OMAC PackML V2022 for S7-1200 / S7-1500

LPMLV2022 for SIMATIC

https://support.industry.siemens.com/cs/ww/en/view/109821198

Siemens Industry Online Support

Legal information

Use of application examples

Application examples illustrate the solution of automation tasks through an interaction of several components in the form of text, graphics and/or software modules. The application examples are a free service by Siemens AG and/or a subsidiary of Siemens AG ("Siemens"). They are non-binding and make no claim to completeness or functionality regarding configuration and equipment. The application examples merely offer help with typical tasks; they do not constitute customer-specific solutions. You yourself are responsible for the proper and safe operation of the products in accordance with applicable regulations and must also check the function of the respective application example and customize it for your system.
Siemens grants you the non-exclusive, non-sublicensable and non-transferable right to have the application examples used by technically trained personnel. Any change to the application examples is your responsibility. Sharing the application examples with third parties or copying the application examples or excerpts thereof is permitted only in combination with your own products. The application examples are not required to undergo the customary tests and quality inspections of a chargeable product; they may have functional and performance defects as well as errors. It is your responsibility to use them in such a manner that any malfunctions that may occur do not result in property damage or injury to persons.

Disclaimer of liability

Siemens shall not assume any liability, for any legal reason whatsoever, including, without limitation, liability for the usability, availability, completeness and freedom from defects of the application examples as well as for related information, configuration and performance data and any damage caused thereby. This shall not apply in cases of mandatory liability, for example under the German Product Liability Act, or in cases of intent, gross negligence, or culpable loss of life, bodily injury or damage to health, non-compliance with a guarantee, fraudulent non-disclosure of a defect, or culpable breach of material contractual obligations. Claims for damages arising from a breach of material contractual obligations shall however be limited to the foreseeable damage typical of the type of agreement, unless liability arises from intent or gross negligence or is based on loss of life, bodily injury or damage to health. The foregoing provisions do not imply any change in the burden of proof to your detriment. You shall indemnify Siemens against existing or future claims of third parties in this connection except where Siemens is mandatorily liable.
By using the application examples you acknowledge that Siemens cannot be held liable for any damage beyond the liability provisions described.

Other information

Siemens reserves the right to make changes to the application examples at any time without notice. In case of discrepancies between the suggestions in the application examples and other Siemens publications such as catalogs, the content of the other documentation shall have precedence.
The Siemens terms of use (https://support.industry.siemens.com) shall also apply.

Security information

Siemens provides products and solutions with industrial security functions that support the secure operation of plants, systems, machines and networks.
In order to protect plants, systems, machines and networks against cyber threats, it is necessary to implement – and continuously maintain – a holistic, state-of-the-art industrial security concept. Siemens' products and solutions constitute one element of such a concept.
Customers are responsible for preventing unauthorized access to their plants, systems, machines and networks. Such systems, machines and components should only be connected to an enterprise network or the internet if and to the extent such a connection is necessary and only when appropriate security measures (e.g. firewalls and/or network segmentation) are in place.
For additional information on industrial security measures that may be implemented, please visit https://www.siemens.com/industrialsecurity.
Siemens' products and solutions undergo continuous development to make them more secure. Siemens strongly recommends that product updates are applied as soon as they are available and that the latest product versions are used. Use of product versions that are no longer supported, and failure to apply the latest updates may increase customer's exposure to cyber threats.
To stay informed about product updates, subscribe to the Siemens Industrial Security RSS Feed under https://www.siemens.com/cert.

Table of contents

Legal information **2**

1 Library Overview **4**

1.1 Different user scenarios 5
1.1.1 OMAC Packaging Working Group 5
1.1.2 OMAC PackML Guidelines 6
1.1.3 Unit Mode and State Manager 6
1.2 Hardware and software requirements 13
1.3 Library resources 13

2 Blocks of the Library **14**

2.1 List of the blocks 14
2.2 Explanation of the blocks 15
2.2.1 FB LPMLV2022_UnitModeStateManager (FB 30110) 15
2.2.2 FB LPMLV2022_UnitModeStateManagerBool (FB 30111) 19
2.2.3 FB LPMLV2022_UnitModeStateTimes (FB 30112) 23
2.2.4 FB LPMLV2022_Stacklight (FB 30113) 25
2.2.5 FC LPMLV2022_ConfigureEnabledModesCfg (FC 30110) 28
2.2.6 FC LPMLV2022_ConfigureDisabledStatesCfg (FC 30111) 31
2.2.7 FC LPMLV2022_ConfigureModeTransitionCfg (FC 30112) 33
2.2.8 FC LPMLV2022_ConfigureHoldCmdCfg (FC 30113) 35
2.2.9 FC LPMLV2022_ConfigureCompleteCmdCfg (FC 30114) 37
2.2.10 PLC data types 38
2.2.11 PLC tags 55

3 Working with the Library **61**

3.1 Integrating the library into STEP 7 61
3.2 Integrating the library blocks into STEP 7 62

4 Notes and Support **63**

5 Appendix **64**

5.1 Service and support 64
5.2 Industry Mall 65
5.3 Application support 65
5.4 Links and literature 65
5.5 Change documentation 66

1 Library Overview

What you get

This document describes the LPMLV2022 block library based on ISA TR88.00.02-2022. The block library provides you with the tested code with clearly defined interfaces. They can be used as a basis for your task to be implemented.

A key concern of the document is to describe

- all blocks of the block library
- the functionality implemented through these blocks.

Furthermore, this documentation shows possible fields of application and helps you integrate the library into your STEP 7 project using step-by-step instructions.

Scope of application

- STEP 7 Basic V15.1
- STEP 7 Professional V15.1
- S7-1200 CPU as of firmware 4.2
- S7-1500 CPU as of firmware 2.6

1.1 Different user scenarios

Possible application for the LPMLV2022 library

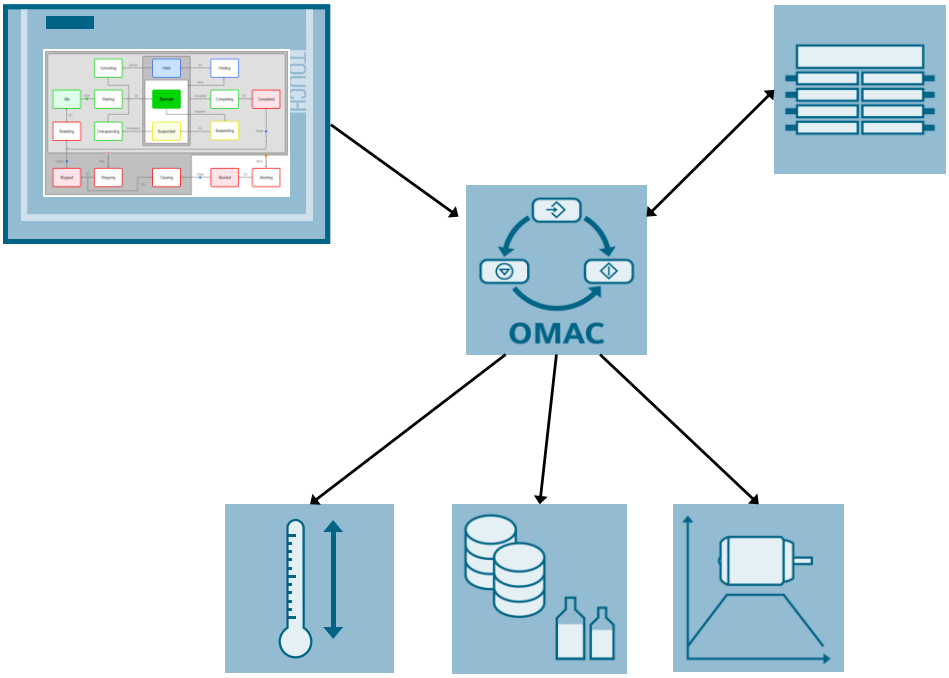
The following section shows a scenario for a possible application of the LPMLV2022 library:

Scenario

A production machine consists of different machine modes (e.g. manual or production mode) and states (e.g. stopped or aborting). The machine state can be controlled with commands (e.g. start or stop).

The library LPMLV2022 provides a standardized mode and state manager to control a machine. User defined code can be filled in an easy to use template structure. In addition, a machine HMI or an upper level system (MES) can be connected by using the standardized machine interface.

Figure 1-1: User scenario



1.1.1 OMAC Packaging Working Group

Structure, contents and aim of the OMAC Packaging Working Group

OMAC (The Organization for Machine Automation and Control) is the organization for automation and manufacturing professionals that is dedicated to supporting the machine automation and operation needs of manufacturing.

The OMAC Packaging Working Group was formed as an initiative from big international end users. In the working group, end users, machine manufacturers (OEMs) and controller manufacturers discuss standards for the automation of production machines in order to reduce the range of variants for different products, technologies and applications.

The objective is to achieve significant improvements regarding the following points:

- Delivery time
- Commissioning time
- Machine dimensions
- Machine performance
- Integration capability
- Format change time
- Flexibility
- Machine modularity
- Machine downtime

Within the OMAC Packaging Working Group, the PackML working group is involved with the definition of guidelines and standards to achieve a standard automation software structure.

NOTE Knowledge about the contents of the basic OMAC documents is an advantage when it comes to understanding the solutions described in this documentation.

See also

OMAC website (https://www.omac.org)

1.1.2 OMAC PackML Guidelines

The major part of the guidelines describes the OMAC mode management (unit mode manager and state machine), see *ISA Technical Report TR88.00.02 Machine and Unit States*.

In addition the PackML pack tags are listed which are used as standardized variable structures (pack tags) for the cross-machine coupling between machine controllers and to higher-level HMI, MES or Enterprise systems, see also *ISA Technical Report TR88.00.02 Machine and Unit States*.

There exist the following variable structures for the pack tags:

- Command tags, to control and parameterize the machine
- Status tags, to provide information about the machine state
- Administration tags, to provide information about the machine efficiency (OEE data) and machine diagnostics

1.1.3 Unit Mode and State Manager

General information

The LPMLV2022 library contains a function block for the unit mode and state management according to PackML V2022.

- Unit modes *Manual*, *Maintenance*, *Production* and *user-defined modes*

- Uniform states within a unit mode
  Defined states, such as *Stopped*, *Starting*, *Execute*, *Aborting*, etc. can be used to handle the machine states within an operating mode. Users can individually remove states that are not used in compliance with the OMAC guidelines.

The machine functionality to be executed in the particular modes and states must be programmed by the user for the specific application.

Modes and states according to PackML V2022

The *Production*, *Maintenance*, *Manual* and the *user-defined modes* with their associated states defined by PackML V2022 are listed in this section. The state machines of the *Manual*, *Maintenance* and the *user-defined modes* are typically a subset of the state machine of the *Production* mode. Which states are used in the individual modes is not standardized and users can define them as required. The state model for the *Production* mode should be considered as the maximum quantity structure, which can be reduced, but should not be increased. This means that the state machine of the *Production* mode is always used and for smaller quantity structures, individual states are directly run-through or skipped.

Figure 1-2: Example of a state machine for the Production mode

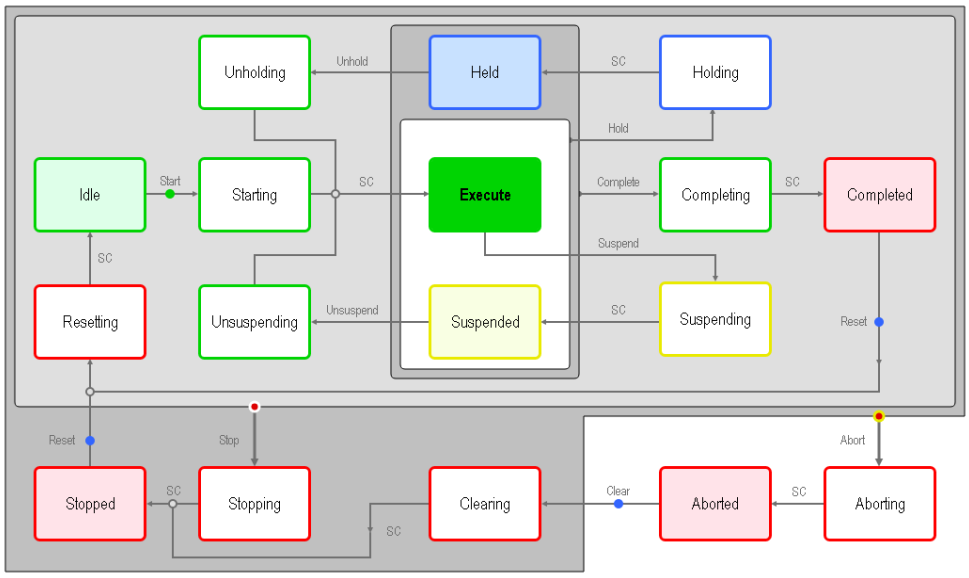
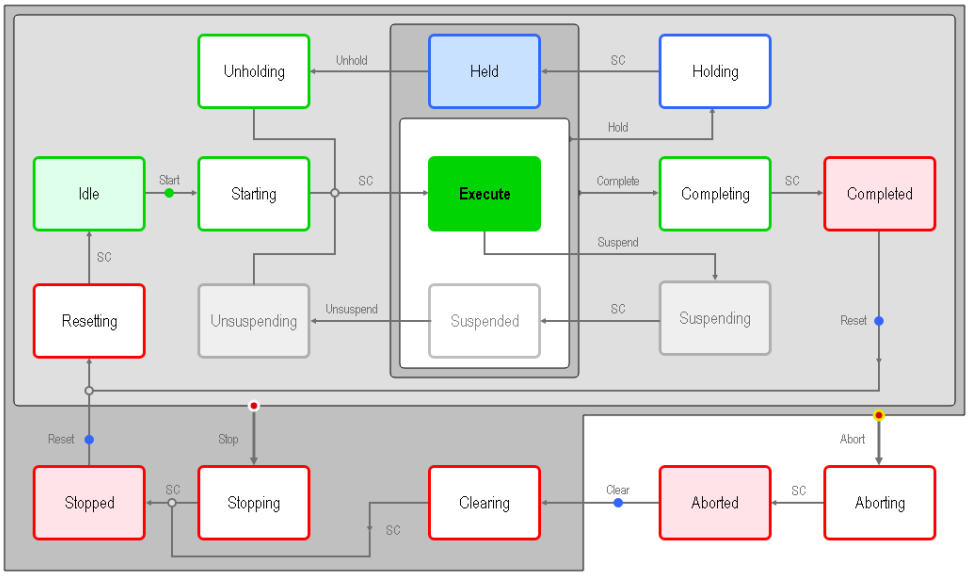


Figure 1-3: Example of a state machine for the Maintenance mode



Unit modes

Table 1-1: Description of the possible unit modes

Number	Unit Mode	Description
0	Invalid	Not a valid unit mode.
1	Production	This represents the mode which is utilized for routine production. The machine executes relevant logic in response to commands which are either entered directly by the operator or issued by another supervisory system.
2	Maintenance	This mode may allow suitably authorized personnel the ability to run an individual machine independent of other machines in a production line. This mode would typically be used for fault finding, machine trials or testing operational improvements. This mode would also allow the speed of the machine to be adjusted (where this feature is available).
3	Manual	This provides direct control of individual machine modules. This feature is available depending upon the mechanical constraints of the mechanisms being exercised. This feature may be used for the commissioning of individual drives, verifying the operation of synchronized drives, testing the drive as a result of modifying parameters etc.
04..31	UserMode01…UserMode28	The requirements for user-defined unit modes differ depending on the machine and application. A typical user-defined unit mode is, for example, a cleaning mode.

