SIEMENS

SIMOTION
LPac Basic Equipment Module Library

Application Manual

Preface

Library description 1

Library structure 2

Integration 3

Function description 4

Alarm and error messages 5

Application description 6

Contact A

V2.1

Legal information

Warning notice system

This manual contains notices you have to observe in order to ensure your personal safety, as well as to prevent damage to property. The notices referring to your personal safety are highlighted in the manual by a safety alert symbol, notices referring only to property damage have no safety alert symbol. These notices shown below are graded according to the degree of danger.

⚠DANGER
indicates that death or severe personal injury **will** result if proper precautions are not taken.

⚠WARNING
indicates that death or severe personal injury **may** result if proper precautions are not taken.

⚠CAUTION
indicates that minor personal injury can result if proper precautions are not taken.

NOTICE
indicates that property damage can result if proper precautions are not taken.

If more than one degree of danger is present, the warning notice representing the highest degree of danger will be used. A notice warning of injury to persons with a safety alert symbol may also include a warning relating to property damage.

Qualified Personnel

The product/system described in this documentation may be operated only by **personnel qualified** for the specific task in accordance with the relevant documentation, in particular its warning notices and safety instructions. Qualified personnel are those who, based on their training and experience, are capable of identifying risks and avoiding potential hazards when working with these products/systems.

Proper use of Siemens products

Note the following:

⚠WARNING
Siemens products may only be used for the applications described in the catalog and in the relevant technical documentation. If products and components from other manufacturers are used, these must be recommended or approved by Siemens. Proper transport, storage, installation, assembly, commissioning, operation and maintenance are required to ensure that the products operate safely and without any problems. The permissible ambient conditions must be complied with. The information in the relevant documentation must be observed.

Trademarks

All names identified by ® are registered trademarks of Siemens AG. The remaining trademarks in this publication may be trademarks whose use by third parties for their own purposes could violate the rights of the owner.

Disclaimer of Liability

We have reviewed the contents of this publication to ensure consistency with the hardware and software described. Since variance cannot be precluded entirely, we cannot guarantee full consistency. However, the information in this publication is reviewed regularly and any necessary corrections are included in subsequent editions.

Siemens AG
Division Digital Factory
Postfach 48 48
90026 NÜRNBERG
GERMANY

℗ 06/2017 Subject to change

Copyright © Siemens AG 2017.
All rights reserved

Preface

General information

Note

The standard applications are not binding and do not claim to be complete regarding configuration, equipment or any eventuality which may arise. The standard applications do not represent specific customer solutions, but are only intended to provide support for typical tasks. You are responsible for the proper operation of the described products. These standard applications do not relieve you of your responsibility regarding the safe handling when using, installing operating and maintaining the equipment. By using these standard applications, you agree that Siemens cannot be made liable for possible damage beyond the mentioned liability clause. We reserve the right to make changes and revisions to these standard applications at any time without prior notice. In the case of any differences between the suggestions made in these standard applications and other publications from Siemens, such as catalogs, the contents of the other documentation have priority.

Warranty conditions, liability, and support

If the application has been made available free of charge, the following applies:

We do not provide a warranty for any of the information contained in this document.

All other rights and claims against Siemens AG irrespective of legal basis are excluded. In particular claims for damages against Siemens AG in the case of product outage, downtime, loss of profit, either directly, indirectly or consequential damage are excluded.

This does not apply when liability is compulsory by law, e. g. in the case of the Product Liability Act, premeditation, an act of gross negligence by superiors and managerial staff of Siemens AG or in cases of fraudulent concealment of defects.

This limitation of liability also applies to sub-contractors, suppliers, delegates, superiors and managerial staff of Siemens AG.

German law shall apply to this agreement for customers with head offices in Germany; Swiss law for customers with head offices outside Germany. Application of the United Nations Convention on Contracts for the International Sale of Goods as of 11.04.1980 (CISG) is excluded.

If the application has been made available against payment, the appropriate alternative applies for the respective business transaction:

- Alternative 1: (Internal business)

If nothing else has been negotiated, then the "Conditions for the supply and services in Siemens internal business" applies in the version that is valid at the time that the equipment is purchased.

- Alternative 2: (Domestic business of Siemens AG)

If nothing else was negotiated, the "General License Conditions for Software for Automation and Drives for Customers with a Registered Office in Germany" valid at the time of sale are applicable.

- Alternative 3: (Direct export business of Siemens AG)

If nothing else has been negotiated, then the "General License Conditions for Software Products for Automation and Drives for Customers with a Seat or Registered Office outside Germany", valid at the time of sale, are applicable.

It is not permitted to distribute or duplicate these application examples in any form including excerpts thereof without the express consent of Siemens Industry Sector.

Notice regarding export identification codes

AL: N

ECCN: N

About this document

Objective

All modules and blocks of the **LPac Basic Equipment Module Library** are described in this application manual. For engineering system details and the SIMOTION SCOUT functions, refer to the *SIMOTION SCOUT standard documentation*.

Note

This document does not claim to contain all details on devices in any version or to take all conceivable operational cases and applications into account.

Should you require further information or encounter specific problems not covered in enough detail for your field of application, please contact your local Siemens office.

Target group

This document is intended for programmers and commissioning engineers.

Table of contents

Preface **3**

1 Library description **11**

1.1 Application 11
1.1.1 Description 11
1.1.2 Field of application 14
1.1.3 System requirements 14
1.1.4 Additional required libraries 15

2 Library structure **17**

2.1 Structure of the LPac basic equipment module library 17

3 Integration **31**

3.1 Integration of the library into a SIMOTION project. 31

4 Function description **33**

4.1 General explanations 33

4.2 General axis functions 34
4.2.1 Enable/disable axis 35
4.2.2 Home axis 35
4.2.3 Jog axis 36
4.2.4 Axis positioning 37
4.2.5 Synchronize/desynchronize axis in gearing 38
4.2.6 Synchronizing/desynchronizing axis in camming 40
4.2.7 Activation of an offset in the active synchronous operation 43
4.2.8 Axis rapid stop 43

4.3 Machine master (leading axis) 45
4.3.1 FBLPacMasterCont - continuous master 46
4.3.1.1 Technological consideration 46
4.3.1.2 Module description 47
4.3.1.3 Integration of the module 66
4.3.2 FBLPacDelayMaster - drop time compensation of the vertical continuous master 68
4.3.2.1 Technological consideration 68
4.3.2.2 Module description 70
4.3.2.3 Embedding the modules 74
4.3.3 FBLPacMasterInt - intermittent master 76
4.3.3.1 Technological consideration 76
4.3.3.2 Module description 76
4.3.3.3 Integration of the module 84
4.3.4 FBLPacExtEncoder - external encoder 86
4.3.4.1 Technological consideration 86
4.3.4.2 Module description 87
4.3.4.3 Embedding the module 89

4.4 Product supply 91
4.4.1 FBLPacTappetChainGear - gearing supply chain 91

4.4.1.1 Technological consideration ... 91
4.4.1.2 Module description ... 91
4.4.1.3 Integration of the module ... 96
4.4.2 FBLPacTappetChainCam - supply chain camming ... 98
4.4.2.1 Technological consideration ... 98
4.4.2.2 Module description ... 98
4.4.2.3 Integration of the module ... 103
4.4.3 FBLPacDosingPos - dosing positioning axis ... 105
4.4.3.1 Technological consideration ... 105
4.4.3.2 Module description ... 106
4.4.3.3 Integration of the module ... 110
4.4.4 FBLPacDosingControl - dosing weigher discharge control ... 111
4.4.4.1 Technological consideration ... 111
4.4.4.2 Module description ... 117
4.4.4.3 Integration of the module ... 124
4.4.5 FBLPacNPNBDetection - product detection shift register ... 126
4.4.5.1 Technological consideration ... 126
4.4.5.2 Module description ... 127
4.4.5.3 Embedding the module ... 131
4.5 Foil transport ... 132
4.5.1 FBLPacFoilFeedPos - foil extractor positioning axis ... 132
4.5.1.1 Technological consideration ... 132
4.5.1.2 Module description ... 132
4.5.1.3 Integration of the module ... 139
4.5.2 FBLPacFoilFeedGear - foil extractor gearing ... 142
4.5.2.1 Technological consideration ... 142
4.5.2.2 Module description ... 142
4.5.2.3 Integration of the module ... 149
4.5.3 FBLPacFoilFeedCam - foil extractor camming ... 151
4.5.3.1 Technological consideration ... 151
4.5.3.2 Module description ... 152
4.5.3.3 Integration of the module ... 158
4.5.4 FBLPacFoilGuide - sidelay control ... 160
4.5.4.1 Technological consideration ... 160
4.5.4.2 Module description ... 162
4.5.4.3 Integration of the module ... 164
4.5.5 FBLPacSlipCompensation - slip compensation ... 165
4.5.5.1 Technological consideration ... 165
4.5.5.2 Module description ... 167
4.5.5.3 Integration of the module ... 168
4.5.6 FBLPacPrintmarkCorrection - print mark correction ... 170
4.5.6.1 Technological consideration ... 170
4.5.6.2 Module description ... 172
4.5.6.3 Integration of the module ... 179
4.5.7 FBLPacPrelimFoilFeed - foil pre-extractor ... 180
4.5.7.1 Technological consideration ... 180
4.5.7.2 Module description ... 182
4.5.7.3 Integration of the module ... 187
4.5.8 FBLPacDancerControl ... 189
4.5.8.1 Technological consideration ... 189
4.5.8.2 Module description ... 190
4.5.8.3 Integration of the module ... 196

4.6 Cross sealing 197
4.6.1 FBLPacCrossSealPos - cross-sealer position axis 197
4.6.1.1 Technological consideration 197
4.6.1.2 Module description 200
4.6.1.3 Integration of the module 207
4.6.2 FBLPacCrossSealCam - cross-sealer cam 209
4.6.2.1 Technological consideration 209
4.6.2.2 Module description 211
4.6.2.3 Integration of the module 216
4.6.3 FBLPacCrossSealChainBag – chain bag control 218
4.6.3.1 Technological consideration 218
4.6.3.2 Module description 221
4.6.3.3 Integration of the module 232
4.6.4 FBLPacStopperCam – cam switching 233
4.6.4.1 Technological consideration 233
4.6.4.2 Module description 238
4.6.4.3 Integration of the module 240

4.7 Cam calculation blocks 241
4.7.1 FBLPacCreaCamBoxMotCS - box motion cross-sealer 241
4.7.1.1 Technological consideration 241
4.7.1.2 Module description 248
4.7.1.3 Integration of the module 249
4.7.2 FBLPacCreaCamCylCS – rotary, cylindrical cross-sealer 251
4.7.2.1 Technological consideration 251
4.7.2.2 Module description 251
4.7.2.3 Integration of the module 259
4.7.3 FBLPacCreaCamCrankLin - slider crank with with constant linear velocity 261
4.7.3.1 Technological consideration 261
4.7.3.2 Module description 261
4.7.3.3 Embedding the module 264

4.8 Auxiliary functions 266
4.8.1 FBLPacTranslationToRotation, FBLPacRotationToTranslation - slider crank transformation 266
4.8.1.1 Technological consideration 266
4.8.1.2 Module description 267
4.8.1.3 Integration of the modules 270
4.8.2 FBLPacGetPosTime (Jerk) – calculate positioning time 272
4.8.2.1 Technological consideration 272
4.8.2.2 Module description 272
4.8.2.3 Integration of the module 274
4.8.3 FBLPacGetPosAtTime – calculate position 275
4.8.3.1 Technological consideration 275
4.8.3.2 Module description 275
4.8.3.3 Embedding the module 277
4.8.4 FBLPacGetTimeAtPosition – calculate motion time 278
4.8.4.1 Technological consideration 278
4.8.4.2 Module description 278
4.8.4.3 Embedding the module 280
4.8.5 FBLPacGetCycleTimeVI – calculate cycle time 281
4.8.5.1 Technological consideration 281
4.8.5.2 Module description 282

4.8.5.3 Embedding the module ... 284
4.8.6 FBLPacSetOverrideVI - override adjustment ... 285
4.8.6.1 Technological consideration ... 285
4.8.6.2 Module description ... 286
4.8.6.3 Embedding the module ... 287
4.8.7 FBLPacChangeTOConfigData – change the TO axis configuration data ... 288
4.8.7.1 Technological consideration ... 288
4.8.7.2 Module description ... 288
4.8.7.3 Integration of the module ... 291
4.8.8 FBLPacOutPutCamControl – control cam controller ... 292
4.8.8.1 Technological consideration ... 292
4.8.8.2 Module description ... 292
4.8.8.3 Embedding the module ... 295
4.8.9 FBLPacCamTrackControl – control cam track ... 296
4.8.9.1 Technological consideration ... 296
4.8.9.2 Module description ... 297
4.8.9.3 Embedding the module ... 301
4.8.10 FBLPacTorqueControl – torque and following error monitoring ... 302
4.8.10.1 Technological consideration ... 302
4.8.10.2 Module description ... 303
4.8.10.3 Embedding the module ... 306
4.8.11 FBLPacProductInSealDetection – product inclusion monitoring ... 308
4.8.11.1 Technological consideration ... 308
4.8.11.2 Module description ... 308
4.8.11.3 Embedding the module ... 312
4.8.12 FBLPacReadSinamicsMessages – read drive faults ... 313
4.8.12.1 Technological consideration ... 313
4.8.12.2 Module description ... 313
4.8.12.3 Embedding the module ... 316

4.9 Temperature control ... 317
4.9.1 FBLPacTOTempCtrl - SIMOTION technology object ... 317
4.9.1.1 Technological consideration ... 317
4.9.1.2 Module description ... 318
4.9.1.3 Integration of the module ... 322
4.9.2 FBLPacPIDTempCtrl – SIMOTION PID controller ... 323
4.9.2.1 Technological consideration ... 323
4.9.2.2 Module description ... 323
4.9.2.3 Integration of the module ... 329
4.9.3 FBLPacPWMCtrl - pulse width modulation ... 330
4.9.3.1 Technological consideration ... 330
4.9.3.2 Module description ... 330
4.9.3.3 Integration of the module ... 331

5 Alarm and error messages ... 333

5.1 Alarm and error messages ... 333

6 Application description ... 335

6.1 LPac project ... 335

6.2 Sample application of a horizontal tubular bagging machine with LPac ... 340

6.3 Application sample of a vertical continuous tubular bagging machine with LPac ... 346

6.4 Sample application of a vertical intermittent tubular bagging machine with LPac 350
6.5 Application sample for possible temperature controls with LPac blocks 353
A Contact 355
A.1 Contacts 355
A.2 Internet addresses 356

Library description 1

1.1 Application

1.1.1 Description

The objective and purpose of the **SIMOTION LPac Basic Equipment Module Library** is to reduce the development time when developing SIMOTION applications, in particular for packaging machines and for machine types with a master-slave axis structure.

The function blocks and functions can be used for this purpose in SIMOTION user programs included in the **SIMOTION LPac Basic Equipment Module Library**. The **LPac** project can be deployed to further reduce the programming effort and to illustrate the block use. This project contains not only the actual **LPac** library but also the other required libraries and blocks (see Section 1.1.4) as well as programs that can be taken as user example that illustrate the call and interconnection of the **LPac** function blocks (see Section 4.2). The **LPac** project also illustrates the use of further standards, such as message handling and operating mode management by programs. Prepared data structures and variables significantly reduce the effort for the user.

The created modules and blocks are intentionally kept simple to permit wide use. The modules serve as a simplified interface between the user program and the technology objects of SIMOTION. The user therefore does not require expert knowledge of the control of technology objects in SIMOTION and can control the required functionality via a PLCopen-compliant interface.

Based on the ISA88 hierarchy for packaging machines, the blocks for implementing specific base functions for equipment modules contained in the **LPac** library can be deployed. Consequently, the terms Basic Equipment Module (BEM) and Control Module (CM) have been introduced for the blocks contained in the **LPac** library and for the auxiliary functions, respectively.

Basically, the modules of the **LPac Basic Equipment Module Library** intended for the motion control encapsulate the **FBLMCBasicAxis** standard axis function block. For details, see the *SIMOTION Axis Function Block* Function Manual.

Unlike the standard axis function block, the implemented functionality and interface has been reduced to just one main functionality, i.e. whereas the standard axis function block covers almost all possible axis functionality from the simple drive axis through to gearing and camming, the LPac modules are restricted to one of the previously named main functions. In addition, depending on the equipment module, most required auxiliary functions, such as disable/enable the axis, homing, jogging and positioning, are implemented.

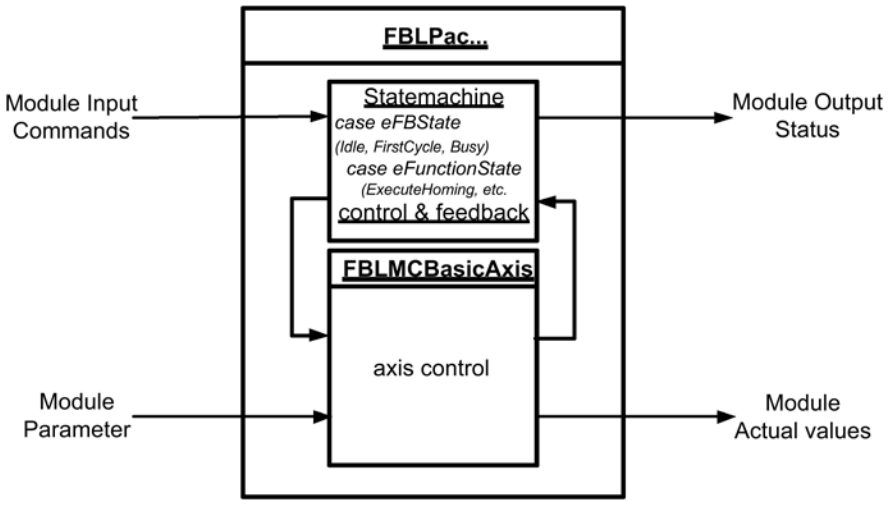


Figure 1-1 Structure of the LPac modules

LPac function blocks that control an axis are configured and structured internally as far as possible in the same way. Only the internal "eFBFunctionState" statemachine differs according to the respective block-specific functions.

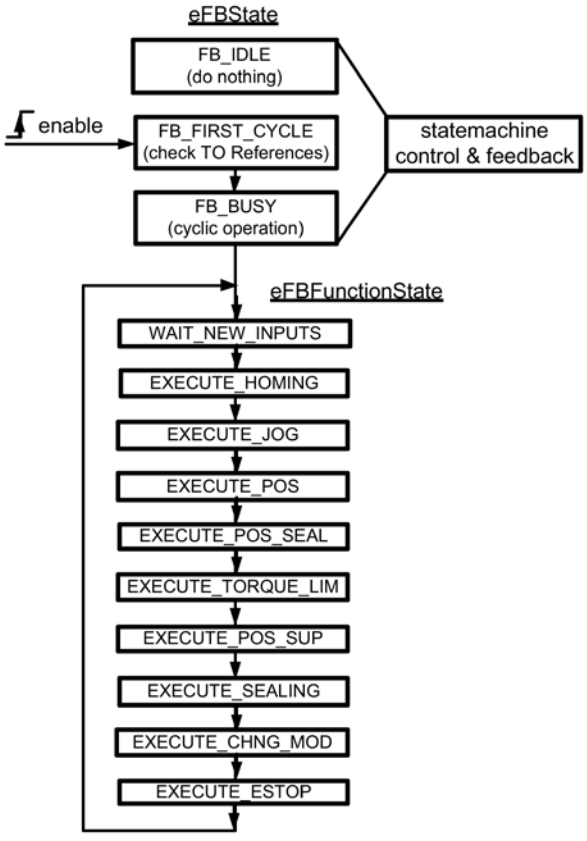


Figure 1-2 FB internal statemachines of the LPAC FB modules (using the FBLPacCrossSealPos block as an example)

In particular, all production machines in the packaging machine area whose axes are mutually coupled with coupling elements such as electronic gearboxes or electronic cams are possible fields of application. For this frequently occurring hierarchical axis structure, a differentiation is made between leading axes (also called master axes) and following axes (also called slave axes).

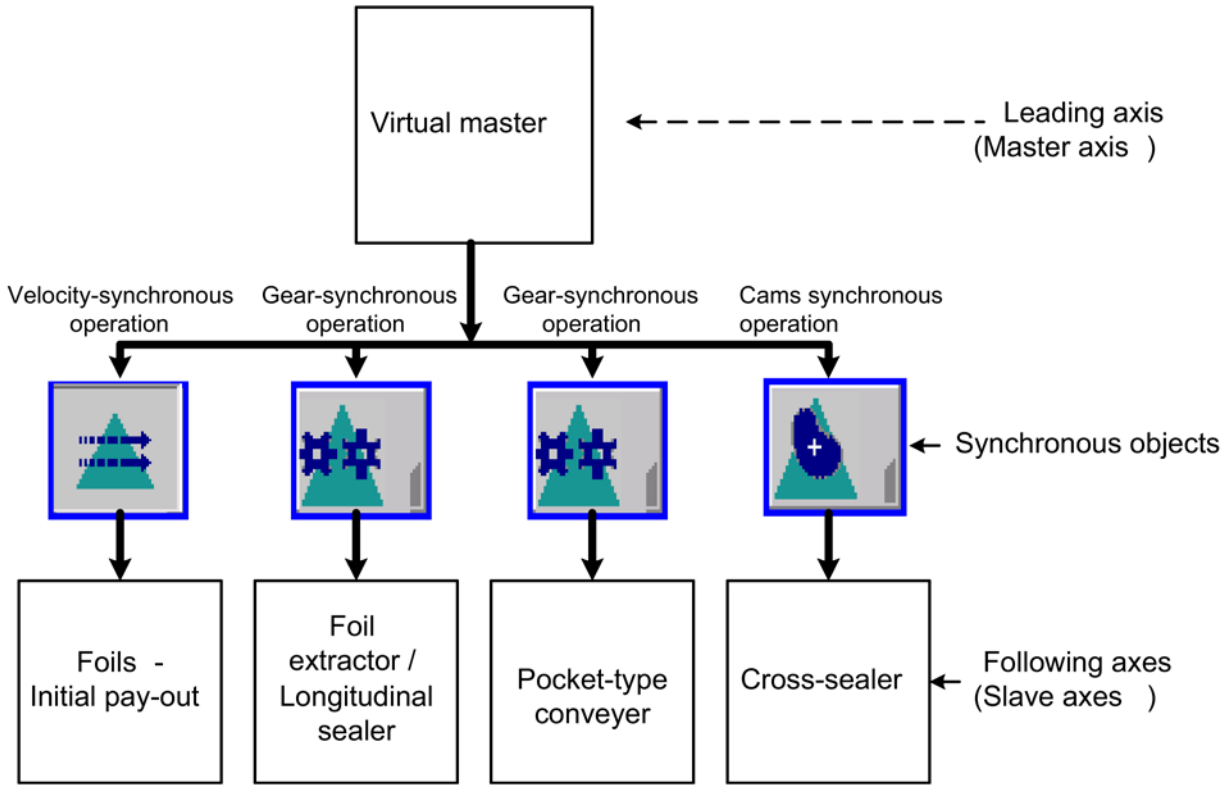


Figure 1-3 Typical axis structure and axis coupling of a packaging machine

1.1.2 Field of application

The target groups for the module library described in this document are

- Machine manufacturers who use SIMOTION and SINAMICS for the automation of packaging machines or general production machines
- Customers, solution partners and application engineers of the SIEMENS application centers

1.1.3 System requirements

The **LPac Basic Equipment Module Library** requires the installation of SIMOTION SCOUT as of version 4.4.

The *CAM* and *TControl* technology packages are also required.

1.1.4 Additional required libraries

The blocks of the **LPac Basic Equipment Module Library** are based, provided block-internal axes are controlled, on the **FBLMCBasicAxis** SIMOTION axis function block from the **LMCBasic** library.

LPac blocks for the cam calculation use internally function blocks of the **LCamHdl** library.

Blocks for simple reading of drive faults from SINAMICS drives use a function block of the **LDPV1** library internally.

Blocks for the implementation of a vertical continuous tubular bagging machine with a dosing unit coupled in gearing use the **LConLib** library internally.

The following libraries are still used in the **LPac** entire project:

- **LMsgHdl**, **LDPV1** for simplifying the message handling.

- **LPMLV30** for the operating mode management of a machine application in accordance with **OMAC V3**.

If you want to use a different operating mode management in your application, this is possible with little effort, because the **LPac** blocks are not linked internally with machine operating modes. The connection to the operating mode management is made only in the programs that call the **LPac** Basic Equipment Modules.

The **SIMOTION StartupCheck** unit is still deployed for the startup check.

Table 1- 1 Additional required libraries

Library	Know-how protec-tion	Required
LMCBasic	**Yes**	Mandatory https://support.industry.siemens.com/cs/de/de/view/48816191
LCamHdl	**Yes**	Mandatory https://support.industry.siemens.com/cs/de/de/view/105644659
LConLib	**No**	Mandatory https://support.industry.siemens.com/cs/ww/en/view/48805235
LPMLV30	**No**	Required only when operating mode management in accordance with OMAC V3 should be deployed https://support.industry.siemens.com/cs/de/de/view/49970441
LMsgHdl	**Yes**	Required only when the message handling should be deployed https://support.industry.siemens.com/cs/de/de/view/48955585
LDPV1	**Yes**	Required only when the message handling should be deployed https://support.industry.siemens.com/cs/ww/en/view/109479553
StartupCh eck	**Yes**	Required only when the startup check is to be should be deployed https://support.industry.siemens.com/cs/de/de/view/48947625

Figure 1-4 Library overview in the LPac project

Library structure

2

2.1 Structure of the LPac basic equipment module library

The **LPac Basic Equipment Module Library** contains solutions for the following technological tasks:

- Machine master
- Product supply and dosing
- Sealing (cross and longitudinal sealing unit) and temperature control
- Foil transport and path control (foil edge control)
- Auxiliary functions such as cam calculation

It consists of the following units:

Unit	Use
aVersion	Information on the version history of the library
cProtected	Internal constants
cPublic	Public constants, error messages
dProtected	Internal data structures
fMaster	Unit with blocks for machine master
fFoilFeed	Unit with blocks for foil transport and for preliminary foil feed
fSealer	Unit with blocks for cross-sealer
fDosing	Unit with blocks for dosing and scales coordination
fSupply	Unit with blocks for product supply (supply chain)
fFunc	Unit with functions / function blocks for call in the LPac function blocks or standalone use (sealing for torque, pressure mark correction, slip compensation)
fTControl	Unit with blocks for temperature control

All blocks and modules are freely available for the user and are not know-how protected.

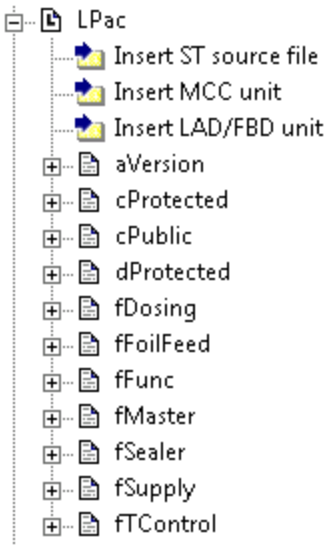


Figure 2-1 Structure of the LPac library

The following figure shows the structure of the **LPac Basic Equipment Module Library**.

Machine master (leading axis)	Product transport / dosing	Sealing units	Foil transport	Auxiliary functions
Master continuous	Pocket-type conveyer (gear or cam)	Cross-sealer (cam)	Foil feed (gear or cam)	Print-mark detection and correction
Master intermittent	Cup dosing unit (positioning)	Cross-sealer (positioning)	Foil feed (positioning)	Slip compensation
External master External encoder	Discharge control	Cam calculation	Preliminary foil feed (drive axis)	Slider-crank conversion
Delay axis		Chain bag control	Foil lateral control	Dancer position control
		PID temperature controller		Cam changer / disenga-ging
		SIMOTION on Temperature controller		Positioning time calculations
		Pulse width modulation		Cam controller / cam track
				Sealing jaws monitoring
				Monitoring Torque Following error
				No Product No Bag control
				Read drive faults

Figure 2-2 Structure of the LPac technology-oriented

Table 2- 1 Machine master

FBLPacMasterCont **Master continuous**	**FBLPacMasterInt** **Master intermittent**	**FBLPacExtEncoder** **External master / external encoder**	**FBLPacDelayMaster** **Delay axis**
Main function			
Production: _move in the continuous, counting, single product count modes	Production: Time-based control of the foil feed and cross-sealer in the continuous, counting, single product count modes	External encoder control	Compensation of the product drop time on vertical continuous tubular bagging machines with continuously running dosing unit
Product counter	Product counter		
InterruptProduction (interruption)	InterruptProduction (interruption)		
SyncProduction (synchronous operation)	Coordination of the dosing unit / discharge control		
Control of all following axes			
Control of cam calculation and activation			
Auxiliary functions			
Enable/disable axis (axes) of the controlled axes	Enable/disable axes of the controlled axes	Enable/disable encoder	Enable/disable axis
Home axis (axes)	Machine quick stop	Home encoder	Home axis
Jog axis / jog machine			Axis quick stop
Position axis			
Machine/axis quick stop			Gearing to the leading axis
Gearing to ext. leading axis incl. offset displacement			
Change modulo length			
Application			
Continuous machines	Vertical intermittent tubular bagging machines	Machine coupling to leading machines via external encoders	Vertical continuous tubular bagging machines with dosing coupled in gearing

Table 2- 2 Product transport / dosing

FBLPacTappetChainGear / cam **Pocket-type conveyor (gear or cam)**	**FBLPacDosingPos** **Cup dosing unit: (positioning)**	**FBLPacDosingControl** **Discharge control**
Main function		
Gearing or camming incl. offset displacement	Absolute/relative positioning Absolute positioning by and on modulo length	Control and coordination of the interface to a multi-head weigher Continuous, counting, single product count modes Product counter
Auxiliary functions		

FBLPacTappetChainGear / cam **Pocket-type conveyor (gear or cam)**	**FBLPacDosingPos** **Cup dosing unit: (positioning)**	**FBLPacDosingControl** **Discharge control**
Enable/disable axis	Enable/disable axis	-
Home axis	Home axis	
Jog axis	Jog axis	
Position axis	Axis quick stop	
Axis quick stop	Change modulo length	
Change modulo length		
Application		
Gearing or camming, e.g. pocket-type conveyors on horizontal tubular bagging machines	Positioning axes, e.g. cup or worm dosing units	Discharge control on vertical tubular bagging machines

Table 2- 3 Sealing units

FBLPac-CrossSeal-Cam **Cross-sealer (cam)**	**FBLPac-CrossSeal-Pos** **Cross-sealer (positioning)**	**FBLPac-CreaCam…** **Cam calculation**	**FBLPac-CrossSeal-ChainBag** **Chain bag control**	**FBLPacTO-TempCtrl** **Temperature controller with SIMOTION TO**	**FBLPac-PIDTemp-Ctrl** **Temperature controller with SIMOTION PID**	**FBLPac-PWMCtrl** **Pulse width modulation (PWM)**
Main function						
Camming incl. offset displacement	Position sealing jaws, complete sealing sequence, slider crank adaptation, sealing torque setting	Cam calculation for rotating and coupled cross-sealer • Cylindrical rotating • Coupled • Slider crank	The block controls a unit comprising two rotating cross-sealers for the production of chain bags. By appropriately adjusting the offset between the two units, a cutting position and a perforation position are approached and the production of chain bags controlled via the chain bag length.	Simple temperature controllers, e.g. cross or longitudinal sealing units with SIMOTION technology object, with self-tuning and PWM control	Simple temperature controllers, e.g. cross or longitudinal sealing units with SIMOTION PID block, without self-tuning	Pulse width-modulated control of semiconductor relays for temperature controls
Auxiliary functions						

Enable/disable axis	Enable/disable axis	-	Home axis grouping	-	-	-
Home axis	Home axis		Position axis grouping in basic position and synchronize to leading axis			
Jog axis	Jog axis		Approach cut position (offset)			
Position axis	Position axis		Approach perforation position (offset)			
Axis quick stop	Axis quick stop		Unit quick stop / abort cycle			
Change modulo length	Change modulo length		Bag counter			
Application						
Camming, e.g. for cont. cross-sealers, rotating/coupled blades	Position axis, e.g. on intermittent cross-sealer	-	Horizontal tubular bagging machine with rotating 2-axis cross-sealer for chain bag production	Temperature control	Simple temperature control	Simple temperature control

Table 2- 4 Foil transport

FBLPacFoilFeedGear / cam **Foil feed (gear or cam)**	**FBLPacFoilFeedPos** **Foil feed (positioning)**	**FBLPacPrelimFoilFeed** **Preliminary foil feed (drive axis)**	**FBLPacFoilGuide** **Foil lateral control**
Main function			
Gearing or camming incl. print mark correction and slip compensation	Absolute/relative positioning with/without print mark correction / slip compensation	_move; synchronous velocity operation incl. dancer control	Foil side-edge control
Auxiliary functions			

FBLPacFoilFeedGear / cam **Foil feed (gear or cam)**	**FBLPacFoilFeedPos** **Foil feed (positioning)**	**FBLPacPrelimFoilFeed** **Preliminary foil feed (drive axis)**	**FBLPacFoilGuide** **Foil lateral control**
Enable/disable axis	Enable/disable axis	Enable/disable axis	Jog right/left
Home axis	Home axis	Jog axis	
Jog axis	Jog axis	Axis quick stop	
Position axis	Axis quick stop	External axis (no TO)	
Change modulo length	Change modulo length		
Application			
Gearing or camming incl. correction functions, e.g. continuous foil feed	Position axis incl. correction functions, e.g. intermittent foil feed	Drive axis on intermittent or cont. packaging machines, e.g. for simple unwinders	Foil side-edge control photocells on packaging machines

Table 2- 5 Auxiliary functions

FBLPac-PrintmarkCorrection **Print-mark detection and correction**	**FBLPacSlipCompensation** **Slip compensation**	**FBLPacTranslationToRotation / RotationToTranslation** **Slider-crank conversion**	**FBLPac-DancerControl** **Dancer position control**	**FBLPacStopperCam** **Cam changer / disengaging**	**FBLPacGetPosTime ff** **Positioning time calculations**
Main function					
Print mark control by means of photocells, compensating motion with superimposed positioning, measurement and correction possible on different axes	Slip measurement and compensation with ext. encoder, compensating motion with superimposed positioning	Position transformation angle <-> distance	Simple dancer position control with • PID analog dancer (potentiometer) or • limit switches	Dynamic change of cyclic cams, e.g. for "on the fly" cut length change, disengaging function for No Product No Bag, No bag No seal product transport	Calculation of positioning times / paths and distance (acceleration time, constant traversing time, delay time for positioning operations with and without jerk limitation!
Auxiliary functions (None)					
Application					
Foil feed	Foil feed	-	-	Product change control, disengaging function	Calculation of cycle times

FBLPacOutput-CamControl / CamTrackCon-trolCamController / Cam track	**FBLPac-ProductInSealDe-tection** **Sealing jaws moni-toring**	**FBLPacTorque-Control** **Setting and moni-toring of torque / following error monitoring**	**FBLPacNPNBDe-tection** **No Product No Bag control**	**FBLPacReadSin-amicsMessages** **Read drive faults**	
Main function					
Parameter as-signment / Control Cam controller / Cam track (max. 32 output cams)	Torque monitoring and recording over a definable posi-tion change	Monitoring/setting of torque and for the following error monitoring	Shift register to detect and control the No Product No Bag function	Readout of drive fault messages from SINAMICS drives	
Auxiliary functions					
Output cam / Ena-ble/disable cam track			Position window for sensor	Acknowledgment of pending faults, alarms and SI messages	
Control cam out-put (forcing)			Run machine until empty		
Move output cam position(s) incre-mentally					
Compress and stretch output cam width(s) incremen-tally					
Application					
All possible pack-aging and produc-tion machines	All possible pack-aging and produc-tion machines	All possible pack-aging and produc-tion machines	For example, horizontal tubular bagging machines (Flowpacker)	All possible pack-aging and produc-tion machines	

The following figure shows the "LPac – function-oriented overview" structure.

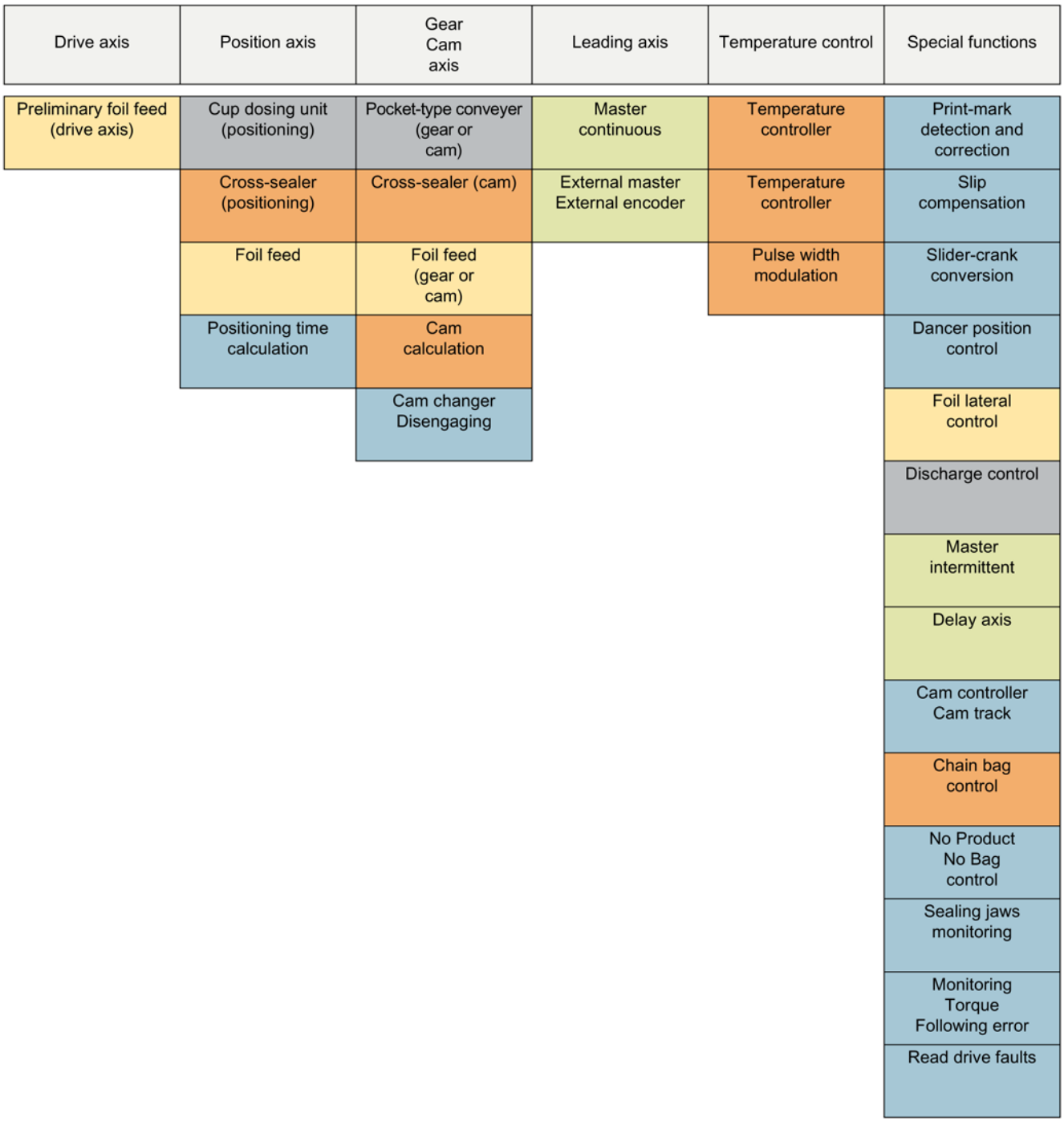


Figure 2-3 Structure of the LPac function-oriented

Table 2- 6 Drive axis

FBLPacPrelim-FoilFeed	Main function	Auxiliary functions	Application
Preliminary foil feed (drive axis)	_move; synchronous velocity operation incl. dancer control	Ena-ble/disable axis	Drive axis on intermittent or cont. packaging machines, e.g. for simple unwinders
		Jog axis	
		Axis quick stop	
		External axis (no TO)	

Table 2- 7 Position axis

FBLPacDosingPos Cup dosing unit: (positioning)	FBLPacCrossSealPos Cross-sealer: (positioning)	FBLPacFoilFeedPos Foil feed (positioning)	FBLPacGetPosTime / Get-PosTimeJerk/GetTimeAtPosition / GetPosAtTime Positioning time calculations
Main function			
Absolute/relative positioning	Position sealing jaws, com-plete sealing sequence	Absolute/relative positioning with/without print mark cor-rection / slip compensation	Calculation of positioning times / paths and distance (acceleration time, constant traversing time, delay time for positioning operations with and without jerk limita-tion!
Absolute positioning on modulo length	Slider crank adaptation, sealing torque setting		
Auxiliary functions			
Enable/disable axis	Enable/disable axis	Enable/disable axis	-
Home axis	Home axis	Home axis	
Jog axis	Jog axis	Jog axis	
Axis quick stop	Position axis	Axis quick stop	
Change modulo length	Axis quick stop	Change modulo length	
	Change modulo length		
Application			
Preferably for modulo/rotary axes, position axes, e.g. cup or worm dosing units	Preferably for linear axes, position axis, e.g. on inter-mittent cross-sealer	Preferably for linear axes, position axis incl. correction functions, e.g. intermittent foil feed	Calculation of cycle times, traversing times and posi-tions

Table 2- 8 Gear / cam axis

FBLPacTappetChainGear / cam **Pocket-type conveyor (gear or cam)**	**FBLPacCrossSealCam** **Cross-sealer: (Cam)**	**FBLPacFoilFeedGear / cam** **Foil feed (gear or cam)**	**FBLPacCreaCam…** **Cam calculation**	**FBLPacStopperCam** **Cam changer / disengaging**
Main function				
Gearing or camming incl. offset displacement	Camming incl. offset displacement	Gearing or camming incl. print mark correction and slip compensation	Cam calculation for rotating and coupled cross-sealer • Cylindrical rotating • Coupled • Slider crank	Dynamic change of cyclic cams, e.g. for "on the fly" cut length change, disengaging function for No Product No Bag, No bag No seal
Auxiliary functions				
Enable/disable axis	Enable/disable axis	Enable/disable axis	-	-
Home axis	Home axis	Home axis		
Jog axis	Jog axis	Jog axis		
Position axis	Position axis	Position axis		
Axis quick stop	Axis quick stop	Axis quick stop		
Change modulo length	Change modulo length	Change modulo length		
Application				
Preferably for modulo axes, gearing or camming, e.g. pocket-type conveyor on horizontal tubular bagging machines	Preferably for modulo axes, camming, e.g. for continuous cross-sealers, rotating/coupled blades on horizontal and vertical tubular bagging machines	Gearing and camming, e.g. foil feed axes on horizontal or vertical tubular bagging machines with integrated print mark correction and slip compensation	Preferably for modulo axes, cam calculation for permanently defined mechanical version on horizontal or vertical tubular bagging machines	For dynamic cam switchover/change, e.g. for "on the fly" product length change, disengaging or engaging function

Table 2- 9 Leading axis

FBLPacMasterCont **Master continuous**	**FBLPacExtEncoder** **External master / external encoder**
Main function	
Production: _move in the continuous, counting, single product count modes	External encoder control
Product counter	
InterruptProduction (interruption)	
SyncProduction (synchronous machine operation)	
Control of all following axes	
Control of cam calculation and activation	
Auxiliary functions	
Enable/disable axis	Enable/disable encoder
Home axis	Home encoder

FBLPacMasterCont **Master continuous**	**FBLPacExtEncoder** **External master / external encoder**
Jog axis	Change modulo length
Position axis	
Machine/axis quick stop	
Gearing to ext. leading axis incl. offset displacement	
Application	
Preferably for virtual modulo axis, continuous machines	Machine coupling to leading machines via external encoders

Table 2- 10 Temperature control

FBLPacTOTempCtrl **Temperature controller**	**FBLPacPIDTempCtrl** **Temperature controller**	**FBLPacPWMCtrl** **Pulse width modulation (PWM)**
Main function		
Simple temperature controllers, e.g. cross or longitudinal sealing units with SIMOTION technology object, with self-tuning and PWM control (license required)	Simple temperature controllers, e.g. cross or longitudinal sealing units with SIMOTION PID block, without self-tuning	Pulse width-modulated control of semiconductor relays for temperature controls

Table 2- 11 Special functions

	Main function	**Auxiliary functions**	**Application**
FBLPacPrintmarkCorrection Print-mark detection and correction	Print mark control by means of photocells, compensating motion with superimposed positioning, measurement and correction possible on different axes	-	Foil feed and correction
FBLPacSlipCompensation Slip compensation	Slip measurement and compensation with ext. encoder, compensating motion with superimposed positioning	-	Foil feed and correction
FBLPacTranslationToRotation / **RotationToTranslation** Slider-crank conversion	Position transformation angle <-> distance	-	-
FBLPacDancerControl Dancer position control	Simple dancer position control with • PID analog dancer (potentiometer) or • limit switches	-	-
FBLPacFoilGuide Foil lateral control	Foil side-edge control	Jog right/left	Foil side-edge control photocells on packaging machines

	Main function	Auxiliary functions	Application
FBLPacDosingControl Discharge control	Control and coordination of the interface to a multi-head weigher	-	Discharge control on vertical tubular bagging machines
	Continuous, counting, single product count modes		
	Product counter		
FBLPacMasterInt Master intermittent	Production: Time-based control of the foil feed and cross-sealer in the continuous, counting, single product count modes	Enable/disable axis of the controlled axes	Vertical intermittent tubular bagging machines
	Product counter		
	InterruptProduction (interruption)		
	Coordination of the dosing unit / discharge control	Machine quick stop	
FBLPacOutputCamControl / CamTrackControl Cam controller / cam track	Parameter assignment / Control Cam controller / Cam track (max. 32 output cams)	Output cam / Enable/disable cam track	All possible packaging and production machines
		Control cam output (forcing)	
		Move cam position(s) incrementally	
		Compress and stretch output cam width(s) incrementally	
FBLPacProductInSealDetection Sealing jaws monitoring	Torque monitoring and recording over a definable position change		All possible packaging and production machines
FBLPacTorqueControl Setting and monitoring of torque / following error monitoring	Monitoring/setting of torque and for the following error monitoring		All possible packaging and production machines
FBLPacNPNBDetection No Product No Bag control	Shift register to detect and control the No Product No Bag function	Position window for sensor	For example, horizontal tubular bagging machines (Flowpacker)
		Run machine until empty	
FBLPacReadSinamicsMessages Read drive faults	Readout of drive fault messages from SINAMICS drives	Acknowledgment of pending faults, alarms and SI messages	All possible packaging and production machines
FBLPacCrossSealChainBag Chain bag control	The block controls a unit comprising two rotating cross-sealers for the production of chain bags. By appropriately adjusting the offset between the two units, a cutting position and a perforation position are approached and the production of chain bags controlled via the chain bag length.	Home axis grouping	Horizontal tubular bagging machine with rotating 2-axis cross-sealer for chain bag production
		Position axis grouping in basic position and synchronize to leading axis	
		Approach cut position (offset)	
		Approach perforation position (offset)	
		Unit quick stop / abort cycle	
		Bag counter	

Integration

3

3.1 Integration of the library into a SIMOTION project.

The **LPac Basic Equipment Module Library** can be integrated in a SIMOTION SCOUT project with the SIMOTION SCOUT "Import library" function. For further information, see the *SIMOTION SCOUT Configuration Manual* and the SIMOTION SCOUT online help.

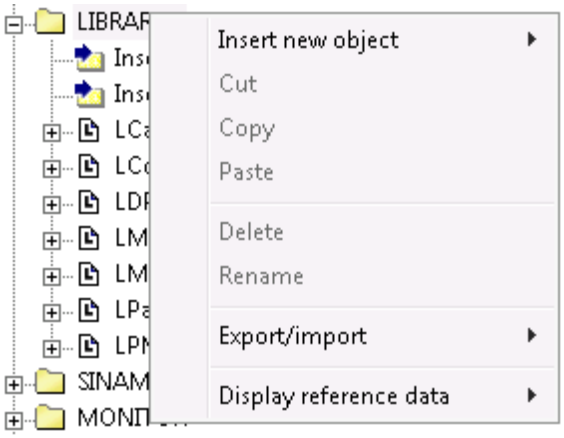


Figure 3-1 Integration of the library into a SIMOTION project

You can still perform a project configuration and create a basic project quickly and easily using the **SIMOTION easyProject** ProjectGenerator. A separate LPac module is available for the ProjectGenerator for this purpose.

The objects available as export files in xml format can be downloaded from the Internet as so-called "equipment modules".

Copy the downloaded xml file to the folder:

ProjectGenerator_V1_3_3_1 -> SIMOTION -> EquipmentModules -> V4_4 -> LPAC_V2

After starting the ProjectGenerator, the desired libraries, sample programs and technology objects are available for integration in any SIMOTION target platform.

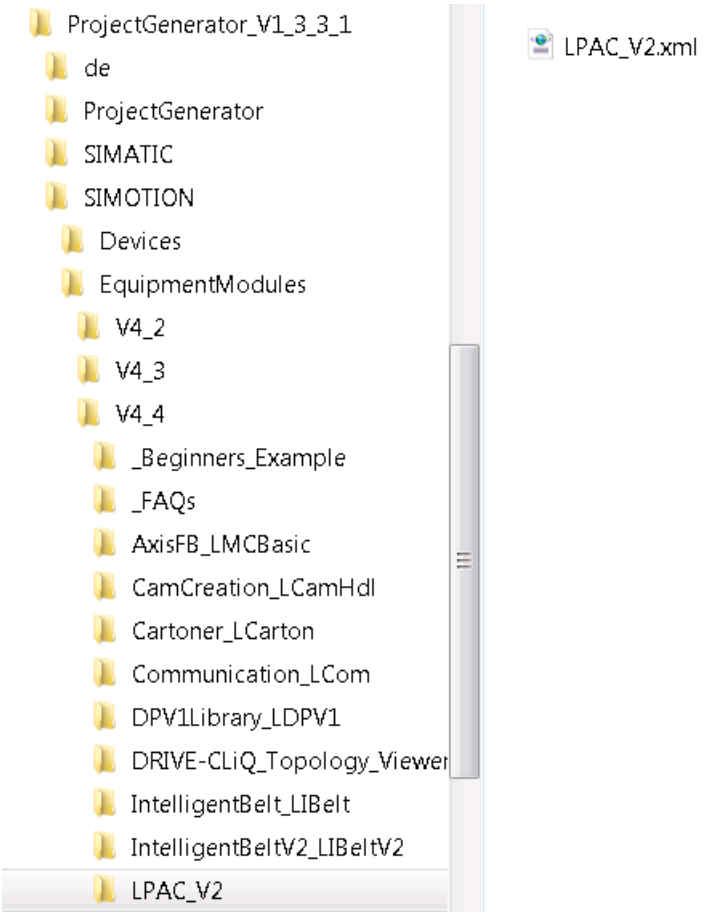


Figure 3-2 Integration of the LPac equipment modules in the ProjectGenerator

Function description

4

4.1 General explanations

A distinction is always made between level-triggered (*enable*) and edge-triggered (*execute*) control for the activation of the various module functions.

The control signals marked with *enable* activate the desired function level-triggered. This means that the function starts as soon as the level switches to high and remains active as long as the high level is present (input = TRUE).

The control signals marked with *execute* activate the desired function edge-triggered. This means that the function starts once with the rising edge of the input signal. These functions usually stop automatically (e.g. when the target position is reached after a positioning command).

The control signals are not reset automatically and must be reset from the calling control program, e.g. after successful execution of the desired function.

Each module has a PLCopen-compliant I/O signals part, e.g.:

- enable: Activates the block
- valid: Block is activated
- busy: Selected/activated functionality being processed
- done: Selected/activated functionality completed successfully
- error: Error occurred with the selected function
- errorID: Error number for diagnostic purposes

The following figure is a signal flow diagram to illustrate the principle behavior of the control and feedback.

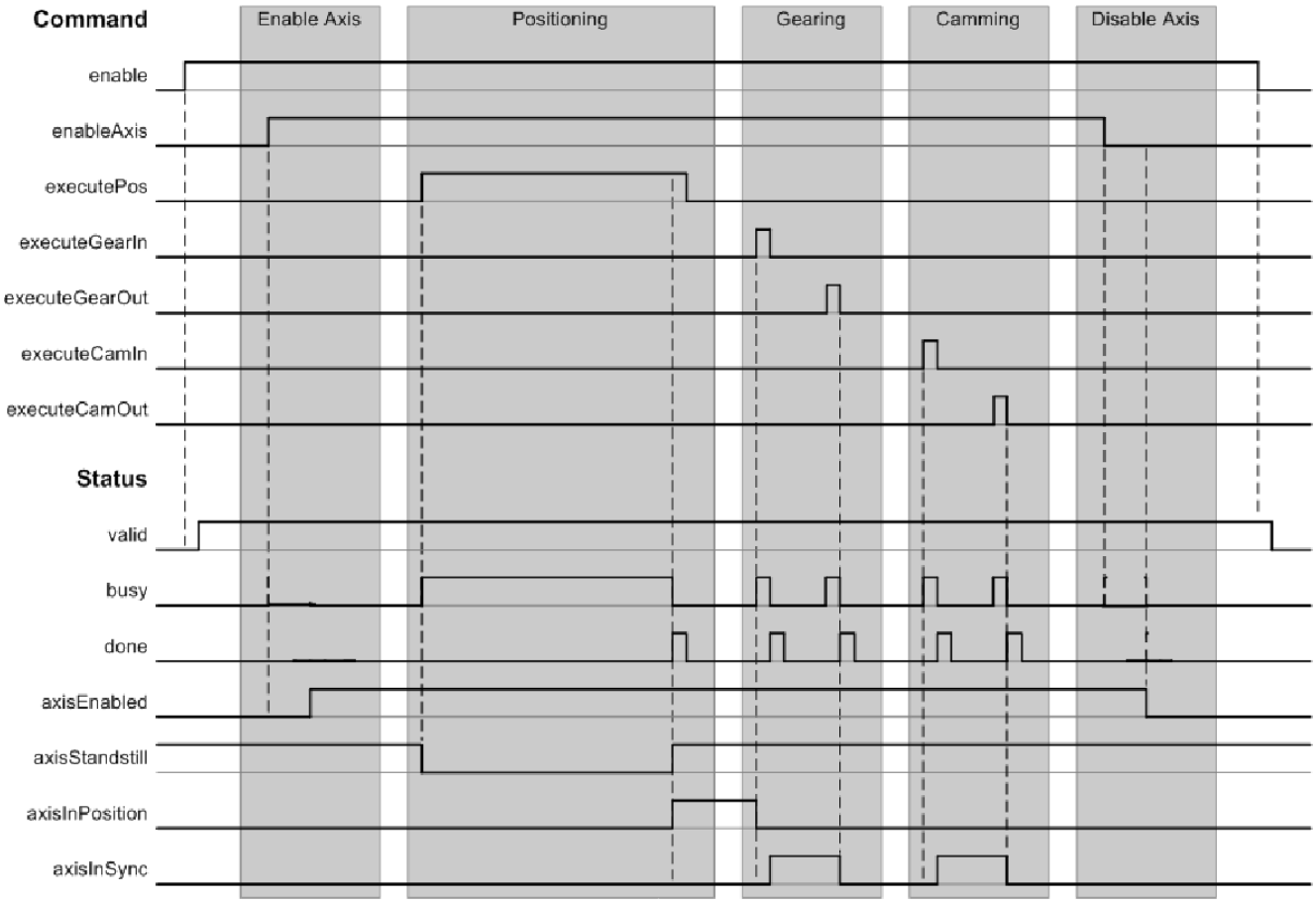


Figure 4-1 Signal flow diagram

4.2 General axis functions

Depending on the requirement, the **LPac** blocks provide various general axis functions such as enable/disable axis, jogging, homing, positioning and quick stop. Detailed descriptions of the used general axis functions can be obtained from the documentation for the SIMOTION **axis function block**. These functions are discussed only briefly below. The parameters required for the associated function can either be entered in the SIMOTION default screen forms (this selection is activated for a value of -1 for the associated parameter) or entered in the appropriate data structures in the project.

The required data structures are provided completely edited when the LPac project is used, or when integrated with the ProjectGenerator.

4.2.1 Enable/disable axis

The Enable/disable axis functionality is provided by the SIMOTION **FBLMCBasicAxis** axis function block. The enable axis action is performed with the positive level at *enableAxis*. Note the associated detailed description in the SIMOTION *Axis Function Block* Function Manual.

Prerequisite for error-free execution is the general block enable that can be polled via the *valid* status signal. In addition, no *error* may be present.

The status is returned via the *axisEnabled* output.

4.2.2 Home axis

The Homing axis functionality is provided by the SIMOTION **FBLMCBasicAxis** axis function block together with the settings for the technology object axis. Homing is performed in the set mode with the positive edge at *executeHoming*. For incremental encoders, the axis is homed actively with a homing procedure. For absolute encoders, an absolute encoder adjustment is performed to a specified zero position.

After completed homing, the status is issued at the *axisHomed* output. The encoder type (incremental or absolute) is detected automatically at the associated axis. Note the associated detailed description in the SIMOTION *Axis Function Block* Function Manual. Prerequisite for error-free execution is the general block enable that can be polled via the *valid* status signal. and the axis enable whose status is polled via the *axisEnabled* signal. In addition, no *error* may be present.

The required parameters for homing (home position, etc.) can be entered in the **sLMCBasicHomingDataType** substructures of the respective modules.

Table 4- 1 sLMCBasicHomingDataType structure

Element	Data type	Initial value	Meaning
r32HomePosition	REAL	0.0	Home position coordinate
eIncHomingMode	eLMCBasicIncHoming-ModeType	ACTIVE	Homing mode ACTIVE: Homing operation in the configured mode PASSIVE: Home with the next zero mark DIRECT: Set the actual position value The ACTIVE, PASSIVE and DIRECT settings only apply for incremental encoders. REDEFINE_POSITION: No homing operation is started. The position of the axis or the external encoder is set via the *redefinePosition* function. This does not affect the *axisHomed* status. This setting can be used for all encoder types.

Element	Data type	Initial value	Meaning
eRedefineSpecification	EnumRedefineSpecification	COMMAND_VALUE	Specifies the assignment destination for the new position. COMMAND_VALUE: The value in r32HomePosition is set as the new position setpoint depending on *redefineMode*, and the actual position is adjusted. ACTUAL_VALUE: The value in r32HomePosition is set as the new actual position depending on *redefineMode*, and the position setpoint is adjusted accordingly. COMMAND_VALUE_BASIC_MOTION: The command refers only to the Basic Coordinate System. COMMAND_VALUE_SUPERIMPOSED_MOTION: The command only refers to the superimposed coordinate system. VIRTUAL_AXIS: Is not permitted in the axis FB and is rejected with error.
eRedefineMode	EnumRedefineMode	ABSOLUTE	Specifies the type of assignment for the new position. ABSOLUTE: The values are assigned as absolute positions. RELATIVE: The values are added to the existing position (offset). With RELATIVE, the response is the same for *redefineSpecification* := COMMAND_VALUE and *redefineSpecification* := ACTUAL_VALUE.

4.2.3 Jog axis

The Jogging axis functionality is provided by the SIMOTION **FBLMCBasicAxis** axis function block. The axis is jogged in the respective direction with a positive level at *jogPositive/Forward/Open* or *jogNegative/Backward/Close*. Jogging is stopped immediately with a negative level. Note the associated detailed description in the SIMOTION *Axis Function Block* Function Manual. Prerequisite for error-free execution is the general block enable that can be polled via the *valid* status signal. and the axis enable whose status is polled via the *axisEnabled* signal. In addition, no *error* may be present.

The selection of the jogging is indicated via the *jogPositve/Foward/CloseSelected* or *jogNegative/Backward/OpenSelected* output.

The required parameters for jogging (velocity, acceleration, deceleration and possibly jerk) can be entered in the **sLPacJogDataType** substructures of the respective modules. The specified value -1 means that the appropriate value is taken from the associated SIMOTION default screen form.

Note

Continuous jogging is always performed with the internal default value.

Table 4- 2 sLPacJogDataType structure

Element	Data type	Initial value	Meaning
r32Velocity	REAL	-1.0	Velocity setpoint for jogging 1)
r32Acceleration	REAL	-1.0	Acceleration setpoint for jogging 1)
r32Deceleration	REAL	-1.0	Deceleration setpoint for jogging 1)
r32Jerk	REAL	-1.0	Jerk for jogging Jerk = 0: Trapezoidal motion profile Jerk = -1.0: Use of the default value on the TO Jerk > 0: Smooth motion profile with specified jerk value 1)

1) With value -1.0, the respective parameter value is taken from the default on the TO

4.2.4 Axis positioning

The Positioning axis functionality is provided by the SIMOTION **FBLMCBasicAxis** axis function block. Depending on the parameterization, a relative or absolute positioning action is performed with the positive edge at *executePos*. Note the associated detailed description in the SIMOTION Axis Function Block Function Manual. Prerequisite for error-free execution is the general block enable that can be polled via the *valid* status signal and the axis enable whose status is polled via the *axisEnabled* signal. In addition, no *error* may be present.

The selection of the positioning operation is indicated via the *posSelected* output.

The required parameters for positioning (velocity, acceleration, deceleration and possibly jerk) can be entered in the **sLPacPosDataType** substructures of the respective modules. The specified value -1.0 means that the appropriate value is taken from the associated SIMOTION default screen form. The *posAbs* control signal sets whether absolute (signal = TRUE) or relative positioning is to be performed. The traversing direction can also be specified via the *eDirection* variable.

As soon as the axis position is in the target window, this is indicated via the *axisInPosition* output.

Table 4- 3 sLPacPosDataType structure

Element	Data type	Initial value	Meaning
r32Position	REAL	0.0	Target position or distance for positioning
r32Velocity	REAL	-1.0	Velocity setpoint for positioning 1)
r32Acceleration	REAL	-1.0	Acceleration setpoint for positioning 1)
r32Deceleration	REAL	-1.0	Deceleration setpoint for positioning 1)
r32Jerk	REAL	-1.0	Jerk for positioning Jerk = 0: Trapezoidal motion profile Jerk = -1.0: Use of the default Jerk > 0: Smooth motion profile with specified jerk value 1)
eDirection	Enum Direction	POSITIVE	Direction specification: EFFECTIVE: Last programmed set direction NEGATIVE: Negative POSITIVE: Positive SHORTEST_WAY: Shortest path USER_DEFAULT: User default BY_VALUE: Is not permitted and is rejected with an error

1) With value -1.0, the respective parameter value is taken from the default on the TO

4.2.5 Synchronize/desynchronize axis in gearing

The Synchronize/desynchronize gearing axis functionality is provided by the SIMOTION **FBLMCBasicAxis** axis function block. The synchronization to the master is started by a positive edge of the Boolean input *executeGearIn*.

Note the associated detailed description in the SIMOTION *Axis Function Block Function Manual*. Prerequisite for error-free execution is the general block enable that can be queried via the *valid* status signal and the axis enable whose status can be queried via the *axisEnabled* signal. In addition, no *error* may be present.

The gear ratio is only transferred by specifying the numerator and the denominator.

The following parameters can be set for the gearing in the **sLPacGearingDataType** substructures:

- Absolute and relative synchronous operation
- Synchronization mode
- Phase shift

Master and slave axis must be homed for the absolute synchronous operation.

Time related or USER_DEFAULT can be selected as synchronization criterion. All synchronization conditions can be executed in SIMOTION with USER_DEFAULT by assigning the appropriate axis default values.

The *axisInSync* output is set as soon as and as long as the slave axis traverses synchronously to the master axis. The *busy* and *done* signals are set as for all other execute jobs.

The axis is desynchronized by a rising edge of the Boolean input *executeGearOut*.

Desynchonization is performed with the PLCopen FB _MC_GearOut(). The default data is used for the desynchronization conditions. The output bit *axisInSync* is reset during the selection.

Table 4- 4 sLPacGearingDataType structure

Element	Data type	Initial value	Meaning
r32Offset	REAL	0.0	Offset displacement between the leading axis and following axis
r32Velocity	REAL	-1.0	Velocity setpoint for synchronization and desynchronization [1)]
r32Acceleration	REAL	-1.0	Acceleration setpoint for synchronization and desynchronization [1)]
r32Deceleration	REAL	-1.0	Deceleration setpoint for synchronization and desynchronization [1)]
r32Jerk	REAL	-1.0	Jerk for synchronization and desynchronization Jerk = 0: Trapezoidal motion profile Jerk = -1.0: Use of the default value on the TO Jerk > 0: Smooth motion profile with specified jerk value [1)]
r32PhaseShift	REAL	0.0	Phase shift for absolute gearing
i32Numerator	DINT	1	Numerator for gear ratio
i32Denominator	DINT	1	Denominator for gear ratio

Element	Data type	Initial value	Meaning
eGearInMode	_MC_ GearInMode	IMMEDIATELY_ BY_TIME_ PROFILE	Specification of the synchronization mode / engage mode USER_DEFAULT: With USER_DEFAULT, the settings for the synchronization are taken from the userDefault system variables (*gearingSettings*, *syncDynamics*, …) of the following object. Except for the values set at the following inputs: • *absolute* (*userDefault.gearingSettings.typeOfGearing* system variable) • *RatioNumerator* (*userDefault.gearingSettings.defineMode*, *userDefault.gearingSettings.numerator* system variables) • *RatioDenominator* (*userDefault.gearingSettings.defineMode*, *userDefault.gearingSettings.denominator* system variables) With IMMEDIATELY_BY_TIME_PROFILE, synchronization occurs immediately by time, taking into account the dynamic response values set on the function block. Synchronous operation is operated with the following parameters: • *userDefault.syncprofile.syncprofilereference* equivalent to RELATE_SYNC_PROFILE_TO_TIME • *userDefault.cammingsettings.synchronizingmode* equivalent to IMMEDIATELY • *userDefault.cammingsettings.synchronizingdirection* equivalent to SYSTEM_DEFINED
boAbsolute	BOOL	TRUE	TRUE: Absolute gearing FALSE: Relative gearing

1) With value -1.0, the respective parameter value is taken from the default on the TO

4.2.6 Synchronizing/desynchronizing axis in camming

The Synchronize/desynchronize camming axis functionality is provided by the SIMOTION **FBLMCBasicAxis** axis function block. The synchronization to the master is started by a positive edge of the Boolean input *executeCamIn*.

Note the associated detailed description in the SIMOTION *Axis Function Block Function Manual*. Prerequisite for error-free execution is the general block enable that can be queried via the *valid* status signal and the axis enable whose status can be queried via the *axisEnabled* signal. In addition, no *error* may be present.

The following parameters can be set for the camming in the **sLPacCammingDataType** substructures:

- Offset for master and slave
- Scaling of the master and slave values
- Relative or absolute reference for master and slave axis
- Cyclic or one-time cam processing mode

After synchronization, the *axisInSync* output is set to TRUE. The *done* and *busy* signals are set as for the other execute jobs.

It is not possible to change the cam during operation. The new cam is used when camming is reselected via a positive edge.

Time related or USER_DEFAULT can be selected as synchronization criterion. All synchronization conditions can be executed in SIMOTION with USER_DEFAULT by assigning the appropriate axis default values.

The axis is desynchronized by a rising edge of the Boolean input *executeCamOut*.

Desynchonization is performed with the PLCopen FB _MC_CamOut(). The default data is used for the desynchronization conditions. The output bit *axisInSync* is reset during the selection.

Table 4- 5 sLPacCammingDataType structure

Element	Data type	Initial value	Meaning
r32Offset	REAL	0.0	Offset displacement between the leading axis and following axis
r32MasterOffset	REAL	0.0	Specification of the offset of the master values in the master coordinates.
r32SlaveOffset	REAL	0.0	Specification of the offset of the slave values in the slave axis coordinates.
r32MasterScaling	REAL	1.0	Specification of the scaling for the master values in the master coordinates.
r32SlaveScaling	REAL	1.0	Specification of the scaling for the slave values in the slave coordinates.
r32Velocity	REAL	-1.0	Specification of the maximum synchronization velocity [1)]
r32Acceleration	REAL	-1.0	Specification of the maximum synchronization acceleration [1)]
r32Deceleration	REAL	-1.0	Specification of the maximum synchronization deceleration [1)]
r32Jerk	REAL	-1.0	Specification of the maximum synchronization jerk [1)]

Element	Data type	Initial value	Meaning
eCamInMode	_MC_CamIn Mode	IMMEDIATELY _BY _TIME_ PROFILE	Specification of the synchronization mode / engage mode USER_DEFAULT: With USER_DEFAULT, the settings for the synchronization are taken from the userDefault system variables (*cammingSettings*, *syncDynamics*, …) of the following object. Except for the values set at the following inputs: • *MasterAbsolute* (*userDefault.cammingSettings.masterMode* system variable) • *SlaveAbsolute* (*userDefault.cammingSettings.slaveMode* system variable) • *CyclicMode* (userDefault.cammingSettings.cammingMode system variable) With IMMEDIATELY_BY_TIME_PROFILE, synchronization occurs immediately by time, taking into account the dynamic response values set on the function block. Synchronous operation is operated with the following parameters: • *userDefault.syncprofile.syncprofilereference* equivalent to RELATE_SYNC_PROFILE_TO_TIME • *userDefault.cammingsettings.synchronizingmode* equivalent to IMMEDIATELY • *userDefault.cammingsettings.synchronizingdirection* equivalent to SYSTEM_DEFINED [2]
boMasterAbsolute	BOOL	TRUE	TRUE: Absolute reference to master
			FALSE: Relative reference to master
boSlaveAbsolute	BOOL	TRUE	TRUE: Absolute reference to slave
			FALSE: Relative reference to slave
boCyclicMode	BOOL	TRUE	TRUE: Cyclic cam
			FALSE: One pass

1) With value -1, the respective parameter value is taken from the default on the TO

2) SYSTEM_DEFINED corresponds to the shortest path setting, although the direction of motion is maintained while the axis is in motion.

4.2.7 Activation of an offset in the active synchronous operation

An additional offset can be activated in the active gearing or camming via a rising edge of the Boolean input *executePhasing*. The area to be offset must be set (master area or slave area). How the offset is to be traversed can be parameterized in the dynamic response values. You can also specify whether the offset is to be calculated absolutely or relatively to the existing offset.

The following parameters can be set for the offset adjustment in the **sLMCBasicPhasingDataType** substructures:

Table 4- 6 sLMCBasicPhasingDataType structure

Element	Data type	Initial value	Meaning
r32Velocity	REAL	-1.0	Velocity setpoint for the offset adjustment motion 1)
r32Acceleration	REAL	-1.0	Acceleration setpoint for the offset adjustment motion 1)
r32Deceleration	REAL	-1.0	Deceleration setpoint for the offset adjustment motion 1)
r32Jerk	REAL	-1.0	Jerk for the offset adjustment motion Jerk = 0: Trapezoidal motion profile Jerk = -1.0: Use of the default value on the TO Jerk > 0: Smooth motion profile with specified jerk value 1)
eOffsetRange	Enum-CamRange	MASTER_ RANGE	Specification of the reference for the offset (master or slave)
r32Offset	REAL	0.0	Offset on the active synchronous operation
eDynamic Reference	EnumFol-lowingObject DynamicRef-erence	TOTAL_ MOVE	Specifies whether the dynamic response values of the command refer to the resulting total motion or to the differential motion. TOTAL_MOVE: Dynamic response parameters refer to the total motion OFFSET_MOVE: Dynamic response parameters refer to the differential motion
boAbsolute	BOOL	TRUE	Specifies the type of phase shift TRUE: The phase shift is taken over as absolute values FALSE: The phase shift is added to an existing offset

1) With value -1.0, the respective parameter value is taken from the default on the TO

4.2.8 Axis rapid stop

The Quick stop axis functionality is provided by the SIMOTION **FBLMCBasicAxis** axis function block. A stopping action is performed with the positive level at *enableEStop*. Note the associated detailed description in the SIMOTION *Axis Function Block* Function Manual.

Prerequisite for error-free execution is the general block enable that can be polled via the *valid* status signal and the axis enable that is polled via the *axisEnabled* signal. In addition, no *error* may be present.

The required parameters for quick stop (stop mode, deceleration and possibly stop ramp time and jerk) can be entered in the sLMCBasicEStopDataType substructures of the respective modules. The specified value -1.0 means that the appropriate value is taken from the associated SIMOTION default screen form.

Table 4- 7 sLMCBasicEStopDataType structure

Element	Data type	Initial value	Meaning
r32Deceleration	REAL	-1.0	Deceleration setpoint for quick stop [1)]
r32Jerk	REAL	-1.0	Jerk for quick stop Jerk = 0: Trapezoidal motion profile Jerk = -1.0: Use of the default value on the TO Jerk > 0: Smooth motion profile with specified jerk value [1)]
eEStopDriveMode	EnumStopDriveMode	STOP_WITH_ DYNAMIC_ PARAMETER	Setting for quick stop (emergency stop) STOP_IN_DEFINED_TIME (136): Stop axis using a time-dependent ramp STOP_WITH_COMMAND_VALUE_ZERO (139): Stop axis using setpoint 0 STOP_WITH_MAXIMAL_DECELERATION (140): Stop axis using maximum deceleration STOP_WITH_DYNAMIC_PARAMETER (326): Stop axis using programmed dynamic values

Element	Data type	Initial value	Meaning
r32StopTime	REAL	0.0	Time for quick stop procedure in seconds [s], if eEStopDriveMode = STOP_IN_DEFINED_TIME
eEstopMode	eLMCBasicEstop-ModeType	STOP_EMERGENCY	Setting whether normal stop or quick stop STOP_WITHOUT_ABORT: Stop without abort STOP_AND_ABORT: Stop with abort STOP_EMERGENCY: Use of the *_stopEmergency* system command

1) With value -1.0, the respective parameter value is taken from the default on the TO

Note

Safety functions, such as emergency off or emergency stop, must always be implemented safely and redundantly by hardware in accordance with the applicable guidelines and standards. The Safety Integrated functions in the SINAMICS drives in conjunction with a SIEMENS safety control can be used for this purpose.

Software-based stoppages of the individual drives may be implemented only in addition to the hardware execution using the input signals available at the blocks. The following procedure has proved to be effective: In coupled axis operation, i.e. in production mode, the axes are initially stopped synchronously by a leading signal to the machine master. The stop action should be time monitored. After expiration of the monitoring time, the individual blocks can also receive the quick stop signal. After expiration of the leading time of the hardware contact, the hardware safety shutdown must ensure the safe standstill or shutdown of the drives.

The Safety Integrated functions of the SINAMICS drives are not taken into account in the LPac modules and in the program units!

4.3 Machine master (leading axis)

A structure of leading axis (axes) and coupled following axes (so-called master – slave structure) is frequently deployed to implement an axis and drive structure for production machines with coupled drives. In this case, the machine master is often implemented by a virtual axis that, for example, controls the following drive continuously or in the product cycle. This slave drive can then be coupled to the leading axis in gearing or camming. The **LPac FBLPacMasterCont** equipment module can be used as machine master.

In particular, the **FBLPacMasterInt** equipment module can be used for intermittently operating vertical tubular bagging machines. This module does not represent a virtual axis,

but rather uses the time specifications to control the foil feed and cross-sealing following axes, and coordinates the interface to the dosing unit (discharge control).

If production machines are to be coupled to an upstream machine or production line, this can be implemented in the SIMOTION system via the distributed synchronous operation (master value coupling via PROFIBUS or PROFINET) or also via an external encoder. With an external encoder, a master value encoder is coupled physically, e.g. to the main shaft of the upstream machine. The continuous master block **FBLPacMasterCont** contained in **LPAC** can be synchronized with this external encoder master value (position and velocity) via gearing. The **FBLPacExtEncoder** block can be deployed to control the external encoder.

4.3.1 FBLPacMasterCont - continuous master

4.3.1.1 Technological consideration

The **FBLPacMasterCont** block simulates the "continuous master" machine module, for example on vertical or horizontal continuous tubular bagging machines or other machine modules with similar function requirements. The block can be deployed together with other blocks of the **LPac** library for implementing a SIMOTION complete application of a continuous tubular bagging machine and also independently for similarly designed machines that require a continuously running master axis. A virtual axis normally serves as master axis.

To also couple complete machines to each other in synchronous operation, the **FBLPacMasterCont** block offers the possibility to operate the controlled axis also in gearing for superimposed or upstream leading axes or machines or external encoders. Real axes can also be controlled. In this case however, the homing function must be parameterized accordingly in the block.

The "Continuous position-controlled traversing" function is available as main function for use as continuous master with virtual leading axis. An integrated product counter is available for use in production machines. The leading axis, and thus the complete machine, can also be operated in the "continuous operation", "single product" and "product counting" modes.

Within a complete application, coupled slave axes are controlled via gearing or camming. The standstill of the complete machine via the master axis is also performed in coupled operation (both normal stop and quick stop).

4.3.1.2 Module description

Scope of functions

The **FBLPacMasterCont** function block offers the main functions and auxiliary functions as well as possible options shown in the following overview.

Function	Description	Implementation
Main function	Traverse axis continuously position-controlled	Axis is started and moved position-controlled via the level-triggered *enableProduction* input (move). Resetting the *enableProduction* signal causes stoppage at the next available cycle end (product end) (pos).
Option	"Continuous operation, single product or product counting" modes	The leading axis can be operated in the following modes by specifying the operation modes via the *machineMode* input:
		Continuous: Endless continuous traversing until stopped.
		Single: Start-stop operation for a product cycle
		Counting: Pass for the defined number of product cycles specified via the product count setpoint
	Product counter	The integrated product counter counts the number of product cycles and can be reset per command
	Synchronous operation with offset adjustment	The master axis, e.g. on superimposed line masters or external encoders per gearing, can be coupled or decoupled by control of the *"executeGearIn"* or *"executeGearOut"* inputs. An offset / phase shift between the line leading axis and the machine master axis can be set with the "*executeChangeOffset*" input.
Auxiliary functions	Enable axis	The axis is enabled via the level-triggered *enableAxis* input. When the production operation is activated, the master axis is automatically enabled or disabled again by the internal sequence control.
	Home axis	The homing procedure of the axis is started via the *executeHoming* input. When the production operation is activated, the master axis is automatically homed by the internal sequence control.
	Jog axis	Positive jogging can be initiated via the level-triggered *JogForward* input. When the production operation is activated, the complete axis grouping, i.e. the complete machine, can be jogged forwards during a production interruption (interruptProduction) with this function.

Function	Description	Implementation
	Position axis	An absolute or relative positioning action can be started via the edge-triggered *executePos* input.
	Axis quick stop	A quick stop can be performed or a motion of the axis prevented via the level-triggered *enableEStop* input. In the LPac sample project, a quick stop of the master axis is performed with a quick stop with specification of the stop ramp time (eEStopMode = STOP_WITHOUT_ABORT and eEStopDriveMode = STOP_IN_DEFINED_TIME). For this purpose, the r32StopTime parameterized by the user in the block (default value 0.5 s) is converted to the required r32Deceleration deceleration based on the current axis velocity and entered in the sLMCBasicStopData structure.
	Change of the axis configuration data	The axis configuration data (modulo length, modulo yes/no) can be changed via the subordinate **FBLPacChangeTOConfigDataMod** function block. The modulo length specified in the ramp-up of the controller is automatically entered and activated by the block in the appropriate configuration data element of the axis. The modulo properties can also be changed at any time in the cyclic program by specifying different values. The user must only ensure that the axis is not enabled. A TO restart must be performed to activate the changed configuration data.

Schematic representation in LAD/FBD

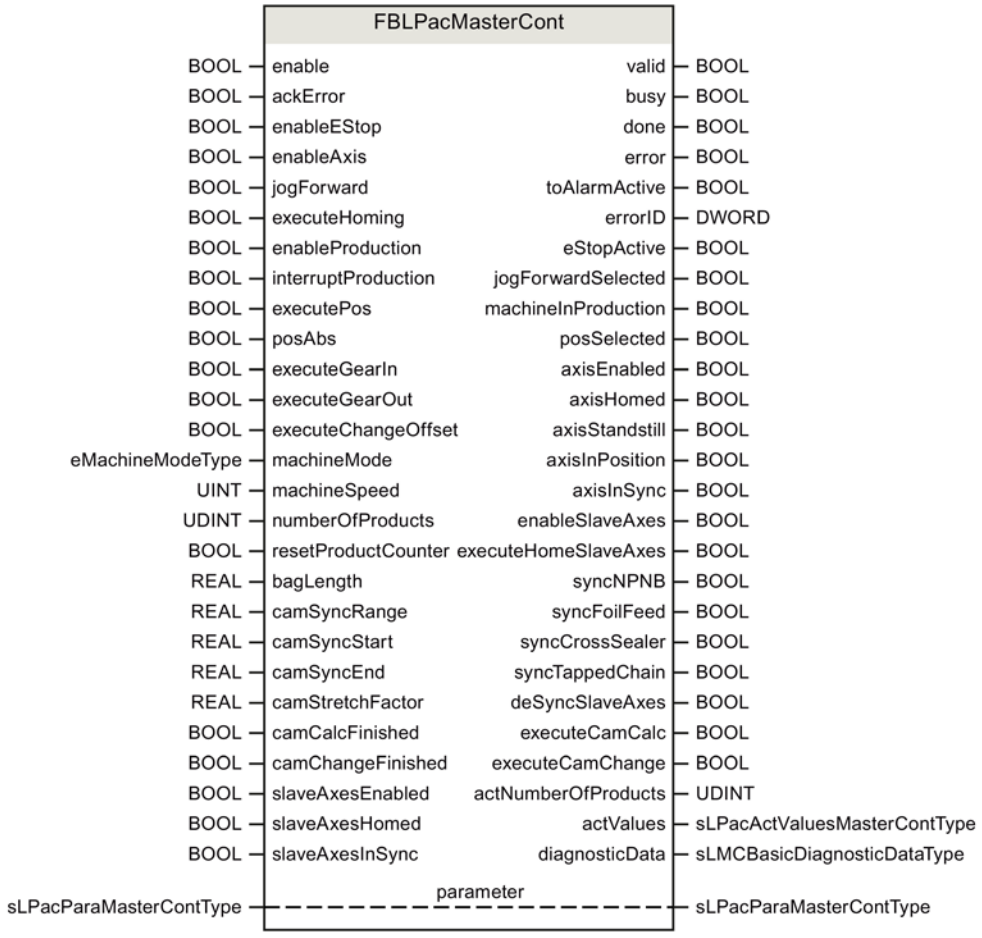


Figure 4-2 FBLPacMasterCont - schematic block view in LAD/FBD

Input/output parameters

Table 4- 8 Input/output signals

Element	P type [1]	Data type	Initial value	Meaning
enable	IN	BOOL	FALSE	Activation of the FB
ackError	IN	BOOL	FALSE	Acknowledge error
enableEStop	IN	BOOL	FALSE	Activate quick stop
enableAxis	IN	BOOL	FALSE	Enable axis

Element	P type [1]	Data type	Initial value	Meaning
jogForward	IN	BOOL	FALSE	Continuous jogging in the positive direction (In the "Machine not in production" state, the master axis is jogged separately, in the "Machine in production" state, the complete machine can only be jogged in the "Interrupt production" mode.)
executeHoming	IN	BOOL	FALSE	Start homing. Direct homing is performed, i.e. the position setpoint is set to the specified home position. When a real axis is deployed, the homing function must be parameterized differently in the block in accordance with the encoder used!
enableProduction	IN	BOOL	FALSE	Start production (position-controlled turning) or stop production at the next achievable cycle end
interruptProduction	IN	BOOL	FALSE	Interrupt production, i.e. stop at the specified stop position without desynchronizing the following axes
executePos	IN	BOOL	FALSE	Start positioning; (possible only when the machine is not in production, i.e. *machineInProduction* = FALSE)
posAbs	IN	BOOL	FALSE	TRUE: The positioning initiated with executePos is absolute FALSE: The positioning initiated with executePos is relative
executeGearIn	IN	BOOL	FALSE	Start synchronization / activate gearing
executeGearOut	IN	BOOL	FALSE	Start desynchronization / deactivate gearing
executeChangeOffset	IN	BOOL	FALSE	Activate start offset displacement; the offset adjustment acts relative, i.e. at each activation, the specified offset value is traversed relatively. Only effective when the master axis is synchronized to an external master value.

Element	P type [1]	Data type	Initial value	Meaning
machineMode	IN	eMachineMode-Type	continued	Production mode The leading axis can be operated in the following modes by specifying the modes via the *machineMode* input: Continuous: Endless continuous traversing until stopped. Single: Start-stop operation for a product cycle Counting: Pass for a defined number of product cycles specified by a product count setpoint
machineSpeed	IN	UINT	0	Production velocity in products/min.
numberOfProducts	IN	UDINT	0	Target number of products
resetProductCounter	IN	BOOL	FALSE	Reset product counter
bagLength	IN	REAL	0.0	Bag length / product length
camSyncRange	IN	REAL	0.0	Length of the synchronous range of the controlled cross-sealer axis
camSyncStart	IN	REAL	0.0	Starting position of the synchronous range of the controlled cross-sealer axis
camSyncEnd	IN	REAL	0.0	End position of the synchronous range of the controlled cross-sealer axis
camStretchFactor	IN	REAL	0.0	Compress/stretch factor for the synchronous range of the controlled cross-sealer axis
camCalcFinished	IN	BOOL	TRUE	Cam calculation complete feedback from slave axes
camChange finished	IN	BOOL	TRUE	Cyclic cam replaced feedback from the FBLPacStopperCam block
slaveAxesEnabled	IN	BOOL	TRUE	Axes enabled feedback from slave axes
slaveAxesHomed	IN	BOOL	TRUE	Axes homed feedback from slave axes
slaveAxesInSync	IN	BOOL	TRUE	Axes synchronized feedback from slave axes
valid	OUT	BOOL	FALSE	Block is active
busy	OUT	BOOL	FALSE	Selected functionality being processed
done	OUT	BOOL	FALSE	Selected functionality completed successfully
error	OUT	BOOL	FALSE	Error occurred
toAlarmActive	OUT	BOOL	FALSE	TO alarm occurred

Element	P type [1]	Data type	Initial value	Meaning
errorID	OUT	DWORD	16#0000_0000	Error code
eStopActive	OUT	BOOL	FALSE	Quick stop active
jogForwardSelected	OUT	BOOL	FALSE	Jogging positive direction selected
machineInProduction	OUT	BOOL	FALSE	Machine is running in production operation
posSelected	OUT	BOOL	FALSE	Positioning selected
axisEnabled	OUT	BOOL	FALSE	Axis enabled
axisHomed	OUT	BOOL	FALSE	Axis homed
axisStandStill	OUT	BOOL	FALSE	Axis at standstill
axisInPosition	OUT	BOOL	FALSE	Axis in position
axisInSync	OUT	BOOL	FALSE	Axis synchronous to the master axis
startCamCalc	OUT	BOOL	FALSE	Start cam calculation on slave axes
enableSlaveAxes	OUT	BOOL	FALSE	Enable slave axes
executeHomeSlaveAxes	OUT	BOOL	FALSE	Home slave axes
syncNPNB	OUT	BOOL	FALSE	Synchronize "No Product No Bag" axis
syncFoilFeed	OUT	BOOL	FALSE	Synchronize foil feed
syncCrossSealer	OUT	BOOL	FALSE	Synchronize cross-sealer
syncTappetChain	OUT	BOOL	FALSE	Synchronize pocket-type conveyor
deSyncSlaveAxes	OUT	BOOL	FALSE	Desynchronize following axes
executeCamCalc	OUT	BOOL	FALSE	Request cam recalculation for new product length
executeCamChange	OUT	BOOL	FALSE	Request to FBLPacStopperCam block to change cyclic cam at cycle end for new product length
actNumberOfProducts	OUT	UDINT	0	Number of produced products (actual count)
actValues	OUT	sLPacActValuesMasterContType	-	Actual values
diagnosticData	OUT	sLMCBasicDiagnosticDataType	-	Diagnostic structure, see Section Alarm and error messages (Page 333)
parameter	IN_OUT	sLPacParaMasterContType	-	Parameter for master axis

1) Parameter types: IN = input parameter, OUT = output parameter, IN_OUT = in-out parameter

Table 4- 9 sLPacActValuesMasterContType structure

Element	Data type	Meaning
r32ActualPostion	REAL	Current position based on the product length
r32ActualVelocity	REAL	Current axis velocity based on the product length
r32ActualBagSpeed	REAL	Current production speed in products/min

Table 4- 10 sLPacParaMasterContType structure

Element	Data type	Meaning
sTOs	sLPacToGearDataType	TO for master axis and, if required, leading axis
sMotionData	sLPacFBMasterContMotionDataType	Parameters for EStop, homing, jog, pos, gearing and offset adjustment
sConfigData	sLPacFBMasterContConfigDataType	Configuration data for master

Table 4- 11 sLPacToGearDataType structure

Element	Data type	Meaning
toAxis	followingAxis	Master axis TO (synchronous axis)
toLeadingValue	MasterType	Leading axis for synchronous operation

Table 4- 12 sLPacFBMasterContMotionDataType structure

Element	Data type	Meaning
sEStopData	sLMCBasicEstopDataType	Parameter for quick stop
sHomingData	sLMCBasicHomingDataType	Parameter for homing
sJogData	sLPacJogDataType	Parameter for jogging
sPosData	sLPacPosDataType	Parameter for positioning
sGearingData	sLPacGearingDataType	Parameter for gearing
sPhasingData	sLMCBasicPhasingDataType	Parameter for offset adjustment

Quick stop parameter – sEStopData structure

See Axis quick stop (Page 43)

Homing parameter – sHomingData structure

See Home axis (Page 35)

Table 4- 13 Different default settings:

Element	Data type	Initial value	Meaning
eIncHoming-Mode	eLMCBasicHoming ModeType	DIRECT	Homing mode; set the current position value.

