

Conformité CEI 61131-3

Utilisation de cette annexe

Pour les informations suivantes	Voir page
Système d'exploitation	B-2
Définitions des données	B-2
Langages de programmation	B-3
Jeu d'instructions	B-4
Portabilité des programmes CEI 61131-3	B-4
Tableaux de conformité CEI	B-5

Introduction

La Commission électrotechnique internationale (CEI) a développé une série de spécifications pour les automates programmables. Ces spécifications ont été créées dans le but de promouvoir une unification internationale des équipements et des langages de programmation utilisés dans l'industrie de la commande. Ces spécifications sont à la base des automates Logix5000 et du logiciel de programmation RSLogix 5000.

Les spécifications de la CEI sur les automates programmables sont regroupées en cinq parties, chacune traitant d'un aspect différent du système de commande :

- Partie 1 : Informations générales
- Partie 2 : Spécifications et essais des équipements
- Partie 3 : Langages de programmation
- Partie 4 : Recommandations à l'usage de l'utilisateur
- Partie 5 : Communications

Dans son ensemble, l'industrie de la commande s'est concentrée sur la partie 3 (CEI 61131-3), Langages de programmation, parce que cette partie est la base de l'application des autres normes et qu'elle apporte à l'utilisateur les avantages les plus significatifs par la réduction des coûts de formation. Pour ces raisons, nous ne traitons que cette partie de la norme CEI 61131-3 ici.

Les spécifications de la norme CEI 61131-3 sur les langages de programmation abordent de nombreux aspects des automates programmables, notamment l'exécution du système d'exploitation, les définitions des données, les langages de programmation et le jeu d'instructions. Les éléments de la norme CEI 61131-3 sont regroupés selon les impératifs de la spécification, des options ou des extensions. De cette façon, la norme CEI 61131-3 fournit un ensemble minimum de fonctionnalités qui peuvent être étendues pour répondre aux besoins des applications de l'utilisateur. Le côté négatif de ce système est que chaque fabricant de système de commande programmable peut utiliser différents composants de la spécification ou fournir différentes extensions.

Système d'exploitation

Le système d'exploitation multitâche préemptif des automates Logix5000 est conforme à la définition donnée par la norme CEI 61131-3. Dans la norme CEI 61131-3, le système d'exploitation des automates programmables peut contenir zéro tâche ou plus, pouvant exécuter un ou plusieurs programmes contenant chacun un ou plusieurs sous-programmes ou fonctions. Selon la norme CEI 61131-3, le nombre de chacun de ces éléments dépend de la mise en application. L'automate Logix5000 fournit plusieurs tâches, chacune contenant plusieurs programmes et un nombre illimité de fonctions ou de sous-programmes.

La norme CEI 61131-3 fournit une option pour la création de différentes classifications pour l'exécution des tâches. Ces tâches peuvent être configurées comme continues, périodiques et basées sur événement. Une tâche continue n'a pas besoin d'être planifiée dans la mesure où elle utilise tout temps de traitement non utilisé lorsque les autres tâches ne sont pas exécutées. Les tâches périodiques sont programmées pour être exécutées selon une période de temps récurrent. La norme CEI 61131-3 ne précise pas de durée pour la configuration des tâches périodiques. Une tâche CEI 61131-3 basée sur événement est lancée au moment de la détection du front montant d'une entrée configurée. Les automates Logix5000 acceptent les tâches continues et périodiques. De plus, la période d'une tâche périodique est configurable avec une durée allant jusqu'à 1 milliseconde (ms).

