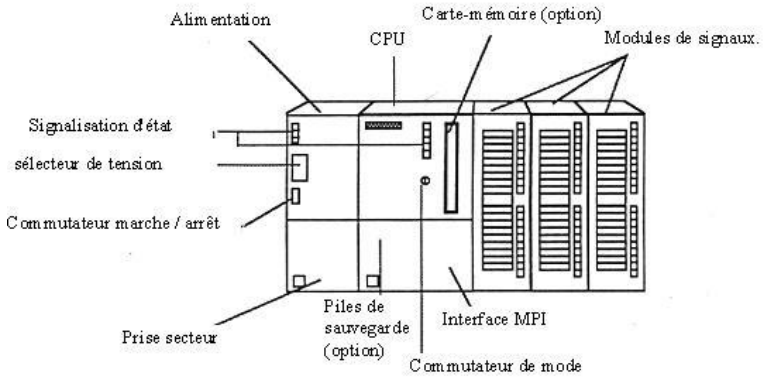
CH V : Programmation Avancée de l'API S7-300 de Siemens

I- Présentation des API S7-300 de Siemens

Le système d'automatisation SIMATIC S7-300 est un automate modulaire de milieu de gamme. Il existe une gamme étendue de modules S7-300 pour répondre de manière optimale à différente tâche d'automatisation (exemple SIMATIC S7-300 CPU 313C).



L'automate S7 est constitué d'une alimentation (Modules PS), d'une CPU ainsi que des modules d'entrées / sorties.

Siemens fournit des :

- Modules d'extension IM pour configuration multirangées du S7-300
- Modules de signaux SM pour entrées et sorties TOR et analogiques
- Modules de fonction FM pour fonctions spéciales (par exemple l'activation d'un moteur pas à pas)
- Processeurs de communication CP pour la connexion au réseau

II- Programmation de l'API S7-300 de Siemens

1- Présentation du langage de programmation

L'API S7-300 est programmable à l'aide d'un PC avec le logiciel STEP 7 (ou à l'aide La plate-forme Siemens TIA Portal (Totally Integrated Automation)) sous Windows qui offre les fonctions suivantes pour l'automatisation d'une installation :

- Configuration et paramétrage du matériel
- Paramétrage de la communication
- Programmation
- Test, mise en service et maintenance
- Documentation, archivage
- Fonctions de diagnostic et d'exploitation

La version de base **STEP 7** permet l'utilisation d'autres logiciels optionnels tels que S7-GRAPH ou S7-PLCSIM.

Le logiciel **STEP 7** permet la programmation **L'API S7-300 en :**

- Ladder Diagram LD (STEP 7 : CONT)·
- Function Block Diagram FBD (STEP 7 : LOG)·
- Sequential Function Chart SFC (STEP 7 : GRAPH7)·
- Instruction List IL (STEP 7 : LIST)·
- Structured Text ST (STEP 7 : SCL) qui est un langage évolué proche du C.

Une liaison MPI (**M**ulti **P**oint **I**nterface ou interface multipoint) est nécessaire pour programmer un SIMATIC S7-300 depuis le PC ou la PG. C'est une interface de communication utilisée pour la programmation, le contrôle-commande avec HMI et l'échange de données entre des CPU SIMATIC S7.

Le domaine d'utilisation de S7-PLCSIM est essentiellement le test de programmes STEP 7 pour la SIMATIC S7-300 et la SIMATIC S7-400, que l'on ne peut pas tester directement par le hardware (simulation du fonctionnement de l'API).

2- Variables et Adressage des E/S :

La mémoire de l'API S7-300 est compartimentée en zone chacune ayant une application particulière :

Zone E : Mémoire image des entrées
Zone A : Mémoire image des sorties
Zone M : Mémoire utilisateur
Zone L : Mémoire locale, associée à un module de programme
Zone P : Accès à la périphérie
Zone T : Mémoire des temporisations
Zone Z : Mémoire des compteurs
Zone DB : Mémoire utilisateur ou système structuré dans des blocs de données

Les objets E, A, M, DB, PE et PA sont rangés dans des octets (8 bits), on peut accéder à un BIT, à un OCTET, à un MOT de 16 bits ou à un DOUBLE MOT (32 bits) (voir *S7-1*).

