

Dédicace

Je dédie ce travail :

A mon tout, à la lumière de ma vie, à celle qui a su me donner amour et joie de vivre, à celle qui a toujours montré affection et compréhension à mon égard : à ma mère.

A l'homme qui a été toujours présent, à celui qui m'a offert tout son aide et son encouragement : à mon père.

A ceux qui m'ont donné joie et bonheur, à ceux qui m'ont soutenu en toutes circonstances à :

Mon cher frère et mes chères sœurs.

A ceux que j'ai la chance de les connaître, dans les meilleurs et pires moments durant tout mon cursus.

A mes amis et toute la famille.

A mon cher binôme Zekraoui Oussama et sa famille, et toute la promo d'automatique 2015.

Abderrezak

Dédicace

Je dédie ce modeste travail :

A mes parents :

Grâce à leurs tendres encouragements et leurs grands sacrifices, ils ont pu créer le climat affectueux et propice à la poursuite de mes études. Aucune dédicace ne pourrait exprimer mon respect, ma considération et mes profonds sentiments envers eux.

Je prie le bon Dieu de les bénir, de veiller sur eux, en espérant qu'ils seront toujours fiers de moi.

A mes frères.

A toute la famille Zekraoui

A mon binôme Boubekri Abderreraïk et sa famille

A tous mes professeurs :

Leur générosité et leur soutien m'oblige de leur témoigner mon profond respect et ma loyale considération.

A tous mes amis et mes collègues :

Qu'ils trouvent ici le témoignage de ma fidélité et d'une amitié infinie envers eux.

Oussama

Remerciements

Tout d'abord, nous remercions notre dieu tout puissant qui nous a armés de courage, de volonté, de patience et de la santé.

Au terme de ce modeste travail, nous tenons à remercier tous ceux qui ont contribué à l'élaboration de ce mémoire :

*En premier, toute notre reconnaissance à notre promotrice **Mme Chentir** maître de conférences et membre dans l'équipe de parole du laboratoire **LATSI** (laboratoire traitement du signal et imagerie) pour avoir assuré notre encadrement ainsi que son aide précieux.*

*Nos remerciements les plus spéciaux à notre **Mr KIAL** (encadreur au sein du **SNVI Rouïba**) pour tous ses encouragements et ses conseils prodigués, sa disponibilité et ses orientations tout au long de notre stage.*

Merci à l'ensemble des membres du jury qui nous font l'honneur d'évaluer ce modeste travail.

*Nous tenons à remercier également tous les enseignants de **SAAD DAHLAB**.*

Enfin, nous exprimons nos sincères gratitude à nos familles et nos amis qui ont partagé au quotidien nos espoirs et nos inquiétudes.

Résumé

Dans ce mémoire de fin d'étude, nous avons fait une étude complète sur l'adaptation d'un automate S7-300 sur une machine à dresser et à centrer. Pour cela, nous avons élaboré une modélisation de la machine à l'aide de GRAFCET. Suite à cette modélisation, nous avons fait une conversion GRAFCET-LADDER pour l'introduire dans le logiciel step7.

Enfin, nous avons fait appel au logiciel de supervision et de visualisations WinCC afin de faciliter la tâche aux opérateurs.

ملخص

في مذكرة التخرج لنهاية الدراسة قمنا بدراسة كاملة على تأقلم آلي من نوع S7--300 على آلة التثبيت و التمرکز و من أجل هذا قمنا بتمثيل هذه الآلة بالاستعانة ببرنامج GRAFCET .

وتباعا لهذا التمثيل قمنا بتحويل برنامج GRAFCET-LADDER وإدراجه في برنامج STEP7.

أخيرا تم التكامل مع برنامج المراقبة و التصور WINCC الذي يقوم بتسهيل عمل المشرف على الآلة.

Abstract

In this master's thesis, we made a comprehensive study of the adaptation of an S7-300 on a straightener and center machine.

For this, we developed a model of the machine using GRAFCET. Following this model, we made a GRAFCET-LADDER conversion to introduce it in the step7 software.

Finally, we used the supervision software WinCC visualisations which make it easier for operators.

Résumés	I
Liste des acronymes	II
Sommaire.....	IV
Liste des figures.....	VII
Introduction générale	01
Chapitre 1 : Généralités	
1.1 Introduction.....	02
1.2 Généralités sur les automates programmables.....	02
1.2.1 Définition	02
1.2.2 Architecture	03
1.2.2.1 Description des éléments d'un API	04
1.2.3 .La programmation des automates	05
1.2.4 L'automate programmable S5-100U	06
1.2.5 Choix d'un API	07
1.3 Description de la machine TEST M25.....	07
1.3.1 Introduction.....	07
1.3.2 Présentation de la MACHINTEST M25.....	08
1.3.3 Principaux éléments de la machine TEST M25.....	08
1.3.3.1 Partie unité de travail	08
1.3.3.2 Partie automatisée.....	10
1.3.4 Principe de fonctionnement	11
1.4 Conclusion	11
Chapitre 2 :Etude théorique de l'automate S7-300	
2.1 Introduction.....	12
2.2 L'automate S7-300	12

2.2.1 Les modules de S7-300	13
2.2.1.1 Le module d'alimentation (PS)	13
2.2.1.2 Unité centrale (CPU)	13
2.2.1.3 Module de signaux (SM)	15
2.2.1.3.1 Les entrées Tout Ou Rien (TOR)	16
2.2.1.3.2 Les entrées analogiques	16
2.2.1.3.3 Les sorties tout ou rien	17
2.2.1.3.4 Les sorties analogiques	17
2.2.2 Avantages de S7-300	17
2.2.3 Fonctionnement de base d'un API	18
2.2.3.1 Le module central CPU	18
2.2.3.2 Réception des informations sur les états du système	18
2.2.3.3 Exécution du programme utilisateur	18
2.2.3.4 La commande du processus	18
2.2.4 Raccordement avec consol de programmation	19
2.2.5 Mise en œuvre d'un automate.....	19
2.3 Interface homme-machine (IHM).....	19
2.3.1 Le choix de l'IHM	20
2.3.2 Les fonctions de système IHM.....	21
2.4 Conclusion.....	22
Chapitre 3 : Simulations et resultats	
3.1 Introduction.....	23
3.2 Plan du travail.....	23
3.3 Définition du STEP7	24
3.3.1 Création d'un nouveau projet	24
3.3.2 Configuration matérielle	26
3.3.3 Table des mnémoniques	27

3.4	La programmation en STEP7	28
3.4.1	Blocs du programme utilisateur	28
3.4.2	Structure du programme	29
3.4.3	Description des différents blocs	30
3.5	Windows Control Center	34
3.5.1	Avantage du WinCC flexible	35
3.5.2	Principales fonctions offertes par WinCC flexible	35
3.6	Simulation	36
3.7	Conclusion	43
	Conclusion Générale.....	44
	Bibliographie.....	45
	Annexe A	
	Annexe B	

Au cours de cette étude, le travail que nous avons effectué au sein de l'entreprise SNVI est d'automatiser la machine MACHINTEST M25 à dresser et à centrer, équipée d'un ancien automate S5-100U, par un automate de nouvelle génération S7-300.

En premier lieu, nous avons procédé à l'étude descriptive de la MACHINTEST M25 ainsi que de son cahier de charge de fonctionnement, ensuite nous avons élaboré une modélisation cohérente de la machine à l'aide de GRAFCET. Le modèle GRAFCET que nous avons développé, nous a beaucoup aidés au passage vers la programmation en langage STEP7 et l'élaboration d'une solution programmable dans l'automate S7-300 utilisé. La programmation de la commande a été effectuée et testée par le logiciel S7-PLCSIM. Nous avons aussi développé un panel d'utilisation sur WinCC qui sert à faciliter, à performer la manipulation de la machine et à informer l'utilisateur en cas danger.

Tout au long de ce projet, nous avons apporté des modifications techniques sur l'installation de notre machine, ce qui contribue à l'amélioration du taux de production, but souhaité par toute automatisation.

Enfin, ce que nous retenons du travail que nous avons effectué durant ces 4 mois au sein du VIR(SNVI) est primordial dans notre vie de masters. Cette formation professionnelle nous a permis de découvrir le monde industriel, de mettre en application une partie de la théorie acquise le long de notre cursus et de tirer profit de l'expérience accumulée par les ingénieurs de la société en collaborant avec eux sur le terrain.

Ayant atteint les objectifs assignés, nous espérons que notre travail verra naître sa concrétisation sur le plan pratique. En perspective, il est envisageable d'élargir l'automatisation à d'autres machines de l'atelier et de prévoir une supervision.

Liste des figures

Figure 1.1 : Aspect extérieur d'un automate.	03
Figure 1.2 : Structure interne d'un API	04
Figure1.3 : La RAM	04
Figure 1.4 : L automate S5-100U	06
Figure 1.5 : Machintest M25.....	07
Figure 1.6: Vue d'ensemble de la machine	09
Figure1.7: Système automatisé	10
Figure 2.1 : API S7-300	12
Figure 2.2 : Les différents modules constituant S7-300	13
Figure 2.3 : CPU S7-300	14
Figure 2.4 : un module (SM) de S7-300	16
Figure 2.5 : panel touche écran SIEMENS	21
Figure 3.1 : Assistant de step7.....	23
Figure 3.2 : Le choix de la CPU.....	24
Figure 3.3 : Le choix du bloc OB1 et le langage de programmation (CONT).....	24
Figure 3.4 : Nom du projet	25
Figure 3.5 : Fenêtre d'un SIMATIC Manager.....	25
Figure 3.6 : Fenêtre du Configuration matériel.....	26
Figure 3.7 : Table mnémonique.....	27
Figure 3.8 : Structure du programme.....	28

Liste des figures (suite)

Figure 3.9 : <i>GRAFCET de la fonction blocage groupe</i>	28
Figure 3.10 : <i>blocage groupe</i>	29
Figure 3.11 : <i>GRAFCET serrage pièce</i>	30
Figure 3.12: <i>serrage piece</i>	30
Figure 3.13 : <i>GRAFCET "avance tête" fraisage droit</i>	30
Figure 3.14 : <i>GRAFCET "avance rapide" fraisage droit</i>	32
Figure 3.15 : <i>GRAFCET "avance travail" fraisage droit</i>	32
Figure 3.16 : <i>Fraisage droit</i>	33
Figure 3.17 : <i>Entourage de WINCC</i>	34
Figure 3.18 : <i>GRAFCET "avance rapide" centrage gauche</i>	36
Figure 3.19 : <i>GRAFCET "avance travail" centrage gauche</i>	36
Figure 3.20 : <i>Verification de la connection dans le STEP7</i>	37
Figure 3.21 : <i>Fenêtre de simulation</i>	38
Figure 3.22 : <i>Mode de simulation</i>	38
Figure 3.23 : <i>Page d'accueil</i>	39
Figure 3.24 : <i>Vue d'accueil</i>	40
Figure 3.25 : <i>Vue du pupitre</i>	41
Figure 3.26 : <i>Vue de diagnostique</i>	41
Figure 3.27 : <i>Vue historique des alarmes</i>	42

1.1 Introduction

Les automates programmables industriels (API) sont apparus aux états unis en 1969, pour répondre aux besoins de l'industrie de l'automobile américaine, afin de développer des chaînes de fabrication automatisées avec plus d'adaptabilité et de flexibilité.

Actuellement, l'API est le constituant le plus répandu des automatismes. On le trouve non seulement dans tous les secteurs de l'industrie, mais aussi dans autres services (gestion de parkings, accès à des bâtiments et dans l'agriculture).

1.2 Généralités sur les automates programmables [1]

1.2.1 Définition

Automate programmable industriel (en anglais Programmable Logic Controller PLC) est un dispositif électronique programmable par un personnel et destiné à piloter en ambiance industrielle et en temps réel des procédés industriels. Il est adaptable à un maximum d'application, d'un point de vue traitement, composants et langages.

Suite à cette définition, on distingue les rôles que l'automate devrait remplir :

- **Rôle de commande** : Où il est un composant d'automatisme, élaborant des actions suivant un algorithme approprié, à partir des informations que lui fournissent des détecteurs (Tout ou Rien) ou des capteurs (analogiques ou numériques).
- **Rôle de communication** : Avec des opérateurs humains (dialogue d'exploitation) ou avec d'autres processeurs, hiérarchiquement supérieurs (calculateur de gestion de production), égaux (autres automates intervenant dans la même chaîne) ou inférieurs (instrumentation intelligente).

Pour étudier cet équipement connecté à des systèmes réels en milieu industriel, il faut prendre en compte de nouvelles exigences telles que : le coût, la sûreté, la gestion de l'énergie, l'évolutivité et le dialogue.

Aujourd'hui le nom d'automate programmable symbolise la démarche d'une intégration totale des systèmes automatisés. Tous les constituants matériels et logiciels sont intégrés au sein d'un système unique.

1.2.2 Architecture Les automates programmables sont caractérisés principalement (figure 1-1) par :

- Un système Compact ou modulaire
- Une tension d'alimentation
- Une taille mémoire
- Une sauvegarde (EPROM, EEPROM, pile, ...)
- Un nombre d'entrées / sorties
- Des modules complémentaires (analogique, communication,..)
- Un langage de programmation

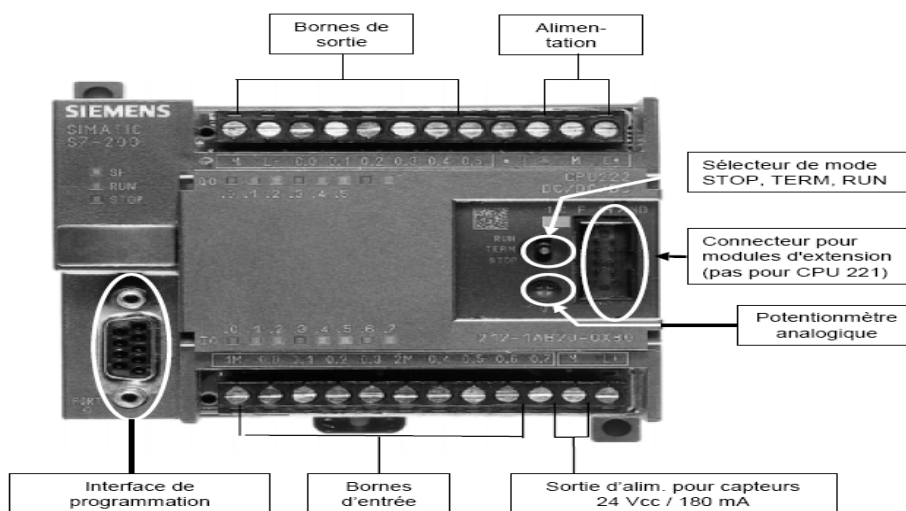


Figure 1-1 Aspect extérieur d'un automate [2]

La structure interne d'un automate programmable industriel (API) est assez voisine de celle d'un système informatique simple, L'unité centrale est le regroupement du processeur et de la mémoire centrale. Elle commande l'interprétation et l'exécution des instructions programme. Les instructions sont effectuées les unes après les autres, séquencées par une horloge.

