnable	Bool	binary	Funktionsblock freischalten
			FALSE Funktionsblock deaktiviert, Verhalten siehe <i>byDisableMode</i>
			TRUE Funktionsblock freigeschaltet
byDisableMode	Byte	number	Verhalten bei deaktiviertem Funktionsblock (<i>bEnable</i> = FALSE)
			0 Drehzahlausgang inaktiv, Position bleibt erhalten.
			1 Position an <i>dnExtMaster_p</i> wird übernommen, Drehzahlausgang aktiv.
dnExtMaster_p	Double Integer	position	Externer Leitwinkel • Dieser Eingang wird direkt auf den Ausgang <i>dnXPosOut_p</i> durchgeschaltet bei <i>bEnable</i> = FALSE und <i>byDisableMode</i> = "1". • Der Wert wird nicht auf die Taktlänge <i>dnActXLength_p</i> begrenzt!
bTpEnable	Bool	binary	Touch-Probe-Auswertung aktivieren TRUE Touch-Probe-Auswertung aktiviert
dnTpPos_p	Double Integer	position	Position des Touch-Probe-Signalgebers (Abstand zum Profilnullpunkt) • Wert wird intern begrenzt auf 0 ... 2 x <i>dnActXLength_p</i>
dnTpIncLastScan_p	Double Integer	position	Zahl der Inkremente, die seit dem Touch-Probe gezählt wurden. • Verbinden Sie diesen Eingang z. B. mit dem Ausgang <i>DFIN_dnIncLastScan_p</i> des SB DFIN_IO_DigitalFrequency .
bTpReceived	Bool	binary	Steuersignal für Touch-Probe-Korrektur, manuell oder über interruptfähigen Eingang • Beim manuellen Start wird kein Interrupt ausgelöst. Somit wird kein Wert vom SB DFIN_IO_DigitalFrequency am Eingang <i>dnTpIncLastScan_p</i> anliegen, der für eine genaue Synchronisation erforderlich wäre.
			TRUE Touch-Probe-Signal empfangen.

Ausgänge ⇒ (Variablentyp: VAR_OUTPUT)			
Bezeichner	Datentyp	Signaltyp	Wert/Bedeutung
dnXPosOut_p	Double Integer	position	Positionsausgang • Verbinden Sie diesen Ausgang beispielsweise mit dem Eingang <i>dnXPosIn_p</i> des FB L_CamData , falls beide in der gleichen Task operieren.
nXNOut_v	Integer	velocity	Drehzahlausgang Königswelle (X-Achse) Hinweis: Das Winkeldifferenzsignal berücksichtigt den inkrementellen Taktsprung von x_{max} auf 0 am Ende/Anfang eines Taktes, nicht jedoch einen eventuellen inkrementellen Nachkommaanteil in der Taktlänge. Verwenden Sie dieses Ausgangssignal folglich nicht, wenn die Taktlänge inkrementell nicht ganzzahlig darstellbar ist!
bXZeroPos	Bool	binary	Meldesignal "Profilnulldurchgang"
			TRUE Profilnulldurchgang
nState	Integer	number	Status
			13 Warnung: Resultierende Positionsänderung pro Zyklus ist zu groß für Normalbetrieb.
			12 Warnung: Nenner Dehn-/Stauchfaktor = 0
			11 Warnung: Startwert ungeeignet (<i>dnXStartPos_p</i> > <i>dnActXLength_p</i>)
			10 Warnung: Startwert ungeeignet (<i>dnXStartPos_p</i> < 0)
			0 OK, kein Fehler
			-10 Fehler: Taktlänge = "0"; kein Betrieb möglich.
			-11 Fehler: Resultierende Positionsänderung pro Zyklus ist zu groß für synchronisiertes Dehnen/Stauchen im Nulldurchgang.
			-20 Fehler: Touch-Probe-Position außerhalb der Taktlänge.



Ausgänge ⇒ (Variablentyp: VAR_OUTPUT)			
Bezeichner	Datentyp	Signaltyp	Wert/Bedeutung
bTpSyncBusy	Bool	binary	Meldesignal "Touch-Probe-Korrektur"
			TRUE Touch-Probe-Korrektur wird durchgeführt.
dnXDifference_p	Double Integer	position	Korrekturwert nach einem Touch-Probe-Signal Abweichung zu <i>dnTpPos_p</i> beim Touch-Probe.
dnTpDifference_p	Double Integer	position	Differenz zwischen zwei Touch-Probe-Ereignissen
dnTpDistance_p	Double Integer	position	Gemessener Abstand des Touch-Probe-Signals vom Kurvenulldurchgang

Funktionsblock freischalten und Leitfrequenz integrieren

Ist *bEnable* auf TRUE gesetzt, ist der Funktionsblock freigeschaltet und das Leitfrequenzsignal *nDfln_v* wird zu einer Position aufintegriert.

- Diese Position ist rein positiv und wird am Ausgang *dnXPosOut_p* in den Grenzen der Taktlänge ausgegeben.
- Die Taktlänge wird über *dnActXLength_p* vorgegeben.
– Verbinden Sie diesen Eingang mit dem Ausgang *dnActXLength_p* des FB **L_CamData**.
- Für die Integration kann ein Startwert an *dnStartPos_p* vorgeben werden. Die Übernahme dieses Wertes erfolgt durch Setzen von *bLoadStartPos* auf TRUE.



Hinweis!

Bei Kurvenumschaltung zwischen Kurven mit unterschiedlicher X-Achsenlänge darf sich der Leitwinkel nur in positive Richtung verdrehen.

Drehrichtung ändern

Über *bCwCcw* legen Sie die Drehrichtung des Leitwertes fest:

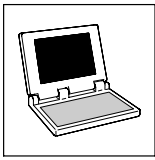
- FALSE: Drehrichtung ist positiv (Cw).
- TRUE: Drehrichtung ist negativ (Ccw).

Externes Master-Signal durchschleifen

Werden die Funktionalitäten des FBs nicht benötigt, kann der FB durch Setzen von *bEnable* auf FALSE inaktiv geschaltet werden.

Ist der FB inaktiv geschaltet und *byDisableMode* auf "1" gesetzt, kann ein an *dnExtMaster_p* angelegter externer Leitwinkel (z. B. das Positionssignal eines Winkelkodierers) direkt auf den Ausgang *dnXPosOut_p* geschaltet werden.

- Das Ausgangssignal wird hierbei nicht auf die Taktlänge begrenzt.
- Eine Differenzbildung wird durchgeführt, um auch die entsprechende Geschwindigkeit ausgeben zu können.
- Durch diese Funktion können Sie ohne großen Aufwand zu einem alternativen Leitwert wechseln, ohne den FB aus Rechenzeitgründen aus dem Projekt entfernen zu müssen.
- Wenn Sie durch Setzen von *bEnable* auf TRUE wieder zur Leitfrequenz wechseln, können Sie durch Setzen von *bLoadStartPos* auf TRUE einen neuen Startwert (*dnXStartPos_p*) laden.



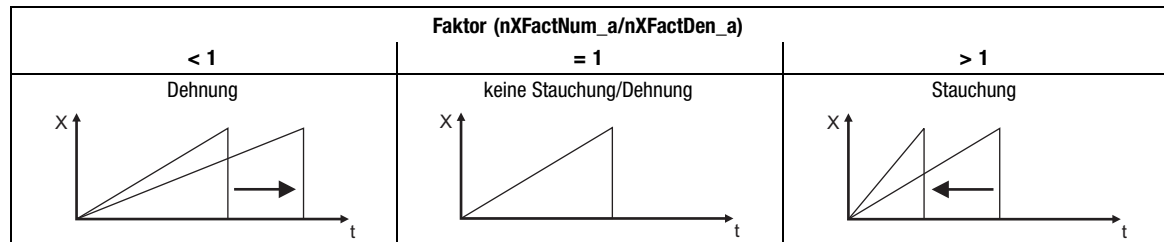
Funktionsbibliothek LenzeCamControl.lib

Funktionsblöcke

L_CamSetXAxisVelocity - Dehnen und Stauchen des Leitwerts

Dehnen/Stauchen des Leitwertsignals

Über $nXFactNum_a$ und $nXFactDen_a$ wird der Faktor festgelegt, um den die Leitwertachse gedehnt oder gestaucht werden soll:



Tipp!

Ein Dehnen/Stauchen bedeutet hierbei nicht die inkrementelle Verlängerung/Kürzung der X-Achse, sondern eine Veränderung der Geschwindigkeit, mit der die Leitwertachse durchlaufen wird.

Ein Dehnen/Stauchen hat also immer Einfluss auf den Geschwindigkeitsausgang $nXNOut_v$!

Synchronisiertes Dehnen und Stauchen

Das Dehnen/Stauchen erfolgt normalerweise "online", d. h. eine Änderung des Faktors wirkt sich umgehend auf den X-Geschwindigkeitsausgang aus, wobei jedoch Inkremente verloren gehen.

Um zu gewährleisten, dass keine Inkremente verloren gehen (z. B. für Vorschub-Applikationen), muss ein synchronisiertes Dehnen/Stauchen erfolgen.

- Synchronisiertes Dehnen/Stauchen bedeutet, dass eine Änderung des Dehnungs-/Stauchungsfaktors immer nur im Nulldurchgang der X-Achse übernommen wird.
- Durch Setzen von $bXSync$ auf TRUE aktivieren Sie das synchronisierte Dehnen/Stauchen.

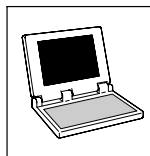


Hinweis!

Beachten Sie beim synchronisierten Dehnen/Stauchen ($bXSync = TRUE$):

- Die maximale Positionsänderung beträgt ± 32767 Inkremente. Das Produkt aus $nDfln_v$, Taskintervall und Dehn-/Stauchfaktor darf diesen Wert nicht überschreiten!

Kontrollieren Sie mit Ihren Maximalwerten das Ergebnis.



Synchronisation der X-Achse über Touch-Probe



Hinweis!

Im folgenden Text wird exemplarisch die Synchronisation mittels Touch-Probe beschrieben.

Bei der **9300 Servo PLC** kann eine Synchronisation über den Nullimpuls eines angeschlossenen Encoders oder über die digitalen Eingänge erfolgen.

Unabhängig davon kann die Touch-Probe-Korrektur auch manuell über den Eingang *bTpReceived* gestartet werden.

- Hierbei wird kein Interrupt ausgelöst. Somit wird kein Wert vom **SB DFIN_IO_DigitalFrequency** am Eingang *dnTpIncLastScan_p* anliegen, der für eine genaue Synchronisation erforderlich wäre.
- Die Synchronisationsgenauigkeit hängt somit vom Intervall der Task ab, in die der **FB L_CamSetXAxisVelocity** eingebunden ist.

Wird *bTpReceived* auf TRUE gesetzt (vom **SB DFIN_IO_DigitalFrequency** oder manuell) und ist *bTpEnable* auf TRUE gesetzt, wird die Differenz zwischen Soll- und Istwert berechnet und an *dnXDifference_p* ausgegeben.

- Die Position des Touch-Probe-Gebers wird über *dnTpPos_p* vorgegeben und muss sich im Bereich 0 ... 2 x *dnActXLength_p* befinden. Beim Vergleichswert handelt es sich um den aktuellen Wert am Positionsausgang *dnXPosOut_p*.
- Sollen eingehende Touch-Probe-Signale ausgeblendet werden, können Sie durch Setzen von *bTpEnable* auf FALSE die Touch-Probe-Synchronisation an entsprechender Stelle deaktivieren, z. B. mit Hilfe des Nockenschaltwerkes **L_CamContactor**. Dies kann nützlich sein, wenn auf Druckmarken oder Nullimpulse eines Gebers synchronisiert werden soll.
- Zum Ausgleich der Differenz kann über *nTpMakeUpLeeway_v* die Geschwindigkeit zum Aufholen vorgegeben werden. Wird keine Geschwindigkeit vorgegeben, wird die berechnete Differenz nicht abgebaut, d. h. die Synchronisation bleibt inaktiv.
- Für einen stoßfreien Ausgleich der Differenz schalten Sie an *nTpMakeUpLeeway_v* den **FB L_CamPosCtrlLin** als Rampengenerator vor:

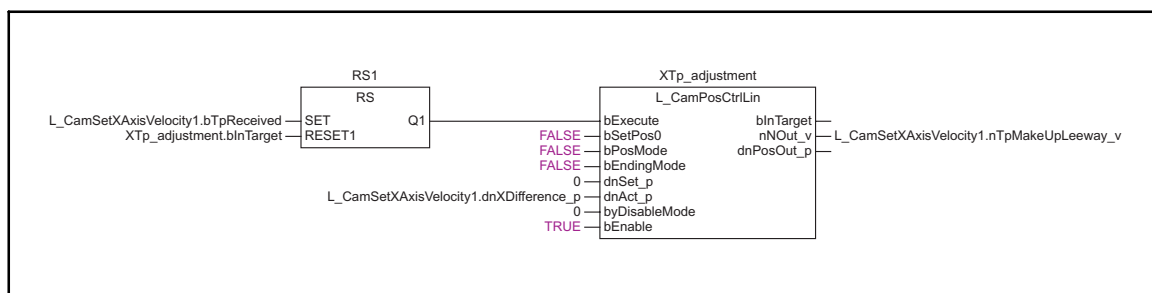


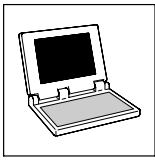
Abb. 2-39

Funktionsblock "L_CamPosCtrlLin" als Rampengenerator



Tipp!

Ausführliche Informationen zur Konfiguration des Touch-Probe (z. B. Aktivierung, Flankenauswahl und Verzögerungszeit) finden Sie im Handbuch zur **9300 Servo PLC**!



Funktionsbibliothek LenzeCamControl.lib

Funktionsblöcke

L_CamSetYAxisVelocity - Dehnen und Stauchen des Sollwerts

2.16

L_CamSetYAxisVelocity - Dehnen und Stauchen des Sollwerts

Mit diesem FB können Sie "online" die Sollposition des Antriebs durch Dehnen/Stauchen des Bewegungsprofils und das Hinzufügen eines Offset verändern.

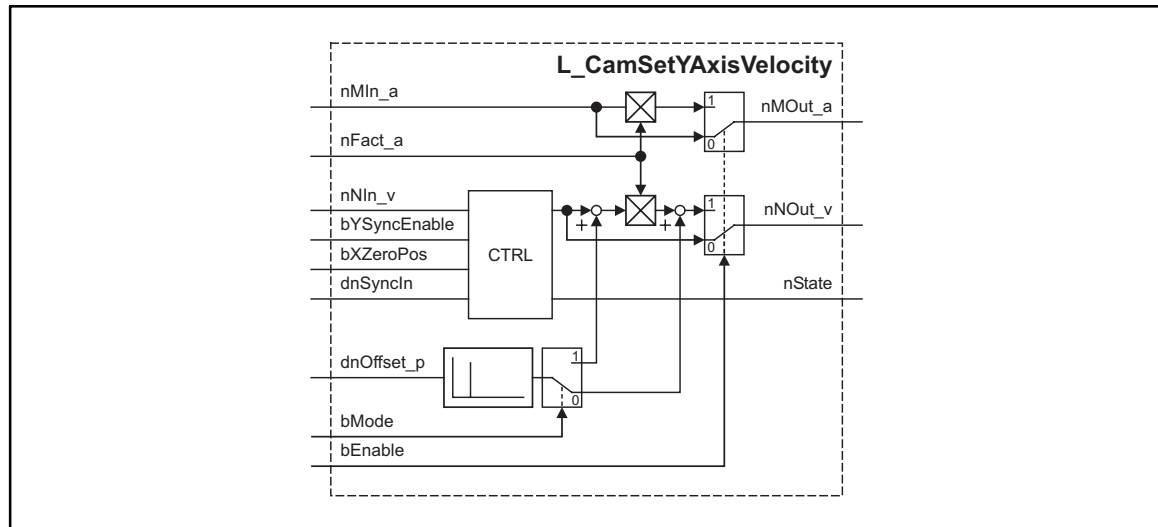
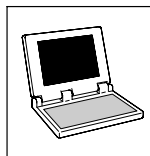


Abb. 2-40

Funktionsblock "L_CamSetYAxisVelocity"

⇒ Eingänge (Variablentyp: VAR_INPUT)									
Bezeichner	Datentyp	Signaltyp	Info/Einstellmöglichkeiten						
nMin_a	Integer	analog	Sollwert Drehmoment 16384 ≙ 100 % - Verbinden Sie diesen Eingang mit dem Ausgang <i>nMSet_a</i> des FB L_CamData.						
nFact_a	Integer	analog	Dehnungs-/Stauchungsfaktor für das Bewegungsprofil. 16384 ≙ 100 % - Bei Vorgabe negativer Werte Invertierung der Drehrichtung. - Bei <i>bMode</i> = TRUE auch Dehnungs-/Stauchungsfaktor für den Offset <i>dnOffset_p</i>. <table><tr><td>< 16384</td><td>Stauchung</td></tr><tr><td>= 16384</td><td>keine Dehnung/Stauchung (Faktor = 100 %)</td></tr><tr><td>> 16384</td><td>Dehnung</td></tr></table>	< 16384	Stauchung	= 16384	keine Dehnung/Stauchung (Faktor = 100 %)	> 16384	Dehnung
< 16384	Stauchung								
= 16384	keine Dehnung/Stauchung (Faktor = 100 %)								
> 16384	Dehnung								
nMin_v	Integer	velocity	Geschwindigkeit - Verbinden Sie diesen Eingang mit dem Ausgang <i>nNOut_v</i> des FB L_CamData. - Maximale Vorgabe: ±16384 [inc/ms] ≙ ±15000 [rpm]						
bYSyncEnable	Bool	position	Synchronisiertes Dehnen/Stauchen aktivieren <table><tr><td>FALSE</td><td>Ein neuer Dehnungs-/Stauchungsfaktor wird unmittelbar übernommen.</td></tr><tr><td>TRUE</td><td>Ein neuer Dehnungs-/Stauchungsfaktor wird nur im Nulldurchgang übernommen.</td></tr></table>	FALSE	Ein neuer Dehnungs-/Stauchungsfaktor wird unmittelbar übernommen.	TRUE	Ein neuer Dehnungs-/Stauchungsfaktor wird nur im Nulldurchgang übernommen.		
FALSE	Ein neuer Dehnungs-/Stauchungsfaktor wird unmittelbar übernommen.								
TRUE	Ein neuer Dehnungs-/Stauchungsfaktor wird nur im Nulldurchgang übernommen.								
bXZeroPos	Bool	position	Steuersignal für synchronisiertes Dehnen/Stauchen Verbinden Sie diesen Eingang mit dem Ausgang <i>bXZeroPos</i> des FB L_CamData. <table><tr><td>FALSE → TRUE</td><td>Übernahme des Dehnungs-/Stauchungsfaktors, wenn sich der Antrieb nicht im Stillstand befindet.</td></tr></table>	FALSE → TRUE	Übernahme des Dehnungs-/Stauchungsfaktors, wenn sich der Antrieb nicht im Stillstand befindet.				
FALSE → TRUE	Übernahme des Dehnungs-/Stauchungsfaktors, wenn sich der Antrieb nicht im Stillstand befindet.								
dnSyncln	Double Integer	-	Korrekturwert für synchronisiertes Dehnen/Stauchen Verbinden Sie diesen Eingang mit dem Ausgang <i>dnYSync</i> des FB L_CamData.						
dnOffset_p	Double Integer	position	Überlagerte Bewegung/Position zum Bewegungsprofil Bei <i>bMode</i> = TRUE wird dieser Wert mit dem Dehnungs-/Stauchungsfaktor <i>nFact_a</i> multipliziert am Ausgang ausgegeben.						
bMode	Bool	binary	Dehnung/Stauchung auch auf den Offset anwenden <table><tr><td>FALSE</td><td><i>dnOffset_p</i> wird unverändert auf den Ausgang addiert.</td></tr><tr><td>TRUE</td><td><i>dnOffset_p</i> wird auf das Drehzahlsignal addiert und mit dem Dehnungs-/Stauchungsfaktor <i>nFact_a</i> multipliziert am Ausgang <i>nNOut_v</i> ausgegeben.</td></tr></table>	FALSE	<i>dnOffset_p</i> wird unverändert auf den Ausgang addiert.	TRUE	<i>dnOffset_p</i> wird auf das Drehzahlsignal addiert und mit dem Dehnungs-/Stauchungsfaktor <i>nFact_a</i> multipliziert am Ausgang <i>nNOut_v</i> ausgegeben.		
FALSE	<i>dnOffset_p</i> wird unverändert auf den Ausgang addiert.								
TRUE	<i>dnOffset_p</i> wird auf das Drehzahlsignal addiert und mit dem Dehnungs-/Stauchungsfaktor <i>nFact_a</i> multipliziert am Ausgang <i>nNOut_v</i> ausgegeben.								