Jog parameter – sJogData structure

See Jog axis (Page 36)

Positioning parameter – sPosData structure

See Position axis (Page 37)

Gearing parameter – sGearingData structure

See Synchronize/desynchronize axis in gearing (Page 38)

Offset adjustment parameter – sPhasingData structure

See Offset adjustment (Page 43)

Table 4- 14 sLPacFBMasterContConfigDataType structure

Element	Data type	Initial value	Meaning
boStartStop-Complete	BOOL	TRUE	TRUE: Desynchronize coupled following axes / units from the master block when stopping the production (enableProduction = FALSE) and reset the axis enable FALSE: Do not desynchronize coupled following axes / units from the master block when stopping the production (enableProduction = FALSE) and do not reset the axis enable
r32AccDec	REAL	0.0	Acceleration and deceleration to start and stop the production or when the production is interrupted (interruptProduction)
r32StopPosition	REAL	0.0	Stop position (must be within the range 0 … bagLength (is converted automatically based on the modulo length)

Element	Data type	Initial value	Meaning
r32maxIncBagLengthChg	REAL	1.0	Limitation for "on the fly" product length change, i.e. maximum value for product length change during production operation
sAxisMechani-calData	sLPacAxisMechani-calDataType		Mechanical data

Table 4- 15 sLPacAxisMechanicalDataType structure

Element	Data type	Initial value	Meaning
r32ModuloLength	REAL	1000.0	Modulo length
boModuloActive	BOOL	TRUE	Activate modulo length

Controlling the machine units

The block provides a predefined control sequence for control of the subordinate subunits of a typical machine configuration. This allows the block based on the "*valid*" initial state, to control the subunits (equipment module) normally available on a horizontal or vertical tubular bagging machine for starting a production from the initial state. The block therefore takes over the complete required control and coordination of the subunits. It controls in a sequencer the axis enables, the homing and the subsequent synchronization. Once the initial state of the machine is established, the block starts the production in the preselected mode (single product, continuous or production of a specified quantity of products). After reaching the stop condition, the block stops the machine again at the specified stop position and ensures the return of the subunits to the initial state (desynchronize and lock axes) when this has been preselected via the *boStartStopComplete* configuration data. If the subunits are not desynchronized by the block and the axis enables not reset, then the above configuration data must be set to FALSE. If return to the initial state is not wanted, selecting *interruptProduction* interrupts the production that can be continued if required. If the production is interrupted, the complete machine can be jogged in the production direction by controlling the *jogForward* input signal.

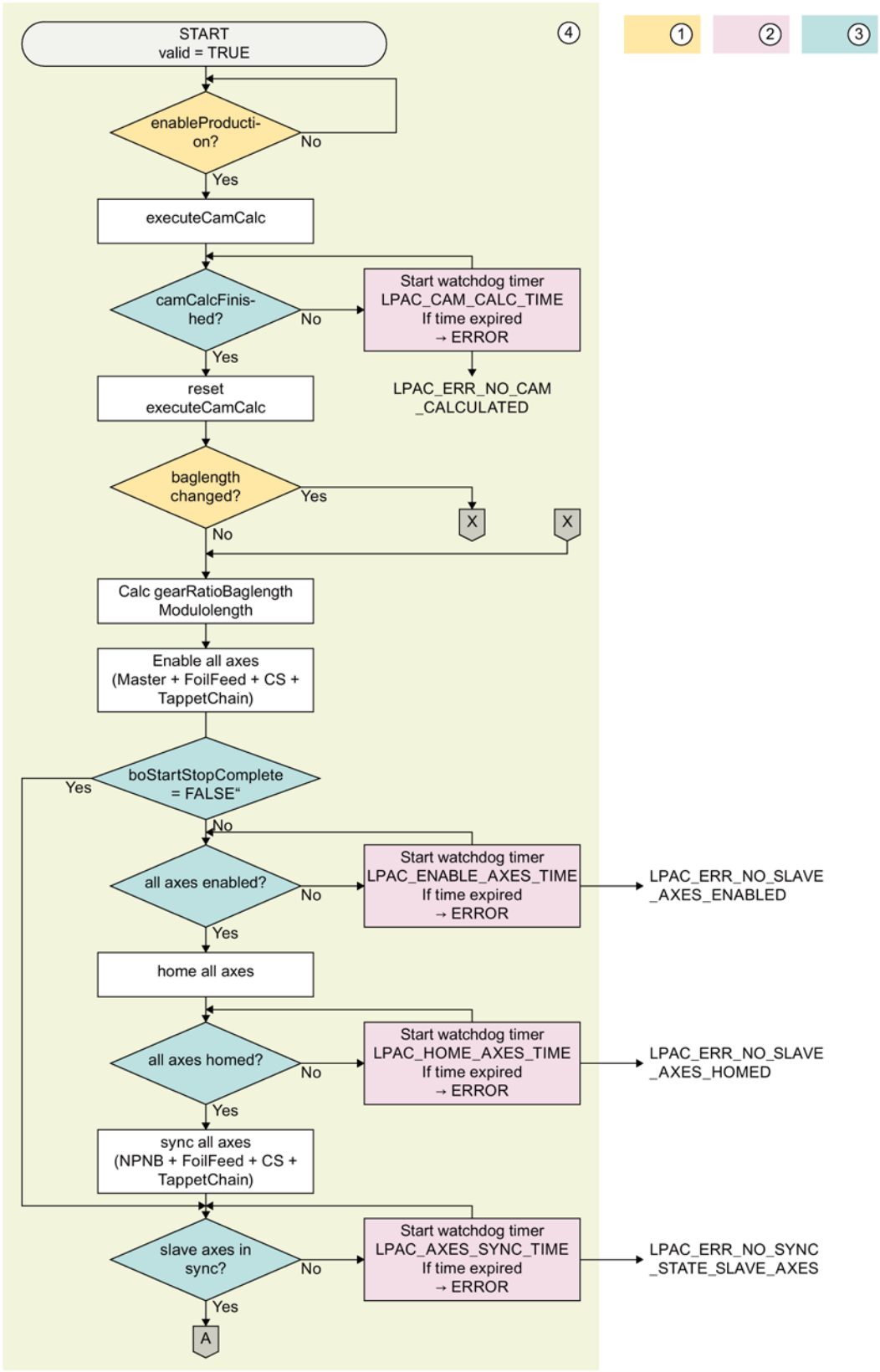
Note

The automatic control sequence for the coupled subunits implemented in the block can be used not only to control the subunits of the vertical and horizontal tubular bagging machines mentioned above, but, because of the generally similar structures, also for many other machine types in the packaging machine sector.

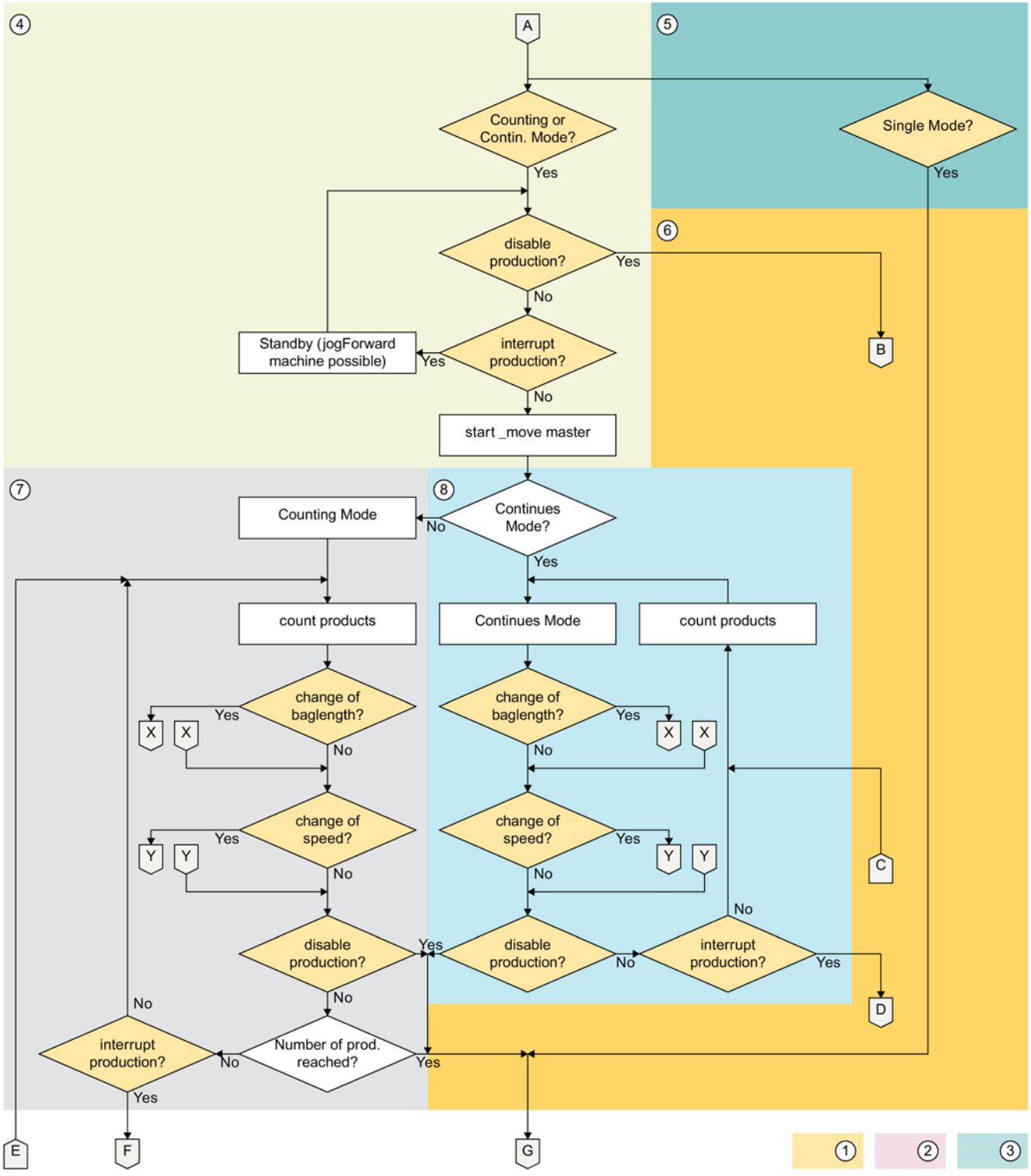
If the control sequence is not suitable or is not used, the relevant sequences can be skipped through permanent interconnection of the feedback inputs with the TRUE level and the block still be used for the control of a master axis.

During running production operation, the machine speed and the bag length can be changed (to a certain extent). The block automatically controls these change and transition states. The sequences are shown in the following flow diagram.

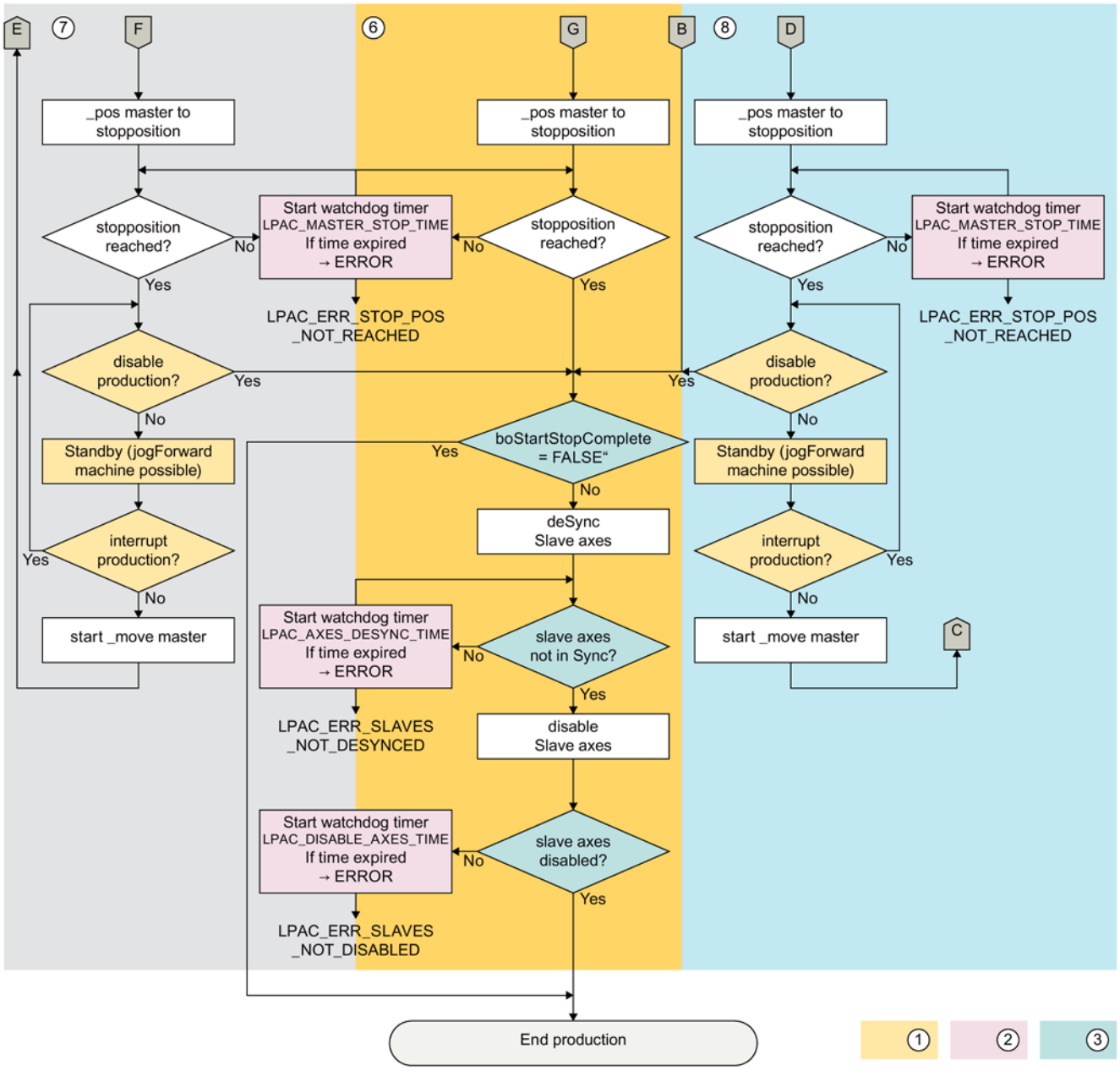
The following figure shows the start sequence of the standalone production with enableProduction.


START
valid = TRUE
4
1
2
3
enableProducti-
on?
No
Yes
executeCamCalc
camCalcFinis-
hed?
No
Start watchdog timer
LPAC_CAM_CALC_TIME
If time expired
→ ERROR
LPAC_ERR_NO_CAM
_CALCULATED
Yes
reset
executeCamCalc
baglength
changed?
Yes
No
X
X
Calc gearRatioBaglength
Modulolength
Enable all axes
(Master + FoilFeed + CS +
TappetChain)
boStartStopComplete
= FALSE“
Yes
No
all axes enabled?
No
Start watchdog timer
LPAC_ENABLE_AXES_TIME
If time expired
→ ERROR
LPAC_ERR_NO_SLAVE
_AXES_ENABLED
Yes
home all axes
all axes homed?
No
Start watchdog timer
LPAC_HOME_AXES_TIME
If time expired
→ ERROR
LPAC_ERR_NO_SLAVE
_AXES_HOMED
Yes
sync all axes
(NPNB + FoilFeed + CS +
TappetChain)
slave axes in
sync?
No
Start watchdog timer
LPAC_AXES_SYNC_TIME
If time expired
→ ERROR
LPAC_ERR_NO_SYNC
_STATE_SLAVE_AXES
Yes
A

The following figure shows the production operation of the standalone production with enableProduction.

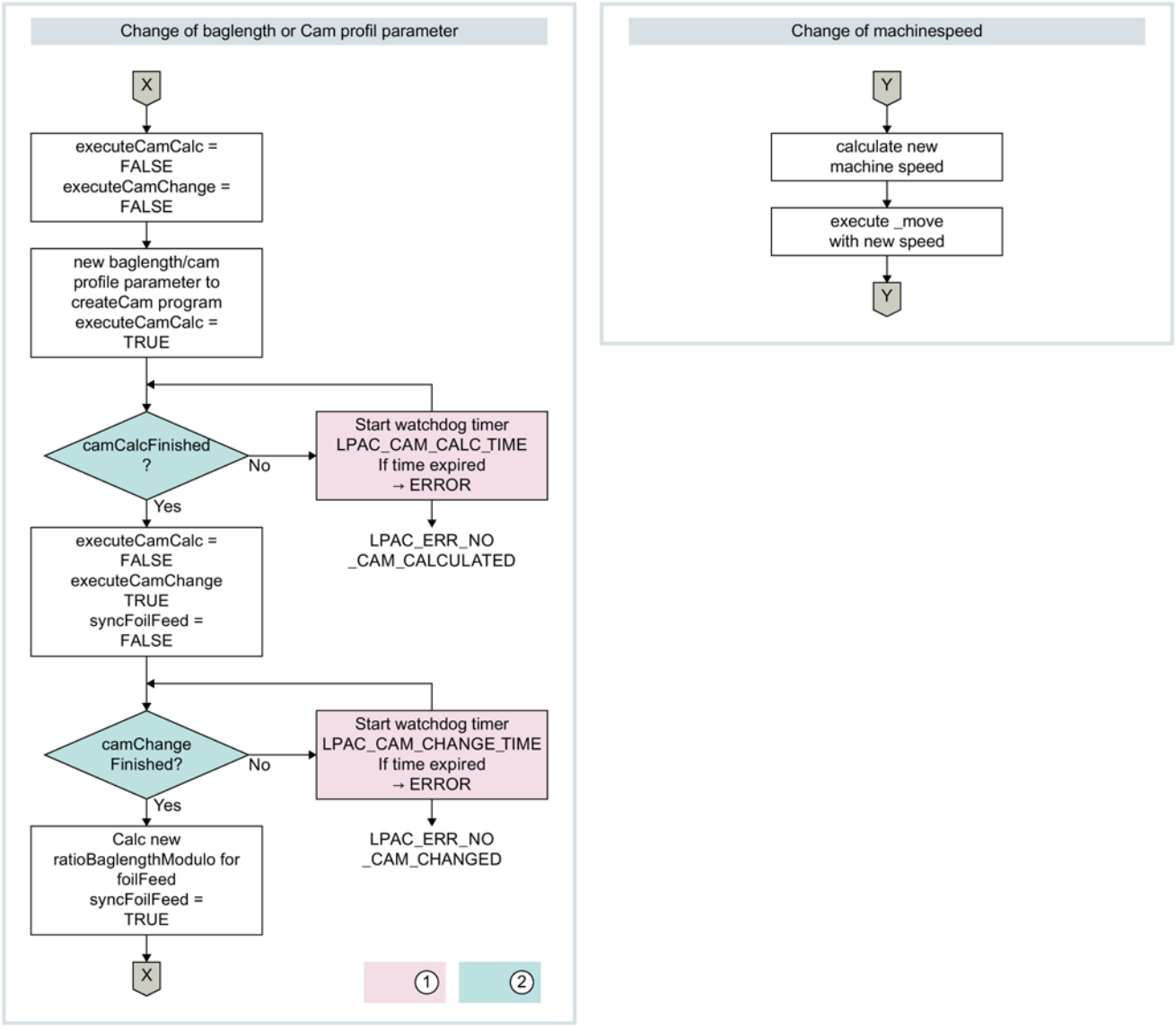


The following figure shows the stop sequence of the standalone production with enableProduction.



① Operator intervention expected/possible
② Fault state possible
③ External feedback expected
④ Start sequence
⑤ Production of a single product
⑥ Stop sequence
⑦ Continuous production
⑧ Production, products counted
X See fig. "Change of bag length sequence"
Y See fig. "Change of production speed sequence"

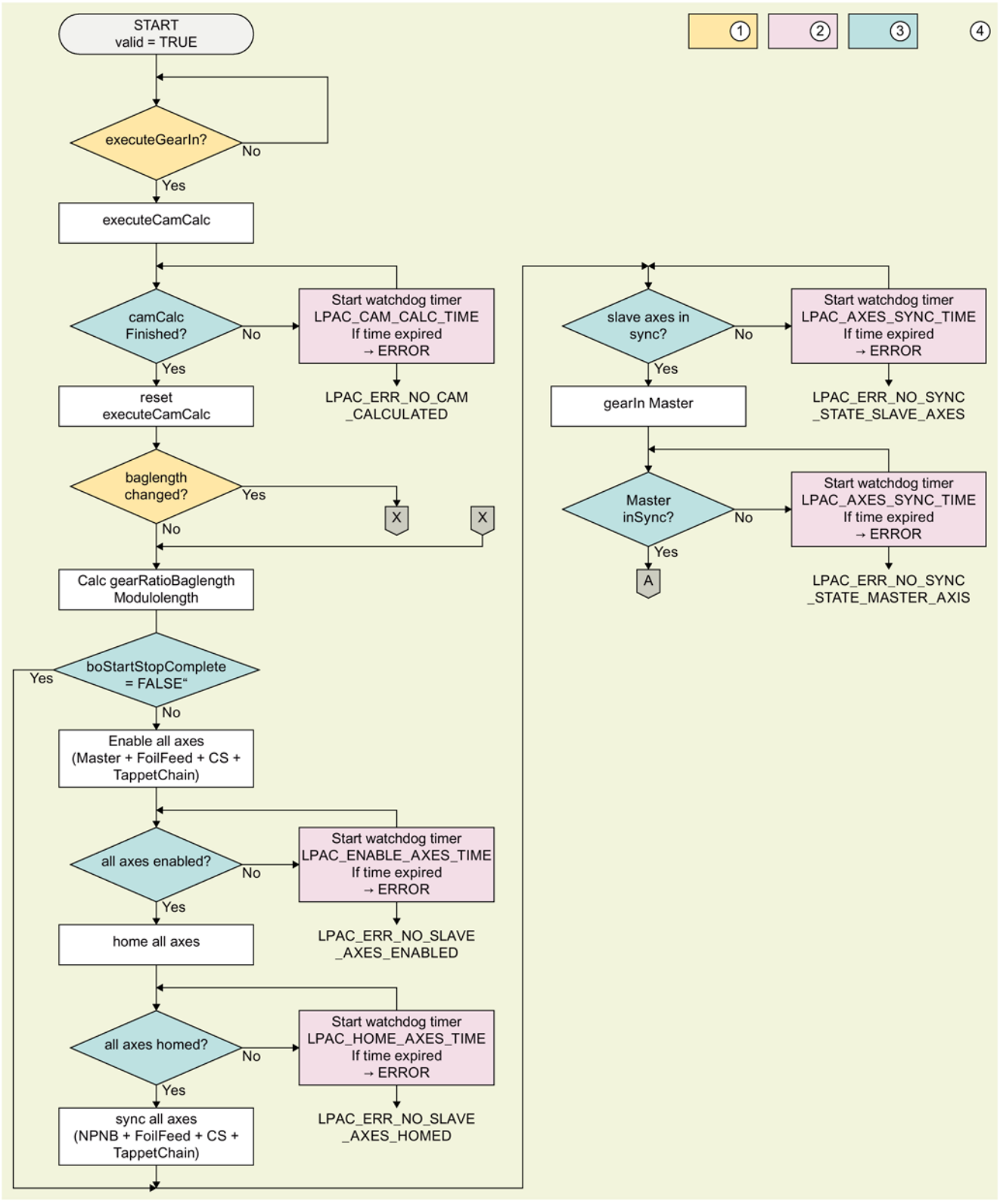
The following figures show the "Change of bag length sequence" and the "Change of production speed sequence".



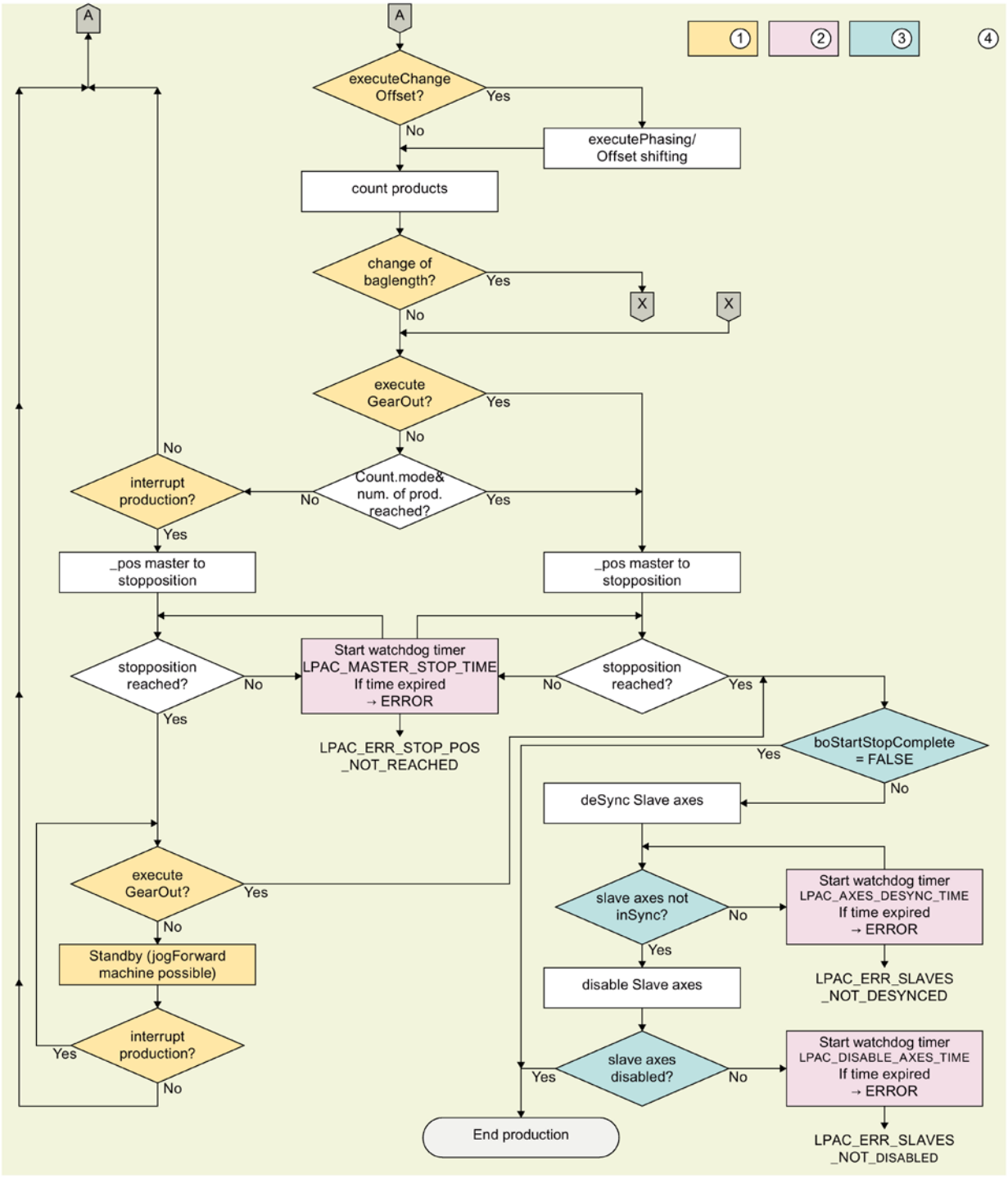
① Fault state possible

② External feedback expected

The following figure shows the start sequence of the "Sequencer for coupled operation". The coupling is performed via gearing on an upstream machine:



The following figure shows the production operation and stop sequence of the "Sequencer for coupled operation".



① Operator intervention expected/possible
② Fault state possible
③ External feedback expected
④ Start sequence
X See fig. "Change of bag length sequence"

Mapping concept of the product length on the modulo length of the master axis

A linear axis with a modulo length of 1000 is used in the **FBLPacMasterCont** master block. The respective product length (bag length) is mapped on this modulo length in the software. In this way, it is not necessary to change the modulo length when the product length is changed, whereby the product length can be changed during axis operation (in small incremental steps). The mapping is performed for coupled camming axes by the output of a trigger signal at the *executeCamCalc* output for the calculation of a new cam which takes into account the changed product length, as well as a trigger signal at the *executeCamChange* output for the **FBLPacStopperCam** block which therefore implements the change between the two cyclic cams at the next end of cycle. At the same time as the change of the two cams, the current gear ratio is activated via the *syncFoilFeed* output. The corresponding implementation and interconnection of the blocks and signals to be controlled has already been shown as an example in the relevant **LPac** program units.

If, for example, a rotary axis with the usual 360 degrees is to be used instead of the linear axis type, this can also be implemented through appropriate parameterization and specification of the modulo length in the startup program.

Changing the bag length during running production operation (on the fly)

During running production operation, the bag length can be changed online in small steps. The maximum increment is limited by a parameterizable limit value in the *r32maxIncBagLengthChg* configuration data.

For a permitted change, the block initiates the calculation of a second cyclic cam by the **FBLPacCreaCam…** blocks and its activation by the **FBLPacStopperCam** cam change block. With activation of the new cam, the new gear ratio is activated simultaneously for the foil feed. The sequence is shown in the *Changing the machine velocity or product length for a running machine* figure.

Note

Note that for "on the fly" bag length changes, only small changes, i.e. just a few mm, are permitted. Otherwise, sudden axis reactions are the result.

Quick stop of the machine

With the appropriate block interconnection by the user, the axis formation can be stopped synchronously in coupled operation through control of the *enableEStop* input signal.

For this purpose, the parameters for the quick stop of the master axis must be set appropriately.

The sequence of a software quick stop in the **FBLPacMasterCont** block is shown in the following figure.

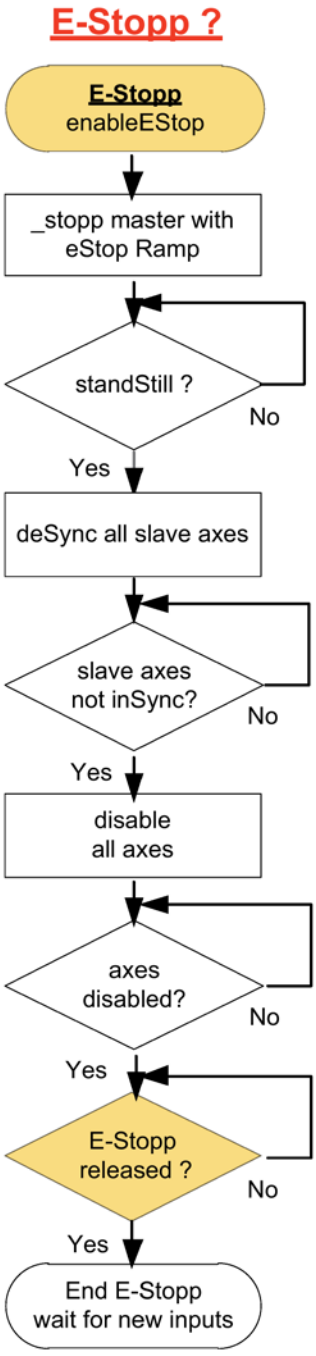
Note

See also Section Axis quick stop (Page 43).

The triggering of a quick stop in the software should only be considered as an additional stop option. The legal requirements of the relevant guidelines for safety functions must be used and observed at all times!

They usually require a quick stop implemented in the hardware.

The following figure shows the "Quick stop" sequence.


E-Stopp ?
E-Stopp
enableEStop
_stopp master with
eStop Ramp
standStill ?
No
Yes
deSync all slave axes
slave axes
not inSync?
No
Yes
disable
all axes
axes
disabled?
No
Yes
E-Stopp
released ?
No
Yes
End E-Stopp
wait for new inputs

4.3.1.3 Integration of the module

Technology objects, basic parameterization and block call

Table 4- 16 TO type and basic parameterizations

Axis / technology object	
TO type	Virtual or real position or following axis
Version	Linear axis
Modulo	The LPac project works with a modulo length of 1000.
Block call	BackgroundTask

Further blocks and libraries

The **FBLPacMasterCont** block uses the following blocks internally:

Block/library	Function
FBLMCBasicAxis / LMCBasic	Axis function block
FBLPacChangeTOConfigDataMod / LPac	Function block to change the configuration data of the TO axis

Some of the internally used blocks originate from other libraries. See Section Additional required libraries (Page 15).

Error messages

Error number [HEX]	Meaning
16#0000	LPAC_ERR_NO; error-free
16#8000	LPAC_ERR_INVALID_STEP; internal error, please get in touch with your Siemens contact partner
16#8001	LPAC_ERR_NO_AXIS; no axis TO for own axis specified
16#8003	LPAC_ERR_TO_ALARM_ACTIVE; TO alarm active; error message from axis function block Error description, see *SIMOTION Axis Function Block* Application Manual and Section Alarm and error messages (Page 333) in this manual
16#8004	LPAC_ERR_AXIS_NOT_DISABLED; axis not disabled for change of the modulo length
16#8005	LPAC_ERR_MODULO_LENGTH; modulo length ≤ 0.0
16#8006	LPAC_ERR_BAGLENGTH_ZERO; bag length ≤ 0.0
16#8008	LPAC_ERR_POS_VELOCITY_ZERO; velocity setpoint during positioning = 0.0
16#8009	LPAC_ERR_POS_ACC_DEC_ZERO; acceleration setpoint and/or deceleration set-point during positioning = 0.0
16#800A	LPAC_ERR_ACCELERATION_ZERO_OR_NEGATIVE; acceleration for production start ≤ 0.0

Error number [HEX]	Meaning
16#800B	LPAC_ERR_POS_ABS_POSITION_VALUE; target position for absolute positioning < 0.0 or > modulo length
16#800C	LPAC_ERR_ PHASING_DYNAMIC_DATA; offset adjustment for gearing; invalid velocity and/or acceleration parameters
16#8010	LPAC_ERR_NO_MASTER_AXIS; no axis TO for leading axis specified
16#8011	LPAC_ERR_AXIS_FUNCTION_BLOC; error message from axis function block For details, see *diagnosticData.b32ErrorCode* output structure. Error description, see *SIMOTION Axis Function Block* Application Manual
16#8015	LPAC_ERR_AXIS_DISABLED_WHILE_MOVING; axis enable reset externally while axis in motion
16#8020	LPAC_ERR_MACHINE_SPEED_ZERO; production speed ≤ 0.0
16#8023	LPAC_ERR_NUMBER_OF_PRODUCTS_ZERO; target number of products ≤ 0 in COUNTING mode
16#8025	LPAC_ERR_NO_SLAVE_AXES_ENABLED; following axis not enabled within the monitoring time 1)
16#8026	LPA_ERR_NO_SYNC_STATE_SLAVE_AXES; following axes not synchronized within the monitoring time 1)
16#8027	LPAC_ERR_NO_CAM_CALCULATED; cross-sealer cam not calculated within the monitoring time 1)
16#8028	LPAC_ERR_NO_CAM_CHANGED; cross-sealer cyclic cam not changed within the monitoring time 1)
16#8029	LPAC_ERR_NO_SLAVE_AXES_HOMED; following axes not homed within the monitoring time 1)
16#802A	LPAC_ERR_INVALID_BAGLENGTH; requested bag length change greater than limit value *r32maxIncBagLengthChg*
16#802B	LPAC_ERR_STOP_POS_NOT_REACHED; stop position not reached within the monitoring time at production stop 1)
16#802C	LPAC_ERR_SLAVES_NOT_DESYNCED; slave axes not desynchronized within the monitoring time at production stop 1)
16#802D	LPAC_ERR_SLAVES_NOT_DISABLED; slave axes not disabled within the monitoring time at production stop 1)
16#8039	LPAC_ERR_NO_SYNC_STATE_MASTER_AXIS; master axis not synchronized within the monitoring time 1)
16#80C3	LPAC_ERR_EXECUTE_DURING_RESET; Execute requested while error acknowledgment active
16#8100	LPAC_ERR_READ_SYSTEM_VARIABLE; error when reading the system variables / variable cannot be read
16#8101	LPAC_ERR_MAX_TIME_FOR_TO_RESTART; TO restart not performed within the monitoring time 1)
16#8153	LPAC_ERR_DECELERATION_ZERO_OR_NEGATIVE; deceleration for production stop ≤ 0.0

1) The constants for the appropriate monitoring times can be found at the end of the cPublic unit, and can be adapted there when required.

Note

The **pMasterCont** (preferably for horizontal machines) and **pMasterContVert** (preferably for vertical machines) program units are available in the LPac project to illustrate the use of the **FBLPacMasterCont** block.

These units contain programs for the required block calls in the StartupTask and BackgroundTask.

4.3.2 FBLPacDelayMaster - drop time compensation of the vertical continuous master

4.3.2.1 Technological consideration

The **FBLPacDelayMasterBGT** and **FBLPacDelayMasterIPO** blocks synchronize the machine axes of the continuous foil feed and continuous cross-sealer with a continuously running dosing unit. The blocks are intended especially for the implementation of a vertical continuous tubular bagging machine that is equipped with a dosing unit coupled in synchronous operation (e.g. Cup dosing unit).

Machines of this type must be able to adapt the changing machine cycle time, e.g. during the acceleration and deceleration phases, to the constant product drop time in such a way that the product discharge from the dosing unit always meets the closed sealing jaws of the cross-sealer.

Together with other blocks of the **LPac** library, the blocks can be used to implement a SIMOTION complete application for a continuous vertical tubular bagging machine.

They control a separate virtual axis (DelayMaster).

As main function when used as so-called "delay" master, the axis is connected between the machine master value and the foil feed or cross-sealer.

The master value vector (velocity, acceleration and position) is first entered in the buffer in the IPO cycle and output from this after the specified delay time (= product drop time) also in the IPO cycle via the MotionIn interface of the dedicated axis (DelayMaster). In this way, the "DelayMaster" always "follows" its leading axis by the programmed delay time.

The following figure shows the position curves of the relevant axes of a typical machine configuration.

The following axis position curves are displayed:

1. Machine leading axis (virtual master controlled by FBLPacMasterCont) – orange
2. Dosing axis; in 1:1 gearing coupled to the machine leading axis – yellow
3. DelayMaster axis; in 1:1 gearing coupled to the machine leading axis; start and stop each delayed by the product drop time (1 s in this case) compared to the machine leading axis – green
4. NoProductNogBag axis; may not be relevant in this case – blue

5. Foil feed axis; coupled in gearing to the DelayMaster axis – light blue
6. Cross-sealer axis; rotating cross-sealer coupled in camming to the DelayMaster axis – ochre
7. Cross-sealer axis, velocity curve; rotating cross-sealer coupled in camming to the DelayMaster axis – ochre
8. Home following axis request signal from master block – violet

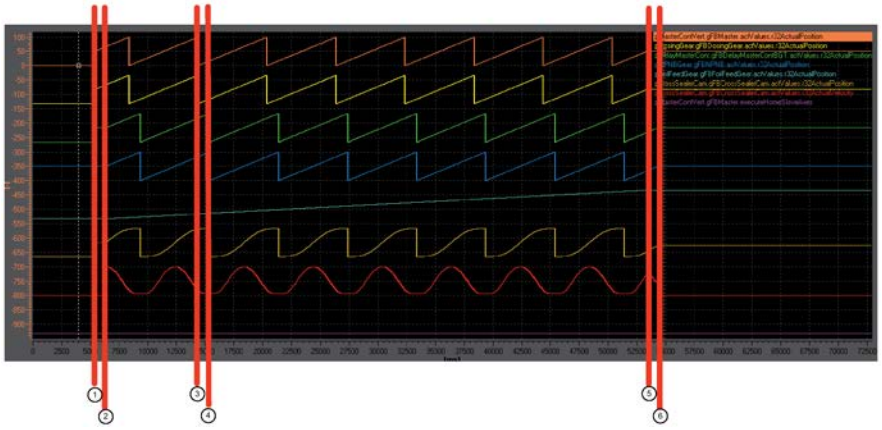


Figure 4-3 Axis position curves of production cycle with FBLPacDelayMaster

The axis position curve is displayed as of homing or start production through to the production stop for the following production parameters:

- Product length = bag length = 100 mm
- Production speed = 10 bags/min
- Product drop time = 1 s

Note that, for test purposes, the displayed axes are virtual axes. For this reason, the homing curve is displayed as actual value setting.

Homing brings the axes into the following positions:

- Master: ½ product length = 50 mm
- Dosing axis: 180° = ½ modulo length
- DelayMaster: ½ modulo length = 500
- Foil feed: 0 mm
- Cross-sealer: 180° = ½ modulo length = open position

The following times in the diagram are described below:

1. Start of production: Machine leading axes and dosing axes start from their home positions without delay.
2. Start of cross-sealer and foil feed: Both following axes start delayed by the product drop time.
3. Modulo break of the machine leading axis and dosing unit: This position is defined as the discharge position/time for the product.

4. Cross-sealer closed position; after the product discharge (3), the cross-sealer has the product drop time to travel to the closed position = sealing position (cross-sealer sealing position = 0°).
5. Stop of production: The production cycle stops at the end of the cycle at the starting position of the machine leading axis = ½ product length = 50 mm
6. Stop of cross-sealer and foil feed; both following axes stop delayed by the product drop time controlled by the DelayMaster.

4.3.2.2 Module description

Scope of functions

The **FBLPacDelayMaster** function block offers the main functions and auxiliary functions as well as possible options shown in the following overview.

Function	Description	Implementation
Main function	Activation of the subordinate machine modules foil feed and cross-sealing for the synchronization of the product discharge with the cross-sealing on vertical continuous tubular bagging machines, delayed by the product drop time	Through the activation of the *executeGearIn* input signal on the FBLPacDelayMasterBGT block, the DelayMaster axis is coupled directly and immediately to the leading axis in 1:1 gearing. The coupling is performed immediately without transition motion or ramp through infeed of the motion vector in the MotionIn interface of the DelayMaster axis. Consequently, coupling can only be performed when the axis is at standstill! Similarly, decoupling (desynchronization) can only be performed when the axes are at standstill via the *executeGearOut* signal. The decoupling of the leading axis is implemented internally via a stop command.
Option	–	–
Auxiliary functions	Enable axis	The axis is enabled via the level-triggered *enableAxis* input.
	Home axis	The homing procedure of the axis is started via the *executeHoming* input.
	Axis quick stop	A quick stop can be performed or a motion of the axis prevented via the level-triggered *enableEStop* input.
	Change of the axis configuration data	The axis configuration data (modulo length, modulo yes/no) can be changed via the subordinate **FBLPacChangeTOConfigDataMod** function block. The modulo length specified in the ramp-up of the controller is automatically entered and activated by the block in the appropriate configuration data element of the axis. The modulo properties can also be changed at any time in the cyclic program by specifying different values. The user must only ensure that the axis is not enabled. A TO restart must be performed to activate the changed configuration data.

Schematic representation in LAD/FBD

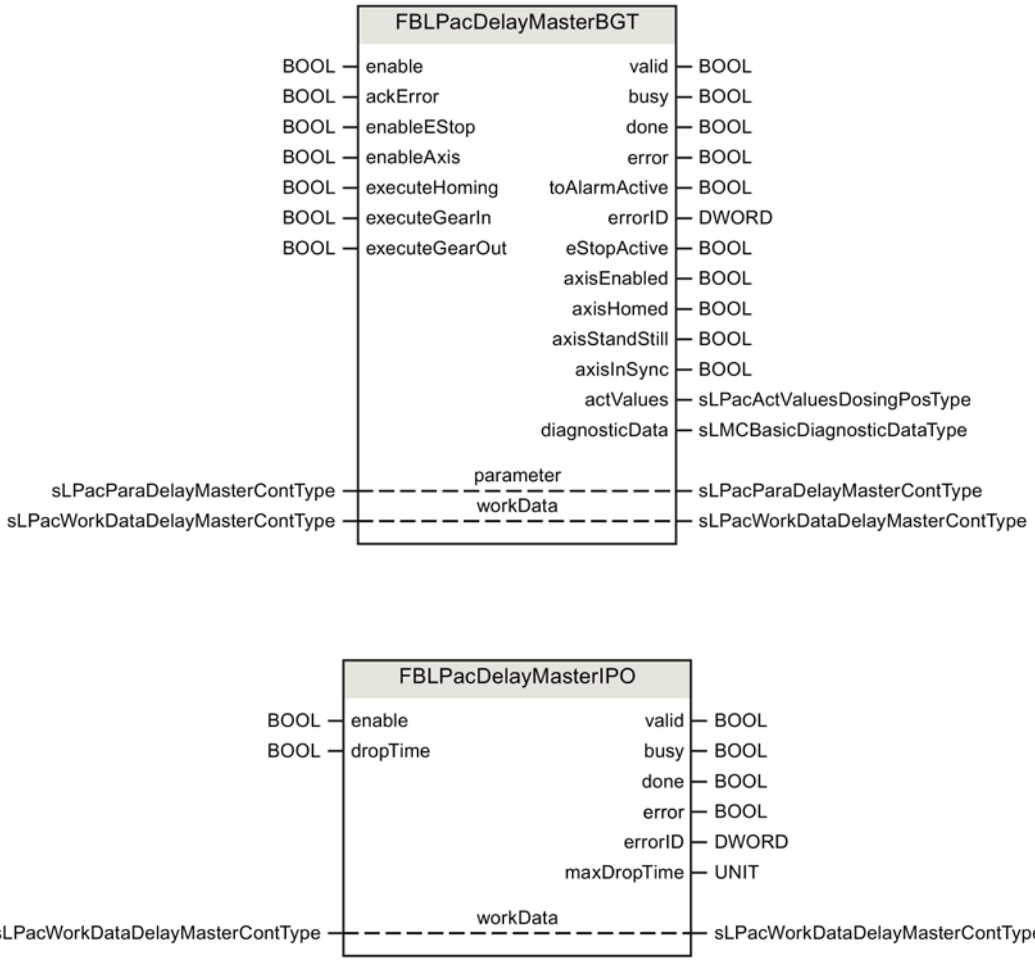


Figure 4-4 FBLPacDelayMaster - schematic block view in LAD/FBD

Input/output parameters

Table 4- 17 Input/output signals

Element	P type [1]	Data type	Initial value	Meaning
enable	IN	BOOL	FALSE	Activation of the FB
ackError	IN	BOOL	FALSE	Acknowledge error
enableEStop	IN	BOOL	FALSE	Activate quick stop
enableAxis	IN	BOOL	FALSE	Enable axis
executeHoming	IN	BOOL	FALSE	Start homing
executeGearIn	IN	BOOL	FALSE	Start synchronization / activate 1:1 gearing

Element	P type [1]	Data type	Initial value	Meaning
executeGearOut	IN	BOOL	FALSE	Start desynchronization / deactivate 1:1 gearing
dropTime	IN	DINT	0	Product drop time [ms]; *only on the FBLPacDelayMasterIPO block*
valid	OUT	BOOL	FALSE	Block is active
busy	OUT	BOOL	FALSE	Selected functionality being processed
done	OUT	BOOL	FALSE	Selected functionality completed successfully
error	OUT	BOOL	FALSE	Error occurred
maxDropTime	OUT	UINT	0	Maximum possible drop time [ms]; *only on the FBLPacDelayMasterIPO block*
toAlarmActive	OUT	BOOL	FALSE	TO alarm occurred
errorID	OUT	DWORD	16#0000_0000	Error code
eStopActive	OUT	BOOL	FALSE	Quick stop active
axisEnabled	OUT	BOOL	FALSE	Axis enabled
axisHomed	OUT	BOOL	FALSE	Axis homed
axisStandStill	OUT	BOOL	FALSE	Axis at standstill
axisInSync	OUT	BOOL	FALSE	Axis synchronous to the master axis
actValues	OUT	sLPacActValuesDosingPosType	-	Current actual velocity and actual position
diagnosticData	OUT	sLMCBasicDiagnosticDataType	-	Diagnostic structure, see Section Alarm and error messages
parameter	IN_OUT	sLPacDelayMasterContType	-	Parameters for delay axis
workData	IN_OUT	sLPacWorkDataDelayMasterContType	-	Data interface to the FBPLacDelayMasterIPO block

1) Parameter types: IN = input parameter, OUT = output parameter, IN_OUT = in-out parameter

Table 4- 18 sLPacActValuesDosingPosType structure

Element	Data type		Meaning
r32ActualVelocity	REAL	0.0	Current velocity
r32ActualPosition	REAL	0.0	Current position

Table 4- 19 sLPacDelayMasterContType structure

Element	Data type	Meaning
sTOs	sLPacToGearDataType	TOs for FBLPacDelayMaster
sMotionData	sLPacDelayMasterContMotionDataType	Motion data for delay axis
sConfigData	sLPacDelayMasterContConfigDataType	Configuration data for the delay axis

Table 4- 20 sLPacToGearDataType structure

Element	Data type	Meaning
toAxis	followingAxis	DelayMaster axis TO (following axis)
toLeadingValue	MasterType	Leading axis for synchronous operation (virtual master)

Table 4- 21 sLPacDelayMasterContMotionDataType structure

Element	Data type	Meaning
sEStopData	sLMCBasicEStopDataType	Parameter for quick stop
sHomingData	sLMCBasicHomingDataType	Parameter for homing

Quick stop parameter – sEStopData structure

See Axis quick stop (Page 43)

Homing parameter – sHomingData structure

See Home axis (Page 35)

Table 4- 22 sLPacDelayMasterContConfigDataType structure

Element	Data type	Initial value	Meaning
sAxisMechanicalData	sLPacAxisMechanicalData	-	Mechanical data

Table 4- 23 sLPacAxisMechanicalDataType structure

Element	Data type	Initial value	Meaning
r32ModuloLength	REAL	0.0	Modulo length
boModuloActive	BOOL	FALSE	Activate modulo length

Table 4- 24 sLPacWorkDataDelayMasterContType structure

Element	Data type	Initial value	Meaning
toAxis	posAxis	TO#NIL	DelayMaster axis TO (following axis)
toMasterAxisInt	posAxis	TO#NIL	Leading axis for synchronous operation (virtual master)
r64Modulo Length	LREAL	0.0	Leading axis modulo length
boAxisInSync	BOOL	FALSE	Delay master synchronized to master
aMaster-ContMotionInS	Array [0.. LPAC_MAX_MOTION-1] OF LREAL[1)]	0.0	Position component MotionIn vector
aMaster-ContMotionInV	Array [0.. LPAC_MAX_MOTION-1] OF LREAL[1)]	0.0	Velocity component MotionIn vector
aMaster-ContMotionInA	Array [0.. LPAC_MAX_MOTION-1] OF LREAL[1)]	0.0	Acceleration component MotionIn vector

1) The constants can be found at the end of the **cPublic** unit, and can be adapted there when required.

4.3.2.3 Embedding the modules

Technology objects, basic parameterization and block call

Table 4- 25 TO type and basic parameterizations

Axis / technology object	
TO type	Following axis
Version	As for leading axis
Modulo	As for leading axis
Block call FBLPacDelayMasterBGT FBLPacDelayMasterIPO	 BackgroundTask IPOSynchronousTask

Further blocks and libraries

The **FBLPacDelayMasterBGT** and **FBLPacDelayMasterIPO** blocks use the following blocks internally.

Block/library	Function
FBLMCBasicAxis / LMCBasic	Axis function block
FBLPacChangeTOConfigDataMod / LPac	Function block to change the configuration data of the TO axis
FBRFG / LConLib	Ramp-function generator

Some of the internally used blocks originate from other libraries. See Section Additional required libraries (Page 15).

Error messages

Error number [HEX]	Meaning
16#0000	LPAC_ERR_NO; error-free
16#8000	LPAC_ERR_INVALID_STEP; internal error, please get in touch with your Siemens contact partner
16#8001	LPAC_ERR_NO_AXIS; no axis TO specified for own axis
16#8003	LPAC_ERR_TO_ALARM_ACTIVE; TO alarm active; error message from axis function block Error description, see *SIMOTION Axis Function Block* Application Manual and Section Alarm and error messages in this manual
16#8004	LPAC_ERR_AXIS_NOT_DISABLED; axis not disabled for change of the modulo length
16#8005	LPAC_ERR_MODULO_LENGTH; modulo length ≤ 0.0
16#8010	LPAC_ERR_NO_MASTER_AXIS; no axis TO specified for master axis
16#8011	LPAC_ERR_AXIS_FUNCTION_BLOC; error message from axis function block For details, see *diagnosticData.b32ErrorCode* output structure. Error description, see *SIMOTION Axis Function Block* Application Manual
16#80C3	LPAC_ERR_EXECUTE_DURING_RESET; Execute requested while error acknowledgment active (message from FBLPacChangeToConfigDataMod block)
16#8100	LPAC_ERR_READ_SYSTEM_VARIABLE; error when reading the system variables / variable cannot be read (message from FBLPacChangeToConfigDataMod block)
16#8101	LPAC_ERR_MAX_TIME_FOR_TO_RESTART; TO restart not performed within the monitoring time (message from FBLPacChangeToConfigDataMod block)

Note

The pDelayMasterCont program unit is available in the LPac project to illustrate the use of the FBLPacDelayMaster… blocks.

This unit contains the programs for the required block calls in the StartupTask, the BackgroundTask and the IPOSynchronousTask.

4.3.3 FBLPacMasterInt - intermittent master

4.3.3.1 Technological consideration

The **FBLPacMasterInt** block simulates the "intermittent master" machine module on intermittently working vertical tubular bagging machines. It can be used together with other blocks of the **LPac** library to implement a SIMOTION complete application of an intermittent tubular bagging machine.

The **FBLPacMasterInt** block does not activate any dedicated axis, but rather uses the specified production parameters, bag length, clock rate, sealing and drop time, for the coordinated activation of the actual equipment modules of a vertical intermittent tubular bagging machine. These are the clocked foil feed operating in positioning mode implemented with **FBLPacFoilFeedPos** and the cross-sealer implemented with **FBLPacCrossSealPos** also operating in positioning mode. The block also implements the interface to the proportioning scale (discharge control).

The intermittent (clocked) production operation is available as main function for use as intermittent master. An integrated product counter is available for use in production machines. The complete machine can also be operated in the "continuous operation", "single product" and "product counting" modes.

Within a complete application, the controlled equipment modules are activated via a digital interface. The software shutdown (both normal stop and quick stop) is performed in coupled operation when the blocks of the complete machine have been interconnected appropriately via the master block.

4.3.3.2 Module description

Scope of functions

The **FBLPacMasterInt** function block offers the main functions and auxiliary functions as well as possible options shown in the following overview.

Function	Description	Implementation
Main function	Activation of the subordinate foil feed, cross-sealing and dosing unit (product discharge) machine modules	The production operation is started via the level-active "*enableProduction*" input. Resetting the enableProduction signal causes stoppage at the next available cycle end (product end).
Options	"Continuous operation", "Single product" or "Product counting" modes	The machine can be operated in the following modes by specifying the operating modes via the *machineMode* input:
		Continued: Endless production until stopped
		Single: Start-stop operation for a product cycle
		Counting: Pass for the defined number of product cycles specified via the product count setpoint

Function	Description	Implementation
	Product counter	The integrated product counter counts the number of product cycles and can be reset per command
Auxiliary functions	Quick stop	A quick stop of the controlled axes can be performed or a motion of the axes prevented via the level-triggered *enableEStop* input.

Schematic representation in LAD/FBD

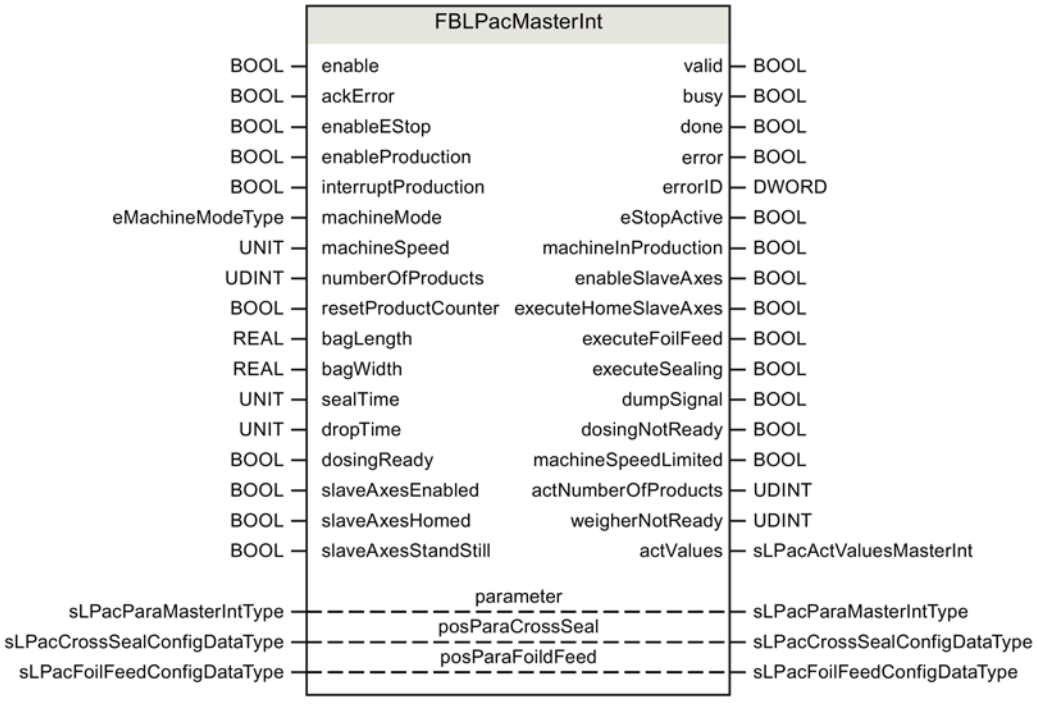


Figure 4-5 Schematic block view in LAD/FBD

Input/output parameters

Table 4- 26 Input/output signals

Element	P type [1)]	Data type	Initial value	Meaning
enable	IN	BOOL	FALSE	Activation of the FB
ackError	IN	BOOL	FALSE	Acknowledge error
enableEStop	IN	BOOL	FALSE	Activate quick stop
enableProduction	IN	BOOL	FALSE	Start or stop production at the next end of cycle that can be reached

Element	P type [1)]	Data type	Initial value	Meaning
machineMode	IN	eMachine-ModeType	contin-ued	Production mode The machine can be operated in the following modes by specifying the modes via the *machineMode* input: Continuous: Endless continuous production until stopped. Single: Start-stop operation for a product cycle Counting: Pass for a defined number of product cycles specified by a product count setpoint
machineSpeed	IN	UINT	0	Production speed in bags/min
numberOfProd-ucts	IN	UDINT	0	Number of products
re-setProductCounter	IN	BOOL	FALSE	Reset product counter
bagLength	IN	REAL	0	Bag length
bagWidth	IN	REAL	0	Bag thickness
sealTime	IN	UINT	0	Sealing time
dropTime	IN	UINT	0	Fall time
dosingReady	IN	BOOL	FALSE	Dosing weigher ready
slaveAxesEnabled	IN	BOOL	TRUE	Axes enabled feedback from slave axes
slaveAxesHomed	IN	BOOL	TRUE	Axes homed feedback from slave axes
slaveAxesStand-Still	IN	BOOL	TRUE	Axes at standstill feedback from slave axes
valid	OUT	BOOL	FALSE	Block is active
busy	OUT	BOOL	FALSE	Selected functionality being processed
done	OUT	BOOL	FALSE	Selected functionality completed success-fully
error	OUT	BOOL	FALSE	Error occurred
errorID	OUT	DWORD	16#000 0 _0000	Error code
eStopActive	OUT	BOOL	FALSE	Quick stop active
machineInProduc-tion	OUT	BOOL	FALSE	Machine in production operation
enableSlaveAxes	OUT	BOOL	FALSE	Enable slave axes
exe-cuteHomeSlave-Axes	OUT	BOOL	FALSE	Home slave axes
executeFoilFeed	OUT	BOOL	FALSE	Start foil feed
executeSealing	OUT	BOOL	FALSE	Start cross-sealer
dumpSignal	OUT	BOOL	FALSE	Product discharge signal
dosingNotReady	OUT	BOOL	FALSE	Dosing not ready

Element	P type [1)]	Data type	Initial value	Meaning
machineSpeedLimchineSpeedLimited	OUT	BOOL	FALSE	Machine velocity has been limited
actNumberOfProducts	OUT	UDINT	0	Number of products
weigherNotReady	OUT	UDINT	0	Number of product discharges for which the dosing unit was not ready
actValues	OUT	sLPacActValuesMasterIntType	-	Actual values
parameter	IN_OUT	sLPacParaMasterIntType	-	Parameter for interm. master
posParaCrossSeal	IN_OUT	sLPacCrossSealConfigDataType	-	Cross-sealer positioning parameter
posParaFoilFeed	IN_OUT	sLPacFoilFeedConfigDataType	-	Foil feed positioning parameter

1) Parameter types: IN = input parameter, OUT = output parameter, IN_OUT = in-out parameter

Table 4- 27 sLPacActValuesMasterIntType structure

Element	Data type	Initial value	Meaning
r32MinCycleTime	REAL	0.0	Minimum cycle time
r32ActCycleTime	REAL	0.0	Current cycle time
r32MinFoilFeedTime	REAL	0.0	Minimum foil feed time
r32ActFoilFeedTime	REAL	0.0	Current foil feed time
r32MinCSCloseTime	REAL	0.0	Minimum cross-sealer closing time
r32ActCSCloseTime	REAL	0.0	Current cross-sealer closing time
r32MinCSOpenTime	REAL	0.0	Minimum cross-sealer opening time
r32ActCSOpenTime	REAL	0.0	Current cross-sealer opening time
r32OverlapTimeFFEnd	REAL	0.0	End of current overlap time between cross-sealer closing and foil feed
r32OverlapTimeFFReStart	REAL	0.0	Start current overlap time between cross-sealer opening and foil feed
r32FFVelOverride	REAL	0.0	Foil feed velocity override
r32FFAccOverride	REAL	0.0	Foil feed acceleration override
r32CSCloseVelOverride	REAL	0.0	Cross-sealer closing motion velocity override
r32CSCloseAccOverride	REAL	0.0	Cross-sealer closing motion acceleration override

Element	Data type	Initial value	Meaning
r32CSOpenVelOverride	REAL	0.0	Cross-sealer opening motion velocity override
r32CSOpenAccOverride	REAL	0.0	Cross-sealer opening motion acceleration override

Table 4- 28 sLPacParaMasterIntType structure

Element	Data type	Meaning
sConfigData	sLPacMasterIntConfigDataType	Configuration data for intermittent master

Table 4- 29 sLPacMasterIntConfigDataType structure

Element	Data type	Initial value	Meaning
boStartStop-Complete	BOOL	TRUE	TRUE: Reset the axis enable for coupled following axes / units when stopping the production (enableProduction = FALSE) FALSE: Do not reset the axis enable for coupled following axes / units when stopping the production (enableProduction = FALSE)
r32EndFoilFeed AtPosition	REAL	0.0	Motion overlap between end of foil feed and start of cross-sealer closing motion; the position of the cross-sealer must be specified at which the foil feed motion should be completed
r32StartFoilFeed AtPosition	REAL	0.0	Motion overlap between start of foil feed and start of cross-sealer opening motion; the position of the cross-sealer must be specified at which the foil feed motion should be restarted

Table 4- 30 sLPacCrossSealConfigDataType structure

Element	Data type	Initial value	Meaning
r32OpenPosition	REAL	0.0	Opening position of cross-sealer
r32ClosePosition	REAL	0.0	Closing position of cross-sealer
r32maxVelocity	REAL	0.0	Maximum velocity of cross-sealer
r32maxAcceleration	REAL	0.0	Maximum acceleration of cross-sealer
r32maxDeceleration	REAL	0.0	Maximum deceleration of cross-sealer
sTorqueSealing	sLPacCrossSeal-TorqueDataType	-	Parameter for sealing to torque

Element	Data type	Initial value	Meaning
sCrankShaft	sLPacCrankShaft-DataType	-	Definition data of slider-crank mechanical system
sInterfacePosDa-ta	sLPac-CSPosMaxData Type	-	Positioning parameter for exchange with cross-sealer block FBLPacCrossealerPos

Table 4- 31 Sealing to torque parameters - sLPacTorqueSealingDataType structure

Element	Data type	Initial value	Meaning
r32SealingTorque	REAL	0.0	Setpoint for torque at sealing
r32LocalSealPosition	REAL	0.0	Distance to still be traversed after the closing of the cross-sealer to build up the sealing torque
r32SupPosSealVel	REAL	0.0	Velocity for sealing to torque (close)
r32SupPosSealAcc	REAL	0.0	Acceleration for sealing to torque (close)
r32SupPosSealDec	REAL	0.0	Deceleration for sealing to torque (close)
r32SupPosReleaseVel	REAL	0.0	Velocity for sealing to torque (open)
r32SupPosReleaseAcc	REAL	0.0	Acceleration for sealing to torque (open)
r32SupPosReleaseDec	REAL	0.0	Deceleration for sealing to torque (open)

Table 4- 32 Slider-crank definition parameters - sLPacCrankShaftDataType structure

Element	Data type	Initial value	Meaning
boCSWithCrankShaft	BOOL	FALSE	Specification of cross-sealer with slid-er-crank gear
sCrankShaftData	sLPacCrank-ShaftConfig-DataType	-	Definition data for slider-crank gear

Table 4- 33 Slider-crank definition parameters - sLPacCrankShaftConfigDataType structure

Element	Data type	Initial value	Meaning
r32CrankLength	REAL	0.0	Crank length
r32ConrodLength	REAL	0.0	Conrod length
r32verticalOffset	REAL	0.0	Offset displacement between pivot point of the crank and connecting rod of the mechanical system
r32LinearOffset	REAL	0.0	Linear maximum distance of the slider crank

Table 4- 34 Cross-sealer positioning parameters - sLPacCSPosMaxDataType structure

Element	Data type	Initial value	Meaning
boCSWithCrankShaft	BOOL	FALSE	Specification of cross-sealer with slider-crank gear
sOpenData	sLPac-PosMaxDyn DataType	-	Positioning parameter for opening motion
sCloseData	sLPac-PosMaxDyn DataType	-	Positioning parameter for closing motion

Table 4- 35 sLPacPosMaxDynDataType structure

Element	Data type	Initial value	Meaning
r32Position	REAL	0.0	Open or closed position of the cross-sealer
r32MaxVelocity	REAL	0.0	Maximum velocity of the axis
r32MaxAcceleration	REAL	0.0	Maximum acceleration of the axis
r32MaxDeceleration	REAL	0.0	Maximum deceleration of the axis
r64VelocityOverride	LREAL	0.0	Velocity override
r64Acceleration Override	LREAL	0.0	Acceleration override

Table 4- 36 sLPacFoilFeedConfigDataType structure

Element	Data type	Initial value	Meaning
r32Position	REAL	0.0	Foil feed length (bag length)
r32maxVelocity	REAL	0.0	Maximum foil feed velocity
r32maxAcceleration	REAL	0.0	Maximum foil feed acceleration
r32maxDeceleration	REAL	0.0	Maximum foil feed deceleration
sPrintMarkCorr	sLPac-PrintMark-CorrType	-	Parameter for print mark correction
sParaSlipComp	sLPacSlip-CompType	-	Parameter for slip compensation
sInterfacePosData	sLPacFF-PosMaxDat aType	-	Positioning parameter for exchange with foil feed block FBLPacFoilFeedPos
sAxisMechanicalData	sLPacAx-isMechani-calData	-	Mechanical definition data of the foil feed axis (a linear axis is expected; data does not have to be specified here)

Table 4- 37 sLPacPrintMarkCorrType structure

Element	Data type	Initial value	Meaning
boPrintMarkCorr	BOOL	FALSE	Print mark correction selected
toMeasuringInput	MeasuringInputType	TO#NIL	Technology object for measuring input
r32PrintMarkPos	REAL	0.0	Position of the print mark
r32RemainingWay FromPrintMark	REAL		Distance between print mark and sealing point that should also be traversed
r32MinPrintMarkCorr	REAL	0.0	Minimum print mark deviation as of which a compensating motion is executed
r32MaxPrintMarkCorr	REAL	0.0	Maximum permissible correction
u32MaxValueFor PrintMarkErrors	REAL	1.0	Maximum number of permitted errors for the print mark detection
r32MeasRangePos Start	REAL	0.0	Start of the measuring range
r32MeasRangePosEnd	REAL	0.0	End of the measuring range
r32CalcFact	REAL	1.0	Factor by which the measured path difference is to be compensated
r32Velocity	REAL	0.0	Correction velocity
r32Acceleration	REAL	0.0	Correction acceleration and deceleration
boEdge	BOOL	TRUE	TRUE: Pos. edge is evaluated FALSE: Neg. edge is evaluated
boCorrInvert	BOOL	FALSE	TRUE: Correction inversion FALSE: No correction inversion

Table 4- 38 sLPacSlipCompType structure

Element	Data type	Initial value	Meaning
boSlipComp	BOOL	FALSE	Activate slip compensation
r32Velocity	REAL	0.0	Correction velocity
r32Acceleration	REAL	0.0	Correction acceleration
r32MaxSlip	REAL	0.0	Slip as of which a compensating motion is executed
toExternalEncoder	externalEncoderType	TO#NIL	TO for external encoder

Table 4- 39 Foil feed positioning parameter - sLPacFFPosMaxDataType structure

Element	Data type	Initial value	Meaning
sMaxPosData	sLPacPosMaxDynDataType	-	Maximum values for the positioning motion of the foil feed

Table 4- 40 sLPacAxisMechanicalDataType structure (mechanical system definition of the foil feed axis, should be specified for foil feed, not relevant in this case)

Element	Data type	Initial value	Meaning
r32ModuloLength	REAL	0.0	Modulo length
boModuloActive	BOOL	FALSE	Activate modulo

See also

Jog axis (Page 36)

Axis positioning (Page 37)

Axis rapid stop (Page 43)

4.3.3.3 Integration of the module

Technology objects, basic parameterization and block call

Table 4- 41 TO type and basic parameterizations

Axis / technology object	
TO type	Controlled axes are position axes
Version	Linear axes
Modulo	Not relevant
Block call	Backgroundtask

Further blocks and libraries

The **FBLPacMasterInt** block uses the following blocks internally:

Block/library	Function
FBLPacGetPosTime / LPac	Function block for calculating the positioning times
FBLPacSetOverrideVI / LPac	Function block for override adjustment
FBLPacGetTimeAtPosition / LPac	Function block for calculating the time for a defined position
FBLPacGetCycleTimeVI / LPac	Function block for calculating the cycle time of a vert. interm. tubular bagging machine
FBLPacTranslationToRotation / LPac	Function block for converting a translatory motion to a rotational motion

No blocks from other libraries are used.

Error messages

Error number [HEX]	Meaning
16#0000	LPAC_ERR_NO; error-free
16#8000	LPAC_ERR_INVALID_STEP; internal error, please get in touch with your Siemens contact partner
16#8006	LPAC_ERR_BAGLENGTH_ZERO; bag length ≤ 0.0
16#8007	LPAC_ERR_SEAL_TIME_ZERO; sealing time ≤ 0.0
16#8020	LPAC_ERR_MACHINE_SPEED_ZERO; production speed ≤ 0.0
16#8021	LPAC_ERR_CYCLE_TIME_ZERO; set cycle time and/or minimum cycle time ≤ 0;
16#8025	LPAC_ERR_NO_SLAVE_AXES_ENABLED; at least one following axis not enabled within the monitoring time
16#8029	LPAC_ERR_NO_SLAVE_AXES_HOMED; at least one following axis not homed within the monitoring time
16#802D	LPAC_ERR_SLAVES_NOT_DISABLED; at least one following axis still enabled after the monitoring time
16#802E	LPAC_ERR_SEALTIME_ZERO; sealing time ≤ 0
16#802F	LPAC_ERR_SEALTIME_TOO_LONG; sealing time > minimum cycle time
16#8035	LPA_ERR_OVERRIDE_CALC_FACTOR; ratio of (set cycle time – sealing time) / (minimum cycle time / sealing time) is negative
16#8036	LPAC_ERR_OVERRIDE_CALC_SPEED_IS_LIMITED_TO_MIN; calculated velocity override has been limited to 1% (minimum override)
16#8037	LPAC_ERR_OVERRIDE_CALC_SPEED_IS_LIMITED_TO_MAX; calculated velocity override has been limited to 100% (maximum override)
16#8038	LPAC_ERR_OVERRIDE_CALC_ACC_IS_LIMITED_TO_MAX; calculated acceleration override has been limited to 100% (maximum override)
16#80A9	LPAC_ERR_DROPTIME_ZERO; drop time ≤ 0.0
16#8150	LPAC_ERR_DISTANCE_LESS_THAN_OR_EQUAL_ZERO; parameter *distance* ≤ 0
16#8151	LPAC_ERR_VELOCITY_ZERO_OR_NEGATIVE; parameter *velocity* ≤ 0
16#8152	LPAC_ERR_ACCELERATION_ZERO_OR_NEGATIVE; parameter *acceleration* ≤ 0
16#8153	LPAC_ERR_DECELERATION_ZERO_OR_NEGATIVE; parameter *deceleration* ≤ 0
16#8154	LPAC_ERR_INVALID_PARAMETERS; invalid parameters require division by 0
16#8155	LPAC_ERR_POSITION_GREATER_THAN_DISTANCE; target position is outside total motion
16#8156	LPAC_ERR_ACCELERATION_OVERRIDE_ZERO_OR_NEGATIVE; acceleration override ≤ 0
16#8157	LPAC_ERR_DECELERATION_OVERRIDE_ZERO_OR_NEGATIVE; deceleration override ≤ 0
16#8158	LPAC_ERR_ACCELERATION_TIME_ZERO_OR_NEGATIVE; acceleration time ≤ 0

Error number [HEX]	Meaning
16#8159	LPAC_ERR_DECELERATION_TIME_ZERO_OR_NEGATIVE; deceleration time ≤ 0
16#815C	LPAC_ERR_INVALID_END_FOIL_FEED_AT_POSITION

Note

The **pMasterInt** program unit is available in the LPac project to illustrate the use of the **FBLPacMasterInt** block.

This unit contains the programs for the required block calls in the StartupTask and the BackgroundTask.

4.3.4 FBLPacExtEncoder - external encoder

4.3.4.1 Technological consideration

As a rule, the **FBLPacExtEncoder** block is used for the coupling of a continuously operating machine to an upstream machine or production line via an external encoder. Through the mechanical coupling of the encoder to the upstream machine, the motion of this leading machine is measured and simulated. This master value encoder serves as "leading axis" to which the following machine can run synchronously in gearing or camming.

To implement the coupling, the continuous master block **FBLPacMasterCont** can couple the virtual machine leading axis in gearing to the external encoder. This is performed by controlling the two *executeGearIn* and *executeGearOut* input signals. To control the synchronization/desynchronization motion, the appropriated parameters must be specified on the **FBLPacMasterCont** block or on other following axes to be coupled.

4.3.4.2 Module description

Scope of functions

The **FBLPacExtEncoder** function block provides the main and auxiliary functions, and possibly options, shown in the following overview.

Function	Description	Implementation
Main function	Control of an external encoder as leading axis	Before being used as master value source, the external encoder must be enabled via the level-triggered *enableAxis* input and, if required, homed via the *executeHoming* input
Options	–	–
Auxiliary functions	Enable axis (encoder)	Axis is enabled via the level-triggered *enableAxis* input
	Home axis (encoder)	Homing procedure of the encoder is started via the *executeHoming* input.
	Change of the axis configuration data	The encoder configuration data (modulo length, modulo yes/no) can be changed via the subordinate **FBLPacChangeTOConfigDataMod** function block. The modulo length specified in the ramp-up of the controller is automatically entered and activated by the block in the appropriate configuration data element of the encoder. The modulo properties can also be changed at any time in the cyclic program by specifying different values. The user must only ensure that the encoder is not enabled. A TO restart must be performed to activate the changed configuration data.

Schematic representation in LAD/FBD

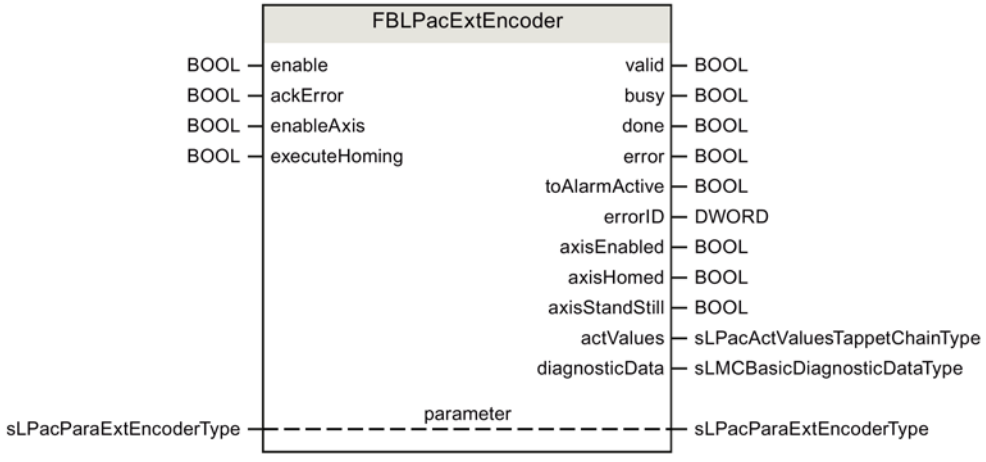


Figure 4-6 FBLPacExtEncoder - schematic block view in LAD/FBD

Input/output parameters

Table 4- 42 Input/output signals

Element	P type [1]	Data type	Initial value	Meaning
enable	IN	BOOL	FALSE	Activation of the FB
ackError	IN	BOOL	FALSE	Acknowledge error
enableAxis	IN	BOOL	FALSE	Enable axis
executeHoming	IN	BOOL	FALSE	Start homing
valid	OUT	BOOL	FALSE	Block is active
busy	OUT	BOOL	FALSE	Selected functionality being processed
done	OUT	BOOL	FALSE	Selected functionality completed successfully
error	OUT	BOOL	FALSE	Error occurred
toAlarmActive	OUT	BOOL	FALSE	TO alarm occurred
errorID	OUT	DWORD	16#0000_0000	Error code
axisEnabled	OUT	BOOL	FALSE	Axis enabled
axisHomed	OUT	BOOL	FALSE	Axis homed
axisStandStill	OUT	BOOL	FALSE	Axis at standstill
actValues	OUT	sLPacActValuesTappetChainType	-	For actual values; see sLPacActValuesTappetChainType structure, page 101
diagnosticData	OUT	sLMCBasicDiagnosticDataType	-	Diagnostic structure, see Section Alarm and error messages (Page 333)
parameter	IN_OUT	sLPacParaExtEncoderType	-	Parameter for external encoder

1) Parameter types: IN = input parameter, OUT = output parameter, IN_OUT = in-out parameter

Table 4- 43 sLPacParaExtEncoderType structure

Element	Data type	Meaning
sTOs	sLPacToExtEncoderDataType	TO for FBLPacExtEncoder
sMotionData	sLPacExtEncoderMotionDataType	Motion data for external encoder
sConfigData	sLPacExtEncoderConfigDataType	Configuration data for external encoder

Table 4- 44 sLPacToExtEncoderDataType structure

Element	Data type	Initial value	Meaning
toEncoder	ExternalEncoderType	TO#NIL	Technology object for external encoder

Table 4- 45 sLPacExtEncoderMotionDataType structure

Element	Data type	Initial value	Meaning
sHomingData	sLMCBasicHoming-DataType	-	Parameter for homing

Homing parameters - sHomingData structure

See Home axis (Page 35)

Table 4- 46 sLPacExtEncoderConfigDataType structure

Element	Data type	Initial value	Meaning
sAxisMechani-calData	sLPacAxisMechani-calDataType	-	Configuration data for axis TO

Table 4- 47 sLPacAxisMechanicalDataType structure

Element	Data type	Initial value	Meaning
r32ModuloLength	REAL	0.0	Modulo length
boModuloActive	BOOL	TRUE	Activate modulo

4.3.4.3 Embedding the module

Technology objects, basic parameterization and block call

Table 4- 48 TO type and basic parameterizations

Axis / technology object	
TO type	External encoder
Version	Rotary axis or linear axis modulo
Modulo	As required, e.g. 360 degrees
Block call	BackgroundTask

Further blocks and libraries

The **FBLPacExtEncoder** block uses the following blocks internally.

Block/library	Function
FBLMCBasicAxis / LMCBasic	Axis function block
FBLPacChangeTOConfigDataMod / LPac	Function block to change the configuration data of the TO axis

Some of the internally used blocks originate from other libraries. See Section Additional required libraries (Page 15).

Error messages

Error number [HEX]	Meaning
16#0000	LPAC_ERR_NO; error-free
16#8000	LPAC_ERR_INVALID_STEP; internal error, please get in touch with your Siemens contact partner
16#8001	LPAC_ERR_NO_AXIS; no axis TO specified for own axis
16#8003	LPAC_ERR_TO_ALARM_ACTIVE; TO alarm active; error message from axis function block Error description, see *SIMOTION Axis Function Block* Application Manual and Section Alarm and error messages (Page 333) in this manual
16#8004	LPAC_ERR_AXIS_NOT_DISABLED; axis not disabled for change of the modulo length
16#8005	LPAC_ERR_MODULO_LENGTH; modulo length ≤ 0.0
16#8011	LPAC_ERR_AXIS_FUNCTION_BLOC; error message from axis function block For details, see *diagnosticData.b32ErrorCode* output structure. Error description, see *SIMOTION Axis Function Block* Application Manual
16#80C3	LPAC_ERR_EXECUTE_DURING_RESET; Execute requested while error acknowledgment active (message from FBLPacChangeToConfigDataMod block)
16#8100	LPAC_ERR_READ_SYSTEM_VARIABLE; error when reading the system variables / variable cannot be read (message from FBLPacChangeToConfigDataMod block)
16#8101	LPAC_ERR_MAX_TIME_FOR_TO_RESTART; TO restart not performed within the monitoring time (message from FBLPacChangeToConfigDataMod block)

Note

The pExtEncoder program unit is available in the LPac project to illustrate the use of the FBLPacExtEncoder block.

This unit contains the programs for the required block calls in the StartupTask and the BackgroundTask.

4.4 Product supply

4.4.1 FBLPacTappetChainGear - gearing supply chain

4.4.1.1 Technological consideration

The **FBLPacTappetChainGear** block controls the "supply chain" (pocket-type conveyor, tappet chain) machine module for horizontal tubular bagging machines. It can be used together with other blocks of the **LPac** library for implementing a SIMOTION complete application of a horizontal tubular bagging machine and also independently for similar mechanically-oriented machine units that require an electronic gearing for a leading axis.

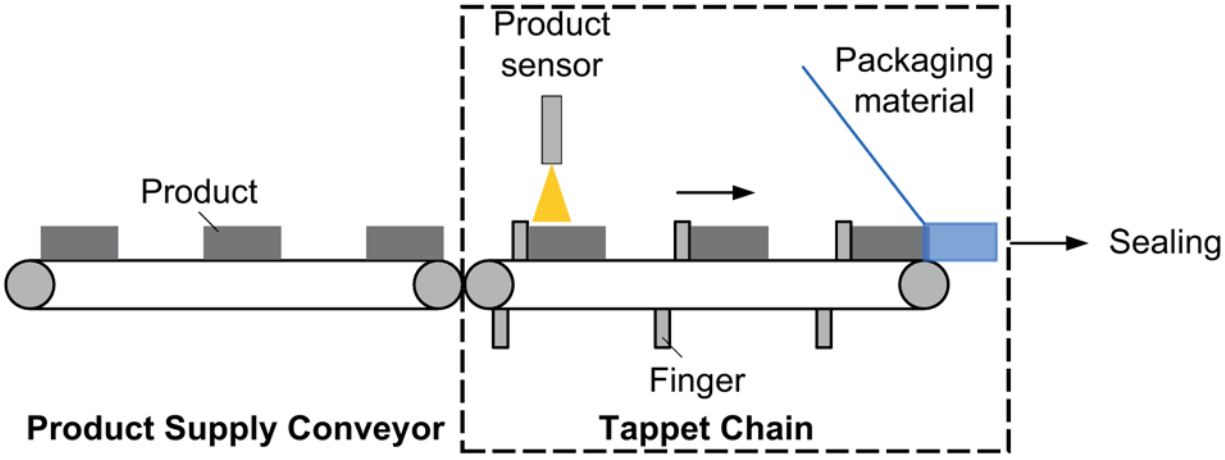


Figure 4-7 Supply chain (tappet chain)

The supply chain is deployed, among other things, in horizontal tubular bagging machines for supplying the product to be packed to the foil tube. Normally, it consists of a rotating transport chain or belt with carrier fingers mounted with the same distance that limit the pockets. A single product is conveyed in the individual pockets of the supply chain. The supply chain then transports the products into the foil tube. To pack a product for each sealing operation, a linear feed must be performed to advance one conveyor pocket length. For a continuous machine, an appropriate motion can be produced via a gear ratio with an adjustable offset to the higher-level master axis.

4.4.1.2 Module description

The **FBLPacTappetChainGear** block is a general module that masters gearing. This makes the block ideal for continuously operating machine variants.

Scope of functions

The **FBLPacTappetChainGear** function block offers the main functions and auxiliary functions as well as possible options shown in the following overview.

Function	Description	Implementation
Main function	Gearing	The gearing can be activated or deactivated via the *executeGearIn* or *executeGearOut* input.
Option	Offset adjustment	In gearing, a relative offset can be set during machine operation. The currently passed offset between the master and slave is reset with every rising edge at *executeChangeOffset*. The set offset is saved retentively and automatically set again at every following synchronization.
Auxiliary functions	Enable axis	The axis is enabled via the level-triggered *enableAxis* input.
	Home axis	The homing procedure of the axis is started via the *executeHoming* input.
	Jog axis	Positive or negative jogging can be started via the level-triggered *jogForward* or *jogBackward* inputs.
	Position axis	An absolute or relative positioning action can be started via the edge-triggered *executePos* input.
	Axis quick stop	A quick stop can be performed or a motion of the axis prevented via the level-triggered *enableEStop* input.
	Change of the axis configuration data	The axis configuration data (modulo length, modulo yes/no) can be changed via the subordinate **FBLPacChangeTOConfigDataMod** function block. The modulo length specified in the ramp-up of the controller is automatically entered and activated by the block in the appropriate configuration data element of the axis. The modulo properties can also be changed at any time in the cyclic program by specifying different values. The user must only ensure that the axis is not enabled. A TO restart must be performed to activate the changed configuration data.

Schematic representation in LAD/FBD

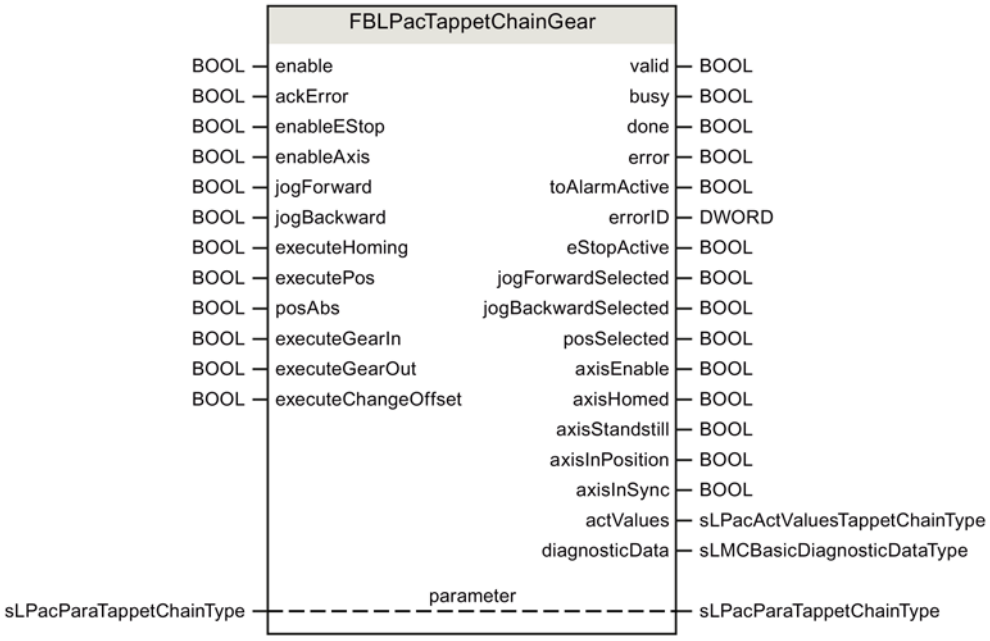


Figure 4-8 FBLPacTappetChain - schematic block view in LAD/FBD

Input/output parameters

Table 4- 49 Input/output signals

Element	P type [1]	Data type	Initial value	Meaning
enable	IN	BOOL	FALSE	Activation of the FB
ackError	IN	BOOL	FALSE	Acknowledge error
enableEStop	IN	BOOL	FALSE	Activate quick stop
enableAxis	IN	BOOL	FALSE	Enable axis
jogForward	IN	BOOL	FALSE	Continuous jogging in the positive direction
jogBackward	IN	BOOL	FALSE	Continuous jogging in the negative direction
executeHoming	IN	BOOL	FALSE	Start homing
executePos	IN	BOOL	FALSE	Start positioning
posAbs	IN	BOOL	FALSE	TRUE: The positioning initiated with executePos is absolute
				FALSE: The positioning initiated with executePos is relative
executeGearIn	IN	BOOL	FALSE	Start synchronization / activate gearing
executeGearOut	IN	BOOL	FALSE	Start desynchronization / deactivate gearing

Element	P type [1]	Data type	Initial value	Meaning
exe-cuteChangeOffset	IN	BOOL	FALSE	Activate start offset shift
valid	OUT	BOOL	FALSE	Block is active
busy	OUT	BOOL	FALSE	Selected functionality being pro-cessed
done	OUT	BOOL	FALSE	Selected functionality completed successfully
error	OUT	BOOL	FALSE	Error occurred
toAlarmActive	OUT	BOOL	FALSE	TO alarm occurred
errorID	OUT	DWORD	16#0000_00 00	Error code
eStopActive	OUT	BOOL	FALSE	Quick stop active
jogForward Selected	OUT	BOOL	FALSE	Jogging positive direction selected
jogBackward-Selected	OUT	BOOL	FALSE	Jogging negative direction selected
posSelected	OUT	BOOL	FALSE	Positioning selected
axisEnabled	OUT	BOOL	FALSE	Axis enabled
axisHomed	OUT	BOOL	FALSE	Axis homed
axisStandStill	OUT	BOOL	FALSE	Axis at standstill
axisInPosition	OUT	BOOL	FALSE	Axis in position
axisInSync	OUT	BOOL	FALSE	Axis synchronous to the master axis
actValues	OUT	sLPacActValues TappetChain-Type	-	Current actual velocity, actual posi-tion and offset
diagnosticData	OUT	sLMCBasicDiag-nostic DataType	-	For the diagnostics structure, see Alarm and error messages (Page 333)
parameter	IN_OUT	sLPacParaTap-pet ChainType	-	Parameter for supply chain

1) Parameter types: IN = input parameter, OUT = output parameter, IN_OUT = in-out parameter

Table 4- 50 sLPacActValuesTappetChainType structure

Element	Data type	Initial value	Meaning
r32ActualOffset	REAL	0.0	Current offset
r32ActualPostion	REAL	0.0	Current position
r32ActualVelocity	REAL	0.0	Current velocity

Table 4- 51 sLPacParaTappetChainType structure

Element	Data type	Meaning
sTOs	sLPacToCamDataType	TOs for supply chain
sMotionData	sLPacTappetChainMotionDataType	Parameters for EStop, homing, jog, pos, gearing
sConfigData	sLPacTappetChainConfigDataType	Configuration data for supply chain

Table 4- 52 sLPacToCamDataType structure

Element	Data type	Meaning
toAxis	FollowingAxis	Axis TO supply chain (following axis)
toLeadingValue	MasterType	TO master axis (at least position axis or external encoder)
toCam	CamType	Not relevant

Table 4- 53 sLPacTappetChainMotionDataType structure

Element	Data type	Meaning
sEStopData	sLMCBasicEStopDataType	Parameter for quick stop
sHomingData	sLMCBasicHomingDataType	Parameter for homing
sJogData	sLPacJogDataType	Parameter for jogging
sPosData	sLPacPosDataType	Parameter for positioning
sGearingData	sLPacGearingDataType	Parameter for gearing; not relevant for FBLPacTappetChainCam
sCammingData	sLPacCammingDataType	Parameter for camming; not relevant for FBLPacTappetChainGear
sPhasingData	sLMCBasicPhasingDataType	Parameter for offset adjustment

Quick stop parameters - sEStopData structure

See Axis quick stop (Page 43)

Homing parameters - sHomingData structure

See Home axis (Page 35)

Jog parameters - sJogData structure

See Jog axis (Page 36)

Positioning parameters - sPosData structure

See Position axis (Page 37)

Gearing parameters - sGearingData structure

See Synchronize/desynchronize axis in gearing (Page 38)

Offset adjustment parameters - sPhasingData structure

See Offset adjustment (Page 43)

Table 4- 54 sLPacTappetChainConfigDataType structure

Element	Data type	Meaning
sAxisMechanicalData	sLPacAxisMechanicalDataType	Mechanical data

Table 4- 55 sLPacAxisMechanicalDataType structure

Element	Data type	Initial value	Meaning
r32ModuloLength	REAL	0.0	Modulo length (conveyor pocket length)
boModuloActive	BOOL	TRUE	Activate modulo length

4.4.1.3 Integration of the module

Technology objects, basic parameterization and block call

Table 4- 56 TO type and basic parameterizations

Axis / technology object	
TO type	Real synchronous axis
Version	Linear axis
Modulo	Conveyor pocket length
Block call	BackgroundTask

Further blocks and libraries

The **FBLPacTappetChainGear** block uses the following blocks internally.

