

Modicon M241 Logic Controller

Fonctions et variables système

Guide de la bibliothèque PLCSystem

Traduction de la notice originale

EIO0000003066.06

12/2024



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
Variables système du contrôleur M241	13
Variables système : définition et utilisation	13
Présentation des variables système.....	13
Utilisation des variables système	14
Structures <i>PLC_R</i> et <i>PLC_W</i>	15
<i>PLC_R</i> : Variables système en lecture seule du contrôleur	16
<i>PLC_W</i> : Variables système en lecture/écriture du contrôleur	19
Structures <i>SERIAL_R</i> et <i>SERIAL_W</i>	19
<i>SERIAL_R</i> [0...1] : Variables système en lecture seule concernant les lignes série	19
<i>SERIAL_W</i> [0...1] : Variables système en lecture/écriture concernant les lignes série.....	20
Structures <i>ETH_R</i> et <i>ETH_W</i>	21
<i>ETH_R</i> : Variables système en lecture seule du port Ethernet	21
<i>ETH_W</i> : Variables système en lecture/écriture des ports Ethernet	24
Structure <i>TM3_MODULE_R</i>	24
<i>TM3_MODULE_R</i> [0...13] : Variables système en lecture seule des modules TM3.....	24
Structure <i>TM3_BUS_W</i>	25
<i>TM3_BUS_W</i> : Variables système de bus TM3	25
Structure <i>PROFIBUS_R</i>	25
<i>PROFIBUS_R</i> : Variables système en lecture seule PROFIBUS	25
Structure <i>CART_R</i>	26
<i>CART_R_STRUCT</i> : Variables système en lecture seule des cartouches.....	26
Fonctions système de M241	27
Fonctions de lecture de M241	27
<i>GetImmediateFastInput</i> : Lire l'entrée d'une E/S experte intégrée.....	27
<i>GetRtc</i> : Obtenir l'horodateur.....	28
<i>HasForcedIo</i> : indique si une entrée ou une sortie est forcée	28
<i>IsFirstMastColdCycle</i> : indique si ce Cycle est le premier cycle de démarrage à froid de MAST	29
<i>IsFirstMastCycle</i> : Indique si Cycle est le premier cycle MAST.....	30
<i>IsFirstMastWarmCycle</i> : Indique si Cycle est le premier cycle de démarrage à chaud MAST	31
Fonctions d'écriture de l'automate M241	32
<i>InhibitBatLed</i> : Activer ou désactiver le voyant de la batterie	32
<i>PhysicalWriteFastOutputs</i> : Ecrire la sortie rapide d'une E/S experte intégrée	33
<i>SetRTCDrift</i> : Définir la valeur de compensation sur l'horodateur	34

Fonctions utilisateur de M241	35
<i>FB_ControlClone</i> : Cloner le contrôleur	35
<i>DataFileCopy</i> : commandes de copie de fichier	36
<i>ExecuteScript</i> : Exécution de commandes de script	38
Fonctions d'espace disque M241	39
<i>FC_GetFreeDiskSpace</i> : Obtient l'espace mémoire disponible	39
<i>FC_GetLabel</i> : Obtient le libellé d'un support mémoire	40
<i>FC_GetTotalDiskSpace</i> : Obtient la taille de la mémoire	41
Fonctions de lecture TM3	42
<i>TM3_GetModuleBusStatus</i> : Obtenir l'état du bus du module TM3	42
<i>TM3_GetModuleFWVersion</i> : Obtenir la version de micrologiciel des modules TM3.....	43
<i>TM3_GetModuleInternalStatus</i> : Obtenir l'état interne du module TM3	44
Types de données de la bibliothèque PLCSystem M241	47
Types de données des variables système <i>PLC_RW</i>	47
<i>PLC_R_APPLICATION_ERROR</i> : Codes d'état d'erreur détecté de l'application	48
<i>PLC_R_BOOT_PROJECT_STATUS</i> : Codes d'état du projet de démarrage	49
<i>PLC_R_IO_STATUS</i> : Codes d'état des E/S	49
<i>PLC_R_SDCARD_STATUS</i> : Codes d'état d'emplacement de carte SD	50
<i>PLC_R_STATUS</i> : Codes d'état du contrôleur	50
<i>PLC_R_STOP_CAUSE</i> : Codes de cause de transition de l'état RUN à un autre état.....	51
<i>PLC_R_TERMINAL_PORT_STATUS</i> : Codes d'état de connexion du port de programmation	52
<i>PLC_R_TM3_BUS_STATE</i> : Codes d'état du bus TM3	52
<i>PLC_W_COMMAND</i> : Codes de commande de contrôle.....	52
Types de données des variables système <i>DataFileCopy</i>	53
<i>DataFileCopyError</i> : Codes d'erreur détectée	53
<i>DataFileCopyLocation</i> : Codes d'emplacement.....	53
Types de données des variables système <i>ExecScript</i>	53
<i>ExecuteScriptError</i> : Codes d'erreur détectée	54
Types de données des variables système <i>ETH_RW</i>	54
<i>ETH_R_FRAME_PROTOCOL</i> : Codes du protocole de transmission de trames.....	54
<i>ETH_R_IP_MODE</i> : Codes de source d'adresse IP	54
<i>ETH_R_PORT_DUPLEX_STATUS</i> : Codes de mode de transmission	55
<i>ETH_R_PORT_IP_STATUS</i> : Codes d'état du port Ethernet TCP/IP	55
<i>ETH_R_PORT_LINK_STATUS</i> : Codes d'état de liaison de communication	55
<i>ETH_R_PORT_SPEED</i> : Codes de vitesse de communication des ports Ethernet.....	56
<i>ETH_R_RUN_IDLE</i> : Codes des états de fonctionnement et d'inactivité Ethernet/IP	56

Types de données des variables système <i>TM3_MODULE_RW</i>	56
<i>TM3_ERR_CODE</i> : Codes d'erreur détectés du module d'extension TM3.....	57
<i>TM3_MODULE_R_ARRAY_TYPE</i> : Type de tableau de lecture du module d'extension TM3	57
<i>TM3_MODULE_STATE</i> : Codes d'état du module d'extension TM3	57
<i>TM3_BUS_W_IOBUSERRMOD</i> : Mode d'erreur du bus TM3	58
Types de données des variables système des cartouches.....	58
<i>CART_R_ARRAY_TYPE</i> : Type tableau de lecture de cartouche	58
<i>CART_R_MODULE_ID</i> : Identificateur de module de lecture de cartouche	58
<i>CART_R_STATE</i> : Etat de lecture de cartouche.....	59
Types de données des fonctions système	59
<i>IMMEDIATE_ERR_TYPE</i> : <i>GetImmediateFastInput</i> - Lire l'entrée des codes d'E/S expertes intégrées	59
<i>RTCSETDRIFT_ERROR</i> : Codes des erreurs détectées par la fonction <i>SetRTCDrift</i>	59
Annexes	61
Représentation des fonctions et blocs fonction	62
Différences entre fonction et bloc fonction	62
Utilisation d'une fonction ou d'un bloc fonction en langage IL	63
Utilisation d'une fonction ou d'un bloc fonction en langage ST	66
Glossaire	69
Index	76

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 disponibles dans le Modicon M241 Logic Controller. La bibliothèque PLCSystem du M241 contient des fonctions et des variables permettant de communiquer avec le système contrôleur (réception d'informations et envoi de commandes).

Ce document décrit les fonctions et variables des types de données de la bibliothèque PLCSystem du contrôleur M241.

Les connaissances préalables suivantes sont requises :

- Connaissances de base sur les fonctionnalités, la structure et la configuration du M241 Logic 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 (EIO0000003065)
- Français (EIO0000003066)
- Allemand (EIO0000003067)
- Espagnol (EIO0000003068)
- Italien (EIO0000003069)
- Chinois (EIO0000003070)

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 M241 Logic Controller – Guide de référence du matériel	EIO0000003083 (ENG)
	EIO0000003084 (FRE)
	EIO0000003085 (GER)
	EIO0000003086 (SPA)
	EIO0000003087 (ITA)
	EIO0000003088 (CHS)
Modicon M241 Logic Controller – Guide de programmation	EIO0000003059 (ENG)
	EIO0000003060 (FRE)
	EIO0000003061 (GER)
	EIO0000003062 (SPA)
	EIO0000003063 (ITA)
	EIO0000003064 (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 IMPREVU DE L'EQUIPEMENT

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

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

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.

Variables système du contrôleur M241

Présentation

Ce chapitre :

- fournit une introduction aux variables système, page 13 ;
- décrit les variables système, page 16 disponibles avec la bibliothèque PLCSystem de M241.

Variables système : définition et utilisation

Présentation

Cette section définit les variables système et explique leur mise en œuvre dans le Modicon M241 Logic 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, il 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*. Le *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

Elles 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 localisées :

- ont un emplacement fixe dans une zone %MW statique : %MW60000 à %MW60199 pour les variables système en lecture seule.
- sont accessibles par l'intermédiaire de requêtes Modbus TCP, Modbus série et EtherNet/IP dans les états RUNNING et STOPPED ;

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.

Utilisation des variables système

Introduction

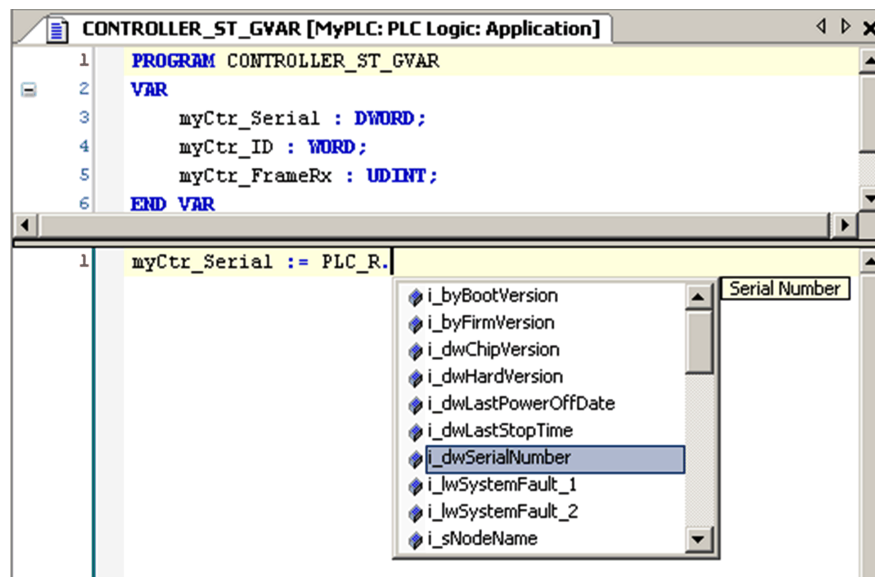
Cette section décrit la procédure de programmation et d'utilisation des variables système dans EcoStruxure Automation Expert - Motion et EcoStruxure Machine Expert.