States

Table 1-2: Description of the possible states

Number	State	Description
0	Undefined	Not a valid state.
1	Clearing	**State Type: Acting** Initiated by a *Clear* command to clear faults that may have occurred and are present in the *Aborted* state before proceeding to a *Stopped* state.
2	Stopped	**State Type: Wait** The machine is powered and stationary after completing the *Stopping* state. All communications with other systems are functioning (if applicable). A *Reset* command will cause a transition from *Stopped* to the *Resetting* state.
3	Starting	**State Type: Acting** The machine completes the steps needed to start. This state is entered as a result of a *Start* command (local or remote). When *Starting* completes, the machine will transition to the *Execute* state.
4	Idle	**State Type: Wait** This is the state that indicates that *Resetting* is complete. The machine will maintain the conditions that were achieved during the *Resetting* state, and perform operations required when the machine is in *Idle*.
5	Suspended	**State Type: Wait** Refer to *Suspending* for when this state is used. In this state the machine does not produce product. It will either stop running or continue to cycle without producing until external process conditions return to normal, at which time, the *Suspended* state will transition to the *Unsuspending* state, typically without any operator intervention.
6	Execute	**State Type: Acting** Once the machine is processing materials it is in the *Execute* state until a transition command is received. Different machine modes will result in specific types of *Execute* activities. For example, if the machine is in the “Production” mode, the *Execute* will result in products being produced, while perhaps in a user-defined “Clean Out” mode the *Execute* state would result in the action of cleaning the machine.
7	Stopping	**State Type: Acting** This state is entered in response to a *Stop* command. While in this state the machine executes the logic that brings it to a controlled stop as reflected by the *Stopped* state. Normal *Starting* of the machine cannot be initiated unless *Resetting* has taken place.
8	Aborting	**State Type: Acting** The *Aborting* state can be entered at any time in response to the *Abort* command, typically triggered by the occurrence of a machine event that warrants an aborting action. The aborting logic will bring the machine to a rapid safe stop.
9	Aborted	**State Type: Wait** The machine maintains status information relevant to the abort condition. The machine can only exit the *Aborted* state after an explicit *Clear* command, subsequent to manual intervention to correct and reset the detected machine faults.
10	Holding	**State Type: Acting** This state is used when **internal** (inside this unit/machine and not from another machine on the production line) machine conditions do not allow the machine to continue producing, that is, the machine leaves the *Execute* or *Suspended* states due to internal conditions. This is typically used for routine machine conditions that require minor operator servicing to continue production. This state can be initiated automatically or by an operator and can be easily recovered from. An example of this would be a machine that

Number	State	Description
		requires an operator to periodically refill a glue dispenser or carton magazine and due to the machine design, these operations cannot be performed while the machine is running. Since these types of tasks are normal production operations, it is not desirable to go through aborting or stopping sequences, and because these functions are integral to the machine they are not considered to be "**external**". While in the *Holding* state, the machine is typically brought to a controlled stop and then transitions to *Held* upon state complete. To be able to restart production correctly after the *Held* state, all relevant process set points and return status of the procedures at the time of receiving the *Hold* command must be saved in the machine controller when executing the *Holding* procedure.
11	Held	**State Type: Wait** Refer to *Holding* for when this state is used. In this state the machine does not produce product. It will either stop running or continue to dry cycle. A transition to the *Unholding* state will occur when **internal** machine conditions change or an *Unhold* command is initiated by an operator.
12	Unholding	**State Type: Acting** Refer to *Holding* for when this state is used. A machine will typically enter into *Unholding* automatically when **internal** conditions, material levels, for example, return to an acceptable level. If an operator is required to perform minor servicing to replenish materials or make adjustments, then the *Unhold* command may be initiated by the operator.
13	Suspending	**State Type: Acting** This state is used when **external** (outside this unit/machine but usually on the same integrated production line) process conditions do not allow the machine to continue producing, that is, the machine leaves *Execute* due to upstream or downstream conditions on the line. This is typically due to a Blocked or Starved event. This condition may be detected by a local machine sensor or based on a supervisory system external command. While in the *Suspending* state, the machine is typically brought to a controlled stop and then transitions to *Suspended* upon state complete. To be able to restart production correctly after the *Suspended* state, all relevant process set points and return status of the procedures at the time of receiving the *Suspend* command must be saved in the machine controller when executing the *Suspending* procedure.
14	Unsuspending	**State Type: Acting** Refer to *Suspending* for when this state is used. This state is a result of process conditions returning to normal. The *Unsuspending* state initiates any required actions or sequences necessary to transition the machine from *Suspended* back to *Execute*. To be able to restart production correctly after the *Suspended* state, all relevant process set points and return status of the procedures at the time of receiving the *Suspend* command must be saved in the machine controller when executing the *Suspending* procedure.
15	Resetting	**State Type: Acting** This state is the result of a *Reset* command from the *Stopped* or *Completed* state. Faults and stop causes are reset. *Resetting* will typically cause safety devices to be energized and place the machine in the *Idle* state where it will wait for a *Start* command. No hazardous motion should happen in this state.
16	Completing	**State Type: Acting** This state is the result of a *Complete* command from the *Execute*, *Held* or *Suspended* states. The *Complete* command may be internally generated, such as reaching the end of a predefined production count where normal operation has run to completion, or externally generated, such as by a supervisory system. The *Completing* state is often used to end a production run and summarize production data.
17	Completed	**State Type: Wait** The machine has finished the *Completing* state and is now waiting for a *Reset* command before transitioning to the *Resetting* state.

Control commands

Table 1-3: Possible control commands

Number	Control command
0	Undefined
1	Reset
2	Start
3	Stop
4	Hold
5	Unhold
6	Suspend
7	Unsuspend
8	Abort
9	Clear
10	Complete

Unit mode transitions

Permitted change of the unit mode

Changing the unit mode is permitted in any state and can be configured in the configuration (*configuration.ModeTransitionCfg).* Typically, the following wait states are used for unit mode changes: *Stopped, Idle, Aborted*. The unit mode change is only possible if the state also exists in the requested unit mode.

State transitions

Example of reading the table

A change is made from the *Idle* state to the *Starting* state with the *Start* command. The further to the right that a command is located in the table, the higher its priority for the state change.

State change with priority assignment

	State Commands										State Complete
Current State	**Start**	**Reset[1]**	**Hold**	**Unhold**	**Suspend**	**Unsuspend**	**Complete**	**Clear[1]**	**Stop[2]**	**Abort[2]**	
IDLE	STARTING								STOPPING	ABORTING	
STARTING									STOPPING	ABORTING	EXECUTE
EXECUTE			HOLDING		SUSPENDING		COMPLETING		STOPPING	ABORTING	
COMPLETING									STOPPING	ABORTING	COMPLETED
COMPLETED		RESETTING							STOPPING	ABORTING	
RESETTING									STOPPING	ABORTING	IDLE
HOLDING									STOPPING	ABORTING	HELD
HELD				UNHOLDING			COMPLETING		STOPPING	ABORTING	
UNHOLDING									STOPPING	ABORTING	EXECUTE
SUSPENDING									STOPPING	ABORTING	SUSPENDED
SUSPENDED			HOLDING			UNSUSPENDING	COMPLETING		STOPPING	ABORTING	
UNSUSPENDING									STOPPING	ABORTING	EXECUTE
STOPPING										ABORTING	STOPPED
STOPPED		RESETTING								ABORTING	
ABORTING											ABORTED
ABORTED								CLEARING			
CLEARING										ABORTING	STOPPED

[1] It is common practice for Clear and Reset to be initiated using the same physical operator interface device.

[2] It is common practice in packaging (but not process) applications to permit use of Stop and Abort commands while in the IDLE, COMPLETED, STOPPED, and RESETTING states.

1.2 Hardware and software requirements

Requirements for this library

To be able to use the functionality of the library described in this document, the following hardware and software requirements must be met:

Hardware

Table 1-4: Hardware components

No.	Component	Article number	Alternative
1.	CPU 1513-1 PN	6ES7513-1AL02-0AB0	Other S7-1500 CPU with FW V2.6
2.	Or CPU 1215C	6ES7215-1AG40-0XB0	Other S7-1200 CPU with FW V4.2

Software

Table 1-5: Software components

No.	Component	Article number	Quantity
1.	STEP 7 Professional V15.1	6ES7822-1..05-..	1
2.	Or STEP 7 Basic V15.1	6ES7822-0A.05-..	1

1.3 Library resources

What will you find in this section?

The following section gives you an overview of the size of the blocks of the LPMLV2022 library in the main, load and retain memory.

Size of the individual blocks [1]

Table 1-6: Size of the blocks

Block	Symbol	Size in main memory [Kbytes]	Size in load memory [Kbytes]	Size in retain memory [Kbytes]
FB 30110	LPMLV2022_UnitModeStateManager	8.3	134	-
FB 30111	LPMLV2022_UnitModeStateManagerBool	1.9	42	-
FB 30112	LPMLV2022_UnitModeStateTimes	1.3	25	-
FB 30113	LPMLV2022_Stacklight	1.3	17	-
FC 30110	LPMLV2022_ConfigureEnabledModesCfg	0.4	9.8	-
FC 30111	LPMLV2022_ConfigureDisabledStatesCfg	0.2	7.4	-
FC 30112	LPMLV2022_ConfigureModeTransitionCfg	0.3	8.0	-
FC 30113	LPMLV2022_ConfigureHoldCmdCfg	0.2	6.7	-
FC 30114	LPMLV2022_ConfigureCompleteCmdCfg	0.1	5.6	-
DB 30110	instLPMLV2022_UnitModeStateManager	0.9	5.9	-
DB 30111	instLPMLV2022_UnitModeStateManagerBool	1.1	7.7	-
DB 30112	instLPMLV2022_UnitModeStateTimes	1.6	3.9	0.7
DB 30113	instLPMLV2022_Stacklight	0.1	1.8	-

[1] Instance data blocks (prefix *instLPMLV2022_*) are not delivered with the library. They will be generated automatically with the call of a function block.

2 Blocks of the Library

What will you find in this section?

This chapter lists and explains all blocks of the LPMLV2022 library. Before that, however, you are informed of the blocks that are essentially involved in the implementation of the functionality.

2.1 List of the blocks

The following table lists all blocks of the LPMLV2022 library.

Table 2-1: List of the blocks

Block	Symbol	Classification
FB 30110	LPMLV2022_UnitModeStateManager	In-house development
FB 30111	LPMLV2022_UnitModeStateManagerBool	In-house development
FB 30112	LPMLV2022_UnitModeStateTimes	In-house development
FB 30113	LPMLV2022_Stacklight	In-house development
FC 30110	LPMLV2022_ConfigureEnabledModesCfg	In-house development
FC 30111	LPMLV2022_ConfigureDisabledStatesCfg	In-house development
FC 30112	LPMLV2022_ConfigureModeTransitionCfg	In-house development
FC 30113	LPMLV2022_ConfigureHoldCmdCfg	In-house development
FC 30114	LPMLV2022_ConfigureCompleteCmdCfg	In-house development

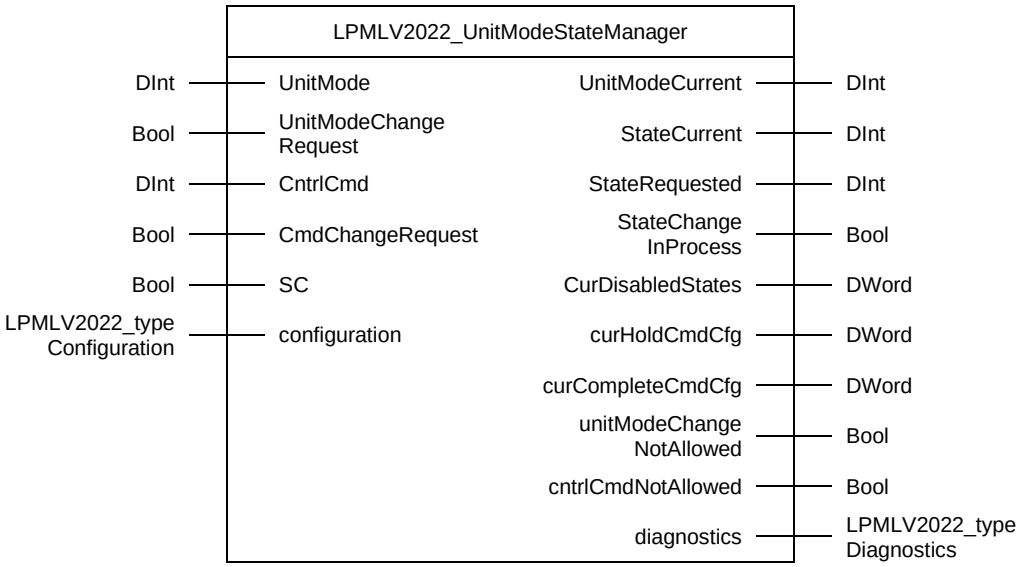
2.2 Explanation of the blocks

The following table explains all blocks of the LPMLV2022 library.

2.2.1 FB LPMLV2022_UnitModeStateManager (FB 30110)

Figure

Figure 2-1: LPMLV2022_UnitModeStateManager

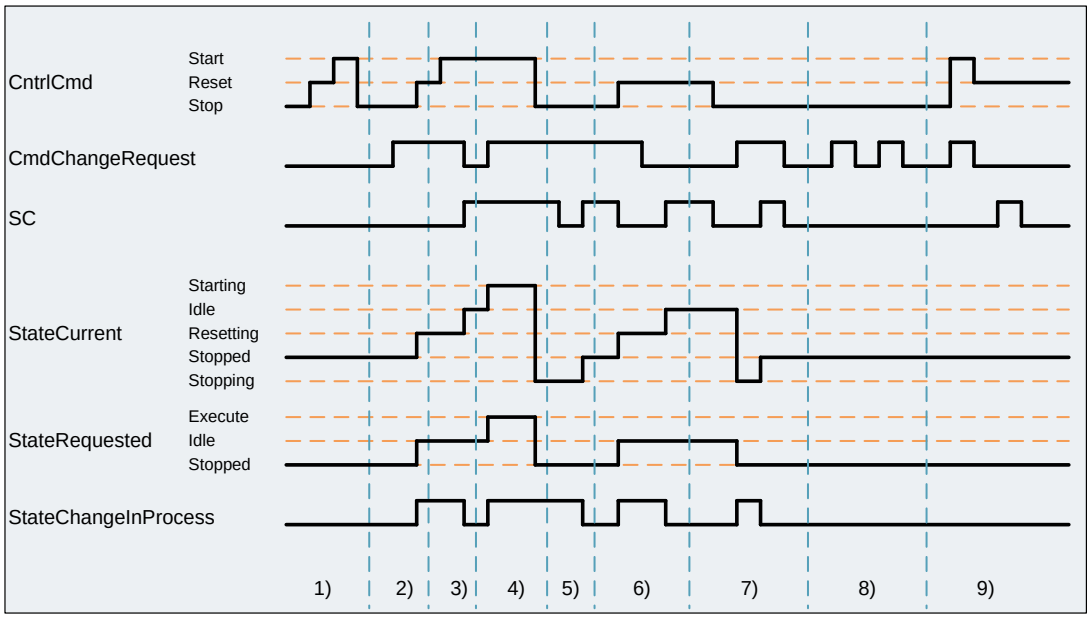


Principle of operation

The function block *LPMLV2022_UnitModeStateManager* is the main part of the block library LPMLV2022 and manages the transitions between the unit modes and states according to the OMAC PackML standard.

NOTE If a boolean interface is preferred, then the block *LPMLV2022_UnitModeStateManagerBool* should be used instead of this block. This is particularly the case, if the block is called in ladder logic.

Function characteristics



1. If *CmdChangeRequest* is not set, every C*ntrlCmd* is ignored.
2. If C*mdChangeRequest* is set to TRUE and a valid C*ntrlCmd* (in this case Reset) is set, the S*tateChangeInProcess* bit is set and the *StateRequested* (in this case Idle) value is set (only wait states possible). A valid C*ntrlCmd* is always necessary if the current state is a wait state.

   If the current state is an acting state (here Resetting), a rising edge at input *SC* is necessary to leave the state.
   If C*mdChangeRequest* is FALSE, the C*ntrlCmd* (here Start) has already been set in the acting state (here Resetting) and the wait state (Idle) is reached through a rising edge at the input *SC*, the next acting state will not be reached automatically. For the change in the next wait state (here Execute) is a rising edge at C*mdChangeRequest* needed.
3. The S*tateChangeInProcess* bit remains set until *StateCurrent* gets the same value as S*tateRequested*. The S*tateRequested* changes from Execute to Stopped, because in Starting state a valid Stop command was set.
4. The *SC* input is not level sensitive. If the *SC* input is already set when reaching the acting state, the *Unit Mode and State Manager* stays in the acting state as long as a rising edge at *SC* input is detected.
5. The SC command is not related to the C*mdChangeRequest* input. Even if C*mdChangeRequest* is FALSE, a state change can happen from acting to a wait state.
6. A new CntrlCmd can also be set if C*mdChangeRequest* is FALSE. If C*mdChangeRequest* changes to TRUE and the control command is valid in this wait state, the state will be changed.
7. If no valid control command for the current state is written to input, edges at *CmdChangeRequest* are ignored. To change from a wait to an acting state, *CmdChangeRequest* has to be TRUE and the according *CntrlCmd* has to be written to input *CntrlCmd*.
8. If no acting state is active the rising edge at the *SC* input is ignored.
9. If an invalid *CntrlCmd* is written to input *CntrlCmd* and *CmdChangeRequest* is set to TRUE, the control command is ignored and an entry is written to the diagnostics buffer of the FB.

