

PLC Designer



L_EPMP_PackMLManager -----

Software-Handbuch

EN



13566602

Contents

1	About this documentation	3
1.1	Document history	4
1.2	Conventions used	5
1.3	Notes used	6
2	Safety instructions	7
3	Prerequisites	8
3.1	System requirements	8
3.2	Establishing communication with the controller	8
4	L_EPMP_PackMLManager library	10
4.1	OMAC PackML	10
4.1.1	PackML machine operating modes (modes)	11
4.1.2	PackML status model (state machine)	12
4.2	Libraries with enumeration types for PackML	15
4.2.1	L_EPMP_PML_States	15
4.2.2	L_EPMP_PML_Commands	15
4.3	Blocks for PackML	16
4.3.1	L_EPMP_PackMLManager	17
4.3.2	L_EPMP_PML_Config	20
4.3.3	L_EPMP_PML_ModeConfig	22
4.3.4	L_EPMP_PML_Line-Interface_TR880002_Minimum	23
4.4	Configuration	24
4.4.1	Mode management	24
4.4.1.1	Specifying modes available	25
4.4.1.2	Specifying the initial mode after system start-up	26
4.4.1.3	Allowing and inhibiting mode change-overs	27
4.4.1.4	Executing a mode change-over	28
4.4.2	State management	29
4.4.2.1	Adapting the state model	30
4.4.2.2	Specifying the initial state after system start-up	31
4.4.2.3	Executing a state change-over	32
5	Line interface and pack tags	33
5.1	Pack tag scope TR880002_Minimum	34
5.2	Use of the line interface block	35
5.2.1	GVL for the line interface	35
5.2.2	Linkage with the line interface block	35
5.2.3	Connection of the GVL instances	35
5.2.4	Limited control scope via the line interface	36
5.3	InternalControl and visualisation frame	37
	Index	39
	Your opinion is important to us	41

1 About this documentation

This documentation describes the "L_EPMP_PackMLManager" Lenze application library.

To understand the basis for handling this library, please refer to the "Lenze FAST" section in the PLC Designer online help.



Stop!

In this manual, the software solution of a task is described. The transferability of the solution described to the respective application **must** be checked by the user. If necessary, the solution must be adapted.

Physical aspects such as the drive dimensioning are not covered by this manual.



Note!

This documentation is a supplement to the »PLC Designer« online help.



Tip!

Information and tools regarding the Lenze products can be found in the download area at:

<http://www.Lenze.com>

This manual is part of the "Controller-based Automation" manual collection. The manual collection consists of the documents:

Documentation type	Subject
System manuals	System overview/example topologies • Controller-based Automation • Visualisation
Communication manuals	Bus systems • Controller-based Automation EtherCAT® • Controller-based Automation CANopen® • Controller-based Automation PROFIBUS® • Controller-based Automation PROFINET®
Online helps/ Software manuals	Lenze Engineering tools • »PLC Designer«: programming • »Engineer«: configuring the inverter • »VisiWinNET® Smart«: visualisation • »Backup & Restore«: backing up/restoring data

More technical documentation with regard to Lenze components

Further information on Lenze products which can be used in connection with Controller-based Automation can be found in the following documentation:

Mounting & wiring
☐ Mounting instructions • Controllers • Communication cards (MC-xxx) • I/O system 1000 (EPM-Sxxx) • Inverter • Communication modules
Sample applications/Using application templates
☐ Online help/software manuals • i700 application sample • Application Samples • ApplicationTemplate • ApplicationTemplate PackML
Parameterisation, configuration, commissioning
☐ Online help/software manuals • Controllers • i700 servo inverter • Servo Drive 9400 HighLine/PLC/regenerative power supply module • Inverter Drive 8400 StateLine/HighLine/TopLine • I/O system 1000 (EPM-Sxxx)
☐ Online help/communication manuals • Bus systems • Communication modules

Icons

- ☐ Printed documentation
- ☐ Online help in the Lenze Engineering tool/
software manuals and communication manuals are provided as PDF files on the Internet.

Target group

This documentation addresses to all persons who plan, commission, and program a Lenze automation system on the basis of the Lenze "ApplicationTemplate PackML" as part of the "Controller-based Automation".

Screenshots/application examples

All screenshots in this documentation are application examples. Depending on the firmware version of the Lenze devices and the software version of the engineering tools installed (»PLC Designer«), the screenshots in this documentation may deviate from the screen representation.

Information regarding the validity

The information in this documentation is valid for the following Lenze software:

Software	From software version
»PLC Designer«	3.9

Document history

Version			Description
1.0	02/2019		First edition

1 About this documentation

1.2 Conventions used

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

Type of information	Highlighting	Examples/notes
Spelling of numbers		
Decimal separator	Point	The decimal point is always used. For example: 1234.56
Text		
Version information	Blue text colour	All information that only applies for or from a certain software version of the inverter are marked accordingly in this documentation. Example: This function extension is available from software version V3.0 onwards!
Program name	» «	»PLC Designer«...
Window	<i>italics</i>	The <i>Message window...</i> / The dialog box <i>Options...</i>
Variable names		By setting <i>bEnable</i> to TRUE...
Control element	Bold	The OK button... / The Copy command... / The Properties tab... / The Name input field...
Sequence of menu commands		If several commands must be used in sequence to carry out a function, the individual commands are separated by an arrow: Select File→Open to...
Shortcut	< bold >	Use < F1 > to open the online help.
		If a shortcut is required for a command to be executed, a "+" has been put between the key identifiers: With < Shift >+< ESC > ...
Hyperlink	underlined	Reference to further information: Hyperlink to further information.
Icons		
Page reference	 5)	Reference to further information: Page number in PDF file.
Step-by-step instructions		Step-by-step instructions are indicated by a pictograph.

1.3

Notes used

The following signal words and symbols are used in this documentation to indicate dangers and important information:

Safety instructions

Structure of the safety instructions:

**Pictograph and signal word!**

(characterise the type and severity of danger)

Note

(describes the danger and explains how to avoid it.)

Pictograph	Signal word	Meaning
	Danger!	Danger of personal injuries through electrical voltage Reference to an imminent danger that may result in death or serious personal injury if the corresponding measures are not taken.
	Danger!	Danger of personal injury through a general source of danger Reference to an imminent danger that may result in death or serious personal injury if the corresponding measures are not taken.
	Stop!	Danger of property damage Reference to a possible danger that may result in property damage if the corresponding measures are not taken.

Application notes

Pictograph	Signal word	Meaning
	Note!	Important note to ensure trouble-free operation
	Tip!	Useful tip for simple handling
		Reference to another document

2 Safety instructions

Please observe the following safety instructions when you want to commission an inverter or system using the »Engineer«.



Read the documentation supplied with the inverter or the individual components of the system carefully before you start commissioning the devices with the »Engineer«!

The device documentation contains safety instructions which must be observed!



Danger!

According to today's scientific knowledge it is not possible to ensure absolute freedom from defects of a software product.

The systems with built-in inverters must be provided with additional monitoring and protective equipment according to relevant safety instructions (e.g. law on technical equipment, regulations for the prevention of accidents), so that an impermissible operating status does not endanger persons or facilities.

During commissioning persons must keep a safe distance from the motor or the machine parts driven by the motor. Otherwise there is a risk of injury by the moving machine parts.



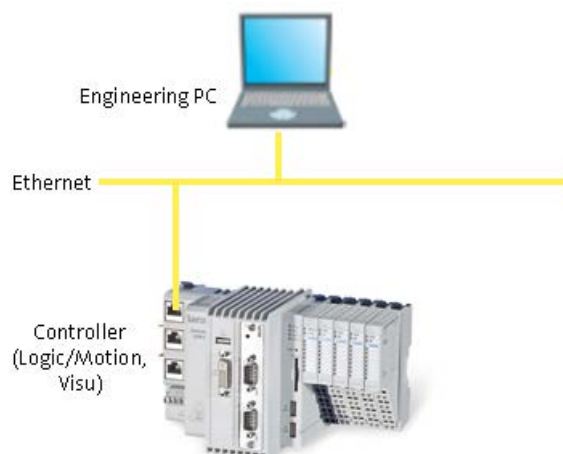
Stop!

If you change parameters in the »Engineer« while the controller is connected online, the changes will be directly accepted by the controller!

A wrong parameter setting can cause unpredictable motor movements. By an unintended direction of rotation, a too high speed, or jerky operation, the driven machine parts may be damaged!

