

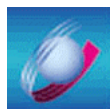
# Lenze

*Handbuch*

IEC 61131-3

*inside*

***Global Drive  
PLC Developer Studio***



***Global Drive***

*Funktionsbibliothek*

*LenzeDrive.lib*

Die Funktionsbibliothek **LenzeDrive.lib** kann für folgende Lenze PLC-Geräte verwendet werden:

|                       | Typenbezeichnung | ab Hardwarestand | ab Softwarestand |
|-----------------------|------------------|------------------|------------------|
| <b>9300 Servo PLC</b> | EVS93XX-xI       | 2K               | 1.0              |
| <b>9300 Servo PLC</b> | EVS93XX-xT       | 2K               | 1.0              |
| <b>Drive PLC</b>      | EPL10200         | VA               | 1.0              |
| <b>ECSxA</b>          | ECSxAxxx         | 1C               | 7.0              |

### 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.

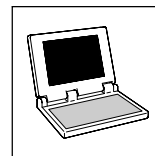
Wir übernehmen keine Verantwortung für direkt oder indirekt entstandene Schäden, z. B. Gewinnverluste, Auftragsverluste oder geschäftliche Beeinträchtigungen jeglicher Art.

© 2006 Lenze Drive Systems GmbH

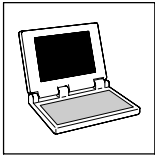
Ohne besondere schriftliche Genehmigung von Lenze Drive Systems GmbH darf kein Teil dieser Dokumentation vervielfältigt oder Dritten zugänglich gemacht werden.

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.



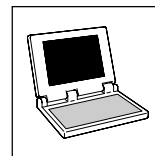
|          |                                                |            |
|----------|------------------------------------------------|------------|
| <b>1</b> | <b>Vorwort und Allgemeines</b>                 | <b>1-1</b> |
| 1.1      | Über dieses Handbuch                           | 1-1        |
| 1.1.1    | Konventionen in diesem Handbuch                | 1-1        |
| 1.1.2    | Aufbau der Beschreibungen                      | 1-2        |
| 1.1.3    | Piktogramme in diesem Handbuch                 | 1-2        |
| 1.1.4    | Verwendete Begriffe                            | 1-2        |
| 1.2      | Lenze-Softwarerichtlinie für Variablennamen    | 1-3        |
| 1.2.1    | Anlehnung an die Ungarische Notation           | 1-3        |
| 1.2.1.1  | Vorschlag zur Kennzeichnung von Variablentypen | 1-4        |
| 1.2.1.2  | Kennzeichnung des Signaltyps im Variablennamen | 1-5        |
| 1.2.1.3  | Sonderbehandlung bei Systemvariablen           | 1-5        |
| 1.3      | Versionskennung der Funktionsbibliothek        | 1-6        |
| <b>2</b> | <b>Funktionsblöcke</b>                         | <b>2-1</b> |
| 2.1      | Signalverarbeitung allgemein                   | 2-2        |
| 2.1.1    | Programmierung Fix-Sollwerte (L_FIXSET)        | 2-2        |
| 2.2      | Analoge Signalverarbeitung                     | 2-4        |
| 2.2.1    | Betragsbildung (L_ABS)                         | 2-4        |
| 2.2.2    | Addition (L_ADD)                               | 2-5        |
| 2.2.3    | Eingangsverstärkung und Offset (L_AIN)         | 2-6        |
| 2.2.4    | Invertierung (L_ANEG)                          | 2-8        |
| 2.2.5    | Ausgangsverstärkung und Offset (L_AOUT)        | 2-9        |
| 2.2.6    | Arithmetik (L_ARIT)                            | 2-11       |
| 2.2.7    | Umschaltung (L_ASW)                            | 2-12       |
| 2.2.8    | Vergleich (L_CMP)                              | 2-13       |
| 2.2.9    | Kennlinienfunktion (L_CURVE)                   | 2-17       |
| 2.2.10   | Totgang (L_DB)                                 | 2-20       |
| 2.2.11   | Differenzierung (L_DT1_)                       | 2-21       |
| 2.2.12   | Begrenzung (L_LIM)                             | 2-22       |
| 2.2.13   | Verzögerung (L_PT1_)                           | 2-23       |
| 2.2.14   | Hochlaufgeber (L_RFG)                          | 2-24       |
| 2.2.15   | Sample & Hold (L_SH)                           | 2-26       |
| 2.2.16   | Hochlaufgeber S-Rampe (L_SRFG)                 | 2-27       |
| 2.3      | Digitale Signalverarbeitung                    | 2-29       |
| 2.3.1    | Logisches UND (L_AND)                          | 2-29       |
| 2.3.2    | Verzögerung (L_DIGDEL)                         | 2-30       |
| 2.3.3    | Auf-/Abwärtszähler (L_FCNT)                    | 2-32       |
| 2.3.4    | Flip-Flop (L_FLIP)                             | 2-33       |
| 2.3.5    | Logisches NICHT (L_NOT)                        | 2-34       |
| 2.3.6    | Logisches ODER (L_OR)                          | 2-35       |
| 2.3.7    | Flankenauswertung (L_TRANS)                    | 2-36       |
| 2.4      | Verarbeitung von Winkelsignalen                | 2-38       |
| 2.4.1    | Arithmetik (L_ARITPH)                          | 2-38       |
| 2.4.2    | Addition (L_PHADD)                             | 2-39       |
| 2.4.3    | Vergleich (L_PHCMP)                            | 2-40       |
| 2.4.4    | Differenz (L_PHDIFF)                           | 2-41       |
| 2.4.5    | Division (L_PHDIV)                             | 2-42       |
| 2.4.6    | Integration (L_PHINT)                          | 2-43       |
| 2.4.7    | Integration (L_PHINTK)                         | 2-45       |



# **Funktionsbibliothek LenzeDrive.lib**

## **Inhalt**

|          |                                                                             |            |
|----------|-----------------------------------------------------------------------------|------------|
| 2.5      | Signalkonvertierung .....                                                   | 2-49       |
| 2.5.1    | Normierung (L_CONV) .....                                                   | 2-49       |
| 2.5.2    | Konvertierung Winkel nach Analog (L_CONVPA) .....                           | 2-50       |
| 2.5.3    | Umrechnung Winkelsignal (L_CONVPP) .....                                    | 2-51       |
| 2.5.4    | Umrechnung (L_CONVVV) .....                                                 | 2-52       |
| 2.5.5    | Normierung mit Begrenzung (L_CONVX) .....                                   | 2-53       |
| 2.6      | Kommunikation .....                                                         | 2-54       |
| 2.6.1    | Typ-Konvertierung (L_ByteArrayToDint) .....                                 | 2-54       |
| 2.6.2    | Typ-Konvertierung (L_DintToByteArray) .....                                 | 2-54       |
| 2.6.3    | Index einer Codestelle (L_FUNCCodeIndexConv) .....                          | 2-54       |
| 2.6.4    | Lesen von Codestellen (L_ParRead) .....                                     | 2-55       |
| 2.6.5    | Beschreiben von Codestellen (L_ParWrite) .....                              | 2-59       |
| 2.7      | Spezielle Funktionen .....                                                  | 2-63       |
| 2.7.1    | Transparenter Modus mit Bedieneinheit 9371BB/9371BC (L_Display9371BB) ..... | 2-63       |
| 2.7.2    | Fehlerauslösung (L_FWM) .....                                               | 2-68       |
| 2.7.3    | Motorpotentiometer (L_MPOT) .....                                           | 2-70       |
| 2.7.4    | Drehzahl-Aufbereitung (L_NSET) .....                                        | 2-73       |
| 2.7.5    | Prozessregler (L_PCTRL) .....                                               | 2-79       |
| 2.7.6    | Rechts/Links/Quickstop (L_RLQ) .....                                        | 2-83       |
| <b>3</b> | <b>Anhang .....</b>                                                         | <b>3-1</b> |
| 3.1      | Codetabelle .....                                                           | 3-1        |
| <b>4</b> | <b>Index .....</b>                                                          | <b>4-1</b> |



## 1 Vorwort und Allgemeines

### 1.1 Über dieses Handbuch

Dieses Handbuch enthält Informationen zu den Funktionsblöcken, die in der Funktionsbibliothek **LenzeDrive.lib** für das **Drive PLC Developer Studio** enthalten sind.

- Die Funktionsblöcke können Sie in den Automatisierungssystemen **9300 Servo PLC**, **Drive PLC** und **ECSxA** verwenden.
- Die Funktionsblöcke basieren auf den Funktionen, die im Servo-Umrichter 9300 (V2.0) zur Verfügung stehen.

Im **Drive PLC Developer Studio** (DDS) nehmen Sie *offline* die Grundeinstellung Ihrer Antriebsaufgabe vor, indem Sie mit Hilfe von Variablen (gemäß dem IEC61131-3 Standard) die entsprechenden Funktionsblöcke parametrieren.

Mit **Global Drive Control** (GDC) oder **Keypad** können Sie dann *online* die genaue Funktionalität Ihrer Antriebsaufgabe parametrieren, indem Sie auf die Codestellen der Funktionsblock-Instanzen zugreifen.

#### 1.1.1 Konventionen in diesem Handbuch

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

##### Variablennamen

sind im erklärenden Text in Kursivschrift dargestellt:

- "Das Signal an *nIn\_a* ..."

##### Lenze Funktionen/Funktionsblöcke

erkennen Sie am Namen. Sie beginnen immer mit einem "L\_":

- "Der FB **L\_ARIT** kann ..."

##### Programmlistings

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

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

##### Instanzen

Bei Funktionsblöcken, die eine oder mehrere Erstinstanzen haben, sind in einer Tabelle die entsprechenden Codestellen beschrieben:

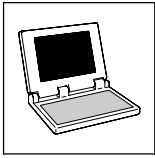
| Variablenname | L_ARIT1 | L_ARIT2 |  | Einstellbereich | Lenze |
|---------------|---------|---------|--|-----------------|-------|
| byFunction    | C0338   | C0600   |  | 0 ... 5         | 1     |

Auf diese Codestellen können Sie *online* mit **Global Drive Control** (GDC) oder **Keypad** zugreifen.



### Tipp!

Über den Parametermanager können Sie diesen Instanzen die gleichen Codestellen wie im Servo-Umrichter 9300 (V2.0) zuordnen.



# Funktionsbibliothek LenzeDrive.lib

## Vorwort und Allgemeines Aufbau der Beschreibungen

### 1.1.2 Aufbau der Beschreibungen

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

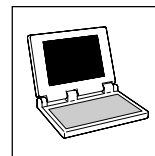
|   | Funktion                                                                                                                    | Funktionsblock (FB)/<br>Systembaustein (SB)                                                                                                       |
|---|-------------------------------------------------------------------------------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

### 1.1.3 Piktogramme in diesem Handbuch

|                         | Verwendete Piktogramme | Signalwörter                                                                                                                                                           |
|-------------------------|------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Warnung vor Sachschäden |                        | <b>Stop!</b><br>Warnt vor <b>möglichen Sachschäden</b> .<br>Mögliche Folgen bei Missachtung:<br>Beschädigung des Antriebsreglers/Antriebssystems oder seiner Umgebung. |
| Sonstige Hinweise       |                        | <b>Tipp!<br/>Hinweis!</b><br>Kennzeichnet einen Tipp bzw. Hinweis.                                                                                                     |

### 1.1.4 Verwendete Begriffe

| Begriff               | Im folgenden Text verwendet für                                                                                             |
|-----------------------|-----------------------------------------------------------------------------------------------------------------------------|
| 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                   | <ul style="list-style-type: none"> <li>9300 Servo PLC</li> <li>Drive PLC</li> <li>Achsmodule ECSxA "Application"</li> </ul> |
| SB                    | Systembaustein                                                                                                              |



## 1.2 Lenze-Softwarerichtlinie für Variablennamen

Die bisherigen Konzepte der Lenze-Antriebsregler basierten auf Codestellen, die die Ein-, Ausgangssignale und Parameter von Funktionsblöcken repräsentieren.

- In der Dokumentation wurden den Codestellen definierte Namen zur besseren Unterscheidung gegeben.
- Zusätzlich wurde eine grafische Symbolik der Signalarten definiert.

Der Anwender konnte auf einen Blick erkennen, was für ein Signal (analog, Winkel usw.) an der entsprechenden Schnittstelle anliegen muss.

**Das Konzept der neuen Automatisierungssysteme verwendet in der Programmierung keine direkten Codestellen. Stattdessen wird der Standard der IEC 61131-3 verwendet.**

- Dieser Standard basiert auf der Struktur von Variablennamen.
- Verwendet der Anwender in seinem Projekt Variablen, so kann er die Variablennamen beliebig benennen.

Damit für bestehende und zukünftige von Lenze-Mitarbeitern programmierte Projekte und Funktionsbibliotheken nicht eine Vielzahl von verschiedenen Konventionen zur Variablennamensbildung entsteht, wurde eine Softwarerichtlinie geschaffen, an die sich jeder Lenze-Mitarbeiter zu halten hat.

Die Firma Lenze hält sich bei der Konvention zur Variablennamensbildung an die "Ungarische Notation", die Lenze spezifisch erweitert hat.

Wenn Sie als Anwender Lenze-spezifische Funktionen/Funktionsblöcke verwenden, sehen Sie auf den ersten Blick, welchen Datentyp Sie zum Beispiel an einen Funktionsblock übergeben müssen und welchen Datentyp Sie als Ausgangswert zurückbekommen.

### 1.2.1 Anlehnung an die Ungarische Notation

Diese Konventionen werden mitunter benutzt, um aus dem Namen einer Programmgröße sofort auf deren wichtigste Eigenschaften schließen zu können.

#### Variablennamen

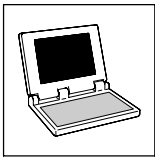
bestehen aus

- einem **Präfix** (wahlweise)
- einer **Datentypangabe**
- und einem **Identifikator**

Präfix und Datentypangabe werden meist durch ein bis zwei Zeichen gebildet. Der Identifikator (der "eigentliche" Name) sollte auf den Verwendungszweck hinweisen und ist deshalb meist länger.

#### Beispiele für den Präfix

| Präfix | Bedeutung                           |
|--------|-------------------------------------|
| a      | Array (zusammengesetzter Typ), Feld |
| p      | Zeiger (pointer)                    |



# Funktionsbibliothek LenzeDrive.lib

## Vorwort und Allgemeines

Anlehnung an die Ungarische Notation

### Beispiele für die Datentypangabe

| Beispiele für den Datentyp | Bedeutung               |
|----------------------------|-------------------------|
| b                          | Bool                    |
| by                         | Byte                    |
| n                          | Integer                 |
| w                          | Word                    |
| dn                         | Double Integer          |
| dw                         | Double Word             |
| s                          | String                  |
| f                          | Real (Float)            |
| sn                         | Short Integer           |
| t                          | Time                    |
| un                         | Unsigned Integer        |
| udn                        | Unsigned Double Integer |
| usn                        | Unsigned Short Integer  |

### Identifikator (eigentlicher Variablenname)

- Ein Identifikator beginnt mit einem Großbuchstaben.
- Setzt sich ein Identifikator aus mehreren "Worten" zusammen, so beginnt jedes "Wort" mit einem Großbuchstaben.
- Alle anderen Buchstaben werden klein geschrieben.

#### Beispiele:

|                  |                                   |
|------------------|-----------------------------------|
| Array of Integer | <i>anJogValue[10];</i>            |
| Bool             | <i>bIsEmpty;</i>                  |
| Word             | <i>wNumberOfValues;</i>           |
| Integer          | <i>nLoop;</i>                     |
| Byte             | <i>byCurrentSelectedJogValue;</i> |

### 1.2.1.1 Vorschlag zur Kennzeichnung von Variablentypen

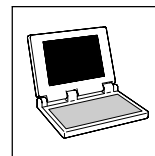
Um den Variablentyp anhand des Variablennamens im Programm zu erkennen, sind folgende Bezeichner, die vor dem eigentlichen Variablennamen mit einem Unterstrich abgetrennt sind, sinnvoll:

|                     |                            |
|---------------------|----------------------------|
| I_<Variablenname>   | VAR_INPUT                  |
| Q_<Variablenname>   | VAR_OUTPUT                 |
| IQ_<Variablenname>  | VAR_IN_OUT                 |
| R_<Variablenname>   | VAR_RETAIN                 |
| C_<Variablenname>   | VAR_CONSTANT               |
| CR_<Variablenname>  | VAR_CONSTANT_RETAIN        |
| g_<Variablenname>   | VAR_GLOBAL                 |
| gR_<Variablenname>  | VAR_GLOBAL_RETAIN          |
| gC_<Variablenname>  | VAR_GLOBAL_CONSTANT        |
| gCR_<Variablenname> | VAR_GLOBAL_CONSTANT_RETAIN |

#### Beispiel

für ein globales Array vom Typ Integer, das fixe Sollwerte (analog) für eine Drehzahl enthält:

*g\_anFixSetSpeedValue\_a*



## 1.2.1.2 Kennzeichnung des Signaltyps im Variablennamen

Ein- bzw. Ausgänge der Lenze-Funktionsblöcke gehören jeweils einem bestimmten Signaltyp an. Hierbei existieren digitale, analoge, Lage- sowie Drehzahl-signale.

Aus diesem Grund wird jedem Variablennamen eine Endung angefügt, die eine Aussage über den Signaltyp trifft.

| Signaltyp                     | Endung        | Bisherige Kennung |
|-------------------------------|---------------|-------------------|
| analog                        | _a (analog)   | ○                 |
| digital                       | _b (binary)   | □                 |
| Winkeldifferenz oder Drehzahl | _v (velocity) | △                 |
| Winkel oder Lage              | _p (position) | ▲                 |



### Tipp!

Normierung auf Signaltyp Winkeldifferenz/Drehzahl: 16384 (INT)  $\triangleq$  15000 rpm

Normierung auf Signaltyp analog: 16384  $\triangleq$  100 %  $\triangleq$  Wert unter [C0011] =  $n_{\max}$

Normierung auf Signaltyp Winkel oder Lage: 65536  $\triangleq$  1 Motorumdrehung

### Beispiele:

| Variablenname         | Signaltyp                 | Variablentyp   |
|-----------------------|---------------------------|----------------|
| nIn_a                 | Analoger Eingangswert     | Integer        |
| dnPhiSet_p            | Winkelsignale             | Double Integer |
| bLoad_b               | Binärer Wert (TRUE/FALSE) | Bool           |
| nDigitalFrequencyIn_v | Drehzahl-Eingangswert     | Integer        |

## 1.2.1.3 Sonderbehandlung bei Systemvariablen

Systemvariablen benötigen eine gesonderte Behandlung, denn die Systemfunktionen in der Steuerungskonfiguration stehen dem Anwender nur als I/O-Verbindung zur Verfügung.

Um beim Programmieren schnell auf eine Systemvariable zugreifen zu können, muss im Variablennamen eine Kennung der Systemfunktion enthalten sein.

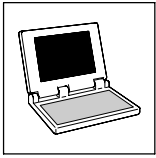
Aus diesem Grund wird dem Variablennamen noch der Name des entsprechenden Systembausteins vorgesetzt.

### Beispiele:

AIN1\_nIn\_a

CAN1\_bCtrlTripSet\_b

DIGIN\_bIn3\_b



# Funktionsbibliothek LenzeDrive.lib

## Vorwort und Allgemeines

Anlehnung an die Ungarische Notation

### 1.3 Versionskennung der Funktionsbibliothek

Die Version der Funktionsbibliothek können Sie über die globale Konstante `C_w[Funktionsbibliotheksname]Version` abfragen.

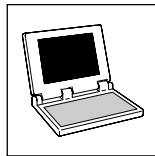
Versionskennung ab PLC Software-Stand 7.x:

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

Version: 01 05 00 00

Der Wert dieser Konstanten ist hexadezimal-kodiert.

- Im Beispiel bedeutet der Wert "01050000" Version "1.05".



## 2 Funktionsblöcke

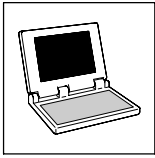


### Hinweis!

Folgende Funktionsblöcke müssen aufgrund ihrer internen Struktur in einer zeitäquidistanten Task aufgerufen werden (z. B. in einer 5 ms-Task):

- L\_DIGDEL
- L\_DT1\_
- L\_MPOT
- L\_NSET
- L\_ParRead
- L\_ParWrite
- L\_PCTRL
- L\_PHDIFF
- L\_PHINT
- L\_PHINTK
- L\_PT1\_
- L\_RFG
- L\_SRFG
- L\_TRANS

**Beachten Sie:** Die zyklische Task PLC\_PRG ist nicht zeitäquidistant!



# Funktionsbibliothek LenzeDrive.lib

## Signalverarbeitung allgemein

### Programmierung Fix-Sollwerte (L\_FIXSET)

## 2.1 Signalverarbeitung allgemein

### 2.1.1 Programmierung Fix-Sollwerte (L\_FIXSET)

Mit diesem FB können Sie bis zu 15 feste Sollwerte programmieren. Die Adressierung des auszugebenden Sollwertes erfolgt über die Booleschen Eingänge.

Fix-Sollwerte lassen sich z. B. nutzen für:

- Verschiedene Tänzersolllagen bei einer Tänzerlageregelung
- Verschiedene Reckverhältnisse (Getriebefaktor) bei einer Drehzahlverhältnisregelung mit Leitfrequenzkopplung

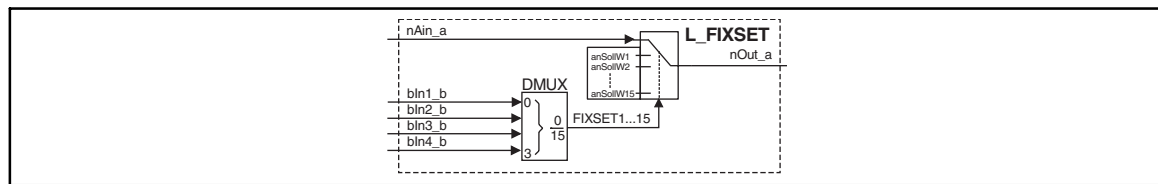


Abb. 2-1

Programmierung Fix-Sollwerte (L\_FIXSET)

| VariableName    | DataType         | SignalType | VariableType        | Note                                                                                                  |
|-----------------|------------------|------------|---------------------|-------------------------------------------------------------------------------------------------------|
| nAin_a          | Integer          | analog     | VAR_INPUT           | nAin_a wird auf nOut_a geschaltet, wenn an allen Selektionseingängen (bln1_b ... bln4_b) FALSE liegt. |
| bln1_b          | Bool             | binary     | VAR_INPUT           | Die Zahl der zu belegenden Eingänge ist abhängig von der Anzahl der benötigten Fix-Sollwerte.         |
| bln2_b          | Bool             | binary     | VAR_INPUT           |                                                                                                       |
| bln3_b          | Bool             | binary     | VAR_INPUT           |                                                                                                       |
| bln4_b          | Bool             | binary     | VAR_INPUT           |                                                                                                       |
| nOut_a          | Integer          | analog     | VAR_OUTPUT          |                                                                                                       |
| anSollW[1...15] | Array of Integer |            | VAR CONSTANT RETAIN | Variablen, denen Sie Fix-Sollwerte zuordnen können.                                                   |

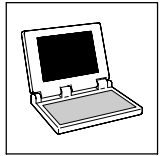
#### Parameter-Codestellen der Instanzen

| VariableName    | L_FIXSET1      |  | SettingRange         | Lenze |
|-----------------|----------------|--|----------------------|-------|
| anSollW[1...15] | C0560/1 ... 15 |  | -199.99 ... 199.99 % | 0.00  |

#### Funktion

nOut\_a lässt sich als Sollwertquelle (Signalquelle) für einen weiteren FB (z. B. Prozessregler, Arithmetikblock usw.) nutzen. Die Parametrierung und Handhabung erfolgt wie bei JOG, ist aber unabhängig von JOG. (2-73: L\_NSET)

- Parametrierung der Fix-Sollwerte:
  - Die einzelnen Fix-Sollwerte parametrieren Sie über anSollW1 ... anSollW15.
- Ausgabe des gewählten Fix-Sollwertes:
  - Bei Ansteuerung der binären Eingänge mit HIGH-Signal wird ein Fix-Sollwert aus der Tabelle auf nOut\_a geschaltet. (2-3)
- Wertebereich:
  - Sie können Werte von -199,99 % ... 199,99 % vorgeben (100 % entspricht 16384).



## 2.1.1.1 Freigabe der Fix-Sollwerte

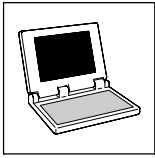
| Anzahl der benötigten Fix-Sollwerte | Anzahl der zu belegenden Eingänge |
|-------------------------------------|-----------------------------------|
| 1                                   | mindestens 1                      |
| 1 ... 3                             | mindestens 2                      |
| 4 ... 7                             | mindestens 3                      |
| 8 ... 15                            | 4                                 |

Schema für die Dekodierung der binären Eingangssignale:

| Ausgangssignal<br>nOut_a = | 1. Eingang<br>bln1_b | 2. Eingang<br>bln2_b | 3. Eingang<br>bln3_b | 4. Eingang<br>bln4_b |
|----------------------------|----------------------|----------------------|----------------------|----------------------|
| nAin_a                     | 0                    | 0                    | 0                    | 0                    |
| anSolIW1                   | 1                    | 0                    | 0                    | 0                    |
| anSolIW2                   | 0                    | 1                    | 0                    | 0                    |
| anSolIW3                   | 1                    | 1                    | 0                    | 0                    |
| anSolIW4                   | 0                    | 0                    | 1                    | 0                    |
| anSolIW5                   | 1                    | 0                    | 1                    | 0                    |
| anSolIW6                   | 0                    | 1                    | 1                    | 0                    |
| anSolIW7                   | 1                    | 1                    | 1                    | 0                    |
| anSolIW8                   | 0                    | 0                    | 0                    | 1                    |
| anSolIW9                   | 1                    | 0                    | 0                    | 1                    |
| anSolIW10                  | 0                    | 1                    | 0                    | 1                    |
| anSolIW11                  | 1                    | 1                    | 0                    | 1                    |
| anSolIW12                  | 0                    | 0                    | 1                    | 1                    |
| anSolIW13                  | 1                    | 0                    | 1                    | 1                    |
| anSolIW14                  | 0                    | 1                    | 1                    | 1                    |
| anSolIW15                  | 1                    | 1                    | 1                    | 1                    |

0 = FALSE

1 = TRUE



# Funktionsbibliothek LenzeDrive.lib

## Analoge Signalverarbeitung

### Betragsbildung (L\_ABS)

## 2.2 Analoge Signalverarbeitung

### 2.2.1 Betragsbildung (L\_ABS)

Dieser FB überführt bipolare Werte in unipolare Werte. Es wird der Absolutwert vom Eingangssignal gebildet.

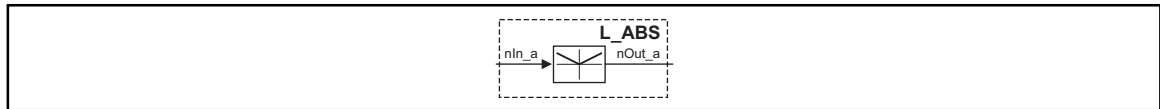
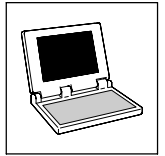


Abb. 2-2 Betragsbildung (L\_ABS)

| VariableName | DataType | SignalType | VariableType | Note |
|--------------|----------|------------|--------------|------|
| nIn_a        | Integer  | analog     | VAR_INPUT    |      |
| nOut_a       | Integer  | analog     | VAR_OUTPUT   |      |



### 2.2.2 Addition (L\_ADD)

Dieser FB addiert oder subtrahiert Eingangswerte, je nach verwendetem Eingang.

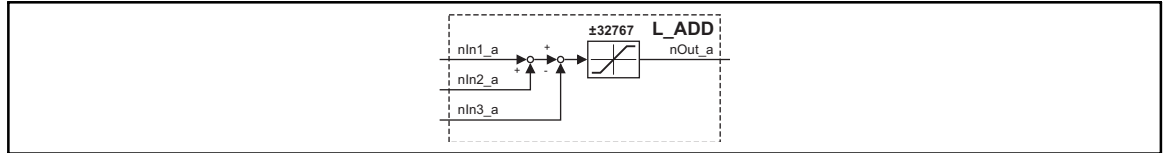


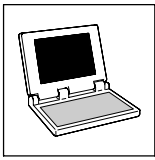
Abb. 2-3

Addition (L\_ADD)

| VariableName | DataType | SignalType | VariableType | Note                                   |
|--------------|----------|------------|--------------|----------------------------------------|
| nIn1_a       | Integer  | analog     | VAR_INPUT    | Additionseingang                       |
| nIn2_a       | Integer  | analog     | VAR_INPUT    | Additionseingang                       |
| nIn3_a       | Integer  | analog     | VAR_INPUT    | Subtraktionseingang                    |
| nOut_a       | Integer  | analog     | VAR_OUTPUT   | Signal wird begrenzt auf $\pm 32767$ . |

#### Funktionsablauf

1. Der Wert an *nIn1\_a* wird mit dem Wert von *nIn2\_a* addiert.
2. Vom errechneten Ergebnis wird der Wert von *nIn3\_a* subtrahiert.
3. Anschließend wird das Ergebnis der Subtraktion auf  $\pm 32767$  begrenzt.



# Funktionsbibliothek LenzeDrive.lib

## Analoge Signalverarbeitung

### Eingangsverstärkung und Offset (L\_AIN)

#### 2.2.3 Eingangsverstärkung und Offset (L\_AIN)

Dieser FB wird vorzugsweise für die Nachschaltung an analoge Eingangsklemmen verwendet, um Offset und Verstärkung einzustellen.

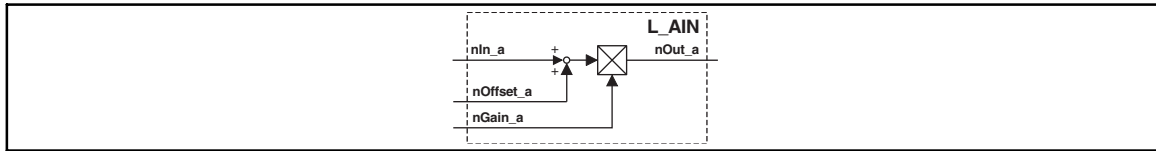


Abb. 2-4

Eingangsverstärkung und Offset (L\_AIN)

| VariableName | DataType | SignalType | VariableType | Note                            |
|--------------|----------|------------|--------------|---------------------------------|
| nIn_a        | Integer  | analog     | VAR_INPUT    | Eingangssignal                  |
| nOffset_a    | Integer  | analog     | VAR_INPUT    | Offset des Eingangssignals      |
| nGain_a      | Integer  | analog     | VAR_INPUT    | Verstärkung des Eingangssignals |
| nOut_a       | Integer  | analog     | VAR_OUTPUT   |                                 |

#### Funktion

- Offset
  - Der Wert an *nOffset\_a* wird zum Wert an *nIn\_a* addiert.
  - Das Ergebnis der Addition wird auf  $\pm 32767$  begrenzt.
- Verstärkung
  - Der begrenzte Wert (nach dem Offset) wird mit dem Wert an *nGain\_a* multipliziert.
  - Abschließend wird das Signal auf  $\pm 32767$  begrenzt.
- Das Signal wird an *nOut\_a* ausgegeben.

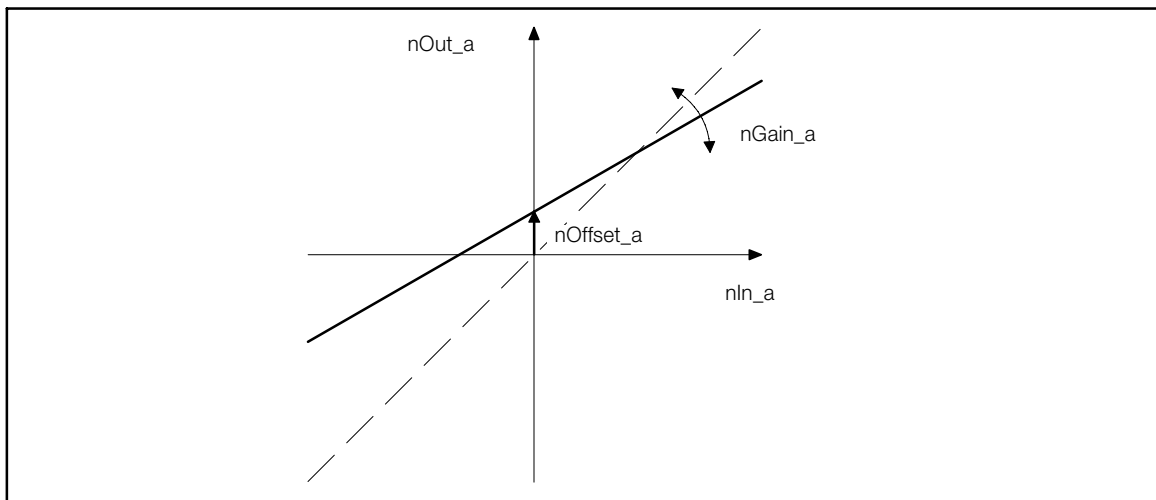
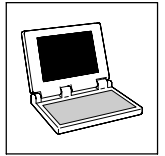


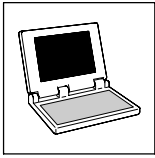
Abb. 2-5

Offset und Verstärkung des Analogeingangs



## Funktion in AWL

```
LD      nIn_a
INT_TO_DINT
ADD      (nOffset_a
INT_TO_DINT
)
LIMIT   -32767,32767
MUL      (nGain_a
INT_TO_DINT
)
DIV      16384
LIMIT   -32767,32767
DINT_TO_INT
ST      nOut_a
```



# Funktionsbibliothek LenzeDrive.lib

## Analoge Signalverarbeitung

### Invertierung (L\_ANEG)

#### 2.2.4 Invertierung (L\_ANEG)

Dieser FB invertiert das Vorzeichen eines Eingangswertes. Der Eingangswert wird mit -1 multipliziert und dann ausgegeben.

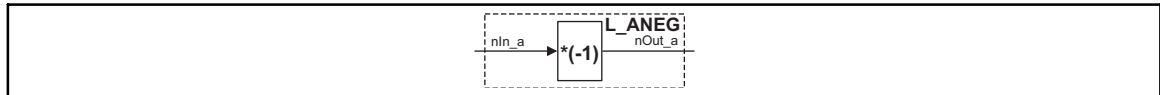
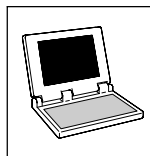


Abb. 2-6

Invertierung (L\_ANEG)

| VariableName | DataType | SignalType | VariableType | Note |
|--------------|----------|------------|--------------|------|
| nIn_a        | Integer  | analog     | VAR_INPUT    |      |
| nOut_a       | Integer  | analog     | VAR_OUTPUT   |      |



### 2.2.5 Ausgangsverstärkung und Offset (L\_AOUT)

Dieser FB wird vorzugsweise für die Vorschaltung an analoge Ausgangsklemmen verwendet, um Offset und Verstärkung einzustellen.

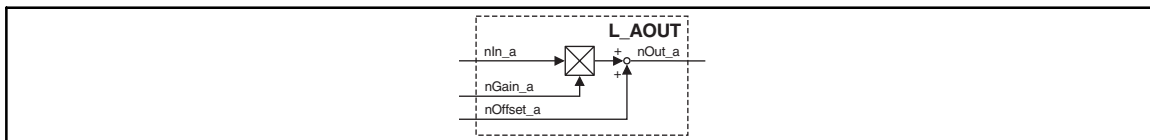


Abb. 2-7 Ausgangsverstärkung und Offset (L\_AOUT)

| VariableName | DataType | SignalType | VariableType | Note                            |
|--------------|----------|------------|--------------|---------------------------------|
| nIn_a        | Integer  | analog     | VAR_INPUT    | Eingangssignal                  |
| nGain_a      | Integer  | analog     | VAR_INPUT    | Verstärkung des Eingangssignals |
| nOffset_a    | Integer  | analog     | VAR_INPUT    | Offset des Eingangssignals      |
| nOut_a       | Integer  | analog     | VAR_OUTPUT   |                                 |

#### Funktion

- Verstärkung
  - Der Wert an  $nIn\_a$  wird mit dem Wert an  $nGain\_a$  multipliziert.
  - Die Multiplikation erfolgt nach der Formel:
 
$$16384 \cdot 16384 \cdot 2^{-14} = 16384 \quad [100 \% \cdot 100 \% = 100 \%]$$
  - Das Ergebnis der Multiplikation wird auf  $\pm 2^{14}$  begrenzt.
- Offset
  - Der begrenzte Wert (nach der Verstärkung) wird zum Wert an  $nOffset\_a$  addiert.
  - Das Ergebnis der Addition wird auf  $\pm 2^{14}$  begrenzt.
- Abschließend wird das Signal auf  $\pm 2^{14}$  begrenzt und an  $nOut\_a$  ausgegeben.

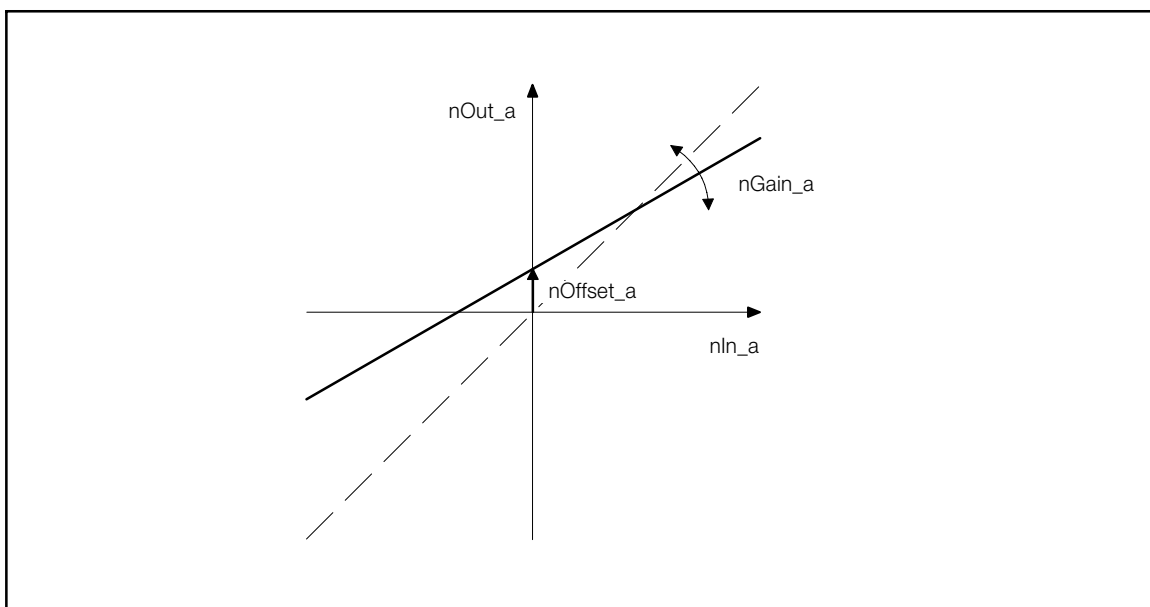
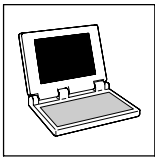


Abb. 2-8 Offset und Verstärkung des Analogausgangs



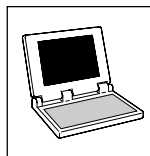
## **Funktionsbibliothek LenzeDrive.lib**

### **Analoge Signalverarbeitung**

#### **Ausgangsverstärkung und Offset (L\_AOUT)**

##### **Funktion in AWL**

```
LD      nIn_a
INT_TO_DINT
MUL      (nGain_a
INT_TO_DINT
)
DIV      -32767,32767
ADD      (nOffset_a
INT_TO_DINT
)
LIMIT    -32767,32767
DINT_TO_INT
ST      nOut_a
```



### 2.2.6 Arithmetik (L\_ARIT)

Dieser FB kann zwei analoge Signale arithmetisch miteinander verknüpfen.