Les principales parties de l'API (Programmable Logic Controller) (figure1-2) sont :

- Une unité traitement (un processeur CPU) ;
- Une mémoire ;
- Des modules d'entrées-sorties ;
- Des interfaces d'entrées-sorties ;
- Une alimentation 230 V, 50/60 Hz (AC) – 24 V (DC) ;

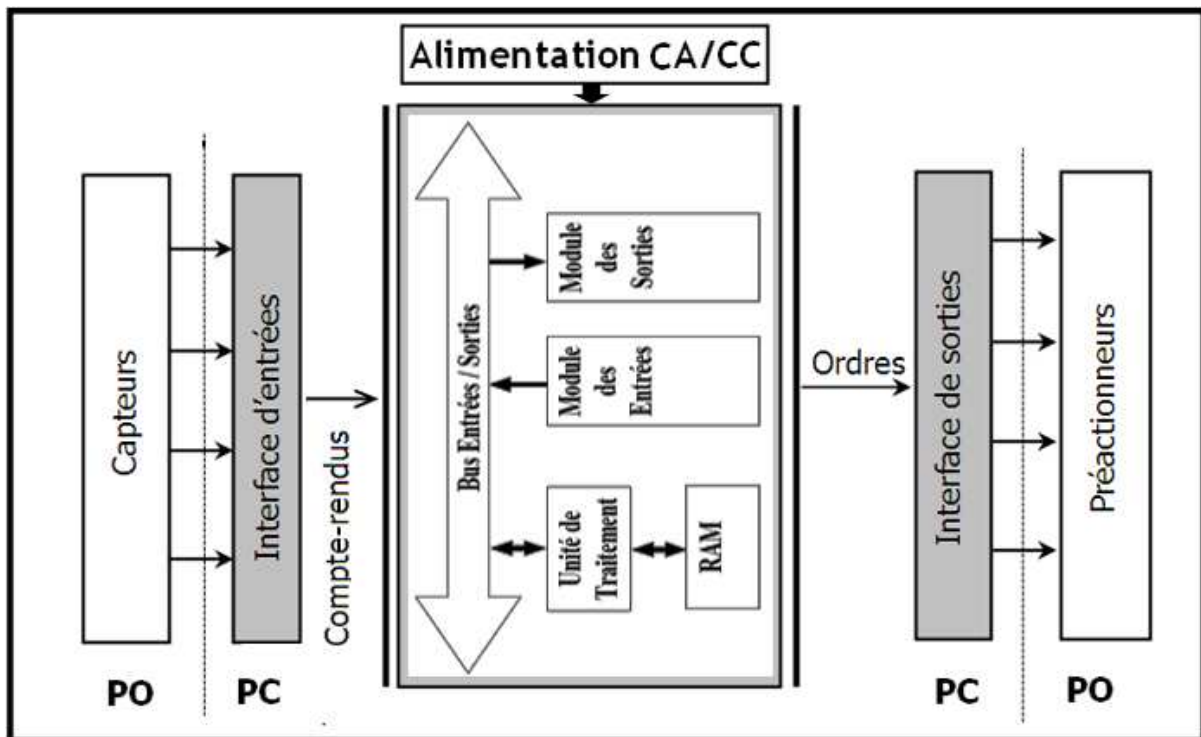


Figure 1-2 Structure interne d'un API [2]

1.2.2.1 Description des éléments d'un API :

➤ La mémoire :

Elle est conçue pour recevoir, gérer et stocker des informations issues des différents secteurs du système que sont le terminal de programmation (PC ou console) et le processeur, qui lui gère et exécute le programme. Elle reçoit également des informations en provenance des capteurs (voir figure 1-3).

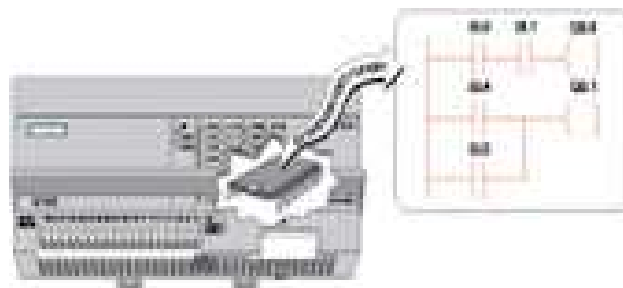


Figure1-3 La RAM [2]

Il existe dans les automates deux types de mémoires qui remplissent des fonctions différentes :

- La mémoire « Langage » où est stocké le langage de programmation. Elle est en général figée, c'est à dire en lecture seulement. C'est la ROM : mémoire morte.
- La mémoire « Travail » utilisable en lecture-écriture pendant le fonctionnement c'est la RAM (mémoire vive). Elle s'efface automatiquement à l'arrêt de l'automate (ou bien nécessite une batterie de sauvegarde).

➤ **Le processeur :**

Son rôle consiste d'une part à organiser les différentes relations entre la zone mémoire et les interfaces d'entrées et de sorties et d'autre part à exécuter les instructions du programme.

➤ **Les interfaces et les cartes d'Entrées / Sorties:**

L'interface d'entrée comporte des adresses d'entrée. Chaque capteur est relié à une de ces adresses. L'interface de sortie comporte de la même façon, des adresses de sortie. Chaque pré-actionneur est relié à une de ces adresses. Le nombre de ces entrées et sorties varie suivant le type d'automate. Les cartes d'E/S ont une modularité de 8, 16 ou 32 voies. Les tensions disponibles sont normalisées (24, 48, 110 ou 230V continu ou alternatif ...).

1.2.3 La programmation des automates

Elle peut s'effectuer de trois manières différentes : sur l'API Lui-même à l'aide de touches, avec une console de programmation reliée par un câble spécifique, ou avec un PC et un logiciel approprié.

Les langages de programmation utilisés sont :

- IL (Instruction List).
- ST (StructuredText).
- LD (LadderDiagram).
- FBD (Function Block Diagram).

1.2.4 L'automate programmable S5-100U [2]

L'automate programmable installé dans la machine sur laquelle nous travaillons dessus est de la gamme S5-100U fabriqué par Siemens (figure 1-4).

C'est un système d'automatisation robuste, sans ventilateur, couvrant toutes les applications de milieu de gamme, avec possibilité de configuration évolutive pour une adaptation exacte à la tâche à résoudre.

Il se compose d'un appareil de base et, suivant les besoins, d'un ou de plusieurs appareils d'extension. L'appareil de base comprend toujours un module d'alimentation et un module unité centrale (CPU).

Suivant l'application l'automate sera configurée avec différents modules de périphérie :

- modules d'entrées et de sorties TOR (Tout Ou Rien)
- modules d'entrées et de sorties analogiques
- processeurs de communication
- cartes périphériques intelligentes
- cartes pour applications spéciales

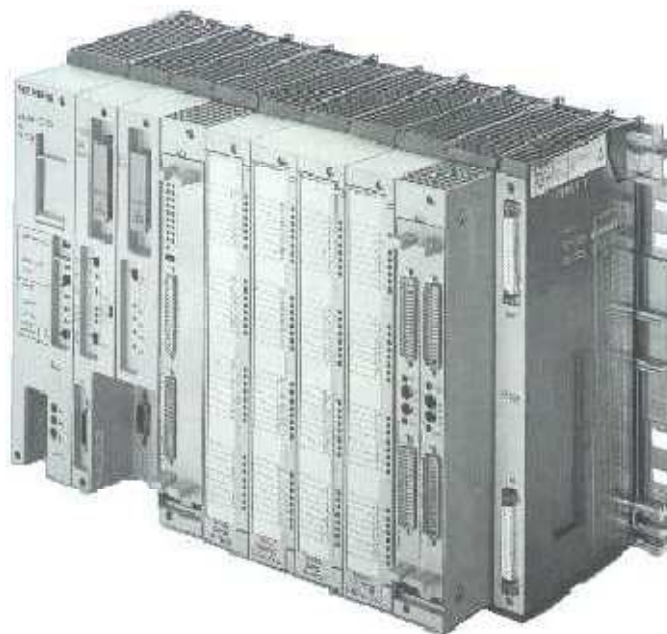


Figure 1-4 L'automate S5-100U [2]

1.2.5 Choix d'un API

Le choix d'un automate programmable est généralement basé sur :

- La capacité de traitement du processeur (vitesse, données, opération, temps réel...).
- Le type et le nombre des entrées/sorties nécessaires.
- Le cout de l'automate.
- La simplicité et la facilité de l'utilisation des logiciels de configuration.
- La qualité du service après-vente

1.3 Description de la machine TEST M25 [3]

1.3.1 Introduction

L'usinage est une étape de fabrication d'une pièce par enlèvement de matière. C'est une opération de base dans la fabrication de produits pour l'industrie mécanique. Le moyen de production associé à ces opérations d'usinage est une machine-outil qui est un équipement mécanique destiné à exécuter un usinage, ou autres tâches répétitives, avec précision et une puissance adaptée.

Parmi ces machines-outils, on trouve la machine "Machintest M25 "(figure1-5) fabriquée par la société italienne TOVAGLIERI s.r.l.



Figure1-5 Machintest M25 [4]

1.3.2 Présentation de la MACHINTEST M25 [3]

➤ La machine “machintest M25” est considérée comme une machine d’importance stratégique dans le parc machine.

C’est une machine à dresser et centrer, qui effectue deux principales opérations : le fraisage et le centrage de la pièce à usiner.

➤ Fraisage : Le fraisage est un procédé d’usinage particulier dans lequel l’outil doit en plus tourner sur lui-même par rapport au bâti de la machine-outil pour pouvoir couper la matière. C’est le mouvement de coupe.

➤ Centrage : C’est un usinage consistant à faire un trou dans une pièce. Ce trou peut traverser la pièce de part en part ou bien ne pas déboucher. On parle alors de trou borgne. Cette opération peut servir à faire passer une pièce (un arbre, un tube), un fluide, ou peut encore être taraudé pour recevoir une vis.

1.3.3 Principaux éléments de la machine TEST M25

La machine “ Machintest M25 ” est constituée de deux parties essentielles :

- Partie unité de travail.
- Partie automatisée.

1.3.3.1 Partie unité de travail

Cette dernière est constituée de différents éléments (figure1-6):

1. Armoire appareil électrique.
2. Protection positions fixes.
3. Protection anticouliss.
4. Volant limite course centrage.
5. Volant course fraisage.
6. Tête à centrer.
7. Tête à fraiser.
8. Banc.
9. Positionneur axial.

10. Lampe.
11. Poussoir de commande.
12. Etau.
13. Distributeur de lubrification.
14. Sous-tête.
15. Chariot.
16. Distributeur hydraulique.
17. Distributeur réfrigérant.
18. Transporteur de copeaux.

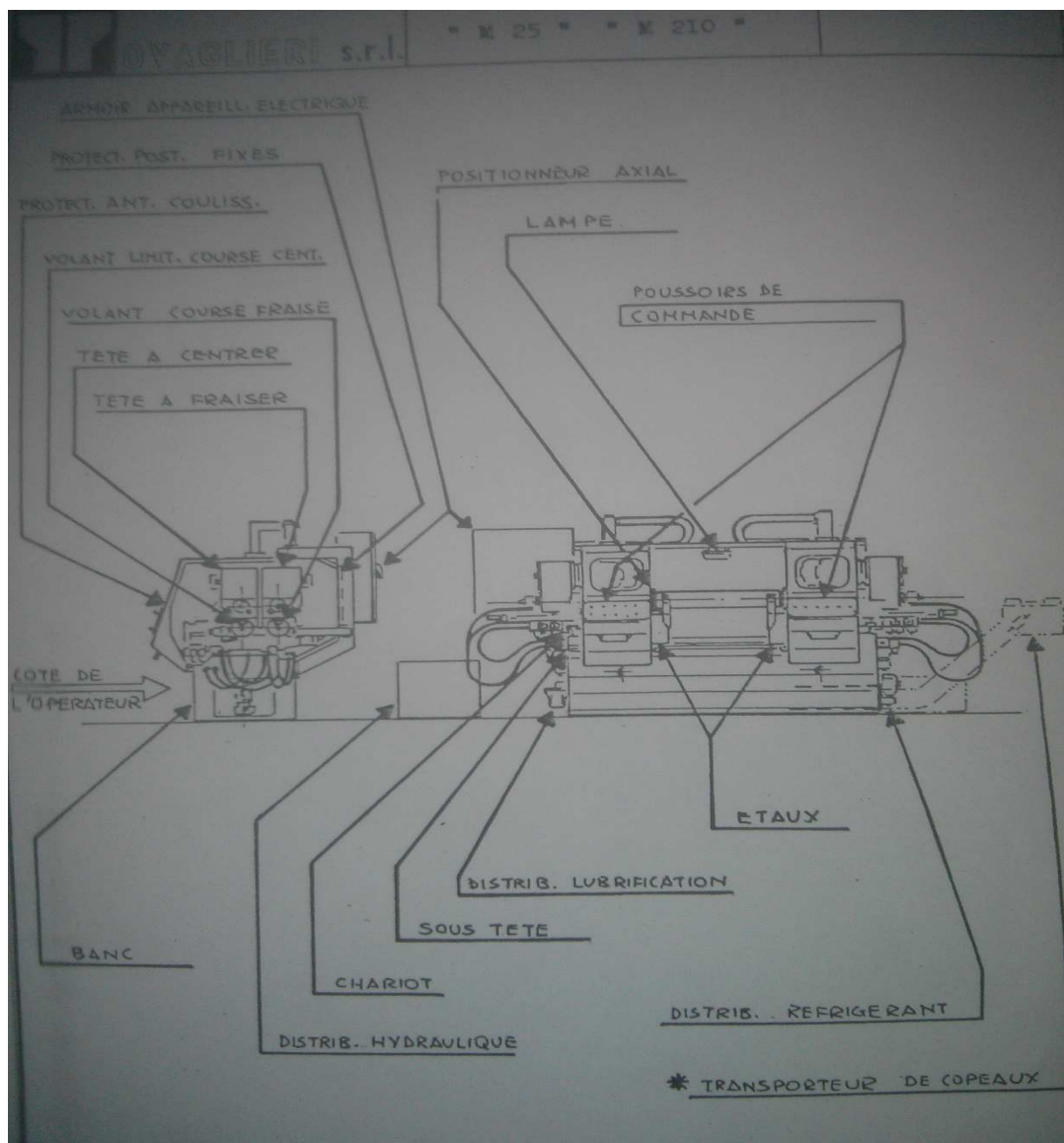


Figure1-6 Vue d'ensemble de la machine [3]

1.3.3.2 Partie automatisée

➤ Définition

L'automatisation est une conception qui permet à un système de passer d'une situation initiale à une situation finale sans une intervention humaine.

Le système automatisé est constitué de deux parties, comme il est indiqué sur la figure ci-dessous (figure 1-7) :

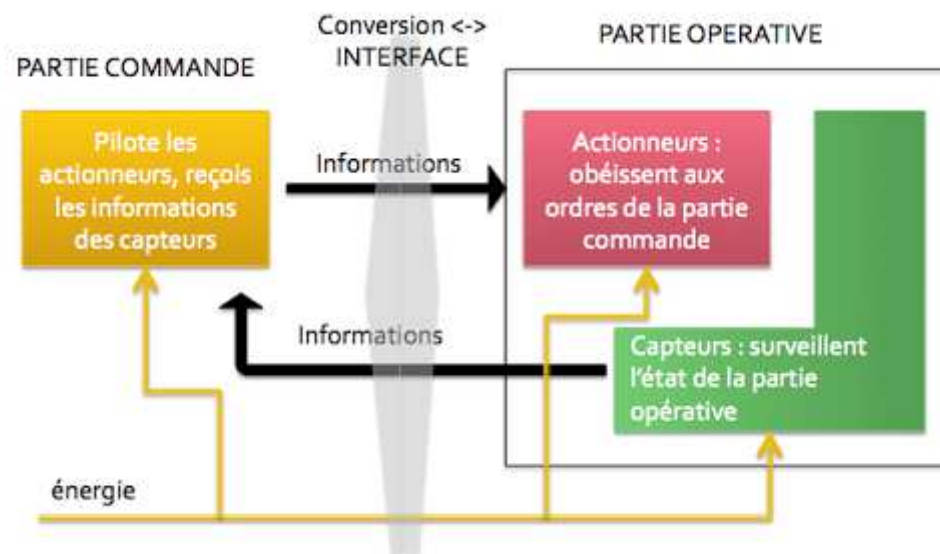


Figure 1-7 Système automatisé [2]

a. Partie commande :

Elle joue le rôle du cerveau du système. Elle ordonne la succession des actions de la partie opérative avec la finalité d'obtenir cette valeur ajoutée.