⇒ Eingänge (Variablentyp: VAR_INPUT)							
Bezeichner	Datentyp	Signaltyp	Info/Einstellmöglichkeiten				
bEnable	Bool	binary	Dehnungs-/Stauchungsfunktion freischalten				
			<table><tr><td>FALSE</td><td>Die Dehnungs-/Stauchungsfunktion ist deaktiviert. • <i>nMIn_a</i> wird direkt an <i>nMOut_a</i> ausgegeben. • Die Summe aus <i>nNIn_v</i> und <i>dnSyncln</i> wird direkt an <i>nNOut_v</i> ausgegeben. • Der Offset <i>dnOffset_p</i> wird nicht berücksichtigt. </td></tr><tr><td>TRUE</td><td>Die Dehnungs-/Stauchungsfunktion ist freigeschaltet.</td></tr></table>	FALSE	Die Dehnungs-/Stauchungsfunktion ist deaktiviert. • <i>nMIn_a</i> wird direkt an <i>nMOut_a</i> ausgegeben. • Die Summe aus <i>nNIn_v</i> und <i>dnSyncln</i> wird direkt an <i>nNOut_v</i> ausgegeben. • Der Offset <i>dnOffset_p</i> wird nicht berücksichtigt.	TRUE	Die Dehnungs-/Stauchungsfunktion ist freigeschaltet.
FALSE	Die Dehnungs-/Stauchungsfunktion ist deaktiviert. • <i>nMIn_a</i> wird direkt an <i>nMOut_a</i> ausgegeben. • Die Summe aus <i>nNIn_v</i> und <i>dnSyncln</i> wird direkt an <i>nNOut_v</i> ausgegeben. • Der Offset <i>dnOffset_p</i> wird nicht berücksichtigt.						
TRUE	Die Dehnungs-/Stauchungsfunktion ist freigeschaltet.						

Ausgänge ⇒ (Variablentyp: VAR_OUTPUT)							
Bezeichner	Datentyp	Signaltyp	Wert/Bedeutung				
nMOut_a	Integer	analog	Drehmoment • 16384 ≙ 100 % • Abhängig von <i>bMode</i> und <i>nFact_a</i>. • Verbinden Sie diesen Ausgang mit dem Eingang <i>nMSet_a</i> des FB L_CamControl.				
nNOut_v	Integer	velocity	Geschwindigkeitssollwert • Verbinden Sie diesen Ausgang mit dem Eingang <i>nMainIn_v</i> des FB L_CamControl.				
nState	Integer	number	Status <table><tr><td>20</td><td>Begrenzung Drehzahlausgang <i>nNOut_v</i> aktiv.</td></tr><tr><td>0</td><td>OK, kein Fehler.</td></tr></table>	20	Begrenzung Drehzahlausgang <i>nNOut_v</i> aktiv.	0	OK, kein Fehler.
20	Begrenzung Drehzahlausgang <i>nNOut_v</i> aktiv.						
0	OK, kein Fehler.						

Dehnungs-/Stauchungsfunktion freischalten

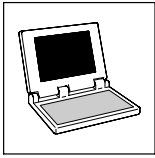
Ist *bEnable* auf TRUE gesetzt, ist die Dehnungs-/Stauchungsfunktion sowie Vorgabe eines Offset freigeschaltet. Bei FALSE werden die Drehmomentvorsteuerung und die Geschwindigkeit direkt zu den entsprechenden Ausgängen durchgeschliffen.

Dehnen & Stauchen

Über den Dehnungs-/Stauchungsfaktor kann das Bewegungsprofil "online" im Hub verändert werden.

Beispiel für die Veränderung des Hub an einer mechanischen Kurvenscheibe:	Dehnungs-/Stauchungsfaktor		Dehnung/Stauchung	Richtungsumkehr
	Vorgabe an nFact_a	≙ Prozentwert		
	+16384	+100 %	nein	nein
	-16384	-100 %	nein	ja, in Y-Position
	> 16384	>100 %	Dehnung	nein
	< 16384	<100 %	Stauchung	nein

- Der Dehnungs-/Stauchungsfaktor darf nur im Stillstand des Motors und auf der Nullposition des Bewegungsprofils umgeschaltet werden.
- Ist eine Umschaltung während der Bewegung erforderlich, verwenden Sie hierfür die Funktion "Synchronisiertes Dehnen/Stauchen".



Funktionsbibliothek LenzeCamControl.lib

Funktionsblöcke

L_CamSetYAxisVelocity - Dehnen und Stauchen des Sollwerts

Beispiele

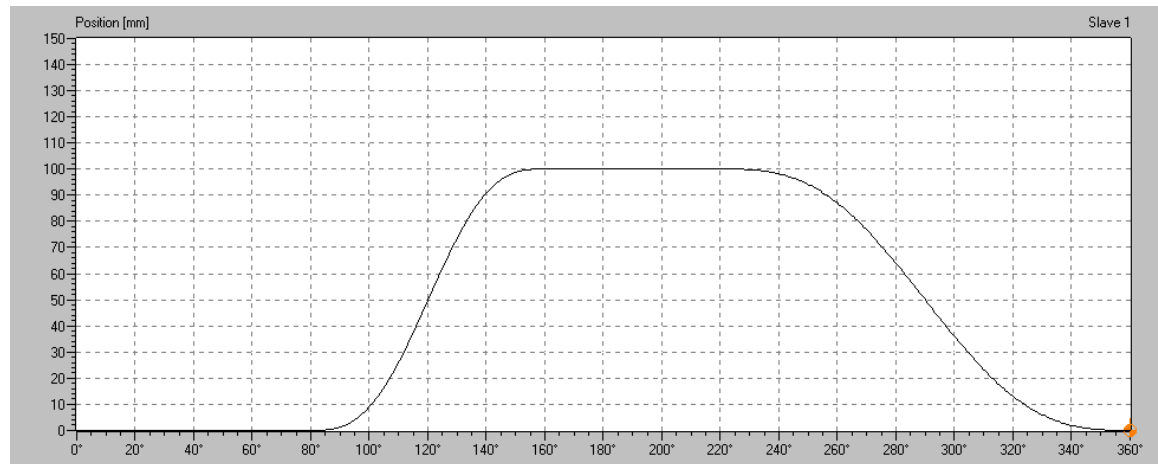


Abb. 2-41

$nFact_a = 16384$ (100 % von 100 mm Hub)

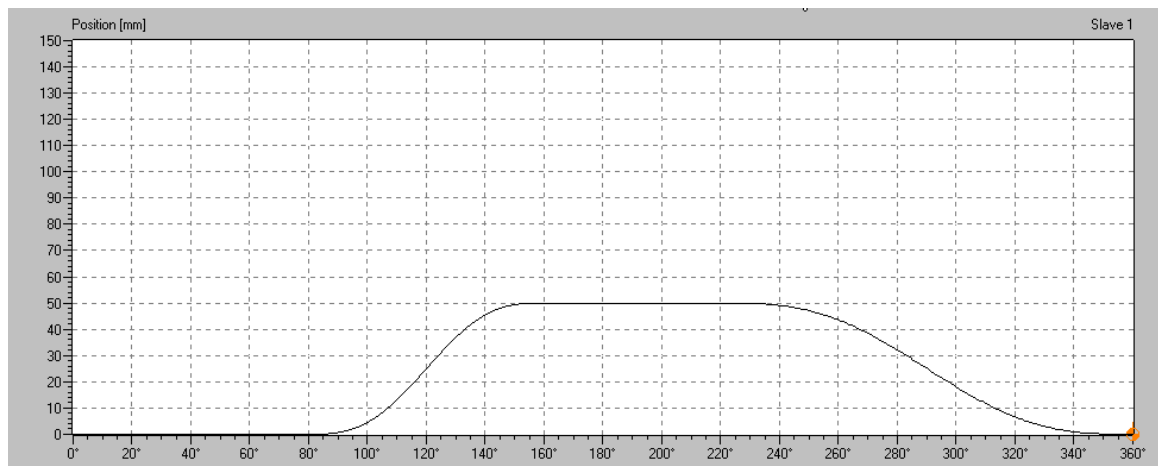


Abb. 2-42

Stauchung: $nFact_a = 8192$ (50 % von 100 mm Hub)

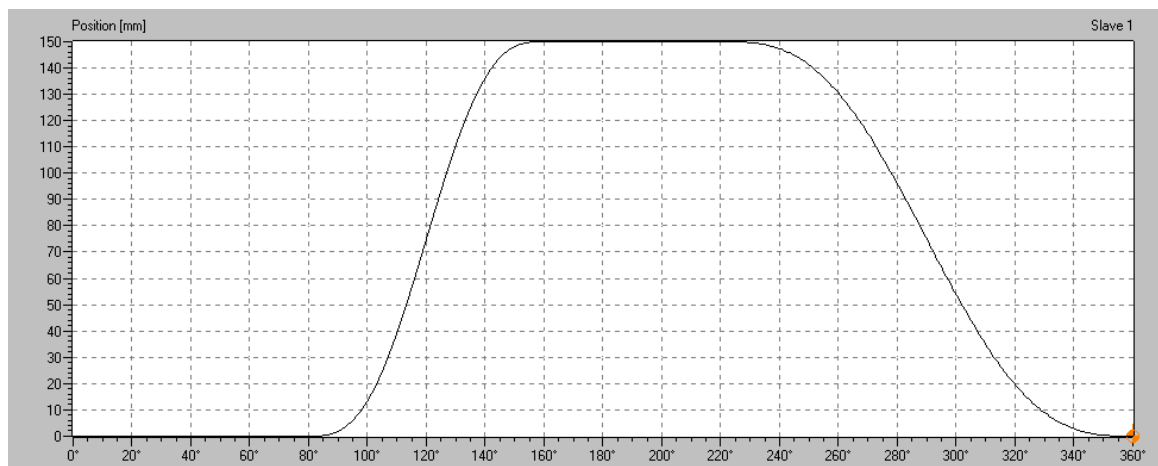
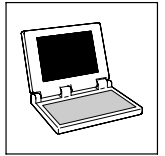


Abb. 2-43

Dehnung: $nFact_a = 24576$ (150 % von 100 mm Hub)



Synchronisiertes Dehnen/Stauchen

Ein neuer Dehnungs-/Stauchungsfaktor am Eingang *nFact_a* wird unmittelbar übernommen und abgearbeitet. Hat zu diesem Zeitpunkt das Bewegungsprofil keine Rast (Steigung ≤ 0), führt die neue Faktorisierung zum Verlust der Y-Nullposition bzw. zum Inkrementenverlust bei einem Vorschubprofil.

Synchronisiertes Dehnen/Stauchen ist daher immer dann erforderlich, wenn die Dehnung/Stauchung des Profils während der Produktbearbeitung geändert werden soll. Ein neuer Dehen-/ Stauchfaktor *nFact_a* wird nur im Kurvenulldurchgang übernommen.

Die Synchronisation erfolgt in Verbindung mit dem FB **L_CamData** im Kurvenulldurchgang des Profils, wobei folgende Voraussetzungen erfüllt sein müssen:

- Beim FB **L_CamData** ist *bYSyncEnable* ebenfalls auf TRUE gesetzt und dadurch die Synchronisation aktiviert.
- Der Eingang *dnSyncIn* ist mit dem Ausgang *dnYSync* des FB **L_CamData** verbunden.
- Der Eingang *bSync* ist mit dem Ausgang *bXZeroPos* des FB **L_CamData** verbunden.



Hinweis!

Aktivieren Sie das synchronisierte Dehnen/Stauchen durch Setzen von *bYSyncEnable* auf TRUE erst, wenn der Antrieb bereits auf der Kurve positioniert ist.

Offset

Über *dnOffset_p* können Sie einen Offset vorgeben, um eine Verschiebung der Y-Position durchzuführen:

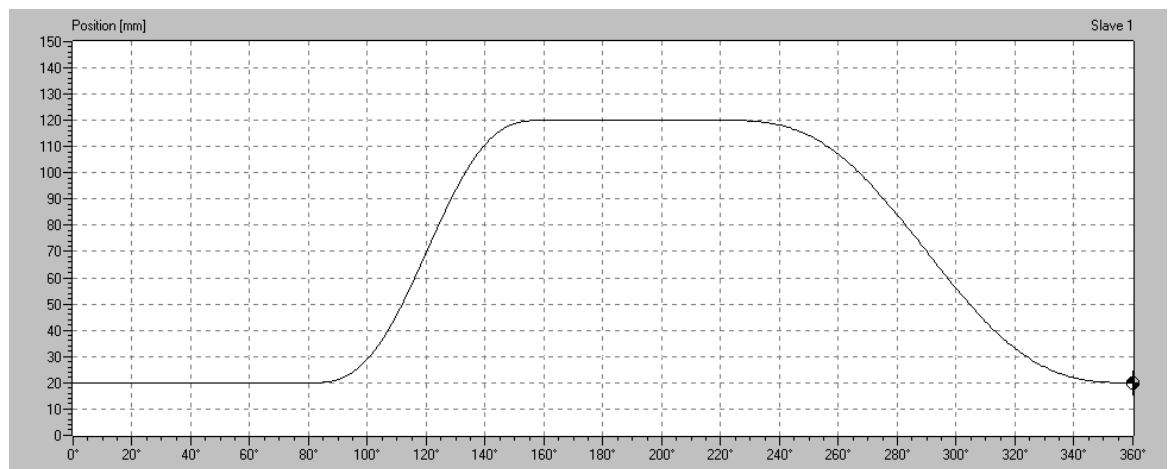
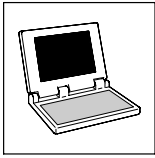


Abb. 2-44

Verschiebung des Profils um 20 mm in Y-Position durch Vorgabe eines Offset.

Ist *bMode* auf TRUE gesetzt, wird der Offset *dnOffset_p* mit dem Dehnungs-/Stauchungsfaktor *nFact_a* multipliziert und am Ausgang ausgegeben.



Funktionsbibliothek LenzeCamControl.lib

Funktionsblöcke

L_CamVMasterPosition - Virtueller Master mit Positionierfunktion

2.17 L_CamVMasterPosition - Virtueller Master mit Positionierfunktion

Über diesen FB können folgende Funktionen realisiert werden:

- Generierung und Positionierung der Königswelle (X-Achse)
- Normalbetrieb oder Taktbetrieb mit Stopp über einstellbare Rampe in gewünschter Position
- Handfahrbetrieb (Tippbetrieb)
- Handradfunktion über Inkrementalgeber
- Nothalt

Zusammenhang zwischen Maschinentakt und Virtuellem Master

Auf Grund der Getriebeübersetzung und der Differenz zwischen Vorschubkonstante und X-Kurvenlänge muss die geforderte Taktzahl auf die Königswelle des Masters umgerechnet werden.

Demnach ergibt sich die einzustellende Leitdrehzahl im Virtuellen Master nach der Formel:

$$\text{Leitdrehzahl [rpm]} = \text{Taktzahl [1/min]} \cdot \frac{\text{X-Kurvenlänge [mm]}}{\text{Vorschubkonstante [mm]}} \cdot \frac{\text{Getriebezahl}}{\text{Getriebebenner}}$$

Verhältnis x-Kurvenlänge zu Vorschubkonstante	
$\frac{\text{X-Kurvenlänge [mm]}}{\text{Vorschubkonstante [mm]}}$	Sind X-Kurvenlänge und Vorschubkonstante unterschiedlich, entspricht eine Umdrehung der Vorschubwalze nicht einem kompletten Takt. Ein entsprechendes Verhältnis von X-Kurvenlänge (ΔX) zu Vorschubkonstante ($\pi \times d$) muss berücksichtigt werden.

Getriebeübersetzung	
$\frac{\text{Getriebezahl}}{\text{Getriebebenner}}$	Dreht der Motor des Masterantriebs um die Getriebeübersetzung schneller als die Vorschubwalze, muss eine entsprechende Übersetzung berücksichtigt werden.