Définitions des données

La norme CEI 61131-3 donne accès à la mémoire par la création de variables dénommées. Les noms CEI 61131-2 des variables sont constitués d'un minimum de six caractères (le logiciel de programmation RSLogix 5000 accepte un minimum de 1 caractère) commençant par un caractère de soulignement « _ » ou par un caractère alphabétique (A-Z), suivi par un ou plusieurs caractères constitués d'un caractère de soulignement « _ », d'une lettre (A-Z) ou d'un nombre (0-9). En option, les caractères alphabétiques en minuscule (a-z) sont acceptables s'il n'y a pas de différence majuscule/minuscule (A = a, B = b, C = c ...). Les automates Logix5000 sont en totale conformité avec cette définition, ils acceptent l'option des minuscules et permettent des noms ayant jusqu'à 40 caractères.

Dans la norme CEI 61131-3, les variables de données peuvent être définies de façon à être accessibles à tous les programmes d'une ressource ou d'un automate, ou de façon qu'un accès limité soit offert aux seules fonctions ou sous-programmes d'un programme donné. Pour passer des données entre plusieurs ressources ou automates, les chemins d'accès peuvent être configurés de façon à définir l'emplacement des données dans un système. Les automates Logix5000 sont conformes à la norme parce qu'ils fournissent des données internes aux programmes ou aux automates, et permettent la configuration de chemins d'accès utilisant des données produites/consommées.

L'interprétation de la mémoire d'une variable dans la norme CEI 61131-3 est définie par l'utilisation d'un type de données élémentaires ou d'un type de données dérivées en option créé à partir d'un groupe de types de données multiples. Les automates Logix5000 acceptent l'utilisation des types de données élémentaires BOOL (1 bit), SINT (entier de 8 bits), INT (entier de 16 bits), DINT (entier de 32 bits) et REAL (IEEE nombre à virgule flottante). En outre, les types de données dérivées en option sont acceptés à travers la création de structures et tableaux définis par l'utilisateur.

Langages de programmation

La norme CEI 61131-3 définit cinq (5) langages de programmation différents et un jeu d'éléments communs. Tous les langages sont définis comme optionnels mais au moins un doit être accepté pour être conforme à la norme. Les composants de langage de programmation CEI 61131-3 sont définis ainsi :

- Eléments de langage commun ;
- Eléments graphiques communs ;
- Eléments du langage de liste d'instructions (IL) ;
- Eléments du langage en texte structuré (ST) ;
- Eléments du langage en diagramme à relais (LD) ;
- Eléments du langage de graphe de fonctionnement séquentiel (SFC) ;
- Eléments du langage de bloc fonctionnel (FBD) .

Les automates Logix5000 et le logiciel RSLogix 5000 acceptent les éléments de langage commun et les options des langages en texte structuré, diagramme à relais, graphe de fonctionnement séquentiel et bloc fonctionnel. En outre, l'environnement utilise un format d'importation/exportation ASCII basé sur le langage en texte structuré. Les fonctions d'échange de jeu d'instructions et de fichier programme sont abordées en détails dans les sections suivantes.

Jeu d'instructions

Le jeu d'instructions indiqué dans la norme CEI 61131-3 est entièrement optionnel. La norme indique un jeu d'instructions limité qui, s'il est appliqué, doit être conforme à l'exécution indiquée et à la représentation visuelle. Cependant, la norme CEI 61131-3 ne limite pas les instructions à celles qu'elle indique. Chaque fabricant d'automates est libre d'intégrer des fonctionnalités supplémentaires sous la forme d'instructions en plus de celles données par la norme. Des exemples de telles instructions étendues sont celles nécessaires pour les diagnostics, la commande de boucle PID, la commande de mouvement et le traitement de fichiers de données. Les instructions étendues n'étant pas définies par la norme CEI 61131-3, il n'y a aucune garantie sur leur comptabilité d'application sur des automates de fabricants différents. Ainsi, l'utilisation de ces instructions peut interdire le transfert de logique entre fabricants.

Les automates Logix5000 et le logiciel RSLogix 5000 fournissent une suite d'instructions qui sont exécutées selon la norme CEI 61131-3. La représentation physique de ces instructions leur garde le même aspect que sur les systèmes existants, de façon à réduire les coûts de formation associés. En plus des instructions conformes à la norme CEI 61131-3, une gamme complète d'instructions venant de produits existants a été introduite, de sorte qu'aucune fonctionnalité n'est perdue.

