

**Document de formation
pour une solution complète d'automatisation
Totally Integrated Automation (T I A)**

Annexe I

**Base de la programmation d'API
avec SIMATIC S7-300**

Ce document a été édité par Siemens A&D SCE (Automatisierungs- und Antriebstechnik, Siemens A&D Cooperates with Education) à des fins de formation.
Siemens ne se porte pas garant de son contenu.

La communication, la distribution et l'utilisation de ce document sont autorisées dans le cadre de formation publique. En dehors de ces conditions, une autorisation écrite par Siemens A&D SCE est exigée (M. Knust: E-Mail: michael.knust@hvr.siemens.de).

Tout non-respect de cette règle entraînera des dommages et intérêts. Tous les droits, ceux de la traduction y compris, sont réservés, en particulier dans le cas de brevets ou de modèles déposés.

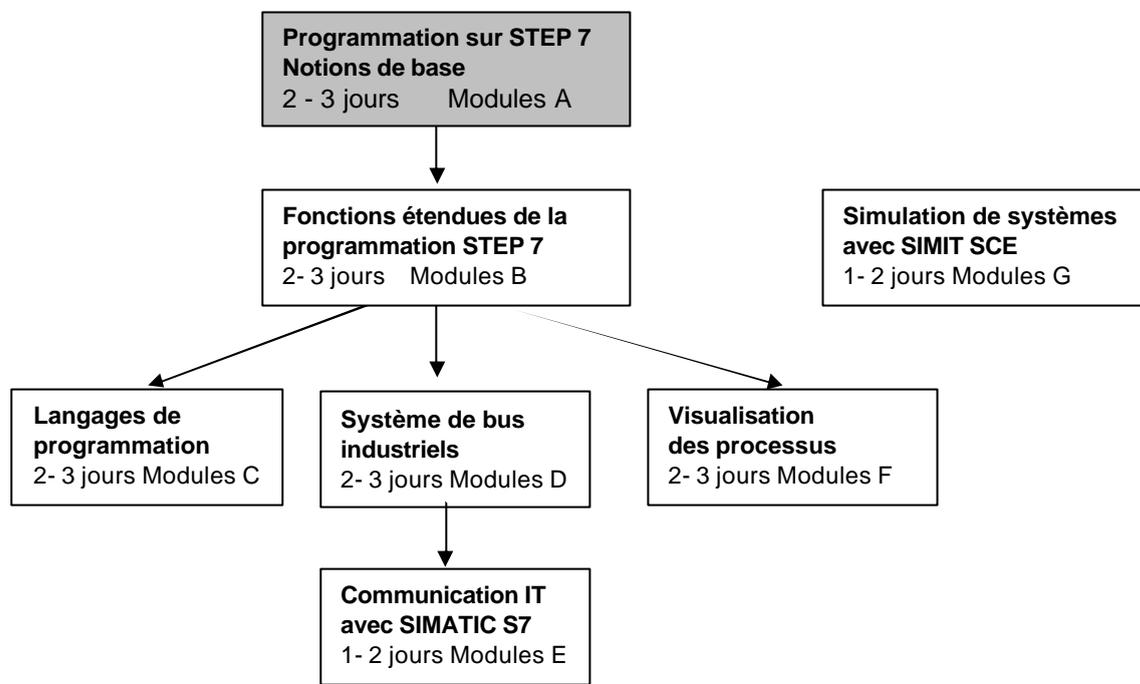
Nous remercions l'entreprise Michael Dziallas Engineering et les enseignants d'écoles professionnelles ainsi que tous ceux qui ont participé à l'élaboration de ce document.

		PAGE :
1.	Avant-propos	5
2.	Structure et fonction d'un API	6
2.1	Catégorie de signaux en technique de régulation	6
2.1.1	Signaux binaires	6
2.1.2	Signaux analogiques	8
2.2	Format de nombre	10
2.2.1	Système décimal	10
2.2.2	Système dual	11
2.2.3	Code BCD (Code 8-4-2-1)	11
2.2.4	Système hexadécimal	12
2.2.5	Représentation des systèmes de nombre	13
2.2.6	Règles de conversion	14
2.3	Notions du monde de l'informatique	16
2.3.1	Bit	16
2.3.2	Octet	16
2.3.3	Mot	16
2.3.4	Mot double	17
2.3.5	Adresse de bit	17
2.3.6	Adresse d'octet	17
2.3.7	Adresse de mot	18
2.3.8	Adresse de mot double	18
2.4	Structure d'un API	19
2.5	Fonctionnement de base d'un API	21
2.5.1	Le module central CPU	21
2.5.2	Le système à bus	22
2.5.3	Le module d'alimentation en courant	23
2.5.4	Mémoire programme	23
2.5.5	RAM	23
2.5.6	Memory Card (Flash- EPROM)	23
2.5.7	Micro Memory Card (MMC)	24
2.5.8	Retirer et enficher des cartes mémoire (Memory Card et MMC)	24
2.6	Système d'automatisation SIMATIC S7	25
2.6.1	SIMATIC S7-300	25

		PAGE :
2.7	Traitement du programme	32
2.7.1	Mémoire programme	32
2.7.2	Traitement linéaire du programme.....	32
2.7.3	Traitement structuré du programme	33
2.7.4	Blocs utilisateur.....	34
2.7.5	Blocs systèmes pour les fonctions standards et système.....	34
3.	Langage de programmation STEP7	35
3.1	Généralités sur le langage de programmation STEP7	35
3.2	Conversion STEP5 ↪ STEP7	35
3.3	La norme IEC 61131	36
3.4	Arborescence	37
3.5	Configurer et paramétrer.....	39
3.6	L'instruction de commande	40
3.6.1	Partie Opération.....	40
3.6.2	Partie Opérande.....	42
3.7	Adressage.....	43
3.7.1	Adressage mnémorique	43
3.7.2	Adressage inconditionnel.....	43
3.8	Représentation du programme.....	45
3.8.1	Schéma à contact CONT.....	45
3.8.2	Plan logique LOG.....	45
3.8.3	Liste d'instructions LIST	46
3.9	Mémoire interne	47
3.9.1	Mémoire interne rémanente.....	47
3.9.2	Mémoire interne non rémanente	47

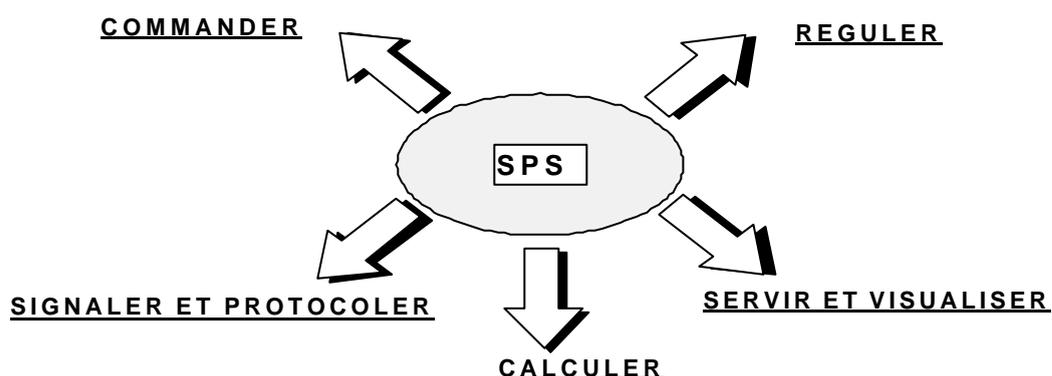
1. AVANT-PROPOS

L'annexe I est le pré-requis pour le traitement du module ,Fondements de la programmation STEP 7'.



Objectif :

Les automates à mémoire programmable, ou API , se sont aujourd’hui imposés comme la partie centrale de tout automate. Avec ces commandes, les tâches d’automatisation peuvent être exécutées économiquement suivant chaque problématique.



On donnera dans ces documents un aperçu sur le système d’automatisation SIMATIC S7-300 et le logiciel de programmation correspondant STEP 7.

Pré-requis :

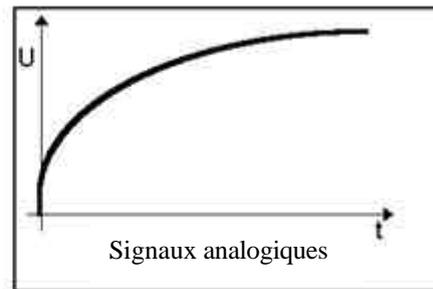
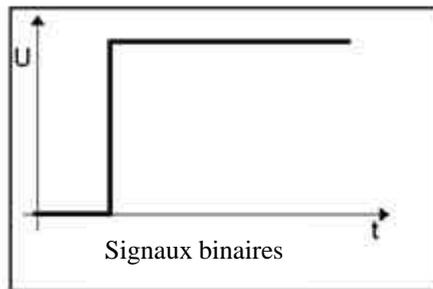
Aucun pré-requis particulier n’est nécessaire pour l’étude de cette annexe, car elle traite elle-même des bases.

2. STRUCTURE ET FONCTION D'UN API

La commande a pour tâche de mener des opérations individuelles d'une machine ou d'un dispositif d'après un déroulement prédéfini de fonctions dépendant de signaux.

2.1 CATEGORIE DE SIGNAUX EN TECHNIQUE DE REGULATION

Les signaux électriques se trouvant aux entrées et sorties peuvent être fondamentalement divisés en deux groupes distincts :



2.1.1 SIGNAUX BINAIRES

Les signaux binaires peuvent seulement prendre deux états de signal. Il s'agit de :

Etat de signal „1,,	=	Tension disponible	=	par ex. Interrupteur fermé
Etat de signal „0,,	=	Tension indisponible	=	par ex. Interrupteur ouvert

En technique des commandes, on utilisera souvent une tension continue de 24V comme « Tension de commande ». Un niveau de tension de + 24V sur une borne d'entrée correspond ainsi à l'état de signal „1,, pour cette entrée. De la même manière, 0V correspond à l'état du signal „0,,. En plus de l'état du signal, le classement logique des capteurs est aussi important. On doit savoir pour le capteur s'il s'agit d'un contact à ouverture ou d'un contact à fermeture. Un contact à ouverture délivre s'il est actif, c'est-à-dire lorsqu'il est actionné, l'état de signal „0,,. On désigne ce comportement commutateur par „actif 0,, ou „actif bas,, (low). De la même manière, un contact à fermeture qui est „actif 1,, / „actif haut,, (high) délivre un signal „1,, lorsqu'il est actionné.

En général, les signaleurs sont des „actifs 1,,. Une application typique pour un capteur „actif 0,, est celle du bouton d'arrêt d'urgence. Un bouton d'arrêt d'urgence est toujours conducteur en état non actionné (bouton de secours non appuyé), c'est-à-dire qu'il délivre à l'entrée connectée le signal „1,, (Fil rupture de sécurité). Si l'action du bouton d'arrêt d'urgence doit entraîner une réaction définie (par exemple, celle de fermer tous les clapets), alors celle-ci doit être déclenchée par l'état de signal „0,,.

Cases binaires :

Un signal binaire peut donc prendre seulement les deux valeurs (états de signal) „0,, ou „1,,. Un tel signal est aussi désigné sous le nom de binaire numérique et prend le nom de « bit » en langage technique. Plusieurs signaux binaires regroupés engendrent un signal numérique suivant un ordre défini. Alors qu'un signal binaire permet seulement la prise en compte d'une grandeur à deux valeurs (par ex. porte ouverte / porte fermée), on peut, par la représentation de cases binaires, construire par ex. un nombre ou un chiffre comme information numérique.

Le regroupement de n cases binaires permet la représentation de 2^n combinaisons différentes.