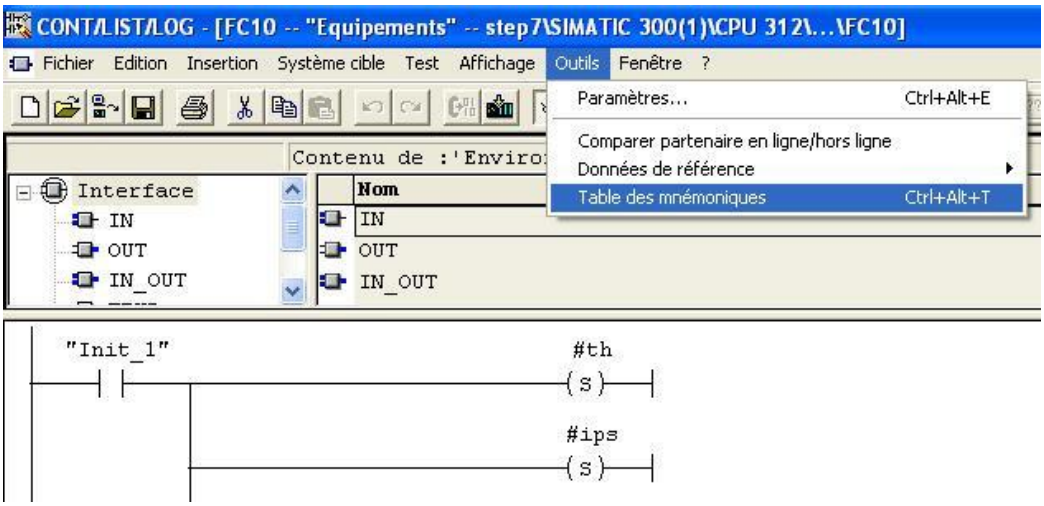
Remarque : Le logiciel step7 permet la programmation en notation allemande (E, A, T, Z) ou en notation anglaise (I, Q, T, C).

3- Adressage mnémonique

L'adressage mnémonique est souvent fort utile pour une meilleure compréhension. Il 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 dans la table des Mnémoniques on y définit Le nom du symbole, son adresse réel, son type et son commentaire.

On peut accéder à la table des symboles depuis l'éditeur CONT/LIST/LOG



4- Configuration :

Utilisé pour les API modulaires, la configuration permet de préciser le nombre, le type et l'emplacement des cartes et modules utilisés.

Pendant le montage de l'API S7-300, la CPU produit une configuration pratique et stocke celle-ci dans les données système (SDB).

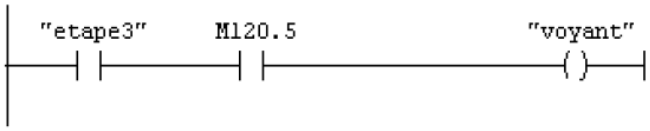
Avec l'outil **'Configuration HW'** il est possible de créer une configuration théorique dérivant de cette dernière et ainsi de configurer une nouvelle conception. De plus, on peut aussi charger une configuration existante depuis une CPU. En plus des modules comme la CPU, d'autres paramètres peuvent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

Le mémento de cadence (clignotement) :

Le mémento de cadence est un octet. Chacun des bits de cet octet change d'état suivant une horloge interne. Une durée de période et la fréquence correspondante sont affectées à chaque bit de l'octet de mémento de cadence :

Bit	7	6	5	4	3	2	1	0
Durée de période (s)	2	1,6	1	0,8	0,5	0,4	0,2	0,1
Fréquence (Hz) :	0,5	0,625	1	1,25	2	2,5	5	10

Exemple **:** Pour le mémento de cadence on choisit l'octet 120. Le bit 5 de MB120 change d'état toutes les secondes :



5- Instructions : Voir *S7-2 à S7-8*

III- Structure d'un programme STEP 7

Le STEP 7 utilise des Blocs d'organisation (OB) et des Fonctions et Blocs fonctionnels (FB et FC) permettant l'écriture d'un programme dans différents modules, chaque module traitera une fonction de l'automatisme qui seront appelles par le programme principale **(OB 1)**.

