Yaskawa America, Inc.

PackML_Toolbox_v203 Function Block Help Document

Help file for toolbox functions

Hunter Stofferahn, Doug Meyer

Ver. 1.0
2/23/2015

Table of Contents

Disclaimer 3

Glossary: 4

PackML Control Functions 4

UN_ModuleSummation 4

EM_ModuleSummation 7

CM_Control_Inputs 9

CM_Control_Outputs (FB & Worksheet) 11

PackML_State_Diagram 14

PackMLCommands_Init 20

PackMLModeStateTimes 21

UN_Control_Inputs (Worksheet) 22

Event Handling Functions 23

PackMLEvents_Init 23

EM_EventManager 25

UN_EventSummation 27

CM_Event 29

Servopack_Event 31

Controller_Event 33

Motion_Event 35

Servopack_Event_Lookup 37

Controller_Event_Lookup 38

Motion_Event_Lookup 39

Motion_Event_Lookup2 40

Disclaimer

The implementation provided in Yaskawa's PackML_Toolbox and PackML_Template is provided only as an example and is neither guaranteed to conform exactly to the PackML specification nor to be suitable for any particular application.

It is highly recommended that you refer to the PackML template to see the way the PackML toolbox functions were intended to be used.

Glossary:

UN – Unit Machine

The Unit Machine is a collection of related modules (mechanical and electrical assemblies) that carry out one or more processing activities as part of a Production Line or Process Cell.

EM – Equipment Module

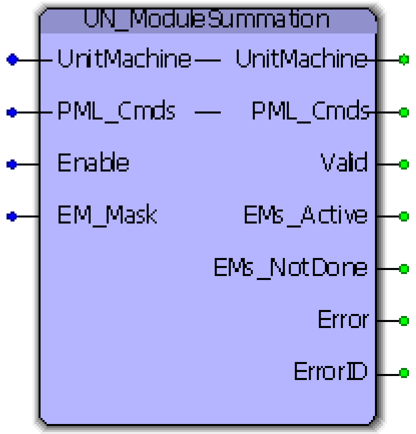
Functional group of modules that carry out a finite number of activities

CM – Control Module

The lowest level of control where a single function is executed

PackML Control Functions

UN_ModuleSummation



The UN_ModuleSummation function block rolls up the Equipment Module (EM) State Complete bits for all active and enabled EMs. The result is an overall PMLs State Complete bit that is transferred to the PackML_StateControl function.

Parameters *

Parameter		Data Type	Description
VAR_IN_OUT			
V	UnitMachine	UNitmachine_STRUCT	Structure containing all the information about the machine's current state and mode of operation for all EMs and CMs
V	PML_Cmds	PackML_Commands_ STRUCT	Structure that contains the current Unit mode of operation and the commands sent by PackML_StateMachine
VAR_INPUT			Default

B	Enable	BOOL	The function will continue to execute while the enable is held high	FALSE
V	EM_Mask	WORD	Mask to deactivate EMs. When an EM is deactivated, commands will not be sent down to the EM and all StateComplete bits will be set to TRUE. Each bit corresponds to the same number EM to deactivate. (Example: to deactivate EM_3, set EM_Mask.X3 =TRUE) 16 Equipment Modules are supported, numbered 0-15.	16#0000
VAR_OUTPUT				
B	Valid	BOOL	Indicates that the outputs of the function are valid	
B	EMs_Active	WORD	The list of active EMs. Same bit scheme as EM_Mask. (Example: if EMs_Active.X4 = TRUE then EM_4 is active)	
B	EMs_NotDone	WORD	A bit pattern of which Equipment Modules have not completed the transition task.	
B	Error	BOOL	Set high if error has occurred during the execution of the function block. This output is cleared when 'Execute' or 'Enable' goes low.	
B	ErrorID	UINT	If Error is true, this output provides the Error ID. This output is reset when 'Execute' or 'Enable' goes low.	

* Parameter Types for all Help Descriptions:

B - basic
V - vendor specific (structures)
E – extended (not needed for critical functionality)

Notes

- The user can identify those EMs stuck in transition by comparing the outputs EMs_Active and EMs_NotDone. These outputs are of WORD datatype, with each bit [x] representing the active status and transition status of EM[x]. Example: If the PackML command STOP was given, and EM[1] was enabled and active, but not finished stopping yet, the output of EMs_Active would be ...00011 while the output of EMs_NotDone would be ...00010. The user then knows that the process is stuck in EM[1]. The user would then go to the EM01_ModuleControl worksheet to further drill into the problem.

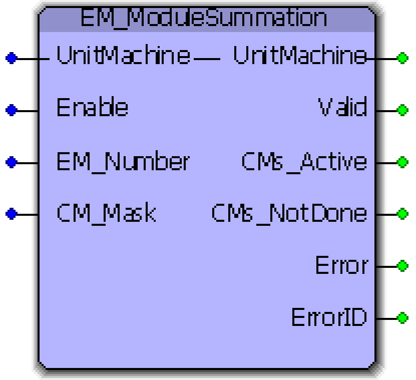
- See template documentation for further details on recommended usage.

Error Description

ErrorID	Meaning
0	No error
*	

***Error code details will be released in next version

EM_ModuleSummation



The EM_Module_Summation function block rolls up the Control Module State Complete bits for all active and enabled CMs. The result is an overall Equipment Module State Complete bit that is transferred to the UN_ModuleControl Worksheet.

Parameters

Parameter		Data Type	Description	
VAR_IN_OUT				
V	UnitMachine	UNitmachine_STRUCT	Structure containing all the information about the machines current state and mode of operation for all EMs and CMs	
VAR_INPUT				**Default**
B	Enable	BOOL	The function will continue to execute while the enable is held high	FALSE
V	EM_Number	INT	The EM number corresponding to the EM in which this FB is located	0
V	CM_Mask	WORD	Mask to deactivate CMs. When a CM is deactivated, commands will not be sent down to the CM and all StateComplete bits will be set to TRUE. Each bit corresponds to the same number CM to deactivate. (Example: to deactivate CM_3, set CM_Mask.X3 =TRUE) 16 Control Modules are supported for each Equipment Module, numbered 0-15.	16#0000

VAR_OUTPUT				
B	Valid	BOOL	Indicates that the outputs of the function are valid	
B	CMs_Active	WORD	The list of active EMs. Same bit scheme as CM_Mask. (Example: if CMs_Active.X4 = TRUE then CM_4 is active)	
B	CMs_NotDone	WORD	A compilation of which Control Modules have not completed the transition task.	
B	Error	BOOL	Set high if error has occurred during the execution of the function block. This output is cleared when 'Execute' or 'Enable' goes low.	
B	ErrorID	UINT	If Error is true, this output provides the Error ID. This output is reset when 'Execute' or 'Enable' goes low.	

