

EcoStruxure Machine Expert

PackML

Guide de la bibliothèque

EIO0000002810.01
12/2023

Mentions légales

Les informations fournies dans ce document contiennent des descriptions générales, des caractéristiques techniques et/ou des recommandations concernant des produits/solutions.

Ce document n'est pas destiné à remplacer une étude détaillée ou un plan de développement ou de représentation opérationnel et propre au site. Il ne doit pas être utilisé pour déterminer l'adéquation ou la fiabilité des produits/solutions pour des applications utilisateur spécifiques. Il incombe à chaque utilisateur individuel d'effectuer, ou de faire effectuer par un professionnel de son choix (intégrateur, spécificateur ou équivalent), l'analyse de risques exhaustive appropriée ainsi que l'évaluation et les tests des produits/solutions par rapport à l'application ou l'utilisation particulière envisagée.

La marque Schneider Electric et toutes les marques de commerce de Schneider Electric SE et de ses filiales mentionnées dans ce document sont la propriété de Schneider Electric SE ou de ses filiales. Toutes les autres marques peuvent être des marques de commerce de leurs propriétaires respectifs.

Ce document et son contenu sont protégés par les lois sur la propriété intellectuelle applicables et sont fournis à titre d'information uniquement. Aucune partie de ce document ne peut être reproduite ou transmise sous quelque forme ou par quelque moyen que ce soit (électronique, mécanique, photocopie, enregistrement ou autre), à quelque fin que ce soit, sans l'autorisation écrite préalable de Schneider Electric.

Schneider Electric n'accorde aucun droit ni aucune licence d'utilisation commerciale de ce document ou de son contenu, sauf dans le cadre d'une licence non exclusive et personnelle, pour le consulter tel quel.

Schneider Electric se réserve le droit d'apporter à tout moment des modifications ou des mises à jour relatives au contenu de ce document ou à son format, sans préavis.

Dans la mesure permise par la loi applicable, Schneider Electric et ses filiales déclinent toute responsabilité en cas d'erreurs ou d'omissions dans le contenu informatif du présent document ou pour toute conséquence résultant de l'utilisation des informations qu'il contient.

Table des matières

Consignes de sécurité	7
QUALIFICATION DU PERSONNEL	7
INSTRUCTIONS D'UTILISATION	8
Avant de commencer	8
Démarrage et test	9
Fonctionnement et réglages	10
A propos de ce manuel	11
Informations générales	19
Présentation de la bibliothèque	20
Informations générales	20
Modes de fonctionnement	22
PackTags	25
Concept de diagnostic	26
Concept de diagnostic	26
Entrées et sorties communes	28
Comportement des blocs fonction avec entrée <i>i_xEnable</i>	28
Types d'unités de données	29
Énumérations	30
<i>ET_Cmd</i>	30
<i>ET_Modes</i>	31
<i>ET_Result</i>	31
<i>ET_StateModelDefinition</i>	33
<i>ET_States</i>	34
Structures	36
Structures Principales PackTags	36
<i>ST_Administration</i>	36
<i>ST_Command</i>	38
<i>ST_Status</i>	39
Autres Structures Principales	40
<i>ST_InitAlarm</i>	40
<i>ST_UnitModeDefinition</i>	41
<i>ST_VisInterface</i>	42
Sous-Structures	45
<i>ST_Alarm</i>	45
<i>ST_CountDescrip</i>	46
<i>ST_Cumulative_Times</i>	47
<i>ST_Descriptor</i>	47
<i>ST_EquipmentInterlock</i>	48
<i>ST_Ingredient</i>	49
<i>ST_Ingredients</i>	49
<i>ST_Interface</i>	50
<i>ST_ModeState_Times</i>	50
<i>ST_Parameter_DINT</i>	51
<i>ST_Parameter_LREAL</i>	52
<i>ST_Parameter_REAL</i>	52
<i>ST_Parameter_STRING</i>	53
<i>ST_ProcessVariables</i>	54
<i>ST_Product</i>	54

<i>ST_Product_Data</i>	55
<i>ST_Recipe</i>	56
<i>ST_StateInfo</i>	57
<i>ST_Timestamp</i>	57
Alias - <i>DateTimeArray</i>	59
<i>DateTimeArray</i>	59
Variables globales	60
Liste des constantes globales	61
GCL	61
Liste des paramètres globaux	62
GPL	62
Interfaces	65
<i>IF_UnitMode</i>	66
<i>IF_UnitMode</i> - Informations générales	66
<i>IF_StateCommands</i>	67
<i>IF_StateCommands</i> - Informations générales	67
<i>CmdReset</i> (méthode)	67
<i>CmdStart</i> (méthode)	68
<i>CmdStop</i> (méthode)	69
<i>CmdHold</i> (méthode)	69
<i>CmdUnHold</i> (méthode)	70
<i>CmdSuspend</i> (méthode)	71
<i>CmdUnSuspend</i> (méthode)	71
<i>CmdAbort</i> (méthode)	72
<i>CmdClear</i> (méthode)	73
<i>StateComplete</i> (méthode)	73
<i>IF_StateModelHandler</i>	75
<i>IF_StateModelHandler</i> - Informations générales	75
Unités d'organisation de programme (POU)	76
Blocs fonction	77
<i>FB_DataManagement</i>	77
<i>FB_ModeManager</i>	79
<i>FB_UnitModeManager2</i>	82
<i>FB_UnitModeManager2</i> - Informations générales	82
Modèle d'état de base PackML	83
Propriétés	84
<i>DefineUnitMode</i> (méthode)	84
<i>DefineUnitModeWithHandler</i> (méthode)	86
<i>RegisterLoggerPoint</i> (méthode)	87
<i>SetApplicationLoggerLogLevel</i> (méthode)	88
<i>ExecuteCurrentState</i> (méthode)	89
<i>CmdChangeUnitMode</i> (méthode)	90
<i>RemoveUnitMode</i> (méthode)	91
<i>GetDefinedUnitModes</i> (méthode)	91
<i>Command</i> (méthode)	92
<i>FB_StateModelHandlerBase</i>	93
<i>FB_StateModelHandlerBase</i> - Informations générales	93
<i>Clearing</i> (méthode)	93
<i>Stopped</i> (méthode)	94
<i>Starting</i> (méthode)	94

<i>Idle</i> (méthode)	95
<i>Suspended</i> (méthode)	95
<i>Execute</i> (méthode)	96
<i>Stopping</i> (méthode)	96
<i>Aborting</i> (méthode)	97
<i>Aborted</i> (méthode)	97
<i>Holding</i> (méthode)	98
<i>Held</i> (méthode)	98
<i>Restarting</i> (méthode)	99
<i>Suspending</i> (méthode)	99
<i>Unsuspending</i> (méthode)	100
<i>Resetting</i> (méthode)	100
<i>Completing</i> (méthode)	101
<i>Complete</i> (méthode)	101
<i>FB_VisController</i>	102
Fonctions	104
<i>FC_CheckCmd</i>	104
<i>FC_CheckCmd2</i>	105
<i>FC_EtResultToString</i>	106
<i>FC_GetDateTimeAsArray</i>	107
<i>FC_InitStateModelChangeStates</i>	107
<i>FC_InitStateModelExistingStates</i>	110
<i>FC_SetAlarm</i>	112
<i>FC_SetWarning</i>	113
Visualisation	115
BackgroundFrames	116
<i>FR_<BackgroundFrame></i>	116
States Frames	118
<i>FR_<état></i>	118
Cadres de visualisation	119
Généralités	119
<i>FR_Alarm</i>	120
<i>FR_AlarmHistory</i>	120
<i>FR_CurrentModeAndStateTime</i>	121
<i>FR_DynStateModel</i>	122
<i>FR_DynStateModel_2022</i>	123
<i>FR_ModeAndStateTime</i>	124
<i>FR_ProdConsumedCount</i>	125
<i>FR_ProdDefectiveCount</i>	126
<i>FR_ProdProcessedCount</i>	127
<i>FR_Warning</i>	128
<i>FR_StopReason</i>	128
<i>FR_StatesDisabled</i>	129
Glossaire	131
Index	132

Consignes de sécurité

Informations importantes

Lisez attentivement ces instructions et examinez le matériel pour vous familiariser avec l'appareil avant de tenter de l'installer, de le faire fonctionner, de le réparer ou d'assurer sa maintenance. Les messages spéciaux suivants que vous trouverez dans cette documentation ou sur l'appareil ont pour but de vous mettre en garde contre des risques potentiels ou d'attirer votre attention sur des informations qui clarifient ou simplifient une procédure.



La présence de ce symbole sur une étiquette "Danger" ou "Avertissement" signale un risque d'électrocution qui provoquera des blessures physiques en cas de non-respect des consignes de sécurité.



Ce symbole est le symbole d'alerte de sécurité. Il vous avertit d'un risque de blessures corporelles. Respectez scrupuleusement les consignes de sécurité associées à ce symbole pour éviter de vous blesser ou de mettre votre vie en danger.

DANGER

DANGER signale un risque qui, en cas de non-respect des consignes de sécurité, **provoque** la mort ou des blessures graves.

AVERTISSEMENT

AVERTISSEMENT signale un risque qui, en cas de non-respect des consignes de sécurité, **peut provoquer** la mort ou des blessures graves.

ATTENTION

ATTENTION signale un risque qui, en cas de non-respect des consignes de sécurité, **peut provoquer** des blessures légères ou moyennement graves.

AVIS

AVIS indique des pratiques n'entraînant pas de risques corporels.

Remarque Importante

L'installation, l'utilisation, la réparation et la maintenance des équipements électriques doivent être assurées par du personnel qualifié uniquement. Schneider Electric décline toute responsabilité quant aux conséquences de l'utilisation de ce matériel.

Une personne qualifiée est une personne disposant de compétences et de connaissances dans le domaine de la construction, du fonctionnement et de l'installation des équipements électriques, et ayant suivi une formation en sécurité leur permettant d'identifier et d'éviter les risques encourus.

QUALIFICATION DU PERSONNEL

Une personne qualifiée est une personne qui a les qualifications suivantes :

- Compétences et connaissances liées à la construction et à l'exploitation d'équipements électriques et à l'installation.
- Connaissance de l'exploitation des fonctionnalités de la machine dans l'implémentation du logiciel.

- A reçu une formation en sécurité permettant de reconnaître et d'éviter les dangers potentiels.

La personne qualifiée doit être capable de détecter d'éventuels dangers qui pourraient découler du paramétrage, de modifications des valeurs de paramétrage et plus généralement des équipements mécaniques, électriques ou électroniques. La personne qualifiée doit connaître les normes, dispositions et réglementations liées à la prévention des accidents de travail, et doit les observer lors de la conception et de l'implémentation du système.

INSTRUCTIONS D'UTILISATION

Ce produit est une bibliothèque à utiliser avec un système de commande et des servo-amplificateurs. La bibliothèque n'est conçue qu'en vue d'une utilisation telle que décrite dans la présente documentation, appliquée au secteur industriel.

Observez en permanence les instructions applicables liées à la sécurité, les conditions spécifiques et les données techniques.

Réalisez une analyse des risques en rapport avec l'utilisation spécifique avant d'utiliser ce produit. Prenez les mesures de sécurité qui découlent des résultats.

Étant donné que ce produit est utilisé au sein d'un système qui l'englobe, vous devez assurer la sécurité du personnel par la conception même du système global (la conception de la machine, par exemple).

Aucune autre utilisation n'est prévue. Toute autre utilisation pourrait être dangereuse.

Avant de commencer

N'utilisez pas ce produit sur les machines non pourvues de protection efficace du point de fonctionnement. L'absence de ce type de protection sur une machine présente un risque de blessures graves pour l'opérateur.

AVERTISSEMENT

EQUIPEMENT NON PROTEGE

- N'utilisez pas ce logiciel ni les automatismes associés sur des appareils non équipés de protection du point de fonctionnement.
- N'accédez pas aux machines pendant leur fonctionnement.

Le non-respect de ces instructions peut provoquer la mort, des blessures graves ou des dommages matériels.

Cet automatisme et le logiciel associé permettent de commander des processus industriels divers. Le type ou le modèle d'automatisme approprié pour chaque application dépendra de facteurs tels que la fonction de commande requise, le degré de protection exigé, les méthodes de production, des conditions inhabituelles, la législation, etc. Dans certaines applications, plusieurs processeurs seront nécessaires, notamment lorsque la redondance de sauvegarde est requise.

Vous seul, en tant que constructeur de machine ou intégrateur de système, pouvez connaître toutes les conditions et facteurs présents lors de la configuration, de l'exploitation et de la maintenance de la machine, et êtes donc en mesure de déterminer les équipements automatisés, ainsi que les sécurités et verrouillages associés qui peuvent être utilisés correctement. Lors du choix de l'automatisme et du système de commande, ainsi que du logiciel associé pour une application particulière, vous devez respecter les normes et réglementations locales et nationales en vigueur. Le document National Safety Council's Accident Prevention Manual (reconnu aux Etats-Unis) fournit également de nombreuses informations utiles.

Dans certaines applications, telles que les machines d'emballage, une protection supplémentaire, comme celle du point de fonctionnement, doit être fournie pour l'opérateur. Elle est nécessaire si les mains ou d'autres parties du corps de l'opérateur peuvent entrer dans la zone de point de pincement ou d'autres zones dangereuses, risquant ainsi de provoquer des blessures graves. Les produits logiciels seuls, ne peuvent en aucun cas protéger les opérateurs contre d'éventuelles blessures. C'est pourquoi le logiciel ne doit pas remplacer la protection de point de fonctionnement ou s'y substituer.

Avant de mettre l'équipement en service, assurez-vous que les dispositifs de sécurité et de verrouillage mécaniques et/ou électriques appropriés liés à la protection du point de fonctionnement ont été installés et sont opérationnels. Tous les dispositifs de sécurité et de verrouillage liés à la protection du point de fonctionnement doivent être coordonnés avec la programmation des équipements et logiciels d'automatisation associés.

NOTE: La coordination des dispositifs de sécurité et de verrouillage mécaniques/électriques du point de fonctionnement n'entre pas dans le cadre de cette bibliothèque de blocs fonction, du Guide utilisateur système ou de toute autre mise en œuvre référencée dans la documentation.

Démarrage et test

Avant toute utilisation de l'équipement de commande électrique et des automatismes en vue d'un fonctionnement normal après installation, un technicien qualifié doit procéder à un test de démarrage afin de vérifier que l'équipement fonctionne correctement. Il est essentiel de planifier une telle vérification et d'accorder suffisamment de temps pour la réalisation de ce test dans sa totalité.

⚠ AVERTISSEMENT

RISQUES INHERENTS AU FONCTIONNEMENT DE L'EQUIPEMENT

- Assurez-vous que toutes les procédures d'installation et de configuration ont été respectées.
- Avant de réaliser les tests de fonctionnement, retirez tous les blocs ou autres cales temporaires utilisés pour le transport de tous les dispositifs composant le système.
- Enlevez les outils, les instruments de mesure et les débris éventuels présents sur l'équipement.

Le non-respect de ces instructions peut provoquer la mort, des blessures graves ou des dommages matériels.

Effectuez tous les tests de démarrage recommandés dans la documentation de l'équipement. Conservez toute la documentation de l'équipement pour référence ultérieure.

Les tests logiciels doivent être réalisés à la fois en environnement simulé et réel

Vérifiez que le système entier est exempt de tout court-circuit et mise à la terre temporaire non installée conformément aux réglementations locales (conformément au National Electrical Code des Etats-Unis, par exemple). Si des tests diélectriques sont nécessaires, suivez les recommandations figurant dans la documentation de l'équipement afin d'éviter de l'endommager accidentellement.

Avant de mettre l'équipement sous tension :

- Enlevez les outils, les instruments de mesure et les débris éventuels présents sur l'équipement.
- Fermez le capot du boîtier de l'équipement.
- Retirez toutes les mises à la terre temporaires des câbles d'alimentation entrants.
- Effectuez tous les tests de démarrage recommandés par le fabricant.

Fonctionnement et réglages

Les précautions suivantes sont extraites du document NEMA Standards
Publication ICS 7.1-1995 :

(En cas de divergence ou de contradiction entre une traduction et l'original
anglais, le texte original en anglais prévaudra.)

- Malgré le soin apporté à la conception et à la fabrication de l'équipement ou au choix et à l'évaluation des composants, des risques subsistent en cas d'utilisation inappropriée de l'équipement.
- Il arrive parfois que l'équipement soit dérégulé accidentellement, entraînant ainsi un fonctionnement non satisfaisant ou non sécurisé. Respectez toujours les instructions du fabricant pour effectuer les réglages fonctionnels. Les personnes ayant accès à ces réglages doivent connaître les instructions du fabricant de l'équipement et les machines utilisées avec l'équipement électrique.
- L'opérateur ne doit avoir accès qu'aux réglages fonctionnels dont il a besoin. L'accès aux autres commandes doit être limité afin d'empêcher les changements non autorisés des caractéristiques de fonctionnement.

A propos de ce manuel

Objectif du document

Ce document décrit la bibliothèque PackML.

Cette bibliothèque fournit des fonctions et des blocs fonction, des structures et des cadres de visualisation qui sont autant d'outils vous permettant de créer des applications en conformité avec les PackML et PackTags définis dans la norme ANSI/ISA TR88.00.02-2022.

La bibliothèque n'est pas (et ne peut être) une implémentation complète de la norme ANSI/ISA TR88.00.02-2022, dans la mesure où vous assumez la responsabilité du code d'application qui supporte votre machine ou votre processus. La bibliothèque est juste un outil qui vous aide à déployer la norme PackML pour une application de machine.

Champ d'application

Ce document a été actualisé pour le lancement de EcoStruxure™ Machine Expert V2.2.

Langues disponibles

Ce document est disponible dans les langues suivantes :

- English (EIO0000002809)
- French (EIO0000002810)
- German (EIO0000002811)
- Italian (EIO0000002812)
- Spanish (EIO0000002813)
- Chinese (EIO0000002814)

Document(s) à consulter

Titre du document	Référence
Meilleures pratiques en matière de cybersécurité	CS-Best-Practices-2019-340
Recommandations de cybersécurité pour les solutions d'automatisme	EIO0000004242
EcoStruxure Machine Expert - Fonctions et bibliothèques - Guide de l'utilisateur	EIO0000002829 (ENG); EIO0000002830 (FRE); EIO0000002831 (GER); EIO0000002832 (ITA); EIO0000002833 (SPA); EIO0000002834 (CHS);
EcoStruxure Machine Expert - Guide de programmation	EIO0000002854 (ENG); EIO0000002855 (FRE); EIO0000002856 (GER); EIO0000002857 (ITA); EIO0000002858 (SPA); EIO0000002859 (CHS);

Les documents sont accessibles à partir du centre de téléchargement Schneider Electric (www.se.com/fr/fr/download/).

Informations produit

⚠ AVERTISSEMENT

PERTE DE CONTROLE

- Réalisez une analyse des modes de défaillance et de leurs effets (FMEA) ou une analyse de risques équivalente sur l'application et appliquez les contrôles de prévention et de détection appropriés avant la mise en œuvre.
- Prévoyez un état de repli pour les événements ou séquences de commande indésirables.
- Le cas échéant, prévoyez des chemins de commande séparés et redondants.
- Définissez les paramètres appropriés, notamment pour les limites.
- Examinez les conséquences des retards de transmission et prenez les mesures correctives nécessaires.
- Examinez les conséquences des interruptions de la liaison de communication et prenez des mesures correctives nécessaires.
- Prévoyez des chemins indépendants pour les fonctions de commande critiques (arrêt d'urgence, dépassement de limites, conditions d'erreur, etc.) en fonction de votre évaluation des risques ainsi que des réglementations et consignes applicables.
- Appliquez les réglementations et les consignes locales de sécurité et de prévention des accidents.¹
- Testez chaque mise en œuvre d'un système pour vérifier son bon fonctionnement avant de le mettre en service.

Le non-respect de ces instructions peut provoquer la mort, des blessures graves ou des dommages matériels.

¹ Pour plus d'informations, consultez le document NEMA ICS 1.1 (dernière édition), *Safety Guidelines for the Application, Installation, and Maintenance of Solid State Control* (Directives de sécurité pour l'application, l'installation et la maintenance de commande statique) et le document NEMA ICS 7.1 (dernière édition), *Safety Standards for Construction and Guide for Selection, Installation, and Operation of Adjustable-Speed Drive Systems* (Normes de sécurité relatives à la construction et manuel de sélection, d'installation et d'exploitation de variateurs de vitesse) ou leur équivalent en vigueur dans votre pays.

Avant de tenter de fournir une solution (machine ou processus) pour une application spécifique en utilisant les POU trouvés dans la bibliothèque, vous devez tenir compte de la réalisation et de l'exécution des bonnes pratiques. La liste non exhaustive de ces pratiques liées à cette bibliothèque inclut l'analyse des risques, la sécurité fonctionnelle, la compatibilité des composants, les tests et la validation du système.

⚠ AVERTISSEMENT

UTILISATION INCORRECTE DES UNITES ORGANISATIONNELLES DU PROGRAMME

- Effectuez une analyse de la sécurité de l'application et des équipements installés.
- Vérifiez que les POU sont compatibles avec les équipements du système et n'ont pas d'effets inattendus sur le bon fonctionnement du système.
- Assurez-vous que l'axe est en position d'origine et que le référencement est valide avant d'utiliser des mouvements absolus ou des POU utilisant des mouvements absolus.
- Utilisez les paramètres appropriés, notamment les valeurs limites, et observez l'usure de la machine et son fonctionnement à l'arrêt.
- Vérifiez que les capteurs et déclencheurs sont compatibles avec les POU sélectionnés.
- Testez de manière approfondie toutes les fonctions durant la vérification et la mise en service dans tous les modes de fonctionnement.
- Indiquez des méthodes indépendantes pour les fonctions de contrôle critiques (arrêt d'urgence, conditions des valeurs limites dépassées, etc.) conformément à une analyse de sécurité, aux règles en vigueur et aux réglementations applicables.

Le non-respect de ces instructions peut provoquer la mort, des blessures graves ou des dommages matériels.

⚠ AVERTISSEMENT

FONCTIONNEMENT IMPREVU DE L'EQUIPEMENT

- N'utilisez que le logiciel approuvé par Schneider Electric pour faire fonctionner cet équipement.
- Mettez à jour votre programme d'application chaque fois que vous modifiez la configuration matérielle physique.

Le non-respect de ces instructions peut provoquer la mort, des blessures graves ou des dommages matériels.

Les transferts incomplets, qu'il s'agisse de fichiers de données, d'application et/ou de micrologiciel, peuvent avoir des conséquences graves sur votre machine ou votre contrôleur. En cas coupure de courant (volontaire ou non) ou d'interruption de la communication pendant un transfert de fichier, votre machine peut devenir inopérante ou votre application peut tenter d'utiliser un fichier de données endommagé. En cas d'interruption, relancez le transfert. Veillez à inclure l'impact des fichiers de données endommagés dans votre analyse des risques.

⚠ AVERTISSEMENT

FONCTIONNEMENT IMPREVU DE L'EQUIPEMENT, PERTE DE DONNEES OU FICHIER ENDOMMAGE

- N'interrompez pas un transfert de données en cours.
- Si le transfert est interrompu pour une raison quelconque, relancez-le.
- Ne mettez pas votre machine en service tant que le transfert de fichier n'est pas terminé, sauf si vous avez pris en compte les fichiers endommagés dans votre analyse des risques et si vous avez mis en place des mesures appropriées pour prévenir les conséquences potentiellement graves dues à des échecs de transfert.

Le non-respect de ces instructions peut provoquer la mort, des blessures graves ou des dommages matériels.

Les transferts incomplets, qu'il s'agisse de fichiers de données, d'application et/ou de micrologiciel, peuvent avoir des conséquences graves sur votre machine ou votre contrôleur. En cas coupure de courant (volontaire ou non) ou d'interruption de la communication pendant un transfert de fichier, votre machine peut devenir inopérante ou votre application peut tenter d'utiliser un fichier de données endommagé. En cas d'interruption, relancez le transfert. Veillez à inclure l'impact des fichiers de données endommagés dans votre analyse des risques.

⚠ AVERTISSEMENT

FONCTIONNEMENT IMPREVU DE L'EQUIPEMENT, PERTE DE DONNEES OU FICHIER ENDOMMAGE

- N'interrompez pas un transfert de données en cours.
- Si le transfert est interrompu pour une raison quelconque, relancez-le.
- Ne mettez pas votre machine en service tant que le transfert de fichier n'est pas terminé, sauf si vous avez pris en compte les fichiers endommagés dans votre analyse des risques et si vous avez mis en place des mesures appropriées pour prévenir les conséquences potentiellement graves dues à des échecs de transfert.

Le non-respect de ces instructions peut provoquer la mort, des blessures graves ou des dommages matériels.

L'utilisation de cette bibliothèque comme outil de contrôle d'une machine nécessite une attention et des dispositions particulières afin d'éviter des conséquences involontaires dues à l'exploitation de la machine commandée, à des changements d'état ou à l'altération de la mémoire de données ou des éléments de fonctionnement de la machine.

⚠ AVERTISSEMENT

FUNCTIONNEMENT IMPRÉVU DE L'ÉQUIPEMENT

- Placer les instruments de l'opérateur du système de commande près de la machine ou à un endroit qui permet d'avoir une vision parfaite de la machine.
- Protégez les commandes opérateur contre tout accès non autorisé.
- Si le contrôle à distance est une caractéristique nécessaire de l'application, veillez à ce qu'une personne qualifiée et compétente soit présente sur place pour surveiller le fonctionnement contrôlé à distance.
- Configurez et installez l'entrée Run/Stop (si elle est présente) ou un autre moyen externe dans l'application, afin que le contrôle local du démarrage ou de l'arrêt de l'équipement puisse être maintenu indépendamment des commandes envoyées à distance à l'équipement.

Le non-respect de ces instructions peut provoquer la mort, des blessures graves ou des dommages matériels.

Les POU fournis avec cette bibliothèque utilisent des variables de type POINTER TO en interne. Ces pointeurs ne sont affectés qu'au début de l'exécution du POU concerné. C'est-à-dire qu'ils ne sont pas réaffectés tant que le bloc fonction indique `Busy`.

⚠ ATTENTION

POINTEUR INCORRECT

N'utilisez pas la commande « Changement en ligne » ou l'option « Se connecter avec changement en ligne » tant que l'un des blocs fonction de cette bibliothèque indique `Busy` dans votre application en cours d'exécution.

Le non-respect de ces instructions peut provoquer des blessures ou des dommages matériels.

Informations concernant la terminologie inclusive/sensible

Schneider Electric s'efforce de mettre constamment à jour ses communications et ses produits pour respecter ses engagements en matière de terminologie inclusive/sensible. Il se peut malgré tout que nos contenus présentent encore des termes jugés inappropriés par certains clients.

Terminologie utilisée dans les normes

Les termes techniques, la terminologie, les symboles et les descriptions correspondantes employés dans ce manuel ou figurant sur les produits eux-mêmes proviennent généralement des normes internationales.

Dans le domaine des systèmes de sécurité fonctionnelle, des variateurs et de l'automatisme en général, il s'agit par exemple de termes tels que *sécurité*, *fonction de sécurité*, *état sécurisé*, *défaut*, *réinitialisation de défaut*, *dysfonctionnement*, *panne*, *erreur*, *message d'erreur*, *dangereux*, etc.

Ces normes incluent notamment :

Norme	Description
IEC 61131-2:2007	Automates programmables, partie 2 : Spécifications et essais des équipements.
ISO 13849-1:2023	Sécurité des machines : Composants liés à la sécurité dans les systèmes de commande. Principes généraux de conception
EN 61496-1:2013	Sécurité des machines : Equipement de protection électrosensible. Partie 1 : Exigences générales et tests.
ISO 12100:2010	Sécurité des machines - Principes généraux de conception - Appréciation du risque et réduction du risque
EN 60204-1:2006	Sécurité des machines - Equipement électrique des machines - Partie 1 : exigences générales
ISO 14119:2013	Sécurité des machines - Dispositifs de verrouillage associés à des protecteurs - Principes de conception et de choix
ISO 13850:2015	Sécurité des machines - Fonction d'arrêt d'urgence - Principes de conception
IEC 62061:2021	Sécurité des machines - Sécurité fonctionnelle des systèmes de commande électrique, électronique et électronique programmables relatifs à la sécurité
IEC 61508-1:2010	Sécurité fonctionnelle des systèmes électriques, électroniques et électroniques programmables liés à la sécurité : Exigences générales.
IEC 61508-2:2010	Sécurité fonctionnelle des systèmes électriques, électroniques et électroniques programmables liés à la sécurité : Exigences concernant la sécurité fonctionnelle des systèmes électriques, électroniques et électroniques programmables liés à la sécurité.
IEC 61508-3:2010	Sécurité fonctionnelle des systèmes électriques, électroniques et électroniques programmables liés à la sécurité : Configuration logicielle requise.
IEC 61784-3:2021	Réseaux de communication industriels - Profils - Partie 3 : Bus de terrain liés à la sécurité fonctionnelle - Règles générales et définitions de profil.
2006/42/EC	Directive Machines
2014/30/EU	Directive sur la compatibilité électromagnétique
2014/35/EU	Directive sur les basses tensions

De plus, des termes utilisés dans le présent document peuvent provenir d'autres normes telles que :

Norme	Description
Série IEC 60034	Machines électriques rotatives
Série IEC 61800	Entraînements électriques de puissance à vitesse variable
Série IEC 61158	Communications numériques pour les systèmes de mesure et de commande – Bus de terrain utilisés dans les systèmes de commande industriels

Enfin, le terme *zone de fonctionnement* peut être utilisé dans le contexte de la description de dangers spécifiques et a la même signification que *zone à risque* ou *zone dangereuse* dans la directive *Machines (2006/42/EC)* et *ISO 12100:2010*.

NOTE: Les normes susmentionnées peuvent s'appliquer ou pas aux produits cités dans la présente documentation. Pour plus d'informations sur chacune des normes applicables aux produits décrits dans le présent document, consultez les tableaux de caractéristiques de ces références de produit.

Informations générales

Contenu de cette partie

Présentation de la bibliothèque20

Concept de diagnostic26

Entrées et sorties communes28

Présentation de la bibliothèque

Contenu de ce chapitre

Informations générales20

Modes de fonctionnement22

PackTags.....25

Informations générales

Présentation de la bibliothèque

La bibliothèque fournit des composants qui facilitent l'implémentation d'une application basée sur la norme PackML. Les noms et la structure des types de données fournis dans cette bibliothèque correspondent aux définitions indiquées dans ANSI/ISA TR88.00.02-2022.

