

Annexe B (informative) Liste des modifications et extensions majeures de la troisième édition	463
Bibliographie.....	464
Figure 1 – Modèle logiciel	247
Figure 2 – Modèle de communication	249
Figure 3 – Combinaison d'éléments de langage pour automate programmable	251
Figure 4 – Déclaration de conformité de l'Intégrateur (exemple).....	253
Figure 5 – Hiérarchie des types de données génériques	265
Figure 6 – Initialisation par des littéraux et des expressions constantes (règles).....	266
Figure 7 – Mots-clés pour une déclaration de variable (résumé)	282
Figure 8 – Utilisation de <code>VAR_GLOBAL</code> , <code>VAR_EXTERNAL</code> et <code>CONSTANT</code> (règles)	283
Figure 9 – Conditions associées à la valeur initiale d'une variable (règles)	289
Figure 10 – Représentation formelle et informelle d'appel (exemples).....	295
Figure 11 – Règles de conversion d'un type de données – implicite et/ou explicite (résumé).....	299
Figure 12 – Conversions de type implicites prises en charge	300
Figure 13 – Utilisation des paramètres d'entrée et de sortie de bloc fonctionnel (règles).....	343
Figure 14 – Utilisation des paramètres d'entrée et de sortie de bloc fonctionnel (illustration des règles)	344
Figure 15 – Blocs fonctionnels normalisés minuteur – diagrammes temporels (règles).....	351
Figure 16 – Présentation de la mise en œuvre d'héritage et d'interface.....	354
Figure 17 – Héritage de classes (illustration)	363
Figure 18 – Interface avec classes dérivées (illustration)	373
Figure 19 – Héritage d'interface et de classe (illustration)	378
Figure 20 – Bloc fonctionnel avec corps et méthodes facultatifs (illustration)	384
Figure 21 – Héritage de corps de bloc fonctionnel avec <code>SUPER()</code> (exemple)	386
Figure 22 – Bloc fonctionnel <code>ACTION_CONTROL</code> – Interface externe (résumé).....	400
Figure 23 – Corps de bloc fonctionnel <code>ACTION_CONTROL</code> (résumé)	402
Figure 24 – Contrôle d'action (exemple).....	404
Figure 25 – Evolution d'un SFC (règles).....	410
Figure 26 – Erreurs d'un SFC (exemple)	411
Figure 27 – Configuration (exemple)	413
Figure 28 – Déclaration de <code>CONFIGURATION</code> et de <code>RESOURCE</code> (exemple)	416
Figure 29 – Accessibilité à l'aide des espaces de noms (règles)	424
Figure 30 – Éléments textuels communs (résumé)	430
Tableau 1 – Jeu de caractères.....	254
Tableau 2 – Identificateurs.....	254
Tableau 3 – Commentaires	256
Tableau 4 – Pragma	256
Tableau 5 – Littéraux numériques	258
Tableau 6 – Littéraux de chaîne de caractères.....	259
Tableau 7 – Combinaisons de deux caractères dans les chaînes de caractères.....	260

Tableau 8 – Littéraux de durée	261
Tableau 9 – Littéraux de date et heure	261
Tableau 10 – Types de données élémentaires	262
Tableau 11 – Déclaration des types de données définis par l'utilisateur et initialisation	266
Tableau 12 – Opérations sur les références	278
Tableau 13 – Déclaration de variables	280
Tableau 14 – Initialisation de variables	281
Tableau 15 – Variables <code>ARRAY</code> de longueur variable	284
Tableau 16 – Variables directement représentées	286
Tableau 17 – Accès partiel aux variables <code>ANY_BIT</code>	292
Tableau 18 – Contrôle de l'exécution en utilisant graphiquement <code>EN</code> et <code>ENO</code>	297
Tableau 19 – Déclaration de fonction	304
Tableau 20 – Appel d'une fonction	306
Tableau 21 – Fonctions typées et en surcharge	308
Tableau 22 – Fonction de conversion de type de données	311
Tableau 23 – Conversion de type de données des types de données numériques	312
Tableau 24 – Conversion de type de données des types de données binaires	315
Tableau 25 – Conversion de type de données des types binaires et numériques	316
Tableau 26 – Conversion de type de données des types date et heure	319
Tableau 27 – Conversion de type de données des types caractère	319
Tableau 28 – Fonctions numériques et arithmétiques	320
Tableau 29 – Fonctions arithmétiques	321
Tableau 30 – Fonctions de décalage de bit	322
Tableau 31 – Fonctions booléennes au niveau du bit	322
Tableau 32 – Fonctions de sélection ^d	323
Tableau 33 – Fonctions de comparaison	324
Tableau 34 – Fonctions de chaîne de caractères	326
Tableau 35 – Fonctions numériques des types de données de temps et de durée	327
Tableau 36 – Fonctions additionnelles des types de données de temps <code>CONCAT</code> et <code>SPLIT</code>	328
Tableau 37 – Fonctions de conversion de boutisme	331
Tableau 38 – Fonctions des types de données énumérés	332
Tableau 39 – Fonctions de validation	332
Tableau 40 – Déclaration du type de bloc fonctionnel	334
Tableau 41 – Déclaration d'instance de bloc fonctionnel	338
Tableau 42 – Appel de bloc fonctionnel	340
Tableau 43 – Blocs fonctionnels normalisés bistables ^a	347
Tableau 44 – Blocs fonctionnels normalisés de détection de front	348
Tableau 45 – Blocs fonctionnels normalisés compteur	348
Tableau 46 – Blocs fonctionnels normalisés minuteur	350
Tableau 47 – Déclaration de programme	352
Tableau 48 – Classe	355
Tableau 49 – Déclaration d'instance de classe	357

