

EcoStruxure Machine Expert

UserMotorTypePlate

Guide de la bibliothèque

26.02.2019

EIO0000003911.01

www.schneider-electric.com

Schneider
 **Electric**

Le présent document comprend des descriptions générales et/ou des caractéristiques techniques des produits mentionnés. Il ne peut pas être utilisé pour définir ou déterminer l'adéquation ou la fiabilité de ces produits pour des applications utilisateur spécifiques. Il incombe à chaque utilisateur ou intégrateur de réaliser l'analyse de risques complète et appropriée, l'évaluation et le test des produits pour ce qui est de l'application à utiliser et de l'exécution de cette application. Ni la société Schneider Electric ni aucune de ses sociétés affiliées ou filiales ne peuvent être tenues pour responsables de la mauvaise utilisation des informations contenues dans le présent document. Si vous avez des suggestions, des améliorations ou des corrections à apporter à cette publication, veuillez nous en informer.

Vous acceptez de ne pas reproduire, excepté pour votre propre usage à titre non commercial, tout ou partie de ce document et sur quelque support que ce soit sans l'accord écrit de Schneider Electric. Vous acceptez également de ne pas créer de liens hypertextes vers ce document ou son contenu. Schneider Electric ne concède aucun droit ni licence pour l'utilisation personnelle et non commerciale du document ou de son contenu, sinon une licence non exclusive pour une consultation « en l'état », à vos propres risques. Tous les autres droits sont réservés.

Toutes les réglementations locales, régionales et nationales pertinentes doivent être respectées lors de l'installation et de l'utilisation de ce produit. Pour des raisons de sécurité et afin de garantir la conformité aux données système documentées, seul le fabricant est habilité à effectuer des réparations sur les composants.

Lorsque des équipements sont utilisés pour des applications présentant des exigences techniques de sécurité, suivez les instructions appropriées.

La non-utilisation du logiciel Schneider Electric ou d'un logiciel approuvé avec nos produits matériels peut entraîner des blessures, des dommages ou un fonctionnement incorrect.

Le non-respect de cette consigne peut entraîner des lésions corporelles ou des dommages matériels.

© 2019 Schneider Electric. Tous droits réservés.

Table des matières



	Consignes de sécurité	5
	A propos de ce manuel.....	11
Chapitre 1	Présentation de la bibliothèque	19
	Informations générales	20
	Concept de diagnostic.....	25
Chapitre 2	Énumérations	29
2.1	ET_DiagExt.....	30
	ET_DiagExt - Informations générales.....	30
2.2	ET_EncoderType	33
	ET_EncoderType - Informations générales.....	33
2.3	ET_MotorType.....	34
	ET_MotorType - Informations générales.....	34
2.4	ET_StorageLocation	35
	ET_StorageLocation - Informations générales	35
Chapitre 3	Blocs fonction	37
3.1	FB_InitMachineEncoder.....	38
	FB_InitMachineEncoder - Informations générales.....	38
3.2	FB_MotorDataDelete	49
	FB_MotorDataDelete - Informations générales	49
3.3	FB_MotorDataRead	55
	FB_MotorDataRead - Informations générales	55
3.4	FB_MotorDataWrite	60
	FB_MotorDataWrite - Informations générales	60
3.5	FB_MotorDataWriteBLH.....	68
	FB_MotorDataWriteBLH - Informations générales	68
3.6	FB_MotorSerialNumberWrite.....	76
	FB_MotorSerialNumberWrite - Informations générales.....	76
Chapitre 4	Fonctions	81
4.1	FC_EtDiagExtToString	82
	FC_EtDiagExtToString - Informations générales	82
4.2	FC_MotorDataFileCreate.....	83
	FC_MotorDataFileCreate - Informations générales.....	83
4.3	FC_MotorDataFileRead.....	95
	FC_MotorDataFileRead - Informations générales	95

Chapitre 5 Structures	99
5.1 ST_UserMachineEncoderData	100
ST_UserMachineEncoderData - Informations générales	100
5.2 ST_UserMotorData	102
ST_UserMotorData - Informations générales	102
5.3 ST_UserMotorDataACIM	103
ST_UserMotorDataACIM - Informations générales	103
5.4 ST_UserMotorDataPMSM	108
ST_UserMotorDataPMSM - Informations générales	108
Chapitre 6 Éléments globaux	115
6.1 GCL (Global Constant List)	116
GCL (Global Constant List) - Informations générales	116
Glossaire	119
Index	121

Consignes de sécurité



Informations importantes

AVIS

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 disposant de compétences et de connaissances dans le domaine de la construction et du fonctionnement des équipements électriques et installations et ayant bénéficié d'une formation de sécurité afin de reconnaître et d'éviter les risques encourus.

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 régulations 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.

DEMARRAGE 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 REGLAGES

Les précautions suivantes sont extraites du document NEMA Standards Publication ICS 7.1-1995 (la version anglaise prévaut) :

- 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.
- Seuls ces réglages fonctionnels, requis par l'opérateur, doivent lui être accessibles. L'accès aux autres commandes doit être limité afin d'empêcher les changements non autorisés des caractéristiques de fonctionnement.

A propos de ce manuel



Présentation

Objectif du document

Ce document décrit les fonctionnalités de la bibliothèque UserMotorTypePlate.

Champ d'application

Ce document a été actualisé suite au lancement d'EcoStruxure™ Machine Expert V1.1.

Les caractéristiques techniques des équipements décrits dans ce document sont également fournies en ligne. Pour accéder à ces informations en ligne :

Etape	Action
1	Accédez à la page d'accueil de Schneider Electric www.schneider-electric.com .
2	Dans la zone Search , saisissez la référence d'un produit ou le nom d'une gamme de produits. <ul style="list-style-type: none">● N'insérez pas d'espaces dans la référence ou la gamme de produits.● Pour obtenir des informations sur un ensemble de modules similaires, utilisez des astérisques (*).
3	Si vous avez saisi une référence, accédez aux résultats de recherche Product Datasheets et cliquez sur la référence qui vous intéresse. Si vous avez saisi une gamme de produits, accédez aux résultats de recherche Product Ranges et cliquez sur la gamme de produits qui vous intéresse.
4	Si plusieurs références s'affichent dans les résultats de recherche Products , cliquez sur la référence qui vous intéresse.
5	Selon la taille de l'écran, vous serez peut-être amené à faire défiler la page pour consulter la fiche technique.
6	Pour enregistrer ou imprimer une fiche technique au format .pdf, cliquez sur Download XXX product datasheet .

AVERTISSEMENT

PERTE DE CONTROLE

- Le concepteur d'un système de commande doit envisager les modes de défaillance possibles des chemins de commande et, pour certaines fonctions de commande critiques, prévoir un moyen d'atteindre un état sécurisé en cas de défaillance d'un chemin, et après cette défaillance. Par exemple, l'arrêt d'urgence, l'arrêt en cas de surcourse, la coupure de courant et le redémarrage sont des fonctions de contrôle cruciales.
- Des canaux de commande séparés ou redondants doivent être prévus pour les fonctions de commande critique.
- Les chemins de commande système peuvent inclure les liaisons de communication. Soyez particulièrement attentif aux implications des retards de transmission imprévus ou des pannes de liaison.
- Respectez toutes les réglementations de prévention des accidents ainsi que les consignes de sécurité locales.¹
- Chaque implémentation de cet équipement doit être testée individuellement et entièrement pour s'assurer du fonctionnement correct avant la mise 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, installation et opération de variateurs de vitesse) ou son 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.
- 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 commande critiques (arrêt d'urgence, conditions de dépassement des valeurs limites, etc.) en fonction d'une analyse de la sécurité, des règles correspondantes et des réglementations.

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

AVERTISSEMENT

FONCTIONNEMENT IMPRÉVU DE L'ÉQUIPEMENT

Évaluez toujours les valeurs de retour lors de l'utilisation des POU d'une bibliothèque.

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

AVERTISSEMENT

FONCTIONNEMENT IMPRÉVU DE L'ÉQUIPEMENT

- N'utiliser 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.

AVERTISSEMENT

FONCTIONNEMENT INATTENDU DE L'EQUIPEMENT

Actualisez votre programme d'application, si nécessaire, en portant une attention particulière aux réglages de l'adresse d'E/S, dès que vous modifiez la configuration matérielle.

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 de 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 IMPRÉVU DE L'ÉQUIPEMENT, PERTE DE DONNÉES OU FICHIER ENDOMMAGÉ

- 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.

AVERTISSEMENT

MOUVEMENT IMPRÉVU DE L'AXE

- Vérifiez le fonctionnement de l'équipement de sécurité fonctionnelle avant la mise en service.
- Vérifiez que vous pouvez arrêter les mouvements de l'axe à tout moment en utilisant l'équipement de sécurité (détecteur de fin de course, arrêt d'urgence) avant et durant la mise en service.

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

AVERTISSEMENT

MOUVEMENT IMPRÉVU DE L'AXE ESCLAVE

Désactivez le POU qui commande l'esclave ou déconnectez sa connexion avec le maître si l'axe de l'esclave s'arrête indépendamment du maître.

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

Les blocs fonction de mouvement (à l'exception des blocs fonction de référencement) peuvent être activés uniquement après l'établissement de la référence de la position mécanique. Cela est particulièrement important après le démarrage du bus de mouvement Sercos.

AVERTISSEMENT

RÉFÉRENCEMENT INCORRECT DU SYSTÈME MÉCANIQUE

Vérifiez la présence d'une référence de position mécanique lors de la réalisation des tests de mise en service pour tous les modes d'exploitation.

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

Document(s) à consulter

Titre de documentation	Référence
EcoStruxure Machine Expert – Basic - Fonctions et bibliothèques - Guide de l'utilisateur	EIO0000002829 (ENG) ; EIO0000002830 (FRE) ; EIO0000002831 (GER) ; EIO0000002833 (SPA) ; EIO0000002832 (ITA) ; EIO0000002834 (CHS)
Lexium 62 Drive Device Objects and Parameters	EIO0000003549 (ENG)
EcoStruxure Machine Expert – Basic 3rd Party Motor - Guide de la bibliothèque	EIO0000003555(ENG)

Vous pouvez télécharger ces publications et autres informations techniques depuis notre site web à l'adresse : <https://www.schneider-electric.com/en/download>

Terminologie utilisée dans les normes

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

Dans les domaines des systèmes de sécurité fonctionnelle, des variateurs et de l'automatisme en général, les termes employés sont *sécurité*, *fonction de sécurité*, *état sécurisé*, *défaut*, *réinitialisation du défaut*, *dysfonctionnement*, *panne*, *erreur*, *message d'erreur*, *dangereux*, etc.

Entre autres, les normes concernées sont les suivantes :

Norme	Description
IEC 61131-2:2007	Automates programmables - Partie 2 : exigences et essais des équipements
ISO 13849-1:2015	Sécurité des machines : parties des systèmes de commande relatives à la sécurité. Principes généraux de conception
EN 61496-1:2013	Sécurité des machines : équipements de protection électro-sensibles. Partie 1 : Prescriptions générales et essais
ISO 12100:2010	Sécurité des machines - Principes généraux de conception - Appréciation du risque et réduction du risque
EN 60204-1:2006	Sécurité des machines - Équipement électrique des machines - Partie 1 : règles 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 62061:2015	Sécurité des machines - Sécurité fonctionnelle des systèmes de commande électrique, électronique et électronique programmable relatifs à la sécurité
IEC 61508-1:2010	Sécurité fonctionnelle des systèmes électriques/électroniques/électroniques programmables relatifs à la sécurité : prescriptions générales.
IEC 61508-2:2010	Sécurité fonctionnelle des systèmes électriques/électroniques/électroniques programmables relatifs à la sécurité : exigences pour les systèmes électriques/électroniques/électroniques programmables relatifs à la sécurité.
IEC 61508-3:2010	Sécurité fonctionnelle des systèmes électriques/électroniques/électroniques programmables relatifs à la sécurité : exigences concernant les logiciels.
IEC 61784-3:2016	Réseaux de communication industriels - Profils - Partie 3 : Bus de terrain de sécurité fonctionnelle - Règles générales et définitions de profils.
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 peuvent être utilisés dans le présent document car ils proviennent 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* utilisé dans le contexte de la description de dangers spécifiques a la même signification que les termes *zone dangereuse* ou *zone de danger* employés dans la *directive Machines (2006/42/EC)* et la norme *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.

Chapitre 1

Présentation de la bibliothèque

Contenu de ce chapitre

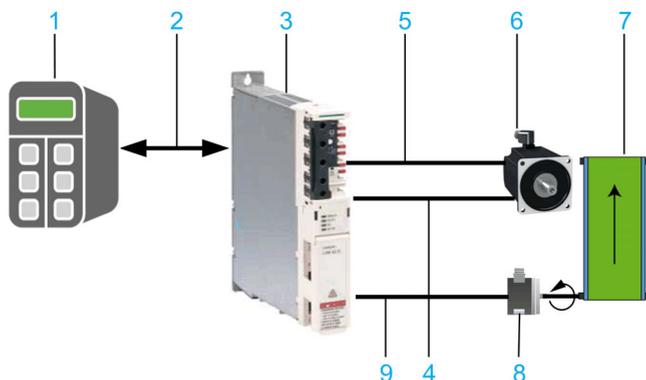
Ce chapitre contient les sujets suivants :

Sujet	Page
Informations générales	20
Concept de diagnostic	25

Informations générales

Présentation de la bibliothèque

La bibliothèque UserMotorTypePlate offre diverses fonctionnalités permettant d'écrire les données de la plaque signalétique d'un moteur dans le codeur ou le variateur du moteur, et de paramétrer un codeur physique.



- 1 Lexium Motion Controller (par exemple, PacDrive LMC)
- 2 Bus Sercos
- 3 Variateur Lexium
- 4 Câble d'alimentation du moteur
- 5 Câble de raccordement moteur/codeur (incompatible avec les moteurs BMP et asynchrones)
- 6 Moteur (éventuellement avec codeur et plage mémoire Hiperface)
- 7 Dispositif mécanique (courroie, par exemple)
- 8 Codeur physique avec plage mémoire Hiperface
- 9 Câble du codeur physique

Présentation fonctionnelle de la bibliothèque

Élément	Composant matériel	Éléments	
1	Lexium Motion Controller	Zone IEC avec variables de type : <ul style="list-style-type: none"> ● ST_UserMotorData (voir page 102) ○ ST_UserMotorDataACIM (voir page 103) ○ ST_UserMotorDataPMSM (voir page 108) ● ST_UserMachineEncoderData (voir page 100) 	Périphérique de stockage de masse (carte Flash, par exemple) <ul style="list-style-type: none"> ● .blh <ul style="list-style-type: none"> ○ Données du moteur fournies par Schneider Electric ● .mdf <ul style="list-style-type: none"> ○ Données du moteur

Elément	Composant matériel	Eléments	
3	Variateur	Composants standard <ul style="list-style-type: none"> ● Micrologiciel du variateur ● FPGA (Field Programmable Gate Array) ● Plaque signalétique du variateur 	Plaque mémoire pour données de moteur de type <code>ST_UserMotorData</code> (<i>voir page 102</i>) pour le moteur raccordé, dans l'impossibilité de stocker directement les données dans le codeur du moteur.
6	Moteur avec codeur optionnel	Codeur de moteur optionnel Eléments <code>ET_EncoderType</code> (<i>voir page 34</i>) compatibles avec les codeurs de moteur : <ul style="list-style-type: none"> ● SinCos Hiperface ● SinCos ● SinCos Hiperface linéaire ● SinCos linéaire 	
8	Codeur physique	Eléments <code>ET_EncoderType</code> (<i>voir page 34</i>) compatibles avec les codeurs physiques : <ul style="list-style-type: none"> ● SinCos Hiperface 	

Conditions préalables/restrictions

- Pour tirer parti des fonctionnalités du codeur physique, celui-ci doit disposer d'une interface Hiperface et vous devez utiliser un variateur Lexium 62 Advanced Plus. Le codeur doit respecter l'une des conditions suivantes, hormis le fait de disposer d'une interface Hiperface :
 - La plaque signalétique du codeur est générique.
 - Le codeur est de type :
 - SC• 60, SR• 50, SK• 36 (•=S ou M)
 - SEK 37, SEL 37, TTK 70 Schneider Electric, L 230
- Les fichiers BLH peuvent être lus sur un contrôleur PacDrive LMC et transférés vers le variateur ou le codeur du moteur. Impossible à créer avec la bibliothèque, ces fichiers sont fournis par Schneider Electric.
- Lorsque le codeur du moteur ou le codeur physique nécessite une plage mémoire Hiperface, la mémoire physique doit faire au moins 2 Ko.

NOTE : avec un moteur tiers, respectez les instructions de la section suivante.

Cas d'utilisation fréquents de la bibliothèque

- Avec un moteur tiers muni d'un codeur SinCos :
 - Si le codeur du moteur dispose de l'interface Hiperface, les données de la plaque signalétique du moteur peuvent être écrites dans ce codeur.
 - En l'absence d'interface Hiperface ou si la mémoire Hiperface est insuffisante, les données de la plaque signalétique du moteur peuvent être écrites directement dans le variateur.
- Avec un moteur tiers sans codeur de moteur : les données de la plaque signalétique du moteur peuvent être écrites directement dans le variateur (moteur asynchrone, par exemple).

- Avec un moteur Lexium BMP et un variateur Lexium 62 Plus ou Lexium 62 Advanced Plus, en écrivant le fichier .blh du moteur dans le variateur.
- Avec Lexium 62 Advanced Plus et un codeur physique disposant d'une interface Hiperface et d'une plage mémoire Hiperface suffisante, en écrivant les données de la plaque signalétique du codeur physique dans le codeur physique.
- Avec un moteur tiers, pour copier les données de codeur du moteur dans un fichier (de sauvegarde).

Utilisation d'un codeur physique

Pour utiliser un codeur physique, procédez comme suit :

Étape	Action
1	Créez les données du codeur à l'aide de la structure <code>ST_UserMachineEncoderData</code> du contrôleur PacDrive LMC. Les données de la structure peuvent être préparées hors ligne. Seule l'étape de transmission des données vers le variateur doit être réalisée pendant la phase 2 du bus Sercos.
2	Transférez les données du codeur vers le codeur physique à l'aide du bloc fonction <code>FB_InitMachineEncoder</code> lorsque le bus Sercos est en phase 2.
3	Redémarrez le bus Sercos (Phase 0 -> Phase 4) pour activer les nouveaux paramètres.

Utilisation d'un moteur tiers

Pour utiliser un moteur tiers, procédez comme suit :

Étape	Action
1	Créez les données du moteur (fichier .mdf) à l'aide de la structure <code>ST_MotorData</code> dans la plage mémoire IEC.
2	Utilisez cette structure comme entrée du bloc fonction <code>FC_MotorDataFileCreate</code> pour créer un fichier de données de moteur dans le contrôleur.
3	Réglez le paramètre <code>MotorIdentification</code> sur la valeur correcte : <ul style="list-style-type: none"> • 0 pour un moteur avec plaque signalétique (plaque stockée dans le codeur). • 2 pour un moteur sans plaque signalétique (plaque stockée dans le variateur).
4	Vérifiez que le bus Sercos est en phase 2.
5	Appelez le bloc fonction <code>FB_MotorDataWrite</code> en utilisant l'objet de variateur et le nom du fichier de données du moteur. Résultat : le bloc fonction signale l'écriture des données du moteur dans le variateur via <code>q_xDone</code> . Vous devez régler le bus Sercos sur la phase 4 dans l'application. L'axe est compatible avec un moteur tiers.

Des exemples d'application avec moteurs tiers sont décrits dans les chapitres Moteur asynchrone (voir *EcoStruxure Machine Expert, 3rd Party Motor, Library Guide*) et Moteur linéaire (voir *EcoStruxure Machine Expert, 3rd Party Motor, Library Guide*).

