

Concept 2.6

Bibliothèque de blocs IEC
Intercalaire : SYSTEM

12/2010

© 2010 Schneider Electric. Tous droits réservés.

Table des matières

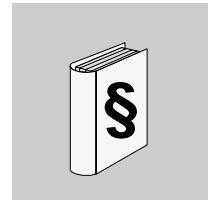


	Consignes de sécurité	7
	A propos de ce manuel	9
Partie I	Généralités sur la bibliothèque de blocs SYSTEM .	11
Chapitre 1	Paramétrage des fonctions et blocs fonction	13
	Paramétrage des fonctions et blocs fonction	13
Partie II	Présentations EFB	17
Chapitre 2	DIOSTAT : Etat de fonction de modules (DIO)	19
	Résumé	20
	Représentation	21
Chapitre 3	FREERUN : Compteur libre	23
	Résumé	24
	Représentation	25
Chapitre 4	GET_IEC_INF : Lecture des indicateurs de statut CEI .	27
	Résumé	28
	Représentation	29
Chapitre 5	GET_TOD : Lecture de l'horloge du matériel (Time Of Day)	31
	Résumé	32
	Représentation	33
Chapitre 6	HSBY_RD : Lecture du registre d'ordre Hot Standby . .	35
	Résumé	36
	Représentation	37
Chapitre 7	HSBY_ST : Lecture du registre d'état Hot Standby . . .	39
	Résumé	40
	Représentation	41
Chapitre 8	HSBY_WR : Ecriture du registre d'ordre Hot Standby .	43
	Résumé	44
	Représentation	45

Chapitre 9	ISECT_ON : déblocage d'une section d'interruption particulière	47
	Résumé	48
	Représentation	49
Chapitre 10	ISECT_OFF : blocage de sections d'interruption particulières	51
	Résumé	52
	Représentation	53
Chapitre 11	ISECT_STAT : état de section d'interruption	55
	Résumé	56
	Représentation	57
Chapitre 12	I_UNLOCK : déblocage de toutes les sections d'interruption	59
	Présentation	60
	Représentation	61
Chapitre 13	I_LOCK : blocage de toutes les sections d'interruption	63
	Présentation	64
	Représentation	66
Chapitre 14	LOOPBACK : Resaut	67
	Résumé	68
	Représentation	69
	Description détaillée	70
Chapitre 15	M1HEALTH : Etat de fonction de modules pour M1	71
	Résumé	72
	Représentation	73
Chapitre 16	I_MOVE : affectation protégée contre les interruptions	75
	Résumé	76
	Représentation	77
Chapitre 17	ONLEVT : Evénement connecté	79
	Résumé	80
	Représentation	81
Chapitre 18	PLCSTAT : Etat de fonction de l'API	83
	Résumé	84
	Représentation	85
	Etat de l'API (PLC_STAT) pour Quantum, Compact, Momentum et Atrium	89
	Etat RIO (RIO_STAT) pour le matériel Quantum / B800, Partie I	94
	Etat RIO (DIO_STAT) pour Quantum, Partie II	96
	Etat d'E/S (RIO_STAT) pour Compact	102
	Etat de bus E/S (RIO_STAT) pour Momentum	103
	Etat E/S global et état d'itération (DIO_STAT) pour Compact	105
Chapitre 19	PRJ_VERS : nom/version de projet	107
	Résumé	108
	Représentation	109

Chapitre 20	RES_IEC_INF : Remise à zéro des indicateurs de statut CEI	111
	Résumé	112
	Représentation	113
Chapitre 21	REV_XFER : Lecture et écriture des deux registres de transfert reverse	115
	Résumé	116
	Représentation	117
Chapitre 22	RIOSTAT : Etat de fonction de modules (RIO)	119
	Résumé	120
	Représentation	121
Chapitre 23	SAMPLETM : Période d'échantillonnage	123
	Brève description	124
	Représentation	125
Chapitre 24	SET_TOD : Réglage de l'horloge du matériel (Time Of Day)	127
	Résumé	128
	Représentation	129
Chapitre 25	SFCCNTRL : Commande SFC	131
	Résumé	132
	Représentation	133
	Description de la fonction	135
	Description des paramètres	137
Chapitre 26	SKP_RST_SCT_FALSE : Sauter la section restante	141
	Résumé	142
	Représentation	143
Chapitre 27	SYSCLOCK : Découpages du système	145
	Résumé	146
	Représentation	147
Chapitre 28	SYSSTATE : Etat de système	149
	Résumé	150
	Représentation	151
Chapitre 29	XSFCCNTRL : Commande SFC étendue	153
	Résumé	154
	Représentation	155
	Description de la fonction	157
	Description des paramètres	159
Glossaire		163
Index		193

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 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.



L'apposition de ce symbole à un panneau de sécurité Danger ou Avertissement signale un risque électrique pouvant entraîner des lésions corporelles en cas de non-respect des consignes.



Ceci est le symbole d'une 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 indique une situation immédiatement dangereuse qui, si elle n'est pas évitée, **entraînera** la mort ou des blessures graves.

AVERTISSEMENT

L'indication **AVERTISSEMENT** signale une situation potentiellement dangereuse et susceptible **d'entraîner la mort** ou des blessures graves.

ATTENTION

L'indication **ATTENTION** signale une situation potentiellement dangereuse et susceptible **d'entraîner des** blessures d'ampleur mineure à modérée.

ATTENTION

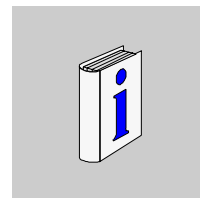
L'indication **ATTENTION**, utilisée sans le symbole d'alerte de sécurité, signale une situation potentiellement dangereuse et susceptible **d'entraîner des** dommages aux équipements.

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 cet appareil.

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

A propos de ce manuel



Présentation

Objectif du document

Cette documentation vous aidera à configurer les fonctions et les blocs fonction.

Champ d'application

Cette documentation s'applique à la version 2.6 de Concept pour Microsoft Windows 98, Microsoft Windows Version 2000, Microsoft Windows XP ou Microsoft Windows NT 4.x.

NOTE : Vous trouverez d'autres Notes de mise à jour dans le fichier README de Concept.

Document à consulter

Vous pouvez télécharger ces publications techniques ainsi que d'autres informations techniques à partir de notre site Web : www.telemecanique.com

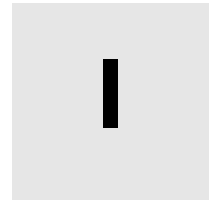
Titre de documentation	Référence
Instructions d'installation de Concept	840 USE 502 01
Manuel utilisateur de Concept	840 USE 503 01
Concept EFB User Manual	840 USE 505 00
Bibliothèque de blocs LL984 de Concept	840 USE 506 01

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

Commentaires utilisateur

Envoyez vos commentaires à l'adresse e-mail techpub@schneider-electric.com

Généralités sur la bibliothèque de blocs SYSTEM



Paramétrage des fonctions et blocs fonction

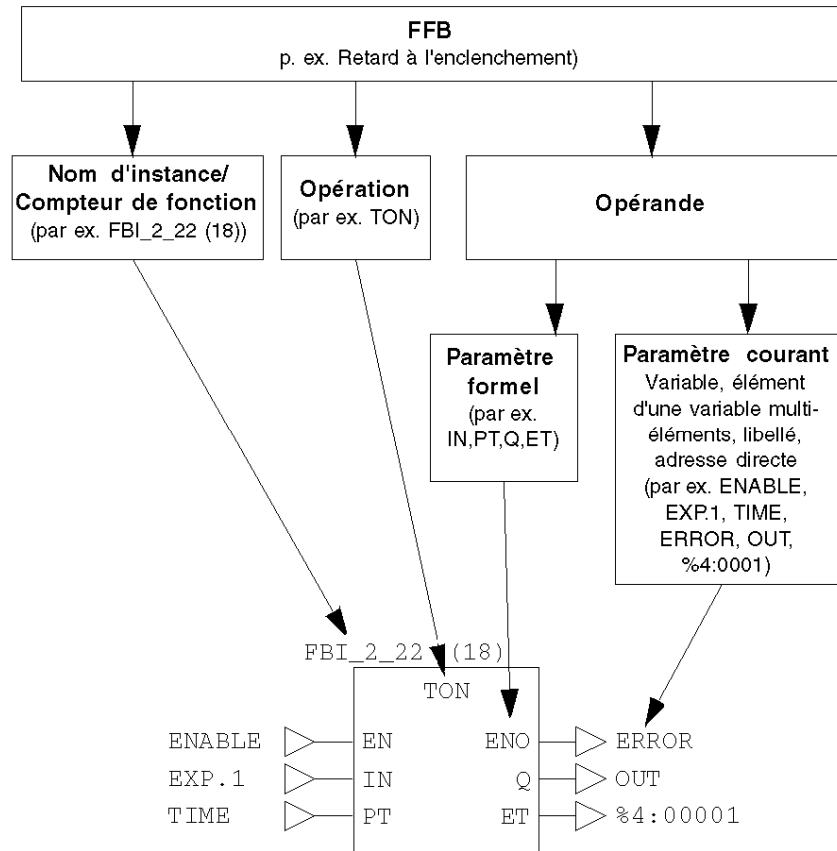


1

Paramétrage des fonctions et blocs fonction

Généralités

Tout FFB se compose d'une opération, des opérands nécessaires à l'opération et d'un nom d'instance/numéro de fonction.



Opération

L'opération détermine la fonctionnalité qui doit être exécutée par le FFB, p. ex. registre à décalage ou opérations de conversion.

Opérande

L'opérande détermine avec quoi l'opération doit être exécutée. Dans les FFB, il est constitué de paramètres formels et de paramètres réels.

Paramètre formel/paramètre réel

Le paramètre formel réserve la place pour un opérande. Lors du paramétrage, un paramètre actualisé (paramètre réel) est affecté au paramètre formel.

Le paramètre réel peut être une variable, une variable multi-éléments, un élément d'une variable multi-éléments, un libellé ou une adresse directe.

Lancement conditionnel/inconditionnel

Chaque FFB peut disposer d'un lancement "conditionnel" ou "non conditionnel". La condition est réalisée par une connexion préalable de l'entrée EN.

- EN démasqué
appel conditionnel (le FFB est traité uniquement lorsque EN = 1)
- EN masqué
appel non conditionnel (le FFB est toujours traité)

NOTE : Si elle n'est pas paramétrée, l'entrée EN doit être masquée. Étant donné que les entrées non paramétrées sont automatiquement occupées par un "0", le FFB ne serait jamais exécuté.

NOTE : Dans le cas des blocs fonction bloqués (EN = 0) disposant d'une fonction temporelle interne (par exemple, DELAY), il semble que le temps continue de s'écouler, car il est calculé à l'aide de l'horloge système, le rendant indépendant du cycle programme et de la validation du bloc.

Appel de fonctions et DE blocs fonction en IL et ST

Pour l'appel des fonctions et des blocs fonction dans IL (liste d'instructions) et ST (littéral structuré), veuillez vous référer aux chapitres correspondants du manuel de l'utilisateur.

Présentations EFB



Introduction

Ces présentations EFB sont documentées en ordre alphabétique.

NOTE : Le nombre d'entrées de certains EFB peut être augmenté à max. 32 par la modification verticale de la taille du symbole FFB. Veuillez vous référer à la présentation des différents EFB pour savoir de quels EFB il s'agit.

Contenu de cette partie

Cette partie contient les chapitres suivants :

Chapitre	Titre du chapitre	Page
2	DIOSTAT : Etat de fonction de modules (DIO)	19
3	FREERUN : Compteur libre	23
4	GET_IEC_INF : Lecture des indicateurs de statut CEI	27
5	GET_TOD : Lecture de l'horloge du matériel (Time Of Day)	31
6	HSBY_RD : Lecture du registre d'ordre Hot Standby	35
7	HSBY_ST : Lecture du registre d'état Hot Standby	39
8	HSBY_WR : Ecriture du registre d'ordre Hot Standby	43
9	ISECT_ON : déblocage d'une section d'interruption particulière	47
10	ISECT_OFF : blocage de sections d'interruption particulières	51
11	ISECT_STAT : état de section d'interruption	55
12	I_UNLOCK : déblocage de toutes les sections d'interruption	59
13	I_LOCK : blocage de toutes les sections d'interruption	63
14	LOOPBACK : Resaut	67
15	M1HEALTH : Etat de fonction de modules pour M1	71
16	I_MOVE : affectation protégée contre les interruptions	75
17	ONLEVT : Evénement connecté	79
18	PLCSTAT : Etat de fonction de l'API	83
19	PRJ_VERS : nom/version de projet	107

Chapitre	Titre du chapitre	Page
20	RES_IEC_INF : Remise à zéro des indicateurs de statut CEI	111
21	REV_XFER : Lecture et écriture des deux registres de transfert reverse	115
22	RIOSTAT : Etat de fonction de modules (RIO)	119
23	SAMPLETM : Période d'échantillonnage	123
24	SET_TOD : Réglage de l'horloge du matériel (Time Of Day)	127
25	SFCCNTRL : Commande SFC	131
26	SKP_RST_SCT_FALSE : Sauter la section restante	141
27	SYSCLOCK : Découpages du système	145
28	SYSSTATE : Etat de système	149
29	XSFCCNTRL : Commande SFC étendue	153

DIOSTAT : Etat de fonction de modules (DIO)

2

Introduction

Ce chapitre décrit le bloc DIOSTAT.

Contenu de ce chapitre

Ce chapitre contient les sujets suivants :

Sujet	Page
Résumé	20
Représentation	21

Résumé

Description de la fonction

Cette fonction indique l'état de fonctionnement des sous-groupes E/S d'une station d'E/S (DIO).

Chacun des modules (emplacement) d'une station d'E/S est représenté par un bit de la sortie "Etat". Le bit de la sortie "Etat" situé le plus à gauche correspond à l'emplacement gauche de la station d' E/S.

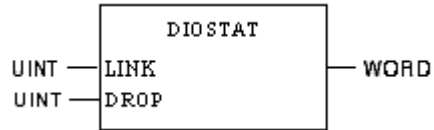
NOTE : Lorsqu'un module d'une station d'E/S est configuré et qu'il fonctionne correctement, le bit correspondant est mis à "1".

Il est possible de configurer EN et ENO comme paramètres supplémentaires.

Représentation

Symbole

Représentation du bloc :



Description des paramètres

Description des paramètres du bloc :

Paramètres	Type de données	Signification
LINK	UINT	No. de liaison (0...2)
STATION	UINT	No. de station d'E/S (1...64)
OUT	WORD	Configuration de bits d'état d'une station d'E/S

FREERUN : Compteur libre

3

Introduction

Ce chapitre décrit le bloc FREERUN.

Contenu de ce chapitre

Ce chapitre contient les sujets suivants :

Sujet	Page
Résumé	24
Représentation	25

Résumé

Description de la fonction

Cette fonction met en oeuvre un compteur libre qui peut être utilisé pour mesurer la durée des sections et des programmes utilisateurs.

Il est possible de projeter EN et ENO comme paramètres supplémentaires.

Déterminer la durée de fonctionnement d'une section

Déterminer la durée de fonctionnement d'une section :

Etape	Action
1	Placez une fonction FREERUN respectivement au début et à la fin de la section.
2	Assurez-vous, par l'ordre d'exécution, que la fonction FREERUN au début de la section soit exécutée comme première fonction de la section et celle à la fin de la section comme dernière fonction.
3	Calculez la différence des deux valeurs constatées. Ce Delta représente le temps enveloppe de la section en microsecondes.

Déterminer la durée de fonctionnement d'un programme

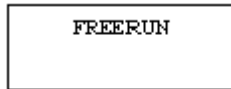
Déterminer la durée de fonctionnement d'un programme :

Etape	Action
1	Placez une fonction FREERUN respectivement au début de la première section du programme et à la fin de la dernière section.
2	Assurez-vous, par l'ordre d'exécution, que la fonction FREERUN soit exécutée au début de la première section comme première et respectivement à la fin de la dernière section comme dernière fonction de cette section.
3	Calculez la différence des deux valeurs déterminées. Ce Delta représente le temps enveloppe du programme en microsecondes.

Représentation

Symbole

Représentation du bloc :



Description des paramètres

Description des paramètres du bloc :

Paramètre	Type de données	Signification
OUT	DINT	Affiche le temps mesuré depuis le lancement du programme en microsecondes.

GET_IEC_INF : Lecture des indicateurs de statut CEI

4

Introduction

Le présent chapitre décrit le bloc GET_IEC_INF.

Contenu de ce chapitre

Ce chapitre contient les sujets suivants :

Sujet	Page
Résumé	28
Représentation	29

Résumé

Description de la fonction

Ce bloc fonction affiche les nouveaux indicateurs des erreurs système CEI sur les automates au niveau des sorties. De plus, la durée moyenne et maximale d'interruption du fonctionnement est indiquée par rapport à la durée de fonctionnement totale (temps de cycle) de l'application. La moyenne des 8 derniers cycles est également déterminée et affichée.

Toutes les données de gestion des interruptions CEI ne sont valables que pour les UC Quantum suivantes :

- 140 CPU 434 12
- 140 CPU 534 14

En cas d'utilisation d'un simulateur, aucune valeur utile n'est affichée pour la durée d'exécution maximale et proportionnelle.

Il est possible de configurer EN et ENO comme paramètres supplémentaires.

Erreur d'exécution

En cas d'erreur d'exécution -2801, la fonction EFB n'est pas disponible si la micrologique ne prend pas cette fonction en charge.

Représentation

Symbole

Représentation du bloc :

GET_IEC_INF	
LOOP_ON	— BOOL
DATA_INX	— BOOL
DIV_ZERO	— BOOL
IR_OVERF	— BOOL
IR_WDT	— BOOL
IR_ULOCK	— BOOL
IR_ALOAD	— BOOL
RFLAG8	— BOOL
RFLAG9	— BOOL
RFLAG10	— BOOL
RFLAG11	— BOOL
RFLAG12	— BOOL
RFLAG13	— BOOL
RFLAG14	— BOOL
RFLAG15	— BOOL
RFLAG16	— BOOL
AVG_IRLD	— INT
MAX_IRLD	— INT
SCANTIME	— INT

Description des paramètres

Description des paramètres du bloc :

Paramètre	Type de données	Signification
LOOP_ON	BOOL	Pour "1" : la boucle de contrôle (<i>voir Concept, Aide de Context,)</i> se ferme, la logique est partiellement exécutée.
DATA_INX	BOOL	Pour "1" : dépassement de plage (<i>voir Concept 2.6, Manuel de l'utilisateur,)</i> , accès incorrect aux données structurées.
DIV_ZERO	BOOL	Pour "1" : division par zéro (<i>voir Concept, Aide de Context,)</i> définie dans le code en ligne (Option "Code le plus rapide").
IR_OVERF	BOOL	Pour "1" : débordement dans une ou plusieurs sections d'interruption.
IR_WDT	BOOL	Pour "1" : le tempo chien de garde supérieur à 20 ms est écoulé sur une ou plusieurs sections d'interruption.

Paramètre	Type de données	Signification
IR_ULOCK	BOOL	Pour "1" : un ou plusieurs blocages sont encore actifs à la fin du temps de cycle. (L'activation n'a pas eu lieu.) Rappel : ne fonctionne pas en cas d'utilisation d'un simulateur.
IR_ALOAD	BOOL	Pour "1" : le temps de cycle moyen maximal pour les sections d'interruption dépassait la limite de 50 % du temps de cycle total. La sortie MAX_IRLD > 50. Rappel : ne fonctionne pas en cas d'utilisation d'un simulateur.
RFLAG8	BOOL	Indicateur réservé pour une utilisation ultérieure.
RFLAG9	BOOL	Indicateur réservé pour une utilisation ultérieure.
RFLAG10	BOOL	Indicateur réservé pour une utilisation ultérieure.
RFLAG11	BOOL	Indicateur réservé pour une utilisation ultérieure.
RFLAG12	BOOL	Indicateur réservé pour une utilisation ultérieure.
RFLAG13	BOOL	Indicateur réservé pour une utilisation ultérieure.
RFLAG14	BOOL	Indicateur réservé pour une utilisation ultérieure.
RFLAG15	BOOL	Indicateur réservé pour une utilisation ultérieure.
RFLAG16	BOOL	Indicateur réservé pour une utilisation ultérieure.
AVG_IRLD	INT	Valeur en pourcentage du temps de cycle moyen des sections d'interruption, mesurée sur le temps de cycle total [0...100]. La réinitialisation est obtenue par le chargement complet ou par le lancement de l'automate. Rappel : ne fonctionne pas en cas d'utilisation d'un simulateur.
MAX_IRLD	INT	Valeur en pourcentage du temps de cycle moyen maximum des sections d'interruption, mesurée sur le temps de cycle total [0...100]. En cas de dépassement de 50 %, l'indicateur IR_ALOAD est activé. La réinitialisation est obtenue par le chargement complet ou par le lancement de l'automate. Rappel : ne fonctionne pas en cas d'utilisation d'un simulateur.
SCAN_TIME	INT	Indication du temps de cycle actuel en millisecondes, calculé comme pour l'automate (moyenne des 8 derniers cycles).

GET_TOD : Lecture de l'horloge du matériel (Time Of Day)

5

Introduction

Ce chapitre décrit le bloc GET_TOD.

Contenu de ce chapitre

Ce chapitre contient les sujets suivants :

Sujet	Page
Résumé	32
Représentation	33

Résumé

Description de la fonction

Ce bloc fonction parcourt (avec les autres blocs fonction du groupe HSBY) la configuration de l'API respectif selon les composants qui lui sont nécessaires. Ces composants se réfèrent toujours au matériel effectivement connecté.

C'est pourquoi un comportement correct de ce bloc fonction ne peut être assuré sur les simulateurs.

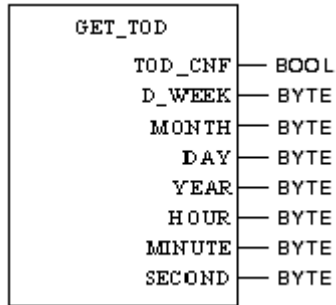
Le bloc fonction GET_TOD sert à lire l'horloge du matériel, si les registres correspondants ont été mis à disposition à l'intérieur de la configuration. Si ces registres ne sont pas disponibles, la sortie TOD_CNF est mise à "0".

Il est possible de projeter EN et ENO comme paramètres supplémentaires.

