

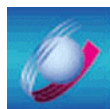
Lenze

Manual

IEC 61131-3

inside

***Global Drive
PLC Developer Studio***



Global Drive

Function library

LenzeElectricalShaft.lib

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

	Type	from hardware version	from software version
9300 Servo PLC	EVS93XX-xI	2K	10
9300 Servo PLC	EVS93XX-xT	2K	10
ECSxA	ECSxAxxx	1C	7.0

Important note:

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

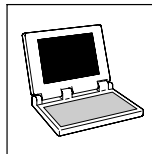
We do not take any liability for direct or indirect damage, e.g. profit loss, order loss or any loss regarding business.

© 2005 Lenze Drive Systems GmbH

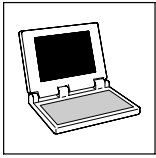
No part of this documentation may be copied or made available to third parties without the explicit written approval of Lenze Drive Systems GmbH.

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

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

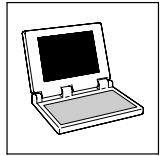


1	Preface and general information	1-1
1.1	About this Manual	1-1
1.1.1	Conventions in this Manual	1-1
1.1.2	Pictograms in this Manual	1-1
1.1.3	Terminology used	1-2
1.2	Lenze software guidelines for variable names	1-3
1.2.1	Hungarian Notation	1-3
1.2.1.1	Recommendation for designating variable types	1-4
1.2.1.2	Designation of the signal type in the variable name	1-5
1.2.1.3	Special handling of system variables	1-5
1.3	Version identifiers of the function library	1-6
2	Function blocks	2-1
2.1	Special functions	2-1
2.1.1	Digital frequency ramp generator (L_DFRFG)	2-1
2.1.2	Digital frequency processing (L_DFSET)	2-8
2.1.3	Gearing compensation (L_GEARCOMP)	2-16
2.1.4	Homing function (L_REF)	2-17
3	Appendix	3-1
3.1	Code table	3-1
3.1.1	L_DFRFG	3-1
3.1.2	L_DFSET	3-2
3.1.3	L_GEARCOMP	3-2
3.1.4	L_REF	3-3
4	Index	4-1



Function block library LenzeElectricalShaft.lib

Contents



1 Preface and general information

1.1 About this Manual

This Manual contains information on the function blocks that are included in the function block library **LenzeElectricalShaft.lib** for the **Drive PLC Developer Studio** .

- These function blocks can be used, for instance, in the **9300 Servo PLC** automation system.
- The function blocks are based on the functions that are available in the 9300 servo controller (V2.0).

In **Drive PLC Developer Studio** (DDS) you make the basic settings for your drive application offline, by using variables (in accordance with the IEC 61131-3 standard) as aids for parameterizing the appropriate function blocks.

By using **Global Drive Control** (GDC) or **keypad** you can then *Online* set the parameters for the required functionality of your drive application, by accessing the code positions for the various instances of the function blocks.

1.1.1 Conventions in this Manual

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

Variable names

are shown in the explanatory texts in italics:

- “The signal at *nIn_a* ...”

can be recognized by the names. They always begin with “L_”:

- “The FB L_ARIT can ...”

Instances

For function blocks that have one or more first instances, there are tables that describe the corresponding codes:

Variable name	L_ARIT1	L_ARIT2		Setting range	Lenze
byFunction	C0338	C0600		0 ... 5	1

You can access these codes *Online* is linked to **Global Drive Control** (GDC) or **keypad** .

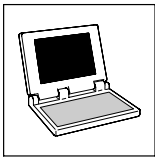


Tip!

You can use the Parameter Manager to assign the same codes to these instances that are assigned in the 9300 servo controller(V2.0).

1.1.2 Pictograms in this Manual

	Use of Pictographs	Signal words	
Warning of material damage		Stop!	Warns of potential damage to material . Possible consequences if disregarded: Damage of the controller/drive system or its environment .
Other notes		Tip!	This note designates general, useful notes. If you observe it, handling of the controller/drive system is made easier.

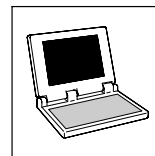


Function block library LenzeElectricalShaft.lib

Preface and general information

1.1.3 Terminology used

Term	In the following text used for
FB	Function block
SB	System block
Parameter codes	Codes for setting the functionality of a function block
GDC	Global Drive Control (parameterization program from Lenze)



1.2 Lenze software guidelines for variable names

The previous concepts for Lenze controllers were based on codes that represented the input and output signals, and the parameters of function blocks.

- For the sake of clarity, names were defined for the codes in the documentation.
- In addition, the signal types were defined by graphical symbols.

The user could see at a glance which kind of signal (analog, phase-angle etc.) had to be present at the particular interface.

The concept for the new automation system does not use direct codes in the programming. The IEC 61131-3 standard is used instead.

- This standard is based on a structure of variable names.
- If the user applies variables in his project, then he can name the variables as he chooses.

In order to avoid the growth of a multitude of different conventions for naming variables in existing and future projects and function libraries that are programmed by Lenze personnel, we have set up software guidelines that must be followed by all Lenze staff.

In this convention for creating variable names, Lenze keeps to the Hungarian Notation that has been specifically expanded by Lenze.

If you make use of Lenze-specific functions or function blocks, you will immediately be able to see, for instance, which data type you must transfer to a function block, and which type of data you will receive as an output value.

1.2.1 Hungarian Notation

These conventions are used so that the most significant characteristics of a program variable can instantly be recognized from its name.

Variable names

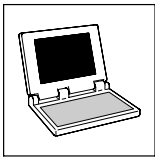
consist of

- a **prefix** (optional)
- a **data-type entry**
- and an **identifier**

The prefix and data-type entry are usually formed by one or two characters. The identifier (the "proper" name) should indicate the application, and is therefore usually somewhat longer.

Prefix examples

prefix	Meaning
a	Array (combined type), field
p	Pointer



Function block library *LenzeElectricalShaft.lib*

Preface and general information

Examples of the data-type entry

Examples of a data-type	Meaning
b	Bool
by	Byte
n	Integer
w	Word
dn	Double integer
dw	Double word
s	String
f	Real (float)
sn	Short integer
t	Time
un	Unsigned integer
udn	Unsigned double integer
usn	Unsigned short integer

Identifier (the proper variable name)

- An identifier begins with a capital letter.
- If an identifier is assembled from several "words", then each "word" must start with a capital letter.
- All other letters are written in lower case.

Examples:

Array of integers	<i>anJogValue[10] ;</i>
Bool	<i>blsEmpty ;</i>
Word	<i>wNumberOfValues ;</i>
Integer	<i>nLoop ;</i>
Byte	<i>byCurrentSelectedJogValue ;</i>

1.2.1.1 Recommendation for designating variable types

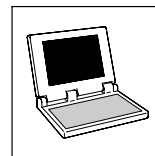
In order to be able to recognize the type of variable in a program according to the name, it makes sense to use the following designations, which are placed in front of the proper variable name and separated from it by an underline stroke:

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

Example

for a global array of type integer that includes fixed setpoints (analog) for a speed setting:

g_anFixSetSpeedValue_a



1.2.1.2 Designation of the signal type in the variable name

The inputs and outputs of the Lenze function blocks each have a specific signal type assigned. These may be: digital, analog, position, or speed signals.

For this reason, each variable name has an ending attached that provides information on the type of signal.

Signal type	Ending	Previous designation
analog	_a (analog)	○
digital	_b (binary)	□
Phase-angle difference or speed	_v (velocity)	△
Phase-angle or position	_p (position)	▲



Tip!

Normalizing to signal type phase-angle difference/speed: $16384 \text{ (INT)} \triangleq 15000 \text{ rpm}$

Normalizing to signal type analog: $16384 \triangleq 100 \% \triangleq \text{value under [C0011]} = n_{\max}$

Normalizing to signal type angle or position: $65536 \triangleq 1 \text{ motor revolution}$

Examples:

Variable name	Signal type	Type of variable
nIn_a	Analog input value	Integer
dnPhiSet_p	Phase signals	Double integer
bLoad_b	Binary value (TRUE/FALSE)	Bool
nDigitalFrequencyIn_v	Speed input value	Integer

1.2.1.3 Special handling of system variables

System variables require special handling, since the system functions are only available for the user as I/O connections in the control configuration.

In order to be able to access a system variable quickly during programming, the variable name must include a label for the system function.

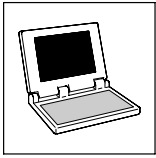
For this reason, the name of the corresponding system block is placed before the name of the variable.

Examples:

AIN1_nIn_a

CAN1_bCtrlTripSet_b

DIGIN_bIn3_b



Function block library *LenzeElectricalShaft.lib*

Preface and general information

1.3 Version identifiers of the function library

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

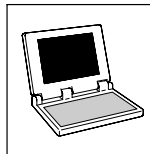
Version identifiers as of PLC software version 7.x:

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

Version: 01 05 00 00

The value of this constant is a hexadecimal code.

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



2 Function blocks

2.1 Special functions

2.1.1 Digital frequency ramp generator (L_DFRFG)

This FB synchronizes the drive (motor shaft) to a digital frequency (phase-angle input). The drive then runs in phase-synchronism with the digital frequency.

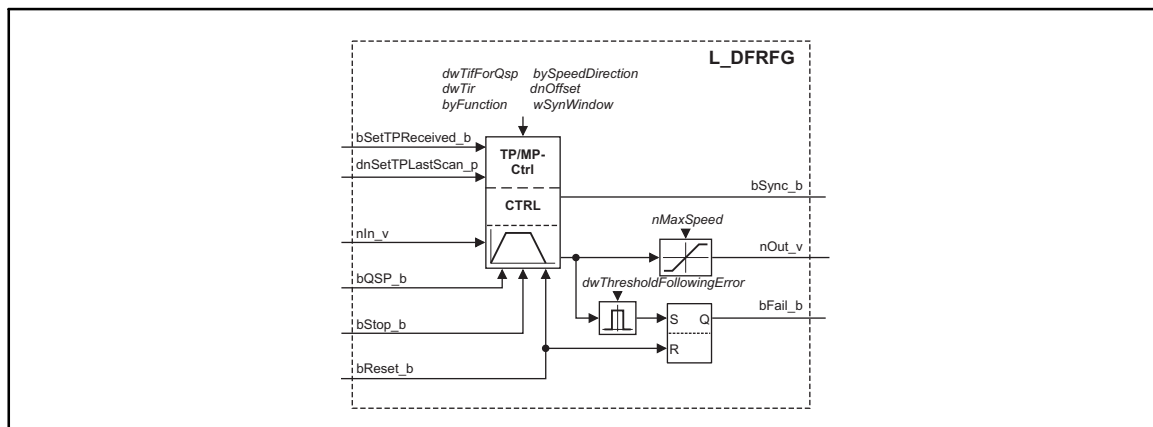
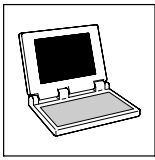


Abb. 2-1

Digital frequency ramp generator (L_DFRFG)

VariableName	DataType	SignalType	VariableType	Note
bSetTPReceived_b	Bool	binary	VAR_INPUT	Zero-pulse or TP (TouchProbe) received
dnSetTPLastScan_p	Double Integer	position	VAR_INPUT	Difference increments between TP and start of the task
nIn_v	Integer	velocity	VAR_INPUT	Speed/phase-angle setpoint (16384 $\equiv$ 15000 rpm)
bQSP_b	Bool	binary	VAR_INPUT	TRUE = Quickstop
bStop_b	Bool	binary	VAR_INPUT	TRUE = save setpoint
bReset_b	Bool	binary	VAR_INPUT	TRUE = reset
bSync_b	Bool	binary	VAR_OUTPUT	TRUE = drive runs synchronously
nOut_v	Integer	velocity	VAR_OUTPUT	Speed/Phase setpoint
bFail_b	Bool	binary	VAR_OUTPUT	TRUE = exceeded phase-angle difference
dwTir	Double Word	-	VAR CONSTANT RETAIN	Acceleration time
nMaxSpeed	Integer	-	VAR CONSTANT RETAIN	max. rate of catching up
dwTiffForQSP	Double Word	-	VAR CONSTANT RETAIN	Deceleration time for QSP (Quickstop)
dwThresholdFollowingError	Double Integer	-	VAR CONSTANT RETAIN	Second contouring error (65335 $\equiv$ 1 turn)
wSynWindow	Word	-	VAR CONSTANT RETAIN	Synchronization window
dnOffset	Double Integer	-	VAR CONSTANT RETAIN	Offset related to the digital frequency input (65335 $\equiv$ 1 turn)
byFunction	Byte	-	VAR CONSTANT RETAIN	Start with/without TP
bySpeedDirection	Byte	-	VAR CONSTANT RETAIN	Input of the direction of rotation



Function block library *LenzeElectricalShaft.lib*

Special functions

Digital frequency ramp generator (*L_DFRFG*)

Parameter codes of the instances

VariableName	L_DFRFG1			SettingRange	Lenze
dwTir	C0751			0.001 ... 999.900 s	1.000
nMaxSpeed	C0752			1 ... 16000 rpm	3000
dwTifForQSP	C0753			0.000 ... 999.900 s	0.000
dwThresholdFollowingError	C0754			10 ... 2000000000 incr.	2000000000
wSynWindow	C0755			0 ... 65535 incr.	100
dnOffset	C0756			-2147483647 ... 2147483647 incr.	0
byFunction	C0757			0, 1	0
bySpeedDirection	C0766			1 ... 3	1

Range of functions

- Profile generator
- Quickstop
- Ramp generator stop
- RESET
- Detect phase difference
- Set offset
- Connection of the function block



Note!

This FB operates with remainder handling.

