

La norme CEI 61131-3 : une ressource de programmation standard

La norme CEI 61131-3 résulte du premier vrai effort de normalisation des langages de programmation pour l'automatisation industrielle. Avec son appui mondial, cette norme est indépendante de n'importe quel fournisseur.

La CEI 61131-3 est la troisième partie de la famille de la norme CEI 61131. Elle se compose des parties suivantes :

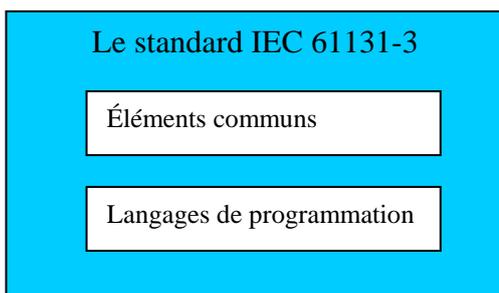
- Partie 1 : Vue d'ensemble Générale.
- Partie 2 : Matériel.
- Partie 3 : Langages de Programmation.
- Partie 4 : Directives d'Utilisateur.
- Partie 5 : Communication.
- Partie 7 : Fuzzy Logic
- Partie 8 : Directives d'Applicaton.

Il y a beaucoup de manières de regarder la partie 3 de cette norme. En voici quelques unes :

- le résultat de la force 3, langages de programmation, dans la CEI TC65 SC65B
- le résultat du travail dur à côté de 7 compagnies internationales ajoutant des dizaines d'années d'expérience dans le domaine de l'automatisation industrielle
- approximativement 200 pages de texte, avec 60 tables, y compris des tables de caractéristiques
- les spécifications de la syntaxe et de la sémantique d'une suite unifiée des langages de programmation, y compris le modèle global de logiciel et une langue structurante.

Une autre vue élégante consiste à dédoubler la norme en deux parties (voir la figure ci-dessous) :

1. Éléments Communs
2. Langages de Programmation



Regardons plus en détail ces éléments :

Éléments Communs

Types des données

Parmi les éléments communs, les types de données sont définis. Le typage des données empêchent des erreurs à un stade initial de la conception. Tous les paramètres utilisés sont typés. Ceci évite par exemple de diviser une date par un nombre entier. Les types de données communs sont les booléens, nombres entiers, réels, octets et mots, mais également Date, Time_of_Day et string.

Sur la base de ces types, on peut définir ses types de données personnels, connus sous le nom de types de données dérivés. De

cette façon on peut définir un canal d'entrée analogique comme un type de données, et réutiliser cette définition aussi souvent que nécessaire.

Variables

Des variables sont uniquement assignées aux adresses explicites du matériel (par exemple entrée et sorties) dans les configurations, les ressources ou les programmes.

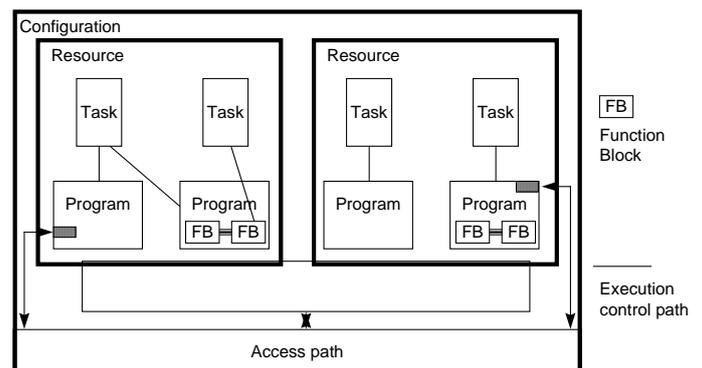
De cette façon un niveau élevé d'indépendance de matériel est créé, aidant à la réutilisabilité du logiciel.

Les portées des variables sont normalement limitées à l'unité d'organisation dans laquelle elles sont déclarées, par exemple localement. Ceci signifie que leurs noms peuvent être réutilisés dans d'autres parties sans aucun conflit, éliminant une autre source d'erreurs.