Représentation

Symbole

Représentation du bloc :



Description des paramètres

Description des paramètres du bloc :

Paramètres	Type de données	Signification
TOD_CNF	BOOL	"1" = Le registre 4x pour l'horloge du matériel a été trouvé et l'horloge est prête à fonctionner. "0" = L'heure est momentanément réglée. Dans ce cas, les autres sorties conservent leurs valeurs actuelles.
D_WEEK	OCTET	Jour de la semaine, 1 = Dimanche .. 7 = Samedi
MONTH	OCTET	Mois 1 à 12
DAY	OCTET	Jour 1 à 31
YEAR	OCTET	Année 0 à 99
HOUR	OCTET	Heure 0 à 23
MINUTE	OCTET	Minute 0 à 59
SECONDE	OCTET	Seconde 0 à 59

HSBY_RD : Lecture du registre d'ordre Hot Standby

6

Introduction

Ce chapitre décrit le bloc HSBY_RD.

Contenu de ce chapitre

Ce chapitre contient les sujets suivants :

Sujet	Page
Résumé	36
Représentation	37

Résumé

Description de la fonction

Ce bloc fonction sert à exploiter la fonctionnalité CEI Hot Standby. Il parcourt (en association avec d'autres blocs fonction du groupe HSBY) la configuration de chaque API à la recherche des composants qui lui sont nécessaires. Ces composants se réfèrent toujours au matériel effectivement connecté.

C'est pourquoi un comportement correct de ce bloc fonction ne peut être assuré sur les simulateurs.

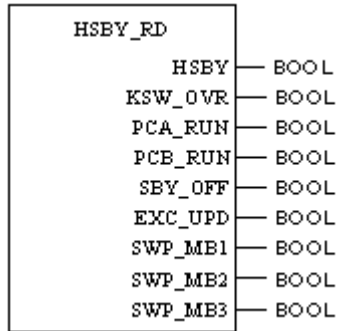
Le bloc fonction HSBY_RD contrôle de lui-même s'il existe une configuration Hot Standby. Si c'est le cas, il délivre le contenu du registre de commande et la sortie HSBY passe à "1". Si la configuration actuelle n'est pas à redondance d'UC, la sortie HSBY est mise à "0".

Il est possible de projeter EN et ENO comme paramètres supplémentaires.

Représentation

Symbole

Représentation du bloc :



Description des paramètres

Description des paramètres du bloc :

Paramètre	Type de données	Signification
HSBY	BOOL	"1" = Configuration de redondance d'UC trouvée
KSW_OVR	BOOL	<p>Keyswitch override</p> <p>"1" =</p> <ul style="list-style-type: none"> ● La position <i>Off Line</i> du commutateur rotatif du module redondant (CHSxxx) est désactivée par le logiciel. ● La position <i>Xfer</i> du commutateur rotatif du module redondant (CHSxxx) est activée par le logiciel. ● La position <i>Run</i> du commutateur rotatif du module redondant (CHSxxx) est désactivée par le logiciel. ● Le système redondant se trouve en mode de marche (DEL Run allumée). <p>"0" = Toutes les positions du commutateur rotatif du module redondant (CHSxxx) sont activées par le logiciel.</p>

Paramètre	Type de données	Signification
PCA_RUN	BOOL	<p>PLC A running</p> <p>"1" = L'API qui possède dans son rack local le module redondant avec le commutateur en position A se trouve dans le mode de marche (DEL Run de l'API et DEL Standby/Primary du module redondant allumées).</p> <p>N'a de signification que si le commutateur manuel est désactivé par logiciel.</p>
PCB_RUN	BOOL	<p>PLC B running</p> <p>"1" = L'API qui possède dans son rack local le module redondant avec le commutateur en position B se trouve dans le mode de marche (DEL Run de l'API et DEL Standby/Primary du module redondant allumées).</p> <p>N'a de signification que si le commutateur manuel est désactivé par logiciel.</p>
SBY_OFF	BOOL	<p>Standby Off on logic mismatch</p> <p>"1" = Différence de logique autorisée (bien que les deux automates contiennent des programmes différents, l'automate redondant reste en mode en ligne)</p> <p>"0" = Différence de logique non autorisée (l'automate redondant passe en mode hors ligne dès que les deux automates contiennent des programmes différents)</p>
EXC_UPD	BOOL	<p>Exec Update</p> <p>"1" = Mise à jour Exec autorisée dans l'API redondant pendant que l'API primaire continue de fonctionner.</p> <p>(Après mise à jour de l'exec, l'API redondant passe à nouveau en mode en ligne.)</p>
SWP_MB1	BOOL	<p>Swap address Modbus Port 1</p> <p>"1" = Changement d'adresse du port Modbus 1 activé</p>
SWP_MB2	BOOL	<p>Swap address Modbus Port 2</p> <p>"1" = Changement d'adresse du port Modbus 2 activé</p>
SWP_MB3	BOOL	<p>Swap address Modbus Port 3</p> <p>"1" = Changement d'adresse du port Modbus 3 activé</p>

HSBY_ST : Lecture du registre d'état Hot Standby

7

Introduction

Ce chapitre décrit le bloc HSBY_ST.

Contenu de ce chapitre

Ce chapitre contient les sujets suivants :

Sujet	Page
Résumé	40
Représentation	41

Résumé

Description de la fonction

Ce bloc fonction sert à exploiter la fonctionnalité de CEI Hot Standby. Il parcourt (en association avec d'autres blocs fonction du groupe HSBY) la configuration de chaque API à la recherche des composants qui lui sont nécessaires. Ces composants se réfèrent toujours au matériel effectivement connecté.

C'est pourquoi un comportement correct de ce bloc fonction ne peut être assuré sur les simulateurs.

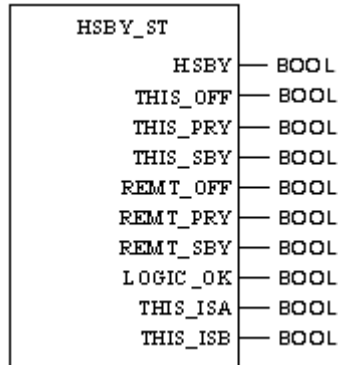
Le bloc fonction sert à lire le registre d'état Hot Standby. Si la configuration actuelle n'est pas à redondance d'UC ou si la configuration présente à redondance d'UC ne possède pas de zone de 'Non-Transfert', contenant le registre d'état, la sortie HSBY est alors mise à "0".

Il est possible de projeter EN et ENO comme paramètres supplémentaires.

Représentation

Symbole

Représentation du composant :



Description des paramètres

Description des paramètres du bloc :

Paramètre	Type de données	Signification
HSBY	BOOL	"1" = Configuration Hot Standby trouvée; zone 'Non-Transfer' inscrite dans celle-ci.
THIS_OFF	BOOL	"1" = Cet API est local
THIS_PRY	BOOL	"1" = Cet API est l'API primaire
THIS_SBY	BOOL	"1" = Cet API est l'API Standby
REMT_OFF	BOOL	"1" = L'autre API (remote) est local
REMT_PRY	BOOL	"1" = L'autre API est l'API primaire
REMT_SBY	BOOL	"1" = L'autre API est l'API Standby
LOGIC_OK	BOOL	"1" = Les programmes des deux API sont identiques
THIS_ISA	BOOL	"1" = Cet API possède le sous-groupe Hot Standby avec le sélecteur A dans le rack local.
THIS_ISB	BOOL	"1" = Cet API possède le sous-groupe Hot Standby avec le sélecteur B dans le rack local.

HSBY_WR : Ecriture du registre d'ordre Hot Standby



Introduction

Ce chapitre décrit le bloc HSBY_WR.

Contenu de ce chapitre

Ce chapitre contient les sujets suivants :

Sujet	Page
Résumé	44
Représentation	45

Résumé

Description de la fonction

Ce bloc fonction sert à exploiter la fonctionnalité de CEI Hot Standby. Il parcourt (en association avec d'autres blocs fonction du groupe HSBY) la configuration de chaque API à la recherche des composants qui lui sont nécessaire. Ces composants se réfèrent toujours au matériel effectivement connecté.

C'est pourquoi un comportement correct de ce bloc fonction ne peut être assuré sur les simulateurs.

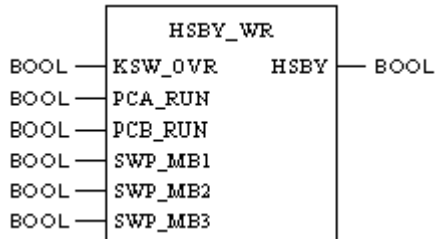
Le bloc fonction HSBY_WR sert à positionner différents modes Hot Standby qui sont autorisés pour le CEI Hot Standby. L'établissement de chaque mode signifie une modification du registre de commande de la redondance, ce qui est exécuté automatiquement par le bloc fonction. Si la configuration actuelle n'est pas à redondance d'UC, la sortie HSBY est mise à "0", sinon elle est mise à 1.

Il est possible de projeter EN et ENO comme paramètres supplémentaires.

Représentation

Symbole

Représentation du bloc :



Description des paramètres

Description des paramètres du bloc :

Paramètres	Type de données	Signification
KSW_OVR	BOOL	Keyswitch override "1" = Le sélecteur sur le module de redondance d'UC (CHSxxx) va être désactivé.
PCA_RUN	BOOL	PLC A Running "1 -> 0" = Et Keyswitch override réglé (KSW_OVR) L'API qui possède le sous-groupe de redondance d'UC avec le sélecteur A dans son rack local est contrainte à passer en mode local.
PCB_RUN	BOOL	PLC B Running "1 -> 0" = Et Keyswitch override réglé (KSW_OVR) L'API qui possède le sous-groupe de redondance d'UC avec le sélecteur B dans son rack local est contrainte à passer en mode local.
SWP_MB1	BOOL	Swap address Modbus Port 1 "0 -> 1" = L'adresse Modbus sur Port 1 du NOUVEL API primaire change si un Switchover s'est produit. (Nouvelle adresse de l'API primaire = ancienne adresse + 128 Nouvelle adresse de l'API Standby = ancienne adresse -128)

Paramètres	Type de données	Signification
SWP_MB2	BOOL	Swap address Modbus Port 2 "0 -> 1" = L'adresse Modbus sur Port 2 du NOUVEL API primaire change si un Switchover s'est produit. (Nouvelle adresse de l'API primaire = ancienne adresse + 128 Nouvelle adresse de l'API Standby = ancienne adresse -128).
SWP_MB3	BOOL	Swap address Modbus Port 3 "0 -> 1" = L'adresse Modbus sur Port 3 du NOUVEL API primaire change si un Switchover s'est produit. (Nouvelle adresse de l'API primaire = ancienne adresse + 128 Nouvelle adresse de l'API Standby = ancienne adresse -128).
HSBY	BOOL	"1" = Configuration de redondance d'UC trouvée.

ISECT_ON : déblocage d'une section d'interruption particulière

9

Introduction

Le présent chapitre décrit le bloc ISECT_ON.

Contenu de ce chapitre

Ce chapitre contient les sujets suivants :

Sujet	Page
Résumé	48
Représentation	49

Résumé

Description fonctionnelle

Ce bloc fonction permet de débloquer une section d'événement de temporisation ou une section d'événement d'E/S particulière, bloquée au préalable à l'aide du bloc ISECT_OFF (*voir page 51*).

Une section débloquée est déclenchée dès que le signal matériel (section d'événement d'E/S) ou l'intervalle de temps approprié (section d'événement de temporisation) apparaît. Les compteurs d'événements et d'activations s'incrémentent. Une interruption éventuelle suspend le traitement de la section. Celui-ci est toutefois repris à la fin de l'interruption.

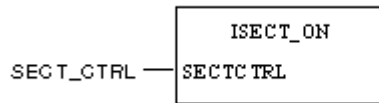
La broche d'entrée SECT_CTRL fournit la variable de commande d'une section particulière. Cette variable contient le nom de la section.

Vous pouvez projeter EN et ENO en tant que paramètres supplémentaires.

Représentation

Symbole

Représentation du bloc :



Description des paramètres

Description des paramètres du bloc :

Paramètres	Type de données	Signification
SECTCTRL	SECT_CTRL	Variable de commande, contient le nom de la section.

ISECT_OFF : blocage de sections d'interruption particulières

10

Introduction

Le présent chapitre décrit le bloc ISECT_OFF.

Contenu de ce chapitre

Ce chapitre contient les sujets suivants :

Sujet	Page
Résumé	52
Représentation	53

Résumé

Description fonctionnelle

Ce bloc fonction permet de bloquer une section d'événement de temporisation ou une section d'événement d'E/S particulière qu'il est ensuite possible de débloquent à l'aide du bloc ISECT_ON (*voir page 47*).

Le blocage inhibe uniquement le déclenchement de la section. Le compteur d'événements continue à compter les signaux matériel et les intervalles de temps arrivant de la section bloquée tandis que le compteur d'activations compte uniquement les sections réellement exécutées ou débloquentées. Une interruption éventuelle n'a aucun effet sur une section d'interruption bloquée.

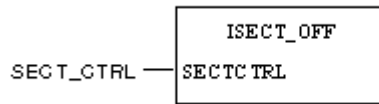
La broche d'entrée SECT_CTRL fournit la variable de commande d'une section particulière. Cette variable contient le nom de la section.

Vous pouvez projeter EN et ENO en tant que paramètres supplémentaires.

Représentation

Symbole

Représentation du bloc :



Description des paramètres

Description des paramètres du bloc :

Paramètres	Type de données	Signification
SECTCTRL	SECT_CTRL	Variable de commande, contient le nom de la section.

ISECT_STAT : état de section d'interruption

11

Introduction

Le présent chapitre décrit le bloc ISECT_STAT.

Contenu de ce chapitre

Ce chapitre contient les sujets suivants :

Sujet	Page
Résumé	56
Représentation	57

Résumé

Description fonctionnelle

Ce bloc fonction lit les états internes d'une section d'interruption et copie ces données dans les structures de données affectées aux sorties correspondantes.

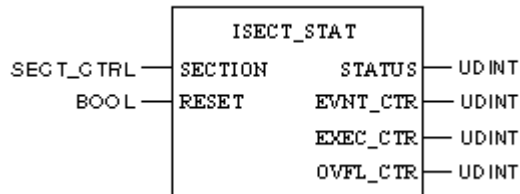
NOTE : Vous pouvez obtenir les informations du tableau d'état via la commande **En_ligne** → **Sections d'événement**.

La broche d'entrée REST permet de remettre toutes les sorties à 0.

Représentation

Symbole

Représentation du bloc :



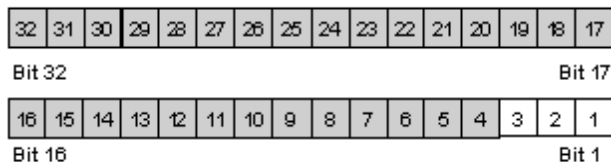
Description des paramètres

Description des paramètres du bloc :

Paramètres	Type de données	Signification
SECTION	SECT_CTRL	Variable de commande = nom de la section dont vous souhaitez voir l'état.
RESET	BOOL	Remet les sorties à 0.
STATUS	UDINT	Contient l'état de la section d'interruption (voir le sous-chapitre "Etat de section d'interruption")
EVNT_CTR	UDINT	Contient le nombre total d'événements.
EXEC_CTR	UDINT	Contient le nombre total d'événements exécutés.
OVFL_CTR	UDINT	Contient le nombre total d'événements qui n'ont pas pu être déclenchés car ils sont survenus lors de l'exécution d'une section.

Etat de section d'interruption

Affectation des bits :



Bit	Affectation
1	Un débordement est survenu, c.-à-d. qu'un événement n'a pas pu être déclenché.
2	La temporisation chien de garde est terminée.
3	Toutes les sections d'interruption bloquées ont été débloquées, mais le compteur de blocages n'est pas à zéro.
4-8	Réservée
9-16	Utilisation interne
17-32	Réservée

I_UNLOCK : déblocage de toutes les sections d'interruption

12

Introduction

Le présent chapitre décrit le bloc I_UNLOCK.

Contenu de ce chapitre

Ce chapitre contient les sujets suivants :

Sujet	Page
Présentation	60
Représentation	61

Présentation

Généralités

Les blocs I_LOCK et I_UNLOCK permettent de protéger une logique qui ne doit pas être interrompue par l'exécution d'une section d'événement, comme par exemple lors de l'accès à des variables utilisées en commun. Pour les opérations de copie non interruptibles, le système utilise le bloc *I_MOVE* : *affectation protégée contre les interruptions*, page 75.

Les blocs I_LOCK et I_UNLOCK peuvent aussi bien être utilisés dans des sections à déclenchement cyclique que dans des sections d'événement.

Description de la fonction

Ce bloc fonction permet de débloquer toutes les sections d'événement de temporisation ou les sections d'événement d'E/S, bloquées au préalable à l'aide du bloc I_LOCK (voir page 63).

Le blocage inhibe uniquement l'exécution immédiate des sections d'événement. Si le blocage est supprimé (par ex., via l'appel du bloc I_UNLOCK), le système traite alors tous les événements s'étant produits jusque-là, c'est-à-dire exécute les sections d'événement correspondantes. Toutefois, seul un événement (donc max. 16 événements de temporisation et 64 événements d'E/S) peut survenir lorsque le blocage est actif. Tout événement supplémentaire incrémente le compteur de débordement de la section d'événement concernée, mais ne déclenche pas d'exécution ultérieure de la section d'événement en question.

Par conséquent, s'il est certain que le blocage ne sera jamais activé plus longtemps que l'intervalle de temps minimal entre deux événements d'une section d'événement, le blocage n'entraîne alors aucune perte d'événement. Dans ce cas toutefois, la section d'événement serait exécutée avec un temps de retard. Pour éviter un blocage permanent de sections d'événement, la broche de sortie LOCKCTR est remise à zéro à la fin d'une section d'événement (appel de I_LOCK dans la section d'événement) ou après chaque cycle (appel de I_LOCK au sein d'une section à déclenchement cyclique), ce qui supprime automatiquement le blocage.

La broche de sortie LOCKCTR donne la valeur actuelle du compteur de blocages général. Cette valeur diminue à chaque appel du bloc I_UNLOCK. Une valeur de compteur égale à zéro signifie que les sections sont débloquées et que la logique associée est exécutée. Si la valeur de la broche de sortie LOCKCTR n'est pas égale à zéro à la fin d'une section d'événement (pas de I_UNLOCK dans la section), le bit 3 d'état de la section d'événement est mis à 1.

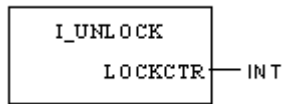
Le compteur d'activations compte uniquement les sections réellement exécutées ou débloquées.

Il est possible de configurer EN et ENO comme paramètres supplémentaires.

Représentation

Symbole

Représentation du bloc :



Description des paramètres

Description des paramètres du bloc :

Paramètres	Type de données	Signification
LOCKCTR	INT	Valeur actuelle du compteur de blocages général. Cette valeur augmente à chaque appel du bloc I_LOCK et diminue à chaque appel du bloc I_UNLOCK. Lorsque le compteur est à 0, toutes les sections sont débloquées.

I_LOCK : blocage de toutes les sections d'interruption

13

Introduction

Le présent chapitre décrit le bloc I_LOCK.

Contenu de ce chapitre

Ce chapitre contient les sujets suivants :

Sujet	Page
Présentation	64
Représentation	66

Présentation

Généralités

Les blocs I_LOCK et I_UNLOCK permettent de protéger une logique qui ne doit pas être interrompue par l'exécution d'une section d'événement, comme par exemple lors de l'accès à des variables utilisées en commun. Pour les opérations de copie non interruptibles, le système utilise le bloc *I_MOVE* : *affectation protégée contre les interruptions*, page 75.

Les blocs I_LOCK et I_UNLOCK peuvent aussi bien être utilisés dans des sections à déclenchement cyclique que dans des sections d'événement.

Description de la fonction

Ce bloc fonction permet de bloquer simultanément toutes les sections d'événement de temporisation ou toutes les sections d'événement d'E/S qu'il est ensuite possible de débloquent à l'aide du bloc I_UNLOCK (*voir page 59*).

Le blocage inhibe uniquement l'exécution immédiate des sections d'événement. Si le blocage est supprimé (par ex., via l'appel du bloc I_UNLOCK), le système traite alors tous les événements s'étant produits jusque-là, c'est-à-dire exécute les sections d'événement correspondantes. Toutefois, seul un événement par section d'événement (donc max. 16 événements de temporisation et 64 événements d'E/S) peut survenir lorsque le blocage est actif. Tout événement supplémentaire incrémente le compteur de débordement de la section d'événement concernée, mais ne déclenche pas d'exécution ultérieure de la section d'événement en question.

Par conséquent, s'il est certain que le blocage ne sera jamais activé plus longtemps que l'intervalle de temps minimal entre deux événements d'une section d'événement, le blocage n'entraîne alors aucune perte d'événement. Dans ce cas toutefois, la section d'événement serait exécutée avec un temps de retard. Pour éviter un blocage permanent de sections d'événement, la broche de sortie LOCKCTR est remise à zéro à la fin d'une section d'événement (appel de I_LOCK dans la section d'événement) ou après chaque cycle (appel de I_LOCK au sein d'une section à déclenchement cyclique), ce qui supprime automatiquement le blocage.

La broche de sortie LOCKCTR donne la valeur actuelle du compteur de blocages général. La valeur de ce compteur augmente à chaque appel du bloc I_LOCK et diminue à chaque appel du bloc I_UNLOCK. Une valeur de compteur égale à zéro signifie que les sections sont débloquentées et que la logique associée est exécutée. Si la valeur de la broche de sortie LOCKCTR n'est pas égale à zéro à la fin d'une section d'événement (pas de I_UNLOCK dans la section), le bit 3 d'état de la section d'événement est mis à 1.

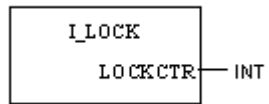
Le compteur d'activations compte uniquement les sections réellement exécutées ou débloquées.

Il est possible de configurer EN et ENO comme paramètres supplémentaires.

Représentation

Symbole

Représentation du bloc :



Description des paramètres

Description des paramètres du bloc :

Paramètres	Type de données	Signification
LOCKCTR	INT	Valeur actuelle du compteur de blocages général. Cette valeur augmente à chaque appel du bloc I_LOCK et diminue à chaque appel du bloc I_UNLOCK. Lorsque le compteur est à 0, toutes les sections sont débloquentées.