Caractéristiques de la bibliothèque

Le tableau suivant indique les caractéristiques de la bibliothèque :

Caractéristique	Valeur
Titre de la bibliothèque	UserMotorTypePlate
Société	Schneider Electric
Catégorie	Application
Composant	UserMotorTypePlate
Espace de noms par défaut	MTP
Attribut du modèle de langage	qualified-access-only (<i>voir EcoStruxure Machine Expert, Fonctions et bibliothèques - Guide de l'utilisateur</i>)
Bibliothèque post-compatible	Oui (<i>voir EcoStruxure Machine Expert, Fonctions et bibliothèques - Guide de l'utilisateur</i>)

NOTE : Cette bibliothèque est paramétrée en Uniquement accès qualifié (qualified-access-only). Cela signifie que l'on ne peut accéder aux POU (program organization unit, unité organisationnelle de programme), 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 **MTP**.

Présentation des POU

Bloc fonction / fonction	Utilisation
FB_InitMachineEncoder (<i>voir page 38</i>)	Lit les données de type ST_UserMachineEncoderData dans la plage mémoire IEC et les écrit dans le codeur physique.
FB_MotorDataDelete (<i>voir page 49</i>)	Supprime les données de codeur du moteur dans le variateur ou ce codeur.
FB_MotorDataRead (<i>voir page 55</i>)	Lit les données de moteur du variateur ou du codeur du moteur, et les enregistre dans un fichier .mdf sur le contrôleur.
FB_MotorDataWrite (<i>voir page 60</i>)	Lit les données de moteur du fichier .mdf sur le contrôleur, et les écrit dans le variateur ou le codeur du moteur.
FB_MotorDataWriteBLH (<i>voir page 68</i>)	Lit les données de moteur du fichier .blh sur le contrôleur, et les écrit dans le variateur ou le codeur du moteur.
FB_MotorSerialNumberWrite (<i>voir page 76</i>)	Ecrit le numéro de série du moteur dans la plaque signalétique du moteur, dans le variateur ou le codeur du moteur.
FC_EtDiagExtToString (<i>voir page 82</i>)	Convertit un élément d'énumération de type ET_DiagExt en une chaîne.

Bloc fonction / fonction	Utilisation
FC_MotorDataFileCreate <i>(voir page 83)</i>	Lit les données de type ST_UserMotorData dans la plage mémoire IEC et crée un fichier .mdf sur le contrôleur.
FC_MotorDataFileRead <i>(voir page 95)</i>	Lit le fichier .mdf sur le contrôleur et transfère les données vers la plage mémoire IEC dans une variable de type ST_UserMotorData.

Présentation des structures dans l'interface propre au module

Structure	Utilisation
ST_UserMachineEncoderData <i>(voir page 100)</i>	Contient les données générales du codeur physique.
ST_UserMotorData <i>(voir page 102)</i>	Contient des données générales concernant tous les types de moteur.
ST_UserMotorDataACIM <i>(voir page 103)</i>	Contient des données propres aux moteurs asynchrones (ACIM et inductifs CA).
ST_UserMotorDataPMSM <i>(voir page 108)</i>	Contient des données propres aux moteurs synchrones (PMSM).

Présentation des énumérations

Énumération	Utilisation
ET_DiagExt <i>(voir page 30)</i>	Indique les messages d'état et de diagnostic propres aux POU.
ET_EncoderType <i>(voir page 33)</i>	Contient les types de codeur pris en charge par les variateurs PacDrive 3.
ET_MotorType <i>(voir page 34)</i>	Contient les types de moteur pris en charge par les variateurs PacDrive 3.
ET_StorageLocation <i>(voir page 35)</i>	Indique les espaces de stockage possibles (variateur ou codeur) pour la plaque signalétique électronique du moteur.

Concept de diagnostic

Présentation

PacDrive 3 vous fournit un concept de diagnostic à trois niveaux pour les bibliothèques. Ce concept est valable pour les bibliothèques Technology/Module du système PacDrive 3 (bibliothèque PD_PacDrive.lib, par exemple) 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 de diagnostic et d'état propres au POU (partie 1) : informations détaillées sur la source à l'origine des messages de diagnostic ou d'état.
3. Messages de diagnostic et d'état propres au POU (partie 2) : informations dynamiques détaillées sur la source à l'origine des 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 de la famille de bibliothèques PacDrive 3 offre les avantages suivants :

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

Structure des informations indépendantes des POU

La sortie de diagnostic `q_etDiag` de type `GD.ET_Diag` fournit des informations de diagnostic indépendantes de la bibliothèque, telles que `InputParameterInvalid`. Il peut s'agir d'une solution au problème diagnostiqué.

Selon la valeur de `GD.ET_Diag`, la sortie est soit une description d'état, soit un message d'exception. Une valeur différente de `GD.ET_Diag.Ok` correspond à un message d'exception.

L'énumération `GD.ET_Diag` et ses éléments figurent dans la bibliothèque `PD_GlobalDiagnostics`. Cette bibliothèque inclut également une fonction de conversion pour l'énumération `GD.ET_Diag`.

L'espace de noms par défaut de la bibliothèque `PD_GlobalDiagnostics` est `GD`. L'adressage des POU, structures de données, énumérations et constantes doit s'effectuer au moyen de cet espace de noms.

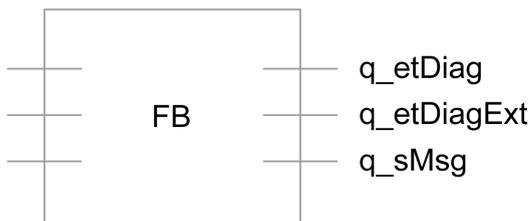
Structure des informations propres aux POU

Les informations de diagnostic des POU peuvent exprimer soit une condition exceptionnelle, soit une condition interne (état) pendant le fonctionnement normal du POU (par exemple, `WaitForStart`). Les informations (condition exceptionnelle ou état) sont rapportées via la même sortie (`q_etDiagExt`). La sortie `q_etDiag` indique si un état ou une exception est signalé.

Blocs fonction

Les blocs fonction comportent trois sorties : `q_etDiag`, `q_etDiagExt` et `q_sMsg` (facultative). Ces sorties sont regroupées, c'est-à-dire qu'elles sont définies l'une après l'autre dans le POU.

L'exemple de bloc fonction ci-après reprend la structure schématique d'un POU :



Sortie	Type de données	Signification
<code>q_etDiag</code>	<code>GD.ET_Diag</code>	<p>Informations de diagnostic générales, par exemple <code>InputParameterInvalid</code>.</p> <p>NOTE : si possible, <code>GD.ET_Diag</code> contient des codes de diagnostic formulés de façon générique (par exemple, <code>DriveConditionInvalid</code> et <code>InputParameterInvalid</code>). Chaque élément d'énumération est représenté par un nom et une valeur. L'IHM peut utiliser ensuite cette valeur afin que le nom de l'énumération puisse être converti dans un langage neutre.</p> <ul style="list-style-type: none"> ● <code>GD.ET_Diag.Ok</code> : le message d'état <code>q_etDiagExt</code> renseigne sur l'état du POU. ● <code><> GD.ET_Diag.Ok</code> : le message de diagnostic <code>q_etDiagExt</code> indique le type d'exception.
<code>q_etDiagExt</code>	<code>ET_DiagExt</code>	<p>Informations de diagnostic détaillées, codées en tant que valeur de la fonction ou du service exécuté dans le POU. Par exemple, <code>AccRange</code> (Accélération hors limites)/<code>WaitForStart</code> peut être renvoyé comme diagnostic ou état.</p> <p><code>q_etDiagExt</code> fournit une valeur numérique faisant office d'index d'une explication plus détaillée de l'origine du message renvoyé, qui peut être utilisée comme index pour des messages d'afficheur dans plusieurs langues.</p>

Sortie	Type de données	Signification
q_sMsg	STRING[80]	<p>Message facultatif déclenché par un événement et fournissant plus de détails sur la condition de diagnostic (par exemple, $0 < i_lrAcc < MaxAcc$).</p> <p>q_sMsg renvoie une chaîne dynamique contenant diverses informations (en anglais) à propos du diagnostic.</p> <p>q_sMsg est modifié en cours d'exécution (par l'exception VelRange: ActualValue: 5003, MaxValue: 5000, par exemple).</p> <p>Au cours du fonctionnement normal des POU ($q_etDiag=GD.ET_Diag.Ok$), q_sMsg peut fournir des informations sur l'état (comme le délai d'étanchéité restant).</p>

Exemple d'informations de diagnostic :

PDL.FB_EndlessFeed		
-i_xEnable		q_xActive
-i_ifDrive		q_xReady
-i_lrVel		q_etDiag InputParameterInvalid
-i_lrAcc		q_etDiagExt AccRange
-i_lrDec		q_sMsg 0 < i_lrAcc < MaxAcc
-i_lrJerk	q_xMotionInstructionActive	
-i_lrPeriod		q_lrPosition
-i_lrStopPosition		q_xInVelocity
-i_xStart		

Fonctions

Les fonctions disposent elles aussi des sorties q_etDiag, q_etDiagExt et q_sMsg (facultative).

S'il n'existe qu'un résultat, il est fourni par la valeur renvoyée directe. Si une fonction possède plusieurs résultats, la valeur renvoyée directe est une structure comprenant les résultats.

Il est également possible de renvoyer les résultats de la fonction via plusieurs sorties. Dans ce cas, la valeur renvoyée directe est une valeur BOOL aléatoire impossible à interpréter.

Lisez la valeur q_etDiag pour vérifier que l'appel de fonction a abouti. En effet, cette information ne figure pas dans la valeur renvoyée directe de la fonction.

Il existe une exception lorsque la valeur renvoyée est $\langle \rangle GD.ET_Diag.Ok$. Les fonctions sont alors entièrement traitées dans la tâche d'appel, leur état est ensuite évalué, puis les sorties q_etDiag et q_etDiagExt indiquent le résultat de l'opération.

Chapitre 2

Énumérations

Contenu de ce chapitre

Ce chapitre contient les sous-chapitres suivants :

Sous-chapitre	Sujet	Page
2.1	ET_DiagExt	30
2.2	ET_EncoderType	33
2.3	ET_MotorType	34
2.4	ET_StorageLocation	35

Sous-chapitre 2.1

ET_DiagExt

ET_DiagExt - Informations générales

Présentation

Type :	Liste
Disponible à partir de la version :	V1.0.0.0

Description

L'énumération ET_DiagExt indique les messages d'état et de diagnostic propres aux POU.

Éléments d'énumération

Nom	Valeur	Description
Ok	0	Aucune erreur détectée.
Init	1	Initialisation
WaitForSercosPhase2	2	Le POU est en attente de la phase 2 du bus Sercos.
WaitForExecute	3	Le POU est en attente d'exécution.
FunctionNotSupportedByThisDevice	4	Le variateur ne prend pas en charge la fonction.
ParameterMotorIdentificationWrong Value	5	La valeur du paramètre MotorIdentification est incorrecte.
CheckingMotorDataInDrive	6	Le POU vérifie si le variateur contient les données du moteur.
SercosCommunicationNotPossible	7	Impossible de communiquer avec le bus Sercos.
NoMotorDataFoundInDrive	8	Aucune donnée de moteur n'a été trouvée dans le variateur.
DeleteMotorDataNotAllowed	9	La suppression des données du moteur est interdite pour les moteurs Schneider Electric.
Done	10	La commande a abouti.
Prepare	11	Le POU prépare son exécution.
MotorDataAlreadyStoredInDrive	12	Les données du moteur figurent déjà dans le variateur.
MotorDataFileNotFound	13	Le fichier est introuvable.
FileAccessNotPossible	14	Le fichier est inaccessible.

Nom	Valeur	Description
ReinitializationFailed	15	La réinitialisation du variateur a échoué.
Disabled	16	Le POU est désactivé.
DeletingMotorData	17	Les données du moteur dans l'axe sont en cours de suppression.
InputStringTooLarge	18	Les chaînes saisies sont trop longues.
NominalSpeedInvalid	19	La vitesse nominale n'est pas valide.
NominalFrequencyInvalid	20	La fréquence nominale n'est pas valide.
NominalVoltageInvalid	21	La tension nominale n'est pas valide.
NominalCurrentInvalid	22	Le courant nominal n'est pas valide.
InvalidNumberOfPolePairs	23	Le nombre de paires de pôles n'est pas valide.
MotorCosPhiInvalid	24	Le cosinus phi du moteur n'est pas valide.
RotatingFieldDirectionInvalid	25	Le sens du champ tournant n'est pas valide.
InvalidMotorName	26	Le nom du moteur n'est pas valide.
InvalidEncoderType	27	Le type de codeur n'est pas valide.
PeakCurrentInvalid	28	Le courant crête du moteur n'est pas valide.
ContinuousStallCurrentInvalid	30	Le courant nominal à l'arrêt n'est pas valide.
ContinuousStallTorqueInvalid	31	Le couple à l'arrêt n'est pas valide.
PeakTorqueInvalid	32	Le couple crête du moteur n'est pas valide.
PhaseResistanceInvalid	33	La résistance d'enroulement n'est pas valide.
QuadraturePhaseInductanceInvalid	34	L'inductance n'est pas valide.
MaxSpeedInvalid	35	La vitesse de rotation maximale n'est pas valide.
MotorInertiaInvalid	36	Le moment d'inertie du moteur n'est pas valide.
BrakeInvalid	37	La valeur <code>uiBrake</code> n'est pas valide.
MandatoryParameterInvalid	38	Le paramètre obligatoire n'est pas valide.
MotorTypeNotSupported	39	Le type de moteur n'est pas pris en charge.
CouldNotCreateFile	40	Impossible de créer le fichier de données du moteur.
InvalidStorageLocation	41	L'espace de stockage sélectionné pour la plaque signalétique électronique du moteur n'est pas valide.
SingleturnResolutionInvalid	42	La résolution monotour du codeur n'est pas valide.
ResolutionFineInvalid	43	La résolution fine du codeur n'est pas valide.
ReadingMotorData	44	Les données du moteur sont lues.
OperationNotAllowed	45	L'axe sélectionné n'autorise pas cette opération.
Executing	46	Le POU est en cours d'exécution.

Énumérations

Nom	Valeur	Description
InvalidDatafield	47	Le numéro du champ de données n'est pas valide.
InvalidAddress	48	L'adresse n'est pas valide.
NominalPowerInvalid	49	La puissance nominale n'est pas valide.
EncoderMaxSpeedInvalid	50	Le paramètre d'entrée <code>EncoderMaxSpeed</code> n'est pas valide.
EncoderMaxTempInvalid	51	Le paramètre d'entrée <code>EncoderMaxTemp</code> n'est pas valide.
EncoderTempSensorInvalid	52	Le paramètre d'entrée <code>EncoderTempSensor</code> n'est pas valide.
EncoderNumberOfTurnsInvalid	53	Le paramètre d'entrée <code>EncoderNumberOfTurns</code> n'est pas valide.
EncoderLinesPerRevolutionInvalid	54	Le paramètre d'entrée <code>EncoderLinesPerRevolution</code> n'est pas valide.
SercosNotInPhaseTwo	55	Le bus Sercos n'est pas en phase 2.
Initializing	56	Initialisation
DriveInvalid	57	Le variateur connecté n'est pas valide.
DriveVirtual	58	Le variateur connecté est virtuel.

Sous-chapitre 2.2

ET_EncoderType

ET_EncoderType - Informations générales

Présentation

Type :	Liste
Disponible à partir de :	V1.1.0.0

Description

L'énumération contient tous les types de codeur pris en charge par les variateurs PacDrive 3.

Éléments d'énumération

Nom	Valeur	Description
None	INT	Le moteur ne dispose pas de codeur.
SincosHiperface	INT	Le moteur dispose d'un codeur rotatif SinCos/Hiperface (par exemple, SKS36).
Sincos	INT	Le moteur dispose d'un codeur rotatif SinCos (1 Vpp).
SincosHiperfaceLinear	INT	Le moteur dispose d'un codeur linéaire SinCos Hiperface (par exemple, TTK70).
SincosLinear	INT	Le moteur dispose d'un codeur linéaire SinCos (1 Vpp).

Sous-chapitre 2.3

ET_MotorType

ET_MotorType - Informations générales

Présentation

Type :	Liste
Disponible à partir de la version :	V1.0.0.0

Description

L'énumération indique le type de moteur.

Éléments d'énumération

Nom	Valeur	Description
RotaryPMSM	INT	Moteur synchrone rotatif (PMSM)
LinearPMSM	INT	Moteur synchrone linéaire (PMSM)
RotaryACIM	INT	Moteur asynchrone rotatif (ACIM)

Sous-chapitre 2.4

ET_StorageLocation

ET_StorageLocation - Informations générales

Présentation

Type :	Énumération
Disponible à partir de :	V1.1.0.0

Description

L'énumération répertorie les espaces de stockage possibles pour la plaque signalétique électronique du moteur.

Éléments d'énumération

Nom	Valeur	Description
Encoder	INT	La plaque signalétique électronique est stockée dans la mémoire du codeur. Le codeur doit disposer d'une interface Hiperface et d'une mémoire physique de 2 Ko minimum.
Drive	INT	La plaque signalétique électronique est stockée dans la mémoire non volatile de l'équipement.

Chapitre 3

Blocs fonction

Contenu de ce chapitre

Ce chapitre contient les sous-chapitres suivants :

Sous-chapitre	Sujet	Page
3.1	FB_InitMachineEncoder	38
3.2	FB_MotorDataDelete	49
3.3	FB_MotorDataRead	55
3.4	FB_MotorDataWrite	60
3.5	FB_MotorDataWriteBLH	68
3.6	FB_MotorSerialNumberWrite	76

Sous-chapitre 3.1

FB_InitMachineEncoder

FB_InitMachineEncoder - Informations générales

Présentation

Type :	Bloc fonction
Disponible à partir de :	V1.3.4.0
Hérite de :	–
Mise en œuvre :	–



Tâche

Ce bloc fonction initialise le codeur physique en fonction d'une plaque signalétique. Les paramètres sont définis dans `i_stUserMachineEncoderData`.

NOTE : Exécutez ce POU uniquement pour les besoins de l'application. Le nombre de cycles d'écriture dans la plage mémoire du codeur étant limité, l'exécution répétée du POU risque d'endommager le codeur.

NOTE : Une plaque signalétique de codeur physique incorrecte influe directement sur le comportement du contrôleur. Cette fonction ne doit être utilisée que par des techniciens qualifiés.

⚠ DANGER

COMPORTEMENT IMPRÉVU DU MOTEUR

- Avant une opération d'écriture, vérifiez les données du codeur physique affectées au moteur.
- Assurez-vous que personne ne se trouve dans la zone d'exploitation au démarrage.

Le non-respect de ces instructions provoquera la mort ou des blessures graves.

Configuration requise

Le bus Sercos doit être en phase 2.

Description fonctionnelle

Ce POU initialise le codeur physique en fonction des paramètres définis dans `i_stUserMachineEncoderData`. Les données du codeur physique sont stockées dans celui-ci.

Les données du codeur physique sont stockées définitivement. Elles ne doivent être écrites qu'une seule fois.

NOTE :

Les données du codeur physique doivent être réécrites dans les cas suivants :

- Le codeur physique est remplacé.
- Le paramétrage est interrompu (par des messages de diagnostic du bus Sercos ou une coupure de courant).
- L'exécution du POU `FB_InitMachineEncoder` a échoué (émission de messages de diagnostic).

NOTE : Le temps de traitement du POU est long (> 1 s). Le POU doit donc être exécuté dans une tâche non soumise à des contraintes de temps, avec un délai de chien de garde adapté.