Caractéristiques de la bibliothèque

Caractéristique	Valeur
Titre de la bibliothèque	PackML
Société	Schneider Electric
Catégorie	Application > Solution > Packaging
Composant	Packaging (Bibliothèques et modèles d'applications > Packaging)
Espace de noms par défaut	PackML
Attribut de modèle de langage	accès qualifié uniquement (voir EcoStruxure Machine Expert, Fonctions et bibliothèques, Guide de l'utilisateur)
Bibliothèque post-compatible	Oui (FCL (voir EcoStruxure Machine Expert, Fonctions et bibliothèques, Guide de l'utilisateur)

NOTE: Cette bibliothèque est paramétrée pour l'accès qualifié uniquement. Il n'est possible d'accéder aux POU, aux structures de données, aux énumérations et aux constantes qu'en utilisant l'espace de noms de la bibliothèque. L'espace de noms par défaut de la bibliothèque est **PackML**.

Exemple de projet

L'exemple de projet PackMLExample.project est fourni avec la bibliothèque. L'exemple de projet montre comment implémenter les composants issus de la bibliothèque PackML.

Il est possible que la configuration spécifique requise pour votre application diffère de celle supposée dans les exemples, modèles, ou architectures décrits ici, quels qu'ils soient. Dans ce cas, il vous faudra adapter à vos propres besoins les informations proposées dans ce document et dans d'autres documents en rapport avec celui-ci. Pour ce faire, vous devrez consulter la documentation spécifique aux composants matériels et/ou logiciels que vous pourriez ajouter ou insérer à la place d'un autre pour obtenir les informations contenues dans cette documentation. Portez une attention particulière à respecter toutes les informations de sécurité, exigences électriques et normes obligatoires pouvant s'appliquer à vos adaptations.

⚠ AVERTISSEMENT

INCOMPATIBILITE AVEC LES REGLEMENTATIONS

Assurez-vous que tous les équipements concernés et les systèmes conçus sont conformes à toutes les normes et réglementations locales, régionales et nationales applicables.

Le non-respect de ces instructions peut provoquer la mort, des blessures graves ou des dommages matériels.

L'exemple de projet est installé sur votre PC avec le logiciel de programmation.
Pour ouvrir l'exemple de projet, procédez comme suit :

Étape	Action	Commentaire
1	Dans EcoStruxure Machine Expert Logic Builder, exécutez la commande Nouveau projet .	–
2	Dans la boîte de dialogue Nouveau projet , sélectionnez A partir de l'exemple dans la liste Type de projet .	–
3	A droite de la boîte de dialogue Nouveau projet , sélectionnez Tout dans la liste Contrôleur .	Résultat : les exemples disponibles sont répertoriés dans la zone de texte Exemples correspondants .
4	Sélectionnez l'exemple dans la liste Exemples correspondants .	–
5	Entrez le nom du nouveau projet et sélectionnez l'emplacement du fichier.	–
6	Cliquez sur le bouton OK .	Résultat : un nouveau projet est créé en fonction de l'exemple sélectionné.

Remarques générales

NOTE: Schneider Electric respecte les bonnes pratiques de l'industrie, en vigueur dans le développement et la mise en œuvre des systèmes de contrôle. Cette approche, dite de « défense en profondeur », permet de sécuriser les systèmes de contrôle industriels. Elle place les contrôleurs derrière des pare-feu pour restreindre leur accès aux seuls personnels et protocoles autorisés.

⚠ AVERTISSEMENT

ACCES NON AUTHENTIFIE ET EXPLOITATION PAR CONSEQUENT NON AUTORISEE DES MACHINES

- Estimez si votre environnement ou vos machines sont connecté(e)s à votre infrastructure vitale et, le cas échéant, prenez les mesures nécessaires de prévention, basées sur le principe de défense en profondeur, avant de connecter le système d'automatisme à un réseau quelconque.
- Limitez au strict nécessaire le nombre d'équipements connectés à un réseau.
- Isolez votre réseau industriel des autres réseaux au sein de votre société.
- Protégez chaque réseau contre les accès non autorisés à l'aide d'un pare-feu, d'un VPN ou d'autres mesures de sécurité éprouvées.
- Surveillez les activités au sein de votre système.
- Empêchez tout accès direct ou liaison directe aux équipements sensibles par des utilisateurs non autorisés ou des actions non authentifiées.
- Préparez un plan de récupération intégrant la sauvegarde des informations de votre système et de votre processus.

Le non-respect de ces instructions peut provoquer la mort, des blessures graves ou des dommages matériels.

Pour plus d'informations sur les mesures organisationnelles et les règles d'accès aux infrastructures, reportez-vous aux normes suivantes : famille de normes ISO/IEC 27000, Critères Communs pour l'évaluation de la sécurité des Technologies de l'Information, ISO/IEC 15408, IEC 62351, ISA/IEC 62443, Cybersecurity Framework (Cadre de cybersécurité) du NIST, Standard of Good Practice for Information Security (Bonne pratique de sécurité de l'information) de l'Information Security Forum. Consultez également le document [Cybersecurity Guidelines for EcoStruxure Machine Expert, Modicon and PacDrive Controllers and Associated Equipment](#).

Modes de fonctionnement

Présentation

Les recommandations PackML définissent un modèle d'état normalisé pour présenter de façon standardisée le comportement fondamental d'une machine d'emballage.

Selon les recommandations PackML, chaque machine d'emballage peut prendre en charge plusieurs modes de fonctionnement.

Chaque mode de fonctionnement peut comprendre jusqu'à 17 états (**Aborted, Aborting, Clearing, Complete, Completing, Execute, Held, Holding, Idle, Resetting, Starting, Stopped, Stopping, Suspended, Suspending, Un-Holding et Un-Suspending**).

Pour passer d'un état à un autre, il existe des transitions entre états. Chaque transition est déclenchée soit par une commande de transition d'état, soit par la condition de SC (State Complete).

L'implémentation de la bibliothèque PackML se concentre sur le mode de contrôle de l'unité tel que décrit dans la norme ANSI/ISA TR88.00.02-2022. Dans tous les cas, lorsque ce document fait référence au mode de fonctionnement, il s'agit spécifiquement du mode de contrôle de l'unité.

Utiliser les modèles d'état

Les modèles d'état figurant dans la bibliothèque PackML sont basés sur le modèle à 17 états tel que défini dans ANSI/ISA TR88.00.02-2022. Voir le diagramme ci-après. Un modèle d'état doit contenir ces états ou un sous-ensemble de ces états.

Un modèle d'état se définit par la spécification des états existants pour le mode de fonctionnement. Les transitions entre états sont créées automatiquement et implicitement sur la base des états sélectionnés.

ST_UnitModeDefinition est la structure qui représente le modèle d'état pour un mode de fonctionnement. Cette structure est requise pour la mise en œuvre du bloc fonction *FB_ModeManager*. La structure peut être initialisée avec les fonctions *FC_InitStateModelExistingStates()* et *FC_InitStateModelChangeStates()*.

Lorsque vous utilisez le bloc fonction *FB_UnitModeManager2*, les modes de fonctionnement souhaités sont définis à l'aide des méthodes *DefineUnitMode()* ou *DefineUnitModeWithHandler()*. La structure *ST_UnitModeDefinition* est fournie par la méthode *GetDefinedUnitModes()*.

Les blocs fonction *FB_ModeManager* ou *FB_UnitModeManager2* vérifient la cohérence du modèle d'état et déterminent si un changement de mode de fonctionnement est possible dans l'état actuel.

FC_CheckCmd2 vérifie si une commande de transition d'état est valide et renvoie l'état cible résultant.

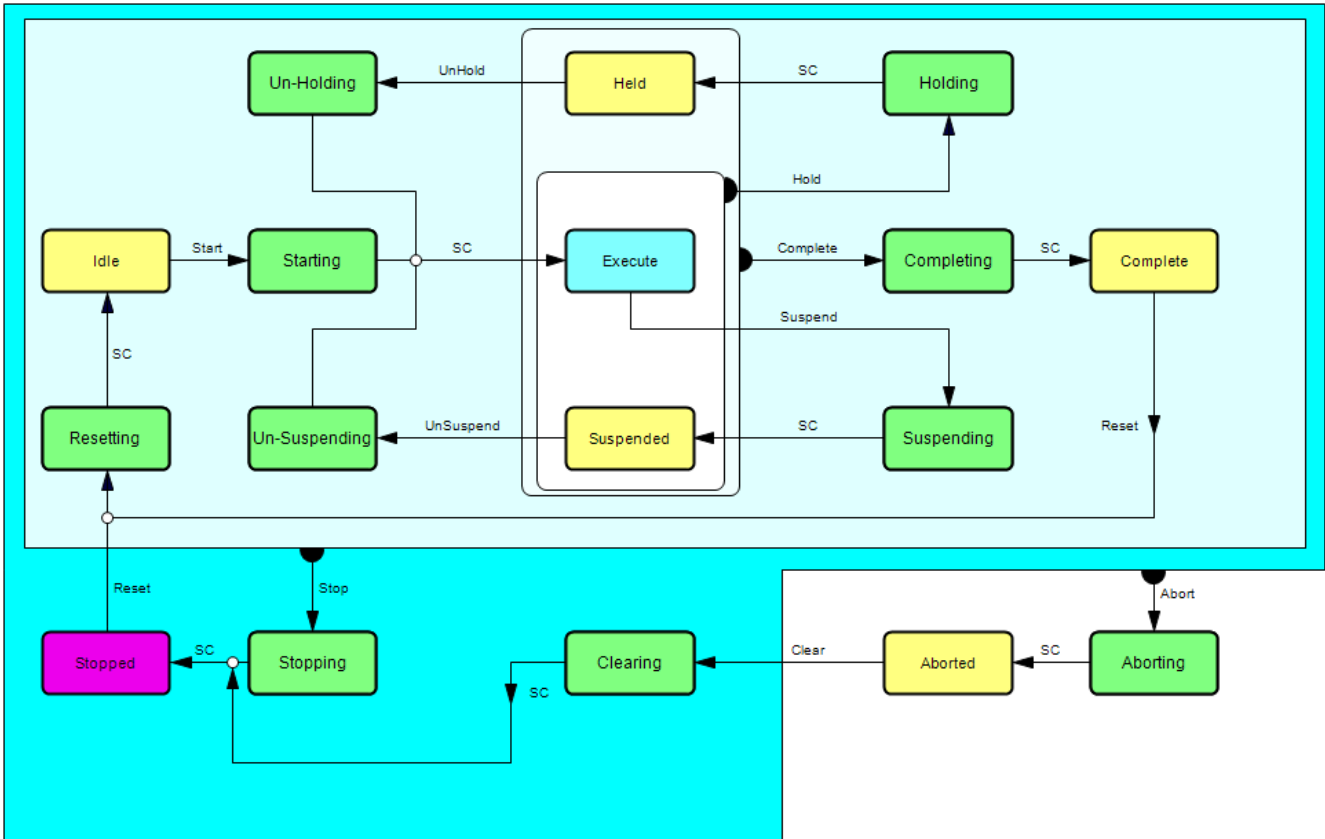
Les cadres de visualisation et *FB_VisController* sont utilisés pour visualiser le modèle d'état et lui fournir une interface de mise en service.

Il vous incombe d'implémenter un flux correct dans le modèle d'état, en accord avec les transitions définies dans le modèle d'état, mais aussi d'implémenter les changements de mode de fonctionnement. Les POU fournis dans la bibliothèque peuvent vous aider à implémenter cette logique (*FB_ModeManager* en combinaison avec *FC_CheckCmd* ou *FC_CheckCmd2*, ou *FB_UnitModeManager2* avec ses méthodes).

Vous devez implémenter le suivi de l'état et du mode de fonctionnement, ainsi qu'un comportement de machine qui correspond aux états, transitions et modes de fonctionnement.

Diagramme de modèle d'état

Le diagramme ci-dessous illustre le modèle d'état pour le mode de fonctionnement à 17 états :



Types d'états

Dans le cadre du modèle d'état, on distingue trois types d'états :

Type d'état	Description issue de ANSI/ISA TR88.00.02-2022	Exemple
État d'attente	Etat utilisé pour indiquer qu'une machine a atteint un ensemble de conditions définies. Dans cet état, la machine reste en attente jusqu'à la transition vers un état d'action.	Idle
État d'action	État qui représente des activités de traitement. Il implique l'exécution unique ou répétée des étapes de traitement dans un ordre logique, pour une durée définie ou jusqu'à ce qu'une condition spécifique soit remplie.	Aborting Execute

Limites

La liste qui suit décrit les limites qui caractérisent les modes de fonctionnement basés sur le modèle d'état :

- Nombre maximal d'états : 17

Les états associés au mode de fonctionnement peuvent être sélectionnés parmi ceux prédéfinis dans le modèle d'état. Chaque état ne peut apparaître qu'une fois, dans une position spécifiée par le modèle d'état. Vous ne pouvez pas ajouter d'états supplémentaires, ni changer l'ordre d'enchaînement des états.

- Le modèle d'état ne fonctionne que dans un sens.
- L'état qui suit un **État d'attente** doit être un **État d'action**.
- Une logique peut être exécutée dans chaque état.

NOTE: D'autres règles s'appliquent pour créer des modes de fonctionnement avec la bibliothèque PackML. Consultez *FB_ModeManager*, page 79 ou *FB_UnitModeManager2*, page 82.

PackTags

Présentation

PackTags fournit un ensemble uniforme de conventions de dénomination pour les éléments de données utilisés dans les éléments procéduraux du modèle d'état de base. Il s'agit d'éléments de données nommés utilisés pour l'architecture ouverte et l'échange de données interopérables dans les machines automatisées.

La bibliothèque fournit les PackTags tels que définis dans la norme ANSI/ISA TR88.00.02-2022.

Variables

Les variables sont divisées en trois types :

Type PackTags	Description	Dans la bibliothèque PackML
commande	Ces variables représentent principalement l'interface de contrôle. L'utilisation de ces balises permet de sélectionner des modes de fonctionnement, de forcer des transitions d'état et de transmettre des recettes et des formats.	<i>ST_Command</i>
état	Ces variables contiennent des informations d'état (indiquant notamment le mode de fonctionnement actif, l'état actif, etc.).	<i>ST_Status</i>
administration	Ces variables contiennent des données importantes OEE (par exemple : temps d'arrêt, motifs des temps d'arrêt, etc.). De plus, des informations relatives à la production.	<i>ST_Administration</i>

Chaque variable de chacun des trois types (commande, état, administration) peut être fournie au contrôleur au sein des structures.

Concept de diagnostic

Contenu de ce chapitre

Concept de diagnostic	26
-----------------------------	----

Concept de diagnostic

Présentation

EcoStruxure Machine Expert fournit un concept de diagnostic à trois niveaux pour les bibliothèques. Ce concept est valable pour les bibliothèques Technologie/Module (par exemple, la bibliothèque CommonToolbox) et utilise des énumérations pour le codage de diagnostic.

En principe, les informations de diagnostic sont organisées selon les niveaux suivants :

1. Informations générales sur l'exception. Aucune connaissance particulière concernant le fonctionnement du POU n'est requise.
2. Messages d'état et de diagnostic propres aux POU (partie 1) : Informations détaillées sur la source qui déclenche les messages de diagnostic ou d'état.
3. Messages d'état et de diagnostic propres aux POU (partie 2) : Informations détaillées et dynamiques sur la source qui déclenche les messages de diagnostic ou d'état.

Ces informations changent lors de l'exécution (par exemple, informations sur la condition des paramètres d'entrée). Cette sortie de diagnostic est facultative pour les POU.

Le concept de diagnostic présente les avantages suivants :

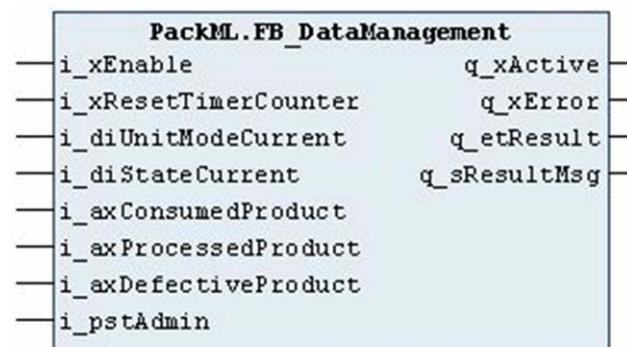
- Affichage en ligne des messages de diagnostic
- Informations détaillées sur les événements de diagnostic via les codes de diagnostic
- Présentation de l'état ou de la condition exceptionnelle d'un POU
- Suggestions pertinentes pour corriger les causes de conditions exceptionnelles
- Messages de diagnostic au format énumération permettant la prise en charge multilingue des afficheurs IHM

Sorties de diagnostic

Les blocs fonction et les fonctions ou méthodes peuvent avoir les trois sorties de diagnostic *q_xError*, *q_etResult* et *q_sResultMsg*. Ces sorties sont définies l'une après l'autre dans le POU.

Sortie	Type de données	Signification
<i>q_xError</i>	BOOL	La sortie <i>q_xError</i> est définie sur TRUE lorsqu'une erreur a été détectée pendant l'exécution du bloc fonction. Les sorties <i>q_etResult</i> et <i>q_sResultMsg</i> fournissent le code d'erreur correspondant et des informations en texte simple.
<i>q_etResult</i>	ET_Result ⁽¹⁾	Fournit des informations de diagnostic et d'état sous la forme d'une valeur numérique. If <i>q_xError</i> = FALSE, <i>q_etResult</i> fournit des informations d'état. If <i>q_xError</i> = TRUE, <i>q_etResult</i> fournit des informations de diagnostic / d'erreur. L'énumération <i>ET_Result</i> contient les valeurs possibles des résultats du fonctionnement du POU.
<i>q_sResultMsg</i>	STRING	Fournit des informations supplémentaires sur l'état actuel du POU. Si le POU est en état d'erreur (<i>q_xError</i> = TRUE), le message fournit des informations supplémentaires sur la cause de l'erreur et suggère éventuellement une solution possible. Si le POU est occupé, le processus ou l'état actuel est indiqué au niveau de cette sortie.
(1) Chaque bibliothèque fournit sa propre énumération <i>ET_Result</i> qui contient la somme des sorties <i>q_etResult</i> possibles dans l'ensemble de la bibliothèque.		

Exemple d'informations de diagnostic :



Entrées et sorties communes

Contenu de ce chapitre

Comportement des blocs fonction avec entrée *i_xEnable*28

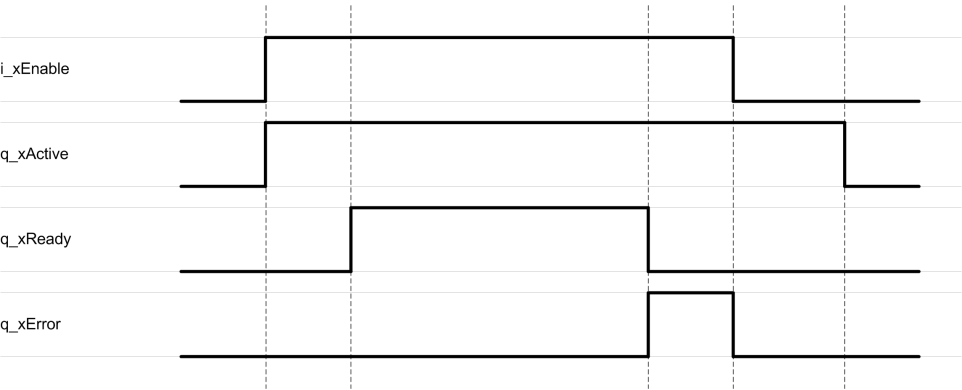
Comportement des blocs fonction avec entrée *i_xEnable*

Informations générales

Avec l'entrée *i_xEnable* définie sur TRUE, le bloc fonction lance le processus d'activation. Le bloc fonction poursuit l'initialisation et la sortie *q_xActive* est définie sur TRUE. Une fois l'initialisation terminée, la sortie *q_xReady* est définie sur TRUE.

Si une erreur est détectée, la sortie *q_xError* conserve la valeur TRUE jusqu'à la désactivation du bloc fonction.

Exemple



Types d'unités de données

Contenu de cette partie

Énumérations 30

Structures 36

Alias - *DateTimeArray*..... 59

Énumérations

Contenu de ce chapitre

<i>ET_Cmd</i>	30
<i>ET_Modes</i>	31
<i>ET_Result</i>	31
<i>ET_StateModelDefinition</i>	33
<i>ET_States</i>	34

ET_Cmd

Présentation

Type :	Énumération
Disponible à partir de la version :	V1.0.1.0

Description

L'énumération inclut les commandes d'état utilisées pour effectuer une transition d'état. On ne peut sortir d'un **état d'Attente**, page 22 qu'avec une commande de transition d'état (*Start*, par exemple).

Éléments de l'énumération

Nom	Type de données	Valeur	Description
<i>Undefined</i>	DINT	0	Les commandes de transition d'un état à un autre sont décrites ici : <i>FC_CheckCmd</i> .
<i>Reset</i>	DINT	1	
<i>Start</i>	DINT	2	
<i>Stop</i>	DINT	3	
<i>Hold</i>	DINT	4	
<i>UnHold</i>	DINT	5	
<i>Suspend</i>	DINT	6	
<i>UnSuspend</i>	DINT	7	
<i>Abort</i>	DINT	8	
<i>Clear</i>	DINT	9	

Utilisé par

Les valeurs de l'énumération peuvent être utilisées comme valeurs d'entrée pour *i_diCmd* de *FC_CheckCmd* ou pour remplir la valeur de la variable *CntrlCmd* de *ST_Command*.

ET_Modes

Présentation

Type :	Énumération
Disponible à partir de :	V1.0.1.0

Description

L'énumération inclut les modes de fonctionnement prédéfinis (production, par exemple). Un mode de fonctionnement est un sous-ensemble ordonné d'états, de commandes, et de transitions qui peut être initialisé via la structure *ST_UnitModeDefinition*.

Éléments d'énumération

Nom	Type de données	Valeur	Description
<i>Undefined</i>	DINT	0	Le mode de fonctionnement est indéfini ou non valide.
<i>Producing</i> ⁽¹⁾	DINT	1	Mode de fonctionnement de production.
<i>Maintenance</i>	DINT	2	Mode de fonctionnement de maintenance.
<i>Manual</i>	DINT	3	Mode de fonctionnement manuel.
(1) Avec la révision 2015 de ANSI/ISA TR88.00.02-2015, le mode de contrôle <i>Producing</i> est devenu <i>Production</i> . Pour des raisons de compatibilité avec la version antérieure de la bibliothèque PackML, le nom <i>Producing</i> a été conservé.			

Utilisé par

Les valeurs de l'énumération peuvent être utilisées pour remplir les valeurs des variables *UnitModeCurrent* de *ST_Status* et *UnitMode* de *ST_Command*.

ET_Result

Présentation

Type :	Enumération
Disponible à partir de :	V1.0.1.0

Description

L'énumération inclut les valeurs possibles qui donnent les résultats des opérations exécutées par le bloc fonction.

Éléments d'énumération

Nom	Type de données	Valeur	Description
Si le paramètre <i>q_xError</i> d'un bloc fonction, page 77 est FALSE, l'un des messages d'état suivants apparaît.			
<i>Ok</i>	DINT	0	Le POU OK, mais il n'est pas en cours d'exécution.
<i>Disabled</i>	DINT	1	Le POU est désactivé.
<i>Initializing</i>	DINT	2	Le POU est en cours d'initialisation.
<i>Running</i>	DINT	4	Le POU est en cours d'exécution.
<i>ResettingTimerAndCounter</i>	DINT	5	Le POU est en train de réinitialiser l'horloge et le compteur.
Si le paramètre <i>q_xError</i> d'un bloc fonction, page 77 a la valeur TRUE, l'un des messages d'erreur suivants s'affiche.			
<i>UnitModeCurrentRange</i>	DINT	11	La valeur de l'entrée <i>i_diUnitModeCurrent</i> est hors plage. La valeur du mode de fonctionnement doit être comprise entre 0 et <i>Gc_uiMaxNumberOfModes</i> .
<i>StateCurrentRange</i>	DINT	12	La valeur de l'entrée <i>i_diStateCurrent</i> est hors plage. La valeur de l'état doit être comprise entre 0 et <i>Gc_uiMaxNumberOfStates</i> .
<i>NumberOfModesRange</i>	DINT	13	Le nombre de modes de fonctionnement est hors plage. Le nombre de modes de fonctionnement doit être compris entre 0 et <i>Gc_uiMaxNumberOfModes</i> .
<i>PointerInitModeInvalid</i>	DINT	14	Le pointeur de mode de fonctionnement initial <i>i_pstInitMode</i> n'est pas valide.
<i>PointerAdminInvalid</i>	DINT	15	Le pointeur admin <i>i_pstAdmin</i> n'est pas valide.
<i>PointerStatusInvalid</i>	DINT	16	Le pointeur d'état <i>i_pstStatus</i> n'est pas valide.
<i>UnitModeRange</i>	DINT	17	Le mode de fonctionnement est hors plage. La plage devrait être : $0 < i_diUnitMode \leq diNumberOfModes$
<i>InvalidStateModelNoExecuteState</i>	DINT	20	Le modèle d'état du mode de fonctionnement n'est pas valide : il n'y a pas d'état Execute .
<i>InvalidStateModelNoStoppedState</i>	DINT	21	Le modèle d'état du mode de fonctionnement n'est pas valide : il n'y a pas d'état Stopped .
<i>InvalidStateModelNoHeldState</i>	DINT	22	Le modèle d'état du mode de fonctionnement n'est pas valide : un état Held doit exister si un état Holding ou un état Un-Holding existe.
<i>InvalidStateModelNoCompleteState</i>	DINT	23	Le modèle d'état du mode de fonctionnement n'est pas valide : si l'état Completing existe, un état Complete doit exister.
<i>InvalidStateModelNoSuspendedState</i>	DINT	24	Le modèle d'état du mode de fonctionnement n'est pas valide : un état Suspended doit exister s'il existe un état Suspending ou Un-Suspending .
<i>InvalidStateModelNoAbortedState</i>	DINT	25	Le modèle d'état du mode de fonctionnement n'est pas valide : un état Aborted doit exister s'il existe un état Aborting ou Clearing .
<i>InvalidStateModelNoIdleState</i>	DINT	26	Le modèle d'état du mode de fonctionnement n'est pas valide : l'état Resetting ne peut être utilisé qu'en lien avec l'état Idle .
<i>InvalidStateModelNoResettingState</i>	DINT	27	Le modèle d'état du mode de fonctionnement n'est pas valide : l'état Idle ne peut être utilisé qu'en lien avec l'état Resetting .
<i>ModeChangeRequestRejected</i>	DINT	28	La requête de changement de mode de fonctionnement est rejetée.
<i>StateChangeRequestRejected</i>	DINT	29	La commande d'état déclenchée n'est pas acceptée dans l'état actuel.
<i>UnknownResult</i>	DINT	30	La valeur de l'entrée <i>i_etResult</i> de la fonction <i>FC_EtResultToString</i> est inconnue.
<i>InvalidInput</i>	DINT	33	Au moins une des entrées spécifiées n'est pas valide.

Nom	Type de données	Valeur	Description
<i>MaxNumberOfUnitModesExceeded</i>	DINT	34	Le nombre maximum de modes d'unité définis est atteint. Supprimez un mode d'unité avant d'en ajouter un nouveau.
<i>NoUnitMode</i>	DINT	35	Aucun mode d'unité n'est actif. L'opération commandée n'est pas autorisée.
<i>UnitModeNotFound</i>	DINT	36	Le mode d'unité spécifié n'est pas défini.
<i>UnitModeActive</i>	DINT	37	Le mode d'unité est actif. L'opération commandée n'est pas autorisée.
<i>NoStateModelHandler</i>	DINT	39	La définition du mode d'unité nécessite une instance de <i>FB_StateModelHandlerBase</i> .
<i>RegisterLoggerPointFailed</i>	DINT	41	Impossible d'enregistrer la source de journalisation.
<i>AlreadyExists</i>	DINT	42	Un mode d'unité de même valeur ou de même nom est déjà défini.
<i>NotReady</i>	DINT	43	Le POU est bloqué par une autre tâche et n'est pas prêt à être exécuté.

Utilisé par

L'énumération *ET_Result* est utilisée par tous les POU de cette bibliothèque.

ET_StateModelDefinition

Présentation

Type :	Énumération
Disponible à partir de :	V1.4.2.0

Description

Cette énumération permet de définir le modèle d'état d'un mode d'unité. Les états 1 à 17 du modèle d'état PackML sont représentés par les bits 1 à 17. L'état 0 étant un état non défini, le bit 0 n'est pas utilisé.

Éléments d'énumération

Nom	Type de données	Valeur	Description
<i>AbortedAndStopped</i>	DINT	000000001000000100 en binaire	Les bits correspondant aux états <i>Aborted</i> et <i>Stopped</i> sont TRUE. Peut être utilisé pour définir les états où un changement de mode est autorisé.
<i>NoStateDisabled</i>	DINT	0 en binaire	Aucun des bits n'est VRAI, tous les états sont activés. Peut être utilisé pour définir les états qui sont désactivés.
<i>HoldDisabled</i>	DINT	1110000000000 en binaire	Les bits correspondant aux états <i>Holding</i> , <i>Held</i> et <i>UnHolding</i> sont TRUE. Peut être utilisé pour définir les états qui sont désactivés.
<i>SuspendDisabled</i>	DINT	110000000100000 en binaire	Les bits correspondant aux états <i>Suspending</i> , <i>Suspended</i> et <i>UnSuspending</i> sont TRUE. Peut être utilisé pour définir les états qui sont désactivés.
<i>CompleteDisabled</i>	DINT	11000000000000000 en binaire	Les bits correspondant aux états <i>Completing</i> et <i>Complete</i> sont TRUE. Peut être utilisé pour définir les états qui sont désactivés.
<i>Clearing</i>	DINT	2 hex	Le bit 1 représente l'état <i>Clearing</i> .
<i>Stopped</i>	DINT	4 hex	Le bit 2 représente l'état <i>Stopped</i> .
<i>Starting</i>	DINT	8 hex	Le bit 3 représente l'état <i>Starting</i> .
<i>Idle</i>	DINT	10 hex	Le bit 3 représente l'état <i>Idle</i> .
<i>Suspended</i>	DINT	20 hex	Le bit 5 représente l'état <i>Suspended</i> .
<i>Execute</i>	DINT	40 hex	Le bit 6 représente l'état <i>Execute</i> .
<i>Stopping</i>	DINT	50 hex	Le bit 7 représente l'état <i>Stopping</i> .
<i>Aborting</i>	DINT	100 hex	Le bit 8 représente l'état <i>Aborting</i> .
<i>Aborted</i>	DINT	200 hex	Le bit 9 représente l'état <i>Aborted</i> .
<i>Holding</i>	DINT	400 hex	Le bit 10 représente l'état <i>Holding</i> .
<i>Held</i>	DINT	800 hex	Le bit 11 représente l'état <i>Held</i> .
<i>UnHolding</i>	DINT	1000 hex	Le bit 12 représente l'état <i>UnHolding</i> .
<i>Suspending</i>	DINT	2000 hex	Le bit 13 représente l'état <i>Suspending</i> .
<i>Unsuspending</i>	DINT	4000 hex	Le bit 14 représente l'état <i>Unsuspending</i> .
<i>Resetting</i>	DINT	8000 hex	Le bit 15 représente l'état <i>Resetting</i> .
<i>Completing</i>	DINT	10000 hex	Le bit 16 représente l'état <i>Completing</i> .
<i>Complete</i>	DINT	20000 hex	Le bit 17 représente l'état <i>Complete</i> .