Block/library	Function
FBLMCBasicAxis / LMCBasic	Axis function block
FBLPacChangeTOConfigDataMod / LPac	Function block to change the configuration data of the TO axis

Some of the internally used blocks originate from other libraries. See Section Additional required libraries (Page 15).

Error messages

Error number [HEX]	Meaning
16#0000	LPAC_ERR_NO; error-free
16#8000	LPAC_ERR_INVALID_STEP; internal error, please get in touch with your Siemens contact partner
16#8001	LPAC_ERR_NO_AXIS; no axis TO specified
16#8003	LPAC_ERR_TO_ALARM_ACTIVE; TO alarm active; error message from axis function block Error description, see *SIMOTION Axis Function Block* Application Manual and Section Alarm and error messages (Page 333) in this manual
16#8004	LPAC_ERR_AXIS_NOT_DISABLED; axis not disabled for change of the modulo length
16#8005	LPAC_ERR_MODULO_LENGTH; invalid modulo length passed (only FBLPacChangeTOConfigDataMod block)
16#8008	LPAC_ERR_POS_VELOCITY_ZERO; positioning velocity = 0.0
16#8009	LPAC_ERR_POS_ACC_DEC_ZERO; acceleration and/or deceleration for positioning = 0.0
16#800B	LPAC_ERR_POS_ABS_POSITION_VALUE; target position for absolute positioning < 0.0 or > modulo length
16#800C	LPAC_ERR_ PHASING_DYNAMIC_DATA; offset adjustment for gearing; invalid velocity and/or acceleration parameters
16#8010	LPAC_ERR_NO_MASTER_AXIS; no axis TO specified for master axis
16#8011	LPAC_ERR_AXIS_FUNCTION_BLOC; error message from axis function block For details, see *diagnosticData.b32ErrorCode* output structure Error description, see *SIMOTION Axis Function Block* Application Manual
16#8013	LPAC_ERR_NO_FOLLOWING_OBJECT; no following object specified or leading axis is not a following axis
16#8016	LPAC_ERR_GEAR_RATIO; gear ratio numerator or denominator = 0
16#8078	LPAC_ERR_SET_GEAR_OR_CAMMING_OFFSET; error message for the offset adjustment from the setGearingsOffset or setCammingOffset command
16#80C3	LPAC_ERR_EXECUTE_DURING_RESET; Execute requested while error acknowledgment active

Error number [HEX]	Meaning
16#8100	LPAC_ERR_READ_SYSTEM_VARIABLE; error when reading the system variables / variable cannot be read
16#8101	LPAC_ERR_MAX_TIME_FOR_TO_RESTART; TO restart not performed within the monitoring time

Note

The **pTappetChainGear** program unit is available in the LPac project to illustrate the use of the **FBLPacTappetChainGear** block.

This unit contains the programs for the required block calls in the StartupTask and the BackgroundTask.

4.4.2 FBLPacTappetChainCam - supply chain camming

4.4.2.1 Technological consideration

The **FBLPacTappetChainCam** block controls the "supply chain" (pocket-type conveyor, tappet chain) machine module for horizontal tubular bagging machines. It can be used together with other blocks of the **LPac** library for implementing a SIMOTION complete application of a horizontal tubular bagging machine and also independently for similar mechanically-oriented machine units that require an electronic camming for a leading axis.

This block differs from the **FBLPacTappetChainGear** block only in the type of synchronous coupling.

The LPac library does not provide any cam calculation blocks for this block because the deployed cams can be created easily with the CamEdit and CamTool tools available in SIMOTION SCOUT; a change at runtime is not normally required.

4.4.2.2 Module description

The **FBLPacTappetChainCam** block provides a general module that masters camming. This makes the block ideal for continuously operating machine variants.

Scope of functions

The **FBLPacTappetChainCam** function block offers the main functions and auxiliary functions as well as possible options shown in the following overview.

Function	Description	Implementation
Main function	Camming	The gearing can be activated or deactivated via the *executeCamIn* or *executeCamOut* edge-triggered input.
Option	Offset adjustment	In camming, a relative offset can be set during machine operation. The currently passed offset between the master and slave is reset with every rising edge at *executeChangeOffset*. The set offset is saved retentively and automatically set again at every following synchronization.
Auxiliary functions	Enable axis	The axis is enabled via the level-triggered *enableAxis* input.
	Home axis	The homing procedure of the axis is started via the *executeHoming* input.
	Jog axis	Positive or negative jogging can be started via the level-triggered *jogForward* or *jogBackward* inputs.
	Position axis	An absolute or relative positioning action can be started via the edge-triggered *executePos* input.
	Axis quick stop	A quick stop can be performed or a motion of the axis prevented via the level-triggered *enableEStop* input.
	Change of the axis configuration data	The axis configuration data (modulo length, modulo yes/no) can be changed via the subordinate **FBLPacChangeTOConfigDataMod** function block. The modulo length specified in the ramp-up of the controller is automatically entered and activated by the block in the appropriate configuration data element of the axis. The modulo properties can also be changed at any time in the cyclic program by specifying different values. The user must only ensure that the axis is not enabled. A TO restart must be performed to activate the changed configuration data.

Schematic representation in LAD/FBD

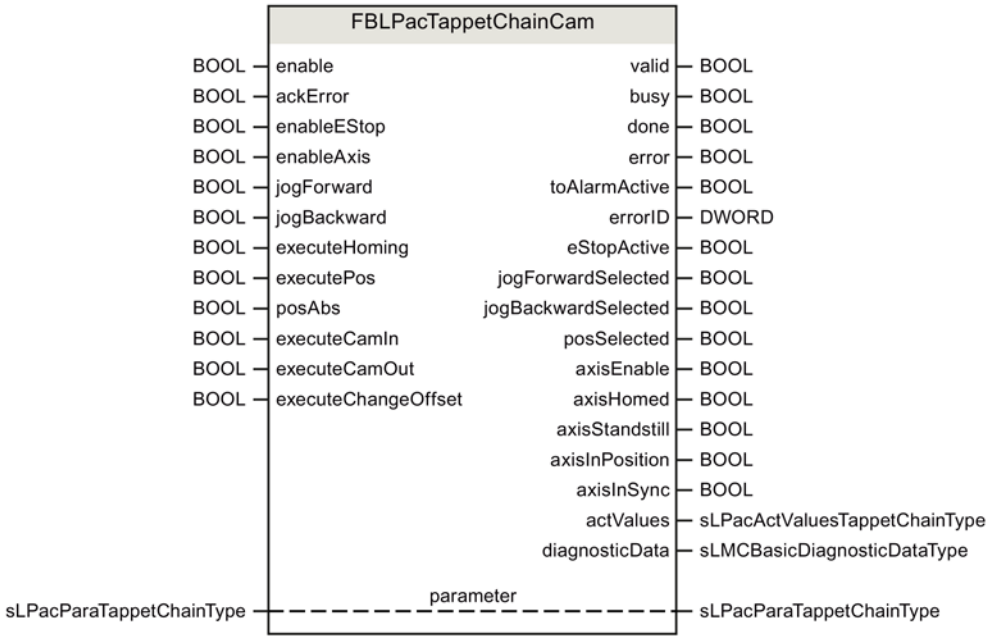


Figure 4-9 FBLPacTappetChainCam - schematic block view in LAD/FBD

Input/output parameters

Table 4- 57 Input/output signals

Element	P type 1)	Data type	Initial value	Meaning
enable	IN	BOOL	FALSE	Activation of the FB
ackError	IN	BOOL	FALSE	Acknowledge error
enableEStop	IN	BOOL	FALSE	Activate quick stop
enableAxis	IN	BOOL	FALSE	Enable axis
jogForward	IN	BOOL	FALSE	Continuous jogging in the positive direction
jogBackward	IN	BOOL	FALSE	Continuous jogging in the negative direction
executeHoming	IN	BOOL	FALSE	Start homing
executePos	IN	BOOL	FALSE	Start positioning
posAbs	IN	BOOL	FALSE	TRUE: The positioning initiated with executePos is absolute
				FALSE: The positioning initiated with executePos is relative
executeCamIn	IN	BOOL	FALSE	Start synchronization / activate camming
executeCamOut	IN	BOOL	FALSE	Start desynchronization / deactivate camming

Element	P type 1)	Data type	Initial value	Meaning
exe-cuteChangeOffset	IN	BOOL	FALSE	Activate start offset shift
valid	OUT	BOOL	FALSE	Block is active
busy	OUT	BOOL	FALSE	Selected functionality being processed
done	OUT	BOOL	FALSE	Selected functionality completed success-fully
error	OUT	BOOL	FALSE	Error occurred
toAlarmActive	OUT	BOOL	FALSE	TO alarm occurred
errorID	OUT	DWORD	16#0000 _0000	Error code
eStopActive	OUT	BOOL	FALSE	Quick stop active
jogForward Selected	OUT	BOOL	FALSE	Jogging positive direction selected
jogBackward-Selected	OUT	BOOL	FALSE	Jogging negative direction selected
posSelected	OUT	BOOL	FALSE	Positioning selected
axisEnabled	OUT	BOOL	FALSE	Axis enabled
axisHomed	OUT	BOOL	FALSE	Axis homed
axisStandStill	OUT	BOOL	FALSE	Axis at standstill
axisInPosition	OUT	BOOL	FALSE	Axis in position
axisInSync	OUT	BOOL	FALSE	Axis synchronous to the master axis
actValues	OUT	sLPacAc-tValues Tap-petChain-Type	-	Current actual velocity, actual position and offset
diagnosticData	OUT	sLM-CBasic Diagnos-ticDataTy pe	-	Diagnostic structure, see Section Alarm and error messages (Page 333)
parameter	IN_OUT	sLPac-ParaTap-pet Chain-Type	-	Parameter for supply chain

1) Parameter types: IN = input parameter, OUT = output parameter, IN_OUT = in-out parameter

Table 4- 58 sLPacActValuesTappetChainType structure

Element	Data type	Initial value	Meaning
r32ActualOffset	REAL	0.0	Current offset
r32ActualPostion	REAL	0.0	Current position
r32ActualVelocity	REAL	0.0	Current velocity

Table 4- 59 sLPacParaTappetChainType structure

Element	Data type	Meaning
sTOs	sLPacTOCamDataType	TO for supply chain
sMotionData	sLPacTappetChainMotionDataType	Parameters for EStop, homing, jog, pos, camming
sConfigData	sLPacTappetChainConfigDataType	Configuration data for supply chain

Table 4- 60 sLPacTOCamDataType structure

Element	Data type	Meaning
toAxis	FollowingAxis	Axis TO supply chain (following axis)
toLeadingValue	MasterType	TO master axis (at least position axis or external encoder)
toCam	CamType	TO cam

Table 4- 61 sLPacTappetChainMotionDataType structure

Element	Data type	Meaning
sEStopData	sLMCBasicEStopDataType	Parameter for quick stop
sHomingData	sLMCBasicHomingDataType	Parameter for homing
sJogData	sLPacJogDataType	Parameter for jogging
sPosData	sLPacPosDataType	Parameter for positioning
sGearingData	sLPacGearingDataType	Parameter for gearing; not relevant for FBL-PacTappetChainCam
sCammingData	sLPacCammingDataType	Parameter for camming
sPhasingData	sLMCBasicPhasingDataType	Parameter for offset adjustment

Quick stop parameters - sEStopData structure

See Axis quick stop (Page 38)

Homing parameters - sHomingData structure

See Home axis (Page 35)

Jog parameters - sJogData structure

See Jog axis (Page 36)

Positioning parameters - sPosData structure

See Position axis (Page 37)

Camming parameters - sCammingData structure

See Synchronizing/desynchronizing axis in camming (Page 40)

Offset adjustment parameters - sPhasingData structure

See Offset adjustment (Page 43)

Table 4- 62 sLPacTappetChainConfigDataType structure

Element	Data type	Meaning
sAxisMechanicalData	sLPacAxisMechanicalDataType	Mechanical data

Table 4- 63 sLPacAxisMechanicalDataType structure

Element	Data type	Initial value	Meaning
r32ModuloLength	REAL	0.0	Modulo length (conveyor pocket length)
boModuloActive	BOOL	TRUE	Activate modulo length

4.4.2.3 Integration of the module

Technology objects, basic parameterization and block call

Table 4- 64 TO type and basic parameterizations

Axis / technology object	
TO type	Real synchronous axis
Version	Linear axis
Modulo	Conveyor pocket length in mm (inches)
Block call	BackgroundTask

Further blocks and libraries

The **FBLPacTappetChainCam** block uses the following blocks internally.

Block/library	Function
FBLMCBasicAxis / LMCBasic	Axis function block
FBLPacChangeTOConfig-DataMod / LPac	Function block to change the configuration data of the TO axis

Some of the internally used blocks originate from other libraries. For information, see Section Additional required libraries (Page 15).

Error messages

Error number [HEX]	Meaning
16#0000	LPAC_ERR_NO; error-free
16#8000	LPAC_ERR_INVALID_STEP; internal error, please get in touch with your Siemens contact partner
16#8001	LPAC_ERR_NO_AXIS; no axis TO specified
16#8003	LPAC_ERR_TO_ALARM_ACTIVE; TO alarm active; error message from axis function block Error description, see *SIMOTION Axis Function Block* Application Manual and Section Alarm and error messages in this manual
16#8004	LPAC_ERR_AXIS_NOT_DISABLED; axis not disabled for change of the modulo length
16#8005	LPAC_ERR_MODULO_LENGTH; invalid modulo length
16#8008	LPAC_ERR_POS_VELOCITY_ZERO; positioning velocity = 0.0
16#8009	LPAC_ERR_POS_ACC_DEC_ZERO; acceleration and/or deceleration for positioning = 0.0
16#800B	LPAC_ERR_POS_ABS_POSITION_VALUE; target position for absolute positioning < 0.0 or > modulo length
16#800C	LPAC_ERR_ PHASING_DYNAMIC_DATA; offset adjustment for gearing; invalid velocity and/or acceleration parameters
16#8010	LPAC_ERR_NO_MASTER_AXIS; no axis TO specified for master axis
16#8011	LPAC_ERR_AXIS_FUNCTION_BLOC; error message from axis function block For details, see *diagnosticData.b32ErrorCode* output structure Error description, see *SIMOTION Axis Function Block* Application Manual
16#8012	LPAC_ERR_NO_CAM; no cam passed
16#8013	LPAC_ERR_NO_FOLLOWING_OBJECT; no following object specified or leading axis is not a following axis
16#8078	LPAC_ERR_SET_GEAR_OR_CAMMING_OFFSET; error message for the offset adjustment from the setGearingsOffset or setCammingOffset command
16#80C3	LPAC_ERR_EXECUTE_DURING_RESET; Execute requested while error acknowledgment active

Error number [HEX]	Meaning
16#8100	LPAC_ERR_READ_SYSTEM_VARIABLE; error when reading the system variables / variable cannot be read
16#8101	LPAC_ERR_MAX_TIME_FOR_TO_RESTART; TO restart not performed within the monitoring time

Note

The **pTappetChainCam** program unit is available in the LPac project to illustrate the use of the **FBLPacTappetChainCam** block.

This unit contains the programs for the required block calls in the StartupTask and the BackgroundTask.

4.4.3 FBLPacDosingPos - dosing positioning axis

4.4.3.1 Technological consideration

The **FBLPacDosingPos** block serves as control for dosing modules operated in positioning mode (e.g. cup or worm dosing unit) or also by other machine modules with similar function requirements. It can be used together with other blocks of the **LPac** library for implementing a SIMOTION complete application of a vertical tubular bagging machine and also independently for similarly designed machines that require an axis operated in positioning mode.

4.4.3.2 Module description

Scope of functions

The **FBLPacDosingPos** function block offers the main functions and auxiliary functions as well as possible options shown in the following overview.

Function	Description	Implementation
Main function	Positioning	An absolute or relative positioning action can be started via the edge-triggered *executePos* input.
Options	Absolute positioning by modulo length	In addition to absolute or relative positioning, absolute positioning by the set modulo length of the axis to the next modulo position is possible by selecting the *posAbsModulo* input signal. In this way, for example, it is possible for a cup dosing unit with n dosing cups, to simply control one dosing cycle when the modulo length of, e.g. 360 degrees, is selected so that it corresponds to one cup sector.
Auxiliary functions	Enable axis	The axis is activated via the level-triggered *enableAxis* input.
	Home axis	The homing procedure of the axis is started via the *executeHoming* input.
	Jog axis	Positive or negative jogging can be started via the level-triggered *jogForward* or *jogBackward* inputs.
	Axis quick stop	A quick stop can be performed or a motion of the axis prevented via the level-triggered *enableEStop* input.
	Change of the axis configuration data	The axis configuration data (modulo length, modulo yes/no) can be changed via the subordinate **FBLPacChangeTOConfigDataMod** function block. The modulo length specified in the ramp-up of the controller is automatically entered and activated by the block in the appropriate configuration data element of the axis. The modulo properties can also be changed at any time in the cyclic program by specifying different values. The user must only ensure that the axis is not enabled. A TO restart must be performed to activate the changed configuration data.

Schematic representation in LAD/FBD

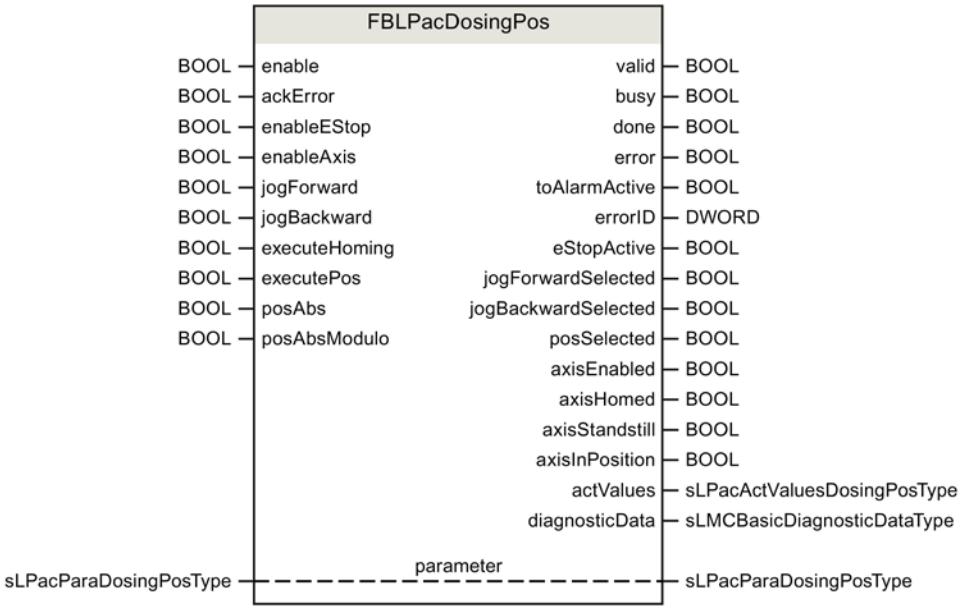


Figure 4-10 FBLPacDosingPos - schematic block view in LAD/FBD

Input and output parameters

Table 4- 65 Input and output parameters

Parameter	P type[1)]	Data type	Initial value	Description
enable	IN	BOOL	FALSE	Activation of the FB
ackError	IN	BOOL	FALSE	Acknowledge error
enableEStop	IN	BOOL	BOOL	Activate quick stop
enableAxis	IN	BOOL	FALSE	Enable axis
jogForward	IN	BOOL	FALSE	Continuous jogging in the positive direction
jogBackward	IN	BOOL	FALSE	Continuous jogging in the negative direction
executeHoming	IN	BOOL	FALSE	Start homing
executePos	IN	BOOL	FALSE	Start positioning
posAbs	IN	BOOL	FALSE	TRUE: The positioning initiated with executePos is absolute
				FALSE: The positioning initiated with executePos is relative
posAbsModulo	IN	BOOL	FALSE	Absolute positioning by modulo length

Parameter	P type[1)]	Data type	Initial value	Description
valid	OUT	BOOL	FALSE	Block has been selected and can be used
busy	OUT	BOOL	FALSE	Selected functionality being processed
done	OUT	BOOL	FALSE	Selected functionality completed successfully
error	OUT	BOOL	FALSE	Error occurred
toAlarmActive	OUT	BOOL	FALSE	TO alarm occurred
errorID	OUT	DWORD	16#0000_0000	Error code
eStopActive	OUT	BOOL	FALSE	Quick stop active
jogForward-Selected	OUT	BOOL	FALSE	Jogging positive direction selected
jogBackward-Selected	OUT	BOOL	FALSE	Jogging negative direction selected
posSelected	OUT	BOOL	FALSE	Positioning selected
axisEnabled	OUT	BOOL	FALSE	Axis enabled
axisHomed	OUT	BOOL	FALSE	Axis homed
axisStandStill	OUT	BOOL	FALSE	Axis at standstill
axisInPosition	OUT	BOOL	FALSE	Axis in position
actValues	OUT	sLPacActValuesMaster DosingPosType	-	Actual values
diagnosticData	OUT	sLMCBasicDiagnos-ticDataType	-	Diagnostic structure, see Section Alarm and error messages (Page 333)
parameter	IN_OUT	sLPacParaDosingPosType	-	Parameter for dosing

1) Parameter types: IN = input parameter, OUT = output parameter, IN_OUT = in-out parameter

Table 4- 66 sLPacParaDosingPosType structure

Element	Data type	Meaning
sTOs	sLPacDosingPosTOsType	TO for master axis
sMotionData	sLPacDosingPosMotionType	Parameters for jog, pos, Estop, homing
sConfigData	sLPacParaDosingPosConfigDataType	Configuration data

Table 4- 67 sLPacDosingPosTOsType structure

Element	Data type	Meaning
toAxis	posAxis	Axis TO (position axis)

Table 4- 68 sLPacDosingPosMotionType structure

Element	Data type	Meaning
sEStopData	sLMCBasicEStopDataType	Parameter for quick stop
sHomingData	sLMCBasicHomingDataType	Parameter for homing
sJogData	sLPacJogDataType	Parameter for jogging
sPosData	sLPacPosDataType	Parameter for positioning

Quick stop parameters - sEStopData structure

See Axis quick stop (Page 43)

Homing parameters - sHomingData structure

See Home axis (Page 35)

Jog parameters - sJogData structure

See Jog axis (Page 36)

Positioning parameters - sPosData structure

See Position axis (Page 37)

Table 4- 69 sLPacParaDosingPosConfigDataType structure

Element	Data type	Meaning
sAxisMechanicalData	sLPacAxisMechanicalDataType	Mechanical data

Table 4- 70 sLPacAxisMechanicalDataType structure

Element	Data type	Initial value	Meaning
r32ModuloLength	REAL	0.0	Modulo length
boModuloActive	BOOL	FALSE	Activate modulo length

4.4.3.3 Integration of the module

Technology objects, basic parameterization and block call

Table 4- 71 TO type and basic parameterizations

Axis / technology object	
TO type	Position axis
Version	Rotary axis
Modulo	Preferred, 360°
Block call	BackgroundTask

Further blocks and libraries

The **FBLPacDosingPos** block uses the following blocks internally.

Block/library	**Function**
FBLMCBasicAxis / LMCBasic	Axis function block
FBLPacChangeTOConfigDataMod / LPac	Function block to change the configuration data of the TO axis

Some of the internally used blocks originate from other libraries. See Section Additional required libraries (Page 15).

Error messages

Error number [HEX]	**Meaning**
16#0000	LPAC_ERR_NO; error-free
16#8000	LPAC_ERR_INVALID_STEP; internal error, please get in touch with your Siemens contact partner
16#8001	LPAC_ERR_NO_AXIS; no axis TO specified
16#8003	LPAC_ERR_TO_ALARM_ACTIVE; TO alarm active; error message from axis function block Error description, see *SIMOTION Axis Function Block* Application Manual and Section Alarm and error messages (Page 333) in this manual
16#8004	LPAC_ERR_AXIS_NOT_DISABLED; axis not disabled for change of the modulo length
16#8005	LPAC_ERR_MODULO_LENGTH; invalid modulo length
16#8008	LPAC_ERR_POS_VELOCITY_ZERO; positioning velocity = 0
16#8009	LPAC_ERR_POS_ACC_DEC_ZERO; acceleration/deceleration = 0
16#8011	LPAC_ERR_AXIS_FUNCTION_BLOC; error message from axis function block For details, see *diagnosticData.b32ErrorCode* output structure Error description, see *SIMOTION Axis Function Block* Application Manual
16#8014	LPAC_ERR_AXIS_NOT_HOMED; axis not homed

Error number [HEX]	Meaning
16#80C3	LPAC_ERR_EXECUTE_DURING_RESET; Execute requested while error acknowledgment active; source FBLPacChangeTOConfigDataMod
16#8100	LPAC_ERR_READ_SYSTEM_VARIABLE; error when reading the system variables / variable cannot be read; source FBLPacChangeTOConfigDataMod
16#8101	LPAC_ERR_MAX_TIME_FOR_TO_RESTART; TO restart not performed within the monitoring time; source FBLPacChangeTOConfigDataMod

Note

The **pDosingPos** program unit is available in the LPac project to illustrate the use of the **FBLPacDosingPos** block.

This unit contains the programs for the required block calls in the StartupTask and the BackgroundTask.

4.4.4 FBLPacDosingControl - dosing weigher discharge control

4.4.4.1 Technological consideration

The **FBLPacDosingControl** block coordinates the product discharge with the actual machine control of a continuous tubular bagging machine.

All clocked product feeds, such as multi-head weighers, cup or worm dosing units in positioning operation can be controlled and coordinated.

The block can be used together with other blocks of the **LPac** library for implementing a SIMOTION complete application of a vertical continuous tubular bagging machine.

The coordination of the individual axes with the discharge control is shown below.

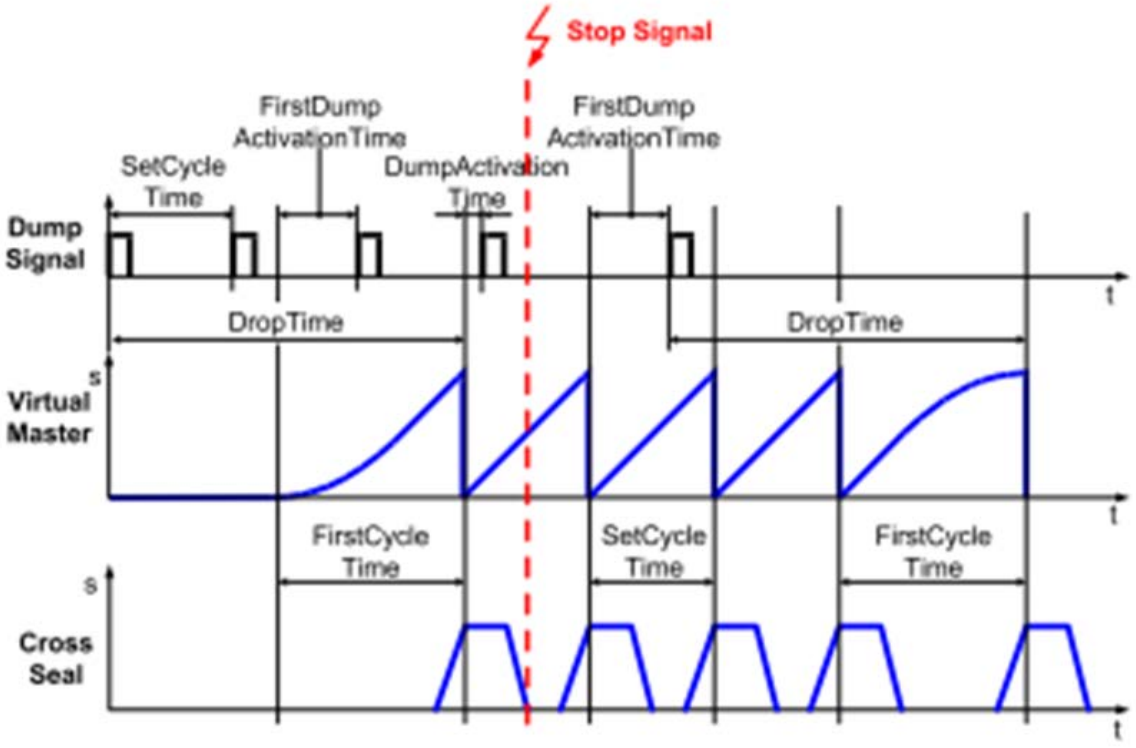


Figure 4-11 Coordination of a virtual master, cross-sealer and dosing unit - continuous machine

The coordination between a dosing weigher or cup or worm dosing unit in positioning mode and a machine (virtual master) is performed via a digital interface in which a handshake is implemented via the exchange of the "weigher ready" and "discharge signal" signals as shown in the following drawing.

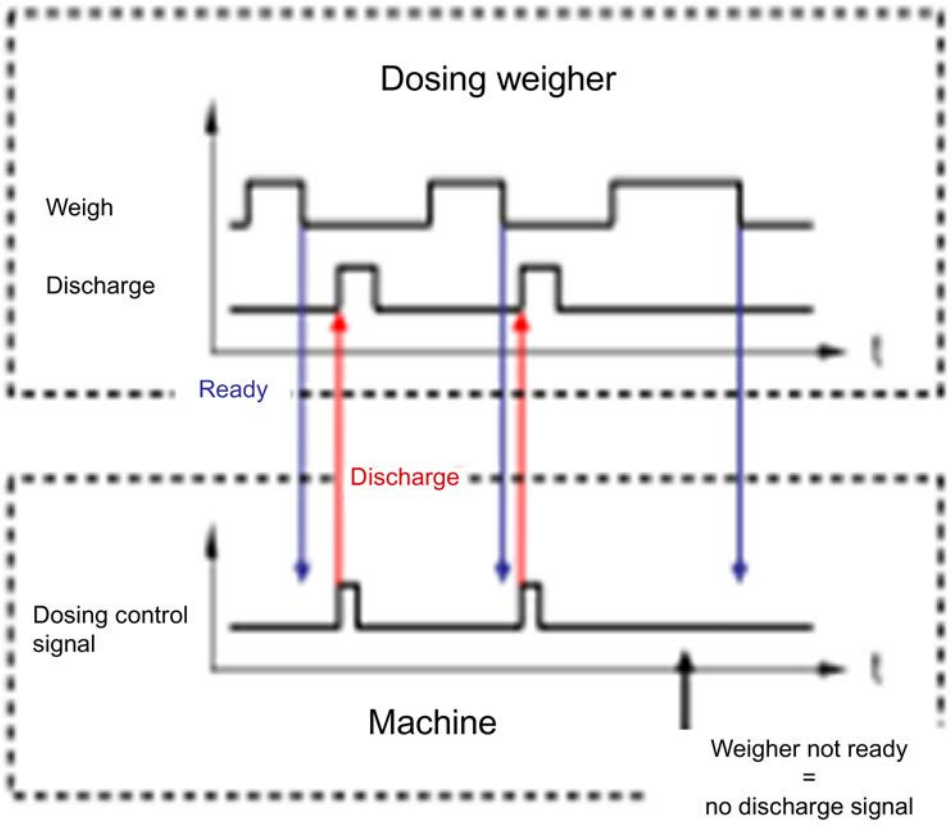


Figure 4-12 Handshake for the dosing weigher

Requirements

The following requirements must be met:

- The drop time must be ≥ 10 ms
- The production cycle time must always be > 0 ms
- The closing time of the cross-sealer must be > 0 ms (only when the "FirstBagFilling" option is activated)
- The bag length must be > 0 mm
- The acceleration must be > 0 mm/s^2 and must be sufficiently high so that the constant machine velocity is attained within the bag length!
- The wait time for the double discharge must be > 0 ms (only when the "DoubleDump" option is activated)
- No jerk limitation may be set on the master axis.

Discharge coordination

Depending on the set discharge time and the production speed (cycle time), the three cases described below are differentiated block-internally:

Case 1: Drop time ≤ cycle time

- The discharge signal is always initiated in the current production cycle

DropTime ≤ SetCycleTime

- Data for the dosing unit:
  FirstDumpActivationTime = FirstCycleTime - DropTime
  DumpActivationTime = SetCycleTime - DropTime

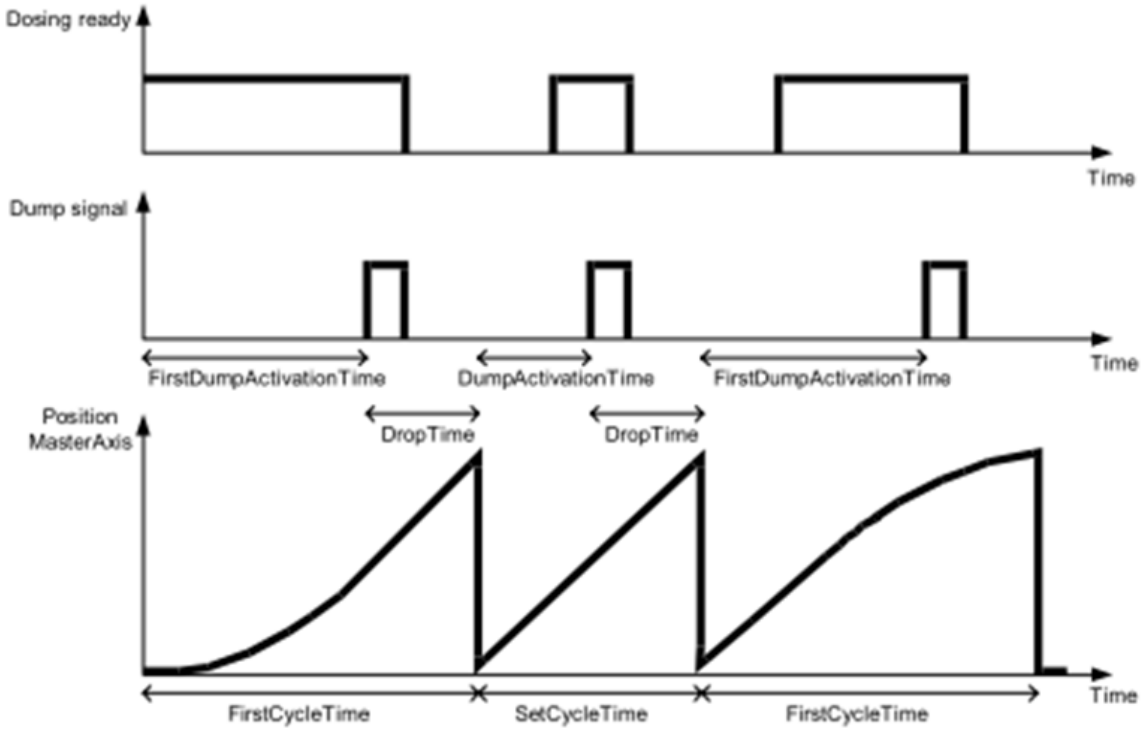


Figure 4-13 Timing of the material discharge for Case 1: Drop time ≤ cycle time

Case 2: Drop time > cycle time and drop time ≤ first/last cycle time

- 1st discharge signal is initiated before or in the first production cycle
- 2nd discharge signal is initiated in the first production cycle

(DropTime > SetCycleTime) AND (DropTime ≤ FirstCycleTime)

- Data for the dosing unit:
  FirstDumpActivationTime = FirstCycleTime - DropTime
  DumpActivationTime = (2 * SetCycleTime - FirstCycleTime) + FirstDumpActivationTime

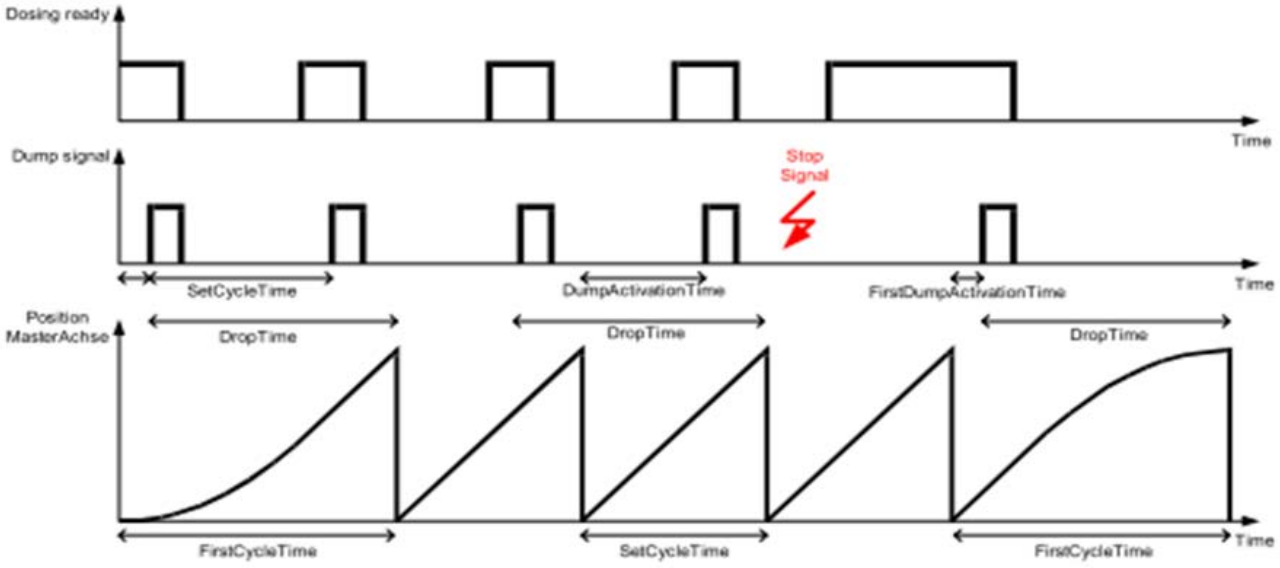


Figure 4-14 Timing of the material discharge for Case 2: Drop time > cycle time AND first or last cycle time

Case 3: Drop time > first/last cycle time

- Discharge signal is always initiated before the production cycle; multiple products are present concurrently in the discharge pipe

DropTime > FirstCycleTime

- Data for the dosing unit (example for the figure):
  DumpActivationTime = 3 * SetCycleTime - DropTime
  FirstDumpActivationTime = FirstCycleTime - SetCycleTime + DumpActivationTime

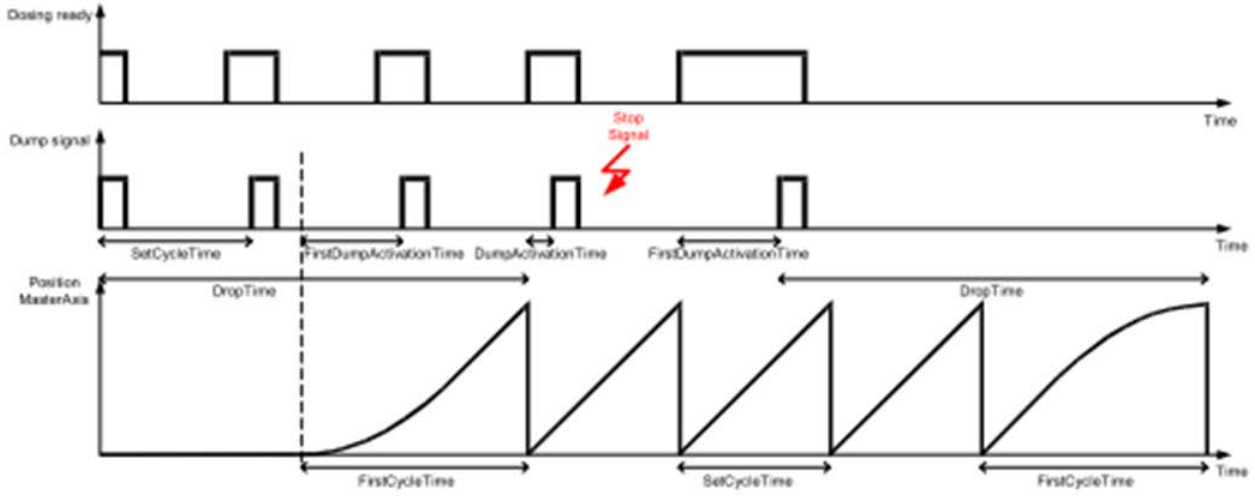


Figure 4-15 Timing of the material discharge for Case 3: Drop time > first or last cycle time

4.4.4.2 Module description

The **FBLPacDosingControl** function block provides the main and auxiliary functions as well as options shown in the following overview.

Function	Description	Implementation
Main function	Dosing unit / weigher control	A digital interface between the machine control and the weigher control coordinates a multi-head weigher or a cup or worm dosing unit in positioning mode with the master axis of a vertical continuous tubular bagging machine. The involved units are controlled by Boolean signals. During the running production cycle, the machine velocity and the drop time can be changed (within certain limits). The discharge control has the same possible operating modes as the master axis block, "continued" = endless bag production until the production stop initiated by the operator, "counting" = production of specified number of products passed to the block. When the specified number is reached, the block stops the production automatically. "singledrop" is performed only in a single production cycle.
		The block outputs the following values as counter values: • *productCounter* (number of bags produced) • *droppedProducts* (discharged products) • *packagedProducts* (discharged products sealed in the bag) • *weigherNotReady* (number of discharge attempts during which the weigher / dosing unit was not ready) The *packagedProducts*, *droppedProducts* and *weigherNotReady* counters are reset automatically at the start of production. The *productCounter* counter can be reset by setting the *resetProductCounter* signal.
Option	Direct machine master control	In addition to controlling the machine master (virtual axis) FB_LPacMasterCont via Boolean signals, the block can control the master axis directly to achieve improved time accuracy via SIMOTION technology commands (move, pos). For this purpose, the axis TO of the master axis must be passed to the block at the *parameter.sTOs.toMasterAxis* parameter input. To permit the block to control the master axis, it must be enabled and homed, and be present in the production initial setting. The axis is not stopped by the module in the event of an emergency stop (EStop). This functionality must be performed by the master block. The *masterControlActive* output specifies whether the master is controlled directly by the block.

Function	Description	Implementation
Option	FirstBagFilling	When this option is set, the block controls the cross-sealer automatically via the *executeSealing* output signal. Thus, the cross-sealer must be able to perform a complete sealing cycle automatically (close, seal, open). The block requires as feedback the "cross-sealer open" state and the *r32CloseTimeSealer* [ms] parameter; this is the time that the cross-sealer requires from the opened position until reaching the closed position.
		With activation of the "FirstBagFilling" option by setting the *parameter.sConfigData.boFirstBagFilling* signal, a product is initially discharged and a sealing cycle initiated for the cross-sealer so that the discharged product meets the closed jaw of the cross-sealer considering the drop time and the closing time of the cross-sealer. After opening the cross-sealer, the "normal" production cycle is started. At production end, the last product discharge is no longer initiated so that the second to last bag can be sealed and separated at the top, and so the base of the last bag is sealed.
		Without activation of this option, the master axis is always started from its initial position (normally position 0), and the product discharged in accordance with the set drop time and cycle time. This causes at production stop the base of the last bag to be sealed and a product discharged again, namely a product remains in the bag that is still open at the top.
Option	StopDosingNotReady	With activation of the option by setting the *parameter.sConfigData.boStopDosingNotReady* signal, the production will be stopped if the weigher is not ready at the discharge time.
		The *parameter.sConfigData.boAutomaticRestart* signal can still be used to select whether the production should be restarted automatically after the stop when the weigher reports "ready" again.
		Without activation of this option, to achieve a "steady" operation of the machine, it is not stopped when the dosing unit / weigher is not ready at the discharge time. This prevents the machine from constantly being stopped and started when the weigher is not currently ready at the discharge time. Consequently, empty bags will be produced if the weigher is not ready at the discharge time.
Option	Doubledump	To discharge a larger product quantity, the block can be informed by transferring a delay time at the *parameter.sConfigData.r32DoubleDumpDelayTime* [ms] parameter that a second discharge should be initiated after the set wait time after the first discharge signal. The "weigher ready" signal is ignored in this case.

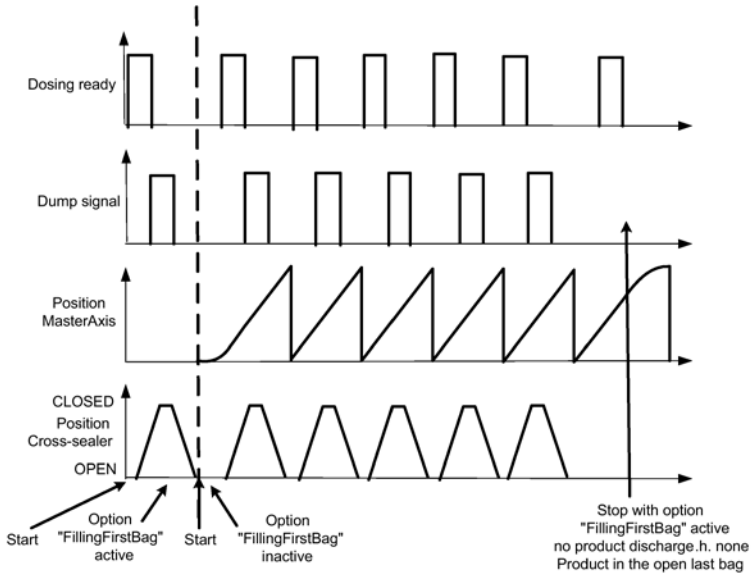


Figure 4-16 Start and stop with "FillingFirstBag" option

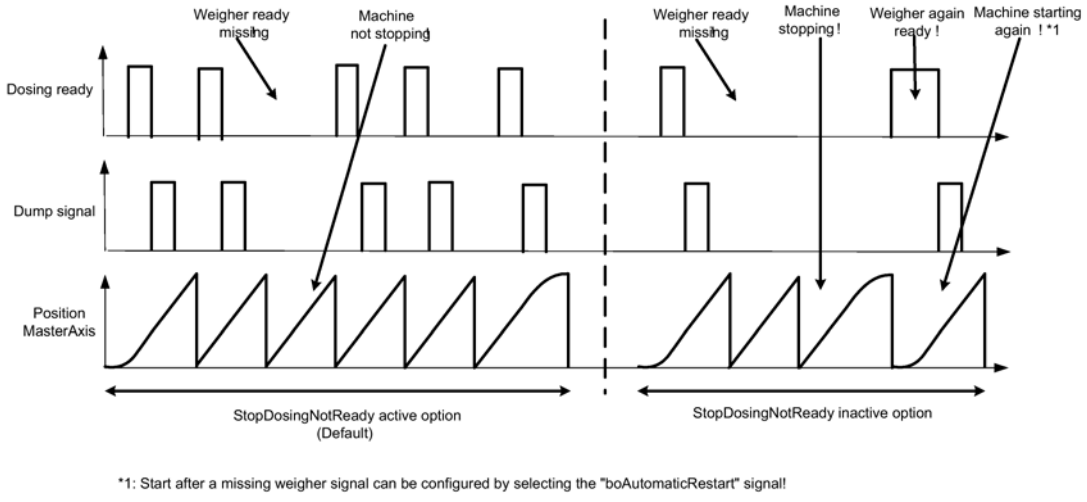
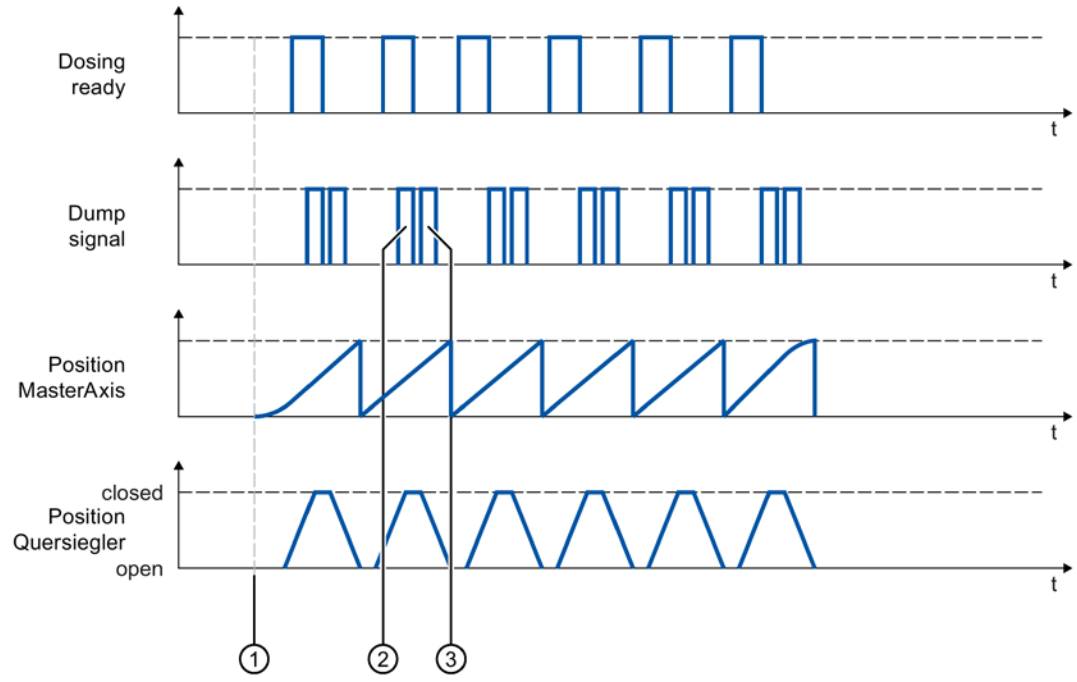


Figure 4-17 Representation of the "StopDosingNotReady" and "AutomaticRestart" options

The following figure shows the "DoubleDump" option.



① Start
② Discharge signal 1
③ Discharge signal 2

Schematic representation in LAD/FBD

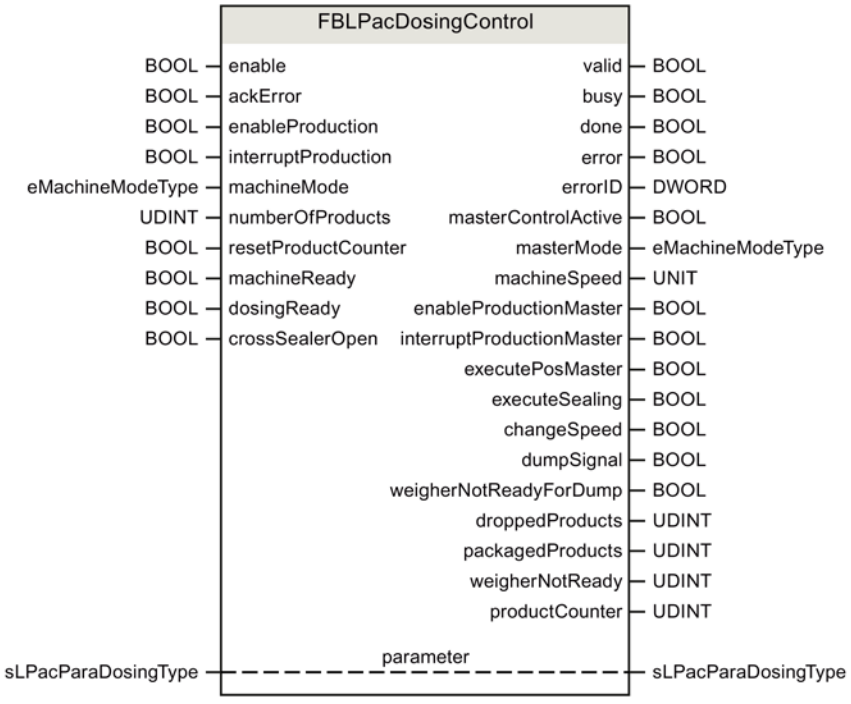


Figure 4-18 FBLPacDosingControl - schematic block view in LAD/FBD

Input and output parameters

Table 4- 72 Input and output parameters

Parameter	P type [1)]	Data type	Initial value	Description
enable	IN	BOOL	FALSE	Activation of the FB
ackError	IN	BOOL	FALSE	Acknowledge error
enableProduction	IN	BOOL	FALSE	Start production cycle
Interrupt Production	IN	BOOL	FALSE	Interrupt production, i.e. stop at the specified stop position without desynchronizing the following axis
machineMode	IN	eMachine Mode-Type	Continued	Specification of the operating mode; see also FBLPacMasterCont; the following modes can be activated:
				SingleDrop: If the *enableProduction* start signal is set, just one product is discharged and one bag is produced;
				Continued: If the *enableProduction* start signal is set, products are discharged and bags produced until the start signal is reset again.

Parameter	P type [1]	Data type	Initial value	Description
				Counting: If the *enableProduction* start signal is set, products are discharged and bags produced until the number of bags as set at the "numberOfProducts" input have been produced. The minimum number is two bags.
numberOf Products	IN	UDINT	0	Target number of products
resetProduct Counter	IN	BOOL	FALSE	Setting the signal resets the product counter to 0.
machineReady	IN	BOOL	FALSE	Following axes (machine) ready for production
dosingReady	IN	BOOL	FALSE	Dosing unit ready for discharge
crossSealerOpen	IN	BOOL	FALSE	Feedback from the cross-sealer; the cross-sealer is at the open position. Production can be started. This signal is required only when the "FirstBagFilling" option is used.
valid	OUT	BOOL	FALSE	Block has been selected and can be used
busy	OUT	BOOL	FALSE	Selected functionality being processed
done	OUT	BOOL	FALSE	Selected functionality completed successfully
error	OUT	BOOL	FALSE	Error occurred
errorID	OUT	DWORD	16#0000_0000	Error code
masterControl Active	OUT	BOOL	FALSE	The signal indicates that the master axis TO is controlled directly by the block specified in the *parameter.sTOs.toMasterAxis* parameter. General prerequisite is that the master axis was enabled previously, is homed and present in the production initial state. The commands _move to start the master or to change the production velocity or _pos for the positioned stop of the master axis at the production initial state are used for controlling.
masterMode	OUT	eMachine Mode-Type	Continued	Output signal for forwarding and controlling the machine master. The FBLPacDosingControl block assumes the operating mode control. Consequently, the master always receives the "continued" operating mode.
machineSpeed	OUT	UINT	0	Production speed in bags/min
enableProduction master	OUT	BOOL	FALSE	Start signal for the master for "block-external" control, i.e. via the Boolean output signals of the FBLPacDosingControl block
interruptProductionMaster	OUT	BOOL	FALSE	Interrupt signal for the master for "block-external" control, i.e. via the Boolean output signals of the FBLPacDosingControl block

Parameter	P type [1]	Data type	Initial value	Description
executePosMaster	OUT	BOOL	FALSE	Start signal for the master for starting a positioning cycle for one cycle (singleDrop) (bag length) for "block-external" control, i.e. via the Boolean output signals of the FBL-PacDosingControl block. This requires that the master axis has been enabled and homed previously, and is present in the production initial state.
executeSealing	OUT	BOOL	FALSE	Control signal for the cross-sealer for starting a sealing cycle (close, seal, open); required only for use of the "fillingFirstBag" option
changeSpeed	OUT	BOOL	FALSE	Change speed; notification for the master
dumpSignal	OUT	BOOL	FALSE	Discharge signal for the dosing unit / weigher
dosingNotReady ForDump	OUT	BOOL	FALSE	Dosing unit / weigher not ready at discharge time
droppedProducts	OUT	UDINT	0	Discharged products
packedProducts	OUT	UDINT	0	Packed products
weigherNotReady	OUT	UDINT	0	Number of discharges for which the weigher / dosing unit was not ready at the discharge time
productCounter	OUT	UDINT	0	Product counter; can be reset to 0 with the *resetProductCounter* signal as long as it is not in active production operation
parameter	IN_OUT	sLPacPara Dosing type	-	Parameter for dosing and options

1) Parameter types: IN = input parameter, OUT = output parameter, IN_OUT = in-out parameter

Table 4- 73 sLPacParaDosingType structure

Element	Data type	Meaning
sTOs	sLPacToPosOcDataType	TOs for dosing
sConfigData	sLPacDosingDataType	Configuration data for dosing

Table 4- 74 sLPacToPosOcDataType structure

Element	Data type	Initial value	Meaning
toAxis	_axis_ref	TO#NIL	Axis TO of the machine master
toOutputCam	outputCamType	TO#NIL	TO cam controller for dosing control in relation to the master axis (output cam type: time-based cam = TYPE_TIME)

Table 4- 75 sLPacDosingDataType structure

Element	Data type	Initial value	Meaning
r32BagLength	REAL	0.0	Bag length in mm (inches)
u16MachineSpeed	UINT	0	Machine velocity in bags/minute
r32AccelerationMaster	REAL	0.0	Master axis acceleration/deceleration; the acceleration/deceleration must be selected sufficiently large so that the specified machine velocity can be attained within the specified bag length, otherwise the master axis does not start.
r32BasePosMaster	REAL	0.0	Master axis start position = production operation initial position (normally master position = 0.0)
r32DropTime	REAL	0.0	Drop time [ms]
r32DoubleDumpDelayTime	REAL	0.0	Double discharge; delay time in ms between the first and second discharge
r32DosingAndSealPosition	REAL	0.0	Reference position of the master axis for discharge control; switch-on position for the output cams required for the discharge coordination
r32SwitchOnDuration	REAL	100.0	Switch-on duration [ms] for the output cams required for the discharge coordination
r32CloseTimeSealer	REAL	0.0	Time [ms] that the cross-sealer requires for the closing motion; required only for the "FirstBagFilling" option
boDoubleDump	BOOL	FALSE	Select "Double discharge" option
boFirstBagFilling	BOOL	FALSE	Select "Fill first bag" option
boPointOfNoReturn	BOOL	FALSE	Currently not used
boStopDosingNotReady	BOOL	FALSE	Select "Stop machine if the weigher is not ready" option
boAutomaticRestart	BOOL	FALSE	"Automatic start if stopped beforehand because the weigher / dosing unit was not ready" option

4.4.4.3 Integration of the module

Technology objects and parameterization

Table 4- 76 TO type and basic parameterizations

Axis / technology object	
TO type master axis	TO axis of the master axis (virtual axis) if this was passed to the block ("Direct machine master control" option)
Cam controller	TO cam controller that coordinates the discharge control in relation to the master axis (activated and parameterized block-internally); output cam type = time-based cam (TYPE_TIME)
Version	Linear axis

Axis / technology object	
Modulo	The LPac project works with 1000
Block call	BackgroundTask or IPOSynchronousTask

Note that some output signals, such as the discharge signal, are present only over one call cycle.

Further blocks and libraries

The **FBLPacDosingControl** block uses the following blocks internally.

Block/library	Function
FBLPacGetPosTime / LPac	Function block for calculating the positioning time in "singleDrop" operating mode

No further blocks from other libraries are called.

Error messages

Error number [HEX]	Meaning
16#0000	ERR_ERR_NO; error-free
16#8000	LPAC_ERR_INVALID_STEP; internal error, please get in touch with your Siemens contact partner
16#8006	LPAC_ERR_BAGLENGTH_ZERO; bag length ≤ 0.0
16#800A	LPAC_ERR_ACCELERATION_ZERO; acceleration ≤ 0.0
16#8002	LPAC_ERR_NO_OUTPUT_CAM; no cam controller specified
16#8020	LPAC_ERR_MACHINE_SPEED_ZERO; production speed ≤ 0.0
16#80A0	LPAC_ERR_NEW_DROPTIME_TO_LONG; drop time too long (only after drop time changes during running production operation)
16#80A1	LPAC_ERR_NEW_DROPTIME_TO_SHORT; drop time ≤ 10.0 ms
16#80A2	LPAC_ERR_TO_MANY_CYCLES; velocity change too large, too many transition cycles required
16#80A3	LPAC_ERR_ENABLE_OUTPUT_CAM; error during the enable of the cam controller -> see diagnosticData
16#80A4	LPAC_ERR_MOVE; error when issuing the _move command to start the master -> see diagnosticData
16#80A5	LPAC_ERR_POS; error when issuing the _pos command to start the master in singleDrop -> see diagnosticData
16#80A6	LPAC_ERR_CS_CLOSETIME_ZERO; cross-sealer closing time ≤ 0.0
16#80A7	LPAC_ERR_ACCELERATION_TOO_SMALL; acceleration too small to accelerate to the specified machine velocity within the bag length
16#80A9	LPAC_ERR_DROPTIME_ZERO; drop time ≤ 0.0 ms
16#80AA	LPAC_ERR_DOUBLE_DUMP_DELAY_TIME_ZERO; delay time for double discharge ≤ 0.0
16#8150	LPAC_ERR_DISTANCE_LESS_THAN_OR_EQUAL_ZERO; parameter *distance* ≤ 0; source FBLPacGetPosTime

Error number [HEX]	Meaning
16#8151	LPAC_ERR_VELOCITY_ZERO_OR_NEGATIVE; parameter *velocity* ≤ 0; source FBLPacGetPosTime
16#8152	LPAC_ERR_ACCELERATION_ZERO_OR_NEGATIVE; parameter *acceleration* ≤ 0; source FBLPacGetPosTime
16#8153	LPAC_ERR_DECELERATION_ZERO_OR_NEGATIVE; parameter *deceleration* ≤ 0; source FBLPacGetPosTime
16#8154	LPAC_ERR_INVALID_PARAMETERS; invalid parameters require division by 0; source FBLPacGetPosTime

Note

The **pDosingControl** program unit is available in the LPac project to illustrate the use of the **FBLPacDosingControl** block.

This unit contains the programs for the required block calls in the StartupTask and the BackgroundTask.

4.4.5 FBLPacNPNBDetection - product detection shift register

4.4.5.1 Technological consideration

The **FBLPacNPNBDetection** block monitors a product flow, e.g. on a pocket-type conveyor using a sensor (e.g. light scanner), and controls the "No Product No Bag" function on horizontal tubular bagging machines.

It can be used together with other blocks of the **LPac** library for implementing a SIMOTION complete application of a horizontal tubular bagging machine and also independently for similar applications that use a simple shift register with binary product information (FALSE = no product available, TRUE = product available).

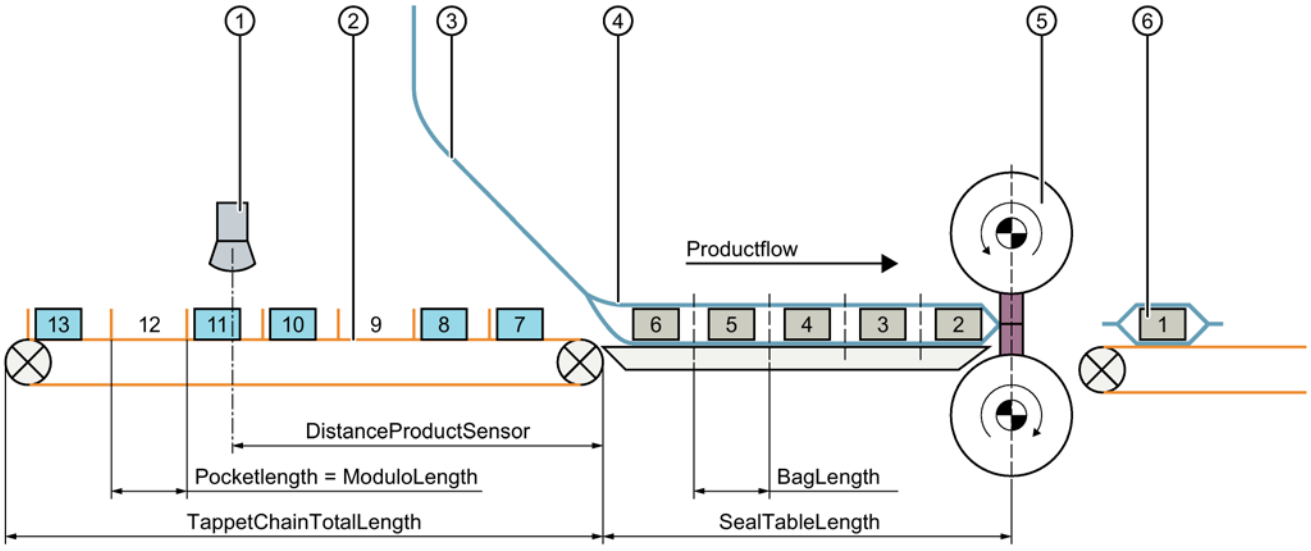


Figure 4-19 Product infeed

4.4.5.2 Module description

Scope of functions

The **FBLPacNPNBDetection** function block provides the main and auxiliary functions, and possibly options, shown in the following overview.

Function	Description	Implementation
Main function	Shift register	The block represents a simple shift register with parameterizable "register depth". The product information is stored in binary format. Using the specified *u16ProductSensorDistance* parameter and the pocket length of the pocket-type conveyor, the block automatically calculates the "register depth" internally (number of machine cycles required to transport a product from the product sensor up to the start of the foil tube).
Options	Position window	A position range must be specified on the block in which the sensor monitoring is active. In this way, for example, unwanted sensor trips through mechanical fingers/slides or similar, can be hidden.
Auxiliary functions	Run machine until empty	When the *emptyingMaschine* input is activated, the following parameters can be used to calculate how many machine cycles are required to completely empty the pocket-type conveyor: *u16TappetChainTotalLength*, *u16TappetChainPocketLength*, *u16SealTableLength* and the *productLength*. The machine cycles are counted and after the number of cycles has expired, the *machineEmpty* signal and the *machineEmptyPulse* are output. After evaluation of these signals, the machine leading axis can be stopped at the exact product.

Description of the product monitoring / shift register (No Product No Bag control)

From the following two parameters, the block calculates the number of product cycles that still have to be executed after the detection of a product event (Product available or No product available), before the NoProductNoBag axis (and therefore the coupled foil feed and cross-sealer axes) must be desynchronized from the machine leading axis and stopped in order that no empty bags are produced when the product is missing:

- *u16ProductSensorDistance* (distance from the product sensor to the end of the pocket-type conveyor)
- *u16TappetChainPocketLength* (pocket length of a pocket of the pocket-type conveyer; this generally corresponds to the modulo length of the axis)

The procedure is the same as for a change of the product event from No product available to Product available. However, the NPNB axis is synchronized to the machine leading axis again with appropriate control. The block inputs *productDetected*, *productDetectedPulse*, *noProductDetected*, *noProductDetectedPulse* and *NPNBEnabled* are used to control the coupling operations. Make sure that these output signals are not output until the calculated number of product cycles have been executed. The procedure is similar for the deactivation of the monitoring and the signals are only reset after the calculated number of product cycles have been executed.

Description of the emptying function

From the specified parameters *u16TappetChainTotalLength*, *u16TappetChainPocketLength*, *u16SealTableLength* and *productLength*, the block calculates the number of product cycles that must be executed after activation of the emptying function with the *emptyingMachine* input signal so that the pocket-type conveyer and the sealing area before the cross-sealer is completely emptied. Prerequisite for this is that the product flow on the pocket-type conveyer is interrupted (e.g. through a stop of the infeed or a mechanical stopper). As soon as the appropriate number of machine cycles has been executed, the *machineEmpty* output signal and *machineEmptyPulse* are output which are used to stop the machine leading axis. During the emptying run, the activated No Product No Bag functionality is processed normally to ensure that no empty bag is produced. An activated emptying run is indicated by the *emptyingActive* output signal.

Schematic representation in LAD/FBD

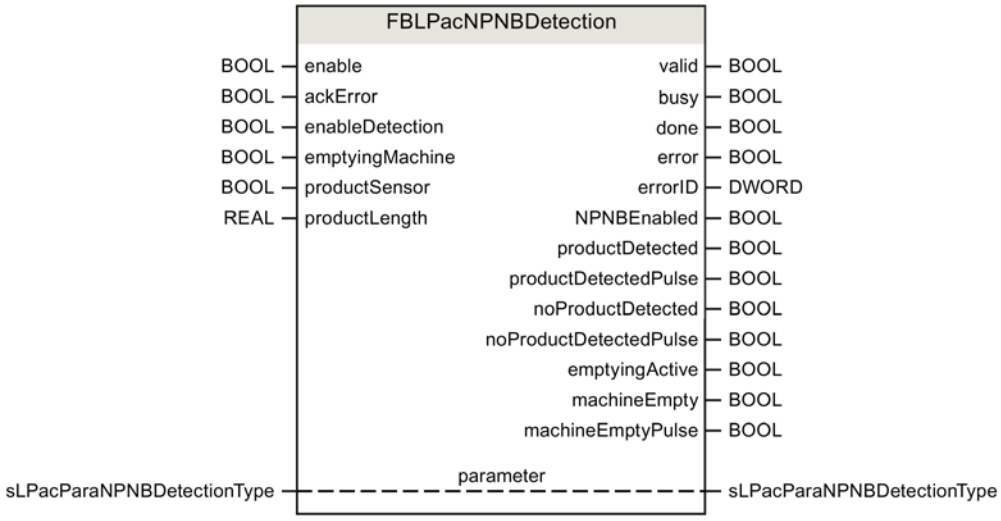


Figure 4-20 FBLPacNPNBDetection - schematic block view in LAD/FBD

Input and output parameters

Table 4- 77 Input and output parameters

Parameter	P type[1)]	Data type	Initial value	Meaning
enable	IN	BOOL	FALSE	Activation of the FB
ackError	IN	BOOL	FALSE	Acknowledge error
enableDetection	IN	BOOL	FALSE	Activate product detection
emptyingMa-chine	IN	BOOL	FALSE	Select "Run machine until empty"; only synchronize "good" events
productSensor	IN	BOOL	FALSE	Product sensor
productLength	IN	REAL	0.0	Product length (bag length)
valid	OUT	BOOL	FALSE	Block activated
busy	OUT	BOOL	FALSE	Selected functionality being processed
done	OUT	BOOL	FALSE	Selected functionality com-pleted successfully
error	OUT	BOOL	FALSE	Error occurred
errorID	OUT	DWORD	16#0000_0000	Error code
NPNBEnabled	OUT	BOOL	FALSE	No Product No Bag detection enabled
productDetected	OUT	BOOL	FALSE	Product detected
productDetect-edPulse	OUT	BOOL	FALSE	Product detected pulse

Parameter	P type[1)]	Data type	Initial value	Meaning
noProductDetected	OUT	BOOL	FALSE	No product detected
noProductDetectedPulse	OUT	BOOL	FALSE	No product detected pulse
emptyingActive	OUT	BOOL	FALSE	Emptying run active
machineEmpty	OUT	BOOL	FALSE	Pocket-type conveyer (machine) has been emptied
machineEmptyPulse	OUT	BOOL	FALSE	Pocket-type conveyer (machine) has been emptied pulse
parameter	IN_OUT	sLPacParaNPNBDetetionType	-	Parameter for No Product No Bag detection

1) Parameter types: IN = input parameter, OUT = output parameter, IN_OUT = in-out parameter

Table 4- 78 sLPacParaNPNBDetetionType structure

Element	Data type	Meaning
sTOs	sLPacToFolDataType	Axis TO of the supply chain
sConfigData	sLPacParaNPNBDetectionConfigDataType	Configuration data for NPNB detection

Table 4- 79 sLPacParaNPNBDetectionConfigDataType structure

Element	Data type	Initial value	Meaning
r32DetectionWindowStart	REAL	0.0	Start of the position window for product detection (in relation to the position of the pocket-type conveyer)
r32DetectionWindowEnd	REAL	0.0	End of the position window for product detection (in relation to the position of the pocket-type conveyer)
u16SealTableLength	UINT	0	Distance between the end of the pocket-type conveyer and the cross-sealer (area of the foil pretensioning or longitudinal sealing in which the foil tube is formed)
u16TappetChainTotalLength	UINT	0	Total length of the pocket-type conveyer
u16TappetChainPocketLength	UINT	0	Pocket length of the pocket-type conveyer (generally corresponds to the modulo length of the axis)
u16ProductSensorDistance	UINT	0	Distance between the product sensor and the start of the foil tube (SealTable)

4.4.5.3 Embedding the module

Technology objects and parameterization

Table 4- 80 TO type and basic parameterizations

Axis / technology object	
TO type	Following axis
Version	Linear axis
Modulo	Pocket length
Block call	BackgroundTask

Further blocks and libraries

The **FBLPacNPNBDetection** block does not use any further blocks internally.

Error messages

Error number [HEX]	**Meaning**
16#0000	LPAC_ERR_NO; error-free
16#8000	LPAC_ERR_INVALID_STEP; internal error, please get in touch with your Siemens contact partner
16#8001	LPAC_ERR_NO_AXIS; no axis TO specified
16#8005	LPAC_ERR_MODULO_LENGTH; invalid modulo length (≤ 0.0)
16#8006	LPAC_ERR_BAGLENGTH_ZERO; product length ≤ 0
16#8240	LPAC_ERR_NO_DETECTION_WINDOW; width of the position window for product detection = 0.0

Note

The **pNPNBGear** program unit is available in the LPac project to illustrate the use of the **FBLPacNPNBDetection** block and the No Product No Bag functionality.

This unit contains the programs for the required block calls in the StartupTask and the BackgroundTask.

4.5 Foil transport

4.5.1 FBLPacFoilFeedPos - foil extractor positioning axis

4.5.1.1 Technological consideration

The **FBLPacFoilFeedPos** block controls the "foil feed" machine module on intermittently operating tubular bagging machines. The foil feed operates in positioning mode with a possible control/coordination by the **FBLPacMasterInt** machine master block. The block also provides the optional print mark correction and slip compensation correction possibilities.

The module can be used together with other blocks of the **LPac** library for implementing a SIMOTION complete application of an intermittently operating tubular bagging machine and also independently for similar mechanically-oriented machine units that require positioning functionality with compensation capabilities.

4.5.1.2 Module description

Scope of functions

The **FBLPacFoilFeedPos** function block offers the main functions and auxiliary functions as well as possible options shown in the following overview.

Function	Description	Implementation
Main function	Positioning functionality	Positioning by a specified position can be started via the edge-triggered *executePos and executePosFoilFeed* inputs.
Options	Print mark correction	The foil position can be corrected via a positioning motion superimposed on the basic motion with the aid of a print mark sensor.
	Slip compensation	A measured slip between the foil feed axis and the foil itself can be corrected via a positioning motion superimposed on the basic motion.
Auxiliary functions	Enable axis	The axis is activated via the level-triggered *enableAxis* input.
	Home axis	The homing procedure of the axis is started via the *executeHoming* input.
	Jog axis	Positive or negative jogging can be started via the level-triggered *jogForward* or *jogBackward* inputs.
	Axis quick stop	A quick stop can be performed or a motion of the axis prevented via the level-triggered *enableEStop* input.

Function	Description	Implementation
	Change of the axis configuration data	The axis configuration data (modulo length, modulo yes/no) can be changed via the subordinate **FBLPacChangeTOConfigDataMod** function block. The modulo length specified in the ramp-up of the controller is automatically entered and activated by the block in the appropriate configuration data element of the axis. The modulo properties can also be changed at any time in the cyclic program by specifying different values. The user must only ensure that the axis is not enabled. A TO restart must be performed to activate the changed configuration data.

Function details

Positioning function:

- A differentiation is made whether or not the print mark correction is activated. The positioning is started with a rising edge at the *executePos input.*
- At a rising edge on the *executePosFoilFeed* input, the actual axis position is first set to 0.0 and then an absolute positioning performed to the specified bag length.
- Without print mark correction, the *posAbs* input signal preselects whether absolute (**posAbs** = TRUE) or relative positioning is to be performed.
- With print mark correction, positioning is always performed absolute. Once the target position has been reached, the actual axis position is reset to 0.0.

Homing:

- Without print mark correction

  In this case, the axis is not traversed for homing.

  The setpoint position and actual position are set automatically to 0.0 for each new axis enable.

- With print mark correction

  In this case, active traversing is made for homing.

  Position 0.0 is set automatically for each new axis enable.

  The position of the axis is initially set to 0.0 via *_redefinePosition* and the measuring input activated without range limits.

  Traversal is then made with active measuring input by the position passed in *posData*.

  The first detected print mark initiates the measuring input and the determined value is increased by *r32RemainingWayFromPrintmark*.

  This determined position value is then approached "absolutely".

  After reaching this position, the axis is reset to start value 0.0 via *redefinePostion*.

  This is equivalent to homing to the position of the measuring input plus an offset by the traversing distance to the sealing point.

  If no print mark is detected during this travel, a homing error is signalled.

Schematic representation in LAD/FBD

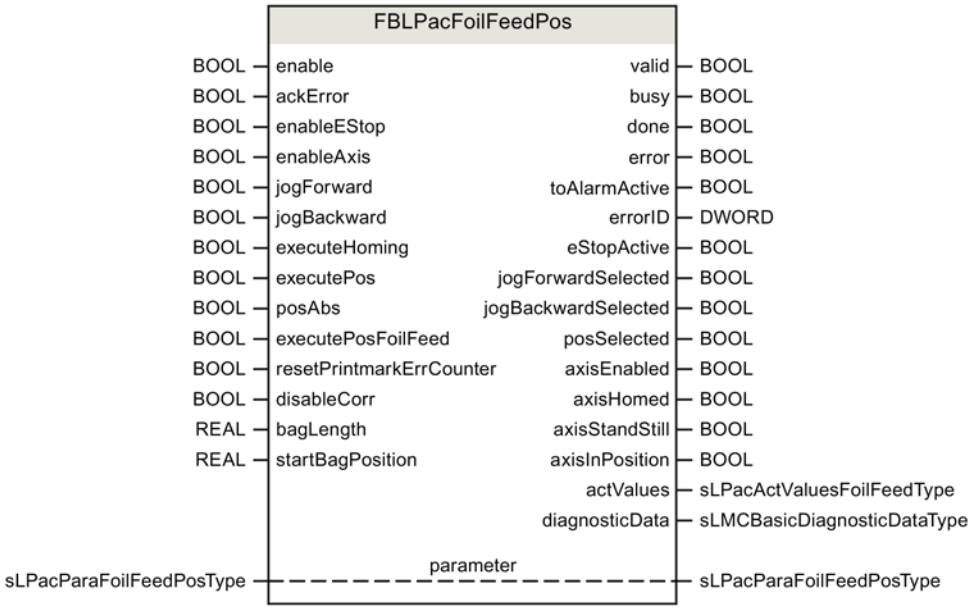


Figure 4-21 Schematic block view in LAD/FBD

Input and output parameters

Table 4- 81 Input and output parameters

Parameter	Data type	P type [1]	Initial value	Meaning
enable	BOOL	IN	FALSE	Activation of the FB
ackError	BOOL	IN	FALSE	Acknowledge error
enableEStop	BOOL	IN	FALSE	Activate quick stop
enableAxis	BOOL	IN	FALSE	Enable axis
jogForward	BOOL	IN	FALSE	Continuous jogging in the positive direction
jogBackward	BOOL	IN	FALSE	Continuous jogging in the negative direction
executeHoming	BOOL	IN	FALSE	Start: Homing
executePos	BOOL	IN	FALSE	Start: Positioning
posAbs	BOOL	IN	FALSE	TRUE: The positioning initiated with *executePos* is absolute

Parameter	Data type	P type [1]	Initial value	Meaning
executePosFoilFeed	BOOL	IN	FALSE	Start positioning sequence by the length specified at the *baglength* input. In contrast to positioning initiated by *executePos*, the current actual position and position setpoint are set to 0.0 before the start of positioning and then positioned absolutely to the parameterized bag length. During positioning, either the print mark correction or the slip compensation can be started.
resetPrintmarkErrCounter	BOOL	IN	FALSE	Reset print mark error counter
disableCorr	BOOL	IN	FALSE	Suppress correction motion through print mark correction or slip compensation
bagLength	REAL	IN	0.0	Product length
valid	BOOL	OUT	FALSE	Block is active
busy	BOOL	OUT	FALSE	Selected functionality being processed
done	BOOL	OUT	FALSE	Selected functionality completed successfully
error	BOOL	OUT	FALSE	Error occurred
toAlarmActive	BOOL	OUT	FALSE	TO alarm occurred
errorID	DWORD	OUT	16#0000_0000	Error code
eStopActive	BOOL	OUT	FALSE	Quick stop active
jogForwardSelected	BOOL	OUT	FALSE	Jogging positive direction selected
jogBackwardSelected	BOOL	OUT	FALSE	Jogging negative direction selected
posSelected	BOOL	OUT	FALSE	Positioning selected
axisEnabled	BOOL	OUT	FALSE	Axis enabled
axisHomed	BOOL	OUT	FALSE	Axis homed
axisStandStill	BOOL	OUT	FALSE	Axis at standstill
axisInPosition	BOOL	OUT	FALSE	Axis in position
actValues	sLPacActValuesFoilFeedType	OUT	-	Actual values of the foil feed
DiagnosticData	sLMCBasicDiagnosticDataType	OUT	-	For the diagnostics structure, see Alarm and error messages (Page 43)
parameter	sLPacParaFoilFeedPosType	IN_OUT	-	Parameter for foil feed

1) Parameter types: IN = input parameter, OUT = output parameter, IN_OUT = in-out parameter

Table 4- 82 sLPacActValuesFoilFeedType structure

Element	Data type	Initial value	Meaning
r32ActualVelocity	REAL	0.0	Actual velocity of the foil feed [mm/s]
r32ActualPosition	REAL	0.0	Current position of the foil feed
r32PrintMarkDifference	REAL	0.0	Measured offset of the print mark with activated print mark correction [mm]
u32PrintMarkErrorCounter	UDINT	0	Current number of print mark errors for which no print mark was detected
u32PrintMarkCorrectionCounter	UDINT	0	Current number of executed correction motions
r32SlipDifference	REAL	0.0	Measured slip with activated slip compensation [mm]

Table 4- 83 sLPacParaFoilFeedPosType structure

Element	Data type	Meaning
toAxis	sLPacToPosDataType	Axis TO for foil feed positioning axis
sMotionData	sLPacFoilFeedIntMotionDataType	Parameters for jog, pos, Estop, homing
sConfigData	sLPacFoilFeedConfigDataType	Configuration data for foil feed

Table 4- 84 sLPacToPosDataType structure

Element	Data type	Initial value	Meaning
toAxis	PosAxis	TO#NIL	Position axis

Table 4- 85 sLPacFoilFeedIntMotionDataType structure

Element	Data type	Meaning
sEStopData	sLMCBasicEStopDataType	Parameter for quick stop
sHomingData	sLMCBasicHomingDataType	Parameter for homing
sJogData	sLPacJogDataType	Parameter for jogging
sPosData	sLPacPosDataType	Parameter for positioning

Quick stop parameters - sEStopData structure

See Axis quick stop

Homing parameters - sHomingData structure

See Home axis (Page 43)

Jog parameters - sJogData structure

See Jog axis (Page 36)

Positioning parameters - sPosData structure

See Position axis (Page 37)

Table 4- 86 sLPacFoilFeedConfigDataType structure

Element	Data type	Initial value	Meaning
r32Position	REAL	0.0	Position
r32maxVelocity	REAL	0.0	Max. velocity
r32maxAcceleration	REAL	0.0	Max. acceleration
r32maxDeceleration	REAL	0.0	Max. deceleration
sPrintMarkCorr	sLPacPrintMark CorrType	-	Parameter for print mark correction
sParaSlipComp	sLPacSlipComp-Type	-	Parameter for slip compensation
sInterfacePosData	sLPacFFPosMax DataType	-	Positioning data interface to the FBLPac-MasterInt block
sAxisMechanicalData	sLPacAxis Mechanical-DataType	-	Axis data

Table 4- 87 sLPacPrintMarkCorrType structure for print mark correction option

Element	Data type	Initial value	Meaning
boPrintMarkCorr	BOOL	FALSE	Print mark correction activation
toMeasuringInput	MeasuringIn-put Type	TO#NIL	Measuring input object
r32PrintMarkPos	REAL	0.0	Position setpoint of the print mark [mm]
r32RemainingWay FromPrintMark	REAL	0.0	Distance between print mark and sealing point that should also be traversed
r32MinPrintMarkCorr	REAL	0.0	Minimum print mark deviation above which a correction motion is performed
r32MaxPrintMarkCorr	REAL	0.0	Maximum permissible correction
u32MaxValueForPrint MarkErrors	UDINT	1	Maximum number of permitted errors for the print mark detection
r32MeasRangePosSta rt	REAL	0.0	Start of the measuring range
r32MeasRangePosEn d	REAL	0.0	End of the measuring range

Element	Data type	Initial value	Meaning
r32CalcFact	REAL	1.0	Factor by which the measured path difference is to compensated
r32Velocity	REAL	0.0	Correction velocity
r32Acceleration	REAL	0.0	Correction acceleration and deceleration
boEdge	BOOL	TRUE	TRUE: Positive edge of the print mark is evaluated
			FALSE: Negative edge of the print mark is evaluated
boCorrInvert	BOOL	FALSE	TRUE: Inversion of the measured deviation of the print mark position

Table 4- 88 sLPacSlipCompType structure for the slip compensation option

Element	Data type	Initial value	Meaning
boSlipComp	BOOL	FALSE	Slip compensation activation
toExternalEncoder	External Encoder Type	TO#NIL	TO for external encoder
r32Velocity	REAL	0.0	Correction velocity
r32Acceleration	REAL	0.0	Correction acceleration
r32MaxSlip	REAL	0.0	Slip as of which a compensating motion is executed

Table 4- 89 sLPacFFPosMaxDataType structure

Element	Data type	Initial value	Meaning
sMaxPosData	sLPacPosMax DynDataType	-	Positioning data interface to the FBL-PacMasterInt block

Table 4- 90 sLPacPosMaxDynDataType structure

Element	Data type	Initial value	Meaning
r32Position	REAL	-1.0	Position
r32MaxVelocity	REAL	-1.0	Maximum velocity 1)
r32MaxAcceleration	REAL	-1.0	Maximum acceleration 1)
r32MaxDeceleration	REAL	-1.0	Maximum deceleration 1)
r64VelocityOverride	LREAL	-1.0	Velocity override
r64AccelerationOverride	LREAL	-1.0	Acceleration override

1) With value -1, the respective parameter value is taken from the default on the TO

Table 4- 91 sLPacAxisMechanicalDataType structure

Element	Data type	Initial value	Meaning
r32ModuloLength	REAL	0.0	Modulo length (conveyor pocket length)
boModuloActive	BOOL	FALSE	Activate modulo length

4.5.1.3 Integration of the module

Technology objects, basic parameterization and block call

Table 4- 92 TO type and basic parameterizations

Axis / technology object	
TO type	Position axis
Version	Linear axis
Modulo	Not relevant, because for use as foil feed block, the actual position is reset to zero after each positioning
Block call	BackgroundTask

Further blocks and libraries

The **FBLPacFoilFeedPos** block uses the following blocks internally.

Block/library	Function
FBLMCBasicAxis / LMCBasic	Axis function block
FBLPacPrintMarkCorrection / LPac	Block for print mark correction
FBLPacSlipCompensation / LPac	Block for slip compensation
FBLPacChangeTOConfigDataMod / LPac	Function block to change the configuration data of the TO axis

Some of the internally used blocks originate from other libraries. See Section Additional required libraries (Page 15).

Error messages

Error number [HEX]	Meaning
16#0000	ERR_ERR_NO; error-free
16#8000	LPAC_ERR_INVALID_STEP; internal error, please get in touch with your Siemens contact partner
16#8001	LPAC_ERR_NO_AXIS; no axis TO specified

Error number [HEX]	Meaning
16#8003	LPAC_ERR_TO_ALARM_ACTIVE; TO alarm active; error message from axis function block Error description, see *SIMOTION Axis Function Block* Application Manual and Section Alarm and error messages (Page 333) in this manual
16#8004	LPAC_ERR_AXIS_NOT_DISABLED; axis not disabled for change of the modulo length
16#8005	LPAC_ERR_MODULO_LENGTH; invalid modulo length
16#8008	LPAC_ERR_POS_VELOCITY_ZERO; positioning velocity = 0
16#8009	LPAC_ERR_POS_ACC_DEC_ZERO; acceleration/deceleration = 0
16#800B	LPAC_ERR_POS_ABS_POSITION_VALUE; target position for absolute positioning < 0.0 or > modulo length
16#8011	LPAC_ERR_AXIS_FUNCTION_BLOC; error message from axis function block For details, see *diagnosticData.b32ErrorCode* output structure Error description, see *SIMOTION Axis Function Block* Application Manual
16#8014	LPAC_ERR_AXIS_NOT_HOMED; axis not homed
16#8060	LPAC_ERR_NO_MEASURING_INPUT; no TO measuringInput specified
16#8061	LPAC_ERR_NO_EXTERNAL_ENCODER; no TO externalEncoder specified
16#8062	LPAC_ERR_PRINT_MARK_VELOCITY_ZERO; correction velocity for print mark correction = 0
16#8063	LPAC_ERR_PRINT_MARK_ACC_ZERO; correction acceleration for print mark correction = 0
16#8064	LPAC_ERR_PRINT_MARK_CALC_FACT_ZERO; correction factor for print mark correction = 0
16#8065	LPAC_ERR_ENABLE_MEASURING_INPUT; error when enabling the measuring input; source *FBLPacPrintmarkCorrection*; return value of the SIMOTION *_enableMeasuringInput* function is equivalent to the *_move* function and can be fetched in hex format in the high word from *errorID*
16#8066	LPAC_ERR_NO_PRINTMARK_FOUND; no print mark found
16#8068	LPAC_ERR_MAX_PRINTMARK_ERRORS_REACHED; max. permissible number of faulty print marks attained
16#8069	LPAC_ERR_MEASURING_ERROR; error on the measuring input TO; source FBLPacPrintmarkCorrection
16#8070	LPAC_ERR_SLIP_COMP_VELOCITY_ZERO; correction velocity for slip compensation ≤ 0.0; source FBLPacSlipCompensation
16#8071	LPAC_ERR_SLIP_COMP_ACC_ZERO; correction acceleration for slip compensation ≤ 0.0; source FBLPacSlipCompensation
16#8073	LPAC_ERR_DISABLE_MEASUREING_INPUT; error from the _disableMeasrureingInput command; see diagnosticData; source FBLPacPrintmarkCorrection; return value of the SIMOTION *_disableMeasuringInput* function is equivalent to the *_move* function and can be fetched in hex format in the high word from *errorID*
16#8074	LPAC_ERR_GET_STATE_OF_AXIS_COMMAND; error from the _getStateOfAxisCommand command; see diagnosticData; source FBLPacPrintmarkCorrection or FBLPacSlipCompensation; return value of the SIMOTION *_getStateOfAxisCommand* function is equivalent to the *_move* function and can be fetched in hex format in the high word from *errorID*

Error number [HEX]	Meaning
16#8075	LPAC_ERR_REDEFINE_POSITION; error from the _redefinePosition command; source FBLPacPrintmarkCorrection or FBLPacSlipCompensation; return value of the SIMOTION *_redefinePosition* function is equivalent to the *_move* function and can be fetched in hex format in the high word from *errorID*
16#8076	LPAC_ERR_STOP; error from the _stop command; source FBLPacPrintmarkCorrection or FBLPacSlipCompensation; return value of the SIMOTION *_stop* function is equivalent to the *_move* function and can be fetched in hex format in the high word from *errorID*
16#8077	LPAC_ERR_ENABLE_EXTERNAL_ENCODER; error from the *_enableExternalEncoder* command; source *FBLPacSlipCompensation*; return value of the SIMOTION *_enableExternalEncoder* function is equivalent to the *_move* function and can be fetched in hex format in the high word from *errorID*
16#8078	LPAC_ERR_REDEFINE_EXTERNAL_ENCODER_POSITION; error from the *_redefineExternalEncoderPosition* command; source *FBLPacPrintmarkCorrection* or FBLPacSlipCompensation; return value of the SIMOTION *_redefineExternalEncoderPosition* function is equivalent to the *_move* function and can be fetched in hex format in the high word from *errorID*
16#8079	LPAC_ERROR_RESET_MEASURING_INPUT; error when resetting the measuring input; return value of the SIMOTION *_resetMeasuringInput* function is equivalent to the *_move* function and can be read in hex format in the high word from *errorID*
16#80A5	LPAC_ERR_POS; error from the _pos command; source FBLPacPrintmarkCorrection or *FBLPacSlipCompensation*; return value of the SIMOTION *_pos* function is equivalent to the *_move* function and can be fetched in hex format in the high word from *errorID*
16#80C3	LPAC_ERR_EXECUTE_DURING_RESET; Execute requested while error acknowledgment active; source FBLPacChangeTOConfigDataMod
16#8100	LPAC_ERR_READ_SYSTEM_VARIABLE; error when reading the system variables / variable cannot be read; source FBLPacChangeTOConfigData
16#8101	LPAC_ERR_MAX_TIME_FOR_TO_RESTART; TO restart not performed within the monitoring time; source FBLPacChangeTOConfigData

Note

The descriptions of the optional Print mark correction and Slip compensation functions are contained in the associated FBLPacPrintmarkCorrection (Page 170) and FBLPacSlipCompensation (Page 165) block descriptions in this manual.