Notes

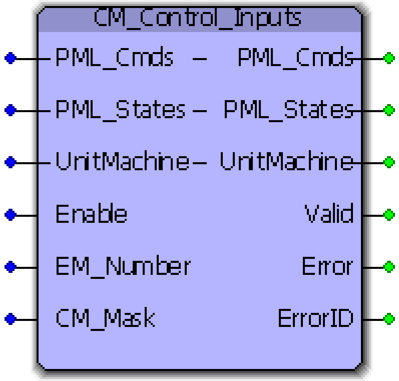
- The user can identify those CMs stuck in transition by comparing the outputs CMs_Active and CMs_NotDone. These outputs are of WORD datatype, with each bit [x] representing the active status and transition status of CM[x]. Example: If the PackML command STOP was given, and CM[1] was enabled and active, but not finished stopping yet, the output of CMs_Active would be ...00111 while the output of CMs_NotDone would be ...00010. The user then knows that the process is stuck in CM[1]. The user would then go to the EM00_CM01_Control_Outputs worksheet to further drill into the problem.
- See template documentation for further details on recommended usage.

Error Description

ErrorID	Meaning
0	No error
*	

***Error code details will be released in next version

CM_Control_Inputs



The CM_Control_Inputs function block passes the high level commands from the PackML_StateControl into each of the enabled and active Control Modules.

Parameters

Parameter		Data Type	Description	
VAR_IN_OUT				
V	PML_Cmds	PackML_Command_STRUCT	Structure that contains the current Unit mode of operation and the commands sent by the PackML_StateMachine	
V	PML_States	PackML_States_STRUCT	Structure containing information about the current state of the Unit machine	
V	UnitMachine	UNitmachine_STRUCT	Structure containing all the information about the machine's current state and mode of operation for all EMs and CMs	
VAR_INPUT				Default
B	Enable	BOOL	The function will continue to execute while the enable is held high	FALSE
V	EM_Number	INT	The EM number corresponding to the EM in which this FB is located	0
V	CM_Mask	WORD	Mask to deactivate CMs. When a CM is deactivated, commands will not be sent down to the CM. Each bit corresponds to the same number CM to deactivate. (Example: to deactivate CM_3, set CM_Mask.X3 =TRUE)	16#0000

			This mask needs to match the one provided to EM_ModuleSummation.	
VAR_OUTPUT				
B	Valid	BOOL	Indicates that the outputs of the function are valid	
B	Error	BOOL	Set high if error has occurred during the execution of the function block. This output is cleared when 'Execute' or 'Enable' goes low.	
B	ErrorID	UINT	If Error is true, this output provides the Error ID. This output is reset when 'Execute' or 'Enable' goes low.	

Notes

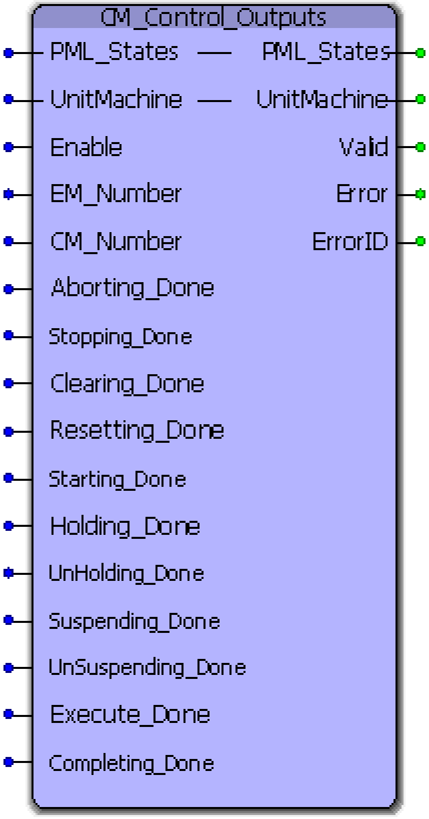
- See template documentation for further details on recommended usage.

Error Description

ErrorID	Meaning
0	No error
*	

*****Error code details will be released in next version

CM_Control_Outputs (FB & Worksheet)



The CM_Control_Outputs function block sets the State Complete bits for the control module to be passed up and assembled into the Equipment Module status in the EM00_ModuleControl worksheet.

Parameters

Parameter		Data Type	Description	
VAR_IN_OUT				
V	PML_States	PackML_States_STRUCT	Structure containing information about the current state of the Unit machine	
V	UnitMachine	UNitmachine_STRUCT	Structure containing all the information about the machine's current state and mode of operation for all EMs and CMs	
VAR_INPUT				Default
B	Enable	BOOL	The function will continue to execute while the enable is held high	FALSE
V	EM_Number	INT	The EM number corresponding to the EM in which this FB is	0