3 Prerequisites

3.1 System requirements



[3-1] Sample configuration using a Controller 3200 C

	Engineering PC	Controllers
Hardware	PC/notebook	PLC (Logic) firmware V3.16.0 or higher
Operating system	Microsoft® Windows® XP Professional (32 bit) from SP3 / Windows® 7 (32 and 64 bit)	Windows CE
Lenze software required	»PLC Designer« V3.16.0.X or higher Contains the "L_EPMP_PackMLManager" Lenze library (from 3.16.0.X)	Runtime software Motion (for this purpose, the project data must be updated: "Update Device")
Further requirements	-	Depending on the application case: - CAN/EtherCAT bus system - CAN-/EtherCAT node

3.2 Establishing communication with the controller

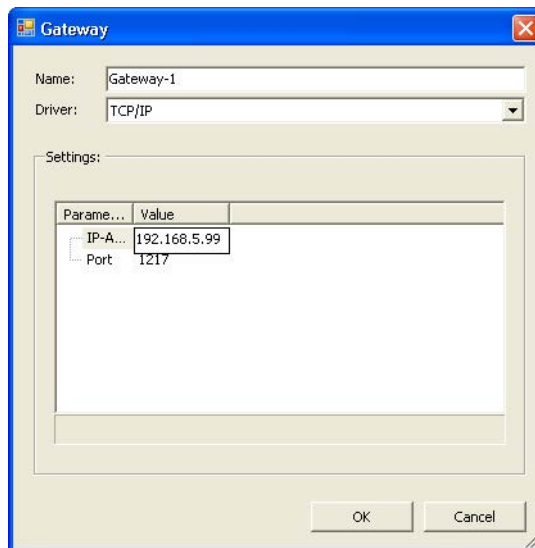
- Connect the Engineering PC with the controller via a network cable. The »PLC Designer« accesses the controller via Ethernet.
- Make the IP settings with the »PLC Designer«.



How to check the communication settings:

1. Double-click the desired controller in the *Devices* view.
2. Make the desired settings on the **Communication settings** tab.
 - Click the **Add gateway** button to insert a gateway.

- Enter the desired IP address of the controller.



[3-2] Example: Enter the IP address of the Controller, standard setting: 192.168.5.99

3. Click **OK** to add the controller as gateway.
4. By double-clicking the desired channel (or clicking the **Set active path** button), set the channel selected in the device view below the gateway as active path to the controller.
 - Thus, all communication actions directly refer to this channel.
 - The currently active path is represented in **bold** in the list and "(active)" is attached:



5. A device represented in *italics* is set as active path but has not been found during the last network scan.



Note!

During initial commissioning, observe the predefined IP address: **192.168.5.99**



Further information can be found in the following documentation:

Controller - Parameter setting & configuration

4 L_EPMP_PackMLManager library

The "L_EPMP_PackMLManager" library contains blocks for the control of machines on the basis of the OMAC PackML.

4.1 OMAC PackML

OMAC: Organization for Machine Automation and Control

PackML: The Packing Machine Language defines a machine language for automated machines in the packaging industry.

The PackML consists of the following components:

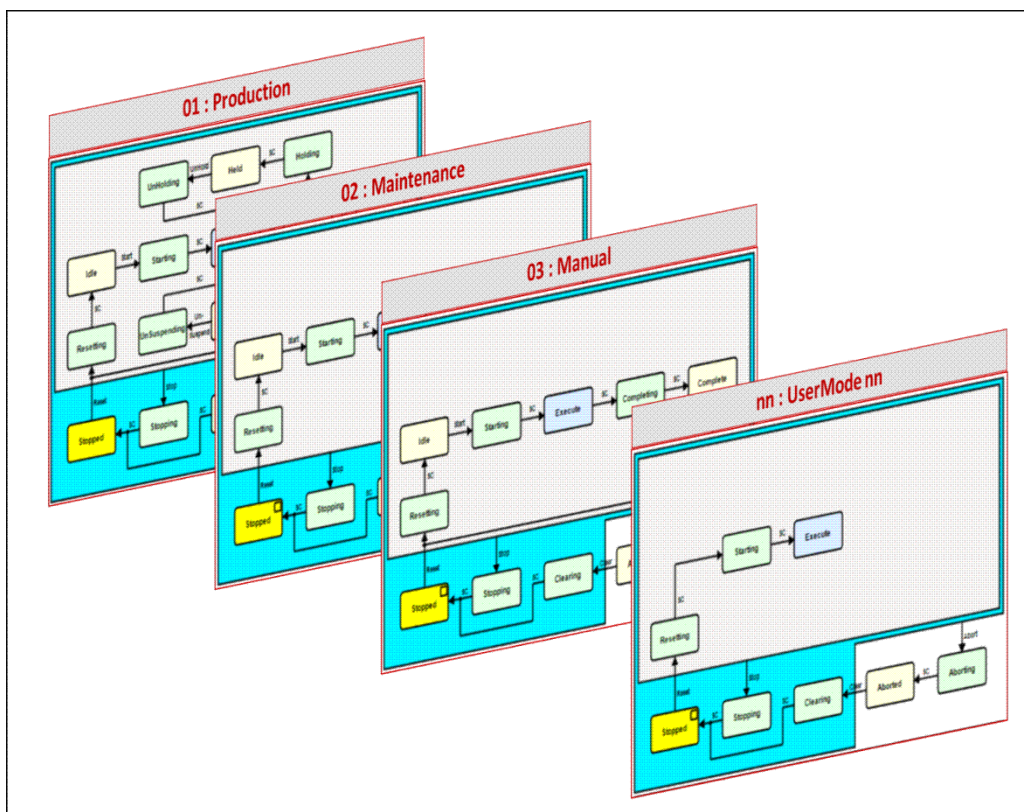
- PackML machine operating modes (modes or unit modes)
- PackML status model (state machine) the structure of which can be configured in the respective mode.

4.1.1 PackML machine operating modes (modes)

The PackML comprises three machine operating modes (modes):

- Production
- Maintenance
- Manual

They are permanently defined and further "User modes" can be added to them.

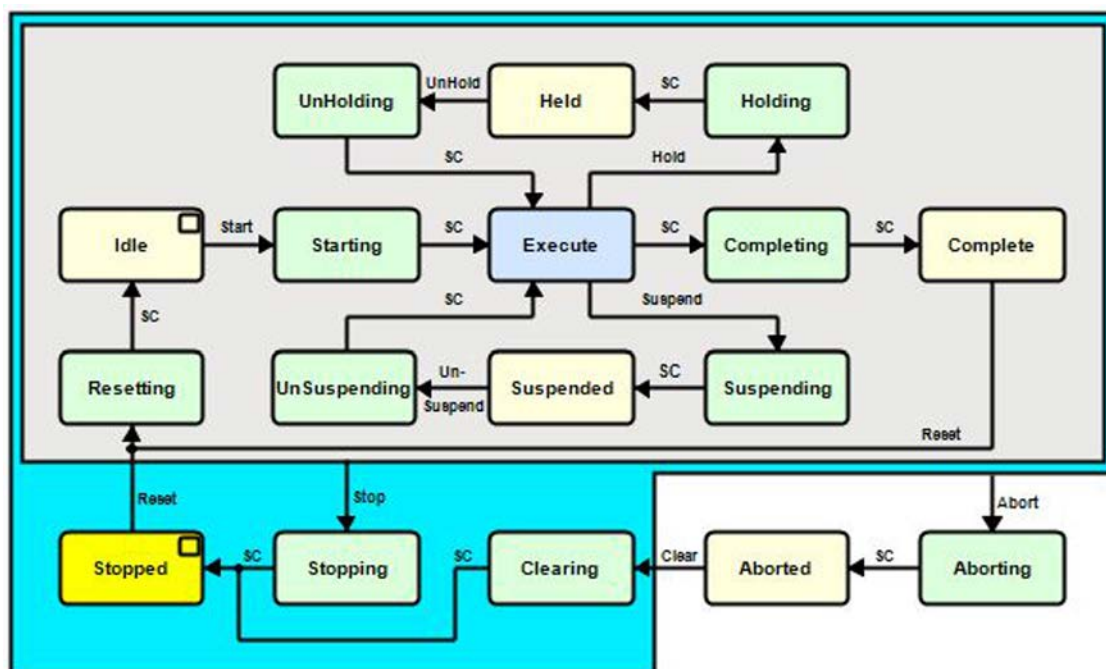


A unique mode number is assigned to each mode:

Control modes in compliance with the TR88.00.02 ISA regulation	
Value (mode number)	Mode
0	Invalid
01	Production
02	Maintenance
03	Manual
04 ... 31	User Defined Mode (user mode) The "L_EPMP_PackMLManager" function block supports max. 5 user modes. ► Specifying modes available (25)

4.1.2 PackML status model (state machine)

The status model of a PackML-compliant application in complete expansion has the following structure:



The PackML states are categorised into so-called WAIT and ACTING states. The EXECUTE state as DUAL state assumes a special role. It is both WAIT and ACTING state.

Furthermore, commands are defined in the PackML, which enable a targeted change-over of a state.

Overview of states

In the table below, all states are described. In addition, it is specified which command triggers a state change-over in the respective state.

State name	State type	Description
STOPPED	WAIT	After the STOPPING state has been left, the machine is switched on and is at a standstill. Communication with other systems is fully functional. The RESET command triggers the change-over to the RESET- TING state from the STOPPED state.
STARTING	ACTING	This state contains the steps required for starting up the machine and is the consequence of a starting command. After a successful start (STATE COMPLETE), a change-over to the EXECUTE state is carried out.
IDLE	WAIT	This state indicates that RESETTING has been completed. The machine states remain unchanged. Operations requested are executed. A change-over to the STARTING state is effected by means of the START command.