Note

The **pFoilFeedPos** program unit is available in the LPac project to illustrate the use of the **FBLPacFoilFeedPos** block.

This unit contains the programs for the required block calls in the StartupTask and the BackgroundTask.

4.5.2 FBLPacFoilFeedGear - foil extractor gearing

4.5.2.1 Technological consideration

The **FBLPacFoilFeedGear** block controls the "foil feed" machine module on continuously operating tubular bagging machines. The foil feed operates in gearing to a machine leading axis (master). Another axis or an external encoder can be used as leading axis. The block also provides the optional print mark correction and slip compensation correction possibilities.

The module can be used together with other blocks of the **LPac** library for implementing a SIMOTION complete application of a continuous tubular bagging machine and also independently for similar mechanically-oriented machine units that require an electronic gearing with compensation options for a leading axis.

4.5.2.2 Module description

Scope of functions

The **FBLPacFoilFeedGear** function block offers the main functions and auxiliary functions as well as possible options shown in the following overview.

Function	Description	Implementation
Main function	Gearing	The gearing can be activated or deactivated via the *executeGearIn* or *executeGearOut* input.
Options	Print mark correction	The foil position can be corrected via a positioning motion superimposed on the basic motion with the aid of a print mark sensor.
	Slip compensation	A measured slip between the foil feed axis and the foil itself can be corrected via a positioning motion superimposed on the basic motion.
Auxiliary functions	Enable axis	The axis is enabled via the level-triggered *enableAxis* input.
	Home axis	The homing procedure of the axis is started via the *executeHoming* input.
	Jog axis	Positive or negative jogging can be started via the level-triggered *jogForward* or *jogBackward* inputs.
	Position axis	An absolute or relative positioning action can be started via the edge-triggered *executePos* input.
	Axis quick stop	A quick stop can be performed or a motion of the axis prevented via the level-triggered *enableEStop* input.

Function	Description	Implementation
	Change of the axis configuration data	The axis configuration data (modulo length, modulo yes/no) can be changed via the subordinate **FBLPacChangeTOConfigDataMod** function block. The modulo length specified in the ramp-up of the controller is automatically entered and activated by the block in the appropriate configuration data element of the axis. The modulo properties can also be changed at any time in the cyclic program by specifying different values. The user must only ensure that the axis is not enabled. A TO restart must be performed to activate the changed configuration data.

Note

A homing of the foil feed axis is not essential because no absolute synchronous operation between the leading axis (machine master or No Product No bag axis) is required. If, however, print mark correction should be used, a homing of the foil feed may still be desirable before starting the machine. In this case, a valid print mark is first sought and then positioning executed with a defined path (*r32RemainingWay).* This path should be specified so that the distance between the print mark and the cross-sealer after completion of this "remaining path" positioning is n x bag length. The print mark is thus located approximately at the desired target area. If no homing and remaining path positioning is performed, no focus area can be specified for the measuring input and the print mark position is located in the desired area only after several "correction cycles".

Schematic representation in LAD/FBD

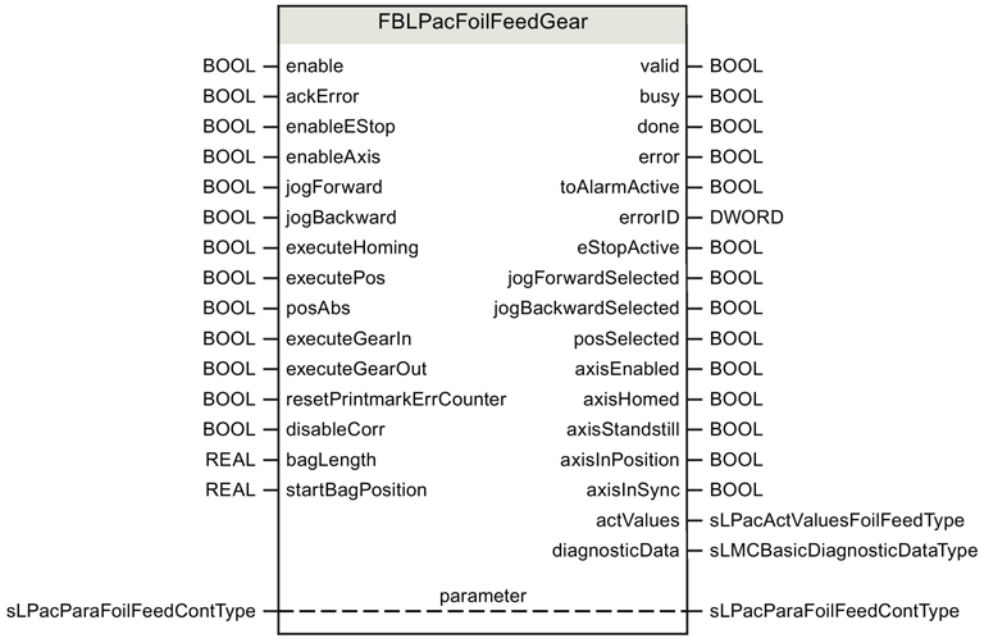


Figure 4-22 FBLPacFoilFeedGear - schematic block view in LAD/FBD

Input/output parameters

Table 4- 93 Input/output signals

Element	P type [1]	Data type	Initial value	Meaning
enable	IN	BOOL	FALSE	Activation of the FB
ackError	IN	BOOL	FALSE	Acknowledge error
enableEStop	IN	BOOL	FALSE	Activate quick stop
enableAxis	IN	BOOL	FALSE	Enable axis
jogForward	IN	BOOL	FALSE	Continuous jogging in the positive direction
jogBackward	IN	BOOL	FALSE	Continuous jogging in the negative direction
executeHoming	IN	BOOL	FALSE	Start homing
executePos	IN	BOOL	FALSE	Start positioning
posAbs	IN	BOOL	FALSE	TRUE: The positioning initiated with executePos is absolute
				FALSE: The positioning initiated with executePos is relative

Element	P type [1]	Data type	Initial value	Meaning
executeGearIn	IN	BOOL	FALSE	Start synchronization / activate gearing
executeGearOut	IN	BOOL	FALSE	Start desynchronization / deactivate gearing
resetPrintmark-ErrCounter	IN	BOOL	FALSE	Reset print mark error counter
disableCorr	IN	BOOL	FALSE	Suppress correction motion through print mark correction or slip compensation
bagLength	IN	REAL	0.0	Product length
startBagPosition	IN	REAL	0.0	Starting position
valid	OUT	BOOL	FALSE	Block is active
busy	OUT	BOOL	FALSE	Selected functionality being processed
done	OUT	BOOL	FALSE	Selected functionality completed successfully
error	OUT	BOOL	FALSE	Error occurred
toAlarmActive	OUT	BOOL	FALSE	TO alarm occurred
errorID	OUT	DWORD	16#0000_0000	Error code
eStopActive	OUT	BOOL	FALSE	Quick stop active
jogForward Selected	OUT	BOOL	FALSE	Jogging positive direction selected
jogBackward-Selected	OUT	BOOL	FALSE	Jogging negative direction selected
posSelected	OUT	BOOL	FALSE	Positioning selected
axisEnabled	OUT	BOOL	FALSE	Axis enabled
axisHomed	OUT	BOOL	FALSE	Axis homed
axisStandStill	OUT	BOOL	FALSE	Axis at standstill
axisInPosition	OUT	BOOL	FALSE	Axis in position
axisInSync	OUT	BOOL	FALSE	Axis synchronous to the master axis
actValues	OUT	sLPacActValues FoilFeedType	-	Actual values of the foil feed
diagnosticData	OUT	sLMCBasicDiagnostic DataType	-	For the diagnostics structure, see Alarm and error messages (Page 43)
parameter	IN_OUT	sLPacParaFoilFeed-ContType	-	Parameter for foil feed

1) Parameter types: IN = input parameter, OUT = output parameter, IN_OUT = in-out parameter

Table 4- 94 sLPacActValuesFoilFeedType structure

Element	Data type	Initial value	Meaning
r32ActualVelocity	REAL	0.0	Actual velocity of the foil feed [mm/s]
r32ActualPosition	REAL	0.0	Current position of the foil feed
r32PrintMarkDifference	REAL	0.0	Measured offset of the print mark with activated print mark correction [mm]
u32PrintMarkErrorCounter	UDINT	0	Current number of print mark errors for which no print mark was detected.
u32PrintMarkCorrectionCounter	UDINT	0	Current number of executed correction motions
r32SlipDifference	REAL	0.0	Measured slip with activated slip compensation [mm]

Table 4- 95 sLPacParaFoilFeedContType structure

Element	Data type	Meaning
sTOs	sLPacToCamDataType	TOs for foil feed
sMotionData	sLPacFoilFeedContMotionDataType	Parameters for EStop, homing, jog, pos, gearing
sConfigData	sLPacFoilFeedConfigDataType	Configuration data for foil feed

Table 4- 96 sLPacToCamDataType structure

Element	Data type	Meaning
toAxis	followingAxis	Foil feed axis TO (following axis)
toLeadingValue	masterType	TO master axis (at least position axis or external encoder)
toCam	CamType	TO for cam (not relevant for FBLPacFoilFeedGear)

Table 4- 97 sLPacFoilFeedContMotionDataType structure

Element	Data type	Meaning
sEStopData	sLMCBasicEStopDataType	Parameter for quick stop
sHomingData	sLMCBasicHomingDataType	Parameter for homing
sJogData	sLPacJogDataType	Parameter for jogging
sPosData	sLPacPosDataType	Parameter for positioning
sGearingData	sLPacGearingDataType	Parameter for gearing
sCammingData	sLPacCammingDataType	Not relevant, only for FBLPacFoilFeedCam

Quick stop parameters - sEStopData structure

See Axis quick stop (Page 43)

Homing parameters - sHomingData structure

See Home axis (Page 35)

Jog parameters - sJogData structure

See Jog axis (Page 36)

Positioning parameters - sPosData structure

See Position axis (Page 37)

Gearing parameters - sGearingData structure

See Synchronize/desynchronize axis in gearing (Page 38)

Table 4- 98 sLPacFoilFeedConfigDataType structure

Element	Data type	Initial value	Meaning
r32Position	REAL	0.0	Position
r32maxVelocity	REAL	0.0	Max. velocity
r32maxAcceleration	REAL	0.0	Max. acceleration
r32maxDeceleration	REAL	0.0	Max. deceleration
sPrintMarkCorr	sLPacPrintMarkCorrType	-	Parameter for print mark correction
sParaSlipComp	sLPacSlipCompType	-	Parameter for slip compensation
sInterfacePosData	sLPacFFPosMaxDataType	-	Positioning data interface to the FBLPacMasterInt block (not relevant)
sAxisMechanicalData	sLPacAxisMechanical-DataType	-	Axis data

Table 4- 99 sLPacPrintMarkCorrType structure for print mark correction option

Element	Data type	Initial value	Meaning
boPrintMarkCorr	BOOL	FALSE	Print mark correction activation
toMeasuringInput	MeasuringInputType	TO#NIL	Measuring input object
r32PrintMarkPos	REAL	0.0	Position setpoint of the print mark [mm]
r32RemainingWay FromPrintMark	REAL	0.0	Distance between print mark and sealing point that should also be traversed
r32MinPrintMarkCorr	REAL	0.0	Minimum print mark deviation above which a correction motion is performed
r32MaxPrintMarkCorr	REAL	0.0	Maximum permissible correction
u32MaxValueForPrintMarkErrors	UDINT	1	Maximum number of permitted errors for the print mark detection
r32MeasRangePosStart	REAL	0.0	Start of the measuring range
r32MeasRangePosEnd	REAL	0.0	End of the measuring range
r32CalcFact	REAL	1.0	Factor by which the measured path difference is to compensated
r32Velocity	REAL	0.0	Correction velocity
r32Acceleration	REAL	0.0	Correction acceleration and deceleration
boEdge	BOOL	TRUE	TRUE: Positive edge of the print mark is evaluated
			FALSE: Negative edge of the print mark is evaluated
boCorrInvert	BOOL	FALSE	TRUE: Inversion of the measured deviation of the print mark position

Table 4- 100 sLPacSlipCompType structure for the slip compensation option

Element	Data type	Initial value	Meaning
boSlipComp	BOOL	FALSE	Slip compensation activation
toExternalEncoder	externalEncoderType	TO#NIL	TO for external encoder
r32Velocity	REAL	0.0	Correction velocity
r32Acceleration	REAL	0.0	Correction acceleration
r32MaxSlip	REAL	0.0	Slip as of which a compensating motion is executed

Table 4- 101 sLPacAxisMechanicalDataType structure

Element	Data type	Initial value	Meaning
r32ModuloLength	REAL	360.0	Modulo length (e.g. bag length)
boModuloActive	BOOL	TRUE	Activate modulo length

4.5.2.3 Integration of the module

Technology objects, basic parameterization and block call

Table 4- 102 TO type and basic parameterizations

Axis / technology object	
TO type	Real synchronous axis
Version	Linear axis
Modulo	Leadscrew pitch of the axis with the result 1 path unit = 1 mm as target
Block call	BackgroundTask

Further blocks and libraries

The **FBLPacFoilFeedGear** block uses the following blocks internally.

Block/library	**Function**
FBLMCBasicAxis / LMCBasic	Axis function block
FBLPacPrintMarkCorrection / LPac	Function block for the print mark correction
FBLPacSlipCompensation / LPac	Function block for the slip compensation
FBLPacChangeTOConfigDataMod / LPac	Function block to change the configuration data of the TO axis

Some of the internally used blocks originate from other libraries. See Section Additional required libraries (Page 15).

Error messages

Error number [HEX]	**Meaning**
16#0000	ERR_ERR_NO; error-free
16#8000	LPAC_ERR_INVALID_STEP; internal error, please get in touch with your Siemens contact partner
16#8001	LPAC_ERR_NO_AXIS; no axis TO specified
16#8003	LPAC_ERR_TO_ALARM_ACTIVE; TO alarm active; error message from axis function block Error description, see *SIMOTION Axis Function Block* Application Manual and Section Alarm and error messages (Page 333) in this manual
16#8004	LPAC_ERR_AXIS_NOT_DISABLED; axis not disabled for change of the modulo length
16#8005	LPAC_ERR_MODULO_LENGTH; invalid modulo length
16#8008	LPAC_ERR_POS_VELOCITY_ZERO; positioning velocity = 0
16#8009	LPAC_ERR_POS_ACC_DEC_ZERO; acceleration/deceleration = 0
16#800B	LPAC_ERR_POS_ABS_POSITION_VALUE; target position for absolute positioning < 0.0 or > modulo length

Error number [HEX]	Meaning
16#8010	LPAC_ERR_NO_MASTER_AXIS; no axis TO specified for leading axis
16#8011	LPAC_ERR_AXIS_FUNCTION_BLOC; error message from axis function block For details, see *diagnosticData.b32ErrorCode* output structure Error description, see *SIMOTION Axis Function Block* Application Manual
16#8013	LPAC_ERR_NO_FOLLOWING_OBJECT; no following object specified or leading axis is not a following axis
16#8016	LPAC_ERR_GEAR_RATIO; gear ratio numerator or denominator = 0
16#8060	LPAC_ERR_NO_MEASURING_INPUT; no TO measuringInput specified
16#8061	LPAC_ERR_NO_EXTERNAL_ENCODER; no TO externalEncoder specified
16#8062	LPAC_ERR_PRINT_MARK_VELOCITY_ZERO; correction velocity for print mark correction = 0
16#8063	LPAC_ERR_PRINT_MARK_ACC_ZERO; correction acceleration for print mark correction = 0
16#8064	LPAC_ERR_PRINT_MARK_CALC_FACT_ZERO; correction factor for print mark correction = 0
16#8065	LPAC_ERR_ENABLE_MEASURING_INPUT; error when enabling the measuring input
16#8066	LPAC_ERR_NO_PRINTMARK_FOUND; no print mark found
16#8068	LPAC_ERR_MAX_PRINTMARK_ERRORS_REACHED; max. permissible number of faulty print marks attained
16#8069	LPAC_ERR_MEASURING_ERROR; error on the measuring input TO; source FBLPacPrintmarkCorrection
16#8070	LPAC_ERR_SLIP_COMP_VELOCITY_ZERO; correction velocity for slip compensation = 0.0; source FBLPacSlipCompensation
16#8071	LPAC_ERR_SLIP_COMP_ACC_ZERO; correction acceleration for slip compensation = 0.0; source FBLPacSlipCompensation
16#8073	LPAC_ERR_DISABLE_MEASUREING_INPUT; error from the _disableMeasrureingInput command; see diagnosticData; source FBLPacPrintmarkCorrection
16#8074	LPAC_ERR_GET_STATE_OF_AXIS_COMMAND; error from the _getStateOfAxisCommand command; see diagnosticData; source FBLPacPrintmarkCorrection or FBLPacSlipCompensation
16#8075	LPAC_ERR_REDEFINE_POSITION; error from the _redefinePosition command; see diagnosticData; source FBLPacPrintmarkCorrection or FBLPacSlipCompensation
16#8076	LPAC_ERR_STOP; error from the _stop command; see diagnosticData; source FBLPacPrintmarkCorrection or FBLPacSlipCompensation
16#80A5	LPAC_ERR_POS; error from the _pos command; see diagnosticData; source FBLPacPrintmarkCorrection
16#8077	LPAC_ERR_ENABLE_EXTERNAL_ENCODER; error from the _enableExternalEncoder command; see diagnosticData
16#8078	LPAC_ERR_REDEFINE_EXTERNAL_ENCODER_POSITION; error from the _redefineExternalEncoderPosition command; see diagnosticData
16#8079	LPAC_ERR_RESET_MEASURING_INPUT; error from the _resetMeasruringInput command; see diagnosticData

Error number [HEX]	Meaning
16#80C3	LPAC_ERR_EXECUTE_DURING_RESET; Execute requested while error acknowledgment active; source FBLPacChangeTOConfigData
16#8100	LPAC_ERR_READ_SYSTEM_VARIABLE; error when reading the system variables / variable cannot be read; source FBLPacChangeTOConfigData
16#8101	LPAC_ERR_MAX_TIME_FOR_TO_RESTART; TO restart not performed within the monitoring time; source FBLPacChangeTOConfigData

Note

The descriptions of the optional Print mark correction and Slip compensation functions are contained in the associated FBLPacPrintmarkCorrection (Page 170) and FBLPacSlipCompensation (Page 165) block descriptions in this manual.

Note

The **pFoilFeedGear** program unit is available in the LPac project to illustrate the use of the **FBLPacFoilFeedGear** block.

This unit contains the programs for the required block calls in the StartupTask and the BackgroundTask.

4.5.3 FBLPacFoilFeedCam - foil extractor camming

4.5.3.1 Technological consideration

The **FBLPacFoilFeedCam** block controls the "foil feed" machine module on continuously operating tubular bagging machines. The foil feed operates in camming to a machine leading axis (master). The block also provides the optional print mark correction and slip compensation correction possibilities.

This block can be used together with other blocks of the **LPac** library for implementing a SIMOTION complete application of a continuous tubular bagging machine and also independently for similar mechanically-oriented machine units that require an electronic camming with compensation options for a leading axis.

4.5.3.2 Module description

Scope of functions

The **FBLPacFoilFeedCam** function block offers the main functions and auxiliary functions as well as possible options shown in the following overview.

Function	Description	Implementation
Main function	Camming	The camming can be activated or deactivated via the *executeCamIn* or *executeCamOut* input.
Options	Print mark correction	The foil position can be corrected via a positioning motion superimposed on the basic motion with the aid of a print mark sensor.
	Slip compensation	A measured slip between the foil feed axis and the foil itself can be corrected via a positioning motion superimposed on the basic motion.
Auxiliary functions	Enable axis	The axis is activated via the level-triggered *enableAxis* input.
	Home axis	The homing procedure of the axis is started via the *executeHoming* input.
	Jog axis	Positive or negative jogging can be started via the level-triggered *jogForward* or *jogBackward* inputs.
	Position axis	An absolute or relative positioning action can be started via the edge-triggered *executePos* input.
	Axis quick stop	A quick stop can be performed or a motion of the axis prevented via the level-triggered *enableEStop* input.
	Change of the axis configuration data	The axis configuration data (modulo length, modulo yes/no) can be changed via the subordinate **FBLPacChangeTOConfigDataMod** function block. The modulo length specified in the ramp-up of the controller is automatically entered and activated by the block in the appropriate configuration data element of the axis. The modulo properties can also be changed at any time in the cyclic program by specifying different values. The user must only ensure that the axis is not enabled. A TO restart must be performed to activate the changed configuration data.

Note

A homing of the foil feed axis is not essential because no absolute synchronous operation between the leading axis (machine master or No Product No bag axis) is required. If, however, print mark correction should be used, a homing of the foil feed may still be desirable before starting the machine. In this case, a valid print mark is first sought and then positioning executed with a defined path (*r32RemainingWay).* This path should be specified so that the distance between the print mark and the cross-sealer after completion of this "remaining path" positioning is n x bag length. The print mark is thus located approximately at the desired target area. If no homing and remaining path positioning is performed, no focus area can be specified for the measuring input and the print mark position is located in the desired area only after several "correction cycles".

Schematic representation in LAD/FBD

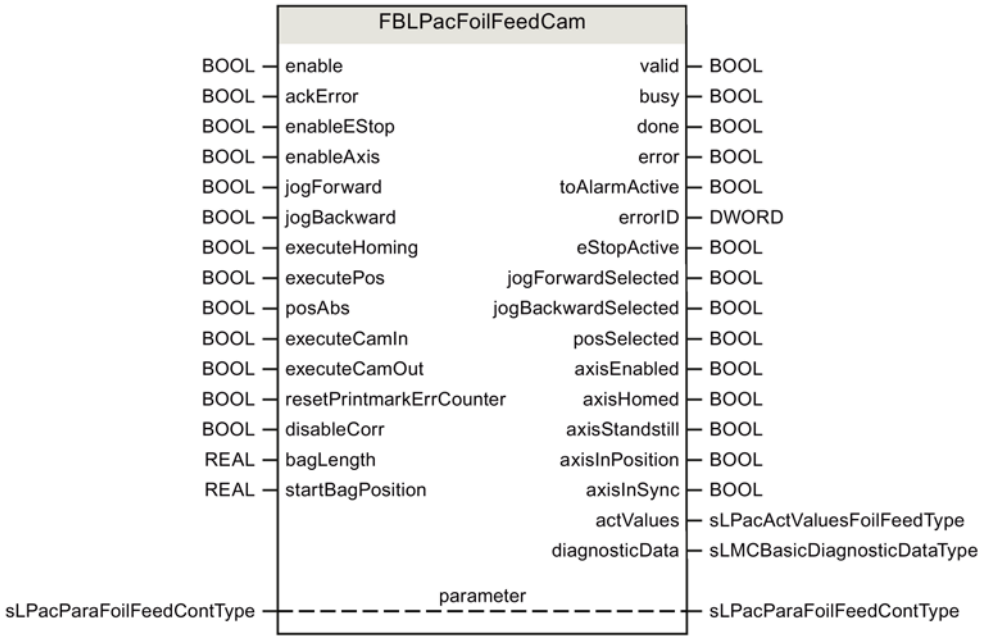


Figure 4-23 FBLPacFoilFeedCam - schematic block view in LAD/FBD

Input and output parameters

Table 4- 103 Input and output parameters

Parameter	P type [1]	Data type	Initial value	Meaning
enable	IN	BOOL	FALSE	Activation of the FB
ackError	IN	BOOL	FALSE	Acknowledge error
enableEStop	IN	BOOL	FALSE	Activate quick stop
enableAxis	IN	BOOL	FALSE	Enable axis

Parameter	P type [1)]	Data type	Initial value	Meaning
jogForward	IN	BOOL	FALSE	Continuous/incremental jogging in the positive direction
jogBackward	IN	BOOL	FALSE	Continuous/incremental jogging in the negative direction
executeHoming	IN	BOOL	FALSE	Homing is being executed
executePos	IN	BOOL	FALSE	Start positioning
posAbs	IN	BOOL	FALSE	TRUE: The positioning initiated with executePos is absolute
				FALSE: The positioning initiated with executePos is relative
executeCamIn	IN	BOOL	FALSE	Activate camming
executeCamOut	IN	BOOL	FALSE	Deactivate camming
resetPrintMark-ErrCounter	IN	BOOL	FALSE	Reset print mark error counter
disableCorr	IN	BOOL	FALSE	Suppress correction motion through print mark correction or slip compensation
bagLength	IN	REAL	0.0	Product length
startBagPosition	IN	REAL	0.0	Starting position
valid	OUT	BOOL	FALSE	Block has been selected and can be used
busy	OUT	BOOL	FALSE	Selected functionality being processed
done	OUT	BOOL	FALSE	Selected functionality completed successfully
error	OUT	BOOL	FALSE	Error occurred
toAlarmActive	OUT	BOOL	FALSE	TO alarm occurred
errorID	OUT	DWORD	16#0000_0000	Error code
eStopActive	OUT	BOOL	FALSE	Quick stop active
jogForward-Selected	OUT	BOOL	FALSE	Jogging positive direction selected
jogBackward-Selected	OUT	BOOL	FALSE	Jogging negative direction selected
posSelected	OUT	BOOL	FALSE	Positioning selected
axisEnabled	OUT	BOOL	FALSE	Axis enabled
axisHomed	OUT	BOOL	FALSE	Axis homed
axisStandstill	OUT	BOOL	FALSE	Axis at standstill
axisInPosition	OUT	BOOL	FALSE	Axis in position
axisInSync	OUT	BOOL	FALSE	Axis synchronous to the master axis
actValues	OUT	sLPacAc-tValuesFoilFeedType	-	Current operating data of the foil feed

Parameter	P type [1]	Data type	Initial value	Meaning
diagnosticData	OUT	sLMCBasic Diagnostic DataType	-	For the diagnostics structure, see Alarm and error messages (Page 43)
parameter	IN_OUT	sLPacPara-FoilFeed-ContType	-	Parameter for foil feed

1) Parameter types: IN = input parameter, OUT = output parameter, IN_OUT = in-out parameter

Table 4- 104 sLPacActValuesFoilFeedType structure

Element	Data type	Initial value	Meaning
r32ActualVelocity	REAL	0.0	Actual velocity of the foil feed [mm/s]
r32ActualPosition	REAL	0.0	Current position of the foil feed
r32PrintMarkDifference	REAL	0.0	Measured offset of the print mark with activated print mark correction [mm]
u32PrintMarkErrorCounter	UDINT	0	Current number of print mark errors for which no print mark was detected
u32PrintMarkCorrectionCounter	UDINT	0	Current number of executed correction motions
r32SlipDifference	REAL	0.0	Measured slip with activated slip compensation [mm]

Table 4- 105 sLPacParaFoilFeedContType structure

Element	Data type	Meaning
sTOs	sLPacToCamDataType	TO for the foil feed
sMotionData	sLPacFoilFeedContMotionDataType	Parameters for EStop, homing, jog, pos, camming
sConfigData	sLPacFoilFeedConfigDataType	Configuration data for foil feed

Table 4- 106 sLPacToCamDataType structure

Element	Data type	Meaning
toAxis	followingAxis	Foil feed axis TO (following axis)
toLeadingValue	masterType	TO master axis (at least position axis or external encoder)
toCam	CamType	TO for cam

Table 4- 107 sLPacFoilFeedContMotionDataType structure

Element	Data type	Meaning
sEStopData	sLMCBasicEStopDataType	Parameter for quick stop
sHomingData	sLMCBasicHomingDataType	Parameter for homing
sJogData	sLPacJogDataType	Parameter for jogging
sPosData	sLPacPosDataType	Parameter for positioning
sGearingData	sLPacGearingDataType	Not relevant, only for FBLPacFoilFeedGear
sCammingData	sLPacCammingDataType	Parameter for camming

Quick stop parameters - sEStopData structure

See Axis quick stop (Page 43)

Homing parameters - sHomingData structure

See Home axis

Jog parameters - sJogData structure

See Jog axis (Page 36)

Positioning parameters - sPosData structure

See Position axis (Page 37)

Camming parameters - sCammingData structure

See Synchronizing/desynchronizing axis in camming (Page 40)

Table 4- 108 sLPacFoilFeedConfigDataType structure

Element	Data type	Initial value	Meaning
r32Position	REAL	0.0	Position
r32maxVelocity	REAL	0.0	Max. velocity
r32maxAcceleration	REAL	0.0	Max. acceleration
r32maxDeceleration	REAL	0.0	Max. deceleration
sPrintMarkCorr	sLPac-PrintMark CorrType	-	Parameter for print mark correction
sParaSlipComp	sLPacSlip-CompType	-	Parameter for slip compensation

Element	Data type	Initial value	Meaning
sInterfacePosData	sLPacFF-PosMax DataType	-	Positioning data interface to the FBLPac-MasterInt block (not relevant)
sAxisMechanicalData	sLPacAxis Mechanical-DataType	-	Axis data

Table 4- 109 sLPacPrintMarkCorrType structure for print mark correction option

Element	Data type	Initial value	Meaning
boPrintMarkCorr	BOOL	FALSE	Print mark correction activation
toMeasuringInput	Measuring InputType	TO#NIL	Measuring input object
r32PrintMarkPos	REAL	0.0	Position setpoint of the print mark [mm]
r32RemainingWay FromPrintMark	REAL	0.0	Distance between print mark and sealing point that should also be traversed
r32MinPrintMarkCorr	REAL	0.0	Minimum print mark deviation above which a correction motion is performed
r32MaxPrintMarkCorr	REAL	0.0	Maximum permissible correction
u32MaxValueForPrint MarkErrors	UDINT	1	Maximum number of permitted errors for the print mark detection
r32MeasRangePosStart	REAL	0.0	Start of the measuring range
r32MeasRangePosEnd	REAL	0.0	End of the measuring range
r32CalcFact	REAL	1.0	Factor by which the measured path differ-ence is to compensated
r32Velocity	REAL	0.0	Correction velocity
r32Acceleration	REAL	0.0	Correction acceleration and deceleration
boEdge	BOOL	TRUE	TRUE: Positive edge of the print mark is evaluated
			FALSE: Negative edge of the print mark is evaluated
boCorrInvert	BOOL	FALSE	TRUE: Inversion of the measured deviation of the print mark position

Table 4- 110 sLPacSlipCompType structure for the slip compensation option

Element	Data type	Initial value	Meaning
boSlipComp	BOOL	FALSE	Slip compensation activation
toExternalEncoder	externalEncoder-Type	TO#NIL	TO for external encoder
r32Velocity	REAL	0.0	Correction velocity
r32Acceleration	REAL	0.0	Correction acceleration
r32MaxSlip	REAL	0.0	Slip as of which a compensating motion is executed

Table 4- 111 sLPacAxisMechanicalDataType structure

Element	Data type	Initial value	Meaning
r32ModuloLength	REAL	360.0	Modulo length (e.g. bag length)
boModuloActive	BOOL	TRUE	Activate modulo length

4.5.3.3 Integration of the module

Technology objects, basic parameterization and block call

Table 4- 112 TO type and basic parameterizations

Axis / technology object	
TO type	Following axis
Version	Linear axis
Modulo	Bag length
Block call	BackgroundTask

Further blocks and libraries

The **FBLPacFoilFeedCam** block uses the following blocks internally.

Block/library	Function
FBLMCBasicAxis / LMCBasic	Axis function block
FBLPacPrintMarkCorrection / LPac	Function block for the print mark correction
FBLPacSlipCompensation / LPac	Function block for the slip compensation
FBLPacChangeTOConfigDataMod / LPac	Function block to change the configuration data of the TO axis

Some of the internally used blocks originate from other libraries. See Section Additional required libraries (Page 15).

Error messages

Error number [HEX]	Meaning
16#0000	ERR_ERR_NO; error-free
16#8000	LPAC_ERR_INVALID_STEP; internal error, please get in touch with your Siemens contact partner
16#8001	LPAC_ERR_NO_AXIS; no axis TO specified
16#8003	LPAC_ERR_TO_ALARM_ACTIVE; TO alarm active; error message from axis function block Error description, see *SIMOTION Axis Function Block* Application Manual and Section Alarm and error messages (Page 333) in this manual

Error number [HEX]	Meaning
16#8004	LPAC_ERR_AXIS_NOT_DISABLED; axis not disabled for change of the modulo length
16#8005	LPAC_ERR_MODULO_LENGTH; invalid modulo length
16#8008	LPAC_ERR_POS_VELOCITY_ZERO; positioning velocity = 0
16#8009	LPAC_ERR_POS_ACC_DEC_ZERO; acceleration/deceleration = 0
16#800B	LPAC_ERR_POS_ABS_POSITION_VALUE; target position for absolute positioning < 0.0 or > modulo length
16#8010	LPAC_ERR_NO_MASTER_AXIS; no axis TO specified for leading axis
16#8011	LPAC_ERR_AXIS_FUNCTION_BLOC; error message from axis function block For details, see *diagnosticData.b32ErrorCode* output structure; Error description, see *SIMOTION Axis Function Block* Application Manual
16#8012	LPAC_ERR_NO_CAM, no cam specified
16#8013	LPAC_ERR_NO_FOLLOWING_OBJECT; no following object specified or leading axis is not a following axis
16#8060	LPAC_ERR_NO_MEASURING_INPUT; no TO measuringInput specified
16#8061	LPAC_ERR_NO_EXTERNAL_ENCODER; no external encoder TO specified
16#8062	LPAC_ERR_PRINT_MARK_VELOCITY_ZERO; correction velocity for print mark correction = 0
16#8063	LPAC_ERR_PRINT_MARK_ACC_ZERO; correction acceleration for print mark correction = 0
16#8064	LPAC_ERR_PRINT_MARK_CALC_FACT_ZERO; correction factor for print mark correction = 0
16#8065	LPAC_ERR_ENABLE_MEASURING_INPUT; error when enabling the measuring input
16#8066	LPAC_ERR_NO_PRINTMARK_FOUND; no print mark found
16#8068	LPAC_ERR_MAX_PRINTMARK_ERRORS_REACHED; max. permissible number of faulty print marks attained
16#8069	LPAC_ERR_MEASURING_ERROR; error on the measuring input TO; source FBLPacPrintmarkCorrection
16#8070	LPAC_ERR_SLIP_COMP_VELOCITY_ZERO; correction velocity for slip compensation = 0; see diagnosticData; source FBLPacSlipCompensation
16#8071	LPAC_ERR_SLIP_COMP_ACC_ZERO; correction acceleration for slip compensation = 0; see diagnosticData; source FBLPacSlipCompensation
16#8073	LPAC_ERR_DISABLE_MEASUREING_INPUT; error from the _disableMeasrureingInput command; see diagnosticData; source FBLPacPrintmarkCorrection
16#8074	LPAC_ERR_GET_STATE_OF_AXIS_COMMAND; error from the _getStateOfAxisCommand command; see diagnosticData; source FBLPacPrintmarkCorrection or FBLPacSlipCompensation
16#8075	LPAC_ERR_REDEFINE_POSITION; error from the _redefinePosition command; see diagnosticData; source FBLPacPrintmarkCorrection or FBLPacSlipCompensation
16#8076	LPAC_ERR_STOP; error from the _stop command; see diagnosticData; source FBLPacPrintmarkCorrection or FBLPacSlipCompensation

Error number [HEX]	Meaning
16#80A5	LPAC_ERR_POS; error from the _pos command; see diagnosticData; source FBLPacPrintmarkCorrection
16#8077	LPAC_ERR_ENABLE_EXTERNAL_ENCODER; error from the _enableExternalEncoder command; see diagnosticData
16#8079	LPAC_ERR_RESET_MEASURING_INPUT; error from the _resetMeasruringInput command; see diagnosticData
16#80A5	LPAC_ERR_POS; error from the _pos command; see diagnosticData; source FBLPacPrintmarkCorrection
16#80C3	LPAC_ERR_EXECUTE_DURING_RESET; Execute requested while error acknowledgment active; source FBLPacChangeTOConfigData
16#8100	LPAC_ERR_READ_SYSTEM_VARIABLE; error when reading the system variables / variable cannot be read; source FBLPacChangeTOConfigData
16#8101	LPAC_ERR_MAX_TIME_FOR_TO_RESTART; TO restart not performed within the monitoring time; source FBLPacChangeTOConfigData

Note

The descriptions of the optional Print mark correction and Slip compensation functions are contained in the associated FBLPacPrintmarkCorrection (Page 170) and FBLPacSlipCompensation (Page 165) block descriptions in this manual.

Note

The **pFoilFeedCam** program unit is available in the LPac project to illustrate the use of the **FBLPacFoilFeedCam** block.

This unit contains the programs for the required block calls in the StartupTask and the BackgroundTask.

4.5.4 FBLPacFoilGuide - sidelay control

4.5.4.1 Technological consideration

The **FBLPacFoilGuide** block enables the lateral foil guide to be influenced as in sidelay control.

The control is made by determining the sidelay position using two sensors.

If the sidelay runs exactly between the two sensors (*innerSensor* and *outerSensor*), the path is correct and will not be corrected. If, however, both sensors are switched or not switched, the path of the sidelay is no longer correct and must be corrected. The sensor signals are passed to the two *InnerSensor* and *OuterSensor* inputs. Both a manual and an automatic controlling of the sidelay are supported. The two *enableManFoilRight* and *enableManFoilLeft* inputs can be used at any time to manually move the sidelay to the right and to the left, respectively. If both inputs are set simultaneously, the traversing motion is stopped.

The *enableAutoFoilGuide* input activates the automatic control of the sidelay that corrects the path of the foil automatically. The manual correction can also be activated during the automatic sidelay control so that the foil can also be traversed manually. The invertSensors input can be used to specify the correction direction of the sidelay. The (invertSensors = FALSE) default setting describes the motion direction of the sidelay when the sensors are positioned on the right-hand side of the foil in the motion direction. If the sensors are located on the left-hand side, *invertSensors* must be assigned TRUE.

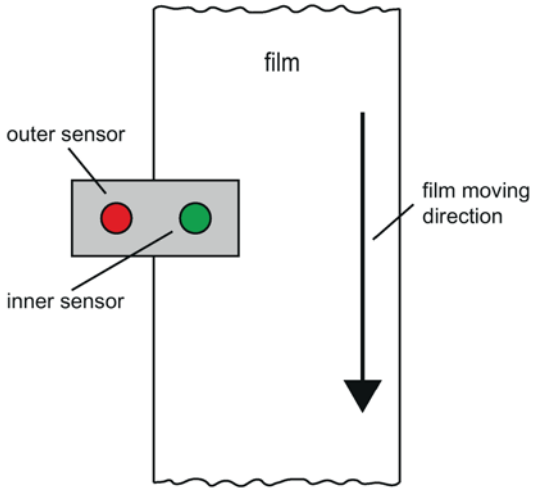


Figure 4-24 FBLPacFoilGuide - position of the sensors for invertSensors = FALSE

The following figure shows an example, with sensors positioned on the right-hand side (in the foil motion direction), of how the sidelay is moved depending on the switch state of the sensors:

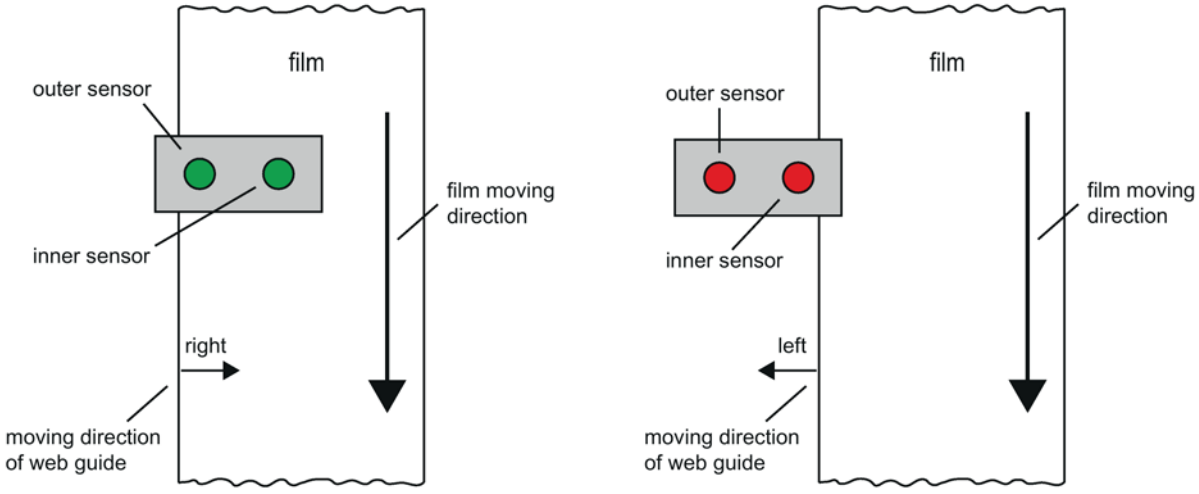


Figure 4-25 FBLPacFoilGuide - motion direction of the sidelay depending on the sensors

If both sensors are assigned (green), the foil has drifted too far to the left. It must be moved to the right so that the foil resumes the correct position. If, however, none of the sensors is assigned (red), the foil has moved too far to the right and so must be moved to the left. The two *moveFoilRight* and *moveFoilLeft* outputs are used to control the motion of the sidelay,

normally using a servo motor. For the correct evaluation of the sensors and the control of the sidelay, the FB must be called in a cyclic program.

The *r32MinOnTime* parameter can also be used to parameterize a minimum ON duration in [ms] for the two *moveFoilRight* and *moveFoilLeft* outputs. The *r32MaxReactionTime* time monitors the ON duration of the outputs. If one of the two *moveFoilRight* or *moveFoilLeft* outputs is activated for longer than the parameterized monitoring time *r32MaxReactionTime*, the two outputs are switched to FALSE and the error: LPAC_ERR_REACTION_TIME_OUT is output. In addition to monitoring the adjustment time, the adjustment range can be limited by two initiators passed to the *limitLeft* and *limitRight* inputs.

4.5.4.2 Module description

Scope of functions

The **FBLPacFoilGuide** function block offers the main functions and auxiliary functions as well as possible options shown in the following overview.

Function	Description	Implementation
Main function	Sidelay control	The sidelay control can be activated or deactivated via the level-triggered *enableAutoFoilGuide* input. The sidelay control is performed by the digital control of a servo motor in conjunction with the two *innerSensor and outerSensor* sensors via the *moveFilmRight* or *moveFilmLeft output*. To prevent an excessive switching of the two outputs, the minimum ON duration can be parameterized with the *r32MinOnTime* parameter. The reaction of the adjustment motion is monitored by a specifiable monitoring time in the *r32MaxReactionTime* parameter. The query logic of the two film sensors can be inverted via the *invertSensors* input.
Options	Limitation of the traversing distance	The traversing distance can be limited via the two level-triggered *limitLeft* and *limitRight* inputs.
Auxiliary functions	Manual control of the traversing distance	The two level-triggered *enableManFoilRight* and *enableManFilmLeft* inputs allow the servo motor to be controlled independent of the two sensors

Schematic representation in LAD/FBD

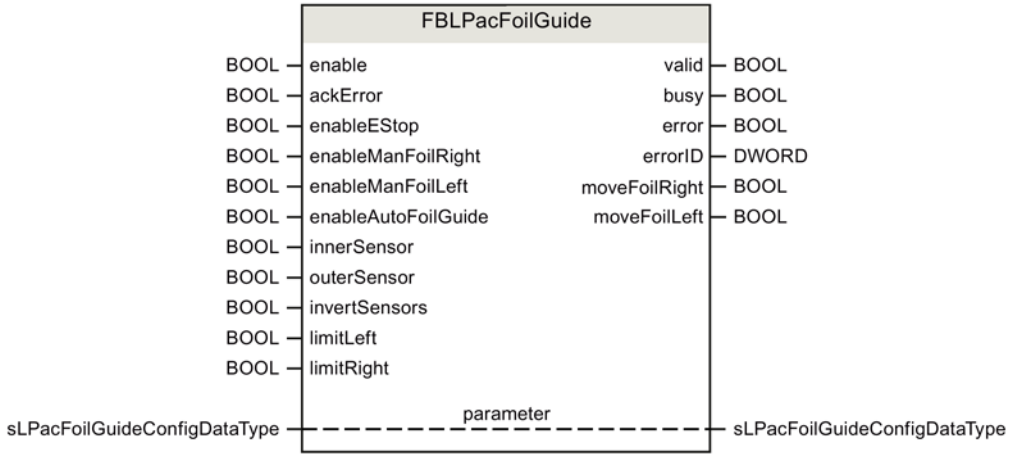


Figure 4-26 FBLPacFoilGuide - schematic block view in LAD/FBD

Input and output parameters

Table 4- 113 Input and output parameters

Parameter	P type [1]	Data type	Initial value	Meaning
enable	IN	BOOL	FALSE	Activation of the FB
ackError	IN	BOOL	FALSE	Acknowledge error
enableEStop	IN	BOOL	FALSE	Activate quick stop
enableManFoil-Right	IN	BOOL	FALSE	Request, traverse to the right
enableManFoilLeft	IN	BOOL	FALSE	Request, traverse to the left
enableAutoFoil-Guide	IN	BOOL	FALSE	Automatic mode activated, superim-posed manual traversing possible
innerSensor	IN	BOOL	FALSE	Signal, foil detected by inner sensor
outerSensor	IN	BOOL	FALSE	Signal, foil detected by outer sensor
invertSensor	IN	BOOL	FALSE	Inversion of the sensor signals
limitLeft	IN	BOOL	TRUE	Signal, left limit switch
limitRight	IN	BOOL	TRUE	Signal, right limit switch
valid	OUT	BOOL	FALSE	Block is active
busy	OUT	BOOL	FALSE	Selected functionality being processed
error	OUT	BOOL	FALSE	Error occurred
errorID	OUT	DWORD	16#0000_0000	Error code
moveFoilRight	OUT	BOOL	FALSE	Move the foil to the right

Parameter	P type [1)]	Data type	Initial value	Meaning
moveFoilLeft	OUT	BOOL	FALSE	Move the foil to the left
parameter	IN_OUT	sLPacPara FoilGuide Config DataType	-	Parameter for foil position

1) Parameter types: IN = input parameter, OUT = output parameter, IN_OUT = in-out parameter

Table 4- 114 sLPacFoilGuideConfigDataType structure

Element	Data type	Initial value	Meaning
r32MinOnTime	REAL	0.0	Minimum ON duration in [ms] for *moveFoilRight* or *moveFoilLeft*
r32MaxReactionTime	REAL	0.0	Monitoring time in [ms] for *moveFoilRight* or *moveFoilLeft*

4.5.4.3 Integration of the module

Technology objects, basic parameterization and block call

Table 4- 115 TO type and basic parameterizations

Axis / technology object	Not relevant
Block call	BackgroundTask

Further blocks and libraries

No further blocks from other libraries are called.

Error messages

Error number [HEX]	Meaning
16#0000	ERR_ERR_NO; error-free
16#8000	ERR_INVALID_STEP; internal error, please get in touch with your Siemens contact partner
16#80F0	LPAC_ERR_REACTION_TIME_OUT, no foil motion within the parameterized monitoring time
16#80F1	LPAC_ERR_INVALID_SENSOR_STATE; undefined sensor status

Note

The **pFoilGuide** program unit is available in the LPac project to illustrate the use of the **FBLPacFoilGuide** block.

This unit contains the programs for the required block calls in the StartupTask and the BackgroundTask.

4.5.5 FBLPacSlipCompensation - slip compensation

4.5.5.1 Technological consideration

The **FBLPacSlipCompensation** block compensates the slip between the drive roller, conveyors, etc. and a foil or paper conveyor, e.g. for tubular bagging machines, that can occur by the path being displaced.

The path position is measured directly on the path via a position encoder placed on the measuring wheel. The block then compares the path position with the motor position of the drive motor.

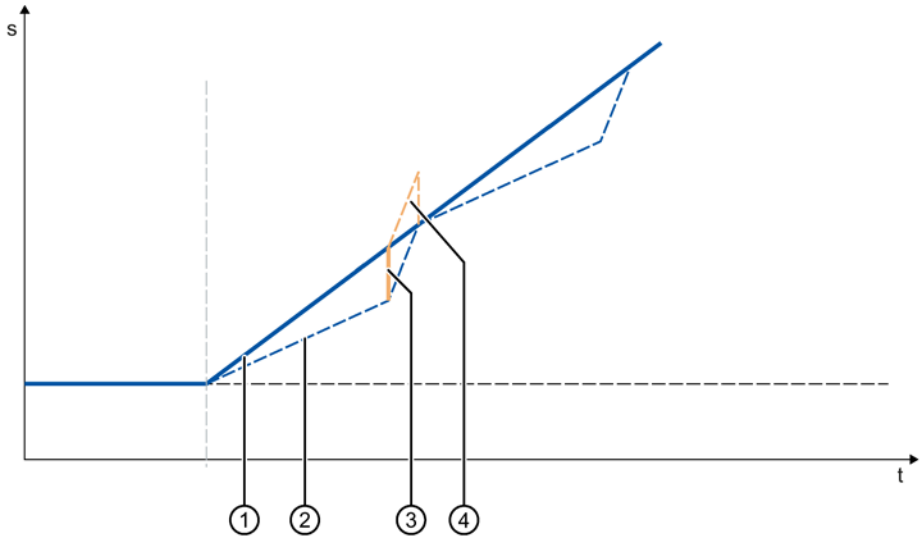
Without slip, both position values are identical or change at the same speed. No compensation would be required in this case.

Slip changes the position of the encoder on the foil path, normally with a lower velocity than the motor position.

The block compares both position values continuously and triggers an appropriate superimposed compensation motion when the parameterized maximum deviation is attained. This is done as a superimposed positioning with parameterizable velocity and acceleration/deceleration as trapezoidal profile. When the correction motion completes, a position correction re-establishes the reference to the transport medium.

The block ensures the required control (enable, set position, etc.) of the external encoder. The user needs only to pass the reference to the associated technology object to the block.

The figure below shows the method of operation of the block:



① Motor encoder position
② Measuring wheel position
③ Slip, maximum tolerated
④ Superimposed correction motion

Note

Because a superimposed positioning for the slip compensation is used, other superimposed motions, such as print mark correction, which also use the superimposed positioning must be coordinated by the application. Both correction possibilities must **not be activated simultaneously** on an axis! For example, either only the slip compensation (for unprinted foil) or the print mark correction is always activated (for printed foil with print marks) depending on the deployed foil of a tubular bagging machine.

4.5.5.2 Module description

Scope of functions

The **FBLPacSlipCompensation** function block offers the main functions and auxiliary functions as well as possible options shown in the following overview.

Function	Description	Implementation
Main function	Slip compensation	The slip monitoring is activated via the *enable* input. If the parameterizable threshold value (maximumSlip) is overshoot, a superimposed positioning compensates for the path difference. Velocity, acceleration/deceleration of the higher-level positioning can be parameterized.
Options	Suppress correction motion	The correction motion can be suppressed by setting the *disableCorr* input signal. This function can be used, for example, when the foil is in the sealing area and a correction motion is not desirable in this area.
Auxiliary functions	-	-

Schematic representation in LAD/FBD

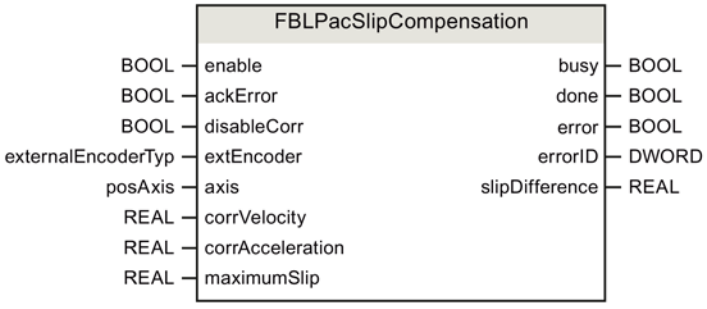


Figure 4-27 FBLPacSlipCompensation - schematic block view in LAD/FBD

Input and output parameters

Table 4- 116 Input and output parameters

Element	P type [1]	Data type	Initial value	Meaning
enable	IN	BOOL	FALSE	Start correction
ackError	IN	BOOL	FALSE	Reset error
disableCorr	IN	BOOL	FALSE	Suppress correction motion
extEncoder	IN	externalEncoder-Type	TO#NIL	External encoder, e.g. measuring wheel as reference position encoder

Element	P type [1]	Data type	Initial value	Meaning
axis	IN	posAxis	TO#NIL	Reference axis with motor encoder on which the compensation can be traversed
corrVelocity	IN	REAL	0.0	Correction velocity
corrAcceleration	IN	REAL	0.0	Acceleration/deceleration for correction motion
maximumSlip	IN	REAL	0.0	Maximum permissible slip as of which a compensating motion is performed
busy	OUT	BOOL	FALSE	Selected functionality being processed
done	OUT	BOOL	FALSE	Compensation motion is finished
error	OUT	BOOL	FALSE	Error occurred
errorID	OUT	DWORD	16#0000_0000	Error code
slipDifference	OUT	REAL	0.0	Current measured slip

1) Parameter types: IN = input parameter, OUT = output parameter

4.5.5.3 Integration of the module

The **FBLPacSlipCompensation** block is already contained in the **LPac** blocks for the **FBLPacFoilFeedPos**, **FBLPacFoilFeedGear** and **FBLPacFoilFeedCam** foil feed. This block is supplied and controlled via the interface. Consequently, a separate call is not required when these blocks are used. An isolated use of the block, however, is possible in other applications. The following supplementary conditions must be observed:

- For an emergency stop of the reference axis, ensure that the superimposed motion is also stopped.
- The block should be called only in cyclic tasks.

Technology objects, basic parameterization and block call

Table 4- 117 TO type and basic parameterizations

Axis / technology object	
TO type	At least position axis
External encoder	External encoder for the slip detection
Block call (for isolated application)	Cyclic call, e.g. IPO, IPO2 task

Further blocks and libraries

The **FBLPacSlipCompensation** block does not use any further blocks internally.

Error messages

Error number [HEX]	Meaning
16#0000	LPAC_ERR_NO; error-free
16#8003	LPAC_ERR_TO_ALARM_ACTIVE; TO alarm active; error message from axis function block Error description, see *SIMOTION Axis Function Block* Application Manual and Section Alarm and error messages (Page 333) in this manual
16#8061	LPAC_ERR_NO_EXTERNAL_ENCODER; no TO externalEncoder transferred
16#8070	LPAC_ERR_SLIP_COMP_VELOCITY_ZERO; velocity for the superimposed correction motion ≤ 0.0
16#8071	LPAC_ERR_SLIP_COMP_ACC_ZERO; setpoint for acceleration/deceleration of the superimposed positioning motion ≤ 0.0
16#8074	LPAC_ERR_GET_STATE_OF_AXIS_COMMAND; error from the _getStateOfAxisCommand command; return value of the SIMOTION *_getStateOfAxisCommand* function is equivalent to the *_move* function and can be fetched in hex format in the high word from *errorID*
16#8075	LPAC_ERR_REDEFINE_POSITION; error from the _redefinePosition command; return value of the SIMOTION *_redefinePosition* function is equivalent to the *_move* function and can be fetched in hex format in the high word from *errorID*
16#8076	LPAC_ERR_STOP; error from the *_stop* command; return value of the SIMOTION *_stop* function is equivalent to the *_move* function and can be fetched in hex format in the high word from *errorID*
16#8077	LPAC_ERR_ENABLE_EXTERNAL_ENCODER; error from the _enableExternalEncoder command; return value of the SIMOTION *_enableExternalEncoder* function is equivalent to the *_move* function and can be fetched in hex format in the high word from *errorID*
16#8079	LPAC_ERROR_TO_EXTERNAL_ENCODER; error on the external encoder technology object
16#80A5	LPAC_ERR_POS; error from the *_pos* command; return value of the SIMOTION *_pos* function is equivalent to the *_move* function and can be fetched in hex format in the high word from *errorID*

4.5.6 FBLPacPrintmarkCorrection - print mark correction

4.5.6.1 Technological consideration

The current position of an axis can be measured with a measuring input on the basis of an external signal (initiator, light pushbutton) with the **FBLPacPrintmarkCorrection** block. The measured value is compared to a setpoint. If the setpoint deviates from the actual value, this difference is compensated for by superimposing a correction motion on the axis.
The block can be used, for example, to set the sealing/cutting position for tubular bagging machines.

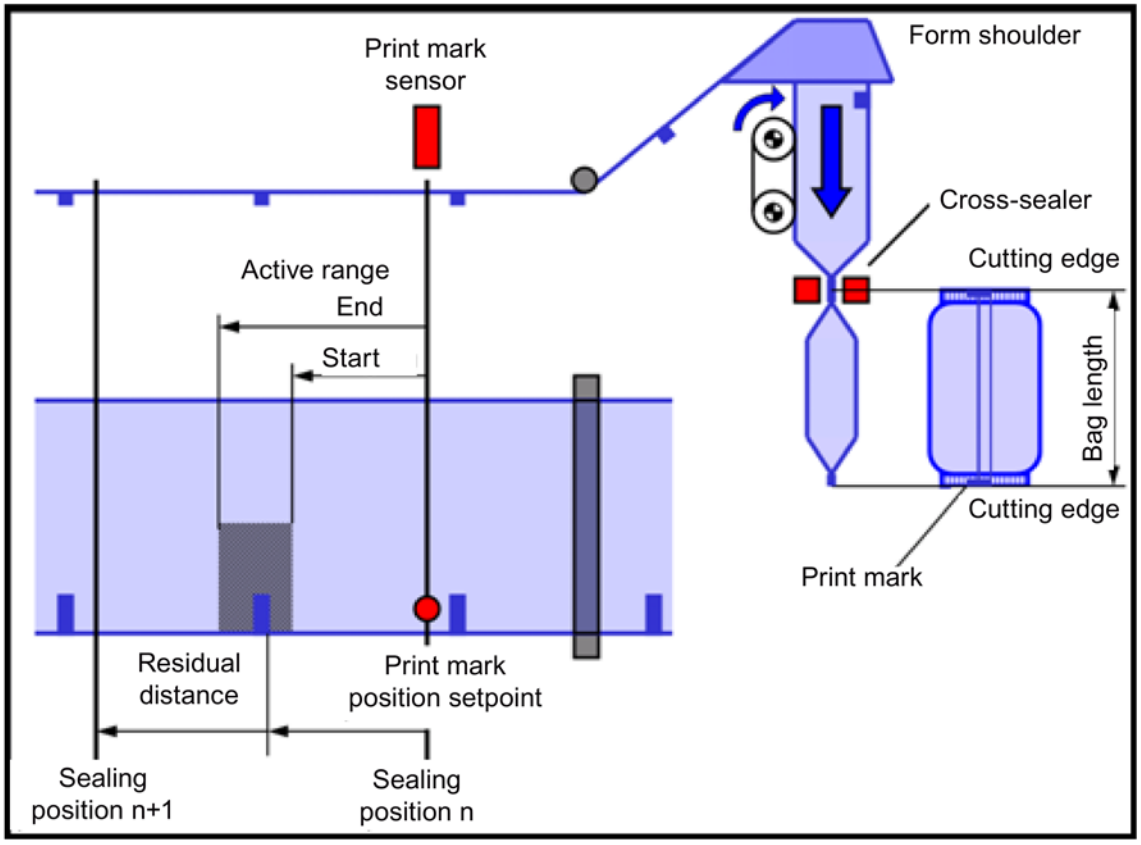


Figure 4-28 Print-mark detection on the foil path of a vertical tubular bagging machine

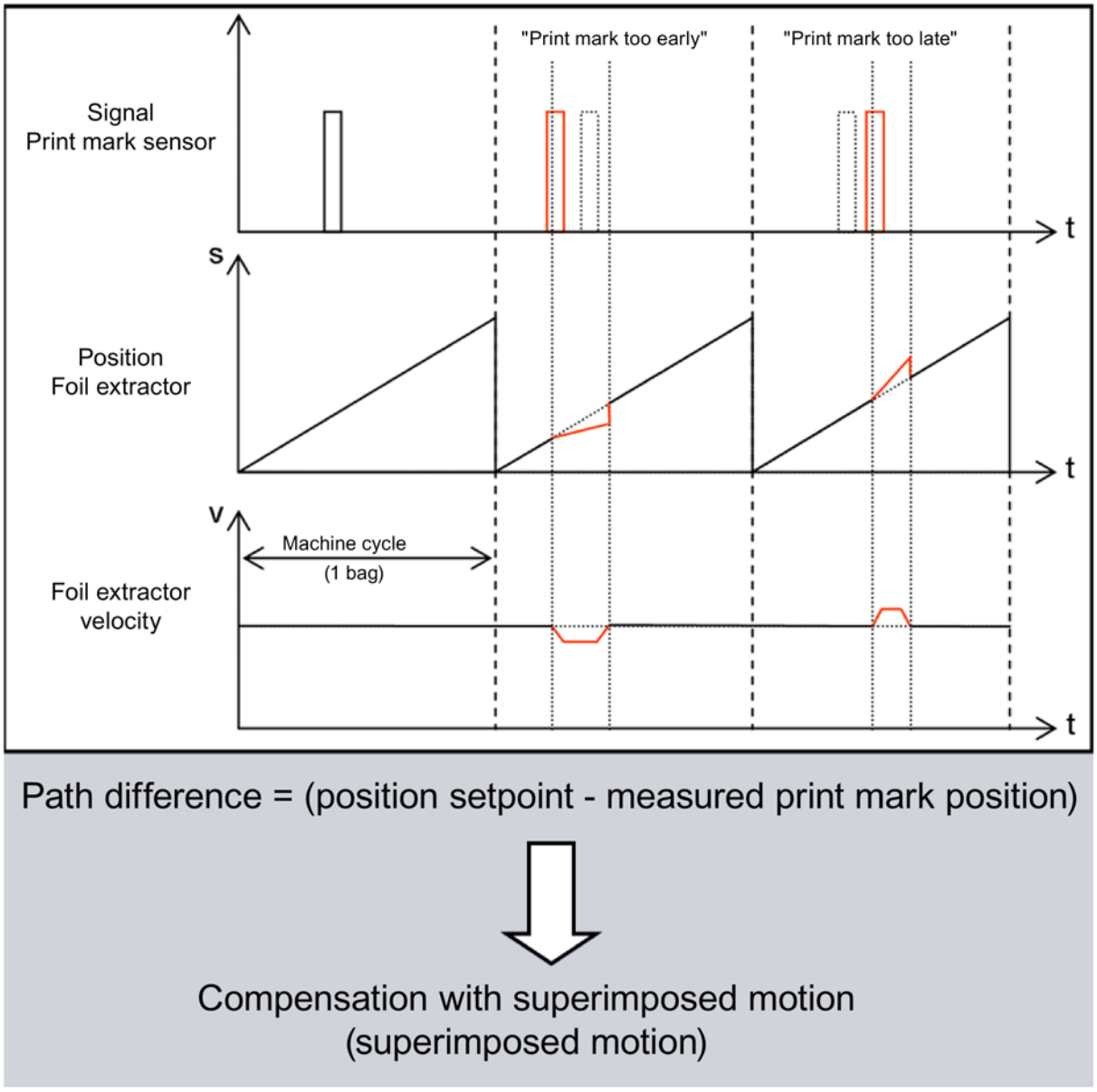


Figure 4-29 Print mark detection and correction movement

Note

Because a superimposed positioning for the print mark correction is used, other superimposed movements, such as slip compensation, which also use the superimposed positioning must be coordinated by the application. Both correction possibilities must **not be activated simultaneously** on an axis! For example, either only the slip compensation (for unprinted foil) or the print mark correction is always activated (for printed foil with print marks) depending on the deployed foil of a tubular bagging machine.

4.5.6.2 Module description

The **FBLPacPrintmarkCorrection** function block offers the main functions and auxiliary functions as well as possible options shown in the following overview.

Function	Description	Implementation
Main function	Print mark correction	Detection and compensation of a position difference between a defined position setpoint (print mark position) and the current axis position, e.g. caused by slip on a foil or paper path, by a superimposed positioning motion on the current motion. Velocity, acceleration and deceleration of the compensation motion can be specified.
Options	Measuring range	The specification of a start and end position defines a measuring range (active range) within which the print mark signal is to be evaluated. This prevents, for example, incorrect triggering caused by other effects (print pattern).
	Limitation of the correction motion	This limits the maximum correction motion that can be performed.
	Suppress correction motion	The correction motion can be suppressed by setting the *disableCorr* input signal. This function can be used, for example, when the foil is in the sealing area.
	Threshold value	This function can specify a minimum value of the deviation below which no correction motion is initiated.
	Edge	This function can select whether a positive or negative signal edge is evaluated as print mark signal.

Detailed description

The function is activated by a rising edge at the *execute* input. A measurement is made in the range specified by *toleranceRangeStart* and *toleranceRangeEnd*. If both values are set the same, the measurement is made without a measuring range. The measurement is initiated by a trigger signal. The *edge* input variable specifies on which edge triggering is performed. When this input is TRUE, the rising edge acts as the trigger signal, and when FALSE, the falling edge acts as the trigger signal. When the trigger condition occurs, the measurement is taken, and the measured value is output via *measuringValue*. The *newMeasuringValue* output is also set. A deviation is determined from the measured value *measuringValue* and the *targetPosition* position setpoint and output by means of the variable *measuringDifference* (difference = setpoint - actual). This value is the basis for the superimposed motion command. The deviation can be manipulated by means of the correction factor *calculationFactor*. The measured difference is multiplied by this value. This value can be inverted (multiplied by -1) to reverse the direction of correction by setting the input *correctionInverter* to TRUE. A superimposed positioning command is started with the last value measured. The correction motion is defined via the *correctionVelocity*, *acceleration* and *deceleration* parameters. When the *correctCommandPos* (default setting = TRUE) bit is set, the superimposed position will be reset once the positioning has been made. If a measurement request, correction motion or the reset of the superimposed position is active, this is indicated by TRUE at the *busy* output. The function block processing can be aborted in any step with a rising edge at the *reset* input. If an error occurs within the block while it is being processed, the output variable error is set, and an error code is output via *errorID*.

Note

If the measurement is made without a range, the input parameters *toleranceRangeStart* and *toleranceRangeEnd* are preset with the value 0.0.

Automatic, dynamic measuring range adjustment

Dynamic measuring range adjustment is necessary because of the dead time that exists between the SIMOTION platform and the drive on which the measurement is taking place. As of version 3.1.1, the system contains configuration data used to automate this dynamic adjustment. The automatic dynamic measuring range adjustment is made by entering a dead time compensation in the configuration data *MeasuringRange.activationTime*. You can find this configuration data by opening the Configuration dialog box under the corresponding measuring input in the SIMOTION SCOUT symbol browser. A time value by which the command to activate the measured value memory on the drive should be moved up can be entered under "Activation time of active range on measuring input".

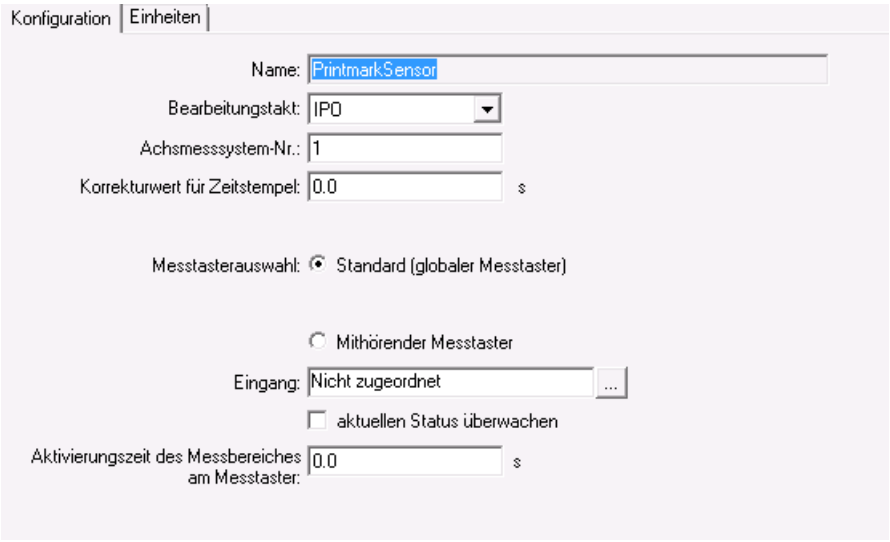


Figure 4-30 Parameter dialog box to enter the activation time of the active range on the measuring input (MeasuringRange.activationTime parameter)

A tool for calculating the time value is provided free of charge with SIMOTION SCOUT on the Utilities/Addition free of charge CD (folder ..\4_TOOLS\MEASURING_INPUT_CALCULATION\). Based on the system clocks and the properties of the measuring input, this tool determines the resulting dead time, the amount of which the measuring range activation should be moved up.

The block supports the measurement of the print mark position and the correction motion on different axes. For this purpose, the block is passed different technology TO references at the inputs *toMeasuringAxis* and *toCorrectionAxis*.

Note

The measuring input must be assigned to the axis on which the measurement is also performed.

Axis transports the print mark

Characteristic for this function is that the correction motion is performed on the same axis on which the measurement is also performed. Namely, the measuring input in SIMOTION SCOUT is configured for the same axis as that passed with the *toCorrectionAxis* input parameter. The TRUE value at the *correctCommandPos* input (corresponds to the default setting) means that after the correction of the axis has been made, its position value will be returned to the superimposed position value. This re-establishes the reference to the transported foil and motor position and the axis once again has its original position value. The *_redefinePosition* system function corrects the axis position.

Sequence of the correction with reset of the superimposed position value:

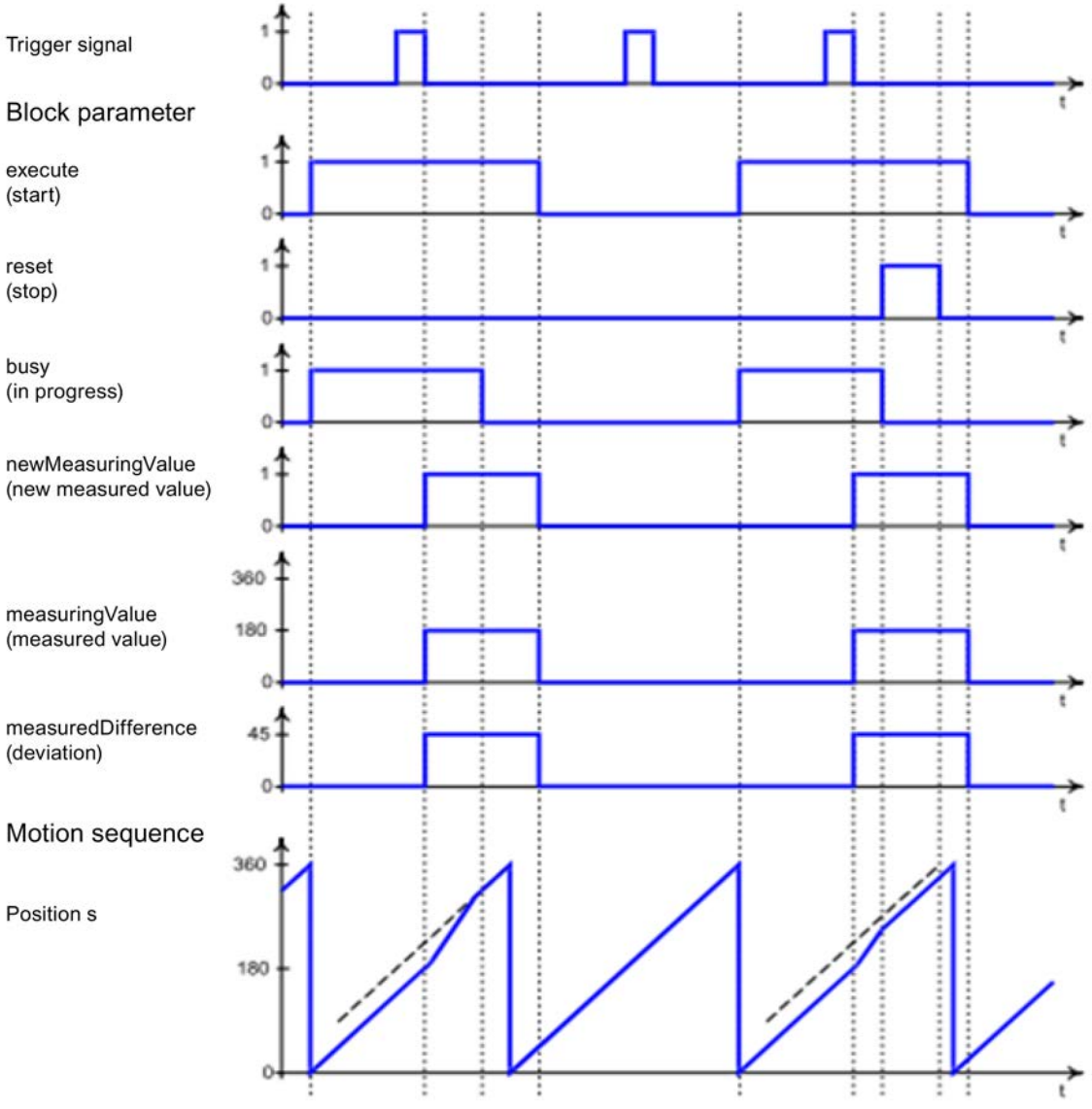


Figure 4-31 Signal sequence diagram with the "Axis transports the print mark" function

Axis does not transport the print mark

The "Axis does not transport the print mark" function ensures that the measurement and the correction cannot be performed on the same axis. The print mark position is measured at the axis passed to the block at the *toMeasuringAxis* input. The correction is performed at the axis passed to the block at the toCorrectionAxis input. The measuring input must be assigned to the axis on which the measurement is also performed. Also here, the TRUE value at the *correctCommandPos* input (corresponds to the default setting) means that after the correction of the axis has been made, its position value will be returned to the superimposed position value. This means that the axis has its original position value once again. The *_redefinePosition* system function corrects the axis position.

The following figure shows a typical signal characteristic. The associated motion sequence is shown in addition to the input and output signals.

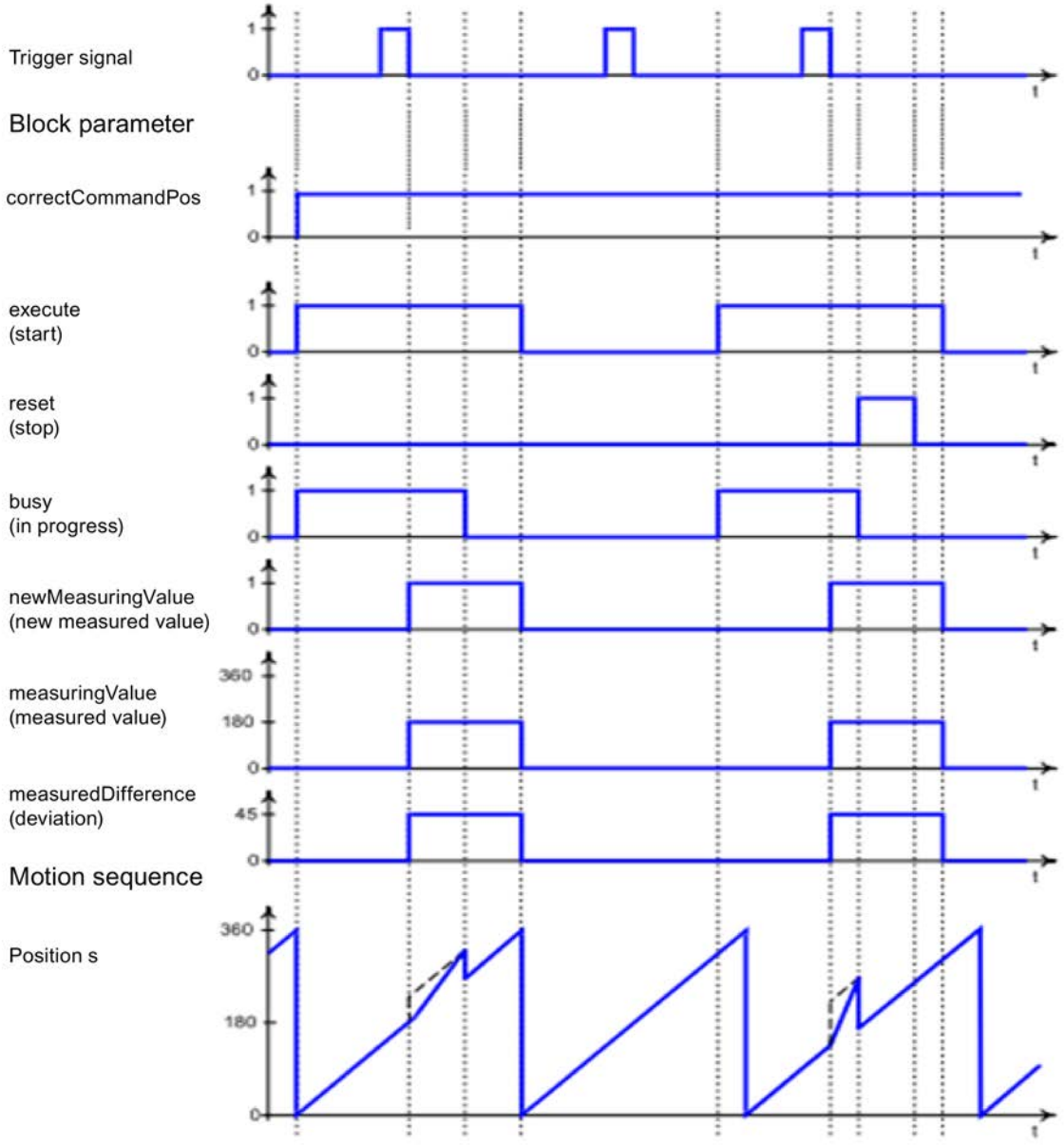


Figure 4-32 Signal sequence diagram without the "Axis transports the print mark" function

Schematic representation in LAD/FBD

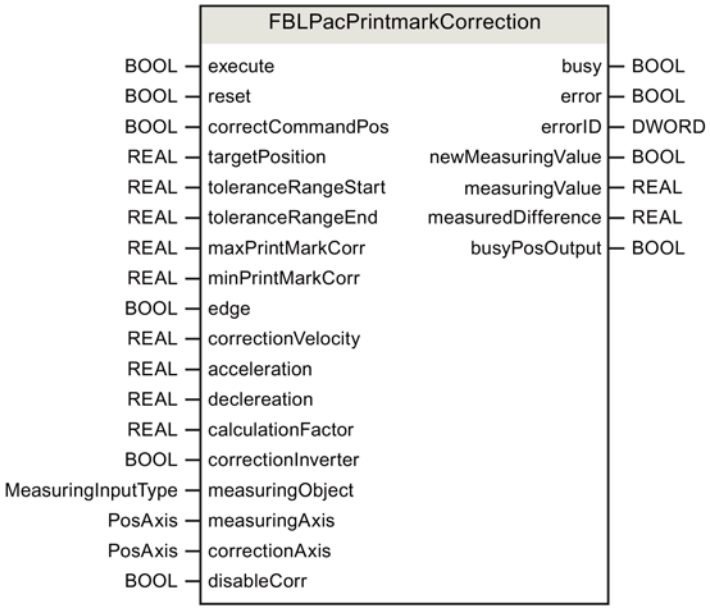


Figure 4-33 FBLPacPrintmarkCorrection - schematic block view in LAD/FBD

Input and output parameters

Table 4- 118 Input and output parameters

Element	P type[1)]	Data type	Initial value	Meaning
execute	IN	BOOL	FALSE	TRUE: Start print mark correction
reset	IN	BOOL	FALSE	Reset the block
correctCommandPos	IN	BOOL	TRUE	Reset superimposed position after correction (redefinePos)
targetPosition	IN	REAL	0.0	Position setpoint of the print mark
toleranceRangeStart	IN	REAL	0.0	Print mark active range start position (measuring range start)
toleranceRangeEnd	IN	REAL	0.0	Print mark active range end position (measuring range end)
maxPrintMarkCorr	IN	REAL	0.0	Maximum correction path
minPrintMarkCorr	IN	REAL	0.0	Minimum deviation of the print mark for which a correction motion is performed
edge	IN	BOOL	TRUE	Evaluate the positive edge of the print mark
correctionVelocity	IN	REAL	10.0	Velocity of the correction motion
acceleration	IN	REAL	1000.0	Acceleration of the correction motion
deceleration	IN	REAL	1000.0	Deceleration of the correction motion
calculationFactor	IN	REAL	1.0	Compression factor for measured print mark deviation
correctionInverter	IN	BOOL	FALSE	Invert the measured print mark deviation

Element	P type[1)]	Data type	Initial value	Meaning
measuringObject	IN	MeasuringInputType	TO#NIL	TO measuringInput for detecting the print mark
measuringAxis	IN	PosAxis	TO#NIL	TO axis on which measuring is performed with the measuring input
correctionAxis	IN	PosAxis	TO#NIL	TO axis of the axis to be corrected
disableCorr	IN	BOOL	FALSE	Suppress correction motion
busy	OUT	BOOL	FALSE	Selected functionality being processed
error	OUT	BOOL	FALSE	Error occurred
errorID	OUT	DWORD	16#0000_0000	Error code
newMeasuringValue	OUT	BOOL	FALSE	New measured value of the measuring input is pending
measuringValue	OUT	REAL	0.0	Measured value of the measuring input
measuredDifference	OUT	REAL	0.0	Determined deviation of the print mark to the position setpoint
busyPosOutput	OUT	BOOL	FALSE	Correction motion active

1) Parameter types: IN = input parameter, OUT = output parameter

4.5.6.3 Integration of the module

The **FBLPacPrintmarkCorrection** block is called in the **LPac** blocks for the **FBLPacFoilFeedPos**, **FBLPacFoilFeedGear** and **FBLPacFoilFeedCam** foil feed. This block is supplied and controlled via the interface. Consequently, a separate call is not required when these blocks are used. Within the LPac **FBLPacFoilFeedGear** and **FBLPacFoilFeedCam** blocks, the print mark block is parameterized so that the print mark position is measured on the virtual master axis of the machine and any required correction motion performed on the foil feed axis. Consequently, the measuring input must be assigned to the master axis.

In other applications for which a position control and correction of the motion should be performed for an external event (trigger), an isolated use of the block is also possible.

The following supplementary conditions must be observed:

- For an emergency stop of the reference axis, ensure that the superimposed motion is also stopped.
- The block should be called only in cyclic tasks.

Technology objects, basic parameterization and block call

Table 4- 119 TO type and basic parameterizations

Axis / technology object	
TO type	At least position axis type
Measuring input type	Local or global measuring input
Block call (for isolated application)	Cyclic call, e.g. IPO, IPO2 task

Further blocks and libraries

The **FBLPacPrintmarkCorrection** block does not use any further blocks internally.

Error messages

Error number [HEX]	Meaning
16#0000	LPAC_ERR_NO; error-free
16#8065	LPAC_ERR_ENABLE_MEASURING_INPUT; error when enabling the measuring input; return value of the SIMOTION *_enableMeasuringInput* function is equivalent to the *_move* function and can be fetched in hex format in the high word from *errorID*
16#8069	LPAC_ERR_MEASURING_ERROR; error at the measuring input TO
16#8073	LPAC_ERR_DISABLE_MEASUREING_INPUT; error from the *_disableMeasrureingInput* command; return value of the SIMOTION *_disableMeasuringInput* function is equivalent to the *_move* function and can be fetched in hex format in the high word from *errorID*
16#8074	LPAC_ERR_GET_STATE_OF_AXIS_COMMAND; error from the *_getStateOfAxisCommand* command; return value of the SIMOTION *_getStateOfAxisCommand* function is equivalent to the *_move* function and can be fetched in hex format in the high word from *errorID*
16#8075	LPAC_ERR_REDEFINE_POSITION; error from the *_redefinePosition* command; return value of the SIMOTION *_redefinePosition* function is equivalent to the *_move* function and can be fetched in hex format in the high word from *errorID*
16#8076	LPAC_ERR_STOP; error from the *_stop* command; return value of the SIMOTION *_stop* function is equivalent to the *_move* function and can be fetched in hex format in the high word from *errorID*
16#80A5	LPAC_ERR_POS; error from the *_pos* command; return value of the SIMOTION *_pos* function is equivalent to the *_move* function and can be fetched in hex format in the high word from *errorID*

4.5.7 FBLPacPrelimFoilFeed - foil pre-extractor

4.5.7.1 Technological consideration

The **FBLPacPrelimFoilFeed** block controls a speed-controlled foil unwinding drive depending on the foil feed speed (so-called preliminary foil feed). The foil feed speed is controlled by the actual foil feed based on the production speed of the machine. It serves as base setpoint for the speed specification to the preliminary foil feed. Core unwinder and circumferential unwinder are available as mechanical version. The block should preferably be used for circumferential unwinders because a diameter calculator is not implemented. If the block is used for core winders, the current foil roller diameter can be measured continuously via a diameter measuring system and transferred to the block via the *sConfigData.r32Diameter* parameter. Alternatively, an average foil roller diameter can be transferred to the block.

To compensate for different foil tensions, the evaluation of a dancer position is possible. Whereby the dancer position can be specified continuously as an analog position value and also in discrete form via two binary limit switches.

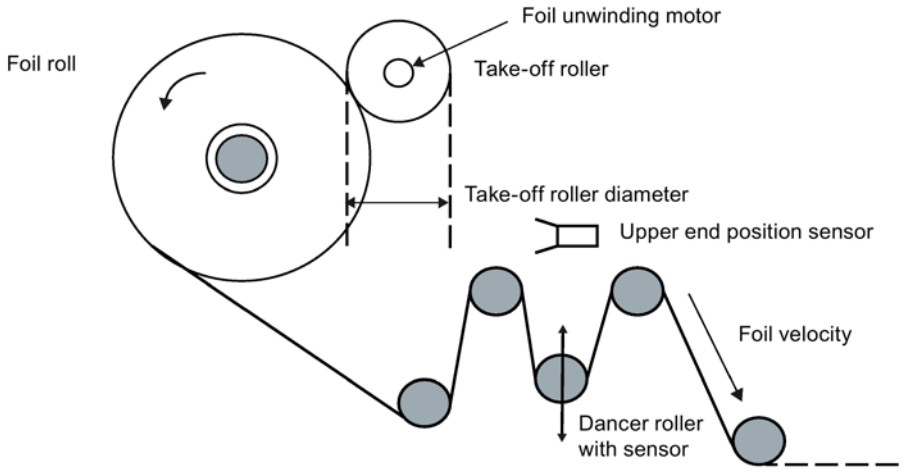


Figure 4-34 Foil unwinder with circumferential unwinder and dancer

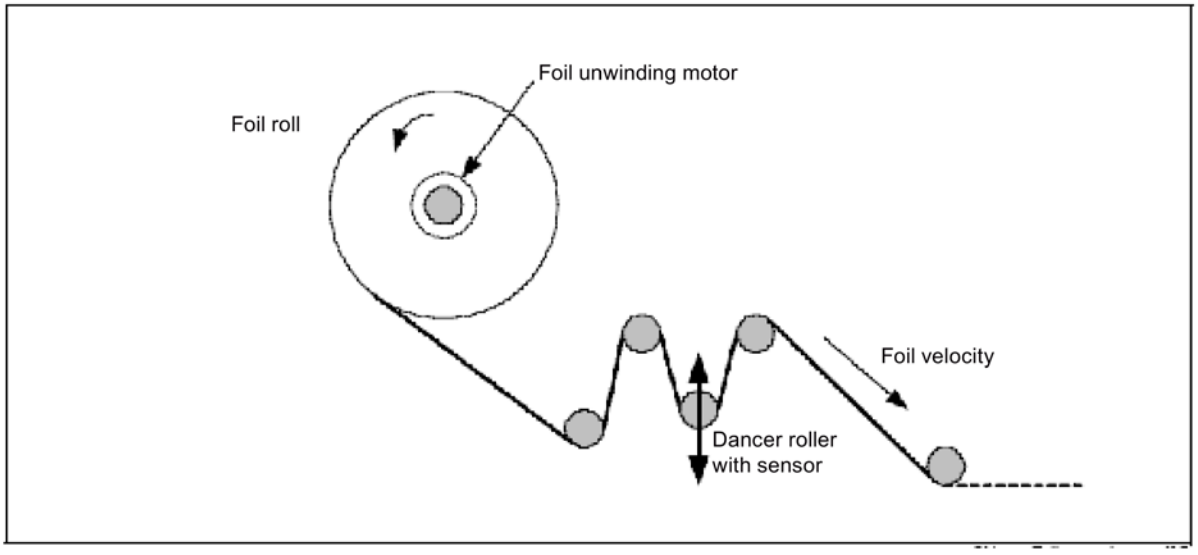


Figure 4-35 Foil unwinder with core drive and dancer

Dancer control

A description of the dancer control integrated in the block can be found in the FBLPacDancerControl (Page 189) block.