Sur notre machine, la partie commande comporte :

- Un pupitre : comportant plusieurs boutons poussoirs pour commander les différentes opérations et des leviers pour la commande manuelle.
- L'automate programmable S5-100U.

b. Partie opérative :

Elle exécute les ordres qu'elle reçoit de la partie commande grâce aux actionneurs. Elle possède aussi des capteurs qui permettent de recueillir des informations.

1.3.4 Principe de fonctionnement

La pièce à usiner est placée dans le système de bridage. Le serrage de la pièce s'effectue par un système hydraulique.

Une action sur le bouton commutateur (fraisage) déclenche le début de l'opération et fait tourner le moteur de la tête qui se déplace horizontalement afin qu'elle se positionne exactement sur le point que nous avons déjà établi. Une fois sur place, la tête se déplace par un rapprochement rapide vers la pièce à l'aide d'un vérin, un capteur (fin de course) arrête l'avance rapide et fait démarrer l'avance travail jusqu'au point où le capteur indiquera la fin du fraisage pressé. Le retour se fait rapidement.

Une action sur le bouton commutateur fait tourner le moteur de la perceuse qui se déplace par un rapprochement rapide jusqu'au capteur qui arrête l'avance rapide tout en déclenchant l'avance travail jusqu'à ce que le capteur indique la fin du centrage soit pressé. Le retour se fait rapidement.

Le desserrage s'effectue aussi par un système hydraulique.

1.4 Conclusion

Dans ce chapitre, nous avons parlé des automates en général puis présenté le principe de fonctionnement de la machine étudiée et ses différents éléments afin de faciliter la modification et la programmation que nous allons étudier par la suite.

2.1 Introduction

Pour une meilleure gestion des automates programmables, nous trouvons les Interfaces homme-machine (HMI) qui définissent les moyens et outils mis en œuvre afin qu'un humain puisse contrôler et communiquer avec une machine.

2.2 L'automate S7-300 [5]

L'automate programmable S7-300 est un mini automate de conception modulaire de la famille SIMATIC, destiné à des tâches d'automatisation moyennes hautes gammes (Voir la figure 2-1).



Figure 2.1 API S7-300 [6]

Son architecture est caractérisée par :

- Une gamme diversifiée de la CPU.
- Une gamme complète de modules.
- La possibilité d'extension jusqu'à 32 bits.
- Possibilité de mise en réseau avec : Profibus, interface multipoint (MPI), industriel Ethernet.
- Raccordement central de la console de programmation (PG) avec accès à tous les modules.
- Configuration et paramétrage à l'aide de l'outil «Configuration matérielle».

2.2.1 Les modules de S7-300

L'automate programmable S7-300 est d'une forme modulaire et permet un vaste choix de gamme de modules suivant (voir la figure 2-2) :

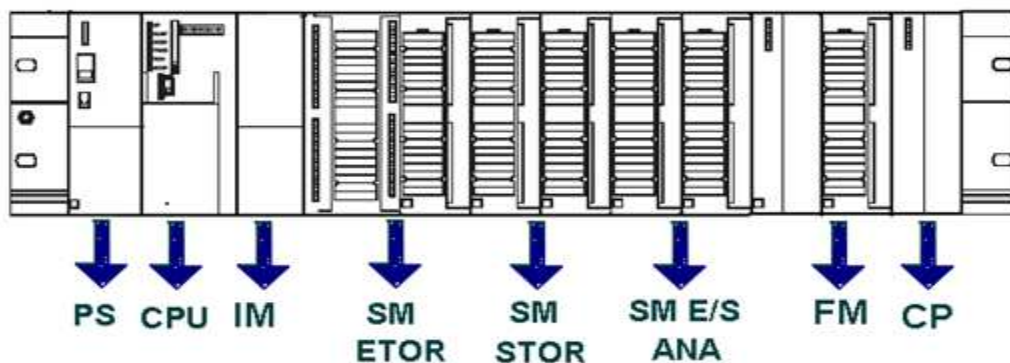


Figure 3.4 Les différents modules constituant S7-300 [5]

2.2.1.1 Le module d'alimentation (PS)

Le module d'alimentation (PS) délivre sous une tension de 24 volts, un courant de sortie assigné de 2A, 5A, et 10A. La tension de sortie est à séparation galvanique pour protéger la CPU contre les courts circuits.

2.2.1.2 Unité centrale (CPU)

C'est une carte électronique bâtie autour d'un ou plusieurs processeurs et mémoires. La CPU possède un système d'exploitation, une unité d'exécution et des interfaces de communication. Essentiellement la CPU lit l'état des signaux d'entrée et exécute le programme utilisateur séquentiellement (Voir la figure 2-3).



Figure 2.3 CPU S7-300 [6]

a) Interfaces MPI

Chaque CPU est équipée d'une interface MPI pour la connexion de la console de programmation (PG) ou un appareil (par exemple adaptateur PC).

b) Commutateur de mode de fonction

Les modes de fonctionnement sont :

- RUN –P : exécution de programme, accès en écriture et en lecture avec la PG.
- RUN : exécution de programme, accès en lecture seule avec la PG.
- STOP : le programme n'est pas exécuté, toutes les fonctions avec la PG sont autorisées.
- MRES : position dans laquelle un effacement général de la CPU peut être effectué.

c) La carte mémoire

Une carte mémoire peut être montée à la CPU ; elle conserve le contenu de programme en cas de coupure de courant, même en l'absence de la pile.

d) Le processeur

C'est le cerveau de l'automate. Son rôle consiste d'une part à organiser les différentes relations entre la zone mémoire et les interfaces d'entrée et de sortie et d'autre part à gérer les instructions du programme. Il est composé :

- D'une Unité Logique(UL) qui traite les opérations logiques ET, OU et la Négation.
- D'une Unité Arithmétique et Logique (UAL) qui traite les opérations de temporisation, de comptage et de calcul.
- D'un Accumulateur qui est un registre de travail dans lequel se range une donnée ou un résultat.
- D'un Décodeur d'Instruction qui décode l'instruction à exécuter en y associant les microprogrammes de traitement.
- D'un Compteur Programme ou Compteur Ordinal qui l'adresse à la prochaine instruction à exécuter et gère ainsi la chronologie de l'exécution des instructions du programme.

e) La mémoire

Le stockage des données et des programmes s'effectue dans la mémoire. Ces mémoires peuvent être :

- Des RAM ou des EPROM durant la phase d'étude et de mise au programme
- Des RAM ou des RAM ou PROM durant la phase d'exploitation.

2.2.1.3 Module de signaux (SM)

Il sert d'interface entre le processus et l'automate. Il existe des module d'entrée TOR, des modules de sortie TOR ainsi que des modules d'entrée analogiques et des modules de sortie analogiques. Les modules d'entrée/sortie sont des interfaces vers les capteurs et les actionneurs d'une machine ou d'une installation (Voir la figure 2-4).



Figure 2.4 un module (SM) de S7-300 [6]

2.2.1.3.1 Les entrées Tout Ou Rien (TOR)

Les modules d'entrées tout ou rien permettent de raccorder à l'automate les différents capteurs logiques. Ils assurent l'adaptation, l'isolement, le filtrage et la mise en forme des signaux électroniques. Une diode électroluminescente située sur la carte donne l'état de chaque entrée. Le nombre d'entrées sur une carte est de : 4, 8, 16, 32. Les tensions d'entrées sont de : 24, 48, 110, 220 volts en courant continu ou alternatif.

2.2.1.3.2 Les entrées analogiques

Les cartes d'entrées analogiques permettent de gérer des grandeurs analogiques en faisant varier un code numérique au sein du module. Il existe 3 types d'entrées analogiques :

- Haut niveau qui accepte en tension 0/10V et en intensité 0/20mA ou 4/20mA.
- Thermocouple avec un signal d'entrée 0/20mV, 0/50mV, 0/100mV.
- Sonde Pt 100 avec un signal d'entrée 0/100mV, 0/250mV, 0/400mV.

Sur le marché, il existe des modules à 2, 4, 8 voies d'entrées.

Les entrées analogiques disposent d'un seul convertisseur analogique/numérique, elles sont scrutées les unes à la suite des autres par un multiplexeur à relais.

2.2.1.3.3 Les sorties tout ou rien

Les modules de sortie tout ou rien permettent de raccorder à l'automate les différents pré-actionneurs.

Les tensions de sorties usuelles sont de 5, 24, 48, 110, 220 volts en continu ou en alternatif. Les courants vont de quelques mA à quelques ampères.

Ces modules possèdent soit des relais, soit des triacs des transistors. L'état de chaque sortie est visualisé par une diode électroluminescente.

2.2.1.3.4 Les sorties analogiques

Les modules de sorties analogiques permettent de gérer des grandeurs analogiques en faisant varier un code numérique au sein du module. Il existe deux grands types de sorties :

- Avec une résolution de 8 bits.
- Avec une résolution de 12 bits.

Les sorties analogiques peuvent posséder un convertisseur par voie. Le nombre de voies sur ces cartes est de 2 ou 4.

2.2.2 Avantages de S7-300

Le S7-300 offre de nombreux avantages :

- Une construction compacte et modulaire, libre de contraintes de configuration.
- Une riche gamme de modules adaptés à tous les besoins du marché est utilisable en architecture centralisée.
- Une large gamme de CPU.
- Une partie de la gamme est déclinée en version SIPLUS.
- Une large plage de température de -25°C à $+60^{\circ}\text{C}$.
- Une meilleure tenue aux sollicitations mécaniques.
- Une résistance à la pollution par des gaz nocifs, poussières et humidité de l'air.
- Une condensation admissible augmentée grâce à un revêtement « ConformalCoating ».

2.2.3 Fonctionnement de base d'un API

2.2.3.1 Le module central CPU

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

2.2.3.2 Réception des informations sur les états du système

Le S7-300 reçoit des informations sur l'état du processus via les capteurs de signaux reliés aux entrées. Il met à jour la mémoire image au début de chaque cycle de programme en transférant l'état des signaux d'entrée des modules vers la mémoire image des entrées ce qui permet à la CPU de savoir l'état de processus.

2.2.3.3 Exécution du programme utilisateur

Après avoir acquis les informations d'entrée et exécuter le système d'exploitation, la CPU passe à l'exécution du programme utilisateur, qui contient la liste d'instructions à exécuter pour faire fonctionner le procédé. Il est composé essentiellement de bloc de données, de bloc de code et de bloc d'organisation.

2.2.3.4 La commande du processus

Pour commander le processus, on doit agir sur les actionneurs. Ces derniers reçoivent l'ordre via le module de sorties du S7-300. Donc l'état des sorties est connu après l'exécution du programme utilisateur par la CPU, puis elle effectue la mise à jour de la mémoire image des sorties pour communiquer au processus le nouvel état.

2.2.4 Raccordement avec console de programmation

Les consoles de programme « SIMATIC » sont des outils pour la saisie, le traitement et l'archivage des données du processus, ainsi que la suppression du programme. Avec l'atelier logiciel « SIMATIC », l'utilisateur dispose d'une gamme d'outils complète de chaque tâche d'automatisation. Le raccordement entre l'automate et la console est réalisé par l'interface multi points (MPI).

2.2.5 Mise en œuvre d'un automate

A partir d'un problème d'automatisme donné, dans lequel on définit les commandes, les capteurs, les organes de sortie et le processus à réaliser, il faut établir :

- Le GRAFCET niveau 1 et le GRAFCET niveau 2.
- Faire le repérage des entrées/sorties.
- Ecrire le programme, le charger dans la mémoire RAM/EPROM et le transférer dans l'unité centrale de l'automate.
- Tester à vide (mise au point).
- Raccorder l'automate à la machine.

2.3 Interface homme-machine (IHM) [5]

Les concepts d'automatisation modernes ont, sans cesse, des exigences croissantes en matière de visualisation des processus. Plus particulièrement, il est impératif que la conduite des processus au niveau machine fournisse une réponse adaptée aux besoins de simplicité et de performance. L'objectif est de présenter rapidement et de manière fiable, des données de processus immédiatement compréhensibles par l'opérateur, par exemple, sous la forme d'une courbe graphique. Il est donc indispensable d'archiver les données de processus dès le niveau machine, d'où l'indispensabilité des interfaces homme-machine.

Une interface homme-machine (IHM) est une interface qui permet une interaction entre un être humain et une machine.

Deux composants sont nécessaires dans une interface homme-machine. La première est une entrée, un utilisateur humain a besoin d'une certaine façon d'injecter des commandes à la

machine, ou la régler. Des exemples de dispositifs d'entrée incluent des claviers, des clés, des commutateurs, des écrans tactiles et des souris. Tous ces dispositifs peuvent être utilisés pour envoyer des commandes à un système, ou même un ensemble interdépendant de systèmes.

L'interface nécessite également une sortie, ce qui permet à l'appareil de maintenir l'utilisateur humain mis à jour, sur l'état d'avancement des commandes ou à exécuter des commandes dans l'espace physique. Par exemple, les utilisateurs disposent d'un écran qui peut afficher des informations. Les sorties peuvent également comprendre des choses aussi simples que des voyants d'état qui alertent les gens.

Les SIMATIC Panels de Siemens font leurs preuves depuis des années dans les applications et les secteurs les plus divers. Ils possèdent non seulement un design innovant, mais offrent également des performances élevées. Avec une efficacité de l'ingénierie inégalée.

Les pupitres SIMATIC sont clairement structurés :

- Les SIMATIC HMI Basic Panels offrent des fonctions de base pour les applications IHM simples.
- Les SIMATIC HMI Confort Panels conviennent aux applications exigeantes.

2.3.1 Le choix de l'IHM

Dans notre projet, on introduit un nouvel objet qui est la station IHM en choisissant le type de pupitre sur lequel, les informations seront transmises.

Pour notre application, on a utilisé un Multi Panel MP 277 10" Touch en liaison avec l'automate (Voir la figure 2-5).

Le Multi Panel MP 277 10" Touch est le représentant de la catégorie de produits appelées "Plate-forme multifonctionnelle". Les pupitres opérateurs offrent des possibilités de communication avec le monde de la bureautique élargies.

Les appareils sont équipés des composants suivants :

- Interface PROFIBUS.
- Interface Ethernet pour la connexion à PROFINET.
- 2 interfaces USB.
- Interface MPI.
- Ecran TFT(LCD) couleurs jusqu'à 64k.



Figure 2.5 panel touche écran SIEMENS [6]

2.3.2 Les fonctions de système IHM

Un système IHM se charge des tâches suivantes:

➤ **Représentation du processus**

Le processus est représenté sur le pupitre opérateur. Lorsqu'un état du processus évolue, par exemple : l'affichage du pupitre opérateur est mis à jour.

➤ **Commande du processus**

L'opérateur peut commander le processus via l'interface utilisateur graphique. Il peut par exemple : définir une valeur de consigne pour un automate ou démarrer un moteur.

➤ **Vue des alarmes**

Lorsque surviennent des états critiques dans le processus, une alarme est immédiatement déclenchée, par exemple : lorsqu'une valeur limite est franchie.

➤ **Archivage de valeurs processus et d'alarmes**

Les alarmes et valeurs processus peuvent être archivées par le système IHM. Vous pouvez ainsi documenter la marche du processus et accéder ultérieurement aux données de la production écoulée.

➤ **Documentation de valeurs processus et d'alarmes**

Les alarmes et valeurs processus peuvent être éditées par le système IHM sous forme de journal. Vous pouvez ainsi consulter les données de production à la fin d'une équipe par exemple.

2.4 Conclusion

L'API est un équipement spécialement conçu pour l'industrie et destiné à piloter des chaînes de montage, production, manutention, robots industriels, machines-outils...