2.1.1.1

Profile generator

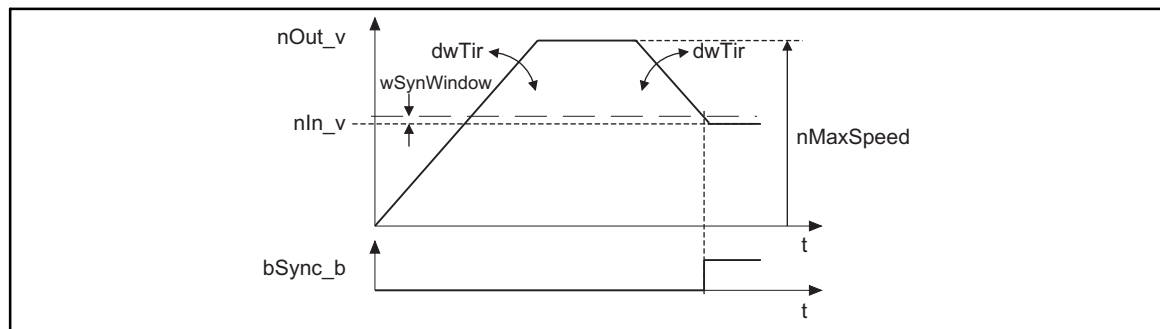


Abb. 2-2

Diagram of the synchronization of *L_DFRFG*

The profile generator generates ramps which lead the setpoint phase to its target.

- Use *dwTir* to set the ramp-up and ramp-down times ($1000 \equiv 1$ s).
- Use *nMaxSpeed* to set the maximum speed ($17476 \equiv 16000$ rpm).
- If the distance and the speed reach their setpoints, the output switches *bSync_b* = TRUE. At the same time, the FB switches the profile generator inactive and *nIn_v* is output to *nOut_v*.
- Use *wSynWindow* to set the changeover point for *bSync_b* ($1 \equiv 1$ incr.).



Stop!

Do not operate the drive at the torque limit with this function i_{max} , I_{max} .

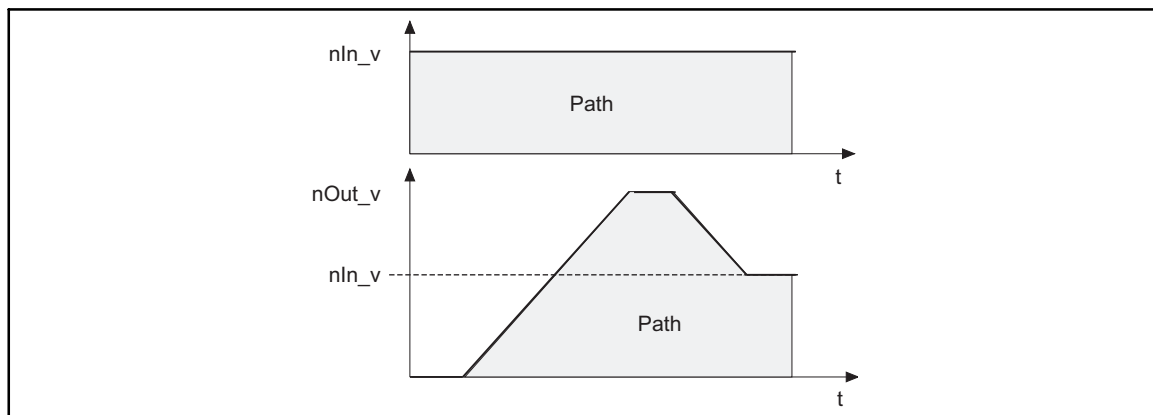
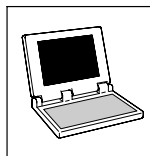


Abb. 2-3

Speed-time diagram for *L_DFRFG*

The number of increments at *nIn_v* (master drive) define the target point. The target can be represented as a distance. In the speed-time diagram, the distance covered (phase) is shown as the area under the speed profile. When synchronization is reached, master and slave have covered the same distance (phase).

2.1.1.2 Quickstop

Removes the drive from the network and brakes it to standstill.

- Quickstop is activated with *bQSP_b* = TRUE
- Select the constant *FIXED0%* (selection number 1000) using *dwTifForQSP* to set the ramp-down time (1000 $\equiv$ 1 s).
- The set phase-angle detected at *nIn_v* is saved (16384 $\equiv$ 15000 rpm).
- After reset of the Quickstop request, the set phase is approached via the profile generator.

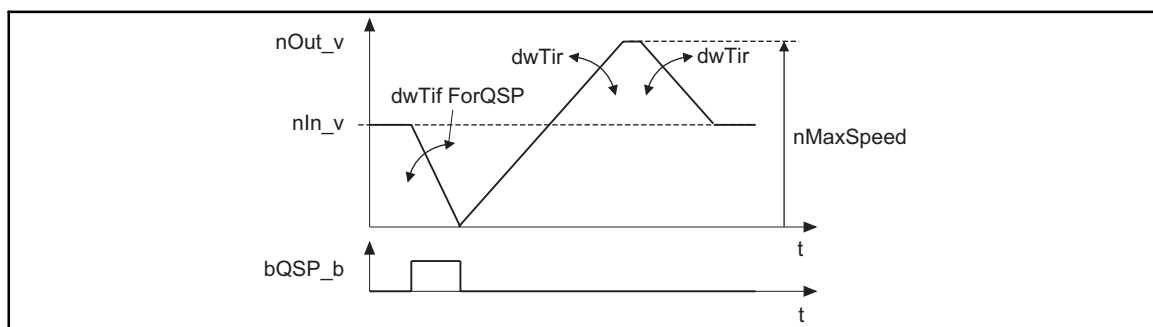
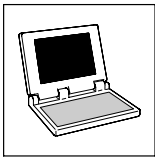


Abb. 2-4

Diagram of the Quickstop function



Function block library *LenzeElectricalShaft.lib*

Special functions

Digital frequency ramp generator (*L_DFRFG*)

2.1.1.3 Ramp generator stop

Maintains the state of the profile generator during operation.

- Select the constant FIXED0% (selection number 1000) using *bStop_b* = TRUE, the function “ramp generator stop” is activated.
- The latest state is output at *nOut_v* (16384 $\equiv$ 15000 rpm).
- The set phase-angle detected at *nIn_v* is stored.
- After reset of the Quickstop request, the set phase-angle is approached via the profile generator.

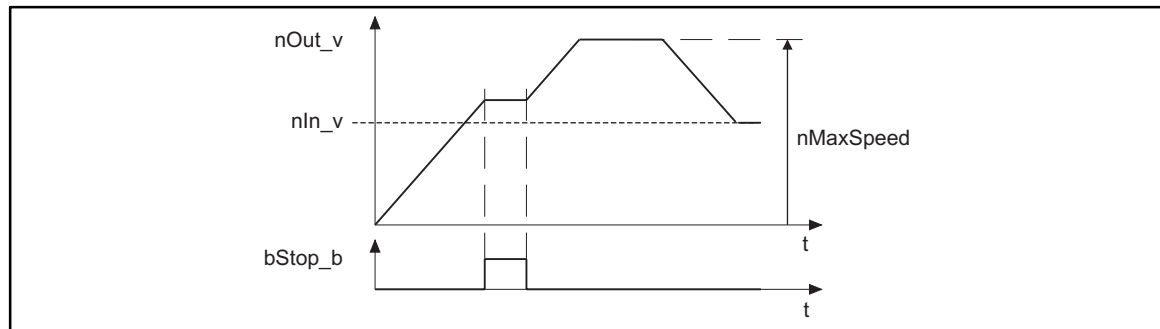


Abb. 2-5

Diagram of the function “ramp generator stop”

2.1.1.4 RESET

bReset_b = TRUE:

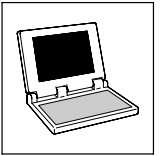
- Resets the setpoint phase which has been internally summed up.
- Activates the profile generator.

A TRUE-FALSE edge at *bReset_b* detects the set phase-angle.

2.1.1.5 Detect phase difference

Monitoring the phase-angle difference between *nIn_v* and *nOut_v*.

- Select the constant FIXED0% (selection number 1000) using *dwThresholdFollowingError* to adjust the limit for the monitoring (1 $\equiv$ 1 incr.).
- If the monitoring is triggered, then *bFail_b* is set = TRUE.
- Select the constant FIXED0% (selection number 1000) using *bReset_b* = TRUE to store the signal.
- The profile generator can accept a phase-angle difference of up to ± 2140000000 incr. (≈ 32000 turns) (65536 $\equiv$ 1 turn).



2.1.1.6 Set offset

The offset is adjusted with *dnOffset* (1 ≡ 1 incr.).

The offset refers to the digital frequency input and is normalised (scaled) to 1 revolution (= 65536 increments).

Select the constant *FIXED0%* (selection number 1000) using *bSetTPReceived_b* = TRUE starts the ramp generator. The lead of the master from the moment of starting, or the resulting distance/phase difference, is taken up during acceleration.

- Setting: positive offset values
 - Causes a time-shift of the TP
 - This means that less time is necessary to obtain synchronism with the master. (For instance, with *dnOffset* = 0 this cannot be achieved).



Note!

With large offset values and low input speeds the drive could change direction. To prevent this, you can use *bySpeedDirection* to fix the direction of rotation:

- *bySpeedDirection*: 1 = cw/ccw, 2 = cw, 3 = ccw

2.1.1.7 Start of the ramp generator

Use *byFunction* to parameterize the start of the ramp generator.

byFunction = 0:

The ramp generator is started immediately on the first call.

byFunction = 1:

The ramp generator is now started through *bSetTPReceived_b*.

2.1.1.8 Use of *L_DFRFG* with the 9300 Servo PLC



Note!

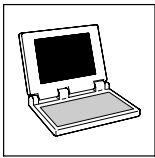
See also Manual "9300 Servo PLC",
Description of the system blocks **DF_IN_DigitalFrequency** and **MCTRL_MotorControl**.

Nature of the task:

- A 9300 Servo PLC is to be synchronized to the speed setpoint at the digital frequency input.
- The digital frequency input provides a constant speed of 867 rpm through *DFIN_nIn_v*:

$$DFIN\_nIn\_v = 947 = \frac{15000rpm \cdot 16384}{867rpm}$$

- The start of synchronization is made by the TouchProbe input of the SB **DF_IN_DigitalFrequency** (X5/E5).
- Through the FB **L_DFSET** a phase-angle difference is controlled down to zero with the help of the phase-angle controllers in the SB **MCTRL_MotorControl**.
- The maximum speed for the ramp generator must not exceed 3000 rpm.
- Both right (cw) and left (ccw) rotation of the motor are permitted.
- The acceleration (run-up) time should be 10 seconds.

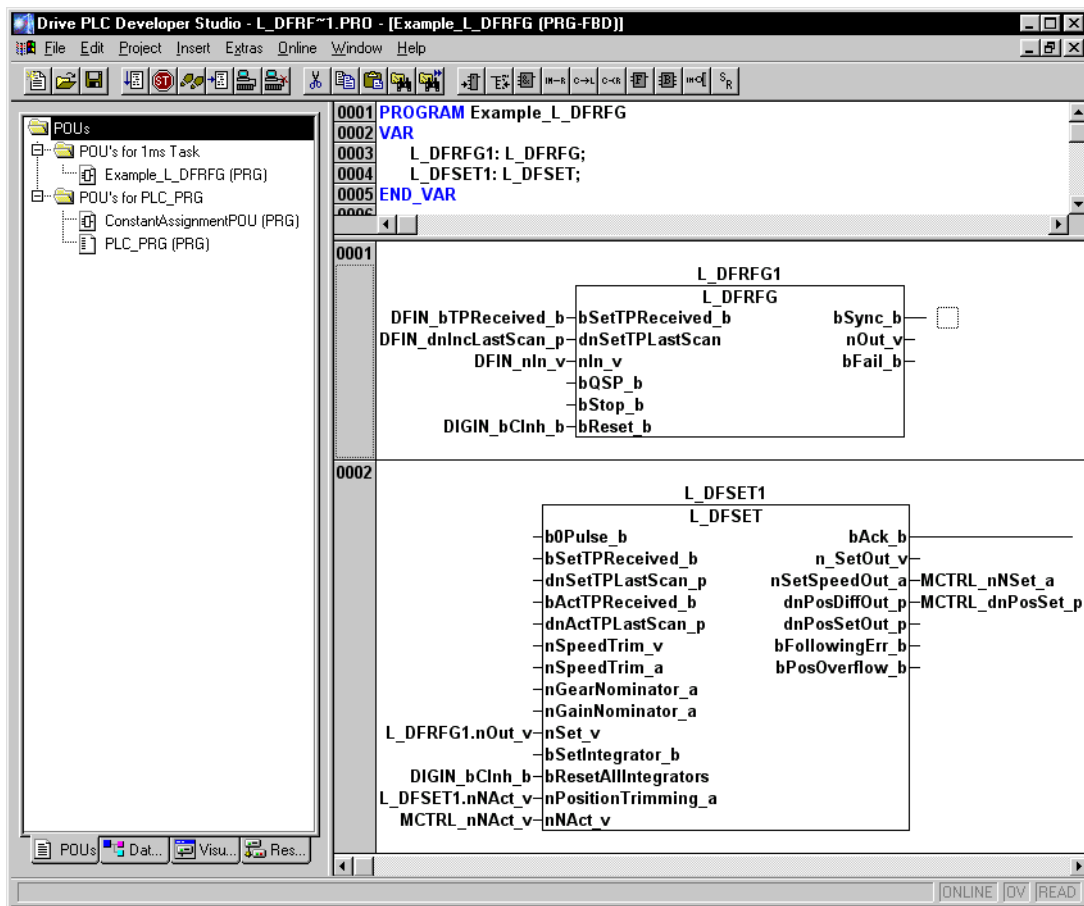


Function block library *LenzeElectricalShaft.lib*

Special functions

Digital frequency ramp generator (*L_DFRFG*)

Implementation:

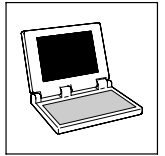


- Create one instance each of the FB **L_DFRFG** and the FB **L_DFSET**.
- Assign the first-instance codes to these instances, using the Instance-Parameter Manager. In this way, all the values for the parameterization of the variables (e.g. for scaling and initialization) will be set to the Lenze standard settings.