Les variables système ont un champ d'application global et vous pouvez les utiliser dans tous les POU (unités organisationnelles 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 variables système incluses dans les structures *PLC_R* et *PLC_W*.

PLC_R : Variables système en lecture seule du contrôleur

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>i_wVendorID</i>	WORD	ID du fournisseur du contrôleur. 101A hex = Schneider Electric
60001	<i>i_wProductID</i>	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).
60002	<i>i_dwSerialNumber</i>	DWORD	Numéro de série du contrôleur.
60004	<i>i_byFirmVersion</i>	ARRAY[0..3] OF BYTE	Version du micrologiciel du contrôleur [aa.bb.cc.dd] : • <i>i_byFirmVersion</i>[0] = aa • ... • <i>i_byFirmVersion</i>[3] = dd
60006	<i>i_byBootVersion</i>	ARRAY[0..3] OF BYTE	Version de démarrage du contrôleur [aa.bb.cc.dd] : • <i>i_byBootVersion</i>[0] = aa • ... • <i>i_byBootVersion</i>[3] = dd
60008	<i>i_dwHardVersion</i>	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>i_dwChipVersion</i>	DWORD	Version du coprocesseur du contrôleur.
60012	<i>i_wStatus</i>	<i>PLC_R_STATUS</i> , page 50	Etat du contrôleur.
60013	<i>i_wBootProjectStatus</i>	<i>PLC_R_BOOT_PROJECT_STATUS</i> , page 49	Renvoie des informations sur l'application de démarrage stockée en mémoire non volatile.
60014	<i>i_wLastStopCause</i>	<i>PLC_R_STOP_CAUSE</i> , page 51	Cause du dernier passage du mode <i>RUN</i> à un autre état.
60015	<i>i_wLastApplicationError</i>	<i>PLC_R_APPLICATION_ERROR</i> , page 48	Cause de la dernière exception du contrôleur.
60016	<i>i_lwSystemFault_1</i>	LWORD	Le champ de bits FFFF FFFF FFFF FFFF hex indique qu'aucune erreur n'a été détectée. Un bit de niveau bas signifie qu'une erreur a été détectée : • bit 0 = erreur d'E/S experte détectée • bit 1 = erreur TM3 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 détectée sur ligne série 2 • bit 6 = erreur CAN 1 détectée • bit 7 = erreur de cartouche 1 détectée • bit 8 = erreur de cartouche 2 détectée • bit 9 = erreur TM4 détectée • bit 10 = erreur de carte SD détectée • bit 11 = erreur de pare-feu détectée • Bit 12 = erreur de serveur DHCP détectée • bit 13 = erreur de serveur OPC UA détectée

Adresse Modbus ⁽¹⁾	Nom de la variable	Type	Commentaire
60020	<i>i_lwSystemFault_2</i>	LWORD	Le champ de bits FFFF hex indique qu'aucune erreur n'a été détectée. Si <i>i_wIOStatus1</i> = <i>PLC_R_IO_SHORTCUT_FAULT</i>, le sens de <i>i_lwSystemFault_2</i> 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>i_wIOStatus1</i>	<i>PLC_R_IO_STATUS</i> , page 49	Etat des E/S expertes intégrées.
60025	<i>i_wIOStatus2</i>	<i>PLC_R_IO_STATUS</i> , page 49	Etat d'E/S TM3.
60026	<i>i_wClockBatteryStatus</i>	WORD	État de la batterie de l'horodateur : • 0 = changement de batterie requis • 100 = batterie en pleine charge Les autres valeurs (1 à 99) représentent le pourcentage de charge. Par exemple, si la valeur est 75, la batterie est chargée à 75 %.
60028	<i>i_dwAppliSignature1</i>	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>i_dwAppliSignature2</i>	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>i_dwAppliSignature3</i>	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>i_dwAppliSignature4</i>	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.
n/a	<i>i_sVendorName</i>	STRING(31)	Nom du fournisseur : "Schneider Electric".
n/a	<i>i_sProductRef</i>	STRING(31)	Référence du contrôleur.
n/a	<i>i_sNodeName</i>	STRING(99)	Nom du nœud sur le réseau logiciel
n/a	<i>i_dwLastStopTime</i>	DWORD	Heure du dernier <i>STOP</i> détecté, en secondes depuis le 1er janvier 1970 à 00:00:00 (UTC).
n/a	<i>i_dwLastPowerOffDate</i>	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 <i>SysTimeRtcConvertUtcToDate</i>. Pour plus d'informations sur la conversion de date et d'heure, reportez-vous au Guide de la bibliothèque System (voir Affichage et réglage de l'horodateur – Guide des bibliothèques SysTimeRtc et SysTimeCore).
n/a	<i>i_uiEventsCounter</i>	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 <i>PLC_W.q_wResetCounterEvent</i>.
n/a	<i>i_wTerminalPortStatus</i>	<i>PLC_R_TERMINAL_PORT_STATUS</i> , page 52	Etat du port de programmation USB (USB mini B).

Adresse Modbus ⁽¹⁾	Nom de la variable	Type	Commentaire
n/a	<i>i_wSdCardStatus</i>	<i>PLC_R_SDCARD_STATUS</i> , page 50	Etat de la carte SD.
n/a	<i>i_wUsrFreeFileHdl</i>	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.
n/a	<i>i_udiUsrFsTotalBytes</i>	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 <i>/usr/</i>.
n/a	<i>i_udiUsrFsFreeBytes</i>	UDINT	Taille de la mémoire libre du système de fichiers de l'utilisateur (en octets).
n/a	<i>i_uiTM3BusState</i>	<i>PLC_R_TM3_BUS_STATE</i> , page 52	État du bus TM3. <i>i_uiTM3BusState</i> 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).
n/a	<i>i_ExpertIO_RunStop_Input</i>	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.
n/a	<i>i_x10msClk</i>	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 logique est dans l'état Stop et dans l'état Run.
n/a	<i>i_x100msClk</i>	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 Logic Controller est dans l'état Arrêter et dans l'état Démarrer.
n/a	<i>i_x1sClk</i>	BOOL	Bit de base de temps : 1 s. Cette variable s'active et se désactive par période = 1 ms. La valeur bascule lorsque le Logic Controller est dans l'état Arrêter et dans l'état Démarrer.
(1) signifie que l'adresse Modbus n'est pas accessible via l'application.			
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

Structure de la variable

Le tableau suivant décrit les paramètres de la variable système *PLC_W* (type *PLC_W_STRUCT*) :

%MW	Nom de la variable	Type	Commentaire
s/o	<i>q_wResetCounterEvent</i>	WORD	Le passage de 0 à 1 réinitialise le compteur d'événements (<i>PLC_R.i_uiEventsCounter</i>). 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	<i>q_uiOpenPLCControl</i>	UINT	Lorsque la valeur de la variable passe de 0 à 6699, la commande précédemment écrite dans le paramètre <i>PLC_W.q_wPLCControl</i> suivant est exécutée.
s/o	<i>q_wPLCControl</i>	<i>PLC_W_COMMAND</i> , page 52	Commande RUN/STOP du contrôleur exécutée lorsque la valeur de la variable système <i>PLC_W.q_uiOpenPLCControl</i> passe de 0 à 6699.
s/o signifie qu'aucune affectation %MW n'est prédéfinie pour cette variable système.			

Structures *SERIAL_R* et *SERIAL_W*

Présentation

Cette section répertorie et décrit les variables système des structures *SERIAL_R* et *SERIAL_W*.

SERIAL_R[0...1] : Variables système en lecture seule concernant les lignes série

Introduction

SERIAL_R est un tableau contenant 2 types *SERIAL_R_STRUCT*. Chaque élément du tableau renvoie des variables système de diagnostic pour la ligne série correspondante.

Pour le M241 Logic Controller :

- *Serial_R*[0] désigne la ligne série 1.
- *Serial_R*[1] désigne la ligne série 2.

Structure de la variable

Ce tableau décrit les paramètres des variables système *SERIAL_R[0...1]* :

%MW	Nom de la variable	Type	Commentaire
Ligne série			
s/o	<i>i_udiFramesTransmittedOK</i>	UDINT	Nombre de trames transmises avec succès.
s/o	<i>i_udiFramesReceivedOK</i>	UDINT	Nombre de trames reçues sans aucune erreur détectée.
s/o	<i>i_udiRX_MessagesError</i>	UINT	Nombre de trames reçues avec erreurs détectées (somme de contrôle, parité).
Spécifique Modbus			
s/o	<i>i_uiSlaveExceptionCount</i>	UINT	Nombre de réponses d'exception Modbus renvoyées par le Logic Controller.
s/o	<i>i_udiSlaveMsgCount</i>	UINT	Nombre de messages reçus du maître et adressés au Logic Controller.
s/o	<i>i_uiSlaveNoRespCount</i>	UINT	Nombre de demandes de diffusion Modbus reçues par le Logic Controller.
s/o	<i>i_uiSlaveNakCount</i>	UINT	Inutilisé
s/o	<i>i_uiSlaveBusyCount</i>	UINT	Inutilisé
s/o	<i>i_uiCharOverrunCount</i>	UINT	Nombre de débordements de caractères.
s/o signifie qu'aucun mappage %MW n'est prédéfini pour cette variable système. Inutilisé signifie que la variable n'est pas gérée par le système et que si sa valeur est différente de zéro, elle doit être considérée comme parasite.			

Les compteurs *SERIAL_R* sont réinitialisés dans les cas suivants :

- Téléchargement.
- Réinitialisation du contrôleur.
- Commande *SERIAL_W[x].q_wResetCounter*.
- commande de réinitialisation associée au code fonction n° 8 de la requête Modbus.

SERIAL_W[0...1] : Variables système en lecture/écriture concernant les lignes série

Introduction

SERIAL_W est un tableau de 2 types *SERIAL_W_STRUCT*. Chaque élément du tableau réinitialise les variables système *SERIAL_R* de la ligne série correspondante.

Pour le M241 Logic Controller :

- *Serial_W[0]* désigne la ligne série 1.
- *Serial_W[1]* désigne la ligne série 2.