Si les variables doivent avoir une portée globale, elles doivent être déclarées en tant que telles (VAR_GLOBAL). On peut leur assigner une valeur initiale au démarrage à chaud ou à froid, afin d'avoir le bon paramètre lors du fonctionnement du système.

Configuration, ressources et tâches

Pour comprendre ces termes, regardons le modèle de logiciel, tel que défini dans la norme (voir ci-dessous).



Au niveau le plus élevé, le logiciel nécessaire pour résoudre un problème particulier de commande peut être appelé une **configuration**. Une configuration est spécifique à un type particulier de système de commande, y compris l'arrangement du matériel, c'est-à-dire les ressources processeurs, les adresses de mémoire pour les canaux d'I/O et les possibilités spécifiques du système.

Dans une configuration on peut définir une ou plusieurs **ressources**. On peut considérer une ressource comme un service qui peut exécuter des programmes CEI.

Dans une ressource, une ou plusieurs **tâches** (tasks) peuvent être définies. Les tâches commandent l'exécution d'un ensemble de programmes et/ou de blocs fonctionnels. Ceux-ci peuvent être exécutés périodiquement ou à l'occurrence d'un événement spécifique, tel que la modification d'une variable.

Les **programmes** sont établis à partir d'un certain nombre de différents éléments de logiciel écrits dans n'importe laquelle des langues CEI. Typiquement, un programme se compose d'un réseau de **fonctions** et de **blocs fonctionnels**, qui peuvent échanger des données. Les

fonctions et les blocs fonctionnels sont les éléments de base, contenant une structure de données et un algorithme.

Comparons ceci à un automate programmable (AP) conventionnel : il contient une ressource exécutant une tâche, contrôlant un programme, fonctionnant dans une boucle fermée. La norme CEI 61131-3 ajoute beaucoup de possibilités à ceci, ouvrant le système vers des applications nouvelles incluant le multitraitement et les programmes pilotés par événement.

Ce futur n'est pas si lointain : il suffit de considérer les systèmes distribués ou les systèmes de commande en temps réel. La norme CEI 61131-3 convient à une large étendue des applications, sans devoir apprendre de nouveaux langages de programmation.

Modules (POU)

Dans la norme CEI 61131-3, les programmes, blocs fonctionnels et fonctions s'appellent modules ou unités d'organisation de programme (POU = Program Organization Unit).

Fonctions

La norme CEI a défini des fonctions standards et des fonctions définies par l'utilisateur. Les fonctions standard sont par exemple ADD(ition), ABS (absolus), SQRT, sinus et COSinus. Les fonctions définies par l'utilisateur peuvent, une fois définies, être réutilisées sans limite.

Blocs fonctionnels (FB)

Les blocs fonctionnels sont l'équivalent de circuits intégrés (ICs), prévus pour effectuer une opération de commande spécialisée. Ils contiennent des données aussi bien que l'algorithme. Ils peuvent ainsi mémoriser le passé (ce qui est l'une des différences d'avec les fonctions).

Ils ont une interface bien définie et des variables internes cachées, comme un IC ou une boîte noire. De cette façon ils déterminent une séparation claire entre différents niveaux de programmeurs, ou de personnes de l'entretien.

Une boucle de régulation de la température, ou PID, est un excellent exemple d'un bloc fonctionnel. Une fois défini, il peut être réemployé à volonté, dans le même programme, différents programmes, ou même différents projets. Ceci le rend fortement réutilisable.

Les blocs fonctionnels peuvent être écrits dans n'importe laquelle des langues du CEI, et même dans la plupart des cas en "C". De cette façon ils peuvent être définis par l'utilisateur. Les blocs fonctionnels dérivés sont basés sur les blocs fonctionnels standards, mais on peut aussi, conformément à la norme, définir des blocs fonctionnels entièrement nouveaux. Le standard définit uniquement le cadre.