Portabilité des programmes CEI 61131-3

L'un des objectifs des utilisateurs qui créent des programmes dans un environnement conforme CEI 61131-3 est le transfert ou la portabilité des programmes entre automates développés par différents fabricants. C'est une faiblesse de la norme CEI 61131-3 dans la mesure où elle ne définit aucun format d'échange de fichier. En d'autres termes, un programme créé dans l'environnement d'un fabricant demandera des manipulations pour être transféré sur le système d'un autre fabricant.

Pour minimiser l'effort nécessaire à la portabilité, le logiciel de programmation RSLogix 5000, pour les automates, comporte un utilitaire complet d'exportation/importation ASCII. De plus, le format de fichier utilisé par cet outil est basé sur une définition hybride du langage en texte structuré de la norme CEI 61131-3. Le système d'exploitation et les définitions des données de l'automate suivent les formats CEI 61131-3 appropriés. Des extensions ont été créées pour convertir la logique de diagramme à relais en texte ASCII, ceci n'étant pas défini par la norme CEI 61131-3.

Pour de plus amples informations sur l'utilitaire d'exportation et d'importation ASCII du logiciel de programmation RSLogix 5000, voir *Logix5000 Controllers Import/Export Reference Manual* (publication 1756-RM084).

Tableaux de conformité CEI

Les automates Logix5000 et le logiciel RSLogix 5000 sont conformes aux directives de la norme CEI 61131-3 pour les fonctions de langage suivantes :

Numéro du tableau ⁽¹⁾	Numéro de la fonction	Description de la fonction	Extensions et notes d'application
1	2	Lettres minuscules	Aucune
1	3a	Symbole de numéro (#)	Utilisé pour désigner le type de données à valeur immédiate
1	4a	Symbole dollar (\$)	Utilisé pour décrire et comme caractère de commande des chaînes
1	6a	Séparateurs d'indice ([])	Indices de tableau
2	1	Identificateurs utilisant des majuscules et des chiffres	Noms de tâche, de programme, de sous-programme, de structure et de point
2	2	Identificateurs utilisant des majuscules, des chiffres et des caractères de soulignement intégrés	Noms de tâche, de programme, de sous-programme, de structure et de point
2	3	Identificateurs utilisant des minuscules, des chiffres et des caractères de soulignement intégrés	Noms de tâche, de programme, de sous-programme, de structure et de point
3	1	Commentaires	Commentaires ST, également /* Commentaire */ et // Fin d'une ligne de commentaires
4	1	Entier au format littéral	12, 0, -12
4	2	Réel au format littéral	12,5, -12,5
4	3	Réel au format littéral avec exposant	-1,34E-12, 1,234E6
4	4	Base 2 au format littéral	2#0101_0101
4	5	Base 8 au format littéral	8#377
4	6	Base 16 au format littéral	16#FF00
4	7	Zéro et un booléens	0, 1
5	1A	Chaîne vide ""	Descriptions et éditeur de chaîne
5	1b	Chaîne de longueur une contenant un caractère 'A'	Descriptions et éditeur de chaîne
5	1C	Chaîne de longueur une contenant un espace ' '	Descriptions et éditeur de chaîne
5	1D	Chaîne de longueur une contenant un caractère de guillemet simple '\$'	Descriptions et éditeur de chaîne
5	1E	Chaîne de longueur une contenant un caractère de guillemet double '"'	Descriptions et éditeur de chaîne
5	1F	Chaîne de longueur deux contenant des caractères CR (retour chariot) LF (changement de ligne)	Descriptions et éditeur de chaîne
5	1G	Chaîne de longueur une contenant le caractère LF '\$OA'	Descriptions et éditeur de chaîne
5	1H	Chaîne de longueur 5 qui est imprimé comme "\$1.00" utilisant '\$\$1.00'	Descriptions et éditeur de chaîne
5	1I	Chaînes équivalentes de longueur deux 'AE' et '\$C4\$CB'	Descriptions et éditeur de chaîne
6	2	Symbole dollar '\$\$' dans une chaîne	Descriptions et éditeur de chaîne
6	3	Guillemet simple '\$' dans une chaîne	Descriptions et éditeur de chaîne

