

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA
RECHERCHE SCIENTIFIQUE



UNIVERSITE DE M'SILA
FACULTE DES SCIENCES ET SCIENCES DE L'INGENIORAT
DEPARTEMENT D'ELECTROTECHNIQUE

MEMOIRE DE FIN D'ETUDES EN VUE DE L'OBTENTION DU DIPLÔME
D'INGENIORAT D'ETAT EN GENIE ELECTROTECHNIQUE

OPTION: COMMANDE ELECTRIQUE

THEME

**COMMANDE D'UN BRAS MANIPULATEUR
PAR AUTOMATE PROGRAMMABLE
INDUSTRIEL (A.P.I)**

Proposé et dirigé par :

Mr. S.BELKHIRI

Présenté par :

**Telli Abderrahim
&
Khodja Choayb**

Année Universitaire : 2008 / 2009

*R*emerciement

*Nous remercions tout if a6ord dieu
qui nous a donné la force et la volonté if'élaborer ce projet*

Nous remercions vivement notre encadreur

*Mr : **Belkhiri salah** , qui a suivi notre travail*

avec leur chers conseilles de nous avoir de plein cœur.

Nous remercions tous nos amis étudiants pour leur aide,

et particulièrement tous ceux qui ont participé

de près ou de foin dans ce travail.

Dédicace

je dédie ce modeste travail

*A tous les membres de ma famille grand et petit surtout mes parents,
mes grands-parents.*

*A tous les enseignants qui m'ont aidé de proche ou de loin pour être un
jour un ingénieur d'état en électrotechnique.*

*A tous mes collègues qui m'a accompagné pendant le long de cette
période pour réaliser ce modeste travail.*

A tous mes amis surtout mon collègue Lachrem Imad

*A tous les étudiants de la faculté en génie électrotechnique surtout
les étudiants de la 5^{ème} année promotion 2009.*

*A tous ceux qui m'ont aidé, de près ou de loin, même qu'il soit un
mot d'encouragement et de gentillesse.*

A tous ceux que j'aime et qui m'aiment.

Telli Abderrahim.

Dédicace

*je dédie ce modeste travail à deux grands cœurs
qui m'ont donné l'amour pendant toute ma vie et qui sont
un pur appuie dans toute ma démarche qui sont
mes parents*

*Et aussi à toute personne qui me connaît de loin et de proche et
vivre avec moi instant de bonheur et de malheur..*

Choayb

Liste des figures

Chapitre I: Introduction aux Systèmes Automatisés

<i>Fig.(I.1): Structure d'un système automatisé.....</i>	6
<i>Fig. (I.2): signaux des données.....</i>	7
<i>Fig.(I.3): Mode de sortie d'un capteur TOR</i>	14
<i>Fig.(I.4): Mode de sortie d'un capteur analogique.</i>	15
<i>Fig.(I.5): Mode de sortie d'un capteur numérique.</i>	16

Chapitre II: les Automates Programmables Industriels

<i>Fig.(II.1): L'automate programmable siemens S7 300</i>	26
<i>Fig.(II.2): Architecture interne d'un automate programmable.</i>	27
<i>Fig.(II.3):L'environnement de l'API</i>	34
<i>Fig.(II.4): Le GRAFCET</i>	35
<i>Fig.(II.5): exemple d'un programme LADDER</i>	36
<i>Fig.(II.6): exemple d'un programme LADDER</i>	37
<i>Fig.(II.7): exemple d'un programme en Fonction Bloc</i>	38
<i>Fig.(II.8): exemple d'un programmation en langage Structuré</i>	38
<i>Fig.(II.9): exemple d'un programme en IL</i>	39

Chapitre III: Automates S7 300

<i>Fig.(III.1): L'API S200</i>	44
<i>Fig.(III.2): L'API S300</i>	44
<i>Fig.(III.3): L'API S400</i>	45
<i>Fig.(III.4) : modules d'un API</i>	46
<i>Fig.(III.5): Unité centrale (CPU) de l'API.....</i>	47
<i>Fig.(III.6): entrée /sortie d'un API.....</i>	48
<i>Fig.(III.7): Vue générale de la CPU S7-314 IFM</i>	50
<i>Fig.(III.8): LED de visualisation d'état de la CPU.....</i>	50
<i>Fig.(III.9): Profil de vitesse d'un moteur à deux vitesse.....</i>	57
<i>Fig.(III.10): Chronogramme des sorties TOR pour la fonction intégrée positionnement. (Positionnement en marche avant)</i>	58
<i>Fig.(III.11): Profils de vitesses et d'accélération pour la fonction positionnement.....</i>	59

Chapitre IV: Logiciel STEP7

<i>Fig.(IV.1): Applications du logiciel de base STEP7.....</i>	63
<i>Fig.(IV.2) :Fenêtre du SIMATIC manager.</i>	64
<i>Fig.(IV.3): Applications techniques.</i>	66

<i>Fig.(IV.4): Logiciels exécutables .</i>	67
<i>Fig. (IV.5): Interfaces homme/machine</i>	67
<i>Fig.(IV.6): Organigramme pour la création de projets sous STEP 7.</i>	68
<i>Fig.(IV.7): Création d'un projet sans l'assistant.</i>	70
<i>Fig.(IV.8): Configuration du matériel.</i>	71
<i>Fig.(IV.9):La fenêtre d'édition de mnémoniques.</i>	72
<i>Fig.(IV.10):L'hierarchie d'un projet STEP 7.</i>	73
<i>Fig.(IV.11): Création de blocs de données</i>	77
<i>Fig.(IV.12): « OU » logique avec CONT.</i>	83
<i>Fig.(IV.13): ET logique avec CONT.</i>	84
<i>Fig. (IV.14): Adressage immédiat et direct avec LOG.</i>	85
<i>Fig.(IV.15): mode de représentation des langages basiques de programmation STEP 7..</i>	87
<i>Fig.(IV.16): fenêtre de simulateur PLC-SIM</i>	88
<i>Fig.(IV.17):fenêtre de S7-GRAPH</i>	90

Chapitre V: Etude et Simulation d'un Bras Manipulateur

<i>Fig.(V.1): le bras manipulateur avec ses axes .</i>	93
<i>Fig.(V.2): Composante d'un bras manipulateur .</i>	94
<i>Fig.(V.3) : billon énergétique d'un moteur à courant continu .</i>	95
<i>Fig.(V.4): détecteur de position .</i>	97
<i>Fig.(V.5): assistant nouveau projet .</i>	99
<i>Fig.(V.6) : vue des composants d'un projet .</i>	99
<i>Fig.(V.7) :configuration matériel(Modules).</i>	100
<i>Fig.(V.8): table mnémonique .</i>	101
<i>Fig.(V.9): création du programme .</i>	102
<i>Fig.(V.10): activer- désactiver le simulateur à partir de l'interface .</i>	102
<i>Fig.(V.11): l'environnement de S7-PLCSIM .</i>	103
<i>Fig.(V.12): réglage des paramètres</i>	103
<i>Fig.(V.13) : chargement du programme à simuler</i>	104
<i>Fig.(V.14) : visualisation des programme</i>	104

Liste des Tableaux

Tableau(III.1): Le commutateur de mode de fonctionnement de l'API.....	51
Tableau(III.2): Tableaux techniques	54
Tableau(III.3): Critères de sélections pour la résolution d'un problème d'automatisation à base d'automates programmables.	55
Tableau(III.4): Sorties associées à la commande d'un moteur à deux vitesses.	56
Tableau(III.5): Sorties associées à la commande d'un moteur par variateur de vitesse...	58
Tableau (IV.1): Classes de priorité des blocs d'organisation	80
Tableau(IV.2): Quelques types d'opérandes.	86

Introduction générale :

L'évolution rapide des technologies nouvelles a permis de contourner la plupart des difficultés rencontrées dans le monde industriel, et a fourni plusieurs possibilités pour satisfaire les exigences et les critères demandés tels que la productivité, la sécurité, l'optimisation des coûts de production et l'amélioration des conditions de travail.

L'automatisation des procédés industriels est actuellement l'un des axes où on fait appel, de plus en plus, aux technologies évoluées à mesure que les exigences du monde industriel ont aussi évolué. Parmi celles-ci, figurent les Automates Programmables Industriels (API), qui offrent la solution adaptée aux besoins exigés.

Un automate programmable est un système électronique, destiné à être utilisé dans un environnement industriel, il utilise une mémoire programmable pour le stockage interne des instructions orientées utilisateur aux fins de mise en oeuvre de fonctions spécifiques, telles que des fonctions de logique, de mise en séquence, de temporisation, de comptage et de calcul arithmétique, pour commander au moyen d'entrées et de sorties divers types de machines ou de processus.

Les API (ou Programmable Logic Controller PLC) sont aujourd'hui les constituants les plus répandus des automatismes. On les trouve non seulement dans tous les secteurs de l'industrie, mais aussi dans les services (gestion de parkings, d'accès à des bâtiments, contrôle du chauffage, de l'éclairage, de la sécurité ou des alarmes) et dans l'agriculture (composition et délivrance de rations alimentaires dans les élevages) et cela est due surtout à leurs performances de sécurité.

Dans tous les secteurs où ils sont utilisés les API doivent remplir des tâches de commande en élaborant des actions suivant une algorithmique appropriée, à partir des informations données par des détecteurs (Tout ou Rien) ou des capteurs (analogiques ou numériques), et des tâches de communication avec des opérateurs humains ou avec d'autres processus (automates, calculateurs de gestion de production,...).

Bien que les automates programmables incluent des modules et des options permettant le contrôle ou la commande d'un système continu, beaucoup d'utilisateurs pensent que les APIs ne sont destinées qu'à la commande séquentielle.

Le but de ce travail est l'élaboration d'une solution à base d'API pour commander un bras manipulateur .

Le présent travail s'articule autour de cinq parties. La première partie présente une description d'un système automatisée .

Dans la deuxième et la troisième partie, on présentera une étude générale sur les automates programmables industriels ainsi que l'API choisi pour effectuer l'automatisation. On présentera aussi, l'interface de programmation STEP7 ainsi que les avantages apportés.

Quant à la quatrième, on présentera l'interface de programmation STEP7 .

Enfin, on terminera l'étude par une cinquième partie dans laquelle ont vira ses différentes composantes et sont programme.

CHAPITRE I

Introduction aux systèmes automatisés

I.1.Introduction:

L'automatisation de la production consiste à transférer tout ou partie des tâches de coordination, auparavant exécutées par des opérateurs humains, dans un ensemble d'objets techniques appelé partie commande.

La Partie Commande mémorise le savoir faire des opérateurs pour obtenir la suite des actions à effectuer sur les matières d'oeuvre afin d'élaborer la valeur ajoutée.

Elle exploite un ensemble d'informations prélevées sur la Partie Opérative pour élaborer la succession des ordres nécessaires pour obtenir les actions souhaitées .

I.2. Objectifs de l'automatisation :

L'automatisation permet d'apporter des éléments supplémentaires à la valeur ajoutée par le système. Ces éléments sont exprimables en termes d'objectifs par :

- accroître la productivité du système c'est-à-dire augmenter la quantité de produits élaborés pendant une durée donnée. Cet accroissement de productivité exprime un gain de valeur ajoutée sous forme :
- d'une meilleure rentabilité.
- d'une meilleure compétitivité.
- améliorer la flexibilité de production .
- améliorer la qualité du produit grâce à une meilleure répétabilité de la valeur ajoutée .
- s'adapter à des contextes particuliers :

adaptation à des environnements hostiles pour l'homme (milieu salin, spatial,nucléaire...), adaptation à des tâches physiques ou intellectuelles pénibles pour l'homme (manipulation de lourdes charges, tâches répétitives parallélisées...) . augmenter la sécurité, etc...

I.3. pourquoi les automatismes ?

une machine ou un complexe industriel qui déroule seul son processus de fonctionnement dès que certaines actions initiales l'ont fait démarrer (pression sur bouton-poussoir ou grandeur physique agissant sur un capteur ,etc.) est ce que l'on appelle communément un processus automatique ou un automatisme tout court. De telles installation ont été mises au point en vue d'adopter des solution à des problèmes à caractère technique , économique , de sécurité et/ ou humains . [8]

un processus automatique doit permettre au moins :

- l'élimination des tâches complexes, fatigantes, rebutantes et dangereuses.
- l'accroissement de la production ainsi que de la productivité en général (industrielle, agricole, alimentaire, etc.)
- la souplesse de changement de fabrication nécessaire, lors de basses répétitivité, pour passer d'un type de fabrication à une autre le plus aisément possible.

simultanément

- l'amélioration de la qualité des produits fabriqués ou conditionnés.
- une plus grande sécurité du personnel.
- la surveillance et la protection des complexes et des machines industrielles.
- la réalisation des économies d'énergie industrielles et de matière.
- enfin, la réduction d'un prix de revient de l'article fabriqué ou conditionné.

I.4. Conduite et surveillance d'un système automatisé :

Il s'avère très difficile en pratique d'intégrer dans une Partie Commande la totalité des savoir-faire humains de sorte que l'automatisation reste souvent partielle :

certaines tâches restent confiées à des intervenants humains.

A ces causes techniques viennent s'ajouter des considérations économiques de compétitivité, des considérations financières imposant un fractionnement des investissements, des considérations sociales d'automatisation douce.

Certaines tâches restent donc manuelles et l'automatisation devra donc prendre en compte la spécificité du travail humain, c'est-à-dire en particulier :

- assurer le dialogue entre les intervenants et le système automatisé,
- assurer la sécurité de ces intervenants dans l'exécution de leurs tâches manuelles. En outre le modèle de fonctionnement de la Partie Commande, choisi par le concepteur du système, ne correspond qu'à un ensemble de situations prévues, c'est-à-dire retenues par le concepteur parmi un ensemble de situations possibles.

Or il est impératif de pouvoir faire face à des situations non prévues (donc non retenues en général pour des raisons économiques compte tenu de leur faible probabilité) . [8]

Seul un opérateur peut alors intervenir et prendre les décisions requises par cette situation : il assure une fonction de conduite et de surveillance du système automatisé. Cette fonction peut être plus ou moins assistée par un ensemble de moyens (pupitres, informatique...).

Le concepteur devra alors :

- fournir à l'intervenant (ou lui permettre de prélever) toutes les informations significatives (ou indices) nécessaires à l'analyse de la situation .
- lui permettre d'agir sur le système, soit directement (dépannage...), soit indirectement (consignes de sécurité, de marches et d'arrêts...).

I.5. Structures d'un système automatisé :

On distingue, dans tout système automatisé, la machine ou l'installation (partie de puissance) et la partie commande constituée par l'appareillage d'automatisme. Cette dernière partie est, assurée par des constituants répondant schématiquement à cinq fonctions de base : [2]

1. L'acquisition des données.
2. Traitement de données.
3. Commande de puissance (près actionneur).
4. Le dialogue homme machine.
5. La protection du système.

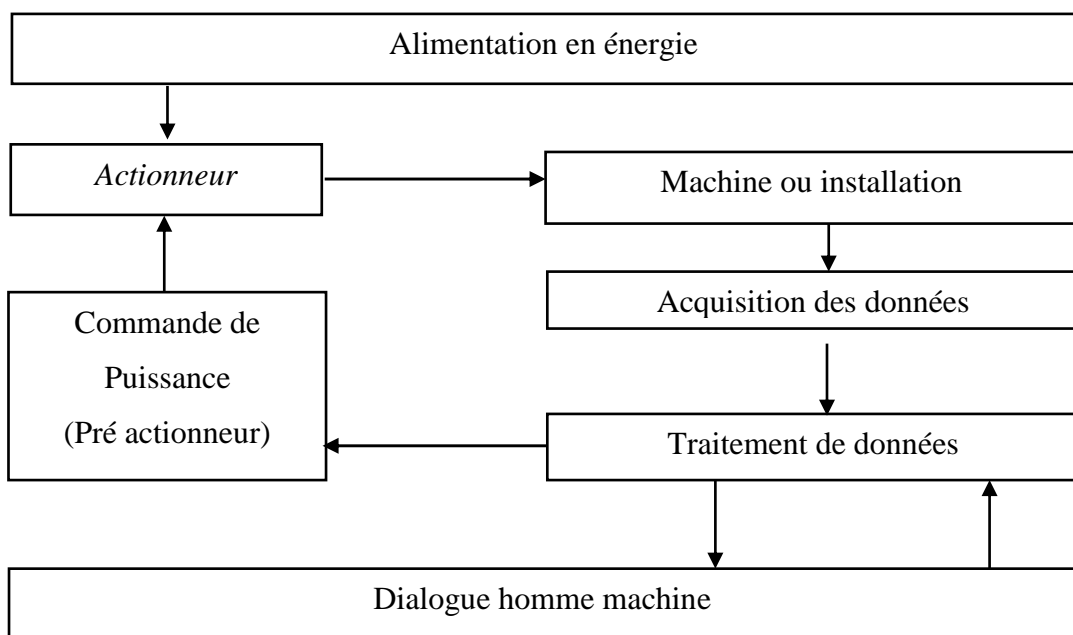


Fig.(I.1) Structure d'un système automatisé

I.5.1. Acquisition dès données :

L'automatisation d'une machine ou d'une installation nécessite la prise en le compte permanente des informations de commande, de position, de température, de vitesse.....

Ces informations sont transmises à la partie commande de la machine sous forme de messages codés et peuvent être de nature:

- **Binaire:** l'information est transmise en tout ou rien (TOR), il est utilisé dans les automatismes simples.
- **Numérique:** l'information est transmise sous la forme d'un code binaire par un mot de plusieurs bits.
- **Analogique:** l'information est transmise sous forme de tension (ou courant) proportionnelle à la grandeur mesurée, et évolue entre deux valeurs limites (ex: 0-20 mA).

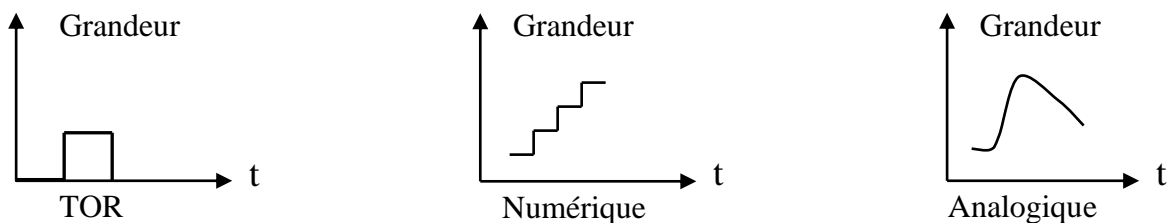


Fig. (I.2) : signaux des données

Ces informations sont transmises à la partie commande de la machine à l'aide des capteurs ou des détecteurs qui informent l'unité de traitement de l'état du système (variables d'entrées). Le choix de ces appareils est fonction de la condition d'utilisation : [7,2]

- Interrupteurs de position actionnés directement par la mécanique .
- Interrupteurs à flotteur pour contrôler un niveau .
- Sélecteurs de position pour suivre de déplacement d'un mobile .
- pour déceler la présence ou réguler une pression .
- Détecteurs de proximité inductifs ou capacitifs statiques lorsque la détection doit s'effectuer sans toucher au mobile à contrôler, ou lorsque les cadences sont trop élevées, ou dans des ambiances particulières .
- Cellules photo électriques pour détecter à des distances plus importantes .

- Détecteurs de vitesse pour contrôler des vitesses de déplacement ou de rotation .

I.5.2. Traitement des données :

L'ensemble des informations saisies par les capteurs est transmis à l'unité de traitement qui élabore les ordres d'actions, selon une procédure bien définie. En fonction de la nature de l'automatisme, le cycle de fonctionnement peut être soit *combinatoire*, soit *séquentiel*.

- **Cycle combinatoire:** le cycle de fonctionnement est réalisé uniquement par la combinaison de valeur primaire. La commande des sorties est directement liée aux informations présentes à un instant donné. Les actions antérieures ne sont pas mémorisées.
- **Cycle séquentiel:** le cycle de fonctionnement est défini en tenant compte des variables et secondaires. La commande des sorties est liée non seulement aux informations présentes, mais également aux actions passées. Ce cycle comporte obligatoirement des mémoires.

Suivant l'importance et la complexité de l'automatisme, le traitement des données est effectué par l'intermédiaire de relais d'automatisme, de contacteurs auxiliaires, de cellules logiques et de séquenceurs, de micro-systèmes ou à l'aide d'un automate programmable . [2]

I.5.3. Commande de puissance (pré actionneur) :

Ils servent à mettre en service ou hors service un actionneur, une machine ou une installation. Ils permettent d'assurer, en toute sécurité, la bonne marche d'un équipement.

Les signaux disponibles à la sortie de l'unité de traitement (variable de sortie) sont appliqués aux circuit de commande d'organes amplificateurs (bobine de relais, de contacteurs, distributeurs. etc.....), dont le circuit puissance alimente les actionneurs (moteurs, vérins, etc.....). Ces amplificateurs sont choisis en fonction de technologie retenue, de la puissance des actionneurs et de leurs conditions de fonctionnement.

Les principaux amplificateurs (ou actionneurs) sont :

- **Appareils de sectionnements :** ils permettent l'isolation ou la mise sous énergie d'un système, ils sont toujours à commande manuelle et intervient sur un circuit à vide (c'est-à-dire aucun actionneur alimente par ce circuit ne fonctionne), exp: sectionneur.

- **Appareil de commutation** : ils permettent de mettre un actionneur sous ou hors énergie, ils peuvent être à commande manuelle ou à commande automatique et interviennent sur un circuit à vide ou en charge (c'est-à-dire lorsqu'un ou plusieurs actionneurs, alimentés par ce circuit, fonctionnent), exp: contacteur, démarreurs, variateurs de vitesse, etc..... [2]

I.5.4. Dialogue homme-machine :

Le dialogue homme-machine est le complément indispensable de tout l'automatisme, il permet à l'opérateur d'intervenir au moment de démarrage ou en cours de cycle, de procéder à un arrêt d'urgence et par l'intermédiaire du système de signalisation, de contrôler en permanence le déroulement des opérations.

Cette fonction de dialogue est assurée par tous les auxiliaires de commande à intervention humaine (boutons, boîtes à boutons, commutateurs) ainsi que par des voyants de signalisation et pour les installations plus complexes, par des pupitres de Commande, des tableaux synoptiques, des boutons touches et claviers, des micro terminaux, etc.... [2]

I.5.5. Protection du système :

Les appareils de protection assurent le bon fonctionnement d'une installation ou d'une machine et la sécurité des personnes contre tout dysfonctionnement d'origine:

- **Electrique**: surintensité (surcharge et court-circuit), surtension.
- **Pneumatique**: surpression .
- **Hydraulique**: surpression .

Les différents types de protection et les produits correspondants utilisés dans les équipements automatiques à contacteur sont les suivants:

- Protection contre les surcharges importantes: relais électromagnétiques.
- Protection contre les courts-circuits: fusibles.
- **Protection contre la marche en monophasé**: relais thermiques différentiels, sectionneurs équipés d'un dispositif adéquat et munis de fusibles à percuteur.
- **Protection à manque de tension**: contacteur avec auto alimentation, relais de mesure.
- Protection à maximum d'intensité: relais de mesure.

- Protection à minimum d'intensité: relais de mesure.
- **Protection contre des démarrages trop longs ou trop fréquents:** contrôle de la durée d'une opération, relais temporisateurs thermiques. [2]

I.5.6. Les actionneurs :

Ce sont des organes destinés à remplacer l'énergie humaine par une énergie électrique, pneumatique ou hydraulique, ils permettent d'obtenir l'énergie nécessaire au bon fonctionnement de la machine a partir de l'énergie disponible dans l'équipement. [5]

Les principales caractéristiques des actionneurs sont :

- Moteur (électrique, pneumatique ou hydraulique) :
- couple.
- fréquence de rotation.
- Vérin (pneumatique ou hydraulique) .
- longueur de déplacement (course).
- vitesse de déplacement de la tige.
- effort possible dans les deux sens ou un seul sens.
- Résistance .
- puissance.
- température de fonctionnement.

I.5.7. Alimentation en énergie :

Différentes sources d'alimentation en énergie peuvent être utilisées dans les machines automatisées ou non automatisées, en fonction :

- de la puissance nécessaire ;
- du coût de revient ;
- du lieu d'exploitation ;
- de l'importance de l'automatisation.

Ces différentes énergies, qui peuvent être produites par des générateurs sur place ou extérieurement à l'installation, sont principalement :

- **Pneumatique:** produite par un compresseur entraîné par un moteur, c'est pour alimenter les machines fonctionnant avec un automatisme simple.
- **Hydraulique:** produite par un pompe entraînée par un moteur, c'est lorsque les puissance mise en jeu par les actionneurs sont importantes.
- **Electrique:**

On peut trouver, dans une même machine automatisée, une source d'énergie (électricité), deux sources d'énergie associées (électricité et hydraulique, ou électricité et pneumatique), ou les trois sources d'énergie. [4,7]

I.6. Connaissance du matériel :

Pour comprendre un système, ses fonctions et ses dysfonctionnements, il est important de bien connaître chacun de ses composants, son mode de fonctionnement, ses caractéristiques, son domaine d'emploi, ses dangers,.....

Dans ce qui suit on donne les informations essentielles de la plupart des composants constituant un système automatisé de production. On renseigne sur ses principales caractéristiques et fonctions pour faciliter les opérations de maintenance.

On distingue, dans tout système automatisé, la machine ou l'installation, et la partie commande constituée par l'appareillage d'automatisme. Cette partie commande est assurée par des constituants répondant schématiquement à cinq fonctions de base: [7,2]

- L'acquisition des données.
- Le traitement des données.
- La commande de puissance.
- Le dialogue homme machine.
- Et la protection des récepteurs et des équipements.

I.6.1. La commande de puissance :

L'énergie électrique, mise à la disposition des industriels ou des particuliers par l'intermédiaire d'un réseau de distribution, ne peut être connectée en permanence sur l'ensemble des récepteurs. Il est donc nécessaire d'employer des systèmes de commande de puissance permettant le transfert ou l'interruption de l'énergie électrique en provenance du réseau, vers le ou les récepteurs .

Ce sont les interrupteurs, disjoncteurs, variateurs électroniques et surtout les contacteurs qui assurent cette fonction appelée « commande de puissance ».

I.6.1.1. Le contacteur :

Le contacteur est un appareil mécanique de jonction commandé par un électro-aimant, il fonctionne par (tout ou rien). Lorsque la bobine de l'électro-aimant (KM1) est alimentée , le contacteur se ferme , établissant, par l'intermédiaire des pôles (1-2, 3-4, 5-6, 13-14, ou 11-12), le circuit entre le réseau d'alimentation et récepteur . [2]

La partie mobile des pôles se déplace :

- soit par rotation, en pivotant sur un axe.
- soit par translation, en glissant parallèlement aux parties fixes.
- soit par un mouvement conjugué des deux.

Dès que la bobine est privée tension, le circuit magnétique se démagnétise et le contacteur s'ouvre sous l'effet :

- des ressorts de pression des pôles.
- du ressort de rappel de l'armature mobile.
- pour certains appareils, de la gravité (l'équipage mobile tendant naturellement à reprendre sa position d'origine).

I.6.2. L'acquisition des données :

L'acquisition des données consiste à prendre en compte les phénomènes extérieurs (Physiques, électriques, électroniques) auxquels est asservi un équipement d'automatisme. Cette fonction est assurée par des capteurs .

I.6 .2.1.Les Capteurs:

Un capteur est un organe de prélèvement d'information qui élabore à partir d'une grandeur physique à une autre grandeur physique de nature différente. Cette grandeur prélevée est utilisable à des fins de mesure ou de commande.

La Nature des informations physiques appréhendées par les systèmes techniques Parmi les informations de toutes natures issues de notre environnement, on distingue les grandeurs physiques associées à des événements climatiques, géométriques ou encore lumineux ou temporels.

Le rôle du capteur est de rendre exploitable ces différentes grandeurs physiques en vue de leur traitement ultérieur.

- Mesure de présence : Indique la présence d'un « objet » à proximité immédiate .
- Mesure de position, de déplacement ou de niveau : Indique la position courante d'un objet animé d'un mouvement de rotation ou de translation.
- Mesure de vitesse : Indique la vitesse linéaire ou angulaire d'un « objet ».
- Mesure d'accélération, de vibrations ou de chocs.
- Mesure de débit, de force, de couples, de pressions.
- Mesure de température, d'humidité.

La diversité des informations physiques à capter et leurs importances au sein d'un système, engendrent un choix cohérent et approfondi des capteurs. La Relation entre l'information à détecter (ou capter) et de son image physique. Le capteur évite à l'homme de s'astreindre à guetter les événements ou phénomènes attendus sur le système.

Les capteurs utilisés dans les systèmes automatisés sont de véritables « organes des sens ». Ils engendrent, après traitements électroniques ou programmés, les actions requises sur la partie opérative du système étudié. La grandeur électrique de sortie du capteur peut varier : [7]

- de manière binaire (information vraie ou fausse), c'est le capteur Tout ou Rien (TOR).
- de façon progressive (variation continue), c'est le capteur analogique.
- d'échelon de tension ou de courant, c'est le capteur numérique.

a)Le capteur TOR :

Ces capteurs génèrent une information électrique de type binaire (« 0 » ou « 1 », vraie ou fausse) qui caractérise le phénomène à détecter ou capter. On distingue les capteurs TOR avec ou sans contact physique vis-à-vis de l'objet à détecter.

Les Capteurs TOR sont les capteurs les plus répandus en automatisation (interrupteurs de position, détecteurs de proximité ...).[7,2]

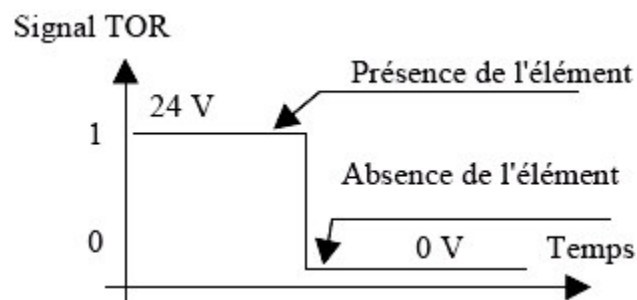


Fig.(I.3) Mode de sortie d'un capteur TOR

b)Le Capteur analogique :

La grandeur électrique délivrée par ce type de capteur est en relation directe avec la grandeur physique à capter. La caractéristique de transfert de ce type de capteur est linéaire dans les cas les plus fréquents (proportionnalité entre la grandeur physique mesurée et le signal délivré). Dans le cas d'une caractéristique non linéaire, le traitement électronique de l'information de sortie du capteur sera plus complexe (linéarisation programmée ou intégrée au capteur dans le cas de capteurs plus sophistiqués). [7,2]

Il traduit des valeurs de température, de position, de pression ... sous forme d'un signal évoluant entre deux valeurs limites.

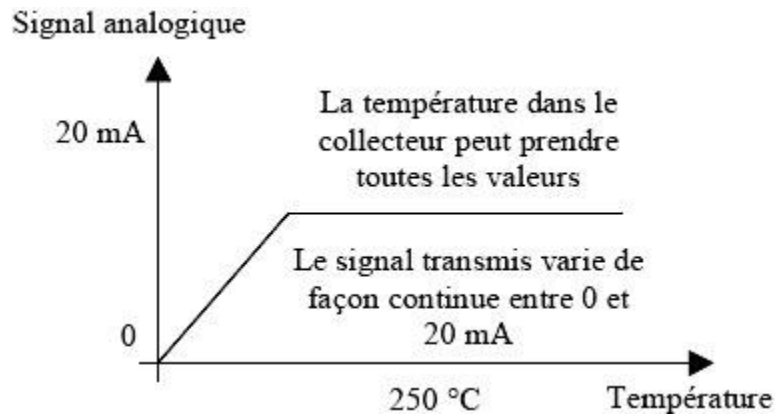


Fig.(I.4) Mode de sortie d'un capteur analogique.

c)Le Capteur numérique :

Ce type de capteur délivre en sortie une information électrique à caractère numérique, image de la grandeur physique à mesurer, c'est-à-dire ne pouvant prendre qu'un nombre limité de valeurs distinctes.

L'évolution étant à la miniaturisation et vers les traitements électroniques micro programmée, ces capteurs quoique encore relativement coûteux sont appelés à se développer.

L'information délivrée par ces capteurs peut être représentée par :

- Soit un signal électrique périodique (signal « carré ») à période variable caractérisant la grandeur physique à capter.
- Soit un signal numérique codé sur (n) variables binaires caractérisant la grandeur physique le protocole de codage étant défini par le fabricant du capteur. [9]

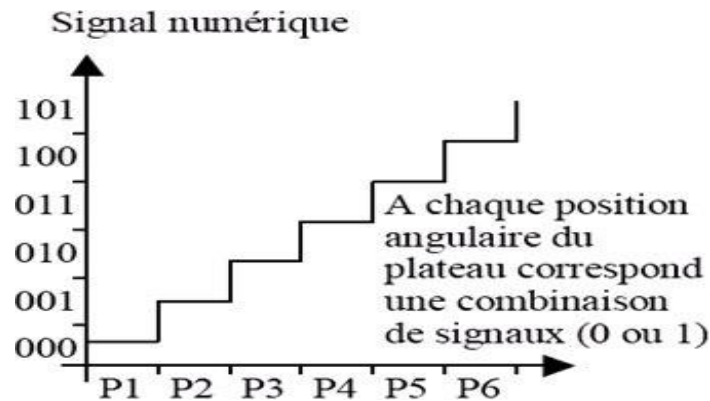


Fig.(I.5): Mode de sortie d'un capteur numérique

I.6.3. Les caractéristiques des capteurs :

On peut alors classer les capteurs en deux catégories, les capteurs à contact qui nécessitent un contact direct avec l'objet à détecter et les capteurs de proximité. Chaque catégorie peut être subdivisée en trois catégories de capteurs : les capteurs mécaniques, électriques, pneumatiques. [7]

Les principales caractéristiques des capteurs sont :

a) L'étendue de la mesure : c'est la différence entre le plus petit signal détecté et le plus grand perceptible sans risque de destruction pour le capteur.

b) La sensibilité : c'est la plus petite variation d'une grandeur physique que peut détecter un capteur.

c) La rapidité : c'est le temps de réaction d'un capteur entre la variation de la grandeur physique qu'il mesure et l'instant où l'information prise en compte par la partie commande.

D) La précision : c'est la capacité de répétabilité d'une information position, d'une vitesse.

I.6.4.Choix d'un capteur :

Tous les capteurs dont les fonctionnements ont été décrits précédemment présentent deux parties distinctes. Une première partie qui a pour rôle de détecter un événement et une deuxième partie qui a pour rôle de traduire événement en un signal compréhensible d'une manière ou d'une autre par une partie PC. Pour choisir correctement un capteur, il faudra définir tout d'abord :

- Le type événement à détecter.
- La nature d'événement.
- La grandeur de l'événement.
- L'environnement de l'événement.

En fonction de ces paramètres on pourra effectuer un ou plusieurs choix pour un type de détection. D'autres éléments peuvent permettre de cibler précisément le capteur à utiliser :

- Ses performances.
- Son encombrement.
- Sa fiabilité (MT BF).
- La nature du signal délivré par le capteur (électrique, pneumatique) ;
- Son prix..

I.7. Domaines de l'automatisme :

I.7.1.Commande de machines et d'installation :

C'est le domaine principal d'application des automates programmables industriels et c'est dans ces domaines qu'ils jouent le rôle principal dans la commande. On parle de commande d'installation lorsque les appareils sont fixés au bâtiment où en font partie. Ces commandes ont plutôt de grandes capacités en signaux, mais le processus est relativement lent. Exemples : installations de chauffages, d'éclairage, de transport.

