Lenze

Manual

Global Drive
PLC Developer Studio

Global Drive

Function library

Lenze32BitTransferDrv.lib

The function library **Lenze32BitTransferDrv.lib** can be used for the following Lenze PLCs:

	Type	from hardware version	from software version
9300 Servo PLC	EVS93XX-xI	6A	6.2
9300 Servo PLC	EVS93XX-xT	6A	6.2
Drive PLC	EPL10200	1A	6.2
ECSxA	ECSxAxxx	1C	7.0

Important note:

The software is supplied to the user as described in this document. Any risks resulting from its quality or use remain the responsibility of the user. The user must provide all safety measures protecting against possible maloperation.

We do not take any liability for direct or indirect damage, e.g. profit loss, order loss or any loss regarding business.

All information given in this documentation has been carefully selected and tested for compliance with the hardware and software described. Nevertheless, discrepancies cannot be ruled out. We do not accept any responsibility or liability for any damage that may occur. Required corrections will be included in updates of this documentation.

All product names mentioned in this documentation are trademarks of the corresponding owners.

Version 1.1 03/2005

1 Preface and general information

1.1 About this Manual

This Manual contains information about the function library **Lenze32BitTransferDrv.lib** for the **Drive PLC Developer Studio**.

- The functions of the function library **Lenze32BitTransferDrv.lib** allow data-consistent copying of 32-bit variables even in case of a task interrupt.

1.1.1 Conventions used in this Manual

This Manual uses the following conventions to distinguish between different types of information:

Variable identifiers

are written in italics in the explanation:

- ”Use *bReset*...”

Tip!

Information about the conventions used for the variables of the Lenze system blocks, function blocks and functions can be found in the appendix of the DDS online documentation ”Introduction into IEC61131-3 programming”. The conventions ensure universal and uniform labelling and make reading the PLC program easier.

Lenze functions/function blocks

can be recognized by their names. They always begin with an ”L_”:

- ”With the function **L_TBConvBitsToByte**...”

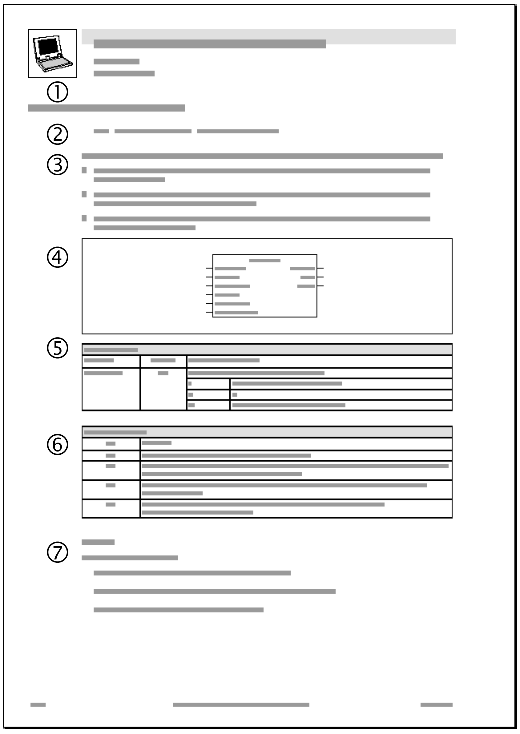
Program listings

are written in ”Courier”, keywords are printed in bold:

- ” `IF (ReturnValue < 0) THEN`...”

1.1.2 Layout of the descriptions

All function/function block and system block descriptions contained in this Manual have the same structure:

		Function	**Function block (FB)/ System block (SB)**
	①	Headline stating the function and the function identifier	
	②	Declaration of the function: • Data type of the return value • Function identifier • List of transfer parameters	-
	③	Short description of the most important properties	
	④	Function chart including all associated variables • Transfer parameters • Return value	FB/SB chart including all associated variables • Input variables • Output variables
	⑤	Table giving information about the transfer parameters: • Identifier • Data type • Possible settings • Info	Table giving information about the input and output variables: • Identifier • Data type • Type of variable • Possible settings • Info
	⑥	Table giving information about the return value: • Data type of the return value • Possible return values and their meaning:	-
	⑦	Additional information (Notes, tips, application examples, etc.)	