LOOPBACK : Resaut

14

Introduction

Ce chapitre décrit le bloc LOOPBACK.

Contenu de ce chapitre

Ce chapitre contient les sujets suivants :

Sujet	Page
Résumé	68
Représentation	69
Description détaillée	70

Résumé

Description de la fonction

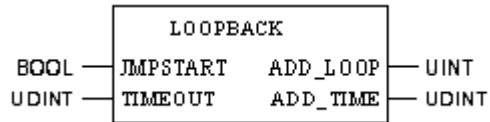
Ce bloc fonction déclenche un saut au début du programme utilisateur (redémarrage du programme utilisateur).

Il est possible de projeter EN et ENO comme paramètres supplémentaires.

Représentation

Symbole

Représentation du bloc :



Description des paramètres

Description des paramètres du bloc :

Paramètre	Type de données	Signification
JMPSTART	BOOL	1 = Le saut va être exécuté
TIMEOUT	UDINT	Durée du chien de garde en microsecondes
ADD_LOOP	UINT	Nombre de cycles de boucle supplémentaires
ADD_TIME	UDINT	Temps en microsecondes pour les cycles supplémentaires

Description détaillée

Déclenchement du saut

Tant que la valeur "0" (FALSE) est présente à l'entrée JMPSTART, aucune fonction ne sera lancée par le bloc fonction. Si la valeur "1" (TRUE) se trouve à l'entrée JMPSTART, le saut vers le début du programme utilisateur sera exécuté aussi longtemps que le temps indiqué sur l'entrée TIMEOUT **n'est pas** encore écoulé.

Adaptation de la durée du chien de garde

Mais le saut n'est effectué que si une durée correcte de chien de garde est indiquée sur l'entrée TIMEOUT. Correcte signifie que la durée de chien de garde doit être supérieure à la durée actuelle d'exécution du programme utilisateur.

NOTE : Tenez compte que la durée du chien de garde est évaluée en **microsecondes** (10 000 correspond à 10 millisecondes). Si l'entrée TIMEOUT a la valeur "0", jamais un saut ne sera effectué.

Nombre de cycles de boucle

La sortie ADD_LOOP indique les cycles supplémentaires de boucle que le programme utilisateur a effectué.

Affichage de la durée pour les cycles supplémentaires

La sortie ADD_TIME indique en microsecondes le temps qui a été nécessaire à l'exécution de ces cycles supplémentaires. Il est possible que cette sortie indique des valeurs inattendues si TIMEOUT a été préréglé avec une petite valeur. C'est pourquoi cette valeur ne doit être utilisée que pour des informations générales (par ex. pour des diagnostics). Elle ne doit pas servir à des calculs ultérieurs.

Résumé

Les sauts au début du programme utilisateur ne sont effectués que si :

- La valeur "1" se trouve sur l'entrée JMPSTART.
- Une durée de chien de garde adéquate (microsecondes) est disponible sur l'entrée TIMEOUT (durée de chien de garde > durée d'exécution du programme utilisateur).
- La durée du chien de garde projetée sur l'entrée TIMEOUT **n'est pas** encore écoulée.

M1HEALTH : Etat de fonction de modules pour M1

15

Introduction

Ce chapitre décrit le bloc M1HEALTH.

Contenu de ce chapitre

Ce chapitre contient les sujets suivants :

Sujet	Page
Résumé	72
Représentation	73

Résumé

Description de la fonction

Ce bloc fonction met à disposition l'état de fonction pour les groupes d'E/S qui sont exploités conjointement avec l'API M1/Momentum.

Une sortie "STATUSx" est respectivement associée à 16 groupes d'E/S. Chaque groupe est représenté par un bit du "STATUSx" de sortie correspondant.

L'association des bits est définie par le câblage des groupes d'E/S. Ici, le bit se trouvant le plus à gauche correspond dans "STATUSx" au groupe d' E/S qui se trouve le plus près de l'API (par rapport à respectivement 16 groupes d'E/S).

Le module local associé à l'API est représenté par la sortie ATIDROP.

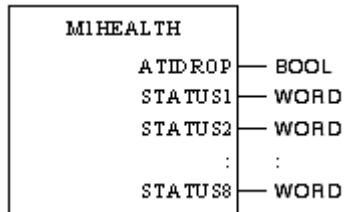
NOTE : Lorsqu'un module de l'embase a été configuré et fonctionne correctement, le bit correspondant est mis à "1".

Il est possible de projeter EN et ENO comme paramètres supplémentaires.

Représentation

Symbole

Représentation du bloc :



Description des paramètres

Description des paramètres du bloc :

Paramètre	Type de données	Signification
ATIDROP	BOOL	Bit d'état de la station locale (ATI=Adaptable I/O Interface)
ETAT1	WORD	Bits d'état des modules E/S 1 ... 16
ETAT2	WORD	Bits d'état des modules E/S 17 ... 32
ETAT3	WORD	Bits d'état des modules E/S 33 .. 48
ETAT4	WORD	Bits d'état des modules E/S 49 ... 64
ETAT5	WORD	Bits d'état des modules E/S 65 .. 80
ETAT6	WORD	Bits d'état des modules E/S 81 .. 96
ETAT7	WORD	Bits d'état des modules E/S 97 .. 112
ETAT8	WORD	Bits d'état des modules E/S 113 ... 128

I_MOVE : affectation protégée contre les interruptions

16

Introduction

Le présent chapitre décrit le bloc I_MOVE.

Contenu de ce chapitre

Ce chapitre contient les sujets suivants :

Sujet	Page
Résumé	76
Représentation	77

Résumé

Description fonctionnelle

Cette fonction affecte la valeur d'entrée à la sortie, ce qui la protège contre les interruptions.

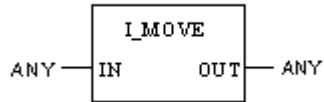
Ce bloc s'avère utile lorsque la variable créée dans ce bloc est utilisée simultanément dans les sections à déclenchement cyclique et les sections d'interruption (sections d'événement de temporisation et d'événement d'E/S). L'affectation de valeur est ainsi protégée contre toute interruption par une section d'événement de temporisation ou une section d'événement d'E/S).

Le bloc MOVE est similaire, mais l'affectation de valeur n'est pas protégée contre les interruptions.

Représentation

Symbole

Représentation du bloc :



Formule

$OUT = IN$

Description des paramètres

Description des paramètres du bloc :

Paramètres	Type de données	Signification
IN	ANY	Valeur d'entrée
OUT	ANY	Valeur de sortie

ONLEVT : Événement connecté

17

Introduction

Ce chapitre décrit le bloc ONLEVT.

Contenu de ce chapitre

Ce chapitre contient les sujets suivants :

Sujet	Page
Résumé	80
Représentation	81

Résumé

Description de la fonction

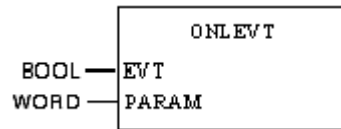
Avec cebloc fonction des états de programme inattendus peuvent être inscrits dans le Buffer de défauts pour l'affichage connecté des évènements. La détection d'erreur "E_EFB_ONLEVT" sert à cet effet. De plus, le paramètre est passé à l'entrée PARAM. Il s'en suit un enregistrement dans le tampon d'erreur lorsque EVT vaut "1".

Il est possible de projeter EN et ENO comme paramètres supplémentaires.

Représentation

Symbole

Représentation du bloc :



Description des paramètres

Description des paramètres du bloc :

Paramètre	Type de données	Signification
EVT	BOOL	"1" : Enregistrement dans le tampon d'erreur de l'affichage en ligne des événements.
PARAM	WORD	Paramètre passé au tampon d'erreur de l'affichage en ligne des événements.

PLCSTAT : Etat de fonction de l'API

18

Introduction

Ce chapitre décrit le bloc PLCSTAT.

Contenu de ce chapitre

Ce chapitre contient les sujets suivants :

Sujet	Page
Résumé	84
Représentation	85
Etat de l'API (PLC_STAT) pour Quantum, Compact, Momentum et Atrium	89
Etat RIO (RIO_STAT) pour le matériel Quantum / B800, Partie I	94
Etat RIO (DIO_STAT) pour Quantum, Partie II	96
Etat d'E/S (RIO_STAT) pour Compact	102
Etat de bus E/S (RIO_STAT) pour Momentum	103
Etat E/S global et état d'itération (DIO_STAT) pour Compact	105

Résumé

Description de la fonction

Ce bloc fonction lie les états internes et les bits de défaut d'un API Quantum et copie ces données dans les structures de données assignées aux sorties correspondantes.

NOTE : Ce bloc fonction n'a été conçu en principe que pour la gamme de produits Quantum. Avec quelques limites, il peut cependant également être utilisé pour les gammes de produits Compact, Momentum et Atrium.

NOTE : Vous pouvez obtenir l'affichage des informations du tableau des états également par la commande **En_ligne** → **Etat API**.

Il est possible de projeter EN et ENO comme paramètres supplémentaires.

Seules les données dont les bits d'entrée (PLC_READ, RIO_READ, DIO_READ) ont la valeur "1" seront lues.

Evaluation pour Quantum

Pour le type d'API Quantum, l'évaluation de PLC_STAT (Etat de l'API), RIO_STAT (Etat E/S) et DIO_STAT (état de communication d'E/S) est possible.

NOTE : Le nom de la sortie DIO_STAT est trompeur. Cette sortie se réfère exclusivement aux informations d'état Remote I/O Drop (S908) et non pas à l'état Distributed I/O. Pour lire l'état Distributed I/O, servez-vous du bloc fonction DIOSTAT (*voir page 19*).

Evaluation pour Compact

Pour le type d'API Compact, l'évaluation de PLC_STAT (Etat de l'API), RIO_STAT (Etat E/S) et DIO_STAT (état de communication d'E/S) est possible.

Evaluation pour Momentum

Pour le type d'API Momentum, ne sont possibles que les évaluations de PLC_STAT (état API) et de RIO_STAT (état E/S).

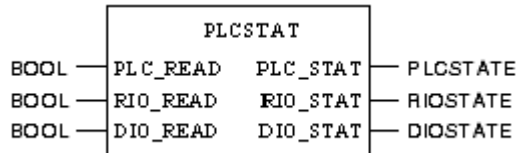
Evaluation pour Atrium

Pour le type d'API Atrium, n'est possible que l'évaluation de PLC_STAT (état API).

Représentation

Symbole

Représentation du bloc :



Description des paramètres PLCSTAT

Description des paramètres du bloc PLCSTAT :

Paramètres	Type de données	Signification
PLC_READ	BOOL	1 = Copie l'état de l'API provenant du tableau d'états sur la sortie PLC_STAT.
RIO_READ	BOOL	1 = Copie l'état RIO provenant du tableau d'états sur la sortie RIO_STAT.
DIO_READ	BOOL	1 = Copie l'état DIO provenant du tableau d'états sur la sortie DIO_STAT.
PLC_STAT	PLCSTATE	Contient l'état de l'API.
RIO_STAT	RIOSTATE	Contient l'état RIO (état E/S) pour le matériel Quantum/B800 ou Contient l'état E/S pour Compact ou Contient l'état bus E/S pour Momentum
DIO_STAT	DIOSTATE	Contient l'état DIO (état de communication E/S) pour Quantum ou Contient l'état E/S global et l'état Répétition pour Compact Rappel : Le nom de cette sortie est trompeur. Cette sortie se réfère exclusivement aux informations d'état Remote I/O Drop (S908) et non pas à l'état Distributed I/O. Pour lire l'état Distributed I/O, servez-vous du bloc fonction DIOSTAT (voir page 19).

Description d'élément PLCSTATE

Description des éléments PLCSTATE :

Elément	Type de données	Signification
word1	WORD	Etat UC
word2	WORD	Etat Hot-Standby (uniquement Quantum)
word3	WORD	Etat API
word4	WORD	Etat RIO (uniquement Quantum, Momentum)
word5	WORD	Etat arrêt API
word6	WORD	Nombre de segments en schéma à contacts (en nombre décimal)
word7	WORD	Adresse du pointeur de fin de logique (EOL)
word8	WORD	Redondance RIO et décommutation de temps (uniquement Quantum, Momentum)
word9	WORD	Etat messages ASCII (uniquement Quantum)
word10	WORD	Etat EN COURS/CHARGE/MISE AU POINT
word11	WORD	Réserve

Description d'élément RIOSTATE pour Quantum

Description des éléments RIOSTATE pour Quantum :

Elément	Type de données	Signification
word1	WORD	Station E/S 1, embase 1
word2	WORD	Station E/S 1, embase 2
...
word5	WORD	Station E/S 1, embase 5
word6	WORD	Station E/S 2, embase 1
word7	WORD	Station E/S 2, embase 2
...
word171	WORD	Station E/S 32, embase 5

Description d'éléments RIOSTATE pour Compact

Description des éléments RIOSTATE pour Compact :

Elément	Type de données	Signification
word1	WORD	Etat E/S, embase 1
word2	WORD	Etat E/S, embase 2
word3	WORD	Etat E/S, embase 3

Elément	Type de données	Signification
word4	WORD	Etat E/S, embase 4
word5	WORD	pas utilisé
...
word160	WORD	pas utilisé

Description d'éléments RIOSTATE pour Momentum

Description des éléments RIOSTATE pour Momentum :

Elément	Type de données	Signification
word1	WORD	Fonctionnement des E/S Momentum local
word2	WORD	Fonctionnement des E/S de bus
word3	WORD	Fonctionnement des E/S de bus
...
word9	WORD	Fonctionnement des E/S de bus
word10	WORD	pas utilisé
...
word160	WORD	pas utilisé

Description d'éléments DIOSSTATE pour Quantum

Description des éléments DIOSSTATE pour Quantum :

Elément	Type de données	Signification
word172	WORD	Numéros de défauts de réarmement : Ce mot est toujours à 0 lorsque le système fonctionne. Si un défaut se produit, l'API ne démarre pas et génère un code d'arrêt
word173	WORD	Câble A défaut
word174	WORD	Câble A défaut
word175	WORD	Câble A défaut
word176	WORD	Câble B défaut
word177	WORD	Câble B défaut
word178	WORD	Câble B défaut
word179	WORD	Défaut de communication global
word180	WORD	Compteur de défaut cumulatif global pour câble A
word181	WORD	Compteur de défaut cumulatif global pour câble B
word182	WORD	Station E/S 1 état Health et compteur de répétition (premier mot)

Elément	Type de données	Signification
word183	WORD	Station E/S 1 état Health et compteur de répétition (deuxième mot)
word184	WORD	Station E/S 1 état Health et compteur de répétition (troisième mot)
word185	WORD	Station E/S 2 état Health et compteur de répétition (premier mot)
...
word275	WORD	Station E/S 32 état Health et compteur de répétition (premier mot)
word276	WORD	Station E/S 32 état Health et compteur de répétition (deuxième mot)
word277	WORD	Station E/S 32 état Health et compteur de répétition (troisième mot)

Description d'éléments DIOSTATE pour Compact

Description des éléments DIOSTATE pour Compact :

Elément	Type de données	Signification
word1	WORD	pas utilisé
...
word10	WORD	pas utilisé
word11	WORD	Etat E/S global
word12	WORD	Compteur de défauts E/S
word13	WORD	Compteur de répétitions PAB
word14	WORD	pas utilisé
...
word106	WORD	pas utilisé

Etat de l'API (PLC_STAT) pour Quantum, Compact, Momentum et Atrium

Généralités

NOTE : Les informations correspondent aux mots du tableau d'état 1 à 11 dans le dialogue **Etat API**.

Lorsque les bits sont à 1, les conditions sont vraies.

Etat de l'API (PLCSTATE : word1)

Occupation des bits :

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
---	---	---	---	---	---	---	---	---	----	----	----	----	----	----	----

Bit	Occupation
6	Autoriser cycle constant
7	Autoriser retard cycle unique
8	1 = 16 bits logique utilisateur 0 = 24 bits logique utilisateur
9	Présence tension AC
10	Témoin Run éteint
11	Protection mémoire désactivée
12	Pile usée
13-16	Réservé

Etat de l'UC redondante (PLCSTATE : word2) (uniquement Quantum)

Occupation des bits :

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
---	---	---	---	---	---	---	---	---	----	----	----	----	----	----	----

Bit	Occupation
1	CHS 110/S911/R911 présente et OK
11	0 = Coulisseau CHS mis sur A 1 = Coulisseau CHS mis sur B
12	0 = Les API ont la même logique 1 = Les API ont une logique qui n'est pas la même

Bit	Occupation
13, 14	Etat du système à distance Déc: binaire 1 0 1 = Offline 2 1 0 = Primaire 3 1 1 = Standby
15, 16	Etat de système local Déc: binaire 1 0 1 = Offline 2 1 0 = Primaire 3 1 1 = Standby

Etat de l'API (PLCSTATE : word3)

Occupation des bits :

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
---	---	---	---	---	---	---	---	---	----	----	----	----	----	----	----

Bit	Occupation
1	Premier cycle
2	Commande de départ non encore exécutée
3	Temps d'échantillonnage constante dépassé
4	Quitter l'état indéfini
13-16	Cycles uniques

Etat du RIO (PLCSTATE : word4) (Quantum)

Occupation des bits :

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
---	---	---	---	---	---	---	---	---	----	----	----	----	----	----	----

Bit	Occupation
1	IOP défectueux
2	Décommutation de temps IOP
3	Boucle de contrôle de l'IOP
4	Défaut mémoire de l'IOP
13-16	00 IO n'a pas répondu 01 pas de réponse 02 boucle de contrôle défailante

Etat du RIO (PLCSTATE : word4) (Momentum)

Pour Momentum, ce mot reçoit le numéro (en format Hex) du premier module défaillant sur le bus.

Etat d'arrêt de l'API (PLCSTATE : word5)

Occupation des bits :

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
---	---	---	---	---	---	---	---	---	----	----	----	----	----	----	----

Bit	Occupation
1	Arrêt du port de périphérique
2	Défaut de parité mémoire étendu (pour automates montés dans boîtier) ou défaut Traffic COP/Quantum/S908 (pour d'autres automates). Si ce bit d'un automate 984B est à 1, une erreur a été détectée dans la mémoire étendue; l'automate fonctionne. Si pour un autre API ce bit = 1, soit une erreur de l'affectation des E/S a été détectée, soit le Quantum/S908 d'une configuration multi-station d'E/S est manquant.
3	API à l'état indéfini
4	Intervention de périphérique incorrecte
5	Ordonnanceur de segment incorrect
6	Le lancement de l'abonné n'a pas lancé le segment
7	Test mémoire d'état a échoué
8	Affectation des E/S incorrecte
9	Temporisation chien de garde écoulée
10	Erreur horloge temps réel
11	Le déclencheur de logique UC n'a pas fonctionné (pour commande montée dans boîtier) ou tableau d'utilisation de sortie/marque/bit (pour une autre commande). Si le bit d'une UC intégrée est à 1, les diagnostics internes ont détecté un blocage de l'UC. Si dans les autres régulateurs ce bit vaut 1, alors la table d'utilisation des bits de sortie/internes ne correspond pas aux bits de sortie/internes de la logique utilisateur.
12	Défaut IOP
13	Abonné incorrect
14	Checksum logique
15	Bit de sortie/interne forcé en mode MARCHE
16	Configuration incorrecte

Etat d'arrêt de l'API (PLCSTATE : word6)

Word 6 indique le nombre de segments; affiche un nombre binaire :

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
---	---	---	---	---	---	---	---	---	----	----	----	----	----	----	----

Bit	Occupation
1-16	Nombre de segments (exprimé en nombre décimal)

Etat d'arrêt de l'API (PLCSTATE : word7)

Word 7 affiche l'adresse du pointeur de fin de logique(EOL) :

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
---	---	---	---	---	---	---	---	---	----	----	----	----	----	----	----

Bit	Occupation
1-16	Adresse du pointeur EOL

Redondance et timeout du RIO (PLCSTATE : word8) (uniquement Quantum, Momentum)

Occupation des bits :

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
---	---	---	---	---	---	---	---	---	----	----	----	----	----	----	----

Bit	Occupation
1	Câble redondant RIO? 0 = non 1 = oui
13-16	Constante timeout RIO

Etat de messages ASCII (PLCSTATE : word9) (uniquement Quantum)

Occupation des bits :

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
---	---	---	---	---	---	---	---	---	----	----	----	----	----	----	----

Bit	Occupation
13	Le nombre de messages et de pointeurs ne correspond pas
14	Pointeur de message incorrect
15	Message incorrect
16	Erreur de checksum message

Etat MARCHE/CHARGEMENT/OPTIMISÉ (PLCSTATE : word10)

Occupation des bits :

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
---	---	---	---	---	---	---	---	---	----	----	----	----	----	----	----

Bit	Occupation
15, 16	Etat de système local
	Déc
	Debug = 0 0 0
	Marche = 0 1 1
	Charger = 1 0 2

PLCSTATE : word11

Réservé pour extensions

Etat RIO (RIO_STAT) pour le matériel Quantum / B800, Partie I

Généralités

NOTE : Ces informations correspondent aux mots 12 à 171 du tableau des états dans la boîte de dialogue Etat de l'API.

Ces mots indiquent l'état de fonctionnement des modules d'E/S.

Respectivement cinq mots sont réservés pour l'une des 32 stations E/S maximales. Chacun de ces mots correspond à l'une des 2 (Quantum) ou 5 (B800) embases possibles (au maximum) dans chaque station E/S.

Affichage du fonctionnement pour le matériel Quantum

Chacune des embases du matériel Quantum peut contenir jusqu'à 15 modules E/S (à l'exception de la première embase ; celle-ci contient au maximum 14 modules E/S). Bit 1... 16 dans chaque mot représente l'affichage du fonctionnement des modules E/S correspondant dans les embases.

Affichage du fonctionnement du matériel B800

Chaque embase de composant B800 peut accepter jusqu'à 11 modules d'E/S. Bit 1... 11 dans chaque mot représente l'affichage fonctionnel des modules E/S correspondant dans les embases.

Etat de fonctionnement des modules E/S

Affectation des bits :

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
---	---	---	---	---	---	---	---	---	----	----	----	----	----	----	----

Bit	Affectation
1	Emplacement 1
2	Emplacement 2
...	...
16	Emplacement 16

Conditions pour un affichage correct du fonctionnement

Afin qu'un module d'E/S puisse délivrer un affichage de fonctionnement correct, quatre conditions doivent être remplies :

- Le trafic des données de l'emplacement doit être surveillé.
- L'emplacement doit être autorisé pour le module installé.