Les interfaces des fonctions et des blocs fonctionnels sont décrits de la même manière :

```

FUNCTION_BLOCK Exemple
VAR_INPUT :
    X:      BOOL;
    Y:      BOOL;
END_VAR;

VAR_OUTPUT :
    Z:      BOOL;
END_VAR;

(* Expressions du corps du bloc fonctionnel *)

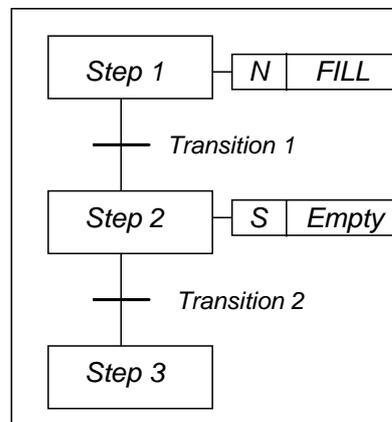
END_FUNCTION_BLOCK
    
```

Les déclarations ci-dessus décrivent l'interface d'un bloc fonctionnel avec deux paramètres booléens d'entrée et un paramètre de sortie booléen.

Programmes

Avec les blocs de base mentionnés ci-dessus, on peut dire qu'un programme est un réseau de fonctions et de blocs fonctionnels. Un programme peut être écrit dans n'importe lequel des langages de programmation définis.

Sequential Function Chart, SFC



SFC décrit graphiquement le comportement séquentiel d'un programme de commande. Il est dérivé des réseaux de Pétri et du CEI 848 Grafcet, avec les adaptations nécessaires pour convertir la représentation d'une norme de documentation en ensemble d'éléments de commande d'exécution. SFC structure l'organisation interne d'un programme, et aide à décomposer un problème de commande en éléments individuels, tout en maintenant la vue d'ensemble.

SFC se compose d'étapes, liées avec des blocs d'action et des transitions. Chaque étape représente un état particulier des systèmes commandés. Une transition est associée à une condition, qui, si elle est vraie, cause l'arrêt de l'étape précédent la transition et l'activation de la prochaine étape. Des étapes sont liées aux blocs d'action, effectuant une certaine action de commande. Chaque élément peut être programmé dans n'importe laquelle des langues de la norme CEI, y compris SFC lui-même.

On peut employer des ordres alternatifs et même des ordres parallèles, tels que généralement requis dans des usages de lot. Par exemple, un ordre est employé pour le procédé primaire, et la seconde pour surveiller les contraintes de fonctionnement globales. En raison de sa structure générale, SFC fournit également un outil de communication, combinant des personnes de différents milieux, des départements ou des pays.

Langages de Programmation

Quatre langages de programmation sont définis dans la norme. Ceci signifie que leur syntaxe et sémantique ont été définies, ne partant aucune liberté pour des dialectes. Une fois que vous les avez appris, vous pouvez employer une grande variété de systèmes basés sur cette norme.

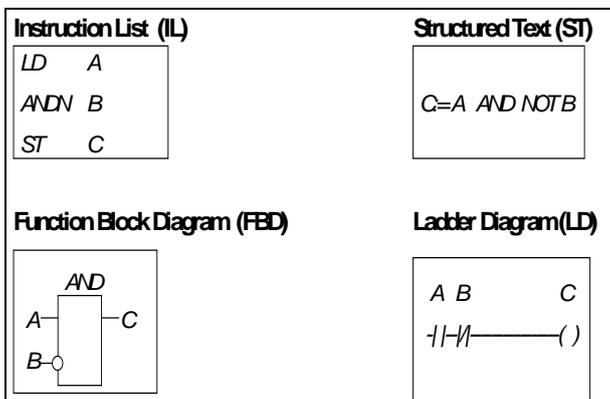
Les langues se composent de deux versions textuelles et deux graphiques :

Textuel :

- Liste d'instructions IL
- Texte structuré ST

Graphique :

- Schéma contact (Ladder) LD
- Schéma en blocs fonctionnels FBD



Dans la figure ci-dessus, chacune des quatre langues décrit le même programme simple.

Le choix du langage de programmation est dépendant :

- Des connaissances du programmeur
- Du problème à résoudre
- Du niveau de description du problème
- De la structure du système de commande
- De l'interface avec les autres programmeurs ou partenaires

Chacune des quatre langues est liée aux autres: elles fournissent à une suite commune, avec un lien à une expérience existante. De cette façon elles fournissent également un outil de communication, combinant des personnes de différents milieux.

