PDF-XCHANGE Click to buy NOW! www.docu-track.com

الجمهورية الجزائرية الديمقراطية الشعبية

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR
ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITE DR. YAHIA FARES DE MEDEA

FACULTE DES SCIENCES ET DE LA TECHNOLOGIE

Département du Génie Électrique et de l'Informatique

Projet de Fin d'Études

Présenté pour l'obtention du Diplôme
de Master en Génie Électrique

Spécialité : Automatique des Systèmes

THEME

Réalisation de la commande et d'une interface Homme-machine à base d'un automate programmable S7-300 de SIEMENS : application à une machine profileuse

Proposé et dirigé par,

Dr. Mounir BOUHEDDA

Préparé et présenté par,

Mr.BENZIANE Amine

Mr.BERKANE Yahia

PDF-XCHANGE
Click to buy NOW!
www.docu-track.com
PDF-XCHANGE
Click to buy NOW!
www.docu-track.com

Remerciement

Avant tous nous tenons à remercier " Allah ".

C'est un devoir bien agréable que de venir rendre hommage, au terme de ce travail, à ceux sans lesquels il n'aurait pas pu être réalisé. L'occasion nous est offerte de remercier ici, les personnes qui nous feront l'honneur de participer au jury chargé de juger ce travail.

Nous tenons tout particulierement à exprimer notre profonde gratitude à notre encadreur ***Dr Mounir BOUHEDDA*** *pour son aide précieuse, son soutien et son professionnalisme. Veuillez Mr agréer l'expression de nos sentiments les plus respectueux.*

Nous adressons aussi notre plus vive reconnaissance à tous nos enseignants de département de génie électrique pour la formation qu'ils nous ont donné pendant les cycles universitaires.

Nous ne manquerons pas de saluer tous ceux qui ont de près ou de loin participé à l'accomplissement de ce travail.

Yahia & Amine

PDF-XCHANGE
Click to buy NOW!
www.docu-track.com

Dédicace

Je m'incline devant Dieu tous qui m'ouvert port du savoir m'aidé à la franchir

Je dédie mon travail :

A ceux qui ont passé des nuits trop longues pour mon bonheur, et qui se sont fatigués pour mon repose M a mère et M on père

A tous mes frères et ma sœur

A tous mes familles

A mon cher binôme BER K ANE Yahia.

A tous les étudiants de M ASTER 2 PROMOTION 2012-2013

A tous mes cher amis de prés au de loin sans exception.

BENZIANE Amine

PDF-XCHANGE
Click to buy NOW!
www.docu-track.com
PDF-XCHANGE
Click to buy NOW!
www.docu-track.com

Dédicaces

Au nom d'Allah le Miséricordieux le très miséricordieux.

J'ai l'immense honneur de dédier ce mémoire :

A mes très chers parents qui étaient présents pour moi durant toute ma vie.

A mes frères et m& sœurs.

A toute ma famille.

A mon cher Binôme BENZIANE Amine.

A tous les professeurs et enseignants qui ont collaboré à ma formation depuis

Mon premier cycle d'étude jusqu'à la fin de mes études universitaires.

ADr.BOUHEDDA Mounir.

A tous ceux qui m'aiment et qui ont cru en moi !

A tous mes amis.

Berkane Yahia

PDF-XCHANGE Click to buy NOW! www.docu-track.com

PDF-XCHANGE Click to buy NOW! www.docu-track.com

TABLE DES MATIERES

Introduction générale……..…..…i

Chapitre I : Etude théorique sur les machines profileuses.

I- Etude théorique sur les machines profileuses…….....1

I-1 introduction……..……..1

I-2 les machines profileuses……..……..…….1

I-2-1 définition……1

I-3 Le profilage de tôles……..……2

I-3-1 Description……..……2

I-4 Le fonctionnement du profilage……3

I-5 Les capacités du profilage……..…4

I-6 Les règles de base……..5

I-6-1 Les déformations transversales……...……...5

I-6-2 La vitesse de la tôle et des outils……6

I-6-3 Les déformations longitudinales……6

I-7 Equipement d'appareillage…….……..…….....7

I-7-1 Dérouleurs……..……7

I-7-2 Outils de perforations et de découpage……...……..7

I-8 conclusion ……...……..8

Chapitre II : les automates programmables industriels et environnement de programmation de l'automate S7-300 de SIEMENS.

II- Introduction……...…….....9

II-1 Historique.....……..……....……...9

II.2 Définition générale .....….…….…..…….....10

II.3 Architecture des automates programmables industriels…….……...……11

II.4 Structure interne des automates programmables ……...……12

II.4.1 Le processeur…….……...…….....13

II.4.2 Les modules d'entrées/sorties ……...……...……...…..13

II.4.3 Les mémoires .....…….……..…….....13

II.4.4 L'alimentation …….....……14

II.4.5 Liaisons de communication ……..……...14

II.5 Présentation de la gamme SIMATIC de SIEMENS ……...……..……14

II.5.1 Les différentes variantes dans la gamme SIMATIC ……..…….15

PDF-XCHANGE Click to buy NOW! www.docu-track.com

TABLE DES MATIERES

a) SIMATIC S7 15
1) S7 200.....15
2) S7 300 15
3) S7 400 16
b) SIMATIC C7 16
c) SIMATIC M7 16
II.6 Description du STEP7 17
II.6.1 Création d'un projet STEP7 pas à pas 18
II.6.2 Configuration du matériel 21
II.6.3 Définition des mnémoniques 23
II.6.4 Edition des programmes 25
II.6.4.1 Blocs d'organisation 27
II.6.4.2 Fonctions et blocs fonctionnels 29
II.6.4.3 Bloc de données 29
II.6.5 Programmation des blocs 30
II.6.5.1 CONT LIST LOG 30
II.6.5.2 SCL 32
II.6.5.3 S7-GRAPH 33
II.6.5.4 avantage de l'utilisation du S7-GRAPH.....36
II.6.6 Simulation de modules 37
II.7 WIN CC Flexible 41
II.7.1 Introduction 41
II.7.2 Progiciel de conception et configuration d'interface Win CC flexible 42
II.7.2.1 Présentation générale.....42
II.7.2.2 Elément du progiciel Win CC flexible 43
II.7.2.3 Utilisation des variables 45
II.7.2.4 Compilation et Simulation 46
II.7.3 Conclusion.....46

Chapitre III : réalisation de la commande et de l'interface homme-machine

III. introduction 47
III.1 le mécanisme de la machine 47
III.1.1 Sous système (1) dérouleur 48
III.1.2 Sous-système (2) station de perforation 49

PDF-XCHANGE Click to buy NOW! www.docu-track.com

III.1.3 Sous-système (3) défileuse 49
III.1.4 Sous-système (4) station de découpage 50
III.2 Partie Automatisme.....51
III.2.1 Réalisation du GRAFCET de la machine profileuse.....51
III.2.2 Configuration des paramètres du S7-300 51
III.2.3 Table des mnémoniques 52
III.2.3.1 Table des mnémoniques de la machine profileuse.....53
III.2.4 Dérouleur 54
III.2.4.1 GRAFCET de fonctionnement 54
III.2.5 Fonctionnement du comparateur 55
III.2.6 Fonction intégrée compteur S1 CU Z1.....56
III.3 L'interface homme machine HMI configurée par le progiciel « Win CC flexible » 58
III.3.1 Création de la station HMI 59
III.3.2 Création de Table des variables 59
III.3.3 Création des vues de projet 60
III.3.3.1 La vue initiale 60
III.3.3.2 La Vue « opérateur panel » 61
III.3.3.3 La vue « animation » 62
III.3.3.4 La vue « alarmes » 63
III.3.4 Etablissement de la liaison Automate-pupitre opérateur TP177B.....64
III.3.5 Transfert du projet sur le pupitre 65
III.3.5.1 Paramétrage de transfert.....65
III.4 conclusion ……..……....66

Annexe

A.1. La CPU 313C ……..……..……67
A.2. Le terminal d'exploitation TP177B 6" mono DP ……..……..……...68
A.3. Configuration matérielle (Partie Hardware)……..……..……70
A.4. Transfert sur les pupitres opérateur TP 177B, TP 177A ……..……..……...70
A.5. Chargement dans le système cible à partir de la PG/PC ……..……..……....71
A.6. Le choix de la vitesse de transmission ……..……..……...72

PDF-XCHANGE Click to buy NOW! www.docu-track.com

Liste des figures

Chapitre I

Figure I-1 : Prototype de la profileuse et ces sous-systèmes ……..…….……2
Figure I-2: Fleur de profilage .....…….…….....……3
Figure I-3 : le fonctionnement du profilage ……...……...……...4
Figure I-4 : sous-système existant sur une profileuse …….....7

Chapitre II

Figure II-1 : l'automate programmable siemens …….....……12
Figure II-2 : structure interne d'un API [8] ……...……..12
Figure II-3 : Structure Présentation de la gamme de SIMATIC [8].....……..14
Figure II-4 : L'API S200 [14].....……...……..15
Figure II-5 : L'API S300 [14].....…….……..15
Figure II-6 : L'API S400 [14].....……..……..16
Figure II-7 : La gamme SIMATIC C7 [14]……..……16
Figure II-8 : La gamme SIMATIC M7 [24].....……..……17
Figure II-9 : Création d'un nouveau projet .....……..19
Figure II-10 : Station PC et PG/PC ……...20
Figure II-11 : Choix de la station de travail .....…….....20
Figure II-12 : Configuration du matériel .....…….……..21
Figure I-13: sélection des modules .....……..……22
Figure II-14 : Edition des mnémoniques.....……..……..…….23
Figure II-15 : Edition des programmes .....……...…….26
Figure II-16 : CONT LIST LOG .....…….31
Figure II-17 : exemple d'un programme en **CONT** .....…….....32
Figure II-18 : exemple d'un programme en **SCL** .....…….... .33
Figure II-19 : Commande d'un chariot .....……34
Figure II-20 : Edition d'un bloc fonctionnel avec **S7-GRAPH** ……....34
Figure II-21 : Edition du **GRAPH** .....……...35
Figure II-22 : Edition des actions & transitions ……...36
Figure II-23: Simulation de modules …….. 37
Figure II-24 : Création d'une fonction ……... 39
Figure II-25 : Définition des arguments d'une fonction ……..... 39
Figure II-26 : Edition du code en LIST ……..…..40
Figure II-27 : Edition du code CONT .....…….. .40

PDF-XCHANGE Click to buy NOW! www.docu-track.com

Figure II-28 : Simulation di programme ……..…..…..41
Figure II-29 : L'interface Homme/Machine dans un processus automatisé ..…….....42
Figure II-30 : Etapes pour la conception d'une interface via Win CC flexible .....…….....43
Figure II-31 : Vue d'ensemble du progiciel WinCC flexible …….....45
Figure II-32 : éditeur de variable dans le WinCC flexible …….....46

Chapitre III

Figure III.1 : Prototype de la profileuse et ces sous-systèmes. …….....48
Figure III.2 : Dérouleur…….....48
Figure III.3 : station de perforation…….....49
figure III.4 : défileuse…….....50
Figure III.5 : station de découpage**.** …….....50
Figure III.6 : configuration du matériel…….....51
figure III.7 : table des mnémonique…….....53
figure III.8 : étape S6 du bloc FB1…….....54
figure III.9 : étape initiale S1 du bloc FB1…….....54
figure III.10 : étape S2 du bloc FB1…….....55
figure III.11 : étape S3 du bloc FB1…….....55
figure III.12 : étape S4 du bloc FB1…….....56
figure III.13 : étape S5 du bloc FB1…….....57
figure III.14 : étape S8 du bloc FB1…….....57
figure III.15 : étape S11 du bloc FB1…….....58
Figure III.16 : Le pupitre opérateur TP177B MONO DP…….....59
Figure III.17 : l'automate programmable S7-300 avec le pupitre opérateur ……59
Figure III.18 : table des variables WINCC……..... 60
Figure III.19 : vue initiale…….....61
Figure III.20 : vue opérateur panel…….....61
Figure III.21 : exemple sur la défileuse ……..62
Figure III.22 : vue animation…….....62
Figure III.23 : Création des variables pour les larmes……..63
Figure III.24 : exemple d'une vue alarme…….....64
Figure III.25 : Configuration de la liaison Panel-Automate…….....65
Figure III.26 : paramétrage du transfère ……..... 65

PDF-XCHANGE Click to buy NOW! www.docu-track.com

Liste des figures

PDF-XCHANGE Click to buy NOW! www.docu-track.com

Annexe

Figure 1 : le pupitre opérateur TP177B 6“ mono DP……68

Figure 2 : vue arrière du pupitre……69

Figure 3 : Configuration matérielle……70

Figure 4 : choix de PC Adapter (MPI)……....71

Figure 5 : choix de la vitesse de transmission……72

PDF-XCHANGE
Click to buy NOW!
www.docu-track.com
PDF-XCHANGE
Click to buy NOW!
www.docu-track.com

Liste des tableaux

Tableau II.1 : les zones mémoires…….25

Tableau II.2 : les actions dans le S7-GRAPH……..36

Tableau A.1 : Positions du commutateur du mode de fonctionnement ……67

Tableau A.2 : tableau de contrôleurs de SIEMENS…….…69

PDF-XCHANGE Click to buy NOW! www.docu-track.com

PDF-XCHANGE Click to buy NOW! www.docu-track.com

I Etude théorique sur les machines profileuses

I.1Introduction

Le développement massif des techniques de l'automatisme a permis le passage de la machine automatisée à celui des systèmes automatisés de production, qui gèrent l'alimentation en énergie et qui permettent d'avoir une meilleur qualité des produits en plus de la sécurité et de la flexibilité des processus, mais cela entraîne un accroissement des besoins, en particulier la manipulation d'un grand nombre de variables et la gestion de véritables flux de communication.

Dans ce chapitre on présentera une description générale sur les machines profileuses afin de comprendre le fonctionnement de chaque sous-système de ces machines, ces détailles permettent de proposer une commande à base d'un automate programmable de ces machines.

I.2 Les machines profileuses

I.2.1 Définitions

Les machines profileuse(s) servent à profiler les tôles (feuillards), feuillard acier, feuillard aluminium, feuillard aluminium, feuillard inox, donc à réaliser les divers types de profils. A plier le feuillard qui passe dans les diverses têtes de la machine à profiler . Pour obtenir la forme du profil souhaitée. La matière peut être de l'acier, de l'inox, de l'aluminium, de l'alu, du cuivre, du titane, ...

La machine se compose d'un dérouleur simple ou bien automatisé, d'une table d'entrée pour le feuillard, de la profileuse, des têtes de profilage (axe supérieur, axe inférieur, support des galets), du dévriller, boite de sortie, table de récupération du profil. ainsi que d'une armoire électrique de la profileuse, commande motorisation, avance de la ligne de profilage suivant les modèles des profileuses et de leur utilisation.

PDF-XCHANGE Click to buy NOW! www.docu-track.com

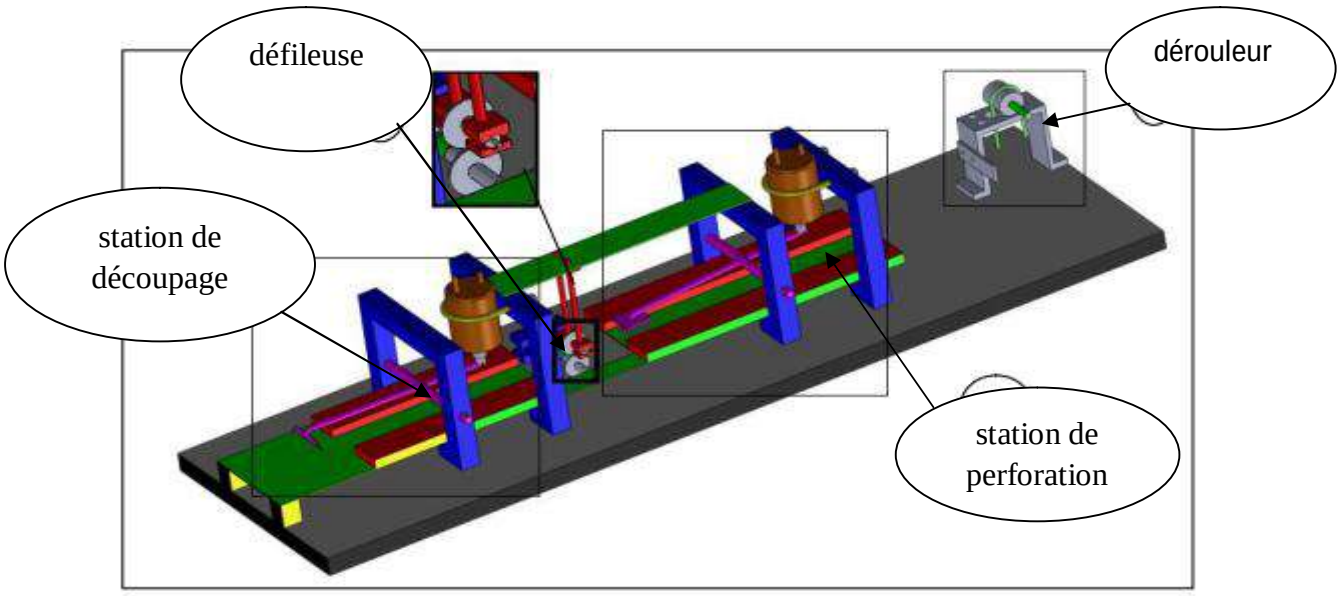


Figure I.1 : Prototype de la profileuse et ces sous-systèmes.

I .3 Le profilage de tôles

I .3 .1 Description

Le profilage à froid est un processus de mise en forme en continu qui permet, à partir de tôles planes en feuilles ou en bobine, d'obtenir des produits de section constante, appelés profilés **[3]**.

Ces profilés peuvent être divisés en trois grandes familles:

• les profilés larges: ils s'agit de produits de grande largeur (600 à 1500 mm environ) comportant plusieurs ondes de forme équivalente ou similaire. On rencontre notamment dans cette catégorie les profilés de couverture.

• les tubes: ils s'agit de produits à section fermé réalisés par profilage avec une soudure en continu;

• les profilés étroits: il s'agit de produits de section ouverte ou fermée sans soudure ne comportant pas de répétition d'une forme d'onde. Cette catégorie comprend, entre autres, les profilés de structure, les chemins de câbles, les glissières de sécurité, ect.



Figure I .2 Fleur de profilage.

I.4 Le fonctionnement du profilage

Le profilage peut être considéré comme un pliage chacune en continu. Au cours du processus, la tôle est pliée plastiquement et progressivement par chacune des têtes, jusqu'à obtenir les angles et la forme souhaités. Chaque jeu de galets (supérieur et inférieur) appelé tête de profilage. La forme que prend la tôle d'acier revêtu métallique à chacune des têtes ou passes de profilage constitue la fleur de profilage **[2]**. Il s'agit d'une caractéristique majeure, car elle définit l'ordre de réalisation des plis ainsi que les valeurs de pliage des angles. Plus généralement, le profilage d'un produit sur une machine définie est essentiellement caractérisé par:

- la fleur de profilage.
- le nombre de passes ou de têtes utilisées (cette valeur est définie indirectement par la fleur de profilage.
- la distance entre les têtes de la profileuse.
- la position du plan de profilage. C'est à dire la position verticale de la tôle dans la profileuse.
- le jeu entre les galets.
- la lubrification.

PDF-XCHANGE Click to buy NOW! www.docu-track.com

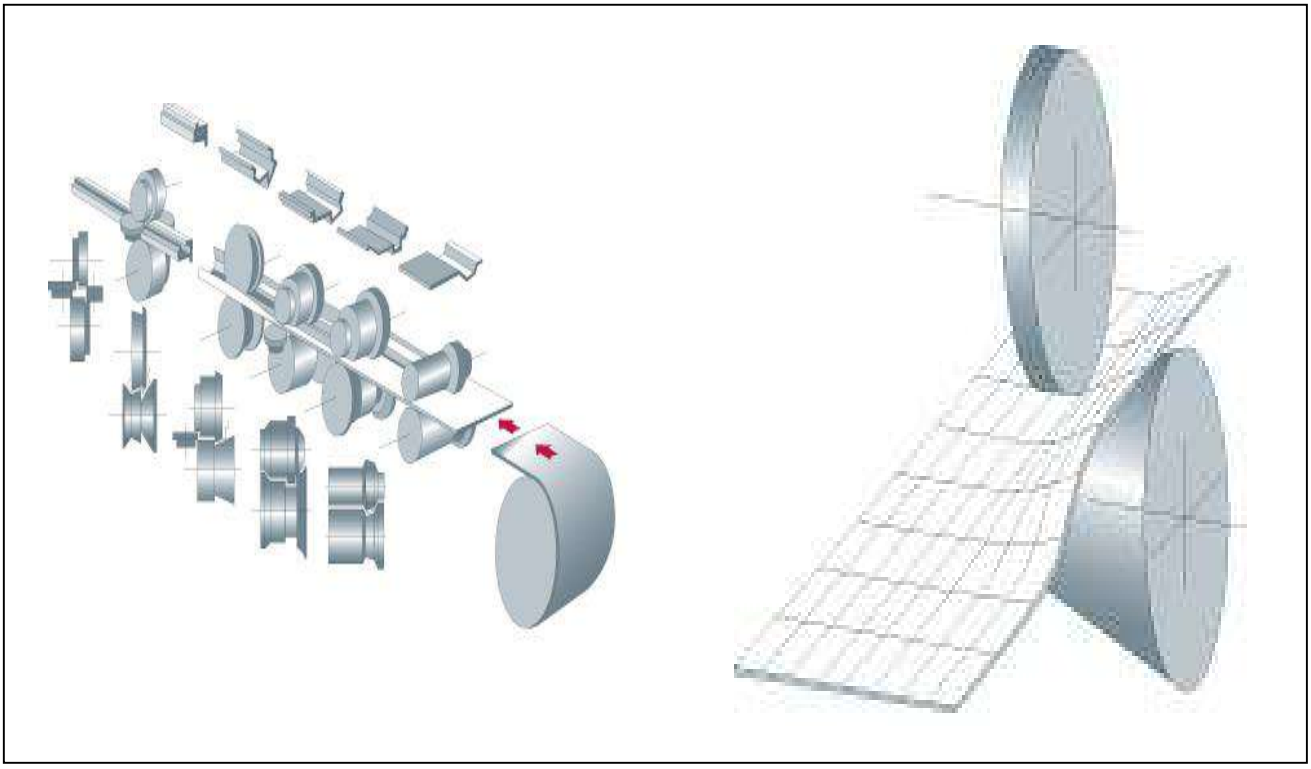


Figure I .3 le fonctionnement du profilage.

I.5 Les capacités du profilage

Le profilage est un processus très souple. Ceci se retrouve dans ses capacités de production, dans les formes réalisables - même si elles doivent rester de section constante dans le sens de la longueur - mais aussi au niveau:

- **des épaisseurs:** elles peuvent varier entre 0,15 mm et 12 mm ;
- **des largeurs :** il n'y a pas de valeur minimale du fait des capacités de refendage des bobines. Pour la valeur maximale, la borne est en réalité due au format réalisable des bobines, c'est-à-dire 2060 mm selon notre programme actuel ;
- **des longueurs :** il n'existe pas de maximum pour la longueur des profilés si ce n'est celui qui existe pour la manutention, le transport, etc. des produits ;
- **des nuances d'acier :** les nuances très ductiles avec une faible limite d'élasticité (aux environs de 150 MPa) sont utilisables ainsi que celles très dures avec limites d'élasticité de 1000 MPa et plus. Avec une technologie adaptée, il est tout à fait possible de réaliser des profilés de qualité dans toutes les nuances existantes :

PDF-XCHANGE Click to buy NOW! www.docu-track.com

Du type du produit de base : le profilage est relativement sensible au type du produit de base : laminé à chaud, laminé à froid, galvanisé, ect. Certes, les mélanges sont fortement déconseillés.

L'encrassement des outils étant inévitable, il faut faire attention à ne pas redéposer des salissures d'un produit vers l'autre.

La plus grande restriction du profilage reste le fait que le produit doit être de forme constante dans sa longueur - ce qui n'exclut pas les produits courbes.

I.6 Les règles de base

Pour obtenir des profilés de qualité en sortie de profileuse, il est nécessaire de respecter quelques règles de base. Elles sont liées aux différents phénomènes intervenant dans le processus.

I.6.1 Les déformations transversales

Pour obtenir un profilé à partir d'une tôle plane, il set nécessaire de créer des plis et donc de plastifier localement la tôle dans son sens transversal. Le niveau de ces déformations plastiques est principalement lié à la géométrie du produit souhaité. Elles peuvent de ce fait être approximées selon les règles suivantes :

Règle n°1

Pour éviter une fissuration du revêtement et une corrosion du profilé, lors de son utilisation future, choisir une géométrie de celui-ci avec un rapport e/R le plus faible possible : avec des rayons importants par rapport à l'épaisseur du matériau **[1]**.

Règle n°2

Ne pas dépasser la limite d'élasticité du matériau pour les déformations longitudinales. Jouer pour cela sur la fleur de profilage voire la conception du profilé. Augmenter la limite d'élasticité du matériau si nécessaire **[1]**.

PDF-XCHANGE Click to buy NOW! www.docu-track.com

Règle n°3

Pour répartir les déformations longitudinales sur la plus grande zone possible, choisir un diamètre des galets le plus grand autorisé ainsi qu'une distance entre deux têtes suffisante (supérieure à 1,5 fois la longueur de déformation pour le profilage étroit) **[1]**.

I.6.2 La vitesse de la tôle et des outils

En premier lieu, il ne faut jamais oublier que la tôle progresse uniquement grâce aux frottements sur les outils dans la profileuse. Si le frottement devenait nul, le profilage ne fonctionnerait plus.

La première difficulté est de faire prendre à la tôle une vitesse constante tout au long de la profileuse, voire légèrement progressive. Cela évite l'apparition de cloque ou inversement une trop grande traction.

La seconde difficulté est d'avoir un frottement de la tôle sur les outils sans glissement relatif. Si en un endroit de la profileuse, un outil patine sur la tôle, il y a à ce moment-là un risque important de marquage.

Règle n°4

Faire prendre à la tôle une vitesse constante, voire légèrement progressive. Pour cela choisir un seul diamètre identique sur l'ensemble des galets moteurs et monter les autres galets sur des roulements **[1]**.

Règle n°5

Pour diminuer le retour élastique, concevoir un profilé avec de petits rayons, une épaisseur importante et une qualité à faible limite d'élasticité. Pour ne pas être en contradiction avec la règle n°1, il est parfois possible de dépasser la valeur nominale de l'angle du pli à obtenir pour contrôler ce phénomène (utilisation d'une tête de «sur-formage») **[1]**.

I.5.3 Les déformations longitudinales

La principale difficulté dans le profilage est de gérer correctement les zones vrillées de la tôle dans la profileuse. Celles-ci sont intrinsèques au processus. Il ne s'agit pas de les éliminer - ce

PDF-XCHANGE Click to buy NOW! www.docu-track.com

qui est totalement impossible - mais de les limiter voire dans certains cas de les répartir correctement. Il faut que les déformations longitudinales engendrées par le vrillage de la tôle dans la profileuse restent dans le domaine élastique du matériau. Si ce n'est pas le cas, le matériau va se plastifier.

La tôle va localement s'allonger de façon irréversible dans son sens longitudinal et se vriller à la sortie de la profileuse.

I.7 Equipement d'appareillage

Une profileuse est composée de plusieurs sous-systèmes :

I.7.1 Dérouleurs : Ils servent à emmener la tôle (feuillard) a la première tête de la profileuse pour la réalisation des profiles dans la ligne de profilage .ll est utilisé pour des charges légères ou bien pour des charges lourdes, avec motorisation. selon le poids des bobines de feuillard. Ces rouleaux de feuillards peuvent être en acier, aluminium, inox, titane, cuivre, laiton, etc.

I.7.2 Outils de perforations et de découpage : Les outils de coupes sont utilisés en complément sur la profileuse, ils servent à faire des découpes diverses avec des formes diverses. Soit à l'entrée du feuillard ou à la sortie de la profileuse soit à l'arrêt ou bien en marche. Ils servent aussi à couper les bandes à l'entrée de la profileuse ou bien à la sortie, à la longueur désirée, lors du passage de la bande de feuillard.

La figure (I.4) montre les quatre sous-systèmes qui compose la profileuse en effet elle peut être devisée en quatre sous-système. Cette décomposition nous permet de voir les choses en claire de point de vue commande, qui est l'un des buts essentiels de notre travail.