State name	State type	Description
SUSPENDING	ACTING	This state should be used if the external process state does not allow the machine to continue producing, i.e. if the upstream or downstream machine in a line leaves the EXECUTE state (typical of blockages or supply bottlenecks in the machine). This state can be detected by a sensor in the machine or by an external control system. Usually, the machine is stopped in a controlled fashion in this state and then a change-over to the SUSPENDED state is effected. In order to be able to start the machine again in a controlled manner after SUSPENDED, all relevant process setpoints and return states of the procedures at the time of the Suspend command must be saved in the machine controller.
SUSPENDED	WAIT	In this state, production of the machine is to pause. It will either be stopped completely or continue moving without producing until the external state has returned to normal. Usually, the state will then change from SUSPENDED to UNSUSPENDING without any intervention by the operator.
UNSUSPENDING	ACTING	This state is reached if the external state has returned to normal. The UNSUSPENDING state initiates all action or sequences required for transferring the machine to the Execute state again. This also includes that the data saved at the time the Suspend command was given can be used for restarting.
Execute	DUAL (WAIT + ACTING)	The machine is in the EXECUTE state when it is performing an action. In the "Producing" mode, for example, the machine will generate products in the EXECUTE state, whereas in the "Clean Out" mode it will use the EXECUTE state to be cleaned.
stopping	ACTING	This state executes the steps which brings the machine to a controlled stop (as also described in STOPPED). The machine cannot be started "normally" until the RESET state has been passed.
ABORTING	ACTING	The ABORTING state can be reached at any time as a result of the Abort command or a machine alarm. The abort logic generates a quick, safe stop of the machine.
ABORTED	WAIT	This state memorises the status information of the machine relevant to an abort process. The machine can only leave the ABORTED state after an express CLEAR command, provided that the triggering error in the machine has been removed and reset.
HOLDING	ACTING	This state should be used if internal machine states prevent the machine from continuing production. This means, the machine leaves the EXECUTE state due to the internal state. Usually, this state is used for machine states occurring at regular intervals, which require intervention by the operator. The state can be initiated automatically or manually by the operator and can also be easily cancelled by the operator again. An example of this would be a machine requiring an operator in order to refill a glue dispenser or carton storage container at regular intervals, which, due to the machine design, cannot be carried out during operation. Since these are normal work routines in the production process, it is not desired that the machine is brought to a standstill via ABORTING or STOPPING. Usually, in this state, the machine is stopped in a controlled fashion and then a change-over to the HELD state is carried out. In order to be able to start the machine again in a controlled manner after the HELD state, all relevant data of the procedures at the time of the Hold command must be saved in the machine controller.
HELD	WAIT	In this state, the machine is not to continue production. The machine stops or continues running in a "dry" fashion. When the internal machine state changes or the UNHOLD command is activated by the operator, the machine changes to the UNHOLDING state.

Versions

- contain all PackML states with all PackML commands (as shown above), or
- contain a subset of all states by masking out states that are irrelevant.

```
graph LR; Idle[Idle] -- Start --> Starting[Starting]; Starting -- SC --> Execute[Execute]; Execute -- SC --> Completing[Completing]; Completing -- SC --> Complete[Complete]; Complete -- Reset --> Resetting[Resetting]; Resetting -- SC --> Idle; Starting -- Stop --> Stopping[Stopping]; Stopping -- SC --> Stopped[Stopped]; Stopped -- SC --> Starting; Clearing[Clearing] -- Clear --> Aborted[Aborted]; Aborting[Aborting] -- SC --> Aborted; Aborting -- Abort --> Aborting;
```

- ▶ Adapting the state model (📖 30)

4.2 Libraries with enumeration types for PackML

The states and commands defined in PackML are represented in libraries with enumeration types:

- L_EPMP_PML_States: representation of the PackML states
- L_EPMP_PML_Commands: representation of the PackML commands

4.2.1 L_EPMP_PML_States

The numerical values assigned to the individual enumeration values comply with the definition laid down in the TR88.00.02 ISA regulation. In addition, the "Undefined" enumeration value is provided (numerical value 0), which represents an invalid status.

Name	Type	Inherited from	Address	Initial	Comment
Undefined	DINT			0	Undefined state
Clearing	DINT			1	State Clearing
Stopped	DINT			2	State Stopped
Starting	DINT			3	State Starting
Idle	DINT			4	State Idle
Suspended	DINT			5	State Suspended
Execute	DINT			6	State Execute
Stopping	DINT			7	State Stopping
Aborting	DINT			8	State Aborting
Aborted	DINT			9	State Aborted
Holding	DINT			10	State Holding
Held	DINT			11	State Held
UnHolding	DINT			12	State Unholding
Suspending	DINT			13	State Suspending
Unsuspending	DINT			14	State Unsuspending
Resetting	DINT			15	State Resetting
Completing	DINT			16	State Completing
Complete	DINT			17	State Complete

4.2.2 L_EPMP_PML_Commands

The numerical values assigned to the individual enumeration values comply with the definition laid down in the TR88.00.02 ISA regulation. In addition, the "None" enumeration value is provided (numerical value 0), to indicate that no command is active.

Name	Type	Inherited from	Address	Initial	Comment
None	DINT			0	No command active
Reset	DINT			1	Cmd Reset
Start	DINT			2	Cmd Start
Stop	DINT			3	Cmd Stop
Hold	DINT			4	Cmd Hold
Unhold	DINT			5	Cmd Unhold
Suspend	DINT			6	Cmd Suspend
Unsuspend	DINT			7	Cmd Unsuspend
Abort	DINT			8	Cmd Abort
Clear	DINT			9	Cmd Clear

4.3 Blocks for PackML

The following function blocks provide the basis for the control of machines based on the OMAC PackML in the »PLC Designer«:

[L_EPMP_PackMLManager](#) (📖 17)

- ▶ [Executing a mode change-over](#) (📖 28)
- ▶ [Executing a state change-over](#) (📖 32)

[L_EPMP_PML_Config](#) (📖 20)

- ▶ [Specifying modes available](#) (📖 25)
- ▶ [Specifying the initial mode after system start-up](#) (📖 26)
- ▶ [Specifying the initial state after system start-up](#) (📖 31)

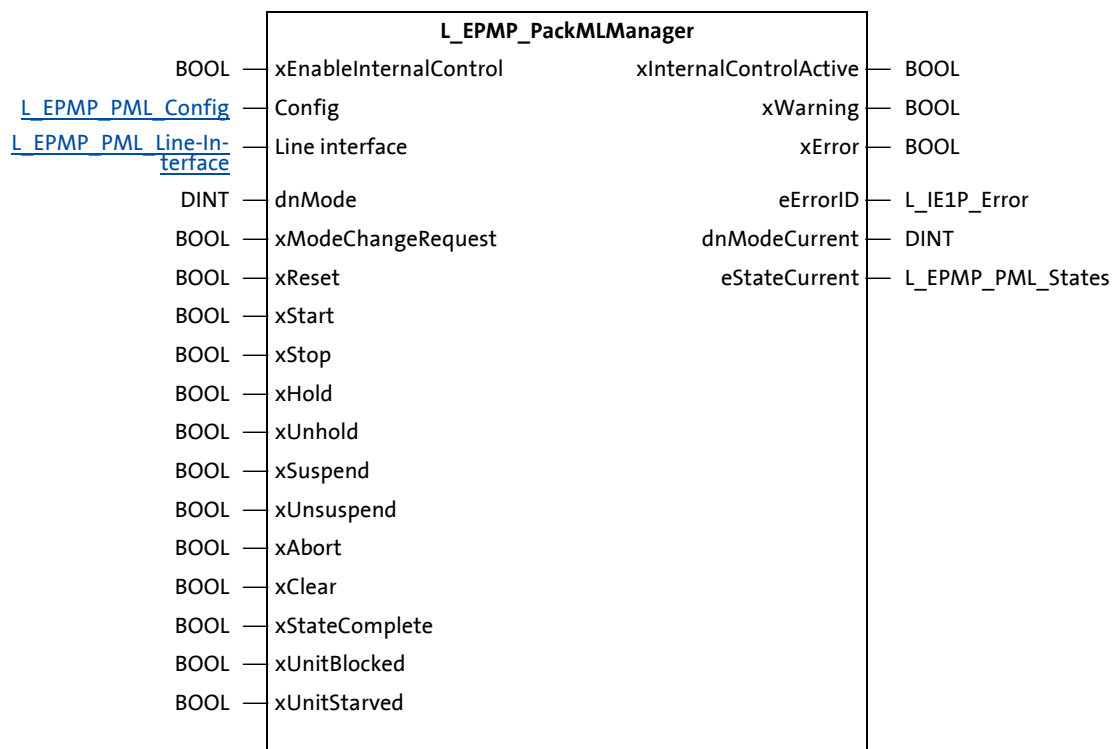
[L_EPMP_PML_ModeConfig](#) (📖 22)

