Modicon M262 Logic/Motion Controller

Fonctions et variables système

Guide de la bibliothèque système

Traduction de la notice originale

EIO0000003668.07
12/2024

www.se.com

Mentions légales

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

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

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

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

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

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

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

Table des matières

Consignes de sécurité 7
À propos de ce manuel 8
M262 PLCSystem 13
Variables système du M262 14
Variables système : définition et utilisation 14
Présentation des variables système 14
Utilisation des variables système 15
Structures *PLC_R* et *PLC_W* 16
PLC_R : Variables système en lecture seule du contrôleur 17
PLC_W : Variables système en lecture/écriture du contrôleur 20
Structures *ETH_R* et *ETH_W* 21
ETH_R : Variables système en lecture seule du port Ethernet 21
ETH_W : Variables système en lecture/écriture des ports Ethernet 24
Fonctions système du M262 25
Fonctions de lecture de M262 25
GetImmediateFastInput : Lire l'entrée d'une E/S experte intégrée 25
GetRtc : Obtenir l'horodateur 26
HasForcedIo : indique si une entrée ou une sortie est forcée 27
IsFirstMastColdCycle : indique si ce Cycle est le premier cycle de démarrage à froid de MAST 28
IsFirstMastCycle : Indique si Cycle est le premier cycle MAST 28
IsFirstMastWarmCycle : Indique si Cycle est le premier cycle de démarrage à chaud MAST 30
GetExternalEventValue : Obtenir la valeur actuelle d'un événement externe 30
Fonctions d'écriture de l'automate M262 31
PhysicalWriteFastOutputs : Écrire la sortie rapide d'une E/S experte intégrée 31
SetRTCDrift : Définir la valeur de compensation sur l'horodateur 32
Fonctions utilisateur de M262 34
FB_GetFreeDiskSpace : Obtient l'espace mémoire disponible de manière asynchrone 34
FB_GetLabel : Obtient le libellé d'un support mémoire 35
FB_GetTotalDiskSpace : Obtient la taille du support mémoire 36
FB_CheckAllowedControllerMacAddr : Vérifier si l'adresse MAC est autorisée par le contrôleur 37
FB_ControlClone : Cloner le contrôleur 38
DataFileCopy : commandes de copie de fichier 39
ExecuteScript : Exécution de commandes de script 41
Types de données de la bibliothèque M262 43
Types de données des variables système *PLC_RW* 43

PLC_R_APPLICATION_ERROR : Codes d'état d'erreur détecté de l'application 43
PLC_R_BOOT_PROJECT_STATUS : Codes d'état du projet de démarrage 45
PLC_R_IO_STATUS : Codes d'état des E/S : 45
PLC_R_SDCARD_STATUS : Codes d'état d'emplacement de carte SD 46
PLC_R_STATUS : Codes d'état du contrôleur 46
PLC_R_STOP_CAUSE : Codes de cause de transition de l'état RUN à un autre état 47
PLC_R_TERMINAL_PORT_STATUS : Codes d'état de connexion du port de programmation 48
PLC_R_TM3_BUS_STATE : Codes d'état du bus TM3 49
PLC_W_COMMAND : Codes de commande de contrôle 49
Types de données des variables système DataFileCopy 49
DataFileCopyError : Codes d'erreur détectée 50
DataFileCopyLocation : Codes d'emplacement 50
Types de données des variables système ExecScript 50
ExecuteScriptError : Codes d'erreur détectée 51
Types de données des variables système *ETH_RW* 51
ETH_R_FRAME_PROTOCOL : Codes du protocole de transmission de trames 51
ETH_R_IPFORWARDING : Transfert IP 52
ETH_R_IP_MODE : Codes de source d'adresse IP 52
ETH_R_ITF_STRUCT : Paramètres de l'interface Ethernet 52
ETH_R_PORT_DUPLEX_STATUS : Codes de mode de transmission 54
ETH_R_PORT_IP_STATUS : Codes d'état du port Ethernet TCP/IP 54
ETH_R_PORT_LINK_STATUS : Codes d'état de Liaison de communication 54
ETH_R_PORT_SPEED : Codes de vitesse de communication des ports Ethernet 55
ETH_R_RUN_IDLE : Codes des états de fonctionnement et d'inactivité Ethernet/IP 55
Types de données des fonctions système 56
IMMEDIATE_ERR_TYPE : Lecture des entrées *GetImmediateFastInput* des codes d'E/S expertes intégrées 56
RTCSETDRIFT_ERROR : Codes des erreurs détectées par la fonction *SetRTCDrift* 56
Système SerialLine 57
Variables système ligne série M262 58
SERIAL_R : Variables de diagnostic de ligne série 58
SERIAL_W : Variables de diagnostic de ligne série 59
Système TM3 60
Variables système du TM3 61
TM3_MODULE_R[0...13] : Variables système en lecture seule des modules TM3 61
Fonctions système du TM3 62
TM3_GetModuleBusStatus : Obtenir l'état du bus du module TM3 62

TM3_GetModuleInternalStatus : Obtenir l'état interne du module TM3 63
Types de données système du TM3 66
TM3_BUS_PARAM_ID : Diagnostic du module d'extension TM3 66
TM3_BUS_W_IOBUSERRMOD : Mode d'erreur du bus TM3 66
TM3_BUS_W_IOBUSINIT : Erreur de la réponse de configuration. 67
TM3_BUS_W : Variables système de bus TM3 67
TM3_ERR_CODE : Codes d'erreur détectés du module d'extension TM3 67
TM3_MODULE_R_ARRAY_TYPE : Type de tableau de lecture du module d'extension TM3 68
TM3_MODULE_STATE : Codes d'état du module d'extension TM3 68
Système TMS 70
Variables système du TMS 71
TMS_BUS_DIAG_R : Codes d'erreur de diagnostic du bus TMS 71
TMS_MODULE_DIAG_R : Codes d'erreur de diagnostic des modules d'extension TMS 71
Types de données système du TMS 73
TMS_IP_STATE : État IP des modules d'extension TMS 73
TMS_MODULE_STATE : Codes d'état des modules d'extension TMS 73
TMS_PIXCMD_STATE : État PIXCMD du module d'extension TMS 74
Annexes 76
Représentation des fonctions et blocs fonction 77
Différences entre fonction et bloc fonction 77
Utilisation d'une fonction ou d'un bloc fonction en langage IL 78
Utilisation d'une fonction ou d'un bloc fonction en langage ST 81
Glossaire 83
Index 90

Consignes de sécurité

Informations importantes

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

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

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

⚠ DANGER

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

⚠ AVERTISSEMENT

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

⚠ ATTENTION

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

AVIS

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

Remarque Importante

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

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

À propos de ce manuel

Objet du document

Ce document est destiné à vous familiariser à l'utilisation des fonctions et variables système disponibles dans le Modicon M262 Logic/Motion Controller. La bibliothèque système M262 contient des fonctions et des variables permettant d'obtenir des informations du système du contrôleur et de lui envoyer des commandes.

Ce document décrit les fonctions et variables des types de données des bibliothèques système suivantes du M262 :

- M262 PLCSystem
- Système ligne série
- Système TM3
- Système TMS

Les connaissances préalables suivantes sont requises :

- Connaissances de base sur les fonctionnalités, la structure et la configuration du M262 Logic/Motion Controller
- Programmation en langages FBD, LD, ST, IL ou CFC
- Variables système (variables globales)

Champ d'application

Pour plus d’informations sur la validité du présent document, consultez l’aide en ligne du produit.

Langues disponibles

Ce document est disponible dans les langues suivantes :

- Anglais (EIO0000003667)
- Français (EIO0000003668)
- Allemand (EIO0000003669)
- Espagnol (EIO0000003670)
- Italien (EIO0000003671)
- Chinois (EIO0000003672)

Documents à consulter

Titre du document	Numéro de référence
EcoStruxure Automation Expert - Motion, EcoStruxure Machine Expert – Guide de programmation	EIO0000002854 (ENG) EIO0000002855 (FRE) EIO0000002856 (GER) EIO0000002858 (SPA) EIO0000002857 (ITA) EIO0000002859 (CHS)
Modicon M262 Logic/Motion Controller – Guide de référence du matériel	EIO0000003659 (ENG) EIO0000003660 (FRE) EIO0000003661 (GER) EIO0000003662 (SPA) EIO0000003663 (ITA) EIO0000003664 (CHS)
Modicon M262 Logic/Motion Controller - Guide de programmation	EIO0000003651 (ENG) EIO0000003652 (FRA) EIO0000003653 (GER) EIO0000003654 (SPA) EIO0000003655 (ITA) EIO0000003656 (CHS)

Pour rechercher des documents en ligne, visitez le centre de téléchargement Schneider Electric (www.se.com/ww/en/download/).

Informations produit

⚠AVERTISSEMENT

PERTE DE CONTROLE

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

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

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

⚠AVERTISSEMENT

FONCTIONNEMENT IMPRÉVU DE L'ÉQUIPEMENT

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

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

Informations concernant la terminologie inclusive/sensible

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

Terminologie utilisée dans les normes

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

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

Ces normes incluent entre autres les éléments suivants :

Norme	Description
IEC 61131-2:2007	Automates programmables, partie 2 : Spécifications et essais des équipements.
ISO 13849-1:2023	Sécurité des machines : Composants liés à la sécurité dans les systèmes de commande. Principes généraux de conception
EN 61496-1:2020	Sécurité des machines : Equipement de protection électrosensible. Partie 1 : Exigences générales et tests.
ISO 12100:2010	Sécurité des machines - Principes généraux de conception - Appréciation du risque et réduction du risque
EN 60204-1:2006	Sécurité des machines - Equipement électrique des machines - Partie 1 : exigences générales
ISO 14119:2013	Sécurité des machines - Dispositifs de verrouillage associés à des protecteurs - Principes de conception et de choix
ISO 13850:2015	Sécurité des machines - Fonction d'arrêt d'urgence - Principes de conception
IEC 62061:2021	Sécurité des machines - Sécurité fonctionnelle des systèmes de commande électrique, électronique et électronique programmables relatifs à la sécurité
IEC 61508-1:2010	Sécurité fonctionnelle des systèmes électriques, électroniques et électroniques programmables liés à la sécurité : Exigences générales.
IEC 61508-2:2010	Sécurité fonctionnelle des systèmes électriques, électroniques et électroniques programmables liés à la sécurité : Exigences concernant la sécurité fonctionnelle des systèmes électriques, électroniques et électroniques programmables liés à la sécurité.
IEC 61508-3:2010	Sécurité fonctionnelle des systèmes électriques, électroniques et électroniques programmables liés à la sécurité : Configuration logicielle requise.
IEC 61784-3:2021	Réseaux de communication industriels - Profils - Partie 3 : Bus de terrain liés à la sécurité fonctionnelle - Règles générales et définitions de profil.
2006/42/EC	Directive Machines
2014/30/EU	Directive sur la compatibilité électromagnétique
2014/35/EU	Directive sur les basses tensions

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

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

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

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

M262 PLCSystem

Contenu de cette partie

Variables système du M262 14
Fonctions système du M262 25
Types de données de la bibliothèque M262 43

Introduction

Cette section décrit la bibliothèque M262 PLCSystem.

Variables système du M262

Contenu de ce chapitre

Variables système : définition et utilisation 14
Structures *PLC_R* et *PLC_W* 16
Structures *ETH_R* et *ETH_W* 21

Présentation

Ce chapitre :

- fournit une introduction aux variables système, page 14 ;
- décrit les variables système, page 17 disponibles avec la bibliothèque PLCSystem de M262.

Variables système : définition et utilisation

Présentation

Cette section définit les variables système et explique leur mise en œuvre dans le Modicon M262 Logic/Motion Controller.

Présentation des variables système

Introduction

Cette section décrit comment les variables système sont mises en œuvre. Les variables système :

- permettent d'accéder à des informations générales sur le système, de réaliser des diagnostics système et de commander des actions simples ;
- sont des variables structurées conformes aux définitions et conventions de nom de la norme CEI 61131-3. Vous pouvez accéder aux variables système à l'aide du nom symbolique CEI *PLC_GVL*. Certaines variables *PLC_GVL* sont en lecture seule (par exemple, *PLC_R*) et d'autres sont en lecture-écriture (par exemple, *PLC_W*).
- sont déclarées automatiquement comme des variables globales. Elles s'appliquent à l'ensemble du système et toute POU (unité organisationnelle de programme) d'une tâche peut y accéder.

Convention de désignation

Les variables système sont identifiées par :

- un nom de structure qui représente la catégorie de variables système. Par exemple, *PLC_R* représente un nom de structure de variables en lecture seule utilisées pour le diagnostic du contrôleur.
- un ensemble de noms de composant qui identifie le rôle de la variable. Par exemple, *i_wVendorID* représente l'ID du fournisseur du contrôleur.

Vous pouvez accéder aux variables système en entrant leur nom de structure suivi du nom du composant.

Voici un exemple de mise en œuvre de variables système :

```
VAR
myCtr_Serial : DWORD;
myCtr_ID : DWORD;
myCtr_FramesRx : UDINT;
END_VAR
myCtr_Serial := PLC_GVL.PLC_R.i_dwSerialNumber;
myCtr_ID := PLC_GVL.PLC.R.i_wVendorID;
myCtr_FramesRx := SERIAL_R[0].i_udiFramesReceivedOK
```

NOTE: Le nom complet de la variable système dans l'exemple ci-dessus est *PLC_GVL.PLC_R*. *PLC_GVL* est implicite lors de la déclaration d'une variable avec l'assistant **Aide à la saisie**, mais vous pouvez également le saisir avec le préfixe. Les bonnes pratiques de programmation préconisent souvent d'utiliser le nom complet de la variable dans les déclarations.

Emplacement des variables système

Deux sortes de variables système sont définies pour la programmation du contrôleur :

- variables localisées
- variables non localisées

Les variables localisées :

- sont accessibles par l'intermédiaire de requêtes Modbus TCP, Modbus série et EtherNet/IP dans les états RUNNING et STOPPED ;
- sont utilisées dans des programmes EcoStruxure Automation Expert - Motion et EcoStruxure Machine Expert conformément à la convention *structure_name.component_name* expliquée précédemment.

  Les adresses %MW de 0 à 59999 sont accessibles directement. Les adresses supérieures sont considérées hors plage par EcoStruxure Automation Expert - Motion et EcoStruxure Machine Expert et sont uniquement accessibles via la convention *structure_name.component_name*.

Les variables non localisées :

- ne se trouvent pas physiquement dans la zone %MW.
- ne sont pas accessibles par le biais de requêtes de bus de terrain ou de réseau, sauf si vous les cherchez dans la table de réaffectation, et elles ne sont alors accessibles que dans les états RUNNING et STOPPED ; La table de réaffectation utilise les zones %MW dynamiques suivantes :
  - %MW60200 à %MW61999 pour les variables système en lecture seule,
  - %MW62200 à %MW63999 pour les variables en lecture/écriture.
- sont utilisées dans des programmes EcoStruxure Automation Expert - Motion et EcoStruxure Machine Expert conformément à la convention *structure_name.component_name* expliquée précédemment.

Utilisation des variables système

Introduction

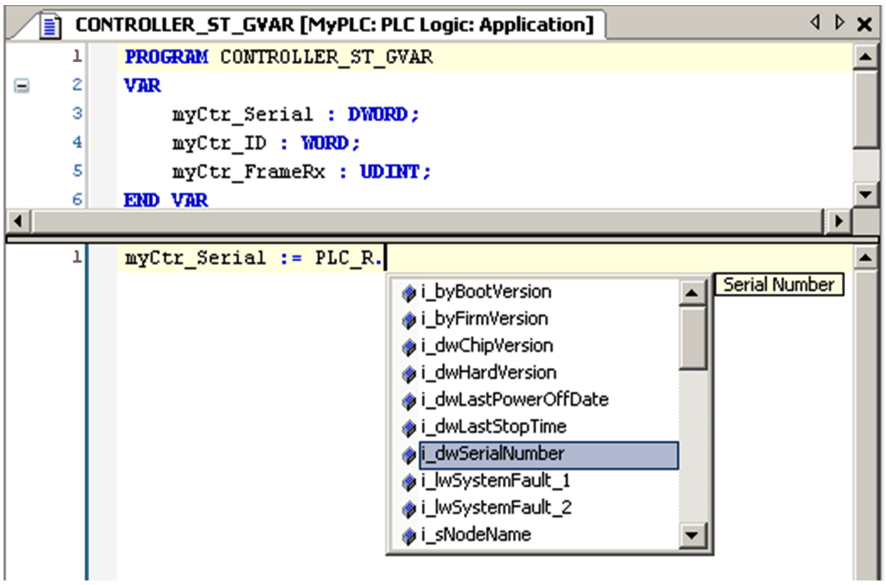
Cette section décrit la procédure de programmation et d'utilisation des variables système dans EcoStruxure Automation Expert - Motion et EcoStruxure Machine Expert.

Les variables système ont un champ d'application global et vous pouvez les utiliser dans tous les POU (unité organisationnelle de programme) de l'application.

Il n'est pas nécessaire de déclarer les variables système dans la liste des variables globales (GVL). Elles sont déclarées automatiquement à partir de la bibliothèque système du contrôleur.

Utilisation des variables système dans une POU

EcoStruxure Automation Expert - Motion et EcoStruxure Machine Expert disposent d'une fonction de saisie semi-automatique. Dans une **POU**, commencez par entrer le nom de structure de la variable système (*PLC_R PLC_W*...) suivi d'un point. Les variables système s'affichent dans l'**Aide à la saisie**. Vous pouvez sélectionner la variable souhaitée ou entrer le nom complet manuellement.



NOTE: Dans l'exemple ci-dessus, une fois que le nom de structure *PLC_R* a été entré, EcoStruxure Automation Expert - Motion et EcoStruxure Machine Expert affichent un menu contextuel des noms de composants/variables possibles.

Exemple

L'exemple suivant décrit l'utilisation de certaines variables système :

```
VAR
myCtr_Serial : DWORD;
myCtr_ID : WORD;
myCtr_FramesRx : UDINT;
END_VAR
myCtr_Serial := PLC_R.i_dwSerialNumber;
myCtr_ID := PLC_R.i_wVendorID;
myCtr_FramesRx := SERIAL_R[0].i_udiFramesReceivedOK;
```

Structures *PLC_R* et *PLC_W*

Présentation

Cette section répertorie et décrit les différentes variables système *SEC.PLC_GVL.PLC_R* et *SEC.PLC_GVL.PLC_W* de la bibliothèque SE_PLCSystem utilisant l'espace de noms SEC.

Les structures des variables sont définies dans la bibliothèque PLCSystemBase.