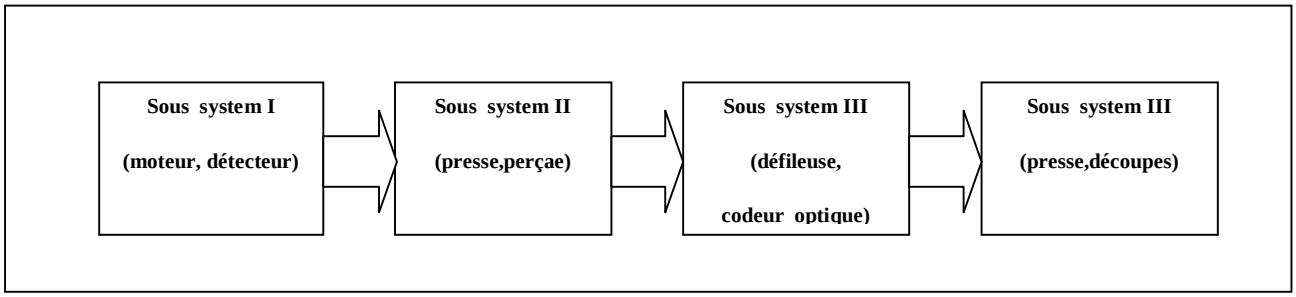


Figure I.4 les différents sous-systèmes existant sur une profileuse.

PDF-XCHANGE Click to buy NOW! www.docu-track.com

PDF-XCHANGE Click to buy NOW! www.docu-track.com

I.8 CONCLUSION

Dans ce chapitre nous avons essayé de donner une description générale des profileuses, montrer ses composant, ses sous-système, et son mode de fonctionnement, pour une éventuelle commande par automate programmable.

PDF-XCHANGE Click to buy NOW! www.docu-track.com

PDF-XCHANGE Click to buy NOW! www.docu-track.com

II 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.

Le présent chapitre donne une description plus ou moins détaillée de l'API et de tout ce qui y est lié en terme de soft et de hard.

II.1 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

PDF-XCHANGE Click to buy NOW! www.docu-track.com

PDF-XCHANGE Click to buy NOW! www.docu-track.com

certaine complexité, mais la situation a très vite changée, ce qui a rendu les systèmes câblés obsolètes.

De nombreux modèles d'automates sont aujourd'hui disponible ; 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.2 Définition générale

L'automate programmable industriel A.P.I ou en anglais PLC (Programmable Logic Controller) est un appareil électronique programmable. Il réalise des fonctions d'automatisme pour assurer la commande de pré-actionneurs et d'actionneurs à partir d'informations logiques, analogiques ou numériques. C'est aujourd'hui le constituant essentiel des automatismes. On le trouve non seulement dans tous les secteurs de l'industrie, mais aussi dans les services et dans l'agriculture.

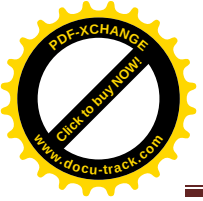
La force principale d'un automate programmable industriel API réside dans sa grande capacité de communication avec l'environnement industriel. Outre son unité centrale et son alimentation, il est constitué essentiellement de modules d'entrées/sorties, qui lui servent d'interface de communication avec le processus industriel de conduite.

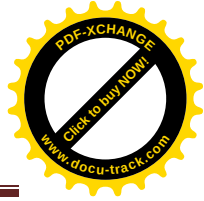
Et il a comme rôles principaux dans un processus :

- D'assurer l'acquisition de l'information fournie par les capteurs ;
- En faire le traitement ;
- Elaborer la commande des actionneurs ;
- Assurer également la communication pour l'échange d'informations avec l'environnement.

Selon la norme française EN 61131-1, un automate programmable est un :

« 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 œuvre 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 ».**[8]**

II.3 Architecture des automates programmables industriels

De forme compacte ou modulaire, les automates sont organisés suivant l'architecture suivante **(figure II.1)** :

• 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 (procédé à conduire).

• Un ou plusieurs modules de sorties 'Tout ou Rien' (TOR) ou analogiques pour transmettre à la partie opérative les signaux de commande. Il y a des modules qui intègrent en même temps des entrées et des sorties **[1]**.

• Un ou plusieurs modules de communication comprenant :

o Interfaces série utilisant dans la plupart des cas comme support de communication, les liaisons RS-232 ou RS422/RS485 ;

o Interfaces pour assurer l'accès à un bus de terrain ;

o Interface d'accès à un réseau Ethernet.

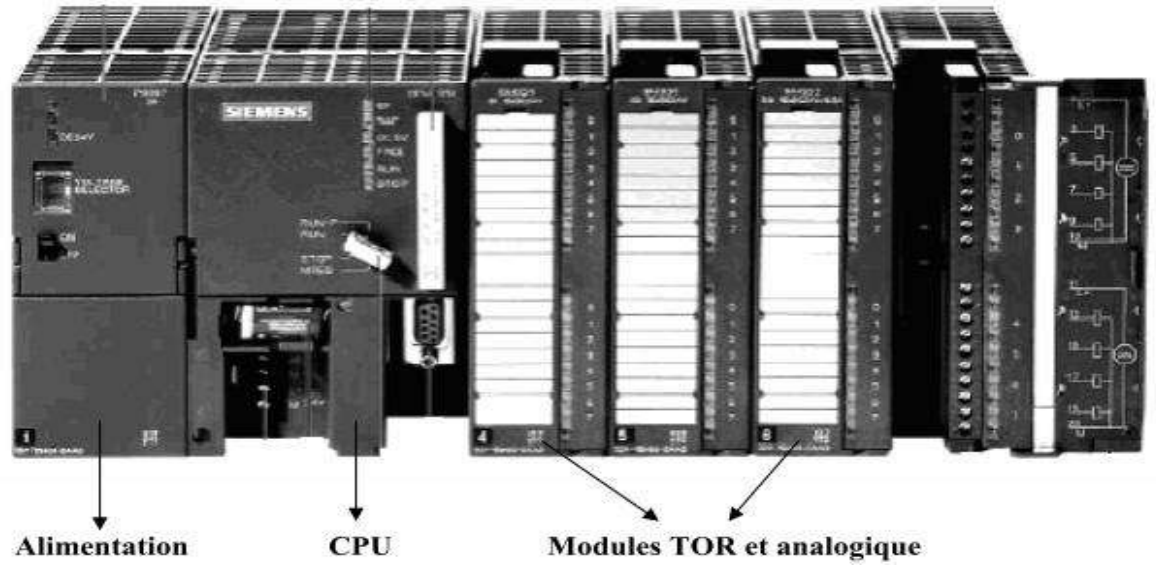


Figure II.1 l'automate programmable siemens **[14]**

II.4 Structure interne des automates programmables

La structure matérielle interne d'un API obéit au schéma donné sur la figure ci dessous **figure II.2**.

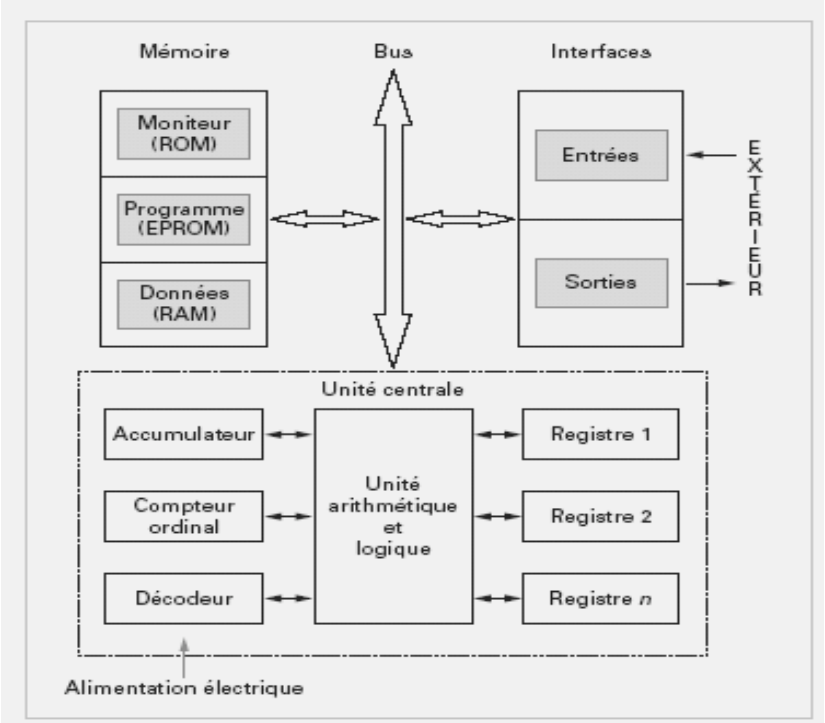


Figure 1I.2 structure interne d'un API **[8]**

Détaillons successivement chacun des composants qui apparaissent sur ce schéma.

PDF-XCHANGE Click to buy NOW! www.docu-track.com

PDF-XCHANGE Click to buy NOW! www.docu-track.com

II.4.1 Le processeur

Il Constitue le cœur de l'appareil dans l'unité centrale ; En fait, un processeur devant être automatisé, se subdivise en une multitude de domaine et processeur partiels plus petits, liés les uns aux autres.

II.4.2 Les modules d'entrées/sorties

Ils assurent le rôle d'interface entre la CPU et le processus, en récupérant les informations sur l'état de ce dernier et en coordonnant les actions.

Plusieurs types de modules sont disponibles sur le marché selon l'utilisation souhaitée

• Modules TOR : l'information traitée ne peut prendre que deux états (vrai/faux, 0 ou 1 …) C'est le type d'information délivrée par une cellule photoélectrique, un bouton poussoir …etc.

• Modules analogiques : l'information traitée est continue et prend une valeur qui évolue dans une plage bien déterminée. C'est le type d'information délivrée par un capteur (débit, niveau, pression, température…etc.).

• Modules spécialisés : l'information traitée est contenue dans des mots codés sous forme binaire ou bien hexadécimale. C'est le type d'information délivrée par un ordinateur ou un module intelligent.

II.4.3 Les mémoires

Un système de processeur est accompagné par un ou plusieurs types de mémoires. Elles permettent :

• De stocker le système d'exploitation dans des ROM ou PROM,

• Le programme dans des EEPROM,

• Les données système lors du fonctionnement dans des RAM. Cette dernière est généralement secourue par pile ou batterie. On peut, en règle générale, augmenter la capacité mémoire par adjonction de barrettes mémoires type PCMCIA.

PDF-XCHANGE Click to buy NOW! www.docu-track.com

II.4.4 L'alimentation

Elle assure la distribution d'énergie aux différents modules. L'automate est alimenté généralement par le réseau monophasé 230V-50 Hz mais d'autres alimentations sont possibles (110V …etc.).

II.4.5 Liaisons de communication

Elles Permettent la communication de l'ensemble des blocs de l'automate et des éventuelles extensions, les liaisons s'effectuent **:**

• avec l'extérieur par des borniers sur lesquels arrivent des câbles transportant le signal électrique.

• avec l'intérieur par des bus reliant divers éléments, afin de d'échanger des données, des états et des adresses.

II.5 Présentation de la gamme SIMATIC de SIEMENS

Siemens reste le seul à proposer une gamme complète de produits pour l'automatisation industrielle, par le biais de sa gamme SIMATIC **(figure II.3).**

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.

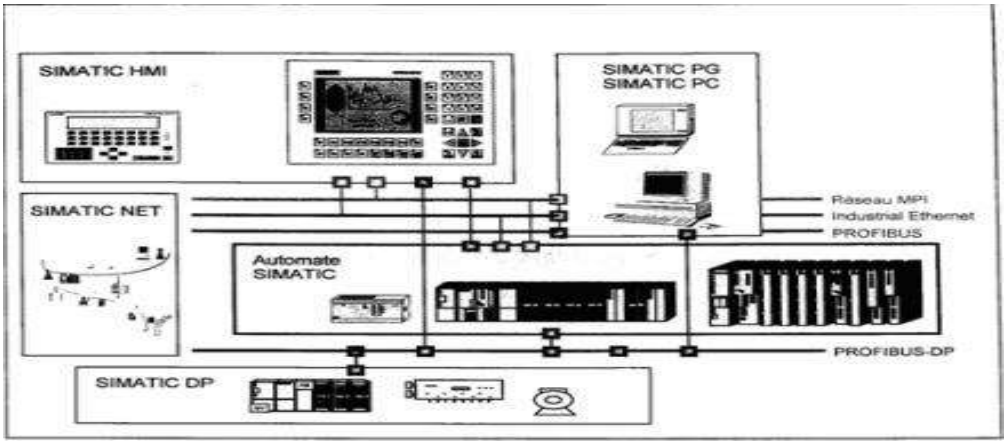


figure II.3 Présentation de la gamme de SIMATIC **[8]**

PDF-XCHANGE Click to buy NOW! www.docu-track.com

PDF-XCHANGE Click to buy NOW! www.docu-track.com

II.5.1 Les différentes variantes dans la gamme SIMATIC

a) SIMATIC S7

Cette gamme d'automates comporte trois familles :

1) S7 200, qui est un Micro-automate modulaire pour les applications simples, avec possibilité d'extensions jusqu'à 7 modules, et une mise en réseau par l'interface multipoint (MPI) ou PROFIBUS.

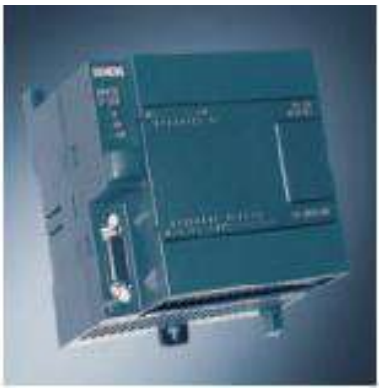


Figure II.4 L'API S200 **[14]**

2) S7300 :est un Mini-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), PROFIBUS et Industriel Ethernet.



Figure II.5 L'API S300 **[14]**

PDF-XCHANGE Click to buy NOW! www.docu-track.com

PDF-XCHANGE Click to buy NOW! www.docu-track.com

3) S7400: est un automate de haute performance pour les applications de milieu et haut de gamme, avec possibilité d'extension à plus de 300 modules, et une possibilité de mise en réseau par l'interface multipoint (MPI), PROFIBUS ou Industrial Ethernet.



Figure II.6 L'API S400 **[14]**

b. SIMATIC C7

Le SIMATIC C7 combine automate programmable et panneau opérateur dans une seule unité .L'automate compte la CPU, les modules d'entrées/sorties, et le panneau opérateur qui est utilisé comme une interface Homme/Machine HMI.

Le C7 (figure II.7) permet la visualisation des états de fonctionnement, des valeurs actuelles du processus et des anomalies.



Figure II.7 La gamme SIMATIC C7 **[14]**

c. SIMATIC M7

Les SIMATIC M7 sont des calculateurs industriels compatibles PC. Il s'agit d'un système modulaire sous boîtier, construit dans la technique des automates SIMATIC S7. Il peut

PDF-XCHANGE Click to buy NOW! www.docu-track.com

PDF-XCHANGE Click to buy NOW! www.docu-track.com

être intégré dans un automate S7 300/400 ou être utilisé comme système autonome avec une périphérie choisie dans la gamme S7.

Le M7 300/400 (figure II.8) est capable d'effectuer simultanément avec une seule CPU des opérations en temps réel, par exemple des algorithmes complexes de commande, de régulation ainsi que des tâches de visualisation et de traitement informatique. Les logiciels sous DOS ou Windows sont exploitables sur le M7-300. Par ailleurs, avec son architecture normalisée PC, il permet une extension programmable et ouverte de la plate-forme d'automatisation S7.

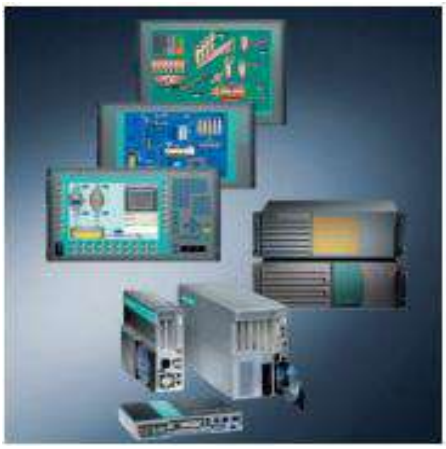


Figure II.8 La gamme SIMATIC M7 **[14]**

II.6 Description du STEP7

STEP7 est le logiciel de base pour la programmation et la configuration de systèmes d'automatisation SIMATIC **[13]**. Il permet : la création et la gestion de projets, la configuration et le paramétrage du matériel et de la communication, la gestion des mnémoniques, la création de programmes.

Il inclut 6 applications :

1. Gestionnaire de projets SIMATIC : il gère toutes les données relatives à un projet d'automatisation. Il démarre automatiquement les applications requises pour le traitement des données sélectionnées.

2. Editeur de mnémoniques : il permet de gérer toutes les variables globales. C'est à dire la définition de désignations symboliques et de commentaires pour les signaux du

PDF-XCHANGE Click to buy NOW! www.docu-track.com

PDF-XCHANGE Click to buy NOW! www.docu-track.com

processus (entrées/sorties), mémentos et blocs, l'importation et l'exportation avec d'autres programmes Windows.

3. Diagnostic du matériel : il 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. De plus permet l'affichage d'informations générales sur le module et son état, l'affichage d'erreurs sur les modules de la périphérie centrale et des esclaves DP et l'affichage des messages de la mémoire tampon de diagnostic.

4. Langages de programmation : trois langages de programmation sont inclus dans le logiciel de base : CONT (LD LadderDiagram), LIST (IL Instruction List) et LOG (FBD Function Bloc Diagram), d'autre langage de programmation peuvent être procurés sous forme de logiciel additionnel : le SCL (ST StructuredText) et le GRAPH (GRAFCET).

5. Configuration matérielle : il permet de configurer et paramétrer le matériel d'un projet d'automatisation. Il suffit juste de sélectionner le châssis (Rack) dans un catalogue électronique et leurs affecter les modules sélectionnés aux emplacements souhaités dans les racks (CPU, SM, FM…). De plus il permet le paramétrage de la CPU (comportement à la mise en route, surveillance du temps de cycle), des modules fonctionnels (FM) et de processeurs de communication (CP).

6. NetPro : il permet le transfert de données via MPI tout en offrant les possibilités de choisir les participants à la communication et de définir les liaisons de communication.

II.6.1 Création d'un projet STEP7 pas à pas

Un projet comprend deux données essentielles, les programmes et la configuration du matériel, on peut commencer par définir l'une ou l'autre, mais tout d'abord il faut démarrer le programme SIMATIC Manager **(figure II.9)**. Ce programme est l'interface graphique qui permet la manipulation du projet et l'accès aux autres programmes de STEP7.



Figure II.9 Création d'un nouveau projet

Pour en créer un nouveau, il suffit de cliquer sur le bouton 'Nouveau projet', attribuer un nom et valider. Ensuite il faut choisir une station de travail.

Une station SIMATIC représente une configuration matérielle S7 comportant un ou plusieurs modules programmables. Il existe différents types:

- SIMATIC 400 : Automate à performances extrêmes, adapté à l'exécution de programme de lourds calculs.
- SIMATIC 300 : Automate à extensibilité modulaire.
- SIMATIC H : Automate insensible aux défaillances, il se compose de 2 CPU du même type, en cas de problème elle commute de l'une vers l'autre sans perte de données.
- SIMATIC PC : ou Station PC, représente un PC ou une station OS contenant des composants SIMATIC : des applications (WinCC, par ex.), un automate logiciel ou une carte CPU enfichée dans le PC.
- Autres stations : se sont soit des appareils d'autres fabricants ou bien des stations de SIMATIC S7 contenus dans un autre projet.
- SIMATIC S5 : liaison vers un projet S5.
- PG/PC : Outils de programmation pour contrôleurs SIMATIC **(figure II.10)**, c'est une console de programmation compatible avec le milieu industriel.



Figure II.10 Station PC et PG/PC

Par exemple on va choisir une station SIMATIC 300

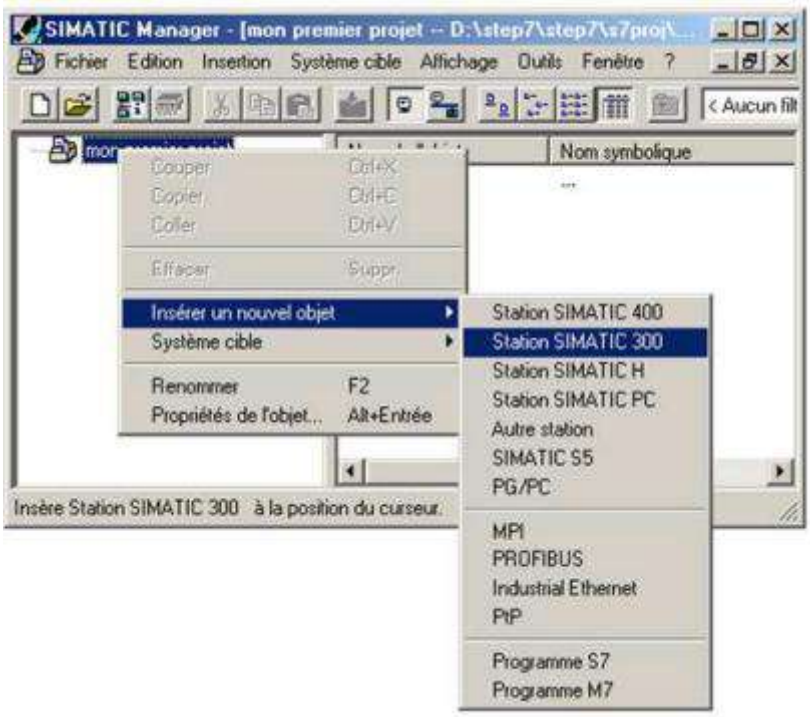


figure II.11 Choix de la station de travail

Pour commencer, le plus simple est de configurer le matériel, d'éditer les programmes puis les charger dans la CPU. Double clique sur 'Matériel' démarre l'application HW Config.

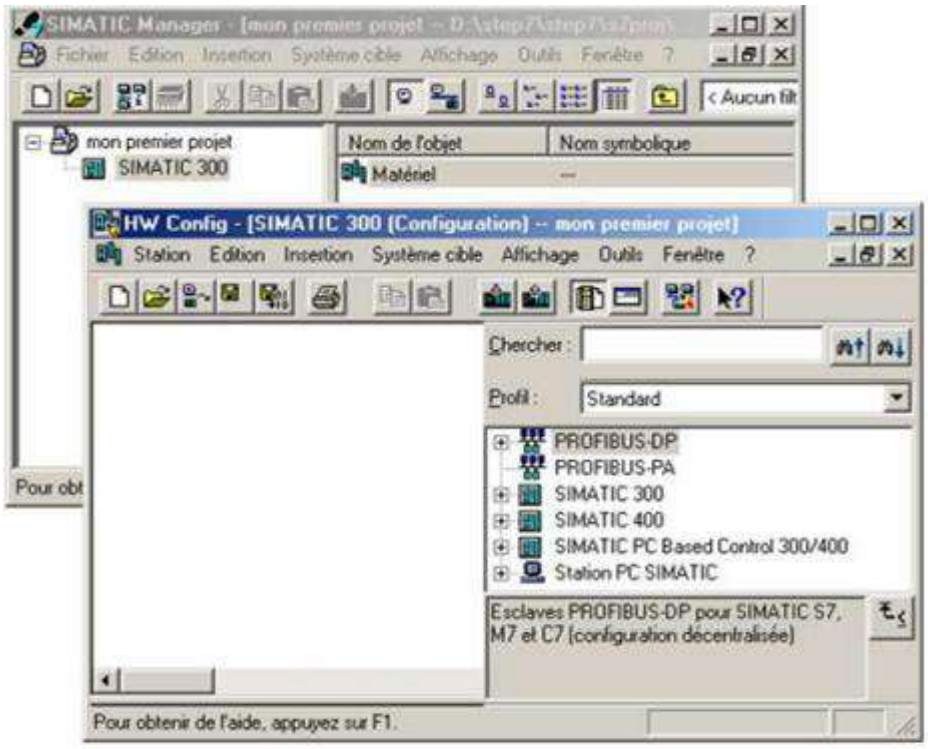


figure II.12 Configuration du matériel

II.6.2 Configuration du matériel

Pour configurer le matériel, il suffit de faire glisser des éléments du catalogue dans l'emplacement approprié, on choisit le 'Rack', l'alimentation, la CPU et les E/S...

Dans le catalogue on trouve les modules qu'on peut affecter à chaque type de station, on distingue:

- C7 : Système intégré compact qui regroupe automate programmable et interface homme-machine (pupitre opérateur) pour la réalisation de commandes de machines sous encombrement réduit.
- CP : Communication Processor, module de communication (PROFIBUS, Industriel Ethernet, Peer to Peer…).
- FM : Function Module, il regroupe les modules de fonctions (régulation, comptage…).
- IM : Coupleurs d'extension, il permet l'ajout d'autres modules.
- M7 : Modules d'extension et cartouches interface pour SIMATIC M7.
- PS : Module d'alimentation.
- Rack : Support mécanique.
- Routeur : Relie Industriel Ethernet à PROFIBUS.

PDF-XCHANGE Click to buy NOW! www.docu-track.com

PDF-XCHANGE Click to buy NOW! www.docu-track.com

- SM : Signal Module, c'est le module d'E/S, il contient le AI module d'entrées analogiques, le AO module de sorties analogiques, le DI module d'entrées TOR et le DO module de sorties TOR.
- • CPU : L'Unité Centrale, noté CPU xxx a b. xxx est la famille de la CPU, a, b sont les propriétés de la CPU (éléments additionnels, port de communication…). Par exemple :

- C : compact, la CPU intègre des modules E/S ainsi que des fonctions spécialisées.
- PtP : Peer to Peer, la CPU intègre un port de communication Point to Point.
- H : Fault-tolerant, des unités de traitements insensibles aux défaillances
- DP : Decentralized Periphery, la CPU intègre un port de communication PROFIBUS

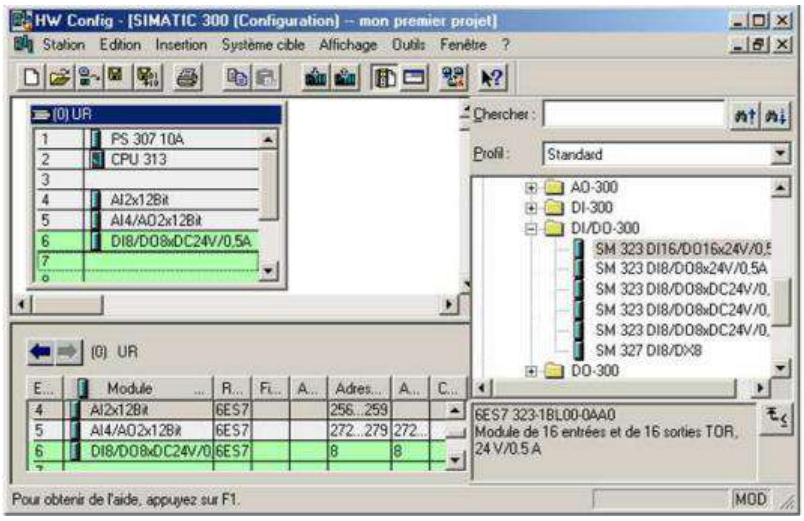


Figure II.13 sélection des modules

Si l'insertion de l'élément choisie est possible dans le Rack, la case appropriée devient verte.

Une fois le matériel choisi on sauvegarde, on compile et on charge dans la CPU.

II.6.3 Définition des mnémoniques

De retour dans le SIMATIC Manager, on trouve de nouveaux éléments. On commence par créer les mnémoniques dans la section programmes **(figure II.14)**.