Dans ce chapitre, nous avons présenté les différents constituants de l'API et IHM, ainsi que leurs avantages.

Enfin, nous avons présenté les caractéristiques de l'API S7-300 et défini les critères qui nous ont amené à faire le choix de ce dernier pour automatiser notre système.

3.1 Introduction

Les automates programmables effectuent des tâches d'automatisation traduites sous formes de programme d'application. L'utilisateur définit la manière dont l'automate doit commander l'installation par une suite d'instructions, le programme doit être écrit dans un langage déterminé avec des règles définies pour que l'automate puisse l'exécuter. Pour cela, les automates de la famille SIEMENS sont programmés grâce au logiciel STEP7 via une console de programmation ou PC et sous un environnement WINDOWS. A l'automatisme s'ajoute le programme de la supervision des états d'une machine ou d'une installation WINCC qui nécessite la surveillance de notre processus.

Notre machine est équipée d'un automate S5-100U qui est un modèle très ancien et qui présente pas mal d'inconvénients dans le milieu industriel. L'inconvénient le plus répandu est son architecture qui pose un problème lors qu'une panne survient.

Pour cela, nous avons choisi :

- Un API (Automate Programmable Industriel) de la gamme S7-300, un module d'E/S TOR (Tout ou Rien) et un autre module de sorties TOR.
- Un panel 277 10" Touch pour la supervision des états de la machine.

3.2 Plan de travail

Après la description des différents blocs de fonctionnement de notre machine, le problème sera de proposer des solutions qui décrivent les relations entre la partie commande et la partie opérative et qui répondent aux exigences du cahier de charge. Pour cela, on a utilisé un outil de modélisation graphique qui est le GRAFCET.(voir définition en annexe A)

On a converti cette modélisation graphique en un langage de programmation (ladder) des systèmes d'automatisation grâce au logiciel STEP7.

Enfin, on a fait une intégration de ce programme à un logiciel de supervision et de commande : le WINCC.

3.3 Présentation du STEP7 [7]

STEP7 est un logiciel de base permettant la configuration et la programmation des systèmes d'automatisation SIMATIC. Il s'exécute sous un environnement Windows, à partir d'une console de programmation ou d'un PC.

Il en existe plusieurs versions : STEP micro /Dos et STEP micro/Win pour les applications S7-300 et S7-400.

Le logiciel STEP7 offre les possibilités suivantes :

- Configuration et paramétrage du matériel et de communication.
- Création et gestion des projets.
- Gestion des mnémoniques.
- Test de l'installation d'automatisation.
- Le diagnostic lors des perturbations dans l'installation.
- Documentation et archivage.

3.3.1 Création d'un nouveau projet

Les procédures de la création de projet sous logiciel STEP7 sont comme suite :

1. Double-cliquer sur l'icône SIMATIC MANAGER. La fenêtre illustrée en figure3.1 apparait, elle permet la création d'un nouveau projet.



Figure 3.1 Assistant de step7 [7]

- En cliquant sur l'icône suivante, la fenêtre qui apparaît nous permet de choisir la CPU. Pour notre projet nous avons choisi la CPU 312 (Voir figure 3.2).

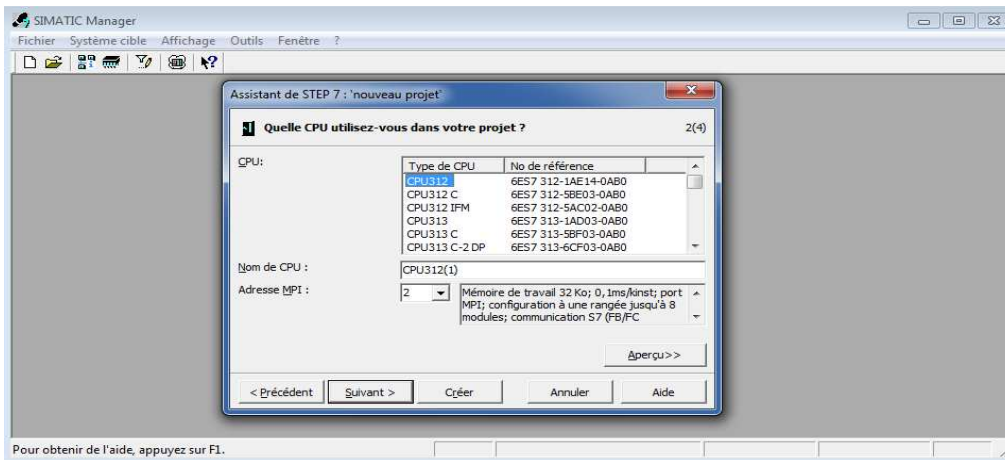


Figure 3.2 Le choix de la CPU [8]

- Après la validation de la CPU, la fenêtre qui apparaît permet de choisir les blocs à insérer, et choisir le langage de programmation (LIST, LOG ou CONT).

Nous avons choisi le bloc OB1 (bloc d'organisation) et le langage à contact (CONT) comme le montre la figure 3.3.

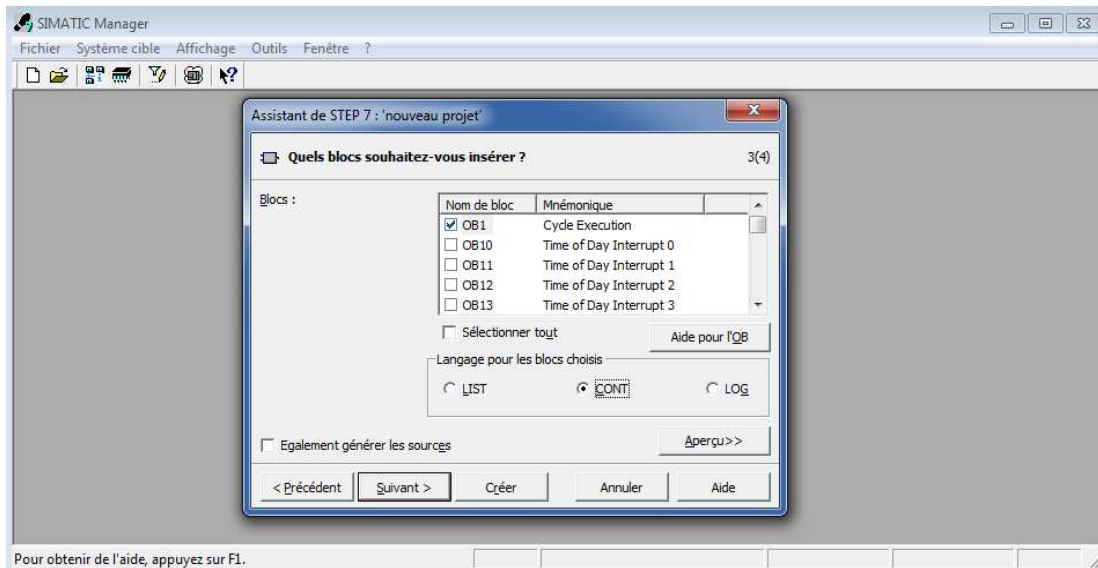


Figure 3.3 Le choix du bloc OB1 et le langage de programmation (CONT) [8]

4. En cliquant sur suivant, l'icône de la création de projet apparait pour le nommer.
(Voir figure 3.4).

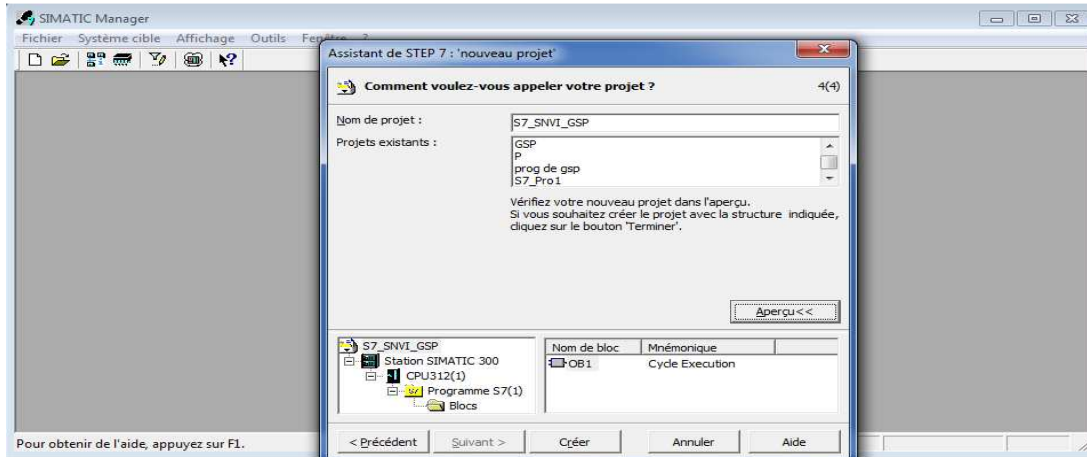


Figure 3.4 Nom du projet [8]

5. On clique sur créer, la fenêtre suivante (Figure 3.5) apparait.

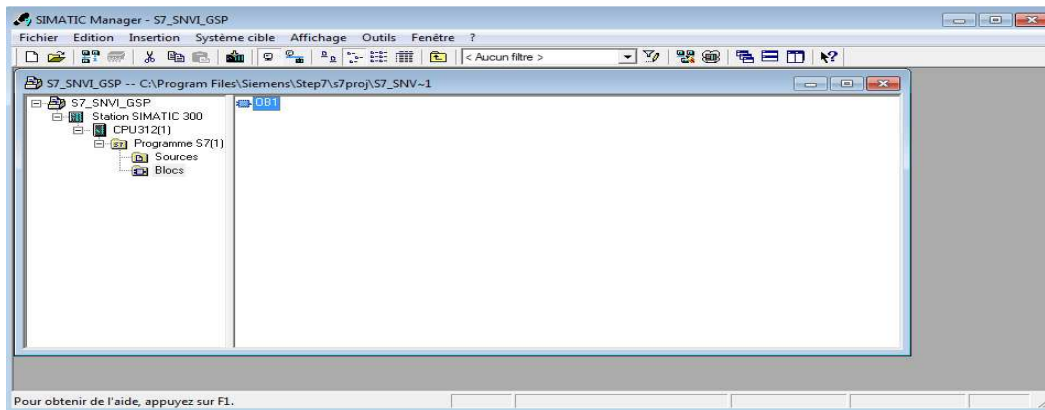


Figure 3.5 Fenêtre d'un SIMATIC Manager [8]

3.3.2 Configuration matérielle :

La configuration matérielle est une étape importante dans la mise en œuvre de notre programmation. Elle consiste à la description et le choix de la disposition des châssis (rack), des modules et périphériques centralisés. Les châssis sont représentés par une table de configuration dans laquelle on peut placer un nombre défini de modules, comme dans les châssis réels.

Nous avons choisi une alimentation PS 307 5 A, un CPU 313C, un module d'E/STOR, et un module de sortie TOR pour établir notre configuration matérielle.

La figure suivante illustre notre configuration matérielle (Voir figure 3.6).

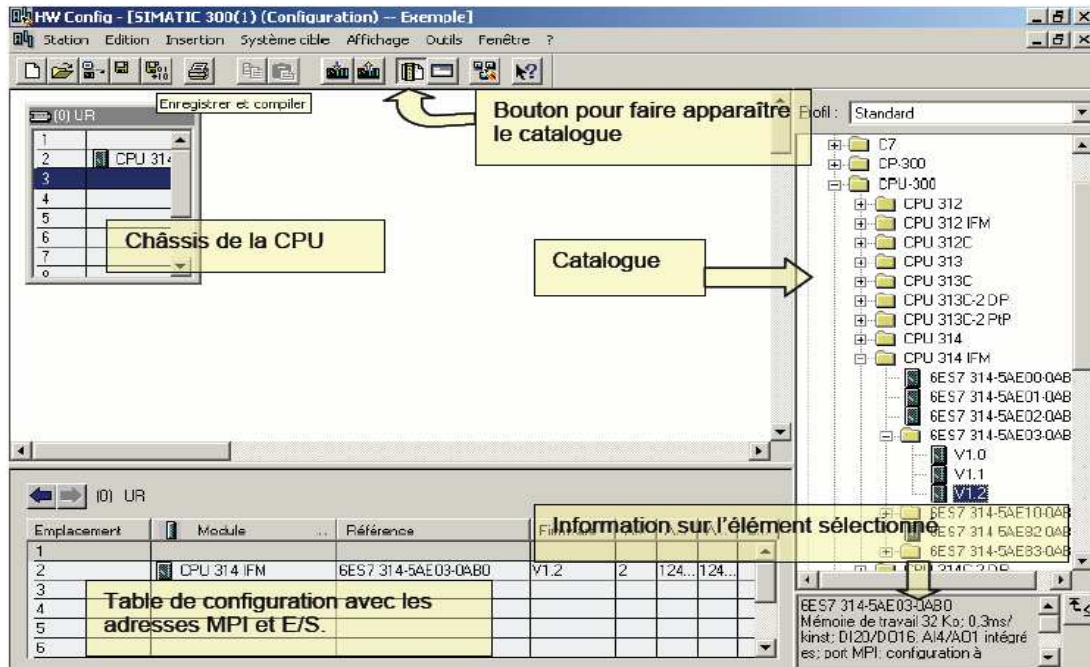


Figure 3.6 Fenêtre du Configuration matériel [8]

3.3.3 Table des mnémoniques

Une mnémonique est un nom que l'utilisateur définit en respectant les règles de la syntaxe imposées. Il est destiné à rendre le programme lisible et aide à gérer facilement les grands nombre de variables couramment utilisées dans le programme. Ce nom peut être utilisé pour la programmation et le contrôle commande une fois son affectation terminée.

La figure suivante (Figure 3.7) illustre une partie de la table des mnémoniques du projet.

	Etat	Mnémonique /	Opérande	Type de d	Commentaire
49		CN31	M 8.6	BOOL	condition necessaire retour etape initiale (urgence) cent droit
50		CN32	M 8.7	BOOL	condition necessaire attente8
51		CN33	M 9.1	BOOL	condition necessaire retour etape initiale (urgence) frais droit
52		CN34	M 9.2	BOOL	conditions necessaire attente9
53		CN35	M 9.4	BOOL	condition necessaire retour etape initiale (urgence) frais gauche
54		CN36	M 9.6	BOOL	Conditions necessaire retour etape initiale frais gauche
55		CN37	M 10.0	BOOL	condition necessaire retour etape initiale frais droit
56		CN4	M 1.2	BOOL	condition necessaire desserrage (cas d'urgence)
57		CN5	M 1.4	BOOL	condition necessaire pour l'indication (pièce desserrée)
58		CN6	M 1.6	BOOL	condition necessaire retour
59		CN7	M 2.0	BOOL	condition necessaire desserrage
60		CN8	M 2.1	BOOL	condition necessaire retour etat initiale
61		CN9	M 2.2	BOOL	
62		CNC AFG	M 6.3	BOOL	conditions nécessaire communes fraisage gauche
63		CNCD	M 4.0	BOOL	Conditions communes centrage droit
64		CNCF	M 5.1	BOOL	conditions nécessaire communes fraisage droit
65		CNCG	M 2.5	BOOL	Conditions communes centrage gauche
66		dem_mot_frais_...	FC 14	FC 14	
67		dem_mot_frais_...	FC 17	FC 17	
68		demmarage_mot...	FC 11	FC 11	
69		demmarage_mot...	FC 8	FC 8	
70		Desserrage	FC 4	FC 4	
71		Desserrage pièce	OB 4	OB 4	
72		etape A	M 1.1	BOOL	etapde d'attente
73		etape A2	M 3.2	BOOL	etapde d'attente2

Figure 3.7 Table mnémonique [9]

3.4 La programmation en STEP7

La programmation sur STEP7 peut s’effectuer par trois langages:

- **Mode liste (LIST)** : image textuelle proche du comportement interne de l’automate.
- **Mode logigramme (LOG)** : langage graphique, utilisant les symboles de l’électronique numérique (portes logiques).
- **Mode contact (CONT)** : suite de réseaux parcourus séquentiellement dont les entrées sont représentées par des interrupteurs (**I**) et les sorties par des bobines (**Q**).

