

Modicon LMC078

Motion Controller

Fonctions et variables système

Guide de la bibliothèque PLCSystem

09/2016

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

Aucune partie de ce document ne peut être reproduite sous quelque forme ou par quelque moyen que ce soit, électronique, mécanique ou photocopie, sans autorisation préalable de Schneider Electric.

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

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

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

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

© 2016 Schneider Electric. Tous droits réservés.

Table des matières



	Consignes de sécurité	5
	A propos de ce manuel	7
Chapitre 1	Types de données système LMC078	9
	ST_RetainImageInfo : informations supplémentaires sur une image mémoire	9
Chapitre 2	Fonctions du système LMC078	11
2.1	Diagnostic	12
	FC_DiagConfigRead2 : lecture de la configuration du diagnostic	13
	FC_DiagConfigSet2 : modification de la configuration du diagnostic	15
	FC_DiagMsgRead : lecture des informations de diagnostic	17
	FC_DiagUserMsgWrite : déclenchement du message de diagnostic	19
	FC_DiagQuit : réinitialisation du message de diagnostic	20
2.2	IEC_Tasks	21
	FC_CycleCheckSet : activation/désactivation des messages de diagnostic pour dépassement de durée de cycle <small>(OVL/OVL/OVL)</small>	22
	FC_CycleCheckTimeGet : identification des valeurs temporelles pour la surveillance du temps de cycle de la tâche CEI	24
	FC_CycleCheckTimeSet : changement de la valeur temporelle pour la surveillance du temps de cycle de la tâche CEI	25
	FC_LzsTaskGetCurrentInterval : définition de l'intervalle de la tâche dans le programme CEI	26
	FC_LzsTaskGetInterval : lecture de l'intervalle de la tâche dans le programme CEI	28
	FC_PrgResetAndStart : réinitialisation et démarrage de l'application en utilisant le programme utilisateur	29
2.3	LogicalAddress	30
	FC_CompareStLogicalAddress : comparaison de deux adresses logiques, i_stLogAddr1 et i_stLogAddr2	31
	FC_IsStLogicalAddressValid : vérification de la validation de l'adresse logique	32
2.4	MessageLogger	33
	FC_MsgLogSave : stockage du fichier journal des messages dans la mémoire de masse	33

2.5	RemoteFile	35
	FC_RemoteDeviceCreate : configuration d'un équipement pour les services de fichiers	36
	FC_RemoteUserIdSet : configuration du nom utilisateur et du mot de passe des services de fichiers distants	38
2.6	Conservation	41
	FC_CheckProgramIdent : vérification de l'adéquation de l'image mémoire avec l'application avant de la charger	42
	FC_GetRetainImageInfo : lecture des informations complémentaires à écrire dans le fichier d'une image mémoire	44
	FC_RetainImageLoad : chargement de l'image mémoire de la mémoire de conservation (se trouvant dans un fichier dans la mémoire de conservation du contrôleur)	46
	FC_RetainImageSave : enregistrement du contenu de la mémoire de conservation dans un fichier	49
2.7	System	51
	FC_GetNVRamStatus : vérification de la validité des données dans la NVRam lors du démarrage du contrôleur	52
	FC_GetBootState : vérification de la validité de tous les paramètres après le redémarrage du contrôleur	53
	FC_SysReset : réinitialisation du contrôleur	55
	FC_SysSaveParameter : enregistrement des paramètres personnalisés sur la carte SD	56
	FC_SysShutdown : vérification de la sécurité du système de fichiers avant la mise hors tension du contrôleur	57
	FC_UserChangePassword : Changement du mot de passe de l'utilisateur	58
2.8	TimeAndSync	60
	FC_GetNsPerCPUClockCycle : mesure haute précision du temps	61
	FC_GetTimeOfDay : lecture de l'heure du système en ms sans la date	62
	FC_GetTSC : mesure haute précision du temps	64
2.9	VolumeOperations	65
	FC_GetFreeDiskSpace : lecture de l'espace mémoire disponible d'un support mémoire	66
	FC_GetTotalDiskSpace : lecture de la taille d'un support mémoire	68
	Glossaire	71
	Index	73

Consignes de sécurité



Informations importantes

AVIS

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



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



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

DANGER

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

AVERTISSEMENT

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

ATTENTION

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

AVIS

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

REMARQUE IMPORTANTE

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

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

A propos de ce manuel



Présentation

Objectif du document

Ce document présente les fonctions et variables système proposées par le contrôleur LMC078 Motion Controller. La bibliothèque LMC078 PLCSystem contient des fonctions et des variables qui permettent d'obtenir des informations et d'envoyer des commandes au système du contrôleur. Ce document décrit les types de données, les fonctions et les variables de cette bibliothèque.

Les connaissances de base requises sont les suivantes :

- connaissances de base sur les fonctionnalités, la structure et la configuration du contrôleur LMC078
- programmation en langages FBD, LD, ST, IL ou CFC
- variables système (variables globales)

Champ d'application

Ce document a été actualisé pour le lancement de SoMachine V4.2.

Document(s) à consulter

Titre de documentation	Référence
Modicon LMC078 Motion Controller - Guide de programmation	EIO0000001909 (ENG) EIO0000001910 (FRE) EIO0000001911 (GER) EIO0000001912 (SPA) EIO0000001913 (ITA) EIO0000001914 (CHS) EIO0000001916 (TUR)

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

AVERTISSEMENT

PERTE DE CONTROLE

- Le concepteur d'un système de commande doit envisager les modes de défaillance possibles des chemins de commande et, pour certaines fonctions de commande critiques, prévoir un moyen d'atteindre un état sécurisé en cas de défaillance d'un chemin, et après cette défaillance. Par exemple, l'arrêt d'urgence, l'arrêt en cas de surcourse, la coupure de courant et le redémarrage sont des fonctions de commande cruciales.
- Des canaux de commande séparés ou redondants doivent être prévus pour les fonctions de commande critiques.
- Les chemins de commande système peuvent inclure les liaisons de communication. Une attention particulière doit être prêtée aux implications des délais de transmission non prévus ou des pannes de la liaison.
- Respectez toutes les réglementations de prévention des accidents ainsi que les consignes de sécurité locales.¹
- Chaque implémentation de cet équipement doit être testée individuellement et entièrement pour s'assurer du fonctionnement correct avant la mise en service.

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

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

AVERTISSEMENT

FONCTIONNEMENT INATTENDU 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.

Chapitre 1

Types de données système LMC078

ST_RetainImageInfo : informations supplémentaires sur une image mémoire

Structure de la variable

Le tableau suivant décrit les paramètres de la variable système ST_RetainImageInfo (ST_RetainImageInfo_STRUCT type) :

%MW	Nom de la variable	Type	Commentaire
s/o	sFileName	STRING[255]	Nom de fichier de l'image conservée (retain) au format 8.3.
s/o	dtRetainSaveDateTime	DATE_AND_TIME	Date de création de l'image conservée au format Date_and_Time. S'il n'y a pas d'horodateur (RTC) sur le contrôleur, la valeur DT#1970-01-01-00:00 est stockée dans la variable.
s/o	sHardwareType	STRING[80]	Type de contrôleur.
s/o	sSetRetainSize	STRING[80]	Taille de la mémoire de conservation (retain).
s/o	sUserInfo	STRING[255]	Texte qui a été indiqué lors de la création de l'image.
s/o	sMD5Checksum	STRING[33]	Valeur du total de contrôle des données conservées.
s/o	diLengthRetainData	DINT	Nombre de données conservées en octets.

Chapitre 2

Fonctions du système LMC078

Présentation

Ce chapitre décrit les fonctions système disponibles dans la bibliothèque Modicon LMC078 Motion Controller PLCSystem.

Contenu de ce chapitre

Ce chapitre contient les sous-chapitres suivants :

Sous-chapitre	Sujet	Page
2.1	Diagnostic	12
2.2	IEC_Tasks	21
2.3	LogicalAddress	30
2.4	MessageLogger	33
2.5	RemoteFile	35
2.6	Conservation	41
2.7	System	51
2.8	TimeAndSync	60
2.9	VolumeOperations	65

Sous-chapitre 2.1

Diagnostic

Présentation

Cette section décrit les fonctions disponibles sous la rubrique Diagnostic.

Contenu de ce sous-chapitre

Ce sous-chapitre contient les sujets suivants :

Sujet	Page
FC_DiagConfigRead2 : lecture de la configuration du diagnostic	13
FC_DiagConfigSet2 : modification de la configuration du diagnostic	15
FC_DiagMsgRead : lecture des informations de diagnostic	17
FC_DiagUserMsgWrite : déclenchement du message de diagnostic	19
FC_DiagQuit : réinitialisation du message de diagnostic	20

FC_DiagConfigRead2 : lecture de la configuration du diagnostic

Description de la fonction

La fonction FC_DiagConfigRead2 lit la configuration du diagnostic. La classe et la sous-classe du diagnostic numéro `i_diDiagCode` sont lues pour tous les équipements ou pour des équipements individuels.

Représentation graphique



Représentation en langage IL et ST

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

Description des variables d'E/S

Le tableau suivant décrit les variables d'entrée :

Entrée	Type	Commentaire
<code>i_diDiagCode</code>	<code>DINT</code>	Numéro du diagnostic
<code>i_stLogAddr</code>	<code>ST_LogicalAddress</code>	Adresse logique de l'équipement

Le tableau suivant décrit les variables de sortie :

Sortie	Type	Commentaire
<code>FC_DiagConfigRead2</code>	<code>DINT</code>	Voir le tableau ci-après.