Interface

Entrée	Type de données	Description
<code>i_xEnable</code>	BOOL	Un front montant FALSE -> TRUE active le POU ; un front descendant TRUE -> FALSE le désactive. Un POU désactivé n'exécute aucune action.
<code>i_xExecute</code>	BOOL	Sur un front montant de cette entrée. Le bloc fonction initialise le codeur physique avec ses données. Pendant l'opération d'initialisation, <code>q_xBusy = TRUE</code> . Une fois les données initialisées, <code>q_xDone</code> devient TRUE.
<code>i_ifDrive</code>	SystemConfigurationIIf.Drive (voir <i>EcoStruxure Machine Expert, SystemConfigurationIIf, Library Guide</i>)	Nom du variateur auquel le codeur physique est raccordé.
<code>i_stUserMachineEncoderData</code>	ST_UserMachineEncoderData (voir <i>page 100</i>)	Structure de données contenant les paramètres du codeur physique.

Sortie	Type de données	Description
q_xActive	BOOL	TRUE : Le POU est actif et son exécution doit se poursuivre. FALSE : Le POU est inactif.
q_xReady	BOOL	TRUE : Le POU est prêt à fonctionner et à accepter les commandes de l'utilisateur. FALSE : Le bloc fonction n'est pas prêt à accepter les commandes de l'utilisateur.
q_xBusy	BOOL	TRUE : Le POU exécute la commande de l'utilisateur. FALSE : Le POU attend d'autres commandes de l'utilisateur.
q_xDone	BOOL	TRUE : La commande de l'utilisateur a été exécutée. FALSE : La commande de l'utilisateur est en cours d'exécution ou l'utilisateur n'a pas émis de commande.
q_etDiag	GD.ET_Diag (voir <i>EcoStruxure Machine Expert, PD_GlobalDiagnostics, Library Guide</i>)	Informations générales sur le diagnostic indépendantes de la bibliothèque. Une valeur différente de GD.ET_Diag.Ok correspond à un message de diagnostic.
q_etDiagExt	ET_DiagExt (voir page 30)	Sortie de diagnostic propre au POU. q_etDiag = GD.ET_Diag.Ok -> message d'état q_etDiag <> GD.ET_Diag.Ok -> message de diagnostic
q_sMsg	STRING[80]	Message déclenché par un événement et fournissant des détails sur l'état du diagnostic.

Messages de diagnostic

q_etDiag	q_etDiagExt	Valeur d'énumération	Description
OK	Disabled (voir page 42)	16	Le POU est désactivé.
OK	Done (voir page 43)	10	La commande a abouti.
OK	Prepare (voir page 46)	11	Le POU prépare son exécution.
OK	WaitForExecute (voir page 47)	3	Le POU est en attente d'exécution.
OK	Executing (voir page 45)	46	Le POU est en cours d'exécution.
InputParameterInvalid	DriveInvalid (voir page 42)	57	Le variateur connecté n'est pas valide.

q_etDiag	q_etDiagExt	Valeur d'énumération	Description
InputParameterInvalid	EncoderLinesPerRevolutionInvalid (voir page 43)	54	Le paramètre d'entrée EncoderLinesPerRevolution n'est pas valide.
InputParameterInvalid	EncoderMaxSpeedInvalid (voir page 43)	50	Le paramètre d'entrée EncoderMaxSpeed n'est pas valide.
InputParameterInvalid	EncoderMaxTempInvalid (voir page 44)	51	Le paramètre d'entrée EncoderMaxTemp n'est pas valide.
InputParameterInvalid	EncoderNumberOfTurnsInvalid (voir page 44)	53	Le paramètre d'entrée EncoderNumberOfTurns n'est pas valide.
InputParameterInvalid	EncoderTempSensorInvalid (voir page 44)	52	Le paramètre d'entrée EncoderTempSensor n'est pas valide.
InputParameterInvalid	FunctionNotSupportedByThisDevice (voir page 45)	4	Le variateur ne prend pas en charge la fonction.
InputParameterInvalid	InvalidEncoderType (voir page 45)	27	Le type de codeur n'est pas valide.
InputParameterInvalid	InvalidStorageLocation (voir page 46)	41	L'espace de stockage sélectionné pour la plaque signalétique électronique du moteur n'est pas valide.
InputParameterInvalid	MEncInObjectNotAppendedToDrive (voir page 46)	59	L'objet MEncIn n'est pas ajouté au variateur dans la configuration de l'automate.
InputParameterInvalid	NoValidMachineEncoderConnected (voir page 46)	60	Aucun codeur physique valide n'est raccordé à l'entrée du matériel.
SercosConditionInvalid	SercosCommunicationNotPossible (voir page 47)	7	Impossible de communiquer avec le bus Sercos.
SercosConditionInvalid	SercosNotInPhaseTwo (voir page 47)	55	Le bus Sercos n'est pas en phase 2.
UnexpectedProgramBehavior	ReinitializationFailed (voir page 47)	15	La réinitialisation du variateur a échoué.

Disabled

Nom d'énumération :	Disabled
Valeur d'énumération :	16
Description :	Le POU est désactivé.

Cause	Solution
Le POU est désactivé.	Faites passer l'entrée <code>i_xEnable</code> de FALSE à TRUE pour activer les POU.

DriveInvalid

Nom d'énumération :	DriveInvalid
Valeur d'énumération :	57
Description :	Le variateur connecté n'est pas valide.

Cause	Solution
Aucun variateur valide n'a été appliqué à l'entrée <code>i_ifDrive</code> du bloc fonction <code>FB_InitMachineEncoder</code> .	Un variateur valide doit être transféré vers <code>i_ifDrive</code> .

DriveVirtual

Nom d'énumération :	DriveVirtual
Valeur d'énumération :	58
Description :	Le variateur connecté est virtuel.

Cause	Solution
Le paramètre <code>WorkingState</code> du variateur raccordé n'est pas de type REAL.	Réglez le paramètre <code>WorkingMode</code> du variateur raccordé sur une valeur REAL.

Done

Nom d'énumération :	Done
Valeur d'énumération :	10
Description :	La commande a abouti.

Les données du moteur ont été supprimées. De nouvelles données de moteur peuvent être écrites dans le variateur.

EncoderLinesPerRevolutionInvalid

Nom d'énumération :	EncoderLinesPerRevolutionInvalid
Valeur d'énumération :	54
Description :	Le paramètre d'entrée <code>EncoderLinesPerRevolution</code> n'est pas valide.

Cause	Solution
Une valeur non valide a été transférée à l'entrée <code>ST_UserMachineEncoderData.uiEncoderLinesPerRevolution</code> .	Transférez une valeur entre <code>MTP.Gc_uiEncoderLinesPerRevolutionMinValue</code> et <code>MTP.Gc_uiEncoderLinesPerRevolutionMaxValue</code> à l'entrée.

EncoderMaxSpeedInvalid

Nom d'énumération :	EncoderMaxSpeedInvalid
Valeur d'énumération :	50
Description :	Le paramètre d'entrée <code>EncoderMaxSpeed</code> n'est pas valide.

Cause	Solution
Une valeur non valide a été transférée à l'entrée <code>ST_UserMachineEncoderData.uiEncoderMaxSpeed</code> .	Transférez une valeur entre <code>MTP.Gc_uiEncoderMaxSpeedMinValue</code> et <code>MTP.Gc_uiEncoderMaxSpeedMaxValue</code> à l'entrée.

EncoderMaxTempInvalid

Nom d'énumération :	EncoderMaxTempInvalid
Valeur d'énumération :	51
Description :	Le paramètre d'entrée EncoderMaxTemp n'est pas valide.

Cause	Solution
Une valeur non valide a été transférée à l'entrée ST_UserMachineEncoderData.uiEncoderMaxTemp.	Transférez une valeur entre MTP.Gc_uiEncoderMaxTempMinValue et MTP.Gc_uiEncoderMaxTempMaxValue à l'entrée.

EncoderNumberOfTurnsInvalid

Nom d'énumération :	EncoderNumberOfTurnsInvalid
Valeur d'énumération :	53
Description :	Le paramètre d'entrée EncoderNumberOfTurns n'est pas valide.

Cause	Solution
Une valeur non valide a été transférée à l'entrée ST_UserMachineEncoderData.uiEncoderNumberOfTurns.	Transférez une valeur entre MTP.Gc_uiEncoderNumberOfTurnsMinValue et MTP.uiEncoderNumberOfTurnsMaxValue à l'entrée.

EncoderTempSensorInvalid

Nom d'énumération :	EncoderTempSensorInvalid
Valeur d'énumération :	52
Description :	Le paramètre d'entrée EncoderTempSensor n'est pas valide.

Cause	Solution
Une valeur non valide a été transférée à l'entrée ST_UserMachineEncoderData.uiEncoderTempSensor.	Transférez la valeur 0: no temperature sensor ou 1: with temperature sensor à l'entrée.

Executing

Nom d'énumération :	Executing
Valeur d'énumération :	46
Description :	Le POU est en cours d'exécution.

Le POU est en cours d'exécution.

FunctionNotSupportedByThisDevice

Nom d'énumération :	FunctionNotSupportedByThisDevice
Valeur d'énumération :	4
Description :	Le variateur ne prend pas en charge la fonction.

Cause	Solution
Type d'objet non valide.	Vérifiez l'objet sélectionné.

InvalidEncoderType

Nom d'énumération :	InvalidEncoderType
Valeur d'énumération :	27
Description :	Le type de codeur n'est pas valide.

Cause	Solution
Une valeur non valide a été transférée à l'entrée ST_UserMachineEncoderData.etEncoderType.	Transférez la valeur ET_EncoderType.SincosHiperface à l'entrée.

InvalidStorageLocation

Nom d'énumération :	InvalidStorageLocation
Valeur d'énumération :	41
Description :	L'espace de stockage sélectionné pour la plaque signalétique électronique du moteur n'est pas valide.

Cause	Solution
Aucun codeur Hiperface n'est raccordé à l'entrée du codeur physique.	Raccordez un codeur Hiperface à l'entrée du codeur physique.

Cause	Solution
La mémoire du codeur est insuffisante ou il n'existe pas de codeur.	Raccordez un codeur avec interface Hiperface disposant d'une mémoire physique de 2 Ko minimum.

MEncInObjectNotAppendedToDrive

Nom d'énumération :	MEncInObjectNotAppendedToDrive
Valeur d'énumération :	59
Description :	L'objet MEncIn n'est pas ajouté au variateur dans la configuration de l'automate.

Cause	Solution
Aucun objet MEncIn n'a été ajouté au variateur.	Ajoutez l'objet MEncIn au variateur dans la configuration de l'automate.
Le variateur dans la configuration de l'automate ne prend pas en charge les objets de codeur physique.	Dans la configuration de l'automate, réglez le variateur sur Lexium 62 Advanced Plus et ajoutez-lui l'objet MEncIn.

NoValidMachineEncoderConnected

Nom d'énumération :	NoValidMachineEncoderConnected
Valeur d'énumération :	60
Description :	Aucun codeur physique valide n'est raccordé à l'entrée du matériel.

Cause	Solution
Aucun codeur SinCos Hiperface n'est raccordé à l'entrée du codeur physique.	Raccordez un codeur SinCos Hiperfac à l'entrée du codeur physique.

Prepare

Nom d'énumération :	Prepare
Valeur d'énumération :	11
Description :	Le POU prépare son exécution.

Le POU est TRUE lors de l'exécution de `q_xBusy`.

ReinitializationFailed

Nom d'énumération :	ReinitializationFailed
Valeur d'énumération :	15
Description :	La réinitialisation du variateur a échoué.

Cause	Solution
La réinitialisation a échoué.	Vérifiez la phase du bus Sercos.

SercosCommunicationNotPossible

Nom d'énumération :	SercosCommunicationNotPossible
Valeur d'énumération :	7
Description :	Impossible de communiquer avec le bus Sercos.

Cause	Solution
Impossible de communiquer avec le bus Sercos.	Vérifiez la phase du bus Sercos et le câblage.

SercosNotInPhaseTwo

Nom d'énumération :	SercosNotInPhaseTwo
Valeur d'énumération :	55
Description :	Le bus Sercos n'est pas en phase 2.

Cause	Solution
Impossible de communiquer avec le bus Sercos. Le bus Sercos doit être en phase 2 pour pouvoir écrire dans le codeur physique.	Mettez le bus Sercos en phase 2.

WaitForExecute

Nom d'énumération :	WaitForExecute
Valeur d'énumération :	3
Description :	Le POU est en attente d'exécution.

Le POU est actif et prêt à être exécuté. `q_xReady` est TRUE.

Exemples

```

VAR
    fbInitMachineEncoder      : MTP.FB_InitMachineEncoder;
    stUserMachineEncoderData  : MTP.ST_UserMachineEncoderData;
    xInitMachineEncoderData   : BOOL := TRUE;
    xEnable                   : BOOL;
    xExecute                   : BOOL;
END_VAR

IF xInitMachineEncoderData THEN
    stUserMachineEncoderData.etEncoderType := MTP.ET_EncoderType.Sincos
Hiperface;
    stUserMachineEncoderData.uiEncoderMaxSpeed      := 12000;
    stUserMachineEncoderData.uiEncoderMaxTemp      := 130;
    stUserMachineEncoderData.uiEncoderTempSensor   := 1;
    stUserMachineEncoderData.uiEncoderNumberOfTurns := 4096;
    stUserMachineEncoderData.uiEncoderLinesPerRevolution := 128;
END_IF

fbInitMachineEncoder(
    i_xEnable           := xEnable,
    i_xExecute          := xExecute,
    i_ifDrive           := DRV_Lexium62AdvancedPlus,
    i_stUserMachineEncoderData := stUserMachineEncoderData,
);

```

Sous-chapitre 3.2

FB_MotorDataDelete

FB_MotorDataDelete - Informations générales

Présentation

Type :	Bloc fonction
Disponible à partir de :	V1.0.0.0
Hérite de :	–
Mise en œuvre :	–



Tâche

Ce bloc fonction supprime les données du moteur dans le variateur ou le codeur, selon l'espace de stockage (`i_etStorageLocation`) sélectionné.

NOTE : Exécutez ce POU uniquement pour les besoins de l'application. Le nombre de cycles d'écriture dans la plage mémoire du codeur étant limité, l'exécution répétée du POU risque d'endommager le codeur.

Configuration requise

Le bus Sercos doit être en phase 2.

Description

Ce bloc fonction supprime les données du moteur dans l'espace de stockage sélectionné. Une fois les anciennes données du moteur supprimées, les nouvelles données peuvent être écrites dans l'axe à l'aide du bloc fonction `FB_MotorDataWrite` (*voir page 60*).

NOTE : Le temps de traitement du POU est long (> 1 s). Le POU doit donc être exécuté dans une tâche non soumise à des contraintes de temps, avec un délai de chien de garde adapté.

Interface

Entrée	Type de données	Description
i_xEnable	BOOL	Un front montant FALSE -> TRUE active le POU ; un front descendant TRUE -> FALSE le désactive. Un POU désactivé n'exécute aucune action.
i_xExecute	BOOL	FALSE -> TRUE : Le bloc fonction supprime les données du moteur dans l'axe sélectionné. Pendant l'opération de suppression, q_xBusy = TRUE. Une fois les données supprimées, q_xDone devient TRUE.
i_ifDrive	SystemConfigurationIlf.IF_Drive <i>(voir EcoStruxure Machine Expert, SystemConfigurationIlf, Library Guide)</i>	Entrée de l'axe à commander.
i_etStorageLocation	ET_StorageLocation <i>(voir page 35)</i>	Espace de stockage à partir duquel sont lues les données du moteur.

Sortie	Type de données	Description
q_xActive	BOOL	TRUE : Le POU est actif et son exécution doit se poursuivre. FALSE : Le POU est inactif.
q_xReady	BOOL	Le POU est prêt à supprimer les données du moteur. Si l'état est FALSE, vérifiez que les conditions sont remplies.
q_xBusy	BOOL	TRUE : Le POU exécute la commande de l'utilisateur (supprimer les données du moteur dans le variateur). FALSE : Le POU attend d'autres commandes de l'utilisateur.
q_xDone	BOOL	TRUE : La commande de l'utilisateur (supprimer les données du moteur dans le variateur) a été exécutée. FB_MotorDataWrite permet de programmer les nouvelles données du moteur. FALSE : La commande de l'utilisateur est en cours d'exécution ou l'utilisateur n'a pas émis de commande.

Sortie	Type de données	Description
q_etDiag	GD.ET_Diag (voir <i>EcoStruxure Machine Expert, PD_GlobalDiagnostics, Library Guide</i>)	Informations générales sur le diagnostic indépendantes de la bibliothèque. Une valeur différente de GD.ET_Diag.Ok correspond à un message de diagnostic.
q_etDiagExt	ET_DiagExt (voir page 30)	Sortie de diagnostic propre au POU. q_etDiag = GD.ET_Diag.Ok -> message d'état q_etDiag <> GD.ET_Diag.Ok -> message de diagnostic
q_sMsg	STRING[80]	Message déclenché par un événement et fournissant des détails sur l'état du diagnostic.

Messages de diagnostic

q_etDiag	q_etDiagExt	Valeur d'énumération	Description
OK	CheckingMotorDataInDrive (voir page 52)	6	Le POU vérifie si le variateur contient les données du moteur.
OK	DeletingMotorData (voir page 52)	17	Les données du moteur dans l'axe sont en cours de suppression.
OK	Disabled (voir page 52)	16	Le POU est désactivé.
OK	Done (voir page 53)	10	La commande a abouti.
OK	WaitForExecute (voir page 54)	3	Le POU est en attente d'exécution.
OK	WaitForSercosPhase2 (voir page 54)	2	Le POU est en attente de la phase 2 du bus Sercos.
DriveConditionInvalid	DeleteMotorDataNotAllowed (voir page 52)	9	La suppression des données du moteur est interdite pour les moteurs Schneider Electric.
DriveConditionInvalid	NoMotorDataFoundInDrive (voir page 54)	8	Aucune donnée de moteur n'a été trouvée dans le variateur.
InputParameterInvalid	FunctionNotSupportedByThisDevice (voir page 53)	4	Le variateur ne prend pas en charge la fonction.
InputParameterInvalid	InvalidStorageLocation (voir page 53)	41	L'espace de stockage sélectionné pour la plaque signalétique électronique du moteur n'est pas valide.
SercosConditionInvalid	SercosCommunicationNotPossible (voir page 54)	7	Impossible de communiquer avec le bus Sercos.

CheckingMotorDataInDrive

Nom d'énumération :	CheckingMotorDataInDrive
Valeur d'énumération :	6
Description :	Le POU vérifie si le variateur contient les données du moteur.

Le bloc fonction est en cours d'exécution. En attente du passage à TRUE de `q_xDone`.

DeleteMotorDataNotAllowed

Nom d'énumération :	DeleteMotorDataNotAllowed
Valeur d'énumération :	9
Description :	La suppression des données du moteur est interdite pour les moteurs Schneider Electric.

Cause	Solution
Les données du moteur dans les axes sélectionnés ne doivent pas être supprimées.	Vérifiez l'axe sélectionné.

DeletingMotorData

Nom d'énumération :	DeletingMotorData
Valeur d'énumération :	17
Description :	Les données du moteur dans l'axe sont en cours de suppression.

Le bloc fonction est en cours d'exécution. En attente du passage à TRUE de `q_xDone`.

Disabled

Nom d'énumération :	Disabled
Valeur d'énumération :	16
Description :	Le POU est désactivé.

Cause	Solution
Le POU est désactivé.	Faites passer l'entrée <code>i_xEnable</code> de FALSE à TRUE pour activer les POU.

Done

Nom d'énumération :	Done
Valeur d'énumération :	10
Description :	La commande a abouti.

