

الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne démocratique et populaire

وزارة التعليم العالي و البحث العلمي
Ministère de l'enseignement supérieur et de la recherche scientifique

جامعة سعد دحلب البليدة
Université SAAD DAHLAB de BLIDA

كلية التكنولوجيا
Faculté de Technologie

قسم الآلية الهندسة الكهربائية
Département Automatique et Electrotechnique



Mémoire de Master

Filière : AUTOMATIQUE

Spécialité : automatique et informatique industriel

Présenté par

safsafi noureddine

&

abdelhine abdelkader

Etude et programmation d'une machine d'injection plastique et supervision

Proposé par : Pr ABDERREZAK . GUESSOUM

Année Universitaire 2020-2021

Remerciements

Tout d'abord, nous remercions Allah d'avoir terminé le projet de fin d'étude.

Nous tenons également à remercier nos parents pour toute l'aide qu'ils nous ont apportée pour notre succès dans cursus universitaire.

Nos remerciements vont également à tous les enseignants de L'Université SAAD DAHLEB BLIDA et particulièrement à ceux du département de Génie Electrique.

Aussi, nous adressons et en particulier, nos vifs remerciements les plus chaleureux à notre promoteur le professeur A. GUESSOUM de nous avoir encadré et nous donner de précieux conseils qui nous ont tant aidés.

Nos remercions les membres de jury qui nous feront l'honneur de juger ce travail.

A la fin nos remerciements l'équipe SLR et tous nos amis pour leur aide et leurs encouragements.

ملخص:

تم تنفيذ العمل المقدم في هذه الأطروحة في إطار مشروع نهاية دراسة المهندس في الآلية والمعلوماتية الصناعية في S.A.R.L SRL

يهدف هذا المشروع إلى دراسة وتصميم تشغيل آلة حقن البلاستيك Ma 4700 ii الهايتية لتحسين نموذج قديم. في الواقع ، يتم ضمان التحكم في هذا النموذج القديم في الوقت الحالي من خلال المنطق السلبي. الأعطال التي واجهها المنطق السلبي كانت متكررة وفي هذا السياق اقترحنا استبدال المنطق السلبي بواجهة API siemens s7-1200 للتغلب على هذه المشكلة.

كانت دراسة مفصلة شرط لا غنى عنه من أجل معرفة بأدق التفاصيل المراحل المختلفة للآلة وأعضاء التحكم والإعدادات من أجل أن تكون قادرة على إنشاء grafcet التشغيل الخاص بها تحديد المدخلات والمخرجات المختلفة للتحكم وبرمجة api التي يمكنها إدارة العملية بأكملها وتلبية توقعات المواصفات.

الكلمات المفتاحية: التحكم ، الإشراف ، آلة حقن البلاستيك

Résume :

L'étude présentée dans ce mémoire a été réalisée en tant que projet final de MASTER en automatique et informatique industrielle au sein de la S.A.R.L SLR Robinetterie.

Le but de ce projet est d'étudier et de concevoir les fonctionnalités de la machine d'injection plastique haïtienne MA 4700, ii afin d'améliorer son fonctionnement. En effet, le contrôle de cet ancien modèle est assuré pour le moment par la logique câblée.

Les dysfonctionnements que rencontrait la logique câblée étaient récurrents. Ainsi, il a été proposé de remplacer la logique filaire par un API Siemens s7-1200 afin de résoudre ce problème.

Cependant, une étude spécialisée de la presse s'imposait pour connaître dans les moindres détails les différentes phases de la machine, les unités de contrôle et les réglages afin de pouvoir établir son grafcet de fonctionnement, définir les différentes entrées et sorties de la section de contrôle et programmer l'API qui sera chargé de gérer l'ensemble du processus et de répondre aux attentes du cahier des charges.

Mots clés : automatique, machine injection plastique, api, grafcet

ABSTRACT:

The study presented in this memory was completed as part of the MASTER degree final project in automation and industrial computing at S.A.R.L SLR.

The goal of the above project is to study and design the functionality of the haitian ma 4700 ii plastic injection machine in order to improve the previous model. Indeed, the control of this old model is ensured at the moment by the wired logic

The dysfunctions that the wired logic encounters were recurrent. Thus, it was suggested to replace the wired logic by an api siemens s7-1200 in order to overcome this problem.

However, a specialized study of the press was a necessary requirement in order to know in the smallest details the various phases of the machine, the control units and the adjustments in order to be able to establish its operating grafcet to define the various entries and exits of the control section and to program the API which will be responsible for managing the whole process and to meet the expectations of the specifications.