Note!

You can find the Lenze standard settings for the instances of an FB in the table “Parameter Codes for the Instances” in the corresponding FB description.



Setting the initial values of L_DFRFG1 in the Instance-Parameter Manager

- Alter the initial value of *byFunction* from 0 to 1.
The start of synchronization is then initiated through the Touchprobe input (X5/E5) of the SB **DF_IN_DigitalFrequency** (*L_DFRFG.bSetTPReceived_b*).
DFIN_dnlncLastScan_p assigns the phase difference between the appearance of the Touchprobe signal and the start of the task in which **L_DFRFG1** is called, for the calculation of the path, to **L_DFRFG1** .
- Alter the initial value of *dwTir* from 1.000 to 10.000, so that the ramp-up/ramp-down time of the ramp generator is 10 seconds.
- The initial value of *bySpeedDirection* includes the Lenze standard setting (DF=1), since both cw (right) and ccw (left) rotation should be possible.
- The initial value of *nMaxSpeed* includes the Lenze standard setting (DF=3000), so the maximum speed of the profile generator will be 3000 rpm. (17476 $\equiv$ 16000 rpm)

Speed setpoint

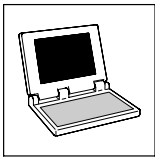
- **L_DFRFG1** assigns the speed setpoint, through **L_DFSET1** (*nSetSpeedOut_a*) to the SB **MCTRL_MotorControl** (*MCTRL_nNSet_a*).
- In **L_DFSET1** a conversion is made from a velocity value (*L_DFRFG1.nOut_v*) to an analog value (*L_DFSET1.nSetSpeedOut_a*):

 <Variable>_v: 16384 $\equiv$ 15000 rpm
 <Variable>_a: 16384 $\equiv$ $n_{\max}$ (C0011)
- During synchronization, the speed setpoint is generated by the synchronization profile (profile generator) and output to *L_DFRFG1.nOut_v* (16384 $\equiv$ 15000 rpm).
- After synchronization, the speed setpoint is switched over from *L_DFRFG1.nIn_v* to *L_DFRFG1.nOut_v* , the setpoint now follows the value of the digital frequency input.

Setpoint for the phase-angle controller

A set/actual phase-angle comparison provides an phase-angle difference that is assigned to the phase-angle controller in the SB **MCTRL_MotorControl**. If there is a difference between the set value and the actual value, the phase-angle controller corrects this deviation.

- The phase-angle controller in the SB **MCTRL_MotorControl** must be activated. (*MCTRL_bPosOn_b* = TRUE)
- The set/actual value comparison is performed in **L_DFSET1** between *L_DFSET1.nSet_v* (setpoint) and *L_DFSET1.nNAct_v* (actual value).
- The phase-angle difference is output via *L_DFSET1.dnPosDiffOut_p*, and then assigned to the phase-angle controller in the SB **MCTRL_MotorControl** through the system variable *MCTRL_dnPosSet_p*.



Function block library *LenzeElectricalShaft.lib*

Special functions

Digital frequency processing (L_DFSET)

2.1.2 Digital frequency processing (L_DFSET)

This FB conditions the digital frequency for the drive controller. You can enter values for the stretch factor and gearing factor, and trim the speed or phase-angle setting.

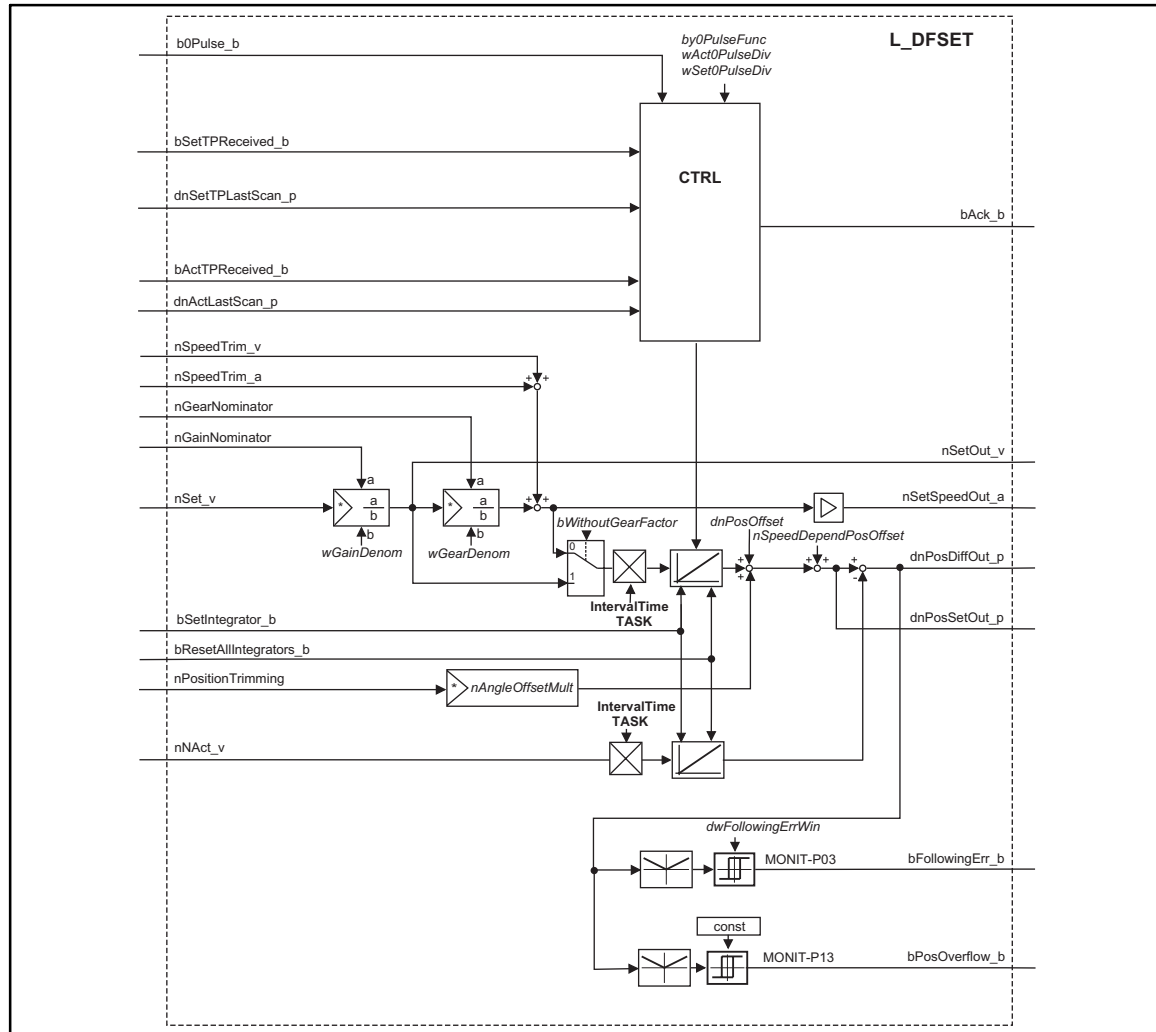
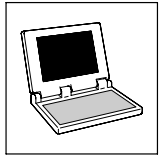
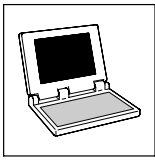


Abb. 2-6 Digital frequency processing (L_DFSET)



VariableName	DataType	SignalType	VariableType	Note
b0Pulse_b	Bool	binary	VAR_INPUT	TRUE = enable 0-pulse synchronization
bSetTPReceived_b	Bool	binary	VAR_INPUT	Setpoint TP (TouchProbe) or zero-pulse received
dnSetTPLastScan_p	Double-integer	position	VAR_INPUT	Phase-angle difference between TP setpoint and start of the task
bActTPReceived_b	Bool	binary	VAR_INPUT	Actual value TP or zero-pulse received
dnActLastScan_p	Double-integer	position	VAR_INPUT	Phase-angle difference between TP actual value and start of the task
nSpeedTrim_v	Integer	velocity	VAR_INPUT	Speed trimming in [rpm]. (16384 = 15000 rpm)
nSpeedTrim_a	Integer	analog	VAR_INPUT	Speed trimming in [%]. (C0011 = 100 % = 16384)
nGearNominator	Integer	analog/velocity	VAR_INPUT	Numerator for gearing factor (100 % = 16384 incr.)
nGainNominator	Integer	analog/velocity	VAR_INPUT	Numerator stretch factor (100 % = 16384 incr.)
nSet_v	Integer	velocity	VAR_INPUT	Speed/Phase setpoint
bSetIntegrator_b	Bool	binary	VAR_INPUT	- TRUE = set phase-angle integrators to be equal - FALSE-TRUE edge sets <i>dnPosDiffOut_p</i> = 0 - TRUE-FALSE edge sets <i>dnPSet_p</i> = actual value of <i>MCTRL_dnPosSet_p</i> <i>bSetIntegrator_b</i> has a higher priority than <i>bResetAllIntegrators_b</i>
bResetAllIntegrators_b	Bool	binary	VAR_INPUT	- TRUE = sets position difference = 0 - TRUE = sets <i>dnPosSetOut_p</i> and <i>dnPosDiffOut_p</i> = 0
nPositionTrimming	Integer	analog/velocity	VAR_INPUT	Phase-angle trimming (100 % = 16384 incr.)
nAct_v	Integer	velocity	VAR_INPUT	Actual value for calculating the actual position
bAck_b	Bool	binary	VAR_OUTPUT	TRUE = synchronizing is performed
nSetOut_v	Integer	velocity	VAR_OUTPUT	Speed/Phase setpoint
nSetSpeedOut_a	Integer	analog	VAR_OUTPUT	in [%] of nmax. (C0011 = 100 % = 16384)
dnPosDiffOut_p	Double-integer	position	VAR_OUTPUT	Contouring error for phase controller
dnPosSetOut_p	Double-integer	position	VAR_OUTPUT	Phase-angle setpoint (65536 incr. = 1 turn)
bFollowingErr_b	Bool	binary	VAR_OUTPUT	Status: TRUE = contouring error
bPosOverflow_b	Bool	binary	VAR_OUTPUT	Status: TRUE = phase-angle controller overflow
wGearDenom	Word	-	VAR CONSTANT RETAIN	Gearbox factor denominator
dnPosOffset	Double-integer	-	VAR CONSTANT RETAIN	Phase offset
nSpeedDependPosOffset	Integer	-	VAR CONSTANT RETAIN	Speed-dependent phase trimming
dwFollowingErrWin	Double-integer	-	VAR CONSTANT RETAIN	Contouring error limit
nAngleOffsetMult	Integer	-	VAR CONSTANT RETAIN	Multiplier for the phase-angle trimming offset
bWithoutGearFactor	Bool	-	VAR CONSTANT RETAIN	Evaluation of the set-phase integrator
wAct0PulseDiv	Word	-	VAR CONSTANT RETAIN	Actual zero-pulse divider
wGainDenom	Word	-	VAR CONSTANT RETAIN	Denominator of the stretch factor
by0PulseFunc	Byte	-	VAR CONSTANT RETAIN	Zero-pulse function (synchronization mode)
wSet0PulseDiv	Word	-	VAR CONSTANT RETAIN	Set zero-pulse divider



Function block library *LenzeElectricalShaft.lib*

Special functions

Digital frequency processing (*L_DFSET*)

Parameter codes of the instances

VariableName	L_DFSET1		SettingRange	Lenze
wGearDenom	C0033		1 ... 32767	1
dnPosOffset	C0252		-245760000 ... 245760000 inc	0
nSpeedDependPos Offset	C0253		-32767 ... 32767 inc	0
dwFollowingErrWin	C0255		10 ... 180000000 inc.	32768
nAngleOffsetMult	C0529		-20000 ... 20000	1
bWithoutGearFactor	C0530		0, 1	0
wActOPulseDiv	C0531		0 ... 16384	0
wGainDenom	C0533		1 ... 32767	1
byOPulseFunc	C0534		0, 1, 2, 10, 11, 12, 13	0
wSetOPulseDiv	C0535		0 ... 16384	0

Range of functions

- Setpoint conditioning with stretch and gearbox factor
- Processing of correction values
- Synchronizing on zero track or touch probe (for resolver feedback: touch probe only)
- Connection of the function block

2.1.2.1 Setpoint conditioning with stretch and gearbox factor

Stretch factor

The stretch factor defines the ratio for the drive running to its setpoint.

- The stretch factor evaluates the setpoints at $nSet\_v$.
The result appears at $nSetOut\_v$ (16384 $\equiv$ 15000 rpm).
- The stretch factor results from numerator and denominator.
 - Numerator: Variable value of $nGainNominator$.
 - Denominator: defined under $wGainDenom$.
- Relationship:

$$nSetOut\_v = nSet\_v \cdot \frac{nGainNominator}{wGainDenom}$$

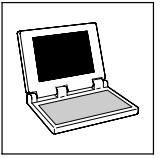
Gearbox factor

The gearing factor defines the gearbox ratio of the drive. Enter the ratio of the drive.