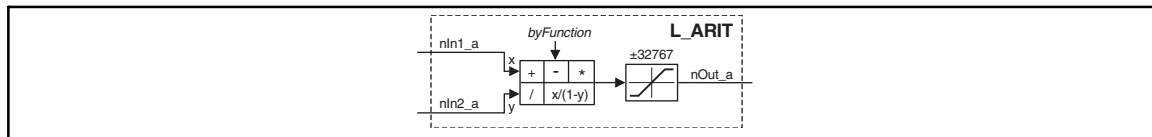


Abb. 2-9

Arithmetik (L\_ARIT)

| VariableName | DataType | SignalType | VariableType        | Note                             |
|--------------|----------|------------|---------------------|----------------------------------|
| nIn1_a       | Integer  | analog     | VAR_INPUT           |                                  |
| nIn2_a       | Integer  | analog     | VAR_INPUT           |                                  |
| nOut_a       | Integer  | analog     | VAR_OUTPUT          | Signal wird begrenzt auf ±32767. |
| byFunction   | Byte     |            | VAR CONSTANT RETAIN | Auswahl der Funktion             |

#### Parameter-Codestellen der Instanzen

| VariableName | L_ARIT1 | L_ARIT2 | SettingRange | Lenze |
|--------------|---------|---------|--------------|-------|
| byFunction   | C0338   | C0600   | 0 ... 5      | 1     |

#### Funktion

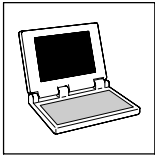
| Auswahl der Funktion | Arithmetische Funktion                                  | Hinweise                                         |
|----------------------|---------------------------------------------------------|--------------------------------------------------|
| byFunction = 0       | $nOut\_a = nIn1\_a$                                     |                                                  |
| byFunction = 1       | $nOut\_a = nIn1\_a + nIn2\_a$                           |                                                  |
| byFunction = 2       | $nOut\_a = nIn1\_a - nIn2\_a$                           |                                                  |
| byFunction = 3       | $nOut\_a = \frac{(nIn1\_a) \cdot (nIn2\_a)}{16384}$     |                                                  |
| byFunction = 4       | $nOut\_a = \frac{nIn1\_a}{nIn2\_a} \cdot 164$           | Ist der Nenner = 0, wird der Nenner = 1 gesetzt. |
| byFunction = 5       | $nOut\_a = \frac{nIn1\_a}{16384 - nIn2\_a} \cdot 16384$ |                                                  |

#### Funktion in ST

```

CASE byFunction OF
0:  nOut_a:=nIn1_a;
1:  nOut_a:=DINT_TO_INT(LIMIT(-32767, (INT_TO_DINT(nIn1_a)+INT_TO_DINT(nIn2_a)),32767));
2:  nOut_a:=DINT_TO_INT(LIMIT(-32767, (INT_TO_DINT(nIn1_a)-INT_TO_DINT(nIn2_a)),32767));
3:  nOut_a:=DINT_TO_INT(LIMIT(-32767, ((INT_TO_DINT(nIn1_a)*INT_TO_DINT(nIn2_a))/16384),32767));
4:  IF (nIn2_a=0) THEN
      nOut_a:=DINT_TO_INT(LIMIT(-32767, ((INT_TO_DINT(nIn1_a)*164)),32767));
    ELSE
      nOut_a:=DINT_TO_INT(LIMIT(-32767, ((INT_TO_DINT(nIn1_a)*164)/ABS(nIn2_a)),32767));
    END_IF
5:  IF (16384-INT_TO_DINT(nIn2_a)=0) THEN
      nOut_a:=DINT_TO_INT(LIMIT(-32767, ((INT_TO_DINT(nIn1_a)*16384),32767));
    ELSEIF (16384-INT_TO_DINT(nIn2_a)>32767) THEN
      nOut_a:=DINT_TO_INT(LIMIT(-32767, ((INT_TO_DINT(nIn1_a)*16384/32767),32767));
    ELSEIF (16384-INT_TO_DINT(nIn2_a)<-32767) THEN
      nOut_a:=DINT_TO_INT(LIMIT(-32767, ((INT_TO_DINT(nIn1_a)*16384/-32767),32767));
    ELSE
      nOut_a:=DINT_TO_INT(LIMIT(-32767, ((INT_TO_DINT(nIn1_a)*16384)/(16384-INT_TO_DINT(nIn2_a))),32767));
    END_IF
END_CASE;

```



# Funktionsbibliothek LenzeDrive.lib

## Analoge Signalverarbeitung Umschaltung (L\_ASW)

### 2.2.7 Umschaltung (L\_ASW)

Dieser FB schaltet zwischen zwei Integer-Werten um. Damit ist es z. B. möglich, beim Wickelvorgang zwischen einem Anfangsdurchmesser und einem errechneten Durchmesser umzuschalten.

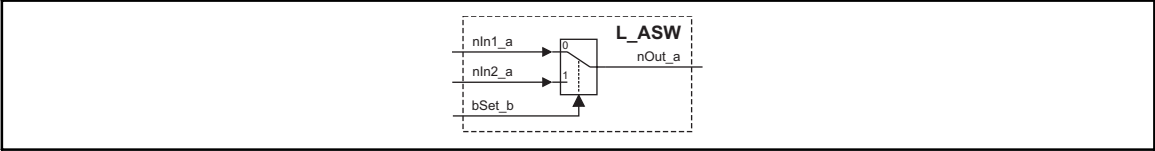
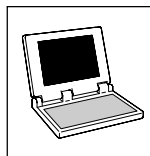


Abb. 2-10 Umschaltung (L\_ASW)

| VariableName | DataType | SignalType | VariableType | Note |
|--------------|----------|------------|--------------|------|
| nln1_a       | Integer  | analog     | VAR_INPUT    |      |
| nln2_a       | Integer  | analog     | VAR_INPUT    |      |
| bSet_b       | Bool     | binary     | VAR_INPUT    |      |
| nOut_a       | Integer  | analog     | VAR_OUTPUT   |      |

#### Funktion

| Steuersignal   | Ausgangssignal  |
|----------------|-----------------|
| bSet_b = TRUE  | nOut_a = nln2_a |
| bSet_b = FALSE | nOut_a = nln1_a |



### 2.2.8 Vergleich (L\_CMP)

Dieser FB vergleicht zwei Integer-Werte miteinander. Mit Vergleichen (Komparatoren) können Sie Schwellwertschalter realisieren.

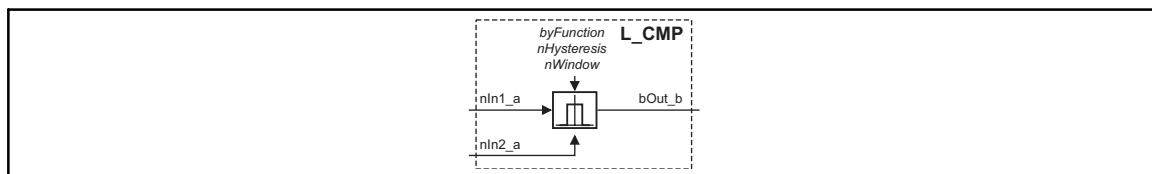


Abb. 2-11

Vergleich (L\_CMP)

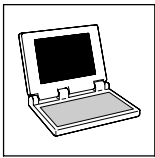
| VariableName | DataType | SignalType | VariableType        | Note                            |
|--------------|----------|------------|---------------------|---------------------------------|
| nln1_a       | Integer  | analog     | VAR_INPUT           |                                 |
| nln2_a       | Integer  | analog     | VAR_INPUT           |                                 |
| bOut_b       | Bool     | binary     | VAR_OUTPUT          |                                 |
| byFunction   | Byte     |            | VAR CONSTANT RETAIN | Vergleichsfunktion der Eingänge |
| nHysteresis  | Integer  |            | VAR CONSTANT RETAIN | Hysteresefunktion               |
| nWindow      | Integer  |            | VAR CONSTANT RETAIN | Fensterfunktion                 |

#### Parameter-Codestellen der Instanzen

| VariableName | L_CMP1 | L_CMP2 | L_CMP3 | SettingRange      | Lenze |
|--------------|--------|--------|--------|-------------------|-------|
| byFunction   | C0680  | C0685  | C0690  | 1 ... 6           | 6     |
| nHysteresis  | C0681  | C0686  | C0691  | 0.00 ... 100.00 % | 1.00  |
| nWindow      | C0682  | C0687  | C0692  | 0.00 ... 100.00 % | 1.00  |

#### Funktion

| Auswahl der Funktion | Vergleichsfunktion      |
|----------------------|-------------------------|
| byFunction = 1       | $nln1\_a = nln2\_a$     |
| byFunction = 2       | $nln1\_a > nln2\_a$     |
| byFunction = 3       | $nln1\_a < nln2\_a$     |
| byFunction = 4       | $ nln1\_a  =  nln2\_a $ |
| byFunction = 5       | $ nln1\_a  >  nln2\_a $ |
| byFunction = 6       | $ nln1\_a  <  nln2\_a $ |



# Funktionsbibliothek LenzeDrive.lib

## Analoge Signalverarbeitung

### Vergleich (L\_CMP)

#### 2.2.8.1 Funktion 1: $nIn1\_a = nIn2\_a$

- Auswahl: *byFunction* = 1
- Diese Funktion vergleicht zwei Signale auf Gleichheit. Sie können z. B. den Vergleich "Istdrehzahl ist gleich Solldrehzahl" ( $n_{ist} = n_{soll}$ ) realisieren.
- Entnehmen Sie die genaue Funktion dem Liniendiagramm. (Abb. 2-12)

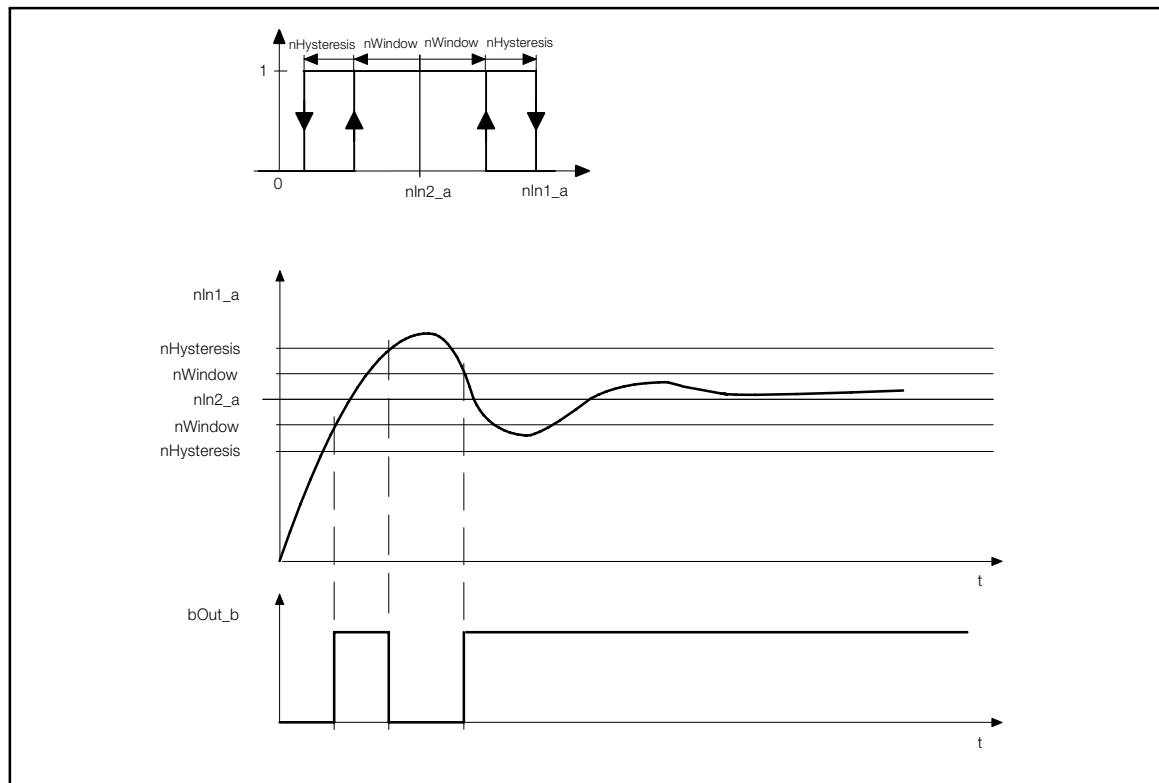


Abb. 2-12

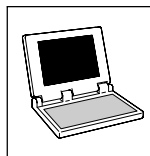
Gleichheit von Signalen ( $nIn1\_a = nIn2\_a$ )

- Mit *nWindow* stellen Sie das Fenster ein, in dem die Gleichheit gelten soll.
- Mit *nHysteresis* stellen Sie eine Hysterese ein, falls die Eingangssignale nicht stabil sind und der Ausgang somit oszilliert.



#### Tipp!

Bei dieser Funktion müssen Sie den FB in einer schnellen Task verwenden, um eine optimale Abtastung der Signale zu erreichen.



## 2.2.8.2 Funktion 2: $nln1\_a > nln2\_a$

- Auswahl: *byFunction* = 2
- Mit dieser Funktion können Sie z. B. den Vergleich "Istdrehzahl ist größer als ein Grenzwert" ( $n_{ist} > n_x$ ) für eine Drehrichtung realisieren.

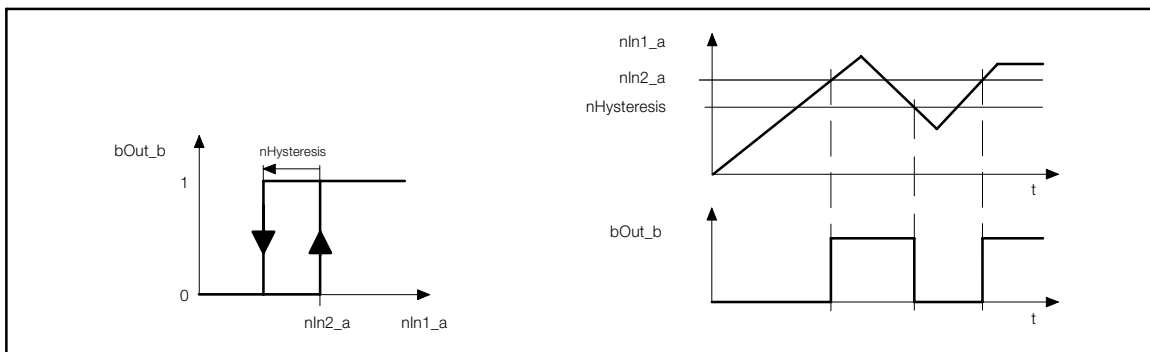


Abb. 2-13

Überschreiten von Signalwerten ( $nln1\_a > nln2\_a$ )

### Funktionsablauf

1. Überschreitet der Wert an  $nln1\_a$  den Wert an  $nln2\_a$ , wechselt  $bOut\_b$  von FALSE auf TRUE.
2. Erst wenn das Signal an  $nln1\_a$  den Wert von  $nln2\_a - nHysteresis$  wieder unterschreitet, wechselt  $bOut\_b$  von TRUE auf FALSE.

## 2.2.8.3 Funktion 3: $nln1\_a < nln2\_a$

- Auswahl: *byFunction* = 3
- Mit dieser Funktion können Sie z. B. den Vergleich "Istdrehzahl ist kleiner als ein Grenzwert" ( $n_{ist} < n_x$ ) für eine Drehrichtung realisieren.

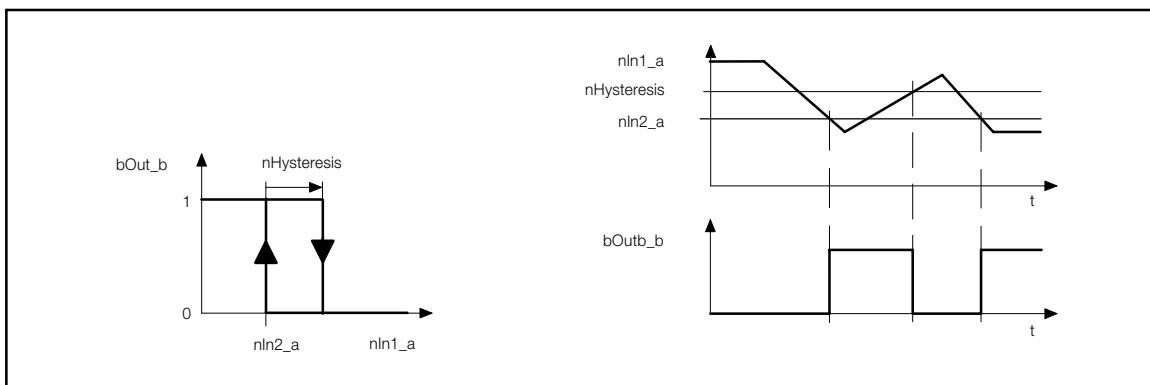
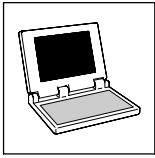


Abb. 2-14

Unterschreiten von Signalwerten ( $nln1\_a < nln2\_a$ )

### Funktionsablauf

1. Unterschreitet der Wert an  $nln1\_a$  den Wert an  $nln2\_a$ , wechselt  $bOut\_b$  von FALSE auf TRUE.
2. Erst wenn das Signal an  $nln1\_a$  den Wert von  $nln2\_a - nHysteresis$  wieder überschreitet, wechselt  $bOut\_b$  von TRUE auf FALSE.



## **Funktionsbibliothek LenzeDrive.lib**

### **Analoge Signalverarbeitung**

#### **Vergleich (L\_CMP)**

#### **2.2.8.4 Funktion 4: $|\text{nl}n1\_a| = |\text{nl}n2\_a|$**

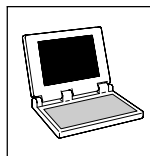
- Auswahl: *byFunction* = 4
- Mit dieser Funktion können Sie z. B. den Vergleich " $n_{\text{ist}} = 0$ " realisieren.
- Diese Funktion verhält sich genau wie Funktion 1. (📖 2-14)
  - Jedoch wird vor der Signalverarbeitung der Betrag von den Eingangssignalen gebildet (ohne Vorzeichen).

#### **2.2.8.5 Funktion 5: $|\text{nl}n1\_a| > |\text{nl}n2\_a|$**

- Auswahl: *byFunction* = 5
- Mit dieser Funktion können Sie z. B. den Vergleich " $|n_{\text{ist}}| > |n_x|$ " unabhängig von der Drehrichtung realisieren.
- Diese Funktion verhält sich genau wie Funktion 2. (📖 2-15)
  - Jedoch wird vor der Signalverarbeitung der Betrag von den Eingangssignalen gebildet (ohne Vorzeichen).

#### **2.2.8.6 Funktion 6: $|\text{nl}n1\_a| < |\text{nl}n2\_a|$**

- Auswahl: *byFunction* = 6
- Mit dieser Funktion können Sie den Vergleich " $|n_{\text{ist}}| < |n_x|$ " unabhängig von der Drehrichtung realisieren.
- Diese Funktion verhält sich genau wie Funktion 3. (📖 2-15)
  - Jedoch wird vor der Signalverarbeitung der Betrag von den Eingangssignalen gebildet (ohne Vorzeichen).



### 2.2.9 Kennlinienfunktion (L\_CURVE)

Dieser FB überführt ein analoges Signal in eine Kennlinie.

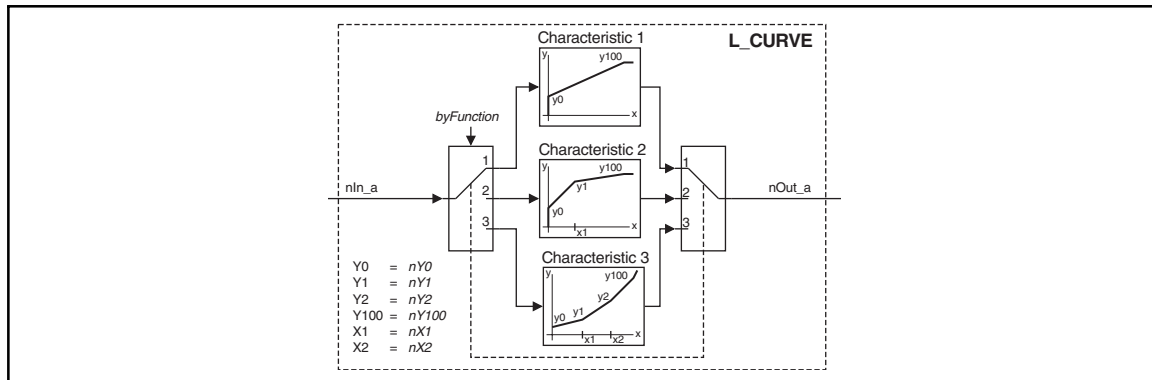


Abb. 2-15

Kennlinienfunktion (L\_CURVE)

| VariableName | DataType | SignalType | VariableType        | Note                                      |
|--------------|----------|------------|---------------------|-------------------------------------------|
| nIn_a        | Integer  | analog     | VAR_INPUT           |                                           |
| nOut_a       | Integer  | analog     | VAR_OUTPUT          |                                           |
| byFunction   | Byte     |            | VAR CONSTANT RETAIN | Auswahl der Kennlinienfunktion            |
| nY0          | Integer  |            | VAR CONSTANT RETAIN | Eingabe von Y0 vom Vektor (0, Y0)         |
| nY1          | Integer  |            | VAR CONSTANT RETAIN | Eingabe von Y1 vom Vektor (X1, Y1)        |
| nY2          | Integer  |            | VAR CONSTANT RETAIN | Eingabe von Y2 vom Vektor (X2, Y2)        |
| nY100        | Integer  |            | VAR CONSTANT RETAIN | Eingabe von Y100 vom Vektor (16384, Y100) |
| nX1          | Integer  |            | VAR CONSTANT RETAIN | Eingabe von X1 vom Vektor (X1, Y1)        |
| nX2          | Integer  |            | VAR CONSTANT RETAIN | Eingabe von X2 vom Vektor (X2, Y2)        |

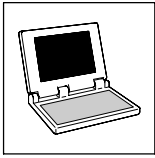
#### Parameter-Codestellen der Instanzen

| VariableName | L_CURVE1 |  | SettingRange     | Lenze  |
|--------------|----------|--|------------------|--------|
| byFunction   | C0960    |  | 1 ... 3          | 1      |
| nY0          | C0961    |  | 0 ... 199.99 %   | 0.00   |
| nY1          | C0962    |  | 0 ... 199.99 %   | 50.00  |
| nY2          | C0963    |  | 0 ... 199.99 %   | 75.00  |
| nY100        | C0964    |  | 0 ... 199.99 %   | 100.00 |
| nX1          | C0965    |  | 0.01 ... 99.99 % | 50.00  |
| nX2          | C0966    |  | 0.01 ... 99.99 % | 75.00  |

#### Funktion

| Auswahl der Funktion | Kennlinienfunktion           | Informationen zur Eingabe der Stützstellen |
|----------------------|------------------------------|--------------------------------------------|
| byFunction = 1       | Kennlinie mit 2 Stützstellen | Abb. 2-16                                  |
| byFunction = 2       | Kennlinie mit 3 Stützstellen | Abb. 2-17                                  |
| byFunction = 3       | Kennlinie mit 4 Stützstellen | Abb. 2-18                                  |

- 100 % entspricht 16384.
- Zwischen den Stützstellen wird linear interpoliert.
- Für negative Werte an *nIn\_a* werden die Einstellwerte der Stützstellen invertiert verarbeitet (siehe Liniendiagramme).
  - Wenn dies nicht gewünscht ist, schalten Sie einen FB **L\_ABS** oder einen FB **L\_LIM** vor oder hinter den FB **L\_CURVE**.



# Funktionsbibliothek LenzeDrive.lib

## Analoge Signalverarbeitung

### Kennlinienfunktion (L\_CURVE)

#### 2.2.9.1 Kennlinie mit 2 Stützstellen

byFunction = 1

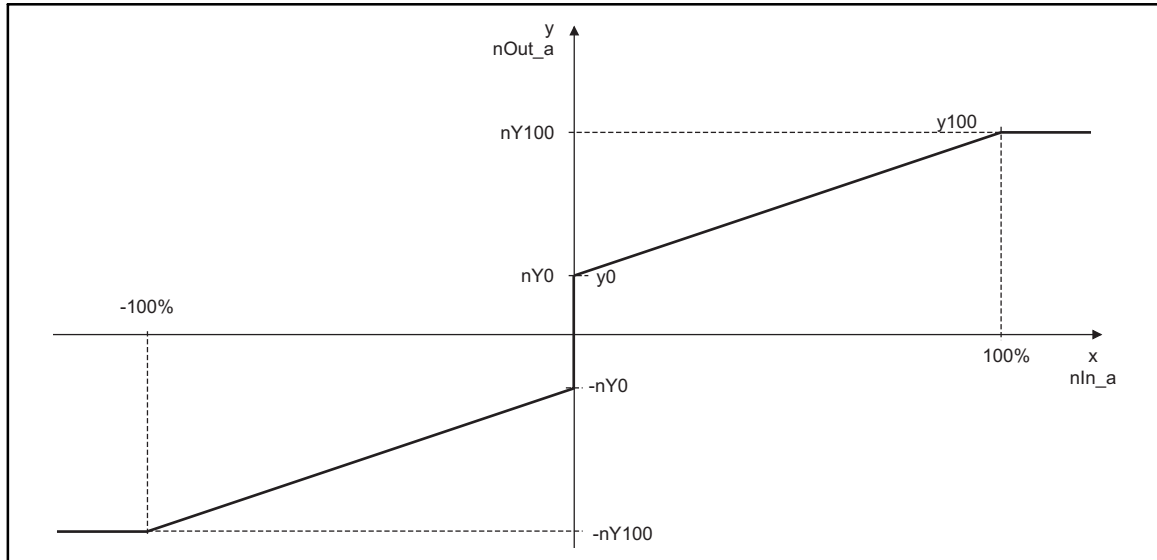


Abb. 2-16 Liniendiagramm Kennlinie mit 2 Stützstellen

#### 2.2.9.2 Kennlinie mit 3 Stützstellen

byFunction = 2

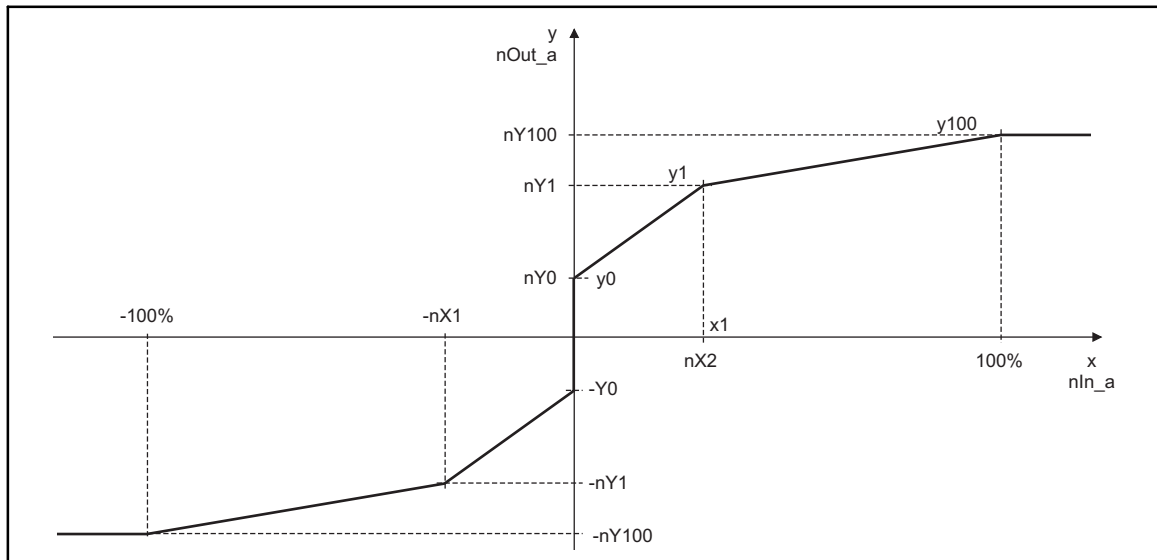
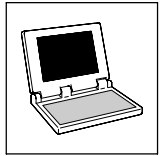


Abb. 2-17 Liniendiagramm Kennlinie mit 3 Stützstellen



## 2.2.9.3 Kennlinie mit 4 Stützstellen

byFunction = 3

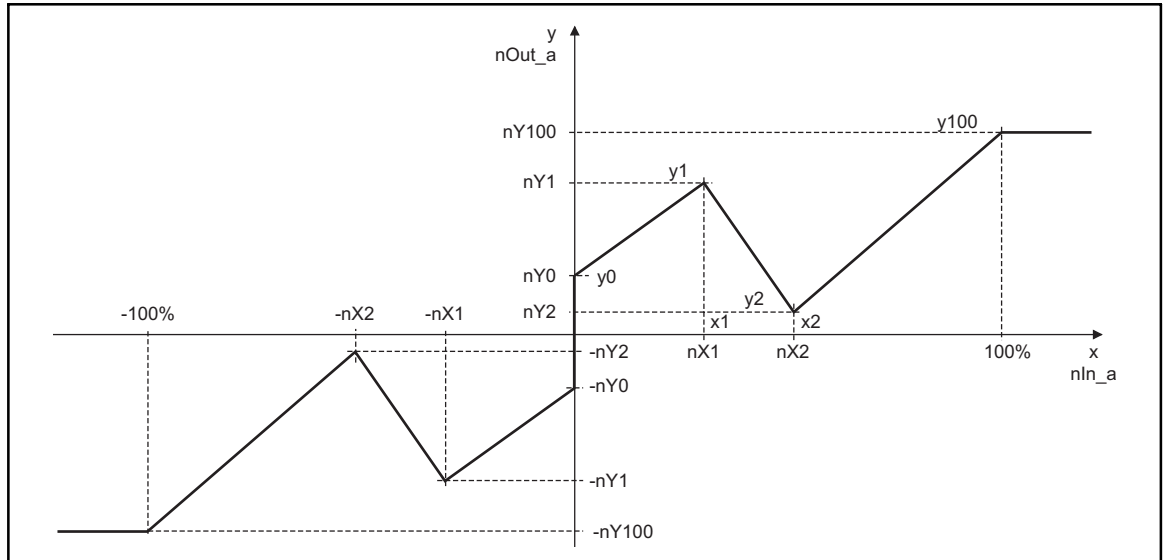
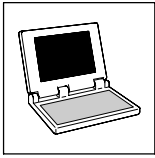


Abb. 2-18

Liniendiagramm Kennlinie mit 4 Stützstellen



# Funktionsbibliothek LenzeDrive.lib

## Analoge Signalverarbeitung

### Totgang (L\_DB)

#### 2.2.10

#### Totgang (L\_DB)

Dieser FB setzt störende Einflüsse um den Nullpunkt auf Null (z. B. Störgrößen auf analogen Eingangsspannungen).

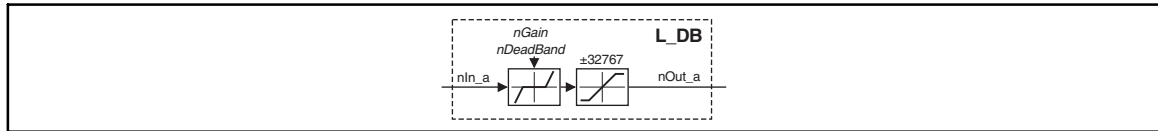


Abb. 2-19

Totgang (L\_DB)

| VariableName | DataType | SignalType | VariableType        | Note                                   |
|--------------|----------|------------|---------------------|----------------------------------------|
| nIn_a        | Integer  | analog     | VAR_INPUT           |                                        |
| nOut_a       | Integer  | analog     | VAR_OUTPUT          | Signal wird begrenzt auf $\pm 32767$ . |
| nGain        | Integer  |            | VAR CONSTANT RETAIN | Verstärkung                            |
| nDeadBand    | Integer  |            | VAR CONSTANT RETAIN | Totgang                                |

#### Parameter-Codestellen der Instanzen

| VariableName | L_DB1 |  |  | SettingRange     | Lenze |
|--------------|-------|--|--|------------------|-------|
| nGain        | C0620 |  |  | -10.00 ... 10.00 | 1.00  |
| nDeadBand    | C0621 |  |  | 0 ... 100.00 %   | 1.00  |

#### Funktion

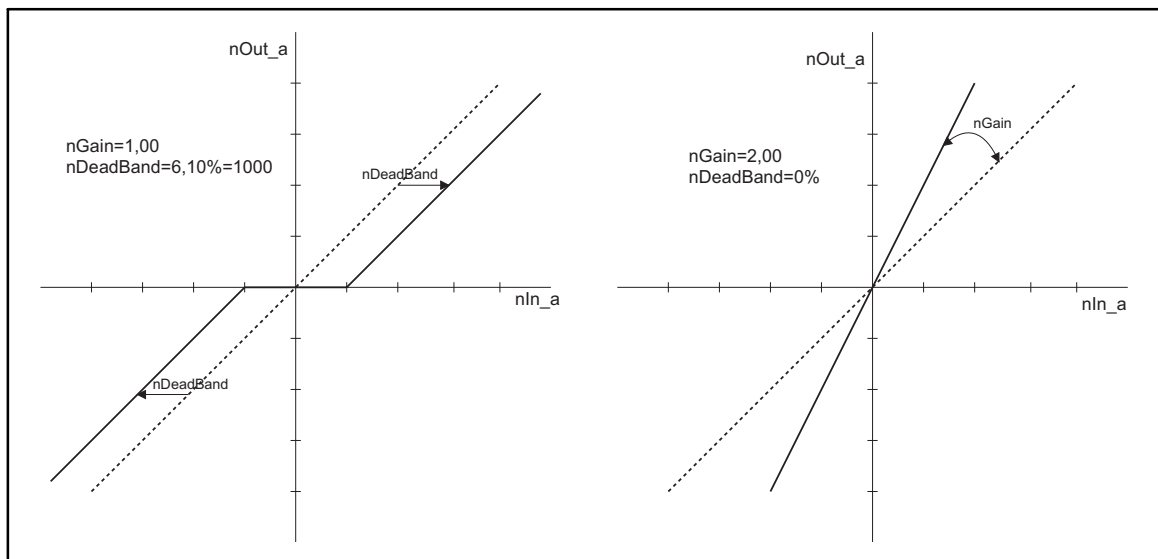
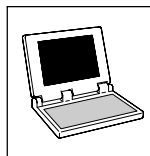


Abb. 2-20

Totgang und Verstärkung

- Über *nDeadBand* parametrieren Sie den Totgang.
- Über *nGain* verändern Sie die Verstärkung.
- 100 % entspricht 16384.



### 2.2.11 Differenzierung (L\_DT1\_)

Dieser FB differenziert Signale. Sie können ihn z. B. für die Beschleunigungsaufschaltung ( $dv/dt$ ) verwenden.

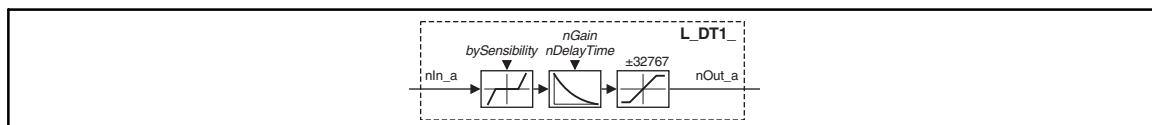


Abb. 2-21

Differenzierung (L\_DT1\_)

| VariableName  | DataType | SignalType | VariableType        | Note                                                                                                                              |
|---------------|----------|------------|---------------------|-----------------------------------------------------------------------------------------------------------------------------------|
| nIn_a         | Integer  | analog     | VAR_INPUT           |                                                                                                                                   |
| nOut_a        | Integer  | analog     | VAR_OUTPUT          | Signal wird begrenzt auf $\pm 32767$ .                                                                                            |
| nGain         | Integer  |            | VAR CONSTANT RETAIN | Verstärkung K                                                                                                                     |
| nDelayTime    | Integer  |            | VAR CONSTANT RETAIN | Verzögerungszeit $T_v$                                                                                                            |
| bySensibility | Byte     |            | VAR CONSTANT RETAIN | Eingangsempfindlichkeit von <i>nIn_a</i> . Entsprechend der Einstellung wertet der FB nur die angegebenen höherwertigen Bits aus. |

#### Parameter-Codestellen der Instanzen

| VariableName  | L_DT1_1 |  |  | SettingRange       | Lenze |
|---------------|---------|--|--|--------------------|-------|
| nGain         | C0650   |  |  | -320.00 ... 320.00 | 1.00  |
| nDelayTime    | C0651   |  |  | 0.005 ... 5.000 s  | 1.000 |
| bySensibility | C0653   |  |  | 1 ... 7            | 1     |

#### Funktion

| Auswahl der Funktion | Funktion |
|----------------------|----------|
| bySensibility = 1    | 15 Bit   |
| bySensibility = 2    | 14 Bit   |
| bySensibility = 3    | 13 Bit   |
| bySensibility = 4    | 12 Bit   |
| bySensibility = 5    | 11 Bit   |
| bySensibility = 6    | 10 Bit   |
| bySensibility = 7    | 9 Bit    |

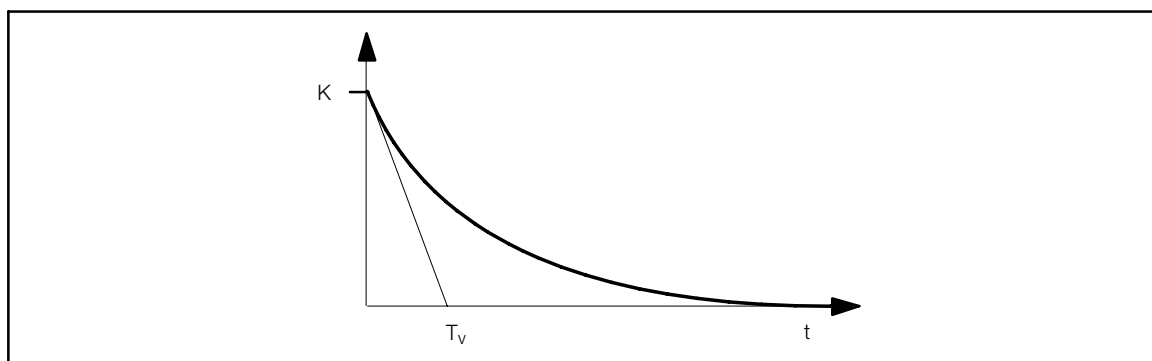
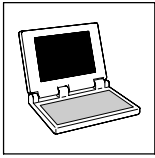


Abb. 2-22

Verzögerungszeit  $T_v$  des Differenziergliedes 1. Ordnung



# Funktionsbibliothek LenzeDrive.lib

## Analoge Signalverarbeitung

### Begrenzung (L\_LIM)

#### 2.2.12

#### Begrenzung (L\_LIM)

Dieser FB begrenzt Signale auf einstellbare Wertebereiche. Mit einer oberen und einer unteren Grenze legen Sie den Wertebereich fest.

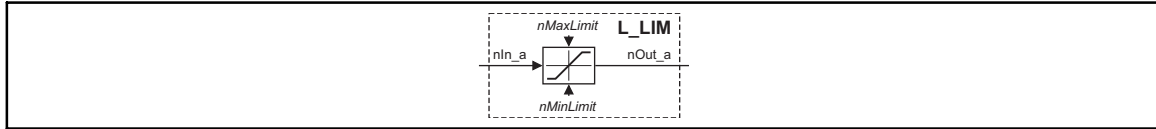


Abb. 2-23

Begrenzung (L\_LIM)

| VariableName | DataType | SignalType      | VariableType        | Note                                                |
|--------------|----------|-----------------|---------------------|-----------------------------------------------------|
| nIn_a        | Integer  | analog/velocity | VAR_INPUT           |                                                     |
| nOut_a       | Integer  | analog/velocity | VAR_OUTPUT          |                                                     |
| nMaxLimit    | Integer  |                 | VAR CONSTANT RETAIN | Definiert die obere Grenze. (100 % $\equiv$ 16384)  |
| nMinLimit    | Integer  |                 | VAR CONSTANT RETAIN | Definiert die untere Grenze. (100 % $\equiv$ 16384) |

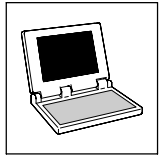
#### Parameter-Codestellen der Instanzen

| VariableName | L_LIM1 |  |  | SettingRange         | Lenze   |
|--------------|--------|--|--|----------------------|---------|
| nMaxLimit    | C0630  |  |  | -199.99 ... 199.99 % | 100.00  |
| nMinLimit    | C0631  |  |  | -199.99 ... 199.99 % | -100.00 |



#### Tipp!

Die untere Grenze muss immer kleiner eingestellt sein, als die obere Grenze. Andernfalls wird *nOut\_a* = 0 gesetzt.



### 2.2.13 Verzögerung (L\_PT1\_)

Dieser FB filtert und verzögert analoge Signale.