Structure de la variable

Ce tableau décrit les paramètres des variables système *SERIAL_W[0...1]* :

%MW	Nom de la variable	Type	Commentaire
s/o	<i>q_wResetCounter</i>	WORD	Le passage de 0 à 1 réinitialise tous les compteurs <i>SERIAL_R[0...1]</i> . Pour réinitialiser à nouveau les compteurs, 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.			

Structures *ETH_R* et *ETH_W*

Présentation

Cette section répertorie et décrit les variables système incluses dans les structures *ETH_R* et *ETH_W*.

ETH_R : Variables système en lecture seule du port Ethernet

Structure de la variable

Le tableau suivant décrit les paramètres de la variable système *ETH_R* (type *ETH_R_STRUCT*) :

%MW	Nom de la variable	Type	Commentaire
60050	<i>i_byIPAddress</i>	ARRAY[0..3] OF BYTE	Adresse IP [aaa.bbb.ccc.ddd] : <i>i_byIPAddress</i>[0] = aaa ... <i>i_byIPAddress</i>[3] = ddd
60052	<i>i_bySubNetMask</i>	ARRAY[0..3] OF BYTE	Masque de sous-réseau [aaa.bbb.ccc.ddd] : <i>i_bySub-netMask</i>[0] = aaa ... <i>i_bySub-netMask</i>[3] = ddd
60054	<i>i_byGateway</i>	ARRAY[0..3] OF BYTE	Adresse de passerelle [aaa.bbb.ccc.ddd] : <i>i_byGateway</i>[0] = aaa ... <i>i_byGateway</i>[3] = ddd
60056	<i>i_byMACAddress</i>	ARRAY[0..5] OF BYTE	Adresse MAC [aa.bb.cc.dd.ee.ff] : <i>i_byMACAddress</i>[0] = aa ... <i>i_byMACAddress</i>[5] = ff
60059	<i>i_sDeviceName</i>	STRING(15)	Nom utilisé pour obtenir l'adresse IP auprès du serveur.
s/o	<i>i_wlpMode</i>	<i>ETH_R_IP_MODE</i> , page 54	Méthode utilisée pour obtenir une adresse IP.
s/o	<i>i_byFDRServerIPAddress</i>	ARRAY[0..3] OF BYTE	Adresse IP [aaa.bbb.ccc.ddd] du serveur DHCP ou BootP : <i>i_byFDRServerIPAddress</i>[0] = aaa ... <i>i_byFDRServerIPAddress</i>[3] = ddd Egale à 0.0.0.0 en cas d'utilisation d'une adresse IP enregistrée ou par défaut.
s/o	<i>i_udiOpenTcpConnections</i>	UDINT	Nombre de connexions TCP ouvertes.

%MW	Nom de la variable	Type	Commentaire
s/o	<i>i_udiFramesTransmittedOK</i>	UDINT	Nombre de trames transmises correctement. Réinitialisation lors de la mise sous tension ou avec la commande de réinitialisation <i>ETH_W.q_wResetCounter</i> .
s/o	<i>i_udiFramedReceivedOK</i>	UDINT	Nombre de trames reçues correctement. Réinitialisation lors de la mise sous tension ou avec la commande de réinitialisation <i>ETH_W.q_wResetCounter</i> .
s/o	<i>i_udiTransmitBufferErrors</i>	UDINT	Nombre de trames transmises avec détection d'erreurs. Réinitialisation lors de la mise sous tension ou avec la commande de réinitialisation <i>ETH_W.q_wResetCounter</i> .
s/o	<i>i_udiReceiveBufferErrors</i>	UDINT	Nombre de trames reçues avec détection d'erreurs. Réinitialisation lors de la mise sous tension ou avec la commande de réinitialisation <i>ETH_W.q_wResetCounter</i> .
s/o	<i>i_wFrameSendingProtocol</i>	<i>ETH_R_FRAME_PROTOCOL</i> , page 54	Protocole Ethernet configuré pour l'envoi des trames (IEEE 802.3 ou Ethernet II).
s/o	<i>i_wPortALinkStatus</i>	<i>ETH_R_PORT_LINK_STATUS</i> , page 55	Liaison du port Ethernet (0 = aucune liaison, 1 = liaison connectée à un autre équipement Ethernet).
s/o	<i>i_wPortASpeed</i>	<i>ETH_R_PORT_SPEED</i> , page 56	Vitesse réseau du port Ethernet (10 Mbits/s, 100 Mbits/s).
s/o	<i>i_wPortADuplexStatus</i>	<i>ETH_R_PORT_DUPLEX_STATUS</i> , page 55	Etat duplex du port Ethernet (0 = semi duplex ou 1 = duplex intégral).
s/o	<i>i_udiPortACollisions</i>	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 <i>ETH_W.q_wResetCounter</i> .
s/o	<i>i_byIPAddress_If2</i>	ARRAY[0..3] OF BYTE	Adresse IP du module d'extension TM4.
s/o	<i>i_bySubNetMask_If2</i>	ARRAY[0..3] OF BYTE	Masque de sous-réseau du module d'extension TM4.
s/o	<i>i_byGateway_If2</i>	ARRAY[0..3] OF BYTE	Adresse de passerelle du module d'extension TM4.
s/o	<i>i_byMACAddress_If2</i>	ARRAY[0..3] OF BYTE	Adresse MAC du module d'extension TM4.
s/o	<i>i_sDeviceName_If2</i>	STRING(15)	Nom permettant d'obtenir l'adresse IP du module d'extension TM4.
s/o	<i>i_wIpMode_If2</i>	<i>ETH_R_IP_MODE</i> , page 54	Méthode permettant d'obtenir l'adresse IP du module d'extension TM4.
s/o	<i>i_wPortALinkStatus_If2</i>	<i>ETH_R_PORT_LINK_STATUS</i> , page 55	Lien du port Ethernet du module d'extension TM4 : 0 : Aucune liaison 1 : Liaison connectée à un autre équipement Ethernet
s/o	<i>i_wPortASpeed_If2</i>	<i>ETH_R_PORT_SPEED</i> , page 56	Débit réseau du port Ethernet du module d'extension TM4 (10 ou 100 Mbits/s).
s/o	<i>i_wPortADuplexStatus_If2</i>	<i>ETH_R_PORT_DUPLEX_STATUS</i> , page 55	Etat duplex du port Ethernet du module d'extension TM4 : 0 : Semi 1 : Duplex intégral
s/o	<i>i_wPortAlpStatus_If2</i>	<i>ETH_R_PORT_IP_STATUS</i> , page 55	Etat de la pile du port Ethernet TCP/IP du module d'extension TM4.
Spécifique à Modbus TCP/IP			
s/o	<i>i_udiModbusMessageTransmitted</i>	UDINT	Nombre de messages Modbus transmis. Réinitialisation lors de la mise sous tension ou avec la commande de réinitialisation <i>ETH_W.q_wResetCounter</i> .
s/o	<i>i_udiModbusMessageReceived</i>	UDINT	Nombre de messages Modbus reçus. Réinitialisation lors de la mise sous tension ou avec la commande de réinitialisation <i>ETH_W.q_wResetCounter</i> .
s/o	<i>i_udiModbusErrorMessage</i>	UDINT	Messages de détection d'erreurs Modbus transmis et reçus. Réinitialisation lors de la mise sous tension ou avec la commande de réinitialisation <i>ETH_W.q_wResetCounter</i> .

%MW	Nom de la variable	Type	Commentaire
Spécifique à EtherNet/IP			
s/o	<i>i_udiETHIP_IOMessagingTransmitted</i>	UDINT	Trames EtherNet/IP de classe 1 transmises. Réinitialisation lors de la mise sous tension ou avec la commande de réinitialisation <code>ETH_W.q_wResetCounter</code> .
s/o	<i>i_udiETHIP_IOMessagingReceived</i>	UDINT	Trames EtherNet/IP de classe 1 reçues. Réinitialisation lors de la mise sous tension ou avec la commande de réinitialisation <code>ETH_W.q_wResetCounter</code> .
s/o	<i>i_udiUCMM_Request</i>	UDINT	Messages EtherNet/IP non connectés reçus. Réinitialisation lors de la mise sous tension ou avec la commande de réinitialisation <code>ETH_W.q_wResetCounter</code> .
s/o	<i>i_udiUCMM_Error</i>	UDINT	Messages EtherNet/IP non connectés non valides reçus. Réinitialisation lors de la mise sous tension ou avec la commande de réinitialisation <code>ETH_W.q_wResetCounter</code> .
s/o	<i>i_udiClass3_Request</i>	UDINT	Requêtes EtherNet/IP de classe 3 reçues. Réinitialisation lors de la mise sous tension ou avec la commande de réinitialisation <code>ETH_W.q_wResetCounter</code> .
s/o	<i>i_udiClass3_Error</i>	UDINT	Requêtes EtherNet/IP de classe 3 non valides reçues. Réinitialisation lors de la mise sous tension ou avec la commande de réinitialisation <code>ETH_W.q_wResetCounter</code> .
s/o	<i>i_uiAssemblyInstanceInput</i>	UINT	Numéro de l'instance d'assemblage d'entrée. Pour plus d'informations, reportez-vous au guide de programmation du contrôleur.
s/o	<i>i_uiAssemblyInstanceInputSize</i>	UINT	Taille de l'instance d'assemblage d'entrée. Pour plus d'informations, consultez le guide de programmation de votre contrôleur.
s/o	<i>i_uiAssemblyInstanceOutput</i>	UINT	Numéro de l'instance d'assemblage de sortie. Pour plus d'informations, consultez le guide de programmation de votre contrôleur.
s/o	<i>i_uiAssemblyInstanceOutputSize</i>	UINT	Taille de l'instance d'assemblage de sortie. Pour plus d'informations, consultez le guide de programmation de votre contrôleur.
s/o	<i>i_uiETHIP_ConnectionTimeouts</i>	UINT	Nombre d'expirations de connexion. Réinitialisation lors de la mise sous tension ou avec la commande de réarmement <code>ETH_W.q_wResetCounter</code> .
s/o	<i>i_ucEipRunIdle</i>	<i>ETH_R_RUN_IDLE</i> , page 56	Drapeau fonctionnement (valeur = 1) / attente (valeur = 0) pour la connexion EtherNet/IP classe 1.
s/o	<i>i_byMasterIpTimeouts</i>	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 <code>ETH_W.q_wResetCounter</code> .
s/o	<i>i_byMasterIpLost</i>	BYTE	État de la liaison maître Ethernet Modbus TCP : 0 = liaison OK, 1 = liaison perdue.
s/o	<i>i_wPortIpStatus</i>	<i>ETH_R_PORT_IP_STATUS</i> , page 55	Etat de la pile du port TCP/IP Ethernet
s/o signifie qu'aucun mappage %MW n'est prédéfini pour cette variable système.			