Tableau 50 – Appel textuel de méthodes – Liste des paramètres formels et informels	361
Tableau 51 – Interface	372
Tableau 52 – Tentative d'affectation	381
Tableau 53 – Bloc fonctionnel orienté objet.....	382
Tableau 54 – Etape d'un SFC	392
Tableau 55 – Transition et condition de transition d'un SFC.....	394
Tableau 56 – Déclaration des actions d'un SFC	396
Tableau 57 – Association étape/action.....	397
Tableau 58 – Bloc d'action.....	398
Tableau 59 – Qualificateurs d'action	399
Tableau 60 – Caractéristiques de contrôle d'action	404
Tableau 61 – Evolution de séquence – graphique	405
Tableau 62 – Déclaration de configuration et de ressource	415
Tableau 63 – Tâche	418
Tableau 64 – Espace de noms.....	426
Tableau 65 – Options de déclaration d'espace de nom imbriqué	427
Tableau 66 – Directive d'espace de noms <code>USING</code>	429
Tableau 67 – Expression entre parenthèses du langage IL	432
Tableau 68 – Opérateurs de liste d'instructions.....	432
Tableau 69 – Appels du langage IL.....	434
Tableau 70 – Opérateurs normalisés de bloc fonctionnel du langage IL	436
Tableau 71 – Opérateurs du langage ST.....	438
Tableau 72 – Enoncés en langage ST.....	439
Tableau 73 – Eléments de contrôle d'exécution graphiques	450
Tableau 74 – Rails de puissance et éléments de liaison	451
Tableau 75 – Contacts	452
Tableau 76 – Bobines	453