On peut ainsi par ex. avec 2 cases binaires soit 2×2 représenter 4 informations différentes :

0	0	Information1	(par ex. les deux interrupteurs sont ouverts)
0	1	Information2	(Commutateur 1 ouvert / Commutateur 2 fermé)
1	0	Information3	(Commutateur 1 fermé / Commutateur 2 ouvert)
1	1	Information4	(les deux interrupteurs sont fermés)

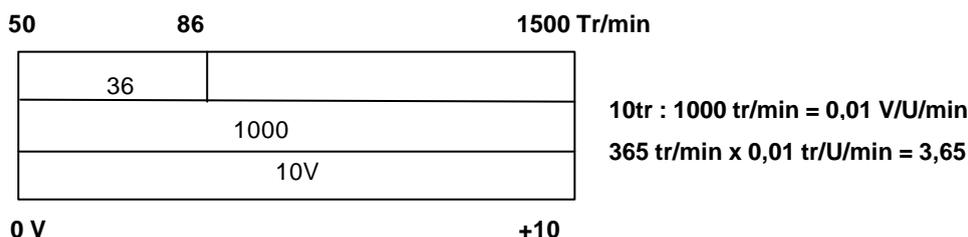
2.1.2 SIGNAUX ANALOGIQUES

Contrairement à un signal binaire, qui peut seulement prendre les deux états de signal „Tension disponible +24V,, et „Tension indisponible 0V,, les signaux analogiques peuvent prendre de nombreuses valeurs à l'intérieur d'un domaine donné. L'exemple typique de capteur analogique est le potentiomètre. Selon la position du curseur sur le potentiomètre, on peut lui faire prendre une résistance quelconque en-dessous de sa résistance maximale.

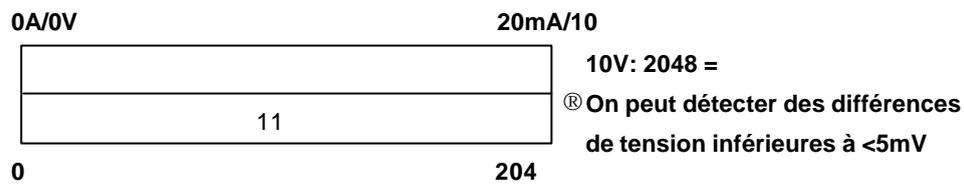
Exemples de grandeurs analogiques en technique des régulations :

- Température -50 ... +150°C
- Débit 0 ... 200l/min
- Nombre de tours 500 ... 1500 U/min
- etc.

Ces caractéristiques physiques sont converties en tensions électriques, courants ou résistances à l'aide d'un convertisseur de mesure. Si l'on doit déterminer par exemple un nombre de tours, on peut convertir un domaine de nombre de tours de 500 à 1500 tours/min dans un domaine de tension de 0 à +10V par un convertisseur de mesure. Si l'on mesure un nombre de tours de 865 tours/min, le convertisseur de mesure nous donne ainsi une tension de + 3,65 V.



Si des données analogiques sont traitées par un SPS, la valeur de tension/courant/impédance lue doit être convertie en une information numérique. Cette conversion est connue sous le nom de conversion analogique/numérique (CAN (A/D)). Ceci signifie qu'une valeur de tension valant par exemple 3,65V doit être traduite dans une série de chiffres binaires comme information. Plus on utilise de chiffres binaires pour la représentation numérique, meilleure est la résolution. Si on avait par exemple un seul bit à disposition pour le domaine de tension 0 ... +10V, on pourrait seulement dire que la tension mesurée se trouve dans le domaine 0 .. +5V ou dans +5V ... +10V. Avec 2 bits, on pourrait déjà diviser le domaine en 4 sous domaines individuels, soit 0 ... 2,5 / 2,5 ... 5 / 5 ... 7,5 / 7,5 ... 10V. Les convertisseurs A/N (A/D) usuels convertissent avec 8 ou 11 bits en technique des régulations. Les convertisseurs 8 bits ont 256 domaines élémentaires et les 11 bits une résolution de 2048 domaines élémentaires.

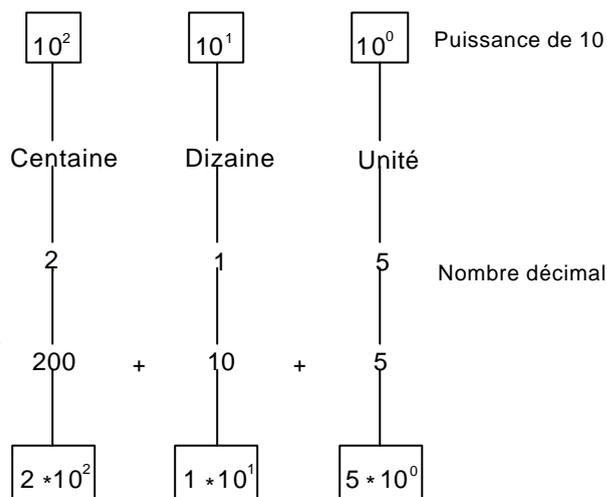


2.2 FORMAT DE NOMBRE

Pour le traitement des adresses de cellules mémoires, d'entrées, de sorties, de durées, de mémoires internes... etc, c'est le système dual qui est employé, et non pas le système décimal.

2.2.1 SYSTEME DECIMAL

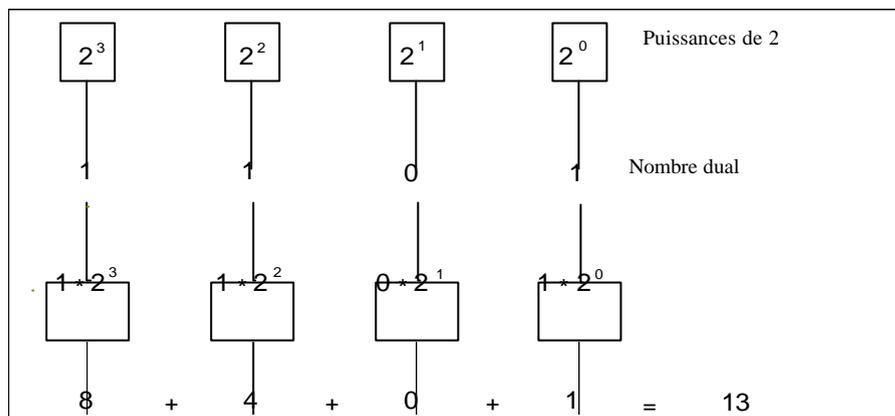
Pour comprendre le système dual, il est d'abord nécessaire d'étudier le système décimal. On souhaite représenter ici le nombre 215. Le 2 y représente les centaines, le 1 les dizaines et le 5 les unités. En fait, on doit écrire 215 comme ceci : $200+10+5$. Si l'on écrit l'expression $200+10+5$, comme représenté ci-dessous, à l'aide de puissances de 10, alors on constate que chaque chiffre à l'intérieur d'une case est attribué à une puissance de 10.



Chaque chiffre du nombre décimal est attribué à une puissance de 10.

2.2.2 SYSTEME DE NOMBRE DUAL

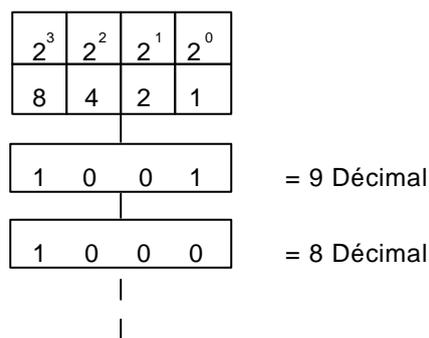
Le système de nombre dual ne connaît que les chiffres 0 et 1, qui sont facilement représentables et évaluables en traitement des données. Il s'agit donc d'un système de nombre binaire. Les valeurs case d'un nombre dual sont ordonnées suivant des puissances de 2, comme représentées ci-dessous.



Chaque chiffre à l'intérieur d'un nombre dual est attribué à une puissance de 2.

2.2.3 CODE BCD (CODE 8-4-2-1)

Pour pouvoir représenter de grands nombres de façon claire, on utilise souvent le code BCD (Binaire Codé Décimal). Le nombre décimal est représenté à l'aide du système de nombre dual. Le chiffre décimal représentable le plus haut est le 9. On a besoin pour la représentation du nombre 9 en case de puissance de 2 jusqu'à 2^3 , soit 4 cases au total.



Comme la représentation du plus grand chiffre décimal requiert 4 cases binaires, on utilise pour chaque chiffre décimal une unité de 4 cases, que l'on dénomme une tétrade. Le code BCD est donc un code 4 bits.

Chaque nombre décimal est codé de manière élémentaire. Le nombre 285 par ex. se compose de 3 chiffres décimaux. Chaque chiffre décimal est représenté dans le code BCD en unité de 4 cases (tétrade).

2	8	5
0010	1000	0101

Chaque chiffre décimal est codé à part et représenté par une tétrade.

2.2.4 SYSTEME DE NOMBRE HEXADECIMAL

Le système de nombre hexadécimal appartient aux systèmes des cases de valeurs. Chaque case emploie une puissance du nombre 16. Le système hexadécimal est donc un système en base 16. Chaque case à l'intérieur d'un nombre hexadécimal est attribuée à une puissance de 16. On a besoin au total de 16 chiffres, 0 compris. Pour les chiffres de 0 à 9, on emploie le système décimal, pour les chiffres de 10 à 15, on emploie les lettres A, B, C, D, E et F.

Chaque case à l'intérieur d'un nombre hexadécimal est attribuée à une puissance de 16.

2.2.5 REPRESENTATION DES SYSTEMES DE NOMBRE

Décimal	Dual					Hexadécimal
	16	8	4	2	1	
0					0	0
1					1	1
2				1	0	2
3				1	1	3
4			1	0	0	4
5			1	0	1	5
6			1	1	0	6
7			1	1	1	7
8		1	0	0	0	8
9		1	0	0	1	9
10		1	0	1	0	A
11		1	0	1	1	B
12		1	1	0	0	C
13		1	1	0	1	D
14		1	1	1	0	E
15		1	1	1	1	F
16	1	0	0	0	0	1 0
17	1	0	0	0	1	1 1
18	1	0	0	1	0	1 2
19	1	0	0	1	1	1 3



2.2.6 REGLE DE CONVERSION

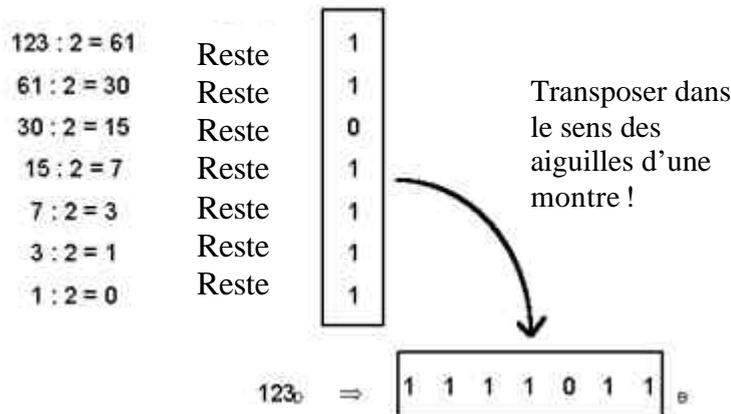
La conversion entre les différents systèmes de représentation est basée sur des règles simples. Celle-ci doit être contrôlée par l'utilisateur du SPS puisqu'elle va être souvent employée dans le cadre de cette technique. Pour identifier le système de nombre sur lequel est représenté un nombre, on va employer un indice se situant à droite du nombre. On utilise „D„ pour décimal, „B„ pour binaire (dual) et „H„ pour hexadécimal. Cette identification est souvent nécessaire car une suite de chiffres représente une valeur complètement différente suivant le système de nombre considéré.

Ainsi, par exemple „111„ en décimal correspond à la valeur 111_D (cent onze), en dual, 111_B correspond à la valeur décimale $7 (1 \times 2^0 + 1 \times 2^1 + 1 \times 2^2)$ et le nombre hexadécimal 111_H correspond à la valeur décimale $273 (1 \times 16^0 + 1 \times 16^1 + 1 \times 16^2)$.

Conversion Décimal → Dual

Les nombres décimaux entiers seront divisés par la base 2 jusqu'à obtenir le résultat zéro. Les restes (0 ou 1) successivement obtenus lors des divisions donnent le nombre dual. Il faut faire attention à l'ordre des restes. Le reste de la première division est le premier bit en partant de la droite (bit de poids faible).