ETH_W : Variables système en lecture/écriture des ports Ethernet

Structure de la variable

Le tableau suivant décrit les paramètres de la variable système *ETH_W* (type *ETH_W_STRUCT*) :

%MW	Nom de la variable	Type	Commentaire
s/o	<i>q_wResetCounter</i>	WORD	Le passage de 0 à 1 réinitialise tous les compteurs <i>ETH_R</i> . 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.			

Structure TM3_MODULE_R

Présentation

Cette section répertorie et décrit les variables système incluses dans la structure *TM3_MODULE_R*.

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

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 M241 Logic 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* [0...13] :

%MW	Nom de la variable	Type	Commentaire
n/a	<i>i_wProductID</i>	WORD	ID du module d'extension TM3.
n/a	<i>i_wModuleState</i>	<i>TM3_MODULE_STATE</i> , page 57	Décrit l'état du module TM3.
s/o signifie qu'aucun mappage %MW n'est prédéfini pour cette variable système.			

Structure *TM3_BUS_W*

Présentation

Cette section répertorie et décrit les variables système incluses dans la structure *TM3_BUS_W*.

TM3_BUS_W : Variables système de bus TM3

Structure de la variable

Le tableau suivant décrit les paramètres de la variable système *TM3_BUS_W* (type *TM3_BUS_W_STRUCT*) :

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

Pour plus d'informations, reportez-vous à la Description générale de la configuration des E/S (voir Modicon M241 Logic Controller, Guide de programmation).

Structure *PROFIBUS_R*

PROFIBUS_R : Variables système en lecture seule PROFIBUS

Structure de la variable

Ce tableau décrit les paramètres des variables système *PROFIBUS_R* (type *PROFIBUS_R_STRUCT*) :

%MW	Nom de la variable	Type	Commentaire
n/a	<i>i_wPNOIdentifier</i>	WORD	Code d'identification d'esclave (1 à 126).
n/a	<i>i_wBusAdr</i>	UINT	Adresse d'esclave PROFIBUS
n/a	<i>i_CommState</i>	UDINT	Valeur représentant l'état du module PROFIBUS : 0x00 : Indéterminé 0x01 : Non configuré 0x02 : arrêt 0x03 : Inactif 0x04 : Marche
n/a	<i>i_CommError</i>	UDINT	Si la valeur est différente de zéro, une erreur de communication a été détectée par le module Profibus, indiquée par un code d'erreur (voir le guide de programmation des modules d'extension TM4).
n/a	<i>i_ErrorCount</i>	UDINT	Compteur d'erreurs de communication.
s/o signifie qu'aucun mappage %MW n'est prédéfini pour cette variable système.			

Structure *CART_R*

CART_R_STRUCT : Variables système en lecture seule des cartouches

Structure de la variable

Le tableau suivant décrit les paramètres de la variable système *CART_R_STRUCT* :

%MW	Nom de la variable	Type	Commentaire
s/o	<i>i_uiModuleId</i>	CART_R_MODULE_ID, page 58	ID module
s/o	<i>i_uifirmwareVersion</i>	UINT	Version du micrologiciel
s/o	<i>i_udiCartState</i>	CART_R_STATE, page 59	Etat de la cartouche
s/o signifie qu'aucun mappage %MW n'est prédéfini pour cette variable système.			

Fonctions système de M241

Présentation

Ce chapitre décrit les fonctions système disponibles dans la bibliothèque PLCSystem de M241.

Fonctions de lecture de M241

Présentation

Cette section décrit les fonctions de lecture de la bibliothèque PLCSystem de M241.

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 les entrées I0 à I7 sont accessibles via cette fonction.

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 62.

Description des variables d'E/S

Le tableau suivant décrit les variables d'entrée :

Entrée	Type	Commentaire
<i>Block</i>	INT	Non utilisé.
<i>Input</i>	INT	Index d'entrée à lire de 0 à 7.

Le tableau suivant décrit la variable de sortie :

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

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

Entrée/sortie	Type	Commentaire
<i>Error</i>	BOOL	FALSE = opération correctement exécutée. TRUE = opération en erreur, la fonction renvoie une valeur qui n'est pas valide.
<i>ErrID</i>	<i>IMMEDIATE_ERR_TYPE</i> , page 59	Code de l'erreur d'opération détectée lorsque <i>Error</i> a la valeur 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)).

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 62.

Description des variables d'E/S

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

Sortie	Type	Commentaire
<i>GetRtc</i>	DINT	Horodatage actuel en secondes au format UNIX.

Exemple

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

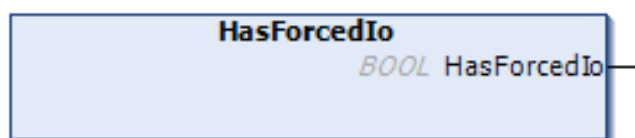
```
VAR
MyRTC : DINT := 0;
END_VAR
MyRTC := GetRtc();
```

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*, page 62.

Description des variables d'E/S

Ce tableau décrit la variable de sortie :

Sortie	Type	Commentaire
<i>HasForcedIo</i>	BOOL	TRUE si une entrée ou une sortie est forcée.

Exemple

L'exemple suivant décrit comment utiliser cette fonction :

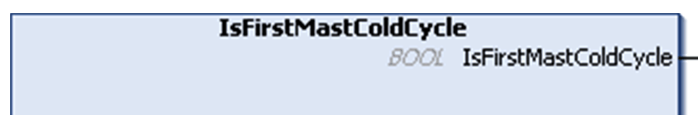
```
VAR
hasIo: BOOL;
END_VAR
hasIo := SEC.HasForcedIo();
```

***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).

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 62.

Description des variables d'E/S

Ce tableau décrit la variable de sortie :

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

Exemple

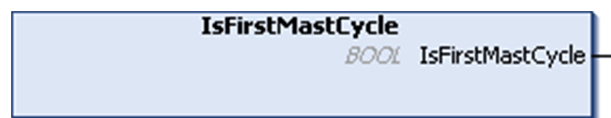
Reportez-vous à la fonction *IsFirstMastCycle*, page 30.

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.

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 62.

Description des variables d'E/S

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

Exemple

Cet exemple décrit les trois fonctions *IsFirstMastCycle*, *IsFirstMastColdCycle* et *IsFirstMastWarmCycle* utilisées 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) :

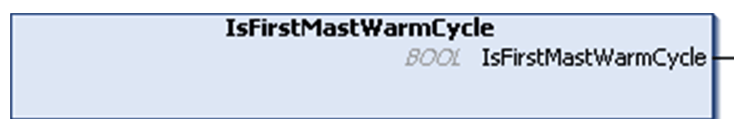
```
VAR
MyIsFirstMastCycle : BOOL;
MyIsFirstMastWarmCycle : BOOL;
MyIsFirstMastColdCycle : BOOL;
END_VAR
MyIsFirstMastWarmCycle := IsFirstMastWarmCycle();
MyIsFirstMastColdCycle := IsFirstMastColdCycle();
MyIsFirstMastCycle := IsFirstMastCycle();
IF (MyIsFirstMastWarmCycle) THEN
(*This is the first Mast 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.

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 62.

Description des variables d'E/S

Ce tableau décrit la variable de sortie :

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

Exemple

Consultez la description de la fonction *IsFirstMastCycle*, page 30.

Fonctions d'écriture de l'automate M241

Vue d'ensemble

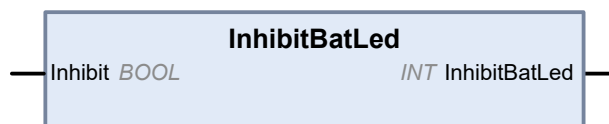
Cette section décrit les fonctions d'écriture de la bibliothèque PLCSystem de l'automate M241.

InhibitBatLed : Activer ou désactiver le voyant de la batterie

Description de la fonction

Cette fonction active ou désactive l'affichage du voyant de la batterie, quel que soit son niveau de charge.

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 62.

Description des variables d'E/S

Le tableau suivant décrit la variable d'entrée :

Entrée	Type	Commentaire
<i>Inhibit</i>	BOOL	Si <i>TRUE</i> , désactive l'affichage du voyant de la batterie. Si <i>FALSE</i> , active l'affichage du voyant de la batterie.

Le tableau suivant décrit la variable de sortie :

Sortie	Type	Commentaire
<i>InhibitBatLed</i>	INT	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.

Exemple

Cet exemple montre comment désactiver l'affichage du voyant de la batterie :
 (* Disable Battery LED Information *)
 SEC.InhibitBatLed(TRUE) ;

PhysicalWriteFastOutputs : Ecrire 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.

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 62.

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, Q0 n'est pas écrite car elle est utilisée par un générateur de fréquence.

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 à plus d'une des sorties intégrées, la dernière d'entre elles (ordre de 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 62.

Description des variables d'E/S

Le tableau suivant décrit les paramètres d'entrée :

Entrées	Type	Commentaire
<i>RtcDrift</i>	<i>SINT (-36..73)</i>	Correction en secondes par semaine (-36 à +73).

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.

Le tableau suivant décrit la variable de sortie :

Sortie	Type	Commentaire
<i>SetRTCDrift</i>	<i>RTCSETDRIFT_ERROR</i> , page 59	Renvoie <i>RTC_OK</i> (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 (-36..+73) := 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 M241

Présentation

Cette section décrit les fonctions *FB_ControlClone*, *DataFileCopy* et *ExecuteScript* disponibles dans la bibliothèque PLCSystem de M241.

FB_ControlClone : Cloner le contrôleur

Description du bloc fonction

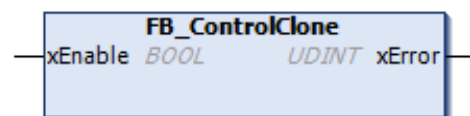
Le clonage est possible par défaut par carte SD ou via l'**Assistant de contrôleur**. Lorsque les droits d'utilisateur sont activés et que le droit d'affichage **ExternalCmd** est refusé au groupe **ExternalMedia**, la fonction de clonage n'est pas autorisée. Dans ce cas, le bloc fonction active la fonctionnalité 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 M241 Logic 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
<i>xEnable</i> = 1	Le clonage est autorisé	Le clonage est autorisé
<i>xEnable</i> = 0	Le clonage n'est pas autorisé	Le clonage n'est pas autorisé

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 62.

Description des variables d'E/S

Le tableau suivant décrit les variables d'entrée :

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

Le tableau suivant décrit les variables de sortie :

Sortie	Type	Commentaire
<i>xError</i>	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
- copier des données en mémoire vers un fichier formaté. Pour plus d'informations, consultez Organisation de la mémoire non volatile (voir Modicon M241 Logic Controller - Guide de programmation).

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 62.

Description des variables d'E/S

Le tableau suivant décrit les variables d'entrée :

Entrées	Type	Commentaire
<i>xExecute</i>	BOOL	Lors d'un 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.
<i>sFileName</i>	STRING	Nom de fichier sans extension (l'extension <i>.DTA</i> est automatiquement ajoutée). N'utilisez que les caractères alphanumériques (a à z, A à Z et 0 à 9).
<i>xRead</i>	BOOL	TRUE : copier les données du fichier identifié par <i>sFileName</i> vers la mémoire interne du contrôleur. FALSE : copier les données de la mémoire interne du contrôleur vers le fichier identifié par <i>sFileName</i> .
<i>xSecure</i>	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.
<i>iLocation</i>	INT	0 : le fichier réside dans le répertoire <i>/usr/DTA</i> du système de fichiers interne. 1 : le fichier réside dans le répertoire <i>/usr/DTA</i> d'un système de fichiers externe (carte SD). NOTE: Si le fichier n'existe pas déjà dans le répertoire, il est créé.
<i>uiSize</i>	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 : <code>Variable : int;</code> <code>uiSize := SIZEOF (Variable);</code>
<i>dwAdd</i>	DWORD	Indique l'adresse en mémoire que la fonction va lire ou écrire. Seules les adresses de variables conformes à la norme CEI 61131-3 (variables, tableaux, structures) sont autorisées. Par exemple : <code>Variable : int;</code> <code>dwAdd := ADR (Variable);</code>

⚠ 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
<i>xDone</i>	BOOL	TRUE = indique que l'action a abouti.
<i>xBusy</i>	BOOL	TRUE = indique que le bloc fonction est en cours d'exécution.
<i>xError</i>	BOOL	TRUE = indique qu'une erreur est détectée et que le bloc fonction a abandonné l'action.
<i>eError</i>	<i>DataFileCopyError</i> , page 53	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*
- *ChangeModbusPort*

Pour plus d'informations sur le format de fichier de script requis, consultez Fichiers de script pour cartes SD (voir Modicon M241 Logic Controller, Guide de programmation).

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 62.

Description des variables d'E/S

Le tableau suivant décrit les variables d'entrée :

Entrée	Type	Commentaire
<i>xExecute</i>	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.
<i>sCmd</i>	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
<i>xDone</i>	BOOL	TRUE indique que l'action a réussi.
<i>xBusy</i>	BOOL	TRUE indique que le bloc fonction est en cours d'exécution.
<i>xError</i>	BOOL	TRUE indique une détection d'erreur ; le bloc fonction annule l'action.
<i>eError</i>	ExecuteScriptError, page 54	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=> );
```

