OMAC PackML State Machine

The library "OMAC PackML State Machine" contains a state machine for packaging controls based on the OMAC PackML standard.

Product description

Licensing:

No license is required.

PackML, which stands for Packaging Machine Language, defines a common approach or machine language for automated machines. The primary goals are to encourage a common "look and feel" across a plant floor and to enable and encourage industry innovation.

This library contains an implementation of the OMAC PackML State Machine. An example project shows the usage of the components.

For more information about OMAC and PackML see www.omac.org.

The "PackML State Machine" defines machine states and state transitions for packaging controls. A set of machines states is called "unit mode". A unit mode can contain a maximum of 17 machine states. The following graphic shows all machine states with the corresponding state transitions.
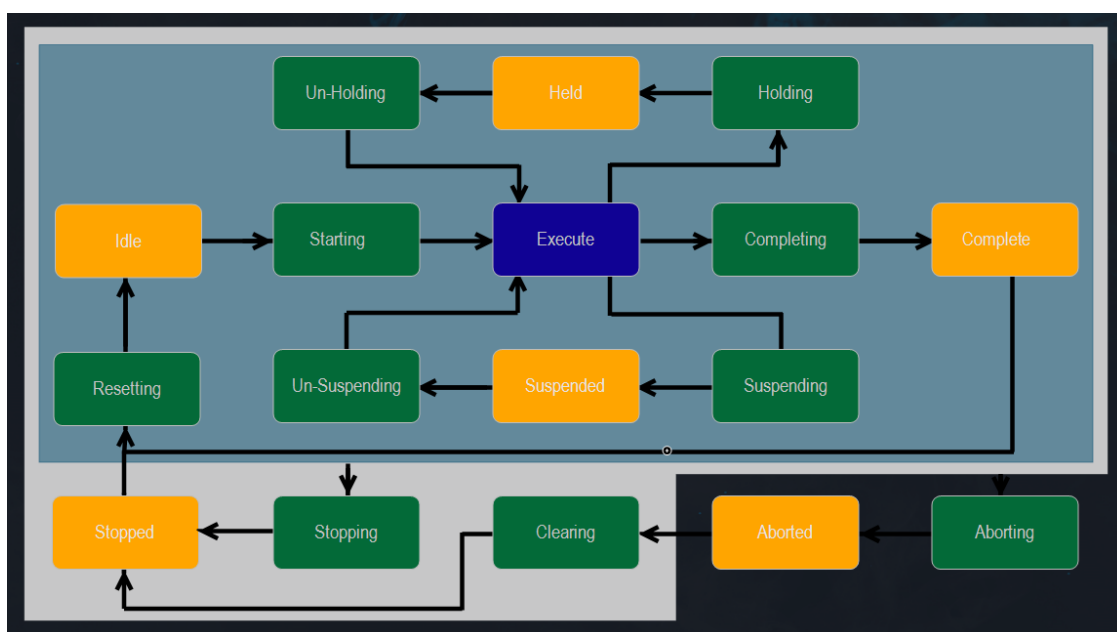
State transitions of PackML

A machine state defines the current condition of a machine. Three machine state types are proposed:

- Acting State (green), which was previously a Transient State, represents some processing activity, i.e. STARTING, HOLDING
- Wait State (olive green), which was previously a Quiescent, or Steady State, is used to identify that a machine has achieved a defined set of conditions, i.e. ABORTED, HELD, SUSPENDED, IDLE
- Dual State (blue) is defined as wait state that is causing the machine to appear as an acting state. There is currently only one dual state: EXECUTE

A state transition is executed by a corresponding command like "Start" or "Stop".

PackML allows to define custom unit modes. For example it is possible to create your own unit mode for setup or maintenance. The machine states of the defined unit modes are a subset of the 17 machine states.

There is only one active unit mode allowed, but it is possible to switch between unit modes. This is handled by a