Par ex. : On doit convertir le nombre décimal 123 en son nombre dual.



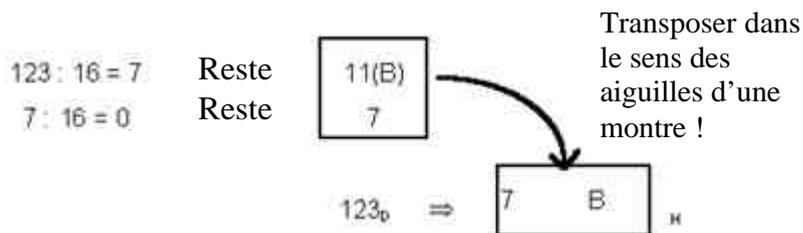
Vérification :

$$\begin{aligned}
 & 1 \quad 1 \quad 1 \quad 1 \quad 0 \quad 1 \quad 1 \\
 & 1 \times 2^6 + 1 \times 2^5 + 1 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 \\
 & 64 + 32 + 16 + 8 + 0 + 2 + 1 = \underline{123}
 \end{aligned}$$

Conversion Décimal → Hexadécimal

Cette conversion s'effectue exactement comme la Décimal → Dual. La différence se situe au niveau de la base employée : ici la base 16 (au lieu de la base 2 dans l'autre cas). On doit donc diviser par 16 au lieu de diviser par 2.

Par ex, le nombre décimal 123 doit être converti en sa représentation hexadécimale.



Vérification :

$$\begin{array}{r}
 7 \qquad \qquad \qquad B \\
 7 \times 16^1 + \quad 11 \times 16^0 \\
 112 \quad + \quad 11 \qquad = \quad \underline{123}
 \end{array}$$

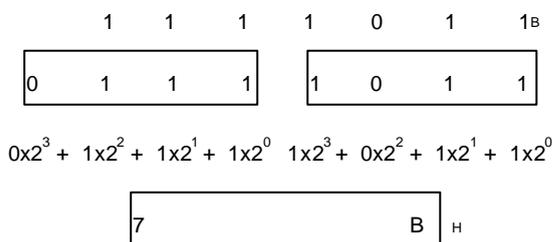
Conversion Dual → Hexadécimal

Pour la conversion d'un nombre dual en un nombre hexadécimal, on pourrait tout d'abord déterminer la valeur décimale du nombre (Addition des valeurs). On pourrait ensuite successivement diviser ce nombre décimal par 16 afin de le convertir en hexadécimal.

Mais il y a aussi la possibilité de convertir directement un nombre dual en sa valeur hexadécimale correspondante.

Pour cela, on va tout d'abord répartir le nombre dual en groupes de 4, en commençant par la droite. Chacun de ces groupes de 4 donne ensuite un nombre hexadécimal. Si nécessaire, on complète les bits manquants du groupe le plus à gauche par des zéros.

On veut par ex. convertir le dual 1111011 directement en un nombre hexadécimal.



2.3.4 DOUBLE MOT

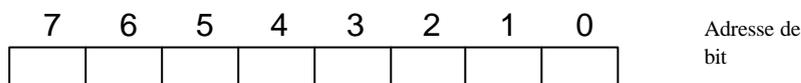
Un double mot a une longueur de mot de 32 chiffres binaires.

Un double mot a donc la taille de 2 mots, de 4 octets, ou encore de 32 bits.

D'autres unités existent : Kilobit ou Kiloctet pour 2^{10} , donc 1024 bits resp. 1024 octets et le Megabit ou Megaoctet pour 1024 Kilobit resp. 1024 Kiloctet.

2.3.5 ADRESSE DE BIT

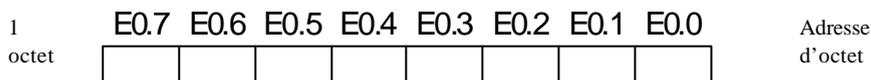
Afin qu'on puisse accéder aux bits élémentaires, chaque bit d'un octet est attribué à un chiffre, l'adresse de bit. On numérote ainsi l'adresse du bit le plus à droite 0. En incrémentant de 1 vers la gauche, on numérote les autres bits, pour arriver à l'adresse de bit 7 tout à gauche.



2.3.6 ADRESSE D'OCTET

Même les octets élémentaires reçoivent des numéros, les adresses d'octet. De plus, l'opérande est aussi identifié, afin que par ex. EB 2 représente l'octet d'entrée 2 ou que AB 4 représente l'octet de sortie 4.

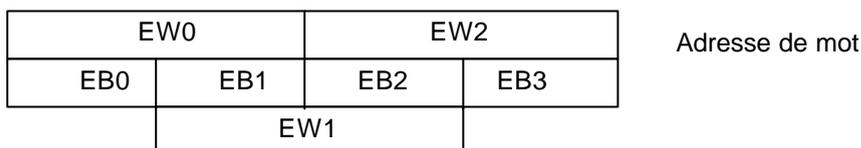
Les bits élémentaires sont adressés de manière unique par la combinaison d'adresse de bit et d'octet. L'adresse de bit est séparée de l'adresse d'octet par un point. A droite du point se trouve l'adresse de bit, à gauche de celui-ci l'adresse de l'octet.



2.3.7 ADRESSE DE MOT

Le numérotage des mots donne l'adresse de mot.

Indication : En employant des mots, par ex. des mots d'entrées (EW), des mots de sortie (AW), des mots de mémoire interne (MW) etc., l'adresse du mot est toujours la plus petite adresse des 2 octets correspondants. Par ex, un mot se composant de EB2 et de EB3, a pour adresse EW2.

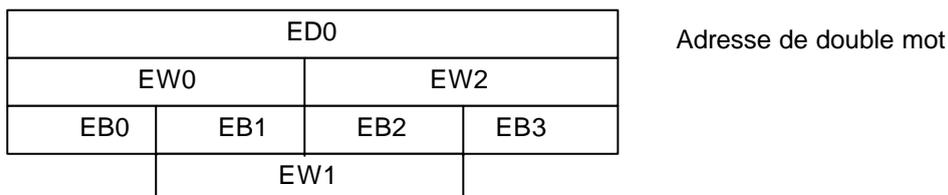


Indication : Lors du traitement de mot, il faut faire particulièrement attention : par ex. le mot d'entrée 0 et le mot d'entrée 1 se recourent dans un octet. En outre, on commence à compter les bits en commençant par la droite. Par ex., le bit 0 de EW1 est E2.0. Le bit 1 de E2.1. ... Le bit 7 de E 2.7. Le bit 8 de E1.0. ... Le bit 15 de E1.7. Entre les bits 7 et 8, il y a ainsi un « saut ».

2.3.8 ADRESSE DE DOUBLE MOT

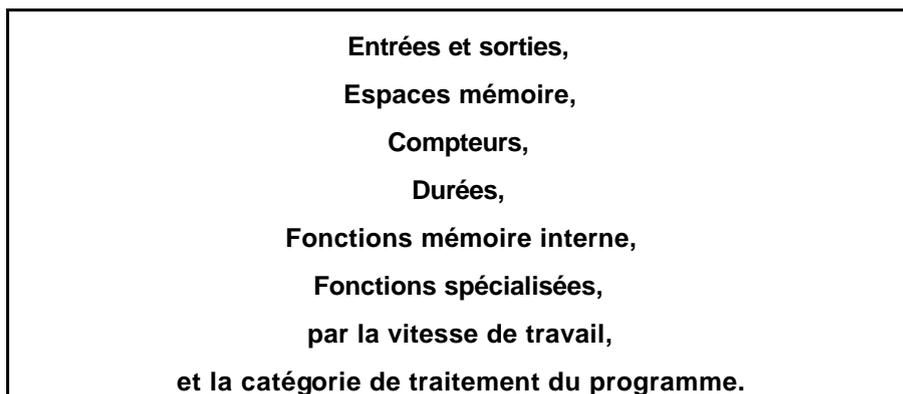
Le numérotage des doubles mots donne l'adresse des doubles mots.

Indication : En employant des mots doubles, par ex ED, AD, MD etc. l'adresse de mot double est toujours l'adresse de mot la plus petite des deux mots correspondants.



2.4 STRUCTURE D'UN API

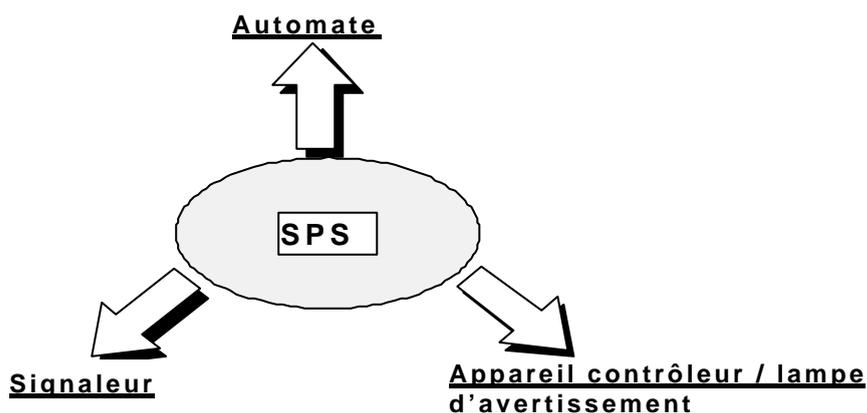
Les automates à commande programmable sont des appareils fabriqués en série, qui sont conçus de manière générique. Tout ce qui est nécessaire à la technique des régulations, les éléments logiques, les fonctions mémoire, les temporisateurs, les compteurs... etc est intégré à l'origine dans l'appareil par le constructeur. Ces éléments, pour la programmation, feront partie de la commande fonctionnelle. Les appareils de commande sont proposés en différentes unités de fonction. Elles se différencient notamment par le nombre possible de :



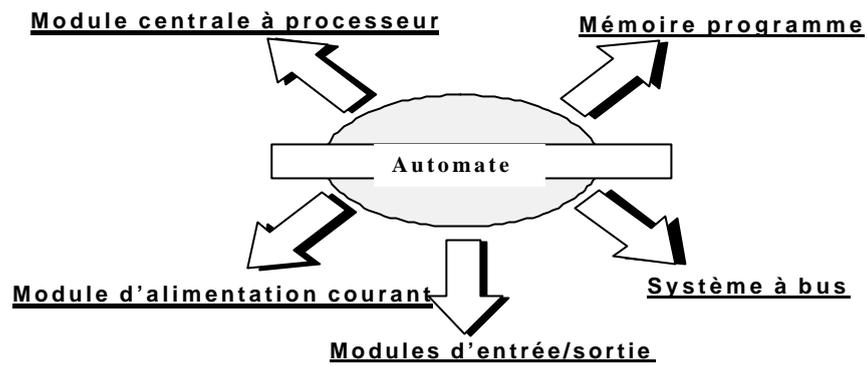
Les appareils de commande importants sont organisés en blocs individuels de façon modulaire. Avec ces systèmes modulaires vous pouvez adapter vos applications pratiques à partir des équipements de base, en assemblant des systèmes SPS.

Pour les petites tâches d'automatisation, des appareils de commande conçus de manière compacte sont proposés. Ils se présentent sous la forme d'unités fermées avec un nombre fixe prédéfini d'entrées et de sorties.