Numéro du tableau ⁽¹⁾	Numéro de la fonction	Description de la fonction	Extensions et notes d'application
6	4	Saut de ligne '\$L' ou '\$l' dans une chaîne	Descriptions et éditeur de chaîne
6	5	Nouvelle ligne '\$N' ou '\$n' dans une chaîne	Descriptions et éditeur de chaîne
6	6	Saut de page '\$P' ou '\$p' dans une chaîne	Descriptions et éditeur de chaîne
6	7	Retour chariot '\$R' ou '\$r' dans une chaîne	Descriptions et éditeur de chaîne
6	8	Tabulation '\$T' ou '\$t' dans une chaîne	Descriptions et éditeur de chaîne
6	9	Guillemet double '\$" dans une chaîne	Descriptions et éditeur de chaîne
10	1	Type de données BOOL	Définition de la variable de point
10	2	Type de données SINT	Définition de la variable de point
10	3	Type de données INT	Définition de la variable de point
10	4	Type de données DINT	Définition de la variable de point
10	10	Type de données REAL	Définition de la variable de point
10	12	Heure	Définition de la variable de point, structure TIMER
10	16	Type de données STRING	8 bits
11	1	Hierarchie des types de données	Aucune
12	1	Dérivation directe à partir des types élémentaires	Structures des types de données définis par l'utilisateur
12	4	Types de données de tableau	Définition de la variable de point
12	5	Types de données structurées	Structures des types de données définis par l'utilisateur
13	1	Valeur initiale BOOL, SINT, INT, DINT, de 0	Définition de la variable de point
13	4	Valeur initiale REAL, LREAL de 0.0	Définition de la variable de point
13	5	Valeur initiale de l'heure T#0s	Définition de la variable de point, instruction de réinitialisation (RES)
13	9	Chaîne vide ""	Descriptions et chaînes
14	1	Initialisation des types dérivés directement	Importation/exportation
14	4	Initialisation des types de données tableau	Importation/exportation
14	5	Initialisation des éléments de type structuré	Importation/exportation
14	6	Initialisation des types de données structurées dérivés	Importation/exportation
19a	2a	Appel textuel, informel	Disponible en ST
20	1	Utilisation de EN et ENO	Fonction présente dans la logique à relais mais non étiquetée. Disponible en FBD
20	2	Utilisation sans EN et ENO	Disponible en FBD
20	3	Utilisation avec EN et sans ENO	Disponible en FBD
20	4	Utilisation sans EN et avec ENO	Disponible en FBD
21	1	Fonctions surchargées ADD(INT, DINT) ou ADD(DINT, REAL)	Tous les types surchargés reconnus sont expliqués avec chaque instruction