Robotique : La robotique fait l'objet de commandes spécifiques. Elles sont spécialisées dans le positionnement d'axes ont de grandes capacités de calcul pour réaliser des déplacements divers. Leurs langages et moyens de programmation divers à la mise en oeuvre des robots. Les petites fonctions de commande directement liées au robot sont également réalisables. Pour des fonctions

plus complexes, il faut faire appel à un automate programmable qui peut être sous forme de modules compatibles à la commande du robot.

Régulation : On rencontre des AP qui offrent, au niveau du jeu d'instruction, des fonctions de régulation (PID par exemple) toutes faites. Si ce n'est pas le cas, on peut les réaliser à partir de

fonctions mathématiques, si elles existent. L'automate doit généralement recevoir des signaux pour la surveillance et la signalisation. Il est donc intéressant de réaliser également la régulation à partir de ces signaux déjà captés.

Communication et traitement de données : Elle permet l'échange d'informations entre un robot, un programmable, une commande d'axes, etc. La mémorisation et traitement de données sont généralement réalisés par un ordinateur qui devra forcément communiquer avec les différentes commandes d'une installation automatisée.

Commande d'axes : C'est un problème de régulation particulier de part sa vitesse de traitement et la précision de positionnement. C'est en fait le problème résolu dans les commandes de robots et les commandes numériques. Dans une application complexe, plusieurs domaines se côtoient et s'interfèrent. [10]

I.8. les solutions programmées :

Un automatisme utilisant la logique programmée se présente sous la forme d'un ordinateur, d'un micro-ordinateur ou d'un automate programmable. Son fonctionnement dépend du programme enregistré dans une mémoire. Elles éliminent tous les relais auxiliaires, les relais temporisés, les pendules et toute la filerie assurant les liaisons entre ces différents éléments. L'encombrement se trouve réduit et la recherche de panne est facilitée. Il ne reste dans l'armoire électrique que : [10]

- L'automate programmable.
- Les sectionneurs.
- les pré-actionneurs (contacteur, discontacteurs).

- les fusibles.
- les transformateurs (si commande en **24V**).
- le bornier.

I.8.1. Inconvénients:

- Utilisation d'un personnel formé à cette technologie .
- Il n'y a peu ou pas de gain de place dans l'armoire électrique, lorsque le fonctionnement est simple (peu de relais).
- le coût de la réalisation reste élevé si le fonctionnement de l'installation est simple.

I.8.2. Avantages:

- encombrement de l'armoire électrique réduit lorsque la complexité de l'installation augmente.
- main-d'oeuvre réduite lors du câblage.
- terminal de programmation pouvant être commun à plusieurs automates.
- modifications possibles sans intervention sur le câblage (à partir d'un terminal de programmation).
- dialogue avec l'installation (diodes électroluminescentes, terminal de programmation, imprimante, etc.).

I.9. Conclusion :

Les outils câblés caractérisés par une mise en œuvre nécessitant uniquement, mais nécessairement, l'établissement de liaisons matérielles (câblage) selon un schéma fourni par la théorie ou l'expérience. Ces outils largement utilisés dans l'industrie où l'on apprécie leurs qualités éprouvées, mais ils souffrent cependant d'un certain nombre de limitations parmi lesquelles nous retiendrons:

- leur encombrement (poids et volume).
- leur manque de souplesse vis-à-vis de la mise au point des commandes et de l'évolution de celles-ci (améliorations, nouvelles fonctions,...).
- la difficulté de maîtriser des problèmes complexes.
- la complexité de recherche des pannes et donc du dépannage.

Alors que les solutions programmées offre une alternative technologique à l'automaticien, et lui ouvre des possibilités nouvelles liées à la puissance de traitement des données. Parmi ces solutions programmées est le « A.P.I » (Automate Programmable Industriel) qui s'est substitué aux armoires à relais de sa souplesse (mise en œuvre , évolution), mais aussi parce que dans les automatismes de commande complexe, les coûts de câblage et de mise au point devenaient trop élevés.

CHAPITRE II

Les Automates

Programmables

Industriels

II.1. INTRODUCTION:

L'industrie moderne que, l'on peut qualifier d'industrie de qualité et de quantité, ne cesse d'exiger un matériel de contrôle de plus en plus performant afin de réaliser les deux objectifs, simultanément. Et c'est pour cette raison qu'on voulait remplacer les dispositifs de commande classiques avec tous les inconvénients qui en découlent (logique câblée très compliquée, encombrement, difficulté d'entretien ...) par un autre beaucoup plus performant et avantageux.

Ce serait certainement l'Automate Programmable Industriel qui devient de nos jours le cœur de toute unité industrielle moderne.

Voici donc une description plus ou moins détaillée de l'API et de tout ce qui y est lié en terme de soft et de hard. [1]

II.2. HISTORIQUE:

Au début des années 50, les ingénieurs étaient déjà confrontés à des problèmes d'automatismes, les composants de base de l'époque étaient les relais électromagnétique à un ou plusieurs contacts. Les circuits conçus comportaient des centaines voire des milliers de relais. Le transistor n'était connu que comme un composant d'avenir et les circuits intégrés étaient inconnus.

Vers 1960, les semi-conducteurs (transistors, diodes) sont apparus dans les automatismes sous forme de circuits digitaux. Ce n'est quelques années plus tard, que l'apparition des circuits intégrés a amorcé une révolution dans la façon de concevoir les automatismes. Ceux-ci étaient très peu encombrants et leur consommation était des plus réduite. On pouvait alors concevoir des fonctions de plus en plus complexes à des coûts toujours décroissants.

C'est en 1969 que les constructeurs américains d'automobiles (General Motors en particulier) ont demandé aux firmes fournissant le matériel d'automatisme des systèmes plus évolués et plus souples pouvant être modifiés simplement sans coût exorbitants.

Les ingénieurs américains ont résolu le problème en créant un nouveau type de produit nommé automates programmables. Ils n'étaient rentables que pour des installations d'une certaine complexité, mais la situation a très vite changée, ce qui a rendu les systèmes câblés obsolètes. [1]

De nombreux modèles d'automates sont aujourd'hui disponibles ; depuis les nano automates bien adaptés aux machines et aux installations simples avec un petit nombre d'entrées/sorties, jusqu'aux automates multifonctions capables de gérer plusieurs milliers d'entrées/sorties, et destinés au pilotage de processus complexes.

II.3. DEFINITION GENERALE :

Un Automate Programmable Industriel API dit Programmable Logic Controller PLC dans le langage industriel, est une machine électronique spécialisée dans la conduite et la surveillance en temps réel de processus industriels et tertiaires. Il exécute une suite d'instructions introduites dans ses mémoires sous forme de programme, et s'apparente par conséquent aux machines de traitement de l'information.

Trois caractéristiques fondamentales le distinguent des outils informatiques tels que les ordinateurs utilisés dans les entreprises et le tertiaire :

- il peut être directement connecté aux capteurs et pré actionneurs grâce à ses Entrées/sorties industrielles.
- il est conçu pour fonctionner dans des ambiances industrielles sévères, (Température, vibration, micro-coupures de la tension d'alimentation, Parasite, etc.).
- enfin, sa programmation à partir de langages spécialement développés pour le Traitement de fonctions d'automatisme facilite son exploitation et sa mise en oeuvre

Selon la norme française EN 61131-1, l'automate programmable est une :

« Système électronique fonctionnant de manière numérique, destiné à être utilisé dans un environnement industriel, qui utilise une mémoire programmable pour Le stockage interne des instructions orientées utilisateurs aux fins de mise en oeuvre des fonctions spécifiques, telles que des fonctions de logique, de mise en séquence, de temporisation, de comptage, et de calcul arithmétique, pour commander au moyen d'entrées et de sorties tout ou rien ou analogiques divers types de machines ou de processus. L'automate programmable et ses périphériques associés sont conçus pour pouvoir facilement s'intégrer à un système d'automatisme industriel et être facilement utilisés dans toutes leurs fonctions prévues. » [11]

A partir de ces définitions, nous distinguerons dans les fonctions que l'automate doit remplir :

- un rôle de **commande** où il est un composant d'automatisme, élaborant des actions, suivant une algorithmique appropriée, à partir des informations que lui fournissent des détecteurs (Tout ou Rien) ou des capteurs (analogiques ou numériques) .
- un rôle de **communication** dans le cadre de la production :
 - avec des opérateurs humains : c'est le dialogue d'exploitation,
 - avec d'autres processeurs, hiérarchiquement supérieurs (calculateur de gestion de production), égaux (autres automates intervenant dans la même chaîne) ou inférieur (instrumentation intelligente).

Autre définition:

D'après la norme NFC 63-850 l' automate programmable est une :

« Appareil électrique qui comporte une mémoire programmable par un utilisateur automatique (non informaticien) à l'aide d'un langage adapté, pour le stockage interne des instruction composant les fonctions d'automatisme comme par exemple :

- Logique séquentielle et combinatoire.
- Temporisation, comptage, décomptage, comparaison.
- Calcul arithmétique.
- Réglage, asservissement, régulation, etc. pour commander, mesuré et contrôler au moyen de modules d'entrées et de sorties (logique, numérique, ou analogique) différentes sorties de machines ou de processus, en environnement industriel ».

II.4. Caractéristiques des automates programmables :

Grâce à l'évolution constante de l'électronique et de l'informatique, les performances des automates programmables croissent sans cesse. On ne peut toutefois pas prétendre qu'un automate programmable peut tout faire et sera toujours la solution à un problème.

On a déjà vu que dans certains domaines de l'automation spécifiques étaient utilisées. Il est donc intéressant de connaître quelles sont les caractéristiques qui fixent les limites des performances d'un automate programmable industriel.

On peut dire que sauf dans des cas particuliers, c'est :

- La capacité d'entrées/sorties

- La capacité de programme
- La capacité de mémorisation de données
- Le temps de cycle et le temps de réaction
- Les possibilités du jeu d'instruction
- Les possibilités de communication

II.5. Les Objectifs:

Les Objectifs d'automate programmable sont :

* Simplifier le travail de l'homme qui, libéré de la machine, peut se consacrer à des activités plus qualifiantes.

* Améliorer la productivité des installations en augmentant leur autonomie de fonctionnement.

* Renforcer la qualité par une surveillance permanente des productions et augmente la sécurité des installations.

* Réaliser, notamment grâce au développement des techniques programmables, de l'installation plus souple, plus flexible d'emploi et susceptible de former des productions différentes par simple changement de programme.

En effet, l'automatisme peut intervenir à tous les stades d'opérations industrielles, dans des domaines aussi divers que les industries de transformations, de fabrication ou de transport, dans les machines-outils ainsi que dans la vie quotidienne (systèmes d'alarmes, automatisation de parking...). [12]

II.6. Architecture des automates programmables industriels:

De forme compacte ou modulaire, les automates sont organisés suivant l'architecture suivante :

- Un module d'unité centrale ou CPU, qui assure le traitement de l'information et la gestion de l'ensemble des unités. Ce module comporte un microprocesseur, des circuits périphériques de gestion des entrées/sorties, des mémoires RAM et EEPROM nécessaire pour stocker les programmes, les données, et les paramètres de configuration du système.

- Un module d'alimentation qui, à partir d'une tension 220V/50Hz ou dans certains cas de 24V fournit les tensions continues +/- 5V, +/-12V ou +/- -15V.

- Un ou plusieurs modules d'entrées 'Tout ou Rien' ou analogiques pour l'acquisition des informations provenant de la partie opérative (procède a conduire).
- Un ou plusieurs modules de sorties 'Tout ou Rien' (TOR) ou analogiques pour transmettre a la partie opérative les signaux de commande. Il y a des modules qui intègrent en même temps des entrées et des sorties.
- Un ou plusieurs modules de communication comprenant :
 - Interfaces série utilisant dans la plupart des cas comme support de communication, les liaisons RS-232 ou RS422/RS485
 - Interfaces pour assurer l'accès a un bus de terrain ;
 - Interface d'accès a un réseau Ethernet.

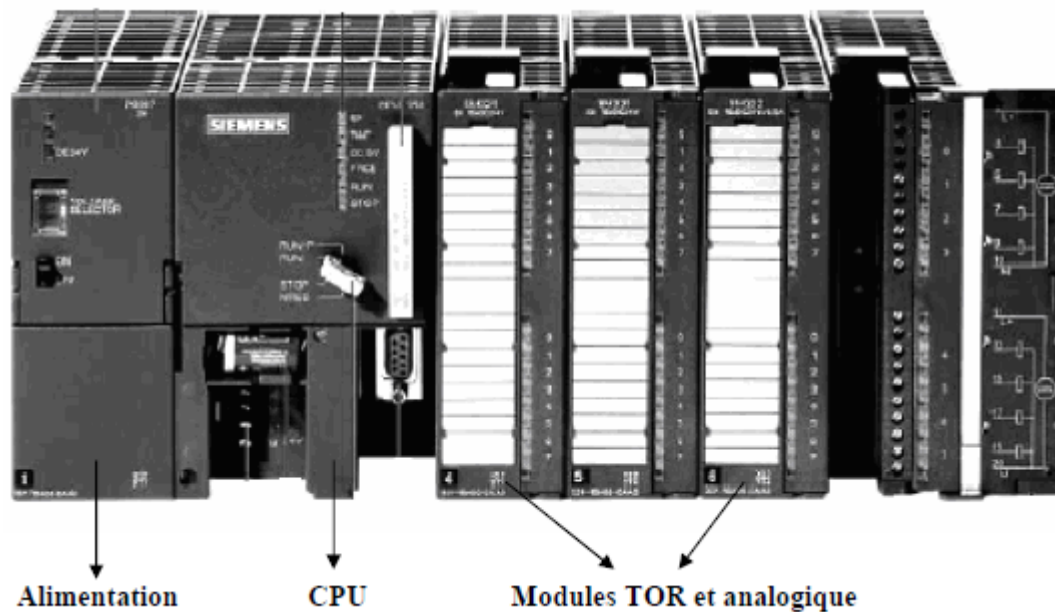


Fig.(II.1): L'automate programmable Siemens S7300

II.7. Architecture interne d'un automate programmable:

La structure matérielle interne d'un API obéit au schéma donné sur la figure dessous:

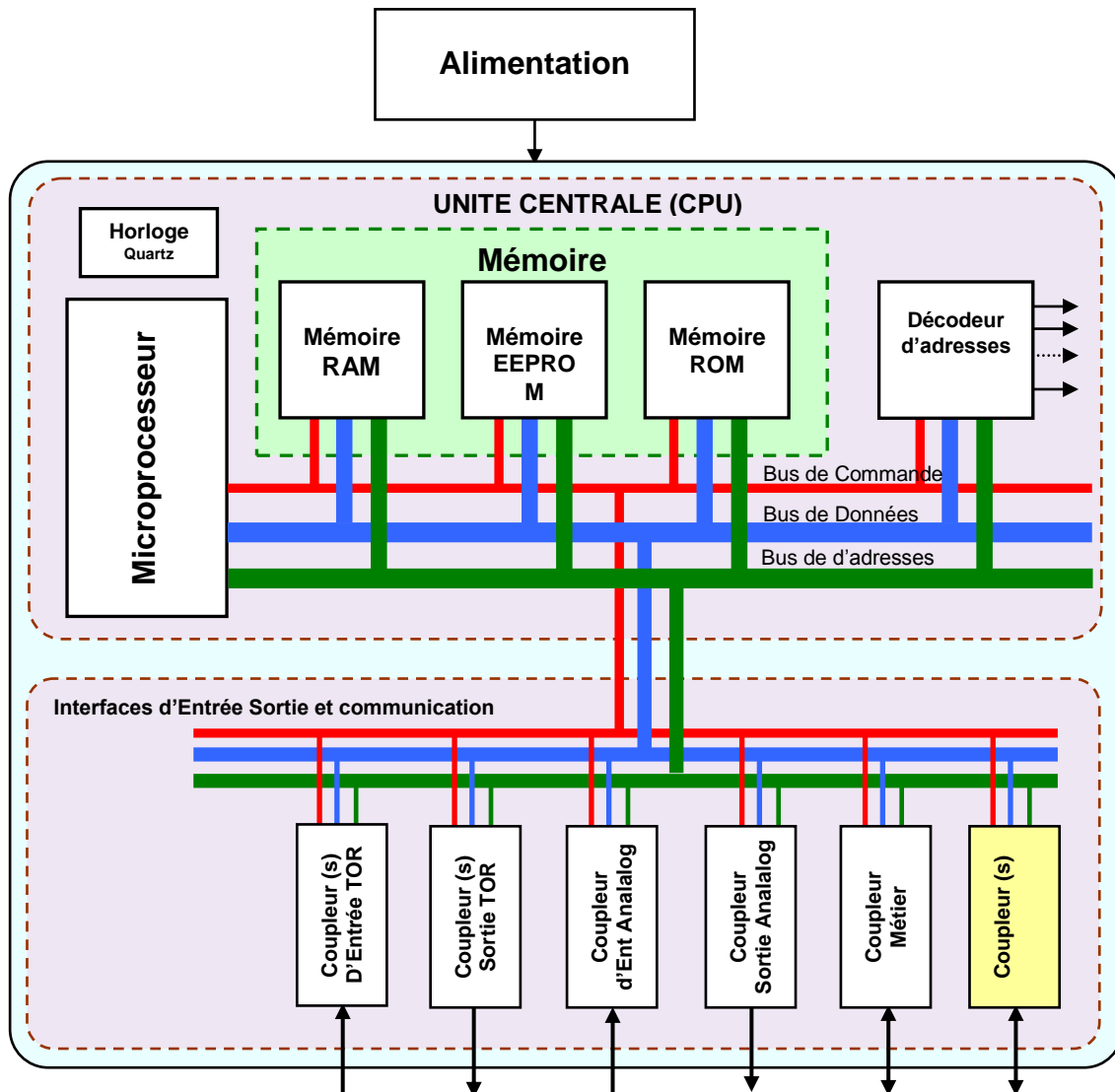


Fig.(II.2) : Architecture interne d'un automate programmable

Nous analyserons successivement chacun des composants qui apparaissent sur ce schéma :

II.7.1. Le Processeur :

Le processeur a pour rôle principal le traitement des instructions qui Constituent le programme de fonctionnement de l'application. Mais en dehors De cette tâche de base, il réalise également d'autres fonctions :

- Gestion des entrées/sorties.
- surveillance et diagnostique de l'automate par une série de teste lancés à la mise sous tension ou cycliquement en cours de fonctionnement.
- Dialogue avec le terminal programmation aussi bien pour l'écriture et la mise au point du programme qu'en cours d'exploitation pour des réglages ou des vérifications de données.

Le processeur est organisé autour d'un certain nombre de registres, ce sont des mémoires rapides permettant la manipulation des informations qu'elles retiennent, ou leur combinaison avec des informations extérieures.

Les principaux registres existants dans un processeur sont :

II.7.1.1. L'accumulateur :

C'est le registre où s'effectuent les opérations du jeu d'instruction, les résultats sont contenus dans ce registre spécial.

II.7.1.2. Le registre d'instruction :

Il reçoit l'instruction à exécuter et décode le code opération. Cette instruction est désignée par le pointeur.

II.7.1.3. Le registre d'adresse :

Ce registre reçoit, parallèlement au registre d'instruction, la partie opérande de l'instruction. Il désigne le chemin par lequel circulera l'information lorsque le registre d'instruction validera le sens et ordonnera le transfert.

II.7.1.4. Le registre d'état :

C'est un ensemble de positions binaires décrivant, à chaque instant, la situation dans laquelle se trouve précisément la machine.

II.7.1.5. Les piles :

Une organisation spéciale de registres constitue une pile, ces mémoires sont utilisées pour contenir le résultat de chaque instruction après exécution.

Ce résultat sera utilisé ensuite par d'autre instruction, et cela pour faire place à la nouvelle information dans l'accumulateur.

II.7.2. Les mémoires :

Un système à processeur est toujours accompagné d'un ou de plusieurs types de mémoires. Les automates programmables industriels possèdent pour la plupart les mémoires suivantes :

II.7.2.1. Mémoire de travail :

La mémoire de travail (mémoire vive) contient les parties du programme significatif pour son exécution. Le traitement du programme a lieu exclusivement dans la mémoire de travail et dans la mémoire système.

II.7.2.2. Mémoire système :

La mémoire système (mémoire vive) contient les éléments de mémoire que chaque CPU met à la disposition du programme utilisateur comme, par exemple, mémoire image des entrées, mémoire image des sorties, mémentos, temporisation et compteurs. La mémoire système contient, en outre, la pile des blocs et la pile des interruptions. Elle fournit aussi la mémoire temporaire allouée au programme (pile des données locales).

II.7.2.3. Mémoire de chargement :

La mémoire de chargement sert à l'enregistrement du programme utilisateurs sans affectation de mnémonique ni de commentaires (ces derniers restent dans la mémoire de la console de programmation).

La mémoire de chargement peut être soit une mémoire vive (RAM), soit une mémoire EPROM.

II.7.2.4. Mémoire RAM non volatile :

Zone de mémoire configurable pour sauvegarder des données en cas de défaut d'alimentation.

II.7.2.5. Mémoire ROM :

Contient le système d'exploitation qui gère la CPU.

II.7.3. Les modules d'entrée/sortie :

Ils traduisent les signaux industriels en informations API réciproquement, appelées aussi coupleurs.

Beaucoup d'automates assurent cet interfaçage par des modules amovibles être modulaires par carte ou par rack. D'autres automates ont une structure monobloc, avec des modules intégrés dans un châssis de base, (cas des automates de télémécanique TSX17 et SIMATIC S7-314 IFM).

Le nombre total de modules est évidemment limité, pour des problèmes physiques :

- Alimentation électrique.
- Gestion informatique.
- Taille du châssis.

Différents types de modules sont disponibles sur le marché selon l'utilisation souhaitée, les plus répandus sont :

II.7.3.1. Entrées/sorties TOR (Tout ou Rien) :

La gestion de ce type de variables constituant le point de départ historique des API reste une de leurs activités majeures.

Leur nombre est en général de 8,16, 24 ou 32 entrées/sorties, qui peuvent fonctionner :

- En continu : 24V, 48V.
- En alternatif : 24V, 48V, 100/120V, 220/240V

II.7.3.2. Entrées/Sorties analogique :

Elles permettent l'acquisition de mesures (entrées analogique), et la commande (sorties analogiques). Ces modules comportent un ou plusieurs convertisseurs Analogique/Numérique (A/N) pour les entrées et Numérique / Analogique (N/A) pour les sorties dont la résolution est de 8 à 16 bits.

Les standards les plus utilisés sont : $\pm 10V$, $0-10V$, $\pm 20mA$, $0-20mA$ et $4-20mA$.

Ces modules sont en général multiplexés en entrée pour n'utiliser qu'un seul convertisseur A/N, alors que les sorties exigent un convertisseur N/A par voie pour pouvoir garder la commande durant le cycle de l'API.

II.7.3.3. Module spécialisés :

Ils assurent non seulement une liaison avec le monde extérieur, mais aussi une partie du traitement pour soulager le processeur et donc améliorer les performances.

Ces modules peuvent posséder un processeur embarqué ou une électronique spécialisée.

On peut citer :

- **Les cartes de comptage rapide :**

Elles permettent saisir des événements plus courts que la durée du cycle travaillant à des fréquences qui peuvent dépasser 10kHz.

- **Les Entrées/Sorties déportées :**

Leur intérêt est de diminuer le câblage en réalisant la liaison avec détecteur, capteurs ou actionneurs au plus près de ceux-ci, ce qui a pour effet d'améliorer la précision de mesure.

La liaison entre le boîtier déporté et l'unité centrale s'effectue par le biais d'un réseau de terrain suivant des protocoles bien définis.

L'utilisation de la fibre optique permet de porter la distance à plusieurs kilomètres.

II.7.4. L'alimentation électrique :

Elle a pour rôle de fournir les tensions continues nécessaires aux composants avec de bonnes performances, notamment face aux micro-coupures du réseau électrique qui constitue la source d'énergie principale.

La tension d'alimentations peut être de 5V, 12V ou 24V.

D'autres alimentations peuvent être nécessaires pour les châssis d'extension et pour les modules entrées/sorties.

Un onduleur est nécessaire pour éviter les risques de coupures non tolérées.

II.7.5. Les liaisons :

Elles s'effectuent :

- avec l'extérieur par des borniers (à vitesse, à clipser, ...), sur lesquels arrivent des câbles transportant les signaux électriques.
- avec l'intérieur par des bus, liaisons parallèles entre les divers éléments. Il existe plusieurs types de bus, car on doit transmettre des données, des états, des adresses.

II.7.6. Eléments auxiliaires :

- Un ventilateur: est indispensable dans les châssis comportant de nombreux modules ou dans le cas où la température ambiante est susceptible de devenir assez élevée.
- Un support mécanique : il peut s'agir d'un rack, l'automate se présente alors.
- Sous forme d'un ensemble de cartes, d'une armoire, d'une grille, et des fixations correspondantes.
- Des indicateurs d'état : concernant la présence de tension, la charge de la batterie, le bon fonctionnement de l'automate etc. [13]

II.8. PROTECTION DE L'AUTOMATE:

La protection des circuits d'entrée contre les parasites électriques est souvent résolue par découplage optoélectrique. Le passage des signaux par un stade de faisceau lumineux assure en effet une séparation entre les circuits internes et externes.

Du côté sorties, on doit assurer le même type de protection, mais amplification de puissance, avec au final un courant continu ou alternatif selon les cas.

Deux types de cartes électroniques sont utilisés :

II.8.1. Les modules à sorties statiques :

Relais statiques intégrant des composants spécialisés : transistor bipolaires, thyristors. Ces composants n'ont aucune usure mécanique et leurs caractéristiques de commutation se maintiennent dans le temps.

II.8.2. Les modules à relais électromagnétique :

Où le découplage résulte de l'existence de deux circuits électriques (Bobines d'excitation, circuits de puissance), ces relais électromagnétique ont l'avantage d'avoir une faible résistance de contact, une faible capacité de sortie et surtout un faible coût, mais une durée de vie et une vitesse de commutation inférieures aux sorties statiques.

II.9. ENVIRONNEMENT:

Dans le cadre d'une évolution conduisant à une automatisation de plus en plus globale, l'automate est de moins en moins acquis seul, et même si c'est le cas, il doit pouvoir se connecter à d'autres matériels à processeur, et dialoguer avec les agents d'exploitation.

Les types de communication supportés par les API modernes sont : la communication avec un opérateur par un pupitre ou un terminal industriel : ils permettent une communication homme-machine, et ce dans les deux sens (clavier, ...) [12]

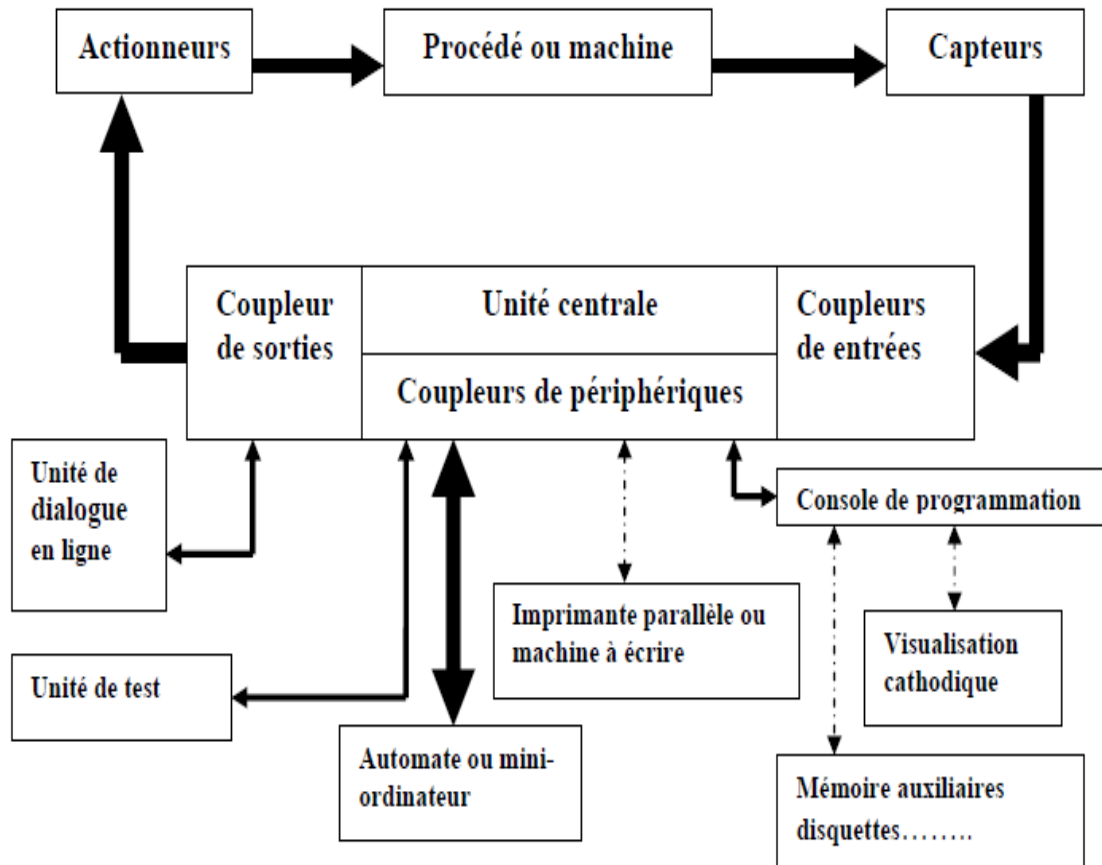


Fig.(II.3):L'environnement de l'API

II.10. LES LANGAGES DE PROGRAMMATION:

II.10.1 Introduction :

La norme IEC 1131-3 (la Commission Électrotechnique Internationale) définit cinq langages qui peuvent être utilisés pour la programmation des automates programmables industriels. Ces langages peuvent être divisés en deux grandes catégories : [14]

• Langages graphiques:

- SFC « Sequential Funiculite Chart » ou GRAFCET
- LD « Ladder Diagram »ou schéma à relais
- FBD « Function Block Diagram »ou schéma par bloc

• Langues textuels:

- ST « structured text » ou texte structuré
- IL « Instruction List » ou Liste d'instructions

II.10.2. Les langages graphiques:**II.10.2.1. Le GRAFCET:**

L'acronyme GRAFCET signifie: GRAPhe Fonctionnel de Commande Etape Transition (SFCSequential Function Chart). C'est une méthode de représentation graphique permettant de décrire le cahier de charge d'un automatisme. Il est adapté aux systèmes à évolution séquentielle, il est défini par un ensemble d'éléments graphiques de base traduisant le comportement de la partie commande vis-à-vis de ses entrées et de ses sorties.

Un programme GRAFCET décrit un procédé comme une suite d'étapes, reliées entre elles par des transitions. À chaque transition est associée une réceptivité, celle-ci est une condition logique qui doit être vraie pour franchir la transition et passer à l'étape suivante. Des actions sont associées aux étapes du programme.

Le format graphique d'un programme GRAFCET est le suivant :

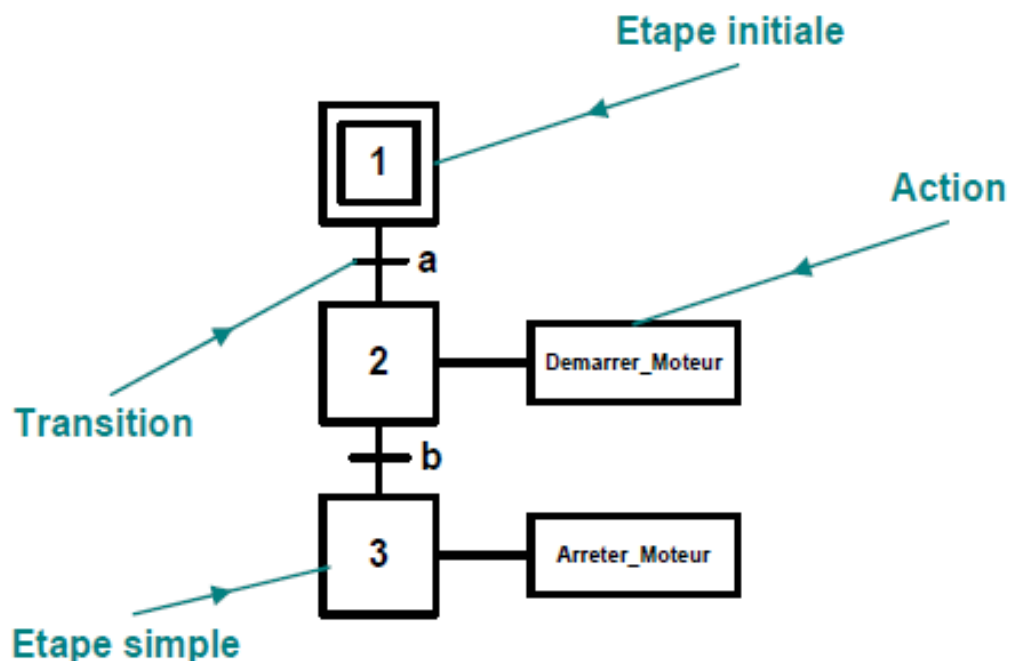


Fig.(II.4): Le GRAFCET

Une étape représentée par un carré qui a un numéro identificateur et les actions associées sont indiquées dans un rectangle relié à la partie droite du carré; (l'étape initiale est représentée par un carré double).

Une liaison orientée représentée par une ligne, parcourue par défaut de haut en bas ou de gauche à droite.

Une transition entre deux étapes à qui est associé une réceptivité inscrite à ça droite, est représentée par une barre perpendiculaire aux liaisons orientées qui relient ces étapes.

II.10.2.2 Ladder Diagram :

Le LD est une représentation graphique qui traduit directement des équations booléennes en un circuit électrique en combinant des contacts et des relais à l'aide des connexions horizontales et verticales, les contacts représentent les entrées (contact normalement ouvert, contact normalement fermé,...) et les relais représentent les sorties (relais directs, relais inversés,...), les diagrammes LD sont limités sur la gauche par une barre d'alimentation et par la masse sur la droite.

Par exemple pour réaliser la fonction logique: $x = (a + b)(c + \bar{d}e)$

voici le programme LD :

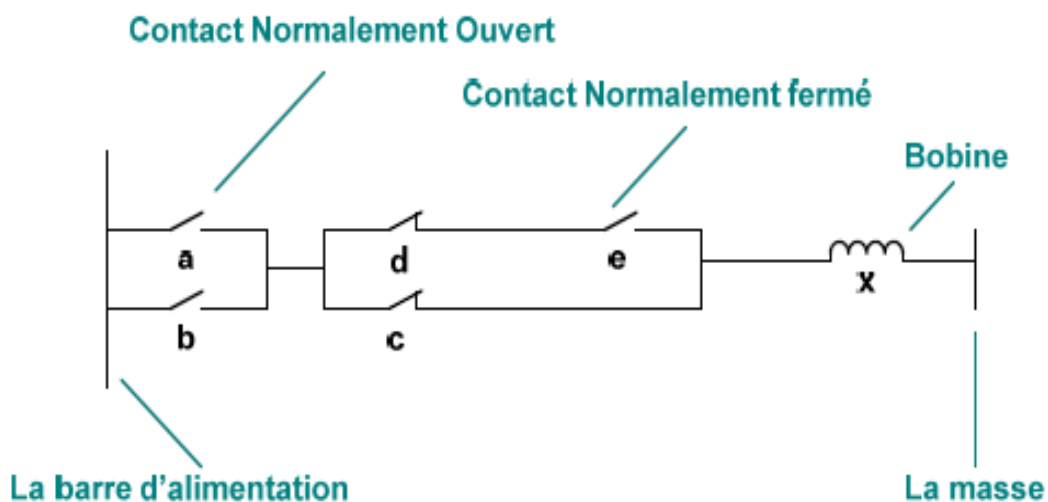


Fig.(II.5): exemple d'un programme LADDER

Le langage LD propose d'autre type de fonction tel que les fonctions de comptages et de temporisations, les fonctions arithmétiques et logiques, les fonctions de comparaison et de transfert. Par exemple pour réaliser la fonction: $z = (x \geq y)$, on utilise directement la fonction déjà disponible.