Le **Schéma contact** a ses racines aux Etats-Unis. Il est basé sur la présentation graphique de la logique câblée de relais.

La **Liste d'instruction** trouve son origine chez les fabricants d'AP européens. Cette langue textuelle ressemble à l'assembleur.

Le **Schéma en blocs fonctionnels** est très courant à l'industrie de processus. Il exprime le comportement des fonctions, des blocs fonctionnels et des programmes comme ensemble de blocs graphiques reliés ensemble, comme sur les schémas de circuits électroniques. Il schématise un système de commande en termes de flux des signaux entre des éléments de traitement.

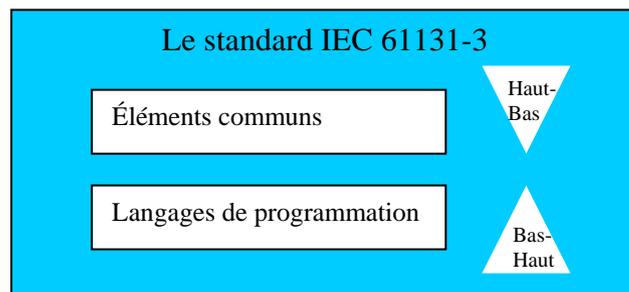
Le **Texte structuré** est un langage de haut niveau très puissant avec ses racines dans l'ADA, le Pascal et le « C ». Il contient tous éléments essentiels d'un langage de programmation moderne, y compris des branchement (IF-THEN-ELSE et CASE) et des boucles d'itération (FOR, WHILE et REPEAT). Ces éléments peuvent également être imbriqués.

Il est parfaitement adapté pour la définition des blocs fonctionnels complexes qui peuvent être employés dans les autres langages.

Exemple en ST :

```
I:=25;
  WHILE J<5 DO
    Z:= F(I+J);
  END_WHILE
IF B_1 THEN
  %QW100:= INT_TO_BCD(Display)
ENDIF
CASE TW OF
  1,5:  TEMP := TEMP_1;
  2:    TEMP := 40;
  4:    TEMP := FTMP(TEMP_2);
ELSE
  TEMP := 0;
  B_ERROR :=1;
END_CASE
```

Du haut vers le bas ou du bas vers le haut



La norme autorise deux manières de développer votre programme : du haut vers le bas (top-down) et du bas vers le haut (bottom-up). Vous pouvez spécifier votre application entière et la diviser en éléments secondaires, déclarer vos variables, et ainsi de suite. Vous pouvez aussi commencer à programmer votre application depuis le bas, par exemple par l'intermédiaire des fonctions dérivées et des blocs fonctionnels. Quel que soit votre choix, l'environnement de développement vous aidera durant le processus tout entier.

Réalisations

Les conditions globales de la norme CEI 61131-3 ne sont pas simples à remplir. Pour cette raison, la norme autorise des réalisations partielles dans divers aspects tels que nombre de langages, de fonctions et de blocs fonctionnels supportés. Ceci laisse de la liberté au fournisseur, mais un utilisateur doit en être conscient pendant sa phase de sélection d'un produit. En outre, une nouvelle version d'un produit peut implémenter bien plus d'éléments de la norme.

Beaucoup d'environnements de programmation CEI 61131-3 offrent tout ce que l'on peut attendre d'environnements modernes de développement : utilisation de la souris, menus contextuels, écrans de programmation graphiques, multi-fenêtrés, aide et références par lien hypertexte, vérification pendant la conception. Ces fonctionnalités ne sont pas spécifiées par la norme elle-même, elles sont l'un des éléments où les fournisseurs peuvent se différencier.