- **Les Blocs d'organisation OB** **OB 1 à OB 122**

Ces blocs déterminent la structure du programme utilisateur. Les OB sont directement appelés par le système d'exploitation de la CPU en réaction à un événement (à condition toutefois de les avoir programmé et insérés dans l'automate).

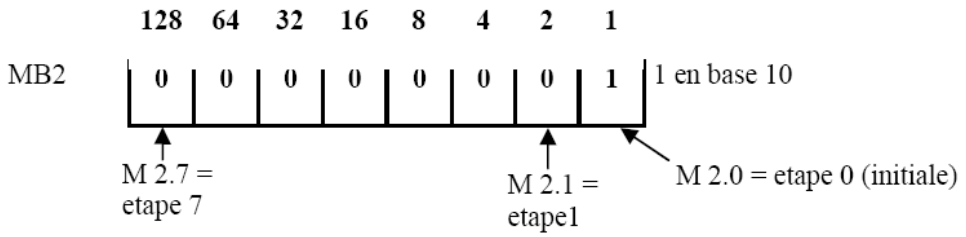
Programme cyclique **OB 1**

Lors d'une exécution normale de programme, les traitements se font de façon cyclique. L'exécution du programme contenu dans l'OB 1 est démarrée une fois par cycle (quand il est fini, il recommence). On peut se servir de l'OB 1 pour appeler des blocs de type FC ou FB.

Programme de démarrage **OB 100**

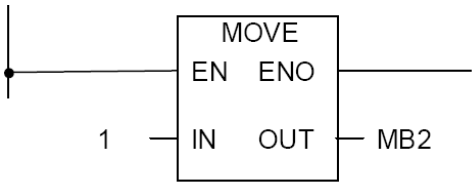
C'est l'OB de démarrage à chaud. Il ne sera exécuté qu'une seule fois à la mise en RUN de l'API. On l'utilise pour initialiser les grafcets par exemple.

Exemple : Soit un grafcet comportant 8 étapes de 0 à 7 avec l'étape 0 comme étape initiale, ce grafcet est mémorisé dans l'octet 2 (MB2).



Initialiser le grafcet consiste à donner la valeur décimale « 1 » à l'octet MB2 à la mise en RUN de l'automate.

- Programme de **OB100** :



OB1	(Programme cyclique)
OB10 à OB17	(OB d'alarme horaire)
OB20 à OB23	(OB d'alarme temporisée)
OB30 à OB38	(OB d'alarme cyclique)
OB40 à OB47	(OB d'alarme de processus)
OB70	(OB d'erreur de redondance dans la périphérie)
OB72	(OB d'erreur de redondance dans la CPU)
OB73	(OB d'erreur de redondance de communication)
OB80	(OB d'erreur de temps)
OB82	(OB d'alarme de diagnostic)
OB81	(OB d'erreur d'alimentation)
OB83	(OB de débrochage/enfichage)
OB84	(OB d'erreur matérielle sur CPU)
OB85	(OB d'erreur d'exécution du programme)
OB86	(OB de défaillance d'unité)
OB87	(OB d'erreur de communication)
OB90	(OB d'arrière-plan)
OB100	(Démarrage à chaud)
OB101	(Redémarrage)
OB102	(Démarrage à froid)
OB121	(OB d'erreur de programmation)
OB122	(OB d'erreur d'accès à la périphérie)

Figure 5-1 : Liste des Blocs d'organisation OB disponible sur l'API S7-300

➢ **Les Fonctions et Blocs fonctionnels FC et FB**

Ce sont des fonctions écrites en LIST, CONTACT ou LOGIGRAMME, qui peuvent recevoir des paramètres d'entrées de sorties. On peut y déclarer des variables locales temporaires.

Le bloc fonctionnel (FB et FC) est subordonné au bloc d'organisation. Il renferme une partie du programme qui peut être appelée autant de fois qu'on le veut dans l'OB1.

Les FB et FC s'adaptent particulièrement bien à la programmation de fonctions récurrentes. Contrairement aux FC, les FB sont des blocs avec mémoire, les paramètres transmis aux FB sont sauvegardés dans un bloc de donnée d'instance DB