

Lenze

Handbuch

IEC 61131-3

inside

***Global Drive
PLC Developer Studio***



Global Drive

Funktionsbibliothek

Lenze32BitTransferDrv.lib

Die Funktionsbibliothek **Lenze32BitTransferDrv.lib** kann für folgende Lenze PLC's verwendet werden:

	Typenbezeichnung	ab Hardwarestand	ab Softwarestand
9300 Servo PLC	EVS93XX-xI	6A	6.2
9300 Servo PLC	ECS93XX-xT	6A	6.2
Drive PLC	EPL10200	1A	6.2
ECSxA	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.

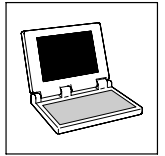
Wir übernehmen keine Verantwortung für direkt oder indirekt entstandene Schäden, z. B. Gewinnverluste, Auftragsverluste oder geschäftliche Beeinträchtigungen jeglicher Art.

© 2005 Lenze Drive Systems GmbH

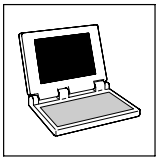
Ohne besondere schriftliche Genehmigung von Lenze Drive Systems GmbH darf kein Teil dieser Dokumentation vervielfältigt oder Dritten zugänglich gemacht werden.

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

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

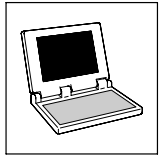


1	Vorwort und Allgemeines	1-1
1.1	Über dieses Handbuch	1-1
1.1.1	Konventionen in diesem Handbuch	1-1
1.1.2	Aufbau der Beschreibungen	1-2
1.1.3	Piktogramme in diesem Handbuch	1-2
1.1.4	Verwendete Begriffe	1-2
1.2	Versionskennung der Funktionsbibliothek	1-3
2	Funktionen	2-1
2.1	Übersicht	2-1
2.2	L_32BitTransferDINT	2-2
2.3	L_32BitTransferDWORD	2-3



Funktionsbibliothek Lenze32BitTransferDrv.lib

Inhalt



1 Vorwort und Allgemeines

1.1 Über dieses Handbuch

Dieses Handbuch enthält Informationen zur Funktionsbibliothek **Lenze32BitTransferDrv.lib** für das **Drive PLCDeveloper Studio**.

- In der Funktionsbibliothek **Lenze32BitTransferDrv.lib** sind Funktionen enthalten, die das datenkonsistente Kopieren von 32-Bit-Variablen auch bei einer Taskunterbrechung ermöglichen.

1.1.1 Konventionen in diesem Handbuch

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

Variablenbezeichner

sind im erklärenden Text in Kursivschrift dargestellt:

- "Mit *bReset...*"



Tip!

Informationen zu den Konventionen, die für die Variablenbezeichner von Lenze Systembausteinen, Funktionsblöcken sowie Funktionen verwendet werden, um eine einheitliche und durchgängige Benennung zu gewährleisten und dadurch die Lesbarkeit von SPS-Programmen zu fördern, finden Sie im Anhang der DDS-Online-Dokumentation "Einführung in die IEC61131-3 Programmierung".

Lenze Funktionen/Funktionsblöcke

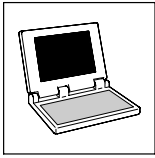
erkennen Sie am Namen. Sie beginnen immer mit einem "L_":

- "Mit der Funktion **L_TBConvBitsToByte...**"

Programmlistings

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

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



Funktionsbibliothek Lenze32BitTransferDrv.lib

Vorwort und Allgemeines

Über dieses Handbuch

1.1.2 Aufbau der Beschreibungen

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

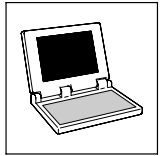
	Funktion	Funktionsblock (FB)/ Systembaustein (SB)
①	Überschrift mit Funktion und Bezeichner	
②	Deklaration der Funktion: • Datentyp des Rückgabewertes • Funktionsbezeichner • Liste der Übergabeparameter	-
③	Kurzbeschreibung mit den wichtigsten Eigenschaften	
④	Grafische Darstellung der Funktion mit den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

1.1.3 Piktogramme in diesem Handbuch

	verwendete Piktogramme	Signalwörter
Warnung vor Sachschäden		Stop! Warnt vor möglichen Sachschäden . Mögliche Folgen bei Mißachtung: Beschädigung des Antriebsreglers/Antriebssystems oder seiner Umgebung.
Sonstige Hinweise		Tip! Hinweis! Kennzeichnet einen Tip 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	9300 Servo PLC - Drive PLC - Achsmodul ECSxA "Application"
SB	Systembaustein



1.2 Versionskennung der Funktionsbibliothek

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

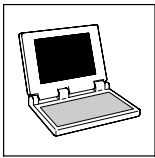
Versionskennung ab PLC Software-Stand 7.x:

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.

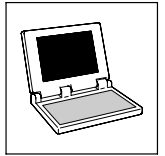
- Im Beispiel bedeutet der Wert "01050000" Version "1.05".