- Il doit exister une communication valide entre le module et l'interface RIO des stations RIO.
- Il doit exister une communication valide entre l'interface RIO de la station RIO et le processeur d'E/S de l'API.

Note : Veuillez observer que les bits de santé de modules d'entrée analogiques sont mis à 1 dans les cas suivants, même si ceci prolonge le temps (1 à 2 secondes) nécessité pour obtenir des valeurs valides.

- Charger le projet dans l'UC et démarrer l'UC (le démarrage / l'arrêt ne prolonge pas la durée de traitement)
- Echange de module pendant le service (modules de remplacement à chaud)
- Activer / désactiver de nouveau les embases locales

Les modules d'entrée analogiques sont les suivants :

- 140 ACI 030 00
- 140 ACI 040 00
- 140 AVI 030 00
- 140 AMM 090 00
- 140 ATI 030 00
- 140 ARI 030 10
- 140 AII 330 00
- 140 AII 330 10
- 140 DDI 364 00

Mots d'état des contrôleurs de manipulation IHM

Les états des panneaux de commande à 32 éléments et des consoles PanelMate d'un réseau RIO peuvent également être contrôlés à l'aide d'un mot d'état de fonctions d'E/S. Les panneaux de commande se trouvent à l'emplacement 4 d'une embase d'E/S et peuvent être surveillés par le bit 4 du mot d'état correspondant. Un PanelMate sur RIO se trouve à l'emplacement 1 de l'embase 1 de la station d'E/S et peut être surveillé par le bit 1 du premier mot d'état de la station d'E/S.

NOTE : La surveillance de l'état de communication du clavier ASCII peut s'effectuer par les codes d'erreur des instructions de lecture/écriture ASCII.

Etat RIO (DIO_STAT) pour Quantum, Partie II

Généralités

NOTE : Ces informations correspondent aux mots 172 à 277 du tableau des états dans la boîte de dialogue Etat de l'API.

Ces mots contiennent l'état de la communication E/S-système (Etat DIO). Les mots 1 à 10 sont des mots d'état globaux. Dans les 96 mots restants, trois sont affectés à chaque fois aux stations d'E/S (jusqu'à 32 stations).

Word 1 mémorise les codes d'erreur à la mise sous tension de Quantum. Ce mot est toujours à 0 lorsque le système fonctionne. Si une erreur survient, l'API ne démarre pas mais génère un état d'arrêt de l'API (Mot 5 de PLC_STAT).

Lorsque les bits sont à 1, les conditions sont vraies.

Codes d'erreur à la mise sous tension (DIOSSTATE : word1)

Lorsque les bits sont à 1, les conditions sont vraies.

Codes d'erreur à la mise sous tension Quantum :

Code	Erreur	Signification (Lieu de l'erreur)
01	BADTCLEN	Longueur de l'affectation des E/S
02	BADLNKNUM	Numéro de liaison RIO
03	BADNUMDPS	Nombre de stations d'E/S dans l'affectation des E/S
04	BADTCSUM	Checksum de l'affectation des E/S
10	BADDLEN	Longueur du descripteur de station d'E/S
11	BADDRPNUM	Numéro de station d'E/S
12	BADHUPTIM	Temps de maintien station d'E/S
13	BADASCNUM	Numéro de port ASCII
14	BADNUMODS	Nombre de modules dans la station d'E/S
15	PRECONDRP	Station d'E/S déjà configurée
16	PRECONPRT	Port déjà configuré
17	TOOMNYOUT	Plus de 1024 points de sortie
18	TOOMNYINS	Plus de 1024 points de sortie
20	BADSLTNUM	Adresse d'emplacement du module
21	BADRCKNUM	Adresse du châssis/de l'embase
22	BADOUTBC	Nombre d'octets de sortie
23	BADINBC	Nombre d'octets d'entrée
25	BADRF1MAP	Premier numéro de référence
26	BADRF2MAP	Deuxième numéro de référence

Code	Erreur	Signification (Lieu de l'erreur)
27	NOBYTES	Pas d'octet d'entrée ou de sortie
28	BADDISMAP	Bit d'E/S-interne n'est pas sur une limite de 16 bits
30	BADODDOUT	Module de sortie impair non apparié
31	BADODDIN	Module d'entrée impair non apparié
32	BADODDREF	Référence de module impair non appariée
33	BAD3X1XRF	1x référence après 3x registre
34	BADDMYMOD	Référence de module factice déjà utilisée
35	NOT3XDMY	Module 3x pas de module vide
36	NOT4XDMY	Module 4x pas de module vide
40	DMYREAL1X	Module factice, puis module 1x réel
41	REALDMY1X	Module réel, puis module factice 1x
42	DMYREAL3X	Module factice, puis module 3x réel
43	REALDMY3X	Module réel, puis module factice 3x

Etat du câble A (DIOSTATE : word2, word3, word4)

Occupation des bits pour word2 :

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
---	---	---	---	---	---	---	---	---	----	----	----	----	----	----	----

Bit	Occupation
1 - 8	Compte les erreurs de trame
9 - 16	Compte les dépassements de réception DMA

Occupation des bits pour word3 :

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
---	---	---	---	---	---	---	---	---	----	----	----	----	----	----	----

Bit	Occupation
1 - 8	Compte les erreurs de réception
9 - 16	Compte les receptions de station d'E/S défectueuses

Occupation des bits pour word4 :

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
---	---	---	---	---	---	---	---	---	----	----	----	----	----	----	----

Bit	Occupation
1	1 = Trame trop courte
2	1 = Pas de fin de trame

Bit	Occupation
13	1 = Erreur CRC
14	1 = Erreur d'alignement
15	1 = Erreur de débordement

Etat du câble B (DIOSTATE : word5, word6, word7)

Occupation des bits pour word5 :

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
---	---	---	---	---	---	---	---	---	----	----	----	----	----	----	----

Bit	Occupation
1 - 8	Compte les erreurs de trame
9 - 16	Compte les dépassements de réception DMA

Occupation des bits pour word6 :

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
---	---	---	---	---	---	---	---	---	----	----	----	----	----	----	----

Bit	Occupation
1 - 8	Compte les erreurs de réception
9 - 16	Compte les receptions de station d'E/S défectueuses

Occupation des bits pour word7 :

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
---	---	---	---	---	---	---	---	---	----	----	----	----	----	----	----

Bit	Occupation
1	1 = Trame trop courte
2	1 = Pas de fin de trame
13	1 = Erreur CRC
14	1 = Erreur d'alignement
15	1 = Erreur de débordement

Etat de la communication globale (DIOSTATE : word8)

Lorsque les bits sont à 1, les conditions sont vraies.

Occupation des bits pour word8 :

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
---	---	---	---	---	---	---	---	---	----	----	----	----	----	----	----

Bit	Occupation
1	Affichage du fonctionnement de comm.
2	Etat câble A
3	Etat câble B
5 - 8	Compteur des communications perdues
9 - 16	Compteur d'itérations cumulatif

Compteur d'erreurs global cumulatif pour le câble A (DIOSTATE : word9)

Lorsque les bits sont à 1, les conditions sont vraies.

Occupation des bits pour word9 :

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
---	---	---	---	---	---	---	---	---	----	----	----	----	----	----	----

Bit	Occupation
1 - 8	Compte les erreurs identifiées
9 - 16	Compte les réponses nulles

Compteur d'erreurs global cumulatif pour le câble B (DIOSTATE : word10)

Lorsque les bits sont à 1, les conditions sont vraies.

Occupation des bits pour word10 :

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
---	---	---	---	---	---	---	---	---	----	----	----	----	----	----	----

Bit	Occupation
1 - 8	Compte les erreurs identifiées
9 - 16	Compte les réponses nulles

Etat RIO (DIOSTATE : word11 à word106)

Les mots 11 à 106 sont utilisés pour décrire l'état de la station RIO; trois mots d'état sont prévus pour chaque station d'E/S.

Le **premier** mot de chaque groupe de trois affiche l'état de communication pour la station E/S correspondante :

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
---	---	---	---	---	---	---	---	---	----	----	----	----	----	----	----

Bit	Occupation
1	Santé de la communication
2	Etat câble A
3	Etat câble B
5 - 8	Compteur des communications perdues
9 - 16	Compteur d'itérations cumulatif

Le **deuxième** mot de chaque groupe de trois est le compteur d'erreurs de station E/S cumulatif sur le câble A de la station E/S correspondante :

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
---	---	---	---	---	---	---	---	---	----	----	----	----	----	----	----

Bit	Occupation
1 - 8	Au moins une erreur dans les mots 2 à 4
9 - 16	Compte les réponses nulles

Le **troisième** mot de chaque groupe de trois est le compteur d'erreurs de station E/S cumulatif sur le câble B pour la station E/S correspondante :

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
---	---	---	---	---	---	---	---	---	----	----	----	----	----	----	----

Bit	Occupation
1 - 8	Au moins une erreur dans les mots 5 à 7
9 - 16	Compte les réponses nulles

NOTE : word11 à word13 sont occupés de la manière suivante pour les API sur lesquelles la station E/S 1 est réservée pour E/S locales :

word11 indique l'état général de la station d'E/S :

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
---	---	---	---	---	---	---	---	---	----	----	----	----	----	----	----

Bit	Occupation
1	Tous les modules OK
9 - 16	compte le nombre de fois qu'un module n'a pas été considéré OK; dépassement du compteur à 255

word12 est utilisé comme compteur d'erreur de bus d'E/S 16 bits .

word13 est utilisé comme compteur d'itération de bus d'E/S 16 bits.

Etat d'E/S (RIO_STAT) pour Compact

Etat d'E/S (RIO_STAT : word1 à 4)

Etat E/S pour word1 à word4 :

word1	Embase 1
word2	Embase 2
word3	Embase 3
word3	Embase 4

Les mots indiquent l'état de fonctionnement des modules d'E/S des 4 embase au maximum.

Chaque mot contient l'état de fonctionnement de 5 modules d'E/S A120 maximum. Le bit de poids fort (à gauche) représente l'état de fonctionnement du module dans l'emplacement 1 du embase.

Lorsqu'un module a été inscrit dans l'affectation des E/S et qu'il est actif, le bit correspondant a la valeur "1". Lorsqu'un module n'a pas été inscrit dans l'affectation des E/S ou qu'il est inactif, le bit correspondant est à "0".

Occupation des bits :

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
---	---	---	---	---	---	---	---	---	----	----	----	----	----	----	----

Bit	Occupation
1	Emplacement 1
2	Emplacement 2
3	Emplacement 3
4	Emplacement 4
4	Emplacement 5

NOTE : Les emplacements 1 et 2 de l'embase 1 (word1) ne sont pas utilisés puisqu'ils sont occupés par l'UC elle-même.

Etat d'E/S (RIO_STAT : word5 à 160)

pas utilisé

Etat de bus E/S (RIO_STAT) pour Momentum

Généralités

NOTE : Ces informations correspondent aux mots 12 à 20 du tableau des états dans la boîte de dialogue Etat de l'API.

Ces mots indiquent l'état de fonctionnement des modules d'E/S.

Le premier mot représente l'aptitude à fonctionner du module local Momentum. Les 8 mots suivants représentent l'aptitude à fonctionner des modules sur le bus, jusqu'à 128.

Lorsque les bits sont à 1, les conditions sont vraies.

Etat RIO (RIOSTATE : word1)

Occupation des bits :

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
---	---	---	---	---	---	---	---	---	----	----	----	----	----	----	----

Bit	Occupation
1	Module local en ordre de marche

Etat RIO (RIOSTATE : word2 à 9)

Occupation des bits :

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
---	---	---	---	---	---	---	---	---	----	----	----	----	----	----	----

Bit	Occupation
1	Module 1
2	Module 2
...	...
16	Module 16

Fonctionnement des modules bus :

word2	Montre le fonctionnement des modules de bus 1 - 16
word3	;Montre le fonctionnement des modules bus 17 - 32
word4	Montre le fonctionnement des modules bus 33 - 48
word5	Montre le fonctionnement des modules bus 49 - 64
word6	Montre le fonctionnement des modules bus 65 - 80
word7	Montre le fonctionnement des modules bus 81 - 96

word8	Montre le fonctionnement des modules bus 97 - 112
word9	Montre le fonctionnement des modules bus 113 - 128

Etat d'E/S (RIO_STAT : word10 à 160)

pas utilisé

Etat E/S global et état d'itération (DIO_STAT) pour Compact

Généralités

NOTE : Ces informations correspondent aux mots 172 à 277 du tableau des états dans la boîte de dialogue Etat de l'API.

Les mots "word11" à "word13" contiennent des informations d'état de fonctionnement et de communication concernant les modules d'E/S installés. Les mots "word1" à "word10" et "word14" à "word106" ne sont pas utilisés.

Lorsque les bits sont à 1, les conditions sont vraies.

DIOSTATE : word1 à 10 et word14 à 106

pas utilisé

Etat d'E/S global (DIOSTATE : word11)

Le bit 1 est positionné lorsque tous les modules sont prêts à fonctionner.

Les bits 9 à 16 sont un compteur indiquant combien de fois un module d'E/S est tombé en panne. Le dépassement de compteur a lieu à 255.

Occupation des bits pour word11 :

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
---	---	---	---	---	---	---	---	---	----	----	----	----	----	----	----

Bit	Occupation
1	Tous les modules OK
9 - 16	Compte le nombre de fois qu'un module n'a pas été considéré OK

Compteur d'erreurs d'E/S (DIOSTATE : word12)

Les bits 9 à 16 sont un compteur indiquant combien de fois un module d'E/S est tombé en panne. Le dépassement de compteur a lieu à 255.

Occupation des bits pour word12 :

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
---	---	---	---	---	---	---	---	---	----	----	----	----	----	----	----

Bit	Occupation
9 - 16	Compte le nombre de fois qu'un module n'a pas été considéré OK

Compteur d'itération BPI (DIOSTATE : word13)

Ce mot indique l'état de communication du BPI (bus parallèle de l'installation). Normalement, ce mot indique la valeur "0". Une erreur est signalée lorsqu'une erreur de bus est encore constatée après 5 répétitions. Dans ce cas, l'API est arrêtée et le code d'erreur "10" est affiché. Des erreurs peuvent par ex. être provoquées par un court-circuit dans l'embase ou par des bruits.

PRJ_VERS : nom/version de projet

19

Introduction

Le présent chapitre décrit le bloc PRJ_VERS.

Contenu de ce chapitre

Ce chapitre contient les sujets suivants :

Sujet	Page
Résumé	108
Représentation	109

Résumé

Description fonctionnelle

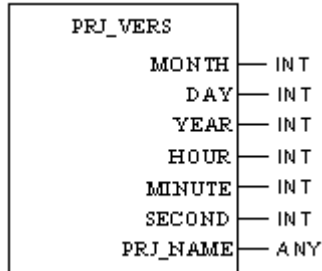
Le bloc indique le nom du projet ainsi que les versions de projet au niveau de ses broches de sortie.

La version de projet est donnée par un groupe date/heure. Le nom du projet peut comprendre jusqu'à 8 caractères par octet.

Représentation

Symbole

Représentation du bloc :



Description des paramètres

Description des paramètres du bloc :

Paramètres	Type de données	Signification
MONTH	INT	Mois : 1-12 (janvier - décembre)
DAY	INT	Jour : 1-31
YEAR	INT	Année (entrée limitée à deux chiffres, par ex. 2001 = 01)
HOUR	INT	Heures : 0-23
MINUTE	INT	Minutes : 0-59
SECOND	INT	Secondes : 0-59
PRJ_NAME	ANY	Nom de projet comportant max. 8 caractères/octet Note : La taille du nom du projet dépend de la taille de la variable définie (en caractères par octet). Si vous placez une variable d'une longueur inférieure à 8 octets, le nom du projet ne peut alors pas dépasser la longueur indiquée.

RES_IEC_INF : Remise à zéro des indicateurs de statut CEI

20

Introduction

Le présent chapitre décrit le bloc RES_IEC_INF.

Contenu de ce chapitre

Ce chapitre contient les sujets suivants :

Sujet	Page
Résumé	112
Représentation	113

Résumé

Description de la fonction

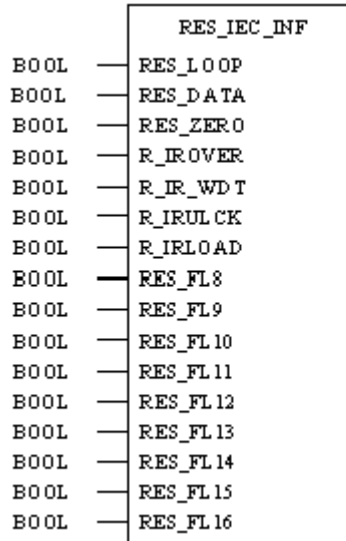
Ce bloc fonction vous permet de remettre individuellement à zéro les indicateurs d'erreur système CEI activés (voir le bloc fonction *GET_IEC_INF : Lecture des indicateurs de statut CEI*, page 27).

Il est possible de configurer EN et ENO comme paramètres supplémentaires.

Représentation

Symbole

Représentation du bloc :



Description des paramètres

Description des paramètres du bloc :

Paramètre	Type de données	Signification
RES_LOOP	BOOL	Pour "1" : l'indicateur LOOP_ON (<i>voir page 29</i>) est remis à zéro.
RES_DATA	BOOL	Pour "1" : l'indicateur DATA_INX (<i>voir page 29</i>) est remis à zéro.
RES_ZERO	BOOL	Pour "1" : l'indicateur DIV_ZERO (<i>voir page 29</i>) est remis à zéro.
R_IROVER	BOOL	Pour "1" : l'indicateur IR_OVERF (<i>voir page 29</i>) est remis à zéro.
R_IR_WDT	BOOL	Pour "1" : l'indicateur IR_WDT (<i>voir page 29</i>) est remis à zéro.
R_IRULCK	BOOL	Pour "1" : l'indicateur IR_ULOCK (<i>voir page 29</i>) est remis à zéro.
R_IRALOAD	BOOL	Pour "1" : l'indicateur IR_ALOAD (<i>voir page 29</i>) est remis à zéro.

Paramètre	Type de données	Signification
RES_F8	BOOL	Pour "1" : l'indicateur RFLAG8 (<i>voir page 29</i>) est remis à zéro. (Indicateur réservé pour une utilisation ultérieure.)
RES_F9	BOOL	Pour "1" : l'indicateur RFLAG9 (<i>voir page 29</i>) est remis à zéro. (Indicateur réservé pour une utilisation ultérieure.)
RES_F10	BOOL	Pour "1" : l'indicateur RFLAG10 (<i>voir page 29</i>) est remis à zéro. (Indicateur réservé pour une utilisation ultérieure.)
RES_F11	BOOL	Pour "1" : l'indicateur RFLAG11 (<i>voir page 29</i>) est remis à zéro. (Indicateur réservé pour une utilisation ultérieure.)
RES_F12	BOOL	Pour "1" : l'indicateur RFLAG12 (<i>voir page 29</i>) est remis à zéro. (Indicateur réservé pour une utilisation ultérieure.)
RES_F13	BOOL	Pour "1" : l'indicateur RFLAG13 (<i>voir page 29</i>) est remis à zéro. (Indicateur réservé pour une utilisation ultérieure.)
RES_F14	BOOL	Pour "1" : l'indicateur RFLAG14 (<i>voir page 29</i>) est remis à zéro. (Indicateur réservé pour une utilisation ultérieure.)
RES_F15	BOOL	Pour "1" : l'indicateur RFLAG15 (<i>voir page 29</i>) est remis à zéro. (Indicateur réservé pour une utilisation ultérieure.)
RES_F16	BOOL	Pour "1" : l'indicateur RFLAG16 (<i>voir page 29</i>) est remis à zéro. (Indicateur réservé pour une utilisation ultérieure.)

REV_XFER : Lecture et écriture des deux registres de transfert reverse

21

Introduction

Ce chapitre décrit le bloc REV_XFER.

Contenu de ce chapitre

Ce chapitre contient les sujets suivants :

Sujet	Page
Résumé	116
Représentation	117

Résumé

Description du fonctionnement

Ce bloc fonction sert à exploiter la fonctionnalité CEI Hot Standby. Il parcourt (en association avec d'autres blocs fonction du groupe HSBY) la configuration de chaque API à la recherche des composants qui lui sont nécessaires. Ces composants se réfèrent toujours au matériel effectivement connecté.

C'est pourquoi un comportement correct de ce bloc fonction ne peut être assuré sur les simulateurs.

Le bloc fonction REV_XFER permet de transmettre deux mots 16 bits (registres 4x) depuis l'API redondant vers l'API primaire. Ceci n'est cependant possible que si la configuration est à redondance d'UC et comprend une zone de non transfert. Les deux registres transmis par ce bloc fonction sont les deux premiers registres 4x de la zone de non transfert (registres de transfert inverse).

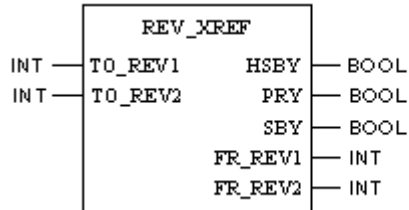
NOTE : Actuellement, ce bloc fonction ne peut être utilisé que de façon restreinte car l'API Standby exécute ici sa logique CEI **PAS**.

Il est possible de projeter EN et ENO comme paramètres supplémentaires.

Représentation

Symbole

Représentation du bloc :



Description des paramètres

Description des paramètres du bloc :

Paramètres	Type de données	Signification
TO_REV1	INT	Décrit le premier registre de transfert en sens inverse, si cette API est l'API en veille.
TO_REV2	INT	Décrit le second registre de transfert en sens inverse, si cette API est l'API en veille.
HSBY	BOOL	1= configuration Redondance d'UC a été détectée et une zone 'Sans transfert' est indiquée dans celle-ci.
PRY	BOOL	1 = cette API est l'API principale.
SBY	BOOL	1 = cette API est l'API en veille.
FR_REV1	INT	Contenu du premier registre de transfert en sens inverse de l'API en veille (Sortie uniquement si HSBY est sur "1").
FR_REV2	INT	Contenu du second registre de transfert en sens inverse de l'API en veille (Sortie uniquement si HSBY est sur "1").