Keywords: automatic, machine injection plastic, api, grafcet

Listes des acronymes et abréviations

VDE : vérin double effet

VSE : vérin simple effet

PWM : pulse width modulation

TOR : tout ou rien

GRAFCET : Graphe Fonctionnel de Commande par Etapes et Transitions

API : automate programmable industrielle

ST :Structure Texte

IL : Instruction List

FBD : fonction bloc diagramme

SFC : Séquentiel Fonction Charte

PC : Partie commande

PO : Partie Opérative

D : Delay

E/S : entrées/ sorties

CPM : capteur position moule

CPV : capteur position vis

CPE : capteur position éjecteur

CFP : capteur fin de cours ponton

CP : capteur fin de cours Porte

CFC : capteur fin de cours capot

CT : capteur température

CVR : capteur vérin de rotation

Table des matières

Introduction générale	1
Chapitre 1	2
1.1 Généralités sur SLR	3
1.1.1 Historique	3
1.1.2 Savoir faire	3
1.1.3 Vision	4
1.1.5 Hiérarchie usine SLR	6
1.1.6 Emplacement géographique.....	6
1.2 Introduction à l'injection	7
1.3 Présentation de la machine	7
1.4 Structure de la presse d'injection	8
1.5 Le procédé d'injection plastique	9
1.5.1 Définition du procédé	9
1.5.2 Principe	9
1.5.3 Déroulement du cycle d'injection	10
1.5.4 Les phases du procédé	10
1.6 Description d'une presse a injection	12
1.6.1 Unité de plastification ou injection	13
1.6.2 Unité de fermeture	17
1.6.3 Le moule	19
1.7 Les composants de la machine	21

1.7.1 Les Pré-actionneurs	20
1.7.2. Les actionneurs	22
1.7.3 Les Capteurs	25
1.8 Les avantages et les inconvénients de l'injection plastique	28
1.8.1 Les avantages	28
1.8.2 Les inconvénients	28
1.9 Conclusion	29
Chapitre 2 GRAFCET	30
Introduction	31
2.1 Définition	31
2.2 Description du GRAFCET	33
2.3 Concepts de base du GRAFCET	35
2.3.1 Etapes	35
2.3.2 Actions associées à l'étape.....	36
2.3.3 Transitions.....	37
2.3.4 Réceptivité associée à la transition	37
2.3.5 Liaisons orientées	38
2.4 Classification des actions associées aux étapes	38
2.4.1 Actions continues	38
2.4.2 Actions conditionnelles	39
2.4.3 Action maintenue sur plusieurs étapes	40
2.4.4 Action mémorisée	41
2.5 Règles d'évolution d'un GRAFCET	42
2.6 Structures de base	42
2.6.1 Séquence unique	42

2.6.2 Séquences simultanées	43
2.6.3 Sélection de séquences	43
2.6.4 Séquences exclusives	43
2.6.5 Saut d'étapes et reprise de séquences	44
2.6.6 Parallélisme interprété	44
2.6.7 Réutilisation d'une même séquence	45
2.6.8 Couplage entre séquences	45
2.6.9 Partage de ressources ou de séquences	46
2.7 Analyse d'un GRAFCET	46
2.7.1 Méthode de réduction.....	46
2.8 Propriétés des GRAFCET	49
2.9 GRAFCET d'automatisation d'injection.....	50
2.11 Conclusion	51
Chapitre 3 automate programmable industriel	52
Introduction :	52
3.2 Systèmes automatisés	52
3.2.1 Structure des systèmes automatisés	52
3.3 Généralités sur les API	53
3.3.1 Définition d'un automate programmable industriel (API)	53
3.3.2 Architecteur des automates	54
3.3.3 Aspect extérieur des API	55
3.3.4 Structure interne d'un automate	56
3.3.5 Description des éléments d'un API	57
3.3.6- Cycle d'un API	59
3.3.7 Critères de choix d'un API	60

3.3.8 Présentation de L'API SIEMENS S7-1200	61
3.4 Présentation du logiciel	65
3.4.1 Logiciel TIA portal	65
3.5 Création d'un projet	67
3.6 Configuration et paramétrage du matériel	68
3.6.1 Adressage des E/S	69
3.6.2 Adresse Ethernet de la CPU	70
3.7 Compilation et chargement de la configuration matérielle	71
3.8 Les variables API	72
3.8.1 Adresse symbolique et absolue	72
3.9 WINCC sur TIA PORTAL	73
3.10 Création du Programme	74
3.10.1 Structure de programmation TIA PORTAL.....	74
3.11 Conclusion	80
Chapitre 4 Supervision	80
Introduction	81
4.1 Définition de la supervision industrielle	81
4.2 Les avantages de la supervision	82
4.3 Structure d'un système de supervision :	82
4.4 Présentation WINCC	82
4.4 les vues de supervision de la machine injection plastique	83
4.5 conclusions	85
Conclusion général	86

Liste des figures

Liste des figures chapitre 1

<i>Figure1.1</i> Design	4
<i>Figure1.2</i> Technologie.....	5
<i>Figure1.3</i> Qualité.....	5
<i>Figure1.4</i> Hiérarchie d'usine SLR	6
<i>Figure1.5</i> Emplacement géographique usine SLR	6
<i>Figure1.6</i> Haitian machine MA 4700	8
<i>Figure1.7</i> Presse à injection horizontale	8
<i>Figure1.8</i> Déroulement du cycle d'injection	10
<i>Figure1.9</i> Fonctionnement Unité de plastification ou injection	12
<i>Figure1.10</i> Presse à injection plastique horizontale	13
<i>Figure1.11</i> Unité d'injection	14
<i>Figure1.12</i> Différents zone de la vis	15
<i>Figure1.13</i> La buse	15
<i>Figure1.14</i> Clapet anti-retour en position avant	16
<i>Figure1.15</i> clapet anti-retour en position arrière	16
<i>Figure1.16</i> Fermeture hydraulique	17
<i>Figure1.17</i> Fermeture simple genouillère	18
<i>Figure1.18</i> Fermeture double genouillère.....	18
<i>Figure1.19</i> Fermeture mixte	18
<i>Figure1.20</i> Présentation du moule	19
<i>Figure1.21</i> Les différents éléments d'un moule d'injection.	20
<i>Figure1.22</i> Etage de puissance	20