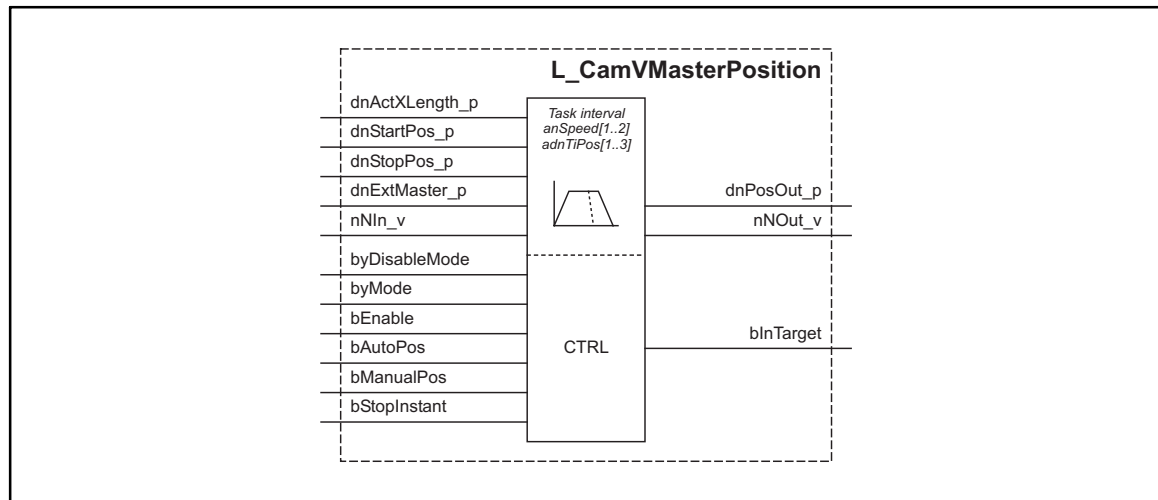
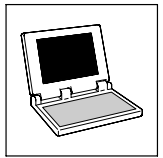


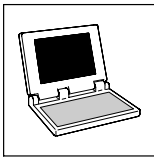
Abb. 2-45

Funktionsblock "L_CamVMasterPosition"

⇒ Eingänge (Variablentyp: VAR_INPUT)			
Bezeichner	Datentyp	Signaltyp	Info/Einstellmöglichkeiten
dnActXLength_p	Double Integer	position	Länge der X-Achse in Inkrementen
dnStartPos_p	Double Integer	position	Start-Position
dnStopPos_p	Double Integer	position	Stopp-Position
dnExtMaster_p	Double Integer	position	Position eines externen Masters (z. B. eines Absolutwertgebers) • Begrenzung: 0 ... dnActXLength_p



⇒ Eingänge (Variablentyp: VAR_INPUT)													
Bezeichner	Datentyp	Signaltyp	Info/Einstellmöglichkeiten										
nNIn_v	Integer	velocity	Leitfrequenzsollwert										
byDisableMode	Byte	number	Verhalten bei nicht freigeschaltetem Funktionsblock (<i>bEnable</i> = FALSE) <table><tr><td>0</td><td>Drehzahlausgang inaktiv, Position bleibt erhalten.</td></tr><tr><td>1</td><td>Drehzahlausgang inaktiv, Position wird auf Start-Position gesetzt.</td></tr><tr><td>2</td><td>Drehzahlausgang aktiv, Handradfunktion (über Inkrementalgeber an <i>nNIn_v</i>)</td></tr><tr><td>3</td><td>Drehzahlausgang aktiv, Position wird auf Null gesetzt.</td></tr><tr><td>4</td><td>Drehzahlausgang aktiv, Durchschleifen der Position an <i>dnExtMaster_p</i>.</td></tr></table>	0	Drehzahlausgang inaktiv, Position bleibt erhalten.	1	Drehzahlausgang inaktiv, Position wird auf Start-Position gesetzt.	2	Drehzahlausgang aktiv, Handradfunktion (über Inkrementalgeber an <i>nNIn_v</i>)	3	Drehzahlausgang aktiv, Position wird auf Null gesetzt.	4	Drehzahlausgang aktiv, Durchschleifen der Position an <i>dnExtMaster_p</i> .
0	Drehzahlausgang inaktiv, Position bleibt erhalten.												
1	Drehzahlausgang inaktiv, Position wird auf Start-Position gesetzt.												
2	Drehzahlausgang aktiv, Handradfunktion (über Inkrementalgeber an <i>nNIn_v</i>)												
3	Drehzahlausgang aktiv, Position wird auf Null gesetzt.												
4	Drehzahlausgang aktiv, Durchschleifen der Position an <i>dnExtMaster_p</i> .												
byMode	Byte	number	Funktionsauswahl <table><tr><td>0</td><td>Normalbetrieb - Aktivierung durch <i>bAutoPos</i> = TRUE </td></tr><tr><td>1</td><td>Taktbetrieb - Aktivierung durch FALSE-TRUE-Flanke an <i>bAutoPos</i> </td></tr><tr><td>2</td><td>Handbetrieb in positiver Richtung (Cw) - Aktivierung durch <i>bManualPos</i> = TRUE </td></tr><tr><td>3</td><td>Handbetrieb in negativer Richtung (Ccw) - Aktivierung durch <i>bManualPos</i> = TRUE </td></tr><tr><td>4</td><td>Positionierung auf <i>dnExtMaster_p</i> - Aktivierung durch FALSE-TRUE-Flanke an <i>bAutoPos</i> </td></tr></table>	0	Normalbetrieb Aktivierung durch <i>bAutoPos</i> = TRUE	1	Taktbetrieb Aktivierung durch FALSE-TRUE-Flanke an <i>bAutoPos</i>	2	Handbetrieb in positiver Richtung (Cw) Aktivierung durch <i>bManualPos</i> = TRUE	3	Handbetrieb in negativer Richtung (Ccw) Aktivierung durch <i>bManualPos</i> = TRUE	4	Positionierung auf <i>dnExtMaster_p</i> Aktivierung durch FALSE-TRUE-Flanke an <i>bAutoPos</i>
0	Normalbetrieb Aktivierung durch <i>bAutoPos</i> = TRUE												
1	Taktbetrieb Aktivierung durch FALSE-TRUE-Flanke an <i>bAutoPos</i>												
2	Handbetrieb in positiver Richtung (Cw) Aktivierung durch <i>bManualPos</i> = TRUE												
3	Handbetrieb in negativer Richtung (Ccw) Aktivierung durch <i>bManualPos</i> = TRUE												
4	Positionierung auf <i>dnExtMaster_p</i> Aktivierung durch FALSE-TRUE-Flanke an <i>bAutoPos</i>												
bEnable	Bool	binary	Funktionsblock freischalten <table><tr><td>FALSE</td><td>Funktionsblock gesperrt, Verhalten siehe <i>byDisableMode</i></td></tr><tr><td>TRUE</td><td>Funktionsblock freigeschaltet</td></tr></table>	FALSE	Funktionsblock gesperrt, Verhalten siehe <i>byDisableMode</i>	TRUE	Funktionsblock freigeschaltet						
FALSE	Funktionsblock gesperrt, Verhalten siehe <i>byDisableMode</i>												
TRUE	Funktionsblock freigeschaltet												
bAutoPos	Bool	binary	Automatikbetrieb mit Stopp in gewünschter Position aktivieren <table><tr><td>TRUE</td><td>Im Normalbetrieb (<i>byMode</i> = 0): Normalbetrieb starten - Die Königswelle läuft über Rampe hoch auf Sollgeschwindigkeit. </td></tr><tr><td>TRUE ↘ FALSE</td><td>Im Normalbetrieb (<i>byMode</i> = 0): - Die Königswelle läuft über Rampe herunter auf die Stopp-Position (automatischer Stopp-Vorgang). </td></tr><tr><td>FALSE ↗ TRUE</td><td>Im Taktbetrieb (<i>byMode</i> = 1): Normalbetrieb stoppen - Die Königswelle läuft über Rampe hoch auf Sollgeschwindigkeit. - Der automatische Stopp-Vorgang wird sofort eingeleitet. </td></tr></table>	TRUE	Im Normalbetrieb (<i>byMode</i> = 0): Normalbetrieb starten Die Königswelle läuft über Rampe hoch auf Sollgeschwindigkeit.	TRUE ↘ FALSE	Im Normalbetrieb (<i>byMode</i> = 0): Die Königswelle läuft über Rampe herunter auf die Stopp-Position (automatischer Stopp-Vorgang).	FALSE ↗ TRUE	Im Taktbetrieb (<i>byMode</i> = 1): Normalbetrieb stoppen - Die Königswelle läuft über Rampe hoch auf Sollgeschwindigkeit. - Der automatische Stopp-Vorgang wird sofort eingeleitet.				
TRUE	Im Normalbetrieb (<i>byMode</i> = 0): Normalbetrieb starten Die Königswelle läuft über Rampe hoch auf Sollgeschwindigkeit.												
TRUE ↘ FALSE	Im Normalbetrieb (<i>byMode</i> = 0): Die Königswelle läuft über Rampe herunter auf die Stopp-Position (automatischer Stopp-Vorgang).												
FALSE ↗ TRUE	Im Taktbetrieb (<i>byMode</i> = 1): Normalbetrieb stoppen - Die Königswelle läuft über Rampe hoch auf Sollgeschwindigkeit. - Der automatische Stopp-Vorgang wird sofort eingeleitet.												
bManualPos	Bool	binary	Handfahren aktivieren - Ohne automatische Positionierung - Dieser Eingang hat eine höhere Priorität als <i>bAutoPos</i> (Normalbetrieb/Taktbetrieb) <table><tr><td>FALSE</td><td>kein Handfahren</td></tr><tr><td>TRUE</td><td>Handfahren (Tippbetrieb) - Wurde die Fahrrichtung nicht explizit über <i>byMode</i> = 2/3 festgelegt, wird in positiver Richtung (Cw) verfahren. </td></tr></table>	FALSE	kein Handfahren	TRUE	Handfahren (Tippbetrieb) Wurde die Fahrrichtung nicht explizit über <i>byMode</i> = 2/3 festgelegt, wird in positiver Richtung (Cw) verfahren.						
FALSE	kein Handfahren												
TRUE	Handfahren (Tippbetrieb) Wurde die Fahrrichtung nicht explizit über <i>byMode</i> = 2/3 festgelegt, wird in positiver Richtung (Cw) verfahren.												
bStopInstant	Bool	binary	Stopp-Funktion: Zwangsstopp über separate Verzögerungsrampe Dieser Eingang hat die höchste Priorität <table><tr><td>TRUE</td><td>Die momentane Verfahrfunktion wird unterbrochen und der Antrieb wird über die eingestellte Verzögerungsrampe in den Stillstand geführt.</td></tr></table>	TRUE	Die momentane Verfahrfunktion wird unterbrochen und der Antrieb wird über die eingestellte Verzögerungsrampe in den Stillstand geführt.								
TRUE	Die momentane Verfahrfunktion wird unterbrochen und der Antrieb wird über die eingestellte Verzögerungsrampe in den Stillstand geführt.												



Funktionsbibliothek LenzeCamControl.lib

Funktionsblöcke

L_CamVMasterPosition - Virtueller Master mit Positionierfunktion

Ausgänge ⇒ (Variablentyp: VAR_OUTPUT)				
Bezeichner	Datentyp	Signaltyp	Wert/Bedeutung	
dnPosOut_p	Double Integer	position	Position (innerhalb der Grenzen der X-Achse)	
nNOut_v	Integer	velocity	Leitfrequenz	
			Hinweis: Das Winkeldifferenzsignal berücksichtigt den inkrementellen Taktsprung von $x_{\max}$ auf 0 am Ende/Anfang eines Taktes, nicht jedoch einen eventuellen inkrementellen Nachkommaanteil in der Taktlänge. Verwenden Sie dieses Ausgangssignal folglich nicht, wenn die Taktlänge inkrementell nicht ganzzahlig darstellbar ist!	
blnTarget	Bool	binary	Meldesignal: Aktuelle Position <i>dnPosOut_p</i> hat die Stoppposition <i>dnStopPos_p</i> erreicht. Signal steht solange an, bis eine neue Verfahraction gewählt wird.	
			FALSE	Die aktuelle Position <i>dnPosOut_p</i> hat die Stoppposition erstmalig (nach Initialisierung) noch nicht verlassen bzw. befindet sich nicht auf der Stoppposition <i>dnStopPos_p</i> .
			TRUE	Die aktuelle Position <i>dnPosOut_p</i> hat die Stoppposition <i>dnStopPos_p</i> angefahren.
Hinweis: Der Ausgang wird nicht auf TRUE gesetzt, wenn der virtuelle Master nach Initialisierung des Zielsystems zwar in der Stoppposition steht, er diese aber noch nicht verlassen hat.				

Parameter (Variablentyp: VAR_CONSTANT_RETAIN)				
Bezeichner	Datentyp	Signaltyp	Info	
Codestelle*			Einstellmöglichkeiten	Voreinstellung
adnTiPos[1..3]	Array of Double Integer	-	Rampen für Beschleunigung/Verzögerung (bezogen auf 4000 rpm)	
			adnTiPos[1] = Verzögerungsrampe für Stopp-Funktion	
C1452/1			0.010 {0.001 s} 130.000	1.000 s
			adnTiPos[2] = Beschleunigungs-/Verzögerungsrampe für Normalbetrieb	
C1452/2			0.010 {0.001 s} 130.000	1.000 s
			adnTiPos[3] = Beschleunigungs-/Verzögerungsrampe für Handfahren	
C1452/3			0.010 {0.001 s} 130.000	1.000 s
anSpeed[1..2]	Array of Integer	-	Geschwindigkeit für Handfahren	
			anSpeed[1] = Geschwindigkeit für Handfahren in positiver Richtung (Cw)	
C1451/1			-15000 {1 rpm} 15000	10 rpm
			anSpeed[2] = Geschwindigkeit für Handfahren in negativer Richtung (Ccw)	
C1451/2			-15000 {1 rpm} 15000	10 rpm

* Codestelle der Erstinstanz

* Codestelle der Erstinstanz

Generierung und Positionierung der Königswelle (X-Achse)

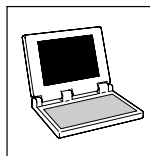
Die Sollwertachse (X-Achse) wird anhand der an *dnActXLength_p* vorgegebenen Länge generiert.

- Die X-Achse ist rein positiv, negative Werte sind nicht zulässig (interne Begrenzung).
- Durch Vorgabe einer negativen Geschwindigkeit am Eingang *nNIn_v* kann die X-Achse rückwärts durchlaufen werden.
- Ein Startwert ungleich Null kann über den Eingang *dnStartPos_p* festgelegt werden. Dieser Startwert muss kleiner als die Länge der X-Achse sein. Eine Wertänderung ist nur wirksam bei *byDisableMode* = 1 und nicht freigeschalteten FB.



Hinweis!

Bei Kurvenumschaltung zwischen Kurven mit unterschiedlicher X-Achsenlänge darf sich der Leitwinkel nur in positive Richtung verdrehen.



Verfahrfunktion aktivieren

Die über *byMode* eingestellte Verfahrfunktion des FB kann nur gestartet werden, wenn der FB über *bEnable* = TRUE freigeschaltet ist.

Normalbetrieb (mit automatischem Stopp-Vorgang)

Der Normalbetrieb wird durch *byMode* = 0 ausgewählt.

- Wird *bAutoPos* auf TRUE gesetzt, läuft die Königswelle über die eingestellte Rampe *adnTiPos[2]* hoch auf Sollgeschwindigkeit (*nNln_v*).
- Die Kurve wird solange abgefahren, solange *bAutoPos* auf TRUE gesetzt bleibt.
- Wird *bAutoPos* zurück auf FALSE gesetzt, wird die Königswelle über die eingestellte Rampe *adnTiPos[2]* in die Stopp-Position gebremst (automatischer Stopp-Vorgang).
- Ein durch Handfahren unterbrochener Normalbetrieb muss durch eine FALSE-TRUE-Flanke an *bAutoPos* neu gestartet werden.

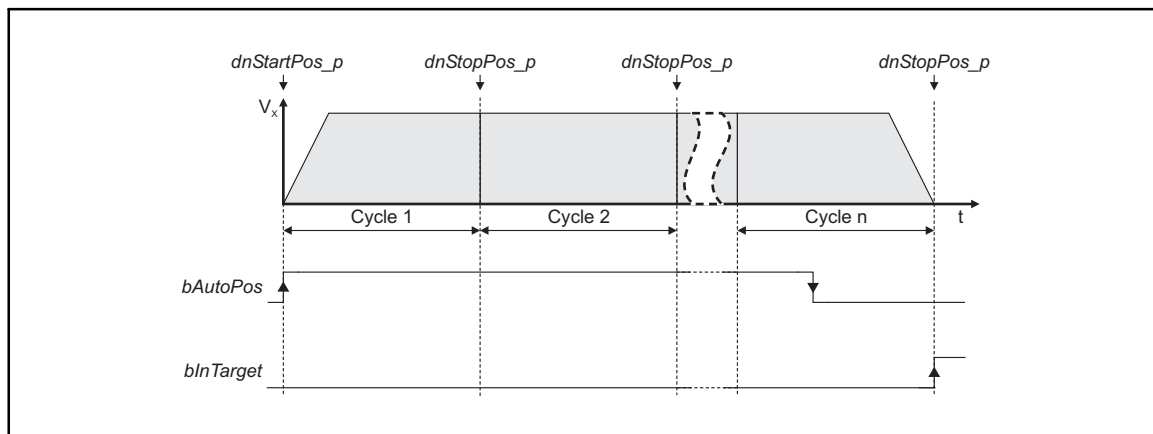
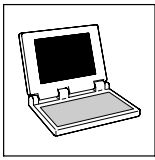


Abb. 2-46

Normalbetrieb (kontinuierliches Abfahren der Kurve)



Funktionsbibliothek LenzeCamControl.lib

Funktionsblöcke

L_CamVMasterPosition - Virtueller Master mit Positionierfunktion

Taktbetrieb (mit automatischem Stopp-Vorgang)

Der Taktbetrieb wird durch *byMode* = 1 ausgewählt.