			located	
V	CM_Number	INT	The CM number corresponding to the CM in which this FB is located	0
B	Aborting_Done	BOOL	Setting this bit indicates that the current CM is done 'Aborting' and is ready to move to the next state	FALSE
B	Stopping_Done	BOOL	Setting this bit indicates that the current CM is done 'Stopping' and is ready to move to the next state	FALSE
B	Clearing_Done	BOOL	Setting this bit indicates that the current CM is done 'Clearing' and is ready to move to the next state	FALSE
B	Resetting_Done	BOOL	Setting this bit indicates that the current CM is done 'Resetting' and is ready to move to the next state	FALSE
B	Starting_Done	BOOL	Setting this bit indicates that the current CM is done 'Starting' and is ready to move to the next state	FALSE
B	Holding_Done	BOOL	Setting this bit indicates that the current CM is done 'Holding' and is ready to move to the next state	FALSE
B	UnHolding_Done	BOOL	Setting this bit indicates that the current CM is done 'UnHolding' and is ready to move to the next state	FALSE
B	Suspending_Done	BOOL	Setting this bit indicates that the current CM is done 'Suspending' and is ready to move to the next state	FALSE
B	UnSuspending_Done	BOOL	Setting this bit indicates that the current CM is done 'UnSuspending' and is ready to move to the next state	FALSE
B	Execute_Done	BOOL	Setting this bit indicates that the current CM is done 'Executing' and is ready to move to the next state	FALSE
B	Completing_Done	BOOL	Setting this bit indicates that the	FALSE

			current CM is done ‘Completing’ and is ready to move to the next state	
VAR_OUTPUT				
B	Valid	BOOL	Indicates that the outputs of the function are valid	
B	Error	BOOL	Set high if error has occurred during the execution of the function block. This output is cleared when 'Execute' or 'Enable' goes low.	
E	ErrorID	UINT	If Error is true, this output provides the Error ID. This output is reset when 'Execute' or 'Enable' goes low.	

Notes

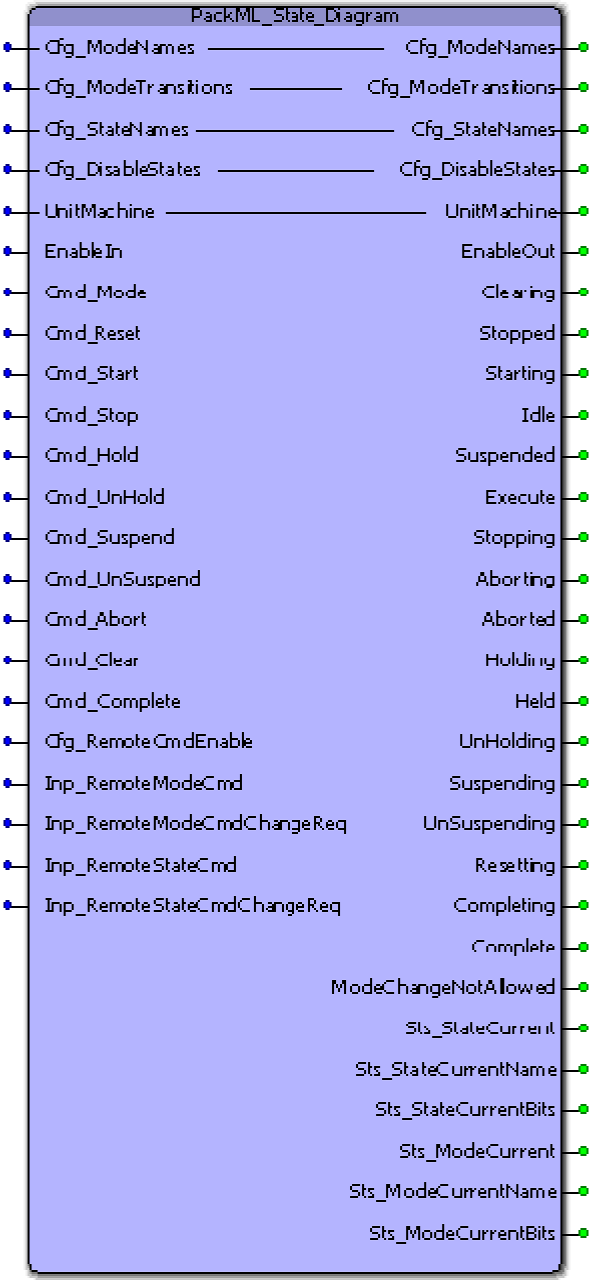
- Typically the individual *_Done Input coils are set in ladder code above this function block. The state of the coil depends on specific Control Module conditions. See template documentation for further details on recommended usage.

Error Description

ErrorID	Meaning
0	No error
*	

*Error code details will be released in next version

PackML_State_Diagram



The PackML_State_Diagram function block handles the operation of the state machine, including mode and state transitions, as defined in the OMAC PackML specification. This function block, when first enabled, initializes the machine to be in mode 3 (Manual Mode) and in the Stopped state.

Parameters

Parameter		Data Type	Description	
VAR_IN_OUT				
V	Cfg_ModeNames	STRING_Array32	An array of strings containing the names of the different Unit modes of operation	
V	Cfg_ModeTransitions	DINT_Array32	An array of acceptable mode transition states. Mode changes into the NEW MODE can only be performed at the chosen states. Each element in the array represents a mode, and each bit in the array element represents a state. (Ex. To allow Mode Transitions for Mode 1 at Aborted (bit 9), Stopped (bit 2), and Idle (bit 4) states 0000 0000 0000 0000 0000 0010 0001 0100 = 16#0000_0214 = DINT#532 = Cfg_ModeTransitions[1])	
V	Cfg_StateNames	STRING_Array18	An array of strings containing the names of all the PackML states	
V	Cfg_DisableStates	DINT_Array32	An array representing each mode and their states. Each mode can disable certain states.(Ex In Manual Mode (Mode 3) disable Holding(10), Held(11), UnHolding(12), Suspended(5), Suspending(13), UnSuspending(14),Completing(16), Complete(17) = 0000 0000 0000 0011 0111 1100 0010 0000 = 16#0003_7C20 = DINT#228384 = Cfg_DisableStates[3])	
V	UnitMachine	UNitmachine_STRUCT	Structure containing all the information about the machines current state of operation for all EMs and CMs	
VAR_INPUT				Default
B	EnableIn	BOOL	The function will continue to execute while the enable is held high.	FALSE
B	Cmd_Mode	DINT	The value of the new mode the machine will transition to if possible. If the input remains unchanged, the machine will stay in the same mode of operation	0
B	Cmd_Reset	BOOL	Setting this bit sends the 'Reset' command to all	FALSE