Conclusion

Les implications techniques de la norme du CEI 61131-3 sont élevées; elles laissent un large champ à la croissance et à la différenciation. La norme est donc appropriée pour évoluer positivement dans le siècle à venir. La norme CEI 61131-3 aura un grand impact dans l'ensemble de l'industrie de l'automatisation. Elle ne se limitera certainement pas au marché conventionnel des AP. Cette norme a déjà été adoptée dans le marché de commande de mouvement (motion control), des systèmes répartis et des systèmes de commande basés sur PC (softlogic), y compris en relation avec des systèmes de supervision. Les domaines d'application se développent chaque jour. Ils apportent par là-même de nombreux avantages aux utilisateurs/programmeurs. En voici pour mémoire quelques-uns:

- Réduction du gaspillage des ressources humaines, dans la formation, la mise au point, l'entretien et le consulting.
- Orientation sur la résolution de problèmes plutôt que sur la programmation, grâce à un niveau élevé de réutilisabilité de logiciel.
- Diminution des mauvaises interprétations et erreurs.
- Techniques de programmation utilisables dans un large environnement : application générique pour la commande industrielle.
- Possibilité de combiner des composants différents de différents programmes, projets, lieux, fournisseurs et/ou pays

Enrichissements de la norme par PLCopen

PLCopen est une association mondiale soutenant le CEI 61131-3, indépendante de tout fournisseur et de tout produit. Grâce à la mise en application de cette norme sur beaucoup d'environnements de développement, les utilisateurs peuvent passer facilement avec le minimum d'effort et de formation entre les différentes marques et types de produits. Le portage des applications exige un minimum de travail. L'organisation PLCopen œuvre pour la promotion de la norme, pour son extension ainsi que pour son évolution dans un sens technique. Ce dernier point inclut la certification et l'interchangeabilité. Les membres de PLCopen peuvent participer aux comités, ils reçoivent des informations préalables, jouissent d'une identité plus forte, et ont une influence sur les résultats. En outre, ils peuvent employer le logo de PLCopen pour leur promotion. Les comités travaillant dans PLCopen et leurs résultats sont décrits brièvement ci-dessous.

Résultats techniques

Les comités techniques (**TC**), avec des représentants des membres de PLCopen, travaillent sur des thèmes spécifiques.

Dans **TC1 - normes**, PLCopen rassemble les propositions de ses membres pour le groupe de travail du CEI 65B WG7, développe une position commune, et distribue l'information relative. Ceci a été spécifiquement concret à la deuxième édition de la norme, qui a été libérée au début de 2001.

TC2 - fonctions définit les bibliothèques communes de la fonction (blocs) pour des domaines d'application spécifiques. Un exemple est la définition d'une bibliothèque des blocs fonctionnels pour la commande de mouvement. Cette standardisation relie la commande de mouvement d'une manière logique à la commande industrielle. En tant que telle, elle fournit une interface commune envers les utilisateurs, programmeurs et les personnes d'installation et d'entretien. Avec des applications multiples de cette bibliothèque, l'extension d'un système de commande est beaucoup plus facile, même s'il y a différentes architectures et/ou marques de contrôleurs utilisés. L'échange d'une partie des programmes par l'intermédiaire du niveau de réutilisabilité de PLCopen (voir ci-dessous pour plus d'information) joue un rôle essentiel ici aussi.

Certification et test de conformité

TC3 - certification définit un système d'homologation pour PSEs, (PSE = Program Support Environments) fournisseurs d'environnements de développement. Chaque PSE peut être examiné pour prouver qu'il est conforme à un sous-ensemble (indiqué par PLCopen) de la norme CEI 61131-3. Cette norme contient un grand nombre de tables de spécifications et PLCopen a défini quels éléments de ces tables doivent être soutenus pour remplir des conditions de conformité. En outre, PLCopen a étendu la norme pour garantir la réutilisabilité des blocs fonctionnels dérivés par utilisateur entre PSEs.