- The gearing factor evaluates the setpoints at $nSet\_v$ multiplied by the stretch factor.
The result appears at $nSetSpeedOut\_a$ (16384 $\equiv$ n_{max} (C0011)).
- The gearing factor is formed from the numerator and denominator.
 - Numerator: Variable value of $nGearNominator$.
 - Denominator: defined under $wGearDenom$.
- Relationship:

$$nSetSpeedOut\_a = Reckfaktor \cdot \frac{nGearNominator}{wGearDenom}$$

$$nSetSpeedOut\_a = nSet\_v \cdot \frac{nGainNominator}{wGainDenom} \cdot \frac{nGearNominator}{wGearDenom}$$



2.1.2.2 Processing of correction values

Speed trimming

Speed trimming is used to add correction values, e.g. from a superimposed closed-loop control. This allows acceleration or deceleration of the drive.

Speed trimming is implemented through addition

- of an analog value at *nSpeedTrim_a* to the speed setpoint ($16384 \equiv n_{\max}$ (C0011))

or

- a speed value at *nSpeedTrim_v* to the speed setpoint ($16384 \equiv 15000$ rpm)
 - The speed trimming via this input is more precise.

Phase trimming

With phase-angle trimming, a set value at *nPositionTrimming* is added to the speed setpoint. This changes the rotor position forwards or backwards relative to the setpoint (drive leading or lagging) depending on the number of increments applied.

The phase trimming can be performed within a range of ± 32767 increments (corresponds to $\pm 1/2$ revolution).

- The input is made in increments (one revolution corresponds to 65536 increments).
- When analog values are entered, 100% correspond to $1/4$ revolution = 16384 increments.
- You can expand the range of adjustment by using the multiplier *nAngleOffsetMult*.

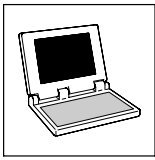
Phase offset

With the phase-angle offset *dnPosOffset* a fixed phase-angle shift is applied to the setpoint for the drive.

Speed-proportional phase setting

The phase-angle shift enables a leading or lagging of the phase with rising speed.

- In *nSpeedDependPosOffset* a corresponding adjustment can be entered in increments.
- The set phase offset is reached at 15000 rpm of the drive (linear relationship).



Function block library *LenzeElectricalShaft.lib*

Special functions

Digital frequency processing (*L_DFSET*)

2.1.2.3 Synchronization to zero track or TouchProbe

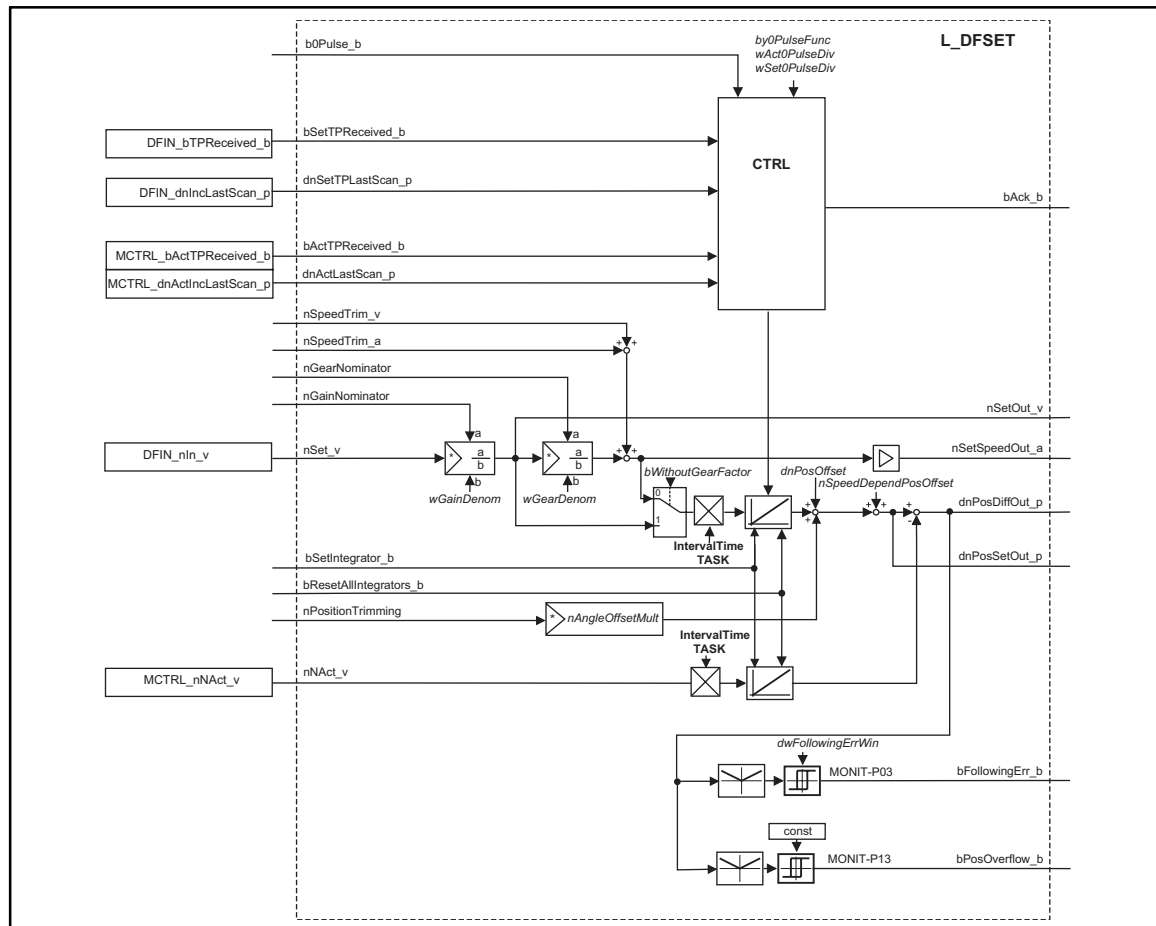


Abb. 2-7 Programming *L_DFSET* for synchronization to zero track or TP (TouchProbe)

The synchronization of the actual value (SB **MCTRL_MotorControl**) to the setpoint (SB **DF_IN_DigitalFrequency**) is made through a zero-pulse or TP (TouchProbe).

- The setpoint is provided through *L_DFSET1.nSet_v*. (16384 $\equiv$ 15000 rpm)
- The actual value is provided through *L_DFSET1.nNAct_v*. (16384 $\equiv$ 15000 rpm)

Zero-pulse from actual value:

- Set code C0911 (SB **MCTRL_MotorControl**) to 0.

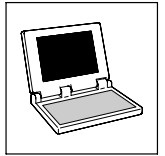


Tip!

Touch probe initiators can have response delay times that cause a speed-dependent phase-angle offset.

Code C0910 can be used to set a correction value (-32767 incr. ... 32767 incr.) for this phase-angle offset (65536 $\equiv$ 1 motor turn).

- Code C0490 is used to select the correct feedback system:
0 Resolver
1 Encoder TTL
2 Encoder sin
3 Absolute ST
4 Absolute MT



Zero-pulse from setpoint:

Set code C0428 (SB **DF_IN_DigitalFrequency**) to 0.



Tip!

Touch probe initiators can have response delay times that cause a speed-dependent phase-angle offset.

Code C0429 can be used to set a correction value (-32767 incr. ... 32767 incr.) for this phase-angle offset.

TouchProbe from actual value:

- Set code C0911 (SB **MCTRL_MotorControl**) to 1.
- The TouchProbe is activated through terminal X5/E4 (actual-value pulse).



Tip!

Touch probe initiators can have response delay times that cause a speed-dependent phase offset.

Code C0910 can be used to set a correction value (-32767 incr. ... 32767 incr.) for this phase-angle offset.

TouchProbe from setpoint:

- Set code C0428 (SB **DF_IN_DigitalFrequency**) to 1.
- The TouchProbe is activated through terminal X5/E5 (setpoint pulse).



Tip!

Touch probe initiators can have response delay times that cause a speed-dependent phase-angle offset.

Code C0429 can be used to set a correction value (-32767 incr. ... 32767 incr.) for this phase-angle offset.

Formula for correction value at C0429 and C0910:

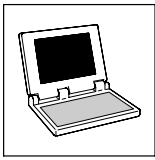
Correction value at C0429/C0910 = 16384 * correction value

- Please obtain the values from the data sheet of the initiator, or contact the manufacturer.



Stop!

If the synchronization via terminals X5/E4 and X5/E5 (C0532 = 2), make sure that no other control signals are taken from these terminals.



Function block library *LenzeElectricalShaft.lib*

Special functions

Digital frequency processing (*L_DFSET*)

A TouchProbe or zero-pulse has occurred:

If a TouchProbe or zero-pulse has appeared at terminal X5/E4 or X5/E5, then the instance of *L_DFSET* receives this information through

- *bSetTPReceived_b* (setpoint of the SB **DF_IN_DigitalFrequency**)
and
- *bActTPReceived_b* (actual value of the SB **MCTRL_MotorControl**)
the next time that the task is called in which the instance of *L_DFSET* is processed.

The phase-angle difference between the appearance of the TouchProbe/zero-pulse signal and the start of this task is assigned to *dnActTPLastScan_p* and *dnSetTPLastScan_p* respectively. The synchronization can be made to be precisely in phase, with the help of this information.

If the synchronization has been successfully concluded, then *L_DFSET1.bAck_b* is set to TRUE.

Synchronization mode

For the synchronization, different modes are available. The setting is made through *by0PulsFunc*.

by0PulsFunc	Synchronization mode	Note
0	inactive	function inactive
1	continuous synchronization with correction in the shortest possible way	The synchronization starts after 2 zero-pulses are received.
2	single synchronization with correction in the shortest possible way	Set <i>b0Pulse_b</i> = TRUE for a single zero-pulse synchronization. The synchronization starts after 2 zero-pulses are received.
10	single synchronization, a phase deviation is corrected in the shortest possible way	
11	single synchronization, a phase deviation is corrected in CW direction	
12	single synchronization, a phase deviation is corrected in CCW direction	
13	single synchronization, a phase difference is determined between setpoint pulse and actual pulse and is corrected to the corresponding direction of rotation according to the sign	Set <i>b0Pulse_b</i> = TRUE for a single zero-pulse synchronization.

2.1.2.4 Calculation of a phase-angle difference

With the aid of a set/actual phase-angle comparison, a phase-angle difference can be determined, and corrected.

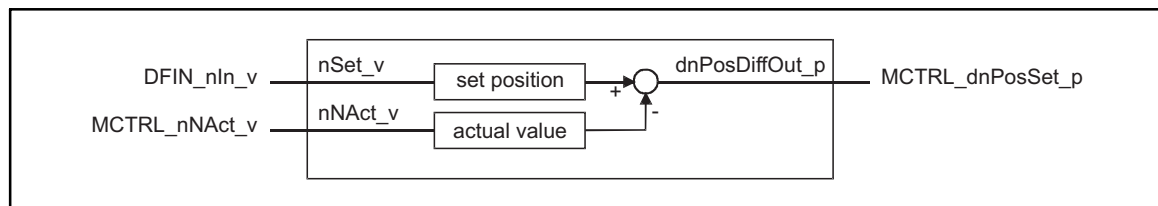
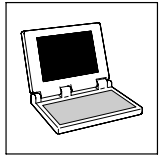


Abb. 2-8

Example of phase-angle difference determination for the 9300 Servo PLC, with *L_DFSET*

- In *nSet_v* (16384 $\equiv$ 15000 rpm) the setpoint speed is read in, and integrated into a path information by an integrator in *L_DFSET*. The processing time for the task, during which the instance of *L_DFSET* is called, is taken into account.
- The stretch and gearing factors, as well as offsets, can also be taken into account in the calculation of the setpoint speed.
- In *nNAct_v* (16384 $\equiv$ 15000 rpm) the actual speed is read in, and integrated into a path information by an integrator in *L_DFSET*. The processing time for the task, during which the instance of *L_DFSET* is called, is taken into account.



- In *dnPosDiffOut_p* the calculated phase-angle difference is given out, and can, if appropriate, be assigned to a phase-angle controller (e.g. for the system variables *MCTRL_dnPosSet_p* of the 9300 Servo PLC).
- In *dnPosSetOut_p* path information is given out from the setpoint integrator.
- The actual-value and setpoint integrators can be set to 0 by *bResetAllIntegrators_b* = TRUE, whereby *dnPosDiffOut_p* and *dnPosSetOut_p* are set to 0.
- By using *bSetIntegrator_b* = TRUE, the actual-value and setpoint integrators are set to be the same.
If a FALSE-TRUE transition occurs at *bSetIntegrator_b* then *dnPosDiffOut_p* is set = 0.
bSetIntegrator_b has a higher priority than *bResetAllIntegrators_b*.

2.1.2.5 Process-monitoring functions

Contouring (following) error (*bFollowingErr_b* = TRUE)

The monitoring reacts if the drive is not able to follow its set phase, because e.g.

- the centrifugal mass is too large for the set acceleration or deceleration time
- or
- the torque limit is reached (load torque > drive torque)

Remedy:

- Unload drive

or

- increase torque limit at the servo controller (if the power limits of the controller are not yet achieved)

The monitoring is derived from the phase difference of set-value integrator minus actual phase integrator. The comparison value (contouring error limit) *dwFollowingErrWin* can be set by a parameter variable.

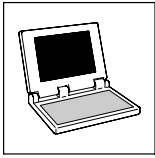
Phase-angle controller overflow (*bPosOverflow_b* = TRUE)

If this monitoring reacts, the phase deviation which can be represented internally, is exceeded. Homing points are lost.



Tip!

If a TRIP is to be triggered, then the status of the appropriate variable must be passed on to the SB **DCTRL_DriveControl**, so that an external TRIP can be triggered with its help.



Function block library *LenzeElectricalShaft.lib*

Special functions

Gearing compensation (*L_GEARCOMP*)

2.1.3 Gearing compensation (*L_GEARCOMP*)

This FB compensates for elasticity in the drive chain.