			enabled and active EMs if it is a legal transition from the current machine state, otherwise the command will be ignored	
B	Cmd_Start	BOOL	Setting this bit sends the ‘Start’ command to all enabled and active EMs if it is a legal transition from the current machine state, otherwise the command will be ignored	FALSE
B	Cmd_Stop	BOOL	Setting this bit sends the ‘Stop’ command to all enabled and active EMs if it is a legal transition from the current machine state, otherwise the command will be ignored	FALSE
B	Cmd_Hold	BOOL	Setting this bit sends the ‘Hold’ command to all enabled and active EMs if it is a legal transition from the current machine state, otherwise the command will be ignored	FALSE
B	Cmd_UnHold	BOOL	Setting this bit sends the ‘UnHold’ command to all enabled and active EMs if it is a legal transition from the current machine state, otherwise the command will be ignored	FALSE
B	Cmd_Suspend	BOOL	Setting this bit sends the ‘Suspend’ command to all enabled and active EMs if it is a legal transition from the current machine state, otherwise the command will be ignored	FALSE
B	Cmd_UnSuspend	BOOL	Setting this bit sends the ‘UnSuspend’ command to all enabled and active EMs if it is a legal transition from the current machine state, otherwise the command will be	FALSE

			ignored	
B	Cmd_Abort	BOOL	Setting this bit sends the ‘Abort’ command to all enabled and active EMs if it is a legal transition from the current machine state, otherwise the command will be ignored	FALSE
B	Cmd_Clear	BOOL	Setting this bit sends the ‘Clear’ command to all enabled and active EMs if it is a legal transition from the current machine state, otherwise the command will be ignored	FALSE
B	Cmd_Complete	BOOL	Setting this bit sends the ‘Complete’ command to all enabled and active EMs if it is a legal transition from the current machine state, otherwise the command will be ignored	FALSE
V	Cfg_RemoteModeCmd	DINT	The remotely requested mode number	0
B	Inp_RemoteModeCmd ChangeReq	BOOL	When this input is set, the machine will transition to the mode set by Cfg_RemoteModeCmd if it is a legal transition from the current state of the machine	FALSE
V	Inp_RemoteStateCmd	DINT	The remotely requested state number	0
B	Inp_RemoteStateCmd ChangeReq	BOOL	When this input is set, the machine will transition to the state set by Cfg_RemoteStateCmd if it is a legal transition from the current state of the machine	FALSE
VAR_OUTPUT				
B	EnableOut	BOOL	Indicates that the outputs of the function are valid	
B	Clearing	BOOL	When this bit is set, the machine is in the ‘Clearing’ state	

B	Stopped	BOOL	When this bit is set, the machine is in the ‘Stopped’ state
B	Starting	BOOL	When this bit is set, the machine is in the ‘Starting’ state
B	Idle	BOOL	When this bit is set, the machine is in the ‘Idle’ state
B	Suspended	BOOL	When this bit is set, the machine is in the ‘Suspended’ state
B	Execute	BOOL	When this bit is set, the machine is in the ‘Execute’ state
B	Stopping	BOOL	When this bit is set, the machine is in the ‘Stopping’ state
B	Aborting	BOOL	When this bit is set, the machine is in the ‘Aborting’ state
B	Aborted	BOOL	When this bit is set, the machine is in the ‘Aborted’ state
B	Holding	BOOL	When this bit is set, the machine is in the ‘Holding’ state
B	Held	BOOL	When this bit is set, the machine is in the ‘Held’ state
B	UnHolding	BOOL	When this bit is set, the machine is in the ‘UnHolding’ state
B	Suspending	BOOL	When this bit is set, the machine is in the ‘Suspending’ state
B	UnSuspending	BOOL	When this bit is set, the machine is in the ‘UnSuspending’ state
B	Resetting	BOOL	When this bit is set, the machine is in the ‘Resetting’ state
B	Completing	BOOL	When this bit is set, the machine is in the ‘Completing’ state
B	Complete	BOOL	When this bit is set, the machine is in the ‘Complete’ state
B	ModeChangeNotAllowed	BOOL	When this bit is set, the requested Mode change isn’t allowed and the machine will remain in the current mode and state.
V	Sts_StateCurrent	DINT	Number in decimal corresponding to the current state of the machine
V	Sts_StateCurrentName	STRING	The name of the current state of the machine
V	Sts_StateCurrentBits	DWORD	DWORD indicating the current state the machine is in (Ex. If Sts_StateCurrentBits[x] = 1, then the machine is in State x)

V	Sts_ModeCurrent	DINT	Number in decimal corresponding to the current mode of the machine
V	Sts_ModeCurrentName	STRING	The name of the current mode of the machine
V	Sts_ModeCurrentBits	DWORD	DWORD indicating the current mode of the machine (Ex. If Sts_ModeCurrentBits[x] = 1, then the machine is in Mode x)

Notes

- Should always be enabled when program is running to ensure proper operation of the state machine.
- See template documentation for further details on recommended usage.

Error Description

ErrorID	Meaning
0	No error
*	

*****Error code details will be released in next version

PackMLCommands_Init

The PackMLCommands_Init function block clears all commands and sets the machine to be in the stopped state.

Parameters

Parameter		Data Type	Description	
VAR_IN_OUT				
V	INP_PackMLCommands	PackML_Module_Commands_STRUCT	Structure containing the current state and commanded actions	
VAR_INPUT				Default
B	EN	BOOL	The function will continue to execute while the enable is held high	FALSE
VAR_OUTPUT				
B	ENO	BOOL	Indicates that the outputs of the function are valid	

Notes

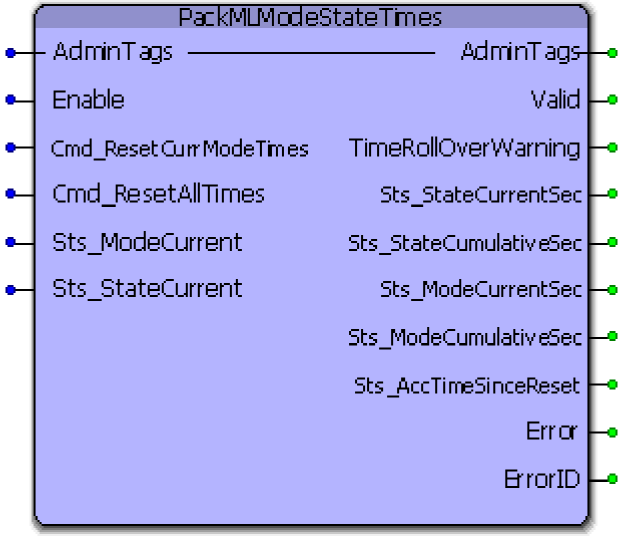
- Intended to be executed when initially entering the stopped state to clear all previous commands.
- Normally, this block is not necessary if proper interlocking is followed for PackML commands. However, using this block provides an added measure of assurance that no commands remain accidentally active.