Le tableau suivant décrit les valeurs renvoyées :

Valeur	Description
0	Action accomplie avec succès
-1	Numéro de diagnostic non valide

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

Entrée/Sortie	Type	Commentaire
iq_diDiagClass	DINT	Classe de diagnostic
iq_diDiagSubClass	DINT	Sous-classe de diagnostic

Exemple

Déclaration

```
VAR
    diDiagCode: DINT := 8788;
    diDiagClass: DINT := 0;
    DiagSubClass: DINT := 0;
    xRead: BOOL := FALSE;
END_VAR
```

Programme

```
IF xRead THEN
    SEC.FC_DiagConfigRead2;
    i_diDiagCode := diDiagCode;
    i_stLogAddr := MyController.stLogicalAdress;
    iq_diDiagClass := diDiagClass;
    iq_diDiagSubClass := diDiagSubClass;
END_IF
```

FC_DiagConfigSet2 : modification de la configuration du diagnostic

Description de la fonction

La fonction FC_DiagConfigSet2 permet de modifier la configuration du message de diagnostic numéro `i_diDiagCode`. La configuration du message de diagnostic contient la classe et la sous-classe du diagnostic.

La réaction du système au message de diagnostic est définie via la classe de diagnostic qui est déclenchée par un certain message de diagnostic. Les sous-classes constituent une subdivision plus précise des classes de diagnostic.

La configuration d'un message de diagnostic peut être modifiée :

- Dans l'ensemble du système (pour tous les objets). La variable d'entrée `i_stLogAddr` a la valeur `Gc_stLogAddrAllTypes`.
- Pour un objet ou un groupe d'objets spécifique. La variable d'entrée `i_stLogAddr` contient l'adresse logique d'un objet ou d'un groupe d'objets. La configuration n'est effectuée que pour cet objet ou ce groupe d'objets.

Si `i_diDiagClass = 3`, `i_diDiagSubClass` doit être spécifié. A défaut, cette variable d'entrée est ignorée.

Représentation graphique



Représentation en langage IL et ST

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

Description des variables d'E/S

Le tableau suivant décrit les variables d'entrée :

Entrée	Type	Commentaire
<code>i_diDiagCode</code>	DINT	Numéro du message de diagnostic à modifier
<code>i_diDiagClass</code>	DINT	Nouvelle classe du message de diagnostic
<code>i_stLogAddr</code>	ST_LogicalAddress	Adresse logique de l'équipement
<code>i_diDiagSubClass</code>	DINT	Nouvelle sous-classe du message de diagnostic

Le tableau suivant décrit la variable de sortie :

Sortie	Type	Commentaire
FC_DiagConfigSet2	DINT	Voir le tableau ci-après.

Le tableau suivant décrit les valeurs renvoyées :

Valeur	Description
0	Action accomplie avec succès
-1	Numéro de diagnostic non valide
-2	Classe de diagnostic non valide
-4	Classe de diagnostic non valide (seules les classes 1, 2 et 3 peuvent être affectées aux codes de diagnostic de l'unité avec la réaction A, B ou C).
-461	Dans la phase actuelle, la lecture de paramètres via la voie de service n'est pas prise en charge.
-462	L'axe concerné ne prend pas en charge cette fonction.

Exemple

Déclaration

```
VAR
    diDiagCode: DINT := 8788;
    diDiagClass: DINT := 0;
    DiagSubClass: DINT := 0;
    xSet: BOOL := FALSE;
END_VAR
```

Programme

```
IF xSet THEN
    SEC.FC_DiagConfigSet2;
    i_diDiagCode := diDiagCode;
    iq_diDiagClass := 2;
    i_stLogAddr := MyController.stLogicalAdress;
    iq_diDiagSubClass := 0;
END_IF
```


FC_DiagMsgRead : lecture des informations de diagnostic

Description de la fonction

La fonction FC_DiagMsgRead lit et efface la mémoire tampon de diagnostic (tampon FIFO) du contrôleur. Elle attend la réception d'un message de diagnostic ou l'expiration du délai indiqué dans `i_diTimeout`. Si `i_diTimeout = 0`, la fonction attend indéfiniment un message.

La mémoire tampon peut stocker environ 320 messages. Le message le plus ancien est proposé le premier en lecture. Si la mémoire tampon est vide ou que le délai d'attente a expiré, la fonction renvoie zéro pour tous les résultats. La lecture de la mémoire tampon ne supprime pas les messages de diagnostic en attente. Vous pouvez lire tous les messages de diagnostic des classes 1 à 7 depuis le démarrage (ou redémarrage) du contrôleur.

Représentation graphique



Représentation en langage IL et ST

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

Description des variables d'E/S

Le tableau suivant décrit la variable d'entrée :

Entrée	Type	Commentaire
<code>i_diTimeout</code>	DINT	Durée d'attente maximum en millisecondes. Si <code>i_diTimeout = 0</code> , la fonction attend indéfiniment un message.

Le tableau suivant décrit la variable de sortie :

Sortie	Type	Commentaire
<code>FC_DiagMsgRead</code>	DINT	Voir le tableau ci-après.

Le tableau suivant décrit les valeurs renvoyées :

Valeur	Description
0	Action accomplie avec succès
-3	Délai d'attente expiré

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

Entrée/Sortie	Type	Commentaire
iq_diNr	DINT	Numéro du diagnostic
iq_diClass	DINT	Classe de diagnostic
iq_stLogAdr	ST_LogicalAddress	Adresse logique du contrôleur

FC_DiagUserMsgWrite : déclenchement du message de diagnostic

Description de la fonction

La fonction `FC_DiagUserMsgWrite` permet de déclencher des messages de diagnostic à partir du programme. La classe de diagnostic (0 à 4) du message est définie par le paramètre `i_diDiagClass`. Le numéro de diagnostic est transféré dans `DiagCode`. La plage 8850 à 8899 des numéros de diagnostic système est réservée. Utilisez `i_sDiagText` pour transférer le texte du diagnostic. La longueur est limitée à 55 caractères.

Représentation graphique



Représentation en langage IL et ST

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

Description des variables d'E/S

Le tableau suivant décrit la variable d'entrée :

Entrée	Type	Commentaire
<code>i_diDiagClass</code>	DINT	Classe de diagnostic
<code>i_diDiagCode</code>	DINT	Numéro du diagnostic
<code>i_sDiagText</code>	STRING(80)	Texte du diagnostic (55 caractères maximum)

Le tableau suivant décrit la variable de sortie :

Sortie	Type	Commentaire
<code>FC_DiagUserMsgWrite</code>	DINT	Voir le tableau ci-après.

Le tableau suivant décrit les valeurs renvoyées :

Valeur	Description
0	Action accomplie avec succès
-2	Classe de diagnostic non valide
-5	Numéro de message non valide

FC_DiagQuit : réinitialisation du message de diagnostic

Description de la fonction

La fonction `FC_DiagQuit` réinitialise le message de diagnostic en attente. Ce message s'affiche dans les paramètres `DiagCode` et `DiagMsg`.

Une perte de position des axes et du codeur physique (SinCos, codeur incrémentiel) est possible à la suite de l'acquiescement de l'erreur du message de diagnostic 8506 (communication impossible avec le maître SERCOS) sans redémarrage du contrôleur.

⚠ ATTENTION

PERTE DE POSITION DUE A UNE ERREUR DE BUS

N'acquiescez le message de diagnostic qu'après réinitialisation ou référencement du système.

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

NOTE : Vous pouvez réinitialiser le système en procédant à une mise hors tension puis sous tension, ou bien par programmation à l'aide de la fonction `FC_SysReset()` ou `FC_PrgResetAndStart()`.

Représentation graphique



Représentation en langage IL et ST

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

Description des variables d'E/S

Le tableau suivant décrit la variable de sortie :

Sortie	Type	Commentaire
<code>FC_DiagQuit</code>	DINT	Voir le tableau ci-après.

Le tableau suivant décrit la valeur renvoyée :

Valeur	Description
0	Action accomplie avec succès

Sous-chapitre 2.2

IEC_Tasks

Présentation

Cette section décrit les fonctions disponibles dans la rubrique IEC_Tasks.

Contenu de ce sous-chapitre

Ce sous-chapitre contient les sujets suivants :

Sujet	Page
FC_CycleCheckSet : activation/désactivation des messages de diagnostic pour dépassement de durée de cycle	22
FC_CycleCheckTimeGet : identification des valeurs temporelles pour la surveillance du temps de cycle de la tâche CEI	24
FC_CycleCheckTimeSet : changement de la valeur temporelle pour la surveillance du temps de cycle de la tâche CEI	25
FC_LzsTaskGetCurrentInterval : définition de l'intervalle de la tâche dans le programme CEI	26
FC_LzsTaskGetInterval : lecture de l'intervalle de la tâche dans le programme CEI	28
FC_PrgResetAndStart : réinitialisation et démarrage de l'application en utilisant le programme utilisateur	29

FC_CycleCheckSet : activation/désactivation des messages de diagnostic pour dépassement de durée de cycle

Description de la fonction

La fonction `FC_CycleCheckSet` active et désactive le message de diagnostic 8317 indiquant un dépassement de durée de cycle.

Si `i_xCheck = FALSE`, les messages de diagnostic sont désactivés.

Si `i_xCheck = TRUE`, les messages de diagnostic sont activés.

L'activation des messages de diagnostic prend effet au démarrage du cycle suivant.

NOTE : L'appel de cette fonction insère une entrée dans le journaliseur de messages.