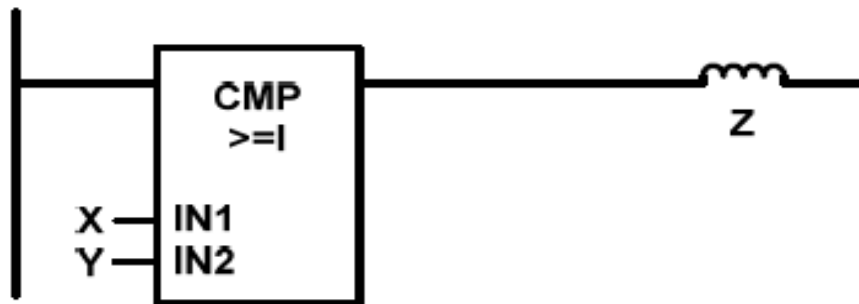


Fig.(II.6): exemple d'un programme LADDER

II.10.2.3 Bloc de Fonction:

C'est un langage graphique qui permet la construction d'équations complexes à partir des opérateurs standard, ou de blocs fonctionnels, il se compose de réseaux de fonctions préprogrammées ou non, représentées par des rectangles qui sont connectés entre eux par des lignes.

La programmation avec le FBD est très souple et facile à apprendre, la plupart des fonctions nécessaires (les fonctions arithmétique et logique, les fonctions de temporisation, des blocs fonctionnels PID...) sont déjà disponible dans la bibliothèque, il suffit juste de les connecter et bien paramétrer les entrées et les sorties, c'est-à-dire respecter le type des variables lors de la connexion.

Par exemple pour réaliser l'opération arithmétique : $\omega = \frac{20(x + y)}{z}$.

on aura besoin de trois blocs : un pour addition, un pour la multiplication et un autre pour la division.



Fig.(II.7): exemple d'un programme en Fonction Bloc

II.10.3. Les langages textuels:

II.10.3.1 Texte Structuré:

Le langage ST (Structured Text) est un langage de programmation textuel de haut niveau dédié aux applications d'automatisation, il est utilisé principalement pour décrire les procédures complexes et difficilement modélisables avec les langages graphiques, il peut aussi être utilisé en tant que sous programme avec d'autre langage de programmation.

Il utilise les même énoncés que les langages de programmation de haut niveau (Pascal, C, C++...) comme: les assignations, les appels de fonction, les énoncés de contrôle (IF, THEN, ELSE, CASE) ou d'itération (FOR, WHILE, REPEAT) en plus des opérations arithmétiques et logiques.

Exemple : calcul de l'écart entre deux points dans un plan cartésien.

```

FUNCTION Ecart: REAL
VAR_INPUT
X1, X2, Y1, Y2 : REAL;
END_VAR
BEGIN
RESULT := SQRT((X1-X2)^2 + (Y1-Y2)^2);
END_FUNCTION
  
```

Fig.(II.8): exemple d'un programmation en langage Structuré.

II.10.3.2 Liste d'instructions:

Le langage IL est un langage textuel de bas niveau (proche du langage machine), qui utilise un jeu d'instruction simple, il trouve sa puissance dans les applications de petites tailles, et dans la création de sous programme ou procédure, car il permet un contrôle totale et une optimisation parfaite du code, par contre dans les grandes applications il est très difficile de programmer avec le IL, les programmes dans ce langage peuvent être traduit ou déduit des autres langages.

Le IL a la même structure que l'assembleur, il utilise un ou plusieurs registres de travail. Les valeurs intermédiaires nécessaires pour l'exécution d'une instruction donnée seront mémorisées dans ces registres le temps de leurs utilisations et il possède un jeu d'instruction d'assez riche pour décrire toutes les opérations arithmétiques et logiques, les opérations de comptage et temporisation, la comparaison et le transfert...

Par exemple pour réaliser l'opération $\omega = \frac{20(x + y)}{z}$, on utilise le code suivant :

```
LD X
LD Y
+R
LD +20
*R
LD Z
/R
ST W
```

Fig.(II.9): exemple d'un programme en IL

II.11. Conclusion:

L'automate est un bon produit, facile à programmer, à connecter, adapté aux conditions industrielles. L'expansion considérable de ses possibilités, et celle corrélative de son marché, le prouvent. Il ne faut pas vouloir en faire une solution miracle. Dans tous les cas :

- une bonne analyse du problème à résoudre.
- le respect des règles d'installation.
- un léger sur dimensionnement pour préserver des marges de modification.

Sont les conditions d'une implantation réussie, dont la durée de vie dépassera largement celles habituelles dans le monde informatique, dont l'API est pour partie issu.

La plus part des grands constructeurs d'automates programmables, fournissent des logiciels de configuration et de programmation munis des langages SFC, LD, FBD, ST et IL.

Le choix d'un langage s'appuie sur la complexité de l'application et de la tâche de commande. Il est préférable d'utiliser les langages graphiques (SFC, LD et FBD) pour la réalisation des programmes de commande séquentielles. Le SFC est la réalisation directe d'un GRAFCET de commande, les langages LD et FBD sont plus utiles pour les opérations combinatoires sur bits ou mots.

Les langages textuels sont beaucoup plus performants pour le traitement de variables continues ou analogiques ainsi que pour la commande des systèmes continus. Les programmes en IL sont un peu fastidieux à mettre en œuvre, mais connaissent une optimisation optimale pour le temps de traitement et l'occupation de la mémoire. Le ST est le langage par excellence, très utiles pour des utilisateurs ayant des connaissances en langages évolués tel que PASCAL.

CHAPITRE III

Automate S7 300

III .1. INTRODUCTION :

Un automate programmable est une machine électrique spécialisée par la commande et la surveillance en temps réel de processus industriels. Elle exécute une suite d'instruction introduite dans ses mémoires sous forme de programme, soit par une console de programmation ou un micro-ordinateur

Actuellement, dans le marché mondial il existe plusieurs types d'automates programmable, qui se différencient par des caractéristiques techniques.

Parmi lesquels on cite : Schneider, Omron, PCD1, SIEMENS, Télémécanique....etc.

Et comme notre travail s'effectue sur un automate programmable de type SIEMENS, donc une description lui sera consacrée.

III. 2. LES AUTOMATES PROGRAMMABLES INDUSTRIEL (A.P.I) SIEMENS :

SIEMENS est l'une des sociétés très connues dans le domaine de la fabrication des automates programmables, et reste le seul à proposer une gamme complète de produits pour l'automatisation industrielle, par le biais de sa gamme **SIMATIC**. L'intégration globale de tout l'environnement d'automatisation est réalisée grâce à :

- Une configuration et une programmation homogène des différentes unités du système.
- Une gestion cohérente des données.
- Une communication globale entre tous les équipements d'automatisme mis en œuvre.

SIEMENS a développé plusieurs types des A.P.I, et les plus connus sont la famille S5 (Step 5) et S7 (Step7). [21]

Permet les S5, Il y a: S5-90U, S5-95U, S5-100U, S5-115U, S5-135U et S5-155U.

En ce qui concerne les S7, SIEMENS a développé d'autres nouvelles générations :

S7-200, S7-300 et S7-400 La figure (1-1) montre un automate programmable de la famille S7 qui est S7-300.

III.3. Les différentes variantes dans la gamme SIMATIC: [16]

1. Automate S7-200:

S7-200 est une Micro-automate modulaire pour les applications d'entrée et de milieu de gamme, avec possibilité d'extensions jusqu'à 7 modules, et une mise en réseau par l'interface multipoint (MPI) ou *PROFEBUS*

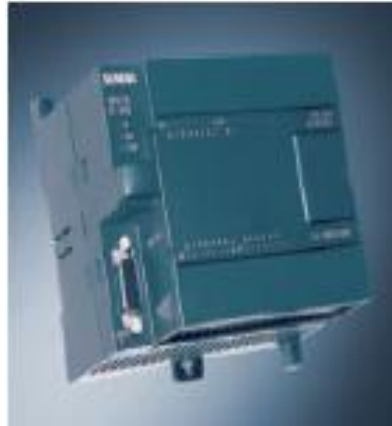


Fig.(III.1): L'API S200

2. Automate S7-300:

S7-300 est une Micro-automate modulaire pour les applications d'entrée et de milieu de gamme, avec possibilité d'extensions jusqu'à 32 modules, et une mise en réseau par l'interface multipoint (MPI) ou *PROFEBUS* et *INDUSTRIEL ETHERNET*



Fig.(III.2): L'API S300

3. Automate S7-400:

S7-400 est une automate de haute performance pour les applications de milieu et haut de gamme, avec possibilité d'extension à plus de 300 modules, et une mise en réseau par l'interface multipoint (MPI) ou *PROFEBUS* et *INDUSTRIEL ETHERNET*



Fig.(III.3): L'API S400

III.4. Présentation des spécification de l' API S7- 300 :

Le S7-300 est l'automate conçu pour des solutions dédiées au système manufacturier et constitue à ce titre une plate-forme d'automatisation universelle pour des applications avec des architectures centralisées et décentralisées. L'innovation est permanente et se manifeste par exemple dans le développement continu de la gamme des CPU avec entre autres des nouveaux modèles orientés sécurité, motion contrôle ou avec interface Ethernet/PROFIT intégrée. Le S7-300 trouve des applications dans des industries comme l'automobile, la construction OEM (Original Equipment Manufacturer), l'emballage, l'agro-alimentaire, la plasturgie. Il peut également s'intégrer dans des solutions compactes avec HMI (Interface Homme - Machine) ou dans des têtes de station pour traitement intelligent décentralisé.

III.4.1. Gamme de modules :

Le SIMATIC S7-300 est un système d'automatisation modulaire offrant la gamme de modules représentés par la figure (1.2) suivantes :

- Unités centrales (CPU) de capacités différentes, certaines avec d'entrées/sorties intégrées (ex : CPU314C) ou avec interface PROFIBUS intégrée (ex : CPU315-2DP).
- Modules d'alimentation PS (Power Supply) avec 2A, 5A ou 10A .
- Modules de signaux SM pour entrées et sorties numériques et analogiques .
- Modules de fonction FM pour fonctions spéciales (ex : pilotage d'un moteur pas à pas)
- Processeurs de communication CP pour la connexion au réseau .

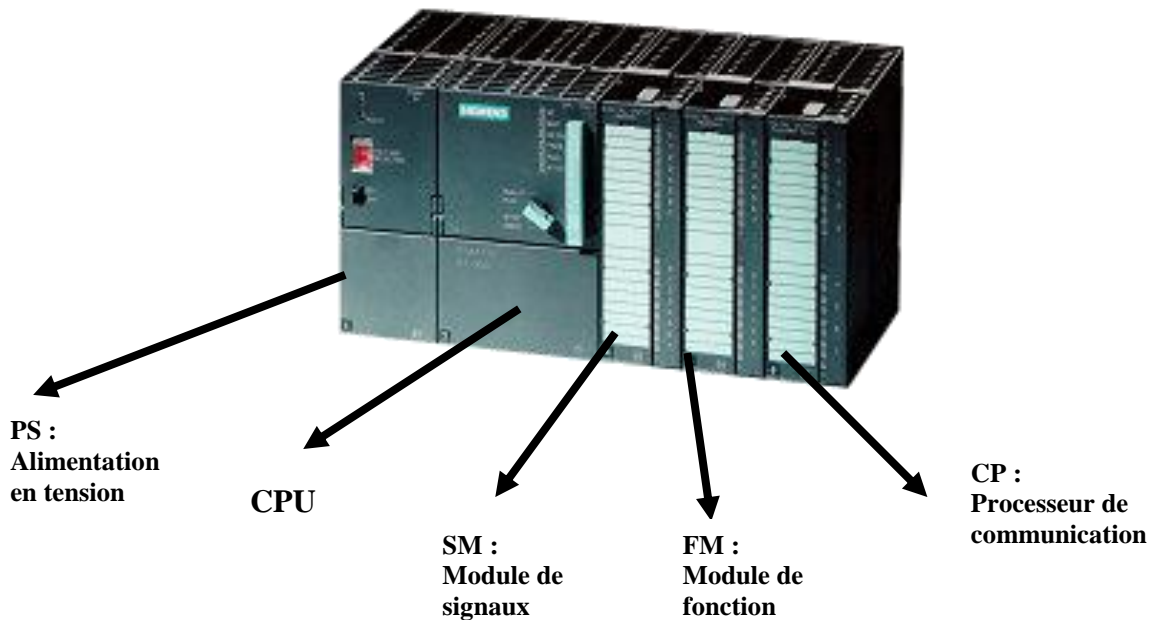


Fig.(III.4) : modules d'un API

III.4.1. le module central CPU :

Dans la CPU (Module central), le processeur traite le programme existant 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 le 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.

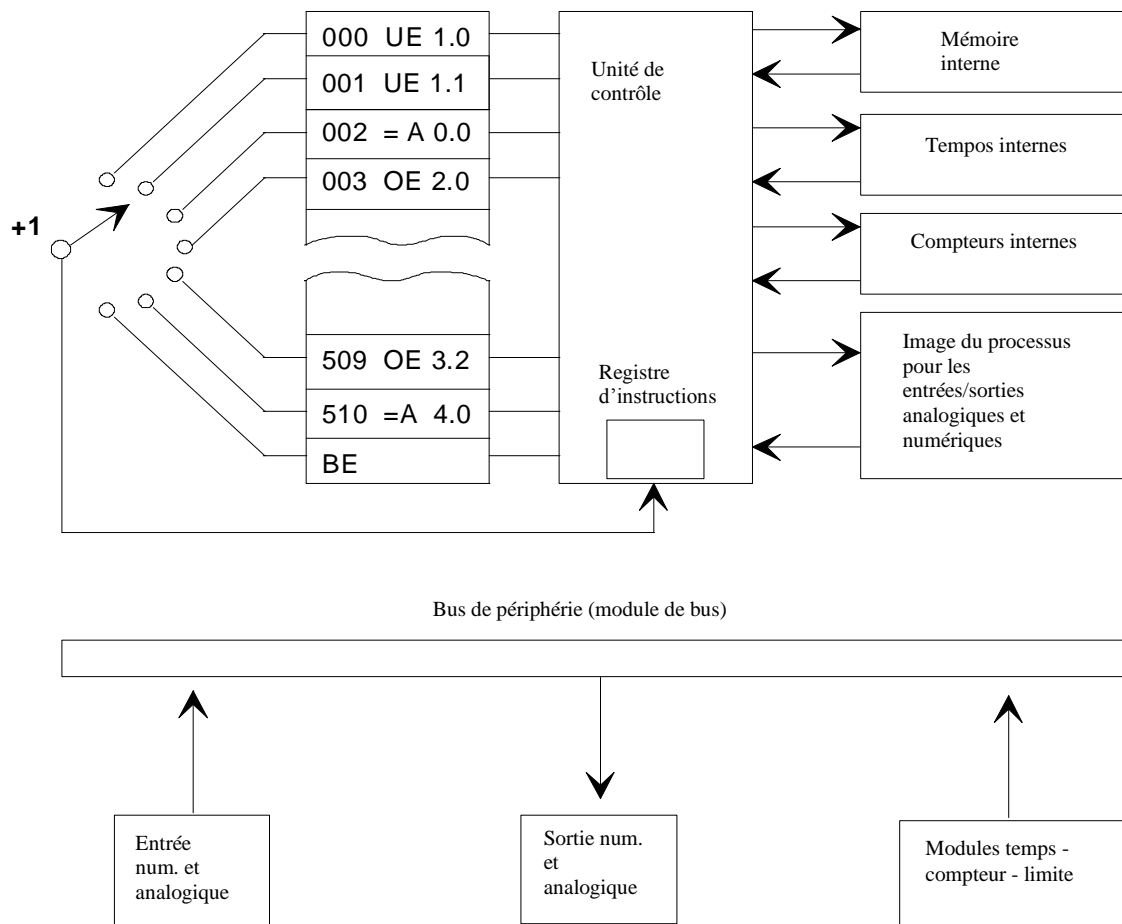


Fig.(III.5) : Unité centrale (CPU) de l'API

III.4.3. Adressage des signaux d'entrée /sortie:

La déclaration d'une entrée ou sortie donnée à l'intérieur d'un programme s'appelle l'adressage.

Les entrées et sorties des automates sont la plupart du temps regroupées en groupes de huit entrées ou sorties numériques. Cette unité de huit entrées ou sorties est appelée un **octet**. Chaque groupe reçoit un numéro que l'on appelle l'**adresse d'octet**.

Afin de permettre l'adressage d'une entrée ou sortie à l'intérieur d'un octet, chaque octet est divisé en huit **bits**. Ces derniers sont numérotés de 0 à 7. On obtient ainsi l'**adresse du bit**.

L'automate programmable représenté ici a les octets d'entrée 0 et 1 ainsi que les octets de sortie 4 et 5. [21]

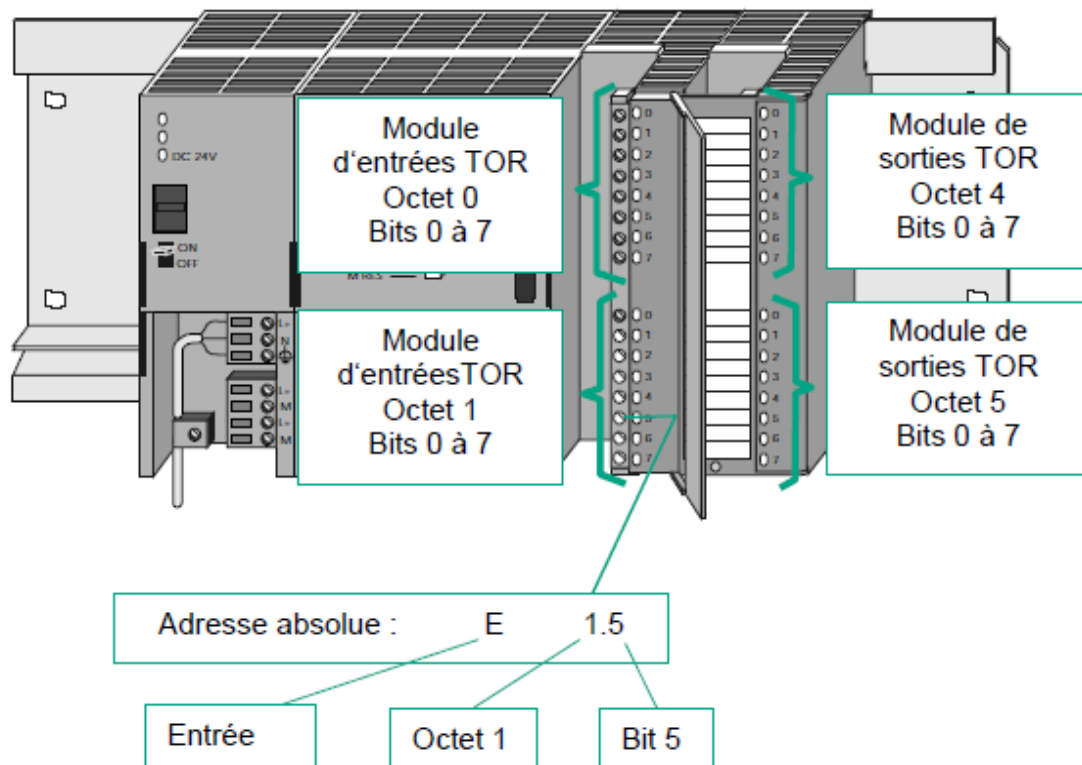


Fig.(III.6): entrée /sortie d'un API

Pour adresser par exemple la cinquième entrée à partir du haut, il faut entrer l'adresse

suivante : **E 1. 4**

E désigne le type de l'adresse : Entrée, **1** l'adresse de l'octet et **4** l'adresse du bit.

L'adresse de l'octet et l'adresse du bit sont toujours séparées par un point.

Pour adresser par exemple la dernière sortie, il faut entrer l'adresse suivante : **A 5 . 7**

A désigne le type de l'adresse : Sortie, **5** l'adresse de l'octet et **7** l'adresse .

III.5. LA CPU S7-314 IFM ET SES FONCTION INTEGREE:

III.5.1. INTRODUCTION:

Après avoir décrit le processus et les API d'une manière générale, nous allons au cours de ce chapitre choisir pour notre application un API et motiver ce choix.

Nous allons décrire toutes les fonctionnalités correspondantes à ce type d'API avec tous les détails que nous jugeons utiles.

III.5.2. PRESENTATION DE LA CPU

SIEMENS propose plusieurs gammes de produits qui peuvent être utilisés dans se projet. On a opté pour l'utilisation d'un automate programmable appartenant à la série S7-300 (S7-314 IFM). Sa caractéristique principale est l'intégration de modules comportant entre autres des fonctions intégrées.

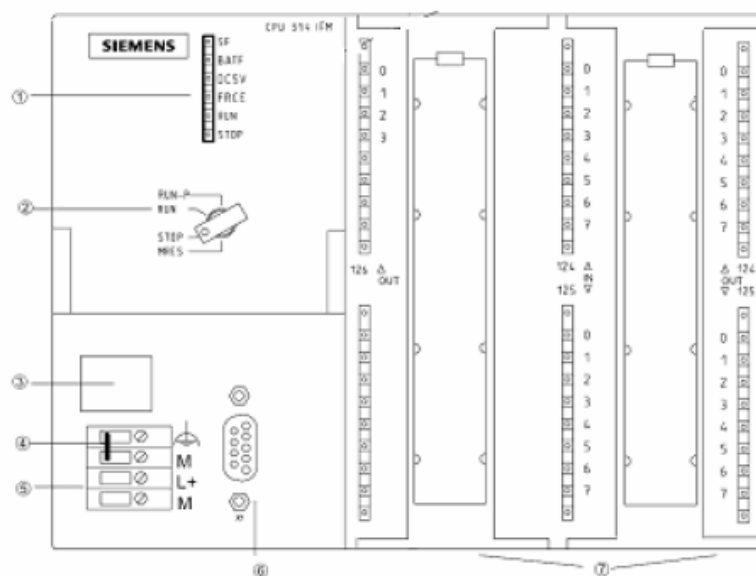


Fig.(III.7): Vue générale de la CPU S7-314 IFM.

Les CPU de la série S7-300 se composent des éléments suivants :

III.5.2.1. LED de visualisation d'état et de défaut :







	(Rouge) SF Défaut de matériel ou de logiciel.
	(Rouge) BATF Défaillance de la pile.
	(Verte) 5V cc L'alimentation 5V cc est correcte .
	(Jaune) FRCE Le forçage permanent est actif.
	(Verte) RUN CPU en RUN.
	(Jaune) STOP CPU en STOP ou ATTENTE ou en démarrage.

Fig.(III.8): LED de visualisation d'état de la CPU

III.5.2.2. Commutateur de mode de fonctionnement :

Le changement de mode se fait à l'aide d'une clé :

Position	Signification	Explication
RUN –P	Mode de fonctionnement RUN-PROGRAM	La CPU traite le programme utilisateur. Le programme peut être modifié. Dans cette position la Clé ne peut pas être retirée.
RUN	Mode de fonctionnement RUN	La CPU traite le programme utilisateur. Le programme ne peut être modifié qu'avec légitimation par mot de passe la Clé peut être retirée.
Position	Signification	Explication
STOP	Mode de fonctionnement	La CPU ne traite aucun programme utilisateur.
	STOP	La Clé peut être retirée.
MRES	Effacement Général	Position instable du commutateur, pour effectuer l'effacement général il faut respecter un ordre particulier de commutation

Tableau(III.1): Le commutateur de mode de fonctionnement de l'API

III.5.2.3. Pile de sauvegarde ou accumulateur :

L'utilisation de l'accumulateur ou de la pile de sauvegarde est nécessaire pour l'horloge temps réel.

La pile de sauvegarde est aussi utilisée pour :

- la sauvegarde du programme utilisateur s'il n'est pas enregistré dans la mémoire morte.
- pour étendre la zone rémanente de données.

L'accumulateur est rechargé à chaque mise sous tension de la CPU. Son autonomie est de quelques jours voir quelques semaines, au maximum. La pile de sauvegarde n'est pas rechargeable mais son autonomie peut aller jusqu'à une année.

III.5.2.4. Carte mémoire :

La CPU S7-314 IFM utilisée n'est pas dotée d'une carte mémoire mais il faut savoir que la plupart des CPU en possèdent, son rôle est de sauvegarder le programme utilisateur.

Le système d'exploitation et les paramètres qui déterminent le comportement de la CPU et des modules en cas de coupure de courant.

III.5.2.5. Interface MPI : (interface multipoint)

L'interface MPI est l'interface de la CPU utilisée pour la console de programmation (PG), le pupitre opérateur (OP) ou pour la communication au sein d'un sous réseau MPI., la vitesse de transmission typique est de 187,5 k bauds.

III.5.3. CARACTERISTIQUES TECHNIQUES DE LA CPU :

la CPU S7-314 IFM possède des entrées/sorties intégrées, leur câblage se fait par le biais de connecteurs frontaux 20 ou 40 points.

III.5.3.1. Tableaux techniques :

Les tableaux ci-dessous résument les principales caractéristiques de la CPU S7-314 IFM

Mémoires	
Mémoire de travail intégrée uniquement	32ko
Mémoire de chargement intégrée	48Ko de RAM 48Ko de FEPRM
Impossibilité d'extension pour la mémoire de travail ainsi que la mémoire de chargement	
Mémentos	
Nombre	2048 bits
Rémanence : réglable par défaut	de MB 0 à MB 143 de MB 0 à MB 15
Mémentos de cadence	Un octet de memento
Blocs de données	
Nombre	Maximum 127 (DB 0 réservé)
Taille	Maximum 8 Ko
Rémanence : réglable par défaut	Maximum 2 DB, 144 octets de données. Pas de rémanence

Blocs	
Blocs d'organisation (OB)	13
Taille	Maximum 8 Ko
Profondeur d'imbrication :	
Par classe de priorité	8
Supplémentaire à l'intérieur d'un OB d'erreur	4
Blocs fonctionnels (FB)	128
Taille	Maximum 8 Ko
Fonctions (FC)	128
Taille	Maximum 8Ko
Temporisations /compteurs	
Compteurs S7	64
Rémanence par défaut	Z0 à Z7
Rémanence réglable	Z0 à Z63
Plage de comptage	0 à 999
Temporisations S7	128
Rémanence par défaut	Aucune temporisation rémanente
Rémanence réglable	T0 à T7
Plage de temps	10 ms à 9990 s

Zones d'adresses (entrées/sorties)	
Numériques	0.0 à 123.7/0.0 à 123.7
Numériques intégrées	124.0 à 125.7/124.0 à 125.7
Spéciales	126.0 à 126.3/124.0 et 124.1
Analogiques	256 à 751/256 à 751
Analogiques intégrées	128 à 135/128 à 129
Mémoire image (non réglable)	128 octets/128 octets
Sauvegarde	
Avec pile	Toutes les données
Sans pile	144 octets

Fonction de test et de diagnostic	
Etat/forçage de variables	Oui
Variable	Entrées, sorties, DB, temporisations, compteurs, mémentos.
Nombre Etat de variables Forçage de variables	Maximum 30 Maximum 14
Forçage permanent Variables Nombre	Oui Entrées, sorties Max 10
Nombre de points d'arrêt	2
Tampon de diagnostic Nombre d'entrées (non réglables)	Oui 100

Interface de communication MPI	
Vitesse de transmission	19.2 , 187.5 k bauds

Tensions, Courants	
Tension d'alimentation	24V cc
Plage admissible	20,4 à 28,8 V
Consommation (en marche à vide)	Typique 1.0 A

Fonctions intégrées	
Compteur	1 ou 2 selon la configuration utilisateur
Fréquencemètre	Maximum 10 KHz
Positionnement	1 voie

Tableau(III.2): Tableaux techniques

III.5.3.2. LES FONCTION INTEGREES DU S7-314 IFM :

les automates programmables S7-314 IFM, ont un module intégré qui comporte plusieurs fonctions intéressantes. Ces fonctions permettent de contrôler des systèmes à des coûts très appréciables. Les autres solutions possibles pour gérer une tâche d'automatisation sont :

- Programmer des blocs pour la tâche.
- Mettre en place des modules spécialisés.

Le tableau suivant montre les 3 possibilités citées ainsi que les différents critères de sélection :

Critère de sélection	Programme utilisateur	Fonctions intégrées	Modules de fonction
Liaison directe avec les E/S	Non	Oui	Oui
Impact sur le temps de cycle	Oui	Minime	Non
Taux de couverture des applications	Faible	Moyen (50%)	Elevé (95%)
performance au niveau du temps de réaction	Faible	Moyen	Elevé
Traitement des défauts de processus	Non	Limité	Oui

Tableau(III.3): Critères de sélections pour la résolution d'un problème d'automatisation à base d'automates programmables.

Les fonctions intégrées à la CPU S7-314 IFM sont :

- Fonction fréquencemètre.
- Fonction compteur (1 compteur pour comptage/décomptage)
- Fonction compteur A/B (2 compteurs A et B pour comptage/décomptage).
- Fonction positionnement.

les modules spécialisés offrent un ajout sur temps de cycle nul, mais il faut noter que le prix d'une solution avec des modules spécialisés est nettement supérieur à celui d'une solution avec CPU dotée de fonctions intégrées. Il est donc préférable d'utiliser une CPU IFM, si les performances d'un module spécialisé ne sont pas nécessaires.

Il existe 4 entrées spéciales sur une CPU S7-314 IFM. Les configurations possibles sont :

- 4 entrées pour les alarmes processus. (Entrées TOR)
- 4 entrées TOR pour la fonction intégrée compteur.
- 4 entrées TOR pour la fonction intégrée compteurs A/B.
- 1 entrée TOR pour la fonction intégrée fréquencemètre et 3 entrées TOR standard.
- 3 entrées TOR pour la fonction intégrée positionnement et 1 entrée TOR standard.

les entrées spéciales qui ne sont pas utilisées pour une fonction intégrée ou pour une alarme de processus peuvent être utilisées comme entrées TOR standard.

III.5.3.3. Activation et paramétrage des fonctions intégrées :

L'activation des fonctions intégrées se fait par le biais de STEP 7.

Pour choisir un mode de fonctionnement, il faut ouvrir HW CONFIG de STEP 7. Après avoir effectué la configuration matérielle, il faut ouvrir les propriétés de la CPU, puis dans l'onglet *fonction intégrée* choisir la fonction à activer.

Un bouton *paramètre* permet une configuration plus approfondie des fonctions intégrées.

III.5.4. Types de connexions possibles avec un moteur :

L'automate programmable S7-314 IFM ne peut pas être raccordé directement à un moteur, il est indispensable d'utiliser une interface de puissance pour la commande en position. Ces interfaces peuvent être de deux types :

- Montage à contacts : permet de commander un moteur à deux vitesses.
- Variateur de fréquence : permet de commander un moteur synchrone ou asynchrone.

III.5.4.1. Commande d'un moteur à deux vitesses :

Pour ce faire, 4 sorties TOR sont utilisées :

Sortie	Adresse	Rôle
Sortie TOR petite vitesse	A124.0	Sorties pour le choix de la vitesse du moteur
Sortie TOR grande vitesse	A124.1	
Sortie TOR marche avant	A124.2	Sorties pour le choix du sens de rotation
Sortie TOR marche arrière	A124.3	

Tableau(III.4): Sorties associées à la commande d'un moteur à deux vitesses.

La figure suivante montre le profil de vitesse que l'on peut avoir pour un moteur à deux vitesses

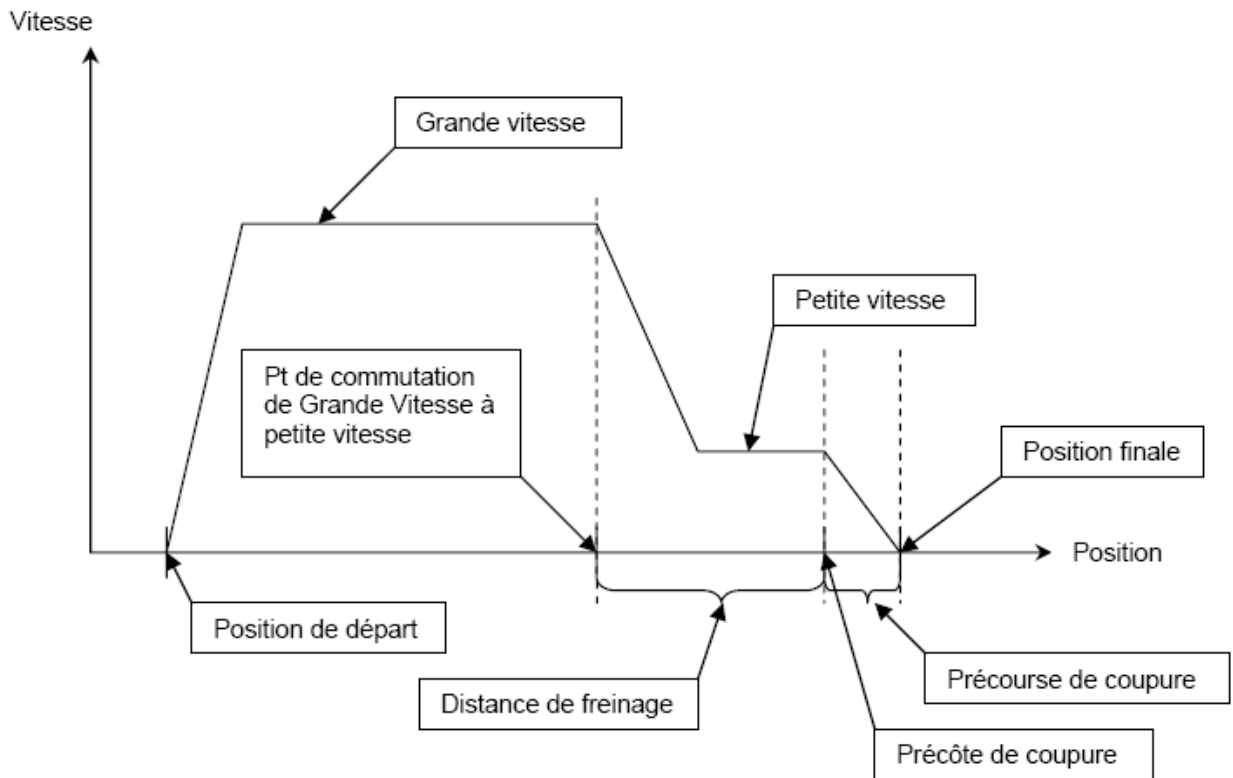


Fig.(III.9): Profil de vitesse d'un moteur à deux vitesses.

Le principe est de faire démarrer le moteur avec la grande vitesse, puis lors de la phase de freinage, on commute sur la petite vitesse. Juste avant d'arriver à la position finale, on coupe la petite vitesse et on laisse le moteur s'arrêter jusqu'à atteindre la position désirée.