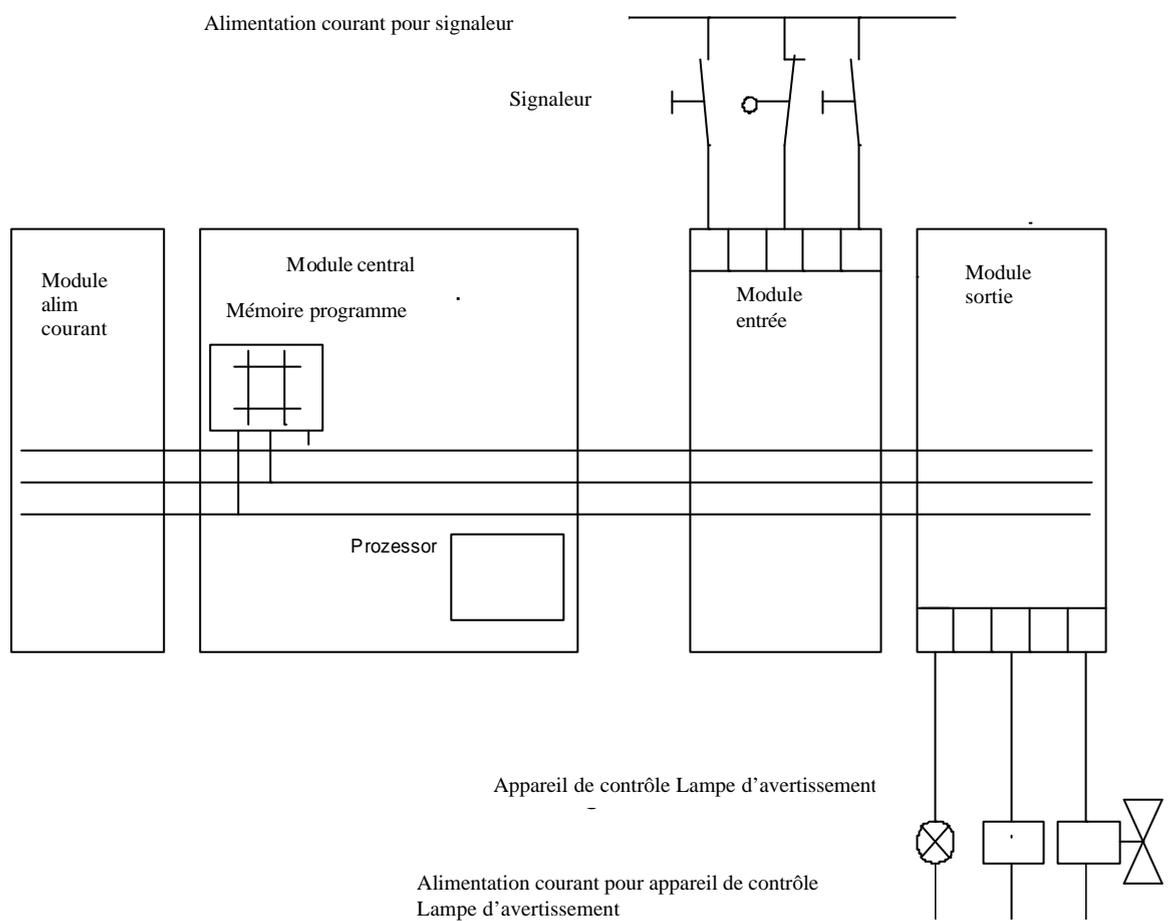
Les éléments suivants appartiennent à un automate à mémoire programmable :



L'automate se compose pour l'essentiel de :



Conception d'une commande à mémoire programmable :

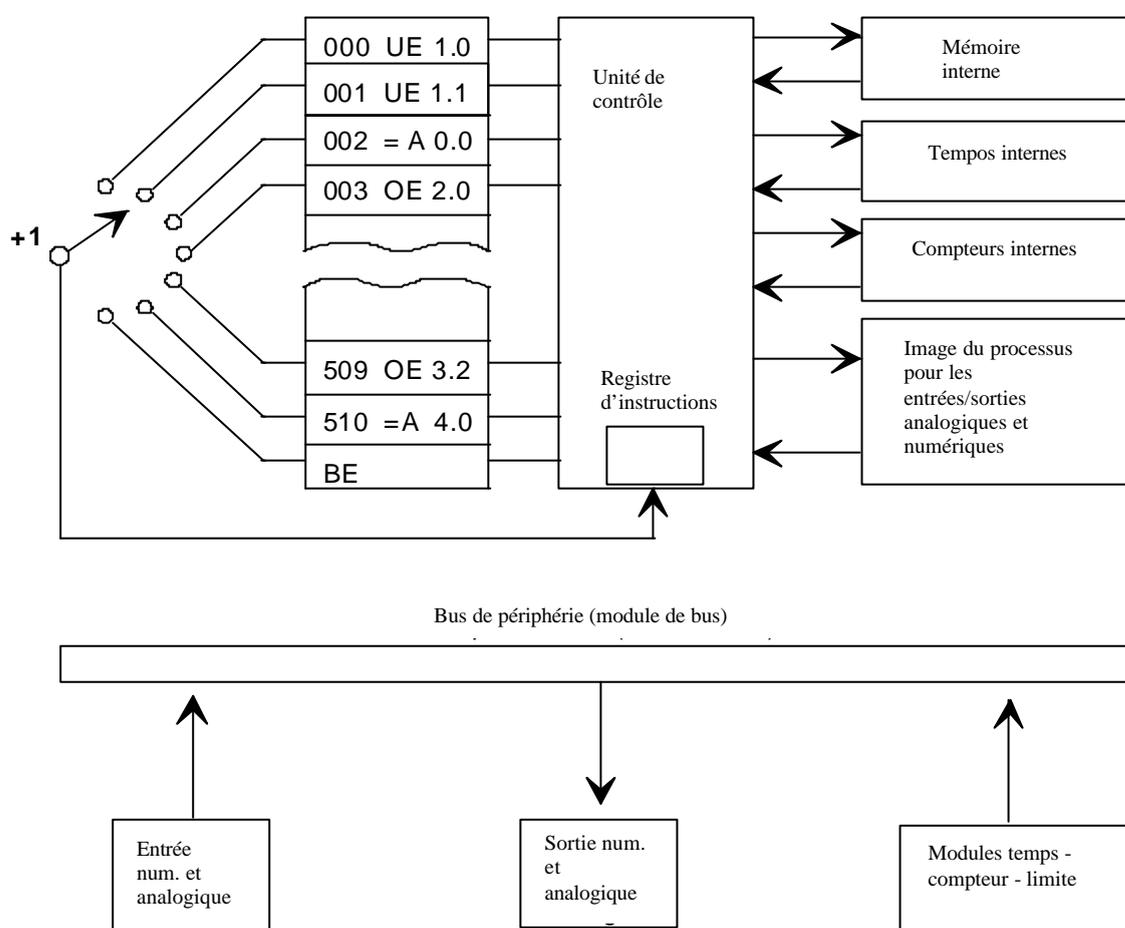


2.5 FONCTIONNEMENT DE BASE D'UN API

2.5.1 LE MODULE CENTRAL CPU

La tension venant du signaleur est connectée sur la barrette de connexion du module d'entrée. Dans la CPU (Module central), le processeur traite le programme se trouvant dans la mémoire et interroge les entrées de l'appareil pour savoir si elles délivrent de la tension ou pas. En fonction de cet état des entrées et du programme se trouvant en mémoire, le processeur ordonne au module de sortie de commuter sur le connecteur de la barrette de connexion correspondante. En fonction de l'état de tension sur les connecteurs des modules de sortie, les appareils à positionner et les lampes indicatrices sont connectés ou déconnectés.

Unité centrale (CPU) de l'API:



Le compteur d'adresses interroge les instructions du programme mémoire les unes après les autres (série) et provoque la transmission d'informations dépendantes du programme depuis la mémoire programme au registre d'instructions. On appelle usuellement les mémoires d'un processeur les registres.

Le système de commandes obtient ses instructions du registre d'instructions. Pendant que le système de commande traite les instructions en cours, le compteur d'adresse décale la prochaine instruction dans le registre d'instruction. A la transmission du statut des entrées dans l'image des processus des entrées (PAE) se succèdent les opérations logiques, la mise en place des membres temporels, les compteurs, les accumulateurs et la transmission du résultat de l'opération logique (VKE) à l'image des processus des sorties (PAA). Si après l'exécution du programme utilisateur, la fin du bloc (BE) est reconnue, alors la transmission de chaque statut a lieu depuis le PAA aux sorties.

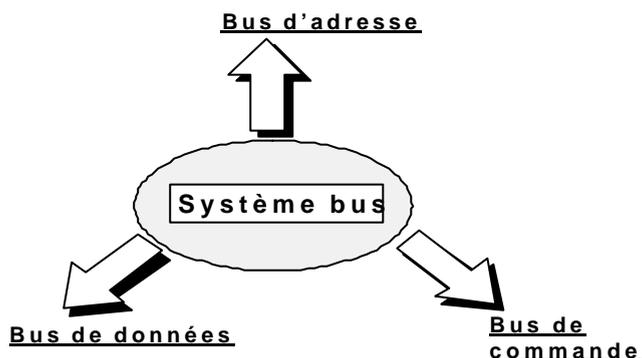
Le bus de périphérie exécute l'échange de données entre le module central et la périphérie.

Les modules numériques d'entrée/sortie, les modules analogiques d'entrée/sortie et les modules de temporisation, de compteur et de frontière appartiennent à la périphérie.

2.5.2 SYSTEME A BUS

Le système à bus est un câble collecteur de signaux pour la transmission. L'échange de signaux dans les automates s'effectue ainsi entre le processeur et les modules d'entrée et de sortie par un système à bus de processus.

Le bus est composé de 3 câbles de signaux parallèles :



- Les modules élémentaires correspondent aux adresses par le bus d'adresse..
- Avec le bus de données, les données sont transférées par exemple des modules d'entrée vers les modules de sortie.
- Les signaux de commande et de surveillance du déroulement des fonctions à l'intérieur de l'appareil d'automatisation sont transmis par le bus de commande.

2.5.3 LE MODULE D'ALIMENTATION EN COURANT

Le module d'alimentation en courant produit, depuis la tension secteur, la tension pour les modules électroniques de l'appareil d'automatisation. La valeur de cette tension vaut 24 Volts. Les tensions pour les signaux, les appareils de contrôle et les lampes d'avertissement, qui marchent sous du 24 Volt, nécessitent un appareil secteur et un transformateur de commande.

2.5.4 MEMOIRE PROGRAMME

Les éléments mémoire sont des éléments de conception, dans lesquels des informations sous la forme de signaux binaires peuvent être stockées (mémorisées).

La mémoire programme est le plus souvent réalisée en mémoire semi-conducteur. Une mémoire se compose de cellules de mémoire au nombre de 512, 1024, 2048 etc. Il est courant d'exprimer la capacité (c'est-à-dire le nombre de cellules mémoire) de la mémoire programme en multiples de 1 K (1 K correspond ici à 1024).

Dans chaque cellule mémoire, on peut écrire (programmer) une instruction de commande à l'aide d'un appareil programmable. Chaque élément binaire d'une cellule mémoire peut prendre le signal d'état "1" ou "0".

2.5.5 RAM

On désigne par RAM une mémoire en écriture et lecture, réalisée par la technique des semi-conducteurs. Les différents emplacements mémoire sont identifiés par des adresses, à l'aide desquelles on peut accéder à ces espaces mémoire.

Dans les cellules mémoires, on peut souvent écrire des informations facultatives. Les informations en sont extraites et ceci sans pour autant perdre le contenu de l'information.

Les mémoires RAM sont pourtant volatiles, par exemple, les informations seront perdues en cas de coupure de la tension d'alimentation. La mémoire RAM est effacée électriquement.

Le programme interne et également la mémoire de travail d'un SIMATIC S7-300 sont précisément de la mémoire RAM. La batterie de mémoire tampon, qu'il est possible de mettre en place dans le SPS, sert à la sécurisation de cette mémoire. Les nouvelles CPU compactes sont des exceptions à ce système. Dans celles-ci, les programmes ne sont plus stockés dans la RAM, mais dans des Micro Memory Card enfichables (MMC) (cf. chapitre 2.5.7). La mémoire de travail total y est mise en tampon. On n'a ainsi plus besoin de la batterie.

2.5.6 CARTE MEMOIRE (MEMORY CARD) (FLASH- EPROM)

EPROM est l'abréviation pour mémoire morte EFFACABLE, PROGRAMMABLE. Le contenu de l'EPROM est effaçable dans sa totalité par la lumière ultra violet ou par une tension, et ensuite de nouveau programmable. Elle est ainsi bien adaptée au transport de données sans perte.