NOTE : Pour que l'horloge de surveillance (chien de garde) reste fonctionnelle, utilisez plutôt la fonction `FC_CycleCheckTimeSet()`.

Représentation graphique



Représentation en langage IL et ST

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

Description des variables d'E/S

Le tableau suivant décrit la variable d'entrée :

Entrée	Type	Commentaire
i_xCheck	BOOL	FALSE : Les messages de diagnostic sont désactivés. TRUE : Les messages de diagnostic sont activés.

Le tableau suivant décrit la variable de sortie :

Sortie	Type	Commentaire
FC_CycleCheckSet	DINT	Voir le tableau ci-après.

Le tableau suivant décrit la valeur renvoyée :

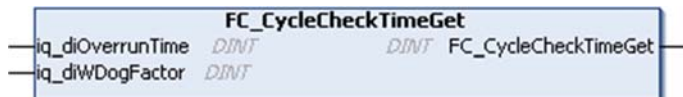
Valeur	Description
0	Action accomplie avec succès

FC_CycleCheckTimeGet : identification des valeurs temporelles pour la surveillance du temps de cycle de la tâche CEI

Description de la fonction

La fonction FC_CycleCheckTimeGet fournit les valeurs temporelles de surveillance de durée de cycle pour la tâche CEI qui l'appelle.

Représentation graphique



Représentation en langage IL et ST

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

Description des variables d'E/S

Le tableau suivant décrit les variables de sortie :

Entrée/Sortie	Type	Commentaire
FC_CycleCheckTimeGet	DINT	Voir le tableau ci-après.

Le tableau suivant décrit les valeurs renvoyées :

Valeur	Description
0	Action accomplie avec succès
-1	Paramètre incorrect

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

Entrée/Sortie	Type	Commentaire
iq_diOverrunTime	DINT	Valeur pour la surveillance de la durée de cycle standard, en millisecondes.
iq_diWDogFactor	DINT	Valeur du facteur d'horloge de surveillance (chien de garde).

FC_CycleCheckTimeSet : changement de la valeur temporelle pour la surveillance du temps de cycle de la tâche CEI

Description de la fonction

La fonction `FC_CycleCheckTimeSet` permet de manipuler les valeurs temporelles de surveillance de durée de cycle pour la tâche CEI qui l'appelle.

La tâche continue d'être appelée selon la fréquence définie dans sa configuration.

Représentation graphique



Représentation en langage IL et ST

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

Description des variables d'E/S

Le tableau suivant décrit les variables d'entrée :

Entrée	Type	Commentaire
<code>i_diOverrunTime</code>	DINT	Valeur pour la surveillance de la durée de cycle standard, en millisecondes. Elle détermine la valeur de la vérification de la durée de cycle (message de diagnostic 8317, dépassement de durée de cycle).
<code>i_diWDogFactor</code>	DINT	Valeur du facteur d'horloge de surveillance (chien de garde). Elle détermine la valeur d'une erreur grave de dépassement de durée de cycle (message de diagnostic 313, dépassement excessif de durée de cycle).

Le tableau suivant décrit la variable de sortie :

Sortie	Type	Commentaire
<code>FC_CycleCheckTimeSet</code>	DINT	Voir le tableau ci-après.

Le tableau suivant décrit la valeur renvoyée :

Valeur	Description
0	Action accomplie avec succès

FC_LzsTaskGetCurrentInterval : définition de l'intervalle de la tâche dans le programme CEI

Description de la fonction

La fonction `FC_LzsTaskGetCurrentInterval` permet de lire la fréquence configurée pour une tâche externe. Selon le déclencheur, il peut s'agir d'une constante ou d'une variable. La fréquence en cours est exprimée en millisecondes. La fonction renvoie la valeur -1 pour les tâches de fréquence variable.

Représentation graphique



Représentation en langage IL et ST

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

Description des variables d'E/S

Le tableau suivant décrit la variable de sortie :

Sortie	Type	Commentaire
<code>FC_LzsTaskGetCurrentInterval</code>	DINT	Voir le tableau ci-après.

Le tableau suivant décrit les valeurs renvoyées :

Déclencheur	Valeur
INIRQ1	-1
INIRQ2	-1
INIRQ3	-1
INIRQ4	-1
INIRQ1_4	-1
RTP_READ	Donne le temps de cycle Sercos dans Sercos phase 4.
RTP_MENC	Donne le temps de cycle Sercos dans Sercos phase 4.
RTP_LENC	Donne le temps de cycle Sercos dans Sercos phase 4.
RTP_AXIS	Donne le temps de cycle Sercos dans Sercos phase 4.
MDT_WRITE_ACCESS	Donne la valeur RTP dans Sercos phase 4.
NOTE : La valeur -1 indique que la tâche a une fréquence variable.	

Exemple

Une tâche est projetée avec un cycle de 50 ms. Le système doit lire sa fréquence réelle.

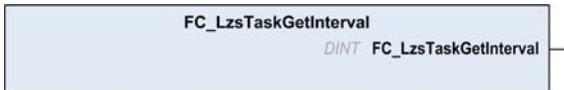
```
diIntervalMs:DINT:=0;
diCurrentIntervalUs:DINT:=0;
diIntervalMs := FC_LzsTaskGetInterval();
(* provides 50 * )
diCurrentIntervalUs := FC_LzsTaskGetCurrentInterval() ;
(* provides 50000 * )
```

FC_LzsTaskGetInterval : lecture de l'intervalle de la tâche dans le programme CEI

Description de la fonction

La fonction `FC_LzsTaskGetInterval` renvoie (en millisecondes) la périodicité configurée pour la tâche qui l'appelle.

Représentation graphique



Représentation en langage IL et ST

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

Description des variables d'E/S

Le tableau suivant décrit la variable de sortie :

Sortie	Type	Commentaire
<code>FC_LzsTaskGetInterval</code>	DINT	Voir le tableau ci-après.

Le tableau suivant décrit la valeur renvoyée :

Valeur	Description
<code>>=0</code>	Périodicité en ms.

FC_PrgResetAndStart : réinitialisation et démarrage de l'application en utilisant le programme utilisateur

Description de la fonction

La fonction FC_PrgResetAndStart démarre une tâche asynchrone qui effectue la réinitialisation (démarrage à chaud) et le démarrage de l'application. Toutes les tâches terminent leur cycle en cours ; même le code situé en aval de la fonction FC_PrgResetAndStart() est traité. La tâche asynchrone est exécutée avec la priorité de tâche CEI maximum.

Représentation graphique



Représentation en langage IL et ST

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

Description des variables d'E/S

Le tableau suivant décrit la variable de sortie :

Sortie	Type	Commentaire
FC_PrgResetAndStart	DINT	Voir le tableau ci-après.

Le tableau suivant décrit la valeur renvoyée :

Valeur	Description
0	Action accomplie avec succès

Sous-chapitre 2.3

LogicalAddress

Présentation

Cette section décrit les fonctions disponibles sous la rubrique LogicalAddress.

Contenu de ce sous-chapitre

Ce sous-chapitre contient les sujets suivants :

Sujet	Page
FC_CompareStLogicalAddress : comparaison de deux adresses logiques, i_stLogAddr1 et i_stLogAddr2.	31
FC_IsStLogicalAddressValid : vérification de la validation de l'adresse logique	32

FC_CompareStLogicalAddress : comparaison de deux adresses logiques, i_stLogAddr1 et i_stLogAddr2.

Description de la fonction

La fonction FC_CompareStLogicalAddress compare l'adresse logique i_stLogAddr1 à l'adresse logique i_stLogAddr2. Si les deux adresses sont identiques, elle renvoie TRUE, sinon elle renvoie FALSE.

Représentation graphique



Représentation en langage IL et ST

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

Description des variables d'E/S

Le tableau suivant décrit la variable d'entrée :

Entrée	Type	Commentaire
i_stLogAddr1	ST_LogicalAddress	Adresse logique 1
i_stLogAddr2	ST_LogicalAddress	Adresse logique 2

Le tableau suivant décrit la variable de sortie :

Sortie	Type	Commentaire
FC_CompareStLogicalAddress	BOOL	Voir le tableau ci-après.

Le tableau suivant décrit la valeur renvoyée :

Valeur	Description
TRUE	L'adresse 1 est identique à l'adresse 2.
FALSE	L'adresse 1 n'est pas identique à l'adresse 2.

FC_IsStLogicalAddressValid : vérification de la validation de l'adresse logique

Description de la fonction

La fonction FC_IsStLogicalAddressValid vérifie la validation de l'adresse logique dans i_stLogAddr.

TRUE = valide et FALSE = non valide.

Représentation graphique



Représentation en langage IL et ST

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

Description des variables d'E/S

Le tableau suivant décrit la variable d'entrée :

Entrée	Type	Commentaire
i_stLogAddr	ST_LogicalAddress	Adresse logique dont la validation doit être vérifiée

Le tableau suivant décrit la variable de sortie :

Sortie	Type	Commentaire
FC_IsStLogicalAddressValid	BOOL	Voir le tableau ci-après.

Le tableau suivant décrit les valeurs renvoyées :

Valeur	Description
TRUE	L'adresse est valide.
FALSE	L'adresse n'est pas valide.

Sous-chapitre 2.4

MessageLogger

FC_MsgLogSave : stockage du fichier journal des messages dans la mémoire de masse

Description de la fonction

La fonction `FC_MsgLogSave` stocke le contenu des fichiers journaux de messages dans la mémoire de masse (carte SD par exemple) sous la forme d'un fichier nommé `i_sFilename`. L'extension de nom de fichier `.mel` est attribuée par défaut. Un nom de fichier par défaut est composé de 1 à 8 caractères alphanumériques (A-Z, 0-9). Le système ajoute automatiquement le nom d'équipement `ide0:` et les extensions.