<i>Figure1.23</i> Illustration d'une Electrovanne	20
<i>Figure1.24</i> Schéma d'un contacteur.....	21
<i>Figure1.25</i> Vue en coupe d'un vérin	23
<i>Figure1.26</i> Vérins à double effet.	23
<i>Figure1.27</i> Servomoteur.....	24
<i>Figure1.28</i> Moteur hydraulique	25
<i>Figure1.29</i> Les Capteurs position	26
<i>Figure1.30</i> Capteur fin de cours	27

Liste des figures chapitre 2

<i>Figure2.1</i> Représentation d'un modèle type de GRAFCET.....	32
<i>Figure2.2</i> unité perçage.....	33
<i>Figure2.3</i> Point de vue Système.....	34
<i>Figure2.3</i> GRAFCET niveau 1.....	34
<i>Figure2.4</i> GRAFCET niveau 2.....	35
<i>Figure2.5</i> Situation des étapes.....	36
<i>Figure2.6</i> Actions associées aux étapes.....	37
<i>Figure2.7</i> Chronogramme de l'action continue.....	39
<i>Figure2.8</i> Chronogramme de l'action conditionnelle simple.....	39
<i>Figure2.9</i> Chronogramme de l'action retardée.....	40
<i>Figure2.10</i> Chronogramme de l'action de durée limitée.....	40
<i>Figure2.11</i> Action maintenue sur plusieurs étapes.....	41
<i>Figure2.12</i> Action à effet Maintenu par une Action Méorisée.....	41
<i>Figure2.13</i> franchissement d'une transition.....	42
<i>Figure2.14</i> Exemple de séquences simultanées.....	43

<i>Figure2.15</i> Exemple de réceptivités exclusives.....	44
<i>Figure2.16</i> Saut et reprise de séquences.....	44
<i>Figure2.17</i> Exemple de parallélisme interprété.....	45
<i>Figure2.18</i> Représentation d'une séquence réutilisable.....	45
<i>Figure2.19</i> Exemple de couplage de séquences.....	45
<i>Figure2.20</i> Exemple de partage de ressources.....	46
<i>Figure2.21</i> Exemple de suppression et de fusion d'étapes dans un GRAFCET.....	47
<i>Figure2.22</i> Exemple de suppression de transition dans un GRAFCET.....	48
<i>Figure2.23</i> Exemple de GRAFCET et son graphe des situations accessibles.....	48
<i>Figure2.24</i> Liaison entre GRAFCET.....	49
<i>Figure2.25</i> GRAFCET d'automatisation d'injection	50

Liste des figures chapitre 3

<i>Figure 3.1</i> : structure d'un système automatisé.....	53
<i>Figure 3.2</i> : Automate compact (Allen-Bradley).....	55
<i>Figure 3.3</i> : Automate Modulaire(Modicon).....	55
<i>Figure 3.4</i> : Automate modulaire (Siemens).....	56
<i>Figure 3.5</i> : Structure interne d'un API.....	57
<i>Figure 3.6</i> : Cycle d'un API.....	60
<i>Figure 3.7</i> : Automate S7-1200	62
<i>Figure 3.8</i> : Module extensions	63
<i>Figure 3.9</i> :Siemens CPU 1214C-6ES7 214-1BG40-0XB0.....	64
<i>Figure 3.10</i> : Création d'un projet.....	68
<i>Figure 3.11</i> : Paramétrage du matériel.....	69
<i>Figure 3.12-</i> : Adressage des E/S.....	70

<i>Figure 3.13 : Adresse Ethernet de la CPU</i>	71
<i>Figure 3.14 : Configuration matérielle</i>	72
<i>Figure 3.15 : Mode de connexion</i>	73
<i>Figure 3.16 : Vue de WINCC</i>	73
<i>Figure 3.17 : Programmation structurée de la machine injection plastique</i>	75
<i>Figure 3.18 : Structure du programme</i>	76
<i>Figure 3.19 : fonctionnement de résistance</i>	77
<i>Figure 3.20 : Table des variables E</i>	78
<i>Figure 3.21 : Table des variables S</i>	79
<i>Figure 3.22 : Simulation du bloc FC4</i>	79-80
<i>Figure 4.1 VUE MODE AUTOMATIQUE</i>	83
<i>Figure4.2 Vue mode manuel</i>	84
<i>Figure4.3 vue des alarmes</i>	84
<i>Figure4.4 vue configuration des alarmes</i>	85
Liste des tableaux :	
Tableaux entrées sorties de grafcet.....	51
Tableaux de caractéristique automate siemens s7-1200	64



Introduction générale

Au cours des dernières décennies, les procédés industriels ont connu un essor vertigineux, grâce à l'automatisme. Ainsi, la mondialisation des économies et l'accentuation de la concurrence poussent davantage les entreprises à automatiser leurs processus de production, afin d'assurer leur pérennité en s'offrant la compétitivité, tout en améliorant les conditions de travail de leur personnel ; en supprimant les tâches pénibles et répétitives.