Numéro du tableau⁽¹⁾	Numéro de la fonction	Description de la fonction	Extensions et notes d'application
22	1	Fonction de conversion _TO_	Instructions RAD, DEG, conversion radians/valeurs décimales. Conversion numérique de chaîne STOD, STOR, RTOS, DTOS. Les autres conversions ne sont pas nécessaires du fait de la surcharge des instructions.
22	2	Fonction de conversion par troncation	Instruction TRN en LD et fonction TRUNC en ST
22	3	Conversion DCB en ENT	Instruction FRD en LD
22	4	Conversion ENT en DCB	Instruction TOD en LD
23	1	Valeur absolue	Instruction ABS
23	2	Racine carrée	Instruction SQR en LD et fonctions FBD et SQRT en ST
23	3	Logarithme naturel	Instruction LN
23	4	Logarithme décimal	Instruction LOG
23	6	Sinus en radians	Instruction / fonction SIN
23	7	Cosinus en radians	Instruction / fonction COS
23	8	Tangente en radians	Instruction / fonction TAN
23	9	Sinus d'arc principal	Instruction ASN en LD et fonctions FBD et ASIN en ST
23	10	Cosinus d'arc principal	Instruction ACS en LD et fonctions FBD et ACOS en ST
23	11	Tangente d'arc principal	Instruction ATN en LD et fonctions FBD et ATAN en ST
24	12	Addition arithmétique	Instruction ADD en LD et FBD, et + en ST
24	13	Multiplication arithmétique	Instruction MUL en LD et FBD, et * en ST
24	14	Soustraction arithmétique	Instruction SUB en LD et FBD, et - en ST
24	15	Division arithmétique	Instruction DIV en LD et FBD, et / en ST
24	16	Modulo	Instruction MOD en LD et ST
24	17	Exponentiel	Instruction XPY en LD et FBD, et ** en ST
24	18	Déplacement de valeur	Instruction MOV en LD et := en ST
25	1	Décalage binaire à gauche	Fonctionnalité contenue dans l'instruction BSL en LD pour un décalage de 1
25	2	Décalage binaire à droite	Fonctionnalité contenue dans l'instruction BSR en LD pour un décalage de 1
25	3	Rotation binaire à gauche	Fonctionnalité contenue dans l'instruction BSL en LD pour un décalage de 1
25	4	Rotation binaire à droite	Fonctionnalité contenue dans l'instruction BSR en LD pour un décalage de 1
26	5	ET	Instruction BAND en FBD et opération "&" en ST
26	6	OU	Instruction BOR en FBD
26	7	XOR	Instruction BXOR en FBD
26	8	NOT	Instruction BNOT en FBD
27	1	Sélection	Instruction SEL en FBD

Numéro du tableau ⁽¹⁾	Numéro de la fonction	Description de la fonction	Extensions et notes d'application
27	2a	Sélection maximum MAX	Fonctionnalité contenue dans l'instruction ESEL en FBD et en ST
27	2b	Sélection minimum MIN	Fonctionnalité contenue dans l'instruction ESEL en FBD et en ST
27	3	Limite haute/basse LIMIT	Instruction HLL en FBD et en ST
27	4	Multiplexeur MUX	Instruction MUX en FBD
28	5	Comparaison Supérieur à	Instruction GRT en LD et FBD, et > en ST
28	6	Comparaison Supérieur ou égal à	Instruction GRE en LD et FBD, et >= en ST
28	7	Comparaison Egal à	Instruction EQU en LD et FBD, et = en ST
28	8	Comparaison Inférieur à	Instruction LES en LD et FBD, et < en ST
28	9	Comparaison inférieur ou égal à	Instruction LEQ en LD et FBD, et <= en ST
28	10	Comparaison Différent de	Instruction NEQ en LD et FBD, et <> en ST
29	1	Longueur de chaîne LEN	Contenue comme paramètre du type de données STRING
29	4	Copie dans une chaîne MID	Instruction MID en LD et en ST
29	5	Concaténation de chaînes CONCAT	Instruction CONCAT en LD et en ST
29	6	Insertion dans une chaîne INSERT	Instruction INSERT en LD et en ST
29	7	Effacement dans une chaîne DELETE	Instruction DELETE en LD et en ST
29	9	Recherche de chaîne FIND	Instruction FIND en LD et en ST
32	1	Lecture d'entrée	FBD et ST
32	2	Écriture d'entrée	FBD et ST
32	3	Lecture de sortie	FBD et ST
32	4	Écriture de sortie	FBD et ST
34	1	Bascule activée prioritaire	Instruction SETD en FBD et en ST
34	2	Bascule mise à zéro prioritaire	Instruction RESD en FBD et en ST
35	1	Détecteur front montant	Instruction OSR en LD et instruction OSRI en FBD et en ST
35	2	Détecteur front descendant	Instruction OSF en LD et instruction OSFI en FBD et en ST
36	1b	Compteur incrémental	Fonctionnalité contenue dans les instructions CTU et RES en LD et dans l'instruction CTUD en FBD et en ST
37	2a	Temporisateur avec délai à l'enclenchement	Fonctionnalité contenue dans l'instruction TON en LD et dans l'instruction TONR en FBD et en ST
37	3a	Temporisateur avec délai au déclenchement	Fonctionnalité contenue dans l'instruction TOF en LD et dans l'instruction TOFR en FBD et en ST
38	2	Temporisation à l'enclenchement	Fonctionnalité contenue dans l'instruction TON en LD et dans l'instruction TONR en FBD et en ST
38	3	Temporisation au déclenchement	Fonctionnalité contenue dans l'instruction TOF en LD et dans l'instruction TOFR en FBD et en ST