Input parameters

Table 2-2: *LPMLV2022_UnitModeStateManager* input parameters

Parameter	Data type	Comment
UnitMode	DInt	Requested unit mode (default: LPMLV2022_MODE_INVALID) For valid unit modes see Table 1-1: Description of the possible unit modes.
UnitModeChange Request	Bool	TRUE: Request unit mode (default: FALSE)
CntrlCmd	DInt	Request control command (default: LPMLV2022_CMD_UNDEFINED) For valid control commands see Table 1-3: Possible control commands
CmdChangeRequest	Bool	TRUE: Enable change into requested state (default: FALSE)
SC	Bool	State change from FALSE to TRUE (rising edge) triggers state complete signal (default: FALSE)
configuration	LPMLV2022_typeConfiguration	FB configuration (is taken into account in first call after STOP/RUN)

Output parameters

Table 2-3: *LPMLV2022_UnitModeStateManager* output parameters

Parameter	Data type	Comment
UnitModeCurrent	DInt	Current unit mode For valid unit modes see Table 1-1: Description of the possible unit modes
StateCurrent	DInt	Current state For valid states see Table 1-2: Description of the possible states
StateRequested	DInt	Requested state
StateChange InProcess	Bool	State change in process
CurDisabledStates	DWord	Disabled states in current unit mode
curHoldCmdCfg	DWord	Bit locations within the DWORD represent State numbers. A value of 1 in a bit location indicates that the Hold control command is taken into account in the corresponding state
curCompleteCmdCfg	DWord	Bit locations within the DWORD represent State numbers. A value of 1 in a bit location indicates that the Complete control command is taken into account in the corresponding state
unitModeChange NotAllowed	Bool	TRUE: Requested unit mode change is not allowed (output is reset with the next successful unit mode change or if input 'UnitMode' is set to 0 or if input 'UnitModeChangeRequest' is set to FALSE)
cntrlCmdNotAllowed	Bool	TRUE: Control command is not allowed (output is reset with the next successful CntrlCmd or if input 'CntrlCmd' is set to 0 or if input 'CmdChangeRequest' is set to FALSE)
diagnostics	LPMLV2022_typeDiagnostics	Diagnostics information of FB

Status and error displays

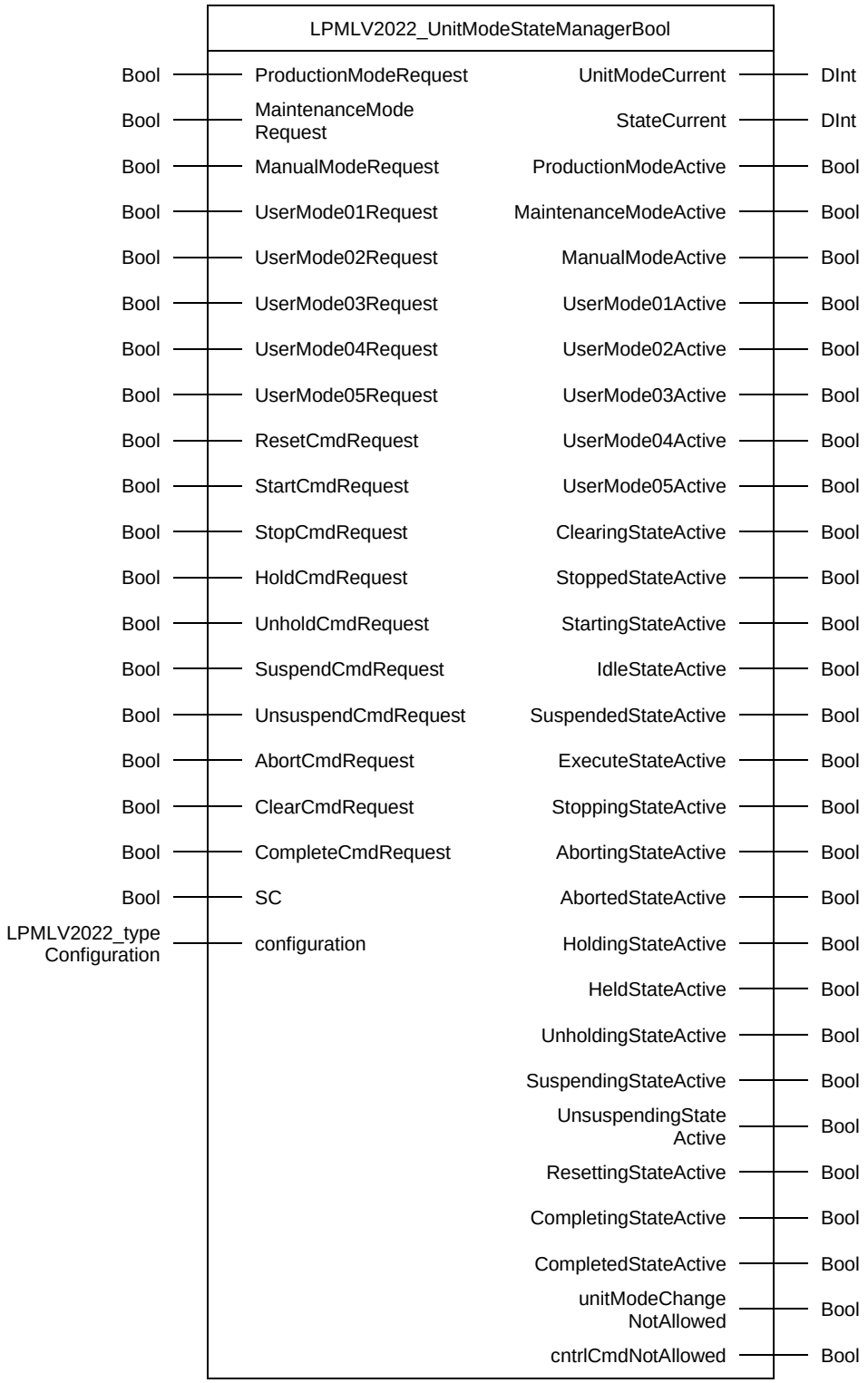
Table 2-4

Status	Meaning	Remedy / notes
16#00	MSG_NO_MESSAGE	Initial value
16#01	MSG_MODE_CHANGED_SUCCESSFULLY	Unit mode changed successfully
16#02	MSG_STATE_CHANGED_SUCCESSFULLY	State changed successfully
16#03	MSG_MODE_ALREADY_ACTIVE	Requested unit mode already active
16#80	MSG_MODE_NOT_DEFINED	Unit mode not defined
16#81	MSG_CMD_NOT_DEFINED	Control command not defined
16#82	MSG_REQ_MODE_NOT_CONFIGURED	Requested unit mode not configured - check 'configuration.EnabledModesCfg'
16#83	MSG_MODE_TRANSITION_NOT_ALLOWED	Unit mode transition in this state not allowed - check 'configuration.ModeTransitionCfg[]' of the current mode and the requested mode. The corresponding state bit must be set for both modes
16#84	MSG_CMD_NOT_ALLOWED	Control command in this state not allowed
16#85	MSG_SC_NOT_ALLOWED	SC in this state not allowed
16#86	MSG_STATE_CONFIG_FORCED	State configuration forced to OMAC standard (corrected configuration see FB output CurDisabledStates) Requirements regarding the state configuration according to OMAC: some states are mandatory (Stopped, Idle, Execute and Aborted) if Holding or Unholding is enabled also Held must be present if Suspending or Unsuspending is enabled also Suspended must be present if Completing is enabled also Completed must be present
16#87	MSG_MODE_TRANSITION_NOT_POSSIBLE	Unit mode transition in this state not possible, because the current state is not available in the requested mode - check 'configuration.DisabledStatesCfg[]' of the requested mode
16#88	MSG_SC_OVERRIDDEN_BY_CMD_HOLD	SC overridden by control command Hold

2.2.2 FB LPMLV2022_UnitModeStateManagerBool (FB 30111)

Figure

Figure 2-2: LPMLV2022_UnitModeStateManagerBool



Principle of operation

The function block *LPMLV2022_UnitModeStateManagerBool* is a wrapping function block with a boolean interface to ease the usage in ladder programming. The *LPMLV2022_UnitModeStateManager* function block is called internally.

NOTE The number of user-defined unit modes can be extended (up to 28). Therefore, the corresponding inputs and outputs have to be added at the *LPMLV2022_UnitModeStateManagerBool* block. Also the prepared corresponding lines of code within the source code of this block must be uncommented. No modifications are necessary to be made at the *LPMLV2022_UnitModeStateManager* block. Typically, the value of the constant *LPMLV2022_MODES_UPPER_LIM* must be extended (relevant for example for the block *LPMLV2022_UnitModeStateTimes*).

Input parameters

Table 2-5: *LPMLV2022_UnitModeStateManagerBool* input parameters

Parameter	Data type	Comment
ProductionMode Request	Bool	Rising edge: Request change to unit mode Production
MaintenanceMode Request	Bool	Rising edge: Request change to unit mode Maintenance
ManualModeRequest	Bool	Rising edge: Request change to unit mode Manual
UserMode01Request	Bool	Rising edge: Request change to user-defined unit mode 01
UserMode02Request	Bool	Rising edge: Request change to user-defined unit mode 02
UserMode03Request	Bool	Rising edge: Request change to user-defined unit mode 03
UserMode04Request	Bool	Rising edge: Request change to user-defined unit mode 04
UserMode05Request	Bool	Rising edge: Request change to user-defined unit mode 05
ResetCmdRequest	Bool	Rising edge: Request control command Reset
StartCmdRequest	Bool	Rising edge: Request control command Start
StopCmdRequest	Bool	Rising edge: Request control command Stop
HoldCmdRequest	Bool	Rising edge: Request control command Hold
UnholdCmdRequest	Bool	Rising edge: Request control command Unhold
SuspendCmdRequest	Bool	Rising edge: Request control command Suspend
UnsuspendCmd Request	Bool	Rising edge: Request control command Unsuspend
AbortCmdRequest	Bool	Rising edge: Request control command Abort
ClearCmdRequest	Bool	Rising edge: Request control command Clear
CompleteCmdRequest	Bool	Rising edge: Request control command Complete
SC	Bool	State change from FALSE to TRUE (rising edge) triggers state complete signal
configuration	LPMLV2022_typeConfiguration	FB configuration (is taken into account in first call after STOP/RUN)

Output parameters

Table 2-6: *LPMLV2022_UnitModeStateManagerBool* output parameters

Parameter	Data type	Comment
UnitModeCurrent	DInt	Current unit mode For valid unit modes see Table 1-1: Description of the possible unit modes
StateCurrent	DInt	Current state For valid states see Table 1-2: Description of the possible states
ProductionModeActive	Bool	TRUE: Unit mode Production is currently active
MaintenanceMode Active	Bool	TRUE: Unit mode Maintenance is currently active
ManualModeActive	Bool	TRUE: Unit mode Manual is currently active
UserMode01Active	Bool	TRUE: User-defined unit mode 01 is currently active
UserMode02Active	Bool	TRUE: User-defined unit mode 02 is currently active
UserMode03Active	Bool	TRUE: User-defined unit mode 03 is currently active
UserMode04Active	Bool	TRUE: User-defined unit mode 04 is currently active
UserMode05Active	Bool	TRUE: User-defined unit mode 05 is currently active
ClearingStateActive	Bool	TRUE: State Clearing is currently active
StoppedStateActive	Bool	TRUE: State Stopped is currently active
StartingStateActive	Bool	TRUE: State Starting is currently active
IdleStateActive	Bool	TRUE: State Idle is currently active
SuspendedStateActive	Bool	TRUE: State Suspended is currently active
ExecuteStateActive	Bool	TRUE: State Execute is currently active
StoppingStateActive	Bool	TRUE: State Stopping is currently active
AbortingStateActive	Bool	TRUE: State Aborting is currently active
AbortedStateActive	Bool	TRUE: State Aborted is currently active
HoldingStateActive	Bool	TRUE: State Holding is currently active
HeldStateActive	Bool	TRUE: State Held is currently active
UnholdingStateActive	Bool	TRUE: State Unholding is currently active
SuspendingState Active	Bool	TRUE: State Suspending is currently active
UnsuspendingState Active	Bool	TRUE: State Unsuspending is currently active
ResettingStateActive	Bool	TRUE: State Resetting is currently active
CompletingStateActive	Bool	TRUE: State Completing is currently active
CompletedStateActive	Bool	TRUE: State Completed is currently active
unitModeChange NotAllowed	Bool	TRUE: Requested unit mode change is not allowed (output is reset with the next successful unit mode change or if all corresponding request inputs are set to FALSE)
cntrlCmdNotAllowed	Bool	TRUE: Control command is not allowed (output is reset with the next successful control command or if all corresponding request inputs are set to FALSE)

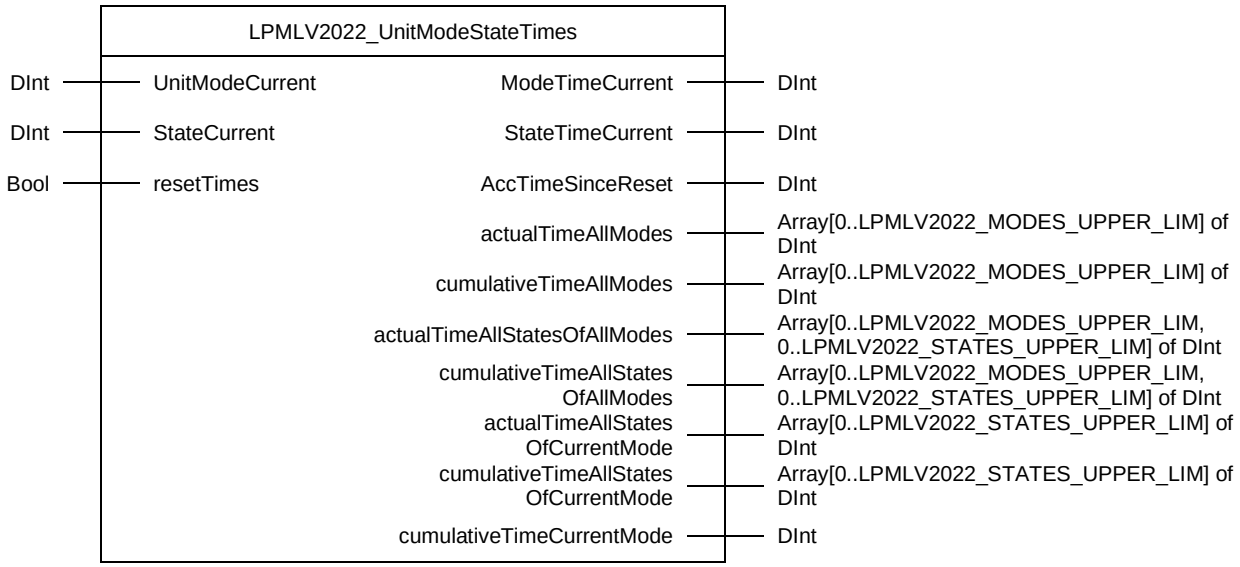
Status and error displays

See *LPMLV2022_UnitModeStateManager*.

2.2.3 FB LPMLV2022_UnitModeStateTimes (FB 30112)

Figure

Figure 2-3: LPMLV2022_UnitModeStateTimes



Principle of operation

The function block *LPMLV2022_UnitModeStateTimes* is optional and counts the time in seconds for every state in every unit mode.

NOTE The value of the constant LPMLV2022_MODES_UPPER_LIM must correspond to the number of modes which are actually used.

NOTE When compiling this block, 6 warnings are issued. These can be ignored.

Input parameters

Table 2-7: *LPMLV2022_UnitModeStateTimes* input parameters

Parameter	Data type	Comment
UnitModeCurrent	DInt	Current unit mode (default: LPMLV2022_MODE_INVALID)
StateCurrent	DInt	Current state (default: LPMLV2022_STATE_UNDEFINED)
resetTimes	Bool	State change from FALSE to TRUE (rising edge) resets unit mode/state times and cumulative times (default: FALSE)

Output parameters

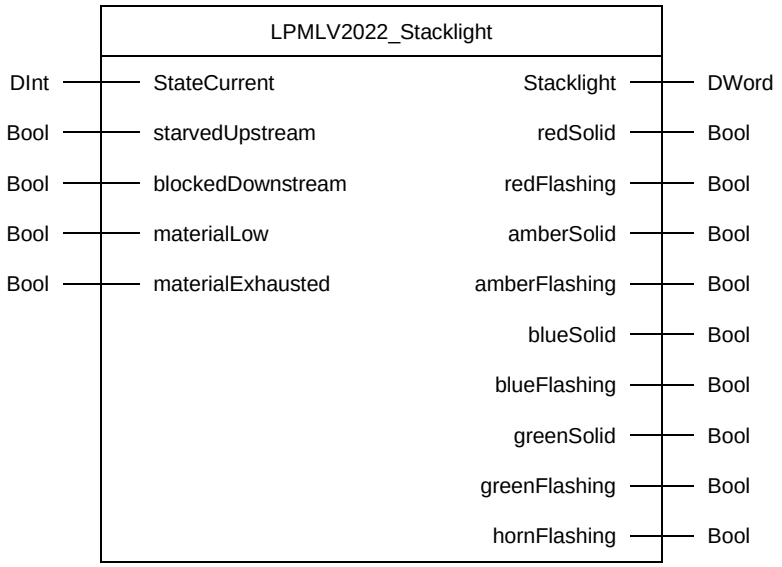
Table 2-8: *LPMLV2022_UnitModeStateTimes* output parameters

Parameter	Data type	Comment
ModeTimeCurrent	DInt	Actual time for current mode in seconds
StateTimeCurrent	DInt	Actual time for current state in seconds
AccTimeSinceReset	DInt	Accumulated time since last reset in seconds
actualTimeAllModes	Array[0..LPMLV2022_MODES_UPPER_LIM] of DInt	Actual time for all modes in seconds (former name: ModeCurrentTime)
cumulativeTimeAllModes	Array[0..LPMLV2022_MODES_UPPER_LIM] of DInt	Cumulative time for all modes in seconds (former name: ModeCumulativeTime)
actualTimeAllStatesOfAllModes	Array[0..LPMLV2022_MODES_UPPER_LIM, 0..LPMLV2022_STATES_UPPER_LIM] of DInt	Actual time for all states in seconds (former name: StateCurrentTime)
cumulativeTimeAllStatesOfAllModes	Array[0..LPMLV2022_MODES_UPPER_LIM, 0..LPMLV2022_STATES_UPPER_LIM] of DInt	Cumulative time for all states in seconds (former name: StateCumulativeTime)
actualTimeAllStatesOfCurrentMode	Array[0..LPMLV2022_STATES_UPPER_LIM] of DInt	Actual time for all states in current mode in seconds
cumulativeTimeAllStatesOfCurrentMode	Array[0..LPMLV2022_STATES_UPPER_LIM] of DInt	Cumulative time for all states in current mode in seconds
cumulativeTimeCurrentMode	DInt	Cumulative time for current mode in seconds

2.2.4 FB LPMLV2022_Stacklight (FB 30113)

Figure

Figure 2-4: LPMLV2022_Stacklight



Principle of operation

The function block *LPMLV2022_Stacklight* can be used to generate the stacklight status information according to ISA TR88.00.02-2022.