RIOSTAT : Etat de fonction de modules (RIO)

22

Introduction

Ce chapitre décrit le bloc RIOSTAT.

Contenu de ce chapitre

Ce chapitre contient les sujets suivants :

Sujet	Page
Résumé	120
Représentation	121

Résumé

Description de la fonction

Ce bloc fonction met à disposition l'état de fonctionnement pour les modules E/S d'une station E/S (E/S locale/distante).

Il est possible d'utiliser Quantum E/S ou 800 E/S.

Une sortie "STATx" est attribuée à chaque embase. Chacun des modules (emplacement) de cette embase est représenté par un bit de la sortie "STATx" correspondante. Le bit de "STATx" situé le plus à gauche correspond à l'emplacement gauche de l'embase x.

Utilisation de "STAT1" à "STAT5" :

- **E/S Quantum**

Une station E/S possède une seule embase, c'est-à-dire que seule l'utilisation de "STAT1" a lieu.

- **E/S 800**

Une station E/S peut comporter jusqu'à 5 embases, c'est-à-dire que "STAT1" correspond à l'embase 1, "STAT5" correspond à l'embase 5.

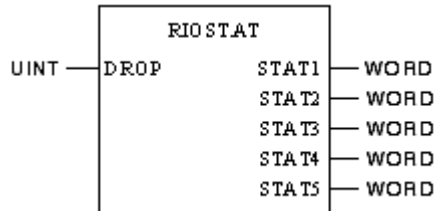
NOTE : Lorsqu'un module de l'embase a été configuré et fonctionne correctement, le bit correspondant est mis à "1".

Il est possible de projeter EN et ENO comme paramètres supplémentaires.

Représentation

Symbole

Représentation du bloc :



Description des paramètres

Description des paramètres du module :

Paramètres	Type de données	Signification
DROP	UINT	Station E/S locale/distante no. (1...32)
STAT1	WORD	Chaîne de bit d'état de l'embase 1
STAT2	WORD	Chaîne de bit d'état de l'embase 2 (800 E/S uniquement)
...
STAT5	WORD	Chaîne de bit d'état de l'embase 5 (800 E/S uniquement)

SAMPLETM : Période d'échantillonnage

23

Introduction

Ce chapitre décrit le bloc SAMPLETM.

Contenu de ce chapitre

Ce chapitre contient les sujets suivants :

Sujet	Page
Brève description	124
Représentation	125

Brève description

Description de la fonction

Ce bloc fonction permet de libérer les blocs fonction de la technique de régulation de manière temporisée.

Pour la commande, la sortie Q du bloc fonction SAMPLETM est reliée à l'entrée EN du bloc fonction de technique de régulation à commander.

Après écoulement de la période saisie à l'entrée INTERVAL, la sortie Q est activée pendant un cycle d'exécution du programme.

L'entrée DELSCAN sert à éviter l'exécution simultanée de plusieurs FFB à traitement cyclique commandés par différents blocs de type SAMPLETM. Cette entrée définit le nombre de cycles devant s'écouler avant l'activation de la sortie Q après une reprise à froid. Cela permet de débloquer pas à pas les blocs fonction à traitement cyclique et de réduire la charge pesant sur l'unité centrale pendant le premier cycle.

Il est possible de configurer EN et ENO comme paramètres supplémentaires.

NOTE : Si ce bloc n'est pas appelé au moins une fois pendant la période correspondant à 2 INTERVAL, une entrée est générée dans l'affichage des événements.

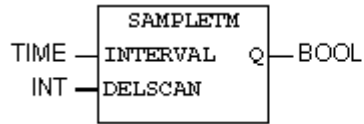
Cela peut être le cas lorsque

- la logique du bloc n'est pas traitée ou
- EN = 0 pendant une période > 2 INTERVAL.

Représentation

Symbole

Représentation du bloc :



Description des paramètres

Description des paramètres du bloc :

Paramètres	Type de données	Signification
INTERVAL	TIME	Période d'échantillonnage pour le bloc fonction technique de régulation raccordé
DELSCAN	INT	Nombre de cycles de retard après une reprise à froid
Q	BOOL	Libération du bloc fonction technique de régulation

SET_TOD : Réglage de l'horloge du matériel (Time Of Day)

24

Introduction

Ce chapitre décrit le bloc SET_TOD.

Contenu de ce chapitre

Ce chapitre contient les sujets suivants :

Sujet	Page
Résumé	128
Représentation	129

Résumé

Description de fonction

Cette fonction parcourt (avec les autres blocs de fonctions du groupe HSBY) la configuration de l'API correspondante à la recherche des composants dont elle a besoin. Ces composants se réfèrent toujours aux composants matériels effectivement connectés.

C'est la raison pour laquelle un comportement correct de ce bloc de fonction sur le simulateur ne peut pas être garanti.

Le bloc de fonction permet de régler l'horloge électronique, si les registres correspondants ont été fournis pour celle-ci dans le cadre de la configuration. Si ces registres ne sont pas disponibles, la sortie TOD_CNF est mise sur "0".

En présence d'un signal "1" à l'entrée S_PULSE, le bloc de fonction lit les valeurs d'entrée et les transmet à l'horloge électronique.

NOTE : Comme S_PULSE est une entrée statique, l'opération d'écriture est active tant que S_PULSE = 1. Cela signifie que S_PULSE doit être remis à 0 à l'issue de l'exécution de l'opération d'écriture, afin de garantir le fonctionnement correct de l'horloge électronique.

Pour toutes les valeurs d'entrée, on applique :

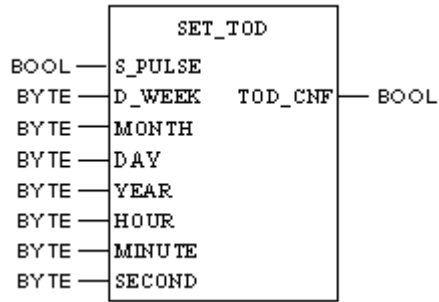
- Si la valeur indiquée est supérieure au maximum, le maximum est utilisé.
- Si la valeur indiquée est inférieure au minimum, le minimum est utilisé.

Les paramètres supplémentaires EN et ENO peuvent être configurés.

Représentation

Symbole

Représentation du bloc :



Description des paramètres

Description des paramètres du bloc :

Paramètres	Type de données	Signification
S_PULSE	BOOL	"1" = Les valeurs d'entrée sont reprises et inscrites dans l'horloge.
D_WEEK	BYTE	Jour de la semaine 1 = dimanche ... 7 = samedi
MONTH	BYTE	Mois 1..12
DAY	BYTE	Jour 1..31
YEAR	BYTE	Année 0..99
HOUR	BYTE	Heure 0..23
MINUTE	BYTE	Minute 0..59
SECOND	BYTE	Seconde 0..59
TOD_CNF	BOOL	"1" = registre 4x pour l'horloge électronique a été détecté et l'horloge est prête à fonctionner. "0" = heure en cours de définition ou horloge électronique introuvable.

SFCCNTRL : Commande SFC

25

Introduction

Ce chapitre décrit le bloc SFCCNTRL.

Contenu de ce chapitre

Ce chapitre contient les sujets suivants :

Sujet	Page
Résumé	132
Représentation	133
Description de la fonction	135
Description des paramètres	137

Résumé

Description de la fonction

Ce bloc fonction sert à commander des chaînes de déroulement.

Il vous permet d'agir sur le déroulement d'une section SFC : A titre d'exemple, des étapes peuvent être commutées, le traitement des conditions de transition peuvent être mises en ou hors service ou la chaîne peut être repositionnée à l'état d'initialisation.

Ce Bloc donne accès aux mêmes actions que les commandes du menu En ligne et du panneau de visualisation. En outre le bloc permet d'éviter un changement du mode opératoire à partir du menu en ligne et du panneau de visualisation.

DANGER

Risque d'opérations incertaines, dangereuses et destructives des outils et processus.

RESETSFC, DISTRANS, DISACT, STEPUN et STEPDEP ne devraient pas être utilisés pour la localisation des défauts sur les commandes de machines-outils, processeurs ou systèmes de gestion des matériaux lorsque ces équipements fonctionnent. Il se pourrait sinon que les outils ou processus concernés exécutent des opérations dangereuses voire même destructrices.

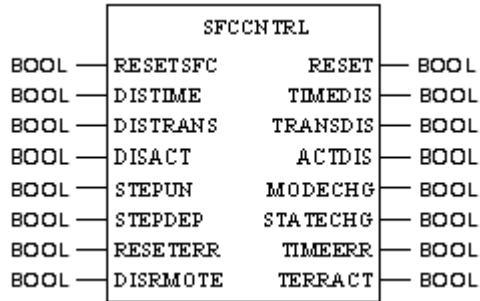
Le non-respect de ces instructions provoquera la mort ou des blessures graves.

Il est possible de projeter EN et ENO comme paramètres supplémentaires.

Représentation

Symbole

Représentation du bloc :



Description des paramètres

Description des paramètres du bloc :

Paramètres	Type de données	Signification
RESETSFC	BOOL	0 -> 1 : Remise à zéro de la chaîne; 1 -> 0 : Démarrer la chaîne de façon normée (pose étape initialer iKette normiert starten (Initialschritt setzen))
DISTIME	BOOL	1 : Arrêt du contrôle de durée des étapes (Ceci n'a aucune influence sur l'animation ou sur la sortie TERRACT.)
DISTRANS	BOOL	1 : Arrêt de l'évaluation des transitions
DISACT	BOOL	1 : Arrêt du traitement des actions et remise à zéro de toutes les actions du graphe.
STEPUN	BOOL	0 -> 1 : Activer la prochaine étape, indépendamment de la transition
STEPDEP	BOOL	0 -> 1 : Activer la prochaine étape, si la condition de transition est remplie
RESETERR	BOOL	0 -> 1 : Eteint l'affichage de tous les défauts de surveillance de temps minimum lors de l'animation de la section SFC . Les défauts de surveillance de temps déjà affichés vont être actualisés. Si aucun défaut de surveillance de temps n'existe, la sortie TERRACT est remise à zéro.
DISRMOTE	BOOL	1 : Empêcher la gestion du graphe via les paramètres de la commande de visualisation en ligne.

Paramètres	Type de données	Signification
RESET	BOOL	1 : Le graphe est initialisé
TIMEDIS	BOOL	1 : Contrôle de durée des étapes arrêté
TRANSDIS	BOOL	1 : Evaluation des transitions arrêtée
ACTDIS	BOOL	1 : Traitement des actions arrêté et toutes les actions du graphe remises à zéro
MODECHG	BOOL	1 : Etat de mode opératoire du graphe modifié
STATECHG	BOOL	1 : Etat du graphe a changé
TIMEERR	BOOL	1 : Erreur survenue dans les temps de contrôle (n'est actif que pendant un cycle)
TERRACT	BOOL	1 : Erreur survenue dans les temps de contrôle (n'est actif que jusqu'à ce que l'erreur passe inactive)

Description de la fonction

Commande d'une chaîne de déroulement

Un bloc fonction SFCCTRL est respectivement valable pour une section SFC.

Il existe 4 possibilités de commande d'un graphe :

- à l'aide des commandes du menu En ligne
- à l'aide du panneau de visualisation (menu En ligne)
- avec le bloc fonction SFCCTRL
- avec le bloc fonction XSFCCTRL

Si une chaîne de déroulement est simultanément commandée par diverses possibilités, ces interventions de commande sont d'égale importance.

Les paramètres d'action de SFC accessibles via les commandes du menu En ligne ou du panneau de visualisation peuvent être verrouillés par le Bloc Fonction SFCCTRL.

NOTE : Pour pouvoir assigner le bloc fonction à une section SFC définie, **le nom de la section SFC** doit être entré en tant que nom de l'instance du bloc fonction SFCCTRL.

On ne peut assurer un traitement correct du module que lorsque celui-ci est placé dans une section exécutée avant la section SFC à commander. Assurez ceci avec l'ordre de menu **Projet** → **Ordre d'exécution...**

Répartition des entrées et sorties

Les entrées et sorties du bloc de fonction se répartissent en 5 groupes :

- Réglages de modes opératoires
 - RESETSFC
 - DISTIME
 - DISTRANS
 - DISACT
- Instructions de contrôle
 - STEPUN
 - STEPDEP
 - RESETERR
- Verrouillage des commandes du menu En ligne.
 - DISRMOTE
- Affichage des réglages de modes opératoires
 - RESET
 - TIMEDIS
 - TRANSDIS
 - ACTDIS (*voir page 139*)

- Affichages généraux
 - MODECHG
 - STATECHG
 - TIMEERR
 - TERRACT

Description des paramètres

Généralités

AVERTISSEMENT

Risque d'opérations dangereuses et destructives des outils ou procédés.

RESETSFC, DISTRANS, DISACT, STEPUN et STEPDEP ne devraient pas être utilisés pour la localisation des défauts sur les commandes de machines-outils, processeurs ou systèmes de gestion des matériaux lorsque ces équipements fonctionnent. Il se pourrait sinon que les outils ou processus concernés exécutent des opérations dangereuses voire même destructrices.

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

RESETSFC

Cette entrée vous permet de remettre le graphe à zéro et de le placer dans l'état initial.

- remettre la chaîne
Avec un front 0 -> 1 à l'entrée, la chaîne est stoppée et toutes les fonctions sont remises à zéro. L'opérateur ne peut pas intervenir.
- Démarrer la chaîne de façon normée
Avec un front 1 -> 0 à l'entrée, la chaîne est remise à zéro, l'étape initiale va donc être activée.

DISTIME (DISable TIME check)

Si un signal 1 est détecté en entrée, la durée des étapes n'est plus contrôlée. Ceci n'a pas d'influence sur l'animation ou sur la sortie TERRACT.

DISTRANS (DISableTRANSitions)

Les transitions ne sont plus évaluées si un signal 1 est détecté en entrée. La séquence conserve son état actuel indépendamment de l'état des transitions. Seules les instructions de commande permettent encore de contrôler la chaîne (RESETSFC, STEPUN, STEPDEP).

DISACT (DISable ACTions)

Si un signal 1 est détecté en entrée, les actions des étapes ne sont plus traitées.

STEPUN (STEP UNconditional)

L'étape suivante est activée avec un front de la donnée en entrée 0 -> 1 et indépendamment de l'état de la transition. Le temps minimal de l'étape (précédente) active devra cependant être écoulé.

Cette instruction active toutes les divergences lorsque des divergences en ET se présentent ; pour les divergences en OU, seule la divergence gauche est activée.

L'ordre STEPDEP sert à activer les dérivations en fonction du process.

STEPDEP (STEP transition DEpendent)

L'étape suivante est activée avec un front 0 -> 1 en entrée et une condition de transition accomplie.

L'ordre de commande ne se justifie qu'avec un signal 1 en entrée DISTRANS.

Si les transitions sont gelées, (DISTRANS = 1), il est possible d'utiliser cette instruction pour éditer progressivement et manuellement les éléments de la chaîne. Dans ce cas, les transitions s'enchaînent en fonction de la condition de transition.

RESETERR (RESET ERRor display)

Avec un front 0 -> 1 à l'entrée, l'affichage de tous les défauts de surveillance de temps minimaux est supprimé lors de l'animation de la section SFC. Les défauts de surveillance de temps déjà affichés sont rafraîchis. La sortie TERRACT est remise à zéro s'il n'y a pas de défauts de surveillance de temps.

DISRMOTE (DISable ReMOTE)

Un signal 1 à l'entrée empêche la commande de SFC par les paramètres d'édition de la commande d'animation connectée (flag de pose/remise à zéro, blocage contrôle de temps, blocage transitions, blocage actions). Néanmoins, le graphe SFC peut être contrôlé par le Bloc Fonction SFCCNTRL.

RESET (mode of RESET)

La sortie prend la valeur 1 lorsque la chaîne est stoppée avec l'ordre Mise à zéro; que cette fonction ait été déclenchée par le bloc fonction (entrée RESETSFC) ou par les ordres connectés SFC. Ainsi, la sortie peut avoir un statut différent de celui de l'entrée RESETSFC.

TIMEDIS (execution mode TIME supervision DISabled)

Prend la valeur 1 lorsque l'affichage des erreurs de temps est hors service, que cette fonction ait été mis hors service par le bloc fonction (entrée DISTIME) ou par les ordres connectés SFC. Ainsi, la sortie peut avoir un statut différent de celui de l'entrée DISTIME.

TRANSDIS (execution mode TRANSitions DISabled)

La sortie prend la valeur 1 lorsque l'évaluation des transitions est stoppée, que cette fonction ait été stoppée par le bloc fonction (entrée DISTRANS) ou par les ordres connectés SFC. Ainsi, la sortie peut avoir un statut différent de celui de l'entrée DISTRANS.

ACTDIS (execution mode ACTions DISabled)

La sortie prend la valeur 1 lorsque la sortie des actions est stoppée, que la sortie ait été stoppée par le bloc fonction (entrée DISACT) ou par les ordres connectés SFC. Ainsi, le statut de la sortie peut être différent de celui de l'entrée DISACT.

MODECHG (execution MODECHanGe)

La sortie prend la valeur 1 pour un cycle lorsqu'un ou plusieurs modes de service de la chaîne ont été modifiés, que cette fonction ait eu lieu par le bloc fonction (entrée RESESTSFC, DISTIME, DISACT ou DISTRANS) ou par les ordres connectés SFC.

STATECHG (sfc STATE CHanGe)

La sortie prend la valeur 1 pour un cycle lorsque l'état de la chaîne a été modifié, que la modification, liée au déroulement de la chaîne, ait été effectuée par le bloc fonction ou par les ordres connectés SFC.

TIMEERR (supervision TIME ERROR)

La sortie prend la valeur 1 pour un cycle lorsqu'un ou plusieurs défauts de surveillance de temps se sont produits.

TERRACT (supervision Time ERRor ACTIVE)

La sortie conserve la valeur 1 tant qu'une ou plusieurs erreurs de temps de contrôle apparaissent.

SKP_RST_SCT_FALSE : Sauter la section restante

26

Introduction

Ce chapitre décrit le bloc SKP_RST_SCT_FALSE.

Contenu de ce chapitre

Ce chapitre contient les sujets suivants :

Sujet	Page
Résumé	142
Représentation	143

Résumé

Description de la fonction

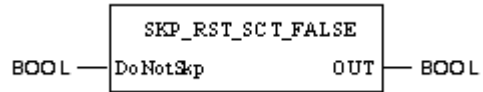
Ce bloc fonction déclenche un saut de la logique qui suit le bloc fonction (en fonction de l'ordre d'exécution FFB) dans la section actuelle. Pour déclencher le saut, l'entrée DoNotSkp nécessite le signal "0" (FALSE).

Il est possible de projeter EN et ENO comme paramètres supplémentaires.

Représentation

Symbole

Représentation du bloc :



Description des paramètres

Description des paramètres du bloc :

Paramètres	Paramètres	Signification
DoNotSkp	BOOL	0 = Le saut va être exécuté
OUT	BOOL	0 = Le saut a été exécuté 1 = Le saut n'a pas été exécuté

SYSCLOCK : Découpages du système

27

Introduction

Ce chapitre décrit le bloc SYSCLOCK.

Contenu de ce chapitre

Ce chapitre contient les sujets suivants :

Sujet	Page
Résumé	146
Représentation	147

Résumé

Description de la fonction

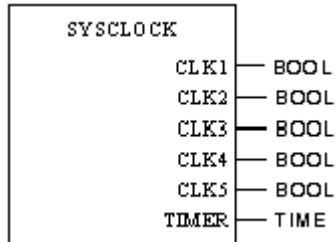
Ce bloc fonction génère des découpages dans les fréquences 0.3125 Hz, 0.6250 Hz, 1.2500 Hz, 2.5000 Hz et 5.0000 Hz. Par ailleurs, le temps depuis le démarrage du système est indiqué.

Il est possible de projeter EN et ENO comme paramètres supplémentaires.

Représentation

Symbole

Représentation du bloc :



Description des paramètres

Description des paramètres du bloc :

Paramètres	Type de données	Signification
CLK1	BOOL	Fréquence 0,3125 Hz (impulsion 3,2 s)
CLK2	BOOL	Fréquence 0.6250 Hz (impulsion 1.6 s)
CLK3	BOOL	Fréquence 1,2500 Hz (impulsion 800 ms)
CLK4	BOOL	Fréquence 2,5000 Hz (impulsion 400 ms)
CLK5	BOOL	Fréquence 5.0000 Hz (impulsion 200 ms)
TIMER	TIME	Temps depuis le démarrage du système (en ms)

SYSSTATE : Etat de système

28

Introduction

Ce chapitre décrit le bloc SYSSTATE.

Contenu de ce chapitre

Ce chapitre contient les sujets suivants :

Sujet	Page
Résumé	150
Représentation	151

Résumé

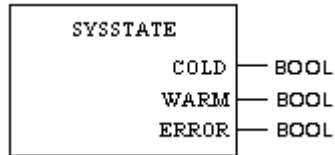
Description de la fonction

Ce bloc fonction affiche l'information d'état du système sur les sorties.
Il est possible de projeter EN et ENO comme paramètres supplémentaires.

Représentation

Symbole

Représentation du bloc :



Description des paramètres

Description des paramètres du bloc :

Paramètres	Type de données	Signification
COLD	BOOL	Pour "1" : Premier cycle, reprise à froid (Est désigné comme reprise à froid le premier démarrage après le chargement complet du projet (En_ligne →Chargement) .)
WARM	BOOL	Pour "1" : Le système est en cycle de reprise à chaud (Est désigné reprise à chaud chaque autre démarrage, par ex. démarrage après la mise sous tension, démarrage de l'API après un arrêt précédent.)
ERROR	BOOL	Pour "1" : Le tampon d'erreur contient des messages d'erreur

NOTE : En cas de reprise à froid sur le premier cycle, les sorties COLD et WARM sont mises à "1".

XSFCCNTRL : Commande SFC étendue

29

Introduction

Ce chapitre décrit le bloc XSFCCNTRL.

Contenu de ce chapitre

Ce chapitre contient les sujets suivants :

Sujet	Page
Résumé	154
Représentation	155
Description de la fonction	157
Description des paramètres	159

Résumé

Description de la fonction

Ce bloc fonction sert à commander des chaînes de déroulement.

Comparé au bloc fonction SFCCTRL , ce bloc fonction offre 2 autres performances.