Les automates programmables industriels apportent la solution sur mesure pour les besoins d'adaptation et de flexibilité de nombreuses activités économiques actuelles. Ils sont devenus aujourd'hui les constituants les plus répandus des installations automatisées.

Notre projet de fin d'étude sera réalisé au niveau de la SARL SLR Robinetterie, un projet qui consiste à faire l'étude et programmation d'une machine injection de plastique et la supervision.

Notre objectif consiste remplacer la logique câblée d'une machine d'injection plastique, de modèle ancien, par un API Siemens S7-1200.

Pour réaliser ce projet de fin d'étude le travail sera organisé comme suit :

- Dans le premier chapitre nous allons faire une présentation générale de l'usine SRL et le fonctionnement de machine d'injection plastique.
- Le deuxième chapitre sera consacré à la modélisation de la machine d'injection plastique avec l'outil GRAFCET,
- Le troisième chapitre est réservé à une introduction aux automates programmables et la réalisation de notre programme sur le logiciel TIA Portal.
- Le quatrième chapitre est dédié à la supervision de notre machine d'injection plastique

Chapitre 1 **Généralité sur usine SLR et fonctionnement de machine d'injection**

1.1 Généralités sur SLR :

1.1.1 Historique :

Créée en 2008, la SARL SLR Robinetterie a pour vocation principale la transformation de métaux, la production de barres de laiton, de robinetterie sanitaire et d'accessoires de raccordement pour plomberie, eau et gaz, elle s'est dotée d'un bureau d'études techniques équipé des moyens les plus performants en matière de développement et conception assistée par ordinateur.

1.1.2 SAVOIR FAIRE :

La SARL SLR maîtrise aujourd'hui plusieurs procédés de fabrication à savoir, l'estompage à chaud, la fonderie en coquille, l'injection, l'usinage à commande numérique et le traitement de surface. Le contrôle d'étanchéité est assuré unitairement pour tous ses produits sur des bancs d'essais. De ce fait, la SARL, s'est spécialisée dans :

- La fabrication de mitigeurs, mélangeurs et robinets pour baignoires, douches, lavabos, éviers, lave-mains, toilettes et bidets (Robinetterie sanitaire).
- La fabrication de pièces de raccordement pour installation en eau et gaz (vannes et robinets d'arrêts), aussi les pièces de raccordement multicouche à sertir et à raccord fabriqué pour la première fois en Algérie, avec un savoir-faire et une main-d'œuvre algérienne.
- La fabrication de robinets de gaz pour installation domestique.

Depuis sa création, SLR a réussi à développer un savoir-faire pour répondre à toute demande, en matière d'articles spécifiques, en plus de ses produits standards présentés sur leur catalogue. Dans le cadre de l'élargissement de ses activités, SLR s'est équipée d'une fonderie moderne, afin de produire, des barres en laiton pleines, creuses, et hexagonales, de différents diamètres et sections produites suivant des normes internationales. Ces barres de laiton de qualité supérieure

leur ont permis de concurrencer des fournisseurs de réputation mondiale et d'accéder à des marchés qui jusque-là leur étaient réservés.

1.1.3 Vision de l'entreprise :

Aujourd'hui, résolument tourné vers l'avenir, l'objectif est la pérennité de l'entreprise à travers une amélioration continue de ses performances, c'est dans ce sens que SLR s'est engagée dans un programme de diversification et de développement de ses activités soutenue par une démarche innovante pour être dans un avenir proche le leader dans ses domaines d'activité.

A Valeurs SLR Robinetterie :



figure1.1 Design

A.1

Un grand sens du détail et un soin porté aux finitions de chaque pièce produite, l'esthétique est au service de la fonction.



Figure1.2 Technologie

A.2

Une usine à la pointe de la technologie, une politique HSE et sécurité aux standard internationaux, le bien-être de ses collaborateurs, de ses clients sont au centre de ses préoccupations.



Figure1.3 Qualité

A.3

Des contrôles qualité, des tests de résistance et durabilité, rien n'est laissé au hasard pour garantir des produits durables, solides, et robustes sur le marché.

1.1.4 Hiérarchie usine SLR :

L'usine fonctionne selon le diagramme de la figure suivante :