Utilisé par

- *FB_UnitModeManager2*

ET_States

Présentation

Type :	Énumération
Disponible à partir de :	V1.0.1.0

Description

Cette énumération représente la liste des états possibles de la machine comme défini dans ANSI/ISA TR88.00.02-2022.

Éléments d'énumération

NOTE: Pour une description détaillée de chaque état individuel, consultez la révision 2015 de la norme ANSI/ISA TR88.00.02-2022.

Nom	Type de données	Valeur	Description
<i>Undefined</i>	DINT	0	Aucun état n'est défini.
<i>Clearing</i>	DINT	1	Etat d'action Clearing
<i>Stopped</i>	DINT	2	Etat d'attente Stopped
<i>Starting</i>	DINT	3	Etat d'action Démarrage
<i>Idle</i>	DINT	4	Etat d'attente Idle
<i>Suspended</i>	DINT	5	Etat d'attente Suspended
<i>Execute</i>	DINT	6	Etat d'action Execute
<i>Stopping</i>	DINT	7	Etat d'action Stopping
<i>Aborting</i>	DINT	8	Etat d'action Aborting
<i>Aborted</i>	DINT	9	Etat d'attente Aborted
<i>Holding</i>	DINT	10	Etat d'action Holding
<i>Held</i>	DINT	11	Etat d'attente Held
<i>UnHolding</i>	DINT	12	Etat d'action Un-Holding
<i>Suspending</i>	DINT	13	Etat d'action Suspending
<i>UnSuspending</i>	DINT	14	Etat d'action Un-Suspending
<i>Resetting</i>	DINT	15	Etat d'action Resetting
<i>Completing</i>	DINT	16	Etat d'action Completing
<i>Complete</i>	DINT	17	Etat d'attente Complete
NOTE: Le contenu de tout état donné dépend de votre application.			

Utilisé par

- *FB_ModeManager*
- *FB_UnitModeManager2*
- *FB_VisController*
- *i_diStateCurrent* de *FC_CheckCmd*
- *StateCurrent* de *ST_Status*
- *StateRequested* de *ST_Status*

Structures

Contenu de ce chapitre

Structures Principales PackTags36

Autres Structures Principales40

Sous-Structures45

Structures Principales PackTags

ST_Administration

Présentation

Type	Structure de données
Disponible à partir de	V1.0.1.0

Description

La structure *ST_Administration* représente la liste des PackTags du type *Administration* tel que défini dans ANSI/ISA TR88.00.02-2022.

Des balises d'administration sont utilisées pour décrire les informations liées à la qualité et aux alarmes de la machine.

Éléments de la structure

Pour les types de données GPL, consultez la section Paramètres Globaux, page 62.

Variable	Type de données	Description
<i>Alarm</i>	ARRAY [1.. <i>GPL.Gc_uiNumberOfAlarms</i>] OF <i>ST_Alarm</i>	Tableau de <i>ST_Alarm</i>
<i>AlarmExtent</i>	DINT	Taille du tableau d'alarmes
<i>AlarmHistory</i>	ARRAY [1.. <i>GPL.Gc_uiNumberOfAlarmHistory</i>] OF <i>ST_Alarm</i>	Tableau de taille donnée pour les codes et les messages correspondant aux erreurs détectées par la machine, page 45
<i>AlarmHistoryExtent</i>	DINT	Taille du tableau d'historique des alarmes
<i>Warning</i>	ARRAY [1.. <i>GPL.Gc_uiMaxNumberOfWarnings</i>] OF <i>ST_Alarm</i>	Le tableau contient les conseils actifs et/ou non acquittés de la machine
<i>WarningExtent</i>	DINT	Taille du tableau de conseil
<i>MachDesignSpeed</i>	REAL	Vitesse nominale de la machine
<i>PLCDateTime</i>	<i>DateTimeArray</i>	Date et heure actuelles du contrôleur logique programmable
<i>StopReason</i> ⁽²⁾	ARRAY [1.. <i>GPL.Gc_uiMaxNumberOfStopReasons</i>] OF <i>ST_Alarm</i>	Informations sur le premier événement capturé durant un événement de type annulation (Abort), pause (Held), suspension (Suspended) ou arrêt (Stop)
<i>StopReasonExtend</i> ⁽²⁾	DINT	Taille du tableau <i>StopReason</i>
<i>Parameter</i> ⁽²⁾	ARRAY [1.. <i>GPL.Gc_iNumOfAdminParameter</i>] OF <i>ST_Descriptor</i>	Tableau de <i>ST_Descriptor</i>
<i>ModeCurrentTime</i> ⁽²⁾	ARRAY [0.. <i>GPL.Gc_uiMaxNumberOfModes</i>] OF UDINT	Durée dans ce mode de fonctionnement.

Variable	Type de données	Description
<i>ModeCumulativeTime</i> ⁽²⁾	ARRAY [0.. <i>GPL.Gc_uiMaxNumberOfModes</i>] OF UDINT	Durée cumulée dans chaque mode de fonctionnement.
<i>StateCurrentTime</i> ⁽²⁾	ARRAY [0.. <i>GPL.Gc_uiMaxNumberOfModes</i> , 0.. <i>GVL.Gc_uiMaxNumberOfStates</i>] OF UDINT	Valeur de la dernière horloge d'état dans chaque mode de fonctionnement/état
<i>StateCumulativeTime</i> ⁽²⁾	ARRAY [0.. <i>GPL.Gc_uiMaxNumberOfModes</i> , 0.. <i>GVL.Gc_uiMaxNumberOfStates</i>] OF UDINT	Durée cumulée dans chaque mode de fonctionnement/état.
<i>ProdConsumedCount</i> ⁽²⁾	ARRAY [1.. <i>GPL.Gc_uiNumberOfConsumedProducts</i>] OF <i>ST_CountDescrip</i>	Tableau de <i>ST_CountDescrip</i> (consommables utilisés dans la machine de production)
<i>ProdProcessedCount</i> ⁽²⁾	ARRAY [1.. <i>GPL.Gc_uiNumberOfProcessedProducts</i>] OF <i>ST_CountDescrip</i>	Tableau de <i>ST_CountDescrip</i> (produits traités par la machine de production)
<i>ProdDefectiveCount</i> ⁽²⁾	ARRAY [1.. <i>GPL.Gc_uiNumberOfDefectiveProducts</i>] OF <i>ST_CountDescrip</i>	Tableau de <i>ST_CountDescrip</i> (produits marqués comme non conformes dans la machine de production)
<i>AccTimeSinceReset</i> ⁽²⁾	DINT	Durée cumulée depuis la dernière commande <i>Reset</i>
<i>StatesDisabled</i> ⁽²⁾	DINT	Les états 1 à 17 du PackML State Model peuvent être désactivés en activant le bit correspondant
<i>PACDateTime</i> ^{(1) (2)}	<i>ST_Timestamp</i>	Horodatage (type de données ISO de date et d'heure, format ISO 8601:1988), page 57
<i>Parameter_REAL</i>	ARRAY [1.. <i>GPL.Gc_uiNumberOfParameter</i>] OF <i>ST_Parameter_REAL</i>	Tableau structuré d'informations de paramètres unité/machine pour des valeurs de type de données REAL
<i>Parameter_STRING</i>	ARRAY [1.. <i>GPL.Gc_uiNumberOfParameter</i>] OF <i>ST_Parameter_STRING</i>	Tableau structuré d'informations de paramètres unité/machine pour des valeurs de type de données STRING
<i>Parameter_LREAL</i>	ARRAY [1.. <i>GPL.Gc_uiNumberOfParameter</i>] OF <i>ST_Parameter_LREAL</i>	Tableau structuré d'informations sur les paramètres unité/machine pour les valeurs de type de données LREAL
<i>Parameter_DINT</i>	ARRAY [1.. <i>GPL.Gc_uiNumberOfParameter</i>] OF <i>ST_Parameter_DINT</i>	Tableau structuré d'informations de paramètres unité/machine pour des valeurs de type de données DINT
<i>StopReason2</i>	<i>ST_Alarm</i>	Informations sur le premier événement capturé lors d'un événement d'abandon (Abort), de pause (Held), de suspension (Suspended) ou d'arrêt (Stop)
<i>ModeTimeCurrent</i>	DINT	Durée du mode actuel
<i>StateTimeCurrent</i>	DINT	Durée de l'état actuel
<i>CumulativeTimes</i>	ARRAY [1.. <i>GPL.Gc_uiNumberOfCumulativeTimes</i>] OF <i>ST_Cumulative_Times</i>	Tableau structuré de valeurs de temporisateur
<i>ProductData</i>	ARRAY [1.. <i>GPL.Gc_uiNumberOfProcessedProducts</i>] OF <i>ST_Product_Data</i>	Tableau structuré de données de flux de produits
<i>DisabledStatesCfg</i>	ARRAY [1.. <i>GPL.Gc_uiMaxNumberOfModes</i>] OF DWORD	Tableau reflétant la configuration des états désactivés pour les modes définis
<i>CurDisabledStates</i>	DWORD	Reflète les états désactivés pour le mode actuel (<i>ST_Status.UnitModeCurrent</i>)
<i>EnabledModesCfg</i>	DWORD	Elément reflétant la configuration des modes activés pour l'unité / la machine
<i>ModeTransitionCfg</i>	ARRAY [1.. <i>GPL.Gc_uiMaxNumberOfModes</i>] OF DWORD	Tableau reflétant la configuration de transition de mode pour les modes de l'unité / la machine
⁽¹⁾ La variable a été remplacée par <i>PLCDateTime</i> dans ANSI/ISA TR88.00.02-2015, mais elle reste dans cette bibliothèque pour des raisons de compatibilité.		
⁽²⁾ Variable conservée pour des raisons de compatibilité (selon ANSI/ISA TR88.00.02-2015).		

Utilisé par

- *FB_DataManagement*
- *FB_VisController*

ST_Command

Présentation

Type	Structure de données
Disponible à partir de	V1.0.1.0

Description

La structure *ST_Command* représente la liste des PackTags du type *Command* tel que défini dans ANSI/ISA TR88.00.02-2022.

Des balises de commande sont utilisées pour contrôler le fonctionnement de la machine.

Éléments de la structure

Pour les types de données GPL, consultez la section Paramètres Globaux, page 62.

Variable	Type de données	Description
<i>UnitMode</i>	DINT	Mode de fonctionnement cible (voir <i>ET_Modes</i>)
<i>UnitModeChangeRequest</i>	BOOL	TRUE = requête de changement de mode de fonctionnement de la machine
<i>MachSpeed</i>	REAL	Vitesse actuelle de la machine
<i>CntrlCmd</i>	DINT	Valeur de la commande qui fournit la commande d'état pour piloter un changement d'état dans le modèle d'état de base.
<i>CmdChangeRequest</i>	BOOL	TRUE = requête de changement d'état de la machine
<i>MaterialInterlocks</i> ⁽¹⁾	ARRAY [0..31] OF BOOL	Indique que les éléments sont prêts à être traités.
<i>RemoteInterface</i> ⁽¹⁾	ARRAY [1..GPL.Gc_uiNumberOfInterface] OF <i>ST_Interface</i>	Permet de coordonner les machines en amont ou en aval dans une cellule de plusieurs machines
<i>Parameter</i> ⁽¹⁾	ARRAY [1.. GPL.Gc_uiNumberOfParameter] OF <i>ST_Descriptor</i>	Tableau de <i>ST_Descriptor</i>
<i>Product</i> ⁽¹⁾	ARRAY [1.. GPL.Gc_uiNumberOfProducts] OF <i>ST_Product</i>	Tableau de <i>ST_Product</i>
<i>MaterialInterlock</i>	DWORD	Éléments prêts
<i>Parameter_REAL</i>	ARRAY [1.. GPL.Gc_uiNumberOfParameter] OF <i>ST_Parameter_REAL</i>	Tableau structuré d'informations de paramètres unité/machine pour des valeurs de type de données REAL
<i>Parameter_STRING</i>	ARRAY [1.. GPL.Gc_uiNumberOfParameter] OF <i>ST_Parameter_STRING</i>	Tableau structuré d'informations de paramètres unité/machine pour des valeurs de type de données STRING
<i>Parameter_LREAL</i>	ARRAY [1.. GPL.Gc_uiNumberOfParameter] OF <i>ST_Parameter_LREAL</i>	Tableau structuré d'informations de paramètres unité/machine pour des valeurs de type de données LREAL
<i>Parameter_DINT</i>	ARRAY [1.. GPL.Gc_uiNumberOfParameter] OF <i>ST_Parameter_DINT</i>	Tableau structuré d'informations de paramètres unité/machine pour des valeurs de type de données DINT

Variable	Type de données	Description
<i>SelectedRecipe</i>	DINT	Utilisé pour désigner la recette qui doit être exécutée sur la machine pour donner le produit de sortie principal, conformément à une procédure de traitement de recette conçue par l'utilisateur
<i>RecipeChangeRequest</i>	BOOL	TRUE = demande de modification de la recette sur l'unité / la machine
<i>Recipe</i>	ARRAY [1.. <i>GPL.Gc_uiNumberOfRecipes</i>] OF <i>ST_Recipe</i>	Tableau structuré d'informations de recette
(1) Variable conservée pour des raisons de compatibilité (selon ANSI/ISA TR88.00.02-2015).		

ST_Status

Présentation

Type	Structure de données
Disponible à partir de	V1.0.1.0

Description

La structure *ST_Status* représente la liste des PackTags du type *Status* tel que défini dans ANSI/ISA TR88.00.02-2022.

Des balises de statut sont utilisées pour décrire le fonctionnement de la machine.

Éléments de la structure

Pour les types de données GPL, consultez la section Paramètres Globaux, page 62.

Variable	Type de données	Description
<i>UnitModeCurrent</i>	DINT	Mode de fonctionnement utilisé
<i>UnitModeRequested</i>	BOOL	Modification demandée du mode de fonctionnement
<i>UnitModeChangeInProgress</i>	BOOL	Modification demandée du mode de fonctionnement en cours
<i>StateCurrent</i>	DINT	Etat actuel de la machine
<i>StateRequested</i>	DINT	Etat / état cible demandé
<i>StateChangeInProgress</i>	BOOL	Transition d'l'état demandée en cours
<i>MachSpeed</i>	REAL	Consigne de vitesse de la machine PackML
<i>CurMachSpeed</i>	REAL	Vitesse actuelle de la machine PackML
<i>MaterialInterlocks</i>	ARRAY [0..31] OF BOOL;	Décrit l'état des éléments prêts à être traités.
<i>EquipmentInterlock</i>	<i>ST_EquipmentInterlock</i>	Voir <i>ST_EquipmentInterlock</i> .
<i>RemoteInterface</i> ⁽¹⁾	ARRAY [1.. <i>GPL.Gc_uiNumberOfInterface</i>] OF <i>ST_Interface</i>	Permet de coordonner les machines en amont ou en aval dans une cellule de plusieurs machines
<i>Parameter</i> ⁽¹⁾	ARRAY [1.. <i>GPL.Gc_uiNumberOfParameter</i>] OF <i>ST_Descriptor</i>	Tableau de paramètres machine <i>ST_Descriptor</i>
<i>Product</i> ⁽¹⁾	ARRAY [1.. <i>GPL.Gc_uiNumberOfProducts</i>] OF <i>ST_Product</i>	Tableau de <i>ST_Product</i> .
<i>MaterialInterlock</i>	DWORD	Éléments prêts

Variable	Type de données	Description
<i>Parameter_REAL</i>	ARRAY [1.. <i>GPL.Gc_uiNumberOfParameter</i>] OF <i>ST_Parameter_REAL</i>	Structure d'informations de paramètres unité/machine pour des valeurs de type de données REAL
<i>Parameter_STRING</i>	ARRAY [1.. <i>GPL.Gc_uiNumberOfParameter</i>] OF <i>ST_Parameter_STRING</i>	Structure d'informations de paramètres unité/machine pour des valeurs de type de données STRING
<i>Parameter_LREAL</i>	ARRAY [1.. <i>GPL.Gc_uiNumberOfParameter</i>] OF <i>ST_Parameter_LREAL</i>	Structure d'informations de paramètres unité/machine pour des valeurs de type de données LREAL
<i>Parameter_DINT</i>	ARRAY [1.. <i>GPL.Gc_uiNumberOfParameter</i>] OF <i>ST_Parameter_DINT</i>	Structure d'informations de paramètres unité/machine pour des valeurs de type de données DINT
<i>RecipeCurrent</i>	DINT	Recette actuellement exécutée en environnement de production
<i>RecipeRequested</i>	DINT	Reflet de la recette actuellement sélectionnée pour l'environnement de production
<i>RecipeChangeInProgress</i>	BOOL	Indicateur correspondant au processus de permutation de recettes défini par l'utilisateur
<i>Recipe</i>	ARRAY [1.. <i>GPL.Gc_uiNumberOfRecipes</i>] OF <i>ST_Recipe</i>	Structure d'informations de recette
<i>Stacklight</i>	ARRAY [1..31 OF <i>DINT</i>	Indicateur d'état des voyants de pile
(1) Variable conservée pour des raisons de compatibilité (selon ANSI/ISA TR88.00.02-2015).		

Utilisé par

- *FB_VisController*

Autres Structures Principales

ST_InitAlarm

Présentation

Type	Structure de données
Disponible à partir de	V1.0.1.0

Description

La structure *ST_InitAlarm* contient les informations sur l'alarme ou le conseil qui est géré via les fonctions *FC_SetAlarm* ou *FC_SetWarning*.

Éléments de la structure

Variable	Type de données	Description
<i>diID</i>	DINT	ID de message d'événement
<i>diValue</i>	DINT	Valeur de message d'événement
<i>sMessage</i>	STRING[<i>GPL.Gc_uiMaxLengthOfAlarmMessage</i>]	Message d'événement
<i>xisInList</i>	BOOL	L'événement est dans la liste

Variable	Type de données	Description
<i>xisAck</i>	BOOL	L'événement est acquitté
<i>Category</i>	DINT	Catégorie de l'événement

Utilisé par

- *FC_SetAlarm*
- *FC_SetWarning*

ST_UnitModeDefinition

Présentation

Type	Structure de données
Disponible à partir de	V1.0.1.0

Description

La structure *ST_UnitModeDefinition* fournit les informations pour initialiser les modes de fonctionnement dans le bloc fonction *FB_ModeManager*.

Éléments de la structure

Variable	Type de données	Description
<i>xModeExistent</i>	BOOL	TRUE : le mode de fonctionnement est défini.
<i>stClearing</i>	<i>ST_StateInfo</i>	Indique si l'état <i>stClearing</i> est défini comme un état possible pour le mode de fonctionnement actuel et si un changement du mode de fonctionnement est autorisé dans cet état.
<i>stStopped</i>	<i>ST_StateInfo</i>	Indique si l'état <i>stStopped</i> est défini comme un état possible pour le mode de fonctionnement actuel et si un changement du mode de fonctionnement est autorisé dans cet état.
<i>stStarting</i>	<i>ST_StateInfo</i>	Indique si l'état <i>stStarting</i> est défini comme un état possible pour le mode de fonctionnement actuel et si un changement du mode de fonctionnement est autorisé dans cet état.
<i>stIdle</i>	<i>ST_StateInfo</i>	Indique si l'état <i>stIdle</i> est défini comme un état possible pour le mode de fonctionnement actuel et si un changement du mode de fonctionnement est autorisé dans cet état.
<i>stResetting</i>	<i>ST_StateInfo</i>	Indique si l'état <i>stResetting</i> est défini comme un état possible pour le mode de fonctionnement actuel et si un changement du mode de fonctionnement est autorisé dans cet état.
<i>stExecute</i>	<i>ST_StateInfo</i>	Indique si l'état <i>stExecute</i> est défini comme un état possible pour le mode de fonctionnement actuel et si un changement du mode de fonctionnement est autorisé dans cet état.
<i>stStopping</i>	<i>ST_StateInfo</i>	Indique si l'état <i>stStopping</i> est défini comme un état possible pour le mode de fonctionnement actuel et si un changement du mode de fonctionnement est autorisé dans cet état.
<i>stAborting</i>	<i>ST_StateInfo</i>	Indique si l'état <i>stAborting</i> est défini comme un état possible pour le mode de fonctionnement actuel et si un changement du mode de fonctionnement est autorisé dans cet état.
<i>stAborted</i>	<i>ST_StateInfo</i>	Indique si l'état <i>stAborted</i> est défini comme un état possible pour le mode de fonctionnement actuel et si un changement du mode de fonctionnement est autorisé dans cet état.
<i>stHolding</i>	<i>ST_StateInfo</i>	Indique si l'état <i>stHolding</i> est défini comme un état possible pour le mode de fonctionnement actuel et si un changement du mode de fonctionnement est autorisé dans cet état.

Variable	Type de données	Description
<i>stHeld</i>	<i>ST_StateInfo</i>	Indique si l'état <i>stHeld</i> est défini comme un état possible pour le mode de fonctionnement actuel et si un changement du mode de fonctionnement est autorisé dans cet état.
<i>stUnHolding</i>	<i>ST_StateInfo</i>	Indique si l'état <i>stUnHolding</i> est défini comme un état possible pour le mode de fonctionnement actuel et si un changement du mode de fonctionnement est autorisé dans cet état.
<i>stSuspended</i>	<i>ST_StateInfo</i>	Indique si l'état <i>stSuspended</i> est défini comme un état possible pour le mode de fonctionnement actuel et si un changement du mode de fonctionnement est autorisé dans cet état.
<i>stUnSuspending</i>	<i>ST_StateInfo</i>	Indique si l'état <i>stUnSuspending</i> est défini comme un état possible pour le mode de fonctionnement actuel et si un changement du mode de fonctionnement est autorisé dans cet état.
<i>stSuspending</i>	<i>ST_StateInfo</i>	Indique si l'état <i>stSuspending</i> est défini comme un état possible pour le mode de fonctionnement actuel et si un changement du mode de fonctionnement est autorisé dans cet état.
<i>stCompleting</i>	<i>ST_StateInfo</i>	Indique si l'état <i>stCompleting</i> est défini comme un état possible pour le mode de fonctionnement actuel et si un changement du mode de fonctionnement est autorisé dans cet état.
<i>stComplete</i>	<i>ST_StateInfo</i>	Indique si l'état <i>stComplete</i> est défini comme un état possible pour le mode de fonctionnement actuel et si un changement du mode de fonctionnement est autorisé dans cet état.

Utilisé par

- *FB_ModeManager*
- *FB_VisController*
- *FC_InitStateModelChangeStates*

ST_VisInterface

Présentation

Type :	Structure de données
Disponible à partir de :	V1.0.1.0

Description

La structure *ST_VisInterface* génère l'interface sur les écrans de visualisation fournis par la bibliothèque PackML.

Éléments de la structure

Variable	Type de données	Description
<i>vq_xBlink</i>	BOOL	Variable utilisée pour flasher l' État d'action actuel.
<i>vq_diNumberOfModes</i>	DINT	Nombre maximum de modes de fonctionnement
<i>vq_diUnitModeCurrent</i>	DINT	Mode de fonctionnement utilisé
<i>vq_diStateCurrent</i>	DINT	État actuel

Variable	Type de données	Description
<i>vq_diModeCurrentTimeActive</i>	DINT	Durée actuelle dans le mode de fonctionnement actif.
<i>vq_diModeCumulativeTimeActive</i>	DINT	Durée cumulée dans le mode de fonctionnement actif.
<i>vq_diStateCurrentTimeActive</i>	DINT	Durée actuelle dans l'état actif.
<i>vq_diStateCumulativeTimeActive</i>	DINT	Durée cumulée dans l'état actif.
<i>vq_astAlarm</i>	ARRAY [0..10] OF <i>ST_Alarm</i>	Tableau de taille donnée pour le code et le message des erreurs de la machine active
<i>vq_astAlarmHistory</i>	ARRAY [0..10] OF <i>ST_Alarm</i>	Tableau de taille donnée pour le code et les messages d'erreurs de la machine active
<i>vq_diProdConsumedId</i>	DINT	ID du consommable utilisé
<i>vq_sProdConsumedName</i>	STRING [GPL.Gc_uiMaxLengthOfParameterName]	Nom du consommable utilisé
<i>vq_sProdConsumedUnit</i>	STRING [5]	Unité du consommable utilisé
<i>vq_diProdConsumedCount</i>	DINT	Quantité de consommables utilisés
<i>vq_diProdConsumedAccCount</i>	DINT	Quantité cumulée des consommables utilisés
<i>vq_diProdProcessedId</i>	DINT	ID du produit traité
<i>vq_sProdProcessedName</i>	STRING [GPL.Gc_uiMaxLengthOfParameterName]	Nom du produit traité
<i>vq_sProdProcessedUnit</i>	STRING [80]	Unité du produit traité
<i>vq_diProdProcessedCount</i>	DINT	Quantité de produits traités
<i>vq_diProdProcessedAccCount</i>	DINT	Quantité cumulée de produits traités
<i>vq_diProdDefectiveId</i>	DINT	ID du produit non conforme
<i>vq_sProdDefectiveName</i>	STRING [GPL.Gc_uiMaxLengthOfParameterName]	Nom du produit non conforme
<i>vq_sProdDefectiveUnit</i>	STRING [5]	Unité du produit non conforme
<i>vq_diProdDefectiveCount</i>	DINT	Quantité de produits non conformes
<i>vq_diProdDefectiveAccCount</i>	DINT	Quantité cumulée de produits non conformes
<i>vq_diModeCurrentTime</i>	DINT	Durée actuelle dans chaque mode de fonctionnement.
<i>vq_diModeCumulativeTime</i>	DINT	Durée cumulée dans chaque mode de fonctionnement.
<i>vq_diStateCurrentTime</i>	DINT	Durée actuelle dans chaque état.
<i>vq_diStateCumulativeTime</i>	DINT	Durée cumulée dans chaque état .
<i>vq_diTimeSinceLastReset</i>	DINT	Durée depuis la dernière commande Reset
<i>vq_diStateCurrentTimeUndefined</i>	DINT	Durée actuelle dans un état non défini.
<i>vq_diStateCurrentTimeClearing</i>	DINT	Durée actuelle dans l'état Clearing
<i>vq_diStateCurrentTimeStopped</i>	DINT	Durée actuelle dans l'état Stopped
<i>vq_diStateCurrentTimeStarting</i>	DINT	Durée actuelle dans l'état Starting
<i>vq_diStateCurrentTimeIdle</i>	DINT	Durée actuelle dans l'état Idle
<i>vq_diStateCurrentTimeSuspended</i>	DINT	Durée actuelle dans l'état Suspended
<i>vq_diStateCurrentTimeExecute</i>	DINT	Durée actuelle dans l'état Execute
<i>vq_diStateCurrentTimeStopping</i>	DINT	Durée actuelle dans l'état Stopping

Variable	Type de données	Description
<i>vq_diStateCurrentTimeAborting</i>	DINT	Durée actuelle dans l'état Aborting
<i>vq_diStateCurrentTimeAborted</i>	DINT	Durée actuelle dans l'état Aborted
<i>vq_diStateCurrentTimeHolding</i>	DINT	Durée actuelle dans l'état Holding
<i>vq_diStateCurrentTimeHeld</i>	DINT	Durée actuelle dans l'état Held
<i>vq_diStateCurrentTimeUnholding</i>	DINT	Durée actuelle dans l'état Un-Holding
<i>vq_diStateCurrentTimeSuspending</i>	DINT	Durée actuelle dans l'état Suspending
<i>vq_diStateCurrentTimeUnsuspending</i>	DINT	Durée actuelle dans l'état Un-Suspending
<i>vq_diStateCurrentTimeResetting</i>	DINT	Durée actuelle dans l'état Resetting
<i>vq_diStateCurrentTimeCompleting</i>	DINT	Durée actuelle dans l'état Completing
<i>vq_diStateCurrentTimeComplete</i>	DINT	Durée actuelle dans l'état Complete
<i>vq_diStateCumulativeTimeUndefined</i>	DINT	Durée cumulée dans un état non défini.
<i>vq_diStateCumulativeTimeClearing</i>	DINT	Durée cumulée dans l'état Clearing
<i>vq_diStateCumulativeTimeStopped</i>	DINT	Durée cumulée dans l'état Stopped
<i>vq_diStateCumulativeTimeStarting</i>	DINT	Durée cumulée dans l'état Starting
<i>vq_diStateCumulativeTimeIdle</i>	DINT	Durée cumulée dans l'état Idle
<i>vq_diStateCumulativeTimeSuspended</i>	DINT	Durée cumulée dans l'état Suspended
<i>vq_diStateCumulativeTimeExecute</i>	DINT	Durée cumulée dans l'état Execute
<i>vq_diStateCumulativeTimeStopping</i>	DINT	Durée cumulée dans l'état Stopping
<i>vq_diStateCumulativeTimeAborting</i>	DINT	Durée cumulée dans l'état Aborting
<i>vq_diStateCumulativeTimeAborted</i>	DINT	Durée cumulée dans l'état Aborted
<i>vq_diStateCumulativeTimeHolding</i>	DINT	Durée cumulée dans l'état Holding
<i>vq_diStateCumulativeTimeHeld</i>	DINT	Durée cumulée dans l'état Held
<i>vq_diStateCumulativeTimeUnholding</i>	DINT	Durée cumulée dans l'état Un-Holding
<i>vq_diStateCumulativeTimeSuspending</i>	DINT	Durée cumulée dans l'état Suspending
<i>vq_diStateCumulativeTimeUnsuspending</i>	DINT	Durée cumulée dans l'état Un-Suspending
<i>vq_diStateCumulativeTimeResetting</i>	DINT	Durée cumulée dans l'état Resetting
<i>vq_diStateCumulativeTimeCompleting</i>	DINT	Durée cumulée dans l'état Completing
<i>vq_diStateCumulativeTimeComplete</i>	DINT	Durée cumulée dans l'état Complete
<i>vq_xStateClearingExistent</i>	BOOL	TRUE : l'état Clearing existe
<i>vq_xStateStoppedExistent</i>	BOOL	TRUE : l'état Stopped existe
<i>vq_xStateStartingExistent</i>	BOOL	TRUE : l'état Starting existe
<i>vq_xStateIdleExistent</i>	BOOL	TRUE : l'état Idle existe
<i>vq_xStateSuspendedExistent</i>	BOOL	TRUE : l'état Suspended existe
<i>vq_xStateExecuteExistent</i>	BOOL	TRUE : l'état Execute existe
<i>vq_xStateStoppingExistent</i>	BOOL	TRUE : l'état Stopping existe
<i>vq_xStateAbortingExistent</i>	BOOL	TRUE : l'état Aborting existe
<i>vq_xStateAbortedExistent</i>	BOOL	TRUE : l'état Aborted existe
<i>vq_xStateHoldingExistent</i>	BOOL	TRUE : l'état Holding existe
<i>vq_xStateHeldExistent</i>	BOOL	TRUE : l'état Held existe
<i>vq_xStateUnholdingExistent</i>	BOOL	TRUE : l'état Un-Holding existe
<i>vq_xStateSuspendingExistent</i>	BOOL	TRUE : l'état Suspending existe

Variable	Type de données	Description
<i>vq_xStateUnsuspendingExistent</i>	BOOL	TRUE : l'état Un-Suspending existe
<i>vq_xStateResettingExistent</i>	BOOL	TRUE : l'état Resetting existe
<i>vq_xStateCompletingExistent</i>	BOOL	TRUE : l'état Completing existe
<i>vq_xStateCompleteExistent</i>	BOOL	TRUE : l'état Complete existe
<i>viq_dilIndexAlarm</i>	DINT	Index de tableau des alarmes actives
<i>viq_dilIndexAlarmHistory</i>	DINT	Index de tableau de l'historique des alarmes actives
<i>viq_dilIndexProdConsumed</i>	DINT	Index de consommable utilisé
<i>viq_dilIndexProdProcessed</i>	DINT	Index de produit traité
<i>viq_dilIndexProdDefective</i>	DINT	Index de produit non conforme
<i>viq_dilIndexAdminParameter</i>	DINT	Index de paramètre d'administration
<i>viq_dilIndexMode</i>	DINT	Index de mode de fonctionnement
<i>viq_dilIndexWarning</i>	DINT	Index du tableau des conseils actifs
<i>vq_astWarning</i>	ARRAY [0..9] OF <i>ST_Alarm</i>	Tableau de la taille donnée pour le code et les messages de conseil de la machine
<i>vq_audiBgColorAlarm</i>	ARRAY [0..10] OF UDINT	Variables de contrôle de la couleur d'arrière-plan des alarmes sur l'affichage <i>FR_Alarm</i> .
<i>vq_audiBgColorAlarmHistory</i>	ARRAY [0..10] OF UDINT	Variables de contrôle de la couleur d'arrière-plan des alarmes sur l'affichage <i>FR_AlarmHistory</i> .
<i>vq_audiBgColorAlarmWarning</i>	ARRAY [0..9] OF UDINT	Variables de contrôle de la couleur d'arrière-plan des conseils de l'affichage <i>FR_Warning</i> .