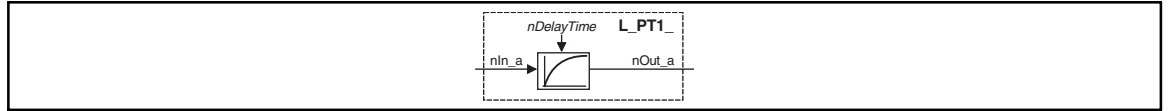


Abb. 2-24

Verzögerung (L\_PT1\_)

| VariableName | DataType | SignalType | VariableType        | Note          |
|--------------|----------|------------|---------------------|---------------|
| nIn_a        | Integer  | analog     | VAR_INPUT           |               |
| nOut_a       | Integer  | analog     | VAR_OUTPUT          |               |
| nDelayTime   | Integer  |            | VAR CONSTANT RETAIN | Zeitkonstante |

#### Parameter-Codestellen der Instanzen

| VariableName | L_PT1_1 |  |  | SettingRange   | Lenze |
|--------------|---------|--|--|----------------|-------|
| nDelayTime   | C0640   |  |  | 0.01 ... 50.00 | 20.00 |

#### Funktion

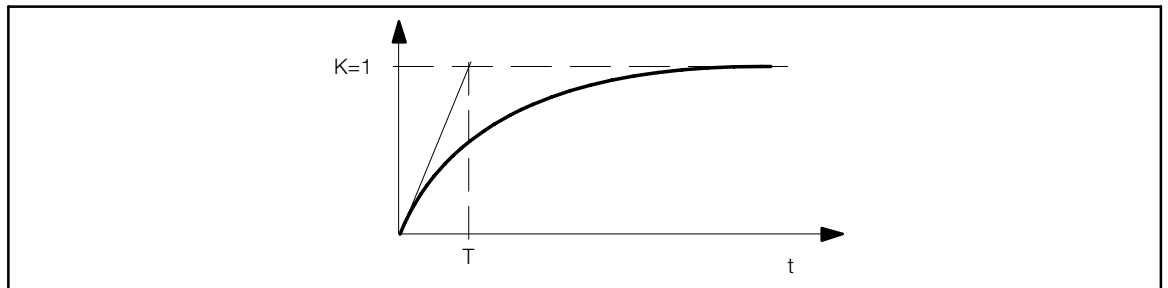
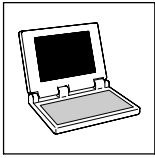


Abb. 2-25

Verzögerungszeit T des Verzögerungsglieds 1. Ordnung

- Mit *nDelayTime* wird die Verzögerungszeit T eingestellt.
- Der Proportionalbeiwert ist mit  $K = 1$  fest vorgegeben.



# Funktionsbibliothek LenzeDrive.lib

## Analoge Signalverarbeitung

### Hochlaufgeber (L\_RFG)

#### 2.2.14 Hochlaufgeber (L\_RFG)

Dieser FB begrenzt als Hochlaufgeber Signale in ihrem zeitlichen Anstieg.

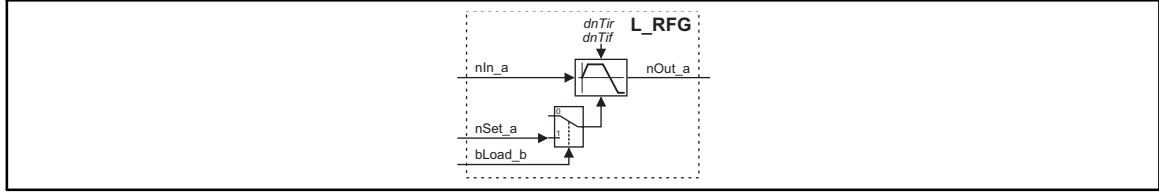


Abb. 2-26

Hochlaufgeber (L\_RFG)

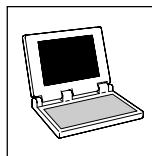
| VariableName | DataType       | SignalType      | VariableType        | Note                  |
|--------------|----------------|-----------------|---------------------|-----------------------|
| nIn_a        | Integer        | analog/velocity | VAR_INPUT           |                       |
| nSet_a       | Integer        | analog/velocity | VAR_INPUT           |                       |
| bLoad_b      | Bool           | binary          | VAR_INPUT           |                       |
| nOut_a       | Integer        | analog/velocity | VAR_OUTPUT          |                       |
| dnTir        | Double Integer |                 | VAR CONSTANT RETAIN | Hochlaufzeit $T_{ir}$ |
| dnTif        | Double Integer |                 | VAR CONSTANT RETAIN | Ablaufzeit $T_{if}$   |

#### Parameter-Codestellen der Instanzen

| Variablenname | L_RFG1 |  |  | SettingRange        | Lenze |
|---------------|--------|--|--|---------------------|-------|
| dnTir         | C0671  |  |  | 0.000 ... 999.999 s | 0.000 |
| dnTif         | C0672  |  |  | 0.000 ... 999.999 s | 0.000 |

#### Funktionsumfang

- Berechnen und Einstellen der Hoch- und Ablaufzeit
- Laden des Hochlaufgebers



### 2.2.14.1 Berechnen und Einstellen der Hoch- und Ablaufzeit

Die Hoch- und Ablaufzeit beziehen sich auf eine Änderung des Ausgangswertes von 0 auf 100 % (100 % = 16384). Die einzustellenden Zeiten  $T_{ir}$  und  $T_{if}$  können nach den Formeln in Abb. 2-27 berechnen:

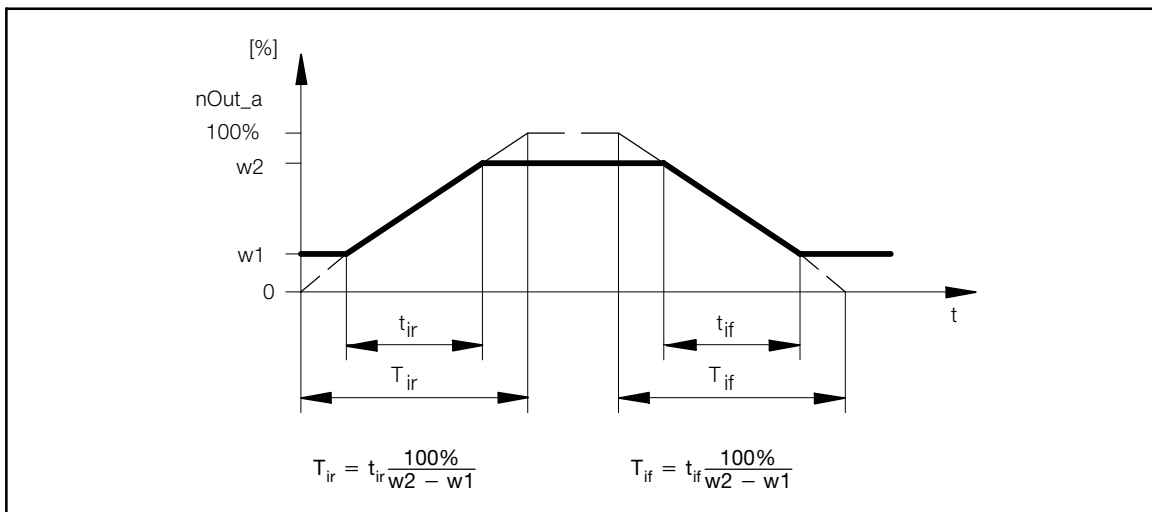


Abb. 2-27

Hoch- und Ablaufzeiten von L\_RFG

$w1, w2$  Wechsel der Hauptsollwerte in Abhängigkeit von  $t_{ir}$  bzw.  $t_{if}$

Hierbei sind  $t_{ir}$  und  $t_{if}$  die gewünschten Zeiten für den Wechsel zwischen  $w1$  und  $w2$ . Die errechneten Werte  $T_{ir}$  und  $T_{if}$  geben Sie dann unter  $dnTir$  und  $dnTif$  ein.

### 2.2.14.2 Laden des Hochlaufgebers

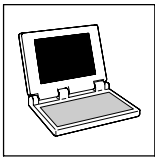
Über  $nSet\_a$  und  $bLoad\_b$  können Sie den Hochlaufgeber mit definierten Werten initialisieren.

- Solange  $bLoad\_b = TRUE$ , wird das Signal an  $nSet\_a$  an  $nOut\_a$  ausgegeben.
- Wird  $bLoad\_b = FALSE$  gesetzt, läuft der Hochlaufgeber mit den eingestellten  $T_i$ -Zeiten von dem geladenen Wert über  $nSet\_a$  zu dem Wert an  $nIn\_a$ .



#### Tipp!

$16384 \equiv 100 \% \equiv C0011$  ( $n_{max}$ )



# Funktionsbibliothek LenzeDrive.lib

## Analoge Signalverarbeitung

### Sample & Hold (L\_SH)

#### 2.2.15 Sample & Hold (L\_SH)

Dieser FB kann analoge Signale speichern. Der gespeicherte Wert ist auch nach dem Netzschalten verfügbar.

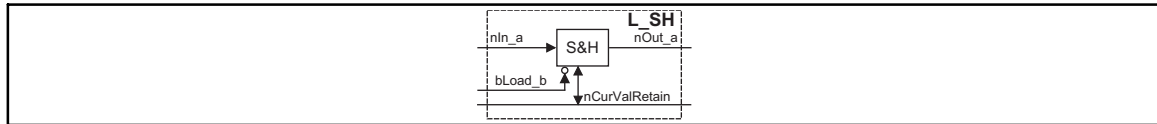


Abb. 2-28

Sample & Hold (L\_SH)

| VariableName  | DataType | SignalType      | VariableType      | Note              |
|---------------|----------|-----------------|-------------------|-------------------|
| nIn_a         | Integer  | analog/velocity | VAR_INPUT         |                   |
| bLoad_b       | Bool     | binary          | VAR_INPUT         | FALSE = speichern |
| nOut_a        | Integer  | analog/velocity | VAR_OUTPUT        |                   |
| nCurValRetain | Integer  |                 | VAR_GLOBAL RETAIN |                   |

#### Funktion

- Mit *bLoad\_b* = TRUE wird das Signal an *nIn\_a* auf *nOut\_a* geschaltet.
- Mit *bLoad\_b* = FALSE wird der zuletzt gültige Wert gespeichert und an *nOut\_a* ausgegeben. Eine Signaländerung an *nIn\_a* bewirkt keine Änderung an *nOut\_a*.
- Speichern beim Netzabschalten:
  - Setzen Sie *bLoad\_b* = FALSE, während sie die Versorgungsspannung abschalten (entweder Netzspannung, Zwischenkreisspannung oder Spannungsversorgung der Steuerklemmen).
  - Setzen Sie *bLoad\_b* = FALSE, während sie die Versorgungsspannung wieder einschalten (entweder Netzspannung, Zwischenkreisspannung oder Spannungsversorgung der Steuerklemmen).

#### Aktuellen Ausgangswert nach Netzausfall speichern

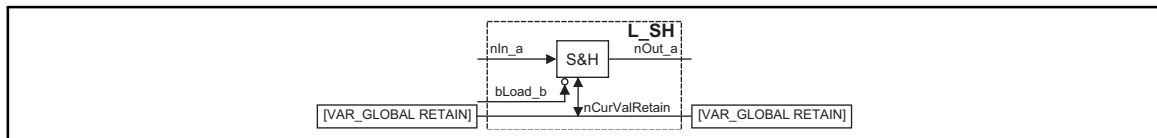
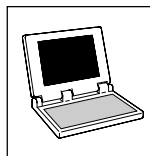


Abb. 2-29

Programmierung, um nach einem Netzausfall den aktuellen Ausgangswert zu speichern

Um nach einem Netzausfall den letzten Istwert an *nOut\_a* zu speichern, müssen Sie eine globale Variable vom Typ RETAIN (VAR\_GLOBAL RETAIN) deklarieren. Verknüpfen Sie diese Variable wie in Abb. 2-29 dargestellt.

- In dieser Variable wird immer der aktuelle Wert an *nOut\_a* gespeichert. Die Variable behält auch nach einem Netzausfall den Wert.
- Ist das Netz wieder eingeschaltet, wird der gespeicherte Wert aus der Variablen in den FB L\_SH eingelesen und als Startwert verarbeitet.



### 2.2.16 Hochlaufgeber S-Rampe (L\_SRFG)

Dieser FB bewertet einen Sollwert über eine S-Form ( $\sin^2$ -Form).

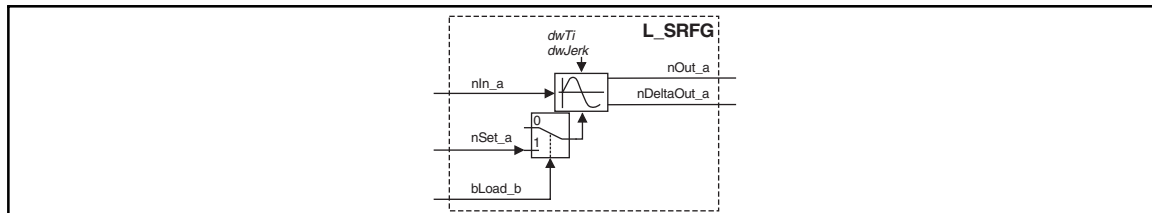


Abb. 2-30

Hochlaufgeber S-Rampe (L\_SRFG)

| VariableName | DataType      | SignalType | VariableType        | Note                                                                                                                                                        |
|--------------|---------------|------------|---------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------|
| nIn_a        | Integer       | analog     | VAR_INPUT           | Eingang                                                                                                                                                     |
| nSet_a       | Integer       | analog     | VAR_INPUT           | Startwert für den Hochlaufgeber. Die Übernahme erfolgt bei <i>bLoad_b</i> = TRUE.                                                                           |
| bLoad_b      | Bool          | binary     | VAR_INPUT           | TRUE = übernimmt den Wert an <i>nSet_a</i> und gibt diesen an <i>nOut_a</i> aus; <i>nDeltaOut_a</i> bleibt dabei auf 0 %.                                   |
| nOut_a       | Integer       | analog     | VAR_OUTPUT          | Signal wird begrenzt auf $\pm 100$ %. (100 % = 16384)                                                                                                       |
| nDeltaOut_a  | Integer       | analog     | VAR_OUTPUT          | <ul style="list-style-type: none"> <li>Gibt die Beschleunigung des Hochlaufgebers aus.</li> <li>Signal wird begrenzt auf <math>\pm 100</math> %.</li> </ul> |
| dwTi         | Unsigned Long |            | VAR CONSTANT RETAIN | Beschleunigung in [%] (100 % = 16384)                                                                                                                       |
| dwJerk       | Unsigned Long |            | VAR CONSTANT RETAIN | Ruck                                                                                                                                                        |

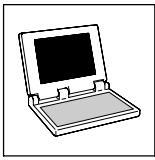
#### Parameter-Codestellen der Instanzen

| VariableName | L_SRFG1 |  |  | SettingRange         | Lenze   |
|--------------|---------|--|--|----------------------|---------|
| dwTi         | C1040   |  |  | 0.001 ... 5000.000 % | 100.000 |
| dwJerk       | C1041   |  |  | 0.001 ... 999.999 s  | 0.200   |



#### Tipp!

16384  $\equiv$  100 %  $\equiv$  C0011 ( $n_{\max}$ )



# Funktionsbibliothek LenzeDrive.lib

## Analoge Signalverarbeitung

### Hochlaufgeber S-Rampe (L\_SRFG)

#### Funktion

##### Hochlaufgeber laden

- Mit `bLoad_b = TRUE` wird der Hochlaufgeber mit dem Signal an `nSet_a` geladen.
- Dieser Wert wird sofort übernommen und an `nOut_a` ausgegeben. Es findet kein Hoch- oder Abflauen über eine S-Form statt.
- Solange `bLoad_b = TRUE` bleibt, ist der Hochlaufgeber gesperrt.

##### Beschleunigung und Ruck

Die maximale Beschleunigung und der Ruck sind separat einstellbar.

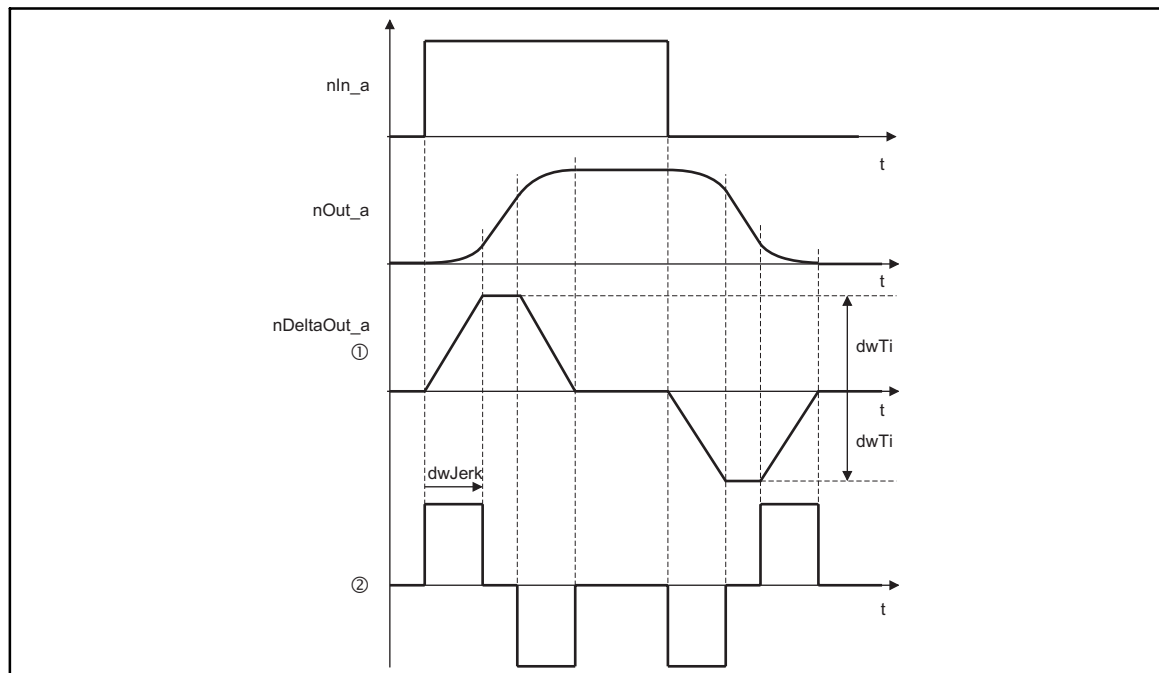
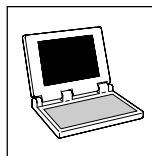


Abb. 2-31

Liniendiagramm

- ① Beschleunigung
- ② Ruck

- Max. Beschleunigung:
  - Mit  $dwTi$  stellen Sie die positive als auch die negative Beschleunigung ein.
  - Die Einstellung erfolgt nach Formel:
 
$$\frac{1 \text{ s} \cdot 100 \%}{dwTi}$$
- Ruck:
  - Mit  $dwJerk$  stellen Sie eine ruckfreie Beschleunigung des Antriebs ein.
  - Der Ruck wird in [s] vorgegeben, bis der Hochlaufgeber mit max. Beschleunigung arbeitet (siehe Abb. 2-31).



## 2.3 Digitale Signalverarbeitung

### 2.3.1 Logisches UND (L\_AND)

Dieser FB realisiert logische UND-Verknüpfungen binärer Signale. Sie können diese Verknüpfungen zur Steuerung von Funktionen oder zur Bildung von Statusinformationen verwenden.

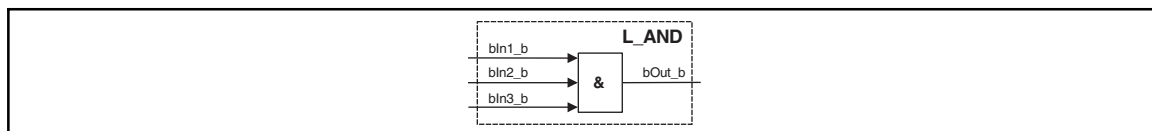


Abb. 2-32

Logisches UND (L\_AND)

| VariableName | DataType | SignalType | VariableType | Note |
|--------------|----------|------------|--------------|------|
| bln1_b       | Bool     | binary     | VAR_INPUT    |      |
| bln2_b       | Bool     | binary     | VAR_INPUT    |      |
| bln3_b       | Bool     | binary     | VAR_INPUT    |      |
| bOut_b       | Bool     | binary     | VAR_OUTPUT   |      |

#### Wahrheitstabelle

| bln1_b | bln2_b | bln3_b | bOut_b |
|--------|--------|--------|--------|
| 0      | 0      | 0      | 0      |
| 1      | 0      | 0      | 0      |
| 0      | 1      | 0      | 0      |
| 1      | 1      | 0      | 0      |
| 0      | 0      | 1      | 0      |
| 1      | 0      | 1      | 0      |
| 0      | 1      | 1      | 0      |
| 1      | 1      | 1      | 1      |

0 = FALSE

1 = TRUE

Die Funktion entspricht einer Reihenschaltung von Schließern bei einer Schützsteuerung.

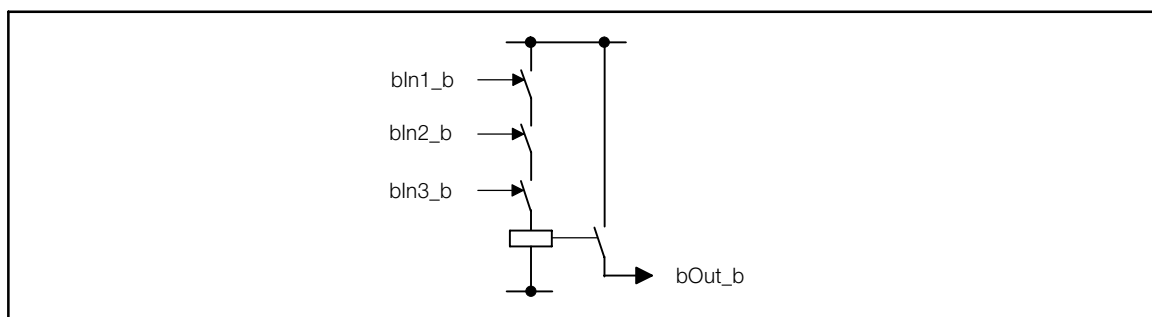


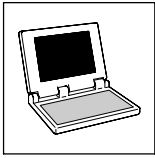
Abb. 2-33

Funktion der UND-Verknüpfung als Reihenschaltung von Schließern



#### Tipp!

Verwenden Sie die Eingänge *bln1\_b* und *bln2\_b*, wenn Sie nur zwei Eingänge benötigen. Legen Sie Eingang *bln3\_b* auf TRUE.



# Funktionsbibliothek LenzeDrive.lib

## Digitale Signalverarbeitung

### Verzögerung (L\_DIGDEL)

## 2.3.2 Verzögerung (L\_DIGDEL)

Dieser FB verzögert zeitlich binäre Signale.

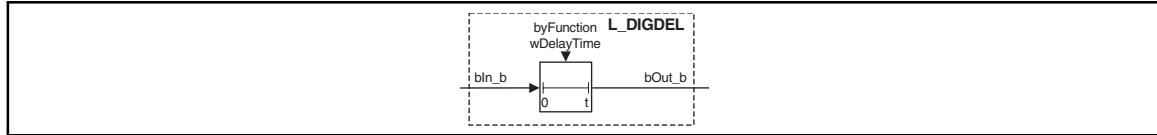


Abb. 2-34

Verzögerungsglied (L\_DIGDEL)

| VariableName | DataType | SignalType | VariableType        | Note                 |
|--------------|----------|------------|---------------------|----------------------|
| bln_b        | Bool     | binary     | VAR_INPUT           |                      |
| bOut_b       | Bool     | binary     | VAR_OUTPUT          |                      |
| byFunction   | Byte     |            | VAR CONSTANT RETAIN | Auswahl der Funktion |
| wDelayTime   | Word     |            | VAR CONSTANT RETAIN | Verzögerungszeit     |

### Parameter-Codestellen der Instanzen

| VariableName | L_DIGDEL1 | L_DIGDEL2 | SettingRange       | Lenze |
|--------------|-----------|-----------|--------------------|-------|
| byFunction   | C0720     |           | 0 ... 2            | 2     |
|              |           | C0725     |                    | 0     |
| wDelayTime   | C0721     | C0726     | 0.001 ... 60.000 s | 1.000 |

### Funktionsumfang

- Anzugsverzögerung
- Abfallverzögerung
- Allgemeine Verzögerung

## 2.3.2.1 Anzugsverzögerung

*byFunction* = 0

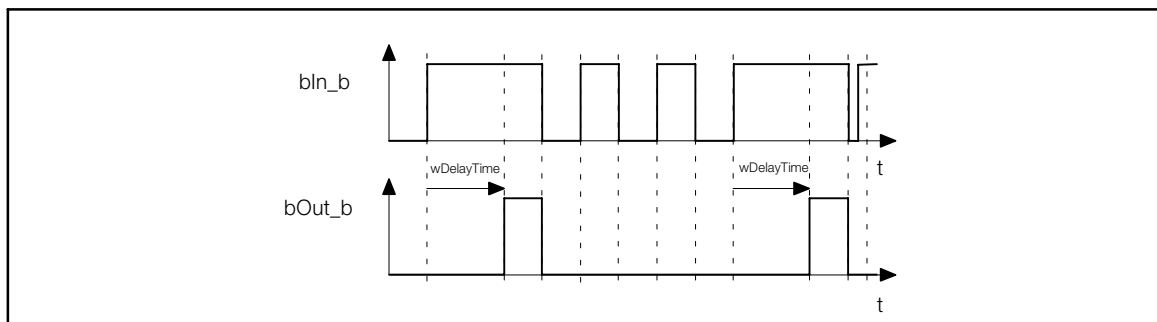


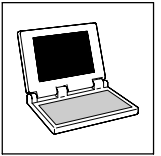
Abb. 2-35

Anzugsverzögerung

Der FB L\_DIGDEL arbeitet bei dieser Funktion wie ein nachtriggerbares Monoflop.

### Funktionsablauf

1. Eine FALSE-TRUE-Flanke an *nIn\_b* startet das Zeitglied.
2. Ist die Verzögerungszeit abgelaufen, die mit *wDelayTime* eingestellt wurde, schaltet *bOut\_b* = TRUE.
3. Eine TRUE-FALSE-Flanke an *nIn\_b* setzt das Zeitglied zurück und schaltet *bOut\_b* = FALSE.



## 2.3.2.2 Abfallverzögerung

byFunction = 1

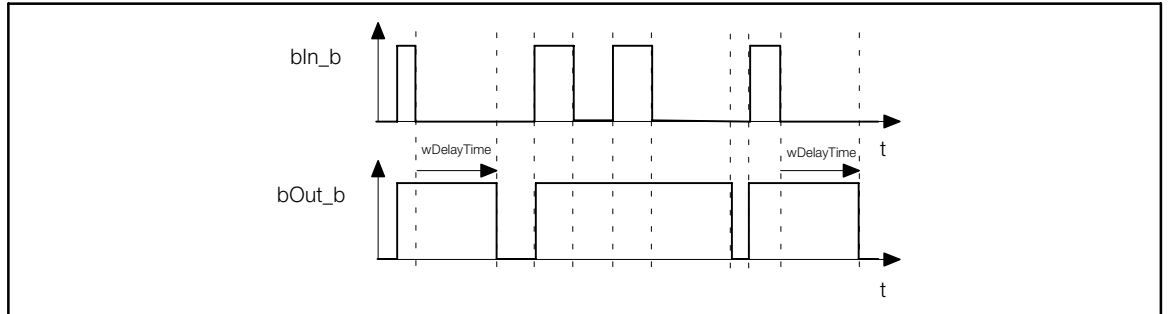


Abb. 2-36

Abfallverzögerung

### Funktionsablauf

1. Eine FALSE-TRUE-Flanke an  $nIn\_b$  schaltet  $bOut\_b = TRUE$  und setzt das Zeitglied zurück.
2. Eine TRUE-FALSE-Flanke an  $nIn\_b$  startet das Zeitglied.
3. Ist die Verzögerungszeit abgelaufen, die mit  $wDelayTime$  eingestellt wurde, schaltet  $bOut\_b = FALSE$ .

## 2.3.2.3 Allgemeine Verzögerung

byFunction = 2

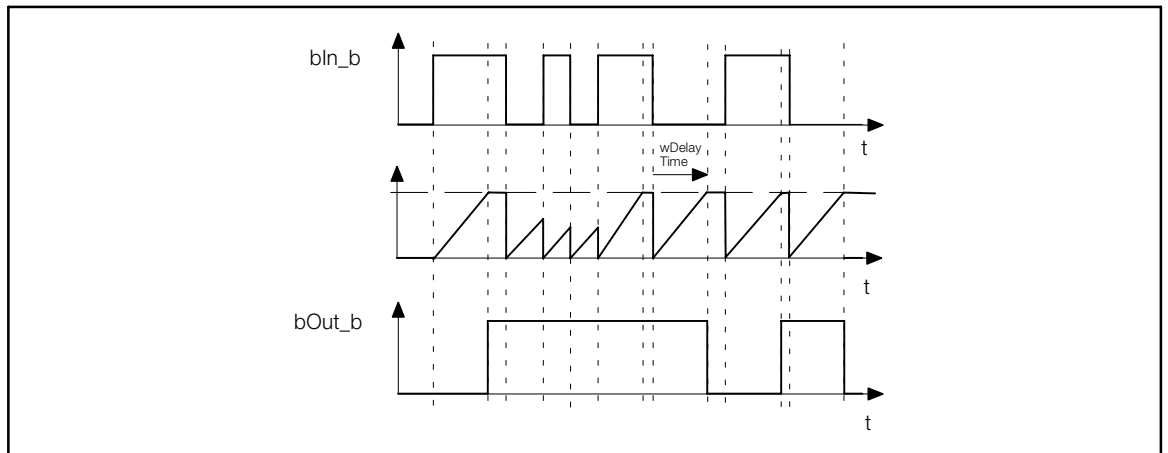
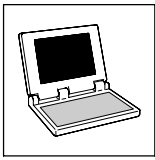


Abb. 2-37

Allgemeine Verzögerung

### Funktionsablauf

1. Eine beliebige Flanke an  $nIn\_b$  setzt das Zeitglied zurück und startet es.
2. Erst wenn die Verzögerungszeit abgelaufen ist, die mit  $wDelayTime$  eingestellt wurde, schaltet  $bOut\_b = nIn\_b$ .



# Funktionsbibliothek LenzeDrive.lib

## Digitale Signalverarbeitung

### Auf-/Abwärtszähler (L\_FCNT)

## 2.3.3 Auf-/Abwärtszähler (L\_FCNT)

Dieser FB ist ein digitaler Aufwärts-und Abwärtszähler, der auf den Eingangswert *nCmpVal\_a* begrenzt wird.

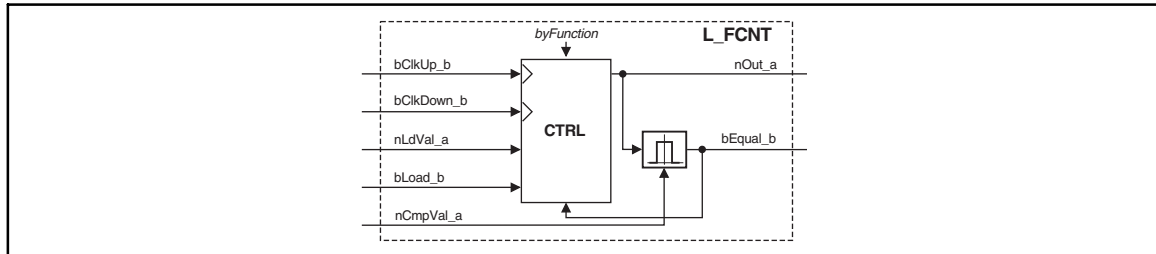


Abb. 2-38

Auf-/Abwärtszähler (L\_FCNT)

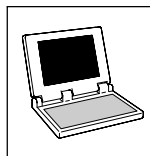
| VariableName | DataType | SignalType | VariableType        | Note                                                                                                                      |
|--------------|----------|------------|---------------------|---------------------------------------------------------------------------------------------------------------------------|
| bCikUp_b     | Bool     | binary     | VAR_INPUT           | FALSE-TRUE-Flanke = zählt um 1 aufwärts.                                                                                  |
| bCikDwn_b    | Bool     | binary     | VAR_INPUT           | FALSE-TRUE-Flanke = zählt um 1 abwärts.                                                                                   |
| nLdVal_a     | Integer  | analog     | VAR_INPUT           | Startwert                                                                                                                 |
| bLoad_b      | Bool     | binary     | VAR_INPUT           | <ul style="list-style-type: none"> <li>TRUE = Startwert übernehmen</li> <li>Der Eingang hat höchste Priorität.</li> </ul> |
| nCmpVal_a    | Integer  | analog     | VAR_INPUT           | Vergleichswert                                                                                                            |
| nOut_a       | Integer  | analog     | VAR_OUTPUT          | Zählerstand wird begrenzt auf $\pm 32767$ .                                                                               |
| bEqual_b     | Bool     | binary     | VAR_OUTPUT          | TRUE = Vergleichswert erreicht.                                                                                           |
| byFunction   | Byte     |            | VAR CONSTANT RETAIN | Auswahl der Funktion                                                                                                      |

### Parameter-Codestellen der Instanzen

| VariableName | L_FCNT1 |  |  | SettingRange | Lenze |
|--------------|---------|--|--|--------------|-------|
| byFunction   | C1100   |  |  | 1 ... 2      | 1     |

### Funktion

| Auswahl der Funktion | Beschreibung                                                                                                                                                                                                                                                                                                                      |
|----------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| byFunction = 1       | <ul style="list-style-type: none"> <li>Ist der <math> Z\ddot{a}hlerstand  \geq  nCmpVal_a </math>, wird der Ausgang <i>bEqual_b</i> auf TRUE gesetzt. Beim nächsten Taktzyklus wird der Zähler auf den Wert <i>nLdVal_a</i> zurückgesetzt und der Ausgang <i>bEqual_b</i> wird auf FALSE zurückgesetzt.</li> </ul>                |
| byFunction = 2       | <ul style="list-style-type: none"> <li>Ist der <math> Z\ddot{a}hlerstand  =  nCmpVal_a </math>, hält der Zähler an (<i>bCikUp_b</i>/<i>bCikDwn_b</i> werden ignoriert).</li> <li><i>bLoad_b</i> = TRUE setzt den Zähler auf den Wert an <i>nLdVal_a</i> und auf <i>bCikUp_b</i>/<i>bCikDwn_b</i> wird wieder reagiert.</li> </ul> |



### 2.3.4 Flip-Flop (L\_FLIP)

Dieser FB ist als D-Flip-Flop ausgeführt. Mit dieser Funktion können Sie digitale Signalfanken auswerten und speichern.

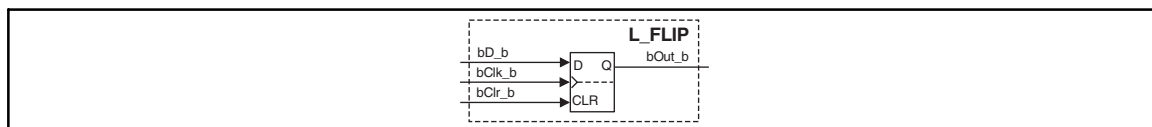


Abb. 2-39

Flip-Flop (L\_FLIP)

| VariableName | DataType | SignalType | VariableType | Note                                                                             |
|--------------|----------|------------|--------------|----------------------------------------------------------------------------------|
| bD_b         | Bool     | binary     | VAR_INPUT    |                                                                                  |
| bClk_b       | Bool     | binary     | VAR_INPUT    | Wertet nur FALSE-TRUE-Flanken aus (Flanken-getriggert).                          |
| bClr_b       | Bool     | binary     | VAR_INPUT    | Wertet nur den Eingangspegel aus; Eingang hat höchste Priorität (Reset-Eingang). |
| bOut_b       | Bool     | binary     | VAR_OUTPUT   |                                                                                  |

### Funktionsablauf

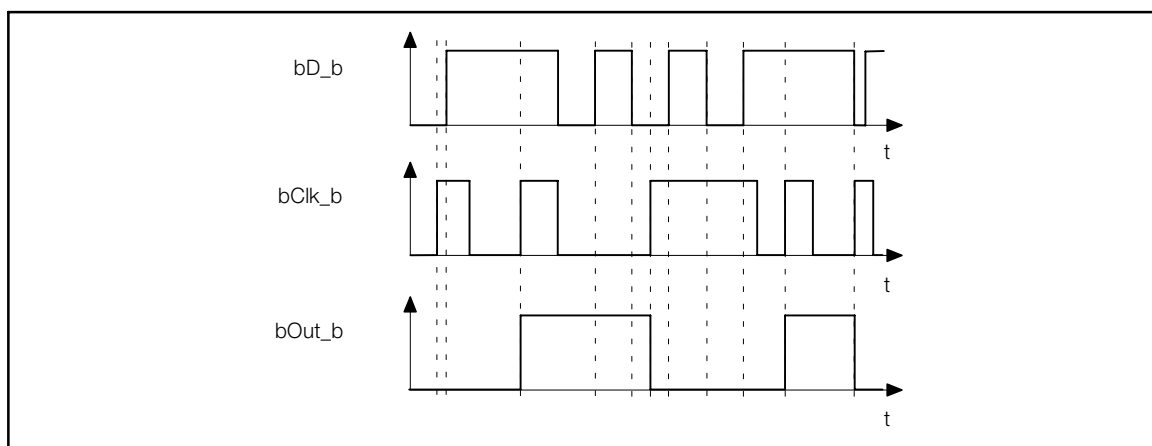
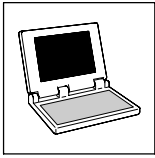


Abb. 2-40

Funktionsablauf des Speicherglieds

*bClr\_b* hat immer Priorität.

1. Wenn *bClr\_b* = TRUE ist, schaltet *bOut\_b* = FALSE. Dieser Zustand wird gehalten, solange *bClr\_b* = TRUE ist.
2. Eine FALSE-TRUE-Flanke an *bClk\_b* schaltet *bD\_b* = *bOut\_b*. Dieser Zustand wird solange gespeichert, bis
  - erneut eine FALSE-TRUE-Flanke an *bClk\_b* erfolgt oder
  - *bClr\_b* = TRUE geschaltet wird.



# Funktionsbibliothek LenzeDrive.lib

## Digitale Signalverarbeitung

### Logisches NICHT (L\_NOT)

## 2.3.5 Logisches NICHT (L\_NOT)

Dieser FB ermöglicht das logische Invertieren von binären Signalen. Sie können den FB zur Steuerung von Funktionen oder zur Bildung von Statusinformationen verwenden.

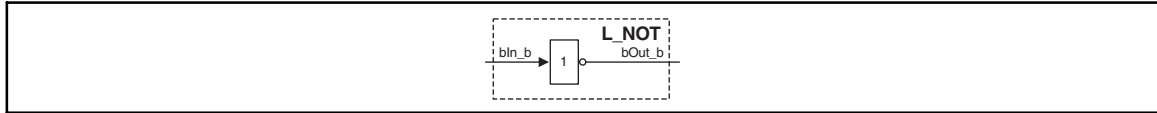


Abb. 2-41

Logisches NICHT (L\_NOT)

| VariableName | DataType | SignalType | VariableType | Note |
|--------------|----------|------------|--------------|------|
| bln_b        | Bool     | binary     | VAR_INPUT    |      |
| bOut_b       | Bool     | binary     | VAR_OUTPUT   |      |

### Wahrheitstabelle

| bln_b | bOut_b |
|-------|--------|
| 0     | 1      |
| 1     | 0      |

0 = FALSE

1 = TRUE

Die Funktion entspricht einer Umsetzung von einem Schließer auf einen Öffner bei einer Schützsteuerung.

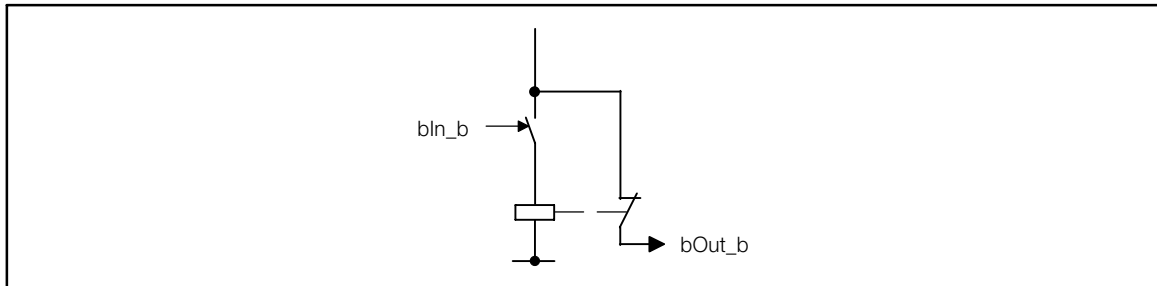
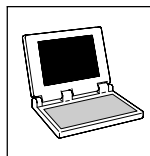


Abb. 2-42

Funktion von L\_NOT als Umsetzung von einem Schließer auf einen Öffner



## 2.3.6 Logisches ODER (L\_OR)

Dieser FB ermöglicht die logische ODER-Verknüpfung von binären Signalen. Sie können die Verknüpfungen zur Steuerung von Funktionen oder zur Bildung von Statusinformationen verwenden.

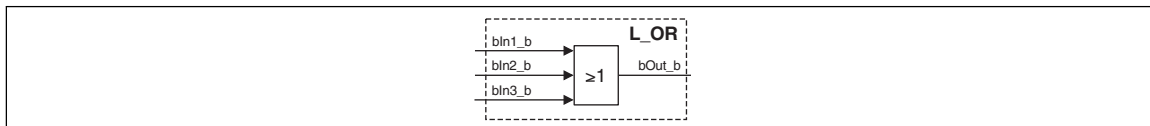


Abb. 2-43

Logisches ODER (L\_OR)

| VariableName | DataType | SignalType | VariableType | Note |
|--------------|----------|------------|--------------|------|
| bln1_b       | Bool     | binary     | VAR_INPUT    |      |
| bln2_b       | Bool     | binary     | VAR_INPUT    |      |
| bln3_b       | Bool     | binary     | VAR_INPUT    |      |
| bOut_b       | Bool     | binary     | VAR_OUTPUT   |      |

### Wahrheitstabelle

| bln1_b | bln2_b | bln3_b | bOut_b |
|--------|--------|--------|--------|
| 0      | 0      | 0      | 0      |
| 1      | 0      | 0      | 1      |
| 0      | 1      | 0      | 1      |
| 1      | 1      | 0      | 1      |
| 0      | 0      | 1      | 1      |
| 1      | 0      | 1      | 1      |
| 0      | 1      | 1      | 1      |
| 1      | 1      | 1      | 1      |

0 = FALSE

1 = TRUE

Die Funktion entspricht einer Parallelschaltung von Schließern bei einer Schützsteuerung.