Les données du moteur ont été supprimées. De nouvelles données de moteur peuvent être écrites dans le variateur.

FunctionNotSupportedByThisDevice

Nom d'énumération :	FunctionNotSupportedByThisDevice
Valeur d'énumération :	4
Description :	Le variateur ne prend pas en charge la fonction.

Cause	Solution
Type d'objet non valide.	Vérifiez l'objet sélectionné.

InvalidStorageLocation

Nom d'énumération :	InvalidStorageLocation
Valeur d'énumération :	41
Description :	L'espace de stockage sélectionné pour la plaque signalétique électronique du moteur n'est pas valide.

Cause	Solution
Aucun codeur Hiperface n'est raccordé à l'entrée du codeur physique.	Raccordez un codeur Hiperface à l'entrée du codeur physique.
La mémoire du codeur est insuffisante ou il n'existe pas de codeur.	Raccordez un codeur avec interface Hiperface disposant d'une mémoire physique de 2 Ko minimum.

NoMotorDataFoundInDrive

Nom d'énumération :	NoMotorDataFoundInDrive
Valeur d'énumération :	8
Description :	Aucune donnée de moteur n'a été trouvée dans le variateur.

Cause	Solution
Aucune donnée de moteur n'est stockée dans l'axe sélectionné.	Vérifiez l'axe sélectionné.

SercosCommunicationNotPossible

Nom d'énumération :	SercosCommunicationNotPossible
Valeur d'énumération :	7
Description :	Impossible de communiquer avec le bus Sercos.

Cause	Solution
Impossible de communiquer avec le bus Sercos.	Vérifiez la phase du bus Sercos et le câblage.

WaitForExecute

Nom d'énumération :	WaitForExecute
Valeur d'énumération :	3
Description :	Le POU est en attente d'exécution.

Le POU est actif et prêt à être exécuté. `q_xReady` est TRUE.

WaitForSercosPhase2

Nom d'énumération :	WaitForSercosPhase2
Valeur d'énumération :	2
Description :	Le POU est en attente de la phase 2 du bus Sercos.

Le POU est actif et est en attente de la phase 2 du bus Sercos. `q_xReady` est FALSE.

Sous-chapitre 3.3

FB_MotorDataRead

FB_MotorDataRead - Informations générales

Présentation

Type :	Bloc fonction
Disponible à partir de :	V1.1.0.0
Hérite de :	-
Mise en œuvre :	-



Tâche

Ce bloc fonction lit les données du moteur et les écrit dans un fichier binaire.

Description

Les données du moteur sont lues dans l'axe sélectionné, puis écrites dans un fichier binaire que vous pouvez renommer. Ce fichier binaire peut être converti en structure de données `ST_UserMotorData` à l'aide de la fonction `FC_MotorDataRead`.

Interface

Entrée	Type de données	Description
i_xEnable	BOOL	Un front montant FALSE -> TRUE active le POU ; un front descendant TRUE -> FALSE le désactive. Un POU désactivé n'exécute aucune action.
i_xExecute	BOOL	FALSE -> TRUE : Le POU lit les données du moteur dans l'axe sélectionné et les écrit dans un fichier binaire. Pendant l'opération, q_xBusy = TRUE. Une fois l'opération terminée, q_xDone devient TRUE.
i_ifDrive	SystemConfigurationIlf.IF_Drive (voir <i>EcoStruxure Machine Expert, SystemConfigurationIlf, Library Guide</i>)	Entrée de l'axe à commander.
i_sFilename	STRING	Nom du fichier dans lequel seront écrites les données du moteur. Ce fichier est créé automatiquement par le bloc fonction.
i_etStorageLocation	ET_StorageLocation (voir page 35)	Espace de stockage à partir duquel sont lues les données du moteur.

Sortie	Type de données	Description
q_xActive	BOOL	TRUE : Le POU est actif et son exécution doit se poursuivre. FALSE : Le POU est inactif.
q_xReady	BOOL	TRUE : Le POU est prêt à fonctionner et à accepter les commandes de l'utilisateur. FALSE : Le bloc fonction n'est pas prêt à accepter les commandes de l'utilisateur.
q_xBusy	BOOL	TRUE : Le POU exécute la commande de l'utilisateur. FALSE : Le POU attend d'autres commandes de l'utilisateur.
q_xDone	BOOL	TRUE : La commande de l'utilisateur a été exécutée. FALSE : La commande de l'utilisateur est en cours d'exécution ou l'utilisateur n'a pas émis de commande.
q_etDiag	GD.ET_Diag (voir <i>EcoStruxure Machine Expert, PD_GlobalDiagnostics, Library Guide</i>)	Informations générales sur le diagnostic indépendantes de la bibliothèque. Une valeur différente de GD.ET_Diag.Ok correspond à un message de diagnostic.
q_etDiagExt	ET_DiagExt (voir page 30)	Sortie de diagnostic propre au POU. q_etDiag = GD.ET_Diag.Ok -> message d'état q_etDiag <> GD.ET_Diag.Ok -> message de diagnostic

Sortie	Type de données	Description
q_sMsg	STRING[80]	Message déclenché par un événement et fournissant des détails sur l'état du diagnostic.

Messages de diagnostic

q_etDiag	q_etDiagExt	Valeur d'énumération	Description
OK	Disabled <i>(voir page 58)</i>	16	Le POU est désactivé.
OK	Done <i>(voir page 58)</i>	10	La commande a abouti.
OK	ReadingMotorData <i>(voir page 58)</i>	44	Les données du moteur sont lues.
OK	WaitForExecute <i>(voir page 59)</i>	3	Le POU est en attente d'exécution.
OK	WaitForSercosPhase2 <i>(voir page 59)</i>	2	Le POU est en attente de la phase 2 du bus Sercos.
FileHandlingInvalid	CouldNotCreateFile <i>(voir page 57)</i>	40	Impossible de créer le fichier de données du moteur.
SercosConditionInvalid	SercosCommunicationNot Possible <i>(voir page 58)</i>	7	Impossible de communiquer avec le bus Sercos.

CouldNotCreateFile

Nom d'énumération :	CouldNotCreateFile
Valeur d'énumération :	40
Description :	Impossible de créer le fichier de données du moteur.

Cause	Solution
Il existe déjà un fichier portant le nom i_sFilename.	Choisissez un autre nom de fichier.
Le disque flash du contrôleur est plein.	Supprimez les fichiers inutilisés sur le disque flash. Utilisez un disque flash de capacité supérieure.

Disabled

Nom d'énumération :	Disabled
Valeur d'énumération :	16
Description :	Le POU est désactivé.

Cause	Solution
Le POU est désactivé.	Faites passer l'entrée <code>i_xEnable</code> de FALSE à TRUE pour activer les POU.

Done

Nom d'énumération :	Done
Valeur d'énumération :	10
Description :	La commande a abouti.

Les données du moteur ont été lues. Un fichier portant le nom `i_sFilename` a été créé.

ReadingMotorData

Nom d'énumération :	ReadingMotorData
Valeur d'énumération :	44
Description :	Les données du moteur sont lues.

Le bloc fonction est en cours d'exécution. En attente du passage à TRUE de `q_xDone`.

SercosCommunicationNotPossible

Nom d'énumération :	SercosCommunicationNotPossible
Valeur d'énumération :	7
Description :	Impossible de communiquer avec le bus Sercos.

Cause	Solution
Impossible de communiquer avec le bus Sercos.	Vérifiez la phase du bus Sercos et le câblage des équipements Sercos.

WaitForExecute

Nom d'énumération :	WaitForExecute
Valeur d'énumération :	3
Description :	Le POU est en attente d'exécution.

Le POU est actif et prêt à être exécuté. `q_xReady` est TRUE.

WaitForSercosPhase2

Nom d'énumération :	WaitForSercosPhase2
Valeur d'énumération :	2
Description :	Le POU est en attente de la phase 2 du bus Sercos.

Le POU est actif et est en attente de la phase 2 du bus Sercos. `q_xReady` est FALSE.

Sous-chapitre 3.4

FB_MotorDataWrite

FB_MotorDataWrite - Informations générales

Présentation

Type :	Bloc fonction
Disponible à partir de :	V1.0.0.0
Hérite de :	-
Met en œuvre :	-



Tâche

Ce bloc fonction écrit les données du moteur dans l'espace de stockage sélectionné. Reportez-vous également au chapitre Moteur asynchrone (*voir EcoStruxure Machine Expert, 3rd Party Motor, Library Guide*).

NOTE : Exécutez ce POU uniquement pour les besoins de l'application. Le nombre de cycles d'écriture dans la plage mémoire du codeur étant limité, l'exécution répétée du POU risque d'endommager le codeur.

NOTE : Une plaque signalétique de codeur physique incorrecte influe directement sur le comportement du contrôleur. Cette fonction ne doit être utilisée que par des techniciens qualifiés.

⚠ DANGER**COMPORTEMENT IMPRÉVU DU MOTEUR**

- Avant une opération d'écriture, vérifiez les données du codeur physique affectées au moteur.
- Assurez-vous que personne ne se trouve dans la zone d'exploitation au démarrage.

Le non-respect de ces instructions provoquera la mort ou des blessures graves.

NOTE : La désactivation d'une fonction de surveillance risque d'entraîner une surchauffe non détectée du moteur.

⚠ DANGER**INCENDIE EN CAS DE SURCHAUFFE DU MOTEUR**

- Avant une opération d'écriture, vérifiez les données de moteur affectées au moteur.
- Utilisez les méthodes appropriées pour vérifier si le moteur est en surchauffe.
- Assurez-vous que personne ne se trouve dans la zone d'exploitation au démarrage.

Le non-respect de ces instructions provoquera la mort ou des blessures graves.

Configuration requise

- Le bus Sercos doit être en phase 2.
- Aucune donnée de moteur ne doit être stockée dans le variateur sélectionné (utilisez `FB_MotorDataDelete` si besoin).
- Le paramètre `MotorIdentification` (voir *Lexium LXM62 Drive, Device Objects and Parameters*,) de l'axe doit être réglé sur la valeur appropriée.

Description

Ce POU reçoit un fichier contenant les données du moteur et écrit ensuite ces données dans l'axe sélectionné. Il faut d'abord créer le fichier à l'aide du bloc fonction `FC_MotorDataFileCreate` (voir page 83).

Les données du moteur sont stockées dans l'espace sélectionné :

- dans la mémoire du codeur, si l'espace libre est suffisant,
- dans le servo-amplificateur, en l'absence de codeur, si le codeur ne dispose pas de mémoire ou si celle-ci est insuffisante.

Les données du moteur sont stockées définitivement. Elles ne doivent être écrites qu'une seule fois.

NOTE : en cas de remplacement de variateurs ou de moteurs, assurez-vous que les données du moteur stockées dans le variateur ou le moteur sont exactes. Il vous faudra parfois les réécrire.

En prévision de la modification du moteur (suite à son remplacement, par exemple), commencez par supprimer les données de moteur existantes à l'aide du bloc fonction `FB_MotorDataDelete` (*voir page 49*). Ces données ne doivent pas être écrasées.

NOTE : Le temps de traitement du POU est long (> 1 s). Le POU doit donc être exécuté dans une tâche non soumise à des contraintes de temps, avec un délai de chien de garde adapté.

Interface

Entrée	Type de données	Description
<code>i_xEnable</code>	BOOL	Un front montant FALSE -> TRUE active le POU ; un front descendant TRUE -> FALSE le désactive. Un POU désactivé n'exécute aucune action.
<code>i_xExecute</code>	BOOL	FALSE -> TRUE : Le bloc fonction écrit les données du moteur dans l'axe sélectionné. Pendant l'opération d'écriture, <code>q_xBusy</code> = TRUE. Une fois les données écrites, <code>q_xDone</code> devient TRUE.
<code>i_ifDrive</code>	SystemConfigurationIltf.IF_Drive (voir <i>EcoStruxure Machine Expert, SystemConfigurationIltf, Library Guide</i>)	Entrée de l'axe à commander.
<code>i_sFilename</code>	STRING	Nom du fichier de données du moteur binaire créé par le bloc fonction <code>FC_MotorDataFileCreate</code> .
<code>i_etStorageLocation</code>	ET_StorageLocation (voir <i>page 35</i>)	Espace de stockage dans lequel seront écrites les données du moteur. Valeur par défaut : variateur. Le paramètre <code>MotorIdentification</code> (voir <i>Lexium LXM62 Drive, Device Objects and Parameters</i> ,) de l'axe doit être réglé en fonction de l'espace de stockage sélectionné. <ul style="list-style-type: none"> ● <code>ET_StorageLocation.Drive</code> => Motor without nameplate / 2 ● <code>ET_StorageLocation.Encoder</code> => Motor with nameplate / 0

Sortie	Type de données	Description
<code>q_xActive</code>	BOOL	TRUE : Le POU est actif et son exécution doit se poursuivre. FALSE : Le POU est inactif.
<code>q_xReady</code>	BOOL	TRUE : Le POU est prêt à fonctionner et à accepter les commandes de l'utilisateur. FALSE : Le bloc fonction n'est pas prêt à accepter les commandes de l'utilisateur.

Sortie	Type de données	Description
q_xBusy	BOOL	TRUE : Le POU exécute la commande de l'utilisateur. FALSE : Le POU attend d'autres commandes de l'utilisateur.
q_xDone	BOOL	TRUE : La commande de l'utilisateur a été exécutée. FALSE : La commande de l'utilisateur est en cours d'exécution ou l'utilisateur n'a pas émis de commande.
q_etDiag	GD.ET_Diag (voir <i>EcoStruxure Machine Expert, PD_GlobalDiagnostics, Library Guide</i>)	Informations générales sur le diagnostic indépendantes de la bibliothèque. Une valeur différente de GD.ET_Diag.Ok correspond à un message de diagnostic.
q_etDiagExt	ET_DiagExt (voir page 30)	Sortie de diagnostic propre au POU. q_etDiag = GD.ET_Diag.Ok -> message d'état q_etDiag <> GD.ET_Diag.Ok -> message de diagnostic
q_sMsg	STRING[80]	Message déclenché par un événement et fournissant des détails sur l'état du diagnostic.

Messages de diagnostic

q_etDiag	q_etDiagExt	Valeur d'énumération	Description
OK	CheckingMotorDataInDrive (voir page 64)	6	Le POU vérifie si le variateur contient les données du moteur.
OK	Disabled (voir page 64)	16	Le POU est désactivé.
OK	Done (voir page 64)	10	La commande a abouti.
OK	Prepare (voir page 66)	11	Le POU prépare son exécution.
OK	WaitForExecute (voir page 67)	3	Le POU est en attente d'exécution.
OK	WaitForSercosPhase2 (voir page 67)	2	Le POU est en attente de la phase 2 du bus Sercos.
DriveConditionInvalid	MotorDataAlreadyStoredIn Drive (voir page 66)	12	Les données du moteur figurent déjà dans le variateur.
FileHandlingInvalid	FileAccessNotPossible (voir page 65)	14	Le fichier est inaccessible.
FileHandlingInvalid	MotorDataFileNotFound (voir page 66)	13	Le fichier est introuvable.
InputParameterInvalid	FunctionNotSupportedBy ThisDevice (voir page 65)	4	Le variateur ne prend pas en charge la fonction.

q_etDiag	q_etDiagExt	Valeur d'énumération	Description
InputParameterInvalid	InvalidStorageLocation <i>(voir page 65)</i>	41	L'espace de stockage sélectionné pour la plaque signalétique électronique du moteur n'est pas valide.
SercosConditionInvalid	SercosCommunicationNotPossible <i>(voir page 67)</i>	7	Impossible de communiquer avec le bus Sercos.
UnexpectedProgramBehavior	ReinitializationFailed <i>(voir page 66)</i>	15	La réinitialisation du variateur a échoué.

CheckingMotorDataInDrive

Nom d'énumération :	CheckingMotorDataInDrive
Valeur d'énumération :	6
Description :	Le POU vérifie si le variateur contient les données du moteur.

Le bloc fonction est en cours d'exécution. En attente du passage à TRUE de q_xDone.

Disabled

Nom d'énumération :	Disabled
Valeur d'énumération :	16
Description :	Le POU est désactivé.

Cause	Solution
Le POU est désactivé.	Faites passer l'entrée i_xEnable de FALSE à TRUE pour activer les POU.

Done

Nom d'énumération :	Done
Valeur d'énumération :	10
Description :	La commande a abouti.

Les données du moteur ont été écrites.

FileAccessNotPossible

Nom d'énumération :	FileAccessNotPossible
Valeur d'énumération :	14
Description :	Le fichier est inaccessible.

Cause	Solution
Le fichier est inaccessible.	Vérifiez si le fichier est utilisé ailleurs dans le projet.

FunctionNotSupportedByThisDevice

Nom d'énumération :	FunctionNotSupportedByThisDevice
Valeur d'énumération :	4
Description :	Le variateur ne prend pas en charge la fonction.

Cause	Solution
Type d'objet non valide.	Vérifiez l'objet sélectionné.

InvalidStorageLocation

Nom d'énumération :	InvalidStorageLocation
Valeur d'énumération :	41
Description :	L'espace de stockage sélectionné pour la plaque signalétique électronique du moteur n'est pas valide.

Cause	Solution
Aucun codeur Hiperface n'est raccordé à l'entrée du codeur physique.	Raccordez un codeur Hiperface à l'entrée du codeur physique.
La mémoire du codeur est insuffisante ou il n'existe pas de codeur.	Raccordez un codeur avec interface Hiperface disposant d'une mémoire physique de 2 Ko minimum.

MotorDataAlreadyStoredInDrive

Nom d'énumération :	MotorDataAlreadyStoredInDrive
Valeur d'énumération :	12
Description :	Les données du moteur figurent déjà dans le variateur.

Cause	Solution
Le variateur contient déjà les données du moteur.	Les données doivent être supprimées avant de pouvoir en écrire de nouvelles. Le bloc fonction FB_MotorDataDelete permet de réaliser cette opération.

MotorDataFileNotFound

Nom d'énumération :	MotorDataFileNotFound
Valeur d'énumération :	13
Description :	Le fichier est introuvable.

Cause	Solution
Le fichier des données du moteur est introuvable.	Vérifiez le nom du fichier.

Prepare

Nom d'énumération :	Prepare
Valeur d'énumération :	11
Description :	Le POU prépare son exécution.

Le POU est TRUE lors de l'exécution de `q_xBusy`.

ReinitializationFailed

Nom d'énumération :	ReinitializationFailed
Valeur d'énumération :	15
Description :	La réinitialisation du variateur a échoué.

Cause	Solution
La réinitialisation a échoué.	Vérifiez la phase du bus Sercos.

SercosCommunicationNotPossible

Nom d'énumération :	SercosCommunicationNotPossible
Valeur d'énumération :	7
Description :	Impossible de communiquer avec le bus Sercos.

Cause	Solution
Impossible de communiquer avec le bus Sercos.	Vérifiez la phase du bus Sercos et le câblage.

WaitForExecute

Nom d'énumération :	WaitForExecute
Valeur d'énumération :	3
Description :	Le POU est en attente d'exécution.

Le POU est actif et prêt à être exécuté. `q_xReady` est TRUE.

WaitForSercosPhase2

Nom d'énumération :	WaitForSercosPhase2
Valeur d'énumération :	2
Description :	Le POU est en attente de la phase 2 du bus Sercos.

Le POU est actif et est en attente de la phase 2 du bus Sercos. `q_xReady` est FALSE.