Dans le SIMATIC S7- 300, vous avez la possibilité de sécuriser votre programme avec le PG sur une dénommée Memory Card (Flash-EPROM) et ainsi de remettre rapidement votre système en fonctionnement après un manque de tension.

Une tension de 5V est nécessaire pour effacer et écrire sur cette Flash- EPROM. Cela peut donc se produire avec un PG ou avec STEP 7, si elle est branchée à la CPU.

2.5.7 CARTE MÉMOIRE MICRO (MICRO MEMORY CARD) (MMC)

Dans les nouvelles CPU compactes, les programmes, qui étaient avant stockés dans la mémoire RAM, sont désormais enregistrés dans les Micro Memory Card (MMC) enfichables.

La mémoire de chargement total se trouve sur cette Micro Memory Card (MMC).

Les MMC existent avec des tailles de mémoire différentes. Ils servent à la réception de blocs de code et de données ainsi que de données systèmes (Configuration, Liaisons, paramètre de modules, etc.).

En outre, toutes les données de planification d'un projet peuvent être stockées sur un MMC.

La mémoire vive toute entière est ici également mise en mémoire tampon. Par contre, ici, aucune batterie de mémoire tampon n'est nécessaire.

La MMC peut être écrite avec un PG ou, connectée dans la CPU, avec STEP 7.

Indication : S'il n'y a pas de MMC branchée sur la CPU, alors la CPU n'est pas en mesure de marcher (il n'y a pas de mémoire chargement disponible). Un fonctionnement n'est possible qu'après le branchement d'une MMC et un formatage.

2.5.8 RETIRER ET ENFICHER DES MEMORY CARD ET MICRO MEMORY CARD (MMC)

La CPU détecte le fait de retirer et de brancher une carte mémoire (Memory Card) dans chaque état de fonctionnement.

Déroulement lors du débranchement :

1. La CPU doit être commutée sur STOP.
2. Aucune fonction PG d'écriture ne doit être active (par ex. Le chargement de blocs)
3. Après le débranchement de la carte mémoire, la CPU sollicite le formatage

Déroulement lors du branchement :

La connexion d'une carte mémoire (Memory Card) avec le programme utilisateur correspondant se déroule suivant les étapes suivantes :

1. Branchez la carte mémoire
2. La CPU sollicite le formatage
3. Confirmez le formatage
4. Démarrez la CPU

2.6 SYSTEME D'AUTOMATISATION SIMATIC S7

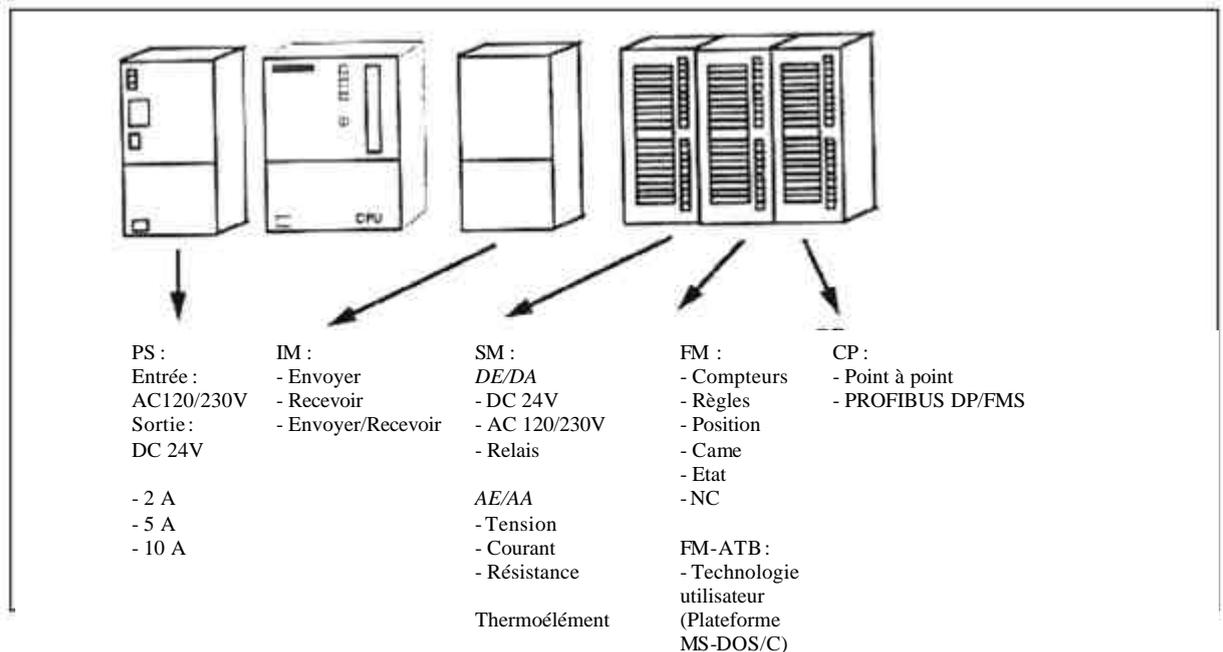
SIMATIC S7 est une désignation de produit de l'entreprise SIEMENS et s'applique pour la série actuelle des automates à mémoire programmable.

Cette famille de systèmes SIMATIC S7 est un module de la solution complète d'automatisation pour les techniques de production et de processus : Totally Integrated Automation.

2.6.1 SIMATIC S7 - 300

Spectre modulaire :

Les différentes catégories de modules suivantes forment, associées à une CPU, la structure du SIMATIC S7-300 :



Une sélection de CPU :

On va maintenant lister des CPU adaptées à des fins de formation. Il y a toutefois d'autres CPU à meilleur fonctionnement de qualité et il sort toujours de nouvelles CPU avec des puissances de calcul plus importantes. Celles-ci ne sont toutefois pas nécessaires pour traiter les programmes qui ne sont pas trop demandeurs en mémoire dans cette formation.

CPU 312 IFM	CPU 313	CPU 314 IFM	CPU 314	CPU 315	CPU315-2DP
2K Instructions 6Koctets Mémoire de travail 20Koctets mémoire charg.	4K Instructions 12 Koctets Mémoire de travail 20Koctets mémoire charg.	8K Instructions 24 Koctets Mémoire de travail 40Koctets mémoire charg.	8K Instructions 24 Koctets Mémoire de travail 40Koctets mémoire charg.	16K Instructions. 48 Koctets Mémoire travail 80Koctets mémoire charg.	16K Instructions. 48 Koctets Mémoire de travail 80K octets mémoire charg.
128 octets DE/DA	128 octets DE/DA	512 octets DE/DA	512 octets DE/DA	1024 octets DE/DA	1024 octets DE/DA
32 octets AE/AA	32 octets AE/AA	64 octets AE/AA	64 octets AE/AA	128 octets AE/AA	128 octets AE/AA
0,6 ms / 1K Commandes	0,6 ms / 1K Commandes	0,3 ms / 1 K Commandes	0,3 ms / 1K Commandes	0,3 ms / 1K Commandes	0,3 ms / 1K Commandes
1024 Bits mémoire interne	2048 Bits mémoire interne	2048 Bits mémoire interne	2048 Bits mémoire interne	2048 Bits mémoire interne	2048 Bits mémoire interne
32 Compteurs	64 Compteurs	64 Compteurs	64 Compteurs	64 Compteurs	64 Compteurs
64 Durées	128 Durées	128 Durées	128 Durées	128 Durées	128 Durées
10 DE/6DA embarqués, dont 4 DE pour des alarmes de processus ou des fonctions intégrées Compteurs rapides EPROM 20Koctets intégré		20 DE/16DA embarqués, dont 4 DE pour des alarmes de processus ou des fonctions intégrées Compteurs rapides 4 AI / 1AA embarqués Résolution : 11Bits + signe EPROM 40Koctets intégré			PROFIBUS DP intégré Connexion (Maître / Esclave) Attribution des adresses paramétrables

CPU compactes :

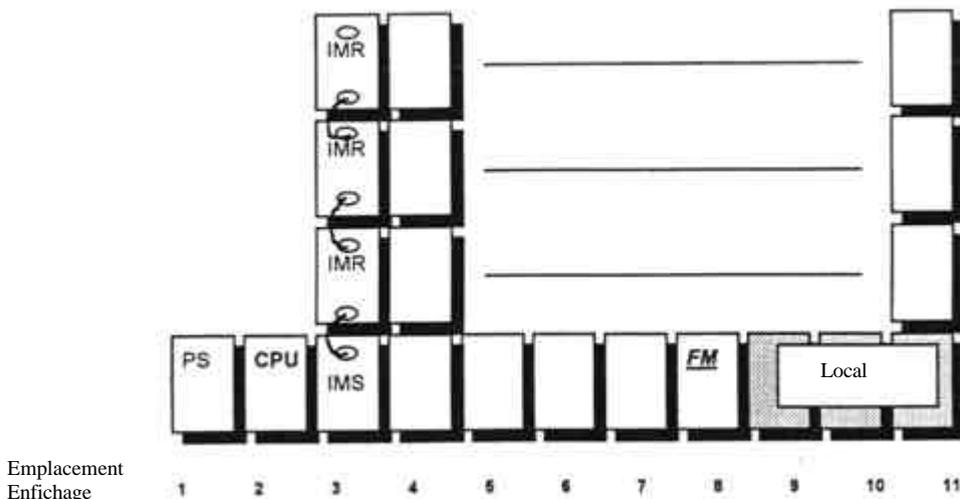
Il y a également un certain nombre de CPU compactes qui sont parfaitement adaptées à la formation du fait de leurs conceptions plus compactes, leurs prix particulièrement bon marché et leurs interfaces de communication éventuelles intégrées.

Celles-ci sont comparables à la fois dans leur fonctionnalité et dans leur maniement avec les CPU standards.

Le tableau suivant présente une sélection de CPU compacts..

	CPU 312C	CPU 313C	CPU 313C-2DP	CPU 314C-2DP
Durée de la commande (Binaire/flottant/référence)	0.2 / 4 / 40µs	0.1 / 2 / 20µs	0.1 / 2 / 20µs	0.1 / 2 / 20µs.
Mémoire de travail	16 KB	32 KB	32 KB	48 KB
Carte mémoire micro SIMATIC	De 64 KB à 4 MB	De 64 KB à 4 MB	De 64 KB à 4 MB	De 64 KB à 4 MB
Périphérie embaquée (On board) DE/DA AE/AA *PT100	10 / 6 -/-	24 / 16 4+1*/2	16 / 16 -/-	24 / 16 4+1* / 2
Fonctions technologiques - Compter/Fréquence - Sorties d'impulsion - Régulation - Positionnement	2 (10KHz) 2 (2,5 KHz) non non	3 (30KHz) 4 (2,5 KHz) oui non	3 (30KHz) 4 (2,5 KHz) oui non	4 (60KHz) 4 (2,5 KHz) oui 1
Interfaces - MPI 187,5 kBaud - Fonctionnalité DP	oui non	oui non	oui Maître/Esclave	oui Maître/Esclave

Possibilité de conception d'un SIMATIC S7 - 300



L'image montre la structure maximale d'une SIMATIC S7-300/CPU314. La conception permet la mise en place jusqu'à 32 modules (CPU 312IFM, CPU 312C et CPU 313 seulement 8 modules), pour chaque rack (Ligne) jusqu'à 8 modules.

En ce qui concerne les emplacements d'enfichage, on observera les règles suivantes :

- Poste d'enfichage 1 : Alimentation courant PS (= Attribution fixe)
- Poste d'enfichage 2 : Module central CPU (= Attribution fixe)
- Poste d'enfichage 3 : Module couplé IM (= Attribution fixe)
- Poste d'enfichage 4-11 : Modules de signal SM, Modules de fonction FM et processeurs de communication CP (= Pas d'attribution fixe)

Le bus est bouclé entre les racks par les modules de connexion (IM 361). La connexion IMS représente l'expéditeur, et l'IMR le récepteur. Les modules de connexion doivent être mis en place aux emplacements d'enfichage donnés.

Dans les racks d'extension, il faut aussi prévoir une alimentation courant supplémentaire.