PLC_R : Variables système en lecture seule du contrôleur

Bibliothèque et espace de noms

Nom de la bibliothèque : **SE_PLCSystem**

Espace de noms : **SEC**

Structure de la variable

Ce tableau décrit les paramètres des variables système *PLC_R* (type *PLC_R_STRUCT*) :

Adresse Modbus (1)	Nom de la variable	Type	Commentaire
60000	*i_wVendorID*	WORD	ID du fournisseur du contrôleur. 101A hex = Schneider Electric
60001	*i_wProductID*	WORD	ID de référence du contrôleur. **NOTE:** L'ID du fournisseur et l'ID de référence constituent l'ID cible du contrôleur, indiqué dans l'écran des paramètres de communication (ID cible = 101A XXXX hex). Valeur par contrôleur : • TM262L01MESE8T : 101A 0816 HEX • TM262L10MESE8T : 101A 0813 HEX • TM262L20MESE8T : 101A 0810 HEX • TM262M05MESS8T : 101A 0815 HEX • TM262M15MESS8T : 101A 0814 HEX • TM262M25MESS8T : 101A 0811 HEX • TM262M35MESS8T : 101A 0812 HEX
60002	*i_dwSerialNumber*	DWORD	Numéro de série du contrôleur.
60004	*i_byFirmVersion*	ARRAY[0..3] OF BYTE	Version du micrologiciel du contrôleur `[aa.bb.cc.dd]` : • i_byFirmVersion[0] = aa • ... • i_byFirmVersion[3] = dd
60006	*i_byBootVersion*	ARRAY[0..3] OF BYTE	Version de démarrage du contrôleur `[aa.bb.cc.dd]` : • i_byBootVersion[0] = aa • ... • i_byBootVersion[3] = dd
60008	*i_dwHardVersion*	DWORD	Version du matériel du contrôleur. **NOTE:** Paramètre réservé exclusivement à un usage interne. Pour la version du produit (PV), consultez l'étiquette du produit.
60010	*i_dwChipVersion*	DWORD	Version du coprocesseur du contrôleur.
60012	*i_wStatus*	*PLC_R_STATUS*, page 46	Etat du contrôleur.
60013	*i_wBootProjectStatus*	*PLC_R_BOOT_PROJECT_STATUS*, page 45	Renvoie des informations sur l'application de démarrage stockée en mémoire non volatile.
60014	*i_wLastStopCause*	*PLC_R_STOP_CAUSE*, page 47	Cause du dernier passage du mode d'exécution (RUN) à un autre état.
60015	*i_wLastApplicationError*	*PLC_R_APPLICATION_ERROR*, page 43	Cause de la dernière exception du contrôleur.

Adresse Modbus (1)	Nom de la variable	Type	Commentaire
60016	*i_lwSystemFault_1*	LWORD	Le champ de bits FFFF FFFF FFFF FFFF hex indique qu'aucune erreur n'a été détectée. Un bit de niveau bas (0) signifie qu'une erreur a été détectée : • bit 0 = erreur d'E/S experte détectée • bit 1 = erreur de bus d'E/S détectée • bit 2 = erreur IF1 Ethernet détectée • bit 3 = erreur IF2 Ethernet détectée • bit 4 = erreur de surintensité détectée sur ligne série 1 • bit 5 = erreur de surintensité détectée sur ligne série 2 • bit 6 = erreur CAN 1 détectée • bit 7 = réservé • bit 8 = réservé • bit 9 = erreur de bus de communication détectée • bit 10 = erreur de carte SD détectée • bit 11 = erreur de pare-feu détectée • bit 12 = erreur de serveur DHCP/FDR détectée • Bit 13 = erreur de serveur OPC UA détectée • bit 14 = erreur de bus de communication détectée • bit 15 = erreur d'intégrité du bus de communication détectée • bit 16 = erreur de racine d'anneau Ethernet IF2 détectée • bit 17 = erreur d'alimentation de codeur détectée • bit 18 = erreur de communication du codeur détectée • bit 19 = TMSES4 - erreur de configuration IF1 configuration détectée • bit 20 = TMSES4 - erreur de configuration IF2 configuration détectée • bit 21 = TMSES4 - erreur de configuration IF3 configuration détectée • bit 22 = erreur d'adresse serveur détectée • bit 23 = erreur NTP détectée • bit 24 = erreur Syslog détectée • bit 25 = erreur détectée sur le serveur NTP primaire • bit 26 = erreur détectée sur le serveur NTP secondaire **NOTE:** Les bits 25 et 26 s'appliquent uniquement lorsque le contrôleur est configuré en tant que client NTP avec les serveurs respectifs.
60020	*i_lwSystemFault_2*	LWORD	Le champ de bits FFFF hex indique qu'aucune erreur n'a été détectée. Si *i_wIOStatus1* = *PLC_R_IO_SHORTCUT_FAULT*, le sens de *i_lwSystemFault_2* est : • Bit 0 = 0 : court-circuit détecté dans le groupe de sorties 0 (Q0 à Q1) • Bit 1 = 0 : court-circuit détecté dans le groupe de sorties 1 (Q2 à Q3) • Bit 2 = 0 : court-circuit détecté dans le groupe de sorties 2 (Q4 à Q7) • Bit 3 = 0 : court-circuit détecté dans le groupe de sorties 3 (Q8 à Q11) • Bit 4 = 0 : court-circuit détecté dans le groupe de sorties 4 (Q12 à Q15)
60024	*i_wIOStatus1*	*PLC_R_IO_STATUS*, page 45	Etat des E/S expertes intégrées.
60025	*i_wIOStatus2*	*PLC_R_IO_STATUS*, page 45	État d'E/S TM3.

Adresse Modbus (1)	Nom de la variable	Type	Commentaire
60026	*i_wClockBatterystatus*	WORD	Etat de la batterie de l'horodateur : 100 = batterie en pleine charge Non applicable au M262 Logic/Motion Controller.
60028	*i_dwAppliSignature1*	DWORD	Premier des 4 DWORD de la signature (16 octets au total). La signature de l'application est générée par le logiciel pendant la construction.
60030	*i_dwAppliSignature2*	DWORD	Deuxième des 4 DWORD de la signature (16 octets au total). La signature de l'application est générée par le logiciel pendant la construction.
60032	*i_dwAppliSignature3*	DWORD	Troisième des 4 DWORD de la signature (16 octets au total). La signature de l'application est créée par le logiciel pendant la génération.
60034	*i_dwAppliSignature4*	DWORD	Quatrième des 4 DWORD de la signature (16 octets au total). La signature de l'application est générée par le logiciel pendant la construction.
s/o	*i_sVendorName*	STRING(31)	Nom du fournisseur : "Schneider Electric".
s/o	*i_sProductRef*	STRING(31)	Référence du contrôleur.
s/o	*i_sNodeName*	STRING(99)	Nom du nœud sur le réseau logiciel
s/o	*i_dwLastStopTime*	DWORD	Heure du dernier arrêt détecté, en secondes depuis le 1er janvier 1970 à 00:00:00 (UTC).
s/o	*i_dwLastPowerOffDate*	DWORD	Date et heure de la dernière mise hors tension détectée, en secondes depuis le 1er janvier 1970 à 00:00:00 (UTC). **NOTE:** Convertissez cette valeur en date et heure avec la fonction *SysTimeRtcConvertUtcToDate*. Pour plus d'informations sur la conversion de date et d'heure, reportez-vous au Guide de la bibliothèque Systime (voir Affichage et réglage de l'horodateur – Guide des bibliothèques SysTimeRtc et SysTimeCore.
s/o	*i_uiEventsCounter*	UINT	Nombre d'événements externes détectés sur des entrées configurées pour la détection d'événements externes depuis le dernier démarrage à froid. Effectuez la réinitialisation par un démarrage à froid ou en exécutant la commande *PLC_W.q_wResetCounterEvent*.
s/o	*i_wTerminalPortStatus*	*PLC_R_TERMINAL_PORT_STATUS*, page 48	Etat du port de programmation USB (USB mini B).
s/o	*i_wSdCardStatus*	*PLC_R_SDCARD_STATUS*, page 46	Etat de la carte SD.
s/o	*i_wUsrFreeFileHdl*	WORD	Nombre de descripteurs de fichier disponibles, y compris les sockets et les canaux au niveau du système. Un descripteur de fichier correspond à la ressource allouée par le système lorsque vous ouvrez un fichier. **NOTE:** Cette valeur n'indique pas le nombre de descripteurs de fichier disponibles pour l'application.
s/o	*i_udiUsrFsTotalBytes*	UDINT	Taille de la mémoire totale du système de fichiers de l'utilisateur (en octets). Il s'agit de la taille de la mémoire non volatile du répertoire "/usr/".
s/o	*i_udiUsrFsFreeBytes*	UDINT	Taille de la mémoire libre du système de fichiers de l'utilisateur (en octets).

Adresse Modbus (1)	Nom de la variable	Type	Commentaire
s/o	*i_uiTM3BusState*	PLC_R_TM3_BUS_STATE, page 49	État du bus TM3. *i_uiTM3BusState* peut avoir les valeurs suivantes : • 1 : TM3_CONF_ERROR La configuration physique et la configuration logicielle ne correspondent pas. • 3 : TM3_OK La configuration physique correspond à la configuration logicielle. • 4 : TM3_POWER_SUPPLY_ERROR Le bus TM3 n'est pas alimenté (par exemple, lorsque le contrôleur est alimenté par USB).
s/o	*i_ExpertIO_RunStop_Input*	BYTE	L'emplacement de l'entrée Run/Stop est : • 16 à FF hex si l'E/S experte n'est pas configurée • 0 pour `%IX0.0` • 1 pour `%IX0.1` • 2 pour `%IX0.2` • ...etc.
s/o	*i_x10msClk*	BOOL	Bit de base de temps : 10 ms. Cette variable s'active et se désactive par période de 10 ms. La valeur bascule lorsque le contrôleur est dans l'état STOPPED et dans l'état RUN.
s/o	*i_x100msClk*	BOOL	Bit de base de temps : 100 ms. Cette variable s'active et se désactive par période de 100 ms. La valeur bascule lorsque le contrôleur est dans l'état STOPPED et dans l'état RUN.
s/o	*i_x1sClk*	BOOL	Bit de base de temps : 1 s. Cette variable s'active et se désactive par période = 1 ms. La valeur bascule lorsque le contrôleur est dans l'état STOPPED et dans l'état RUN.
(1) Non accessible via l'application en tant que %MW. **s/o** signifie qu'aucun mappage d'adresse Modbus n'est prédéfini pour cette variable système.			

PLC_W : Variables système en lecture/écriture du contrôleur

Bibliothèque et espace de noms

Nom de la bibliothèque : **SE_PLCSystem**

Espace de noms : **SEC**

Structure de la variable

Ce tableau décrit les paramètres des variables système *PLC_W* (type *PLC_W_STRUCT*) :

%MW	Nom de la variable	Type	Commentaire
s/o	*q_wResetCounterEvent*	WORD	Le passage de 0 à 1 réinitialise le compteur d'événements (*PLC_R.i_uiEventsCounter*). Pour réinitialiser à nouveau le compteur, il est nécessaire d'écrire 0 dans cette variable afin permettre une autre transition de 0 à 1.
s/o	*q_uiOpenPLCControl*	UINT	Lorsque la valeur de la variable passe de 0 à 6699, la commande précédemment écrite dans le paramètre *PLC_W.q_wPLCControl* suivant est exécutée.
s/o	*q_wPLCControl*	*PLC_W_COMMAND*, page 49	Commande RUN/STOP du contrôleur exécutée lorsque la valeur de la variable système *PLC_W.q_uiOpenPLCControl* passe de 0 à 6699.
s/o signifie qu'aucune affectation %MW n'est prédéfinie pour cette variable système.			

Structures *ETH_R* et *ETH_W*

Présentation

Cette section répertorie et décrit les différentes variables système *SEC.PLC_GVL.ETH_R* et *SEC.PLC_GVL.ETH_W* de la bibliothèque SE_PLCSystem utilisant l'espace de noms SEC.

Les structures des variables sont définies dans la bibliothèque PLCSystemBase.

ETH_R : Variables système en lecture seule du port Ethernet

Bibliothèque et espace de noms

Nom de la bibliothèque : **SE_PLCSystem**

Espace de noms : **SEC**

Structure de la variable

Ce tableau décrit les paramètres des variables système *ETH_R* (type *ETH_R_STRUCT*). Il existe une structure pour chaque port Ethernet :

%MW	Nom de la variable	Type	Commentaire
60050	*i_byIPAddress*	ARRAY[0..3] OF BYTE	Adresse IP de l'interface Ethernet_1 ou Ethernet_2 `[aaa.bbb.ccc.ddd]` : • i_byIPAddress[0] = aaa • ... • i_byIPAddress[3]= ddd
60052	*i_bySubNetMask*	ARRAY[0..3] OF BYTE	Masque de sous-réseau de l'interface Ethernet_1 ou Ethernet_2 `[aaa.bbb.ccc.ddd]` : • i_bySub-netMask[0] = aaa • ... • i_bySub-netMask[3] = ddd

%MW	Nom de la variable	Type	Commentaire
60054	*i_byGateway*	ARRAY[0..3] OF BYTE	Adresse de passerelle de l'interface Ethernet_1 ou Ethernet_2 `[aaa.bbb.ccc.ddd]` : • i_byGateway[0] = aaa • ... • i_byGateway[3] = ddd
60056	*i_byMACAddress*	ARRAY[0..5] OF BYTE	Adresse MAC de l'interface Ethernet_1 ou Ethernet_2 `[aa.bb.cc.dd.ee.ff]` : • i_byMACAddress[0] = aa • ... • i_byMACAddress[5]= ff
60059	*i_sDeviceName*	STRING(15)	Nom utilisé pour obtenir l'adresse IP auprès du serveur.
s/o	*i_ucIPForwarding*	*ETH_R_IPFROWARDING*, page 52	Transfert IP.
s/o	*i_wIpMode*	*ETH_R_IP_MODE*, page 52	Méthode utilisée pour obtenir une adresse IP.
s/o	*i_byFDRServerIPAddress*	ARRAY[0..3] OF BYTE	Adresse IP `[aaa.bbb.ccc.ddd]` du serveur DHCP ou BootP : • i_byFDRServerIPAddress[0] = aaa • ... • i_byFDRServerIPAddress[3] = ddd Egale à 0.0.0.0 en cas d'utilisation d'une adresse IP enregistrée ou par défaut.
s/o	*i_udiOpenTcpConnections*	UDINT	Nombre de connexions TCP ouvertes.
s/o	*i_udiFramesTransmittedOK*	UDINT	Nombre de trames transmises correctement. Réinitialisation lors de la mise sous tension ou avec la commande de réinitialisation *ETH_W.q_wResetCounter*.
s/o	*i_udiFramedReceivedOK*	UDINT	Nombre de trames reçues correctement. Réinitialisation lors de la mise sous tension ou avec la commande de réinitialisation *ETH_W.q_wResetCounter*.
s/o	*i_udiTransmitBufferErrors*	UDINT	Nombre de trames transmises avec détection d'erreurs. Réinitialisation lors de la mise sous tension ou avec la commande de réinitialisation *ETH_W.q_wResetCounter*.
s/o	*i_udiReceiveBufferErrors*	UDINT	Nombre de trames reçues avec détection d'erreurs. Réinitialisation lors de la mise sous tension ou avec la commande de réinitialisation *ETH_W.q_wResetCounter*.
s/o	*i_wFrameSendingProtocol*	ETH_R_FRAME_PROTOCOL, page 51	Protocole Ethernet configuré pour l'envoi des trames (IEEE 802.3 ou Ethernet II).
s/o	*i_wPortALinkStatus*	ETH_R_PORT_LINK_STATUS, page 54	Liaison du port Ethernet (0 = aucune liaison, 1 = liaison connectée à un autre équipement Ethernet).
s/o	*i_wPortASpeed*	ETH_R_PORT_SPEED, page 55	Vitesse réseau du port Ethernet (10 Mb/s, 100 Mb/s ou 1 Gb/s).
s/o	*i_wPortADuplexStatus*	*ETH_R_PORT_DUPLEX_STATUS*, page 54	Etat duplex du port Ethernet (0 = semi duplex ou 1 = duplex intégral).
s/o	*i_udiPortACollisions*	UDINT	Nombre de trames impliquées dans une ou plusieurs collisions et transmises correctement par la suite. Réinitialisation lors de la mise sous tension ou avec la commande de réinitialisation *ETH_W.q_wResetCounter*.
s/o	*i_wPortAIpStatus*	*ETH_R_PORT_IP_STATUS*, page 54	État de la pile du port TCP/IP Ethernet
s/o	*i_ethInterface*	ARRAY[1..6] OF *ETH_R_ITF_STRUCT*, page 52	Structure des paramètres communs de l'interface Ethernet.
Spécifique à Modbus TCP/IP			

%MW	Nom de la variable	Type	Commentaire
s/o	*i_udiModbusMessageTransmitted*	UDINT	Nombre de messages Modbus transmis. Réinitialisation lors de la mise sous tension ou avec la commande de réinitialisation *ETH_W.q_wResetCounter*.
s/o	*i_udiModbusMessageTransmitted*	UDINT	Nombre de messages Modbus reçus. Réinitialisation lors de la mise sous tension ou avec la commande de réinitialisation *ETH_W.q_wResetCounter*.
s/o	*i_udiModbusErrorMessage*	UDINT	Messages de détection d'erreurs Modbus transmis et reçus. Réinitialisation lors de la mise sous tension ou avec la commande de réinitialisation *ETH_W.q_wResetCounter*.
s/o	*i_byMasterIpTimeouts*	BYTE	Compteur d'événements de dépassement de délai TCP maître Ethernet Modbus. Réinitialisation lors de la mise sous tension ou avec la commande de réinitialisation *ETH_W.q_wResetCounter*.
s/o	*i_byMasterIpLost*	BYTE	État de la liaison du maître Modbus TCP Ethernet : 0 = liaison OK, 1 = liaison perdue.
Spécifique à EtherNet/IP			
s/o	*i_udiETHIP_IOMessagingTransmitted*	UDINT	Trames EtherNet/IP de classe 1 transmises. Réinitialisation lors de la mise sous tension ou avec la commande de réinitialisation *ETH_W.q_wResetCounter*.
s/o	*i_udiETHIP_IOMessagingReceived*	UDINT	Trames EtherNet/IP de classe 1 reçues. Réinitialisation lors de la mise sous tension ou avec la commande de réinitialisation `ETH_W.q_wResetCounter`.
s/o	*i_udiUCMM_Request*	UDINT	Messages EtherNet/IP non connectés reçus. Réinitialisation lors de la mise sous tension ou avec la commande de réinitialisation *ETH_W.q_wResetCounter*.
s/o	*i_udiUCMM_Error*	UDINT	Messages EtherNet/IP non connectés non valides reçus. Réinitialisation lors de la mise sous tension ou avec la commande de réinitialisation *ETH_W.q_wResetCounter*.
s/o	*i_udiClass3_Request*	UDINT	Requêtes EtherNet/IP de classe 3 reçues. Réinitialisation lors de la mise sous tension ou avec la commande de réinitialisation *ETH_W.q_wResetCounter*.
s/o	*i_udiClass3_Error*	UDINT	Requêtes EtherNet/IP de classe 3 non valides reçues. Réinitialisation lors de la mise sous tension ou avec la commande de réinitialisation *ETH_W.q_wResetCounter*.
s/o	*i_uiAssemblyInstanceInput*	UINT	Numéro de l'instance d'assemblage d'entrée. Pour plus d'informations, reportez-vous au guide de programmation du contrôleur.
s/o	*i_uiAssemblyInstanceInputSize*	UINT	Taille de l'instance d'assemblage d'entrée. Pour plus d'informations, consultez le guide de programmation de votre contrôleur.
s/o	*i_uiAssemblyInstanceOutput*	UINT	Numéro de l'instance d'assemblage de sortie. Pour plus d'informations, consultez le guide de programmation de votre contrôleur.
s/o	*i_uiAssemblyInstanceOutputSize*	UINT	Taille de l'instance d'assemblage de sortie. Pour plus d'informations, consultez le guide de programmation de votre contrôleur.
s/o	*i_uiETHIP_ConnectionTimeouts*	UINT	Nombre d'expirations de connexion. Réinitialisation lors de la mise sous tension ou avec la commande de réarmement *ETH_W.q_wResetCounter*.
s/o	*i_ucEipRunIdle*	*ETH_R_RUN_IDLE*, page 55	Drapeau fonctionnement (valeur = 1) / attente (valeur = 0) pour la connexion EtherNet/IP classe 1.
s/o signifie qu'aucun mappage %MW n'est prédéfini pour cette variable système.			