Numéro du tableau⁽¹⁾	Numéro de la fonction	Description de la fonction	Extensions et notes d'application
40	1a	SFC - Etape	
40	1b	SFC - Etape initiale	
40	2a	SFC - Etape textuelle	Importation/exportation, le nom de l'étape est indiqué comme suit : « Opérande := nom_étape »
40	2b	SFC - Etape initiale textuelle	Importation/exportation, utilise le paramètre « InitialStep » ; le nom de l'étape est indiqué comme suit : « Opérande := nom_étape »
40	3a	SFC - Forme générale de l'indicateur d'étape	Point de retour d'une étape
40	4	Forme générale du temps écoulé pour une étape	Point de retour d'une étape
41	1	Transition utilisant du ST	
41	5	Forme textuelle d'une transition	Importation/exportation avec différents formats
41	7	Nom d'une transition	Point de retour d'une transition
41	7a	Transition activée par LD	Point de retour d'une transition
41	7b	Transition activée par FBD	Point de retour d'une transition
41	7d	Transition activée par ST	Point de retour d'une transition
42	1	Action booléenne	Point de retour d'une action
42	3s	Représentation textuelle d'une action	Importation/exportation
43	1	Association de l'action d'une l'étape	
43	2	Etape avec actions de concaténation	
43	3	Corps de l'étape textuelle	Importation/exportation avec différents formats
43	4	Champ du corps de l'action	ST incorporé
44	1	Qualificateur de bloc d'une action	
44	2	Nom de bloc d'une action	
44	3	Point indicateur d'une action	Etendre ceci pour accepter DINT, INT, SINT ou REAL en plus de BOOL
44	5	Action utilisant du ST	Accepte le ST incorporé et une instruction JSR vers un sous-programme ST
44	6	Action utilisant du LD	Utilise une instruction JSR vers un sous-programme LD
44	7	Action utilisant du FBD	Utilise une instruction JSR vers un sous-programme FBD
45	1	Aucun qualificateur d'action	Le qualificateur par défaut est N lorsque aucun n'est explicitement entré
45	2	Qualificateur d'action N - Non stocké	
45	3	Qualificateur d'action R - Réinitialisé	
45	4	Qualificateur d'action S - Activé / Stocké	
45	5	Qualificateur d'action L - Limité dans le temps	
45	6	Qualificateur d'action D - Différé	
45	7	Qualificateur d'action P - Impulsion	
45	8	Qualificateur d'action SD - Stocké et différé	
45	9	Qualificateur d'action DS - Différé et stocké	

