Lenze

Manual

Global Drive

Drive PLC

This Manual applies to the following PLCs:

	Type	From hardware version	From software version
Drive PLC	EPL10200	Px	6.0

What's new?

Version		Changes
1.1	07/2000	Revised edition for the Drive PLC as of software version V1.0
2.1	07/2001	Revised edition for the Drive PLC as of software version V2.0
3.0	06/2003	Revised edition for the Drive PLC as of software version V6.0

Important note:

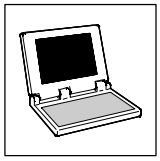
The software is supplied to the user as described in this document. Any risks resulting from its quality or use remain the responsibility of the user. The user must provide all safety measures protecting against possible maloperation.

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

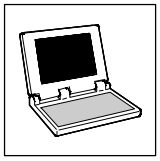
All information given in this online documentation has been carefully selected and tested for compliance with the hardware and software described. Nevertheless, discrepancies cannot be ruled out. We do not accept any responsibility or liability for any damage that may occur. Required corrections will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

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

Version 3.0 06/2003 TD11



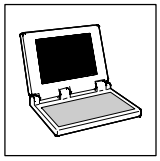
* see Manual “System bus (CAN) for Servo PLC & Drive PLC”

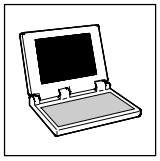


Drive PLC

Contents

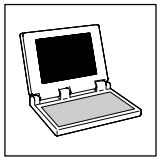
* see Manual “System bus (CAN) for Servo PLC & Drive PLC”





Drive PLC

Contents



1 Preface and general information

1.1 About this Manual

This Manual describes the system block functions which can be selected and parameterised in the control configuration of the **Drive PLC Developer Studio** (DDS) for the **Drive PLC.**

1.1.1 Conventions used in this Manual

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

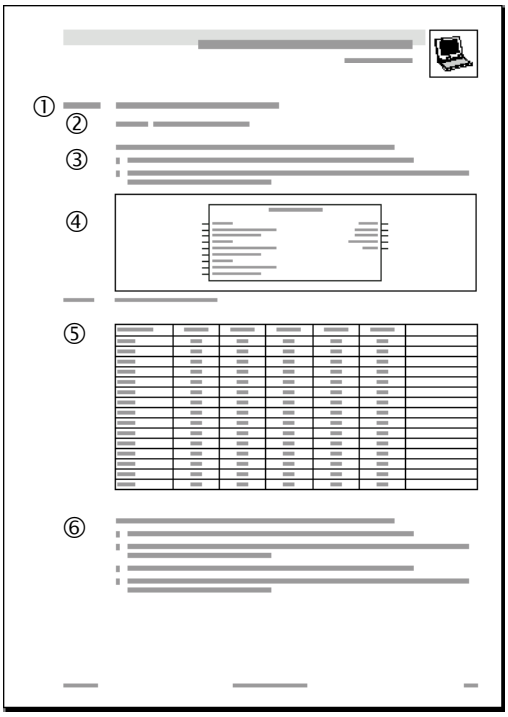
Information type	Distinction (in text)	Example
System block name	**bold**	The **DIGITAL_IO** ...
System block variable identifier	*italics*	The input *DIGIN_bIn1_b* ...

Tip!

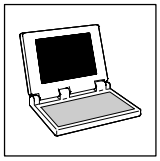
For information about the conventions used for variables of Lenze system organisation units, function blocks and functions can be obtained from the appendix of the DDS online documentation "Introduction into IEC 61131-3 programming". The conventions ensure universal and uniform labelling and support the readability of PLC programs.

1.1.2 System block descriptions

All system block descriptions given in this Manual have the same structure:



①	Headline with SB identifier
②	SB function and module number
③	Short description of the SB and its most important features
④	System block chart including all corresponding variables • Input variables • Output variables
⑤	Table giving information about input and output variables: • Identifier • Data type • Signal type • Address • Display code • Display format • Info
⑥	Detailed SB description

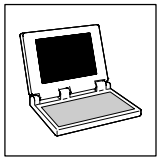


1.1.3 Pictograms in this Manual

	Use of pictograms	Signal words	
Warning of material damage		**Stop!**	Warns of **potential damage to material** . Possible consequences if disregarded: Damage of the Drive PLC or its environment.
More notes		**Tip!** **Note!**	Indicates a tip or note.

1.1.4 Terminology used

Term	In this Manual used for
AIF	Automation interface
DDS	Drive PLC Developer Studio
FIF	Function interface
GDC	Global Drive Control (parameter setting program from Lenze)
SB	System block
System bus	System bus (CAN): Lenze standard bus system similar to *CANopen*



1.2 System block introduction

For a long time, Lenze has followed the principle of describing inverter functions with the aid of function blocks (FB's). This principle can also be found in the IEC 61131-3 standard.

- Functions which can be used as software functions in projects are stored in function libraries as **function blocks** or **functions**. Required SBs must be explicitely linked to the project via the control configuration of DDS.
- In addition, quasi-hardware functions are available, as **system blocks** (SBs).

1.2.1 System block principle

The system-block principle can be explained by means of a PLC system in a rack:

- The rack contains the CPU, digital I/Os, analog I/Os, counter card, positioning card, etc. as additional cards:

CPU	x	x	x	x	x	x

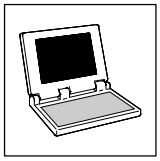
x = Additional cards

- The CPU can directly access the additional cards, and process the resulting information.
- Additional cards have fixed addresses for access.

With Lenze PLCs, system blocks can be compared with these additional cards!

System blocks are special (hardware) function blocks permanently integrated into the run-time system of the PLC.

- SBs can address real hardware.
- SB are assigned/identified through so-called module numbers. (📖 1-4)
- SB inputs and outputs are accessed via system variables or absolute memory addresses. (📖 1-5)
- Inputs/outputs are always classified from the program's point of view. (📖 1-6)
- Required SBs must be explicitely linked to the project via the control configuration of DDS. (📖 1-7)



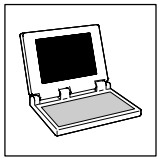
1.2.2 Module numbers

The system blocks of the **Drive PLC** carry the following module numbers:

Module number	System block	Notes
1	DIGITAL_IO	Digital inputs/outputs
2	DIGITAL_IO_EB1	Digital inputs/outputs, internal extension module **Extension board 1**
3	DIGITAL_IO_EB2	Digital inputs/outputs, internal extension module **Extension board 2**
4	DIGITAL_IO_EB3	Digital inputs/outputs, internal extension module **Extension board 3**
11	ANALOG1_IO	Analog inputs/outputs 1
12	ANALOG2_IO	Analog input 2
13	ANALOG3_IO	Analog input 3
14	ANALOG4_IO_EB3	Analog input 4, internal extension module **Extension board 3**
15	ANALOG5_IO_EB3	Analog input 5, internal extension module **Extension board 3**
21	DFIN_IO_DigitalFrequency	Encoder input, internal extension module **Extension board 3**
31	CAN1_IO	System bus (CAN)[1]
32	CAN2_IO	
33	CAN3_IO	
34	FIF_CAN1_IO	System bus (CAN), FIF module **CAN-I/O**[1]
35	FIF_CAN2_IO	
36	FIF_CAN3_IO	
41	AIF1_IO_AutomationInterface	Automation interface
42	AIF2_IO_AutomationInterface	
43	AIF3_IO_AutomationInterface	
101	CAN_Management	System bus (CAN) management[1]
102	CAN_Synchronisation	System bus (CAN) synchronisation[1]
111	FIF_CAN_Management	System bus (CAN) management, FIF module **CAN-I/O** [1]
141	FCODE_FreeCodes	Free codes
150	AD_Channels	AD channels (24 V supply voltage)
151	SYSTEM_FLAGS	System flags
161	AIF_IO_Management	Automation interface management
200	DIGITAL_IO_FIF	Digital inputs/outputs, FIF module **Standard-I/O**
201	ANALOG_IO_FIF	Analog inputs/outputs, FIF module **Standard-I/O**
202	FIF_CAN_DIGITAL_IN	Digital input, FIF module **CAN-I/O**

[1] SBs for system bus (CAN) are described in the “System bus (CAN) for Servo PLC & Drive PLC“ Manual.

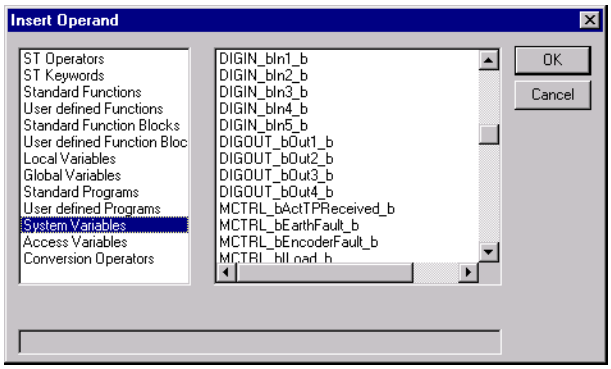
The module number is part of an absolute SB address (see chapter 1.2.4). (📖 1-5)



1.2.3 Access via system variables

System variables can be used for your project after you have integrated a system block into the DDS control configuration.

- Open the DDS editors and **<F2>** to call up the input help which gives you a list of all available system variables.



- This Manual lists the system variables in the corresponding system block table:

Variable name	Data type	Signal type	Address	DIS	DIS format	Note
DIGIN_bIn1_b ... DIGIN_bIn8_b	Bool	binary	%IX1.0.0 ... %IX1.0.7	C0443/1 ... C0443/8	bin	

Example: Table with SB DIGITAL_IO inputs of the Dive PLC

1.2.4 Access via absolute addresses

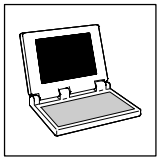
System block inputs and outputs can also be accessed via absolute addresses according to the IEC 61131-3 standard:

For inputs:	For outputs:	
%IXa.b.c	%QXa.b.c	a = Module number b = Word address c = Bit address

- This Manual lists the addresses in the corresponding system block table:

Variable name	Data type	Signal type	Address	DIS	DIS format	Note
DIGIN_bIn1_b ... DIGIN_bIn8_b	Bool	binary	%IX1.0.0 ... %IX1.0.7	C0443/1 ... C0443/8	bin	

Example: Table with SB DIGITAL_IO inputs of the Dive PLC



1.2.5 Definition of inputs/outputs

The user program is connected with the hardware by linking system blocks with program organisation units (POUs):

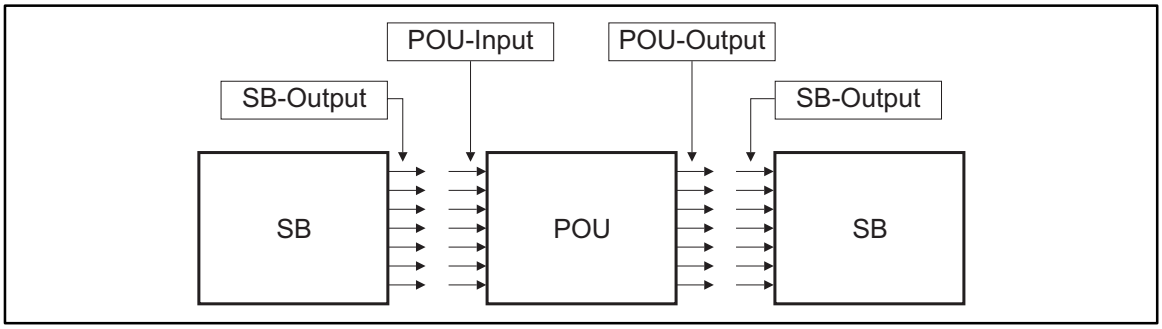


Fig. 1-1 Principle: Link between system blocks and program organisation units (POUs)

Tip!

The input and output assignments are always made from the program viewpoint!

- Logic SB inputs are hardware outputs of the PLC.
- Logic SB outputs are hardware inputs of the PLC.

Example: The system block DIGITAL_IO of the 9300 Servo PLC

If you want to use the digital input 1 and ditigal output 1 of the **9300 Servo PLC** proceed as follows:

1. Link the system block **DIGITAL_IO** explicitely with the DDS control configuration. (📖 1-7)
2. Access to digital input 1:
   Assign a POU input to the system variable *DIGIN_bIn1_b*.
3. Access to digital output 1:
   Assign a POU output to the system variable *DIGOUT_bOut1_b*.

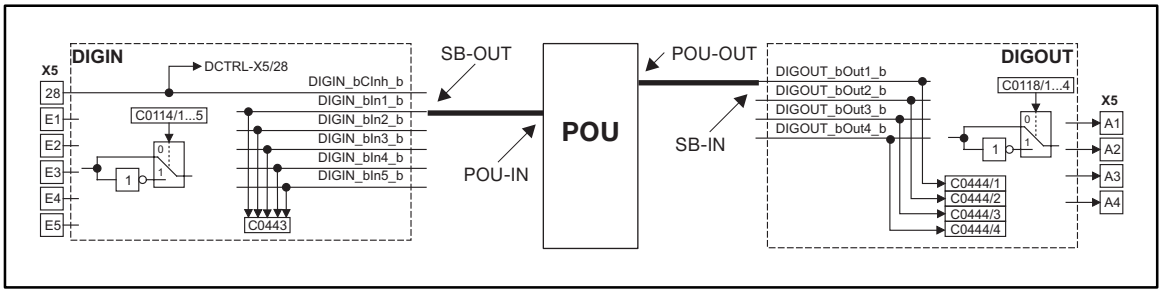


Fig. 1-2 Principle: Connection of the system block DIGITAL_IO of the 9300 Servo PLC with a POU

Tip!

According to the IEC 61131-3 standard, only one copy of the digital input 1 and digital output 1 can be transferred.

1.2.6 Linking of system blocks with DDS

In DDS the required SB must be explicitely linked with the project via the control configuration.

- The control configuration is an object in the register card **Resources** in the *Object Organizer*.
- The control configuration lists all inputs and outputs including the identifiers of the corresponding I/O variable, the absolute address and the data type of the I/O variable for every linked SB.

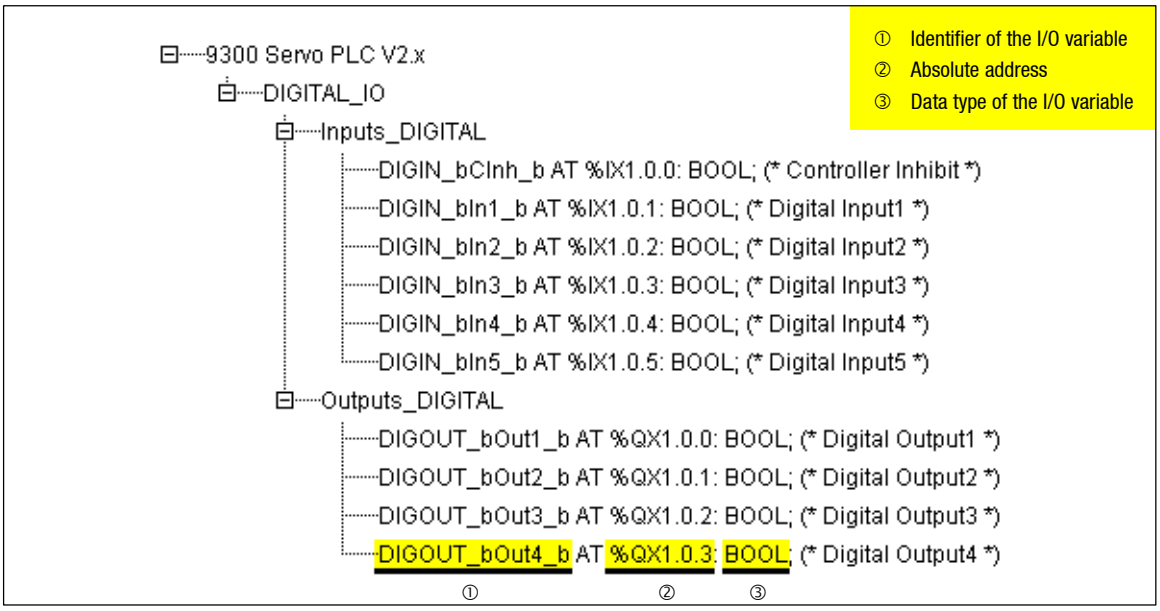
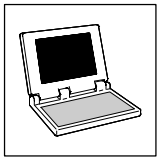


Fig. 1-3 Example: Control configuration for 9300 Servo PLC with linked SB DIGITAL_IO

Tip!

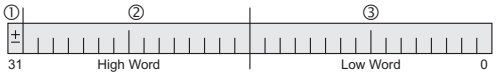
The control configuration can be accessed using the right mouse key. A context menu helps you to add and delete SBs.



1.2.7 Signal types and normalisations

Most inputs and outputs of Lenze function blocks/system blocks can be assigned to a certain signal type. We distinguish between digital, analog, position and speed signals.

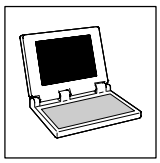
The identifier of the corresponding input/output variable has an ending (starting with an underscore) which indicates the signal type.

Signal type	Ending	Memory location	Normalisation (external ≡ internal)	Previous designation
analog	_a (analog)	16 bit	100 % ≡ 16384	○
digital	_b (binary)	8 bit	0 ≡ FALSE; 1 ≡ TRUE	□
Phase-angle difference or speed	_v (velocity)	16 bit	15000 rpm ≡ 16384	△
	• Phase difference/speed ref. to 1 ms • Normalisation example: $Speed\ (motor) = 15000\ [rpm] = \frac{15000}{60\ [s]}$ $1\ motor\ revolution = 65536\ [inc]$ $Variable\ value\ (..._v) = \frac{15000}{60\ [s]} \cdot 65536\ [inc] = \frac{15000}{60000\ [ms]} \cdot 65536\ [inc] = 16384 \left[\frac{inc}{ms}\right]$			
Phase-angle or position	_p (position)	32 bit	1 motor revolution ≡ 65536	▲
	 ① Direction (0 ≡ CW; 1 ≡ CCW) ② No. of motor revolutions (0 ... 32767) ③ Phase or angle (0 ... 65535)			

Note!

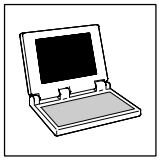
Because of their normalisation, analog signals use an unsymmetrical resolution area. (-200 % ... +199.99 %):

External:	-200 %	-100 %	0	+100 %	+199.99 %
Internal:	-32768	-16384	0	+16384	+32767



2 System blocks

The following sections inform about the system blocks of the basic unit.



2.1 AD_CHANNELS

2.1.1 Inputs_AD_CHANNELS

A/D conversion of 24 V voltage supply (module number 150)

This SB indicates the voltage of terminals X1/+24, ⊥24 as integer value with a resolution of 100 mV.

- If 24 V DC is applied to the terminals, the *V24PowerSupply* has the value “240” (100 mV · 240 = 24 V).

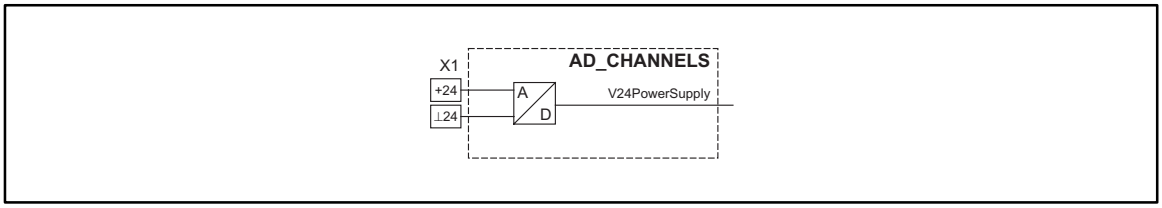


Fig. 2-1 AD_CHANNELS

VariableName	DataType	SignalType	Address	DIS	DIS format	Note
V24PowerSupply	Integer	-	%IW150.0	-	-	1 ≡ 100 mV

Electrical data of the input terminal

Terminal	Use	Data
+24	Supply voltage	Level: + 18 V_{DC} ... +30 V_{DC}
⊥24	Supply voltage reference potential	

2.2 AIF1_IO_AutomationInterface

2.2.1 Inputs_AIF1

Automation interface (module number 41)

This SB is used as interface for input signals (e. g. setpoints/actual values) from attached fieldbus modules (e. g. INTERBUS, PROFIBUS-DP).

- The process image is
  – created every 10 ms.
  – created in an interval task within the time set for this task.
  – read at the beginning of the task and written at its end.

Tip!

Please observe the corresponding Operating Instructions for the fieldbus module.

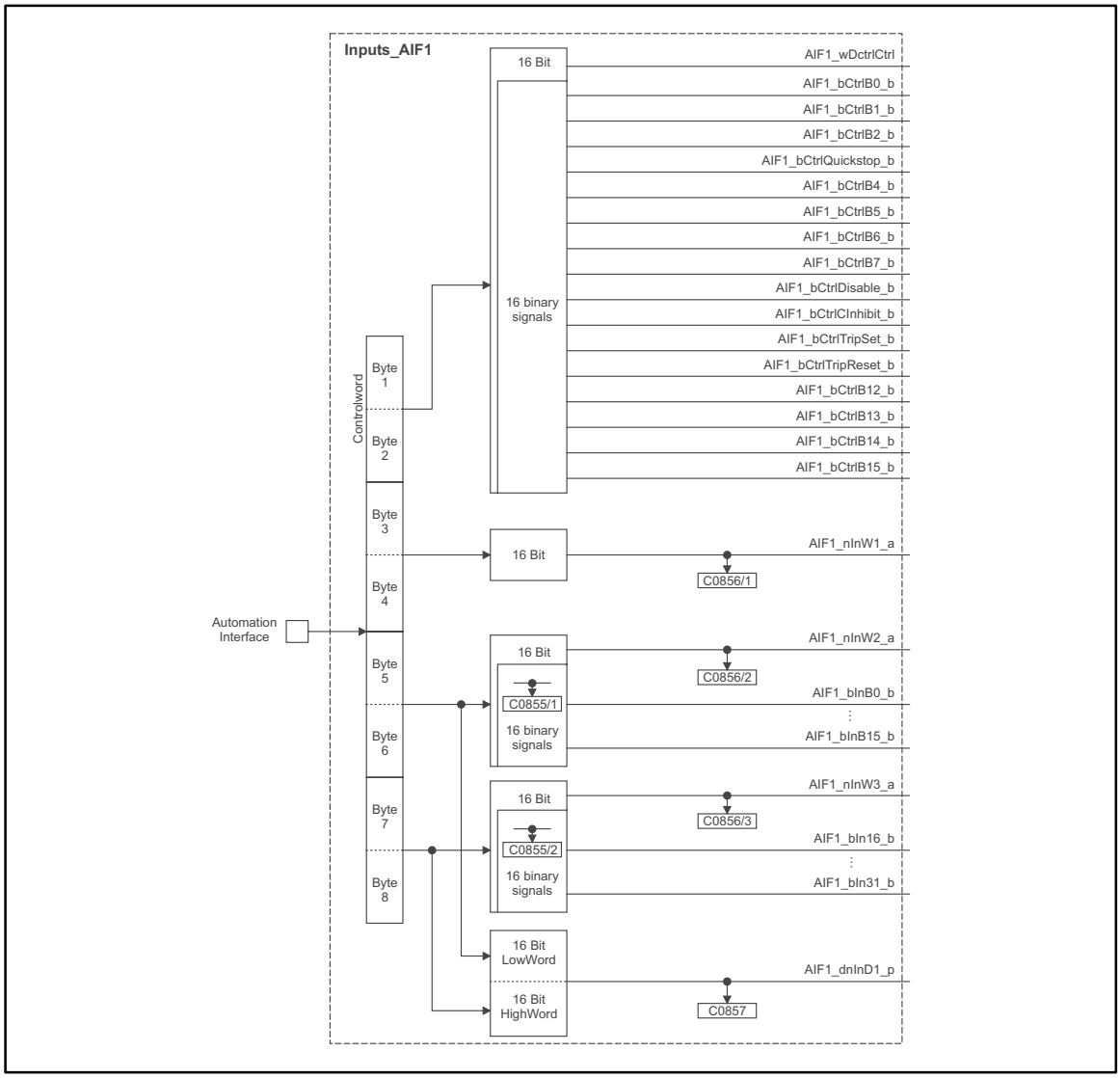
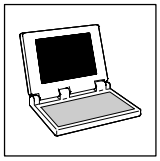


Fig. 2-2 Inputs_AIF1

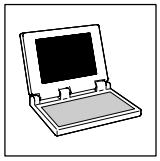


Drive PLC

System blocks

AIF1_IO_AutomationInterface

Variable name	Data type	Signal type	Address	DIS	DIS format	Note
AIF1_wDctrlCtrl	Word	-	%IX41.0	C0136/3	hex	
AIF1_bCtrlB0_b	Bool	binary	%IX41.0.0	C0136/3	bin	
AIF1_bCtrlB1_b			%IX41.0.1			
AIF1_bCtrlB2_b			%IX41.0.2			
AIF1_bCtrlQuickstop_b			%IX41.0.3			
AIF1_bCtrlB4_b			%IX41.0.4			
AIF1_bCtrlB5_b			%IX41.0.5			
AIF1_bCtrlB6_b			%IX41.0.6			
AIF1_bCtrlB7_b			%IX41.0.7			
AIF1_bCtrlDisable_b			%IX41.0.8			
AIF1_bCtrlClnhibit_b			%IX41.0.9			
AIF1_bCtrlTripSet_b			%IX41.0.10			
AIF1_bCtrlTripReset_b			%IX41.0.11			
AIF1_bCtrlB12_b			%IX41.0.12			
AIF1_bCtrlB13_b			%IX41.0.13			
AIF1_bCtrlB14_b			%IX41.0.14			
AIF1_bCtrlB15_b			%IX41.0.15			
AIF1_nInW1_a	Integer	analog	%IW41.1	C0856/1	dec [%]	
AIF1_nInW2_a			%IW41.2	C0856/2		
AIF1_nInW3_a			%IW41.3	C0856/3		
AIF1_bInB0_b ... AIF1_bInB15_b	Bool	binary	%IX41.2.0 ... %IX41.2.15	C0855/1	hex	
AIF1_bInB16_b ... AIF1_bInB31_b			%IX41.3.0 ... %IX41.3.15	C0855/2		
AIF1_dnInD1_p	Double integer	position	%ID41.1	C0857	dec [inc]	



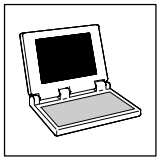
User data

The received 8 bytes user data are assigned to several variables of different data types and can be evaluated as

- binary information (1 bit)
- control word/quasi analog value (16 bit)
- angle information (32 bit)

if needed.

<table>
<tr><th>Byte</th><th colspan="2">Variable (1 bit)</th><th>Variable (16 bit)</th><th>Variable (32 bit)</th></tr>
<tr><td>1, 2</td><td colspan="2">AIF1_bInB0_b
AIF1_bInB1_b
AIF1_bInB2_b
AIF1_bCtrlQuickstop_b
AIF1_bInB4_b
...
AIF1_bInB7_b
AIF1_bCtrlDisable_b
AIF1_bCtrlCInhibit_b
AIF1_bCtrlTripSet_b
AIF1_bCtrlTripReset_b
AIF1_bInB12_b
...
AIF1_bInB15_b</td><td>AIF1_wDctrlCtrl</td><td></td></tr>
<tr><td></td><td>Notes:</td><td>Drive PLC:
All variables assigned to byte 1/2 can be freely used in the PLC program.</td><td colspan="2"></td></tr>
<tr><td></td><td></td><td>9300 Servo PLC:
The controller-internal control word is not assigned to byte 1/2 as fixed word.
• The signals for Quick stop (QSP), DISABLE, CINH, TRIP-SET and TRIP-RESET can be written to the SB DCTRL_DriveControl .
– To do this, connect the variable AIF1_wDctrlCtrl with the variable DCTRL_wAIF1Ctrl of the SB DCTRL_DriveControl.
• The signals can also be read and processed via the following variables:
– AIF1_bCtrlQuickstop_b
– AIF1_bCtrlDisable_b
– AIF1_bCtrlCInhibit_b
– AIF1_bCtrlTripSet_b
– AIF1_bCtrlTripReset_b</td><td colspan="2"></td></tr>
<tr><td>3, 4</td><td colspan="2"></td><td>AIF1_nInW1_a</td><td></td></tr>
<tr><td>5, 6</td><td colspan="2">AIF1_bInB0_b
...
AIF1_bInB15_b</td><td>AIF1_nInW2_a</td><td rowspan="2">AIF1_dnInD1_p</td></tr>
<tr><td>7, 8</td><td colspan="2">AIF1_bInB16_b
...
AIF1_bInB31_b</td><td>AIF1_nInW3_a</td></tr>
</table>



2.2.2 Outputs_AIF1

Automation interface (module number 41)

This SB is used as interface for output signals (e.g. setpoint/actual value) from attached fieldbus modules (e. g. INTERBUS, PROFIBUS-DP).

- The process image is
  – created every 10 ms.
  – created in an interval task within the time set for this task.
  – read at the beginning of the task and written at its end.