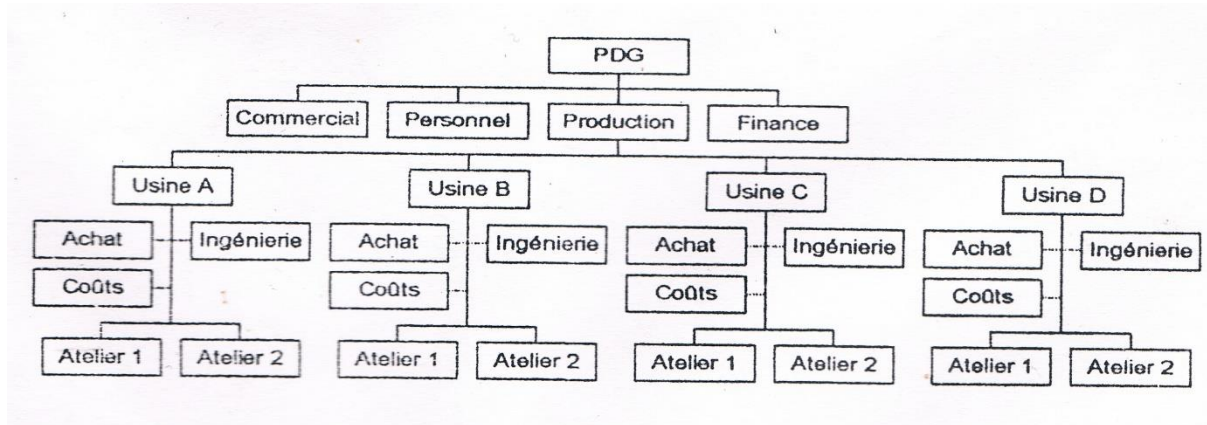


Figure 1.4 Hiérarchie de l'usine SLR

1.1.5 Emplacement géographique :

L'usine se situe à l'adresse : MAKTAO KHEIRA, LOT 7, GROUPE 9, Kolea, Tipaza (42), Algérie

Notre projet de fin d'étude, nous l'avons fait dans une succursale de SLR, qui est située à Beni Tamou. L'usine produit des boîtes où sont placés les robinets.



Figure 1.5 Emplacement géographique de l'usine SLR

1.2 Introduction à l'injection :

L'injection est aujourd'hui l'un des procédés les plus utilisés dans le monde de l'industrie. C'est le choix premier de n'importe quel fabricant qui souhaite produire à grande échelle. Il est généralement utilisé pour des productions de grandes et très grandes séries allant jusqu'à plusieurs milliers de pièces pour l'automobile ou l'électroménager par exemple, ou pour des séries plus réduites en aéronautique. Le procédé d'injection permet d'obtenir une productivité élevée avec une très bonne reproductibilité des pièces.

De ce fait, le principe du procédé de moulage par injection consiste à injecter et transformer une matière de polymères thermoplastiques, préalablement chauffée est injectée dans un moule. Le polymère se solidifiant dans le moule créera un solide épousant la forme et les dimensions de l'empreinte du moule.

1.3 Présentation de la machine :

Notre machine est une presse horizontale Haitian ma 4700 ii. Elle est à fermeture à genouillère avec verrouillage hydraulique de moule. Les genouillères sont actionnées par un vérin et l'éjection est également assurée hydrauliquement. Le bloc de plastification est équipé d'une buse ouverte. La rotation à la vis est obtenue grâce à un vérin hydraulique rotatif transmettant son mouvement de rotation à la vis sans fin. Un vérin commande l'avance de la vis lors de l'injection et un autre déplace l'ensemble du bloc de plastification.

Le circuit hydraulique est équipé avec les pompes basses pression et haute pression entraînées par le moteur électrique. Fournissent l'huile sous pression. Le volume d'huile nécessaire au bon fonctionnement du circuit est stocké dans le réservoir et le refroidissement à circulation d'eau maintient la température du fluide à une valeur acceptable.