Figure 2-5

```
// Example of how the stacklight can be made to correspond to certain machine conditions and PackML States

//                  |Aborting |Clearing | Aborted |Completed| Stopped |Stopping |Resetting|  Idle   |Starting |Execute  | Comple- | Holding |Held (No |Unholding| Suspen- |Suspended|Unsuspen-|
//                  |         |         |         |         |         |         |         |         |         |         |  ting   |         |Product.)|         |  ding   |         |  ding   |
//------------------+---------+---------+---------+---------+---------+---------+---------+---------+---------+---------+---------+---------+---------+---------+---------+---------+---------+--------------------
// Abnormal Stop    |    F    |    F    |    F    |         |         |         |         |         |         |         |         |         |         |         |         |         |         | Red Lamp Flashing
//------------------+---------+---------+---------+---------+---------+---------+---------+---------+---------+---------+---------+---------+---------+---------+---------+---------+---------+--------------------
// Controlled Stop  |         |         |         |    S    |    S    |    S    |    S    |         |         |         |         |         |         |         |         |         |         | Red Lamp Solid
//------------------+---------+---------+---------+---------+---------+---------+---------+---------+---------+---------+---------+---------+---------+---------+---------+---------+---------+--------------------
// Starved Upstream |         |         |         |         |         |         |         |         |         |         |         |         |         |         |    F    |    F    |         | Amber Lamp Flashing
//------------------+---------+---------+---------+---------+---------+---------+---------+---------+---------+---------+---------+---------+---------+---------+---------+---------+---------+--------------------
// Blocked Downstream |       |         |         |         |         |         |         |         |         |         |         |         |         |         |    S    |    S    |         | Amber Lamp Solid
//------------------+---------+---------+---------+---------+---------+---------+---------+---------+---------+---------+---------+---------+---------+---------+---------+---------+---------+--------------------
// Low Material     |    F    |    F    |    F    |    F    |    F    |    F    |    F    |    F    |    F    |    F    |    F    |         |         |    F    |    F    |    F    |    F    | Blue Lamp Flashing
//------------------+---------+---------+---------+---------+---------+---------+---------+---------+---------+---------+---------+---------+---------+---------+---------+---------+---------+--------------------
// Material Exhausted |       |         |         |         |         |         |         |         |         |         |         |    S    |    S    |         |         |         |         | Blue Lamp Solid
//------------------+---------+---------+---------+---------+---------+---------+---------+---------+---------+---------+---------+---------+---------+---------+---------+---------+---------+--------------------
// Ready to Start   |         |         |         |         |         |         |         |    F    |         |         |         |         |         |         |         |         |         | Green Lamp Flashing
//------------------+---------+---------+---------+---------+---------+---------+---------+---------+---------+---------+---------+---------+---------+---------+---------+---------+---------+--------------------
// Running          |         |         |         |         |         |         |         |         |    S    |    S    |    S    |         |         |    S    |         |         |    S    | Green Lamp Solid
//------------------+---------+---------+---------+---------+---------+---------+---------+---------+---------+---------+---------+---------+---------+---------+---------+---------+---------+--------------------
// Starting/Restarting|       |         |         |         |         |         |         |         |    F    |         |         |         |         |    F    |         |         |    F    | Horn
//------------------+---------+---------+---------+---------+---------+---------+---------+---------+---------+---------+---------+---------+---------+---------+---------+---------+---------+--------------------
```

Input parameters

Table 2-9: *LPMLV2022_Stacklight* input parameters

Parameter	Data type	Comment
StateCurrent	DInt	Current state
starvedUpstream	Bool	TRUE: upstream system is not able to supply products
blockedDownstream	Bool	TRUE: downstream system is not able to accept products
materialLow	Bool	TRUE: low material
materialExhausted	Bool	TRUE: material exhausted

Output parameters

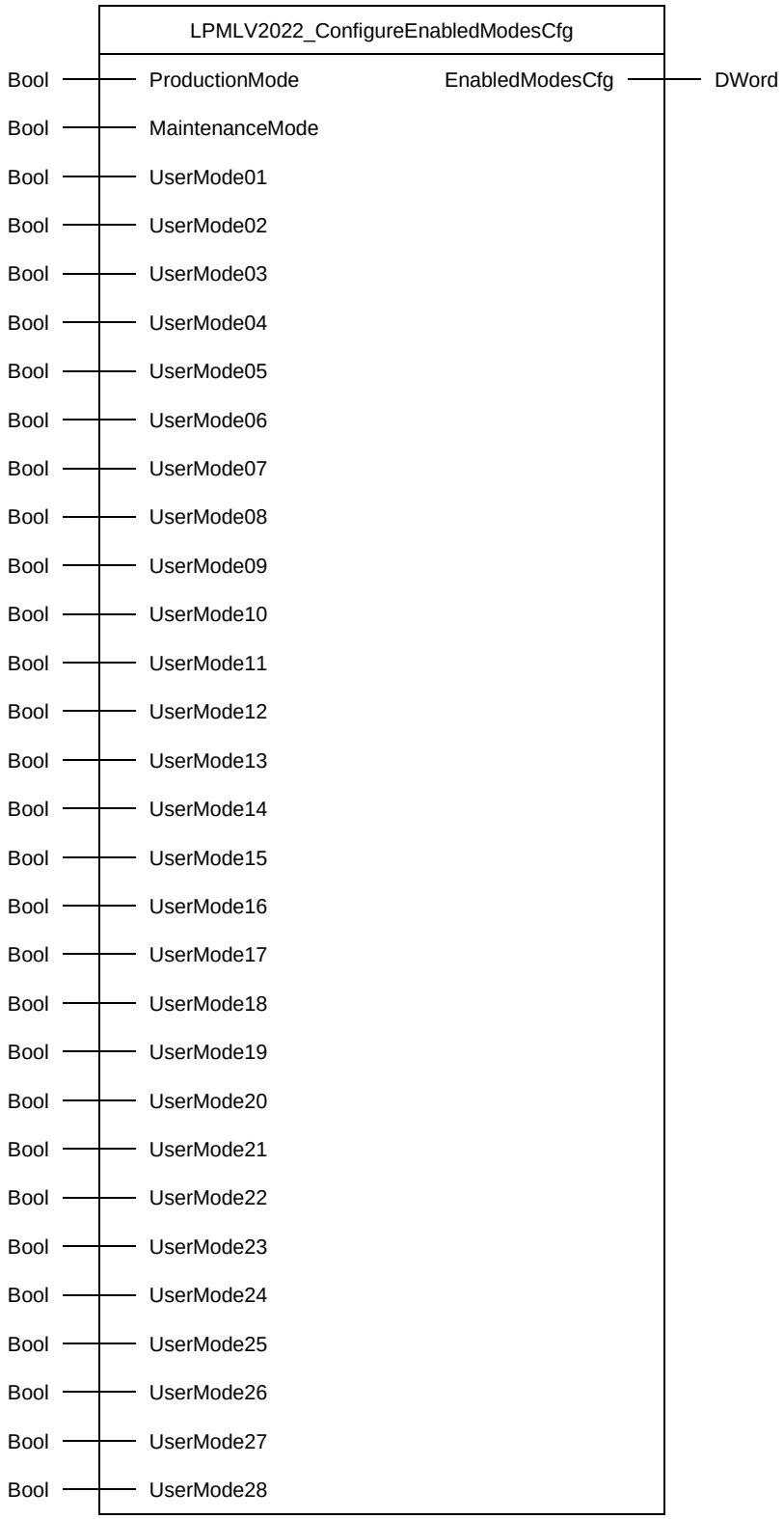
Table 2-10: *LPMLV2022_Stacklight* output parameters

Parameter	Data type	Comment
Stacklight	DWord	Indicator for the current stacklight status -> PackTags
redSolid	Bool	Status of stacklight bit 0
redFlashing	Bool	Status of stacklight bit 1
amberSolid	Bool	Status of stacklight bit 2
amberFlashing	Bool	Status of stacklight bit 3
blueSolid	Bool	Status of stacklight bit 4
blueFlashing	Bool	Status of stacklight bit 5
greenSolid	Bool	Status of stacklight bit 6
greenFlashing	Bool	Status of stacklight bit 7
hornFlashing	Bool	Status of stacklight bit 9

2.2.5 FC LPMLV2022_ConfigureEnabledModesCfg (FC 30110)

Figure

Figure 2-6: LPMLV2022_ConfigureEnabledModesCfg



Principle of operation

This function allows the user to set the unit mode configuration for the FB *LPMLV2022_UnitModeStateManager* and the FB *LPMLV2022_UnitModeStateManagerBool* easily. Of course, it is also possible to set the unit mode configuration directly in the FB *LPMLV2022_UnitModeStateManager* configuration.

With the function, the user has to set the associated inputs for the different unit modes to “TRUE”, e.g. *“MaintenanceMode := TRUE*” for enabling the unit mode *Maintenance*.

To write the unit mode configuration from the function output to the according *“Unit Mode and State Manager”,* the output *EnabledModesCfg* has to be assigned to the configuration of the corresponding FB *LPMLV2022_UnitModeStateManager*.

NOTE

The unit mode *Manual* is mandatory and is therefore always enabled, i.e. no input exists at the FC *LPMLV2022_ConfigureEnabledModesCfg* for mode *Manual*. If the mode *Manual* is nevertheless disabled at the “*Unit Mode and State Manager”*, it will automatically be enabled.

Input parameters

Table 2-11: *LPMLV2022_ConfigureEnabledModesCfg* input parameters

Parameter	Data type	Comment
ProductionMode	Bool	TRUE: Enable unit mode Production
MaintenanceMode	Bool	TRUE: Enable unit mode Maintenance
UserMode01	Bool	TRUE: Enable user-defined unit mode 01
UserMode02	Bool	TRUE: Enable user-defined unit mode 02
UserMode03	Bool	TRUE: Enable user-defined unit mode 03
UserMode04	Bool	TRUE: Enable user-defined unit mode 04
UserMode05	Bool	TRUE: Enable user-defined unit mode 05
UserMode06	Bool	TRUE: Enable user-defined unit mode 06
UserMode07	Bool	TRUE: Enable user-defined unit mode 07
UserMode08	Bool	TRUE: Enable user-defined unit mode 08
UserMode09	Bool	TRUE: Enable user-defined unit mode 09
UserMode10	Bool	TRUE: Enable user-defined unit mode 10
UserMode11	Bool	TRUE: Enable user-defined unit mode 11
UserMode12	Bool	TRUE: Enable user-defined unit mode 12
UserMode13	Bool	TRUE: Enable user-defined unit mode 13
UserMode14	Bool	TRUE: Enable user-defined unit mode 14
UserMode15	Bool	TRUE: Enable user-defined unit mode 15
UserMode16	Bool	TRUE: Enable user-defined unit mode 16
UserMode17	Bool	TRUE: Enable user-defined unit mode 17
UserMode18	Bool	TRUE: Enable user-defined unit mode 18
UserMode19	Bool	TRUE: Enable user-defined unit mode 19
UserMode20	Bool	TRUE: Enable user-defined unit mode 20
UserMode21	Bool	TRUE: Enable user-defined unit mode 21
UserMode22	Bool	TRUE: Enable user-defined unit mode 22
UserMode23	Bool	TRUE: Enable user-defined unit mode 23
UserMode24	Bool	TRUE: Enable user-defined unit mode 24
UserMode25	Bool	TRUE: Enable user-defined unit mode 25
UserMode26	Bool	TRUE: Enable user-defined unit mode 26
UserMode27	Bool	TRUE: Enable user-defined unit mode 27
UserMode28	Bool	TRUE: Enable user-defined unit mode 28

Output parameters

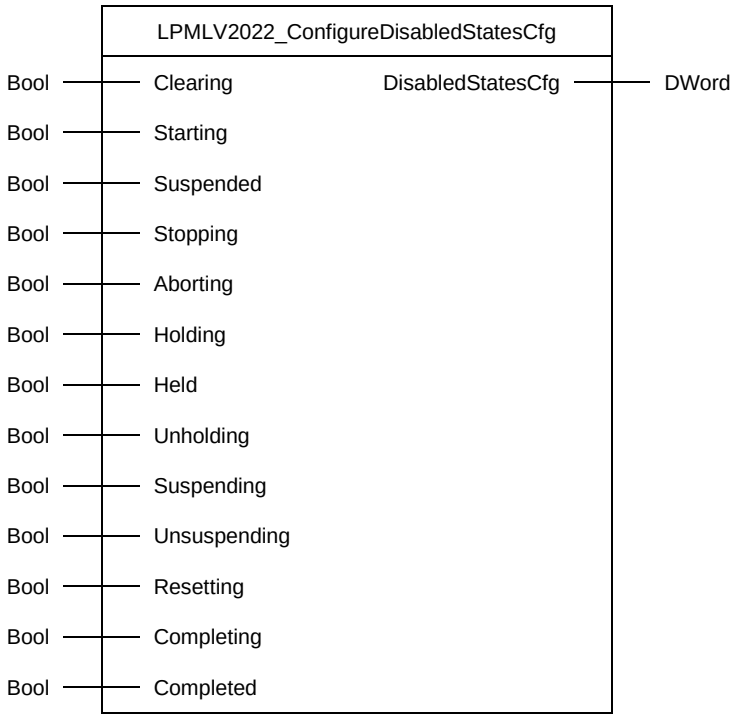
Table 2-12: *LPMLV2022_ConfigureEnabledModesCfg* output parameters

Parameter	Data type	Comment
EnabledModesCfg	DWord	Enabled unit modes configuration for direct connection with the input *configuration.EnabledModesCfg* of the desired *LPMLV2022_UnitModeStateManager* FB instance

2.2.6 FC LPMLV2022_ConfigureDisabledStatesCfg (FC 30111)

Figure

Figure 2-7: LPMLV2022_ConfigureDisabledStatesCfg



Principle of operation

This function allows the user to set the state configuration for every unit mode in the FB *LPMLV2022_UnitModeStateManager* and the FB *LPMLV2022_UnitModeStateManagerBool* easily. Of course, it is also possible to set the state configurations directly in the FB *LPMLV2022_UnitModeStateManager* configuration.

With the function, the user has to set the associated inputs for the different states to “TRUE”, e.g. “*Held := TRUE*” for disabling the state *Held*.

The function generates a DWord value which represents the state configuration for one unit mode. This value is bit coded and means that every bit represents a switch where states can be disabled or enabled for a unit mode, e.g. disabling the state *Held* the bit number 11 has to be set to “TRUE”. As can be seen in the example the state numbers according to the OMAC standard also define the bit numbers in the DWord value (see Table 1-2: Description of the possible states).

To write the state configuration from the function output to the according *“Unit Mode and State Manager”,* the output *DisabledStatesCfg* has to be connected to the configuration of the corresponding FB *LPMLV2022_UnitModeStateManager*.

NOTE

According to the OMAC standard some states are mandatory (Stopped, Idle, Execute and Aborted) and cannot be disabled. If they are nevertheless disabled, the “*Unit Mode and State Manager”* will enable these states automatically and provide the corrected configuration as a DWord value at output *CurDisabledStates*.