4.5.7.2 Module description

Scope of functions

The **FBLPacPrelimFoilFeed** function block offers the main functions and auxiliary functions as well as possible options shown in the following overview.

Function	Description	Implementation
Main function	Velocity-controlled operation	The level-triggered *enableMove* input starts a velocity-controlled traversing of the axis. The foil feed velocity [mm/s] applied at the *foilSpeed* input serves as base velocity. The block converts the foil velocity [mm/s] based on a specified drive roller diameter automatically to the required speed [rpm]. When an analog-controlled external drive axis is used, the rated motor speed, the scaling for the analog setpoint, the reference speed and the gear ratio for any existing load gear must also be specified.
Options	Dancer control	A superimposed dancer control can be activated via the level-triggered *enableDancerControl* input. Two different methods of operation are available: • The tension control can be used via a dancer position control. The dancer position must be applied as analog actual value at the *dancerPos* input and the appropriate control parameters parameterized for the internal PID controller. • Alternatively, a simple control via the acquisition of two limit switches can be used. If the respective limit switch is approached, the unwinding speed is increased/reduced according to a parameterizable time and size in steps until the limit switch becomes free again.
Auxiliary functions	Enable axis	The axis is enabled via the level-triggered *enableAxis* input.
	Jog axis	Positive or negative jogging can be started via the level-triggered *jogForward* or *jogBackward* inputs.
	Axis quick stop	A quick stop can be performed or a motion of the axis prevented via the level-triggered *enableEStop* input.

Schematic representation in LAD/FBD

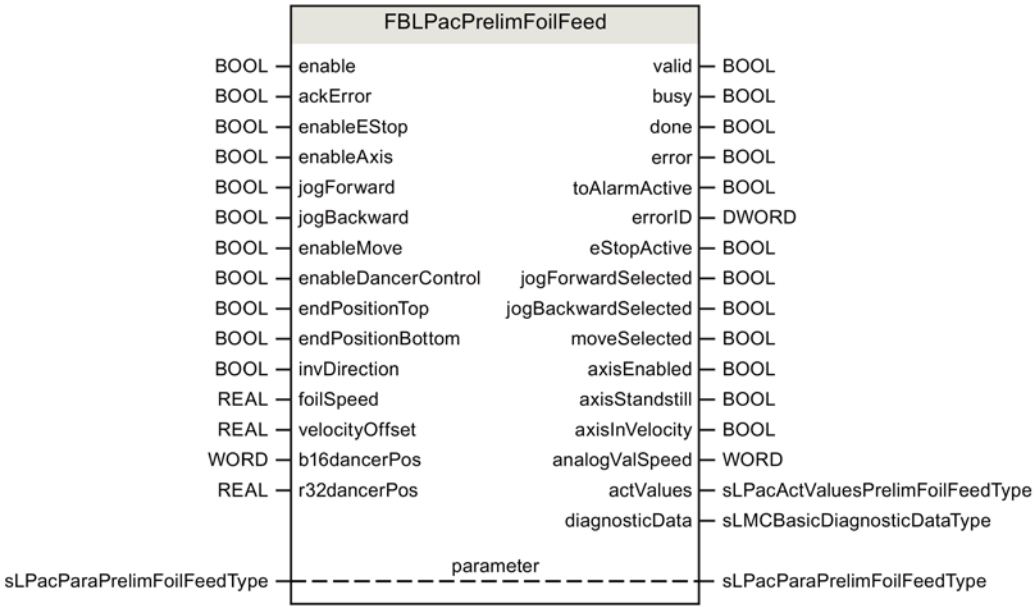


Figure 4-36 FBLPacPrelimFoilFeed - schematic block view in LAD/FBD

Input and output parameters

Table 4- 120 Input and output parameters

Element	P type [1)]	Data type	Initial value	Meaning
enable	IN	BOOL	FALSE	Activate FB
ackError	IN	BOOL	FALSE	Acknowledge error
enableEStop	IN	BOOL	FALSE	Activate quick stop
enableAxis	IN	BOOL	FALSE	Enable axis
jogForward	IN	BOOL	FALSE	Continuous jogging in the positive direction
jogBackward	IN	BOOL	FALSE	Continuous jogging in the negative direction
enableMove	IN	BOOL	FALSE	Start speed specification The axis follows the foil velocity continuously. The block calculates the required speed based on the diameter of the take-off roller.
velocityOffset	IN	REAL	1.0	Velocity offset to the setting of the foil tension
enable-DancerControl	IN	BOOL	FALSE	Activate dancer control
endPositionTop	IN	BOOL	FALSE	Top end position reached

Element	P type [1]	Data type	Initial value	Meaning
endPosition Bottom	IN	BOOL	FALSE	Bottom end position reached
invDirection	IN	BOOL	FALSE	Direction of rotation reversal
foilSpeed	IN	REAL	0.0	Speed setpoint in mm/s
velocityOffset	IN	REAL	0.0	Velocity offset to the setting of the foil tension [%]
b16dancerPos	IN	WORD	16#0000	Dancer position scaled by the analog input (Word data type, value range 16#6C00 = 27648 = 100% … 16#9400 = -27648 = - 100%)
r32dancerPos	IN	REAL	0.0	Dancer position scaled by the analog input (Real data type, value range +100.0% …-100.0%)
valid	OUT	BOOL	FALSE	Block is active
busy	OUT	BOOL	FALSE	Selected functionality being processed
done	OUT	BOOL	FALSE	Selected functionality completed successfully
error	OUT	BOOL	FALSE	Error occurred
toAlarmActive	OUT	BOOL	FALSE	TO alarm occurred
errorID	OUT	DWORD	16#0000 _0000	Error code
eStopActive	OUT	BOOL	FALSE	Quick stop active
jogForward Selected	OUT	BOOL	FALSE	Jogging positive direction selected
jogBackward-Selected	OUT	BOOL	FALSE	Jogging negative direction selected
moveSelected	OUT	BOOL	FALSE	Speed specification active
axisEnabled	OUT	BOOL	FALSE	Axis enabled
axisStandStill	OUT	BOOL	FALSE	Axis at standstill (is not output for an external analog axis)
axisInVelocity	OUT	BOOL	FALSE	Axis has reached target velocity (is not output for an external analog axis)
analogValSpeed	OUT	WORD	16#0000	External speed setpoint scaled (if no TO specified)
actValues	OUT	sLPacActValues Prelim-FoilFeedType	-	Current preliminary foil feed speed in rpm (is not output for an external analog axis)
diagnostics	OUT	sLMCBasic DiagnosticType	-	Diagnostics structure (is not output for an external analog axis), see Alarm and error messages (Page 333)
parameter	IN_OUT	sLPacParaPrelim FoilFeedType	-	Parameter for preliminary foil feed

1) Parameter types: IN = input parameter, OUT = output parameter, IN_OUT = in-out parameter

Note

The status messages and return values, such as *axisStandstill*, *axisInVelocity*, etc., are relevant only when a SIMOTION TO axis is used. These signals are not issued for an analog-controlled external axis! Only the velocity specification function (*enableMove*) is available for an analog axis. All required ramps and setpoints for the acceleration and deceleration must be implemented in the external frequency converter!

Table 4- 121 sLPacActValuesPrelimFoilFeedType structure

Element	Data type	Initial value	Meaning
r32ActualSpeed	REAL	0.0	Current motor speed [rpm]; (is not output for an external analog axis)
r32ActualVelocity	REAL	0.0	Current foil feed velocity [mm/s]; (is not output for an external analog axis)

Table 4- 122 sLPacParaPrelimFoilFeedType structure

Element	Data type	Meaning
sTOs	sLPacToDriveDatasType	TO for preliminary foil feed
sMotionData	sLPacPrelimFoilFeedMotionDataType	Parameters for EStop, jog, move
sConfigData	sLPacPrelimFoilFeedConfigDataType	Mechanical configuration data (take-off roller diameter, etc.)

Table 4- 123 sLPacToDriveDatasType structure

Element	Data type	Initial value	Meaning
toAxis	driveAxis	TO#NIL	Drive axis, for TO#Nil control of the axis via analog output

Table 4- 124 sLPacPrelimFoilFeedMotionDataType structure

Element	Data type	Meaning
sEStopData	sLMCBasicEStopDataType	Parameter for quick stop
sJogData	sLPacJogDataType	Parameter for jogging
sMoveData	sLPacMoveDataType	Parameter for speed specification

Quick stop parameters - sEStopData structure

See Axis quick stop (Page 43)

Jog parameters - sJogData structure

See Jog axis (Page 36)

Different default settings that cannot be changed by the user:

The jog mode is preset internally to "incremental jog" in the block.

Velocity specification parameters - sLPacMoveData structure

Table 4- 125 sLPacMoveDataType structure

Element	Data type	Initial value	Meaning
r32Velocity	REAL	-1.0	Velocity setpoint 1) Not evaluated, velocity setpoint comes from the *foilSpeed* input
r32Acceleration	REAL	-1.0	Acceleration setpoint min^{-2} 1)
r32Deceleration	REAL	-1.0	Deceleration setpoint min^{-2} 1)
r32Jerk	REAL	-1.0	Jerk setpoint 1)

1) With value -1, the respective parameter value is taken from the default on the TO

Note

The acceleration and deceleration values in the *sLPacMoveData* structure must be set at least as high as for the leading axis or for the machine master. Otherwise, the preliminary foil feed cannot follow the foil velocity during acceleration/braking and foil overstretching or tearing can occur.

Note

The acceleration and deceleration values in the *sLPacMoveData* structure are relevant only when a SIMOTION technology axis is used. If an analog-controlled external drive axis is used, the required ramp times must be implemented and set in the external frequency converter. The analog setpoint calculated by the block is not maintained by a ramp ngenerator. The previous note applies similarly to these externally set ramps.

Table 4- 126 sLPacPrelimFoilFeedConfigDataType structure

Element	Data type	Initial value	Meaning
r32Diameter	REAL	0.0	Diameter of the preliminary foil feed in mm (take-off roller)
i16SetSpeedScaling	INT	3.000	Analog scaling value for speed setpoint, i.e. motor speed for 10 V analog voltage setpoint (only for external analog axis)
i16SetAnalogValue Scaling	INT	27648	Scaling of the analog output, e.g. 27648 d = 10 V = 3000 rpm
u32LoadGearBoxNu mFactor	UDINT	1	Load gear, number of motor revolutions (only for external analog axis)

Element	Data type	Initial value	Meaning
u32LoadGearBoxDenFactor	UDINT	1	Load gear, number of load revolutions (only for external analog axis)
sDancerData	sLPacDanc-Danc-erDataType	-	Configuration data for dancer control

sLPacDancerDataType structure

See FBLPacDancerControl block (Page 189)

4.5.7.3 Integration of the module

Technology objects, basic parameterization and block call

Table 4- 127 TO type and basic parameterizations

Axis / technology object	
TO type	Drive axis or external axis
Version	Not relevant
Modulo	Not relevant
Block call	BackgroundTask

Further blocks and libraries

The **FBLPacPrelimFoilFeed** block uses the following blocks internally.

Block/library	Function
FBLMCBasicAxis / LMCBasic	Axis function block
FBLPacDancerControl/LPac	Simple dancer control

Some of the internally used blocks originate from other libraries. See Section Additional required libraries (Page 15).

Error messages

Error number [HEX]	Meaning
16#0000	LPAC_ERR_NO; error-free
16#8000	ERR_INVALID_STEP; internal error, please get in touch with your Siemens contact partner

Error number [HEX]	Meaning
16#8003	LPAC_ERR_TO_ALARM_ACTIVE; TO alarm active; error message from axis function block Error description, see *SIMOTION Axis Function Block* Application Manual and Section Alarm and error messages (Page 333) in this manual
16#8011	LPAC_ERR_AXIS_FUNCTION_BLOC error message from axis function block For details, see *diagnosticData.b32ErrorCode* output structure; Error description, see *SIMOTION Axis Function Block* Application Manual
16#8080	LPAC_ERR_SET_ANALOG_SCALING_ZERO; analog output scaling factor ≤ 0
16#8081	LPAC_ERR_DIAMETER_ZERO; take-off roller diameter ≤ 0.0
16#8082	LPAC_ERR_LOWER_LIMIT_TIMEOUT; dancer lower end position monitoring time expired; source FBLPacDancerControl
16#8083	LPAC_ERR_UPPER_LIMIT_TIMEOUT; dancer upper end position monitoring time expired; source FBLPacDancerControl
16#8084	LPAC_ERR_PID_CTRL_PRELIM_FOIL_FEED; integrated ctrlpid block error message; source FBLPacDancerControl
16#8085	LPAC_ERR_SET_SPEED_SCALING_ZERO; rated motor speed ≤ 0
16#8086	LPAC_ERR_OVERFLOW_ANALOG_SPEED; analog setpoint overflow
16#8087	LPAC_ERR_UNDERFLOW_ANALOG_SPEED; analog setpoint underflow
16#8088	LPAC_ERR_LOAD_GEAR_NUM_FACT_ZERO; load gear, number of motor revolutions ≤ 0
16#8089	LPAC_ERR_LOAD_GEAR_DEN_FACT_ZERO; load gear, number of load revolutions ≤ 0
16#808A	LPAC_ERR_UPPER_SPEED_LIMIT_REACHED; upper speed limit (200%) reached; source FBLPacDancerControl
16#808B	LPAC_ERR_LOWER_SPEED_LIMIT_REACHED; lower speed limit (0%) reached; source FBLPacDancerControl

Note

The **pPrelimFoilFeed** program unit is available in the LPac project to illustrate the use of the **FBLPacPrelimFoilFeed** block.

This unit contains the programs for the required block calls in the StartupTask and the BackgroundTask.

4.5.8 FBLPacDancerControl

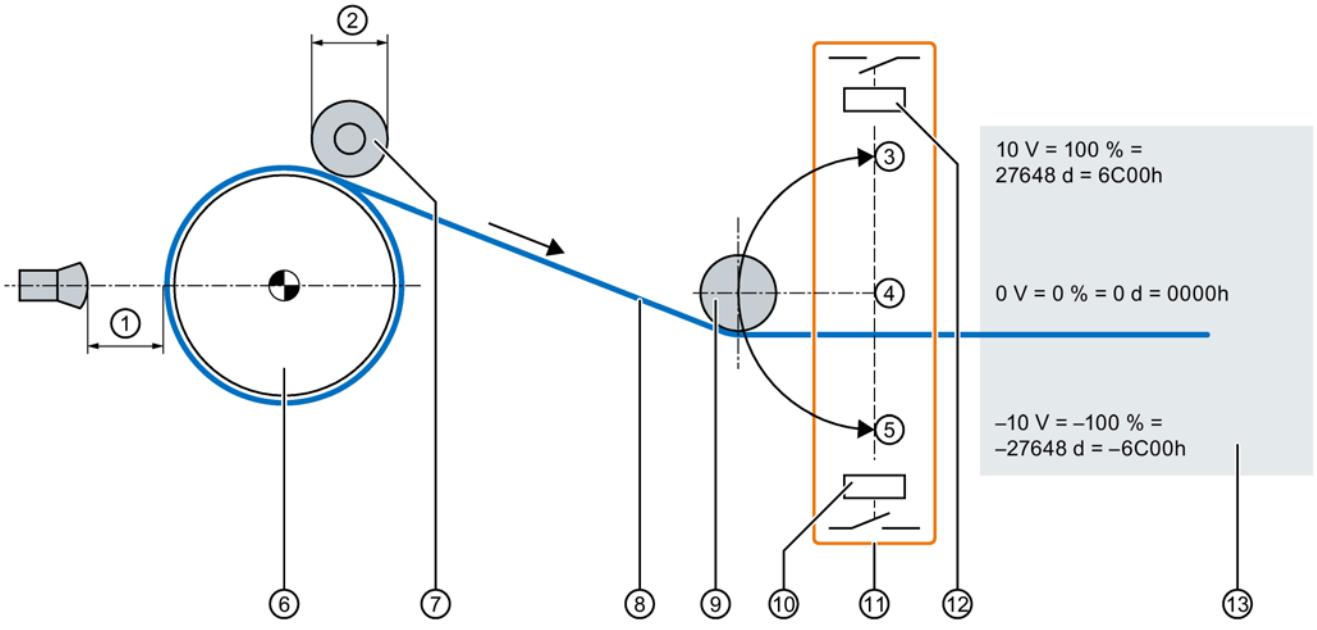
4.5.8.1 Technological consideration

The **FBLPacDancerControl** block implements a simple dancer position control, for example, in conjunction with the **FBLPacPrelimFoilFeed** block, a simple tension control on a speed-controlled preliminary foil feed unit. The block provides a dancer control with integrated SIMOTION PID controller block and analog position measurement, e.g. via a potentiometer.

Furthermore, a simple "discrete" position control can be implemented by sensing the dancer end positions via proximity switches, for instance.

As part of the **LPac**, the block is called and used in the **FBLPacPrelimFoilFeed** block. An isolated use is also possible.

The following figure below shows the foil unwinder with dancer roll:



① Diameter measurement for core winders
② Take-off roller diameter
③ Top end position
④ Center position
⑤ Bottom end position
⑥ Unwinder
⑦ Take-off roller
⑧ Foil path
⑨ Dancer roll
⑩ Lower limit switch
⑪ Discrete position sensing
⑫ Upper limit switch
⑬ Analog position sensing

4.5.8.2 Module description

Scope of functions

The **FBLPacDancerControl** function block offers the main functions and auxiliary functions as well as possible options shown in the following overview.

Function	Description	Implementation
Main function	Analog dancer control	A dancer control can be activated via the level-triggered *enableDancerControl* input. The dancer position must be applied as analog actual value at one of the following two inputs and the appropriate control parameters parameterized for the internal PID controller: • *b16dancerPos* (Word data type, value range 16#6C00 = 27648 = 100% … 16#9400 = -27648 = -100%) • *r32dancerPos* (Real data type, value range +100.0% … -100.0%) To influence the unwinder speed, an *analogSpeedFactor* setpoint factor is output in the range 0.0 … 2.0%. At the dancer center position, the output factor is 1.0 = 100%.
Option	Discrete dancer control	As an alternative, a simple control via the acquisition of two limit switches is available. If the respective limit switch is approached, the unwinding speed is increased/reduced according to a parameterizable time and size in steps until the limit switch becomes free again. The staircase function increases or decreases the velocity in percent after expiration of a parameterizable time within which the dancer was not able to leave the respective limit switch. When the respective end position is exited again, the preliminary foil feed traverses again with the machine velocity. If the associated end position is not exited again within a parameterizable maximum time, an error will be issued. The percentage value for the staircase function to be increased applies to the associated scaling velocity of the used output. The number of velocity changes within the stepcase function is parameterized with the LPAC_MAX_NUMBER_OF_VELOCITY_CHANGES constant of the **cPublic** library unit.

Schematic block view in LAD/FBD

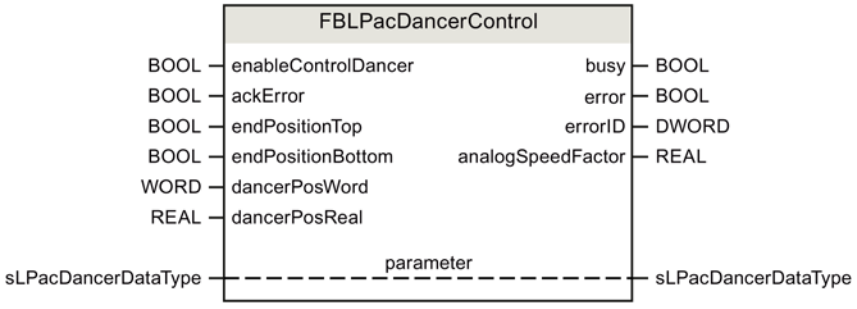


Figure 4-37 FBLPacDancerControl - schematic block view in LAD/FBD

Table 4- 128 Input and output parameters

Element	P type [1]	Data type	Initial value	Meaning
enable-DancerControl	IN	BOOL	FALSE	Activate dancer control
ackError	IN	BOOL	FALSE	Acknowledge error
endPosition-Top	IN	BOOL	FALSE	Dancer top end position reached
endPosition Bottom	IN	BOOL	FALSE	Dancer bottom end position reached
Worddancer Pos	IN	WORD	16#0000	Dancer position scaled by the analog input (Word data type, value range 16#6C00 = 27648 = 100% … 16#9400 = -27648 = -100%)
realdancerPos	IN	REAL	0.0	Dancer position scaled by the analog input (Real data type, value range +100.0% …-100.0%)
busy	OUT	BOOL	FALSE	Selected functionality being processed
error	OUT	BOOL	FALSE	Error occurred
errorID	OUT	DWORD	16#0000 _0000	Error code
analogSpeed-Factor	OUT	REAL	1.0	Severity factor of the dancer control; value range 0.0 … 2.0
parameter	IN_ OUT	sLPac DancerDataType	-	Parameter structure for PID variant of the dancer control

[1] Parameter types: IN = input parameter, OUT = output parameter, IN_OUT = in-out parameter

Table 4- 129 sLPacDancerDataType structure

Element	Data type	Initial value	Meaning
boDancerViaPos	BOOL	FALSE	TRUE: Dancer control via position (analog potentiometer) FALSE: Dancer control via Top/Bottom inputs
r32WatchTimeLimit	Real	0.0	Maximum time monitoring for the top or bottom limit in [ms]
r32VelocityStep	REAL	0.0	Value for the percentage change of the velocity when a threshold switch or "staircase step" is reached [%], 0.01 = 1% (parameter only relevant for activation of the "discrete" dancer control via the *endPositionTop* and *endPositionBottom* limit switches)
r32ChangeTime	REAL	0.0	Time for execution of the staircase function for changing the velocity in [ms] (parameter only relevant for activation of the "discrete" dancer control via the *endPositionTop* and *endPositionBottom* limit switches)
sPid_Parameter	Struct_CTRL__dataPid		Data structure for parameterization of the internal PID controller block (only relevant for activation of the analog dancer control *boDancerViaPos* = TRUE)

Note

When the "discrete" variant is used (monitoring of the dancer end position through limit switches), make sure that the limit switches remain activated at "overtravel". The limit switches must not be enabled again after overtravel in the direction of the end positions. Otherwise the detection does not funtion.

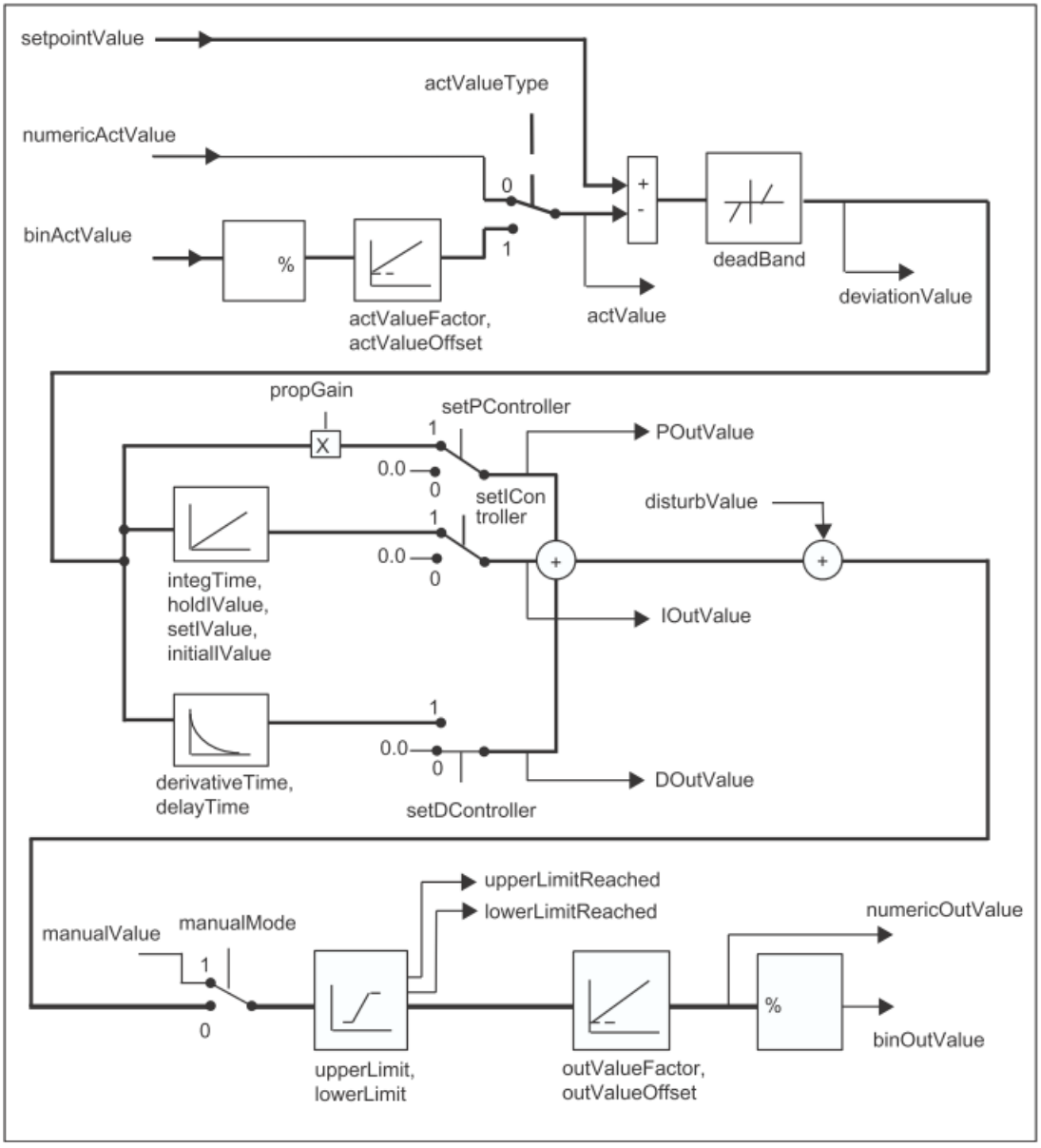


Figure 4-38 Block diagram of the PID controller CTRL_pid (only relevant for analog dancer control using the PID controller)

Table 4- 130 Struct_CTRL_dataPid structure

Element	Data type	Initial value	Meaning
reset 1)	BOOL	FALSE	Initialize _Ctrl_pid block If "reset" is set, all outputs of the lower-level PID block are set to FALSE or Null and the integrator set to the "initialIValue" value.
manualMode 1)	BOOL	FALSE	Activate manual mode If "manualMode" is set, the control loop is interrupted. The value at "ManualValue" is specified as manipulated variable.
actValueType 1)	BOOL	TRUE	Activate actual I/O value If the actual value should be read from the I/O, "binActValue" must be interconnected with the I/O and "boactValueType" set: The parameter can be parameterized by the user in order to select the source of the analog dancer position.
setPController 1)	BOOL	TRUE	Activate P component The PID components can be activated and deactivated individually in the PID algorithm. The P component is activated when "setPController" is set.
setIController 1)	BOOL	TRUE	Activate I component The PID components can be activated and deactivated individually in the PID algorithm. The I component is activated when "setIController" is set.
holdIValue 1)	BOOL	FALSE	Freeze I component The integrator output can be frozen. This requires that "holdIValue" is set.
setIValue 1)	BOOL	FALSE	Set I component The integrator output can be set to the "initialIValue" input. This requires that "setIValue" is set.
initialIValue 1)	REAL	0.0	Initialization value for I component The integrator output can be set to the "initialIValue" initialization value with the "setIValue" input.
setDController 1)	BOOL	FALSE	Activate D component. The PID components can be activated and deactivated individually in the PID algorithm. The D component is activated when "setDController" is set.
cycleTime 1)	DINT	0	Sampling time of the controller block [ms] The time between block calls must be constant. "cycleTime" specifies the sampling time between two block calls in ms. The call interval is measured automatically by the block and entered in the variable.
setpointValue 1)	REAL	0.0	Setpoint: "setPointValue" specifies the setpoint. The setpoint must be specified in the same unit as the actual value (-100.0 … +100.0 (%))

Element	Data type	Initial value	Meaning
numericAc-tValue [1]	REAL	0.0	Actual value: "numericActValue" specifies the actual dancer position. The actual value must be specified in the same unit as the setpoint (-100.0 … +100.0 (%)) The parameter is interconnected to the *r32dancerPos* block input. Value range +100.0 … -100.0%
binActValue [1]	WORD	16#0000	Actual I/O value As an alternative to the interconnection of the actual dancer position to "numericActValue", the actual value can be inter-connected to "binActValue" in I/O format. The parameter is interconnected to the *b16dancerPos* block input. Value range 27648 d = 6C00 h = +100.0 … -27648 d = 9400 h = -100.0%.
manualValue [1]	REAL	0.0	Manual manipulated variable "manualValue" specifies the input of a manual manipulated variable value that can be output by setting "manualMode".
propGain [1]	REAL	2.0	The proportional gain specifies the controller gain.
integTime [1]	DINT	20.000	The integration time [ms] specifies the runtime behavior of the integrator.
derivativeTime [1]	DINT	10.000	Differentiation time [ms] "i32derivativeTime" specifies the time response of the differen-tiator. The value must be ≥ i32cycleTime.
delayTime [1]	DINT	2.000	Delay time of the D component in ms. The algorithm of the D component includes a delay that can be parameterized at "i32delayTime". The value must be ≥ cy-cleTime/2
deadBand [1]	REAL	0.0	Dead band width The control deviation is routed over a dead band. "deadBand" specifies the size of the dead band. The value must be speci-fied in the same unit as the setpoint (-100.0…+100.0%)
upperLimit [1]	REAL	100.0	Upper limit of the manipulated variable The manipulated variable is always limited to an upper limit and a lower limit. "upperLimit" specifies the upper limit. The value must be specified in the same unit as the manipulated variable (-100.0 … +100.0%)
lowerLimit [1]	REAL	0.0	Lower limit of the manipulated variable The manipulated variable is always limited to an upper and a lower limit. "lowerLimit" specifies the lower limit. The value must be specified in the same unit as the manipulated variable (-100.0 … +100.0%)
actValueFactor [1]	REAL	1.0	Actual value factor The actual value is multiplied by the "actValueFactor" to adapt the actual value range
actValueOffset [1]	REAL	0.0	Actual value offset "actValueOffset" is added to the actual value as adaptation.
outValueFactor [1]	REAL	1.0	Manipulated variable factor The manipulated variable is multiplied by the "outValueFactor" to adapt the manipulated variable range

Element	Data type	Initial value	Meaning
outValueOffset [1)]	REAL	0.0	Manipulated variable offset "outValueOffset" is added to the manipulated variable as adaptation.
disturbValue [1)]	REAL	0.0	Disturbance variable A disturbance variable can be applied to "disturbValue". The value must be specified in the same unit as the manipulated variable (-100.0 … +100.0%)
Numeric OutValue	REAL	1.0	Manipulated variable The parameter is permanently assigned to the "analogSpeedFactor" block output.
binOutValue	WORD	16#0000	I/O manipulated variable
upperLimit Reached	BOOL	FALSE	Upper limit of the manipulated variable reached
lowerLimit Reached	BOOL	FALSE	Lower limit of the manipulated variable reached
POutValue	REAL	0.0	P component on the manipulated variable
IOutValue	REAL	0.0	I component on the manipulated variable
DOutValue	REAL	0.0	D component on the manipulated variable
actValue	REAL	0.0	Actual value
derivationValue	REAL	0.0	Control deviation

[1)] Standard parameter of the _CTRL_pid block. For a detailed description, see the SIMOTION Basic Control Function Manual or the online help for the block.

4.5.8.3 Integration of the module

Technology objects, basic parameterization and block call

Table 4- 131 TO type and basic parameterizations

Axis / technology object	
TO type	Not relevant
Version	Not relevant
Modulo	Not relevant
Block call	BackgroundTask The block determines automatically its own call time.

Further blocks and libraries

The **FBLPacDancerControl** block is based on the _CTRL_pid standard block from the SIMOTION command library integrated in SIMOTION.

Note

Detailed descriptions of individual subfunctions and parameters of the SIMOTION internal PID controller block are available in the *SIMOTION Basic Control* Function Manual and in the online help for the **_CTRL_pid** block.

Block/library	Function
_CTRL_pid / SIMOTION command library	PID controller

Error messages

Error number [HEX]	Meaning
16#0000	LPAC_ERR_NO; error-free
16#8082	LPAC_ERR_LOWER_LIMIT_TIMEOUT; dancer lower end position monitoring time expired
16#8083	LPAC_ERR_UPPER_LIMIT_TIMEOUT; dancer upper end position monitoring time expired
16#8084	LPAC_ERR_PID_CTRL_PRELIM_FOIL_FEED; integrated ctrlpid block error message
16#808A	LPAC_ERR_UPPER_SPEED_LIMIT_REACHED; upper speed limit (200%) reached
16#808B	LPAC_ERR_UPPER_SPEED_LIMIT_REACHED; lower speed limit (0%) reached

4.6 Cross sealing

4.6.1 FBLPacCrossSealPos - cross-sealer position axis

4.6.1.1 Technological consideration

The **FBLPacCrossSealPos** block controls the sealing-jaw motion of a cross-sealer drive or longitudinal-sealer drive on intermittent or continuous tubular bagging machines. Prerequisite is a mechanical version in which the sealing jaws are traversed through suitable mechanical coupling elements with a mechanical linear motion.

The module can be used together with other blocks of the LPac library for implementing a SIMOTION complete application of an intermittent or continuous tubular bagging machine

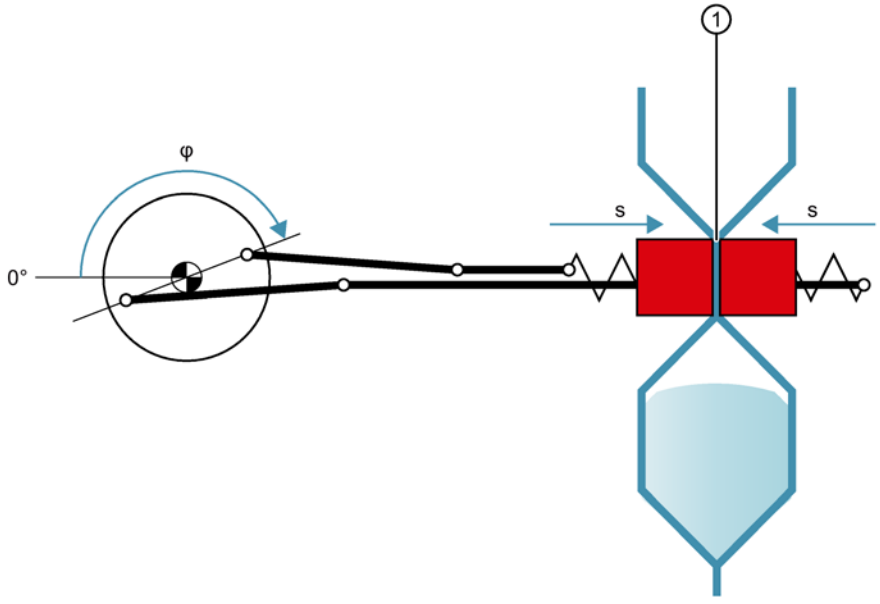
and also independently for similar mechanically-oriented machine units that require positioning functionality.

The mechanical coupling of the sealing jaws on the drive motor can be performed either via belt / geared drive or via a slider-crank mechanical system for which the mechanical coupling elements convert the rotary motor motion to a linear push/pull motion. In the case of a slider-crank mechanical system, the target positions of the sealing jaws are specified as linear positions and converted by the block to the corresponding motor angle. The slider-crank mechanical system is described in the **FBLPacTranslationToRotation** and **FBLPacRotationToTranslation** blocks.

The positioning is performed with parameterizable accelerations/decelerations and with a constant positioning velocity based on the motor angle. The slider-crank mechanical system causes the jaws to be moved linearly with a non-linear path, velocity and force characteristic.

As option, a defined motor torque can be specified at the sealing time so that the sealing is performed with a defined force.

The following figure shows the cross-sealer with slider-crank mechanical system in the "closed" position.

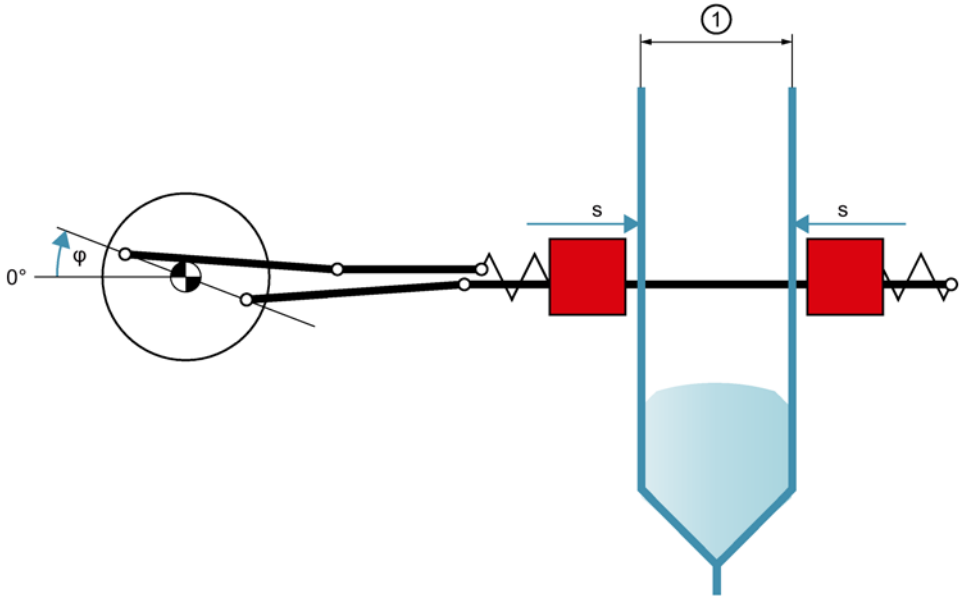


① Definition:
Sealing position for fully closed sealing jaws = 0 mm. The "Sealing jaws closed" position is defined as the mechanical zero position.

φ Motor angle

s Linear jaws motion

The following figure shows the cross-sealer with slider-crank mechanical system in the "open" position.



① Definition:
Open position for opened sealing jaws = bag thickness = x mm

φ Motor angle

s Linear jaws motion

The "Open" position of the sealing jaws is defined as the bag thickness x.

The operator must always specify the linear positions in relation to the sealing jaws as the opening width. The conversion to the motor position is performed automatically by the block. As a result, the displayed sealing jaws position traverses to positive values when opened from 0 mm. The corresponding motor angle changes in the opposite direction.

4.6.1.2 Module description

Scope of functions

The **FBLPacCrossSealPos** function block provides the main and auxiliary functions, and possibly options, shown in the following overview.

Function	Description	Implementation
Main function	Axis sealing procedure (positioning procedure)	Absolute positioning action to perform a sealing sequence in the order: Position the sealing jaws at the closing position → wait the parameterized sealing time, and, when selected, seal with the defined sealing force (torque) → position sealing jaws to the defined opening position. The sealing sequence is performed by controlling the *executeSealing* input. As soon as the sealing time has started, the sealing can be aborted by activating the *executePosOpen* input signal and the cross-sealer traversed immediately to the opening position again.
Options	Mechanical system adaptation	The *boCSWithCrank* signal selects whether a slider-crank mechanical system is supported.
	Sealing to torque	The *enableTorqueSealing* signal specifies whether a defined motor torque should be applied to the sealing jaws in the closed position.
Auxiliary functions	Enable axis	The axis is enabled via the level-triggered *enableAxis* input.
	Home axis	The homing procedure of the axis is started via the *executeHoming* input.
	Jog axis	Positive or negative jogging can be initiated via the level-triggered *jogOpen* and *jogClose* inputs.
	Position axis	The absolute positioning to the parameterizable open/closed position is controlled via the *executePosOpen* and *executePosClose* signals. In addition, relative positioning can be controlled.
	Axis quick stop	A quick stop can be performed or a motion of the axis prevented via the level-triggered *enableEStop* input.

Schematic representation in LAD/FBD

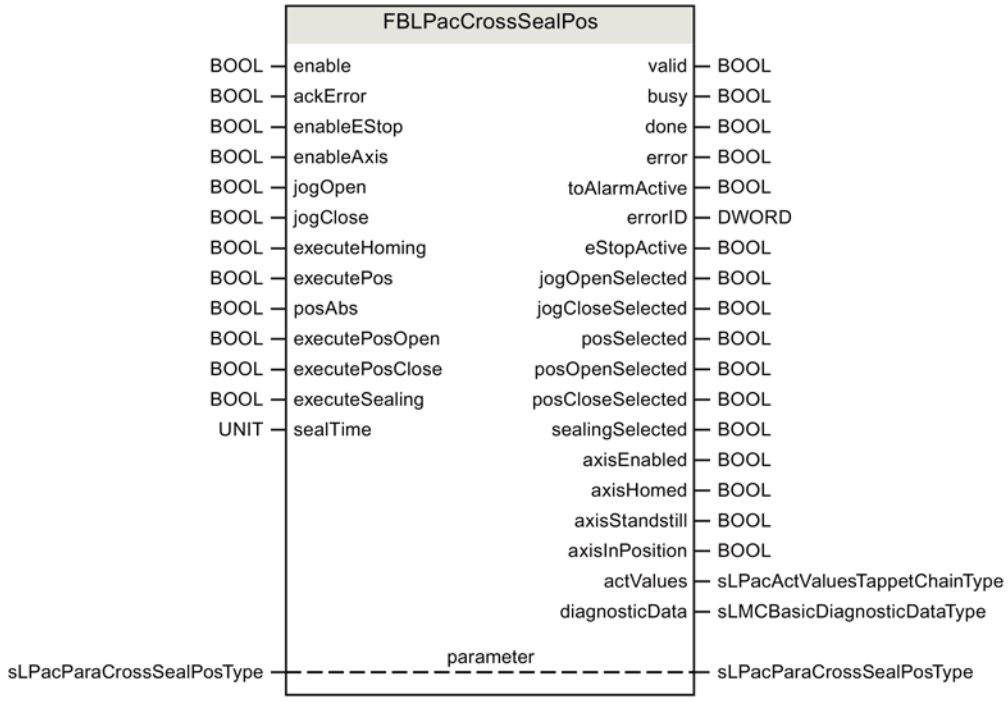


Figure 4-39 FBLPacCrossSealPos – schematic block view in LAD/FBD

Input/output parameters

Table 4- 132 Input and output parameters

Element	P type [1)]	Data type	Initial value	Meaning
enable	IN	BOOL	FALSE	Activation of the FB
ackError	IN	BOOL	FALSE	Acknowledge error
enableEStop	IN	BOOL	FALSE	Activate quick stop
enableAxis	IN	BOOL	FALSE	Enable axis
jogOpen	IN	BOOL	FALSE	Jogging to the open jaws position
jogClose	IN	BOOL	FALSE	Jogging to the close jaws position
exe-cuteHoming	IN	BOOL	FALSE	Start homing
executePos	BOOL	IN	FALSE	Start positioning
posAbs	IN	BOOL	FALSE	TRUE: The positioning initiated with executePos is absolute
				FALSE: The positioning initiated with executePos is relative
exe-cutePosOpen	IN	BOOL	FALSE	Start positioning to the open jaws position
executePos Close	IN	BOOL	FALSE	Start positioning to the close jaws position

Element	P type [1]	Data type	Initial value	Meaning
executeSealing	IN	BOOL	FALSE	Start sealing cycle
sealTime	IN	UINT	FALSE	Sealing time setpoint [ms]
valid	OUT	BOOL	FALSE	Block is active
busy	OUT	BOOL	FALSE	Selected functionality being processed
done	OUT	BOOL	FALSE	Selected functionality completed successfully
error	OUT	BOOL	FALSE	Error occurred
toAlarmActive	OUT	BOOL	FALSE	TO alarm occurred
errorID	OUT	DWORD	16#0000_0000	Error code
eStopActive	OUT	BOOL	FALSE	Quick stop active
jogOpenSelected	OUT	BOOL	FALSE	Jog in the open position direction selected
jogCloseSelected	OUT	BOOL	FALSE	Jog in the closed position direction selected
posSelected	OUT	BOOL	FALSE	Positioning selected
posOpen Selected	OUT	BOOL	FALSE	Positioning to the open position selected
posClose Selected	OUT	BOOL	FALSE	Positioning to the closed position selected
Sealing Selected	OUT	BOOL	FALSE	Sealing cycle active
axisEnabled	OUT	BOOL	FALSE	Cross-sealing axis enabled
axisHomed	OUT	BOOL	FALSE	Axis homed
axisStandstill	OUT	BOOL	FALSE	Axis standstill
axisInPosition	OUT	BOOL	FALSE	Axis has reached the target position
actValues	OUT	sLPacActValuesTappetChainType	-	Actual values
diagnosticData	OUT	sLMCBasicDiagnosticDataType	-	For the diagnostics structure, see Alarm and error messages (Page 333)
parameter	IN_OUT	sLPacParaCrossSealPosType	-	Parameter for cross-sealer

[1] Parameter types: IN = input parameter, OUT = output parameter, IN_OUT = in-out parameter

Table 4- 133 Current values - sLPacActValuesTappetChainType structure

Element	Data type	Initial value	Meaning
r32ActualVelocity	REAL	0.0	Current velocity of the cross-sealer
r32ActualPosition	REAL	0.0	Current position of the cross-sealer
r32ActualOffset	REAL	0.0	Current offset (not relevant)

Table 4- 134 sLPacParaCrossSealPosType structure

Element	Data type	Meaning
sTOs	sLPacToPosDataType	Axis TO for cross-sealer
sMotionData	sLPacCrossSealMo-tionDataType	Parameters for EStop, homing, jog, pos
sConfigData	sLPacCrossSealConfig-DataType	Configuration data for cross-sealer

Table 4- 135 sLPacToPosDataType structure

Element	Data type	Initial value	Meaning
tolAxis	PosAxis	TO#NIL	Position axis for cross-sealer

Table 4- 136 sLPacCrossSealMotionDataType structure

Element	Data type	Meaning
sEStopData	sLMCBasicEStopDataType	Parameter for quick stop
sHomingData	sLMCBasicHomingDataType	Parameter for homing
sJogData	sLPacJogDataType	Parameter for jogging
sPosData	sLPacPosDataType	Parameter for positioning

Quick stop parameters - sEStopData structure

See Axis quick stop (Page 43)

Homing parameters - sHomingData structure

See Home axis (Page 35)

Jog parameters - sJogData structure

See Jog axis (Page 36)

Positioning parameters - sPosData structure

See Position axis (Page 37)

Table 4- 137 Cross-sealer configuration data - sLPacCrossSealConfigDataType structure

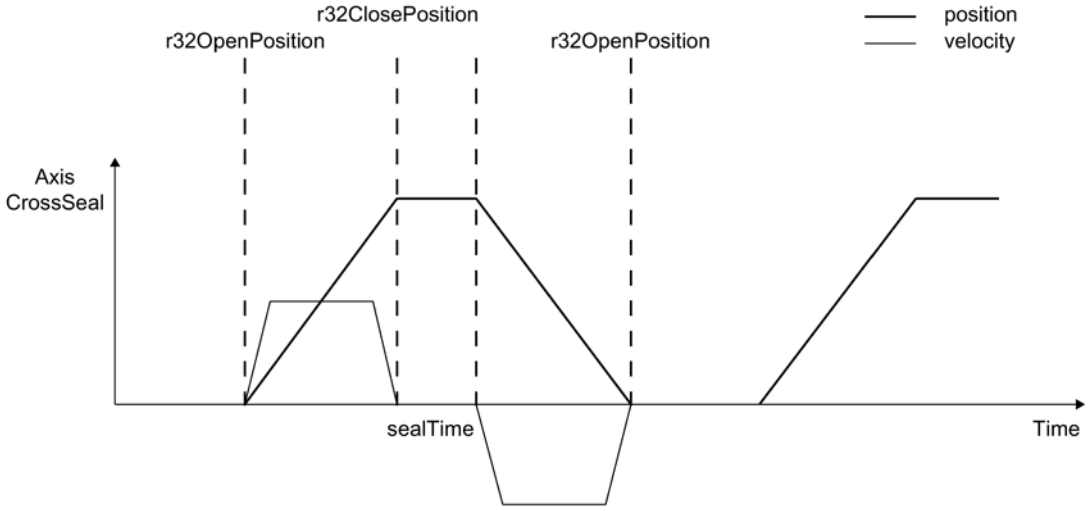
Element	Data type	Initial value	Meaning
r32OpenPosition	REAL	0.0	Linear position for sealing jaws open [mm]
r32ClosePosition	REAL	0.0	Linear position for sealing jaws closed [mm]
r32maxVelocityl	REAL	-1.0	Maximum velocity [1)]
r32maxAcceleration	REAL	-1.0	Maximum acceleration [1)]
r32maxDeceleration	REAL	-1.0	Maximum deceleration [1)]
sTorqueSealing	sLPacCrossSeal-TorqueDataType	-	Parameter for sealing to torque
sCrankShaft	sLPacCrankShaft-DataType	-	Definition data of slider-crank mechanical system
sInterfacePosData	sLPac-CSPosMaxDataType	-	Positioning data interface to the FBLPac-MasterInt block

[1)] With value -1, the respective parameter value is taken from the default on the TO

Table 4- 138 Sealing to torque parameters - sLPacCrossSealTorqueDataType structure

Element	Data type	Initial value	Meaning
boEnableTorqueSealing	BOOL	FALSE	Enable for sealing to torque (the parameter must only be switched when the axis is at standstill!)
sTorqueSealingData	sLPacTor-queSealing-DataType	-	Parameter for sealing to torque
r32SealingTorque	REAL	0.0	Percentage value of max. torque for sealing to torque [%]
r32LocalSealPosition	REAL	0.0	Superimposed position for sealing to torque At which position can the sealing position be overtravelled in order to build up the required torque.
r32SupPosSealVel	REAL	0.0	Velocity for sealing to torque (close)
r32SupPosSealAcc	REAL	0.0	Acceleration for sealing to torque (close)
r32SupPosSealDec	REAL	0.0	Deceleration for sealing to torque (close)
r32SupPosReleaseVel	REAL	0.0	Velocity for sealing to torque (open)
r32SupPosReleaseAcc	REAL	0.0	Acceleration for sealing to torque (open)
r32SupPosReleaseDec	REAL	0.0	Deceleration for sealing to torque (open)

Driving diagram



executePosClose = TRUE: move to r32ClosePosition

executePosOpen = TRUE: move to r32OpenPosition

executeSealing = TRUE: move to r32ClosePosition; wait sealTime at close position; move to r32OpenPosition

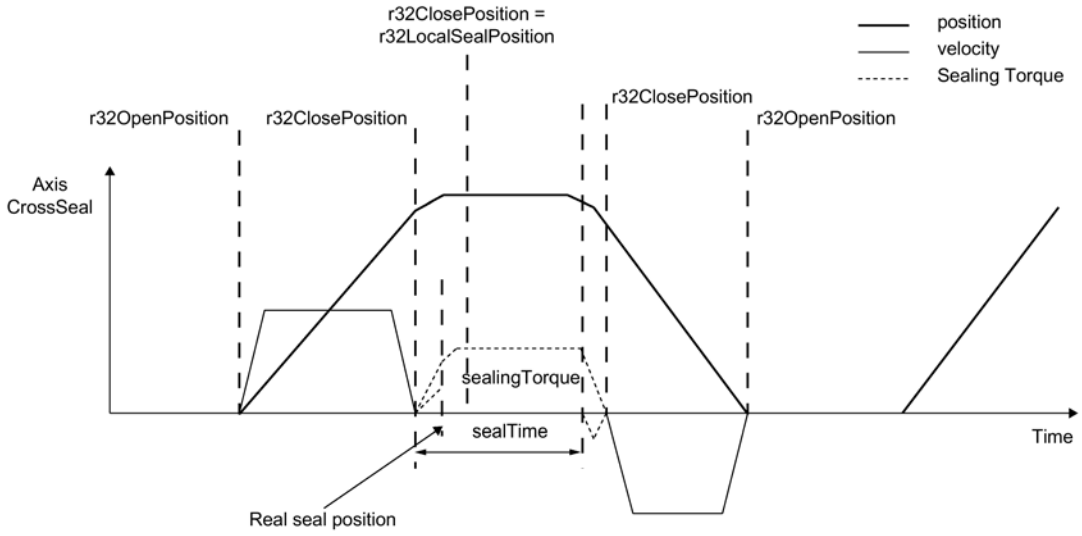
Figure 4-40 Cross-sealer positioning without sealing to torque traversing diagram (motor angle and motor velocity)

Sealing to torque

Sealing to torque is executed as follows:

The axis is first traversed to the passed *closePosition*. The torque limiting is then activated and then a superimposed motion *r32LocalSealPosition* performed for which the torque should be established.

When the sealing action is terminated after expiration of the sealing time, the superimposed positioning is reverse-traversed again and the torque limiting deactivated again. After deactivation, the *openPosition* is approached.



executePosClose = TRUE: move to r32ClosePosition; move superimposed r32LocalSealPosition until sealing torque reached

executePosOpen = TRUE: move superimposed r32LocalSealPosition; move to r32OpenPosition

executeSealing = TRUE: move to r32ClosePosition; move superimposed r32LocalSealPosition until sealing torque reached; wait sealTime; move superimposed r32LocalSealPosition; move to r32OpenPosition

Figure 4-41 Cross-sealer positioning without sealing to torque traversing diagram (motor angle, motor velocity and motor torque)

Table 4- 139 Slider-crank definition parameters - sLPacCrankShaftDataType structure

Element	Data type	Initial value	Meaning
boCSWithCrank-Shaft	BOOL	FALSE	Specification of cross-sealer with slider-crank gear (the parameter must only be switched when the axis is at standstill!)
sCrankShaftData	sLPacCrank-ShaftConfig-DataType	-	Definition data for slider-crank gear

Table 4- 140 Slider-crank definition parameters - sLPacCrankShaftConfigDataType structure

Element	Data type	Initial value	Meaning
r32CrankLength	REAL	0.0	Crank length
r32ConrodLength	REAL	0.0	Conrod length

Element	Data type	Initial value	Meaning
r32verticalOffset	REAL	0.0	Offset displacement between pivot point of the crank and connecting rod of the mechanical system
r32LinearOffset	REAL	0.0	Linear maximum distance of the slider crank

Table 4- 141 Cross-sealer positioning parameters - sLPacCSPosMaxDataType structure

Element	Data type	Initial value	Meaning
boCSWithCrank-Shaft	BOOL	FALSE	Specification of cross-sealer with slider-crank gear (the parameter must only be switched when the axis is at standstill!)
sOpenData	sLPacPosMaxDyn-DataType	-	Positioning parameter for opening motion
sCloseData	sLPacPosMaxDyn-DataType	-	Positioning parameter for closing motion

Table 4- 142 sLPacPosMaxDynDataType structure

Element	Data type	Initial value	Meaning
r32Position	REAL	0.0	Open or closed position of the cross-sealer
r32MaxVelocity	REAL	0.0	Maximum velocity of the axis
r32MaxAcceleration	REAL	0.0	Maximum acceleration of the axis
r32MaxDeceleration	REAL	0.0	Maximum deceleration of the axis
r64VelocityOverride	LREAL	0.0	Velocity override
r64AccelerationOverride	LREAL	0.0	Acceleration override

4.6.1.3 Integration of the module

Technology objects, basic parameterization and block call

Table 4- 143 TO type and basic parameterizations

Axis / technology object	
TO type	Position axis
Version	Linear axis
Modulo	The block automatically sets the modulo length to 0 and the Modulo yes/no property to no.
Block call	BackgroundTask

Further blocks and libraries

The **FBLPacCrossSealPos** block uses the following blocks internally.

Block/library	Function
FBLMCBasicAxis / LMCBasic	Axis function block
FBLPacTranslationToRotation / LPac	Linear position conversion of the sealing jaws in the motor angle for the slider-crank mechanical system
FBLPacRotationToTranslation / LPac	Conversion of the motor angle in the linear position of the sealing jaws for the slider-crank mechanical system

Some of the internally used blocks originate from other libraries. See Section Additional required libraries (Page 15).

Error messages

Error number [HEX]	Meaning
16#0000	ERR_ERR_NO; error-free
16#8000	ERR_INVALID_STEP; internal error, please get in touch with your Siemens contact partner
16#8001	LPAC_ERR_NO_AXIS; no axis TO specified
16#8003	LPAC_ERR_TO_ALARM_ACTIVE; TO alarm active; error message from axis function block Error description, see *SIMOTION Axis Function Block* Application Manual and Section Alarm and error messages (Page 333) in this manual
16#8004	LPAC_ERR_AXIS_NOT_DISABLED; axis not disabled for change of the modulo length
16#8008	LPAC_ERR_POS_VELOCITY_ZERO; positioning velocity ≤ 0
16#8009	LPAC_ERR_POS_ACC_DEC_ZERO; acceleration/deceleration for positioning ≤ 0
16#8011	LPAC_ERR_AXIS_FUNCTION_BLOC; error message from axis function block For details, see *diagnosticData.b32ErrorCode* output structure; Error description, see *SIMOTION Axis Function Block* Application Manual
16#8014	LPAC_ERR_AXIS_NOT_HOMED; axis not homed
16#8043	LPAC_ERR_ENABLE_TORQUE_LIMIT; error for the err_enableTorqueLimiting function; the return value of the function is output in the high word of *ErrorID*
16#8044	LPAC_ERR_DISABLE_TORQUE_LIMIT; error for the _disableTorqueLimiting function; the return value of the function is output in the high word of *ErrorID*
16#80D0	LPAC_ERR_INVALID_INPUT_PARAMETER; invalid mechanical definition parameter (crank length, conrod length, vertical offset); source FBLPacTranslationToRotation
16#80D1	LPAC_ERR_VERTICAL_OFFSET_LENGTH_INVALID; invalid mechanical definition parameter (vertical offset); source FBLPacTranslationToRotation or FBLPacRotationToTranslation
16#80D2	LPAC_ERR_TRANSLAT_POS_INVALID; linear position outside the maximum possible position range; source FBLPacTranslationToRotation

Error number [HEX]	Meaning
16#80E0	LPAC_ERR_CRANK_OR_CONROD_LENGTH_INVALID; invalid mechanical definition parameter (crank length, conrod length); source FBLPacRotationTo-Translation
16#80E1	LPAC_ERR_ROTATION_ANGLE_INVALID; crank angle outside the maximum possible angle range; source FBLPacRotationToTranslation

Note

The **pCrossSealPos** program unit is available in the LPac project to illustrate the use of the **FBLPacCrossSealPos** block.

This unit contains the programs for the required block calls in the StartupTask and the BackgroundTask.

4.6.2 FBLPacCrossSealCam - cross-sealer cam

4.6.2.1 Technological consideration

The **FBLPacCrossSealCam** block is used to control the sealing motion of a continuous cross-sealer drive. Prerequisite is a mechanical version in which the sealing jaws are traversed through suitable mechanical coupling elements with a rotary motion. These mechanical coupling elements can be, for example, a simple gearbox or a crank or belt gearbox. The
motion control is based on a cam function. The use of different cams allows different mechanical versions to be implemented.

The module can be used together with other blocks of the LPac library for implementing a SIMOTION complete application of a continuous tubular bagging machine and also independently for similar mechanically-oriented machine units that require an electronic camming for a leading axis.

The "No product, no bag" functionality can be implemented with the aid of the synchronization function. In intermittent operation, the axis is thereby synchronized or desynchronized via the *executeCamIn* and *executeCamOut* inputs after completion of the relevant cam cycle in order to avoid empty bags. As a rule, the detection of a missing product, e.g. an empty compartment in a product supply chain, is at a certain distance before the actual sealing, so that in this case a shift register to track the empty pocket must be implemented in the application. The settings for the cam synchronization must be taken into account for the axis parameterization and in conjunction with the standard axis function block.

To implement a "No Product No Bag" function in the LPac project, an additional so-called NPNB axis is connected as virtual axis between the machine leading axis (virtual master) and the cross-sealer axis and foil feed axis. The NPNB axis is coupled as virtual master axis with a 1:1 gearing. Cross-sealer and foil feed are coupled to the NPNB axis instead of the leading axis. In this way, the desynchronization and synchronization functions on the NPNB axis can desynchronize or synchronize the two relevant following axes simultaneously and

therefore "cut" the corresponding product cycles when the product is missing. A preconfigured program module (pNPNBGear) is available in the LPac project for the simple implementation of the NPNB functionality.

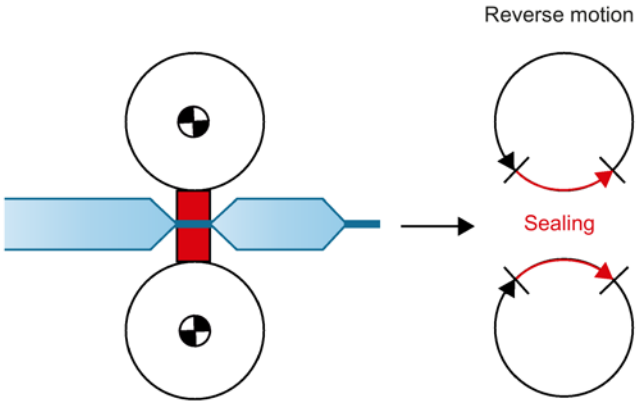


Figure 4-42 Rotating cross-sealer variant 1

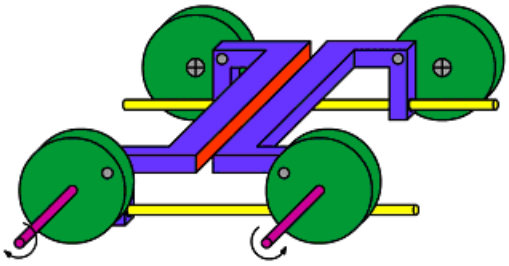


Figure 4-43 Rotating cross-sealer variant 2

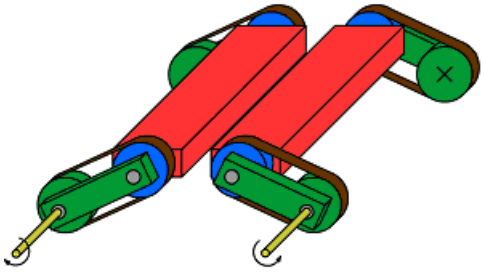


Figure 4-44 Rotating cross-sealer variant 3

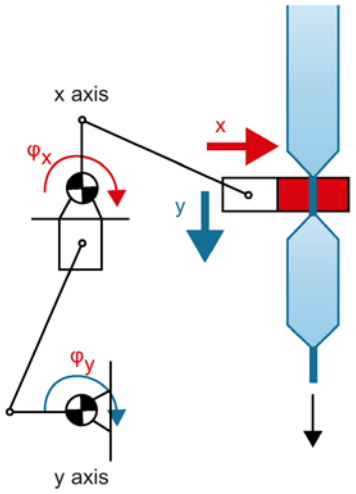


Figure 4-45 Box motion cross-sealer (FBLPacCrossSealCam for y-axis, FBLPacCrossSealPos for x-axis)

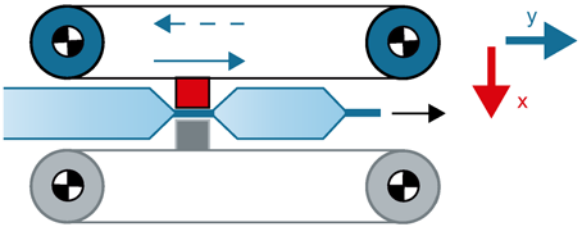


Figure 4-46 Coupled motion cross-sealer

4.6.2.2 Module description

The **FBLPacCrossSealCam** block provides a general module that masters camming. This makes the block ideal for continuously operating machine modules.

The block can be passed cams that were created with the **FBLPacCreaCamBoxMotCS**, **FBLPacCreaCamCSRot**, **FBLPacCreaCamCrankLin** function blocks or in some other manner.

Different mechanical versions of the cross-sealer are taken into account by using different cams.

Scope of functions

The **FBLPacCrossSealCam** function block provides the main and auxiliary functions, and possibly options, shown in the following overview.

Function	Description	Implementation
Main function	Camming	Camming with static or dynamic synchronization/desynchronization (*executeCamIn*/*executeCamOut*) on the master
Options	Offset adjustment	In camming, a relative offset can be set during machine operation. The currently passed offset between the master and slave is reset with every rising edge at *executeChangeOffset*. The set offset is saved retentively and automatically set again at every following synchronization.
Auxiliary functions	Enable axis	The axis is enabled via the level-triggered *enableAxis* input.
	Home axis	The homing procedure of the axis is started via the *executeHoming* input.
	Jog axis	Positive or negative jogging can be started via the level-triggered *jogForward* or *jogBackward* inputs.
	Position axis	An absolute or relative positioning action can be started via the edge-triggered *executePos* input.
	Axis quick stop	A quick stop can be performed or a motion of the axis prevented via the level-triggered *enableEStop* input.
	Change of the axis configuration data	The axis configuration data (modulo length, modulo yes/no) can be changed via the subordinate **FBLPacChangeTOConfigDataMod** function block. The modulo length specified in the ramp-up of the controller is automatically entered and activated by the block in the appropriate configuration data element of the axis. The modulo properties can also be changed at any time in the cyclic program by specifying different values. The user must only ensure that the axis is not enabled. A TO restart must be performed to activate the changed configuration data.

Schematic block view in LAD/FBD

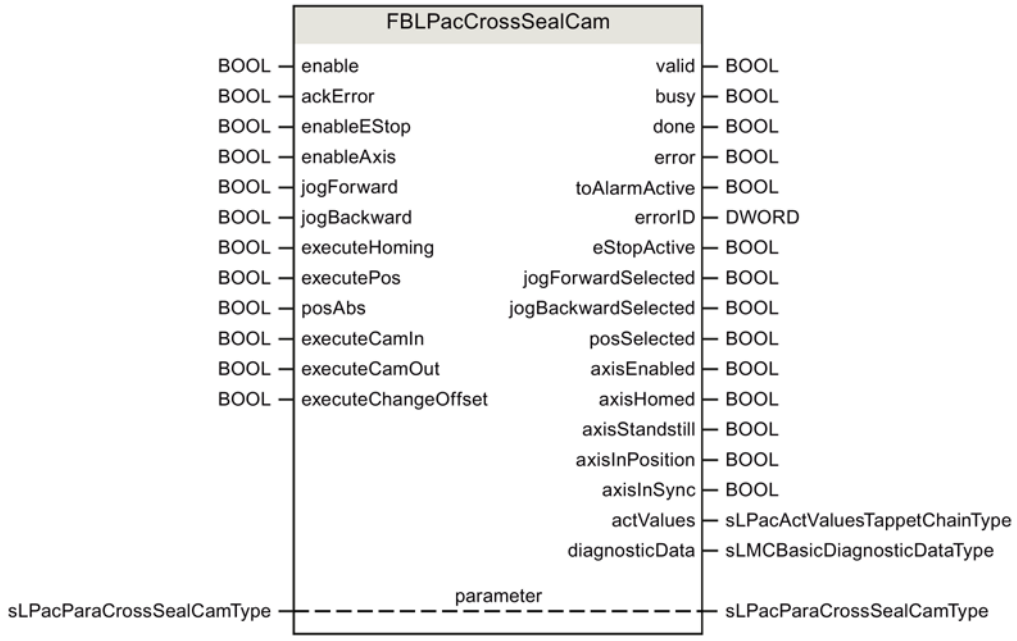


Figure 4-47 FBLPacCrossSealCam – schematic block view in LAD/FBD

Input/output parameters

Table 4- 144 Input and output parameters

Element	P type [1]	Data type	Initial value	Meaning
enable	IN	BOOL	FALSE	Activation of the FB
ackError	IN	BOOL	FALSE	Acknowledge error
enableEStop	IN	BOOL	FALSE	Activate quick stop
enableAxis	IN	BOOL	FALSE	Enable axis
jogForward	IN	BOOL	FALSE	Continuous jogging in the positive direction
jogBackward	IN	BOOL	FALSE	Continuous jogging in the negative direction
executeHoming	IN	BOOL	FALSE	Start homing
executePos	IN	BOOL	FALSE	Start positioning
posAbs	IN	BOOL	FALSE	TRUE: The positioning initiated with *executePos* is absolute FALSE: The positioning initiated with *executePos* is relative
executeCamIn	IN	BOOL	FALSE	Activate camming / start synchronization
executeCamOut	IN	BOOL	FALSE	Deactivate camming / start desynchronization
executeChange Offset	IN	BOOL	FALSE	Activate start offset displacement; the offset adjustment acts relative, i.e. at each activation, the specified offset value is traversed relatively.

Element	P type [1]	Data type	Initial value	Meaning
valid	OUT	BOOL	FALSE	Block is active
busy	OUT	BOOL	FALSE	Selected functionality being processed
done	OUT	BOOL	FALSE	Selected functionality completed successfully
error	OUT	BOOL	FALSE	Error occurred
toAlarmActive	OUT	BOOL	FALSE	TO alarm occurred
errorID	OUT	DWORD	16#0000_0000	Error code
eStopActive	OUT	BOOL	FALSE	Quick stop active
jogForwardSelected	OUT	BOOL	FALSE	Jogging positive direction selected
jogBackwardSelected	OUT	BOOL	FALSE	Jogging negative direction selected
posSelected	OUT	BOOL	FALSE	Positioning selected
axisEnabled	OUT	BOOL	FALSE	Axis enabled
axisHomed	OUT	BOOL	FALSE	Axis homed
axisStandStill	OUT	BOOL	FALSE	Axis at standstill
axisInPosition	OUT	BOOL	FALSE	Axis in position
axisInSync	OUT	BOOL	FALSE	Axis synchronous to the master axis
actValues	OUT	sLPacActValuesTappetChainType	-	Actual values
diagnosticData	OUT	sLMCBasicDiagnosticDataType	-	For the diagnostics structure, see Alarm and error messages (Page 333)
parameter	IN_OUT	sLPacParaCrossSealCamType	-	Parameter for cross-sealer

[1] Parameter types: IN = input parameter, OUT = output parameter, IN_OUT = in-out parameter

Table 4- 145 sLPacActValuesTappetChainType structure

Element	Data type	Initial value	Meaning
r32ActualVelocity	REAL	0.0	Current velocity of the cross-sealer
r32ActualPosition	REAL	0.0	Current position of the cross-sealer
r32ActualOffset	REAL	0.0	Current offset of the cross-sealer

Table 4- 146 sLPacParaCrossSealCam structure

Element	Data type	Meaning
sTOs	sLPacToCamDataType	TOs for cross-sealer
sMotionData	sLPacCrossSealCamMotionDataType	Parameters for EStop, homing, jog, pos, camming, offset adjustment
sConfigData	sLPacCrossSealCamConfigDataType	Configuration data for cross-sealer

Table 4- 147 sLPacToCamDataType structure

Element	Data type	Initial value	Meaning
toAxis	followingAxis	TO#NIL	Synchronous axis for cross-sealer
toLeadingValue	masterType	TO#NIL	Technology object for the leading axis PosAxis or external encoder
toCam	camType	TO#NIL	Cam for synchronous operation

Table 4- 148 sLPacCrossSealCamMotionDataType structure

Element	Data type	Meaning
sEStopData	sLMCBasicEStopDataType	Parameter for quick stop
sHomingData	sLMCBasicHomingDataType	Parameter for homing
sJogData	sLPacJogDataType	Parameter for jogging
sPosData	sLPacPosDataType	Parameter for positioning
sCammingData	sLPacCammingDataType	Parameter for camming
sPhasingData	sLMCBasicPhasingDataType	Parameter for offset adjustment

Quick stop parameters - sEStopData structure

See Axis quick stop (Page 43)

Homing parameters - sHomingData structure

See Home axis (Page 35)

Jog parameters - sJogData structure

See Jog axis (Page 36)

Positioning parameters - sPosData structure

See Position axis (Page 37)

Camming parameters - sCammingData structure

See Synchronizing/desynchronizing axis in camming (Page 40)

Offset adjustment parameters - sPhasingData structure

See Offset adjustment (Page 43)

Table 4- 149 sLPacCrossSealCamConfigDataType structure

Element	Data type	Meaning
sAxisMechanicalData	sLPacAxisMechanical-DataType	Mechanical data

Table 4- 150 sLPacAxisMechanicalDataType structure

Element	Data type	Initial value	Meaning
r32ModuloLength	REAL	0.0	Modulo length
boModuloActive	BOOL	TRUE	Activate modulo length

4.6.2.3 Integration of the module

Technology objects, basic parameterization and block call

Table 4- 151 TO type and basic parameterizations

Axis / technology object	
TO type	Following axis
Version	Rotary axis or linear axis depending on the application
Modulo	For example, 360° or sealing length
Block call	BackgroundTask

Further blocks and libraries

The **FBLPacCrossSealCam** block uses the following blocks internally.

Block/library	Function
FBLMCBasicAxis / LMCBasic	Axis function block
FBLPacChangeTOConfigDataMod / LPac	Function block to change the configuration data of the TO axis

Some of the internally used blocks originate from other libraries. See Section Additional required libraries (Page 15).

Error messages

Error number [HEX]	Meaning
16#0000	ERR_ERR_NO; error-free
16#8000	ERR_INVALID_STEP; internal error, please get in touch with your Siemens contact partner
16#8001	LPAC_ERR_NO_AXIS; no axis TO specified
16#8003	LPAC_ERR_TO_ALARM_ACTIVE; TO alarm active; error message from axis function block Error description, see *SIMOTION Axis Function Block* Application Manual and Section Alarm and error messages (Page 333) in this manual
16#8004	LPAC_ERR_AXIS_NOT_DISABLED; axis not disabled for change of the modulo length
16#8005	LPAC_ERR_MODULO_LENGTH; invalid modulo length
16#8008	LPAC_ERR_POS_VELOCITY_ZERO; positioning velocity ≤ 0
16#8009	LPAC_ERR_POS_ACC_DEC_ZERO; acceleration/deceleration for positioning ≤ 0
16#800B	LPAC_ERR_POS_ABS_POSITION_VALUE; target position for absolute positioning < 0.0 or > modulo length
16#800C	LPAC_ERR_ PHASING_DYNAMIC_DATA; offset adjustment for gearing; invalid velocity and/or acceleration parameters
16#8010	LPAC_ERR_NO_MASTER_AXIS; no axis TO specified for leading axis
16#8011	LPAC_ERR_AXIS_FUNCTION_BLOC error message from axis function block For details, see *diagnosticData.b32ErrorCode* output structure; Error description, see *SIMOTION Axis Function Block* Application Manual
16#8012	LPAC_ERR_NO_CAM, no cam specified
16#8013	LPAC_ERR_NO_FOLLOWING_OBJECT; no following object specified or leading axis is not a following axis
16#8061	LPAC_ERR_NO_EXTERNAL_ENCODER; no TO externalEncoder specified
16#8078	LPAC_ERR_SET_GEAR_OR_CAMMING_OFFSET; error message for the offset adjustment from the setGearingsOffset or setCammingOffset command
16#80C3	LPAC_ERR_EXECUTE_DURING_RESET; Execute requested while error acknowledgment active
16#8100	LPAC_ERR_READ_SYSTEM_VARIABLE; error when reading the system variables / variable cannot be read
16#8101	LPAC_ERR_MAX_TIME_FOR_TO_RESTART; TO restart not performed within the monitoring time

Note

The pCrossSealerCam and pCrossSealerGear program units are available in the LPac project to illustrate the use of the FBLPacCrossSealCam and FBLPacCrossSealGear blocks.

These units contain the programs for the required block calls in the StartupTask and the BackgroundTask.

4.6.3 FBLPacCrossSealChainBag – chain bag control

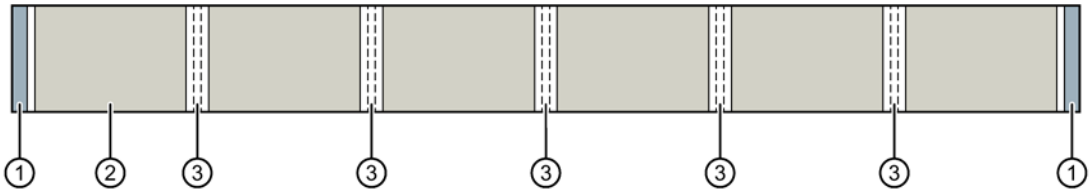
4.6.3.1 Technological consideration

The **FBLPacCrossSealChainBag** block is used to control a cross-sealer unit comprising two rotating cross-sealers for the production of so-called chain bags. The block can perform the following subfunctions:

- Control of the homing sequence
- Control of the travel to the basic position and synchronization to the machine leading axis
- Offset adjustment between the two cross-sealer drives coupled in gearing for the implementation of cut and perforation

The module can be used together with other blocks of the LPac library for implementing a SIMOTION complete application of a continuous tubular bagging machine and also independently for similar mechanically-oriented machine units.

Chain bags are any number of consecutive single bags with the same content. The individual bags can be perforated by the user for simple portioning.



① Sealing with cut
② Single bag
③ Sealing with perforation

Rotating cross-cutter units as shown in the following figure are frequently used for the production of chain bags on horizontal tubular bagging machines.

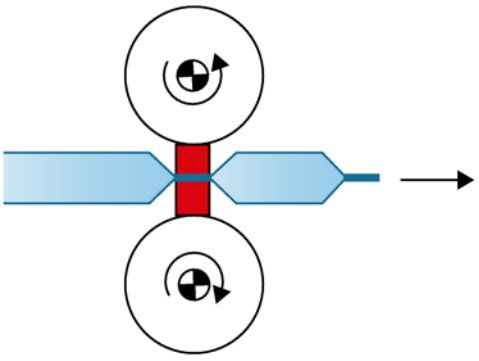
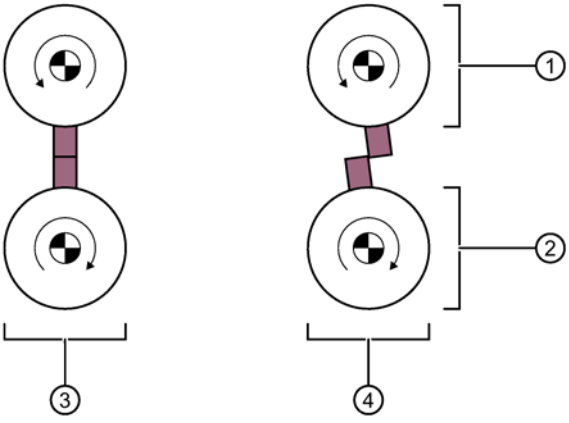


Figure 4-48 Rotating cross-sealer with two axes

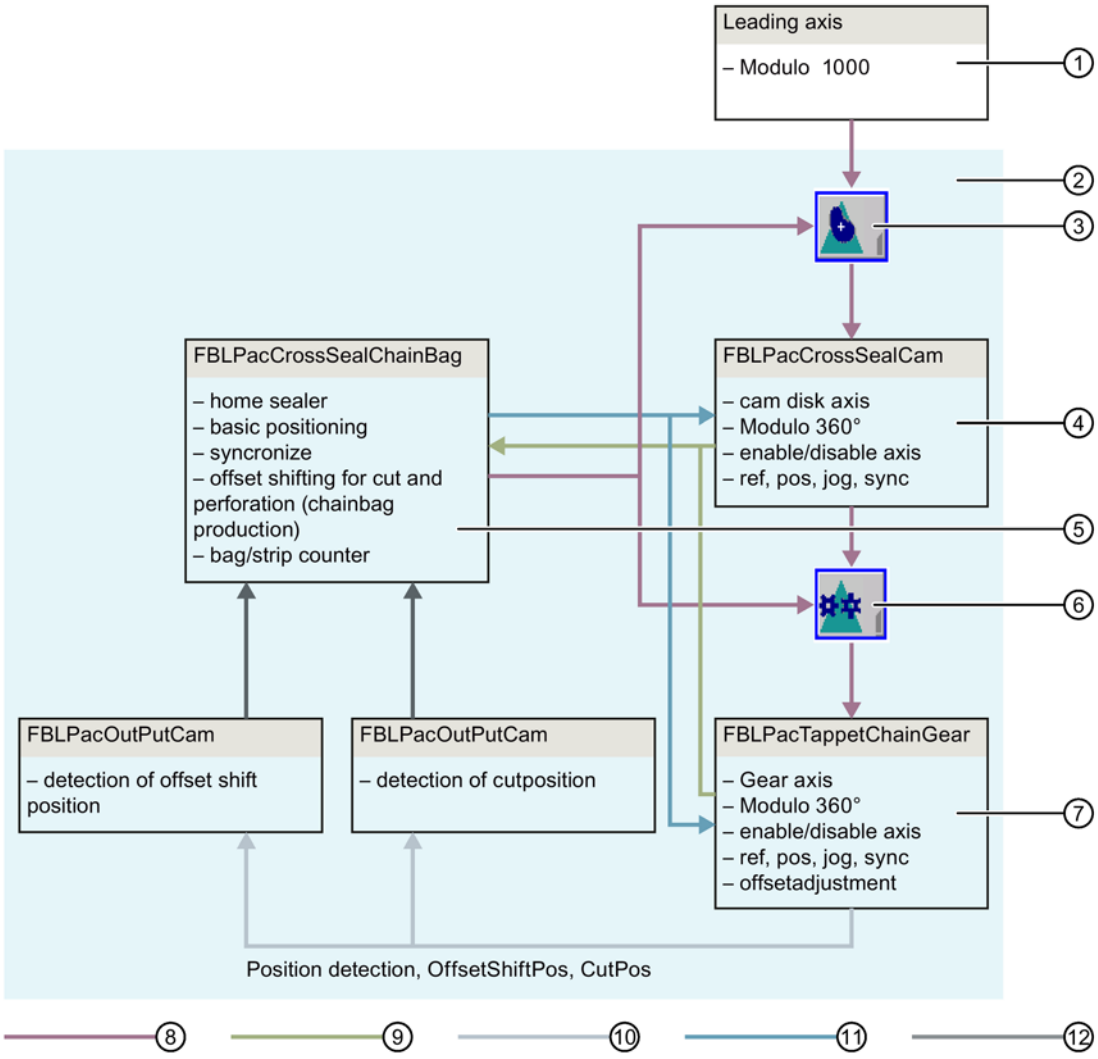
The following diagram provides a schematic representation of the cut and perforation position.



① Axis 1: Cross-sealer as leading axis
② Axis 2: Cross-sealer as following axis
③ Sealing drum in the sealing/cut position without offset
④ Sealing drum in the perforation position with offset

Both cross-sealing axes are coupled in gearing and the cut and perforation position is set through adjustment of the offset on the following axis (Axis 2).

The block supports the following axis configuration shown in schematic representation:



① Machine leading axis (FBLPacMasterCont) or NPNB axis (FBLPacTappetChainGear)
② Structure for the control of a rotating 2-axis cross-sealer for chain bag production with FBL-PacCrossSealChainBag
③ Camming with cam disk 360/1000
④ Upper sealing drum = cross-sealer leading axis = Axis 1
⑤ Control of sealing axes and chain bag production
⑥ Gearing with 1:1 gear ratio
⑦ Lower sealing drum = cross-sealer following axis = Axis 2
⑧ Offset setpoints
⑨ Status signal: enabled, homed, standStill, inPosition, inSync
⑩ Axis position
⑪ Control communication: home, jog, pos, sync
⑫ Output cam: Cut position, Offset shift position

In the axis configuration shown above, one of the two cross-sealer drives is displayed as a real cam axis (in this case, the "upper sealing drum" called Axis 1). On this leading axis of

the cross-sealer unit, the second cross-sealer axis (in this case, the "lower sealing drum" called Axis 2) is coupled in 1:1 gearing.

The block designates the leading axis as Axis 1 and the following axis as Axis 2.

4.6.3.2 Module description

The **FBLPacCrossSealChainBag** block represents a special block for use in horizontal tubular bagging machines.

Scope of functions

The **FBLPacCrossSealChainBag** function block provides the main and auxiliary functions, and possibly options, shown in the following overview.

Function	Description	Implementation
Main functions	Home axis grouping	Passive (flying) homing is performed on both cross-sealer axes. To do this, both axes are traversed simultaneously with a jog command with the same traversing parameters (dynamic response) until the "Axis home" feedback signal is received from both axes. Both axes are then stopped simultaneously and the *SealerHomed* signal output. The step can be skipped by setting the *boSkipHomingJog* configuration switch in order to also be able to operate axes with absolute encoders. The user must ensure that both axes are supplied with the same parameters for the jog motion in order to avoid a collision.
	Position axis grouping in basic position and synchronize to the machine leading axis	After homing, the two axes can be positioned in succession at a specifiable basic position, and then synchronized to the respective leading axis.
	Automatic offset adjustment for chain bag production	When the automatic offset adjustment is activated, the offset position for the cutting position of the axes is set once. The first cut (start of the chain bag) is made with the next axis rotation of the cross-sealer. When the specifiable offset adjustment position is reached, the offset is set to the specifiable value for the perforation position, and the chain bag counter incremented. When the counter reaches the set number of bags, the offset required for the cutting position is activated once again, and the cycle starts again. The number of produced chain bags as well as the single bags are continuously counted and displayed.

Function	Description	Implementation
Options	-	-
Auxiliary functions	Quick stop / abort cycle	The active execution sequences for homing, travel to basic position / synchronization as well as the chain bag production sequence can be aborted via the level-triggered *enableEStop* input. The relevant axes are not stopped via this block!
	Activate cut position	When synchronous operation is activated between the two sealing axes, the offset position for the cut position is activated via the edge-triggered *executeCut* input.
	Activate perforation position	When synchronous operation is activated between the two sealing axes, the offset position for the perforation position is activated via the edge-triggered *executePerforation* input.
	Reset bag counter	Single and chain bag counters are reset to 0 with the activation of the edge-triggered *resetCounter* input.

Schematic block view in LAD/FBD



Figure 4-49 FBLPacCrossSealChainBag – schematic block view in LAD/FBD

Input/output parameters

Table 4- 152 Input and output parameters

Element	P type [1]	Data type	Initial value	Description
enable	IN	BOOL	FALSE	Activation of the FB
ackError	IN	BOOL	FALSE	Acknowledge error
enableEStop	IN	BOOL	FALSE	Cancel active cycles; no quick stop of the axes
executeHoming	IN	BOOL	FALSE	Start homing sequence
executeSyncIn	IN	BOOL	FALSE	Start travel to the basic position and synchronization sequence
Enable Production	IN	BOOL	FALSE	Start chain bag production / offset adjustment sequence
executeCut	IN	BOOL	FALSE	Activate cross-sealer following axis (Axis 2) offset for cut position
Execute Perforation	IN	BOOL	FALSE	Activate cross-sealer following axis (Axis 2) offset for perforation position
axis1Enabled	IN	BOOL	FALSE	Axis enabled feedback signal from cross-sealer leading axis (Axis 1)
axis1Homed	IN	BOOL	FALSE	Axis homed feedback signal from cross-sealer leading axis (Axis 1)
axis1StandStill	IN	BOOL	FALSE	Axis at standstill feedback signal from cross-sealer leading axis (Axis 1)
axis1PosSelected	IN	BOOL	FALSE	Positioning selected feedback signal from cross-sealer leading axis (Axis 1)
axis1InPosition	IN	BOOL	FALSE	Axis position reached feedback signal from cross-sealer leading axis (Axis 1)
axis1InSync	IN	BOOL	FALSE	Axis synchronized feedback signal from cross-sealer leading axis (Axis 1)
axis2Enabled	IN	BOOL	FALSE	Axis enabled feedback signal from cross-sealer following axis (Axis 2)
axis2Homed	IN	BOOL	FALSE	Axis homed feedback signal from cross-sealer following axis (Axis 2)
axis2StandStill	IN	BOOL	FALSE	Axis at standstill feedback signal from cross-sealer following axis (Axis 2)
axis2PosSelected	IN	BOOL	FALSE	Positioning selected feedback signal from cross-sealer following axis (Axis 2)
axis2InPosition	IN	BOOL	FALSE	Axis position reached feedback signal from cross-sealer following axis (Axis 2)
axis2InSync	IN	BOOL	FALSE	Axis synchronized feedback signal from cross-sealer following axis (Axis 2)
axis2InOffsetShiftPosition	IN	BOOL	FALSE	Cross-sealer in offset adjustment position (output signal from cam controller)
axis2InCutPosition	IN	BOOL	FALSE	Cross-sealer in cut position (output signal from cam controller)

Element	P type [1]	Data type	Initial value	Description
reset-BagCounter	IN	BOOL	FALSE	Reset bag and chain bag counters
bagUnitLength	IN	UINT	0	Number of bags per chain bag
valid	OUT	BOOL	FALSE	Block is active
busy	OUT	BOOL	FALSE	Selected functionality being processed
done	OUT	BOOL	FALSE	Selected functionality completed successfully
error	OUT	BOOL	FALSE	Error occurred
errorID	OUT	DWORD	16#0000_0000	Error code
homingActive	OUT	BOOL	FALSE	Homing sequence active
syncActive	OUT	BOOL	FALSE	Travel to the basic position and synchronization sequence active
productionActive	OUT	BOOL	FALSE	Chain bag production active
sealerHomed	OUT	BOOL	FALSE	Cross-sealer unit homed
sealerInSync	OUT	BOOL	FALSE	Cross-sealer unit synchronized
executeHome-Axes	OUT	BOOL	FALSE	Cross-sealer axis passive homing
jogAxes	OUT	BOOL	FALSE	Jog cross-sealer axis for passive homing
executePos Axis1	OUT	BOOL	FALSE	Start basic positioning of cross-sealer leading axis (Axis 1)
executeSync Axis1	OUT	BOOL	FALSE	Synchronize cross-sealer leading axis (Axis 1)
executeDe SyncAxis1	OUT	BOOL	FALSE	Desynchronize cross-sealer leading axis (Axis 1)
executePos Axis2	OUT	BOOL	FALSE	Start basic positioning of cross-sealer following axis (Axis 2)
executeSync Axis2	OUT	BOOL	FALSE	Synchronize cross-sealer following axis (Axis 2)
executeDe SyncAxis2	OUT	BOOL	FALSE	Desynchronize cross-sealer following axis (Axis 2)
executeOff-setCut	OUT	BOOL	FALSE	Activate cross-sealer following axis (Axis 2) offset for cut position
executeOff-setPer	OUT	BOOL	FALSE	Activate cross-sealer following axis (Axis 2) offset for perforation position
bagCounter	OUT	DINT	0	Number of single bags in the chain bag
totalBagCounter	OUT	DINT	0	Total number of single bags
stripCounter	OUT	DINT	0	Number of chain bags
actualOff-setAxis2	OUT	REAL	0	Current position offset on the cross-sealer following axis
parameter	IN_OUT	sLPacPara-CrossSeal-ChainBagType	-	Parameter for cross-sealer unit for the chain bag production

[1] Parameter types: IN = input parameter, OUT = output parameter, IN_OUT = in-out parameter

Table 4- 153 sLPacParaCrossSealChainBagType structure

Element	Data type	Meaning
sTOs	sLPacToPosDataType	TO for cross-sealer following axis (Axis 2)
sMotionData	sLPacChainBagMotionDataType	Parameters for jog, pos, offset adjustment
sChainBagData	sLPacChainBagDataType	Parameter for chain bag production
sConfigData	sLPacConfigChainBagType	Configuration switch for sequence control during homing and synchronization
sOffsetShiftOutPutCamData	sLPacParaOutPutCamType	Parameter for cam controller to detect the offset shift position
sCuttingOutPutCamData	sLPacParaOutPutCamType	Parameter for cam controller to detect the cut position

Table 4- 154 sLPacToPosDataType structure

Element	Data type	Initial value	Meaning
toAxis	FollowingAxis	TO#NIL	Gearing axis for cross-sealer following axis (Axis 2)

Table 4- 155 sLPacChainBagMotionDataType structure

Element	Data type	Meaning
sJogData	sLPacJogDataType	Parameter for jogging
sPosData	sLPacPosDataType	Parameter for positioning
sPhasingData	sLMCBasicPhasingDataType	Parameter for offset adjustment

Jog parameters - sJogData structure

See Jog axis (Page 36)

Positioning parameters - sPosData structure

See Position axis (Page 37)

Offset adjustment parameters - sPhasingData structure

See Offset adjustment (Page 43)

Table 4- 156 sLPacChainBagDataType structure

Element	Data type	Initial value	Meaning
r32Perforation Offset	REAL	0.0	Offset for perforation position on the cross-sealer following axis (Axis 2)
r32CuttingOffset	REAL	0.0	Offset for cut position on the cross-sealer following axis (Axis 2)
r32CutStart Position	REAL	0.0	Starting position of the cut position on the cross-sealer following axis (Axis 2); defines the output cam required for the bag count
r32CutEnd Position	REAL	0.0	End position of the cut position on the cross-sealer following axis (Axis 2); defines the output cam required for the bag count
r32StartOffset ShiftPosition	REAL	0.0	Starting position of the offset shift position on the cross-sealer following axis (Axis 2); defines the output cam required for the start of the offset adjustment
r32EndOffset ShiftPosition	REAL	0.0	End position of the offset shift position on the cross-sealer following axis (Axis 2); defines the output cam required for the end of the offset adjustment

Table 4- 157 sLPacConfigChainBagType structure

Element	Data type	Initial value	Meaning
boSki-pHomingJog	BOOL	FALSE	Configuration switch for the homing sequence. The "Passive homing and jog" step is skipped when this switch is set. In this way, the sequence can also be used for axes with absolute encoders that do not have to be homed.
boSkipSync Axis2Pos	BOOL	FALSE	Configuration switch for the travel to the basic position of the cross-sealer following axis (Axis 2). The "Basic positioning" step is skipped when this switch is set.
boSkipSync Axis2Sync	BOOL	FALSE	Configuration switch for the synchronization of the cross-sealer following axis (Axis 2) to the cross-sealer leading axis (Axis 1). The "Gear synchronization" step is skipped when this switch is set.
boSkipgSync Axis1Pos	BOOL	FALSE	Configuration switch for the travel to the basic position of the cross-sealer leading axis (Axis 1). The "Basic positioning" step is skipped when this switch is set.
boSkipSync Axis1Sync	BOOL	FALSE	Configuration switch for the synchronization of the cross-sealer leading axis (Axis 1) to the machine leading axis. The "Cam synchronization" step is skipped when this switch is set.