Un nom complexe se compose d'un nom d'équipement, d'un nom de fichier et d'une extension. Si `i_xReset` a la valeur `TRUE`, le contenu du fichier journal de messages est supprimé une fois stocké. Les messages générés pendant le traitement de la fonction `FC_MsgLogSave` ne sont pas perdus.

NOTE : Le temps de traitement de cette fonction est de l'ordre de quelques centaines de millisecondes. Lorsque vous utilisez cette fonction, vous devez augmenter le délai de chien de garde à l'aide de la fonction `FC_CycleCheckTimeSet`.

Représentation graphique



Représentation en langage IL et ST

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

Description des variables d'E/S

Le tableau suivant décrit la variable d'entrée :

Entrée/Sortie	Type	Commentaire
i_sFilename	STRING [80]	Nom du fichier stocké.
i_xReset	BOOL	Si la valeur est TRUE, le contenu du fichier journal de messages est supprimé une fois stocké (en tant que i_sFilename).

Le tableau suivant décrit la variable de sortie :

Sortie	Type	Commentaire
FC_MsgLogSave	DINT	Voir le tableau ci-après.

Le tableau suivant décrit les valeurs renvoyées :

Valeur	Description
0	Action accomplie avec succès
-1	Echec de l'écriture du fichier

Sous-chapitre 2.5

RemoteFile

Présentation

Cette section décrit les fonctions disponibles sous la rubrique RemoteFile.

Contenu de ce sous-chapitre

Ce sous-chapitre contient les sujets suivants :

Sujet	Page
FC_RemoteDeviceCreate : configuration d'un équipement pour les services de fichiers	36
FC_RemoteUserIdSet : configuration du nom utilisateur et du mot de passe des services de fichiers distants	38

FC_RemoteDeviceCreate : configuration d'un équipement pour les services de fichiers

Description fonctionnelle

La fonction `FC_RemoteDeviceCreate` crée un périphérique de fichiers distant nommé `i_sDevName`. Vous pouvez créer un périphérique réseau pour chaque ordinateur distant dont les données doivent être accessibles. L'ordinateur distant est indiqué par l'adresse IP `i_sHostIpAddr`.

Le nom d'ordinateur est suivi du signe deux-points et d'un nom de périphérique réseau. L'accès aux fichiers est assuré par des services client FTP. Configurez un serveur FTP pour accéder à un ordinateur distant. Lorsque vous ouvrez un fichier distant, il est entièrement copié dans un tampon local via le réseau. Si un fichier distant est créé, un tampon local vide est ouvert. Les opérations de lecture, d'écriture et d'accès iotcl sont effectuées sur la copie locale du fichier. Lorsque la copie locale d'un fichier ouvert pour modification est fermée, elle est renvoyée à l'ordinateur distant via le réseau.

NOTE : Lorsque vous accédez à l'équipement distant, les sous-répertoires ne contiennent pas forcément la chaîne racine.

NOTE : Le temps de traitement de cette fonction est de l'ordre de quelques centaines de millisecondes. Lorsque vous utilisez cette fonction, vous devez augmenter le délai de chien de garde à l'aide de la fonction `FC_CycleCheckTimeSet`.

Représentation graphique



Représentation en langage IL et ST

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

Description des variables d'E/S

Le tableau suivant décrit la variable d'entrée :

Entrée	Type	Commentaire
i_sDevName	STRING [80]	Nom de l'équipement de fichiers distant.
i_sHostIpAddr	STRING [80]	Adresse IP de l'ordinateur distant.

Le tableau suivant décrit la variable de sortie :

Sortie	Type	Commentaire
FC_RemoteDeviceCreate	DINT	Voir le tableau ci-après.

Le tableau suivant décrit les valeurs renvoyées :

Valeur	Description
0	Action accomplie avec succès
-1	Erreur détectée

Exemple

```
FC_RemoteUserIdSet('otto','secret');
FC_RemoteDeviceCreate('usr:', '190.201.100.99');
pHandle := SysFileOpen('usr:/myfile',AM_WRITE,pResult);
```

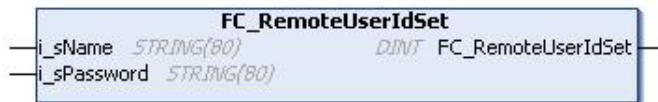
FC_RemoteUserIdSet : configuration du nom utilisateur et du mot de passe des services de fichiers distants

Description fonctionnelle

La fonction FC_RemoteUserIdSet définit le nom utilisateur et le mot de passe à utiliser pour accéder à l'ordinateur distant. Le nom (i_sName) et le mot de passe (i_sPassword) sont utilisés pour accéder à distance à des fichiers via les services FTP. Les valeurs par défaut du contrôleur sont USER (nom utilisateur) et USER (mot de passe).

NOTE : Vous trouverez des informations importantes sur la sécurité Internet dans le manuel *LMC078 Motion Controller - Guide de programmation (voir Modicon LMC078, Motion Controller, Programming Guide)*.

Représentation graphique



Représentation en langage IL et ST

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

Description des variables d'E/S

Le tableau suivant décrit la variable d'entrée :

Entrée	Type	Commentaire
i_sName	STRING [80]	Nom utilisateur
i_sPassword	STRING [80]	Mot de passe

Le tableau suivant décrit la variable de sortie :

Sortie	Type	Commentaire
FC_RemoteUserIdSet	DINT	Voir le tableau ci-après.

Le tableau suivant décrit les valeurs renvoyées :

Valeur	Description
0	Action accomplie avec succès
-1	Erreur détectée

Exemple**Déclaration**

```

PROGRAM RemoteFile
VAR
    bWriteTest: BOOL := FALSE;
    bReadTest: BOOL := FALSE;
    hReadFileId: RTS_IEC_HANDLE := 0;
    hWriteFileId: RTS_IEC_HANDLE := 0;
    lResultWriteToFile: UDINT := 99;
    lResultFC_ReadFromFile: UDINT := 99;
    lResultFC_CloseFile: UDINT := 99;
    lResultFC_RemoteDeviceCreate: DINT := 99;
    lResultFC_RemoteUserIdSet: DINT := 99;
    pszRemoteDeviceName: STRING := 'remote0: ';
    pszRemoteIpAddress: STRING := '10.128.156.14';
    pszRemoteFtpUserId: STRING := 'USER';
    pszRemoteFtpPassword: STRING := 'PASSWORD';
    pszFileName: STRING := 'remote0:/myfile.txt';
    pszOutText: STRING := 'Schneider Electric LMC078';
    pszInText: STRING;
    lTextSize: UDINT := 25
    bInitOk: BOOL := TRUE;
    pResult: RTS_IEC_RESULT := 0;
END_VAR

```

Programme

```

IF NOT bInitOk THEN
    FC_CycleCheckTimeSet(1000, 60);
    lResultFC_RemoteUserIdSet := FC_RemoteUserIdSet(pszRemoteFtpUserId, pszRemoteFtpPassword);
    lResultFC_RemoteDeviceCreate := FC_RemoteDeviceCreate(pszRemoteDeviceName, pszRemoteIpAddress);
    bInitOk := TRUE;
END_IF

```

```
IF bWriteTest THEN
  hWriteFileId := SysFile.SysFileOpen(pszFileName,AM_WRITE,pResult);
  IF hWriteFileId > 0 THEN
    lResultFC_WriteToFile := SysFile.SysFileWrite(hWriteFileId,
ADR(pszOutText), lTextSize, pResult);
    lResultFC_CloseFile := SysFile.SysFileClose(hWriteFileId);
  END_IF
  bWriteTest:= FALSE;
END_IF
IF bReadTest = TRUE THEN
  hReadFileId := SysFile.SysFileOpen(pszFileName,AM_READ,pResult);
  IF hReadFileId > 0 THEN
    lResultFC_ReadFromFile := SysFile.SysFileRead(hReadFileId,ADR(pszI
nText), lTextSize, pResult);
  lResultFC_CloseFile := SysFile.SysFileClose(hWriteFileId);;
  END_IF
  bReadTest := FALSE;
END_IF
```

Sous-chapitre 2.6

Conservation

Présentation

Cette section décrit les fonctions qui agissent sur la mémoire de conservation (retain) du contrôleur.

Contenu de ce sous-chapitre

Ce sous-chapitre contient les sujets suivants :

Sujet	Page
FC_CheckProgramIdent : vérification de l'adéquation de l'image mémoire avec l'application avant de la charger	42
FC_GetRetainImageInfo : lecture des informations complémentaires à écrire dans le fichier d'une image mémoire	44
FC_RetainImageLoad : chargement de l'image mémoire de la mémoire de conservation (se trouvant dans un fichier dans la mémoire de conservation du contrôleur)	46
FC_RetainImageSave : enregistrement du contenu de la mémoire de conservation dans un fichier	49

FC_CheckProgramIdent : vérification de l'adéquation de l'image mémoire avec l'application avant de la charger

Description fonctionnelle

La fonction `FC_CheckProgramIdent` lit le paramètre `ProgramIdent` de l'image mémoire dans le fichier indiqué et le compare au paramètre `ProgramIdent` de l'application actuellement en mémoire. Utilisez cette fonction avant la fonction `FC_RetainImageLoad`.

NOTE : Le temps d'exécution de cette fonction sur un contrôleur est généralement de 50 ms si le fichier de données conservées se trouve sur la carte SD.