La figure suivante montre le comportement des sorties TOR lors de la commande :

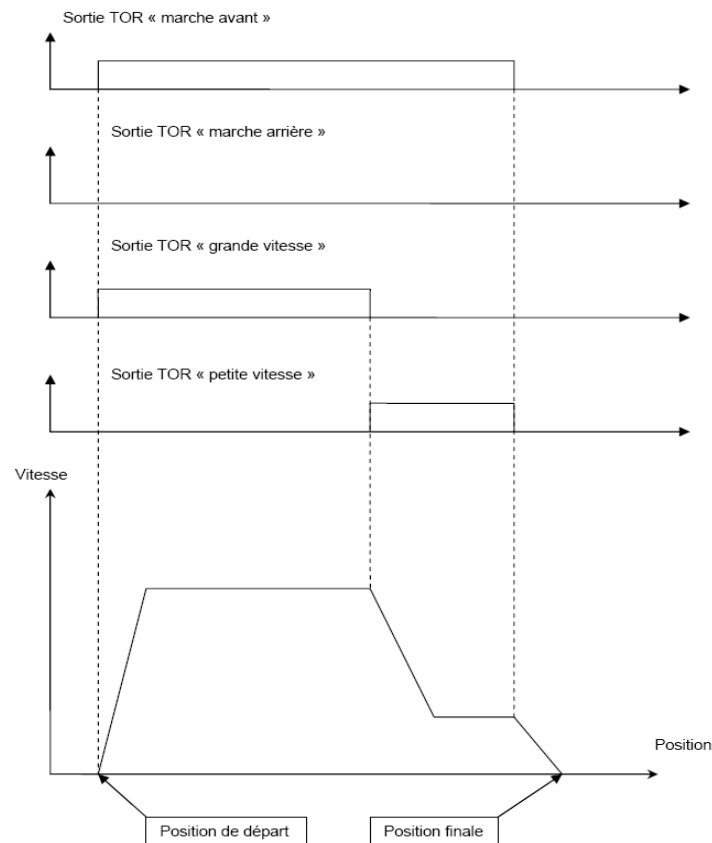


Fig.(III.10): Chronogramme des sorties TOR pour la fonction intégrée positionnement.
(Positionnement en marche avant)

III.5.4.2. Commande par convertisseur de fréquence :

Cette méthode est utilisée pour actionner des moteurs synchrones ou asynchrones.

Les sorties utilisées sont :

Sortie	Adresse	Rôle
Sortie TOR marche arrière	A124.2	Utilisé si le convertisseur ne gère que les signaux analogiques positifs.
Sortie TOR marche avant	A124.3	
Sortie analogique vitesse	PAW 128	Donne la valeur de la vitesse : - f Délivre une sortie de 0-20mA ou de 0-10V dans le cas d'un convertisseur qui ne gère que des signaux analogiques positifs. - f Délivre une sortie de ± 20 mA ou ± 10 V pour un convertisseur qui gère des signaux positifs et négatifs.

Tableau (III.5): Sorties associées à la commande d'un moteur par variateur de vitesse.

La commande par convertisseur de fréquence a comme contraintes principales :

- La vitesse maximale admise par le système.
- L'accélération maximale admise par le système.

La commande donnée fait en sorte que ces valeurs ne soient jamais dépassées tout en optimisant le temps.

La figure qui suit montre les profils de vitesse et d'accélération utilisés par la fonction intégrée positionnement de la CPU S7-314 IFM:

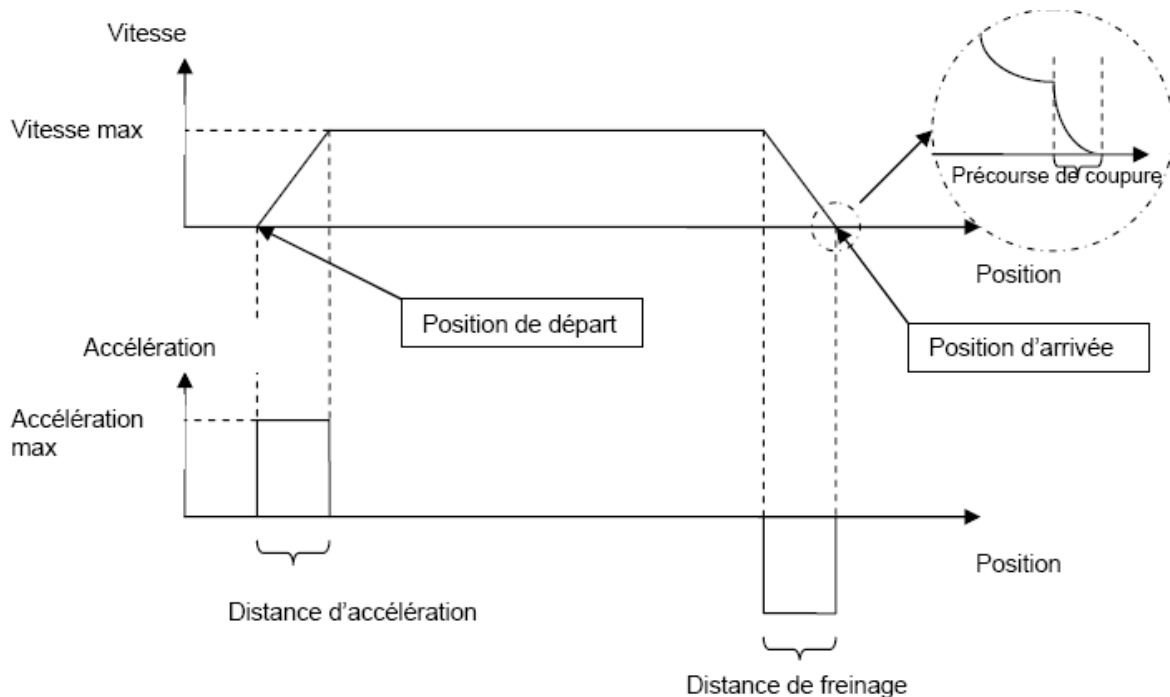


Fig.(III.11): Profils de vitesses et d'accélération pour la fonction positionnement.

La vitesse maximale est définie dans le programme utilisateur.

La distance d'accélération et la distance de freinage sont à paramétrer avec *STEP 7*.

On remarque qu'il existe une précouse de coupure pour l'arrêt du moteur, ce paramètre est donné dans le programme utilisateur.

Valeurs sur les sorties :

Comme indiqué précédemment, deux configurations sont possibles pour la commande :

- 1 sortie analogique (pour la valeur de la vitesse) qui délivre 0-20mA ou 0-10V Avec deux sorties TOR (1 pour marche arrière et 1 pour marche avant).
- 1 sorties analogique (pour la valeur de la vitesse et le sens) qui délivre $\pm 20\text{mA}$ ou $\pm 10\text{V}$.

Il faut savoir que la sortie analogique est donnée sous forme d'échelons. Ces échelons sont au nombre de 10 et ont la même largeur, mais possèdent des hauteurs différentes.

La hauteur des échelons est prédéfinie par la fonction intégrée positionnement.

La largeur est calculée de la manière suivante :

$$\text{Largeur de l'échelon} = \frac{\text{Distance d'accélération/freinage}}{10}$$

Cette valeur est tronquée de sa partie réelle pour être sûr que la distance d'accélération/de freinage parcourue effectivement ne soit jamais supérieure à la distance paramétrée.

La valeur analogique délivrée par la sortie se calcule de la manière suivante :

$$\text{ou } V_{\text{Max}} = \frac{10V}{256} \times (256 - \text{BREAK}) \qquad I_{\text{Max}} = \frac{20mA}{256} \times (256 - \text{BREAK})$$

La valeur de BREAK est donnée dans le SFB 39. [22]

III.6 CONCLUSION:

Nous avons vu, dans le présent chapitre, une description détaillée de l'API S7-300 (S7-314 IFM) produit par la firme SIMENS. Nous avons aussi décrit la plupart des fonctions qui y sont intégrées et qui, comme nous l'avons vu, jouent un rôle primordial dans le fonctionnement de l'API.

CHAPITRE IV

Logiciel STEP 7

IV.1. Description du STEP 7:

STEP7 fait partie de l'industrie logicielle SIMATIC. Il représente le logiciel de base pour la configuration et la programmation de systèmes d'automatisation.

Le progiciel de base STEP 7 existe en plusieurs version :

- STEP 7 Micro/DOS et STEP 7 Micro Win pour des applications autonomes simple SIMATICS7-200
- STEP 7 Mini pour des applications autonomes simple sur SIMATIC S7-300 et SIMATIC C7- 620.
- STEP 7 pour des applications sur SIMATIC s7-300/400 ;SIMATIC M7-300/400et SIMATIC C7

présentant des fonctionnalités supplémentaires :

- 1- possibilité d'extension grâce aux application proposées par l'industrie logiciel SIMATIC.
- 2- possibilité de paramétrage de modules fonctionnels et de modules de communication
- 3- Forçage et fonctionnement multiprocesseur .
- 4- Communication par donnés globales.
- 5- Transfert de données commande par événement à l'aide de blocs fonctionnels.
- 6- configuration de liaisons.

Les tâches de bases qu'il offre à son utilisateur lors de la création d'une solution d'automatisation Sont.

- La création et gestion de projets.
- La configuration et le paramétrage du matériel et de la communication.
- La gestion des mnémoniques.
- La création des programmes.
- Le chargement de programmes dans les systèmes cibles.
- Le test de l'installation d'automatisation.
- Le diagnostic lors des perturbations dans l'installation.

Il s'exécute sous les systèmes d'exploitation de MICROSOFT à partir de la version Windows 95. Et s'adapte par conséquent à l'organisation graphique orientée objet qu'offre ces systèmes d'exploitation. [20,21]

IV.2. Applications du logiciel de base STEP7 :

Le logiciel STEP7 met à disposition les applications de base suivantes :

- Le gestionnaire de projets.
- La configuration du matériel.
- L'éditeur de mnémoniques.
- L'éditeur de programmes CONT, LOG, LIST.
- Le S7GRAPH pour la programmation séquentielle.
- La configuration de la communication NETPRO.
- Le diagnostic du matériel.

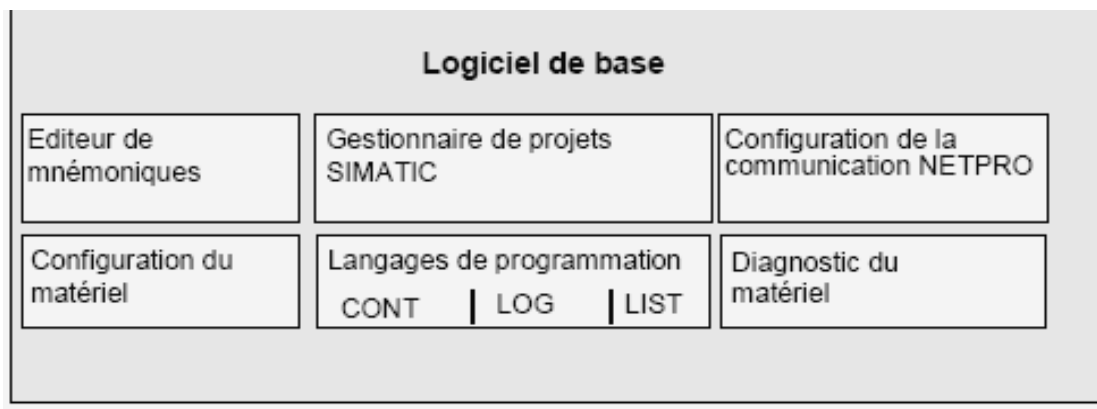


Fig.(IV.1): Applications du logiciel de base STEP7.

IV.2.1. Gestionnaire de projets SIMATIC Manager :

Le gestionnaire de projets SIMATIC Manager gère toutes les données relatives à un projet d'automatisation, il démarre automatiquement les applications requises pour le traitement des données sélectionnées.

Le schéma suivant représente la fenêtre qui apparaît à l'ouverture du SIMATIC Manager

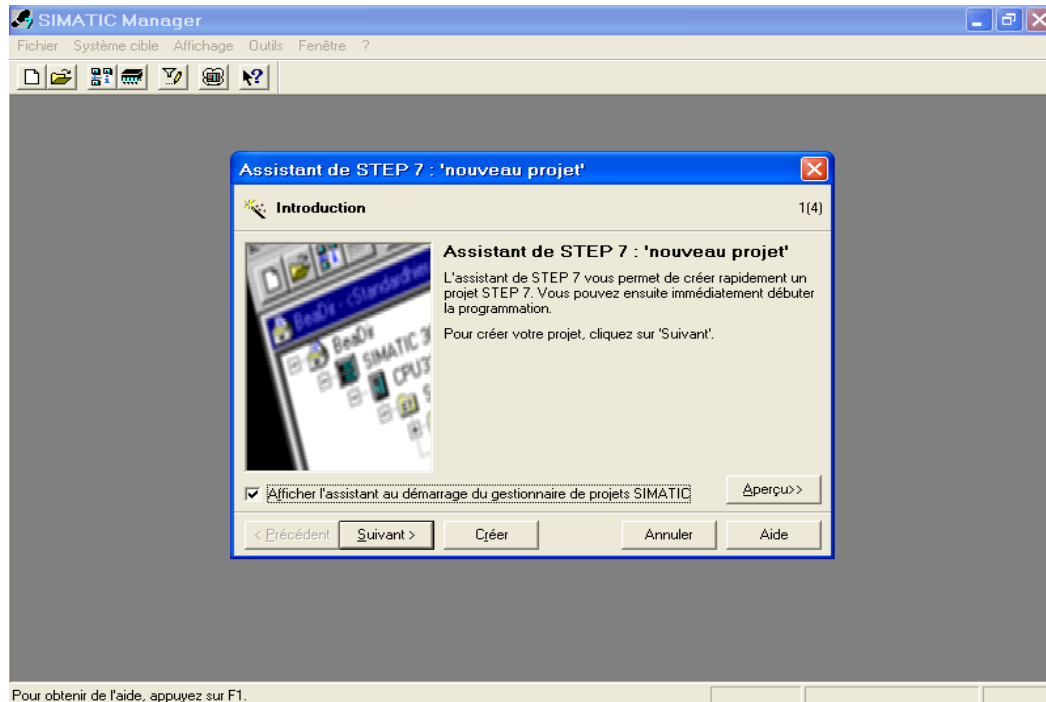


Fig.(IV.2) : Fenêtre du SIMATIC manager.

IV.2.2. Configuration du matériel HW Config :

HW Config est utilisé pour configurer et paramétrer le support matériel dans un projet d'automatisation.

IV.2.3. Editeur de mnémoniques :

Il permet la gestion de toutes les variables globales. En effet, il définit des désignations symboliques et des commentaires pour les signaux du processus (Entrées/Sorties), les mémentos, les blocs de données, les temporisations et les compteurs.

La table des mnémoniques qui en résulte est mise à disposition de toutes les applications. La modification de l'un des paramètres d'un mnémonique est de ce fait reconnue automatiquement par toutes les applications.

IV.2.4. Editeur de programme :

Les langages de base proposés sont :

- Le schéma à contact (CONT), langage graphique similaire aux schémas de circuits à relais, il permet de suivre facilement le trajet du courant.

- La liste d'instruction (LIST), langage textuel de bas niveau, à une instruction par ligne, similaire au langage assembleur.
- Le logigramme (LOG), langage de programmation graphique qui utilise les boîtes de l'algèbre de BOOLE pour représenter les opérations logiques.
L'éditeur de programmes permet aussi la visualisation et forçage de variables,

IV.2.5. Le S7Graph :

Le S7Graph est conçu pour la programmation graphique des processus, en plus des possibilités que nous offre le STEP7, il nous permet une programmation claire et rapide.

IV.2.6. Configuration de la communication Net Pro :

La configuration et le paramétrage de réseaux se font à l'aide de l'application NetPro. Elle permet de :

- Créer une vue graphique du réseau en question ainsi que les sous-réseaux qui le constituent.
- Déterminer les propriétés et les paramètres de chaque sous-réseau.

IV.2.7. Diagnostic du matériel :

Le diagnostic du matériel fournit un aperçu de l'état du système d'automatisation. Dans une représentation d'ensemble, un symbole permet de préciser pour chaque module, s'il est défaillant ou pas. Un double clic sur le module défaillant permet d'afficher des informations détaillées sur le défaut. Avec le diagnostic, on peut avoir des informations générales sur les modules, les causes des erreurs, comme on peut détecter les causes des défaillances dans un programme. [17,20,21]

IV.3. Possibilité d'extension du logiciel de base STEP7 :

Cette extension est réalisée à l'aide de logiciels optionnels, regroupés dans trois catégories

IV.3.1. Applications techniques :

Ensemble de langages de programmation évolués pour programmeur, des langages graphiques pour l'ingénieur en technologie et enfin des logiciels complémentaires pour le diagnostic, la simulation, la maintenance à distance, et la documentation de l'installation.

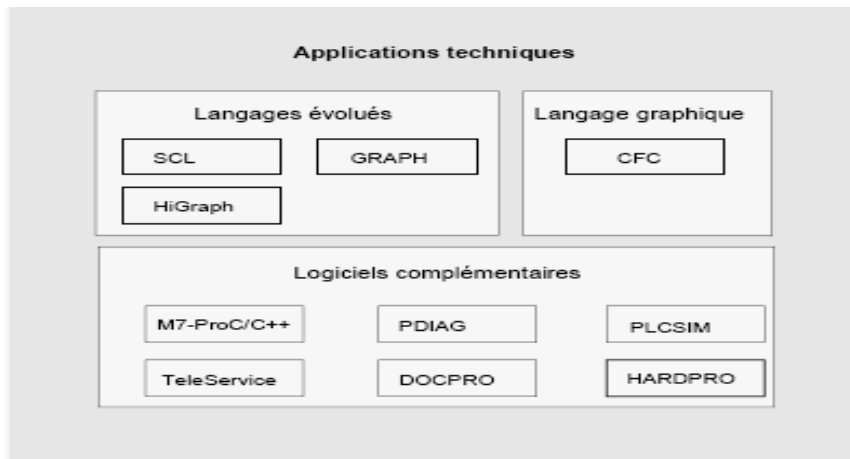


Fig.(IV.3): Applications techniques.

IV.3.2. Logiciels exécutables :

Solutions finies programmées pouvant être appelées dans le programme utilisateur, et d'être directement intégrées dans la solution d'automatisation.

Exemple :

- Le contrôle PID qui permet l'intégration de régulateurs à action continue et de régulateurs à impulsion dans le programme utilisateur. L'application de paramétrage à laquelle la définition du régulateur est intégrée, permet le paramétrage rapide et le réglage optimal du régulateur.
- Le contrôle avec logique floue. Utilisé lorsque des processus ne peuvent pas ou peuvent difficilement être décrits mathématiquement, lorsque le déroulement de mécanismes et de processus est imprévisible, lorsque des comportements non linéaires surviennent alors que l'on dispose d'une connaissance acquise par expérience du processus.

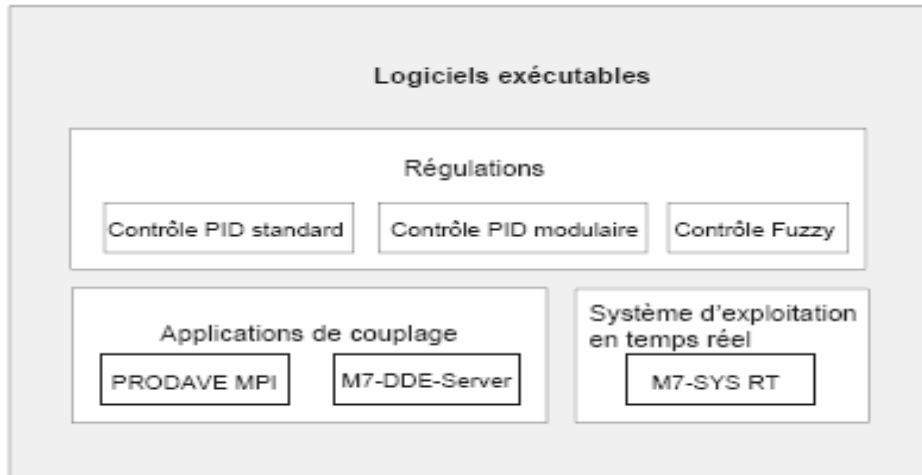


Fig.(IV.4): Logiciels exécutables .

IV.3.3. Interfaces homme/machine :

Ce sont des logiciels spécifiques au contrôle-commande dans SIMATIC. On peut citer :

- Le système de visualisation du processus SIMATIC WinCC qui est un système de base indépendant des branches et technologies d'utilisation et qui comporte toutes les fonctions indispensables au contrôle-commande.
- ProAgent qui permet un diagnostic précis et rapide du processus dans les installations et les machines en fournissant des informations relatives à la localisation et à la cause des erreurs. [21]

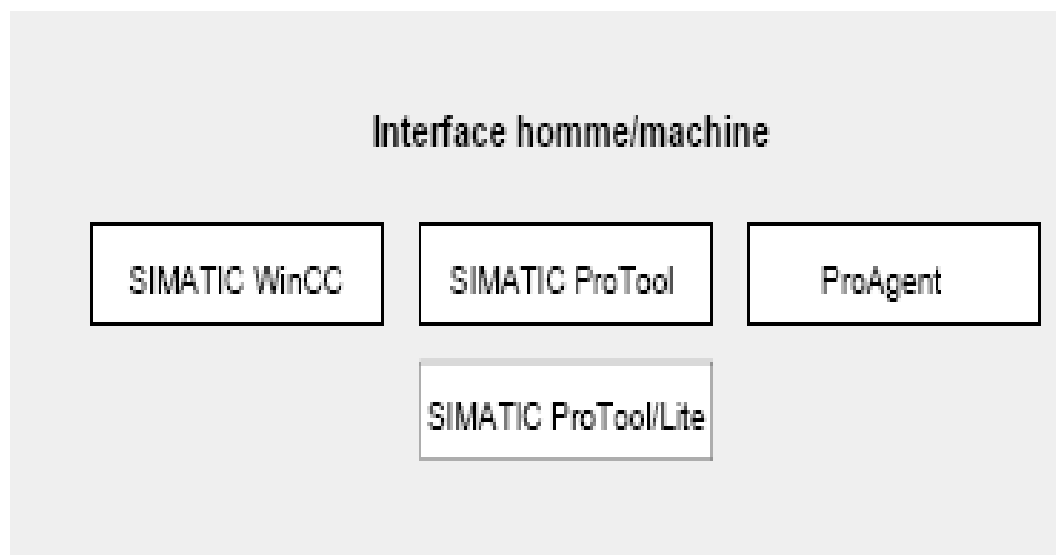


Fig. (IV.5): Interfaces homme/machine

IV.4. PREMIER PAS VERS STEP7 :

Pour concevoir un projet avec STEP7, il existe 2 approches :

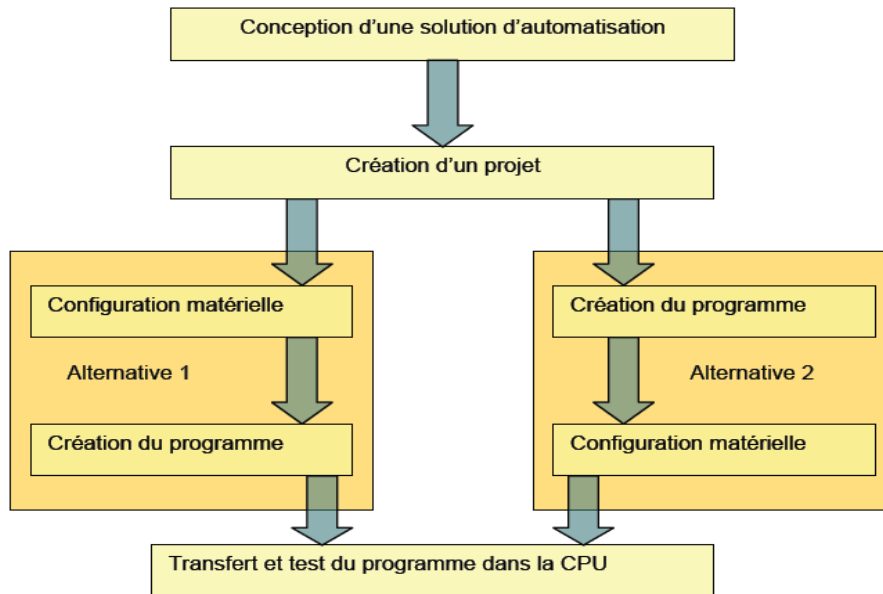


Fig.(IV.6): Organigramme pour la création de projets sous STEP 7.

Il faut noter que pour un système contenant beaucoup de variables, la seconde alternative n'est pas très pratique.

Nous optons donc pour la première approche. [20]

IV.5. Création du projet avec STEP7 :

Pour créer un projet avec STEP 7, on peut lancer l'assistant de création de projet de STEP7, ou créer directement un projet que l'on configurera soi-même.

IV.5.1 Utilisation de l'assistant de création de projets :

Par défaut l'assistant de création de projets apparaît à chaque démarrage du SIMATIC Manager, si ce n'est pas le cas, son lancement se fait en passant par le menu Fichier>Assistante Nouveau projet'.

Cet assistant permet de créer avec une interface simple.

Les étapes à suivre sont les suivantes :

Etape1 :

Cliquer sur le bouton « suivant »

Etape 2 :

- Il faut choisir la CPU utilisée pour le projet, la liste contient normalement toutes les CPU supportées par la version de STEP7 utilisée.
- Dans le champ « Nom de CPU », il faut donner un nom à la CPU. Cela peut
- S'avérer utile dans le cas où l'on utilise plusieurs CPU dans un même projet.
- Il faut aussi choisir une adresse MPI pour la CPU. Si on utilise une seule CPU
- La valeur par défaut est 2.
- Cliquer sur suivant.

❖ Remarque :

Lors de la sélection de la CPU une brève description est disponible dans la petite fenêtre à coté du choix de l'adresse MPI.

Etape 3 :

Cet écran permet d'insérer des blocs. Ces blocs seront décrits plus loin.

- Pour commencer on se contentera de OB1 seulement qui est le bloc principal (Équivalent du MAIN pour le langage C).
- On doit aussi choisir un langage de programmation parmi les trois proposés (LISTE, CONT ou LOG).
- Cliquer sur suivant.

Etape 4 :

Nommer le projet et cliquer sur créer.

Le projet est maintenant créé, on peut voir arborescence à gauche de la fenêtre Qui s'est ouverte.

IV.5.2. Création d'un nouveau projet sans l'assistant de création de projet :

Cette méthode est un peu plus compliquée, mais permet de mieux gérer le projet. Dans la fenêtre SIMATIC Manager, cliquer sur Fichier>Nouveau (ou encore CTRL+N), une fenêtre demandant un nom de projet s'ouvre. Il faut donc donner un nom au projet puis valider par OK. La fenêtre du projet s'ouvre.

Le projet est vide, il faut lui insérer une station SIMATIC, cela est possible en cliquant sur le projet avec le bouton droit puis Insérer un nouvel objet >station SIMATIC 300.

La station SIMATIC n'est toujours pas configurée, il faut passer à l'étape de configuration matérielle.

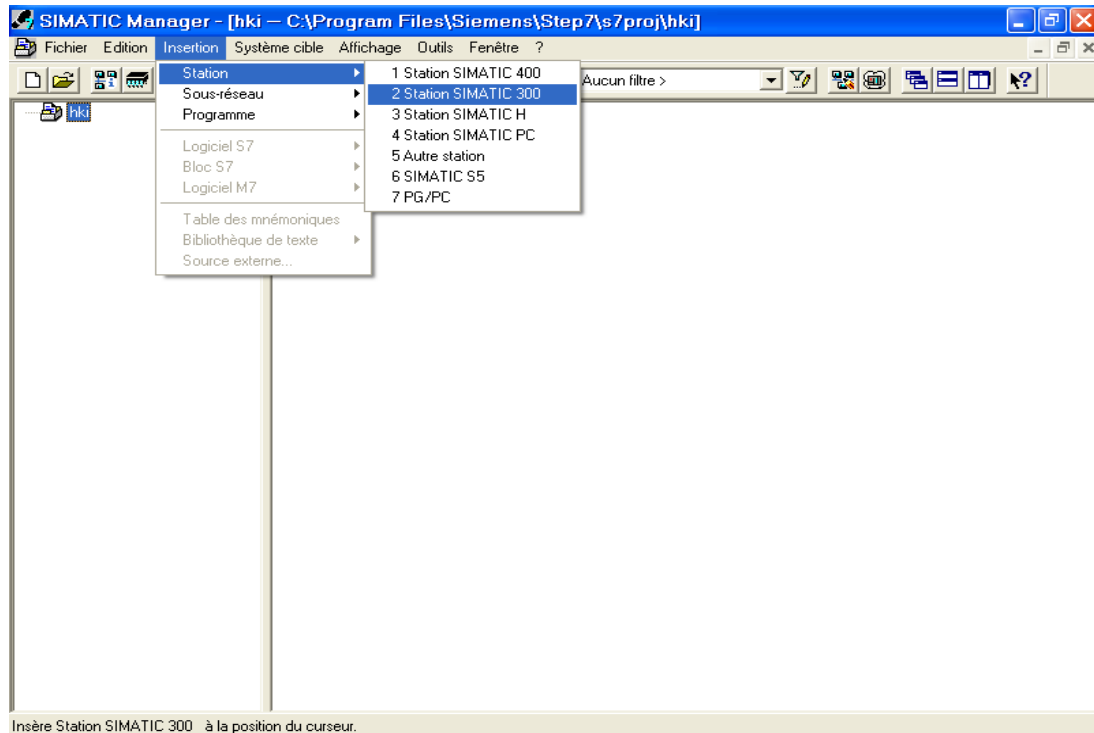


Fig.(IV.7): Création d'un projet sans l'assistant.

IV.5.2. Configuration matérielle :

La configuration matérielle est une étape très importante, elle permet de reproduire à l'identique le système utilisé (alimentation, CPU, modules etc....).

Pour effectuer cette configuration, il faut aller sur l'icône Station SIMATIC300, sur la fenêtre de droite s'affichent deux icônes : « Matériel » et le nom de la CPU. il faut ouvrir l'icône matériel : la fenêtre HW Config s'ouvre.

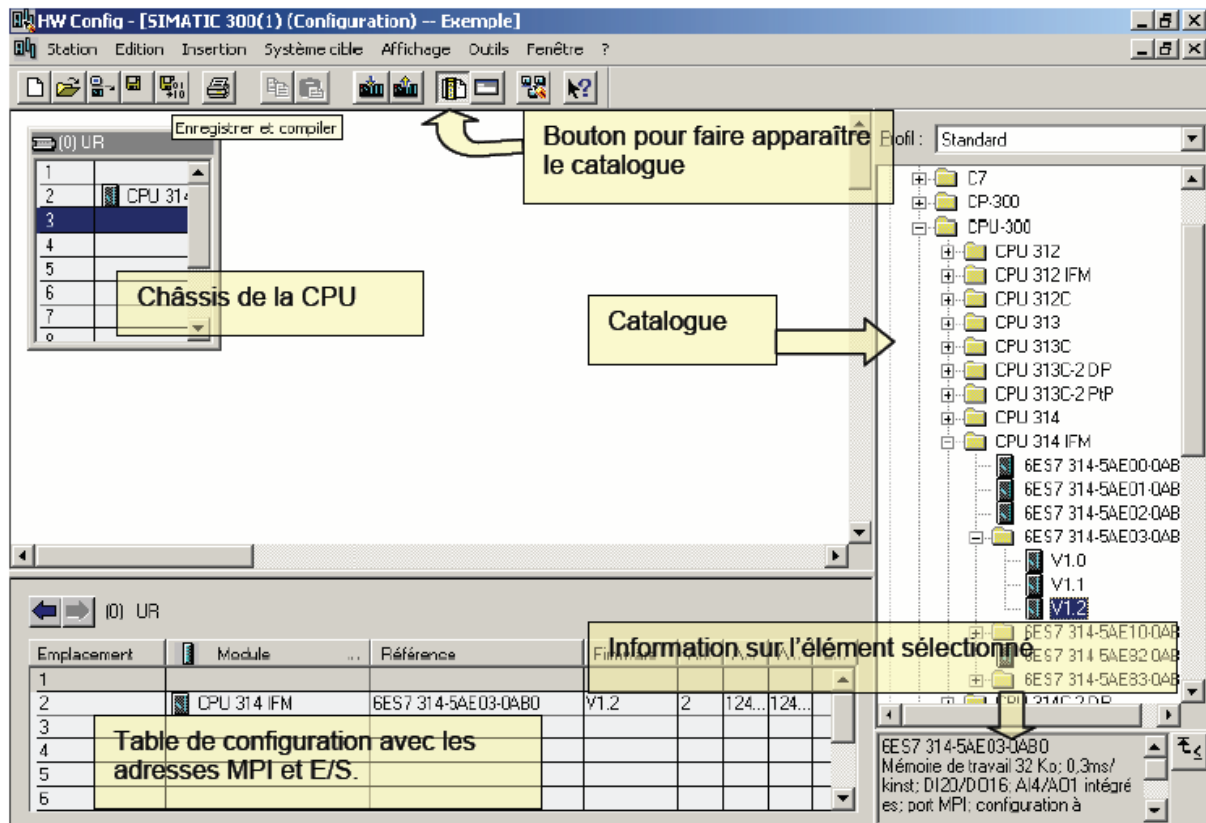


Fig.(IV.8): Configuration du matériel.

Nous avons tout d'abord besoin d'un châssis ou RACK puis d'une alimentation. Dans le catalogue ouvrir le dossier PS-300 qui se trouve dans SIMATIC-300 puis choisir un modèle PS 307 2A.

Pour l'installer sur le châssis, sélectionner l'emplacement 1 à l'aide de la souris, puis double-cliquer sur l'alimentation.

On peut de la même manière insérer des composants sur le châssis en fonction de la configuration réelle. Dans notre cas, l'insertion de la CPU S7 314 IFM de référence 6ES7314-5EA03-0AB0 se fera dans l'emplacement 2.

Le paramétrage de la CPU, à l'aide de menu, permet de définir des caractéristiques, telles que : le comportement à la mise en route, la surveillance du temps de cycle ainsi que l'activation et la désactivation des plages de rémanences et les fonctions intégrées, ces données sont enregistrées dans les blocs de données systèmes.

Le paramétrage des modules est réalisé automatiquement au démarrage de la CPU. Ainsi, le remplacement d'un module est ainsi possible sans nouveau paramétrage.

A la fin de la configuration, il suffit de cliquer sur Station>Enregistrer et compiler pour valider les changements apportés au châssis. De cette manière les changements seront pris en compte dans le reste du projet.

IV.5.3. Définition des mnémoniques :

Il faut définir les variables qui vont être utilisées lors des étapes de programmation. L'utilisation de noms communs est plus aisée que la manipulation de chiffres (ex : utiliser « moteur » au lieu du bit de sortie A0.0).

Pour accéder à la table des mnémoniques : cliquer sur le dossier programme dans la fenêtre du projet, puis sur l'icône mnémoniques.

L'utilisation de cette table consiste à :

- Donner un nom à la mnémonique dans la première colonne.
- Donner la variable associée à cette mnémonique dans la seconde colonne.
- Le type de la donnée est automatiquement généré par STEP 7.
- Ecrire éventuellement un commentaire dans la colonne prévu à cet effet.

Après avoir défini toutes les mnémoniques, il suffit d'enregistrer pour que les changements soient pris en compte dans le reste du projet.