Fonctions d'espace disque M241

Présentation

Cette section décrit les fonctions d'espace disque incluses dans cette bibliothèque.

FC_GetFreeDiskSpace : Obtient l'espace mémoire disponible

Description de la fonction

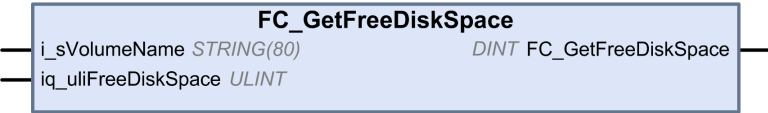
Cette fonction obtient la quantité d'espace libre d'un support mémoire (disque utilisateur, disque système, carte SD), en octets.

Le nom du support mémoire est transféré :

- Disque utilisateur = "/usr"
- Disque système = "/sys"
- Carte SD = "/sd0"

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".

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 62.

Description des variables d'E/S

Le tableau suivant décrit les variables d'entrée :

Entrée	Type de données	Description
i_sVolumeName	STRING[80]	Nom de l'équipement dont l'espace mémoire disponible doit être consulté
iq_uliFreeDiskSpace	ULINT	Espace mémoire libre en octets

Le tableau suivant décrit les variables de sortie :

Sortie	Type de données	Description
FC_GetFreeDiskSpace	DINT	0 : La quantité d'espace mémoire libre a été obtenue. -1 : Erreur lors de la tentative d'accès à la quantité de mémoire libre. Par exemple, un équipement non valide ou un équipement distant a été sélectionné -318 : Paramètre non valide (i_sVolumeName)

FC_GetLabel : Obtient le libellé d'un support mémoire

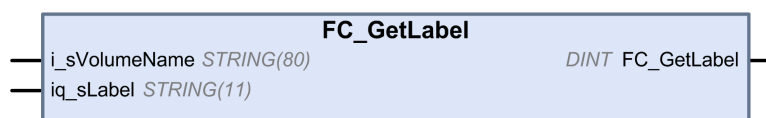
Description de la fonction

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

Le nom du support mémoire (disque utilisateur, disque système, carte SD) est transféré :

- Disque utilisateur = "/usr"
- Disque système = "/sys"
- Carte SD = "/sd0"

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 62.

Description des variables d'E/S

Le tableau suivant décrit les variables d'entrée :

Entrée	Type de données	Description
<code>i_sVolumeName</code>	STRING[80]	Nom de l'équipement dont le libellé doit être consulté
<code>iq_sLabel</code>	STRING[11]	Libellé de l'équipement

Le tableau suivant décrit les variables de sortie :

Sortie	Type de données	Description
<code>FC_GetLabel</code>	DINT	0 : Le libellé a bien été récupéré -1 : Erreur lors de l'accès au libellé -318 : Paramètre incorrect

FC_GetTotalDiskSpace : Obtient la taille de la mémoire

Description de la fonction

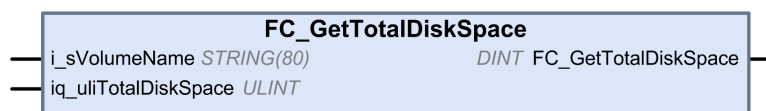
Cette fonction obtient la taille d'un support mémoire (disque utilisateur, disque système, carte SD), en octets.

Le nom du support mémoire est transféré :

- Disque utilisateur = "/usr"
- Disque système = "/sys"
- Carte SD = "/sd0"

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".

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 62.

Description des variables d'E/S

Le tableau suivant décrit les variables d'entrée :

Entrée	Type de données	Description
<i>i_sVolumeName</i>	STRING[80]	Nom de l'équipement dont la taille de la mémoire doit être consultée
<i>iq_uliTotalDiskSpace</i>	ULINT	Taille du support mémoire en octets

Le tableau suivant décrit les variables de sortie :

Sortie	Type de données	Description
<i>FC_GetTotalDiskSpace</i>	DINT	0 : La taille a bien été récupérée -1 : Erreur lors de la lecture de la taille -318 : Au moins un des paramètres n'est pas valide

Fonctions de lecture TM3

Présentation

Cette section décrit les fonctions de lecture TM3 incluses dans la bibliothèque PLCSystem du M241.

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.

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 62.

Description des variables d'E/S

Le tableau suivant décrit la variable d'entrée :

Entrée	Type	Commentaire
<i>ModuleIndex</i>	BYTE	Index du module (0 pour la première extension, 1 pour la deuxième, etc.).

Le tableau suivant décrit la variable de sortie :

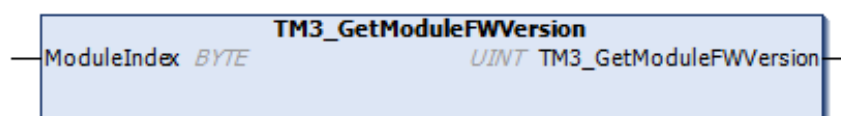
Sortie	Type	Commentaire
<i>TM3_GetModuleBusStatus</i>	<i>TM3_ERR_CODE</i> , page 57	Renvoie <code>TM3_OK</code> (00 hex) si la commande est correcte ou renvoie le code d'identification de l'erreur détectée.

TM3_GetModuleFWVersion : Obtenir la version de micrologiciel des modules TM3

Description de la fonction

Cette fonction renvoie la version du micrologiciel du module TM3 spécifiée.

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 62.

Description des variables d'E/S

Le tableau suivant décrit les variables d'entrée :

Entrée	Type	Commentaire
<i>ModuleIndex</i>	BYTE	Index du module (0 pour la première extension, 1 pour la deuxième, etc.).

Le tableau suivant décrit la variable de sortie :

Sortie	Type	Commentaire
<i>TM3_GetModuleFWVersion</i>	UINT	Renvoie la version du micrologiciel du module ou <code>FFFF</code> hex si l'information n'est pas lisible. Par exemple, <code>001A</code> hex indique la version 26 du micrologiciel.

TM3_GetModuleInternalStatus : Obtenir l'état interne du module TM3

Description de la fonction

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 42.

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 **Etat activé** de l'onglet **Configuration d'E/S** soit désactivé.

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 62.

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
<i>ModuleIndex</i>	BYTE	Index du module d'extension (0 pour le plus proche du contrôleur, 1 pour le suivant, et ainsi de suite)
<i>StatusOffset</i>	BYTE	Décalage du premier état à lire dans la table d'états.
<i>StatusSize</i>	BYTE	Nombre d'octets à lire dans la table d'états.
<i>pStatusBuffer</i>	POINTER TO BYTE	Mémoire tampon contenant la table d'état de lecture (IBStatusIWx / IBStatusQWx).

Le tableau suivant décrit la variable de sortie :

Sortie	Type	Commentaire
<i>TM3_GetModuleInternalStatus</i>	TM3_ERR_CODE, page 57	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 occulter l'état IW0)
- *StatusSize* = 2 (2 états à lire)
- *pStatusBuffer* doit être d'au moins 2 octets

Types de données de la bibliothèque PLCSystem M241

Présentation

Ce chapitre décrit les types de données de la bibliothèque PLCSystem de M241.