- Erhält *bAutoPos* eine FALSE-TRUE-Flanke, läuft die Königswelle über die eingestellte Rampe *adnTiPos[2]* hoch auf Sollgeschwindigkeit (*nNln_v*) und wird automatisch in die Stopp-Position gebremst (automatischer Stopp-Vorgang).
- Die Kurve wird im Taktbetrieb maximal einmal abgefahren (aktuelle Position bis Stopp-Position).
- Sind die Beschleunigungs-/Verzögerungsrampen zu flach eingestellt, kann die Sollgeschwindigkeit nicht erreicht werden.
- Für einen weiteren Durchlauf ist eine erneute FALSE-TRUE-Flanke an *bAutoPos* erforderlich.
- Ein durch Handfahren unterbrochener Taktbetrieb muss ebenfalls durch eine erneute FALSE-TRUE-Flanke an *bAutoPos* neu gestartet werden.

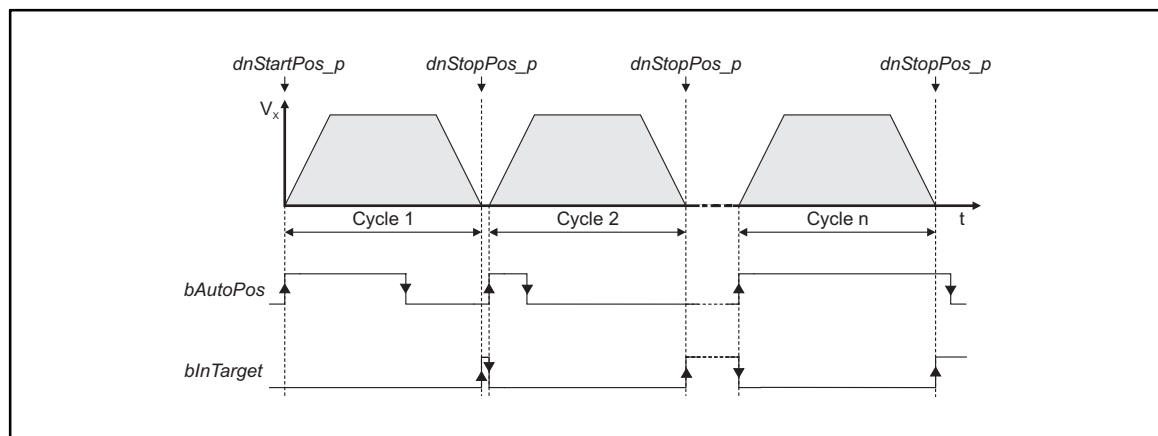


Abb. 2-47

Taktbetrieb (einmaliges Abfahren der Kurve)

Handfahrbetrieb (Tippbetrieb)

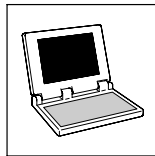
Über *bManualPos* = TRUE wird der Handfahrbetrieb aktiviert.

- Der Handfahrbetrieb hat eine höhere Priorität als der Normalbetrieb/Taktbetrieb.
- Über *adnTiPos[1..3]* können für den Handfahrbetrieb separate Beschleunigungs-/Verzögerungsrampen eingestellt werden.
- Wurde die Verfahrrichtung nicht explizit über *byMode* = 2 bzw. 3 festgelegt, wird in positiver Richtung (Cw) verfahren.

Handradfunktion über Inkrementalgeber

Die Handradfunktion ist nur möglich, wenn die Verfahrfunktion über *bEnable* = FALSE deaktiviert und *byDisableMode* = 2 eingestellt ist.

- Das an *nNln_v* vorgegebene Gebersignal wird integriert und in den Grenzen der Taktlänge am Positionsausgang *dnPosOut_p* ausgegeben.



Zwangsstopp

Durch Setzen von *bStopInstant* = TRUE kann ein sofortiger Stopp aktiviert werden.

- Der Antrieb wird unmittelbar über die eingestellte Verzögerungsrampe *adnTiPos[1]* in den Stillstand geführt, es findet somit keine Positionierung auf die Zielposition statt.



Hinweis!

Die Funktion ist nur bei freigeschalteten FB möglich (*bEnable* = TRUE) und hat Priorität vor allen anderen Verfahrfunktionen.

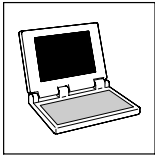
- Eine durch Zwangsstopp unterbrochene Verfahrfunktion muss durch das entsprechende Signal erneut gestartet werden.
-

Externe Masterposition

Durch Setzen von *bEnable* auf FALSE und *byDisableMode* = 4 kann eine externe Masterposition über *dnExtMaster_p* im Bereich von 0 bis zur maximalen Taktlänge (*dnActXLength_p*) vorgegeben werden.

Zuvor muss aber die Istposition *dnPosOut_p* auf diese externe Master-Position geführt werden.

- Verwenden Sie hierfür den Verfahrmodus *byMode* = 4, der durch eine FALSE-TRUE-Flanke an *bAutoPos* gestartet wird.



Funktionsbibliothek LenzeCamControl.lib

Funktionsblöcke

L_CamVMasterVelocity - Virtueller Master

2.18 L_CamVMasterVelocity - Virtueller Master

Über diesen FB können folgende Funktionen realisiert werden:

- Erzeugung einer virtuellen Leitfrequenz
- Handfahrbetrieb (Tippbetrieb)
- Stopp über einstellbare Rampe

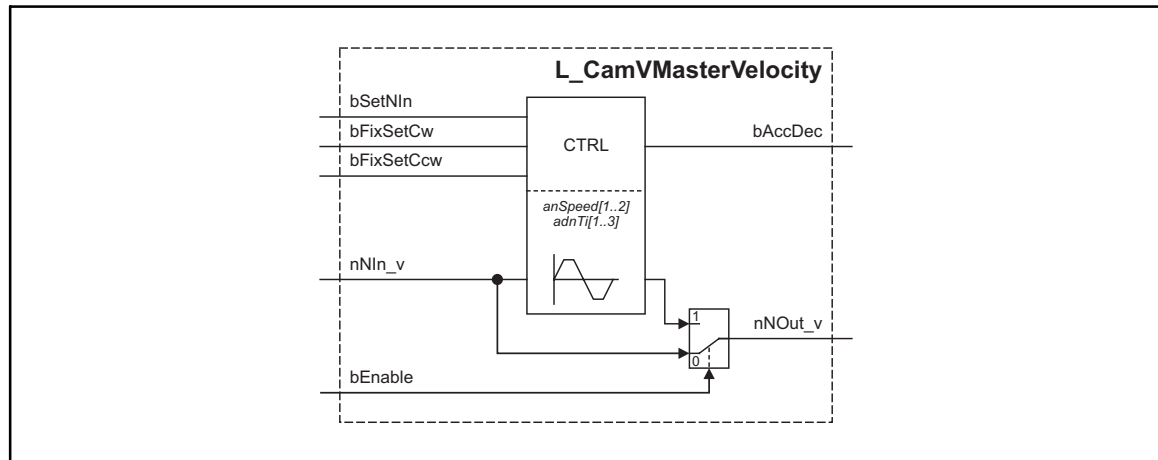
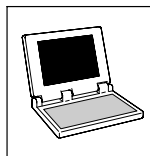


Abb. 2-48

Funktionsblock "L_CamVMasterVelocity"

⇒ Eingänge (Variablentyp: VAR_INPUT)			
Bezeichner	Datentyp	Signaltyp	Info/Einstellmöglichkeiten
bEnable	Bool	binary	Hochlaufgeber aktivieren/deaktivieren
			FALSE Hochlaufgeber ist deaktiviert
			• Der Leitfrequenzsollwert wird direkt auf den Leitfrequenz- ausgang geschaltet.
			TRUE Hochlaufgeber ist aktiviert
			• Werden die drei Eingänge <i>bSetNIn</i> , <i>bFixSetCw</i> und <i>bFix- SetCcw</i> alle auf FALSE gesetzt, wird der Antrieb über die eingestellte Verzögerungsrampe <i>adnTi[1]</i> in den Stillstand geführt (Stopp-Funktion).
bSetNIn	Bool	binary	Zielgeschwindigkeit Hochlaufgeber
			FALSE Zielgeschwindigkeit = 0 rpm
			TRUE Zielgeschwindigkeit = Vorgabe an <i>nNIn_v</i>
bFixSetCw	Bool	binary	Handfahren
			• Dieser Eingang hat Priorität vor <i>bFixSetCcw</i> .
			TRUE Handfahren in positiver Richtung (Cw)
bFixSetCcw	Bool	binary	Handfahren
			TRUE Handfahren in negativer Richtung (Ccw)
nNIn_v	Integer	velocity	Leitfrequenzsollwert



Ausgänge → (Variablentyp: VAR_OUTPUT)					
Bezeichner	Datentyp	Signaltyp	Wert/Bedeutung		
bAccDec	Bool	binary	Status Hochlaufgeber <table><tr><td>TRUE</td><td>Hochlaufgeberfunktion aktiv, d. h. es findet gerade ein Beschleunigungs- oder Verzögerungsvorgang statt.</td></tr></table>	TRUE	Hochlaufgeberfunktion aktiv, d. h. es findet gerade ein Beschleunigungs- oder Verzögerungsvorgang statt.
TRUE	Hochlaufgeberfunktion aktiv, d. h. es findet gerade ein Beschleunigungs- oder Verzögerungsvorgang statt.				
nNOut_v	Integer	velocity	Leitfrequenz		

Parameter (Variablentyp: VAR_CONSTANT_RETAIN)				
Bezeichner	Datentyp	Signaltyp	Info	
Codestelle*			Einstellmöglichkeiten	Voreinstellung
adnTi[1..3]	Array of Double Integer	-	Rampen für Beschleunigung/Verzögerung (bezogen auf 4000 rpm)	
			adnTi[1] = Verzögerungsrampe für Stopp-Funktion	
C1462/1			0.010 {0.001 s} 130.000	1.000 s
			adnTi[2] = Beschleunigungs-/Verzögerungsrampe für Normalbetrieb	
C1462/2			0.010 {0.001 s} 130.000	1.000 s
			adnTi[3] = Beschleunigungs-/Verzögerungsrampe für Handfahren	
C1462/3			0.010 {0.001 s} 130.000	1.000 s
anSpeed[1..2]	Array of Integer	-	Geschwindigkeit für Handfahren	
			anSpeed[1] = Geschwindigkeit für Handfahren in positiver Richtung (Cw)	
C1461/1			0 {1 rpm} 15000	200 rpm
			anSpeed[2] = Geschwindigkeit für Handfahren in negativer Richtung (Ccw)	
C1461/2			0 {1 rpm} 15000	200 rpm
* Codestelle der Erstinstanz				

* Codestelle der Erstinstanz

Erzeugung der virtuellen Leitfrequenz

Die Kurvenscheibe benötigt für den Betrieb ein Leitsystem (Geschwindigkeit oder Position der X-Achse). Läuft die Kurvenscheibe nicht im Verbund mit anderen Antrieben oder ist kein Leitgeber vorhanden, kann mit Hilfe dieses FBs eine virtuelle Leitfrequenz erzeugt werden.

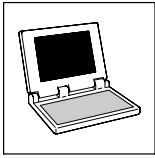
- Der Geschwindigkeitssollwert wird über *nNIn_v* in inc/ms vorgegeben.

Hochlaufgeber freischalten

Ist *bEnable* auf TRUE gesetzt, ist der Hochlaufgeber freigeschaltet. Bei FALSE wird der Geschwindigkeitssollwert direkt zum Leitfrequenzausgang durchgeschliffen.

Normalbetrieb

Wird *bSetNIn* auf TRUE gesetzt, fährt der (freigeschaltete) Hochlaufgeber über die eingestellte Rampe *adnTi[1]* hoch auf die über *nNIn_v* vorgegebene Sollgeschwindigkeit.



Funktionsbibliothek *LenzeCamControl.lib*

Funktionsblöcke

L_CamVMasterVelocity - Virtueller Master

Handfahrbetrieb (Tippbetrieb)

Über *bFixSetCw* und *bFixSetCcw* wird die Handfahrfunktion gemäß folgender Tabelle gesteuert:

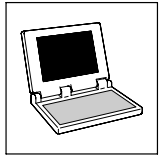
<i>bFixSetCw</i>	<i>bFixSetCcw</i>	
0	0	Handfahren deaktiviert • <i>bSetNIn</i> = FALSE: Leitfrequenz <i>nNout_v</i> = 0 rpm • <i>bSetNIn</i> = TRUE: Leitfrequenz <i>nNout_v</i> = <i>nNIn_v</i>
0	1	Handfahren in negativer Richtung (Ccw) mit Geschwindigkeit <i>anSpeed</i> [2].
1	0 oder 1	Handfahren in positiver Richtung (Cw) mit Geschwindigkeit <i>anSpeed</i> [1].

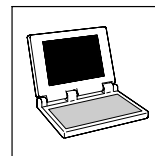
- Der Handfahrbetrieb hat eine höhere Priorität als der Normalbetrieb.
- Über *anSpeed*[1..2] können für den Handfahrbetrieb separate Beschleunigungs-/Verzögerungsrampen eingestellt werden.

Stopp-Funktion

Werden die drei Eingänge *bSetNIn*, *bFixSetCw* und *bFixSetCcw* bei freigeschalteten Hochlaufgeber (*bEnable* = TRUE) alle auf FALSE gesetzt, wird der Antrieb über die eingestellte Verzögerungsrampe *adnTi*[1] in den Stillstand geführt.

- Die Stopp-Funktion hat die höchste Priorität.



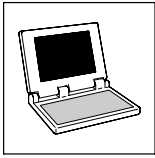


3 Funktionsblöcke *LenzeCamControl2V02xx.lib*

Die Funktionsblöcke für die Technologiefunktion "Kurvenscheibe" sind auf die zwei Funktionsbibliotheken **LenzeCamControl1V02xx.lib** und **LenzeCamControl2V02xx.lib** aufgeteilt.



In diesem Kapitel erhalten Sie Informationen zu den Funktionsblöcken, die in der Funktionsbibliothek **LenzeCamControl2V02xx.lib** enthalten sind.



Funktionsbibliothek LenzeCamControl.lib

Funktionsblöcke

L_CamAddVelocity - Drehzahl-Addition

3.1 L_CamAddVelocity - Drehzahl-Addition

Über diesen FB können Drehzahlssignale addiert bzw. subtrahiert werden.

Der Ausgang ist auf ± 16383 inc/ms begrenzt. Durch den internen "Nachlaufspeicher" tritt kein Positionsverlust auf, wenn die Begrenzung anspricht.

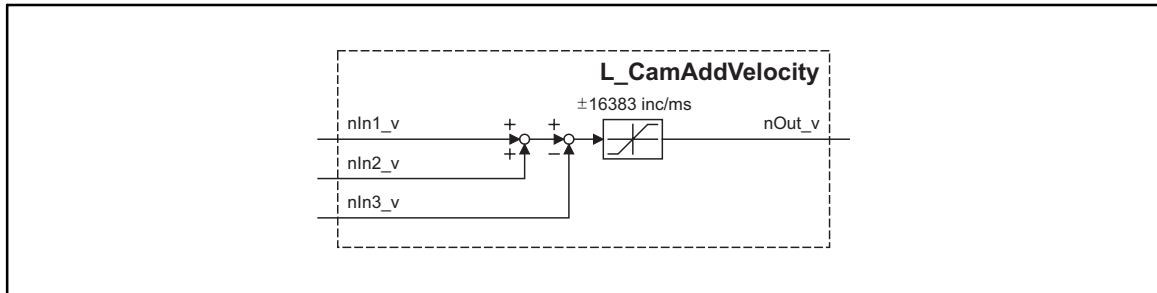
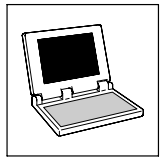


Abb. 3-1

Funktionsblock "L_CamAddVelocity"

⇒ Eingänge (Variablentyp: VAR_INPUT)			
Bezeichner	Datentyp	Signaltyp	Info/Einstellmöglichkeiten
nIn1_v	Integer	velocity	Additionseingang
nIn2_v	Integer	velocity	Additionseingang
nIn3_v	Integer	velocity	Subtraktionseingang

Ausgänge ⇒ (Variablentyp: VAR_OUTPUT)			
Bezeichner	Datentyp	Signaltyp	Wert/Bedeutung
nOut_v	Integer	velocity	Ausgang - Der Ausgang ist auf ± 16383 inc/ms begrenzt (ohne Positionsverlust). ± 16383 inc/ms $\equiv$ 14999 rpm



3.2 L_CamAdjustDfIn - Leitfrequenzanpassung

Mit diesem FB können Sie eine Anpassung des verwendeten Inkrementalgebers vornehmen.

- Der FB erhält sein Eingangssignal direkt vom Systembaustein **DFIN**.
- Die Anpassung erfolgt über die Geberkonstante *wEncConst* und dem Reckfaktor *x/y*.

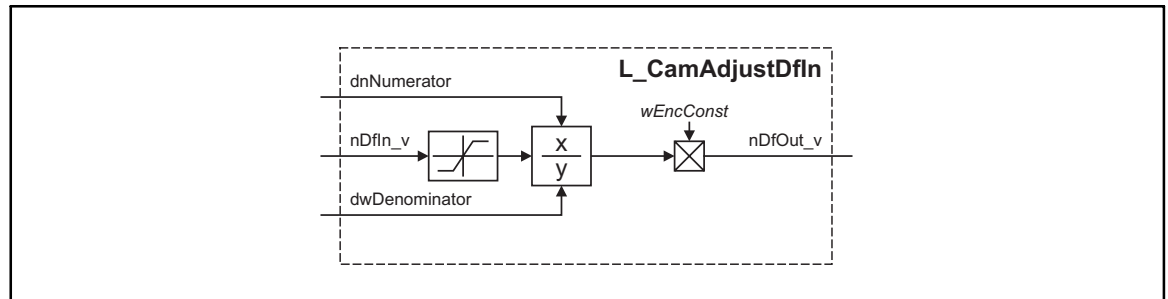


Abb. 3-2 Funktionsblock "L_Ca mAd justDfIn"

⇒ Eingänge (Variablentyp: VAR_INPUT)			
Bezeichner	Datentyp	Signaltyp	Info/Einstellmöglichkeiten
nDfIn_v	Integer	velocity	Eingang für das Geschwindigkeitssignal vom SB DFIN in [inc/ms] • Begrenzung: 2147 [inc/ms]
dnNumerator	Double Integer	position	Reckfaktor (Zähler) • Begrenzung: -100000 ... +100000
dwDenominator	Double Word	position	Reckfaktor (Nenner) • Begrenzung: 1 ... 2 ³¹

Ausgänge ⇒ (Variablentyp: VAR_OUTPUT)			
Bezeichner	Datentyp	Signaltyp	Wert/Bedeutung
nDfOut_v	Integer	velocity	Ausgabe der angepassten Geschwindigkeit in [inc/ms] • 16384 [inc/ms] ≙ 15000 rpm

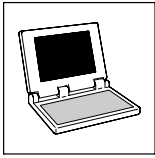
Parameter (Variablentyp: VAR_CONSTANT_RETAIN)				
Bezeichner	Datentyp	Signaltyp	Info	
Codestelle*			Einstellmöglichkeiten	Voreinstellung
wEncConst	Word	number	Geberkonstante des Encoders an nDfIn_v	
C1480			10 {1} 32767	16384

* Codestelle der Erstinstanz



Hinweis!