Numéro du tableau ⁽¹⁾	Numéro de la fonction	Description de la fonction	Extensions et notes d'application
45	10	Qualificateur d'action SL - Stocké et différé	
45	11	Qualificateur d'action P1 - Impulsion front montant	
45	12	Qualificateur d'action P0 - Impulsion front descendant	
45a	1	Commande de l'action	
45a	2	Commande de l'action	
46	1	SFC - Séquence unique	
46	2a	SFC - Sélection de divergence de séquence	Utilise des connexions de ligne plutôt que des astérisques
46	2b	SFC - Sélection de divergence de séquence avec ordre d'exécution	
46	3	SFC - Sélection de convergence de séquence	
46	4a	SFC - Divergence de séquence simultanée	
46	4b	SFC - Convergence de séquence simultanée	
46	5a, b, c	SFC - Saut de séquence	
46	6a, b, c	SFC - Boucle de séquence	
46	7	SFC - Flèche de direction de boucle	Lorsqu'une liaison est masquée
47	1	Représentation graphique d'un SFC	
47	4	Représentation graphique d'un SFC	
48	1	SFC - Impératifs minimum de conformité d'étape	Voir les notes des tableaux individuels ci-dessus
48	2	SFC - Impératifs minimum de conformité de transition	Voir les notes des tableaux individuels ci-dessus
48	3	SFC - Impératifs minimum de conformité d'action	Voir les notes des tableaux individuels ci-dessus
48	4	SFC - Impératifs minimum de conformité du corps d'action	Voir les notes des tableaux individuels ci-dessus
48	5	SFC - Impératifs minimum de conformité du qualificateur d'action	Voir les notes des tableaux individuels ci-dessus
48	6	SFC - Impératifs minimum de conformité de branche	Voir les notes des tableaux individuels ci-dessus
48	7	SFC - Impératifs minimum de conformité de blocs de connexion	Voir les notes des tableaux individuels ci-dessus
55	1	ST - Mise entre parenthèses (expression)	
55	2	Evaluation de fonction ST	Utilisation de format informel pour l'appel de fonctions intégrées. Instruction JSR utilisée dans le langage ST pour appeler le code développé par l'utilisateur.
55	3	ST - Exposant **	
55	4	ST - Inversion -	
55	5	ST - Négation NOT	
55	6	ST - Multiplication *	
55	7	ST - Division /	

Numéro du tableau ⁽¹⁾	Numéro de la fonction	Description de la fonction	Extensions et notes d'application
55	8	ST - Modulo MOD	
55	9	ST - Addition +	
55	10	ST - Soustraction -	
55	11	ST - Comparaison <, >, <=, >=	
55	12	ST - Egalité =	
55	13	ST - Inégalité <>	
55	14	ST - ET booléen (&)	
55	15	ST - ET booléen	
55	16	ST - OU exclusif booléen XOR	
55	17	ST - OU booléen	
56	1	ST - Affectation :=	
56	2	ST - Appel de bloc fonctionnel	
56	3	ST - RETURN	RET() avec plusieurs paramètres
56	4	ST - IF / ELSIF / ELSE / END_IF	
56	5	ST - CASE OF / ELSE / END_CASE	
56	6	ST - FOR / END_FOR	
56	7	ST - WHILE DO / END_WHILE	
56	8	ST - REPEAT / UNTIL / END_REPEAT	
56	9	ST - EXIT	
56	10	ST - Instruction vide ;	
57	1, 2	Ligne horizontale	Editeur LD, éditeur FBD
57	3, 4	Ligne verticale	Editeur LD, éditeur FBD
57	5, 6	Connexion horizontale / verticale connexion	Editeur LD, éditeur FBD
57	7, 8	Croisements de lignes sans connexion	Editeur FBD
57	9, 10	Coins de connexion et sans connexion	Editeur LD, éditeur FBD
57	11, 12	Blocs avec connexions	Editeur LD, éditeur FBD
57	13,14	Connecteurs	Editeur FBD
58	2	Saut inconditionnel	Instruction JMP en LD
58	3	Cible de saut	Instruction LBL en LD
58	4	Saut conditionnel	Instruction JMP en LD
58	5	Retour conditionnel	Instruction RET en LD
58	8	Retour inconditionnel	Instruction RET en LD
59	1	Rail d'alimentation gauche	Editeur LD
59	2	Rail d'alimentation droit	Editeur LD
60	1	Lien horizontal	Editeur LD
60	2	Lien vertical	Editeur LD
61	1, 2	Contact normalement ouvert -- --	Instruction XIC en LD
61	3, 4	Contact normalement fermé -- / --	Instruction XIO en LD