ETH_W : Variables système en lecture/écriture des ports Ethernet

Bibliothèque et espace de noms

Nom de la bibliothèque : **SE_PLCSystem**

Espace de noms : **SEC**

Structure de la variable

Ce tableau décrit les paramètres des variables système *ETH_W* (type *ETH_W_STRUCT*) :

%MW	Nom de la variable	Type	Commentaire
s/o	*q_wResetCounter*	WORD	Le passage de 0 à 1 réinitialise tous les compteurs *ETH_R*. Pour effectuer la réinitialisation à nouveau, il est nécessaire d'écrire 0 dans cette variable pour permettre une autre transition de 0 à 1.
s/o signifie qu'aucun mappage %MW n'est prédéfini pour cette variable système.			

Fonctions système du M262

Contenu de ce chapitre

Fonctions de lecture de M262 25
Fonctions d'écriture de l'automate M262 31
Fonctions utilisateur de M262 34

Présentation générale

Ce chapitre décrit les fonctions système disponibles dans la bibliothèque PLCSystem de M262.

Fonctions de lecture de M262

Présentation

Cette section décrit les fonctions de lecture de la bibliothèque PLCSystem de M262.

GetImmediateFastInput : Lire l'entrée d'une E/S experte intégrée

Description de la fonction

Cette fonction renvoie la valeur de l'entrée, laquelle peut être différente de la valeur logique de cette entrée. La valeur est lue directement sur le matériel au moment de l'appel de la fonction. Seules I0 à I3 sont accessibles via cette fonction.

Bibliothèque et espace de noms

Nom de la bibliothèque : **SE_PLCSystem**

Espace de noms : **SEC**

Représentation graphique

Représentation en langage IL et ST

Pour voir la représentation générale en langage IL ou ST, consultez le chapitre *Représentation des fonctions et blocs fonction*, page 77.

Description des variables d'E/S

Le tableau suivant décrit les variables d'entrée :

Entrée	Type	Commentaire
Block	INT	Non utilisé.
Input	INT	Index d'entrée à lire de 0 à 3.

Le tableau suivant décrit la variable de sortie :

Sortie	Type	Commentaire
GetImmediateFastInput	BOOL	Valeur de l'entrée <*Input*> – FALSE/ TRUE.

Le tableau suivant décrit les paramètres d'entrée/sortie :

Entrée/sortie	Type	Commentaire
Error	BOOL	FALSE = opération correctement exécutée. TRUE = opération en erreur, la fonction renvoie une valeur qui n'est pas valide.
ErrID	*IMMEDIATE_ERR_TYPE*, page 56	Code d'erreur de fonctionnement lorsque *Error* est TRUE.

GetRtc : Obtenir l'horodateur

Description de la fonction

Cette fonction renvoie l'horodatage en secondes au format UNIX (nombre de secondes écoulées depuis le 1/1/1970 à minuit (UTC)).

Bibliothèque et espace de noms

Nom de la bibliothèque : **PLCSystemBase**

Espace de nom : **PLCSystemBase**

Représentation graphique

Représentation en langage IL et ST

Pour voir la représentation générale en langage IL ou ST, consultez le chapitre *Représentation des fonctions et blocs fonction*, page 77.

Description des variables d'E/S

Le tableau suivant décrit la variable d'entrée/sortie :

Sortie	Type	Commentaire
GetRtc	DINT	Horodatage actuel en secondes au format UNIX.

Exemple

L'exemple suivant montre comment obtenir la valeur d'horodatage :

```
VAR
MyRTC : DINT := 0;
END_VAR
MyRTC := GetRtc();
```

HasForcedIo : indique si une entrée ou une sortie est forcée

Description de la fonction

Cette fonction renvoie TRUE si une entrée ou une sortie est forcée.

Représentation graphique

Représentation en langage IL et ST

Pour voir la représentation générale en langage IL ou ST, consultez le chapitre *Représentation des fonctions et blocs fonction*.

Description des variables d'E/S

Ce tableau décrit la variable de sortie :

Sortie	Type	Commentaire
HasForcedIo	BOOL	TRUE si une entrée ou une sortie est forcée.

Exemple

L'exemple suivant décrit comment utiliser cette fonction :

```
VAR
hasIo: BOOL;
END_VAR
hasIo := SEC.HasForcedIo();
```

IsFirstMastColdCycle : indique si ce Cycle est le premier cycle de démarrage à froid de MAST

Description de la fonction

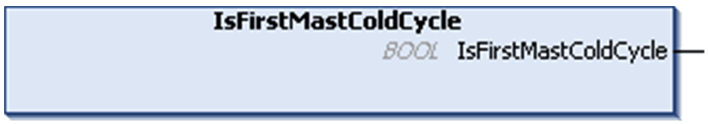
Cette fonction renvoie TRUE au cours du premier cycle MAST après un démarrage à froid (premier cycle après téléchargement ou réinitialisation à froid).

Bibliothèque et espace de noms

Nom de la bibliothèque : **PLCSystemBase**

Espace de nom : **PLCSystemBase**

Représentation graphique



Représentation en langage IL et ST

Pour voir la représentation générale en langage IL ou ST, consultez le chapitre *Représentation des fonctions et blocs fonction*, page 77.

Description des variables d'E/S

Ce tableau décrit la variable de sortie :

Sortie	Type	Commentaire
IsFirstMastColdCycle	BOOL	TRUE au cours du premier cycle de la tâche MAST après un démarrage à froid.

Exemple

Reportez-vous à la fonction *IsFirstMastCycle*, page 28.

IsFirstMastCycle : Indique si Cycle est le premier cycle MAST

Description de la fonction

Cette fonction renvoie TRUE lors du premier cycle MAST après un démarrage.

Bibliothèque et espace de noms

Nom de la bibliothèque : **PLCSystemBase**

Espace de nom : **PLCSystemBase**

Représentation graphique

Représentation en langage IL et ST

Pour voir la représentation générale en langage IL ou ST, consultez le chapitre *Représentation des fonctions et blocs fonction*, page 77.

Description des variables d'E/S

Sortie	Type	Commentaire
IsFirstMastCycle	BOOL	TRUE lors du premier cycle de la tâche MAST après un démarrage.

Exemple

Cet exemple décrit les trois fonctions *IsFirstMastCycle*, *IsFirstMastColdCycle* et *IsFirstMastWarmCycle* utilisés ensemble.

Utilisez cet exemple dans la tâche MAST. Sinon, il peut s'exécuter plusieurs fois ou jamais (une tâche supplémentaire peut être appelée plusieurs fois ou éventuellement aucune fois pendant un cycle de tâche MAST) :

```
VAR
MyIsFirstMastCycle : BOOL;
MyIsFirstMastWarmCycle : BOOL;
MyIsFirstMastColdCycle : BOOL;
END_VAR
MyIsFirstMastWarmCycle := IsFirstMastWarmCycle();
MyIsFirstMastColdCycle := IsFirstMastColdCycle();
MyIsFirstMastCycle := IsFirstMastCycle();
IF (MyIsFirstMastWarmCycle) THEN
(*This is the first Mast Cycle after a Warm Start: all
variables are set to their initialization values except the
Retain variables*)
(*=> initialize the needed variables so that your
application runs as expected in this case*)
END_IF;
IF (MyIsFirstMastColdCycle) THEN
(*This is the first Mast Cycle after a Cold Start: all
variables are set to their initialization values including
the Retain Variables*)
(*=> initialize the needed variables so that your
application runs as expected in this case*)
END_IF;
IF (MyIsFirstMastCycle) THEN
(*This is the first Mast Cycle after a Start, i.e. after a
Warm or Cold Start as well as STOP/RUN commands*)
(*=> initialize the needed variables so that your
application runs as expected in this case*)
END_IF;
```

IsFirstMastWarmCycle : Indique si Cycle est le premier cycle de démarrage à chaud MAST

Description de la fonction

Cette fonction renvoie TRUE lors du premier cycle MAST après un démarrage à chaud.

Bibliothèque et espace de noms

Nom de la bibliothèque : **PLCSystemBase**

Espace de nom : **PLCSystemBase**

Représentation graphique

Représentation en langage IL et ST

Pour voir la représentation générale en langage IL ou ST, consultez le chapitre *Représentation des fonctions et blocs fonction*, page 77.

Description des variables d'E/S

Ce tableau décrit la variable de sortie :

Sortie	Type	Commentaire
IsFirstMastWarmCycle	BOOL	TRUE au cours du premier cycle de la tâche MAST après un démarrage à chaud.

Exemple

Reportez-vous à la fonction *IsFirstMastCycle*, page 28.

GetExternalEventValue : Obtenir la valeur actuelle d'un événement externe

Description de la fonction

Utilisez cette fonction pour obtenir la valeur associée à une tâche d'événement externe.

NOTE: La fonction doit être appelée depuis une tâche d'événement externe.

Représentation graphique

Représentation en langage IL et ST

Pour voir la représentation générale en langage IL ou ST, consultez le chapitre *Représentation des fonctions et blocs fonction*, page 77.

Description des variables d'E/S

Ce tableau décrit les variables d'entrée :

Entrées	Type	Commentaire
pValue	POINTER TO DINT	Adresse de la variable dans laquelle la valeur est copiée si la fonction renvoie *EXTEVT_VAL_OK*.

Le tableau suivant décrit les variables de sortie :

Sorties	Type	Commentaire
GetExternalEventValue	EXTEVT_VAL_RES	Renvoie une des valeurs suivantes : • *EXTEVT_VAL_OK* : valeur valide • *EXTEVT_VAL_FAILED* : impossible d'obtenir la valeur • *EXTEVT_VAL_NOT_IN_EXT_EVT_TASK* : la fonction n'a pas été appelée depuis une tâche d'événement externe • *EXTEVT_VAL_NOT_AVAILABLE* : aucune valeur disponible pour cette tâche externe

Fonctions d'écriture de l'automate M262

Vue d'ensemble

Cette section décrit les fonctions d'écriture de la bibliothèque PLCSystem de l'automate M262.

PhysicalWriteFastOutputs : Écrire la sortie rapide d'une E/S experte intégrée

Description de la fonction

Cette fonction écrit un état dans les sorties Q0 à Q3 au moment où elle est appelée.

Bibliothèque et espace de noms

Nom de la bibliothèque : **SE_PLCSystem**

Espace de noms : **SEC**

Représentation graphique

Représentation en langage IL et ST

Pour voir la représentation générale en langage IL ou ST, consultez le chapitre *Représentation des fonctions et blocs fonction*, page 77.

Description des variables d'E/S

Le tableau suivant décrit les variables d'entrée :

Entrée	Type	Commentaire
Q0Value	BOOL	La valeur est demandée pour la sortie 0.
Q1Value	BOOL	La valeur est demandée pour la sortie 1.
Q2Value	BOOL	La valeur est demandée pour la sortie 2.
Q3Value	BOOL	La valeur est demandée pour la sortie 3.

Le tableau suivant décrit la variable de sortie :

Sortie	Type	Commentaire
PhysicalWriteFastOutputs	WORD	Valeur de sortie de la fonction.

NOTE: Seuls les 4 premiers bits de la valeur de sortie sont significatifs et utilisés comme champ de bits pour indiquer si la sortie est écrite.

- Si le bit correspondant à la sortie est 1, l''écriture de cette sortie a réussi.
- Si le bit correspondant à la sortie est 0, l'écriture n'a pas été effectuée car cette sortie est déjà utilisée par une fonction experte.
- Si le bit correspondant à la sortie est 1111 en binaire, l'écriture des 4 sorties est correcte.
- Si le bit correspondant à la sortie est 1110 en binaire, la sortie Q0 n'est pas écrite car elle est utilisée par une sortie d'alarme.

NOTE: Les valeurs sont appliquées au début et à la fin d'un cycle de traitement. La fonction applique une valeur au cours du cycle.

NOTE: Si une variable est mappée sur plusieurs sorties intégrées, la dernière d'entre elles (dans l'ordre Q0 à Q3) définit la valeur de la variable à la fin de l'exécution du bloc fonction.

SetRTCDrift : Définir la valeur de compensation sur l'horodateur

Description de la fonction

Cette fonction accélère ou ralentit la fréquence de l'horodateur afin de donner la main à l'application pour compenser l'horodateur en fonction de l'environnement de fonctionnement (temperature, …). La valeur de compensation est donnée en secondes par semaine. Elle peut être positive (accélération) ou négative (ralentissement).

NOTE: La fonction *SetRTCDrift* ne doit être appelée qu'une seule fois. Chaque nouvel appel remplace la valeur de compensation précédente. La valeur est conservée dans le matériel du contrôleur pendant que l'horodateur est alimenté par la source principale ou par la batterie. Si la batterie et la source d'alimentation sont retirées, la valeur de compensation de l'horodateur n'est pas disponible.

Représentation graphique

Représentation en langage IL et ST

Pour voir la représentation générale en langage IL ou ST, consultez le chapitre *Représentation des fonctions et blocs fonction*, page 77.

Description des variables d'E/S

Le tableau suivant décrit les paramètres d'entrée :

Entrées	Type	Commentaire
RtcDrift	*SINT (-29..29)*	Correction en secondes par semaine (-29 à +29).

NOTE: Les paramètres *Day*, *Hour* et *Minute* sont utilisés uniquement pour assurer la compatibilité descendante.

NOTE: Si la valeur entrée pour *RtcDrift* dépasse la valeur de la limite, le micrologiciel du contrôleur définit la valeur sur la valeur maximale.

Ce tableau décrit la variable de sortie :

Sortie	Type	Commentaire
SetRTCDrift	*RTCSETDRIFT_ERROR*, page 56	Renvoie *RTC_OK* (00 hex) si la commande est correcte ou renvoie le code d'identification de l'erreur détectée.

Exemple

Dans cet exemple, la fonction est appelée une seule fois pendant le premier cycle de tâche MAST. Elle accélère l'horodateur de 4 secondes par semaine (18 secondes par mois).

```
VAR
MyRTCDrift : SINT (-29..+29) := 0;
MyDay : sec.DAY_OF_WEEK;
MyHour : sec.HOUR;
MyMinute : sec.MINUTE;
END_VAR
IF IsFirstMastCycle() THEN
MyRTCDrift := 4;
MyDay := 0;
MyHour := 0;
MyMinute := 0;
SetRTCDrift(MyRTCDrift, MyDay, MyHour, MyMinute);
END_IF
```

Fonctions utilisateur de M262

Présentation

Cette section décrit les fonctions utilisateur suivantes :

Fonction	Description
FB_GetFreeDiskSpace	Extrait la quantité d'espace mémoire libre sur un support de mémoire, en octets. Voir *FB_GetFreeDiskSpace* : Obtient l'espace mémoire disponible de manière asynchrone, page 34
FB_GetLabel	Extrait le libellé d'un support de mémoire. Voir *FB_GetLabel* : Obtient le libellé d'un support mémoire, page 35
FB_GetTotalDiskSpace	Extrait la taille de l'espace d'un support de mémoire, en octets. Voir *FB_GetTotalDiskSpace* : Obtient la taille du support mémoire, page 36
FB_CheckAllowedControllerMacAddr	Vérifier si une adresse MAC spécifique se trouve dans la plage valide du contrôleur. Voir *FB_CheckAllowedControllerMacAddr* : Vérifier si l'adresse MAC est autorisée par le contrôleur, page 37
FB_ControlClone	Activer ou désactiver la fonction de clonage du contrôleur. Voir *FB_ControlClone* : Cloner le contrôleur, page 38
DataFileCopy	Copier les données en mémoire dans un fichier, et vice versa. Voir *DataFileCopy* : commandes de copie de fichier, page 39
ExecuteScript	Exécuter les commands de script. Voir *ExecuteScript* : Exécution de commandes de script, page 41

FB_GetFreeDiskSpace : Obtient l'espace mémoire disponible de manière asynchrone

Description du bloc fonction

Ce bloc fonction obtient la quantité d'espace libre d'un support de mémoire (disque utilisateur, disque système, carte SD), en octets. Le nom du support mémoire est transféré.

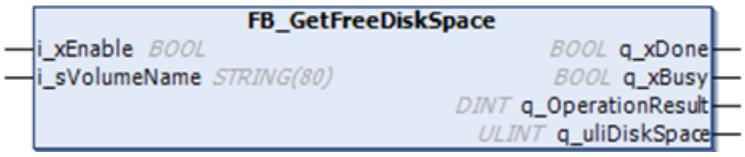
L'espace mémoire disponible sur un équipement distant n'est pas accessible. Si le paramètre d'entrée désigne un équipement distant, la fonction renvoie la valeur « –1 ».

Bibliothèque et espace de noms

Nom de la bibliothèque : **PLCSystemBase**

Espace de nom : **PLCSystemBase**

Représentation graphique



Représentation en langage IL et ST

Pour voir la représentation générale en langage IL ou ST, consultez le chapitre *Représentation des fonctions et blocs fonction*, page 77.

Description des variables d'E/S

Le tableau suivant décrit les variables d'entrée :

Entrée	Type de données	Description
i_xEnable	BOOL	Entrée d'activation, exécute l'opération lorsque la valeur est `TRUE`.
i_sVolumeName	STRING[80]	Nom de l'équipement dont l'espace mémoire disponible doit être consulté • Disque système : `'/sys'` • Disque utilisateur : `'/usr'` • Carte SD : `'/sd0'`

Le tableau suivant décrit les variables de sortie :

Sortie	Type de données	Description
q_xDone	BOOL	Défini sur `TRUE` lorsque l'exécution du bloc fonction est terminée.
q_xBusy	BOOL	Défini sur `TRUE` lorsque l'exécution du bloc fonction a commencé et se poursuit encore.
q_OperationResult	DINT	Résultat de l'opération ; une valeur différente de zéro indique une erreur.
q_uliDiskSpace	ULINT	Espace mémoire en octets.

FB_GetLabel : Obtient le libellé d'un support mémoire

Description du bloc fonction

Ce bloc fonction obtient le libellé d'un support de mémoire. Si un équipement n'a pas de libellé, une chaîne vide est renvoyée.

Bibliothèque et espace de noms

Nom de la bibliothèque : **PLCSystemBase**

Espace de nom : **PLCSystemBase**

Représentation graphique

Représentation en langage IL et ST

Pour voir la représentation générale en langage IL ou ST, consultez le chapitre *Représentation des fonctions et blocs fonction*, page 77.

Description des variables d'E/S

Le tableau suivant décrit les variables d'entrée :

Entrée	Type de données	Description
i_xEnable	BOOL	Entrée d'activation, exécute l'opération lorsque la valeur est `TRUE`.
i_sVolumeName	STRING[80]	Nom de l'équipement dont l'espace mémoire disponible doit être consulté • Disque système : `'/sys'` • Disque utilisateur : `'/usr'` • Carte SD : `'/sd0'`

Le tableau suivant décrit les variables de sortie :

Sortie	Type de données	Description
q_xDone	BOOL	Défini sur `TRUE` lorsque l'exécution du bloc fonction est terminée.
q_xBusy	BOOL	Défini sur `TRUE` lorsque l'exécution du bloc fonction a commencé et se poursuit encore.
q_OperationResult	DINT	Résultat de l'opération ; une valeur différente de zéro indique une erreur.
q_sLabel	STRING[11]	Etiquette de l'appareil.

FB_GetTotalDiskSpace : Obtient la taille du support mémoire

Description du bloc fonction

Ce bloc fonction obtient la quantité d'espace libre d'un support de mémoire (disque utilisateur, disque système, carte SD), en octets.

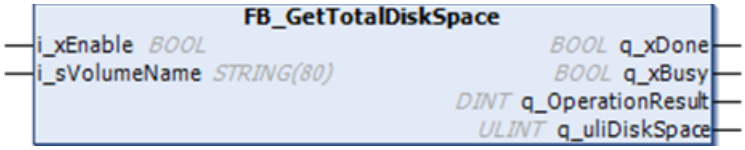
La taille d'un équipement distant n'est pas accessible. Si le paramètre d'entrée désigne un équipement distant, la fonction renvoie la valeur "-1".