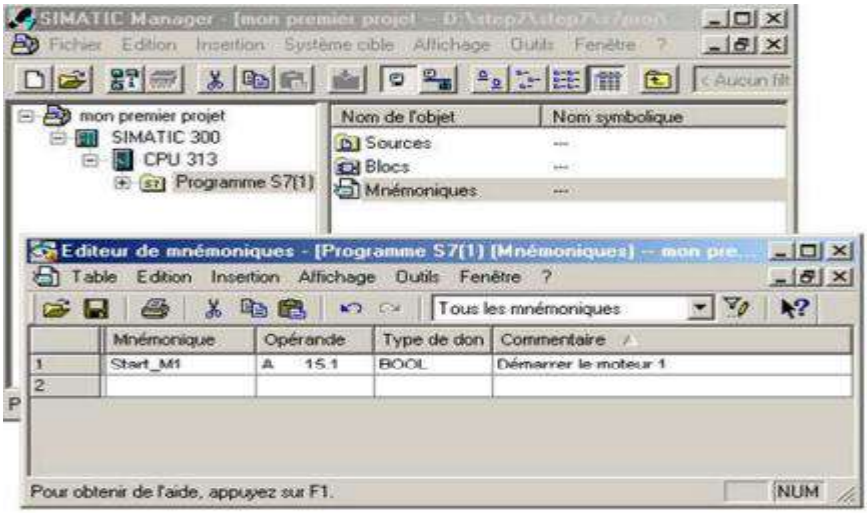


Figure II.14 Edition des mnémoniques

En affectant des noms symboliques aux adresses absolues, les programmes deviennent plus lisible, faciles à corrigé et à mettre à jour.

Il y quatre différents types d'opérande : le bit, l'octet, le mot et le double mot. Ces types définissent l'accès à une zone mémoire. Pour chaque opérande un certain types de données est permet :

- Pour le bit : BOOL : variable booléenne (True ou False, 1 ou 0).
- Pour l'octet : deux types de données sont possibles :
    1. BYTE: nombre hexadécimal de B#16#0 à B#16#FF.
    2. CHAR : Caractère ASCII, 'A', 'B'…
- Pour le mot : quatre types de données sont possible :
    1. WORD : nombre hexadécimal de W#16#0 à W#16#FFFF.
    2. INT : nombre entier de -32768 à 32767.
    3. S5TIME : Durée S7 en pas de 10 ms (valeur par défaut), de S5T#0H_0M_0S_10MS à S5T#2H_46M_30S_0MS.

PDF-XCHANGE Click to buy NOW! www.docu-track.com

PDF-XCHANGE Click to buy NOW! www.docu-track.com

4. DATE : Date en incréments de 1 jour, de D#1990-1-1 à D#2168-12-31.

• Pour le double mot : cinq types de données :

1. DWORD : nombre hexadécimal de DW#16#0000_0000 à DW#16#FFFF_FFFF.

2. DINT : nombre entier de L#-2147483648 à L#2147483647.

3. REAL : nombre à virgule flottante, Limite supérieure : ±3.402823e+38

Limite inférieure : 1.175 495e-38.

4. TIME : Durée en incréments de 1 ms, de -T≠24D_20H_31M_23S_648MS à T#24D_20H_31M_23S_647MS.

5. TIME_OF_DAY : Heure en pas de 1 ms, de TOD#0:0:0.0 à TOD#23:59:59.999.

Le tableau suivant présente les différentes zones mémoire :

Nom de la zone	**description**	**Accès à la zone par**
Mémoire image des entrées (MIE)	Au début du cycle le système d'exploitation lit les entrées provenant du processus et enregistre ces valeurs dans la MIE. Le programme utilise ces valeurs pendant son traitement normal.	Bit : E Octet : EB Mot : EW Double mot : ED
Mémoire Image des Sorties (MIS)	Pendant le cycle le programme calcule les valeurs de sortie et les déposes dans la MIS. A la fin du cycle le système d'exploitation lit les valeurs de soties figurées dans la MIS et les transmet aux sorties du processus.	Bit : A Octet : AB Mot : AW Double mot : AD
Mémentos	Se sont des zones mémoires qui permettent de sauvegarder les résultats intermédiaires calculés dans le programme.	Bit : M Octet : MB

PDF-XCHANGE Click to buy NOW! www.docu-track.com

PDF-XCHANGE Click to buy NOW! www.docu-track.com

		Mot : MW Double mot : MD
Périphérie d'entrée et de sortie	Cette zone permet d'accéder directement aux modules d'entrées et de sorties.	Octet : PEB, PAB Mot : PEW, PAW Double mot : PED ou PAD
Temporisations	Cette zone sert d'espace mémoire pour les cellules de temporisation, l'horloge accède à ces cellules afin de les mettre à jour en décrémentant la valeur de temps.	T
Compteurs	Cette zone mémoire, sert d'espace mémoire pour les opérations de comptage.	Z
données locales	Cette zone contient les données temporaires, elle est utilisé dans les blocs de code (OB, FB ou FC). Ces données sont rangées dans la pile des données locales, elles seront perdues une fois le bloc de code achevé.	Bit : L Octet : LE Mot : LW Double mot : LD

Tableau II.1 Les zones mémoires

II.6.4 Edition des programmes

Dans la section 'bloc' du SIMATIC Manager, on trouve par défaut le bloc d'organisation 1 'OB1' qui représente le programme cyclique. On peut rajouter d'autres blocs à tout moment par une clique droite dans la section Bloc de SIMATIC Manager **(figure II.15)**.

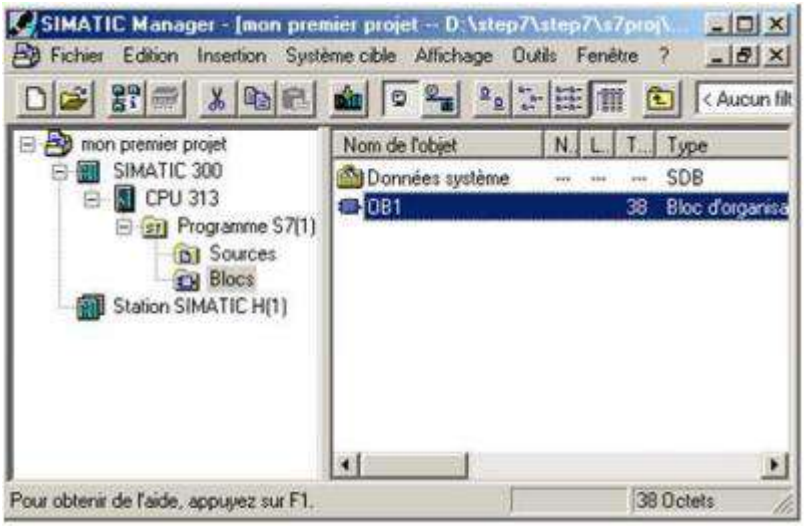


Figure II.15 Edition des programmes

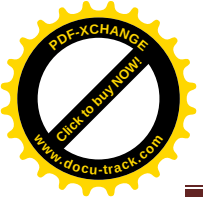
Deux programmes différents s'exécutent dans la CPU: le système d'exploitation et le programme utilisateur.

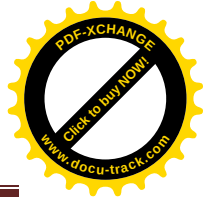
Le système d'exploitation, organise toutes les fonctions et procédures dans la CPU qui ne sont pas liées à une tâche d'automatisation spécifique. Il gère le déroulement du démarrage à chaud et du redémarrage, l'actualisation de la mémoire image des entrées et l'émission de la mémoire image des sorties, l'appel du programme utilisateur, la gestion des zones de mémoire l'enregistrement des alarmes et l'appel des OB d'alarme…

Le programme utilisateur contient toutes les fonctions nécessaires au traitement des tâches d'automatisation spécifique. Ce programme doit être créé et chargé dans la CPU par l'utilisateur.

Il détermine les conditions pour le démarrage à chaud et le redémarrage de la CPU (par exemple, initialiser des signaux), il traite les données du processus (par exemple, combiner des signaux binaires, lire et exploiter des valeurs analogiques), il doit réagir aux alarmes et traiter les perturbations dans le déroulement normal du programme.

Le STEP 7 permet de structurer le programme utilisateur en le subdivisant en différentes parties autonomes ou dépendantes. Ceci permet d'écrire des programmes importants mais clairs, simples à tester et à modifier.





II.6.4.1 Blocs d'organisation

Les blocs d'organisation (OB) constituent l'interface entre le système d'exploitation et le programme utilisateur. Ils sont appelés par le système d'exploitation et gèrent le traitement du programme cyclique et des programmes déclenchés par alarmes, ainsi que le comportement à la mise en route de l'automate programmable et le traitement des erreurs **[12]**.

Les blocs d'organisation définissent l'ordre (événements de déclenchement) dans lequel les différentes parties du programme sont traitées.

L'exécution d'un OB peut être interrompue par l'appel d'un autre OB (les OB de priorité plus élevée interrompent les OB de priorité plus faible). Les blocs d'organisations les plus prioritaires sont ceux de la mise en route (OB100, OB101 et OB102) et le moins prioritaire est le cycle en arrière plan (OB90).

On appelle alarmes les événements qui déclenchent l'appel d'un OB donné. Le type d'alarme définit la classe de priorité de celle-ci. Il existe 13 types d'alarmes :

1. Cycle libre (OB1) : il s'exécute d'une façon continue. Son traitement constitue le traitement normal pour les automates programmables. Le système d'exploitation appelle l'OB1 cycliquement et déclenche ainsi le traitement cyclique du programme utilisateur.

2. Alarmes horaires (OB10 à OB17) : elles peuvent être déclenchées une seule fois à un moment donné (indication de temps absolue avec date) ou périodiquement avec indication du commencement et de la fréquence de répétition (par exemple, toutes les minutes, toutes les heures, tous les jours.

3. Alarmes temporisées (OB20 à OB23) : elles permettent l'exécution retardée de certaines parties du programme utilisateur. Ce retard doit être définit précédemment.

4. Alarmes cycliques (OB30 à OB38) : elles sont déclenchées à des intervalles de temps précis. La période de déclenchement est toujours un multiple entier de la période de base de 1 ms.

5. Alarmes de processus (OB40 à OB47) : elles réagissent à des signaux provenant des modules (SM, CP ou FM). Elles sont déclenchées lorsqu'un module de signaux pouvant générer des alarmes de processus, avec validation d'alarme de processus paramétrée,

PDF-XCHANGE Click to buy NOW! www.docu-track.com

PDF-XCHANGE Click to buy NOW! www.docu-track.com

transmet un signal de processus reçu à la CPU ou lorsqu'un module de fonction de la CPU signale une alarme.

6. Alarme DPV1 (OB55 à OB57) : les esclaves DPV1 (modules décentralisés) peuvent déclencher des alarmes de diagnostic, de processus et de débrochage.

7. Alarme multiprocesseur (OB60) : le mode "multiprocesseur" correspond au fonctionnement simultané de plusieurs CPU (quatre au maximum) dans un châssis central. Lors du traitement de l'alarme multiprocesseur, le programme utilisateur émetteur ainsi que les programmes utilisateur s'exécutant dans les autres CPU vérifient s'ils connaissent la tâche et réagissent ensuite selon la programmation.

8. Alarmes de synchronisme d'horloge (OB61 à OB64) : elle permet la synchronisation du programme utilisateur contenu dans une CPU avec d'autres programmes répartis dans un réseau.

9. Erreur de redondance (OB70 et OB72) : elle est appelée lorsqu'une perte de redondance se produit sur le réseau PROFIBUS DP, perte de redondance des CPU, erreur de synchronisation ou erreur dans un module SYNC.

10. Erreurs asynchrones (OB80 à OB87) : elles sont appelées lorsqu'apparaît :

a. une erreur de temps : le dépassement du temps de cycle maximal, le saut d'alarmes horaires parce que l'heure a été avancée, un retard excessif pour le traitement d'une classe de priorité.

b. une défaillance dans l'appareil de base ou dans un appareil d'extension : de la tension d'alimentation 24 V, d'une pile, du système de sauvegarde entier.

c. débrochage : erreur sur l'interface au réseau MPI, erreur d'exécution du programme (l'OB correspondant ne démarre pas) ou erreur de communication.

11. Cycle en arrière plan (OB90) : Il permet d'exécuter des processus à durée non critique et ainsi d'éviter des temps d'attente.

12. Mise en route (OB100 à 102) : On distingue entre les modes de mise en route suivants:

a. redémarrage (n'existe pas pour les S7-300 et S7-400H) : OB101

PDF-XCHANGE Click to buy NOW! www.docu-track.com

PDF-XCHANGE Click to buy NOW! www.docu-track.com

b. démarrage à chaud (le seul possible pour S7-300) : OB100

c. démarrage à froid : OB102

13. Erreurs synchrones (OB121 et OB122) : Elles sont appelées lorsqu'apparaît une erreur de programmation (des temporisations adressées manquent, un bloc appelé n'est pas chargé…) ou lorsqu'une opération STEP 7 accède à une entrée ou une sortie d'un module de signaux à laquelle aucun module n'était associé lors du dernier démarrage, par exemple : erreur en cas d'accès direct à la périphérie (module défaillant ou manquant), accès à une adresse de périphérie inconnue de la CPU.

II.6.4.2 Fonctions et blocs fonctionnels

On peut programmer chaque bloc d'organisation en tant que programme structuré en créant des fonctions (FC) et des blocs fonctionnels (FB) **[12].**

• Les blocs fonctionnels (FB) sont des blocs de code associés a des blocs de données d'instance, dans les quels sont sauvegardés les paramètres effectifs et les données statique des blocs fonctionnels.

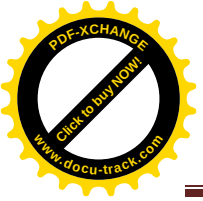
• Les fonctions (FC) sont des blocs de code sans rémanence, c'est-à-dire qu'ils ne sont pas associe à des blocs de données, les paramètres effectifs ne sont pas sauvegardés automatiquement.

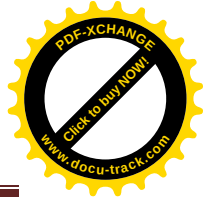
De plus il existe les blocs fonctionnels système (SFB) et les fonctions système (SFC), qui sont des fonctions préprogrammés. Ils peuvent être appelés à partir du programme utilisateur. On trouve des fonctions système pour la copie de blocs de données, le contrôle du programme utilisateur, la gestion des alarmes horaires et temporisées…

II.6.4.3 Bloc de données

Les blocs de données (DB) servent à l'enregistrement de données utilisateur. Les blocs de données globaux servent à l'enregistrement de données qui peuvent être utilisées par les autres blocs. Les blocs de données d'instances sont affectés à des blocs fonctionnels **[12]**.

Les différents blocs cités ci-dessus peuvent être édités avec l'application 'CONT LIST LOG'.





II.6.5 Programmation des blocs

La programmation des blocs de codes peut se faire à l'aide de trois applications :

1. CONT LIST LOG : elle permet de programmer des blocs d'organisations 'OB', des blocs fonctionnels 'FB' et des fonctions 'FC'.

1. GRAPH : elle permet de programmer des blocs fonctionnels 'FB'.

2. SCL : elle permet de créer des sources de code. Une source de code est un fichier texte, qui contient une suite d'instructions, une fois compilé il peut être transférer dans la CPU.

On peut trouver dans un même fichier source tout le programme utilisateur, c'est-à-dire les blocs d'organisations, les blocs fonctionnels et les fonctions.

Nota : les deux dernières applications GRAPH et SCL n'existent pas dans la version de base du STEP7, il faut les installées séparément.

On va présenter comment éditer les Blocs de code avec ces trois applications.

II.6.5.1 CONT LIST LOG

Un système de contrôle de qualité d'un certain produit fonction comme suit : L'élément X se déplace sur un tapis roulant, un premier capteur 'Cap1' vérifier sa hauteur, s'il est supérieur à une certaine hauteur h0, l'élément est éjecté en dehors du tapis grâce à un piston Pi1, sinon il poursuit son parcours vers un autre capteur 'Cap2' qui vérifier sa largeur, si elle n'est pas conforme à une certaine consigne, un piston Pi2 le pousse vers un casier en dehors du tapis.

Un front montant dans le capteur 1 (capteur 2) déclenche le piston 1 (piston 2).

On va éditer le bloc d'organisation 1, pour contrôler la gestion de ce système.

Dans SIMATIC Manager, cliquer sur le dossier programmes, ensuite sur Blocs, double clique sur l'icône du l'OB1ouvre l'application 'CONT LIST LOG'.

PDF-XCHANGE Click to buy NOW! www.docu-track.com

PDF-XCHANGE Click to buy NOW! www.docu-track.com

Choisissons par exemple le langage CONT (**figure II.16**):

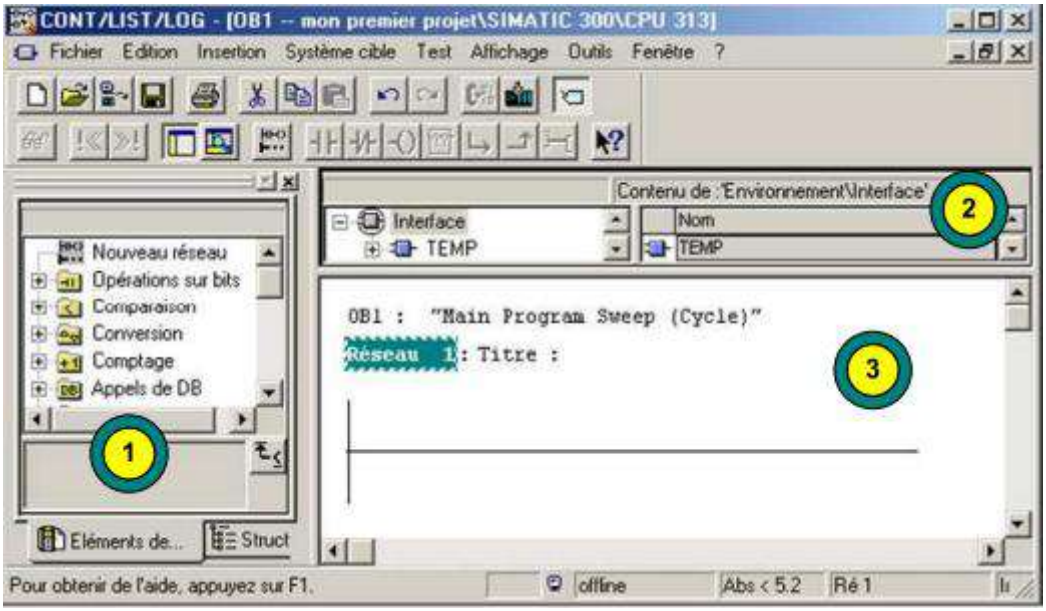


Figure II.16 CONT LIST LOG

On peut distinguer trois Zones principales **[9]** :

1. Le catalogue : c'est la bibliothèque des opérations et instructions du langage choisit (ici CONT), ces différentes opérations sont regroupées par famille : opération sur les bits, sur les mots, sur les entiers… pour utilisé une opération il suffit de faire glisser l'élément vers zone souhaitée.

2. La zone de déclarations : dans cette zone on définit les variables locales, les variables d'entrées et sorties ou arguments pour les fonctions et les blocs fonctionnelles.

3. La zone d'édition : elle est structurée en réseaux, chaque réseau finit par une affectation ou un saut.

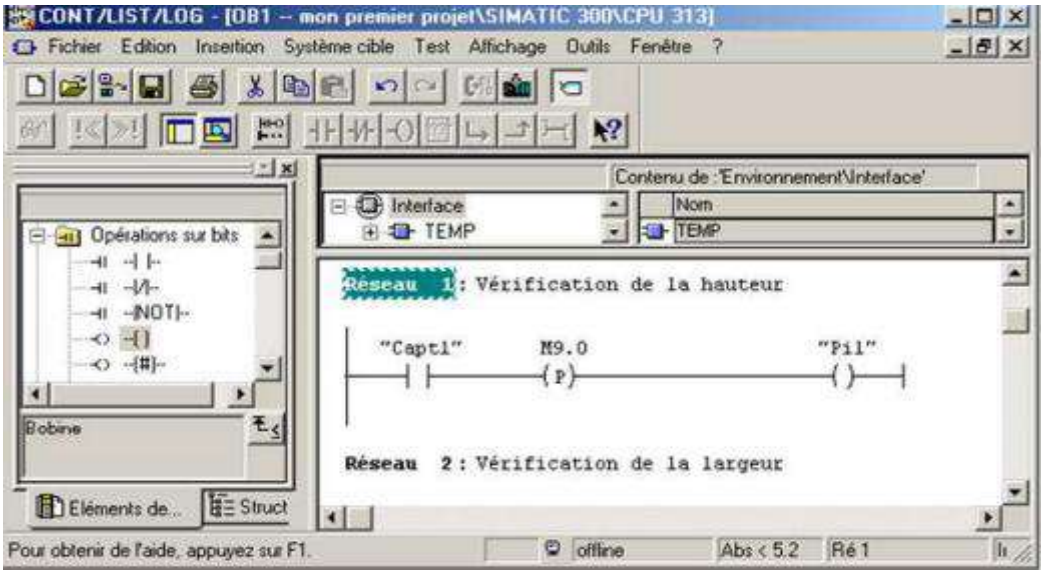


Figure II.17 exemple d'un programme en **CONT**

Pour notre exemple, on a 2 réseaux seulement. La détection du front montant se fait à l'aide de l'opération –(P)– . Elle nécessite un bit pour sauvegarder l'état des opérations qui la précède, au prochain cycle, elle compare ce bit avec le nouvel état de ces opérations.

Il est très utile d'affecter des titres aux réseaux ainsi que des commentaires. Sa permet une détection plus rapide des erreurs et permet d'écrire des programmes plus clair (en programmant, il faut penser toujours que ce n'est pas nous qui va le lire).

II.6.5.2 SCL

Le SCL est un langage de haut niveau, il est très utile pour la programmation de blocs de code complexes. Contrairement aux autres langages, le SCL peut contenir dans un seul fichier source tout le programme utilisateur (blocs d'organisation, fonctions et bloc fonctionnelles), il suffit de déclarer chaque bloc en respectant l'ordre d'appel, c'est-à-dire que si un bloc X appel un autre bloc Y, ce dernier doit être déclaré en premier dans le fichier source.

Comme application de démonstration, on se propose de calculer le gain en décibels d'un circuit électrique donné : on envoi a l'entrée du système une tension via le module de sorties analogique, ensuite on récupère la tension de sortie du système via le module d'entrées de l'automate. Il suffit après de calculer .

Dans SIMATIC Manager, cliquer su 'Programmes' ensuite sur source et créer un nouveau fichier source SCL. Double clique sur le nouveau objet inséré ouvre l'application d'édition.

PDF-XCHANGE Click to buy NOW! www.docu-track.com

PDF-XCHANGE Click to buy NOW! www.docu-track.com

Pour ajouter un bloc d'organisation, cliquer sur 'insérer', 'model de bloc' ensuite sur OB, l'application génère automatiquement quelque lignes de code.

Remplacer les xxx par le numéro du bloc, il faut faire attention de respecter la numérotation tel qu'elle est définit par la classe des blocs d'organisation.

Compléter le bloc, sauvegarder et compiler, ceci génère les blocs ajouter dans le dossier 'Bloc' de l'application SIMATIC Manager **[9]**.

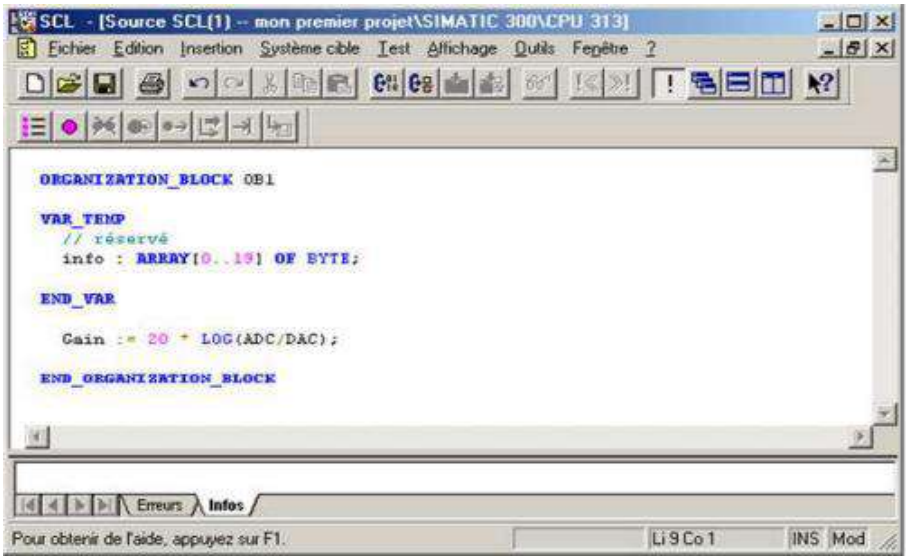


Figure II.18 exemple d'un programme en **SCL**

Bien sûre, il faut déclarer les variables Gain, ADC et DAC dans l'éditeur des mnémoniques.

II.6.5.3 S7-GRAPH

Cette application permet l'édition des blocs fonctionnels, en effet elle ne permet pas d'éditer des fonctions car elles n'ont pas des blocs de données associes pour mémorisé l'état des différentes transitions et étapes **[9]**.

On va illustrer l'édition avec S7-GRAPH, par cet exemple :

Un chariot de transport **(figure II.19)** dans une mine se trouve à l'état initiale au point A, en appuyant sur le

bouton poussoir m on démarre le moteur M1, arrivé au point B on démarre M2 jusqu'au point C, on arrête M1 et on continu avec M2 jusqu'au point D, on arrête le chariot.



Figure II.19 Commande d'un chariot

Pour satisfaire ce cahier de charge, on commence par créer un bloc fonctionnel et un autre de donnée d'instance dans le dossier Blocs du SIMTAIC Manager.

On choisit le GRAPH comme langage de programmation et on lance l'application.

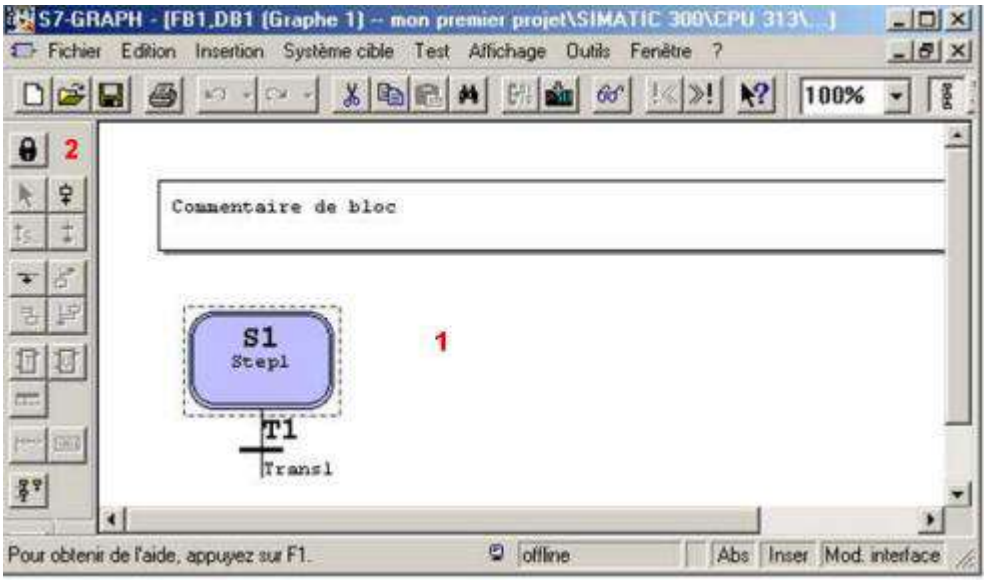


Figure II.20 Edition d'un bloc fonctionnel avec **S7-GRAPH**

On remarque deux parties, la première est la zone d'édition, la deuxième inclus les éléments à ajouter (des transitions, des branchements, des étapes…).

Pour ajouter un élément à la zone d'édition, il suffit de cliquer sur l'emplacement souhaité puis sur l'élément.

PDF-XCHANGE Click to buy NOW! www.docu-track.com

Notre exemple de commande de chariot comprend cinq étapes et quatre transitions. On commence par les crées **(figure II.21).**

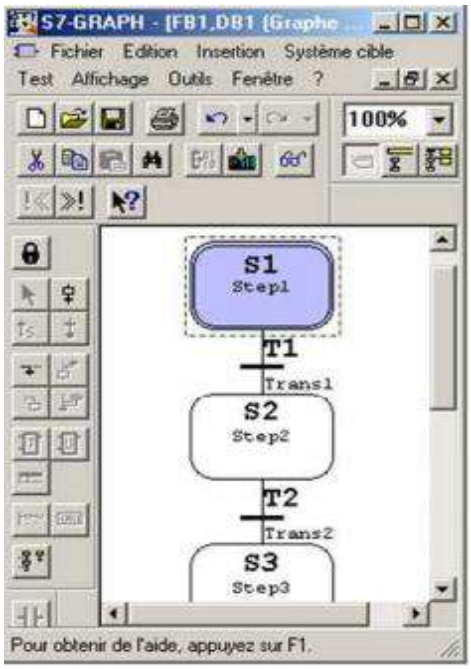


Figure II.21 Edition du **GRAPH**

Ensuite pour l'édition des actions et des transitions on change l'affichage de 'Graph' à 'Étape'.