Tip!

Please observe the Operating Instructions for the attached fieldbus module.

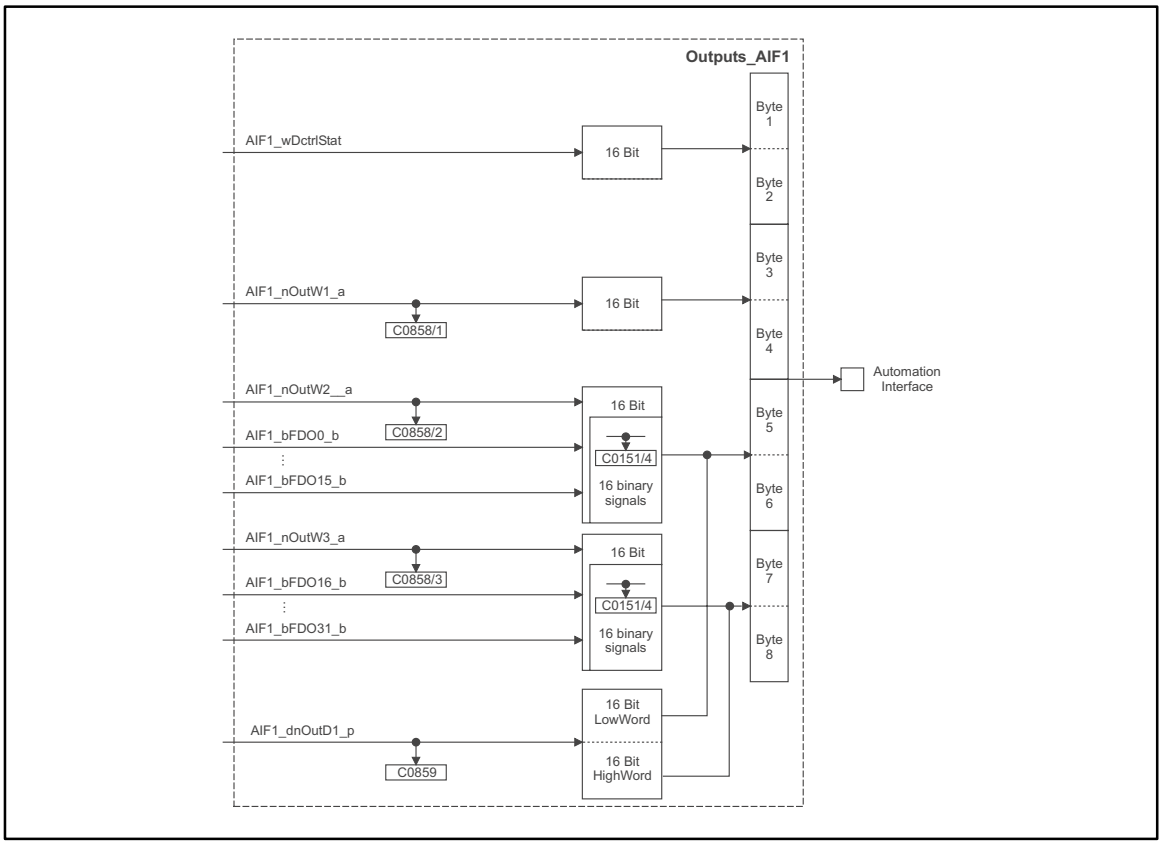
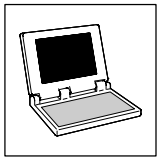


Fig. 2-3 Outputs_AIF1



Variable name	Data type	Signal type	Address	DIS	DIS format	Note
AIF1_wDctrlStat	Word	-	%QW41.0	-	-	
AIF1_nOutW1_a	Integer	analog	%QW41.1	C0858/1	dec [%]	
AIF1_nOutW2_a			%QW41.2	C0858/2		
AIF1_nOutW3_a			%QW41.3	C0858/3		
AIF1_bFDO0_b .. AIF1_bFDO15_b	Bool	binary	%QX41.2.0 ... %QX41.2.15	C0151/4	hex	Display code in hex as double word
AIF1_bFDO16_b .. AIF1_bFDO31_b			%QX41.3.0 ... %QX41.3.15			
AIF1_dnOutD1_p	Double-integer	position	%QD41.1	C0859	dec [inc]	

User data

The 8 bytes user data to be sent can be written via several variables of different data types at the same time. Data can therefore be transferred as

- binary information (1 bit)
- status word/quasi analog value (16 bit)
- angle information (32 bit)

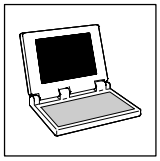
by the PLC program.

Byte	Variable (1 bit)	Variable (16 bit)	Variable (32 bit)
1, 2		AIF1_wDctrlStat	
	Notes: **Drive PLC:** All variables assigned to byte 1/2 can be overwritten by the PLC program.		
	9300 Servo PLC: Bytes 1 and 2 can be used to transfer the status wrd from the SB **DCTRL_DriveControl** • To do this, connect the variable *DCTRL_wStat* of the SB **DCTRL_DriveControl** with the variable *AIF1_wDctrlStat*. • In addition to signals such as IMP and CINH, the SB status word **DCTRL_DriveControl** contains some freely assignable signals which can be overwritten via the variables *DCTRL_bStateB.._b* of the SB **DCTRL_DriveControl** .		
3, 4		AIF1_nOutW1_a	
5, 6	AIF1_bFDO0_b ... AIF1_bFDO15_b	AIF1_nOutW2_a	AIF1_dnOutD1_p
7, 8	AIF1_bFDO16_b ... AIF1_bFDO31_b	AIF1_nOutW3_a	

Tip!

Avoid simultaneous overwriting via different variable types to ensure data consistency.

- Thus bytes 5 and 6 should only be overwritten
  – by the variable *AIF1_dnOutD1_p* ,
  – by the variable *AIF1_nOutW2_a* or
  – by the variables *AIF1_bFDO0_b* ... *AIF1_bFDO15_b* .



2.3 AIF2_IO_AutomationInterface

2.3.1 Inputs_AIF2

Automation interface (module number 42)

This SB is used as interface for input signals (e. g. setpoints/actual values) from attached fieldbus modules (e. g. INTERBUS, PROFIBUS-DP).

- The process image is
  – created every 10 ms.
  – created in an interval task within the time set for this task.
  – read at the beginning of the task and written at its end.

Tip!

Please observe the corresponding Operating Instructions for the fieldbus module that is plugged in.

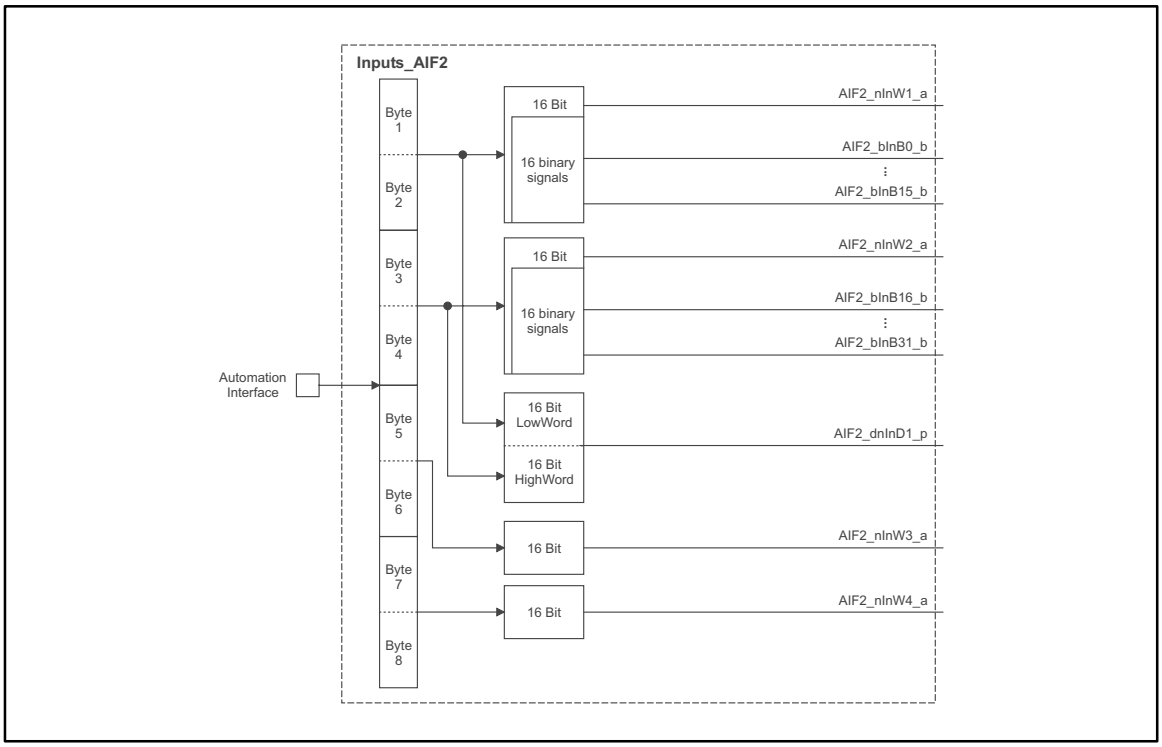
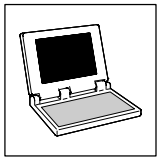


Fig. 2-4 Inputs_AIF2



Variable name	Data type	Signal type	Address	DIS	DIS format	Note
AIF2_nInW1_a	Integer	analog	%IW42.0			
AIF2_nInW2_a			%IW42.1			
AIF2_nInW3_a			%IW42.2			
AIF2_nInW4_a			%IW42.3			
AIF2_bInB0_b ... AIF2_bInB15_b	Bool	binary	%IX42.0.0 ... %IX42.0.15			
AIF2_bInB16_b ... AIF2_bInB31_b			%IX42.1.0 ... %IX42.1.15			
AIF2_dnInD1_p	Double-integer	position	%ID42.0			

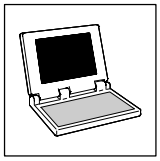
User data

The 4 first bytes of the received 8 bytes user data are assigned to several variables of different data types and can be evaluated as

- binary information (1 bit)
- quasi analog value (16 bit)
- angle information (32 bit)

if needed.

Byte	Variable (1 bit)	Variable (16 bit)	Variable (32 bit)
1, 2	AIF2_bInB0_b ... AIF2_bInB15_b	AIF2_nInW1_a	AIF2_dnInD1_p
3, 4	AIF2_bInB16_b ... AIF2_bInB31_b	AIF2_nInW2_a	
5, 6		AIF2_nInW3_a	
7, 8		AIF2_nInW4_a	



2.3.2 Outputs_AIF2

Automation interface (module number 42)

This SB is used as interface for output signals (e.g. setpoint/actual value) from attached fieldbus modules (e. g. INTERBUS, PROFIBUS-DP).

- The process image is
  – created every 10 ms.
  – created in an interval task within the time set for this task.
  – read at the beginning of the task and written at its end.

Tip!

Please observe the corresponding Operating Instructions for the fieldbus module that is plugged in.

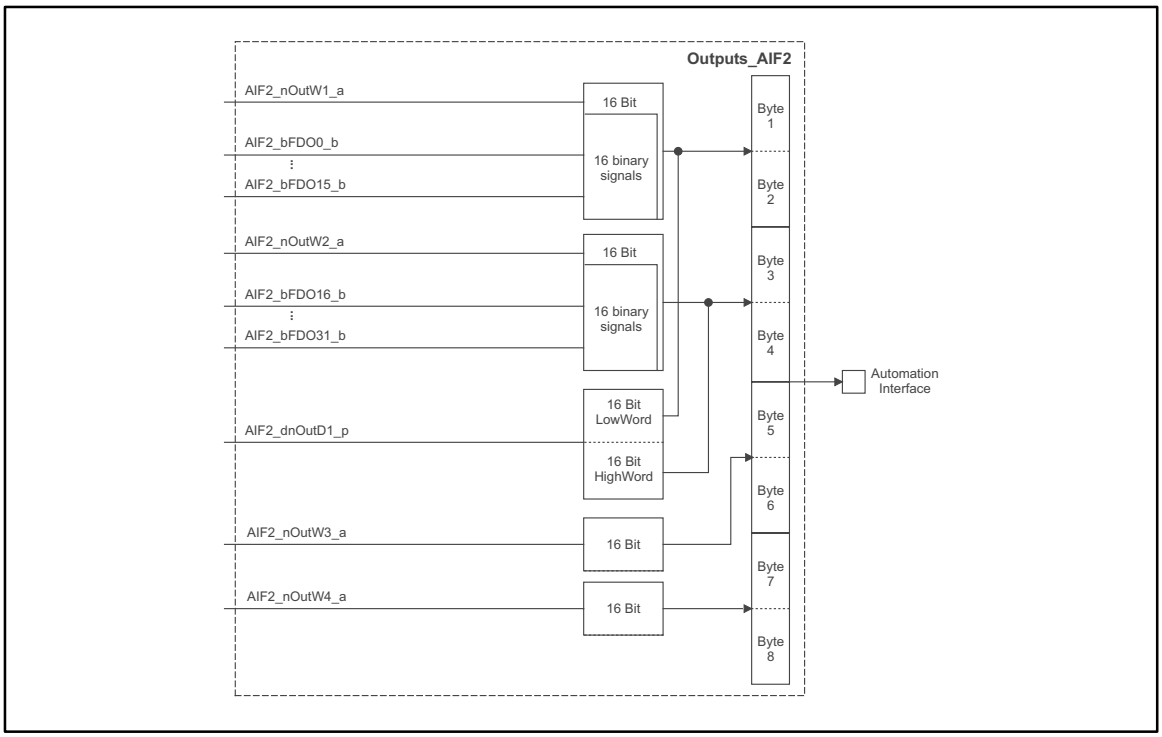
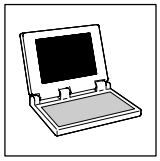


Fig. 2-5 Outputs_AIF2

Variable name	Data type	Signal type	Address	DIS	DIS format	Note
AIF2_nOutW1_a	Integer	analog	%QW42.0			
AIF2_nOutW2_a			%QW42.1			
AIF2_nOutW3_a			%QW42.2			
AIF2_nOutW4_a			%QW42.3			
AIF2_bFDO0_b ... AIF2_bFDO15_b	Bool	binary	%QX42.0.0 ... %QX42.0.15			
AIF2_bFDO16_b ... AIF2_bFDO31_b			%QX42.1.0 ... %QX42.1.15			
AIF2_dnOutD1_p	Double-integer	position	%QD42.0			



User data

The first 4 bytes of the 8 bytes user data to be sent can be written via several variables of different data types at the same time. Data can therefore be transferred as:

- binary information (1 bit)
- quasi analog value (16 bit)
- angle information (32 bit)

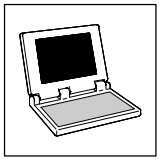
by the PLC program.

Byte	Variable (1 bit)	Variable (16 bit)	Variable (32 bit)
1, 2	AIF2_bFDO0_b ... AIF2_bFDO15_b	AIF2_nOutW1_a	AIF2_dnOutD1_p
3, 4	AIF2_bFDO16_b ... AIF2_bFDO31_b	AIF2_nOutW2_a	
5, 6		AIF2_nOutW3_a	
7, 8		AIF2_nOutW4_a	

Tip!

Avoid simultaneous overwriting via different variable types to ensure data consistency.

- Thus bytes 1 and 2 should only be overwritten
  – by the variable *AIF2_dnOutD1_p*,
  – by the variable *AIF2_nOutW1_a* or
  – by the variables *AIF2_bFDO0_b* ... *AIF2_bFDO15_b*.



2.4 AIF3_IO_AutomationInterface

2.4.1 Inputs_AIF3

Automation interface (module number 43)

This SB is used as interface for input signals (e. g. setpoints/actual values) from attached fieldbus modules (e. g. INTERBUS, PROFIBUS-DP).

- The process image is
  – created every 10 ms.
  – created in an interval task within the time set for this task.
  – read at the beginning of the task and written at its end.

Tip!

Please observe the corresponding Operating Instructions for the fieldbus module that is plugged in.

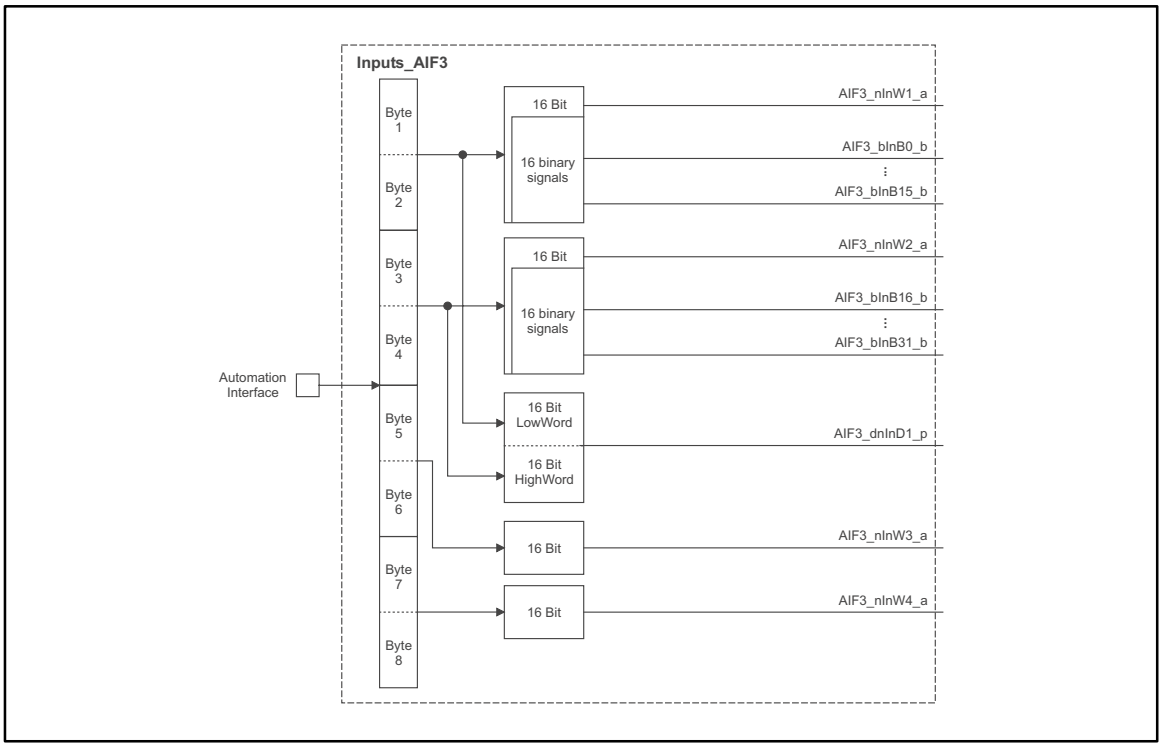
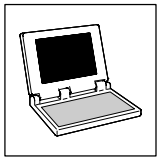


Fig. 2-6 Inputs_AIF3



Variable name	Data type	Signal type	Address	DIS	DIS format	Note
AIF3_nInW1_a	Integer	analog	%IW43.0			
AIF3_nInW2_a			%IW43.1			
AIF3_nInW3_a			%IW43.2			
AIF3_nInW4_a			%IW43.3			
AIF3_bInB0_b ... AIF3_bInB15_b	Bool	binary	%IX43.0.0 ... %IX43.0.15			
AIF3_bInB16_b ... AIF3_bInB31_b			%IX43.1.0 ... %IX43.1.15			
AIF3_dnInD1_p	Double-integer	position	%ID43.0			

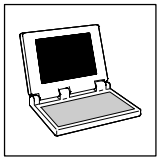
User data

The 4 first bytes of the received 8 bytes user data are assigned to several variables of different data types and can be evaluated as

- binary information (1 bit)
- quasi analog value (16 bit)
- angle information (32 bit)

if needed.

Byte	Variable (1 bit)	Variable (16 bit)	Variable (32 bit)
1, 2	AIF3_bInB0_b ... AIF3_bInB15_b	AIF3_nInW1_a	AIF3_dnInD1_p
3, 4	AIF3_bInB16_b ... AIF3_bInB31_b	AIF3_nInW2_a	
5, 6		AIF3_nInW3_a	
7, 8		AIF3_nInW4_a	



2.4.2 Outputs_AIF3

Automation interface (module number 43)

This SB is used as interface for output signals (e.g. setpoint/actual value) from attached fieldbus modules (e. g. INTERBUS, PROFIBUS-DP).

- The process image is
  – created every 10 ms.
  – created in an interval task within the time set for this task.
  – read at the beginning of the task and written at its end.

Tip!

Please observe the corresponding Operating Instructions for the fieldbus module that is plugged in.

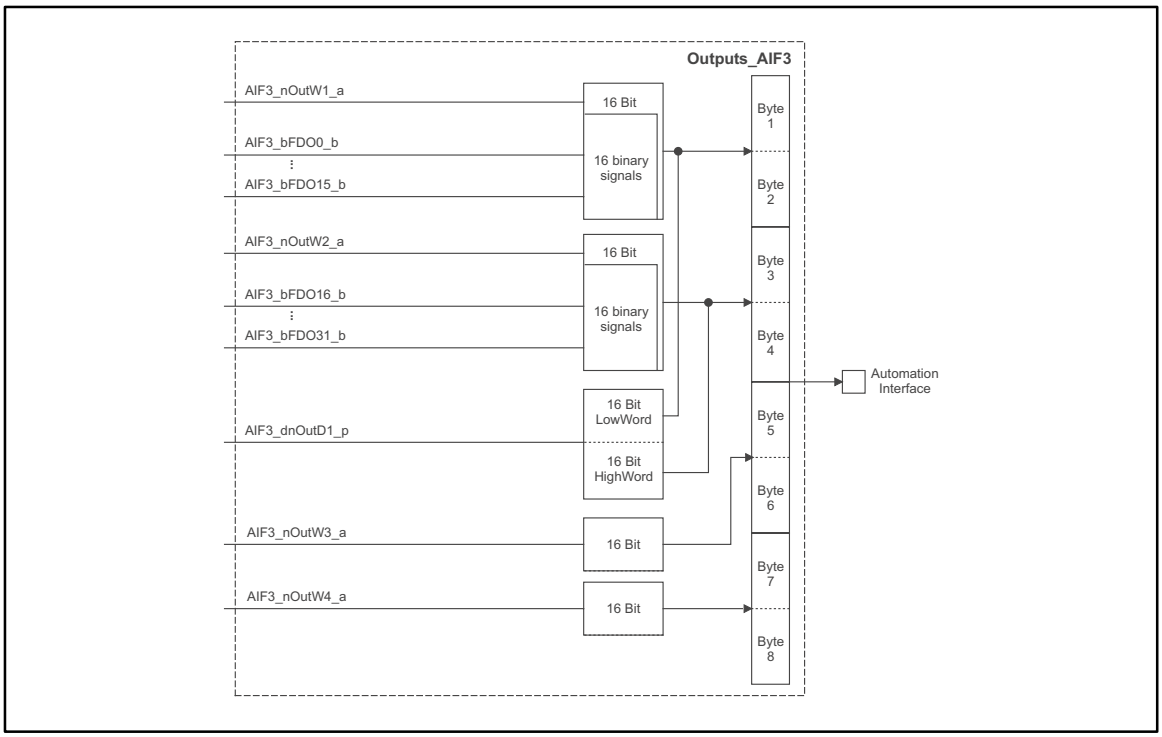
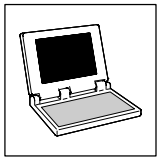


Fig. 2-7 Outputs_AIF3

Variable name	Data type	Signal type	Address	DIS	DIS format	Note
AIF3_nOutW1_a	Integer	analog	%QW43.0			
AIF3_nOutW2_a			%QW43.1			
AIF3_nOutW3_a			%QW43.2			
AIF3_nOutW4_a			%QW43.3			
AIF3_bFDO0_b ... AIF3_bFDO15_b	Bool	binary	%QX43.0.0 ... %QX43.0.15			
AIF3_bFDO16_b ... AIF3_bFDO31_b			%QX43.1.0 ... %QX43.1.15			
AIF3_dnOutD1_p	Double-integer	position	%QD43.0			



User data

The first 4 bytes of the 8 bytes user data to be sent can be written via several variables of different data types at the same time. Data can therefore be transferred as:

- binary information (1 bit)
- quasi analog value (16 bit)
- angle information (32 bit)

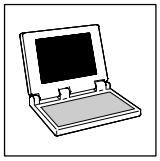
by the PLC program.

Byte	Variable (1 bit)	Variable (16 bit)	Variable (32 bit)
1, 2	AIF3_bFDO0_b ... AIF3_bFDO15_b	AIF3_nOutW1_a	AIF3_dnOutD1_p
3, 4	AIF3_bFDO16_b ... AIF3_bFDO31_b	AIF3_nOutW2_a	
5, 6		AIF3_nOutW3_a	
7, 8		AIF3_nOutW4_a	

Tip!

Avoid simultaneous overwriting via different variable types to ensure data consistency.

- Thus bytes 1 and 2 should only be overwritten
  – by the variable *AIF3_dnOutD1_p*,
  – by the variable *AIF3_nOutW1_a* or
  – by the variables *AIF3_bFDO0_b* ... *AIF3_bFDO15_b*.



2.5 AIF_IO_Management

Automation interface management (module number 161)

This SB detects communication errors by means of a fieldbus module connected to the automation interface (AIF).

- In the event of an error *AIF_bCe0CommErr_b* is set to **TRUE** and the communciation error CE0 (LECOM no. 61) is set. The response to this can be configured under C0126 (default setting: off).
- New AIF fieldbus modules (e. g. 2133 and 2175) also use *AIF_bFieldBusStateBit0_b* ... *AIF_bFieldBusStateBit7_b* to transfer an error number from the fieldbus.

Tip!

Please observe the corresponding Operating Instructions for the field bus module that is plugged in.

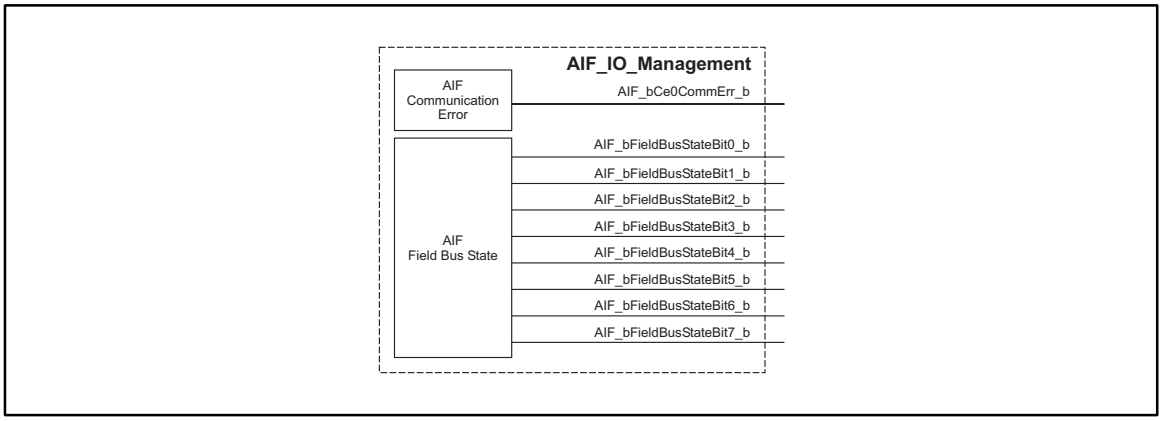


Fig. 2-8 AIF_IO_Management

Variable name	Data type	Signal type	Address	DIS	DIS format	Note
AIF_bCe0CommErr_b	Bool	binary	%IX161.0.0			Communication error “CE0”
AIF_bFieldBusStateBit0_b			%IX161.1.0			Error number - Bit 0
AIF_bFieldBusStateBit1_b			%IX161.1.1			Error number - Bit 1
AIF_bFieldBusStateBit2_b			%IX161.1.2			Error number - Bit 2
AIF_bFieldBusStateBit3_b			%IX161.1.3			Error number - Bit 3
AIF_bFieldBusStateBit4_b			%IX161.1.4			Error number - Bit 4
AIF_bFieldBusStateBit5_b			%IX161.1.5			Error number - Bit 5
AIF_bFieldBusStateBit6_b			%IX161.1.6			Error number - Bit 6
AIF_bFieldBusStateBit7_b			%IX161.1.7			Error number - Bit 7

Communication error CE0

- The response to the communication error “CE0” can be configured under C0126:

Code	LCD	Possible settings		Info
		Lenze	Selection	
C0126	MONIT CE0	3		Configuration Communciation error (CE0)
			0 TRIP 2 Warning 3 Off	

2.6 ANALOG1_IO

2.6.1 Inputs_ANALOG1

Analog input 1 (module number 11)

This SB forms the interface for analog signals via terminal AI1, A⊥ as setpoint input or actual value input.