Error Description

ErrorID	Meaning
0	No error
*	

***Error code details will be released in next version**

PackMLModeStateTimes



The PackMLModeStateTimes function block keeps track of the times spent in each mode and state of operation for the machine. Whenever the cpu is running, time is added to the current state of the current mode every second.

Parameters

Parameter		Data Type	Description	
VAR_IN_OUT				
V	AdminTags	PackTags_Admin_STRUCT	Structure containing alarm data from the machine.	
VAR_INPUT				Default
B	Enable	BOOL	The function will continue to execute while the enable is held high.	FALSE
B	Cmd_ResetCurrModeTimes	BOOL	When set, all time counting will be stalled and all of the times being counted for the Sts_ModeCurrent will be set to zero.	FALSE
B	Cmd_ResetAllTimes	BOOL	When set, all times being monitored will be reset to zero. Time counting will also be stalled as long as this input is held high	FALSE
V	Sts_ModeCurrent	DINT	The current mode of the machine	0

V	Sts_StateCurrent	DINT	The current state of the machine	0
VAR_OUTPUT				
B	Valid	BOOL	Indicates that the outputs of the function are valid	
B	TimeRollOverWarning	BOOL	A warning is sent when any of the time accumulators is approaching rolling over	
B	Sts_StateCurrentSec	DINT	Time (in seconds) spent in the current state	
V	Sts_StateCumulativeSec	StateCumulative Array	An array containing the times spent operating in different modes and states	
B	Sts_ModeCurrentSec	DINT	Time (in seconds) spent in the current mode	
V	Sts_ModeCumulativeSec	DINT_Array32	An array of times spent in each mode	
B	Sts_AccTimeSinceReset	DINT	Accumulated time since Cmd_ResetAllTimes was activated	
B	Error	BOOL	Set high if error has occurred during the execution of the function block. This output is cleared when 'Execute' or 'Enable' goes low.	
B	ErrorID	UINT	If Error is true, this output provides the Error ID. This output is reset when 'Execute' or 'Enable' goes low.	

Notes

- The output times are also internally stored in the AdminTags structure. Typically this is a Retained variable, and so all the time values will remain intact through power loss.
- See template documentation for further details on recommended usage.

Error Description

ErrorID	Meaning
0	No error
*	

*****Error code details will be released in next version

UN_Control_Inputs (Worksheet)

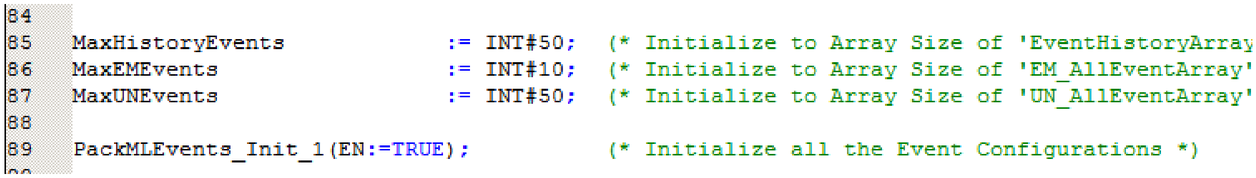
The UN_Control_Inputs worksheet is where the user customizes the state model for different Unit modes and where the use can tap into the commands to move to the next state. It is located in the PackML template and more information can be found in the template Documentation.

Event Handling Functions

The function in this section may or may not be used depending on the programmer's scheme for handling Faults. In the toolbox/template, Errors, Faults and Warnings are considered Events. While the functions help to capture and organize the active events, it is left up to the user to ultimately determine what action the machine should take (i.e. Abort, Stop, continue) and when the events should be reset.

PackMLEvents_Init

The PackMLEvents_Init function block is a freely editable block in which the user may create custom (user-defined) event descriptions. The events here are designed to be used with the CM_Event function block to feed the event occurrence up to be summarized at the EM and UN levels. This block is called from within the PackML_Initialize worksheet of the Initialize POU so the custom events are loaded on program start.



```

MaxHistoryEvents           := INT#50;  (* Initialize to Array Size of 'EventHistoryArray
MaxEMEvents                := INT#10;  (* Initialize to Array Size of 'EM_AllEventArray'
MaxUNEvents                := INT#50;  (* Initialize to Array Size of 'UN_AllEventArray'

PackMLEvents_Init_1(EN:=TRUE);         (* Initialize all the Event Configurations *)
```

Parameters

Parameter		Data Type	Description	
VAR_INPUT				Default
B	EN	BOOL	The function will continue to execute while the enable is held high	FALSE
VAR_OUTPUT				
B	ENO	BOOL	Indicates that the outputs of the function are valid	

Notes

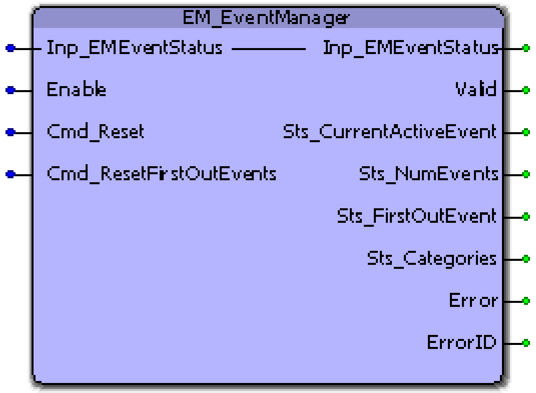
- The user may create any number of global Event Variables of datatype EventCfgArray
- The array size for each is [0..100]
- Ex:
  - Global Variable 'MyEvent' is created of EventCfgArray datatype
  - MyEvent[1].ID := UDINT#1001;
  - MyEvent[1].Message := 'My first custom event was triggered'
  - MtEvent[1].Description := 'The first defined custom event. Recovery from this one is easy';
  - MyEvent[1].Category := DWORD#16#0100; (* Cycle Stop *)

Error Description

ErrorID	Meaning
0	No error
*	

*Error code details will be released in next version

EM_EventManager



The EM_EventManager function block collects all the events that occurred in the enabled and active Control Modules within a particular Equipment Module. Active categories are OR'd together and outputs are provided for the number of active events and which event was the first to occur. Activating Cmd_Reset will clear all the stored events for the module.