Pour ajouter une action, cliquer par le bouton droit de la souris sur la colonne à droite et choisissez 'insérer un nouvel élément' puis sur 'action'. Il existe différent type d'action, le tableau suivant présente les plus utilisées :

Opération	opérande	Signification
N	A, E, M ou D	Tant que l'étape est active l'opérande est à 1.
S	A, E, M ou D	SET : Dès que l'étape est active, l'opérande est mis à 1.
R	A, E, M ou D	RESET : Dès que l'étape est active, l'opérande est mis à 0.
D	A, E, M ou D	Delay : n secondes après l'activation de l'étape, l'opérande se mis à 1 pour la durée de l'activation.
L	A, E, M ou D	Impulsion : quand l'étape est active, l'opérande se mis à 1 pendant n secondes

CALL	FB, FC, SFB ou SFC	Appel de bloc : tant que l'étape est active, le bloc spécifié est appelé.

Tableau II.2 Les action dans le S7-GRAPH

Pour les transitions, elles peuvent être présentées en langage CONT ou LOG, cliquer sur la colonne à gauche et insérer des éléments (contacts…) depuis le catalogue.

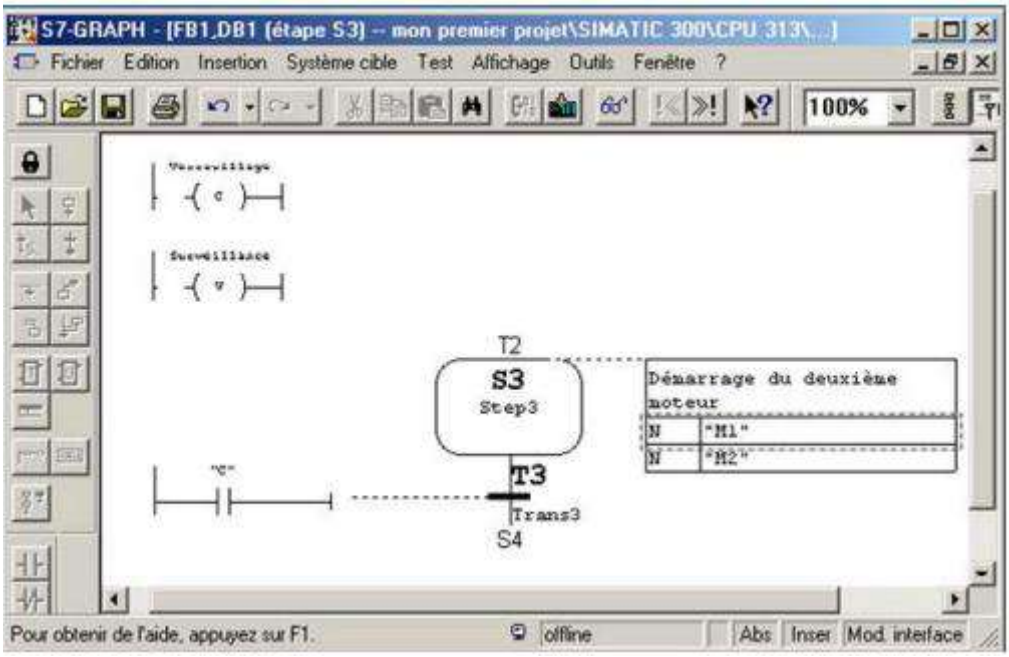


Figure II.22 Edition des actions & transitions

En haut à gauche, on remarque deux autres éléments :

1. Verrouillage : l'Interlock, c'est une condition programmable servant à verrouiller une étape et influençant l'exécution de certaines actions.

2. Surveillance : Supervision, c'est une condition programmable servant à surveiller une étape et influençant l'évolution du graphe entre cette étape et la suivante.

II.6.5.4 Avantages De L'utilisation Du S7- Graph

- Avec le langage de programmation S7-GRAPH, l'étendue des fonctions de STEP7 va être élargie à une programmation graphique pour l'enchaînement des commandes.

- Avec S7-GRAPH, vous pouvez programmer l'enchaînement des commandes clairement et rapidement.

PDF-XCHANGE Click to buy NOW! www.docu-track.com

PDF-XCHANGE Click to buy NOW! www.docu-track.com

- Le processus sera pour cela décomposé en unités et représenté graphiquement. Dans les unités, les actions à accomplir seront définies.

- Les conditions de la prochaine étape (réceptivités) peuvent être programmées en CONT ou en LOG.

Le langage de programmation S7-GRAPH correspond à la norme DIN EN 61131-3 (IEC 61131-3) définissant le langage SFC « Sequential Function Chart ».

II.6.6 Simulation de modules

L'application de simulation de modules S7-PLCSIM permet l'exécution et le teste du programme utilisateur destinés aux CPU S7-300 et aux CPU S7-400, ainsi qu'à WinLC.

La simulation étant complètement réalisée au sein du logiciel STEP 7 (**figure II.23**), il n'est pas nécessaire qu'une liaison soit établie avec un matériel S7 quelconque. Lorsque S7-PLCSIM s'exécute, toute nouvelle liaison est automatiquement dirigée vers la CPU de simulation.

S7-PLCSIM dispose d'une interface simple permettant de visualiser et de forcer les différents paramètres utilisés par le programme (par exemple, d'activer ou de désactiver des entrées).

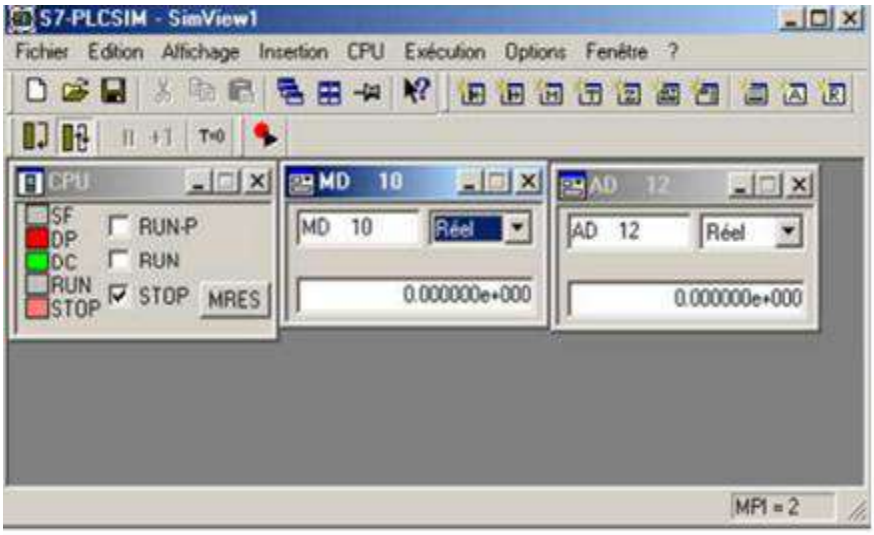
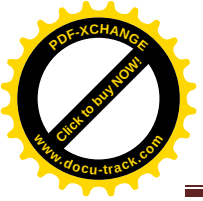
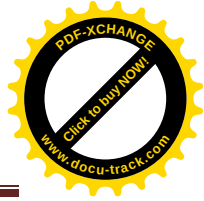


Figure II.23 Simulation de modules

En outre, S7-PLCSIM possède les fonctions suivantes :

- On peut créer des "fenêtres" dans lesquelles on a la possibilité d'accéder aux zones de mémoire d'entrée et de sortie, aux accumulateurs ainsi qu'aux registres de la CPU de simulation. On peut également accéder à la mémoire par adressage symbolique (il faut juste charger la table des mnémoniques dans 'options', puis sur 'outils' 'insérer mnémoniques').

- On peut sélectionner l'exécution automatique des temporisations ou encore les définir et les réinitialiser manuellement.

- On a la possibilité de changer l'état de fonctionnement de la CPU (STOP, RUN et RUN-P) comme pour une CPU réelle. De plus, on dispose d'une fonction de pause qui permet d'interrompre momentanément la CPU, sans affecter l'état du programme.

Bien que l'AP de simulation soit essentiellement logiciel, STEP 7 le considère comme une réelle composante matérielle, à quelques différences près :

- Contrairement à ce qui se passe avec une CPU réelle lors de la mise à l'arrêt de la CPU, l'état des sorties ne change pas.

- La CPU n'attend pas le début ou la fin du cycle pour actualiser une donnée qu'on a modifiée. Toute modification dans une fenêtre entraîne l'actualisation immédiate du contenu de l'adresse en mémoire.

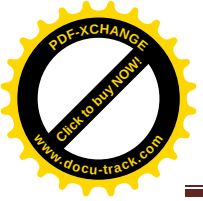
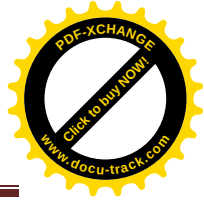
- Les options d'exécution permettent de choisir le mode d'exécution du programme par la CPU :

    - La commande Cycle unique exécute un cycle du programme, puis attend qu'on démarre l'exécution du cycle suivant.

    - La commande Cycle continu exécute le programme de la même manière que dans un AP réel : elle démarre un nouveau cycle aussitôt que le cycle précédent est terminé.

- On peut déclencher manuellement les OB d'alarme (aller dans 'exécution' puis 'déclenchement OB Erreur').

- Les modules fonctionnels (FM) ne sont pas pris en charge.

- La communication d'égal à égal n'est pas possible.

On va essayer le fonctionnement du S7-PLCSIM avec un petit exemple. Dans le dossier Programmes du SIMATIC Manager, on ajoute un bloc, par exemple une fonction

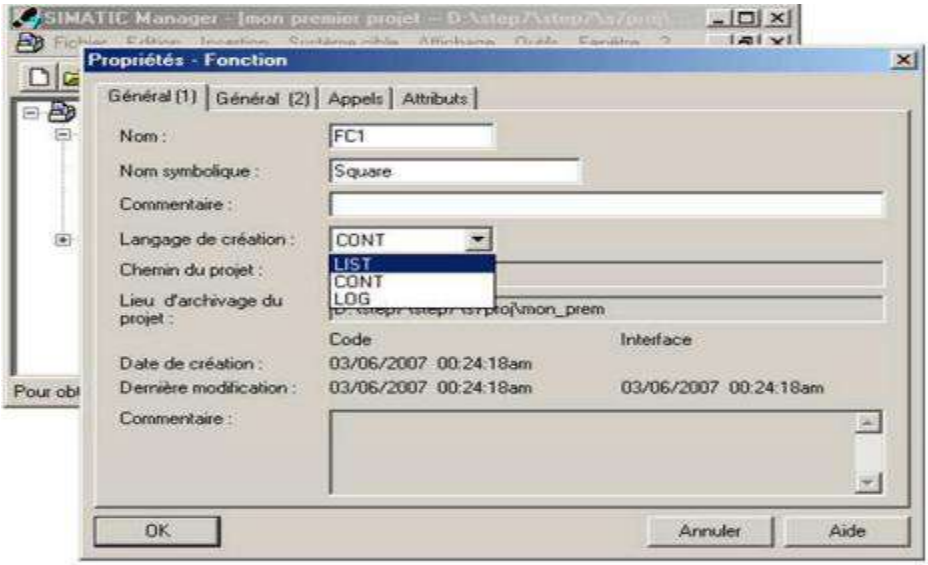


Figure II.24 Création d'une fonction

On donne à la fonction un nom symbolique, on choisit un langage d'édition parmi les trois proposés et on valide. Par exemple on va réaliser la fonction carrée, celle-ci nécessite deux arguments, un en entrée (x) et un autre en sortie (y) **figure II.25**.

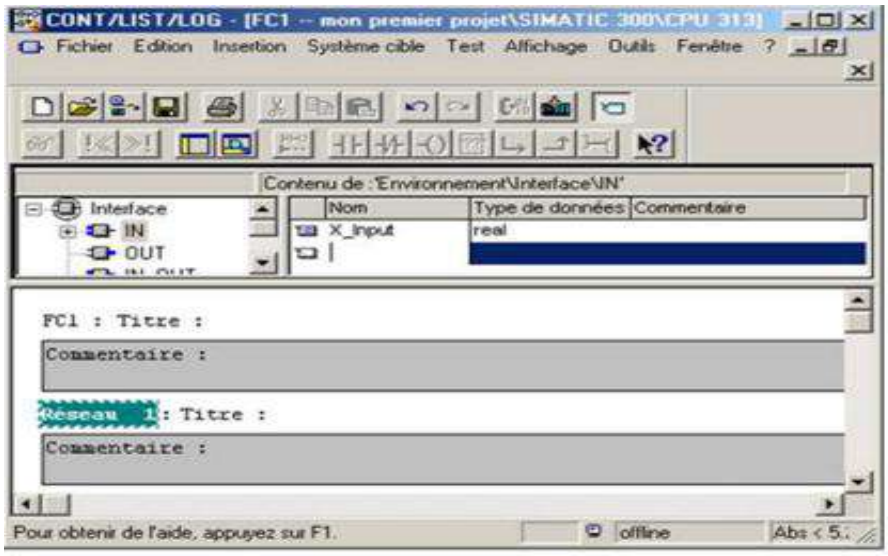
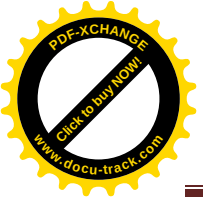
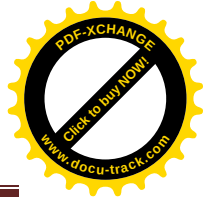


Figure II.25 Définition des arguments d'une fonction

On édit le code de cette fonction et on sauvegarde **figure II.26**.

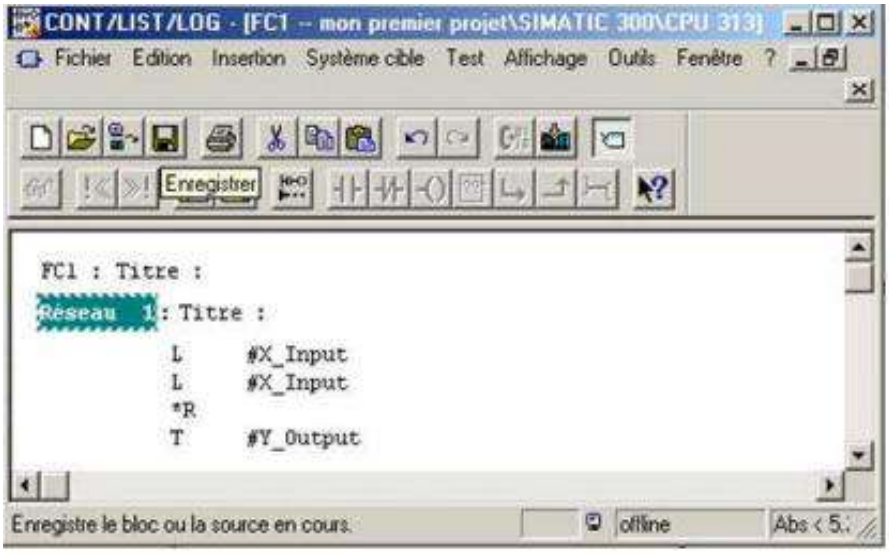


figure II.26 Edition du code en LIST

On revient sur SIMATIC Manager, on ouvre l'OB1, on change le langage d'édition dans le menu supérieur 'Affichage' puis sur 'CONT'. Maintenant sur le catalogue à gauche on appel la fonction que nous venant de créer en lui affectant les variables d'E/S **figure II.27**.

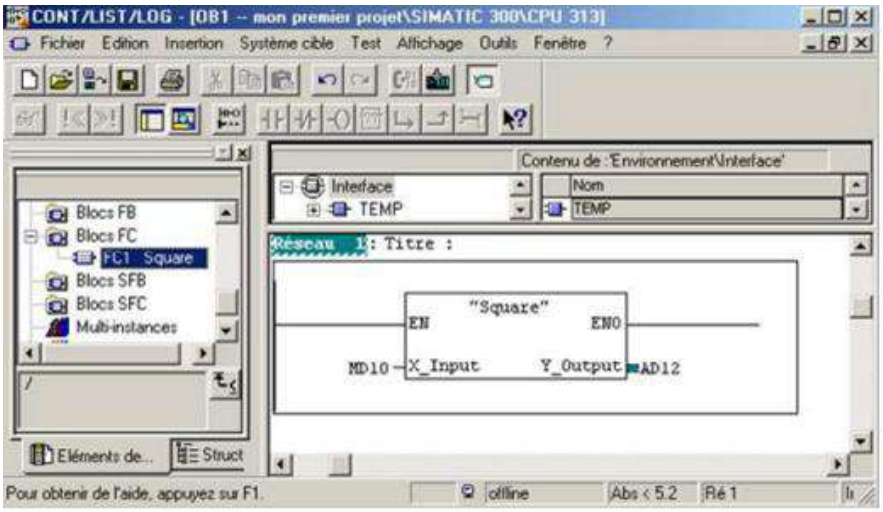


Figure II.27 Edition du code CONT

Dans cette exemple, le bloc d'organisation Cyclique OB1 appel la fonction FC1 qui va affecter au double mot de la sortie analogique 'AD12' le carrée du double mot contenu dans

le mémento 'MD10'. Le contenu de ces adresses mémoires va être interprété comme nombre réel.

On lance l'application S7-PLCSIM (aller dans 'Options' puis cliquer sur 'Simulation des modules') et on charge les blocs de code dans la CPU.

Dans le menu insertion, on ajoute le mémento 'MD10' et le double mot de la sortie analogique 'AD12', il faut sélectionner Réel comme type sinon on se retrouve avec des résultats erronés.

On démarre la simulation en cochant la case RUN, ensuite on introduit différents valeurs dans le mémento MD10 et on vérifié la valeur correspondante dans le double mot de la sortie analogique AD12 **figure II.28**.

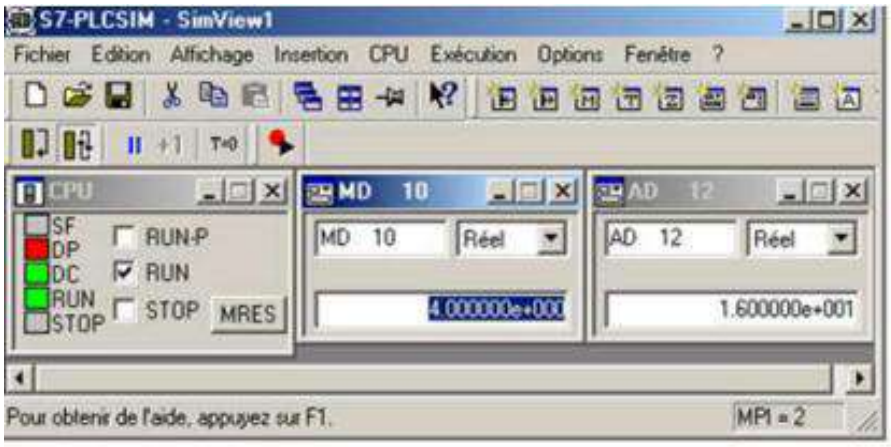


Figure II.28 Simulation di programme

II.7 WIN CC Flexible

II.7.1 Introduction

Suite à l'automatisation industrielle, l'opérateur humain a été contraint de conduire ou de superviser des machines automatisées, en réduisant les prises d'information et les actions directes sur le processus, ce qui conduit à l'élaboration d'interface d'interaction Homme/Machine, flexible et aussi lisible pour un simple opérateur. Le dialogue est d'autant plus facilité que l'écran comporte des images avec des synoptiques, des graphes, des paragraphes…etc.

Siemens avec sa gamme SIMATIC C7, contribue dans ce sens, et met à la disposition de cette demande exigeante, un grand choix en terme de pupitre de supervision et de contrôle, paramétrable par le biais d'outils divers.

Le progiciel Win CC flexible, reste le plus approprié, en terme de simplicité, pour la conception ce type d'interface de commande/diagnostic.

II.7.2 Progiciel de conception et configuration d'interface Win CC flexible

II.7.2.1 Présentation générale

Win CC flexible, est un progiciel partagé dans l'environnement STEP7, et propose pour la configuration de divers pupitres opérateurs, une famille de systèmes d'ingénierie évolutifs adaptés aux tâches de configuration.

Créer l'interface graphique et les variables, c'est pouvoir lire les valeurs du processus via l'automate, les afficher pour que l'opérateur puisse les interpréter, et ajuster, éventuellement, le processus, toujours via l'automate **Figure II.29**.

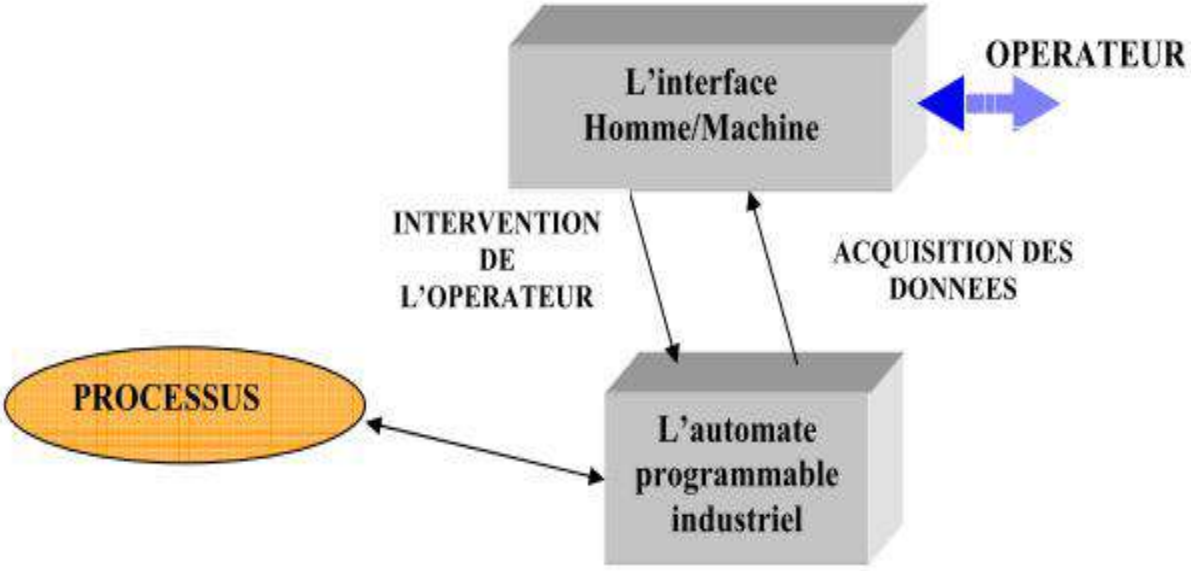


Figure II.29 L'interface Homme/Machine dans un processus automatisé

II.7.2.2 Elément du progiciel Win CC flexible

L'utilisation de ce logiciel pour la conception d'une interface, passe par plusieurs étapes résumées dans la **Figure II.30** :

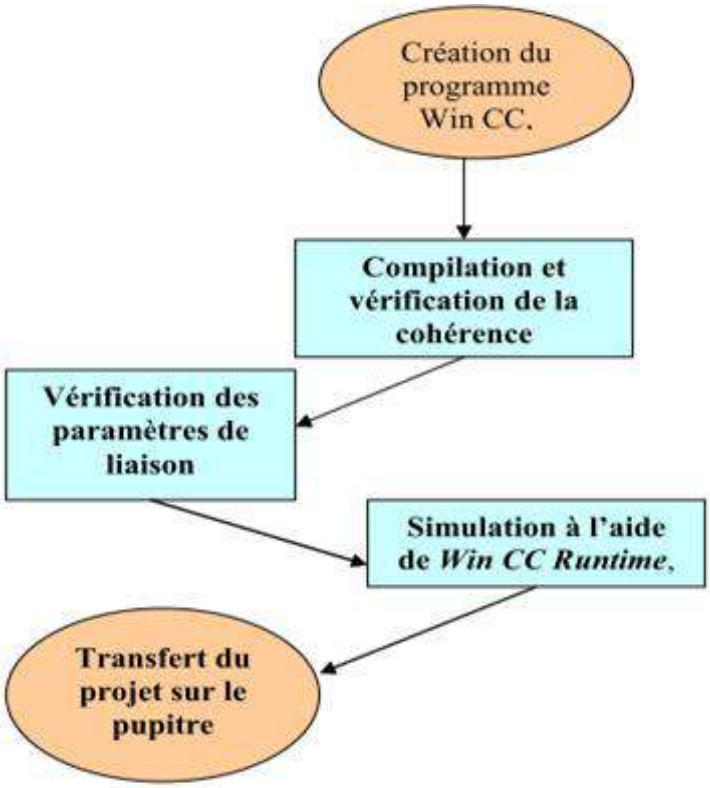


Figure II.30 Etapes pour la conception d'une interface via Win CC flexible

On commence par l'environnement de cet outil, pour terminer expliquer les différentes étapes dans les détails précédemment énumérés.

L'environnement de travail de **Win CC flexible** se compose de plusieurs éléments.

Certains de ces éléments sont liés à des éditeurs particuliers, visibles uniquement lorsque et éditeur est activé. Il met à disposition un éditeur spécifique pour chaque tâche de configuration.

On peut configurer par ex. l'interface utilisateur graphique d'un pupitre opérateur avec l'éditeur "Vues". Pour la configuration d'alarmes, on peut utiliser p. ex. l'éditeur "Alarmes TOR" ou autre.

PDF-XCHANGE Click to buy NOW! www.docu-track.com

PDF-XCHANGE Click to buy NOW! www.docu-track.com

Dans **Win CC flexible**, chaque projet créé contient principalement des vues que l'on crée pour le contrôle-commande de machines et d'installations. Lors de la création des vues, vous disposez d'objets prédéfinis permettant d'afficher des procédures et de définir des valeurs du processus.

Les différents outils et barres de l'éditeur de vues sont représentés dans la figure qui suit **Figure II.31** :

- **Barre des menus**

La barre des menus contient toutes les commandes nécessaires à l'utilisation de Win CC flexible. Les raccourcis disponibles sont indiqués en regard de la commande du menu.

- **Barres d'outils**

La barre d'outils permet d'afficher tout dont le programmeur a besoin.

- **Zone de travail**

La zone de travail sert à configurer des vues, de façon qu'il soit le plus compréhensible par l'utilisateur, et très facile à manipuler et consulter les résultats.

- **Boîte à outils**

La fenêtre des outils propose un choix d'objets simples ou complexes qu'on insère dans les vues, p. ex. des objets graphiques et éléments de commande.

- **Fenêtre des propriétés**

Le contenu de la fenêtre des propriétés dépend de la sélection actuelle dans la zone de travail, lorsqu'un objet est sélectionné, on peut éditer les propriétés de l'objet en question dans la fenêtre des propriétés.

PDF-XCHANGE Click to buy NOW! www.docu-track.com

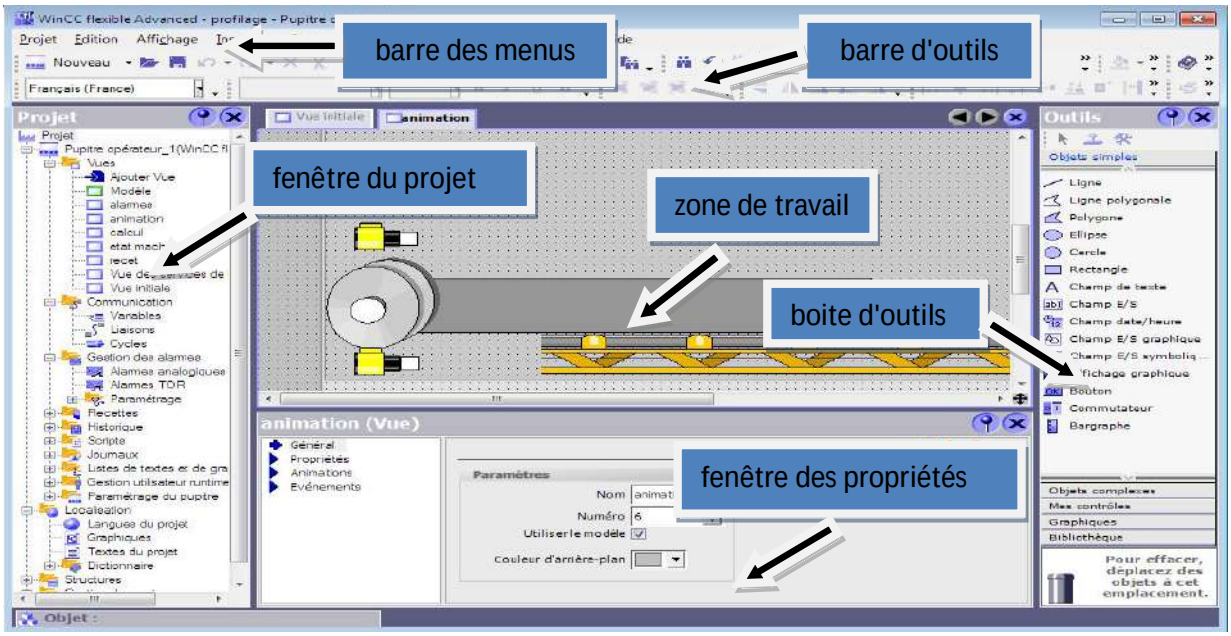


Figure II.31 Vue d'ensemble du progiciel Win CC flexible

II.7.2.3 Utilisation des variables

On distingue deux types de variables, les variables externes et les variables internes Les variables externes permettent de communiquer, c.-à-d. d'échanger des données entre les composants d'un processus automatisé, entre un pupitre opérateur et un automate par exemple.