- ▶ [Allowing and inhibiting mode change-overs](#) (📖 27)
- ▶ [Adapting the state model](#) (📖 30)

[L_EPMP_PML_Line-Interface](#) (📖 23)

- ▶ [Line interface and pack tags](#) (📖 33)

4.3.1 L_EPMP_PackMLManager



inputs

Designator	Data type	Description
xEnableInternalControl	BOOL	TRUE Internal control of the axis can be selected via the "Internal Control" button (button is visible in the visualisation).
Config L_EPMP_PML Config		Configuration of the PackML mode and state model. An instance of the "L_EPMP_PML_Config" block by means of which the PackML mode and state model is configured can be connected to this input. - The configuration is imported once in the first cycle following PLC reset. - If the <i>Config</i> input is not connected, the following standard configuration is activated: - Mode 01: PRODUCTION - Mode 02: MAINTENANCE - Mode 03: MANUAL - In all modes, all PackML states are available. - A mode change-over can be effected in the "Stopped", "Idle", "Complete", "Aborted", "Held" and "Suspended" states. - Initial mode after PLC reset is mode 01 (Production). - Initial state after PLC reset is "Stopped"
Line interface L_EPMP_PML Line-Interface		Connection of the PackML line interface block for controlling the unit/machine by a higher-level line controller.
dnMode	DINT	Mode number which is to be valid for one mode change request.
xModeChangeRequest	BOOL	FALSE↗TRUE A rising edge activates a mode change request. The mode number is specified at the <i>dnMode</i> input. In order to ensure that a mode change request is successful, the following conditions must be met: - The mode number created must be valid and configured. - The state model must be in a state in which a mode change-over is permissible.

Designator	Data type	Description	
xReset	BOOL	FALSE→TRUE	Activate a "Reset" PackML command.
xStart	BOOL	FALSE→TRUE	Activate a "Start" PackML command.
xStop	BOOL	FALSE→TRUE	Activate a "Stop" PackML command.
xHold	BOOL	FALSE→TRUE	Activate a "Hold" PackML command.
xUnhold	BOOL	FALSE→TRUE	Activate an "Unhold" PackML command.
xSuspend	BOOL	FALSE→TRUE	Activate a "Suspend" PackML command.
xUnsuspend	BOOL	FALSE→TRUE	Activate an "Unsuspend" PackML command.
xAbort	BOOL	FALSE→TRUE	Activate an "Abort" PackML command.
xClear	BOOL	FALSE→TRUE	Activate a "Clear" PackML command.
xStateComplete	BOOL	FALSE→TRUE	Activate a "StateComplete" event. The machine or unit logic uses this to indicate that a so-called PackML acting state (e.g. "Starting") is completed.
xUnitBlocked	BOOL	TRUE	The product processed cannot be transferred to the subsequent unit, for example because the output buffer is full.
xUnitStarved	BOOL	TRUE	No products for processing in this machine are provided in the upstream unit (e.g. input buffer empty).

outputs

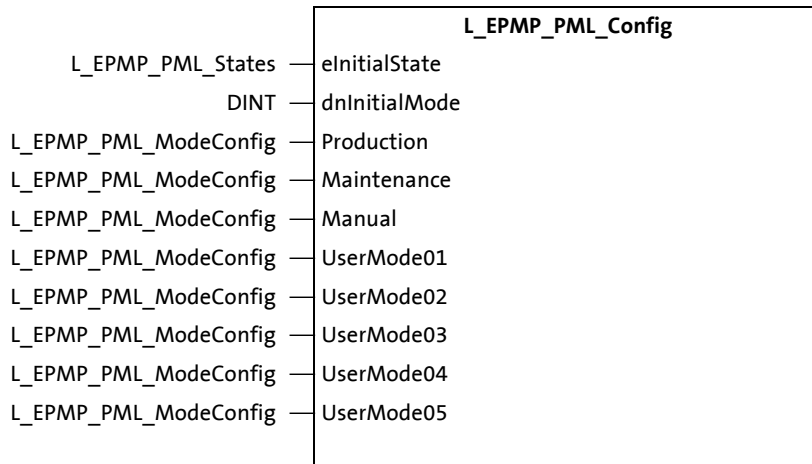
Designator	Data type	Description	
xInternalControlActive	BOOL	TRUE	The internal control unit via the internal control visualisation is active.
xWarning	BOOL	TRUE	There is a warning in the FB.
xError	BOOL	TRUE	There is an error in the FB.
eErrorID	L_IE1P_Error	ID of the error or warning message if xError = TRUE or xWarning = TRUE. "FAST technology modules" reference manual: Here you can find information on error or warning messages.	
dnModeCurrent	DINT	Number of the PackML mode currently active	
eStateCurrent	L_EPMP_PML_States	PackML state currently active	

Errors and warnings

The FB indicates the following control-related errors and warnings at the eErrorID output.

Designator	Description
None [0]	No warning active / no error active
ModeNumber OutsideRange_1_32 [18001] (ERROR)	A mode number was requested, which is outside the valid range (1 ... 32).
ModeNumber NotUsedInConfig [18002] (ERROR)	A mode number was requested which is not used in the module configuration.
ModeChange NotAllowedInCurrentMode [18003] (warning)	A mode change was requested, however, it is impermissible due to the mode configuration of the current mode.
ModeChange NotAllowedInNextMode [18004] (warning)	A mode change was requested, however, it is impermissible due to the mode configuration of the next mode.
LI_ModeNumber OutsideRange_1_32 [18005] (ERROR)	A mode number was requested via the line interface, which is outside the valid range (1 ... 32).
LI_ModeNumber NotUsedInConfig [18006] (ERROR)	A mode number was requested via the line interface, which is not used in the module configuration.
LI_ModeChangeNot AllowedRemote [18007] (ERROR)	The only remote mode change-over permissible via the line interface is a change-over to the "Production" mode (1).
LI_ModeChange NotAllowedInCurrentMode [18008] (warning)	A mode change-over was requested via the line interface, however, it is impermissible due to the configuration of the current mode in the current state.
LI_ModeChange NotAllowedInNextMode [18009] (warning)	A mode change-over was requested via the line interface, however, in the current state, it is impermissible due to the configuration of the mode to which the change-over is to take place.
MoreThanOne CommandBitActive [18011] (warning)	A TRUE signal is pending on at least one command input (<i>xReset ... xClear</i>). Since the command inputs work with edge evaluation, this is to be avoided.
MultipleCommandError [18012] (ERROR)	A rising edge was recognised isochronously on several command inputs (<i>xReset ... xClear</i>). Due to this ambiguity, no command is executed.
CommandNotAllowed InThisState [18013] (warning)	The PackML command requested is not permissible in the current state and is therefore ignored.
LI_CommandsOnly InModeProduction [18014] (warning)	PackML commands can only be processed via the line interface in the "Production" mode (1).
LI_CommandNotAllowed Remote [18015] (ERROR)	A command was requested via the line interface, which is not permissible in the "Remote" mode.
LI_CommandNotAllowed InThisState [18016] (warning)	The PackML command requested via the line interface is not permissible in the current state and is therefore ignored.

4.3.2 L_EPMP_PML_Config



inputs

Designator	Data type	Description
elInitialState L_EPMP_PML_States		Define initial state after PLC reset.
dnInitialMode DINT		Define the initial mode after PLC reset via the mode number.
Production L_EPMP_PML_ModeConfig		Here, an instance of the "L_EPMP_PML_ModeConfig" block can be connected, in order to configure the "Production" mode (mode number 01). With an unconnected input, the following standard configuration is used: • Mode name: "Production" • All PackML states are available. • A mode change-over can be effected in the "Stopped", "Idle", "Complete", "Aborted", "Held" and "Suspended" states.
Maintenance L_EPMP_PML_ModeConfig		Here, an instance of the "L_EPMP_PML_ModeConfig" block can be connected, in order to configure the "Maintenance" mode (mode number 02). With an unconnected input, the following standard configuration is used: • Mode name: "Maintenance" • All PackML states are available. • A mode change-over can be effected in the "Stopped", "Idle", "Complete", "Aborted", "Held" and "Suspended" states.
Manual L_EPMP_PML_ModeConfig		Here, an instance of the "L_EPMP_PML_ModeConfig" block can be connected, in order to configure the "Manual" mode (mode number 03). With an unconnected input, the following standard configuration is used: • Mode name: "Manual" • All PackML states are available. • A mode change-over can be effected in the "Stopped", "Idle", "Complete", "Aborted", "Held" and "Suspended" states.
UserMode01 L_EPMP_PML_ModeConfig		Here, an instance of the "L_EPMP_PML_ModeConfig" block can be connected, in order to configure UserMode01 (mode number 04). With an unconnected input, this UserMode is not used.
UserMode02 L_EPMP_PML_ModeConfig		Here, an instance of the "L_EPMP_PML_ModeConfig" block can be connected, in order to configure UserMode02 (mode number 05). With an unconnected input, this UserMode is not used.
UserMode03 L_EPMP_PML_ModeConfig		Here, an instance of the "L_EPMP_PML_ModeConfig" block can be connected, in order to configure UserMode03 (mode number 06). With an unconnected input, this UserMode is not used.