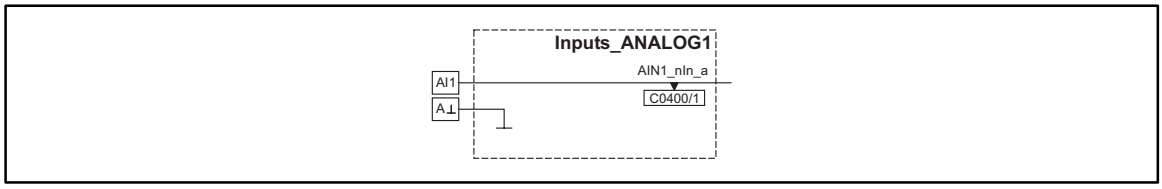
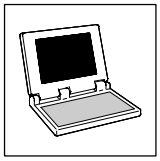


Fig. 2-9 Inputs_ANALOG1

Variable name	Data type	Signal type	Address	DIS	DIS format	Note
AIN1_nIn_a	Integer	analog	%IW11.0	C0400/1	dec [%]	Analog input 1

Terminal assignment

Terminal	Use	Data
AI1	Analog input 1	Level: -10 V ... +10 V Resolution: 10 bits + sign Internal resistance: 100 k Ω Normalisation: ±10 V ≡ ±16384 ≡ ±100 %
A⊥	Internal ground, GND	-



2.6.2 Outputs_ANALOG1

Analog output 1 (module number 11)

You can use this SB as a monitor output. You can output internal analog signals via

- terminal AOV as voltage signals and
- terminal AOI as current signals

and use them, e.g. as display or setpoint values for following drives.

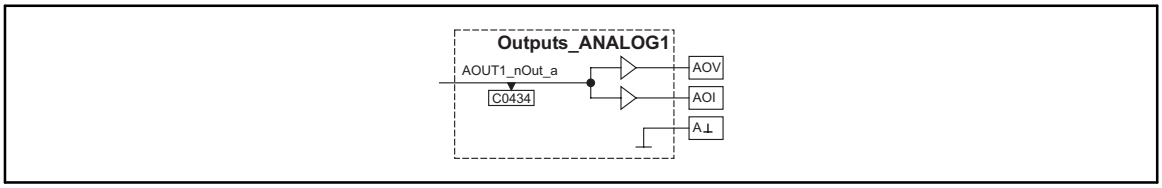


Fig. 2-10 Outputs_ANALOG1

Variable name	Data type	Signal type	Address	DIS	DIS format	Note
AOUT1_nOut_a	Integer	analog	%QW11.0	C0434	dec [%]	Analog output 1

Function

- A voltage of 10 V is output at terminal A0V, if the signal on *AOUT1_nOut_a* = 16384 = 100 %
- A current of 20 mA is output at terminal AOI, if the signal on *AOUT1_nOut_a* = 16384 = 100 %

Terminal assignment

Terminal	Use	Data
AOV	Analog output 1 (voltage signal)	Level: -10 V ... +10 V (max. 2 mA) Resolution: 10 bit + sign Normalisation: ±10 V ≡ ±16384 ≡ ±100 %
AOI	Analog output 1 (current signal)	Level: -20 mA ... +20 mA Resolution: 10 bit + sign Normalisation: ±20 mA ≡ ±16384 ≡ ±100 %
A⊥	Internal ground, GND	-

2.7 ANALOG2_IO

2.7.1 Inputs_ANALOG2

Analog input 2 (module number 12)

This SB forms the interface for analog signals via terminal AI2, A⊥.

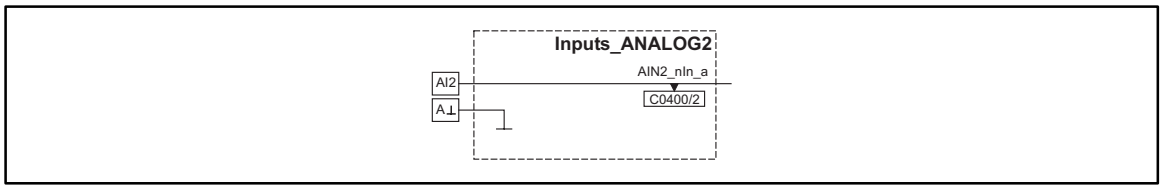
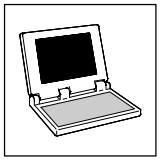


Fig. 2-11 Inputs_ANALOG2

Variable name	Data type	Signal type	Address	DIS	DIS format	Note
AIN2_nIn_a	Integer	analog	%IW12.0	C0400/2	dec [%]	Analog input 2

Terminal assignment

Terminal	Use	Data
AI2	Analog input 2	Level: -10 V ... +10 V Resolution: 10 bit + sign Internal resistance: 100 k Ω Normalisation: ±10 V ≡ ±16384 ≡ ±100 %
A⊥	Internal ground, GND	-



2.8 ANALOG3_IO

2.8.1 Inputs_ANALOG3

Analog input 3 (module number 13)

This SB forms the interface for analog signals via terminal AI3, A⊥.

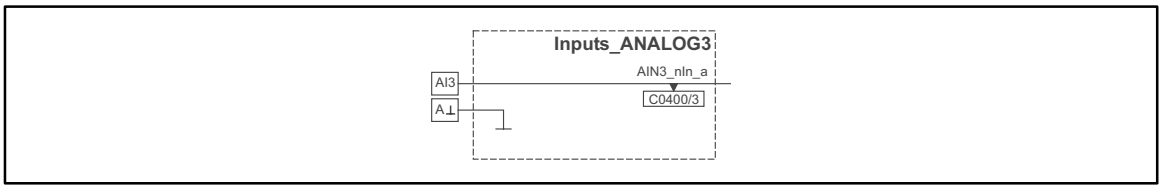


Fig. 2-12 Inputs_ANALOG3

Variable name	Data type	Signal type	Address	DIS	DIS format	Note
AIN3_nIn_a	Integer	analog	%IW13.0	C0400/3	dec [%]	Analog input 3

Terminal assignment

Terminal	Use	Data
AI3	Analog input 3	Level: -10 V ... +10 V Resolution: 10 bits + sign Internal resistance: 100 k Ω Normalisation: ±10 V ≡ ±16384 ≡ ±100 %
A⊥	Internal ground, GND	-

2.9 DIGITAL_IO

2.9.1 Inputs_DIGITAL

Digital inputs (module number 1)

This SB reads the signals at terminals I1 ... I8 and conditions them.

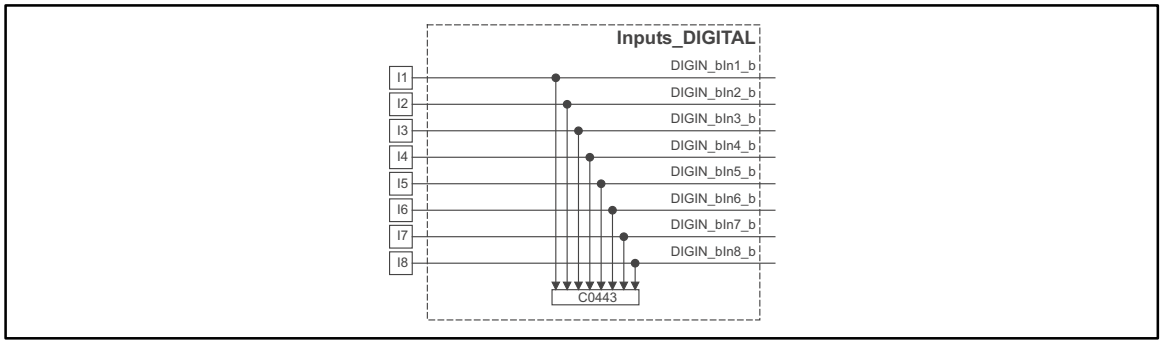


Fig. 2-13 Inputs_DIGITAL

Variable name	Data type	Signal type	Address	DIS	DIS format	Note
DIGIN_bIn1_b	Bool	binary	%IX1.0.0	C0443 - Bit1	bin	Digital inputs I1 ... I8
DIGIN_bIn2_b			%IX1.0.1	C0443 - Bit2		
DIGIN_bIn3_b			%IX1.0.2	C0443 - Bit3		
DIGIN_bIn4_b			%IX1.0.3	C0443 - Bit4		
DIGIN_bIn5_b			%IX1.0.4	C0443 - Bit5		
DIGIN_bIn6_b			%IX1.0.5	C0443 - Bit6		
DIGIN_bIn7_b			%IX1.0.6	C0443 - Bit7		
DIGIN_bIn8_b			%IX1.0.7	C0443 - Bit8		

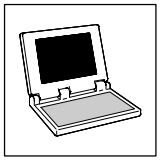
Electrical data of the input terminals

Terminal	Use	Data
I1	freely assignable	LOW level: 0 ... +4 V HIGH level: +13 ... +30 V Input current: 8 mA per input (at 24 V)
I2		
I3		
I4		
I5		
I6		
I7		
I8		

Tip!

You can use I2 ... I4 as real interrupt inputs. The references to the hardware interrupt inputs are in the task configuration. Response time of the interrupt task: < 250 µsec.

More detailed information about Drive PLC terminal strips/control connections can be found in the corresponding Mounting Instructions!



2.9.2 Outputs_DIGITAL

Digital outputs (module number 1)

This SB conditions digital signals, and outputs them at terminals O1 ... O4.

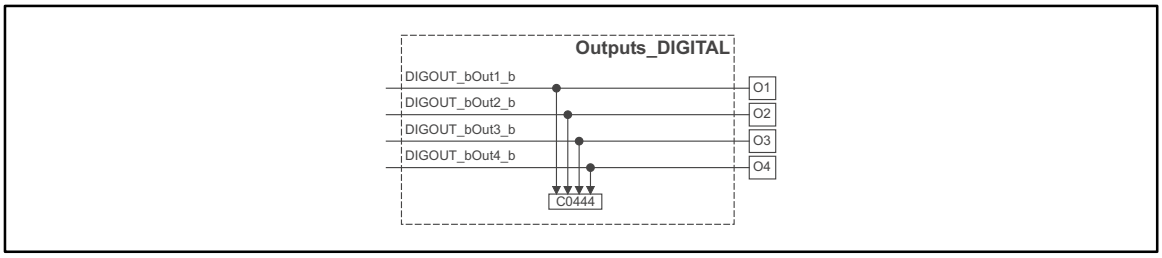


Fig. 2-14 Outputs_DIGITAL

VariableName	DataType	SignalType	Address	DIS	DIS format	Note
DIGOUT_bOut1_b	Bool	binary	%QX1.0.0	C0444 - Bit1	bin	Digital outputs 01 ... 04
DIGOUT_bOut2_b			%QX1.0.1	C0444 - Bit2		
DIGOUT_bOut3_b			%QX1.0.2	C0444 - Bit3		
DIGOUT_bOut4_b			%QX1.0.3	C0444 - Bit4		

Electrical data of the output terminals

Terminal	Use	Data
01	freely assignable	LOW level: 0 ... +4 V HIGH level: +13 ... +30 V Output current: max. 1 A per output
02		
03		
04		

Tip!

If the user program have not been started, all outputs are defined as “LOW” after switch-on.

More detailed information about Drive PLC terminal strips/control connections can be found in the corresponding Mounting Instructions!

2.10 FCODE_FreeCodes

(Module number 141)

At Lenze, controller parameters are called codes. The PLC can be adapted to your application without additional programming by changing codes.

This SB provides several variables which directly read “free“ PLC codes and processes them further in the PLC program or the codes can be written by the PLC program.

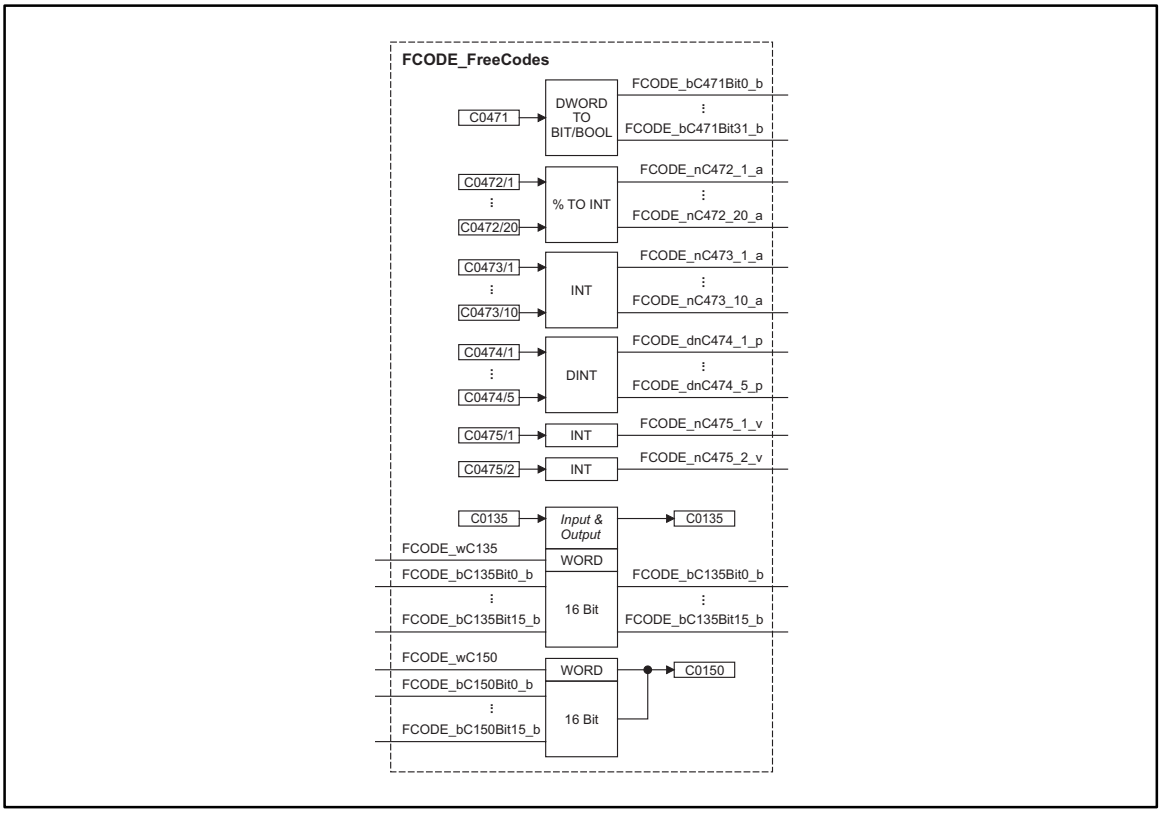
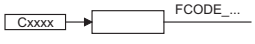
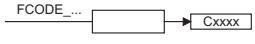
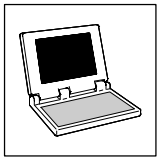


Fig. 2-15 FCODE_FreeCodes

Code ⇨ PLC program	**PLC program ⇨ Code**
	
The PLC codes listed in Fig. 2-15 in boxes on the left-hand side are directly assigned to the variables listed on the right-hand side. • These variables can only be read by the PLC program. • See chapter 2.10.1, “Inputs_FCODE”	The variables listed in Fig. 2-15 on the left-hand side are assigned to the codes listed in boxes on the right-hand side. • These codes can only be read from “outside”. • See chapter 2.10.2, “Outputs_FCODE”
• Codes are converted into variables (and vice versa) according to a fixed routine. • In the code table, you can find the options that can be set, and the Lenze settings. 🕮 8-15	

Tip!

Code C0135 can be read and written via the variables *FCODE_bC135Bit0_b* ... *FCODE_bC135Bit15_b* (input & output).



2.10.1 Inputs_FCODE

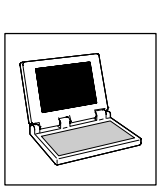
VariableName	DataType	SignalType	Address	DIS	DIS-Format	Note
FCODE_bC135Bit0_b ... FCODE_bC135Bit15_b	Bool	binary		-	-	default = 0 Also output variable (see table below)
FCODE_bC471Bit0_b ... FCODE_bC471Bit15_b			%IX141.0.0 ... %IX141.0.15			default = 0
FCODE_bC471Bit16_b ... FCODE_bC471Bit31_b			%IX141.1.0 ... %IX141.1.15			
FCODE_nC472_1_a ... FCODE_nC472_20_a	Integer	analog	%IW141.2 ... %IW141.21	-	-	default = 0.00 %
FCODE_nC473_1_a ... FCODE_nC473_10_a			%IW141.22 ... %IW141.31			default = 0 C0473/1,2 = 1
FCODE_dnC474_1_p ... FCODE_dnC474_5_p	Double Integer	position	%ID141.16 ... %ID141.20	-	-	default = 0
FCODE_nC475_1_v	Integer	velocity	%IW141.42	-	-	default = 0
FCODE_nC475_2_v			%IW141.43			

Example

It is possible to enter a percentage [%] under code C0472/1 of the PLC by using, for instance, the keypad. This value is directly assigned to the variable *FCODE_nC472_1_a* (data type “Integer”) and can be processed further in the PLC program.

2.10.2 Outputs_FCODE

Variable name	Data type	Signal type	Address	DIS	DIS format	Note
FCODE_wC135	Word	-	%QW141.0	-	-	
FCODE_bC135Bit0_b ... FCODE_bC135Bit15_b	boolean	binary	%QX141.0.0 ... %QX141.0.15	-	-	Also input variable (see table above)
FCODE_wC150	Word	-	%QW141.1	-	-	
FCODE_bC150Bit0_b ... FCODE_bC150Bit15_b	boolean	binary	%QX141.1.0 ... %QX141.1.15	-	-	



2.11 SYSTEM_FLAGS

(Module number 151)

System flags are global variables which are permanently integrated into the run-time system. They include functions that facilitate programming.

The following system flags are included in the **9300 Servo PLC**/**Drive PLC**:

VariableName	DataType	Address	Note
SYSTEM_bClock01Hz	Bool	%IX151.0.0	0.1 Hz System clock
SYSTEM_bClock1Hz		%IX151.0.8	1.0 Hz System clock
SYSTEM_bClock10Hz		%IX151.1.0	10 Hz System clock
SYSTEM_bClock0100Hz		%IX151.1.8	100 Hz System clock
SYSTEM_bTogCycleTask		%IX151.2.0	Toggle marker cyclic task
SYSTEM_b1LoopCyclicTask		%IX151.2.8	First loop cyclic task
SYSTEM_b1LoopTask2		%IX151.3.0	First loop task ID2
SYSTEM_b1LoopTask3		%IX151.3.8	First loop task ID3
SYSTEM_b1LoopTask4		%IX151.4.0	First loop task ID4
SYSTEM_b1LoopTask5		%IX151.4.8	First loop task ID5
SYSTEM_b1LoopTask6		%IX151.5.0	First loop task ID6
SYSTEM_b1LoopTask7		%IX151.5.8	First loop task ID7
SYSTEM_b1LoopTask8		%IX151.6.0	First loop task ID8
SYSTEM_b1LoopTask9		%IX151.6.8	First loop task ID9
SYSTEM_nTaskInterval	Integer	%IW151.7	Interval of current task
SYSTEM_nTaskID		%IW151.8	ID-number of current task

Tip!

The system variables are not generated in simulation mode.

SYSTEM_bClockxHz

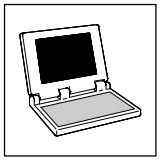
These system flags output a fixed clock pulse with an equal pulse/pause ratio.

- The flag is toggled in real time.
- When you use this system flag, take care with the frequency used for polling the flag (aliasing effect). You should use at least twice the toggle frequency.

Example:

You would like to use the system flag *SYSTEM_bClock100Hz* as a clock for a counter.

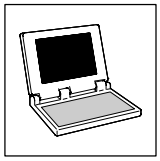
- The pulse/pause ratio is 5 msec/5 msec.
- To avoid an aliasing effect, the counter must always be polled with an INTERVAL-TASK



Drive PLC

System blocks

SYSTEM_FLAGS



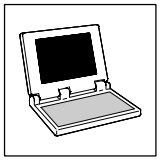
3 Standard-I/O FIF module

The **Standard-I/O** FIF module extends the Drive PLC by the following inputs and outputs:

	Signal type	Number	SB			Module number	Info
Inputs	Analog	1	ANALOG_IO_FIF	→ Inputs_ANALOG_FIF	(AIN1_FIFSTDIO)	201	🕮 3-2
	Digital	5	DIGITAL_IO_FIF	→ Inputs_DIGITAL_FIF	(DIGIN_FIFSTDIO)	200	🕮 3-3
Outputs	Analog	1	ANALOG_IO_FIF	→ Outputs_ANALOG_FIF	(AOUT1_FIFSTDIO)	201	🕮 3-4
	Digital	1	DIGITAL_IO_FIF	→ Outputs_DIGITAL_FIF	(DIGOUT_FIFSTDIO)	200	🕮 3-5

Tip!

Observe the installation kit for the **Standard-I/O FIF module**!



3.1 ANALOG_IO_FIF

3.1.1 Inputs_ANALOG_FIF

Analog input of FIF module “Standard-I/O” (module number 201)

This SB forms the interface for analog signals via terminal 8, of the FIF module. **Standard-I/O** as setpoint input or actual value input.

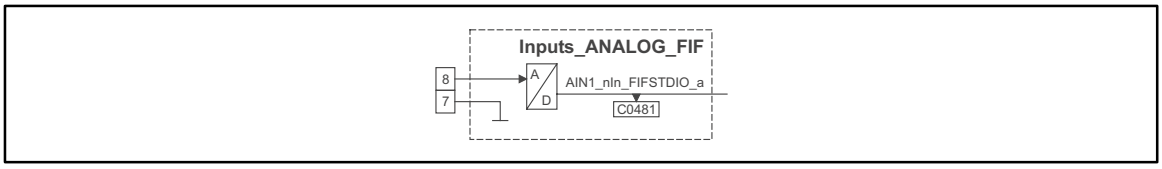


Fig. 3-1 Inputs_ANALOG_FIF

Variable name	Data type	Signal type	Address	DIS	DIS format	Note
AIN1_nIn_FIFSTDIO_a	Integer	analog	%IW201.0	C0481	dec [%]	Analog input 1 Standard-I/O

Input signal setting

By suitable switch settings at the FIF module **Standard-I/O** the following input signals can be processed:

Input range[1] for terminal 8	Switch setting				
	1	2	3	4	5
0 ... +5 V	OFF	OFF	ON	OFF	OFF
0 ... +10 V (Lenze setting)	**OFF**	**OFF**	**ON**	**OFF**	**ON**
-10 ... +10 V	ON	ON	OFF	OFF	OFF
0 ... +20 mA	OFF	OFF	ON	ON	OFF

[1] The selected input range is adjusted to the integer range (0 ... 16384) of the variable *AIN1_nIn_FIFSTDIO_a*.

Example: selected input range = 0 ... +10 V			
Input signal (terminal 8)	0 V	+ 5 V	+ 10 V
AIN1_nIn_FIFSTDIO_a	0	8192	16384

Electrical data of input terminals:

Terminal	Use	level	Data
8	Analog input 1	0 ... +5 V 0 ... +10 V -10 ... +10 V 0 ... +20 mA	Resolution: 10 bit Linearity error: ±0.5 % Temperature error: ±0.3 % (0 ... 60 °C) Input resistance: > 50 kΩ (voltage signal) 250 Ω (current signal)
7	GND1, reference potential for analog signals	-	-

Tip!

Detailed information about the terminal assignment for the FIF module **Standard-I/O** can be found in the corresponding Mounting Instructions!

3.1.2 Outputs_ANALOG_FIF

Analog “Standard-I/O” FIF module output (module number 201)

This SB can be used as a monitor output. Internal signals can be given out as voltage signals via terminal 62 of the **Standard-I/O** FIF module and used, e.g. as display or setpoint values for following drives.

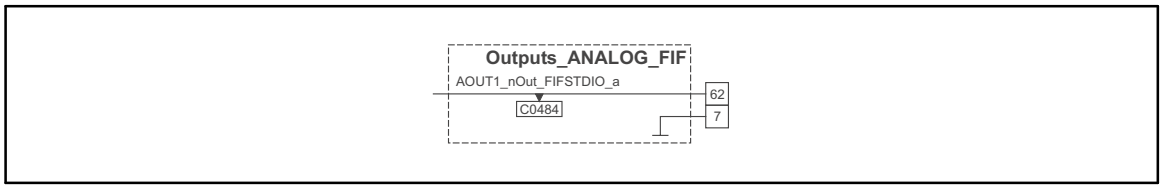


Fig. 3-2 Outputs_ANALOG_FIF

VariableName	DataType	SignalType	Address	DIS	DIS format	Note
AOUT1_nOut_FIFSTDIO_a	Integer	analog	%QW201.0	C0484	dec [%]	Analog output 1 Standard-I/O

function

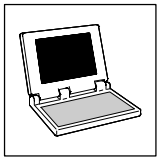
A voltage of 10 V is given out at terminal 62 of the **Standard-I/O** FIF module if the signal at *AOUT1_nOut_FIFSTDIO_a* = 16384 = 100 %

Electrical data of the output terminals:

Terminal	Use	level	Data
62	Analog output 1	0 ... +10 V	Resolution: 10 bit Linearity error: ±0.5 % Temperature error: ±0.3 % (0 ... 60 °C) Load capability: max. 2 mA
7	GND1, reference potential for analog signals	-	-

Tip!

More detatiled information about the terminal assignment of the **Standard-I/O** FIF module can be found in the corresponding Mounting Instructions!



3.2 DIGITAL_IO_FIF

3.2.1 Inputs_DIGITAL_FIF

Digital inputs of “Standard-I/O” FIF module (module number 200)

This SB reads the signals at terminals E1 ... E4, 28 of the **Standard-I/O** FIF module and conditions them.

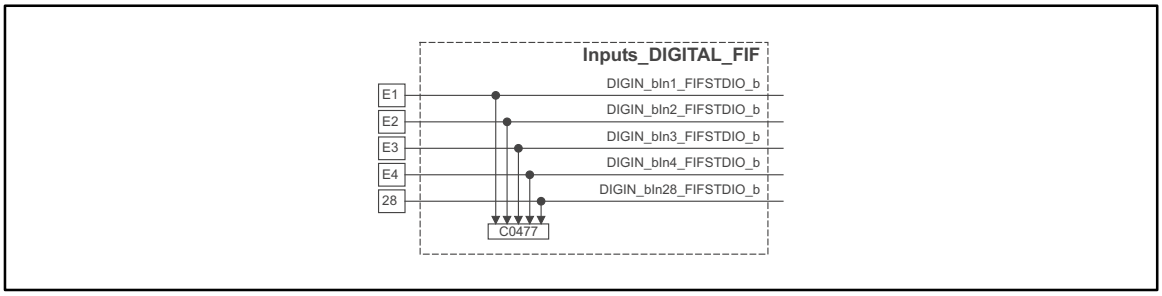


Fig. 3-3 Inputs_DIGITAL_FIF

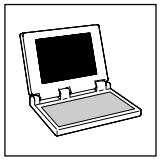
Variable name	Data type	Signal type	Address	DIS	DIS format	Note
DIGIN_bIn1_FIFSTDIO_b	boolean	binary	%IX200.0.0	C0477 - Bit2	bin	
DIGIN_bIn2_FIFSTDIO_b			%IX200.0.1	C0477 - Bit3		
DIGIN_bIn3_FIFSTDIO_b			%IX200.0.2	C0477 - Bit4		
DIGIN_bIn4_FIFSTDIO_b			%IX200.0.3	C0477 - Bit5		
DIGIN_bIn28_FIFSTDIO_b			%IX200.0.4	C0477 - Bit1		

Electrical data of input terminals:

Terminal	Use	Data	
E1	freely assignable	LOW level:	0 ... +3 V
E2		HIGH level:	+12 ... +30 V (PLC level, HTL)
E3		Input resistance:	3.3 kΩ
E4			
28			

Tip!

Detailed information about the terminal assignment for the **Standard-I/O** FIF module can be found in the corresponding Mounting Instructions!



3.2.2 Outputs_DIGITAL_FIF

Digital “Standard-I/O” FIF module output (module number 200)

This SB conditions a digital signal and transmits it to terminal A1 of the **Standard-I/O** FIF module.

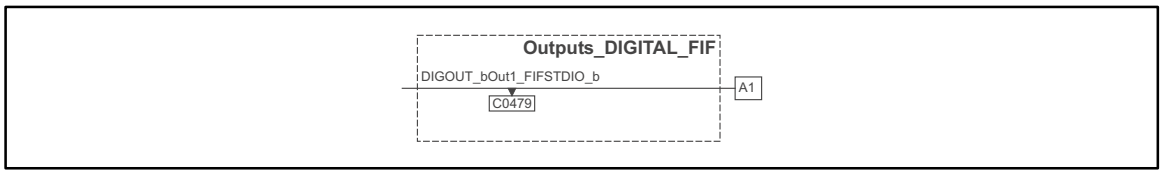


Fig. 3-4 Outputs_DIGITAL_FIF

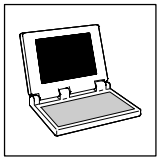
Variable name	Data type	Signal type	Address	DIS	DIS format	Note
DIGOUT_bOut1_FIFSTDIO_b	boolean	binary	%QX200.0.0	C0479 - Bit1	bin	

Electrical data of output terminal

Terminal	Use	Data	
A1	freely assignable	Output voltage: Output current:	0/+20 V with DC internal 0/+24 V with DC external 10 mA with DC internal 50 mA with DC external

Tip!