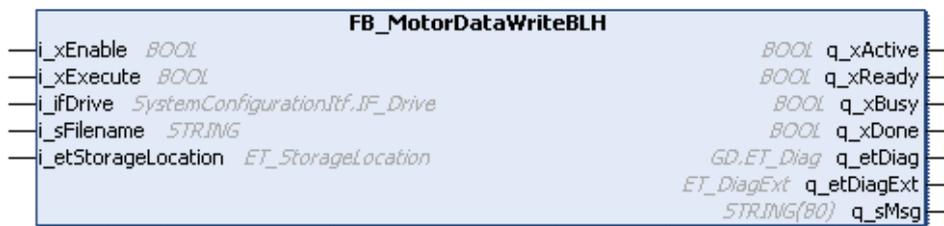
Sous-chapitre 3.5

FB_MotorDataWriteBLH

FB_MotorDataWriteBLH - Informations générales

Présentation

Type :	Bloc fonction
Disponible à partir de :	V1.1.1.5
Hérite de :	-
Met en œuvre :	-



Tâche

Ce bloc fonction écrit les données du moteur dans le variateur. Reportez-vous également au chapitre Moteur asynchrone (*voir EcoStruxure Machine Expert, 3rd Party Motor, Library Guide*).

NOTE : Exécutez ce POU uniquement pour les besoins de l'application. Le nombre de cycles d'écriture dans la plage mémoire du codeur étant limité, l'exécution répétée du POU risque d'endommager le codeur.

NOTE : Une plaque signalétique de codeur physique incorrecte influe directement sur le comportement du contrôleur. Cette fonction ne doit être utilisée que par des techniciens qualifiés.

⚠ DANGER

COMPORTEMENT IMPRÉVU DU MOTEUR

- Avant une opération d'écriture, vérifiez les données du codeur physique affectées au moteur.
- Assurez-vous que personne ne se trouve dans la zone d'exploitation au démarrage.

Le non-respect de ces instructions provoquera la mort ou des blessures graves.

NOTE : La désactivation d'une fonction de surveillance risque d'entraîner une surchauffe non détectée du moteur.

⚠ DANGER

INCENDIE EN CAS DE SURCHAUFFE DU MOTEUR

- Avant une opération d'écriture, vérifiez les données de moteur affectées au moteur.
- Utilisez les méthodes appropriées pour vérifier si le moteur est en surchauffe.
- Assurez-vous que personne ne se trouve dans la zone d'exploitation au démarrage.

Le non-respect de ces instructions provoquera la mort ou des blessures graves.

Configuration requise

- Le bus Sercos doit être en phase 2.
- Aucune donnée de moteur ne doit être stockée dans le variateur sélectionné (utilisez `FB_MotorDataDelete` si besoin).
- Le paramètre `MotorIdentification` (voir *Lexium LXM62 Drive, Device Objects and Parameters*,) de l'axe doit être réglé sur la valeur appropriée.

Description

Ce POU reçoit un fichier BLH contenant les données du moteur et écrit ensuite ces données dans l'axe sélectionné.

Les données du moteur sont stockées dans l'espace sélectionné :

- dans la mémoire du codeur, si l'espace libre est suffisant,
- dans le servo-amplificateur, en l'absence de codeur, si le codeur ne dispose pas de mémoire ou si celle-ci est insuffisante.

Les données du moteur sont stockées définitivement. Elles ne doivent être écrites qu'une seule fois.

NOTE : en cas de remplacement de variateurs ou de moteurs, assurez-vous que les données du moteur stockées dans le variateur ou le moteur sont exactes. Il vous faudra parfois les réécrire.

En prévision de la modification du moteur (suite à son remplacement, par exemple), commencez par supprimer les données de moteur existantes à l'aide du bloc fonction `FB_MotorDataDelete` (voir page 49). Ces données ne doivent pas être écrasées.

NOTE : Le temps de traitement du POU est long (> 1 s). Le POU doit donc être exécuté dans une tâche non soumise à des contraintes de temps, avec un délai de chien de garde adapté.

Interface

Entrée	Type de données	Description
i_xEnable	BOOL	Un front montant FALSE -> TRUE active le POU ; un front descendant TRUE -> FALSE le désactive. Un POU désactivé n'exécute aucune action.
i_xExecute	BOOL	FALSE -> TRUE : Le bloc fonction écrit les données du moteur dans l'axe sélectionné. Pendant l'opération d'écriture, q_xBusy = TRUE. Une fois les données écrites, q_xDone devient TRUE.
i_ifDrive	SystemConfigurationIcf.IF_Drive <i>(voir EcoStruxure Machine Expert, SystemConfigurationIcf, Library Guide)</i>	Entrée de l'axe à commander.
i_sFilename	STRING	Nom du fichier BLH.
i_etStorageLocation	ET_StorageLocation <i>(voir page 35)</i>	Espace de stockage dans lequel seront écrites les données du moteur. Valeur par défaut : variateur. Le paramètre MotorIdentification <i>(voir Lexium LXM62 Drive, Device Objects and Parameters,)</i> de l'axe doit être réglé en fonction de l'espace de stockage sélectionné. <ul style="list-style-type: none"> ● ET_StorageLocation.Drive => Motor without nameplate / 2 ● ET_StorageLocation.Encoder => Motor with nameplate / 0

Sortie	Type de données	Description
q_xActive	BOOL	TRUE : Le POU est actif et son exécution doit se poursuivre. FALSE : Le POU est inactif.
q_xReady	BOOL	TRUE : Le POU est prêt à fonctionner et à accepter les commandes de l'utilisateur. FALSE : Le bloc fonction n'est pas prêt à accepter les commandes de l'utilisateur.
q_xBusy	BOOL	TRUE : Le POU exécute la commande de l'utilisateur. FALSE : Le POU attend d'autres commandes de l'utilisateur.
q_xDone	BOOL	TRUE : La commande de l'utilisateur a été exécutée. FALSE : La commande de l'utilisateur est en cours d'exécution ou l'utilisateur n'a pas émis de commande.

Sortie	Type de données	Description
q_etDiag	GD.ET_Diag (<i>voir EcoStruxure Machine Expert, PD_GlobalDiagnostics, Library Guide</i>)	Informations générales sur le diagnostic indépendantes de la bibliothèque. Une valeur différente de GD.ET_Diag.Ok correspond à un message de diagnostic.
q_etDiagExt	ET_DiagExt (<i>voir page 30</i>)	Sortie de diagnostic propre au POU. q_etDiag = GD.ET_Diag.Ok -> message d'état q_etDiag <> GD.ET_Diag.Ok -> message de diagnostic
q_sMsg	STRING[80]	Message déclenché par un événement et fournissant des détails sur l'état du diagnostic.

Messages de diagnostic

q_etDiag	q_etDiagExt	Valeur d'énumération	Description
OK	CheckingMotorDataInDrive (<i>voir page 72</i>)	6	Le POU vérifie si le variateur contient les données du moteur.
OK	Disabled (<i>voir page 72</i>)	16	Le POU est désactivé.
OK	Done (<i>voir page 72</i>)	10	La commande a abouti.
OK	Prepare (<i>voir page 74</i>)	11	Le POU prépare son exécution.
OK	WaitForExecute (<i>voir page 74</i>)	3	Le POU est en attente d'exécution.
OK	WaitForSercosPhase2 (<i>voir page 75</i>)	2	Le POU est en attente de la phase 2 du bus Sercos.
DriveConditionInvalid	MotorDataAlreadyStoredIn Drive (<i>voir page 73</i>)	12	Les données du moteur figurent déjà dans le variateur.
FileHandlingInvalid	FileAccessNotPossible (<i>voir page 72</i>)	14	Le fichier est inaccessible.
InputParameterInvalid	FunctionNotSupportedBy ThisDevice (<i>voir page 73</i>)	4	Le variateur ne prend pas en charge la fonction.
InputParameterInvalid	InvalidStorageLocation (<i>voir page 73</i>)	41	L'espace de stockage sélectionné pour la plaque signalétique électronique du moteur n'est pas valide.
SercosConditionInvalid	SercosCommunicationNot Possible (<i>voir page 74</i>)	7	Impossible de communiquer avec le bus Sercos.
UnexpectedProgram Behavior	ReinitializationFailed (<i>voir page 74</i>)	15	La réinitialisation du variateur a échoué.

CheckingMotorDataInDrive

Nom d'énumération :	CheckingMotorDataInDrive
Valeur d'énumération :	6
Description :	Le POU vérifie si le variateur contient les données du moteur.

Le bloc fonction est en cours d'exécution. En attente du passage à TRUE de `q_xDone`.

Disabled

Nom d'énumération :	Disabled
Valeur d'énumération :	16
Description :	Le POU est désactivé.

Cause	Solution
Le POU est désactivé.	Faites passer l'entrée <code>i_xEnable</code> de FALSE à TRUE pour activer les POU.

Done

Nom d'énumération :	Done
Valeur d'énumération :	10
Description :	La commande a abouti.

Les données du moteur ont été écrites.

FileAccessNotPossible

Nom d'énumération :	FileAccessNotPossible
Valeur d'énumération :	14
Description :	Le fichier est inaccessible.

Cause	Solution
Le fichier est inaccessible.	Vérifiez si le fichier est utilisé ailleurs dans le projet.

FunctionNotSupportedByThisDevice

Nom d'énumération :	FunctionNotSupportedByThisDevice
Valeur d'énumération :	4
Description :	Le variateur ne prend pas en charge la fonction.

Cause	Solution
Type d'objet non valide.	Vérifiez l'objet sélectionné.

InvalidStorageLocation

Nom d'énumération :	InvalidStorageLocation
Valeur d'énumération :	41
Description :	L'espace de stockage sélectionné pour la plaque signalétique électronique du moteur n'est pas valide.

Cause	Solution
Aucun codeur Hiperface n'est raccordé à l'entrée du codeur physique.	Raccordez un codeur Hiperface à l'entrée du codeur physique.
La mémoire du codeur est insuffisante ou il n'existe pas de codeur.	Raccordez un codeur avec interface Hiperface disposant d'une mémoire physique de 2 Ko minimum.

MotorDataAlreadyStoredInDrive

Nom d'énumération :	MotorDataAlreadyStoredInDrive
Valeur d'énumération :	12
Description :	Les données du moteur figurent déjà dans le variateur.

Cause	Solution
Le variateur contient déjà les données du moteur.	Les données doivent être supprimées avant de pouvoir en écrire de nouvelles. Le bloc fonction FB_MotorDataDelete permet de réaliser cette opération.

Prepare

Nom d'énumération :	Prepare
Valeur d'énumération :	11
Description :	Le POU prépare son exécution.

Le POU est TRUE lors de l'exécution de `q_xBusy`.

ReinitializationFailed

Nom d'énumération :	ReinitializationFailed
Valeur d'énumération :	15
Description :	La réinitialisation du variateur a échoué.

Cause	Solution
La réinitialisation a échoué.	Vérifiez la phase du bus Sercos.

SercosCommunicationNotPossible

Nom d'énumération :	SercosCommunicationNotPossible
Valeur d'énumération :	7
Description :	Impossible de communiquer avec le bus Sercos.

Cause	Solution
Impossible de communiquer avec le bus Sercos.	Vérifiez la phase du bus Sercos et le câblage.

WaitForExecute

Nom d'énumération :	WaitForExecute
Valeur d'énumération :	3
Description :	Le POU est en attente d'exécution.

Le POU est actif et prêt à être exécuté. `q_xReady` est TRUE.

WaitForSercosPhase2

Nom d'énumération :	WaitForSercosPhase2
Valeur d'énumération :	2
Description :	Le POU est en attente de la phase 2 du bus Sercos.

Le POU est actif et est en attente de la phase 2 du bus Sercos. `q_xReady` est FALSE.

Sous-chapitre 3.6

FB_MotorSerialNumberWrite

FB_MotorSerialNumberWrite - Informations générales

Présentation

Type :	Bloc fonction
Disponible à partir de :	V1.1.1.5
Hérite de :	-
Mise en œuvre :	-



Tâche

Ce bloc fonction écrit le numéro de série du moteur.

Description

Ce bloc fonction écrit le numéro de série du moteur dans la plaque signalétique électronique du moteur.

Interface

Entrée	Type de données	Description
<code>i_xEnable</code>	BOOL	Un front montant FALSE -> TRUE active le POU ; un front descendant TRUE -> FALSE le désactive. Un POU désactivé n'exécute aucune action.
<code>i_xExecute</code>	BOOL	FALSE -> TRUE : Le POU écrit le numéro de série du moteur dans l'axe sélectionné. Pendant l'opération d'écriture, <code>q_xBusy</code> = TRUE. Une fois les données écrites, <code>q_xDone</code> devient TRUE.

Entrée	Type de données	Description
i_ifDrive	SystemConfigurationIlf.IF_Drive (voir <i>EcoStruxure Machine Expert, SystemConfigurationIlf, Library Guide</i>)	Entrée de l'axe à commander.
i_etStorageLocation	ET_StorageLocation (voir <i>page 35</i>)	Espace de stockage dans lequel sera écrit le numéro de série (valeur par défaut : variateur).
i_sMotorSerialNumber	STRING	Numéro de série du moteur, composé de 20 caractères maximum.
i_xReinit	BOOL	TRUE : Suite à l'écriture du numéro de série du moteur, le variateur est réinitialisé et la plaque signalétique du moteur est lue à nouveau. FALSE : Aucune réinitialisation du variateur et pas de nouvelle lecture de la plaque signalétique du moteur.

Sortie	Type de données	Description
q_xActive	BOOL	TRUE : Le POU est actif et son exécution doit se poursuivre. FALSE : Le POU est inactif.
q_xReady	BOOL	TRUE : Le POU est prêt à fonctionner et à accepter les commandes de l'utilisateur. FALSE : Le bloc fonction n'est pas prêt à accepter les commandes de l'utilisateur.
q_xBusy	BOOL	TRUE : Le POU exécute la commande de l'utilisateur. FALSE : Le POU attend d'autres commandes de l'utilisateur.
q_xDone	BOOL	TRUE : La commande de l'utilisateur a été exécutée. FALSE : La commande de l'utilisateur est en cours d'exécution ou l'utilisateur n'a pas émis de commande.
q_etDiag	GD.ET_Diag (voir <i>EcoStruxure Machine Expert, PD_GlobalDiagnostics, Library Guide</i>)	Informations générales sur le diagnostic indépendantes de la bibliothèque. Une valeur différente de GD.ET_Diag.Ok correspond à un message de diagnostic.
q_etDiagExt	ET_DiagExt (voir <i>page 30</i>)	Sortie de diagnostic propre au POU. q_etDiag = GD.ET_Diag.Ok -> message d'état q_etDiag <> GD.ET_Diag.Ok -> message de diagnostic
q_sMsg	STRING[80]	Message déclenché par un événement et fournissant des détails sur l'état du diagnostic.

Messages de diagnostic

q_etDiag	q_etDiagExt	Valeur d'énumération	Description
OK	Disabled <i>(voir page 78)</i>	16	Le POU est désactivé.
OK	Done <i>(voir page 78)</i>	10	La commande a abouti.
OK	Executing <i>(voir page 79)</i>	46	Le POU est en cours d'exécution.
OK	WaitForExecute <i>(voir page 80)</i>	3	Le POU est en attente d'exécution.
OK	WaitForSercosPhase2 <i>(voir page 80)</i>	2	Le POU est en attente de la phase 2 du bus Sercos.
DriveConditionInvalid	OperationNotAllowed <i>(voir page 79)</i>	45	L'axe sélectionné n'autorise pas cette opération.
InputParameterInvalid	FunctionNotSupportedByThisDevice <i>(voir page 79)</i>	4	Le variateur ne prend pas en charge la fonction.
InputParameterInvalid	InputStringTooLarge <i>(voir page 79)</i>	18	Les chaînes saisies sont trop longues.
SercosConditionInvalid	SercosCommunicationNotPossible <i>(voir page 80)</i>	7	Impossible de communiquer avec le bus Sercos.

Disabled

Nom d'énumération :	Disabled
Valeur d'énumération :	16
Description :	Le POU est désactivé.

Cause	Solution
Le POU est désactivé.	Faites passer l'entrée <code>i_xEnable</code> de FALSE à TRUE pour activer les POU.

Done

Nom d'énumération :	Done
Valeur d'énumération :	10
Description :	La commande a abouti.

Le numéro de série du moteur a été écrit.

Executing

Nom d'énumération :	Executing
Valeur d'énumération :	46
Description :	Le POU est en cours d'exécution.

Le POU est en cours d'exécution.

FunctionNotSupportedByThisDevice

Nom d'énumération :	FunctionNotSupportedByThisDevice
Valeur d'énumération :	4
Description :	Le variateur ne prend pas en charge la fonction.

Cause	Solution
Type d'objet non valide.	Vérifiez l'objet sélectionné.

InputStringTooLarge

Nom d'énumération :	InputStringTooLarge
Valeur d'énumération :	18
Description :	Les chaînes saisies sont trop longues.

Cause	Solution
Le numéro de série du moteur dépasse 20 caractères.	Indiquez un numéro de série inférieur à 20 caractères pour le moteur.

OperationNotAllowed

Nom d'énumération :	OperationNotAllowed
Valeur d'énumération :	45
Description :	L'axe sélectionné n'autorise pas cette opération.

Cause	Solution
L'axe sélectionné n'autorise pas l'opération (un servo-moteur Schneider Electric est utilisé, par exemple).	Sélectionnez un axe autorisant cette opération.

SercosCommunicationNotPossible

Nom d'énumération :	SercosCommunicationNotPossible
Valeur d'énumération :	7
Description :	Impossible de communiquer avec le bus Sercos.

Cause	Solution
Impossible de communiquer avec le bus Sercos.	Vérifiez la phase du bus Sercos et le câblage.

WaitForExecute

Nom d'énumération :	WaitForExecute
Valeur d'énumération :	3
Description :	Le POU est en attente d'exécution.

Le POU est actif et prêt à être exécuté. `q_xReady` est TRUE.

WaitForSercosPhase2

Nom d'énumération :	WaitForSercosPhase2
Valeur d'énumération :	2
Description :	Le POU est en attente de la phase 2 du bus Sercos.

Le POU est actif et est en attente de la phase 2 du bus Sercos. `q_xReady` est FALSE.

Chapitre 4

Fonctions

Contenu de ce chapitre

Ce chapitre contient les sous-chapitres suivants :

Sous-chapitre	Sujet	Page
4.1	FC_EtDiagExtToString	82
4.2	FC_MotorDataFileCreate	83
4.3	FC_MotorDataFileRead	95

Sous-chapitre 4.1

FC_EtDiagExtToString

FC_EtDiagExtToString - Informations générales

Présentation

Type :	Fonction
Disponible à partir de :	V1.0.0.0

Tâche

Cette fonction convertit en chaîne un élément d'énumération de type ET_DiagExt (*voir page 30*) de la bibliothèque UserMotorTypePlate.

Description

Chaque élément d'énumération est représenté par un nom ou une valeur. La valeur renvoyée par la fonction correspond au nom de l'élément d'énumération.

Interface

Entrée	Type de données	Description
i_etDiagExt	ET_DiagExt (<i>voir page 30</i>)	Valeur ET_DiagExt à convertir.

Sortie	Type de données	Description
q_etDiag	GD.ET_Diag (<i>voir EcoStruxure Machine Expert, PD_GlobalDiagnostics, Library Guide</i>)	Informations générales sur le diagnostic indépendantes de la bibliothèque. Une valeur différente de ET_Diag.Ok correspond à un message de diagnostic.
q_etDiagExt	ET_DiagExt (<i>voir page 30</i>)	Sortie de diagnostic propre au POU. q_etDiag = ET_Diag.Ok -> message d'état q_etDiag <> ET_Diag.Ok -> message de diagnostic

Valeur retournée

Type de données	Description
STRING[80]	Nom de l'élément d'énumération transmis au niveau de l'entrée i_etDiagExt.