Cam controller parameters - sOffsetShiftOutPutCamData and sCuttingOutPutCamData structures

See Cam controller FBLPacOutPutCamControl function block

The output cam functions required for the detection of the cut position and the offset adjustment position can be parameterized with this function block.

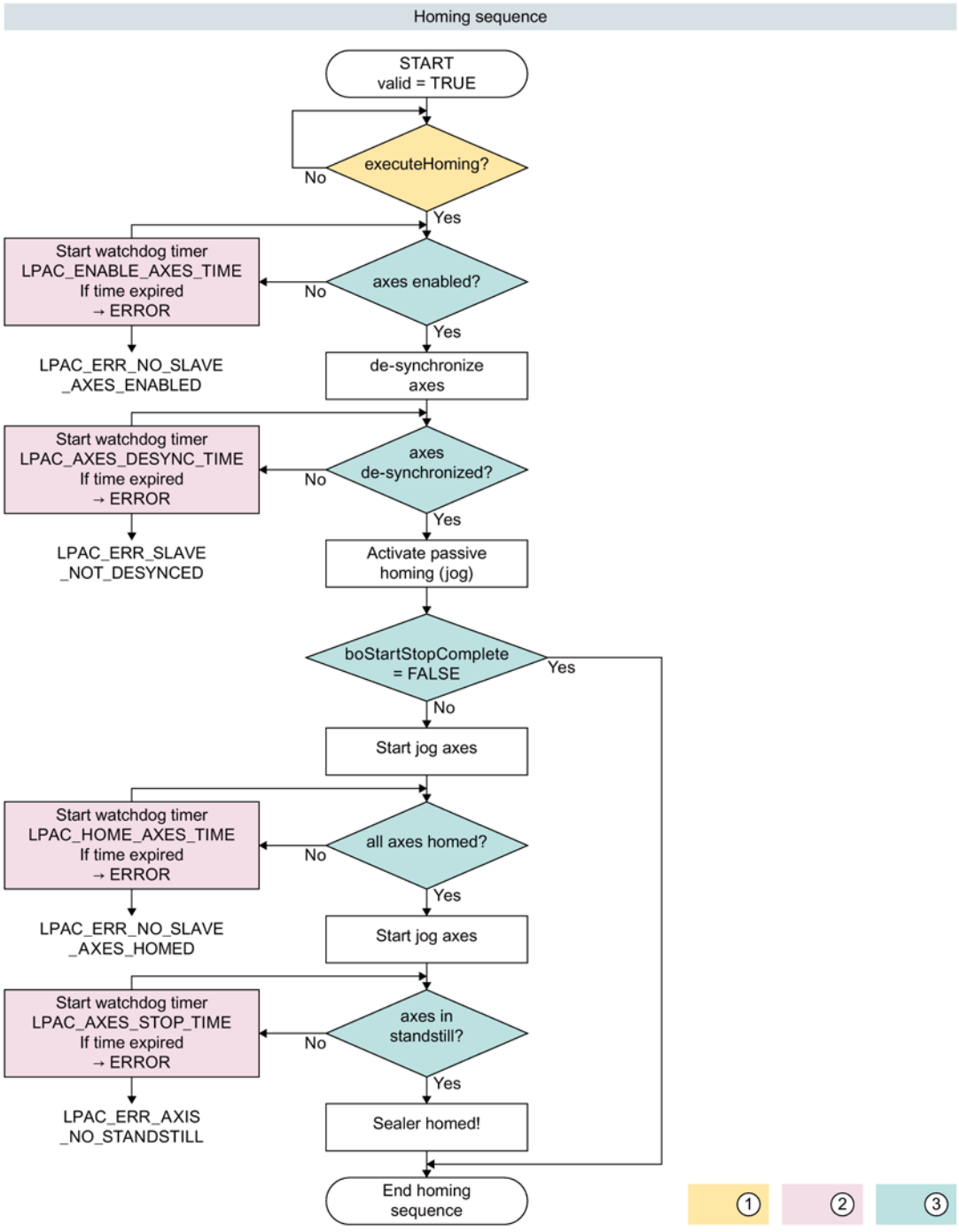
Control of the cross-sealer axes, homing sequence and basic positioning

The block provides a predefined control sequence for control of the subordinate subunits of a typical machine configuration. This allows the block based on the *valid* initial state, to control the subunits (equipment module) normally available on a horizontal tubular bagging machine for starting a chain bag production from the initial state. The block therefore takes over the required control and coordination of the subunits. It controls the homing, any basic positioning and subsequent synchronizing in various sequencers.

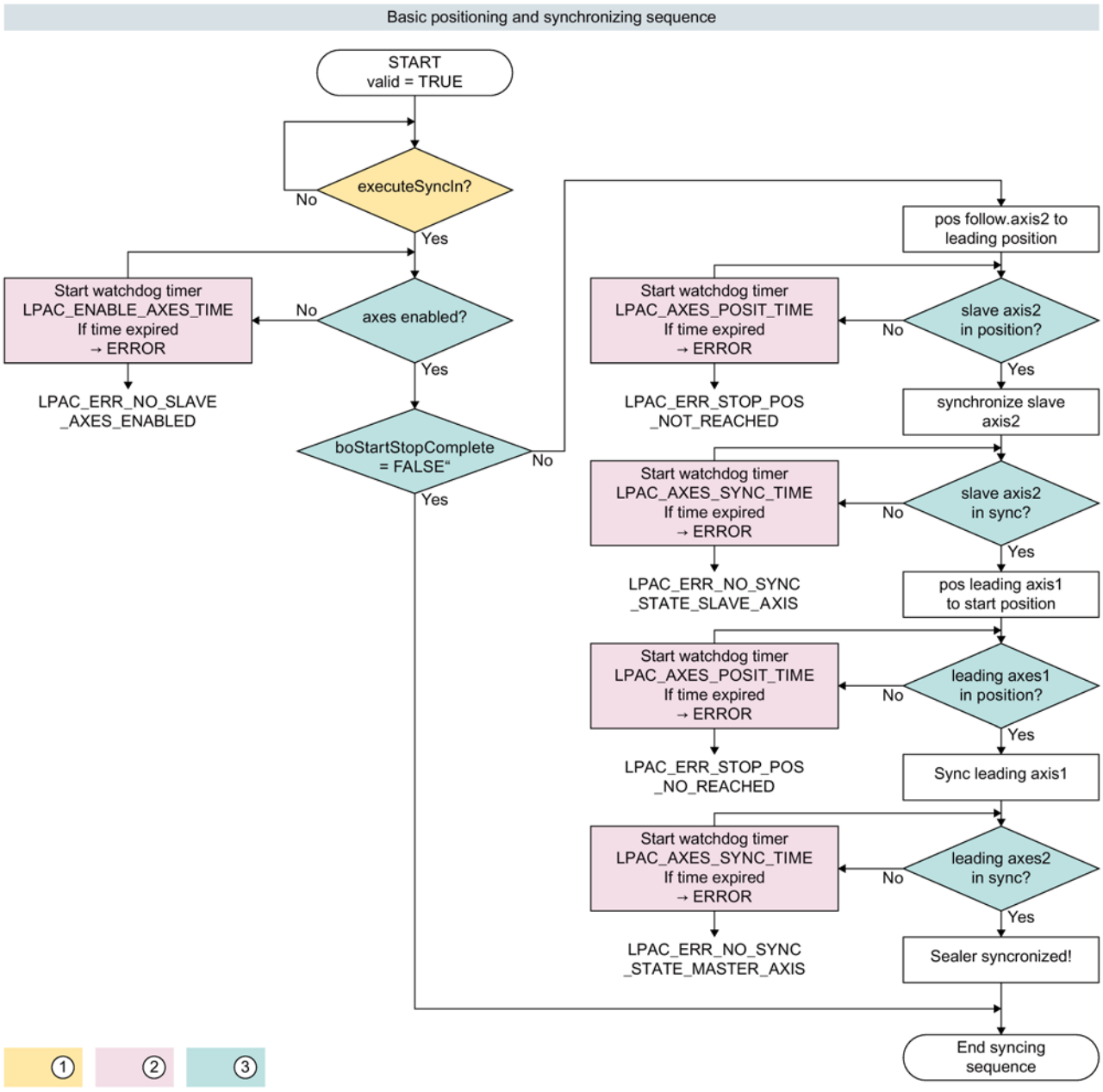
Once the initial state of the subunits is established, the block starts the cyclic offset adjustment for the production of the chain bags.

NOTICE
Damage to property Motions and procedures triggered by unsuitable parameterization can result in damage to property. Through suitable parameterization of the subfunctions passive homing, jogging, basic positioning and synchronization, make sure that triggered motions and procedures do not cause any mechanical damage, e.g. through collision.

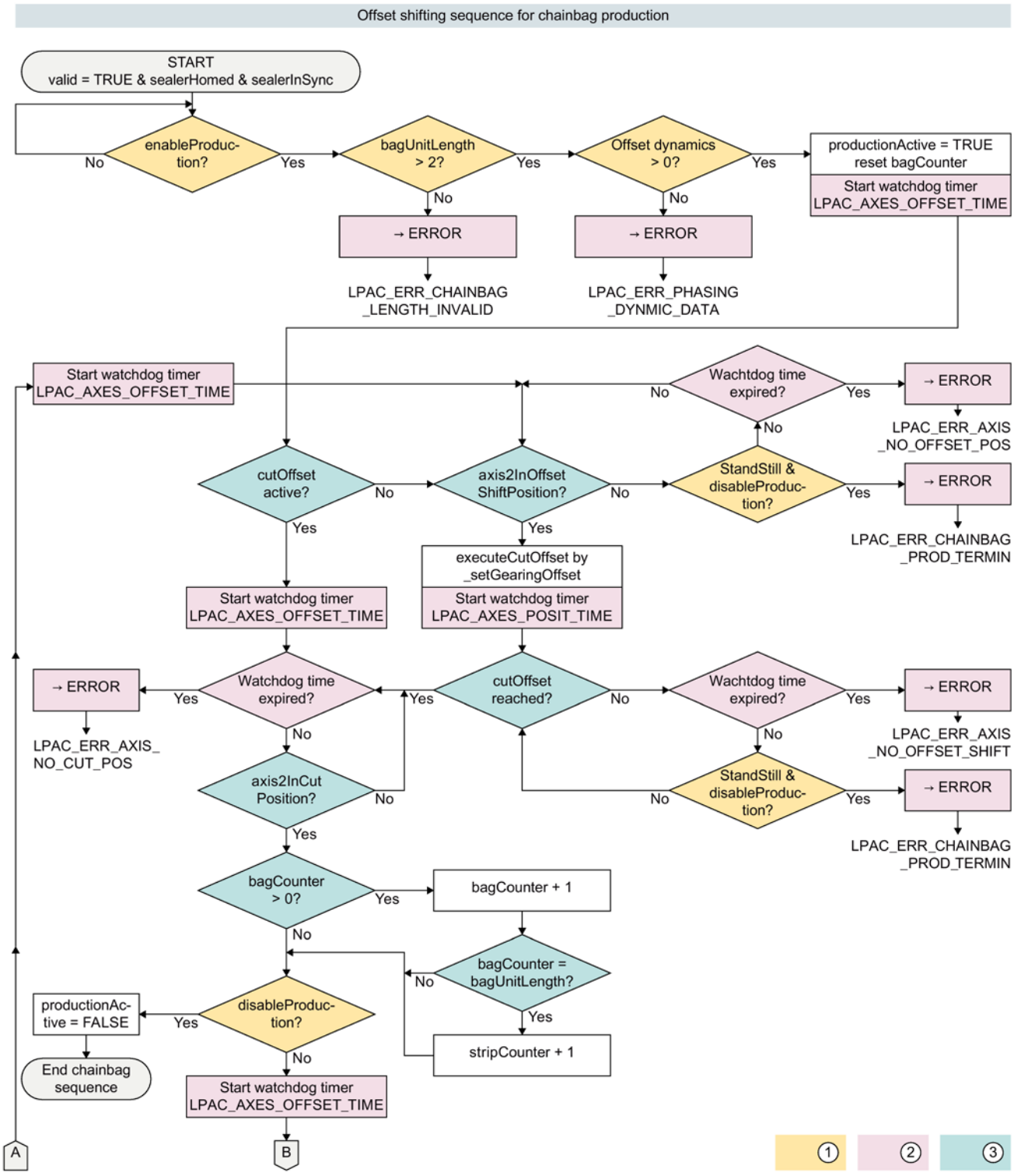
The following figure shows the sequencers for homing and basic positioning.



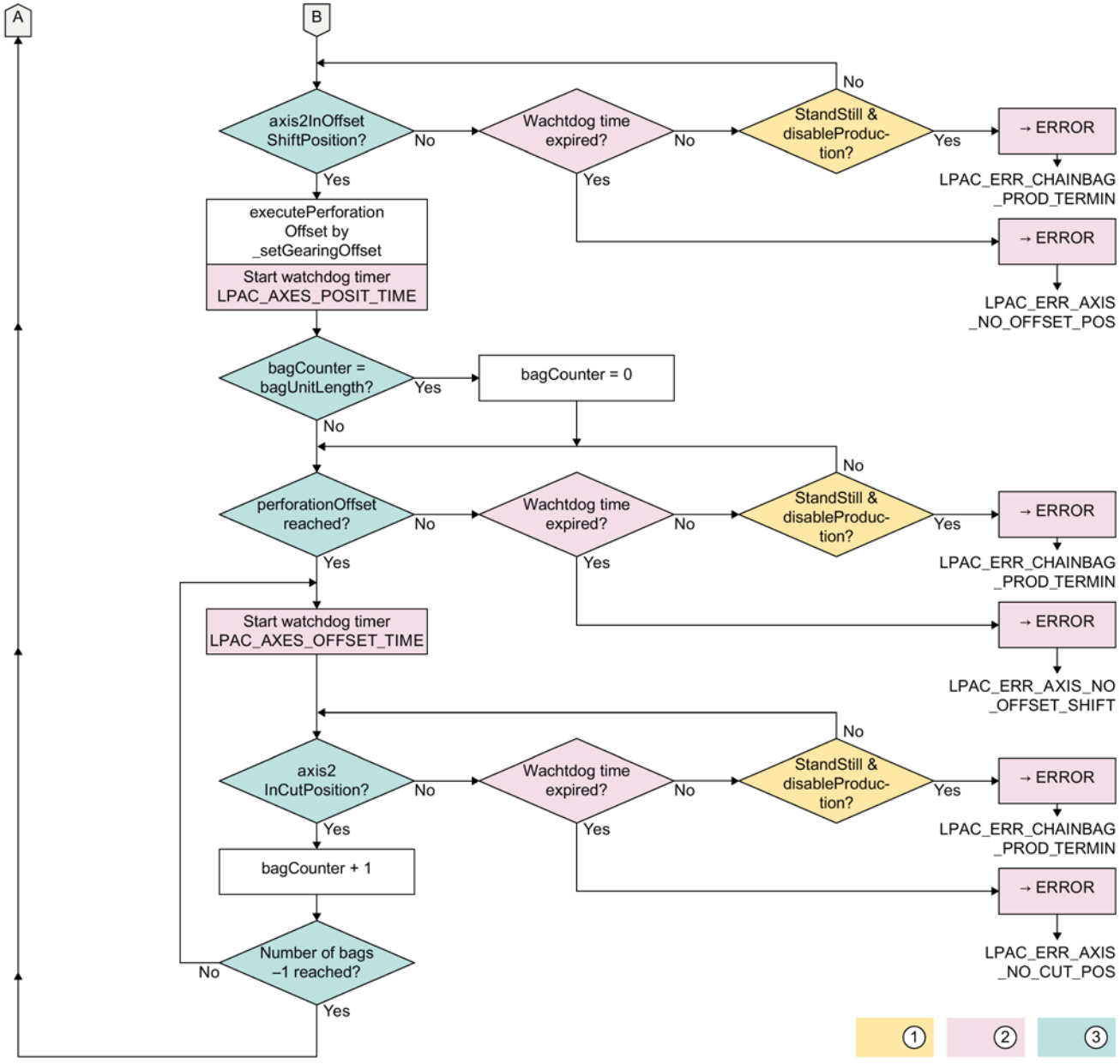
The following figure shows the sequencers for homing and synchronization.



The following figure shows the control sequence for the cyclic offset adjustment.



The following figure shows the continuation of the control sequence for the cyclic offset adjustment.



① Operator intervention possible/expected
② Fault state possible
③ External feedback expected

4.6.3.3 Integration of the module

Technology objects, basic parameterization and block call

Table 4- 158 TO type and basic parameterizations

Axis / technology object	
TO types	Following axes
Version	Rotary axis or linear axis depending on the application
Modulo	For example, 360°
Block call	BackgroundTask

Further blocks and libraries

The **FBLPacCrossSealChainBag** block does not use any further blocks internally.

Error messages

Error number [HEX]	**Meaning**
16#0000	ERR_ERR_NO; error-free
16#8000	LPAC_ERR_INVALID_STEP; internal error, please get in touch with your Siemens contact partner
16#8001	LPAC_ERR_NO_AXIS; no axis TO specified
16#800C	LPAC_ERR_ PHASING_DYNAMIC_DATA; offset adjustment for gearing; invalid velocity and/or acceleration parameters
16#8013	LPAC_ERR_NO_FOLLOWING_OBJECT; no following object specified or leading axis is not a following axis
16#8014	LPAC_ERR_AXIS_NOT_HOMED; cross-sealer axis (axes) not homed within the monitoring time *1
16#8025	LPAC_ERR_NO_SLAVE_AXES_ENABLED; following axis (axes) not enabled within the monitoring time *1
16#8026	LPA_ERR_NO_SYNC_STATE_SLAVE_AXES; following axis not synchronized within the monitoring time *1
16#802B	LPAC_ERR_STOP_POS_NOT_REACHED; following axis (axes) target position not reached within the monitoring time *1
16#802C	LPAC_ERR_SLAVES_NOT_DESYNCED; slave axis (axes) not desynchronized within the monitoring time *1
16#8039	LPAC_ERR_NO_SYNC_STATE_MASTER_AXIS; master axis not synchronized within the monitoring time *1
16#8078	LPAC_ERR_SET_GEAR_OR_CAMMING_OFFSET; error message for the offset adjustment from the setGearingOffset command
16#8200	LPAC_ERR_AXIS_NO_STANDSTILL; cross-sealer axis (axes) no standstill within the monitoring time *1

Error number [HEX]	Meaning
16#8201	LPAC_ERR_AXIS_NO_OFFSET_POS; cross-sealer following axis has not reached the offset shift position within the monitoring time *1
16#8202	LPAC_ERR_NO_CUT_POS; cross-sealer following axis has not reached the cut position within the monitoring time *1
16#8203	LPAC_ERR_CHANINBAG_PROD_TERMIN; chain bag production terminated by operator
16#8204	LPAC_ERR_CHAINBAG_UNIT_LENGTH_INVALID; chain bag length (number of bags per chain bag < 2)
16#8205	LPAC_ERR_AXIS_NO_OFFSET_SHIFT; cross-sealer following axis has not reached the offset within the monitoring time *1

Note

The **pCrossSealerChainBag** program unit is available in the LPac project to illustrate the use of the **FBLPacCrossSealChainBag** block.

This unit contains the programs for the required block calls in the StartupTask and the BackgroundTask.

4.6.4 FBLPacStopperCam – cam switching

4.6.4.1 Technological consideration

The **FBLPacStopperCam** block switches between various cams on a following object for the simple implementation, e.g. for a suspend function on a cross-sealer drive. To do this, the block is passed as many as two cyclic cams, as well as a start, a stop and a standstill cam as TO reference. Starting with the coupled cyclic operation using one of the two cyclic cams, the control of the **executeStop** input at the end of the current cycle switches to the stop cam and so stops the relevant drive at a defined stop position while the master axis continues to run. If no **executeStart** signal is controlled in the next cycle, the block loads the standstill cam automatically at the cycle end. With the control of the **executeStart** signal, the start cam is also loaded at the next cycle end and a switch then made to one of the two cyclic cams.

In addition, with the control of the "changeSyncCam" signal, the respective cyclic cam that is not active is loaded at the next cycle end.

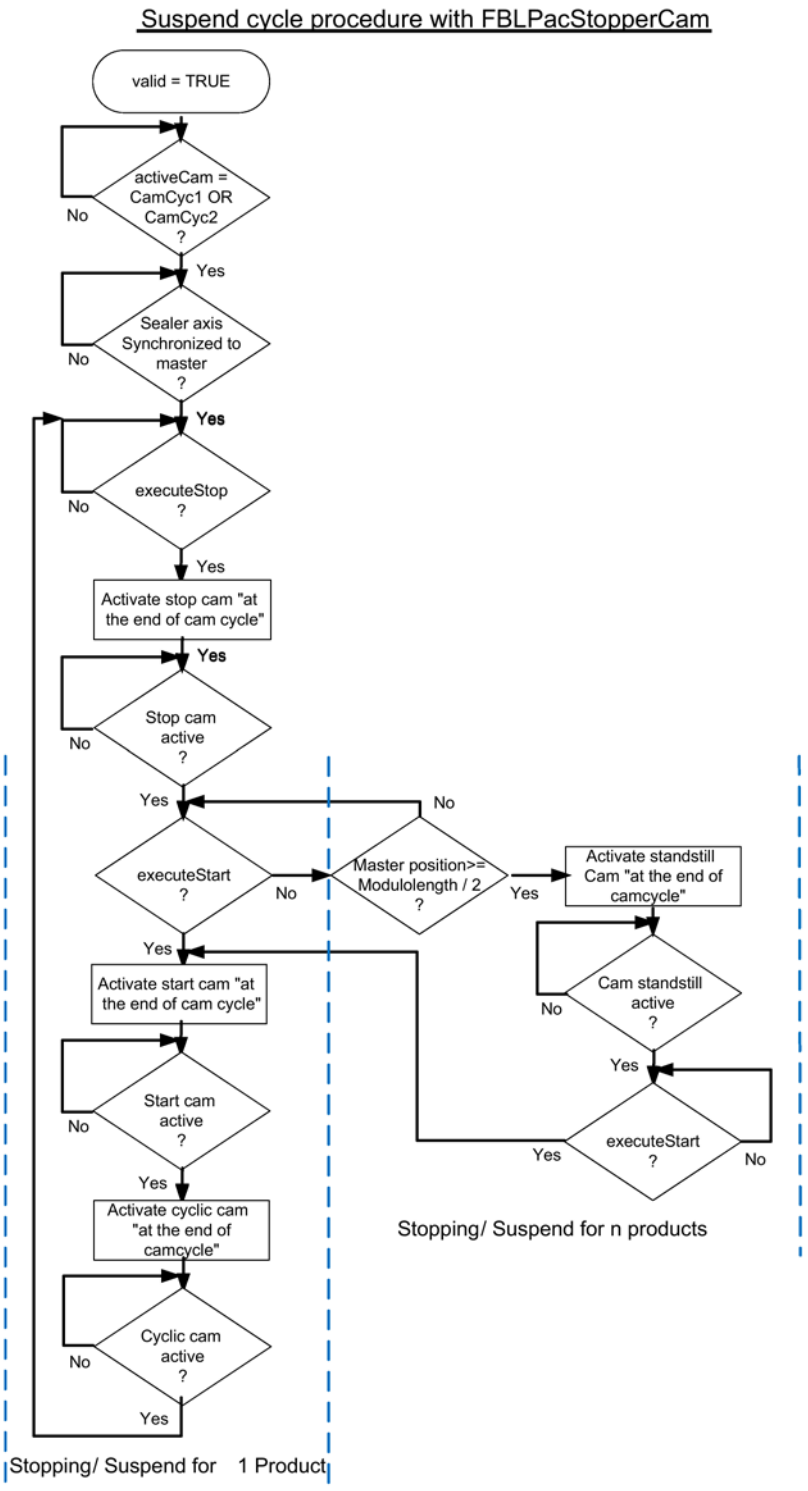


Figure 4-50 Suspend with FBLPac StopperCam sequence

The following figure shows the "Cam change with FBLPacStopperCam" sequence.

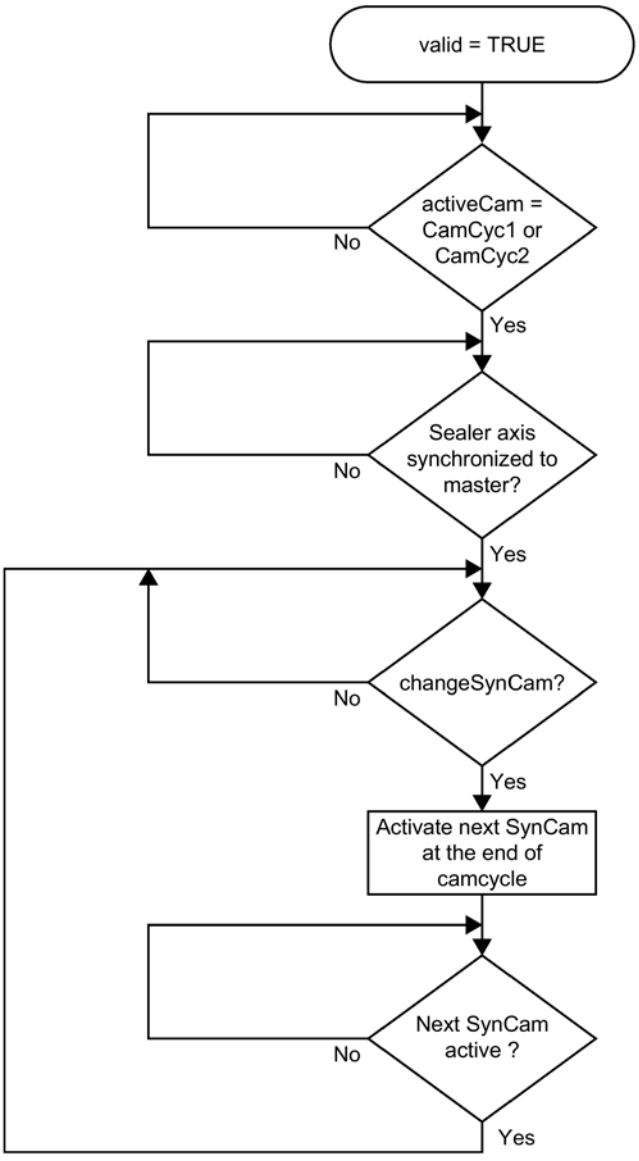
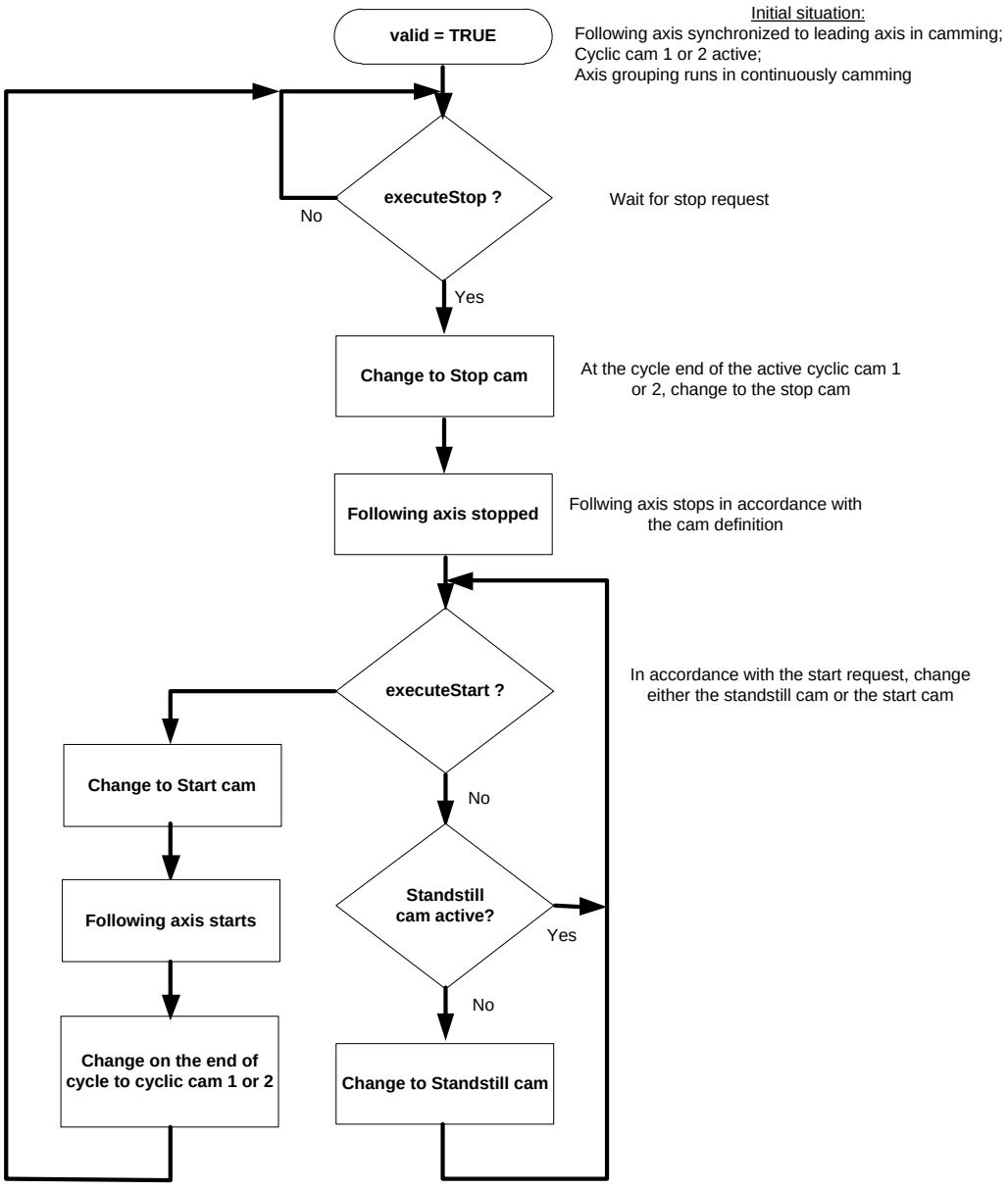


Figure 4-51 "On the fly" change of the cyclic cam sequence

The following figure shows the "Suspend stop cycle" sequence.



① Initial situation:
- Following axis synchronized to leading axis in camming
- Cyclic cam 1 or 2 active
- Axis grouping runs continuously in camming

② Wait for stop request

③ At the cycle end of the active cyclic cam 1 or 2, load the stop curve

④ Following axis stops in accordance with the cam definition

⑤ In accordance with the start request, load either the standstill cam or the start cam

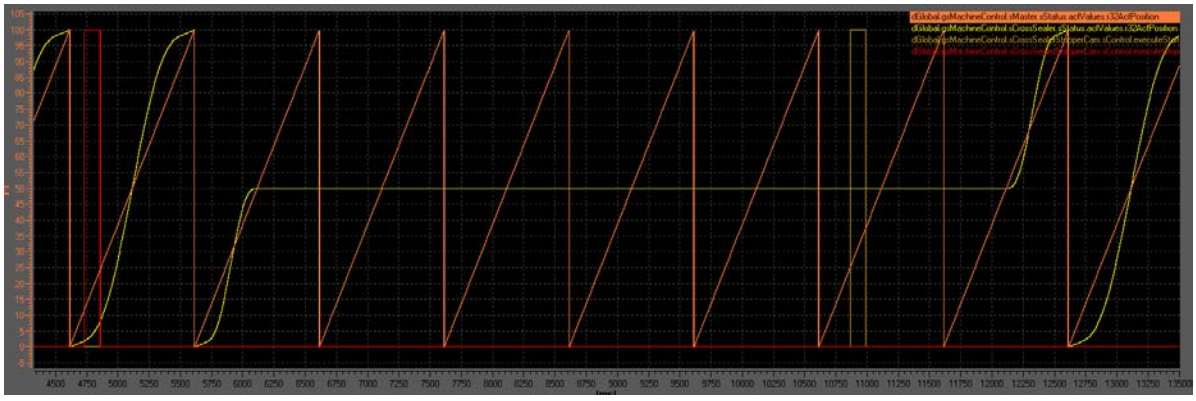


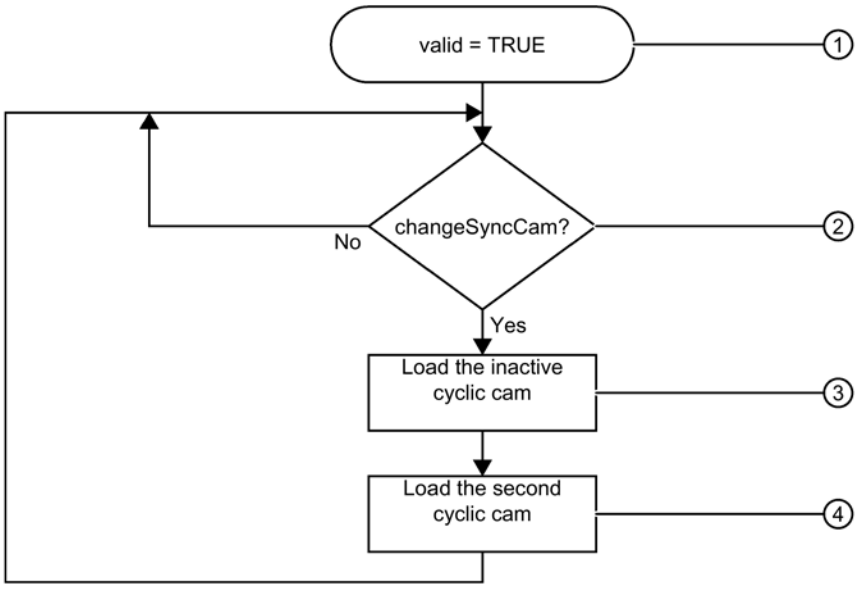
Figure 4-52 Suspend cycle signal characteristic

The following signals are shown:

- Orange: Current master position = dGlobal.gsMachineControl.sMaster.sStatus.actValues.r32ActPosition variable
- Yellow: Current position of the cross-sealer = dGlobal.gsMachineControl.sCrossSealer.sStatus.actValues.r32ActPosition variable
- Green: Current velocity of the cross-sealer = dGlobal.gsMachineControl.sCrossSealer.sStatus.actValues.r32ActVelocity variable
- Red: Stop/suspend cross-sealer signal = dGlobal.gsMachineControl.sCrossSealerStopperCam.sControl.executeStop variable
- Ochre: Start / terminate suspend cross-sealer signal = dGlobal.gsMachineControl.sCrossSealerStopperCam.sControl.executeStart variable

Note that the stop and start signals must be triggered timely before the modulo break because otherwise execution is no longer possible in the next cycle.

The following figure shows the "On the fly cut length change" sequence.



① Initial situation:

- Following axis synchronized to leading axis in camming
- Cyclic cam 1 or 2 active
- Axis grouping runs continuously in camming

When the cut length is changed, by triggering the respective used cam calculation blocks, a second cyclic cam that is not active must be calculated. Only after this has be fully calculated, may the cut length change be triggered.

② Wait for "on the fly" cut length change request

③ At the cycle end of the active cyclic cam, load the inactive cyclic cam

④ Following axis runs in camming to the leading axis with the second cyclic cam

4.6.4.2 Module description

The **FBLPacStopperCam** block provides a general module for cam switching. This makes the block ideal for continuously operating machine modules.

The block can be passed to cams that were created with the **FBLPacCreaCamBoxMotCS**, **FBLPacCreaCamCylCS**, **FBLPacCreaCaCrankLin** function blocks or in some other manner.

Scope of functions

The **FBLPacStopperCam** function block provides the main and auxiliary functions, and possibly options, shown in the following overview.

Function	Description	Implementation
Main function	Suspend function for camming	The activation of the *executeStop and executeStart* signals implements a suspend functionality
Options	Cam change	Control of the *changeSyncCam* signal switches between the two **CamSync1** and **CamSync2** cams at the end of the active cycle and so permits a change of the cut lengths during operation ("on the fly"). Switching is possible only for small changes; large changes of the cut length must be performed during standstill of the synchronous operation unit.

Schematic block view in LAD/FBD

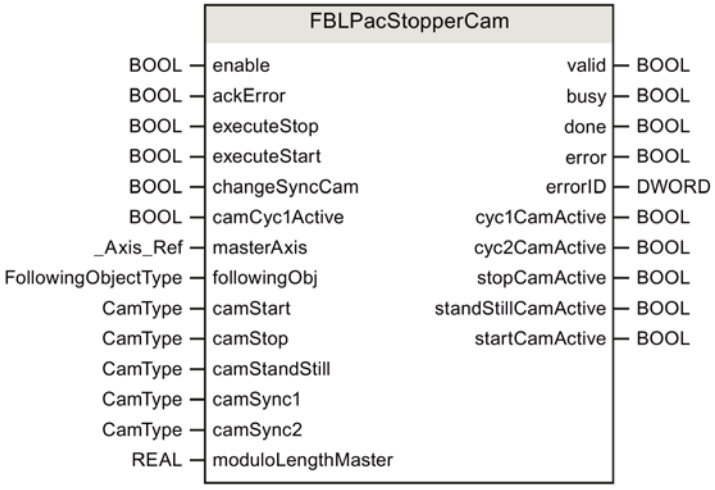


Figure 4-53 FBLPacStopperCam - schematic block view in LAD/FBD

Input and output parameters

Table 4- 159 Input/output signals

Element	P type [1)]	Data type	Initial value	Meaning
enable	IN	BOOL	FALSE	Activation of the FB
ackError	IN	BOOL	FALSE	Acknowledge error
executeStop	IN	BOOL	FALSE	Stop cam axis, load stop curve
executeStart	IN	BOOL	FALSE	Start cam axis, load start curve
changeSyncCam	IN	BOOL	FALSE	Change cyclic cams
camCyc1Active	IN	BOOL	FALSE	Cyclic cam 1 active
masterAxis	IN	Axis_Ref	TO#NIL	Technology object of the master axis

Element	P type [1]	Data type	Initial value	Meaning
followingObj	IN	Following Object Type	TO#NIL	Technology object of the camming
camStart	IN	camType	TO#NIL	Start cam
camStop	IN	camType	TO#NIL	Stop cam
camStandStill	IN	camType	TO#NIL	Standstill cam
camSync1	IN	camType	TO#NIL	Cyclic cam 1
camSync2	IN	camType	TO#NIL	Cyclic cam 2
moduloLength Master	IN	REAL	0.0	Master axis modulo length
valid	OUT	BOOL	FALSE	Block is active
busy	OUT	BOOL	FALSE	Selected functionality being processed
done	OUT	BOOL	FALSE	Selected functionality completed successfully
error	OUT	BOOL	FALSE	Error occurred
errorID	OUT	DWORD	16#0000 _0000	Error code
cyc1CamActive	OUT	BOOL	FALSE	Cyclic cam 1 active
cyc2CamActive	OUT	BOOL	FALSE	Cyclic cam 2 active
stopCamActive	OUT	BOOL	FALSE	Stop - cam active
standStillCam Active	OUT	BOOL	FALSE	Standstill - cam active
startCamActive	OUT	BOOL	FALSE	Start - cam active

[1] Parameter types: IN = input parameter, OUT = output parameter

4.6.4.3 Integration of the module

Technology objects, basic parameterization and block call

Table 4- 160 TO type and basic parameterizations

Axis / technology object	
TO type	Cam axis
Version	Rotary axis or linear axis
Modulo	In accordance with the cam definition
Following object	Following object of the assigned cam axis
Cams	Five cams for the states: • Cyclic operation 1/2 • Stop cycle • Start cycle • Standstill cycle
Block call	BackgroundTask

Further blocks and libraries

The **FBLPacStopperCam** block does not use any further blocks.

Error messages

Error number [HEX]	Meaning
16#0000	LPAC_ERR_NO; error-free
16#8000	ERR_INVALID_STEP; internal error, please get in touch with your Siemens contact partner
16#8010	LPAC_ERR_NO_MASTER_AXIS; no axis TO for leading axis specified
16#8012	LPAC_ERR_NO_CAM, no cam specified
16#8013	LPAC_ERR_NO_FOLLOWING_OBJECT; no following object specified or leading axis is not a following axis
16#8131	LPAC_ERR_ENABLE_CAMMING; error from the _enableCamming command; the return value is equivalent to the *_move* function and can be fetched in hex format in the high word from *errorID*

Note

Tip

The **pStopperCam** program unit is available in the LPac project to illustrate the use of the **FBLPacStopperCam** block.

This unit contains programs for the required block calls in the BackgroundTask.

4.7 Cam calculation blocks

4.7.1 FBLPacCreaCamBoxMotCS - box motion cross-sealer

4.7.1.1 Technological consideration

The **FBLPacCreaCamBoxMotCS** block calculates the cam(s) for traversing a so-called box motion cross-sealer.

Further possible fields of application are:

- Simple flying saw (the SIMOTION flying saw standard application should be used for more complex requirements)
- Cutting edges
- Perforating
- Embossing

The synchronous motion to the foil can be performed using a reversing belt drive or a slider crank. Prerequisite for use of a slider crank, however, is that the synchronous motion to the foil is almost linear. Thus, the path of the circle segment can be approximated using a straight line.

Begin and end of the curve lie in the middle of the synchronous run from the cross-sealer to the foil. The bag start is defined as position 0: This also means that the synchronous start and the synchronous end must be parameterized with negative and positive values, respectively. This also applies to the specification of the mechanical limits.

The block also calculates a second cyclic cam, for example, to perform small changes to the bag length during operation "on the fly" via a cam switchover. The cyclic cam not currently active on the following object is always recalculated. For the simple implementation of a "no product, no bag" function, the block also calculates cams for stopping, starting and for the standstill cycle. Using the **FBLPacStopperCam** block to load the cams dynamically at the associated cycle end on external request allows the simple implementation of a so-callled suspension operation.

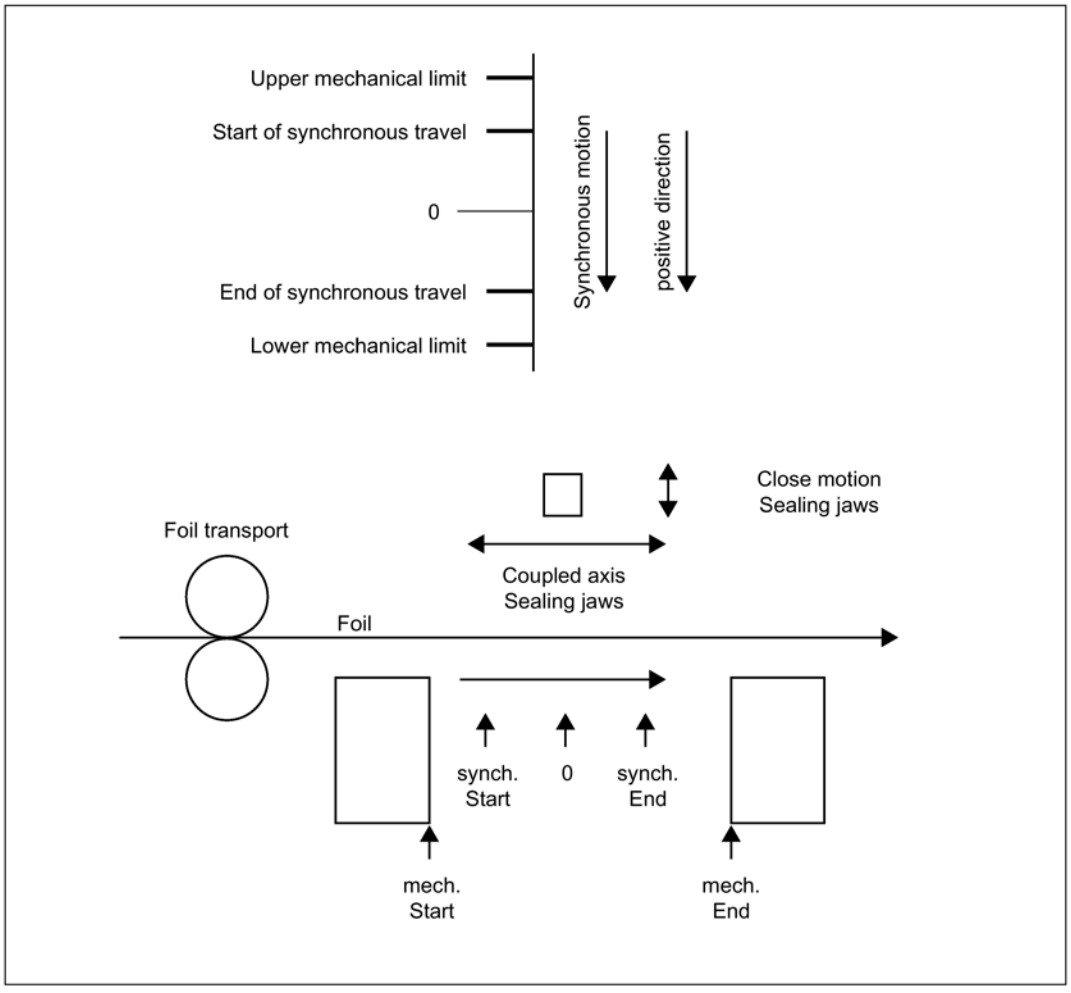


Figure 4-54 Cross-sealer construction in the transport direction, top vertical, bottom horizontal

The cams calculated by the block are shown as an example below.

The following parameters were specified for the calculation:

- Machine leading axis modulo length (master): 1000
- Modulo length: Linear axis
- Bag length: 360 mm
- Synchronous range start: -100 mm
- Synchronous range end: 100 mm
- Mechanical traversing range start: -300 mm
- Mechanical traversing range end: 300 mm
- Stretch factor: 1.0
- Cut position: 1000 (leading axis), 0 mm (cross-sealer)
- Standstill position: 500 (leading axis), 0 mm (cross-sealer)

1. FB_LPacCreateCamBoxMotCS: Cyclic cam for coupled cross-sealer

   Machine leading axis modulo length (master): 1000

   Modulo length: Linear axis

   Bag length: 360 mm

   Synchronous range start: -100 mm

   Synchronous range end: 100 mm

   Mechanical traversing range start: -300 mm

   Mechanical traversing range end: 300 mm

   Stretch factor: 1.0

   Cut position: 1000 (leading axis), 0 mm (cross-sealer)

   Standstill position: 500 (leading axis), 0 mm (cross-sealer)

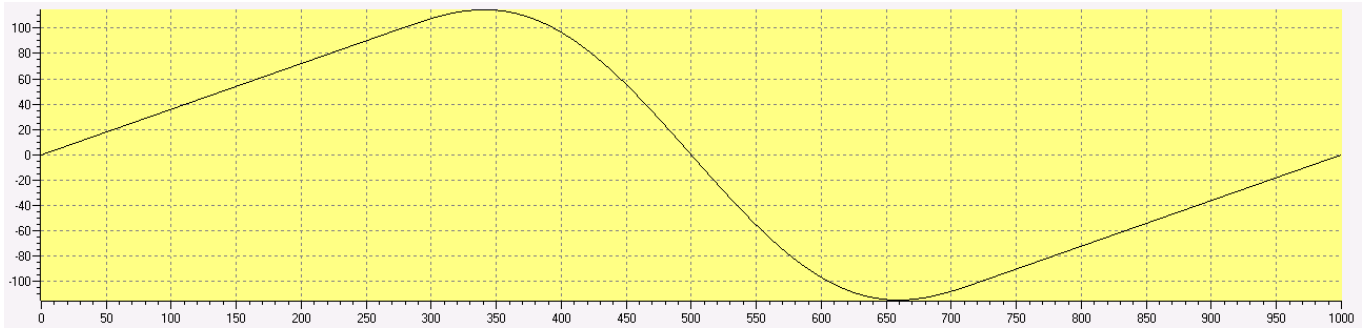


Figure 4-55 Cam for cyclic operation (CamEdit representation)

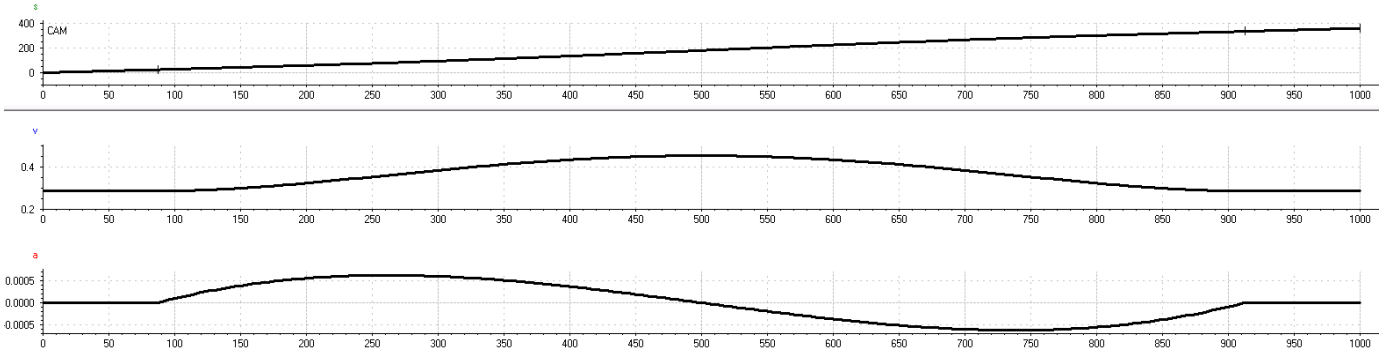


Figure 4-56 Cam for cyclic operation (CamTool representation)

2. FB_LPacCreateCamBoxMotCS: Stop cam for coupled cross-sealer stop position: 500 / 0

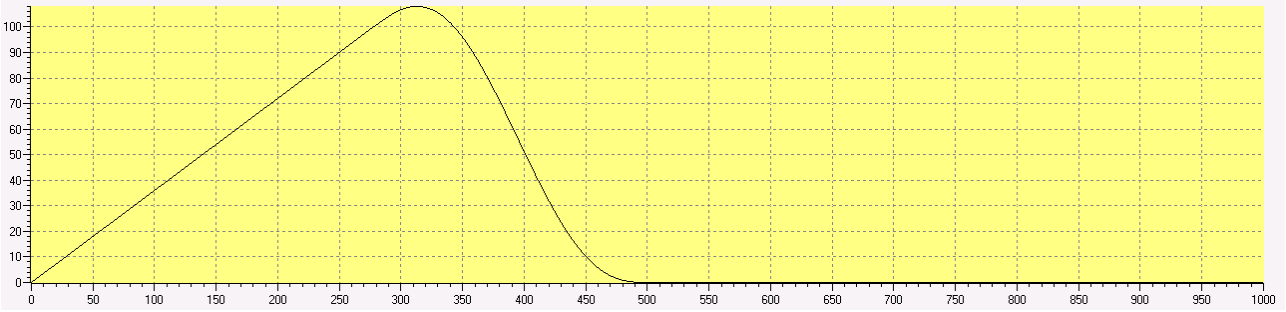


Figure 4-57 Stop cam (CamEdit representation)

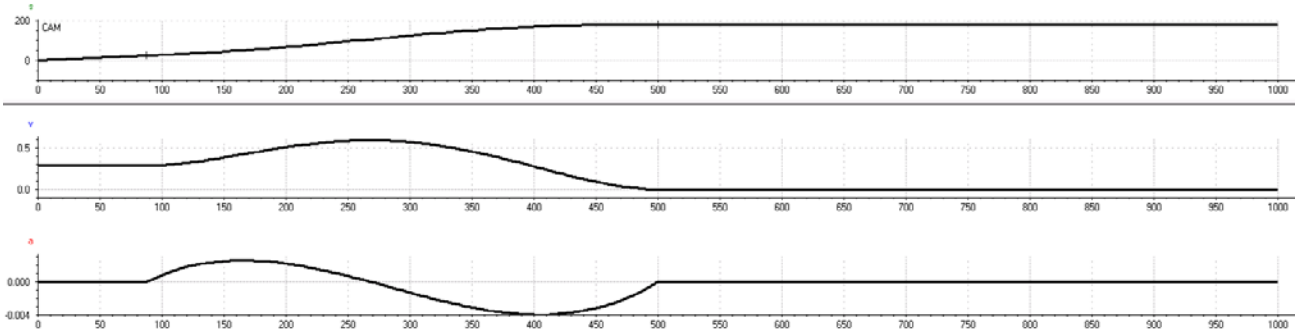


Figure 4-58 Stop cam (CamTool representation)

3. FB_LPacCreateCamBoxMotCS: Standstill cam for coupled cross-sealer standstill at 0

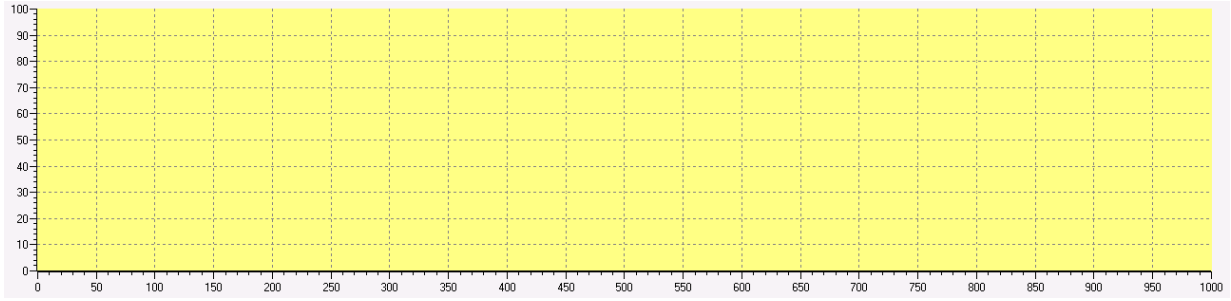


Figure 4-59 Standstill cam (CamEdit representation)

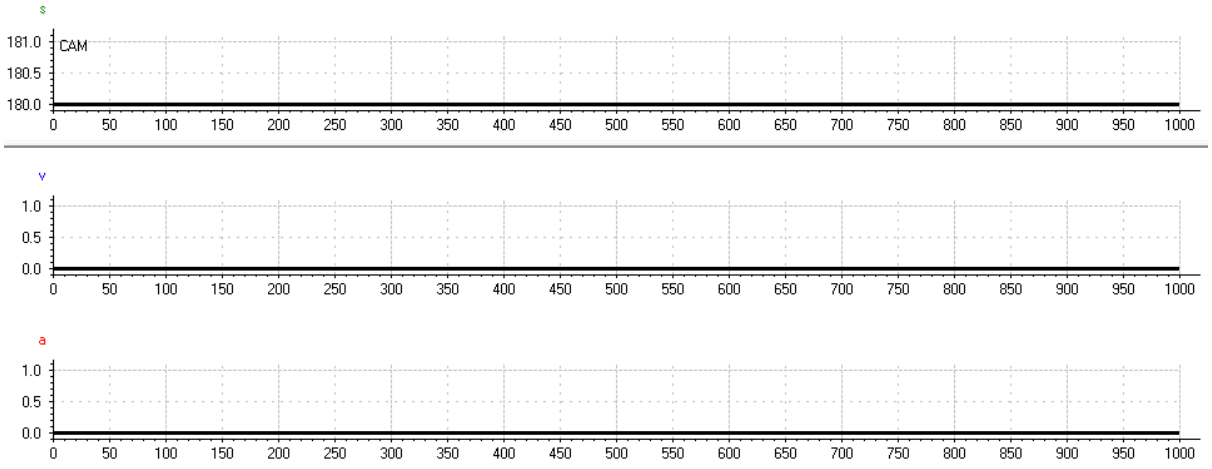


Figure 4-60 Standstill cam (CamTool representation)

4. FB_LPacCreateCamBoxMotCS: Start cam for coupled cross-sealer start position: 500 / 0

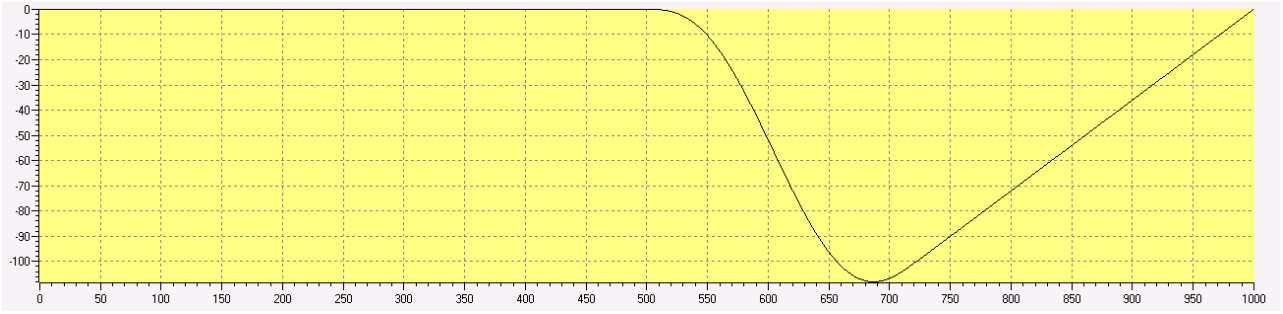


Figure 4-61 Start cam (CamEdit representation)

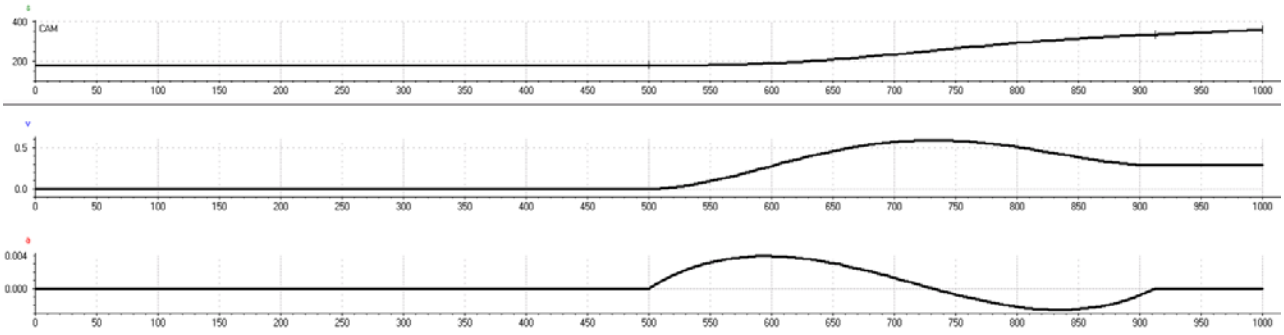


Figure 4-62 Start cam (CamTool representation)

4.7.1.2 Module description

Schematic block view in LAD/FBD

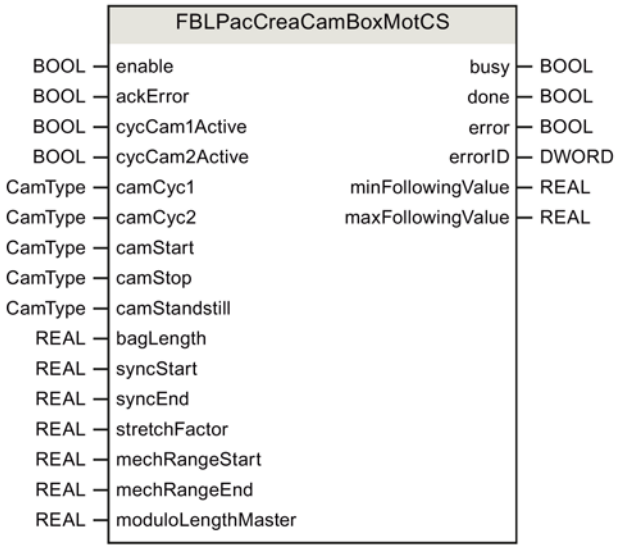


Figure 4-63 FBLPacCreaCamBoxMotCS - schematic block view in LAD/FBD

Input/output parameters

Table 4- 161 Input and output parameters

Element	P type [1]	Data type	Initial value	Meaning
execute	IN	BOOL	FALSE	Perform cam calculation once
ackError	IN	BOOL	FALSE	Acknowledge error
cycCam1Active	IN	BOOL	FALSE	Cyclic cam 1 activated
cycCam2Active	IN	BOOL	FALSE	Cyclic cam 2 activated
camCyc1	IN	CamType	TO#NIL	Technology object of the cyclic cam 1 to be recalculated
camCyc2	IN	CamType	TO#NIL	Technology object of the cyclic cam 2 to be recalculated
camStart	IN	CamType	TO#NIL	Technology object of the start cam to be recalculated
camStop	IN	CamType	TO#NIL	Technology object of the stop cam to be recalculated
camStandStill	IN	CamType	TO#NIL	Technology object of the standstill cam to be recalculated
bagLength	IN	REAL	0.0	Product length (bag length) [mm]
syncStart	IN	REAL	0.0	Start point of the synchronous run to the foil [mm] < 0
syncEnd	IN	REAL	0.0	End point of the synchronous run to the foil [mm] > 0
stretchFactor	IN	REAL	0.0	Scaling factor for the coupled, synchronous motion to the foil [-] >1.0: Synchronous motion faster than the foil <1.0: Synchronous motion slower than the foil, the scaling factor must lie in the range >=0.5 and <=2.0

Element	P type [1]	Data type	Initial value	Meaning
mechRangeStart	IN	REAL	0.0	Smallest possible mechanical position of the jaws cradle (start position) [mm] < 0
mechRangeEnd	IN	REAL	0.0	Largest possible mechanical position of the jaws cradle (end position) [mm] > 0
moduloLengthMaster	IN	REAL	0.0	Modulo length of the master axis (leading axis)
busy	OUT	BOOL	FALSE	Calculation of the cam is active
done	OUT	BOOL	FALSE	Calculation of the cam has completed
error	OUT	BOOL	FALSE	Error occurred
errorID	OUT	DWORD	16#0000_0000	Error code
minFollowingValue	OUT	REAL	0.0	Minimum of the cam, reversal point before synchronous motion [mm]
maxFollowingValue	OUT	REAL	0.0	Maximum of the cam, reversal point after synchronous motion [mm]

[1] Parameter types: IN = input parameter, OUT = output parameter

4.7.1.3 Integration of the module

Technology objects, basic parameterization and block call

Table 4- 162 TO type and basic parameterizations

Cams	Five cams for the states: • Cyclic operation 1/2 • Stop cycle • Start cycle • Standstill cycle
Block call	MotionTask

Further blocks and libraries

The **FBLPacCreaCamBoxMotCS** block uses the following blocks internally.

Block/library	Function
FCLPacCheckProfileBoxMotCS / LPac	Check of the profile parameters
FBLCamHdlCreateCam / LCamHdl	Cam calculation block

Some of the internally used blocks originate from other libraries. For further details, see Section 1.1.4.

Detailed descriptions for the FBLCamHdlCreateCam block are contained in the SIMOTION **LCamHdl** Library User Manual at

https://support.industry.siemens.com/cs/ww/en/view/105644659

Error messages

Error number [HEX]	Meaning
16#0000	LPAC_ERR_NO; error-free
16#4000…4019	Error message of the FBLCamHdlCreateCam block; see **LCamHdl** SIMOTION Library Manual
16#8000	ERR_INVALID_STEP; internal error, please get in touch with your Siemens contact partner
16#8006	LPAC_ERR_BAGLENGTH_ZERO; bag length <= 0
16#8012	LPAC_ERR_NO_CAM; at least one cam object is not specified
16#80B0	LPAC_ERR_SYNCEND_OUT_OF_RANGE; end of synchronous range outside the permissible range
16#80B1	LPAC_ERR_SYNCSTART_OUT_OF_RANGE; start of synchronous range outside the permissible range
16#80B2	LPAC_ERR_STRETCH_FACTOR_OUT_OF_RANGE; stretch factor < 0.5 or > 2.0
16#80B3	LPAC:ERR_SYNC_TO_LONG (syncEnd + syncStart >= baglength); synchronous range too long
16#80BD	LPAC_ERR_MODULO_LENGHT_MASTER; master axis modulo length ≤ 0
16#80C0	LPAC_ERR_RANGE_END_BEFORE_START; start of synchronous range ≤ end of synchronous range
16#80C1	LPAC_ERR_CAM_OUT_OF_MECH_RANGE; cams outside the mechanical limits
16#80C3	LPAC_ERR_EXECUTE_DURING_RESET; Execute requested while error acknowledgment active
16#80C4	LPAC_ERR_GET_CAM_FOLLOWING_MIN_MAX; error of the _getCamFollowingMinMax function; the return value is equivalent to the *_move* function and can be fetched in hex format in the high word from *errorID*

Note

Tip

The **pCreaCamBoxMotCS** program unit is available in the LPac project to illustrate the use of the **FBLPacCreaCamBoxMotCS** block.

This unit contains programs for the required block calls in the MotionTask.

4.7.2 FBLPacCreaCamCylCS – rotary, cylindrical cross-sealer

4.7.2.1 Technological consideration

The **FBLPacCreaCamCylCS** function block coordinates the calculation of cams for a rotating, cylindrical cross-sealer (rotating blade), e.g. for continuously operating horizontal tubular bagging machines. The block calculates the cams required for the cyclic operation. To do this - depending on the bag length, synchronous range length, cylinder diameter and number of blades at the cylinder circumference input parameters as well as the modulo length of the master and the slave axes - cams are calculated that consist of two synchronous straights (together they represent the cutting area) and a format range. The format area connects the two synchronous ranges with a 5th order polynomial. It is assumed that the cut is always made at position 0 of the cross-sealer. The synchronous range to be parameterized, in which the peripheral speed of the cross-sealer is equal to the foil speed, is always divided symmetrically at the cut angle. In accordance with these boundary conditions, a cam is always calculated with a slave modulo length between two blades; this allows a "No product, no bag" function to be implemented easily for which the cross-sealer for a missing product in the center position, namely can always be brought to a standstill at 180º. The block also allows a second cyclical cam to be calculated, for example, to perform small changes to the bag length during operation "on the fly" via a cam switchover. The cyclical cam disk not currently active on the synchronous object is always recalculated. For the simple implementation of a "no product, no bag" function, the block also calculates cams for stopping, starting and for the standstill cycle. Using the **FBLPacStopperCam** block to load the cams dynamically at the associated cycle end on external request allows the simple implementation of a so-callled suspension operation.

4.7.2.2 Module description

In the synchronous range, the curves are interpolated so that the peripheral speed is the same as the foil speed. The system calculates the peripheral speed outside the synchronous range depending on the ratio of the circumference to the bag length.

Note

The circumference or the distance between two sealing jaws should be of the same order of magnitude as the bag length.

This allows a curve to be generated with smallest possible accelerations. If the circumference is significantly smaller than the bag length, the block generates a curve with a wait phase during which the cross-sealer is brought to a stop. A speed overshoot occurs before and after the synchronous run; the system calculates this speed overshoot using transitions optimized to the fullest extent.

The calculation of the cam assumes that one slave modulo length (normally always 360°) elapses between two blades.

If more than one blade is present at the cross-sealer circumference, the gear ratio of the load gear must be adapted appropriately (only possible at standstill for a non-enabled axis).

This has the advantage that a standstill of the cross-sealer always occurs at the same position, irrespective of the number of blades distributed symmetrically at the circumference.

The block can calculate two cams alternatively for the continuous running of the cross-sealer. Consequently, in conjunction with the FBLPacCrossSealCam cross-sealer block, an "on the fly" change of the bag length in "small" incremental steps is possible during running operation. When passed a new bag length, the FBLPacCreaCamCylCS block always recalculates the cam not currently being "accessed". Once the calculation has been performed, this is informed to the FBLPacMasterCont master block via the "done" output, which by the appropriate activation of the FBLPacCrossSealCam cross-sealer block loads the newly calculated bag at cycle end so that the cross-sealer cuts the newly changed bag length.

To implement a "no product, no bag" function, the FBLPacCreaCamCylCS block calculates three further cams for stopping, for a standstill cycle and for a start cycle of the cross-sealer. These three additional cams can also be passed to the FBLPacCrossSealCam cross-sealer block. This block then provides a control signal for loading the appropriate cam and thus for stopping, bringing to a standstill and starting the cross-sealer from the running production.

The synchronous range to be parameterized is divided symmetrically at the blade position 0º.

The stop/start position is always specified with 180º or for 360º deviating modulo length of the cross-sealer to half the modulo length.

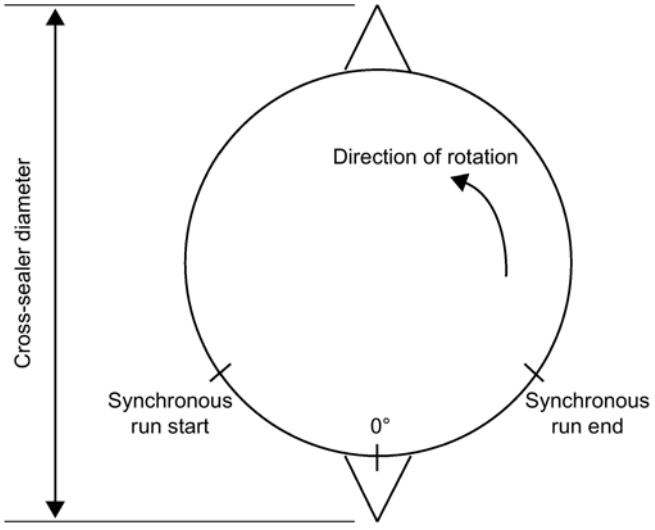


Figure 4-64 Principle of the rotating cross-sealer

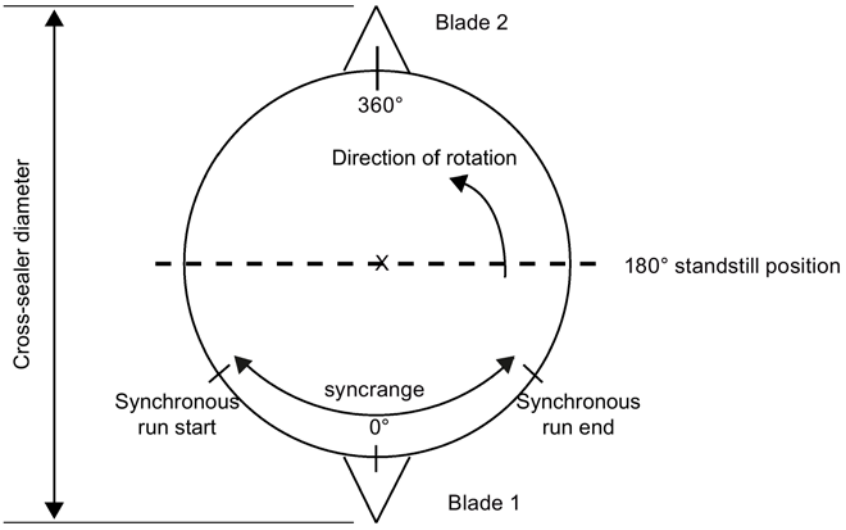


Figure 4-65 Principle of the rotating cylindrical cross-sealer with two blades

Before calculating the cam, the following input parameters are checked for plausibility.

When the

bagLength	<= 0.0
syncRange	< 0° or > 180° or >= moduloLengthSlave
diameter	<= 0.0
StretchFactor	< 0.9 or > 1.1
NumberOfJaws	<= 0

moduloLengthSlave, moduloLengthMaster <= 0

then an appropriate error number is output.

The following condition must also be fulfilled:

(crossSealSyncRange / (2 * moduloLengthSlave)) * circumferential length between two blades < 0.95 * baglength / 2

If an error occurs during the check, the cam is not calculated.

The cams calculated by the block are shown as an example below.

The following parameters were specified for the calculation:

- Machine leading axis modulo length (master) 1000
- Cross-sealer modulo length 360˚
- Bag length 360 mm
- Cross-sealer diameter 200 mm
- Number of blades 1
- Stretch factor 1.0
- Cut position 1000 (leading axis), 0˚ (cross-sealer)

- Standstill position 500 (leading axis), 180˚ (cross-sealer)
- Synchronous range 10˚

1. FB_LPacCreateCamCylCS cyclic cam for rotating cross-sealer

   Master cycle length 1000

   Cross-sealer cycle length 360 degrees, bag length 360

   Stretch factor 1, cut at 1000 / 360

   Standstill position at 500 / 180

   SyncRange 10 synchronous range 995 … 5

   CrossSealDiameter 200

   numberOfJaws 1

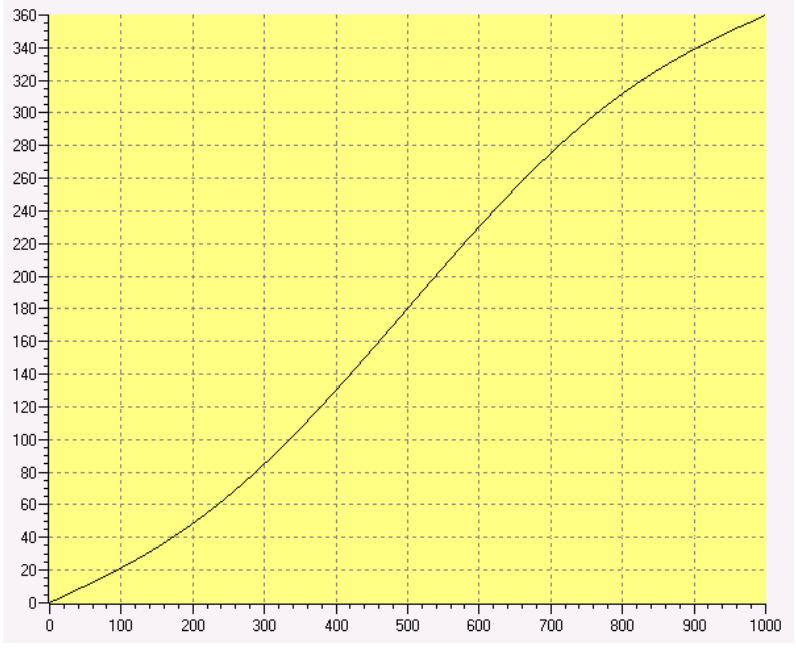


Figure 4-66 Cam for cyclic operation (CamEdit representation)

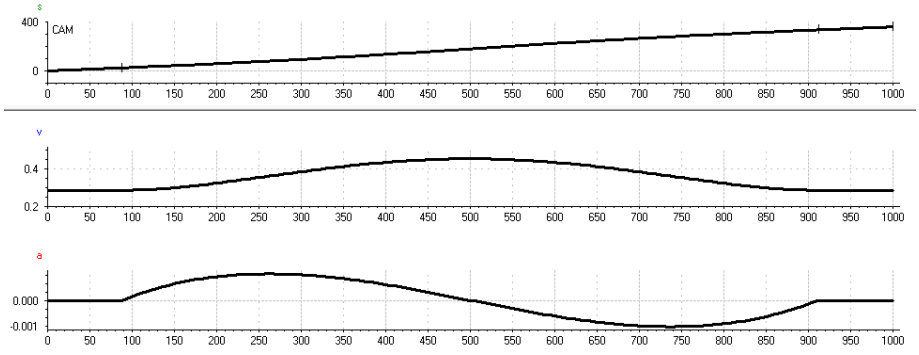


Figure 4-67 Cam for cyclic operation (CamTool representation)

2. FB_LPacCreateCamCylCS: Stop cam for rotating cross-sealer stop position: 500 / 180 degrees

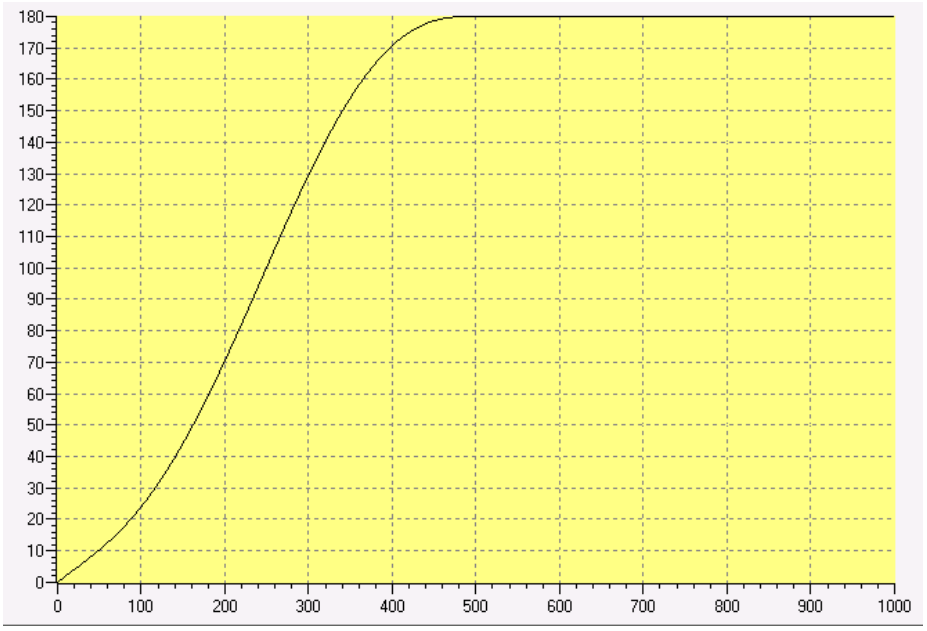


Figure 4-68 Stop cam (CamEdit representation)

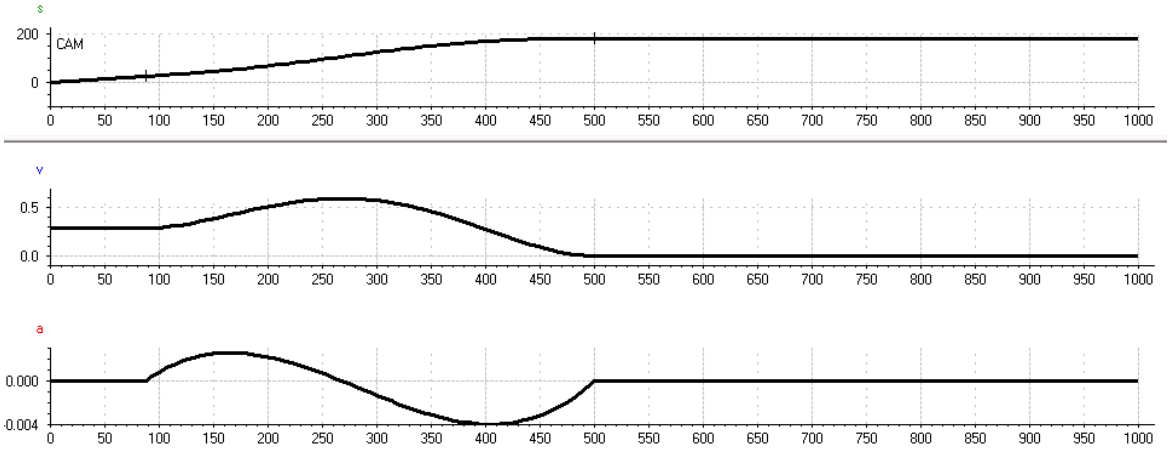


Figure 4-69 Stop cam (CamTool representation)

3. FB_LPacCreateCamCylCS: Standstill cam for rotating cross-sealer standstill at 180 degrees

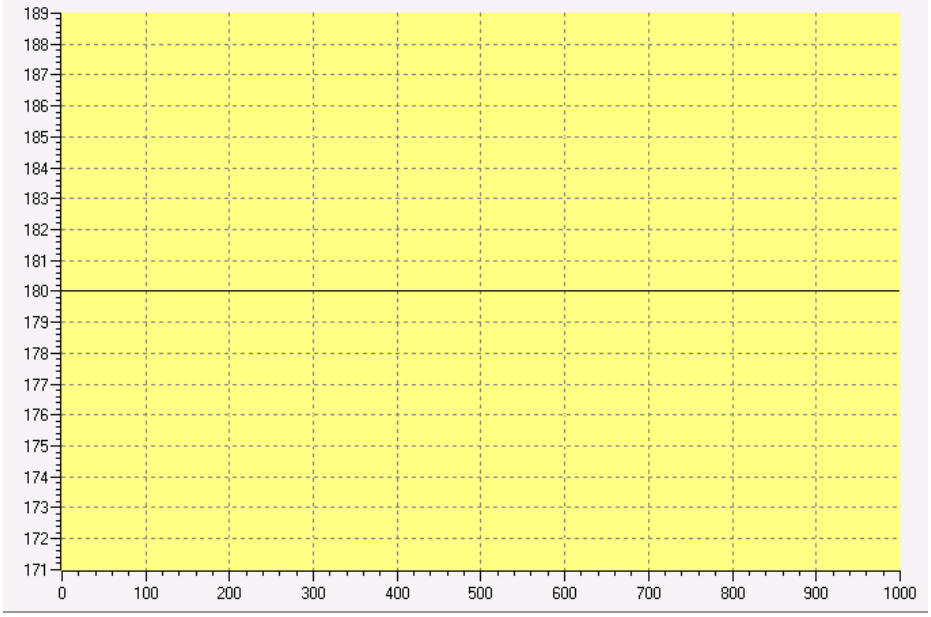


Figure 4-70 Standstill cam (CamEdit representation)

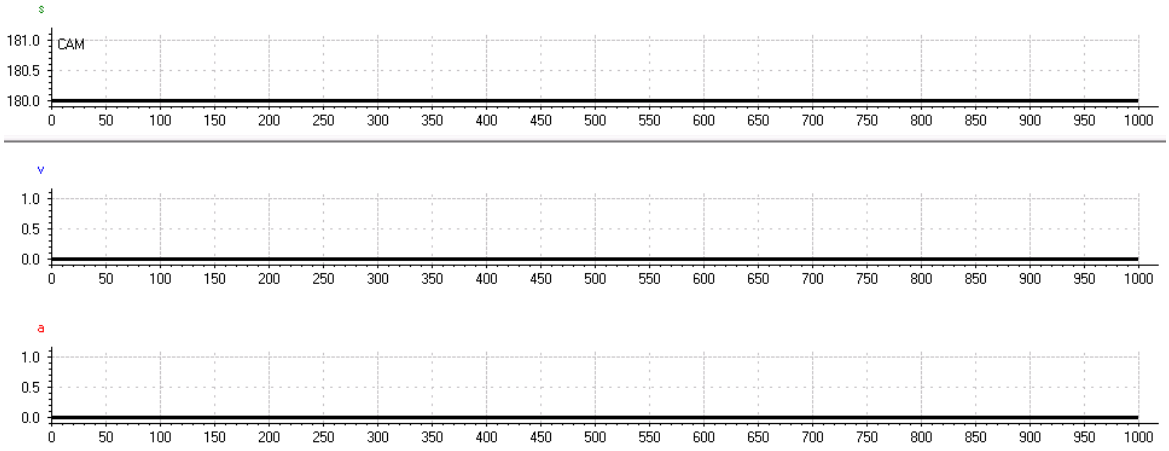


Figure 4-71 Standstill cam (CamTool representation)

4. FB_LPacCreateCamCylCS: start cam for rotating cross-sealer start position: 500 / 180 degrees

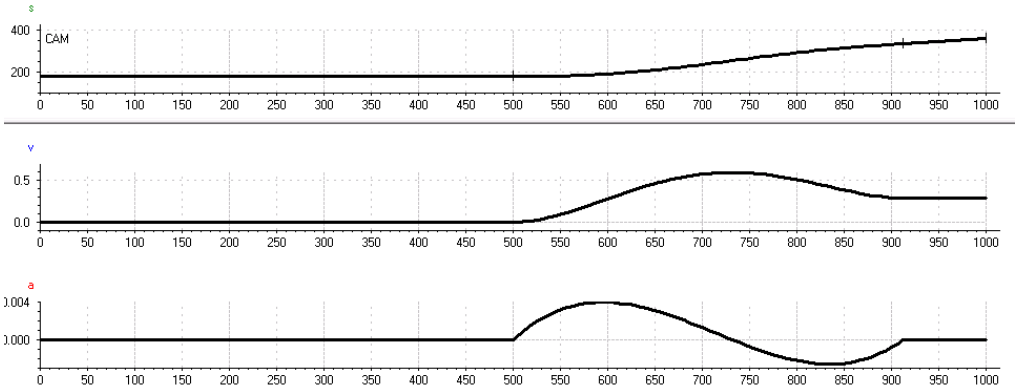


Figure 4-72 Start cam (CamTool representation)

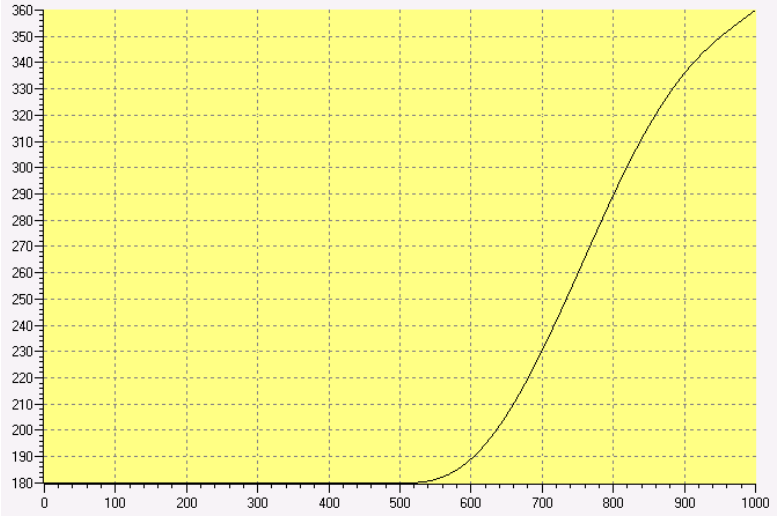


Figure 4-73 Start cam (CamEdit representation)

Schematic block view in LAD/FBD

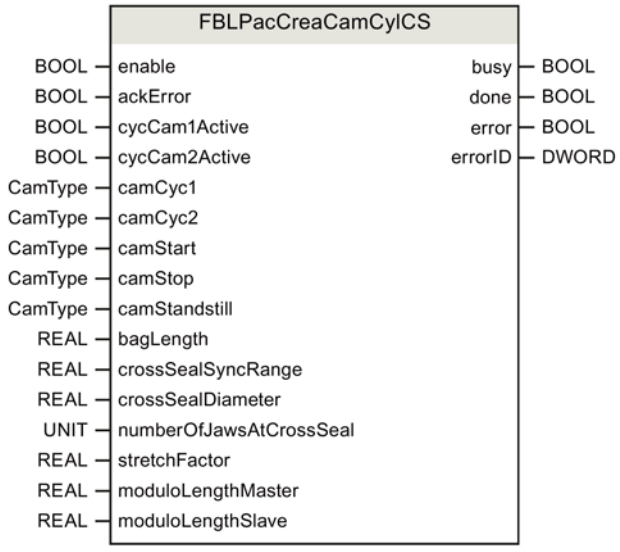


Figure 4-74 FBLPacCreaCamCylCS - schematic block view in LAD/FBD

Input and output parameters

Table 4- 163 Input and output parameters

Element	P type [1]	Data type	Initial value	Meaning
execute	IN	BOOL	FALSE	Start of the cam calculation
ackError	IN	BOOL	FALSE	Acknowledge error
cycCam1Active	IN	BOOL	FALSE	Cyclic cam 1 active in synchronous operation
cycCam2Active	IN	BOOL	FALSE	Cyclic cam 2 active in synchronous operation
camCyc1	IN	CamType	TO#NIL	Technology object of the first cyclic cam to be calculated
camCyc2	IN	CamType	TO#NIL	Technology object of the second cyclic cam to be calculated
camStart	IN	CamType	TO#NIL	Technology object of the start cam to be calculated
camStop	IN	CamType	TO#NIL	Technology object of the stop cam to be calculated
camStandStill	IN	CamType	TO#NIL	Technology object of the standstill cam to be calculated
bagLength	IN	REAL	0.0	Bag length [mm]
crossSealSyncRange	IN	REAL	0.0	Synchronous range of the cross-sealer to the foil [°] is divided symmetrically at the blade position
crossSealDiameter	IN	REAL	0.0	Cylindrical sealer diameter including cutters [mm]
numberOfJaws AtCrossSeal	IN	UINT	1	Number of sealing jaws on the cross-sealer

Element	P type [1]	Data type	Initial value	Meaning
stretchFactor	IN	REAL	1.0	Scaling factor for the coupled, synchronous motion to the foil 1.0: no scaling >1.0: synchronous motion faster than the foil <1.0: synchronous motion slower than the foil; permitted range of values 0.9…1.1, i.e. ± 10%
moduloLengthMaster	IN	REAL	1000.0	Modulo length of the master
moduloLengthSlave	IN	REAL	360.0	Modulo length of the slave (cross-sealer)
busy	OUT	BOOL	FALSE	Calculation of the cam active
done	OUT	BOOL	FALSE	Calculation of the cam completed
error	OUT	BOOL	FALSE	Error occurred
errorID	OUT	DWORD	16#0000_0000	Error code

[1] Parameter types: IN = input parameter, OUT = output parameter

4.7.2.3 Integration of the module

Technology objects, basic parameterization and block call

Table 4- 164 TO type and basic parameterizations

Cams	Five cams for the states: • Cyclic operation 1/2 • Stop cycle • Start cycle • Standstill cycle
Block call	MotionTask

Further blocks and libraries

The **FBLPacCreaCamCylCS** block uses the following blocks internally.