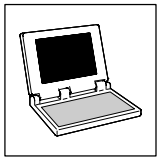
More detailed information about the terminal assignment of the **Standard-I/O** FIF module can be found in the corresponding Mounting Instructions!



Drive PLC

System blocks - FIF Standard-I/O

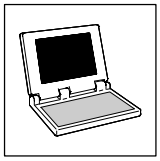
DIGITAL_IO_FIF



4 CAN-I/O FIF module

The **CAN-I/O** FIF module extends the Drive PLC by the following inputs and outputs:

	Signal type	Number	SB			Module number	Info
Inputs	digital	1	FIF_CAN_DIGITAL_IN	→ FIF_CAN_Inputs_DIGITAL		202	🕮 4-2
	System bus (CAN)		FIF_CAN1_IO	→ FIF_Inputs_CAN1	(FIF-CAN1_IN)	34	*
			FIF_CAN2_IO	→ FIF_Inputs_CAN2	(FIF-CAN2_IN)	35	
			FIF_CAN3_IO	→ FIF_Inputs_CAN3	(FIF-CAN3_IN)	36	
Outputs	System bus (CAN)		FIF_CAN1_IO	→ FIF_Outputs_CAN1	(FIF-CAN1_OUT)	34	
			FIF_CAN2_IO	→ FIF_Outputs_CAN2	(FIF-CAN2_OUT)	35	
			FIF_CAN3_IO	→ FIF_Outputs_CAN3	(FIF-CAN3_OUT)	36	
* see “System bus (CAN) for Servo PLC & Drive PLC” Manual							



4.1 FIF_CAN_DIGITAL_IN

4.1.1 FIF_CAN_Inputs_DIGITAL

Digital input of "CAN-I/O" FIF module (module number 202)

This SB reads the signal at terminal 28 of the **CAN-I/O** FIF module and preprocesses it.

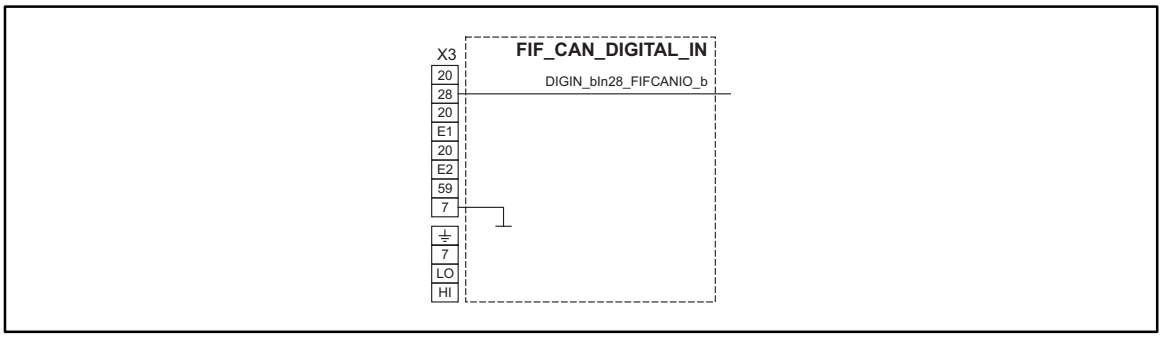


Fig. 4-1 FIF_CAN_DIGITAL_IN

Variable name	Data type	Signal type	Address	DIS	DIS format	Note
DIGIN_bIn28_FIFCANIO_b	boolean	binary	%IX202.0.0	-	-	

Electrical data of input terminal

Terminal	Use	Data
28	Freely assignable	LOW level: 0 ... +3 V HIGH level: +12 ... +30 V (PLC level, HTL) Input resistance: 3.3 kΩ

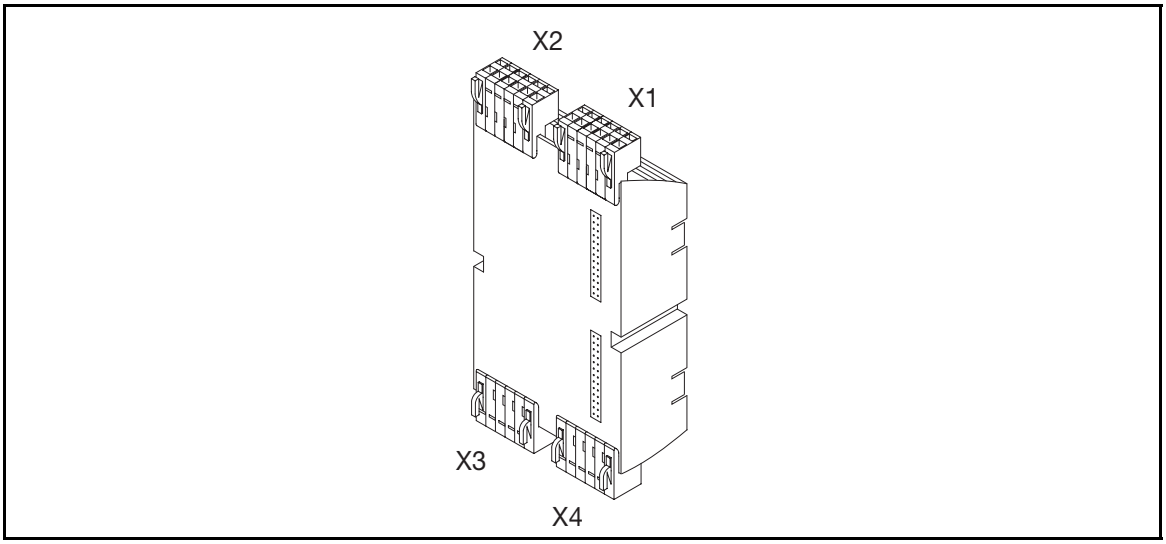
Tip!

More detatiled information about the terminal assignment of the **CAN-I/O** FIF module can be found in the corresponding Mounting Instructions!

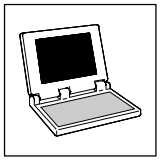
5 Extension board 1

The internal **Extension board 1** extends the Drive PLC by the following inputs/outputs:

	Signal type	Number	SB			Module number	Info
Inputs	Digital	6	DIGITAL_IO_EB1	→ Inputs_DIGITAL_EB1	(DIGIN_EB1)	2	📖 5-2
Outputs	Digital	6	DIGITAL_IO_EB1	→ Outputs_DIGITAL_EB1	(DIGOUT_EB1)		📖 5-3



Terminal	Assignment
X1	GND, reference potential
X2	DC supply voltage (switched via X1/+024 at the Drive PLC)
X3	Digital outputs O5 ... O10
X4	Digital inputs I9 ... I14



5.1 DIGITAL_IO_EB1

5.1.1 Inputs_DIGITAL_EB1

Digital inputs of extension board 1 (module number 2)

This SB reads the signals at the terminals I9 ... I14 of the **extension board 1** and conditions them.

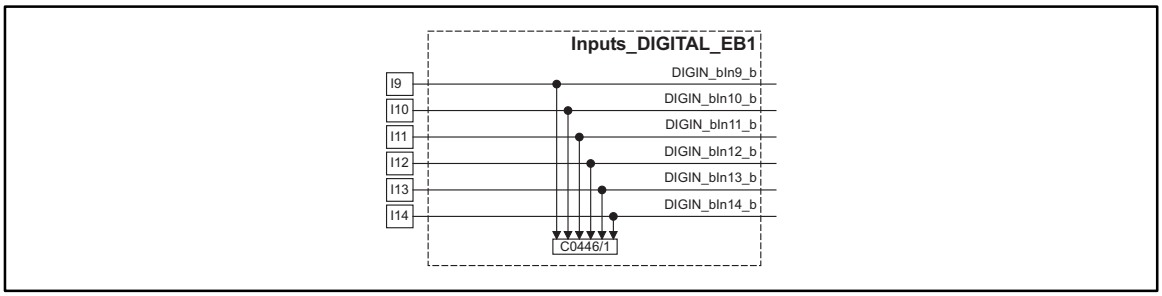


Fig. 5-1 Inputs_DIGITAL_EB1

Variable name	Data type	Signal type	Address	DIS	DIS format	Note
DIGIN_bIn9_b ... DIGIN_bIn14_b	boolean	binary	%IX2.0.0 ... %IX2.0.5	C0446/1-Bit1 ... C0446/1-bit6	bin	

Electrical data of input terminals

Terminal	Use	Data	
I9 ... I14	freely assignable	LOW level: HIGH level: Input current:	0 ... +4 V +13 ... +30 V 8 mA per input (at 24 V)

Tip!

Detailed information about the terminal assignment for the **extension board 1** can be found in the corresponding Mounting Instructions!

5.1.2 Outputs_DIGITAL_EB1

Digital extension board 1 outputs (module number 2)

This SB conditions the digital signals, and outputs them at terminals O5 ... O10 of the **extension board 1**.

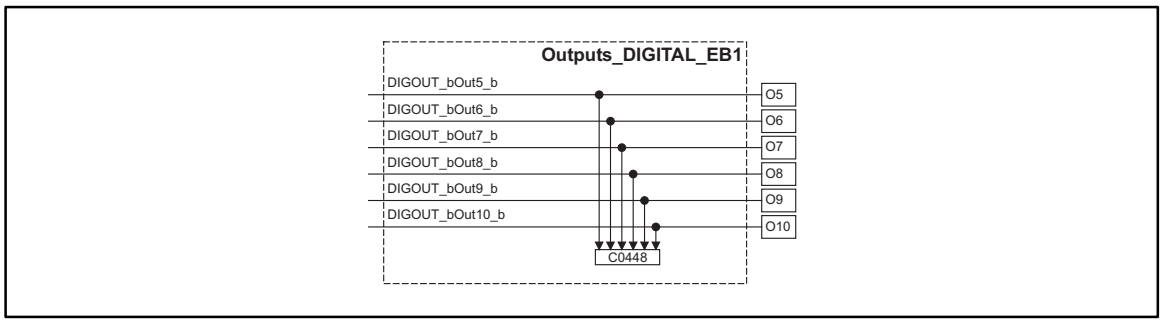


Fig. 5-2 Outputs_DIGITAL_EB1

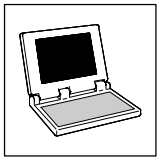
Variable name	Data type	Signal type	Address	DIS	DIS format	Note
DIGOUT_bOut5_b ... DIGOUT_bOut10_b	Bool	binary	%QX2.0.0 ... %QX2.0.5	C0448-Bit1 ... C0448-Bit6	bin	

Electrical data of output terminals

Terminal	Use	Data	
O5 ... O10	freely assignable	LOW level: HIGH level: Output current:	0 ... +4 V +13 ... +30 V max. 1 A per output

Tip!

More detatiled information about the terminal assignment of the **extension board 1** can be found in the corresponding Mounting Instructions!



Drive PLC

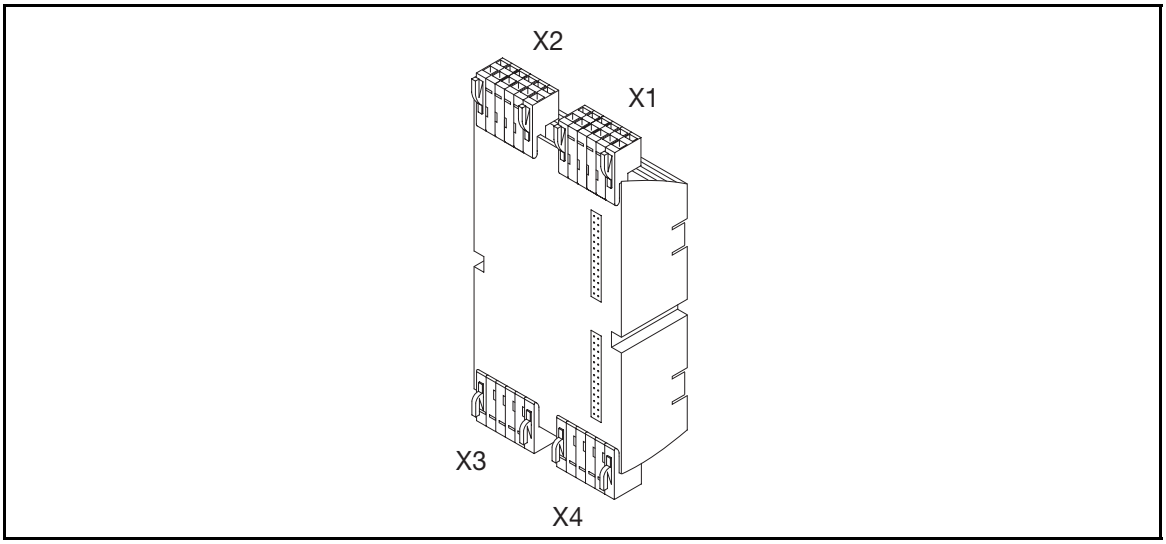
System blocks - Extension Board 1

DIGITAL_IO_EB1

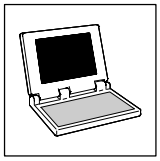
6 Extension board 2

The internal **extension board 2** extends the Drive PLC by the following inputs/outputs:

	Signal type	Number	SB			Module number	Info
Inputs	Digital	14	DIGITAL_IO_EB2	→ Inputs_DIGITAL_EB2	(DIGIN_EB2)	3	📖 6-2
Outputs	Digital	8	DIGITAL_IO_EB2	→ Outputs_DIGITAL_EB2	(DIGOUT_EB2)		📖 6-3



Terminal	Assignment
X1	Digital outputs O5 ... O10
X2	Digital inputs I21, I22
	Digital outputs O11, O12
	GND, reference potential
	DC supply voltage (switched via X1/+O24 at the Drive PLC)
X3	Digital inputs I15 ... I20
X4	Digital inputs I9 ... I14



6.1 DIGITAL_IO_EB2

6.1.1 Inputs_DIGITAL_EB2

Digital extension board 2 inputs (module number 3)

This SB reads the signals at terminals I9 ... I22 of the **extension board 2** and conditions them.

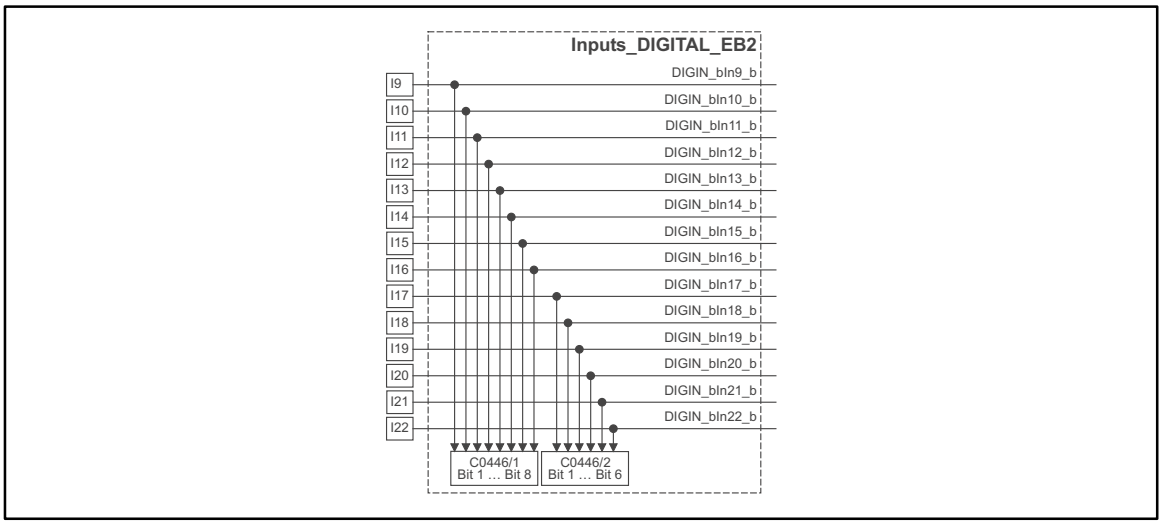


Fig. 6-1 Inputs_DIGITAL_EB2

Variable name	Data type	Signal type	Address	DIS	DIS format	Note
DIGIN_bIn9_b	boolean	binary	%IX3.0.0	C0446/1-Bit1	bin	
...			...	...		
DIGIN_bIn16_b			%IX3.0.7	C0446/1-Bit8		
DIGIN_bIn17_b			%IX3.0.8	C0446/2-Bit1		
...			...	...		
DIGIN_bIn22_b			%IX3.0.13	C0446/2 bit6		

Electrical data of input terminals

Terminal	Use	Data	
I9	freely assignable	LOW level:	0 ... +4 V
...		HIGH level:	+13 ... +30 V
I22		Input current:	8 mA per input (at 24 V)

Tip!

More detailed information about the terminal assignment of the **extension board 2** can be found in the corresponding Mounting Instructions!

6.1.2 Outputs_DIGITAL_EB2

Digital extension board 2 outputs (module number 3)

This SB conditions the digital signals, and outputs them at terminals O5 ... O12 of the **extension board 2** .

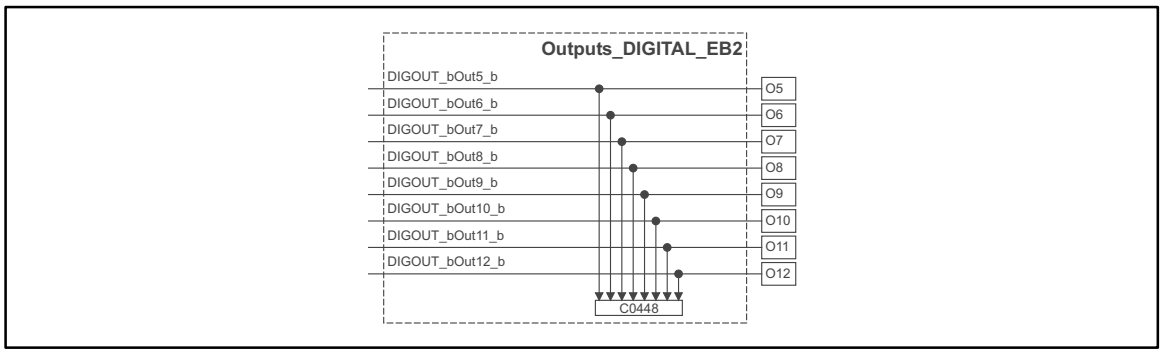


Fig. 6-2 Outputs_DIGITAL_EB2

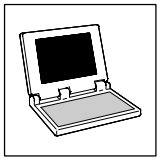
Variable name	Data type	Signal type	Address	DIS	DIS format	Note
DIGOUT_bOut5_b ... DIGOUT_bOut12_b	Bool	binary	%QX3.0.0 ... %QX3.0.7	C0448-Bit1 ... C0448-Bit8	bin	

Electrical data of the output terminals

Terminal	Use	Data	
O5 ... O12	freely assignable	LOW level: HIGH level: Output current:	0 ... +4 V +13 ... +30 V max. 1 A per output

Tip!

More detailed information about the terminal assignment of the **extension board 2** can be found in the corresponding Mounting Instructions!



Drive PLC

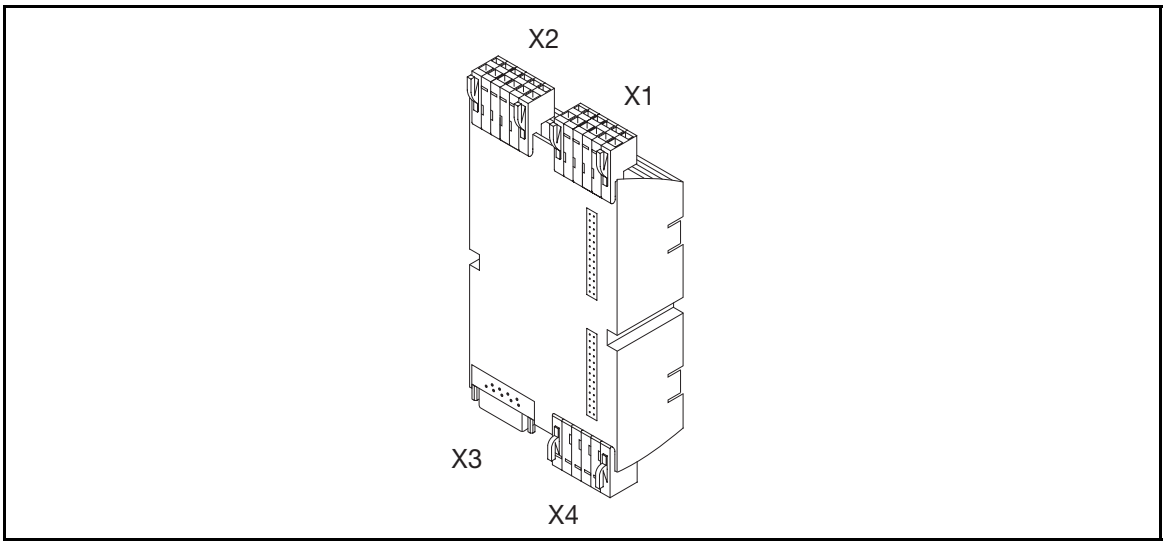
System blocks - Extension Board 2

DIGITAL_IO_EB2

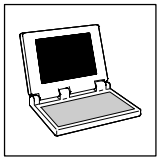
7 Extension board 3

The internal **extension board 3** extends the Drive PLC by the following inputs/outputs:

	Signal type	Number	SB			Module number	Info
Inputs	Analog	2	ANALOG4_IO_EB3	→ Inputs_ANALOG4	(AIN4)	14	📖 7-2
			ANALOG5_IO_EB3	→ Inputs_ANALOG5	(AIN5)	15	📖 7-3
	Digital	8	DIGITAL_IO_EB3	→ Inputs_DIGITAL_EB3	(DIGIN_EB3)	4	📖 7-9
	Encoder	1	DFIN_IO_ DigitalFrequency	→ Inputs_DFIN	(DF_IN)	21	📖 7-4
Outputs	Digital	4	DIGITAL_IO_EB3	→ Outputs_DIGITAL_EB3	(DIGOUT_EB3)	4	📖 7-10



Terminal	Assignment
X1	Digital inputs I15, I16
	Digital outputs O5 ... O8
X2	Analog inputs AI4, AI5
	GND, reference potential
	DC supply voltage (switched via X1/+O24 at the Drive PLC)
X3	Incremental encoder/counter input (Sub-D connector, 9 pole)
X4	Digital inputs I9 ... I14



7.1 ANALOG4_IO_EB3

7.1.1 Inputs_ANALOG4

Analog input 4 extension board 3 (module number 14)

This SB is the interface for analog signals via terminal AI4, A⊥ of the **extension board 3**.

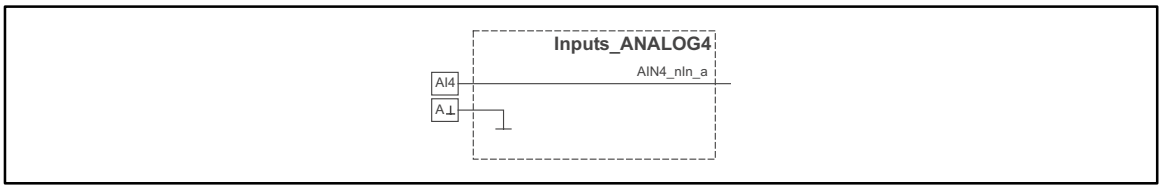


Fig. 7-1 Inputs_ANALOG4

Variable name	Data type	Signal type	Address	DIS	DIS format	Note
AIN4_nIn_a	Integer	analog	%IW14.0	-	dec [%]	Analog input 4

Terminal assignment

Terminal X2	Use	Data
AI4	Analog input 4	Level: -10 V ... +10 V Resolution: 10 bits + sign Internal resistance: 100 k Ω Normalisation: ±10 V ≡ ±16384 ≡ ±100 %
A⊥	Internal ground, GND	-

7.2 ANALOG5_IO_EB3

7.2.1 Inputs_ANALOG5

Analog input 5 extension board 3 (module number 15)

This SB is the interface for analog signals via terminal AI5, A⊥ of the **extension board 3**.

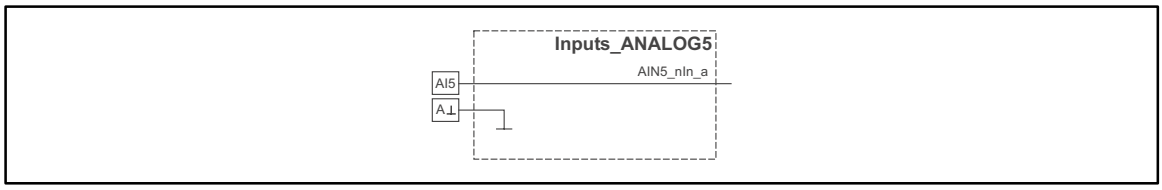
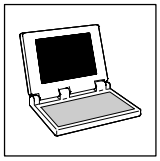


Fig. 7-2 Inputs_ANALOG5

Variable name	Data type	Signal type	Address	DIS	DIS format	Note
AIN5_nIn_a	Integer	analog	%IW15.0	-	-	Analog input 5

Terminal assignment

Terminal X2	Use	Data
AI5	Analog input 5	Level: -10 V ... +10 V Resolution: 10 bits + sign Internal resistance: 100 k Ω Normalisation: ±10 V ≡ ±16384 ≡ ±100 %
A⊥	Internal ground, GND	-



7.3 DFIN_IO_DigitalFrequency (encoder input)

7.3.1 Inputs_DFIN

Extension board 3 encoder input (module number 21)

This SB can convert and normalise a pulse current at the encoder input X3 of the **extension board 3** into a speed value.

- The transmission is very precise without offset and gain errors.
- In addition, this SB provides the angle correction value *DFIN_dnIncLastScan_p* which is required for the angle processing in touch probe processes. (📖 7-7)

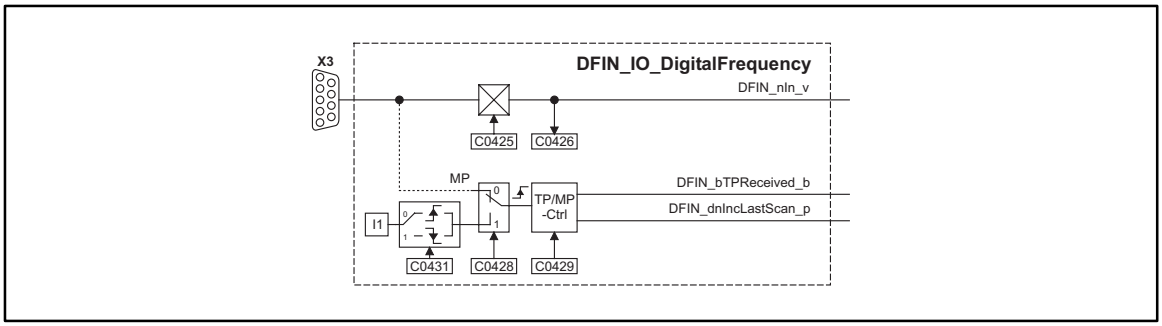


Fig. 7-3 DFIN_IO_DigitalFrequency

Variable name	Data type	Signal type	Address	DIS	DIS format	Note
DFIN_nIn_v	Integer	velocity	%IW21.0	C0426	dec [rpm]	
DFIN_bTPReceived_b	Bool	binary	%IX21.4.0	-	-	
DFIN_dnIncLastScan_p	Double integer	position	%ID21.1	-	-	

Tip!

The process image is newly created for every task the SB is used in.

- If *DFIN_nIn_v*, *DFIN_dnIncLastScan_p* and *DFIN_bTPReceived_b* are used in several tasks, every task gets its own SB process image.
- This process is different from the previous process image principle!

7.3.1.1 Encoder input X3

- Encoder input X3 of the **extension board 3** is designed for signals with TTL level.
- The input of a zero track is optional.
- The drive can be adapted to the connected encoder or preconnected controller with master frequency cascade or master frequency bus via under C0425.

Code	LCD	Possible settings		IMPORTANT
		Lenze	Selection	
C0425	DFIN const	3		Extension board 3: DFIN_IO Increments of the encoder input
			0 256 inc/rev	
			1 512 inc/rev	
			2 1024 inc/rev	
			3 2048 inc/rev	
			4 4096 inc/rev	
			5 8192 inc/rev	
			6 16384 inc/rev	

Evaluation of master frequency signals

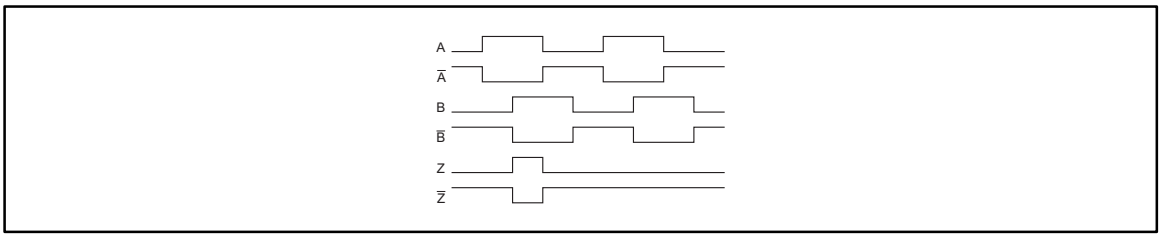


Fig. 7-4 Signal sequence with phase shift (CW rotation)

- CW rotation: Track A leads track B by 90 ° (positive value at *DFIN_nIn_v*).
- CCW rotation: Track A lags track B by 90 ° (negative value at *DFIN_nIn_v*).