Bibliothèque et espace de noms

Nom de la bibliothèque : **PLCSystemBase**

Espace de nom : **PLCSystemBase**

Représentation graphique



Représentation en langage IL et ST

Pour voir la représentation générale en langage IL ou ST, consultez le chapitre *Représentation des fonctions et blocs fonction*, page 77.

Description des variables d'E/S

Le tableau suivant décrit les variables d'entrée :

Entrée	Type de données	Description
i_xEnable	BOOL	Entrée d'activation, exécute l'opération lorsque la valeur est `TRUE`.
i_sVolumeName	STRING[80]	Nom de l'équipement dont la taille de la mémoire doit être consultée • Disque système : `'/sys'` • Disque utilisateur : `'/usr'` • Carte SD : `'/sd0'`

Le tableau suivant décrit les variables de sortie :

Sortie	Type de données	Description
q_xDone	BOOL	Défini sur `TRUE` lorsque l'exécution du bloc fonction est terminée.
q_xBusy	BOOL	Défini sur `TRUE` lorsque l'exécution du bloc fonction a commencé et se poursuit encore.
q_OperationResult	DINT	Résultat de l'opération ; une valeur différente de zéro indique une erreur.
q_uliDiskSpace	ULINT	Espace mémoire en octets.

FB_CheckAllowedControllerMacAddr : Vérifier si l'adresse MAC est autorisée par le contrôleur

Description du bloc fonction

Ce bloc fonction vérifie si une adresse MAC spécifiée est comprise dans la plage des adresses MAC autorisées pour le contrôleur. L'application ne poursuit son exécution que si l'adresse MAC est conforme. Sinon, l'application s'arrête, le contrôleur passe à l'état HALT et la variable système *i_wLastApplicationError*, page 17 est mise à jour en conséquence.

Bibliothèque et espace de noms

Nom de la bibliothèque : **PLCSystemBase**

Espace de nom : **PLCSystemBase**

Représentation graphique

Représentation en langage IL et ST

Pour voir la représentation générale en langage IL ou ST, consultez le chapitre *Représentation des fonctions et blocs fonction*, page 77.

Description des variables d'E/S

Le tableau suivant décrit les variables d'entrée :

Entrée	Type	Commentaire
byAllowedMacAddress	ARRAY[0...5] OF BYTE	Adresse MAC à vérifier `[aa.bb.cc.dd.ee.ff]` : • i_byMACAddress[0] = aa • ... • i_byMACAddress[5] = ff

Le tableau suivant décrit les variables de sortie :

Sortie	Type	Commentaire
xControllerMacAddressOk	BOOL	TRUE = indique que l'adresse MAC est autorisée pour ce contrôleur.

FB_ControlClone : Cloner le contrôleur

Description du bloc fonction

Le clonage est possible par carte SD ou par **Controller Assistant**. Lorsque les droits d'utilisateur sont activés et que le droit d'affichage **FrmUpdate** est refusé au groupe **ExternalMedia**, la fonction de clonage n'est pas autorisée. Dans ce cas, le bloc fonction active la fonctionnalité de clonage une fois lors de la mise sous tension suivante du contrôleur.

NOTE: Vous pouvez choisir d'inclure ou non les droits d'utilisateur dans le clone sur la page **Clone Management** du serveur Web (voir Modicon M262 – Logic/Motion Controller – Guide de programmation).

Le tableau suivant indique comment définir le bloc fonction et les droits d'utilisateur :

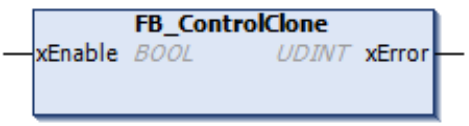
Configuration du bloc fonction	Avec droits d'utilisateur activés	Avec droits d'utilisateur désactivés
xEnable = 1	Le clonage est autorisé	Le clonage est autorisé
xEnable = 0	Le clonage n'est pas autorisé	Le clonage n'est pas autorisé

Bibliothèque et espace de noms

Nom de la bibliothèque : **PLCSystemBase**

Espace de nom : **PLCSystemBase**

Représentation graphique



Représentation en langage IL et ST

Pour voir la représentation générale en langage IL ou ST, consultez le chapitre *Représentation des fonctions et blocs fonction*, page 77.

Description des variables d'E/S

Le tableau suivant décrit les variables d'entrée :

Entrée	Type	Commentaire
xEnable	BOOL	Si `TRUE`, la fonctionnalité de clonage est activée une seule fois. Si `FALSE`, la fonctionnalité de clonage est désactivée.

Le tableau suivant décrit les variables de sortie :

Sortie	Type	Commentaire
xError	UDINT	La valeur `0` indique qu'aucune erreur n'a été détectée lors de l'exécution du bloc fonction. Une valeur différente de zéro indique qu'une erreur a été détectée.

DataFileCopy : commandes de copie de fichier

Description du bloc fonction

Ce bloc fonction copie les données en mémoire dans un fichier, et inversement. Le fichier réside dans le système de fichiers interne ou dans un système de fichiers externe (carte SD).

Le bloc fonction *DataFileCopy* peut :

- lire les données d'un fichier formaté ou
- copier des données en mémoire vers un fichier formaté. Pour plus d'informations, consultez Organisation de la mémoire non volatile (voir Modicon M262 Logic/Motion Controller - Guide de programmation).

Bibliothèque et espace de noms

Nom de la bibliothèque : **PLCSystemBase**

Espace de nom : **PLCSystemBase**

Représentation graphique

Représentation en langage IL et ST

Pour voir la représentation générale en langage IL ou ST, consultez le chapitre *Représentation des fonctions et blocs fonction*, page 77.

Description des variables d'E/S

Le tableau suivant décrit les variables d'entrée :

Entrées	Type	Commentaire
xExecute	BOOL	Sur le front montant, lance l'exécution du bloc fonction. Sur le front descendant, réinitialise les sorties du bloc fonction lorsque toute exécution en cours prend fin. **NOTE:** Sur front descendant, la fonction continue de s'exécuter jusqu'au bout et elle met à jour ses sorties en conséquence. Les sorties sont conservées pendant un cycle avant d'être réinitialisées.
sFileName	STRING	Nom de fichier sans extension (l'extension *.DTA* est automatiquement ajoutée). N'utilisez que les caractères alphanumériques a...z, A...Z, 0...9.
xRead	BOOL	TRUE : copier les données du fichier identifié par *sFileName* vers la mémoire interne du contrôleur. FALSE : copier les données de la mémoire interne du contrôleur vers le fichier identifié par *sFileName*.
xSecure	BOOL	TRUE : L 'adresse MAC est toujours stockée dans le fichier. Seul un contrôleur ayant la même adresse MAC peut lire le contenu du fichier. FALSE : Un autre contrôleur disposant du même type de mémoire peut lire le fichier.
iLocation	INT	0 : le fichier réside dans le répertoire */usr/DTA* du système de fichiers interne. 1 : le fichier réside dans le répertoire */usr/DTA* du système de fichiers externe (carte SD). **NOTE:** Si le fichier n'existe pas déjà dans le répertoire, il est créé.

uiSize	UINT	Indique la taille en octets. La taille maximale est 65534 octets.
		Seules les adresses de variables conformes à la norme CEI 61131-3 (variables, tableaux, structures) sont autorisées. Par exemple :
		`Variable : int;`
		`uiSize := SIZEOF (Variable);`

dwAdd	DWORD	Indique l'adresse en mémoire que la fonction valire ou écrire.
		Seules les adresses de variables conformes à la norme CEI 61131-3 (variables, tableaux, structures) sont autorisées. Par exemple :
		`Variable : int;`
		`dwAdd := ADR (Variable);`

⚠AVERTISSEMENT

FONCTIONNEMENT IMPRÉVU DE L'ÉQUIPEMENT

Vérifiez que la taille de la mémoire et le type du fichier sont corrects avant de copier le fichier dans la mémoire.

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

Le tableau suivant décrit les variables de sortie :

Sortie	Type	Commentaire
xDone	BOOL	`TRUE` = indique que l'action a abouti.
xBusy	BOOL	`TRUE` = indique que le bloc fonction est en cours d'exécution.
xError	BOOL	`TRUE` = indique qu'une erreur est détectée et que le bloc fonction a abandonné l'action.
eError	*DataFileCopyError*, page 50	Indique le type de l'erreur détectée lors de la copie du fichier de données.

NOTE: Si vous modifiez des données dans la mémoire (variables, tableaux, structures) utilisée pour écrire le fichier, une erreur d'intégrité CRC se produit.

Exemple

L'exemple suivant montre comment utiliser les commandes de copie de fichier :

```
VAR
LocalArray : ARRAY [0..29] OF BYTE;
myFileName: STRING := 'exportfile';
EXEC_FLAG: BOOL;
DataFileCopy: DataFileCopy;
END_VAR
DataFileCopy(
xExecute:= EXEC_FLAG,
sFileName:= myFileName,
xRead:= FALSE,
xSecure:= FALSE,
iLocation:= DFCL_INTERNAL,
uiSize:= SIZEOF(LocalArray),
dwAdd:= ADR(LocalArray),
xDone=> ,
xBusy=> ,
xError=> ,
eError=> );
```

ExecuteScript : Exécution de commandes de script

Description du bloc fonction

Ce bloc fonction peut exécuter les commandes de script de carte SD suivantes :

- *Download*
- *Upload*
- *SetNodeName*
- *Delete*
- *Reboot*

Pour plus d'informations sur le format de fichier de script requis, reportez-vous à la section Création de fichiers de script (voir Modicon M262 - Logic/Motion Controller - Guide de programmation).

Bibliothèque et espace de noms

Nom de la bibliothèque : **PLCSystemBase**

Espace de nom : **PLCSystemBase**

Représentation graphique

Représentation en langage IL et ST

Pour voir la représentation générale en langage IL ou ST, consultez le chapitre *Représentation des fonctions et blocs fonction*, page 77.

Description des variables d'E/S

Le tableau suivant décrit les variables d'entrée :

Entrées	Type	Commentaire
xExecute	BOOL	En cas de détection d'un front montant, lance l'exécution du bloc fonction. En cas de détection d'un front descendant, réinitialise les sorties du bloc fonction lorsque toute exécution en cours prend fin. **NOTE:** Sur front descendant, la fonction continue de s'exécuter jusqu'au bout et elle met à jour ses sorties en conséquence. Les sorties sont conservées pendant un cycle avant d'être réinitialisées.
sCmd	STRING	Syntaxe de commande de script de carte SD. L'exécution simultanée de commandes n'est pas autorisée : si une commande est exécutée par un autre bloc fonctionnel ou un script de carte SD, le bloc fonctionnel met la commande en file d'attente et ne l'exécute pas immédiatement. **NOTE:** un script de carte SD exécuté à partir d'une carte SD est considéré en cours d'exécution jusqu'au retrait de la carte SD.

Le tableau suivant décrit les variables de sortie :

Sortie	Type	Commentaire
xDone	BOOL	`TRUE` indique que l'action a abouti.
xBusy	BOOL	`TRUE` indique que le bloc fonction est en cours d'exécution.
xError	BOOL	`TRUE` indique une détection d'erreur ; le bloc fonction annule l'action.
eError	*ExecuteScriptError*, page 51	Indique le type de l'erreur détectée lors de l'exécution du script.

Exemple

Cet exemple décrit comment exécuter une commande de script *Upload* :

```
VAR
EXEC_FLAG: BOOL;
ExecuteScript: ExecuteScript;
END_VAR
ExecuteScript(
xExecute:= EXEC_FLAG,
sCmd:= 'Upload "/usr/Syslog/*"',
xDone=> ,
xBusy=> ,
xError=> ,
eError=> );
```

Types de données de la bibliothèque M262

Contenu de ce chapitre

Types de données des variables système *PLC_RW* 43
Types de données des variables système DataFileCopy 49
Types de données des variables système ExecScript 50
Types de données des variables système *ETH_RW* 51
Types de données des fonctions système 56

Présentation

Ce chapitre décrit les types de données de la bibliothèque PLCSystem de M262.

Deux types de données sont disponibles :

- Les types de données de variable système sont utilisés par les variables système, page 14 de la bibliothèque PLCSystem de la bibliothèque PLCSystem M262 (PLC_R, PLC_W,...).
- Les types de données de fonction système sont utilisés par les fonctions système, page 25 de lecture/écriture de la bibliothèque PLCSystem de M262.

Types de données des variables système *PLC_RW*

Présentation

Cette section répertorie et décrit les types de données de variable système, inclus dans les structures *PLC_R* et *PLC_W*.

PLC_R_APPLICATION_ERROR : Codes d'état d'erreur détecté de l'application

Bibliothèque et espace de noms

Nom de la bibliothèque : **PLCSystemBase**

Espace de nom : **PLCSystemBase**

Description du type Énumération

Le type de données énumération *PLC_R_APPLICATION_ERROR* contient les valeurs suivantes :

Énumérateur	Valeur	Commentaire	Que faire
PLC_R_APP_ERR_UNKNOWN	FFFF hex	Erreur indéfinie détectée.	Contactez le représentant local Schneider Electric.
PLC_R_APP_ERR_NOEXCEPTION	0000 hex	Aucune erreur détectée.	–
PLC_R_APP_ERR_WATCHDOG	0010 hex	Horloge de surveillance de tâche arrivée à expiration.	Vérifiez votre application et effectuez les corrections si nécessaire. Voir Horloges de surveillance du système et des tâches (voir Modicon M262 - Logic/Motion Controller - Guide de programmation). Une réinitialisation est nécessaire pour entrer en mode RUNNING.
PLC_R_APP_ERR_HARDWAREWATCHDOG	0011 hex	L'horloge de surveillance matérielle interne a expiré.	Si le problème est reproductible, vérifiez qu'il n'y a aucun port de communication configuré mais déconnecté. Sinon, mettez à jour le micrologiciel. Si le problème persiste, contactez le service de maintenance Schneider Electric.
PLC_R_APP_ERR_IO_CONFIG_ERROR	0012 hex	Paramètres de configuration d'E/S incorrects détectés.	Il est possible que votre application soit endommagée. Pour résoudre ce problème, rechargez l'application à partir du PC. Sinon, contactez le représentant local Schneider Electric.
PLC_R_APP_ERR_UNRESOLVED_EXTREFS	0018 hex	Fonctions non définies détectées.	Supprimez les fonctions non résolues de l'application.
PLC_R_APP_ERR_IEC_TASK_CONFIG_ERROR	0025 hex	Paramètres de configuration de tâche incorrects détectés.	Il est possible que votre application soit endommagée. Pour résoudre ce problème, rechargez l'application à partir du PC. Sinon, contactez le représentant local Schneider Electric.
PLC_R_APP_ERR_TARGET_MISMATCH	0026 hex	Le contrôleur n'est pas autorisé à exécuter cette application CEI.	L'application n'a pas pu être exécutée car ce n'est pas le type de cible ou de contrôleur correct.
PLC_R_APP_ERR_ILLEGAL_INSTRUCTION	0050 hex	Instruction indéfinie détectée.	Procédez au débogage de votre application pour résoudre le problème.
PLC_R_APP_ERR_ACCESS_VIOLATION	0051 hex	Tentative d'accès à la zone mémoire réservée.	Procédez au débogage de votre application pour résoudre le problème.
PLC_R_APP_ERR_DIVIDE_BY_ZERO	0102 hex	Division d'un entier par 0 détectée.	Procédez au débogage de votre application pour résoudre le problème.
PLC_R_APP_ERR_PROCESSORLOAD_WATCHDOG	0105 hex	Le processeur est surchargé par des tâches d'application et l'horloge de surveillance système a expiré.	Réduisez la charge de travail de l'application en améliorant son architecture. Réduisez la fréquence des événements.
PLC_R_APP_ERR_DIVIDE_REAL_BY_ZERO	0152 hex	Division d'un réel par 0 détectée.	Procédez au débogage de votre application pour résoudre le problème.
PLC_R_APP_ERR_EXPIO_EVENTS_COUNT_EXCEEDED	4E20 hex	Trop d'événements sur les E/S expertes sont détectés.	Réduisez le nombre de tâches d'événement externes.
PLC_R_APP_ERR_APPLICATION_VERSION_MISMATCH	4E21 hex	Discordance détectée dans la version de l'application.	La version de l'application dans le contrôleur ne correspond pas à celle dans le logiciel. Consultez le document Applications, voir EcoStruxure Automation Expert - Motion / EcoStruxure Machine Expert – Guide de programmation.

PLC_R_BOOT_PROJECT_STATUS : Codes d'état du projet de démarrage

Bibliothèque et espace de noms

Nom de la bibliothèque : **PLCSystemBase**

Espace de nom : **PLCSystemBase**

Description du type Énumération

Le type de données énumération *PLC_R_BOOT_PROJECT_STATUS* contient les valeurs suivantes :

Énumérateur	Valeur	Commentaire
PLC_R_NO_BOOT_PROJECT	0000 hex	Le projet de démarrage n'existe pas dans la mémoire non volatile.
PLC_R_BOOT_PROJECT_CREATION_IN_PROGRESS	0001 hex	Le projet de démarrage est en cours de création.
PLC_R_DIFFERENT_BOOT_PROJECT	0002 hex	Le projet de démarrage en mémoire non volatile est différent du projet chargé dans la mémoire.
PLC_R_VALID_BOOT_PROJECT	FFFF hex	Le projet de démarrage en mémoire non volatile est identique au projet chargé dans la mémoire.

Description du type Énumération

Le type de données énumération *PLC_R_BOOT_PROJECT_STATUS* contient les valeurs suivantes :

Énumérateur	Valeur	Commentaire
PLC_R_NO_BOOT_PROJECT	0000 hex	Le projet de démarrage n'existe pas dans la mémoire non volatile.
PLC_R_BOOT_PROJECT_CREATION_IN_PROGRESS	0001 hex	Le projet de démarrage est en cours de création.
PLC_R_DIFFERENT_BOOT_PROJECT	0002 hex	Le projet de démarrage en mémoire non volatile est différent du projet chargé dans la mémoire.
PLC_R_VALID_BOOT_PROJECT	FFFF hex	Le projet de démarrage en mémoire non volatile est identique au projet chargé dans la mémoire.

PLC_R_IO_STATUS : Codes d'état des E/S :

Bibliothèque et espace de noms

Nom de la bibliothèque : **PLCSystemBase**

Espace de nom : **PLCSystemBase**

Description du type Énumération

Le type de données énumération *PLC_R_IO_STATUS* contient les valeurs suivantes :

Énumérateur	Valeur	Commentaire
PLC_R_IO_OK	FFFF hex	Les entrées/sorties sont opérationnelles.
PLC_R_IO_NO_INIT	0001 hex	Les entrées/sorties ne sont pas initialisées.
PLC_R_IO_CONF_FAULT	0002 hex	Paramètres de configuration d'E/S incorrects détectés.
PLC_R_IO_SHORTCUT_FAULT	0003 hex	Court-circuit des entrées/sorties détecté.
PLC_R_IO_POWER_SUPPLY_FAULT	0004 hex	Erreur d'alimentation des E/S détectée.

PLC_R_SDCARD_STATUS : Codes d'état d'emplacement de carte SD

Bibliothèque et espace de noms

Nom de la bibliothèque : **PLCSystemBase**

Espace de nom : **PLCSystemBase**

Description du type Énumération

Le type de données énumération *PLC_R_SDCARD_STATUS* contient les valeurs suivantes :

Énumérateur	Valeur	Commentaire
NO_SDCARD	0000 hex	Aucune carte SD n'est détectée dans l'emplacement ou l'emplacement n'est pas connecté.
SDCARD_READONLY	0001 hex	La carte SD est en mode de lecture seule.
SDCARD_READWRITE	0002 hex	La carte SD est en mode de lecture/écriture.
SDCARD_ERROR	0003 hex	Erreur détectée sur la carte SD. Plus de détails sont écrits dans le fichier FwLog.txt.