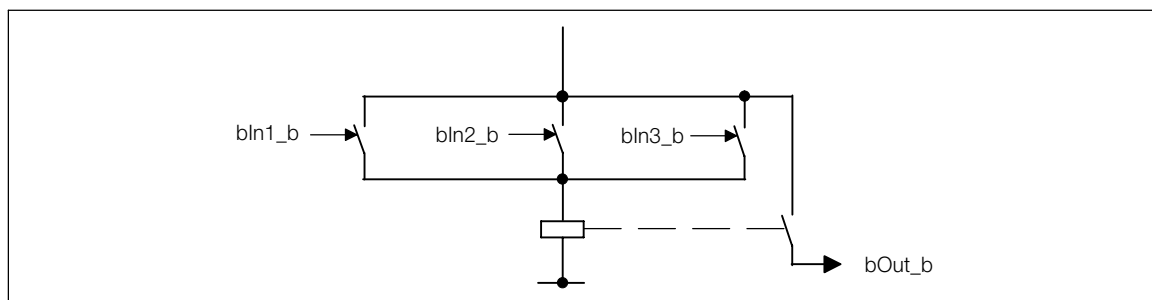


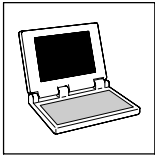
Abb. 2-44

Funktion von L\_OR als Parallelschaltung von Schließern



### Tipp!

Wenn Sie nur 2 Eingänge benötigen, verwenden Sie die Eingänge *bln1\_b* und *bln2\_b*. Legen Sie den Eingang *bln3\_b* auf FALSE.



# Funktionsbibliothek LenzeDrive.lib

## Digitale Signalverarbeitung

### Flankenwertung (L\_TRANS)

## 2.3.7 Flankenwertung (L\_TRANS)

Dieser FB ist ein nachtriggerbarer Flankenauswerter. Mit ihm können Sie digitale Signalfanken auswerten und in zeitlich definierte Impulse umwandeln.

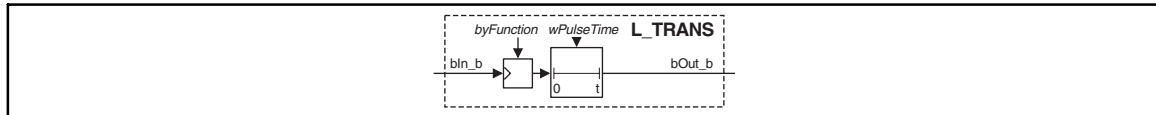


Abb. 2-45

Flankenwertung (L\_TRANS)

| VariableName | DataType | SignalType | VariableType        | Note                            |
|--------------|----------|------------|---------------------|---------------------------------|
| bIn_b        | Bool     | binary     | VAR_INPUT           |                                 |
| bOut_b       | Bool     | binary     | VAR_OUTPUT          | (nachtriggerbar)                |
| byFunction   | Byte     |            | VAR CONSTANT RETAIN | Auswahl der Funktion            |
| wPulseTime   | Word     |            | VAR CONSTANT RETAIN | Impulsdauer des Ausgangssignals |

### Parameter-Codestellen der Instanzen

| VariableName | L_TRANS1 | L_TRANS2 | L_TRANS3 | SettingRange       | Lenze |
|--------------|----------|----------|----------|--------------------|-------|
| byFunction   | C0710    | C0715    | C1140    | 0 ... 2            | 0     |
| wPulseTime   | C0711    | C0716    | C1141    | 0.001 ... 60.000 s | 0.001 |

| VariableName | L_TRANS4 |  |  | SettingRange       | Lenze |
|--------------|----------|--|--|--------------------|-------|
| byFunction   | C1145    |  |  | 0 ... 2            | 0     |
| wPulseTime   | C1146    |  |  | 0.001 ... 60.000 s | 0.001 |

### Funktionsumfang

- Steigende Signalfanken auswerten
- Fallende Signalfanken auswerten
- Steigende und fallende Signalfanken auswerten

## 2.3.7.1 Steigende Signalfanken auswerten

*byFunction* = 0

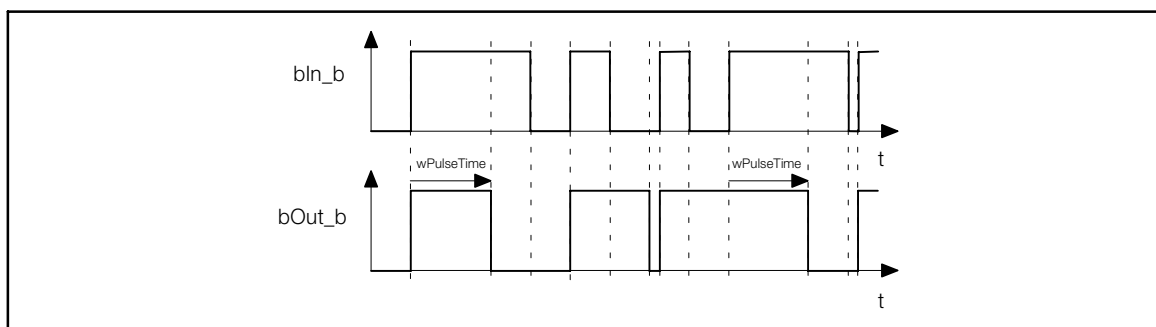
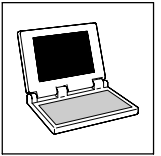


Abb. 2-46

Auswertung von FALSE-TRUE-Flanken

### Funktionsablauf

1. Erfolgt an *nIn\_b* eine FALSE-TRUE-Flanke, schaltet *bOut\_b* = TRUE.
2. Nach Ablauf der mit *wPulseTime* eingestellten Zeit schaltet *bOut\_b* = FALSE, sofern inzwischen keine weitere FALSE-TRUE-Flanke an *nIn\_b* erfolgt ist.



## 2.3.7.2 Fallende Signalflanken auswerten

byFunction = 1

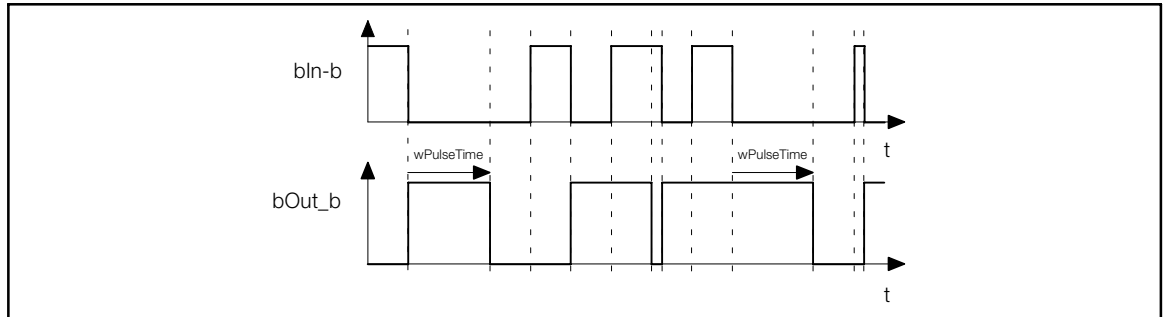


Abb. 2-47

Auswertung TRUE-FALSE-Flanken

### Funktionsablauf

1. Erfolgt an *nIn\_b* eine TRUE-FALSE-Flanke, schaltet *bOut\_b* = TRUE.
2. Nach Ablauf der mit *wPulseTime* eingestellten Zeit schaltet *bOut\_b* = FALSE, sofern inzwischen keine weitere TRUE-FALSE-Flanke an *nIn\_b* erfolgt ist.

## 2.3.7.3 Steigende und fallende Signalflanken auswerten

byFunction = 2

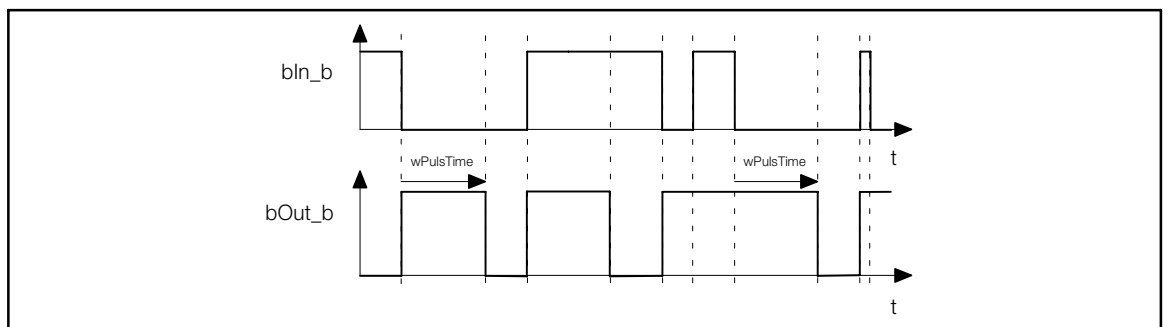
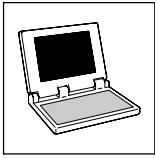


Abb. 2-48

Auswertung beider Flanken

### Funktionsablauf

1. Erfolgt an *nIn\_b* eine FALSE-TRUE-Flanke oder TRUE-FALSE-Flanke, schaltet *bOut\_b* = TRUE.
2. Nach Ablauf der mit *wPulseTime* eingestellten Zeit schaltet *bOut\_b* = FALSE, sofern inzwischen keine weitere FALSE-TRUE-Flanke oder TRUE-FALSE-Flanke an *nIn\_b* erfolgt ist.



# Funktionsbibliothek LenzeDrive.lib

## Verarbeitung von Winkelsignalen

### Arithmetik (L\_ARITPH)

## 2.4 Verarbeitung von Winkelsignalen

### 2.4.1 Arithmetik (L\_ARITPH)

Dieser FB errechnet aus zwei Winkeleingangssignalen ein Winkelausgangssignal.

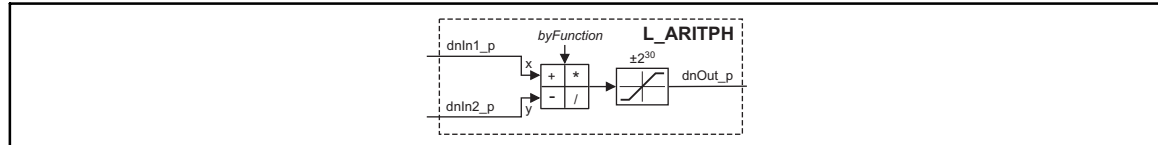


Abb. 2-49

Arithmetik (L\_ARITPH)

| VariableName | DataType       | SignalType | VariableType        | Note                                    |
|--------------|----------------|------------|---------------------|-----------------------------------------|
| dnln1_p      | Double Integer | position   | VAR_INPUT           |                                         |
| dnln2_p      | Double Integer | position   | VAR_INPUT           |                                         |
| dnOut_p      | Double Integer | position   | VAR_OUTPUT          | Signal wird begrenzt auf $\pm 2^{30}$ . |
| byFunction   | Byte           |            | VAR_CONSTANT RETAIN | Auswahl der Funktion                    |

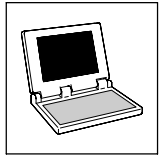
#### Parameter-Codestellen der Instanzen

| VariableName | L_ARITPH1 |  |  | SettingRange        | Lenze |
|--------------|-----------|--|--|---------------------|-------|
| byFunction   | C1010     |  |  | 0 ... 3, 14, 21, 22 | 1     |

#### Funktion

| Auswahl der Funktion | Arithmetische Funktion                          | Begrenzung des Ergebnisses | Bemerkung                    |
|----------------------|-------------------------------------------------|----------------------------|------------------------------|
| byFunction = 0       | $dnOut\_p = dnln1\_p$                           | ohne                       | dnOut_p wird nicht begrenzt. |
| byFunction = 1       | $dnOut\_p = dnln1\_p + dnln2\_p$                | $2^{30}$                   |                              |
| byFunction = 2       | $dnOut\_p = dnln1\_p - dnln2\_p$                | $2^{30}$                   |                              |
| byFunction = 3       | $dnOut\_p = (dnln1\_p \cdot dnln2\_p) / 2^{30}$ | $2^{30}$                   | (ohne Restwertbehandlung)    |
| byFunction = 14      | $dnOut\_p = dnln1\_p / dnln2\_p$                | $2^{30}$                   | (ohne Restwertbehandlung)    |
| byFunction = 21      | $dnOut\_p = dnln1\_p + dnln2\_p$                | ohne                       | mit Überlauf                 |
| byFunction = 22      | $dnOut\_p = dnln1\_p - dnln2\_p$                | ohne                       | mit Überlauf                 |

- byFunction = 21/22:  
Beachten Sie, dass ein Überlauf vorkommen kann und der Zahlenwert von *dnOut\_p* nicht dem Ergebnis entspricht.
- byFunction = 14:  
Wird der Nenner = 0, so wird  $dnOut\_p = \pm 2^{30}$  gesetzt. Das Vorzeichen hängt vom Vorzeichen von *dnln1\_p* ab.



### 2.4.2 Addition (L\_PHADD)

Dieser FB addiert oder subtrahiert Winkelsignale, je nach verwendetem Eingang.

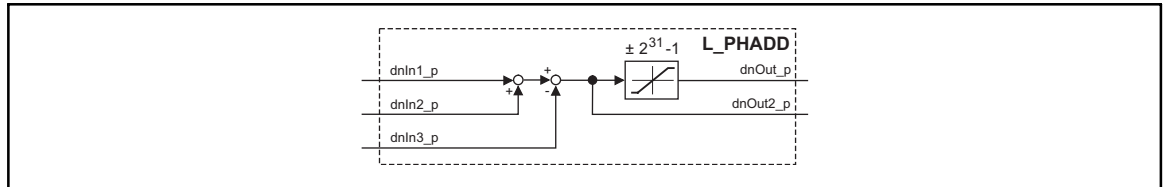


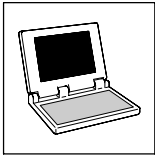
Abb. 2-50

Addition (L\_PHADD)

| VariableName | DataType       | SignalType | VariableType | Note                                      |
|--------------|----------------|------------|--------------|-------------------------------------------|
| dnIn1_p      | Double Integer | position   | VAR_INPUT    | Additionseingang                          |
| dnIn2_p      | Double Integer | position   | VAR_INPUT    | Additionseingang                          |
| dnIn3_p      | Double Integer | position   | VAR_INPUT    | Subtraktionseingang                       |
| dnOut_p      | Double Integer | position   | VAR_OUTPUT   | Signal wird begrenzt auf $\pm 2147483647$ |
| dnOut2_p     | Double Integer | position   | VAR_OUTPUT   | Signal ohne Begrenzung/mit Überlauf       |

#### Funktionsablauf

1. Das Signal an *dnIn1\_p* wird zum Signal an *dnIn2\_p* addiert.
2. Vom errechneten Ergebnis wird das Signal an *dnIn3\_p* subtrahiert.
3. Anschließend wird das Ergebnis der Subtraktion auf  $\pm 2147483647$  begrenzt an *dnOut\_p* und unbegrenzt an *dnOut2\_p* ausgegeben.  
Beachten Sie, dass an *dnOut2\_p* ein Überlauf und somit eine falsche Ausgabe entstehen kann.



# Funktionsbibliothek LenzeDrive.lib

## Verarbeitung von Winkelsignalen

### Vergleich (L\_PHCMP)

## 2.4.3 Vergleich (L\_PHCMP)

Dieser FB vergleicht zwei Winkelsignale (Wege) miteinander.

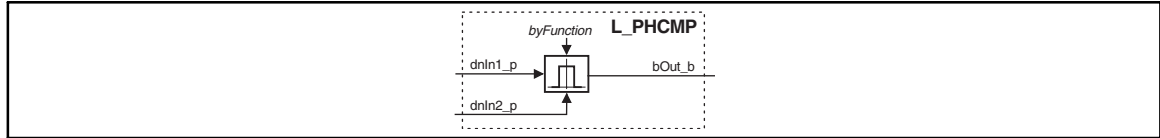


Abb. 2-51

Vergleich (L\_PHCMP)

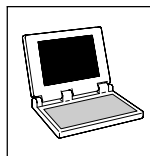
| VariableName | DataType       | SignalType | VariableType        | Note                     |
|--------------|----------------|------------|---------------------|--------------------------|
| dnln1_p      | Double Integer | position   | VAR_INPUT           | zu vergleichendes Signal |
| dnln2_p      | Double Integer | position   | VAR_INPUT           | Vergleichswert           |
| bOut_b       | Bool           | binary     | VAR_OUTPUT          |                          |
| byFunction   | Byte           |            | VAR CONSTANT RETAIN | Auswahl der Funktion     |

### Parameter-Codestellen der Instanzen

| VariableName | L_PHCMP1 | L_PHCMP2 | L_PHCMP3 | SettingRange | Lenze |
|--------------|----------|----------|----------|--------------|-------|
| byFunction   | C0695    | C1207    | C1272    | 1 ... 2      | 2     |

### Funktion

| Auswahl        | Vergleichsfunktion           | Wenn die Vergleichsfunktion erfüllt ist | Bemerkung                          |
|----------------|------------------------------|-----------------------------------------|------------------------------------|
| byFunction = 1 | $dnln1\_p < dnln2\_p$        | bOut_b = HIGH                           |                                    |
|                | $dnln1\_p \geq dnln2\_p$     | bOut_b = LOW                            |                                    |
| byFunction = 2 | $ dnln1\_p  <  dnln2\_p $    | bOut_b = HIGH                           | Vergleicht den Betrag der Eingänge |
|                | $ dnln1\_p  \geq  dnln2\_p $ | bOut_b = LOW                            |                                    |



### 2.4.4 Differenz (L\_PHDIFF)

Dieser FB addiert gezielt ein Winkelsignal zum Winkelsollwert. Außerdem ist ein Soll/Ist-Vergleich von Winkelsignalen möglich.

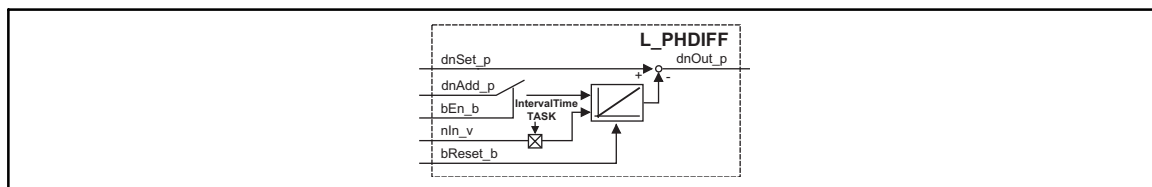


Abb. 2-52

Differenz (L\_PHDIFF)

| VariableName | DataType       | SignalType | VariableType | Note                                                        |
|--------------|----------------|------------|--------------|-------------------------------------------------------------|
| dnSet_p      | Double Integer | position   | VAR_INPUT    | Vorgabe eines Positionssollwertes                           |
| dnAdd_p      | Double Integer | position   | VAR_INPUT    | Adaptiver Positionswert zur Istposition                     |
| bEn_b        | Bool           | binary     | VAR_INPUT    | TRUE = Adaptiver Positionswert wird hinzuaddiert.           |
| nIn_v        | Integer        | velocity   | VAR_INPUT    | Vorgabe der Istdrehzahl zur Umrechnung in den Positionswert |
| bReset_b     | Bool           | binary     | VAR_INPUT    | TRUE = Istwinkelintegrator wird auf 0 gesetzt.              |
| dnOut_p      | Double Integer | position   | VAR_OUTPUT   | Signal ist ohne Begrenzung.                                 |

#### Funktionsablauf

Wenn *bEn\_b* = TRUE ist:

1. Das Drehzahlsignal an *nIn\_v* wird vom Winkelintegrator aufintegriert.
2. Das Winkelsignal an *dnAdd\_p* wird zum integrierten Drehzahlsignal in jedem Taskzyklus hinzuaddiert.
3. Das Ergebnis des Winkelintegrators wird vom Winkelsignal an *dnSet\_p* subtrahiert und anschließend an *dnOut\_p* ausgegeben.

Wenn *bEn\_b* = FALSE ist:

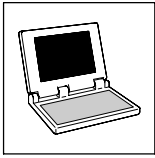
1. Das Drehzahlsignal an *nIn\_v* wird vom Winkelintegrator aufintegriert.
2. Das Ergebnis des Winkelintegrators wird vom Winkelsignal an *dnSet\_p* subtrahiert und anschließend an *dnOut\_p* ausgegeben.



#### Tipp!

Über den Winkelintegrator wird aus einer Drehzahl eine Position ermittelt.

- Über *nIn\_v* kann die Istdrehzahl vorgegeben werden (16384  $\equiv$  15000 rpm ).
- (INT)65536 entspricht einer Geberumdrehung.



# Funktionsbibliothek LenzeDrive.lib

## Verarbeitung von Winkelsignalen

### Division (L\_PHDIV)

#### 2.4.5 Division (L\_PHDIV)

Dieser FB dividiert oder multipliziert Winkelsignale im Zweierpotenzformat.

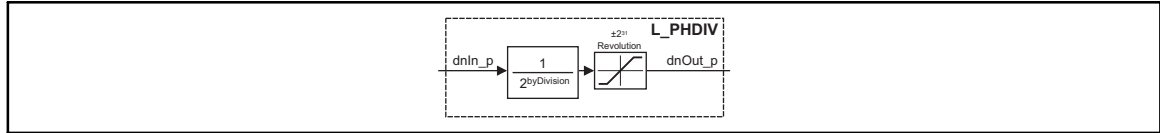


Abb. 2-53

Division (L\_PHDIV)

| VariableName | DataType       | SignalType | VariableType        | Note                         |
|--------------|----------------|------------|---------------------|------------------------------|
| dnIn_p       | Double integer | position   | VAR_INPUT           |                              |
| dnOut_p      | Double integer | position   | VAR_OUTPUT          | 65536 inc = 1 Geberumdrehung |
| byDivision   | Short Integer  |            | VAR CONSTANT RETAIN | Potenzwert des Divisors      |

#### Parameter-Codestellen der Instanzen

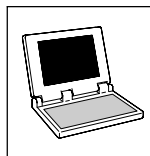
| VariableName | L_PHDIV1 |  |  | SettingRange | Lenze |
|--------------|----------|--|--|--------------|-------|
| byDivision   | C0995    |  |  | -31 ... 31   | 0     |

#### Funktion

Das Ergebnis der arithmetischen Funktion berechnen Sie nach der Formel:

$$dnOut\_p = \frac{dnIn\_p}{2^{byDivision}}$$

- Positive Werte in *byDivision* führen zu einer Division.
- Negative Werte in *byDivision* führen zu einer Multiplikation.
- Das Ausgangssignal wird auf  $\pm 2^{31}-1$  Geberumdrehungen begrenzt.
  - Das Ausgangssignal kann diesen Grenzwert nicht überschreiten.



### 2.4.6 Integration (L\_PHINT)

Dieser FB kann eine Drehzahl oder eine Geschwindigkeit zu einem Winkel (Weg) aufintegrieren. Der Integrator kann max.  $\pm 32000$  Geberumdrehungen aufnehmen.

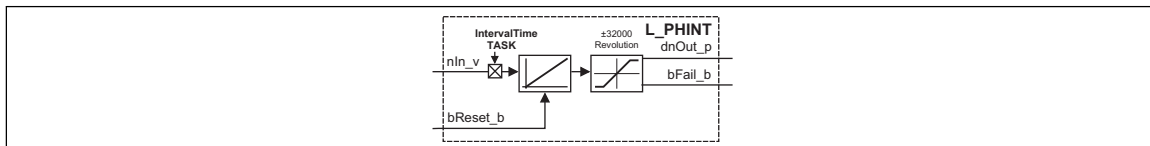


Abb. 2-54

Integration (L\_PHINT)

| VariableName | DataType       | SignalType | VariableType | Note                                                 |
|--------------|----------------|------------|--------------|------------------------------------------------------|
| nIn_v        | Integer        | velocity   | VAR_INPUT    | Drehzahlwert: 16384 $\equiv$ 15000 rpm               |
| bReset_b     | Bool           | binary     | VAR_INPUT    | TRUE setzt Winkelintegrator = 0 und bFail_b = FALSE. |
| dnOut_p      | Double Integer | position   | VAR_OUTPUT   | 65536 inc = 1 Geberumdrehung (Überlauf ist möglich). |
| bFail_b      | Bool           | binary     | VAR_OUTPUT   | TRUE = Überlauf ist erfolgt.                         |

#### Funktionsumfang

- Konstanter Eingangswert
- Berechnung des Ausgangssignals

#### 2.4.6.1 Konstanter Eingangswert

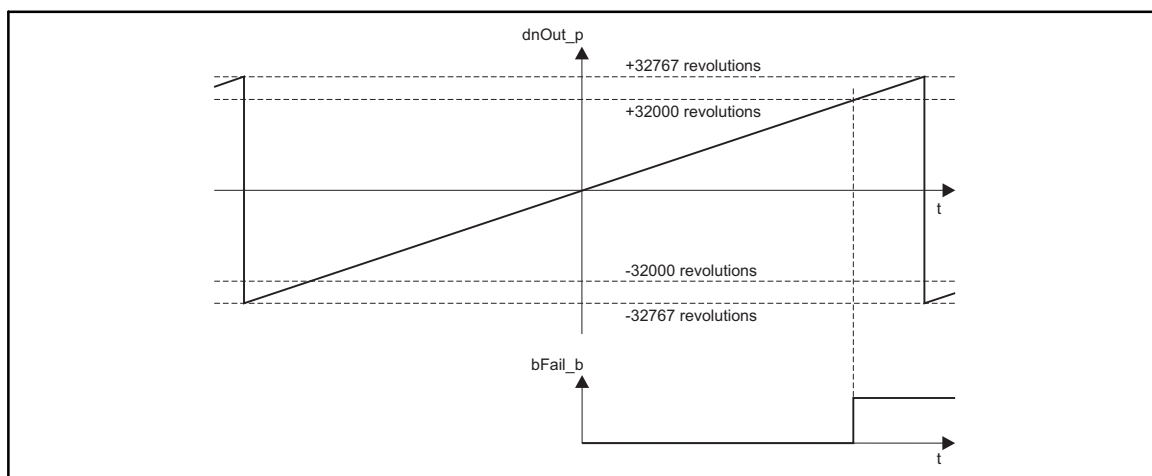
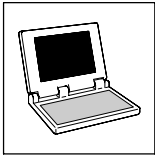


Abb. 2-55

Funktion von L\_PHINT mit konstantem Eingangswert

- Ein positives Signal an *nIn\_v* wird inkrementiert (der Zählerstand wird bei jedem Funktionsaufruf erhöht).
- Ein negatives Signal an *nIn\_v* wird dekrementiert (der Zählerstand wird bei jedem Funktionsaufruf verringert).
- *dnOut\_p* gibt den Zählerstand des bipolaren Integrators aus.
- Überschreitet der Zählerstand den Wert von +32000 Geberumdrehungen, schaltet *bFail\_b* = TRUE.
- Überschreitet der Zählerstand den Wert von +32767 Geberumdrehungen (entspricht +2147483647 inc), erfolgt ein Überlauf und der Zählvorgang wird beim Wert -32768 Geberumdrehungen fortgesetzt.



## Funktionsbibliothek LenzeDrive.lib

### Verarbeitung von Winkelsignalen

#### Integration (L\_PHINT)

- Unterschreitet der Zählerstand den Wert von -32000 Geberumdrehungen, schaltet  $bFail\_b = TRUE$ .
- Unterschreitet der Zählerstand den Wert von -32768 Geberumdrehungen (entspricht -2147483648 inc), erfolgt ein Überlauf und der Zählvorgang beginnt beim Wert +32767 Geberumdrehungen.
- $bReset\_b = TRUE$ :
  - schaltet der Integrator auf 0.
  - setzt  $dnOut\_p$  auf 0, solange an  $nIn\_v$  ein positives Signal anliegt.
  - schaltet  $bFail\_b = FALSE$ .

### 2.4.6.2 Berechnung des Ausgangssignals

Den Ausgangswert an  $dnOut\_p$  ermitteln Sie nach der Formel:

$$dnOut\_p [inc] = nIn\_v [rpm] \cdot t [s] \cdot 65536 [inc/Umdr.]$$

( $t$  = Integrationszeit, 16384  $\approx$  15000 rpm, 1 inc = 1)

#### Beispiel:

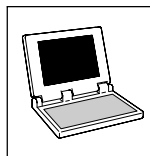
Sie möchten den Zählerstand des Integrators bei einer bestimmten Drehzahl am Eingang und einer bestimmten Integrationszeit  $t$  ermitteln.

- Vorgegebene Werte:
  - $nIn\_v = 1000 \text{ rpm} \approx (INT)1092$
  - $t = 10 \text{ s}$
  - Startwert des Integrators ist 0.
- Lösung:
  - Umrechnung des Eingangssignals an  $nIn\_v$ :

$$1000 \text{ rpm} = \frac{1000 \text{ Umdr.}}{60 \text{ s}}$$

- Berechnung des Ausgangswerts:

$$dnOut\_p = \frac{1000 \text{ Umdr.}}{60 \text{ s}} \cdot 10 \text{ s} \cdot \frac{65536 \text{ inc}}{\text{Umdr.}} = 10922666 \text{ inc}$$



### 2.4.7 Integration (L\_PHINTK)

Der FB kann eine Drehzahl oder eine Geschwindigkeit zu einem Winkel (Weg) aufintegrieren. Zusätzlich kann er eine relativ zurückgelegte Wegstrecke erkennen. Der Integrator kann max.  $\pm 32000$  Geberumdrehungen aufnehmen.

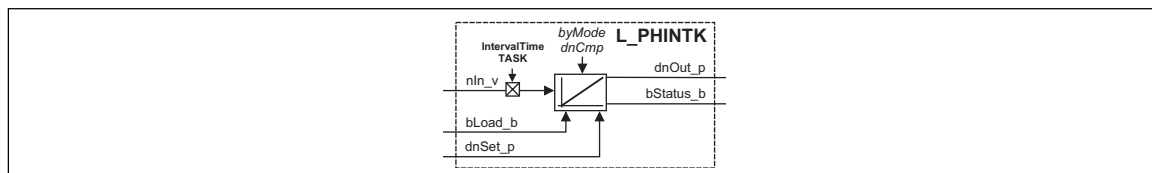


Abb. 2-56

Integration (L\_PHINTK)

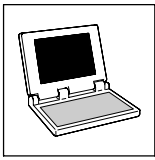
| VariableName | DataType       | SignalType | VariableType        | Note                                                                           |
|--------------|----------------|------------|---------------------|--------------------------------------------------------------------------------|
| nIn_v        | Integer        | velocity   | VAR_INPUT           | Drehzahlwert: 16384 $\equiv$ 15000 rpm                                         |
| bLoad_b      | Bool           | binary     | VAR_INPUT           | TRUE setzt den Winkelintegrator auf das Signal an nIn_v und bStatus_b = FALSE. |
| dnSet_p      | Double Integer | position   | VAR_INPUT           |                                                                                |
| dnOut_p      | Double Integer | position   | VAR_OUTPUT          | 65536 inc = 1 Geberumdrehung (Überlauf ist möglich.)                           |
| bStatus_b    | Bool           | binary     | VAR_OUTPUT          | TRUE = Überlauf ist erfolgt oder Wegstrecke abgearbeitet.                      |
| byMode       | Byte           |            | VAR CONSTANT RETAIN | Auswahl der Funktion                                                           |
| dnCmp        | Double Integer |            | VAR CONSTANT RETAIN | Vergleichswert                                                                 |

#### Parameter-Codestellen der Instanzen

| VariableName | L_PHINTK1 |  |  | SettingRange     | Lenze      |
|--------------|-----------|--|--|------------------|------------|
| byMode       | C1150     |  |  | 0 ... 2          | 0          |
| dnCmp        | C1151     |  |  | 0 ... 2000000000 | 2000000000 |

#### Funktionsumfang

- Konstanter Eingangswert
- Eingangswert mit Vorzeichenwechsel
- Berechnungen des Ausgangssignals



# Funktionsbibliothek LenzeDrive.lib

## Verarbeitung von Winkelsignalen

### Integration (L\_PHINTK)

#### 2.4.7.1 Konstanter Eingangswert

Ihnen stehen 3 Funktionen zur Verfügung, die Sie mit *byMode* wählen können.

##### byMode = 0

Der Eingang *bLoad\_b* ist zustandsgesteuert (TRUE-Pegel).

- *bLoad\_b* = TRUE
  - Der Integrator wird mit dem Wert an *dnSet\_p* geladen.
  - Der FB schaltet *bStatus\_b* = FALSE.

##### byMode = 1

Der Eingang *bLoad\_b* ist flankengesteuert (FALSE-TRUE-Flanke).

- *bLoad\_b* = FALSE-TRUE-Flanke
  - Der Integrator wird mit dem Wert an *dnSet\_p* geladen und integriert von dort sofort weiter auf.
  - Der FB schaltet *bStatus\_b* = FALSE.

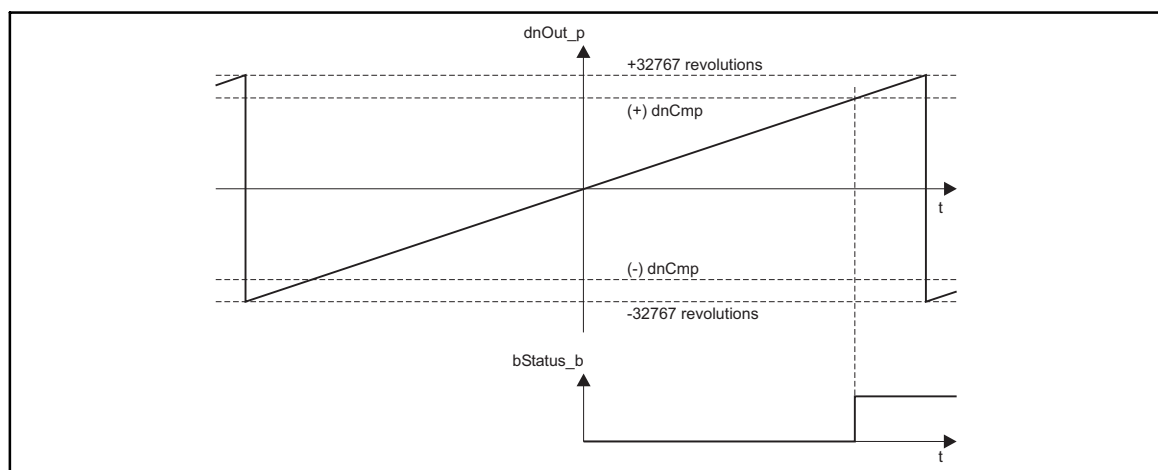
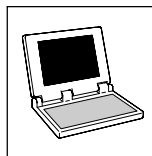


Abb. 2-57 Funktion von L\_PHINTK mit konstantem Eingangswert

(+) Variable mit positivem Wert

(-) Variable mit negativem Wert

- Ein positiver Wert an *nIn\_v* wird inkrementiert (der Zählerstand wird bei jedem Funktionsaufruf erhöht).
- Ein negativer Wert an *nIn\_v* wird dekrementiert (der Zählerstand wird bei jedem Funktionsaufruf verringert).
- *dnOut\_p* gibt den Zählerstand des bipolaren Integrators aus.
- Überschreitet der Zählerstand den Wert von +32767 Geberumdrehungen (entspricht +2147483647 inc):
  - Erfolgt ein Überlauf und der Zählvorgang wird fortgesetzt beim Wert -32768 Geberumdrehungen.
  - Schaltet *bStatus\_b* = TRUE, wenn ein positiv eingestellter Wert in *dnCmp* erreicht ist.
- Unterschreitet der Zählerstand den Wert von -32768 Geberumdrehungen (entspricht -2147483648 inc):
  - Erfolgt ein Überlauf und der Zählvorgang beginnt beim Wert +32767 Geberumdrehungen.
  - Schaltet *bStatus\_b* = TRUE, wenn ein negativ eingestellter Wert in *dnCmp* erreicht ist.



### 2.4.7.2 Eingangswert mit Vorzeichenwechsel

**byMode = 2**

Der Eingang *bLoad\_b* ist zustandsgesteuert (TRUE-Pegel).

- *bLoad\_b* = TRUE
  - Der Integrator wird mit dem Wert an *dnSet\_p* geladen.
  - Der FB schaltet *bStatus\_b* = FALSE.

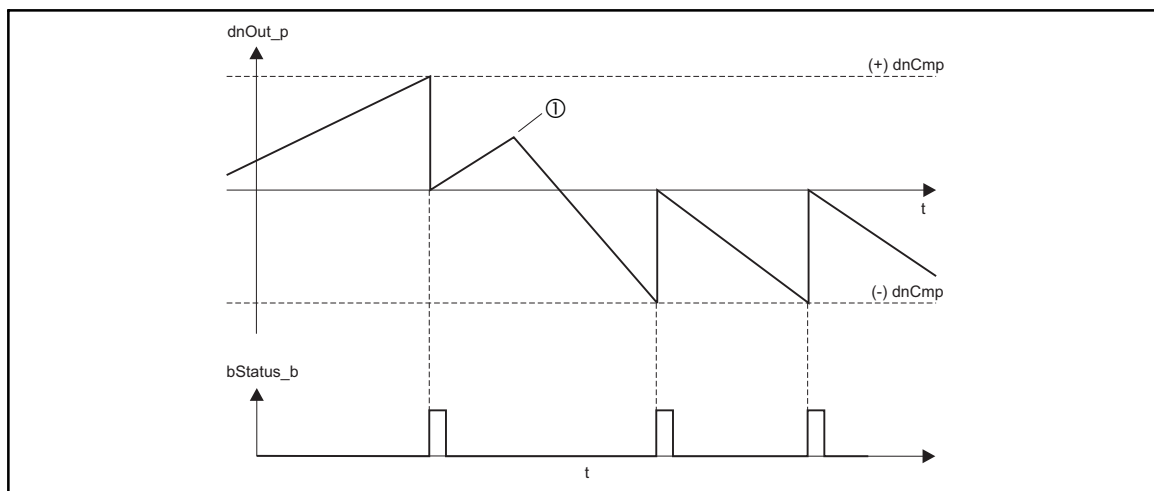


Abb. 2-58

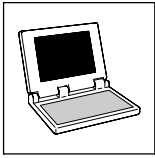
Funktion von L\_PHINTK mit Vorzeichenwechsel des Eingangswerts

(+) Variable mit positivem Wert

(-) Variable mit negativem Wert

① Vorzeichenwechsel des Werts an *nIn\_v*

- Ein positiver Wert an *nIn\_v* wird inkrementiert (der Zählerstand wird bei jedem Funktionsaufruf erhöht).
- Ein negativer Wert an *nIn\_v* wird dekrementiert (der Zählerstand wird bei jedem Funktionsaufruf verringert).
- *dnOut\_p* gibt den Zählerstand des bipolaren Integrators aus.
- Überschreitet der Zählerstand einen positiv eingestellten Wert in *dnCmp*:
  - Erfolgt eine Subtraktion des Zählerstandes um den Wert von *dnCmp*.
  - Schaltet für die Zeit von einem Taskzyklus *bStatus\_b* = TRUE.
- Unterschreitet der Zählerstand einen negativ eingestellten Wert in *dnCmp*:
  - Erfolgt eine Addition des Zählerstandes um den Wert von *dnCmp*.
  - Schaltet für die Zeit von einem Taskzyklus *bStatus\_b* = TRUE.



#### 2.4.7.3 Berechnung des Ausgangssignals

Den Ausgangswert an *dnOut\_p* ermitteln Sie nach der Formel:

$$\text{dnOut\_p [inc]} = \text{nIn\_v [rpm]} \cdot \text{t [s]} \cdot 65536 [\text{inc/Umdr.}]$$

(t = Integrationszeit, 16384  $\approx$  15000 rpm, 1 inc = 1)

#### Beispiel:

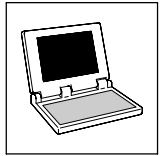
Sie möchten den Zählerstand des Integrators bei einer bestimmten Drehzahl am Eingang und einer bestimmten Integrationszeit t ermitteln.