Etat	Mnémonique	Opérande	Type de don	Commentaire
1	A	E 124.1	BOOL	Détecteur A
2	B	E 124.2	BOOL	Détecteur B
3	C	E 124.3	BOOL	Détecteur C
4	COMPLETE RESTART	OB 100	OB 100	Complete Restart
5	D	E 124.4	BOOL	Détecteur D
6	FB1OK	A 125.5	BOOL	Sortie signalant le bon fonctionnement du bloc de fonct...
7	M	E 124.0	BOOL	Bouton de mise en marche du système
8	M1	A 125.0	BOOL	Sortie de commande du moteur M1
9	M2	A 125.1	BOOL	Sortie de commande du moteur M2
10	SENS	A 125.4	BOOL	Sortie commandant le sens de rotation des moteurs
11	TEMPO1	A 125.2	BOOL	Sortie indiquant l'état de la première temporisation
12	TEMPO2	A 125.3	BOOL	Sortie indiquant l'état de la seconde temporisation
13				

*Fig.(IV.9):*La fenêtre d'édition de mnémoniques.

Si on a besoin d'insérer de nouveaux objets dans le projet (ex : d'autre blocs de programme) il suffit de cliquer avec le bouton droit de la souris sur le dossier ou l'on veut ajouter l'objet puis insérer nouvel objet, et dans le menu sélectionner l'objet voulu. [17]

IV.6. ORGANISATION D'UN PROGRAMME UTILISATEUR :

Le logiciel de base STEP 7, permet de structurer le programme utilisateur. Cette structuration est réalisée par la subdivision du programme en différentes parties autonomes. Il en résulte les avantages suivants :

- Ecrire des programmes importants et clairs.
- Standardiser certaines parties du programme.
- Simplifier l'organisation du programme.
- Modifier facilement le programme.
- Simplifier le test du programme, car on peut l'exécuter section par section.
- Faciliter la mise en service.

IV.6.1. Hiérarchisation dans un projet :

Chaque programme utilisateur est structuré sous forme de projet. Un projet représente l'ensemble des données et programmes d'une solution d'automatisation et se trouve à la tête d'une hiérarchie d'objets qui sont :

- objet projet.
- objet station.
- objet Modules programmables.
- objet Programme S7/M7.
- objet dossier Sources.
- objet dossier Blocs.

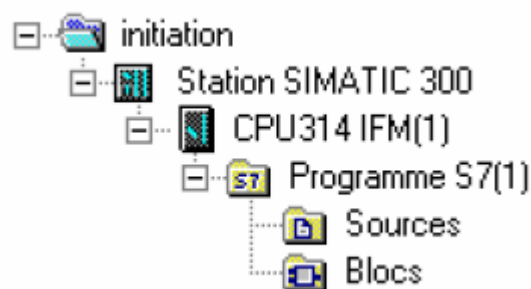


Fig.(IV.10):L'hiérarchie d'un projet STEP 7.

IV.6.1.1. Objet station :

Une station SIMATIC 300/400 représente une configuration matérielle S7. Elle comporte un plusieurs modules programmable.

IV.6.1.2. Objet modules programmables :

Représente les données de paramétrage d'un module comme les CPU et les modules fonctionnels.

IV.6.1.3. Objet programme S7/M7 :

Un programme (S7/M7) est un dossier contenant les logiciels pour les modules programmable. Dans un programme S7/M7 figure déjà :

Une table de mnémoniques, un dossier « Blocs » et un dossier « Sources ».

IV.6.1.4. Objet dossier sources :

Il contient les programmes source sous forme de texte. On peut saisir une partie ou tout le programme sous forme de source que l'on compilera ensuite en blocs.

Les avantages de la création de programme sous forme de source sont :

- La possibilité de programmer plusieurs blocs dans une même source.
- L'avantage d'enregistrer la source malgré la présence éventuelle d'erreurs de syntaxe, ce qui n'est pas possible lors de la création de blocs de code avec vérification de syntaxe. Cela signifie que les erreurs de syntaxe ne seront signalées que lors de la compilation de la source.

Elle peut être créée et traitée avec le choix de l'éditeur ASCII, puis importée et compilée en blocs individuels. La compilation entraîne la génération des différents blocs et leur sauvegarde dans le programme utilisateur S7.

La source doit être écrite en utilisant la syntaxe du langage de programmation « liste d'instructions (LIST) », et nécessite par conséquent l'application de règles précises.

IV.6.1.5. Objet dossier Blocs :

Le dossier Blocs contient les blocs que l'on doit charger dans la CPU pour réaliser la tâche d'automatisation.

Il englobe les blocs de code (OB, FB, SFB, FC, SFC) qui contiennent les programmes qu'on doit charger dans la CPU, et les blocs de données (DB d'instance et DB globaux) qui contiennent les paramètres du programme.

a) Les blocs d'organisation (OB) :

Ils constituent l'interface entre le système d'exploitation et le programme utilisateur. Ils sont appelés par le système d'exploitation et gèrent le traitement de programme cyclique et déclenché par alarme, ainsi que le comportement à la mise en route de l'automate programmable et le traitement des erreurs. On peut programmer les blocs d'organisation et déterminer ainsi le comportement de la CPU.

Le bloc d'organisation **OB1** est généré automatiquement lors de la création d'un projet, il représente le programme principal (MAIN dans le langage C). En effet, c'est le programme appelé cycliquement par le système d'exploitation.

Les autres blocs, existant dans le projet seront exécutés à leur appel par l'OB1.

b) Les blocs fonctionnels (FB) :

C'est un sous programme écrit par l'utilisateur, il facilite la programmation de fonctions complexes souvent utilisées. Il exécute par l'appel d'autre bloc de code.

Un bloc de données d'instance, qui constitue sa mémoire, lui est associé. Ce dernier contient les paramètres transmis au FB ainsi que les variables statiques.

c) Le bloc fonctionnel système (SFB) :

C'est un bloc fonctionnel intégré à la CPU S7. Les SFB font partie du système d'exploitation, par conséquent, ils ne sont pas chargés en tant que partie du programme. Comme les FB, les SFB sont des blocs avec mémoire. On doit donc également créer pour les SFB des blocs de données d'instance que l'on charge dans la CPU en tant que partie du programme.

Ils sont utilisés pour des fonctions spéciales intégrées de la CPU 314 IFM, comme ils peuvent être utilisés pour la communication via des liaisons configurées.

d) Les fonctions (FC) :

Elle contient des routines pour les fonctions fréquemment utilisées, comme le renvoi d'une valeur au bloc appelant. Elle est sans mémoire et contient uniquement des variables

temporaires qui sont sauvegardées dans la pile de données locales et perdues à l'achèvement de cette fonction.

Mais elle peut faire appel à des blocs de données globaux pour la sauvegarde de ses données.

e) La fonction système (SFC) :

C'est une fonction intégrée dans la CPU S7, pré-programmée et testée. Elle est appelée à partir du programme. Comme ces fonctions font partie du système d'exploitation, elles ne sont pas chargées en tant partie du programme. Comme les FC, les SFC constituent des blocs sans mémoire.

Parmi les fonctionnalités qu'elles proposent :

Le contrôle du programme, la gestion des alarmes horaires et temporisées, la mise à jour de la mémoire image du processus, l'adressage de modules et la création de messages relatifs aux blocs.

f) Les blocs de données d'instance (DB d'instance) :

Associé à chaque bloc fonctionnel, il contient les paramètres effectifs et les données statiques du FB.

On peut utiliser plusieurs DB pour un même FB ; par exemple, un FB pour la commande de plusieurs moteurs, les données de chaque moteur sont sauvegardées dans différents DB.

g) Les blocs de données globaux (DB) :

A l'opposé des DB d'instance qui ne sont associés qu'au blocs fonctionnels, les DB globaux servent à l'enregistrement de données utilisateur pouvant être utilisées par tous les autres blocs de code.

❖ Remarque :

A l'appel d'un FB, le DB d'instance lui correspondant est automatiquement généré une fois qu'il est inséré à son emplacement.

Mais si on veut créer un bloc de données à partir du dossier bloc, on procède comme suit :

- On clique avec le bouton droit de la souris, puis sur Insérer un nouvel objet>Bloc de données,

▪ Ou Dans la fenêtre SIMATIC Manager, on clique sur le menu Insertion> BlocS7>bloc de données.

Dans les deux cas, le type du bloc de données sera demandé (DB d'instance ou DB global) [17,19,21]

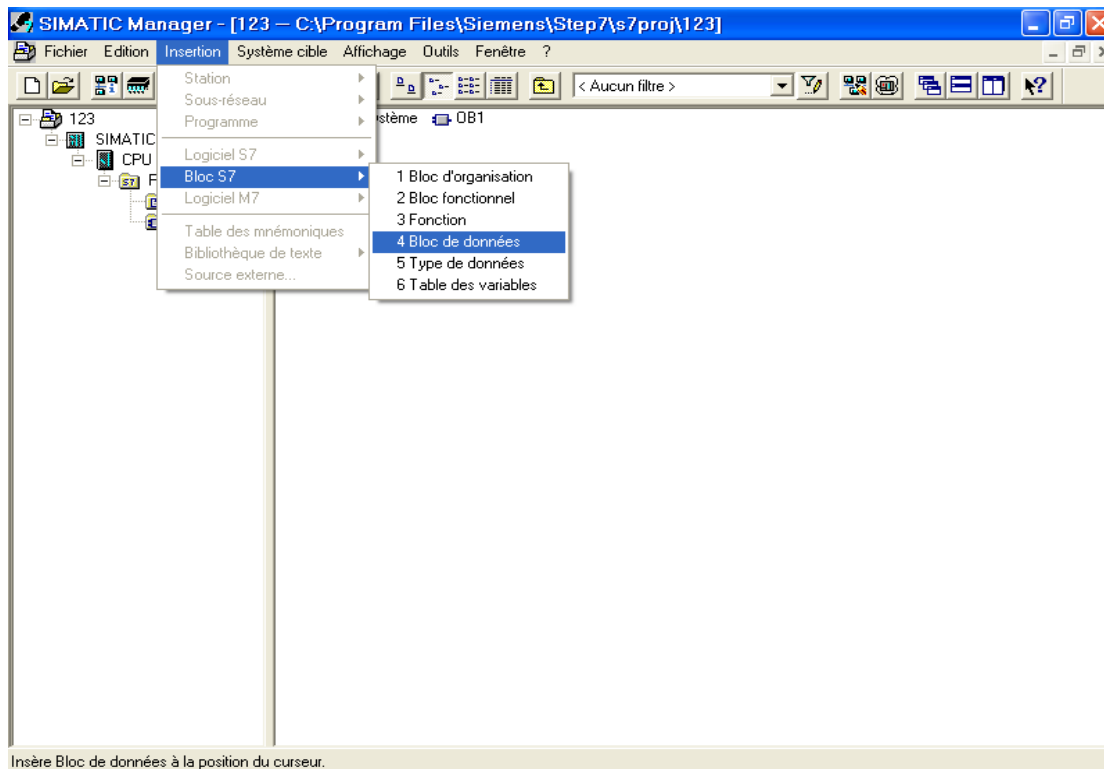


Fig.(IV.11): Création de blocs de données

IV.7. BLOCS D'ORGANISATION :

Ils permettent l'organisation totale du programme utilisateur, en déclenchant l'exécution conditionnelle de certaines parties du programme, ceci se fait :

- A la mise en route de la CPU.
- A des horaires précis ou cycliquement.
- A l'apparition d'erreurs.
- A l'apparition d'alarmes de processus.

IV.7.1. Blocs d'organisation dans la CPU S7-314 IFM :

Les blocs d'organisation qui existe dans la CPU S7-314 IFM sont :

IV.7.1.1. L'OB1 (programme cyclique) :

Comme dans toutes les CPU S7, l'OB1 est exécuté de manière cyclique.

Il peut être interrompu par tous les autres OB car il dispose de la classe de priorité la plus basse. Les événements suivants provoquent son appel par le système d'exploitation :

- La fin du traitement de la mise en route.
- La fin du traitement de son cycle précédent.

IV.7.1.2. L'OB100 (Mise en route) :

Il contient le programme à exécuter lors d'une mise en route.

IV.7.1.3. Les OB d'alarmes :

On appelle alarmes les événements qui déclenchent l'appel d'un OB donné.

IV.7.1.4. Les OB d'erreurs :

- **OB82 (diagnostic des modules) :**

Le diagnostic signale les erreurs survenant sur les modules et permet alors l'exécution de l'OB82. Cet OB contient, dans ses variables locales, l'adresse de base logique du module erroné ainsi que des informations de diagnostic de quatre octets de long.

Si l'OB82 n'a pas été programmé, la CPU passe à l'état arrêt à la survenue d'une erreur.

- **OB80 (Erreur asynchrone) :**

Il est appelé si l'une des erreurs suivantes se produit lors de l'exécution d'un OB :

- Dépassement du temps de cycle.
- Erreur d'acquiescement lors de l'exécution d'un OB.
- Saut de l'heure de déclenchement d'un OB (horloge avancée).

Si l'OB80 n'a pas été programmé, la CPU passe à l'état arrêt à la survenue d'une erreur.

- **OB81 (Erreur d'alimentation) :**

Cet OB est exécuté si un événement provoque une erreur d'alimentation ; par exemple, absence ou rétablissement de la tension de sauvegarde dans la CPU.

Si l'OB81 n'a pas été programmé, la CPU ne passe pas à l'état arrêt à la survenue d'une erreur.

▪ **OB85 (Erreur d'exécution de programme) :**

Cet OB est exécuté si l'un des événements suivants se produit :

- Absence d'OB dans la CPU lors de son appel.
- Erreur lors de l'accès du système d'exploitation à un bloc.
- Erreur d'accès à la périphérie lors de l'actualisation de la mémoire image
Des entrées.
- Erreur d'accès à la périphérie lors du transfert de la mémoire image aux
Modules de sorties.

Si l'OB n'a pas été programmé, la CPU passe à l'état arrêt à la survenue d'une erreur.

▪ **OB87 (Erreur de communication) :**

Le programme de cet OB est exécuté, si une erreur de communication se produit. Par exemple, l'état d'ensemble des données globales ne peut pas être écrit dans le bloc de données.

Si l'OB87 n'a pas été programmé, la CPU passe à l'état arrêt si une erreur survient.

▪ **OB121 (Erreur de programmation) :**

L'apparition d'erreur lors du traitement du programme utilisateur, déclenche l'appel de ce bloc. Cette erreur peut être :

- Une erreur de conversion BCD.
- Un numéro de temporisation erroné.
- Une erreur de numéro de compteur.
- Une erreur d'écriture à l'accès au DB.
- Une erreur de numéro de bloc à l'ouverture d'un DB.
- Une erreur de numéro de bloc à l'appel d'une FC.
- Une erreur de bloc à l'appel d'un FB.
- Un DB non chargé.
- Un FC non chargé.
- Un FB non chargé.

Si l'OB121 n'a pas été programmé, la CPU passe à l'état arrêt à la survenue d'une erreur.

▪ **OB122 (Erreur d'accès à la périphérie) :**

Cet OB se déclenche à l'apparition d'erreur d'accès à la périphérie en lecture ou en écriture.

Si l'OB122 n'a pas été programmé, la CPU passe à l'état arrêt à la survenue d'une erreur.

❖ **Remarque**

Les erreurs résolues par l'appel des deux blocs OB121 et OB122 peuvent être masquées, démasquées ou lues, par les fonctions systèmes suivantes :

- La SFC36 : masque certains codes d'erreurs.
- La SFC37 : démasque les codes d'erreurs qui ont été masquées par la SFC36.
- La SFC38 : lit le registre d'erreur.

IV.7.2. Classes de priorité des blocs d'organisation :

Les blocs d'organisation définissent l'ordre dans lequel les différentes parties du programme sont traitées. L'exécution d'un OB peut être interrompue par l'appel d'un autre OB.

Cette interruption se fait selon la priorité : les OB de priorité plus élevée interrompent les OB de priorité plus faible.

OB1	1
OB10	2
OB20	3
OB35	12
OB40	16
OB80	26
OB81	26
OB82	26
OB85	26
OB100	27
OB121	Selon l'OB responsable de l'interruption
OB122	

Tableau (IV.1) : Classes de priorité des blocs d'organisation

❖ **Remarque :**

STEP7 permet la modification des classes de priorité pour les alarmes horaires, alarmes temporisées, alarmes cycliques de processus. Mais cette propriété n'est pas possible dans les CPU S7 300. [22]

IV.8. DEROULEMENT D'UN PROGRAMME :

IV.8.1. Les programmes existants dans la CPU :

Dans une CPU, s'exécutent deux programmes ; le système d'exploitation et le programme utilisateur.

IV.8.1.1 Le système d'exploitation :

Il regroupe toutes les fonctions et procédures qui ne sont pas liées à la tâche de programmation, ces fonctions et procédures sont :

- La mise en route.
- L'actualisation de la mémoire image des entrées MIE et l'émission de la mémoire image des sorties MIS.
- L'appel du programme utilisateur.
- L'enregistrement des alarmes et l'appel des OB d'alarme.
- La détection et le traitement des erreurs.
- La gestion des zones de mémoires.
- La communication avec les différents partenaires de communication.

La modification des paramètres par défaut du système d'exploitation permet d'influer sur le comportement de la CPU dans des cas précis. Par exemple, le changement des classes de priorités des blocs d'organisation dans les CPU S 400.

IV.8.1.2. Le programme utilisateur :

Il est créé par l'utilisateur puis chargé dans la CPU, il contient toutes les fonctions nécessaires au traitement de la tâche d'automatisation. il doit :

- Déterminer les conditions pour le démarrage à chaud, à froid ou pour le démarrage de la CPU.
- traiter des données du processus (par exemple, combiner des signaux binaires, lire et exploiter des valeurs analogiques, définir des signaux pour la sortie, écrire des valeurs analogiques).

- Réagir aux alarmes.
- Traiter les perturbations dans le déroulement normal du programme.

Le déroulement d'un programme commence par une mise en route qui n'est exécutée qu'une seule fois, suivie de l'exécution cyclique du programme utilisateur. [22]

IV.9. LES LANGAGES DE PROGRAMME : [17]

IV.9.1. Le cont :

C'est un langage dont la logique est inspirée des réseaux électriques, ce qui en fait un langage facile pour les habitués des montages à base de relais.

Le langage à contact, est aussi appelé CONT ou LADDER. C'est un langage entièrement graphique, bien adapté au traitement logique simple de type combinatoire. Il utilise les symboles graphiques comme des contacts à ouverture ou à fermeture et des bobines. Ainsi un programme écrit en langage à contacts ne se présente pas sous la forme d'une liste d'instructions, mais comme une liste de schémas développés classique.


Des blocs d'opérations logiques ou arithmétiques pré-programmés peuvent être insérés dans les réseaux de contacts.

Les opérations combinatoires sur bits utilise deux chiffres : 1 et 0. Pour les contacts et les bobines, 1 signifie activé ou excité et 0 signifie désactivé ou désexcité.

Les opérations de combinaison sur bits évaluent les états de signal 1 et 0. et les combinent selon la logique booléenne. Le résultat de ces combinaisons est égal à 1 ou 0. Il s'agit du résultat logique (RLG).

Les éléments principaux d'un réseau en langage à contacts sont :

Barre d'alim  n (à gauche).

Barre représ  la masse (à droite).

Contact assc  ne variable.

Bobine de s 

IV.9.1.1. Barre d'alimentations et liaisons :

Un diagramme CONT est limité à sa gauche par des barres d'alimentation et à sa droite par une barre représentant la masse.

Les différents composants sont reliés entre eux par des arcs de liaisons horizontaux ou verticaux. Chaque segment de liaison peut prendre les états TRUE ou FALSE (1 ou 0).

L'état d'une liaison est propagé à sa droite.

Toute liaison horizontale connectée à une barre d'alimentation à gauche est active.

On peut avoir des liaisons multiples en combinant plusieurs arcs de liaisons horizontaux à un arc de liaison vertical.

Avec ces quelques règles on peut déjà écrire un réseau basic :

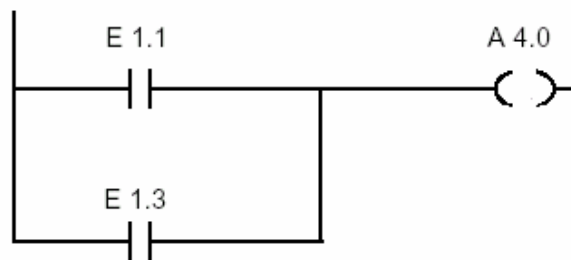




Fig.(IV.12): « OU » logique avec CONT.

A4.0 n'est mise à 1 que si : E1.1 **ou** E1.3 sont à 1

IV.9.1.2. Les contacts :

Les principaux contacts qui peuvent être utilisés dans un diagramme sont :

- contact à fermeture 
- contact à ouverture 

❖ Remarques :

- Le nom de la variable associée est inscrit au-dessus du symbole graphique.
- Les contacts ne représentent pas ce qui est relié physiquement à l'automate, mais sont une interrogation à 1 ou à 0 de l'opérande associé.
- Un contact représente une liaison entre l'état d'un arc et la variable booléenne associée à ce contact.

En effet l'état de la liaison à droite du contact n'est que le résultat du ET logique entre l'état de l'arc de gauche et :

- L'état de la variable associée au contact pour un contact à fermeture.
- L'inverse de l'état de la variable associée au contact pour un contact à ouverture.

IV.9.1.3. Les bobines :

Les bobines standard sont une affectation du résultat logique à la variable associée.





Fig.(IV.13): ET logique avec CONT.

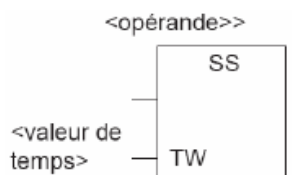
A4.1 n'est mise à 1 que si : E0.0 **et non** (E0.1) sont à 1 simultanément.

IV.9.2. Le LOG :

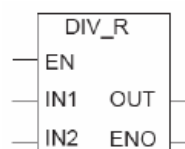
Le langage LOG est un langage graphique qui utilise des boites logiques d'algèbre de BOOL. La base de ce langage est la logique binaire, mais on peut aussi faire des opérations plus complexes telles que les opérations mathématiques à l'aide de blocs.

Les instructions LOG peuvent être sous quatre formes :

- sous forme d'éléments : comme une inversion logique. 
- sous forme de boite avec opérande : comme une affectation 
- sous forme de boite avec opérande et valeur : comme une temporisation.



- sous forme de boite avec paramètre : comme diviser un nombre réel.



❖ Remarque :

- Pour l'activation d'une boite, il faut que EN soit à l'état haut.
- Si la boite s'exécute sans erreur alors ENO est mise à 1, dans le cas contraire ENO est à 0.

Adressage :

Il n'y a que deux types d'adressage possibles avec LOG :

- Adressage immédiat : donne une constante comme opérande.
- Adressage direct : donne une adresse comme opérande.

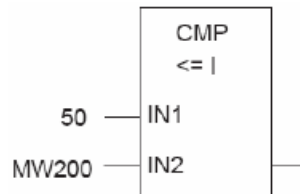
Exemple :

Fig.(IV.14): Adressage immédiat et direct avec LOG.

50 est la valeur effective du premier paramètre, c'est un adressage immédiat.

MW200 est la zone mémoire où est stockée la valeur du second paramètre : c'est un adressage direct.

IV.9.3. Le LIST :**IV.9.3.1. Présentation :**

Le langage LIST figure parmi les langages de base du logiciel STEP7, sa syntaxe est similaire à celle de l'assembleur.

C'est le langage le plus proche du langage machine MC7, des CPU S7, ce qui lui donne l'avantage d'être le langage le plus adapté pour la programmation avec optimisation d'espace mémoire et de temps d'exécution.

Il dispose d'un jeu d'instruction très important permettant la création de programmes utilisateur complets.

Tout programme écrit en CONT ou en LOG peut être réécrit en LIST.

IV.9.3.2. Structure et éléments de LIST :

Dans le langage LIST, l'éditeur de programme divise l'espace de programmation en deux parties ; partie programme et partie commentaire, ces deux parties sont séparées par *II*.

IV.9.3.3. Structure d'une instruction :

Une instruction LIST est constituée soit :

- En une opération seulement,
- En une opération + opérande.

L'opérande peut indiquer :

- une constante.
- un bit du mot d'état.
- une mnémonique.
- un bloc de données.
- un identificateur d'opérande d'une zone mémoire précise + une adresse.

Exemple :

Type d'opérande		Exemple	Explication
Constante	Entier	L 25	Charger l'entier 25
	Temporisation	L S5T#12s	Charger 12 secondes comme valeur initiale de la temporisation dans l'accumulateur 1
Adresse dans le mot d'état	Bit dans le mot d'état	U BIE	Interroger l'état du bit RB du mot d'état
Mnémonique	Nom symbolique	L vitesse	Charger l'octet, le mot ou le double mot dont le mnémonique est vitesse
Bloc de Données	Donnée du bloc de données.	U DB4.DBX5.1	Interroger l'état du bit 5.1 du bloc de donnée 4

Tableau(IV.2): Quelques types d'opérandes.

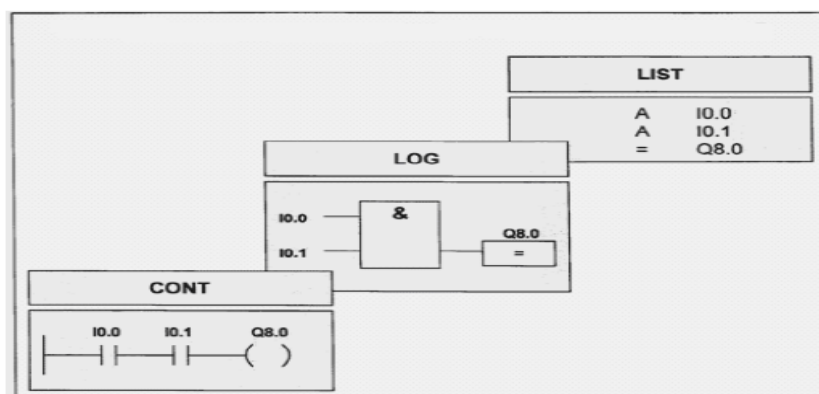


Fig.(IV.15): mode de représentation des langages basiques de programmation STEP 7

IV.9.4. S7-PLCSIM: [18]

IV.9.4.1. Test du logiciel sans CPU:

Les systèmes de simulation peuvent offrir une aide efficace au développement de programmes et, par la suite, à leur utilisation en production. En automatisation, la simulation de l'environnement de test, automate et processus compris, écourte par exemple les temps de mise en service et réduit ainsi les coûts.

Le détection précoce des erreurs de programmation et l'optimisation préalable des parties de programme concourent à la fourniture sur le site de programmes débogués, prêts à l'emploi. Après modification, le programme pourra être testé avant d'être rechargé dans l'API de l'installation.

IV.9.4.2. Domaine d'application

S7-PLCSIM fournit un environnement d'installation simulé pour le test fonctionnel des blocs et programmes utilisateur destinés au S7-300 et au S7-400 sur la console PG ou le PC que vous soyez ou non connecté à un matériel cible (CPU, modules de signaux, etc.). Les accès en ligne et les fonctions de test des outils de programmation sont possibles comme avec une CPU réelle. Le programme peut donc être entièrement testé au laboratoire de développement.

IV.9.4.3. Fonction:

S7-PLCSIM exécute le programme utilisateur comme une CPU réelle (fonctions spéciales telles que sécurité uniquement sous certaines conditions).

Durant l'exécution du programme, différentes valeurs du processus peuvent être visualisées et modifiées à travers une simple interface utilisateur (par ex. activation/désactivation d'entrées/sorties).

IV.9.4.4. Le simulateur des programmes PLCSIM :

L'application de simulation de modules S7-PLCSIM permet d'exécuter et de tester le programme dans un automate programmable (AP) qu'on simule dans un ordinateur ou dans une console de programmation. La simulation étant complètement réalisée au sein du logiciel STEP7, il n'est pas nécessaire qu'une liaison soit établie avec un matériel S7 quelconque (CPU ou

module de signaux). L'AP S7 de simulation permet de tester des programmes destinés aux CPU S7-300 et aux CPU S7-400, et de remédier à d'éventuelles erreurs .

S7-PLCSIM dispose d'une interface simple permettant de visualiser et de forcer les différents paramètres utilisés par le programme (comme, par exemple, d'activer ou de désactiver des entrées). Tout en exécutant le programme dans l'AP de simulation, on a également la possibilité de mettre en œuvre les diverses applications du logiciel STEP 7 comme, par exemple, la table des variables (VAT) afin d'y visualiser et d'y forcer des variables.

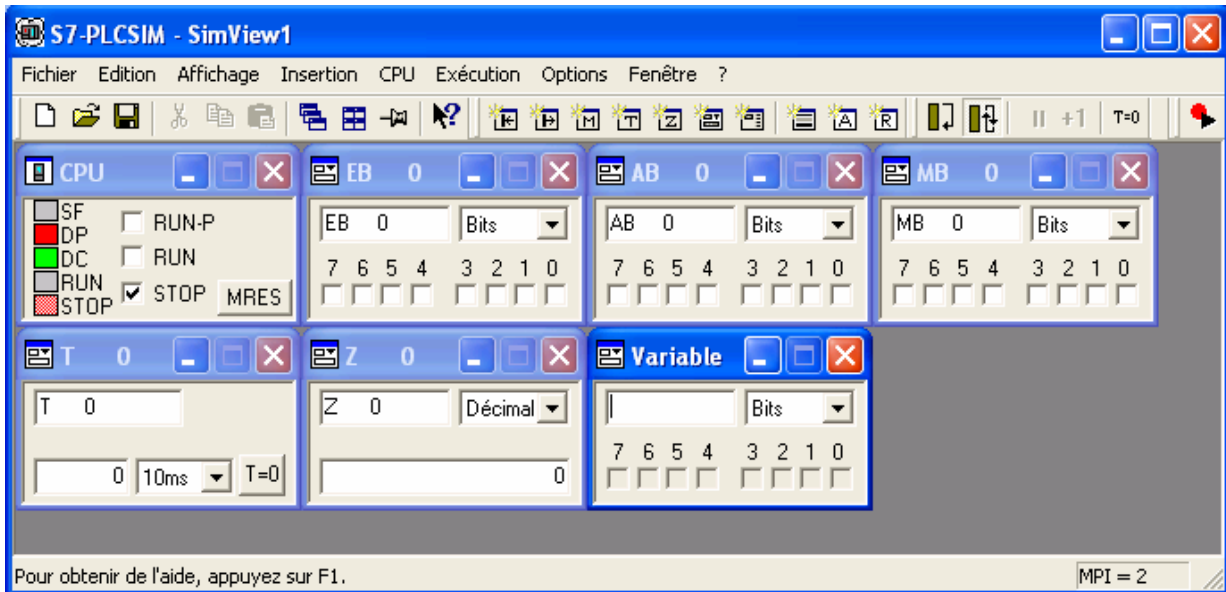


Fig.(IV.16): fenêtre de simulateur PLC-SIM

IV.9.5. S7-SCL:

S7-SCL (Structured Control Language) correspond au langage littéral structuré ST (Structured Text) défini dans la norme CEI 61131-3 et est conforme au Base Level et au Reusability Level selon PLC open. S7-SCL convient notamment pour la programmation d'algorithmes et de fonctions mathématiques complexes ou pour des tâches de traitement de l'information.

IV.9.6. S7-GRAPH:

IV.9.6.1. Programmation de commandes séquentielles:

Le logiciel SIMATIC S7-GRAPH s'appuie sur l'environnement d'ingénierie STEP 7. Il

sert à la description de commandes séquentielles à l'aide de séquences exclusives et simultanées

Le déroulement des étapes est configuré et programmé avec clarté et rapidité dans un mode de représentation standardisé (selon CEI 61131-3, DIN EN 61131).

IV.9.6.2. Avantages par rapport à STEP 7 Basis:

- CONT, LOG et LIST sont axés sur la commande combinatoire. S7-GRAPH est orienté sur le déroulement du processus
- Représentation graphique claire du processus au moyen de séquences, d'où une maintenance facilitée et la possibilité d'adapter les programmes.
- Recherche de défauts dans le processus par des fonctions de diagnostic intégrées, d'où une réduction des arrêts de production et des coûts occasionnés.

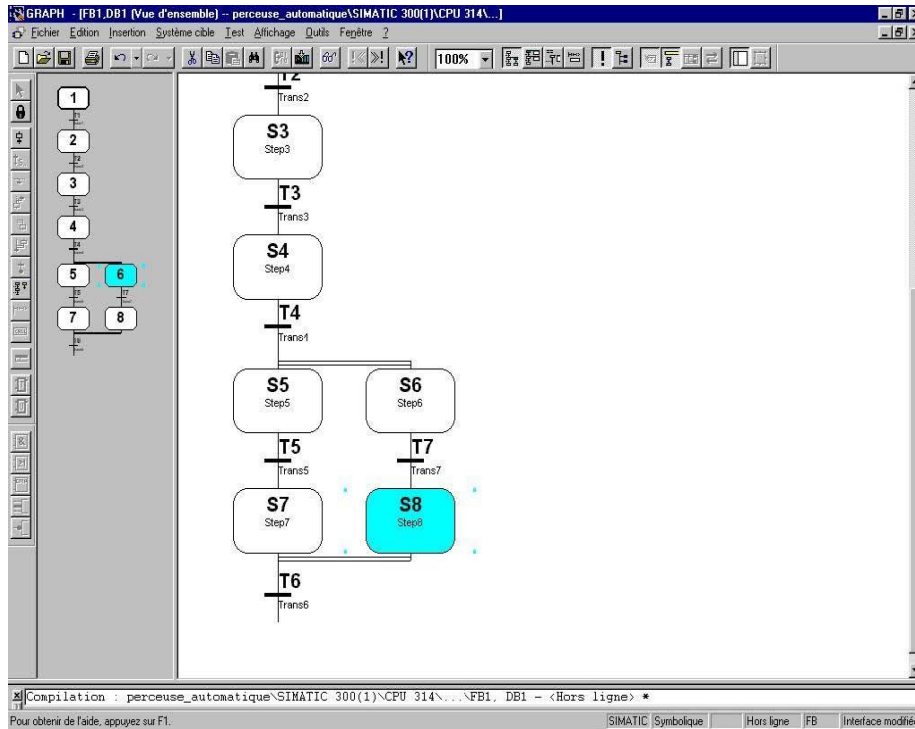


Fig.(IV.17):fenêtre de S7-GRAPH

IV.10. Conclusion :

Dans ce chapitre on a vu une description détaillée de logiciel STEP7, et les applications qui la concernent. Et aussi les langages de programmation qu'il contient comme contacts et logigramme, et comment créer un projet dans le logiciel STEP7, et la méthode de simulation d'un projet.

CHAPITRE V

Etude et simulation

D'un

Bras manipulateur

V.1. Introduction:

Les premières applications de la robotique étaient la saisie et le transfert d'objets, le moulage par injection et l'emboutissage où le robot est simplement équipé d'une presse pour décharger et transférer ou empiler la partie finie. Ces premiers robots étaient capables d'être programmés pour exécuter une séquence de mouvement, comme aller à un emplacement donné, fermer une pince, etc. mais n'avaient pas de capteurs externes. Certaines applications plus compliquées, comme la soudure, le meulage, l'ébavurage, et l'assemblage exigent non seulement un mouvement plus complexe mais également un certain nombre de capteurs externes comme la vision et les capteurs de force vu la forte interaction du robot avec son environnement.

Aujourd'hui les robots ont pu remplacer l'homme dans son travail directement en ayant la possibilité d'effectuer la quasi-totalité des travaux industriels.

Le robot industriel est un instrument sophistiqué, dont le rôle est principalement la manipulation de pièces ou d'outils. Sa conception fait appel à des compétences dans divers domaines tels que la mécanique l'électrotechnique ou l'hydraulique, l'informatique, l'électronique numérique, etc...

Les progrès réalisées dans un seul de ces domaines permettant d'améliorer les performances de l'ensemble: de ce fait, les besoins de l'industrie qui étaient faibles il y a quelques années ne cessent d'augmenter avec simultanément un élargissement des domaines d'utilisation .