3.4.1 Blocs du programme utilisateur

Le logiciel STEP7 dans ces différents langages de programmation possède un nombre important de blocs, destinés à structurer le programme d’utilisateur dont on peut citer les blocs importants suivants :

- Bloc d’organisation (OB).
- Bloc fonctionnel (FB).

- Bloc de données d'instance (DB d'instance).
- Type de données globales (TB).
- Les fonctions (FC).

3.4.2 Structure du programme

L'écriture du programme utilisateur complet peut se faire par bloc d'organisation OB1, cela n'est recommandé que pour les programmes de petite taille.

Pour les automatismes complexes, la subdivision en parties plus petite est recommandée. Celles-ci correspondent aux fonctions technologiques du processus, et sont appelées blocs (programmation structurée). Cette structure offre les avantages suivants :

- Standardiser certaines parties du programme.
- Simplifier l'organisation du programme.
- Modifier facilement le programme.
- Simplifier le test du programme en l'exécutant section par section.
- Faciliter la mise en service.

Pour cela, nous avons subdivisé notre programme comme le montre la figure 3.8 ci-dessous :

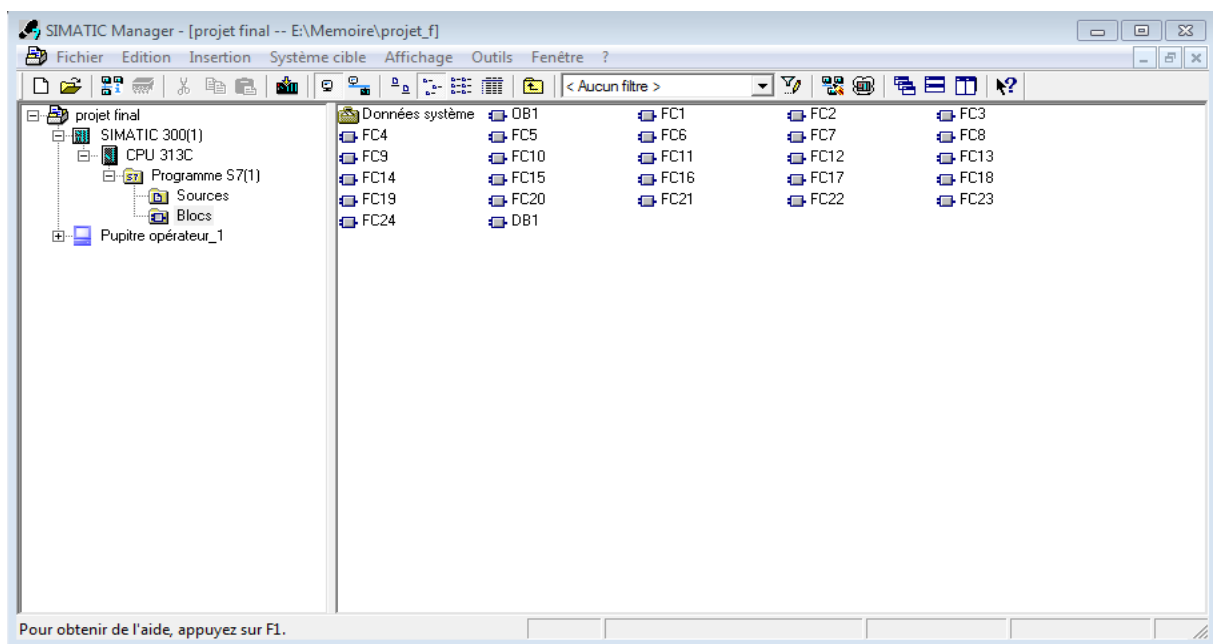


Figure 3.8 Structure du programme [9]

3.4.3 Description des différents blocs

➤ **Bloc OB1**

OB1 est le bloc principal d'organisation qui fait appel aux fonctions (FC). Autrement dit , ce bloc réunit toutes les fonctions utilisées au cours de notre programme.

➤ **Bloc FC1 (blocage groupe)**

Le bocage groupe se réalise par le biais d'un bouton commutateur **S12** puis la pression du bouton de cycle manuel **S4** .Ceci excite l'électrovalve **EV23** du vérin de blocage qui se déplace jusqu'au capteur de fin de course tout en avant, ce qui allume la lampe de blocage **L9**.

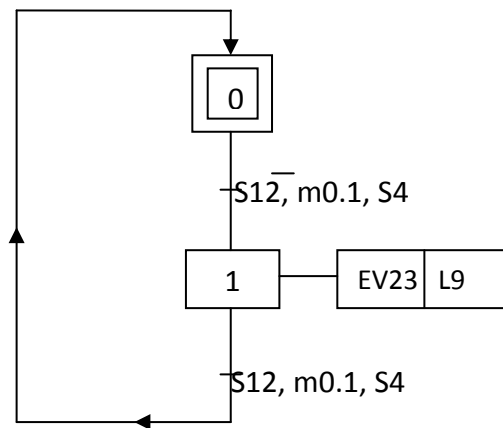


Figure 3.9 GRAFCET de la fonction blocage groupe

Où S12, S4, m0.1 Etc. Représentent des boutons ou des conditions de sécurité (voir mnémonique utilisée en Annexe B).

Voici un aperçu sur le bloc FC1 (figure 3.10) dans l'OB1

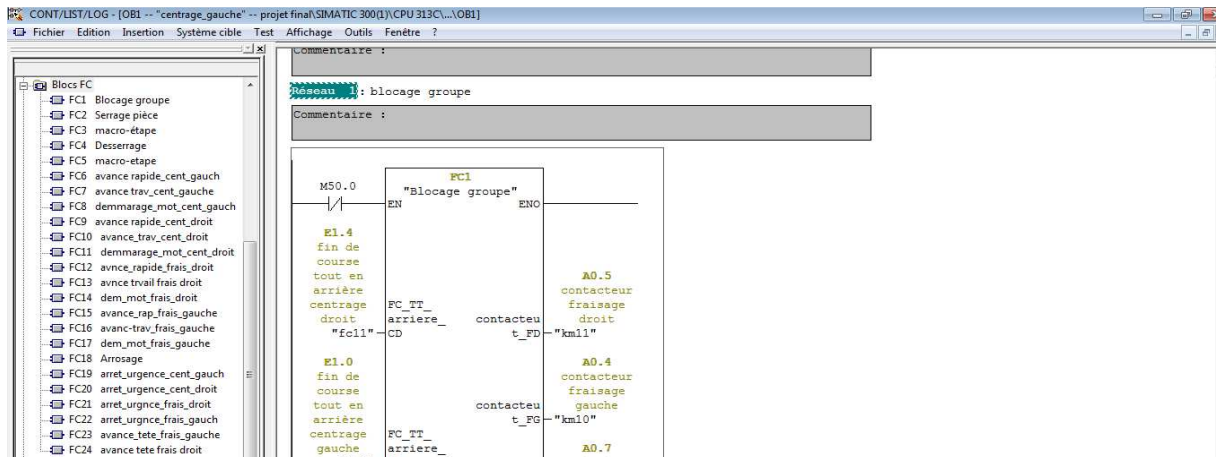


Figure 3.10 Blocage groupe [9]

➤ **Bloc FC2/FC3 (Serrage pièce)**

Après le **(blocage table)** et tout en assurant que les conditions de sécurité soient mis en place, L'opérateur pose la pièce manuellement et presse le bouton de serrage **P8**. Les vérins de serrage se déplacent pour serrer la pièce à usiner.

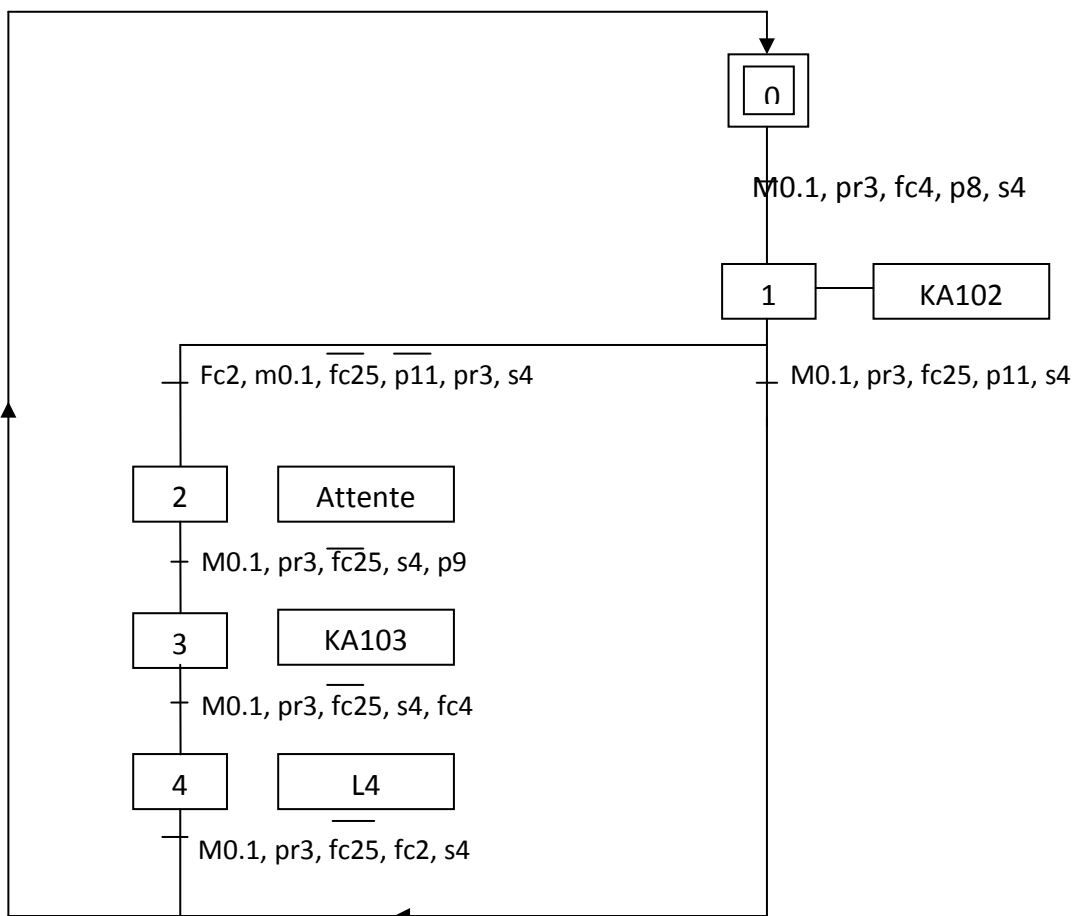


Figure 3.11 GRAFCET Serrage pièce

Voici un aperçu sur le programme de serrage dans OB1. Voir (figure 3.12).

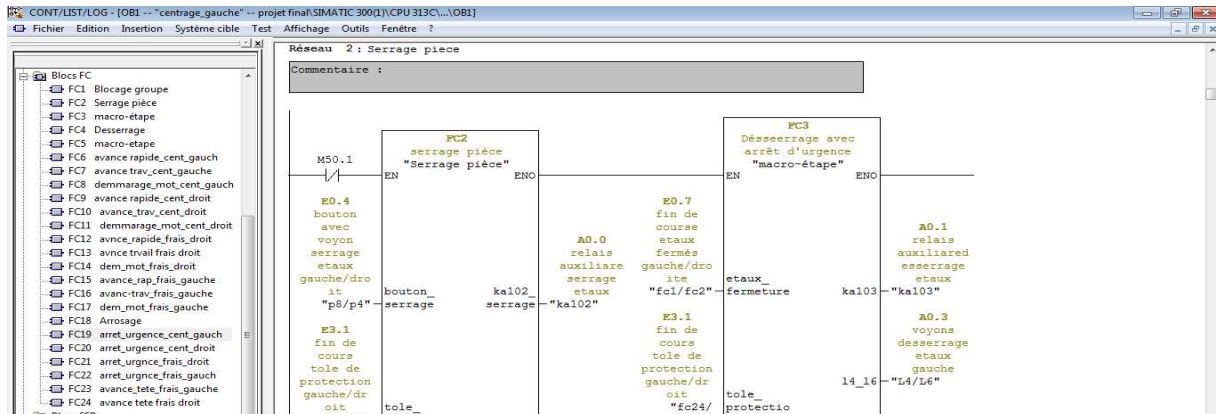


Figure 3.12 Serrage pièce [9]

Bloc FC12/13 (Fraisage pièce droit)

Avant de commencer l’opération de fraisage, il faut vérifier que la pièce soit serrée et le groupe soit bloqué et les fins de courses du fraisage et du centrage soient tout en arrière. En pressant le bouton de fraisage **S9** l’opération passe par 3 périodes successives :

- avance tête (figure 3.13),
- avance rapide (figure 3.14),
- puis avance travail qui assure le fraisage de la pièce (figure 3.15).