- Vorgegebene Werte:
  - *nIn\_v* = 1000 rpm  $\approx$  (INT)1092
  - t = 10 s
  - Startwert des Integrators ist 0.
- Lösung:
  - Umrechnung des Eingangssignals an *nIn\_v*:

$$1000 \text{ rpm} = \frac{1000 \text{ Umdr.}}{60 \text{ s}}$$

- Berechnung des Ausgangswerts:

$$\text{dnOut\_p} = \frac{1000 \text{ Umdr.}}{60 \text{ s}} \cdot 10 \text{ s} \cdot \frac{65536 \text{ inc}}{\text{Umdr.}} = 10922666 \text{ inc}$$



## 2.5 Signalkonvertierung

### 2.5.1 Normierung (L\_CONV)

Dieser FB normiert Signale. Die Umrechnung erfolgt hochgenau mit Restwertverarbeitung mit Vorgabe des Umrechnungsfaktors als Zähler und Nenner.

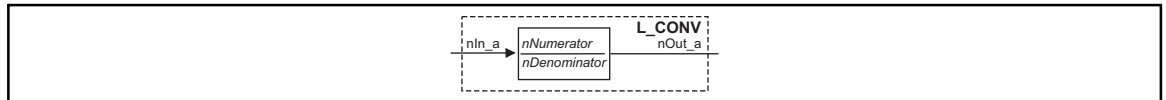


Abb. 2-59

Normierung (L\_CONV)

| VariableName | DataType | SignalType | VariableType        | Note                                                            |
|--------------|----------|------------|---------------------|-----------------------------------------------------------------|
| nIn_a        | Integer  | analog     | VAR_INPUT           | 100 % $\equiv$ 16384 $\equiv$ C0011 ( $n_{max}$ )               |
| nOut_a       | Integer  | analog     | VAR_OUTPUT          | Signal wird begrenzt auf $\pm 199.99$ % (100 % $\equiv$ 16384). |
| nNumerator   | Integer  |            | VAR CONSTANT RETAIN | Zähler                                                          |
| nDenominator | Integer  |            | VAR CONSTANT RETAIN | Nenner                                                          |

#### Parameter-Codestellen der Instanzen

| VariableName | L_CONV1 | L_CONV2 | L_CONV3 | SettingRange     | Lenze |
|--------------|---------|---------|---------|------------------|-------|
| nNumerator   | C0940   | C0945   | C0950   | -32767 ... 32767 | 1     |
| nDenominator | C0941   | C0946   | C0951   | 1 ... 32767      | 1     |

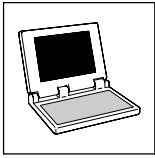
  

| VariableName | L_CONV4 | L_CONV5 | L_CONV6 | SettingRange     | Lenze |
|--------------|---------|---------|---------|------------------|-------|
| nNumerator   | C0955   | C0655   | C1170   | -32767 ... 32767 | 1     |
| nDenominator | C0956   | C0656   | C1171   | 1 ... 32767      | 1     |

#### Funktion

Die Multiplikation oder Division von Signalen erfolgt nach der Formel:

$$nOut = nIn \cdot \frac{nNumerator}{nDenominator}$$



# Funktionsbibliothek LenzeDrive.lib

## Signalkonvertierung

### Konvertierung Winkel nach Analog (L\_CONVPA)

## 2.5.2 Konvertierung Winkel nach Analog (L\_CONVPA)

Dieser FB konvertiert ein Winkelsignal in ein Integer-Signal.  
Die Funktion entspricht der des FB CONVPHA im Servo-Umrichter 9300.

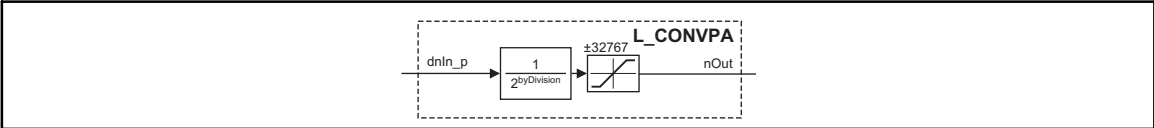


Abb. 2-60 Konvertierung Winkel nach Analog (L\_CONVPA)

| VariableName | DataType       | SignalType | VariableType        | Note                                                                                                            |
|--------------|----------------|------------|---------------------|-----------------------------------------------------------------------------------------------------------------|
| dnIn_p       | Double Integer | position   | VAR_INPUT           |                                                                                                                 |
| nOut         | Integer        | analog     | VAR_OUTPUT          | <ul style="list-style-type: none"><li>• Signal wird begrenzt auf ±32767.</li><li>• Restwertbehandlung</li></ul> |
| byDivision   | Byte           |            | VAR CONSTANT RETAIN | Teilfaktor                                                                                                      |

### Parameter-Codestellen der Instanzen

| VariableName | L_CONVPA1 |  |  | SettingRange | Lenze |
|--------------|-----------|--|--|--------------|-------|
| byDivision   | C1000     |  |  | 0 ... 31     | 1     |

### Funktion

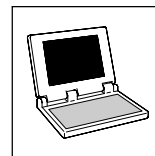
Die Umrechnung erfolgt nach der Formel:

$$nOut\_a = dnIn\_p \cdot \frac{1}{2^{byDivision}}$$



### Tipp!

Dieser FB arbeitet mit Restwertbehandlung.



### 2.5.3 Umrechnung Winkelsignal (L\_CONVPP)

Dieser FB rechnet ein Winkelsignal mit dynamischem Bruch um.

Die Funktion entspricht der des FB CONVPPH im Servo-Umrichter 9300.

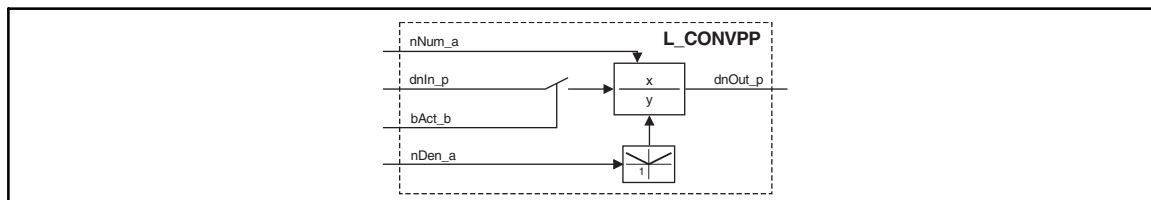


Abb. 2-61

Umrechnung Winkelsignal (L\_CONVPP)

| VariableName | DataType       | SignalType | VariableType | Note                                                                                                      |
|--------------|----------------|------------|--------------|-----------------------------------------------------------------------------------------------------------|
| nNum_a       | Integer        | analog     | VAR_INPUT    | Zähler                                                                                                    |
| dnIn_p       | Double Integer | position   | VAR_INPUT    |                                                                                                           |
| bAct_b       | Bool           | binary     | VAR_INPUT    |                                                                                                           |
| nDen_a       | Integer        | analog     | VAR_INPUT    | Nenner (mit Betragsbildung)                                                                               |
| dnOut_p      | Double Integer | position   | VAR_OUTPUT   | <ul style="list-style-type: none"> <li>Signal wird nicht begrenzt.</li> <li>Restwertbehandlung</li> </ul> |

#### Funktion



#### Stop!

Das Umrechnungsergebnis wird nicht begrenzt. Das Ergebnis darf daher den Zahlenbereich von  $\pm 2147483647$  nicht überschreiten.

Die Umrechnung erfolgt nach der Formel:

- Bei  $bAct\_b = TRUE$

$$dnOut\_p = dnIn\_p \cdot \frac{nNum\_a}{nDen\_a}$$

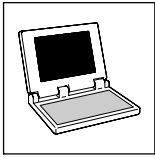
- Bei  $bAct\_b = FALSE$

$$dnOut\_p = \text{Restwert} \cdot \frac{nNum\_a}{nDen\_a}$$



#### Tipp!

- Dieser FB arbeitet mit Restwertbehandlung.
- Der Nenner kann nur  $\geq 1$  werden.
- (INT)65536 entspricht einer Geberumdrehung (eine Geberumdrehung wird in 65536 Inkremente dargestellt).



## Funktionsbibliothek LenzeDrive.lib

### Signalkonvertierung

#### Umrechnung (L\_CONVV)

### 2.5.4 Umrechnung (L\_CONVV)

Dieser FB rechnet ein Signal mit dynamischem Bruch um.

Die Funktion entspricht der des FB CONVPP im Servo-Umrichter 9300.

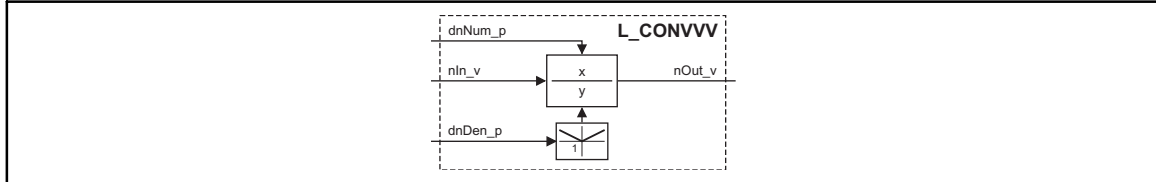


Abb. 2-62

Umrechnung (L\_CONVV)

| VariableName | DataType       | SignalType      | VariableType | Note                                                                                                       |
|--------------|----------------|-----------------|--------------|------------------------------------------------------------------------------------------------------------|
| dnNum_p      | Double Integer | position        | VAR_INPUT    | Zähler                                                                                                     |
| nIn_v        | Integer        | analog/velocity | VAR_INPUT    |                                                                                                            |
| dnDen_p      | Double Integer | position        | VAR_INPUT    | Nenner (mit Betragsbildung)                                                                                |
| nOut_v       | Integer        | analog/velocity | VAR_OUTPUT   | <ul style="list-style-type: none"><li>• Signal wird nicht begrenzt.</li><li>• Restwertbehandlung</li></ul> |

#### Funktion



#### Stop!

Das Umrechnungsergebnis wird nicht begrenzt. Das Ergebnis darf daher den Zahlenbereich von  $\pm 32767$  nicht überschreiten.

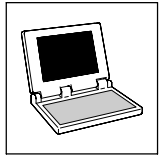
Die Umrechnung erfolgt nach der Formel:

$$nOut_v = nIn_v \cdot \frac{dnNum_p}{dnDen_p}$$



#### Tipp!

- Dieser FB arbeitet mit Restwertbehandlung.
- Der Nenner kann nur  $\geq 1$  werden.
- $16384 \equiv 15000 \text{ rpm}$ .



### 2.5.5 Normierung mit Begrenzung (L\_CONVX)

Dieser FB normiert Signale. Die Umrechnung erfolgt hochgenau mit Restwertverarbeitung mit Vorgabe des Umrechnungsfaktors als Zähler und Nenner.

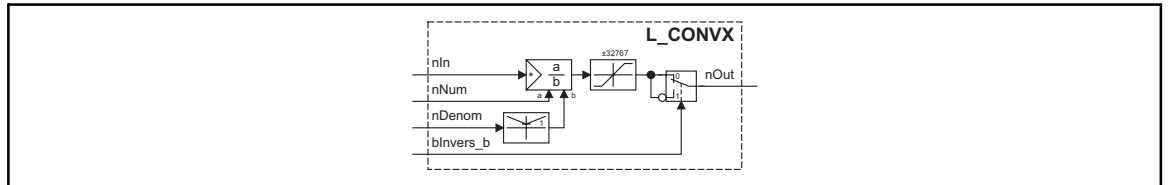


Abb. 2-63

Umrechnung (L\_CONVX)

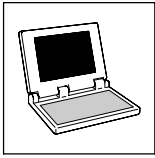
| VariableName | DataType | SignalType      | VariableType | Note   |
|--------------|----------|-----------------|--------------|--------|
| nIn          | Integer  | analog/velocity | VAR_INPUT    |        |
| nNum         | Integer  | analog/velocity | VAR_INPUT    | Zähler |
| nDenom       | Integer  | analog/velocity | VAR_INPUT    | Nenner |
| blnvers_b    | Bool     | binary          | VAR_INPUT    |        |
| nOut         | Integer  | analog/velocity | VAR_OUTPUT   |        |

#### Funktion

Die Multiplikation/Division von Signalen erfolgt nach der Formel:

$$nOut = nIn \cdot \frac{dnNum}{|dnDenom|} \cdot \text{sgn } |blnvers|$$

- Ein Division durch Null wird abgefangen, indem bei  $nDenom = 0$  der Wert von  $nDenom$  auf 1 gesetzt wird.
- Es erfolgt eine Betragsbildung des Eingangswertes von  $nDenominator$ .
- $blnvers\_b = \text{TRUE}$  bewirkt eine Vorzeichenumkehr der Ausgangsgröße  $nOut$ .
- Die Umrechnung erfolgt mit Restwertbehandlung.



# Funktionsbibliothek LenzeDrive.lib

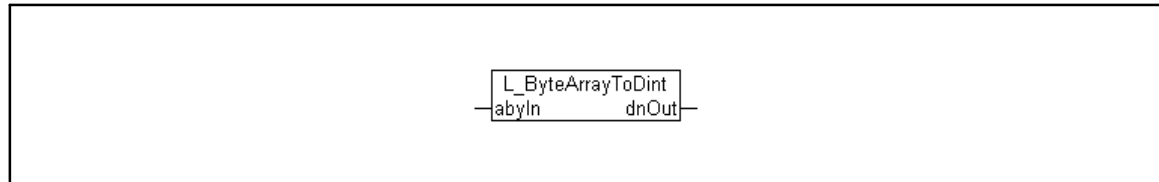
## Kommunikation

### Typ-Konvertierung (L\_ByteArrayToDint)

## 2.6 Kommunikation

### 2.6.1 Typ-Konvertierung (L\_ByteArrayToDint)

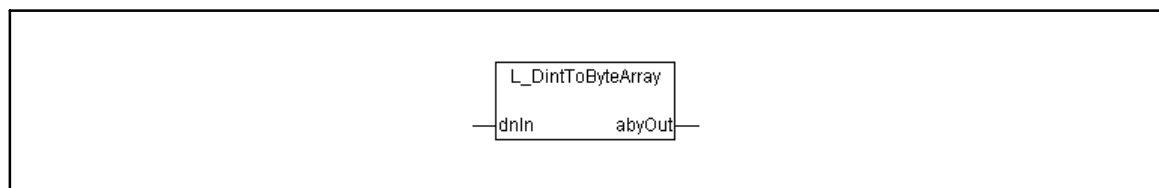
Dieser FB konvertiert ein 4-Byte-Array in eine Variable vom Typ **DINT**.



| VariableName | DataType       | SignalType | VariableType | Note |
|--------------|----------------|------------|--------------|------|
| abyIn        | Byte [0 ... 3] |            | VAR_INPUT    |      |
| dnOut        | Double Integer |            | VAR_OUTPUT   |      |

### 2.6.2 Typ-Konvertierung (L\_DintToByteArray)

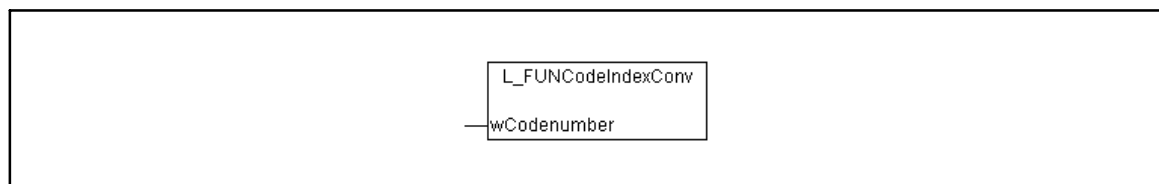
Dieser FB konvertiert eine Variable vom Typ **DINT** in ein 4-Byte-Array, wie es z. B. der FB **L\_ParWrite** für den Eingang **abyData** erwartet.



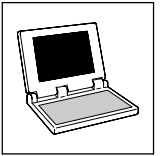
| VariableName | DataType       | SignalType | VariableType | Note |
|--------------|----------------|------------|--------------|------|
| dnIn         | Double Integer |            | VAR_INPUT    |      |
| abyOut       | Byte [0 ... 3] |            | VAR_OUTPUT   |      |

### 2.6.3 Index einer Codestelle (L\_FUNCodeIndexConv)

Diese Funktion überprüft den Wertebereich 1 ... 8000 des Indexes einer Codestelle und liefert den Index z. B. an den FB **L\_ParWrite** über den Eingang **wIndex**. Bei ungültigen Codenummern liefert die Funktion den Index 0000.

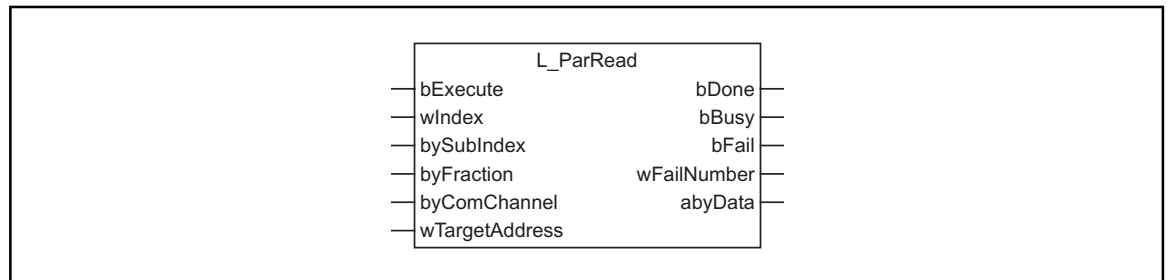


| VariableName | DataType | SignalType | VariableType | Note |
|--------------|----------|------------|--------------|------|
| wCodenummer  | Word     |            | VAR_INPUT    |      |

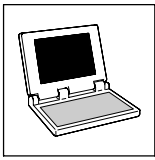


### 2.6.4 Lesen von Codestellen (L\_ParRead)

Dieser FB dient zum Lesen von Parametern, bei Lenze die sogenannten Codestellen. Neben den eigenen Codestellen der PLC können auch Codestellen anderer Geräte über Systembus (CAN) gelesen werden.



| VariableName   | DataType       | VariableType           | Note                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|----------------|----------------|------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| wTimeOut       | Word           | VAR_CONSTANT<br>RETAIN | 1 ... 65535<br>Time-Out-Zeit in ms, in der der Auftrag abgearbeitet werden muss.<br>• Initialisiert mit 1000 ms.<br>• Diese Variable kann mit einer User-Codestelle parametrisiert werden.                                                                                                                                                                                                                                                                                                                                           |
| bExecute       | Bool           | VAR_INPUT              | FALSE-TRUE-Flanke löst Leseauftrag aus.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| wIndex         | Word           | VAR_INPUT              | 0 ... 65535<br>Index der Codestelle<br>Umrechnungsformel: Index = 24575 - Codenummer                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| bySubIndex     | Byte           | VAR_INPUT              | 0 ... 255<br>Subindex (Subcodenummer) der Codestelle                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| byFraction     | Byte           | VAR_INPUT              | 0 ... 254<br>Anzahl der Nachkommastellen der zu lesenden Codestelle<br>255<br>Codestelle ohne Nachkommastellen (z. B. Codestelle im HEX-Format).                                                                                                                                                                                                                                                                                                                                                                                     |
| byComChannel   | Byte           | VAR_INPUT              | 0<br>Konstante: <b>C_PLC</b><br>Codestelle von der PLC lesen.<br>10<br>Konstante: <b>C_SYSTEMBUS_CAN</b><br>Codestelle von einem über Systembus (CAN) angeschlossenen Gerät lesen.<br>11<br>Konstante: <b>C_SYSTEMBUS_CAN</b><br>Nur bei Drive PLC und ECSxA!<br>Codestelle von einem über Systembus (FIF-CAN/CAN-AUX) angeschlossenen Gerät lesen.                                                                                                                                                                                  |
| wTargetAddress | Word           | VAR_INPUT              | Auswahl des Parameterdaten-Kanals beim Zielgerät (nur bei <i>byComChannel</i> = 10 oder 11, s. o.)<br>1 ... 64<br>Die Übertragung erfolgt über SDO1 des Zielgerätes, wobei gilt: <i>wTargetAddress</i> = CAN-Geräteadresse des Zielgerätes<br>65 ... 127<br>Die Übertragung erfolgt über SDO2 des Zielgerätes, wobei gilt: <i>wTargetAddress</i> = CAN-Geräteadresse des Zielgerätes + 64                                                                                                                                            |
| bDone          | Bool           | VAR_OUTPUT             | TRUE<br>Auftrag wurde bearbeitet ( <i>bFail</i> beachten).                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| bBusy          | Bool           | VAR_OUTPUT             | TRUE<br>Auftrag in Bearbeitung.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| bFail          | Bool           | VAR_OUTPUT             | TRUE<br>Fehler aufgetreten.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| wFailNumber    | Word           | VAR_OUTPUT             | 0<br>OK - Leseauftrag wurde fehlerfrei ausgeführt.<br>1<br>Fehler bei der Datenübertragung über Systembus (CAN).<br>2<br>Externes Gerät antwortete nicht innerhalb der eingestellten TimeOut-Zeit.<br>4<br>Subindex existiert nicht.<br>5<br>Index existiert nicht.<br>13<br>Zu lesender Parameterwert liegt außerhalb des gültigen Bereiches.<br>117<br>Kommunikationskanal ( <i>byComChannel</i> ) ist ungültig.<br>118<br>Es stehen nicht genügend freie CAN-Objekte zur Verfügung.<br>119<br>Der Sendeauftragsspeicher ist voll. |
| abyData        | Byte [0 ... 3] | VAR_OUTPUT             | Die vier Datenbytes mit dem gelesenen Wert der Codestelle.                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |



# Funktionsbibliothek LenzeDrive.lib

## Kommunikation

### Lesen von Codestellen (L\_ParRead)



#### Hinweis!

Der FB **L\_ParRead** muss zwingend zyklisch aufgerufen werden, damit auch die Antwort auf den Leseauftrag ("Read Response") erfasst wird. Diese kann nämlich bedingt durch die Zykluszeit des Empfängers erst einige Programmzyklen später wieder bei der PLC eintreffen!

Wird der FB nicht zyklisch aufgerufen (z. B. in einer ereignisgesteuerten Task), so kann dies zu einer "Verklemmung" des FBs führen.

#### Kanalwahl

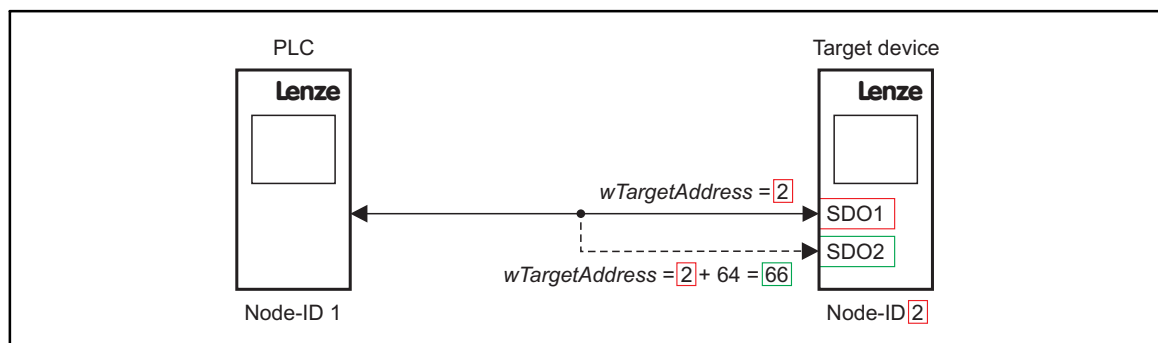
Die Auswahl des Übertragungskanals erfolgt über Codestelle C2118:

| Code  | LCD | Einstellmöglichkeiten |                                   | Info                                                                                                                                                                                                                                                                                                                                                      |
|-------|-----|-----------------------|-----------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|       |     | Lenze                 | Auswahl                           |                                                                                                                                                                                                                                                                                                                                                           |
| C2118 |     | 0                     | 0 PDO-Kanal (CAN1_IO ... CAN3_IO) | Übertragung erfolgt über einen freien PDO-Kanal der PLC.<br>Für diese Auswahl werden benötigt: <ul style="list-style-type: none"><li>Ein freies CAN-Sendeobjekt (CAN1_OUT ... CAN3_OUT) zum Senden des Leseauftrags.</li><li>Ein freies CAN-Empfangsobjekt (CAN1_IN ... CAN3_IN) zum Empfangen der Antwort ("Read Response") vom anderen Gerät.</li></ul> |
|       |     | 1                     | SDO2-Kanal                        | Übertragung erfolgt über den SDO2-Kanal der PLC.                                                                                                                                                                                                                                                                                                          |

#### Auswahl des Busteilnehmers & Parameterdaten-Kanals des Busteilnehmers

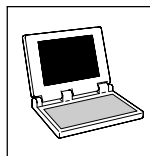
Über *wTargetAddress* erfolgt die Auswahl des entsprechenden Busteilnehmers, bei dem der Codestellenzugriff erfolgen soll. Gleichzeitig erfolgt über *wTargetAddress* auch die Auswahl des zu verwendenden Parameterdaten-Kanals (SDO1 oder SDO2) beim Busteilnehmer:

- Für eine Übertragung über den Parameterdaten-Kanal SDO1 des Busteilnehmers stellen Sie in *wTargetAddress* einfach die entsprechende CAN-Geräteadresse (1 ... 64) des Busteilnehmers ein.
- Für eine Übertragung über den Parameterdaten-Kanal SDO2 des Busteilnehmers stellen Sie in *wTargetAddress* stattdessen die entsprechende CAN-Geräteadresse (1 ... 64) des Busteilnehmers **erhöht um den Wert 64** ein, also einen Wert zwischen 65 und 127.



#### Hinweis!

Stellen Sie sicher, dass auf einen Busteilnehmer nicht zur gleichen Zeit über den gleichen Parameterdaten-Kanal ein Codestellenzugriff von verschiedenen anderen Busteilnehmern erfolgt, da diese "Kollision" dazu führen würde, dass der Systembus in den Zustand "Bus-Off" geht.

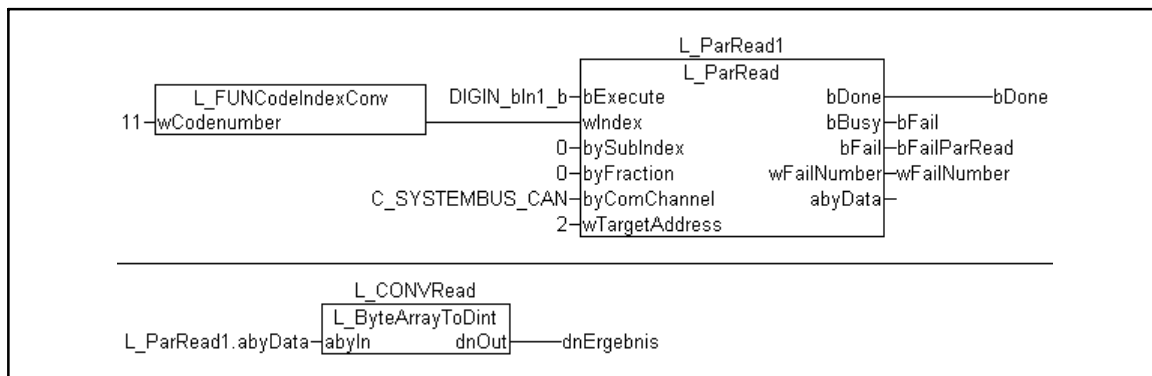


### Tipp!

Allgemeine Informationen zu den CAN-Objekten sowie zum Systembus (CAN) finden Sie in der Dokumentation "Systembus (CAN) bei PLC-Geräten".

### Beispiel

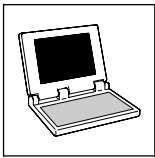
Wert der Codestelle C0011 des Gerätes mit der CAN-Geräteadresse 2 lesen:



### Tipp!

Zum Umrechnen der Codestellennummer in den benötigten Wert für **wIndex** können Sie die Funktion **L\_FUNCodeIndexConv** verwenden (siehe Beispiel).

Wenn Sie den gelesenen Wert im DINT-Format weiterverarbeiten möchten, können Sie zur Konvertierung des 4-Byte-Arrays **abyData** in einen DINT-Wert den FB **L\_DintToByteArray** verwenden (siehe Beispiel).



# Funktionsbibliothek LenzeDrive.lib

## Kommunikation

### Lesen von Codestellen (L\_ParRead)

#### Parameterwerte mit Nachkommastellen



#### Tipp!

Die Parameter der Lenze Geräte sind in unterschiedlichen Formaten abgelegt.

Ausführliche Informationen hierzu finden Sie in der "Attributtabelle" im Systemhandbuch zum entsprechenden Antriebsregler.

Verwendet die zu lesende Codestelle ein Datenformat mit Nachkommastellen, so muss die Anzahl der Nachkommastellen dem Funktionsblock **L\_ParRead** über den Eingang **byFraction** mitgeteilt werden.

Folgende Formate gelten:

| byFraction (Anzahl Nachkommastellen) | Vom FB L_ParRead ausgegebener Wert | Aus Codestelle gelesener Wert |
|--------------------------------------|------------------------------------|-------------------------------|
| 0                                    | 1                                  | 1                             |
| 1                                    | 10                                 | 1.0                           |
| 2                                    | 100                                | 1.00                          |
| 3                                    | 1000                               | 1.000                         |
| 4                                    | 10000                              | 1.0000                        |

#### Beispiel:

Eine Codestelle im Fixed32-Datenformat mit dem Wert "43" soll gelesen werden.

- Fixed32 ist ein Festkommaformat mit 4 Nachkommastellen, für die Übertragung wird der Wert daher mit 10000 multipliziert:

$$Data_{1...4} = 43 \cdot 10000 = 430000 = 00\ 06\ 8F\ B0_{hex}$$

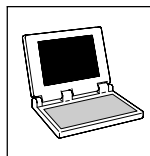
- Am Eingang **byFraction** ist der Wert "4" anzulegen.
- Vom FB **L\_ParRead** wird der Wert "430000" ausgegeben.



#### Tipp!

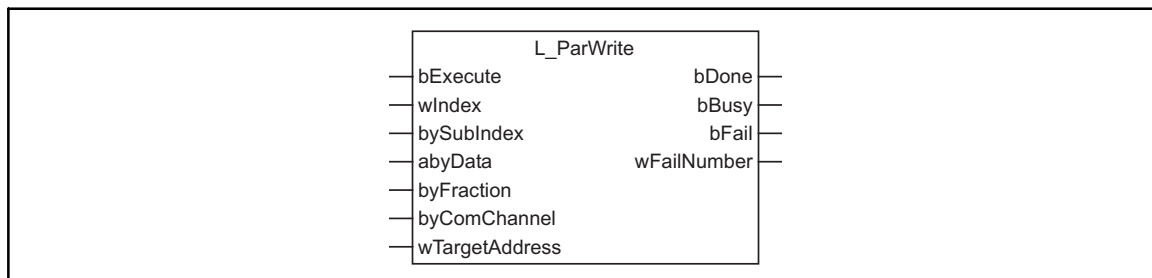
Verwendet die Codestelle nicht das Fixkommaformat, so muss am Eingang **byFraction** der Wert "255" angelegt werden.

Ein ausführliches Beispiel zum Lesen und Schreiben von Parametern über Systembus (CAN) finden Sie in der Dokumentation "Systembus (CAN) bei PLC-Geräten".

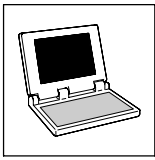


#### 2.6.5 Beschreiben von Codestellen (L\_ParWrite)

Dieser FB dient zum Beschreiben von Parametern, bei Lenze die sogenannten Codestellen. Neben den eigenen Codestellen der PLC können auch Codestellen anderer Geräte über Systembus (CAN) beschrieben werden.



| VariableName   | DataType       | VariableType           | Note                                                                                                                                                                                                                                                                                                                                                                                      |
|----------------|----------------|------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| wTimeOut       | Word           | VAR_CONSTANT<br>RETAIN | 1 ... 65535<br>Time-Out-Zeit in ms, in der der Auftrag abgearbeitet werden muss.<br><ul style="list-style-type: none"> <li>Initialisiert mit 1000 ms.</li> <li>Diese Variable kann mit einer User-Codestelle parametrisiert werden.</li> </ul>                                                                                                                                            |
| bExecute       | Bool           | VAR_INPUT              | FALSE-TRUE-Flanke löst Schreibauftrag aus.                                                                                                                                                                                                                                                                                                                                                |
| wIndex         | Word           | VAR_INPUT              | 0 ... 65535<br>Index der Codestelle<br>Umrechnungsformel: Index = 24575 - Codenummer                                                                                                                                                                                                                                                                                                      |
| bySubIndex     | Byte           | VAR_INPUT              | 0 ... 255<br>Subindex (Subcodenummer) der Codestelle                                                                                                                                                                                                                                                                                                                                      |
| abyData        | Byte [0 ... 3] | VAR_INPUT              | Die vier Datenbytes mit dem zu schreibenden Wert der Codestelle.                                                                                                                                                                                                                                                                                                                          |
| byFraction     | Byte           | VAR_INPUT              | 0 ... 254<br>Anzahl der Nachkommastellen der zu beschreibenden Codestelle.<br>255<br>Codestelle ohne Nachkommastellen (z. B. Codestelle im HEX-Format).                                                                                                                                                                                                                                   |
| byComChannel   | Byte           | VAR_INPUT              | 0<br>Konstante: <b>C_PLC</b><br>Codestelle in der PLC beschreiben.<br>10<br>Konstante: <b>C_SYSTEMBUS_CAN</b><br>Codestelle in einem über Systembus (CAN) angeschlossenen Gerät beschreiben.<br>11<br>Konstante: <b>C_SYSTEMBUS_CAN</b><br>Nur bei Drive PLC und ECSxA!<br>Codestelle von einem über Systembus (FIF-CAN/CAN-AUX) angeschlossenen Gerät lesen.                             |
| wTargetAddress | Word           | VAR_INPUT              | Auswahl des Parameterdaten-Kanals beim Zielgerät (nur bei <i>byComChannel</i> = 10 oder 11, s. o.)<br>1 ... 64<br>Die Übertragung erfolgt über SDO1 des Zielgerätes, wobei gilt: <i>wTargetAddress</i> = CAN-Geräteadresse des Zielgerätes<br>65 ... 127<br>Die Übertragung erfolgt über SDO2 des Zielgerätes, wobei gilt: <i>wTargetAddress</i> = CAN-Geräteadresse des Zielgerätes + 64 |
| bDone          | Bool           | VAR_OUTPUT             | TRUE<br>Auftrag wurde bearbeitet ( <i>bFail</i> beachten).                                                                                                                                                                                                                                                                                                                                |
| bBusy          | Bool           | VAR_OUTPUT             | TRUE<br>Auftrag in Bearbeitung.                                                                                                                                                                                                                                                                                                                                                           |
| bFail          | Bool           | VAR_OUTPUT             | TRUE<br>Fehler aufgetreten.                                                                                                                                                                                                                                                                                                                                                               |



# Funktionsbibliothek LenzeDrive.lib

## Kommunikation

### Beschreiben von Codestellen (L\_ParWrite)

| VariableName | DataType | VariableType | Note |                                                                                                                       |
|--------------|----------|--------------|------|-----------------------------------------------------------------------------------------------------------------------|
| wFailNumber  | Word     | VAR_OUTPUT   | 0    | OK - Schreibauftrag wurde fehlerfrei ausgeführt.                                                                      |
|              |          |              | 1    | Fehler bei der Datenübertragung über Systembus (CAN).                                                                 |
|              |          |              | 2    | Externes Gerät antwortete nicht innerhalb der eingestellten TimeOut-Zeit.                                             |
|              |          |              | 4    | Subindex existiert nicht.                                                                                             |
|              |          |              | 5    | Index existiert nicht.                                                                                                |
|              |          |              | 7    | Im Zielgerät ist die Reglersperre nicht gesetzt, die für das Beschreiben der angegebenen Codestelle erforderlich ist. |
|              |          |              | 13   | Zu schreibender Parameterwert liegt außerhalb des gültigen Bereiches.                                                 |
|              |          |              | 117  | Kommunikationskanal (byComChannel) ist ungültig.                                                                      |
|              |          |              | 118  | Es stehen nicht genügend freie CAN-Objekte zur Verfügung.                                                             |
|              |          |              | 119  | Der Sendeauftragsspeicher ist voll.                                                                                   |



### Hinweis!

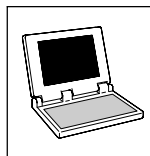
Der FB **L\_ParWrite** muss zwingend zyklisch aufgerufen werden, damit auch die Quittierung des Schreibauftrags ("Write Response") erfasst wird. Diese kann nämlich bedingt durch die Zykluszeit des Empfängers erst einige Programmzyklen später wieder bei der PLC eintreffen!

Wird der FB nicht zyklisch aufgerufen (z. B. in einer ereignisgesteuerten Task), so kann dies zu einer "Verklemmung" des FBs führen.

### Kanalwahl

Die Auswahl des Übertragungskanals erfolgt über Codestelle C2118:

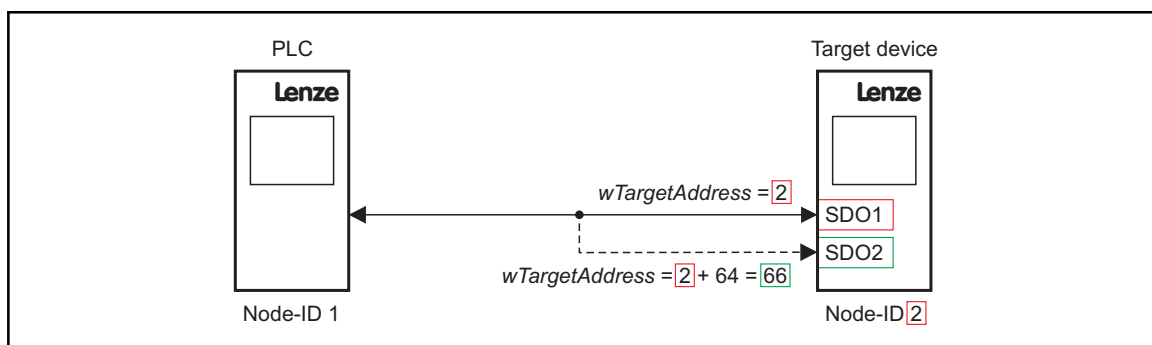
| Code  | LCD | Einstellmöglichkeiten |                                   | Info                                                                                                                                                                                                                                                                                                                                                                    |
|-------|-----|-----------------------|-----------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|       |     | Lenze                 | Auswahl                           |                                                                                                                                                                                                                                                                                                                                                                         |
| C2118 |     | 0                     | 0 PDO-Kanal (CAN1_IO ... CAN3_IO) | Übertragung erfolgt über einen freien PDO-Kanal der PLC.<br>Für diese Auswahl werden benötigt:<br><ul style="list-style-type: none"> <li>Ein freies CAN-Sendeobjekt (CAN1_OUT ... CAN3_OUT) zum Senden des Schreibauftrags.</li> <li>Ein freies CAN-Empfangsobjekt (CAN1_IN ... CAN3_IN) zum Empfangen der Quittierung ("Write Response") vom anderen Gerät.</li> </ul> |
|       |     |                       | 1 SD02-Kanal                      | Übertragung erfolgt über den SD02-Kanal der PLC.                                                                                                                                                                                                                                                                                                                        |



### Auswahl des Busteilnehmers & Parameterdaten-Kanals des Busteilnehmers

Über *wTargetAddress* erfolgt die Auswahl des entsprechenden Busteilnehmers, bei dem der Codestellenzugriff erfolgen soll. Gleichzeitig erfolgt über *wTargetAddress* auch die Auswahl des zu verwendenden Parameterdaten-Kanals (SDO1 oder SDO2) beim Busteilnehmer:

- Für eine Übertragung über den Parameterdaten-Kanal SDO1 des Busteilnehmers stellen Sie in *wTargetAddress* einfach die entsprechende CAN-Geräteadresse (1 ... 64) des Busteilnehmers ein.
- Für eine Übertragung über den Parameterdaten-Kanal SDO2 des Busteilnehmers stellen Sie in *wTargetAddress* stattdessen die entsprechende CAN-Geräteadresse (1 ... 64) des Busteilnehmers **erhöht um den Wert 64** ein, also einen Wert zwischen 65 und 127.



### Hinweis!

Stellen Sie sicher, dass auf einen Busteilnehmer nicht zur gleichen Zeit über den gleichen Parameterdaten-Kanal ein Codestellenzugriff von verschiedenen anderen Busteilnehmern erfolgt, da diese "Kollision" dazu führen würde, dass der Systembus in den Zustand "Bus-Off" geht.

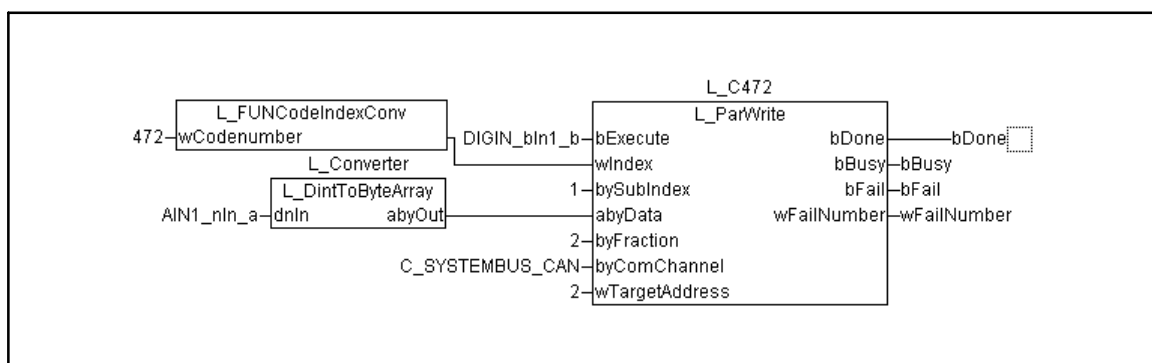


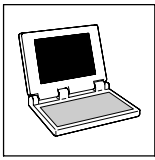
### Tipp!