- Il vous offre la possibilité (entrée ALLTRANS) d'éditer toutes les sections de transition de la section SFC assignée au bloc fonction (même si l'étape correspondante n'est pas active).
- Il vous offre également la possibilité d'obtenir un diagnostic étendu sur les transitions. Pour la valorisation de ce diagnostic de transitions, il vous faut un logiciel spécial de diagnostic de transitions.

Avec ce bloc fonction, vous pouvez influencer le traitement d'une section SFC. A titre d'exemple, des étapes peuvent être commutées, le traitement des conditions de transition peuvent être mises en ou hors service ou la chaîne peut être repositionnée à l'état d'initialisation.

Ce Bloc donne accès aux mêmes actions que les commandes du menu En ligne et du panneau de visualisation. En outre le bloc permet d'éviter un changement du mode opératoire à partir du menu en ligne et du panneau de visualisation.

AVERTISSEMENT

Risque d'opérations incertaines, dangereuses et destructives des outils ou process.

RESETSFC, DISTRANS, DISACT, STEPUN et STEPDEP ne devraient pas être utilisés pour la localisation des défauts sur les commandes de machines-outils, processeurs ou systèmes de gestion des matériaux lorsque ces équipements fonctionnent. Il se pourrait sinon que les outils ou processus concernés exécutent des opérations dangereuses voire même destructrices.

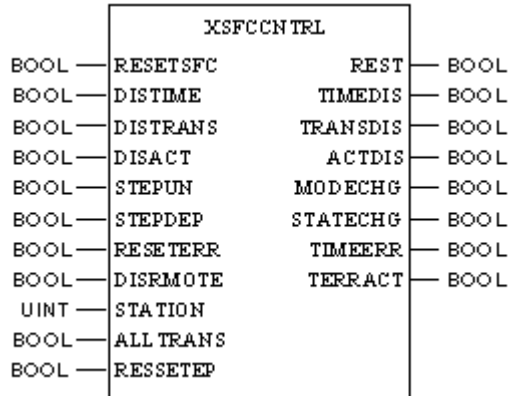
Le non-respect de ces instructions peut provoquer la mort, des blessures graves ou des dommages matériels.

Il est possible de projeter EN et ENO comme paramètres supplémentaires.

Représentation

Symbole

Représentation du bloc :



Description des paramètres

Description des paramètres du bloc :

Paramètres	Type de données	Signification
RESETSFC	BOOL	0 -> 1: Réinitialiser la chaîne ; 1 -> 0: Initialiser la chaîne (mettre au point de départ)
DISTIME	BOOL	1: Désactiver le temporisateur (Ceci n'a aucun effet sur l'animation ou sur la sortie TERRACT.)
DISTRANS	BOOL	1: Désactiver le contrôle des transitions
DISACT	BOOL	1: Désactiver l'édition des actions et réinitialiser toutes les actions de la chaîne
STEPUN	BOOL	0 -> 1: Activer l'étape suivante indépendamment de la transition
STEPDEP	BOOL	0 -> 1: Activer l'étape suivante, si la condition de transition est remplie
RESETEERR	BOOL	0 -> 1: Désactive l'affichage de l'ensemble des erreurs de temporisation minimum lors de l'animation de la section SFC. Les erreurs de temporisation déjà affichées sont mises à jour. Si aucune erreur de temporisation n'est disponible, la sortie TERRACT est réinitialisée.

Paramètres	Type de données	Signification
DISRMOTE	BOOL	1: Empêcher la commande de la SFC à l'aide des paramètres de traitement de la commande d'animation en ligne
STATION	UINT	Numéro de station (le numéro de station "0" est utilisé si aucun numéro n'est fourni.)
ALLTRANS	BOOL	1: Toutes les sections de transitions de la section SFC affectée au bloc de fonction sont traitées.
RESSTEPT	BOOL	1: Le calcul de la durée n'est pas actif. Toutes les durées d'étapes, toutes les erreurs de temporisation et la sortie TERRACT sont réinitialisées tant que le signal est présent. 0: Le calcul de la durée n'est pas actif.
RESET	BOOL	1: La chaîne a été réinitialisée.
TIMEDIS	BOOL	1: La temporisation a été désactivée
TRANSDIS	BOOL	1: Le contrôle des transitions a été désactivé.
ACTDIS	BOOL	1: Désactiver l'édition des actions et réinitialiser toutes les actions de la chaîne
MODECHG	BOOL	1: Le mode de fonctionnement de la chaîne a été modifié.
STATECHG	BOOL	1: L'état de la chaîne a été modifié.
TIMEERR	BOOL	1: Erreur survenue au niveau de la temporisation (présente uniquement pour un cycle)
TERRACT	BOOL	1: Erreur survenue au niveau de la temporisation d'une transition (présente jusqu'au moment où l'erreur est désactivée)

Description de la fonction

Commande d'une chaîne de déroulement

Un bloc fonction XSFCNTRL est respectivement valable pour une section SFC.

Il existe 4 possibilités de commande d'un graphe :

- à l'aide des commandes du menu En ligne
- à l'aide du panneau de visualisation (menu En ligne)
- avec le bloc fonction SFCCNTRL
- à l'aide du bloc fonction XSFCNTRL

Si une chaîne de déroulement est simultanément commandée par diverses possibilités, ces interventions de commande sont de même rang.

Les paramètres d'action de SFC accessibles via les commandes du menu En ligne ou du panneau de visualisation peuvent être verrouillés par le Bloc Fonction SFCCNTRL.

NOTE : Pour pouvoir assigner le bloc fonction à une section SFC définie, le nom de la section SFC doit être indiqué en tant que nom de l'instance du bloc fonction XSFCNTRL.

On ne peut assurer un traitement correct du module que lorsque celui-ci est placé dans une section exécutée avant la section SFC à commander. Assurez ceci avec l'ordre de menu **Projet** → **Ordre d'exécution...**

Répartition des entrées et sorties

Les entrées et sorties du bloc de fonction se répartissent en 5 groupes :

- Réglages de modes opératoires
 - RESETSFC
 - DISTIME
 - DISTRANS
 - DISACT
- Instructions de contrôle
 - STEPUN
 - STEPDEP
 - RESETERR
 - STATION
 - ALLTRANS
 - RESSTPEPT
- Verrouillage des commandes du menu En ligne.
 - DISRMOTE

- Affichage des réglages de modes opératoires
 - RESET
 - TIMEDIS
 - TRANSDIS
 - ACTDIS
- Affichages généraux
 - MODECHG
 - STATECHG
 - TIMEERR
 - TERRACT

Description des paramètres

Généralités

AVERTISSEMENT

Danger dû à des outils ou des processus risquant de ne pas fonctionner en toute sûreté, de fonctionner dangereusement ou de manière destructive.

RESETSFC, DISTRANS, DISACT, STEPUN et STEPDEP ne devraient pas être utilisées pour la détection d'erreurs de commande d'outils mécaniques, processus ou systèmes de gestion du matériel, si ceux-ci sont en cours de fonctionnement. Les outils ou processus raccordés à cette commande risquent de ne pas fonctionner en toute sécurité, de fonctionner dangereusement ou de manière destructive.

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

RESETSFC

Cette entrée vous permet de réinitialiser la chaîne et de la mettre au point de départ.

- Réinitialiser la chaîne
Un front 0 -> 1 à l'entrée permet d'arrêter la chaîne et de réinitialiser toutes les actions. Impossible d'intervenir.
- Initialiser la chaîne
Un front 1 -> 0 à l'entrée permet de réinitialiser la chaîne. Cela signifie que le point de départ est actif.

DISTIME (DISable TIME check)

En présence d'un signal 1 à l'entrée, il n'y a plus de temporisation des étapes. Ceci n'a aucun effet sur l'animation ou sur la sortie TERRACT.

DISTRANS (DISableTRANSitions)

En présence d'un signal 1 à l'entrée, les états des transitions ne sont plus contrôlés. La chaîne conserve son état actuel, indépendamment des signaux perçus aux transitions. La chaîne ne peut plus être utilisée qu'à l'aide des commandes (RESETSFC, STEPUN, STEPDEP).

DISACT (DISable ACTIONS)

En présence d'un signal 1 à l'entrée, les actions des étapes ne sont plus traitées.

STEPUN (STEP UNconditional)

Un front 0 -> 1 à l'entrée permet d'activer l'étape suivante indépendamment de l'état de la transition ; cependant uniquement si le délai d'attente de l'étape active est écoulé.

En cas de branchements simultanés, cette commande permet toujours d'activer l'ensemble des branchements et en cas de branchements alternatifs, celui situé sur la gauche.

La commande STEPDEP permet d'activer des branchements en fonction du processus.

STEPDEP (STEP transition DEpendent)

Un front 0 -> 1 à l'entrée et une condition de transition remplie permettent d'activer l'étape suivante.

La commande n'est utile qu'en présence d'un signal 1 à l'entrée DISTRANS.

En figeant les transitions (DISTRANS = 1), cette commande permet de traiter les éléments de la chaîne manuellement et par étape. A cette occasion, les transitions se poursuivent indépendamment de la condition de transition.

RESETERR (RESET ERRor display)

Un front 0 -> 1 à l'entrée permet de désactiver l'affichage de l'ensemble des erreurs de temporisation minimum lors de l'animation de la section SFC. Les erreurs de temporisation déjà affichées sont mises à jour. Si aucune erreur de temporisation n'est disponible, la sortie TERRACT est réinitialisée.

DISRMOTE (DISable ReMOTE)

Un signal 1 à l'entrée empêche la commande de la SFC à l'aide des paramètres de traitement de la commande d'animation en ligne (drapeau positionnement/remise à zéro, blocage vérification de durée, blocage transitions, blocage actions). Le bloc de fonction SFCCTRL permet tout de même de commander la SFC.

STATION (numéro de STATION)

Numéro de station pour diagnostic de transition. Le numéro de station "0" est utilisé si aucun numéro n'est fourni.

ALLTRANS (traiter TOUTES les TRANSistions)

En présence d'un signal 1 à l'entrée, toutes les sections de transitions de la section SFC affectée au bloc de fonction sont traitées (même si l'étape correspondante n'est pas active). Seul l'état des transitions est déterminé. Ceci n'a aucun effet sur le comportement de la séquence.

Si la case à cocher **Animer toutes les conditions des sections de transitions** est active dans la boîte de dialogue **Options** → **Environnement** → **Editeurs graphiques**, vous pouvez activer l'animation de ces transitions et de cette manière faire afficher l'état détecté des transactions.

NOTE : Le traitement supplémentaire des sections de transitions dont l'étape n'est pas active risque de rallonger énormément le temps de cycle du programme.

RESSTEPT (RESet STEP Time)

Un signal 1 désactive le calcul de la durée. Toutes les durées d'étapes (le temps écoulé depuis qu'une étape a été activée), toutes les erreurs de temporisation et la sortie TERRACT sont réinitialisées, tant que le signal 1 est présent. Tous les affichages d'étapes incorrectes sont désactivés.

NOTE : Pour les experts :

1. En présence d'un signal 1 à l'entrée, le processeur SFC retire les erreurs du tampon de diagnostics.
2. L'entrée n'a aucun effet sur l'"Acquittement automatique".

RESET (mode of RESET)

La sortie devient 1 en cas d'arrêt de la chaîne par la commande de remise à zéro, indépendamment du fait que la remise à zéro ait été déclenchée via le bloc de fonction lui-même (entrée RESETSFC) ou via les commandes SFC en ligne. L'état de sortie et l'état d'entrée RESETSFC peuvent de ce fait diverger.

TIMEDIS (execution mode TIME supervision DISabled)

La sortie devient 1 en cas de désactivation de l'affichage des erreurs de durée, indépendamment du fait que l'affichage ait été désactivé via le bloc de fonction lui-même (entrée DISTIME) ou via les commandes SFC en ligne. L'état de sortie et l'état d'entrée DISTIME. peuvent de ce fait diverger.

TRANSDIS (execution mode TRANSitions DISabled)

La sortie devient 1 en cas d'arrêt du contrôle des transitions, indépendamment du fait que le contrôle ait été désactivé via le bloc de fonction lui-même (entrée DISTRANS) ou via les commandes SFC en ligne. L'état de sortie et l'état d'entrée DISTRANS peuvent de ce fait diverger.

ACTDIS (execution mode ACTIONs DISabled)

La sortie devient 1 en cas de désactivation de la sortie des actions, indépendamment du fait que la sortie ait été désactivée via le bloc de fonction lui-même (entrée DISACT) ou via les commandes SFC en ligne. L'état de sortie et l'état d'entrée DISACT peuvent de ce fait diverger.

MODECHG (execution MODECHAnGe)

La sortie devient 1 pour un cycle, en cas de modification d'un ou plusieurs modes de fonctionnement de la chaîne, indépendamment du fait que la modification ait été apportée via le bloc de fonction lui-même (entrée RESETSFC, DISTIME, DISACT ou DISTRANS) ou via les commandes SFC en ligne.

STATECHG (sfc STATE CHAnGe)

La sortie devient 1 pour un cycle, en cas de modification de l'état de la chaîne, indépendamment du fait que la modification due à la séquence ait eu lieu via le bloc de fonction lui-même ou via les commandes SFC en ligne.

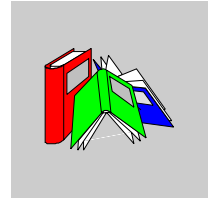
TIMEERR (supervision TIME ERROR)

La sortie devient 1 pour un cycle, en cas de survenance d'une ou plusieurs erreurs de temporisation.

TERRACT (supervision Time ERRor ACTive)

La sortie reste sur 1, tant qu'une ou plusieurs erreurs de temporisation surviennent.

Glossaire



A

Abonné de réseau

Un abonné est un appareil avec une adresse (1 à 64) sur le réseau Modbus Plus.

Abonné local du réseau

L'abonné local est celui qui est projeté à l'instant.

Adresse abonné

L'adresse abonné sert à la désignation univoque d'un abonné du réseau dans l'itinéraire de routage. L'adresse est réglée directement sur l'abonné, p. ex. via le commutateur rotatif situé sur la face arrière du module.

Adresses

Les adresses (directes) sont des zones de mémoire dans l'API. Celles-ci se trouvent dans la mémoire d'état et peuvent être affectées à des modules d'entrée/sortie.

L'affichage/la saisie d'adresses directes est possible dans les formats suivants :

- Format standard (400001)
- Format séparateur (4:00001)
- Format compact (4:1)
- Format CEI (QW1)

Affectation des E/S

L'affectation des E/S est une liste d'affectation générée à partir de la liste d'affectation de l'utilisateur. L'affectation des E/S est gérée dans l'API et contient p. ex. des informations sur l'état des stations et modules E/S, en supplément de la liste d'affectation de l'utilisateur.

ANL_IN

ANL_IN est le type de données "entrée analogique" et est utilisé pour le traitement des valeurs analogiques. Les références 3x du module d'entrée analogique configuré déterminées dans la liste d'affectation des E/S sont affectées automatiquement au type de données et doivent de ce fait être occupées uniquement par des variables non localisées.

ANL_OUT

ANL_OUT est le type de données "sortie analogique" et est utilisé pour le traitement des valeurs analogiques. Les références 4x du module de sortie analogique configuré déterminées dans la liste d'affectation des E/S sont affectées automatiquement au type de données et doivent de ce fait être occupées uniquement par des variables non localisées.

ANY

Dans la présente version, "ANY" comprend les types de données élémentaires BOOL, BYTE, DINT, INT, REAL, UDINT, UINT, TIME et WORD ainsi que les types de données qui en sont dérivés.

ANY_BIT

Dans la présente version, "ANY_BIT" comprend les types de données BOOL, BYTE et WORD.

ANY_ELEM

Dans la présente version, "ANY_ELEM" comprend les types de données BOOL, BYTE, DINT, INT, REAL, UDINT, UINT, TIME et WORD.

ANY_INT

Dans la présente version, "ANY_INT" comprend les types de données DINT, INT, UDINT et UINT.

ANY_NUM

Dans la présente version, "ANY_NUM" comprend les types de données DINT, INT, REAL, UDINT et UINT.

ANY_REAL

Dans la présente version, "ANY_REAL" correspond au type de données REAL.

API

Automate programmable industriel

Appel

La procédure par laquelle l'exécution d'une opération est lancée.

Argument

Synonyme de paramètre réel.

Atrium

L'automate basé sur PC est monté sur platine standard AT et s'utilise au sein d'un ordinateur hôte dans un emplacement de bus ISA. Ce module possède une carte mère (nécessite un pilote SA85) avec deux emplacements pour cartes filles PC104. L'une des cartes filles PC104 sert d'UC et l'autre à la commande INTERBUS.

Avertissement

Si un état critique est identifié lors du traitement d'un FFB ou d'une étape (p. ex. des valeurs d'entrée critiques ou des limites temporelles dépassées), un avertissement est généré. Celui-ci peut être visualisé à l'aide de la commande **En ligne** → **Affichage événements....** Sur les FFB, la sortie ENO reste sur "1".

B**Base de données de projet**

La base de données du PC, contenant les informations de configuration d'un projet.

Bibliothèque

Ensemble d'objets logiciels prévus pour la réutilisation lors de la programmation de nouveaux projets, ou bien même pour l'élaboration de nouvelles bibliothèques. Les exemples sont les bibliothèques des types de blocs fonction élémentaires.

Les bibliothèques EFB peuvent être subdivisées en groupes.

Bits de sortie/bits internes (Références 0x)

Un bit de sortie/bit interne peut être utilisé pour commander des données de sortie réelles via une unité de sortie du système de contrôle, ou pour définir une ou plusieurs sorties TOR dans la mémoire d'état. Remarque : le x suivant immédiatement le premier chiffre du type de référence, représente un emplacement mémoire sur 5 chiffres dans la mémoire de données utilisateur, p. ex. la référence 000201 signifie un bit interne ou de sortie à l'adresse 201 de la mémoire d'état.

Bits d'entrée (Références 1x)

L'état 1/0 des bits d'entrée est commandé par les données du procédé arrivant depuis un périphérique d'entrée dans l'UC.

NOTE : Le x suivant le premier chiffre du type de référence représente un emplacement à cinq chiffres dans la mémoire de données utilisateur, p. ex. la référence 100201 signifie un bit d'entrée à l'adresse 201 de la mémoire d'état.

Bits d'état

Il existe un bit d'état pour chaque abonné à entrée globale, entrée ou sortie spécifique de données de diffusion. Si un groupe de données défini a pu être transmis avec succès avant écoulement du timeout réglé, le bit d'état correspondant est mis à 1. Dans le cas contraire, ce bit est mis à 0 et toutes les données appartenant à ce groupe (à 0) sont effacées.

Bloc fonction (instance) (BF)

Un bloc fonction est une unité d'organisation de programme, qui, en fonction de sa fonctionnalité définie dans la description de type de bloc fonction, calcule des valeurs pour ses sorties et variable(s) interne(s), lorsqu'elle est appelée comme instance particulière. Toutes les valeurs des sorties et variables internes d'une instance particulière de bloc fonction sont conservées d'un appel du bloc fonction au suivant. Des appels répétés de la même instance de bloc fonction avec les mêmes arguments (valeurs des paramètres d'entrée) ne délivrent de ce fait pas forcément la (les) même(s) valeur(s) de sortie.

Chaque instance de bloc fonction est représentée graphiquement par un symbole rectangulaire. Le nom du type de bloc fonction est situé en haut au milieu, à l'intérieur du rectangle. Le nom de l'instance de bloc fonction est également en haut, bien qu'à l'extérieur du rectangle. Il est généré automatiquement à la création d'une instance mais peut, le cas échéant, être modifié par l'utilisateur. Les entrées sont représentées à gauche, les sorties à droite du bloc. Les noms des paramètres formels d'entrée/sortie sont indiqués à l'intérieur du rectangle aux places correspondantes.

La description ci-dessus de la représentation graphique est valable de principe également pour les appels de fonction et pour les appels DFB. Les différences sont décrites dans les définitions correspondantes.

Bobine

Une bobine est un élément LD transmettant sans le modifier l'état de la liaison horizontale sur sa gauche à la liaison horizontale sur sa droite. L'état est alors mémorisé dans la variable/adresse directe associée.

BOOL

BOOL signifie type de données "booléen". La longueur des éléments de données est 1 bit (stocké en mémoire sur 1 octet). La plage de valeurs des variables de ce type de données est 0 (FALSE) et 1 (TRUE).

Bridge

Un bridge est un dispositif permettant de relier des réseaux. Il permet la communication entre abonnés de deux réseaux. Chaque réseau possède sa propre séquence de rotation de jeton - le jeton n'est pas transmis par les bridges.

BYTE

BYTE est le type de données "cordon de bits 8". L'entrée peut se faire en libellé en base 2, libellé en base 8 ou libellé en base 16. La longueur des éléments de données est de 8 bits. Il n'est pas possible d'affecter une plage de valeurs numériques à ce type de données.

C**CEI 611313**

Norme internationale : Automates programmables Partie 3 : Langages de programmation.

Code de section

Le code de section est le code exécutable d'une section. La taille du code de section dépend principalement du nombre de blocs dans la section.

Code DFB

Le code DFB est le code DFB exécutable d'une section. La taille du code DFB dépend principalement du nombre de modules dans la section.

Code EFB

Le code EFB est le code exécutable de tous les EFB utilisés. Les EFB utilisés dans les DFB sont également pris en compte.

Configuration de transmission de données

Paramètres déterminant comment les informations sont transmises depuis votre PC vers l'API.

Connexion série

En connexion série (COM), les informations sont transmises bit par bit.

Constantes

Les constantes sont des variables non localisées, auxquelles est affectée une valeur qui ne peut être modifiée par la logique de programme (lecture seule).

Contact

Un contact est un élément LD transmettant un état sur la liaison horizontale située à sa droite. Cet état est le résultat d'une liaison ET booléenne entre l'état de la liaison horizontale sur sa gauche et l'état de la variable/adresse directe qui lui est affectée. Un contact ne modifie pas la valeur de la variable/adresse directe associée.

Convention CEI sur les noms (Identificateur)

Un identificateur est une suite de lettres, chiffres et caractères de soulignement devant commencer par une lettre ou un caractère de soulignement (p. ex. nom d'un type de bloc fonction, d'une instance, d'une variable ou d'une section). Les lettres des polices de caractères nationales (p. ex. : ö, ü, é, ò) peuvent être utilisées sauf dans les noms de projets et de DFB.