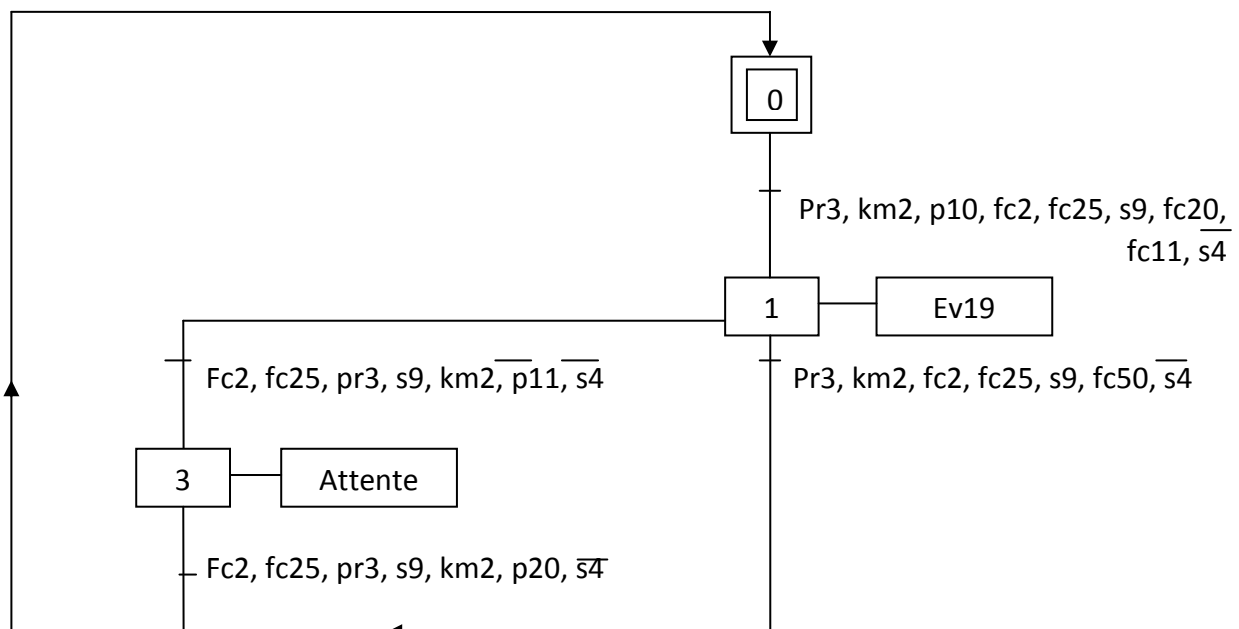


Figure 3.13 GRAFCET “avance tête” fraisage droit

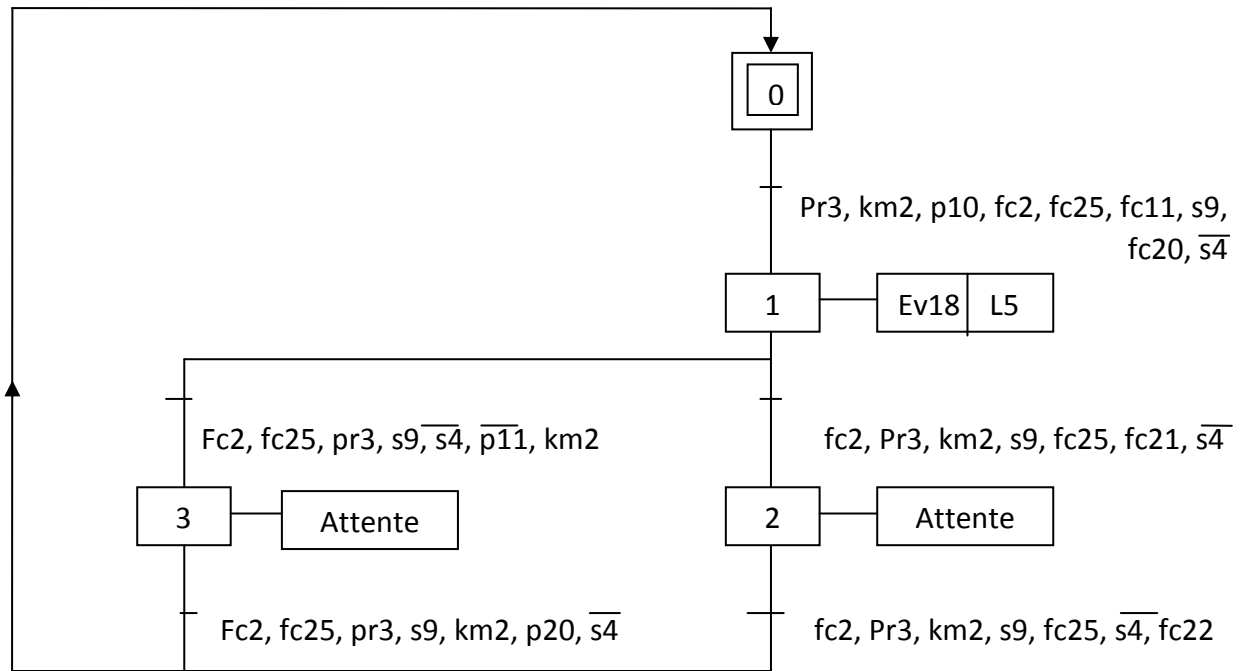


Figure 3.14 GRAFCET "avance rapide" fraisage droit

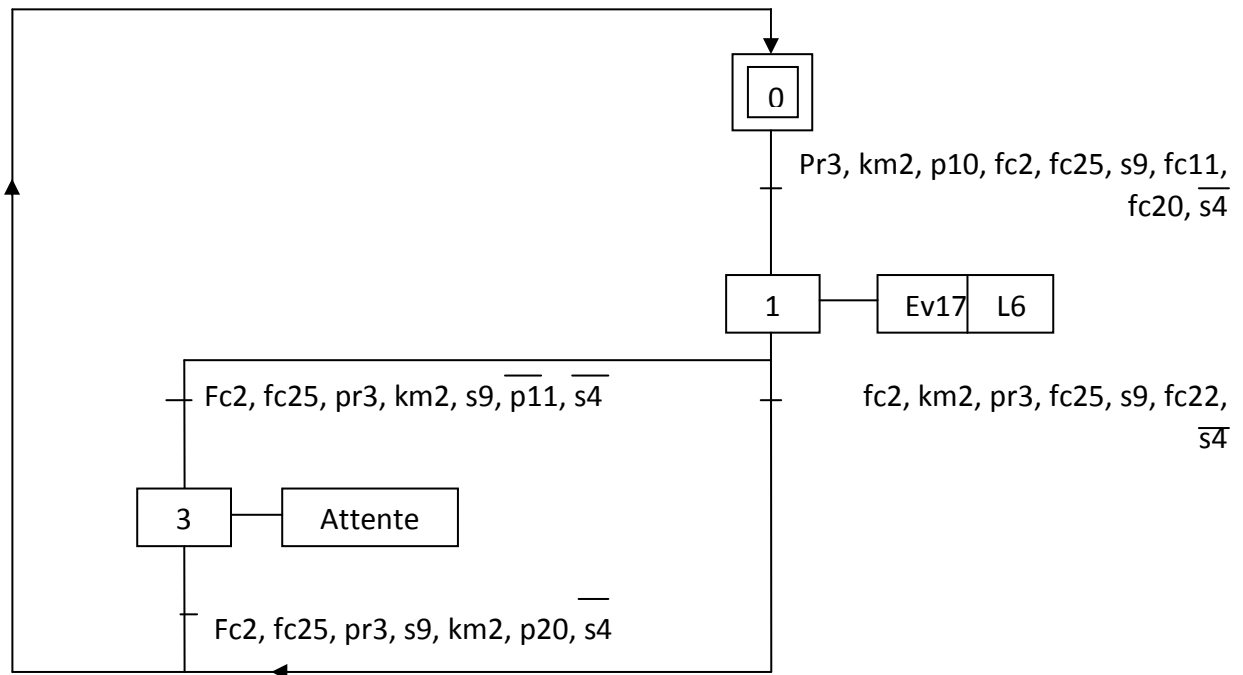


Figure 3.15 GRAFCET "avance travail" fraisage droit

La figure 3.16 donne un aperçu sur le programme du fraisage dans OB1.

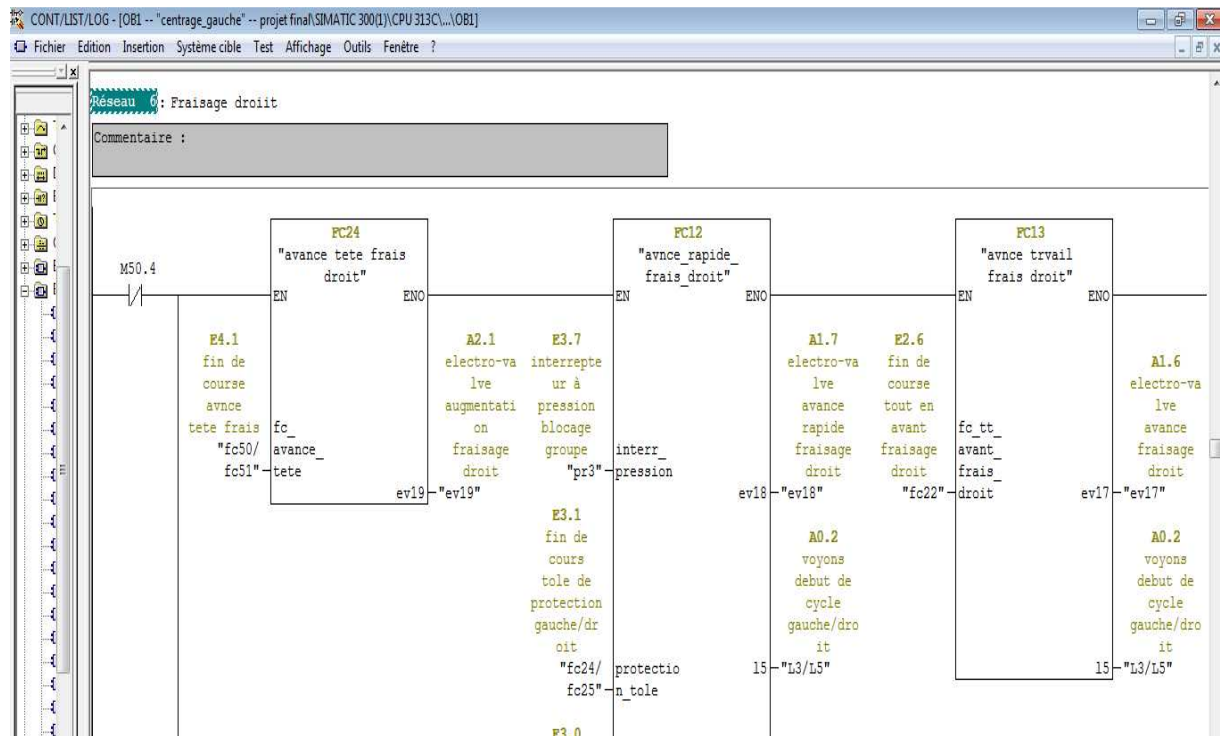


Figure 3.16 Fraisage droit [9]

3.5 Windows Control Center [10]

WinCC flexible est un système IHM performant utilisable sous Microsoft Windows 2000 SP3 ou SP4 et Windows XP pro SP1 ou SP2. Le contrôle est assuré par les automates programmables. Une communication s'établit donc entre WinCC flexible et l'opérateur d'une part et les automates programmables d'autre part. La figure suivante (figure 3.17) représente WinCC flexible et son entourage.

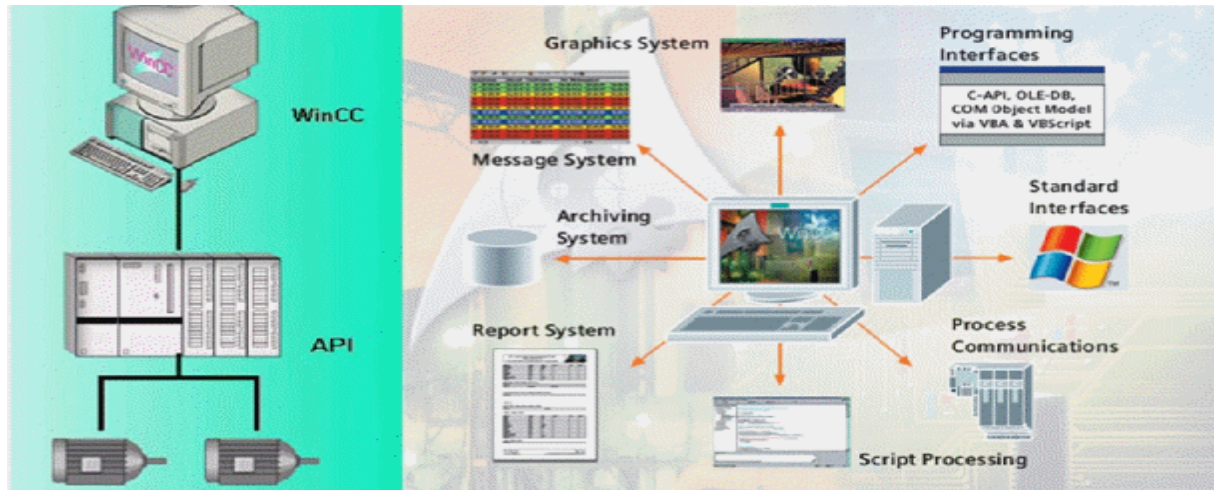


Figure 3.17 Entourage de WinCC [10]

3.5.1 Avantage du WinCC flexible

WinCC flexible permet à l'opérateur de surveiller le système. Pour ce faire, ce dernier est visualisé sur l'écran. Il permet aussi de commander le système à partir de l'interface utilisateur graphique. Il fait partie du concept TIA de Siemens (Totally Integrated Automation). Il s'avère particulièrement efficace dans le cadre d'une mise en œuvre avec des automates programmables de la famille SIMATIC. Les automates programmables d'autres marques sont également pris en charge.

Il s'adapte de façon optimale aux exigences de tout type d'installation et supporte de nombreuses configurations qui s'étendent du système monoposte aux systèmes répartis à plusieurs serveurs en passant par les systèmes client-serveur.

3.5.2 Principales fonctions offertes par WinCC flexible

- **WinCC flexible RT** : Gestion centrale du projet pour un accès rapide à toutes les données projet et aux paramètres centraux.
- **Communication** : Déclaration et adressage des variables. Les variables peuvent être internes ou externes.
- **Vue** : Système graphique qui permet la visualisation librement configurable et le pilotage via des objets entièrement graphiques.

- **Historique** : Compression et archivage des mesures et des alarmes.
- **Gestion des alarmes** : Système de messages pour la saisie des alarmes analogique et des alarmes TOR qui permet la visualisation des alarmes.
- **Paramétrages** : Système qui fait les paramétrages des alarmes.
- **Journaux** : Systèmes de journalisation pour la documentation à déclenchement temporel ou événementiel de messages, d'actions opérateur et de données de processus temps réel sous forme de rapport utilisateur ou de documentation de projet avec mise en page au choix.
- **Gestion utilisateur runtime** : Sécurité et paramétrage de runtime.

Autres outils : Recettes, Scripts, listes de textes graphique, Dictionnaires, structures, Gestion des versions, Scripts, Diagnostics, paramétrage des pupitres, localisation.

3.6 Simulation

En l'absence de l'automate, nous avons utilisé un logiciel optionnel de step7, PLCSIM qui permet de simuler un automate de la famille SIEMENS avec tous ces modules.

Notre simulateur présente une interface simple. En effet, pour changer l'état d'une entrée, il suffit de cocher la case correspondante. Les états des sorties changent automatiquement selon l'évolution du programme lors de la simulation et dans la fenêtre de programmation, où chaque contact représente une variable. Ceci permet de suivre l'évolution du programme en détails.

La simulation nous a permis de tester les différentes situations que peut affronter le système de gestion des alarmes de production.

Exemple : Opération de centrage

Cette opération est réalisée en deux étapes : avance rapide puis avance travail.