Utilisé par

- *FB_VisController*
- Cadres de visualisation

Sous-Structures

ST_Alarm

Présentation

Type	Structure de données
Disponible à partir de	V1.0.1.0

Description

La structure *ST_Alarm* est une collection de balises nécessaires pour décrire un événement (par exemple, les alarmes ou les conseils).

Éléments de la structure

Variable	Type de données	Description
<i>ID</i>	DINT	Identificateur unique de l'événement
<i>Value</i>	DINT	Valeur associée à l'événement pour l'ajout d'informations ou pour détailler les informations de l'ID d'événement ID.
<i>Message</i>	STRING[GPL.Gc_uiMaxLengthOfAlarmMessage]	Message associé à l'événement
<i>TimeEvent</i> ⁽¹⁾	ST_TimeStamp	Horodatage (type de données ISO) indiquant l'heure de l'événement
<i>TimeAck</i> ⁽²⁾	ST_TimeStamp	Horodatage de l'acquittement de l'événement
<i>Trigger</i>	BOOL	TRUE : indique que l'événement est actif.
<i>Category</i>	DINT	Catégorie de l'événement
<i>DateTime</i>	DateTimeArray	Date et heure de l'événement
<i>AckDateTime</i>	DateTimeArray	Date et heure de l'acquittement de l'événement
(1) La variable a été remplacée par <i>DateTime</i> dans ANSI/ISA TR88.00.02-2015 mais reste dans cette bibliothèque pour des raisons de compatibilité. (2) La variable a été remplacée par <i>AckDateTime</i> dans ANSI/ISA TR88.00.02-2015 mais reste dans cette bibliothèque pour des raisons de compatibilité.		

Utilisé par

- *FC_SetAlarm*
- *ST_Administration*
- *ST_VisInterface*

ST_CountDescrip

Présentation

Type	Structure de données
Disponible à partir de	V1.0.1.0

Description

La structure *ST_CountDescrip* est une collection de balises utilisées pour décrire une valeur de compteur.

Éléments de la structure

Variable	Type de données	Description
<i>ID</i>	DINT	ID de <i>CountDescrip</i>
<i>Name</i>	STRING[GPL.Gc_uiMaxLengthOfParameter-Name]	Nom de <i>CountDescrip</i>
<i>Unit</i>	STRING[5]	Chaîne d'unités de <i>CountDescrip</i>

Variable	Type de données	Description
<i>Count</i>	DINT	Valeur de comptage de <i>CountDescrip</i>
<i>AccCount</i>	DINT	Valeur de comptage cumulée de <i>CountDescrip</i>

Utilisé par

- *ST_Administration*

ST_Cumulative_Times

Présentation

Type	Structure de données
Disponible à partir de	V1.4.2.0

Description

La structure *ST_Cumulative_Times* est un ensemble de balises fournissant des informations sur les valeurs de temporisateur.

Éléments de la structure

Variable	Type de données	Description
<i>AccTimeSinceReset</i>	DINT	Temps cumulé depuis la dernière réinitialisation
<i>ModeStateTimes</i>	ARRAY [1.. <i>Gc_uiMaxNumOfModes</i>] OF <i>ST_ModeState_Times</i>	Tableau structuré de valeurs temporelles de mode et d'état pour chaque suivi de temps cumulé

Utilisé par

- *ST_Administration*

ST_Descriptor

Présentation

Type	Structure de données
Disponible à partir de	V1.0.1.0

Description

La structure *ST_Descriptor* est le type de données utilisé pour les balises de paramètres de PackTags. Il sert de descripteur de la balise de paramètre correspondante.

Éléments de la structure

Variable	Type de données	Description
<i>ID</i>	DINT	ID du paramètre
<i>Name</i>	STRING[GPL.Gc_uiMaxLengthOfParameterName]	Nom du paramètre
<i>Unit</i>	STRING[5]	Chaîne d'unités de paramètre
<i>Value</i>	REAL	Valeur du paramètre

Utilisé par

- *ST_Administration*
- *ST_Command*
- *ST_Status*
- *ST_Ingredient*
- *ST_Interface*
- *ST_Product*

ST_EquipmentInterlock

Présentation

Type	Structure de données
Disponible à partir de	V1.1.0.0

Description

La structure *ST_EquipmentInterlock* indique si le système aval ou amont est à l'origine de l'état suspendu d'une machine.

Éléments de la structure

Variable	Type de données	Description
<i>Blocked</i>	BOOL	TRUE : indique qu'un système en aval ne peut pas accepter le produit.
<i>Starved</i>	BOOL	TRUE : indique qu'un système en amont ne peut pas fournir le produit.

Utilisé par

- *ST_Status*

ST_Ingredient

Présentation

Type	Structure de données
Disponible à partir de	V1.0.1.0

Description

La structure *ST_Ingredient* est une collection de balises utilisées pour décrire les matériaux bruts nécessaires pour le produit.

Éléments de la structure

Variable	Type de données	Description
<i>IngredientID</i>	DINT	Nombre arbitraire associé au matériau brut, ou ingrédient d'un numéro de produit particulier.
<i>Parameter</i>	ARRAY [0.. <i>Gc_uiNumberOfIngredientParameter</i>] OF <i>ST_Descriptor</i>	Le tableau structuré décrit les ingrédients spécifiques ou les matériaux bruts, qui sont nécessaires ou étaient utilisés par la machine pour fabriquer le produit spécifique.

Utilisé par

- *ST_Product*

ST_Ingredients

Présentation

Type	Structure de données
Disponible à partir de	V1.4.2.0

Description

La structure *ST_Ingredients* est un ensemble de balises fournissant des informations sur les ingrédients de recette.

Éléments de la structure

Variable	Type de données	Description
<i>Parameter_REAL</i>	ARRAY [1.. <i>GPL.Gc_uiNumOfParamREALRecipeIngredients</i>] OF <i>ST_Parameter_REAL</i>	Tableau structuré d'informations sur les ingrédients pour des valeurs de type de données REAL
<i>Parameter_STRING</i>	ARRAY [1.. <i>GPL.Gc_uiNumOfParamSTRINGRecipeIngredients</i>] OF <i>ST_Parameter_STRING</i>	Tableau structuré d'informations sur les ingrédients pour des valeurs de type de données STRING

Variable	Type de données	Description
<i>Parameter_LREAL</i>	ARRAY [1.. <i>GPL.Gc_uiNumOfParamLREALRecipeIngredients</i>] OF <i>ST_Parameter_LREAL</i>	Tableau structuré d'informations sur les ingrédients pour des valeurs de type de données LREAL
<i>Parameter_DINT</i>	ARRAY [1.. <i>GPL.Gc_uiNumOfParamDINTRecipeIngredients</i>] OF <i>ST_Parameter_DINT</i>	Tableau structuré d'informations sur les ingrédients pour des valeurs de type de données DINT

Utilisé par

- *ST_Recipe*

ST_Interface

Présentation

Type	Structure de données
Disponible à partir de	V1.0.1.0

Description

La structure *ST_Interface* est une collection de balises utilisées pour décrire les valeurs des commandes de communication entre les machines via les structures PackTag.

Éléments de la structure

Variable	Type de données	Description
<i>Number</i>	DINT	Nombre d'identification de la machine amont ou aval
<i>ControlCmdNumber</i>	DINT	Numéro de la commande de contrôle de la machine en amont ou aval
<i>CmdValue</i>	DINT	Valeur de la commande associée au numéro de commande de contrôle.
<i>Parameter</i>	ARRAY [0.. <i>GPL.Gc_uiNumOfParam</i>] OF <i>ST_Descriptor</i>	Tableau de <i>ST_Descriptor</i> , balises de paramètres associées à l'interface distante

Utilisé par

- *ST_Command*
- *ST_Status*

ST_ModeState_Times

Présentation

Type	Structure de données
Disponible à partir de	V1.4.2.0

Description

La structure *ST_ModeState_Times* est un ensemble de balises fournissant des informations pour le suivi de temps cumulé.

Éléments de la structure

Variable	Type de données	Description
<i>Mode</i>	DINT	Représente le temps cumulé (en secondes) passé dans chaque mode depuis la dernière réinitialisation du compteur et du temporisateur
<i>State</i>	ARRAY [1.. <i>GPL.Gc_uiMaxNumberOfModes</i>] OF DINT	Représente le temps cumulé (en secondes) passé dans chaque état d'un mode particulier depuis la dernière réinitialisation du compteur et du temporisateur

Utilisé par

- *ST_Cumulative_Times*

ST_Parameter_DINT

Présentation

Type	Structure de données
Disponible à partir de	V1.4.2.0

Description

La structure *ST_Parameter_DINT* fournit des informations sur des paramètres ou des valeurs de type de données DINT.

Éléments de la structure

Variable	Type de données	Description
<i>ID</i>	DINT	ID du paramètre
<i>Name</i>	STRING(<i>GPL.Gc_uiMaxLengthOfParameter-Name</i>)	Nom du paramètre
<i>Unit</i>	STRING[6]	Unité du paramètre
<i>Value</i>	REAL	Valeur du paramètre

Utilisé par

- *ST_Administration*
- *ST_Command*
- *ST_Status*
- *ST_Ingredients*
- *ST_ProcessVariables*

ST_Parameter_LREAL

Présentation

Type	Structure de données
Disponible à partir de	V1.4.2.0

Description

La structure *ST_Parameter_LREAL* fournit des informations sur des paramètres ou des valeurs de type de données LREAL.

Éléments de la structure

Variable	Type de données	Description
<i>ID</i>	DINT	ID du paramètre
<i>Name</i>	STRING(GPL.Gc_uiMaxLengthOfParameter-Name)	Nom du paramètre
<i>Unit</i>	STRING[6]	Unité du paramètre
<i>Value</i>	REAL	Valeur du paramètre

Utilisé par

- *ST_Administration*
- *ST_Command*
- *ST_Status*
- *ST_Ingredients*
- *ST_ProcessVariables*

ST_Parameter_REAL

Présentation

Type	Structure de données
Disponible à partir de	V1.4.2.0

Description

La structure *ST_Parameter_REAL* fournit des informations sur des paramètres ou des valeurs de type de données REAL.

Éléments de la structure

Variable	Type de données	Description
<i>ID</i>	DINT	ID du paramètre
<i>Name</i>	STRING(GPL.Gc_uiMaxLengthOfParameter-Name)	Nom du paramètre
<i>Unit</i>	STRING[6]	Unité du paramètre
<i>Value</i>	REAL	Valeur du paramètre

Utilisé par

- *ST_Administration*
- *ST_Command*
- *ST_Status*
- *ST_Ingredients*
- *ST_ProcessVariables*

ST_Parameter_STRING

Présentation

Type	Structure de données
Disponible à partir de	V1.4.2.0

Description

La structure *ST_Parameter_STRING* fournit des informations sur des paramètres ou des valeurs de type de données STRING.

Éléments de la structure

Variable	Type de données	Description
<i>ID</i>	DINT	ID du paramètre
<i>Name</i>	STRING(GPL.Gc_uiMaxLengthOfParameter-Name)	Nom du paramètre
<i>Unit</i>	STRING[6]	Unité du paramètre
<i>Value</i>	STRING	Valeur du paramètre

Utilisé par

- *ST_Administration*
- *ST_Command*
- *ST_Status*
- *ST_Ingredients*
- *ST_ProcessVariables*

ST_ProcessVariables

Présentation

Type	Structure de données
Disponible à partir de	V1.4.2.0

Description

La structure *ST_ProcessVariables* est un ensemble de balises fournissant des informations pour les variables de processus de recette.

Éléments de la structure

Variable	Type de données	Description
<i>Parameter_REAL</i>	ARRAY [1.. <i>GPL. Gc_uiNumOfParamREALRecipeProcessVar</i>] OF <i>ST_Parameter_REAL</i>	Tableau structuré d'informations sur les variables de processus de recette pour les valeurs de type de données REAL
<i>Parameter_STRING</i>	ARRAY [1.. <i>GPL. Gc_uiNumOfParamSTRINGRecipeProcessVar</i>] OF <i>ST_Parameter_STRING</i>	Tableau structuré d'informations sur les variables de processus de recette pour les valeurs de type de données STRING
<i>Parameter_LREAL</i>	ARRAY [1.. <i>GPL. Gc_uiNumOfParamLREALRecipeProcessVar</i>] OF <i>ST_Parameter_LREAL</i>	Tableau structuré d'informations sur les variables de processus de recette pour les valeurs de type de données LREAL
<i>Parameter_DINT</i>	ARRAY [1.. <i>GPL. Gc_uiNumOfParamDINTRecipeProcessVar</i>] OF <i>ST_Parameter_DINT</i>	Tableau structuré d'informations sur les variables de processus de recette pour les valeurs de type de données DINT

Utilisé par

- *ST_Recipe*

ST_Product

Présentation

Type	Structure de données
Disponible à partir de	V1.0.1.0

Description

La structure *ST_Product* est une collection de balises utilisées pour décrire le produit que la machine fabrique.

Éléments de la structure

Variable	Type de données	Description
<i>ProductID</i>	DINT	ID du produit, utilisé pour indiquer à la machine le produit qu'elle fabrique.
<i>ProcessVariable</i>	ARRAY [0.. <i>GPL.Gc_uiNumberOfProcessVariable</i>] OF <i>ST_Descriptor</i>	Le tableau structuré décrit les variables de processus spécifiques, qui sont nécessaires ou étaient utilisées par la machine pour traiter le produit spécifique.
<i>Ingredients</i>	ARRAY [0.. <i>GPL.Gc_uiNumberOfIngredients</i>] OF <i>ST_Ingredient</i>	Le tableau structuré contient les informations relatives aux matériaux bruts, qui sont nécessaires ou étaient utilisés par la machine pour fabriquer le produit spécifique.

Utilisé par

- *ST_Command*
- *ST_Status*

ST_Product_Data

Présentation

Type	Structure de données
Disponible à partir de	V1.4.2.0

Description

La structure *ST_Product_Data* est un ensemble de balises fournissant des informations sur les données de flux de produits.

Éléments de la structure

Variable	Type de données	Description
<i>ID</i>	DINT	ID de chaque flux de produits
<i>Name</i>	STRING	Nom du flux de produits
<i>Unit</i>	STRING[6]	Tableau structuré d'informations sur les variables de processus des ingrédients pour les valeurs de type de données LREAL
<i>PrimaryQty</i>	REAL	Valeur de quantité principale de produit pour chaque flux de produits
<i>ConsumedCount</i>	DINT	Nombre de consommables utilisés pour chaque flux de produits
<i>ProcessedCount</i>	DINT	Nombre de marchandises produites pour chaque flux de produits
<i>DefectiveCount</i>	DINT	Nombre de produits marqués comme non acceptables pour chaque flux de produits

Variable	Type de données	Description
<i>AccConsumedCount</i>	DINT	Nombre cumulé de consommables utilisés pour chaque flux de produits depuis la dernière réinitialisation
<i>AccProcessedCount</i>	DINT	Nombre cumulé de marchandises produites pour chaque flux de produits depuis la dernière réinitialisation
<i>AccDefectiveCount</i>	DINT	Nombre cumulé de produits marqués comme non acceptables pour chaque flux de produits depuis la dernière réinitialisation

Utilisé par

- *ST_Administration*

ST_Recipe

Présentation

Type	Structure de données
Disponible à partir de	V1.4.2.0

Description

La structure *ST_Recipe* est un ensemble de balises fournissant des informations sur les recettes

Éléments de la structure

Variable	Type de données	Description
<i>ID</i>	DINT	ID de <i>ST_Recipe</i>
<i>Name</i>	STRING	Nom de <i>ST_Recipe</i>
<i>Unit</i>	STRING[6]	Unité de <i>ST_Recipe</i>
<i>PrimaryQty</i>	REAL	Valeur de quantité principale de la recette
<i>ProcessVariables</i>	<i>ST_ProcessVariables</i>	Variables de processus pour la recette
<i>Ingredients</i>	<i>ST_Ingredients</i>	Informations sur les ingrédients de la recette

Utilisé par

- *ST_Command*
- *ST_Status*

ST_StateInfo

Présentation

Type	Structure de données
Disponible à partir de	V1.0.1.0

Description

La structure *ST_StateInfo* indique si l'état est défini pour un mode de fonctionnement donné et si un changement de mode de fonctionnement est possible. L'information est utilisée pour initialiser le gestionnaire des modes de fonctionnement.

Éléments de la structure

Variable	Type de données	Description
<i>xStateExistent</i>	BOOL	TRUE : Indique que l'état est défini pour le mode de fonctionnement donné.
<i>xModeChangePossible</i>	BOOL	TRUE : Indique qu'un changement de mode de fonctionnement est possible lorsque cet état est actif.

Utilisé par

- *ST_UnitModeDefinition*

ST_Timestamp

Présentation

Type	Structure de données
Disponible à partir de	V1.0.1.0

Description

La structure *ST_TimeStamp* contient la date et l'heure.

NOTE: La structure est devenue obsolète avec la révision 2015 de ANSI/ISA TR88.00.02. Cependant, il est conservé dans la bibliothèque afin de rester compatible avec les applications qui ont été créées avec les versions antérieures de cette bibliothèque.

Éléments de la structure

Variable	Type de données	Description
<i>AlmDate</i>	DATE	Contient l'intervalle en secondes depuis le 01.01.1970, 00:00.
<i>AlmTime</i>	TOD	Contient l'intervalle en millisecondes depuis 00:00.

Utilisé par

- *ST_Administration*
- *ST_Alarm*

Alias - *DateTimeArray*

Contenu de ce chapitre

DateTimeArray..... 59

DateTimeArray

Présentation

Type	Structure de données
Disponible à partir de	V1.1.0.0

Description

L'alias *DateTimeArray* représente un type de données complexe, qui est utilisé dans cette bibliothèque.

Éléments de la structure

Variable	Type de données	Type de données
<i>DateTimeArray</i>	ARRAY [0..6] OF DINT	Les éléments de données associés à la date et à l'heure doivent être au format ISO 8601. [0] = année, [1] = mois, [2] = jour, [3] = heure, [4] = min, [5] = sec, [6] = μ s (fraction de secondes)

Variables globales

Contenu de cette partie

Liste des constantes globales.....61

Liste des paramètres globaux.....62

Liste des constantes globales

Contenu de ce chapitre

GCL61

GCL

Présentation

Type :	Constantes globales
Disponible à partir de :	V1.0.1.0

Description

La liste des constantes globales contient les constantes globales de la bibliothèque PackML.

Constantes globales

Variable	Type de données	Valeur	Description
<i>Gc_sLibraryVersion</i>	STRING[80]	Vx.x.x.0 ¹	Version de bibliothèque
<i>Gc_uiMaxNumberOfStates</i>	UINT	17	Nombre maximum d'états
¹ Cette valeur varie selon la version de la bibliothèque.			

Liste des paramètres globaux

Contenu de ce chapitre

GPL 62

GPL

Présentation

Type :	Paramètres globaux
Disponible à partir de :	V1.0.1.0

Description

La liste des paramètres globaux contient les paramètres globaux de la bibliothèque PackML qui peuvent être remplacés pour s'adapter au projet dans le gestionnaire de bibliothèque.

Paramètres globaux

Variable	Type de données	Valeur	Description
<i>Gc_uiNumberOfInterface</i>	UINT[1..65535]	2	Nombre d'interfaces distantes, détermine le tableau <i>ST_Command.RemoteInterface</i>
<i>Gc_uiNumOfParam</i>	UINT[1..65535]	10	Nombre de paramètres pour une interface, détermine le tableau <i>ST_Interface.Parameter</i>
<i>Gc_uiNumOfAdminParameter</i>	UINT[1..65535]	100	Nombre de paramètres pour l'interface locale, détermine le tableau <i>ST_Administration.Parameter</i>
<i>Gc_uiNumberOfParameter</i>	UINT[1..65535]	50	Nombre de paramètres dans la structure d'état et de commande, détermine les tableaux <i>ST_Status.Parameter</i> et <i>ST_Command.Parameter</i>
<i>Gc_uiNumberOfProducts</i>	UINT[1..65535]	3	Nombre de produits pouvant être traités par la machine, détermine les tableaux <i>ST_Status.Product</i> et <i>ST_Command.Product</i>
<i>Gc_uiNumberOfConsumedProducts</i>	UINT[1..65535]	10	Nombre de produits consommés par la machine, détermine le tableau <i>ST_Administration.Prod ConsumedCount</i>
<i>Gc_uiNumberOfProcessedProducts</i>	UINT[1..65535]	10	Nombre de produits traités par la machine, détermine le tableau <i>ST_Administration.Prod ProcessedCount</i>
<i>Gc_uiNumberOfDefectiveProducts</i>	UINT[1..65535]	10	Nombre de produits marqués comme inacceptables dans la machine de production, détermine le tableau <i>ST_Administration.Prod DefectiveCount</i>

Variable	Type de données	Valeur	Description
<i>Gc_uiNumberOfProcessVariable</i>	UINT[1..65535]	50	Nombre de variables de processus par produit, détermine le tableau <i>ST_Product.ProcessVariable</i>
<i>Gc_uiNumberOfIngredientParameter</i>	UINT[1..65535]	50	Nombre de paramètres par ingrédient, détermine le tableau <i>ST_Ingredients.Parameter</i>
<i>Gc_uiNumberOfIngredients</i>	UINT[1..65535]	10	Nombre d'ingrédients par produit, détermine le tableau <i>ST_Product.Ingredients</i>
<i>Gc_uiMaxNumberOfAlarms</i>	UINT[1..65535]	100	Nombre maximum d'alarmes, détermine le tableau <i>ST_Administration.Alarm</i>
<i>Gc_uiNumberOfAlarmHistory</i>	UINT[1..65535]	100	Nombre maximum d'alarmes dans l'historique, détermine le tableau <i>ST_Administration.AlarmHistory</i>
<i>Gc_uiMaxNumberOfModes</i>	UINT[1..65535]	31	Nombre maximum de modes
<i>Gc_uiMaxLengthOfParameterName</i>	UINT[1..80]	80	Longueur maximum du nom de paramètre dans <i>ST_Descriptor</i>
<i>Gc_uiMaxNumberOfWarnings</i>	UINT[1..65535]	100	Nombre maximum de conseils, détermine le tableau <i>ST_Administration.Warning</i>
<i>Gc_uiMaxLengthOfAlarmMessage</i>	UINT[1..255]	80	Longueur maximum d'un message d'alarme défini avec <i>ST_Alarm</i>
<i>Gc_udiBgColorAlarmActive</i>	UDINT	FFCFCF hex	Utilisé pour la visualisation, code de couleurs RGB pour les alarmes actives, FFCFCF hex = rouge clair
<i>Gc_udiBgColorAlarmInactive</i>	UDINT	FFFF80 hex	Utilisé pour la visualisation, code de couleurs RGB pour les alarmes inactives et non acquittées, FFFF80 hex = jaune clair
<i>Gc_udiBgColorAlarmAcknowledged</i>	UDINT	AAD7FF hex	Utilisé pour la visualisation, code de couleurs RGB pour les alarmes inactives et acquittées, AAD7FF hex = bleu clair
<i>Gc_udiBgColorAlarmHistoryUnacknowledged</i>	UDINT	FFFF80 hex	Utilisé pour la visualisation, code de couleur RGB pour les alarmes non acquittées dans l'historique, FFFF80 hex = jaune clair
<i>Gc_udiBgColorWarningActive</i>	UDINT	FFCFCF hex	Utilisé pour la visualisation, code de couleur RGB pour les conseils actifs, FFCFCF hex = rouge clair
<i>Gc_udiBgColorWarningInactive</i>	UDINT	FFFF80 hex	Utilisé pour la visualisation, code de couleur RGB pour les conseils inactifs et non acquittés, FFFF80 hex = jaune clair
<i>Gc_udiBgColorWarningAcknowledged</i>	UDINT	AAD7FF hex	Utilisé pour la visualisation, code de couleurs RGB pour les conseils inactifs et acquittés, AAD7FF hex = bleu clair
<i>Gc_uiNumberOfParameterREAL</i>	UINT	10	Nombre de paramètres de type REAL
<i>Gc_uiNumberOfParameterSTRING</i>	UINT	10	Nombre de paramètres de type STRING
<i>Gc_uiNumberOfParameterLREAL</i>	UINT	10	Nombre de paramètres de type LREAL

Variable	Type de données	Valeur	Description
<i>Gc_uiNumberOfParameterDINT</i>	UINT	10	Nombre de paramètres de type DINT
<i>Gc_uiNumberOfRecipe</i>	UINT	3	Nombre de recettes
<i>Gc_uiNumberOfCumulativeTimes</i>	UINT	10	Nombre de fois cumulées
<i>Gc_uiNumOfParamREALRecipeProcessVar</i>	UINT	10	Nombre de paramètres de type REAL pour les variables de processus de recette
<i>Gc_uiNumOfParamSTRINGRecipeProcessVar</i>	UINT	10	Nombre de paramètres de type STRING pour les variables de processus de recette
<i>Gc_uiNumOfParamLREALRecipeProcessVar</i>	UINT	10	Nombre de paramètres de type LREAL pour les variables de processus de recette
<i>Gc_uiNumOfParamDINTRecipeProcessVar</i>	UINT	10	Nombre de paramètres de type DINT pour les variables de processus de recette
<i>Gc_uiNumOfParamREALRecipeIngredients</i>	UINT	10	Nombre de paramètres de type REAL pour les ingrédients de recette
<i>Gc_uiNumOfParamSTRINGRecipeIngredients</i>	UINT	10	Nombre de paramètres de type STRING pour les ingrédients de recette
<i>Gc_uiNumOfParamLREALRecipeIngredients</i>	UINT	10	Nombre de paramètres de type LREAL pour les ingrédients de recette
<i>Gc_uiNumOfParamDINTRecipeIngredients</i>	UINT	10	Nombre de paramètres de type DINT pour les ingrédients de recette

Interfaces

Contenu de cette partie

IF_UnitMode 66

IF_StateCommands 67

IF_StateModelHandler..... 75

IF_UnitMode

Contenu de ce chapitre

IF_UnitMode - Informations générales	66
--	----

IF_UnitMode - Informations générales

Présentation

Type :	Interface
Disponible à partir de :	V1.4.2.0
Hérite de :	–

Tâche

Interface permettant d'extraire les informations concernant le mode de contrôle d'unité actuel.

Description

Cette interface fournit des informations sur le mode de contrôle d'unité actuel.

Propriétés

Nom	Type de données	Accès	Description
<i>sName</i>	STRING	Lecture	Indique le nom du mode d'unité actuel.
<i>diValue</i>	DINT	Lecture	Indique la valeur numérique du mode d'unité actuel.
<i>diStatesDisabled</i>	DINT	Lecture	Fournit des informations sur le modèle d'état. Les bits 0 à 16 de DINT représentent les différents états définis dans <i>ET_StateModelDefinition</i> . La valeur de bit TRUE indique que l'état correspondant est désactivé dans le modèle d'état.
<i>diStatesMode-ChangeAllowed</i>	DINT	Lecture	Fournit des informations sur le modèle d'état. Les bits 0 à 16 de DINT représentent les différents états définis dans <i>ET_StateModelDefinition</i> . La valeur de bit TRUE indique qu'un changement de mode est autorisé dans l'état correspondant.
<i>etInitialState</i>	<i>ET_States</i> , page 34	Lecture	Indique l'état initial dans le modèle d'état configuré.

IF_StateCommands

Contenu de ce chapitre

<i>IF_StateCommands</i> - Informations générales	67
<i>CmdReset</i> (méthode)	67
<i>CmdStart</i> (méthode)	68
<i>CmdStop</i> (méthode)	69
<i>CmdHold</i> (méthode)	69
<i>CmdUnHold</i> (méthode)	70
<i>CmdSuspend</i> (méthode)	71
<i>CmdUnSuspend</i> (méthode)	71
<i>CmdAbort</i> (méthode)	72
<i>CmdClear</i> (méthode)	73
<i>StateComplete</i> (méthode)	73

IF_StateCommands - Informations générales

Présentation

Type :	Interface
Disponible à partir de :	V1.4.2.0
Hérite de :	—

Tâche

Interface permettant d'extraire les commandes d'état et les transitions d'état.