Représentation graphique



Représentation en langage IL et ST

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

Description des variables d'E/S

Le tableau suivant décrit la variable d'entrée :

Entrée/Sortie	Type	Commentaire
<code>i_sRetainImageFileName</code>	STRING [80]	Nom du fichier dont il faut comparer le <code>ProgramIdent</code> avec celui du programme CEI actuellement situé dans la mémoire principale. Le nom de fichier reçoit automatiquement l'extension <code>.ret</code> . Le fichier peut se trouver dans n'importe quel répertoire existant du contrôleur. Le nom de fichier ne peut pas contenir de caractères génériques.

Le tableau suivant décrit la variable de sortie :

Sortie	Type	Commentaire
<code>FC_CheckProgramIdent</code>	DINT	Voir le tableau ci-après.

Le tableau suivant décrit les valeurs renvoyées :

Valeur	Description
0	Les ProgramIdent du programme CEI et de l'image mémoire correspondent.
-1	La comparaison n'a pas pu être effectuée car un nom de fichier non valide a été transmis. Le nom de fichier est trop long ou il contient l'un des caractères - ; \, /, *ou ?.
-2	Les informations complémentaires n'ont pas pu être lues car le répertoire indiqué n'existe pas.
-3	La comparaison n'a pas pu être effectuée car le nom de fichier indiqué n'existe pas.
-4	Les ProgramIdent du programme CEI et de l'image mémoire ne correspondent pas.
-5	La comparaison n'a pas pu être effectuée car le fichier indiqué n'est pas un fichier d'image d'une mémoire de conservation (retain).
-6	La comparaison n'a pas pu être effectuée car le fichier indiqué ne contient pas une image mémoire de la mémoire de conservation (retain).

Exemples

```
diResult := FC_CheckProgramIdent('carton');
```

Le ProgramIdent de l'application est comparé au ProgramIdent stocké dans le fichier 'carton.ret'. Le fichier 'carton.ret' se trouve sur la carte SD.

```
diResult := FC_CheckProgramIdent('ide0:\retain\plister');
```

Le ProgramIdent de l'application est comparé au ProgramIdent stocké dans le fichier 'plister.ret'. Le fichier 'plister.ret' se trouve dans le répertoire 'retain' sur la carte SD.

```
diResult := FC_CheckProgramIdent('remote:ide0:\label');
```

Le ProgramIdent de l'application est comparé au ProgramIdent stocké dans le fichier 'label.ret'. Le fichier 'label.ret' se trouve sur l'unité distante 'remote' sur la carte SD.

FC_GetRetainImageInfo : lecture des informations complémentaires à écrire dans le fichier d'une image mémoire

Description de la fonction

La fonction `FC_GetRetainImageInfo` lit les informations complémentaires sur une image mémoire à partir du fichier indiqué et les place dans la structure transmise. Ces informations sont les suivantes :

- Nom du fichier
- Date mémoire
- Type de contrôleur
- Taille de la mémoire de conservation (retain) du contrôleur
- Informations utilisateur
- Valeur du total de contrôle
- Taille de l'image mémoire

NOTE : Si le fichier de données conservées se trouve sur l'unité `ide0:`, le temps d'exécution de la fonction sur un contrôleur est généralement de 50 ms.

Représentation graphique



Représentation en langage IL et ST

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

Description des variables d'E/S

Le tableau suivant décrit la variable d'entrée :

Entrée	Type	Commentaire
i_sRetainImage- FileName	STRING [80]	Nom du fichier dont il faut comparer le ProgramIdent avec celui du programme CEI actuellement situé dans la mémoire principale. Le nom de fichier reçoit automatiquement l'extension .ret. Le fichier peut se trouver dans n'importe quel répertoire existant du contrôleur. Le nom de fichier ne doit pas contenir de caractère générique.

Le tableau suivant décrit la variable de sortie :

Sortie	Type	Commentaire
FC_GetRetainImage- Info	DINT	Voir le tableau ci-après.

Le tableau suivant décrit les valeurs renvoyées :

Valeur	Description
0	Les informations complémentaires ont été lues correctement dans le fichier indiqué.
-1	La comparaison n'a pas pu être effectuée car un nom de fichier non valide a été transmis. Le nom de fichier est trop long ou il contient l'un des caractères - ; \, /, * ou ?.
-2	Les informations complémentaires n'ont pas pu être lues car le répertoire indiqué n'existe pas.
-3	Les informations complémentaires n'ont pas pu être lues car le nom de fichier indiqué n'existe pas.
-4	Les informations complémentaires n'ont pas pu être lues car le fichier indiqué n'est pas un fichier d'image d'une mémoire de conservation (retain).
-5	Les informations complémentaires n'ont pas pu être lues car une erreur générale a été détectée dans la fonction FC_GetRetainImageInfo().

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

Entrée/Sortie	Type	Commentaire
iq_stInfo	ST_RetainImageInfo	Transfert des informations complémentaires

FC_RetainImageLoad : chargement de l'image mémoire de la mémoire de conservation (se trouvant dans un fichier dans la mémoire de conservation du contrôleur)

Description fonctionnelle

La fonction `FC_RetainImageLoad` lit l'image mémoire dans le fichier indiqué et la place dans la mémoire de conservation (retain). Vous pouvez choisir de vérifier le `ProgramIdent` ou pas.

`ProgramIdent` est un identificateur unique d'application. Il est indispensable que la mémoire de conservation et la configuration des variables conservées de l'application correspondent. La vérification du `ProgramIdent` permet de confirmer cette correspondance. Cette opération compare les `ProgramIdent` de l'image mémoire et de l'application actuellement en mémoire et s'ils correspondent, l'image mémoire est chargée.

Une application reçoit un nouveau `ProgramIdent` lors de chaque modification. Vous pouvez donc refuser la vérification du `ProgramIdent` si seules des modifications mineures ont été effectuées dans le code et que la déclaration des variables conservées n'a pas changé.

La mémoire de conservation (retain) et l'application doivent correspondre pour que le contrôleur fonctionne correctement. Vérifiez la taille de l'image mémoire et la taille de mémoire de conservation requise par l'application. L'image mémoire n'est chargée que si les deux tailles coïncident.

NOTE : Le temps d'exécution de cette fonction sur un contrôleur est généralement de 100 ms si le fichier de données conservées se trouve sur la carte SD.

⚠ AVERTISSEMENT

FONCTIONNEMENT INATTENDU DE L'ÉQUIPEMENT

N'accédez pas aux données conservées pendant l'exécution de la fonction `FC_RetainImageLoad()`.

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

Représentation graphique



Représentation en langage IL et ST

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

Description des variables d'E/S

Le tableau suivant décrit la variable d'entrée :

Entrée/Sortie	Type	Commentaire
i_sRetainImageFileName	STRING [80]	Nom du fichier à partir duquel charger l'image mémoire de la mémoire de conservation. Le nom de fichier reçoit automatiquement l'extension .ret. Le fichier peut se trouver dans n'importe quel répertoire existant du contrôleur. Le nom de fichier ne doit pas contenir de caractère générique.
i_xProgramIdentCheck	BOOL	Vous pouvez choisir de vérifier le ProgramIdent ou pas. TRUE : L'image mémoire est chargée uniquement si son ProgramIdent correspond à celui de l'application en cours d'exécution. FALSE : Le ProgramIdent n'est pas vérifié. Si vous ne vérifiez pas le ProgramIdent, assurez-vous que l'application correspond à l'image mémoire de la mémoire de conservation (retain) à charger. Sinon, une erreur du contrôleur est détectée.

Le tableau suivant décrit la variable de sortie :

Sortie	Type	Commentaire
FC_RetainImageLoad	DINT	Voir le tableau ci-après.

Le tableau suivant décrit les valeurs renvoyées :

Valeur	Description
0	L'image mémoire a été chargée dans la mémoire de conservation.
-1	L'image mémoire n'a pas pu être chargée. Le nom de fichier indiqué n'est pas valide. Le nom de fichier est trop long ou il contient l'un des caractères - ; \ / , * ou ?.
-2	L'image mémoire n'a pas pu être chargée car le répertoire indiqué n'existe pas.
-3	L'image mémoire n'a pas pu être chargée car le nom de fichier indiqué n'existe pas.
-4	L'image mémoire n'a pas pu être chargée dans la mémoire de conservation car son ProgramIdent ne correspond pas à celui de l'application actuellement en mémoire.
-5	L'image mémoire n'a pas pu être chargée dans la mémoire de conservation car ses données ne sont pas valides. Il est probable que les données ont été modifiées manuellement. C'est en comparant la taille de l'image mémoire à celle des données conservées dans le fichier que cette erreur a été révélée.
-6	L'image mémoire n'a pas pu être chargée dans la mémoire de conservation car ses fichiers ne sont pas valides. C'est la vérification du total de contrôle qui a révélé cette erreur.

Valeur	Description
-7	L'image mémoire n'a pas pu être chargée dans la mémoire de conservation car le fichier indiqué n'est pas un fichier d'image d'une mémoire de conservation (retain).
-8	L'image mémoire n'a pas pu être chargée dans la mémoire de conservation car une erreur générale a été détectée dans la fonction <code>FC_RetainImageLoad()</code> .

Exemples

```
diResult := FC_RetainImageLoad('carton', TRUE) ;
```

La mémoire de conservation est chargée à partir du fichier 'carton.ret' sur la carte SD. Le ProgramIdent est vérifié.

```
diResult := FC_RetainImageLoad('ide0:\retain\plister', FALSE) ;
```

La mémoire de conservation est chargée à partir du fichier 'plister.ret' situé dans le répertoire 'retain' sur la carte SD. Le ProgramIdent n'est pas vérifié.

```
diResult := FC_RetainImageLoad('remote:ide0:\label', TRUE) ;
```

La mémoire de conservation est chargée à partir du fichier 'label.ret' qui se trouve sur la carte SD du contrôleur 'remote'. Le ProgramIdent est vérifié.