V.2. Définition :

Un bras manipulateur est un système composé de pièces mécaniques, de composants électriques et électronique qui remplit un ensemble de fonctions spécialisées. Le contrôle du robot s'effectue par le biais de circuits électriques et électroniques qui opèrent des variations de courant et de voltage de façon à obtenir un comportement donné.

Une manière de décrire le bras consiste à préciser leur architecture (ou leur morphologie). Cette description donne des indications sur la façon dont les divers segments d'un bras sont organisés et comment ils se déplacent les uns par rapport aux autres.

Ce sont les équipements les plus utilisés en milieu industriel, ils sont conçus pour saisir et manipuler des objets avec ou sans assistance humaine.

Ils réalisent des tâches répétitives et souvent pénibles pour l'homme. Ils sont rapides, précis, fidèles et manipulent des objets en milieu inaccessible, ou des produits dangereux ou en milieu nocif pour nous humains.

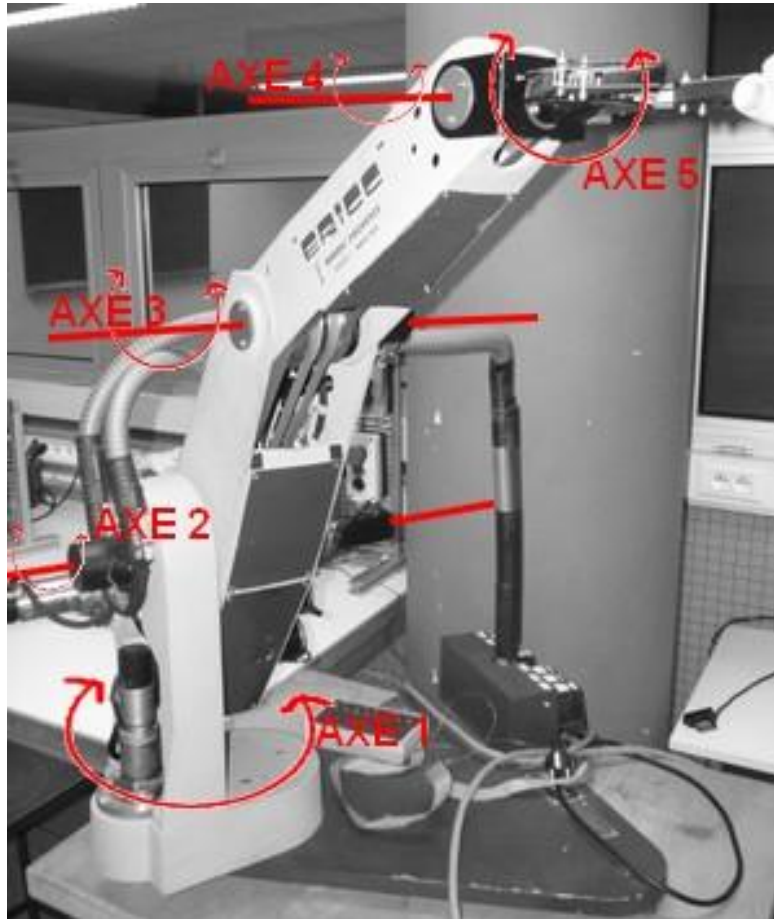


Fig.(V.1): le bras manipulateur avec ses axes

V.3. Utilisation d'un bras manipulateur

Le bras, facilement transportable, permet d'effectuer les tâches de manutention les plus diverses:

- Chargement/ déchargement et postes de production.
- Rangement des pièces ou d'outils sur magasin intermédiaire.
- Montage/ démontage simple.

V.4. Choix technologiques :

Les matériaux et les dimensions du bras manipulateur sont déterminés pour une utilisation dans des milieux divers.

Les moteurs d'axes peuvent être du type pas à pas ou du type courant continu, suivant que l'on privilégie la facilité de positionnement ou la facilité de mise en œuvre.

On utilisera des moteurs à courant continu pour la facilité de mettre en œuvre de systèmes asservis analogiques.

V.5. Principe d'un bras manipulateur :

Il se compose de plusieurs parties comme :

- ✚ Le socle
- ✚ La pince.
- ✚ Le coude
- ✚ l'épaule.

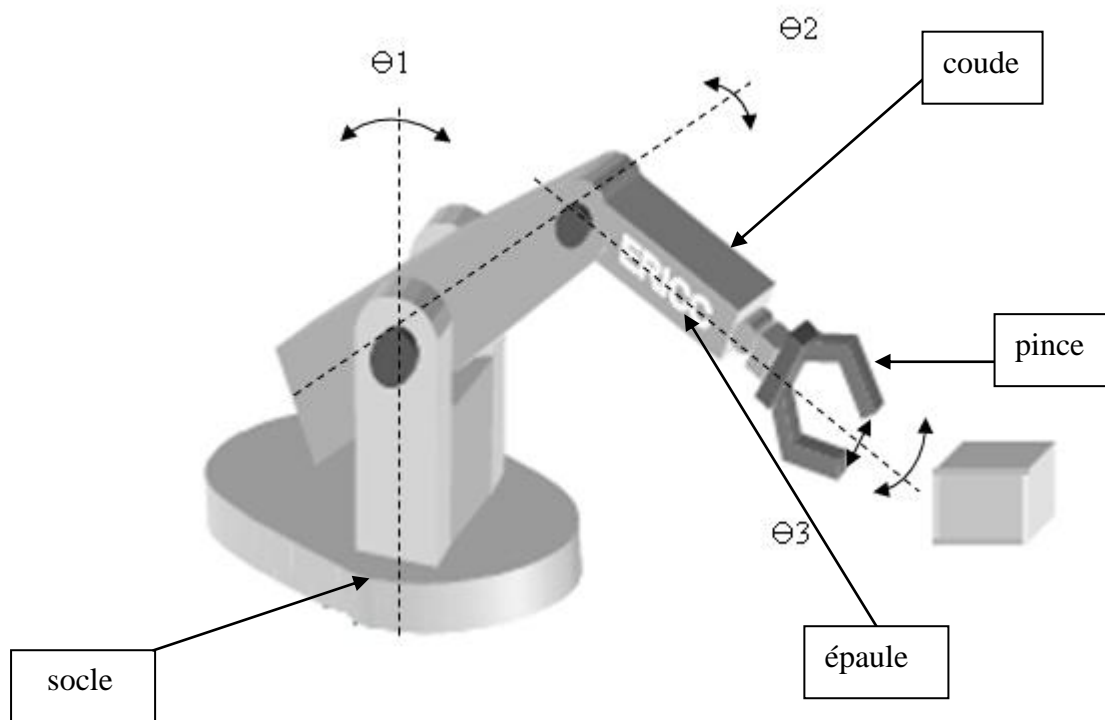


Fig.(V.2): Composante d'un bras manipulateur

Il peut comporter plusieurs degrés de liberté :

On définit ainsi les mouvements de :

bras (θ_1), Epaule (θ_2), Coude (θ_3), , ouverture et fermeture pince.

Les plages de mouvement minimales et maximales pour chaque articulation, sont :

-1^{ème} articulation (bras) :

1^{ème} degré de liberté 180° (rotation du bras).

- 2^{ème} articulation (épaule) :

2^{ème} degré de liberté 45° (translation de l'épaule).

- 3^{ème} articulation (coude) :

3^{ème} degré de liberté 90° (translation du coude).

- 4^{ème} articulation (pince) :

Ouverture et fermeture pince.

V.6. Moteur a courant continue :

Le moteur à courant continu est un convertisseur électrique en énergie mécanique avec quelques pertes.

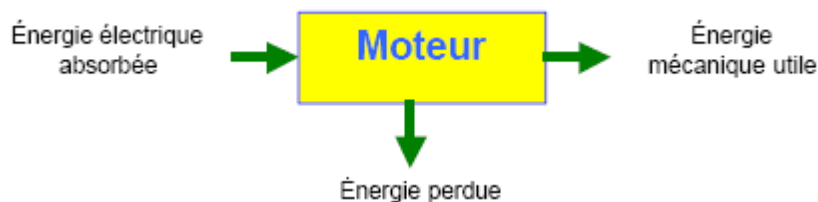


Fig.(V.3) : billon énergétique d'un moteur à courant continu

L'usage des moteurs à courant continu est plutôt restreint, car la distribution se fait à courant alternatif .Cependant, pour certaines applications il est parfois avantageux d'utiliser des

moteurs à courant continu alimentés par des convertisseurs qui transforment le courant alternatif en courant continu.

La supériorité de ces moteurs réside dans le fait qu'ils se prêtent facilement à un contrôle souple, continu et presque instantané de leur vitesse.

Le moteur à courant continu Comprend quatre parties principales: l'inducteur, l'induit, le collecteur et les balais. Nous les examinons successivement dans les section qui suivent :

V.6.1. L'inducteur ou Le stator :

il est formé d'aimant pour les « petits » et de bobine parcouru par un courant pour les moteurs plus puissants. Grâce aux progrès sur les aimants ses derniers sont de plus en plus employés. Les aimants ou bobines permettent la création d'un flux magnétique.

V.6.2. L'induit ou le rotor :

Il est composé d'un ensemble de bobines identique réparties uniformément autour d'un noyau cylindrique. Il est monté sur un arbre et tourne entre les pôles de l'inducteur. L'induit constitue donc un ensemble de conducteurs qui coupent le flux magnétique .

V.6.3. Le collecteur :

le Collecteur est un ensemble cylindrique de lames de cuivre isolées les unes des autres par des feuilles de mica il est monté sur l'arbre de machine, mais isolé de celui-ci les deux fils sortant de chaque bobine de l'induit sont successivement et symétriquement soudés aux lames du collecteur.[15]

V.7.Les détecteurs de position :

Les capteurs mécaniques de position, appelés aussi interrupteurs de position, sont surtout employés dans les systèmes automatisés pour assurer la fonction détecter les positions. On par le aussi de détecteurs de présence.

Ils sont réalisés à base de microcontacts placés dans un corps de protection et muni d'un système de commande ou tête de commande .voir la Fig.(V.4)

V.7.1. Principe :

C'est un commutateur, commandé par le déplacement d'un organe de commande (corps d'épreuve). Lorsque le corps d'épreuve est actionné, il ouvre ou ferme un contact électrique.

De nombreux modèles peuvent être associés au corps : tête à mouvement rectiligne, angulaire ou multi direction associée à différents dispositifs d'attaque (à poussoir, à levier, à tige). [23]

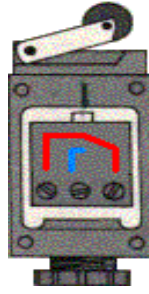


Fig.(V.4) : détecteur de position

V.8. Cahier des charges :

Le bras manipulateur à automatiser permet de prendre et chargement ou déchargement des pièces. Ce bras se compose de :

- **Quatre moteurs à courant contenu :**

- 1) Le premier moteur pour la rotation vers la droite et la gauche du bras.
- 2) La deuxième moteur pour la rotation au niveau bas et haut de l'épaule.
- 3) La troisième moteur pour la rotation au niveau bas et haut du coude.
- 4) La quatrième moteur pour l'ouverture et fermeture de la pince .

- **Sept détecteurs de position :**

pour les moteurs (épaule, coude) chaque moteur besoin deux détecteurs de position et le premier moteur du bras à besoin de trois détecteurs .

- **Les détecteurs de position définir :**

Les positions de descente et de monté :

- Niveau bas pour l'épaule **E125.4**
- Niveau bas pour coude **E125.6**
- Niveau haut pour l'épaule **E125.3**
- Niveau haut pour coude **E125.5**
- Position initial **E125.0**

Les rotations possible :

- Vers la gauche **E125.1**
- Vers la droite **E125.2**

Il n'y a aucun capteur qui indique que la pince est ouverte ou fermée. Pour effectuer ces de opération mais surtout contrôler le temps d'ouverture ou de fermeture.

- ouverture de la pince et alimentation de T1 (30s) en fin de temporisation .
- fermeture de la pince et alimentation de T2 (10s) en fin de temporisation .

Le bras manipulateur tourne vers la droite, l'épaule décent à une angle de 45° , après le coude décent à une angle de 90° , ouverture de la pince après fermeture de la pince. donc le pince ramène l'objet (la pièce), après cet étape le coude monte vers la même angle et le bras tourne à une angle de 90° vers la gauche et après le coude décent de la même angle et après l'ouverture de la pince donc l'objet est posé, le coude monte à une angle 90° et après le fermeture de la pince et finalement l'épaule monte à 45° .

V.9. Programmation :

Dans ce chapitre , on va essayer d'exploiter les informations que nous avons déjà connues dans les chapitres précédents pour automatiser le bras manipulateur avec l'automate programmable S7 300 , on utilisant le logiciel STEP 7 pour la programmation , et le logiciel PLCSIM pour la simulation.

V.9.1. Création d'un projet avec STEP7 :

Le logiciel SIMATIC Manager , nous permet de créer de nouveaux projet à l'aide de L'assistant selon la fig.(V.5).

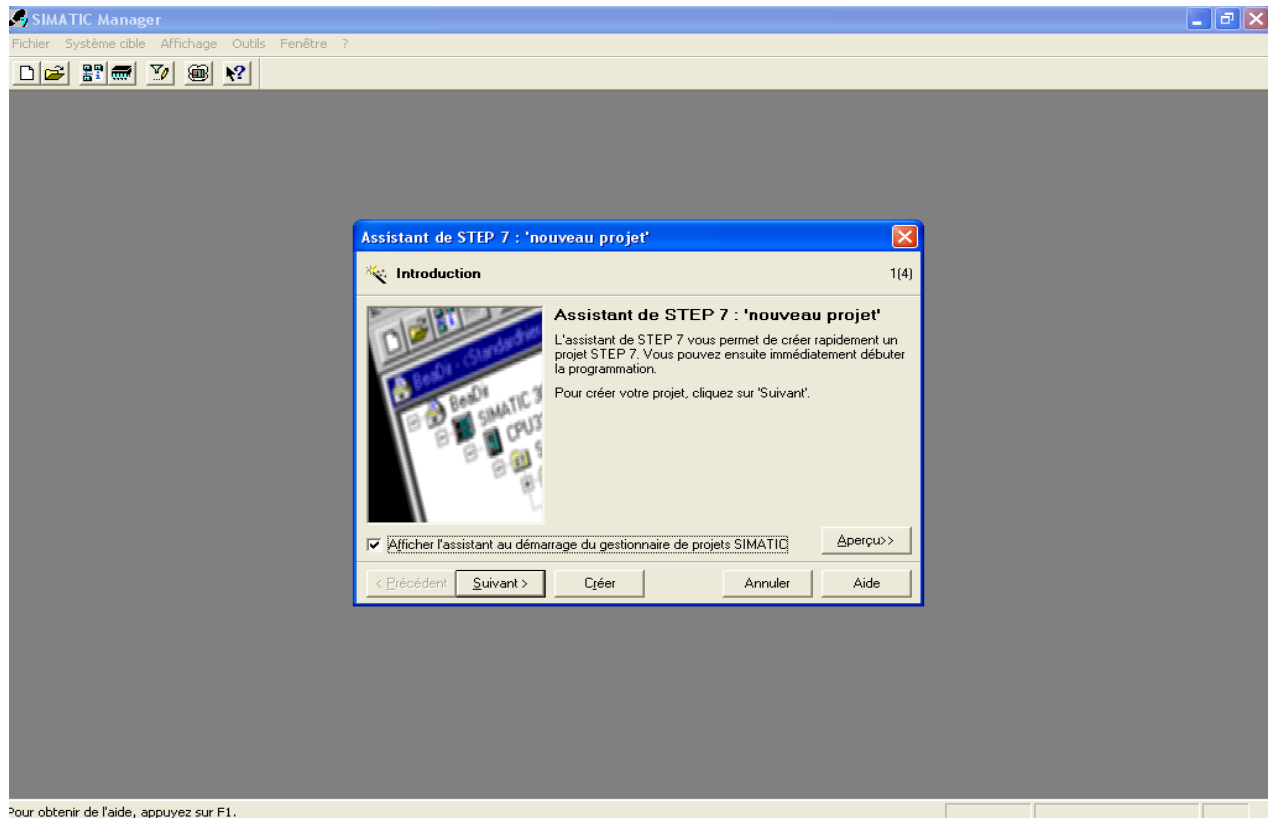


Fig.(V.5) : assistant nouveau projet

Dans notre projet on choisi le langage de programmation à contacte , c'est le langage le plus utilisé par les automaticiens .

Les étapes précédentes nous permettent d'obtenir une nouvelle fenêtre qui porte le nom de notre projet fig.(V.6).

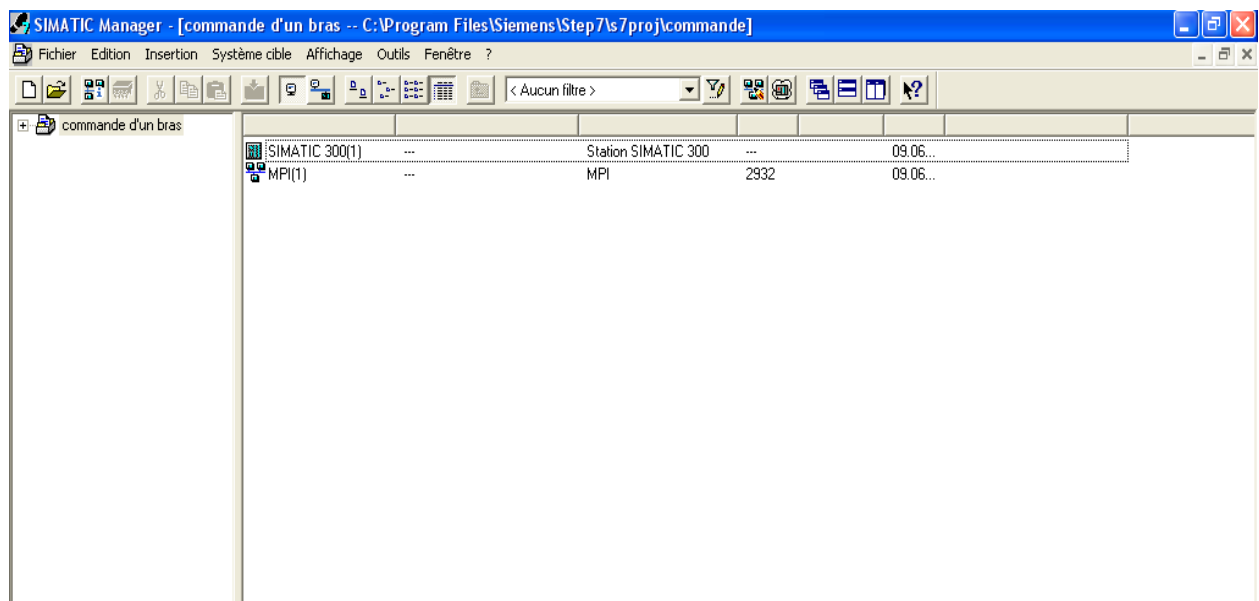


Fig.(V.6) : vue des composants d'un projet

V.9.2. Configuration matérielle :

La fenêtre de projet est partagée en deux volets comme l'indique la figure .le volet gauche représente l'arborescence du projet .celui du droite affiche le contenu de l'objet sélectionné dans le volet gauche . la sélection de la bras manipulateur nous engage à la configuration de notre matériel physique à l'aide du logiciel de configuration – HW config- , ce dernier nous permet de choisir des matériels de l'application .

Dans la fenêtre de configuration de matériel ,il apparaît plusieurs cases dont à la première case on choisi l'alimentation – PS307 2A- , a la deuxième case on met le processeur – CPU314IFM et dans les autre emplacements on choisi les modules des signaux – DI,DO

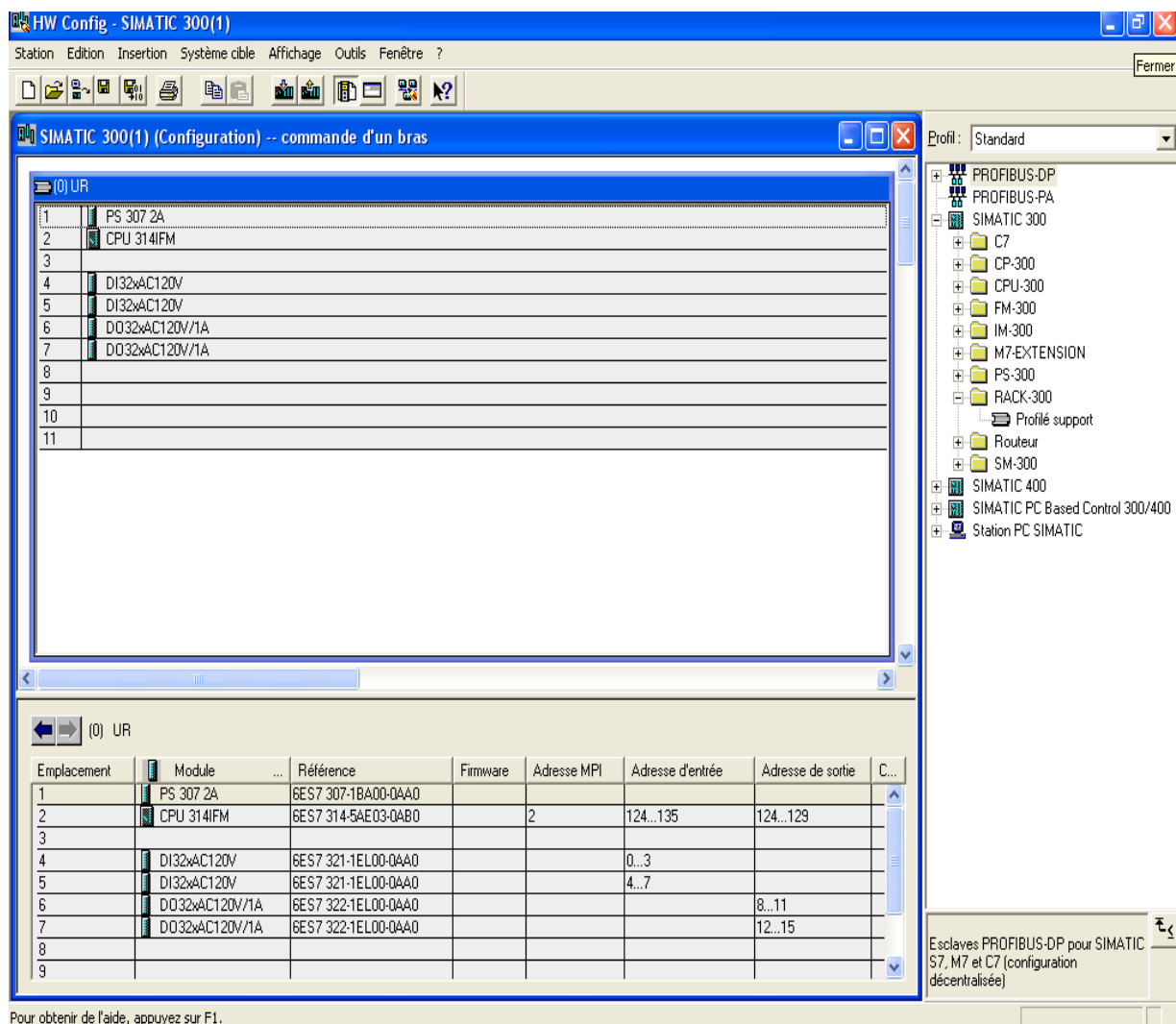


Fig.(V.7) :configuration matériel(Modules)

V.9.3. Création du programme :

Pour créer un programme avec STEP7 nous avons besoin de déclarer les entrées , les sorties Et les mémoros utilisées dans notre projet .

Editeur de mnémoniques - [Programme S7(1) (Mnémoniques) -- commande d'un brasSIMA...]

Table Edition Insertion Affichage Outils Fenêtre ?

Tous les mnémoniques

	Etat	Mnémonique ▲	Opérande	Type de don	Commentaire
1		ar-ba	E 126.1	BOOL	arrêt le mouvement du bras-moteur1-manuel
2		ar-co	E 126.5	BOOL	arrêt le mouvement de coude -moteur3-manuel
3		ar-ep	E 126.3	BOOL	arrêt le mouvement de l'épaule -moteur2-manuel
4		ar-p	E 126.7	BOOL	arrêt fermeture de la pince-moteur4-manuel
5		D1	E 124.2	BOOL	arrêt d'urgence
6		de-ba-co	E 125.6	BOOL	détecteur de position bas-coude-
7		de-ba-ep	E 125.4	BOOL	détecteur de position bas-épaule-
8		de-dr-br	E 125.1	BOOL	détecteur de position droite -bras-
9		de-ga-br	E 125.2	BOOL	détecteur de position gauche -bras-
10		de-ht-co	E 125.5	BOOL	détecteur de position haut-coude-
11		de-ht-ep	E 125.3	BOOL	détecteur de position haut-épaule-
12		et-in-ep	A 124.0	BOOL	étape initiale-mouvement de l'épaule vers le haut-moteur2-
13		fe1-pin	A 125.6	BOOL	fermeture de la pince -moteur4-
14		fe-pin	A 124.6	BOOL	fermeture de la pince-moteur4-
15		f-pin	E 127.0	BOOL	fermeture de la pince-moteur4-manuel
16		f-pin-m	A 126.7	BOOL	fermeture de la pince -moteur4-manuel
17		m-au/man	E 124.0	BOOL	marche auto/ manuel
18		m-co1-ba	A 125.3	BOOL	mouvement de coude vers la bas-moteur2-
19		m-co1-ht	A 125.5	BOOL	mouvement de coude vers le haut-moteur3-
20		m-co-ba	A 124.4	BOOL	mouvement de coude vers la bas -moteur3-
21		m-co-ba-m	A 126.4	BOOL	mouvement de coude vers la bas -moteur3-manuel
22		m-co-ht	A 124.7	BOOL	mouvement de coude vers le haut
23		m-co-ht-m	A 126.5	BOOL	mouvement de coude vers le haut -moteur3-manuel
24		m-dr	A 125.2	BOOL	mouvement le bras vers la droite-moteur1-
25		m-dr-m	A 126.0	BOOL	le bras marche vers la droite-moteur1-manuel
26		m-ep-ba	A 124.3	BOOL	mouvement d'épaule vers la bas -moteur2-
27		m-ep-ba-m	A 126.2	BOOL	mouvement d'épaule vers la bas -moteur2-manuel
28		m-ep-ht-m	A 126.3	BOOL	mouvement d'épaule vers le haut -moteur2-manuel
29		m-gau	A 125.1	BOOL	mouvement de bras vers la gauche-moteur1-
30		m-gau-m	A 126.1	BOOL	le bras marche vers la gauche-moteur1-manuel
31		mr-dr	A 124.1	BOOL	le bras marche vers la droite-moteur1-
32		mr-gau	A 124.2	BOOL	le bras marche vers la gauche-moteur1-
33		o-pin	E 126.6	BOOL	ouverture de la pince-moteur4-manuel
34		o-pin-m	A 126.6	BOOL	ouverture de la pince -moteur 4-manuel
35		ou1-pin	A 125.4	BOOL	ouverture de la pince -moteur4-
36		ou-pin	A 124.5	BOOL	ouverture de la pince-moteur4-
37		pos-0-	E 125.0	BOOL	détecteur de position initiale-0-
38		r/droite	E 124.4	BOOL	sens de rotation vers la droite-moteur1-
39		r/gauche	E 124.5	BOOL	sens de rotation vers la gauche-moteur1-
40		relth	E 124.3	BOOL	relais thermique
41		se-co-ba	E 126.4	BOOL	sens de rotation vers la bas -coude-moteur 3-manuel
42		se-ep-ba	E 126.2	BOOL	sens de rotation vers la bas -épaule -moteur2-manuel
43		se-ht-co	E 127.1	BOOL	sens de rotation vers le haut-coude-moteur3-manuel
44		se-ht-ep	E 127.2	BOOL	sens de rotation vers le haut -épaule-moteur2-manuel
45		tm1	T 1	TIMER	temporisation pour l'ouverture 1 de la pince 30s
46		tm2	T 2	TIMER	temporisation pour fermeture 2 de pince 10 s
47		tm3	T 3	TIMER	temporisation pour l'ouverture 2 de la pince 10s
48		tm4	T 4	TIMER	temporisation pour fermeture 2 de la pince 30s
49		vol	E 124.1	BOOL	disponibilité d'électricité
50					

Pour obtenir de l'aide, appuyez sur F1. NUM

Fig.(V.8) : table mnémotique

L'automatisation et la surveillance de notre projet nécessite un bloc d'organisation et deux fonctions fig.(V.9).

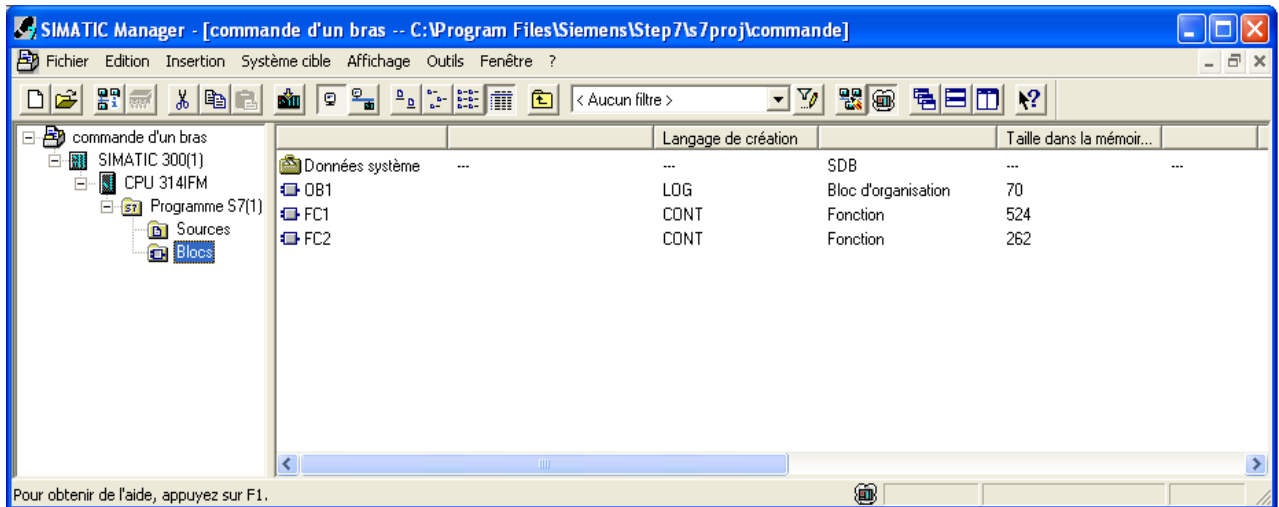



Fig.(V.9) : création du programme

V.10. Simulation du programme :

Après la réalisation du programme d'automatisation de bras manipulateur ,on peut simuler notre programme à l'aide des étapes suivantes :

1- lancement et configuration de S7-PLCSIM pour tester le programme sans connecter la PG à un automate , il suffit d'activer le simulateur .tous les accès à l'interface de l'automate sont simuler de manière interne par logiciel de simulation S7-PLCSIM .

Pour lancer PLCSIM ,il faut activer le << simulateur >> par l'icône: 

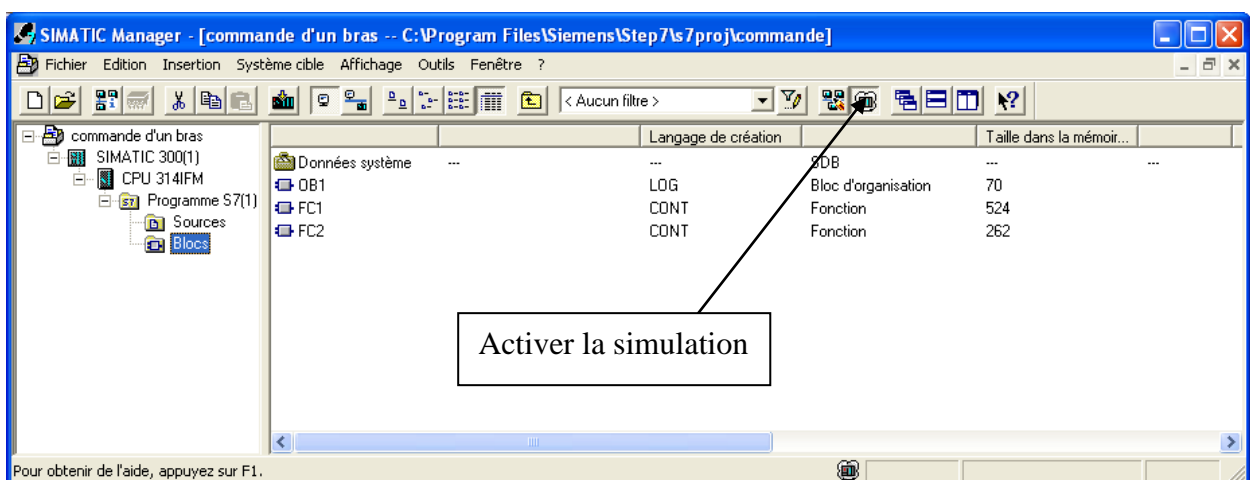


Fig.(V.10) : activer- désactiver le simulateur à partir de l'interface

2- il nous reste plus qu'insérer à l'aide de menu << insertion >> toutes les << entrées >> et <<sortie>> utilisées dans le programme que nous désirons tester . << memento>> , << temporisation >> et << compteurs>> peuvent être également représentées fig.(V.11) .