Comme variante bon marché pour une conception biligne, on trouve le module de connexion IM 365 câblé de manière fixe avec lequel il n'y a pas besoin d'alimentation courant supplémentaire.

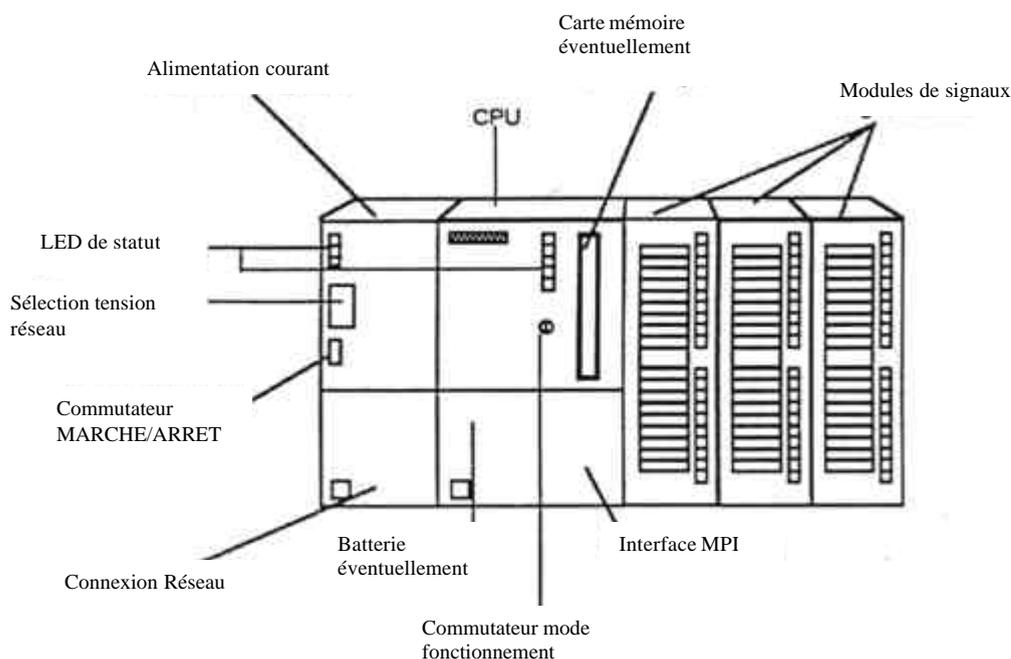
Les caractéristiques suivantes de câbles entres les lignes sont à disposition :

- conception bi-ligne avec IM 365 1m max.
- conception multiligne avec IM 361 de 10m max.

Les modules de fonction, par ex. SINUMERIK FM NC, peuvent être attribués à quelques périphéries. De cette manière, un module FM a son propre domaine de périphérie et peut y accéder rapidement. On appelle ce domaine de périphérie un segment local.

Il existe un segment local par rack. Il n'est donc plus possible d'accéder à ces périphéries par la CPU lorsque le système est en marche.

Éléments importants de l'alimentation en tension et de la CPU :

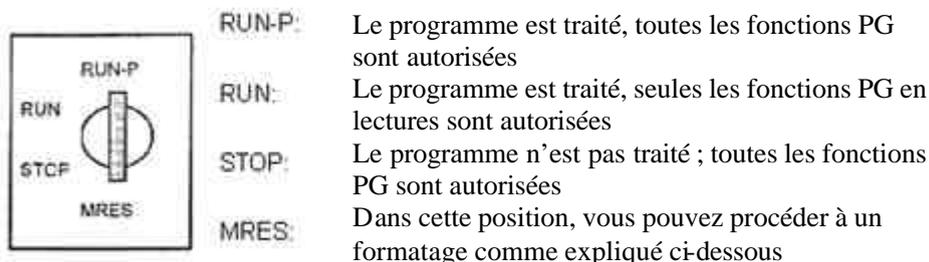


Indicateurs de statut et d'erreur de la CPU :

Indicateurs	Signification	Explication
SF (rouge)	Liste d'erreurs	Les modules capables de diagnostiquer affichent ici une liste d'erreurs.
BF (rouge)	Erreur de bus (si disponible)	Indique les perturbations du PROFIBUS DP
BAF (rouge)	Erreur de batterie	Indique si la batterie mémoire tampon ne délivre pas assez de tension ou carrément pas du tout
DC5V (vert)	Alimentation DC5V pour la CPU et le bus panneau arrière	Indicateur pour l'alimentation interne fonctionnelle 5V de la CPU
FRCE (vert)	Forçages	Indicateurs d'état de la CPU, dans les entrées et les sorties par une fonction de test de forçage de commande
RUN (vert)	Etat de fonctionnement RUN	Clignote lors du démarrage de la CPU – Fixe lorsque la CPU est dans l'état Run
STOP (orange)	Etat de fonctionnement STOP	Clignote lorsque le formatage est demandé – Fixe lorsque la CPU est à l'état arrêt

Concept de protection de la CPU:

Chaque CPU possède un commutateur de mode de fonctionnement pour commuter entre les différentes catégories de fonctionnement. Celui-ci est la plupart du temps conçu sous la forme d'un commutateur à clé, qui peut être placé sur les modes de fonctionnement RUN et STOP. Les modes de fonctionnement suivants sont possibles :



Indication : Pour les CPU des séries CPU31xC, ce commutateur est conçu sous forme d'interrupteur à bascule. Le mode de fonctionnement Run à protection de lecture manque dans ce cas. Ici aussi, les fonctions PG en écriture sont autorisées.

Le concept de protection pour la SIMATIC S7-300 vous permet de protéger des parties définies du système d'automatisation d'accès non autorisés. Cela concerne :

- les CPU et tous les modules programmables
- tous les objets (comme par ex. les blocs)

Ces parties à protéger peuvent être influencées par les appareils PG ou B&B.

Le concept de protection de la CPU est réparti en trois niveaux de protection. Ce paramétrage de niveaux de protection définit ce qui est autorisé pour un utilisateur :

- Etape 1 – Position commutateur à clé Run-P ou Stop : pas de protection, toutes les fonctions sont autorisées.
- Etape 2 – Position commutateur à clé Run (**Pas pour les CPU31xC !**) : Les fonctions protégées en écriture, les fonctions en lecture sont autorisées, par ex les fonctions de visualisation, les fonctions d'informations, le chargement depuis la CPU.
- Etape 3 – Paramétrage par la configuration S7 (Protection par mot de passe). Seules sont autorisées les fonctions de visualisation et d'information si l'utilisateur ne connaît pas le mot de passe.

Formatage de la CPU:

Avec le commutateur de mode de fonctionnement, on peut aussi procéder au formatage de la CPU :

Etape	Exécution	Résultat
1	Placez le commutateur de mode de fonctionnement sur STOP .	Indicateur STOP s'allume
2	Placez le commutateur en mode MRES et maintenez le dans cette position (environ 3 secondes) jusqu'à ce que l'indicateur STOP se rallume	L'indicateur STOP s'éteint et après environ 3 secondes il s'allume de nouveau. Dans les nouvelles CPU, attendre que l'indicateur STOP s'allume pour la deuxième fois. Important : Entre l'étape 2 et l'étape 3, il doit se passer 3 secondes max..
3	Placez le commutateur de nouveau en position STOP et finalement remettez le sur la position MRES avant que 2 secondes ne se soient écoulées	L'indicateur STOP clignote pendant environ 3 secondes et s'allume ensuite de nouveau : tout s'est bien passé ; la CPU est formatée

2.7 TRAITEMENT DU PROGRAMME

2.7.1 MEMOIRE PROGRAMME

Lors du traitement du programme, il y a deux possibilités différentes, dépendantes de l'appareil de commande utilisé et de la programmation.

Le traitement de chaque instruction nécessite un certain temps (de l'ordre de la microseconde). On appelle durée de cycle, la durée d'un (seul) traitement de toutes les instructions. C'est le temps d'un cycle programme.

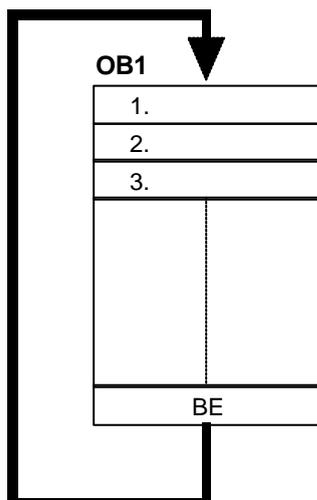
2.7.2 TRAITEMENT LINEAIRE DE PROGRAMME

Lors de la programmation linéaire, les instructions de l'appareil de commande sont traitées les unes après les autres, telles qu'écrites dans la mémoire programme. Si la fin du programme (BE) est atteinte, le traitement du programme recommence du début.

On parle de traitement cyclique.

La durée, nécessaire à un appareil pour une itération de traitement, est appelée durée de cycle.

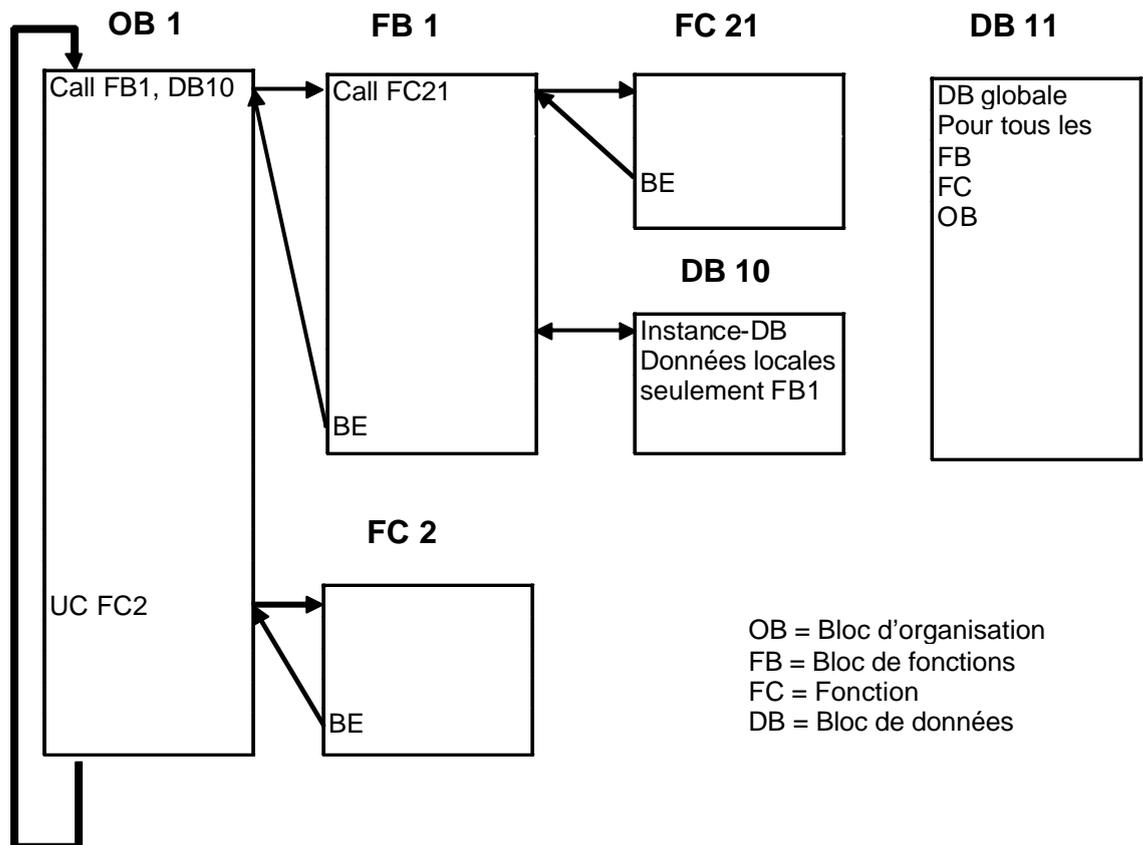
Le traitement linéaire du programme est la plupart du temps utilisé pour des commandes simples et pas trop volumineuses. Il peut être réalisé dans un seul OB.