FC_RetainImageSave : enregistrement du contenu de la mémoire de conservation dans un fichier

Description fonctionnelle

La fonction `FC_RetainImageSave` place tout le contenu de la mémoire de conservation (retain) dans le fichier indiqué. Comme l'adressage et l'affectation des variables dans la mémoire de conservation ne sont pas prévisibles, la mémoire entière est stockée. Des informations personnelles peuvent également être stockées dans le fichier. S'il existe un fichier de même nom au même emplacement, il est remplacé.

NOTE : Ne modifiez pas les données de la mémoire de conservation pendant l'exécution de cette fonction. Elles risqueraient d'être incohérentes lors d'un chargement ultérieur. Assurez-vous que les données conservées ne font l'objet d'aucun accès pendant l'exécution de la fonction `FC_RetainImageSave()`.

NOTE : Le temps d'exécution typique de cette fonction sur un contrôleur est de 400 ms pour stocker la mémoire de conservation sur la carte SD.

NOTE : Le temps de traitement de cette fonction est de l'ordre de quelques centaines de millisecondes. Lorsque vous utilisez cette fonction, vous devez augmenter le délai de chien de garde à l'aide de la fonction `FC_CycleCheckTimeSet` Par exemple : `CycleCheckTimeSet(500, 1000)`.

Représentation graphique



Représentation en langage IL et ST

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

Description des variables d'E/S

Le tableau suivant décrit la variable d'entrée :

Entrée	Type	Commentaire
i_sRetainImage-FileName	STRING[80]	Nom du fichier dans lequel le contenu de la mémoire de conservation (retain) doit être stocké. Le nom de fichier reçoit automatiquement l'extension .ret. Le fichier peut se trouver dans n'importe quel répertoire existant du contrôleur. Le nom de fichier ne doit pas contenir de caractère générique.
i_sUserInfo	STRING[255]	Informations à stocker également dans le fichier d'image de la mémoire de conservation.

Le tableau suivant décrit la variable de sortie :

Sortie	Type	Commentaire
FC_RetainImageSave	DINT	Voir le tableau ci-après.

Le tableau suivant décrit les valeurs renvoyées :

Valeur	Description
0	La mémoire de conservation a été stockée dans le fichier indiqué.
-1	La mémoire de conservation n'a pas pu être stockée car le nom de fichier indiqué n'est pas valide. Le nom de fichier est trop long ou il contient l'un des caractères - ; \, /, * ou ?.
-2	La mémoire de conservation n'a pas pu être stockée car le répertoire de destination indiqué pour l'image mémoire n'existe pas.
-3	La mémoire de conservation n'a pas pu être stockée car une erreur générale a été détectée dans la fonction FC_RetainImageSave().

Exemples

```
diResult := FC_RetainImageSave('carton', 'Dump Cartoner');
```

La mémoire de conservation est enregistrée dans le fichier 'carton.ret' sur la carte SD.

```
diResult := FC_RetainImageSave('ide0:\retain\plister', 'Speicherabild Plister');
```

La mémoire de conservation est enregistrée dans le fichier 'plister.ret' situé dans le répertoire 'retain' sur la carte SD.

```
diResult := FC_RetainImageSave('remote:ide0:\label', 'Dump Labeler');
```

La mémoire de conservation est enregistrée dans le fichier 'label.ret' qui se trouve sur la carte SD du contrôleur 'remote'.

Sous-chapitre 2.7

System

Présentation

Cette section décrit les fonctions disponibles sous la rubrique System.

Contenu de ce sous-chapitre

Ce sous-chapitre contient les sujets suivants :

Sujet	Page
FC_GetNVRamStatus : vérification de la validité des données dans la NVRam lors du démarrage du contrôleur	52
FC_GetBootState : vérification de la validité de tous les paramètres après le redémarrage du contrôleur	53
FC_SysReset : réinitialisation du contrôleur	55
FC_SysSaveParameter : enregistrement des paramètres personnalisés sur la carte SD	56
FC_SysShutdown : vérification de la sécurité du système de fichiers avant la mise hors tension du contrôleur	57
FC_UserChangePassword : Changement du mot de passe de l'utilisateur	58

FC_GetNVRamStatus : vérification de la validité des données dans la NVRam lors du démarrage du contrôleur

Description fonctionnelle

La fonction `FC_GetNVRamStatus` lit le statut de la mémoire NVRam pendant le démarrage. Cette opération est importante lorsque l'alimentation du contrôleur est coupée pendant une période prolongée. Si la vérification de la mémoire NVRam au démarrage aboutit à un échec, le système consigne par défaut une entrée dans le journaliseur de messages (316 : détection d'une erreur CRC NvRam).

Représentation graphique



Représentation en langage IL et ST

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

Description des variables d'E/S

Le tableau suivant décrit la variable de sortie :

Sortie	Type	Commentaire
FC_GetNVRamStatus	BOOL	Voir le tableau ci-après.

Le tableau suivant décrit les valeurs renvoyées :

Valeur	Description
TRUE	La vérification de la NVRam au démarrage a réussi.
FALSE	Une erreur a été détectée lors de la vérification de la NVRam.

FC_GetBootState : vérification de la validité de tous les paramètres après le redémarrage du contrôleur

Description fonctionnelle

La fonction `FC_GetBootState` renvoie le statut de redémarrage du contrôleur. Cette action est nécessaire dans la mesure où le programme CEI est déjà chargé avant la fin du processus de redémarrage, voire lancé en fonction du paramètre `AutoRun`.

La fonction `FC_GetBootState()` permet de vérifier si le processus de redémarrage s'est exécuté complètement et avec succès. Tant que le processus de redémarrage n'est pas terminé, l'initialisation du système n'est pas considérée comme complète et tous les objets (codeurs logiques, par exemple) et paramètres ne sont pas validés.

Un bon usage de cette fonction consiste à s'assurer que le codeur physique est initialisé et fonctionne correctement. Par exemple, la durée du processus de redémarrage dépend de la taille du projet.

Représentation graphique



Représentation en langage IL et ST

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

Description des variables d'E/S

Le tableau suivant décrit la variable de sortie :

Sortie	Type	Commentaire
<code>FC_GetBootState</code>	DINT	Voir le tableau ci-après.

Le tableau suivant décrit les valeurs renvoyées :

Valeur	Description
0	Le processus de redémarrage n'est pas encore terminé.
1	Le processus de redémarrage est terminé.

Exemple

```
CASE lState OF
1:
    lBootReady:=FC_GetBootState();
    IF lBootReady=1 THEN
        lState:=lState+1;
    END_IF;
2:
    ....;
END_CASE;
```

FC_SysReset : réinitialisation du contrôleur

Description fonctionnelle

La fonction `FC_SysReset` redémarre le contrôleur en lançant une commande de réinitialisation. Pour plus d'informations, reportez-vous au document traitant des transitions entre états de commande (*voir Modicon LMC078, Motion Controller, Programming Guide*).

NOTE : Assurez-vous que le processus d'initialisation du contrôleur est terminé avant de demander un redémarrage avec la fonction `FC_SysReset ()`. Vous pouvez pour cela utiliser la fonction `FC_GetBootState ()`.

Représentation graphique



Représentation en langage IL et ST

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

Description des variables d'E/S

Le tableau suivant décrit la variable de sortie :

Sortie	Type	Commentaire
<code>FC_SysReset</code>	DINT	Voir le tableau ci-après.

Le tableau suivant décrit les valeurs renvoyées :

Valeur	Description
0	Action accomplie avec succès
-1	Contrôleur non réinitialisé

FC_SysSaveParameter : enregistrement des paramètres personnalisés sur la carte SD

Description fonctionnelle

La fonction `FC_SysSaveParameter` enregistre les paramètres configurables sur la carte SD. Ce fichier sert à initialiser le système au démarrage.

NOTE : Le temps de traitement de cette fonction est de l'ordre de quelques centaines de millisecondes. Lorsque vous utilisez cette fonction, vous devez augmenter le délai de chien de garde à l'aide de la fonction `FC_CycleCheckTimeSet` Par exemple : `CycleCheckTimeSet(500, 1000)`.

Représentation graphique



Représentation en langage IL et ST

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

Description des variables d'E/S

Le tableau suivant décrit la variable de sortie :

Sortie	Type	Commentaire
<code>FC_SysSaveParameter</code>	DINT	Voir le tableau ci-après.

Le tableau suivant décrit les valeurs renvoyées :

Valeur	Description
0	Action accomplie avec succès
-1	Erreur détectée

FC_SysShutdown : vérification de la sécurité du système de fichiers avant la mise hors tension du contrôleur

Description fonctionnelle

La fonction `FC_SysShutdown` permet de garantir que le contrôleur ferme le système de fichiers et tous les fichiers ouverts.

Elle commence par générer le message de diagnostic 8023 (arrêt du contrôleur) pour le journaliseur de messages et elle arrête toutes les tâches CEI. Ensuite, elle ferme le système de fichiers et les fichiers ouverts avant d'arrêter l'UC. Vous pouvez alors déconnecter le système de l'alimentation.

Pour redémarrer le système, il faut procéder à une réinitialisation matérielle (RESET) sur le panneau avant du contrôleur ou à une mise hors tension puis sous tension (OFF/ON).