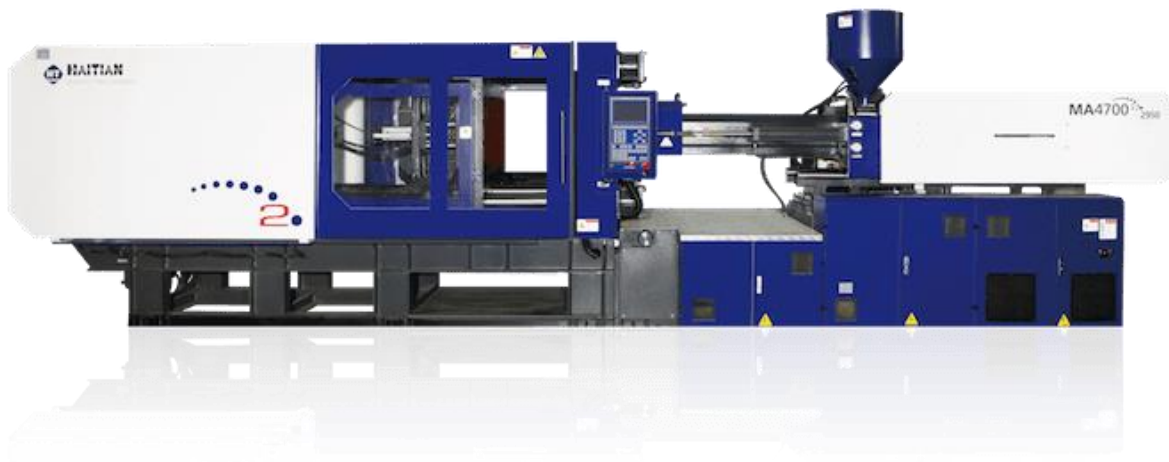


Figure1.6 La machine Haitian MA 4700

1.4 Structure de la presse d'injection :

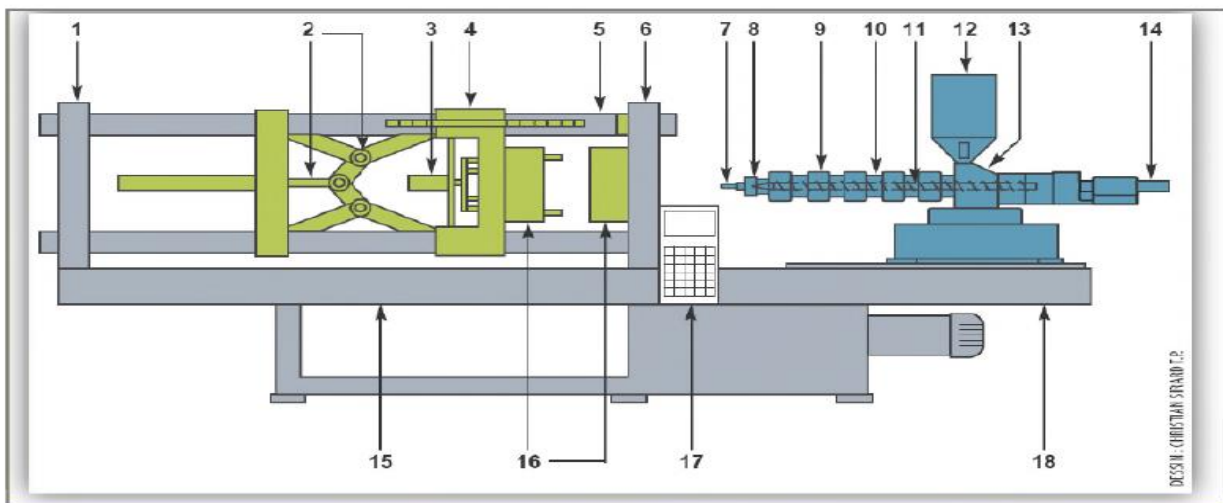


Figure1.7 Presse à injection horizontale

La machine est structurée comme suit :

1-plateau arrière fixe	10- baril d'injection
2-mécanisme de fermeture (ex : genouillère, vérin)	11- vis sans fin
3-éjecteur	12- trémie d'alimentation
4-plateau mobile	13- goulotte d'alimentation
5- colonne de guidage	14- motorisation de la vis
6- plateau fixe d'injection	15- évacuation des pièces
7- buse d'injection	16- moule
8- tête de baril	17- console de commande
9- collier chauffant	18- bâti

1.5 Le procédé d'injection plastique :

1.5.1 Définition du procédé :

Ce procédé d'obtention permet de produire très rapidement (en une seule opération) des objets de qualité en grande quantité, même si les formes sont complexes, dans des gammes de poids allant de quelques grammes à plusieurs kilogrammes. On utilise l'injection plastique dans de très nombreux domaines : l'électronique, la robotique, l'aéronautique, Téléphones, seringues, pare chocs, capots, carters, boîtes [8]...

1.5.2 Principe :

Le moulage permet de fabriquer un objet immédiatement utilisable, l'organe central de la machine est le moule contenant l'empreinte en creux de la pièce à réaliser. La machine doit permettre de remplir, cette forme avec la matière plastique choisie par l'utilisateur. La matière ne peut remplir, l'empreinte que sous forme fondue. Pour y parvenir, la matière doit être chauffée à une température supérieure au point de transition vitreuse ou au point de fusion. Sous cette

forme, la matière peut être contrainte de remplir le moule par plusieurs techniques ou procédés dont le moulage par injection [3].

1.5.3 Déroulement du cycle d'injection :

Le diagramme suivant décrit ce processus :