Sous-chapitre 4.2

FC_MotorDataFileCreate

FC_MotorDataFileCreate - Informations générales

Présentation

Type :	Fonction
Disponible à partir de :	V1.0.0.0
Hérite de :	–
Met en œuvre :	–

Tâche

Cette fonction crée un fichier binaire de données de moteur. Reportez-vous également au chapitre Moteur asynchrone (*voir EcoStruxure Machine Expert, 3rd Party Motor, Library Guide*).

NOTE : Une plaque signalétique de codeur physique incorrecte influe directement sur le comportement du contrôleur. Cette fonction ne doit être utilisée que par des techniciens qualifiés.

DANGER

COMPORTEMENT IMPRÉVU DU MOTEUR

- Les données techniques fournies par le fabricant du moteur doivent être appliquées et respectées.
- Les variables moteur paramétrées doivent correspondre aux données de moteur du fabricant.
- En cas de conflit entre la définition des données de moteur et le moteur, ajustez la définition en fonction des données de moteur requises.
- Assurez-vous que personne ne se trouve dans la zone d'exploitation au démarrage.

Le non-respect de ces instructions provoquera la mort ou des blessures graves.

NOTE : La désactivation d'une fonction de surveillance risque d'entraîner une surchauffe non détectée du moteur.

⚠ DANGER

INCENDIE EN CAS DE SURCHAUFFE DU MOTEUR

- Les données techniques fournies par le fabricant du moteur doivent être appliquées et respectées.
- Les variables moteur paramétrées doivent correspondre aux données de moteur du fabricant.
- En cas de conflit entre la définition des données de moteur et le moteur, ajustez la définition en fonction des données de moteur requises.
- Utilisez les méthodes appropriées pour vérifier si le moteur est en surchauffe.
- Assurez-vous que personne ne se trouve dans la zone d'exploitation au démarrage.

Le non-respect de ces instructions provoquera la mort ou des blessures graves.

Description

Cette fonction crée un fichier binaire de données de moteur pouvant être combiné avec le bloc fonction `FB_MotorDataWrite` (*voir page 60*) afin d'écrire les données du moteur dans un axe. Vous devez indiquer le nom du fichier et une structure contenant les données du moteur.

NOTE : Le temps de traitement du POU est long (> 1 s). Le POU doit donc être exécuté dans une tâche non soumise à des contraintes de temps, avec un délai de chien de garde adapté.

Interface

Entrée	Type de données	Description
<code>i_sFilename</code>	STRING[80]	Nom du fichier à créer.
<code>i_stUserMotorData</code>	ST_UserMotorData (<i>voir page 102</i>)	Structure contenant les données du moteur écrites dans le fichier.

Sortie	Type de données	Description
<code>q_etDiag</code>	GD.ET_Diag (<i>voir EcoStruxure Machine Expert, PD_GlobalDiagnostics, Library Guide</i>)	Informations générales sur le diagnostic indépendantes de la bibliothèque. Une valeur différente de <code>GD.ET_Diag.Ok</code> correspond à un message de diagnostic.
<code>q_etDiagExt</code>	ET_DiagExt (<i>voir page 30</i>)	Sortie de diagnostic propre au POU. <code>q_etDiag = GD.ET_Diag.Ok</code> -> message d'état <code>q_etDiag <> GD.ET_Diag.Ok</code> -> message de diagnostic
<code>q_sMsg</code>	STRING[80]	Message déclenché par un événement et fournissant des détails sur l'état du diagnostic.

Valeur retournée

Type de données	Description
BOOL	TRUE si le fichier a été créé.

Messages de diagnostic

q_etDiag	q_etDiagExt	Valeur d'énumération	Description
OK	Disabled (<i>voir page 88</i>)	16	Le POU est désactivé.
FileHandlingInvalid	CouldNotCreateFile (<i>voir page 87</i>)	40	Impossible de créer le fichier de données du moteur.
InputParameterInvalid	BrakeInvalid (<i>voir page 86</i>)	37	La valeur uiBrake n'est pas valide.
InputParameterInvalid	ContinuousStallCurrentInvalid (<i>voir page 87</i>)	30	Le courant nominal à l'arrêt n'est pas valide.
InputParameterInvalid	ContinuousStallTorqueInvalid (<i>voir page 87</i>)	31	Le couple à l'arrêt n'est pas valide.
InputParameterInvalid	InputStringTooLarge (<i>voir page 88</i>)	18	Les chaînes saisies sont trop longues.
InputParameterInvalid	InvalidEncoderType (<i>voir page 88</i>)	27	Le type de codeur n'est pas valide.
InputParameterInvalid	InvalidMotorName (<i>voir page 89</i>)	26	Le nom du moteur n'est pas valide.
InputParameterInvalid	InvalidNumberOfPolePairs (<i>voir page 89</i>)	23	Le nombre de paires de pôles n'est pas valide.
InputParameterInvalid	MandatoryParameterInvalid (<i>voir page 89</i>)	38	Le paramètre obligatoire n'est pas valide.
InputParameterInvalid	MaxSpeedInvalid (<i>voir page 89</i>)	35	La vitesse de rotation maximale n'est pas valide.
InputParameterInvalid	MotorCosPhiInvalid (<i>voir page 90</i>)	24	Le cosinus phi du moteur n'est pas valide.
InputParameterInvalid	MotorInertiaInvalid (<i>voir page 90</i>)	36	Le moment d'inertie du moteur n'est pas valide.
InputParameterInvalid	MotorTypeNotSupported (<i>voir page 90</i>)	39	Le type de moteur n'est pas pris en charge.
InputParameterInvalid	NominalCurrentInvalid (<i>voir page 91</i>)	22	Le courant nominal n'est pas valide.

q_etDiag	q_etDiagExt	Valeur d'énumération	Description
InputParameterInvalid	NominalFrequencyInvalid <i>(voir page 91)</i>	20	La fréquence nominale n'est pas valide.
InputParameterInvalid	NominalPowerInvalid	49	La puissance nominale n'est pas valide.
InputParameterInvalid	NominalSpeedInvalid <i>(voir page 92)</i>	19	La vitesse nominale n'est pas valide.
InputParameterInvalid	NominalVoltageInvalid <i>(voir page 92)</i>	21	La tension nominale n'est pas valide.
InputParameterInvalid	PeakCurrentInvalid <i>(voir page 92)</i>	28	Le courant crête du moteur n'est pas valide.
InputParameterInvalid	PeakTorqueInvalid <i>(voir page 93)</i>	32	Le couple crête du moteur n'est pas valide.
InputParameterInvalid	PhaseResistanceInvalid <i>(voir page 93)</i>	33	La résistance d'enroulement n'est pas valide.
InputParameterInvalid	QuadraturePhaseInductanceInvalid <i>(voir page 93)</i>	34	L'inductance n'est pas valide.
InputParameterInvalid	RotatingFieldDirectionInvalid <i>(voir page 94)</i>	25	Le sens du champ tournant n'est pas valide.
InputParameterInvalid	SingleTurnResolutionInvalid <i>(voir page 94)</i>	42	La résolution monotour du codeur n'est pas valide.

BrakeInvalid

Nom d'énumération :	BrakeInvalid
Valeur d'énumération :	37
Description :	La valeur uiBrake n'est pas valide.

Cause	Solution
La valeur de uiBrake n'est pas valide. Seules les valeurs 0 et 1 sont autorisées.	Réglez uiBrake sur une valeur valide.

ContinuousStallCurrentInvalid

Nom d'énumération :	ContinuousStallCurrentInvalid
Valeur d'énumération :	30
Description :	Le courant nominal à l'arrêt n'est pas valide.

Cause	Solution
Le courant nominal à l'arrêt doit être supérieur à 0.	Ajustez le courant nominal à l'arrêt.

ContinuousStallTorqueInvalid

Nom d'énumération :	ContinuousStallTorqueInvalid
Valeur d'énumération :	31
Description :	Le couple à l'arrêt n'est pas valide.

Cause	Solution
Le couple à l'arrêt doit être supérieur à 0.	Ajustez le couple à l'arrêt.

CouldNotCreateFile

Nom d'énumération :	CouldNotCreateFile
Valeur d'énumération :	40
Description :	Impossible de créer le fichier de données du moteur.

Cause	Solution
Il existe déjà un fichier portant le même nom.	Choisissez un autre nom de fichier.
L'espace disponible sur le disque flash est insuffisant.	Libérez de l'espace sur le disque ou utilisez un disque Flash de capacité supérieure.
Disque flash défectueux.	Remplacez le disque flash.

Disabled

Nom d'énumération :	Disabled
Valeur d'énumération :	16
Description :	Le POU est désactivé.

Cause	Solution
Le POU est désactivé.	Faites passer l'entrée <code>i_xEnable</code> de FALSE à TRUE pour activer les POU.

InputStringTooLarge

Nom d'énumération :	InputStringTooLarge
Valeur d'énumération :	18
Description :	Les chaînes saisies sont trop longues.

Cause	Solution
La longueur du paramètre <code>sMotorname</code> , <code>sMotorArticleNumber</code> ou <code>sMotorSerialnumber</code> n'est pas valide. La limite est fixée à 20 caractères.	Ajustez la longueur de la chaîne.

InvalidEncoderType

Nom d'énumération :	InvalidEncoderType
Valeur d'énumération :	27
Description :	Le type de codeur n'est pas valide.

Cause	Solution
La valeur de <code>uiEncoderType</code> n'est pas valide.	Ajustez la valeur.

InvalidMotorName

Nom d'énumération :	InvalidMotorName
Valeur d'énumération :	26
Description :	Le nom du moteur n'est pas valide.

Cause	Solution
Les noms de moteur SH, BSH et BMH sont interdits.	Utilisez un nom de moteur autorisé.

InvalidNumberOfPolePairs

Nom d'énumération :	InvalidNumberOfPolePairs
Valeur d'énumération :	23
Description :	Le nombre de paires de pôles n'est pas valide.

Cause	Solution
Le nombre de paires de pôles ne doit pas être égal à 0.	Indiquez le nombre de paires de pôles du moteur.

MandatoryParameterInvalid

Nom d'énumération :	MandatoryParameterInvalid
Valeur d'énumération :	38
Description :	Le paramètre obligatoire n'est pas valide.

Cause	Solution
Un paramètre obligatoire a une valeur non valide.	Ajustez le paramètre.

MaxSpeedInvalid

Nom d'énumération :	MaxSpeedInvalid
Valeur d'énumération :	35
Description :	La vitesse de rotation maximale n'est pas valide.

Cause	Solution
La valeur 0 n'est pas autorisée.	Ajustez la valeur.

MotorCosPhiInvalid

Nom d'énumération :	MotorCosPhiInvalid
Valeur d'énumération :	24
Description :	Le cosinus phi du moteur n'est pas valide.

Cause	Solution
Le cosinus phi du moteur doit être supérieur à 0 et inférieur à 1.	Ajustez le valeur du cosinus phi.

MotorInertiaInvalid

Nom d'énumération :	MotorInertiaInvalid
Valeur d'énumération :	36
Description :	Le moment d'inertie du moteur n'est pas valide.

Cause	Solution
Le moment d'inertie du moteur ne doit pas être égal à 0.	Ajustez le moment d'inertie.

MotorTypeNotSupported

Nom d'énumération :	MotorTypeNotSupported
Valeur d'énumération :	39
Description :	Le type de moteur n'est pas pris en charge.

La bibliothèque prend en charge les types de moteur suivants :

- Moteurs asynchrones rotatifs
- Moteurs synchrones linéaires
- Moteurs synchrones rotatifs à six paires de pôles et plus

NominalCurrentInvalid

Nom d'énumération :	NominalCurrentInvalid
Valeur d'énumération :	22
Description :	Le courant nominal n'est pas valide.

Cause	Solution
Le courant nominal doit être supérieur à 0.	Ajustez le courant nominal.

NominalFrequencyInvalid

Nom d'énumération :	NominalFrequencyInvalid
Valeur d'énumération :	20
Description :	La fréquence nominale n'est pas valide.

Cause	Solution
La fréquence nominale ne doit pas être égale à 0.	Ajustez la fréquence nominale.

NominalPowerInvalid

Nom d'énumération :	NominalPowerInvalid
Valeur d'énumération :	49
Description :	La puissance nominale n'est pas valide.

Cause	Solution
Le paramètre <code>i_stUserMotorData.stMotorDataACIM.rNominalPower</code> est hors limites.	<code>i_stUserMotorData.stMotorDataACIM.rNominalPower</code> doit être compris entre 0 et 110, et lorsque <code>i_stUserMotorData.stMotorDataACIM.rNominalPower = 0</code> , la valeur de <code>i_stUserMotorData.stMotorDataACIM.uiNominalPower</code> doit être supérieure à 0. Ajustez les valeurs tel que décrit.

NominalSpeedInvalid

Nom d'énumération :	NominalSpeedInvalid
Valeur d'énumération :	19
Description :	La vitesse nominale n'est pas valide.

Cause	Solution
La vitesse nominale ne doit pas être égale à 0.	Ajustez la vitesse nominale.

NominalVoltageInvalid

Nom d'énumération :	NominalVoltageInvalid
Valeur d'énumération :	21
Description :	La tension nominale n'est pas valide.

Cause	Solution
La tension nominale ne doit pas être égale ni inférieure à 0.	Ajustez la tension nominale.

PeakCurrentInvalid

Nom d'énumération :	PeakCurrentInvalid
Valeur d'énumération :	28
Description :	Le courant crête du moteur n'est pas valide.

Cause	Solution
Le courant crête doit être supérieur à 0.	Ajustez le courant crête.

PeakTorqueInvalid

Nom d'énumération :	PeakTorqueInvalid
Valeur d'énumération :	32
Description :	Le couple crête du moteur n'est pas valide.

Cause	Solution
Le couple crête doit être supérieur à 0.	Ajustez le couple crête.

PhaseResistanceInvalid

Nom d'énumération :	PhaseResistanceInvalid
Valeur d'énumération :	33
Description :	La résistance d'enroulement n'est pas valide.

Cause	Solution
La résistance d'enroulement doit être supérieure à 0.	Ajustez la résistance d'enroulement.

QuadraturePhaseInductanceInvalid

Nom d'énumération :	QuadraturePhaseInductanceInvalid
Valeur d'énumération :	34
Description :	L'inductance n'est pas valide.

Cause	Solution
L'inductance doit être supérieure à 0.	Ajustez la valeur.

RotatingFieldDirectionInvalid

Nom d'énumération :	RotatingFieldDirectionInvalid
Valeur d'énumération :	25
Description :	Le sens du champ tournant n'est pas valide.

Cause	Solution
Le sens du champ tournant doit être égal à 0 ou 1.	Ajustez le sens du champ tournant.

SingleturnResolutionInvalid

Nom d'énumération :	SingleturnResolutionInvalid
Valeur d'énumération :	42
Description :	La résolution monotour du codeur n'est pas valide.

Cause	Solution
La longueur des données monotour et la résolution monotour ne correspondent pas.	Ajustez la longueur des données monotour. Ajustez la résolution monotour.

Sous-chapitre 4.3

FC_MotorDataFileRead

FC_MotorDataFileRead - Informations générales

Présentation

Type :	Fonction
Disponible à partir de :	V1.1.0.0
Hérite de :	-
Mise en œuvre :	-

Tâche

Cette fonction convertit un fichier binaire de données de moteur en structure de données.

Description

Cette fonction lit le fichier binaire indiqué et écrit les données du moteur qu'il contient dans une structure `ST_UserMotorData`.

Interface

Entrée	Type de données	Description
<code>i_sFilename</code>	STRING[80]	Nom du fichier binaire contenant les données du moteur qui doit être lu.

Sortie	Type de données	Description
<code>q_etDiag</code>	GD.ET_Diag (voir <i>EcoStruxure Machine Expert</i> , <i>PD_GlobalDiagnostics</i> , <i>Library Guide</i>)	Informations générales sur le diagnostic indépendantes de la bibliothèque. Une valeur différente de <code>GD.ET_Diag.Ok</code> correspond à un message de diagnostic.
<code>q_etDiagExt</code>	ET_DiagExt (voir page 30)	Sortie de diagnostic propre au POU. <code>q_etDiag = GD.ET_Diag.Ok</code> -> message d'état <code>q_etDiag <> GD.ET_Diag.Ok</code> -> message de diagnostic
<code>q_sMsg</code>	STRING[80]	Message déclenché par un événement et fournissant des détails sur l'état du diagnostic.

Entrée/sortie	Type de données	Description
iq_stUserMotorData	ST_UserMotorData	Les données du moteur lues dans le fichier i_sFilename sont écrites dans cette structure.

Valeur retournée

Type de données	Description
BOOL	TRUE = Le fichier a été lu et les données du moteur ont été écrites dans la structure iq_stUserMotorData. FALSE = Le fichier n'a pas été lu ou une erreur est survenue lors de la lecture des données du moteur (pour en savoir plus, reportez-vous aux messages de diagnostic).

Messages de diagnostic

q_etDiag	q_etDiagExt	Valeur d'énumération	Description
OK	Disabled (<i>voir page 96</i>)	16	Le POU est désactivé.
FileHandlingInvalid	FileAccessNotPossible (<i>voir page 97</i>)	14	Le fichier est inaccessible.
FileHandlingInvalid	MotorDataFileNotFound (<i>voir page 97</i>)	13	Le fichier est introuvable.

Disabled

Nom d'énumération :	Disabled
Valeur d'énumération :	16
Description :	Le POU est désactivé.

Faites passer l'entrée i_xEnable de FALSE à TRUE pour activer les POU.

FileAccessNotPossible

Nom d'énumération :	FileAccessNotPossible
Valeur d'énumération :	14
Description :	Le fichier est inaccessible.

Cause	Solution
Le fichier est utilisé par un autre bloc fonction.	L'accès au fichier est autorisé à condition qu'aucun autre bloc fonction ne l'utilise.

MotorDataFileNotFound

Nom d'énumération :	MotorDataFileNotFound
Valeur d'énumération :	13
Description :	Le fichier est introuvable.

Cause	Solution
Le nom de fichier indiqué est incorrect.	Vérifiez le paramètre <code>i_sFilename</code> .

Chapitre 5

Structures

Contenu de ce chapitre

Ce chapitre contient les sous-chapitres suivants :

Sous-chapitre	Sujet	Page
5.1	ST_UserMachineEncoderData	100
5.2	ST_UserMotorData	102
5.3	ST_UserMotorDataACIM	103
5.4	ST_UserMotorDataPMSM	108

Sous-chapitre 5.1

ST_UserMachineEncoderData

ST_UserMachineEncoderData - Informations générales

Présentation

Type :	Structure de données
Disponible à partir de :	V1.0.0.0
Hérite de :	-

Description

Cette structure contient les données générales du codeur physique.

Éléments de la structure

Variable	Type de données	Description
etEncoderType	ET_EncoderType <i>(voir page 33)</i>	Type de codeur (par exemple, SinCos Hiperface, SinCos). Paramètre obligatoire
uiEncoderMaxSpeed	UINT	Vitesse maximale du codeur en [tr/min]. Facultatif Valeur par défaut : 0 Si uiEncoderMaxSpeed est égal à 0, la vitesse du codeur n'est pas surveillée. Si uiEncoderMaxSpeed est supérieur à 0, la vitesse du codeur est surveillée.
uiEncoderMaxTemp	UINT	Température maximale du codeur en [°C]. Facultatif Valeur par défaut : 0 Si uiEncoderTempSensor est égal à 0, la valeur est ignorée. La température maximale du codeur n'est pas surveillée. Si uiEncoderTempSensor est égal à 1 ET uiEncoderMaxTemp est différent de 0, la température du codeur est surveillée par rapport au niveau de température.