Description

Cette interface fournit les commandes d'état et les transitions d'état du modèle d'état de base PackML.

L'interface est implémentée par le bloc fonction *FB_UnitModeManager2*.

CmdReset (méthode)

Présentation

Type :	Interface
Disponible à partir de :	V1.4.2.0
Hérite de :	—

Description

Cette méthode est utilisée pour déclencher la commande d'état PackML *Reset*. Voir Commandes d'état, page 82.

Interface

Sortie	Type de données	Description
<i>q_etResult</i>	<i>ET_Result</i> , page 31	Fournit des informations de diagnostic et d'état sous la forme d'une valeur numérique. <i>If q_xError = FALSE</i> , <i>q_etResult</i> fournit des informations d'état. <i>If q_xError = TRUE</i> , <i>q_etResult</i> fournit des informations de diagnostic / d'erreur.
<i>q_sResultMsg</i>	STRING	Fournit des informations de diagnostic et d'état supplémentaires sous la forme d'un message textuel.

Valeur renvoyée

Sortie	Type de données	Description
<i>CmdReset</i>	BOOL	Indique TRUE si la méthode a abouti.

CmdStart (méthode)

Présentation

Type :	Interface
Disponible à partir de :	V1.4.2.0
Hérite de :	–

Description

Cette méthode est utilisée pour déclencher la commande d'état PackML *Start*.
Voir *Commandes d'état*, page 82.

Interface

Sortie	Type de données	Description
<i>q_etResult</i>	<i>ET_Result</i> , page 31	Fournit des informations de diagnostic et d'état sous la forme d'une valeur numérique. <i>If q_xError = FALSE</i> , <i>q_etResult</i> fournit des informations d'état. <i>If q_xError = TRUE</i> , <i>q_etResult</i> fournit des informations de diagnostic / d'erreur.
<i>q_sResultMsg</i>	STRING	Fournit des informations de diagnostic et d'état supplémentaires sous la forme d'un message textuel.

Valeur renvoyée

Sortie	Type de données	Description
<i>CmdStart</i>	BOOL	Indique TRUE si la méthode a abouti.

CmdStop (méthode)

Présentation

Type :	Interface
Disponible à partir de :	V1.4.2.0
Hérite de :	–

Description

Cette méthode est utilisée pour déclencher la commande d'état PackML *Stop*.
Voir [Commandes d'état](#), page 82.

Interface

Sortie	Type de données	Description
<i>q_etResult</i>	<i>ET_Result</i> , page 31	Fournit des informations de diagnostic et d'état sous la forme d'une valeur numérique. <i>If q_xError = FALSE</i> , <i>q_etResult</i> fournit des informations d'état. <i>If q_xError = TRUE</i> , <i>q_etResult</i> fournit des informations de diagnostic / d'erreur.
<i>q_sResultMsg</i>	STRING	Fournit des informations de diagnostic et d'état supplémentaires sous la forme d'un message textuel.

Valeur renvoyée

Sortie	Type de données	Description
<i>CmdStop</i>	BOOL	Indique TRUE si la méthode a abouti.

CmdHold (méthode)

Présentation

Type :	Interface
Disponible à partir de :	V1.4.2.0
Hérite de :	–

Description

Cette méthode est utilisée pour déclencher la commande d'état PackML *Hold*.
Voir Commandes d'état, page 82.

Interface

Sortie	Type de données	Description
<i>q_etResult</i>	<i>ET_Result</i> , page 31	Fournit des informations de diagnostic et d'état sous la forme d'une valeur numérique. <i>If q_xError = FALSE, q_etResult</i> fournit des informations d'état. <i>If q_xError = TRUE, q_etResult</i> fournit des informations de diagnostic / d'erreur.
<i>q_sResultMsg</i>	STRING	Fournit des informations de diagnostic et d'état supplémentaires sous la forme d'un message textuel.

Valeur renvoyée

Sortie	Type de données	Description
<i>CmdHold</i>	BOOL	Indique TRUE si la méthode a abouti.

CmdUnHold (méthode)

Présentation

Type :	Interface
Disponible à partir de :	V1.4.2.0
Hérite de :	–

Description

Cette méthode est utilisée pour déclencher la commande d'état PackML *UnHold*.
Voir Commandes d'état, page 82.

Interface

Sortie	Type de données	Description
<i>q_etResult</i>	<i>ET_Result</i> , page 31	Fournit des informations de diagnostic et d'état sous la forme d'une valeur numérique. <i>If q_xError = FALSE, q_etResult</i> fournit des informations d'état. <i>If q_xError = TRUE, q_etResult</i> fournit des informations de diagnostic / d'erreur.
<i>q_sResultMsg</i>	STRING	Fournit des informations de diagnostic et d'état supplémentaires sous la forme d'un message textuel.

Valeur renvoyée

Sortie	Type de données	Description
<i>CmdUnHold</i>	BOOL	Indique TRUE si la méthode a abouti.

CmdSuspend (méthode)

Présentation

Type :	Interface
Disponible à partir de :	V1.4.2.0
Hérite de :	–

Description

Cette méthode est utilisée pour déclencher la commande d'état PackML *Suspend*. Voir [Commandes d'état](#), page 82.

Interface

Sortie	Type de données	Description
<i>q_etResult</i>	<i>ET_Result</i> , page 31	Fournit des informations de diagnostic et d'état sous la forme d'une valeur numérique. <i>If q_xError = FALSE</i> , <i>q_etResult</i> fournit des informations d'état. <i>If q_xError = TRUE</i> , <i>q_etResult</i> fournit des informations de diagnostic / d'erreur.
<i>q_sResultMsg</i>	STRING	Fournit des informations de diagnostic et d'état supplémentaires sous la forme d'un message textuel.

Valeur renvoyée

Sortie	Type de données	Description
<i>CmdSuspend</i>	BOOL	Indique TRUE si la méthode a abouti.

CmdUnSuspend (méthode)

Présentation

Type :	Interface
Disponible à partir de :	V1.4.2.0
Hérite de :	–

Description

Cette méthode est utilisée pour déclencher la commande d'état PackML *UnSuspend*. Voir *Commandes d'état*, page 82.

Interface

Sortie	Type de données	Description
<i>q_etResult</i>	<i>ET_Result</i> , page 31	Fournit des informations de diagnostic et d'état sous la forme d'une valeur numérique. <i>If q_xError = FALSE, q_etResult</i> fournit des informations d'état. <i>If q_xError = TRUE, q_etResult</i> fournit des informations de diagnostic / d'erreur.
<i>q_sResultMsg</i>	STRING	Fournit des informations de diagnostic et d'état supplémentaires sous la forme d'un message textuel.

Valeur renvoyée

Sortie	Type de données	Description
<i>CmdUnSuspend</i>	BOOL	Indique TRUE si la méthode a abouti.

CmdAbort (méthode)

Présentation

Type :	Interface
Disponible à partir de :	V1.4.2.0
Hérite de :	–

Description

Cette méthode est utilisée pour déclencher la commande d'état PackML *Abort*. Voir *Commandes d'état*, page 82.

Interface

Sortie	Type de données	Description
<i>q_etResult</i>	<i>ET_Result</i> , page 31	Fournit des informations de diagnostic et d'état sous la forme d'une valeur numérique. <i>If q_xError = FALSE, q_etResult</i> fournit des informations d'état. <i>If q_xError = TRUE, q_etResult</i> fournit des informations de diagnostic / d'erreur.
<i>q_sResultMsg</i>	STRING	Fournit des informations de diagnostic et d'état supplémentaires sous la forme d'un message textuel.

Valeur renvoyée

Sortie	Type de données	Description
<i>CmdAbort</i>	BOOL	Indique TRUE si la méthode a abouti.

CmdClear (méthode)

Présentation

Type :	Interface
Disponible à partir de :	V1.4.2.0
Hérite de :	–

Description

Cette méthode est utilisée pour déclencher la commande d'état PackML *Clear*. Voir Commandes d'état, page 82.

Interface

Sortie	Type de données	Description
<i>q_etResult</i>	<i>ET_Result</i> , page 31	Fournit des informations de diagnostic et d'état sous la forme d'une valeur numérique. <i>If q_xError = FALSE</i> , <i>q_etResult</i> fournit des informations d'état. <i>If q_xError = TRUE</i> , <i>q_etResult</i> fournit des informations de diagnostic / d'erreur.
<i>q_sResultMsg</i>	STRING	Fournit des informations de diagnostic et d'état supplémentaires sous la forme d'un message textuel.

Valeur renvoyée

Sortie	Type de données	Description
<i>CmdClear</i>	BOOL	Indique TRUE si la méthode a abouti.

StateComplete (méthode)

Présentation

Type :	Interface
Disponible à partir de :	V1.4.2.0
Hérite de :	–

Description

La méthode est utilisée pour effectuer la transition d'un état d'action à l'état résultant conformément au modèle d'état.

Interface

Sortie	Type de données	Description
<i>q_etResult</i>	<i>ET_Result</i> , page 31	Fournit des informations de diagnostic et d'état sous la forme d'une valeur numérique. <i>If q_xError = FALSE, q_etResult</i> fournit des informations d'état. <i>If q_xError = TRUE, q_etResult</i> fournit des informations de diagnostic / d'erreur.
<i>q_sResultMsg</i>	STRING	Fournit des informations de diagnostic et d'état supplémentaires sous la forme d'un message textuel.

Valeur renvoyée

Sortie	Type de données	Description
<i>StateComplete</i>	BOOL	Indique TRUE si la méthode a abouti.

IF_StateModelHandler

Contenu de ce chapitre

IF_StateModelHandler - Informations générales 75

IF_StateModelHandler - Informations générales

Présentation

Type :	Interface
Disponible à partir de :	V1.4.2.0
Hérite de :	–

Description

L'interface est implémentée par le bloc fonction *FB_StateModelHandlerBase*.

Unités d'organisation de programme (POU)

Contenu de cette partie

Blocs fonction	77
Fonctions.....	104

Blocs fonction

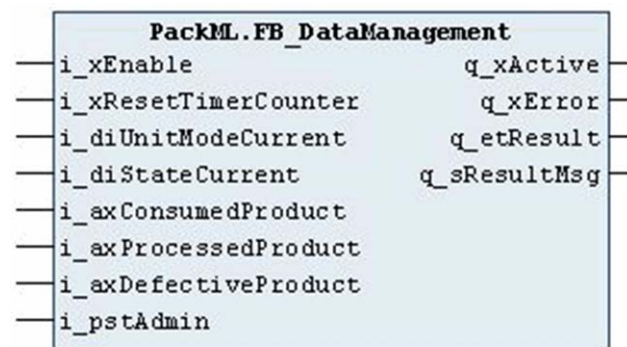
Contenu de ce chapitre

<i>FB_DataManagement</i>	77
<i>FB_ModeManager</i>	79
<i>FB_UnitModeManager2</i>	82
<i>FB_StateModelHandlerBase</i>	93
<i>FB_VisController</i>	102

FB_DataManagement

Présentation

Type :	Bloc fonction
Disponible à partir de :	V1.0.1.0



Description fonctionnelle

Le bloc fonction *FB_DataManagement* est utilisé pour enregistrer certaines données machine et mettre à jour les balises associées dans *ST_Administration*. La structure globale du type *ST_Administration* est transmise au bloc fonction via l'entrée *i_pstAdmin*.

Les valeurs temporelles sont mises à jour en permanence en fonction des valeurs actuelles du mode d'unité (*i_diUnitModeCurrent*) et de l'état de la machine (*i_diStateCurrent*). Les valeurs du mode de contrôle et de l'état de la machine correspondent aux définitions fournis pour les énumérations *ET_Modes*, page 31 et *ET_States*, page 34.

Les différents compteurs de produits sont déclenchés par les signaux correspondants dans les tableaux fournis par les entrées *i_ax...Product*.

Une réinitialisation globale des valeurs de temporisateur et de compteur peut être déclenchée via l'entrée *i_xResetTimerCounter*.

NOTE: Les valeurs de compteur fournies par la balise *AccCount* dans la structure *ST_CountDescrip* sont fournies en tant que compteurs sans réinitialisation ; elles ne sont donc pas remises à zéro par ce bloc fonction.

La conservation des données après des événements externes, par exemple un démarrage à chaud du contrôleur, n'est pas couverte par cette fonction.

Les balises d'administration suivantes sont traitées par le bloc fonction :

- *ST_Administration.ModeCurrentTime[#]*
- *ST_Administration.ModeCummulativeTime[#]*
- *ST_Administration.StateCurrentTime[#,#]*

- *ST_Administration.StateCummulativeTime[#,#]*
- *ST_Administration.ProdConsumedCount[#]*
- *ST_Administration.ProdProcessedCount[#]*
- *ST_Administration.ProdDefectiveCount[#]*
- *ST_Administration.AccTimeSinceReset*
- *ST_Administration.AlarmExtent*
- *ST_Administration.AlarmHistoryExtent*
- *ST_Administration.WarningExtent*
- *ST_Administration.PLCDateTime*
- *ST_Administration.PACDateTime*
- *ST_Adminsitration.ModeTimeCurrent*
- *ST_Adminsitration.StateTimeCurrent*

Interface

Entrée	Type de données	Description
<i>i_xEnable</i>	BOOL	Activation et initialisation du bloc fonction.
<i>i_xResetTimerCounter</i>	BOOL	Sur un front montant, les valeurs des temporisateurs et compteurs sont réinitialisées.
<i>i_diUnitModeCurrent</i>	DINT	Le mode de fonctionnement actuel est transmis au bloc fonction via cette entrée. Le paramètre <i>Status.UnitModeCurrent</i> doit être appliqué à l'entrée.
<i>i_diStateCurrent</i>	DINT	L'état actuel de la machine est transmis au bloc fonction via cette entrée. Le paramètre <i>Status.StateCurrent</i> doit être appliqué à l'entrée.
<i>i_axConsumedProduct</i>	ARRAY [1..GPL. Gc_ uiNumberOfCon- sumedProducts] OF BOOL	Sur un front montant d'une variable booléenne du tableau, la valeur du compteur correspondant dans la balise d'administration est augmentée.
<i>i_axProcessedProduct</i>	ARRAY [1..GPL. Gc_ uiNumberOfCon- sumedProducts] OF BOOL	Sur un front montant d'une variable booléenne du tableau, la valeur du compteur correspondant dans la balise d'administration est augmentée.
<i>i_axDefectiveProduct</i>	ARRAY [1..GPL. Gc_ uiNumberOfDe- fectiveProducts] OF BOOL	Sur un front montant d'une variable booléenne du tableau, la valeur du compteur correspondant dans la balise d'administration est augmentée.
<i>i_pstAdmin</i>	POINTER TO ST_ Administration	Via cette entrée, l'adresse du pointeur vers les balises d'administration est transmise au bloc fonction.

Pour empêcher tout accès non autorisé pouvant être causé par un accès de pointeur à la mémoire, assurez-vous que le pointeur désigne une variable de type *ST_Administration*.

L'exécution de la commande **Changement en ligne** peut déplacer des variables dans la mémoire. Le décalage des variables peut avoir pour effet que les variables `POINTER` pointent vers une mémoire non valide. Par conséquent, assurez-vous que les pointeurs ne sont pas conservés d'un cycle à l'autre mais réaffectés à chaque cycle.

⚠ AVERTISSEMENT

FONCTIONNEMENT IMPRÉVU DE L'ÉQUIPEMENT

Attribuez la valeur des variables de type `POINTER TO` avant leur première utilisation dans un bloc fonction et lors de chaque cycle suivant.

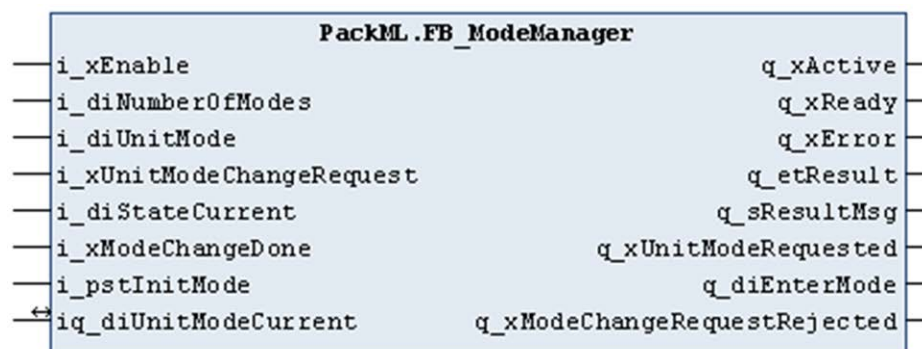
Le non-respect de ces instructions peut provoquer la mort, des blessures graves ou des dommages matériels.

Sortie	Type de données	Description
<code>q_xActive</code>	BOOL	Si cette sortie est définie sur TRUE, le bloc fonction est actif.
<code>q_xError</code>	BOOL	Si cette sortie est définie sur TRUE, une erreur a été détectée. Voir <i>ET_Result</i> .
<code>q_etResult</code>	<i>ET_Result</i>	Enumération avec le résultat.
<code>q_sResultMsg</code>	STRING[80]	Message de résultat supplémentaire.

FB_ModeManager

Présentation

Type :	Bloc fonction
Disponible à partir de :	V1.0.1.0



Description fonctionnelle

Le bloc fonction *FB_ModeManager* peut être utilisé pour gérer les demandes de changement de mode dans une application de machine basée sur le modèle d'états, page 23 PackML. A partir de la définition des modes de l'unité, le bloc fonction vérifie si le changement de mode demandé est possible ou non.

La définition des modes de l'unité inclut la définition des états possibles de chaque mode de contrôle disponible et l'état à partir duquel un changement de mode est possible. Pour l'exécution de la définition du mode de l'unité, les fonctions auxiliaires *FC_InitStateModelStateChange* et *FC_InitStateModelExistingStates* peuvent être utilisées.

Enfin, la définition du mode de l'unité doit être fournie dans un tableau de type *ST_UnitModeDefinition*. Les index du tableau correspondent à la valeur numérique des modes de contrôle disponibles (consultez *ET_Modes*).

La définition du mode de l'unité doit être effectuée au début (phase d'initialisation) d'une application, avant le premier appel avec `i_xEnable = TRUE` de l'instance du bloc fonction *FB_ModeManager*.

Sur un front montant de l'entrée *i_xEnable*, la phase d'initialisation du bloc fonction *FB_ModeManager* est exécutée. Lors de cette phase, les définitions des états de l'unité pour chaque mode de contrôle disponible sont vérifiées pour déterminer si un modèle d'état conforme à PackML peut être créé. Si l'initialisation est correctement effectuée, le bloc fonction est prêt à fonctionner, ce qui est indiqué par la sortie *q_xReady*.

Lors de l'initialisation, les conditions suivantes sont vérifiées :

- Nombre maximal d'états : 17
- Les états **Execute** et **Stopped** sont des états obligatoires pour chaque mode de contrôle.
- L'état **Completing** requiert l'état **Complete**.
- Les états **Holding** et **Unholding** requièrent dans chaque cas l'état **Held**.
- Les états **Suspending** et **UnSuspending** requièrent dans chaque cas l'état **Suspended**.
- Les états **Aborting** et **Clearing** requièrent dans chaque cas l'état **Aborted**.
- Les états **Resetting** et **Idle** peuvent être utilisés uniquement combinés.

Durant le fonctionnement normal, le bloc fonction vérifie la demande de changement de mode, qui est indiquée par les entrées *i_diUnitMode* et *i_xUnitModeChange*, en fonction du mode de contrôle actuel (*iq_diUnitModeCurrent*) et de l'état actuel (*i_diStateCurrent*). Seul un front montant sur *i_xUnitModeChange* déclenche l'exécution du bloc fonction. Le résultat est indiqué par les sorties *q_diEnterMode* et *q_xModeChangeRequestRejected*. Si le changement de mode est autorisé, la sortie *q_xModeChangeRequestRejected* indique FALSE et la sortie *q_diEnterMode* est mise à jour avec la valeur du mode demandé. Dans ce cas, les étapes requises pour changer le mode de contrôle doivent être exécutées en fonction de votre application. Ensuite les informations indiquant que le changement de mode a été exécuté doivent être transmises au bloc fonction via l'entrée *i_xModeChangeDone*. Si l'entrée indique TRUE, la variable liée à l'entrée/sortie *iq_diUnitModeCurrent* est mise à jour avec le mode de contrôle demandé précédemment.

Interface

Entrée	Type de données	Description
<i>i_xEnable</i>	BOOL	Activation et initialisation du bloc fonction.
<i>i_diNumberOfModes</i>	DINT	Nombre de modes de fonctionnement Si la valeur est modifiée, la réinitialisation du bloc fonction est nécessaire.
<i>i_diUnitMode</i>	DINT	Mode de fonctionnement demandé PackTag <i>Command.UnitMode</i> doit être appliqué à l'entrée.
<i>i_xUnitModeChangeRequest</i>	BOOL	Sur un front montant, le bloc fonction vérifie si un changement de mode de fonctionnement est possible. PackTag <i>Command.UnitModeChangeRequest</i> doit être appliqué à l'entrée.
<i>i_diStateCurrent</i>	DINT	Etat actuel de la machine PackTag <i>Status.StateCurrent</i> doit être appliqué à l'entrée.

Entrée	Type de données	Description
<i>i_xModeChangeDone</i>	BOOL	Retour d'information de l'application indiquant que le changement de mode demandé a été exécuté. Sur un front montant, la variable liée à <i>iq_diUnitModeCurrent</i> est mise à jour en conséquence.
<i>i_pstInitMode</i>	POINTER TO <i>ST_UnitModeDefinition</i>	Via cette entrée, l'adresse du pointeur des définitions du mode de l'unité est transférée au bloc fonction.* * La définition du mode de l'unité fournit l'état de chaque mode de fonctionnement disponible et doit être fourni dans un tableau de <i>ST_UnitModeDefinition</i> . Les index du tableau correspondent à la valeur numérique des modes de contrôle disponibles (consultez <i>ET_Modes</i>). Par conséquent, le pointeur doit pointer sur l'index qui est associé au mode de fonctionnement <i>Producing</i> , qui est le premier mode.

Entrée/sortie	Type de données	Description
<i>iq_diUnitModeCurrent</i>	DINT	Via la variable liée à cette entrée/sortie, le bloc fonction obtient le mode de fonctionnement actuel lors d'une demande de changement, et il écrit le nouveau mode de fonctionnement une fois le changement exécuté. <i>PackTag Status.UnitModeCurrent</i> doit être appliqué à l'entrée/sortie.

Sortie	Type de données	Description
<i>q_xActive</i>	BOOL	Si cette sortie est définie sur TRUE, le bloc fonction est actif.
<i>q_xError</i>	BOOL	Si cette sortie est TRUE, une erreur a été détectée. Consultez <i>ET_Result</i> .
<i>q_xReady</i>	BOOL	Si cette sortie est définie sur TRUE, le bloc fonction est prêt à fonctionner.
<i>q_etResult</i>	<i>ET_Result</i>	Énumération avec le résultat.
<i>q_sResultMsg</i>	STRING[80]	Message de résultat supplémentaire.
<i>q_xUnitModeRequested</i>	BOOL	Renvoie l'entrée <i>i_xUnitModeChangeRequest</i> . <i>PackTag Status.UnitModeRequested</i> doit être appliqué à cette sortie.
<i>q_diEnterMode</i>	DINT	Indique le mode de fonctionnement pour lequel on veut changer. Si le bloc fonction est désactivé, la sortie est définie sur 0.
<i>q_xModeChangeRequestRejected</i>	BOOL	Tant que cette sortie indique TRUE, une demande de changement de mode n'est pas autorisée par ce bloc fonction et ne doit pas être effectuée.

FB_UnitModeManager2

FB_UnitModeManager2 - Informations générales

Présentation

Type :	Bloc fonction
Disponible à partir de :	V1.4.2.0

Description fonctionnelle

Le bloc fonction *FB_UnitModeManager2* est utilisé pour mettre en œuvre différents modes de contrôle d'unité dans une application. Ce bloc fonction traite les commandes de commutation de mode d'unité et les commandes de contrôle du modèle d'état pour le mode de contrôle d'unité actuel.

Les commandes sont déclenchées par les méthodes fournies par le bloc.

Lorsqu'une commutation de mode de contrôle d'unité est demandée, son autorisation dans l'état actuel est vérifiée. Si la commutation de mode est autorisée, le mode de contrôle d'unité demandé est défini. Dans le cas contraire, le mode actuel est conservé et un message d'erreur est émis.

Les commandes d'état sont utilisées pour effectuer les transitions d'état du modèle d'état actif. Si une commande est déclenchée, son autorisation dans l'état actuel est vérifiée en fonction du modèle d'état configuré. Si la commande est autorisée, elle est acceptée et l'état résultant est défini. Si la commande n'est pas autorisée, un message d'erreur correspondant est émis.

Les différents modes de contrôle d'unité doivent être spécifiés à l'aide de la méthode *DefineUnitMode()*. Le bloc fonction *FB_UnitModeManager2* prend en charge jusqu'à 31 modes d'unité.

Changement de mode

Les conditions suivantes doivent être remplies pour un changement de mode de contrôle d'unité :

- Le mode de contrôle d'unité souhaité est connu du *FB_UnitModeManager2*. Un mode de contrôle d'unité est identifié par sa valeur numérique.
- La sortie du mode de contrôle d'unité actuel est autorisée dans l'état actuel.
- L'état actuel existe également dans le modèle d'état du mode de contrôle d'unité cible. Les transitions entre les modes de contrôle d'unité s'effectuent uniquement dans un état commun aux deux modes actuel et cible.

Commandes d'état

Les méthodes fournies pour déclencher les commandes d'état ou les transitions d'état correspondent au modèle d'état de base PackML tel que défini dans ANSI/ISA TR88.00.02 – 2022. Voir [Modèle d'état de base PackML](#), page 83.

La méthode *StateComplete()* est utilisée pour effectuer la transition de l'état d'action actuel à l'état suivant en fonction du modèle d'état défini. Voir [StateComplete \(méthode\)](#), page 73.

Les méthodes *Cmd<command name>()* permettent de déclencher la commande d'état correspondante. Selon l'état actuel et le modèle d'état configuré, la transition d'état est effectuée ou la commande est rejetée. Voir [IF_StateCommands - Informations générales](#), page 67.

Commandes d'état prises en charge et leurs réactions

Méthode de commande	Description
<i>CmdReset()</i>	Commande acceptée uniquement dans les états <i>Stopped</i> et <i>Complete</i> . Réaction : Transition vers l'état <i>Resetting</i> .
<i>CmdStart()</i>	Commande acceptée uniquement dans les états <i>Idle</i> ou dans l'état <i>Stopped</i> si l'état <i>Idle</i> n'existe pas. Réaction : Transition vers l'état <i>Starting</i> .
<i>CmdHold()</i>	Commande acceptée uniquement dans les états <i>Execute</i> et <i>Suspended</i> . Réaction : Transition vers l'état <i>Holding</i> .
<i>CmdUnHold()</i>	Commande acceptée uniquement dans l'état <i>Held</i> . Réaction : Transition vers l'état <i>Unholding</i> .
<i>CmdSuspend()</i>	Commande acceptée uniquement dans l'état <i>Execute</i> . Réaction : Transition vers l'état <i>Suspending</i> .
<i>CmdUnSuspend()</i>	Commande acceptée uniquement dans l'état <i>Suspended</i> . Réaction : Transition vers l'état <i>Unsuspending</i> .
<i>CmdComplete()</i>	Commande acceptée uniquement dans les états <i>Execute</i> , <i>Held</i> et <i>Suspended</i> . Réaction : Transition vers l'état <i>Completing</i> .
<i>CmdAbort()</i>	Commande acceptée dans tous les états sauf <i>Aborting</i> et <i>Aborted</i> . Réaction : Transition vers l'état <i>Aborting</i> .
<i>CmdClear()</i>	Commande acceptée uniquement dans l'état <i>Aborted</i> . Réaction : Transition vers l'état <i>Clearing</i> .
<i>CmdStop()</i>	Commande acceptée dans tous les états sauf <i>Aborting</i> , <i>Aborted</i> , <i>Clearing</i> , <i>Stopping</i> et <i>Stopped</i> . Réaction : Transition vers l'état <i>Stopping</i> .
<i>StateComplete()</i>	Commande acceptée uniquement dans les états <i>Resetting</i> , <i>Starting</i> , <i>Holding</i> , <i>Unholding</i> , <i>Suspending</i> , <i>Unsuspending</i> , <i>Aborting</i> , <i>Clearing</i> et <i>Stopping</i> . Réaction : Transition vers l'état suivant d'après le modèle d'état.

Applications multitâches

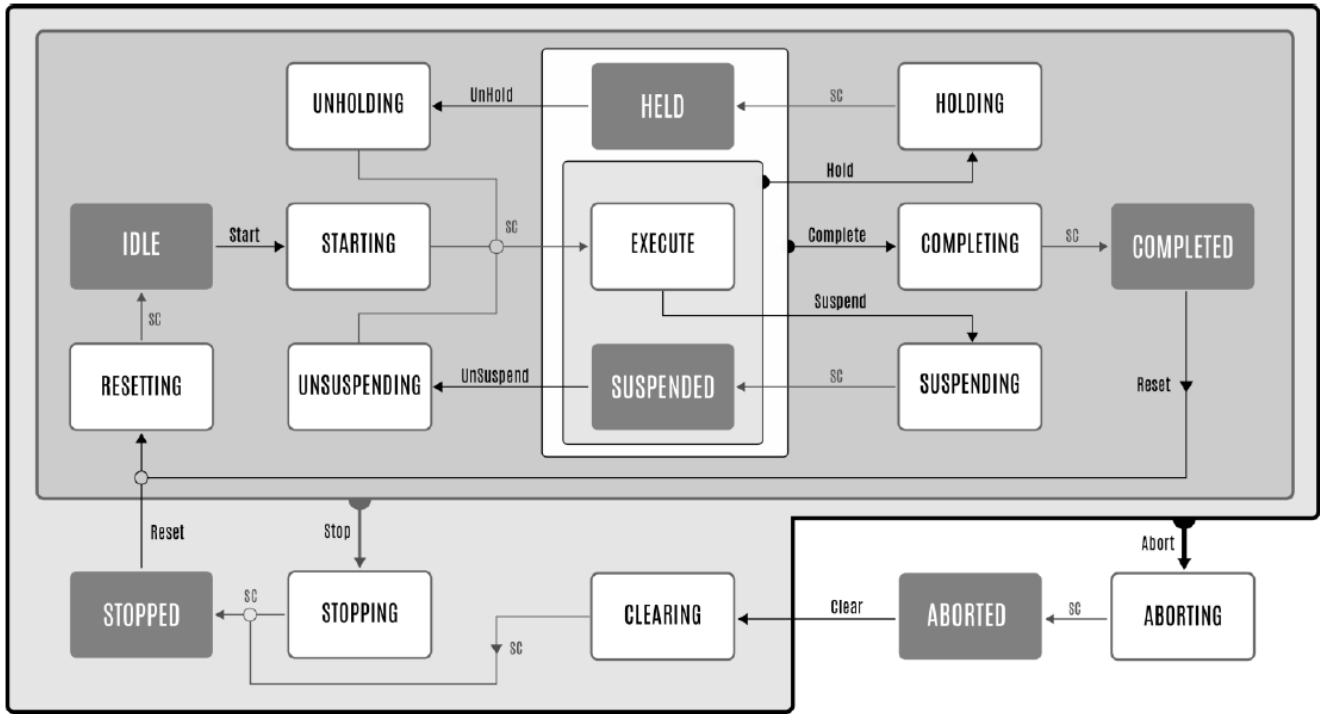
En raison de l'implémentation à l'aide de méthodes et de propriétés, il est possible que les méthodes soient appelées par différentes tâches. Pour assurer le traitement complet d'une commande, les méthodes correspondantes sont implémentées de telle sorte qu'une commande n'est acceptée que si aucune autre commande n'est en cours. Si une commande est rejetée en raison d'un traitement actif, la méthode renvoie un message de diagnostic correspondant.