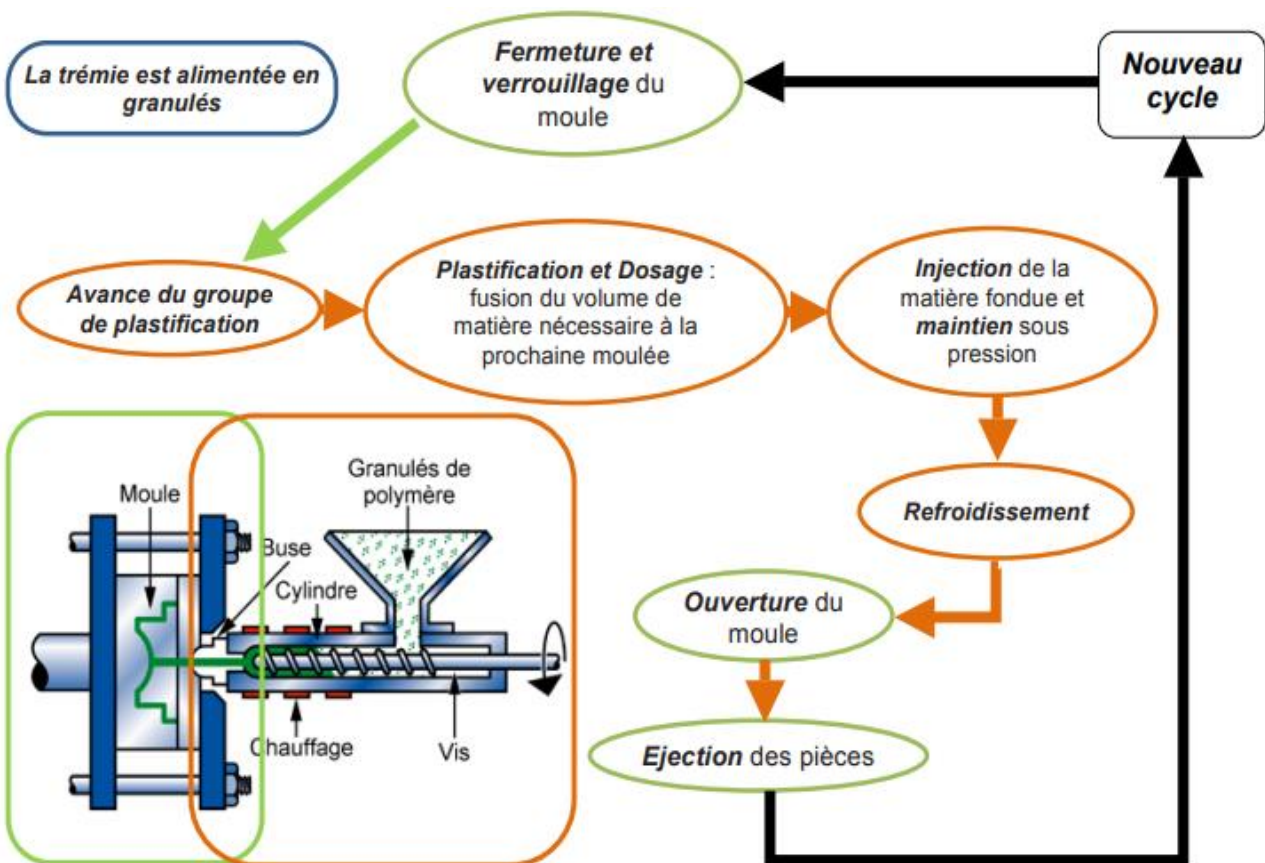


Figure 1.8 Déroulement du cycle d'injection [10].

1.5.4 Les phases du procédé :

Le procédé d'injection se déroule en cinq phases :

A La phase de dosage :

Les granulés de polymère tombent depuis une trémie dans un ensemble vis-fourreau. Ils sont ensuite fondus progressivement par actions conjuguées des colliers chauffant le fourreau, du cisaillement de la matière, provoqué par la rotation de la vis et la friction des granulés (entre eux et contre les parois du fourreau et de la vis). La matière fondue est alors convoyée en avant de la vis (via un clapet anti-retour), dans l'espace créé par le recul progressif de cette dernière lors de sa rotation.

B La phase d'injection :

Grâce à un mouvement de translation de la vis, le polymère fondu est transféré du réservoir à un moule via des canaux. Ce mouvement de translation est généré par un vérin, à travers le clapet anti-retour qui, dans cette phase, empêche la matière de rebrousser chemin.

C La phase de compactage :

Le moule est composé de deux parties, un fixe et un mobile. Durant le compactage, le moule est maintenu sous pression pour rester fermé, pendant que la matière est compactée par la vis. Pendant cette phase, le polymère fondu continue à être injecté pour compenser le rétrécissement, ou retrait, de la matière qui refroidit.

D La phase de refroidissement :

Lorsque le polymère est entièrement figé au niveau du ou des seuils d'injection, il n'est plus nécessaire d'appliquer une pression de maintien, et la pièce continue de se refroidir jusqu'à ce qu'elle soit complètement solidifiée. La vis reprend son cycle de rotation à la phase 1.

E La phase d'éjection :

Le moule s'ouvre, et la pièce formée est éjectée du moule .