Variable	Type de données	Description
uiEncoderTempSensor	UINT (0...1)	Indique si le codeur intègre un capteur de température : Facultatif 0 : aucun capteur de température ; en présence d'un capteur, la surveillance de la température du codeur est désactivée. 1 : un capteur de température est présent ; lorsque uiEncoderMaxTemp est différent de 0, la température du codeur est surveillée par rapport à uiEncoderMaxTemp.
uiEncoderNumberOfTurns	UINT	Nombre de rotations affichables du codeur. Facultatif Codeurs Hiperface : la valeur est ignorée, les valeurs suivantes sont considérées à la place : <ul style="list-style-type: none"> ● Codeurs avec plaque signalétique générique : les valeurs sont celles de la plaque signalétique générique du codeur. ● Codeurs sans plaque signalétique générique SC• 60, SR• 50, SK• 36 (*=S ou M), SEK 37, SEL 37, TTK 70 Schneider Electric version, L 230 : les valeurs codées en dur sont utilisées.
uiEncoderLinesPerRevolution	UINT	Nombre de périodes par rotation du codeur. Facultatif Codeurs Hiperface : la valeur est ignorée, les valeurs suivantes sont considérées à la place : <ul style="list-style-type: none"> ● Codeurs avec plaque signalétique générique : les valeurs sont celles de la plaque signalétique générique du codeur. ● Codeurs sans plaque signalétique générique SC• 60, SR• 50, SK• 36 (*=S ou M), SEK 37, SEL 37, TTK 70 Schneider Electric version, L 230 : les valeurs codées en dur sont utilisées.

Sous-chapitre 5.2

ST_UserMotorData

ST_UserMotorData - Informations générales

Présentation

Type :	Structure de données
Disponible à partir de :	V1.0.0.0
Hérite de :	-

Description

Cette structure contient des données générales concernant tous les types de moteur.

Éléments de la structure

Variable	Type de données	Description
etMotorType	ET_MotorType <i>(voir page 34)</i>	Types de moteur : <ul style="list-style-type: none"> ● Rotary_PMSM ● Linear_PMSM ● Rotary_ACIM Paramètre obligatoire
sMotorname	STRING[80]	Nom du moteur (affiché par le paramètre <code>MotorType</code>), composé de 20 caractères maximum. Facultatif
sMotorSerialNumber	STRING[80]	Numéro de série du moteur (affiché par le paramètre <code>SerialNumberMotor</code>). Facultatif
sMotorArticleNumber	STRING[80]	Référence du moteur (affiché par le paramètre <code>PartNumberMotor</code>). Facultatif
stMotorDataACIM	ST_UserMotorData ACIM <i>(voir page 103)</i>	Données des moteurs asynchrones. Cette structure doit être renseignée lorsque <code>etMotorType</code> est réglé sur <code>Rotary_ACIM</code> .
stMotorDataPMSM	ST_UserMotorData PMSM <i>(voir page 108)</i>	Données des moteurs synchrones. Cette structure doit être renseignée lorsque <code>etMotorType</code> est réglé sur <code>Rotary_PMSM</code> ou <code>Linear_PMSM</code> .

Sous-chapitre 5.3

ST_UserMotorDataACIM

ST_UserMotorDataACIM - Informations générales

Présentation

Type :	Structure de données
Disponible à partir de :	V1.0.0.0
Hérite de :	-

Description

Cette structure contient des données propres aux moteurs asynchrones (ACIM et inductifs CA).

Éléments de la structure

Variable	Type de données	Description
uiNominalSpeed	UINT	Vitesse nominale en [tr/min]. Paramètre obligatoire
uiNominalFrequency	UINT	Fréquence nominale en [Hz]. Paramètre obligatoire
rNominalPower	REAL	Puissance nominale en [kW]. Paramètre obligatoire
uiNominalPower	UINT	Le paramètre rNominalPower est recommandé. Le paramètre uiNominalPower est utilisé lorsque rNominalPower est égal à 0. Puissance nominale en [W] Facultatif
rNominalVoltage	REAL	Tension nominale en [Veff]. Paramètre obligatoire
rNominalCurrent	REAL	Courant nominal en [Aeff]. Paramètre obligatoire
rMotorCosPhi	REAL	Cosinus (phi) du moteur. Paramètre obligatoire

Variable	Type de données	Description
uiEncoderType	UINT	Type de codeur (voir ET_EncoderType (<i>voir page 33</i>)) : <ul style="list-style-type: none"> ● 0 : aucun codeur ● 1 : codeur SinCos Hiperface ● 2 : codeur SinCos Paramètre obligatoire
uiBrake	UINT	Indique si le moteur intègre un frein : <ul style="list-style-type: none"> ● 0 : aucun frein intégré ● 1 : frein intégré Paramètre obligatoire
uiTempSensorType	UINT	Type de capteur thermique : <ul style="list-style-type: none"> ● 0 : sans fonction ● 1 : capteur non linéaire (par exemple, PTC) ● 2 : capteur linéaire (par exemple, KTY) Paramètre obligatoire
rStatorResistance	REAL	Résistance statorique du moteur (phase à phase) en [Ω]. Facultatif
rRotorResistance	REAL	Résistance rotorique du moteur (phase à phase) en [Ω]. Facultatif
rMainInductance	REAL	Inductance principale du moteur (phase à phase) en [μH]. Facultatif
rStatorLeakageInductance	REAL	Inductance de fuite statorique du moteur (phase à phase) en [μH]. Facultatif
rRotorLeakInductance	REAL	Inductance de fuite rotorique du moteur (phase à phase) en [μH]. Facultatif
uiPolePair	UINT	Nombre de paires de pôles. Si uiPolePair est à 0, ce paramètre est calculé à l'aide du bloc fonction FC_MotorDataFileCreate. Facultatif
uiRotatingFieldDirection	UINT	Sens du champ tournant : <ul style="list-style-type: none"> ● 0 : sens anti-horaire ● 1 : sens horaire Si les phases du moteur, <ul style="list-style-type: none"> ● U -> U ● V -> V ● W -> W sont raccordées au variateur, ce paramètre doit être réglé sur 1 (sens horaire). Facultatif

Variable	Type de données	Description
<code>uiMaxMotorTemperature</code>	UINT	Température maximale du moteur en [°C] (valeur par défaut : 130 °C). Facultatif
<code>uiTempSensorResistanceOvertemp</code>	UINT	Résistance minimale du capteur de température en cas de température excessive en [Ω] (valeur par défaut : 4 k Ω). Ce paramètre est utilisé seulement pour les capteurs PTC. Le bloc fonction <code>FC_MotorDataFileCreate</code> détermine les caractéristiques du capteur en fonction de ce paramètre. La valeur est utilisée avec un capteur de température de type 1 uniquement. Facultatif
<code>auiTempSensorChracteristics</code>	ARRAY [0..19] OF UINT	Caractéristiques du capteur thermique. Ce paramètre est utilisé seulement pour les capteurs linéaires. Dans ce cas, vous devez indiquer les valeurs suivantes : <ul style="list-style-type: none"> ● <code>auiTempSensorChracteristics [0]</code> : résistance en [Ω] à -30 °C ● <code>auiTempSensorChracteristics [1]</code> : résistance en [Ω] à -20 °C ● <code>auiTempSensorChracteristics [2]</code> : résistance en [Ω] à -10 °C ● ... ● <code>auiTempSensorChracteristics [19]</code> : résistance en [Ω] à 160 °C Facultatif
<code>uiMaxSpeed</code>	UINT	Vitesse de rotation maximale du moteur en [tr/min]. Facultatif
<code>rPeakCurrent</code>	REAL	Courant crête du moteur en [Aeff]. Si <code>rPeakCurrent</code> est égal à 0, la valeur utilisée correspond à 1,5 fois la valeur du courant nominal. Facultatif
<code>uiEncoderMaxSpeed</code>	UINT	Vitesse de rotation maximale du codeur en [tr/min]. Facultatif
<code>uiEncoderMaxTemp</code>	UINT	Température maximale du codeur en [°C]. Facultatif
<code>uiEncoderTempSensor</code>	UINT	Indique si le codeur intègre un capteur de température : <ul style="list-style-type: none"> ● 0 : aucun capteur de température ● 1 : capteur de température Facultatif
<code>uiEncoderNumberOfTurns</code>	UINT	Nombre de rotations affichables du codeur. Paramètre obligatoire pour les codeurs SinCos (sans interface Hiperface)

Variable	Type de données	Description
uiEncoderLinesPerRevolution	UINT	Nombre de périodes par rotation du codeur. Paramètre obligatoire pour les codeurs SinCos (sans interface Hiperface)
wBrakeType	WORD	Type de frein si celui-ci est présent (valeur par défaut : 0) : 0 : frein de maintien par défaut 1 : frein de maintien inversé Facultatif
uiBrakeDisconnectionTime	UINT	Délai de relâchement du frein en [ms]. Si uiBrakeDisconnectionTime est égal à 0, le variateur utilise la valeur par défaut 100 ms. Facultatif
uiBrakeCouplingTime	UINT	Délai d'actionnement du frein en [ms]. Si uiBrakeCouplingTime est égal à 0, le variateur utilise la valeur par défaut 100 ms. Facultatif
rBrakeMinVoltage	REAL	Tension de contrôle minimale pour utiliser le frein en [V]. Si rBrakeMinVoltage est égal à 0, le variateur ne vérifie pas la tension du frein. Facultatif
rBrakeMaxVoltage	REAL	Tension de contrôle maximale pour utiliser le frein en [V]. Si rBrakeMaxVoltage est égal à 0, le variateur ne vérifie pas la tension du frein. Facultatif
uiBrakeNomCurrent	UINT	Courant nominal du frein en [mA]. Facultatif
uiThermalConstant	UINT	Constante de temps permettant de surveiller la surchauffe du moteur en [ms]. Ce paramètre définit la durée pendant laquelle le moteur peut fonctionner au courant crête sans surchauffer. La valeur par défaut est 1 000 ms. Facultatif

Exemple

Dans l'exemple suivant, un moteur asynchrone est configuré pour le mode de fonctionnement U/f. Les données du moteur sont les suivantes :

- Vitesse nominale = 1 380 tr/min
- Cosinus phi = 0,7
- Fréquence nominale = 50 Hz
- Puissance nominale = 370 W
- Tension nominale = 400 V
- Courant nominal = 1,14 A
- Le courant de démarrage est 3,5 fois supérieur au courant nominal (dans cet exemple, le courant de démarrage est utilisé comme courant crête).
- Le moteur ne dispose pas de codeur.
- Le moteur dispose d'un capteur de température non linéaire. En cas de température excessive au niveau du moteur, la résistance de ce capteur est supérieure à 4 kΩ.
- Le moteur ne dispose pas de frein.
- Les phases du moteur U => U, V => V, W => W sont raccordées au variateur (champ tournant électrique dans le sens horaire).

Les données du moteur sont intégrées dans la structure de données comme suit :

```

stUserMotorData.eMotorType := Rotary_ACIM;
stUserMotorData.sMotorname := 'SEW-Eurodrive';
stUserMotorData.sMotorArticleNumber := 'DRS71S4/FF/TH/AV1H';
stUserMotorData.sMotorSerialNumber := '01.1710743001.0002X'; // 11
stUserMotorData.stMotorData_ACIM.uiNominalSpeed := 1380;
stUserMotorData.stMotorData_ACIM.uiPolePair := 2;
stUserMotorData.stMotorData_ACIM.rMotorCosPhi := 0.70;
stUserMotorData.stMotorData_ACIM.uiNominalFrequency := 50;
stUserMotorData.stMotorData_ACIM.rNominalPower := 0.37;
stUserMotorData.stMotorData_ACIM.uiNominalVoltage := 400;
stUserMotorData.stMotorData_ACIM.rNominalCurrent := 1.14; // A
stUserMotorData.stMotorData_ACIM.rPeakCurrent :=
    stUserMotorData.stMotorData_ACIM.rNominalCurrent * 3.5;
stUserMotorData.stMotorData_ACIM.uiEncoderType :=
    MTP.ET_EncoderType.None; stUserMotorData.stMotorData_ACIM.uiTempSen-
    sorType := 1; stUserMotorData.stMotorData_ACIM.uiMaxSpeed := 1920;
stUserMotorData.stMotorData_ACIM.uiBrake := 0;
stUserMotorData.stMotorData_ACIM.uiRotatingFieldDirection := 1;
stUserMotorData.stMotorData_ACIM.uiTempSensorResistanceOvertemp :=
    4000;

```

Sous-chapitre 5.4

ST_UserMotorDataPMSM

ST_UserMotorDataPMSM - Informations générales

Présentation

Type :	Structure de données
Disponible à partir de :	V1.0.0.0
Hérite de :	-

Description

Cette structure contient des données propres aux moteurs PMSM (moteurs synchrones à aimants permanents).

Éléments de la structure

Variable	Type de données	Description
uiEncoderType	UINT	Type de codeur (voir ET_EncoderType (<i>voir page 33</i>)) : <ul style="list-style-type: none"> • 1 : codeur SinCos Hiperface (rotatif) • 2 : codeur SinCos (rotatif) • 257 : codeur SinCos Hiperface (linéaire) • 258 : codeur SinCos (linéaire) Paramètre obligatoire
uiNominalSpeed	UINT	Vitesse nominale en [tr/min] ou [mm/s]. Paramètre obligatoire
rNominalVoltage	REAL	Tension nominale en [Veff]. $rNominalVoltage = PhaseVoltage * root3$ Paramètre obligatoire
rPeakCurrent	REAL	Courant crête du moteur en [Aeff]. Paramètre obligatoire
rNominalCurrent	REAL	Courant nominal du moteur à la vitesse nominale en [Aeff]. Lorsque la fiche de données techniques du moteur ne fait pas la distinction entre le courant nominal à l'arrêt et le courant nominal à la vitesse nominale, réglez rNominalCurrent et rContStallCurrent sur la même valeur. Paramètre obligatoire

Variable	Type de données	Description
rContStallCurrent	REAL	Courant nominal du moteur à l'arrêt [Aeff]. Lorsque la fiche de données techniques du moteur ne fait pas la distinction entre le courant nominal à l'arrêt et le courant nominal à la vitesse nominale, réglez rNominalCurrent et rContStallCurrent sur la même valeur. Paramètre obligatoire
uiPolePair	UINT	Nombre de paires de pôles. Paramètre obligatoire pour les moteurs rotatifs
rConstStallTorque	REAL	Couple nominal du moteur en [Nm] ou force nominale du moteur en [N]. Paramètre obligatoire
rPeakTorque	REAL	Couple crête du moteur en [Nm] ou force crête du moteur en [N]. Paramètre obligatoire
rPhaseResistance	REAL	Résistance d'enroulement du moteur (phase à phase) en [Ω]. Paramètre obligatoire
rQuadraturePhaseInductance	REAL	Inductance d'enroulement (composant q) du moteur (phase à phase) en [μ H]. Lorsque la fiche de données techniques du moteur ne fait pas la distinction entre le composant d et le composant q, réglez rDirectPhaseInductance et rQuadraturePhaseInductance sur la même valeur. Paramètre obligatoire
uiRotatingFieldDirection	UINT	Sens du champ tournant (valeur par défaut 0) : <ul style="list-style-type: none"> ● 0 : sens anti-horaire ● 1 : sens horaire Si les phases du moteur, <ul style="list-style-type: none"> ● U -> U ● V -> V ● W -> W sont raccordées au variateur, ce paramètre doit être réglé sur 1 (sens horaire). Facultatif
uiMaxSpeed	UINT	Vitesse de rotation maximale du moteur en [tr/min] ou [mm/s]. Paramètre obligatoire
udiMotorIntertia	UDINT	Moment d'inertie du moteur en [gcm ²] (moteur rotatifs) ou masse du moteur en [g] (moteurs linéaires). Paramètre obligatoire

Variable	Type de données	Description
uiBrake	UINT	Indique si le moteur intègre un frein : <ul style="list-style-type: none"> ● 0 : aucun frein intégré ● 1 : frein intégré Paramètre obligatoire
uiTempSensorType	UINT	Type de capteur thermique : <ul style="list-style-type: none"> ● 0 : aucun capteur ● 1 : capteur non linéaire (par exemple, PTC) ● 2 : capteur linéaire (par exemple, KTY) Paramètre obligatoire
uiMaxMotorTemperature	UINT	Température maximale du moteur en [°C] (valeur par défaut : 130 °C). Facultatif
uiTempSensorResistanceOvertemp	UINT	Résistance minimale du capteur de température en cas de température excessive en [Ω] (valeur par défaut : 4 k Ω). Ce paramètre est utilisé seulement pour les capteurs PTC. Le bloc fonction FC_MotorDataFileCreate détermine les caractéristiques du capteur en fonction de ce paramètre. La valeur est utilisée avec un capteur de température de type 1 uniquement. Facultatif
auTempSensorCharacteristics	ARRAY [0..19] OF UINT	Courbe caractéristique du capteur de température. Ce paramètre est utilisé seulement pour les capteurs linéaires. Dans ce cas, vous devez indiquer les valeurs suivantes : <ul style="list-style-type: none"> ● auTempSensorChracteristics[0] : résistance en [Ω] à -30 °C ● auTempSensorChracteristics[1] : résistance en [Ω] à -20 °C ● auTempSensorChracteristics[2] : résistance en [Ω] à -10 °C ● ... ● auTempSensorChracteristics[19] : résistance en [Ω] à 160 °C Facultatif

Variable	Type de données	Description
rInsulationSystemVoltage	REAL	Tension d'isolement du moteur en [V]. La tension d'isolement du moteur est liée à la tension maximale admise au niveau d'une phase moteur, et non à la tension du bus CC. La formule suivante permet de convertir la tension sur le bus CC en tension au niveau d'une phase moteur : $U_{\text{Motorphase}} := (U_{\text{Dcbus}} * 1,15) / (1,4142 * 2)$ Exemple : si la tension maximale autorisée sur le bus CC pour un moteur est 500 V, la tension maximale admise au niveau d'une phase moteur est égale à $(500 \text{ V} * 1,15) / (1,4142 * 2) = 203$. Facultatif
uiEncoderMaxSpeed	UINT	Vitesse de rotation maximale du codeur en [tr/min]. Facultatif
uiEncoderMaxTemp	UINT	Température maximale du codeur en [°C]. Facultatif
uiEncoderTempSensor	UINT	Indique si le codeur intègre un capteur de température : <ul style="list-style-type: none"> ● 0 : aucun capteur de température ● 1 : capteur de température Facultatif
uiEncoderNumberOfTurns	UINT	Nombre de rotations affichables du codeur. Paramètre obligatoire pour les codeurs SinCos (sans interface Hiperface)
uiEncoderLinesPerrevolution	UINT	Nombre de périodes par rotation du codeur. Paramètre obligatoire pour les codeurs SinCos (sans interface Hiperface)
rDirectPhaseInductance	REAL	Inductance d'enroulement (composant d) du moteur (phase à phase) en [μH]. Lorsque la fiche de données techniques du moteur ne fait pas la distinction entre le composant d et le composant q, réglez rDirectPhaseInductance et rQuadraturePhaseInductance sur la même valeur. Paramètre obligatoire
rEMK_Constant	REAL	Constante EMF (force électromotrice) (phase à phase) en [Veff / 1 000 tr/min] ou [Veff / m/s]. NOTE : La formule suivante permet de convertir la constante EMF d'une phase (valeur de type chaîne) en valeur phase à phase : $\text{EMF_Constant_Phase_to_Phase} := \text{EMF_Constant_Phase} * \text{SQRT}(3);$ Paramètre obligatoire

Variable	Type de données	Description
uiModelC1	UINT	Paramètre C1 pour le modèle thermique. Facultatif
uiModelA12	UINT	Paramètre A12 pour le modèle thermique. Facultatif
uiModelDeltaA	UINT	Paramètre DeltaA pour le modèle thermique. Facultatif
uiModelDeltaB	UINT	Paramètre DeltaB pour le modèle thermique. Facultatif
wBrakeType	WORD	Type de frein si celui-ci est présent (valeur par défaut : 0) : 0 : frein de maintien par défaut 1 : frein de maintien inversé Facultatif
uiBrakeDisconnectionTime	UINT	Délai de relâchement du frein en [ms]. Si uiBrakeDisconnectionTime est égal à 0, le variateur utilise la valeur par défaut 100 ms. Facultatif
uiBrakeCouplingTime	UINT	Délai d'actionnement du frein en [ms]. Si uiBrakeCouplingTime est égal à 0, le variateur utilise la valeur par défaut 100 ms. Facultatif
rBrakeMinVoltage	REAL	Tension de contrôle minimale pour utiliser le frein en [V]. Si rBrakeMinVoltage est égal à 0, la tension du frein du variateur n'est pas surveillée. Facultatif
rBrakeMaxVoltage	REAL	Tension de contrôle maximale pour utiliser le frein en [V]. Si rBrakeMaxVoltage est égal à 0, la tension du frein du variateur n'est pas surveillée. Facultatif
uiBrakeNomCurrent	UINT	Courant nominal du frein en [mA]. Facultatif
uiThermalConstant	UINT	Constante de temps permettant de surveiller la surchauffe du moteur en [ms]. Ce paramètre définit la durée pendant laquelle le moteur peut fonctionner au courant crête sans surchauffer. La valeur par défaut est 1 000 ms. Facultatif
rPeriodLength	REAL	Longueur de période du codeur en [µm]. Paramètre obligatoire pour les moteurs linéaires
rPolePairPitch	REAL	Distance entre deux paires de pôles en [µm]. Paramètre obligatoire pour les moteurs linéaires

Exemple

Dans l'exemple suivant, un moteur synchrone linéaire est configuré. Les données du moteur sont les suivantes :

- Vitesse nominale = 3 200 mm/s
- Tension nominale = 600 V
- Courant nominal = 9,4 A
- Courant crête = 20,7 A
- Force nominale = 5 N
- Force crête = 200 N
- Résistance d'enroulement = 4,34 Ω
- Inductance d'enroulement = 32 mH
- Constante EMF = 920 V / m/s
- Longueur de période = 5 mm
- Distance entre deux paires de pôles = 2,4 mm
- Masse du moteur = 6,3 kg
- Le moteur dispose d'un codeur linéaire SinCos Hiperface.
- Le moteur dispose d'un capteur de température non linéaire. En cas de température excessive au niveau du moteur, la résistance de ce capteur est supérieure à 4 k Ω .
- Le moteur ne dispose pas de frein.