Block/library	Function
FCLPacCheckProfileCylCS / LPac	Check of the profile parameters
FBLCamHdlCreateCam / LCamHdl	Cam calculation block

Some of the internally used blocks originate from other libraries. For further details, see Section 1.1.4.

Detailed descriptions for the FBLCamHdlCreateCam block are contained in the SIMOTION **LCamHdl** Library User Manual at

https://support.industry.siemens.com/cs/ww/en/view/105644659

Error messages

Error number [HEX]	Meaning
16#0000	LPAC_ERR_NO; error-free
16#4000…4019	Error message of the FBLCamHdlCreateCam block; see **LCamHdl** SIMOTION Library Manual
16#8000	ERR_INVALID_STEP; internal error, please get in touch with your Siemens contact partner
16#8005	LPAC_ERR_MODULO_LENGTH; master or slave modulo length <= 0
16#8006	LPAC_ERR_BAGLENGTH_ZERO; bag length <= 0
16#8012	LPAC_ERR_NO_CAM; at least one cam object is not specified
16#80B2	LPAC_ERR_STRETCH_FACTOR_OUT_OF_RANGE; stretch factor <0.9 or >1.1
16#80B4	LPAC_ERR_SYNCRANGE_OUT_OF_RANGE; synchronous range either < 0º or >180º or > cross-sealer modulo length
16#80B5	LPAC_ERR_DIAMETER_OUT_OF_RANGE; cross-sealer diameter <= 0.0
16#80B6	LPAC_ERR_NUMBER_OF_JAWS; blade count <= 0
16#80B7	LPAC_ERR_IN_PARAMETER_CYLINDER_PROFILE; synchronous range relative to the bag length too large
16#80B8	LPAC_ERR_CROSS_SEALER_RUNS_REVERSAL; calculated cam results in reverse motion of the cross-sealer
16#80C3	LPAC_ERR_EXECUTE_DURING_RESET; block started with execute during error acknowledgment with ackError
16#80C4	LPAC_ERR_GET_CAM_FOLLOWING_MIN_MAX; error of the *_getCamFollowingMinMax* function; the return value is equivalent to the *_move* function and can be fetched in hex format in the high word from *errorID*

Note

Tip

The **pCreaCamCylCS** program unit is available in the LPac project to illustrate the use of the **FBLPacCreaCamCylCS** block.

This unit contains programs for the required block calls in the MotionTask.

4.7.3 FBLPacCreaCamCrankLin - slider crank with with constant linear velocity

4.7.3.1 Technological consideration

The **FBLPacCreaCamCrankLin** function block coordinates the calculation of cams for a slider-crank mechanical system. The cam is calculated in such a way that a constant linear mechanical feedrate results in the desired synchronous range. A box motion cross-sealer with slider crank as mechanical coupling element can be implemented with this. Further applications are, for example, axes that have a slider crank as coupling element, and that also require a constant feedrate at the output of the slider crank when the leading axis velocity is also constant in the synchronous range. Centric and non-centric slider cranks are permitted. The exact definition and description of the slider cranks and definition parameters are described in Section FBLPacTranslationToRotation or FBLPacRotationToTranslation (Page 266).

4.7.3.2 Module description

In the synchronous range, the curves are interpolated so that the linear feedrate is constant at constant leading axis velocity. The system calculates the velocity outside the synchronous range.

The cams calculated by the block are shown as an example below.

The following parameters were specified for the calculation:

- Machine leading axis modulo length (master): 1000
- Following axis modulo length: 360°

1. **FBLPacCreaCamCrankLin:** Cyclic cam for slider crank

   Leading axis cycle length: 1000

   Following axis cycle length: 360 degrees: Bag length 360

   Standstill position at 500 / 180

   SyncRange: 10 Synchronous range: 995 … 5

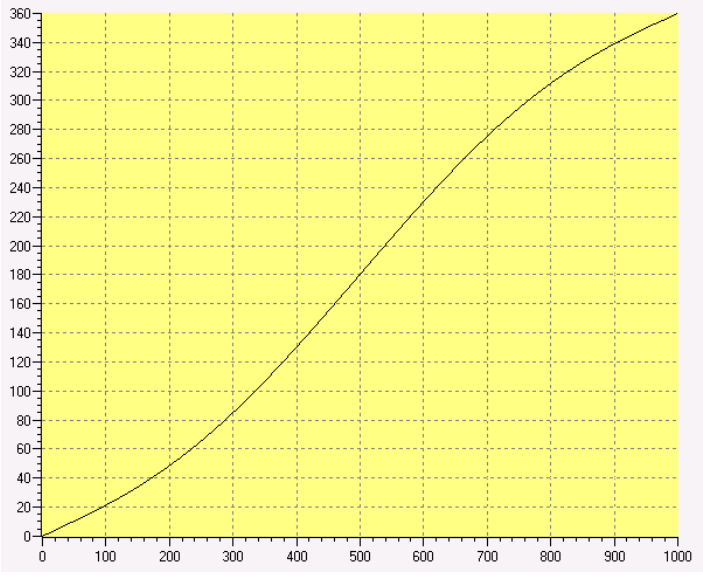


Figure 4-75 Cam for cyclic operation (CamEdit representation)

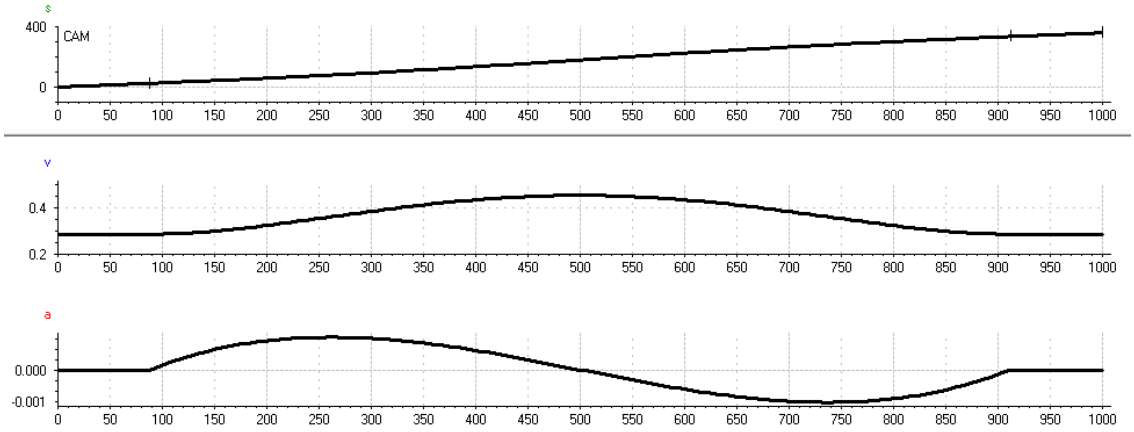


Figure 4-76 Cam for cyclic operation (CamTool representation)

Before calculating the cam, the following input parameters are checked for plausibility.

When the	
crankLength	<= 0.0 or
conrodLength	<= 0.0 or
syncStart	< mechRangeStart or
syncStart	> mechRangeEnd or
syncEnd	< syncStart or
syncEnd	>= mechRangeEnd

then an appropriate error number is output.

If an error occurs during the check, the cam is not calculated.

Schematic block view in LAD/FBD

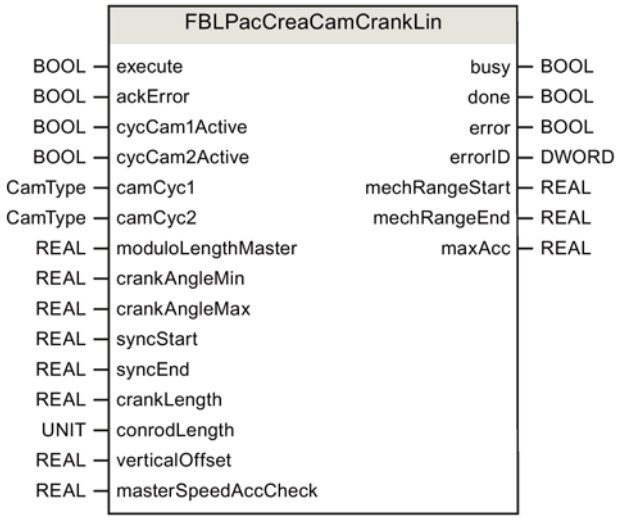


Figure 4-77 FBLPacCreaCamCrankLin - schematic block view in LAD/FBD

Input and output parameters

Table 4- 165 Input and output parameters

Element	P type [1]	Data type	Initial value	Meaning
execute	IN	BOOL	FALSE	Start of the cam calculation
ackError	IN	BOOL	FALSE	Acknowledge error
cycCam1Active	IN	BOOL	FALSE	Cyclic cam 1 active in synchronous operation
cycCam2Active	IN	BOOL	FALSE	Cyclic cam 2 active in synchronous operation
camCyc1	IN	CamType	TO#NIL	Technology object of the first cyclic cam to be calculated
camCyc2	IN	CamType	TO#NIL	Technology object of the second cyclic cam to be calculated
moduloLength-Master	IN	REAL	1000.0	Modulo length of the master
crankAngleMin	IN	REAL	0.0	Start angle of the crank motion (motor-side)
crankAngleMax	IN	REAL	0.0	End angle of the crank motion (motor-side)
syncStart	IN	REAL	0.0	Start point of the synchronous run for the leading axis [mm] < 0
syncEnd	IN	REAL	0.0	End point of the synchronous run for the leading axis [mm] > 0
crankLength	IN	REAL	0.0	Crank length [mm]

Element	P type [1]	Data type	Initial value	Meaning
conrodLength	IN	REAL	0.0	Conrod length [mm]
verticalOffset	IN	REAL	0.0	Vertical offset [mm]
masterSpeed-AccCheck	IN	REAL	1000.0	Maximum velocity of the leading axis in mm/s to calculate the maximum acceleration of the following axis
busy	OUT	BOOL	FALSE	Calculation of the cam active
done	OUT	BOOL	FALSE	Calculation of the cam completed
error	OUT	BOOL	FALSE	Error occurred
errorID	OUT	DWORD	16#0000_0000	Error code
mechRangeStart	OUT	REAL	0.0	Lowest possible mechanical linear position on the slider-crank output
mechRangeEnd	OUT	REAL	0.0	Highest possible mechanical linear position on the slider-crank output
maxAcc	OUT	REAL	0.0	Maximum motor acceleration in degrees/s^2

1) Parameter types: IN = input parameter, OUT = output parameter

4.7.3.3 Embedding the module

Technology objects, basic parameterization and block call

Table 4- 166 TO type and basic parameterizations

Cams	Two cams for the states: • Cyclic operation 1/2
Block call	MotionTask

Further blocks and libraries

The **FBLPacCreaCamCrankLin** block uses the following blocks internally.

Block/library	Function
FBLPacTranslationToRotation / LPac	Check of the profile parameters
FBLCamHdlCreateCam / LCam-Hdl	Cam calculation block

Some of the internally used blocks originate from other libraries (Page 15).

Detailed descriptions for the **FBLCamHdlCreateCam** block are contained in the SIMOTION **LCamHdl** Library User Manual at

https://support.industry.siemens.com/cs/ww/en/view/105644659

Error messages

Error number [HEX]	Meaning
16#0000	LPAC_ERR_NO; error-free
16#4000…4019	Error message of the FBLCamHdlCreateCam block; see **LCamHdl** SIMOTION Library Manual
16#8000	ERR_INVALID_STEP; internal error, please get in touch with your Siemens contact partner
16#8012	LPAC_ERR_NO_CAM; at least one cam object is not specified
16#80BA	LPAC_ERR_CRANK_LENGTH_IS_ZERO_OR_NEGATIVE; crank length ≤ 0
16#80BC	LPAC_ERR_CONROD_LENGTH_IS_ZERO_OR_NEGATIVE; conrod length ≤ 0
16#80C3	LPAC_ERR_EXECUTE_DURING_RESET; Execute requested while error acknowledgment active
16#80C5	LPAC_ERR_SYNCSTART_SMALER_THAN_MECH_RANGE_START; start of the synchronous range is less than the start of the mechanical traversing range
16#80C6	LPAC_ERR_SYNCSTART_GREATER_THAN_MECH_RANGE_END; start of the synchronous range is greater than the end of the mechanical traversing range
16#80C7	LPAC_ERR_SYNCEND_SMALER_THAN_SYNCSTART; end of the synchronous range is less then the start of the synchronous range
16#80C8	LPAC_ERR_SYNCEND_GREATER_THAN_MECH_RANGE_END; end of the synchronous range is greater than the end of the mechanical traversing range
16#80C9	LPAC_ERR_SYNCEND_EQUAL_THAN_MECH_RANGE_END; end of the synchronous range is equal to the end of the mechanical traversing range
16#80CA	LPAC_ERR_BOTH_CYCLIC_CAMS_ACTIVE; both cyclic cams are active
16#80CB	LPAC_ERR_LINEAR_DIFFERENCE_ZERO; linear distance on the slider-crank output is equal to 0

Note

Tip

The **pCreaCamCrankLin** program unit is available in the LPac project to illustrate the use of the **FBLPacCreaCamCrankLin** block.

This unit contains programs for the required block calls in the MotionTask.

4.8 Auxiliary functions

4.8.1 FBLPacTranslationToRotation, FBLPacRotationToTranslation - slider crank transformation

4.8.1.1 Technological consideration

The mechanical system consisting of a crank and a conrod converts the rotary motion of the motor to a translatory motion (slider-crank mechanical system).

The center point of the crank circle is defined as the zero position. For a spinning axis, two possible solutions for the angle result from a position. Both angles are output; the user must decide which angle should be used.

The parallel to the effective direction of the cross-sealer is defined as zero angle. The angle increases in the clockwise direction.

The direction of movement of the conrod end point does not need to lie in an effective line with the center point of the crank circle. This vertical shift is designed as vertical offset.

The slider crank is shown schematically in the following figure.

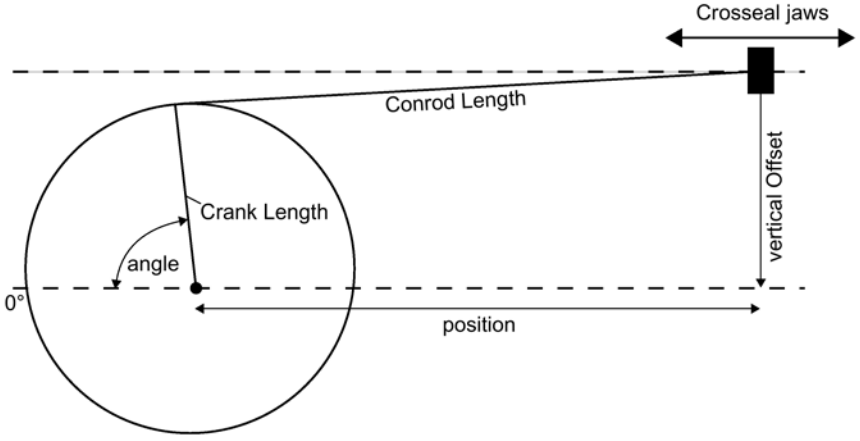


Figure 4-78 Mechanical construction of a slider crank

The**FBLPacTranslationToRotation** block transforms the linear position of the slider-crank mechanical system (position of the conrod end point) into the angle of a rotary axis (crank angle).

The**FBLPacRotationToTranslation** block transforms the angles of a rotary axis into the linear position passed via a slider-crank mechanical system. It so operates in the opposite direction to the block FBLPacTranslationToRotation**.** The minimum and maximum angles that can be achieved with the passed mechanical system are also calculated.

4.8.1.2 Module description

Scope of functions

The **FBLPacTranslationToRotation** function block provides the main and auxiliary functions, and possibly options, shown in the following overview.

Function	Description	Implementation
Main function	Calculation of the angular position of a specified slider-crank mechanical system as well as the maximum and minimum possible linear positions.	With the specification of the mechanical definition of the crank length and the conrod length as well as possibly a vertical offset (vertical shift of the conrod end compared to the turning center of the crank), the associated angular position of the crank is calculated for a specified linear position of the conrod.
Options	Calculation of the minimum and maximum paths of the conrod	The minimum and maximum position that can be achieved with the passed geometry is calculated. An error message is issued if the passed position does not lie within the range between the minimum and the maximum position.
Auxiliary functions	Conversion of force to torque	To estimate the torque required on the slider crank, the block enables a force value specified on the input side to be converted to the torque required on the crank. The force value refers to the pivot point of the conrod on the linear slide. **The calculation only considers the static case. The masses, accelerations, decelerations, moments of inertia, frictions as well as the gearbox efficiency to be considered in the dynamic case are not included in the calculation!**

The **FBLPacRotationToTranslation** function block provides the main and auxiliary functions, and possibly options, shown in the following overview.

Function	Description	Implementation
Main function	Calculation of the linear position of a specified slider-crank mechanical system as well as the maximum and minimum possible crank angles.	With the specification of the mechanical definition of the crank length and the conrod length as well as possibly a vertical offset (vertical shift of the conrod end compared to the turning center of the crank), the associated linear position of the conrod is calculated for a specified crank angle.
Options	Calculating the minimum and maximum angle of the crank	The minimum and maximum angle that can be achieved with the passed geometry is calculated. An error message is issued if the passed angle does not lie within the range between the minimum and the maximum angle.
Auxiliary functions	Conversion of torque to force	To estimate the force resulting on the linear slide for a torque specified on the slider crank on the input side, the block enables a torque value specified on the input side to be converted to the resulting force on the linear slide. The force value refers to the pivot point of the conrod on the linear slide. **The calculation only considers the static case. The masses, accelerations, decelerations, moments of inertia, frictions as well as the gearbox efficiency to be considered in the dynamic case are not included in the calculation!**

Requirements

The following requirements must be satisfied:

- The *crankLength* and *conrodLength* must be greater than 0.
- The offset between the center point and the conrod end point (*verticalOffset*) must not be greater than the sum of the crank and conrod lengths.

Note

The blocks calculate the linear position or the crank angle in a cycle. They are called in a cyclic or sequential task.

Schematic block view in LAD/FBD

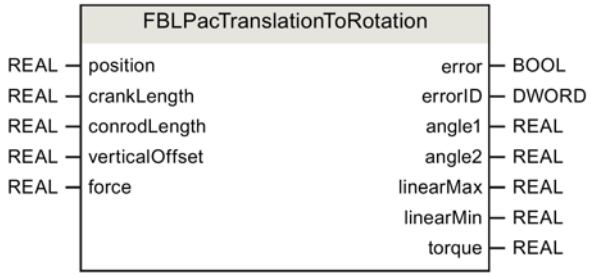


Figure 4-79 FBLPacTranslationToRotation - schematic block view in LAD/FBD

FBLPacTranslationToRotation input and output parameters

Table 4- 167 Input and output parameters

Parameter	Data type	P type [1]	Initial value	Meaning
position	IN	REAL	0.0	Linear position of the conrod to be converted [mm]
crankLength	IN	REAL	0.0	Crank length [mm]
conrodLength	IN	REAL	0.0	Conrod length [mm]
verticalOffset	IN	REAL	0.0	Vertical offset [mm]
force	IN	REAL	0.0	Desired force on the linear slide
error	OUT	BOOL	FALSE	Error while processing the block
errorID	OUT	DWORD	16#0000 _0000	Error code
angle1	OUT	REAL	0.0	Angle of rotation 1
angle2	OUT	REAL	0.0	Angle of rotation 2
linearMax	OUT	REAL	0.0	Maximum possible linear position [mm]
linearMin	OUT	REAL	0.0	Minimum possible linear position [mm]
torque	OUT	REAL	0.0	Required torque on the slider crank for a linear force specified on the input side

[1] Parameter types: IN = input parameter, OUT = output parameter

Schematic block view in LAD/FBD

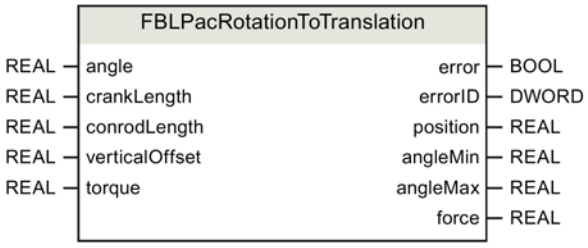


Figure 4-80 FBLPacRotationToTranslation - schematic block view in LAD/FBD

FBLPacRotationToTranslation input and output parameters

Table 4- 168 Input and output parameters

Parameter	P type 1)	Data type	Initial value	Meaning
angle	IN	REAL	0.0	Angle [°]
crankLength	IN	REAL	0.0	Crank length [mm]
conrodLength	IN	REAL	0.0	Conrod length [mm]
verticalOffset	IN	REAL	0.0	Vertical offset [mm]
torque	IN	REAL	0.0	Torque on the slider crank specified on the input side
error	OUT	BOOL	FALSE	Error occurred
errorID	OUT	DWORD	16#0000_0000	Error code
position	OUT	REAL	0.0	Position in mm
angleMin	OUT	REAL	0.0	Minimum angle depending on the input parameter [°]
angleMax	OUT	REAL	0.0	Maximum angle depending on the input parameters [°]
force	OUT	REAL	0.0	Force on the linear slide resulting from the torque specified on the input side.

1) Parameter types: IN = input parameter, OUT = output parameter

4.8.1.3 Integration of the modules

Technology objects, basic parameterization and block call

Table 4- 169 TO type and basic parameterizations

Axis / technology object	
TO type	Not relevant
Version	Not relevant
Modulo	Not relevant
Block call	BackgroundTask or MotionTask

Further blocks and libraries

The **FBLPacTranslationToRotation** and **FBLPacRotationToTranslation** blocks do not use any further blocks.

FBLPacTranslationToRotation error messages

Error number [HEX]	Meaning
16#0000	LPAC_ERR_NO; error-free
16#80D0	LPAC_ERR_INVALID_INPUT_PARAMETER; invalid position value
16#80D1	LPAC_ERR_VERTICAL_OFFSET_LENGTH_INVALID; invalid mechanical definition parameter (vertical offset)
16#80D2	LPAC_ERR_TRANSLAT_POS_INVALID; linear position outside the maximum possible position range
16#80E0	LPAC_ERR_CRANK_OR_CONROD_LENGTH_INVALID; invalid mechanical definition parameter (crank length, conrod length)

FBLPacRotationToTranslation error messages

Error number [HEX]	Meaning
16#0000	LPAC_ERR_NO; error-free
16#80D1	LPAC_ERR_VERTICAL_OFFSET_LENGTH_INVALID; invalid mechanical definition parameter (vertical offset)
16#80E0	LPAC_ERR_CRANK_OR_CONROD_LENGTH_INVALID; invalid mechanical definition parameter (crank length, conrod length)
16#80E1	LPAC_ERR_ROTATION_ANGLE_INVALID; crank angle outside the maximum possible angle range

Note

Tip

The **pSliderCrank** program unit is available in the LPac project to illustrate the use of the **FBLPacTranslationToRotation** and **FBLPacRotationToTranslation** blocks.

This unit contains the program for the required block call in the BackgroundTask.

4.8.2 FBLPacGetPosTime (Jerk) – calculate positioning time

4.8.2.1 Technological consideration

The **FBLPacGetPosTime** block calculates for trapezoidal or triangular positioning profiles, the associated motion times (total time, acceleration time, constant traversing time and deceleration time) for the specified distance, velocity, acceleration and deceleration. Application examples include, for example, opening and closing motions for cross-sealers.

The **FBLPacGetPosTimeJerk** block is also available for jerk-limited motions. When a jerk value is specified at the **"jerk"** input, it calculates for trapezoidal or triangular positioning profiles, the associated motion times (total time, acceleration time, constant traversing time and deceleration time) for the specified distance, velocity, acceleration and deceleration.

4.8.2.2 Module description

Scope of functions

The **FBLPacGetPosTime** and **FBLPacGetPosTimeJerk** function blocks provide the main and auxiliary functions, and possibly options, shown in the following overview.

Function	Description	Implementation
Main function	Calculation of the complete motion time, the acceleration and deceleration time, and the constant traversing time of a specified positioning motion.	The *distance, velocity*, *acceleration* and *deceleration* input parameters must be specified for the block. The block then calculates automatically the times for the various phases of the positioning motion.
Options	-	-
Auxiliary functions	-	-

Requirements

The following requirements must be satisfied:

- The positioning distance must be greater than or equal to 0
- The velocity, acceleration and deceleration must not be less than or equal to 0; this also applies for the jerk in the **FBLPacGetPosTimeJerk** block.
- The velocity characteristic must be trapezoidal or triangular
- Velocity and acceleration overrides must be 100%

Note

The block assumes for the calculation a velocity and acceleration override of 100% and a trapezoidal or triangular motion profile!

Schematic block view in LAD/FBD

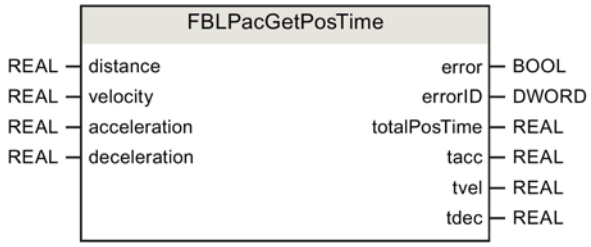


Figure 4-81 FBLPacGetPosTime – schematic block view in LAD/FBD

Input and output parameters

Table 4- 170 Input and output parameters

Parameter	P type [1)]	Data type	Initial value	Meaning
distance	IN	REAL	0.0	Total positioning distance [mm or º]
velocity	IN	REAL	0.0	Maximum velocity [mm/s or º/s]
acceleration	IN	REAL	0.0	Acceleration [mm/s2 or º/s2]
deceleration	IN	REAL	0.0	Deceleration [mm/s2 or º/s2]
jerk (only on the **FBLPacGetPosTimeJerk** block)	IN	REAL	0.0	Jerk [mm/s3 or º/s3]
error	OUT	BOOL	FALSE	Error occurred
errorID	OUT	DWORD	16#0000_0000	Error code
totalPosTime	OUT	REAL	0.0	Time for the total motion [s]
tacc	OUT	REAL	0.0	Acceleration time [s]
tvel	OUT	REAL	0.0	Constant traversing time [s]
tdec	OUT	REAL	0.0	Deceleration time [s]
taccFlank (only on the **FBLPacGetPosTimeJerk** block)	OUT	REAL	0.0	Traversing time for the start of acceleration or end of acceleration [s]
tdecFlank (only on the **FBLPacGetPosTimeJerk** block)	OUT	REAL	0.0	Traversing time for the start of deceleration or end of deceleration [s]
tConstAccRamp (only on the **FBLPacGetPosTimeJerk** block)	OUT	REAL	0.0	Traversing time for the acceleration action with constant acceleration [s]

Parameter	P type [1]	Data type	Initial value	Meaning
tConstDecRamp (only on the **FBLPacGetPosTimeJerk** block)	OUT	REAL	0.0	Traversing time for the deceleration action with constant deceleration [s]
maxVelocity (only on the **FBL-PacGetPosTimeJerk** block)	OUT	REAL	0.0	Maximum occurring velocity [mm/s or °/s]

[1] data types: IN = input parameter, OUT = output parameter

The units depend on the axis units. For linear axes, the units mm, mm/s and mm/s^2 are used; for rotary axes °, °/s and °/s^2.

4.8.2.3 Integration of the module

Technology objects, basic parameterization and block call

Table 4- 171 TO type and basic parameterizations

Axis / technology object	
TO type	Not relevant
Version	Not relevant
Modulo	Not relevant
Block call	BackgroundTask or MotionTask

Further blocks and libraries

The **FBLPacGetPosTime** and **FBLPacGetPosTimeJerk** blocks do not use any further blocks.

Error messages

Error number [HEX]	Meaning
16#0000	LPAC_ERR_NO; error-free
16#8150	LPAC_ERR_DISTANCE_LESS_THAN_OR_EQUAL_TO_ZERO; parameter *distance* ≤ 0
16#8151	LPAC_ERR_VELOCITY_ZERO_OR_NEGATIVE; parameter *velocity* ≤ 0
16#8152	LPAC_ERR_ACCELERATION_ZERO_OR_NEGATIVE; parameter *acceleration* ≤ 0
16#8153	LPAC_ERR_DECELERATION_ZERO_OR_NEGATIVE; parameter *deceleration* ≤ 0
16#8154	LPAC_ERR_INVALID_PARAMETERS; invalid parameters require division by 0
16#815F	LPAC_ERR_JERK_NEGATIVE; parameter jerk < 0 (only on the **FBLPacGet-PosTimeJerk** block)

4.8.3 FBLPacGetPosAtTime – calculate position

4.8.3.1 Technological consideration

The **FBLPacGetPosAtTime** block calculates for trapezoidal or triangular positioning profiles, the target position than can be reached with specified time, velocity, velocity override, constant traversing time, acceleration, acceleration time, acceleration override, deceleration and deceleration time.

4.8.3.2 Module description

Scope of functions

The **FBLPacGetPosAtTime** function block provides the main and auxiliary functions, and possibly options, shown in the following overview.

Function	Description	Implementation
Main function	Calculation of the position that is reached within a specified time of a specified positioning motion.	The input parameters *posTime*, *distance, velocity*, *acceleration, deceleration, acceleration and velocity overrides as well as the acceleration, constant traversing and deceleration times* must be specified on the block. Using these parameters, the block calculates the position reached after a specified time.
Options	-	-
Auxiliary functions	-	-

Requirements

The following requirements must be satisfied:

- The positioning distance must be greater than or equal to 0
- The velocity, acceleration, deceleration and velocity and acceleration override must not be less than or equal to 0
- The velocity characteristic must be trapezoidal or triangular

Note

The block assumes for the calculation a trapezoidal or triangular motion profile!

Schematic representation in LAD/FBD

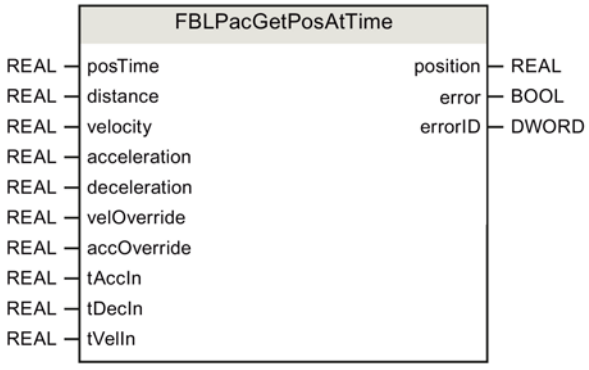


Figure 4-82 FBLPacGetPosAtTime – schematic block view in LAD/FBD

Input and output parameters

Table 4- 172 Input and output parameters

Parameter	P type [1)]	Data type	Initial value	Meaning
posTime	IN	REAL	0.0	Traversing time [s]
distance	IN	REAL	0.0	Total positioning distance [mm or º]
velocity	IN	REAL	0.0	Velocity [mm/s or º/s]
acceleration	IN	REAL	0.0	Acceleration [mm/s^2 or º/s^2]
deceleration	IN	REAL	0.0	Deceleration [mm/s^2 or º/s^2]
velOverride	IN	REAL	0.0	Velocity override [%]
accOverride	IN	REAL	0.0	Acceleration override [%]
tAccIn	IN	REAL	0.0	Acceleration time [s]
tDecIn	IN	REAL	0.0	Deceleration time [s]
tVelIn	IN	REAL	0.0	Constant traversing time [s]
Error	OUT	BOOL	FALSE	Error occurred
errorID	OUT	DWORD	16#0000 _0000	Error code
position	OUT	REAL	0.0	Position reached in the specified traversing time [mm or º]

1) Data types: IN = input parameter, OUT = output parameter

The units depend on the axis units. For linear axes, the units mm, mm/s and mm/s^2 are used; for rotary axes º, º/s and º/s^2.

4.8.3.3 Embedding the module

Technology objects, basic parameterization and block call

Table 4- 173 TO type and basic parameterizations

Axis / technology object	
TO type	Not relevant
Version	Not relevant
Modulo	Not relevant
Block call	BackgroundTask or MotionTask

Further blocks and libraries

The **FBLPacGetPosAtTime** block does not use any further blocks.

Error messages

Error number	Meaning
16#0000	LPAC_ERR_NO; error-free
16#8150	LPAC_ERR_DISTANCE_LESS_THAN_OR_EQUAL_ZERO; parameter distance ≤ 0
16#8151	LPAC_ERR_VELOCITY_ZERO_OR_NEGATIVE; parameter velocity ≤ 0
16#8152	LPAC_ERR_ACCELERATION_ZERO_OR_NEGATIVE; parameter acceleration ≤ 0
16#8153	LPAC_ERR_DECELERATION_ZERO_OR_NEGATIVE; parameter *deceleration* ≤ 0
16#8156	LPAC_ERR_ACCELERATION_OVERRIDE_ZERO_OR_NEGATIVE; acceleration override ≤ 0
16#8155	LPAC_ERR_POSITION_GREATER_THAN_DISTANCE; position > distance
16#8157	LPAC_ERR_VELOCITY_OVERRIDE_ZERO_OR_NEGATIVE; velocity override ≤ 0
16#8158	LPAC_ERR_ACCELERATION_TIME_ZERO_OR_NEGATIVE; acceleration time ≤ 0
16#8159	LPAC_ERR_DECELERATION_TIME_ZERO_OR_NEGATIVE; deceleration time ≤ 0

4.8.4 FBLPacGetTimeAtPosition – calculate motion time

4.8.4.1 Technological consideration

The **FBLPacGetTimeAtPosition** block calculates for trapezoidal or triangular positioning profiles, the traversing time until a specified position is reached with specified distance, velocity, velocity override, constant traversing time, acceleration, acceleration time, acceleration override, deceleration and deceleration time.

4.8.4.2 Module description

Scope of functions

The **FBLPacGetTimeAtPosition** function block offers the main functions and auxiliary functions as well as possible options shown in the following overview.

Function	Description	Implementation
Main function	Calculation of time until a specifiable position is reached of a specified positioning motion.	The input parameters *position*, *distance, velocity*, *acceleration, deceleration, acceleration and velocity overrides as well as the acceleration, constant traversing and deceleration times* must be specified on the block. Using these parameters, the block calculates the time until the specified position is reached.
Options	-	-
Auxiliary functions	-	-

Requirements

The following requirements must be satisfied:

- The positioning distance must be greater than or equal to 0
- The velocity, acceleration, deceleration and velocity and acceleration override must not be less than or equal to 0
- The velocity characteristic must be trapezoidal or triangular

Note

The block assumes for the calculation a trapezoidal or triangular motion profile!

Schematic block view in LAD/FBD

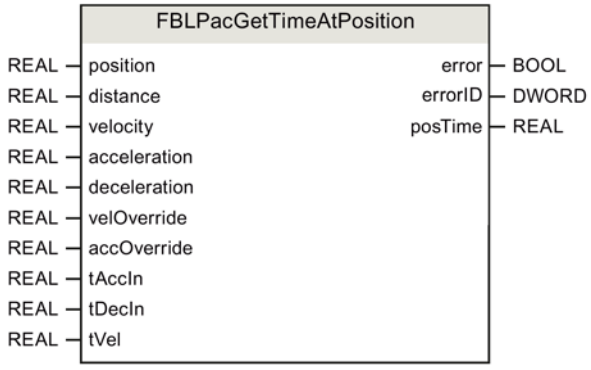


Figure 4-83 FBLPacGetTimeAtPosition – schematic block view in LAD/FBD

Table 4- 174 Input and output parameters

Parameter	P type 1)	Data type	Initial value	Meaning
position	IN	REAL	0.0	Position up to which the traversing time is to be calculated [mm or º]
distance	IN	REAL	0.0	Total positioning distance [mm or º]
velocity	IN	REAL	0.0	Velocity [mm/s or º/s]
acceleration	IN	REAL	0.0	Acceleration [mm/s² or º/s²]
deceleration	IN	REAL	0.0	Deceleration [mm/s² or º/s²]
velOverride	IN	REAL	0.0	Velocity override [%]
accOverride	IN	REAL	0.0	Acceleration override [%]
tAccIn	IN	REAL	0.0	Acceleration time [s]
tDecIn	IN	REAL	0.0	Deceleration time [s]
tVel	IN	REAL	0.0	Constant traversing time [s]
error	OUT	BOOL	FALSE	Error occurred
errorID	OUT	DWORD	16#0000_0000	Error code
posTime	OUT	REAL	0.0	Time until the specified position is reached [s]

1) Data types: IN = input parameter, OUT = output parameter

The units depend on the axis units. For linear axes, the units mm, mm/s and mm/s² are used; for rotary axes º, º/s and º/s².

4.8.4.3 Embedding the module

Technology objects, basic parameterization and block call

Table 4- 175 TO type and basic parameterizations

Axis / technology object	
TO type	Not relevant
Version	Not relevant
Modulo	Not relevant
Block call	BackgroundTask or MotionTask

Further blocks and libraries

The **FBLPacGetTimeAtPosition** block does not use any further blocks.

Error messages

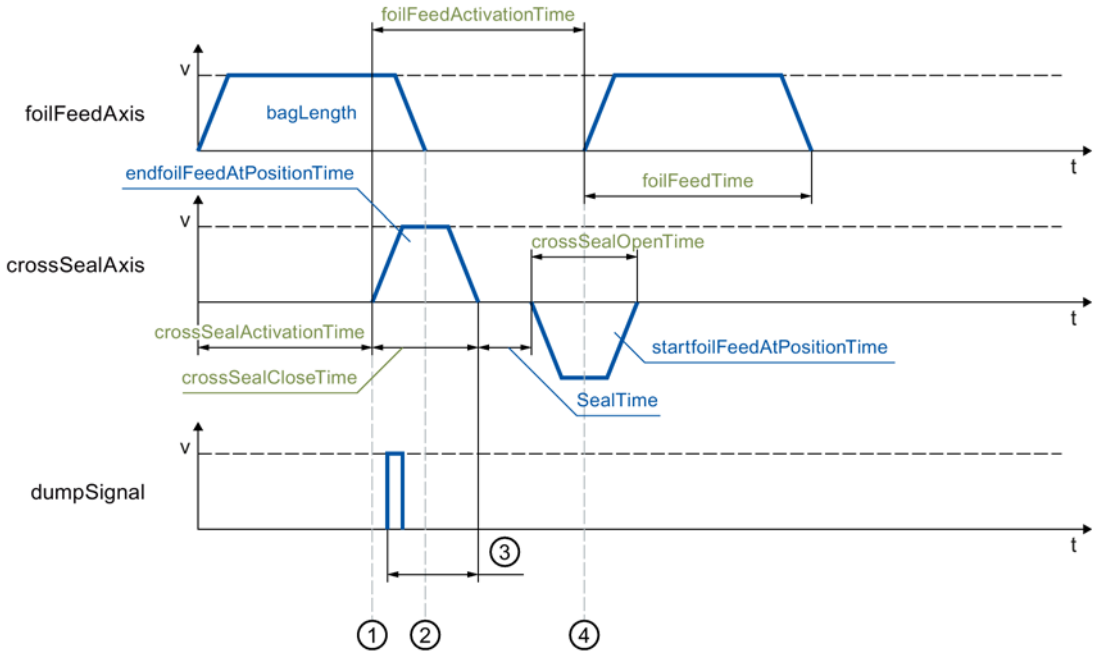
Error number [HEX]	**Meaning**
16#0000	LPAC_ERR_NO; error-free
16#8150	LPAC_ERR_DISTANCE_LESS_THAN_OR_EQUAL_ZERO; parameter *distance* ≤ 0
16#8151	LPAC_ERR_VELOCITY_ZERO_OR_NEGATIVE; parameter velocity ≤ 0
16#8152	LPAC_ERR_ACCELERATION_ZERO_OR_NEGATIVE; parameter acceleration ≤ 0
16#8153	LPAC_ERR_DECELERATION_ZERO_OR_NEGATIVE; parameter *deceleration* ≤ 0
16#8155	LPAC_ERR_POSITION_GREATER_THAN_DISTANCE; target position is outside the total motion
16#8156	LPAC_ERR_ACCELERATION_OVERRIDE_ZERO_OR_NEGATIVE; acceleration override ≤ 0
16#8157	LPAC_ERR_VELOCITY_OVERRIDE_ZERO_OR_NEGATIVE; velocity override ≤ 0
16#8158	LPAC_ERR_ACCELERATION_TIME_ZERO_OR_NEGATIVE; acceleration time ≤ 0
16#8159	LPAC_ERR_DECELERATION_TIME_ZERO_OR_NEGATIVE; deceleration time ≤ 0

4.8.5 FBLPacGetCycleTimeVI – calculate cycle time

4.8.5.1 Technological consideration

The **FBLPacGetCycleTimeVI** block calculates the cycle time for a single production cycle of a vertical intermittent tubular bagging machine. Required for this are the appropriate specifications as well as trapezoidal or triangular positioning operations of the relevant foil feed and cross-sealer axes. Jerk-limited positioning motions cause incorrect results. The block is called internally in the master block for the vertical intermittent machine **FBLPacMasterInt** in order to calculate the shortest possible cycle time for maximum dynamic traversing response as well as the cuurrent cycle time.

The following figure shows the coordination and motion superimposition of the foil feed, cross-sealer and dosing units for a vertical intermittent tubular bagging machine. It illustrates how the production cycle is made up. The required and the calculated values are also displayed.



① crossSealerOpen
② endfoilFeedAtPosition
③ dropTime
④ startfoilFeedAtPosition

To calculate the current cycle time, the block first calculates the current traversing velocities, accelerations and decelerations internally from the maximum velocities and the current overrides for velocity and acceleration specified in the two parameter data structures *sLPacPosMaxDynDataType* (for the foil feed) and *sLPacCSPosMaxDynDataType* (for the cross-sealer).

From the calculated velocities, accelerations and decelerations, the traversing times for the foil feed *foilFeedTime* as well as the opening and closing times of the cross-sealer *crossSealOpenTime* and *crossSealCloseTime* are calculated.

Finally the total cycle time is calculated with the default values for the sealing time *sealTime* and the externally calculated motion overlap times *endFoilFeedAtPositionTime* and *startFoilFeedAtPositionTime*.

cycleTime =
foilFeedTime + crossSealCloseTime + (sealTime/1000) + crossSealOpenTime
– endFoilFeedAtPositionTime – startFoilFeedAtPositionTime

4.8.5.2 Module description

Scope of functions

The **FBLPacGetCycleTimeVI** function block offers the main functions and auxiliary functions as well as possible options shown in the following overview.

Function	Description	Implementation
Main function	Calculation of the cycle time	The input parameters bag length, sealing time, time at which the foil feed is ended, time after which the foil feed starts, posTime, distance, velocity, acceleration, deceleration, acceleration and velocity overrides as well as the acceleration, constant traversing and deceleration times must be specified on the block. From these parameters, the block calculates the traversing times for the foil feed *foilFeedTime,* the opening and closing times of the cross-sealer *crossSealOpenTime* and *crossSealCloseTime*, the total cycle time as well as the activation times for cross-sealer and foil feed.
Options	-	-
Auxiliary functions	-	-

Requirements

The following requirements must be satisfied:

- The positioning distance must be greater than or equal to 0
- The velocity, acceleration, deceleration, velocity and acceleration overrides must not be less than or equal to 0
- The velocity characteristic must be trapezoidal or triangular

Note

The block assumes for the calculation a trapezoidal or triangular motion profile!

Schematic representation in LAD/FBD

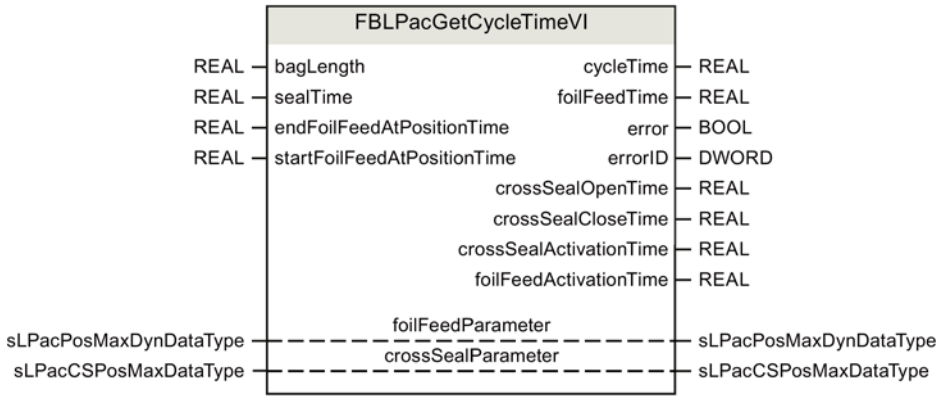


Figure 4-84 FBLPacGetCycleTimeVI - schematic block view in LAD/FBD

Input and output parameters

Table 4- 176 Input and output parameters

Parameter	P type [1)]	Data type	Initial value	Meaning
baglength	IN	REAL	0.0	Bag length [mm]
sealTime	IN	REAL	0.0	Sealing time [ms]
endFoilFeedAt PositionTime	IN	REAL	0.0	Time at which the foil feed motion is ended
startFoilFeedAt-PositionTime	IN	REAL	0.0	Time at which the next foil feed starts
cycleTime	OUT	REAL	0.0	Current cycle time
foilFeedTime	OUT	REAL	0.0	Motion time that the foil feed requires to feed the specified bag length with the traversing data specified in *foilFeedParamter*
crossSealOpen-Time	OUT	REAL	0.0	Motion time that the cross-sealer requires for the opening motion
crossSeal CloseTime	OUT	REAL	0.0	Motion time that the cross-sealer requires for the closing motion
crossSeal ActivationTime	OUT	REAL	0.0	Time after which the cross-sealer is closed after the start of the cycle
foilFeed ActivationTime	OUT	REAL	0.0	Time after which the foil feed is started for the next foil feed after the closing of the cross-sealer
error	OUT	BOOL	FALSE	Error occurred
errorID	OUT	DWORD	16#0000 _0000	Error code

Parameter	P type [1]	Data type	Initial value	Meaning
foilFeedParameter	IN_OUT	sLPacPosMax DynDataType	-	Parameter for the foil feed motion
crossSeal Parameter	IN_OUT	sLPac-CSPosMaxData Type	-	Parameter for the cross-sealer closing and opening motion
1) Data types: IN = input parameter, OUT = output parameter, IN_OUT = in-out parameter				

Table 4- 177 sLPacPosMaxDynDataType structure

Element	Data type	Initial value	Meaning
r32Position	REAL	0.0	Not relevant for the foil feed, only for the cross-sealer; open and closed position there
r32MaxVelocity	REAL	0.0	Maximum velocity of the axis
r32MaxAcceleration	REAL	0.0	Maximum acceleration of the axis
r32MaxDeceleration	REAL	0.0	Maximum deceleration of the axis
r64VelocityOverride	LREAL	0.0	Not relevant in this block
r64AccelerationOverride	LREAL	0.0	Not relevant in this block

Table 4- 178 sLPacCSPosMaxDataType structure

Element	Data type	Initial value	Meaning
boCSWithCrank	BOOL	FALSE	Selection of mechanical system with slider crank
sOpenData	sLPacPosMaxDyn-DataType	-	Maximum dynamic response values for opening motion
sCloseData	sLPacPosMaxDyn-DataType	-	Maximum dynamic response values for closing motion

4.8.5.3 Embedding the module

Technology objects, basic parameterization and block call

Table 4- 179 TO type and basic parameterizations

Axis / technology object	
TO type	Position axes
Version	Linear axis

Modulo	No
Block call	BackgroundTask or MotionTask

Further blocks and libraries

The **FBLPacGetCycleTimeVI** block uses the following blocks internally.

Block/library	Function
FCLPacGetPosTime / LPac	Calculation of positioning times

Error messages

Error number [HEX]	Meaning
16#0000	LPAC_ERR_NO; error-free
16#8150	LPAC_ERR_DISTANCE_LESS_THAN_OR_EQUAL_ZERO; parameter distance ≤ 0; error message from internally called FBLPacGetPosTime block
16#8151	LPAC_ERR_VELOCITY_ZERO_OR_NEGATIVE; parameter velocity ≤ 0; error message from internally called FBLPacGetPosTime block
16#8152	LPAC_ERR_ACCELERATION_ZERO_OR_NEGATIVE; parameter *acceleration* ≤ 0; error message from internally called FBLPacGetPosTime block
16#8153	LPAC_ERR_DECELERATION_ZERO_OR_NEGATIVE; parameter *deceleration* ≤ 0; error message from internally called FBLPacGetPosTime block
16#8154	LPAC_ERR_INVALID_PARAMETERS; invalid parameters require division by 0; error message from internally called FBLPacGetPosTime block
16#815C	LPAC_ERR_INVALID_END_FOIL_FEED_AT_POSITION; invalid position specification for end of foil feed

4.8.6 FBLPacSetOverrideVI - override adjustment

4.8.6.1 Technological consideration

In the application of a vertical intermittent tubular bagging machine, the **FBLPacSetOverrideVI** block compares the current desired cycle time with the shortest possible cycle time and then adapts the velocity and acceleration overrides accordingly for the relevant foil feed and cross-sealer axes. They are based on override values of 100% each for velocity and acceleration at the shortest cycle time (i.e. for the fastest traversing motions). Corresponding to the ratio of the desired cycle time to the shortest cycle time, the velocity overrides are scaled down linearly, but the acceleration overrides quadratically.

4.8.6.2 Module description

Scope of functions

The **FBLPacSetOverrideVI** function block offers the main functions and auxiliary functions as well as possible options shown in the following overview.

Function	Description	Implementation
Main function	Calculation of the velocity and acceleration overrides for the foil feed and the cross-sealer to achieve the desired cycle time.	The *SetCycleTime, sealTime and minCycleTime* input parameters must be specified on the block. From these parameters, the block calculates the velocity and acceleration overrides for the foil feed and the cross-sealer.
Options	-	-
Auxiliary functions	-	-

Requirements

The following requirements must be satisfied:

- The velocity and acceleration overrides must be 100% for the specified minimum cycle time
- The velocity characteristic must be trapezoidal or triangular

Note

The block assumes for the calculation a trapezoidal or triangular motion profile!

Schematic block view in LAD/FBD

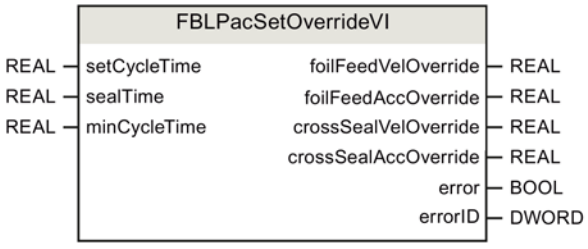


Figure 4-85 FBLPacSetOverrideVI - schematic block view in LAD/FBD

Input and output parameters

Table 4- 180 Input and output parameters

Parameter	P type [1]	Data type	Initial value	Meaning
setCycleTime	IN	REAL	0.0	Set cycle time [s]
sealTime	IN	REAL	0.0	Sealing time [ms]
minCycleTime	IN	REAL	0.0	Minimum cycle time [s]
foilFeedVelOverride	OUT	REAL	0.0	Foil feed velocity override [%]
foilFeedAccOverride	OUT	REAL	0.0	Foil feed acceleration override [%]
crossSealVelOverride	OUT	REAL	0.0	Cross-sealer velocity override [%]
crossSealAccOverride	OUT	REAL	0.0	Cross-sealer acceleration override [%]
error	OUT	BOOL	FALSE	Error occurred
errorID	OUT	DWORD	16#0000 _0000	Error code

1) Data types: IN = input parameter, OUT = output parameter

4.8.6.3 Embedding the module

Technology objects, basic parameterization and block call

Table 4- 181 TO type and basic parameterizations

Axis / technology object	
TO type	Not relevant
Version	Not relevant
Modulo	Not relevant
Block call	The block is called automatically in the FBLPacMasterInt block

Further blocks and libraries

The **FBLPacSetOverrideVI** block does not use any further blocks.

Error messages

Error number [HEX]	Meaning
16#0000	LPAC_ERR_NO; error-free
16#8021	LPAC_ERR_CYCLE_TIME_ZERO; set cycle time and/or minimum cycle time ≤ 0;
16#802E	LPAC_ERR_SEALTIME_ZERO; sealing time ≤ 0
16#802F	LPAC_ERR_SEALTIME_TOO_LONG; sealing time > minimum cycle time
16#8035	LPA_ERR_OVERRIDE_CALC_FACTOR; ratio of (set cycle time – sealing time) / (minimum cycle time / sealing time) is negative
16#8036	LPAC_ERR_OVERRIDE_CALC_SPEED_IS_LIMITED_TO_MIN; calculated velocity override has been limited to 1% (minimum override)
16#8037	LPAC_ERR_OVERRIDE_CALC_SPEED_IS_LIMITED_TO_MAX; calculated velocity override has been limited to 100% (maximum override)
16#8038	LPAC_ERR_OVERRIDE_CALC_ACC_IS_LIMITED_TO_MAX; calculated acceleration override has been limited to 100% (maximum override)

4.8.7 FBLPacChangeTOConfigData – change the TO axis configuration data

4.8.7.1 Technological consideration

The **FBLPacChangeTOConfigDataMod and FBLPacChangeTOConfigDataGear** blocks permit the programmed change of the **modulo length (toAxis.setConfigdata.Modulo.length)** and **modulo active (toAxis.steconfigData.Modulo.state) configuration data or load gear factor (toAxis.setConfigData.denFactor and numFactor)** on the TO axis. This is required, for example, to represent product lengths, package lengths or tool lengths on the modulo length of the axis or mechanical changes, such as the number of blades on the cross-sealer.

In contrast to the **FBLPacChangeTOConfigDataGear** block, the **FBLPacChangeTOConfigDataMod** block can also be used to change the modulo properties for external encoders.

4.8.7.2 Module description

Scope of functions

The **FBLPacChangeTOConfigDataMod** and **FBLPacChangeTOConfigDataGear** function blocks provide the main and auxiliary functions, and possibly options, shown in the following overview.

Function	Description	Implementation
Main function	Change the *toAxis.Modulo.length* and *toAxis.Modulo.state* configuration data or the *toAxis.denFactor* and *numFactor* load gear factors on the axis technology object.	The *moduloLength*, *moduloActive* input parameters or the numerator and denominator load gear factors and the axis TO must be specified on the block. The change is performed automatically by setting the *execute* input signal. A TO restart is then performed automatically.
Options	-	-
Auxiliary functions	-	-

Requirements

The following requirements must be met:

The axis must not be enabled when the configuration data is changed.

Note

The block initiates a restart of the axis TO after the configuration data has been changed.

No further change of the configuration data can be requested during an active TO restart.

Detailed information about changing the configuration data is contained in the *"SIMOTION Runtime Basic Functions"* SIMOTION Function Manual, Section 4.2.10 Configuration Data.

Schematic block views in LAD/FBD

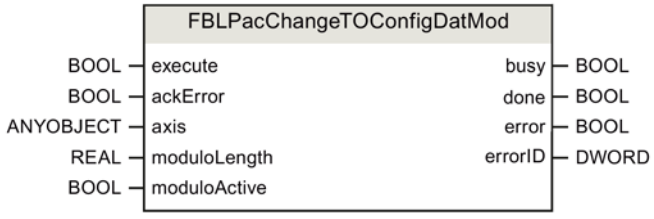


Figure 4-86 FBLPacChangeTOConfigDataMod - schematic block view in LAD/FBD

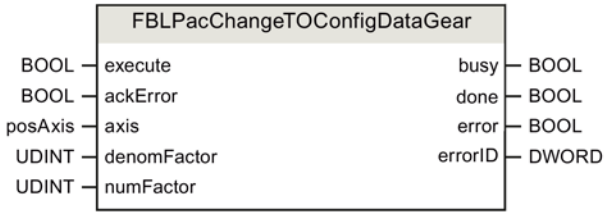


Figure 4-87 FBLPacChangeTOConfigDataGear - schematic block view in LAD/FBD

FBLPacChangeTOConfigDataMod input and output parameters

Table 4- 182 Input and output parameters

Parameter	P type [1]	Data type	Initial value	Meaning
execute	IN	BOOL	FALSE	Rising edge: change of the configuration data requested
ackError	IN	BOOL	FALSE	Acknowledge error
axis	IN	posAxis or also ExternalEncoderType possible for the **FBLPacChangeTOConfigDataMod** block	TO#NIL	Axis on which the configuration data should be changed
moduloLength	IN	REAL	0.0	Modulo length
moduloActive	IN	BOOL	FALSE	Modulo active yes/no
busy	OUT	BOOL	FALSE	Selected functionality being processed
done	OUT	BOOL	FALSE	Selected functionality completed successfully
error	OUT	BOOL	FALSE	Error occurred
errorID	OUT	DWORD	16#0000 _0000	Error code

1) Parameter types: IN = input parameter, OUT = output parameter

FBLPacChangeTOConfigDataGear input and output parameters

Table 4- 183 Input and output parameters

Parameter	P type [1]	Data type	Initial value	Meaning
execute	IN	BOOL	FALSE	Rising edge: change of the configuration data requested
ackError	IN	BOOL	FALSE	Acknowledge error
axis	IN	posAxis	TO#NIL	Axis on which the configuration data should be changed
denomFactor	IN	UDINT	1	Load gear, number of load revolutions
numFactor	IN	UDINT	1	Load gear, number of motor revolutions
busy	OUT	BOOL	FALSE	Selected functionality being processed
done	OUT	BOOL	FALSE	Selected functionality completed successfully

Parameter	P type [1)]	Data type	Initial value	Meaning
error	OUT	BOOL	FALSE	Error occurred
errorID	OUT	DWORD	16#'0000 _0000	Error code

1) Parameter types: IN = input parameter, OUT = output parameter

4.8.7.3 Integration of the module

Technology objects, basic parameterization and block call

Table 4- 184 TO type and basic parameterizations

Axis / technology object	
TO type	At least position axis
Version	Rotary or linear axis
Modulo	Parameterized on the block
Load gear ratio	Parameterized on the block
Block call	BackgroundTask

Further blocks and libraries

The **FBLPacChangeTOConfigDataMod** and **FBLPacChangeTOConfigDataGear** blocks do not use any further blocks.

Error messages

Error number [HEX]	Meaning
16#0000	LPAC_ERR_NO; error-free
16#8000	LPAC_ERR_INVALID_STEP; internal error, please get in touch with your Siemens contact partner
16#8001	LPAC_ERR_NO_AXIS; no reference to axis TO passed
16#8005	LPAC_ERR_MODULO_LENGTH; invalid modulo length passed (only **FBLPacChangeTOConfigDataMod** block)
16#80C3	LPAC_ERR_EXECUTE_DURING_RESET; Execute requested while error acknowledgment active
16#8089	LPAC_ERR_LOAD_GEAR_DEN_FACT_ZERO; load gear ratio error, number of load revolutions <= 0 (only for **FBLPacChangeTOConfigDataGear** block)
16#8088	LPAC_ERR_LOAD_GEAR_NUM_FACT_ZERO; load gear ratio error, number of motor revolutions <= 0 (only for **FBLPacChangeTOConfigDataGear** block)
16#8100	LPAC_ERR_READ_SYSTEM_VARIABLE; error when reading the system variables / variable cannot be read
16#8101	LPAC_ERR_MAX_TIME_FOR_TO_RESTART; TO restart not performed within the monitoring time

4.8.8 FBLPacOutPutCamControl – control cam controller

4.8.8.1 Technological consideration

The **FBLPacOutPutCamControl** block is used to control and parameterize a cam controller. Position-based or time-based cams can be used, uni-directional output cams are not supported.

4.8.8.2 Module description

Scope of functions

The **FBLPacOutPutCamControl** function block offers the main functions and auxiliary functions as well as possible options shown in the following overview.

Function	Description	Implementation
Main function	Parameterization and control of a cam controller of the position-based or time-based cam type.	The block must be passed the values for the activation and deactivation position (or ON duration), effective direction, switching logic, actuation time for the activation and deactivation edge as well as hysteresis in the **"parameter"** parameter structure. The cam controller can be activated and deactivated via the **"enableOutputCam"** input.
Options	-	-
Auxiliary functions	Control cam output (forcing)	The cam output can be specifically switched on and off again by controlling the "forceOutput" input. If the cam controller was already activated before the forcing, it is activated again after the forcing.
	Move output cam position	The output cam can be moved completely in the positive or negative direction incrementally by controlling the **"shiftCamPositive"** and **"shiftCamNegative"** inputs. The value for the shift is specified in the **"r32IncShiftValue"** parameter.
	Expand or compress output cam	The output cam can be expanded of compressed incrementally by controlling the "expandCam" and "compressCam"inputs. The value for the expansion or compression is specified in the "r32IncShiftValue" parameter.

Note

The block cannot be used to control uni-directional output cams. All parameters and settings not in the parameter structure of the block can be set in the default settings dialog box of the cam controller.

Schematic representation in LAD/FBD

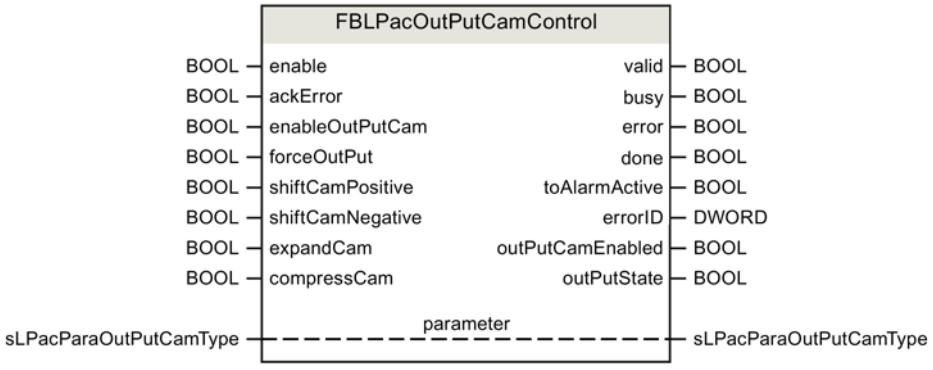


Figure 4-88 FBLPacOutPutCamControl – schematic block view in LAD/FBD

Input and output parameters

Table 4- 185 Input and output parameters

Element	P type [1]	Data type	Initial value	Meaning
enable	IN	BOOL	FALSE	Activation of the FB
ackError	IN	BOOL	FALSE	Acknowledge error
enableOutPutCam	IN	BOOL	FALSE	Enable cam controller
forceOutPut	IN	BOOL	FALSE	Switch cam output on or off again specifically (forcing). After resetting the **forceOutPut** input signal, the operating state is activated again that was active before the control of the input.
shiftCamPositive	IN	BOOL	FALSE	Move output cam incrementally in positive direction. The value by which the output cam position is to be moved is set in the **"r32IncShiftValue"** parameter.
shiftCamNegative	IN	BOOL	FALSE	Move output cam incrementally in negative direction. The value by which the output cam position is to be moved is set in the **"r32IncShiftValue"** parameter.
expandCam	IN	BOOL	FALSE	Expand output cam incrementally. The value by which the output cam is to be expanded is set in the **"r32IncShiftValue"** parameter.
compressCam	IN	BOOL	FALSE	Compress output cam incrementally. The value by which the output cam is to be compressed is set in the **"r32IncShiftValue"** parameter.
valid	OUT	BOOL	FALSE	Block is active
busy	OUT	BOOL	FALSE	Selected functionality being processed

Element	P type 1)	Data type	Initial value	Meaning
done	OUT	BOOL	FALSE	Selected functionality completed successfully
error	OUT	BOOL	FALSE	Error occurred
toAlarmActive	OUT	BOOL	FALSE	TO alarm occurred
errorID	OUT	DWORD	16#0000 _0000	Error code
outPutCamEnabled	OUT	BOOL	FALSE	Enable cam controller processing
outPutState	OUT	BOOL	FALSE	Switching state of the output cam
parameter	IN_OUT	sLPacParaOutPutCamType	-	Parameter for cam controller

1) Parameter types: IN = input parameter, OUT = output parameter, IN_OUT = in-out parameter

The units depend on the axis units. For linear axes, the unit mm is used; for rotary axes º.

Table 4- 186 sLParaOutPutCamType structure

Element	Data type	Meaning
sTOs	sLPacToOutPutCamDataType	TO for cam controller
sOutPutCamData	sLPacParaOutPutCamDataType	Parameter for output cam
sConfigData	sLPacConfigOutPutCamType	Configuration data for output cam processing

Table 4- 187 sLPacToOutPutCamDataType structure

Element	Data type	Initial value	Meaning
toOutputCam	outputCamType	TO#NIL	Technology object of the cam controller

Table 4- 188 sLPacParaOutPutCamDataType structure

Element	Data type	Initial value	Meaning
r32SwitchOnPosition	REAL	0.0	Activation position of the output cam
r32SwitchOffPosition	REAL	0.0	Deactivation position of the output cam (for position-based cams)
r32OnDuration	REAL	0.0	ON duration of the output cam (for time-based cams)
eForceDirection	EnumForceDirection	BOTH	Effective direction

eInvertOutput	EnumOutputCamInvert	NO	Inversion of the switching logic
r32ActivationTime	REAL	0.0	Actuation time on the activation edge
r32DeactivationTime	REAL	0.0	Actuation time on the deactivation edge
r32NoSwitchingRange	REAL	0.0	Hysteresis range

Note

The parameters of the *sLPacParaOutPutCamDataType* structure are transferred at each activation of the cam controller. The parameters are also transferred at every value change when the cam controller is activated.

Table 4- 189 sLPacConfigOutPutCamType structure

Element	Data type	Initial value	Meaning
r32IncShiftValue	REAL	1.0	Incremental value by which the output cam position is to be moved for the **shiftCamPositive, shiftCamNegative** functions, or by which the output cams are to be expanded or compressed for the **expandCam, compressCam** functions.

4.8.8.3 Embedding the module

Technology objects, basic parameterization and block call

Table 4- 190 TO type and basic parameterizations

Axis / technology object	
TO type	OutputcamType
Version	Position-based or time-based cams
Modulo	Not relevant
Block call	BackgroundTask

Further blocks and libraries

The **FBLPacOutPutCamControl** block does not use any further blocks.

Error messages

Error number [HEX]	Meaning
16#0000	LPAC_ERR_NO; error-free
16#8000	LPAC_ERR_INVALID_STEP; internal error, please get in touch with your Siemens contact partner
16#8002	LPAC_ERR_NO_OUTPUT_CAM; the block has not been passed a "Cam controller" technology object.
16#8003	LPAC_ERR_TO_ALARM_ACTIVE; TO alarm active; for the error message see Section Alarm and error messages in this manual
16#80A3	LPAC_ERR_ENABLE_OUTPUT_CAM; error from the _enableOutPutCam command; return value of the SIMOTION _enableOutputcam function is equivalent to the *_move* function and can be fetched in hex format in the high word from *errorID*
16#8170	LPAC_ERR_DISABLE_OUTPUT_CAM; error from the _disableOutPutCam command; return value of the SIMOTION _disableOutPutCam function is equivalent to the *_move* function and can be fetched in hex format in the high word from *errorID*
16#8171	LPAC_ERR_SET_OUTPUT_CAM_STATE; error from the _setOutPutCamState command; return value of the SIMOTION _setOutPutCamState function is equivalent to the *_move* function and can be fetched in hex format in the high word from *errorID*
16#8172	LPAC_ERR_WRONG_OUTPUT_CAM_TYPE; passed cam controller is not of the position-based or time-base cam type.

Note

The **pOutPutCam** program unit is available in the LPac project to illustrate the use of the **FBLPacOutPutCamControl** block.

This unit contains programs for the required block calls in the StartupTask and BackgroundTask.

4.8.9 FBLPacCamTrackControl – control cam track

4.8.9.1 Technological consideration

The **FBLPacCamTrackControl** block is used to control and parameterize a cam track (cam controller with up to 32 output cams). Position-based or time-based cams or time-based cams with maximum ON length can be used.

4.8.9.2 Module description

Scope of functions

The **FBLPacCamTrackControl** function block provides the main and auxiliary functions, and possibly options, shown in the following overview.

Function	Description	Implementation
Main function	Parameterization and control of a cam track with up to 32 output cams of the position-based or time-based cam type with maximum ON length.	The block must be passed the values for the actuation time for the activation and deactivation edge as well as hysteresis in the **"parameter"** parameter structure. The cam track can be activated and deactivated via the **"enableCamTrack"** input. In addition, output cams of the cam track can be selectively deactivated and activated by specifying a bit pattern at the **"enableValidCam"** input
Options	-	-
Auxiliary functions	Control cam output (forcing)	The cam track output can be specifically switched on and off again by controlling the **"forceOutPut"** input. If the cam track was already activated before the forcing, it is activated again after the forcing.
	Move output cam position	By controlling the **"shiftCamPositive"** and **"shiftCamNegative"** inputs, a selected output cam of the cam track can be moved in the positive or negative direction incrementally by entering the output cam number at the **"camNumber"** input. The value for the shift is specified in the **"r32IncShiftValue"** parameter.
	Expand or compress output cam	By controlling the **"expandCam"** and **"compressCam"** inputs, a selected output cam of the cam track can be expanded or compressed incrementally by entering the output cam number at the **"camNumber"** input. The value for the expansion or compression is specified in the **"r32IncShiftValue"** parameter.

Note

All parameters and settings not in the parameter structure of the block can be set in the default settings dialog box of the cam track.

Schematic block views in LAD/FBD

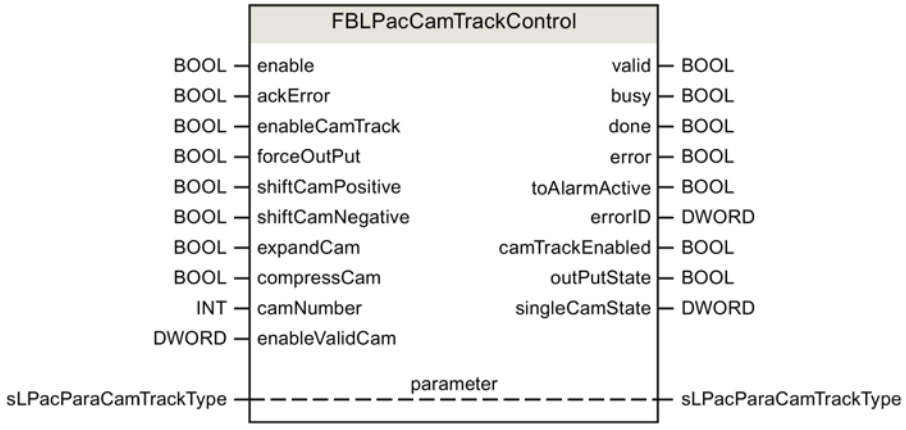


Figure 4-89 FBLPacCamTrackControl – schematic block view in LAD/FBD

Table 4- 191 Input and output parameters

Element	P type [1)]	Data type	Initial value	Meaning
enable	IN	BOOL	FALSE	Activation of the FB
ackError	IN	BOOL	FALSE	Acknowledge error
enableCamTrack	IN	BOOL	FALSE	Activate cam track
forceOutPut	IN	BOOL	FALSE	Switch cam track output on or off again specifically (forcing). After resetting the **forceOutPut** input signal, the operating state is activated again that was active before the control of the input.
shiftCamPositive	IN	BOOL	FALSE	Move output cam incrementally in positive direction. The value by which the output cam position is to be moved is set in the **"r32IncShiftValue"** parameter. The output cam is selected by entering the output cam number at the **"camNumber"** input.
shiftCamNegative	IN	BOOL	FALSE	Move output cam incrementally in negative direction. The value by which the output cam position is to be moved is set in the **"r32IncShiftValue"** parameter. The output cam is selected by entering the output cam number at the **"camNumber"** input.
expandCam	IN	BOOL	FALSE	Expand output cam incrementally. The value by which the output cam is to be expanded is set in the **"r32IncShiftValue"** parameter. The output cam is selected by entering the output cam number at the **"camNumber"** input.

Element	P type [1]	Data type	Initial value	Meaning
compressCam	IN	BOOL	FALSE	Compress output cam incrementally. The value by which the output cam is to be compressed is set in the **"r32IncShiftValue"** parameter. The output cam is selected by entering the output cam number at the **"camNumber"** input.
camNumber	IN	INT	1	Number of the output cam of the cam track that is to be influenced
enableValidCam	IN	DWORD		Specific output cams can be selectively deactivated and activated by specifying a bit pattern.
valid	OUT	BOOL	FALSE	Block is active
busy	OUT	BOOL	FALSE	Selected functionality being processed
done	OUT	BOOL	FALSE	Selected functionality completed successfully
error	OUT	BOOL	FALSE	Error occurred
toAlarmActive	OUT	BOOL	FALSE	TO alarm occurred
errorID	OUT	DWORD	16#0000 _0000	Error code
camTrackEnabled	OUT	BOOL	FALSE	Enable cam track processing
outPutState	OUT	BOOL	FALSE	Switching state of the cam track output
singleCamState	OUT	DWORD	16#0000	Bit mask or hexadecimal code for the selective display of which output cam of the cam track is active
parameter	IN_OUT	sLPac-ParaCam Track-Type	-	Parameter for cam track

1) Parameter types: IN = input parameter, OUT = output parameter, IN_OUT = in-out parameter

The units depend on the axis units. For linear axes, the unit mm is used; for rotary axes º.

Table 4- 192 sLParaCamTrackType structure

Element	Data type	Meaning
sTOs	sLPacToCamTrackDataType	TO for cam track
sCamTrackData	sLPacParaCamTrack-DataType	Parameter for cam track
sConfigData	sLPacConfigCamTrackType	Configuration data for cam track

Table 4- 193 sLPacToCamTrackDataType structure

Element	Data type	Initial value	Meaning
toCamTrack	camTrackType	TO#NIL	Technology object of the cam track

Table 4- 194 sLPacParaCamTrackDataType structure

Element	Data type	Initial value	Meaning
sCam	StructCamTrackArray-OfSingleCamSettings		Array for the parameters of the up to 32 single output cams of the cam track
r32ActivationTime	REAL	0.0	Actuation time on the activation edge
r32DeactivationTime	REAL	0.0	Actuation time on the deactivation edge
r32NoSwitchingRange	REAL	0.0	Hysteresis range

Note

The parameters of the *sLPacParaCamTrackDataType* structure are valid for all output cams of the cam track and are taken over at every activation of the cam track. The parameters are also taken over at every value change when the cam track is activated.

Table 4- 195 StructCamTrackArrayOfSingleCamSettings structure

Element	Data type	Initial value	Meaning
cam[]	ARRAY [0..31] OF StructCamTrackSingleCam-Settings	-	Array for the parameters of the up to 32 single output cams of the cam track

Table 4- 196 StructCamTrackArrayOfSingleCamSettings structure

Element	Data type	Initial value	Meaning
startPosition	LREAL	0.0	Activation position of the output cam
endPosition	LREAL	0.0	Deactivation position of the output cam (for position-based cams)
onDuration	LREAL	0.0	ON duration of the output cam (for time-based cams)
maxLength	LREAL	0.0	Maximum ON length of the time-based cam (for time-based cams with maximum length)
validity	EnumYesNo	YES	Validity

Table 4- 197 sLPacConfigCamTrackType structure

Element	Data type	Initial value	Meaning
r32IncShiftValue	REAL	1.0	Incremental value by which the output cam position is to be moved for the **shiftCamPositive, shiftCamNegative** functions, or by which the output cams are to be expanded or compressed for the **expandCam, compressCam** functions.
eInvertOutput	EnumYesNo	NO	Inversion of the switching logic
r32AxisReferencePosition	REAL	0.0	Reference position of the cam track
r32CamTrackLength	REAL	0.0	Length of the cam track
eActivationMode	EnumCamTrack-ActivationMode	CYCLIC	Specifies the cam track activation
eStartMode	EnumCamTrack-StartMode	IMMEDIATELY	Specifies when the cam track processing is activated
eStopMode	EnumCamTrack-StopMode	IMMEDIATELY	Specifies the deactivation of cam track processing

4.8.9.3 Embedding the module

Technology objects, basic parameterization and block call

Table 4- 198 TO type and basic parameterizations

Axis / technology object	
TO type	CamTrackType
Version	Position-based or time-based cams or time-based cams with maximum ON length
Modulo	Not relevant
Block call	BackgroundTask

Further blocks and libraries

The **FBLPacCamTrackControl** block does not use any further blocks.

Error messages

Error number [HEX]	Meaning
16#0000	LPAC_ERR_NO; error-free
16#8000	LPAC_ERR_INVALID_STEP; internal error, please get in touch with your Siemens contact partner
16#8003	LPAC_ERR_TO_ALARM_ACTIVE; TO alarm active; for the error description see Section Alarm and error messages in this manual
16#8017	LPAC_ERR_NO_CAMTRACK; the block has not been passed a "Cam track" technology object.
16#8102	LPAC_ERR_ENABLE_CAMTRACK; error from the _enableCamTrack command; return value of the SIMOTION _enableCamTrack function is equivalent to the *_move* function and can be fetched in hex format in the high word from *errorID*
16#8180	LPAC_ERR_DISABLE_CAMTRACK; error from the _disableCamTrack command; return value of the SIMOTION _disableCamTrack function is equivalent to the _move function and is in hex format in the high word from errorID
16#8181	LPAC_ERR_SET_CAMTRACK_STATE; error from the _setCamTrackState command; return value of the SIMOTION _setCamTrackState function is equivalent to the _move function and is in hex format in the high word from errorID

Note

Tip

The **pCamTrack** program unit is available in the LPac project to illustrate the use of the **FBLPacCamTrackControl** block.

This unit contains programs for the required block calls in the StartupTask and BackgroundTask.

4.8.10 FBLPacTorqueControl – torque and following error monitoring

4.8.10.1 Technological consideration

The **FBLPacTorqueControl** block is used to monitor and set the torque as well as for the following error monitoring on a real axis.

4.8.10.2 Module description

Scope of functions

The **FBLPacTorqueControl** function block provides the main and auxiliary functions, and possibly options, shown in the following overview.

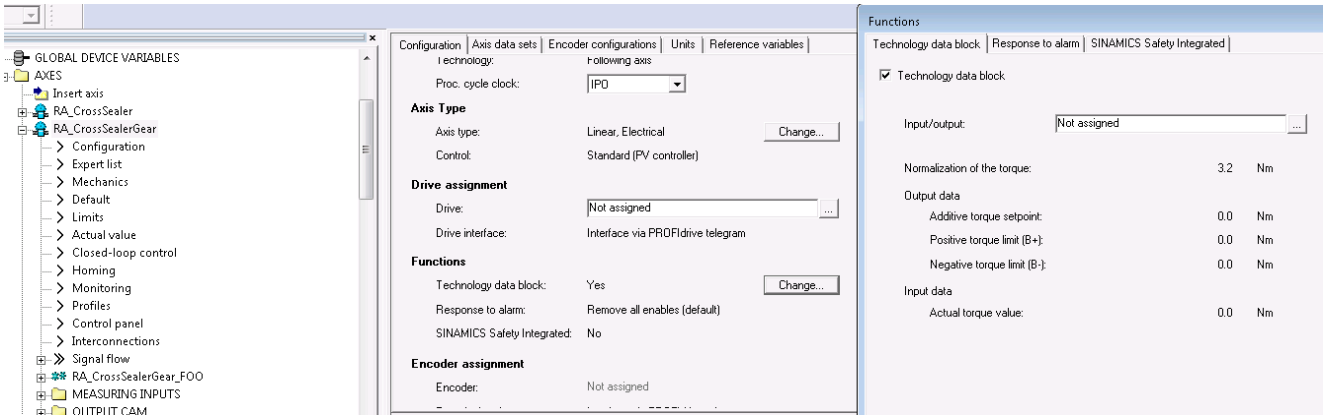
Function	Description	Implementation
Main function	Torque monitoring	The block cyclically monitors the torque utilization of a SINAMICS drive (parameter r81) passed on the **"actInTorqueRate"** input parameter for overshoot of a parameterizable limit value. At overshoot of the limit value, a message is output at the **"overload"** output. The user can also specify whether an internal OFF2 command or a positioning, e.g. to a "relief position", is to be activated.
Options	Torque setting	By activating the **"enableTorqueLimiting"** input signal, the block can activate a torque limit specified in the **"torqueLimitValue"** parameter in the SINAMICS drive. When torque reduction is activated, the **"torqueLimitReached"** output shows when the drive is working at the set torque limit.
	Following error monitoring	The block cyclically monitors the following error of a selected axis for overshoot of a parameterizable limit value. At overshoot of the limit value, a message is output at the **"followingError"** output. The user can also specify whether an internal OFF2 command or a positioning, e.g. to a "relief position", is to be activated.
Auxiliary functions	-	-

Note

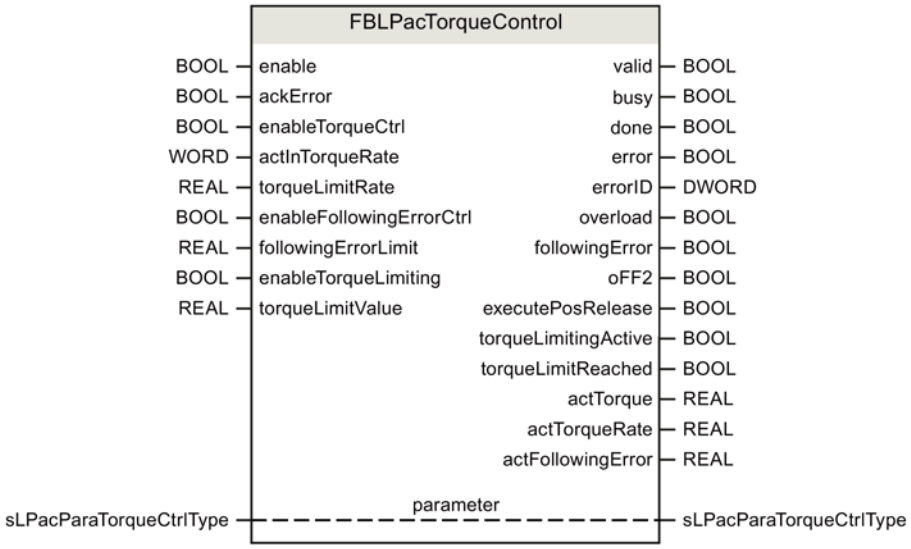
To monitor the torque utilization, the block must be cyclically passed the content of SINAMICS parameter r81. For this purpose, the parameter must be passed cyclically from the drive on a SIMOTION I/O input word via telegram coupling. The appropriate interconnection in the drive must be made by the user.

To activate a torque reduction, the so-called "technology data block" must be activated in SIMOTION in the configuration of the axis.

Example in the following figures.



Schematic block view



Input and output parameters

Table 4- 199 Input and output parameters

Element	P type [1]	Data type	Initial value	Meaning
enable	IN	BOOL	FALSE	Activation of the FB
ackError	IN	BOOL	FALSE	Acknowledge error
enableTorqueCtrl	IN	BOOL	FALSE	Enable monitoring of the torque utilization
actInTorqueRate	IN	WORD	16#0000	Torque utilization (r81 from the drive)
torqueLimitRate	IN	REAL	0.0	Torque utilization limit value
enableFollowingErrorCtrl	IN	BOOL	FALSE	Enable following error monitoring
followingErrorLimit	IN	REAL	0.0	Following error limit value
enableTorqueLimiting	IN	BOOL	FALSE	Enable torque reduction
torqueLimitValue	IN	REAL	0.0	Setpoint for torque reduction
valid	OUT	BOOL	FALSE	Block is active
busy	OUT	BOOL	FALSE	Selected functionality being processed
done	OUT	BOOL	FALSE	Selected functionality completed successfully
error	OUT	BOOL	FALSE	Error occurred
errorID	OUT	DWORD	16#0000_0000	Error code
overload	OUT	BOOL	FALSE	Torque utilization has reached or overshot the specified limit value
followingError	OUT	BOOL	FALSE	Following error has reached or overshot the specified limit value
oFF2	OUT	BOOL	FALSE	Internal OFF2 command has been triggered
executePosRelease	OUT	BOOL	FALSE	Activate positioning to "relief position"
torqueLimitingActive	OUT	BOOL	FALSE	Torque reduction activated on the drive
torqueLimitReached	OUT	BOOL	FALSE	Drive operating at the torque limit
actTorque	OUT	REAL	0.0	Current drive torque
actTorqueRate	OUT	REAL	0.0	Current drive torque utilization
actFollowingError	OUT	REAL	0.0	Current drive following error
parameter	IN_OUT	sLPacParaTorqueCtrlType	-	Parameters for torque monitoring, reduction and following error monitoring

1) Parameter types: IN = input parameter, OUT = output parameter, IN_OUT = in-out parameter

The units depend on the axis units. For linear axes, the unit mm is used; for rotary axes º.

Table 4- 200 sLPac ParaTorqueCtrlType structure

Element	Data type	Meaning
sTOs	sLPacToPosDataType	TO for axis
sMotionData	sLPacPosDataType	Parameter for positioning to relief position
sConfigData	sLPacConfigTorqueCtrl-Type	Configuration data for torque monitoring, reduction and following error monitoring

Table 4- 201 sLPacToPosDataType structure

Element	Data type	Initial value	Meaning
toAxis	PosAxis	TO#NIL	Position axis

Positioning parameters - sMotionDataa structure

See Position axis (Page 37)

Table 4- 202 sLPacConfigTorqueCtrlType structure

Element	Data type	Initial value	Meaning
boEnableOFF2ByOverload	BOOL	FALSE	Enable internal OFF2 command at overshoot of the set torque utilization
boEna-bleOFF2ByFollowingError	BOOL	FALSE	Enable internal OFF2 command at overshoot of the specified following error
r32TorqueDelayTime	REAL	0.0	Delay time for output and response at over-shoot of the specified limit value for the torque utilization
r32FolErrDelayTime	REAL	0.0	Delay time for output and response at over-shoot of the specified limit value for the fol-lowing error

4.8.10.3 Embedding the module

Technology objects, basic parameterization and block call

Table 4- 203 TO type and basic parameterizations

Axis / technology object	
TO type	At least position axis
Version	Not relevant

Modulo	Not relevant
Block call	ServoSynchronousTask or BackgroundTask depending on the requirement on the response time

Further blocks and libraries

The **FBLPacTorqueControl** block does not use any further blocks.

Error messages

Error number [HEX]	Meaning
16#0000	LPAC_ERR_NO; error-free
16#8000	LPAC_ERR_INVALID_STEP; internal error, please get in touch with your Siemens contact partner
16#8001	LPAC_ERR_NO_AXIS; no axis TO specified for own axis
16#8043	LPAC_ERR_ENABLE_TORQUE_LIMIT; error from the _enableTorqueLimiting command; return value of the SIMOTION _enableTorqueLimiting function is equivalent to the *_move* function and can be fetched in hex format in the high word from *errorID*
16#8044	LPAC_ERR_DISABLE_TORQUE_LIMIT; error from the _disableTorqueLimiting command; return value of the SIMOTION _disableTorqueLimiting function is equivalent to the *_move* function and can be fetched in hex format in the high word from *errorID*
16#8045	LPAC_ERR_MOTOR_OVERLOAD; limit value of the torque utilization overshot
16#8046	LPAC_ERR_AXIS_FOLLOWING_ERROR; limit value of the following error overshot

Note

Tip

The **pCrossSealerTorqueControl** program unit is available in the LPac project to illustrate the use of the **FBLPacTorqueControl** block.

This unit contains programs for the required block calls in the StartupTask and BackgroundTask.

4.8.11 FBLPacProductInSealDetection – product inclusion monitoring

4.8.11.1 Technological consideration

The **FBLPacProductInSealDetection** block is used to monitor and record the torque over a definable position range of a real axis. If a torque value other than a "normal value" is detected in this position range, this is signaled and the torque switched off on the axis with an OFF2 command. This is used, for example, to protect sealing jaws of a cross-sealer against damage when a product becomes trapped in the sealing area. The position range to be monitored is defined via an output cam

4.8.11.2 Module description

Scope of functions

The **FBLPacProductInSealDetection** function block offers the main functions and auxiliary functions as well as possible options shown in the following overview.

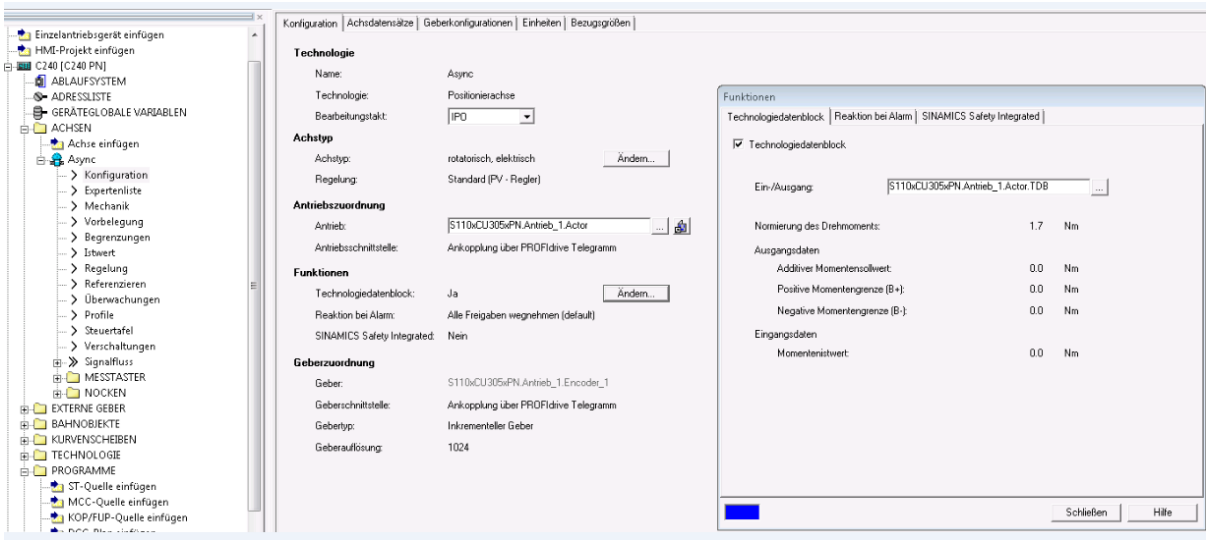
Function	Description	Implementation
Main function	Torque monitoring and rapid shutdown message	The block cyclically monitors the current torque signaled by the drive. If the torque associated with the current position of the axis is outside a preset tolerance, an error message is output. This can be used later for a rapid shutdown.
Options	-	-
Auxiliary functions	-	-

Note

To monitor the torque value, the technology data block function must be activated on the axis so that the actual torque values from the drive unit are transferred cyclically to SIMOTION.