Vous devez toutefois tenir compte du comportement des applications multitâches lorsque vous implémentez votre application.

Modèle d'état de base PackML

Présentation

Le schéma suivant illustre le modèle d'état de base PackML selon ANSI/ISA TR88.00.02 – 2022. Le modèle d'état de base représente l'ensemble complet des états, commandes d'état et transitions d'état définis.



SC : Etat terminé (transition d'un état d'action sur fond blanc à l'état suivant sur fond gris)

Propriétés

Présentation

Nom	Type de données	Accès	Description
<i>ifUnitModeCurrent</i>	IF_UnitMode, page 66	Lecture	Interface du mode d'unité actuel. Cette interface fournit des informations d'état sur le mode de contrôle d'unité actuel.
<i>etStateCurrent</i>	ET_States, page 34	Lecture	Indique l'état actuel.
<i>timUnitModeCurrentTime</i>	TIME	Lecture	Indique le temps écoulé dans le mode actuel.
<i>timStateCurrentTime</i>	TIME	Lecture	Indique le temps écoulé dans l'état actuel.

DefineUnitMode (méthode)

Présentation

Type :	Interface
Disponible à partir de :	V1.4.2.0
Hérite de :	–

Description

Cette méthode est utilisée pour définir un mode de contrôle d'unité dans *FB_UnitModeManager2*.

Pour chaque mode de contrôle d'unité, un modèle d'état individuel peut être défini à l'aide du paramètre *i_diStatesDisabled*. La méthode vérifie si le modèle d'état défini est plausible en fonction des règles de dépendance.

Interface

Entrée	Type de données	Description
<i>i_sName</i>	STRING	Spécifie le nom du mode de contrôle d'unité. Voir Modes de contrôle d'unité, page 85.
<i>i_diValue</i>	UDINT	Valeur numérique du mode de contrôle d'unité Reportez-vous au tableau ci-après. Voir Modes de contrôle d'unité, page 85. Plage valide : 1...31
<i>i_diStatesDisabled</i>	DINT	Ce paramètre permet de déterminer le modèle d'état associé au mode de contrôle d'unité. Les états 1 à 17 du modèle d'état PackML peuvent être désactivés via l'activation du bit correspondant dans le DINT. Les bits 0 à 16 représentent les différents états définis dans <i>ET_StateModelDefinition</i> , page 33. Reportez-vous aux règles de définition de modèle d'état (Règles de dépendance, page 86).
<i>i_diStatesModeChangeAllowed</i>	DINT	Ce paramètre permet de déterminer dans quel état un changement de mode d'unité est autorisé. Les états 1 à 17 du modèle d'état PackML pour lesquels un changement de mode est autorisé sont définis via l'activation du bit correspondant dans le DINT. Les bits 0 à 16 représentent les différents états définis dans <i>ET_StateModelDefinition</i> , page 33.
<i>i_etInitialState</i>	<i>ET_States</i> , page 34	Ce paramètre permet de déterminer l'état initial associé au mode d'unité. Cette entrée est facultative. En l'absence de valeur, l'état initial est défini sur <i>ET_States.Stopped</i> .

Sortie	Type de données	Description
<i>q_etResult</i>	<i>ET_Result</i> , page 31	Fournit des informations de diagnostic et d'état sous la forme d'une valeur numérique. <i>If q_xError = FALSE, q_etResult</i> fournit des informations d'état. <i>If q_xError = TRUE, q_etResult</i> fournit des informations de diagnostic / d'erreur.
<i>q_sResultMsg</i>	STRING	Fournit des informations de diagnostic et d'état supplémentaires sous la forme d'un message textuel.

Valeur renvoyée

Sortie	Type de données	Description
<i>DefineUnitMode</i>	BOOL	Indique TRUE si la méthode a abouti.

Modes de contrôle d'unité

La norme ANSI/ISA TR88.00.02-2015 définit un ensemble de modes de contrôle d'unité. Ils sont répertoriés dans le tableau ci-dessous. Les noms et les valeurs correspondantes sont réservés dans *FB_UnitModeManager2*.

Mode de contrôle d'unité	Valeur	Description
Non valide	0	Valeur de mode de contrôle d'unité non valide.
Production ⁽¹⁾	1	Ce mode est utilisé pour la production régulière.
Maintenance	2	Ce mode peut permettre au personnel autorisé d'exploiter une machine individuelle indépendamment des autres machines au sein d'une ligne de production.
Manuel	3	Ce mode permet de contrôler directement des modules de machine individuels.
Définissable par utilisateur	4 à 31	Etats définissables par l'utilisateur
(1) Avec la révision 2015 de ANSI/ISA TR88.00.02-2015, le mode de contrôle <i>Producing</i> est devenu <i>Production</i> . Pour des raisons de compatibilité avec la version précédente de la bibliothèque PackML, le nom <i>Producing</i> a été conservé.		

Règles de dépendance

Règles de dépendance pour la définition d'un modèle d'état selon ANSI/ISA TR88.00.02-2022 :

- Les états *Stopped*, *Aborted* et *Execute* sont obligatoires.
- Si l'état *Resetting* existe, l'état *Idle* doit exister. Toutefois, l'état *Idle* ne requiert pas nécessairement l'état *Resetting*.
- Si l'état *Completing* existe, l'état *Complete* doit exister.
- Si les états *Holding* ou *UnHolding* existent, l'état *Held* doit exister.
- Si les états *Suspending* ou *UnSuspending* existent, l'état *Suspended* doit exister.

DefineUnitModeWithHandler (méthode)

Présentation

Type :	Interface
Disponible à partir de :	V1.4.2.0
Hérite de :	—

Description

La méthode *DefineUnitModeWithHandler* a la même fonctionnalité que la méthode *DefineUnitMode()*. Toutefois, la méthode *DefineUnitModeWithHandler* offre en outre la possibilité de transmettre une instance du bloc fonction *FB_StateModelHandlerBase*. Cela permet à *FB_UnitModeManager2* d'accéder à cette instance et, en utilisant la méthode *ExecuteCurrentState()*, la méthode d'état correspondante de l'instance est appelée.

Interface

Entrée	Type de données	Description
<i>i_sName</i>	STRING	Reportez-vous à l'interface de <i>DefineUnitMode</i> (méthode), page 84.
<i>i_diValue</i>	UDINT	
<i>i_diStatesDisabled</i>	DINT	

Entrée	Type de données	Description
<i>i_diStatesModeChangeAllowed</i>	DINT	
<i>i_etInitialState</i>	<i>ET_States</i> , page 34	
<i>i_ifStateModelHandler</i>	<i>IF_StateModelHandler</i> , page 75	Ce paramètre est utilisé pour transmettre l'interface à l'instance de <i>FB_StateModelHandlerBase</i> .

Sortie	Type de données	Description
<i>q_etResult</i>	<i>ET_Result</i> , page 31	Fournit des informations de diagnostic et d'état sous la forme d'une valeur numérique. <i>If q_xError = FALSE</i> , <i>q_etResult</i> fournit des informations d'état. <i>If q_xError = TRUE</i> , <i>q_etResult</i> fournit des informations de diagnostic / d'erreur.
<i>q_sResultMsg</i>	STRING	Fournit des informations de diagnostic et d'état supplémentaires sous la forme d'un message textuel.

Valeur renvoyée

Sortie	Type de données	Description
<i>DefineUnitModeWithHandler</i>	BOOL	Indique TRUE si la méthode a abouti.

RegisterLoggerPoint (méthode)

Présentation

Type :	Interface
Disponible à partir de :	V1.4.2.0
Hérite de :	—

Description

Avec la méthode *RegisterLoggerPoint*, le bloc fonction *FB_UnitModeManager2* est enregistré en tant que source de journalisation dans le journaliseur d'application global.

Le nom de la source de journalisation dans le journaliseur d'application est défini par l'entrée *i_sName*.

L'entrée *i_ifParent* indique la source de journalisation parente sous laquelle la source de journalisation du bloc fonction. *FB_UnitModeManager2* est enregistrée dans l'arborescence des sources de journalisation.

Lorsque l'exécution réussit, la méthode renvoie la valeur TRUE.

Pour plus d'informations sur le journaliseur d'application, consultez la documentation sur l'utilisation de la bibliothèque ApplicationLogger (voir EcoStruxure Machine Expert - Guide de la bibliothèque ApplicationLogger).

NOTE: Condition préalable à l'utilisation de la méthode *RegisterLoggerPoint* () : le plug-in **Journaliseur d'application** doit être ajouté au projet et le service **Journaliseur d'application** doit être enregistré.

Une fois la source de journalisation enregistrée, vous devez définir le niveau de journalisation souhaité à l'aide de la méthode *SetApplicationLoggerLogLevel()*.

FB_UnitModeManager2 prend en charge la création d'entrées de journal pour les événements suivants :

Événement	Niveau de journalisation
L'exécution de la méthode <i>DefineUnitMode()</i> a échoué.	<i>APL.ET_LogLevel.Warning</i>
Une commande d'état est déclenchée.	<i>APL.ET_LogLevel.DebugMessage</i>
Une commande d'état est exécutée.	<i>APL.ET_LogLevel.StatusMessage</i>
Une commande d'état est rejetée.	<i>APL.ET_LogLevel.Warning</i>
Une commande de changement de mode est déclenchée.	<i>APL.ET_LogLevel.DebugMessage</i>
Un changement de mode est effectué.	<i>APL.ET_LogLevel.StatusMessage</i>
Une commande de changement de mode est rejetée.	<i>APL.ET_LogLevel.Warning</i>

Interface

Entrée	Type de données	Description
<i>i_ifParent</i>	<i>APL.IF_LoggerPoint</i>	Source de journalisation parente sous laquelle la source de journalisation du bloc fonction est enregistrée. Si la valeur d'entrée est 0, la source de journalisation globale <i>APL.GVL.G_ifApplicationLogger</i> est définie comme source parente.
<i>i_sName</i>	STRING	Nom de la source de journalisation affichée dans le journaliseur d'application.

Sortie	Type de données	Description
<i>q_etResult</i>	<i>ET_Result</i> , page 31	Fournit des informations de diagnostic et d'état sous la forme d'une valeur numérique. <i>If q_xError = FALSE, q_etResult fournit des informations d'état.</i> <i>If q_xError = TRUE, q_etResult fournit des informations de diagnostic / d'erreur.</i>
<i>q_sResultMsg</i>	STRING	Fournit des informations de diagnostic et d'état supplémentaires sous la forme d'un message textuel.

Valeur renvoyée

Sortie	Type de données	Description
<i>RegisterLoggerPoint</i>	BOOL	Indique TRUE si la méthode a abouti.

SetApplicationLoggerLogLevel (méthode)

Présentation

Type :	Interface
Disponible à partir de :	V1.4.2.0
Hérite de :	—

Description

Cette méthode est utilisée pour spécifier le niveau de journalisation de la source de journalisation du *FB_UnitModeManager2*. Le niveau de journalisation permet contrôler les types d'entrées de journal créées par le bloc fonction.

Pour plus d'informations sur les niveaux de journalisation, consultez la documentation sur la bibliothèque ApplicationLogger (voir EcoStruxure Machine Expert - Guide de la bibliothèque ApplicationLogger).

Interface

Entrée	Type de données	Description
<i>i_etLogLevel</i>	<i>APL.ET_LogLevel</i>	Niveau de journalisation du journaliseur d'application. Le niveau indique les types d'informations affichées dans les messages du journaliseur d'application.

Sortie	Type de données	Description
<i>q_etResult</i>	<i>ET_Result</i> , page 31	Fournit des informations de diagnostic et d'état sous la forme d'une valeur numérique. <i>If q_xError = FALSE, q_etResult</i> fournit des informations d'état. <i>If q_xError = TRUE, q_etResult</i> fournit des informations de diagnostic / d'erreur.
<i>q_sResultMsg</i>	STRING	Fournit des informations de diagnostic et d'état supplémentaires sous la forme d'un message textuel.

Valeur renvoyée

Sortie	Type de données	Description
<i>SetApplicationLoggerLogLevel</i>	BOOL	Indique TRUE si la méthode a abouti.

ExecuteCurrentState (méthode)

Présentation

Type :	Interface
Disponible à partir de :	V1.4.2.0
Hérite de :	—

Description

Si une instance de *FB_StateModelHandlerBase* est affectée au mode de contrôle de l'unité actuelle, cette méthode est utilisée pour appeler la méthode d'état de l'état actuel. Pour affecter une instance de *FB_StateModelHandlerBase* à un mode de contrôle d'unité, le mode de l'unité doit être défini à l'aide de la méthode *DefineUnitModeWithHandler()*.

Interface

Sortie	Type de données	Description
<i>q_etResult</i>	<i>ET_Result</i> , page 31	Fournit des informations de diagnostic et d'état sous la forme d'une valeur numérique. <i>If q_xError = FALSE, q_etResult fournit des informations d'état.</i> <i>If q_xError = TRUE, q_etResult fournit des informations de diagnostic / d'erreur.</i>
<i>q_sResultMsg</i>	STRING	Fournit des informations de diagnostic et d'état supplémentaires sous la forme d'un message textuel.

Valeur renvoyée

Sortie	Type de données	Description
<i>ExecuteCurrentState</i>	BOOL	Indique TRUE si la méthode a abouti.

CmdChangeUnitMode (méthode)

Présentation

Type :	Interface
Disponible à partir de :	V1.4.2.0
Hérite de :	–

Description

Cette méthode est utilisée pour modifier le mode de contrôle d'unité.

Interface

Entrée	Type de données	Description
<i>i_diValue</i>	<i>DINT</i>	Valeur numérique du mode de contrôle d'unité cible

Sortie	Type de données	Description
<i>q_etResult</i>	<i>ET_Result</i> , page 31	Fournit des informations de diagnostic et d'état sous la forme d'une valeur numérique. <i>If q_xError = FALSE, q_etResult fournit des informations d'état.</i> <i>If q_xError = TRUE, q_etResult fournit des informations de diagnostic / d'erreur.</i>
<i>q_sResultMsg</i>	STRING	Fournit des informations de diagnostic et d'état supplémentaires sous la forme d'un message textuel.

Valeur renvoyée

Sortie	Type de données	Description
<i>CmdChangeUnitMode</i>	BOOL	Indique TRUE si la méthode a abouti.

RemoveUnitMode (méthode)

Présentation

Type :	Interface
Disponible à partir de :	V1.4.2.0
Hérite de :	—

Description

Cette méthode est utilisée pour supprimer du bloc fonction un mode de contrôle d'unité défini.

Interface

Entrée	Type de données	Description
<i>i_diValue</i>	<i>DINT</i>	Valeur numérique du mode de contrôle d'unité

Sortie	Type de données	Description
<i>q_etResult</i>	<i>ET_Result</i> , page 31	Fournit des informations de diagnostic et d'état sous la forme d'une valeur numérique. <i>If q_xError = FALSE, q_etResult</i> fournit des informations d'état. <i>If q_xError = TRUE, q_etResult</i> fournit des informations de diagnostic / d'erreur.
<i>q_sResultMsg</i>	STRING	Fournit des informations de diagnostic et d'état supplémentaires sous la forme d'un message textuel.

Valeur renvoyée

Sortie	Type de données	Description
<i>RemoveUnitMode</i>	BOOL	Indique TRUE si la méthode a abouti.

GetDefinedUnitModes (méthode)

Présentation

Type :	Interface
Disponible à partir de :	V1.4.2.0
Hérite de :	—

Description

Cette méthode permet d'obtenir les définitions des modes de contrôle d'unité. Les informations sur les modes d'unité définis sont stockées dans le tableau affecté. Ce tableau peut être traité par d'autres fonctions de cette bibliothèque, par exemple par le bloc fonction *FB_VisController*.

Interface

Entrée/sortie	Type de données	Description
<i>iq_astUnitModeDefinitions</i>	ARRAY [1..GPL.Gc_uiMaxNumberOfModes] OF ST_UnitModeDefinition	Tableau stockant les informations sur les modes d'unité définis. Les modes sont stockés dans le tableau en fonction de leur valeur. Par exemple : Le mode d'unité de valeur 7 est stocké dans l'élément de tableau [7].

Sortie	Type de données	Description
<i>q_etResult</i>	ET_Result, page 31	Fournit des informations de diagnostic et d'état sous la forme d'une valeur numérique. If <i>q_xError</i> = FALSE, <i>q_etResult</i> fournit des informations d'état. If <i>q_xError</i> = TRUE, <i>q_etResult</i> fournit des informations de diagnostic / d'erreur.
<i>q_sResultMsg</i>	STRING	Fournit des informations de diagnostic et d'état supplémentaires sous la forme d'un message textuel.

Valeur renvoyée

Sortie	Type de données	Description
<i>GetDefinedUnitModes</i>	BOOL	Indique TRUE si la méthode a abouti.

Command (méthode)

Présentation

Type :	Interface
Disponible à partir de :	V1.4.2.0
Hérite de :	–

Description

Le bloc fonction *FB_UnitModeManager2* implémente l'interface *IF_StateCommands*. Il implémente par conséquent les méthodes de cette interface.

Pour plus d'informations sur les méthodes implémentées par l'interface, voir *IF_StateCommands*, page 67.

FB_StateModelHandlerBase

FB_StateModelHandlerBase - Informations générales

Présentation

Type :	Bloc fonction
Disponible à partir de :	V1.4.2.0

Description fonctionnelle

Le bloc fonction *FB_StateModelHandlerBase* est utilisé pour implémenter la logique de contrôle pour chaque état d'un modèle d'état.

Ce bloc fonction ne doit pas être instancié directement ; il doit être étendu par un bloc fonction personnalisé. Dans votre propre bloc fonction de gestion de modèle d'état, utilisez le mot clé *EXTENDS* en combinaison avec le nom du bloc fonction. Ainsi, votre bloc fonction hérite des méthodes, propriétés et variables locales du bloc fonction de base.

```

1 FUNCTION_BLOCK FB_StateModelManual EXTENDS PackML.FB_StateModelHandlerBase
2 VAR_INPUT

```

Le bloc fonction fournit les méthodes correspondant aux états 1 à 17 du modèle d'état de base PackML. Ces méthodes sont appelées dans l'application via la méthode *ExecuteCurrentState()* du bloc *FB_UnitModeManager2*. Pour cela, l'instance du bloc fonction qui étend *FB_StateModelHandlerBase* doit être affectée à un mode d'unité géré par *FB_UnitModeManager2*. La définition du mode d'unité doit être effectuée à l'aide de la méthode *FB_UnitModeManager2.DefineUnitModeWithHandler()*.

Les méthodes d'état de *FB_StateModelHandlerBase* ne contiennent pas la logique personnalisée. La logique personnalisée est ajoutée en remplaçant les méthodes sous le bloc d'extension.

Les entrées des méthodes d'état permettent d'accéder aux commandes d'état et au mode d'unité actuel.

Avec l'entrée *i_ifCommands*, il est possible d'émettre une commande d'état au sein des méthodes en appelant *FB_UnitModeManager2*.

L'entrée *i_ifUnitMode* donne accès aux informations sur le mode d'unité actuel. Il est ainsi possible d'effectuer certaines actions au sein de la méthode en fonction du mode d'unité. Cela permet d'utiliser le même gestionnaire de modèle d'état pour différents modes d'unité.

Une autre option consiste à créer plusieurs gestionnaires de modèle d'état personnalisés qui ont accès à une implémentation commune via l'extension ou l'héritage. Avec cette option, vous pouvez implémenter uniquement les différences en remplaçant la méthode concernée.

Clearing (méthode)

Présentation

Type :	Interface
Disponible à partir de :	V1.4.2.0
Hérite de :	—

Description

La méthode représente l'état *Clearing* (1) du modèle d'état PackML.

Interface

Entrée	Type de données	Description
<i>i_xIsFirstCycle</i>	BOOL	Indique s'il s'agit du premier appel de la méthode d'état.
<i>i_ifCommands</i>	<i>IF_StateCommands</i> , page 67	Donne accès aux commandes d'état. Cela vous permet de déclencher une commande à l'intérieur de la méthode d'état.
<i>i_ifUnitMode</i>	<i>IF_UnitMode</i> , page 66	Fournit des informations sur le mode de contrôle d'unité.

Stopped (méthode)

Présentation

Type :	Interface
Disponible à partir de :	V1.4.2.0
Hérite de :	–

Description

La méthode représente l'état *Stopped* (2) du modèle d'état PackML.

Interface

Entrée	Type de données	Description
<i>i_xIsFirstCycle</i>	BOOL	Indique s'il s'agit du premier appel de la méthode d'état.
<i>i_ifCommands</i>	<i>IF_StateCommands</i> , page 67	Donne accès aux commandes d'état. Cela vous permet de déclencher une commande à l'intérieur de la méthode d'état.
<i>i_ifUnitMode</i>	<i>IF_UnitMode</i> , page 66	Fournit des informations sur le mode de contrôle d'unité.

Starting (méthode)

Présentation

Type :	Interface
Disponible à partir de :	V1.4.2.0
Hérite de :	–

Description

La méthode représente l'état *Starting* (3) du modèle d'état PackML.

Interface

Entrée	Type de données	Description
<i>i_xlsFirstCycle</i>	BOOL	Indique s'il s'agit du premier appel de la méthode d'état.
<i>i_ifCommands</i>	<i>IF_StateCommands</i> , page 67	Donne accès aux commandes d'état. Cela vous permet de déclencher une commande à l'intérieur de la méthode d'état.
<i>i_ifUnitMode</i>	<i>IF_UnitMode</i> , page 66	Fournit des informations sur le mode de contrôle d'unité.

Idle (méthode)

Présentation

Type :	Interface
Disponible à partir de :	V1.4.2.0
Hérite de :	—

Description

La méthode représente l'état *Idle* (4) du modèle d'état PackML.

Interface

Entrée	Type de données	Description
<i>i_xlsFirstCycle</i>	BOOL	Indique s'il s'agit du premier appel de la méthode d'état.
<i>i_ifCommands</i>	<i>IF_StateCommands</i> , page 67	Donne accès aux commandes d'état. Cela vous permet de déclencher une commande à l'intérieur de la méthode d'état.
<i>i_ifUnitMode</i>	<i>IF_UnitMode</i> , page 66	Fournit des informations sur le mode de contrôle d'unité.

Suspended (méthode)

Présentation

Type :	Interface
Disponible à partir de :	V1.4.2.0
Hérite de :	—

Description

La méthode représente l'état *Suspended* (5) du modèle d'état PackML.

Interface

Entrée	Type de données	Description
<i>i_xIsFirstCycle</i>	BOOL	Indique s'il s'agit du premier appel de la méthode d'état.
<i>i_ifCommands</i>	<i>IF_StateCommands</i> , page 67	Donne accès aux commandes d'état. Cela vous permet de déclencher une commande à l'intérieur de la méthode d'état.
<i>i_ifUnitMode</i>	<i>IF_UnitMode</i> , page 66	Fournit des informations sur le mode de contrôle d'unité.

Execute (méthode)

Présentation

Type :	Interface
Disponible à partir de :	V1.4.2.0
Hérite de :	–

Description

La méthode représente l'état *Execute* (6) du modèle d'état PackML.

Interface

Entrée	Type de données	Description
<i>i_xIsFirstCycle</i>	BOOL	Indique s'il s'agit du premier appel de la méthode d'état.
<i>i_ifCommands</i>	<i>IF_StateCommands</i> , page 67	Donne accès aux commandes d'état. Cela vous permet de déclencher une commande à l'intérieur de la méthode d'état.
<i>i_ifUnitMode</i>	<i>IF_UnitMode</i> , page 66	Fournit des informations sur le mode de contrôle d'unité.

Stopping (méthode)

Présentation

Type :	Interface
Disponible à partir de :	V1.4.2.0
Hérite de :	–

Description

La méthode représente l'état *Stopping* (7) du modèle d'état PackML.

Interface

Entrée	Type de données	Description
<i>i_xlsFirstCycle</i>	BOOL	Indique s'il s'agit du premier appel de la méthode d'état.
<i>i_ifCommands</i>	<i>IF_StateCommands</i> , page 67	Donne accès aux commandes d'état. Cela vous permet de déclencher une commande à l'intérieur de la méthode d'état.
<i>i_ifUnitMode</i>	<i>IF_UnitMode</i> , page 66	Fournit des informations sur le mode de contrôle d'unité.

Aborting (méthode)

Présentation

Type :	Interface
Disponible à partir de :	V1.4.2.0
Hérite de :	—

Description

La méthode représente l'état *Aborting* (8) du modèle d'état PackML.

Interface

Entrée	Type de données	Description
<i>i_xlsFirstCycle</i>	BOOL	Indique s'il s'agit du premier appel de la méthode d'état.
<i>i_ifCommands</i>	<i>IF_StateCommands</i> , page 67	Donne accès aux commandes d'état. Cela vous permet de déclencher une commande à l'intérieur de la méthode d'état.
<i>i_ifUnitMode</i>	<i>IF_UnitMode</i> , page 66	Fournit des informations sur le mode de contrôle d'unité.

Aborted (méthode)

Présentation

Type :	Interface
Disponible à partir de :	V1.4.2.0
Hérite de :	—

Description

La méthode représente l'état *Aborted* (9) du modèle d'état PackML.

Interface

Entrée	Type de données	Description
<i>i_xIsFirstCycle</i>	BOOL	Indique s'il s'agit du premier appel de la méthode d'état.
<i>i_ifCommands</i>	<i>IF_StateCommands</i> , page 67	Donne accès aux commandes d'état. Cela vous permet de déclencher une commande à l'intérieur de la méthode d'état.
<i>i_ifUnitMode</i>	<i>IF_UnitMode</i> , page 66	Fournit des informations sur le mode de contrôle d'unité.

Holding (méthode)

Présentation

Type :	Interface
Disponible à partir de :	V1.4.2.0
Hérite de :	–

Description

La méthode représente l'état *Holding* (10) du modèle d'état PackML.

Interface

Entrée	Type de données	Description
<i>i_xIsFirstCycle</i>	BOOL	Indique s'il s'agit du premier appel de la méthode d'état.
<i>i_ifCommands</i>	<i>IF_StateCommands</i> , page 67	Donne accès aux commandes d'état. Cela vous permet de déclencher une commande à l'intérieur de la méthode d'état.
<i>i_ifUnitMode</i>	<i>IF_UnitMode</i> , page 66	Fournit des informations sur le mode de contrôle d'unité.

Held (méthode)

Présentation

Type :	Interface
Disponible à partir de :	V1.4.2.0
Hérite de :	–

Description

La méthode représente l'état *Held* (11) du modèle d'état PackML.

Interface

Entrée	Type de données	Description
<i>i_xlsFirstCycle</i>	BOOL	Indique s'il s'agit du premier appel de la méthode d'état.
<i>i_ifCommands</i>	<i>IF_StateCommands</i> , page 67	Donne accès aux commandes d'état. Cela vous permet de déclencher une commande à l'intérieur de la méthode d'état.
<i>i_ifUnitMode</i>	<i>IF_UnitMode</i> , page 66	Fournit des informations sur le mode de contrôle d'unité.

Restarting (méthode)

Présentation

Type :	Interface
Disponible à partir de :	V1.4.2.0
Hérite de :	—

Description

La méthode représente l'état *Restarting* (12) du modèle d'état PackML.

Interface

Entrée	Type de données	Description
<i>i_xlsFirstCycle</i>	BOOL	Indique s'il s'agit du premier appel de la méthode d'état.
<i>i_ifCommands</i>	<i>IF_StateCommands</i> , page 67	Donne accès aux commandes d'état. Cela vous permet de déclencher une commande à l'intérieur de la méthode d'état.
<i>i_ifUnitMode</i>	<i>IF_UnitMode</i> , page 66	Fournit des informations sur le mode de contrôle d'unité.

Suspending (méthode)

Présentation

Type :	Interface
Disponible à partir de :	V1.4.2.0
Hérite de :	—

Description

La méthode représente l'état *Suspending* (13) du modèle d'état PackML.

Interface

Entrée	Type de données	Description
<i>i_xIsFirstCycle</i>	BOOL	Indique s'il s'agit du premier appel de la méthode d'état.
<i>i_ifCommands</i>	<i>IF_StateCommands</i> , page 67	Donne accès aux commandes d'état. Cela vous permet de déclencher une commande à l'intérieur de la méthode d'état.
<i>i_ifUnitMode</i>	<i>IF_UnitMode</i> , page 66	Fournit des informations sur le mode de contrôle d'unité.

Unsuspending (méthode)

Présentation

Type :	Interface
Disponible à partir de :	V1.4.2.0
Hérite de :	–

Description

La méthode représente l'état *Unsuspending* (14) du modèle d'état PackML.

Interface

Entrée	Type de données	Description
<i>i_xIsFirstCycle</i>	BOOL	Indique s'il s'agit du premier appel de la méthode d'état.
<i>i_ifCommands</i>	<i>IF_StateCommands</i> , page 67	Donne accès aux commandes d'état. Cela vous permet de déclencher une commande à l'intérieur de la méthode d'état.
<i>i_ifUnitMode</i>	<i>IF_UnitMode</i> , page 66	Fournit des informations sur le mode de contrôle d'unité.