Input parameters

Table 2-13: *LPMLV2022_ConfigureDisabledStatesCfg* input parameters

Parameter	Data type	Comment
Clearing	Bool	TRUE: Disable state Clearing
Starting	Bool	TRUE: Disable state Starting
Suspended	Bool	TRUE: Disable state Suspended
Stopping	Bool	TRUE: Disable state Stopping
Aborting	Bool	TRUE: Disable state Aborting
Holding	Bool	TRUE: Disable state Holding
Held	Bool	TRUE: Disable state Held
Unholding	Bool	TRUE: Disable state Unholding
Suspending	Bool	TRUE: Disable state Suspending
Unsuspending	Bool	TRUE: Disable state Unsuspending
Resetting	Bool	TRUE: Disable state Resetting
Completing	Bool	TRUE: Disable state Completing
Completed	Bool	TRUE: Disable state Completed

Output parameters

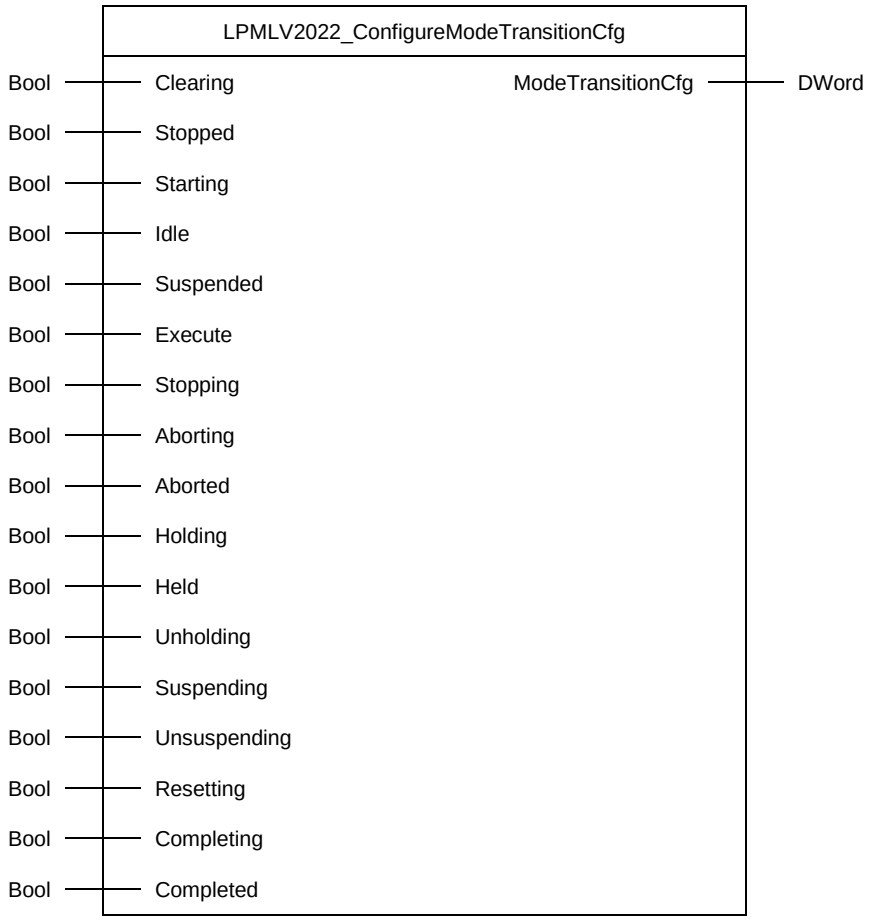
Table 2-14: *LPMLV2022_ConfigureDisabledStatesCfg* output parameters

Parameter	Data type	Comment
DisabledStatesCfg	DWord	Disabled states configuration for one unit mode for direct connection with the input *configuration.DisabledStatesCfg[<unit mode>]* of the desired FB *LPMLV2022_UnitModeStateManager* instance. Bit locations within the DWORD value represent State numbers

2.2.7 FC LPMLV2022_ConfigureModeTransitionCfg (FC 30112)

Figure

Figure 2-8: LPMLV2022_ConfigureModeTransitionCfg



Principle of operation

This function allows the user to set the Mode Transition Configuration for one unit mode of the FB *LPMLV2022_UnitModeStateManager* and the FB *LPMLV2022_UnitModeStateManagerBool* easily. The output of this function can directly be assigned to the *LPMLV2022_UnitModeStateManager* configuration *ModeTransitionCfg[<unit mode>]*. The bit locations within the *ModeTransitionCfg* DWord represent state numbers. A set bit means, that a mode transition to or from the corresponding state is allowed. Note, that the corresponding state bit must be set for both the current mode and the requested mode.

NOTE This function is only needed, if another configuration than the default configuration of the *ModeTransitionCfg* of the *LPMLV2022_UnitModeStateManager* shall be used (default: 16#0000_0214, i.e. a mode change is possible in the states Stopped, Idle and Aborted).

Input parameters

Table 2-15: *LPMLV2022_ConfigureModeTransitionCfg* input parameters

Parameter	Data type	Comment
Clearing	Bool	TRUE: Allow a mode transition to/from state Clearing
Stopped	Bool	TRUE: Allow a mode transition to/from state Stopped
Starting	Bool	TRUE: Allow a mode transition to/from state Starting
Idle	Bool	TRUE: Allow a mode transition to/from state Idle
Suspended	Bool	TRUE: Allow a mode transition to/from state Suspended
Execute	Bool	TRUE: Allow a mode transition to/from state Execute
Stopping	Bool	TRUE: Allow a mode transition to/from state Stopping
Aborting	Bool	TRUE: Allow a mode transition to/from state Aborting
Aborted	Bool	TRUE: Allow a mode transition to/from state Aborted
Holding	Bool	TRUE: Allow a mode transition to/from state Holding
Held	Bool	TRUE: Allow a mode transition to/from state Held
Unholding	Bool	TRUE: Allow a mode transition to/from state Unholding
Suspending	Bool	TRUE: Allow a mode transition to/from state Suspending
Unsuspending	Bool	TRUE: Allow a mode transition to/from state Unsuspending
Resetting	Bool	TRUE: Allow a mode transition to/from state Resetting
Completing	Bool	TRUE: Allow a mode transition to/from state Completing
Completed	Bool	TRUE: Allow a mode transition to/from state Completed

Output parameters

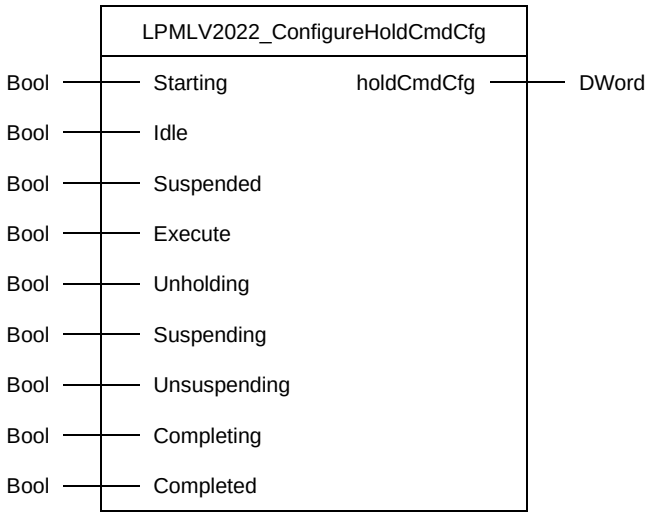
Table 2-16: *LPMLV2022_ConfigureModeTransitionCfg* output parameters

Parameter	Data type	Comment
ModeTransitionCfg	DWord	Mode Transition Configuration for one unit mode for direct connection with the input *configuration.ModeTransitionCfg[<unit mode>]* of the desired instance of the FB *LPMLV2022_UnitModeStateManager*. Bit locations within the DWORD value represent State numbers

2.2.8 FC LPMLV2022_ConfigureHoldCmdCfg (FC 30113)

Figure

Figure 2-9: LPMLV2022_ConfigureHoldCmdCfg



Principle of operation

This function allows the user to set the Hold control command configuration for all unit modes of the *LPMLV2022_UnitModeStateManager* easily (one config. for all modes). The output of this function can directly be assigned to the *LPMLV2022_UnitModeStateManager* configuration *holdCmdCfg*. The bit locations within the *holdCmdCfg* DWord represent state numbers. A set bit means, that the Hold command is taken into account by the *LPMLV2022_UnitModeStateManager* in the corresponding state.

NOTE This function is only needed, if another configuration than the default configuration of the holdCmdCfg of the LPMLV2022_UnitModeStateManager shall be used (default: 16#0000_0060, i.e. only the bits for the states Suspended and Execute are set to comply with ANSI/ISA-TR88.00.02-2022).

Input parameters

Table 2-17: *LPMLV2022_ConfigureHoldCmdCfg* input parameters

Parameter	Data type	Comment
Starting	Bool	TRUE: The Hold control command is taken into account in state Starting
Idle	Bool	TRUE: The Hold control command is taken into account in state Idle
Suspended	Bool	TRUE: The Hold control command is taken into account in state Suspended
Execute	Bool	TRUE: The Hold control command is taken into account in state Execute
Unholding	Bool	TRUE: The Hold control command is taken into account in state Unholding
Suspending	Bool	TRUE: The Hold control command is taken into account in state Suspending
Unsuspending	Bool	TRUE: The Hold control command is taken into account in state Unsuspending
Completing	Bool	TRUE: The Hold control command is taken into account in state Completing
Completed	Bool	TRUE: The Hold control command is taken into account in state Completed

Output parameters

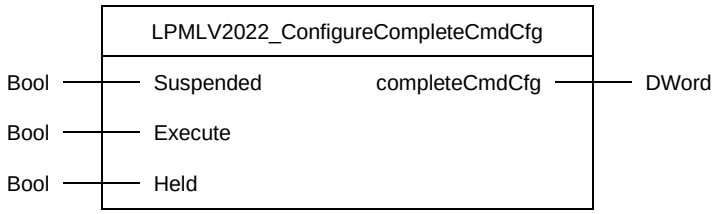
Table 2-18: *LPMLV2022_ConfigureHoldCmdCfg* output parameters

Parameter	Data type	Comment
holdCmdCfg	DWord	Hold control command configuration for all unit modes (one config. for all modes) for direct connection with the input *configuration.holdCmdCfg* of the desired FB *LPMLV2022_UnitModeStateManager* instance. Bit locations within the DWORD value represent State numbers

2.2.9 FC LPMLV2022_ConfigureCompleteCmdCfg (FC 30114)

Figure

Figure 2-10: LPMLV2022_ConfigureCompleteCmdCfg



Principle of operation

This function allows the user to set the Complete control command configuration for all unit modes of the *LPMLV2022_UnitModeStateManager* and the FB *LPMLV2022_UnitModeStateManagerBool* easily (one config. for all modes). The output of this function can directly be assigned to the *LPMLV2022_UnitModeStateManager* configuration *completeCmdCfg*. The bit locations within the *completeCmdCfg* DWord represent state numbers. A set bit means, that the Complete command is taken into account by the *LPMLV2022_UnitModeStateManager* in the corresponding state.

NOTE This function is only needed, if another configuration than the default configuration of the completeCmdCfg of the LPMLV2022_UnitModeStateManager shall be used (default: 16#0000_0860, i.e. the bits for the states Suspended, Execute and Held are set to comply with ANSI/ISA-TR88.00.02-2022).

Input parameters

Table 2-19: *LPMLV2022_ConfigureCompleteCmdCfg* input parameters

Parameter	Data type	Comment
Suspended	Bool	TRUE: The Complete control command is taken into account in state Suspended
Execute	Bool	TRUE: The Complete control command is taken into account in state Execute
Held	Bool	TRUE: The Complete control command is taken into account in state Held

Output parameters

Table 2-20: *LPMLV2022_ConfigureCompleteCmdCfg* output parameters

Parameter	Data type	Comment
completeCmdCfg	DWord	Complete control command configuration for all unit modes (one config. for all modes) for direct connection with the input *configuration.completeCmdCfg* of the desired FB *LPMLV2022_UnitModeStateManager* instance. Bit locations within the DWORD value represent State numbers

2.2.10 PLC data types

LPMLV2022_typeConfiguration

Table 2-21: *LPMLV2022_typeConfiguration*

Name	Data type	Comment
EnabledModesCfg	DWord	Bit locations within the DWORD represent Mode numbers. A value of 1 in a bit location [0-31] indicates that the corresponding Mode number is enabled. Mode 0 is an Invalid Mode, therefore bit 0 is unused (default: 16#0000_01FE) Note: 16#0000_01FE -> Production, Maintenance, Manual, UserMode01..05
DisabledStatesCfg	Array[0.."LPMLV2022_MAX_MODES_UPPER_LIM"] of DWord	The array index represents the Mode number. Bit locations within the DWORD value represent State numbers. A value of 1 in a bit location indicates that the corresponding state number is disabled. State 0 is an Undefined State, therefore bit 0 is unused (default: 16#0000_0000)
ModeTransitionCfg	Array[0.."LPMLV2022_MAX_MODES_UPPER_LIM"] of DWord	The array index represents the Mode number. Bit locations within the DWORD value represent State numbers. A value of 1 in a bit location indicates that a mode transition from the corresponding state number is allowed (default: 16#0, 31(16#0000_0214)) Note: 16#0000_0214 -> Stopped, Idle, Aborted
holdCmdCfg	DWord	Bit locations within the DWORD value represent State numbers. A value of 1 in a bit location indicates that the Hold control command is taken into account in the corresponding state. Note: a state change is only performed if the Holding/Held states are not disabled. The 'holdCmdCfg' can only be activated for the following states: Idle, Starting, Execute, Completing, Completed, Suspending, Suspended, Unsuspending, Unholding (default: 16#0000_0060) Note: 16#0000_0060 -> Suspended, Execute
completeCmdCfg	DWord	Bit locations within the DWORD value represent State numbers. A value of 1 in a bit location indicates that the Complete control command is taken into account in the corresponding state. Note: a state change is only performed if the Completing/Completed states are not disabled. The 'completeCmdCfg' can only be activated for the following states: Execute, Suspended, Held (default: 16#0000_0860) Note: 16#0000_0860 -> Suspended, Execute, Held

LPMLV2022_typeDiagnostics

Table 2-22: *LPMLV2022_typeDiagnostics*

Name	Data type	Comment
bufferIndex	Int	Index of actual buffer entry (default: -1)
buffer	Array[0.. LPMLV2022_DIAG_BUFFER_UPPER_LIM] of LPMLV2022_typeDiagnosticsEntry	Diagnostics information buffer

LPMLV2022_typeDiagnosticsEntry

Table 2-23: *LPMLV2022_typeDiagnosticsEntry*

Name	Data type	Comment
timestamp	DTL	Timestamp for this entry (default: DTL#1970-01-01-00:00:00)
UnitModeCurrent	SInt	Current unit mode (default: 0)
StateCurrent	SInt	Current state (default: 0)
UnitMode	SInt	Requested unit mode (default: 0)
CntrlCmd	SInt	Request control command (default: 0)
SC	Bool	State complete signal (default: FALSE)
message	Byte	Message for this entry (default: 16#00)

LPMLV2022_typePackTags

Table 2-24: *LPMLV2022_typePackTags*

Name	Data type	Comment
Command	LPMLV2022_typeCommand	Command tags are used to control the operation of the unit/machine. Command tags include unit state commands which control the state transitions in the base state model. The command tags also include parameters and process variables which control how the machine operates
Status	LPMLV2022_typeStatus	Status tags are used to describe the operation of the unit/machine. Status tags include state commands which describe the state transitions in the base state model. The status tags also include parameters and process variables which describe how the machine operates
Admin	LPMLV2022_typeAdmin	Administration tags are used to describe the quality and alarm information of the unit/machine. Administration tags include alarm parameters which describe the conditions within the base state model typically for production data acquisition (PDA) systems. The administration tags also include parameters which can describe how well the machine operates, or specific information on the product quality produced by the machine

LPMLV2022_typeAdmin

Table 2-25: *LPMLV2022_typeAdmin*

Name	Data type	Comment
Parameter_REAL	Array[0..LPMLV2022_ADMIN_PARAMETER_REAL_UPPER_LIM] of LPMLV2022_typeParameter_REAL	Structured Array of Unit/Machine Parameter information for values with REAL data type
Parameter_STRING	Array[0..LPMLV2022_ADMIN_PARAMETER_STRING_UPPER_LIM] of LPMLV2022_typeParameter_STRING	Structured Array of Unit/Machine Parameter information for values with STRING data type
Parameter_LREAL	Array[0..LPMLV2022_ADMIN_PARAMETER_LREAL_UPPER_LIM] of LPMLV2022_typeParameter_LREAL	Structured Array of Unit/Machine Parameter information for values with LREAL data type
Parameter_DINT	Array[0..LPMLV2022_ADMIN_PARAMETER_DINT_UPPER_LIM] of LPMLV2022_ty	Structured Array of Unit/Machine Parameter information for values with DINT data type