Designator	Data type	Description
UserMode04	L_EPMP_PML_ModeConfig	Here, an instance of the "L_EPMP_PML_ModeConfig" block can be connected, in order to configure UserMode04 (mode number 07). With an unconnected input, this UserMode is not used.
UserMode05	L_EPMP_PML_ModeConfig	Here, an instance of the "L_EPMP_PML_ModeConfig" block can be connected, in order to configure UserMode05 (mode number 08). With an unconnected input, this UserMode is not used.

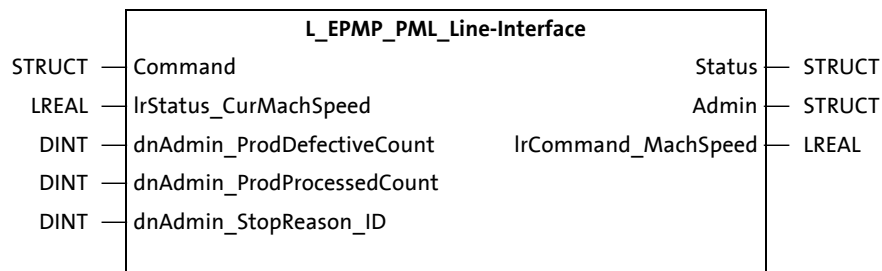
4.3.3 L_EPMP_PML_ModeConfig



inputs

Designator	Data type	Description
sName	STRING	TRUE Plain text name of the mode configured.
xDisableStateNnn	BOOL	TRUE The state specified from the state model of the mode configured is masked out.
xEnableModeTransNnn	BOOL	TRUE In the respective state, a mode change-over to or from the mode configured is permissible.

4.3.4 L_EPMP_PML_Line-Interface_TR880002_Minimum

**inputs**

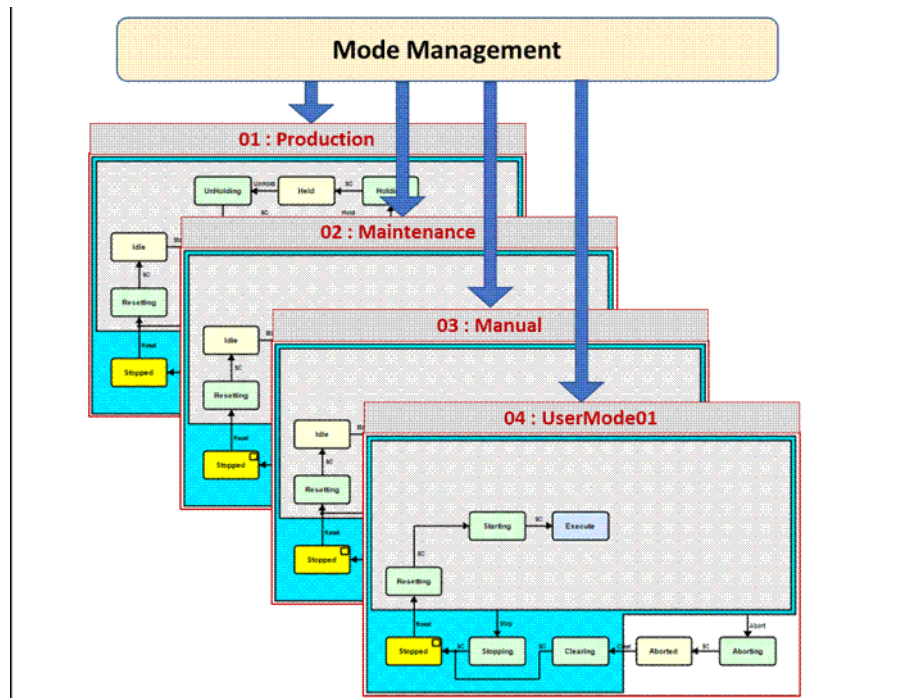
Designator	Data type	Description
Command	STRUCT	Connection of the global structure instance for the command pack tags. ► Line interface and pack tags (□ 33)
IrStatus_CurMachSpeed	LREAL	Value of the current machine speed from the application. This value is transferred to the line controller in the status pack tags.
dnAdmin_ProdDefectiveCount	DINT	Current value of the defective products counter from the application. This value is transferred to the line controller in the admin pack tags.
dnAdmin_ProdProcessedCount	DINT	Current value of the processed products counter from the application. This value is transferred to the line controller in the admin pack tags.
dnAdmin_StopReason_ID	DINT	Number of the cause of the last STOP activated in the unit. This value is transferred to the line controller in the admin pack tags.

outputs

Designator	Data type	Description
Status	STRUCT	Connection of the global structure instance for the status PackTags.
Admin	STRUCT	Connection of the global structure instance for the admin pack tags.
IrCommand_MachSpeed	LREAL	Default value (setpoint) for the machine speed. This value is received from the line controller in the command pack tags.

4.4 Configuration

4.4.1 Mode management



Overview of the configuration

Specifying modes available: Specify modes that are to be available in the application.

Specifying the initial mode after system start-up: Specify mode adopted by a system for the first time if it has been started initially (mains switching, PLC reset, etc.).

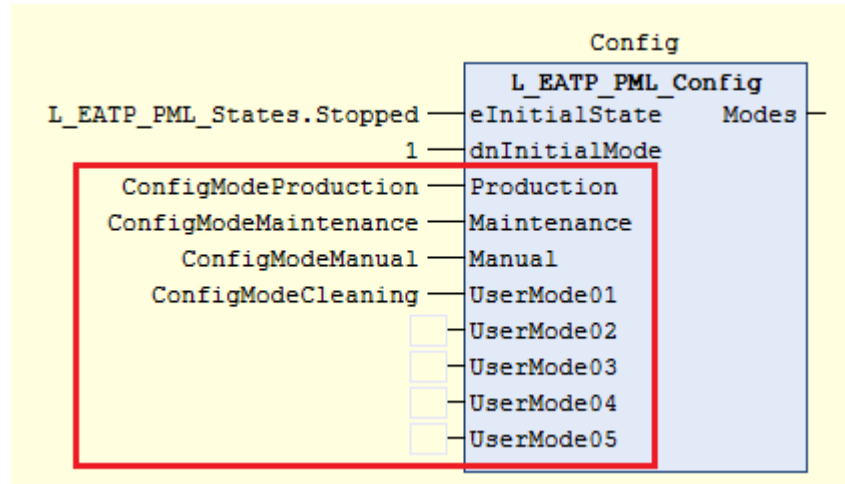
Allowing and inhibiting mode change-overs: As laid down in the specifications of the TR88.00.02 ISA regulation, mode change-overs are generally only possible if the system is in a so-called "WaitState" or "DualState" in the current mode. Depending on the application, this condition can be restricted to an extent only making it permissible to carry out a mode change-over in a subset of this state. Therefore, the configuration for each mode provides settings for defining in which states a mode change-over to this mode or from this mode is permissible.

For a specific mode change-over from "M1" to "M2" (assuming that the "S" state is currently active) this means that the mode change-over is only permissible if both "M1" and "M2" are configured so that a mode change-over in state S is permitted.

Executing a mode change-over: An input circuit is required, initiating and executing a mode change-over. Furthermore, an output circuit is required for the purpose of feedback, specifying which mode is currently active.

4.4.1.1 Specifying modes available

Which modes are available in an application is defined via the central "L_EPMP_PML_Config" configuration block:

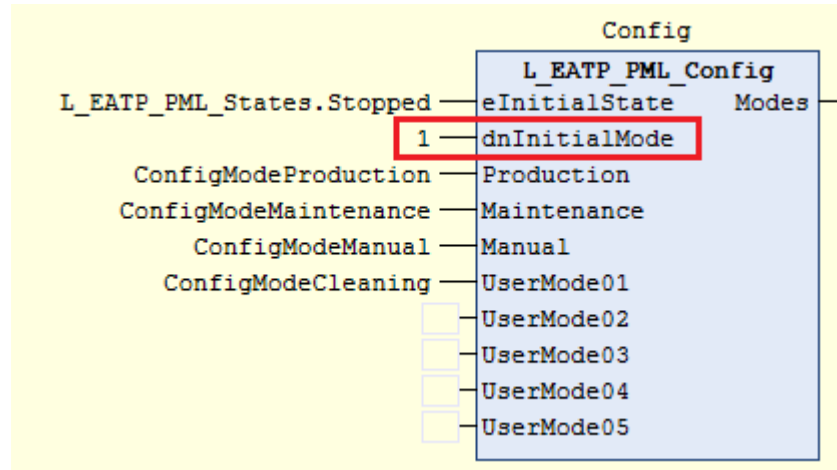


Mode configuration blocks for the 3 standard modes and up to 5 user modes can be connected to the inputs of this FB.