Resetting (méthode)

Présentation

Type :	Interface
Disponible à partir de :	V1.4.2.0
Hérite de :	–

Description

La méthode représente l'état *Resetting* (15) du modèle d'état PackML.

Interface

Entrée	Type de données	Description
<i>i_xlsFirstCycle</i>	BOOL	Indique s'il s'agit du premier appel de la méthode d'état.
<i>i_ifCommands</i>	<i>IF_StateCommands</i> , page 67	Donne accès aux commandes d'état. Cela vous permet de déclencher une commande à l'intérieur de la méthode d'état.
<i>i_ifUnitMode</i>	<i>IF_UnitMode</i> , page 66	Fournit des informations sur le mode de contrôle d'unité.

Completing (méthode)

Présentation

Type :	Interface
Disponible à partir de :	V1.4.2.0
Hérite de :	—

Description

La méthode représente l'état *Completing* (16) du modèle d'état PackML.

Interface

Entrée	Type de données	Description
<i>i_xlsFirstCycle</i>	BOOL	Indique s'il s'agit du premier appel de la méthode d'état.
<i>i_ifCommands</i>	<i>IF_StateCommands</i> , page 67	Donne accès aux commandes d'état. Cela vous permet de déclencher une commande à l'intérieur de la méthode d'état.
<i>i_ifUnitMode</i>	<i>IF_UnitMode</i> , page 66	Fournit des informations sur le mode de contrôle d'unité.

Complete (méthode)

Présentation

Type :	Interface
Disponible à partir de :	V1.4.2.0
Hérite de :	—

Description

La méthode représente l'état *Complete* (17) du modèle d'état PackML.

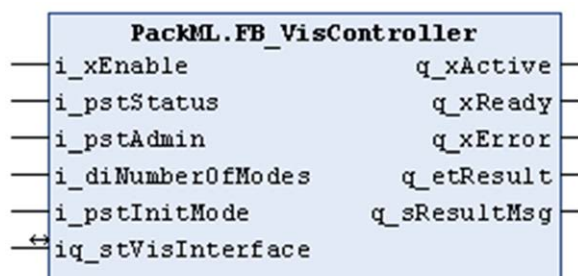
Interface

Entrée	Type de données	Description
<i>i_xIsFirstCycle</i>	BOOL	Indique s'il s'agit du premier appel de la méthode d'état.
<i>i_ifCommands</i>	<i>IF_StateCommands</i> , page 67	Donne accès aux commandes d'état. Cela vous permet de déclencher une commande à l'intérieur de la méthode d'état.
<i>i_ifUnitMode</i>	<i>IF_UnitMode</i> , page 66	Fournit des informations sur le mode de contrôle d'unité.

FB_VisController

Présentation

Type :	Bloc fonction
Disponible à partir de :	V1.0.1.0



Description fonctionnelle

Le *FB_VisController* permet de transférer des données (PackTags) de l'application dans la structure de données de type *ST_VisInterface* qui est utilisée par les cadres de visualisation fournis avec la bibliothèque PackML. De plus, le bloc fonction implémente des fonctions permettant de contrôler l'apparence de la visualisation. Par exemple : le défilement dans la liste d'alarmes et les couleurs d'arrière-plan des alarmes indiquant l'état.

Interface

Entrée	Type de données	Description
<i>i_xEnable</i>	BOOL	Activation et initialisation du bloc fonction.
<i>i_pstStatus</i>	POINTER TO <i>ST_Status</i>	Via cette entrée, l'adresse du pointeur sur PackTags du type <i>ST_Status</i> est transféré au bloc fonction.
<i>i_pstAdmin</i>	POINTER TO <i>ST_Administration</i>	Via cette entrée, l'adresse du pointeur su PackTags du type <i>ST_Administration</i> est transférée au bloc fonction.

Entrée	Type de données	Description
<i>i_diNumberOfModes</i>	DINT	Nombre de modes de fonctionnement. Si la valeur est modifiée, la réinitialisation du bloc fonction est nécessaire.
<i>i_pstInitMode</i>	POINTER TO <i>ST_UnitModeDefinition</i>	Via cette entrée, l'adresse du pointeur des définitions du mode de l'unité est transférée au bloc fonction.* * La définition du mode de l'unité fournit l'état de chaque mode de fonctionnement disponible et doit être fourni dans un tableau de <i>ST_UnitModeDefinition</i> . Les index du tableau correspondent à la valeur numérique des modes de contrôle disponibles (consultez <i>ET_Modes</i>). Par conséquent, le pointeur doit pointer sur l'index qui est associé au mode de fonctionnement Production, qui est le premier mode.

Entrée/sortie	Type de données	Description
<i>iq_stVisInterface</i>	<i>ST_VisInterface</i>	Via cette entrée / sortie, la variable est transmise au bloc fonction qui est lié aux visualisations fournies dans la bibliothèque PackML.

Sortie	Type de données	Description
<i>q_xActive</i>	BOOL	Si cette sortie est définie sur TRUE, le bloc fonction est actif.
<i>q_xReady</i>	BOOL	Si cette sortie est définie sur TRUE, le bloc fonction est prêt à fonctionner.
<i>q_xError</i>	BOOL	Si TRUE est attribué à cette sortie, une erreur a été détectée.
<i>q_etResult</i>	<i>ET_Result</i>	Résultat. Consultez <i>ET_Result</i> .
<i>q_sResultMsg</i>	STRING[80]	Message de résultat supplémentaire.

Fonctions

Contenu de ce chapitre

<i>FC_CheckCmd</i>	104
<i>FC_CheckCmd2</i>	105
<i>FC_EtResultToString</i>	106
<i>FC_GetDateTimeAsArray</i>	107
<i>FC_InitStateModelChangeStates</i>	107
<i>FC_InitStateModelExistingStates</i>	110
<i>FC_SetAlarm</i>	112
<i>FC_SetWarning</i>	113

FC_CheckCmd

Présentation

Type :	Fonction
Disponible à partir de :	V1.0.4.0



Description fonctionnelle

La fonction *FC_CheckCmd* est utilisée pour vérifier la plausibilité d'une commande de transition d'état (*i_diCmd*) dans l'état actuel (*i_diStateCurrent*) de la machine. La vérification de la plausibilité est effectuée en fonction de la définition du mode d'unité (*i_stUnitModeDefinition*), laquelle est fournie dans un tableau de type *ST_UnitModeDefinition*. La définition du mode d'unité est supposée validée (et jugée correcte) par le bloc fonction *FB_ModeManager*.

Si la commande est jugée plausible, la sortie *q_xCommandAcknowledge* indique TRUE et l'état cible est fourni sur la sortie *q_diStateRequested*.

FC_CheckCmd prend en charge le modèle d'état de base PackML tel que défini dans ANSI/ISA TR88.00.02-2015.

Le tableau suivant indique quelles commandes sont prises en charge dans quels états ainsi que l'état cible résultant provenant dérivé de la définition des états de l'unité.

Etat initial	Commande de transition	Etat cible
-	<i>Undefined</i>	Indéfini, la sortie <i>q_xCommandAcknowledge</i> est FALSE.
Execute	<i>Reset</i>	Resetting si les états Complete et Completing n'existent pas.
Complete	<i>Reset</i>	Resetting ou Starting , si l'état Resetting n'existe pas, ou Execute , si les états Resetting et Starting n'existent pas.
Stopped		
Idle	<i>Start</i>	Starting , ou Execute si Starting n'existe pas.
Stopped	<i>Start</i> (commande traitée uniquement si Idle n'existe pas)	Starting , ou Execute si Starting n'existe pas.

Etat initial	Commande de transition	Etat cible
N'importe quel état sauf Aborted , Aborting , Clearing , Stopping et Stopped	<i>Stop</i>	Stopping , ou Stopped si Stopping n'existe pas.
Execute	<i>Hold</i>	Holding , ou Held si Holding n'existe pas.
Held	<i>UnHold</i>	Un-Holding , ou Execute si Un-Holding n'existe pas.
Execute	<i>Suspend</i>	Suspending , ou Suspended si Suspending n'existe pas.
Suspended	<i>UnSuspend</i>	Un-Suspending , ou Execute si Un-Suspending n'existe pas.
Tout état sauf Aborted et Aborting	<i>Abort</i>	Aborting , ou Aborted si Aborting n'existe pas.
Aborted	<i>Clear</i>	Clearing , ou Stopped si Clearing n'existe pas.

Interface

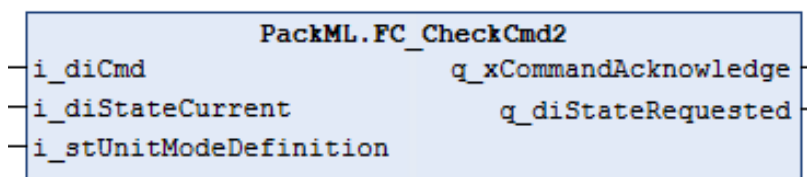
Entrées	Type de données	Description
<i>i_diCmd</i>	DINT	Commande de transition d'état ; PackTag <i>ST_Command.CntrlCmd</i> doit être appliqué à l'entrée.
<i>i_diStateCurrent</i>	DINT	Etat présent de la machine ; PackTag <i>ST_Status.StateCurrent</i> doit être appliqué à l'entrée.
<i>i_stUnitModeDefinition</i>	<i>ST_UnitModeDefinition</i>	Définition du mode de fonctionnement actuel ; PackTag <i>ST_Status.UnitModeCurrent</i> doit être utilisé comme variable d'index du tableau de <i>ST_UnitModeDefinition</i> .

Sortie	Type de données	Description
<i>q_xCommandAcknowledge</i>	BOOL	Indique TRUE si la commande indiquée est plausible.
<i>q_diStateRequested</i>	DINT	Fournit l'état cible si la commande indiquée est plausible.

FC_CheckCmd2

Présentation

Type :	Fonction
Disponible à partir de :	V1.4.2.0



Description fonctionnelle

La fonction *FC_CheckCmd2* est un développement ultérieur de la fonction *FC_CheckCmd*.

Par rapport à la fonction *FC_CheckCmd*, *FC_CheckCmd2* prend en charge le modèle d'état de base PackML tel que défini dans ANSI/ISA TR88.00.02-2022.

Pour la description fonctionnelle et la documentation de l'interface, reportez-vous à la fonction *FC_CheckCmd*, page 104.

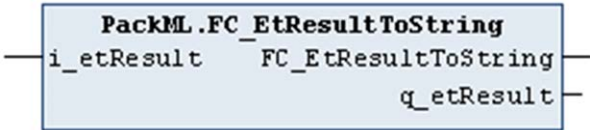
Le tableau suivant indique les commandes prises en charge dans chaque état et précise l'état cible résultant qui est dérivé de la définition d'état d'unité concernée.

Etat initial	Commande de transition	Etat cible
Tout état, ou Non défini	<i>Undefined</i>	Non défini , la sortie <i>q_xCommandAcknowledge</i> est FALSE.
Stopped ou Complete	<i>Reset</i>	Resetting , ou Idle si Resetting n'existe pas.
Stopped ou Idle	<i>Start</i>	La commande <i>Start</i> dans l'état Stopped est prise en charge uniquement si l'état Idle n'existe pas. Starting , ou Execute si Starting n'existe pas.
Execute ou Suspended	<i>Hold</i>	Holding , ou Held si Holding n'existe pas.
Execute	<i>Suspend</i>	Suspending , ou Suspended si Suspending n'existe pas.
Execute , Held ou Suspended	<i>Complete</i>	Completing , ou Completed si Completing n'existe pas.
Tout état sauf Aborting , Aborted , Clearing , Stopping et Stopped	<i>Stop</i>	Stopping , ou Stopped si Stopping n'existe pas.
Tout état sauf Aborting ou Aborted	<i>Abort</i>	Aborting , ou Aborted si Aborting n'existe pas.
Aborted	<i>Clear</i>	Clearing , ou Stopped si Clearing n'existe pas.

FC_EtResultToString

Présentation

Type :	Fonction
Disponible à partir de :	V1.0.1.0



Tâche

Convertir un élément d'énumération de type *ET_Result* en une valeur de chaîne

Description fonctionnelle

En utilisant la fonction *FC_EtResultToString*, vous pouvez convertir un élément d'énumération de type *ET_Result* en une valeur de chaîne.

Interface

Entrée	Type de données	Description
<i>i_etResult</i>	<i>ET_Result</i>	Énumération avec le résultat.

Sortie	Type de données	Description
<i>q_etResult</i>	<i>ET_Result</i>	Énumération avec le résultat.

Valeur retournée

Type de données	Description
STRING(80)	Le <i>ET_Result</i> converti en valeur de chaîne.

FC_GetDateTimeAsArray

Présentation

Type :	Fonction
Disponible à partir de :	V1.1.0.0

```
PackML.FC_GetDateTimeAsArray  
FC_GetDateTimeAsArray
```

Description fonctionnelle

La fonction *FC_GetDateTimeAsArray* permet d'obtenir l'horodatage (RTC) du contrôleur et de le convertir en données de type *DateTimeArray*.

La valeur de retour de la fonction est de type *DateTimeArray* et elle représente la valeur de l'horodateur du contrôleur.

FC_InitStateModelChangeStates

Présentation

Type :	Fonction
Disponible à partir de :	V1.0.1.0



Tâche

Utilisez la fonction *FC_InitStateModelChangeStates* pour définir, pour un mode de fonctionnement, de quel état un changement de mode de fonctionnement est possible.

Description fonctionnelle

A l'aide de l'entrée/sortie *iq_stStateModel*, une structure de type *ST_UnitModeDefinition* est transmise à la fonction.

Utilisez les 17 entrées de type *BOOL*, qui reflètent les états d'un mode de fonctionnement, pour définir les états desquels un changement de mode de fonctionnement est possible.

TRUE indique qu'un changement de mode de fonctionnement est possible ; *FALSE* indique que cet état ne permet pas de tenter un changement de mode de fonctionnement.

Suite à l'initialisation réussie d'un mode de fonctionnement, la fonction fournit un *TRUE*.

Interface

Entrée	Type de données	Description
<i>i_xStopping</i>	<i>BOOL</i>	Si <i>TRUE</i> est attribué à cette entrée, l'état Stopping est autorisé à changer le mode de fonctionnement.
<i>i_xStopped</i>	<i>BOOL</i>	Si <i>TRUE</i> est attribué à cette entrée, l'état Stopped est autorisé à changer le mode de fonctionnement.
<i>i_xResetting</i>	<i>BOOL</i>	Si <i>TRUE</i> est attribué à cette entrée, l'état Resetting est autorisé à changer le mode de fonctionnement.

Entrée	Type de données	Description
<i>i_xIdle</i>	BOOL	Si TRUE est attribué à cette entrée, l'état Idle est autorisé à changer le mode de fonctionnement.
<i>i_xStarting</i>	BOOL	Si TRUE est attribué à cette entrée, l'état Starting est autorisé à changer le mode de fonctionnement.
<i>i_xExecute</i>	BOOL	Si TRUE est attribué à cette entrée, l'état Execute est autorisé à changer le mode de fonctionnement.
<i>i_xHolding</i>	BOOL	Si TRUE est attribué à cette entrée, l'état Holding est autorisé à changer le mode de fonctionnement.
<i>i_xHeld</i>	BOOL	Si TRUE est attribué à cette entrée, l'état Held est autorisé à changer le mode de fonctionnement.
<i>i_xUnHolding</i>	BOOL	Si TRUE est attribué à cette entrée, l'état Un-Holding est autorisé à changer le mode de fonctionnement.
<i>i_xSuspending</i>	BOOL	Si TRUE est attribué à cette entrée, l'état Suspending est autorisé à changer le mode de fonctionnement.
<i>i_xSuspended</i>	BOOL	Si TRUE est attribué à cette entrée, l'état Suspended est autorisé à changer le mode de fonctionnement.
<i>i_xUnSuspending</i>	BOOL	Si TRUE est attribué à cette entrée, l'état Un-Suspending est autorisé à changer le mode de fonctionnement.
<i>i_xCompleting</i>	BOOL	Si TRUE est attribué à cette entrée, l'état Completing est autorisé à changer le mode de fonctionnement.
<i>i_xComplete</i>	BOOL	Si TRUE est attribué à cette entrée, l'état Complete est autorisé à changer le mode de fonctionnement.
<i>i_xAborting</i>	BOOL	Si TRUE est attribué à cette entrée, l'état Aborting est autorisé à changer le mode de fonctionnement.
<i>i_xAborted</i>	BOOL	Si TRUE est attribué à cette entrée, l'état Aborted est autorisé à changer le mode de fonctionnement.
<i>i_xClearing</i>	BOOL	Si TRUE est attribué à cette entrée, l'état Clearing est autorisé à changer le mode de fonctionnement.

Entrée/sortie	Type de données	Description
<i>iq_stStateModel</i>	<i>ST_UnitModeDefinition</i>	Cette structure renvoie le mode de fonctionnement qui doit être initialisé.

Valeur retournée

Type de données	Description
BOOL	TRUE si l'initialisation du mode de fonctionnement est un succès.

FC_InitStateModelExistingStates

Présentation

Type :	Fonction
Disponible à partir de :	V1.0.1.0



Tâche

Utilisez la fonction *FC_InitStateModelExistingStates* pour initialiser le modèle d'état et définir un mode de fonctionnement. En attribuant TRUE ou FALSE aux entrées, vous sélectionnez les états qui feront partie de votre modèle d'état.

Description fonctionnelle

A l'aide de l'entrée/sortie *iq_stStateModel*, une structure de type *ST_UnitModeDefinition* est transmise à la fonction.

Cette structure renvoie un mode de fonctionnement. En utilisant les 18 entrées de type BOOL qui reflètent chacune l'existence d'un mode de fonctionnement et ses états, il est possible de définir quels états constituent le mode de fonctionnement en question.

TRUE signifie que l'état existe ; FALSE signifie que l'état n'existe pas.

Suite à l'initialisation réussie d'un mode de fonctionnement, la fonction fournit un TRUE.

Interface

Entrée	Type de données	Description
<i>i_xModeExistent</i>	BOOL	Si TRUE est attribué à cette entrée, le mode de fonctionnement est disponible.
<i>i_xStopping</i>	BOOL	Si TRUE est attribué à cette entrée, l'état Stopping est disponible.
<i>i_xStopped</i>	BOOL	Si TRUE est attribué à cette entrée, l'état Stopped est disponible.
<i>i_xResetting</i>	BOOL	Si TRUE est attribué à cette entrée, l'état Resetting est disponible.
<i>i_xIdle</i>	BOOL	Si TRUE est attribué à cette entrée, l'état Idle est disponible.
<i>i_xStarting</i>	BOOL	Si TRUE est attribué à cette entrée, l'état Starting est disponible.
<i>i_xExecute</i>	BOOL	Si TRUE est attribué à cette entrée, l'état Execute est disponible.
<i>i_xHolding</i>	BOOL	Si TRUE est attribué à cette entrée, l'état Holding est disponible.
<i>i_xHeld</i>	BOOL	Si TRUE est attribué à cette entrée, l'état Held est disponible.
<i>i_xUnHolding</i>	BOOL	Si TRUE est attribué à cette entrée, l'état Un-Holding est disponible.
<i>i_xSuspending</i>	BOOL	Si TRUE est attribué à cette entrée, l'état Suspending est disponible.
<i>i_xSuspended</i>	BOOL	Si TRUE est attribué à cette entrée, l'état Suspended est disponible.
<i>i_xUnSuspending</i>	BOOL	Si TRUE est attribué à cette entrée, l'état Un-Suspending est disponible.
<i>i_xCompleting</i>	BOOL	Si TRUE est attribué à cette entrée, l'état Completing est disponible.
<i>i_xComplete</i>	BOOL	Si TRUE est attribué à cette entrée, l'état Complete est disponible.
<i>i_xAborting</i>	BOOL	Si TRUE est attribué à cette entrée, l'état Aborting est disponible.
<i>i_xAborted</i>	BOOL	Si TRUE est attribué à cette entrée, l'état Aborted est disponible.
<i>i_xClearing</i>	BOOL	Si TRUE est attribué à cette entrée, l'état Clearing est disponible.

Entrée/sortie	Type de données	Description
<i>iq_stStateModel</i>	<i>ST_UnitModeDefinition</i>	Cette structure renvoie le mode de fonctionnement qui doit être initialisé.

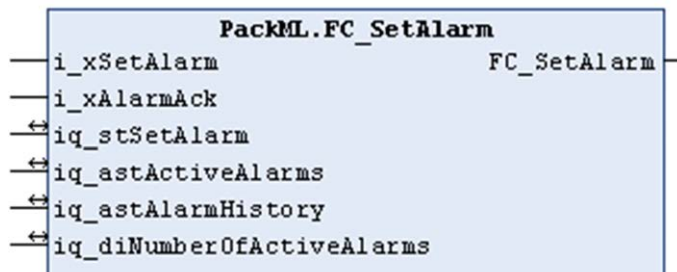
Valeur retournée

Type de données	Description
BOOL	TRUE si l'initialisation du mode de fonctionnement est un succès.

FC_SetAlarm

Présentation

Type :	Fonction
Disponible à partir de :	V1.0.1.0



Description fonctionnelle

La fonction *FC_SetAlarm* est une fonction auxiliaire qui permet d'écrire des messages d'alarme dans les balises d'administration appropriées, ainsi que de les supprimer. Une alarme est identifiée exclusivement par un identifiant unique et la valeur associée. L'identifiant et sa valeur sont définis par le paramètre d'entrée/sortie *iq_stSetAlarm* qui correspond au paramètre d'entrée/sortie *iq_astAlarms[#]*.

Cette fonction ne peut pas être utilisée simultanément dans plusieurs tâches.

En utilisant l'entrée *iq_stSetAlarm*, un message d'alarme spécifique à une machine peut être transmis à la fonction.

Via le paramètre d'entrée/sortie *iq_astAlarms*, la balise d'administration *Admin.Alarm[#]* doit être transmise à la fonction. Elle représente la liste des alarmes actives dans l'unité de la machine dans l'ordre chronologique en commençant par la première alarme survenue et toujours active.

Via le paramètre d'entrée/sortie *iq_stAlarmHistory*, la balise d'administration *Admin.AlarmHistory[#]* doit être transmise à la fonction. Elle représente l'historique des alarmes de l'unité de la machine contenant les alarmes dans l'ordre chronologique en commençant par la dernière alarme réinitialisée.

Si la fonction est appelée avec l'entrée *i_xSetAlarm = TRUE*, la fonction vérifie si le message d'alarme est déjà dans la liste d'alarmes actives. Si ce n'est pas le cas, la fonction obtient l'horodatage du contrôleur et l'écrit avec le message d'alarme dans la liste d'alarmes actives. Si le nombre maximal de messages dans la liste est atteint, aucun nouveau messages n'est ajouté.

Si la fonction est appelée avec *i_xSetAlarm = FALSE*, le message d'alarme est supprimé de la liste d'alarmes actives. Il est ensuite écrit en première position dans l'historique des alarmes et les entrées existantes sont déplacées d'une position vers le bas dans la liste. Si la liste est pleine, l'alarme réinitialisée la plus ancienne est supprimée.

Si la fonction est appelée avec les entrées *i_xAckAlarm* et *i_xSetAlarm = TRUE*, elle obtient l'horodatage du contrôleur et met à jour le paramètre (balise) *AckDateTime* du message d'alarme dans la liste des alarmes actives. Lorsqu'une alarme est acquittée, toute nouvelle requête d'acquiescement est sans effet.

La valeur de retour de la fonction indique TRUE si l'alarme a été écrite dans la liste, supprimée de la liste ou a été acquittée. Si la fonction renvoie la valeur FALSE, aucune action n'a été demandée ou bien le nombre maximal de messages dans la liste est atteint.

Interface

Entrée	Type de données	Description
<i>i_xSetAlarm</i>	BOOL	- TRUE : le code motif ou l'alarme spécifique à la machine transmis à l'entrée/sortie <i>iq_stSetAlarm</i> est écrit dans la balise de l'entrée/sortie <i>iq_stActiveAlarm</i>. - FALSE : le message est effacé de <i>iq_stActiveAlarms</i> et ajouté à <i>iq_stAlarmHistory</i>.
<i>i_xAlarmAck</i>	BOOL	TRUE : le message a été détecté. L'horodatage approprié à <i>Admin.Alarm[#].TimeAck</i> est déterminé.

Entrée/sortie	Type de données	Description
<i>iq_stSetAlarm</i>	<i>ST_InitAlarm</i>	Le code motif ou l'alarme spécifique à la machine est transmis à cette entrée/sortie.
<i>iq_astActiveAlarms</i>	ARRAY [1..Gc_uiMaxNumberOfAlarms] OF <i>ST_Alarm</i>	La balise d'administration <i>Admin.Alarm[#]</i> devrait être appliquée à cette entrée/sortie.
<i>iq_astAlarmHistory</i>	ARRAY [1..Gc_uiNumberOfAlarmHistory] OF <i>ST_Alarm</i>	La balise d'administration <i>Admin.AlarmHistory[#]</i> devrait être attribuée à cette entrée/sortie.
<i>iq_diNumberOfActiveAlarms</i>	DINT	Fournit le nombre de messages d'alarme actifs.

FC_SetWarning

Présentation

Type :	Fonction
Disponible à partir de :	V1.1.0.0



Description fonctionnelle

La fonction *FC_SetWarning* est une fonction auxiliaire qui permet d'écrire des messages de conseil dans les balises d'administration appropriées, ainsi que de les supprimer. Un conseil est identifié exclusivement par un identifiant unique et la valeur associée. L'identifiant et sa valeur sont définis par le paramètre d'entrée/sortie *iq_stSetWarning* qui correspond au paramètre d'entrée/sortie *iq_astWarnings[#]*.

Cette fonction ne peut pas être utilisée simultanément dans plusieurs tâches.

En utilisant l'entrée *iq_stSetWarning*, un message de conseil spécifique à une machine peut être transmis à la fonction.

Via le paramètre d'entrée/sortie *iq_astWarnings*, la balise d'administration *Admin.Warning[#]* doit être transmise à la fonction. Elle représente la liste des conseils actifs dans l'unité de la machine dans l'ordre chronologique en commençant par le premier conseil apparu et toujours actif.

Si la fonction est appelée avec l'entrée *i_xSetWarning* = *TRUE*, la fonction vérifie si le message de conseil est déjà dans la liste de conseils actifs. Si ce n'est pas le cas, la fonction obtient l'horodatage du contrôleur et l'écrit avec le message de conseil dans la liste de conseils actifs. Si le nombre maximal de messages dans la liste est atteint, aucun nouveau messages n'est ajouté.

Si la fonction est appelée avec *i_xSetWarning* = *FALSE*, le message de conseil est supprimé de la liste de conseils actifs.

Si la fonction est appelée avec les entrées *i_xAckWarning* et *i_xSetWarning* = *TRUE*, elle obtient l'horodatage du contrôleur et met à jour le paramètre (balise) *AckDateTime* du message de conseil dans la liste de conseils actifs. Lorsqu'un conseil est acquitté, toute nouvelle requête d'acquiescement est sans effet.

La valeur de retour de la fonction indique *TRUE* si le conseil a été écrit dans la liste, supprimé de la liste ou a été acquitté. Si la fonction renvoie la valeur *FALSE*, aucune action n'a été demandée ou bien le nombre maximal de messages dans la liste est atteint.

Interface

Entrée	Type de données	Description
<i>i_xSetWarning</i>	BOOL	Requête d'écriture (<i>TRUE</i>) ou de suppression (<i>FALSE</i>) du message dans/de la liste liée à <i>iq_astWarnings</i> .
<i>i_xWarningAck</i>	BOOL	Requête d'acquiescement d'un conseil actif.

Entrée/sortie	Type de données	Description
<i>iq_stSetWarning</i>	<i>ST_InitAlarm</i>	Indique le message de conseil à traiter par la fonction.
<i>iq_astWarnings</i>	ARRAY [1..Gc_uiMaxNumberOfWarnings] OF <i>ST_Alarm</i>	La balise d'administration <i>Admin.Warning[#]</i> doit être liée à cette entrée/sortie.
<i>iq_diNumberOfActiveWarnings</i>	DINT	Fournit le nombre de conseils de la liste.

Visualisation

Contenu de cette partie

BackgroundFrames 116

States Frames 118

Cadres de visualisation..... 119

BackgroundFrames

Contenu de ce chapitre

FR_<BackgroundFrame> 116

FR_<BackgroundFrame>

Présentation

Type :	Cadre de visualisation
Disponible à partir de :	V1.0.1.0
Implémenté :	VisuElems.IVisualization

Tâche

Afficher un cadre en arrière-plan.

Description fonctionnelle

Les cadres de visualisation *FR_<BackgroundFrame>* (par exemple, *FR_Control1024x768*) sont des cadres d'arrière-plan qui permettent de regrouper les autres cadres et de réaliser des fonctionnalités de contrôle et de navigation. Voir les exemples ci-après.

Les cadres d'arrière-plan suivants sont disponibles :

Cadre	Description
<i>FR_Control1024x768</i>	Cadre de contrôle, taille 1024x768 pixels
<i>FR_Nav1024x768</i>	Cadre de navigation, taille 1024x768 pixels
<i>FR_NavControl1024x768</i>	Cadre de navigation et de contrôle, taille 1024x768 pixels
<i>FR_Control800x600</i>	Cadre de contrôle, taille 800x600 pixels
<i>FR_Nav800x600</i>	Cadre de navigation, taille 800x600 pixels
<i>FR_NavControl800x600</i>	Cadre de navigation et de contrôle, taille 800x600 pixels

Interface

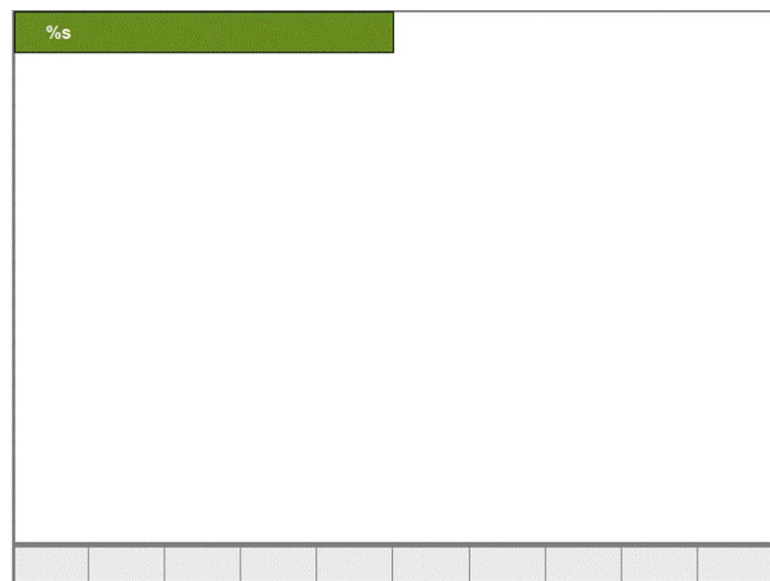
Entrée	Type de données	Description
<i>i_sTitle</i>	STRING(80)	Titre du cadre d'arrière-plan

Exemples

FR_Control1024x768 avec contrôles sur le côté gauche



FR_Nav1024x768 avec boutons de navigation en bas



States Frames

Contenu de ce chapitre

FR_<état> 118

FR_<état>

Présentation

Type :	Cadre de visualisation
Disponible à partir de :	V1.0.1.0
Implémenté :	VisuElems.IVisualization

Tâche

Afficher les différents états du modèle d'état du mode de fonctionnement.