Transmission function

$$\text{DFIN\_nIn\_v} = \text{f [Hz]} \cdot \frac{60}{\text{increments from C0425}} \cdot \frac{2^{14}}{15000}$$

Example:

- Input frequency = 200 kHz
- C0425 = 3, this corresponds to 2048 increments/rev.

$$\text{DFIN\_nIn\_v [rpm]} = 200000\ \text{Hz} \cdot \frac{60}{2048} = 5859\ \text{rpm}$$

Signal adaptation

Finer resolutions can be achieved by adding a following FB (e.g. L_CONVV from LenzeDrive.lib):

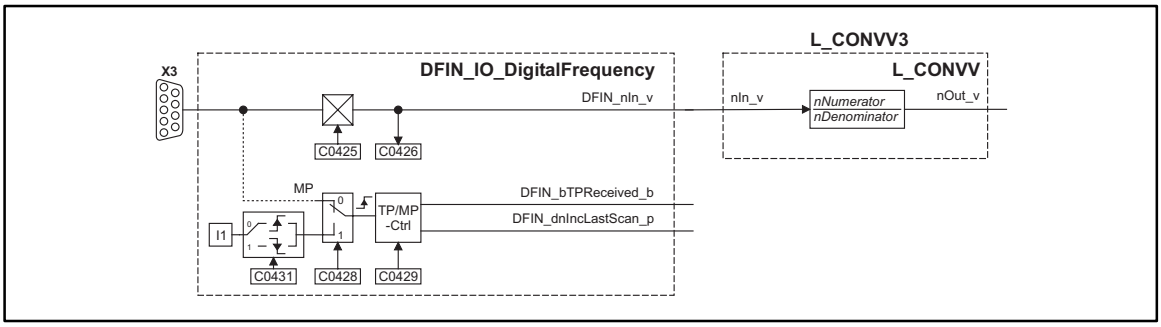
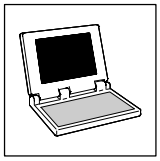


Fig. 7-5 Encoder input (DFIN_IO_DigitalFrequency) with following FB L_CONVV for normalisation

$$\text{nOut\_a} = \text{f [Hz]} \cdot \frac{60}{\text{increments from C0425}} \cdot \frac{\text{n numerator}}{\text{n denominator}} \cdot \frac{2^{14}}{15000}$$



7.3.1.2 Connection of incremental encoders to terminal X3

Stop!

Observe connection voltage of the incremental encoder used!

Incremental encoder with TTL level

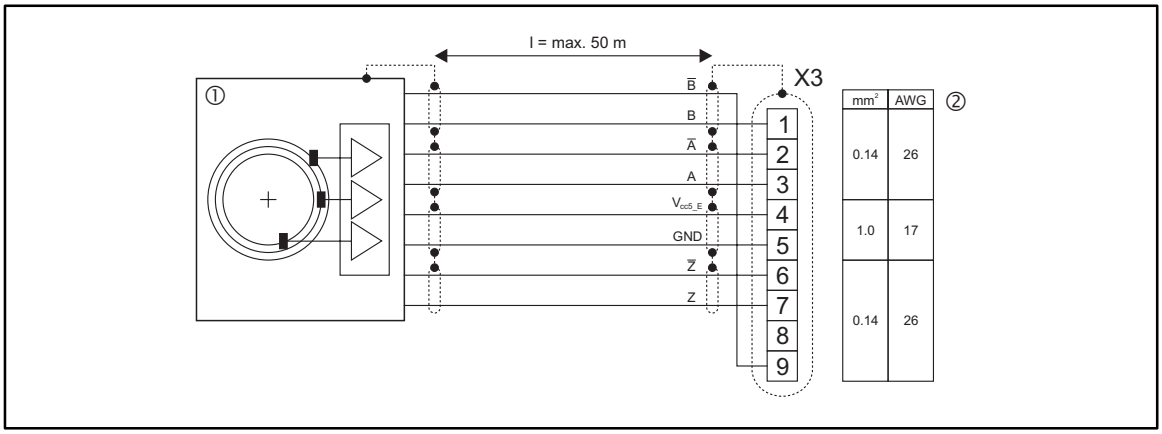


Fig. 7-6 Connection of incremental encoder with TTL level to terminal X3

① Incremental encoder
② Cross-sections to be used

Assignment of the Sub-D connector (X3)									
PIN	1	2	3	4	5	6	7	8	9
Signal	B	$\overline{A}$	A	V_{cc5_E}	GND	$\overline{Z}$	Z	-	$\overline{B}$

Incremental encoder with HTL level

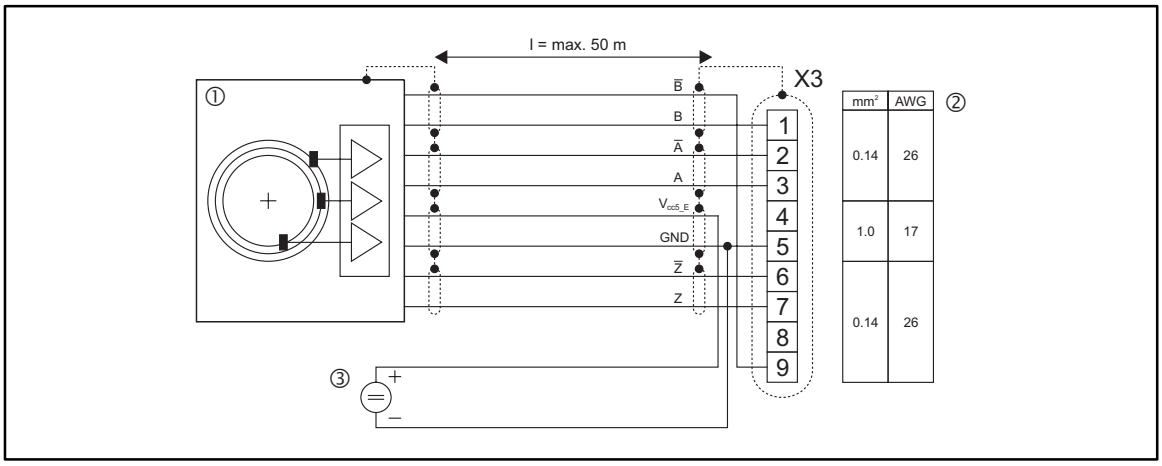


Fig. 7-7 Connection of incremental encoder with HTL level to terminal X3

① Incremental encoder
② Cross-sections to be used
③ Supply voltage for incremental encoder

Assignment of the Sub-D connector (X3)									
PIN	1	2	3	4	5	6	7	8	9
Signal	B	$\overline{A}$	A	+5 V	GND	$\overline{Z}$	Z	-	$\overline{B}$

Tip!

You can connect incremental encoders with HTL level that supply signals A and B only, to PIN 2 and PIN 9. The inputs at PIN 3 and PIN 1 must then be connected to the supply voltage of the incremental encoder.

The connection is as shown in the figures:

- Use twisted pair cables and screened pair cables.
- Connect the screen at both ends.
- Do no change the cable cross-sections indicated.

7.3.1.3 Touch probe (TP)

Process: The current angle value (encoder input value) is saved by a quick interrupt in the operating system when a signal changes at the TP activating input (e.g. X3/I1).

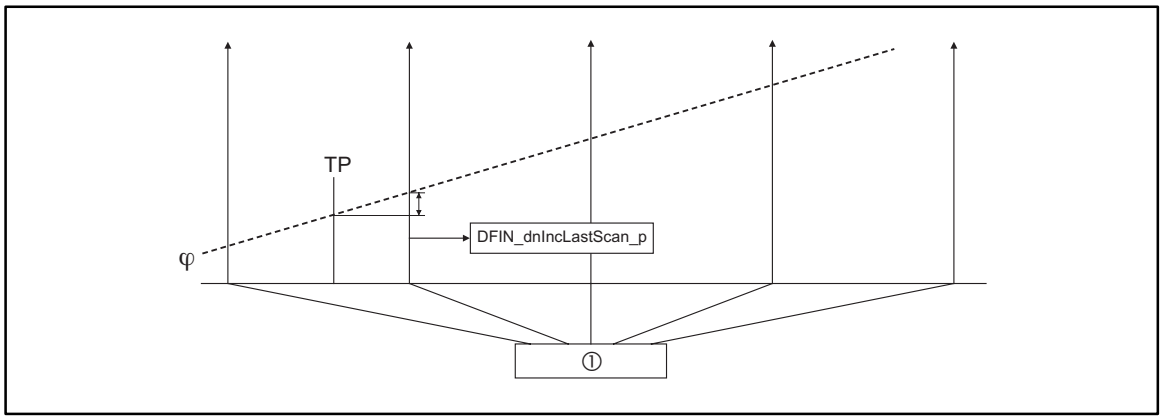
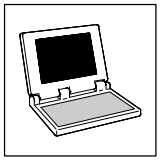


Fig. 7-8 Function diagram of a TP

① Time-equidistant start of an interval-task
φ Phase-angle signal

Touch probe configuration

Code	LCD	Possible settings		IMPORTANT
		Lenze	Selection	
C0428	DFIN TP sel.	0	0 Touch probe via zero pulse 1 Touch probe via digital input X3/I1	Touch probe selection
C0429	TP delay	0	-32767 {1 inc} 32767	Touch probe delay • Compensation of delay times of TP signal source at X3/I1
C0431	DFIN TP EDGE	0	0 Activation with positive signal 1 Activation with negative signal	Touch probe activation • For touch probe via digital input X3/I1 (C0428 = 1)



Drive PLC

System blocks - Extension Board 3

DFIN_IO_DigitalFrequency (encoder input)

Functional sequence

1. The TP is signal-controlled via the digital input X3/I1 of the basic unit or a zero pulse (only if an encoder is connected).
2. If a TP has occurred, then *DFIN_bTPReceived_b* is set = TRUE.
3. After the start of the task *DFIN_dnIncLastScan_p* indicates the number of increments [inc] counted since the TP.
4. Following, *DFIN_bTPReceived_b*= FALSE is set.

Note!

It is necessary that all three output (*DFIN_nIn_v*, *DFIN_bTPReceived_b* and *DFIN_dnIncLastScan_p*) are processed in the task even if just one signal is required.

DFIN_nIn_v

- The value *DFIN_nIn_v* is scaled in increments per millisecond.
  (INT) 16384 corresponds to 15000 rpm. See chapter 1.2.7, “Signal types and normalisations”. (📖 1-8)
- For every task in which *DFIN_nIn_v* is used, the operating system creates an individual counter that is reset after every start of the task (task-internal process image).

Example (*DFIN_nIn_v* in a 10 msec task):

- When the 10 msec task starts, the value of the integrator is stored in a local area of the task and the integrator is reset. The value in the local area gives an average value in increments per 1 msec.
- If a position value is to be derived from this value, then it must be multiplied by *SYSTEM_nTaskInterval* / 4, to get the result in increments per 10 msec, as in the example.
  Example: In a 1 msec task *SYSTEM_nTaskInterval* is 4 (4 * 250 µs = 1 ms)
- For Lenze FBs, this process has already been implemented in the FBs.

7.4 DIGITAL_IO_EB3

7.4.1 Inputs_DIGITAL_EB3

Digital inputs of extension board 3 (module number 4)

This SB reads the signals at terminals I9 ... I16 of the **extension board 3** and conditions them.

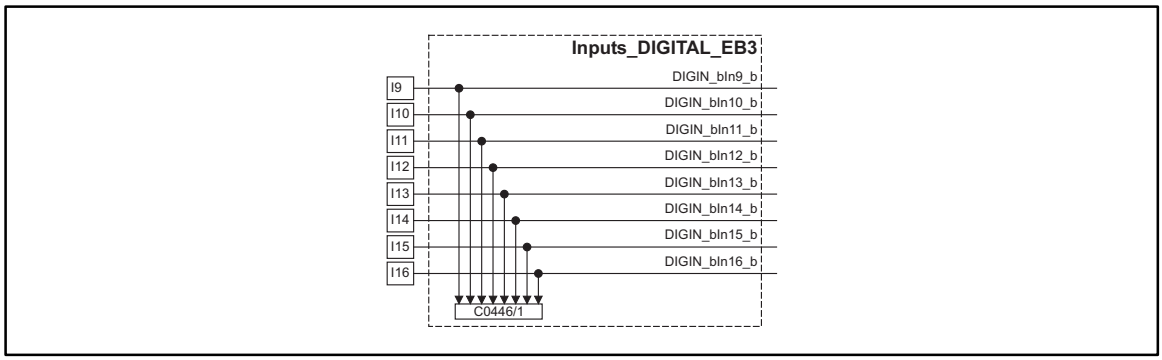


Fig. 7-9 Inputs_DIGITAL_EB3

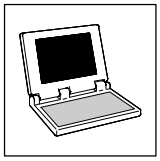
Variable name	Data type	Signal type	Address	DIS	DIS format	Note
DIGIN_bIn9_b ... DIGIN_bIn16_b	boolean	binary	%IX4.0.0 ... %IX4.0.7	C0446/1-bit1 ... C0446/1-bit8	bin	

Electrical data of input terminals

Terminal	Use	Data	
I9 ... I16	freely assignable	LOW level: HIGH level: Input current:	0 ... +4 V +13 ... +30 V 8 mA per input (at 24 V)

Tip!

Detailed information about the terminal assignment for the **extension board 3** can be found in the corresponding Mounting Instructions!



7.4.2 Outputs_DIGITAL_EB3

Digital extension board 3 outputs (module number 4)

This SB conditions the digital signals, and outputs them at terminals O5 ... O8 of the **extension board 3** .

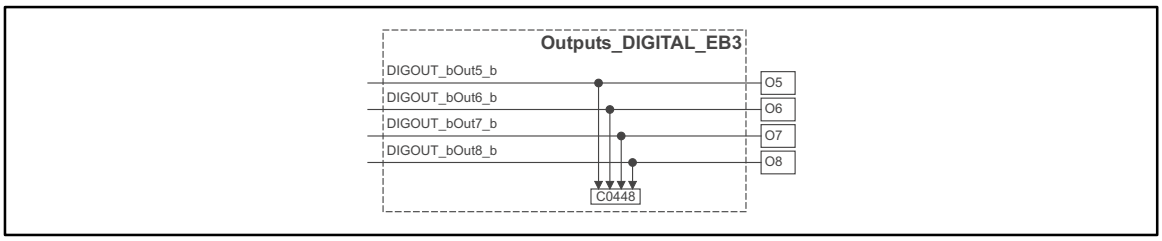


Fig. 7-10 Outputs_DIGITAL_EB3

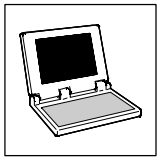
Variable name	Data type	Signal type	Address	DIS	DIS format	Note
DIGOUT_bOut5_b ... DIGOUT_bOut8_b	boolean	binary	%QX4.0.0 ... %QX4.0.3	C0448-Bit1 ... C0448-Bit4	bin	

Electrical data of the output terminals

Terminal	Use	Data	
O5 ... O8	freely assignable	LOW level: HIGH level: Output current:	0 ... +4 V +13 ... +30 V max. 1 A per output

Tip!

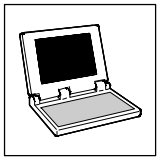
More detailed information about the terminal assignment of the **extension board 3** can be found in the corresponding Mounting Instructions!



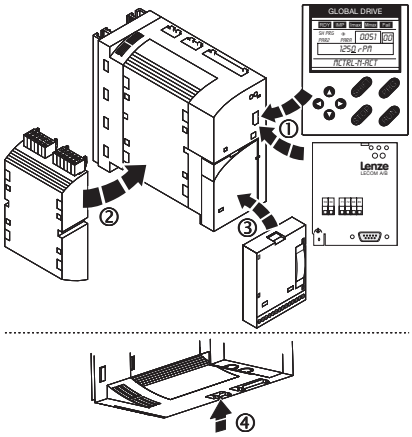
8 Appendix

8.1 PLC functionality

Field		Number	Description	
Inputs	digital	8	free inputs (3 of them interruptible)	24 V DC / 8 mA each input
	analog	3	free inputs (10 bit + sign)	±10 V
Outputs	digital	4	free outputs	24 V DC / max. 1 A each output
	analog	1	free output (10 bit + sign)	±10 V / max. 2 mA
Operation set			According to IEC61131-3	
Counter/times			According to IEC61131-3, depending on the data memory available	
Fast counter		1	0 ... 500 kHz	
Label		1000	Label words	
Memory			See chapter 8.3 (8-4)	
Processing time (1-bit operation)			1 µs	
Task types		8	Time or event-controlled tasks (1 ms ... 16 s)	
		1	Cyclic task	
Programming software			Drive PLC Developer Studio • Programming languages according to IEC61131-3 standard (IL, LD, FBD, ST, SFC) as well as CFC editor • Monitoring, visualization, simulation and debugging	



8.2 Extendability/networking



① **Automation interface (AIF)**
for keypad and the following AIF modules:
- 2102 LECOM A/B/LI
- 2103 FP interface (RS-232C)
- 2111 INTERBUS
- 2112 INTERBUS loop
- 2133 PROFIBUS-DP
- 2174 CAN addressing module
- 2175 DeviceNet/CANopen

Others in preparation

② **Plug in station for extension boards**

③ **Function interface (FIF)**
for FIF modules

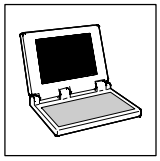
④ **Integrated system bus interface**
(Terminal X5 at the bottom of the Drive PLC)

Extension boards ②

	Inputs/outputs		Number	Description	
Extension board 1					
	Inputs	digital	j6	Free inputs	24 V DC / 8 mA each input
	Outputs	digital	j6	Free outputs	24 V DC / max. 1 A each output
Extension board 2					
	Inputs	digital	14	Free inputs	24 V DC / 8 mA each input
	Outputs	digital	8	Free outputs	24 V DC / max. 1 A each output
Extension board 3					
	Inputs	digital	j6	Free inputs	24 V DC / 8 mA each input
		analog	2	Free inputs (10 bit + sign)	±10 V
			1	Encoder input	TTL/HTL level
	Outputs	digital	j6	Free outputs	24 V DC / max. 1 A each output

Modules for function interfaces (FIF) ③

	Inputs/outputs		Number	Description	
Standard-I/O FIF module					
	Inputs	digital	5	Free inputs	24 V DC / 8 mA each input
		analog	1	Free input (10 bit + sign)	±10 V
	Outputs	digital	1	Free output	24 V DC / max. 1 A
		analog	1	Free output (10 bit + sign)	±10 V / max. 2 mA
FIF module CAN-I/O					
	Inputs	digital	1	Free input	24 V DC / 8 mA
				System bus (CAN)	
	Outputs			System bus (CAN)	

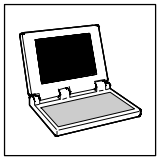


System bus (CAN)

Interface	Available CAN objects	
Integrated system bus interface ④	PDOs	CAN1_IN/CAN1_OUT CAN2_IN/CAN2_OUT CAN3_IN/CAN3_OUT
	SDOs	SDO1 (parameter channel 1) SDO2 (parameter channel 2)
		L_ParRead/L_ParWrite functionality
	Sync telegram	
	Synchronisation of the internal time basis by receiving sync telegrams	
	Free CAN objects	
	CanDSx driver for mapping indices to codes and for bus monitoring functions ”Heartbeat” and ”Node Guarding” (see Manual “Function library LenzeCanDSxDrv.lib”).	
Automation interface (AIF) ① with corresponding fieldbus module (e. g. 2175)	PDOs	XCAN1_IN/XCAN1_OUT XCAN2_IN/XCAN2_OUT XCAN3_IN/XCAN3_OUT
	SDOs	XSDO1 (parameter channel 1) XSDO2 (parameter channel 2)
	XSync telegram	
	AifParMap driver for mapping code accesses via AIF to other codes (see Manual “Function library LenzeAifParMapDrv.lib”).	
Function interface (FIF) ③ with corresponding function module (e. g. CAN-I/O system bus)	PDOs	FIF-CAN1_IN/FIF-CAN1_OUT FIF-CAN2_IN/FIF-CAN2_OUT FIF-CAN3_IN/FIF-CAN3_OUT
	SDOs	FIF-SDO1 (parameter channel 1) FIF-SDO2 (parameter channel 2)
		L_ParRead/L_ParWrite functionality
	Sync telegram	
More detailed information about the system bus (CAN) can be found in the Manual “System bus (CAN) for Servo PLC & Drive PLC”		

Tip!

More detailed information about the system bus (CAN) can be found in the Manual “System bus (CAN) for Servo PLC & Drive PLC”



8.3 Memories

The below table gives you an overview of the memories available:

Memory		Size	Info
ROM			
	Program memory	256 kbytes	Re-written whenever the program is downloaded
RAM			
	PLC data memory	10 kbytes	Can be symbolically used for FB instances and PLC variables.
	Application data memory	2 blocks à 64 kbytes	Data get lost after mains disconnection.
E2PROM-buffered memory			
	Retain memory	200 bytes	See subsection 8.3.1

Tip!

Function library **LenzeMemDrv.lib** includes functions for read/write access to the additional backup memory (application data memory) of the PLC.

- Additional information can be found in the Manual for function library **LenzeMemDrv.lib**.

8.3.1 Retain memory

The values of the retain variables are permanently saved in the retain memory. In this way, they are even after mains disconnection available to the program.

- Retain variables are declared by means of the variable class `VAR RETAIN`.
- Retain variables are used as a memory which can be symbolically addressed.
- Whenever the program is downloaded, the retain variables are reset to their initialization value. If there has no initialization value been selected the corresponding retain variable will be initialized with ”0”.
- In the DDS online mode you can use the commands **Online →Reset (cold)** and / or **Online →Reset (original)** to reset the retain variables to their initialization value.

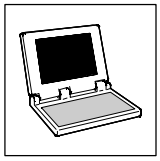
8.3.2 Downloading data

In DDS (as of version 2.0) it is possible to attach a file to your project whose data are automatically downloaded to the PLC when the program is downloaded.

- This mechanism is, for instance, used in the **Software Package - Cam** to download movement profiles.

Note!

- With the 9300 Servo PLC the additional data are loaded into the application FLASH.
- With the Drive PLC the additional data are directly attached to the PLC program because the Drive PLC does not have any application FLASH.

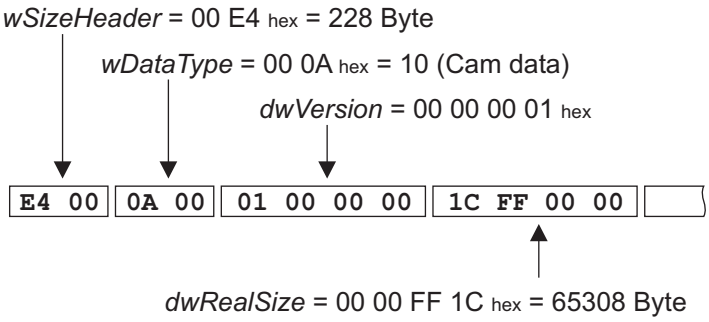


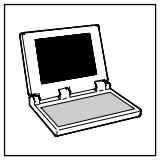
Data will be downloaded if the following conditions are met:

1. The PLC program must have been stopped.
2. The header of the file attached to the project must have the following structure:

Name	Data type	Data length in byte	Contents
wSizeHeader	WORD	2	Header length in bytes
wDataType	WORD	2	Data specification identifier • This information can be found under C2131 after data have been downloaded.
			0 ... 10000: Lenze-specific data
			> 10000: User data
dwVersion	DWORD	4	Data version • This information can be found under C2132 after data have been downloaded.
dwRealSize	DWORD	4	User data length in bytes (without header)
dwTimeStamp	DWORD	4	Time stamp of the last data change • This information can be found under C2133 after data have been downloaded.
wLicenseInfo	WORD	2	*Reserved for future extensions*
wSizeSymbolicName	WORD	2	Length of the symbolic file name
achSymbolicName	ACH	wSizeSymbolicName	Character array including the symbolic file name • This information can be found under C2130 after data have been downloaded.
wCopyToRam	WORD	2	Selection whether data will be automatically copied into the application RAM of the PLC after downloading. • Maximum data length = 128 kbytes (RAM block 1 & 2)
			0: Data are not copied into the application RAM.
			1: Data are copied into the application RAM.
			2 ... 65535: Reserved
dwReserved	DWORD	4	*Reserved for future extensions*
awSizeAddInfo	DWORD	190	

Interpretation of the header information: least significant byte first:





8.4 System POUs

System POUs are POUs of the “program” type which accept the features of a certain name, i.e. they are started by an event registered by the PLC.

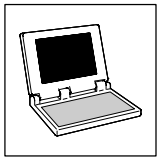
- The program includes max. 1000 instructions for system POUs.
- Unlike tasks or PLC_PRG, system POUs are not run-time monitored by a “watchdog”.
- The special POU names and the corresponding event for starting the POU are listed in the following table.

POU name	Event for starting POU	The POU starts if
PLC_TaskOverrun	Task overflow	...the task monitoring time will be exceeded.
PLC_RealError	Floating-point error	...a floating-point error occurs.
PLC_FailTripping	TRIP	...a trip is set.
PLC_WarningTripping	Warning	...a warning is activated.
PLC_MessageTripping	Message	...a message is activated. [1]
PLC_FailQspTripping	FAIL-QSP	...a FAIL-QSP is activated. [1]
PLC_CANError	CAN bus error	...a CAN bus error occurs (e. g. BUS-OFF).
PLC_AIFError	AIF bus error	...an AIF bus error occurs.
PLC_Restart	Start	...the START command is activated after a STOP.
PLC_ColdStart	Cold start	...the START command is activated after a RESET/program download. [2]
PLC_Stop	PLC stop	...the STOP command is activated.

[1] not available for Drive PLC.
[2] The CAN/AIF bus is ready after this POU has been executed

Tip!

If you need a system POU for an event-controlled start, simply create a “program” POU and use the name of this POU as POU name related to the corresponding event.



8.5 System error messages

Overview of the error sources detected by the PLC and the corresponding reactions.

Tip!

- The current error number is also displayed in the variable *DCTRL_wFaultNumber* after the PLC has been started.
- The history buffer (C0168/x) stores system error messages with an offset which indicates the type of response:

Type of response	No. of system error message		
0 ... 4	x	x	x

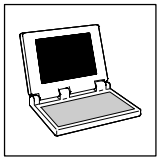
0 - TRIP 1 - Message 2 - Warning 3 - Off 4 - Error/QSP

Example: The value “2061” in code C0168/1 means:

“x061” ⇨ The current fault (subcode 1 of C0168) is a communication error between the AIF module and PLC (see error message no. 061)

“2xxx” ⇨ The response is a warning.