Parameters

Parameter		Data Type	Description	
VAR_IN_OUT				
V	Inp_EMEventStatus	EM_EventStatus_STRUCT	A variable that contains all the active event information for an Equipment Module	
VAR_INPUT				Default
B	Enable	BOOL	The function will continue to execute while the enable is held high	FALSE
B	Cmd_Reset	BOOL	Clears the entire list of events except for the FirstOutEvent	
B	Cmd_ResetFirstOutEvents	BOOL	Clears only the FirstOutEvent. Note that both inputs must be set to clear all events completely	
V	Sts_ModeCurrent	DINT	The current mode of the machine	
V	Sts_StateCurretn	DINT	The current state of the machine	
VAR_OUTPUT				
B	Valid	BOOL	Indicates that the outputs of the function are valid	
B	Sts_CurrentActiveEvent	BOOL	Indicates if an event has occurred and been collected	
B	Sts_NumEvents	INT	A summary count of all currently active events	

V	Sts_FirstOutEvent	Event_STRUCT	The event that occurred first
V	Sts_Categories	DWORD	A combination of all the individual event categories. Each event Category is OR'd to create a single indicator of event severity
B	Error	BOOL	Set high if error has occurred during the execution of the function block. This output is cleared when 'Execute' or 'Enable' goes low.
B	ErrorID	UINT	If Error is true, this output provides the Error ID. This output is reset when 'Execute' or 'Enable' goes low.

Notes

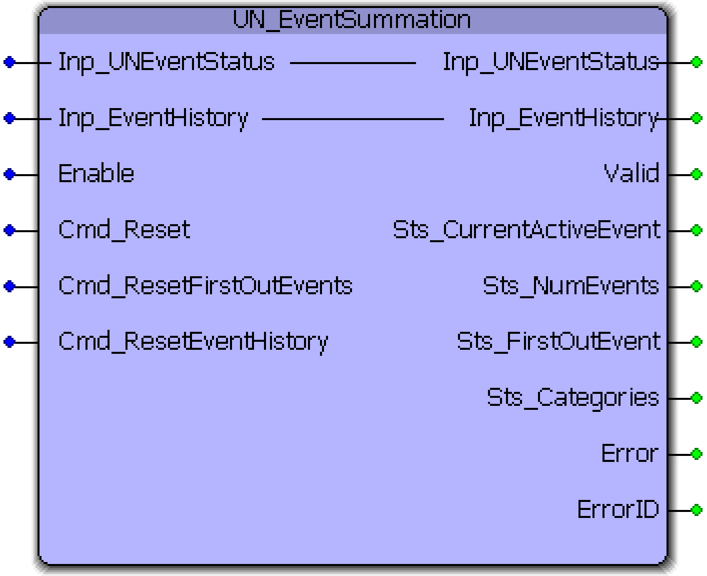
- See template documentation for further details on recommended usage.

Error Description

ErrorID	Meaning
0	No error
*	

*Error code details will be released in next version

UN_EventSummation



The UN_EventSummation function block summarizes all EM events to create a single listing of all active Events in the machine. All events are added to the Event History FIFO (max 50 events), typically stored in a retained variable.

Parameters

Parameter		Data Type	Description	
VAR_IN_OUT				
V	Inp_UNEventStatus	UN_EventSummation _STRUCT	A variable that contains all the active event information for a machine	
V	Inp_EventHistory	EventHistoryArray	FIFO to store information on the last 50 events	
VAR_INPUT				**Default**
B	Enable	BOOL	The function will continue to execute while the enable is held high	FALSE
V	Cmd_Reset	BOOL	Clears the entire list of events except for the FirstOutEvent	
V	Cmd_ResetFirstOut Events	BOOL	Clears only the FirstOutEvent. Note that both inputs must	

			be set to clear all events completely	
V	Cmd_ResetEventHistory	DINT	Clears the Event History FIFO	
VAR_OUTPUT				
B	Valid	BOOL	Indicates that the outputs of the function are valid	
B	Sts_CurrentActiveEvent	BOOL	Indicates if an event has occurred and been collected	
B	Sts_NumEvents	INT	A summary count of all currently active events on the machine	
V	Sts_FirstOutEvent	Event_STRUCT	The event that occurred first on the machine	
B	Sts_Categories	DWORD	A combination of all the individual event categories. Each event Category is OR’d to create a single indicator of event severity	
B	Error	BOOL	Set high if error has occurred during the execution of the function block. This output is cleared when 'Execute' or 'Enable' goes low.	
B	ErrorID	UINT	If Error is true, this output provides the Error ID. This output is reset when 'Execute' or 'Enable' goes low.	

Notes

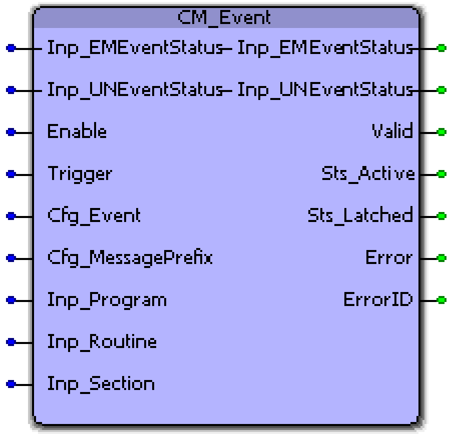
- See template documentation for further details on recommended usage.