2.7.3 TRAITEMENT STRUCTURE DU PROGRAMME

On répartit le programme d'un ensemble volumineux de tâches de commande en blocs de programmes petits, clairs, associés à des fonctions. Cela présente l'avantage de pouvoir tester les blocs de manière individuelle et de les faire fonctionner ensemble par une fonction globale.

Les blocs de programme doivent être appelés par des commandes d'appel de blocs (Call xx / UC xx / CC xx). Si la fin du bloc est reconnue, le programme qui a appelé le bloc est de nouveau traité.



2.7.4 BLOCS UTILISATEUR

STEP 7 offre pour la programmation structurée les blocs utilisateur suivants :

- **OB (Bloc Organisation) :**
Un OB est appelé cycliquement par le système d'exploitation et réalise ainsi l'interface entre le programme utilisateur et le système d'exploitation. Le dispositif de commande est informé dans cet OB par des commandes d'appel de blocs, de quels blocs de programme il doit traiter.
- **FB (Bloc de fonction) :**
Le FB est à disposition via un espace mémoire correspondant. Si un FB est appelé, il lui est attribué un bloc de données (DB). On peut accéder aux données de cette instance DB par des appels depuis le FB. Un FB peut être attribué à différents DB. D'autres FB et d'autres FC peuvent être appelés dans un bloc de fonction par des commandes d'appel de blocs.
- **FC (Fonction) :**
Une FC ne possède pas un espace mémoire attribué. Les données locales d'une fonction sont perdues après le traitement de la fonction. D'autres FB et FC peuvent être appelés dans une fonction par des commandes d'appel de blocs.
- **DB (Bloc de données) :**
Les DB sont employés afin de tenir à disposition de l'espace mémoire pour les variables de données. Il y a deux catégories de blocs de données. Les DB globaux où tous les OB, FB et FC peuvent lire des données enregistrées et écrire eux-mêmes des données dans le DB. Les instances DB sont attribuées à un FB défini.

2.7.5 BLOCS SYSTEME POUR LES FONCTIONS STANDARDS ET LES FONCTIONS SYSTEMES

Dans les blocs système, il s'agit de fonctions déjà prêtes, enregistrées dans la CPU. Ces blocs peuvent être appelés par l'utilisateur et être utilisés dans le programme.

STEP 7 offre les blocs système suivants :

- **SFB (Bloc de fonctions système) :**
Enregistré dans le système d'exploitation de la CPU et bloc de fonction appelable par l'utilisateur.
- **SFC (Fonction système):**
Enregistrée dans le système d'exploitation de la CPU et fonction appelable par l'utilisateur.
- **SDB (Données système):**
Espace mémoire dans le programme qui est créé par plusieurs outils STEP 7 (par ex. : Configuration matérielle, NetPro ...), afin d'enregistrer des données pour le système d'automatisation.

3. LANGAGE DE PROGRAMMATION STEP 7

3.1 GENERALITES SUR LE LANGAGE DE PROGRAMMATION STEP 7

A la suite du remplacement de **SIMATIC S5** par **SIMATIC S7** un nouveau logiciel de programmation (**STEP7**), basé sur la norme **IEC 61131**, a été développé.

STEP 7 marche sous **WINDOWS 95, 98** ou **NT** et possède une interface utilisateur graphique.



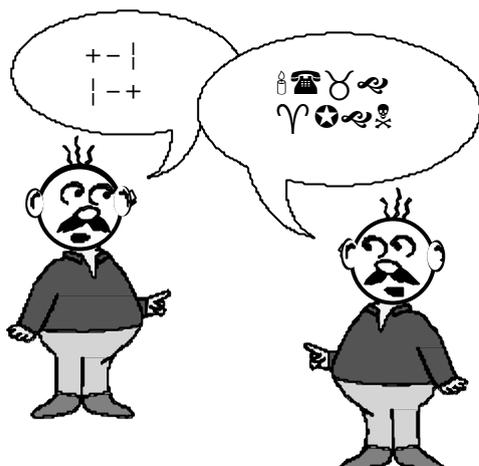
3.2 CONVERSION STEP 5 ➔ STEP 7

En STEP 7, vous avez la possibilité de convertir vos programmes STEP 5 vers STEP 7 par la fonction 'Convertir fichier S5'.

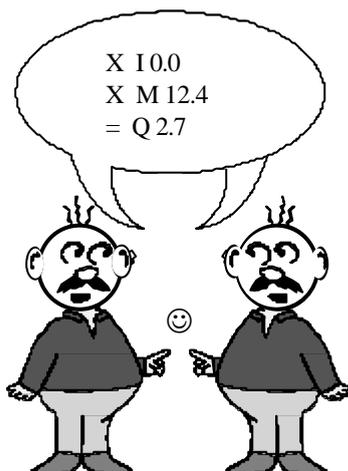


Vous trouverez de plus amples informations sur la conversion de STEP 5 dans les manuels de référence STEP 7 ou dans les instructions de conversion STEP 5 -> STEP 7 du 'Module B6 - Conversion STEP 5 -> STEP 7'.

3.3 LA NORME IEC 61131 POUR SPS



Jusque là, les nouveaux développements de la technique SPS ont conduit à une quantité de langages et dialectes spécifiques aux constructeurs. La langue commune de base s'est peu à peu perdue et, conséquence de cette tendance, il arrivait même des problèmes de communication entre différents SPS.



Par la norme IEC 61131, la technique des SPS a été pour la première fois établie sur une unité de base mondiale. Sous la présidence des Etats-Unis, la commission internationale de l'électrotechnique (IEC) a adopté la normalisation SPS suivant les cinq thèmes suivants :

- Partie 1 : Définitions générales et caractéristiques typiques des fonctions
- Partie 2 : Cahiers des charges électriques, mécaniques et fonctionnels des appareils
- Partie 3 : Cinq langages de programmation
- Partie 4 : Lignes directrices des phases de projet
- Partie 5 : Communication entre SPS de différents constructeurs

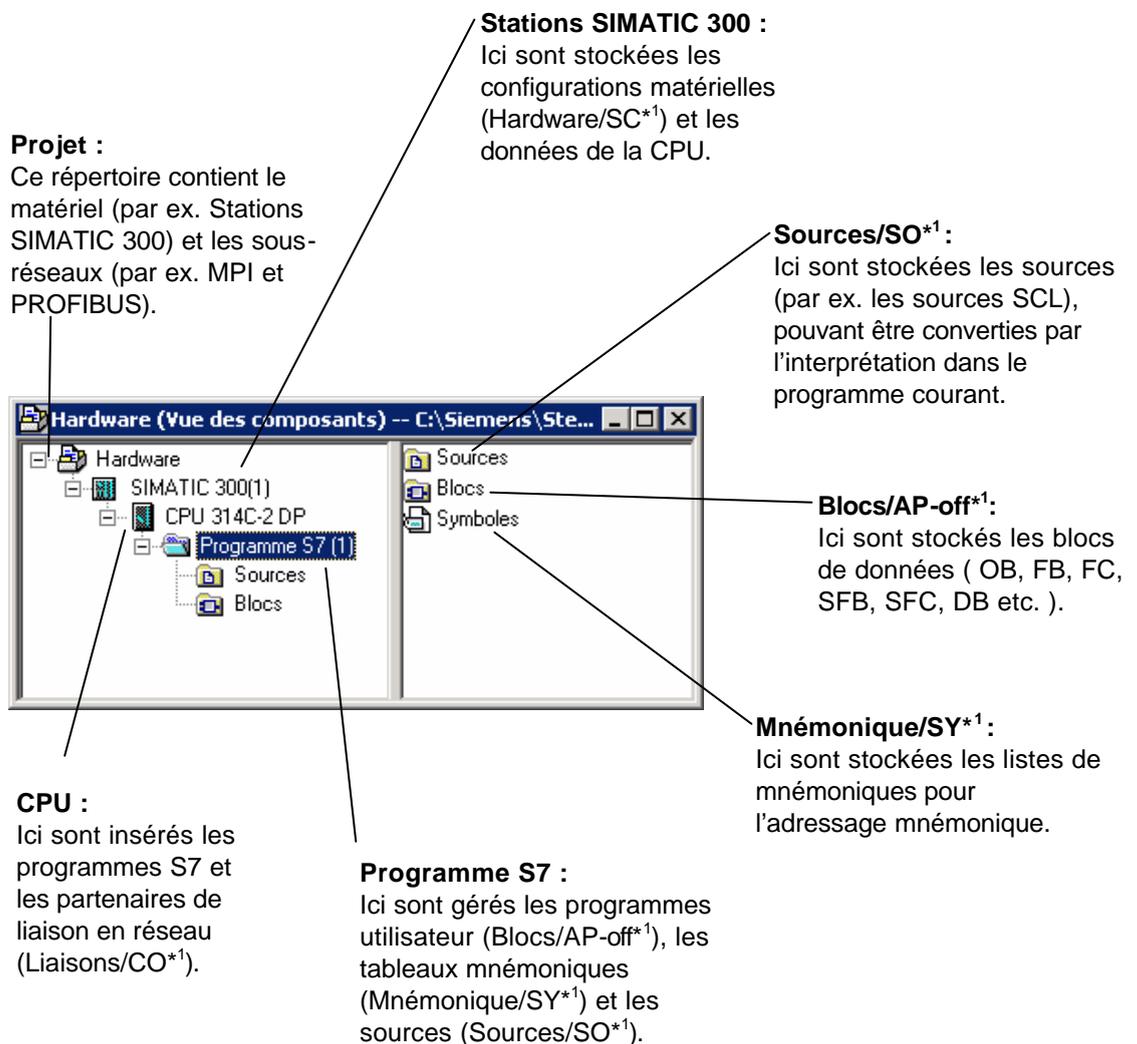
3.4 ARBORESCENCE

La gestion des fichiers en STEP 7 s'effectue avec '**SIMATIC Manager**'. Ici il est possible par ex. de copier les blocs de programme ou d'appeler d'autres traitements avec d'autres outils en cliquant avec la souris. L'utilisation correspond aux standards habituels WINDOWS. (donc on a entre autres la possibilité de faire apparaître un menu contextuel sur chaque composant par un bouton droit de la souris).

En STEP 7, chaque projet est stocké dans une structure fixe prédéfinie.

Dans les répertoires '**Station SIMATIC 300**' et '**CPU**' on met en place la conception matérielle du SPS. En conséquence, on voit toujours un tel projet d'un point de vue de son matériel.

Les programmes sont enregistrés dans les répertoires suivants :



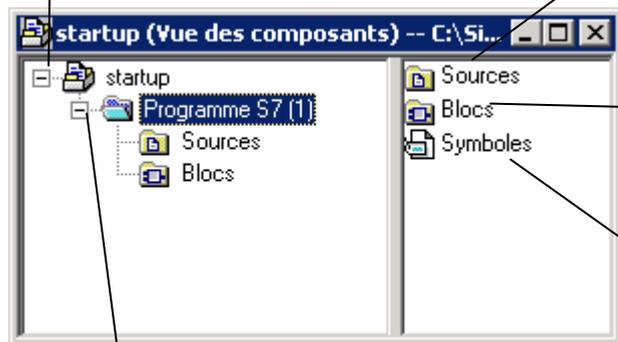
*1 Dénominations de STEP 7 Version 2.x

Pour créer un projet indépendant de la configuration matérielle, il y a toutefois la possibilité de l'aménager de telle sorte qu'il ne soit pas contenu dans ce répertoire.

Il a ainsi la structure suivante :

Projet:

Ce répertoire contient le matériel (par ex. Stations SIMATIC 300) et les sous-réseaux (par ex. MPI et PROFIBUS)..



Sources/SO*¹:

Ici sont stockées les sources (par ex. les sources SCL), pouvant être converties par l'interpréteur dans le programme courant.

Blocs/AP-off*¹:

Ici sont stockés les blocs de données (OB, FB, FC, SFB, SFC, DB etc.).

Mnémonique/SY*¹:

Ici sont stockées les listes de mnémoniques pour l'adressage mnémonique.