PLC_R_STATUS : Codes d'état du contrôleur

Bibliothèque et espace de noms

Nom de la bibliothèque : **PLCSystemBase**

Espace de nom : **PLCSystemBase**

Description du type Énumération

Le type de données énumération *PLC_R_STATUS* contient les valeurs suivantes :

Énumérateur	Valeur	Commentaire
PLC_R_EMPTY	0000 hex	Le contrôleur ne contient aucune application.
PLC_R_STOPPED	0001 hex	Le contrôleur est arrêté.
PLC_R_RUNNING	0002 hex	Le contrôleur fonctionne.
PLC_R_HALT	0004 hex	Le contrôleur est dans un état HALT (voir le schéma des états de contrôleur dans le guide de programmation (voir Modicon M262 - Logic/Motion Controller - Guide de programmation)) de votre contrôleur.
PLC_R_BREAKPOINT	0008 hex	Le contrôleur s'est interrompu au niveau d'un point d'arrêt.

PLC_R_STOP_CAUSE : Codes de cause de transition de l'état RUN à un autre état

Bibliothèque et espace de noms

Nom de la bibliothèque : **PLCSystemBase**

Espace de nom : **PLCSystemBase**

Description du type Énumération

Le type de données énumération *PLC_R_STOP_CAUSE* contient les valeurs suivantes :

Énumérateur	Valeur	Commentaire	Que faire
PLC_R_STOP_REASON_UNKNOWN	00 hex	La valeur initiale ou la cause de l'arrêt n'est pas déterminable.	Contactez votre représentant Schneider Electric local.
PLC_R_STOP_REASON_HW_WATCHDOG	01 hex	Arrêté suite au timeout du chien de garde matériel	Contactez votre représentant Schneider Electric local.
PLC_R_STOP_REASON_RESET	02 hex	Arrêté suite à une réinitialisation.	Voir les possibilités de réinitialisation dans le Schéma des états de contrôleur (voir Modicon M262 - Logic/Motion Controller - Guide de programmation).
PLC_R_STOP_REASON_EXCEPTION	03 hex	Arrêté suite à une exception.	Vérifiez votre application et effectuez les corrections si nécessaire. Voir Horloges de surveillance du système et des tâches (voir Modicon M262 - Logic/Motion Controller - Guide de programmation). Une réinitialisation est nécessaire pour entrer en mode Run.
PLC_R_STOP_REASON_USER	04 hex	Arrêté suite à une requête de l'utilisateur.	Consultez la section Commande d'arrêt (Stop) dans Commande de transitions d'état (voir Modicon M262 - Logic/Motion Controller - Guide de programmation).
PLC_R_STOP_REASON_IECPROGRAM	05 hex	Arrêté suite à une requête de commande de programme (par exemple, commande de contrôle avec le paramètre *PLC_W.q_wPLCControl:=PLC_W_COMMAND.PLC_W_STOP;*).	–
PLC_R_STOP_REASON_DELETE	06 hex	Arrêté suite à une commande de suppression d'application.	Reportez-vous à l'onglet **Applications** de l'Editeur d'appareil du contrôleur (voir Modicon M262 - Logic/Motion Controller - Guide de programmation).

Énumérateur	Valeur	Commentaire	Que faire
PLC_R_STOP_REASON_DEBUGGING	07 hex	Arrêté suite au passage en mode de débogage.	–
PLC_R_STOP_FROM_NETWORK_ REQUEST	0A hex	Arrêté suite à une requête du réseau (clé USB ou commande *PLC_W*).	–
PLC_R_STOP_FROM_INPUT	0B hex	Arrêt requis par une entrée du contrôleur.	–
PLC_R_STOP_FROM_RUN_STOP_SWITCH	0C en hexadéci-mal	Arrêt demandé par le commutateur du contrôleur.	–
PLC_R_STOP_REASON_RETAIN_ MISMATCH	0D en hexadéci-mal	Arrêté suite à un échec du test de vérification du contexte lors du redémarrage.	Certaines variables conservées dans la mémoire non volatile n'existent pas dans l'application en cours d'exécution. Vérifiez votre application, effectuez les corrections si nécessaire, puis rétablissez l'application de démarrage.
PLC_R_STOP_REASON_BOOT_APPLI_ MISMATCH	0E hex	Arrêté suite à un échec de la comparaison entre l'application de démarrage et celle qui était en mémoire avant le redémarrage.	Créez une application de démarrage valide.
PLC_R_STOP_REASON_POWERFAIL	0F hex	Arrêté suite à une coupure de courant.	Reportez-vous aux possibilités de réinitialisation dans le Schéma des états du contrôleur (voir Modicon M262 - Logic/Motion Controller - Guide de programmation).

Pour plus d'informations sur les raisons de l'arrêt du contrôleur, reportez-vous à la Description des états de contrôleur (voir Modicon M262 - Logic/Motion Controller - Guide de programmation).

PLC_R_TERMINAL_PORT_STATUS : Codes d'état de connexion du port de programmation

Bibliothèque et espace de noms

Nom de la bibliothèque : **PLCSystemBase**

Espace de nom : **PLCSystemBase**

Description du type Énumération

Le type de données énumération *PLC_R_TERMINAL_PORT_STATUS* contient les valeurs suivantes :

Énumérateur	Valeur	Commentaire
TERMINAL_NOT_CONNECTED	00 hex	Aucun PC n'est connecté au port de programmation.
TERMINAL_CONNECTION_IN_PROGRESS	01 hex	Connexion en cours.
TERMINAL_CONNECTED	02 hex	PC connecté au port de programmation.
TERMINAL_ERROR	0F en hexadécimal	Erreur détectée lors de la connexion.

PLC_R_TM3_BUS_STATE : Codes d'état du bus TM3

Bibliothèque et espace de noms

Nom de la bibliothèque : **PLCSystemBase**

Espace de nom : **PLCSystemBase**

Description du type Énumération

Le type de données énumération *PLC_R_TM3_BUS_STATE* contient les valeurs suivantes :

Énumérateur	Valeur	Commentaire
TM3_CONF_ERROR	01 hex	Erreur détectée en raison d'une incohérence entre la configuration physique et la configuration dans le logiciel.
TM3_BUS_ERROR	02 hex	Erreur détectée dans la bibliothèque System *q_wIOBusErrPassiv*, page 67 et TM3.
TM3_OK	03 hex	La configuration physique correspond à la configuration dans le logiciel.
TM3_POWER_SUPPLY_ERROR	04 hex	Erreur détectée dans l'alimentation.

PLC_W_COMMAND : Codes de commande de contrôle

Bibliothèque et espace de noms

Nom de la bibliothèque : **PLCSystemBase**

Espace de nom : **PLCSystemBase**

Description du type Énumération

Le type de données énumération *PLC_W_COMMAND* contient les valeurs suivantes :

Énumérateur	Valeur	Commentaire
PLC_W_STOP	0001 hex	Commande d'arrêt du contrôleur.
PLC_W_RUN	0002 hex	Commande d'exécution du contrôleur.
PLC_W_RESET_COLD	0004 hex	Commande de lancement d'une réinitialisation à froid du contrôleur.
PLC_W_RESET_WARM	0008 hex	Commande de lancement d'une réinitialisation à chaud du contrôleur.

Types de données des variables système DataFileCopy

Présentation

Cette section répertorie et décrit les types de données des variables système incluses dans les structures *DataFileCopy*.

DataFileCopyError : Codes d'erreur détectée

Bibliothèque et espace de noms

Nom de la bibliothèque : **PLCSystemBase**

Espace de nom : **PLCSystemBase**

Description du type Énumération

Le type de données énumération *DataFileCopyError* contient les valeurs suivantes :

Énumérateur	Valeur	Description
ERR_NO_ERR	00 hex	Aucune erreur détectée.
ERR_FILE_NOT_FOUND	01 hex	Fichier inexistant.
ERR_FILE_ACCESS_REFUSED	02 hex	Ouverture du fichier impossible.
ERR_INCORRECT_SIZE	03 hex	Requête d'une taille différente de celle indiquée dans le fichier.
ERR_CRC_ERR	04 hex	CRC incorrect. Le fichier est considéré comme endommagé.
ERR_INCORRECT_MAC	05 hex	Le contrôleur tentant de lire le fichier n'a pas la même adresse MAC que celle indiquée dans le fichier.

DataFileCopyLocation : Codes d'emplacement

Bibliothèque et espace de noms

Nom de la bibliothèque : **PLCSystemBase**

Espace de nom : **PLCSystemBase**

Description du type Énumération

Le type de données énumération *DataFileCopyLocation* contient les valeurs suivantes :

Énumérateur	Valeur	Description
DFCL_INTERNAL	00 hex	Le fichier de donnée d'extension DTA se trouve dans le répertoire */usr/Dta*.
DFCL_EXTERNAL	01 hex	Le fichier de donnée d'extension DTA se trouve dans le répertoire */sd0/usr/Dta*.
DFCL_TBD	02 hex	Non utilisé.

Types de données des variables système ExecScript

Présentation

Cette section répertorie et décrit les types de données des variables système incluses dans les structures *ExecScript*.

ExecuteScriptError : Codes d'erreur détectée

Bibliothèque et espace de noms

Nom de la bibliothèque : **PLCSystemBase**

Espace de nom : **PLCSystemBase**

Description du type Énumération

Le type de données énumération *ExecuteScriptError* contient les valeurs suivantes :

Énumérateur	Valeur	Description
CMD_OK	00 hex	Aucune erreur détectée.
ERR_CMD_UNKNOWN	01 hex	La commande n'est pas valide.
ERR_SD_CARD_MISSING	02 hex	Carte SD absente.
ERR_SEE_FWLOG	03 hex	Erreur détectée lors de l'exécution de la commande, voir `FwLog.txt`. Pour plus d'informations, reportez-vous à la section Type de fichier (voir Modicon M262 - Logic/Motion Controller - Guide de programmation).
ERR_ONLY_ONE_COMMAND_ALLOWED	04 hex	Tentative d'exécution de plusieurs scripts simultanément.
CMD_BEING_EXECUTED	05 hex	Un script est déjà en cours.

Types de données des variables système *ETH_RW*

Présentation

Cette section répertorie et décrit les types de données de variable système, inclus dans les structures *ETH_R* et *ETH_W*.

ETH_R_FRAME_PROTOCOL : Codes du protocole de transmission de trames

Bibliothèque et espace de noms

Nom de la bibliothèque : **PLCSystemBase**

Espace de nom : **PLCSystemBase**

Description du type Énumération

Le type de données énumération *ETH_R_FRAME_PROTOCOL* contient les valeurs suivantes :

Énumérateur	Valeur	Commentaire
ETH_R_802_3	00 hex	Le protocole utilisé pour la transmission de trames est IEEE 802.3.
ETH_R_ETHERNET_II	01 hex	Le protocole utilisé pour la transmission des trames est Ethernet II.

ETH_R_IPFORWARDING : Transfert IP

Bibliothèque et espace de noms

Nom de la bibliothèque : **PLCSystemBase**

Espace de nom : **PLCSystemBase**

Description du type Énumération

Le type de données énumération *ETH_R_IPFORWARDING* contient les valeurs suivantes :

Énumérateur	Valeur	Commentaire
DISABLED	00 hex	Transfert IP désactivé.
ENABLED	01 hex	Transfert IP activé.

ETH_R_IP_MODE : Codes de source d'adresse IP

Bibliothèque et espace de noms

Nom de la bibliothèque : **PLCSystemBase**

Espace de nom : **PLCSystemBase**

Description du type Énumération

Le type de données énumération *ETH_R_IP_MODE* contient les valeurs suivantes :

Énumérateur	Valeur	Commentaire
ETH_R_STORED	00 hex	L'adresse IP stockée est utilisée.
ETH_R_BOOTP	01 hex	Le protocole Bootstrap (BOOTP) est utilisé pour obtenir une adresse IP.
ETH_R_DHCP	02 hex	Le protocole DHCP est utilisé pour obtenir une adresse IP.
ETH_DEFAULT_IP	FF hex	L'adresse IP par défaut est utilisée.

ETH_R_ITF_STRUCT : Paramètres de l'interface Ethernet

Bibliothèque et espace de noms

Nom de la bibliothèque : **PLCSystemBase**

Espace de nom : **PLCSystemBase**

Description du type Énumération

Le type de données énumération *ETH_R_ITF_STRUCT* contient les valeurs suivantes :

%MW	Nom de la variable	Type	Commentaire
s/o	*i_byIPAddress*	ARRAY[0..3] OF BYTE	Adresse IP de cette interface Ethernet `[aaa.bbb.ccc.ddd]` : • i_byIPAddress[0] = aaa • ... • i_byIPAddress[3]= ddd
s/o	*i_bySubNetMask*	ARRAY[0..3] OF BYTE	Masque de sous-réseau de cette interface Ethernet `[aaa.bbb.ccc.ddd]` : • i_bySub-netMask[0] = aaa • ... • i_bySub-netMask[3] = ddd
s/o	*i_byGateway*	ARRAY[0..3] OF BYTE	Adresse de passerelle de cette interface Ethernet `[aaa.bbb.ccc.ddd]` : • i_byGateway[0] = aaa • ... • i_byGateway[3] = ddd
s/o	*i_byMACAddress*	ARRAY[0..5] OF BYTE	Adresse MAC de cette interface Ethernet `[aa.bb.cc.dd.ee.ff]` : • i_byMACAddress[0] = aa • ... • i_byMACAddress[5]= ff
s/o	*i_sDeviceName*	STRING(15)	Nom utilisé pour obtenir l'adresse IP auprès du serveur.
s/o	*i_wIpMode*	*ETH_R_IP_MODE*, page 52	Méthode utilisée pour obtenir une adresse IP.
s/o	*i_byFDRServerIPAddress*	ARRAY[0..3] OF BYTE	Adresse IP `[aaa.bbb.ccc.ddd]` du serveur DHCP ou BootP : • i_byFDRServerIPAddress[0] = aaa • ... • i_byFDRServerIPAddress[3] = ddd Egale à 0.0.0.0 en cas d'utilisation d'une adresse IP enregistrée ou par défaut.
s/o	*i_udiFramesTransmittedOK*	UDINT	Nombre de trames transmises correctement. Réinitialisation lors de la mise sous tension ou avec la commande de réinitialisation *ETH_W.q_wResetCounter*.
s/o	*i_udiFramedReceivedOK*	UDINT	Nombre de trames reçues correctement. Réinitialisation lors de la mise sous tension ou avec la commande de réinitialisation *ETH_W.q_wResetCounter*.
s/o	*i_udiTransmitBufferErrors*	UDINT	Nombre de trames transmises avec détection d'erreurs. Réinitialisation lors de la mise sous tension ou avec la commande de réinitialisation *ETH_W.q_wResetCounter*.
s/o	*i_udiReceiveBufferErrors*	UDINT	Nombre de trames reçues avec détection d'erreurs. Réinitialisation lors de la mise sous tension ou avec la commande de réinitialisation *ETH_W.q_wResetCounter*.
s/o	*i_wPortALinkStatus*	*ETH_R_PORT_LINK_STATUS*, page 54	Liaison du port Ethernet (0 = aucune liaison, 1 = liaison connectée à un autre équipement Ethernet).
s/o	*i_wPortASpeed*	*ETH_R_PORT_SPEED*, page 55	Vitesse réseau du port Ethernet (10 Mbits/s, 100 Mbits/s ou 1 Gbit/s).
s/o	*i_wPortADuplexStatus*	*ETH_R_PORT_DUPLEX_STATUS*, page 54	Etat duplex du port Ethernet (0 = semi duplex ou 1 = duplex intégral).
s/o	*i_wPortAIpStatus*	*ETH_R_PORT_IP_STATUS*, page 54	Etat de la pile du port TCP/IP Ethernet
s/o signifie qu'aucun mappage %MW n'est prédéfini pour cette variable système.			

ETH_R_PORT_DUPLEX_STATUS : Codes de mode de transmission

Bibliothèque et espace de noms

Nom de la bibliothèque : **PLCSystemBase**

Espace de nom : **PLCSystemBase**

Description du type Énumération

Le type de données énumération *ETH_R_PORT_DUPLEX_STATUS* contient les valeurs suivantes :

Énumérateur	Valeur	Commentaire
ETH_R_PORT_HALF_DUPLEX	00 hex	Le mode de transmission en semi-duplex est utilisé.
ETH_R_FULL_DUPLEX	01 hex	Le mode de transmission en duplex intégral est utilisé.
ETH_R_PORT_NA_DUPLEX	03 hex	Le mode de transmission sans duplex est utilisé.

ETH_R_PORT_IP_STATUS : Codes d'état du port Ethernet TCP/IP

Bibliothèque et espace de noms

Nom de la bibliothèque : **PLCSystemBase**

Espace de nom : **PLCSystemBase**

Description du type Énumération

Le type de données énumération *ETH_R_PORT_IP_STATUS* contient les valeurs suivantes :

Énumérateur	Valeur	Commentaire
WAIT_FOR_PARAMS	00 hex	Attente de paramètres.
WAIT_FOR_CONF	01 hex	Attente de configuration.
DATA_EXCHANGE	02 hex	Prêt pour l'échange de données.
ETH_ERROR	03 hex	Erreur détectée sur le port Ethernet TCP/IP (câble déconnecté, configuration non valide, etc.).
DUPLICATE_IP	04 hex	L'adresse IP est déjà utilisée par un autre équipement.

ETH_R_PORT_LINK_STATUS : Codes d'état de Liaison de communication

Bibliothèque et espace de noms

Nom de la bibliothèque : **PLCSystemBase**

Espace de nom : **PLCSystemBase**

Description du type Énumération

Le type de données énumération *ETH_R_PORT_LINK_STATUS* contient les valeurs suivantes :

Énumérateur	Valeur	Commentaire
ETH_R_LINK_DOWN	00 hex	Liaison de communication non disponible pour un autre équipement.
ETH_R_LINK_UP	01 hex	Liaison de communication disponible pour un autre équipement.

ETH_R_PORT_SPEED : Codes de vitesse de communication des ports Ethernet

Bibliothèque et espace de noms

Nom de la bibliothèque : **PLCSystemBase**

Espace de nom : **PLCSystemBase**

Description du type Énumération

Le type de données énumération *ETH_R_PORT_SPEED* contient les valeurs suivantes :

Énumérateur	Valeur	Commentaire
ETH_R_SPEED_NA	0 déc	Le débit réseau est de 0 mégabits par seconde.
ETH_R_SPEED_10_MB	10 déc	Le débit réseau est de 10 mégabits par seconde.
ETH_R_100_MB	100 déc	Le débit réseau est de 100 mégabits par seconde.
ETH_R_1_GB	1000 déc	Le débit réseau est de 1 gigabit par seconde.

ETH_R_RUN_IDLE : Codes des états de fonctionnement et d'inactivité Ethernet/IP

Bibliothèque et espace de noms

Nom de la bibliothèque : **PLCSystemBase**

Espace de nom : **PLCSystemBase**

Description du type Énumération

Le type de données énumération *ETH_R_RUN_IDLE* contient les valeurs suivantes :

Énumérateur	Valeur	Commentaire
IDLE	00 hex	La connexion EtherNet/IP est au repos.
RUN	01 hex	La connexion EtherNet/IP est en fonctionnement.

Types de données des fonctions système

Présentation

Cette section décrit les différents types de données des fonctions système de la bibliothèque PLCSystem de M262.