Allgemeine Informationen zu den CAN-Objekten sowie zum Systembus (CAN) finden Sie in der Dokumentation "Systembus (CAN) bei PLC-Geräten".

### Beispiel

Wert des analogen Eingangs AIN1 (Klemme 1/2 bei 9300 Servo PLC) auf die Codestelle C0472/1 des Gerätes mit der CAN-Geräteadresse 2 übertragen:





# Funktionsbibliothek LenzeDrive.lib

## Kommunikation

### Beschreiben von Codestellen (L\_ParWrite)



#### Tipp!

Zum Umrechnen der Codestellennummer in den benötigten Wert für **wIndex** können Sie die Funktion **L\_FUNCodeIndexConv** verwenden (siehe Beispiel).

Wenn Sie den zu schreibenden Wert im DINT-Format vorgeben möchten, können Sie zur Konvertierung des DINT-Wertes in das für **abyData** benötigte 4-Byte-Array den FB **L\_DintToByteArray** verwenden (siehe Beispiel).

#### Parameterwerte mit Nachkommastellen



#### Tipp!

Die Parameter der Lenze Geräte sind in unterschiedlichen Formaten abgelegt.

Ausführliche Informationen hierzu finden Sie in der "Attributtabelle" im Systemhandbuch zum entsprechenden Antriebsregler.

Verwendet die zu beschreibende Codestelle ein Datenformat mit Nachkommastellen, so muss die Anzahl der Nachkommastellen dem Funktionsblock **L\_ParWrite** über den Eingang **byFraction** mitgeteilt werden.

Folgende Formate gelten:

| byFraction (Anzahl Nachkommastellen) | Am FB L_ParWrite anliegender Wert | In Codestelle zu schreibender Wert |
|--------------------------------------|-----------------------------------|------------------------------------|
| 0                                    | 1                                 | 1                                  |
| 1                                    | 10                                | 1.0                                |
| 2                                    | 100                               | 1.00                               |
| 3                                    | 1000                              | 1.000                              |
| 4                                    | 10000                             | 1.0000                             |

Beispiel:

Für eine Codestelle im Fixed32-Datenformat soll der Wert "20" übertragen werden.

- Fixed32 ist ein Festkommaformat mit 4 Nachkommastellen, für die Übertragung wird der Wert daher mit 10000 multipliziert:

$$Data_{1...4} = 20 \cdot 10000 = 200000 = 00\ 03\ 0D\ 40_{hex}$$

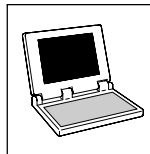
- Am Eingang **byFraction** ist der Wert "4" anzulegen.
- In die Codestelle wird der Wert "20.0000" geschrieben.



#### Tipp!

Verwendet die Codestelle nicht das Fixkommaformat, so muss am Eingang **byFraction** der Wert "255" angelegt werden.

Ein ausführliches Beispiel zum Lesen und Schreiben von Parametern über Systembus (CAN) finden Sie in der Dokumentation "Systembus (CAN) bei PLC-Geräten".



## 2.7 Spezielle Funktionen

### 2.7.1 Transparenter Modus mit Bedieneinheit 9371BB/9371BC (L\_Display9371BB)

Mit der **Bedieneinheit 9371BB und 9371BC** können über eine Tastatur Parameter (z. B. Sollwerte) vorgegeben, Betriebsdaten angezeigt und Parametersätze zu anderen Zielsystemen transferiert werden.

Mit Hilfe dieses FBs kann die Bedieneinheit in einen sogenannten "transparenten" Modus geschaltet werden, so dass vom Programm aus auf alle Anzeigelemente und Tasten der Bedieneinheit 9371BB/9371BC zugegriffen werden kann.



#### Hinweis!

- Der FB muss aufgrund seiner internen Struktur in einer zeitäquidistanten Task (30 ... 50 ms) aufgerufen werden.
- Nur bei Änderungen an den FB-Eingängen werden Daten an die Bedieneinheit gesendet.

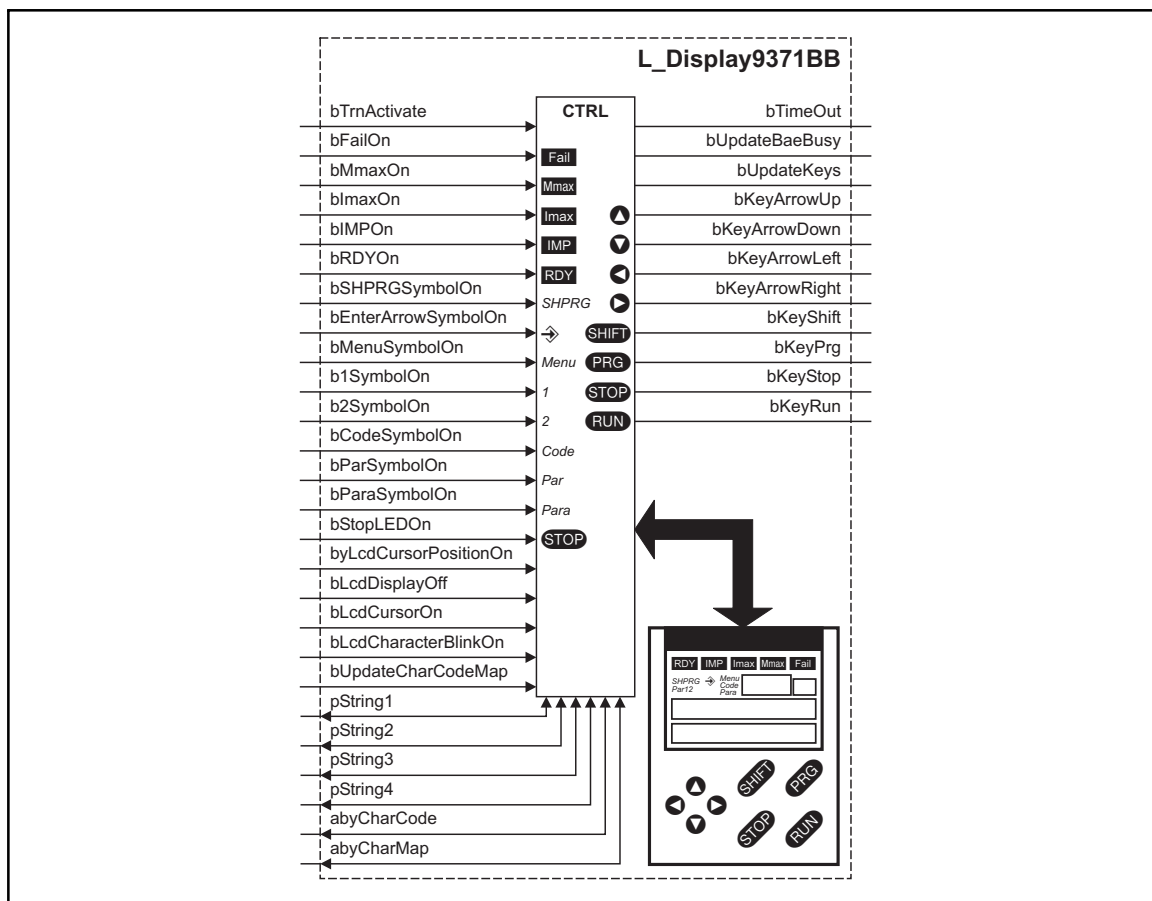
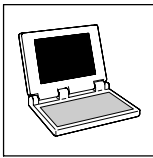


Abb. 2-64

FB L\_Display9371BB



# Funktionsbibliothek LenzDrive.lib

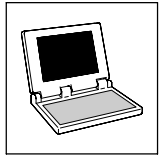
## Spezielle Funktionen

### Transparenter Modus mit Bedieneinheit 9371BB/9371BC (L\_Display9371BB)

| VariableName          | DataType      | SignalType | VariableType | Note                                                                                                                                                                                                                                       |
|-----------------------|---------------|------------|--------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| bTrnActivate          | Bool          | binary     | VAR_INPUT    | TRUE: Transparenten Modus aktivieren.                                                                                                                                                                                                      |
| bFailOn               | Bool          | binary     | VAR_INPUT    | TRUE: Auf dem Display wird <b>Fail</b> angezeigt.                                                                                                                                                                                          |
| bMmaxOn               | Bool          | binary     | VAR_INPUT    | TRUE: Auf dem Display wird <b>Mmax</b> angezeigt.                                                                                                                                                                                          |
| blmaxOn               | Bool          | binary     | VAR_INPUT    | TRUE: Auf dem Display wird <b>lmax</b> angezeigt.                                                                                                                                                                                          |
| bIMPOn                | Bool          | binary     | VAR_INPUT    | TRUE: Auf dem Display wird <b>IMP</b> angezeigt.                                                                                                                                                                                           |
| brDYOn                | Bool          | binary     | VAR_INPUT    | TRUE: Auf dem Display wird <b>RDY</b> angezeigt.                                                                                                                                                                                           |
| bSHPRGSymbolOn        | Bool          | binary     | VAR_INPUT    | TRUE: Auf dem Display wird "SHPRG" angezeigt.                                                                                                                                                                                              |
| bEnterArrowSymbolOn   | Bool          | binary     | VAR_INPUT    | TRUE: Auf dem Display wird  angezeigt.                                                                                                                                                                                                     |
| bMenuSymbolOn         | Bool          | binary     | VAR_INPUT    | TRUE: Auf dem Display wird "Menu" angezeigt.                                                                                                                                                                                               |
| b1SymbolOn            | Bool          | binary     | VAR_INPUT    | TRUE: Auf dem Display wird "1" angezeigt.                                                                                                                                                                                                  |
| b2SymbolOn            | Bool          | binary     | VAR_INPUT    | TRUE: Auf dem Display wird "2" angezeigt.                                                                                                                                                                                                  |
| bCodeSymbolOn         | Bool          | binary     | VAR_INPUT    | TRUE: Auf dem Display wird "Code" angezeigt.                                                                                                                                                                                               |
| bParSymbolOn          | Bool          | binary     | VAR_INPUT    | TRUE: Auf dem Display wird "Par" angezeigt.                                                                                                                                                                                                |
| bParaSymbolOn         | Bool          | binary     | VAR_INPUT    | TRUE: Auf dem Display wird "Para" angezeigt.                                                                                                                                                                                               |
| bStopLEDOn            | Bool          | binary     | VAR_INPUT    | TRUE: Die Taste <b>STOP</b> leuchtet.                                                                                                                                                                                                      |
| byLcdCursorPositionOn | Bool          | binary     | VAR_INPUT    | Vorgabe der Cursor-Position (0 ... 30) <ul style="list-style-type: none"> <li>• String 1: Position 0 ... 3</li> <li>• String 2: Position 4 ... 5</li> <li>• String 3: Position 6 ... 17</li> <li>• String 4: Position 18 ... 30</li> </ul> |
| bLcdDisplayOff        | Bool          | binary     | VAR_INPUT    | TRUE: Display ist ausgeschaltet.                                                                                                                                                                                                           |
| bLcdCursorOn          | Bool          | binary     | VAR_INPUT    | TRUE: Cursor ist aktiv geschaltet.                                                                                                                                                                                                         |
| bLcdCharacterBlinkOn  | Bool          | binary     | VAR_INPUT    | TRUE: Das Zeichen auf der Cursor-Position blinkt.                                                                                                                                                                                          |
| bUpdateCharCodeMap    | Bool          | binary     | VAR_INPUT    | TRUE: Die Eingänge <i>abyCharCode</i> und <i>abyCharMap</i> mit der Sonderzeichen-Definition werden erneut eingelesen.                                                                                                                     |
| pString1              | String        | -          | VAR_IN_OUT   | Zeichenkette 1, Länge: 4 Zeichen                                                                                                                                                                                                           |
| pString2              | String        | -          | VAR_IN_OUT   | Zeichenkette 2, Länge: 2 Zeichen                                                                                                                                                                                                           |
| pString3              | String        | -          | VAR_IN_OUT   | Zeichenkette 3, Länge: 12 Zeichen                                                                                                                                                                                                          |
| pString4              | String        | -          | VAR_IN_OUT   | Zeichenkette 4, Länge: 13 Zeichen                                                                                                                                                                                                          |
| abyCharCode           | Array of Byte | -          | VAR_IN_OUT   | Zugewiesener ASCII-Code für Sonderzeichen 0 ... 6                                                                                                                                                                                          |
| abyCharMap            | Array of Byte | -          | VAR_IN_OUT   | Definition der Sonderzeichen 0 ... 6                                                                                                                                                                                                       |
| bTimeOut              | Bool          | binary     | VAR_OUTPUT   | TRUE: Die Bedieneinheit wurde von der PLC abgezogen und die TimeOut-Zeit ist abgelaufen.                                                                                                                                                   |
| bUpdateBaeBusy        | Bool          | binary     | VAR_OUTPUT   | TRUE: Die Anzeige der Bedieneinheit wird aktualisiert, es sollte keine Änderung der Eingänge erfolgen.                                                                                                                                     |
| bUpdateKeys           | Bool          | binary     | VAR_OUTPUT   | TRUE: Auf der Bedieneinheit werden Tasten betätigt.                                                                                                                                                                                        |
| bKeyArrowUp           | Bool          | binary     | VAR_OUTPUT   | TRUE: Auf der Bedieneinheit wird die Taste  betätigt.                                                                                                                                                                                      |
| bKeyArrowDown         | Bool          | binary     | VAR_OUTPUT   | TRUE: Auf der Bedieneinheit wird die Taste  betätigt.                                                                                                                                                                                      |
| bKeyArrowLeft         | Bool          | binary     | VAR_OUTPUT   | TRUE: Auf der Bedieneinheit wird die Taste  betätigt.                                                                                                                                                                                      |
| bKeyArrowRight        | Bool          | binary     | VAR_OUTPUT   | TRUE: Auf der Bedieneinheit wird die Taste  betätigt.                                                                                                                                                                                      |
| bKeyShift             | Bool          | binary     | VAR_OUTPUT   | TRUE: Auf der Bedieneinheit wird die Taste <b>SHIFT</b> betätigt.                                                                                                                                                                          |
| bKeyPrg               | Bool          | binary     | VAR_OUTPUT   | TRUE: Auf der Bedieneinheit wird die Taste <b>PRG</b> betätigt.                                                                                                                                                                            |
| bKeyStop              | Bool          | binary     | VAR_OUTPUT   | TRUE: Auf der Bedieneinheit wird die Taste <b>STOP</b> betätigt.                                                                                                                                                                           |
| bKeyRun               | Bool          | binary     | VAR_OUTPUT   | TRUE: Auf der Bedieneinheit wird die Taste <b>RUN</b> betätigt.                                                                                                                                                                            |

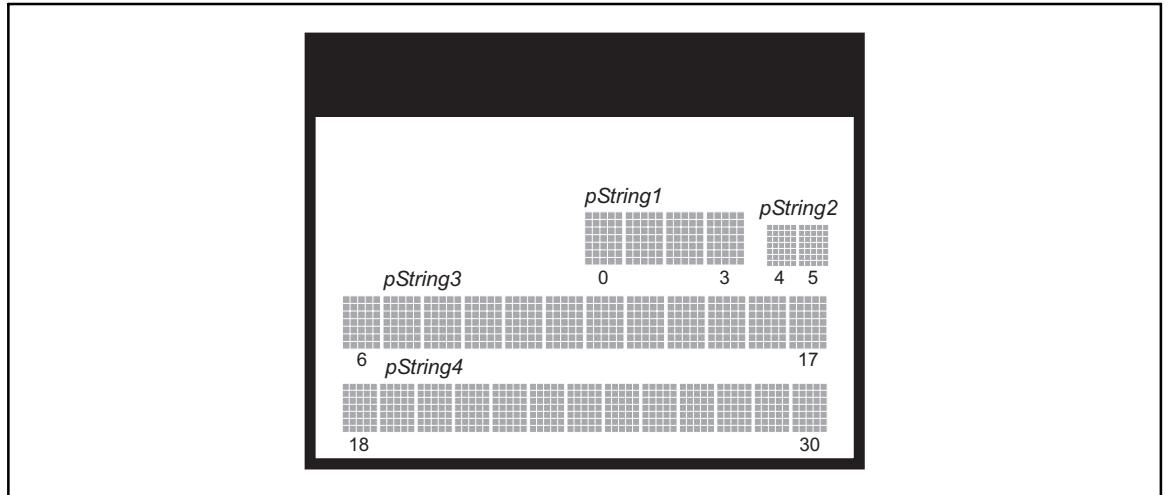
### Transparenten Modus aktivieren

Durch Setzen von *bTrnActivate* auf TRUE wird der transparente Modus in der Bedieneinheit aktiviert, so dass über den FB auf alle Anzeigelemente und Tasten der Bedieneinheit 9371BB/9371BC zugegriffen werden kann.

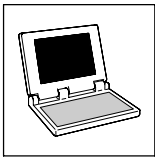


#### Anzeige von ASCII-Zeichen auf der Bedieneinheit

Die Bedieneinheit verfügt über vier Felder zur Anzeige von ASCII-Zeichenketten:



- Über *pString1* ... *pString4* geben Sie die anzuzeigenden ASCII-Zeichenketten vor.
- Über *byLcdCursorPositionOn* können Sie die Position des Cursors (0 ... 30) vorgeben.
- Durch Setzen von *bLcdCursorOn* auf TRUE wird der Cursor angezeigt.
- Durch Setzen von *bLcdCharacterBlinkOn* auf TRUE wird das Zeichen auf der Cursor-Position blinkend angezeigt.



# Funktionsbibliothek LenzeDrive.lib

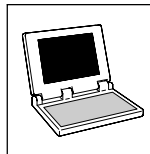
## Spezielle Funktionen

Transparenter Modus mit Bedieneinheit 9371BB/9371BC (L\_Display9371BB)

### ASCII-Zeichentabelle

Die folgende Tabelle enthält die Zeichen des ASCII-Zeichensatzes, die auf der Bedieneinheit angezeigt werden können:

| HEX | DEC | CHAR    | HEX | DEC | CHAR | HEX | DEC | CHAR |
|-----|-----|---------|-----|-----|------|-----|-----|------|
| 20  | 32  | [Space] | 40  | 64  | @    | 60  | 96  | `    |
| 21  | 33  | !       | 41  | 65  | A    | 61  | 97  | a    |
| 22  | 34  | "       | 42  | 66  | B    | 62  | 98  | b    |
| 23  | 35  | #       | 43  | 67  | C    | 63  | 99  | c    |
| 24  | 36  | \$      | 44  | 68  | D    | 64  | 100 | d    |
| 25  | 37  | %       | 45  | 69  | E    | 65  | 101 | e    |
| 26  | 38  | &       | 46  | 70  | F    | 66  | 102 | f    |
| 27  | 39  | '       | 47  | 71  | G    | 67  | 103 | g    |
| 28  | 40  | (       | 48  | 72  | H    | 68  | 104 | h    |
| 29  | 41  | )       | 49  | 73  | I    | 69  | 105 | i    |
| 2A  | 42  | *       | 4A  | 74  | J    | 6A  | 106 | j    |
| 2B  | 43  | +       | 4B  | 75  | K    | 6B  | 107 | k    |
| 2C  | 44  | ,       | 4C  | 76  | L    | 6C  | 108 | l    |
| 2D  | 45  | -       | 4D  | 77  | M    | 6D  | 109 | m    |
| 2E  | 46  | .       | 4E  | 78  | N    | 6E  | 100 | n    |
| 2F  | 47  | /       | 4F  | 79  | O    | 6F  | 111 | o    |
| 30  | 48  | 0       | 50  | 80  | P    | 70  | 112 | p    |
| 31  | 49  | 1       | 51  | 81  | Q    | 71  | 113 | q    |
| 32  | 50  | 2       | 52  | 82  | R    | 72  | 114 | r    |
| 33  | 51  | 3       | 53  | 83  | S    | 73  | 115 | s    |
| 34  | 52  | 4       | 54  | 84  | T    | 74  | 116 | t    |
| 35  | 53  | 5       | 55  | 85  | U    | 75  | 117 | u    |
| 36  | 54  | 6       | 56  | 86  | V    | 76  | 118 | v    |
| 37  | 55  | 7       | 57  | 87  | W    | 77  | 119 | w    |
| 38  | 56  | 8       | 58  | 88  | X    | 78  | 120 | x    |
| 39  | 57  | 9       | 59  | 89  | Y    | 79  | 121 | y    |
| 3A  | 58  | :       | 5A  | 90  | Z    | 7A  | 122 | z    |
| 3B  | 59  | ;       | 5B  | 91  | [    | 7B  | 123 | {    |
| 3C  | 60  | <       | 5C  | 92  | ¥    | 7C  | 124 |      |
| 3D  | 61  | =       | 5D  | 93  | ]    | 7D  | 125 | }    |
| 3E  | 62  | >       | 5E  | 94  | ^    | 7E  | 126 | →    |
| 3F  | 63  | ?       | 5F  | 95  | _    | 7F  | 127 | ←    |



#### Darstellung und Definition von Sonderzeichen

Der FB ermöglicht die Definition von bis zu 7 Sonderzeichen, wobei jedem dieser Sonderzeichen ein individueller ASCII-Code zugewiesen werden kann.

- Die 5 x 7 Matrix eines Zeichens wird über die Array-Variable *abyCharMap* definiert, wobei jedes Array-Element vom Datentyp Byte genau eine Spalte der Matrix bitcodiert definiert (0 = weißes Pixel, 1 = schwarzes Pixel).
- Über die Array-Variable *abyCharCode* wird der ASCII-Code für das entsprechende Sonderzeichen festgelegt.

Beispiel:

Als zweites Sonderzeichen soll ein "µ" mit ASCII-Code "200" definiert werden:

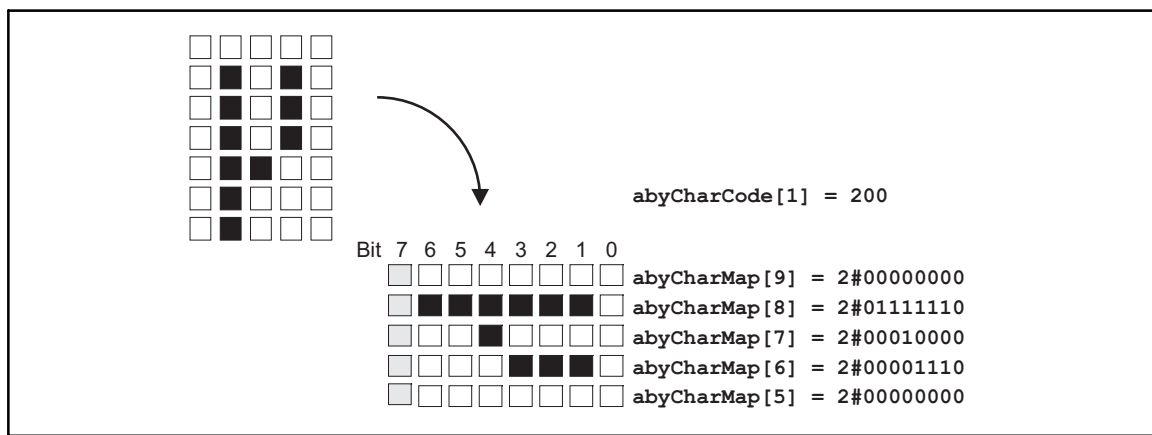


Abb. 2-65

Beispiel: Definition der Matrix eines Sonderzeichens

Die folgende Abbildung verdeutlicht nochmal die Zuordnung von Array-Index zu Matrix-Spalte:

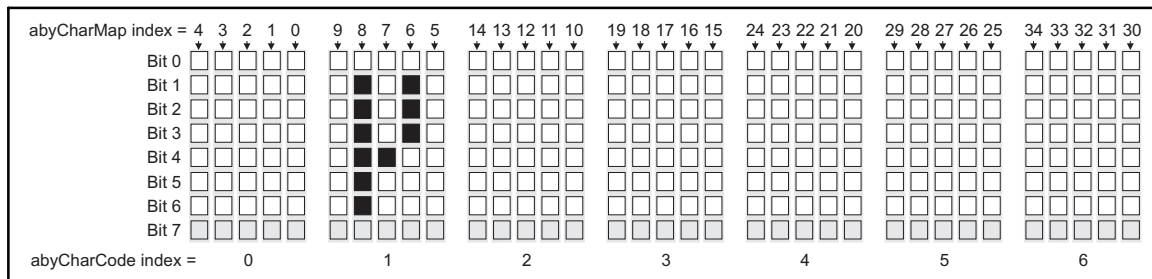
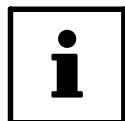


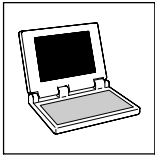
Abb. 2-66

Zuordnung von Array-Index [0...30] zu Matrix-Spalte



#### Hinweis!

- Wird ein Sonderzeichen mit einem ASCII-Code zwischen 32 und 127 definiert, so wird das Sonderzeichen statt dem entsprechenden Standard-ASCII-Zeichen angezeigt.
- Die definierten Sonderzeichen werden in die Zeichenketten *pString1* ... *pString4* nur übernommen, wenn *bUpdateCharCodeMap* auf TRUE gesetzt ist.



# Funktionsbibliothek LenzeDrive.lib

## Spezielle Funktionen

### Fehlerrückmeldung (L\_FWM)

## 2.7.2

### Fehlerrückmeldung (L\_FWM)



#### Hinweis!

Der FB **L\_FWM** der Funktionsbibliothek **LenzeDrive0201.lib** ist überarbeitet worden und kann für folgende Lenze PLCs verwendet werden:

- 9300 Servo PLC EVS93XX-xl - Version 7.0
- 9300 Servo PLC EVS93XX-xT - Version 7.0
- Drive PLC EPL10200 - Version 7.2
- ECSxAxxx - Version 7.0

Die Verwendung des FBs **L\_FWM** aus älteren LenzeDrive.lib-Versionen sollte ersetzt werden, oder nur nach Rücksprache mit Lenze verwendet werden!

Der FB **L\_FWM** dient zur Übermittlung von Fehlermeldungen an die PLC. Dadurch kann aus dem SPS-Programmablauf heraus TRIP, FAIL-QSP, Meldung oder Warnung in der PLC ausgelöst werden.

Die übertragene Fehlernummer wird im Historienspeicher der PLC (C0168/x) gespeichert.

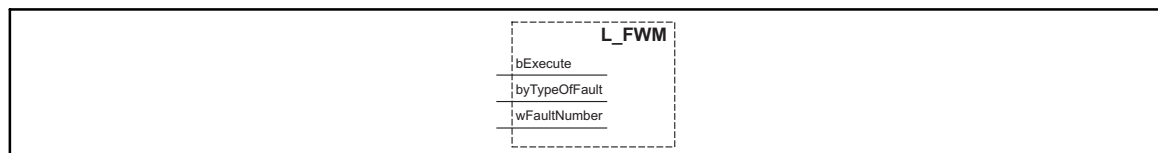
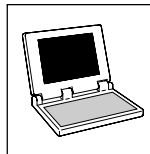


Abb. 2-67

Fehlerrückmeldung (L\_FWM)

| VariableName  | DataType | SignalType | VariableType | Note                                                                                                                              |
|---------------|----------|------------|--------------|-----------------------------------------------------------------------------------------------------------------------------------|
| bExecute      | Bool     |            | VAR_INPUT    | High-Flanke löst Fehler aus                                                                                                       |
| byTypeOfFault | Byte     |            | VAR_INPUT    | Art des auszulösenden Fehlers<br>0: TRIP<br>1: Meldung<br>2: Warnung<br>3: ausgeschaltet<br>4: FAIL-QSP                           |
| wFaultNumber  | Word     |            | VAR_INPUT    | Fehlernummer (400-999)<br>Die Fehlernummer wird im Historienspeicher der PLC (C0168/x) mit dem Offset des Fehlertyps gespeichert. |



|      |                                                                               |
|------|-------------------------------------------------------------------------------|
| 0001 | PROGRAM PLC_PRG                                                               |
| 0002 | VAR                                                                           |
| 0003 | L_FWM1: L_FWM;                                                                |
| 0004 | L_FWM2: L_FWM;                                                                |
| 0005 | L_FWM3: L_FWM;                                                                |
| 0006 | L_FWM4: L_FWM;                                                                |
| 0007 | L_FWM5: L_FWM;                                                                |
| 0008 | END_VAR                                                                       |
| 0001 | (*Trip*)                                                                      |
| 0002 | CAL L_FWM1(bExecute := DIGIN_bIn1_b, byTypeOfFault := 0, wFaultNumber := 450) |
| 0003 |                                                                               |
| 0004 | (*Meldung*)                                                                   |
| 0005 | CAL L_FWM2(bExecute := DIGIN_bIn2_b, byTypeOfFault := 1, wFaultNumber := 451) |
| 0006 |                                                                               |
| 0007 | (*Warnung*)                                                                   |
| 0008 | CAL L_FWM3(bExecute := DIGIN_bIn3_b, byTypeOfFault := 2, wFaultNumber := 452) |
| 0009 |                                                                               |
| 0010 | (*Fail-QSP*)                                                                  |
| 0011 | CAL L_FWM5(bExecute := DIGIN_bIn5_b, byTypeOfFault := 4, wFaultNumber := 454) |
| 0012 |                                                                               |
| 0013 | (*Keine Fehlerrückmeldung*)                                                   |
| 0014 | CAL L_FWM4(bExecute := DIGIN_bIn4_b, byTypeOfFault := 3, wFaultNumber := 453) |

Abb. 2-68

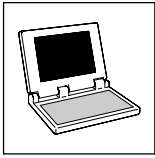
Konfiguration für die Auslösung der verschiedenen Fehlerarten



### Hinweis!

Es darf eine Instanz eines Funktionsblocks immer nur einmal aufgerufen werden! Benötigen Sie den FB **L\_FWM** mehrfach, so legen Sie entsprechende Instanzen des FBs an.

- Existieren im Programm mehrere Instanzen des FB **L\_FWM** gleichen Fehlertyps, wird nur der erste erkannte Fehler dieses Fehlertyps eingetragen.
- Beim gleichzeitigen Auftreten mehrerer Störungen mit unterschiedlicher Reaktion ist im Fehlerspeicher nur die Störung eingetragen, deren Reaktion die höchste Priorität hat:  
– TRIP (höchste) → Meldung → FAIL-QSP → Warnung (niedrigste).
- L\_FWM() nur in einer Task bearbeiten.
- Besitzt *byTypeOfFault* einen ungültigen Wert, wird für *byTypeOfFault* der Wert 0 (TRIP) verwendet.
- Ist der Wert von *wFaultNumber* < 400, wird für *wFaultNumber* der Wert 400 verwendet, ist *wFaultNumber* > 999, wird für *wFaultNumber* der Wert 999 verwendet.
- Steht ein Fehler an, kann dieser aktuelle Fehler mit Hilfe der Systemvariable **DCTRL\_wFaultNumber** ausgelesen werden.



# Funktionsbibliothek LenzeDrive.lib

## Spezielle Funktionen

### Motorpotentiometer (L\_MPOT)

## 2.7.3 Motorpotentiometer (L\_MPOT)

Dieser FB ersetzt ein Hardware-Motorpotentiometer. Er dient als alternative Sollwertquelle, die über zwei Eingänge gesteuert wird.

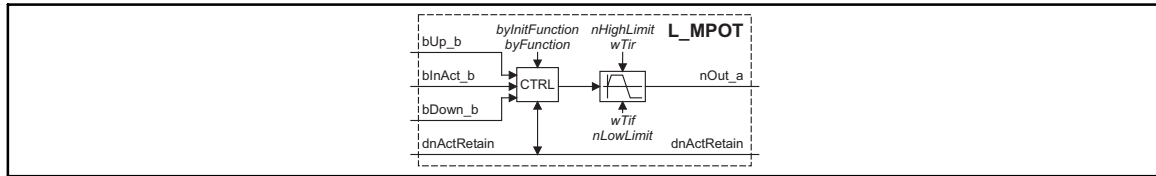


Abb. 2-69

Motorpotentiometer (L\_MPOT)

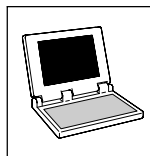
| VariableName   | DataType         | SignalType | VariableType        | Note                                               |
|----------------|------------------|------------|---------------------|----------------------------------------------------|
| bUp_b          | Bool             | binary     | VAR_INPUT           | Motorpoti läuft zur oberen Grenze.                 |
| bDown_b        | Bool             | binary     | VAR_INPUT           | Motorpoti läuft zur unteren Grenze.                |
| bInAct_b       | Bool             | binary     | VAR_INPUT           | Aktivierung einer Funktion                         |
| nOut_a         | Integer          | analog     | VAR_OUTPUT          | Sollwertausgang                                    |
| dnActRetain    | Double Integer   |            | VAR_IN_OUT          | Zur Speicherung des aktuellen Sollwertes           |
| nHighLimit     | Integer          |            | VAR CONSTANT RETAIN | Obere Grenze (16384 $\equiv$ 100% $\equiv$ C0011)  |
| nLowLimit      | Integer          |            | VAR CONSTANT RETAIN | Untere Grenze (16384 $\equiv$ 100% $\equiv$ C0011) |
| wTir           | Unsigned Integer |            | VAR CONSTANT RETAIN | Hochlaufzeit Tir (10 $\equiv$ 1 s)                 |
| wTif           | Unsigned Integer |            | VAR CONSTANT RETAIN | Ablaufzeit Tif (10 $\equiv$ 1 s)                   |
| byFunction     | Byte             |            | VAR CONSTANT RETAIN | Deaktivierungsfunktion                             |
| byInitFunction | Byte             |            | VAR CONSTANT RETAIN | Initialisierungsfunktion                           |

### Parameter-Codestellen der Instanzen

| VariableName   | L_MPOT1 |  |  | SettingRange         | Lenze   |
|----------------|---------|--|--|----------------------|---------|
| nHighLimit     | C0260   |  |  | -199.99 ... 199.99 % | 100.00  |
| nLowLimit      | C0261   |  |  | -199.99 ... 199.99 % | -100.00 |
| wTir           | C0262   |  |  | 0.1 ... 6000.0 s     | 10.00   |
| wTif           | C0263   |  |  | 0.1 ... 6000.0 s     | 10.00   |
| byFunction     | C0264   |  |  | 0 ... 5              | 0       |
| byInitFunction | C0265   |  |  | 0 ... 2              | 0       |

### Funktionsumfang

- Steuerung des Motorpotentiometers
- Initialisierung des Motorpotentiometers
- Aktuelle Ausgangswert nach Netzausfall speichern.



### 2.7.3.1 Steuerung des Motorpotentiometers

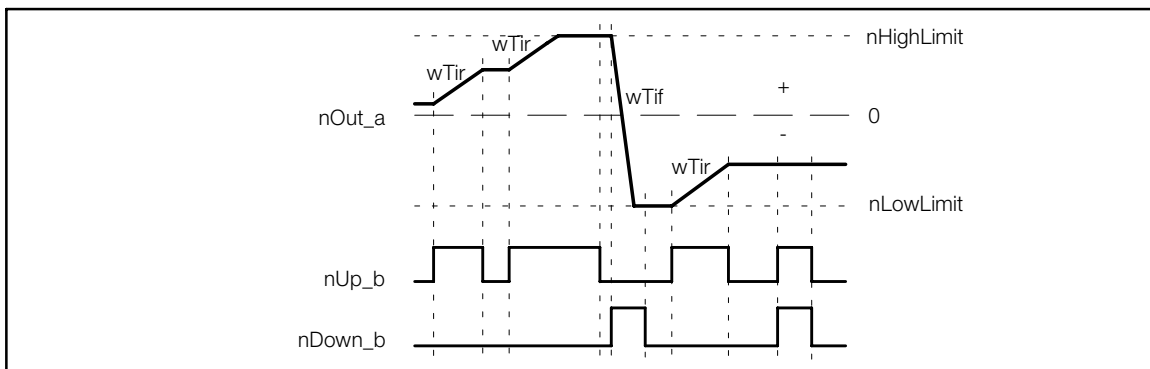


Abb. 2-70

Steuersignale des Motorpotentiometers

- $bUp\_b = TRUE$ :  
– Das Signal an  $nOut\_a$  läuft zu seinem oberen Grenzwert ( $nHighLimit$ ).
- $nDown\_b = TRUE$ :  
– Das Signal an  $nOut\_a$  läuft zu seinem unteren Grenzwert ( $nDownLimit$ ).
- $bUp\_b = FALSE$  und  $nDown\_b = FALSE$  oder  $bUp\_b = TRUE$  und  $nDown\_b = TRUE$ :  
– Das Signal an  $nOut\_a$  ändert sich nicht.

### Deaktivieren des Motorpotentiometers

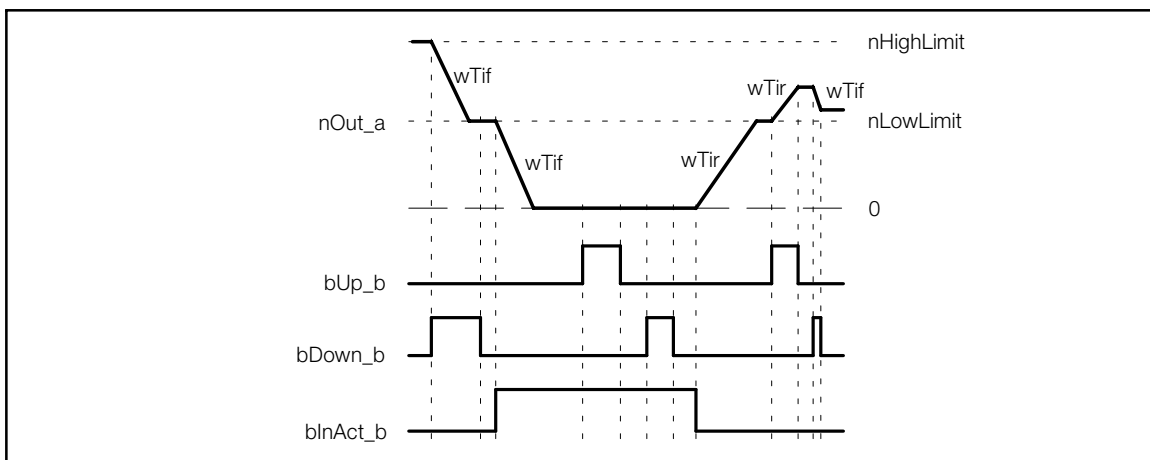


Abb. 2-71

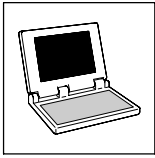
Deaktivieren des Motorpotentiometers

- Ist  $bInAct\_b = TRUE$ , dann ist das Motorpotentiometer deaktiviert.

$bInAct\_b$  hat Priorität gegenüber  $bUp\_b$  und  $bDown\_b$ .

Bei Deaktivierung des Motorpotentiometers folgt das Signal an  $nOut\_a$  der mit  $byFunction$  eingestellten Funktion. Folgende Funktionen können Sie einstellen:

| byFunction | Bedeutung                                                                                                  |
|------------|------------------------------------------------------------------------------------------------------------|
| 0          | Keine weitere Aktion; $nOut\_a$ behält seinen Wert bei.                                                    |
| 1          | Das Motorpotentiometer läuft mit der Ablaufzeit Tif auf 0 % zurück.                                        |
| 2          | Das Motorpotentiometer läuft mit der Ablaufzeit Tif auf den unteren Grenzwert ( $nLowLimit$ ).             |
| 3          | Das Motorpotentiometer wechselt sofort seinen Ausgang auf 0 % ( <b>Wichtig für die NOT-AUS-Funktion</b> ). |
| 4          | Das Motorpotentiometer wechselt sofort seinen Ausgang auf auf den unteren Grenzwert ( $nLowLimit$ ).       |
| 5          | Das Motorpotentiometer läuft mit der Hochlaufzeit Tir auf den oberen Grenzwert ( $nHighLimit$ ).           |



# Funktionsbibliothek LenzeDrive.lib

## Spezielle Funktionen

### Motorpotentiometer (L\_MPOT)

#### Aktivieren des Motorpotentiometers

- Ist *blnAct\_b* = FALSE, dann ist das Motorpotentiometer aktiviert. Die nun ablaufende Funktion ist abhängig von
  - dem aktuellen Ausgangssignal.
  - den eingestellten Grenzwerten.
  - den Steuersignalen an *bUp\_b* und *bDown\_b*.
- Befindet sich das Signal an *nOut\_a*
  - außerhalb der eingestellten Grenzen, läuft das Ausgangssignal mit den eingestellten *T<sub>i</sub>*-Zeiten zur nächsten Grenze. Dieser Ablauf ist unabhängig von den Steuersignalen an *bUp\_b* und *bDown\_b*.
  - innerhalb der eingestellten Grenzen, verläuft das Ausgangssignal entsprechend den Steuersignalen an *bUp\_b* und *bDown\_b*.