System error message				Possible settings/response • Factory setting ✓ Setting possible						Available in	
No.	Display	Source	Meaning	TRIP	Message	Warning	Error/QSP	Off	Code	Drive PLC	Servo PLC
011	OC1	MCTRL	Motor cable short circuit	•							✓
012	OC2	MCTRL	Motor cable earth fault	•							✓
015	OC5	MCTRL	I x t overload	•							✓
020	OU	MCTRL	Overvoltage in the DC bus		•						✓
030	LU	MCTRL	Undervoltage in the DC bus		•						✓
050	OH	MCTRL	Heatsink temperature higher than fixed limit temperature	•							✓
053	OH3	MCTRL	Motor temperature higher than fixed limit temperature	•				✓	C0583		✓
054	OH4	MCTRL	Heatsink temperature higher than variable limit temperature (C0122)			•		✓	C0582		✓
057	OH7	MCTRL	Motor temperature higher than variable limit temperature (C0121)			•		✓	C0584		✓
058	OH8	MCTRL	Motor temperature via inputs T1/T2 too high	✓		✓		•	C0585		✓
061	CE0	AIF	Communication error in AIF module ⇔PLC	✓		✓		•	C0126	✓	✓
			CAN communication error:								
062	CE1	CAN1	CAN1_IN (monitoring time can be set under C0357/1)	✓		✓		•	C0591	✓	✓
063	CE2	CAN2	CAN2_IN (monitoring time can be set under C0357/2)						C0592		
064	CE3	CAN3	CAN3_IN (monitoring time can be set under C0357/3)						C0593		
065	CE4	CAN	CAN BUS-OFF status (too many faulty telegrams received)	✓		✓		•	C0595	✓	✓
066	CE5	CAN	CAN timeout (gateway function)	✓		✓		•	C0603	✓	✓
070	U15	internal	Undervoltage of internal 15 V voltage supply	•						✓	✓
071	CCr	internal	Internal error	•						✓	✓
072	PR1	internal	Check sum error in parameter set 1	•						✓	✓
074	PEr	internal	Program error	•						✓	✓
075	PR0	internal	General error in the parameter sets	•						✓	✓
079	PI	internal	Error during parameter initialization	•						✓	✓
080	PR6	internal	Too many user codes	•						✓	✓



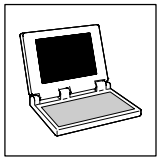
Drive PLC

Appendix

System error messages

System error message				Possible settings/response • Factory setting ✓ Setting possible						Available in	
No.	Display	Source	Meaning	TRIP	Message	Warning	Error/QSP	Off	Code	Drive PLC	Servo PLC
082	Sd2	MCTRL	Resolver error	•		✓		✓	C0586		✓
083	Sd3	MCTRL	Encoder error at X9 PIN 8	✓		✓		•	C0587		✓
085	Sd5	MCTRL	Enocder error at X6 PIN 1 and 2 (C0034 = 1)	✓		✓		•	C0598		✓
086	Sd6	MCTRL	Sensor error: motor temperature (X7 or X8)	•		✓		✓	C0594		✓
087	Sd7	MCTRL	Absolute-value encoder error at X8	✓				•	C0025		✓
088	Sd8	MCTRL	SinCos encoder error	✓				•	C0580		✓
091	EEr	FWM	External monitoring activated via DCTRL	•	✓	✓	✓	✓	C0581		✓
105	H05	internal	Internal error (memory)	•						✓	✓
107	H07	internal	Internal error (power stage)	•							✓
108	H08	internal	Extension board not connected correctly or not supported by program	•						✓	
110	H10	FWM	Sensor error: heatsink temperature	•				✓	C0588		✓
111	H11	FWM	Sensor error: inside temperature	•				✓			✓
			Communication error FIF-CAN:								
122	CE11	FIF-CAN1	FIF-CAN1_IN (monitoring time can be set under C2457/1)	✓		✓		•	C0591	✓	
123	CE12	FIF-CAN2	FIF-CAN2_IN (monitoring time can be set under C2457/2)						C0592		
124	CE13	FIF-CAN3	FIF-CAN3_IN (monitoring time can be set under C2457/3)						C0593		
125	CE14	FIF CAN	BUS-OFF status FIF-CAN (too many faulty telegrams received)	✓		✓		•	C0595	✓	
190	nErr	MCTRL	Speed beyond tolerance margin (C0576)	✓	✓	✓	✓	•	C0579		✓
200	NMAX	MCTRL	Maximum speed exceeded (C0596)	•							✓
			Time-out (see task configuration):								
201	overrun Task1	internal	Task with Id.2	•		✓	✓		[1]	✓	✓
202	overrun Task2		Task with Id.3								
203	overrun Task3		Task with Id.4								
204	overrun Task4		Task with Id.5								
205	overrun Task5		Task with Id.6								
206	overrun Task6		Task with Id.7								
207	overrun Task7		Task with Id.8								
208	overrun Task8		Task with Id.9								
			Floating-point error (REAL) in:								
209	float Sys-T	internal	System task	•		✓	✓		[1]	✓	✓
210	float Cycl.-T		Cyclic task (PLC_PRG)								
211	float T Id2		Task with Id.2								
212	float T Id3		Task with Id.3								
213	float T Id4		Task with Id.4								
214	float T Id5		Task with Id.5								
215	float T Id6		Task with Id.6								
216	float T Id7		Task with Id.7								
217	float T Id8		Task with Id.8								
218	float T Id9		Task with Id.9								

[1] In DDS adjustable under **Project→Exceptional response**



system error message				Possible settings/response • Factory setting ✓ Setting possible						Available in	
No.	Display	Source	Meaning	TRIP	Message	Warning	Error/QSP	Off	Code	Drive PLC	Servo PLC
219	overrun Cycl.-T	internal	Time-out in cyclic task (PLC_PRG)	•		✓	✓		1)	✓	✓
220	NoT-FktCredit	internal	Not enough technology units available in the PLC	•						✓	✓
230	No program	internal	No PLC program loaded in PLC	•						✓	✓
231	Unallowed Lib	internal	You have called the library function in the PLC program. This function is not supported.	•						✓	✓
232	NoCamData	internal	Movement profiles (cam data) are not available	•						✓	✓
			Free CAN objects								
240	ovrTransQueue	Free CAN obj.	Overflow of send order memory	•	✓	✓	✓2)	✓	C0608	✓	✓
241	ovr Receive		Too many receive telegrams	•			✓2)		C0609		✓
			Application memory (FLASH):								
250	2.Flash Err	internal	Access not possible (FLASH memory defective or not available)	•							✓
251	AddData CsErr	internal	Check sum error occurred when loading data into the FLASH memory								
252	AddData DlErr	internal	Error occurred when downloading data into the FLASH memory (e.g. time-out)								
260	Err NodeGuard	Node guarding	"Life guarding event": The PLC as a CAN slave has not received any "node guarding" telegram within the node lifetime of the CAN master.	•	✓	✓	✓2)	✓	C0384	✓	✓

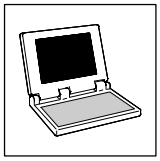
1) In DDS adjustable under **Project→Exceptional response**
2) Only with 9300 Servo PLC

8.5.1 System error message reset

Reaction	Measures to reset system error messages
TRIP/FAIL-QSP	A reset of TRIP/FAIL-QSP requires an acknowledgement. • If the TRIP source is still active, the current TRIP cannot be reset. A TRIP/FAIL-QSP can be acknowledged by: • Keypad ⇨ Pressing the STOP key before the RUN key (which enables the PLC again). • Code C0043 ⇨ Set C0043 = 0
Message	The message auotmatically disappears after the error has been eliminated, the drive automatically restarts!
Warning	The message automatically disappears after the error has been eliminated.

8.5.2 Cause and remedy

No.	Display	Meaning	Cause	Remedy
---	---	No error	-	-
011	OC1	Motor cable short circuit	Short circuit	Find reason for short circuit (check cable)
			Excessive capacitive charging current of the motor cable	Use shorter or lower capacity motor cable
012	OC2	Motor cable earth fault	One of the motor phases contacts earth	• Check motor • Check supply cables
			Excessive capacitive charging current of the motor cable	Use shorter or lower capacity motor cable
015	OC5	I x t overload	• Frequent and too long acceleration with overcurrent • Continuous overload with $I_{motor} > 1.05 \times I_{rx}$	Check drive dimensioning

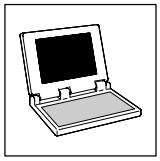


Drive PLC

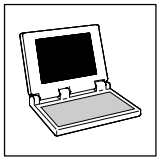
Appendix

System error messages

No.	Display	Meaning	Cause	Remedy
020	OU	Overvoltage in DC bus		
030	LU	Undervoltage in DC bus	DC bus voltage is lower than the value fixed under C0173	• Check mains voltage • Check supply cable
050	OH	Heatsink temperature higher than fixed limit temperature	Ambient temperature T_{amb} > 40 °C or 50 °C	• Allow controller to cool and ensure better ventilation • Check ambient temperature in the control cabinet
			Heat sink very dirty	Clean heat sink
			Incorrect mounting position	Change mounting position
053	OH3	Motor temperature higher than fixed limit temperature • For temperate detection via resolver or incremental encoder	• Motor too hot because of impermissibly high currents • Frequent and too long acceleration	Check drive dimensioning
			No PTC connected	Connect PTC or switch off monitoring (C0583=3)
054	OH4	Heat sink temperature higher than variable limit temperature (C0122)	Ambient temperature T_{amb} > 40 °C or 50 °C	• Allow controller to cool and ensure better ventilation • Check ambient temperature in the control cabinet
			Heat sink very dirty	Clean heat sink
			Incorrect mounting position	Change mounting position
			Value set under C0122 too low	Enter higher value under C0122
057	OH7	Motor temperature higher than variable limit temperature (C0121) • For temperate detection via resolver or incremental encoder	• Motor too hot because of impermissibly high currents • Frequent and too long acceleration	Check drive dimensioning
			No PTC connected	Connect PTC or switch off monitoring (C0584=3)
			Value set under C0121 too low	Enter higher value under C0121
058	OH8	Motor temperature via inputs T1/T2 too high	• Motor too hot because of impermissibly high currents • Frequent and too long acceleration	Check drive dimensioning
			Terminals T1, T2 are not assigned	Connect PTC or thermostat or switch off monitoring (C0585=3)
061	CE0	Communication error AIF module ⇔Basic unit	Interference during transfer of control commands via automation interface (AIF)	Plug in automation module firmly, bolt down, if necessary
062	CE1	CAN1_IN communication error	CANx_IN receives faulty data or communication interrupted	• Check cable at X4 • Check transmitter • If possible, increase monitoring time under C0357/x
063	CE2	CAN2_IN communication error		
064	CE3	CAN3_IN communication error		
065	CE4	CAN BUS-OFF status (too many faulty telegrams received)	PLC has received too many faulty telegrams send via system bus and has disconnected itself from the bus	• Check wiring • Check bus termination (if any) • Check screen contact of the cables • Check PE connection • Check bus load • Reduce baud rate (observe cable length)
066	CE5	CAN time-out (gateway function)	For tele-parameterization via system bus (C0370): • Slave does not respond • Communication monitoring time exceeded	• Check wiring of the system bus • Check system bus configuration
070	U15	Undervoltage of internal 15 V voltage supply		Check voltage supply for Drive PLC
071	CCR	Internal error	The program sequence of the processor was interfered.	• Shield control and motor cables as necessary • Check PE wiring and PE connections
072	PR1	Check sum error in parameter set 1 **CAUTION:** Default setting is loaded automatically!	• Error while reading a parameter set • Interruption of parameter set transfer via keypad (e.g. by disconnection of the keypad)	• Set the desired parameters and save them under C0003 • For PRO the supply voltage must be switched off additionally • Check use of pointers
			The saved parameters do not match the loaded software version	Before an error can be acknowledged, the parameter set must be manually saved under C0003



No.	Display	Meaning	Cause	Remedy
074	PEr	Program error	Error detected in program	• Check use of pointers • Send controller with parameter set and PLC program (on diskette) to Lenze
075	PR0	General error in parameter sets **CAUTION:** Default setting is loaded automatically!	See Cause & Remedy, no. 072 (PR1)	
079	PI	Error during parameter initialization	• Error detected during parameter transfer between two controllers • Parameter set does not match the PLC or controller (e.g. data transfer from a high-performance controller to a lower-performance controller)	• Correct the parameter set • Check code initialization values
080	PR6	Too many user codes		Reduce the number of user codes
082	Sd2	Resolver fault	Resolver cable interrupted	• Check resolver cable for open circuit • Check resolver • Switch off monitoring (C0586 = 3)
083	Sd3	Encoder fault at X9 PIN 8	Cable interrupted	Check cable for open circuit
			Input X9 PIN 8 not assigned	Assign input X9 PIN 8 with 5V or switch off monitoring (C0587 = 3)
085	Sd5	Encoder error at X6 PIN 1 and 2 (C0034 = 1)	Master current value at X6 PIN 1 and 2 < 2mA	• Check cable for open circuit • Check master current source
086	Sd6	Sensor fault: motor temperature (X7 or X8)	Encoder of the motor temperature detection at X7 or X8 indicates undefined values	• Check supply cable for firm connection • If necessary, switch off monitoring (C0594 = 3)
087	Sd7	Absolute-value encoder error at X8	Absolute value encoder with RS485 interface does not transmit data	• Check supply cable • Check encoder • Check supply voltage (C0421) • No Stegmann encoder connected
088	Sd8	SinCos encoder error	SinCos encoder not connected or does not any send plausible data	• Check supply cable • Check encoder • Check voltage supply • No Stegmann encoder connected
091	EEr	External monitoring activated via DCTRL	A digital input assigned to the TRIP-Set function has been activated.	Check external encoder
105	H05	Internal error (memory)		Contact Lenze
107	H07	Internal error (power stage)	During initialization of the controller, an incorrect power stage was detected	Contact Lenze
108	H08	Extension board error	Extension board not connected correctly	• Connect extension board properly • Connection plug EB ⇔Check PLC
			Extension board is not supported by PLC program	• Adapt PLC program to extension board • Use extension board which is supported by PLC
110	H10	Sensor fault Heat sink temperature	Sensor for heat sink temperature detection indicates indefinite values	Contact Lenze
111	H11	Sensor fault Inside temperature	Sensor for indoor temperature detection indicates indefinite values	Contact Lenze
122	CE11	Communication error FIF-CAN1_IN	FIF-CANx_IN receives faulty data or communication interrupted	• Check cable • Check transmitter • If possible, increase monitoring time under C2457/x
123	CE12	Communication error FIF-CAN2_IN		
124	CE13	Communication error FIF-CAN3_IN		
125	CE14	FIF-CAN BUS-OFF status (too many faulty telegrams received)	PLC has received too many faulty telegrams send via system bus and has disconnected itself from the bus	• Check wiring • Check bus termination (if any) • Check screen contact of the cables • Check PE connection • Check bus load • Reduce baud rate (observe cable length)
190	nErr	Speed beyond tolerance margin (C0576)	• Active load (e.g. for hoists) too high • Mechanical blockades on the load side	Check drive dimensioning

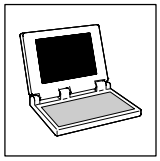


Drive PLC

Appendix

System error messages

No.	Display	Meaning	Cause	Remedy
200	NMAX	Maximum speed exceeded (C0596)	• Active load (e.g. for hoists) too high • Drive is not speed-controlled, torque excessively limited	• Check drive dimensioning • If necessary, increase torque limit
	Time-out (see task configuration):			
201	oveR-RUN TASK1	Task with Id.2	Task processing takes longer than the monitoring time set	• Set task run-time • If necessary, change monitoring time • Determine time-out reason by checking the task run-times at the task monitor. • Swap out time-critical program parts to a slower task
202	oveR-RUN TASK2	Task with Id.3		
203	oveR-RUN TASK3	Task with Id.4		
204	oveR-RUN TASK4	Task with Id.5		
205	oveR-RUN TASK5	Task with Id.6		
206	oveR-RUN TASK6	Task with Id.7		
207	oveR-RUN TASK7	Task with Id.8		
208	oveR-RUN TASK8	Task with Id.9		
	Floating point error (REAL) in:			
209	FLOAT SYS-T	System task	Error in real calculation (e.g. divided by 0)	• Check real calculation (program code) • The error is eliminated as in a task overflow
210	FLOAT CYCL.-T	Cyclic task (PLC_PRG)		
211	FLOATTASK1	Task with Id.2		
212	FLOATTASK2	Task with Id.3		
213	FLOATTASK3	Task with Id.4		
214	FLOATTASK4	Task with Id.5		
215	FLOATTASK5	Task with Id.6		
216	FLOATTASK6	Task with Id.7		
217	FLOATTASK7	Task with Id.8		
218	FLOATTASK8	Task with Id.9		
219	overrun Cycl.-T	Time-out in cyclic task (PLC_PRG)	Task processing takes longer than the monitoring time set	• Set task run-time • If necessary, change monitoring time • Determine time-out reason by checking the task run-times at the task monitor. • Swap out time-critical program parts in a slower task
220	NoT-FKT-CREDIT	Insufficient units on the target system	It was tried to load a program with technology functions to a PLC not providing corresponding units.	Use the T-variant of the PLC
230	No program	PLC program error	No PLC program loaded	Load PLC program to PLC
231	Unallowed Lib	Library function is not supported	You have called a library function in the PLC program. This function is not supported, e.g. because the required hardware is not available.	Remove library function or ensure that the required hardware devices are provided
232	NoCamData	Movement profiles (cam data) are not available	When functions of the LenzeCamControl.lib function library were called, the controller recognized that there are no movement profiles (cam data) loaded in the PLC memory	• Ensure that valid cam data were attached to the project via the DDS cam support • Re-load the PLC program (maybe the **Online →Reset (original)** command has been executed in DDS)
240	ovrTransQueue	Error “Free CAN objects”	Overflow of send order memory	• Reduce number of send orders • Prolong cycle time
241	ovr Receive	Error “Free CAN objects”	Too many receive telegrams	Reduce number of telegrams on bus
250	2.Flash Err	Fault during FLASH memory access	The PLC program tries to access a faulty FLASH memory or FLASH memory does not exist	Ensure that the PLC has a FLASH memory. If FLASH memory already exists, contact Lenze.
251	AddData CsErr	Fault during FLASH memory access	Check sum error occurred when loading data into the FLASH memory	Check the check sum of the file to be loaded and repeat data transfer
252	AddData DlErr	Fault during FLASH memory access	Error occurred when downloading data into the FLASH memory (e.g. time-out, transmission error, mains failure during data transfer)	Check/repeat data transfer
260	Err NodeGuard	”Life guarding event”	PLC has been configured as a CAN slave with ”node guarding” and has not received any ”node guarding” telegram within the node lifetime of the CAN master.	• Check system bus (CAN) and CAN configuration • Ensure that ”node guarding” has been activated in the CAN master • Adapt node lifetime to CAN master setting



8.5.3 Error analysis using the history buffer

The PLC history buffer consists of 8 memory loaction which store the following information about the active error and the 7 previous error, depending on their sequence of occurrence.

- No. of system error message (📖 8-7)
- Response to error (warning, message, TRIP, etc.) (📖 8-7)
- Time of error (ref. to when the PLC has been switched on, e.g. “1234567 s”)
- Occurrence of one error is sequence

The history buffer information are saved under codes C0168/x ... C0170/x:

C0168	C0169	C0170	Subcode	contains information about
No. of system error message and **response**	**Time** of occurrence	**Frequency of** the same error in sequence	1	Active error
			2	Last error
			3	Last but one error
			4	Last but two erro
			5	Last but three error
			6	Last but four error
			7	Last but five error
			8	Last but six error

Tip!

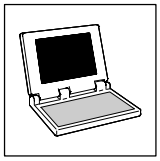
The history buffer works according to the principles of a shift register.

If the active error is no longer active or has been acknowledged by a TRIP RESET all history buffer information are automatically shifted up one subcode .

- Information about the previous error is now under subcode 2.
- Information about the last but six error are deleted from the history buffer and cannot be called up any more.

Note!

- Several errors causing different responses at the same time:
  – The hisory buffer only holds the error with the response of highest priority (priority = TRIP → message → FAIL-QSP → warning).
- Several errors causing the same response at the same time (e. g. 2 messages):
  – The history buffer only holds the first error.
- The same error again in sequence:
  – The history buffer only holds the time of the last occurrence.



TRIP reset

The active error can be reset under C0043.

Code	LCD	Possible settings		Info
		Lenze	Selection	
C0043	Trip reset	0		TRIP reset
			0 TRIP reset (reset of current TRIP)	
			1 TRIP active	

Delete entries from history buffer

The entries in the history buffer can be deleted under C0167.

- This function is only applicable if the display does not indicate any errors.

Code	LCD	Possible settings		Info
		Lenze	Selection	
C0167	Reset failmem	0		Reset history buffer
			0 No function	
			1 Delete entries from history buffer	

8.5.4 Fault analysis via the LED of the PLC

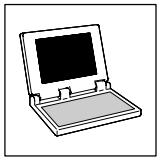
Two LEDs at the front of the PLC indicate the controller status:

LED green	LED red	Controller status	Check test
--	■	Fault: TRIP	C0183, C0168/1
★★	☐	PLC program not loaded	
☐	☐	PLC program loaded	
■	☐	PLC program running	
★	☐	PLC program loaded but stopped	
■ an ☐ aus ★ blinkend (1 s-Takt) ★★ blinkend (0.5 s-Takt) -- Zustand ohne Bedeutung			

8.5.5 Error analysis via the 9371BB keypad

Displayed status messages indicate the controller status:

Display	Controller status	Check test
RDY	PLC ready for operation	C0183, C0168/1
Fail	Error indicated by TRIP, message or warning	

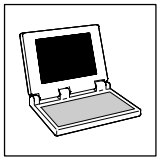


8.6 Code table

How to read the code table:

Column	Abbreviation	Meaning
Code	C0168	Code C0168
	1	Subcode 1 of code C0168
	2	Subcode 2 of code C0168
	…	…
	8	Subcode 8 of code C0168
	[C0156]	Parameter value of code can only be changed when controller is inhibited.
LCD		Keypad LCD
Lenze		Factory setting of the code
	Disp	Display code (can only be displayed)
	*	The column "Info" contains more information
Selection	1 {1 %} 99	Minimum value {smallest step/unit} maximum value
Info		Additional information on code

Code	LCD	Possible settings		Info
		Lenze	Selection	
[C0002]	Par load	0		Load parameter set • Only possible when PLC has been stopped. • Parameter set 1 is automatically loaded after power-on.
			0 Load factory setting into RAM 1 Load parameter set 1 into the RAM	
C0003	Par save	0		Save parameter set • Also possible when PLC is in operation.
			0 Saving completed 1 Non-volatile saving of parameter set 1	
C0004	Op display	372		Keypad operating display • Keypad will indicate selected code in the operating level if no other status indications of C0183 are active.
			All available codes	
C0009	LECOM address	1		LECOM controller address (bus device number for operation via interface) • 10, 20, ..., 90 reserved for broadcast to device groups for RS232, RS485, fibre optics.
			1 {1} 99	
C0011	Nmax	3000		Reference code for speed scaling
			500 {1 rpm} 16000	
C0043	Trip reset	0		Trip reset
			0 Trip reset (current TRIP reset) 1 Error occurred (TRIP is active)	
C0067	Act trip	Disp		Current error message
C0093	PLC ident	Disp		Controller identification
C0094	Password	0		Keypad password protection • Parameter password protection for the keypad • When the password is activated, only user-menu codes can be accessed. • For extended password protection please see C0096.
			0 {1} 9999 0 = No password protection	

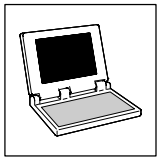


Drive PLC

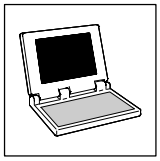
Appendix

Code table

Code	LCD	Possible settings		Info
		Lenze	Selection	
C0096	AIF/CAN prot.		0 No password protection 1 Read protection 2 Write protection 3 Read/write protection	AIF-/CAN password protection • Extended password protection for bus systems when password is activated (C0094). • All codes in the user menu can be accessed.
1		0		AIF password protection
2		0		CAN password protection
C0099	S/W version	Disp		Software version of operating system
C0125	Baud rate	0	0 9600 baud 1 4800 baud 2 2400 baud 3 1200 baud 4 19200 baud	LECOM baud rate for accessory module 2102
C0126	MONIT CE0	3	0 TRIP 2 Warning 3 Off	Configuration for communication error with automation interface (CE0)
C0135	System Var	0		Internal code • FCODE_bC135Bit0_b ... FCODE_bC135Bit15_b
C0150	Status word	Disp	0 {1} 65535	Status word hexadecimal for network via automation interface (AIF) • FCODE_bC150Bit0_b ... FCODE_bC150Bit15_b
C0161	Act trip	Disp		Current error message (see C0168/1)
C0167	Reset failmem	0	0 No function 1 Delete all entries in history buffer	History buffer reset
C0168	Fail number	Disp	All error messages	History buffer: Error messages • List with error messages in order of occurrence.
1				Current error message
2				Last error message
3				Last but one error message
4				Last but two error message
5				Last but three error message
j6				Last but four error message
7				Last but five error message
8				Last but six error message



Code	LCD	Possible settings				Info
		Lenze	Selection			
C0169	Failtime	Disp				History buffer: Power-up time • List with the power-up times until an error message occurs in C0168/x. • Refers to the elapsed time meter (C0179)
			0	{1 sec}	65535	
1						Current error message
2						Last error message
3						Last but one error message
4						Last but two error message
5						Last but three error message
j6						Last but four error message
7						Last but five error message
8						Last but six error message
C0170	Counter	Disp				History buffer: Frequency of error messages • List which indicates how many times an error occurred in C0168/x.
			0	{1}	65535	
1						Current error message
2						Last error message
3						Last but one error message
4						Last but two error message
5						Last but three error message
j6						Last but four error message
7						Last but five error message
8						Last but six error message
C0179	Mains timer	Disp				Elapsed time meter • Power-up time
			0	{1 sec}	4294967295	
C0183	Diagnostics	Disp				Drive diagnostics • Indicates error and status information. • If several error or status information occur at the same time, the information with the lowest number will be displayed
			0	OK		No fault
			101	Initialization		Initialisation phase
			102	TRIP/error		TRIP active
			103	Emergency stop		Emergency stop was released
			104	IMP message		Message active
			105	Power off		
			111	BSP C135		Operation inhibit C135
			112	BSP AIF		Operation inhibit AIF
			113	BSP CAN		Operation inhibit CAN
			121	RSP terminal 28		Controller inhibited via X5/28
			122	RSP internal 1		DCTRL-CINH1
			123	RSP internal 2		DCTRL-CINH2
			124	RSP C135/STOP		STOP key of 9371BB
			125	RSP AIF		Controller inhibited via AIF
			126	RSP CAN		Controller inhibited via CAN
			141	Power-up inhibit		Restart protection active
			142	IMP inhibit		Power outputs with high resistance
			151	QSP external terminal		QSP via MCTRL-QSP
			152	QSP C135/STOP		QSP via STOP key
			153	QSP AIF		QSP via AIF
			154	QSP CAN		QSP via CAN
			250	Warning		Warning active (C0168)
C0199	BuildNumber	Disp				BS software creation number
C0200	S/W Id	Disp				BS software product code
C0201	S/W date	Disp				BS software creation date

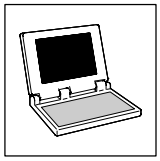


Drive PLC

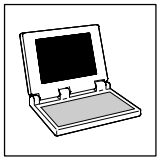
Appendix

Code table

Code	LCD	Possible settings		Info
		Lenze	Selection	
C0202	Internal ID	Disp		Internal identification
1				EKZ1
2				EKZ2
3				EKZ3
4				EKZ4
C0203	Comm.-No.	Disp		Commission number
C0204	Serial-No.	Disp		Serial number
C0205	Target Id	Disp		Identification number of the PLC
C0206	Product date	Disp		Production date
C0207	DL info 1	Disp		Download info 1
C0208	DL info 2	Disp		Download info 2
C0209	DL info 3	Disp		Download info 3
C0210	EB-IDENT	Disp		Extension board identification
C0300				Internal error diagnostics
C0301				Internal error diagnostics
C0350	CAN address	1		System bus: Node address
			1 {1} 63	
C0351	CAN baud rate	0		System bus: Baud rate
			0 500 kbit/s 1 250 kbit/s 2 125 kbit/s 3 50 kbit/s 4 1000 kbit/s 5 20 kbit/sec	
C0352	CAN mst	0		System bus: Master/slave configuration of the PLC • With selection 1 or 2, the PLC sends a system bus boot-up and is thus “quasi” master. • Additional information about the ”heartbeat” and ”node guarding” function can be found in the Manual for the function library **LenzeCanDSxDrv.lib**.
			0 Slave (boot-up not active) 1 Master (boot-up active) 2 Master with Node Guarding (SyncReceived no longer possible) 3 Slave and ”heartbeat” producer 4 Slave with Node Guarding	
C0353	CAN addr sel			System bus: Source for PDO identifiers
			0 Identifier assignment under C0350 + basic identifier 1 Identifier assignment under C0354/x	
1		0		CAN1_IN/OUT
2		0		CAN2_IN/OUT
3		0		CAN3_IN/OUT
C0354	CAN addr			System bus: Definition of individual PDO identifiers • Value to be entered = Identifier - 384
			1 {1} 512	
1		129		CAN1_IN
2		1		CAN1_OUT
3		257		CAN2_IN
4		258		CAN2_OUT
5		385		CAN3_IN
j6		386		CAN3_OUT



Code	LCD	Possible settings				Info
		Lenze	Selection			
C0355	CAN Id	Disp				System bus: PDO identifiers
			385	{1}	896	
1						CAN1_IN
2						CAN1_OUT
3						CAN2_IN
4						CAN2_OUT
5						CAN3_IN
j6						CAN3_OUT
C0356	CAN boot up					System bus: Time settings
1		3000	0	{1 msec}	65000	Delay time after power-on for initialisation through the “quasi” master.
2		0	0 0 = Event-controlled transmission	{1}	65000	Task time factor for sending the CAN2_OUT process data object.
3		0				Task time factor for sending the CAN3_OUT process data object.
4		20	0	{1 msec}	65000	Delay for sending the process data object
C0357	CE monit time					System bus: Monitoring time for process data input objects
			0	{1 msec}	65000	
1		3000				CE1monit time
2		3000				CE2monit time
3		3000				CE3monit time
C0358	Reset node	0				System bus: Reset node
			0 No function 1 CAN reset node			
C0359	CAN state	Disp				System bus: Status
			0 Operational 1 Pre-operational 2 Warning 3 Bus off 4 Stopped			
C0360	CAN message	Disp				System bus: Telegram counter (number of telegrams) • For values > 65535 counting restarts with 0
			0	{1 msec}	65535	
1						All sent (without free CAN objects)
2						All received (without free CAN objects)
3						Sent to CAN1_OUT
4						Sent to CAN2_OUT
5						Sent to CAN3_OUT
j6						Sent to parameter channel 1
7						Sent to parameter channel 2
8						Received by CAN1_IN
9						Received by CAN2_IN
10						Received by CAN3_IN
11						Received by parameter channel 1
12						Received by parameter channel 2