Une variable externe est l'image d'une cellule mémoire définie de l'automate. L'accès en lecture et en écriture à cette cellule mémoire est possible aussi bien à partir du pupitre opérateur que de l'automate.

Les variables internes ne possèdent aucun lien avec l'automate. Elles sont enregistrées dans la mémoire du pupitre. Le pupitre opérateur en question peut donc accéder en lecture et en écriture aux variables internes. Les variables internes sont créées p. ex. pour exécuter des calculs locaux.

La déclaration des variables s'effectue dans l'éditeur "Variables" **Figure II.32**. Lors de leur déclaration, une configuration de base est attribuée aux variables. L'éditeur "Variables" vous permet d'adapter la configuration des variables aux besoins de votre projet ; L'éditeur s'ouvre systématiquement par ajout de variable.

PDF-XCHANGE Click to buy NOW! www.docu-track.com

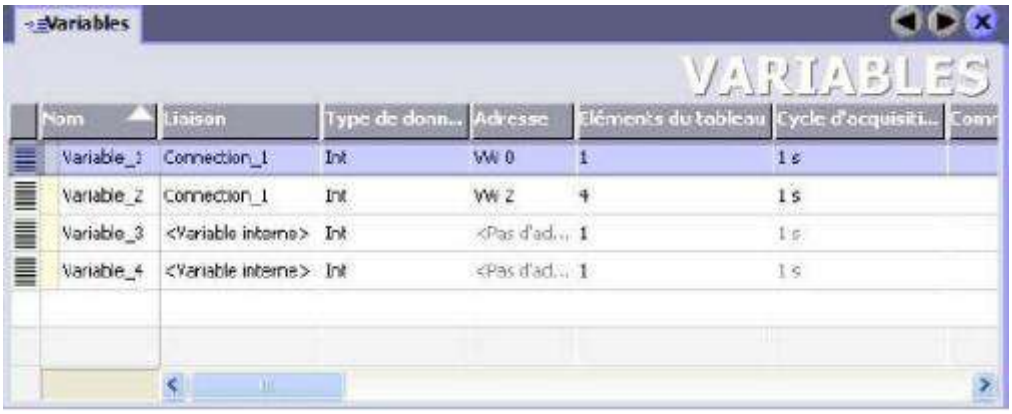


Figure II.32 éditeur de variable dans le Win CC flexible

Chaque variable peut être configurée selon le besoin et cela en accédant à sa fenêtre de priorité après l'avoir sectionnée.

II.7.2.4 Compilation et Simulation

Après avoir créé le projet et terminé la configuration, il est indispensable de vérifier la cohérence du projet, de contrôler la cohérence et de chercher les erreurs, à l'aide de la commande sur la barre du menu « contrôle de la cohérence ». Après le contrôle de cohérence, le système crée un fichier de projet compilé.

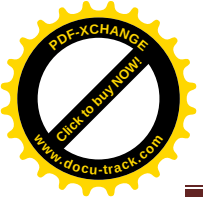
La simulation permet de détecter des erreurs logiques de configuration, par exemple, des valeurs limites incorrectes, et cela à l'aide du simulateur Runtime par la commande « démarrer le système Runtime du simulateur ».

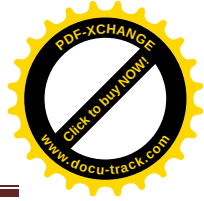
II.7.3 Conclusion

On constate la facilité et la souplesse qu'offre L'A.P.I pour sa programmation, connexion, et adaptation aux conditions industrielles, avec toutes les fonctionnalités indispensables à l'automatisation des processus. La diversité des possibilités, mise en œuvre et son coût, le rendent incontournable lors de l'élaboration d'une solution.

Mais il ne faut pas oublier qu'il est nécessaire d'avoir une bonne analyse du problème à résoudre tout en assurant le respect des règles d'installation.

La description précédente de l'outil Win CC flexible, prépare le terrain en vue de l'application «Réalisation d'un interface IHM d'une profileuse à base d'automates





programmables SIEMENS », et pendant laquelle une mise en œuvre complète sera décrite pour élaborer une interface Homme/machine.

PDF-XCHANGE Click to buy NOW! www.docu-track.com

PDF-XCHANGE Click to buy NOW! www.docu-track.com

II 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.

Le présent chapitre donne une description plus ou moins détaillée de l'API et de tout ce qui y est lié en terme de soft et de hard.

II.1 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

PDF-XCHANGE Click to buy NOW! www.docu-track.com

PDF-XCHANGE Click to buy NOW! www.docu-track.com

certaine complexité, mais la situation a très vite changée, ce qui a rendu les systèmes câblés obsolètes.

De nombreux modèles d'automates sont aujourd'hui disponible ; 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.2 Définition générale

L'automate programmable industriel A.P.I ou en anglais PLC (Programmable Logic Controller) est un appareil électronique programmable. Il réalise des fonctions d'automatisme pour assurer la commande de pré-actionneurs et d'actionneurs à partir d'informations logiques, analogiques ou numériques. C'est aujourd'hui le constituant essentiel des automatismes. On le trouve non seulement dans tous les secteurs de l'industrie, mais aussi dans les services et dans l'agriculture.

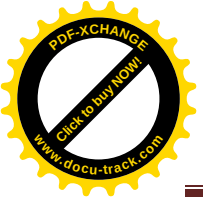
La force principale d'un automate programmable industriel API réside dans sa grande capacité de communication avec l'environnement industriel. Outre son unité centrale et son alimentation, il est constitué essentiellement de modules d'entrées/sorties, qui lui servent d'interface de communication avec le processus industriel de conduite.

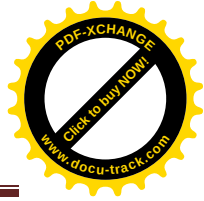
Et il a comme rôles principaux dans un processus :

- D'assurer l'acquisition de l'information fournie par les capteurs ;
- En faire le traitement ;
- Elaborer la commande des actionneurs ;
- Assurer également la communication pour l'échange d'informations avec l'environnement.

Selon la norme française EN 61131-1, un automate programmable est un :

« 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 œuvre 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 ».**[8]**

II.3 Architecture des automates programmables industriels

De forme compacte ou modulaire, les automates sont organisés suivant l'architecture suivante **(figure II.1)** :

• 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 (procédé à conduire).

• Un ou plusieurs modules de sorties 'Tout ou Rien' (TOR) ou analogiques pour transmettre à la partie opérative les signaux de commande. Il y a des modules qui intègrent en même temps des entrées et des sorties **[1]**.

• Un ou plusieurs modules de communication comprenant :

o Interfaces série utilisant dans la plupart des cas comme support de communication, les liaisons RS-232 ou RS422/RS485 ;

o Interfaces pour assurer l'accès à un bus de terrain ;

o Interface d'accès à un réseau Ethernet.

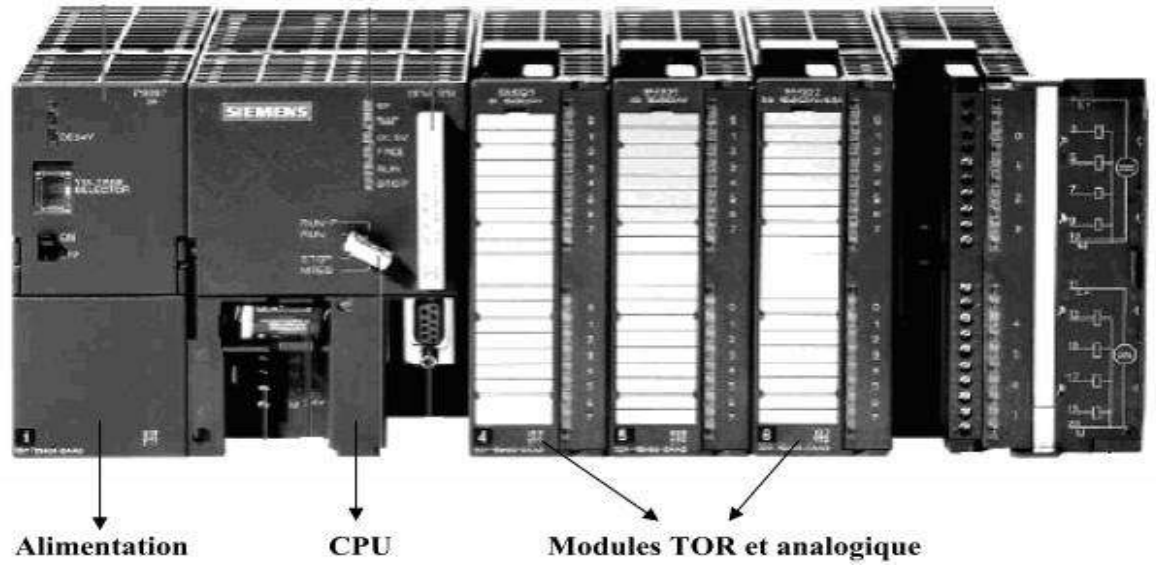


Figure II.1 l'automate programmable siemens **[14]**

II.4 Structure interne des automates programmables

La structure matérielle interne d'un API obéit au schéma donné sur la figure ci dessous **figure II.2**.

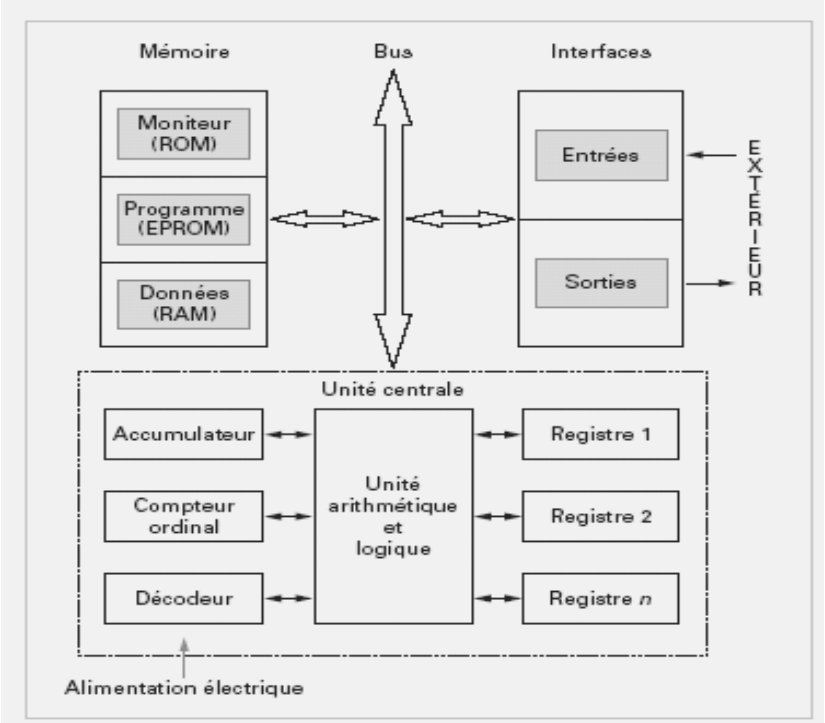


Figure 1I.2 structure interne d'un API **[8]**

Détaillons successivement chacun des composants qui apparaissent sur ce schéma.

PDF-XCHANGE Click to buy NOW! www.docu-track.com

PDF-XCHANGE Click to buy NOW! www.docu-track.com

II.4.1 Le processeur

Il Constitue le cœur de l'appareil dans l'unité centrale ; En fait, un processeur devant être automatisé, se subdivise en une multitude de domaine et processeur partiels plus petits, liés les uns aux autres.

II.4.2 Les modules d'entrées/sorties

Ils assurent le rôle d'interface entre la CPU et le processus, en récupérant les informations sur l'état de ce dernier et en coordonnant les actions.

Plusieurs types de modules sont disponibles sur le marché selon l'utilisation souhaitée

• Modules TOR : l'information traitée ne peut prendre que deux états (vrai/faux, 0 ou 1 …) C'est le type d'information délivrée par une cellule photoélectrique, un bouton poussoir …etc.

• Modules analogiques : l'information traitée est continue et prend une valeur qui évolue dans une plage bien déterminée. C'est le type d'information délivrée par un capteur (débit, niveau, pression, température…etc.).

• Modules spécialisés : l'information traitée est contenue dans des mots codés sous forme binaire ou bien hexadécimale. C'est le type d'information délivrée par un ordinateur ou un module intelligent.

II.4.3 Les mémoires

Un système de processeur est accompagné par un ou plusieurs types de mémoires. Elles permettent :

• De stocker le système d'exploitation dans des ROM ou PROM,

• Le programme dans des EEPROM,

• Les données système lors du fonctionnement dans des RAM. Cette dernière est généralement secourue par pile ou batterie. On peut, en règle générale, augmenter la capacité mémoire par adjonction de barrettes mémoires type PCMCIA.

PDF-XCHANGE Click to buy NOW! www.docu-track.com

II.4.4 L'alimentation

Elle assure la distribution d'énergie aux différents modules. L'automate est alimenté généralement par le réseau monophasé 230V-50 Hz mais d'autres alimentations sont possibles (110V …etc.).

II.4.5 Liaisons de communication

Elles Permettent la communication de l'ensemble des blocs de l'automate et des éventuelles extensions, les liaisons s'effectuent **:**

• avec l'extérieur par des borniers sur lesquels arrivent des câbles transportant le signal électrique.

• avec l'intérieur par des bus reliant divers éléments, afin de d'échanger des données, des états et des adresses.

II.5 Présentation de la gamme SIMATIC de SIEMENS

Siemens reste le seul à proposer une gamme complète de produits pour l'automatisation industrielle, par le biais de sa gamme SIMATIC **(figure II.3).**

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.

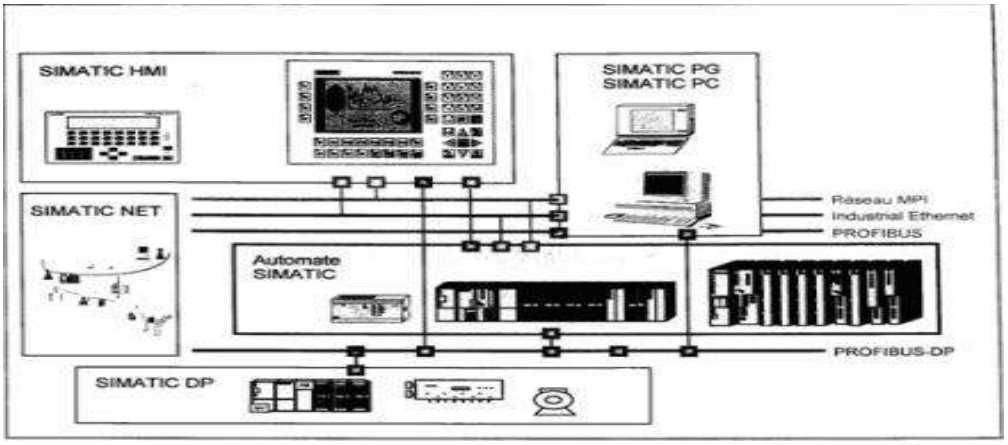


figure II.3 Présentation de la gamme de SIMATIC **[8]**

PDF-XCHANGE Click to buy NOW! www.docu-track.com

PDF-XCHANGE Click to buy NOW! www.docu-track.com

II.5.1 Les différentes variantes dans la gamme SIMATIC

a) SIMATIC S7

Cette gamme d'automates comporte trois familles :

1) S7 200, qui est un Micro-automate modulaire pour les applications simples, avec possibilité d'extensions jusqu'à 7 modules, et une mise en réseau par l'interface multipoint (MPI) ou PROFIBUS.

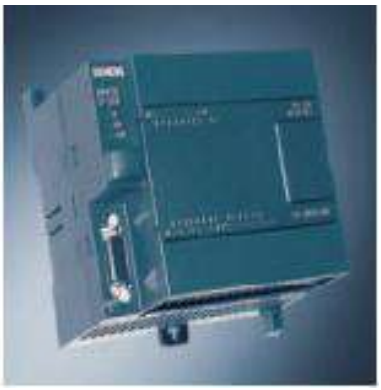


Figure II.4 L'API S200 **[14]**

2) S7300 :est un Mini-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), PROFIBUS et Industriel Ethernet.



Figure II.5 L'API S300 **[14]**

PDF-XCHANGE Click to buy NOW! www.docu-track.com

PDF-XCHANGE Click to buy NOW! www.docu-track.com

3) S7400: est un automate de haute performance pour les applications de milieu et haut de gamme, avec possibilité d'extension à plus de 300 modules, et une possibilité de mise en réseau par l'interface multipoint (MPI), PROFIBUS ou Industrial Ethernet.



Figure II.6 L'API S400 **[14]**

b. SIMATIC C7

Le SIMATIC C7 combine automate programmable et panneau opérateur dans une seule unité .L'automate compte la CPU, les modules d'entrées/sorties, et le panneau opérateur qui est utilisé comme une interface Homme/Machine HMI.

Le C7 (figure II.7) permet la visualisation des états de fonctionnement, des valeurs actuelles du processus et des anomalies.



Figure II.7 La gamme SIMATIC C7 **[14]**

c. SIMATIC M7

Les SIMATIC M7 sont des calculateurs industriels compatibles PC. Il s'agit d'un système modulaire sous boîtier, construit dans la technique des automates SIMATIC S7. Il peut

PDF-XCHANGE Click to buy NOW! www.docu-track.com

PDF-XCHANGE Click to buy NOW! www.docu-track.com

être intégré dans un automate S7 300/400 ou être utilisé comme système autonome avec une périphérie choisie dans la gamme S7.

Le M7 300/400 (figure II.8) est capable d'effectuer simultanément avec une seule CPU des opérations en temps réel, par exemple des algorithmes complexes de commande, de régulation ainsi que des tâches de visualisation et de traitement informatique. Les logiciels sous DOS ou Windows sont exploitables sur le M7-300. Par ailleurs, avec son architecture normalisée PC, il permet une extension programmable et ouverte de la plate-forme d'automatisation S7.

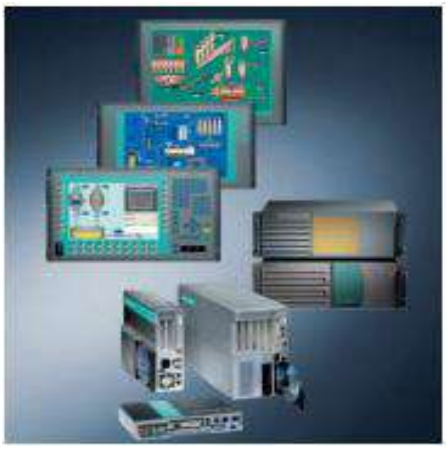


Figure II.8 La gamme SIMATIC M7 **[14]**

II.6 Description du STEP7

STEP7 est le logiciel de base pour la programmation et la configuration de systèmes d'automatisation SIMATIC **[13]**. Il permet : la création et la gestion de projets, la configuration et le paramétrage du matériel et de la communication, la gestion des mnémoniques, la création de programmes.

Il inclut 6 applications :

1. Gestionnaire de projets SIMATIC : il gère toutes les données relatives à un projet d'automatisation. Il démarre automatiquement les applications requises pour le traitement des données sélectionnées.

2. Editeur de mnémoniques : il permet de gérer toutes les variables globales. C'est à dire la définition de désignations symboliques et de commentaires pour les signaux du

PDF-XCHANGE Click to buy NOW! www.docu-track.com

PDF-XCHANGE Click to buy NOW! www.docu-track.com

processus (entrées/sorties), mémentos et blocs, l'importation et l'exportation avec d'autres programmes Windows.

3. Diagnostic du matériel : il 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. De plus permet l'affichage d'informations générales sur le module et son état, l'affichage d'erreurs sur les modules de la périphérie centrale et des esclaves DP et l'affichage des messages de la mémoire tampon de diagnostic.

4. Langages de programmation : trois langages de programmation sont inclus dans le logiciel de base : CONT (LD LadderDiagram), LIST (IL Instruction List) et LOG (FBD Function Bloc Diagram), d'autre langage de programmation peuvent être procurés sous forme de logiciel additionnel : le SCL (ST StructuredText) et le GRAPH (GRAFCET).

5. Configuration matérielle : il permet de configurer et paramétrer le matériel d'un projet d'automatisation. Il suffit juste de sélectionner le châssis (Rack) dans un catalogue électronique et leurs affecter les modules sélectionnés aux emplacements souhaités dans les racks (CPU, SM, FM…). De plus il permet le paramétrage de la CPU (comportement à la mise en route, surveillance du temps de cycle), des modules fonctionnels (FM) et de processeurs de communication (CP).

6. NetPro : il permet le transfert de données via MPI tout en offrant les possibilités de choisir les participants à la communication et de définir les liaisons de communication.

II.6.1 Création d'un projet STEP7 pas à pas

Un projet comprend deux données essentielles, les programmes et la configuration du matériel, on peut commencer par définir l'une ou l'autre, mais tout d'abord il faut démarrer le programme SIMATIC Manager **(figure II.9)**. Ce programme est l'interface graphique qui permet la manipulation du projet et l'accès aux autres programmes de STEP7.



Figure II.9 Création d'un nouveau projet

Pour en créer un nouveau, il suffit de cliquer sur le bouton 'Nouveau projet', attribuer un nom et valider. Ensuite il faut choisir une station de travail.

Une station SIMATIC représente une configuration matérielle S7 comportant un ou plusieurs modules programmables. Il existe différents types:

- SIMATIC 400 : Automate à performances extrêmes, adapté à l'exécution de programme de lourds calculs.
- SIMATIC 300 : Automate à extensibilité modulaire.
- SIMATIC H : Automate insensible aux défaillances, il se compose de 2 CPU du même type, en cas de problème elle commute de l'une vers l'autre sans perte de données.
- SIMATIC PC : ou Station PC, représente un PC ou une station OS contenant des composants SIMATIC : des applications (WinCC, par ex.), un automate logiciel ou une carte CPU enfichée dans le PC.
- Autres stations : se sont soit des appareils d'autres fabricants ou bien des stations de SIMATIC S7 contenus dans un autre projet.
- SIMATIC S5 : liaison vers un projet S5.
- PG/PC : Outils de programmation pour contrôleurs SIMATIC **(figure II.10)**, c'est une console de programmation compatible avec le milieu industriel.



Figure II.10 Station PC et PG/PC

Par exemple on va choisir une station SIMATIC 300

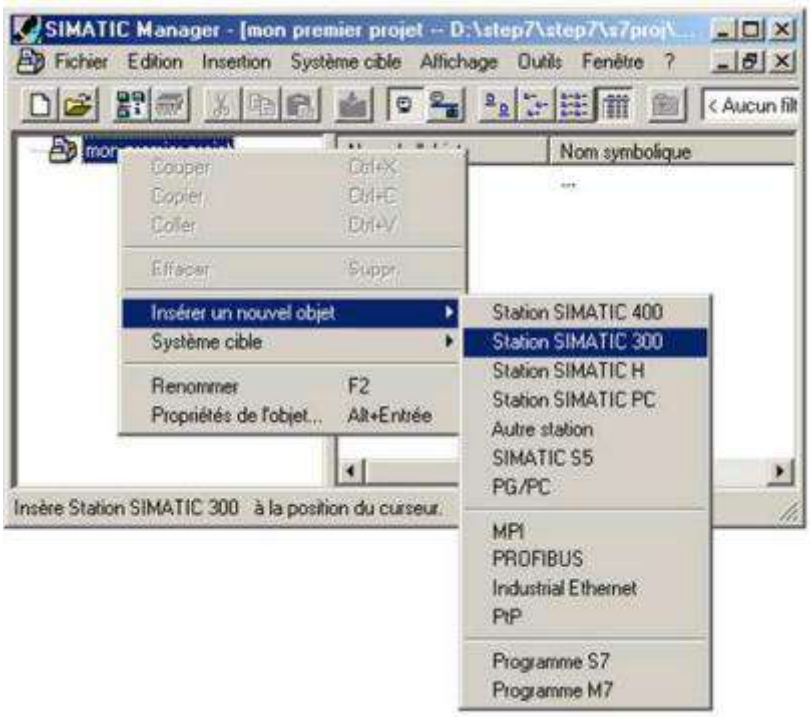


figure II.11 Choix de la station de travail

Pour commencer, le plus simple est de configurer le matériel, d'éditer les programmes puis les charger dans la CPU. Double clique sur 'Matériel' démarre l'application HW Config.

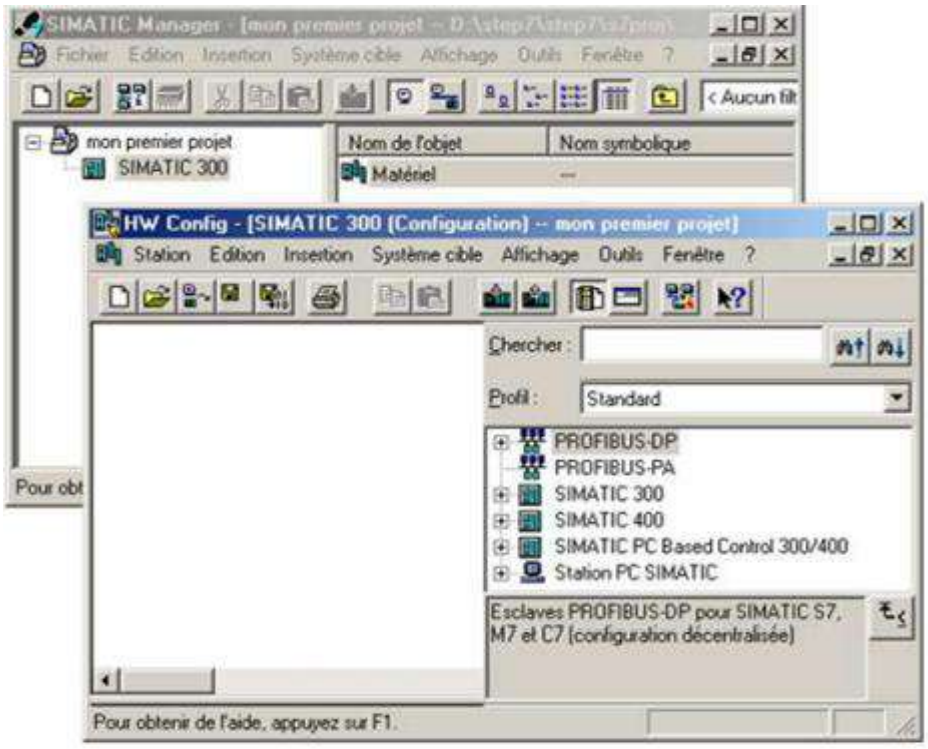


figure II.12 Configuration du matériel

II.6.2 Configuration du matériel

Pour configurer le matériel, il suffit de faire glisser des éléments du catalogue dans l'emplacement approprié, on choisit le 'Rack', l'alimentation, la CPU et les E/S...

Dans le catalogue on trouve les modules qu'on peut affecter à chaque type de station, on distingue:

- C7 : Système intégré compact qui regroupe automate programmable et interface homme-machine (pupitre opérateur) pour la réalisation de commandes de machines sous encombrement réduit.
- CP : Communication Processor, module de communication (PROFIBUS, Industriel Ethernet, Peer to Peer…).
- FM : Function Module, il regroupe les modules de fonctions (régulation, comptage…).
- IM : Coupleurs d'extension, il permet l'ajout d'autres modules.
- M7 : Modules d'extension et cartouches interface pour SIMATIC M7.
- PS : Module d'alimentation.
- Rack : Support mécanique.
- Routeur : Relie Industriel Ethernet à PROFIBUS.