IMMEDIATE_ERR_TYPE : Lecture des entrées *GetImmediateFastInput* des codes d'E/S expertes intégrées

Bibliothèque et espace de noms

Nom de la bibliothèque : **PLCSystemBase**

Espace de nom : **PLCSystemBase**

Description du type Énumération

Le type de données énumération contient les valeurs suivantes :

Énumérateur	Type	Commentaire
IMMEDIATE_NO_ERROR	Mot	Aucune erreur détectée.
IMMEDIATE_UNKNOWN	Mot	La référence de la fonction *Immediate* est incorrecte ou non configurée.
IMMEDIATE_UNKNOWN_PARAMETER	Mot	Une référence de paramètre est incorrecte.

RTCSETDRIFT_ERROR : Codes des erreurs détectées par la fonction *SetRTCDrift*

Description du type énumération

Le type de données énumération *RTCSETDRIFT_ERROR* contient les valeurs suivantes :

Enumérateur	Valeur	Commentaire
RTC_OK	00 hex	L'écart de l'horodateur est configuré correctement.
RTC_BAD_DAY	01 hex	Non utilisé.
RTC_BAD_HOUR	02 hex	Non utilisé.
RTC_BAD_MINUTE	03 hex	Non utilisé.
RTC_BAD_DRIFT	04 hex	Paramètre d'écart de l'horodateur hors limites.
RTC_INTERNAL_ERROR	05 hex	Paramètres d'écart de l'horodateur rejetés sur détection d'une erreur interne.

Système SerialLine

Contenu de cette partie

Variables système ligne série M262 58

Introduction

Cette section décrit la bibliothèque M262 SerialLine.

Variables système ligne série M262

Contenu de ce chapitre

SERIAL_R : Variables de diagnostic de ligne série 58
SERIAL_W : Variables de diagnostic de ligne série 59

Présentation

Ce chapitre décrit les variables système ligne série de la bibliothèque M262 PLCSystem.

SERIAL_R : Variables de diagnostic de ligne série

Bibliothèque et espace de noms

Nom de la bibliothèque : **SerialLineSystem**

Espace de noms : **SEC_SLSYS**

Introduction

La variable système *SERIAL_R* (type *SERIAL_R_ARRAY_TYPE*) est un tableau de *SERIAL_R_STRUCT*.

Structure de la variable

Ce tableau décrit les paramètres de *SERIAL_R_STRUCT* :

Nom de la variable	Type	Commentaire
i_udiFramesTransmittedOK	UDINT	Indique le nombre de trames transmises depuis le démarrage par le bus de ligne série.
i_udiFramesReceivedOK	UDINT	Indique le nombre de trames reçues depuis le démarrage par le bus de ligne série.
i_udiRX_MessagesError	UDINT	Indique que le registre *RX Error Counter* compte le nombre de trames reçues avec des erreurs détectées, pour tous les types de trame.
i_uiSlaveExceptionCount	UINT	Indique le nombre d'erreurs d'exception de bus détectées.
i_uiSlaveMsgCount	UINT	Indique le nombre de messages d'esclave.
i_uiSlaveNoRespCount	UINT	Indique le nombre de réponses absentes attendues des esclaves.
i_uiSlaveNakCount	UINT	Indique le nombre d'indications NAK d'esclave.
i_uiSlaveBusyCount	UINT	Indique le nombre d'indications d'esclave occupé
i_uiCharOverrunCount	UINT	Indique le nombre de dépassements de caractères rencontrés.

SERIAL_W : Variables de diagnostic de ligne série

Bibliothèque et espace de noms

Nom de la bibliothèque : **SerialLineSystem**

Espace de noms : **SEC_SLSYS**

Introduction

La variable système *SERIAL_W* (type *SERIAL_W_ARRAY_TYPE*) est un tableau de *SERIAL_W_STRUCT*.

Structure de la variable

Ce tableau décrit les paramètres de *SERIAL_W_STRUCT* :

Nom de la variable	Type	Commentaire
q_wResetCounter	WORD	La transition de 0 à 1 réinitialise le compteur d'événements. Pour réinitialiser à nouveau le compteur, il est nécessaire d'écrire 0 dans cette variable afin de permettre une autre transition de 0 à 1.

Système TM3

Contenu de cette partie

Variables système du TM3 61
Fonctions système du TM3 62
Types de données système du TM3 66

Introduction

Cette section décrit la bibliothèque système des TM3

Variables système du TM3

Contenu de ce chapitre

TM3_MODULE_R[0...13] : Variables système en lecture seule des modules TM3 61

TM3_MODULE_R[0...13] : Variables système en lecture seule des modules TM3

Bibliothèque et espace de noms

Nom de la bibliothèque : **TM3System**

Espace de noms : **SEC_TM3Sys**

Introduction

Le *TM3_MODULE_R* est un tableau de type 14 *TM3_MODULE_R_STRUCT*. Chaque élément du tableau renvoie des variables système de diagnostic pour le module d'extension TM3 correspondant.

Pour le Modicon M262 Logic/Motion Controller :

- *TM3_MODULE_R[0]* désigne le module d'extension TM3 numéro 0
- ...
- *TM3_MODULE_R[13]* désigne le module d'extension TM3 numéro 13

Structure de la variable

Le tableau suivant décrit les paramètres de la variable système *TM3_MODULE_R [13]* :

%MW	Nom de la variable	Type	Commentaire
s/o	*i_wProductID*	WORD	ID du module d'extension TM3.
s/o	*i_wModuleState*	*TM3_MODULE_STATE*, page 68	Décrit l'état du module TM3.

NOTE: s/o signifie qu'aucun mappage %MW n'est prédéfini pour cette variable système.

Fonctions système du TM3

Contenu de ce chapitre

TM3_GetModuleBusStatus : Obtenir l'état du bus du module TM3 62
TM3_GetModuleInternalStatus : Obtenir l'état interne du module TM3.....63

TM3_GetModuleBusStatus : Obtenir l'état du bus du module TM3

Description de la fonction

Cette fonction renvoie le statut de bus du module. L'index du module est fourni en tant que paramètre d'entrée.

Bibliothèque et espace de noms

Nom de la bibliothèque : **TM3System**

Espace de noms : **SEC_TM3Sys**

Représentation graphique

Représentation en langage IL et ST

Pour voir la représentation générale en langage IL ou ST, consultez le chapitre *Représentation des fonctions et blocs fonction*, page 77.

Description des variables d'E/S

Le tableau suivant décrit la variable d'entrée :

Entrées	Type	Commentaire
ModuleIndex	BYTE	Index du module d'extension (0 pour le module le plus proche du contrôleur, 1 pour le second plus proche, etc.)

Le tableau suivant décrit la variable de sortie :

Sortie	Type	Commentaire
TM3_GetModuleBusStatus	*TM3_ERR_CODE*	Renvoie *TM3_NO_ERR* (00 hex) si la commande est correcte ou renvoie le code d'identification de l'erreur détectée.

TM3_GetModuleInternalStatus : Obtenir l'état interne du module TM3

Description de la fonction

Cette fonction remplit le *pStatusBuffer* avec le statut de bus du module *ModuleIndex*.

Cette fonction lit sélectivement l'état des voies d'E/S d'un module TM3 analogique ou de température, indiqué par *ModuleIndex*. Le bloc fonction écrit l'état de chaque voie demandée en commençant à l'emplacement mémoire indiqué par *pStatusBuffer*.

NOTE: Ce bloc fonction est destiné à être utilisé avec des modules d'E/S analogiques et thermiques. Pour obtenir des informations d'état sur les modules d'E/S numériques, consultez *TM3_GetModuleBusStatus*, page 62.

NOTE: Il est possible de mettre à jour la valeur des octets de diagnostic en appelant la fonction *TM3_GetModuleInternalStatus*, à condition que le paramètre d'activation de l'état soit **désactivé** dans l'onglet **Configuration d'E/S**.

Bibliothèque et espace de noms

Nom de la bibliothèque : **TM3System**

Espace de noms : **SEC_TM3Sys**

Représentation graphique

Représentation en langage IL et ST

Pour voir la représentation générale en langage IL ou ST, consultez le chapitre *Représentation des fonctions et blocs fonction*, page 77.

Description des variables d'E/S

Chaque voie d'E/S analogique/thermique du module demandé requiert un octet de mémoire. Si la mémoire allouée au tampon n'est pas suffisante pour le nombre d'états de voie de module d'E/S demandé, il est possible que la fonction écrase la mémoire allouée à d'autres fins ou essaie d'écraser une zone de mémoire à usage restreint.

⚠AVERTISSEMENT

FONCTIONNEMENT IMPRÉVU DE L'ÉQUIPEMENT

Assurez-vous que *pStatusBuffer* pointe vers une zone mémoire suffisante pour le nombre de voies à lire.

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

Le tableau suivant décrit les variables d'entrée :

Entrée	Type	Commentaire
ModuleIndex	BYTE	Index du module d'extension (0 pour le module le plus proche du contrôleur, 1 pour le second plus proche, etc.)
StatusOffset	BYTE	Décalage du premier état à lire dans la table d'états.
StatusSize	BYTE	Nombre d'octets à lire dans la table d'états.
pStatusBuffer	POINTER TO BYTE	Mémoire tampon contenant la table d'états lus (IBStatusIWx / IBStatusQWx).

Le tableau suivant décrit la variable de sortie :

Sortie	Type	Commentaire
TM3_ GetModuleInternalStatus	*TM3_ERR_CODE*, page 67	Renvoie *TM3_NO_ERR* (00 hex) si la commande est correcte, sinon renvoie le code d'identification de l'erreur. Dans le cadre de ce bloc fonction, toute valeur renvoyée différente de zéro indique que le module n'est pas compatible avec la requête d'état ou qu'il rencontre d'autres problèmes de communication.

Exemple

Les exemples suivants décrivent comment obtenir l'état interne du module :

```
VAR
TM3AQ2_Channel_0_Output_Status: BYTE;
END_VAR
TM3AQ2 is on position 1
Status of channel 0 is at offset 0
We read 1 channel
TM3_GetModuleInternalStatus(1, 0, 1, ADR(TM3AQ2_Channel_0_
Output_Status));
status of channel 0 is in TM3AQ2_Channel_0_Output_Status
```

Module TM3AQ2 (2 sorties)

Obtention de l'état de la première sortie QW0

- *StatusOffset* = 0 (0 entrée x 2)
- *StatusSize* = 1 (1 état à lire)
- *pStatusBuffer* doit être d'au moins 1 octet

```
VAR
TM3AM6_Channels_1_2_Input_Status: ARRAY[1..2] OF BYTE;
END_VAR
TM3AM6 is on position 1
Status of channel 1 is at offset 1
We read 2 consecutive channels
TM3_GetModuleInternalStatus(1, 1, 2, ADR(TM3AM6_Channels_1_
2_Input_Status));
status of channel 1 is in TM3AM6_Channels_1_2_Input_Status
[1]
status of channel 2 is in TM3AM6_Channels_1_2_Input_Status
[2]
```

Module TM3AM6 (4 entrées, 2 sorties)

Obtention de l'état des entrées IW1 et IW2 (IW0 étant la première)

- *StatusOffset* = 1 (1 pour ignorer l'état IW0)
- *StatusSize* = 2 (2 états à lire)
- *pStatusBuffer* doit être d'au moins 2 octets

Types de données système du TM3

Contenu de ce chapitre

TM3_BUS_PARAM_ID : Diagnostic du module d'extension TM3 66
TM3_BUS_W_IOBUSERRMOD : Mode d'erreur du bus TM3 66
TM3_BUS_W_IOBUSINIT : Erreur de la réponse de configuration. 67
TM3_BUS_W : Variables système de bus TM3 67
TM3_ERR_CODE : Codes d'erreur détectés du module d'extension TM3 67
TM3_MODULE_R_ARRAY_TYPE : Type de tableau de lecture du module d'extension TM3 68
TM3_MODULE_STATE : Codes d'état du module d'extension TM3 68

TM3_BUS_PARAM_ID : Diagnostic du module d'extension TM3

Description du type Énumération

Le type de données énumération *TM3_BUS_PARAM_ID* contient les valeurs suivantes :

Énumérateur	Type	Valeur	Commentaire
ID_TM3_MODULE_DIAG	DWORD	10001 déc	-
ID_TM3_BUS_W	DWORD	10002 déc	-

TM3_BUS_W_IOBUSERRMOD : Mode d'erreur du bus TM3

Description du type Énumération

Le type de données énumération *TM3_BUS_W_IOBUSERRMOD* contient les valeurs suivantes :

Énumérateur	Valeur	Commentaire
IOBUS_ERR_ACTIVE	00 hex	Mode actif. Le contrôleur logique arrête tous les échanges d'E/S sur le bus TM3 en cas de détection d'une erreur permanente. Reportez-vous à la section Description générale de la configuration des E/S (voir Modicon M262 - Logic/Motion Controller - Guide de programmation).
IOBUS_ERR_PASSIVE	01 hex	Mode passif. Les échanges d'E/S se poursuivent sur le bus TM3, même si une erreur est détectée.

TM3_BUS_W_IOBUSINIT : Erreur de la réponse de configuration.

Description du type Énumération

Le type de données énumération *TM3_BUS_W_IOBUSINIT* contient les valeurs suivantes :

Énumérateur	Type	Valeur
CMD_INIT_OFF	WORD	00 hex
CMD_INIT_ON	WORD	01 hex

TM3_BUS_W : Variables système de bus TM3

Structure de la variable

Ce tableau décrit les paramètres des variables système *TM3_BUS_W* (type *TM3_BUS_W_STRUCT*) :

Nom de la variable	Type	Commentaire
q_wIOBusErrPassiv	*TM3_BUS_W_IOBUSERRMOD*	Lorsqu'elle est définie sur *ERR_ACTIVE* (valeur par défaut), les erreurs de bus détectées sur les modules d'extension TM3 arrêtent les échanges d'E/S. Lorsque la valeur est *ERR_PASSIVE*, le traitement passif des erreurs d'E/S est utilisé : le contrôleur essaie de poursuivre les échanges de bus de données.
q_wIOBusRestart	*TM3_BUS_W_IOBUSINIT*	Lorsqu'elle est mise à 1, redémarre le bus d'extension d'E/S. Cela n'est nécessaire que lorsque *q_wIOBusErrPassiv* est réglé sur *ERR_ACTIVE* et au moins un bit de *TM3_MODULE_R[i] .i_wModuleState* est réglé sur *TM3_BUS_ERROR*.

Pour plus d'informations, voir Description générale de la configuration des E/S (voir Modicon M262 – Logic/Motion Controller – Guide de programmation).

TM3_ERR_CODE : Codes d'erreur détectés du module d'extension TM3

Bibliothèque et espace de noms

Nom de la bibliothèque : **TM3System**

Espace de noms : **SEC_TM3Sys**

Description du type Énumération

Le type de données énumération *TM3_ERR_CODE* contient les valeurs suivantes :

Énumérateur	Valeur	Commentaire
TM3_NO_ERR	00 hex	Le dernier échange du bus avec le module d'extension a réussi.
TM3_ERR_FAILED	01 hex	Erreur détectée suite à l'échec du dernier échange du bus avec le module d'extension.
TM3_ERR_PARAMETER	02 hex	Erreur de paramètre détectée dans le dernier échange du bus avec le module.
TM3_ERR_COK	03 hex	Erreur matérielle temporaire ou permanente détectée sur l'un des modules d'extension TM3.
TM3_ERR_BUS	04 hex	Erreur de bus détectée dans le dernier échange du bus avec le module d'extension.

TM3_MODULE_R_ARRAY_TYPE : Type de tableau de lecture du module d'extension TM3

Bibliothèque et espace de noms

Nom de la bibliothèque : **TM3System**

Espace de noms : **SEC_TM3Sys**

Description

Le *TM3_MODULE_R_ARRAY_TYPE* est un tableau de 0-13 *TM3_MODULE_R_STRUCT*.

TM3_MODULE_STATE : Codes d'état du module d'extension TM3

Bibliothèque et espace de noms

Nom de la bibliothèque : **TM3System**

Espace de noms : **SEC_TM3Sys**

Description du type Énumération

Le type de données énumération *TM3_MODULE_STATE* contient les valeurs suivantes :

Énumérateur	Valeur	Commentaire
TM3_EMPTY	00 hex	Absence de module.
TM3_CONF_ERROR	01 hex	Le module d'extension physique ne correspond pas à celui configuré dans le logiciel.
TM3_BUS_ERROR	02 hex	Erreur de bus détectée dans le dernier échange avec le module.
TM3_OK	03 hex	Le dernier échange du bus avec ce module a réussi.
TM3_POWER_SUPPLY_ ERROR	04 hex	Erreur liée à l'alimentation externe du module.
TM3_MISSING_OPT_MOD	05 hex	Module facultatif non physiquement présent.

Système TMS

Contenu de cette partie

Variables système du TMS 71
Types de données système du TMS 73

Introduction

Cette section décrit la bibliothèque système des TMS

Variables système du TMS

Contenu de ce chapitre

TMS_BUS_DIAG_R : Codes d'erreur de diagnostic du bus TMS.....71
TMS_MODULE_DIAG_R : Codes d'erreur de diagnostic des modules d'extension TMS 71

Présentation générale

Ce chapitre décrit les variables système, page 14 disponibles dans la bibliothèque système du TMS.

TMS_BUS_DIAG_R : Codes d'erreur de diagnostic du bus TMS

Bibliothèque et espace de noms

Nom de la bibliothèque : **TMSSystem**

Espace de noms : **TMS**

Structure de la variable

Ce tableau décrit les paramètres des variables système *TMS_BUS_DIAG_R* (type *STRUCT_TMS_BUS_DIAG*) :

Énumérateur	Type	Commentaire
ConfState	UNIT	Décrit l'état de la configuration du TMS : • 0 : Absence de configuration • 1 : Configuration non valide (non-concordance de module) • 2 : Configuration valide, mais booster de puissance inactif • 3 : Configuration valide et appliquée
NbModules	UNIT	Indique le nombre de modules détectés sur le bus.

TMS_MODULE_DIAG_R : Codes d'erreur de diagnostic des modules d'extension TMS

Bibliothèque et espace de noms

Nom de la bibliothèque : **TMSSystem**

Espace de noms : **TMS**

Introduction

La variable système *TMS_MODULE_DIAG_R* (type *TMS_MODULE_R_ARRAY_TYPE*) est un tableau de *STRUCT_TMS_MODULE_DIAG*.

Structure de la variable

Ce tableau décrit les paramètres de *STRUCT_TMS_MODULE_DIAG* :

Énumérateur	Type	Commentaire
Name	STRING(15)	Nom du module d'extension TMS.
MajorType	WORD	Type de module d'extension TMS.
SubType	WORD	Sous-type de module d'extension TMS.
Version	STRING(15)	Version du micrologiciel du module d'extension TMS.
ModuleState	*TMS_MODULE_STATE*, page 73	Etat du module d'extension TMS.
IpState	*TMS_IP_STATE*, page 73	Accessibilité TCP/IP du module d'extension TMS depuis le contrôleur sur le **bus COM**.
PixCmdState	*TMS_PIXCMD_STATE*, page 74	Etat du module d'extension.