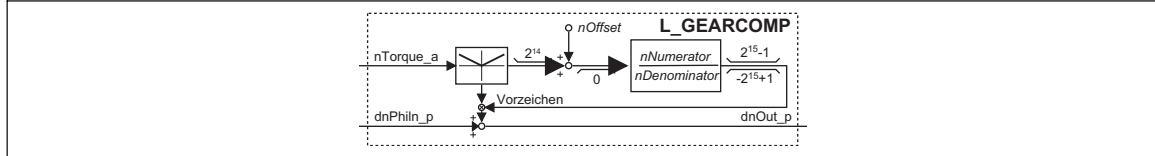


Abb. 2-9

Gearing compensation (*L_GEARCOMP*)

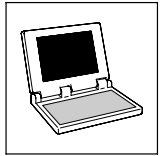
VariableName	DataType	SignalType	VariableType	Note
nTorque_a	Integer	analog	VAR_INPUT	Input value
dnPhin_p	Double Integer	position	VAR_INPUT	Input value
dnOut_p	Double Integer	position	VAR_OUTPUT	Initial value
nOffset	Integer	-	VAR CONSTANT RETAIN	Offset
nNumerator	Integer	-	VAR CONSTANT RETAIN	Numerator, dynamic switch-off at $nNumerator = 0$
nDenominator	Integer	-	VAR CONSTANT RETAIN	Denominator

Parameter codes of the instances

VariableName	<i>L_GEARCOMP1</i>			SettingRange	Lenze
nOffset	C1260			-16383 ... 16383	0
nNumerator	C1261			-32767 ... 32767	1
nDenominator	C1262			1 ... 32767	1

Functional sequence

1. The signal at *nTorque* is divided into the absolute value and the sign.
2. The value is converted (with *nNumerator*, *nDenominator*, *nOffset*).
3. The result is evaluated with the sign and added to the signal at *dnPhin_p*



2.1.4 Homing function (L_REF)

This FB is used to bring the drive shaft to a specific position.

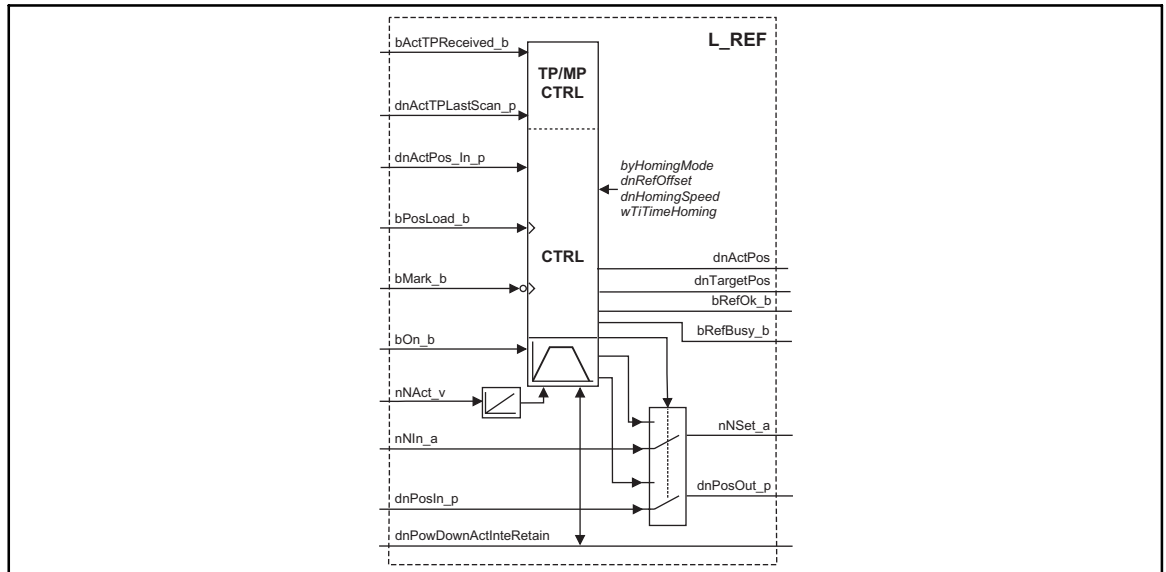
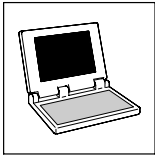


Abb. 2-10 Homing function (L_REF)

VariableName	DataType	SignalType	VariableType	Note
bActTPReceived_b	Bool	binary	VAR_INPUT	TouchProbe received
dnActTPLastScan_p	Double Integer	position	VAR_INPUT	Phase-angle difference between TouchProbe and start of task
dnActPos_In_p	Double Integer	position	VAR_INPUT	Loading value for the current position
bPosLoad_b	Bool	binary	VAR_INPUT	FALSE-TRUE edge = phase-angle at input <i>dnActPosIn_p</i> is loaded into REF-ACTPOS (start value)
bMark_b	Bool	binary	VAR_INPUT	Home position switch
bOn_b	Bool	binary	VAR_INPUT	TRUE = start homing
nNAct_v	Integer	velocity	VAR_INPUT	Actual speed, for calculating the actual position (16384 $\equiv$ 15000 rpm)
nNIn_a	Integer	analog	VAR_INPUT	Speed setpoint in [%] of n_{max} (C0011) (C0011 $\equiv$ 100 % $\equiv$ 16384)
dnPosIn_p	Double Integer	position	VAR_INPUT	<i>dnPosIn_p</i> is switched through to <i>dnPosOut_p</i> after homing
dnActPos	Double Integer	position	VAR_OUTPUT	Actual position of the calculated position profile
dnTargetPos	Double Integer	position	VAR_OUTPUT	Target position of the calculated position profile
bRefOk_b	Bool	binary	VAR_OUTPUT	TRUE = homing completed/reference point known
bRefBusy_b	Bool	binary	VAR_OUTPUT	TRUE = homing function active
nNSet_a	Integer	analog	VAR_OUTPUT	Speed setpoint for n-controller
dnPosOut_p	Double Integer	position	VAR_OUTPUT	Phase setpoint (contouring error for phase controller in FB MCTRL)
dnPowDownActInte Retain	Double Integer	-	VAR_IN_OUT	For Mode 21, to save the position after switching off the supply power
byHomingMode	Byte	-	VAR CONSTANT RETAIN	Setting for homing modes
dnRefOffset	Double Integer	-	VAR CONSTANT RETAIN	Setting for homing offset (1 $\equiv$ 1 incr.)
dnHomingSpeed	Double Integer	-	VAR CONSTANT RETAIN	Setting for homing speed (286331153 $\equiv$ 16000 rpm)
wTiTimeHoming	Word	-	VAR CONSTANT RETAIN	Setting for ramp time (100 $\equiv$ 1 s)



Function block library *LenzeElectricalShaft.lib*

Special functions

Homing function (*L_REF*)

Parameter codes of the instances

VariableName	L_REF1			SettingRange	Lenze
byHomingMode	C0932			0, 1, 8, 9, 20, 21	0
dnRefOffset	C0934			-2140000000 ... 2140000000 inc	0
dnHomingSpeed	C0935			0 ... 16000 rpm	100
wTiTimeHoming	C0936			0.01 ... 650.00 s	1.00



Note!

Use the FB *L_REF* only in tasks with an interval time in the range from 1 ... 10msec!

Range of functions

- Profile generator
- Homing modes
- Control via input signals
- Output of status signals
- Store the present output value after power interruption
- Connection of the function block

2.1.4.1 Profile generator

The homing speed profile can be adapted to the application.

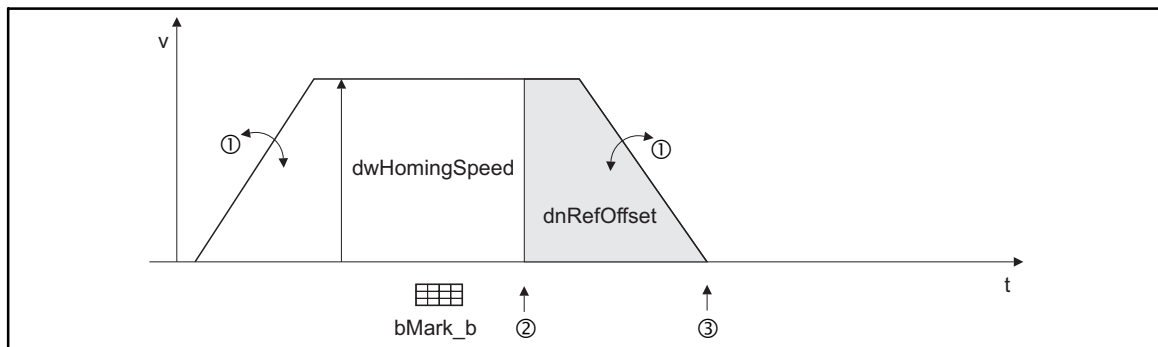
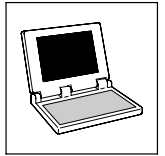


Abb. 2-11

Homing speed profile

- ① *wTiTimeHoming*
- ② Zero pulse
- ③ Home position

Variable	Meaning	Note
dnRefOffset	Homing point offset = number of increments after the zero pulse	The reference is: 65536 inc = 1 revolution. An input of up to 2140000000 incr. is possible
dnHomingSpeed	Maximum traversing speed	Entry in [rpm]. (286331153 $\approx$ 16000 rpm)
wTiTimeHoming	Ramp-up/down time	linear ramp (100 $\equiv$ 1 s)



The profile generator calculates the speed profile from the set profile parameters.

- You can make parameter changes during the homing run.
 - *dnHomingSpeed* and *wTiTimeHoming* become effective when *bOn_b* is set = FALSE
- Do not operate the drive at the torque limit (*MCTRL_bMMax_b* = TRUE), otherwise the drive cannot follow the velocity profile.
 - Increase the up/down time *wTiTimeHoming* until *MCTRL_bMMax_b* no longer responds.
- The phase controller in the SB MCTRL must be switched to active.

Note for use of the FB L_REF

The L_REF function block calculates the path by means of a $v(t)$ profile:

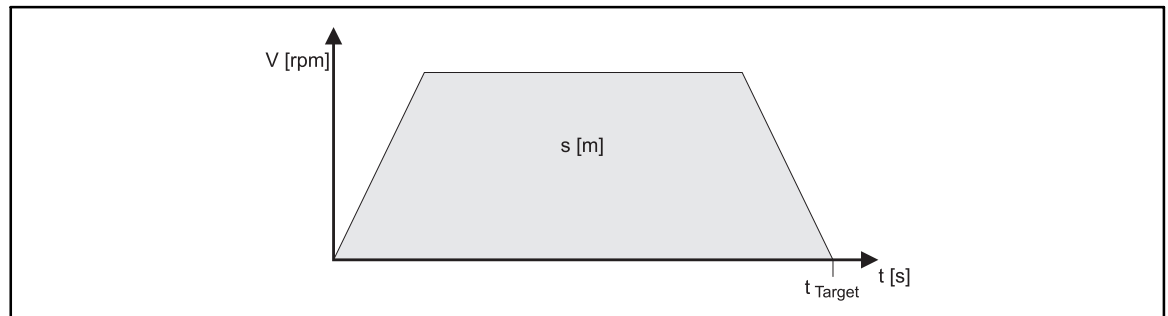
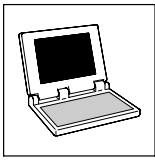


Abb. 2-12

$v(t)$ profile

- Through the input *nNAct_v*, the actual speed can be read in, and the actual position can be calculated. From this calculation, via the phase-angle controller (with *MCTRL_bPosOn_b* = TRUE and if *L_REF.dnPosOut_p* is linked to *MCTRL_dnPosSet_p*) the position can be corrected, if a deviation occurs. The correction is made up to the time t_{Target} . If the deviation is so large that the phase-angle controller cannot correct it within the time t_{Target} , then a contouring error evaluation must be performed, with the aid of the FB L_DFSET.
- The calculated position profile operates the motor control through the output *L_REF.dnPosOut_p.nNSet_a* (the position profile is followed in open-loop control, not closed-loop).
When the calculated position profile has reached *L_REF.dnTargetPos_p*, the homing is finished, and *L_REF.bRefOk_b* is set =TRUE and *L_REF.bRefBusy_b* is set =FALSE.
- The output *L_REF.dnActPos_p* indicates the actual position that L_REF has calculated, through *L_REF.nNAct_v*.
- The evaluation of a Touch Probe or zero-pulse can be made through the TP/MP-CTRL interface (homing modes 0, 1, 8 and 9).
- When the end position has been reached, the internal switch of the FB L_REF changes over, and the values from the inputs *L_REF.nNIn_a* and *L_REF.dnPosIn_p* are connected through to the outputs *L_REF.nNSet_a* and *L_REF.dnPosOut_p*.
- If, during homing, the input *bOn_b* changes to FALSE, then the homing/profile is stopped, and then continued after a fresh *bOn_b*=TRUE.
- The position profile will only be processed if the drive controller is enabled.



Function block library *LenzeElectricalShaft.lib*

Special functions

Homing function (*L_REF*)

2.1.4.2 Homing Modes

The home position is defined by:

- the homing mode *byHomingMode*
- the reference point offset *dnRefOffset*



Note!

When using feedback from a resolver, the zero position is used, not the zero-pulse (depending on the resolver attached to the motor), and, for homing with the TouchProbe, the TouchProbe phase-angle.

Homing with a reference switch to zero pulse (MP)/zero position or TouchProbe (TP) via the TP/MP-CTRL interface

The reference point is situated after the TRUE-FALSE edge of the reference switch *bMark_b*, at the next zero-pulse/zero position, or TP plus the reference mark offset:

- It depends on the setting of the system block that provides the actual speed, whether homing is made to zero-pulse/zero position or TouchProbe:
 - With SB **MCTRL_MotorControl** the setting is made through C0911 (0=MP, 1=TP via E4).
 - With SB **DF_IN_DigitalFrequency** the setting is made through C0428 (0=MP, 1=TP via E5).
 - For further information, see the description for SB **MCTRL_MotorControl** and SB **DF_IN_DigitalFrequency** in the Manual “9300 Servo PLC”.
- Mode 0 (*byHomingMode* = 0):
 - Move to the home position in CW rotation.
 - Home position offset *dnRefOffset* is entered as positive.