Programme S7 :

Ici sont gérés les programmes utilisateur (Blocs/AP-off*¹), les tableaux mnémoniques (Mnémonique/SY*¹) et les sources (Sources/SO*¹).

*¹ Dénominations de STEP 7 Version 2.x

3.5 CONFIGURER ET PARAMETRER

Pendant le montage, la CPU produit une configuration pratique et stocke celle-ci dans les données système (SDB).

Avec l'outil 'Configuration HW' il est possible de créer une configuration théorique dérivant de cette dernière et ainsi de configurer une nouvelle conception. De plus, on peut aussi charger une configuration existante depuis une CPU. En plus des modules comme la CPU, d'autres paramètres peuvent être prédéfinis (par ex. comportement de démarrage et de cycle d'une CPU).

The screenshot shows the 'HW Config' software interface for a SIMATIC 300(1) system. The main window displays a rack configuration with a PS 307 5A power supply at slot 1 and a CPU 314C-2 DP at slot 2. The CPU is connected to a PROFIBUS DP network. Below the rack view is a table with the following data:

Emplacement	Module	Référence	Firmware	Adresse MPI	Adresse d'entrée	Adresse de sortie	Commentaire
1	PS 307 5A	6ES7 307-1EA00-0AA0					
2	CPU 314C-2 DP	6ES7 314-6CF00-0A80	V1.0	2			
X2	DP				1023*		
2.2	DI24/DO16				124...126	124...125	
2.3	AI5/AO2				762...767	762...765	
2.4	Comptage				768...783	768...783	
2.5	Positionnement				784...799	784...799	
3							
4							

At the bottom of the window, there is a note: "Pour obtenir de l'aide, appuyez sur F1."

3.6 L'INSTRUCTION DE COMMANDE

Pour le traitement par une commande à mémoire programmable, la tâche de programmation est décomposée en instructions élémentaires de commande. Une instruction de commande est l'unité autonome d'un programme de commande. Elle représente une directive de travail pour le système de contrôle.

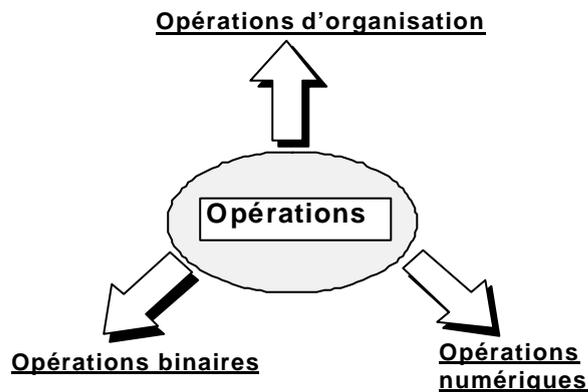
Les dénominations, les identificateurs et les symboles des instructions de commandes sont imposés par la norme DIN 19 239.

Une instruction de commande est structurée de la manière suivante :

Instruction de commande		
Partie opération	Partie opérande	
	Identificateur	Paramètres
U	E	0.0

3.6.1 PARTIE OPERATION

L'opération décrit la fonction à exécuter. DIN 19 239 distingue :

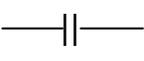
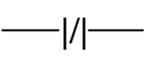
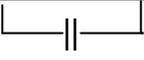
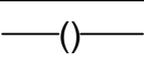


Exemples d'opérations numériques :

L	Charger
T	Transférer
>I	Plus grand que, pour des entiers
==R	Egalité, pour des réels
etc		

Exemples d'opérations binaires:

Extrait de DIN 19 239

LOG	CONT	LIST	
		U	ET
		N	NON
		O	OU
		=	ATTRIBUTION

etc

Exemples d'opérations d'organisation :

CC	Appel de bloc conditionnel
UC	Appel de bloc inconditionnel
AUF	Ouvrir bloc de données
SPA	Saut inconditionnel
SPB	Saut conditionnel
BEA	Terminaison de bloc inconditionnelle
BEB	Terminaison de bloc conditionnelle
etc		

3.6.2 PARTIE OPERANDE

La partie opérande contient toutes les indications nécessaires pour l'exécution d'opérations.
Elle détermine avec quoi le système de contrôle doit exécuter une opération.
L'identificateur des opérandes comprend la catégorie de l'opérande.

Il y a par exemple :

E pour Entrée
A pour Sortie
M pour Mémoire Interne
L pour données Locales (variables internes au bloc)
T pour le Temps
Z pour les Compteurs
OB pour bloc d'organisation
FB pour bloc de fonction
FC pour Fonction
DB pour bloc de données
SFB pour bloc de fonctions système
SFC pour fonction système
L#.pour Constante 32 bits etc.



Le paramètre de l'opérande donne l'adresse des opérandes.

3.7 ADRESSAGE

3.7.1 ADRESSAGE MNEMONIQUE

L'adressage mnémonique est souvent fort utile pour une meilleure compréhension. Il vous permet d'associer une adresse absolue définie à un nom mnémonique. Par exemple, on peut attribuer à l'entrée E 0.0 le nom END_STOP et au type de données BOOL. Chaque nom mnémonique ne doit être utilisé qu'une fois. L'attribution des associations s'effectue avec l'outil éditeur de mnémoniques que vous démarrez depuis SIMATIC Manager.

3.7.2 ADRESSAGE ABSOLU

En STEP 7 il y a les catégories suivantes d'adressage absolu :

- adressage immédiat
- adressage direct
- adressage mémoire indirect

Adressage immédiat :

Lors de l'adressage immédiat, l'opérande est directement codé dans l'opération, c'est-à-dire qu'il a directement la valeur avec laquelle l'opération doit être traitée, soit que l'opération fournisse implicitement l'opérande.

Exemple :	SET	Met le VKE (Résultat logique) à 1.
	+D	Ajoute le contenu de ACCU1 au contenu de ACCU 2 et stocke le résultat dans ACCU 1.

Adressage direct :

Lors de l'adressage direct, l'adresse de l'opérande est codée dans l'opération, c'est-à-dire que l'opérande donne l'adresse de la valeur que l'opération doit traiter.

L'opérande se compose d'un identificateur d'opérande et d'un paramètre. Il pointe directement sur l'adresse de la valeur.

Exemple :	U E 0.0	Exécute un ET logique du bit d'entrée E 0.0.
	L EB 0	Charge l'octet d'entrée EB 0 dans l' ACCU 1.
	= A 4.0	Attribue à VKE le bit de sortie A 4.0.

Adressage mémoire indirect :

Dans le cas de l'adressage mémoire indirect, l'adresse de l'opérande est prédéfinie par un autre opérande qui contient l'adresse du premier. Autrement dit, l'opérande donne l'adresse de la valeur sur laquelle doit être effectuée l'opération.

L'opérande se compose d'un identificateur d'opérande et d'un des pointeurs suivants :

- un mot qui contient le numéro d'un temps (T), d'un compteur (Z), d'un bloc de données (DB), d'une fonction (FC) ou d'un bloc de fonctions (FB).
- un double mot qui contient l'adresse exacte d'une valeur à l'intérieur de l'espace mémoire, la valeur donnant l'identificateur de l'opérande

L'adresse de la valeur ou du numéro est fournie indirectement via le pointeur. Le mot ou le double mot peuvent se trouver dans une mémoire interne (M), dans un bloc de données (DB), dans un bloc de données instance (DI) ou dans des données locales (L).

Exemple :	U E [MD 3]	Exécute une opération logique ET du bit d'entrée. L'adresse exacte se trouve dans la mémoire interne double mot MD 3.
	L EB [DID 4]	Charge l'octet d'entrée dans ACCU 1. L'adresse exacte se trouve dans l'instance du mot double de données DID 4.
	AUF DB [MW 2]	Ouvre le bloc de données. Le numéro du bloc de données se trouve dans le mot de mémoire interne MW 2.

3.8 REPRESENTATION DE PROGRAMME

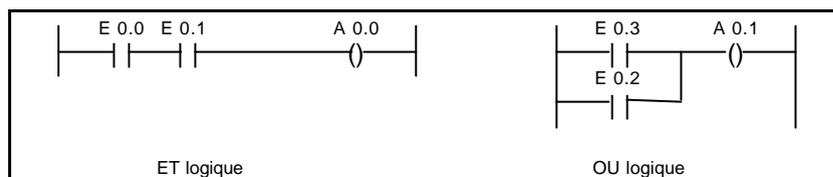
En STEP 7, un programme dans l'outil '**Programmer des blocs CONT/LIST/LOG**' peut être programmé et représenté dans trois catégories différentes :

- Schéma à contact CONT
- Plan logique LOG
- Liste d'instructions LIST

En option, il y a encore les packs logiciel programmation de langage haut niveau (SCL), de programmation graph-7 (S7-GRAPH) et la programmation d'état graphique (S7-HiGraph).

3.8.1 SCHEMA A CONTACT CONT

Le schéma à contact est la représentation imagée des tâches de commandes avec des mnémoniques suivant DIN 19 239, qui est aussi courante aux Etats-Unis. Il a beaucoup de ressemblances avec le traditionnel schéma de connexion, à la différence notable que la représentation à l'écran utilise des traits horizontaux, et non pas verticaux, pour les branches de circuit.



3.8.2 PLAN LOGIQUE LOG (EN STEP 7 A PARTIR DES VERSIONS 3.x)

Le plan logique est la représentation imagée des tâches de commande avec des mnémoniques suivant DIN 40 700 et DIN 19 239. Les différentes fonctions sont représentées par un symbole qui les identifie. A gauche du symbole, on trouve les entrées et à droite, les sorties.



3.8.3 LISTE D'INSTRUCTIONS LIST

En liste d'instructions, les tâches de commande sont décrites par des instructions de commande. Les instructions (Opération et opérande) de commande représentent les tâches au moyen d'abréviations (évidentes) des désignations de fonction. (suivant DIN 19 239)

Partie opération :	Partie opérande :		
	Symbole	Paramètre	
U	E	0.0	ET logique
U	E	0.1	
=	A	4.0	OU logique
O	E	0.2	
O	E	0.3	
=	A	4.1	

Chaque représentation possède ses propres propriétés et ses limites définies. Si lors de la programmation les règles précises sont suivies, alors ce n'est pas un problème d'effectuer une traduction du problème dans les trois modes de représentation. Si des programmes de commande ont été écrits en CONT ou en LOG, ils peuvent systématiquement être traduits en LIST. Dans la mémoire programme de l'appareil de commande, le programme est toujours stocké en LIST (en langage machine, en fait).

3.9 MEMOIRE INTERNE

On mettra en place des mémoires internes sur des liaisons de commandes, pour lesquelles il n'y a pas de transfert en dehors de la commande.

Les mémoires internes sont des éléments mémoire électroniques (Bascule R-S), dans lesquels les deux états de signaux "0" et "1" peuvent être stockés.

Chaque SPS dispose d'un grand nombre de mémoires internes. Elles sont programmées comme les sorties. En cas de coupure de la tension système, le contenu stocké dans la mémoire interne est perdu.

3.9.1 MEMOIRE INTERNE REMANENTE

Une partie de cette mémoire interne est pourtant rémanente (tension nulle sécurisée). A l'aide d'une batterie tampon dans le SPS, la coupure de tension est pontée. Ces états logiques sont ainsi conservés.

Mémoire interne rémanente

Conserve le dernier état d'avant la coupure de la tension d'alimentation

Conserve le dernier état avant le changement de régime "RUN>STOP"

Peut être réinitialisée par "AG>FORMATER" (en plus de la possibilité de le faire par le programme utilisateur)



En employant la mémoire rémanente, on peut sauvegarder le dernier état du dispositif ou des machines avant de quitter l'état de fonctionnement courant. En redémarrant, on a la possibilité de travailler avec le dispositif ou la machine dans l'état où il s'est arrêté.

L'espace de rémanence est défini dans les propriétés de la CPU, au niveau de l'outil **'Configuration matérielle'** sous **'Rémanence'**.

3.9.2 MEMOIRE INTERNE NON REMANENTE



Les mémoires internes sont réinitialisées par commutation sur les modes de fonctionnement „RUN>STOP„ et "RESEAU MARCHE".