► [L_EPMP_PackMLManager](#) (17)

4.4.1.2 Specifying the initial mode after system start-up

The initial mode after the system start-up is defined via the central "L_EPMP_PML_Config" configuration block:



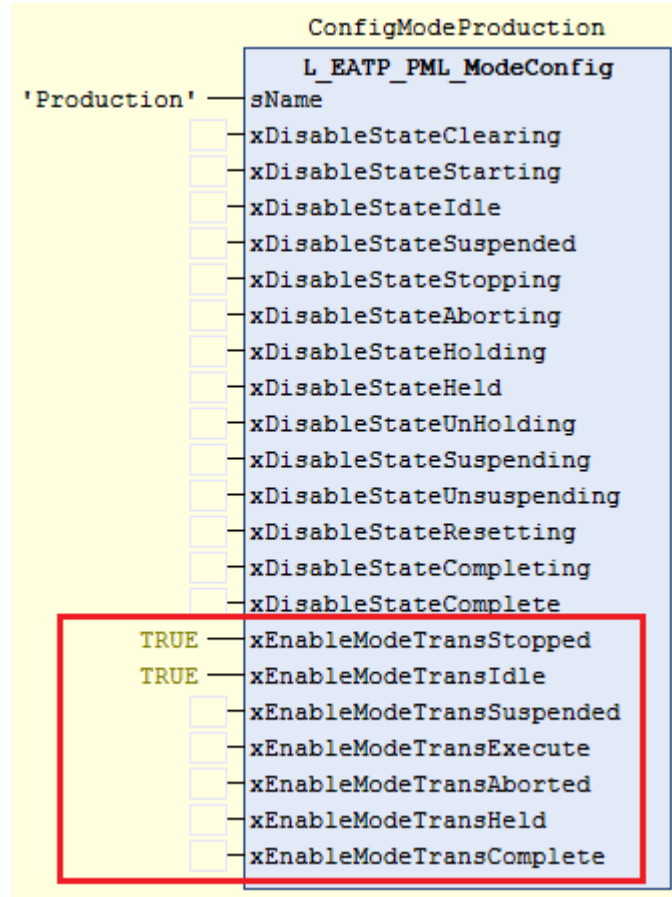
At the *dnInitialMode* input, the mode number of the desired initial mode is expected.

Mode number	Mode
0	Invalid
01	Production
02	Maintenance
03	Manual
04 ... 08	User Defined Mode (UserMode)

► [L_EPMP_PackMLManager](#) (17)

4.4.1.3 Allowing and inhibiting mode change-overs

The states in which a mode change-over is to be permissible are specified at various inputs of the "L_EPMP_ModeConfig" mode configuration block:

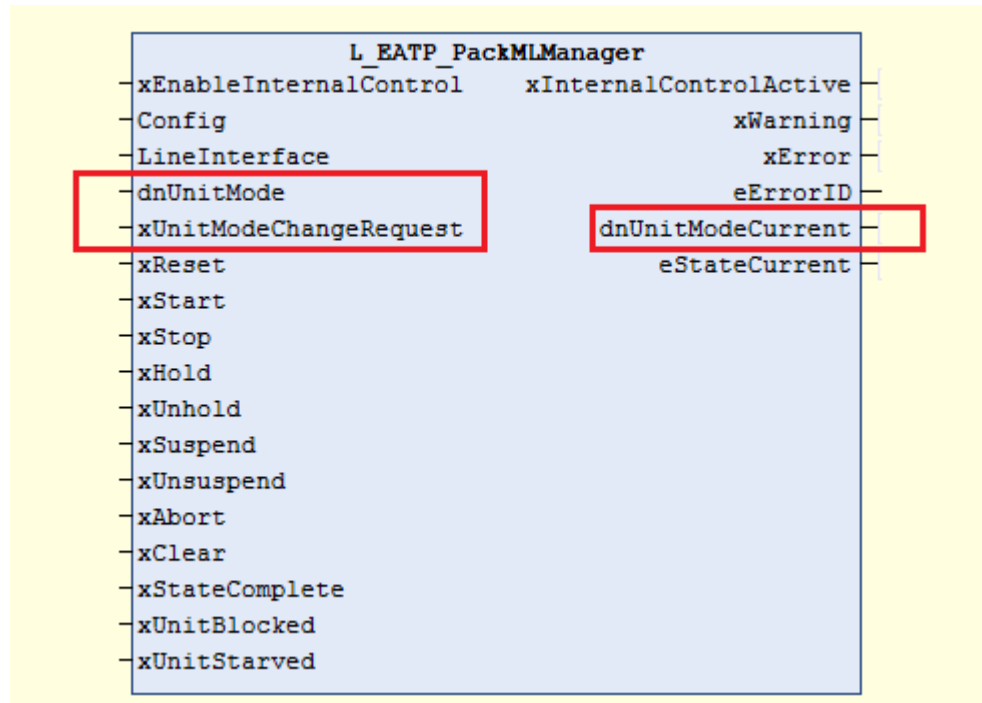


Each *xEnableModeTrans...* input that is assigned with TRUE, enables the mode change-over in this state. In the example shown this would be the "Stopped" and "Idle" states.

► [L_EPMP_PackMLManager](#) (17)

4.4.1.4 Executing a mode change-over

Control and feedback of a mode change-over are executed at the central "L_EPMP_PackMLManager" block:

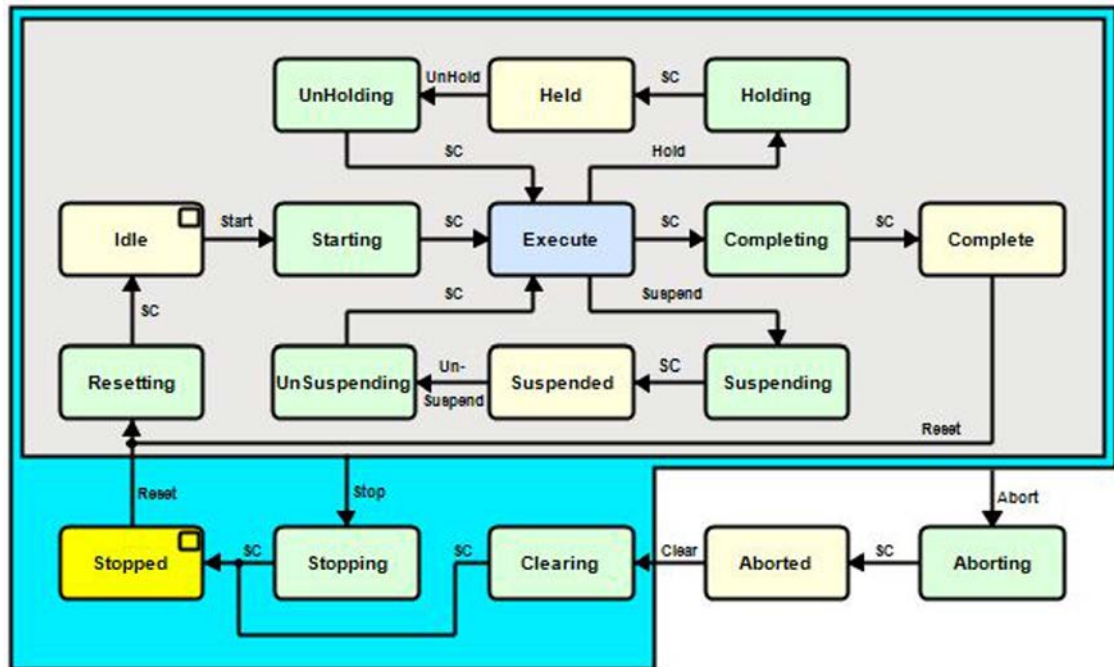


As regards the signal response, the mode change-over control via inputs *dnMode* and *xModeChangeRequest* complies with the definition of the pack tags from the TR88.00.02 ISA regulation. This also applies to the feedback of the current mode at the *dnModeCurrent* block output.

► [L_EPMP_PackMLManager](#) (17)

4.4.2 State management

The term "state management" refers to the mechanisms ensuring the management of the current state along with the PackML commands in the state model.



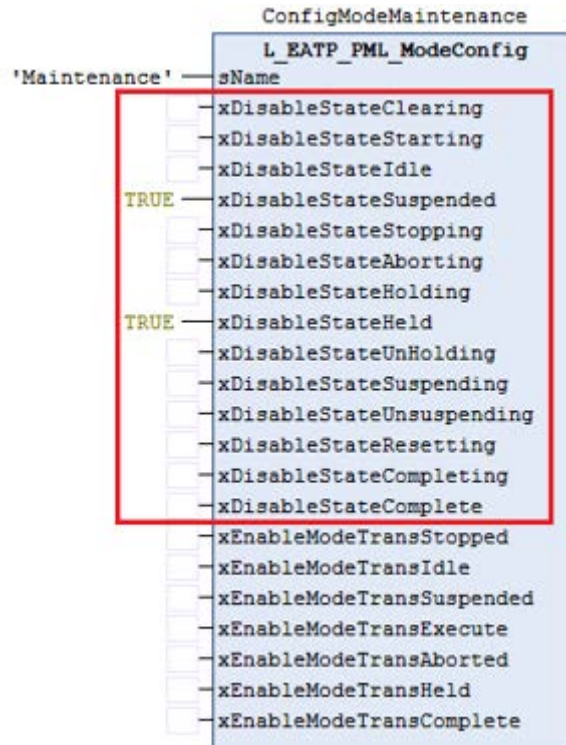
The individual aspects comprise:

[Specifying the initial state after system start-up](#): Specify state adopted by a system for the first time if it has been started initially (mains switching, PLC reset, etc.).