PDF-XCHANGE Click to buy NOW! www.docu-track.com

PDF-XCHANGE Click to buy NOW! www.docu-track.com

- SM : Signal Module, c'est le module d'E/S, il contient le AI module d'entrées analogiques, le AO module de sorties analogiques, le DI module d'entrées TOR et le DO module de sorties TOR.
- • CPU : L'Unité Centrale, noté CPU xxx a b. xxx est la famille de la CPU, a, b sont les propriétés de la CPU (éléments additionnels, port de communication…). Par exemple :

- C : compact, la CPU intègre des modules E/S ainsi que des fonctions spécialisées.
- PtP : Peer to Peer, la CPU intègre un port de communication Point to Point.
- H : Fault-tolerant, des unités de traitements insensibles aux défaillances
- DP : Decentralized Periphery, la CPU intègre un port de communication PROFIBUS

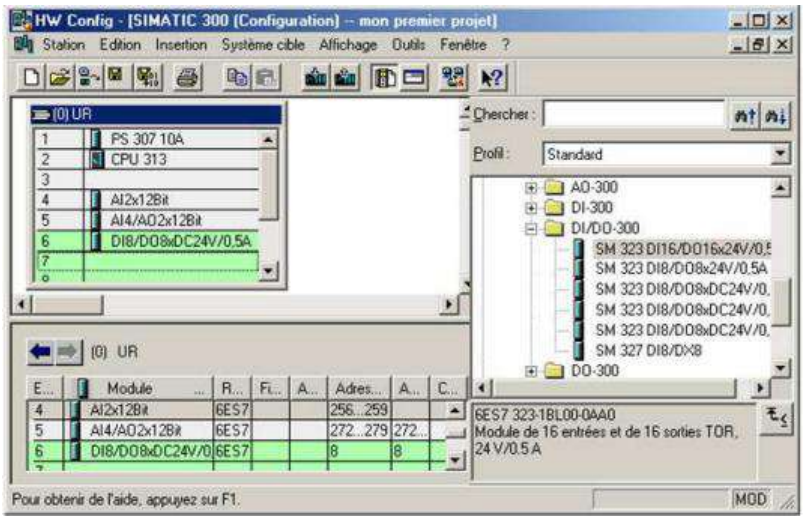


Figure II.13 sélection des modules

Si l'insertion de l'élément choisie est possible dans le Rack, la case appropriée devient verte.

Une fois le matériel choisi on sauvegarde, on compile et on charge dans la CPU.

II.6.3 Définition des mnémoniques

De retour dans le SIMATIC Manager, on trouve de nouveaux éléments. On commence par créer les mnémoniques dans la section programmes **(figure II.14)**.

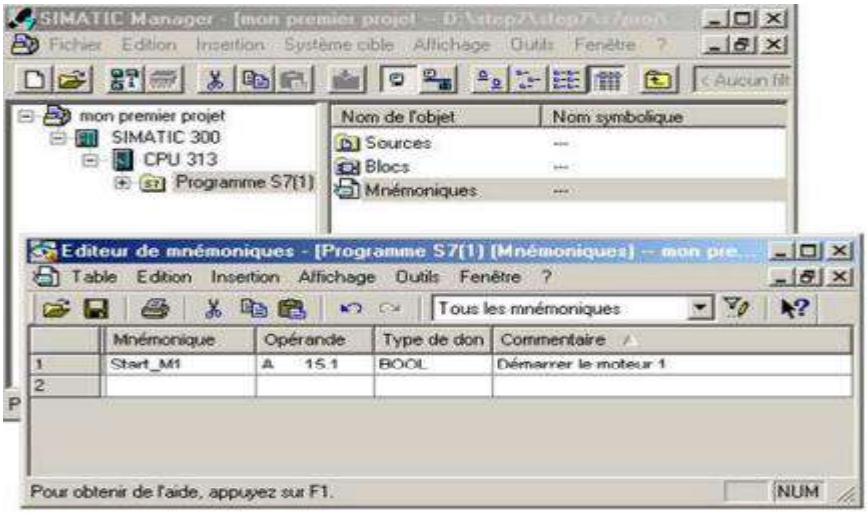


Figure II.14 Edition des mnémoniques

En affectant des noms symboliques aux adresses absolues, les programmes deviennent plus lisible, faciles à corrigé et à mettre à jour.

Il y quatre différents types d'opérande : le bit, l'octet, le mot et le double mot. Ces types définissent l'accès à une zone mémoire. Pour chaque opérande un certain types de données est permet :

- Pour le bit : BOOL : variable booléenne (True ou False, 1 ou 0).
- Pour l'octet : deux types de données sont possibles :
    1. BYTE: nombre hexadécimal de B#16#0 à B#16#FF.
    2. CHAR : Caractère ASCII, 'A', 'B'…
- Pour le mot : quatre types de données sont possible :
    1. WORD : nombre hexadécimal de W#16#0 à W#16#FFFF.
    2. INT : nombre entier de -32768 à 32767.
    3. S5TIME : Durée S7 en pas de 10 ms (valeur par défaut), de S5T#0H_0M_0S_10MS à S5T#2H_46M_30S_0MS.

PDF-XCHANGE Click to buy NOW! www.docu-track.com

PDF-XCHANGE Click to buy NOW! www.docu-track.com

4. DATE : Date en incréments de 1 jour, de D#1990-1-1 à D#2168-12-31.

• Pour le double mot : cinq types de données :

1. DWORD : nombre hexadécimal de DW#16#0000_0000 à DW#16#FFFF_FFFF.

2. DINT : nombre entier de L#-2147483648 à L#2147483647.

3. REAL : nombre à virgule flottante, Limite supérieure : ±3.402823e+38

Limite inférieure : 1.175 495e-38.

4. TIME : Durée en incréments de 1 ms, de -T≠24D_20H_31M_23S_648MS à T#24D_20H_31M_23S_647MS.

5. TIME_OF_DAY : Heure en pas de 1 ms, de TOD#0:0:0.0 à TOD#23:59:59.999.

Le tableau suivant présente les différentes zones mémoire :

Nom de la zone	**description**	**Accès à la zone par**
Mémoire image des entrées (MIE)	Au début du cycle le système d'exploitation lit les entrées provenant du processus et enregistre ces valeurs dans la MIE. Le programme utilise ces valeurs pendant son traitement normal.	Bit : E Octet : EB Mot : EW Double mot : ED
Mémoire Image des Sorties (MIS)	Pendant le cycle le programme calcule les valeurs de sortie et les déposes dans la MIS. A la fin du cycle le système d'exploitation lit les valeurs de soties figurées dans la MIS et les transmet aux sorties du processus.	Bit : A Octet : AB Mot : AW Double mot : AD
Mémentos	Se sont des zones mémoires qui permettent de sauvegarder les résultats intermédiaires calculés dans le programme.	Bit : M Octet : MB

PDF-XCHANGE Click to buy NOW! www.docu-track.com

PDF-XCHANGE Click to buy NOW! www.docu-track.com

		Mot : MW Double mot : MD
Périphérie d'entrée et de sortie	Cette zone permet d'accéder directement aux modules d'entrées et de sorties.	Octet : PEB, PAB Mot : PEW, PAW Double mot : PED ou PAD
Temporisations	Cette zone sert d'espace mémoire pour les cellules de temporisation, l'horloge accède à ces cellules afin de les mettre à jour en décrémentant la valeur de temps.	T
Compteurs	Cette zone mémoire, sert d'espace mémoire pour les opérations de comptage.	Z
données locales	Cette zone contient les données temporaires, elle est utilisé dans les blocs de code (OB, FB ou FC). Ces données sont rangées dans la pile des données locales, elles seront perdues une fois le bloc de code achevé.	Bit : L Octet : LE Mot : LW Double mot : LD

Tableau II.1 Les zones mémoires

II.6.4 Edition des programmes

Dans la section 'bloc' du SIMATIC Manager, on trouve par défaut le bloc d'organisation 1 'OB1' qui représente le programme cyclique. On peut rajouter d'autres blocs à tout moment par une clique droite dans la section Bloc de SIMATIC Manager **(figure II.15)**.

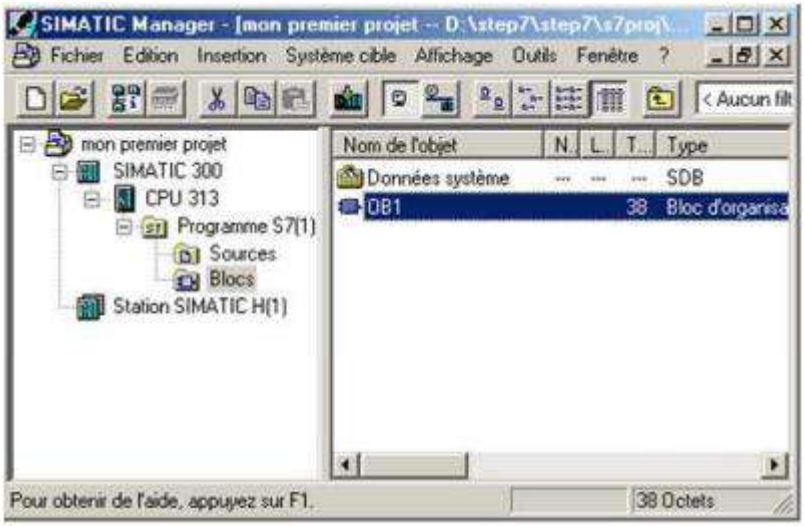


Figure II.15 Edition des programmes

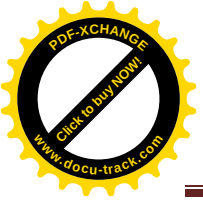
Deux programmes différents s'exécutent dans la CPU: le système d'exploitation et le programme utilisateur.

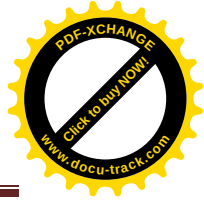
Le système d'exploitation, organise toutes les fonctions et procédures dans la CPU qui ne sont pas liées à une tâche d'automatisation spécifique. Il gère le déroulement du démarrage à chaud et du redémarrage, l'actualisation de la mémoire image des entrées et l'émission de la mémoire image des sorties, l'appel du programme utilisateur, la gestion des zones de mémoire l'enregistrement des alarmes et l'appel des OB d'alarme…

Le programme utilisateur contient toutes les fonctions nécessaires au traitement des tâches d'automatisation spécifique. Ce programme doit être créé et chargé dans la CPU par l'utilisateur.

Il détermine les conditions pour le démarrage à chaud et le redémarrage de la CPU (par exemple, initialiser des signaux), il traite les données du processus (par exemple, combiner des signaux binaires, lire et exploiter des valeurs analogiques), il doit réagir aux alarmes et traiter les perturbations dans le déroulement normal du programme.

Le STEP 7 permet de structurer le programme utilisateur en le subdivisant en différentes parties autonomes ou dépendantes. Ceci permet d'écrire des programmes importants mais clairs, simples à tester et à modifier.





II.6.4.1 Blocs d'organisation

Les blocs d'organisation (OB) constituent l'interface entre le système d'exploitation et le programme utilisateur. Ils sont appelés par le système d'exploitation et gèrent le traitement du programme cyclique et des programmes déclenchés par alarmes, ainsi que le comportement à la mise en route de l'automate programmable et le traitement des erreurs **[12]**.

Les blocs d'organisation définissent l'ordre (événements de déclenchement) dans lequel les différentes parties du programme sont traitées.

L'exécution d'un OB peut être interrompue par l'appel d'un autre OB (les OB de priorité plus élevée interrompent les OB de priorité plus faible). Les blocs d'organisations les plus prioritaires sont ceux de la mise en route (OB100, OB101 et OB102) et le moins prioritaire est le cycle en arrière plan (OB90).

On appelle alarmes les événements qui déclenchent l'appel d'un OB donné. Le type d'alarme définit la classe de priorité de celle-ci. Il existe 13 types d'alarmes :

1. Cycle libre (OB1) : il s'exécute d'une façon continue. Son traitement constitue le traitement normal pour les automates programmables. Le système d'exploitation appelle l'OB1 cycliquement et déclenche ainsi le traitement cyclique du programme utilisateur.

2. Alarmes horaires (OB10 à OB17) : elles peuvent être déclenchées une seule fois à un moment donné (indication de temps absolue avec date) ou périodiquement avec indication du commencement et de la fréquence de répétition (par exemple, toutes les minutes, toutes les heures, tous les jours.

3. Alarmes temporisées (OB20 à OB23) : elles permettent l'exécution retardée de certaines parties du programme utilisateur. Ce retard doit être définit précédemment.

4. Alarmes cycliques (OB30 à OB38) : elles sont déclenchées à des intervalles de temps précis. La période de déclenchement est toujours un multiple entier de la période de base de 1 ms.

5. Alarmes de processus (OB40 à OB47) : elles réagissent à des signaux provenant des modules (SM, CP ou FM). Elles sont déclenchées lorsqu'un module de signaux pouvant générer des alarmes de processus, avec validation d'alarme de processus paramétrée,

PDF-XCHANGE Click to buy NOW! www.docu-track.com

PDF-XCHANGE Click to buy NOW! www.docu-track.com

transmet un signal de processus reçu à la CPU ou lorsqu'un module de fonction de la CPU signale une alarme.

6. Alarme DPV1 (OB55 à OB57) : les esclaves DPV1 (modules décentralisés) peuvent déclencher des alarmes de diagnostic, de processus et de débrochage.

7. Alarme multiprocesseur (OB60) : le mode "multiprocesseur" correspond au fonctionnement simultané de plusieurs CPU (quatre au maximum) dans un châssis central. Lors du traitement de l'alarme multiprocesseur, le programme utilisateur émetteur ainsi que les programmes utilisateur s'exécutant dans les autres CPU vérifient s'ils connaissent la tâche et réagissent ensuite selon la programmation.

8. Alarmes de synchronisme d'horloge (OB61 à OB64) : elle permet la synchronisation du programme utilisateur contenu dans une CPU avec d'autres programmes répartis dans un réseau.

9. Erreur de redondance (OB70 et OB72) : elle est appelée lorsqu'une perte de redondance se produit sur le réseau PROFIBUS DP, perte de redondance des CPU, erreur de synchronisation ou erreur dans un module SYNC.

10. Erreurs asynchrones (OB80 à OB87) : elles sont appelées lorsqu'apparaît :

a. une erreur de temps : le dépassement du temps de cycle maximal, le saut d'alarmes horaires parce que l'heure a été avancée, un retard excessif pour le traitement d'une classe de priorité.

b. une défaillance dans l'appareil de base ou dans un appareil d'extension : de la tension d'alimentation 24 V, d'une pile, du système de sauvegarde entier.

c. débrochage : erreur sur l'interface au réseau MPI, erreur d'exécution du programme (l'OB correspondant ne démarre pas) ou erreur de communication.

11. Cycle en arrière plan (OB90) : Il permet d'exécuter des processus à durée non critique et ainsi d'éviter des temps d'attente.

12. Mise en route (OB100 à 102) : On distingue entre les modes de mise en route suivants:

a. redémarrage (n'existe pas pour les S7-300 et S7-400H) : OB101

PDF-XCHANGE Click to buy NOW! www.docu-track.com

PDF-XCHANGE Click to buy NOW! www.docu-track.com

b. démarrage à chaud (le seul possible pour S7-300) : OB100

c. démarrage à froid : OB102

13. Erreurs synchrones (OB121 et OB122) : Elles sont appelées lorsqu'apparaît une erreur de programmation (des temporisations adressées manquent, un bloc appelé n'est pas chargé…) ou lorsqu'une opération STEP 7 accède à une entrée ou une sortie d'un module de signaux à laquelle aucun module n'était associé lors du dernier démarrage, par exemple : erreur en cas d'accès direct à la périphérie (module défaillant ou manquant), accès à une adresse de périphérie inconnue de la CPU.

II.6.4.2 Fonctions et blocs fonctionnels

On peut programmer chaque bloc d'organisation en tant que programme structuré en créant des fonctions (FC) et des blocs fonctionnels (FB) **[12].**

• Les blocs fonctionnels (FB) sont des blocs de code associés a des blocs de données d'instance, dans les quels sont sauvegardés les paramètres effectifs et les données statique des blocs fonctionnels.

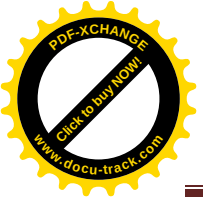
• Les fonctions (FC) sont des blocs de code sans rémanence, c'est-à-dire qu'ils ne sont pas associe à des blocs de données, les paramètres effectifs ne sont pas sauvegardés automatiquement.

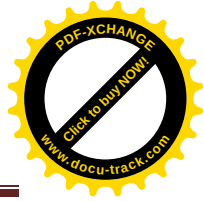
De plus il existe les blocs fonctionnels système (SFB) et les fonctions système (SFC), qui sont des fonctions préprogrammés. Ils peuvent être appelés à partir du programme utilisateur. On trouve des fonctions système pour la copie de blocs de données, le contrôle du programme utilisateur, la gestion des alarmes horaires et temporisées…

II.6.4.3 Bloc de données

Les blocs de données (DB) servent à l'enregistrement de données utilisateur. Les blocs de données globaux servent à l'enregistrement de données qui peuvent être utilisées par les autres blocs. Les blocs de données d'instances sont affectés à des blocs fonctionnels **[12]**.

Les différents blocs cités ci-dessus peuvent être édités avec l'application 'CONT LIST LOG'.





II.6.5 Programmation des blocs

La programmation des blocs de codes peut se faire à l'aide de trois applications :

1. CONT LIST LOG : elle permet de programmer des blocs d'organisations 'OB', des blocs fonctionnels 'FB' et des fonctions 'FC'.

1. GRAPH : elle permet de programmer des blocs fonctionnels 'FB'.

2. SCL : elle permet de créer des sources de code. Une source de code est un fichier texte, qui contient une suite d'instructions, une fois compilé il peut être transférer dans la CPU.

On peut trouver dans un même fichier source tout le programme utilisateur, c'est-à-dire les blocs d'organisations, les blocs fonctionnels et les fonctions.

Nota : les deux dernières applications GRAPH et SCL n'existent pas dans la version de base du STEP7, il faut les installées séparément.

On va présenter comment éditer les Blocs de code avec ces trois applications.

II.6.5.1 CONT LIST LOG

Un système de contrôle de qualité d'un certain produit fonction comme suit : L'élément X se déplace sur un tapis roulant, un premier capteur 'Cap1' vérifier sa hauteur, s'il est supérieur à une certaine hauteur h0, l'élément est éjecté en dehors du tapis grâce à un piston Pi1, sinon il poursuit son parcours vers un autre capteur 'Cap2' qui vérifier sa largeur, si elle n'est pas conforme à une certaine consigne, un piston Pi2 le pousse vers un casier en dehors du tapis.

Un front montant dans le capteur 1 (capteur 2) déclenche le piston 1 (piston 2).

On va éditer le bloc d'organisation 1, pour contrôler la gestion de ce système.

Dans SIMATIC Manager, cliquer sur le dossier programmes, ensuite sur Blocs, double clique sur l'icône du l'OB1ouvre l'application 'CONT LIST LOG'.

PDF-XCHANGE Click to buy NOW! www.docu-track.com

PDF-XCHANGE Click to buy NOW! www.docu-track.com

Choisissons par exemple le langage CONT (**figure II.16**):

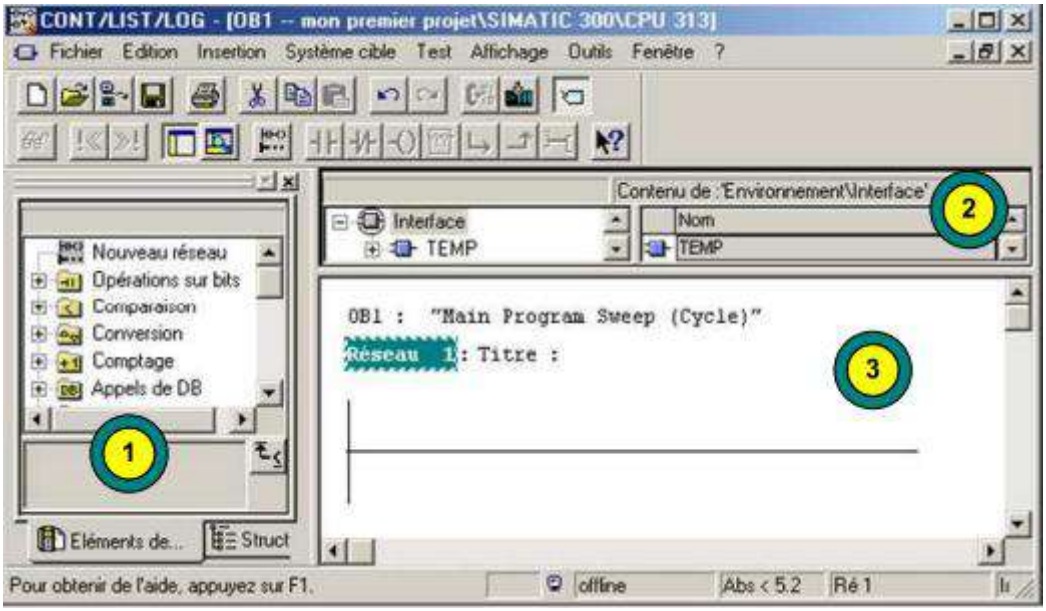


Figure II.16 CONT LIST LOG

On peut distinguer trois Zones principales **[9]** :

1. Le catalogue : c'est la bibliothèque des opérations et instructions du langage choisit (ici CONT), ces différentes opérations sont regroupées par famille : opération sur les bits, sur les mots, sur les entiers… pour utilisé une opération il suffit de faire glisser l'élément vers zone souhaitée.

2. La zone de déclarations : dans cette zone on définit les variables locales, les variables d'entrées et sorties ou arguments pour les fonctions et les blocs fonctionnelles.

3. La zone d'édition : elle est structurée en réseaux, chaque réseau finit par une affectation ou un saut.

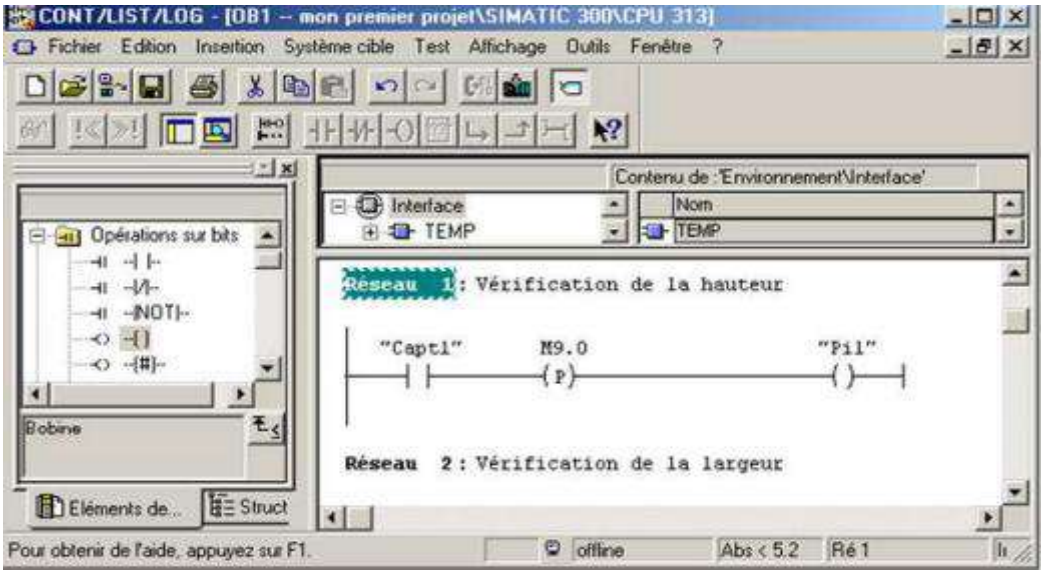


Figure II.17 exemple d'un programme en **CONT**

Pour notre exemple, on a 2 réseaux seulement. La détection du front montant se fait à l'aide de l'opération –(P)– . Elle nécessite un bit pour sauvegarder l'état des opérations qui la précède, au prochain cycle, elle compare ce bit avec le nouvel état de ces opérations.

Il est très utile d'affecter des titres aux réseaux ainsi que des commentaires. Sa permet une détection plus rapide des erreurs et permet d'écrire des programmes plus clair (en programmant, il faut penser toujours que ce n'est pas nous qui va le lire).

II.6.5.2 SCL

Le SCL est un langage de haut niveau, il est très utile pour la programmation de blocs de code complexes. Contrairement aux autres langages, le SCL peut contenir dans un seul fichier source tout le programme utilisateur (blocs d'organisation, fonctions et bloc fonctionnelles), il suffit de déclarer chaque bloc en respectant l'ordre d'appel, c'est-à-dire que si un bloc X appel un autre bloc Y, ce dernier doit être déclaré en premier dans le fichier source.

Comme application de démonstration, on se propose de calculer le gain en décibels d'un circuit électrique donné : on envoi a l'entrée du système une tension via le module de sorties analogique, ensuite on récupère la tension de sortie du système via le module d'entrées de l'automate. Il suffit après de calculer .

Dans SIMATIC Manager, cliquer su 'Programmes' ensuite sur source et créer un nouveau fichier source SCL. Double clique sur le nouveau objet inséré ouvre l'application d'édition.

PDF-XCHANGE Click to buy NOW! www.docu-track.com

PDF-XCHANGE Click to buy NOW! www.docu-track.com

Pour ajouter un bloc d'organisation, cliquer sur 'insérer', 'model de bloc' ensuite sur OB, l'application génère automatiquement quelque lignes de code.

Remplacer les xxx par le numéro du bloc, il faut faire attention de respecter la numérotation tel qu'elle est définit par la classe des blocs d'organisation.

Compléter le bloc, sauvegarder et compiler, ceci génère les blocs ajouter dans le dossier 'Bloc' de l'application SIMATIC Manager **[9]**.

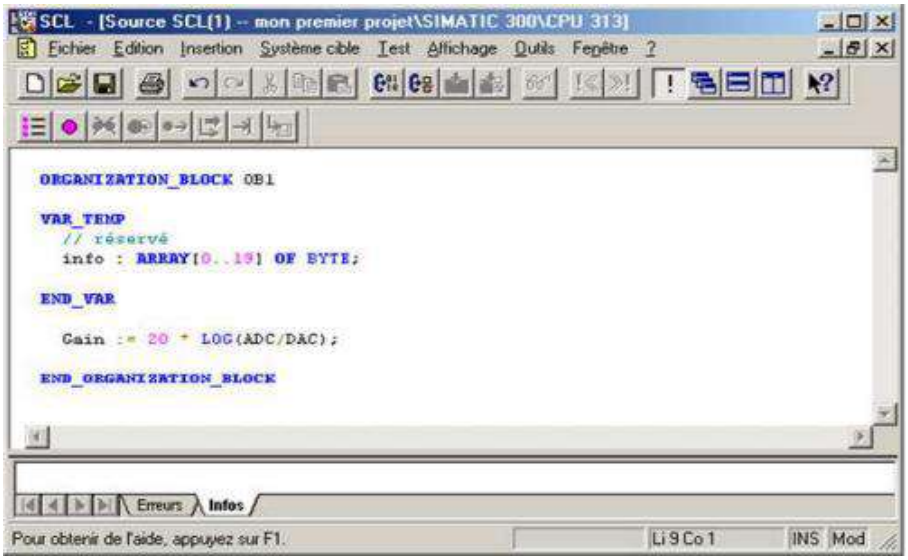


Figure II.18 exemple d'un programme en **SCL**

Bien sûre, il faut déclarer les variables Gain, ADC et DAC dans l'éditeur des mnémoniques.

II.6.5.3 S7-GRAPH

Cette application permet l'édition des blocs fonctionnels, en effet elle ne permet pas d'éditer des fonctions car elles n'ont pas des blocs de données associes pour mémorisé l'état des différentes transitions et étapes **[9]**.

On va illustrer l'édition avec S7-GRAPH, par cet exemple :

Un chariot de transport **(figure II.19)** dans une mine se trouve à l'état initiale au point A, en appuyant sur le

bouton poussoir m on démarre le moteur M1, arrivé au point B on démarre M2 jusqu'au point C, on arrête M1 et on continu avec M2 jusqu'au point D, on arrête le chariot.