In order to be able to use the functionality, you must ensure that the cam controller is active. Otherwise an error is output when attempting to switch on.

The output cam must also switch on and off cyclically. After the output cam has deactivated the output, mean values are calculated. For this reason, interruption-free monitoring is not possible. The monitored range should only be as large as required, as this reduces the computing power and the precision is still guaranteed. With very large monitoring ranges, not every application cycle is used for the evaluation and the monitoring is less precise.



Schematic representation in LAD/FBD

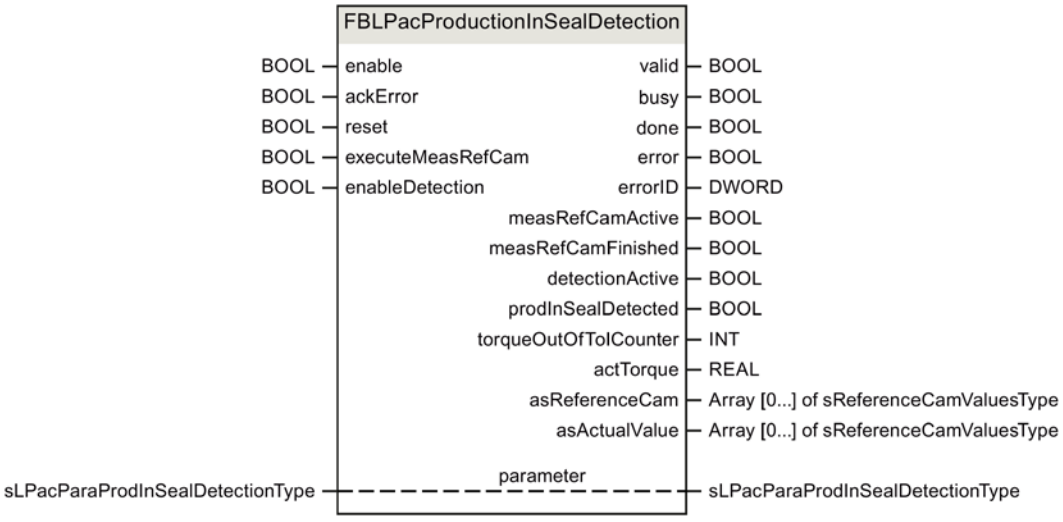


Figure 4-90 FBLPacProductInSealDetection – schematic block view in LAD/FBD

Input and output parameters

Table 4- 204 Input and output parameters

Element	P type [1)]	Data type	Initial value	Meaning
enable	IN	BOOL	FALSE	Activation of the FB
ackError	IN	BOOL	FALSE	Acknowledge error

Element	P type [1]	Data type	Initial value	Meaning
reset	IN	BOOL	FALSE	Reset reference values
executeMeasRef-Cam	IN	BOOL	FALSE	Start reference value acquisition
enableDetection	IN	BOOL	FALSE	Activation of the torque monitoring
valid	OUT	BOOL	FALSE	Block is active
busy	OUT	BOOL	FALSE	Selected functionality being processed
done	OUT	BOOL	FALSE	Selected functionality completed successfully
error	OUT	BOOL	FALSE	Error occurred
errorID	OUT	DWORD	16#0000 _0000	Error code
measRefCamActive	OUT	BOOL	FALSE	The reference value acquisition is active
measRefCamFinished	OUT	BOOL	FALSE	The reference value acquisition is finished
detectionActive	OUT	BOOL	FALSE	The torque monitoring is active
prodInSealDetected	OUT	BOOL	FALSE	Increased torque values have been determined, the corresponding drive has been switched off. Equivalent to error in conjunction with error number 16#8220
torqueOutOfTol-Counter	OUT	INT	0	Torque outside tolerance range counter
asReferenceCam	OUT	sReference CamValuesType	-	Reference values for comparison with current values
asActualValue	OUT	sReferenceCamValuesType	-	Actual values
parameter	IN_OUT	sLPacParaProdInSealDetectionType	-	Parameter

1) Parameter types: IN = input parameter, OUT = output parameter, IN_OUT = in-out parameter

The units depend on the axis units. For linear axes, the unit mm is used; for rotary axes º.

Table 4- 205 sReferenceCamValuesType structure

Element	Data type	Initial value	Meaning
r32JawsActPosition	REAL	0.0	Current position of the sealing jaws
r32JawsActTorque	REAL	0.0	Current torque of the sealing jaws

Table 4- 206 sLPacParaProdInSealDetectionType structure

Element	Data type	Meaning
sTOs	sLPacToAxisOutput-CamDataType	TOs for axis and output cam
sConfigData	sProdInSealDetectionConfig-DataType	Configuration data for the torque monitoring

Table 4- 207 sLPacToAxisOutputCamDataType structure

Element	Data type	Initial value	Meaning
toAxis	PosAxis	TO#NIL	Position axis
toOutputCam	OutputCam	TO#NIL	Output cam

Table 4- 208 sProdInSealDetectionConfigDataType structure

Element	Data type	Initial value	Meaning
boContinuesMode	BOOL	TRUE	When activated, the current values are taken over as reference values after cycles without a trapped product, so that the slowly changing torque values (e.g. through wear, pollution, etc.) are not detected as errors.
i16NumberOfMeasurements	INT	100	Number of measurements per product cycle Higher values increase the accuracy, but at the same time also the required computing time.
i16NumberOfTestSeries	INT	3	Number of cycles used to calculate the reference value. An increase here also increases the computing time
i16CyclesForMaxOverShoot	INT	5	Number of consecutive torque values used to calculate a mean value. A low value should be used for large position-dependent torque changes.
i16ConsecutiveOvershoots	INT	15	Number of consecutive torque overshoots until a product is detected. Is reset when the following torque value is within the tolerance range. Low values increase the danger of incorrect shutdown, high values increase the response time (n cycles of the task)
i16ConsecutiveDetectedProds	INT	1	Number of consecutive trapped products before tripping
r32MaxTorqueDifference	REAL	20.0	Permissible difference of the current torque value to the reference value in percent

4.8.11.3 Embedding the module

Technology objects, basic parameterization and block call

Table 4- 209 TO type and basic parameterizations

Axis / technology object	
TO type	At least position axis
Version	Not relevant
Modulo	Not relevant
Block call	ServoSynchronousTask or IPOSynchronousTask

Further blocks and libraries

The **FBLPacProductInSealDetection** block does not use any further blocks.

Error messages

Error number [HEX]	Meaning
16#0000	LPAC_ERR_NO; error-free
16#8000	LPAC_ERR_INVALID_STEP; internal error, please get in touch with your Siemens contact partner
16#8001	LPAC_ERR_NO_AXIS; no axis TO specified for own axis
16#8002	LPAC_ERR_NO_OUTPUT_CAM; no TO camController specified
16#80D0	LPAC_ERR_INVALID_INPUT_PARAMETER; invalid input parameters
16#8221	LPAC_ERR_OUTPUT_CAM_NOT_ENABLED; cam controller not active
16#8220	LPAC_ERR_PRODUCT_IN_SEAL_DETECTED; overload or trapped product detected
16#8222	LPAC_ERR_MEASUREMENT_ARRAY_TOO_SMALL;

Note

The **pProductInSealDetection** program unit is available in the LPac project to illustrate the use of the **FBLPacProductInSealDetection** block.

This unit contains programs for the required block calls in the StartupTask, the ServoSynchronousTask and the BackgroundTask.

4.8.12 FBLPacReadSinamicsMessages – read drive faults

4.8.12.1 Technological consideration

The **FBLPacReadSinamicsMessages** block is used to read out faults, alarms and Safety Integrated messages from SINAMICS drives.

4.8.12.2 Module description

Scope of functions

The **FBLPacReadSinamicsMessages** function block offers the main functions and auxiliary functions as well as possible options shown in the following overview.

Function	Description	Implementation
Main function	Reads faults, alarms and messages from any Safety Integrated Extended Functions used in the drive	With appropriate parameterization, the block cyclically calls drive faults, alarms and messages of the Safety Integrated Extended Functions; the corresponding time stamp of the event is still output for the "incoming" and "outgoing" times. A group error bit is output for simple error detection for each event group (fault, alarm, SI message).
Options	Acknowledgment of pending faults, alarms and SI messages	All events to the drive are acknowledged/reset by controlling the "ackSinamicsError" input.
Auxiliary functions	-	-

Schematic representation in LAD/FBD

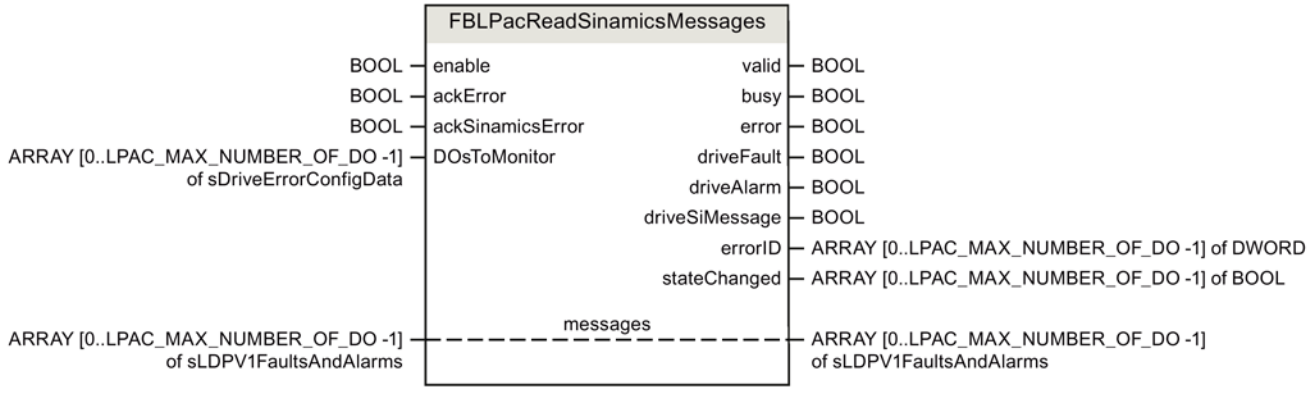


Figure 4-91 FBLPacReadSinamiceMessages – schematic block view in LAD/FBD

Input and output parameters

Table 4- 210 Input and output parameters

Element	P type 1)	Data type	Initial value	Meaning
enable	IN	BOOL	FALSE	Activation of the FB
ackError	IN	BOOL	FALSE	Acknowledge block internal errors
ackSinamicsError	IN	BOOL	FALSE	Acknowledge drive faults
DOsToMonitor	IN	ARRAY [0..LPAC_MAX_NUMBER_OF_DO -1] of sDriveErrorConfigData	-	Description data of the drive objects (drives) to be checked
valid	OUT	BOOL	FALSE	Block is active
busy	OUT	BOOL	FALSE	Selected functionality being processed
error	OUT	BOOL	FALSE	Block error has occurred
driveFault	OUT	BOOL	FALSE	Group error, drive fault active
driveAlarm	OUT	BOOL	FALSE	Group error, drive alarm active
driveSiMessage	OUT	BOOL	FALSE	Group error Safety Integrated message active
errorID	OUT	ARRAY [0..LPAC_MAX_NUMBER_OF_DO -1] of DWORD	16#0000	Error code
stateChanged	OUT	ARRAY [0..LPAC_MAX_NUMBER_OF_DO -1] of BOOL	FALSE	Pulse status of the faults, alarms or SI messages has changed (only one cycle active)
messages	IN_OUT	ARRAY [0..LPAC_MAX_NUMBER_OF_DO -1] of sLDPV1FaultsAndAlarms	-	Structure of the drive faults, alarms and Safety Integrated messages

1) Parameter types: IN = input parameter, OUT = output parameter, IN_OUT = in-out parameter

Table 4- 211 sDriveErrorConfigData structure

Element	Data type	Meaning
eTypeOfDO	eLDPV1GetFaultDoType	Specifies the type of the drive object to be monitored. The following values are possible: DO_WITH_TO DO_WITH_CYCLIC_ZSW1 DO_WITHOUT_CYCLIC_ZSW1
toAxis	DriveAxis	Technology object of the axis

b32IOVariable ZSW1	WORD	Input variable that was assigned to the address of the ZSW1 in the I/O area in the cyclic communication. Applies for drive objects without technology objects, but with cyclic communication
i32LogAdress	DINT	Logical address of the drive object Different cases depending on **eTypeOfDO** 1. DO_WITH_TO Not relevant 1. DO_WITH_CYCLIC_ZSW1 Logical base address of the drive object 1. DO_WITHOUT_CYCLIC_ZSW1 Any logical address of the device on which the drive object is located. If no DO number can be passed, the logical base address must be passed
u8doNumber	USINT	DO number of the drive object to be monitored. Three different cases: 1. DO_WITH_TO Not relevant 1. DO_WITH_CYCLIC_ZSW1 Not relevant 1. DO_WITHOUT_CYCLIC_ZSW1 Drive object number of the drive object to be monitored without cyclic status word. If the DO is passed via the base address of the drive in i32LogAdress, the doNumber does not have to be supplied or can be set to 255
tRestartDelay	TIME	Wait time that must pass between two monitoring cycles. It is therefore the time that must pass between two monitoring attempts.

Table 4- 212 sLDPV1FaultsAndAlarms structure

Element	Data type	Meaning
boFaultActive	BOOL	1. Fault active on the object
i16NumberOfFaults	INT	Number of currently active faults
au16FaultCode[0..LDPV1_MAX_NO_OF_INFORMATION]	ARRAY OF UINT	Output of max. eight most recent fault messages from SINAMICS parameter r945
ai32FaultInfo[0..LDPV1_MAX_NO_OF_INFORMATION]	ARRAY OF DINT	Output of the additional information from SINAMICS parameter r949
adtFaultComeDateAnd-And-Time[0..LDPV1_MAX_NO_OF_INFORMATION]	ARRAY OF DT	Output of the system time at the time of the "incoming fault", either in the "Time since <switch on>" unit or in the UTC format in the Date_And_Time data format from the SINAMICS parameters r2109/2136
adtFaultGoneDateAnd-And-Time[0..LDPV1_MAX_NO_OF_INFORMATION]	ARRAY OF DT	Output of the system time at the time of the "fault corrected", either in the "Time since <switch on>" unit or in the UTC format in the Date_And_Time data format from the SINAMICS parameters r2109/2136

toAlarmActive	BOOL	Alarm active on the object
i16NumberOfAlarms	INT	Number of currently active alarms (only the active alarms with filteredAlarmInfo = TRUE)
au16AlarmCode[0..LDPV1_MAX_NO_OF_INFORMATION]	ARRAY OF UINT	Output of max. eight most recent alarm messages from SINAMICS parameter r2122
ai32AlarmInfo[0..LDPV1_MAX_NO_OF_INFORMATION]	ARRAY OF DINT	Output of the additional information from SINAMICS parameter r2124
adtAlarmComeDateAnd-And-Time[0..LDPV1_MAX_NO_OF_INFORMATION]	ARRAY OF DT	Output of the system time at the time of the "incoming alarm", either in the "Time since <switch on>" unit or in the UTC format in the Date_And_Time data format from the SINAMICS parameters r2123/2145
adtAlarmGoneDateAnd-And-Time[0..LDPV1_MAX_NO_OF_INFORMATION]	ARRAY OF DT	Output of the system time at the time of the "outgoing alarm", either in the "Time since <switch on>" unit or in the UTC format in the Date_And_Time data format from the SINAMICS parameters r2125/2146
boSIMessageActive	BOOL	SI message active on the object
i16NumberOfMessages	INT	Number of currently active messages
au16MessageCode[0..LDPV1_MAX_NO_OF_INFORMATION]	ARRAY OF UINT	Output of max. eight most recent messages from SINAMICS parameter r9747
ai32MessageInfo[0..LDPV1_MAX_NO_OF_INFORMATION]	ARRAY OF DINT	Output of the additional information from SINAMICS parameter r9749
ar32MessageInfo[0..LDPV1_MAX_NO_OF_INFORMATION]	ARRAY OF REAL	Output of the additional information from SINAMICS parameter r9753 in the Float format
adtMessageComeDateAnd-And-Time[0..LDPV1_MAX_NO_OF_INFORMATION]	ARRAY OF DT	Output of the system time at the time of the "incoming message", either in the "Time since <switch on>" unit or in the UTC format in the Date_And_Time data format from the SINAMICS parameters r9745/9748
adtMessageGoneDateAnd-And-Time[0..LDPV1_MAX_NO_OF_INFORMATION]	ARRAY OF DT	Output of the system time at the time of the "outgoing message", either in the "Time since <switch on>" unit or in the UTC format in the Date_And_Time data format from the SINAMICS parameters r9756/9755

4.8.12.3 Embedding the module

Technology objects, basic parameterization and block call

Table 4- 213 TO type and basic parameterizations

Axis / technology object	
TO type	Not relevant
Version	Not relevant
Modulo	Not relevant
Block call	BackgroundTask / cyclic

Further blocks and libraries

The **FBLPacReadSinamicsMessages** block calls the **FBLDPV1GetDriveObjectsAlarmsAndFaults** block internally from the LDPV1 library.

The block is available in the DPV1Library_LDPV1 equipment module.

SIMOTION easyProject ProjectGenerator (https://support.industry.siemens.com/cs/ww/en/view/51339107)

https://support.industry.siemens.com/cs/ww/en/view/62049135

Error messages

Error number [HEX]	Meaning
16#0000	LPAC_ERR_NO; error-free
16#8000	LPAC_ERR_INVALID_STEP; internal error, please get in touch with your Siemens contact partner

Note

The **pLPacReadSinamicsMessages** program unit is available in the LPac project to illustrate the use of the **FBLPacReadSinamicsMessages** block.

This unit contains programs for the required block calls in the StartupTask and BackgroundTask.

4.9 Temperature control

4.9.1 FBLPacTOTempCtrl - SIMOTION technology object

4.9.1.1 Technological consideration

The **FBLPacToTempCtrl** block implements a temperature control via a PID controller and actuating signal output via a pulse-width modulated digital output. The "temperature channel" technology object integrated in SIMOTION is used. A self-tuning routine is also implemented in this technology object.

The block provides a simple interface to the technology object and communicates via system functions with the assigned temperature channel technology object. You can use the block to activate/deactivate the temperature channel (controller) and to perform a self-tuning. You can also specify the controller parameters, the temperature setpoint and the temperature tolerances.

Note

A detailed description of the temperature channel is available in the **SIMOTION System and Function Descriptions Function Manual, Part 8: SIMOTION Motion Control, Supplementary Technology Functions** An overview of all system variables and configuration data is available in the **SIMOTION Programming - References List Manual, Part 18: SIMOTION TControl technology package**.

4.9.1.2 Module description

Scope of functions

The **FBLPacTOTempCtrl** function block provides the main and auxiliary functions, and possibly options, shown in the following overview.

Function	Description	Implementation
Main function	Temperature control	The temperature controller is switched on/off via the level-triggered *enableController* input.
Options	-	-
Auxiliary functions	Self-tuning the temperature controller	The controller self-tuning is started via the level-triggered *enableSelfTune* input.

The standard configuration requires an analog input for the measured value acquisition for each temperature channel. The manipulated variable is output pulse-width modulated at a digital output. The controlling element should be activated via semiconductor relays.

The block checks whether a parameter (controller gain, integral time, derivative time) or setpoint (temperature setpoint, tolerance limit) has changed.

If at least one value changes for the temperature channel, the FB supplies the associated technology object with the new value.

The specifications as described in the *SIMOTION Motion Control, Supplementary Technology Objects* System and Function Description govern the self-tuning.

You should check some boundary conditions before starting the self-tuning of the TO.

Check the parameterized start conditions for the self-tuning and the minimum setpoint step increment. The relevant configuration data is contained in the *TO_Temperature.identification.modifiedTangentMethod* structure.

The self-tuning routine is set by the block so that individual tuning states are advanced automatically. (Configuration data *identification.modifiedTangentMethod.transitionMode* = AUTOMATICALLY)

Once the self-tuning has been completed successfully, the determined controller parameters are taken over directly.

Configuration

You can access all data that belongs to the **temperature channel TO** via the expert list selection. Basic settings are saved directly in the expert list. The interface supplies only the data required for running operation. A detailed description of the interface is provided at the end of the section.

Note

The complete functionality of the temperature channel is not described within this section. The description refers to the functionality of a PID standard controller with autotuning (system identification)

Detailed descriptions of limit values, error messages and configurations are included in the **SIMOTION System and Function Description Function Manual, Section 8: SIMOTION Motion Control, Supplementary Technology Functions**

Schematic representation in LAD/FBD

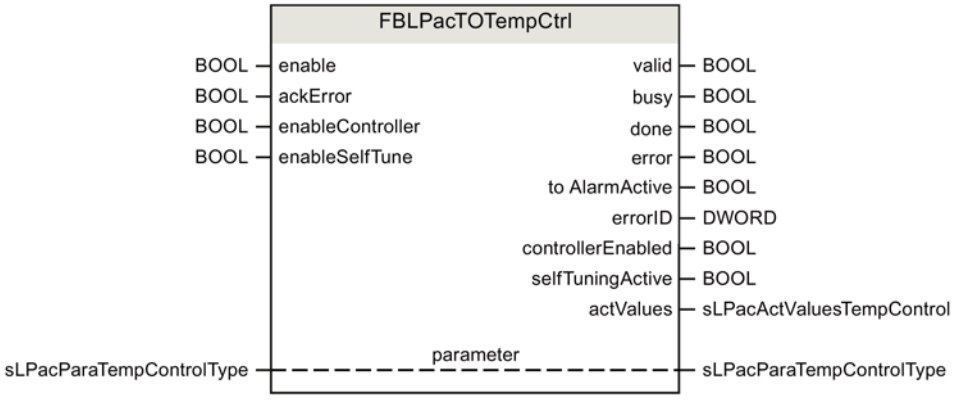


Figure 4-92 FBLPacTOTempCtrl - schematic block view in LAD/FBD

Input and output parameters

Table 4- 214 Input and output parameters

Element	P type [1)]	Data type	Initial value	Meaning
enable	IN	BOOL	FALSE	Activation of the FB
ackError	IN	BOOL	FALSE	Acknowledge error
Enable Controller	IN	BOOL	FALSE	Temperature controller enable
enable-SelfTune	IN	BOOL	FALSE	Self-tuning enable
valid	OUT	BOOL	FALSE	Block selected and can be used
busy	OUT	BOOL	FALSE	Selected functionality being processed

Element	P type [1)]	Data type	Initial value	Meaning
done	OUT	BOOL	FALSE	Selected functionality completed successfully
error	OUT	BOOL	FALSE	Error occurred
errorID	OUT	DWORD	16#0000 _0000	Error code
toAlarmActive	OUT	BOOL	FALSE	Technological alarm active; the errorGroup is issued at the errorID variable; see SIMOTION Manual *Basic Functions*, Section 6.2 *Technological alarms*; the pending technological alarm can be fetched via the *TaskStartInfo* of the *TecFaultTask*. The TO alarm must also be acknowledged there. If the TO alarm is acknowledged, the *toAlarmActive* output is cleared again.
Controller Enabled	OUT	BOOL	FALSE	Controller enabled, closed-loop control active
selfTuning Active	OUT	BOOL	FALSE	Controller self-tuning active
actValues	OUT	sLPacActValues TempControlType	-	Current operating data of the temperature channel
parameter	IN_OUT	sLPac-ParaTemp ControlType	-	Parameter for temperature control

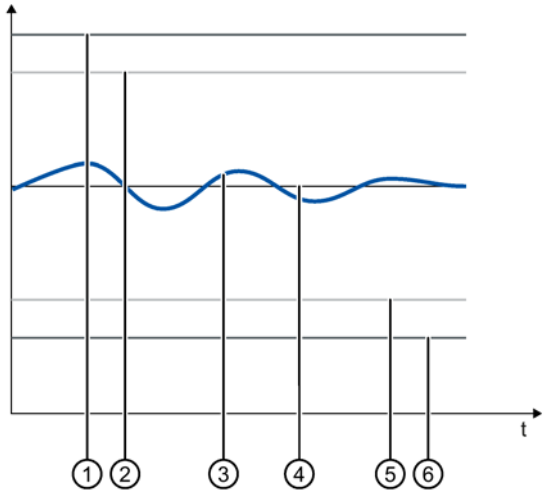
Parameter types: IN = input parameter, OUT = output parameter, IN_OUT = in-out parameter

Table 4- 215 sLPacParaTempControlType structure

Element	Data type	Initial value	Meaning
toTempController	Temperature ControllerType	TO#NIL	Technology object of the temperature channel
r32TempSetpoint	REAL	0.0	Temperature setpoint
r32PIDGain	REAL	0.0	Proportional gain
r32PIDIntegTime	REAL	0.0	Reset time
r32PIDDerivTime	REAL	0.0	Actuation time
r32HighLimit2	REAL	0.0	2nd (upper outer) temperature limit
			Value = 0.0: Limit value inactive
			Value > 0.0: Limit value relative
r32HighLimit1	REAL	0.0	1st (upper inner) temperature limit
			Value = 0.0: Limit value inactive
			Value > 0.0: Limit value relative
r32LowLimit1	REAL	0.0	1st (lower inner) temperature limit
			Value = 0.0: Limit value inactive
			Value > 0.0: Limit value relative

Element	Data type	Initial value	Meaning
r32LowLimit2	REAL	0.0	2nd (lower outer) temperature limit
			Value = 0.0: Limit value inactive
			Value > 0.0: Limit value relative

The following figure shows the inner and outer tolerance bands.



① 2nd upper tolerance, outer upper tolerance
② 1st upper tolerance, inner upper tolerance
③ Actual value
④ Setpoint
⑤ 1st lower tolerance, inner lower tolerance
⑥ 2nd lower tolerance, outer lower tolerance

Table 4- 216 sLPacActValuesTempControlType structure

Element	Data type	Initial value	Meaning
r32ActualTemp	REAL	0.0	Actual temperature value
r32OutputValue	REAL	0.0	Current output value
enumSelfTuningState	enumTControllerIdentificationModifiedTangentMethodStage	INACTIVE	Operating state of the temperature channel
u32ErrorState	UDINT	0	Current value of the TO system variable actualinputdata.state

4.9.1.3 Integration of the module

Technology objects, basic parameterization and block call

Table 4- 217 TO type

Technology object	
TO type	TemperatureControllerType
Block call	MotionTask

When integrating the **FBLPacTOTempCtrl** block, note that the block must not be called in a time-monitored task because block-internal system commands are used with the "WHEN_COMMAND_DONE" step enabling condition. **The block should preferably be called in a motion task.**

Further blocks and libraries

The **FBLPacTOTempCtrl** block does not use any further blocks.

Error messages

Error number [HEX]	Meaning
16#0000	LPAC_ERROR_NO; error-free
16#8120	LPAC_ERR_SETTCONTROLLERIDENTIFICATIONMODIFIEDTANGENTMETHODPARAMETER; error of the _setTControllerIdentificationModifiedtangentmethodParameter system function; return value see _setTempControllerOperatingMode (see *Technology Packages System Functions SIMOTION Manual,* Section 3.1.2.6)
16#8121	LPAC_ERR_SETTCONTROLLEROPERATINGMODE; error of the _setTControllerOperatingMode system function; return value see _setTempControllerOperatingMode (see *Technology Packages System Functions SIMOTION Manual,* Section 3.1.2.6)
16#8122	LPAC_ERR_SETTCONTROLLERPIDPARAMETER; error of the _setTControllerPidParameter system function; return value see _setTempControllerOperatingMode (see *Technology Packages System Functions SIMOTION Manual,* Section 3.1.2.6)
16#8123	LPAC_ERR_SETTCONTROLLERINPUTLIMITCHECKPARAMETER; error of the _setTControllerInputLimitCheckParameter system function; return value see _setTempControllerOperatingMode (see *Technology Packages System Functions SIMOTION Manual,* Section 3.1.2.6)
16#8124	LPAC_ERR_SETTCONTROLLERSETPOINT; error of the _setTControllerSetPoint system function; return value see _setTempControllerOperatingMode (see *Technology Packages System Functions SIMOTION Manual,* Section 3.1.2.6)
16#8125	LPAC_ERR_CALCULATETCONTROLLERPARAMETER; error of the _setTControllerSetPoint system function; return value see _calculateTControllerParameter (see *Technology Packages System Functions SIMOTION Manual,* Section 3.1.2.6)

Error number [HEX]	Meaning
16#8126	LPAC_ERR_NO_TO_TEMP_CTRL; no reference to a temperature channel technology object passed to the block
Bit position-coded	When a technological alarm is pending, the errorGroup is output; see *SIMOTION Basic Functions Manual*, *Section 6.2 Technological alarms*

Note

The **pTOTempControl** program unit is available in the LPac project to illustrate the use of the **FBLPacTOTempCtrl** block.

This unit contains the programs for the required block calls in the TimerInterruptTask.

4.9.2 FBLPacPIDTempCtrl – SIMOTION PID controller

4.9.2.1 Technological consideration

The **FBLPacToTempCtrl** block implements a simple temperature control via a PID controller and actuating signal output via a pulse width-modulated digital output. The actual temperature value can be supplied via an analog input (thermocouple or PT100 resistance thermometer). For the pulse width-modulated actuating signal output, the FBLPacPWMCtrl block can be combined with the FBLPacPIDTempCtrl controller block.

The **FBLPacPIDTempCtrl** block provides an interface for the PID controller block. Some supplementary functions, such as monitoring the temperature rise (sensor monitoring), monitoring for warning and limiting values as well as a switchover of the controller parameters (gain and reset time) are implemented depending on the control deviation.

A self-tuning routine does not exist for this block.

4.9.2.2 Module description

Scope of functions

The **FBLPacPIDTempCtrl** function block provides the main and auxiliary functions, and possibly options, shown in the following overview.

Function	Description	Implementation
Main function	Temperature control	Simple PID temperature control with analog setpoint channel (e.g. PT100) and analog actuating signal to a contactor control block for the pulse width-modulated control of a digital output (e.g. FBLPacPWMCtrl).
Options	Parameter switchover	Depending on a parameterizable control deviation, various control parameters (gain and reset time) are switched automatically.

Function	Description	Implementation
Auxiliary functions	Sensor monitoring, temperature rise monitoring	This monitors whether the actual temperature value has overshot the lower alarm value within a parameterizable time after activating the controller. An external "sensor fault" signal can also be evaluated.
-	Alarm and limit value monitoring	Monitoring is performed whether the actual temperature value lies within a tolerance band. The tolerance band is defined by lower and upper alarm values as well as lower and upper limit values. Messages are issued when the alarm and limits values are overshot or undershot. When the limit values are overshot or undershot, error messages are issued after a parameterizable delay time and the controller is shutdown.

Schematic representation in LAD/FBD

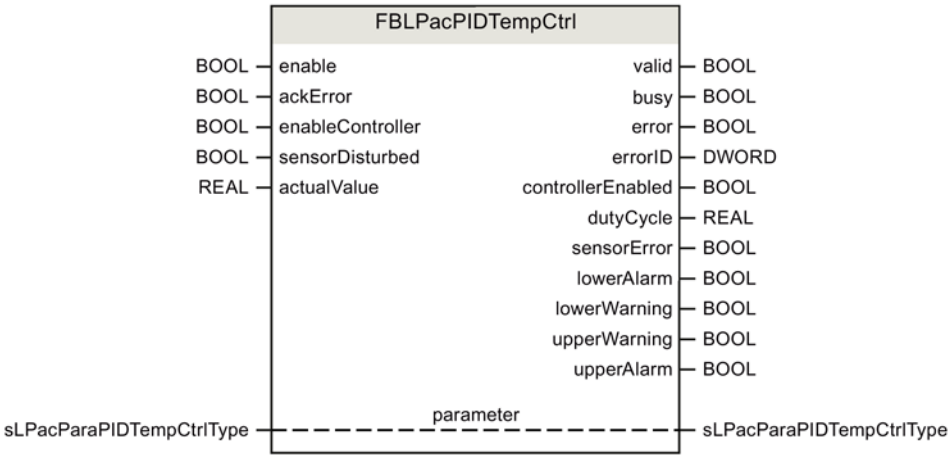


Figure 4-93 FBLPacPIDTempCtrl - schematic block view in LAD/FBD

Input and output parameters

Table 4- 218 Input and output parameters

Element	P type [1)]	Data type	Initial value	Meaning
enable	IN	BOOL	FALSE	Activate the block
ackError	IN	BOOL	FALSE	Acknowledge error
enableController	IN	BOOL	FALSE	Controller enable
sensorDisturbed	IN	BOOL	FALSE	Infeed of an externally generated "temperature sensor faulty" signal. If the external signal is pending, the temperature controller is switched off and the "SensorError" error message issued.
actualValue	IN	REAL	0.0	Actual temperature value (in the same unit as the setpoint)

Element	P type [1]	Data type	Initial value	Meaning
valid	OUT	BOOL	FALSE	Block is active
busy	OUT	BOOL	FALSE	Controller enabled and no error (control mode active)
error	OUT	BOOL	FALSE	Error occurred (group error)
errorID	OUT	DWORD	16#0000_0000	Error code, see table; output made in hex format!
control-lerEnabled	OUT	BOOL	FALSE	Temperature controller enabled
dutyCycle	OUT	REAL	0.0	Duty cycle in %
sensorError	OUT	BOOL	FALSE	Temperature increase error or sensor error
lowerAlarm	OUT	BOOL	FALSE	Lower alarm value undershot
lowerWarning	OUT	BOOL	FALSE	Lower warning value undershot
upperWarning	OUT	BOOL	FALSE	Upper warning value overshot
upperAlarm	OUT	BOOL	FALSE	Upper alarm value overshot
parameter	IN_OUT	sLPac-ParaPIDTempCtrlType	-	Parameter structure of the integrated PID controller block _CTRL_pid

[1] Parameter types: IN = input parameter, OUT = output parameter, IN_OUT = in-out parameter

Table 4- 219 sLPacParaPIDTempCtrlType structure

Element	Data type	Initial value	Meaning
r32SetpointLowerAlarm [2]	REAL	0.0	Lower limit value in the tolerance band of the actual value. If the actual value undershoots the lower limit value, after a parameterized delay time, an error message is issued ("error" and "lowerAlarm" are output at the FB output), Id 80F0 is displayed at the errorId output and the controller is shutdown.
r32SetpointLowerWarning [2]	REAL	0.0	Lower warning value in the tolerance band of the actual value. If the actual value undershoots the lower warning value, a warning message is issued immediately (at the "low-erWarning" FB output).
r32SetpointUpperWarning [2]	REAL	0.0	Upper warning value in the tolerance band of the actual value. If the actual value overshoots the upper warning value, a warning message is issued immediately (at the "upper-Warning" FB output).
r32SetpointUpperAlarm [2]	REAL	0.0	Upper limit value in the tolerance band of the actual value. If the actual value overshoots the upper limit value, after a parameterized delay time, an error message is issued ("error" and "upperAlarm" are output at the FB output), errorId 80F1 displayed at the output and the controller is shutdown.

Element	Data type	Initial value	Meaning
r32PropGainHigh [2]	REAL	0.0	Startup value proportional gain If the deviation between the setpoint and the actual value overshoots the value parameterized in "r32DeltaToSwtchPara" (e.g. when starting the controller), the startup value of the gain parameterized here is used automatically as control parameter.
r32PropGainLow [2]	REAL	0.0	Proportional gain operating value If the deviation between the setpoint and the actual value undershoots the value parameterized in "r32DeltaToSwtchPara" (e.g. actual value steady in the tolerance band), the operating value of the gain parameterized here is used automatically as control parameter.
i32IntegTimeHigh [2]	DINT	0	Integration time operating value in ms If the deviation between the setpoint and the actual value undershoots the value parameterized in "r32DeltaToSwtchPara" (e.g. actual value steady in the tolerance band), the operating value of the integration time parameterized here is used automatically as control parameter.
i32IntegTimeLow [2]	DINT	0	Integration time startup value in ms If the deviation between the setpoint and the actual value overshoots the value parameterized in "r32DeltaToSwtchPara" (e.g. when starting the controller), the startup value of the integration time parameterized here is used automatically as control parameter.
r32DeltaToSwtchPara [2]	REAL	0.0	Switchover threshold value for switching the controller parameters from startup parameters to operating parameters in °C. If the deviation between the setpoint and the actual value undershoots the value specified in "r32deltaToSwtchPara", the startup parameters are used, otherwise the operating parameters.
i32TemperatureRiseTime [2]	DINT	300000	Temperature increase time monitoring in ms If no temperature increase above the lower warning value is performed within the time interval parameterized here, the "error" and "SensorError" messages are output with errorID 80F2 and the controller is shutdown again.
i32AlarmDelayTime [2]	DINT	5000	Temperature alarm delay time in ms If the actual temperature value leaves the upper or lower alarm threshold, an error message is issued after the time interval parameterized here and the controller is shutdown.
boManualMode [1]	BOOL	FALSE	Activate manual mode If "boManualMode" is set, the control loop is interrupted. The value at "r32manualValue" is specified as manipulated variable.

Element	Data type	Initial value	Meaning
boReset 1)	BOOL	FALSE	Initialize _Ctrl_pid block If "boReset" is set, all outputs of the lower-level PID block are set to FALSE or zero and the integrator set to the "r32InitialIValue" value.
boActValueType 1)	BOOL	FALSE	Activate actual I/O value If the actual value should be read from the I/O, "b16binActValue" must be interconnected to the I/O and "boActValueType" set to True:
boSetPController 1)	BOOL	TRUE	Activate P component The PID components can be activated and deactivated individually in the PID algorithm. The P component is activated when "boSetPController" is set.
boSetIController 1)	BOOL	TRUE	Activate I component The PID components can be activated and deactivated individually in the PID algorithm. The I component is activated when "boSetIController" is set.
boHoldIValue 1)	BOOL	FALSE	Freeze I component The integrator output can be frozen. This requires that "boHoldIValue" is set.
boSetIValue 1)	BOOL	FALSE	Set I component The integrator output can be set to the "r32InitialIValue" input. This requires that "boSetIValue" is set.
r32InitialIValue 1)	REAL	0.0	Initialization value for I component The integrator output can be set to the "r32InitialIValue" initialization value with the "boSetIValue" input.
boSetDController 1)	BOOL	FALSE	Activate D component The PID components can be activated and deactivated individually in the PID algorithm. The D component is activated when "boSetDController" is set.
i32CycleTime 1)	DINT	300	Sampling time of the controller block in ms. The time between block calls must be constant. "i16CycleTime" specifies the sampling time between two block calls in ms.
r32SetpointValue 1)	REAL	0.0	Temperature setpoint. "r32SetPointValue" specifies the temperature setpoint. The setpoint must be specified in the same unit as the actual value (either -100.0…+100.0 (%) or in °C).
r32NumericActValue 1)	REAL	0.0	Actual temperature value. "r32NumericActValue" specifies the actual temperature value. The actual value must be specified in the same unit as the setpoint (either -100.0…+100.0 (%) or in °C).
b16BinActValue 1)	WORD	16#0000	Actual temperature value of the I/O devices As alternative to the interconnection of the actual value to "r32NumericActValue", the actual value can be interconnected to "b16BinActValue" in I/O format.

Element	Data type	Initial value	Meaning
r32ManualValue 1)	REAL	0.0	Temperature manual manipulated variable. "r32ManualValue" specifies the input of a manual manipulated variable that can be output by setting "boManualMode".
i32DerivativeTime 1)	DINT	10000	Derivative time in ms "i32DerivativeTime" specifies the runtime behavior of the differentiator. The value must be ≥ i32CycleTime!
i32DelayTime 1)	DINT	2000	Delay time of D component in ms The algorithm of the D component includes a delay that can be parameterized at "i32DelayTime". The value must be ≥ i32cycleTime / 2!
r32DeadBand 1)	REAL	0.0	Dead band width The control deviation is routed over a dead band. "r32DeadBand" specifies the size of the dead band. The value must be specified in the same unit as the setpoint (either -100.0…+100.0 (%) or in °C).
r32UpperLimit 1)	REAL	100.0	Upper limit of the manipulated variable The manipulated variable is always limited to an upper limit and a lower limit. "r32UpperLimit" specifies the upper limit. The value must be specified in the same unit as the manipulated variable (either -100.0…+100.0 (%) or in °C).
r32LowerLimit 1)	REAL	0.0	Lower limit of the manipulated variable The manipulated variable is always limited to an upper limit and a lower limit. "r32LowerLimit" specifies the lower limit. The value must be specified in the same unit as the manipulated variable (either -100.0…+100.0 (%) or in °C).
r32ActValueFactor 1)	REAL	1.0	Actual value factor The actual value is multiplied by "r32ActValueFactor" to adapt the actual value range.
r32ActValueOffset 1)	REAL	0.0	Actual value offset "r32ActValueOffset" is added to the actual value for the adaptation.
r32OutValueFactor 1)	REAL	1.0	Manipulated variable factor The manipulated variable is multiplied by "r32OutValueFactor" to adapt the manipulated variable range.
r32OutValueOffset 1)	REAL	0.0	Manipulated variable offset "r32OutValueOffset" is added to the manipulated variable for the adaptation.
r32DisturbValue 1)	REAL	0.0	Disturbance variable A disturbance variable can be applied to "r32DisturbValue". The value must be specified in the same unit as the manipulated variable (either -100.0…+100.0 (%) or in °C).

1) Standard parameter of the _CTRL_pid block. For a detailed description, see the SIMOTION Basic Control Function Manual or the online help for the block.

2) Specific supplementary parameter of the LPac FBLPacPIDTempCtrl block

4.9.2.3 Integration of the module

Technology objects, basic parameterization and block call

Table 4- 220 TO type

Technology object	
TO type	Not relevant
Block call	TimerTask

Further blocks and libraries

The **FBLPacPIDTempCtrl** block uses the following blocks internally.

Block/library	Function
_CTRL_pid / SIMOTION command library	PID controller

Detailed descriptions of individual sub-functions and parameters of the SIMOTION internal PID controller block are available in the *SIMOTION Basic Control* Function Manual and in the online help for the _CTRL_pid block.

Error messages

Error number [HEX]	Meaning
16#0000	LPAC_ERR_NO; error-free
16#8000	LPAC_ERR_INVALID_STEP; internal error, please get in touch with your Siemens contact partner
16#8110	LPAC_ERR_TEMPERATURE_LOWER_ALARM; lower limit value undershot
16#8111	LPAC_ERR_TEMPERATURE_UPPER_ALARM; upper limit value overshot
16#8112	LPAC_ERR_TEMPERATURE_SENSOR; sensor error, no temperature increase in the parameterized time after controller enable

Note

Tip

The **pPIDTempControl** program unit is available in the LPac project to illustrate the use of the **FBLPacPIDTempCtrl** and **FBLPacPWMCtrl** blocks.

This unit contains the programs for the required block calls in the TimerInterruptTasks.

4.9.3 FBLPacPWMCtrl - pulse width modulation

4.9.3.1 Technological consideration

The **FBLPacPWMCtrl** block implements a pulse width-modulated activation of a digital output of a simple temperature control. The **FBLPacPIDTempCtrl** block can be combined with this block as control loop block.

4.9.3.2 Module description

Scope of functions

The **FBLPacPWMCtrl** function block provides the main and auxiliary functions, and possibly options, shown in the following overview.

Function	Description	Implementation
Main function	Pulse width modulation	Pulse width-modulated activation of a digital signal depending on the specified duty cycle. The controlling element should be activated via semiconductor relays.
Options	-	-
Auxiliary functions	-	-

Schematic representation in LAD/FBD

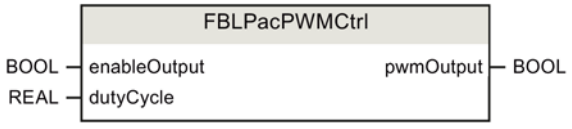


Figure 4-94 FBLPacPWMCtrl - schematic block view in LAD/FBD

Input and output parameters

Element	P type [1]	Data type	Initial value	Meaning
enableOutput	IN	BOOL	FALSE	Enable the PMW control in accordance with the supplied duty cycle
dutyCycle	IN	REAL	0.0	Duty cycle in %
pwmOutput	OUT	BOOL	FALSE	PWM output

[1] Parameter types: IN = input parameter, OUT = output parameter, IN_OUT = in-out parameter

4.9.3.3 Integration of the module

Technology objects, basic parameterization and block call

Table 4- 221 TO type

Technology object	
TO type	Not relevant
Block call	TimerTask

The **FBLPacPWMCtrl** block must be called in a timer task.
To prevent beat states, ensure that the call cycle of the task is not identical with the time period of the power frequency (e.g. 50 Hz -> time period = 20 ms -> call cycle, e.g. 27 ms).

Further blocks and libraries

The **FBLPacPWMCtrl** block does not use any further blocks.

Error messages

The block does not generate any own messages.

Note

Tip

The **pPIDTempControl** program unit is available in the LPac project to illustrate the use of the **FBLPacPIDTempCtrl** and **FBLPacPWMCtrl** blocks.

This unit contains the programs for the required block calls in the TimerInterruptTasks.

Alarm and error messages 5

5.1 Alarm and error messages

Response to errors

If an error is determined while processing the function blocks, the *error* bit is set and an error code issued in the range 16#8000 – 16#8200 at the *errorID* output. If the controlled axis is in motion, it is not brought to a standstill.

Behavior for error messages from SIMOTION system functions

If SIMOTION system functions called internally in the LPac blocks issue an error, the *error* bit is set and an appropriate error code issued in the range 16#8000 – 16#8200 in the low word at the *errorID* output. The return value of the SIMOTION system function is also issued in hex format in the high word of *errorID*. The significance of the return values is explained either in the SIMOTION online help or decoded with the Systemfunktionen_TP_Referenzliste SIMOTION Manual.

The return values are described for the SIMOTION _move system function.

For the system functions for the temperature channel technology object, the return values are described by the *_setTempControllerOperatingMode* function.

The return values specified in the descriptions have the DINT format. The output to the LPac blocks, however, is made in the DWORD format.

Behavior for error messages of internally called blocks

If in the LPac blocks, blocks called internally, such as the **FBLMCBasicAxis** or **FBLCamHdlCreateCam** axis function block, issue an error, the *error* bit is also set and an appropriate error code issued in the range 16#8000 – 16#8200 (the error code of the called block) is output at the *errorID* output. If the internally-called **FBLMCBasicAxis axis function block reports an error**, the LPAC_ERR_AXIS_FUNCTION_BLOC message is issued to errorID. The error code is also output in the *diagnosticData* output structure*. The b32ErrorCode structure element contains the Errorcode issued by the axis function block (the associated description is contained in the SIMOTION Axis Function Block Application Manual) as well as any additional error information, e.g. the return code from internally used PLCopen functions in the b32AdditionalValue5 structure element.*

Behavior for TO alarms

The LPac blocks forward TO alarms caused by the internally-called FBLMCBasicAxis externally at the toAlarmActive output. Whereby, only TO alarms with *errorReaction* <> NONE are indicated. Only these TO alarms can also be reset with the ackError input. All other TO alarms must be reset externally, e.g. by using the standard "message handling". A detailed description of the behavior for TO alarms is available in the SIMOTION Axis Function Block Application Manual.

Table 5- 1 sLMCBasicDiagnosticDataType diagnostic structure

Element	Data type	Meaning
u8ErrorClass	USINT	Error class
u8ToType	USINT	Not used
i16ToNumber	INT	Not used
b32ErrorCode	DWORD	Error code
i32AdditionalValue1	DINT	Additional information for brake control, following error monitoring and *redefinePosition*
i32AdditionalValue2	DINT	Not used
i32AdditionalValue3	DINT	Not used
i32AdditionalValue4	DINT	Not used
b32AdditionalValue5	DWORD	Return code of the PLCopen function (if error has been triggered by the PLCopen function)
r32AdditionalValue6	REAL	Not used

Acknowledgment of alarms and errors

All error messages are acknowledged and deleted with a rising edge at the *ackError* block input.

Error messages

The error codes generated by the **LPac** blocks are assigned to the appropriate constants in the **cPublic** unit.

The error messages possible for each block are provided as table at the end of each block description.

Application description

6

6.1 LPac project

To illustrate the application and for further simplification and reduction of the development time, the user of the LPac library is provided with a SIMOTION complete project.

The following components are already implemented and created as example there:

- Hardware configuration for a SIMOTION controller
- Various technology objects (axes, measuring inputs, cams, etc.)
- Program examples for the LPac equipment modules that show the following:
  - Parameter initialization in a separate startup program
  - Creation of a function block instance for the module and the associated block call
  - The calling program for the module, i.e. supply of the module input signals, module output signals and parameters
- Global data structures that contain the control and status signals as well as the parameters of the modules
- Program examples for the mode management implemented in accordance with the OMAC V3 model
- Program examples for the deployed standard message handling
- Program examples for the deployed startup check

- Integration of the **LPac** library and all other required libraries
- HMI operating screens for WINCCflexible

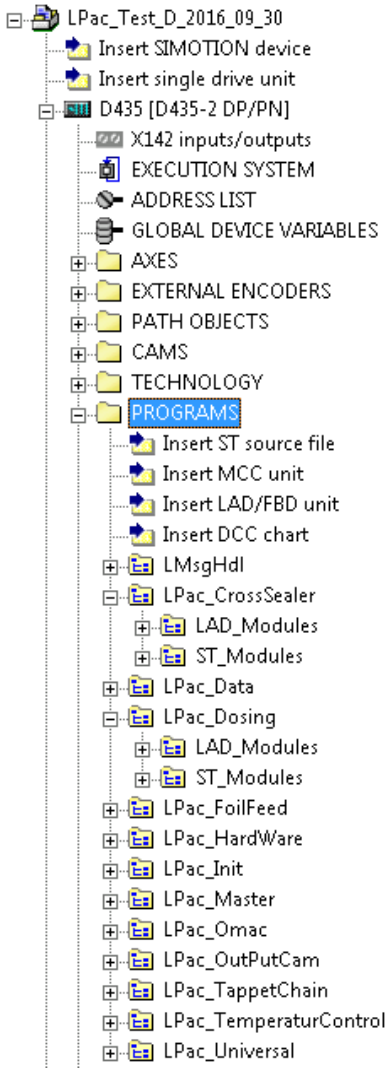


Figure 6-1 LPac example project

Program examples for every equipment module are available in the program container.

Each program example contains only the call and the supply and disposal of the module inputs/outputs for a module.

The program examples are available in the ST (Structured Text) and in the LAD (Ladder logic) programming languages.

A separate startup program (ending STT) and a cyclic program (endings BGT = BackgroundTask, MT = Motiontask or TT = Timertask) are created for each equipment module.

In this way, it is easier to identify in which execution task the respective programs are to be called and the program examples are modular and easy to modify, and can be combined into a complete application.

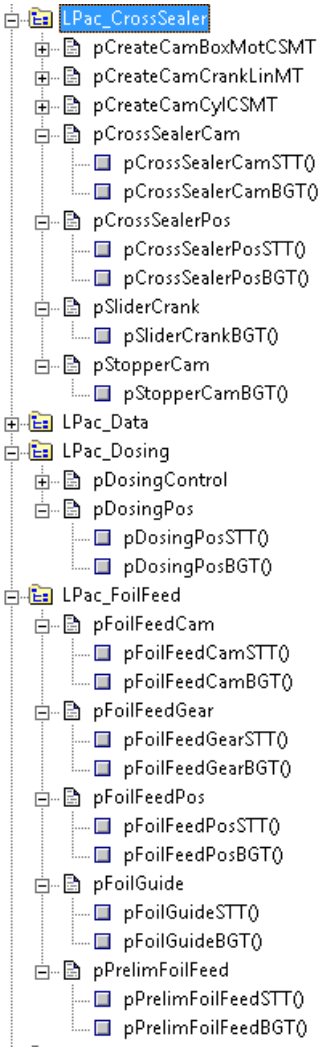


Figure 6-2 LPac program units with easy to identify assignment to the execution levels

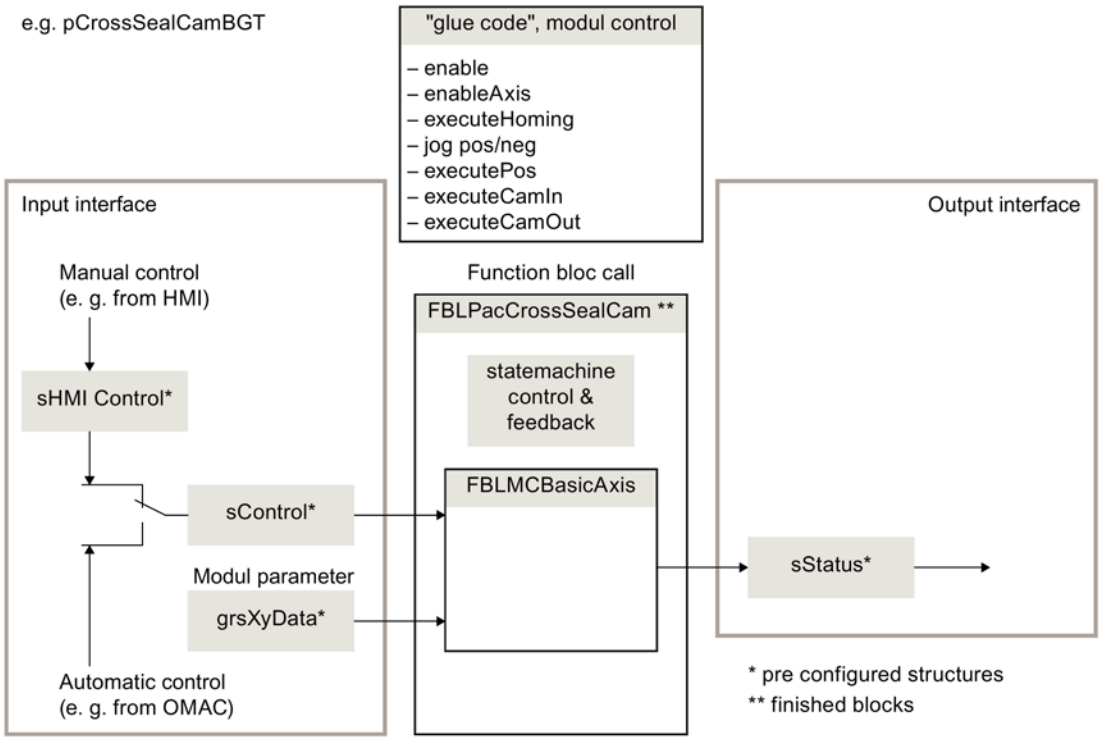


Figure 6-3 Basic design of an Lpac program example module

To easily create a complete application, the associated required program units can be integrated in the execution system of the SIMOTION controller.

To improve clarity of the complete application, any superfluous program units can be deleted from the program container later.

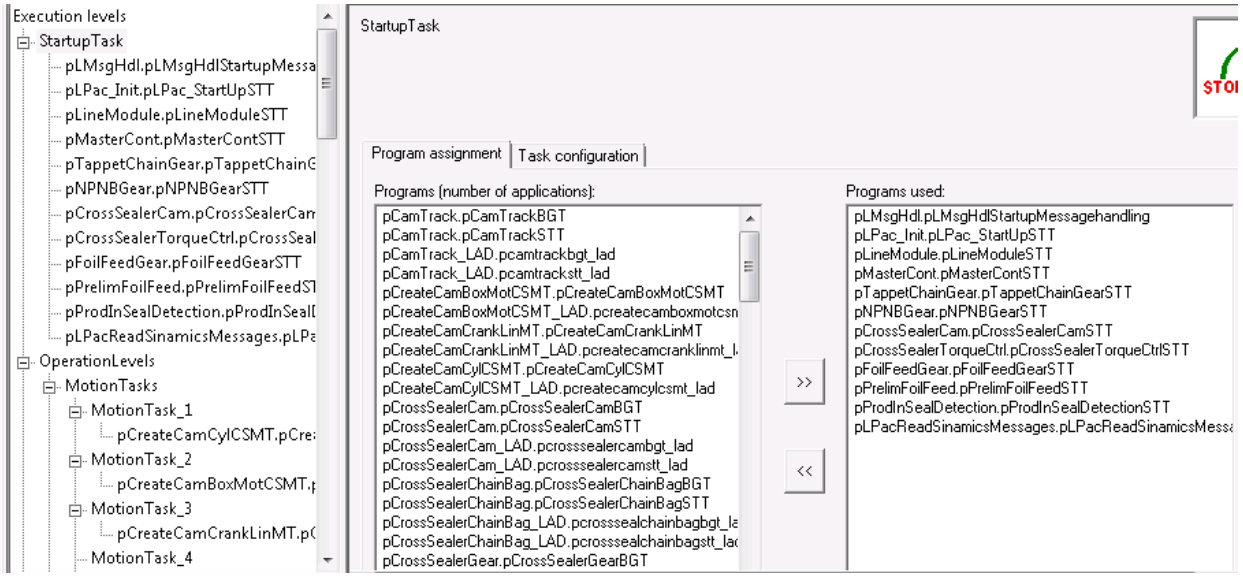


Figure 6-4 SIMOTION execution system with the required program calls of a sample horizontal tubular bagging machine

The program examples and program units are named in accordance with the LPac blocks used. Consequently, based on the program and unit names, it is very easy to recognize which functionality or which machine module is being processed in the associated program.

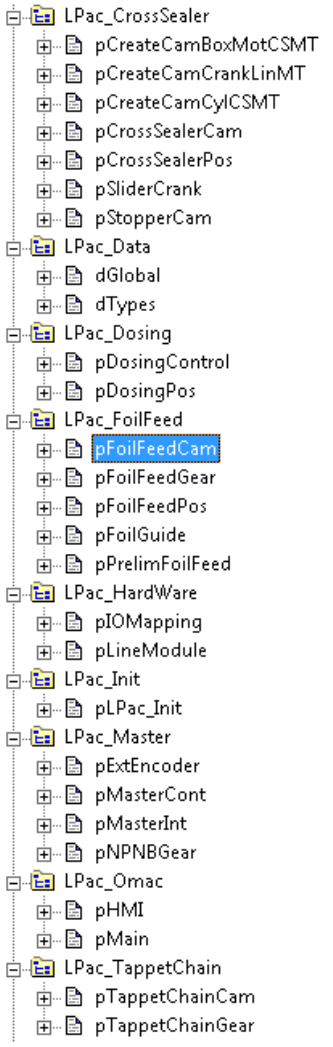


Figure 6-5 Program examples overview

6.2 Sample application of a horizontal tubular bagging machine with LPac

The application of a horizontal tubular bagging machine is described as example below. Note that not all connections and interconnections are shown. Other required general functions, such as operating mode management, message handling, I/O interconnection, are not shown because they can vary greatly depending on the application case.

Description of the deployed technology objects / axes / synchronous operations

1. Machine leading axis controlled with FBLPacMasterCont

Task:	The axis serves as central master value encoder for the machine. The axis runs either in "standalone" operation with the specified product speed or in coupled operation. e.g. with an external encoder as master value source for coupling to an upstream machine.
Technology:	Position or following axis
Axis type:	Virtual linear axis
Modulo length:	1000
Leading axis:	In coupled operation, e.g. an external encoder or also a superimposed SIMOTION leading axis of another machine (external synchronous operation)

2. No Product No Bag axis controlled with FBLPacTappetChainGear

Task:	Coupled axis for the common synchronous stopping/starting of the foil feed and the cross-sealer to implement the "No Product No Bag" function. Whereby, a sensor (e.g. light scanner) normally mounted on the pocket-type conveyer senses whether a product is present in each conveyor pocket. If a product is missing, a shift register that maps the products to the cross-sealer, outputs a signal for stopping the cross-sealer and the foil feed. Both axes then stop within half a product cycle by desynchronizing the gearing. If a product is missing, the two axes start again to traverse synchronously to the leading axis in the next cycle, i.e. the two axes "suspend" for one product cycle. This allows the production to be inhibited for so-called "empty bags".
Technology:	Following axis
Axis type:	Virtual linear axis
Modulo length:	1000
Leading axis:	Master axis
Coupling:	Gearing with gear ratio 1:1

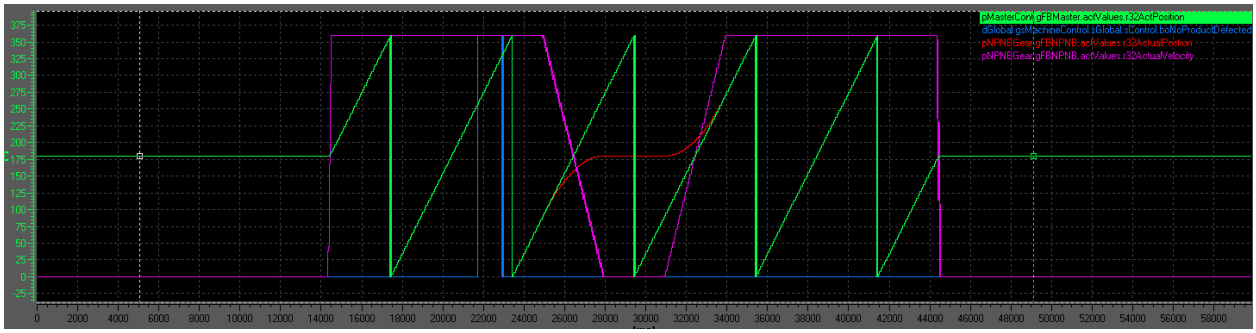


Figure 6-6 Production cycle with stop of the "No product, no bag" axis for a product

The following are shown:

- Green: Position change of the master axis for 360 mm bag length and a speed of 60 mm/s.
- Red: Position change of the NPNB axis. Except for the suspension cycle, the position change is identical as for the master axis.
- Violet: Speed change of the NPNB axis.
- Blue: "NoProduct" signal, for example, derived from a light scanner on the supply chain (tappet chain) and via a shift register triggers the desynchronization of the NPNB axis at the correct time. The associated "Product" signal that represents a detected product in the pocket-type conveyer is not shown here.

It shows for the correct triggering of the synchronization or desynchronization of the NPNB axis that it is held in position for at least one product cycle.

Note

Note that for the deployment of "No Product No Bag" functionality, the coupled drives must also be able to stop or accelerate from or to the maximum production speed within half a product cycle.

3. Pocket-type conveyer / feed chain controlled via FBLPacTappetChainGear

Task:	Insert the product in the foil tube
Technology:	Gear synchronous axis
Axis type:	Real linear axis
Modulo length:	Conveyor pocket length
Leading axis:	Master axis

Coupling:	Gearing with with gear ratio, master axis conveyor pocket length / modulo length
Functions:	Whereby the feed chain should be brought to a standstill and restarted defined from running operation, e.g. by pressing an operator button. The "Chain stop" function is implemented on the gearing by desynchronization/resynchronization. This can be implemented, e.g. with the **LPacTappetChainGear** block by the appropriate parameterization and control of the *executeGearOut* or *executeGearIn* inputs. When the LPac **FBLPacTappetChainCam** block is used, it can be implemented with the **FBLPacStopperCam** suspension block by loading the stop, standstill and start cams. Using the "executeChangeOffset" input signal, the phase position of the feed chain can be moved incrementally with regard to the leading axis based on the selected parameterization and so the feed time of the product in the foil tube adjusted. The required parameter settings have already been preset in the LPac program modules. A set offset is saved retentively and automatically restored at the next synchronization (even after a CPU stop and power failure).

4. Foil feed controlled with FBLPacFoilFeedGear

Task:	Foil tube pretension
Technology:	Gear synchronous axis
Axis type:	Real linear axis
Modulo length:	Leadscrew pitch
Leading axis:	"No Product No Bag" axis or master axis
Coupling:	Gearing with bag length gear ratio / master axis modulo length; the gear ratio is calculated automatically by the master block ("ratioBaglengthModulo" variable) to permit "on the fly" changes to the bag length. For this purpose, the master block controls not only the gear ratio, but also calculates a new cyclic cam and its loading at the product cycle end.

Note

Note that "on the fly" bag length changes are permitted only with small changes of the bag length, i.e. only a few mm, because otherwise sudden axis reactions can occur.

Functions: Print mark correction: The LPac project and the **FBLPacFoilFeedGear** and **FBLPacFoilFeedCam** blocks contain a print mark correction function. This function operates in accordance with the following principle: the required measuring input for acquiring the print mark position is assigned to the master axis. Consequently, the print mark position on the master axis is measured. The measured actual position is compared with the print mark setpoint position. The print mark setpoint position must be specified in the range 0…bag length. The block converts it to modulo length 1000 of the master. The correction motion is performed as superimposed positioning on the foil feed axis. For this purpose, the measured and calculated difference positions are converted to the bag length and used as setpoint position for the relatively performed superimposed positioning motion.

Slip compensation: The LPac project and the **FBLPacFoilFeedGear** and **FBLPacFoilFeedCam** blocks contain a slip compensation. For the function description, please refer to the description of the FBLPacSlipCompensation (Page 165) block.

Note

The proper functioning of the print mark correction requires a constant foil tension. Otherwise the exact actual print mark position using a light scanner is not possible.

5. Rotating cross-sealer controlled with FBLPacCrossSealCam

Task:	Rotating cross-sealer
Technology:	Cam synchronous axis
Axis type:	Real linear or rotary axis
Modulo length:	360° (between two cutter bars)
Leading axis:	"No Product No Bag" axis or master axis
Coupling:	Camming with cross-sealer modulo length cams / master axis modulo length. The required cams can be calculated for various mechanical versions of the cross-sealer by LPac blocks at runtime.
Functions:	Bag length change "on the fly". The **FBLPacStopperCam** block can be used at runtime to make an "on the fly" small change to the bag length. The calculation of the required cams and their loading is controlled by the master block together with changing the gear ratio for the foil feed.

Note

Note that "on the fly" bag length changes are permitted only with small changes of the bag length, i.e. only a few mm, because otherwise sudden axis reactions can occur.

No Gap No Seal: For this purpose, the cross-sealer is stopped in the open position if no gap was created between two successive products. This function prevents the product from being "crushed" by the cross-sealer if no gap was created between two successive products. The **FBLPacStopperCam** block can be used to implement the function. With an appropriate

control of the two "executeStop" and "executeStart" signal inputs via a shift register at the correct time (i.e. until shortly before the cycle end), a switch is made from the cyclic cam to the stop cam and in the subsequent processing a switch made to the standstill or start cam and so a suspend function generated for one or several product cycles.

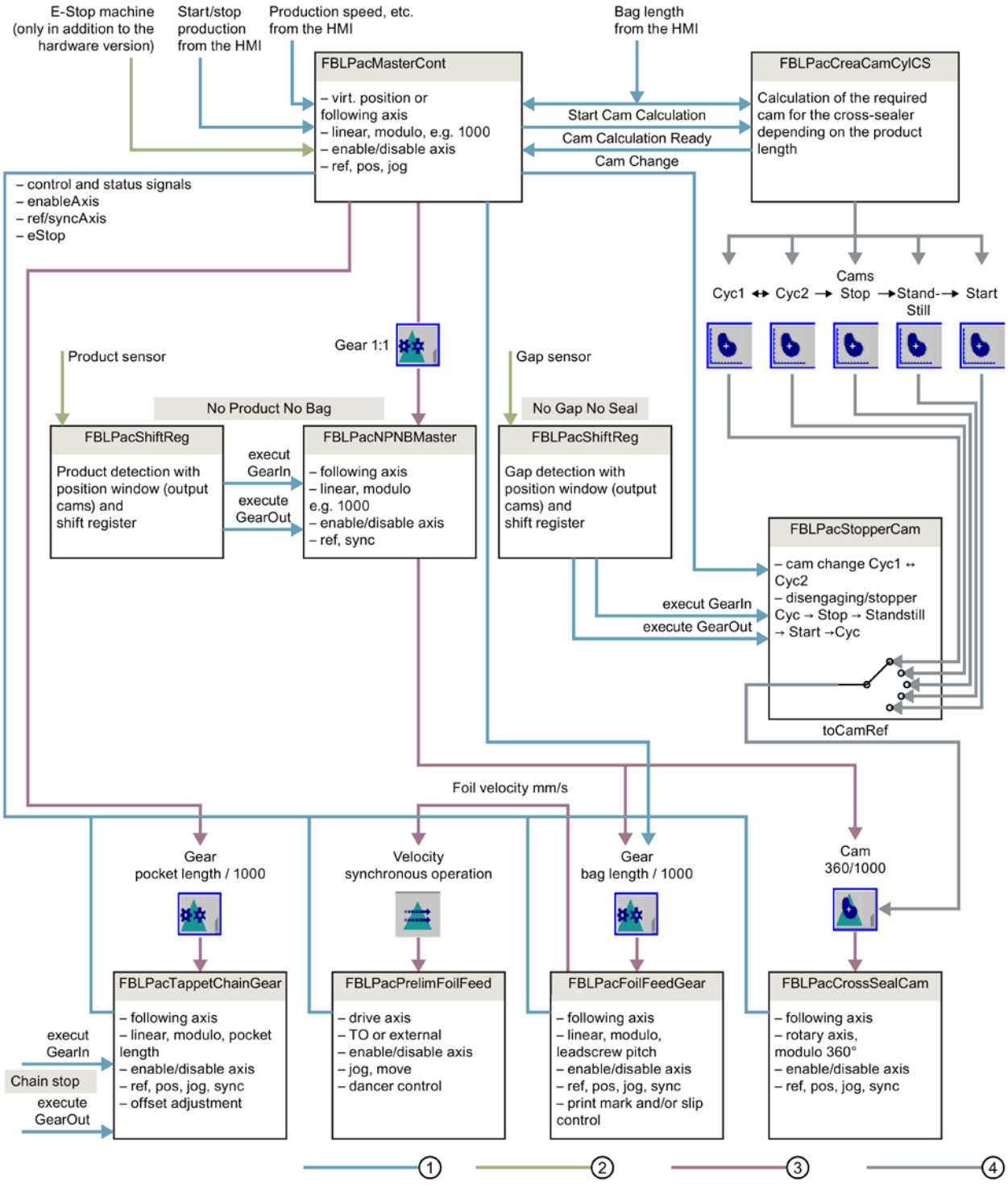
6. Preliminary foil feed controlled with FBLPacPrelimFoilFeed

Task:	Foil unwinder
Technology:	Drive axis
Axis type:	Real rotary axis
Modulo length:	Not relevant
Functions:	Tension setting

Note

An additional dancer control should be deployed for setting the tension. It is integrated in the LPac **FBLPacPrelimFoilFeed** block. If only a mechanical dancer rocker is present, a velocity difference should be set between the foil feed (leading velocity) and the preliminary foil feed to establish a defined foil tension. The velocity difference can be specified at the "velocityOffset" input of the **FBLPacPreimFoilFeed** block in % (generally a negative offset is required). A correct setting of the mechanical dancer rocker must also be ensured.

The following figure shows a sample application of a horizontal tubular bagging machine with LPac.



① Control/status signals
② I/O signals
③ Master value v, s, a
④ TO reference

6.3 Application sample of a vertical continuous tubular bagging machine with LPac

The application of a vertical continuous tubular bagging machine is described as example below. Note that not all connections and interconnections are shown. Other required general functions, such as operating mode management, message handling, I/O interconnection, are not shown because they can vary greatly depending on the application case.

Description of the deployed technology objects / axes / synchronous operations

1. Machine leading axis controlled with FBLPacMasterCont

Task:	The axis serves as central master value encoder for the machine. The axis runs with the specified product speed.
Technology:	Position axis
Axis type:	Virtual linear axis
Modulo length:	1000
Leading axis:	Not relevant

2. No Product No Bag axis controlled with FBLPacTappetChainGear

Task:	Coupled axis for the common synchronous stopping/starting of the foil feed and the cross-sealer to implement the "No Product No Bag" function. Whereby, the dosing control specifies whether or not the dosing weigher is ready for product discharge. If no product is discharged, a signal for stopping the cross-sealer and the foil feed is issued. Both axes then stop within half a product cycle by desynchronizing the gearing. If a product is missing, the two axes start again to traverse synchronously to the leading axis in the next cycle, i.e. the two axes "suspend" for one product cycle. This allows the production to be inhibited for so-called "empty bags".
Technology:	Following axis
Axis type:	Virtual linear axis
Modulo length:	1000
Leading axis:	Master axis
Coupling:	Gearing with gear ratio 1:1

Note

Note that for the deployment of "No Product No Bag" functionality, the coupled drives must also be able to stop or accelerate from or to the maximum production speed within half a product cycle.

3. Foil feed controlled with FBLPacFoilFeedGear

Task:	Foil tube pretension
Technology:	Gear synchronous axis
Axis type:	Real linear axis
Modulo length:	Leadscrew pitch
Leading axis:	"No Product No Bag" axis or master axis
Coupling:	Gearing with bag length gear ratio / master axis modulo length; the gear ratio is calculated automatically by the master block ("ratioBaglengthModulo" variable) to permit "on the fly" changes to the bag length. For this purpose, the master block controls not only the gear ratio, but also calculates a new cyclic cam and its loading at the product cycle end.

Note

Note that "on the fly" bag length changes are permitted only with small changes of the bag length, i.e. only a few mm, because otherwise sudden axis reactions can occur.

Functions:	Print mark correction: The LPac project and the **FBLPacFoilFeedGear** and **FBLPacFoilFeedCam** blocks contain a print mark correction function. This function operates in accordance with the following principle: the required measuring input for acquiring the print mark position is assigned to the master axis. Consequently, the print mark position on the master axis is measured. The measured actual position is compared with the print mark setpoint position. The print mark setpoint position must be specified in the range 0…bag length. The block converts it to modulo length 1000 of the master. The correction motion is performed as superimposed positioning on the foil feed axis. For this purpose, the measured and calculated difference positions are converted to the bag length and used as setpoint position for the relatively performed superimposed positioning motion. Slip compensation: The LPac project and the **FBLPacFoilFeedGear** and **FBLPacFoilFeedCam** blocks contain a slip compensation. For the function description, please refer to the description of the FBLPacSlipCompensation (Page 165) block.

Note

The proper functioning of the print mark correction requires a constant foil tension. Otherwise the exact actual print mark position using a light scanner is not possible.

4. Rotating cross-sealer controlled with FBLPacCrossSealCam

Task:	Rotating or coupled (box motion) cross-sealer
Technology:	Cam synchronous axis
Axis type:	Real rotary axis or linear axis (for box motion)

Modulo length:	360° (between two cutter bars), not relevant for box motion
Leading axis:	"No Product No Bag" axis or master axis
Coupling:	Camming with cross-sealer modulo length cams / master axis modulo length. The required cams can be calculated for various mechanical versions of the cross-sealer by LPac blocks at runtime.
Functions:	"On the fly" bag length change The **FBLPacStopperCam** block can be used at runtime to make an "on the fly" small change to the bag length. The calculation of the required cams and their loading is controlled by the master block together with changing the gear ratio for the foil feed.

Note

Note that "on the fly" bag length changes are permitted only with small changes of the bag length, i.e. only a few mm, because otherwise sudden axis reactions can occur.

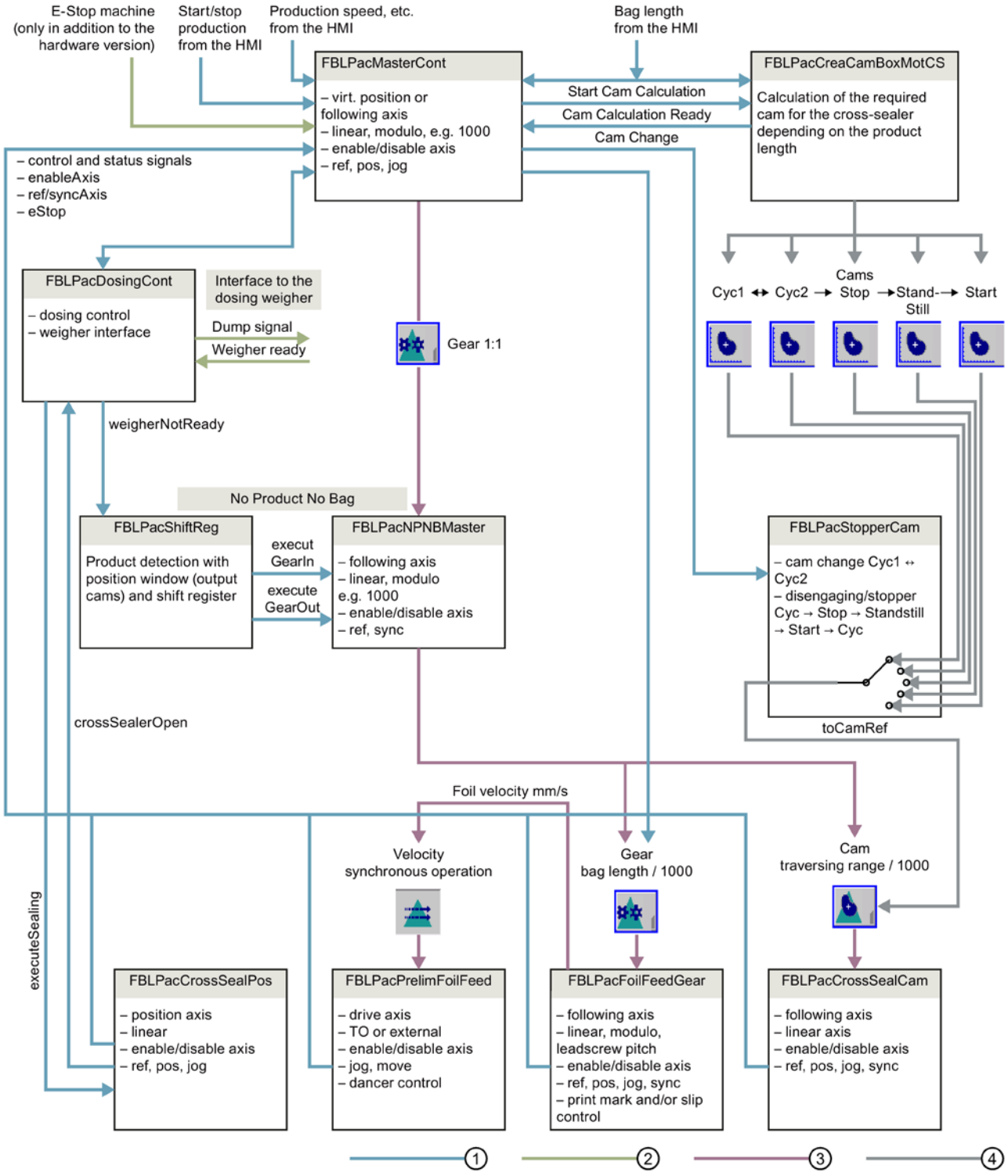
5. Preliminary foil feed controlled with FBLPacPrelimFoilFeed

Task:	Foil unwinder
Technology:	Drive axis
Axis type:	Real rotary axis
Modulo length:	Not relevant
Functions:	Tension setting

Note

An additional dancer control should be used for setting the tension. It is integrated in the LPac **FBLPacPrelimFoilFeed** block. If only a mechanical dancer rocker is present, a velocity difference should be set between the foil feed (leading velocity) and the preliminary foil feed to establish a defined foil tension. The velocity difference can be specified at the *velocityOffset* input of the **FBLPacPreimFoilFeed** block in % (generally a negative offset is required). A correct setting of the mechanical dancer rocker must also be ensured.

The following figure shows a sample application of a vertical continuous tubular bagging machine with LPac.



① Control/status signals
② I/O signals
③ Master value v, s, a
④ TO reference

6.4 Sample application of a vertical intermittent tubular bagging machine with LPac

The application of a vertical intermittent tubular bagging machine is described as example below. Note that not all connections and interconnections are shown. Other required general functions, such as operating mode management, message handling, I/O interconnection, are not shown because they can vary greatly depending on the application case.

Description of the deployed technology objects / axes / synchronous operations

1. Machine master controlled with FBLPacMasterInt

Task:	The master block triggers the positioning motions of the foil feed and the cross-sealer with a time-controlled execution sequence and binary output signals. It also ensures the coordination with the external dosing weigher. This is also coordinated via a binary interface.
Technology:	Time-based execution sequence
Axis type:	Not relevant, no axis TO used
Modulo length:	Not relevant

2. Foil feed controlled with FBLPacFoilFeedPos

Task:	Foil tube pretension
Technology:	Position axis
Axis type:	Real linear axis
Modulo length:	Not relevant
Leading axis:	Triggering via the master block
Coupling:	Not relevant
Functions:	Print mark correction: The LPac project and the **FBLPacFoilFeedPos** block contains a print mark correction function. This function operates in accordance with the following principle: the required measuring input for acquiring the print mark position is assigned to the foil feed axis. The measured actual position is compared with the print mark setpoint position. The print mark setpoint position must be specified in the range 0…bag length. The correction motion is performed as superimposed positioning on the foil feed axis. For this purpose, the measured and calculated difference positions are used as setpoint position for the relatively performed superimposed positioning motion.

Note

The proper functioning of the print mark correction requires a constant foil tension. Otherwise the exact actual print mark position using a light scanner is not possible.

3. Cross-sealer controlled with FBLPacCrossSealPos

Task:	Cross-sealer with fixed sealing jaws
Technology:	Position axis
Axis type:	Real linear axis
Modulo length:	Not relevant
Leading axis:	Not relevant, control by binary-triggered master block
Coupling:	Not relevant.

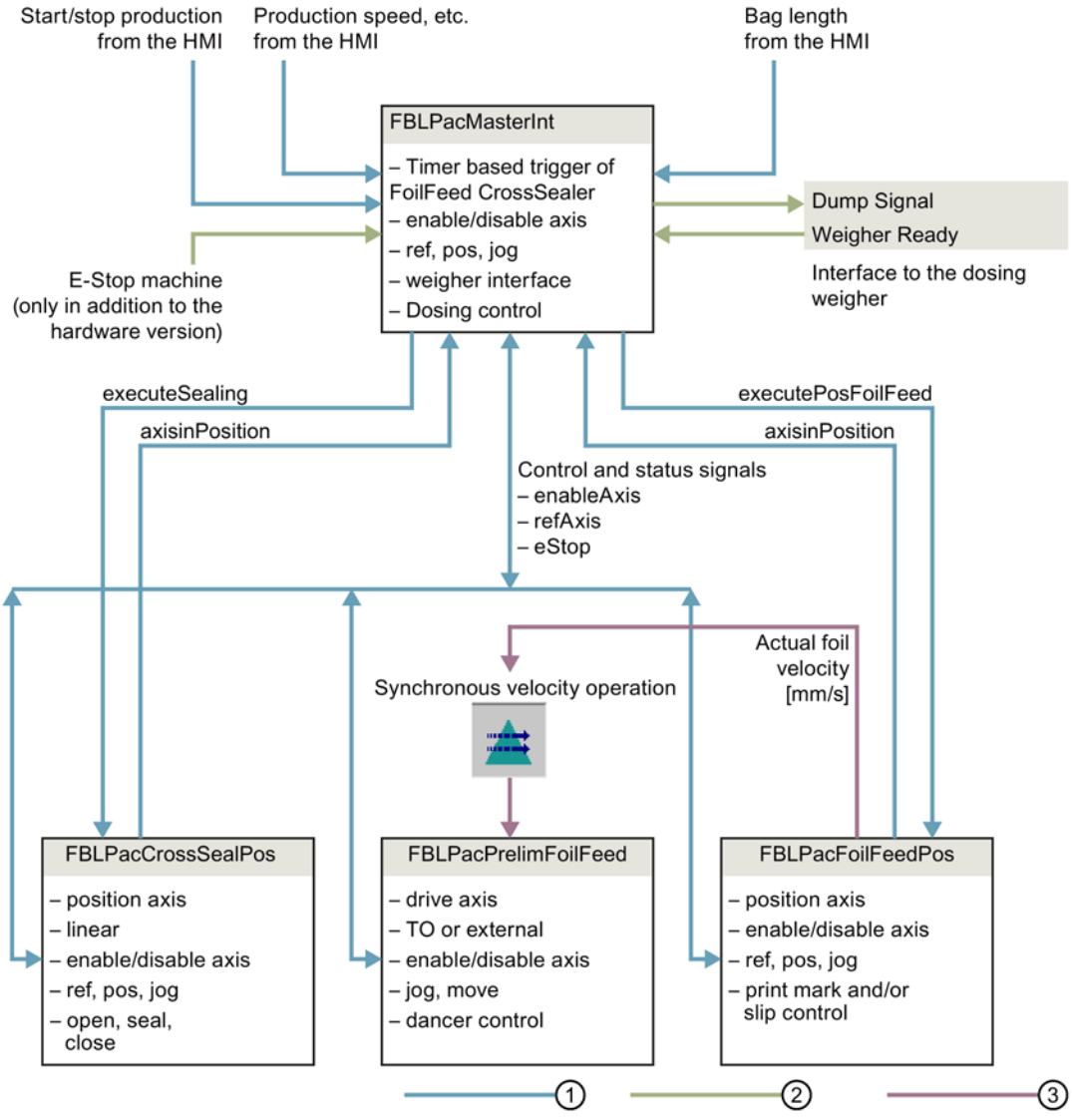
4. Preliminary foil feed controlled with FBLPacPrelimFoilFeed

Task:	Foil unwinder
Technology:	Drive axis axis in synchronous velocity operation to the foil feed
Axis type:	Real rotary axis
Modulo length:	Not relevant
Leading axis:	Foil feed coupled via the actual velocity
Functions:	Tension setting

Note

An additional dancer control should be used for setting the tension. It is integrated in the LPac **FBLPacPrelimFoilFeed** block. If only a mechanical dancer rocker is present, a velocity difference should be set between the foil feed (leading velocity) and the preliminary foil feed to establish a defined foil tension. The velocity difference can be specified at the *velocityOffset* input of the **FBLPacPreimFoilFeed** block in % (generally a negative offset is required). A correct setting of the mechanical dancer rocker must also be ensured.

The following figure shows a sample application of a vertical intermittent tubular bagging machine with LPac.



① Control/status signals
② I/O signals
③ Master value v, s, a

6.5 Application sample for possible temperature controls with LPac blocks

The two possible basic structures for temperature control with LPac blocks are shown schematically below. Note that not all connections and interconnections are shown. Other required general functions, such as operating mode management, message handling, I/O interconnection, are not shown because they can vary greatly depending on the application case.

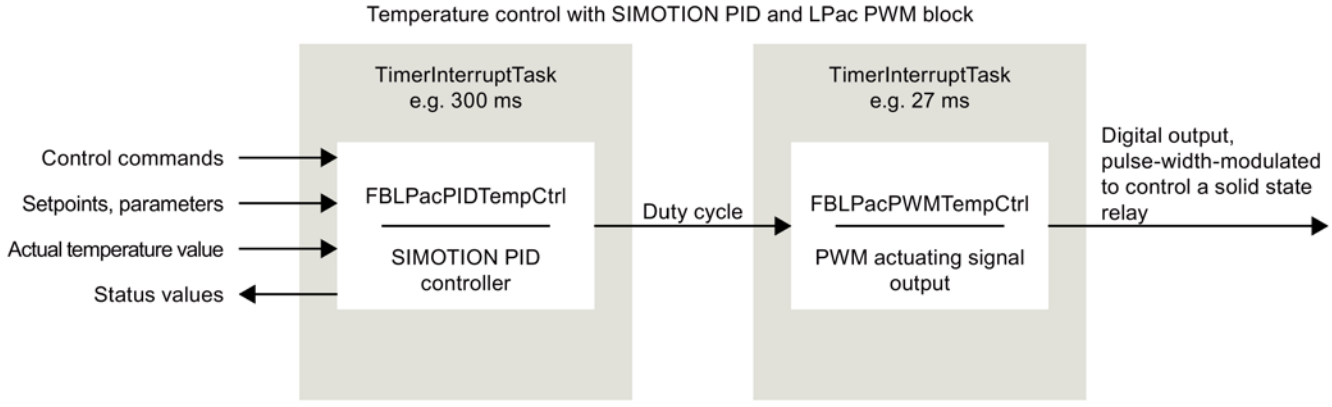


Figure 6-7 Schematic diagram of a single-channel temperature control implemented with the two LPac FBLPacPIDCtrl and FBLPacPWMCtrl blocks

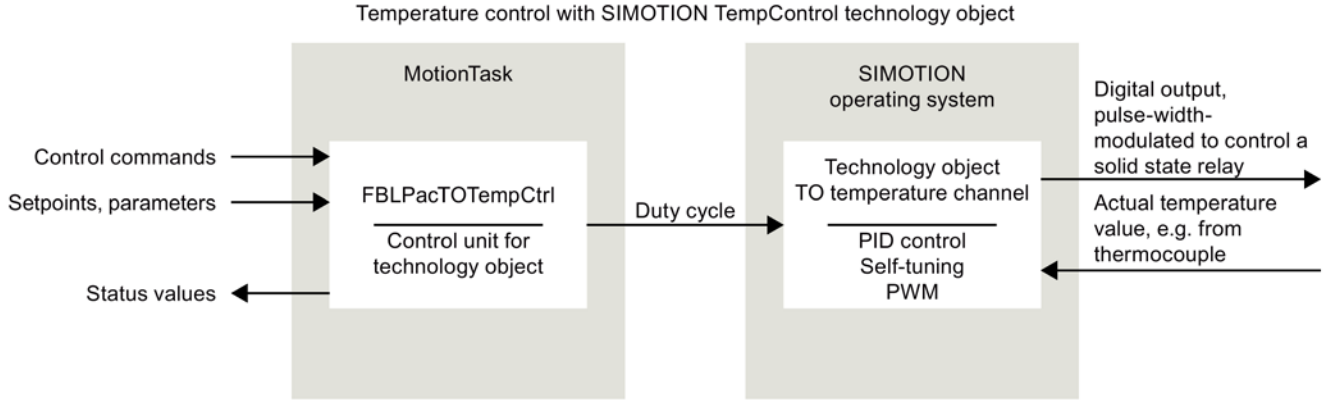


Figure 6-8 Schematic diagram of a single-channel temperature control implemented with the SIMOTION temperature channel TO and the FBLPacToTempCtrl contactor block

Contact

A

A.1 Contacts

Siemens AG
Digital Factory
Factory Automation
Production Machines
DF FA PMA APC
Frauenauracher Strasse 80
D-91056 Erlangen, Germany
Fax.: +49 9131 98 1297
mailto:tech.team.motioncontrol@siemens.com

A.2 Internet addresses

Additional information on various topics is provided on the following Internet pages.

SIMOTION

SIMOTION LPac basic equipment module library	https://support.industry.siemens.com/cs/ww/en/view/109475570
Axis function block	https://support.industry.siemens.com/cs/ww/en/view/48816191
SIMOTION easyProject ProjectGenerator	https://support.industry.siemens.com/cs/ww/en/view/51339107
SIMOTION	www.siemens.com/simotion
SINAMICS	www.siemens.com/sinamics
Motion control / Application center	www.siemens.com/motioncontrol/apc
Packaging	www.siemens.com/packaging