Error Description

ErrorID	Meaning
0	No error
*	

*****Error code details will be released in next version

CM_Event



The CM_Event function block captures User-Defined Events that occur in a Control Module to be fed up to the EM_EventSummation block. Custom User Events are configured in the PackMLEvents_Init function block. Additional inputs are provided for the user to attach descriptive strings that can further identify where the event originated. As events are triggered, the time of the event is time stamped. When the event is cleared, another timestamp records the acknowledged time.

Parameters

Parameter		Data Type	Description	
VAR_IN_OUT				
V	Inp_EMEventSttus	EM_EventStatus_STRUCT		
V	Inp_UNEventStatus	UN_EventSummation_ STRUCT		
VAR_INPUT				**Default**
B	Enable	BOOL	The function will continue to execute while the enable is held high	FALSE
B	TRIGGER	BOOL	The condition that triggers the event to be captured	
V	Cfg_Event	EventCfg_STRUCT	The description of the event to be captured. Events are predefined, so this input is an element of the pre-defined array of events.	
V	Cfg_MessagePrefix	STRING	Prefix to be sent with error message	

V	Inp_Program	STRING	Program where CM_Event block is located	
V	Inp_Routine	STRING	Routine where CM_Event block is located	
V	Inp_Section	STRING	Section where CM_Event block is located	
VAR_OUTPUT				
B	Valid	BOOL	Indicates that the outputs of the function are valid	
B	Sts_Active	BOOL	Indicates that the trigger condition is active and that an event has been captured. If the trigger condition goes to FALSE, this output will turn off.	
B	Sts_Latched	BOOL	Indicates that an event is still present and has not yet been cleared. If trigger condition goes to FALSE, this output will remain ON. It will turn off when the events is cleared at the EM level by EM_EventSummation.	
B	Error	BOOL	Set high if error has occurred during the execution of the function block. This output is cleared when 'Execute' or 'Enable' goes low.	
B	ErrorID	UINT	If Error is true, this output provides the Error ID. This output is reset when 'Execute' or 'Enable' goes low.	

Notes

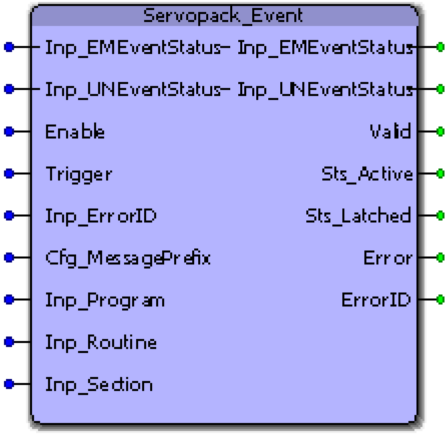
- See template documentation for further details on recommended usage.

Error Description

ErrorID	Meaning
0	No error
*	

*Error code details will be released in next version

Servopack_Event



The Servopack_Event function block is designed specifically for capturing Yaskawa SERVOPACK alarms. It features an internal lookup that provides alarm information based on the error code. Typically, an Axis_Control function block from the Yaskawa PLCopen Toolbox triggers the Servopack_Event block.

Parameters

Parameter		Data Type	Description	
VAR_IN_OUT				
V	Inp_EMEventStatus	EM_EventStatus_STRUCT		
V	Inp_UNEventStatus	UN_EventSummation_STRUCT		
VAR_INPUT				**Default**
B	Enable	BOOL	The function will continue to execute while the enable is held high	FALSE
B	Trigger	BOOL	Drive alarm or warning output from Axis_Control block indicating a servo alarm has occurred	
V	Inp_ErrorID	UINT	Drive AlarmID or WarningID from the Axis_Control block	
V	Cfg_MessagePrefix	STRING	Prefix to be sent with error message	
V	Inp_Program	STRING	Program where Servopack _Event block is located	
V	Inp_Routine	STRING	Routine where Servopack _Event block is located	

V	Inp_Section	STRING	Section where ServoPack _Event block is located	
VAR_OUTPUT				
B	Valid	BOOL	Indicates that the outputs of the function are valid	
B	Sts_Active	BOOL	Indicates that the trigger condition is active and that an event has been captured. If the trigger condition goes to FALSE, this output will turn off.	
B	Sts_Latched	BOOL	Indicates that an event is still present and has not yet been cleared. If trigger condition goes to FALSE, this output will remain ON. It will turn off when the events is cleared at the EM level by EM_EventSummation.	
B	Error	BOOL	Set high if error has occurred during the execution of the function block. This output is cleared when 'Execute' or 'Enable' goes low.	
B	ErrorID	UINT	If Error is true, this output provides the Error ID. This output is reset when 'Execute' or 'Enable' goes low.	

Notes

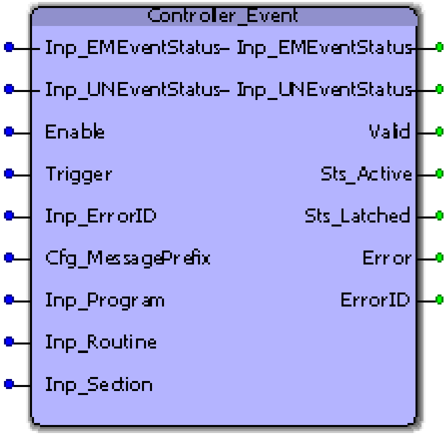
- See template documentation for further details on recommended usage.

Error Description

ErrorID	Meaning
0	No error
*	

*Error code details will be released in next version

Controller_Event



The Controller_Event function block is designed specifically for capturing Yaskawa MPiec Controller alarms. It features an internal lookup that provides alarm information based on the error code. Typically, an Axis_Control function block from the Yaskawa PLCopen Toolbox triggers the Controller_Event block.

