

La norme CEI 61131

Norme définissant le
fonctionnement des API

1 – INTRODUCTION

2 – MODULES LOGICIELS

3 – VARIABLES

4 – LANGAGES

1 – Introduction

2 – MODULES LOGICIELS

3 – VARIABLES

4 – LANGAGES

Introduction

- ▶ La norme CEI 61131 (en anglais, IEC 61131) a été définie par la Commission électrotechnique internationale (CEI)
 - ▶ Première version en 1993
 - ▶ Elle est révisée tous les 10 ans environ
 - ▶ La dernière version (troisième édition) date de 2013
- ▶ Elle définit des caractéristiques communes à tous les automates afin de garantir une normalisation des aspects principaux
 - ▶ Prescriptions électriques
 - ▶ Prescriptions mécaniques
 - ▶ Prescriptions fonctionnelles
 - ▶ Procédures
- ▶ Dans sa section 3 (CEI 61131-3), elle définit des langages de programmation standards pour les API
 - ▶ Syntaxe
 - ▶ Sémantique
 - ▶ Représentation
 - ▶ C'est cette section qui nous intéressera dans ce cours

La norme représente la *théorie*, dans la pratique, les constructeurs peuvent prendre certaines libertés

1 – INTRODUCTION

2 – Modules logiciels

3 – VARIABLES

4 – LANGAGES

Modules logiciels

- ▶ Un certain nombre de modules logiciels (Program Organization Units) sont définis pour structurer les programmes
 - ▶ La fonction (Function)
 - ▶ Une ou plusieurs variables d'entrées
 - ▶ Une seule variable de sortie
 - ▶ Pas de mémoire interne
 - ▶ Peut avoir une entrée EN et une sortie ENO indiquant l'absence d'erreur
 - ▶ Permet le chaînage entre plusieurs blocs
 - ▶ Le bloc fonctionnel (Function block)
 - ▶ Peut avoir plusieurs sorties
 - ▶ Peut avoir une mémoire interne
 - ▶ Le programme (Program)
 - ▶ Module logiciel construit à l'aide de fonctions et blocs fonctionnels
 - ▶ Peut déclarer des variables globales

1 – INTRODUCTION

2 – MODULES LOGICIELS

3 – Variables

4 – LANGAGES

Emplacement

- ▶ 3 emplacements existent pour les variables
 - ▶ Entrée
 - ▶ Mémoire interne
 - ▶ Sortie
- ▶ Chaque emplacement a un préfixe associé

Emplacement	Préfixe associé
Entrée	I
Mémoire	M
Sortie	Q

Types natifs

- ▶ Un certain nombre de types de variables sont définis
- ▶ Chaque type a une taille en bits
- ▶ Chaque type a un préfixe associé

Vecteurs de bits

Type	Taille	Préfixe
Bool	1 bit	Aucun
Byte	8 bits (1 octet)	B
Word	16 bits (2 octets)	W
Dword (Double word)	32 bits (4 octets)	D
Lword (Long word)	64 bits (8 octets)	L

Types dérivés

- ▶ Par ailleurs, d'autres types de données existent, basés sur ces tailles
 - ▶ Les entiers, signés et non signés
 - ▶ Les réels (en virgule flottante)
 - ▶ Les unités de temps
 - ▶ Les caractères
 - ▶ Etc.
- ▶ Les préfixes associé aux types dérivés sont les mêmes que ceux des types natifs ayant la même taille

Type	Taille	Signé
SINT (Short int)	1 octet	Oui
INT	2 octets	Oui
DINT (Double int)	4 octets	Oui
USINT	1 octet	Non
UINT	2 octets	Non
UDINT	4 octets	Non

Type	Taille
REAL	4 octets
LREAL	8 octets

Type
TIME
LTIME
DATE
LDATE

Type
CHAR
WCHAR
STRING
WSTRING

Déclaration d'une variable

10

- ▶ La notation d'une variable se construit par la concaténation des éléments suivants :
 - ▶ Le symbole « % »
 - ▶ Le préfixe d'emplacement
 - ▶ Le préfixe de taille
 - ▶ Un ou plusieurs entiers non signés
 - ▶ Si plusieurs entiers, il doivent être séparés par le symbole « . »
 - ▶ On n'a plusieurs entiers que dans le cas de la définition d'un bit : on place après le point la position du bit dans l'octet

▶ Exemples

- ▶ Entrée d'un seul bit
 - ▶ %I3.5
- ▶ Mémoire sur un octet
 - ▶ %MB2
- ▶ Sortie sur 2 octets
 - ▶ %QW42