component called unit mode manager. Before the unit mode will be switched, the unit mode manger checks if the current machine state allows leaving and if the target state allows entering.
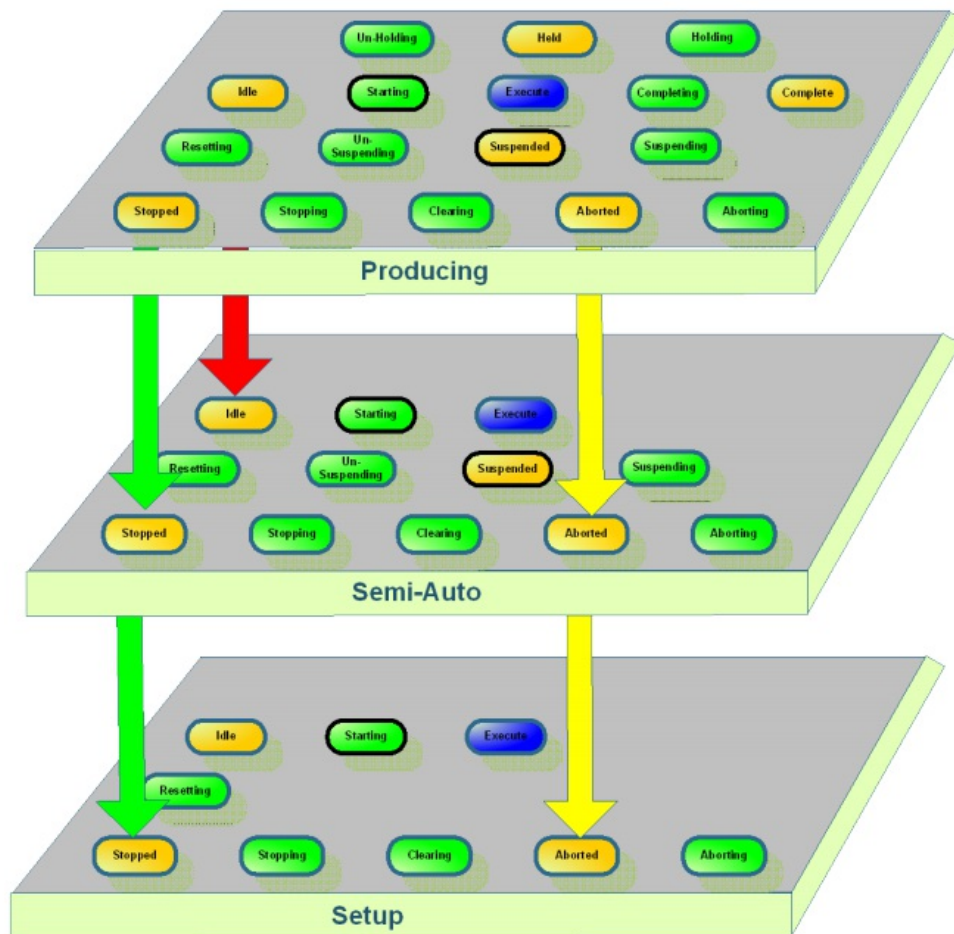
Unit mode transitions

Library OMAC_PackML_State_Machine

The package contains a library named OMAC_PackML_State_Machine, which includes all required components to build a state machine based on PackML.

Unit modes can be defined with the function block UnitMode. The desired machine states can be set by the input field dwSupportedStates. The permissions to switch between unit modes are handled by the input fields dwAllowsLeavingFromStates and dwAllowsEnteringIntoStates. The machine state can be changed by calling the corresponding method in folder Commands (Start, Stop, Abort etc.).

The function block UnitModeManager enables the switching of unit modes. All unit modes and the current state are managed by this function block.

Example project PackML_Example

The project PackML_Example in the installation folder of the package includes two example applications with implemented PackML state machines.

Application 'PackMLUnitModeManager':

The application shows how to define own unit modes. The program PLC_PROG defines three unit modes (Production, Semi Auto and Setup). At first the unit modes must be registered at the UnitModeMananger. After that the current machine state and the current unit mode can be requested via the UnitModeManager. The current machine state and current unit mode are displayed by visualization.
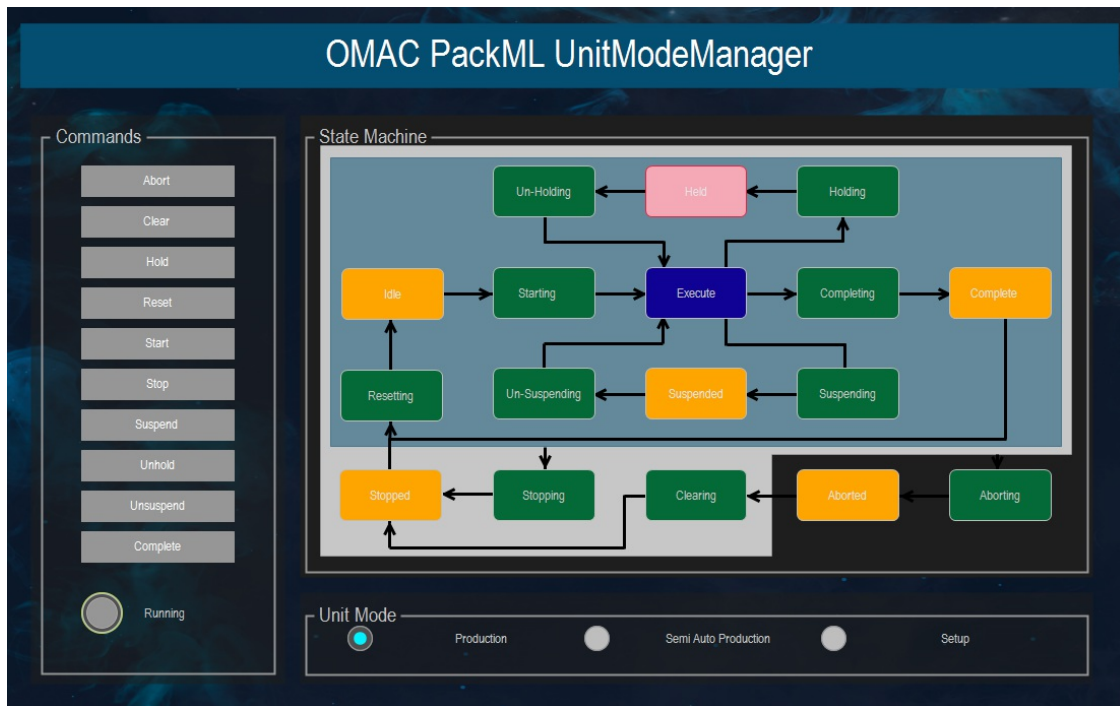
Visualization of the application PackMLUnitModeManager