Les caractères de soulignement sont significatifs dans les identificateurs ; p. ex. "A_BCD" et "AB_CD" seront interprétés comme des identificateurs différents. Plusieurs caractères de soulignement de tête ou de suite ne sont pas autorisés.

Les identificateurs ne doivent pas comporter d'espaces. Les majuscules/minuscules ne sont pas significatives ; p. ex. "ABCD" et "abcd" seront interprétés comme le même identificateur.

Les identificateurs ne doivent pas être des mots-clés.

Cordon de bits

C'est un élément de données constitué d'un ou de plusieurs bits.

Cycle programme

Un cycle programme consiste en la lecture des entrées, le traitement de la logique de programme et l'édition des sorties.

D

DDE (Echange dynamique de données)

L'interface DDE permet à deux programmes sous Windows d'échanger des données en dynamique. L'utilisateur peut se servir de l'interface DDE en moniteur étendu afin d'appeler ses propres applications d'affichage. Avec cette interface, l'utilisateur (c.-à-d. le client DDE) peut non seulement lire des données du moniteur étendu (le serveur DDE), mais peut également écrire des données sur l'API via le serveur. L'utilisateur peut ainsi modifier directement des données dans l'API tout en surveillant et en analysant les résultats. Lors de l'utilisation de cette interface, l'utilisateur peut créer son propre "Outil graphique", "Face Plate" ou "Outil de réglage", et intégrer celui-ci dans le système. Ces outils peuvent être écrits dans n'importe quel langage que le DDE prend en charge, p. ex. Visual Basic, VisualC++. Ils sont appelés lorsque l'utilisateur actionne l'un des boutons de commande de la boîte de dialogue Moniteur étendu. Outil graphique Concept : grâce au lien DDE entre Concept et l'outil Graphique Concept, il est possible de représenter les signaux d'une configuration sous forme de chronogramme.

Déclaration

Le mécanisme qui permet d'établir la définition d'un élément de langage. Normalement, une déclaration nécessite le rattachement d'un identificateur à l'élément de langage et l'affectation d'attributs, tels que les types de données et les algorithmes.

Défaut

Si, lors du traitement d'un FFB ou d'une étape, une erreur est détectée (p. ex. valeurs d'entrée non autorisées ou erreur de durée), un message d'erreur est généré, lequel peut être visualisé à l'aide de la commande **En ligne → Affichage événements...** Sur les FFB la sortie ENOest mise à "0".

Défragmentation

La défragmentation permet de supprimer les trous indésirables dans la zone mémoire (générés, p. ex., en effaçant des variables inutilisées).

Voyez également **Sélection automate** dans l'aide de contexte.

Derived Function Block (DFB) (Bloc fonction dérivé)

Un bloc fonction dérivé représente l'appel d'un type de bloc fonction dérivé. Vous trouverez des détails de la forme graphique de l'appel dans la définition "Bloc fonction (instance)". Contrairement aux appels de types d'EFB, les appels de types DFB sont caractérisés par des lignes verticales doubles sur les côtés gauche et droit du symbole rectangulaire du bloc.

Le corps d'un type de bloc fonction dérivé est projeté en langage FBD, langage LD, langage ST et langage IL quoique seulement dans la version actuelle du système de programmation. Les fonctions dérivées ne peuvent pas encore être définies dans la version actuelle.

On fait la distinction entre les DFB locaux et globaux.

DFB globaux

Les DFB globaux sont disponibles dans tout projet Concept. Le stockage des DFB globaux dépend de la configuration dans le fichier CONCEPT.INI.

DFB locaux

Les DFB locaux ne sont disponibles que dans un seul projet Concept et sont enregistrés dans le répertoire DFB sous le répertoire de projet.

Diagramme fonctionnel en séquence (SFC)

Les éléments de langage SFC permettent de subdiviser une unité d'organisation de programme en un certain nombre d'étapes et de transitions, reliées entre elles par des liaisons dirigées. A chaque étape correspond un nombre d'actions et à chaque transition est associée une condition de transition.

DINT

DINT signifie type de données "entier double (double integer)". L'entrée s'effectue en libellé entier, libellé en base 2, libellé en base 8 ou libellé en base 16. La longueur des éléments de données est de 32 bits. La plage de valeurs pour les variables de ce type de données va de $-2 \exp(31)$ à $2 \exp(31) - 1$.

Données d'instance DFB

Les données d'instance DFB sont des données internes des instructions chargeables dérivées utilisées dans le programme.

Données de section

Les données de section sont les données locales d'une section, comme par ex. les libellés, les liaisons entre blocs, les entrées et sorties de bloc non liées, la mémoire d'état interne des EFB.

NOTE : Les données qui sont configurées dans les DFB de cette section ne sont pas des données de section.

Données globales

Les données globales sont des variables non localisées.

DP (PROFIBUS)

DP = Dezentrale Peripherie (périphérie décentralisée)

DX Zoom

Cette caractéristique vous permet de vous raccorder sur un objet de programmation afin d'en surveiller des valeurs et de les modifier, si nécessaire.

E

Elément de langage

Chaque élément de base dans l'un des langages de programmation CEI, p. ex. une étape en SFC, une instance de bloc fonction en FBD ou la valeur de départ d'une variable.

EN / ENO (autorisation / affichage d'erreur)

Si la valeur de EN vaut "0", lorsque le FFB est lancé, les algorithmes définis par le FFB ne sont pas exécutés et toutes les sorties conservent leur valeur précédente. La valeur de ENO est dans ce cas mise automatiquement à "0". Si la valeur de EN est "1" lors de l'appel du FFB, les algorithmes définis par le FFB seront exécutés. Après l'exécution sans erreur de ces algorithmes, la valeur de ENO est mise automatiquement à "1". Si une erreur survient lors de l'exécution de ces algorithmes, ENO est mis automatiquement à "0". Le comportement de sortie des FFB est indépendant du fait que ceux-ci sont appelés sans EN/ENO ou avec EN=1. Si l'affichage de EN/ENO est activé, l'entrée EN doit absolument être câblée. Le FFB n'est sinon jamais exécuté. L'activation/la désactivation de EN et ENO se fait dans la boîte de dialogue des caractéristiques du bloc fonction. Cette boîte de dialogue est appelée via **Objets** → **Propriétés...** ou en double-cliquant sur le FFB.

Erreur d'exécution

Erreur survenant lors du traitement du programme sur l'API sur des objets SFC (p. ex. des étapes) ou des FFB. Il s'agit p. ex. de dépassement de plage de valeurs sur les compteurs ou bien d'erreurs temporelles sur les étapes.

Etape

Élément de langage SFC : situation dans laquelle le comportement d'un programme suit, en fonction de ses entrées et sorties, les opérations définies par les actions correspondantes de l'étape.

Etape initiale (Etape de départ)

L'étape de démarrage d'une séquence. Une étape initiale doit être définie dans chaque séquence. La séquence est démarrée à son premier appel par l'étape initiale.

Evaluation

C'est le processus par lequel est déterminé une valeur d'une fonction ou des sorties d'un bloc fonction lors de l'exécution du programme.

Expression

Les expressions sont constituées d'opérateurs et d'opérandes.

F

Fenêtre active

Il s'agit de la fenêtre momentanément sélectionnée. Pour un instant donné, seule une fenêtre peut être active. Lorsqu'une fenêtre devient active, la couleur de sa barre de titre change afin de la distinguer des autres fenêtres. Les fenêtres non sélectionnées ne sont pas actives.

Fenêtre de document

Une fenêtre contenue dans une fenêtre d'application. Plusieurs fenêtres de document peuvent être ouvertes simultanément dans une fenêtre d'application. Mais seule une fenêtre de document peut être active. Les fenêtres de document dans Concept sont p. ex. les sections, la fenêtre des messages, l'éditeur de données de référence et la configuration de l'automate.

Fenêtre d'application

Il s'agit de la fenêtre contenant l'espace de travail, la barre de menus et la barre d'outils du programme applicatif. Le nom du programme applicatif apparaît dans la barre de titre. Une fenêtre d'application peut contenir plusieurs fenêtres de document. Dans Concept, la fenêtre d'application correspond à un projet.

FFB (fonctions/blocs fonction)

Terme générique désignant les EFB (fonctions/blocs fonction élémentaires) et les DFB (blocs fonction dérivés)

Fichier de code source (Concept-EFB)

Le fichier de code source est un fichier source ordinaire en C++. Après exécution de la commande **Bibliothèque** → **Créer des fichiers**, ce fichier contient un cadre de code EFB dans lequel vous devez porter un code spécifique de l'EFB sélectionné. Pour ce faire, lancez la commande **Objets** → **Source**.

Fichier de définition (Concept-EFB)

Le fichier de définition contient des informations générales de description de l'EFB sélectionné et ses paramètres formels.

Fichier de sauvegarde (Concept-EFB)

Le fichier de sauvegarde est une copie du dernier fichier de code source. Le nom de ce fichier de sauvegarde est "backup??.c" (on suppose ce faisant que vous n'avez jamais plus de 100 copies de votre fichier de sauvegarde). Le premier fichier de sauvegarde porte le nom "backup00.c". Si vous avez procédé à des modifications dans le fichier de définition n'entraînant pas de modification d'interface pour l'EFB, vous pouvez vous dispenser de créer un fichier de sauvegarde en éditant son fichier de code source (**Objets** → **Source**). Si un fichier de sauvegarde est créé, vous pouvez lui donner le nom Fichiersource.

Fichier factice

Il s'agit d'un fichier vide constitué d'un en-tête contenant diverses informations générales sur le fichier, comme l'auteur, la date de création, la désignation de l'EFB, etc. L'utilisateur doit procéder à la préparation de ce fichier factice à l'aide d'entrées supplémentaires.

Fichier prototype (Concept-EFB)

Le fichier prototype contient tous les prototypes des fonctions affectées. On indique en outre, si elle existe, une définition type de la structure de la situation interne.

Fichier Template (Concept-EFB)

Le fichier Template est un fichier ASCII contenant des informations de mise en page pour l'éditeur FBD de Concept, ainsi que des paramètres pour la génération de code.

Filtre RIF

(Filtre Finite Impulse Response) Filtre à réponse impulsionnelle finie

Filtre RII

(Filtre Infinite Impulse Response) Filtre à réponse impulsionnelle infinie

Fonction (FUNK)

Une unité d'organisation de programme délivrant à l'exécution exactement un élément de donnée. Une fonction ne dispose pas d'information de situation interne. Les appels répétés de la même fonction avec les mêmes paramètres d'entrée délivrent toujours les mêmes valeurs de sortie.

Vous trouverez des détails de la forme graphique des appels de fonction dans la définition "Bloc fonction (instance)". Contrairement aux appels de blocs fonction, les appels de fonction ne disposent que d'une unique sortie sans nom, son nom étant le nom de la fonction elle-même. En FBD, chaque appel est caractérisé par un numéro unique par le bloc graphique ; ce numéro est créé automatiquement et ne peut pas être modifié.

Fonctions/blocs fonction élémentaires (EFB)

Caractérisation des fonctions ou des blocs fonction, dont les définitions de type n'ont pas été formulées dans l'un des langages CEI, c.-à-d. dont les corps p. ex. ne peuvent être modifiés à l'aide de l'éditeur DFB (Concept-DFB). Les types EFB sont programmés en "C" et sont mis à disposition en forme précompilée par les bibliothèques.

Format CEI (QW1)

Au début de l'adresse se trouve un identificateur conforme à CEI, suivi de l'adresse à cinq chiffres :

- %0x12345 = %Q12345
- %1x12345 = %I12345
- %3x12345 = %IW12345
- %4x12345 = %QW12345

Format compact (4:1)

Le premier chiffre (la référence) est séparé par deux points (:) de l'adresse suivante, les zéros de tête n'étant pas indiqués dans l'adresse.

Format séparateur (délimiteur) (4:00001)

Le premier chiffre (la référence) est séparé par deux-points (:) de l'adresse à cinq caractères.

Format standard (400001)

L'adresse à cinq positions se situe juste après le premier chiffre (la référence).

G**Groupes (EFB)**

Quelques bibliothèques EFB (p. ex. la bibliothèque CEI) sont subdivisées en groupes. Cela simplifie, particulièrement dans les importantes bibliothèques, la recherche des EFB.

I**Instanciation**

La création d'une instance.

Instruction (IL)

Les instructions sont des "commandes" du langage de programmation IL. Chaque instruction commence à une nouvelle ligne et est suivie d'un opérateur, le cas échéant avec modificateur, et, si nécessaire pour l'opération concernée, d'un ou de plusieurs opérandes. Si l'instruction utilise plusieurs opérandes, ceux-ci sont séparés par des virgules. Devant l'instruction peut se trouver une étiquette suivie de deux points. Le commentaire doit, s'il existe, être le dernier élément de la ligne.

Instruction (LL984)

La mission d'un utilisateur lors de la programmation d'automatismes électriques est de mettre en oeuvre des instructions codées de façon opérationnelle sous forme d'objets imagés classés selon les formes identifiables de contact. Les objets du programme ainsi conçus sont convertis au niveau utilisateur en codes opérands utilisables par l'ordinateur, et ce lors de la procédure de chargement. Les codes opérands sont décodés dans l'UC et traités par les fonctions micrologicielles du contrôleur, de sorte que la commande désirée soit ainsi mise en oeuvre.

Instruction (ST)

Les instructions sont des "commandes" du langage de programmation ST. Les instructions doivent se terminer par des points-virgules. Plusieurs instructions (séparées par des points-virgules) peuvent se trouver sur une même ligne.

INT

INT correspond au type de données "nombre entier (integer)". L'entrée s'effectue en libellé entier, libellé en base 2, libellé en base 8 ou libellé en base 16. La longueur des éléments de données est de 16 bits. La plage de valeurs pour les variables de ce type de données va de $-2 \exp (15)$ à $2 \exp (15) - 1$.

Interbus S (PCP)

Afin d'utiliser le canal PCP de l'Interbus S et le prétraitement de données de procédé Interbus S (PDV), le configurateur Concept propose maintenant le nouveau type de station d'E/S Interbus S (PCP). A ce type de station d'E/S est affecté de manière fixe le module de connexion Interbus 180-CRP-660-01.

Le module 180-CRP-660-01 se distingue du 180-CRP-660-00 seulement par une plage d'E/S sensiblement plus importante dans la mémoire d'état de l'automate.

J

Jeton

Le jeton du réseau régit la possession momentanée du droit de transmission d'un abonné individuel. Le jeton circule entre les abonnés dans un sens circulaire (croissant) des adresses. Tous les abonnés suivent la rotation du jeton et peuvent obtenir toute sorte de données qui y sont véhiculées.

L

Langage en blocs fonctionnels (FBD)

Une ou plusieurs sections contenant des réseaux représentés graphiquement composés de fonctions, blocs fonction et liaisons.

Liaison

Une liaison de contrôle ou de données entre objets graphiques (p. ex. étapes dans l'éditeur SFC, blocs fonction dans l'éditeur FBD) au sein d'une section, graphiquement représenté par une ligne.

Liaison locale (Local Link)

La liaison locale de réseau est le réseau reliant l'abonné local à d'autres abonnés, soit directement soit par l'amplificateur de bus.

Liaisons binaires

Il s'agit de liaisons entre des sorties et des entrées de FFB de type de données BOOL.

Libellé

Les libellés servent à fournir des valeurs directement aux entrées des FFB, conditions de transition etc... Ces valeurs ne peuvent pas être écrasées par la logique du programme (lecture seule). Le système distingue les libellés génériques des libellés classés par type.

De plus, les libellés servent à affecter une valeur à une constante ou une valeur initiale à une variable.

L'entrée se fait en libellé en base 2, libellé en base 8, libellé en base 16, libellé entier, libellé réel ou libellé réel avec exposant.

Libellé de durée

Les unités permises pour les durées (TIME) sont les jours (J), les heures (H), les minutes (M), les secondes (S) et les millisecondes (MS) ou une combinaison de ceux-ci. La durée doit être caractérisée par le préfixe t#, T#, time# ou TIME#. Le "dépassement" de l'unité de plus grande valeur est admise; p. ex. l'entrée T#25H15M est permise.

Exemple

t#14MS, T#14.7S, time#18M, TIME#19.9H, t#20.4D, T#25H15M,
time#5D14H12M18S3.5MS

Libellé en base 16

Les libellés en base 16 servent à codifier les entiers dans le système hexadécimal. La base doit être repérée par le préfixe 16#. Les valeurs doivent être non signées (+/). Les caractères de soulignement individuels (_) entre les chiffres ne sont pas significatifs.

Exemple

16#F_F ou 16#FF (décimal 255)

16#E_0 ou 16#E0 (décimal 224)

Libellé en base 2

Les libellés en base 2 servent à la codification de valeurs entières dans le système de base 2. La base doit être repérée par le préfixe 2#. Les valeurs doivent être non signées (+/). Les caractères de soulignement individuels (_) entre les chiffres ne sont pas significatifs.

Exemple

2#1111_1111 ou 2#11111111 (255 décimal)

2#1110_0000 ou 2#11100000 (224 décimal)

Libellé en base 8

Les libellés en base 8 servent à codifier les entiers dans le système de base 8. La base doit être repérée par le préfixe 8#. Les valeurs doivent être non signées (+/). Les caractères de soulignement individuels (_) entre les chiffres ne sont pas significatifs.

Exemple

8#3_77 ou 8#377 (255 décimal)

8#34_0 ou 8#340 (décimal 224)

Libellé entier

Les libellés entiers servent à indiquer des valeurs entières dans le système décimal. Les valeurs peuvent être signées (+/). Les caractères de soulignement individuels (_) entre les chiffres ne sont pas significatifs.

Exemple

-12, 0, 123_456, +986

Libellés classés par type

Si vous voulez déterminer le type de données d'un libellé, vous pouvez le faire avec la construction suivante : 'nomtypedonnée'#'Valeur du libellé'

Exemple

INT#15 (type de données : entier, valeur : 15),

BYTE#00001111 (type de données : octet, valeur : 00001111)

REAL#23.0 (type de données : réel, valeur : 23,0)

Pour l'affectation du type de données REAL, vous pouvez indiquer la valeur de la manière suivante : 23.0.

En indiquant ce point décimal, le type de données REAL est affecté automatiquement.

Libellés génériques

Si le type de données d'un libellé n'a pas d'importance pour vous, indiquez la valeur du libellé. Dans ce cas, Concept affecte automatiquement un type de données adéquat au libellé.

Libellés réels

Les libellés réels servent à indiquer les valeurs à virgule flottante dans le système décimal. Les libellés réels s'identifient au point décimal. Les valeurs peuvent être signées (+/-). Les caractères de soulignement individuels (_) entre les chiffres ne sont pas significatifs.

Exemple

-12.0, 0.0, +0.456, 3.14159_26

Libellés réels avec exposant

Les libellés réels avec exposant servent à indiquer les valeurs à virgule flottante dans le système décimal. Les libellés réels avec exposant se caractérisent par le point décimal. L'exposant donne la puissance de dix avec lequel le chiffre de devant doit être multiplié pour obtenir la valeur à représenter. La base peut être précédée d'un signe moins (). L'exposant peut être signé (+/-). Les caractères de soulignement individuels (_) entre les chiffres ne sont pas significatifs. (Uniquement entre les chiffres, et non avant ou après la virgule ou avant ou après "E", "E+" ou "E-")

Exemple

-1.34E-12 ou -1.34e-12

1.0E+6 ou 1.0e+6

1.234E6 ou 1.234e6

Liste d'affectation des E/S

Dans la liste d'affectation des E/S, on configure les modules d'E/S et modules experts des différentes unités centrales.

Liste d'instructions (IL)

IL est un langage littéral conforme à la norme CEI 1131, dans lequel les opérations, telles que les appels sur ou sans condition de blocs fonction et de fonctions, les sauts conditionnels ou sans condition, etc., sont représentées par des instructions.

Littéral structuré (ST)

ST est un langage littéral conforme à la CEI 1131, dans lequel les opérations, comme le lancement de blocs fonction et de fonctions, les exécutions conditionnelles d'instructions, la répétition d'instructions, etc. sont représentés par des instructions.

M

Macro

Les macros sont créées à l'aide du logiciel Concept-DFB.

Les macros servent à dupliquer des sections et des réseaux fréquemment utilisés (y compris leur logique, leurs variables et leur déclaration de variable).

On fait la distinction entre les macros locales et globales.

Les macros possèdent les caractéristiques suivantes :

- Les macros ne peuvent être créées qu'avec les langages FBD et LD
- Les macros ne contiennent qu'une seule section
- Elles peuvent contenir une section d'une complexité quelconque
- D'un point de vue programme, une macro instanciée, c.-à-d. une macro insérée dans une section, ne se distingue pas d'une section créée de manière conventionnelle.
- Appel de DFB dans une macro
- Déclaration de variables
- Utilisation de structures de données propres aux macros
- Validation automatique des variables déclarées dans la macro
- Valeurs initiales des variables
- Instanciation multiple d'une macro dans tout le programme avec différentes variables
- Le nom de la section, les noms des variables et le nom de la structure de données peuvent comporter jusqu'à 10 marques d'échange (@0 à @9) différentes.

Macros globales

Les macros globales sont disponibles dans tout projet Concept et sont enregistrées dans le répertoire DFB directement situé sous le répertoire Concept.

Macros locales

Les macros locales ne sont disponibles que dans un seul projet Concept et sont enregistrées dans le répertoire DFB sous le répertoire de projet.

Mémoire du programme CEI

La mémoire du programme CEI comprend le code programme, le code EFB, les données de section et les données d'instance DFB.

Mémoire d'état

La mémoire d'état est l'emplacement mémoire pour toutes les grandeurs sollicitées dans le programme utilisateur par des références (représentation directe). Par exemple les bits d'entrée, les bits de sortie/bits internes, les mots d'entrée et mots de sortie/mots internes se trouvent en mémoire d'état.

MMI

Interface Homme-Machine

Mode ASCII

American Standard Code for Information Interchange. Le mode ASCII est utilisé pour la communication avec différents équipements hôte. ASCII fonctionne sur 7 bits de données.

Mode RTU

Remote Terminal Unit

Le mode RTU est utilisé pour la communication entre l'API et un ordinateur personnel compatible IBM. RTU fonctionne sur 8 bits de données.

Module SA85

Le module SA85 est une carte Modbus Plus pour ordinateur IBM-AT ou compatible.