Le bit 5 de l'octet 3
de la mémoire
d'entrée

L'octet 2 de la
mémoire interne

L'octet 42 de la
mémoire de sortie

Et comme on est sur 2
octets (W pour word),
l'octet 43 qui suit fait
également partie de la
variable)

Utilisation de l'espace mémoire

11

- ▶ Les notations indiquent uniquement l'emplacement de *départ* de la variable
- ▶ Par exemple, une variable de type « Double word » utilisera 4 octets
- ▶ Si on la déclare à l'emplacement %M0, elle utilisera les emplacements suivants
 - ▶ %M0
 - ▶ %M1
 - ▶ %M2
 - ▶ %M3
- ▶ Il ne faut donc surtout pas déclarer une autre variable à l'emplacement %M2 par exemple
 - ▶ Sinon il y a chevauchement, et l'une des variables va écraser l'autre

Exemple

- ▶ Placer dans la mémoire interne ci-contre les variables suivantes
 - ▶ %M3.5
 - ▶ %MB2
 - ▶ %MB3
 - ▶ %MW6
 - ▶ %MB7
 - ▶ %M7.3
 - ▶ Attention aux conflits !

- ▶ Attention, une restriction existe sur l'espace mémoire interne
- ▶ L'espace mémoire entre 0 et 99 est réservé aux variables nécessaires au fonctionnement du système
- ▶ L'utilisateur peut donc déclarer ses propres variables à partir de l'adresse %MB100
- ▶ Cette restriction ne s'applique pas aux espaces mémoire d'entrée et de sortie

1 – INTRODUCTION

2 – MODULES LOGICIELS

3 – VARIABLES

4 – Languages

Les langages

- ▶ La norme définit plusieurs langages, graphiques et textuels
 - ▶ Langage IL (Instruction List)
 - ▶ Langage proche de l'assembleur
 - ▶ Marqué comme obsolète dans la dernière édition, on ne l'étudiera pas
 - ▶ Langage ST (Structured text)
 - ▶ Langage de type Pascal
 - ▶ À peu près de même niveau que le C
 - ▶ Langage LD (Ladder Diagram)
 - ▶ Langage graphique basé sur une analogie électrique
 - ▶ Langage FBD (Function Block Diagrams)
 - ▶ Langage graphique permettant de faire des assemblages de composants
 - ▶ Langage SFC (Sequential Function Chart)
 - ▶ Très inspiré du Grafcet

[Evaluer](#)

SIMATIC S7 et la norme CEI 61131-3

[Contribution](#) [Concerne le/les produits\(s\)](#)

QUESTION:
Avec quel langage SIMATIC S7 peut-on programmer des automates selon la norme CEI 61131-3?

REPONSE:
Veuillez trouver dans le tableau 1 ci-dessous, une vue d'ensemble:

Langage CEI 61131-3	Langage SIMATIC S7
IL - Liste d'Instruction	LIST (Couvre la plupart des fonctions définie dans la norme CEI 61131-3 - voir *)
ST - Texte Structuré	S7-SCL (Conforme au standard "PLCopen Basis Level" pour texte structuré de la norme CEI 61131-3)
FBD - Diagramme de blocs Fonctionnels	LOG (Couvre la plupart des fonctions définie dans la norme CEI 61131-3 - voir *)
LD - Ladder Diagramme ou langage à Contact	CONT (Couvre la plupart des fonctions définie dans la norme CEI 61131-3 - voir *)
SFC - Diagramme Fonctionnel en Séquence	S7-GRAFH (Conforme au standard "PLCopen Basis Level" pour texte structuré de la norme CEI 61131-3)

* : Des informations détaillées sont indiquées dans "norm_tab.wri" ou "norm_tbl.wri"

Tableau 1: Les langages de programmation SIMATIC S7 face à ceux de la CEI

Vous trouverez les fichiers "norm_tab.wri" (En Allemand) ou "norm_tbl.wri" (En Anglais) après avoir installé STEP7 sur contre console de programmation, sous la racine suivante ..\SIEMENS\STEP7.

Vous pouvez aussi télécharger les fichiers "norm_tab.wri" ou "norm_tbl.wri" pour STEP 7 V5.2 directement au format PDF en cliquant sur les liens ci-dessous:

[PDF norm_tab.pdf \(26 KB \)](#)

[PDF norm_tbl.pdf \(25 KB \)](#)