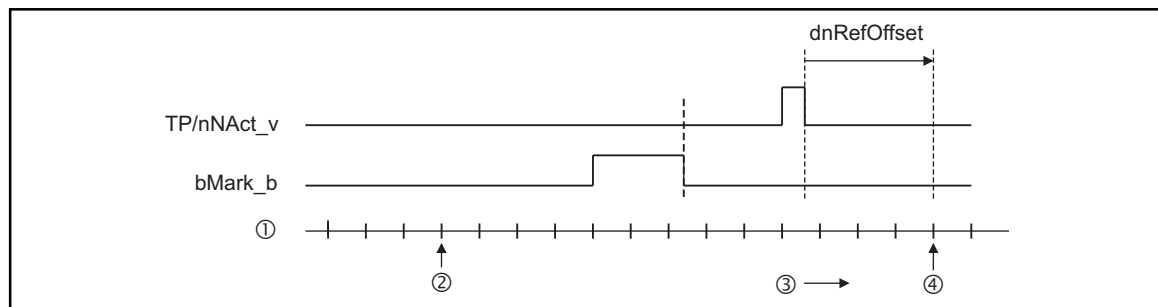
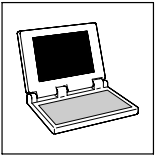


Abb. 2-13

Homing with zero-pulse/zero position or TouchProbe; moving to the reference mark in the cw direction of rotation

- ① Distance
- ② Start position
- ③ Direction
- ④ Home position



- Mode 1 (*byHomingMode* = 1):
 - Move to the home position in CCW rotation.
 - Home position offset *dnRefOffset* is entered as negative.

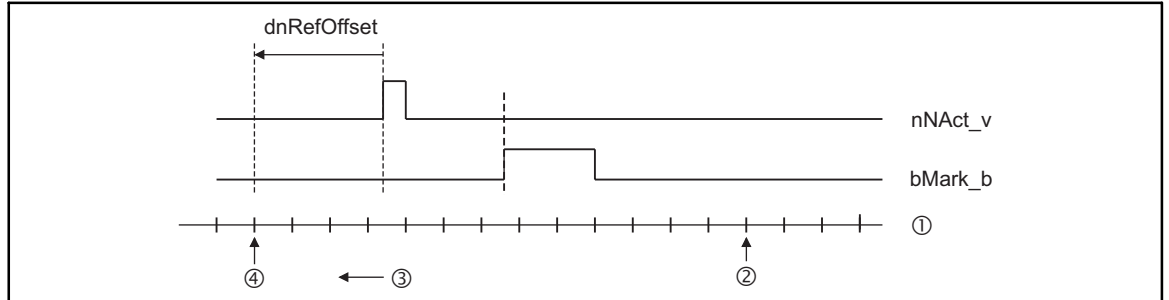


Abb. 2-14

Homing with zero-pulse/zero position or TouchProbe; moving to the reference mark in the ccw (left) direction of rotation

- ① Distance
- ② Start position
- ③ Direction
- ④ Home position

Function for homing with a reference switch to zero-pulse (MP)/zero position via the TP/MP-CTRL interface

- In this case, the reference offset is set to *dnRefOffset* = 0.
- If MP is known, the drive moves back to the last zero pulse after a negative edge of *L_REF.nMark_b*.
- If the drive starts from a position on an MP, after *L_REF.nMark_b* = TRUE, then it moves to the next MP.

This applies to Modes 0 and 1, with homing to zero-pulse (MP)/zero position, using a reference switch.

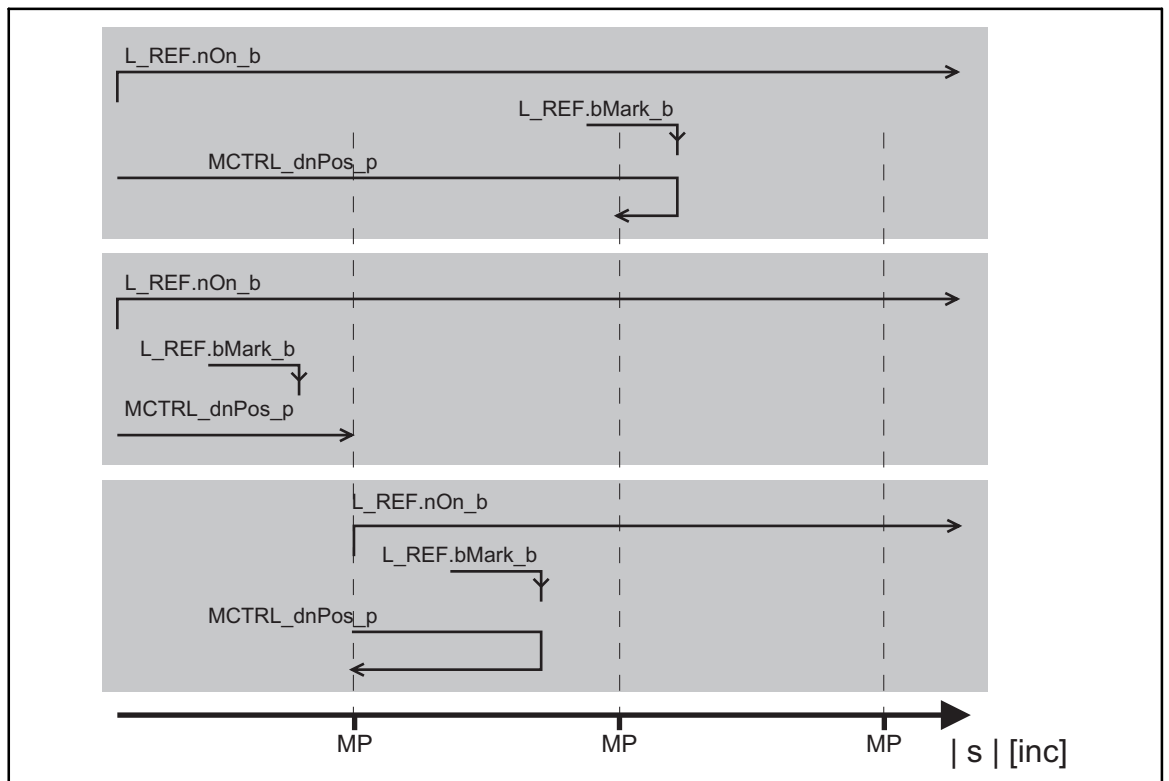
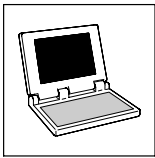


Abb. 2-15

Function for homing with a reference switch to zero-pulse / zero position



Function block library *LenzeElectricalShaft.lib*

Special functions

Homing function (*L_REF*)



Note!

If the drive is to stop at the next zero-pulse after detecting the mark, then an offset must be assigned, under *dnRefOffset* .

- Mode 0: *dnRefOffset* = 65536 ($\equiv$ 1 motor turn)
- Mode 1: *dnRefOffset* = -65536 ($\equiv$ 1 motor turn)

Function for homing with a reference switch and TouchProbe (TP)

- In this case, the reference offset is set to *dnRefOffset* =0.

This applies to Modes 0 and 1, for homing with a reference switch and TouchProbe

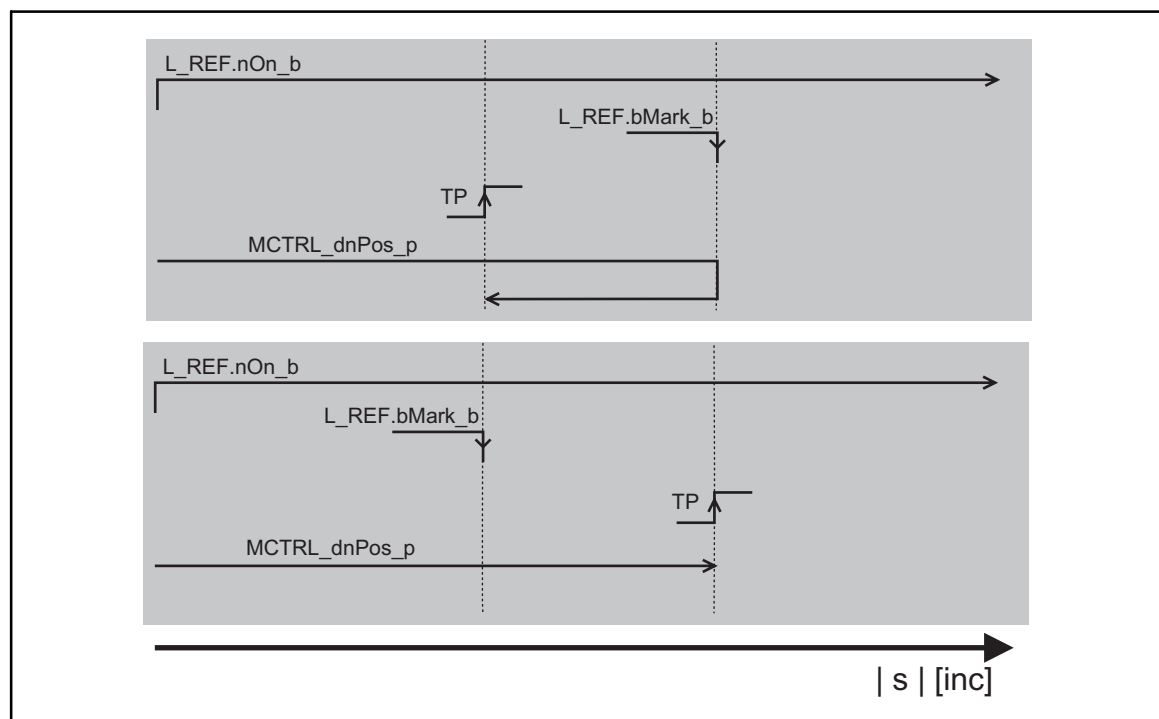
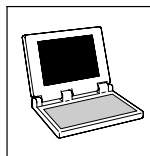


Abb. 2-16

Function for homing with a reference switch and Touch-Probe



Homing with TouchProbe (TP) or zero-pulse (MP)

The reference point is at the next TouchProbe signal, or zero-pulse plus the reference point offset.

- Mode 8 (*byHomingMode* = 8):
 - Move to the home position in CW rotation.
 - Home position offset *dnRefOffset* is entered as positive.
- Mode 9 (*byHomingMode* = 9):
 - Move to the home position in CCW rotation.
 - Enter the reference point offset *dnRefOffset* or zero-pulse as a negative value.

Take care that the TP has been parameterized in the system block!

Example: Mode 8 with a 9300 Servo PLC through MCTRL_MotorControl

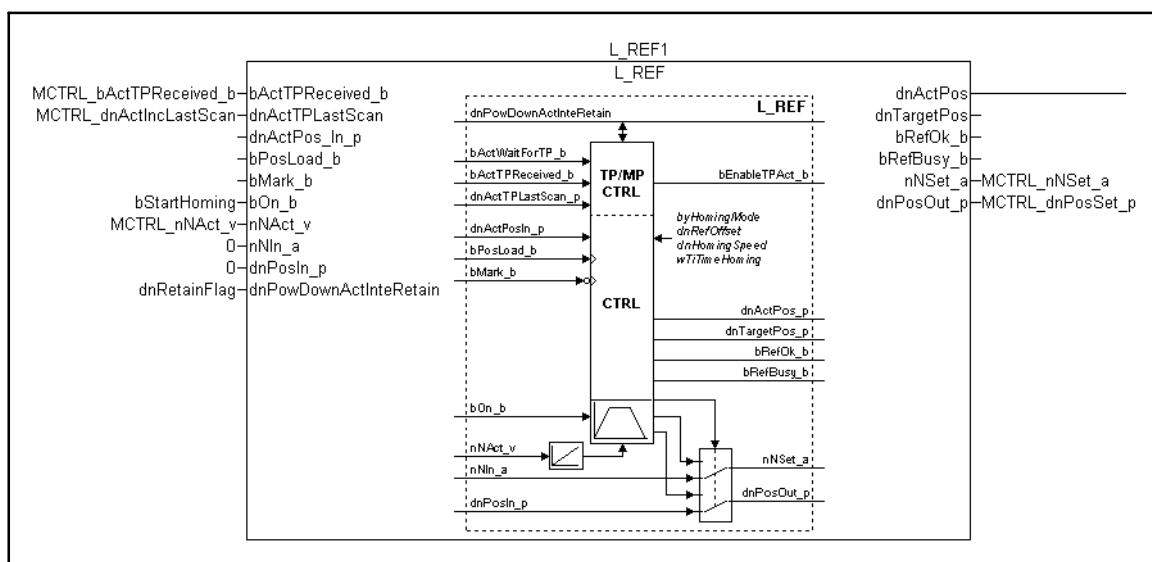
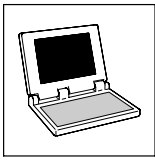


Abb. 2-17

Example: Mode 8 with a 9300 Servo PLC through MCTRL_MotorControl

Settings

- In Parameter Manager: *byHomingMode* = 8
- C0911=0, homing to zero-pulse or
C0911=1, homing to TouchProbe (terminal X5/E4).
- With C0911=1, C0912 can be used to select whether a rising or falling edge will trigger a TouchProbe.
(C0912 = 0: rising edge; C0912 = 1: falling edge)
- With C0910 a delay can be set for a TP/MP.



Function block library *LenzeElectricalShaft.lib*

Special functions

Homing function (*L_REF*)

Direct homing

The home position is on the home position offset.