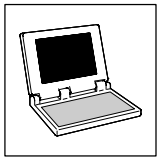


Drive PLC

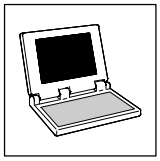
Appendix

Code table

Code	LCD	Possible settings		Info
		Lenze	Selection	
C0361	Load IN/OUT	Disp		System bus: Bus load • Trouble-free operation demands that the total bus load (all connected devices) does not exceed 80 %.
			0 {1 %} 100	
1				All sent (without free CAN objects)
2				All received (without free CAN objects)
3				Sent to CAN1_OUT
4				Sent to CAN2_OUT
5				Sent to CAN3_OUT
j6				Sent to parameter channel 1
7				Sent to parameter channel 2
8				Received by CAN1_IN
9				Received by CAN2_IN
10				Received by CAN3_IN
11				Received by parameter channel 1
12				Received by parameter channel 2
C0362	Sync cycle	Disp		System bus: Time between two sync telegrams
			1 {1 msec} 30	
C0363	Sync corr	1		System bus: Sync correction increments
			1 0.4 µs/ms 2 0.8 µs/ms 3 1.2 µs/ms 4 1.6 µs/ms 5 2.0 µs/ms	
C0365	DIS:CAN active	Disp		Input signal CAN active
			0 CAN not active 1 CAN active	
C0366	Sync response	1		System bus: Sync response
			0 No response 1 Response to Sync	No response PLC responds to a sync telegram by sending a CAN1_OUT object
C0367	Sync Rx Id	128		System bus: Sync Rx identifier Receive identifier of the sync telegram.
			1 {1} 256	
C0368	Sync Tx Id	128		System bus: Sync Tx identifier Send identifier of the sync telegram.
			1 {1} 256	
C0369	Sync Tx Time	0		System bus: CAN sync send telegram cycle • A sync telegram with the identifier of C0368 is sent with the set cycle time.
			0 {1} 65000 0 = off	
C0370	Gateway addr.	0		System bus: Activate remote parameter setting • When selecting a setting ≠ 0 all code read/write accesses will be redirected to the system bus device with the corresponding node address. • The codes are accessed via the SDO1 parameter channel of the target device.
			0 {1} 63 0 = Remote parameter setting deactivated	
C0372	FIF module Id	Disp		FIF module HW identification
			0 No module 1 Standard-I/O 2 System bus (CAN) j6 Intelligent module 10 No valid identification	



Code	LCD	Possible settings				Info
		Lenze	Selection			
C0373	EB module Id	Disp				Extension board HW identification
			0	No module		
			1	Standard module (Extension board 1)		
			2	CAN module		
			3	Intelligent module		
C0381	HeartProTime	0				System bus: Heartbeat (slave): HeartbeatProducerTime Time interval for sending the Heartbeat message. • Only relevant for setting C0352 = 3. • Additional information about the "heartbeat" function can be found in the Manual for the function library **LenzeCanDSxDrv.lib**.
			0	{1 msec}	65535	
C0382	GuardTime	0				System bus: Node Guarding (slave): NodeGuardTime Time interval for the status check of the master. • Only relevant for setting C0352 = 4. • Additional information about the "node guarding" function can be found in the Manual for the function library **LenzeCanDSxDrv.lib**.
			0	{1 msec}	65535	
C0383	LifeTimeFact.	0				System bus: Node Guarding (slave): NodeLifeTime factor Factor for the NodeLifeTime monitoring time: NodeLifeTime = C0383 x C0382 (NodeGuardTime) • Only relevant for setting C0352 = 4. • Additional information about the "node guarding" function can be found in the Manual for the function library **LenzeCanDSxDrv.lib**.
			0	{1}	255	
C0384	Err NodeGuard	3				System bus: Node guarding (slave): Reaction when a NodeGuard event occurs. • Only relevant for setting C0352 = 4. • Additional information about the "node guarding" function can be found in the Manual for the function library **LenzeCanDSxDrv.lib**.
			0	TRIP		
			1	Message		
			2	Warning		
			3	Off		
			4	Fail QSP		
C0400	DIS: OUT	Disp				Analog inputs
			-199.99	{0.01 %}	199.99	
1						AIN1_nIn_a
2						AIN2_nIn_a
3						AIN3_nIn_a
C0405	DIS: OUT	Disp				Extension board 3: Analog inputs
			-199.99	{0.01 %}	199.99	
1						AIN4_nIn_a
2						AIN5_nIn_a
C0425	DFIN const	3				Extension board 3: DFIN_IO Increments of the encoder input
			0	256 inc/rev		
			1	512 inc/rev		
			2	1024 inc/rev		
			3	2048 inc/rev		
			4	4096 inc/rev		
			5	8192 inc/rev		
			j6	16384 inc/rev		
C0426	DIS: OUT	Disp				Extension board 3: DFIN_IO DFIN_nIn_v
			-32767	{1 rpm}	32767	

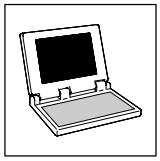


Drive PLC

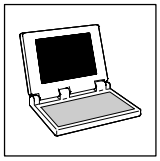
Appendix

Code table

Code	LCD	Possible settings		Info
		Lenze	Selection	
C0428	DFIN TP sel.	0	0 Touch probe via zero pulse 1 Touch probe via digital input X3/I1	Extension board 3: DFIN_IO Touch probe selection
C0429	TP delay	0	-32767 {1 inc} 32767	Extension board 3: DFIN_IO Touch probe delay • Compensation of delay times of TP signal source at X3/I1
C0431	DFIN TP EDGE	0	0 Activation with positive signal 1 Activation with negative signal	Extension board 3: DFIN_IO Touch probe activation • For touch probe via digital input X3/I1 (C0428 = 1)
C0434	AOUT1	Disp	-199.99 {0.01 %} 199.99	Analog output AOUT1_nOut_a
C0443	DIGIN 8-1	Disp	0 {dec} 255 Decimal value is bit-coded: Bit 0 DIGIN_bIn1_b Bit 1 DIGIN_bIn2_b Bit 2 DIGIN_bIn3_b Bit 3 DIGIN_bIn4_b Bit 4 DIGIN_bIn5_b Bit 5 DIGIN_bIn6_b Bit 6 DIGIN_bIn7_b Bit 7 DIGIN_bIn8_b	Digital inputs I1 I2 I3 I4 I5 I6 I7 I8
C0444	DIGOUT 4-1	Disp	0 {dec} 255 Decimal value is bit-coded: Bit 0 DIGOUT_bOut1_b Bit 1 DIGOUT_bOut2_b Bit 2 DIGOUT_bOut3_b Bit 3 DIGOUT_bOut4_b Bit 4 Not assigned ... ... Bit 7 Not assigned	Digital outputs O1 O2 O3 O4
C0446	22-17 DI 16-9	Disp	0 {dec} 255 Decimal value is bit-coded:	Extension board 1/2: Digital inputs • I15 ... I22 only with extension board 2
1			Bit 0 DIGIN_bIn9_b Bit 1 DIGIN_bIn10_b Bit 2 DIGIN_bIn11_b Bit 3 DIGIN_bIn12_b Bit 4 DIGIN_bIn13_b Bit 5 DIGIN_bIn14_b Bit 6 DIGIN_bIn15_b Bit 7 DIGIN_bIn16_b	I9 I10 I11 I12 I13 I14 I15 I16
2			Bit 0 DIGIN_bIn17_b Bit 1 DIGIN_bIn18_b Bit 2 DIGIN_bIn19_b Bit 3 DIGIN_bIn20_b Bit 4 DIGIN_bIn21_b Bit 5 DIGIN_bIn22_b Bit 6 Not assigned Bit 7 Not assigned	I17 I18 I19 I20 I21 I22



Code	LCD	Possible settings		Info
		Lenze	Selection	
C0448	DIGOUT 10-5	Disp		Extension board 1: Digital outputs
			0 {dec} 255	
			Decimal value is bit-coded:	
			Bit 0 DIGOUT_bOut5_b	05
			Bit 1 DIGOUT_bOut6_b	06
			Bit 2 DIGOUT_bOut7_b	07
			Bit 3 DIGOUT_bOut8_b	08
			Bit 4 DIGOUT_bOut9_b	09
			Bit 5 DIGOUT_bOut10_b	010
			Bit 6 Not assigned	
			Bit 7 Not assigned	
C0470	FCODE 8bit			Freely configurable codes (digital signals) • C0470 has the same memory address as code C0471.
			0 {1} 255	
1		0		C0470/1 = C0471, bit 0 ... 7
2		0		C0470/2 = C0471, bit 8 ... 15
3		0		C0470/3 = C0471, bit 16 ... 23
4		0		C0470/4 = C0471, bit 24 ... 31
C0471	FCODE 32bit	0		Freely configurable codes (digital signals) • Variables assigned via FCODE_FreeCodes: FCODE_bC471Bit0_b ... FCODE_bC471Bit31_b • C0471 has the same memory address as code C0470.
			0 {1} 4294967296	
C0472	FCODE analog			Freely configurable codes (relative analog signals)
			-199.99 {0.01 %} 199.99	
1		0		FCODE_bC472_1_a
...		...		...
20		0		FCODE_bC472_20_a
C0473	FCODE abs			Freely configurable codes (absolute analog signals)
			-32767 {1} 32767	
1		1		FCODE_bC473_1_a
2		1		FCODE_bC473_2_a
3		0		FCODE_bC473_3_a
...		...		...
10		0		FCODE_bC473_10_a
C0474	FCODE Lint			Freely configurable codes (angle/position signals)
			-2147483648 {1} 2147483647	
1		0		FCODE_dnC474_1_p
...		...		...
5		0		FCODE_dnC474_5_p
C0475	FCODE int			Freely configurable codes (speed signals)
			-16000 {1 rpm} 16000	
1		0		FCODE_nC475_1_v
2		0		FCODE_nC475_2_v
C0477	DIGIN FIF	Disp		FIF module Standard-I/O: Digital inputs
			0 {dec} 255	
			Decimal value is bit-coded:	
			Bit 0 DIGIN_bIn1_FIFSTDIO_b	E1
			Bit 1 DIGIN_bIn2_FIFSTDIO_b	E2
			Bit 2 DIGIN_bIn3_FIFSTDIO_b	E3
			Bit 3 DIGIN_bIn4_FIFSTDIO_b	E4
			Bit 4 DIGIN_bIn28_FIFSTDIO_b	28
			Bit 5 Not assigned	
			Bit 6 Not assigned	
			Bit 7 Not assigned	

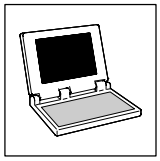


Drive PLC

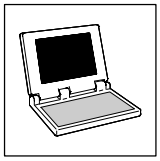
Appendix

Code table

Code	LCD	Possible settings		Info
		Lenze	Selection	
C0479	DIGOUT FIF	Disp		FIF module Standard-I/O: Digital output
			0 {dec} 255 Decimal value is bit-coded: Bit 0 DIGOUT_bOut1_FIFSTDIO_b Bit 1 Not assigned ... ... Bit 7 Not assigned	A1
C0481	AIN FIF	Disp		FIF module Standard-I/O: Analog input AIN1_nIn_FIFSTDIO_a
			-199.99 {0.01 %} 199.99	
C0484	AOUT FIF	Disp		FIF module Standard-I/O: Analog output AOUT1_nOut_FIFSTDIO_a
			-199.99 {0.01 %} 199.99	
C0504				PLC memory: RAM access via codes • More detailed information can be found in chapter 8.6.2, "RAM access via codes". 📖 8-36
C0505				
C0506				
C0507				
C0508				
C0509				
C0517	User menu			User menu with up to 32 entries • The numbers of desired codes are entered under the subcodes. • The data is input in xxx.yy format – xxx: Code number – yy: Subcode for code • There is no check whether the entered code exists.
			0.00 {0.01} 7999.00	
1		372.00	C0372/0 FIF module Id	
2		373.00	C0373/0 EB module Id	
3		2113.00	C2113/0 PLC Prog Name	
4		2108.00	C2108/0 PLC run/stop	
5		0.00	Not assigned	
j6		400.01	C0400/1 DIS: OUT (AIN1_nIn_a)	
7		400.02	C0400/2 DIS: OUT (AIN2_nIn_a)	
8		400.03	C0400/3 DIS: OUT (AIN3_nIn_a)	
9		434.00	C0434/0 DIS: IN (AOUT1_nOut_a)	
10		443.00	C0443/0 DIGIN 8-1	
11		444.00	C0444/0 DIGOUT 4-1	
12		0.00	Not assigned	
13		0.00	Not assigned	
14		2106.00	C2106/0 In preparation	
15		0.00	Not assigned	
...		...	...	
18		0.00	Not assigned	
19		2100.00	C2100/0 Time slice	
20		2102.00	C2102/0 Task switch	
21		2104.00	C2104/0 PLC Autorun	
22		0.00	Not assigned	
23		2108.00	C2108/0 PLC run/stop	
24		2110.00	C2110/0	
25		0.00	Not assigned	
...		...	...	
31		0.00	Not assigned	
32		94.00	C0094/0 Password	



Code	LCD	Possible settings		Info
		Lenze	Selection	
C0591	MONIT CE1	3	0 TRIP 2 Warning 3 Off	System bus: Monitoring configuration CAN1_IN communication error (CE1)
C0592	MONIT CE2	3	0 TRIP 2 Warning 3 Off	System bus: Monitoring configuration CAN2_IN communication error (CE2)
C0593	MONIT CE3	3	0 TRIP 2 Warning 3 Off	System bus: Monitoring configuration CAN3_IN communication error (CE3)
C0595	MONIT CE4	3	0 TRIP 2 Warning 3 Off	System bus: Monitoring configuration “BusOffState” (CE4)
C0603	MONIT CE5	3	0 TRIP 2 Warning 3 Off	System bus: Monitoring configuration Time-out when remote parameter setting activated (C0370)
C0608	ovr Tx-Queue	0	0 TRIP 1 Message 2 Warning 3 Off	System bus: Monitoring configuration Tx-Buffer (send memory overflow)
C0855	DIS: IN 15-0	Disp	0000 {hex} FFFF Hexadecimal value is bit-coded:	Automation interface (AIF): Process input words
1			Bit 00 AIF1_bInB0_b Bit 01 AIF1_bInB1_b ... Bit 15 AIF1_bInB15_b	AIF1_IN: Process input word 3
2			Bit 00 AIF1_bInB16_b Bit 01 AIF1_bInB17_b ... Bit 15 AIF1_bInB31_b	AIF1_IN: Process input word 4
C0856	DIS: IN.Wx	Disp	-32768 {1 %} 32767	Automation interface (AIF): Process input words
1				AIF1_IN: Process input word 2 (AIF1_nInW1_a)
2				AIF1_IN: Process input word 3 (AIF1_nInW2_a)
3				AIF1_IN: Process input word 4 (AIF1_nInW3_a)
C0858	DIS: OUT.Wx	Disp	-32768 {1 %} 32767	Automation interface (AIF): Process output words
1				AIF1_OUT: Process output word 2 (AIF1_nOutW1_a)
2				AIF1_OUT: Process output word 3 (AIF1_nOutW2_a)
3				AIF1_OUT: Process output word 4 (AIF1_nOutW3_a)

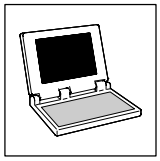


Drive PLC

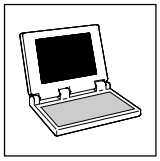
Appendix

Code table

Code	LCD	Possible settings		Info
		Lenze	Selection	
C0863	DIS: INx dig x	Disp		System bus: Process input words
			0000 {hex} FFFF Hexadecimal value is bit-coded:	
1			Bit 00 CAN1_bInB0_b Bit 01 CAN1_bInB1_b ... Bit 15 CAN1_bInB15_b	CAN1_IN: Process input word 1
2			Bit 00 CAN1_bInB16_b Bit 01 CAN1_bInB17_b ... Bit 15 CAN1_bInB31_b	CAN1_IN: Process input word 2
3			Bit 00 CAN2_bInB0_b Bit 01 CAN2_bInB1_b ... Bit 15 CAN2_bInB15_b	CAN2_IN: Process input word 1
4			Bit 00 CAN2_bInB16_b Bit 01 CAN2_bInB17_b ... Bit 15 CAN2_bInB31_b	CAN2_IN: Process input word 2
5			Bit 00 CAN3_bInB0_b Bit 01 CAN3_bInB1_b ... Bit 15 CAN3_bInB15_b	CAN3_IN: Process input word 1
j6			Bit 00 CAN3_bInB16_b Bit 01 CAN3_bInB17_b ... Bit 15 CAN3_bInB31_b	CAN3_IN: Process input word 2
C0866	DIS: INx.Wx	Disp		System bus: Process input words
			-32768 {1 %} 32767	
1				CAN1_nInW1_a
2				CAN1_nInW2_a
3				CAN1_nInW3_a
4				CAN2_nInW1_a
5				CAN2_nInW2_a
j6				CAN2_nInW3_a
7				CAN2_nInW4_a
8				CAN3_nInW1_a
9				CAN3_nInW2_a
10				CAN3_nInW3_a
11				CAN3_nInW4_a
C0868	DIS: OUTx.Wx	Disp		System bus: Process output words
			-32768 {1 %} 32767	
1				CAN1_nOutW1_a
2				CAN1_nOutW2_a
3				CAN1_nOutW3_a
4				CAN2_nOutW1_a
5				CAN2_nOutW2_a
j6				CAN2_nOutW3_a
7				CAN2_nOutW4_a
8				CAN3_nOutW1_a
9				CAN3_nOutW2_a
10				CAN3_nOutW3_a
11				CAN3_nOutW4_a



Code	LCD	Possible settings		Info
		Lenze	Selection	
C1120	Sync mode	0	0 Off 1 Synchronisation via system bus (CAN) 2 Synchronisation via terminal X3/I1	System bus: Sync source
C1121	Sync cycle	2	1 {1 msec} 13	System bus: Synchronisation cycle • Definition of the cycle time of the sync telegram/signal. • Parameterisation is only required for the slave!
C1122	Sync phase	0	0 {0.001 ms} 6.5	System bus: Synchronisation phase • Phase shift between the sync telegram/signal and the start of the internal control program.
C1123	Sync window	0	0 {0.001 ms} 6.5	System bus: Synchronisation window • If the sync telegram/signal sent from the master is within the "time window", CAN_bSyncInsideWindow_b = TRUE.
C1810	S/W Id keypad	Disp		Keypad identification
C1811	S/W date keypad	Disp		Keypad creation date
C2100	Time slice	13	j6 {1 msec} 26	Time slice for task switch between system task and cyclic task (PLC_PRG)
C2102	Task switch	0	Activation after: 0 Time slice 1 Time slice + end PLC_PRG 2 Time slice + end PLC_PRG + end system task	Configuration of task switch process
C2104	PLC Autorun	0	0 No auto start 1 Auto start after mains connection	Automatic start of the PLC program after power switch-on
C2106	Downl.protect	0	0 New download possible 1 No new download possible	Write protection for the PLC program
C2107	PwDownlProt.	0	0 {1} 4294967295	Password for download write protection (C2106)
C2108	PLC run/stop	0	0 No function 1 PLC run 2 PLC stop 3 PLC reset	PLC program start stop reset
C2110				Internal code
C2111	GDC Id	Disp		Compilation time/date of PLC program
C2113	PLC Prog Name	Disp		Name of the PLC program
C2115	T-Fct credit	Disp		Technology units
C2116	CreditPinCode	0	0 {1} 4294967295	PIN code for enabling technology units in case of servicing (please contact Lenze)
C2117	Full credit	Disp		Service code
C2118	ParWriteChan.	0	0 PDO channel (CAN1_IO ... CAN3_IO) 1 SDO2 channel	CAN object for L_ParRead and L_ParWrite

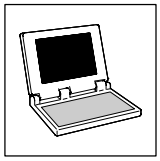


Drive PLC

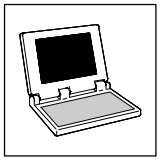
Appendix

Code table

Code	LCD	Possible settings		Info
		Lenze	Selection	
C2120	AIF: Control		0 No command 1 Read CAN code + new initialisation 2 Read XCAN code 10 Read XCAN C2356/1..4 11 Read XCAN C2357 12 Read XCAN C2375 13 Read XCAN C2376 ... C2378 14 Read XCAN C2382 15 Not assigned	AIF command
C2121	AIF: state	Disp	0 {dec} 255 Decimal value is bit-coded: Bit 0 XCAN1_IN monitoring time Bit 1 XCAN2_IN monitoring time Bit 2 XCAN3_IN monitoring time Bit 3 XCAN bus off Bit 4 XCAN operational Bit 5 XCAN pre-operational Bit 6 XCAN warning Bit 7 Internally assigned	AIF-CAN: Status • Detailed information can be found in the description for the corresponding fieldbus module.
C2130	FileNameAddDa	Disp	Symbolic data name	Information about the additional data that were transferred together with the PLC program to the PLC. • More detailed information can be found in chapter 8.3.2, "Downloading data". 📖 8-4
C2131	Type AddData	Disp	Data type	
C2132	VersionAddData	Disp	Data version	
C2133	TimeStamp	Disp	Time stamp of the data	
C2350	XCAN address	1	1 {1} 63	AIF-CAN: Node address
C2351	XCAN baud rate	0	0 500 kbit/s 1 250 kbit/s 2 125 kbit/s 3 50 kbit/s 4 1000 kbit/s 5 20 kbit/s j6 10 kbit/s	AIF-CAN: Baud rate
C2352	XCAN mst	0	0 Boot up not active 1 Boot up active	AIF-CAN: Set-up quasi master • Device sends system bus boot up and is thus the "quasi" master
C2353	XCAN addrselx		0 Identifier assignment via C2350 + basic identifier 1 Identifier assignment via C2354/x	AIF-CAN: Source for PDO identifiers
1		0		XCAN1_IN/OUT
2		0		XCAN2_IN/OUT
3		0		XCAN3_IN/OUT
C2354	XCAN sel.addr		0 {1} 1663	AIF-CAN: Definition of individual PDO identifiers • Value to be entered = Identifier - 384
1		129		XCAN1_IN
2		1		XCAN1_OUT
3		257		XCAN2_IN
4		258		XCAN2_OUT
5		385		XCAN3_IN
j6		386		XCAN3_OUT



Code	LCD	Possible settings				Info
		Lenze	Selection			
C2355	XCAN Id	Disp				AIF-CAN: PDO system bus identifiers
			384	{1}	2047	
1						XCAN1_IN
2						XCAN1_OUT
3						XCAN2_IN
4						XCAN2_OUT
5						XCAN3_IN
j6						XCAN3_OUT
C2356	CAN boot up					AIF-CAN: Time settings
			0	{1 msec}	65000	Delay time after power-on for initialisation by “quasi” master
1		3000				
			0 0 = Event-controlled transmission	{1 msec}	65000	Cycle time for transmitting the process data object
2		0				XCAN1_OUT
3		0				XCAN2_OUT
4		0				XCAN3_OUT
5		0				XCAN Sync Tx cycle time
C2357	CExmonit time					AIF-CAN: Monitoring time for process data input objects
			0	{1 msec}	65000	
1		3000				XCAN1_IN
2		3000				XCAN2_IN
3		3000				XCAN3_IN
4		1				Bus off
C2359	AIF HW Set.	Disp				AIF module DIP switch settings
			0	{1}	65535	
C2367	Sync Rx Id	128				AIF-CAN: Sync Rx identifier • Receive identifier for the sync telegram
			1	{1}	256	
C2368	Sync Tx Id	128				AIF-CAN: Sync Tx Identifier • Send identifier for the sync telegram
			1	{1}	256	

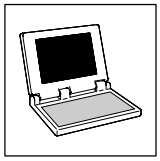


Drive PLC

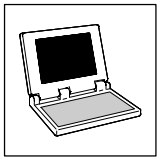
Appendix

Code table

Code	LCD	Possible settings		Info
		Lenze	Selection	
C2373	XSyncRate IN		1 {1} 240	AIF-CAN: Sync counter
1		1		XCAN1_IN
2		1		XCAN2_IN
3		1		XCAN3_IN
C2374	XSyncRate OUT		1 {1} 240	AIF-CAN: Sync counter
1		1		XCAN1_OUT
2		1		XCAN2_OUT
3		1		XCAN3_OUT
C2375	XCAN Tx mode		0 Sync with response 1 Sync without response 2 Event-controlled (with mask)/cyclically 3 Event-controlled (with mask) with cyclic overlay	AIF-CAN: TX mode for XCANx_OUT • Selection of cycle time under C2356
1		0		XCAN1_OUT
2		0		XCAN2_OUT
3		0		XCAN3_OUT
C2376	XCAN1 Mask		0 {hex} FFFF	AIF-CAN: XCAN1_OUT mask
1		FFFF		Mask for process output word 1
2		FFFF		Mask for process output word 2
3		FFFF		Mask for process output word 3
4		FFFF		Mask for process output word 4
C2377	XCAN2 Mask		0 {hex} FFFF	AIF-CAN: XCAN2_OUT mask
1		FFFF		Mask for process output word 1
2		FFFF		Mask for process output word 2
3		FFFF		Mask for process output word 3
4		FFFF		Mask for process output word 4
C2378	XCAN3 Mask		0 {hex} FFFF	AIF-CAN: XCAN3_OUT mask
1		FFFF		Mask for process output word 1
2		FFFF		Mask for process output word 2
3		FFFF		Mask for process output word 3
4		FFFF		Mask for process output word 4
C2382	XCAN Conf. CE		0 Off 1 No function 2 No function	AIF-CAN: Monitoring configuration
1		0		XCAN1_IN (no telegram received)
2		0		CAN2_IN (no telegram received)
3		0		XCAN3_IN (no telegram received)
4		0		Bus off
5		0		Life Guarding Event
C2450	CAN1 address	1	1 {1} 63	FIF-CAN: Node address



Code	LCD	Possible settings		Info
		Lenze	Selection	
C2451	CAN1 baud rate	0	0 500 kbit/s 1 250 kbit/s 2 125 kbit/s 3 50 kbit/s 4 1000 kbit/s	FIF-CAN: Baud rate
C2452	CAN1 mst	0	0 Boot up not active 1 Boot up active	FIF-CAN: Set-up quasi master • Device sends system bus boot up and is thus the "quasi" master
C2453	CAN addr sel		0 Identifier assignment under C2450 + basic identifier 1 Identifier assignment under C2454/x	FIF-CAN: Source for PDO identifiers
1		0		FIF-CAN1_IN/OUT
2		0		FIF-CAN2_IN/OUT
3		0		FIF-CAN3_IN/OUT
C2454	CAN addr		1 {1} 512	FIF-CAN: Definition of individual PDO identifiers
1		129		FIF-CAN1_IN
2		1		FIF-CAN1_OUT
3		257		FIF-CAN2_IN
4		258		FIF-CAN2_OUT
5		385		FIF-CAN3_IN
j6		386		FIF-CAN3_OUT
C2455	CAN Id	Disp	385 {1} 896	FIF-CAN: PDO identifiers
1				FIF-CAN1_IN
2				FIF-CAN1_OUT
3				FIF-CAN2_IN
4				FIF-CAN2_OUT
5				FIF-CAN3_IN
j6				FIF-CAN3_OUT
C2456	CAN1 boot up			FIF-CAN: Time settings
1		3000	0 {1 msec} 65000	Delay time after power-on for initialisation through the "quasi" master.
2		0	0 {1} 65000 0 = Event-controlled transmission	Task time factor for sending the FIF-CAN2_OUT process data object.
3		0		Task time factor for sending the FIF-CAN3_OUT process data object.
4		20	0 {1 msec} 65000	Delay for sending the process data object
C2457	CE monit time		0 {1 msec} 65000	FIF-CAN: Monitoring time for process data input objects
1		3000		CE11monit time
2		3000		CE12monit time
3		3000		CE13monit time
C2458	Reset node	0	0 No function 1 FIF-CAN reset node	FIF-CAN: Reset node
C2459	CAN1 state	Disp	0 Operational 1 Pre-operational 2 Warning 3 Bus off	FIF-CAN: Status