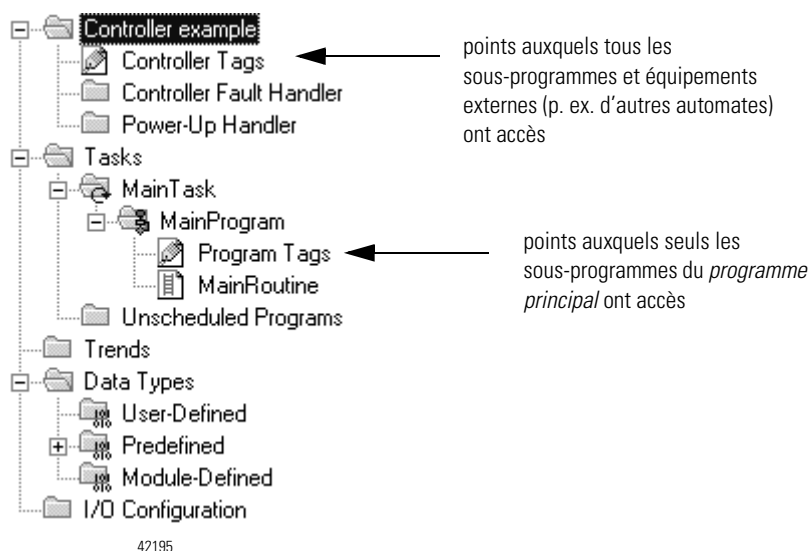
Numéro du tableau⁽¹⁾	Numéro de la fonction	Description de la fonction	Extensions et notes d'application
61	5, 6	Contact de détection de transition positive - P -	Instruction ONS en LD
62	1	Bobine --()--	Instruction OTE en LD
62	3	Activer (verrouiller) bobine	Fonctionnalité figurant dans l'instruction OTL en LD
62	4	Réinitialiser (déverrouiller) bobine	Fonctionnalité figurant dans l'instruction OTU en LD
62	8	Bobine de détection de transition positive	Instruction OSR en LD
62	9	Bobine de détection de transition négative	Instruction OSF en LD

⁽¹⁾ Les tableaux associés à des langages autres que le texte structuré, graphe de fonctionnement séquentiel, diagramme à relais et diagramme de blocs fonctionnels ont été ignorés.

A

accès

Définit le type d'accès et l'endroit où vous pouvez accéder à un jeu de points particulier. Lorsque vous créez un point, vous l'affectez comme point d'automate ou comme point de programme pour un programme particulier, comme décrit ci-dessous.



Vous pouvez avoir plusieurs points avec le même nom :

- Chaque point doit avoir une affectation différente. Par exemple, un des points peut être un point d'automate et les autres points peuvent être des points de programme pour différents programmes. Ou, chaque point peut être un point de programme pour des programmes différents.
- Dans un même programme, vous ne pouvez pas faire référence à un point d'automate si un point avec le même nom existe comme point de programme pour ce programme.

Voir *accès automate*, *accès programme*.

accès automate

Données accessibles de n'importe où dans l'automate. L'automate comporte une série de points auxquels les sous-programmes et les points d'alias de n'importe quel programme, ainsi que d'autres alias d'accès automate peuvent renvoyer. accès programme.

accès programme

Données uniquement accessibles à partir du programme en cours. Chaque programme possède une collection de points qui ne peuvent être référencés que par les sous-programmes et points d'alias dans ce programme. accès automate.