Figure II.19 Commande d'un chariot

Pour satisfaire ce cahier de charge, on commence par créer un bloc fonctionnel et un autre de donnée d'instance dans le dossier Blocs du SIMTAIC Manager.

On choisit le GRAPH comme langage de programmation et on lance l'application.

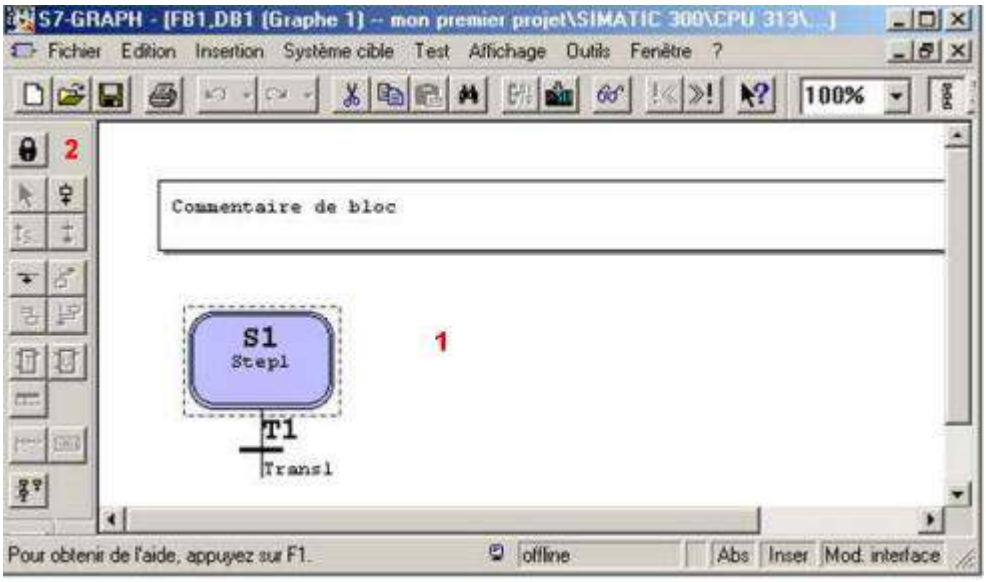


Figure II.20 Edition d'un bloc fonctionnel avec **S7-GRAPH**

On remarque deux parties, la première est la zone d'édition, la deuxième inclus les éléments à ajouter (des transitions, des branchements, des étapes…).

Pour ajouter un élément à la zone d'édition, il suffit de cliquer sur l'emplacement souhaité puis sur l'élément.

PDF-XCHANGE Click to buy NOW! www.docu-track.com

Notre exemple de commande de chariot comprend cinq étapes et quatre transitions. On commence par les crées **(figure II.21).**

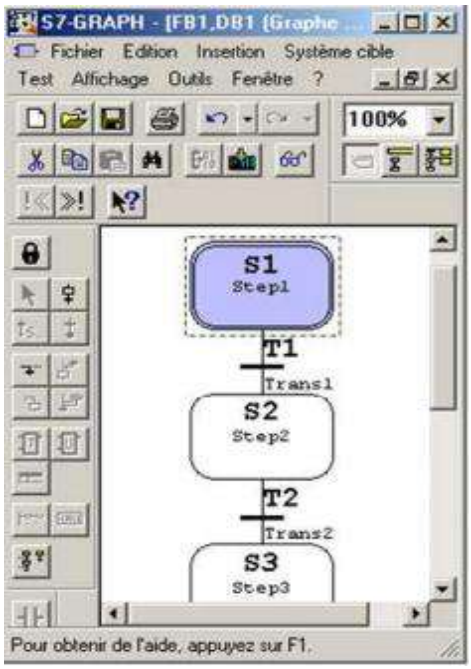


Figure II.21 Edition du **GRAPH**

Ensuite pour l'édition des actions et des transitions on change l'affichage de 'Graph' à 'Étape'.

Pour ajouter une action, cliquer par le bouton droit de la souris sur la colonne à droite et choisissez 'insérer un nouvel élément' puis sur 'action'. Il existe différent type d'action, le tableau suivant présente les plus utilisées :

Opération	opérande	Signification
N	A, E, M ou D	Tant que l'étape est active l'opérande est à 1.
S	A, E, M ou D	SET : Dès que l'étape est active, l'opérande est mis à 1.
R	A, E, M ou D	RESET : Dès que l'étape est active, l'opérande est mis à 0.
D	A, E, M ou D	Delay : n secondes après l'activation de l'étape, l'opérande se mis à 1 pour la durée de l'activation.
L	A, E, M ou D	Impulsion : quand l'étape est active, l'opérande se mis à 1 pendant n secondes

CALL	FB, FC, SFB ou SFC	Appel de bloc : tant que l'étape est active, le bloc spécifié est appelé.

Tableau II.2 Les action dans le S7-GRAPH

Pour les transitions, elles peuvent être présentées en langage CONT ou LOG, cliquer sur la colonne à gauche et insérer des éléments (contacts…) depuis le catalogue.

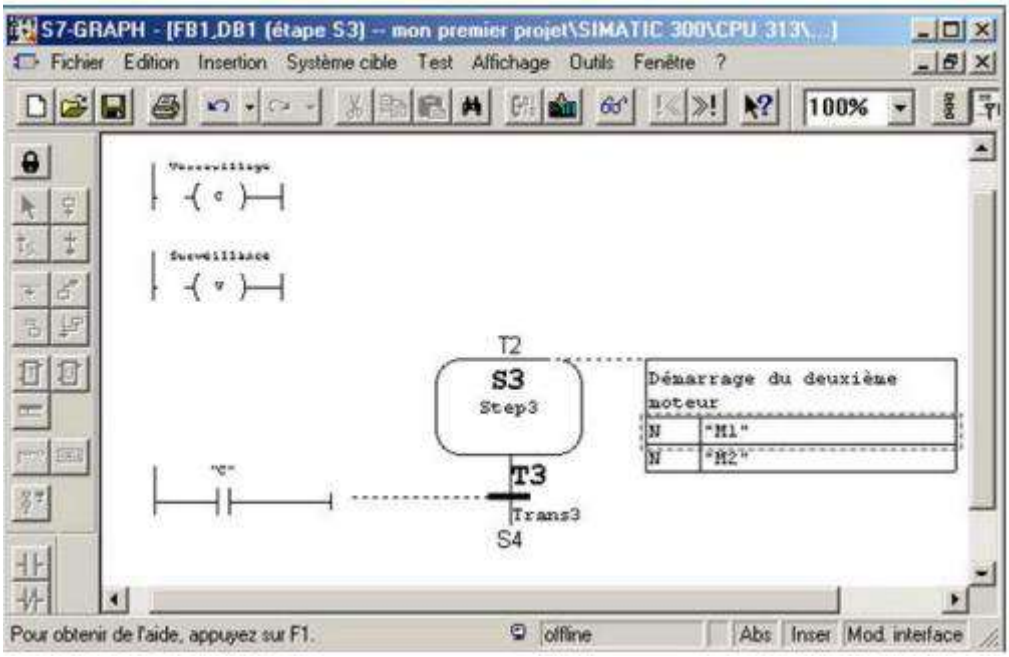


Figure II.22 Edition des actions & transitions

En haut à gauche, on remarque deux autres éléments :

1. Verrouillage : l'Interlock, c'est une condition programmable servant à verrouiller une étape et influençant l'exécution de certaines actions.

2. Surveillance : Supervision, c'est une condition programmable servant à surveiller une étape et influençant l'évolution du graphe entre cette étape et la suivante.

II.6.5.4 Avantages De L'utilisation Du S7- Graph

- Avec le langage de programmation S7-GRAPH, l'étendue des fonctions de STEP7 va être élargie à une programmation graphique pour l'enchaînement des commandes.

- Avec S7-GRAPH, vous pouvez programmer l'enchaînement des commandes clairement et rapidement.

PDF-XCHANGE Click to buy NOW! www.docu-track.com

PDF-XCHANGE Click to buy NOW! www.docu-track.com

- Le processus sera pour cela décomposé en unités et représenté graphiquement. Dans les unités, les actions à accomplir seront définies.

- Les conditions de la prochaine étape (réceptivités) peuvent être programmées en CONT ou en LOG.

Le langage de programmation S7-GRAPH correspond à la norme DIN EN 61131-3 (IEC 61131-3) définissant le langage SFC « Sequential Function Chart ».

II.6.6 Simulation de modules

L'application de simulation de modules S7-PLCSIM permet l'exécution et le teste du programme utilisateur destinés aux CPU S7-300 et aux CPU S7-400, ainsi qu'à WinLC.

La simulation étant complètement réalisée au sein du logiciel STEP 7 (**figure II.23**), il n'est pas nécessaire qu'une liaison soit établie avec un matériel S7 quelconque. Lorsque S7-PLCSIM s'exécute, toute nouvelle liaison est automatiquement dirigée vers la CPU de simulation.

S7-PLCSIM dispose d'une interface simple permettant de visualiser et de forcer les différents paramètres utilisés par le programme (par exemple, d'activer ou de désactiver des entrées).

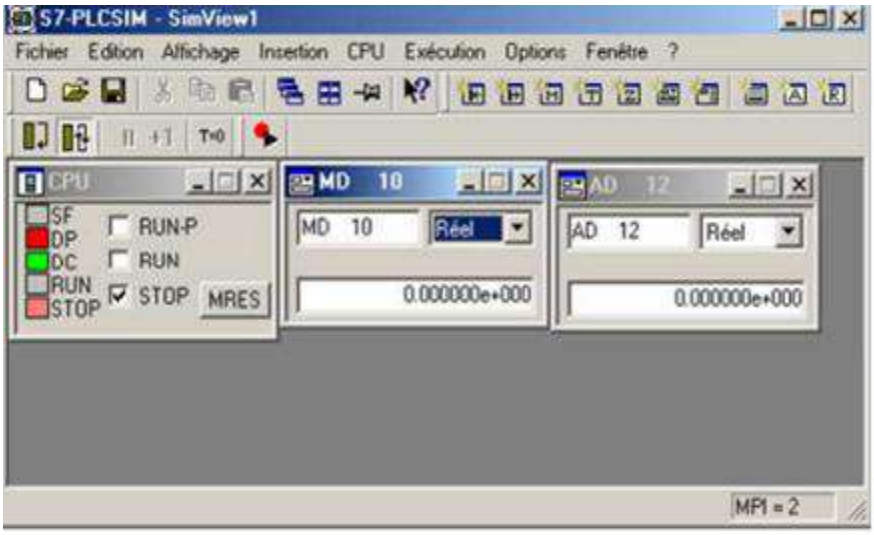
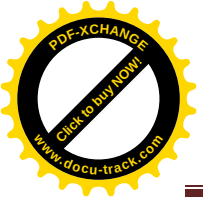
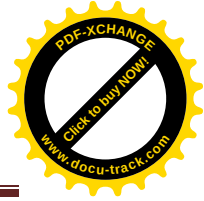


Figure II.23 Simulation de modules

En outre, S7-PLCSIM possède les fonctions suivantes :

- On peut créer des "fenêtres" dans lesquelles on a la possibilité d'accéder aux zones de mémoire d'entrée et de sortie, aux accumulateurs ainsi qu'aux registres de la CPU de simulation. On peut également accéder à la mémoire par adressage symbolique (il faut juste charger la table des mnémoniques dans 'options', puis sur 'outils' 'insérer mnémoniques').

- On peut sélectionner l'exécution automatique des temporisations ou encore les définir et les réinitialiser manuellement.

- On a la possibilité de changer l'état de fonctionnement de la CPU (STOP, RUN et RUN-P) comme pour une CPU réelle. De plus, on dispose d'une fonction de pause qui permet d'interrompre momentanément la CPU, sans affecter l'état du programme.

Bien que l'AP de simulation soit essentiellement logiciel, STEP 7 le considère comme une réelle composante matérielle, à quelques différences près :

- Contrairement à ce qui se passe avec une CPU réelle lors de la mise à l'arrêt de la CPU, l'état des sorties ne change pas.

- La CPU n'attend pas le début ou la fin du cycle pour actualiser une donnée qu'on a modifiée. Toute modification dans une fenêtre entraîne l'actualisation immédiate du contenu de l'adresse en mémoire.

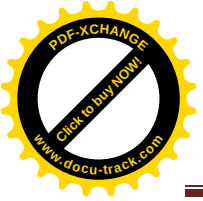
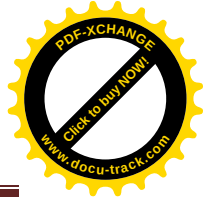
- Les options d'exécution permettent de choisir le mode d'exécution du programme par la CPU :

    - La commande Cycle unique exécute un cycle du programme, puis attend qu'on démarre l'exécution du cycle suivant.

    - La commande Cycle continu exécute le programme de la même manière que dans un AP réel : elle démarre un nouveau cycle aussitôt que le cycle précédent est terminé.

- On peut déclencher manuellement les OB d'alarme (aller dans 'exécution' puis 'déclenchement OB Erreur').

- Les modules fonctionnels (FM) ne sont pas pris en charge.

- La communication d'égal à égal n'est pas possible.

On va essayer le fonctionnement du S7-PLCSIM avec un petit exemple. Dans le dossier Programmes du SIMATIC Manager, on ajoute un bloc, par exemple une fonction

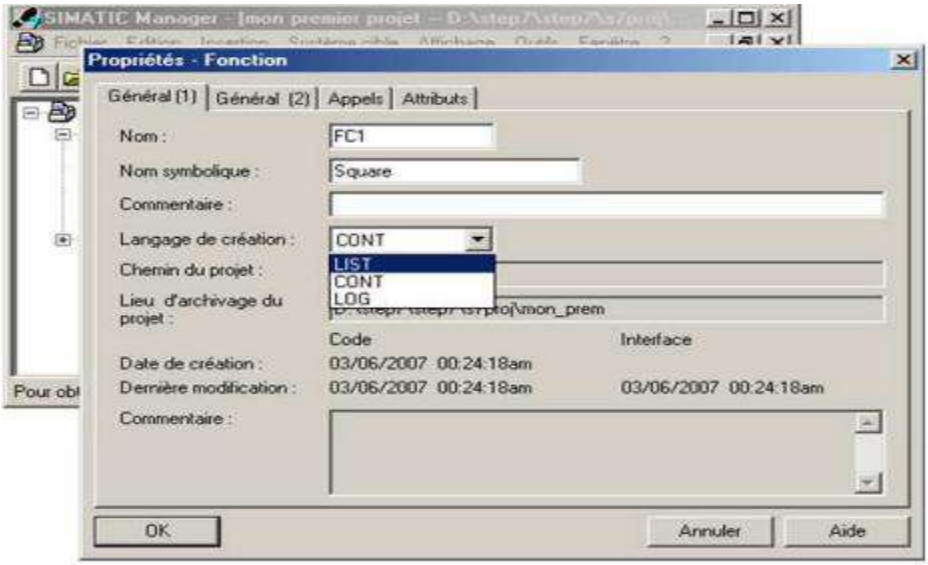


Figure II.24 Création d'une fonction

On donne à la fonction un nom symbolique, on choisit un langage d'édition parmi les trois proposés et on valide. Par exemple on va réaliser la fonction carrée, celle-ci nécessite deux arguments, un en entrée (x) et un autre en sortie (y) **figure II.25**.

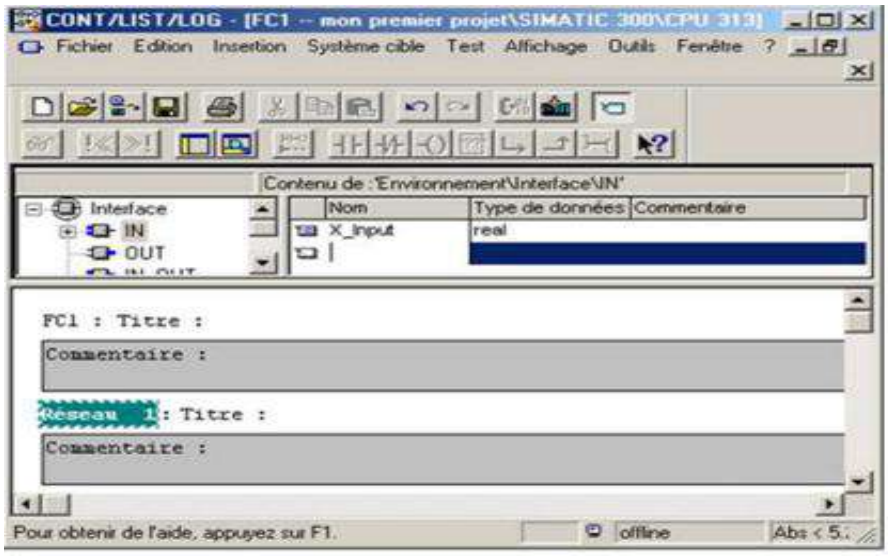
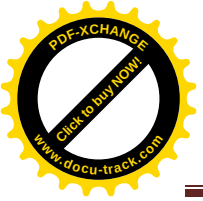
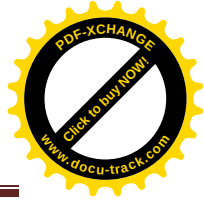


Figure II.25 Définition des arguments d'une fonction

On édit le code de cette fonction et on sauvegarde **figure II.26**.

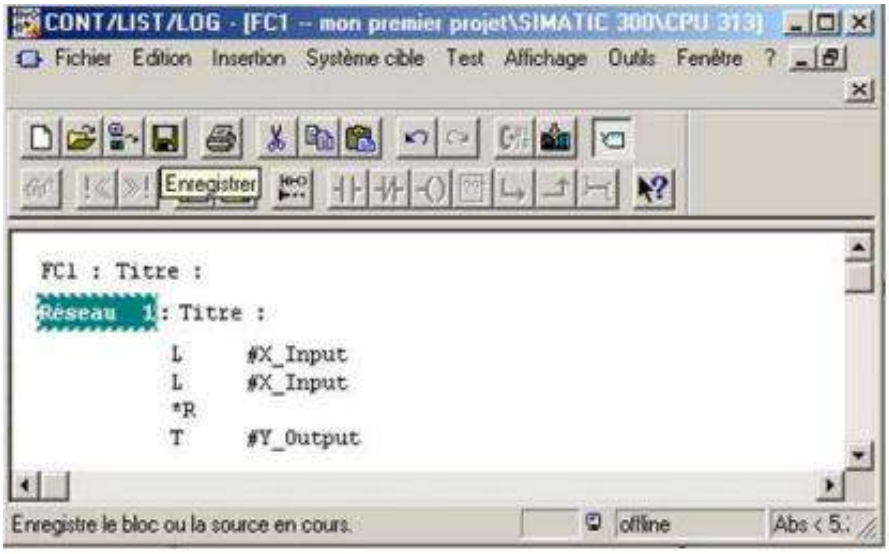


figure II.26 Edition du code en LIST

On revient sur SIMATIC Manager, on ouvre l'OB1, on change le langage d'édition dans le menu supérieur 'Affichage' puis sur 'CONT'. Maintenant sur le catalogue à gauche on appel la fonction que nous venant de créer en lui affectant les variables d'E/S **figure II.27**.

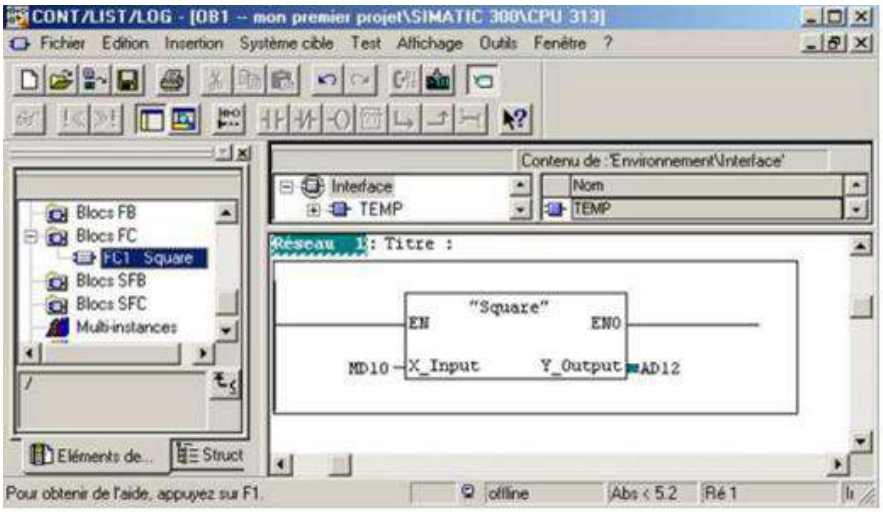


Figure II.27 Edition du code CONT

Dans cette exemple, le bloc d'organisation Cyclique OB1 appel la fonction FC1 qui va affecter au double mot de la sortie analogique 'AD12' le carrée du double mot contenu dans

le mémento 'MD10'. Le contenu de ces adresses mémoires va être interprété comme nombre réel.

On lance l'application S7-PLCSIM (aller dans 'Options' puis cliquer sur 'Simulation des modules') et on charge les blocs de code dans la CPU.

Dans le menu insertion, on ajoute le mémento 'MD10' et le double mot de la sortie analogique 'AD12', il faut sélectionner Réel comme type sinon on se retrouve avec des résultats erronés.

On démarre la simulation en cochant la case RUN, ensuite on introduit différents valeurs dans le mémento MD10 et on vérifié la valeur correspondante dans le double mot de la sortie analogique AD12 **figure II.28**.

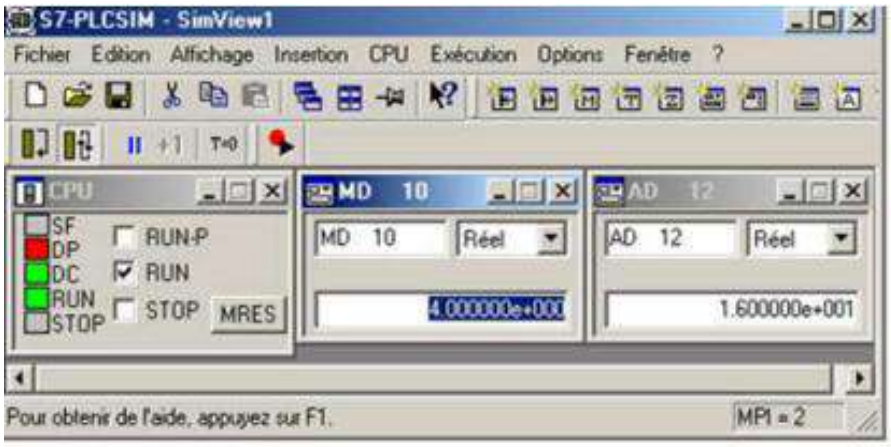


Figure II.28 Simulation di programme

II.7 WIN CC Flexible

II.7.1 Introduction

Suite à l'automatisation industrielle, l'opérateur humain a été contraint de conduire ou de superviser des machines automatisées, en réduisant les prises d'information et les actions directes sur le processus, ce qui conduit à l'élaboration d'interface d'interaction Homme/Machine, flexible et aussi lisible pour un simple opérateur. Le dialogue est d'autant plus facilité que l'écran comporte des images avec des synoptiques, des graphes, des paragraphes…etc.

Siemens avec sa gamme SIMATIC C7, contribue dans ce sens, et met à la disposition de cette demande exigeante, un grand choix en terme de pupitre de supervision et de contrôle, paramétrable par le biais d'outils divers.

Le progiciel Win CC flexible, reste le plus approprié, en terme de simplicité, pour la conception ce type d'interface de commande/diagnostic.

II.7.2 Progiciel de conception et configuration d'interface Win CC flexible

II.7.2.1 Présentation générale

Win CC flexible, est un progiciel partagé dans l'environnement STEP7, et propose pour la configuration de divers pupitres opérateurs, une famille de systèmes d'ingénierie évolutifs adaptés aux tâches de configuration.

Créer l'interface graphique et les variables, c'est pouvoir lire les valeurs du processus via l'automate, les afficher pour que l'opérateur puisse les interpréter, et ajuster, éventuellement, le processus, toujours via l'automate **Figure II.29**.

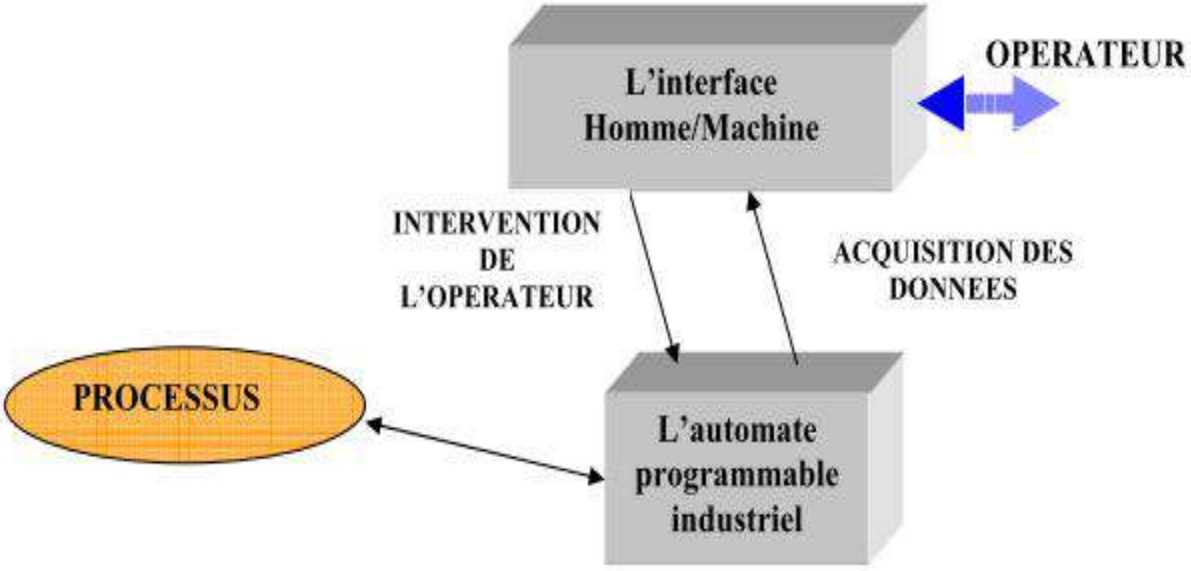


Figure II.29 L'interface Homme/Machine dans un processus automatisé

II.7.2.2 Elément du progiciel Win CC flexible

L'utilisation de ce logiciel pour la conception d'une interface, passe par plusieurs étapes résumées dans la **Figure II.30** :

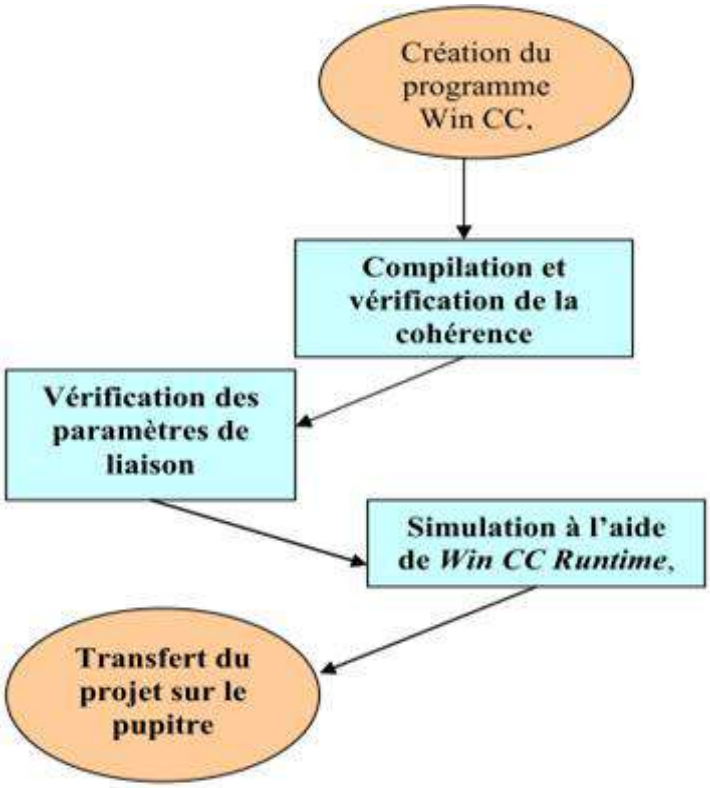


Figure II.30 Etapes pour la conception d'une interface via Win CC flexible

On commence par l'environnement de cet outil, pour terminer expliquer les différentes étapes dans les détails précédemment énumérés.