Verwenden Sie diesen FB nicht, wenn für den Leitwert eine Markensynchronisierung (Touch-Probe) durchgeführt wird. (Siehe FB **L_CamSetXAxisVelocity** (2-59))



Funktionsbibliothek LenzCamControl.lib

Funktionsblöcke

L_CamDifferentiate - Konverter/Differenzierer

3.3 L_CamDifferentiate - Konverter/Differenzierer

Mit diesem FB können Sie eine Winkeländerung am Eingang *dnIn_p* in eine Drehzahl umsetzen.

Der FB besitzt einen Nachlaufspeicher, damit es nicht zum Positions- oder Inkrementeverlust kommt, falls der Drehzahlausgang durch eine zu große Positionsänderung am Eingang begrenzt wird.

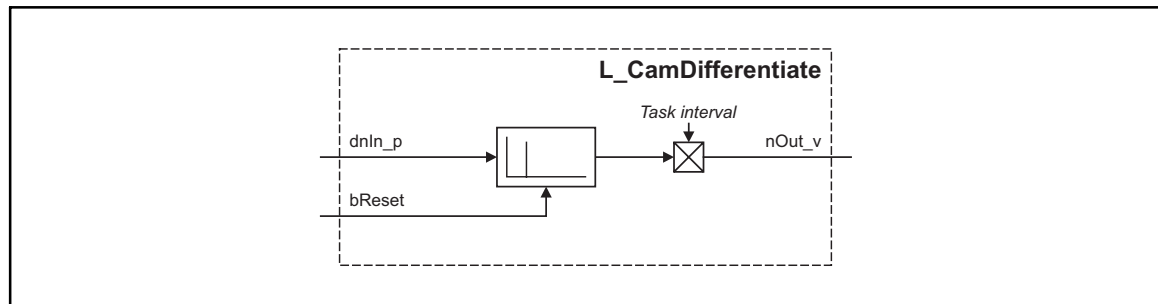


Abb. 3-3 Funktionsblock "L_CamDifferentiate"

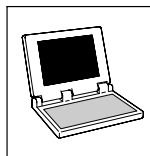
⇒ Eingänge (Variablentyp: VAR_INPUT)				
Bezeichner	Datentyp	Signaltyp	Info/Einstellmöglichkeiten	
dnIn_p	Double Integer	position	Winkelsignal, das in eine Drehzahl umgerechnet werden soll.	
bReset	Bool	binary	Ausgang zurücksetzen	
			FALSE	Am Ausgang liegt das in eine Drehzahl konvertierte Winkelsignal an.
			TRUE	Der Ausgang ist zurückgesetzt, als Drehzahl werden 0 rpm ausgegeben.

Ausgänge ⇒ (Variablentyp: VAR_OUTPUT)			
Bezeichner	Datentyp	Signaltyp	Wert/Bedeutung
nOut_v	Integer	velocity	Drehzahl 16384 ≙ 15000 rpm - max. Ausgangswert $\pm 32767 \equiv \pm 30000$ rpm



Tipp!

Benutzen Sie zur Differenzierung des Leitwertes (X-Achse) in Kurvenscheibenanwendungen den FB **L_CamExtrapolate** (mit der Einstellung *bEnable* = FALSE).



3.4 L_CamEditProfileData - Online-Änderung der Profilpunkte

Mit diesem FB können die Stütz- oder Beschleunigungspunkte eines erstellten Bewegungsprofils geändert werden. Des Weiteren kann eine Kontrolle der geänderten Stützpunkte in X-Richtung durchgeführt werden.



Hinweis!

Mit diesem FB kann kein Bewegungsprofil (*.lc9) generiert werden. Es lassen sich nur vorhandene Bewegungsprofile verändern, die mit dem **CamDesigner** im absoluten Datenmodell erstellt wurden.

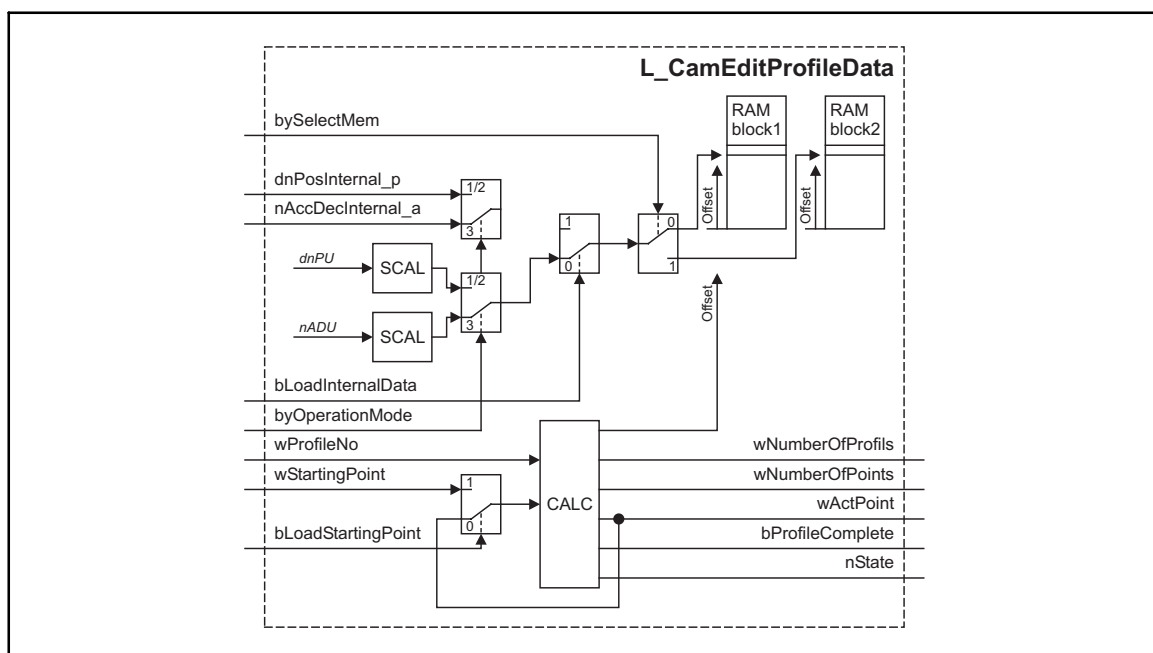
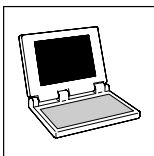


Abb. 3-4 Funktionsblock "L_CamEditProfileData"

⇒ Eingänge (Variablentyp: VAR_INPUT)				
Bezeichner	Datentyp	Signaltyp	Info/Einstellmöglichkeiten	
bySelectMem	Byte	number	Auswahl der *.lc9-Datei im Speicher	
			1	RAM-Block 1 (Vordergrunddatenfeld, Standard)
			2	RAM-Block 2 (Hintergrunddatenfeld)
dnPosInternal_p	Double Integer	position	Positionswert in [inc] - Geben Sie entsprechend der gewählten Betriebsart dieser Eingangsvariablen den neuen X- bzw. Y-Positionswert für den aktuellen Stützpunkt vor. - X-Positionswerte dürfen nur im positiven Bereich liegen. - Der Positionswert wird durch eine TRUE-FALSE-Flanke an <i>bLoadInternalData</i> für den aktuellen Stützpunkt <i>wActPoint</i> übernommen.	
nAccDecInternal_a	Integer	analog	Vorgabe der Beschleunigungswerte als analoger Wert ±16384 ≡ ±100 %	
bLoadInternalData	Bool	binary	Übernahme des Positionswertes an <i>dnPosInternal_p</i> bzw. des Beschleunigungswertes an <i>nAccDecInternal_a</i> für den aktuellen Stützpunkt.	
			TRUE ⇨ FALSEÜbernahme des Wertes an <i>dnPosInternal_p</i> bzw. <i>nAccDecInternal_a</i> und automatischer Übergang zum nächsten Stützpunkt (Auto-Inkrement-Funktion).	



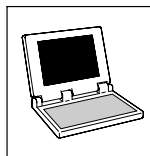
Funktionsbibliothek LenzCamControl.lib

Funktionsblöcke

L_CamEditProfileData - Online-Änderung der Profilpunkte

⇒ Eingänge (Variablentyp: VAR_INPUT)				
Bezeichner	Datentyp	Signaltyp	Info/Einstellmöglichkeiten	
byOperationMode	Byte	number	Auswahl der Betriebsart	
			0	Reset (<i>nState</i> , <i>wActPoint</i>)
			1	Stützpunkte der X-Achse ändern
			2	Stützpunkte der Y-Achse ändern
			3	Beschleunigungswerte ändern
			4	Kontrolle durchführen, ob die Stützpunkte auf der X-Achse aufsteigend vorliegen (Sägezahn).
wProfileNo	Word	number	Auswahl des Bewegungsprofils, dessen Stützpunkte geändert werden sollen. Profilnummer 0 ... <i>wNumberOfProfiles</i>	
wStartingPoint	Word	number	Auswahl der Nummer des Stützpunktes, deren X-/Y-Positionswerte oder Beschleunigungswerte (siehe <i>byOperationMode</i>) geändert werden sollen. Stützpunktnummer 0 ... <i>wNumberOfPoints</i> eingeben	
bLoadStartingPoint	Bool	binary	Zur gewählten Stützpunktnummer <i>wStartingPoint</i> springen, z. B. zum erneuten Ändern der Werte.	
			TRUE ↘ FALSE	Aktueller Stützpunkt <i>wActPoint</i> = <i>wStartingPoint</i>

Ausgänge ⇨ (Variablentyp: VAR_OUTPUT)																	
Bezeichner	Datentyp	Signaltyp	Wert/Bedeutung														
wNumberOfProfiles	Word	number	Anzahl der Profile in der *.lc9-Datei. Es wird die Kurve mit der höchsten Nummer ausgegeben. (z. B. <i>wNumberOfProfiles</i> = "2": Die Kurven 0,1 und 2 sind enthalten.)														
wNumberOfPoints	Word	number	Anzahl der Kurvenstützpunkte im aktuell gewählten Profil. Die Variable <i>wNumberOfPoints</i> führt die Anzahl der Kurvenstützpunkte - 1. (z. B. <i>wNumberOfPoints</i> = "99": die Kurve hat 100 Stützpunkte (0 ... 99))														
wActPoint	Word	number	Aktuelle Stützpunktnummer Durch eine TRUE-FALSE-Flanke an <i>bLoadInternalData</i> oder Beschreiben der Parameter <i>dnPU</i> (C1491) bzw. <i>nADU</i> (C1492) werden die im entsprechenden Modus (<i>byOperationMode</i>) geänderten Werte dieses Stützpunktes übernommen.														
bProfileComplete	Bool	binary	Meldesignal "Letzter Stützpunkt des Profils erreicht" <table><tr><td>TRUE</td><td>Letzter Stützpunkt des Profils wurde erreicht und verändert.</td></tr></table>	TRUE	Letzter Stützpunkt des Profils wurde erreicht und verändert.												
TRUE	Letzter Stützpunkt des Profils wurde erreicht und verändert.																
nState	Integer	number	Status <table><tr><td>20</td><td>Vorgabe für den ersten zu ändernden Stützpunkt (<i>wStartingPoint</i>) ist größer als die Anzahl der im Profil vorhandenen Stützpunkte.</td></tr><tr><td>15</td><td>Initialisierung aktiv, Daten werden aus *.lc9-Datei gelesen.</td></tr><tr><td>10</td><td>Vorgabe für das zu ändernde Bewegungsprofil (<i>wProfileNo</i>) größer als die Anzahl der in der *.lc9-Datei vorhandenen Bewegungsprofile.</td></tr><tr><td>0</td><td>FB ist betriebsbereit.</td></tr><tr><td>-2</td><td>Keine Profildaten in der *.lc9-Datei vorhanden.</td></tr><tr><td>-4</td><td>Profildaten liegen nicht im absoluten Datenmodell vor.</td></tr><tr><td>-10</td><td>Stützpunkte auf der X-Achse sind nicht aufsteigend.</td></tr></table>	20	Vorgabe für den ersten zu ändernden Stützpunkt (<i>wStartingPoint</i>) ist größer als die Anzahl der im Profil vorhandenen Stützpunkte.	15	Initialisierung aktiv, Daten werden aus *.lc9-Datei gelesen.	10	Vorgabe für das zu ändernde Bewegungsprofil (<i>wProfileNo</i>) größer als die Anzahl der in der *.lc9-Datei vorhandenen Bewegungsprofile.	0	FB ist betriebsbereit.	-2	Keine Profildaten in der *.lc9-Datei vorhanden.	-4	Profildaten liegen nicht im absoluten Datenmodell vor.	-10	Stützpunkte auf der X-Achse sind nicht aufsteigend.
20	Vorgabe für den ersten zu ändernden Stützpunkt (<i>wStartingPoint</i>) ist größer als die Anzahl der im Profil vorhandenen Stützpunkte.																
15	Initialisierung aktiv, Daten werden aus *.lc9-Datei gelesen.																
10	Vorgabe für das zu ändernde Bewegungsprofil (<i>wProfileNo</i>) größer als die Anzahl der in der *.lc9-Datei vorhandenen Bewegungsprofile.																
0	FB ist betriebsbereit.																
-2	Keine Profildaten in der *.lc9-Datei vorhanden.																
-4	Profildaten liegen nicht im absoluten Datenmodell vor.																
-10	Stützpunkte auf der X-Achse sind nicht aufsteigend.																



Stop!

Wenn Sie die Stützpunkte einer Kurve über den **FB L_CamEditProfileData** verändern möchten, stellen Sie sicher, dass das zu bearbeitende Bewegungsprofil nicht in einer Instanz des FBs **L_CamData** aufgerufen wird. Andernfalls kann das geänderte Bewegungsprofil Unstetigkeiten in der Bewegung aufweisen. Beachten Sie auch, dass die x- oder y-Restwerte eines Profiles nicht mit verändert oder neu erstellt werden können.

Falls aus prozesstechnischen Gründen das zu ändernde Bewegungsprofil im **FB L_CamData** angewählt bleibt, achten Sie darauf, dass der x- und y-Endwert des Bewegungsprofils identisch zu den bisherigen Werten bleibt (nur Änderung von Zwischenwerten erlaubt).

Parameter (Variablentyp: VAR_CONSTANT_RETAIN)				
Bezeichner	Datentyp	Signaltyp	Info	
Codestelle*			Einstellmöglichkeiten	Voreinstellung
dnPU C1491	Double Integer	position [unit]	Positionsvorgabe in [unit], die durch die Skalieroutine umgerechnet und für den aktuellen Stützpunkt <i>wActPoint</i> übernommen wird. • Entsprechend der gewählten Betriebsart werden hier X- oder Y-Positionswerte vorgegeben. Als X-Koordinaten dürfen nur positive Werte vorgegeben werden.	
			-214000.0000 {0.001 [unit]} 214000.0000	0.0000 [unit]
nADU C1492	Integer	analog [%]	Beschleunigungsvorgabe in [%], die durch die Skalieroutine umgerechnet und für den aktuellen Stützpunkt <i>wActPoint</i> übernommen wird. • $\pm 100\%$ ± 16384	
			-100.00 {0.01 %}	100.00 0.00 %

* Codestelle der Erstinstanz

Auswahl von Achse, Bewegungsprofil und Stützpunktnummer um Werte zu ändern

Mit dem **FB L_CamEditProfileData** können entweder die X- oder die Y-Koordinaten eines Bewegungsprofils verändert werden. Sollen die X- sowie die Y-Koordinaten parallel verändert werden, muss der FB hierzu instanziiert werden. Des Weiteren können mit diesem FB die Beschleunigungswerte eines Bewegungsprofils verändert werden.

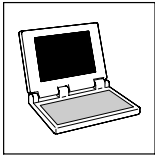
- Über *byOperationMode* erfolgt die Auswahl, welche Art von Werten geändert werden soll:
 - 1 = Stützpunkte auf der X-Achse
 - 2 = Stützpunkte auf der Y-Achse
 - 3 = Beschleunigungswerte
- Über *wProfileNo* wählen Sie das zu verändernde Bewegungsprofil aus.
 - Am Ausgang *wNumberOfPoints* wird angezeigt, wieviel Stützpunkte im ausgewählten Profil vorhanden sind.
- Über *wStartingPoint* wählen Sie die Nummer des Stützpunktes aus, dessen Werte geändert werden sollen.
 - Stützpunktnummer 0 ... *wNumberOfPoints* eingeben.
 - Die Übernahme erfolgt durch eine TRUE-FALSE-Flanke an *bLoadStartingPoint* (*wActPoint* = *wStartingPoint*).

Soll eine Stützpunktnummer angesprungen werden, die außerhalb von 0 ... *wNumberOfPoints* liegt, erfolgt eine Fehlermeldung (*nState* = 20) und die aktuelle Nummer des Stützpunktes wird auf "0" gesetzt (*wActPoint* = 0).

Vorgabe eines neuen Positionswertes

Ein neuer Positionswert kann vorgegeben werden als:

- physikalische Größe [unit] über den Parameter *dnPU* (C1491)
- inkrementelle Größe [inc] an *dnPosInternal_p*



Funktionsbibliothek LenzeCamControl.lib

Funktionsblöcke

L_CamEditProfileData - Online-Änderung der Profilpunkte

Vorgabe eines neuen Beschleunigungswertes

Ein neuer Beschleunigungswert kann vorgegeben werden als:

- physikalische Größe [%] über den Parameter *nADU* (C1492)
- inkrementelle Größe an *nAccDecInternal_a*

Übernahme der neuen Werte für den aktuellen Stützpunkt

- Erfolgt die Vorgabe über die Parameter *dnPU* bzw. *nADU*, wird der Wert über die entsprechende Skaliererroutine umgerechnet und automatisch für den aktuellen Stützpunkt *dnActPoint* übernommen.
- Erfolgt die Vorgabe stattdessen über die Eingangsvariablen *dnPosInternal_p* bzw. *nAccDecInternal_a*, ist für die Übernahme des entsprechenden Wertes eine TRUE-FALSE-Flanke an *bLoadInternalData* für jeden Stützpunkt erforderlich.
- Nach der Übernahme des Wertes für den aktuellen Stützpunkt wird automatisch der nächste Stützpunkt des Bewegungsprofils zum aktuellen Stützpunkt ("Automatisches Inkrementieren").