- Mode 20 (*byHomingMode* = 20):
 - The drive moves immediately after activation (*bOn_b* = TRUE) from the present position to the reference point.
 - You can load the actual position value before, with the input value *dnActPosIn_p* (2-24)
 - The distance and direction results from the actual position and the set reference point offset. *dnRefOffset*.

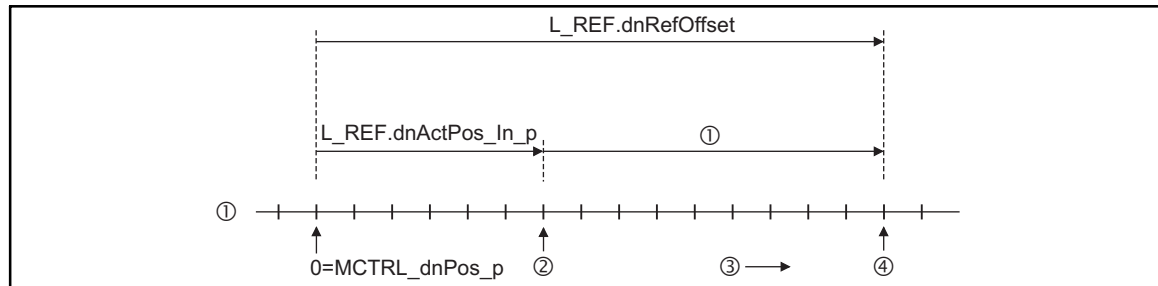


Abb. 2-18

Direct homing; move to the home position in CW rotation

- ① Distance
- ② Start position
- ③ Direction
- ④ Home position

- Mode 21 (*byHomingMode* = 21), as mode 20, but also:
 - The actual position is saved during power disconnection, and is loaded again at power-on (set C2104 = 1).



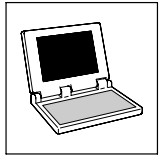
Note!

One turn = 2^{16}

2.1.4.3

Control via input signals

- *bOn_b* = FALSE-TRUE edge starts the homing:
 - The input must remain TRUE until homing is completed. Homing is canceled if the input is set to LOW before the reference point is reached. *bOn_b* is set = FALSE.
- *bOn_b* = FALSE interrupts the homing:
 - The drive is decelerated to zero speed along the preset ramp. *dwTiTimeHoming*
 - The inputs *nNIn_a* and *dnPosIn_p* are connected through to the outputs *nNSet_a* and *dnPosOut_p*
 - The state has no effect if homing is already completed (*bBusy_b* = FALSE).
- *bPosLoad_b* = FALSE-TRUE edge:
 - The profile generator accepts the phase applied at input *dnActPosIn_p* as a start value for the actual position value.
 - The function is effective only if *bOn_b* is set = FALSE
 - The function is effective only in modes 20 and 21.



Example

The motor should make a 10-turn (left/ccw rotation) relative positioning movement, and then stop.

Settings:

- *byHomingMode* = 20
- *L_REF.RefOffset* = 0

After operation of *DIGIN_bIn1_b* = TRUE, the motor rotates for 10 turns in a left/ccw direction.

- If the value *L_REF.dnActPosIn_p* is positive, the motor rotates left/ccw, if it is negative, the motor rotates in a cw direction!
- If a reference point offset is also defined then with a
 - positive offset, the relative position = *L_REF.dnActPosIn_p* - *L_REF.RefOffset*
 - negative offset, the relative position = *L_REF.dnActPosIn_p* + *L_REF.RefOffset*

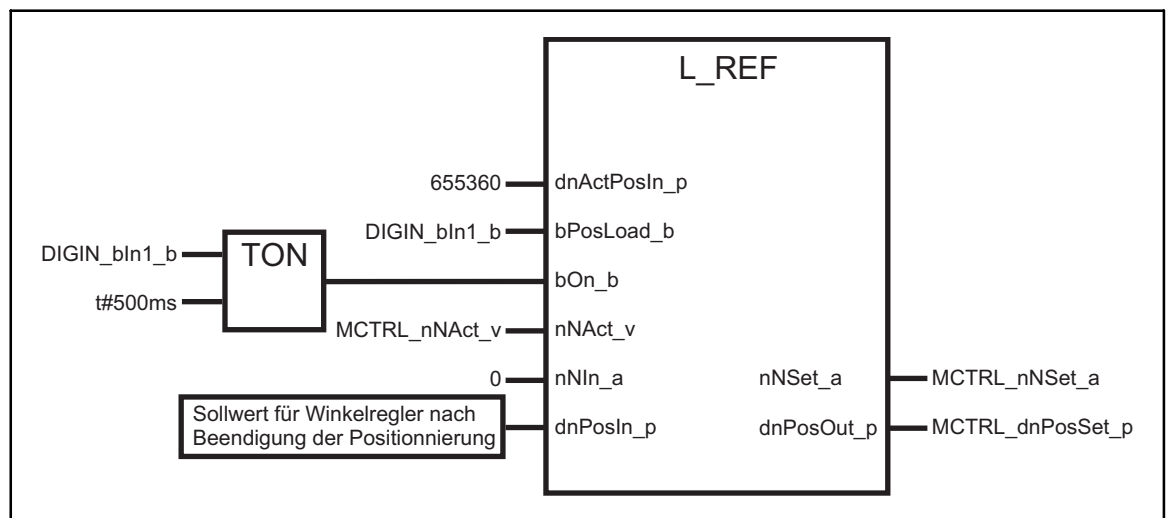
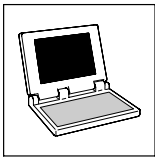


Abb. 2-19

Example of relative positioning
(Legend: Sollwert für Winkelregler nach = Setpoint for phase-angle controller at the end of positioning)

2.1.4.4 Output of status signals

- With *bBusy_b* = TRUE, the homing function is active:
 - The profile generator is connected to the outputs *dnPosOut_p* and *nNSet_a*
- With *bBusy_b* = FALSE, the homing function is not active (or finished):
 - The inputs *dnPosIn_p* and *nNIn_a* are connected through to the outputs *dnPosOut_p* and *nNSet_a*
- With *bRefOk_b* = TRUE, the homing run has been ended successfully:
 - Homing is completed if the setpoint of the profile generator has reached the home position.
 - Any possible contouring error is transmitted to the FB *L_DFSET* and compensated unless the drive is operating at the torque limit.
- With *bRefOk_b* = FALSE:
 - Homing is currently being carried out or
 - the reference point is no longer known, e.g. due to interference, or
 - homing was interrupted.



Function block library *LenzeElectricalShaft.lib*

Special functions

Homing function (*L_REF*)

2.1.4.5 Store the present output value after power interruption

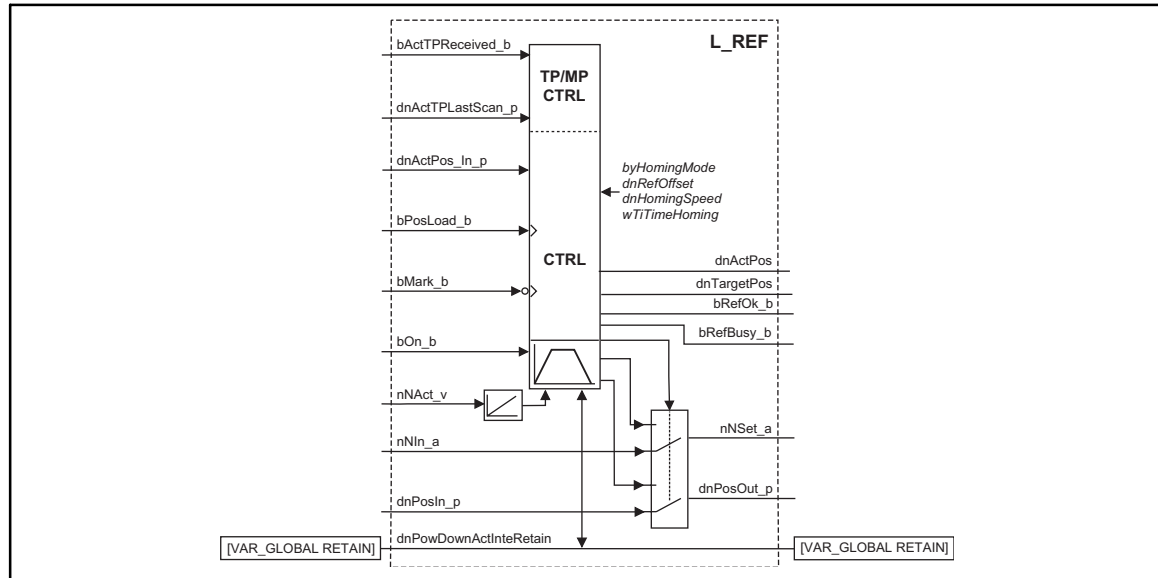


Abb. 2-20 Programming to store the present output value after a supply interruption

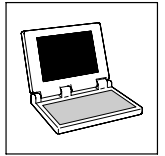
In order to store the latest value at `dnActPos_p` after a supply interruption, you must declare a global variable of type RETAIN (`VAR_GLOBAL RETAIN`). Link this variable as shown in Abb. 2-20

- In this variable, the present value is always stored at `dnActPos_p`. The variable will hold the value after a supply interruption.
- When the supply is switched on again, the stored value is read into the FB `L_REF` from the variable and applied as the starting value.



Note!

This only applies to Mode 21, no RETAIN-type variable needs to be defined for the other modes.



2.1.4.6 Linking the function block for zero-pulse/zero position or TP

- Important for Modes 0, 1, 8 and 9.
- *dnPosOut* supplies the set phase-angle for *nNSet_a* (contouring error) for the phase-angle controller in SB MCTRL.
- For correct homing, both signals (*dnPosOut_p* and *nNSet_a*) must be processed.
- The homing function must be linked to the FB L_DFSET.
(see, for example, signal flow diagram for the template “DFMasterInternal24VSupply_CFG5010FUP.lpc”).
- Otherwise, cumulative phase errors may occur.

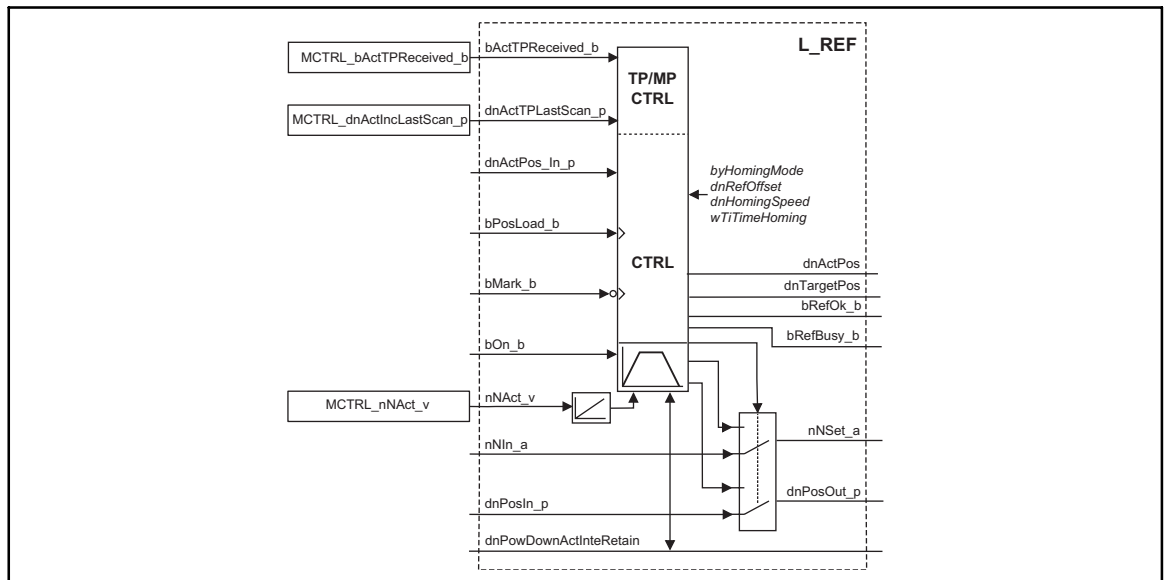
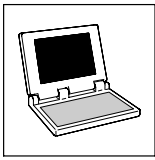


Abb. 2-21 Programming the FB L_REF for a precise phase-angle

The FB L_REF reads in the momentary speed signal at *nNAct_v*. Using TP/MP-CTRL, the actual phase-angle difference between a TouchProbe in SB **MCTRL_MotorControl** and the call of the FB L_REF is determined. In this way, a phase-angle deviation is acquired, corrected, and stored in FB L_REF.

After reading in the phase-angle difference at *dnActTPLastScan_p*, *bEnableTPAct_b* is set = TRUE, to start a new measurement in the SB. **MCTRL_MotorControl**.

Further information: (2-20 ff)



Function block library *LenzeElectricalShaft.lib*

Special functions

Homing function (L_REF)

2.1.4.7 Example: homing to a TouchProbe

Nature of the task:

- Homing is to be carried out with HomingMode = 0 (moving to the reference point with right/cw direction of rotation).
- The digital input X5/E4 (TouchProbe for the SB **MCTRL_MotorControl**) is to set the reference point.
- The homing speed is 1000 rpm, and the ramp-up time is 1 second.
- During and after the homing, any phase- angle difference between the setpoint and actual value is to be corrected.