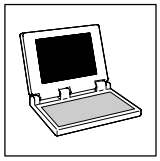


Drive PLC

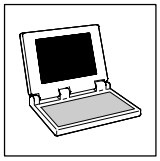
Appendix

Code table

Code	LCD	Possible settings		Info
		Lenze	Selection	
C2460	CAN message	Disp		FIF-CAN: Telegram counter (number of telegrams) • For values > 65535 counting restarts with 0
			0 {1 msec} 65535	
1				All sent
2				All received
3				Sent to FIF-CAN1_OUT
4				Sent to FIF-CAN2_OUT
5				Sent to FIF-CAN3_OUT
j6				Sent to parameter channel 1
7				Sent to parameter channel 2
8				Received by FIF-CAN1_IN
9				Received by FIF-CAN2_IN
10				Received by FIF-CAN3_IN
11				Received by parameter channel 1
12				Received by parameter channel 2
C2461	Load IN/OUT	Disp		FIF-CAN: Bus load • Trouble-free operation demands that the total bus load (all connected devices) does not exceed 80 %.
			0 {1 %} 100	
1				All sent
2				All received
3				Sent to FIF-CAN1_OUT
4				Sent to FIF-CAN2_OUT
5				Sent to FIF-CAN3_OUT
j6				Sent to parameter channel 1
7				Sent to parameter channel 2
8				Received by FIF-CAN1_IN
9				Received by FIF-CAN2_IN
10				Received by FIF-CAN3_IN
11				Received by parameter channel 1
12				Received by parameter channel 2
C2466	Sync response	1		FIF-CAN: Sync response
			0 No response 1 Response to Sync	No response The PLC reacts to a sync telegram with transmitting the FIF-CAN1_OUT object.
C2467	Sync Rx Id	128		FIF-CAN: Sync Rx identifier • Receive identifier of the sync telegram.
			1 {1} 256	
C2468	Sync Tx Id	128		FIF-CAN: Sync Tx identifier • Send identifier of the sync telegram.
			1 {1} 256	
C2469	Sync Tx Time	0		FIF-CAN: Sync send telegram cycle • A sync telegram with the identifier of C2468 is sent with the set cycle time.
			0 {1} 65000 0 = off	
C2470	ParWriteChan.	0		FIF-CAN object for L_ParRead and L_ParWrite
			0 FIF-PDO channel (FIF_CAN1_IO ... FIF_CAN3_IO) 1 FIF-SDO2 channel	
C2481	MONIT CE11	3		FIF-CAN: Monitoring configuration FIF-CAN1_IN communication error (CE11)
			0 TRIP 2 Warning 3 Off	



Code	LCD	Possible settings		Info
		Lenze	Selection	
C2482	MONIT CE12	3	0 TRIP 2 Warning 3 Off	FIF-CAN: Monitoring configuration FIF-CAN2_IN communication error (CE12)
C2483	MONIT CE13	3	0 TRIP 2 Warning 3 Off	FIF-CAN: Monitoring configuration FIF-CAN3_IN communication error (CE13)
C2484	MONIT CE14	3	0 TRIP 2 Warning 3 Off	FIF-CAN: Monitoring configuration “BusOffState” (CE14)
C2485	MONIT CE15	3	0 TRIP 3 Off	FIF-CAN: Monitoring configuration “Timeout” (CE15)
C2491	DIS: INx dig x	Disp	0000 {hex} FFFF Hexadecimal value is bit-coded:	FIF-CAN: Process input words
1			Bit 00 FIF_CAN1_bInB0_b Bit 01 FIF_CAN1_bInB1_b ... Bit 15 FIF_CAN1_bInB15_b	FIF-CAN1_IN: process input word 1
2			Bit 00 FIF_CAN1_bInB16_b Bit 01 FIF_CAN1_bInB17_b ... Bit 15 FIF_CAN1_bInB31_b	FIF-CAN1_IN: Process input word 2
3			Bit 00 FIF_CAN2_bInB0_b Bit 01 FIF_CAN2_bInB1_b ... Bit 15 FIF_CAN2_bInB15_b	FIF-CAN2_IN: Process input word 1
4			Bit 00 FIF_CAN2_bInB16_b Bit 01 FIF_CAN2_bInB17_b ... Bit 15 FIF_CAN2_bInB31_b	FIF-CAN2_IN: Process input word 2
5			Bit 00 FIF_CAN3_bInB0_b Bit 01 FIF_CAN3_bInB1_b ... Bit 15 FIF_CAN3_bInB15_b	FIF-CAN3_IN: Process input word 1
j6			Bit 00 FIF_CAN3_bInB16_b Bit 01 FIF_CAN3_bInB17_b ... Bit 15 FIF_CAN3_bInB31_b	FIF-CAN3_IN: Process input word 2

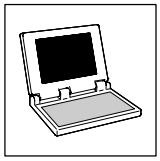


Drive PLC

Appendix

Code table

Code	LCD	Possible settings				Info
		Lenze	Selection			
C2492	DIS: INx.Wx	Disp				FIF-CAN: Process input words
			-32768	{1 %}	32767	
1						FIF_CAN1_nInW1_a
2						FIF_CAN1_nInW2_a
3						FIF_CAN1_nInW3_a
4						FIF_CAN2_nInW1_a
5						FIF_CAN2_nInW2_a
j6						FIF_CAN2_nInW3_a
7						FIF_CAN2_nInW4_a
8						FIF_CAN3_nInW1_a
9						FIF_CAN3_nInW2_a
10						FIF_CAN3_nInW3_a
11						FIF_CAN3_nInW4_a
C2493	DIS: OUTx.Wx	Disp				FIF-CAN: Process output words
			-32768	{1 %}	32767	
1						FIF_CAN1_nOutW1_a
2						FIF_CAN1_nOutW2_a
3						FIF_CAN1_nOutW3_a
4						FIF_CAN2_nOutW1_a
5						FIF_CAN2_nOutW2_a
j6						FIF_CAN2_nOutW3_a
7						FIF_CAN2_nOutW4_a
8						FIF_CAN3_nOutW1_a
9						FIF_CAN3_nOutW2_a
10						FIF_CAN3_nOutW3_a
11						FIF_CAN3_nOutW4_a
C2500						Temporary codes (see chapter 8.6.1)
			0	{1}	65535	
1		0				%MW 0
...		...				...
255		0				%MW 254
C2501						Temporary codes (see chapter 8.6.1)
			0	{1}	65535	
1		0				%MW 255
...		...				...
255		0				%MW 509

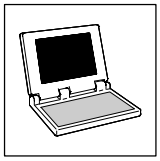


8.6.1 Temporary codes

Codes C2500 and C2501 are temporary codes, i.e. their data

- do not use any EEPROM memory space.
- cannot be saved in the parameter set of the unit using C0003 = 1.
- will be lost after mains disconnection or power failure.
- are linked with the PLC flag area.

- Temporary codes should be used for parameters which are accessed during a switch-on cycle of the PLC.
- It is also possible to directly access the flag are of the PLC (e.g via HMI) without having to create a variable.



8.6.2 RAM access via codes

If you want to access the RAM of the PLC from external controls or PC tools, for instance to manipulate movement profile data online, you can use the following codes:

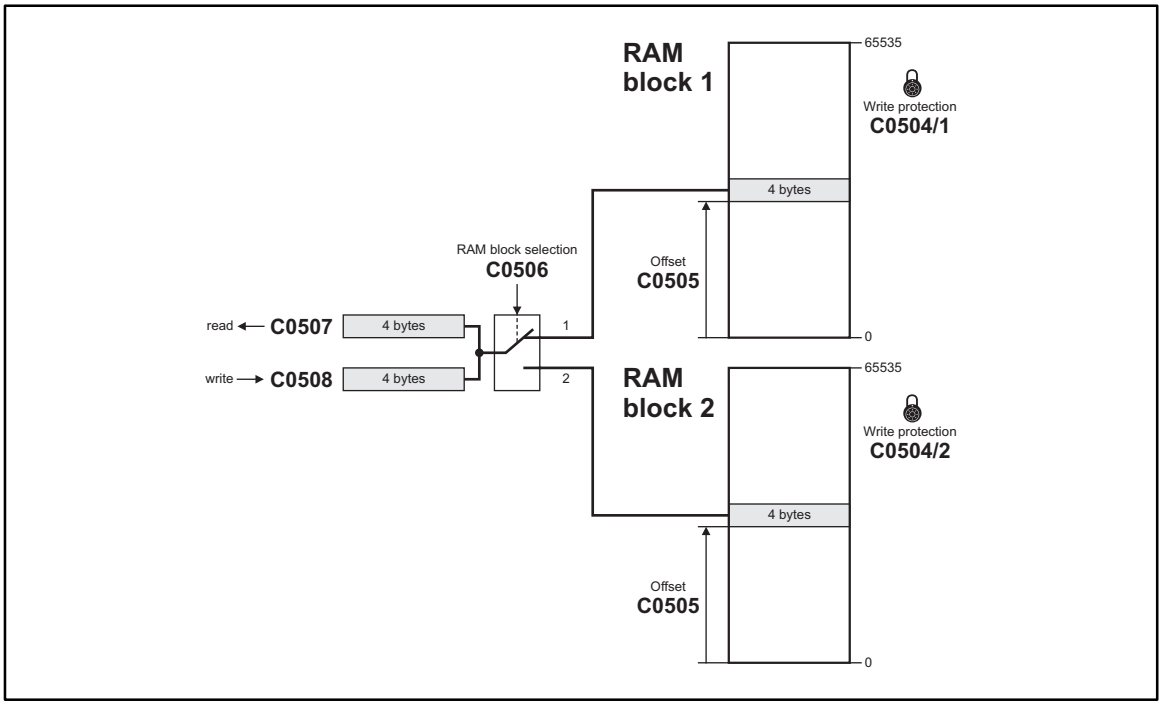


Fig. 8-1 Codes for RAM access

Parameters					
Code / **Subcode**	**Data type**	**Access**	**Info**		
			Possible settings		**Presetting**
C0504	-	R / W	Activate/deactivate write protection for RAM • When the write protection is activated, the RAM cannot be written via LenzeMemDrv.lib codes or functions.		
1			0	Deactivate write protection for RAM block 1	0
			1	Activate write protection for RAM block 1	
2			0	Deactivate write protection for RAM block 2	0
			1	Activate write protection for RAM block 2	
C0505	-	W	Offset address within the RAM block selected under C0506		
			0 {1} 65532		0
C0506	-	W	Selection of the RAM block for access under C0508/C0509		
			1	RAM block 1	
			2	RAM block 2	
C0507	Double integer	R	Value read from the RAM block • After the RAM block has been read, the pointer to the memory address is automatically incremented by 4 bytes.		
C0508	Double integer	W	Value to be written in the RAM block • After the RAM block has been written, the pointer to the memory address is automatically incremented by 4 bytes.		
C0509	-	R / W	Check sum test		
			0	Deactivated	0
			1	Activated	

Note!

- The RAM access is processed in parallel with the PLC program in the system task. The processing time therefore depends on the workload of the system.
- You can use the functions of the **LenzeMemDrv.lib** function library to access the RAM from your IEC 61131 program.

Auto-increment access

The four data bytes are always read/written through “auto-increment access”, i.e. the pointer to the address in the selected RAM block is automatically incremented by four bytes after reading code C0507 or writing code C0508:

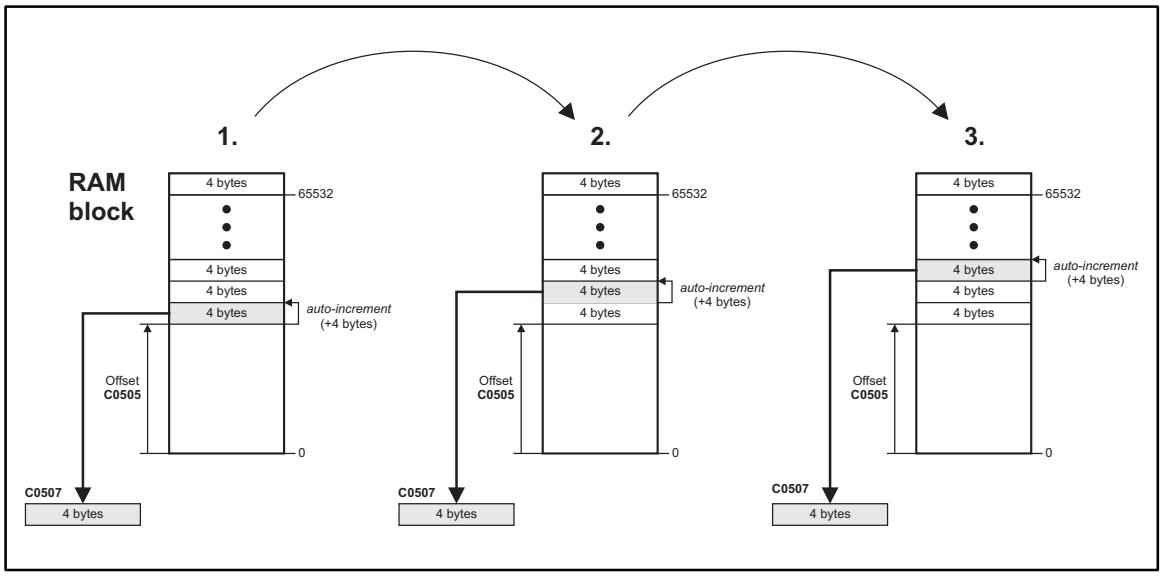
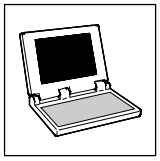


Fig. 8-2 Example: Reading successive double integer values from the RAM block through auto-increment access



8.7 Attribute table

If you want to create your own program, you need the information given in the attribute table. It informs about communication with the PLC via parameters.

How to read the attribute table:

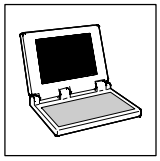
Column		Meaning	Entry	
Code		Name of the Lenze code	Cxxxx	
Index	dec	Index used to address a parameter A subindex of an array variable is the same as a Lenze subcode number	24575 - Lenze codes	Only needed for control via INTERBUS-S, PROFIBUS-DP or system bus (CAN).
	hex		5FFFh - Lenze code	
Data	DS	Data structure	U	Single variable (one parameter element only)
			A	Array variable (several parameter elements
	DA	Number of array elements (subcodes)		
	DT	Data type	B8	1 byte bit-coded
			B16	2 byte bit-coded
			B32	4 byte bit-coded
			FIX32	32 bit value with sign; decimal with four decimal positions
			I32	4 byte with sign
			U16	2 byte without sign
			U32	4 byte without sign
			VS	ASCII string
	Format	LECOM format (see also the Operating Instructions for the 2102 fieldbus module)	VD	ASCII decimal format
			VH	ASCII hexadecimal format
			VS	String format
			VO	Octett string format for data blocks
	DL	Data length in byte		
	Decimal position	Number of decimal positions		
Access	LCM-R/W	Access authorisation for LECOM	Ra	Reading is always permitted
			Wa	Writing is always permitted
			W	Writing is bound to a condition
	Condition	Writing condition	CINH	Writing is only permitted when the controller is inhibited (CINH) [1]
			PLC run	Writing is only permitted when the program is running.
				[1] Only with 9300 Servo PLC

Code	Index		Data						Access	
	dec	hex	DS	DA	DT	Format	DL	Decimal position	LCM-R/W	Condition
C0002	24573	5FFDh	U	1	FIX32	VD	4	0	Ra/W	PLC run
C0003	24572	5FFCh	U	1	FIX32	VD	4	0	Ra/Wa	
C0004	24571	5FFBh	U	1	FIX32	VD	4	0	Ra/Wa	
C0009	24566	5FF6h	U	1	FIX32	VD	4	0	Ra/Wa	
C0011	24564	5FF4h	U	1	FIX32	VD	4	0	Ra/Wa	
C0043	24532	5FD4h	U	1	FIX32	VD	4	0	Ra/Wa	
C0067	24508	5FBCh	U	1	FIX32	VD	4	0	Ra	
C0093	24482	5FA2h	U	1	FIX32	VD	4	0	Ra	
C0094	24481	5FA1h	U	1	FIX32	VD	4	0	Ra/Wa	
C0096	24479	5F9Fh	A	2	FIX32	VD	4	0	Ra/Wa	
C0099	24476	5F9Ch	U	1	FIX32	VD	4	1	Ra	
C0125	24450	5F82h	U	1	FIX32	VD	4	0	Ra/Wa	
C0126	24449	5F81h	U	1	FIX32	VD	4	0	Ra/Wa	
C0135	24440	5F78h	U	1	B16	VH	2	0	Ra/Wa	

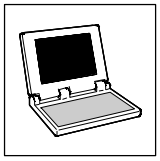
Drive PLC

Appendix

Attribute table



Code	Index		Data						Access	
	dec	hex	DS	DA	DT	Format	DL	Decimal position	LCM-R/W	Condition
C0150	24425	5F69h	U	1	B16	VH	2	0	Ra	
C0161	24414	5F5Eh	U	1	FIX32	VD	4	0	Ra	
C0167	24408	5F58h	U	1	FIX32	VD	4	0	Ra/Wa	
C0168	24407	5F57h	A	8	FIX32	VD	4	0	Ra	
C0169	24406	5F56h	A	8	U32	VH	4	0	Ra	
C0170	24405	5F55h	A	8	FIX32	VD	4	0	Ra	
C0179	24396	5F4Ch	U	1	U32	VH	4	0	Ra	
C0183	24392	5F48h	U	1	FIX32	VD	4	0	Ra	
C0199	24376	5F38h	U	1	FIX32	VD	4	0	Ra	
C0200	24375	5F37h	U	1	VS	VS	14	0	Ra	
C0201	24374	5F36h	U	1	VS	VS	20	0	Ra	
C0202	24373	5F35h	U	1	FIX32	VD	4	0	Ra	
C0203	24372	5F34h	U	1	VS	VS	12	0	Ra	
C0204	24371	5F33h	U	1	FIX32	VD	4	0	Ra	
C0206	24369	5F31h	U	1	VS	VS	13	0	Ra	
C0207	24368	5F30h	U	1	VS	VS	14	0	Ra	
C0208	24367	5F2Fh	U	1	VS	VS	14	0	Ra	
C0209	24366	5F2Eh	U	1	VS	VS	14	0	Ra	
C0210	24365	5F2Dh	U	1	VS	VS	14	0	Ra	
C0300	24275	5ED3h	U	1	FIX32	VD	4	0	Ra	
C0301	24274	5ED2h	U	1	FIX32	VD	4	0	Ra	
C0350	24225	5EA1h	U	1	FIX32	VD	4	0	Ra/Wa	
C0351	24224	5EA0h	U	1	FIX32	VD	4	0	Ra/Wa	
C0352	24223	5E9Fh	U	1	FIX32	VD	4	0	Ra/Wa	
C0353	24222	5E9Eh	A	3	FIX32	VD	4	0	Ra/Wa	
C0354	24221	5E9Dh	A	6	FIX32	VD	4	0	Ra/Wa	
C0355	24220	5E9Ch	A	6	FIX32	VD	4	0	Ra	
C0356	24219	5E9Bh	A	4	FIX32	VD	4	0	Ra/Wa	
C0357	24218	5E9Ah	A	3	FIX32	VD	4	0	Ra/Wa	
C0358	24217	5E99h	U	1	FIX32	VD	4	0	Ra/Wa	
C0359	24216	5E98h	U	1	FIX32	VD	4	0	Ra	
C0360	24215	5E97h	A	12	FIX32	VD	4	0	Ra	
C0361	24214	5E96h	A	12	FIX32	VD	4	2	Ra	
C0362	24213	5E95h	U	1	FIX32	VD	4	3	Ra	
C0363	24212	5E94h	U	1	FIX32	VD	4	0	Ra/Wa	
C0365	24210	5E92h	U	1	FIX32	VD	4	0	Ra	
C0366	24209	5E91h	U	1	FIX32	VD	4	0	Ra/Wa	
C0367	24208	5E90h	U	1	FIX32	VD	4	0	Ra/Wa	
C0368	24207	5E8Fh	U	1	FIX32	VD	4	0	Ra/Wa	
C0369	24206	5E8Eh	U	1	FIX32	VD	4	0	Ra/Wa	
C0372	24203	5E8Bh	U	1	FIX32	VD	4	0	Ra	
C0373	24202	5E8Ah	U	1	B8	VH	1	0	Ra	
C0400	24175	5E6Fh	U	1	FIX32	VD	4	2	Ra	
C0405	24170	5E6Ah	U	1	FIX32	VD	4	2	Ra	
C0425	24150	5E56h	U	1	FIX32	VD	4	0	Ra/Wa	
C0426	24149	5E55h	U	1	FIX32	VD	4	0	Ra	
C0428	24147	5E53h	U	1	FIX32	VD	4	0	Ra/Wa	
C0429	24146	5E52h	U	1	FIX32	VD	4	0	Ra/Wa	
C0431	24144	5E50h	U	1	FIX32	VD	4	0	Ra/W	
C0434	24141	5E4Dh	A	3	FIX32	VD	4	2	Ra	
C0443	24132	5E44h	U	1	B8	VH	1	0	Ra	

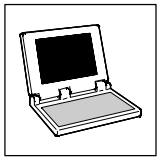


Drive PLC

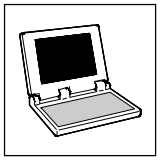
Appendix

Attribute table

Code	Index		Data						Access	
	dec	hex	DS	DA	DT	Format	DL	Decimal position	LCM-R/W	Condition
C0444	24131	5E43h	A	4	FIX32	VD	4	0	Ra	
C0446	24129	5E41h	A	2	B8	VH	1	0	Ra	
C0448	24127	5E3Fh	U	1	B8	VH	1	0	Ra	
C0470	24105	5E29h	A	4	B8	VH	1	0	Ra/Wa	
C0471	24104	5E28h	U	1	B32	VH	4	0	Ra/Wa	
C0472	24103	5E27h	A	20	FIX32	VD	4	2	Ra/Wa	
C0473	24102	5E26h	A	10	FIX32	VD	4	0	Ra/Wa	
C0474	24101	5E25h	A	5	I32	VH	4	0	Ra/Wa	
C0475	24100	5E24h	A	2	FIX32	VD	4	0	Ra/Wa	
C0477	24098	5E22h	U	1	B8	VH	1	0	Ra	
C0479	24096	5E20h	U	1	B8	VH	1	0	Ra	
C0481	24094	5E1Eh	U	1	FIX32	VD	4	2	Ra	
C0484	24091	5E1Bh	U	1	FIX32	VD	4	2	Ra	
C0517	24058	5DFAh	A	32	FIX32	VD	4	2	Ra/Wa	
C0591	23984	5DB0h	U	1	FIX32	VD	4	0	Ra/Wa	
C0592	23983	5DAFh	U	1	FIX32	VD	4	0	Ra/Wa	
C0593	23982	5DAEh	U	1	FIX32	VD	4	0	Ra/Wa	
C0595	23980	5DACh	U	1	FIX32	VD	4	0	Ra/Wa	
C0603	23972	5DA4h	U	1	FIX32	VD	4	0	Ra/Wa	
C0608	23967	5D9Fh	U	1	FIX32	VD	4	0	Ra/Wa	
C0855	23720	5CA8h	A	2	B16	VH	2	0	Ra	
C0856	23719	5CA7h	A	3	I32	VH	4	2	Ra	
C0858	23717	5CA5h	A	3	I32	VH	4	2	Ra	
C0863	23712	5CA0h	A	6	B16	VH	2	0	Ra	
C0866	23709	5C9Dh	A	11	FIX32	VD	4	2	Ra	
C0868	23707	5C9Bh	A	11	FIX32	VD	4	2	Ra	
C1120	23455	5B9Fh	U	1	FIX32	VD	4	0	Ra/Wa	
C1121	23454	5B9Eh	A	2	FIX32	VD	4	0	Ra/Wa	
C1122	23453	5B9Dh	U	1	FIX32	VD	4	3	Ra/Wa	
C1123	23452	5B9Ch	A	2	FIX32	VD	4	3	Ra/Wa	
C1810	22765	58EDh	U	1	VS	VS	14	0	Ra	
C1811	22764	58ECh	U	1	VS	VS	14	0	Ra	
C2100	22475	57CBh	U	1	FIX32	VD	4	0	Ra/Wa	
C2102	22473	57C9h	U	1	FIX32	VD	4	0	Ra/Wa	
C2104	22471	57C7h	U	1	FIX32	VD	4	0	Ra/Wa	
C2108	22467	57C3h	U	1	FIX32	VD	4	0	Ra/Wa	
C2111	22464	57C0h	U	1	VS	VS	14	0	Ra	
C2113	22462	57BDh	U	1	VS	VS	14	0	Ra	
C2115	22460	57BBh	E1	1	U16	VH	2	0	Ra/Wa	
C2117	22458	57B9h	U	1	FIX32	VD	4	0	Ra	
C2118	22457	57B8h	U	1	FIX32	VD	4	0	Ra/Wa	
C2120	22455	57B7h	U	1	FIX32	VD	4	0	Ra/Wa	
C2121	22454	57B6h	U	1	B8	VH	1	0	Ra	
C2350	22225	56D1h	U	1	FIX32	VD	4	0	Ra/Wa	
C2351	22224	56D0h	U	1	FIX32	VD	4	0	Ra/Wa	
C2352	22223	56CFh	U	1	FIX32	VD	4	0	Ra/Wa	
C2354	22221	56CDh	A	6	FIX32	VD	4	0	Ra/Wa	
C2355	22220	56CCh	A	6	FIX32	VD	4	0	Ra/Wa	
C2356	22219	56CBh	A	5	FIX32	VD	4	0	Ra/Wa	
C2357	22218	56CAh	A	4	FIX32	VD	4	0	Ra/Wa	
C2359	22216	56C8h	E1	1	FIX32	VD	4	0	Ra/Wa	



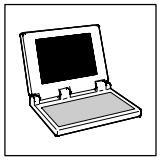
Code	Index		Data						Access	
	dec	hex	DS	DA	DT	Format	DL	Decimal position	LCM-R/W	Condition
C2367	22208	56C0h	U	1	FIX32	VD	4	0	Ra/Wa	
C2368	22207	56BFh	U	1	FIX32	VD	4	0	Ra/Wa	
C2373	22202	56BAh	A	3	FIX32	VD	4	0	Ra/Wa	
C2374	22201	56B9h	A	3	FIX32	VD	4	0	Ra/Wa	
C2375	22200	56B8h	A	3	FIX32	VD	4	0	Ra/Wa	
C2376	22199	56B7h	A	4	FIX32	VD	4	0	Ra/Wa	
C2377	22198	56B6h	A	4	FIX32	VD	4	0	Ra/Wa	
C2378	22197	56B5h	A	4	FIX32	VD	4	0	Ra/Wa	
C2382	22193	56B1h	A	5	FIX32	VD	4	0	Ra/Wa	
C2450	22125	566Dh	U	1	FIX32	VD	4	0	Ra/Wa	
C2451	22124	566Ch	U	1	FIX32	VD	4	0	Ra/Wa	
C2452	22123	566Bh	U	1	FIX32	VD	4	0	Ra/Wa	
C2453	22122	566Ah	A	3	FIX32	VD	4	0	Ra/Wa	
C2454	22121	5669h	A	6	FIX32	VD	4	0	Ra/Wa	
C2455	22120	5668h	A	6	FIX32	VD	4	0	Ra	
C2456	22119	5667h	A	4	FIX32	VD	4	0	Ra/Wa	
C2457	22118	5666h	A	3	FIX32	VD	4	0	Ra/Wa	
C2458	22117	5665h	U	1	FIX32	VD	4	0	Ra/Wa	
C2459	22116	5664h	U	1	FIX32	VD	4	0	Ra	
C2460	22115	5663h	A	12	FIX32	VD	4	0	Ra	
C2461	22114	5662h	A	12	FIX32	VD	4	2	Ra	
C2466	22109	565Dh	U	1	FIX32	VD	4	0	Ra/Wa	
C2467	22108	565Ch	U	1	FIX32	VD	4	0	Ra/Wa	
C2468	22107	565Bh	U	1	FIX32	VD	4	0	Ra/Wa	
C2469	22106	565Ah	U	1	FIX32	VD	4	0	Ra/Wa	
C2470	22105	5659h	U	1	FIX32	VD	4	0	Ra/Wa	
C2481	22094	564Eh	U	1	FIX32	VD	4	0	Ra/Wa	
C2482	22093	564Dh	U	1	FIX32	VD	4	0	Ra/Wa	
C2483	22092	564Ch	U	1	FIX32	VD	4	0	Ra/Wa	
C2484	22091	564Bh	U	1	FIX32	VD	4	0	Ra/Wa	
C2485	22090	564Ah	U	1	FIX32	VD	4	0	Ra/Wa	
C2491	22084	5644h	A	6	B16	VH	2	0	Ra	
C2492	22083	5643h	A	11	FIX32	VD	4	2	Ra	
C2493	22082	5642h	A	11	FIX32	VD	4	2	Ra	
C2500	22075	563Bh	A	255	FIX32	VD	4	0	Ra/Wa	
C2501	22074	563Ah	A	255	FIX32	VD	4	0	Ra/Wa	



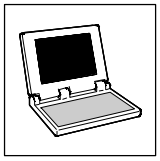
Drive PLC

Appendix

Attribute table

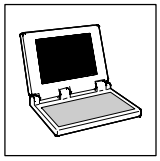


9 Index



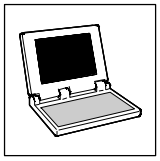
Drive PLC

Index



S

T



Drive PLC

Index