Deux types de données sont disponibles :

- Les types de données de variable système sont utilisés par les *variables système*, page 13 de la bibliothèque PLCSystem du M241 (*PLC_R*, *PLC_W*, etc.).
- Les types de données de fonction système sont utilisés par les *fonctions système*, page 27 de lecture/écriture de la bibliothèque PLCSystem du M241.

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

Description du type énumération

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

Énumérateur	Valeur	Commentaire	Que faire
<i>PLC_R_APP_ERR_UNKNOWN</i>	FFFF hex	Erreur indéfinie détectée.	Contactez le service de maintenance de Schneider Electric.
<i>PLC_R_APP_ERR_NOEXCEPTION</i>	0000 hex	Aucune erreur détectée.	–
<i>PLC_R_APP_ERR_WATCHDOG</i>	0010 hex	Tâche chien de garde expirée.	Vérifiez votre application Une réinitialisation est nécessaire pour entrer en mode Démarrer.
<i>PLC_R_APP_ERR_HARDWAREWATCHDOG</i>	0011 hex	Chien de garde du système 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 encore, contactez votre service d'assistance Schneider Electric.
<i>PLC_R_APP_ERR_IO_CONFIG_ERROR</i>	0012 hex	Paramètres de configuration d'E/S incorrects détectés.	Il est possible que votre application soit endommagée. Pour résoudre ce problème, utilisez l'une de ces méthodes : 1. Build > Clean All 2. Exportez/Importez votre application. 3. Mettez à niveau le logiciel avec la dernière version.
<i>PLC_R_APP_ERR_UNRESOLVED_EXTREFS</i>	0018 hex	Fonctions indéfinies détectées.	Supprimez les fonctions non résolues de l'application.
<i>PLC_R_APP_ERR_IEC_TASK_CONFIG_ERROR</i>	0025 hex	Paramètres de configuration de tâche incorrects détectés.	Il est possible que votre application soit endommagée. Pour résoudre ce problème, utilisez l'une de ces méthodes : 1. Build > Clean All 2. Exportez/Importez votre application. 3. Mettez à niveau le logiciel avec la dernière version.
<i>PLC_R_APP_ERR_ILLEGAL_INSTRUCTION</i>	0050 hex	Instruction indéfinie détectée.	Procédez au débogage de votre application pour résoudre le problème.
<i>PLC_R_APP_ERR_ACCESS_VIOLATION</i>	0051 hex	Tentative d'accès à la zone mémoire réservée.	Procédez au débogage de votre application pour résoudre le problème.
<i>PLC_R_APP_ERR_DIVIDE_BY_ZERO</i>	0102 hex	Division d'un entier par 0 détectée.	Procédez au débogage de votre application pour résoudre le problème.
<i>PLC_R_APP_ERR_PROCESSORLOAD_WATCHDOG</i>	0105 hex	Processeur surchargé par les tâches de l'application.	Réduisez la charge de travail de l'application en améliorant son architecture. Augmentez la durée du cycle de tâche. Réduisez la fréquence des événements.
<i>PLC_R_APP_ERR_DIVIDE_REAL_BY_ZERO</i>	0152 hex	Division d'un réel par 0 détectée.	Procédez au débogage de votre application pour résoudre le problème.

Énumérateur	Valeur	Commentaire	Que faire
<i>PLC_R_APP_ERR_EXPIO_EVENTS_COUNT_EXCEEDED</i>	4E20 hex	Trop d'événements sur les E/S expertes sont détectés.	Réduisez le nombre de tâches d'événement.
<i>PLC_R_APP_ERR_APPLICATION_VERSION_MISMATCH</i>	4E21 en hexadécimal	Discordance détectée dans la version de l'application.	La version de l'application dans le Logic Controller 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**

Description du type énumération

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

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

Description du type énumération

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

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

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

Description du type énumération

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

Enumérateur	Valeur	Commentaire
<i>NO_SDCARD</i>	0000 hex	Aucune carte SD n'est détectée dans l'emplacement ou l'emplacement n'est pas connecté.
<i>SDCARD_READONLY</i>	0001 hex	La carte SD est en mode de lecture seule.
<i>SDCARD_READWRITE</i>	0002 hex	La carte SD est en mode de lecture/écriture.
<i>SDCARD_ERROR</i>	0003 hex	Erreur détectée sur la carte SD. Pour plus d'informations sur l'erreur, consultez le fichier FwLog.txt.

PLC_R_STATUS : Codes d'état du contrôleur

Description du type énumération

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

Enumérateur	Valeur	Commentaire
<i>PLC_R_EMPTY</i>	0000 hex	Le contrôleur ne contient aucune application.
<i>PLC_R_STOPPED</i>	0001 hex	Le contrôleur est arrêté.
<i>PLC_R_RUNNING</i>	0002 hex	Le contrôleur fonctionne.
<i>PLC_R_HALT</i>	0004 hex	Le contrôleur est dans un état HALT (voir le schéma des états de contrôleur dans le guide de programmation (voir Modicon M241 Logic Controller, Guide de programmation)).
<i>PLC_R_BREAKPOINT</i>	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

Description du type Enumération

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

Enumérateur	Valeur	Commentaire	Que faire
<i>PLC_R_STOP_REASON_UNKNOWN</i>	00 hex	La valeur initiale ou la cause de l'arrêt n'est pas déterminable.	Contactez le représentant local Schneider Electric.
<i>PLC_R_STOP_REASON_HW_WATCHDOG</i>	01 hex	Arrêté suite au timeout du chien de garde matériel	Contactez le représentant local Schneider Electric.
<i>PLC_R_STOP_REASON_RESET</i>	02 hex	Arrêté suite à une réinitialisation.	Voir les possibilités de réinitialisation dans le schéma des états de contrôleur.
<i>PLC_R_STOP_REASON_EXCEPTION</i>	03 hex	Arrêté suite à une exception.	Vérifiez votre application et effectuez les corrections si nécessaire. Reportez-vous à Horloges de surveillance du système et des tâches (voir Modicon M241 Logic Controller, Guide de programmation). Une réinitialisation est nécessaire pour entrer en mode Run.
<i>PLC_R_STOP_REASON_USER</i>	04 hex	Arrêté suite à une requête de l'utilisateur.	Consultez la section Commande d'arrêt (Stop) dans Commandes de transition d'état (voir Modicon M241 Logic Controller, Guide de programmation).
<i>PLC_R_STOP_REASON_IECPROGRAM</i>	05 hex	Arrêté suite à une requête de commande de programme (par exemple, commande de contrôle avec le paramètre <i>PLC_W.q_wPLCControl:=PLC_W_COMMAND.PLC_W_STOP;</i>).	–
<i>PLC_R_STOP_REASON_DELETE</i>	06 hex	Arrêté suite à une commande de suppression d'application.	Consultez l'onglet Applications de l'éditeur d'équipement de contrôleur (voir Modicon M241 Logic Controller, Guide de programmation).
<i>PLC_R_STOP_REASON_DEBUGGING</i>	07 hex	Arrêté suite au passage en mode de débogage.	–
<i>PLC_R_STOP_FROM_NETWORK_REQUEST</i>	0A hex	Arrêté après une demande du réseau, du serveur Web du contrôleur ou de la commande <i>PLC_W</i> .	–
<i>PLC_R_STOP_FROM_INPUT</i>	0B hex	Arrêt requis par une entrée du contrôleur.	–
<i>PLC_R_STOP_FROM_RUN_STOP_SWITCH</i>	0C hex	Arrêt demandé par le commutateur du contrôleur.	–
<i>PLC_R_STOP_REASON_RETAIN_MISMATCH</i>	0D hex	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.
<i>PLC_R_STOP_REASON_BOOT_APPLI_MISMATCH</i>	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.
<i>PLC_R_STOP_REASON_POWERFAIL</i>	0F hex	Arrêté suite à une coupure de courant.	–

Pour plus d'informations sur les raisons de l'arrêt du contrôleur, reportez-vous à la section Description des états du contrôleur.

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

Description du type énumération

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

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

PLC_R_TM3_BUS_STATE : Codes d'état du bus TM3

Description du type énumération

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

Énumérateur	Valeur	Commentaire
<i>TM3_CONF_ERROR</i>	01 hex	Erreur détectée en raison d'une incohérence entre la configuration physique et la configuration dans le logiciel.
<i>TM3_OK</i>	03 hex	La configuration physique correspond à la configuration dans le logiciel.
<i>TM3_POWER_SUPPLY_ERROR</i>	04 hex	Erreur détectée dans l'alimentation.

PLC_W_COMMAND : Codes de commande de contrôle

Description du type énumération

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

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

Description du type énumération

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

Enumérateur	Valeur	Description
<i>ERR_NO_ERR</i>	00 hex	Aucune erreur détectée.
<i>ERR_FILE_NOT_FOUND</i>	01 hex	Fichier inexistant.
<i>ERR_FILE_ACCESS_REFUSED</i>	02 hex	Ouverture du fichier impossible.
<i>ERR_INCORRECT_SIZE</i>	03 hex	Requête d'une taille différente de celle indiquée dans le fichier.
<i>ERR_CRC_ERR</i>	04 hex	CRC incorrect. Le fichier est considéré comme endommagé.
<i>ERR_INCORRECT_MAC</i>	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

Description du type énumération

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

Enumérateur	Valeur	Description
<i>DFCL_INTERNAL</i>	00 hex	Le fichier de donnée d'extension DTA se trouve dans le répertoire <i>/usr/Dta</i> .
<i>DFCL_EXTERNAL</i>	01 hex	Le fichier de donnée d'extension DTA se trouve dans le répertoire <i>/sd0/usr/Dta</i> .
<i>DFCL_TBD</i>	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

Description du type énumération

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

Enumérateur	Valeur	Description
<i>CMD_OK</i>	00 hex	Aucune erreur détectée.
<i>ERR_CMD_UNKNOWN</i>	01 hex	La commande n'est pas valide.
<i>ERR_SD_CARD_MISSING</i>	02 hex	Carte SD absente.
<i>ERR_SEE_FWLOG</i>	03 hex	Erreur détectée lors de l'exécution de la commande, voir <i>FwLog.txt</i> . Pour plus d'informations, consultez Type de fichier (voir Modicon M241 Logic Controller, Guide de programmation).
<i>ERR_ONLY_ONE_COMMAND_ALLOWED</i>	04 hex	Tentative d'exécution de plusieurs scripts simultanément.
<i>CMD_BEING_EXECUTED</i>	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

Description du type énumération

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

Enumérateur	Valeur	Commentaire
<i>ETH_R_802_3</i>	00 hex	Le protocole utilisé pour la transmission de trames est IEEE 802.3.
<i>ETH_R_ETHERNET_II</i>	01 hex	Le protocole utilisé pour la transmission des trames est Ethernet II.