Représentation graphique



Représentation en langage IL et ST

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

Description des variables d'E/S

Le tableau suivant décrit la variable de sortie :

Sortie	Type	Commentaire
<code>FC_SysShutdown</code>	DINT	Voir le tableau ci-après.

Le tableau suivant décrit les valeurs renvoyées :

Valeur	Description
0	Action accomplie avec succès
-1	Le contrôleur n'a pas été arrêté correctement.

FC_UserChangePassword : Changement du mot de passe de l'utilisateur

Description fonctionnelle

La fonction FC_UserChangePassword modifie le mot de passe d'un utilisateur existant. Le nom d'utilisateur peut être utilisé pour les services de fichiers distants. Vous devez vous souvenir du mot de passe existant.

NOTE : Vous trouverez des informations importantes sur la sécurité des mots de passe dans le manuel *LMC078 Motion Controller - Guide de programmation (voir Modicon LMC078, Motion Controller, Programming Guide)*.

Représentation graphique



Représentation en langage IL et ST

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

Description des variables d'E/S

Le tableau suivant décrit la variable d'entrée :

Entrée	Type	Commentaire
i_sUserName	STRING(80)	Nom utilisateur
i_sCurrentPassword	STRING(80)	Mot de passe actuel
i_sNewPassword	STRING(80)	Nouveau mot de passe

Le tableau suivant décrit la variable de sortie :

Sortie	Type	Commentaire
FC_UserChangePassword	DINT	Voir le tableau ci-après.

Le tableau suivant décrit les valeurs renvoyées :

Valeur	Description
0	Action accomplie avec succès
-1	Erreur interne
-2	Utilisateur inconnu
-3	Mot de passe non valide

Sous-chapitre 2.8

TimeAndSync

Présentation

Cette section décrit les fonctions disponibles sous la rubrique TimeAndSync.

Contenu de ce sous-chapitre

Ce sous-chapitre contient les sujets suivants :

Sujet	Page
FC_GetNsPerCPUClockCycle : mesure haute précision du temps	61
FC_GetTimeOfDay : lecture de l'heure du système en ms sans la date	62
FC_GetTSC : mesure haute précision du temps	64

FC_GetNsPerCPUClockCycle : mesure haute précision du temps

Description de la fonction

La fonction `FC_GetNsPerCPUClockCycle` renvoie la durée d'un cycle d'horloge d'UC en nanosecondes (ns). Vous pouvez utiliser cette valeur pour convertir en ns la valeur renvoyée par la fonction `FC_GetTSC`.

Représentation graphique



Représentation en langage IL et ST

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

Description des variables d'E/S

Le tableau suivant décrit la variable de sortie :

Sortie	Type	Commentaire
<code>FC_GetNsPerCPUClockCycle</code>	LREAL	Nombre de nanosecondes par cycle d'horloge de l'UC.

Exemple

Déclaration :

```
ulStartClock:UDINT;
ulEndClock:UDINT;
ulClock:UDINT;
lrTimeUs:LREAL;
```

Programme :

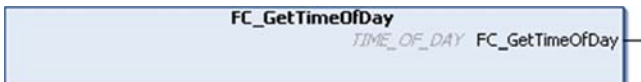
```
ulStartClock := FC_GetTSC();
:<Here put the code you want to measure the time execution>
ulEndClock := FC_GetTSC();
ulClock := ulEndClock - ulStartClock;
lrTimeUs:= UDINT_TO_LREAL(ulClock) * FC_GetNsPerCPUClockCycle() /1000.0
;
```

FC_GetTimeOfDay : lecture de l'heure du système en ms sans la date

Description de la fonction

La fonction `FC_GetTimeOfDay` lit l'heure actuelle du système à la milliseconde près, mais sans la date. Le programme CEI exige un horodatage comprenant la date, avec une valeur de retour au format `TimeOfDay`.

Représentation graphique



Représentation en langage IL et ST

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

Description des variables d'E/S

Le tableau suivant décrit la variable de sortie :

Sortie	Type	Commentaire
<code>FC_GetTimeOfDay</code>	<code>TIME_OF_DAY</code>	Heure au format <code>TimeOfDay</code> (de 00 h00m00s000ms à 23 h59min59s999ms)

Exemple

Pour intégrer une fonction de journalisation au programme CEI, le journal doit enregistrer les messages de diagnostic du système et les messages propres au client. Chaque message enregistré recevra un horodatage en millisecondes similaire à celui du journaliseur de messages.

L'horodatage doit présenter le format `YYYY-MM-DD-hh:mm:ss.000`. La fonction suivante enregistre la date et l'heure dans deux variables qui sont converties en chaînes et combinées.

VAR

```
dtActualDate : DATE;
todActualTimeOfDay: TOD;
sActualDate : STRING;
sActualTimeOfDay : STRING;
sTimestamp : STRING;
```

END_VAR

```
dtActualDate:= DT_TO_DATE(MyController.Realtimeclock) ;
```

```
todActualTimeOfDay:= FC_GetTimeOfDay() ;
sActualDate := DATE_TO_STRING(dtActualDate) ;
sActualDate := DELETE(sActualDate,2,1);
sActualTimeOfDay := TOD_TO_STRING(todActualTimeOfDay);
sActualTimeOfDay := DELETE(sActualTimeOfDay,4,1) ;
sTimestamp := CONCAT(sActualDate, '-') ;
sTimestamp := CONCAT(sTimestamp,sActualTimeOfDay) ;
```

Cette fonction montre comment obtenir l'heure actuelle en millisecondes et la date actuelle à partir du paramètre `RealTimeClock` et de la méthode `FC_GetTimeOfDay()`. Vous pouvez ensuite enregistrer ou afficher l'heure sous forme de chaîne avec un message.

FC_GetTSC : mesure haute précision du temps

Description de la fonction

La fonction `FC_GetTSC` lit la valeur sur 32 bits inférieure du temporisateur interne de l'UC. Le processeur Pentium a un temporisateur sur 64-bits qui est incrémenté à chaque cycle d'horloge.

Représentation graphique



Représentation en langage IL et ST

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

Description des variables d'E/S

Le tableau suivant décrit la variable de sortie :

Sortie	Type	Commentaire
FC_GetTSC	UDINT	Etat actuel du compteur de temporisation de l'UC (valeur 32 bits inférieure).

Exemple

Voir la fonction `FC_GetNsPerCPUClockCycle` (*voir page 61*).

Sous-chapitre 2.9

VolumeOperations

Présentation

Cette section décrit les fonctions disponibles sous la rubrique VolumeOperations.

Contenu de ce sous-chapitre

Ce sous-chapitre contient les sujets suivants :

Sujet	Page
FC_GetFreeDiskSpace : lecture de l'espace mémoire disponible d'un support mémoire	66
FC_GetTotalDiskSpace : lecture de la taille d'un support mémoire	68

FC_GetFreeDiskSpace : lecture de l'espace mémoire disponible d'un support mémoire

Description de la fonction

La fonction `FC_GetFreeDiskSpace` lit l'espace disponible sur un support mémoire (disque flash, disque RAM, clé USB) en octets. Le nom du support mémoire est transféré :

- Carte SD = ide0:
- Disque RAM = ram0:
- Support USB = usb2msd:0 (sans table de partition) ou usb2msd:1 (avec table de partition).

Il n'est pas possible d'obtenir l'espace libre sur un dispositif distant. 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*.

Description des variables d'E/S

Le tableau suivant décrit la variable d'entrée :

Entrée	Type	Commentaire
<code>i_sVolumeName</code>	STRING [80]	Nom de l'équipement où lire l'espace mémoire libre

Le tableau suivant décrit la variable de sortie :

Sortie	Type	Commentaire
<code>FC_GetFreeDiskSpace</code>	DINT	Voir le tableau ci-après.

Le tableau suivant décrit les valeurs renvoyées :

Valeur	Description
0	L'espace mémoire libre a été déterminé.
-1	Une erreur a été détectée lors de la lecture. Par exemple, l'équipement indiqué n'est pas valide ou il s'agit d'un équipement distant.
-318	Paramètre incorrect (<code>i_sVolumeName</code>).

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

Entrée/Sortie	Type	Commentaire
iq_uliFreeDiskSpace	ULINT	Espace mémoire libre en octets.

Exemple

Besoin :

Il faut déterminer l'espace mémoire libre sur un périphérique USB. Pour cela, le périphérique USB doit au préalable être détecté.

Déclaration

```
PROGRAM FreeUSBDiskSpace
```

```
VAR
```

```
    uliFreeDiskSpace : ULINT := 0;
```

```
    diStatus : DINT := 0;
```

```
    diBytesOfMegaByte : DINT := 1048576; // = 1024 * 1024
```

```
    diFreeDiskSpaceInMB : LINT := 0;
```

```
END_VAR
```

Programme

```
diStatus := FC_GetFreeDiskSpace ( 'usb2msd:1', uliFreeDiskSpace );
```

```
IF (diStatus = 0) THEN
```

```
    diFreeDiskSpaceInMB := ULINT_TO_LINT(uliFreeDiskSpace) / diBytesOf  
MegaByte;
```

```
END_IF
```

Résultat

diFreeDiskSpaceInMB contient la quantité d'espace mémoire libre sur le support USB en mégaoctets.