Name	Data type	Comment
	peParameter_DINT	
Parameter_BOOL	Array[0..LPMLV2022_ADMIN_PARAMETER_BOOL_UPPER_LIM] of LPMLV2022_typeParameter_BOOL	Structured Array of Unit/Machine Parameter information for values with BOOL data type
Alarm	Array[0..LPMLV2022_ADMIN_ALARM_UPPER_LIM] of LPMLV2022_typeEvent	Array of Given Size for Machine Alarms. The alarm tags associated to the local interface are typically used as parameters that are displayed or used on the unit locally, for example from an HMI. These alarm parameters can be used to display any alarm, or machine downtime cause that is currently occurring in the system
AlarmExtent	DInt	Extent of Alarm Array. The alarm extent is associated with the maximum number of alarms needed for the machine annunciation or reporting
AlarmHistory	Array[0..LPMLV2022_ADMIN_ALARM_HISTORY_UPPER_LIM] of LPMLV2022_typeEvent	Array of Given Size for Machine Fault Number and Messaging History. The AlarmHistory array is reserved for alarms that have occurred on the unit/machine and can be sorted in chronological order with the most recently occurring alarmed indexed as Admin.AlarmHistory[0]
AlarmHistoryExtent	DInt	Extent of Alarm History Array. The alarm history extent is associated with the maximum number of alarms needed to be archived or tagged as alarm history for the machine
StopReason	LPMLV2022_typeEvent	Machine Stop Reason is typically used for "First Out Fault" Reporting and Other Stoppage Events. The stop reason is the first event captured during an abort, held, suspended or stop event
Warning	Array[0..LPMLV2022_ADMIN_WARNING_UPPER_LIM] of LPMLV2022_typeEvent	Machine warnings are for general events that do not cause the machine to stop, but may require operator action as a stoppage may be imminent. Warnings are not typically stored in history
WarningExtent	DInt	Extent of Warning Array. The warning extent is associated with the maximum number of warnings needed to be archived or tagged as warnings for the machine
ModeTimeCurrent	DInt	Current Mode Time. This tag represents the current amount of time (in seconds) spent in the current Mode as indicated by Status.UnitModeCurrent. The value will start from 0 every time the Mode is changed
StateTimeCurrent	DInt	Current State Time. This tag represents the current amount of time (in seconds) spent in the current State as indicated by Status.StateCurrent. The value will start from 0 every time the State is changed
CumulativeTimes	Array[0..LPMLV2022_ADMIN_CUMULATIVE_TIMES_UPPER_LIM] of LPMLV2022_typeCumulativeTimes	Structured Array of Timer Values. This tag represents a collection of accumulated time (in seconds) spent in any defined state of any defined mode

Name	Data type	Comment
ProductData	Array[0..LPMLV2022_ADMIN_PRODUCT_DATA_UPPER_LIM] of LPMLV2022_typeProductData	Structured array of Product Stream Data. This tag represents a collection of information about each unique input or output product stream of the machine unit. The extent of the array is typically limited to the maximum number of input and output product streams
MachDesignSpeed	Real	Machine Design Speed. This tag represents the maximum design speed of the machine in primary packages per minute for the package configuration being run. This speed is NOT the maximum speed as specified by the manufacturer, but rather the speed of the machine is designed to run in its installed environment
DisabledStatesCfg	Array[0..LPMLV2022_ADMIN_NUMBER_OF_MODES_UPPER_LIM] of DWord	An array reflecting the Disabled States Configuration for all defined Modes. The array index represents the Mode number. Bit locations within the DWORD value represent State numbers. A value of 1 in a bit location indicates that the corresponding state number is disabled. State 0 is an Undefined State, therefore bit 0 is unused
CurDisabledStates	DWord	Reflects the disabled states for the current Mode (Status.UnitModeCurrent). State numbers are represented by bit location. A value of 1 in a certain bit location indicates that the corresponding state number is disabled. State 0 is an Undefined State, therefore bit 0 is unused
EnabledModesCfg	DWord	An element reflecting the Enabled Modes Configuration for the unit/machine. Bit locations within the DWORD represent Mode numbers. A value of 1 in a bit location [0-31] indicates that the corresponding Mode number is enabled. Mode 0 is an Invalid Mode, therefore bit 0 is unused
ModeTransitionCfg	Array[0..LPMLV2022_ADMIN_NUMBER_OF_MODES_UPPER_LIM] of DWord	An array reflecting the Mode Transition Configuration for all modes of the unit/machine. The array index represents the Mode number. Bit locations within the DWORD value represent State numbers. A value of 1 in a bit location indicates that a mode transition from the corresponding state number is allowed
PLCDateTime	LPMLV2022_typeDateTime	Current Date and Time of the Programmable Logic Controller
PLCDateTimeDTL	DTL	Current Date and Time (DTL) of the Programmable Logic Controller

LPMLV2022_typeCommand

Table 2-26: *LPMLV2022_typeCommand*

Name	Data type	Comment
UnitMode	DInt	Unit Mode Target (0: Invalid, 1: Production, 2: Maintenance, 3: Manual, 04 - 31: User Definable). Minimum required for supervisory control
UnitModeChangeRequest	Bool	Request Unit Mode Change to Unit Mode Target. Minimum required for supervisory control
MachSpeed	Real	Current Machine Speed Setpoint; Unit of Measure: Primary Packages/Minute. Minimum required for supervisory control
MaterialInterlock	DWord	Materials Ready. It is comprised of a series of bits with 1 equaling ready or not low, 0 equaling not ready, or low. Each bit represents a different user material. Unused bits should be forced to a value of 1 or TRUE

Name	Data type	Comment
CntrlCmd	DInt	Control Command (0: Undefined, 1: Reset, 2: Start, 3: Stop, 4: Hold, 5: UnHold, 6: Suspend, 7: UnSuspend, 8: Abort, 9: Clear, 10: Complete). Minimum required for supervisory control
CmdChangeRequest	Bool	State Change Request. This CmdChangeRequest bit will command the machine to proceed to change the state to the target state. Minimum required for supervisory control
Parameter_REAL	Array[0..LPMLV2022_COMMAND_PARAMETER_REAL_UPPER_LIM] of LPMLV2022_typeParameter_REAL	Structured Array of Unit/Machine Parameter information for values with REAL data type
Parameter_STRING	Array[0..LPMLV2022_COMMAND_PARAMETER_STRING_UPPER_LIM] of LPMLV2022_typeParameter_STRING	Structured Array of Unit/Machine Parameter information for values with STRING data type
Parameter_LREAL	Array[0..LPMLV2022_COMMAND_PARAMETER_LREAL_UPPER_LIM] of LPMLV2022_typeParameter_LREAL	Structured Array of Unit/Machine Parameter information for values with LREAL data type
Parameter_DINT	Array[0..LPMLV2022_COMMAND_PARAMETER_DINT_UPPER_LIM] of LPMLV2022_typeParameter_DINT	Structured Array of Unit/Machine Parameter information for values with DINT data type
Parameter_BOOL	Array[0..LPMLV2022_COMMAND_PARAMETER_BOOL_UPPER_LIM] of LPMLV2022_typeParameter_BOOL	Structured Array of Unit/Machine Parameter information for values with BOOL data type
SelectedRecipe	DInt	Recipe Selection. This tag is used to designate which recipe should be run on the machine to produce the primary output product, according to a user-design recipe handling procedure. It is designed to correspond directly to the listing in the Recipe Array
RecipeChangeRequest	Bool	Recipe change request. This tag is used to trigger a user-defined recipe changeover process that will configure the machine to produce the primary product designated in Command.SelectedRecipe
Recipe	Array[0..LPMLV2022_COMMAND_RECIPE_U	Structured Array of Recipe Information. The recipe data type can be used for defining product ingredient and product processing parameter variables

Name	Data type	Comment
	PPER_LIM] of LPMLV2022_typeCommandRecipe	

LPMLV2022_typeStatus

Table 2-27: *LPMLV2022_typeStatus*

Name	Data type	Comment
UnitModeCurrent	DInt	Current Unit Mode Number (0: Invalid, 1: Production, 2: Maintenance, 3: Manual, 04 - 31: User Definable). Minimum required for information/machine monitoring
UnitModeRequested	Bool	Requested Unit Mode Change. When a unit mode request takes place a numerical value must be present in the unit mode target to change the unit mode
UnitModeChangeInProcess	Bool	Requested Unit Mode Change In Process
StateCurrent	DInt	Current State Number (0: Undefined, 1: CLEARING, 2: STOPPED, 3: STARTING, 4: IDLE, 5: SUSPENDED, 6: EXECUTE, 7: STOPPING, 8: ABORTING, 9: ABORTED; 10: HOLDING, 11: HELD, 12: UNHOLDING, 13: SUSPENDING, 14: UNSUSPENDING, 15: RESETTING, 16: COMPLETING, 17: COMPLETED). Minimum required for information/machine monitoring
StateRequested	DInt	Target State. This value is used for state transition checking, to ensure that transition to a target state can be achieved
StateChangeInProcess	Bool	State Change in Process. This bit indicates that a change in state is in progress following a state change request command
MachSpeed	Real	Current Machine Speed Setpoint. This describes the set point for the current speed of the machine in primary packages per minute. Minimum required for information/machine monitoring
CurMachSpeed	Real	Current Actual Machine Speed. This is the actual value of the machine speed in primary packages per minute. Minimum required for information/machine monitoring
MaterialInterlock	DWord	Materials Ready. MaterialInterlock describes the status of the materials that are ready for processing. It is comprised of a series of bits with 1 equaling ready or not low, 0 equaling not ready, or low. Each bit represents a different user material
EquipmentInterlock	LPMLV2022_typeEquipmentInterlock	Is a collection of tags that are used to provide an indicator of Blocked or Starved conditions at the machine unit when integrated into a complete production line
Parameter_REAL	Array[0..LPMLV2022_STATUS_PARAMETER_REAL_UPPER_LIM] of LPMLV2022_typeParameter_REAL	Structured Array of Unit/Machine Parameter information for values with REAL data type
Parameter_STRING	Array[0..LPMLV2022_STATUS_PARAMETER	Structured Array of Unit/Machine Parameter information for values with STRING data type

Name	Data type	Comment
	_STRING_UPPER_LIM] of LPMLV2022_typeParameter_STRING	
Parameter_LREAL	Array[0..LPMLV2022_STATUS_PARAMETER_LREAL_UPPER_LIM] of LPMLV2022_typeParameter_LREAL	Structured Array of Unit/Machine Parameter information for values with LREAL data type
Parameter_DINT	Array[0..LPMLV2022_STATUS_PARAMETER_DINT_UPPER_LIM] of LPMLV2022_typeParameter_DINT	Structured Array of Unit/Machine Parameter information for values with DINT data type
Parameter_BOOL	Array[0..LPMLV2022_STATUS_PARAMETER_BOOL_UPPER_LIM] of LPMLV2022_typeParameter_BOOL	Structured Array of Unit/Machine Parameter information for values with BOOL data type
RecipeCurrent	DInt	The recipe currently running in production
RecipeRequested	DInt	A reflection of the recipe currently selected for production
RecipeChangeInProcess	Bool	An indicator for the user-defined recipe changeover process
Recipe	Array[0..LPMLV2022_STATUS_RECIPE_UPPER_LIM] of LPMLV2022_typeStatusRecipe	Structured Array of Recipe Information. The recipe data type can be used for defining product ingredient and product processing parameter variables
Stacklight	Array[0..LPMLV2022_STATUS_STACK_LIGHT_UPPER_LIM] of DWord	Indicator for the current stacklight status. This tag can be used simultaneously for reporting stacklight conditions and as control bits for physical outputs. The status of a light in the stack is associated to a particular bit location within the register (Bit 0: Red Solid, 1: Red Flashing, 2: Amber Solid, 3: Amber Flashing, 4: Blue Solid, 5: Blue Flashing, 6: Green Solid, 7: Green Flashing, 8: Horn Solid, 9: Horn Flashing, 10..31: User-Defined)

LPMLV2022_typeEvent

Table 2-28: *LPMLV2022_typeEvent*

Name	Data type	Comment
Trigger	Bool	Alarm Message Trigger. Alarm trigger should be turned on only when the alarm is currently active
ID	DInt	Alarm Message Identification Number. The alarm ID number is a unique value assigned to each alarm. Minimum required for information/machine monitoring
Value	DInt	Alarm Message Number. The alarm message number is a value that is associated to the alarm allowing for user specific detail or to break down the Alarm.ID to greater detail
Message	String[80]	Alarm Message. The alarm message is the actual text of the alarm for those machines capable of providing string information
Category	DInt	Alarm Category. Alarm category is used to identity what type of alarm has occurred (e.g. electrical, mechanical, materials, utilities, etc.)
DateTime	LPMLV2022_typeDateTime	Date and Time the Alarm Occurred. The timestamp for when the alarm was first triggered
DateTimeDTL	DTL	Date and Time (DTL) the Alarm Occurred. The timestamp for when the alarm was first triggered
AckDateTime	LPMLV2022_typeDateTime	Date and Time the Alarm was Acknowledged. The timestamp for when the alarm was acknowledged by the operator, or cleared
AckDateTimeDTL	DTL	Date and Time (DTL) the Alarm was Acknowledged. The timestamp for when the alarm was acknowledged by the operator, or cleared

LPMLV2022_typeCommandIngredients

Table 2-29: *LPMLV2022_typeCommandIngredients*

Name	Data type	Comment
Parameter_REAL	Array[0..LPMLV2022_COMMAND_INGREDIENTS_PARAMETER_REAL_UPPER_LIM] of LPMLV2022_typeParameter_REAL	Structured Array of Ingredient information for values with REAL data type
Parameter_STRING	Array[0..LPMLV2022_COMMAND_INGREDIENTS_PARAMETER_STRING_UPPER_LIM] of LPMLV2022_typeParameter_STRING	Structured Array of Ingredient information for values with STRING data type
Parameter_LREAL	Array[0..LPMLV2022_COMMAND_INGREDIENTS_PARAMETER_LREAL_U	Structured Array of Ingredient information for values with LREAL data type

Name	Data type	Comment
	PPER_LIM] of LPMLV2022_typeParameter_LREAL	
Parameter_DINT	Array[0..LPMLV2022_COMMAND_INGREDIENTS_PARAMETER_DINT_UPPER_LIM] of LPMLV2022_typeParameter_DINT	Structured Array of Ingredient Information for values with DINT data type
Parameter_BOOL	Array[0..LPMLV2022_COMMAND_INGREDIENTS_PARAMETER_BOOL_UPPER_LIM] of LPMLV2022_typeParameter_BOOL	Structured Array of Ingredient Information for values with BOOL data type

LPMLV2022_typeCommandProcessVariables

Table 2-30: *LPMLV2022_typeCommandProcessVariables*

Name	Data type	Comment
Parameter_REAL	Array[0..LPMLV2022_COMMAND_PROCESS_VARIABLES_PARAMETER_REAL_UPPER_LIM] of LPMLV2022_typeParameter_REAL	Structured Array of Recipe Process Variable information for values with REAL data type
Parameter_STRING	Array[0..LPMLV2022_COMMAND_PROCESS_VARIABLES_PARAMETER_STRING_UPPER_LIM] of LPMLV2022_typeParameter_STRING	Structured Array of Recipe Process Variable information for values with STRING data type
Parameter_LREAL	Array[0..LPMLV2022_COMMAND_PROCESS_VARIABLES_PARAMETER_LREAL_UPPER_LIM] of LPMLV2022_typeParameter_LREAL	Structured Array of Recipe Process Variable information for values with LREAL data type
Parameter_DINT	Array[0..LPMLV	Structured Array of Recipe Process Variable Information for

Name	Data type	Comment
	2022_COMMAND_PROCESS_VARIABLES_PARAMETER_DINT_UPPER_LIM] of LPMLV2022_typeParameter_DINT	values with DINT data type
Parameter_BOOL	Array[0..LPMLV2022_COMMAND_PROCESS_VARIABLES_PARAMETER_BOOL_UPPER_LIM] of LPMLV2022_typeParameter_BOOL	Structured Array of Recipe Process Variable Information for values with BOOL data type

LPMLV2022_typeCommandRecipe

Table 2-31: *LPMLV2022_typeCommandRecipe*

Name	Data type	Comment
ID	DInt	ID Value of the Recipe. The ID is used to indicate to the machine which recipe is being produced
Name	String[80]	Name of Recipe. The Recipe Name is used to further identify the product being produced, especially when alphanumeric characters are used (i.e. SKU or UPC)
Unit	String[6]	Unit of Measure of Recipe. The Recipe Unit is used to describe the unit of production for the product being produced
PrimaryQty	Real	Primary Quantity Value of Recipe. PrimaryQty is used to provide a common reference for manufacturing throughput across all the machines on a production line. This element describes the number of Primary Products per recipe Unit for the product being produced by this unit/machine
ProcessVariables	LPMLV2022_typeCommandProcessVariables	Process Variables for each Recipe. The ProcessVariables structure can be used to hold particular set points needed by the unit/machine for the processing of a specific recipe
Ingredients	LPMLV2022_typeCommandIngredients	Ingredient Information for each Recipe. The Ingredients structure can be used to hold information for the raw materials that are used by the unit/machine in the processing of a particular product or recipe

LPMLV2022_typeCumulativeTimes

Table 2-32: *LPMLV2022_typeCumulativeTimes*

Name	Data type	Comment
AccTimeSinceReset	DInt	Accumulated Time Since Last Reset. The tag represents the amount of time (in seconds) since the reset has been triggered for a particular collection of cumulative times. When a reset for a particular collection is triggered, all times inside the structure for that collection are reset
ModeStateTimes	Array[0..LPMLV2022_ADMIN_NUMBER_OF_MODES_UPPER_LIM] of LPMLV2022_typeModeStateTimes	Structured Array of Mode and State Time Values for each Cumulative Time Tracker. This tag is a collection of times for modes and states since the last timer reset was executed

LPMLV2022_typeDateTime

Table 2-33: *LPMLV2022_typeDateTime*

Name	Data type	Comment
Year	Int	
Month	Int	
Day	Int	
Hour	Int	
Minute	Int	
Second	Int	
Millisecond	Int	