[Executing a state change-over](#): An input circuit is required, initiating and executing a state change-over. Furthermore, an output circuit is required for the purpose of feedback, specifying which state is currently active.

4.4.2.1 Adapting the state model

Via inputs of the "L_EPMP_ModeConfig" configuration block, the state model can be adapted by masking out states that are not required.



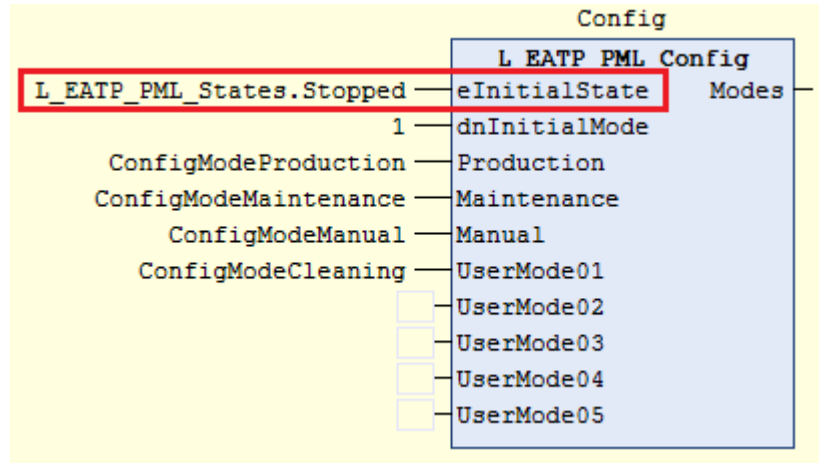
If TRUE is pending on a "xDisableState..." input, the state assigned is deactivated, i.e. masked out.

The following basic conditions and automatic mechanisms have to be taken into consideration:

- The "Aborted", "Stopped" and "Execute" states cannot be masked out.
- If the "Held" state is deactivated, the "Holding" and "UnHolding" states are automatically deactivated at the same time.
- If the "Suspended" state is deactivated, the "Suspending" and "Unsuspending" states are automatically deactivated at the same time.
- If the "Complete" state is deactivated, the "Completing" state is automatically deactivated at the same time.

4.4.2.2 Specifying the initial state after system start-up

The initial state after system start-up is defined via the central "L_EPMP_PML_Config" configuration block:

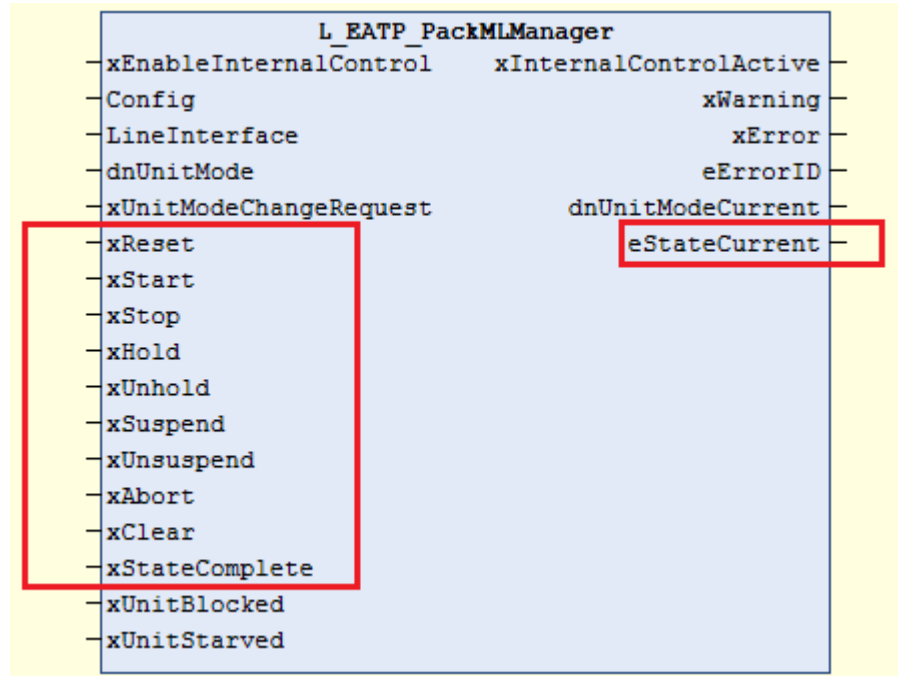


At the *eInitialState* input, the state enumeration constant of the desired initial state after system start-up is expected.

► [L_EPMP_PackMLManager](#) (17)

4.4.2.3 Executing a state change-over

Control and feedback of a state change-over are executed at the central "L_EPMP_PackMLManager" block:



A state change-over is executed via the xReset ... xClear PackML command inputs and via the xState-Complete input. This activation complies with the state model of the TR88.00.02 ISA regulation, which also applies to the feedback of the current state at the eStateCurrent block output.

► [L_EPMP_PackMLManager](#) (17)

5 Line interface and pack tags

The term "Line interface" in the TR88.00.02 ISA regulation refers to the interface to a higher-level control system for a production line (line controller). The basis for mapping the data of the interface is provided by so-called "Pack tags", which are divided into the groups "ControlTags" (control of the machine/unit), "StatusTags" (monitoring of the machine/unit) and "AdminTags" (data of the machine/unit for further production analysis).

In the regulation and the corresponding implementation guide, the "PackTags" are defined in different degrees of detail. A basic scope is presented by the "TR880002_Minimum" subset for which data structures and a function block are included in the library described here, enabling the provision of a line interface with this basic scope on "PackTags" with minimum application expenditure.

In the following sections, we'll first describe the scope of the PackTag in detail before the applicative provision of a line interface based on this is shown by the use of examples. Eventually, the mode and state command scopes are differentiated, which, compared to the complete PackML mode and state model, are limited since control is executed by a higher-level line controller (that is usually located at a distance).

5 Line interface and pack tags

5.1 Pack tag scope TR880002_Minimum

5.1 Pack tag scope TR880002_Minimum

The pack tags containing an X in the following graphics in the "TR 88.00.02" column are implemented.

PackTag type	PackTag	Example of End user term	Datatype	TR 88.00.02 Minimum tags	End user Minimum tags
Status	StateCurrent	State	INT(32)	X	X
Status	UnitModeCurrent	Mode	INT(32)	X	X
Status	MachSpeed	Nominal Speed	REAL	X	X
Status	CurMachSpeed	Current Speed	REAL	X	X
Status	EquipmentInterlock.Blocked	Blockage	BIT	X	X
Status	EquipmentInterlock.Starved	Starvation	BIT	X	X
Status	Parameter [#]	Machine data/parameter	Array Structure		X
Status	Parameter[#].ID	Parameter ID	INT(32)		X
Status	Parameter[#].Name	Name of parameter	STRING		X
Status	Parameter[#].Unit	Unit of measure	STRING[5]		X
Status	Parameter[#].Value	Value of parameter	User Defined		X
Status	RemoteInterface.Parameter[#]	Additional production data	Structure		X
Status	RemoteInterface.Parameter[#].ID	Parameter ID	INT(32)		X
Status	RemoteInterface.Parameter[#].Name	Name of parameter	STRING		X
Status	RemoteInterface.Parameter[#].Unit	Unit of measure	STRING[5]		X
Status	RemoteInterface.Parameter[#].Value	Value of parameter	REAL		X
Admin	Warning[#]	Warning	Array Structure		X
Admin	Warning[#].Trigger	Trigger	Bool		X
Admin	Warning[#].ID	ID	Int (32bit)		X
Admin	Warning[#].Value	Value	Int (32bit)		X
Admin	ProdDefectiveCount	OEE.Bad count	INT(32)	X	X
Admin	ProdProcessedCount	OEE.Total count	INT(32)	X	X
Admin	StopReason.ID	Event and stop reason	INT(32)	X	X
Admin	StopReason.Value	Detailed Error Information	INT(32)		X
Command	CntrlCmd	Command	INT(32)	X	X
Command	Parameter [#]	Machine data/parameter	Array Structure		X
Command	Parameter[#].ID	Parameter ID	INT(32)		X
Command	Parameter[#].Name	Name of parameter	STRING		X
Command	Parameter[#].Unit	Unit of measure	STRING[5]		X
Command	Parameter[#].Value	Value of parameter	User Defined		X
Command	RemoteInterface.Parameter [#]	Additional Production data	Array Structure		X
Command	RemoteInterface.Parameter[#].ID	Parameter ID	INT(32)		x
Command	RemoteInterface.Parameter[#].Name	Name of parameter	STRING		x
Command	RemoteInterface.Parameter[#].Unit	Unit of measure	STRING[5]		x
Command	RemoteInterface.Parameter[#].Value	Value of parameter	REAL		x
Command	UnitMode	Mode	INT(32)	X	X
Command	UnitModeChangeRequest	Change mode	BOOL	X	X
Command	MachSpeed ²⁶	Mach Speed	REAL	X	X
Command	CmdChangeRequest	Change command	BOOL	X	X

[5-1] Source of the graphics: PACKML UNIT/MACHINE IMPLEMENTATION GUIDE, Table 19.