COMMISSION ÉLECTROTECHNIQUE INTERNATIONALE

AUTOMATES PROGRAMMABLES –**Partie 3: Langages de programmation****AVANT-PROPOS**

- 1) La Commission Electrotechnique Internationale (CEI) est une organisation mondiale de normalisation composée de l'ensemble des comités électrotechniques nationaux (Comités nationaux de la CEI). La CEI a pour objet de favoriser la coopération internationale pour toutes les questions de normalisation dans les domaines de l'électricité et de l'électronique. A cet effet, la CEI – entre autres activités – publie des Normes internationales, des Spécifications techniques, des Rapports techniques, des Spécifications accessibles au public (PAS) et des Guides (ci-après dénommés "Publication(s) de la CEI"). Leur élaboration est confiée à des comités d'études, aux travaux desquels tout Comité national intéressé par le sujet traité peut participer. Les organisations internationales, gouvernementales et non gouvernementales, en liaison avec la CEI, participent également aux travaux. La CEI collabore étroitement avec l'Organisation Internationale de Normalisation (ISO), selon des conditions fixées par accord entre les deux organisations.
- 2) Les décisions ou accords officiels de la CEI concernant les questions techniques représentent, dans la mesure du possible, un accord international sur les sujets étudiés, étant donné que les Comités nationaux de la CEI intéressés sont représentés dans chaque comité d'études.
- 3) Les Publications de la CEI se présentent sous la forme de recommandations internationales et sont agréées comme telles par les Comités nationaux de la CEI. Tous les efforts raisonnables sont entrepris afin que la CEI s'assure de l'exactitude du contenu technique de ses publications; la CEI ne peut pas être tenue responsable de l'éventuelle mauvaise utilisation ou interprétation qui en est faite par un quelconque utilisateur final.
- 4) Dans le but d'encourager l'uniformité internationale, les Comités nationaux de la CEI s'engagent, dans toute la mesure possible, à appliquer de façon transparente les Publications de la CEI dans leurs publications nationales et régionales. Toutes divergences entre toutes Publications de la CEI et toutes publications nationales ou régionales correspondantes doivent être indiquées en termes clairs dans ces dernières.
- 5) La CEI elle-même ne fournit aucune attestation de conformité. Des organismes de certification indépendants fournissent des services d'évaluation de conformité et, dans certains secteurs, accèdent aux marques de conformité de la CEI. La CEI n'est responsable d'aucun des services effectués par les organismes de certification indépendants.
- 6) Tous les utilisateurs doivent s'assurer qu'ils sont en possession de la dernière édition de cette publication.
- 7) Aucune responsabilité ne doit être imputée à la CEI, à ses administrateurs, employés, auxiliaires ou mandataires, y compris ses experts particuliers et les membres de ses comités d'études et des Comités nationaux de la CEI, pour tout préjudice causé en cas de dommages corporels et matériels, ou de tout autre dommage de quelque nature que ce soit, directe ou indirecte, ou pour supporter les coûts (y compris les frais de justice) et les dépenses découlant de la publication ou de l'utilisation de cette Publication de la CEI ou de toute autre Publication de la CEI, ou au crédit qui lui est accordé.
- 8) L'attention est attirée sur les références normatives citées dans cette publication. L'utilisation de publications référencées est obligatoire pour une application correcte de la présente publication.
- 9) L'attention est attirée sur le fait que certains des éléments de la présente Publication de la CEI peuvent faire l'objet de droits de brevet. La CEI ne saurait être tenue pour responsable de ne pas avoir identifié de tels droits de brevets et de ne pas avoir signalé leur existence.

La Norme internationale CEI 61131-3 a été établie par le sous-comité 65B: Equipements de mesure et de contrôle-commande, du comité d'études 65 de la CEI: Mesure, commande et automation dans les processus industriels.

Cette troisième édition de la CEI 61131-3 annule et remplace la deuxième édition publiée en 2003. Cette édition constitue une révision technique.

La présente édition inclut les modifications techniques majeures suivantes par rapport à l'édition précédente:

Cette troisième édition est une extension compatible de la deuxième édition. Les principales extensions concernent de nouveaux types de données et de nouvelles fonctions de conversion, des références, des espaces de noms et des classes de caractéristiques orientées objet, et des blocs fonctionnels. Voir Annexe B.

Le texte de cette norme est issu des documents suivants:

FDIS	Rapport de vote
65B/858/FDIS	65B/863/RVD

Le rapport de vote indiqué dans le tableau ci-dessus donne toute information sur le vote ayant abouti à l'approbation de cette norme.

Cette publication a été rédigée selon les Directives ISO/CEI, Partie 2.

Une liste de toutes les parties de la série CEI 61131, publiée sous le titre général *Automates programmables*, peut être consultée sur le site web de la CEI.

Le comité a décidé que le contenu de cette publication ne sera pas modifié avant la date de stabilité indiquée sur le site web de la CEI sous "<http://webstore.iec.ch>" dans les données relatives à la publication recherchée. A cette date, la publication sera

- reconduite,
- supprimée,
- remplacée par une édition révisée, ou
- amendée.

AUTOMATES PROGRAMMABLES –

Partie 3: Langages de programmation

1 Domaine d'application

La présente partie de la CEI 61131 spécifie la syntaxe et la sémantique des langages de programmation utilisés pour les automates programmables tels que définis dans la Partie 1 de la CEI 61131.

Les fonctions d'entrée de programme, d'essai, de surveillance, de système d'exploitation, etc. du système sont spécifiées dans la Partie 1 de la CEI 61131.

La présente partie de la CEI 61131 spécifie la syntaxe et la sémantique d'une suite unifiée de langages de programmation utilisés pour les automates programmables (AP). Cette suite est constituée de deux langages textuels, liste d'instructions (IL, Instruction List) et texte structuré (ST, Structured Text), et de deux langages graphiques, diagramme à contacts (LD, Ladder Diagram) et diagramme de bloc fonctionnel (FBD, Function Block Diagram).