Kontrolle der geänderten Stützpunkte

Für die X-Achse ist eine Kontrolle der eingegebenen Daten möglich. Hierbei handelt es sich um einen automatischen Ablauf, der die einzelnen Stützpunkte der X-Achse durchläuft und kontrolliert, ob die Punkte in aufsteigender Reihenfolge vorliegen.

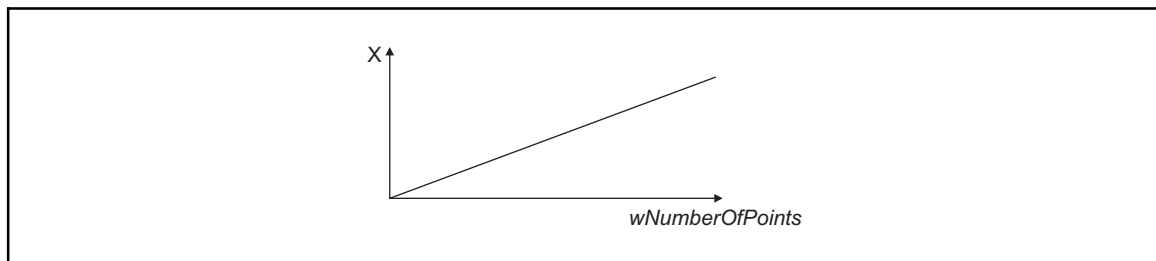
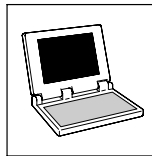


Abb. 3-5

Kontrolle der geänderten Stützpunkte

Setzen Sie *byOperationMode* auf den Wert "4", um die Kontrolle zu aktivieren.

- Liegen alle Stützpunkte der X-Achse in aufsteigender Reihenfolge vor, wird der Ausgang *nState* auf den Wert "0" und der Ausgang *bProfileComplete* auf TRUE gesetzt.
- Wird während der Kontrolle ein X-Wert gefunden, der nicht größer als der vorhergehende X-Wert ist, wird die Kontrolle an dieser Stelle mit einer Fehlermeldung (*nState* = "-10") beendet.
 - Setzen Sie *byOperationMode* auf den Wert "0", um die Fehlermeldung zurückzusetzen.
- Für eine erneute Kontrolle setzen Sie vor dem eigentlichen Start *byOperationMode* auf den Wert "0", um die Kontrollfunktion neu zu initialisieren.
- Die Dauer der Überprüfung richtet sich nach der Anzahl der Stützpunkte und nach dem Taskintervall, in dem der FB abgearbeitet wird. Pro FB-Aufruf werden 10 Stützpunkte durchgerechnet.



Beispiel: Änderung der Positionen Y-seitig über die Variable *dnPosInternal_p*

1. FB einschalten.
2. RAM-Baustein auswählen über den Eingang *bySelectMem* = 1 oder 2.
3. Modus einstellen: *byOpMode* = 2
4. Zu veränderndes Profil einstellen: z. B. *wProfileNo* = 2
 - Sind n Profile anwählbar, kann ein Profil über Zahlenwerte (*wProfileNo* = 0 ... n-1) angewählt werden.
5. Nummer des zu ändernden Stützpunktes auswählen (optional):
 - *wStartingPoint* = Stützpunktnummer 0 ... *wNumberOfPoints*
 - *bLoadStartPoint* = TRUE => FALSE
 - (Sprung zum gewählten Stützpunkt: *wActPoint* = *wStartingPoint*)
6. Den Wert der unter 5. eingestellten Stützpunktnummer verändern, durch Beschreiben des Parameters *dnPU* (C1491).
7. Der nächstfolgende Stützpunkt wird angewählt (der Ausgang *wActPoint* wird automatisch inkrementiert).

Der FB ist nun bereit zum Beschreiben des nächsten Stützpunktes. Wurde der letzte Stützpunkt erreicht und beschrieben, wird der Ausgang *bProfileComplete* = TRUE gesetzt. Dieser Ausgang wird auf FALSE zurückgesetzt, wenn ein anderer Stützpunkt angesprungen wird (siehe 5.) oder wenn der Modus *byOpMode* auf den Wert "0" gesetzt wird (siehe 3.).

Bei Änderungen über den Parameter *nADU* halten Sie dieselben Schritte ein. Setzen Sie hier aber *byOpMode* = 3, da die Werte sonst nicht angenommen werden.

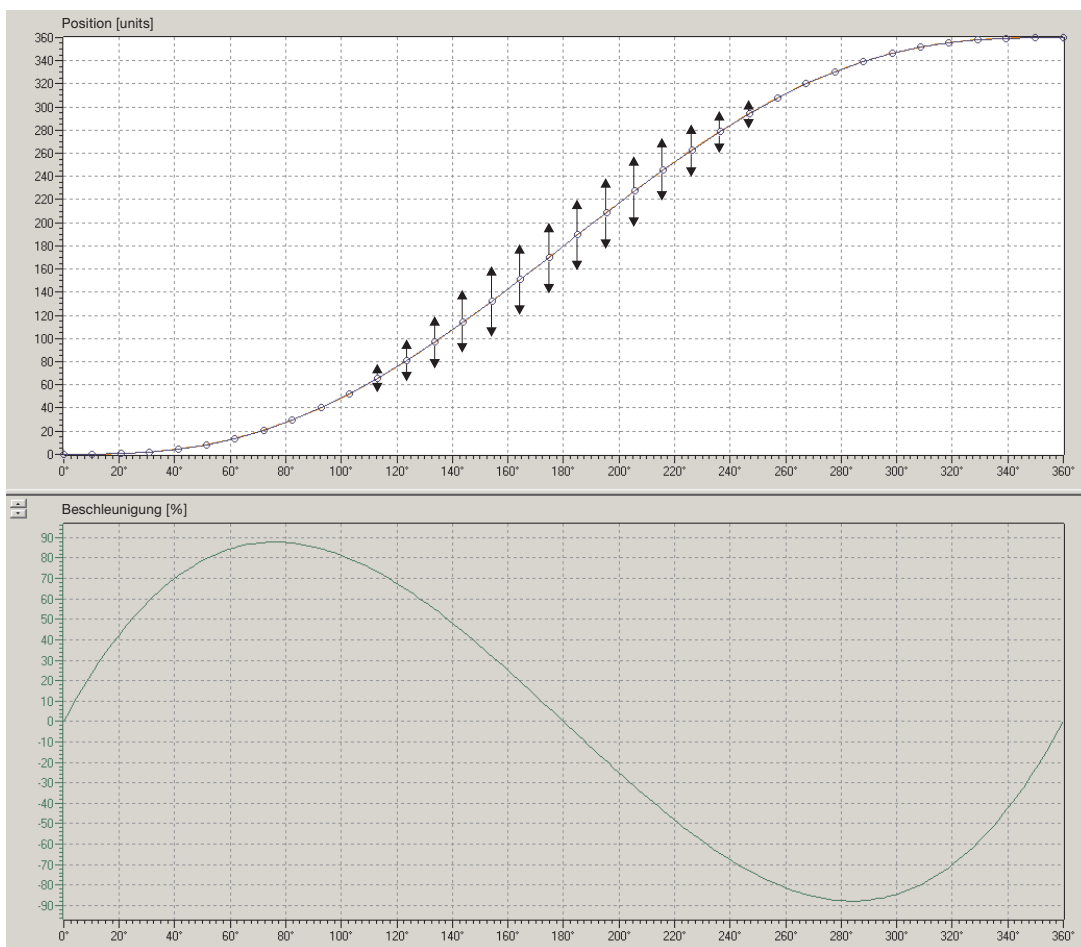
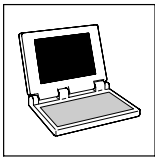


Abb. 3-6

Beispiel: Änderung der Positionen Y-seitig



Funktionsbibliothek *LenzeCamControl.lib*

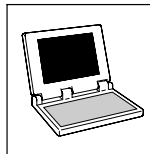
Funktionsblöcke

L_CamEditProfileData - Online-Änderung der Profilpunkte

Beispiel: Änderung der Positionen X-seitig über den Parameter *dnPU*

1. FB einschalten.
2. RAM-Baustein auswählen über den Eingang *bySelectMem* = 1 oder 2.
3. Modus einstellen: *byOpMode* = 1
4. Zu veränderndes Profil einstellen: z. B. *wProfileNo* = 2
 - Sind n Profile anwählbar, kann ein Profil über Zahlenwerte (*wProfileNo* = 0 ... n-1) angewählt werden.
5. Nummer des zu ändernden Stützpunktes auswählen (optional):
 - *wStartingPoint* = Stützpunktnummer 0 ... *wNumberOfPoints*
 - *bLoadStartPoint* = TRUE => FALSE
(Sprung zum gewählten Stützpunkt: *wActPoint* = *wStartingPoint*)
6. Den Wert der unter 5. eingestellten Stützpunktnummer verändern, durch Beschreiben der Variablen *dnPosInternal_p*.
7. *bLoadInternalData* = TRUE => FALSE setzen.
 - Der Ausgang *wActPoint* zeigt den nächsten Stützpunkt an.

Der FB ist nun bereit zum Beschreiben des nächsten Stützpunktes. Wurde der letzte Stützpunkt erreicht und beschrieben, wird der Ausgang *bProfileComplete* = TRUE gesetzt. Dieser Ausgang wird auf FALSE zurückgesetzt, wenn ein anderer Stützpunkt angesprungen wird (siehe 5.) oder wenn der Modus *byOpMode* auf den Wert "0" gesetzt wird (siehe 3.).



3.5 L_CamIntegrateVarLim - Winkelintegrator

Dieser FB stellt einen Winkelintegrator mit "online" veränderbaren Grenzen zur Verfügung.

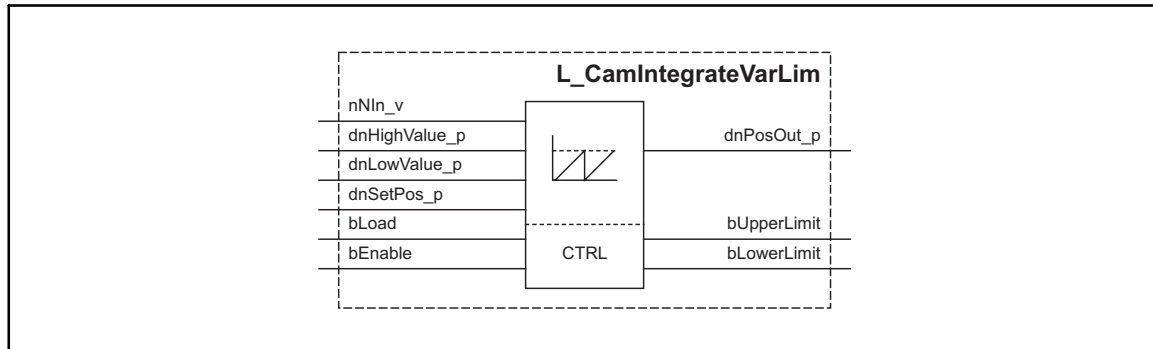
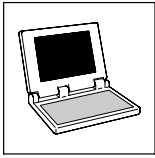


Abb. 3-7

Funktionsblock "L_CamIntegrateVarLim"

⇒ Eingänge (Variablentyp: VAR_INPUT)			
Bezeichner	Datentyp	Signaltyp	Info/Einstellmöglichkeiten
nNIn_v	Integer	velocity	Geschwindigkeit/Leitfrequenz in [inc/ms]
dnHighValue_p	Double Integer	position	Oberer Wert für den Rücksprung nach <i>dnLowValue_p</i>
dnLowValue_p	Double Integer	position	Unterer Wert für den Rücksprung nach <i>dnHighValue_p</i>
dnSetPos_p	Double Integer	position	Startwert für den Integrator • Wird mit Low-High-Flanke an <i>bLoad</i> in den Integrator geladen.
bLoad	Bool	binary	Startwert in Integrator laden FALSE ↗ TRUE Startwert <i>dnSetPos_p</i> in Integrator laden.
bEnable	Bool	binary	Startwert permanent in Integrator laden FALSE Startwert <i>dnSetPos_p</i> wird permanent in den Integrator geladen (Reset-Funktion). TRUE Integrator aktiv.

Ausgänge ⇒ (Variablentyp: VAR_OUTPUT)			
Bezeichner	Datentyp	Signaltyp	Wert/Bedeutung
dnPosOut_p	Double Integer	position	Ausgabe der Position • 65535 ≙ eine Motor-/Geberumdrehung
bUpperLimit	Bool	binary	Meldesignal "Oberer Wert erreicht" FALSE ↗ TRUE Integratorwert hat <i>dnHighValue_p</i> in positiver Zählrichtung erreicht; der Zählvorgang wird ab <i>dnLowValue_p</i> fortgesetzt. TRUE Rücksprung ist nicht möglich, da <i>dnHighValue_p</i> zu klein für die Geschwindigkeit an <i>nNIn_v</i> ist.
bLowerLimit	Bool	binary	Meldesignal "Unterer Wert erreicht" FALSE ↗ TRUE Integratorwert hat <i>dnLowValue_p</i> in negativer Zählrichtung erreicht; der Zählvorgang wird ab <i>dnHighValue_p</i> fortgesetzt. TRUE Rücksprung ist nicht möglich, da <i>dnLowValue_p</i> zu klein für die Geschwindigkeit an <i>nNIn_v</i> ist.



Funktionsbibliothek LenzeCamControl.lib

Funktionsblöcke

L_CamIntegrateVarLim - Winkelintegrator

Drehzahl zu einem Winkel integrieren

- Mit pos. Eingangswert an $nNIn_v$ vergrößert sich der Ausgangswert an $dnPosOut_p$.
- Mit neg. Eingangswert an $nNIn_v$ verringert sich der Ausgangswert an $dnPosOut_p$.

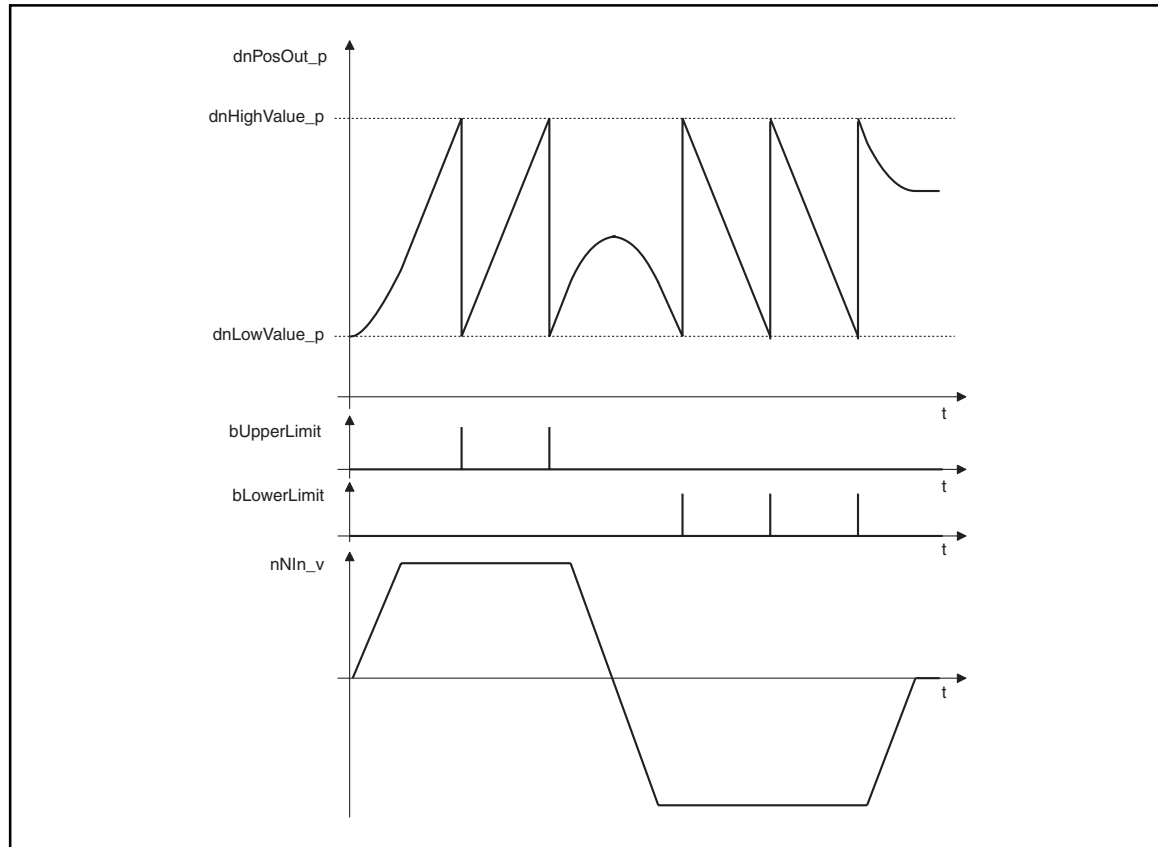


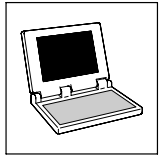
Abb. 3-8

Signalverlauf: Drehzahl zu einem Winkel integrieren



Tipp!

- Die eingestellten Werte für den Rücksprung ($dnHighValue_p$ und $dnLowValue_p$) sind keine "harten" Grenzen, d. h. der Integrator kann auch andere Werte annehmen (z. B. über $dnSetPos_p$).
- Sind $dnHighValue_p$ und $dnLowValue_p$ auf den Wert "0" eingestellt, läuft der Integrator frei.
- Alle Positionen sind ladbar. Der FB verhält sich wie in den Zeichnungen auf der folgenden Seite beschrieben.