LPMLV2022_typeEquipmentInterlock

Table 2-34: *LPMLV2022_typeEquipmentInterlock*

Name	Data type	Comment
Blocked	Bool	Indicator for Unit/Machine Blocked. This bit, when set to 1, indicates that a downstream system is not able to accept product. In this condition, the equipment is capable of producing product but is in a suspended state due to a downstream system
Starved	Bool	Indicator for Unit/Machine Starved. This bit, when set to 1, indicates that an upstream system is not able to supply product. In this condition, the equipment is capable of producing product but is in a suspended state due to an upstream system

LPMLV2022_typeModeStateTimes

Table 2-35: *LPMLV2022_typeModeStateTimes*

Name	Data type	Comment
Mode	DInt	Mode Time Values for each Mode. This tag represents the cumulative amount of time (in seconds) spent in each Mode since the last timer and counter reset was executed
State	Array[0..LPMLV2022_ADMIN_NUMBER_OF_STATES_UPPER_LIM] of DInt	State Time Values for each State in each Mode. This tag represents the cumulative amount of time (in seconds) spent in each state of a particular Mode since the last timer and counter reset was executed

LPMLV2022_typeParameter_BOOL

Table 2-36: *LPMLV2022_typeParameter_BOOL*

Name	Data type	Comment
ID	DInt	ID Value of Parameter. This is the arbitrary (user defined) ID value of the parameter
Name	String[80]	Name of Parameter. The literal parameter name is used to describe the parameter number, and its associated value
Unit	String[6]	Unit of Measure of Parameter. The parameter unit is typically not used for boolean values
Value	Bool	Value of Parameter. This is the boolean value of the parameter (0: false, 1: true)

LPMLV2022_typeParameter_DINT

Table 2-37: *LPMLV2022_typeParameter_DINT*

Name	Data type	Comment
ID	DInt	ID Value of Parameter. This is the arbitrary (user defined) ID value of the parameter
Name	String[80]	Name of Parameter. The literal parameter name is used to describe the parameter number, and its associated value
Unit	String[6]	Unit of Measure of Parameter. An example parameter unit of measure may be DegF, secs, PPM, revs, mm, etc.
Value	DInt	Value of Parameter. This is the numeric value of the parameter

LPMLV2022_typeParameter_LREAL

Table 2-38: *LPMLV2022_typeParameter_LREAL*

Name	Data type	Comment
ID	DInt	ID Value of Parameter. This is the arbitrary (user defined) ID value of the parameter
Name	String[80]	Name of Parameter. The literal parameter name is used to describe the parameter number, and its associated value
Unit	String[6]	Unit of Measure of Parameter. An example parameter unit of measure may be DegF, secs, PPM, revs, mm, etc.
Value	LReal	Value of Parameter. This is the numeric value of the parameter

LPMLV2022_typeParameter_REAL

Table 2-39: *LPMLV2022_typeParameter_REAL*

Name	Data type	Comment
ID	DInt	ID Value of Parameter. This is the arbitrary (user defined) ID value of the parameter
Name	String[80]	Name of Parameter. The literal parameter name is used to describe the parameter number, and its associated value
Unit	String[6]	Unit of Measure of Parameter. An example parameter unit of measure may be DegF, secs, PPM, revs, mm, etc.
Value	Real	Value of Parameter. This is the numeric value of the parameter

LPMLV2022_typeParameter_STRING

Table 2-40: *LPMLV2022_typeParameter_STRING*

Name	Data type	Comment
ID	DInt	ID Value of Parameter. This is the arbitrary (user defined) ID value of the parameter
Name	String[80]	Name of Parameter. The literal parameter name is used to describe the parameter number, and its associated value
Unit	String[6]	Unit of Measure of Parameter. The parameter unit is typically not used for string values
Value	String[80]	Value of Parameter. This is the string value of the parameter

LPMLV2022_typeProductData

Table 2-41: *LPMLV2022_typeProductData*

Name	Data type	Comment
ID	DInt	ID Number for each Product Stream. The product IDs are used to report product identity for each product stream
Name	String[80]	Name of Product Stream. The Name element is used to further report the identity of the product within the product stream, especially when alphanumeric characters are used (i.e. SKU or UPC)
Unit	String[6]	Unit of Measure of each Product Stream. The Unit element is used to report the unit of production for the product within the product stream. An example product unit of measure may be Pc, Crtn, Case, etc
PrimaryQty	Real	Product Primary Quantity Value of each Product Stream. The PrimaryQty element is used to provide a common reference for manufacturing throughput across all the machines on a production line. This element relates the number of Primary Products per Unit for a particular product stream. It can be used as a conversion factor for a local machine reference speed since Command.MachSpeed is given in units of PrimaryProducts/Min
ConsumedCount	DInt	Consumed Count of each Product Stream. This tag represents the material used/consumed in the production machine for a particular input or output product stream. An example of tag usage would be the number of bags consumed in a filler, or bagger packaging machine, or the amount of linear length used, or the number caps used

Name	Data type	Comment
ProcessedCount	DInt	Processed Count of each Product Stream. This tag represents the number of products from a particular input or output product stream processed by the production machine. An example of tag usage would be the number of products that were made, including all good and defective products. The number of products processed minus the number of defective products is the number of products actually made by the machine. Minimum required for information/machine monitoring
DefectiveCount	DInt	Defective Count of each Product Stream. This tag represents the number of products from a particular input or output product stream that is marked as defective in the production machine, to be used if applicable. An example of tag usage would be the number of products rejected or products that are termed defective. Minimum required for information/machine monitoring
AccConsumedCount	DInt	Accumulated Consumption Count of each Product Stream Consumption Since Last Reset. This tag represents the total accumulated material used/consumed in the production machine for a particular input or output product stream
AccProcessedCount	DInt	Accumulated Processed Count of each Product Stream Since Last Reset. This tag represents the total accumulated number of products from a particular input or output product stream processed by the production machine. An example of tag usage would be the number of products that were made, including all good and defective products. The number of products processed minus the number of defective products is the number of products actually made by the machine
AccDefectiveCount	DInt	Accumulated Defective Count of each Product Stream Since Last Reset. This tag represents the total accumulated number of products from a particular input or output product stream that is marked as defective in the production machine, to be used if applicable. An example of tag usage would be the number of products rejected or products that are termed defective

LPMLV2022_typeStatusIngredients

Table 2-42: *LPMLV2022_typeStatusIngredients*

Name	Data type	Comment
Parameter_REAL	Array[0..LPMLV2022_STATUS_INGREDIENTS_PARAMETER_REAL_UPPER_LIM] of LPMLV2022_typeParameter_REAL	Structured Array of Ingredient information for values with REAL data type
Parameter_STRING	Array[0..LPMLV2022_STATUS_INGREDIENTS_PARAMETER_STRING_UPPER_LIM] of LPMLV2022_typeParameter_STRING	Structured Array of Ingredient information for values with STRING data type

Name	Data type	Comment
Parameter_LREAL	Array[0..LPMLV2022_STATUS_INGREDIENTS_PARAMETER_LREAL_UPPER_LIM] of LPMLV2022_typeParameter_LREAL	Structured Array of Ingredient information for values with LREAL data type
Parameter_DINT	Array[0..LPMLV2022_STATUS_INGREDIENTS_PARAMETER_DINT_UPPER_LIM] of LPMLV2022_typeParameter_DINT	Structured Array of Ingredient Information for values with DINT data type
Parameter_BOOL	Array[0..LPMLV2022_STATUS_INGREDIENTS_PARAMETER_BOOL_UPPER_LIM] of LPMLV2022_typeParameter_BOOL	Structured Array of Ingredient Information for values with BOOL data type

LPMLV2022_typeStatusProcessVariables

Table 2-43: *LPMLV2022_typeStatusProcessVariables*

Name	Data type	Comment
Parameter_REAL	Array[0..LPMLV2022_STATUS_PROCESS_VARIABLES_PARAMETER_REAL_UPPER_LIM] of LPMLV2022_typeParameter_REAL	Structured Array of Recipe Process Variable information for values with REAL data type
Parameter_STRING	Array[0..LPMLV2022_STATUS_PROCESS_VARIABLES_PARAMETER_STRING_UPPER_LIM] of LPMLV2022_typeParameter_STRING	Structured Array of Recipe Process Variable information for values with STRING data type
Parameter_LREAL	Array[0..LPMLV2022_STATUS_PROCESS_VARIABLES_PARAMETER_LREAL_UPPER_L	Structured Array of Recipe Process Variable information for values with LREAL data type

Name	Data type	Comment
	IM] of LPMLV2022_typeParameter_LREAL	
Parameter_DINT	Array[0..LPMLV2022_STATUS_PROCESS_VARIABLES_PARAMETER_DINT_UPPER_LIM] of LPMLV2022_typeParameter_DINT	Structured Array of Recipe Process Variable Information for values with DINT data type
Parameter_BOOL	Array[0..LPMLV2022_STATUS_PROCESS_VARIABLES_PARAMETER_BOOL_UPPER_LIM] of LPMLV2022_typeParameter_BOOL	Structured Array of Recipe Process Variable Information for values with BOOL data type

LPMLV2022_typeStatusRecipe

Table 2-44: *LPMLV2022_typeStatusRecipe*

Name	Data type	Comment
ID	DInt	ID Value of the Recipe. The ID is used to indicate to the machine which recipe is being produced
Name	String[80]	Name of Recipe. The Recipe Name is used to further identify the product being produced, especially when alphanumeric characters are used (i.e. SKU or UPC)
Unit	String[6]	Unit of Measure of Recipe. The Recipe Unit is used to describe the unit of production for the product being produced
PrimaryQty	Real	Primary Quantity Value of Recipe. PrimaryQty is used to provide a common reference for manufacturing throughput across all the machines on a production line. This element describes the number of Primary Products per recipe Unit for the product being produced by this unit/machine
ProcessVariables	LPMLV2022_typeStatusProcessVariables	Process Variables for each Recipe. The ProcessVariables structure can be used to reflect and confirm particular set points needed by the unit/machine for the processing of a specific recipe or to report production statistics
Ingredients	LPMLV2022_typeStatusIngredients	Ingredient Information for each Recipe. The Ingredients structure can be used to hold information for the raw materials that are used by the unit/machine in the processing of a particular product or recipe. Each element in one of the parameter arrays is considered to be an ingredient

2.2.11 PLC tags

The PLC tag table *LPMLV2022_Constants* contains user constants for unit modes, states, control commands and array boundaries.

Unit Modes

Table 2-45: Constants for unit mode

Name	Data type	Value	Comment
LPMLV2022_MODE_INVALID	DInt	0	OMAC PackMLV2022 unit mode Invalid
LPMLV2022_MODE_PRODUCTION	DInt	1	OMAC PackMLV2022 unit mode Production
LPMLV2022_MODE_MAINTENANCE	DInt	2	OMAC PackMLV2022 unit mode Maintenance
LPMLV2022_MODE_MANUAL	DInt	3	OMAC PackMLV2022 unit mode Manual
LPMLV2022_MODE_USER_01	DInt	4	OMAC PackMLV2022 user-defined unit mode 01
LPMLV2022_MODE_USER_02	DInt	5	OMAC PackMLV2022 user-defined unit mode 02
LPMLV2022_MODE_USER_03	DInt	6	OMAC PackMLV2022 user-defined unit mode 03
LPMLV2022_MODE_USER_04	DInt	7	OMAC PackMLV2022 user-defined unit mode 04
LPMLV2022_MODE_USER_05	DInt	8	OMAC PackMLV2022 user-defined unit mode 05
LPMLV2022_MODE_USER_06	DInt	9	OMAC PackMLV2022 user-defined unit mode 06
LPMLV2022_MODE_USER_07	DInt	10	OMAC PackMLV2022 user-defined unit mode 07
LPMLV2022_MODE_USER_08	DInt	11	OMAC PackMLV2022 user-defined unit mode 08
LPMLV2022_MODE_USER_09	DInt	12	OMAC PackMLV2022 user-defined unit mode 09
LPMLV2022_MODE_USER_10	DInt	13	OMAC PackMLV2022 user-defined unit mode 10
LPMLV2022_MODE_USER_11	DInt	14	OMAC PackMLV2022 user-defined unit mode 11
LPMLV2022_MODE_USER_12	DInt	15	OMAC PackMLV2022 user-defined unit mode 12
LPMLV2022_MODE_USER_13	DInt	16	OMAC PackMLV2022 user-defined unit mode 13
LPMLV2022_MODE_USER_14	DInt	17	OMAC PackMLV2022 user-defined unit mode 14
LPMLV2022_MODE_USER_15	DInt	18	OMAC PackMLV2022 user-defined unit mode 15
LPMLV2022_MODE_USER_16	DInt	19	OMAC PackMLV2022 user-defined unit mode 16
LPMLV2022_MODE_USER_17	DInt	20	OMAC PackMLV2022 user-defined unit mode 17
LPMLV2022_MODE_USER_18	DInt	21	OMAC PackMLV2022 user-defined unit mode 18
LPMLV2022_MODE_USER_19	DInt	22	OMAC PackMLV2022 user-defined unit mode 19

Name	Data type	Value	Comment
LPMLV2022_MODE_USER_20	DInt	23	OMAC PackMLV2022 user-defined unit mode 20
LPMLV2022_MODE_USER_21	DInt	24	OMAC PackMLV2022 user-defined unit mode 21
LPMLV2022_MODE_USER_22	DInt	25	OMAC PackMLV2022 user-defined unit mode 22
LPMLV2022_MODE_USER_23	DInt	26	OMAC PackMLV2022 user-defined unit mode 23
LPMLV2022_MODE_USER_24	DInt	27	OMAC PackMLV2022 user-defined unit mode 24
LPMLV2022_MODE_USER_25	DInt	28	OMAC PackMLV2022 user-defined unit mode 25
LPMLV2022_MODE_USER_26	DInt	29	OMAC PackMLV2022 user-defined unit mode 26
LPMLV2022_MODE_USER_27	DInt	30	OMAC PackMLV2022 user-defined unit mode 27
LPMLV2022_MODE_USER_28	DInt	31	OMAC PackMLV2022 user-defined unit mode 28

States

Table 2-46: Constants for states

Name	Data type	Value	Comment
LPMLV2022_STATE_UNDEFINED	DInt	0	OMAC PackMLV2022 state Undefined
LPMLV2022_STATE_CLEARING	DInt	1	OMAC PackMLV2022 state Clearing
LPMLV2022_STATE_STOPPED	DInt	2	OMAC PackMLV2022 state Stopped
LPMLV2022_STATE_STARTING	DInt	3	OMAC PackMLV2022 state Starting
LPMLV2022_STATE_IDLE	DInt	4	OMAC PackMLV2022 state Idle
LPMLV2022_STATE_SUSPENDED	DInt	5	OMAC PackMLV2022 state Suspended
LPMLV2022_STATE_EXECUTE	DInt	6	OMAC PackMLV2022 state Execute
LPMLV2022_STATE_STOPPING	DInt	7	OMAC PackMLV2022 state Stopping
LPMLV2022_STATE_ABORTING	DInt	8	OMAC PackMLV2022 state Aborting
LPMLV2022_STATE_ABORTED	DInt	9	OMAC PackMLV2022 state Aborted
LPMLV2022_STATE_HOLDING	DInt	10	OMAC PackMLV2022 state Holding
LPMLV2022_STATE_HELD	DInt	11	OMAC PackMLV2022 state Held
LPMLV2022_STATE_UNHOLDING	DInt	12	OMAC PackMLV2022 state Unholding
LPMLV2022_STATE_SUSPENDING	DInt	13	OMAC PackMLV2022 state Suspending
LPMLV2022_STATE_UNSUSPENDING	DInt	14	OMAC PackMLV2022 state Unsuspending
LPMLV2022_STATE_RESETTING	DInt	15	OMAC PackMLV2022 state Resetting
LPMLV2022_STATE_COMPLETING	DInt	16	OMAC PackMLV2022 state Completing
LPMLV2022_STATE_COMPLETED	DInt	17	OMAC PackMLV2022 state Completed

Control commands

Table 2-47: Constants for control commands

Name	Data type	Value	Comment
LPMLV2022_CMD_UNDEFINED	DInt	0	OMAC PackMLV2022 control command Undefined
LPMLV2022_CMD_RESET	DInt	1	OMAC PackMLV2022 control command Reset
LPMLV2022_CMD_START	DInt	2	OMAC PackMLV2022 control command Start
LPMLV2022_CMD_STOP	DInt	3	OMAC PackMLV2022 control command Stop
LPMLV2022_CMD_HOLD	DInt	4	OMAC PackMLV2022 control command Hold
LPMLV2022_CMD_UNHOLD	DInt	5	OMAC PackMLV2022 control command Unhold
LPMLV2022_CMD_SUSPEND	DInt	6	OMAC PackMLV2022 control command Suspend
LPMLV2022_CMD_UNSUSPEND	DInt	7	OMAC PackMLV2022 control command Unsuspend
LPMLV2022_CMD_ABORT	DInt	8	OMAC PackMLV2022 control command Abort
LPMLV2022_CMD_CLEAR	DInt	9	OMAC PackMLV2022 control command Clear
LPMLV2022_CMD_COMPLETE	DInt	10	OMAC PackMLV2022 control command Complete

Array boundaries

Table 2-48: Constants for array boundaries

Name	Data type	Value	Comment
LPMLV2022_DIAG_BUFFER_UPPER_LIM	Int	7	Diagnostics buffer array upper boundary (0-based)
LPMLV2022_MODES_UPPER_LIM	Int	8	(Number of unit modes - 1) -> Array[0..LPMLV2022_MODES_UPPER_LIM]
LPMLV2022_STATES_UPPER_LIM	Int	17	(Number of states - 1) -> Array[0..LPMLV2022_STATES_UPPER_LIM]
LPMLV2022_MAX_MODES_UPPER_LIM	Int	31	(Maximum number of unit modes - 1) -> Array[0..LPMLV2022_MAX_MODES_UPPER_LIM]

The PLC tag table *LPMLV2022_PackTags_Constants* contains user constants for the PackTags array boundaries.