### 2.7.3.2 Initialisierung des Motorpotentiometers

Mit der Initialisierung wird das Motorpotentiometer beim Netzeinschalten mit einem bestimmten Ausgangswert geladen. Mit *byInitFunction* können Sie die Initialisierungsfunktion wählen.

| byInitFunction | Funktion                                                                                                                                                                                         |
|----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0              | Der Ausgangswert, der beim Netzabschalten ausgegeben wurde, wird im internen Speicher des Antriebsreglers nicht-flüchtig gesichert. Beim Netzeinschalten wird dieser Ausgangswert zurückgeladen. |
| 1              | Beim Netzeinschalten wird der untere Grenzwert (definiert mit <i>nLowLimit</i> ) geladen.                                                                                                        |
| 2              | Beim Netzeinschalten wird ein Ausgangswert = 0 % geladen.                                                                                                                                        |

### 2.7.3.3 Aktuellen Ausgangswert nach Netzausfall speichern

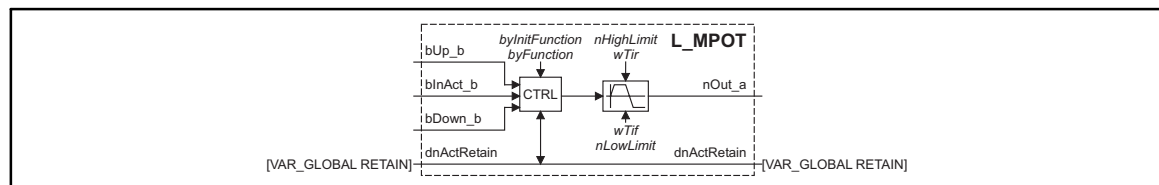
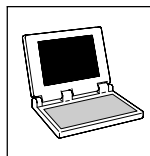


Abb. 2-72

Programmierung, um nach einem Netzausfall den aktuellen Ausgangswert zu speichern

Um nach einem Netzausfall den letzten Istwert an *nOut\_a* zu speichern, müssen Sie eine globale Variable vom Typ RETAIN (VAR\_GLOBAL RETAIN) deklarieren. Verknüpfen Sie diese Variable wie in Abb. 2-72 dargestellt.

- In dieser Variable wird immer der aktuelle Wert an *nOut\_a* gespeichert. Die Variable behält auch nach einem Netzausfall den Wert.
- Ist das Netz wieder eingeschaltet, wird der gespeicherte Wert aus der Variablen in den FB **L\_MPOT** eingelesen und als Startwert verarbeitet.



### 2.7.4 Drehzahl-Aufbereitung (L\_NSET)

Dieser FB bereitet über Hochlaufgeber oder Festdrehzahlen den Drehzahlhauptsollwert sowie einen Zusatzsollwert (oder auch andere Signale) für die nachfolgende Regelstruktur auf.

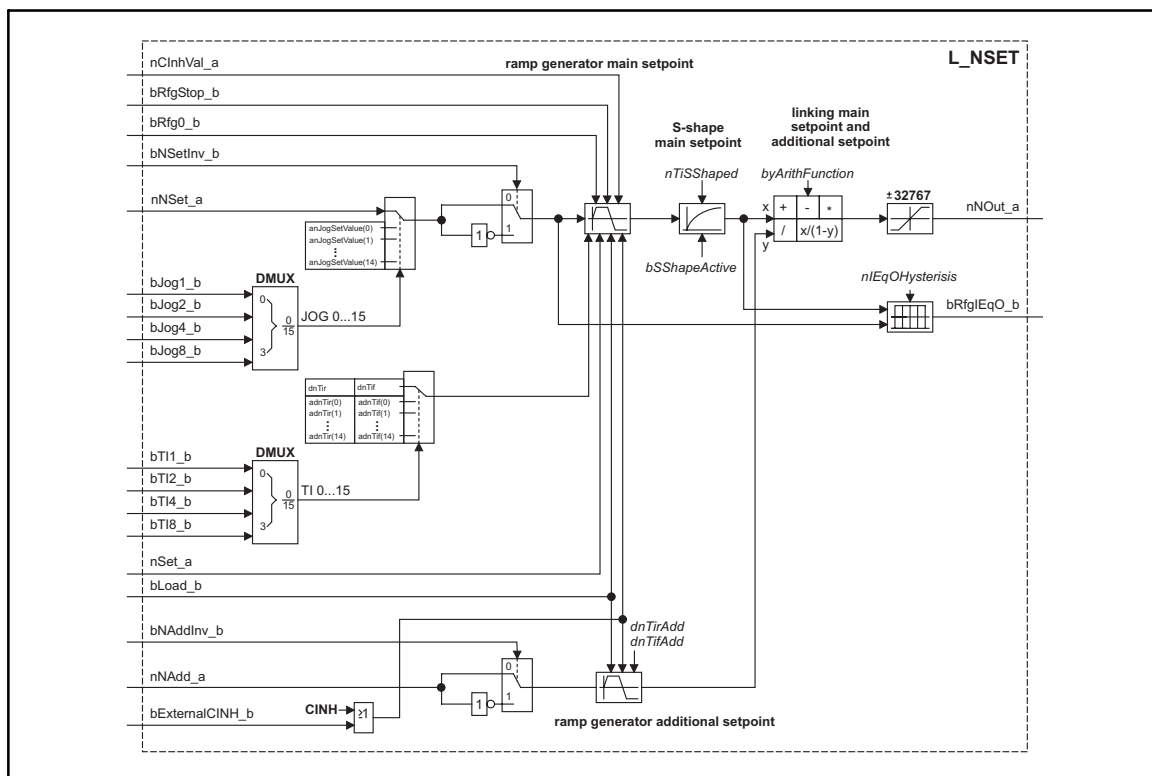
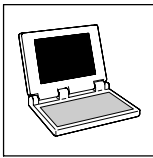


Abb. 2-73 Drehzahl-Vorverarbeitung (L\_NSET)

| Variablenname | Datentyp | Signaltyp | Variablentyp | Bemerkung                                                                                                          |
|---------------|----------|-----------|--------------|--------------------------------------------------------------------------------------------------------------------|
| nClnhVal_a    | Integer  | analog    | VAR_INPUT    | Hier wird das Signal angelegt, das der Hauptsollwert-Integrator bei gesetzter Reglersperre (CINH) übernehmen soll. |
| bRfgStop_b    | Bool     | binary    | VAR_INPUT    | Halten (Einfrieren) des Hauptsollwert-Integrators auf seinem aktuellen Wert.                                       |
| bRfg0_b       | Bool     | binary    | VAR_INPUT    | Führen des Hauptsollwert-Integrators über die aktuellen $T_i$ -Zeiten auf 0.                                       |
| bNSetInv_b    | Bool     | binary    | VAR_INPUT    | Steuerung der Signalinvertierung für den Hauptsollwert                                                             |
| nSet_a        | Integer  | analog    | VAR_INPUT    | Vorgesehen für Hauptsollwert; andere Signale zulässig.                                                             |
| bJog1_b       | Bool     | binary    | VAR_INPUT    | Auswahl und Steuerung von ablösenden "Fest-Sollwerten" für den Hauptsollwert                                       |
| bJog2_b       | Bool     | binary    | VAR_INPUT    |                                                                                                                    |
| bJog4_b       | Bool     | binary    | VAR_INPUT    |                                                                                                                    |
| bJog8_b       | Bool     | binary    | VAR_INPUT    |                                                                                                                    |
| bTl1_b        | Bool     | binary    | VAR_INPUT    | Auswahl und Steuerung von alternativen "Fest-Sollwerten" für den Hauptsollwert                                     |
| bTl2_b        | Bool     | binary    | VAR_INPUT    |                                                                                                                    |
| bTl4_b        | Bool     | binary    | VAR_INPUT    |                                                                                                                    |
| bTl8_b        | Bool     | binary    | VAR_INPUT    |                                                                                                                    |
| nSet_a        | Integer  | analog    | VAR_INPUT    | Hier wird das Signal angelegt, das der Hauptsollwert-Integrator bei $bLoad_b = TRUE$ übernehmen soll.              |
| bLoad_b       | Bool     | binary    | VAR_INPUT    | Steuerung der beiden Hochlaufgeber in besonderen Situationen, z. B. Schnellhalt (QSP)                              |
| bAddInv_b     | Bool     | binary    | VAR_INPUT    | Steuerung der Signalinvertierung für den Zusatzsollwert                                                            |



# Funktionsbibliothek LenzeDrive.lib

## Spezielle Funktionen

### Drehzahl-Aufbereitung (L\_NSET)

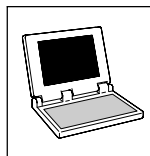
| Variablenname         | Datentyp                | Signaltyp | Variablentyp        | Bemerkung                                                            |
|-----------------------|-------------------------|-----------|---------------------|----------------------------------------------------------------------|
| nNAdd_a               | Integer                 | analog    | VAR_INPUT           | Vorgesehen für Zusatzsollwert; andere Signale zulässig               |
| bExternalCINH         | Bool                    | binary    | VAR_INPUT           | Rücksetzen des Hochlaufgebers und des zusätzlichen Rampengenerators. |
| nNOut_a               | Integer                 | analog    | VAR_OUTPUT          | Drehzahlsollwert (16384 $\equiv$ 100 % $\equiv$ $n_{max}$ (C0011))   |
| bRfIqEq0_b            | Bool                    | binary    | VAR_OUTPUT          | Statusabfrage                                                        |
| dnTir                 | Double Integer          |           | VAR CONSTANT RETAIN | Hochlaufzeit Tir für den Hauptsollwert                               |
| dnTif                 | Double Integer          |           | VAR CONSTANT RETAIN | Ablaufzeit Tif für den Hauptsollwert                                 |
| adnTir[0...14]        | Array of Double Integer |           | VAR CONSTANT RETAIN | Hochlaufzeit Tir für den Hauptsollwert                               |
| adnTif[0...14]        | Array of Double Integer |           | VAR CONSTANT RETAIN | Ablaufzeit Tif für den Hauptsollwert                                 |
| anJogSetValue[0...14] | Array of Integer        |           | VAR CONSTANT RETAIN | Wählbare Festdrehzahlen                                              |
| bSShapeActive         | Bool                    |           | VAR CONSTANT RETAIN | S-Form-Hochlaufgeber-Kennlinie für den Hauptsollwert                 |
| nTiSShaped            | Integer                 |           | VAR CONSTANT RETAIN | T <sub>i</sub> -Zeit des S-Form-Hochlaufgebers                       |
| byArithFunction       | Byte                    |           | VAR CONSTANT RETAIN | Arithmetikfunktion, verknüpft Haupt- und Zusatzsollwert.             |
| dnTirAdd              | Double Integer          |           | VAR CONSTANT RETAIN | Hochlaufzeit T <sub>ir</sub> für den Zusatzsollwert                  |
| dnTifAdd              | Double Integer          |           | VAR CONSTANT RETAIN | Ablaufzeit T <sub>if</sub> für den Zusatzsollwert                    |
| nIEqOHysteresis       | Integer                 |           | VAR CONSTANT RETAIN | Schwelle Hochlaufgeber für den Hauptsollwert                         |

### Parameter-Codestellen der Instanzen

| Variablenname         | L_NSET1        |  |  | Einstellbereich      | Lenze |
|-----------------------|----------------|--|--|----------------------|-------|
| dnTir                 | C0012          |  |  | 0.000 ... 999.900 s  | 0.000 |
| adnTir[0...14]        | C0101/1 ... 15 |  |  | 0.000 ... 999.900 s  | 0.000 |
| dnTif                 | C0013          |  |  | 0.000 ... 999.900 s  | 0.000 |
| adnTif[0...14]        | C0103/1 ... 15 |  |  | 0.000 ... 999.900 s  | 0.000 |
| anJogSetValue[0...14] | C0039/1 ... 15 |  |  | -199.99 ... 199.99 % | 0.00  |
| bSShapeActive         | C0134          |  |  | 0 ... 1              | 0     |
| nTiSShaped            | C0182          |  |  | 0.01 ... 50.00 s     | 20.00 |
| byArithFunction       | C0190          |  |  | 0 ... 5              | 0     |
| dnTirAdd              | C0220          |  |  | 0.000 ... 999.900 s  | 0.000 |
| dnTifAdd              | C0221          |  |  | 0.000 ... 999.900 s  | 0.000 |
| nIEqOHysteresis       | C0241          |  |  | 0.00 ... 100.00 %    | 1.00  |

## 2.7.4.1 Hauptsollwertpfad

- Die Signale im Hauptsollwertpfad sind auf den Wertebereich  $\pm 32767$  begrenzt.
- Das Signal an *nNSet\_a* wird zunächst über die Funktion JOG-Auswahl geführt.
- Ein ausgewählter JOG-Wert schaltet den Eingang *nNSet\_a* inaktiv. Dann arbeitet die nachfolgende Signalaufbereitung mit dem JOG-Wert.



### 2.7.4.2 JOG-Sollwerte

JOG-Sollwerte sind feste Werte, die unter *anJogSetValue[0]* ... *anJogSetValue[14]* im Speicher abgelegt sind.

- Die JOG-Werte sind über *bJog1\_b* ... *bJog8\_b* aus dem Speicher abrufbar.  
– Diese Eingänge sind binär-kodiert, so dass 15 JOG-Werte abrufbar sind.
- Die Dekodierung für die Freigabe der JOG-Werte (abrufen aus dem Speicher) erfolgt nach folgendem Schema:

| Hauptsollwert von | bJog8_b | bJog4_b | bJog2_b | bJog1_b |
|-------------------|---------|---------|---------|---------|
| nNSet_a           | 0       | 0       | 0       | 0       |
| anJogSetValue[0]  | 0       | 0       | 0       | 1       |
| anJogSetValue[1]  | 0       | 0       | 1       | 0       |
| anJogSetValue[2]  | 0       | 0       | 1       | 1       |
| anJogSetValue[3]  | 0       | 1       | 0       | 0       |
| anJogSetValue[4]  | 0       | 1       | 0       | 1       |
| anJogSetValue[5]  | 0       | 1       | 1       | 0       |
| anJogSetValue[6]  | 0       | 1       | 1       | 1       |
| anJogSetValue[7]  | 1       | 0       | 0       | 0       |
| anJogSetValue[8]  | 1       | 0       | 0       | 1       |
| anJogSetValue[9]  | 1       | 0       | 1       | 0       |
| anJogSetValue[10] | 1       | 0       | 1       | 1       |
| anJogSetValue[11] | 1       | 1       | 0       | 0       |
| anJogSetValue[12] | 1       | 1       | 0       | 1       |
| anJogSetValue[13] | 1       | 1       | 1       | 0       |
| anJogSetValue[14] | 1       | 1       | 1       | 1       |

0 = FALSE

1 = TRUE

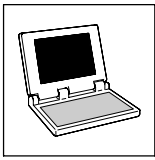
- Die Anzahl der zu belegenden VAR\_INPUT-Variablen ist abhängig von der Anzahl der benötigten JOG-Sollwerte. Höchstens 4 VAR\_INPUT-Variablen und damit 15 Auswahlmöglichkeiten stehen zur Verfügung.

| Anzahl der benötigten JOG-Sollwerte | Anzahl der zu belegenden VAR_INPUT (bJog1_b ... bJog8_b) |
|-------------------------------------|----------------------------------------------------------|
| 1                                   | mindestens 1                                             |
| 1 ... 3                             | mindestens 2                                             |
| 4 ... 7                             | mindestens 3                                             |
| 8 ... 15                            | 4                                                        |

### 2.7.4.3 Sollwert-Invertierung

Das Ausgangssignal der JOG-Funktion wird über einen Inverter geführt.

Das Vorzeichen des Sollwerts kehrt sich um, wenn *bNSetInv\_b* = TRUE geschaltet wird.



# Funktionsbibliothek LenzDrive.lib

## Spezielle Funktionen

### Drehzahl-Aufbereitung (L\_NSET)

#### 2.7.4.4 Hochlaufgeber für den Hauptsollwert

Der Sollwert wird anschließend über einen Hochlaufgeber mit linearer Charakteristik geführt. Der Hochlaufgeber überführt Sollwertsprünge am Eingang in eine Rampe.

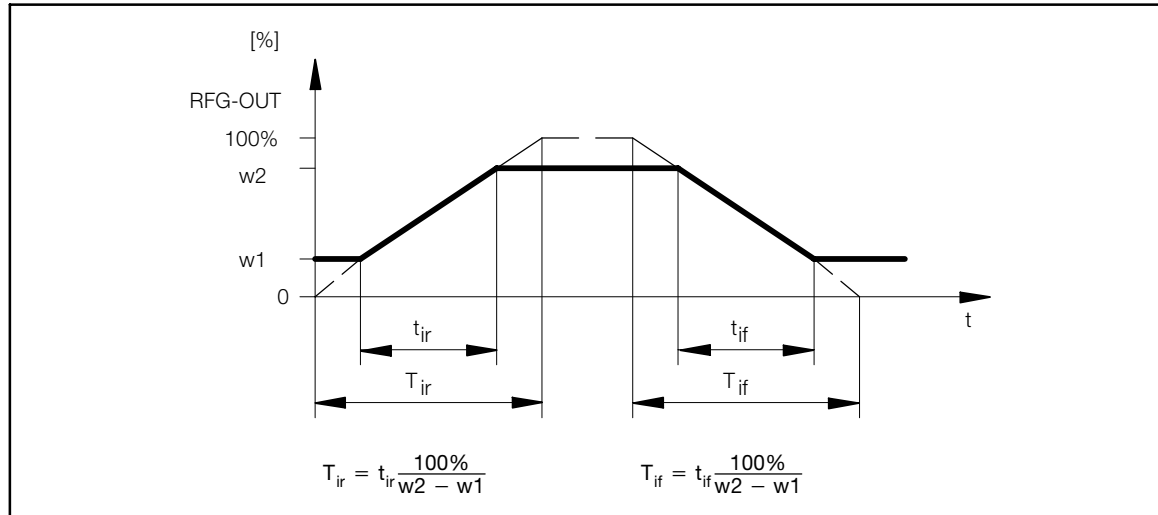


Abb. 2-74

Hoch- und Ablaufzeiten des Hochlaufgebers

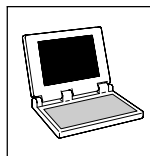
w1 ,w2 Wechsel der Hauptsollwerts in Abhängigkeit von t<sub>ir</sub> bzw. t<sub>if</sub>  
RFG-OUT Ausgang des Hochlaufgebers

- T<sub>i</sub>-Zeiten sind feste Werte, die unter
  - adnTir[0] ... adnTir[14] (Hochlaufzeiten) und
  - adnTif[0] ... adnTif[14] (Ablaufzeiten) im Speicher abgelegt sind.
- Die T<sub>i</sub>-Zeiten sind über bTI1\_b ... bTI8\_b aus dem Speicher abrufbar.
  - Diese Eingänge sind binär-kodiert, so dass 16 T<sub>i</sub>-Zeiten abrufbar sind.
  - Die T<sub>i</sub>-Zeiten können Sie nur paarweise aktivieren.
- Die Dekodierung für die Freigabe der T<sub>i</sub>-Zeiten (abrufen aus dem Speicher) erfolgt nach folgendem Schema:

| Hochlaufzeit | Ablaufzeit | bTI8_b | bTI4_b | bTI2_b | bTI1_b |
|--------------|------------|--------|--------|--------|--------|
| dnTir        | dnTif      | 0      | 0      | 0      | 0      |
| adnTir[0]    | adnTif[0]  | 0      | 0      | 0      | 1      |
| adnTir[1]    | adnTif[1]  | 0      | 0      | 1      | 0      |
| adnTir[2]    | adnTif[2]  | 0      | 0      | 1      | 1      |
| adnTir[3]    | adnTif[3]  | 0      | 1      | 0      | 0      |
| adnTir[4]    | adnTif[4]  | 0      | 1      | 0      | 1      |
| adnTir[5]    | adnTif[5]  | 0      | 1      | 1      | 0      |
| adnTir[6]    | adnTif[6]  | 0      | 1      | 1      | 1      |
| adnTir[7]    | adnTif[7]  | 1      | 0      | 0      | 0      |
| adnTir[8]    | adnTif[8]  | 1      | 0      | 0      | 1      |
| adnTir[9]    | adnTif[9]  | 1      | 0      | 1      | 0      |
| adnTir[10]   | adnTif[10] | 1      | 0      | 1      | 1      |
| adnTir[11]   | adnTif[11] | 1      | 1      | 0      | 0      |
| adnTir[12]   | adnTif[12] | 1      | 1      | 0      | 1      |
| adnTir[13]   | adnTif[13] | 1      | 1      | 1      | 0      |
| adnTir[14]   | adnTif[14] | 1      | 1      | 1      | 1      |

0 = FALSE

1 = TRUE



- Bei gesetzter Reglersperre (CINH) übernimmt der Hochlaufgeber den Wert, der an *nClnhVal\_a* liegt und gibt diesen an die nachgeschaltete Funktion weiter. Diese Funktion besitzt die Priorität vor allen anderen Funktionen.
- *bRfgStop\_b* = TRUE:
  - Der Hochlaufgeber wird angehalten. Änderungen am Eingang des Hochlaufgebers wirken sich nicht auf das Ausgangssignal aus.
- *bRfg0\_b* = TRUE:
  - Der Hochlaufgeber läuft an seiner Ablauframpe auf 0.
- Weiterhin besteht die Möglichkeit den Hochlaufgeber online mit einem definierten Wert zu laden. Dazu muss *bLoad\_b* = TRUE gesetzt sein. Solange dieser Eingang gesetzt ist, wird der Wert an *nNSet\_a* in den Hochlaufgeber übernommen und am Ausgang ausgegeben.

### Prioritäten:

| CINH | bLoad_b | bRfg0_b | bRfgStop_b | Funktion                                                                        |
|------|---------|---------|------------|---------------------------------------------------------------------------------|
| 0    | 0       | 0       | 0          | RFG folgt dem Eingangswert über die eingestellten Rampen.                       |
| 0    | 0       | 0       | 1          | Der Wert am Ausgang von RFG wird eingefroren.                                   |
| 0    | 0       | 1       | 0          | RFG läuft über die eingestellte Ablaufzeit auf 0.                               |
| 0    | 0       | 1       | 1          |                                                                                 |
| 0    | 1       | 0       | 0          | RFG übernimmt den Wert an <i>nSet_a</i> und gibt ihn an seinem Ausgang aus.     |
| 0    | 1       | 0       | 1          |                                                                                 |
| 0    | 1       | 1       | 0          |                                                                                 |
| 0    | 1       | 1       | 1          |                                                                                 |
| 1    | 0       | 0       | 0          | RFG übernimmt den Wert an <i>nClnhVal_a</i> und gibt ihn an seinem Ausgang aus. |
| 1    | 0       | 0       | 1          |                                                                                 |
| 1    | 0       | 1       | 0          |                                                                                 |
| 1    | 0       | 1       | 1          |                                                                                 |
| 1    | 1       | 0       | 0          |                                                                                 |
| 1    | 1       | 0       | 1          |                                                                                 |
| 1    | 1       | 1       | 0          |                                                                                 |
| 1    | 1       | 1       | 1          |                                                                                 |

0 = FALSE

1 = TRUE

### 2.7.4.5 S-Rampe

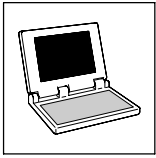
Dem linearen Hochlaufgeber ist ein PT1-Glied nachgeschaltet. Diese Anordnung realisiert eine S-Rampe für einen nahezu ruckfreien Hoch- und Ablauf.

- Das PT1-Glied wird mit *bSShapeActive* zu- bzw. abgeschaltet.
- Die Zeitkonstante wird mit *nTISShaped* eingestellt.

### 2.7.4.6 Arithmetische Verknüpfung

Der Arithmetik-Baustein verknüpft arithmetisch den Hauptsollwert und den Zusatzsollwert. Die arithmetische Verknüpfung wird über *byArithFunction* ausgewählt.

| byArithFunction | Funktion                                  | Beispiel                                     |
|-----------------|-------------------------------------------|----------------------------------------------|
| 0               | $nNout\_a = x$ (y wird nicht verarbeitet) |                                              |
| 1               | $nNout\_a = x + y$                        |                                              |
| 2               | $nNout\_a = x - y$                        |                                              |
| 3               | $nNout\_a = x \cdot y$                    | $nNout\_a = \frac{x \cdot y}{16384}$         |
| 4               | $nNout\_a = x /  y $                      | $nNout\_a = \frac{x}{ y } \cdot 164$         |
| 5               | $nNout\_a = x / (100\% - y)$              | $nNout\_a = \frac{x}{16384 - y} \cdot 16384$ |



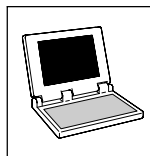
## **Funktionsbibliothek LenzeDrive.lib**

### **Spezielle Funktionen**

#### **Drehzahl-Aufbereitung (L\_NSET)**

#### **2.7.4.7 Zusatzsollwert**

- Über *nNAdd\_a* können Sie einen Zusatzsollwert (z. B. ein Korrektursignal) mit dem Hauptsollwert verknüpfen.
- Über *bNAddInv\_b* können Sie das Eingangssignal invertieren, bevor dieses auf den Hochlaufgeber wirkt. Der Hochlaufgeber hat eine lineare Charakteristik. Seine T<sub>i</sub>-Zeiten stellen Sie mit *dnTirAdd* (Hochlaufzeit) und *dnTifAdd* (Ablaufzeit) ein.
- Bei *bLoad\_b* = TRUE wird der Hochlaufgeber auf 0 gesetzt und dort gehalten, ohne Berücksichtigung der T<sub>i</sub>-Zeiten. Gleiches gilt bei gesetzter Reglersperre (CINH).



### 2.7.5 Prozessregler (L\_PCTRL)

Dieser FB kann z. B. als überlagerter Regler (Tänzerlagereger, Zugregler, Druckregler usw.) verwendet werden.

Soll- und Istwert werden dem Prozessregler über die entsprechenden Eingänge zugeführt und entsprechend dem gewählten Regelalgorithmus (PID-, PI- oder P-Algorithmus) weiterverarbeitet.

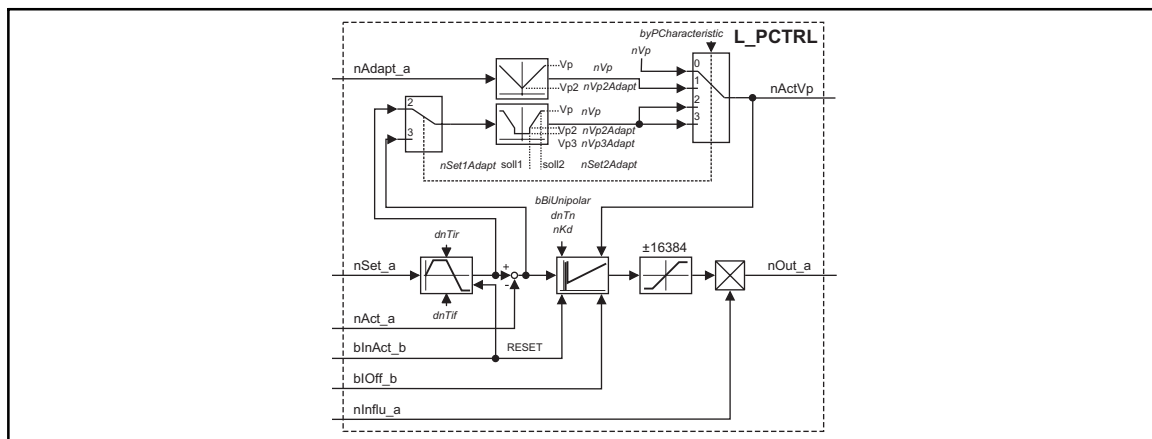
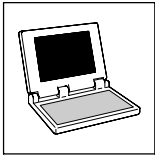


Abb. 2-75

Prozessregler (L\_PCTRL)

| VariableName      | DataType       | SignalType | VariableType        | Note                                                                                                                                                                                               |
|-------------------|----------------|------------|---------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| nAdapt_a          | Integer        | analog     | VAR_INPUT           | Verstärkung $V_p$<br>• Wertebereich: $\pm 32767$<br>• Sie können die Verstärkung online verändern.                                                                                                 |
| nSet_a            | Integer        | analog     | VAR_INPUT           | Eingang des Prozess-Sollwertes.<br>• Möglicher Wertebereich: $\pm 32767$<br>• Über den Hochlaufgeber können Sprungsignale in ihrem zeitlichen Verlauf verzögert werden (mit $dnTir$ und $dnTif$ ). |
| nAct_a            | Integer        | analog     | VAR_INPUT           | Istwert-Eingang.<br>• Wertebereich: $\pm 32767$                                                                                                                                                    |
| bInAct_b          | Bool           | binary     | VAR_INPUT           | Deaktivieren des Prozessreglers<br>• Sie können die Funktion online ausführen.                                                                                                                     |
| bOff_b            | Bool           | binary     | VAR_INPUT           | I-Anteil auf 0 setzen.<br>• Sie können die Funktion online ausführen.                                                                                                                              |
| nInflu_a          | Integer        | analog     | VAR_INPUT           | Bewerten oder Ausblenden des Ausgangssignals.<br>• Wertebereich: $\pm 32767$                                                                                                                       |
| nOut_a            | Integer        | analog     | VAR_OUTPUT          | Ausgangssignal Wertebereich $\pm 16384$ (bipolar), bzw. 0 ... 16384 (unipolar)                                                                                                                     |
| nActVp            | Integer        |            | VAR_OUTPUT          | Zeigt den aktuellen Wert der Verstärkung an.                                                                                                                                                       |
| nVp               | Integer        |            | VAR CONSTANT RETAIN | Verstärkung $V_p$ ( $10 \equiv 1.0$ )                                                                                                                                                              |
| dnTn              | Double Integer |            | VAR CONSTANT RETAIN | Nachstellzeit $T_n$ ( $20 \equiv 20$ ms)                                                                                                                                                           |
| nKd               | Integer        |            | VAR CONSTANT RETAIN | Differenzialanteil $K_d$ ( $10 \equiv 1.0$ )                                                                                                                                                       |
| nVp2Adapt         | Integer        |            | VAR CONSTANT RETAIN | $V_p2$ - Prozessregleradaption ( $64 \equiv 1.0$ )                                                                                                                                                 |
| nVp3Adapt         | Integer        |            | VAR CONSTANT RETAIN | $V_p3$ - Prozessregleradaption ( $64 \equiv 1.0$ )                                                                                                                                                 |
| nSet1Adapt        | Integer        |            | VAR CONSTANT RETAIN | soll1 - Prozessregleradaption ( $16384 \equiv 100.00$ %)                                                                                                                                           |
| nSet2Adapt        | Integer        |            | VAR CONSTANT RETAIN | soll2 - Prozessregleradaption ( $16384 \equiv 100.00$ %)                                                                                                                                           |
| byPCharacteristic | Byte           |            | VAR CONSTANT RETAIN | Funktionsauswahl für die Vorgabe der P-Verstärkung                                                                                                                                                 |
| dnTir             | Double Integer |            | VAR CONSTANT RETAIN | Hochlaufzeit $T_{ir}$ ( $1000 \equiv 1.000$ s)                                                                                                                                                     |
| dnTif             | Double Integer |            | VAR CONSTANT RETAIN | Ablaufzeit $T_{if}$ ( $1000 \equiv 1.000$ s)                                                                                                                                                       |
| bBiUnipolar       | Bool           |            | VAR CONSTANT RETAIN | Wertebereich des Ausgangssignals                                                                                                                                                                   |



# Funktionsbibliothek LenzeDrive.lib

## Spezielle Funktionen

### Prozessregler (L\_PCTRL)

#### Parameter-Codestellen der Instanzen

| VariableName      | L_PCTRL1 |  |  | SettingRange        | Lenze  |
|-------------------|----------|--|--|---------------------|--------|
| nVp               | C0222    |  |  | 0.1 ... 500.0       | 1.0    |
| dnTn              | C0223    |  |  | 20 ... 99999 ms     | 400    |
| nKd               | C0224    |  |  | 0.0 ... 5.0         | 0.0    |
| nVp2Adapt         | C0325    |  |  | 0.1 ... 500.0       | 1.0    |
| nVp3Adapt         | C0326    |  |  | 0.1 ... 500.0       | 1.0    |
| nSet1Adapt        | C0328    |  |  | 0.00 ... 100.00 %   | 0.00   |
| nSet2Adapt        | C0327    |  |  | 0.00 ... 100.00 %   | 100.00 |
| byPCharacteristic | C0329    |  |  | 0 ... 3             | 0      |
| dnTir             | C0332    |  |  | 0.000 ... 999.900 s | 0.000  |
| dnTif             | C0333    |  |  | 0.000 ... 999.900 s | 0.000  |
| bBiUnipolar       | C0337    |  |  | 0 ... 1             | 0      |

#### Funktionsumfang

- Regelcharakteristik
- Hochlaufgeber
- Wertebereich des Ausgangssignals
- Bewertung des Ausgangssignals
- Deaktivierung des Prozessreglers

### 2.7.5.1 Regelcharakteristik

In der Werkseinstellung ist der PID-Algorithmus aktiv.

#### Differenzialanteil $K_d$

Sie können den  $K_d$ -Anteil deaktivieren, indem Sie  $nKd = 0.0$  setzen. Damit wird der Regler zu einem PI-Regler (bzw. P-Regler, wenn auch der I-Anteil abgeschaltet ist).

#### Integralanteil I

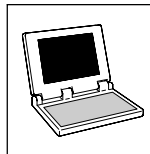
Sie können den I-Anteil mit

- $bOff\_b = TRUE$  abschalten.
- $bOff\_b = FALSE$  zuschalten.

Das Zu- und Abschalten ist auch online möglich. Soll der I-Anteil dauerhaft abgeschaltet bleiben, legen Sie  $bOff\_b$  auf TRUE.

#### Nachstellzeit $T_n$

Mit  $nTn$  können Sie die Nachstellzeit parametrieren.



### Verstärkung $V_p$

Die Verstärkung  $V_p$  können Sie unterschiedlich einstellen. Mit *byPCharacteristic* wählen Sie die gewünschte Funktion aus.

Über *nActVp* können Sie sich den aktuellen Wert der Verstärkung  $V_p$  anzeigen lassen.

*byPCharacteristic* = 0:

- Die Vorgabe der Verstärkung  $V_p$  erfolgt über *nVp*.

*byPCharacteristic* = 1:

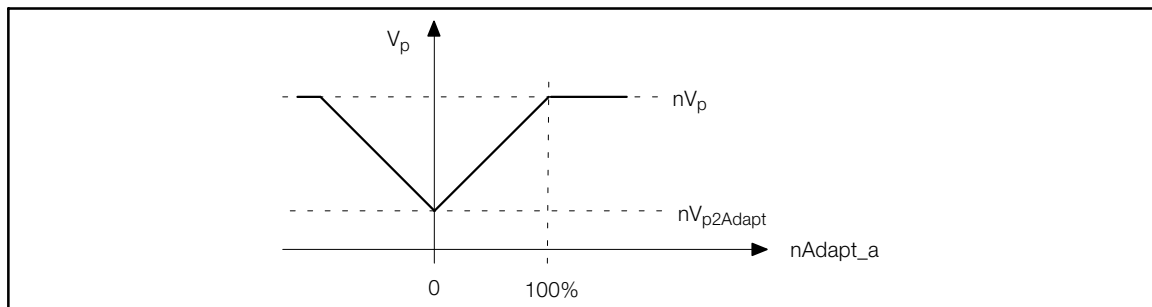


Abb. 2-76

Vorgabe der Verstärkung  $V_p$  über *nAdapt\_a*

- Die Vorgabe der Verstärkung  $V_p$  erfolgt über *nAdapt\_a*. Dabei wird der Eingangswert über eine Kennlinie mit linearer Charakteristik geführt. Die Steigung der Kennlinie wird mit *nVp* (obere Grenze) und *nVp2Adapt* (untere Grenze) festgelegt. Der Wert in *nVp* gilt, wenn der Eingangswert = +100 % oder -100 % beträgt (100 % = 16384). Der Wert in *nVp2Adapt* gilt, wenn der Eingangswert = 0 beträgt.

*byPCharacteristic* = 2:

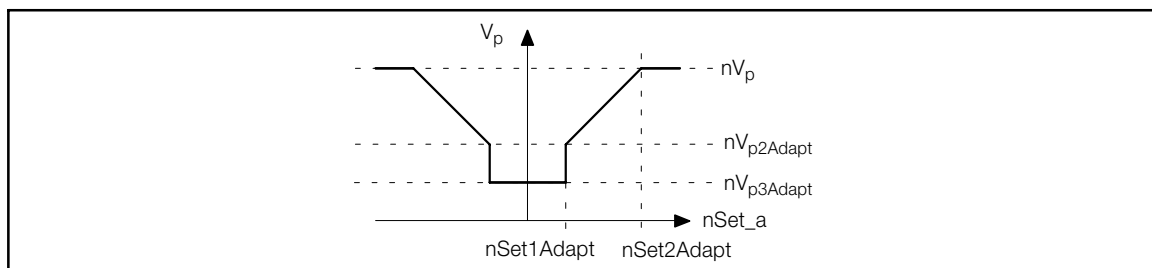


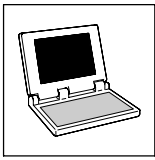
Abb. 2-77

Vorgabe der Verstärkung  $V_p$ , abgeleitet vom Prozesssollwert *nSet\_a*

- Die Vorgabe der Verstärkung  $V_p$  wird vom Prozesssollwert *nSet\_a* abgeleitet. Der Sollwert wird hinter dem Hochlaufgeber abgenommen und über eine Kennlinie mit 3 Stützstellen berechnet.

*byPCharacteristic* = 3:

- Die Vorgabe der Verstärkung  $V_p$  wird von der Regelabweichung abgeleitet und über die gleiche Kennlinienbildung geführt wie bei *byPCharacteristic* = 2.



# Funktionsbibliothek LenzeDrive.lib

## Spezielle Funktionen Prozessregler (L\_PCTRL)

### 2.7.5.2 Hochlaufgeber

Der Sollwert an *nSet\_a* wird über einen Hochlaufgeber mit linearer Charakteristik geführt (100 %  $\equiv$  16384  $\equiv$   $n_{\max}$  (C0011)). Hierdurch lassen sich Sollwertsprünge am Eingang in eine Rampe überführen.

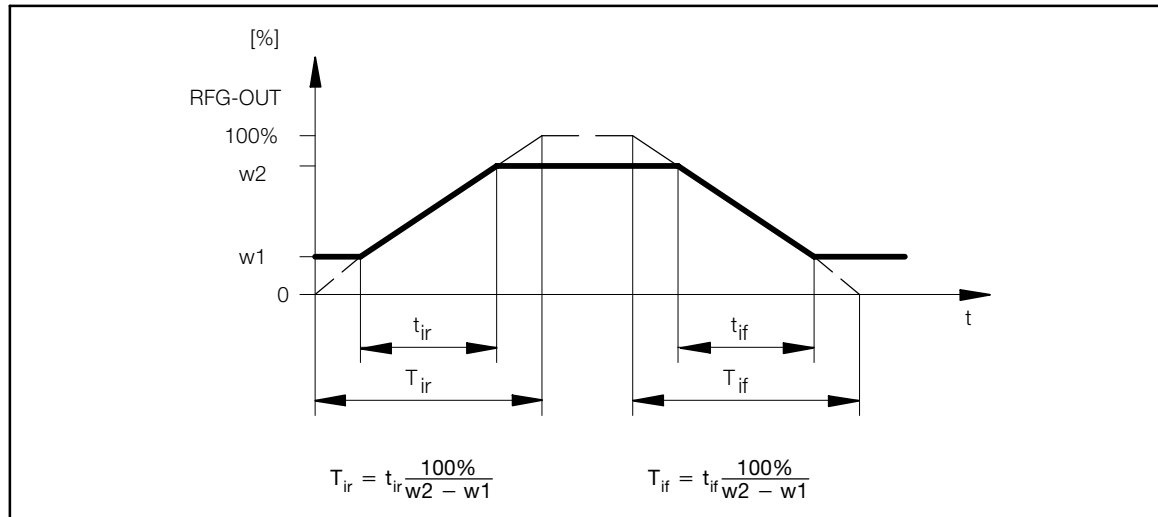


Abb. 2-78

Hoch- und Ablaufzeiten des Hochlaufgebers

w1, w2 Wechsel der Hauptsollwerts in Abhängigkeit von  $t_{ir}$  bzw.  $t_{if}$   
RFG-OUT Ausgang des Hochlaufgebers

- Die Rampen für Hoch- und Ablauf sind getrennt einstellbar:
  - Hochlaufzeit  $t_{ir}$  mit *dnTir*
  - Ablaufzeit  $t_{if}$  mit *dnTif*
- Mit *blnAct\_b* = TRUE wird der Hochlaufgeber sofort auf 0 gesetzt.