Les données du moteur sont intégrées dans la structure de données comme suit :

```
stUserMotorData.eMotorType := Linear_PMSM;
stUserMotorData.sMotorname := 'Tec-TL30_N';
stUserMotorData.sMotorArticleNumber := 'ETYP00056';
stUserMotorData.sMotorSerialNumber := '201138.00';
stUserMotorData.stMotorDataPMSM.uiEncoderType :=
    MTP.ET_EncoderType.SincosHiperfaceLinear;
stUserMotorData.stMotorDataPMSM.uiPeriodLength := 5000; //  $\mu$ m
stUserMotorData.stMotorDataPMSM.uiNominalSpeed := 3200; // mm/s
stUserMotorData.stMotorDataPMSM.uiNominalVoltage := 600; // V
stUserMotorData.stMotorDataPMSM.rNominalCurrent := 9.4; // A
stUserMotorData.stMotorDataPMSM.rPeakCurrent := 20.7; // A
stUserMotorData.stMotorDataPMSM.rContStallCurrent := 9.4; // A
stUserMotorData.stMotorDataPMSM.rConstStallTorque := 5; // N
stUserMotorData.stMotorDataPMSM.rPeakTorque := 200; // N
stUserMotorData.stMotorDataPMSM.rPhaseResistance := 4.34; // Ohm
stUserMotorData.stMotorDataPMSM.rQuadraturePhaseInductance :=
    32000; // uH
stUserMotorData.stMotorDataPMSM.rDirectPhaseInductance := 32000; // uH
```

```
stUserMotorData.stMotorDataPMSM.uiRotatingFieldDirection := 1;
stUserMotorData.stMotorDataPMSM.udiEMK_Constant := 920; // V / m/s
stUserMotorData.stMotorDataPMSM.uiPolePairPitch := 2400; // µm
stUserMotorData.stMotorDataPMSM.uiMaxMotorTemperature := 130; //°C
stUserMotorData.stMotorDataPMSM.uiTempSensorType := 1;
stUserMotorData.stMotorDataPMSM.uiMaxSpeed := 3200; // mm/s
stUserMotorData.stMotorDataPMSM.udiMotorIntertia := 6300; // g
stUserMotorData.stMotorDataPMSM.uiBrake := 0;
stUserMotorData.stMotorDataACIM.uiTempSensorResistanceOvertemp := 4000;
```

Chapitre 6

Éléments globaux

Sous-chapitre 6.1

GCL (Global Constant List)

GCL (Global Constant List) - Informations générales

Présentation

Type :	Constantes globales
Disponible à partir de la version :	V1.0.0.0

Description

Liste des constantes globales (GCL) de la bibliothèque UserMotorTypePlate.

Constantes globales

Variable	Type de données	Valeur	Description
Gc_sLibraryVersion	STRING [80]	Vx.x.x.x	Version de la bibliothèque.
Gc_byMotorTypePlateVersion	BYTE	2	Version de la plaque signalétique du moteur.
Gc_byMotorTypePlateRevision	BYTE	29	Révision de la plaque signalétique du moteur.
Gc_wMotorTypePlateSubrevision	WORD	1	Sous-révision de la plaque signalétique du moteur.
Gc_uiEncoderMaxSpeedMinValue	UDINT	0	Valeur minimale du paramètre d'entrée EncoderMinSpeed.
Gc_uiEncoderMaxSpeedMaxValue	UDINT	65534	Valeur maximale du paramètre d'entrée EncoderMaxSpeed.
Gc_uiEncoderMaxTempMinValue	UDINT	0	Valeur minimale du paramètre d'entrée EncoderMinTemp.
Gc_uiEncoderMaxTempMaxValue	UDINT	65534	Valeur maximale du paramètre d'entrée EncoderMinTemp.
Gc_uiEncoderTempSensorMinValue	UDINT	0	Valeur minimale du paramètre d'entrée EncoderTempSensor.
Gc_uiEncoderTempSensorMaxValue	UDINT	1	Valeur maximale du paramètre d'entrée EncoderTempSensor.

Variable	Type de données	Valeur	Description
Gc_uiEncoderNumberOfTurnsMinValue	UDINT	0	Valeur minimale du paramètre d'entrée EncoderNumberOfTurns.
Gc_uiEncoderNumberOfTurnsMaxValue	UDINT	65534	Valeur maximale du paramètre d'entrée EncoderNumberOfTurns.
Gc_uiEncoderLinesPerRevolutionMinValue	UDINT	0	Valeur minimale du paramètre d'entrée EncoderLinesPerRevolution.
Gc_uiEncoderLinesPerRevolutionMaxValue	UDINT	65534	Valeur maximale du paramètre d'entrée EncoderLinesPerRevolution.



B

bloc fonction

Unité de programmation qui possède 1 ou plusieurs entrées et renvoie 1 ou plusieurs sorties. Les blocs fonction (FBs) sont appelés via une instance (copie du bloc fonction avec un nom et des variables dédiés) et chaque instance a un état persistant (sorties et variables internes) d'un appel au suivant.

Exemples : temporisateurs, compteurs

BSH

Servomoteur Lexium de Schneider Electric.

C

codeur

Equipement de mesure de longueur ou d'angle (codeurs linéaires ou rotatifs).

contrôleur

Automatise des processus industriels. On parle également de Logic Controller programmable (PLC) ou de contrôleur programmable.

F

firmware

Représente le BIOS, les paramètres de données et les instructions de programmation qui constituent le système d'exploitation d'un contrôleur. Le firmware est stocké dans la mémoire non volatile du contrôleur.

O

octet

Type codé sur 8 bits, de 00 à FF au format hexadécimal.

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.

S

Sercos

(*serial real-time communications system*) Bus de contrôle numérique qui interconnecte des contrôles de mouvement, des variateurs de vitesse, des E/S, des capteurs et des actionneurs pour des machines et des systèmes commandés numériquement. Il s'agit d'une interface standard et ouverte de communication contrôleur-équipement numérique intelligent, conçue pour la transmission série haut débit de données standard en boucle fermée et en temps réel.



B

bibliothèques

UserMotorTypePlate, 20

BrakeInvalid

FC_MotorDataFileCreate, 86

UserMotorTypePlate - ET_DiagExt, 31

C

CheckingMotorDataInDrive

FB_MotorDataDelete, 52

FB_MotorDataWrite, 64

FB_MotorDataWriteBLH, 72

UserMotorTypePlate - ET_DiagExt, 30

concept de diagnostic, 25

ContinuousStallCurrentInvalid

FC_MotorDataFileCreate, 87

UserMotorTypePlate - ET_DiagExt, 31

ContinuousStallTorqueInvalid

FC_MotorDataFileCreate, 87

UserMotorTypePlate - ET_DiagExt, 31

CouldNotCreateFile

FB_MotorDataRead, 57

FC_MotorDataFileCreate, 87

UserMotorTypePlate - ET_DiagExt, 31

D

DeleteMotorDataNotAllowed

FB_MotorDataDelete, 52

UserMotorTypePlate - ET_DiagExt, 30

DeletingMotorData

FB_MotorDataDelete, 52

UserMotorTypePlate - ET_DiagExt, 31

Disabled

FB_InitMachineEncoder, 42

FB_MotorDataDelete, 52

FB_MotorDataRead, 58

FB_MotorDataWrite, 64

FB_MotorDataWriteBLH, 72

FB_MotorSerialNumberWrite, 78

FC_MotorDataFileCreate, 88

FC_MotorDataFileRead, 96

UserMotorTypePlate - ET_DiagExt, 31

Done

FB_InitMachineEncoder, 43

FB_MotorDataDelete, 53

FB_MotorDataRead, 58

FB_MotorDataWrite, 64

FB_MotorDataWriteBLH, 72

FB_MotorSerialNumberWrite, 78

UserMotorTypePlate - ET_DiagExt, 30

DriveInvalid

FB_InitMachineEncoder, 42

UserMotorTypePlate - ET_DiagExt, 32

DriveVirtual

FB_InitMachineEncoder, 42

UserMotorTypePlate - ET_DiagExt, 32

E

EncoderLinesPerRevolutionInvalid

FB_InitMachineEncoder, 43

UserMotorTypePlate - ET_DiagExt, 32

EncoderMaxSpeedInvalid

FB_InitMachineEncoder, 43

UserMotorTypePlate - ET_DiagExt, 32

EncoderMaxTempInvalid

FB_InitMachineEncoder, 44

UserMotorTypePlate - ET_DiagExt, 32

EncoderNumberOfTurnsInvalid

FB_InitMachineEncoder, 44

UserMotorTypePlate - ET_DiagExt, 32

EncoderTempSensorInvalid
 FB_InitMachineEncoder, *44*
 UserMotorTypePlate - ET_DiagExt, *32*
 ET_DiagExt, *30*
 ET_EncoderType, *33*
 ET_MotorType, *34*
 ET_StorageLocation, *35*
 Executing
 FB_InitMachineEncoder, *45*
 FB_MotorSerialNumberWrite, *79*
 UserMotorTypePlate - ET_DiagExt, *31*

F

FB_InitMachineEncoder, *38*
 FB_MotorDataDelete, *49*
 FB_MotorDataRead, *55*
 FB_MotorDataWrite, *60*
 FB_MotorDataWriteBLH, *68*
 FB_MotorSerialNumberWrite, *76*
 FC_EtDiagExtToString, *82*
 FC_MotorDataFileCreate, *83*
 FC_MotorDataFileRead, *95*
 FileAccessNotPossible
 FB_MotorDataWrite, *65*
 FB_MotorDataWriteBLH, *72*
 FC_MotorDataFileRead, *97*
 UserMotorTypePlate - ET_DiagExt, *30*
 FunctionNotSupportedByThisDevice
 FB_InitMachineEncoder, *45*
 FB_MotorDataDelete, *53*
 FB_MotorDataWrite, *65*
 FB_MotorDataWriteBLH, *73*
 FB_MotorSerialNumberWrite, *79*
 UserMotorTypePlate - ET_DiagExt, *30*

G

GCL (global constant list), *116*

I

Init
 UserMotorTypePlate - ET_DiagExt, *30*
 Initializing
 UserMotorTypePlate - ET_DiagExt, *32*
 InputStringTooLarge
 FB_MotorSerialNumberWrite, *79*
 FC_MotorDataFileCreate, *88*
 UserMotorTypePlate - ET_DiagExt, *31*
 InvalidAddress
 UserMotorTypePlate - ET_DiagExt, *32*
 InvalidDatafield
 UserMotorTypePlate - ET_DiagExt, *32*
 InvalidEncoderType
 FB_InitMachineEncoder, *45*
 FC_MotorDataFileCreate, *88*
 UserMotorTypePlate - ET_DiagExt, *31*
 InvalidMotorName
 FC_MotorDataFileCreate, *89*
 UserMotorTypePlate - ET_DiagExt, *31*
 InvalidNumberOfPolePairs
 FC_MotorDataFileCreate, *89*
 UserMotorTypePlate - ET_DiagExt, *31*
 InvalidStorageLocation
 FB_InitMachineEncoder, *45*
 FB_MotorDataDelete, *53*
 FB_MotorDataWrite, *65*
 FB_MotorDataWriteBLH, *73*
 UserMotorTypePlate - ET_DiagExt, *31*

M

MandatoryParameterInvalid
 FC_MotorDataFileCreate, *89*
 UserMotorTypePlate - ET_DiagExt, *31*
 MaxSpeedInvalid
 FC_MotorDataFileCreate, *89*
 UserMotorTypePlate - ET_DiagExt, *31*
 MEnclnObjectNotAppendedToDrive
 FB_InitMachineEncoder, *46*
 MotorCosPhiInvalid
 FC_MotorDataFileCreate, *90*
 UserMotorTypePlate - ET_DiagExt, *31*

MotorDataAlreadyStoredInDrive
 FB_MotorDataWrite, *66*
 FB_MotorDataWriteBLH, *73*
 UserMotorTypePlate - ET_DiagExt, *30*

MotorDataFileNotFound
 FB_MotorDataWrite, *66*
 FC_MotorDataFileRead, *97*
 UserMotorTypePlate - ET_DiagExt, *30*

MotorInertiaInvalid
 FC_MotorDataFileCreate, *90*
 UserMotorTypePlate - ET_DiagExt, *31*

MotorTypeNotSupported
 FC_MotorDataFileCreate, *90*
 UserMotorTypePlate - ET_DiagExt, *31*

N

NominalCurrentInvalid
 FC_MotorDataFileCreate, *91*
 UserMotorTypePlate - ET_DiagExt, *31*

NominalFrequencyInvalid
 FC_MotorDataFileCreate, *91*
 UserMotorTypePlate - ET_DiagExt, *31*

NominalPowerInvalid
 FC_MotorDataFileCreate, *91*
 UserMotorTypePlate - ET_DiagExt, *32*

NominalSpeedInvalid
 FC_MotorDataFileCreate, *92*
 UserMotorTypePlate - ET_DiagExt, *31*

NominalVoltageInvalid
 FC_MotorDataFileCreate, *92*
 UserMotorTypePlate - ET_DiagExt, *31*

NoMotorDataFoundInDrive
 FB_MotorDataDelete, *54*
 UserMotorTypePlate - ET_DiagExt, *30*

NoValidMachineEncoderConnected
 FB_InitMachineEncoder, *46*

O

Ok
 UserMotorTypePlate - ET_DiagExt, *30*

OperationNotAllowed
 FB_MotorSerialNumberWrite, *79*
 UserMotorTypePlate - ET_DiagExt, *31*

P

ParameterMotorIdentificationWrongValue
 UserMotorTypePlate - ET_DiagExt, *30*

PeakCurrentInvalid
 FC_MotorDataFileCreate, *92*
 UserMotorTypePlate - ET_DiagExt, *31*

PeakTorqueInvalid
 FC_MotorDataFileCreate, *93*
 UserMotorTypePlate - ET_DiagExt, *31*

PhaseResistanceInvalid
 FC_MotorDataFileCreate, *93*
 UserMotorTypePlate - ET_DiagExt, *31*

Prepare
 FB_InitMachineEncoder, *46*
 FB_MotorDataWrite, *66*
 FB_MotorDataWriteBLH, *74*
 UserMotorTypePlate - ET_DiagExt, *30*

Q

QuadraturePhaseInductanceInvalid
 FC_MotorDataFileCreate, *93*
 UserMotorTypePlate - ET_DiagExt, *31*

R

ReadingMotorData
 FB_MotorDataRead, *58*
 UserMotorTypePlate - ET_DiagExt, *31*

ReinitializationFailed
 FB_InitMachineEncoder, *47*
 FB_MotorDataWrite, *66*
 FB_MotorDataWriteBLH, *74*
 UserMotorTypePlate - ET_DiagExt, *31*

ResolutionFineInvalid
 UserMotorTypePlate - ET_DiagExt, *31*

RotatingFieldDirectionInvalid
 FC_MotorDataFileCreate, *94*
 UserMotorTypePlate - ET_DiagExt, *31*

S

SercosCommunicationNotPossible
 FB_InitMachineEncoder, *47*
 FB_MotorDataDelete, *54*
 FB_MotorDataRead, *58*
 FB_MotorDataWrite, *67*
 FB_MotorDataWriteBLH, *74*
 FB_MotorSerialNumberWrite, *80*
 UserMotorTypePlate - ET_DiagExt, *30*

SercosNotInPhaseTwo
 FB_InitMachineEncoder, *47*
 UserMotorTypePlate - ET_DiagExt, *32*

SingleturnResolutionInvalid
 FC_MotorDataFileCreate, *94*
 UserMotorTypePlate - ET_DiagExt, *31*

ST_UserMachineEncoderData, *100*
ST_UserMotorData, *102*
ST_UserMotorDataACIM, *103*
ST_UserMotorDataPMSM, *108*

U

UserMotorTypePlate, *20*
 ET_DiagExt, *30*
 ET_EncoderType, *33*
 ET_MotorType, *34*
 ET_StorageLocation, *35*
 FB_MotorDataDelete, *49*
 FB_MotorDataRead, *55*
 FB_MotorDataWrite, *60*
 FB_MotorDataWriteBLH, *68*
 FB_MotorSerialNumberWrite, *76*
 FC_EtDiagExtToString, *82*
 FC_MotorDataFileCreate, *83*
 FC_MotorDataFileRead, *95*
 GCL (global constant list), *116*
 ST_UserMachineEncoderData, *100*
 ST_UserMotorData, *102*
 ST_UserMotorDataACIM, *103*
 ST_UserMotorDataPMSM, *108*

W

WaitForExecute
 FB_InitMachineEncoder, *47*
 FB_MotorDataDelete, *54*
 FB_MotorDataRead, *59*
 FB_MotorDataWrite, *67*
 FB_MotorDataWriteBLH, *74*
 FB_MotorSerialNumberWrite, *80*
 UserMotorTypePlate - ET_DiagExt, *30*

WaitForSercosPhase2
 FB_MotorDataDelete, *54*
 FB_MotorDataRead, *59*
 FB_MotorDataWrite, *67*
 FB_MotorDataWriteBLH, *75*
 FB_MotorSerialNumberWrite, *80*
 UserMotorTypePlate - ET_DiagExt, *30*