Tout d'abord, nous avons commencé par établir les GRAFCET correspondants à ces deux étapes avant de passer à la programmation step7. Les figures 3.18 et 3.19 illustrent les GRAFCETS que nous avons développés :

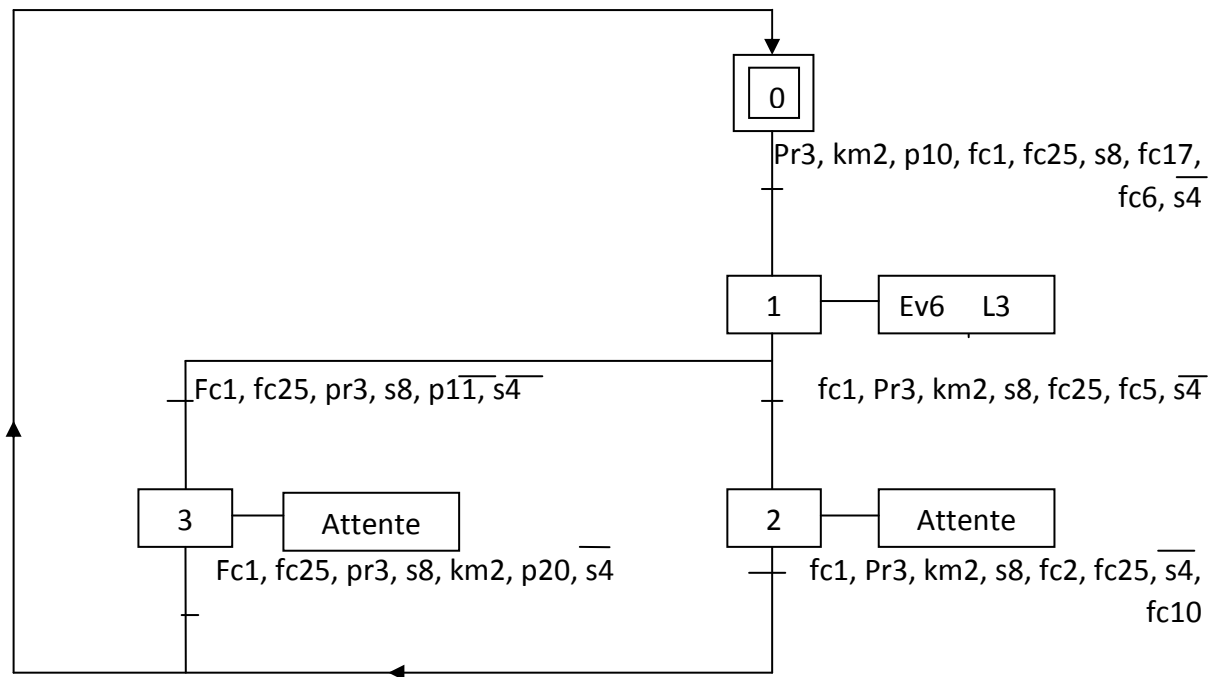


Figure 3.18 GRAFCET "avance rapide" centrage gauche

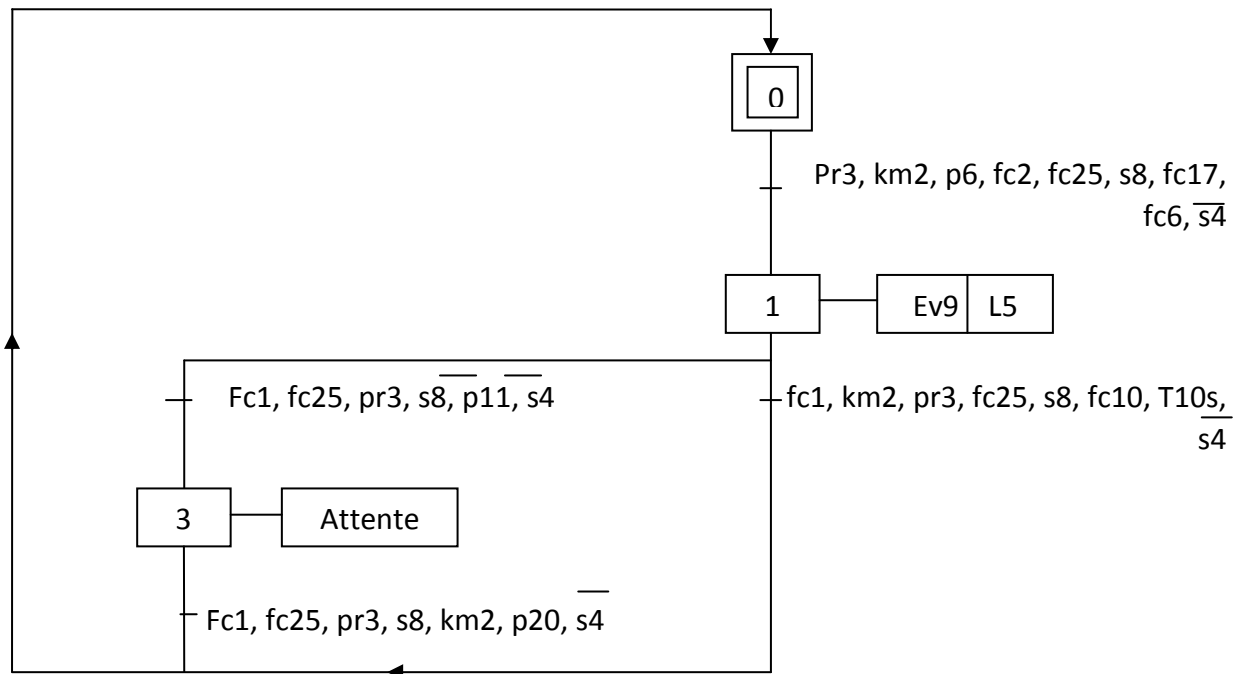


Figure 3.19 GRAFCET "avance travail" centrage gauche

Une fois les GRAFCETS réalisés, nous les avons convertis en programme LADDER afin de pouvoir réaliser la simulation sur le logiciel STEP7.

En même temps, nous avons développé notre propre panel pour la supervision des états de la machine.

Avant de commencer la simulation, il faut s'assurer que la liaison automate-IHM est bien établie, comme le montre la figure (3.20)

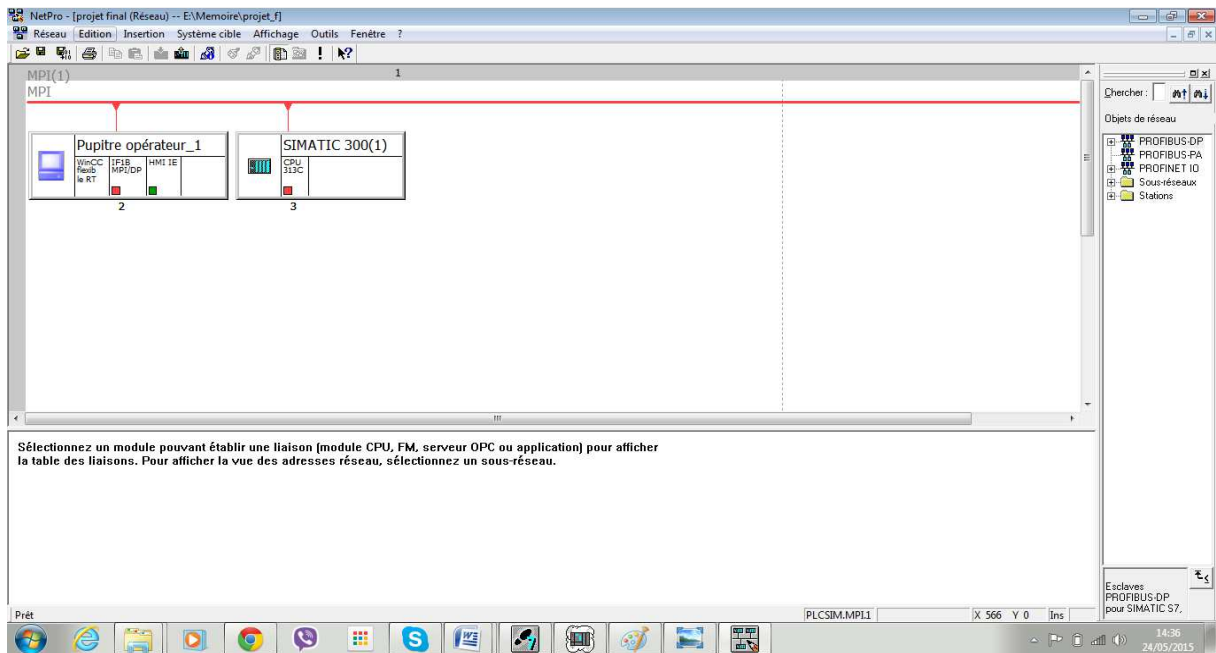



Figure 3.20 Vérification de la connexion dans le STEP7 [10]

Dans ce qui suit, nous allons montrer les différentes étapes de la simulation de l'opération de **centrage gauche** :

Un seul clic sur l'icône  , qui se trouve sur la barre d'outils est suffisant pour activer la simulation.

La fenêtre de simulation s'ouvre, comme le montre la figure (3.21)

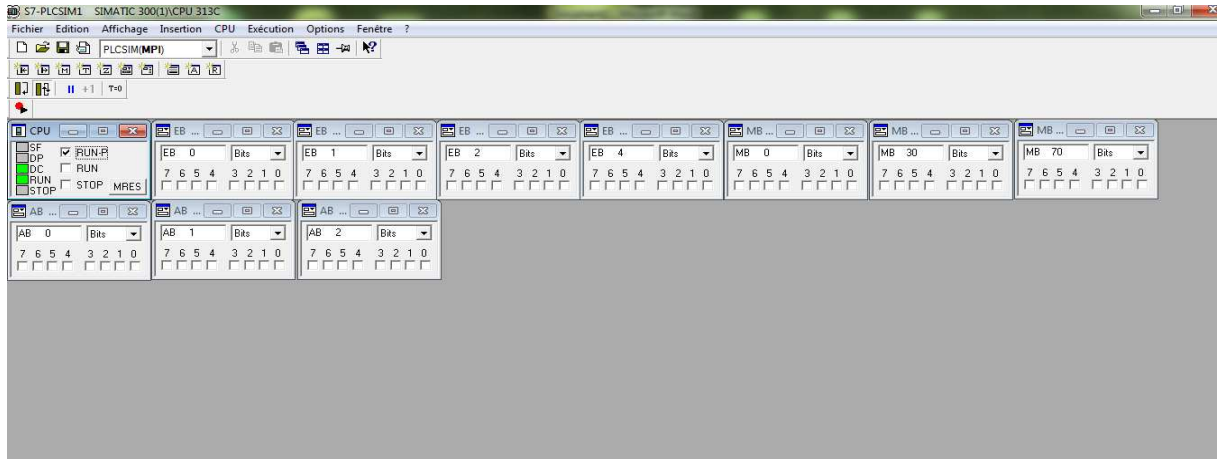



Figure 3.21 Fenêtre de simulation [9]



On clique sur RUN-P pour commencer la simulation.

Puis sur  afin de nous permettre la visualisation du programme, et passer en mode en ligne.

La figure 3.22 montre bien la simulation du programme

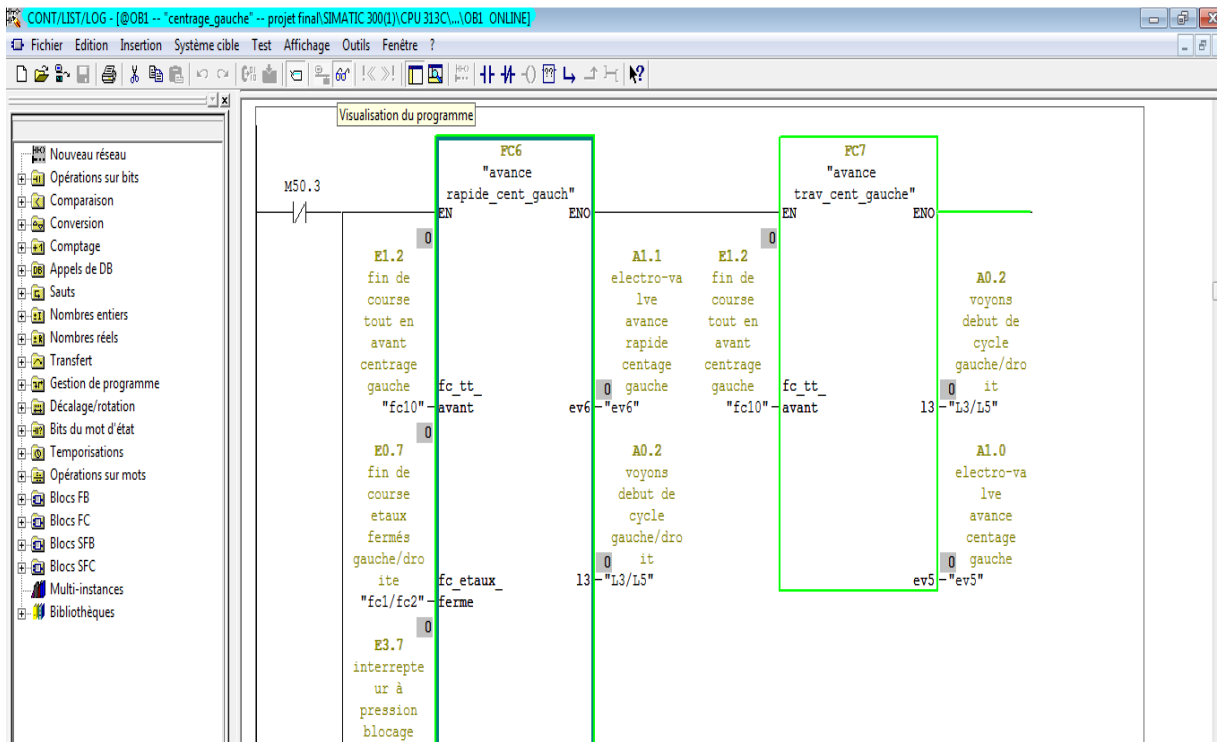


Figure 3.22 Mode de simulation

On laisse le simulateur du logiciel STEP7 en mode exécution et on va ouvrir le WinCC pour commander l'opération qu'on veut effectuer par notre machine.

La figure (3.23) montre la page d'accueil obtenue lors de l'ouverture de notre programme WinCC.

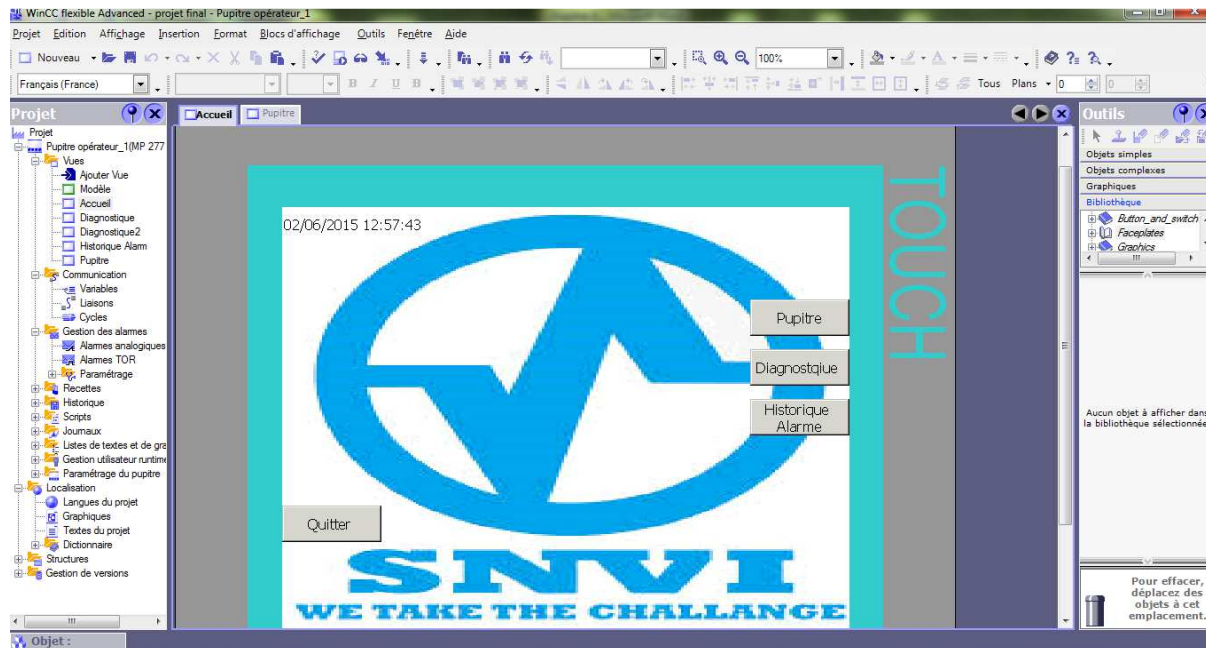


Figure 3.23 page d'accueil [10]

On clique sur  pour démarrer le système RunTime .

La fenêtre de Runtime s'ouvre et lance le mode simulation comme le montre la figure (3.24) suivante :



Figure 3.24 Vue d'accueil [10]

Pour commander les opérations souhaitées sur notre machine, on clique sur le bouton Pupitre présent sur la page d'accueil.

Une fois la fenêtre du pupitre ouverte, on règle les paramètres nécessaires pour l'opération du centrage gauche.

La régulation de l'opération centrage gauche est comme suit :

Il faut d'abord bloquer le groupe en mettant le commutateur sur **bloc** et aussi mettre le commutateur sur **Cent G**, puis on met le cycle sur **auto**, après on clique sur le bouton **Start** pour commencer l'opération.

La figure (3.25) ci-dessous montre la régulation de cette opération.