ETH_R_IP_MODE : Codes de source d'adresse IP

Description du type énumération

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

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

ETH_R_PORT_DUPLEX_STATUS : Codes de mode de transmission

Description du type énumération

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

Enumérateur	Valeur	Commentaire
<i>ETH_R_PORT_HALF_DUPLEX</i>	00 hex	Le mode de transmission en semi-duplex est utilisé.
<i>ETH_R_FULL_DUPLEX</i>	01 hex	Le mode de transmission en duplex intégral est utilisé.
<i>ETH_R_PORT_NA_DUPLEX</i>	03 hex	Le mode de transmission sans duplex est utilisé.

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

Description du type énumération

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

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

ETH_R_PORT_LINK_STATUS : Codes d'état de liaison de communication

Description du type énumération

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

Enumérateur	Valeur	Commentaire
<i>ETH_R_LINK_DOWN</i>	00 hex	Liaison de communication non disponible pour un autre équipement.
<i>ETH_R_LINK_UP</i>	01 hex	Liaison de communication disponible pour un autre équipement.

***ETH_R_PORT_SPEED* : Codes de vitesse de communication des ports Ethernet**

Description du type énumération

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

Enumérateur	Valeur	Commentaire
<i>ETH_R_SPEED_NA</i>	0 déc	Le débit réseau n'est pas disponible.
<i>ETH_R_SPEED_10_MB</i>	10 déc	Le débit réseau est de 10 mégabits par seconde.
<i>ETH_R_100_MB</i>	100 déc	Le débit réseau est de 100 mégabits par seconde.

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

Description du type énumération

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

Enumérateur	Valeur	Commentaire
<i>IDLE</i>	00 hex	La connexion EtherNet/IP est au repos.
<i>RUN</i>	01 hex	La connexion EtherNet/IP est en fonctionnement.

Types de données des variables système *TM3_MODULE_RW*

Présentation

Cette section répertorie et décrit les types de données de variable système, inclus dans les structures *TM3_MODULE_R* et *TM3_MODULE_W*.

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

Description du type énumération

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

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

Description

TM3_MODULE_R_ARRAY_TYPE est un tableau de 0 à 13 variables de type *TM3_MODULE_R_STRUCT*.

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

Description du type énumération

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

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

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 :

Enumérateur	Valeur	Commentaire
<i>IOBUS_ERR_ACTIVE</i>	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 Description générale de la configuration des E/S (voir Modicon M241 Logic Controller, Guide de programmation).
<i>IOBUS_ERR_PASSIVE</i>	01 hex	Mode passif. Les échanges d'E/S se poursuivent sur le bus TM3, même si une erreur est détectée.

Types de données des variables système des cartouches

Présentation

Cette section répertorie et décrit les types de données des variables système comprises dans la structure *Cartridge*.

CART_R_ARRAY_TYPE : Type tableau de lecture de cartouche

Description

CART_R_ARRAY_TYPE est un tableau de 0 ou 1 variables de type *CART_R_STRUCT*.

CART_R_MODULE_ID : Identificateur de module de lecture de cartouche

Description du type énumération

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

Enumérateur	Valeur	Description
<i>CART_R_MODULE_ID</i>	40 hex	TMC4AI2
<i>CART_R_MODULE_ID</i>	41 hex	TMC4AQ2
<i>CART_R_MODULE_ID</i>	42 hex	TMC4TI2
<i>CART_R_MODULE_ID</i>	48 hex	TMC4HOIS01
<i>CART_R_MODULE_ID</i>	49 hex	TMC4PACK01
<i>CART_R_MODULE_ID</i>	FF hex	Aucun(e)

CART_R_STATE : Etat de lecture de cartouche

Description du type énumération

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

Enumérateur	Valeur	Commentaire
<i>CONFIGURED</i>	00 hex	La cartouche est configurée.
<i>INITIALIZED_NOT_CONFIGURED</i>	01 hex	La cartouche est initialisée mais pas configurée.
<i>NOT_INITIALIZED</i>	02 hex	La cartouche n'est pas initialisée.

Types de données des fonctions système

Présentation

Cette section décrit les différents types de données des fonctions système de la bibliothèque PLCSystem de M241.

IMMEDIATE_ERR_TYPE : GetImmediateFastInput - Lire l'entrée des codes d'E/S expertes intégrées

Description du type énumération

Le type de données énumération contient les valeurs suivantes :

Enumérateur	Type	Commentaire
<i>IMMEDIATE_NO_ERROR</i>	<i>WORD</i>	Aucune erreur détectée.
<i>IMMEDIATE_UNKNOWN</i>	<i>WORD</i>	La référence de la fonction <i>Immediate</i> est incorrecte ou non configurée.
<i>IMMEDIATE_UNKNOWN_PARAMETER</i>	<i>WORD</i>	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
<i>RTC_OK</i>	00 hex	L'écart de l'horodateur est configuré correctement.
<i>RTC_BAD_DAY</i>	01 hex	Non utilisé.
<i>RTC_BAD_HOUR</i>	02 hex	Non utilisé.
<i>RTC_BAD_MINUTE</i>	03 hex	Non utilisé.
<i>RTC_BAD_DRIFT</i>	04 hex	Paramètre d'écart de l'horodateur hors limites.
<i>RTC_INTERNAL_ERROR</i>	05 hex	Paramètres d'écart de l'horodateur rejetés sur détection d'une erreur interne.

Annexes

Contenu de cette partie

Représentation des fonctions et blocs fonction	62
--	----

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	62
Utilisation d'une fonction ou d'un bloc fonction en langage IL	63
Utilisation d'une fonction ou d'un bloc fonction en langage ST	66

Présentation

Chaque fonction peut être représentée dans les langages suivants :