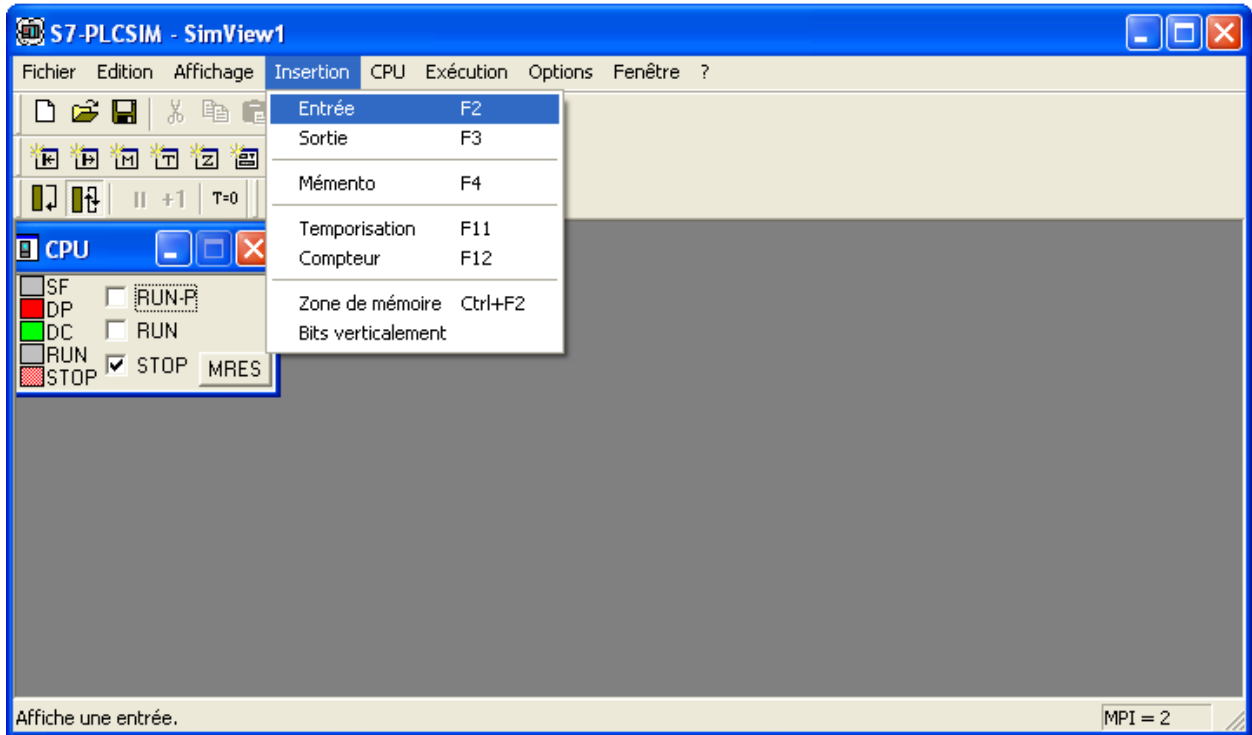


Fig.(V.11) : l'environnement de S7-PLCSIM

Il suffit d'entrer les adresses voulues , EB , AB et le mode de représentation par bits . fig.(V.12)

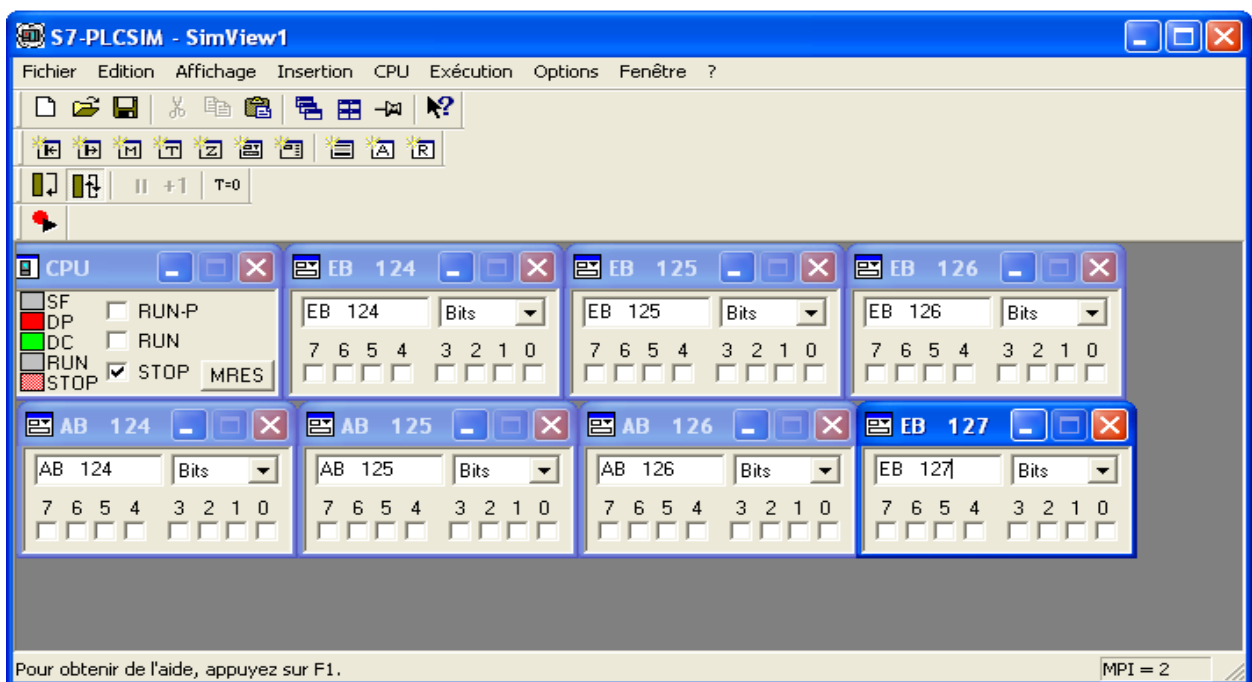


Fig.(V.12): réglage des paramètres

On peut maintenant charger le programme S7 à tester dans l'automate simulé :

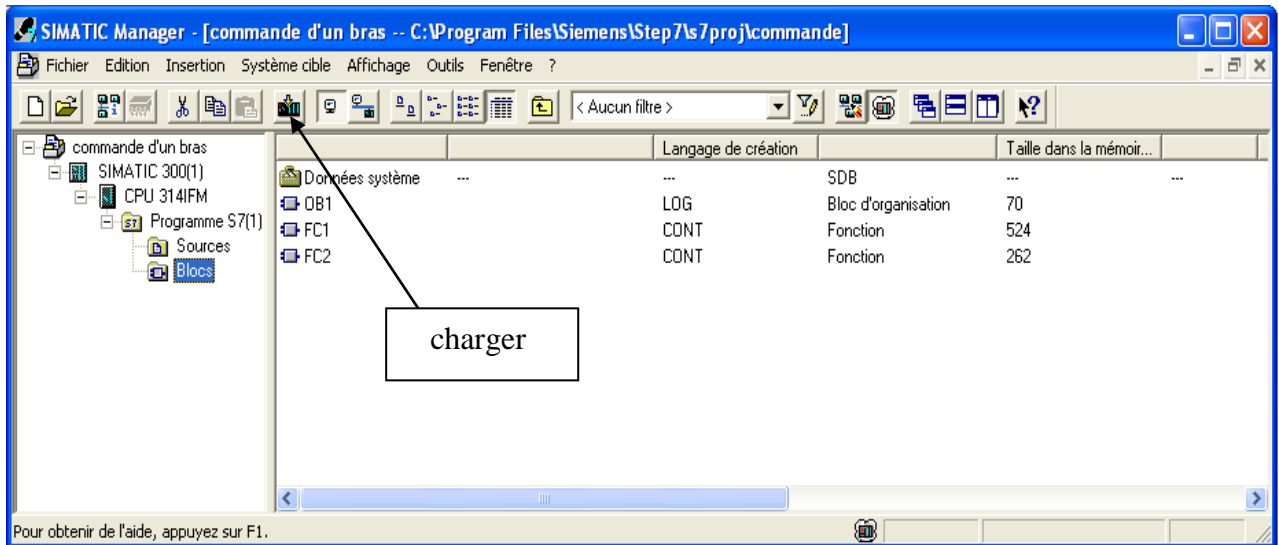


Fig.(V.13) : chargement du programme à simuler

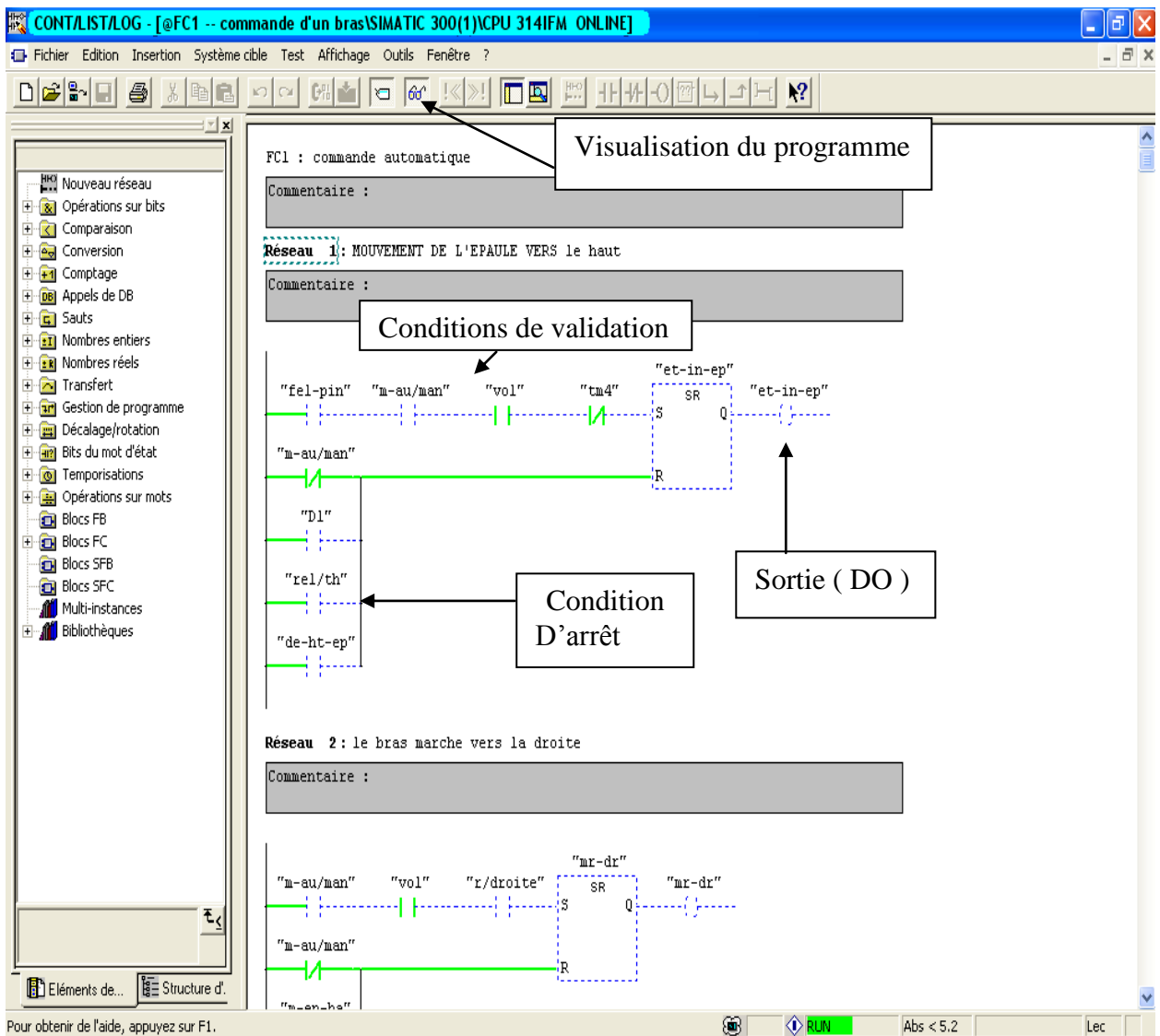


Fig.(V.14) : visualisation des programme

V.10.1. Description du programme :

Afin d'illustrer notre travail , nous donnons comme exemple le programme détaillé pour une partie du programme (moteur1 : marche automatique , sens droit, réseau 2) .

V.10.1.1. Conditions de validation :

Pour que le moteur 1 fonctionne il faut :

- Sélectionner le bouton insertion (E124.4, sens de rotation vers la droite) .
- Sélectionner le bouton insertion (E124.0. marche auto / manuel) .
- Sélectionner le bouton insertion (E124.1, disponibilité d'électricité) .

V.10.1.2. Conditions d'arrêt :

Le moteur 1 s'arrête s'il y a une de ces conditions :

- Le bouton d'arrêt d'urgence (E124.2) .
- Le bouton auto-man (E124.0) .
- Le relais thermique (E124.3) .
- Mouvement d'épaule vers la bas (moteur2, A124.3) .

V.11. Le programme de Commande automatique d'un bras manipulateur :

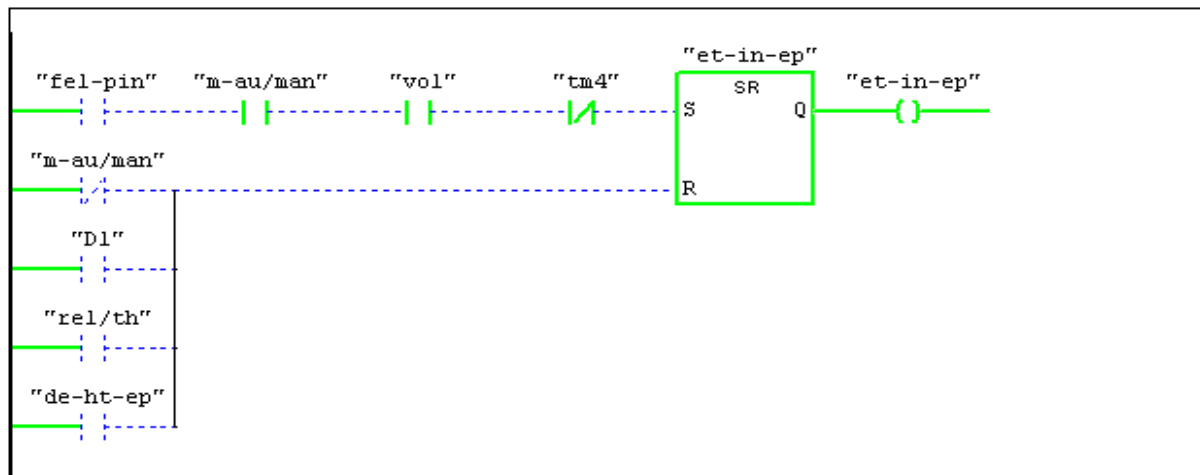
Le bloc FC1 représente le programme automatique :

FC1 : COMMANDE AUTOMATIQUE

Commentaire :

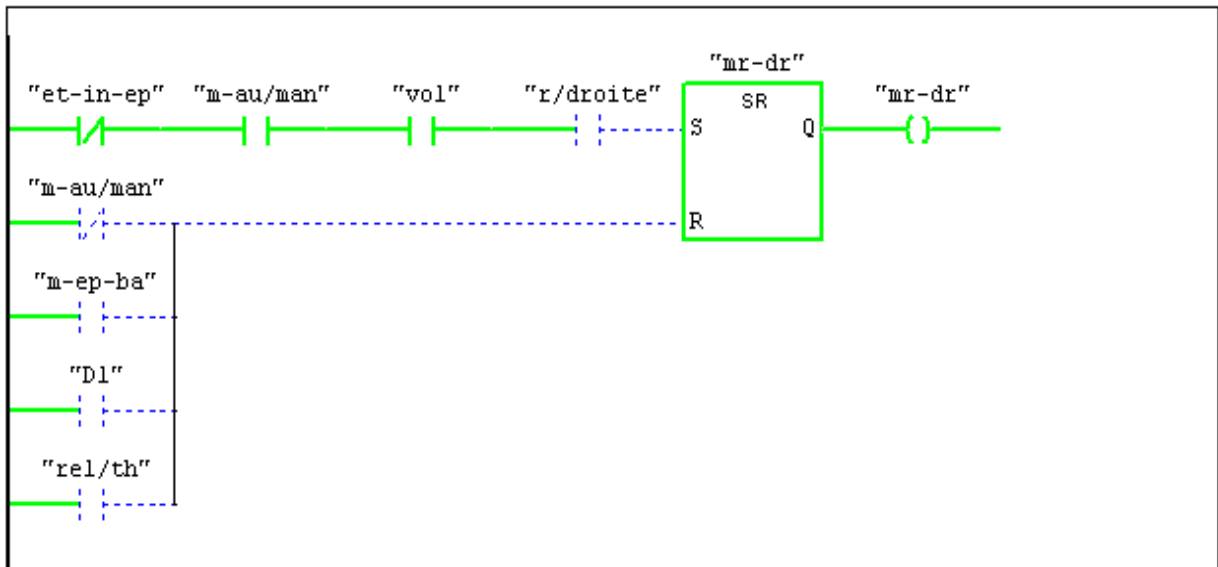
Réseau 1: ETAPE INITIALE (mouvement d'épaule vers le haut)

Commentaire :



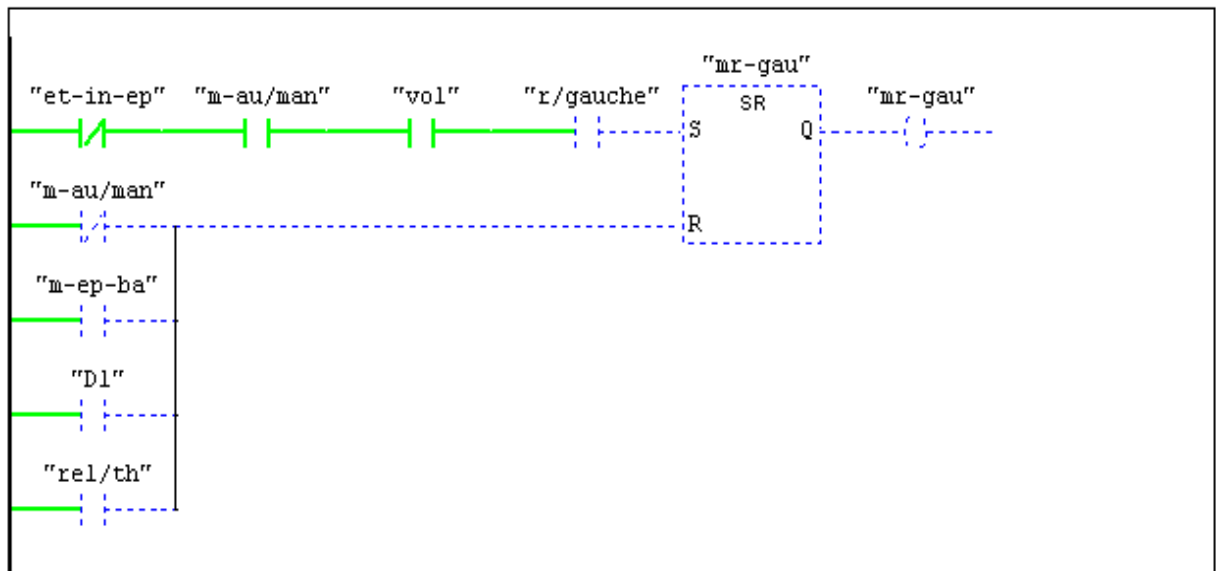
Réseau 2: le bras marche vers la droite

Commentaire :



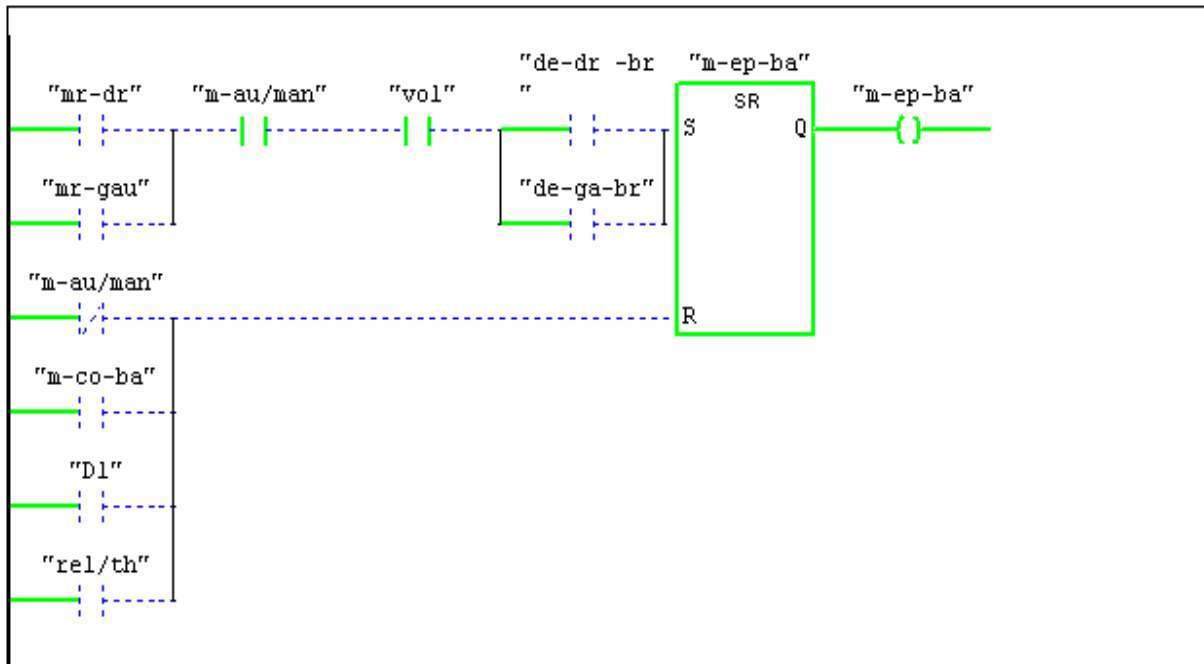
Réseau 3: le bras marche vers la gauche

Commentaire :



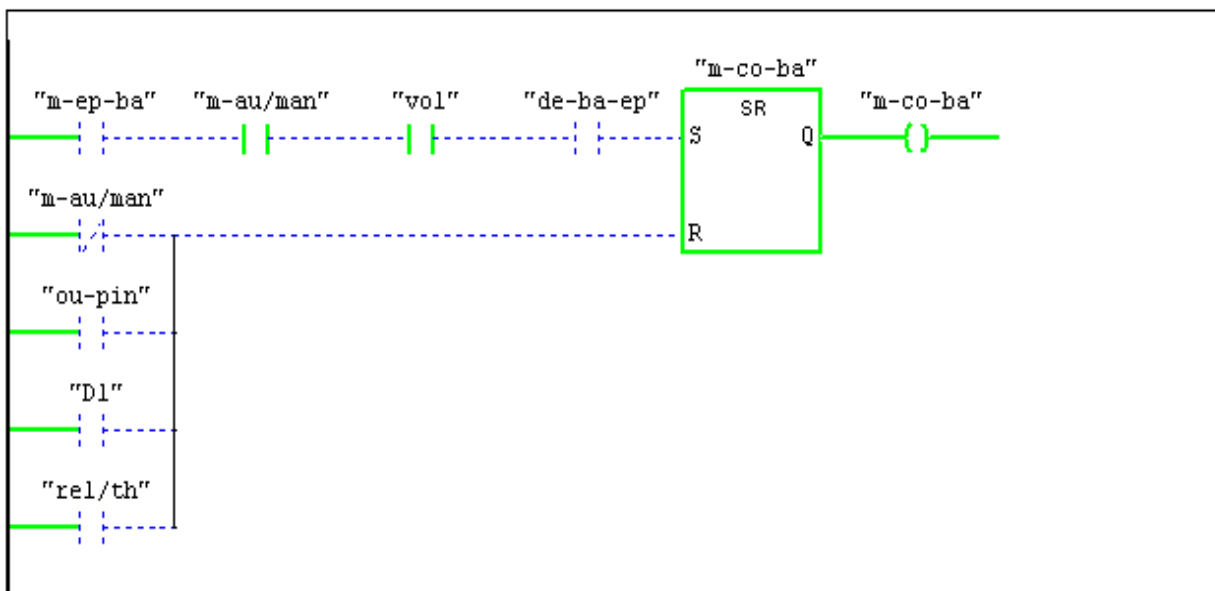
Réseau 4 : mouvement d'épaule vers la bas

Commentaire :



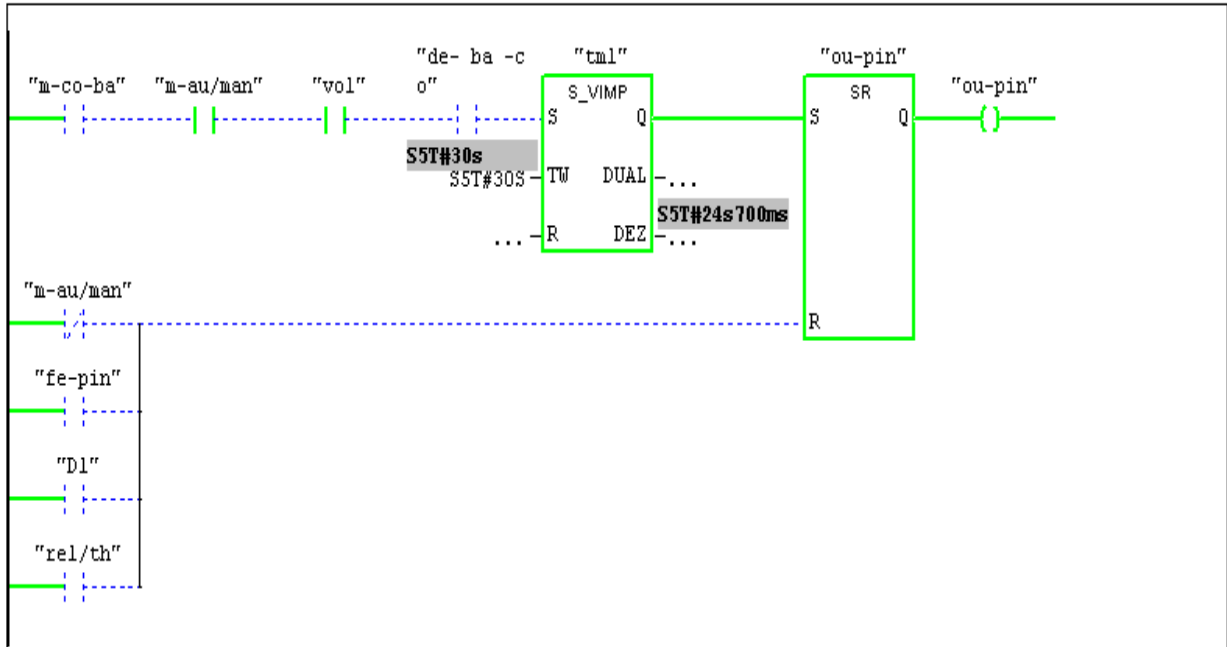
Réseau 5 : mouvement de coude vers la bas

Commentaire :



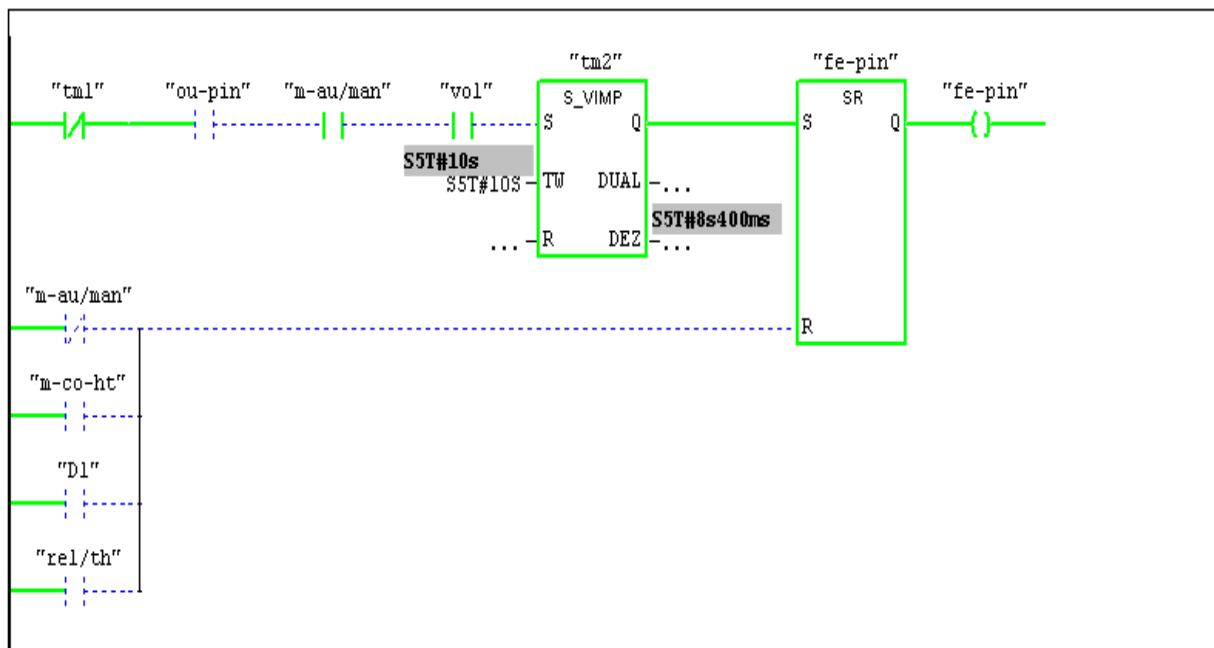
Réseau 6: ouverture de la pince

Commentaire :



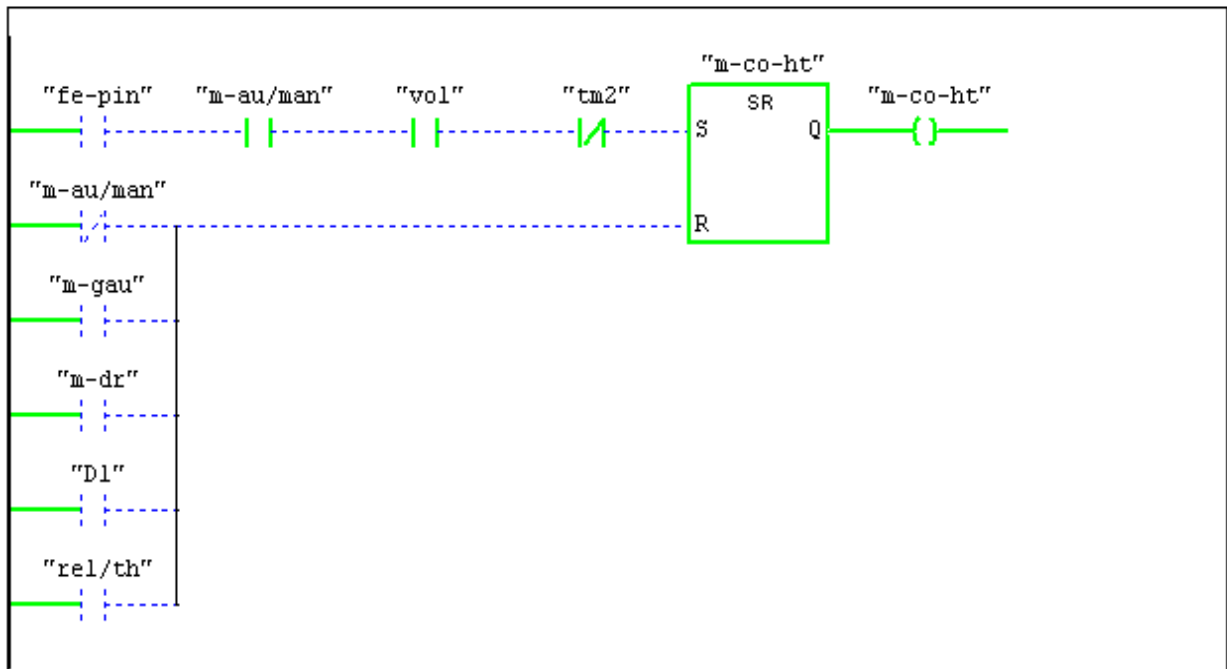
Réseau 7: fermeture de la pince

Commentaire :



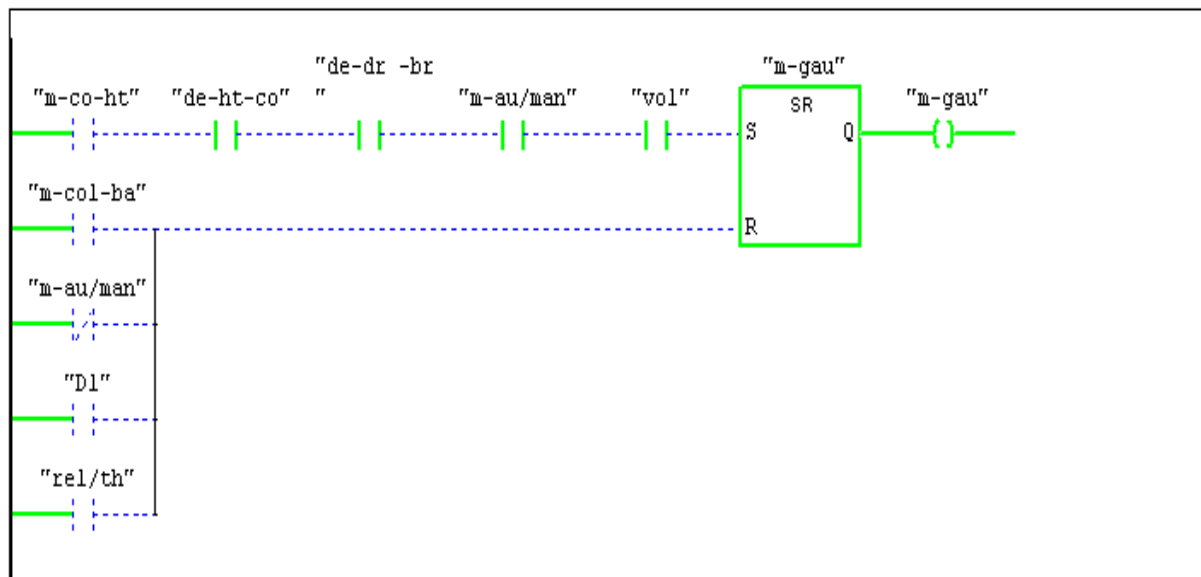
Réseau 8: mouvement de coude vers le haut

Commentaire :



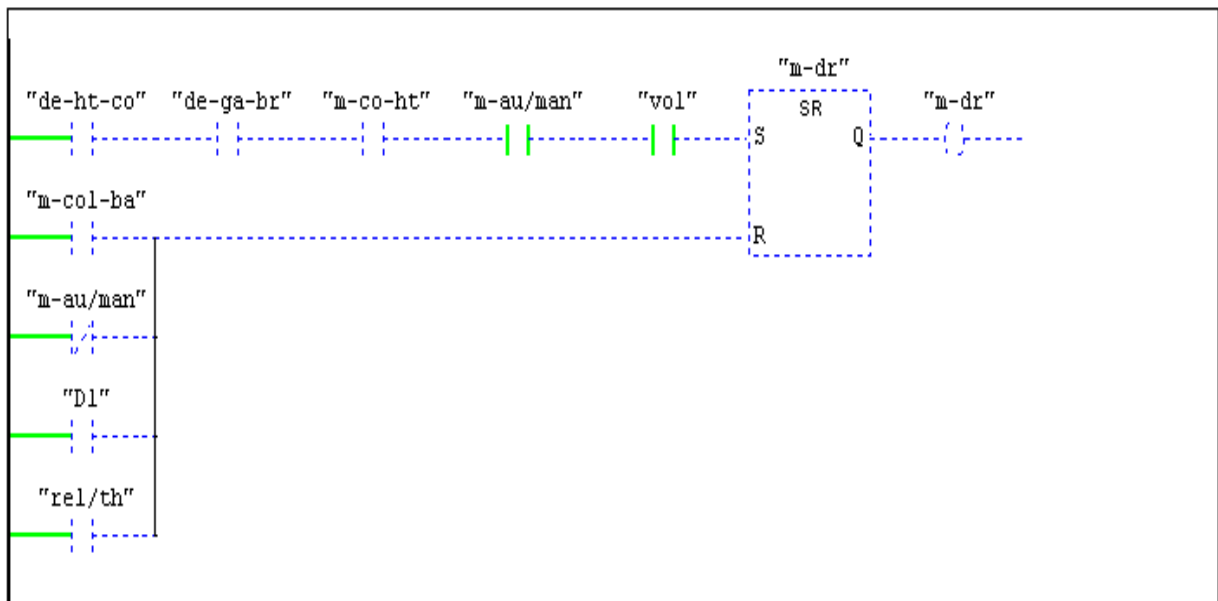
Réseau 9: mouvement de bras vers la gauche

Commentaire :