Description fonctionnelle

Pour chaque état spécifié dans le modèle d'état (voir *ET_States*), une visualisation est fournie.

Les cadres sont utilisés par *FR_DynStateModel* pour afficher les différents états du modèle d'état. Voir l'exemple suivant.

Interface

Entrée/sortie	Type de données	Description
<i>iq_stVisInterface</i>	<i>ST_VisInterface</i>	Interface vers <i>FB_VisController</i>

Exemple



Cadres de visualisation

Contenu de ce chapitre

Généralités	119
<i>FR_Alarm</i>	120
<i>FR_AlarmHistory</i>	120
<i>FR_CurrentModeAndStateTime</i>	121
<i>FR_DynStateModel</i>	122
<i>FR_DynStateModel_2022</i>	123
<i>FR_ModeAndStateTime</i>	124
<i>FR_ProdConsumedCount</i>	125
<i>FR_ProdDefectiveCount</i>	126
<i>FR_ProdProcessedCount</i>	127
<i>FR_Warning</i>	128
<i>FR_StopReason</i>	128
<i>FR_StatesDisabled</i>	129

Généralités

Présentation

La bibliothèque fournit plusieurs cadres de visualisation.

Vous pouvez utiliser ces cadres pour afficher des informations portant sur :

- Les alarmes
- Les modes de fonctionnement et les états d'une machine
- Les données de production

Bloc Fonction *FB_VisController*

La tâche principale du bloc fonction *FB_VisController* est de traiter les données fournies par les balises d'administration et les balises d'état, et de transmettre ces données aux cadres de visualisation, via l'interface de visualisation entrées/sorties *iq_stVisInterface* de type *ST_VisInterface*.

Cadres de visualisation

- *FR_Alarm*
- *FR_AlarmHistory*
- *FR_CurrentModeAndStateTime*
- *FR_DynStateModel*
- *FR_ModeAndStateTime*
- *FR_ProdConsumedCount*
- *FR_ProdDefectiveCount*
- *FR_ProdProcessedCount*
- *FR_Warning*
- *FR_StopReason*
- *FR_StatesDisabled*

FR_Alarm

Présentation

Type :	Cadre de visualisation
Disponible à partir de :	V1.0.1.0
Implémenté :	VisuElems.IVisualization

Tâche

Afficher une liste d'alarmes des alarmes actives.

Description fonctionnelle

FR_Alarm est un cadre de visualisation qui sert à afficher la liste d'alarmes des alarmes actives (*Admin.Alarm[#]*). Voyez l'exemple ci-après.

Interface

Entrée / sortie	Type de données	Description
<i>iq_stVisInterface</i>	<i>ST_VisInterface</i>	Interface vers le <i>FB_VisController</i>

Exemple

#	ID	Value	Cat.	Message	Trigger	DateTime	AckDateTime
1	39	2	2	This alarm is active.	TRUE	2017 - 9 - 14 - 14:15:59.619000	2017 - 9 - 14 - 14:15:59.619000
2	1234	1	1	This alarm is inactive but not acknowledged.	FALSE	2017 - 9 - 14 - 14:38:23.2385	0 - 0 - 0 - 0:0:0.0
3	456	63	4	This alarm is inactive and acknowledged.	FALSE	2017 - 9 - 14 - 14:52:26.56987	2017 - 9 - 14 - 14:53:11.369
4	0	0	0		FALSE	0 - 0 - 0 - 0:0:0.0	0 - 0 - 0 - 0:0:0.0
5	0	0	0		FALSE	0 - 0 - 0 - 0:0:0.0	0 - 0 - 0 - 0:0:0.0
6	0	0	0		FALSE	0 - 0 - 0 - 0:0:0.0	0 - 0 - 0 - 0:0:0.0
7	0	0	0		FALSE	0 - 0 - 0 - 0:0:0.0	0 - 0 - 0 - 0:0:0.0
8	0	0	0		FALSE	0 - 0 - 0 - 0:0:0.0	0 - 0 - 0 - 0:0:0.0
9	0	0	0		FALSE	0 - 0 - 0 - 0:0:0.0	0 - 0 - 0 - 0:0:0.0
▼	0	0	0		FALSE	0 - 0 - 0 - 0:0:0.0	0 - 0 - 0 - 0:0:0.0

FR_AlarmHistory

Présentation

Type :	Cadre de visualisation
Disponible à partir de :	V1.0.1.0
Implémenté :	VisuElems.IVisualization

Tâche

Afficher l'historique des alarmes.

Description fonctionnelle

FR_AlarmHistory est un cadre de visualisation qui sert à afficher l'historique des alarmes (*Admin.AlarmHistory[#]*). Voyez l'exemple ci-après.

Interface

Entrée / sortie	Type de données	Description
<i>iq_stVisInterface</i>	<i>ST_VisInterface</i>	Interface vers le <i>FB_VisController</i>

Exemple

#	ID	Value	Cat.	Message	DateTime	AckDateTime
1	38	1	19	This alarm is not yet acknowledged.	2017 - 9 - 14 - 14:15:57.549000	0 - 0 - 0 - 0 : 0 : 0.0
2	37	1	65	This alarm is already acknowledged.	2017 - 9 - 14 - 14:14:30.540000	2017 - 9 - 14 - 14:14:41.233
3	0	0	0		0 - 0 - 0 - 0 : 0 : 0.0	0 - 0 - 0 - 0 : 0 : 0.0
4	0	0	0		0 - 0 - 0 - 0 : 0 : 0.0	0 - 0 - 0 - 0 : 0 : 0.0
5	0	0	0		0 - 0 - 0 - 0 : 0 : 0.0	0 - 0 - 0 - 0 : 0 : 0.0
6	0	0	0		0 - 0 - 0 - 0 : 0 : 0.0	0 - 0 - 0 - 0 : 0 : 0.0
7	0	0	0		0 - 0 - 0 - 0 : 0 : 0.0	0 - 0 - 0 - 0 : 0 : 0.0
8	0	0	0		0 - 0 - 0 - 0 : 0 : 0.0	0 - 0 - 0 - 0 : 0 : 0.0
9	0	0	0		0 - 0 - 0 - 0 : 0 : 0.0	0 - 0 - 0 - 0 : 0 : 0.0
v	0	0	0		0 - 0 - 0 - 0 : 0 : 0.0	0 - 0 - 0 - 0 : 0 : 0.0

FR_CurrentModeAndStateTime

Présentation

Type :	Cadre de visualisation
Disponible à partir de :	V1.0.1.0
Implémenté :	VisuElems.IVisualization

Tâche

Afficher la durée de fonctionnement de la machine dans le mode de fonctionnement et l'état actuels.

Description fonctionnelle

FR_CurrentModeAndStateTime est un cadre de visualisation qui sert à afficher la durée de fonctionnement de la machine dans le mode de fonctionnement (*Admin.ModeCurrentTime[#]* & *Admin.ModeCumulativeTime[#]*) et l'état (*Admin.StateCurrentTime[#,#]* & *Admin.StateCumulativeTime[#,#]*) actuels. Voyez l'exemple ci-après.

Interface

Entrée / sortie	Type de données	Description
<i>iq_stVisInterface</i>	<i>ST_VisInterface</i>	Interface vers le <i>FB_VisController</i>

Exemple

Current Unit Mode:	1
Time in Current Mode:	58
Cummulative Time in Current Mode:	4885
Current State:	6
Time in Current State:	40
Cummulative Time in Current State:	40

NOTE: Cet exemple ne montre que les informations de PackTag, qui, selon les normes en vigueur, sont exprimés en secondes.

FR_DynStateModel

Présentation

Type :	Cadre de visualisation
Disponible à partir de :	V1.0.1.0
Implémenté :	VisuElems.IVisualization

Tâche

Afficher dynamiquement la visualisation du modèle d'état pour le mode de fonctionnement pendant le fonctionnement en ligne.

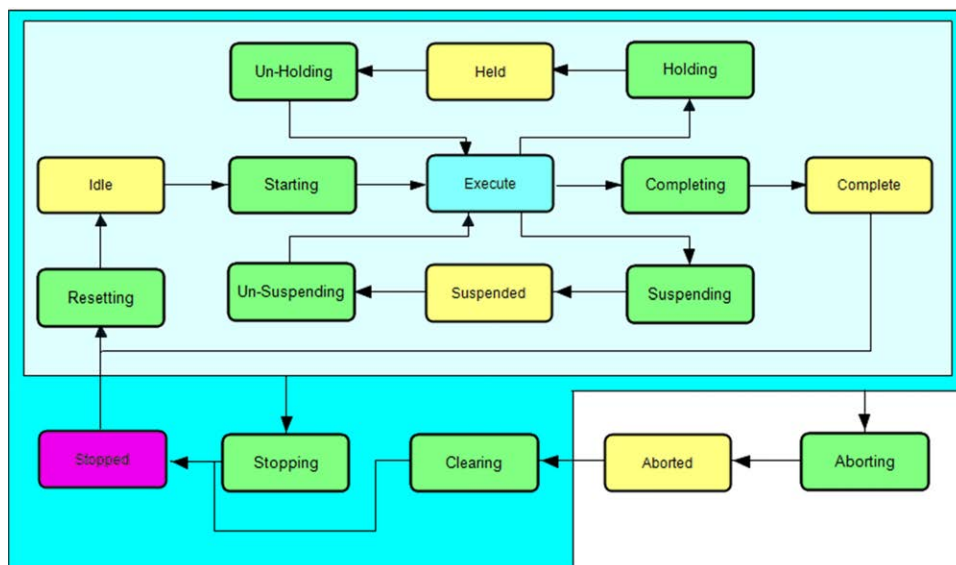
Description fonctionnelle

FR_DynStateModel fournit un cadre de visualisation qui sert à afficher les modèles d'état. Ce cadre génère dynamiquement la visualisation du modèle d'état correspondant pendant le fonctionnement en ligne.

Interface

Entrée / sortie	Type de données	Description
<i>iq_stVisInterface</i>	<i>ST_VisInterface</i>	Interface vers le <i>FB_VisController</i>

Exemple



FR_DynStateModel_2022

Présentation

Type :	Cadre de visualisation
Disponible à partir de :	V1.4.2.0

Tâche

Afficher dynamiquement la visualisation du modèle d'état pour le mode de l'unité actuelle.

Description fonctionnelle

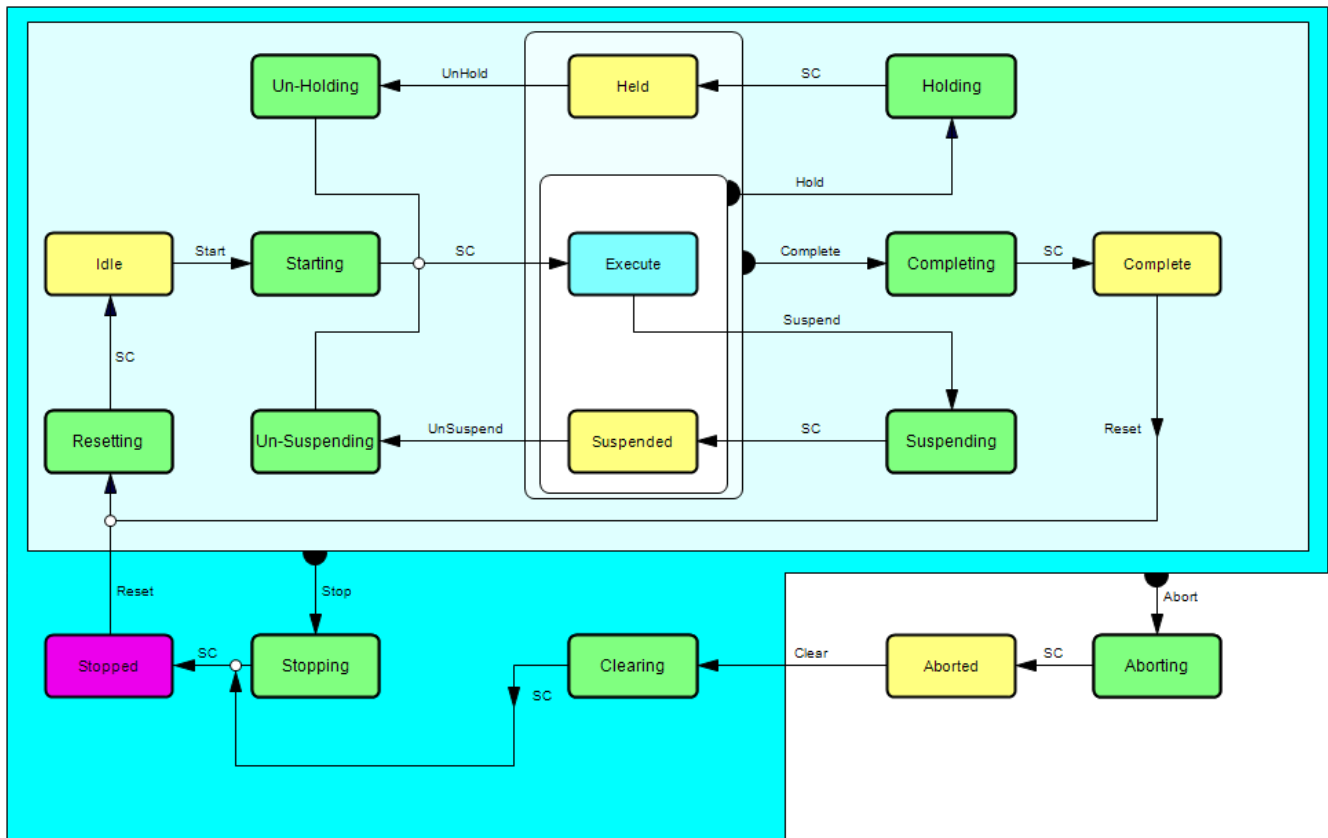
FR_DynStateModel_2022 affiche dynamiquement la visualisation du modèle d'état pendant le fonctionnement en ligne. Le modèle d'état est conforme au modèle d'état de base PackML tel que défini dans ANSI/ISA TR88.00.02-2022.

Interface

Entrée/sortie	Type de données	Description
<i>i_stVisInterface</i>	<i>ST_VisInterface</i>	Interface vers <i>FB_VisController</i>

Example

Le modèle d'état affiché est adapté dynamiquement en fonction du modèle d'état défini.



FR_ModeAndStateTime

Présentation

Type :	Cadre de visualisation
Disponible à partir de :	V1.0.1.0
Implémenté :	VisuElems.IVisualization

Tâche

Afficher la durée de fonctionnement de tous les modes de fonctionnement et leurs états respectifs.

Description fonctionnelle

FR_ModeAndStateTime est un cadre de visualisation qui sert à afficher la durée de fonctionnement de tous les modes de fonctionnement (*Admin.ModeCurrentTime[#]* & *Admin.ModeCumulativeTime[#]*) et leurs états respectifs (*Admin.StateCurrentTime[#,#]* & *Admin.StateCumulativeTime[#,#]*).

Interface

Entrée / sortie	Type de données	Description
<i>iq_stVisInterface</i>	<i>ST_VisInterface</i>	Interface vers le <i>FB_VisController</i>

Exemple

Unit Mode [0]			>
Current Time	0		
Cumulative Time	18		
States			
Current Time		Cumulative Time	
Undefined	0	Undefined	18
Execute	0	Execute	0
Stopping	0	Stopping	0
Stopped	0	Stopped	0
Resetting	0	Resetting	0
Idle	0	Idle	0
Starting	0	Starting	0
Holding	0	Holding	0
Held	0	Held	0
Unholding	0	Unholding	0
Suspending	0	Suspending	0
Suspended	0	Suspended	0
Unsuspending	0	Unsuspending	0
Completing	0	Completing	0
Complete	0	Complete	0
Aborting	0	Aborting	0
Aborted	0	Aborted	0
Clearing	0	Clearing	0

FR_ProdConsumedCount

Présentation

Type :	Cadre de visualisation
Disponible à partir de :	V1.0.1.0
Implémenté :	VisuElems.IVisualization

Tâche

Afficher les chiffres de comptage produits des consommables.

Description fonctionnelle

FR_ProdConsumedCount est un cadre de visualisation qui sert à afficher les chiffres de comptage produits des consommables (*Admin.ProdConsumedCount* [#]).

Interface

Entrée / sortie	Type de données	Description
<i>iq_stVisInterface</i>	<i>ST_VisInterface</i>	Interface vers le <i>FB_VisController</i>

Exemple

Product Consumed Counter [1] >	
ID	0
Name	
Unit	
Count	0
AccCount	0

FR_ProdDefectiveCount

Présentation

Type :	Cadre de visualisation
Disponible à partir de :	V1.0.1.0
Implémenté :	VisuElems.IVisualization

Tâche

Afficher les chiffres de comptage produits des marchandises détectées comme non conformes.

Description fonctionnelle

FR_ProdDefectiveCount est un cadre de visualisation servant à afficher les chiffres de comptage produits des marchandises détectées comme non conformes (*Admin.ProdDefectiveCount*[#]).

Interface

Entrée / sortie	Type de données	Description
<i>iq_stVisInterface</i>	<i>ST_VisInterface</i>	Interface vers le <i>FB_VisController</i>

Exemple

Product Defective Counter [1] >	
ID	0
Name	
Unit	
Count	0
AccCount	0

FR_ProdProcessedCount

Présentation

Type :	Cadre de visualisation
Disponible à partir de :	V1.0.1.0
Implémenté :	VisuElems.IVisualization

Tâche

Afficher les chiffres de comptage produits des marchandises produites.

Description fonctionnelle

FR_ProdProcessedCount est un cadre de visualisation qui sert à afficher les chiffres de comptage produits des marchandises produites (*Admin. ProdProcessedCount[#]*).

Interface

Entrée / sortie	Type de données	Description
<i>iq_stVisInterface</i>	<i>ST_VisInterface</i>	Interface vers le <i>FB_VisController</i>

Exemple

Product Processed Counter [1] >	
ID	0
Name	
Unit	
Count	0
AccCount	0

FR_Warning

Présentation

Type :	Cadre de visualisation
Disponible à partir de :	V1.1.0.0
Implémenté :	VisuElems.IVisualization

Tâche

Afficher la liste de conseils actifs.

Description fonctionnelle

FR_Warning est un cadre de visualisation qui permet d'afficher la liste de conseils actifs (*Admin.Warning[#]*). Voyez l'exemple ci-après.

Interface

Entrée / sortie	Type de données	Description
<i>iq_stVisInterface</i>	<i>ST_VisInterface</i>	Interface vers le <i>FB_VisController</i>

Exemple

#	ID	Value	Cat.	Message	Trigger	DateTime	AckDateTime
1	123	1	1	This Warning is active.	TRUE	2017 - 9 - 14 - 14:58:00.0	0 - 0 - 0 - 0:0:0.0
2	456	2	2	This warning is inactive but not acknowledged.	FALSE	2017 - 14 - 9 - 14:58:11.123	0 - 0 - 0 - 0:0:0.0
3	789	3	3	This warning is inactive and acknowledged.	FALSE	2017 - 9 - 14 - 14:58:32.1258	2017 - 9 - 14 - 14:59:11.0
4	0	0	0		FALSE	0 - 0 - 0 - 0:0:0.0	0 - 0 - 0 - 0:0:0.0
5	0	0	0		FALSE	0 - 0 - 0 - 0:0:0.0	0 - 0 - 0 - 0:0:0.0
5	0	0	0		FALSE	0 - 0 - 0 - 0:0:0.0	0 - 0 - 0 - 0:0:0.0
7	0	0	0		FALSE	0 - 0 - 0 - 0:0:0.0	0 - 0 - 0 - 0:0:0.0
8	0	0	0		FALSE	0 - 0 - 0 - 0:0:0.0	0 - 0 - 0 - 0:0:0.0
9	0	0	0		FALSE	0 - 0 - 0 - 0:0:0.0	0 - 0 - 0 - 0:0:0.0
▼	0	0	0		FALSE	0 - 0 - 0 - 0:0:0.0	0 - 0 - 0 - 0:0:0.0

FR_StopReason

Présentation

Type :	Cadre de visualisation
Disponible à partir de :	V1.1.0.0
Implémenté :	VisuElems.IVisualization

Tâche

Afficher le PackTag *Admin.StopReason*.

Description fonctionnelle

FR_StopReason est un cadre de visualisation qui permet d'afficher le PackTag *Admin.StopReason*.

Interface

Entrée / sortie	Type de données	Description
<i>iq_stVisInterface</i>	<i>ST_VisInterface</i>	Interface vers le <i>FB_VisController</i>

Exemple

ID	Value	Cat.	Message	DateTime
10	8	21	Abort command detetcted.	2017 - 10 - 23 - 13:15:26.532000

FR_StatesDisabled

Présentation

Type :	Cadre de visualisation
Disponible à partir de :	V1.1.0.0
Implémenté :	VisuElems.IVisualization

Tâche

Surveiller et contrôler le PackTag *Admin.StatesDisabled*.

Description fonctionnelle

FR_StatesDisabled est un cadre de visualisation qui permet de surveiller et de contrôler le PackTag *Admin.StatesDisabled*.

Interface

Entrée/sortie	Type de données	Description
<i>iq_stVisInterface</i>	<i>ST_VisInterface</i>	Interface vers <i>FB_VisController</i>

Exemple

StatesDisabled	
☐	Clearing
☐	Stopped
☐	Starting
☐	Idle
☐	Suspended
☐	Execute
☐	Stopping
☐	Aborting
☐	Aborted
☐	Holding
☐	Held
☐	UnHolding
☐	Suspending
☐	UnSuspending
☐	Resetting
☐	Completing
☐	Complete

Glossaire

P

POU:

Acronyme de *program organization unit*, unité organisationnelle de programme. Déclaration de variables dans le code source et jeu d'instructions correspondant. Les POU facilitent la réutilisation modulaire de programmes logiciels, de fonctions et de blocs fonction. Une fois déclarées, les POU sont réutilisables.

R

RTC:

Acronyme de *real-time clock*, horloge en temps réel. Horloge horaire et calendrier supportée par une batterie qui fonctionne en continu, même lorsque le contrôleur n'est pas alimenté, jusqu'à la fin de l'autonomie de la batterie.

Index

A		
Aborted	97	
Aborting.....	97	
B		
BackgroundFrames	116	
C		
Cadres de visualisation.....	119	
Clearing.....	93	
CmdAbort	72	
CmdChangeUnitMode	90	
CmdClear	73	
CmdHold	69	
CmdReset	67	
CmdStart	68	
CmdStop	69	
CmdSuspend.....	71	
CmdUnHold.....	70	
CmdUnSuspend.....	71	
Commande.....	92	
Complete.....	101	
Completing	101	
Concept de diagnostic	26	
D		
DateTimeArray.....	59	
DefineUnitMode	84	
DefineUnitModeWithHandler.....	86	
E		
entrées et sorties communes		
comportement des blocs fonction avec entrée <i>i_</i>		
<i>xEnable</i>	28	
ET_Cmd	30	
abort	105	
Abort	30	
clear	105	
Clear	30	
hold	105	
Hold	30	
indéfini	104	
reset	104	
Reset	30	
start	104	
Start.....	30	
stop	105	
Stop.....	30	
suspend	105	
Suspend.....	30	
Undefined.....	30	
UnHold.....	30, 105	
UnSuspend	30, 105	
ET_Modes	31	
Maintenance.....	31	
Manual.....	31	
Producing.....	31	
Undefined.....	31	
ET_Result	31	
AlreadyExists.....	33	
Disabled	32	
Initializing	32	
InvalidInput	32	
InvalidStateModelNoAbortedState	32	
InvalidStateModelNoCompleteState	32	
InvalidStateModelNoExecuteState	32	
InvalidStateModelNoHeldState	32	
InvalidStateModelNoIdleState	32	
InvalidStateModelNoResettingState	32	
InvalidStateModelNoStoppedState	32	
InvalidStateModelNoSuspendedState	32	
MaxNumberOfUnitModesExceeded	33	
ModeChangeRequestRejected	32	
NoStateModelHandler	33	
NotReady	33	
NoUnitMode	33	
NumberOfModesRange	32	
OK	32	
PointerAdminInvalid	32	
PointerInitModelInvalid	32	
PointerStatusInvalid	32	
RegisterLoggerPointFailed	33	
ResettingTimerAndCounter	32	
Running	32	
StateChangeRequestRejected	32	
StateCurrentRange	32	
UnitModeActive	33	
UnitModeCurrentRange	32	
UnitModeNotFound	33	
UnitModeRange.....	32	
UnknownResult	32	
ET_StateModelDefinition	33	
Aborted	34	
AbortedAndStopped.....	34	
Aborting	34	
Clearing	34	
Complete	34	
CompleteDisabled	34	
Completing	34	
Execute.....	34	
Held	34	
HoldDisabled	34	
Holding	34	
Idle	34	
NoStateDisabled.....	34	
Resetting.....	34	
Starting	34	
Stopped	34	
Stopping.....	34	
SuspendDisabled.....	34	
Suspended	34	
Suspending	34	
UnHolding	34	
Unsuspending	34	
ET_States	34	
Aborted	35	
Aborting	35	
Clearing	35	
Complete	35	
Completing	35	
Execute.....	35	
Held	35	
Holding	35	
Idle	35	
Non défini	35	
Resetting.....	35	
Starting	35	
Stopped	35	

Stopping.....	35
Suspended.....	35
Suspending.....	35
UnHolding.....	35
UnSuspending.....	35
Execute.....	96
ExecuteCurrentState.....	89

F

FB_DataManagement.....	77
FB_ModeManager.....	79
FB_StateModelHandlerBase.....	93
Aborted.....	97
Aborting.....	97
Clearing.....	93
Complete.....	101
Completing.....	101
Execute.....	96
Held.....	98
Holding.....	98
Idle.....	95
Resetting.....	100
Restarting.....	99
Starting.....	94
Stopped.....	94
Stopping.....	96
Suspended.....	95
Suspending.....	99
Unsuspending.....	100
FB_UnitModeManager2.....	82
CmdChangeUnitMode.....	90
Commande.....	92
DefineUnitMode.....	84
DefineUnitModeWithHandler.....	86
ExecuteCurrentState.....	89
GetDefinedUnitModes.....	91
propriétés.....	84
RegisterLoggerPoint.....	87
RemoveUnitMode.....	91
SetApplicationLoggerLogLevel.....	88
FB_VisController.....	102
FC_CheckCmd.....	104
FC_CheckCmd2.....	105
FC_EtResultToString.....	106
FC_GetDateTimeAsArray.....	107
FC_InitStateModelChangeStates.....	107
FC_InitStateModelExistingStates.....	110
FC_SetAlarm.....	112
FC_SetWarning.....	113
FR_<BackgroundFrame>.....	116
FR_<état>.....	118
FR_Aborted.....	118
FR_Aborting.....	118
FR_Alarm.....	120
FR_AlarmHistory.....	120
FR_Clearing.....	118
FR_Complete.....	118
FR_Completing.....	118
FR_Control1024x768.....	116
FR_Control800x600.....	116
FR_CurrentModeAndStateTime.....	121
FR_DynStateModel.....	122
FR_DynStateModel_2022.....	123
FR_Execute.....	118
FR_Held.....	118
FR_Holding.....	118
FR_Idle.....	118
FR_ModeAndStateTime.....	124

FR_Nav1024x768.....	116
FR_Nav800x600.....	116
FR_NavControl1024x768.....	116
FR_NavControl800x600.....	116
FR_ProdConsumedCount.....	125
FR_ProdDefectiveCount.....	126
FR_ProdProcessedCount.....	127
FR_Resetting.....	118
FR_Starting.....	118
FR_StatesDisabled.....	129
FR_Stopped.....	118
FR_Stopping.....	118
FR_StopReason.....	128
FR_Suspended.....	118
FR_Suspending.....	118
FR_Undefined.....	118
FR_UnHolding.....	118
FR_UnSuspending.....	118
FR_Warning.....	128

G

GCL.....	61
PackML.....	61
GetDefinedUnitModes.....	91
GPL.....	62
PackML.....	62

H

Held.....	98
-----------	----

I

Idle.....	95
IF_StateCommands.....	67
CmdAbort.....	72
CmdClear.....	73
CmdHold.....	69
CmdReset.....	67
CmdStart.....	68
CmdStop.....	69
CmdSuspend.....	71
CmdUnHold.....	70
CmdUnSuspend.....	71
StateComplete.....	73
IF_StateModelHandler.....	75
IF_UnitMode.....	66

M

modèle d'état de base PackML.....	83
-----------------------------------	----

P

PackML.....	
GCL.....	61
GPL.....	62
PackML, modèle d'état de base.....	83
propriétés.....	
FB_UnitModeManager2.....	84

Q

q_etResult.....	26
q_sResultMsg.....	26

q_xError	26
----------------	----

R

RegisterLoggerPoint.....	87
RemoveUnitMode	91
Resetting	100
Restarting	99
Rétention	98

S

SetApplicationLoggerLogLevel	88
ST_Administration	36
ST_Alarm	45
Starting	94
StateComplete	73
States frames	118
ST_Command	38
ST_CountDescrip	46
ST_Cumulative_Times	47
ST_Descriptor	47
ST_EquipmentInterlock	48
ST_Ingredient	49
ST_Ingredients	49
ST_InitAlarm	40
ST_Interface	50
ST_ModeState_Times	50
Stopped	94
Stopping	96
ST_Parameter_DINT	51
ST_Parameter_LREAL	52
ST_Parameter_REAL	52
ST_Parameter_STRING	53
ST_ProcessVariables	54
ST_Product	54
ST_Product_Data	55
ST_Recipe	56
ST_StateInfo	57
ST_Status	39
ST_Timestamp	57
ST_UnitModeDefinition	41
ST_VisInterface	42
Suspended	95
Suspending	99

U

Unsuspending	100
--------------------	-----

Schneider Electric
35 rue Joseph Monier
92500 Rueil Malmaison
France

+ 33 (0) 1 41 29 70 00

www.se.com

Les normes, spécifications et conceptions pouvant changer de temps à autre, veuillez demander la confirmation des informations figurant dans cette publication.

© 2023 Schneider Electric. Tous droits réservés.

EIO0000002810.01