### 2.7.5.3 Wertebereich des Ausgangssignals

- Der Prozessregler arbeitet in der Werkseinstellung bipolar (*bBiUnipolar* = 0).
  - Das Ausgangssignal wird auf  $\pm 100$  % ( $\pm 16384$ ) begrenzt.
- Mit *bBiUnipolar* = 1 arbeitet der Prozessregler unipolar.
  - Das Ausgangssignal wird auf 0 ... +100 % (0 ... 16384) begrenzt.

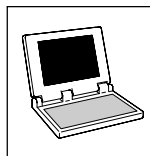
### 2.7.5.4 Bewertung des Ausgangssignals

- Der Begrenzung folgt eine Bewertung des Ausgangssignals über *nInflu\_a*.
  - Die Berechnung erfolgt nach der Formel:

$$nOut_a = \frac{(\text{Werte nach der Begrenzung}) \cdot nInflu_a}{16384}$$

### 2.7.5.5 Deaktivierung des Prozessreglers

- blnAct\_b* = TRUE deaktiviert den Prozessregler. Damit wird
  - nOut\_a* = 0 gesetzt.
  - der I-Anteil = 0 gesetzt.
  - der Hochlaufgeber = 0 gesetzt.



### 2.7.6 Rechts/Links/Quickstop (L\_RLQ)

Dieser FB verknüpft drahtbruchsicher die Vorgabe einer Drehrichtung und der QSP-Funktion.

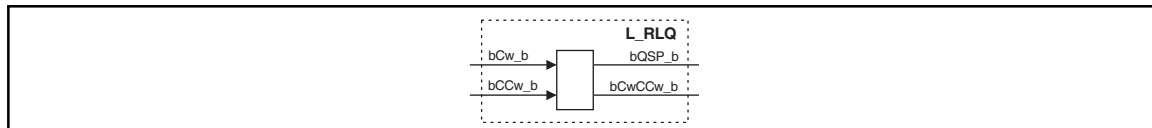


Abb. 2-79

Rechts/Links/Quickstop (L\_RLQ)

| VariableName | DataType | SignalType | VariableType | Note                |
|--------------|----------|------------|--------------|---------------------|
| bCw_b        | Bool     | binary     | VAR_INPUT    | Rechts              |
| bCCw_b       | Bool     | binary     | VAR_INPUT    | Links               |
| bCwCCw_b     | Bool     | binary     | VAR_OUTPUT   | Status Rechts/Links |
| bQSP_b       | Bool     | binary     | VAR_OUTPUT   | Quickstop setzen    |

#### Funktion

- Nach dem Netzeinschalten **und** gleichzeitigem TRUE an beiden Eingängen sind die Ausgänge wie folgt geschaltet:

| Eingänge |        | Ausgänge |        |
|----------|--------|----------|--------|
| bCw_b    | bCCw_b | bCwCCw_b | bQSP_b |
| 1        | 1      | 0        | 1      |

0 = FALSE

1 = TRUE

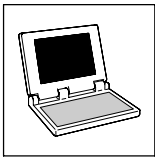
- Erst wenn einer der Eingänge nach dem Netzeinschalten einmal auf TRUE gesetzt wurde, gilt folgende Wahrheitstabelle:

| Eingänge |        | Ausgänge    |             |
|----------|--------|-------------|-------------|
| bCw_b    | bCCw_b | bCwCCw_b    | bQSP_b      |
| 0        | 0      | 0           | 1           |
| 1        | 0      | 1           | 0           |
| 0        | 1      | 1           | 0           |
| 1        | 1      | unverändert | unverändert |

0 = FALSE

1 = TRUE

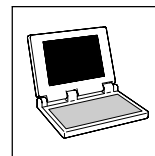
Wenn Sie während des Betriebs beide Eingänge auf TRUE setzen, behalten beide Ausgänge ihren vorher ausgegebenen Wert



## ***Funktionsbibliothek LenzeDrive.lib***

### ***Spezielle Funktionen***

***Rechts/Links/Quickstop (L\_RLQ)***



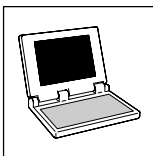
## 3 Anhang

### 3.1 Codetabelle

So lesen Sie die Codetabelle:

| Spalte  | Abkürzung  | Bedeutung                                                                                                                                                 |
|---------|------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------|
| Code    | Cxxxx      | Codestelle Cxxxx                                                                                                                                          |
|         | 1          | Subcodestelle 1 der Codestelle Cxxxx                                                                                                                      |
|         | 2          | Subcodestelle 2 der Codestelle Cxxxx                                                                                                                      |
|         | ...        | ...                                                                                                                                                       |
|         | [Cxxxx]    | Parameterwert der Codestelle kann nur bei gesperrtem Regler geändert werden.                                                                              |
| LCD     |            | LCD-Anzeige der Bedieneinheit <ul style="list-style-type: none"> <li>• DIS: Nur Anzeige (Display)</li> <li>• Alle übrigen sind Parameterwerte.</li> </ul> |
| Lenze   |            | Lenze-Einstellung der Codestelle                                                                                                                          |
|         | →          | Die Spalte WICHTIG enthält weitere Informationen.                                                                                                         |
| Auswahl | 1 {1 %} 99 | Minimaler Wert {Kleinste Schrittweite/Einheit} Maximaler Wert                                                                                             |
| WICHTIG | -          | Zusätzliche, wichtige Erläuterungen zur Codestelle                                                                                                        |

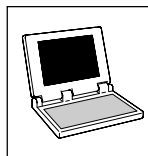
| Code  | LCD             | Einstellmöglichkeiten |                         | WICHTIG                                                                                                                                                                                                                                                               | Info         |
|-------|-----------------|-----------------------|-------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------|
|       |                 | Lenze                 | Auswahl                 |                                                                                                                                                                                                                                                                       |              |
| C0012 | dnTir           | 0.000                 | 0.000 {0,001 s} 999.900 | Hochlaufzeit $T_{ir}$ für den Hauptsollwert von L_NSET1 <ul style="list-style-type: none"> <li>• Bezogen auf Drehzahländerung <math>0 \dots \eta_{max}</math></li> </ul>                                                                                              | 2-73         |
| C0013 | dnTif           | 0.000                 | 0.000 {0,001 s} 999.900 | Ablaufzeit $T_{if}$ für den Hauptsollwert von L_NSET1 <ul style="list-style-type: none"> <li>• Bezogen auf Drehzahländerung <math>\eta_{max} \dots 0</math></li> </ul>                                                                                                | 2-73         |
| C0039 | anJOGSetValue1  | 0.00                  | -199.99 {0.01} 199.99   | Mit digitalen Eingängen wählbare Festdrehzahlen (JOG-Sollwerte) für L_NSET1                                                                                                                                                                                           | 2-73         |
|       | ...             | ...                   |                         |                                                                                                                                                                                                                                                                       |              |
|       | anJOGSetValue14 | 0.00                  |                         |                                                                                                                                                                                                                                                                       |              |
|       | anJOGSetValue15 | 0.00                  |                         |                                                                                                                                                                                                                                                                       |              |
| C0101 | adnTir1         | 0.000                 | 0.000 {0,001 s} 999.900 | Zusätzliche Hochlaufzeiten $T_{ir}$ für den Hauptsollwert von L_NSET1 <ul style="list-style-type: none"> <li>• Bezogen auf Drehzahländerung <math>0 \dots \eta_{max}</math></li> </ul>                                                                                | 2-73         |
|       | adnTir2         | 0.000                 |                         |                                                                                                                                                                                                                                                                       |              |
|       | ...             | ...                   |                         |                                                                                                                                                                                                                                                                       |              |
|       | adnTir15        | 0.000                 |                         |                                                                                                                                                                                                                                                                       |              |
| C0103 | adnTif1         | 0.000                 | 0.000 {0,001 s} 999.900 | Zusätzliche Ablaufzeiten $T_{if}$ für den Hauptsollwert von L_NSET1 <ul style="list-style-type: none"> <li>• Bezogen auf Drehzahländerung <math>\eta_{max} \dots 0</math></li> </ul>                                                                                  | 2-73         |
|       | adnTif2         | 0.000                 |                         |                                                                                                                                                                                                                                                                       |              |
|       | ...             | ...                   |                         |                                                                                                                                                                                                                                                                       |              |
|       | adnTif15        | 0.000                 |                         |                                                                                                                                                                                                                                                                       |              |
| C0134 | bSShapeActive   | 0                     | 0 linear<br>1 S-shaped  | Hochlaufgeber-Kennlinie für Hauptsollwert von L_NSET1<br>Linear<br>S-förmig                                                                                                                                                                                           | 2-73<br>2-73 |
| C0182 | nTiSShaped      | 20.00                 | 0.01 s {0,01 s} 50.00 s | $T_i$ -Zeit des S-Form-Hochlaufgebers für L_NSET1<br>Bestimmt die Form der S-Kurve <ul style="list-style-type: none"> <li>• Kleine Werte <math>\Rightarrow</math> kleiner S-Verschleiß</li> <li>• Große Werte <math>\Rightarrow</math> großer S-Verschleiß</li> </ul> | 2-73         |



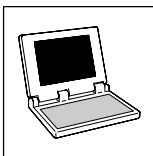
# Funktionsbibliothek LenzeDrive.lib

## Anhang

| Code  | LCD             | Einstellmöglichkeiten |                                                                                                  | WICHTIG                                                                                                                                                                                   | Info |
|-------|-----------------|-----------------------|--------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------|
|       |                 | Lenze                 | Auswahl                                                                                          |                                                                                                                                                                                           |      |
| C0190 | byArithFunction | 0                     | 0 OUT = C46<br>1 C46 + C49<br>2 C46 - C49<br>3 C46 * C49<br>4 C46 / C49<br>5 C46 / (100 - C49)   | Arithmetikblock im Funktionsblock L_NSET1<br>• Verknüpft Hauptsollwert C0046 und Zusatzsollwert C0049.                                                                                    | 2-73 |
| C0220 | dnTirAdd        | 0.000                 | 0.000 {0.001 s} 999.900                                                                          | Hochlaufzeit $T_{ir}$ des Zusatzsollwertes für L_NSET1<br>• Bezogen auf Drehzahländerung 0 ... $n_{max}$                                                                                  | 2-73 |
| C0221 | dnTifAdd        | 0.000                 | 0.000 {0.001 s} 999.900                                                                          | Ablaufzeit $T_{if}$ des Zusatzsollwertes für L_NSET1<br>• Bezogen auf Drehzahländerung $n_{max}$ ... 0.                                                                                   | 2-73 |
| C0222 | nVp             | 1.0                   | 0.1 {0.1} 500.0                                                                                  | Verstärkung $V_p$ von L_PCTRL1                                                                                                                                                            | 2-79 |
| C0223 | nTn             | 400                   | 20 {1 ms} 99999 ms abgeschaltet                                                                  | Integralanteil $T_n$ von L_PCTRL1                                                                                                                                                         | 2-79 |
| C0224 | nKd             | 0.0                   | 0.0 {0.1} 5.0                                                                                    | Differentialanteil $K_d$ von L_PCTRL1                                                                                                                                                     | 2-79 |
| C0241 | niEq0Hysteresis | 1.00                  | 0.00 {0.01 %} 100.00<br>100 % = $n_{max}$                                                        | Schwelle Hochlaufgeber für Hauptsollwert von L_NSET1<br>Eingang = Ausgang                                                                                                                 | 2-73 |
| C0260 | nHighLimit      | 100.00                | -199.99 {0.01 %} 199.99                                                                          | Obere Grenze von L_MPOT1<br>• Es muss gelten: C0260 > C0261                                                                                                                               | 2-70 |
| C0261 | nLowLimit       | -100.0                | -199.99 {0.01 %} 199.99                                                                          | Untere Grenze von L_MPOT1<br>• Es muss gelten: C0261 < C0260                                                                                                                              | 2-70 |
| C0262 | wTir            | 10.0                  | 0.1 {0.1 s} 6000.0                                                                               | Hochlaufzeit $T_{ir}$ von L_MPOT1<br>• Bezogen auf Änderung 0 ... 100 %.                                                                                                                  | 2-70 |
| C0263 | wTif            | 10.0                  | 0.1 {0.1 s} 6000.0                                                                               | Ablaufzeit $T_{if}$ von L_MPOT1<br>• Bezogen auf Änderung 0 ... 100 %.                                                                                                                    | 2-70 |
| C0264 | byFunction      | 0                     |                                                                                                  | Deaktivierungsfunktion von L_MPOT1<br>• Funktion, die ausgeführt wird, wenn Motorpoti über Eingang MPOT1-IN-ACTIVE deaktiviert wird.                                                      | 2-70 |
|       |                 |                       | 0 No function<br>1 Down to 0 %<br>2 Down to C261<br>3 Jump 0 %<br>4 Jump to C261<br>5 Up to C260 | Keine Änderung<br>Ablauf mit $T_{if}$ auf 0 %<br>Ablauf mit $T_{if}$ auf C0261<br>Sprung mit $T_{if}$ = 0 auf 0 %<br>Sprung mit $T_{if}$ = 0 auf C0261<br>Hochlauf mit $T_{ir}$ auf C0260 | 2-70 |
| C0265 | byInitFunction  | 0                     |                                                                                                  | Initialisierungsfunktion von L_MPOT1<br>• Wert, der beim Netzschalten und aktiviertem Motorpoti übernommen wird.                                                                          | 2-70 |
|       |                 |                       | 0 Power off<br>1 C0261<br>2 0 %                                                                  | Wert bei Netzausfall<br>Unterer Grenzwert aus C0261: 0 %                                                                                                                                  | 2-70 |
| C0325 | nVp2Vdapt       | 1.0                   | 0.1 {0.1} 500.0                                                                                  | Adaption Verstärkung ( $V_{p2}$ ) von L_PCTRL1                                                                                                                                            | 2-79 |
| C0326 | nVp3Adapt       | 1.0                   | 0.1 {0.1} 500.0                                                                                  | Adaption Verstärkung ( $V_{p3}$ ) von L_PCTRL1                                                                                                                                            | 2-79 |
| C0327 | nSet2Adapt      | 100.00                | 0.00 {0.01 %} 100.00                                                                             | Adaption $n_{soll2}$ von L_PCTRL1<br>Solldrehzahlschwelle der Prozessregleradaption<br>• Es muss gelten: C0327 > C0328                                                                    | 2-79 |
| C0328 | nSet1Adapt      | 0.00                  | 0.00 {0.01 %} 100.00                                                                             | Adaption $n_{soll1}$ von L_PCTRL1<br>Solldrehzahlschwelle der Prozessregleradaption<br>• Es muss gelten: C0328 < C0327                                                                    | 2-79 |



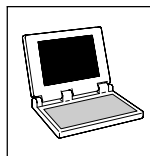
| Code  | LCD               | Einstellmöglichkeiten |                                                                                                                                                        | WICHTIG                                                                                                                                          | Info         |
|-------|-------------------|-----------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------|--------------|
|       |                   | Lenze                 | Auswahl                                                                                                                                                |                                                                                                                                                  |              |
| C0329 | byPCharacteristic | 0                     | 0 no<br>1 Extern Vp<br>2 Set-value<br>3 Ctrl diff                                                                                                      | Adaption von L_PCTRL1 aktivieren<br>keine Prozessregleradaption<br>extern über Eingang<br>Adaption über Sollwert<br>Adaption über Regeldifferenz | 2-79<br>2-79 |
| C0332 | PCTRLdnTir        | 0.000                 | 0.000 {0.001 s} 999.900                                                                                                                                | Hochlaufzeit $T_{ir}$ von L_PCTRL1<br>• Bezogen auf Sollwertänderung 0 ... 100 %.                                                                | 2-79         |
| C0333 | dnTif             | 0.000                 | 0.000 {0.001 s} 999.900                                                                                                                                | Ablaufzeit $T_{if}$ von L_PCTRL1<br>• Bezogen auf Sollwertänderung 0 ... 100 %.                                                                  | 2-79         |
| C0337 | bBiUnipolar       | 0                     | 0 bipolar<br>1 unipolar                                                                                                                                | Wirkungsbereich bipolar/unipolar von L_PCTRL1                                                                                                    | 2-79         |
| C0338 | byFunction        | 1                     | 0 $OUT = nln1\_a$<br>1 $nln1\_a + nln2\_a$<br>2 $nln1\_a - nln2\_a$<br>3 $nln1\_a * nln2\_a$<br>4 $nln1\_a / nln2\_a$<br>5 $nln1\_a / (100 - nln2\_a)$ | Funktion Arithmetikblock L_ARIT1<br>• Verknüpft Eingänge $nln1\_a$ und $nln2\_a$ .                                                               | 2-11         |
| C0560 | anSollW1          | 0.00                  | -199.99 {0.01 %} 199.99                                                                                                                                | Festsollwerte von L_FIXSET1                                                                                                                      | 2-2          |
| ...   | ...               | ...                   |                                                                                                                                                        |                                                                                                                                                  |              |
| 14    | anSollW14         | 0.00                  |                                                                                                                                                        |                                                                                                                                                  |              |
| 15    | anSollW15         | 0.00                  |                                                                                                                                                        |                                                                                                                                                  |              |
| C0600 | byFunction        | 1                     | 0 $OUT = nln1\_a$<br>1 $nln1\_a + nln2\_a$<br>2 $nln1\_a - nln2\_a$<br>3 $nln1\_a * nln2\_a$<br>4 $nln1\_a / nln2\_a$<br>5 $nln1\_a / (100 - nln2\_a)$ | Funktion Arithmetikblock L_ARIT2<br>• Verknüpft Eingänge $nln1\_a$ und $nln2\_a$ .                                                               | 2-11         |
| C0620 | nGain             | 1.00                  | -10.00 {0.01} 10.00                                                                                                                                    | Verstärkung Totgangglied L_DB1                                                                                                                   | 2-20         |
| C0621 | nDeadBand         | 1.00                  | 0.00 {0.01 %} 100.00                                                                                                                                   | Totgang von L_DB1                                                                                                                                | 2-20         |
| C0630 | nMaxLimit         | 100.00                | -199.99 {0.01 %} 199.99                                                                                                                                | Obere Grenze des Begrenzers L_LIM1                                                                                                               | 2-22         |
| C0631 | nMinLimit         | -100.0                | -199.99 {0.01 %} 199.99                                                                                                                                | Untere Grenze des Begrenzers L_LIM1                                                                                                              | 2-22         |
| C0640 | nDelayTime        | 20.00                 | 0.01 {0.01 s} 50.00                                                                                                                                    | Zeitkonstante von L_PT1_1                                                                                                                        | 2-23         |
| C0650 | nGain             | 1.00                  | -320.00 {0.01} 320.00                                                                                                                                  | Verstärkung von L-DT1_1                                                                                                                          | 2-21         |
| C0651 | nDelayTime        | 1.000                 | 0.005 {0.001 s} 5.000                                                                                                                                  | Zeitkonstante von L_DT1_1                                                                                                                        | 2-21         |
| C0653 | bySensibility     | 1                     | 1 15 Bit<br>2 14 Bit<br>3 13 Bit<br>4 12 Bit<br>5 11 Bit<br>6 10 Bit<br>7 9 Bit                                                                        | Eingangsempfindlichkeit von L_DT1_1                                                                                                              | 2-21         |
| C0655 | nNumerator        | 1                     | -32767 {1} 32767                                                                                                                                       | Zähler für L_CONV5                                                                                                                               | 2-49         |
| C0656 | nDenominator      | 1                     | 1 {1} 32767                                                                                                                                            | Nenner für L_CONV5                                                                                                                               | 2-49         |
| C0671 | dnTir             | 0.000                 | 0.000 {0.01 s} 999.900                                                                                                                                 | Hochlaufzeit $T_{ir}$ von Hochlaufgeber L_RFG1                                                                                                   | 2-24         |
| C0672 | dnTif             | 0.000                 | 0.000 {0.01 s} 999.900                                                                                                                                 | Ablaufzeit $T_{if}$ von L_RFG1                                                                                                                   | 2-24         |
| C0680 | byFunction        | 6                     | 1 $nln1 = nln2$<br>2 $nln1 > nln2$<br>3 $nln1 < nln2$<br>4 $lnln1 = lnln2$<br>5 $lnln1 > lnln2$<br>6 $lnln1 < lnln2$                                   | Funktion Komparator L_CMP1<br>• Vergleicht die Eingänge $nln1$ und $nln2$                                                                        | 2-13         |
| C0681 | nHysteresis       | 1.00                  | 0.00 {0.01 %} 100.00 %                                                                                                                                 | Hysteresis von L_CMP1                                                                                                                            | 2-13         |
| C0682 | nWindow           | 1.00                  | 0.00 {0.01 %} 100.00 %                                                                                                                                 | Fenster von L_CMP1                                                                                                                               | 2-13         |



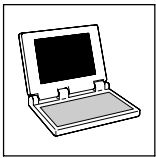
# Funktionsbibliothek LenzeDrive.lib

## Anhang

| Code  | LCD          | Einstellmöglichkeiten |                            |                                                                                                   | WICHTIG                                                                                                       | Info                                                    |        |
|-------|--------------|-----------------------|----------------------------|---------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------|---------------------------------------------------------|--------|
|       |              | Lenze                 | Auswahl                    |                                                                                                   |                                                                                                               |                                                         |        |
| C0685 | byFunction   | 1                     | 1<br>2<br>3<br>4<br>5<br>6 | nln1 = nln2<br>nln1 > nln2<br>nln1 < nln2<br> nln1  =  nln2 <br> nln1  >  nln2 <br> nln1  <  nln2 | Funktion Komparator L_CMP2<br>• Vergleicht die Eingänge <i>nln1</i> und <i>nln2</i>                           |                                                         |        |
| C0686 | nHysteresis  | 1.00                  | 0.00                       | {0.01 %}                                                                                          | 100.00 %                                                                                                      | Hysteresis von L_CMP2                                   | 📖 2-13 |
| C0687 | nWindow      | 1.00                  | 0.00                       | {0.01 %}                                                                                          | 100.00 %                                                                                                      | Fenster von L_CMP2                                      | 📖 2-13 |
| C0690 | byFunction   | 1                     | 1<br>2<br>3<br>4<br>5<br>6 | nln1 = nln2<br>nln1 > nln2<br>nln1 < nln2<br> nln1  =  nln2 <br> nln1  >  nln2 <br> nln1  <  nln2 | Funktion Komparator L_CMP3<br>• Vergleicht die Eingänge <i>nln1</i> und <i>nln2</i>                           |                                                         |        |
| C0691 | nHysteresis  | 1.00                  | 0.00                       | {0.01 %}                                                                                          | 100.00 %                                                                                                      | Hysteresis von L_CMP3                                   | 📖 2-13 |
| C0692 | nWindow      | 1.00                  | 0.00                       | {0.01 %}                                                                                          | 100.00 %                                                                                                      | Fenster von L_CMP3                                      | 📖 2-13 |
| C0695 | byFunction   | 2                     | 1<br>2                     | dnln1_p < dnln2_p<br> dnln1_pl  <  dnln2_pl                                                       | Funktion Komparator für Winkelsignale L_PHCMP1<br>• Vergleicht die Eingänge <i>dnln1_p</i> und <i>dnln2_p</i> | 📖 2-40                                                  |        |
| C0710 | byFunction   | 0                     | 0<br>1<br>2                | Rising trans<br>Falling trans<br>Both trans                                                       | Funktion Flankenbewertung von L_TRANS1                                                                        | 📖 2-36                                                  |        |
| C0711 | wPulseTime   | 0.001                 | 0.001                      | {0.001 s}                                                                                         | 60.000                                                                                                        | Impulsdauer von TRANS1                                  | 📖 2-36 |
| C0715 | byFunction   | 0                     | 0<br>1<br>2                | Rising trans<br>Falling trans<br>Both trans                                                       | Funktion Flankenbewertung von L_TRANS2                                                                        | 📖 2-36                                                  |        |
| C0716 | wPulseTime   | 0.001                 | 0.001                      | {0.001 s}                                                                                         | 60.000                                                                                                        | Impulsdauer von L_TRANS2                                | 📖 2-36 |
| C0720 | byFunction   | 2                     | 0<br>1<br>2                | On delay<br>Off delay<br>On/Off delay                                                             | Funktion digitales Verzögerungsglied L_DIGDEL1                                                                | 📖 2-30                                                  |        |
| C0721 | wDelayTime   | 1.000                 | 0.001                      | {0.001 s}                                                                                         | 60.000                                                                                                        | Verzögerungszeit von L_DIGDEL1                          | 📖 2-30 |
| C0725 | byFunction   | 0                     | 0<br>1<br>2                | On delay<br>Off delay<br>On/Off delay                                                             | Funktion digitales Verzögerungsglied L_DIGDEL2                                                                | 📖 2-30                                                  |        |
| C0726 | wDelayTime   | 1.000                 | 0.001                      | {0.001 s}                                                                                         | 60.000                                                                                                        | Verzögerungszeit von L_DIGDEL2                          | 📖 2-30 |
| C0940 | nNumerator   | 1                     | -32767                     | {1}                                                                                               | 32767                                                                                                         | Zähler für L_CONV1                                      | 📖 2-49 |
| C0941 | nDenominator | 1                     | 1                          | {1}                                                                                               | 32767                                                                                                         | Nenner für L_CONV1                                      | 📖 2-49 |
| C0945 | nNumerator   | 1                     | -32767                     | {1}                                                                                               | 32767                                                                                                         | Zähler für L_CONV2                                      | 📖 2-49 |
| C0946 | nDenominator | 1                     | 1                          | {1}                                                                                               | 32767                                                                                                         | Nenner für L_CONV2                                      | 📖 2-49 |
| C0950 | nNumerator   | 1                     | -32767                     | {1}                                                                                               | 32767                                                                                                         | Zähler für L_CONV3                                      | 📖 2-49 |
| C0951 | nDenominator | 1                     | 1                          | {1}                                                                                               | 32767                                                                                                         | Nenner für L_CONV3                                      | 📖 2-49 |
| C0955 | nNumerator   | 1                     | -32767                     | {1}                                                                                               | 32767                                                                                                         | Zähler für L_CONV4                                      | 📖 2-49 |
| C0956 | nDenominator | 1                     | 1                          | {1}                                                                                               | 32767                                                                                                         | Nenner für L_CONV4                                      | 📖 2-49 |
| C0960 | byFunction   | 1                     | 1<br>2<br>3                | Function 1<br>Function 2<br>Function 3                                                            | Kennlinie CURVE1-IN                                                                                           | 📖 2-17                                                  |        |
| C0961 | ny0          | 0.00                  | 0.00                       | {0.01 %}                                                                                          | 199.99                                                                                                        | Ordinate des Wertepaars (x = 0 % / y0) von L_CURVE1     | 📖 2-17 |
| C0962 | ny1          | 50.00                 | 0.00                       | {0.01 %}                                                                                          | 199.99                                                                                                        | Ordinate des Wertepaars (x1 / y1) von L_CURVE1          | 📖 2-17 |
| C0963 | ny2          | 75.00                 | 0.00                       | {0.01 %}                                                                                          | 199.99                                                                                                        | Ordinate des Wertepaars (x2 / y2) von L_CURVE1          | 📖 2-17 |
| C0964 | ny100        | 100.00                | 0.00                       | {0.01 %}                                                                                          | 199.99                                                                                                        | Ordinate des Wertepaars (x = 100 % / y100) von L_CURVE1 | 📖 2-17 |
| C0965 | nx1          | 50.00                 | 0.01                       | {0.01 %}                                                                                          | 100.00                                                                                                        | Abszisse des Wertepaars (x1 / y1) von L_CURVE1          | 📖 2-17 |

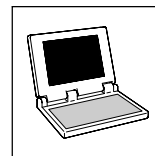


| Code  | LCD          | Einstellmöglichkeiten |                                                                                                                                                                                    | WICHTIG                                                                                         | Info         |
|-------|--------------|-----------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------|--------------|
|       |              | Lenze                 | Auswahl                                                                                                                                                                            |                                                                                                 |              |
| C0966 | nx2          | 75.00                 | 0.01 {0.01 %} 99.00                                                                                                                                                                | Abszisse des Wertepaars (x2 / y2) von L_CURVE1                                                  | 2-17         |
| C0995 | byDivision   | 0                     | -31 {1} 31                                                                                                                                                                         | Teilfaktor Winkelteiler L_PHDIV1                                                                | 2-42         |
| C1000 | nDivision    | 1                     | 0 {1} 31                                                                                                                                                                           | Teilfaktor                                                                                      | 2-50         |
| C1010 | byFunction   | 1                     | 0 OUT = dnln1_p<br>1 dnln1_p + dnln2_p<br>2 dnln1_p - dnln2_p<br>3 dnln1_p * dnln2_p<br>14 dnln1_p / dnln2_p<br>21 dnln1_p + dnln2_p (no limit)<br>22 dnln1_p - dnln2_p (no limit) | Funktion von L_ARITPH1                                                                          | 2-38         |
| C1040 | dwTi         | 100.000               | 0.001 {0.001 %} 5000.000                                                                                                                                                           | Beschleunigung von L_SRFG1                                                                      | 2-28         |
| C1041 | dwJerk       | 0.200                 | 0.001 {0.001 s} 999.999                                                                                                                                                            | Ruck von L_SRFG1                                                                                | 2-28         |
| C1100 | byFunction   | 1                     | 1 Return<br>2 Hold                                                                                                                                                                 | Funktion von L_FCNT1                                                                            | 2-32         |
| C1140 | byFunction   | 0                     | 0 Rising trans<br>1 Falling trans<br>2 Both trans                                                                                                                                  | Funktion Flankenwertung von L_TRANS3                                                            | 2-36         |
| C1141 | wPulseTime   | 0.001                 | 0.001 {0.001 s} 60.000                                                                                                                                                             | Impulsdauer von L_TRANS3                                                                        | 2-36         |
| C1145 | byFunction   | 0                     | 0 Rising trans<br>1 Falling trans<br>2 Both trans                                                                                                                                  | Funktion Flankenwertung von L_TRANS4                                                            | 2-36         |
| C1146 | wPulseTime   | 0.001                 | 0.001 {0.001 s} 60.000                                                                                                                                                             | Impulsdauer von L_TRANS4                                                                        | 2-36         |
| C1150 | byMode       | 0                     | 0 Load perm<br>1 Load edge<br>2 Cmp & sub                                                                                                                                          | Funktion von L_PHINTK                                                                           | 2-43         |
| C1151 | dnCmp        | 2 · 10 <sup>9</sup>   | 0 {1} 2000000000                                                                                                                                                                   | Vergleichswert von L_PHINTK                                                                     | 2-43         |
| C1170 | nNumerator   | 1                     | -32767 {1} 32767                                                                                                                                                                   | Zähler für L_CONV6                                                                              | 2-49         |
| C1171 | nDenominator | 1                     | 1 {1} 32767                                                                                                                                                                        | Nenner für L_CONV6                                                                              | 2-49         |
| C1207 | byFunction   | 2                     | 1 dnln1_p < dnln2_p<br>2 ldln1_pl < ldln2_pl                                                                                                                                       | Funktion Komparator für Winkelsignale L_PHCMP2<br>• Vergleicht die Eingänge dnln1_p und dnln2_p | 2-40         |
| C1272 | byFunction   | 2                     | 1 dnln1_p < dnln2_p<br>2 ldln1_pl < ldln2_pl                                                                                                                                       | Funktion Komparator für Winkelsignale L_PHCMP3<br>• Vergleicht die Eingänge dnln1_p und dnln2_p | 2-40         |
| C2118 |              | 0                     | 0 Kommunikation über 2 freie PDO-Kanäle<br>1 Kommunikation über SDO 2-Kanal                                                                                                        | Kanalwahl für Kommunikation über Systembus mit L_ParWrite/L_ParRead                             | 2-55<br>2-59 |



# ***Funktionsbibliothek LenzeDrive.lib***

## ***Anhang***



## 4 Index

### A

Addition (L\_ADD), 2-5  
Anhang, 3-1  
Arithmetik (L\_ARIT), 2-11  
Arithmetik (L\_ARITPH), 2-38  
ASCII-Zeichentabelle, 2-66  
Auf-/Abwärtszähler (L\_FCNT), 2-32  
Ausgangsverstärkung und Offset (L\_AOUT), 2-9

### B

Bedieneinheit, 2-65  
Begrenzung (L\_LIM), 2-22  
Begriffsdefinitionen, 1-2  
Beschreiben von Codestellen (L\_ParWrite), 2-59  
Betragsbildung (L\_ABS), 2-4

### C

Codetabelle, 3-1

### D

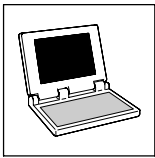
Datentypangabe, Erläuterung zu, 1-4  
Differenzierung (L\_DT1\_), 2-21  
Division (L\_PHDIV), 2-42  
Drehzahl-Aufbereitung (L\_NSET), 2-73  
Drehzahl-Vorverarbeitung (L\_NSET)  
    Haupsollwert, 2-74  
    JOG-Sollwerte, 2-75  
    S-Rampe, PT1-Glied, 2-77  
    Sollwert-Invertierung, Hochlaufgeber, Haupsollwert, 2-75  
    Zusatzsollwert, 2-78

### E

Eingangsverstärkung und Offset (L\_AIN), 2-6

### F

Fehlerrückmeldung (L\_FWM), 2-68  
Flankenauswertung (L\_TRANS), 2-36  
Funktionsblöcke  
    Addition (L\_ADD), 2-5  
    Addition (L\_PHADD), 2-39  
    Arithmetik (L\_ARIT), 2-11  
    Arithmetik (L\_ARITPH), 2-38  
    Auf-/Abwärtszähler (L\_FCNT), 2-32  
    Ausgangsverstärkung und Offset (L\_AOUT), 2-9  
    Begrenzung (L\_LIM), 2-22  
    Betragsbildung (L\_ABS), 2-4  
    Differenz (L\_PHDIFF), 2-41  
    Differenzierung (L\_DT1\_), 2-21  
    Division (L\_PHDIV), 2-42  
    Drehzahl-Aufbereitung (L\_NSET), 2-73  
    Eingangsverstärkung und Offset (L\_AIN), 2-6  
    Fehlerrückmeldung (L\_FWM), 2-68  
    Flankenauswertung (L\_TRANS), 2-36  
    Flip-Flop (L\_FLIP), 2-33  
    Hochlaufgeber (L\_RFG), 2-24  
    Hochlaufgeber S-Rampe (L\_SRFG), 2-27  
    Integration (L\_PHINT), 2-43  
    Integration (L\_PHINTK), 2-45  
    Invertierung (L\_ANEG), 2-8  
    Kennlinienfunktion (L\_CURVE), 2-17  
    Konvertierung Winkel nach Analog (L\_CONVPA), 2-50  
    L\_ByteArrayToDint, 2-54  
    L\_DintToByteArray, 2-54  
    L\_FUNCodeIndexConv, 2-54  
    L\_ParRead, 2-55  
    L\_ParWrite, 2-59  
    Logisches NICHT (L\_NOT), 2-34  
    Logisches ODER (L\_OR), 2-35  
    Logisches UND (L\_AND), 2-29  
    Motorpotentiometer (L\_MPOT), 2-70  
    Normierung (L\_CONV), 2-49  
    Normierung mit Begrenzung (L\_CONVX), 2-53  
    Programmierung Fix-Sollwerte (L\_FIXSET), 2-2  
    Prozessregler (L\_PCTRL), Tänzerlager-, Zug-, Druckregler, 2-79  
    Rechts/Links/Quickstop (L\_RLQ), 2-83  
    Sample & Hold (L\_SH), 2-26  
    Totgang (L\_DB), 2-20  
    Transparenter Modus mit EMZ9371BB/BC (L\_Display9371BB), 2-63  
    Umrechnung (L\_CONVW), 2-52  
    Umrechnung Winkelsignal (L\_CONVPP), 2-51  
    Umschaltung (L\_ASW), 2-12  
    Vergleich (L\_CMP), 2-13  
    Vergleich (L\_PHCMP), 2-40  
    Verzögerung (L\_DIGDEL), 2-30  
    Verzögerung (L\_PT1\_), 2-23



# Funktionsbibliothek LenzeDrive.lib

## Index

### H

Hauptsollwertpfad, 2-74  
Hochlaufgeber, 2-82  
Hochlaufgeber (L\_RFG), 2-24  
Hochlaufgeber S-Rampe (L\_SRFG), 2-27

### I

Identifikator, Erläuterung zu, 1-4  
Index einer Codestelle (L\_FUNCodelIndexConv), 2-54  
Integration (L\_PHINT), 2-43  
    Ausgangssignal berechnen, 2-44  
    konstanter Eingangswert, 2-43  
Integration (L\_PHINTK), 2-45  
    Ausgangssignal berechnen, 2-48  
    Eingangswert mit Vorzeichenwechsel, 2-47  
    konstanter Eingangswert, 2-46  
Invertierung (L\_ANEG), 2-8  
Istwinkelintegrator (L\_PHDIFF), 2-41

### J

JOG-Sollwerte, 2-75

### K

Kennlinienfunktion (L\_CURVE), 2-17  
Konvertierung Winkel nach Analog (L\_CONVPA), 2-50

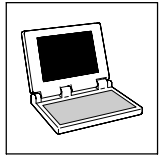
### L

L\_ABS, 2-4  
L\_ADD, 2-5  
L\_AIN, 2-6  
L\_AND, 2-29  
L\_ANEG, 2-8  
L\_AOUT, 2-9  
L\_ARIT, 2-11  
L\_ARITPH, 2-38  
L\_ASW, 2-12  
L\_ByteArrayToDint, 2-54  
L\_CMP, 2-13  
L\_CONV, 2-49  
L\_CONVPA, 2-50  
L\_CONVPP, 2-51  
L\_CONVVV, 2-52  
L\_CONVX, 2-53

L\_CURVE, 2-17  
L\_DB, 2-20  
L\_DIGDEL, 2-30  
L\_DintToByteArray, 2-54  
L\_Display9371BB, 2-63  
L\_DT1\_, 2-21  
L\_FCNT, 2-32  
L\_FIXSET, 2-2  
L\_FLIP, 2-33  
L\_FUNCodelIndexConv, 2-54  
L\_FWM, 2-68  
L\_LIM, 2-22  
L\_MPOT, 2-70  
L\_NOT, 2-34  
L\_NSET, 2-73  
L\_OR, 2-35  
L\_ParRead, 2-55  
L\_ParWrite, 2-59  
L\_PHADD, 2-39  
L\_PHCMP, 2-40  
L\_PHDIFF, 2-41  
L\_PHDIV, 2-42  
L\_PHINT, 2-43  
L\_PHINTK, 2-45  
L\_PT1\_, 2-23  
L\_RFG, 2-24  
L\_RLQ, 2-83  
L\_SH, 2-26  
L\_SRFG, 2-27  
L\_TRANS, 2-36  
Lenze-Softwarerichtlinie, Ungarische Notation, 1-3  
Lesen von Codestellen (L\_ParRead), 2-55  
Logisches NICHT (L\_NOT), 2-34  
Logisches ODER (L\_OR), 2-35  
Logisches UND (L\_AND), 2-29

### M

Motorpotentiometer (L\_MPOT), 2-70



## N

Normierung (L\_CONV), 2-49

Normierung mit Begrenzung (L\_CONVX), 2-53

## P

Präfix, Erläuterung zu, 1-3

Programmierung Fix-Sollwerte (L\_FIXSET), 2-2

Prozessregler (L\_PCTRL)

Hochlaufgeber, 2-82

Regelcharakteristik, 2-80

Tänzerlager-, Zug-, Druckregler, 2-79

## R

Rechts/Links/Quickstop (L\_RLQ), 2-83

Regelcharakteristik, 2-80

## S

S-Rampe, PT1-Glied, 2-77

Sample & Hold (L\_SH), 2-26

Sicherheitshinweise, Gestaltung

Sonstige Hinweise, 1-2

Warnung vor Sachschäden, 1-2

Signaltyp, Erläuterung zu, 1-5

sin(2)-Form, 2-27

Sollwert-Invertierung, Hochlaufgeber, Hauptsollwert, 2-75

Speicherglied (FLIP), 2-33

Systemvariablen, Erläuterung zu, 1-5

## T

Totgang (L\_DB), 2-20

Transparenter Modus mit EMZ9371BB/BC  
(L\_Display9371BB), 2-63

Typ-Konvertierung (L\_ByteArrayToDint), 2-54

Typ-Konvertierung (L\_DintToByteArray), 2-54

## U

Umrechnung (L\_CONVVV), 2-52

Umrechnung Winkelsignal (L\_CONVPP), 2-51

Umschaltung (L\_ASW), 2-12

## V

Variablennamen

Konventionen, Ungarische Notation, 1-3

Lenze-Softwarerichtlinie, Erläuterung zu, 1-3

Variablentyp, Kennzeichnung, 1-4

Vergleich (L\_CMP), 2-13

Versionskennung der Funktionsbibliothek, 1-6

Verzögerung (L\_DIGDEL), 2-30

Verzögerung (L\_PT1\_), 2-23

## W

Winkeladditionsblock (L\_PHADD), 2-39

Winkelvergleicher (L\_PHCMP), 2-40

## Z

Zusatzsollwert, 2-78