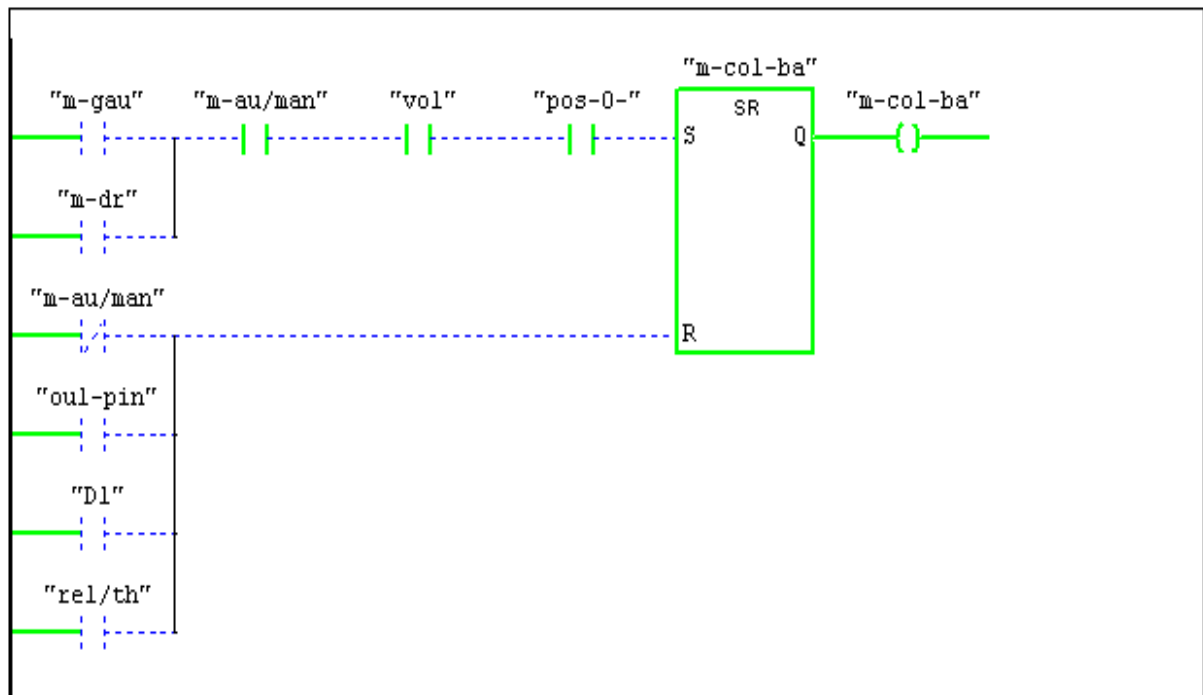
Réseau 10: mouvement de bras vers la droite

Commentaire :



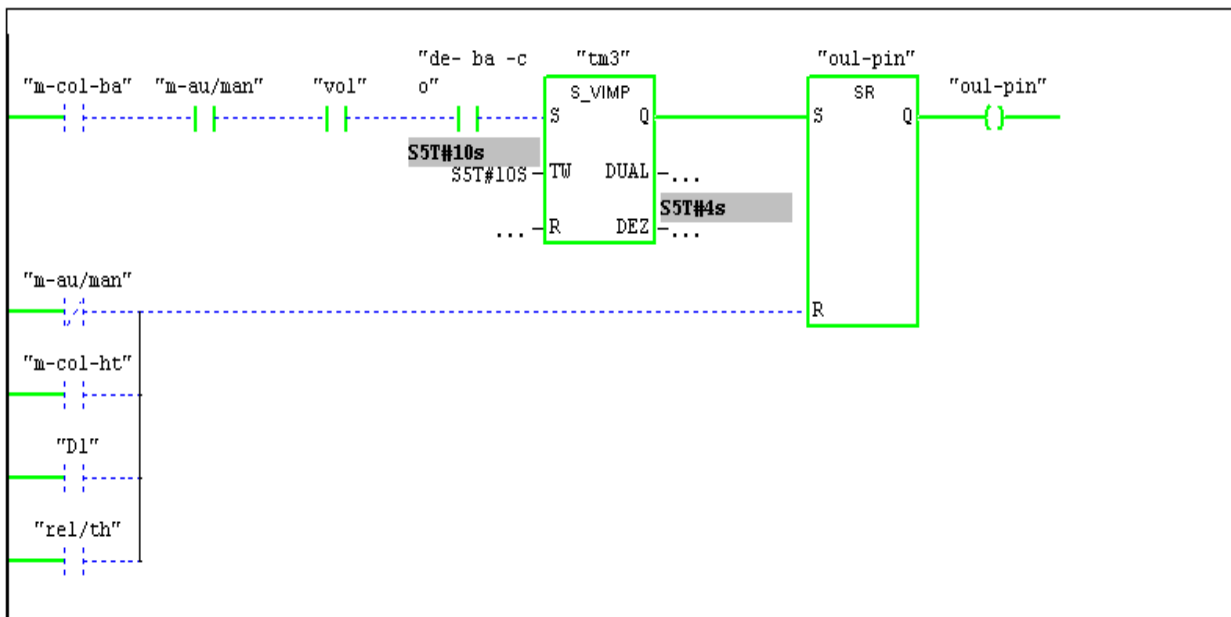
Réseau 11: mouvement de coude vers la bas

Commentaire :



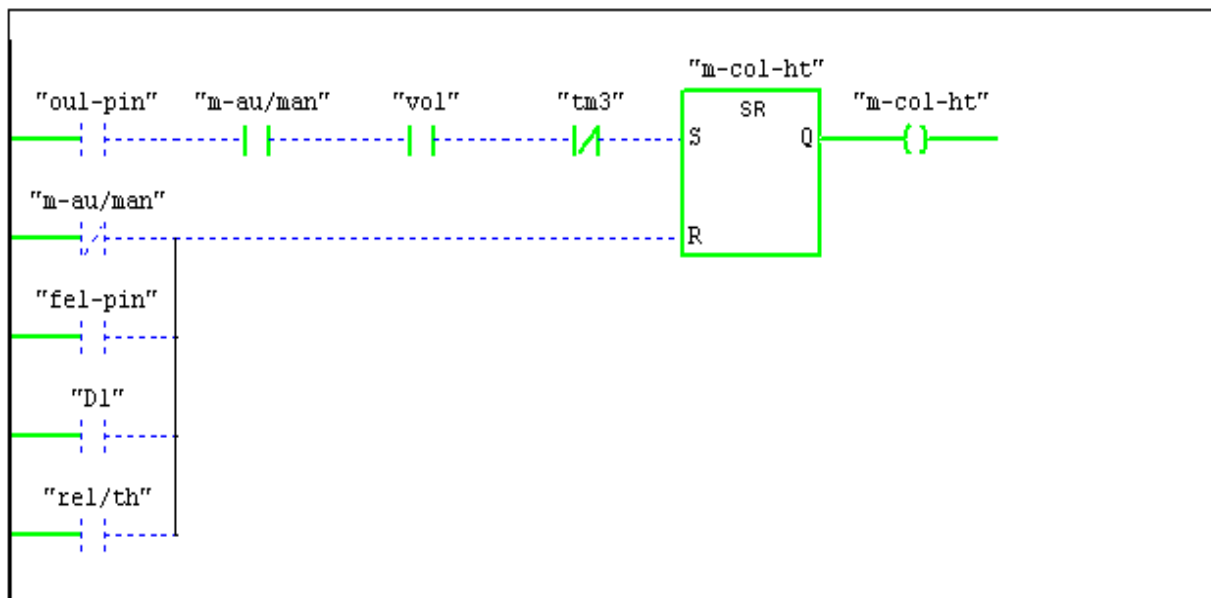
Réseau 12: ouverture de la pince

Commentaire :



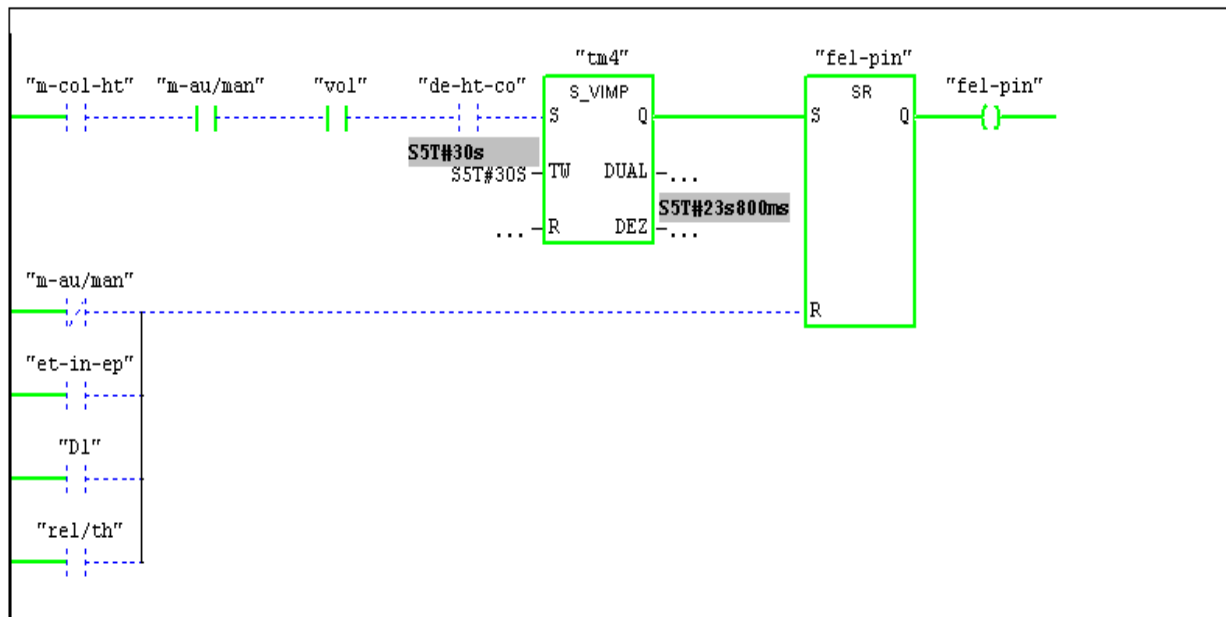
Réseau 13: mouvement de coude vers le haut

Commentaire :

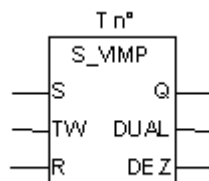


Réseau 14 : fermeture de la pince

Commentaire :



❖ Remarque :



S_VIMP (Paramétrer et démarrer temporisation sous forme d'impulsion prolongée)

Cette opération démarre la temporisation précisée en cas de front montant à l'entrée de démarrage S. Un changement d'état de signal est toujours nécessaire pour activer une temporisation. La valeur de temps indiquée à l'entrée TW continue à s'écouler même si l'état de signal à l'entrée S passe à 0 avant expiration du temps. Tant que la temporisation s'exécute, l'état de signal à la sortie Q égale 1. La temporisation est redémarrée avec la valeur de temps prédéfinie si l'état de signal à l'entrée S passe de 0 à 1 alors que la temporisation s'exécute.

En cas de passage de 0 à 1 à l'entrée de remise à zéro R pendant que la temporisation s'exécute, cette dernière est remise à zéro. La valeur de temps en cours et la base de temps sont alors également mises à 0.

La valeur de temps en cours peut être lue en format binaire à la sortie DUAL et en format décimal codé binaire à la sortie DEZ. La valeur de temps en cours correspond à la valeur initiale en TW moins la valeur de temps écoulée depuis le démarrage de la temporisation.

V.12. Le programme de Commande manuel d'un bras manipulateur :

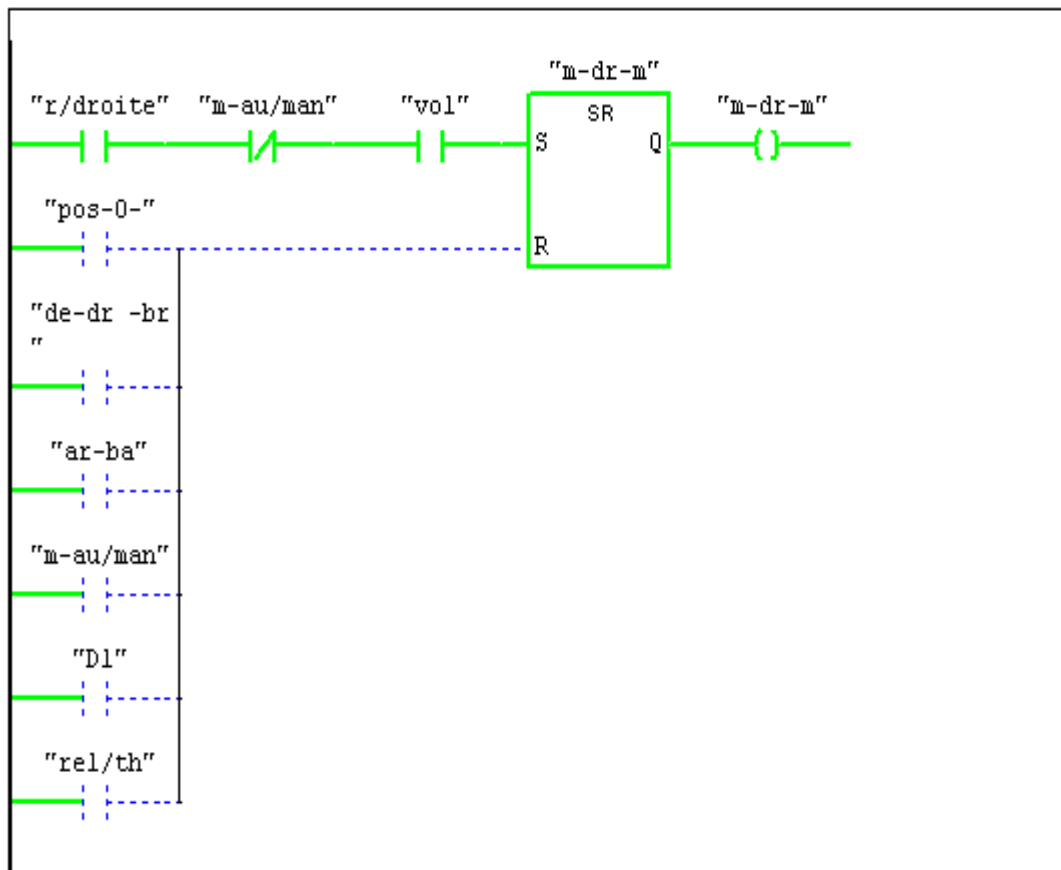
Le bloc FC2 représente le programme de commande manuel :

FC2 : COMMANDE MANUELLE DU BRAS

Commentaire :

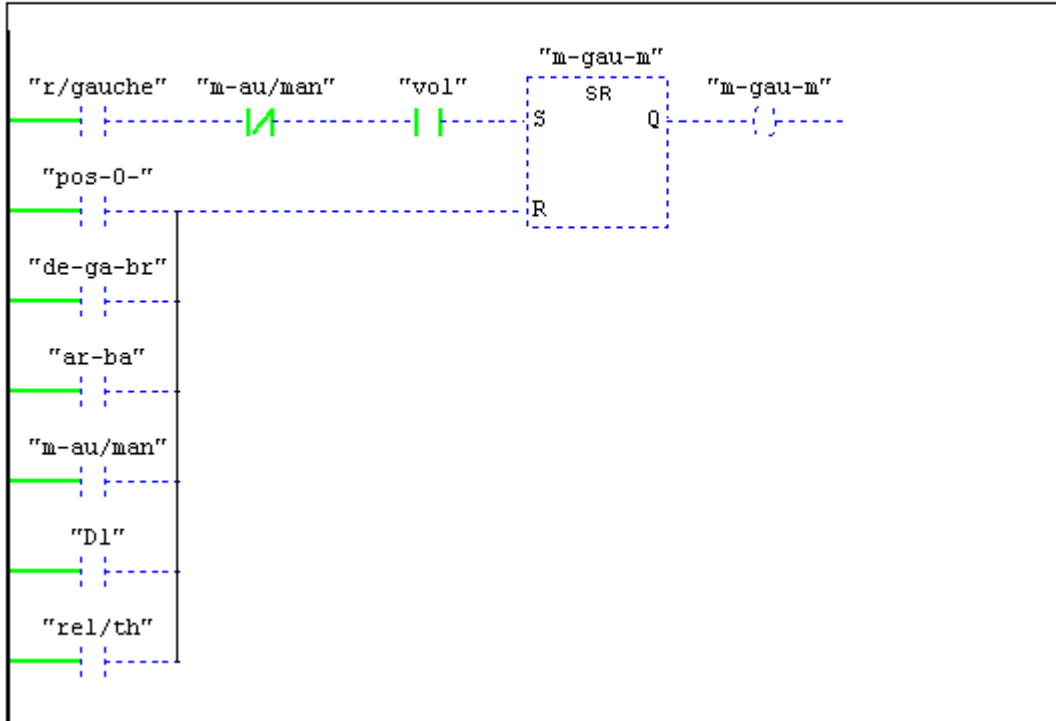
Réseau 1: le bras marche vers la droite

Commentaire :



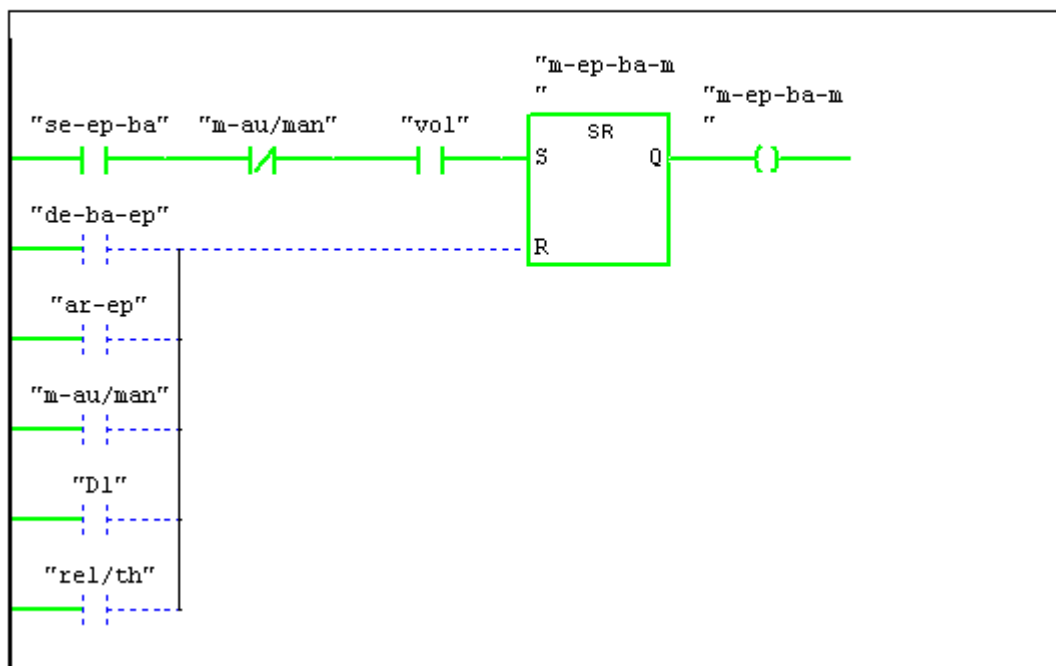
Réseau 2: le bras marche vers la gauche

Commentaire :



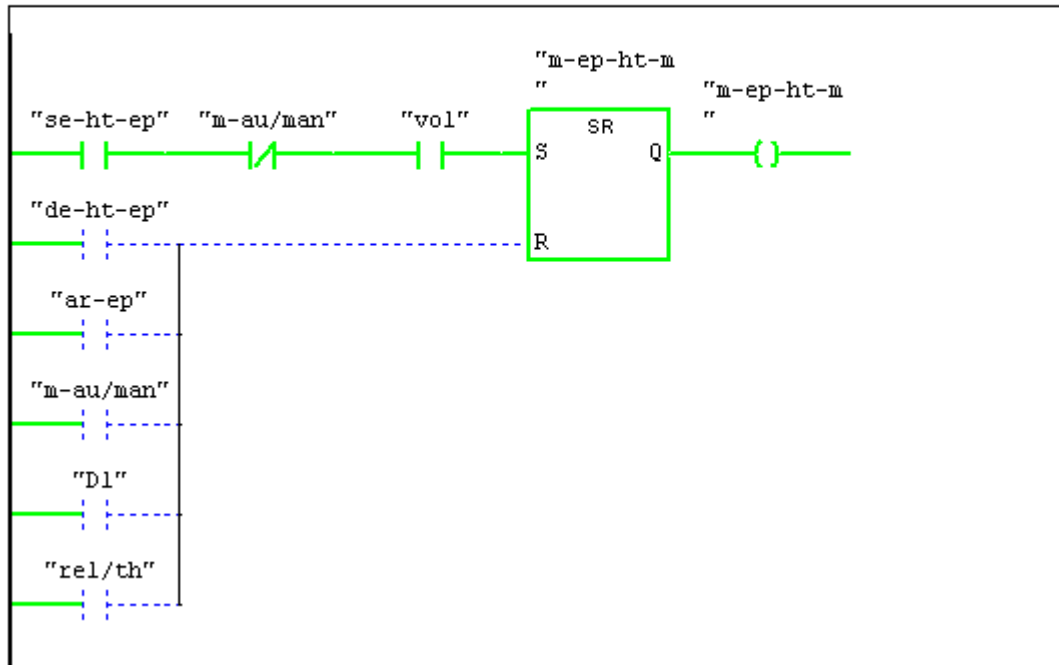
Réseau 3: mouvement d'épaule vers la bas

Commentaire :



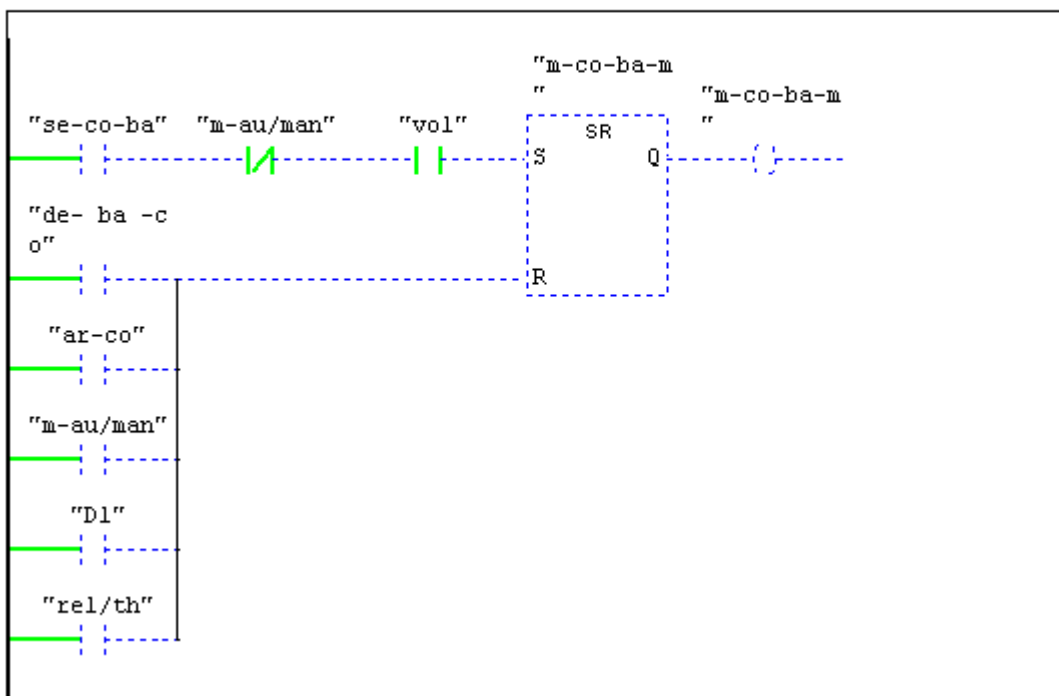
Réseau 4 : mouvement d'épaule vers le haut

Commentaire :



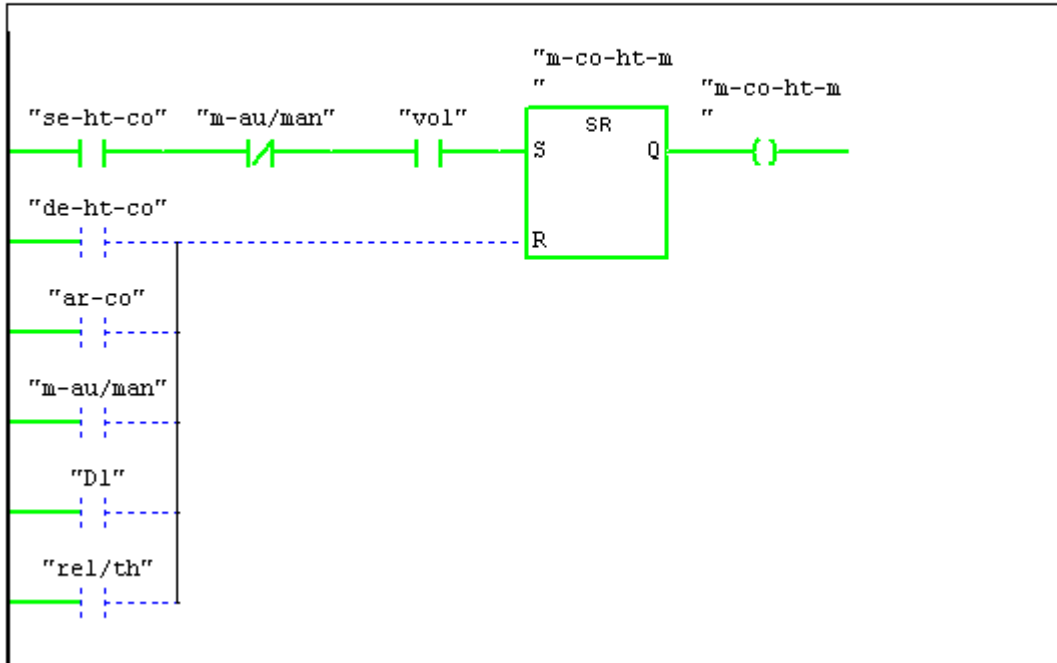
Réseau 5 : mouvement de coude vers la bas

Commentaire :



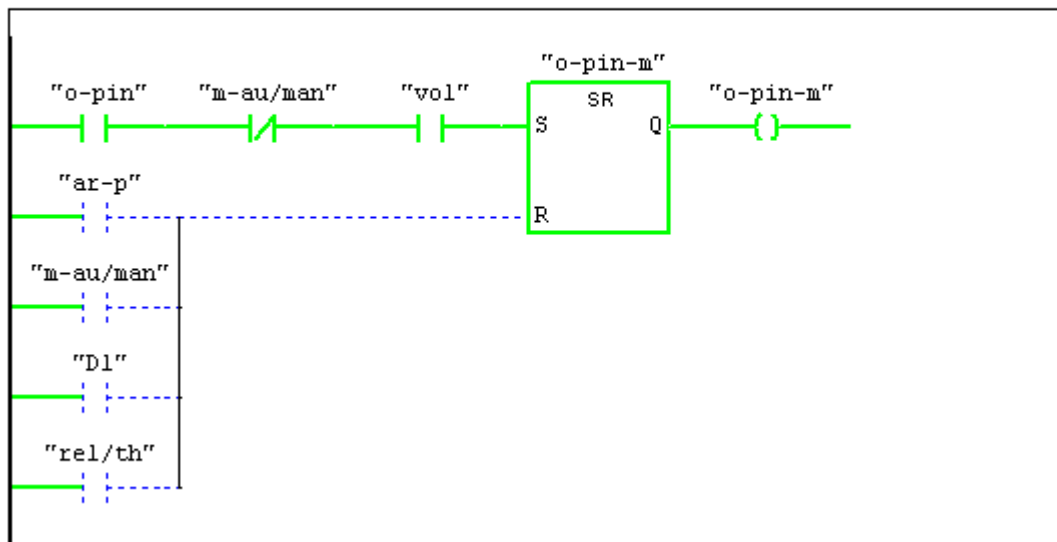
Réseau 6: mouvement de coude vers le haut

Commentaire :



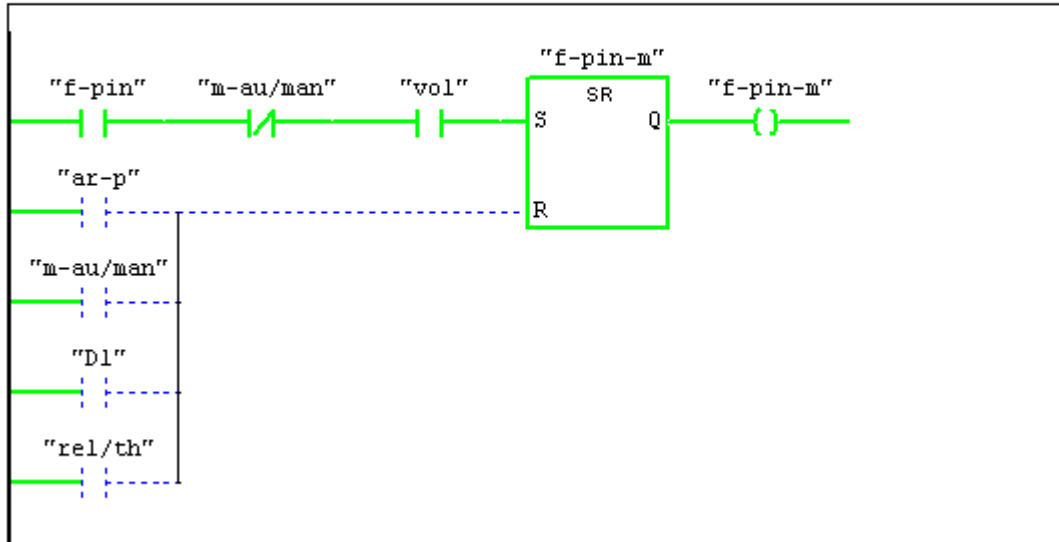
Réseau 7: ouverture de la pince

Commentaire :



Réseau 8: fermeture de la pince

Commentaire :

**V.13. Conclusion :**

Dans ce chapitre nous avons exposé une étude de programmation et de simulation d'une commande automatique et manuelle d'un bras manipulateur à l'aide du logiciel STEP7, nous sommes préoccupés plus particulièrement et baser sur deux parties principales. la première partie elle concerne l'étude Théorique des différentes composantes de bras manipulateur (mécanique et électrique) et la deuxième partie concerne la programmation du bras manipulateur.

Conclusion générale:

L'étude proposée à pour suit d'expliquer l'intérêt et l'importance de l'utilisation de l'API dans le domaines de l'industrie (passage de la logique câblée à logique programmer) d'une part et d'autre part on utilisent cette séquence pour la commande d'un bras manipulateur donc on a partager notre travail en cinq chapitres:

Dans le premier chapitre on vue l'importance d'un système automatiser et leur composante ainsi que leurs avantages et Inconvénients ... etc.

Ensuite, dans le deuxième chapitre, l'étude théorique sur les API a été établie. Nous avons décrit leurs architectures comme : le CPU et les modules d'entrées/sorties ... en terme matériel ; l'interface de programmation ... en terme logiciel.

Dans le troisième chapitre, nous avons présenté une étude assez détaillée sur le SIMATIC S7-314 IFM de SIEMENS, qui a été notre choix pour commander la bras manipulateur.

Quant au quatrième chapitre on a vue une description détaillée de logiciel STEP7, et les applications qui la concerne Et aussi les langues de programmation qu' il contient comme contacts et logigramme, et comment créer un projet dans le logiciel STEP7, et la méthode de simulation d'un projet.

Finalement dans le cinquième chapitre nous proposons une étude complète de programmation et simulation pour commander un bras manipulateur automatique et manuel par l'automate programmable S7 300 (S7 314IFM) à l'aide du logiciel STEP7.

Annexes

Liste des symboles

TOR	: Tout ou – rien
CPU (Central processing unit)	: Unité centrale
Profibus (Process Field Bus)	: Processeur de communication
HMI (Humann machine interfaces)	: dialogue homme machine
PS (power supply)	: module d'alimentation
A4.0	: Adresse de sortie
E1.1	: Adresse d'Entrée
PG	: console de programmation
T4	: symbole de temporisation
API	: Automate Programmable industriel
CONT	: Le langage a base de schemas de contacts
DI	: Entree TOR
DO	: Sortie TOR
EEPROM	: Electrically Erasable Programmable Read Only Memory
FB	: Bloc de fonction
FC	: Fonction
LOG	: Le langage a base de logigramme
LIST	: Le langage de liste d'instructions
OB	: Bloc d'organisation
PROM	: Programmable Read Only Memory
PS	: Gamme des alimentations stabilisees de Siemens
PG	: La console de programmation sur le terrain
SIMATIC	: Siemens Automatic
SM	: Gamme des modules E/S des automates de Siemens
RAM	: Random Access Memory
ROM	: Programmable Read Only Memory

Bibliographie

[1]. G.MICHEL:

Les A.P.I. Architecture et applications des automates programmables industriels, 1988.

[2]. Télémécanique:

SCHEMATIQUE ELECTROTECHNIQUE, 1986.

[3]. M.AUMIAUX, G.RODDE:

Automatiser la production, 1988

[4]. F.CASTELLAZZI, Y.GANGLOFF, D.COIGNIEL:

MEMOTECH, Maintenance industrielle, CASTELLA, 1998.

[5]. HENRI NEY :

Technologie et schéma d'électricité, NATHAN TECHNIQUE, 1980.

[6]. J.C.BOSSY, D.MERAT:

AUTOMATISME APPLIQUE, 1985.

[7]. Alain REILLER:

Analyse et maintenance des automatismes industriels, ellipses, 1999.

[8]. YVES G. PALAU

l'automate programmable , s cahiers de la technologie 1983 .

Les automates programmables industriels et les réseaux 2000.

[9]. GUILLOSOU BERNARD

Technique numérique (série 1).

[10]. SONATRACH, CENTRE DE PERFECTIONNEMENT DE L'ENTRPRISE (A.AMRI)

[11]. A. SIMON, « Automate Programmable Industriel, Niveau 1 », Edition L'ELAN, 1991.

[12]. MICHEL BERTRAND

«Automate programmable industrielle » Docteur-Ingénieur. École Nationale Supérieure d'Arts et Métiers ENSAM, Centre d'Enseignement et de Recherche de Lille.

[13]. BENARIBAMMAR. BELMILOUD ABD ELAZIZ

« Étude de réalisation d'un API a bas d'un microcontrôleur » Mémoire d'ingénieur, université de M'sila 2006 (fichier PDF).

[14]. "langage de programmation pour API , norme IEC1131 -3"

Technique de l'ingénieur vol. S8.030.

[15]. électrotechnique THEDOR WILDI , 2000

[16]. N.BOUAYED" télégestion " , projet de fin d'étude ,école nationale polytechnique

Manuel :

[17]. SIEMENS "STEP7 ,Getting stard " SIMATIC 2002

[18]. SIEMENS " S7 PLCSIM, testez vos programme " SIMATIC 2002

[19]. SIEMENS " programmer avec STEP7 " SIMATIC 2000

[20]. Help de STEP7

[21]. BROCHURE DE COURS STEP7 SIEMENS

[22] . SIMATIC- automate programmable, S7300

Installation et configuration –caractéristique de CPU.

Internet :

[23] WWW. SIEMENS . COM (photo)

MEMOIRE DE FIN D'ETUDES EN VUE DE L'OBTENTION DU
DIPLOME
D'INGENIEUR D'ETAT EN GENIE ELECTROTECHNIQUE

OPTION: Commande Electrique

Proposé et dirigé par : - *Monsieur: S.BELKHIRI*

Présenté par : *TELLI Abderrahim*
KHODJA Choayb

Thème :

**COMMANDE D'UN BRAS MANIPULATEUR
PAR AUTOMATE PROGRAMMABLE
INDUSTRIEL
(A.P.I)**

Résumé :

Ce projet de fin d'étude présente une étude de programmation et simulation d'un bras manipulateur à trois degrés de liberté entraîné par quatre moteurs à courant continu et commandé par un automate programmable industriel (API) S7 300 de SIEMENS, à l'aide du logiciel de programmation STEP7.

Mots Clés:

API, bras manipulateur, STEP7, langage, commande, simulation