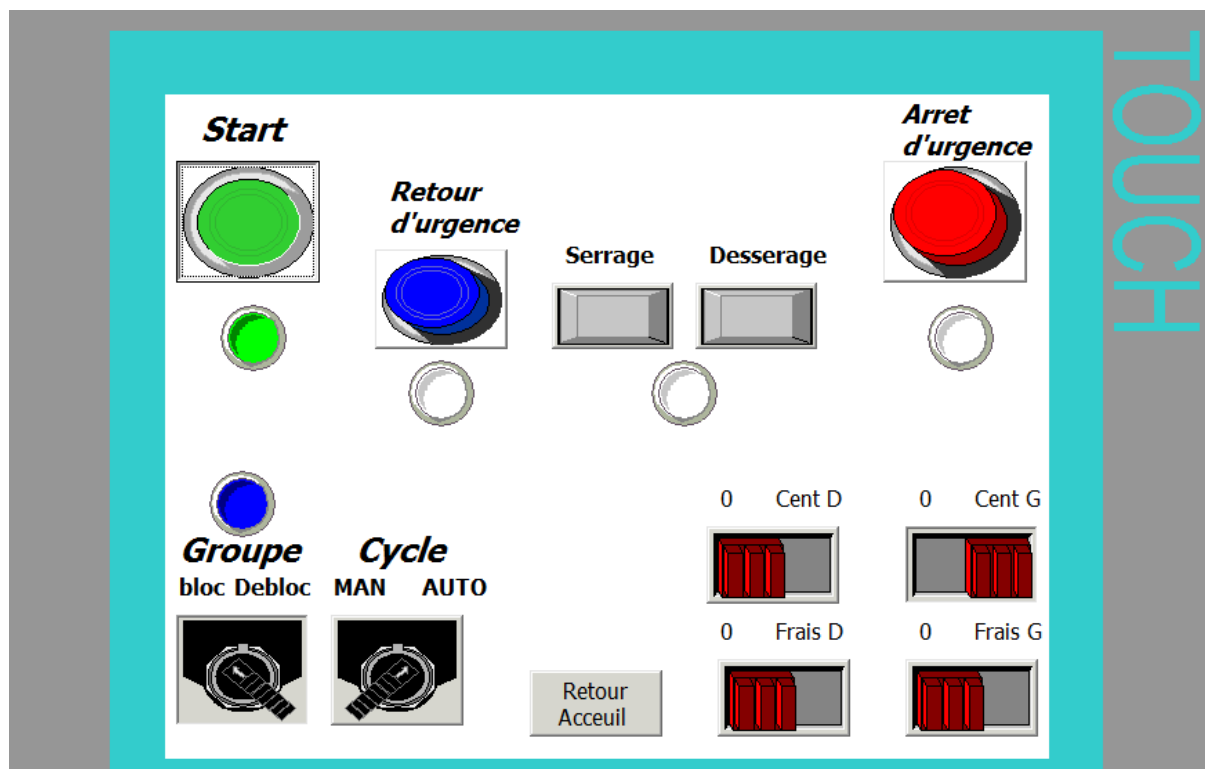


Figure 3.25 Vue du pupitre [10]

Une fois l'opération de centrage lancée, nous basculons sur le mode diagnostique afin de s'assurer que les E/S du programme fonctionnent bien et vérifier aussi les résultats obtenus. La figure (3.26) montre le bon fonctionnement de notre programme.

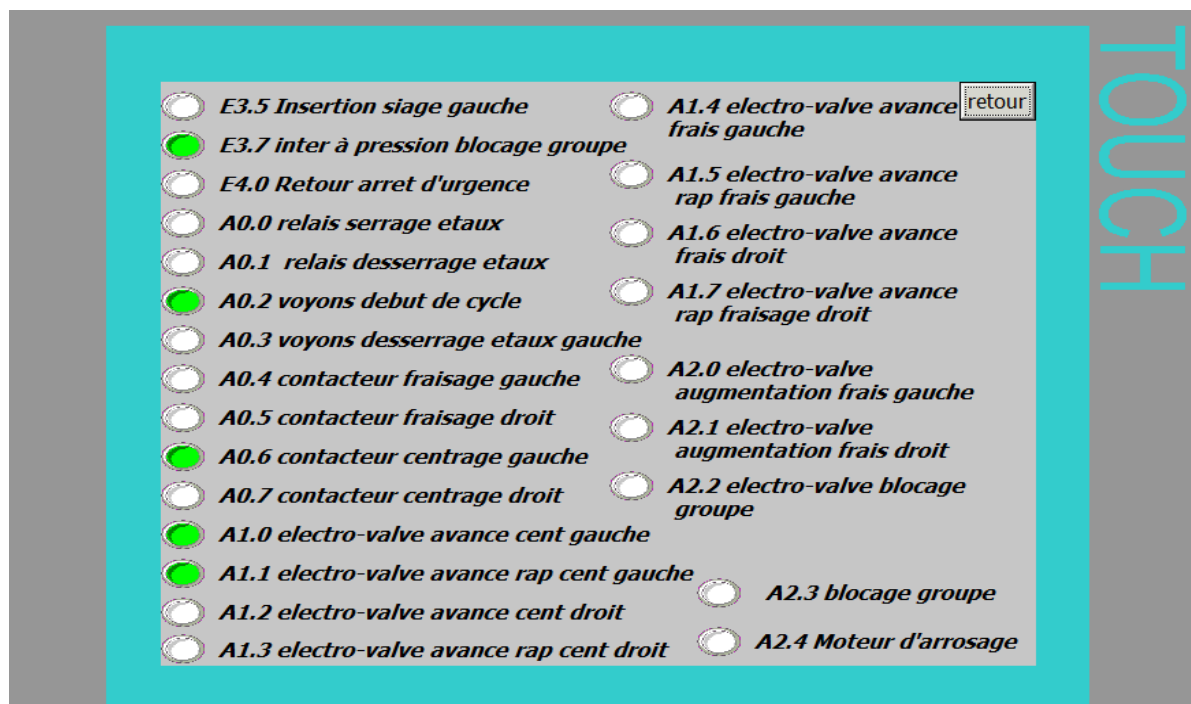


Figure 3.26 Vue de diagnostique [10]

Pour une bonne supervision de notre programme, nous avons inclus dans notre programme des alarmes et des avertissements afin de faciliter la détection de la panne ou de l'erreur éventuelle.

Ces alarmes peuvent être lues sur le pupitre ou sur l'historique des alarmes comme le montre la figure (3.27) suivante, cas où l'opérateur presse le bouton d'arrêt d'urgence.

N°	Heure	Date	Etat	Texte	GR
!	1	16:57:49	02/06/2015	(A)D arret ugence	0
!	1	16:57:49	02/06/2015	A arret ugence	0
!	1	16:57:10	02/06/2015	(A)D arret ugence	0
!	1	16:57:10	02/06/2015	A arret ugence	0
!	1	16:56:47	02/06/2015	(A)D arret ugence	0
!	1	16:56:47	02/06/2015	A arret ugence	0
!	1	16:51:42	02/06/2015	(A)D arret ugence	0
!	1	16:51:42	02/06/2015	A arret ugence	0
!	1	16:50:58	02/06/2015	(A... arret ugence	0
!	1	16:49:34	02/06/2015	(A)D arret ugence	0
!	1	16:49:34	02/06/2015	A arret ugence	0
!	1	16:49:28	02/06/2015	(A)D arret ugence	0
!	1	16:49:28	02/06/2015	A arret ugence	0
!	1	16:48:32	02/06/2015	(A)D arret ugence	0
!	1	16:48:32	02/06/2015	A arret ugence	0
!	1	16:48:23	02/06/2015	(A)D arret ugence	0
!	1	16:48:23	02/06/2015	A arret ugence	0
!	1	16:47:55	02/06/2015	(A)D arret ugence	0
!	1	16:47:55	02/06/2015	A arret ugence	0
\$	140000	16:05:55	02/06/2015	A Liaison établie : Liaison_1, station 3,...	0

Figure 3.27 Historique des alarmes [10]

De la même façon, nous avons développé d'autres fonctions nécessaires pour le fraisage gauche, fraisage droit et centrage droit et nous les avons intégrés à notre programme final. L'exécution finale de notre programme répond exactement aux exigences du cahier des charges et peut donc être transféré du PC vers l'automate qui lui correspond

3.7 Conclusion

Les solutions programmées nous procurent plusieurs avantages tels que la flexibilité, la facilité d'extension de ces modules et la possibilité de visualisation du programme établi avant son implantation sur l'automate réel grâce à son logiciel de simulation S7-PLCSIM. Ce dernier nous a donc permis de tester la solution programmée que nous avons développée et de valider et visualiser le comportement des E/S.

Liste des acronymes

API : Automate Programmable Industriel.

CPU : Computer Process Unit.

DVI : Division Véhicules Industrielles.

DB : Bloc de Données.

E/S : Entrée/Sortie.

FBD : Function Block Diagram.

FC : les Fonctions.

HMI : Human-Machine Interface.

I/O : Input/Output.

IL : Instruction List.

LD : Ladder Diagram.

MPI : Multi Point Interface.

OB : Organisation Bloc.

PLC : Programmable Logic Controller.

PC : Console.

PG : La console de programmation.

Liste des acronymes (suite)

PLCSIM : simulateur de logiciel step7.

RAM :Memoire Vive.

ROM : Memoire Morte.

SNVI : Société National Véhicules Industriels.

SONACOME : SOciété NATional de COnstruction MEcanique.

S5 100 U : Un automate de la marque siemens S5.

S7-300 : un mini automate de conception modulaire de la famille SIMATIC.

ST : Structured Text.

SM :Signals Module.

SIMATIC : Console de programme.

S7 : STEP7.

TOR : Tout Ou Rien.

TIA : Totally Integrated Automation.

UAL : Unité Arithmétique et Logique.

UL : Unité Logique.

WinCC : Windows Control Center .

Introduction Générale

Tout au long de l'histoire, l'homme a toujours cherché à faciliter son travail dans tous les domaines de la vie. L'automatisation est une des solutions les plus impressionnantes de nos jours.

Grâce au développement de l'informatique et de la microélectronique, l'automatisation est assurée par des automates programmables industriels qui intègrent ces nouvelles technologies.

L'apparition des **API** (Automates Programmables Industriels) a permis le développement des systèmes de production en vue de réduire les pannes, tout en augmentant la sécurité de l'environnement du travail et la flexibilité. Les **API** sont utilisés à tous les stades du processus productif comme la production, le contrôle de qualité des produits, etc.....

Dans le cadre de notre projet de fin d'étude, nous avons effectué un stage pratique au sein de la société nationale **SNVI-ROUIBA**. Celle-ci dispose de divers équipements automatisés qui datent des années 70. Ces installations sont actuellement vieillissantes et présentent plusieurs inconvénients.

C'est pourquoi, les responsables du complexe de véhicules industriels(**DVI**), unité de la société national **SNVI**, nous a proposé le changement de la commande actuelle de la machine centreuse pilotée par un automate **100U** (programmé en LIST) par un automate **Simatic 300** de nouvelle génération, pour faciliter l'opération d'entretien et de maintenance en respectant le cahier de charge délivré par l'entreprise.

Notre mémoire est organisé comme suit :

- Un premier chapitre consacré à des généralités sur les API et la description de la machine étudiée.
- Un deuxième chapitre consacré à l'automate utilisé dans notre travail (SIMATIC 300) ainsi que la présentation du principe des HMI.
- Un troisième chapitre consacré aux résultats et simulation de notre programme.
- Une conclusion générale viendra clôturer notre mémoire.

Bibliographie

- [1] PatrickJackard, 'Automates programmables industriels ', EditionPyc Livres, 1 janvier 1993.
- [2] ' Les Automates programmables industriels API', <http://www.technologuepro.com/cours-automate-programmable-industriel/Les-automates-programmables-industriels-API.htm>.
- [3] 'Machintest M25 TOVAGLIERI s.r.l.', Documentation SNVI.
- [4] ' Machine à dresser et centrer Tovagliari M25', <http://www.exapro.fr/machine-a-dresser-et-centrer-tovagliari-m25-doccasion-p30405102/#/>.
- [5] Atemane Nadia et BilaidiSabha, 'Adaptation d'un API S7-300 sur la machine aléseuse GPS', PFE Ingénieurs d'état en Automatique, université Mouloud MAMMERI Tizi-Ouzou, 2011.
- [6] 'API S7-300 ' <http://www.industry.siemens.com/>.
- [7] ' L'aide du logiciel Siemens Step7 version 5.5', Logiciel de programmation step7.
- [8]' SIEMENS : Langage CONT pour S7-300/400', Logiciel de programmation step7.
- [9]'SIEMENS : Programmation avec STEP7', Manuel simatic step7, Logiciel de programmation step7.
- [10]'SIEMENS : SIMATIC WinCC flexible 2008', Logiciel de supervision et de visualisation WinCC.

Définition du GRAFCET

Le GRAFCET est un langage graphique qui sert à décrire, étudier, réaliser et exploiter les automatismes.

Le mot GRAFCET écrit en lettres capitales fait référence à l'outil de modélisation.

Lorsque le mot GRAFCET est écrit en minuscule, il fait alors référence à un modèle obtenu à l'aide des règles du GRAFCET.

Structure graphique du GRAFCET

Le GRAFCET représente graphiquement la dynamique d'un système d'une manière simple à comprendre par un ensemble d'étapes, de transitions et de liaisons.

1. Etape :

L'étape correspond à une situation élémentaire ayant un comportement stable, elle se représente par un carré repéré par un nombre, placé de préférence de la moitié supérieure.

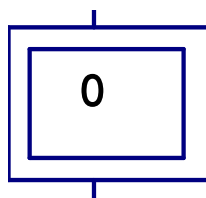


Figure: Étape initiale.

2. Action associée à l'étape :

On précise pour chaque étape les actions à effectuer et leur enchaînement lorsque l'étape est active, sont décrites de façon littérale (partie droite de l'étape) ou symbolique.

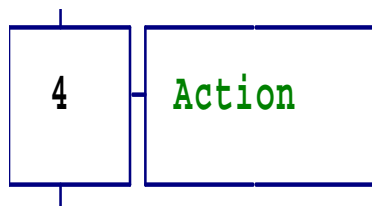


Figure: L'action.

3. Transition :

Les transitions indiquent la possibilité d'évolution d'une étape à l'étape suivante, à chaque transition est associée à une condition logique qui traduit la notion de réceptivité.

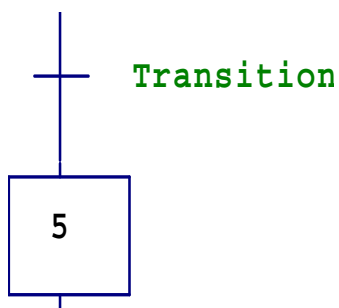


Figure : Transition

Règles d'évolution du GRAFCET :

Un grafcet possède un comportement dynamique dirigé par cinq règles, elles précisent les causes et les effets du franchissement des transitions.

Règle 1 : La situation initiale d'un grafcet caractérise le comportement initial de la partie commande vis-à-vis de la partie opérative et de l'opérateur.

Règle 2 : Une transition est dit validée lorsque toutes les étapes en amont (immédiatement précédentes à cette transition) sont actives.

Règle 3 : Le franchissement d'une transition entraîne simultanément l'activation de toutes les étapes immédiatement suivantes et la désactivation de toutes les étapes immédiatement précédentes.

Règle 4 : Plusieurs transitions simultanément franchissables sont simultanément franchies.

Règle 5 : Si au cours du fonctionnement, la même étape est simultanément activée et désactivée, elle reste activée.

Niveau d'un grafcet :

Grafcet niveau 1 : Il décrit le comportement de la partie commande vis-à-vis de la partie opérative et l'opérateur.

Grafcet niveau 2 : Pour décrire précisément comment l'automatisme devra physiquement s'insérer dans l'ensemble qu'il constitue avec son fonctionnement.

Grafcet niveau 3 : Il est connu sous le nom «point de vue automate», il affecte des consignes aux étiquettes d'entrée de l'automate et des ordres aux étiquettes de sortie.