Positionen können wie folgt geladen werden:

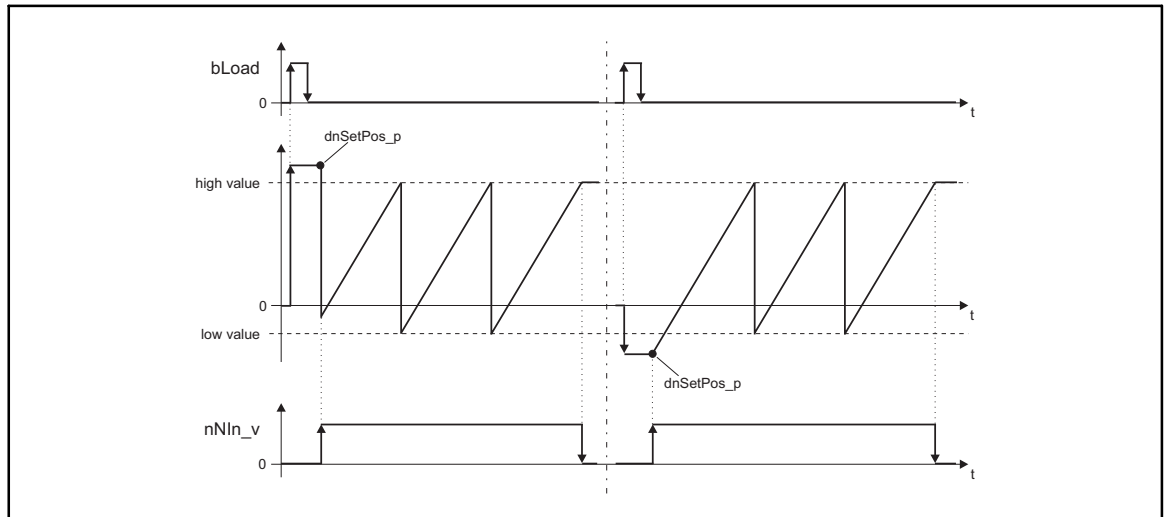


Abb. 3-9

Signalverlauf: Laden einer Position und Verfahren mit positiver Geschwindigkeit

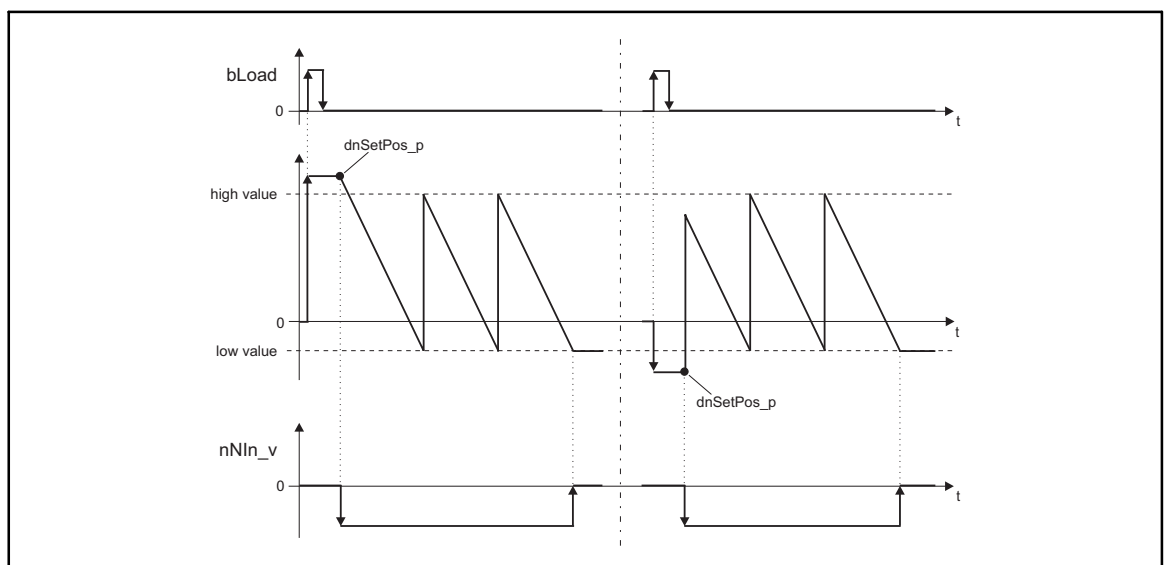
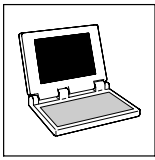


Abb. 3-10

Signalverlauf: Laden einer Position und Verfahren mit negativer Geschwindigkeit



Funktionsbibliothek LenzeCamControl.lib

Funktionsblöcke

L_CamSectionShift - Abschnittsverschiebung

3.6 L_CamSectionShift - Abschnittsverschiebung

Mit diesem FB können Sie einen Profilabschnitt durch Verzögerung der Leitachse verschieben.

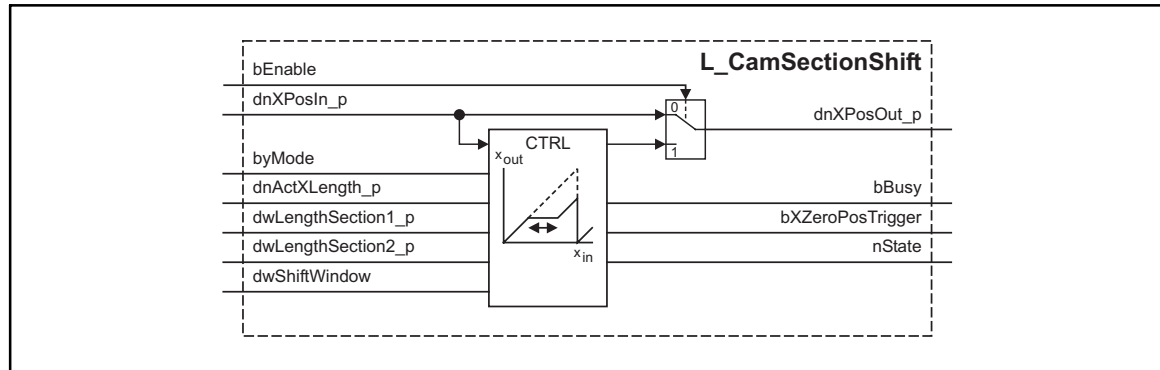
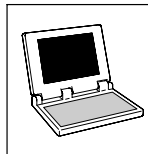


Abb. 3-11

Funktionsblock "L_CamSectionShift"

⇒ Eingänge (Variablentyp: VAR_INPUT)				
Bezeichner	Datentyp	Signaltyp	Info/Einstellmöglichkeiten	
bEnable	Bool	binary	Funktionsblock freischalten	
			FALSE	Funktionsblock gesperrt, X-Achse wird 1:1 durchgeschliffen
			TRUE	Funktionsblock freigeschaltet
dnXPosIn_p	Double Integer	position	Position der X-Achse Begrenzung: 0 ... <i>dnActXLength_p</i>	
byMode	Byte	number	Verschiebungsmodi Ein Moduswechsel kann nur bei Stillstand der X-Achse im ersten Abschnitt erfolgen.	
			0	Zeitgesteuert
			1	Winkelgesteuert mit gespeicherten Verschiebungssollwert
			2	Winkelgesteuert mit dynamischen Verschiebungssollwert
dnActXLength_p	Double Integer	position	Länge der X-Achse Verbinden Sie diesen Eingang mit dem Ausgang <i>dnActXLength_p</i> des FB L_CamData.	
dwLengthSection1_p	Double Word	position	Länge des Abschnitt 1 in [inc] - Ab dieser Position beginnt die Verschiebung. - Begrenzung: 0 ... 2³¹ [inc]	
dwLengthSection2_p	Double Word	position	Länge des Abschnitt 2 in [inc] - Dieser Wert wird zur Kontrolle der Verschiebungsbreite <i>dwShiftWindow</i> benötigt. - Begrenzung: 0 ... 2³¹ [inc]	
dwShiftWindow	Double Word	position	Verschiebungswert, abhängig vom eingestellten Modus: - Bei <i>byMode</i> = 0: - Vorgabe der Verschiebung in [ms], begrenzt auf 0 ... 60000 [ms]. - Bei <i>byMode</i> = 1 oder 2: - Vorgabe der Verschiebung in [inc]. - Maximaler Verschiebungswert = Länge des 3. Profilabschnitts.	



Ausgänge ⇒ (Variablentyp: VAR_OUTPUT)			
Bezeichner	Datentyp	Signaltyp	Wert/Bedeutung
dnXPosOut_p	Double Integer	position	Positionsangang • Bei <i>bEnable</i> = FALSE wird der Eingangswert <i>dnXPosIn_p</i> unverändert ausgegeben.
bBusy	Bool	binary	Meldesignal "Abschnittsverschiebung aktiv" TRUE Abschnittsverschiebung aktiv.
bXZeroPosTrigger	Bool	binary	Meldesignal "Nulldurchgang" • Verbinden Sie diesen Ausgang zur Initialisierung der Profildaten mit dem gleichnamigen Eingang des FB L_CamData . FALSE → TRUE Nulldurchgang • Der Zustand TRUE steht nur einen FB-Aufrufzyklus an.
nState	Integer	number	Status 12 Verschiebungswert <i>dwShiftWindow</i> zu groß in Modus 2 11 Verschiebungswert <i>dwShiftWindow</i> zu groß in Modus 1 10 Verschiebungswert <i>dwShiftWindow</i> zu groß in Modus 0 0 OK, kein Fehler -1 Fehler: Falsche Drehrichtung der X-Achse (Eingang <i>dnXPosIn_p</i>) -2 Fehler: Vorgabe an <i>dnXPosIn_p</i> außerhalb der Begrenzung

Voraussetzungen für eine Verschiebung

Für die Durchführung einer Verschiebung müssen folgende Voraussetzungen erfüllt sein:

1. Das Bewegungsprofil muss sich aus 3 Abschnitten zusammensetzen.
2. Der letzte Abschnitt muss eine Rastphase sein.
3. Die Drehrichtung der X-Achse muss positiv sein (Positionsänderung an *dnXPosIn_p*)

Beispiel:

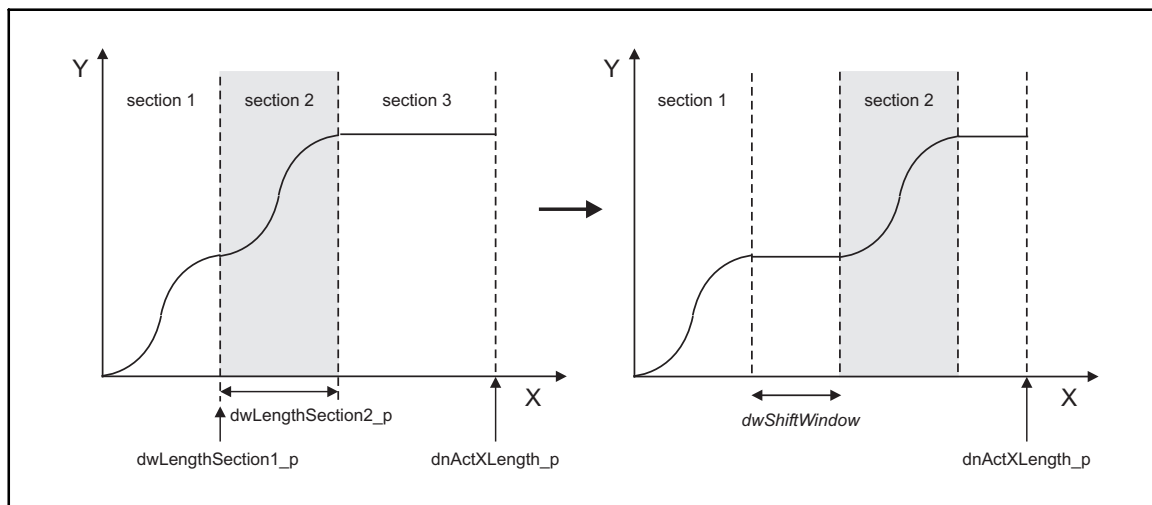
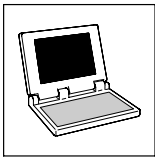


Abb. 3-12

Beispiel: Verschiebung des 2. Profilsabschnitts



Funktionsbibliothek LenzeCamControl.lib

Funktionsblöcke

L_CamSectionShift - Abschnittsverschiebung

Verschiebungsmodi

Es stehen folgende Verschiebungsmodi zur Verfügung:

Modus 0 - Zeitgesteuerte Verschiebung

- Am Eingang *dwShiftWindow* wird eine Zeit bzw. das Verhältnis aus Verschiebungszeit [ms] zu Zykluszeit [ms] vorgegeben. Der maximale Wert für *dwShiftWindow* ist 60000, was bei einer Zykluszeit von 1 ms eine Verschiebung von einer Minute entspricht:

$$\text{Vorgabe an } dwShiftWindow = \frac{\text{gewünschte Verschiebungszeit}}{\text{Taskintervall}}$$

- Ein neuer Wert wird nur übernommen, wenn sich die X-Achse im ersten Profilabschnitt befindet.



Tipp!

Typischer Anwendungsfall für diesen Modus ist eine Schweißbalkensteuerung, bei der eine definierte Schweißzeit bzw. ein definierter Stillstand des Leitwerts erforderlich ist.

Modus 1 - Winkelgesteuerte Verschiebung mit gespeichertem Verschiebungssollwert

- In diesem Modus wird die Verschiebung als Positionswert vorgegeben. Maximaler Wert ist wie für alle Modi die verbleibende Länge des dritten Profilabschnitts.
- Ein neuer Sollwert wird wie im Modus 0 nur im ersten Abschnitt übernommen.

Modus 2 - Winkelgesteuerte Verschiebung mit dynamischen Sollwert

- In diesem Modus wird ein neuer Verschiebungswinkel sofort übernommen.



Stop!

Im Modus 2 kann es durch die sofortige Übernahme des neuen Verschiebungswinkels zu Drehmomentsprüngen kommen!

Eine Umschaltung zwischen den Modi kann nur bei Stillstand der X-Achse im ersten Profilabschnitt erfolgen.

Verschiebung durchführen

Für die Durchführung und Kontrolle der Verschiebung geben Sie die Längen der Abschnitte 1 und 2 (*dwLengthSection1_p* und *dwLengthSection2_p*) sowie die Länge der X-Achse (*dnActXLength_p*) vor.

Der vorgegebene Verschiebungswinkel *dwShiftWindow* wird intern anhand dieser Parameter überwacht und ggf. auf die Größe der Rastphase (Sektion 3) begrenzt:

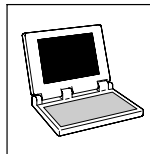
$$dwShiftWindow \leq dnActXLength_p - (dwLengthSection1_p + dwLengthSection2_p)$$

- Die Verschiebung tritt ein, wenn die an *dnXPosIn_p* vorgegebene X-Position die Start-Position *dwLengthSection1_p* erreicht.
- Solange die an *dnXPosIn_p* vorgegebene X-Position kleiner als die Summe aus *dwShiftWindow_p* + *dwLengthSection1_p* ist, bleibt die an *dnXPosOut_p* ausgegebene X-Position konstant.
- Während dieser Verschiebungsphase ist der Statusausgang *bBusy* auf TRUE gesetzt.



Tipp!

Ist eine Verschiebung zeitweise nicht erwünscht, kann der FB durch Setzen von *bEnable* auf FALSE deaktiviert werden. An *dnXPosOut_p* wird dann der Eingangswert *dnXPosIn_p* unverändert ausgegeben.



3.7 L_CamSelPosition - Leitwertauswahl

Mit diesem FB kann eine "1 aus 4"-Auswahl eines Leitwertes (Leitwinkel) realisiert werden.

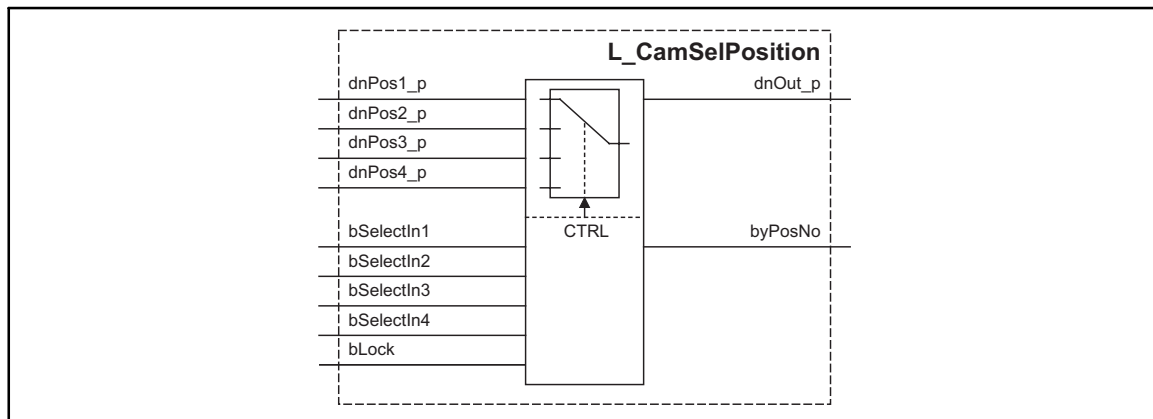


Abb. 3-13 Funktionsblock "L_CamSelPosition"

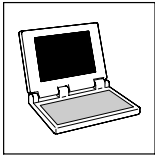
⇒ Eingänge (Variablentyp: VAR_INPUT)			
Bezeichner	Datentyp	Signaltyp	Info/Einstellmöglichkeiten
dnPosIn1_p ... dnPosIn4_p	Double Integer	position	Leitwert 1 ... Leitwert 4
bSelectIn1 ... bSelectIn4	Bool	binary	Auswahl Leitwert 1 ... Auswahl Leitwert 4
bLock	Bool	binary	Verriegelung der Auswahl
			TRUE Auswahl eines anderen Leitwerteingangs nicht möglich.