Implementation:

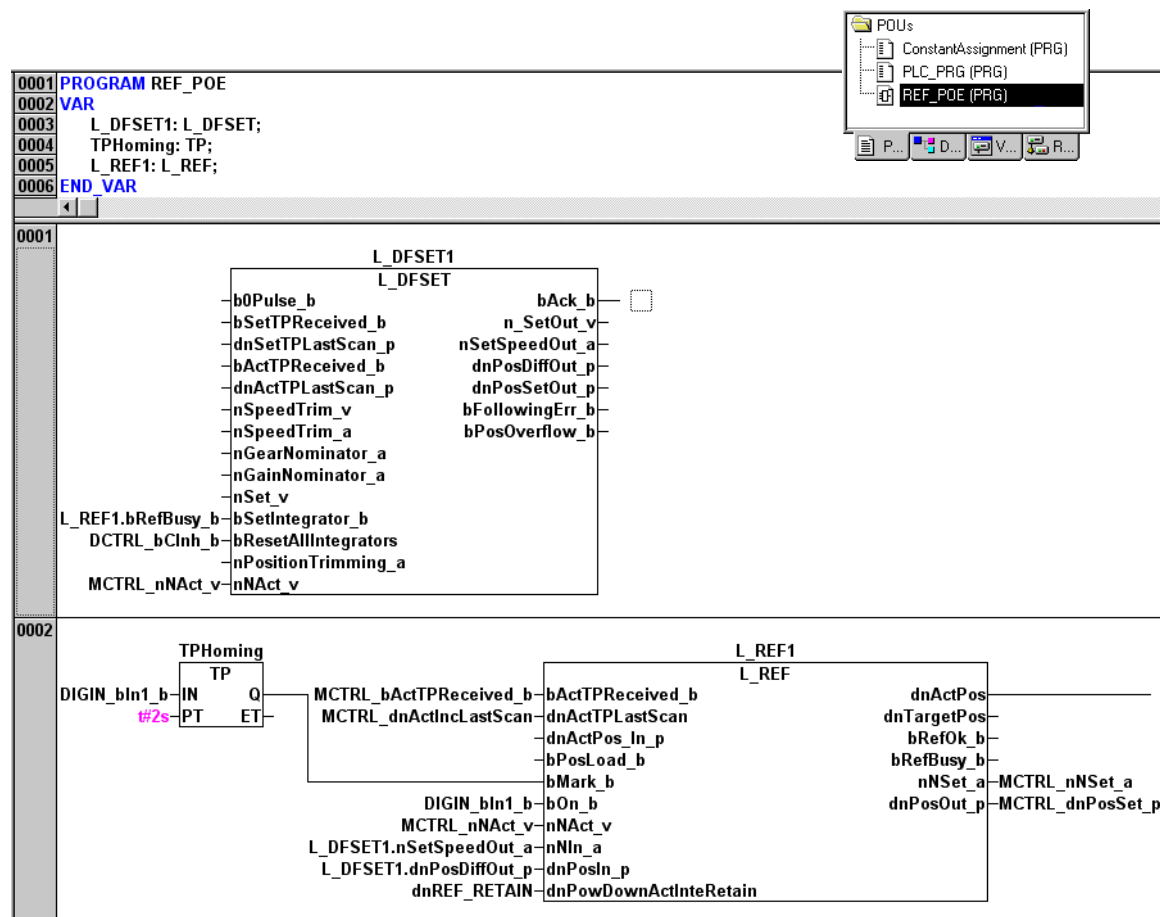


Abb. 2-22

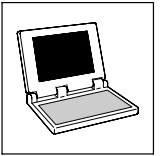
Programming the FB L_REF for homing to a TouchProbe

- Create one instance each of the FB **L_DFSET** and the FB **L_REF**.
- Assign the first-instance codes to these instances, using the Instance-Parameter Manager. In this way, all the values for the parameterization of the variables (e.g. for scaling and initialization) will be set to the Lenze standard settings.



Note!

You can find the Lenze standard settings for the instances of an FB in the table “Parameter Codes for the Instances” in the corresponding FB description.



Setting the initial values of *L_REF1* in the Instance-Parameter Manager

- Alter the initial value of *dnHomingSpeed* (maximum traversing speed) to 1000 rpm (286331153 corresponds to 16000 rpm).
- The initial value of *byHomingMode* keeps the Lenze standard setting (0 = move to the home position in CW rotation).
- The initial value of *wTiTimeHoming* keeps the Lenze standard setting (100 = 1 second ramp up/down time).

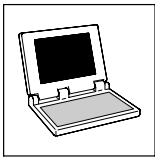
Function:



Note!

The phase-angle controller in the SB **MCTRL_MotorControl** must be activated. (*MCTRL_bPosOn_b* = TRUE).

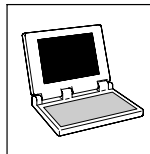
- The homing mode is started via *DIGIN_bIn1_b*.
- The speed setpoint is calculated by the ramp generator, and given out by *L_REF1.nNset_a*
- The ramp-up (acceleration) is made in accordance with the ramp-up time (*L_REF1.wTiTimeHoming*) and the traversing speed that has been set for homing (*L_REF1.dnHomingSpeed*).
- After 2 seconds, the marker (*L_REF1.bMark_b*) is set to FALSE by *L_REF1*. Now, a TouchProbe signal can be set by *L_REF1.bActTPReceived_b*.
- The TouchProbe is triggered via the digital input X5/E4 (TouchProbe of the SB **MCTRL_MotorControl**).
- *MCTRL_dnActIncLastScan* assigns the phase difference between the appearance of the Touchprobe signal and the start of the task in which *L_REF1* is processed to *L_REF1* for the calculation of the path.
- During homing, the phase-angle difference between the setpoint (ramp generator) and the actual value (*L_REF1.nNAct_v*) is calculated.
- The phase-angle difference is output via *L_REF1.dnPosOut_p*, and then assigned to the phase-angle controller in the SB **MCTRL_MotorControl** through the system variable *MCTRL_dnPosSet_p*.
- After homing, *L_DFSET1* calculates the phase-angle difference, and assigns the value to *L_REF1.dnPosIn_p*. The value is switched from *L_REF1.dnPosIn_p* through to *L_REF1.dnPosOut_p*.
- The setpoint/actual-value comparison is calculated in *L_DFSET1* from *L_DFSET1.nSet_v* (setpoint) and *L_DFSET1.nNAct_v* (actual value) and given out via *L_DFSET1.dnPosDiffOut_p*



Function block library LenzeElectricalShaft.lib

Special functions

Homing function (L_REF)



3 Appendix

3.1 Code table

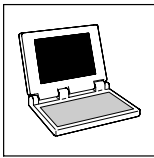
How to read the code table:

Column	Abbreviation	Meaning
Code	C0039	Code C0039
	1	Subcode 1 of code C0039
	2	Subcode 2 of code C0039
	...	...
	14	Subcode 14 of code C0039
	15	Subcode 15 of code C0039
	[C0156]	Parameter value of the code can only be modified when the controller is inhibited
LCD		Keypad LCD - DIS: ... display only - all others are parameter values
Lenze		Factory setting of the code
	*	The column "Important" contains further information
Choice	1 {1 %} 99	Minimum value {smallest step/unit} maximum value
IMPORTANT	-	Additional, important explanation of the code

3.1.1 L_DFRFG

FB description: (2-1)

Code	LCD	Possible settings		IMPORTANT
		Lenze	Choice	
C0750	WVpDenom	16	1 $V_p = 1$ 2 $V_p = 1/2$ 4 $V_p = 1/4$ 8 $V_p = 1/8$ 16 $V_p = 1/16$ 32 $V_p = 1/32$ 64 $V_p = 1/64$ 128 $V_p = 1/128$ 256 $V_p = 1/256$ 512 $V_p = 1/512$ 1024 $V_p = 1/1024$ 2048 $V_p = 1/2048$ 4096 $V_p = 1/4096$ 8192 $V_p = 1/8192$ 16384 $V_p = 1/16384$	Denominator for gain of L_DFRFG1
C0751	dwTir	1.000	0.000 {0.001 sec} 999.900	Acceleration time T_{ir} of L_DFRFG1
C0752	nMaxSpeed	3000	1 {1 rpm} 16000	Maximum catch-up speed of L_DFRFG1
C0753	dwTifForQSP	0.000	0.000 {0.001 sec} 999.900	Deceleration time T_{if} for QSP of L_DFRFG1
C0754	dwThresholdFollowingError	2 · 10 ⁹	10 {1 inc} 2000000000	Contouring error of L_DFRFG1 • 1 turn = 65535 inc
C0755	wSynWindow	100	0 {1 inc} 65535	Synchronization window of L_DFRFG1
C0756	dnOffset	0	-1000000000 {1 inc} 1000000000	Offset of L_DFRFG1 • 1 turn = 65535 inc
C0757	byFunction	0	0 no TP start 1 with TP start	
C0766	bySpeedDirection	1	1 cw / ccw 2 cw 3 ccw	Rot. direction selection L_DFRFG1 Rotation CW/CCW CW rotation CCW rotation



Function block library LenzeElectricalShaft.lib

Appendix L_DFSET

3.1.2 L_DFSET

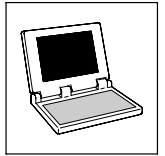
FB description: (□ 2-8)

Code	LCD	Possible settings		IMPORTANT
		Lenze	Choice	
C0033	nGearDenominator	1	1 {1} 32767	Gearing factor (denominator) of L_DFSET1
C0252	dnPosOffset	0	-245760000 {1 inc} 245760000	Phase-angle offset of L_DFSET1 - Fixed phase offset for digital frequency configuration - 1 turn = 65536 inc
C0253	nSpeedDependPosOffset	0	-32767 {1 inc} 32767	Phase-angle trimming of L_DFSET1 - Speed-dependent phase trimming - Depending on C0025. The change of C0025 resets C0253 to the Lenze default setting. 1 turn = 65536 inc - C0253 is reached at 15000 rpm
C0255	dwFollowingErrWin	327680	10 {1 inc} 1800000000	Contouring-error limit for error "P03" of L_DFSET1 1 turn = 65536 inc - Contouring error > C0025 triggers fault "P03"
C0529	nAngelOffsetMult	1	-20000 {1} 20000	Offset multiplier of L_DFSET1
C0530	bWithoutGearfactor	1	0 with g factor 1 no g factor	Evaluation of the setpoint integrator of L_DFSET1 (with/without gearing factor)
C0531	wAct0PulseDiv	0	0 {1} 16384	Actual zero-pulse divider of L_DFSET1
C0533	wGainDenominator	1	1 {1} 32767	Gain factor (denominator) of L_DFSET1
C0534	by0PulseFunc	0	0 Inactive 1 Continuous 2 Cont. switch 10 Once, fast way 11 Once, CW 12 Once, CCW 13 Once, 2*0-puls	Zero-pulse function of L_DFSET1
C0535	wSet0PulseDiv	0	0 {1} 16384	Set zero-pulse divider of L_DFSET1

3.1.3 L_GEARCOMP

FB description: (□ 2-16)

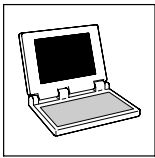
Code	LCD	Possible settings		IMPORTANT
		Lenze	Choice	
C1260	nOffset	0	-16383 {1} 16383	Offset of L_GEARCOMP
C1261	nNumerator	1	-32767 {1} 32767	Numerator of L_GEARCOMP
C1262	nDenominator	1	1 {1} 32767	Denominator of L_GEARCOMP



3.1.4 L_REF

FB description: (□ 2-17)

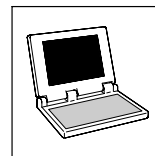
Code	LCD	Possible settings			IMPORTANT
		Lenze	Choice		
C0932	byHomingMode	0	0	Mode 0	Homing mode of L_REF1
			1	Mode 1	
			6	Mode 6	
			7	Mode 7	
			8	Mode 8	
			9	Mode 9	
			20	Mode 20	
			21	Mode 21	
C0934	dnRefOffset	0	-2140000000	{1 inc} 2140000000	Reference point offset of L_REF1
C0935	dnHomingSpeed	100	0	{1 rpm} 16000	Speed for homing mode of L_REF1
C0936	dwTiTimeHoming	1.00	0.01	{0.01 sec} 990.00	T _i -time homing of L_REF1 • Tir and Tif are identical



Function block library LenzeElectricalShaft.lib

Appendix

L_REF



4 Index

A

Appendix, 3-1

C

Code table, 3-1

Codes

- Digital frequency processing (L_DFSET) , 3-2
- Digital frequency ramp generator (L_DFRFG), 3-1
- Gearing compensation (L_GEARCOMP), 3-2
- Homing function (L_REF), 3-3

D

data-type entry, Explanation of, 1-4

Definitions, 1-2

Digital frequency processing (L_DFSET) , 2-8

Digital frequency ramp generator (L_DFRFG), 2-1

F

Function blocks

- Digital frequency processing (L_DFSET) , 2-8
- Digital frequency ramp generator (L_DFRFG), 2-1
- Gearing compensation (L_GEARCOMP), 2-16
- Homing function (L_REF), 2-17

G

Gearing compensation (L_GEARCOMP), 2-16

H

Homing function (L_REF), 2-17

- Homing modes, 2-20
- Profile generator, 2-18

Homing modes, 2-20

I

identifier, Explanation of, 1-4

L

L_DFRFG, 2-1

L_DFSET, 2-8

L_GEARCOMP, 2-16

L_REF, 2-17

Lenze software guidelines, Hungarian Notation, 1-3

M

Monitoring functions

- Phase-angle controller overflow, 2-15
- Second contouring error, 2-15

P

Phase-angle controller overflow, 2-15

prefix, Explanation of, 1-3

Profile generator, 2-18

S

Safety information, Layout

- Other notes, 1-1
- Warning of damage to material, 1-1

Second contouring error, 2-15

Signal type, Explanation of, 1-5

System variables, Explanation of, 1-5

T

Type of variable, Identification, 1-4

V

Variable names

- Conventions, Hungarian Notation, 1-3
- Lenze software guidelines, Explanation of, 1-3

Version identifiers of the function library, 1-6