Parameters

Parameter		Data Type	Description	
VAR_IN_OUT				
V	Inp_EMEventStatus	EM_EventStatus_STRUCT		
V	Inp_UNEventStatus	UN_EventSummation_STRUCT		
VAR_INPUT				**Default**
B	Enable	BOOL	The function will continue to execute while the enable is held high	FALSE
B	Trigger	BOOL	Controller alarm output from Axis_Control block indicating a controller alarm has occurred	
V	Inp_ErrorID	UINT	Controller AlarmID from the Axis_Control block	
V	Cfg_MessagePrefix	STRING	Prefix to be sent with error message	
V	Inp_Program	STRING	Program where Controller_Event block is located	
V	Inp_Routine	STRING	Routine where Controller	

			_Event block is located	
V	Inp_Section	STRING	Section where Controller _Event block is located	
VAR_OUTPUT				
B	Valid	BOOL	Indicates that the outputs of the function are valid	
B	Sts_Active	BOOL	Indicates that an event is still present and has not yet been cleared. If trigger condition goes to FALSE, this output will remain ON. It will turn off when the events is cleared at the EM level by EM_EventSummation.	
B	Sts_Latched	BOOL	Set high if error has occurred during the execution of the function block. This output is cleared when 'Execute' or 'Enable' goes low.	
B	Error	BOOL	Set high if error has occurred during the execution of the function block. This output is cleared when 'Execute' or 'Enable' goes low.	
B	ErrorID	UINT	If Error is true, this output provides the Error ID. This output is reset when 'Execute' or 'Enable' goes low.	

Notes

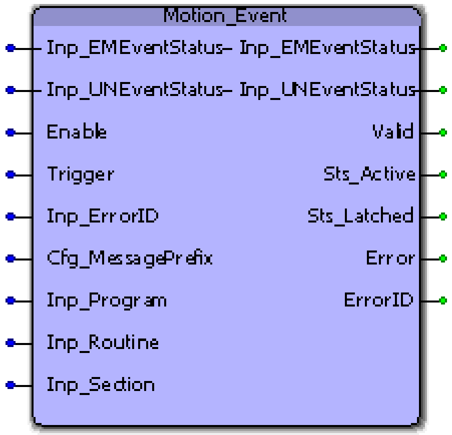
- See template documentation for further details on recommended usage.

Error Description

ErrorID	Meaning
0	No error
*	

*Error code details will be released in next version

Motion_Event



The Motion_Event function block is designed specifically for capturing Yaskawa Motion Function Block alarms. It features an internal lookup that provides alarm information based on the error code. Typically, a function block from the Yaskawa firmware library triggers the Motion_Event block.

Parameters

Parameter		Data Type	Description	
VAR_IN_OUT				
V	Inp_EMEventStatus	EM_EventStatus_STRUCT		
V	Inp_UNEventStatus	UN_EventSummation_STRUCT		
VAR_INPUT				**Default**
B	Enable	BOOL	The function will continue to execute while the enable is held high	FALSE
B	Trigger	BOOL	ERROR output from a firmware function block indicating an alarm has occurred	
V	Inp_ErrorID	UINT	ERROR_ID output from a firmware function block describing which alarm occurred	
V	Cfg_MessagePrefix	STRING	Prefix to be sent with error message	
V	Inp_Program	STRING	Program where Motion _Event block is located	

V	Inp_Routine	STRING	Routine where Motion _Event block is located	
V	Inp_Section	STRING	Section where Motion _Event block is located	
VAR_OUTPUT				
B	Valid	BOOL	Indicates that the outputs of the function are valid	
B	Sts_Active	BOOL	Indicates that an event is still present and has not yet been cleared. If trigger condition goes to FALSE, this output will remain ON. It will turn off when the events is cleared at the EM level by EM_EventSummation.	
B	Sts_Latched	BOOL	Set high if error has occurred during the execution of the function block. This output is cleared when 'Execute' or 'Enable' goes low.	
B	Error	BOOL	Set high if error has occurred during the execution of the function block. This output is cleared when 'Execute' or 'Enable' goes low.	
B	ErrorID	UINT	If Error is true, this output provides the Error ID. This output is reset when 'Execute' or 'Enable' goes low.	

Notes

- See template documentation for further details on recommended usage.

Error Description

ErrorID	Meaning
0	No error
*	

*Error code details will be released in next version

Servopack_Event_Lookup

Supporting function block for Servopack_Event

Parameters

Parameter		Data Type	Description
VAR_IN_OUT			
V	Lookup	EventCfg_STRUCT	Structure containing the alarm description, etc for the requested Error_ID
VAR_INPUT			
B	Fault_Number	UINT	Error_ID code [in decimal] to be looked up.

Notes

- This function is called internally by the Servopack_Event function block but can also be used independently in the program if the built-in event handling is not desired.

Error Description

ErrorID	Meaning
0	No error
*	

*Error code details will be released in next version

Controller_Event_Lookup

Supporting function block for Controller_Event

Parameters

Parameter		Data Type	Description
VAR_IN_OUT			
V	Lookup	EventCfg_STRUCT	Structure containing the alarm description, etc for the requested Error_ID
VAR_INPUT			
B	Fault_Number	UDINT	Error_ID code [in decimal] to be looked up.

Notes

- This function is called internally by the Controller_Event function block but can also be used independently in the program if the built-in event handling is not desired.

Error Description

ErrorID	Meaning
0	No error
*	

*****Error code details will be released in next version

Motion_Event_Lookup

Supporting function block for Motion_Event

Parameters

Parameter		Data Type	Description
VAR_IN_OUT			
V	Lookup	EventCfg_STRUCT	Structure containing the alarm description, etc for the requested Error_ID
VAR_INPUT			
B	Fault_Number	UINT	Error_ID code [in decimal] to be looked up.

Notes

- This function is called internally by the Motion_Event function block but can also be used independently in the program if the built-in event handling is not desired.

Error Description

ErrorID	Meaning
0	No error
*	

***Error code details will be released in next version

Motion_Event_Lookup2

Supporting function block for Motion_Event. This function block is called from within Motion_Event_Lookup to handle Fault numbers above 32767

Parameters

Parameter		Data Type	Description
VAR_IN_OUT			
V	Lookup	EventCfg_STRUCT	Structure containing the alarm description, etc for the requested Error_ID
VAR_INPUT			
B	Fault_Number	UINT	Error_ID code [in decimal] to be looked up.

Notes

- This function is called internally by the Motion_Event_Lookup function block. It is best not to use this block independently.

Error Description

ErrorID	Meaning
0	No error
*	

*****Error code details will be released in next version