FC_GetTotalDiskSpace : lecture de la taille d'un support mémoire

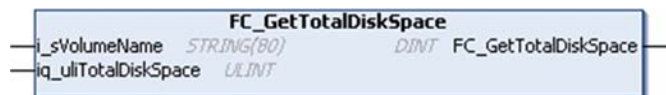
Description de la fonction

La fonction `FC_GetTotalDiskSpace` lit la taille d'un support mémoire (carte SD, disque RAM, clé USB) en octets. Le nom du support mémoire est transféré :

- Carte SD = `ide0`;
- Disque RAM = `ram0`;
- Support USB = `usb2msd:0` (sans table de partition) ou `usb2msd:1` (avec table de partition).

Il n'est pas possible d'obtenir la taille d'un support mémoire distant. 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*.

Description des variables d'E/S

Le tableau suivant décrit la variable d'entrée :

Entrée	Type	Commentaire
<code>i_sVolumeName</code>	<code>STRING [80]</code>	Nom du support mémoire dont la taille doit être déterminée.

Le tableau suivant décrit la variable de sortie :

Sortie	Type	Commentaire
<code>FC_GetTotalDiskSpace</code>	<code>DINT</code>	Voir le tableau ci-après.

Le tableau suivant décrit les valeurs renvoyées :

Valeur	Description
0	L'espace mémoire libre a été déterminé.
-1	Erreur détectée lors de la lecture de la taille.
-318	L'un des paramètres au moins n'est pas valide.

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

Entrée/Sortie	Type	Commentaire
iq_uliTotalDiskSpace	ULINT	Taille du support mémoire en octets.

Exemple

Besoin :

Déterminer la taille de la carte SD du contrôleur en Mo.

Déclaration

```
PROGRAM RamDiskSize
```

```
VAR
```

```
    uliTotalDiskSpace : ULINT := 0;
    diStatus : DINT := 0;
    diBytesOfMegaByte : DINT := 1048576; // = 1024 * 1024
    diSizeOfRamdiskInMB : LINT := 0;
```

```
VAR
```

Programme

```
diStatus := FC_GetTotalDiskSpace ( 'ide0:', uliTotalDiskSpace);
IF (diStatus = 0) THEN
    diSizeOfRamdiskInMB := ULINT_TO_LINT(uliTotalDiskSpace)/diBytesOfMegaByte;
END_IF
```

Résultat

diSizeOfRamdiskInMB contient la taille de la carte SD en Mo.



A

application

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

B

bus d'extension

Bus de communication électronique entre des modules d'E/S d'extension et un contrôleur.

C

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.

contrôleur

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

E

E/S

Entrée/sortie

P

programme

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



A

activation/désactivation des messages de diagnostic pour dépassement de durée de cycle

FC_CycleCheckSet, 22

C

changement de la valeur temporelle pour la surveillance du temps de cycle de la tâche CEI

FC_CycleCheckTimeSet, 25

changement du mot de passe de l'utilisateur

FC_UserChangePassword, 58

chargement de l'image mémoire de la mémoire de conservation

FC_RetainImageLoad, 46

comparaison de deux adresses logiques

FC_CompareStLogicalAddress, 31

configuration d'un équipement pour les services de fichiers

FC_RemoteDeviceCreate, 36

configuration du nom utilisateur et du mot de passe des services de fichiers distants

FC_RemoteUserIdSet, 38

D

déclenchement du message de diagnostic

FC_DiagUserMsgWrite, 19

définition de l'intervalle de la tâche dans le programme CEI

FC_LzsTaskGetCurrentInterval, 26

E

enregistrement des paramètres personnalisés sur la carte SD

FC_SysSaveParameter, 56

enregistrement du contenu de la mémoire de

conservation

FC_RetainImageSave, 49

F

FC_CheckProgramIdent

vérification de l'adéquation de l'image mémoire avec l'application, 42

FC_CompareStLogicalAddress

comparaison de deux adresses logiques, 31

FC_CycleCheckSet

activation/désactivation des messages de diagnostic pour dépassement de durée de cycle, 22

FC_CycleCheckTimeGet

identification des valeurs temporelles pour la surveillance du temps de cycle de la tâche CEI, 24

FC_CycleCheckTimeSet

changement de la valeur temporelle pour la surveillance du temps de cycle de la tâche CEI, 25

FC_DiagConfigRead2

lecture de la configuration du diagnostic, 13

FC_DiagConfigSet2

modification de la configuration du diagnostic, 15

FC_DiagMsgRead

lecture des informations de diagnostic, 17

FC_DiagQuit

réinitialisation du message de diagnostic, 20

FC_DiagUserMsgWrite

déclenchement du message de diagnostic, 19

FC_GetBootState

identification de l'état du redémarrage du contrôleur, 53

- FC_GetFreeDiskSpace
 - lecture de l'espace mémoire disponible, *66*
- FC_GetNsPerCPUClockCycle
 - mesure haute précision du temps, *61*
- FC_GetNVRamStatus
 - lecture de l'état de la NVRam lors du démarrage, *52*
- FC_GetRetainImageInfo
 - lecture des informations complémentaires à écrire dans le fichier d'une image mémoire, *44*
- FC_GetTimeOfDay
 - lecture de l'heure du système, *62*
- FC_GetTotalDiskSpace
 - lecture de la taille d'une mémoire, *68*
- FC_GetTSC
 - mesure haute précision du temps, *64*
- FC_IsStLogicalAddressValid
 - vérification de la validation de l'adresse logique, *32*
- FC_LzsTaskGetCurrentInterval
 - définition de l'intervalle de la tâche dans le programme CEI, *26*
- FC_LzsTaskGetInterval
 - lecture de l'intervalle de la tâche dans le programme CEI, *28*
- FC_MsgLogSave
 - stockage du fichier journal des messages dans la mémoire de masse, *33*
- FC_PrgResetAndStart
 - réinitialisation et démarrage de l'application en utilisant le programme utilisateur, *29*
- FC_RemoteDeviceCreate
 - configuration d'un équipement pour les services de fichiers, *36*
- FC_RemoteUserIdSet
 - configuration du nom utilisateur et du mot de passe des services de fichiers distants, *38*
- FC_RetainImageLoad
 - chargement de l'image mémoire de la mémoire de conservation, *46*
- FC_RetainImageSave
 - enregistrement du contenu de la mémoire de conservation, *49*
- FC_SysReset
 - réinitialisation du contrôleur, *55*
- FC_SysSaveParameter
 - enregistrement des paramètres personnalisés sur la carte SD, *56*
- FC_SysShutdown
 - vérification de la sécurité du système de fichiers avant la mise hors tension, *57*
- FC_UserChangePassword
 - changement du mot de passe de l'utilisateur, *58*

fonction

FC_CheckProgramIdent, 42
 FC_CompareStLogicalAddress, 31
 FC_CycleCheckSet, 22
 FC_CycleCheckTimeGet, 24
 FC_CycleCheckTimeSet, 25
 FC_DiagConfigRead2, 13
 FC_DiagConfigSet2, 15
 FC_DiagMsgRead, 17
 FC_DiagQuit, 20
 FC_DiagUserMsgWrite, 19
 FC_GetBootState, 53
 FC_GetFreeDiskSpace, 66
 FC_GetNsPerCPUClockCycle, 61
 FC_GetNVRamStatus, 52
 FC_GetRetainImageInfo, 44
 FC_GetTimeOfDay, 62
 FC_GetTotalDiskSpace, 68
 FC_GetTSC, 64
 FC_IsStLogicalAddressValid, 32
 FC_LzsTaskGetCurrentInterval, 26
 FC_LzsTaskGetInterval, 28
 FC_MsgLogSave, 33
 FC_PrgResetAndStart, 29
 FC_RemoteDeviceCreate, 36
 FC_RemoteUserIdSet, 38
 FC_RetainImageLoad, 46
 FC_RetainImageSave, 49
 FC_SysReset, 55
 FC_SysSaveParameter, 56
 FC_SysShutdown, 57
 FC_UserChangePassword, 58

I

identification de l'état du redémarrage du contrôleur

FC_GetBootState, 53

identification des valeurs temporelles pour la surveillance du temps de cycle de la tâche CEI

FC_CycleCheckTimeGet, 24

L

lecture de l'espace mémoire disponible

FC_GetFreeDiskSpace, 66

lecture de l'état de la NVRam lors du démarrage

FC_GetNVRamStatus, 52

lecture de l'heure du système

FC_GetTimeOfDay, 62

lecture de l'intervalle de la tâche dans le programme CEI

FC_LzsTaskGetInterval, 28

lecture de la configuration du diagnostic

FC_DiagConfigRead2, 13

lecture de la taille d'une mémoire

FC_GetTotalDiskSpace, 68

lecture des informations complémentaires à écrire dans le fichier d'une image mémoire

FC_GetRetainImageInfo, 44

lecture des informations de diagnostic

FC_DiagMsgRead, 17

M

mesure haute précision du temps

FC_GetNsPerCPUClockCycle, 61

FC_GetTSC, 64

modification de la configuration du diagnostic

FC_DiagConfigSet2, 15

R

réinitialisation du contrôleur

FC_SysReset, 55

réinitialisation du message de diagnostic

FC_DiagQuit, 20

réinitialisation et démarrage de l'application

en utilisant le programme utilisateur

FC_PrgResetAndStart, 29

S

ST_RetainImageInfo

type de données, 9

stockage du fichier journal des messages

dans la mémoire de masse
FC_MsgLogSave, 33

T

type de données
ST_RetainImageInfo, 9

V

vérification de l'adéquation de l'image mémoire avec l'application
FC_CheckProgramIdent, 42
vérification de la sécurité du système de fichiers avant la mise hors tension
FC_SysShutdown, 57
vérification de la validation de l'adresse logique
FC_IsStLogicalAddressValid, 32