Le niveau de conformité, CL

Avec la large étendue des domaines d'applications de la norme CEI 61131-3, toutes les implémentations n'emploient pas exactement les mêmes types de données. En conséquence, la certification selon le niveau de conformité, CL, implique que le fournisseur d'un PSE sélectionne les types de données soutenus par son produit conformément à son rapport de conformité. Toutes les spécifications de la norme CEI sont examinées. Ceci signifie que, bien que l'essai soit de type Oui/Non (conforme ou non-conforme), il peut y avoir des différences entre deux produits certifiés. Ces différences peuvent influencer la réutilisabilité des blocs fonctionnels dérivés par l'utilisateur. Le nombre total de types de données spécifié par la norme s'élève à 26 (tableau 10 et 12 de la norme). Ceux-ci vont des booléens aux structures complexes. Par conséquent, CL a 26 options : le type de

données est soutenu ou non soutenu. Seuls les types de données soutenus sont employés pour le test.

En outre, le niveau de **réutilisabilité, RL**, est consacré à rendre les fonctions et les blocs fonctionnels réutilisables sur un PSE

différent. Ceci est fait par l'intermédiaire de l'échange d'un fichier texte en langage structuré ST. Pour la représentation dans d'autres langues, un outil de conversion vers ou depuis ST peut être inclus. C'est une contribution majeure, mais naturelle, de PLCopen à la norme CEI 61131-3.

Il existe historiquement une classe inférieure appelée niveau de base (Base Level), qui définit un noyau par langue de la norme. Bien que plutôt limité, elle est utilisable pour développer des applications basées sur lui. Le niveau de base fournit un niveau d'entrée pour les fournisseurs, leur permettant de démontrer leur engagement à la norme. Pour les utilisateurs, il fournit une interprétation standardisée de la norme, particulièrement importante s'ils doivent travailler avec des systèmes de différents fournisseurs. Les spécifications détaillées de la plupart des langages du CEI 61131-3 sont déjà terminées. Le travail est en marche pour les langages restants. La méthode de test de conformité et la procédure d'accréditation pour des laboratoires d'essai ont été établis. Des laboratoires indépendants d'essai ont été accrédités et un nombre croissant de produits ont été certifiés. Pour une liste complète référez-vous svp au site www.plcopen.org.

TC4 - communications travaille sur le thème de la relation entre la communication et les langages de programmation, c'est-à-dire pour faire correspondre Profibus et CANopen par l'intermédiaire de CEI 61131-5 à la norme CEI 61131-3.

TC5 - sécurité du logiciel prépare des recommandations pour appliquer la norme CEI 61131-3 à des applications liées à la sécurité (y compris l'adaptation de la norme en ce sens), en particulier par rapport aux nouvelles normes de sûreté CEI 61508 et 61511.

TC6 - XML travaille sur les spécifications de schémas XML pour tous les langages, aussi bien que les projets complets. Ces spécifications fourniront la base pour l'échange, aussi bien que l'interfaçage à d'autres outils logiciels, y compris les outils de développement de haut niveau, les outils de documentation et de vérification.

Evénements promotionnels

Une mission importante de PLCopen est d'informer les utilisateurs/programmeurs sur la méthodologie de programmation standardisée des commandes industrielles normalisées, par les moyens suivants:

- site web de PLCopen : www.plcopen.org ;
- publication d'un bulletin électronique et imprimé gratuit, appelée "PLCopening" ;
- publications dans la presse
- participation aux expositions commerciales et aux conférences ;
- organisation de conférences, comme l'ICP en octobre, et conférences.

Les comités promotionnels **PC1, PC3, PC4 et PC5** soutiennent ces activités. Les membres de PLCopen sont considérés sur le marché comme défenseurs des standards ouverts.

PC2 - formation commune a défini une base commune pour la formation. Ceci inclut la certification. De cette façon, des centres de formation certifiés pour former sur le CEI 61131-3 peuvent être facilement identifiés.

Avantages liés à l'adhésion

Une adhésion à PLCopen garantit beaucoup d'avantages aux fournisseurs, et aux institutions. PLCopen soutient fortement la communauté d'utilisateurs. Des catégories additionnelles d'adhésion ont été créées à cette fin. En rejoignant PLCopen, on témoigne d'un engagement clair à la norme du CEI 61131-3, on se démarque en tant que tel, on peut employer le logo de PLCopen, et on peut avoir des informations préliminaires ainsi que de l'influence sur le travail effectué. Pour plus d'informations, examinez svp le site Web www.plcopen.org,