Types de données système du TMS

Contenu de ce chapitre

TMS_IP_STATE : État IP des modules d'extension TMS 73
TMS_MODULE_STATE : Codes d'état des modules d'extension TMS..... 73
TMS_PIXCMD_STATE : État PIXCMD du module d'extension TMS 74

TMS_IP_STATE : État IP des modules d'extension TMS

Bibliothèque et espace de noms

Nom de la bibliothèque : **TMSSystem**

Espace de noms : **TMS**

Description

Le type de données énumération *TMS_IP_STATE* contient les valeurs suivantes :

Nom	Type	Valeur	Commentaire
TMS_IP_PING_SUCCESS	DWORD	0	Interface IP configurée.
TMS_IP_CONFIG_CMD_ ERROR	DWORD	1	Echec de l'envoi de configuration.
TMS_IP_CONFIG_RESP_WAIT	DWORD	2	En attente de réponse de configuration (transitoire).
TMS_IP_CONFIG_RESP_ ERROR	DWORD	3	Erreur de la réponse de configuration.
TMS_IP_CONFIG_RESP_NONE	DWORD	4	Aucune réponse de configuration.
TMS_IP_CONFIG_SUCCESS	DWORD	5	Configuration réussie.
TMS_IP_PING_CMD_ERROR	DWORD	6	Echec de l'envoi de commande ping.
TMS_IP_PING_RESP_WAIT	DWORD	7	En attente de réponse à la commande ping (transitoire).
TMS_IP_PING_RESP_ERROR	DWORD	8	Erreur de la réponse à la commande ping.
TMS_IP_PING_RESP_NONE	DWORD	9	Aucune réponse à la commande ping.
TMS_IP_NOT_CONFIGURED	DWORD	10	Interface IP non configurée.

TMS_MODULE_STATE : Codes d'état des modules d'extension TMS

Bibliothèque et espace de noms

Nom de la bibliothèque : **TMSSystem**

Espace de noms : **TMS**

Description du type Énumération

Le type de données énumération *TMS_MODULE_STATE* contient les valeurs suivantes :

Nom	Type	Valeur	Commentaire
TMS_MODULE_POWERED	DWORD	0	Module sous tension.
TMS_MODULE_INITIALIZED	DWORD	1	Module initialisé et détecté.
TMS_MODULE_CONFIGURED	DWORD	2	Module configuré et fonctionnant normalement.
TMS_MODULE_EXCHANGE_FAULT	DWORD	3	Délai de détection du module expiré.
TMS_MODULE_ERROR	DWORD	4	Erreur détectée par le module.
TMS_MODULE_HEALTH_SEND_FAULT	DWORD	5	Echec de l'envoi du rapport d'intégrité du module.
TMS_MODULE_RCV_TIMEOUT	DWORD	6	Dépassement du délai de réception du module.
TMS_MODULE_RCV_MISC	DWORD	7	Erreur de réception du module autre qu'un dépassement de délai.
TMS_MODULE_RESP_ERR	DWORD	8	Erreur de la réponse du module.
TMS_MODULE_DISCOVERY	DWORD	9	Erreur de détection du module.

TMS_PIXCMD_STATE : État PIXCMD du module d'extension TMS

Bibliothèque et espace de noms

Nom de la bibliothèque : **TMSSystem**

Espace de noms : **TMS**

Description

Le type de données énumération *TMS_PIXCMD_STATE* contient les valeurs suivantes :

Nom	Type	Valeur	Commentaire
TMS_PIXCMD_EXCHING	DWORD	0	Echange en cours.
TMS_PIXCMD_CONFIG_NONE	DWORD	1	Pas de configuration.
TMS_PIXCMD_CONFIG_CMD_ERROR	DWORD	2	Échec de l'envoi de configuration.
TMS_PIXCMD_CONFIG_RESP_WAIT	DWORD	3	En attente de réponse de configuration (transitoire).
TMS_PIXCMD_CONFIG_RESP_ERROR	DWORD	4	Erreur de la réponse de configuration.
TMS_PIXCMD_CONFIG_ONLY	DWORD	5	Succès de la configuration, aucun échange pour l'instant.
TMS_PIXCMD_CONFIG_SUCCESS	DWORD	6	Succès de la configuration (transitoire).
TMS_PIXCMD_ENABLE_CMD_ERROR	DWORD	7	Erreur de commande d'activation.
TMS_PIXCMD_ENABLE_RESP_WAIT	DWORD	8	En attente de réponse à la commande d'activation (transitoire).
TMS_PIXCMD_ENABLE_RESP_ERROR	DWORD	9	Erreur de la réponse à la commande d'activation.
TMS_PIXCMD_EXCH_ERROR	DWORD	10	Erreur d'échange.
TMS_PIXCMD_DISABLING	DWORD	11	Echange en cours de désactivation (transitoire)
TMS_PIXCMD_DISABLED	DWORD	12	Echange désactivé.

Annexes

Contenu de cette partie

Représentation des fonctions et blocs fonction 77

Vue d'ensemble

Cette annexe reprend des extraits du guide de programmation aux fins de faciliter la compréhension technique de la documentation de la bibliothèque.

Représentation des fonctions et blocs fonction

Contenu de ce chapitre

Différences entre fonction et bloc fonction 77
Utilisation d'une fonction ou d'un bloc fonction en langage IL 78
Utilisation d'une fonction ou d'un bloc fonction en langage ST 81

Présentation générale

Chaque fonction peut être représentée dans les langages suivants :

- IL : Liste d'instructions
- ST : Texte structuré
- LD : Schéma à contacts
- FBD : Langage en blocs fonction
- CFC : Diagramme fonctionnel continu

Ce chapitre fournit des exemples de représentations de fonctions et blocs fonction et explique comment les utiliser dans les langages IL et ST.

Différences entre fonction et bloc fonction

Fonction

Une fonction :

- est une POU (Program Organization Unit ou unité organisationnelle de programme) qui renvoie un résultat immédiat ;
- est directement appelée par son nom (et non par une instance) ;
- ne conserve pas son état entre deux appels ;
- peut être utilisée en tant qu'opérande dans des expressions.

Exemples : opérateurs booléens (`AND`), calculs, conversions (`BYTE_TO_INT`)

Bloc fonction

Un bloc fonction :

- est une POU qui renvoie une ou plusieurs sorties ;
- doit être appelé par une instance (copie de bloc fonction avec nom et variables dédiés).
- Chaque instance conserve son état (sorties et variables internes) entre deux appels à partir d'un bloc fonction ou d'un programme.

Exemples : temporisateurs, compteurs

Dans l'exemple, `Timer_ON` est une instance du bloc fonction `TON` :

```
PROGRAM MyProgram_ST
VAR
    Timer_ON: TON;  // Function Block Instance
    Timer_RunCd: BOOL;
    Timer_PresetValue: TIME := T#5S;
    Timer_Output: BOOL;
    Timer_ElapsedTime: TIME;
END_VAR
```

```
Timer_ON(
    IN:=Timer_RunCd,
    PT:=Timer_PresetValue,
    Q=>Timer_Output,
    ET=>Timer_ElapsedTime);
```

Utilisation d'une fonction ou d'un bloc fonction en langage IL

Informations générales

Cette partie explique comment mettre en œuvre une fonction et un bloc fonction en langage IL.

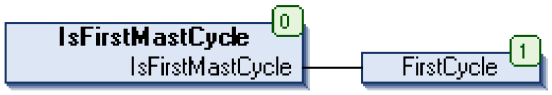
Les fonctions `IsFirstMastCycle` et `SetRTCDrift` ainsi que le bloc fonction `TON` sont utilisés à titre d'exemple pour illustrer les implémentations.

Utilisation d'une fonction en langage IL

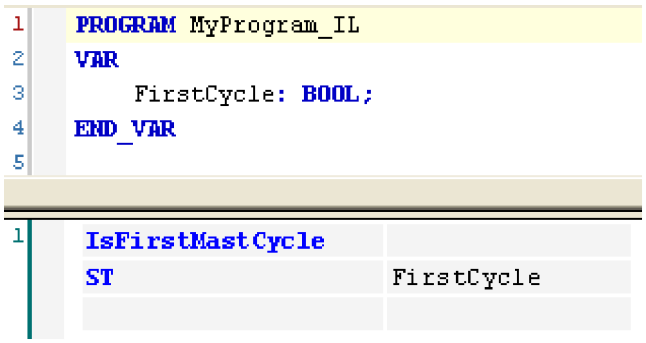
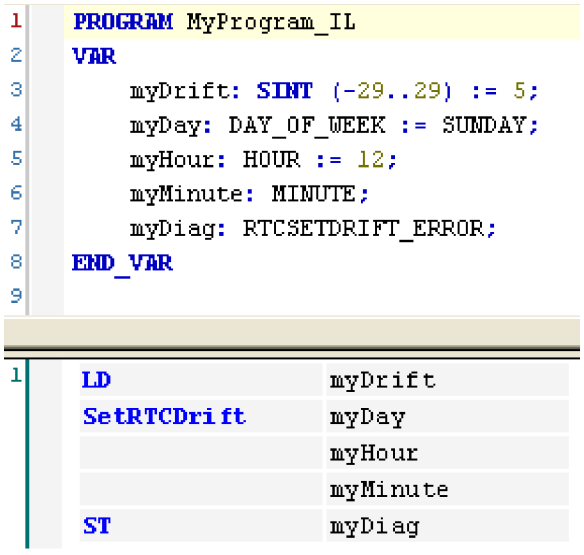
La procédure suivante explique comment insérer une fonction en langage IL :

Etape	Action
1	Ouvrez ou créez une POU en langage IL (Instruction List, ou liste d'instructions). **NOTE:** La procédure de création d'une POU n'est pas détaillée ici. Pour plus d'informations, consultez la documentation Ajout et appel de POU (voir EcoStruxure Automation Expert - Motion / EcoStruxure Machine Expert – Guide de programmation).
2	Créez les variables nécessaires à la fonction.
3	Si la fonction possède une ou plusieurs entrées, chargez la première entrée en utilisant l'instruction LD.
4	Insérez une nouvelle ligne en dessous et : • saisissez le nom de la fonction dans la colonne de l'opérateur (champ de gauche), ou • utilisez l'**Aide à la saisie** pour sélectionner la fonction (sélectionnez **Insérer module(s)** dans le menu contextuel).
5	Si la fonction a plus de 1 entrée et que l'aide à la saisie est utilisée, le nombre requis de lignes est automatiquement créé avec `???` dans les champs situés à droite. Remplacez les `???` par la valeur ou la variable appropriée compte tenu de l'ordre des entrées.
6	Insérez une nouvelle ligne pour stocker le résultat de la fonction dans la variable appropriée : saisissez l'instruction ST dans la colonne de l'opérateur (champ de gauche) et le nom de la variable dans le champ de droite.

Pour illustrer la procédure, utilisons les fonctions `IsFirstMastCycle` (sans paramètre d'entrée) et `SetRTCDrift` (avec paramètres d'entrée) représentées graphiquement ci-après :

Fonction	Représentation graphique
sans paramètre d'entrée : `IsFirstMastCycle`	
avec paramètres d'entrée : `SetRTCDrift`	

En langage IL, le nom de la fonction est utilisé directement dans la colonne de l'opérateur :

Fonction	Représentation dans l'éditeur IL de POU
Exemple en IL d'une fonction sans paramètre d'entrée : `IsFirstMastCycle`	
Exemple en IL d'une fonction avec paramètres d'entrée : `SetRTCDrift`	

Utilisation d'un bloc fonction en langage IL

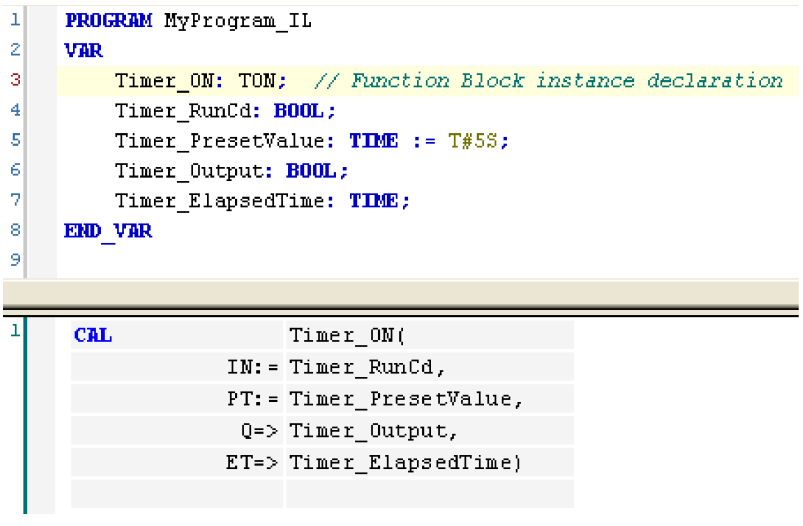
La procédure suivante explique comment insérer un bloc fonction en langage IL :

Etape	Action
1	Ouvrez ou créez une POU en langage IL (Instruction List, ou liste d'instructions). **NOTE:** La procédure de création d'une POU n'est pas détaillée ici. Pour plus d'informations, consultez la documentation Ajout et appel de POU (voir EcoStruxure Automation Expert - Motion / EcoStruxure Machine Expert – Guide de programmation).
2	Créez les variables nécessaires au bloc fonction (y compris le nom de l'instance).
3	L'appel de blocs fonction nécessite l'utilisation d'une instruction `CAL` : • Utilisez l'**Aide à la saisie** pour sélectionner le bloc fonction (cliquez avec le bouton droit de la souris et sélectionnez **Insérer module(s)** dans le menu contextuel). • L'instruction `CAL` et les E/S nécessaires sont automatiquement créées. Chaque paramètre (E/S) est une instruction : • Les valeurs des entrées sont définies à l'aide de ":=". • Les valeurs des sorties sont définies à l'aide de "=>".
4	Dans le champ `CAL` à droite, remplacez les `???` par le nom de l'instance.
5	Remplacez les autres `???` par une variable ou une valeur immédiate appropriée.

Pour illustrer la procédure, utilisons le bloc fonction `TON` représenté graphiquement ci-après :

Bloc fonction	Représentation graphique
`TON`	

En langage IL, le nom du bloc fonction est utilisé directement dans la colonne de l'opérateur :

Bloc fonction	Représentation dans l'éditeur IL de POU
`TON`	

Utilisation d'une fonction ou d'un bloc fonction en langage ST

Informations générales

Cette partie explique comment mettre en œuvre une fonction ou un bloc fonction en langage ST.

La fonction `SetRTCDrift` et le bloc fonction `TON` sont utilisés comme exemples pour illustrer les implémentations.

Utilisation d'une fonction en langage ST

La procédure suivante explique comment insérer une fonction en langage ST :

Etape	Action
1	Ouvrez ou créez une POU en langage ST (Structured Text ou Littéral structuré). **NOTE:** La procédure de création d'une POU n'est pas détaillée ici. Pour plus d'informations, consultez la documentation Ajout et appel de POU (voir EcoStruxure Automation Expert - Motion / EcoStruxure Machine Expert – Guide de programmation).
2	Créez les variables nécessaires à la fonction.
3	Utilisez la syntaxe générale dans l'**éditeur ST de POU** pour la représentation en langage ST d'une fonction. La syntaxe générale est la suivante : `FunctionResult:= FunctionName(VarInput1, VarInput2,.. VarInputx);`

Pour illustrer la procédure, examinez la fonction `SetRTCDrift` représentée graphiquement ci-après :

Fonction	Représentation graphique
`SetRTCDrift`	

La représentation en langage ST de cette fonction est la suivante :

Fonction	Représentation dans l'éditeur ST de POU

`SetRTCDrift`	`PROGRAM MyProgram_ST`
	`VAR`
	`myDrift: SINT (-29..+29) := 5;`
	`myDay: sec.DAY_OF_WEEK := SUNDAY;`
	`myHour: sec.HOUR := 12;`
	`myMinute: sec.MINUTE;`
	`myRTCAdjust: sec.RTCDRIFT_ERROR;`
	`END_VAR`
	`myRTCAdjust:= SetRTCDrift(myDrift, myDay, myHour, myMinute);`

Utilisation d'un bloc fonction en langage ST

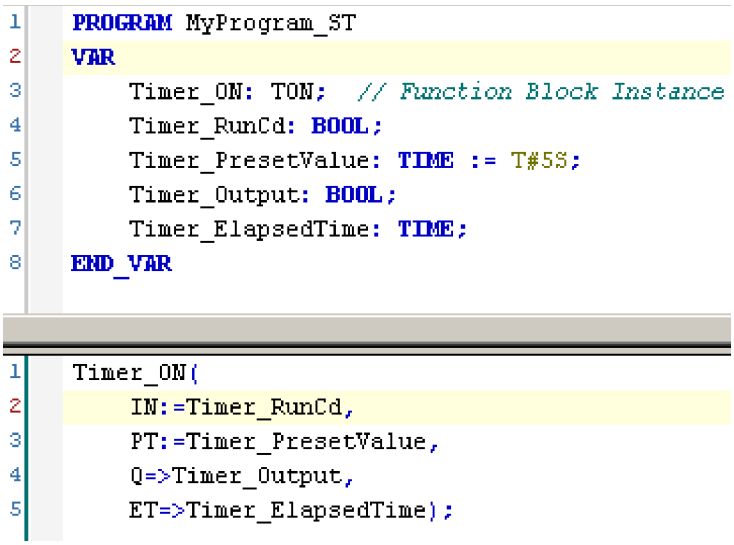
La procédure suivante explique comment insérer un bloc fonction en langage ST :

Etape	Action
1	Ouvrez ou créez une POU en langage ST (Structured Text ou Littéral structuré). **NOTE:** La procédure de création d'une POU n'est pas détaillée ici. Pour plus d'informations, consultez la documentation Ajout et appel de POU (voir EcoStruxure Automation Expert - Motion / EcoStruxure Machine Expert – Guide de programmation).
2	Créez les variables d'entrée, les variables de sortie et l'instance requises pour le bloc fonction : • Les variables d'entrée sont les paramètres d'entrée requis par le bloc fonction. • Les variables de sortie reçoivent la valeur renvoyée par le bloc fonction.
3	Utilisez la syntaxe générale dans l'**éditeur ST de POU** pour la représentation en langage ST d'un bloc fonction. La syntaxe générale est la suivante : `FunctionBlock_InstanceName(Input1:=VarInput1, Input2:=VarInput2,... Ouput1=>VarOutput1, Ouput2=>VarOutput2,...);`

Pour illustrer la procédure, utilisons le bloc fonction `TON` représenté graphiquement ci-après :

Bloc fonction	Représentation graphique
TON	

Le tableau suivant montre plusieurs exemples d'appel de bloc fonction en langage ST :

Bloc fonction	Représentation dans l'éditeur ST de POU
TON	

Glossaire

A

adresse MAC:

(*media access control*) Nombre unique sur 48 bits associé à un élément matériel spécifique. L'adresse MAC est programmée dans chaque carte réseau ou équipement lors de la fabrication.

anciens projets:

Projets d'application créés avec SoMachine, SoMachine Motion ou une version précédente de EcoStruxure Machine Expert.

application de démarrage:

(*boot application*). Fichier binaire qui contient l'application. En général, il est stocké dans le contrôleur et permet à ce dernier de démarrer sur l'application que l'utilisateur a générée.

application:

Programme comprenant des données de configuration, des symboles et de la documentation.

ARRAY:

Agencement systématique d'objets de données d'un même type sous la forme d'un tableau défini dans la mémoire d'un Logic Controller. La syntaxe est la suivante : `ARRAY [<dimension>] OF <Type>`

Exemple 1 : `ARRAY [1..2] OF BOOL` est un tableau à 1 dimension comportant 2 éléments de type `BOOL`.

Exemple 2 : `ARRAY [1..10, 1..20] OF INT` est un tableau à 2 dimensions comportant 10 x 20 éléments de type `INT`.

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

BOOL:

(*booléen*) Type de données informatique standard. Une variable de type `BOOL` peut avoir l'une des deux valeurs suivantes : 0 (`FALSE`), 1 (`TRUE`). Un bit extrait d'un mot est de type `BOOL` ; par exemple, `%MW10.4` est le cinquième bit d'un mot mémoire numéro 10.