5 Line interface and pack tags

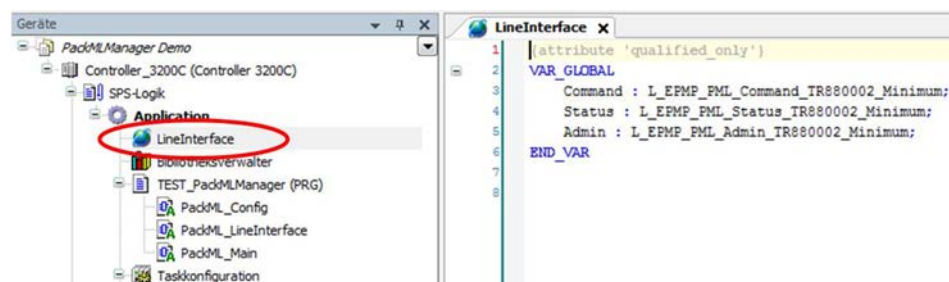
5.2 Use of the line interface block

5.2 Use of the line interface block

In this section, we'll describe by means of an example in which way the line interface block can be used in an application.

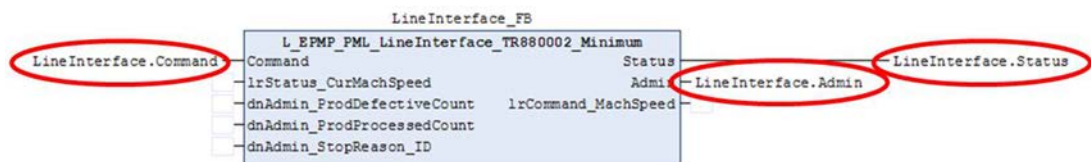
5.2.1 GVL for the line interface

A GVL (name "Line interface") is created, into which an instance of the command, status and admin structure are included in each case.



5.2.2 Linkage with the line interface block

The global instances of the command, status and admin structure are linked with the line interface block.



5.2.3 Connection of the GVL instances

The global instances of the command, status and admin structure in the GVL can now for example be included in the symbol configuration in order to link them with the external line controller via OPC/UA or another communication system.

The precise procedure depends on the respective communication system.

5.2.4 Limited control scope via the line interface

A limited scope of control commands is provided via the line interface.

Mode change-over

A change-over to the "Production" mode (01) can be carried out if the other basic conditions are met.

PackML commands

The following PackML commands are provided in the "Production" mode:

- Clear
- Reset
- Start
- Stop,
- Suspend
- Unsuspend.

5 Line interface and pack tags

5.3 InternalControl and visualisation frame

5.3 InternalControl and visualisation frame

For the InternalControl mechanism, analogous to the Lenze technology modules, an input and output, respectively, are provided at the "L_EPMP_PackMLManager" basic block:



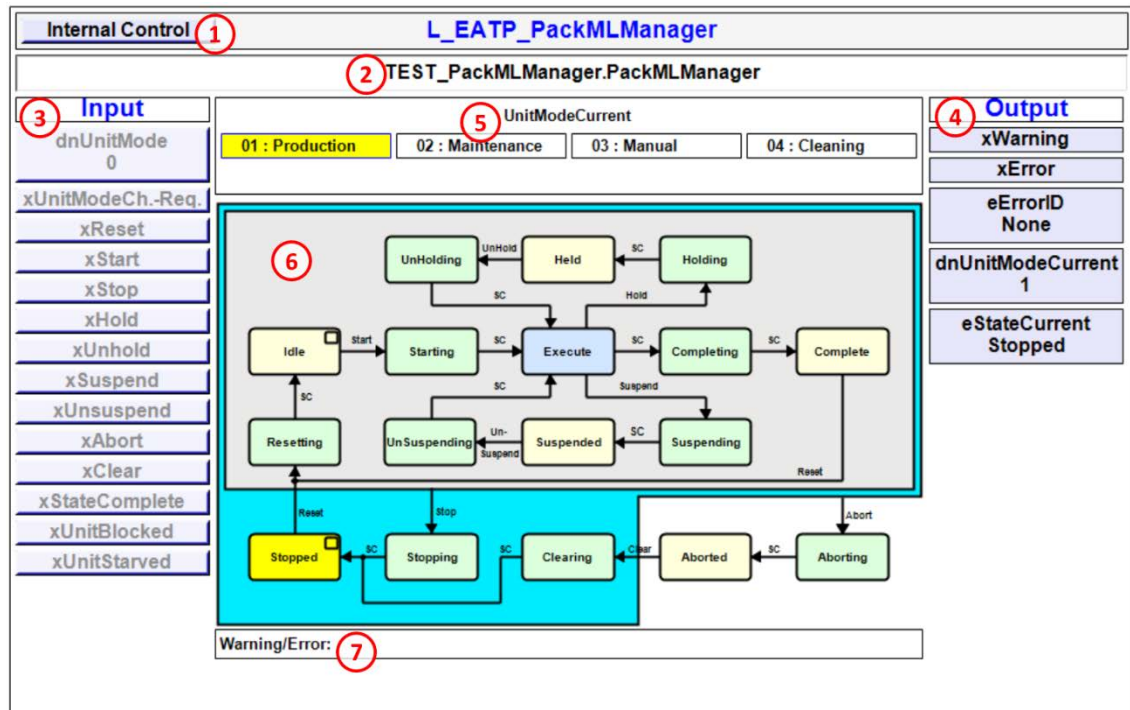
If `TRUE` is assigned to the `xEnableInternalControl` input, the "InternalControl" mechanism can be activated via the visualisation frame. With `FALSE` it is inhibited.

The `xInternalControlActive` output shows that "InternalControl" is currently activated.

5 Line interface and pack tags

5.3 InternalControl and visualisation frame

The "L_EPMP_VisPackMLManager" visualisation frame of the "L_EPMP_PackMLManager" FB has the following structure:



It contains the following elements:

1. Button for activating InternalControl
2. Full block instance name in the application
3. Visualisation of the block inputs
4. Visualisation of the block outputs
5. Visualisation of the modes available and of the active mode
6. Visualisation of the state model and the active state
7. Visualisation of the block errors and warnings

Index

A

Application notes (representation) [6](#)

C

Conventions used [5](#)

D

Document history [4](#)

E

E-mail to Lenze [41](#)

Establishing communication with the controller [8](#)

F

Feedback to Lenze [41](#)

I

inputs [17](#), [20](#), [22](#), [23](#)

L

L_EPMP_PackMLManager [10](#), [17](#)

L_EPMP_PackMLManager library [10](#)

L_EPMP_PML_Commands [15](#)

L_EPMP_PML_Config [20](#)

L_EPMP_PML_Line-Interface_TR880002_Minimum [23](#)

L_EPMP_PML_ModeConfig [22](#)

L_EPMP_PML_States [15](#)

Libraries with enumeration types for PackML [15](#)

Line interface [33](#)

M

Mode [11](#)

Allowing/inhibiting change-overs [27](#)

Executing a mode change-over [28](#)

Initial mode [26](#)

Management [24](#)

Modes available [25](#)

N

Notes used [6](#)

O

OMAC [10](#)

outputs [18](#), [23](#)

P

Pack tags [33](#)

PackML [10](#)

Blocks [16](#)

Machine operating modes [11](#)

Status model [12](#)

S

Safety [7](#)

Safety instructions (representation) [6](#)

Screenshots [4](#)

State

Adapting the model [30](#)

Executing a state change-over [32](#)

Initial state [31](#)

Machine [12](#)

Management [29](#)

States available [15](#)

T

Target group [4](#)

V

Validity of the documentation [4](#)



Your opinion is important to us

These instructions were created to the best of our knowledge and belief to give you the best possible support for handling our product.

Perhaps we have not succeeded in achieving this objective in every respect. If you have suggestions for improvement, please e-mail us to:

feedback-docu@lenze.com

Thank you very much for your support.

Your Lenze documentation team

Lenze Automation GmbH
Postfach 10 13 52, 31763 Hameln
Hans-Lenze-Straße 1, 31855 Aersen
Germany
HR Hannover B 205381
 +49 5154 82-0
 +49 5154 82-2800
 sales.de@lenze.com
 www.lenze.com

Service

Lenze Service GmbH
Breslauer Straße 3, 32699 Extertal
Germany
 008000 24 46877 (24 h helpline)
 +49 5154 82-1112
 service.de@lenze.com