Mots de sortie/mots internes (Références 4x)

Un mot de sortie/mot interne peut être utilisé pour la mémorisation de données numériques (binaires ou décimales) en mémoire d'état, ou bien pour envoyer des données depuis l'UC vers une unité de sortie du système de contrôle. Remarque : le x suivant immédiatement le premier chiffre du type de référence, représente un emplacement mémoire à cinq chiffres dans la mémoire de données utilisateur, p.ex. la référence 400201 signifie un mot de sortie/mot interne de 16 bits à l'adresse 201 de la mémoire d'état.

Mots d'entrée (Références 3x)

Un mot d'entrée contient des informations émanant d'une source externe et par lesquelles un nombre sur 16 bits est représenté. Un registre 3x peut également contenir 16 bits successifs lus dans le registre au format binaire ou BCD (binaire codé décimal). Remarque : le x suivant immédiatement le premier chiffre du type de référence, représente un emplacement mémoire à cinq chiffres dans la mémoire de données utilisateur, p.ex. la référence 300201 signifie un mot d'entrée de 16 bits à l'adresse 201 de la mémoire d'état.

Mots-clés

Les mots-clés sont des combinaisons uniques de caractères utilisés comme éléments spéciaux de syntaxe comme il est défini à l'annexe B de la CEI 1131-3. Tous les mots-clés utilisés dans la CEI 1131-3 et donc dans Concept, sont listés en annexe C de la CEI 1131-3. Ces mots-clés répertoriés ne doivent être utilisés à aucune autre fin, p. ex. pas comme nom de variable, nom de section, nom d'instance, etc.

N

Node

Un node est une cellule de programmation dans un réseau LL984. Une cellule/un node comprend une matrice 7x11, c.-à-d. 7 lignes de 11 éléments.

Nom d'étape

Le nom d'étape sert à la désignation unique d'une étape dans une unité d'organisation de programme. Le nom d'étape est créé automatiquement, mais peut être édité. Il doit être unique dans toute l'unité d'organisation de programme, sinon un message d'erreur apparaît.

Le nom d'étape créé automatiquement a toujours la structure suivante : S_n_m

S = Etape

n = Numéro de la section (numéro courant)

m = Numéro de l'étape dans la section (numéro courant)

Nom d'instance

Un identificateur, associé à une instance spécifique de bloc fonction.. Le nom d'instance sert au repérage sans univoque d'un bloc fonction au sein d'une unité d'organisation de programme. Le nom d'instance est créé automatiquement, mais peut être édité. Le nom d'instance doit être unique dans toute l'unité d'organisation de programme, la distinction Majuscule/Minuscule n'est pas faite. Si le nom saisi existe déjà, vous en êtes averti et vous devez choisir un autre nom. Le nom d'instance doit satisfaire aux conventions de noms CEI, sinon un message d'erreur apparaît. Le nom d'instance créé automatiquement a toujours la structure suivante : FBI_n_m

FBI = Instance de bloc fonction

n = Numéro de la section (numéro courant)

m = Numéro de l'objet FFB dans la section (numéro courant)

Numéro d'identification

Le numéro d'identification sert à caractériser de manière unique une fonction dans un programme ou DFB. Le numéro d'identification ne peut être édité et est attribué automatiquement. Il a toujours la structure : .n.m

n = Numéro de la section (numéro courant)

m = Numéro de l'objet FFB dans la section (numéro courant)

O

Opérande

Un opérande est un libellé, une variable, un appel de fonction ou une expression.

Opérateur

Un opérateur est un symbole d'une opération arithmétique ou booléenne à exécuter.

P

Paramètre de sortie (Sortie)

Un paramètre avec lequel est (sont) retourné(s) le(s) résultat(s) de l'évaluation d'un FFB.

Paramètre d'entrée (Entrée)

Transmet lors de l'appel d'un FFB l'argument s'y rapportant.

Paramètre réel

Paramètre d'entrée/sortie actuellement attribué.

Paramètres formels

Paramètres d'entrée/sortie, utilisés au sein de la logique d'un FFB et sortant du FFB en entrées ou en sorties.

Paysage

Le format paysage signifie que la page, au regard du texte imprimé, est plus large que haute.

PC

Le matériel et le logiciel gérant (supportant) la programmation, l'élaboration, le test, la mise en service et la recherche de défauts dans les applications API ainsi que dans les applications système décentralisées, afin de rendre possible la documentation et l'archivage des sources. Le cas échéant, le PC peut également être utilisé pour la visualisation du procédé.

Portrait

Portrait signifie que la page, au regard du texte imprimé, est plus haute que large.

Presse-papiers

Le presse-papiers est une mémoire temporaire pour les objets coupés ou copiés. Ces objets peuvent être collés dans des sections. A chaque nouveau "couper" ou "copier", l'ancien contenu du presse-papiers est écrasé.

Processeur de communication

Le processeur de communication traite les passages de jeton et le flux de données entre le réseau Modbus Plus et la logique utilisateur de l'API.

Programmation de la redondance d'UC (Hot Standby)

Un système redondant est constitué de deux API configurés de manière identique qui communiquent entre eux à l'aide de processeurs redondants. En cas de panne de l'API primaire, l'API secondaire prend le contrôle de l'automatisme. Dans les conditions normales, l'API secondaire n'effectue aucune fonction de commande mais il vérifie les informations d'état afin de déceler les erreurs.

Programme

La plus haute unité d'organisation de programme. Un programme est chargé en entier sur un seul API.

Projet

Appellation générale du niveau le plus élevé d'une arborescence logicielle, qui définit le nom de projet supérieur d'une application d'API. Après avoir défini le nom du projet, vous pouvez sauvegarder votre configuration système et votre programme de commande sous ce nom. Toutes les données apparaissant lors de la création de la configuration et du programme font partie de ce projet supérieur pour cette tâche spéciale d'automatisation.

Désignation générale du jeu complet d'informations de programmation et de configuration dans la base de données de projet, laquelle représente le code source décrivant l'automatisation d'une installation.

R**REAL**

REAL correspond au type de données "nombre à virgule flottante". L'entrée se fait en libellé réel ou en libellé réel avec exposant. La longueur des éléments de données est de 32 bits. Plage des valeurs des variables de ce type de données : +/- 3.402823E+38.

NOTE : En fonction du type de processeur mathématique de l'UC, différentes zones de cette plage de valeurs permise ne peuvent pas être affichées. Cela s'applique aux valeurs tendant vers ZERO et aux valeurs tendant vers l'INFINI. Dans ces cas, une valeur NAN (**N**ot **A** **N**umber) ou INF (**I**N**F**inite (infini)) est affichée en mode Animation.

Référence

Toute adresse directe est une référence commençant par un code indiquant s'il s'agit d'une entrée ou d'une sortie et s'il s'agit d'un bit ou d'un mot. Les références commençant par le chiffre 6 représentent des registres de la mémoire étendue de la mémoire d'état.

Plage 0x = bits internes/de sortie

Plage 1x = bits d'entrée

Plage 3x = mots d'entrée

Plage 4x = mots internes/de sortie

Plage 6x = registres dans la mémoire étendue

NOTE : Le x suivant immédiatement le premier chiffre de chaque type de référence représente un emplacement mémoire à cinq chiffres dans la mémoire de données utilisateur, p.ex. la référence 400201 signifie un mot de sortie/mot interne de 16 bits à l'adresse 201 de la mémoire d'état.

Registres dans la mémoire étendue (référence 6x)

Les références 6x sont des mots indicateurs dans la mémoire étendue de l'API. Ils ne peuvent être utilisés que pour les programmes utilisateur LL984 et seulement sur les UC CPU 213 04 ou CPU 424 02.

Représentation directe

Une méthode pour représenter une variable dans un programme d'API, à partir de laquelle peut être déterminée directement une correspondance avec un emplacement logique, et indirectement avec l'emplacement physique.

Réseau

Un réseau est une connexion commune d'appareils sur une voie de données commune qui communiquent entre eux à l'aide d'un protocole commun.

Réseau décentralisé (DIO)

Une programmation décentralisée dans le réseau Modbus Plus permet une performance maximale de l'échange de données et n'a aucune exigence particulière sur les liaisons. La programmation d'un réseau décentralisé est simple. La configuration du réseau ne nécessite pas de logique de schéma à contacts supplémentaire. Toutes les conditions du transfert de données sont remplies en renseignant les paramètres correspondants du processeur de communication.

RIO (E/S décentralisée)

L'E/S décentralisée indique un emplacement physique des appareils E/S à commande par point par rapport au processeur qui les gère. Les entrées/sorties décentralisées sont reliées avec l'appareil de commande via un câble de communication.

S**Saut**

Élément du langage SFC. Les sauts sont utilisés pour éviter des zones de la séquence.

Schéma à contacts (LD)

Le schéma à contacts est un langage de programmation graphique conforme à la CEI1131, dont l'aspect visuel suit les "échelons" d'un schéma à relaying.

Schéma à contacts 984 (LL)

Comme leur nom l'indique, les schémas à contacts comportent des contacts. Contrairement à un schéma électrique, les électrotechniciens se servent d'un schéma à contacts pour dessiner un circuit (à l'aide de symboles électriques). Celui-ci doit montrer l'évolution d'événements, et non les fils en présence qui relient les différentes parties entre elles. Une interface de schéma à contacts permet de réaliser une interface utilisateur traditionnelle pour commander les actions des constituants d'automatisme, afin que les électrotechniciens ne soient pas obligés d'apprendre un langage de programmation avec lequel ils ne seraient pas à l'aise.

La construction d'un schéma à contacts effectif permet de relier des éléments électriques de manière à créer une sortie de commande. Celle-ci dépend d'un flux d'énergie logique passant par les objets électriques utilisés, lesquels représentent la condition préalable nécessaire d'un appareil électrique physique.

Sous une forme simple, l'interface utilisateur est un écran vidéo élaboré par l'application de programmation d'API, organisant un quadrillage vertical et horizontal dans lequel sont rangés des objets de programmation. Le schéma reçoit du courant par le côté gauche du quadrillage, et par connexion à des objets activés, le courant circule de gauche à droite.

Section

Une section peut par exemple être utilisée pour décrire le principe de fonctionnement d'une unité technologique telle qu'un moteur.

Un programme ou un DFB est constitué d'une ou de plusieurs sections. Les sections peuvent être programmées à l'aide des langages de programmation CEI FBD et SFC. Au sein d'une même section, seul un des langages de programmation mentionnés peut être utilisé.

Dans Concept, chaque section a sa propre fenêtre de document. Cependant, pour des raisons de clarté, il est conseillé de subdiviser une grande section en plusieurs petites. La barre de défilement sert à se déplacer au sein d'une section.

Station d'E/S DCP

A l'aide d'un processeur de contrôle distribué (D908), vous pouvez configurer un réseau décentralisé piloté par un API. Lorsque l'on utilise un D908 avec API décentralisé, l'API pilote considère l'API décentralisé comme une station d'E/S décentralisée. Le D908 et l'API décentralisé communiquent par le bus système, ce qui permet une grande performance pour un effet minimal sur le temps de cycle. L'échange de données entre le D908 et l'API pilote s'effectue par le bus d'E/S décentralisé à 1,5 Mégabit par seconde. Un API pilote peut gérer jusqu'à 31 processeurs D908 (adresse 2-32).

SY/MAX

Dans les automates Quantum, Concept gère la mise à disposition des modules d'E/S SY/MAX sur l'affectation des E/S pour la commande RIO par l'API Quantum. Le châssis distant SY/MAX dispose d'une carte d'E/S distante à l'emplacement 1, laquelle communique par un système d'E/S Modicon S908 R. Les modules d'E/S SY/MAX vous sont listés pour la sélection et la prise en compte dans l'affectation des E/S de la configuration Concept.

Symbole (icône)

Représentation graphique de différents objets sous Windows, p. ex. lecteurs, programmes utilisateur et fenêtre de document.

T

Tas CEI

Le tas CEI comprend la mémoire du programme CEI et les données globales.

TIME

TIME est le type de données "durée". L'entrée se fait sous forme de libellé de durée. La longueur des éléments de données est de 32 bits. La plage de valeurs des variables de ce type de données va de 0 à $2^{\text{exp}(32)}-1$. L'unité du type de données TIME est 1 ms.

Transition

La condition par laquelle la commande d'une ou de plusieurs étapes précédentes passe à une ou plusieurs étapes suivantes le long d'une liaison.

Type de bloc fonction

Un élément de langage constitué de : 1. la définition d'une structure de données, subdivisée en variables d'entrée, de sortie et internes ; 2. un jeu d'opérations exécutées avec les éléments de la structure de données, lorsqu'une instance du type de bloc fonction est appelée. Ce jeu d'opérations peut être formulé soit dans l'un des langages CEI (type DFB) ou en "C" (type EFB). Un type de bloc fonction peut être instancié (appelé) plusieurs fois.

Type de données dérivé

Les types de données dérivés sont des types de données qui ont été dérivés des types de données élémentaires et/ou d'autres types de données dérivés. La définition des types de données dérivés s'effectue dans l'éditeur de type de données de Concept.

On fait la distinction entre les types de données globaux et les types de données locaux.

Type de données générique

Un type de données représentant plusieurs autres types de données.

Types de données

La vue d'ensemble montre la hiérarchie des types de données et comment ils sont utilisés aux entrées et sorties des fonctions et blocs fonction. Les types de données génériques sont caractérisés par le préfixe "ANY".

- ANY_ELEM
 - ANY_NUM
 - ANY_REAL (REAL)
 - ANY_INT (DINT, INT, UDINT, UINT)
 - ANY_BIT (BOOL, BYTE, WORD)
 - TIME
- Types de données système (Extension CEI)
- Dérivé (des types de données 'ANY')

Types de données dérivés globaux

Les types de données dérivés globaux sont disponibles dans tout projet Concept et sont enregistrés dans le répertoire DFB directement situé sous le répertoire Concept.

Types de données dérivés locaux

Les types de données dérivés locaux ne sont disponibles que dans un seul projet Concept et ses DFB locaux et sont enregistrés dans le répertoire DFB sous le répertoire de projet.

U

UDEFB

Fonctions/Blocs fonction élémentaires défini(e)s par l'utilisateur

Fonctions ou blocs fonction créés en langage de programmation C et que Concept met à votre disposition dans des bibliothèques.

UDINT

UDINT représente le type de données "entier double non signé (unsigned double integer)". L'entrée s'effectue en libellé entier, libellé en base 2, libellé en base 8 ou libellé en base 16. La longueur des éléments de données est de 32 bits. La plage de valeurs des variables de ce type de données va de 0 à $2^{\text{exp}(32)}-1$.

UINT

UINT représente le type de données "entier non signé (unsigned integer)". L'entrée s'effectue en libellé entier, libellé en base 2, libellé en base 8 ou libellé en base 16. La longueur des éléments de données est de 16 bits. La plage des valeurs des variables de ce type de données va de 0 à $2^{\text{exp}(16)}-1$.

Unité d'organisation de programme

Une fonction, un bloc fonction ou un programme. Ce terme peut se rapporter à un type ou à une instance.

V

Valeur initiale

La valeur affectée à une variable lors du lancement du programme. L'affectation de la valeur s'effectue sous forme d'un libellé.

Variable localisée

Une adresse de mémoire d'état (adresses de références 0x, 1x, 3x, 4x) est affectée aux variables localisées. La valeur de ces variables est enregistrée dans la mémoire d'état et peut être modifiée en ligne au moyen de l'éditeur de données de référence. Ces variables peuvent être adressées avec leur nom symbolique ou avec leur adresse de référence.

Toutes les entrées et les sorties de l'API sont reliées à la mémoire d'état. L'accès du programme aux signaux des périphériques connectés à l'API ne se fait que via des variables localisées. Les accès de l'extérieur via les interfaces Modbus ou Modbus Plus de l'API, p. ex. des systèmes de visualisation, sont également possibles via des variables localisées.

Variable non localisée

Aucune adresse de mémoire d'état n'est affectée aux variables non localisées. Elles n'occupent donc pas non plus d'adresse de mémoire d'état. La valeur de ces variables est enregistrée dans le système et peut être modifiée en ligne au moyen de l'éditeur de données de référence. Ces variables ne sont adressées que par leur nom symbolique.

Les signaux ne disposant pas d'accès à la périphérie, p. ex. résultats intermédiaires, repères systèmes, etc., doivent être de préférence déclarés comme variable non localisée.

Variables

Les variables servent à l'échange de données au sein de sections, entre plusieurs sections et entre le programme et l'API.

Les variables consistent au moins en un nom de variable et un type de données.

Si une adresse directe (référence) est affectée à une variable, on parle alors de variable localisée. Si aucune adresse directe n'est affectée à une variable, on parle alors de variable non localisée. Si un type de données dérivé est affecté à une variable, on parle alors d'une variable multi-éléments.

Il existe en outre des constantes et des libellés.

Variables de tableau

Variables auxquelles sont affectées un type de données dérivé défini à l'aide du mot clé ARRAY (tableau). Un tableau est un ensemble d'éléments de données appartenant au même type.

Variables multi-éléments

Variables, auxquelles est affecté un type de données dérivé défini avec STRUCT ou ARRAY.

On fait ici la distinction entre variables de tableau et variables structurées.

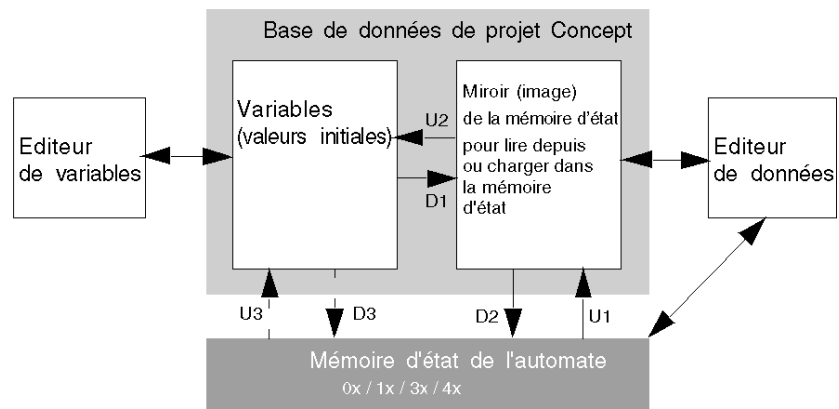
Variables structurées

Variables auxquelles est affecté un type de données dérivé défini avec STRUCT (structure).

Une structure est un ensemble d'éléments de données avec en général différents types de données (types de données élémentaires et/ou types de données dérivés).

Vue d'ensemble de la mémoire d'état lors de la lecture et du chargement

Vue d'ensemble :

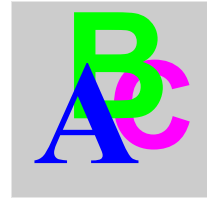


W

WORD

WORD correspond au type de données "Cordon de bits 16". L'entrée peut se faire en libellé en base 2, libellé en base 8 ou libellé en base 16. La longueur des éléments de données est de 16 bits. Il n'est pas possible d'affecter une plage de valeurs numériques à ce type de données.

Index



A

Affectation protégée contre les interruptions, 75

B

Bloc fonction

Paramétrage, 13, 13

Blocage de sections d'interruption particulières, 51

Blocage de toutes les sections d'interruption, 63

C

Commande SFC, 131

Commande SFC étendue, 145

Compteur libre, 23

D

Déblocage d'une section d'interruption particulière, 47

Déblocage de toutes les sections d'interruption, 59

Découpages du système, 145

DIOSTAT, 19

E

Ecriture du registre d'ordre Hot Standby, 43

Ecriture et lecture des deux registres de transfert reverse, 115

Etat de fonction API, 83

Etat de fonction module (DIO), 19

Etat de fonction module pour M1, 71

Etat de fonction modules (RIO), 119

Etat de section d'interruption, 55

Etat de système, 149

Evénement connecté, 79

F

Fonction

Paramétrage, 13, 13

FREERUN, 23

G

GET_IEC_INF, 27

GET_TOD, 31

H

HSBY

GET_TOD, 31

HSBY_RD, 35

HSBY_ST, 39

HSBY_WR, 43

REV_XFER, 115

SET_TOD, 127

HSBY_RD, 35
HSBY_ST, 39
HSBY_WR, 43

I

I_LOCK, 63
I_MOVE, 75
I_UNLOCK, 59
ISECT_OFF, 51
ISECT_ON, 47
ISECT_STAT, 55

L

Lecture de l'horloge du matériel (Time Of Day), 31
Lecture des indicateurs de statut CEI, 27
Lecture du registre d'état Hot Standby, 39
Lecture du registre d'ordre Hot Standby, 35
LOOPBACK, 67

M

M1HEALTH, 71

N

Nom/version de projet, 107

O

ONLEVT, 79

P

Paramétrage, 13, 13
Période d'échantillonnage, 123
PLCSTAT, 83
PRJ_VERS, 107

R

Réglage de l'horloge du matériel (Time Of Day), 127

Remise à zéro des indicateurs de statut CEI, 111

RES_IEC_INF, 111

Resaut, 67

REV_XFER, 115

RIOSTAT, 119

S

SAMPLETM, 123

Sauter la section restante, 141

SET_TOD, 127

SFCCNTRL, 131

SKP_RST_SCT_FALSE, 141

Special

SKP_RST_SCT_FALSE, 141

Specials

FREERUN, 23

LOOPBACK, 67

ONLEVT, 79

SAMPLETM, 123

SFCCNTRL, 131

XSFCNTRL, 153

SYSCLOCK, 145

SYSSTATE, 149

Système

DIOSTAT, 19, 19

FREERUN, 23

GET_IEC_INF, 27

GET_TOD, 31

HSBY_RD, 35

HSBY_ST, 39

HSBY_WR, 43

I_LOCK, 63

I_MOVE, 75

I_UNLOCK, 59

ISECT_OFF, 51

ISECT_ON, 47

ISECT_STAT, 55

LOOPBACK, 67

M1HEALTH, 71, 71

ONLEVT, 79

PLCSTAT, 83, 83

SYSTEME

PRJ_VERS, 107

Système

RES_IEC_INF, 111
REV_XFER, 115
RIOSTAT, 119, 119
SAMPLETM, 123
SET_TOD, 127
SFCCNTRL, 131
SKP_RST_SCT_FALSE, 141
SYSCLOCK, 145, 145
SYSSTATE, 149, 149
XSFCNTRL, 153

X

XSFCNTRL, 153