Ausgänge ⇒ (Variablentyp: VAR_OUTPUT)			
Bezeichner	Datentyp	Signaltyp	Wert/Bedeutung
dnOut_p	Double Integer	position	Ausgewählter Leitwert (<i>dnPosIn1_p</i> ... <i>dnPosIn4_p</i>)
byPosNo	Byte	number	Nummer (1 ... 4) des momentan ausgewählten Leitwertes

"1 aus 4"-Auswahl

Den 4 Eingängen *dnPosIn1_p* ... *dnPosIn4_p* können Leitwertquellen zugeordnet werden, von denen eine auf den Leitwertausgang *dnOut_p* entsprechend folgender Logik durchgeschaltet werden kann:

Zyklus	Auswahleingang				Verriegelung bLock	Ausgabe an nOut_v	siehe Anmerkung
	bSelectIn1	bSelectIn2	bSelectIn3	bSelectIn4			
n	FALSE ↗ TRUE	FALSE	FALSE ↗ TRUE	FALSE	FALSE	<i>dnPosIn1_p</i>	①
n + 1	TRUE	FALSE ↗ TRUE	FALSE	FALSE	FALSE	<i>dnPosIn2_p</i>	②
n + 2	TRUE	TRUE	FALSE	FALSE ↗ TRUE	TRUE	<i>dnPosIn2_p</i>	③
Anmerkungen:							
①	Die Auswahl des durchzuschaltenden Leitwerteingangs erfolgt durch eine FALSE-TRUE-Flanke am entsprechenden Auswahl-eingang <i>bSelectIn1</i> ... <i>bSelectIn4</i> . • Der entsprechende Auswahleingang muss nach der FALSE-TRUE-Flanke auf TRUE gesetzt bleiben. • Bei gleichzeitiger FALSE-TRUE-Flanke im selben Rechenzyklus an mehreren Auswahleingängen hat der Auswahleingang mit der niedrigeren Nummer Priorität (hier: <i>bSelectIn1</i> hat Priorität gegenüber <i>bSelectIn3</i>).						
②	Der zuletzt mit einer FALSE-TRUE-Flanke angesteuerte Eingang hat Priorität.						
③	Durch Setzen von <i>bLock</i> auf TRUE kann die Umschaltung auf einen anderen Leitwert verriegelt werden.						



Funktionsbibliothek LenzeCamControl.lib

Funktionsblöcke

L_CamSelVelocity - Leitfrequenzauswahl

3.8 L_CamSelVelocity - Leitfrequenzauswahl

Mit diesem FB kann eine "1 aus 4"-Auswahl einer Leitfrequenz realisiert werden.

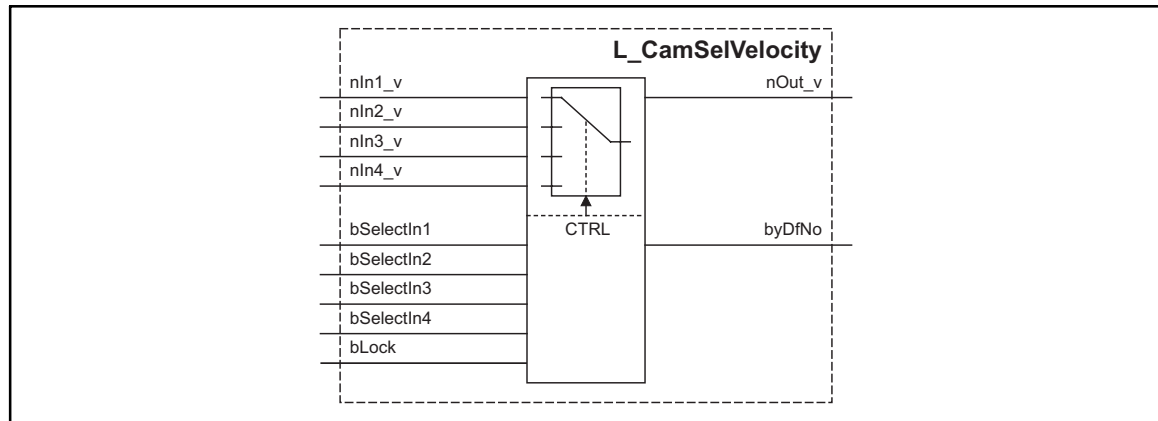


Abb. 3-14 Funktionsblock "L_CamSelVelocity"

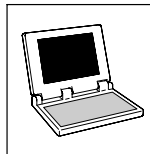
⇒ Eingänge (Variablentyp: VAR_INPUT)			
Bezeichner	Datentyp	Signaltyp	Info/Einstellmöglichkeiten
nIn1_v ... nIn4_v	Integer	velocity	Leitfrequenz 1 ... Leitfrequenz 4
bSelectIn1 ... bSelectIn4	Bool	binary	Auswahl Leitfrequenz 1 ... Auswahl Leitfrequenz 4
bLock	Bool	binary	Verriegelung der Auswahl TRUE Auswahl eines anderen Leitfrequenzeingangs nicht möglich.

Ausgänge ⇒ (Variablentyp: VAR_OUTPUT)			
Bezeichner	Datentyp	Signaltyp	Wert/Bedeutung
nOut_v	Integer	velocity	Ausgewählte Leitfrequenz (nIn1_v ... nIn4_v)
byDfNo	Byte	number	Nummer (1 ... 4) der momentan ausgewählten Leitfrequenz

"1 aus 4"-Auswahl

Den 4 Eingängen *nIn1_v* ... *nIn4_v* können Leitfrequenzquellen zugeordnet werden, von denen eine auf den Leitfrequenzausgang *nOut_v* entsprechend folgender Logik durchgeschaltet werden kann:

Zyklus	Auswahleingang				Verriegelung bLock	Ausgabe an nOut_v	siehe Anmerkung
	bSelectIn1	bSelectIn2	bSelectIn3	bSelectIn4			
n	FALSE ↗ TRUE	FALSE	FALSE ↗ TRUE	FALSE	FALSE	nIn1_v	①
n + 1	TRUE	FALSE ↗ TRUE	FALSE	FALSE	FALSE	nIn2_v	②
n + 2	TRUE	TRUE	FALSE	FALSE ↗ TRUE	TRUE	nIn2_v	③
Anmerkungen:							
①	Die Auswahl des durchzuschaltenden Leitfrequenzeingangs erfolgt durch eine FALSE-TRUE-Flanke am entsprechenden Auswahleingang <i>bSelectIn1</i> ... <i>bSelectIn4</i> . • Der entsprechende Auswahleingang muss nach der FALSE-TRUE-Flanke auf TRUE gesetzt bleiben. • Bei gleichzeitiger FALSE-TRUE-Flanke im selben Rechenzyklus an mehreren Auswahleingängen hat der Auswahleingang mit der niedrigeren Nummer Priorität (hier: <i>bSelectIn1</i> hat Priorität gegenüber <i>bSelectIn3</i>).						
②	Der zuletzt mit einer FALSE-TRUE-Flanke angesteuerte Eingang hat Priorität.						
③	Durch Setzen von <i>bLock</i> auf TRUE kann die Umschaltung auf eine andere Leitfrequenz verriegelt werden.						



3.9 L_CamVelocityFilter - Filter (PT1) für Geschwindigkeitssignale

Mit diesem FB können Sie ein Geschwindigkeitssignal nach dem PT1-Prinzip filtern bzw. verzögern. Der hierbei entstehende Positionsverlust wird über einen internen Nachlaufregler kompensiert.



Hinweis!

Verwenden Sie diesen FB nur in einer Task mit festem Zeitintervall (z. B. T#2ms).

Um ein Driften der Position zu vermeiden, deaktivieren Sie die Filterfunktion während Beschleunigungs- und Verzögerungsphasen. Setzen Sie hierzu *bEnable* auf FALSE.

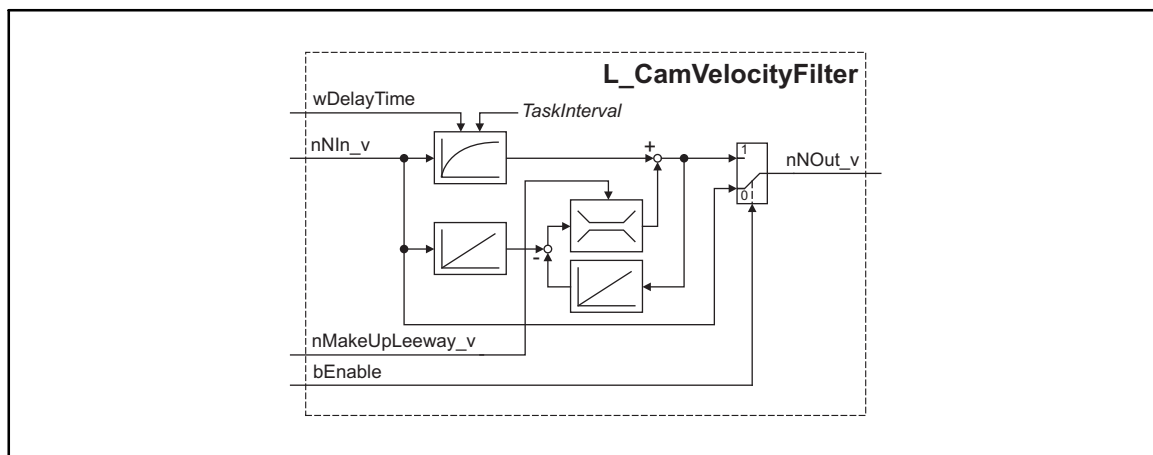
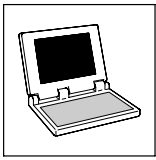


Abb. 3-15

Funktionsblock "L_CamVelocityFilter"

⇒ Eingänge (Variablentyp: VAR_INPUT)				
Bezeichner	Datentyp	Signaltyp	Info/Einstellmöglichkeiten	
wDelayTime	Word	time	Verzögerungszeit in [ms] - Einstellbereich: 1 ... 65535 [ms] - Bei aktiviertem Filter muss die Verzögerungszeit größer als das Taskintervall eingestellt sein.	
nNIn_v	Integer	velocity	Geschwindigkeitseingang 16384 inc/ms ≙ 15000 rpm	
nMakeUpLeeway_v	Integer	velocity	Ausgleichsgeschwindigkeit zur Kompensation des Positionsverlustes 16384 inc/ms ≙ 15000 rpm	
bEnable	Bool	binary	Aktivierung des Filters	
			FALSE	Filter deaktiviert
			TRUE	Filter aktiviert

Ausgänge ⇒ (Variablentyp: VAR_OUTPUT)			
Bezeichner	Datentyp	Signaltyp	Wert/Bedeutung
nNOut_v	Integer	velocity	Geschwindigkeitsausgang 16384 inc/ms ≙ 15000 rpm



Funktionsbibliothek *LenzeCamControl.lib*

Funktionsblöcke

L_CamVelocityFilter - Filter (PT1) für Geschwindigkeitssignale

Einstellungen

- Geben Sie über *wDelayTime* die gewünschte Verzögerungszeit T in [ms] vor, diese muss für einen einwandfreien Betrieb des Filters größer als das Taskintervall sein.
- Der Proportionalanteil ist mit $K = 1$ fest vorgegeben:

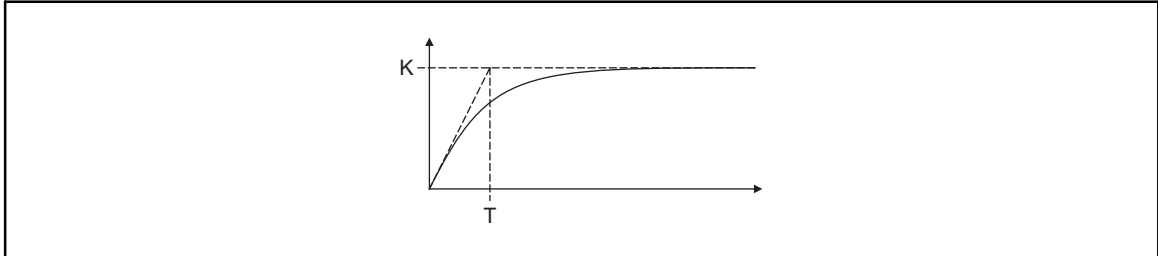
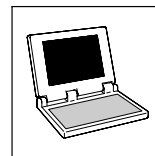


Abb. 3-16

Verzögerungszeit T in [ms]

- Geben Sie über *nMakeUpLeeway_v* eine Ausgleichsgeschwindigkeit > 0 [inc/ms] vor, um einen Positionsverlust zu vermeiden.



4 Index

A

Abschnittsverschiebung, 3-14
Absolute Positionierung, 2-34 , 2-48
Auf Nullposition fahren, 2-48
Auskuppeln über Rampenfunktion, 2-6
Auswahl/Umschaltung des Bewegungsprofils, 2-33

B

Begrenzung des Positionswertes bei einer Rundachse, 2-18
Begrenzungseingänge, 2-41
Begrenzungsglied, 2-41
Begriffsdefinitionen, 1-2
Beschleunigungs-/Verzögerungsrampen, 1-5

C

CAM_PROFILE_REF, 2-51

D

Datenstruktur CAM_PROFILE_REF, 2-51
Dauerbetrieb, 2-55
Deaktivierung der Schleppfehlerüberwachung, 2-45
Dehnen & Stauchen des Leitwerts, 2-59
Dehnen & Stauchen des Sollwerts, 2-64
Drehmomentvorsteuerung, 2-27 , 2-37
Drehrichtung ändern, 2-61
Drehrichtungserkennung, 2-17
Drehrichtungsvorgabe, 2-9
Drehzahl zu einem Winkel integrieren, 3-12
Drehzahl-Addition, 3-2
Drehzahlbegrenzung ohne Positionsverlust, 2-41
Drehzahlvorsteuerung, 2-27
Dynamische Verschiebung, 2-18
Dynamische Verstellung der Nocken/Hysterese, 2-16

E

Einheitenumrechnung, 2-40 , 2-58
Einkuppeln über Rampenfunktion, 2-5
Endabschaltung, 2-48
Erzeugung der virtuellen Leitfrequenz, 2-75
Externe Masterposition, 2-73
Externes Master-Signal durchschleifen, 2-61
Extrapolation, 2-38

F

Filter (PT1) für Geschwindigkeitssignale, 3-19
Freischalteingang "bEnable", 1-5
Funktionsbibliotheken, Versionsinfo, 1-3
Funktionsblöcke mit *.lc9-Zugriff, 1-5

G

Generierung und Positionierung der Königswelle (X-Achse), 2-70
Geschwindigkeitssignale filtern, 3-19
Getrennte Rückführung für Drehzahl- und Lageregelung, 2-26
Glättung eines gering auflösenden Absolutwertgebers, 2-39

H

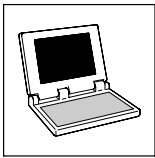
Handfahrbetrieb (Tippbetrieb), 2-72 , 2-76
Handradfunktion, 2-72
Hauptlagesollwert, 2-23
Hilfslagesollwert zum abgelösten Positionieren, 2-24
Hochlaufgeber freischalten, 2-75
Hysterese, 2-19

I

Integration von Leitfrequenzsignalen zu einer Lageposition, 2-23

K

Kompensation von Busübertragungszyklen > 1ms, 2-39
Differenzierer, 3-4
Kupplungsfunktion aktivieren, 2-4



Funktionsbibliothek LenzeCamControl.lib

Index

L

L_CamAddVelocity, 3-2
L_CamAdjustDfIn, 3-3
L_CamClutchXAxis, 2-2
L_CamContactor, 2-7
L_CamContactorVPos, 2-16
L_CamControl, 2-20
L_CamData, 2-28
L_CamDifferentiate, 3-4
L_CamEditProfileData, 3-5
L_CamExtrapolate, 2-38
L_CamGetPosition, 2-40
L_CamIntegrateVarLim, 3-11
L_CamLimitVelocity, 2-41
L_CamMachineData, 2-42
L_CamMonitFollowError, 2-44
L_CamPosCtrlLin, 2-46
L_CamProfileData, 2-50
L_CamSectionShift, 3-14
L_CamSelPosition, 3-17
L_CamSelVelocity, 3-18
L_CamSequence, 2-52
L_CamSetPosition, 2-58
L_CamSetXAxisVelocity, 2-59
L_CamSetYAxisVelocity, 2-64
L_CamVelocityFilter, 3-19
L_CamVMasterPosition, 2-68
L_CamVMasterVelocity, 2-74
Laden des Hilfsintegrators, 2-24
Laden wichtiger Profilpunkte, 2-50
Ic9-Datei, 1-5
Leitfrequenzanpassung, 3-3
Leitfrequenzauswahl, 3-18
Leitwert (X-Position) für den Kurvenantrieb vorgeben, 2-32
Leitwertauswahl, 3-17
Linearer Profilgenerator, 2-46

M

Maschinenparameter, 2-42 , 2-43
Maximale Anzahl der Stützpunkte, 2-37

N

Nockenschaltwerk, 2-7
Normalbetrieb, 2-75
Normalbetrieb (mit automatischem Stoppvorgang), 2-71
Nullposition anfahren, 2-48

O

Offset, 2-67
Offset für den Leitwinkel, 2-32
Online-Änderung der Profilpunkte, 3-5

P

Positionen laden, 3-13
Positionierung stoppen, 2-48
Profilpunkte online ändern, 3-5
Profilverwaltung, 2-28

R

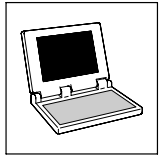
Relative Positionierung, 2-35 , 2-48
Rückführung für Drehzahl- und Lageregelung, 2-26

S

Schleppfehlerüberwachung, 2-44
Sicherheitshinweise, Gestaltung
 Sonstige Hinweise, 1-1
 Warnung vor Sachschäden, 1-1
Sofortiges Auskuppeln (Zwangsöffnung), 2-6
Sollwertaufbereitung, 2-20
Statische Verschiebung, 2-19
Statusausgang "nState", 1-5
Stopp-Funktion, 2-76
Synchronisation der Y-Achse über Touch-Probe, 2-25
Synchronisiertes Dehnen und Stauchen der Y-Achse, 2-36 , 2-67
Synchronisierung der X-Achse über Touch-Probe, 2-63

T

Taktbetrieb, 2-55
Taktbetrieb (mit automatischem Stoppvorgang), 2-72
Touch-Probe, 2-25 , 2-63
Typ des Bewegungsprofils, 2-34



U

Übergabe der Maschinenparameter, 2-43
unit, 1-2

V

Verfahrenfunktion aktivieren, 2-71
Verschiebung durchführen, 3-16
Verschiebungsmodi, 3-16
Virtuelle Kupplung, 2-2
Virtueller Master, 2-74
Virtueller Master mit Positionierfunktion, 2-68
Voraussetzungen für eine Verschiebung, 3-15

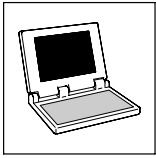
Vorgabe einer Profilsequenz, 2-52
Vorgabe eines Offset für die Y-Achse, 2-49
Vorschubantrieb über Kurven, 2-35

W

Wegnocke, 2-15
Winkelintegrator, 3-11

Z

Zeitäquidistanter Aufruf, 1-4
Zeitnocke, 2-15
Zwangsextrapolation, 2-39
Zwangsstopp, 2-73



Funktionsbibliothek LenzeCamControl.lib

Index