PackTags

Table 2-49: Constants for PackTags

Name	Data type	Value	Comment
LPMLV2022_COMMAND_PARAMETER_BOOL_UPPER_LIM	UInt	0	Size [0..UPPER_LIM] of structured Array of Unit/Machine Parameter information for values with BOOL data type
LPMLV2022_COMMAND_PARAMETER_DINT_UPPER_LIM	UInt	0	Size [0..UPPER_LIM] of structured Array of Unit/Machine Parameter information for values with DINT data type
LPMLV2022_COMMAND_PARAMETER_REAL_UPPER_LIM	UInt	0	Size [0..UPPER_LIM] of structured Array of Unit/Machine Parameter information for values with REAL data type
LPMLV2022_COMMAND_PARAMETER_LREAL_UPPER_LIM	UInt	0	Size [0..UPPER_LIM] of structured Array of Unit/Machine Parameter information for values with LREAL data type
LPMLV2022_COMMAND_PARAMETER_STRING_UPPER_LIM	UInt	0	Size [0..UPPER_LIM] of structured Array of Unit/Machine Parameter information for values with STRING data type
LPMLV2022_COMMAND_RECIPE_UPPER_LIM	UInt	0	Size [0..UPPER_LIM] of structured Array of Recipe Information
LPMLV2022_COMMAND_PROCESS_VARIABLES_PARAMETER_BOOL_UPPER_LIM	UInt	0	Size [0..UPPER_LIM] of structured Array of Recipe Process Variable information for values with BOOL data type
LPMLV2022_COMMAND_PROCESS_VARIABLES_PARAMETER_DINT_UPPER_LIM	UInt	0	Size [0..UPPER_LIM] of structured Array of Recipe Process Variable Information for values with DINT data type
LPMLV2022_COMMAND_PROCESS_VARIABLES_PARAMETER_REAL_UPPER_LIM	UInt	0	Size [0..UPPER_LIM] of structured Array of Recipe Process Variable information for values with REAL data type
LPMLV2022_COMMAND_PROCESS_VARIABLES_PARAMETER_LREAL_UPPER_LIM	UInt	0	Size [0..UPPER_LIM] of structured Array of Recipe Process Variable information for values with LREAL data type
LPMLV2022_COMMAND_PROCESS_VARIABLES_PARAMETER_STRING_UPPER_LIM	UInt	0	Size [0..UPPER_LIM] of structured Array of Recipe Process Variable information for values with STRING data type
LPMLV2022_COMMAND_INGREDIENTS_PARAMETER_BOOL_UPPER_LIM	UInt	0	Size [0..UPPER_LIM] of structured Array of Ingredient information for values with BOOL data type
LPMLV2022_COMMAND_INGREDIENTS_PARAMETER_DINT_UPPER_LIM	UInt	0	Size [0..UPPER_LIM] of structured Array of Ingredient Information for values with DINT data type
LPMLV2022_COMMAND_INGREDIENTS_PARAMETER_REAL_UPPER_LIM	UInt	0	Size [0..UPPER_LIM] of structured Array of Ingredient information for values with REAL data type
LPMLV2022_COMMAND_INGREDIENTS_PARAMETER_LREAL_UPPER_LIM	UInt	0	Size [0..UPPER_LIM] of structured Array of Ingredient information for values with LREAL data type
LPMLV2022_COMMAND_INGREDIENTS_PARAMETER_STRING_UPPER_LIM	UInt	0	Size [0..UPPER_LIM] of structured Array of Ingredient information for values with STRING data type

Name	Data type	Value	Comment
LPMLV2022_STATUS_PARAMETER_BOOL_UPPER_LIM	UInt	0	Size [0..UPPER_LIM] of structured Array of Unit/Machine Parameter information for values with BOOL data type
LPMLV2022_STATUS_PARAMETER_DINT_UPPER_LIM	UInt	0	Size [0..UPPER_LIM] of structured Array of Unit/Machine Parameter information for values with DINT data type
LPMLV2022_STATUS_PARAMETER_REAL_UPPER_LIM	UInt	0	Size [0..UPPER_LIM] of structured Array of Unit/Machine Parameter information for values with REAL data type
LPMLV2022_STATUS_PARAMETER_LREAL_UPPER_LIM	UInt	0	Size [0..UPPER_LIM] of structured Array of Unit/Machine Parameter information for values with LREAL data type
LPMLV2022_STATUS_PARAMETER_STRING_UPPER_LIM	UInt	0	Size [0..UPPER_LIM] of structured Array of Unit/Machine Parameter information for values with STRING data type
LPMLV2022_STATUS_RECIPE_UPPER_LIM	UInt	0	Size [0..UPPER_LIM] of structured Array of Recipe Information
LPMLV2022_STATUS_PROCESS_VARIABLES_PARAMETER_BOOL_UPPER_LIM	UInt	0	Size [0..UPPER_LIM] of structured Array of Recipe Process Variable information for values with BOOL data type
LPMLV2022_STATUS_PROCESS_VARIABLES_PARAMETER_DINT_UPPER_LIM	UInt	0	Size [0..UPPER_LIM] of structured Array of Recipe Process Variable Information for values with DINT data type
LPMLV2022_STATUS_PROCESS_VARIABLES_PARAMETER_REAL_UPPER_LIM	UInt	0	Size [0..UPPER_LIM] of structured Array of Recipe Process Variable information for values with REAL data type
LPMLV2022_STATUS_PROCESS_VARIABLES_PARAMETER_LREAL_UPPER_LIM	UInt	0	Size [0..UPPER_LIM] of structured Array of Recipe Process Variable information for values with LREAL data type
LPMLV2022_STATUS_PROCESS_VARIABLES_PARAMETER_STRING_UPPER_LIM	UInt	0	Size [0..UPPER_LIM] of structured Array of Recipe Process Variable information for values with STRING data type
LPMLV2022_STATUS_INGREDIENTS_PARAMETER_BOOL_UPPER_LIM	UInt	0	Size [0..UPPER_LIM] of structured Array of Ingredient information for values with BOOL data type
LPMLV2022_STATUS_INGREDIENTS_PARAMETER_DINT_UPPER_LIM	UInt	0	Size [0..UPPER_LIM] of structured Array of Ingredient Information for values with DINT data type
LPMLV2022_STATUS_INGREDIENTS_PARAMETER_REAL_UPPER_LIM	UInt	0	Size [0..UPPER_LIM] of structured Array of Ingredient information for values with REAL data type
LPMLV2022_STATUS_INGREDIENTS_PARAMETER_LREAL_UPPER_LIM	UInt	0	Size [0..UPPER_LIM] of structured Array of Ingredient information for values with LREAL data type
LPMLV2022_STATUS_INGREDIENTS_PARAMETER_STRING_UPPER_LIM	UInt	0	Size [0..UPPER_LIM] of structured Array of Ingredient information for values with STRING data type
LPMLV2022_STATUS_STACK_LIGHT_UPPER_LIM	UInt	0	Size [0..UPPER_LIM] of Indicator for the current stacklight status
LPMLV2022_ADMIN_PARAMETER_BOOL_UPPER_LIM	UInt	0	Size [0..UPPER_LIM] of structured Array of Unit/Machine Parameter information for values with BOOL data type
LPMLV2022_ADMIN_PARAMETER_DINT_UPPER_LIM	UInt	0	Size [0..UPPER_LIM] of structured Array of Unit/Machine Parameter information for values

Name	Data type	Value	Comment
			with DINT data type
LPMLV2022_ADMIN_PARAMETER_REAL_UPPER_LIM	UInt	0	Size [0..UPPER_LIM] of structured Array of Unit/Machine Parameter information for values with REAL data type
LPMLV2022_ADMIN_PARAMETER_LREAL_UPPER_LIM	UInt	0	Size [0..UPPER_LIM] of structured Array of Unit/Machine Parameter information for values with LREAL data type
LPMLV2022_ADMIN_PARAMETER_STRING_UPPER_LIM	UInt	0	Size [0..UPPER_LIM] of structured Array of Unit/Machine Parameter information for values with STRING data type
LPMLV2022_ADMIN_ALARM_UPPER_LIM	UInt	0	Size [0..UPPER_LIM] of Array for Machine Alarms
LPMLV2022_ADMIN_ALARM_HISTORY_UPPER_LIM	UInt	0	Size [0..UPPER_LIM] of Array for Machine Fault Number and Messaging History
LPMLV2022_ADMIN_WARNING_UPPER_LIM	UInt	0	Size [0..UPPER_LIM] of Machine warnings for general events that do not cause the machine to stop
LPMLV2022_ADMIN_CUMULATIVE_TIMES_UPPER_LIM	UInt	0	Size [0..UPPER_LIM] of structured Array of Timer Values
LPMLV2022_ADMIN_NUMBER_OF_MODES_UPPER_LIM	UInt	8	Size [0..UPPER_LIM] of structured Array of Mode and State Time Values for each Cumulative Time Tracker
LPMLV2022_ADMIN_NUMBER_OF_STATES_UPPER_LIM	UInt	17	Size [0..UPPER_LIM] of state Time Values for each State in each Mode
LPMLV2022_ADMIN_PRODUCT_DATA_UPPER_LIM	UInt	0	Size [0..UPPER_LIM] of structured array of Product Stream Data

3 Working with the Library

What will you find in this section?

This chapter consists of instructions for integrating the LPMLV2022 library into your STEP 7 project and instructions for using the library blocks.

3.1 Integrating the library into STEP 7

The table below lists the steps for integrating the LPMLV2022 library into your STEP 7 project. Subsequently, you can use the blocks of the LPMLV2022 library.

Note The following section assumes that a STEP 7 project exists.

Table 3-1: Integrating the library into STEP 7

No.	Action	Note
1.	Extract the library LPMLV2022_V1_x_x.zip to a local folder.	
2.	In TIA Portal select “Options” -> “Global libraries” -> “Open library…”.	
3.	Browse to the file LPMLV2022.al15_1. It can be found in the subfolder LPMLV2022 of the extracted zip file.	
4.	Open the global library in read-only mode.	
5.	The LPMLV2022 library is now available in the task card “Global libraries”	

3.2 Integrating the library blocks into STEP 7

The table below lists the steps for integrating the blocks of the LPMLV2022 library into your STEP 7 program.

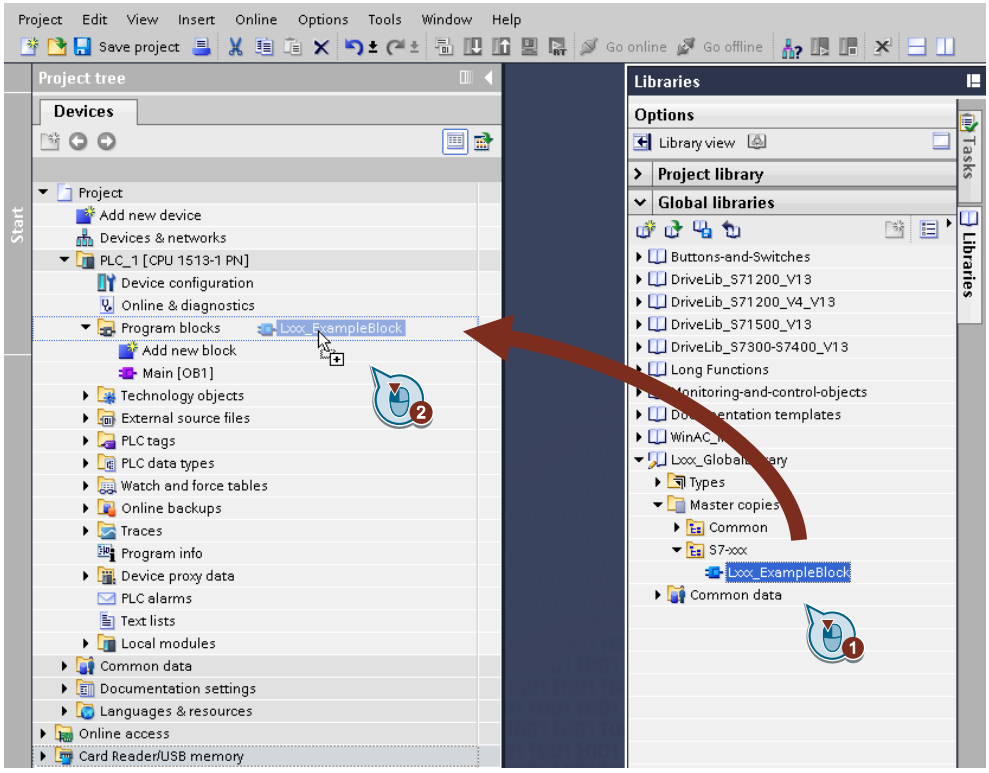


Table 3-2: Integrating the library blocks into STEP 7

No.	Action	Note
1.	Copy the folder *ModeStateManagement->LPMLV2022_Tags* with Drag & Drop into the folder “PLC tags” in the PLC.	
2.	Copy the folder *ModeStateManagement->LPMLV2022_Types* with Drag & Drop into the folder “PLC data types” in the PLC.	
3.	Copy the *LPMLV2022_UnitModeStateManager* FB in subfolder *ModeStateManagement->LPMLV2022_Blocks* with Drag & Drop into the “Program blocks” in the PLC. Alternatively copy the whole folder *ModeStateManagement->LPMLV2022_Blocks* via Drag & Drop into the “Program blocks” in the PLC. In this case also additional and optional blocks of LPMLV2022 library are available in the user program (e.g. *LPMLV2022_UnitModeStateTimes*, *LPMLV2022_ConfigureDisabledStates*).	
4.	Now the blocks can be configured and called in the user program.	
5.	Optional: copy the folder *PackTags->LPMLV2022_PackTags_Tags* with Drag & Drop into the folder “PLC tags” in the PLC.	
6.	Optional: copy the folder *PackTags->LPMLV2022_PackTags_Types* with Drag & Drop into the folder “PLC data types” in the PLC.	

4 Notes and Support

What will you find in this section?

This chapter provides further support in handling the described LPMLV2022 library.

5 Appendix

5.1 Service and support

Industry Online Support

Do you have any questions or need assistance?

Siemens Industry Online Support offers round the clock access to our entire service and support know-how and portfolio.

The Industry Online Support is the central address for information about our products, solutions and services.

Product information, manuals, downloads, FAQs, application examples and videos – all information is accessible with just a few mouse clicks:

support.industry.siemens.com

Technical Support

The Technical Support of Siemens Industry provides you fast and competent support regarding all technical queries with numerous tailor-made offers – ranging from basic support to individual support contracts.

Please send queries to Technical Support via Web form:

siemens.com/SupportRequest

SITRAIN – Digital Industry Academy

We support you with our globally available training courses for industry with practical experience, innovative learning methods and a concept that’s tailored to the customer’s specific needs.

For more information on our offered trainings and courses, as well as their locations and dates, refer to our web page:

siemens.com/sitrain

Service offer

Our range of services includes the following:

- Plant data services
- Spare parts services
- Repair services
- On-site and maintenance services
- Retrofitting and modernization services
- Service programs and contracts

You can find detailed information on our range of services in the service catalog web page:

support.industry.siemens.com/cs/sc

Industry Online Support app

You will receive optimum support wherever you are with the "Siemens Industry Online Support" app. The app is available for iOS and Android:

support.industry.siemens.com/cs/ww/en/sc/2067

5.2 Industry Mall

The Siemens Industry Mall is the platform on which the entire siemens Industry product portfolio is accessible. From the selection of products to the order and the delivery tracking, the Industry Mall enables the complete purchasing processing – directly and independently of time and location:
mall.industry.siemens.com

5.3 Application support

Siemens AG
Digital Industries
Factory Automation
Production Machines
DI FA PMA APC
Frauenauracher Str. 80
91056 Erlangen, Germany

mailto: tech.team.motioncontrol@siemens.com

5.4 Links and literature

Table 5-1

No.	Topic
\1\	Siemens Industry Online Support https://support.industry.siemens.com
\2\	Link to this entry page of this application example https://support.industry.siemens.com/cs/ww/en/view/109821198
\3\	OMAC https://www.omac.org
\4\	Siemens Packaging http://www.siemens.com/packaging
\5\	SIMATIC CPG Template https://support.industry.siemens.com/cs/ww/en/view/109475572

5.5 Change documentation

Table 5-2

Version	Date	Modifications
V1.0	12/2022	First version