BOOTP:

(*bootstrap protocol*). Protocole réseau UDP qu'un client réseau peut utiliser pour obtenir automatiquement une adresse IP (et éventuellement d'autres données) à partir d'un serveur. Le client s'identifie auprès du serveur à l'aide de son adresse MAC. Le serveur, qui gère un tableau préconfiguré des adresses MAC des équipements client et des adresses IP associées, envoie au client son adresse IP préconfigurée. A l'origine, le protocole BOOTP était utilisé pour amorcer à distance les hôtes sans lecteur de disque à partir d'un réseau. Le processus BOOTP affecte une adresse IP de durée illimitée. Le service BOOTP utilise les ports UDP 67 et 68.

C

CAN:

Acronyme de *Controller Area Network*. Protocole (ISO 11898) pour réseaux de bus série qui assure l'interconnexion d'équipements intelligents (de différentes marques) dans des systèmes intelligents pour les applications industrielles en temps réel. Développé initialement pour l'industrie automobile, le protocole CAN est désormais utilisé dans tout un éventail d'environnements de surveillance d'automatisme.

CFC:

Acronyme de *continuous function chart* (diagramme fonctionnel continu). Langage de programmation graphique (extension de la norme IEC 61131-3) basé sur le langage de diagramme à blocs fonction et qui fonctionne comme un diagramme de flux. Toutefois, il n'utilise pas de réseaux et le positionnement libre des éléments graphiques est possible, ce qui permet les boucles de retour. Pour chaque bloc, les entrées se situent à gauche et les sorties à droite. Vous pouvez lier les sorties de blocs aux entrées d'autres blocs pour créer des expressions complexes.

chaîne:

Variable composée d'une série de caractères ASCII.

chien de garde:

Temporisateur spécial utilisé pour garantir que les programmes ne dépassent pas le temps de scrutation qui leur est alloué. Le chien de garde est généralement réglé sur une valeur supérieure au temps de scrutation et il est remis à 0 à la fin de chaque cycle de scrutation. Si le temporisation chien de garde atteint la valeur prédéfinie (par exemple, lorsque le programme est bloqué dans une boucle sans fin) une erreur est déclarée et le programme s'arrête.

configuration :

Agencement et interconnexions des composants matériels au sein d'un système, ainsi que les paramètres matériels et logiciels qui déterminent les caractéristiques de fonctionnement du système.

CRC:

Contrôle de redondance cyclique. Méthode utilisée pour déterminer la validité d'une transmission de communication. La transmission contient un champ de bits qui constitue un total de contrôle. Le message est utilisé pour le calcul de ce total de contrôle par l'émetteur en fonction du contenu du message. Les noeuds récepteurs recalculent ensuite ce champ de la même manière. Tout écart entre les deux calculs de CRC indique que le message émis et le message reçu sont différents.

D

DHCP:

Acronyme de *dynamic host configuration protocol*. Extension avancée du protocole BOOTP. Bien que DHCP soit plus avancé, DHCP et BOOTP sont tous les deux courants. (DHCP peut gérer les requêtes de clients BOOTP.)

DWORD:

Abréviation de *double word*, mot double. Codé au format 32 bits.

E

E/S:

Entrée/sortie

EtherNet/IP:

Acronyme de *Ethernet Industrial Protocol*, protocole industriel Ethernet. Protocole de communication ouvert pour les solutions d'automatisation de la production dans les systèmes industriels. EtherNet/IP est une famille de réseaux mettant en œuvre le protocole CIP au niveau des couches supérieures. L'organisation ODVA spécifie qu'EtherNet/IP permet une adaptabilité générale et une indépendance des supports.

Ethernet:

Technologie de couche physique et de liaison de données pour les réseaux locaux (LANs) également appelée IEEE 802.3.

F

FB:

Acronyme de *function block*, bloc fonction. Mécanisme de programmation commode qui consolide un groupe d'instructions de programmation visant à effectuer une action spécifique et normalisée telle que le contrôle de vitesse, le contrôle d'intervalle ou le comptage. Un bloc fonction peut comprendre des données de configuration, un ensemble de paramètres de fonctionnement interne ou externe et généralement une ou plusieurs entrées et sorties de données.

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.

fonction:

Unité de programmation possédant 1 entrée et renvoyant 1 résultat immédiat. Contrairement aux blocs fonction (FBs), une fonction est appelée directement par son nom (et non via une instance), elle n'a pas d'état persistant d'un appel au suivant et elle peut être utilisée comme opérande dans d'autres expressions de programmation.

Exemples : opérateurs booléens (AND), calculs, conversion (BYTE_TO_INT).

G

GVL:

Acronyme de *Global Variable List*, liste de variables globales. Permet de gérer les variables globales d'un projet EcoStruxure Machine Expert.

H

hex:

(*hexadécimal*)

I

ID:

(*identificateur/identification*)

IEC 61131-3:

Partie 3 d'une norme en 3 parties de l'IEC pour les équipements d'automatisation industriels. La norme IEC 61131-3 traite des langages de programmation des contrôleurs. Elle définit 2 normes pour la programmation graphique et 2 normes pour la programmation textuelle. Les langages de programmation graphiques sont le schéma à contacts (LD) et le langage à blocs fonction (FBD). Les langages textuels comprennent le texte structuré (ST) et la liste d'instructions (IL).

IEC:

Acronyme de *International Electrotechnical Commission*, Commission Electrotechnique Internationale (CEI). Organisation internationale non gouvernementale à but non lucratif, qui rédige et publie les normes internationales en matière d'électricité, d'électronique et de domaines connexes.

IEEE 802.3:

Ensemble de normes IEEE définissant la couche physique et la sous-couche MAC de la couche de liaison de données de l'Ethernet câblé.

IL:

Acronyme de *instruction list*, liste d'instructions. Un programme écrit en langage IL est composé d'instructions textuelles qui sont exécutées séquentiellement par le contrôleur. Chaque instruction comprend un numéro de ligne, un code d'instruction et un opérande (voir la norme IEC 61131-3).

INT:

Abréviation de *integer*, nombre entier codé sur 16 bits.

IP:

Acronyme de *Internet Protocol*, protocole Internet. Le protocole IP fait partie de la famille de protocoles TCP/IP, qui assure le suivi des adresses Internet des équipements, achemine les messages sortants et reconnaît les messages entrants.

L

langage en blocs fonctionnels:

Un des 5 langages de programmation de logique ou de commande pris en charge par la norme IEC 61131-3 pour les systèmes de commande. FBD est un langage de programmation orienté graphique. Il fonctionne avec une liste de réseaux où chaque réseau contient une structure graphique de zones et de lignes de connexion représentant une expression logique ou arithmétique, un appel de bloc fonction ou une instruction de retour.

LD:

Acronyme de *ladder diagram*, schéma à contacts. Représentation graphique des instructions d'un programme de contrôleur, avec des symboles pour les contacts, les bobines et les blocs dans une série de réseaux exécutés séquentiellement par un contrôleur (voir IEC 61131-3).

LWORD:

Abréviation de *long word*, mot long. Type de données codé sur 64 bits.

M

MAST:

Tâche de processeur exécutée par le biais de son logiciel de programmation. La tâche MAST comprend deux parties :

- **IN :** les entrées sont copiées dans la section IN avant l'exécution de la tâche MAST.
- **OUT :** les sorties sont copiées dans la section OUT après l'exécution de la tâche MAST.

Modbus:

Protocole qui permet la communication entre de nombreux équipements connectés au même réseau.

%MW:

Selon la norme IEC, %MW représente un registre de mots mémoire (par exemple, un objet langage de type mot mémoire).

N

NVM:

(Non-volatile memory) Mémoire non volatile qui peut être écrasée. Elle est stockée dans une puce EEPROM spéciale, effaçable et reprogrammable.

O

octet:

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

P

PLC:

Acronyme de *programmable logic controller*, Logic Controller programmable. Ordinateur industriel utilisé pour automatiser des processus de fabrication et autres processus électromécaniques. Les PLCs diffèrent des ordinateurs courants par le fait qu'ils sont conçus pour utiliser plusieurs tableaux d'entrées et de sorties et pour accepter des conditions de choc, de vibration, de température et d'interférences électriques plus rudes.

POU:

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

programme:

Composant d'une application constitué de code source compilé qu'il est possible d'installer dans la mémoire d'un Logic Controller.

protocole:

Convention ou définition standard qui contrôle ou permet la connexion, la communication et le transfert de données entre 2 systèmes informatiques et leurs équipements.

R

réseau de commande:

Réseau incluant des contrôleurs logiques, des systèmes SCADA, des PC, des IHM, des commutateurs, etc.

Deux types de topologies sont pris en charge :

- à plat : tous les modules et équipements du réseau appartiennent au même sous-réseau.
- à 2 niveaux : le réseau est divisé en un réseau d'exploitation et un réseau intercontrôleurs.

Ces deux réseaux peuvent être indépendants physiquement, mais ils sont généralement liés par un équipement de routage.

réseau:

Système d'équipements interconnectés qui partageant un chemin de données et un protocole de communications communs.

run:

Commande qui ordonne au contrôleur de scruter le programme d'application, lire les entrées physiques et écrire dans les sorties physiques en fonction de la solution de la logique du programme.

S

%:

Selon la norme IEC, % est un préfixe qui identifie les adresses mémoire internes des contrôleurs logiques pour stocker la valeur de variables de programme, de constantes, d'E/S, etc.

STOP:

Commande ordonnant au contrôleur de cesser d'exécuter un programme d'application.

ST:

Acronyme de *structured text*, texte structuré. Langage composé d'instructions complexes et d'instructions imbriquées (boucles d'itération, exécutions conditionnelles, fonctions). Le langage ST est conforme à la norme IEC 61131-3.

T

tâche:

Ensemble de sections et de sous-programmes, exécutés de façon cyclique ou périodique pour la tâche MAST, ou périodique pour la tâche FAST.

Une tâche présente un niveau de priorité et des entrées et sorties du contrôleur lui sont associées. Ces E/S sont actualisées par rapport à la tâche.

Un contrôleur peut comporter plusieurs tâches.

TCP:

Acronyme de *transmission control protocol*, protocole de contrôle de transmission. Protocole de couche de transport basé sur la connexion qui assure la transmission de données simultanée dans les deux sens. Le protocole TCP fait partie de la suite de protocoles TCP/IP.

U

UDINT:

Abréviation de *unsigned double integer*, entier double non signé. Valeur codée sur 32 bits.

UINT:

Abréviation de *unsigned integer*, entier non signé. Valeur codée sur 16 bits.

V

variable non localisée:

Variable qui n'a pas d'adresse (voir *variable localisée*).

variable système:

Variable qui fournit des données de contrôleur et des informations de diagnostic et permet d'envoyer des commandes au contrôleur.

variable:

Unité de mémoire qui est adressée et modifiée par un programme.

W

WORD:

Type de données codé sur 16 bits.

Index

B

blocs fonction
DataFileCopy....39
ExecuteScript....41
FB_CheckAllowedControllerMacAddr....37
FB_ControlClone....38
FB_GetFreeDiskSpace....34
FB_GetLabel....35
FB_GetTotalDiskSpace....36

C

commandes de copie de fichier
DataFileCopy....39
commandes de script
ExecuteScript....41
cycle
GetExternalEventValue....30
IsFirstMastColdCycle....28
IsFirstMastCycle....28
IsFirstMastWarmCycle....30

D

DataFileCopy
copie de données vers ou depuis un fichier....39
DataFileCopyError
types de données....50
DataFileCopyLocation
type de données....50
DAY_OF_WEEK....33

E

entrée/sortie
HasForcedIo....27
E/S intégrée
GetImmediateFastInput....25
E/S intégrées
PhysicalWriteFastOutputs....31
état du bus du module TM3
TM3_GetModuleBusStatus....62
état interne du module TM3
TM3_GetModuleInternalStatus....63
ETH_R
variable système....21
ETH_R_FRAME_PROTOCOL
type de données....51
ETH_R_IP_MODE
type de données....52
ETH_R_IPFORWARDING
type de données....52
ETH_R_ITF_STRUCT
type de données....52
ETH_R_PORT_DUPLEX_STATUS
type de données....54
ETH_R_PORT_IP_STATUS
type de données....54
ETH_R_PORT_LINK_STATUS
type de données....54
ETH_R_PORT_SPEED
type de données....55
ETH_R_RUN_IDLE
type de données....55
ETH_R_STRUCT....21
ETH_W
variable système....24
ETH_W_STRUCT....24
ExecuteScript
exécution de commandes de script....41
ExecuteScriptError
type de données....51
EXTEVT_VAL_RES....30

F

FB_CheckAllowedControllerMacAddr
bloc fonction....37
FB_ControlClone
bloc fonction....38
FB_GetFreeDiskSpace
bloc fonction....34
FB_GetLabel
bloc fonction....35
FB_GetTotalDiskSpace
bloc fonction....36
fonctions
différences entre fonction et bloc fonction....77
utilisation d'une fonction ou d'un bloc fonction en langage IL....78
utilisation d'une fonction ou d'un bloc fonction en langage ST....81

G

GetExternalEventValue
obtenir la valeur actuelle d'un événement externe....30
GetImmediateFastInput
obtention de la valeur d'une entrée rapide....25
GetRtc
obtention de la valeur de l'horodateur (RTC)....26

H

HasForcedIo
indiquer si une entrée/sortie est forcée....27
horloge en temps réel
GetRtc....26
SetRTCDrift....32
HOUR....33

I

IMMEDIATE_ERR_TYPE
type de données....56
IsFirstMastColdCycle
premier cycle de démarrage à froid....28
IsFirstMastCycle
premier cycle MAST....28
IsFirstMastWarmCycle
premier cycle de démarrage à chaud....30

M

M262 PLCSystem
GetExternalEventValue....30
GetImmediateFastInput....25
PhysicalWriteFastOutputs....31
SetRTCDrift....32

MINUTE 33

P

PhysicalWriteFastOutputs
writing output of an embedded expert I/O 31
PLC_GVL 14
PLC_R
variable système 17
PLC_R_APPLICATION_ERROR
type de données 43
PLC_R_BOOT_PROJECT_STATUS
data type 45
PLC_R_IO_STATUS
type de données 45
PLC_R_SDCARD_STATUS
type de données 46
PLC_R_STATUS
type de données 46
PLC_R_STOP_CAUSE
type de données 47
PLC_R_STRUCT 17
PLC_R_TERMINAL_PORT_STATUS
type de données 48
PLC_R_TM3_BUS_STATE
type de données 49
PLC_W
variable système 20
PLC_W_COMMAND
type de données 49
PLC_W_STRUCT 20
PLCSystemBase
DataFileCopy 39
DataFileCopyError 50
DataFileCopyLocation 50
ETH_R 21
ETH_R_FRAME_PROTOCOL 51
ETH_R_IP_MODE 52
ETH_R_IPFORWARDING 52
ETH_R_ITF_STRUCT 52
ETH_R_PORT_DUPLEX_STATUS 54
ETH_R_PORT_IP_STATUS 54
ETH_R_PORT_LINK_STATUS 54
ETH_R_PORT_SPEED 55
ETH_R_RUN_IDLE 55
ETH_W 24
ExecuteScript 41
ExecuteScriptError 51
FB_CheckAllowedControllerMacAddr 37
FB_ControlClone 38
FB_GetFreeDiskSpace 34
FB_GetLabel 35
FB_GetTotalDiskSpace 36
GetRtc 26
IMMEDIATE_ERR_TYPE 56
IsFirstMastColdCycle 28
IsFirstMastCycle 28
IsFirstMastWarmCycle 30
PLC_R 17
PLC_R_APPLICATION_ERROR 43
PLC_R_BOOT_PROJECT_STATUS 45
PLC_R_IO_STATUS 45
PLC_R_SDCARD_STATUS 46
PLC_R_STATUS 46
PLC_R_STOP_CAUSE 47
PLC_R_TERMINAL_PORT_STATUS 48
PLC_R_TM3_BUS_STATE 49
PLC_W 20
PLC_W_COMMAND 49

R

RTC
GetRtc 26
SetRTCDrift 32
RTCSETDRIFT_ERROR
Types de données 56

S

SERIAL_R
variable système 58
SERIAL_R_STRUCT 58
SERIAL_W
variable système 59
SERIAL_W_STRUCT 59
SerialLineSystem
SERIAL_R 58
SERIAL_W 59
SetRTCDrift
accélération ou ralentissement de la fréquence horodateur 32
STRUCT_TMS_BUS_DIAG 71
STRUCT_TMS_MODULE_DIAG 71
Système M262
HasForcedIo 27

T

TM3_BUS_PARAM_ID
type de données 66
TM3_BUS_W
variable système 67
TM3_BUS_W_IOBUSERRMOD
type de données 66
TM3_BUS_W_IOBUSINIT
type de données 67
TM3_BUS_W_STRUCT 67
TM3_ERR_CODE
type de données 67
TM3_GetModuleBusStatus
obtention de l'état du bus d'un module TM3 62
TM3_GetModuleInternalStatus
obtention de l'état interne d'un module TM3 63
TM3_MODULE_R
variable système 61
TM3_MODULE_R_ARRAY_TYPE
type de données 68
TM3_MODULE_R_STRUCT 61
TM3_MODULE_STATE
type de données 68
TM3System
TM3_BUS_PARAM_ID 66
TM3_BUS_W 67
TM3_BUS_W_IOBUSERRMOD 66
TM3_BUS_W_IOBUSINIT 67
TM3_ERR_CODE 67
TM3_GetModuleBusStatus 62–63
TM3_MODULE_R 61
TM3_MODULE_R_ARRAY_TYPE 68
TM3_MODULE_STATE 68
TMS_BUS_DIAG_R
variable système 71
TMS_IP_STATE
type de données 73
TMS_MODULE_DIAG_R
variable système 71
TMS_MODULE_R_ARRAY_TYPE 71
TMS_MODULE_STATE

type de données 73
TMS_PIXCMD_STATE
types de données 74
TMSSystem
TMS_BUS_DIAG_R 71
TMS_IP_STATE 73
TMS_MODULE_DIAG_R 71
TMS_MODULE_STATE 73
TMS_PIXCMD_STATE 74
types de données
DataFileCopyError 50
DataFileCopyLocation 50
ETH_R_FRAME_PROTOCOL 51
ETH_R_IP_MODE 52
ETH_R_IPFORWARDING 52
ETH_R_ITF_STRUCT 52
ETH_R_PORT_DUPLEX_STATUS 54
ETH_R_PORT_IP_STATUS 54
ETH_R_PORT_LINK_STATUS 54
ETH_R_PORT_SPEED 55
ETH_R_RUN_IDLE 55
ExecuteScriptError 51
IMMEDIATE_ERR_TYPE 56
PLC_R_APPLICATION_ERROR 43
PLC_R_BOOT_PROJECT_STATUS 45
PLC_R_IO_STATUS 45
PLC_R_SDCARD_STATUS 46
PLC_R_STATUS 46
PLC_R_STOP_CAUSE 47
PLC_R_TERMINAL_PORT_STATUS 48
PLC_R_TM3_BUS_STATE 49
PLC_W_COMMAND 49
TM3_BUS_PARAM_ID 66
TM3_BUS_W_IOBUSERRMOD 66
TM3_BUS_W_IOBUSINIT 67
TM3_ERR_CODE 67
TM3_MODULE_R_ARRAY_TYPE 68
TM3_MODULE_STATE 68
TMS_IP_STATE 73
TMS_MODULE_STATE 73
TMS_PIXCMD_STATE 74
Types de données
RTCSETDRIFT_ERROR 56

V

variables système
définition 14
ETH_R 21
ETH_W 24
PLC_R 17
PLC_W 20
SERIAL_R 58
SERIAL_W 59
TM3_BUS_W 67
TM3_MODULE_R 61
TMS_BUS_DIAG_R 71
TMS_MODULE_DIAG_R 71
utilisation 15

Schneider Electric
35 rue Joseph Monier
92500 Rueil Malmaison
France

+ 33 (0) 1 41 29 70 00

www.se.com

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

© 2024 – Schneider Electric. Tous droits réservés.

EIO0000003668.07