Un autre ensemble d'éléments graphiques et textuels équivalents appelé "diagramme fonctionnel séquentiel" (SFC, Sequential Function Chart) est défini pour structurer l'organisation interne des programmes pour automate programmable et des blocs fonctionnels. En outre, des éléments de configuration qui prennent en charge l'installation des programmes pour automate programmable dans des systèmes d'automate programmable sont définis.

De plus, des caractéristiques sont définies pour faciliter la communication entre les automates programmables et les autres composants des systèmes automatisés.

2 Références normatives

Les documents suivants sont cités en référence de manière normative, en intégralité ou en partie, dans le présent document et sont indispensables pour son application. Pour les références datées, seule l'édition citée s'applique. Pour les références non datées, la dernière édition du document de référence s'applique (y compris les éventuels amendements).

CEI 61131-1, *Automates programmables – Partie 1: Informations générales*

CEI 61131-5, *Automates programmables – Partie 5: Communications*

ISO/CEI 10646:2012, *Technologies de l'information – Jeu universel de caractères codés (JUC)*

ISO/CEI/IEEE 60559, *Information technology – Microprocessor Systems – Floating-Point arithmetic* (disponible en anglais seulement)

3 Termes et définitions

Pour les besoins du présent document, les termes et définitions donnés dans la CEI 61131-1 ainsi que les suivants s'appliquent.

3.1

temps absolu

combinaison d'informations d'heure et de date

3.2

chemin d'accès

association d'un nom symbolique à une variable dans le cadre d'une communication ouverte

3.3

action

variable booléenne ou collection d'opérations à effectuer, conjointement avec une structure de commande associée

3.4

bloc d'action

élément de langage graphique qui utilise une variable d'entrée booléenne pour déterminer la valeur d'une variable de sortie booléenne ou la condition d'activation d'une action, conformément à une structure de commande prédéterminée

3.5

agrégat

collection structurée d'objets de données formant un type de données

[SOURCE: ISO/AFNOR:1989]

3.6

tableau

agrégat constitué d'objets de données ayant des attributs identiques, dont chacun peut être référencé de manière univoque par indigage

[SOURCE: ISO/AFNOR:1989]

3.7

affectation

mécanisme permettant d'attribuer une valeur à une variable ou à un agrégat

[SOURCE: ISO/AFNOR:1989]

3.8

type de base

type de données, type de bloc fonctionnel ou classe à partir desquels d'autres types sont hérités/dérivés

3.9

nombre en base

nombre représenté dans une base spécifiée autre que dix

3.10

décimal codé binaire

BCD

codage de nombres décimaux dans lequel chaque chiffre est représenté par sa propre séquence binaire

3.11

bloc fonctionnel bistable

bloc fonctionnel ayant deux états stables contrôlés par une ou plusieurs entrées

3.12**chaîne de bits**

élément de données constitué d'un ou plusieurs bits

3.13**littéral de chaîne de bits**

littéral qui représente directement une valeur de chaîne de bits de type de données BOOL, BYTE, WORD, DWORD ou LWORD

3.14**corps**

ensemble d'opérations de l'unité d'organisation de programme

3.15**appel**

construction de langage provoquant l'exécution d'une fonction, d'un bloc fonctionnel ou d'une méthode

3.16**chaîne de caractères**

agrégat constitué d'une séquence ordonnée de caractères

3.17**littéral de chaîne de caractères**

littéral qui représente directement une valeur de caractère ou de chaîne de caractères de type de données CHAR, WCHAR, STRING ou WSTRING

3.18**classe**

unité d'organisation de programme constituée:

- de la définition d'une structure de données;
- d'un ensemble de méthodes à effectuer sur la structure de données

3.19**commentaire**

construction de langage, permettant d'insérer dans un programme des textes quelconques, sans incidence sur l'exécution du programme

[SOURCE: ISO/AFNOR:1989]

3.20**configuration**

élément de langage correspondant à un système d'automate programmable

3.21**constante**

élément de langage qui déclare un élément de données avec une valeur fixe

3.22**bloc fonctionnel compteur**

bloc fonctionnel qui accumule une valeur correspondant au nombre de modifications détectées à une ou plusieurs entrées spécifiées

3.23**type de données**

ensemble de valeurs associé à un ensemble d'opérations permises

[SOURCE: ISO/AFNOR:1989]

3.24

date et heure

date de l'année et heure du jour représentées sous la forme d'un élément de langage unique

3.25

déclaration

mécanisme permettant d'établir la définition d'un élément de langage

3.26

délimiteur

caractère ou combinaison de caractères utilisés pour séparer des éléments de langage de programme

3.27

classe dérivée

classe créée par héritage d'une autre classe

Note 1 à l'article: La classe dérivée est également appelée "classe étendue" ou "classe enfant".