Funktionsbibliothek Lenze32BitTransferDrv.lib

Vorwort und Allgemeines

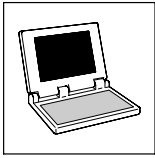
Versionskennung der Funktionsbibliothek



2 Funktionen

2.1 Übersicht

Funktion		Info
L_32BitTransferDINT	(FUN)	Datenkonsistentes Kopieren einer Variablen vom Typ "Double Integer" (32 Bit)  2-2
L_32BitTransferDWORD	(FUN)	Datenkonsistentes Kopieren einer Variablen vom Typ "Double Word" (32 Bit)  2-3



Funktionsbibliothek Lenze32BitTransferDrv.lib

Funktionen

L_32BitTransferDINT

2.2 L_32BitTransferDINT

Funktion

Mit dieser Funktion kann der Wert einer Variablen vom Typ "Double Integer" (32 Bit) datenkonsistent in eine andere Variable von Typ "Double Integer" kopiert werden.

- Die Funktion kann im Hauptprogramm PLC_PRG oder in einer User-Task aufgerufen werden.
- Beim Start der Funktion werden die Tasks gesperrt, damit ein datenkonsistenter Kopiervorgang stattfinden kann.
- Nach Abarbeitung der Funktion werden die gesperrten Tasks automatisch wieder freigegeben.
- Während der Abarbeitung einlaufende Taskereignisse werden nach der Abarbeitung der Funktion nachgeholt, so daß kein Startsignal verloren geht.

Deklaration	
<pre>BOOL L_32BitTransferDINT (pdnSource, pdnDestination);</pre>	

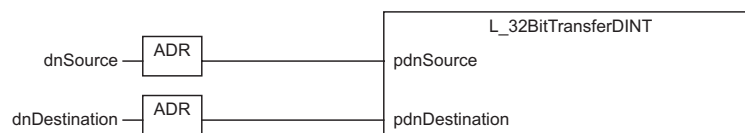
Übergabeparameter	Datentyp	Info/Einstellmöglichkeiten
pdnSource	Pointer to Double Integer	Adresse der zu kopierenden Variable vom Typ "Double Integer"
pdnDestination	Pointer to Double Integer	Adresse der Ziel-Variable vom Typ "Double Integer"

Rückgabewert	Datentyp	Wert/Bedeutung
	Bool	Status
		TRUE Datentransfer abgeschlossen

Beispiele

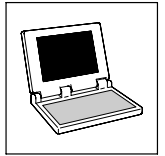
In den folgenden beiden Beispielen werden für die Dauer des Kopiervorganges die Tasks gesperrt und der Wert der Variablen *dnSource* wird in die Variable *dnDestination* kopiert.

- Aufruf der Funktion in FUP:



- Aufruf der Funktion in ST:

```
pdnSource := ADR(dnSource);
pdnDestination := ADR(dnDestination);
bTransfer := L_32BitTransferDINT (pdnSource,
                                   pdnDestination);
```



2.3 L_32BitTransferDWORD

Funktion

Mit dieser Funktion kann der Wert einer Variablen vom Typ "Double Word" (32 Bit) datenkonsistent in eine andere Variable von Typ "Double Word" kopiert werden.

- Die Funktion kann im Hauptprogramm PLC_PRG oder in einer User-Task aufgerufen werden.
- Beim Start der Funktion werden die Tasks gesperrt, damit ein datenkonsistenter Kopiervorgang stattfinden kann.
- Nach Abarbeitung der Funktion werden die gesperrten Tasks automatisch wieder freigegeben.
- Während der Abarbeitung einlaufende Taskereignisse werden nach der Abarbeitung der Funktion nachgeholt, so daß kein Startsignal verloren geht.

Deklaration	
BOOL	L_32BitTransferDWORD (pdwSource, pdwDestination);

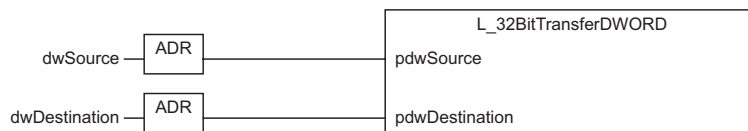
Übergabeparameter	Datentyp	Info/Einstellmöglichkeiten
pdwSource	Pointer to Double Word	Adresse der zu kopierenden Variable vom Typ "Double Word"
pdwDestination	Pointer to Double Word	Adresse der Ziel-Variable vom Typ "Double Word"

Rückgabewert	Datentyp	Wert/Bedeutung
	Bool	Status
		TRUE Datentransfer abgeschlossen

Beispiele

In den folgenden beiden Beispielen werden für die Dauer des Kopiervorganges die Tasks gesperrt und der Wert der Variablen *dwSource* wird in die Variable *dwDestination* kopiert.

- Aufruf der Funktion in FUP:



- Aufruf der Funktion in ST:

```
pdwSource := ADR(dwSource);
pdwDestination := ADR(dwDestination);
bTransfer := L_32BitTransferDWORD (pdwSource,
                                   pdwDestination);
```