Application 'PackMLAxisControl':

The application `PackMLAxisControl` shows how to set the execution state of a virtual SoftMotion axis via the PackML state machine. The function block `AxisControl` represents an axis, which is controlled according to its active unit mode.
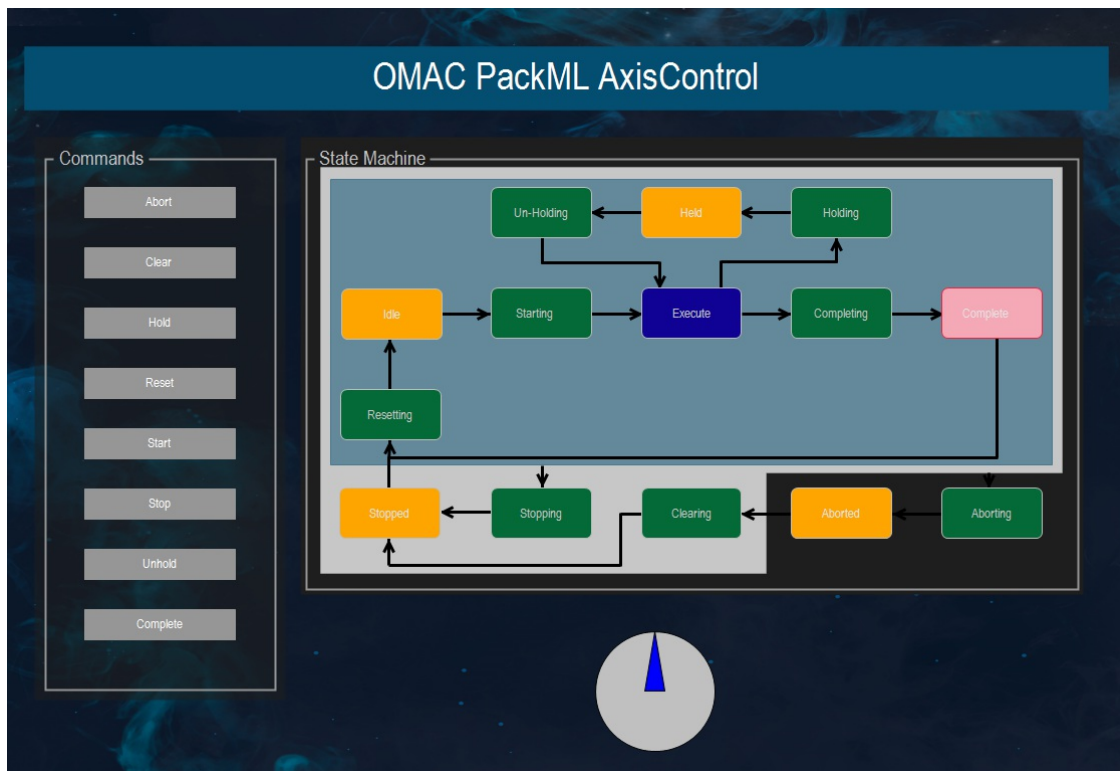
Visualization of the application PackMLAxisControl

General information

Supplier:

CODESYS GmbH
Memminger Strasse 151
87439 Kempten
Germany

Support:

https://support.codesys.com

Item:
OMAC PackML State Machine
Item number:
000060
Sales:

CODESYS Store
https://store.codesys.com

Included in delivery:
Package with library and example

System requirements and restrictions

Programming System	CODESYS Development System V3.5.10.0 or higher
Runtime System	CODESYS Control Version 3.5.10.0 or higher
Supported Platforms/ Devices	Note: Use the project "Device Reader" to find out the supported features of your device. "Device Reader" is available for free in the CODESYS Store.
Additional Requirements	-
Restrictions	-
Licensing	No license is required.
Required Accessories	-

Note: Not all CODESYS features are available in all territories. For more information on geographic restrictions, please contact sales@codesys.com.
Note: Technical specifications are subject to change. Errors and omissions excepted. The content of the current online version of this document applies.