L'environnement de travail de **Win CC flexible** se compose de plusieurs éléments.

Certains de ces éléments sont liés à des éditeurs particuliers, visibles uniquement lorsque et éditeur est activé. Il met à disposition un éditeur spécifique pour chaque tâche de configuration.

On peut configurer par ex. l'interface utilisateur graphique d'un pupitre opérateur avec l'éditeur "Vues". Pour la configuration d'alarmes, on peut utiliser p. ex. l'éditeur "Alarmes TOR" ou autre.

PDF-XCHANGE Click to buy NOW! www.docu-track.com

PDF-XCHANGE Click to buy NOW! www.docu-track.com

Dans **Win CC flexible**, chaque projet créé contient principalement des vues que l'on crée pour le contrôle-commande de machines et d'installations. Lors de la création des vues, vous disposez d'objets prédéfinis permettant d'afficher des procédures et de définir des valeurs du processus.

Les différents outils et barres de l'éditeur de vues sont représentés dans la figure qui suit **Figure II.31** :

- **Barre des menus**

La barre des menus contient toutes les commandes nécessaires à l'utilisation de Win CC flexible. Les raccourcis disponibles sont indiqués en regard de la commande du menu.

- **Barres d'outils**

La barre d'outils permet d'afficher tout dont le programmeur a besoin.

- **Zone de travail**

La zone de travail sert à configurer des vues, de façon qu'il soit le plus compréhensible par l'utilisateur, et très facile à manipuler et consulter les résultats.

- **Boîte à outils**

La fenêtre des outils propose un choix d'objets simples ou complexes qu'on insère dans les vues, p. ex. des objets graphiques et éléments de commande.

- **Fenêtre des propriétés**

Le contenu de la fenêtre des propriétés dépend de la sélection actuelle dans la zone de travail, lorsqu'un objet est sélectionné, on peut éditer les propriétés de l'objet en question dans la fenêtre des propriétés.

PDF-XCHANGE Click to buy NOW! www.docu-track.com

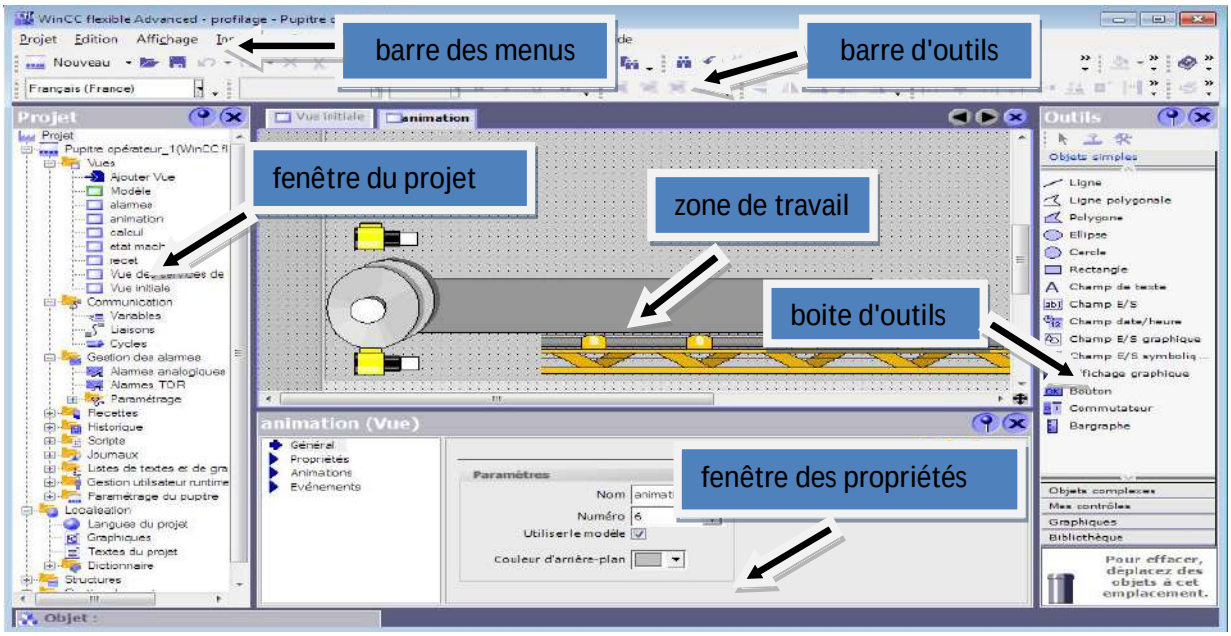


Figure II.31 Vue d'ensemble du progiciel Win CC flexible

II.7.2.3 Utilisation des variables

On distingue deux types de variables, les variables externes et les variables internes Les variables externes permettent de communiquer, c.-à-d. d'échanger des données entre les composants d'un processus automatisé, entre un pupitre opérateur et un automate par exemple.

Une variable externe est l'image d'une cellule mémoire définie de l'automate. L'accès en lecture et en écriture à cette cellule mémoire est possible aussi bien à partir du pupitre opérateur que de l'automate.

Les variables internes ne possèdent aucun lien avec l'automate. Elles sont enregistrées dans la mémoire du pupitre. Le pupitre opérateur en question peut donc accéder en lecture et en écriture aux variables internes. Les variables internes sont créées p. ex. pour exécuter des calculs locaux.

La déclaration des variables s'effectue dans l'éditeur "Variables" **Figure II.32**. Lors de leur déclaration, une configuration de base est attribuée aux variables. L'éditeur "Variables" vous permet d'adapter la configuration des variables aux besoins de votre projet ; L'éditeur s'ouvre systématiquement par ajout de variable.

PDF-XCHANGE Click to buy NOW! www.docu-track.com

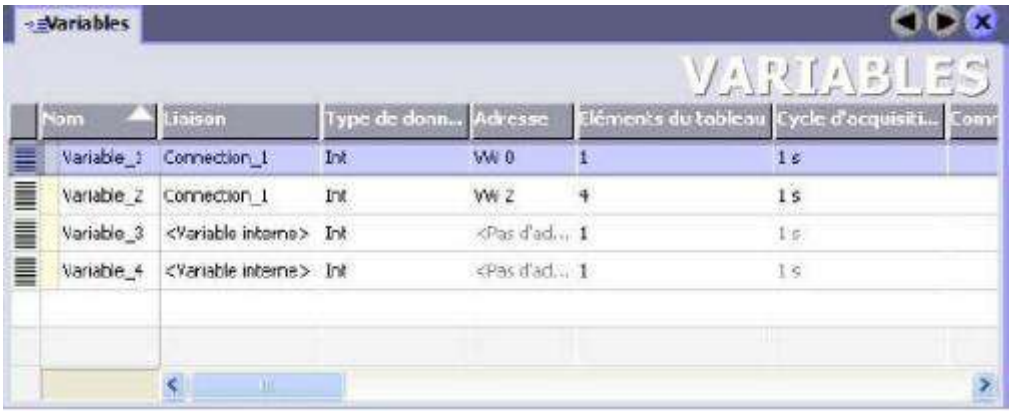


Figure II.32 éditeur de variable dans le Win CC flexible

Chaque variable peut être configurée selon le besoin et cela en accédant à sa fenêtre de priorité après l'avoir sectionnée.

II.7.2.4 Compilation et Simulation

Après avoir créé le projet et terminé la configuration, il est indispensable de vérifier la cohérence du projet, de contrôler la cohérence et de chercher les erreurs, à l'aide de la commande sur la barre du menu « contrôle de la cohérence ». Après le contrôle de cohérence, le système crée un fichier de projet compilé.

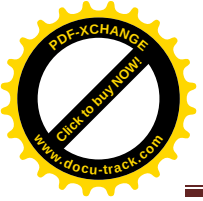
La simulation permet de détecter des erreurs logiques de configuration, par exemple, des valeurs limites incorrectes, et cela à l'aide du simulateur Runtime par la commande « démarrer le système Runtime du simulateur ».

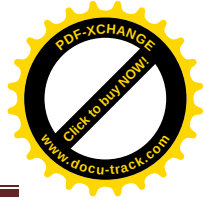
II.7.3 Conclusion

On constate la facilité et la souplesse qu'offre L'A.P.I pour sa programmation, connexion, et adaptation aux conditions industrielles, avec toutes les fonctionnalités indispensables à l'automatisation des processus. La diversité des possibilités, mise en œuvre et son coût, le rendent incontournable lors de l'élaboration d'une solution.

Mais il ne faut pas oublier qu'il est nécessaire d'avoir une bonne analyse du problème à résoudre tout en assurant le respect des règles d'installation.

La description précédente de l'outil Win CC flexible, prépare le terrain en vue de l'application «Réalisation d'un interface IHM d'une profileuse à base d'automates





programmables SIEMENS », et pendant laquelle une mise en œuvre complète sera décrite pour élaborer une interface Homme/machine.

PDF-XCHANGE Click to buy NOW! www.docu-track.com
PDF-XCHANGE Click to buy NOW! www.docu-track.com

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.

L'automate programmable industriel est l'organe principal de contrôle placée dans un procédé industriel, en vue de le contrôler. Il a pour tâche principale, la récolte des informations relatives à l'état du système, à partir des différents capteurs via ses interfaces d'entrées, et les traiter pour prendre une décision ; et ainsi commander les actionneurs via ses interfaces de sorties suivant une logique de fonctionnement mise en évidence, par un programme inscrit dans la mémoire.

Le but de ce travail est l'élaboration d'une solution à base d'API pour automatiser une machine de production industrielle et de réaliser une interface homme-machine sur un écran tactile.

Le présent travail s'articule autour de trois parties. La première partie présente une description du processus industriel effectué, les différents constituants de la machine, les différentes phases de production et la méthode de contrôle utilisée.

Dans la deuxième, 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, ainsi que l'interface de programmation WINCC flexible.

Quant à la troisième et dernière partie, elle sera consacrée à l'élaboration du programme de commande de la machine profileuse avec tous les détails utiles.

Enfin, on terminera l'étude par une conclusion générale qui discutera les avantages apportés et les perspectives visées en termes de réalisation et installation.

PDF-XCHANGE Click to buy NOW! www.docu-track.com

Conclusion générale

Au cours de notre étude, et en vue de remplacer le dispositif de contrôle de la machine profileuse au sein de laboratoire de mécanique, nous avons procédé par une étude descriptive du processus et de la machine ainsi que tous les accessoires complémentaires.

Nous avons jugé intéressant et commode de faire appel à la technologie programmée (Automate programmable Industriel), et ce pour des raisons de :

- Complexité de l'installation existante (logique électrique câblée),
- Diagnostic des pannes et entretien relativement difficiles,
- Problème de l'inflexibilité (installation de contrôle figée),
- Efficacité minime et autres problèmes économiques tels que : taux de productivité, coûts d'entretien…etc.

Ensuite, dans la deuxième partie, 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, ainsi que avons présenté une étude assez détaillée sur le SIMATIC S7-313C, qui a été notre choix pour l'application puis on a présenté une étude sur le WINCC flexible pour la commande de la machine.

Ainsi que les raisons et les motifs du choix qui se manifestent essentiellement dans :

- Performances techniques (fréquence de traitement, type de signaux traités … etc.),
- Capacité mémoire assez suffisante.
- Possibilité d'extension en terme du nombre élevé des entrées et des sorties du processus, exigé par certaines applications. Laquelle nous a permis d'intégrer trois autres modules de 32 entrées TOR et deux modules de 32 sorties TOR.
- Possibilité d'intégrer des modules spéciaux (modules d'entrées ou de sorties analogiques, régulateur de température, de pression, …etc.).
- Disposition du CPU 314 IFM de fonctions intégrés telles que : le fréquencemètre, les compteurs, …etc.

Quant à la troisième partie, elle a été consacré à la programmation graphique dite **GRAFCET** du processus de la machine profileuse et la programmation avec **WINCC** flexible. Nous avons présenté un programme englobant le contrôle de la machine à l'aide d'un panel tactile ou l'on peut visualiser et contrôles le processus.

Enfin, on pourrait dire que remplacer les dispositifs classiques de contrôle par les Automates Programmables Industriels révèle comme étant la solution qui subviendrait aux besoins du monde industriel futur voire même actuel qui ne cesse de trop exigé pour ce qui est de qualité et quantité des produits.

PDF-XCHANGE Click to buy NOW! www.docu-track.com

ANNEXE

1. La CPU 313C
2. Le terminal d'exploitation TP177B 6“ mono DP
3. Configuration matérielle (Partie Hardware)
4. Transfert sur les pupitres opérateur TP 177B, TP 177A
5. Chargement dans le système cible à partir de la PG/PC
6-Le chois de la vitesse de transmission

A.1. La CPU 313C :

Sur la plus part des CPU 300, on peut remarquer les éléments suivants: des LEDs, un Commutateur, une pile, une carte mémoire et une interface MPI.

Commutateur du mode de fonctionnement :

Position	Signification	Explications
RUN-P	RUN-PROGRAM	La CPU traite le programme utilisateur. La clé ne peut être retirée dans cette position.
RUN	Mode de marche	La CPU traite le programme utilisateur. Ce dernier ne peut être modifie.
STOP	Mode d’arrêt	La CPU ne traite aucun programme utilisateur
MRES	Effacement général	Position instable du commutateur pour effacement général de la CPU.

Tableau A.1 Positions du commutateur du mode de fonctionnement

Les LEDs de visualisations :

-SF (rouge) : défaut de matériel ou de logiciel

-BATF (rouge) : défaillance de la pile

-5v cc(vert) :l’alimentation 5v cc et bus s7-300 est correct

-FRCE (jaune) :le forçage permanât est actif

-Run (vert) : cpu en run :la LED clignote a 1 Hz au démarrage ;a 0.5 Hz en mode d’attent.

PDF-XCHANGE Click to buy NOW! www.docu-track.com

-Stop (jaune) : cpu en stop ou ATTENT ou en démarrage ;la LED clignote en cas demande d'effacement générale

Les caractéristiques techniques de CPU313C :

-Mémoire de travail 128 Ko; 0,07ms/kinst;

- DI24/DO16; AI5/AO2 intégrées; 3 sorties impulsion (2,5kHz);

-comptage et mesure sur 3 voies avec codeurs incrémentaux 24V (30kHz);

- port MPI;

-configuration multi-rangées jusqu'à 31 modules; firmware V3.3

- N° série 6ES7 313-5BG04-0AB0

A.2. Le terminal d'exploitation TP177B 6“ mono DP

Les TP 177A, TP 177B et panneaux 177B OP offrent commissionner vite des périodes, grand utilisateur de la mémoire et le rendement élevé， , et eux sont optimise pour des projets base sur WinCC flexible

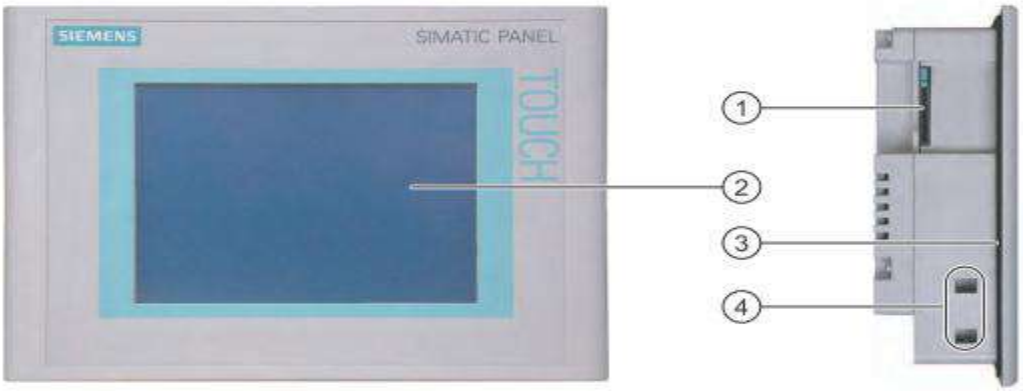


Figure 1 le pupitre TP177B 6“ mono DP

1-Fente pour une carte de MultiMedia

2-cran d'affichage (tactile)

3- Joint de support

4- Cavité de bride de support

PDF-XCHANGE Click to buy NOW! www.docu-track.com

PDF-XCHANGE Click to buy NOW! www.docu-track.com

Vue arrière :



Figure 2 vue arrière du pupitre

1-Fente pour une carte de MultiMedia

2-étiquette d'évaluation

3-Contact DIP

4-Nom d'interface

Contrôleurs de Siemens

La table suivante montre les contrôleurs de Siemens et les protocoles ou les profils qui peuvent être employé

Controller	Protocol/Profile	TP 177B 4" PN/DP	TP 177B 6" DP	TP 177B 6" PN/DP	OP 177B DP	OP 177B PN/DP
SIMATIC S7-300/400	MPI	Yes	Yes	Yes	Yes	Yes
	PROFIBUS DP up to 12 Mbps	Yes	Yes	Yes	Yes	Yes
	PROFINET	Yes	No	Yes	No	Yes
SIMATIC S5	PROFIBUS DP up to 12 Mbps	Yes	Yes	Yes	Yes	Yes
SIMATIC S7-200	PPI	Yes	Yes	Yes	Yes	Yes
	MPI	Yes	Yes	Yes	Yes	Yes
	PROFIBUS DP CPU 215	Yes	Yes	Yes	Yes	Yes
	PROFIBUS DP standard	Yes	Yes	Yes	Yes	Yes
SIMATIC 500/505	NITP	Yes 1)	Yes 1)	Yes 1)	Yes	Yes
	PROFIBUS DP up to 12 Mbps	Yes	Yes	Yes	Yes	Yes

Tableau A.2 tableau de contrôleurs de SIEMENS

PDF-XCHANGE Click to buy NOW! www.docu-track.com

A.3.Configuration matérielle (Partie Hardware)

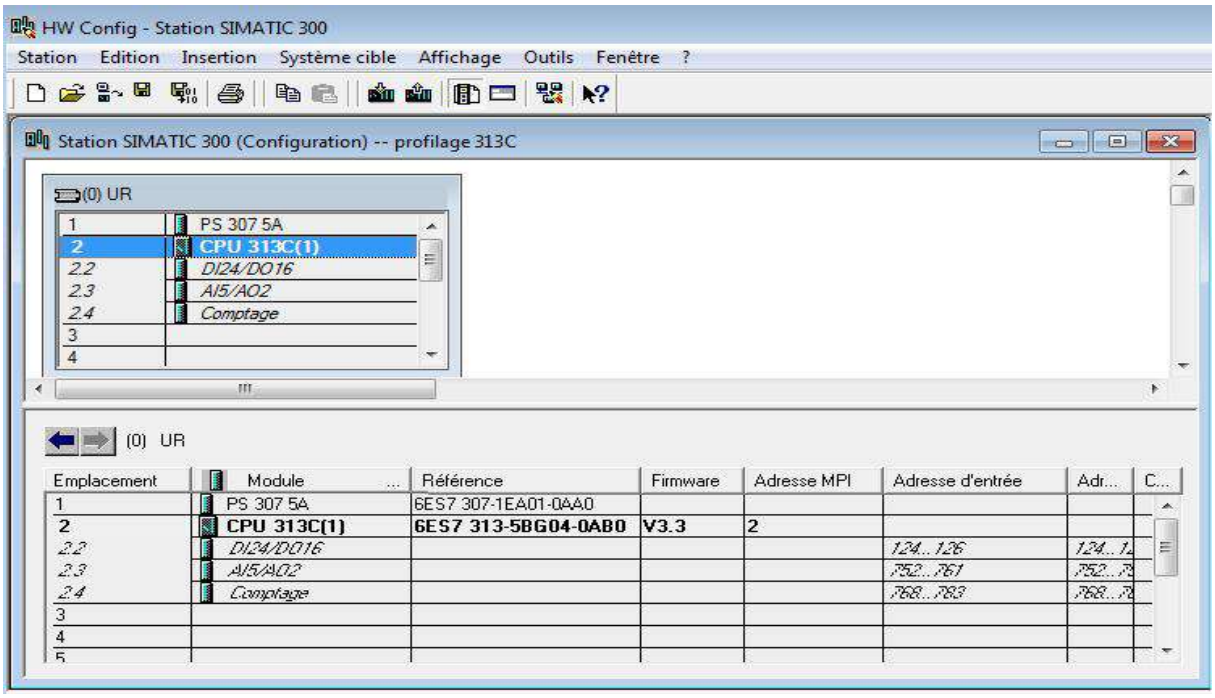


Figure 3 Configuration matérielle

A.4. Transfert sur les pupitres opérateur TP 177B, TP 177A

Pour le transfert via l'interface série du projet sur les pupitres opérateur OP 73micro, TP 170micro, TP 177micro, OP 73, OP 77A et TP 177A, vous avez besoin du câble "RS 232 / PPI Multi Master Câble" (6ES7 901-3CB30-0XA0). Le commutateur multiple 5 doit Impérativement être mis sur 0.

Seul un transfert en série est possible pour les pupitres opérateurs Micro. Sur les autres Pupitres opérateurs, le transfert MPI est également possible. En cas de modification du type de transfert dans les paramètres de transfert, l'autre voie concernée est désactivée.

PDF-XCHANGE Click to buy NOW! www.docu-track.com

A.5. Chargement dans le système cible à partir de la PG/PC

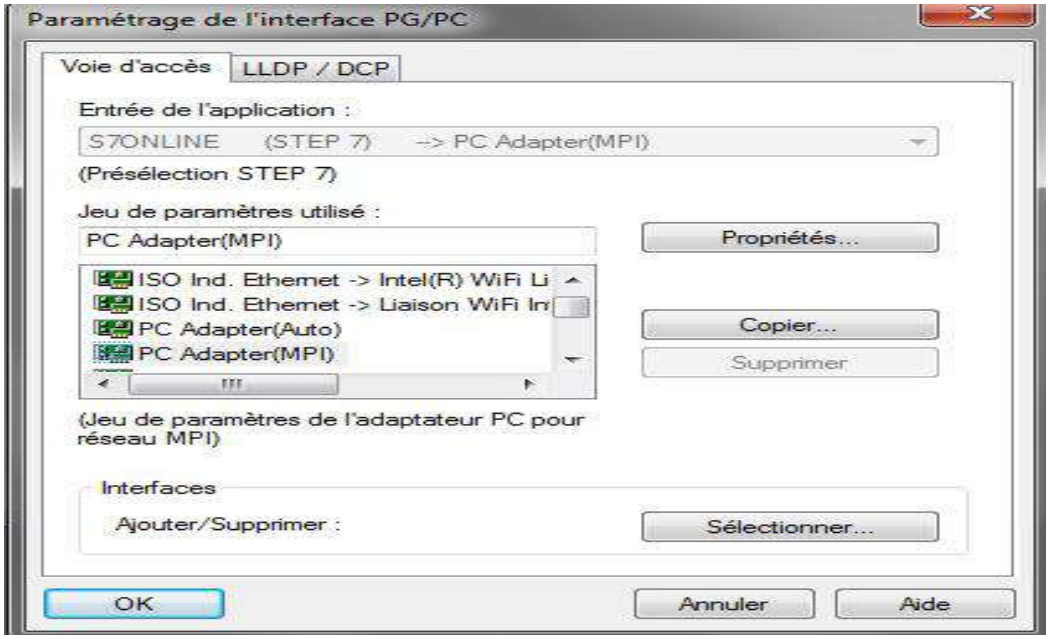


Figure 4 choix de PC Adapter (MPI)

Le chargement du programme dans le système cible se fait sous certaines conditions :

• Une liaison est établie entre la PG et la CPU du système cible (via l'interface MPI ou PROFIBUS).

• L'accès au système cible est possible.

• La compilation du programme a charger doit se faire sans erreur.

• La CPU doit se trouver dans un etat de fonctionnement autorisant le chargement (STOP ou RUN-P).

Si la syntaxe est correcte, le bloc est ensuite compile en code machine, enregistre et charge par la commande **Système cible** > **Charger**, ou en sélectionnant tous les blocs et charger par la fonction de chargement  dans la barre d'outils du SIMATIC Manager.

A.6. Le choix de la vitesse de transmission

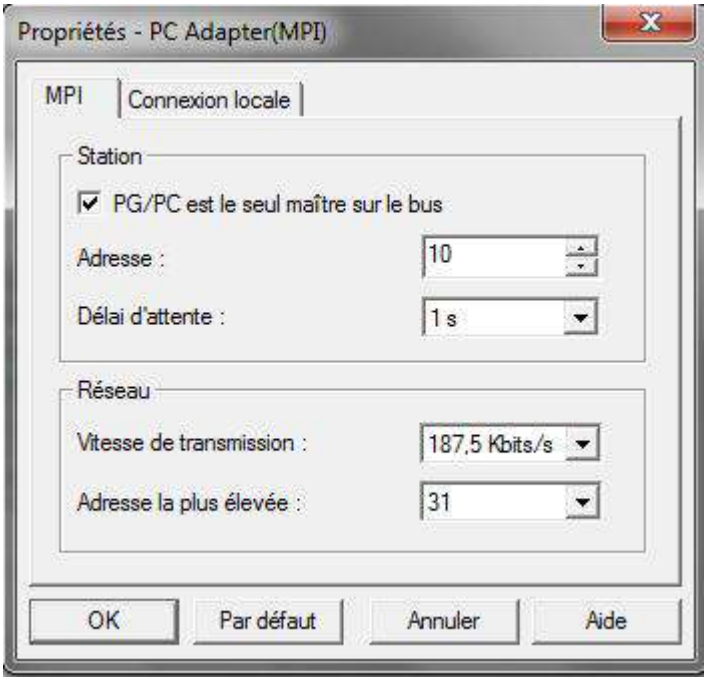


Figure 5 choix de la vitesse de transmission

PDF-XCHANGE Click to buy NOW! www.docu-track.com
PDF-XCHANGE Click to buy NOW! www.docu-track.com

Référence

[1] Site web : WWW.euro-profilage.com (05/05/2012)

[2] JEAN-JAQUES BERTRAND « profilage » BM 7574

[3] **E.M. BERKOUK** thème «Réalisation et gestion d'un prototype de station de pompage à base d'automates programmables industriels SIEMENS» , projet de fin d'étude, Ecole Nationale Polytechnique 2006.

[4] **R.BOUROUBI**, «Mise en œuvre du logiciel STEP 7, application à l'automate programmable S7-314 IFM ».

[5] **N.BOUAYED**, thème «La télégestion» , projet de fin d'étude, Ecole Nationale Polytechnique 2006.

[6] **E.BAJIC et B.BOUARD**, « Réseau PROFIBUS », Techniques de l'ingénieur,

[7] **G. PUJOLLES**, « Les réseaux », Edition Eyrolles, 1997.

[8] **M. BERTRAND**, « Automates programmables industriels », Techniques de l'ingénieur, Vol. S 8 015.

[9] **P.JARGOT**, « Langages de programmation pour API. », Techniques de l'ingénieur.

[10] **G. MICHEL**, « Les A.P.I Architecture et application des automates programmables industriels »

Manuals

[11] SIEMENS, «Appareils de terrain pour l'automatisation des processus».

[12] SIEMENS, «Programmation avec STEP 7 », SIMATIC.

[13] SIEMENS, «STEP 7, Régulation PID», SIMATIC.

PDF-XCHANGE Click to buy NOW! www.docu-track.com
PDF-XCHANGE Click to buy NOW! www.docu-track.com

ملخص:

العمل المنجز في المذكرة يتمحور حول برمجة وسيط إنسان آلة من أجل التحكم والمشاهدة والصيانة بالاعتماد على مسير صناعي مبرمج S7-300 **و** شاشة رقمية **"سيمنس"**. ومن أجل مشاهدة النتائج قمنا بتطبيق البرمجة على نموذج آلة تشكيل صفائح الحديد profileuse.

الهدف من هذا العمل هو إعطاء رؤية حول مستقبل هذه التكنولوجيا التي تعتمد على البرمجة والآليات لتسييرها مثل STEP7-S300

الكلمة المفتاح :

متمن , آلة تشكيل صفائح الحديد, وسيط إنسان آلة ,مسيرصناعي مبرمج, مشغل وحدة التحكم.

ABSTRACT:

The aim of this work is the programming of the human-machine interface for control, visualization and maintenance based on a programmable automat S7-300 and touch screen SIEMENS.

Key words:

GRAFCET, Rollformer, human-machine interface, programmable logic contrôle, panel.

RESUME:

Le but de ce travail est la programmation de l'interface homme-machine pour la commande, la visualisation et la maintenance à base d'un automate programmable S7-300 et écran tactile SIEMENS.

Mots clés : Grafcet, profileuse, Interface homme-machine, automate programmable, pupitre.