3.28

type de données dérivé

type de données créé à l'aide d'un autre type de données

3.29

type de bloc fonctionnel dérivé

type de bloc fonctionnel créé par héritage d'un autre type de bloc fonctionnel

3.30

représentation directe

moyen de représentation d'une variable dans un programme pour automate programmable à partir duquel une correspondance spécifiée par mise en œuvre à un emplacement physique ou logique peut être directement déterminée

3.31

double mot

élément de données contenant 32 bits

3.32

liaison dynamique

situation dans laquelle l'instance d'un appel de méthode est extraite pendant l'exécution en fonction du type réel d'une instance ou d'une interface

3.33

évaluation

processus d'établissement d'une valeur pour une expression ou une fonction, ou pour les sorties d'un réseau ou d'une instance de bloc fonctionnel, pendant l'exécution d'un programme

3.34

élément de contrôle d'exécution

élément de langage qui contrôle le flux d'exécution d'un programme

3.35

front descendant

passage de 1 à 0 d'une variable booléenne

3.36

fonction

élément de langage qui, lorsqu'il est exécuté, produit généralement un résultat sur des éléments de données et, éventuellement, des variables de sortie additionnelles

3.37**instance de bloc fonctionnel**

instance d'un type de bloc fonctionnel

3.38**type de bloc fonctionnel**

élément de langage constitué:

- de la définition d'une structure de données divisée en variables d'entrée, de sortie et internes; et
- d'un ensemble d'opérations ou de méthodes à effectuer sur les éléments de la structure de données lorsqu'une instance du type de bloc fonctionnel est appelée

3.39**diagramme de bloc fonctionnel**

réseau dans lequel les nœuds sont des instances de bloc fonctionnel, des fonctions ou appels de méthode représentés graphiquement, des variables, des littéraux et des étiquettes

3.40**type de données générique**

type de données qui représente plusieurs types de données

3.41**variable globale**

variable dont la portée est globale

3.42**adressage hiérarchique**

représentation directe d'un élément de données en tant que membre d'une hiérarchie physique ou logique

EXEMPLE Un point dans un module qui est contenu dans une baie, qui est elle-même contenue dans une armoire, etc.

3.43**identificateur**

combinaison de lettres, de chiffres et de caractères de soulignement qui commence par une lettre ou un caractère de soulignement, et qui nomme un élément de langage

3.44**mise en œuvre**

version produit d'un automate programmable ou d'un outil de programmation et de débogage fourni par l'Intégrateur

3.45**Intégrateur**

fabricant de l'automate programmable ou de l'outil de programmation et de débogage fourni à l'utilisateur pour programmer une application pour automate programmable

3.46**héritage**

création d'une nouvelle classe, d'un nouveau type de bloc fonctionnel ou d'une nouvelle interface respectivement sur la base d'une classe, d'un type de bloc fonctionnel ou d'une interface existants

3.47**valeur initiale**

valeur affectée à une variable au démarrage du système

3.48

variable d'entrée-sortie

variable utilisée pour fournir une valeur à une unité d'organisation de programme et pour retourner une valeur depuis l'unité d'organisation de programme

3.49

variable d'entrée

variable qui est utilisée pour fournir une valeur à une unité d'organisation de programme autre qu'une classe

3.50

instance

copie individuelle nommée de la structure de données associée à un type de bloc fonctionnel, à une classe ou à un type de programme, qui conserve ses valeurs d'un appel des opérations associées au suivant

3.51

nom d'instance

identificateur associé à une instance spécifique

3.52

instanciation

création d'une instance

3.53

entier

nombre entier qui peut contenir des valeurs positives, nulles et négatives

3.54

littéral entier

littéral qui représente directement une valeur entière

3.55

interface

élément de langage dans le contexte de la programmation orientée objet contenant un ensemble de prototypes de méthode

3.56

mot-clé

unité lexicale qui caractérise un élément de langage

3.57

étiquette

construction de langage nommant une instruction, un réseau ou un groupe de réseaux et comprenant un identificateur

3.58

élément de langage

tout élément identifié par un symbole sur le côté gauche d'une règle de production dans la spécification formelle

3.59

littéral

unité lexicale qui représente directement une valeur

[SOURCE: ISO/AFNOR:1989]