1.1.3 Pictographs used in this Manual

	Use of pictograms	**Signal words**	
Warning of material damage	STOP	**Stop!**	Warns of **potential damage to material**. Possible consequences if disregarded: Damage to the controller/drive system or its environment.
Other notes	i	**Tip!** **Note!**	Indicates a tip or note.

1.1.4 Terminology used

Term	**In the following text used for**
DDS	Drive PLC Developer Studio
FB	Function block
GDC	Global Drive Control (parameterization program from Lenze)
Parameter codes	Codes for setting the functionality of a function block
PLC	• 9300 Servo PLC • Drive PLC • ECSxA ”Application” axis module
SB	System block

1.2 Version identifiers of the function library

The version of the function library can be found under the global constant `C_w[Function library name]Version`.

Version identifiers as of PLC software version 7.x:

Constant	Meaning	Example value
`C_w[FunctionLibraryName]VersionER`	External Release	01
`C_w[FunctionLibraryName]VersionEL`	External Level	05
`C_w[FunctionLibraryName]VersionIR`	Internal Release	00
`C_w[FunctionLibraryName]VersionBN`	Build No.	00

Version: 01 05 00 00

The value of this constant is a hexadecimal code.

- In the example, ”01050000” stands for version ”1.05”.

2 Functions

2.1 Overview

Function		Info	
L_32BitTransferDINT	(FUN)	Data-consistent copying of variables of type “double integer” (32 bits)	📖 2-2
L_32BitTransferDWORD	(FUN)	Data-consistent copying of variables of type “double word” (32 bits)	📖 2-3

2.2 L_32BitTransferDINT

Function

This function is used to copy the value of variables of type “double integer” (32 bits) data-consistently to other variables of type “double integer”.

- The function can be called in the main program PLC_PRG or in a user task.
- When the function is started, the tasks are inhibited to ensure a data-consistent copying process.
- After copying has been completed, the blocked tasks will be automatically released.
- Task events coming in during copying will be processed as soon as the function has been executed. This ensures that no start signals are lost.

Declaration
`BOOL L_32BitTransferDINT (pdnSource, pdnDestination);`

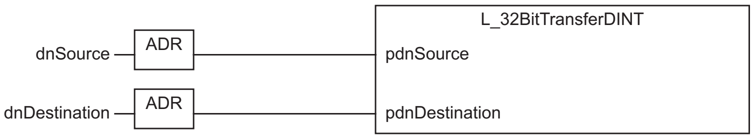
Transfer parameters	Data type	Info/possible settings
pdnSource	Pointer to double integer	Address of the variable of type ”double integer” to be copied
pdnDestination	Pointer to double integer	Address of the destination variable of type ”double integer”

Return value	Data type	Value/meaning	
	Bool	Status	
		TRUE	Data transfer has been completed.

Examples

In the following examples, tasks are blocked during copying and the value of the variable *dnSource* is copied to the variable *dnDestination*.

- Calling the function in FBD:



- Calling the function in ST:

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

2.3 L_32BitTransferDWORD

Function

This function is used to copy the value of variables of type “double word” (32 bits) data-consistently to other variables of type “double word”.

- The function can be called in the main program PLC_PRG or in a user task.
- When the function is started, the tasks are inhibited to ensure a data-consistent copying process.
- After copying has been completed, the blocked tasks will be automatically released.
- Task events coming in during copying will be processed as soon as the function has been executed. This ensures that no start signals are lost.

Declaration
`BOOL L_32BitTransferDWORD (pdwSource, pdwDestination);`

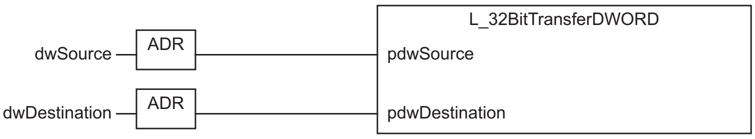
Transfer parameters	Data type	Info/possible settings
pdwSource	Pointer to double word	Address of the variable of type ”double word” to be copied
pdwDestination	Pointer to double word	Address of the destination variable of type ”double word”

Return value	Data type	Value/meaning	
	Bool	Status	
		TRUE	Data transfer has been completed.

Examples

In the following examples, tasks are blocked during copying and the value of the variable *dwSource* is copied to the variable *dwDestination*.

- Calling the function in FBD:



- Calling the function in ST:

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