- IL : (Instruction List) liste d'instructions
- ST : (Structured Text) littéral structuré
- LD : (Ladder Diagram) schéma à contacts
- FBD : Function Block Diagram (Langage à blocs fonction)
- CFC : Continuous Function Chart (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 :

```

1  PROGRAM MyProgram_ST
2  VAR
3      Timer_ON: TON; // Function Block Instance
4      Timer_RunCd: BOOL;
5      Timer_PresetValue: TIME := T#5S;
6      Timer_Output: BOOL;
7      Timer_ElapsedTime: TIME;
8  END_VAR

1  Timer_ON(
2      IN:=Timer_RunCd,
3      PT:=Timer_PresetValue,
4      Q=>Timer_Output,
5      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.

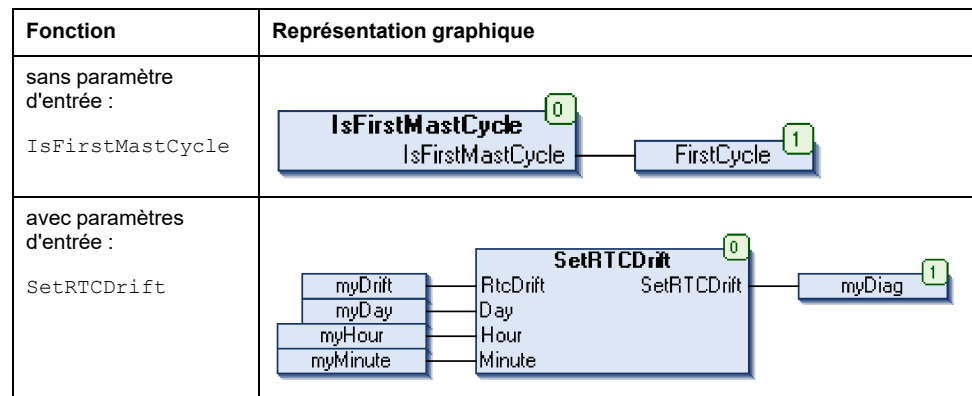
Les fonctions `IsFirstMastCycle` et `SetRTCDrift` ainsi que le bloc fonction `TON` sont utilisés à titre d'exemple pour illustrer les implémentations.

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 :



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

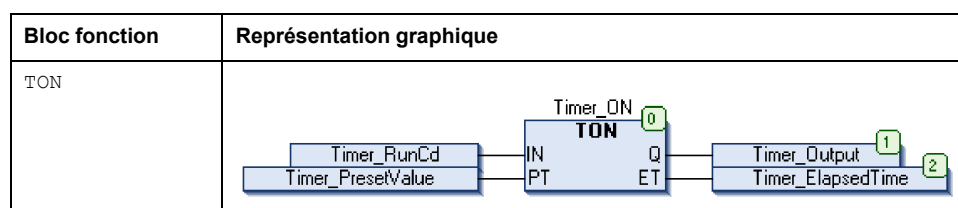
Fonction	Représentation dans l'éditeur IL de POU
Exemple en IL d'une fonction sans paramètre d'entrée : <code>IsFirstMastCycle</code>	<pre> 1 PROGRAM MyProgram_IL 2 VAR 3 FirstCycle: BOOL; 4 END_VAR 5 </pre> <pre> 1 IsFirstMastCycle ST FirstCycle </pre>
Exemple en IL d'une fonction avec paramètres d'entrée : <code>SetRTCDrift</code>	<pre> 1 PROGRAM MyProgram_IL 2 VAR 3 myDrift: SINT (-29..29) := 5; 4 myDay: DAY_OF_WEEK := SUNDAY; 5 myHour: HOUR := 12; 6 myMinute: MINUTE; 7 myDiag: RTCSETDRIFT_ERROR; 8 END_VAR 9 </pre> <pre> 1 LD myDrift SetRTCDrift myDay myHour myMinute ST myDiag </pre>

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 :



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	<pre> 1 PROGRAM MyProgram_IL 2 VAR 3 Timer_ON: TON; // Function Block instance declaration 4 Timer_RunCd: BOOL; 5 Timer_PresetValue: TIME := T#5S; 6 Timer_Output: BOOL; 7 Timer_ElapsedTime: TIME; 8 END_VAR 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 </pre> <pre> 1 CAL Timer_ON(IN:= Timer_RunCd, PT:= Timer_PresetValue, Q=> Timer_Output, ET=> Timer_ElapsedTime) </pre>

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.

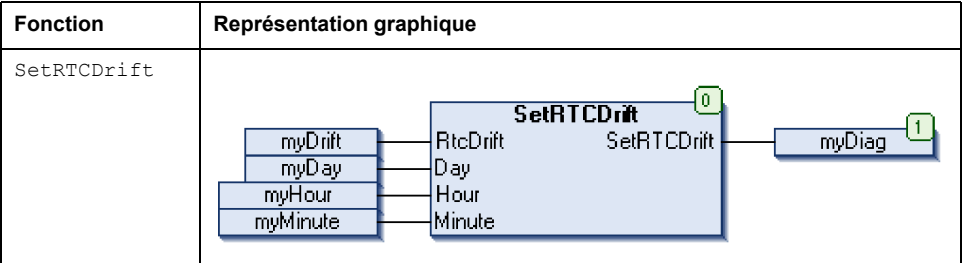
La fonction `SetRTCDrift` et le bloc fonction `TON` sont utilisés comme exemples pour illustrer les implémentations.

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 : <code>FunctionResult:= FunctionName (VarInput1, VarInput2,.. VarInputx);</code>

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



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

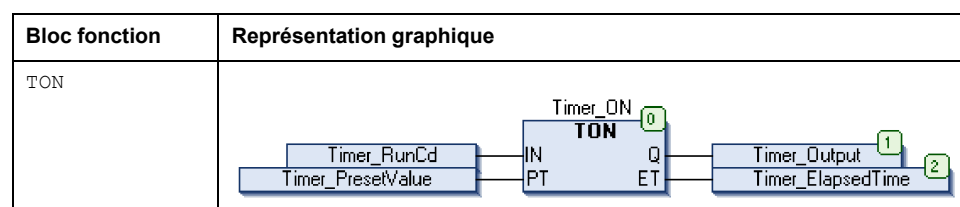
Fonction	Représentation dans l'éditeur ST de POU
SetRTCDrift	<pre>PROGRAM MyProgram_ST VAR myDrift: SINT (-36..+73) := 5; myDay: sec.DAY_OF_WEEK := SUNDAY; myHour: sec.HOUR := 12; myMinute: sec.MINUTE; myRTCAdjust: sec.RTCDRIFT_ERROR; END_VAR myRTCAdjust:= SetRTCDrift(myDrift, myDay, myHour, myMinute);</pre>

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 : <pre>FunctionBlock_InstanceName(Input1:=VarInput1, Input2:=VarInput2, ... Ouput1=>VarOutput1, Ouput2=>VarOutput2, ...);</pre>

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



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

Bloc fonction	Représentation dans l'éditeur ST de POU
TON	<pre> 1 PROGRAM MyProgram_ST 2 VAR 3 Timer_ON: TON; // Function Block Instance 4 Timer_RunCd: BOOL; 5 Timer_PresetValue: TIME := T#5S; 6 Timer_Output: BOOL; 7 Timer_ElapsedTime: TIME; 8 END_VAR 1 Timer_ON(2 IN:=Timer_RunCd, 3 PT:=Timer_PresetValue, 4 Q=>Timer_Output, 5 ET=>Timer_ElapsedTime); </pre>

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.

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**élément:**

Raccourci pour l'élément d'un ARRAY.

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).

LED:

Acronyme de *light emitting diode*, diode électroluminescente (DEL). Indicateur qui s'allume sous l'effet d'une charge électrique de faible niveau.

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**PCI:**

Acronyme de *Peripheral Component Interconnect*, interconnexion de composants périphériques. Standard industriel de bus pour la connexion de périphériques.

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 POU facilitent la réutilisation modulaire de programmes logiciels, de fonctions et de blocs fonction. Une fois déclarées, les POU 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 partagent un chemin de données et un protocole de communications communs.

RTC:

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

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.

SINT:

Abréviation de *signed integer*, entier signé. Valeur sur 15 bits plus signe.

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	
FB_ControlClone	35

C

CART_R_ARRAY_TYPE	
Types de données.....	58
CART_R_MODULE_ID	
Types de données.....	58
CART_R_STATE	
Types de données.....	59
CART_R_STRUCT	
Variable système	26
commandes de copie de fichier	
DataFileCopy.....	36
commandes de script	
ExecuteScript	38
cycle	
IsFirstMastColdCycle	29
IsFirstMastCycle	30
IsFirstMastWarmCycle	31

D

DataFileCopy	
copie de données vers ou depuis un fichier	36
DataFileCopyError	
Types de données.....	53
DataFileCopyLocation	
Types de données.....	53
DAY OF WEEK	34

E

entrée/sortie	
HasForcedIo.....	28
E/S intégrée	
GetImmediateFastInput.....	27
PhysicalWriteFastOutputs	33
état du bus du module TM3	
TM3_GetModuleBusStatus.....	42
état interne du module TM3	
TM3_GetModuleInternalStatus	44
ETH_R	
variable système.....	21
ETH_R_FRAME_PROTOCOL	
Types de données.....	54
ETH_R_IP_MODE	
Types de données.....	54
ETH_R_PORT_DUPLEX_STATUS	
Types de données.....	55
ETH_R_PORT_LINK_STATUS	
Types de données.....	55
ETH_R_PORT_SPEED	
Types de données.....	56
ETH_R_STRUCT.....	21
ETH_W	
variable système.....	24
ETH_W_STRUCT.....	24
ExecuteScript	
exécution de commandes de script.....	38
ExecuteScriptError	

Types de données.....54

F

FB_ControlClone	
bloc fonction	35
FC_GetFreeDiskSpace	
fonction	39
FC_GetLabel	
fonction	40
FC_GetTotalDiskSpace	
fonction	41
fonction	
FC_GetFreeDiskSpace	39
FC_GetLabel	40
FC_GetTotalDiskSpace	41
fonctions	
différences entre fonction et bloc fonction	62
utilisation d'une fonction ou d'un bloc fonction en	
langage IL	63
utilisation d'une fonction ou d'un bloc fonction en	
langage ST	66

G

GetImmediateFastInput	
obtention de la valeur d'une entrée rapide	27
GetRtc	
obtention de la valeur de l'horodateur (RTC)	28

H

HasForcedIo	
indiquer si une entrée/sortie est forcée	28
horodateur	
GetRtc	28
SetRTCDrift	34
Horodateur	
GetRtc	28
HOURL	34

1

IMMEDIATE_ERR_TYPE	
Types de données.....	59
InhibitBatLed	
Enabling or disabling the Battery led	32
IsFirstMastColdCycle	
premier cycle de démarrage à froid	29
IsFirstMastCycle	
premier cycle MAST	30
IsFirstMastWarmCycle	
premier cycle de démarrage à chaud	31

M

M241 PLCSystem	
DataFileCopy	36
ExecuteScript	38
GetImmediateFastInput	27
GetRtc	28
HasForcedIo	28
InhibitBatLed	32
IsFirstMastColdCycle	29
IsFirstMastCycle	30
IsFirstMastWarmCycle	31

PhysicalWriteFastOutputs	33
SetRTCDrift	34
TM3_GetModuleBusStatus	44
MINUTE	34

P

PhysicalWriteFastOutputs	
writing output of an embedded expert I/O	33
PLC_GVL	13
PLC_R	
variable système	16
PLC_R_APPLICATION_ERROR	
Types de données	48
PLC_R_BOOT_PROJECT_STATUS	
Types de données	49
PLC_R_IO_STATUS	
Types de données	49
PLC_R_SDCARD_STATUS	
Types de données	50
PLC_R_STATUS	
Types de données	50
PLC_R_STOP_CAUSE	
Types de données	51
PLC_R_STRUCT	16
PLC_R_TERMINAL_PORT_STATUS	
Types de données	52
PLC_R_TM3_BUS_STATE	
Types de données	52
PLC_W	
variable système	19
PLC_W_COMMAND	
Types de données	52
PLC_W_STRUCT	19
PROFIBUS_R	
variable système	25
PROFIBUS_R_STRUCT	25

R

RTCSETDRIFT_ERROR	
Types de données	59

S

SERIAL_R	
variable système	19
SERIAL_R_STRUCT	19
SERIAL_W	
variable système	20
SERIAL_W_STRUCT	20
SetRTCDrift	
accélération ou ralentissement de la fréquence	
horodateur	34

T

TM3_BUS_W	
variable système	25
TM3_BUS_W_IOBUSERRMOD	
Types de données	58
TM3_BUS_W_IOBUSINIT	25
TM3_BUS_W_STRUCT	25
TM3_ERR_CODE	
Types de données	57
TM3_GetModuleBusStatus	
obtention de l'état du bus d'un module TM3	42

TM3_GetModuleFWVersion	
obtention de la version de micrologiciel d'un module	
TM3	43
TM3_GetModuleInternalStatus	
obtention de l'état interne d'un module TM3	44
TM3_MODULE_R	
variable système	24
TM3_MODULE_R_ARRAY_TYPE	
Types de données	57
TM3_MODULE_R_STRUCT	24
TM3_MODULE_STATE	
Types de données	57
Types de données	
CART_R_ARRAY_TYPE	58
CART_R_MODULE_ID	58
CART_R_STATE	59
DataFileCopyError	53
DataFileCopyLocation	53
ETH_R_FRAME_PROTOCOL	54
ETH_R_IP_MODE	54
ETH_R_PORT_DUPLEX_STATUS	55
ETH_R_PORT_IP_STATUS	55
ETH_R_PORT_LINK_STATUS	55
ETH_R_PORT_SPEED	56
ETH_R_RUN_IDLE	56
ExecuteScriptError	54
IMMEDIATE_ERR_TYPE	59
PLC_R_APPLICATION_ERROR	48
PLC_R_BOOT_PROJECT_STATUS	49
PLC_R_IO_STATUS	49
PLC_R_SDCARD_STATUS	50
PLC_R_STATUS	50
PLC_R_STOP_CAUSE	51
PLC_R_TERMINAL_PORT_STATUS	52
PLC_R_TM3_BUS_STATE	52
PLC_W_COMMAND	52
RTCSETDRIFT_ERROR	59
TM3_BUS_W_IOBUSERRMOD	58
TM3_ERR_CODE	57
TM3_MODULE_R_ARRAY_TYPE	57
TM3_MODULE_STATE	57

V

variable système	
ETH_R	21
ETH_W	24
PLC_R	16
PLC_W	19
PROFIBUS_R	25
SERIAL_R	19
SERIAL_W	20
TM3_BUS_W	25
TM3_MODULE_R	24
Variable système	
CART_R_STRUCT	26
variables système	
Utilisation	14
Variables système	
Définition	13
version de micrologiciel des modules TM3	
TM3_GetModuleFWVersion	43
voyant de batterie	
InhibitBatLed	32

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.

EIO0000003066.06