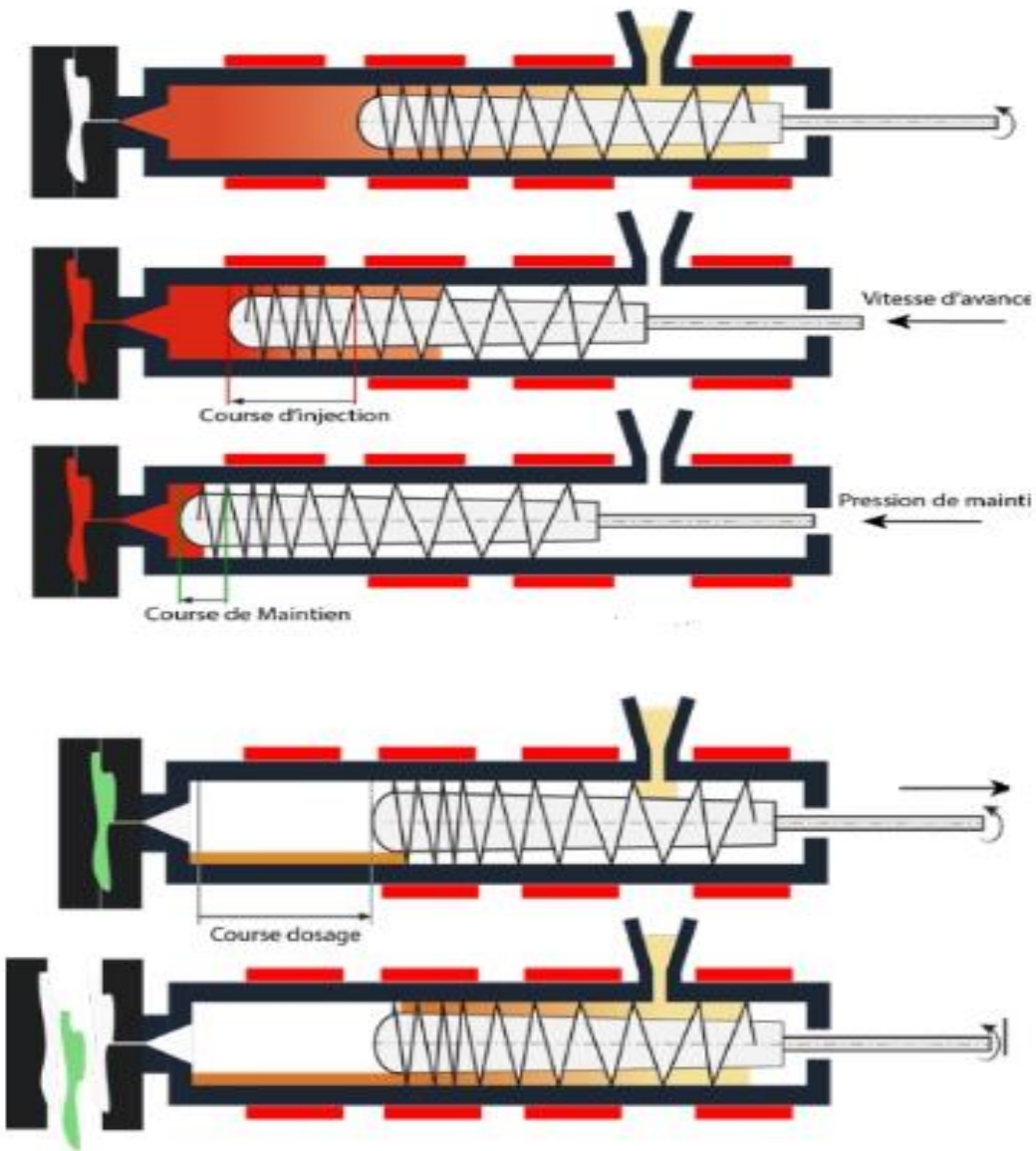


Figure 1.9 Fonctionnement de l'unité de plastification ou injection [2].

1.6 Description d'une presse à injection :

La presse à injecter se compose de plusieurs parties. Voici les principales : buse, plateau mobile, plateau fixe, collier chauffant, trémie d'alimentation, vis sans fin (ou vis de plastification),

système d'évacuation, mécanisme de fermeture et console. En outre, il est possible de catégoriser les éléments d'une presse à injecter en deux ensembles principaux [1].

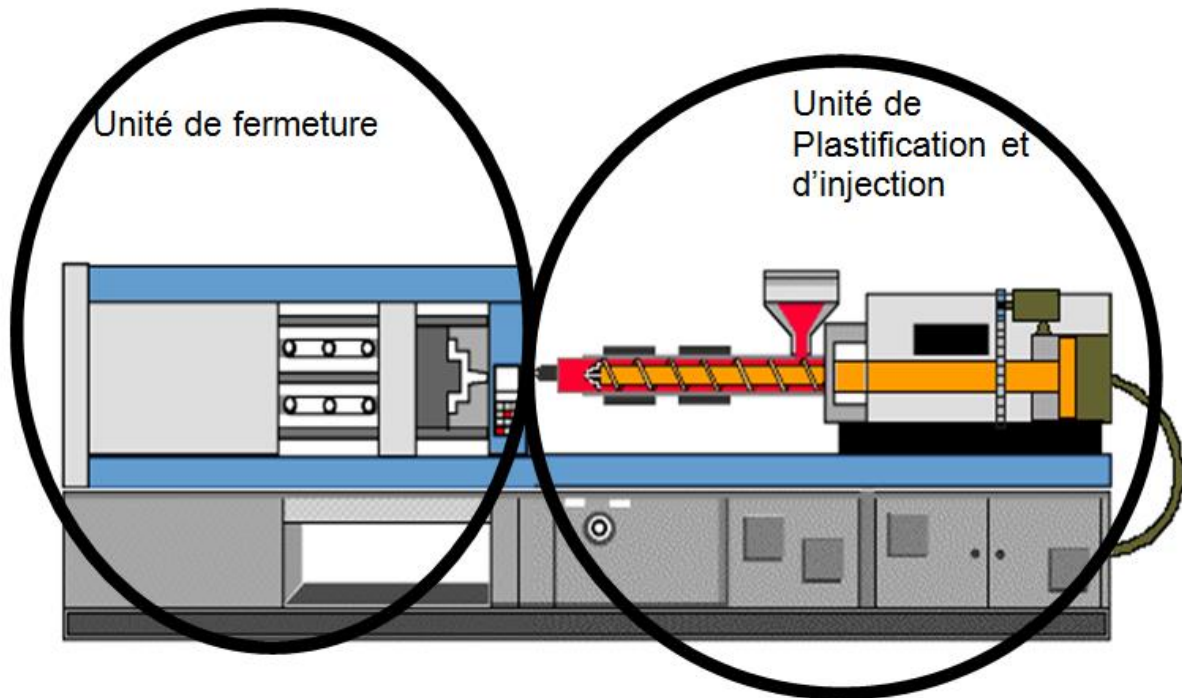


Figure 1.10 Presse à injection plastique horizontale

1.6.1 Unité de plastification ou injection :

L'unité d'injection est responsable de l'échauffement et de l'injection de la matière du polymère dans un moule. La première partie de cette unité est appelée trémie, elle est construite de façon à contenir une grande quantité de granulés du polymère. Pour le fonctionnement de la trémie, on entame un premier processus de l'opération et qui consiste à alimenter la machine en matière première. Une fois le mécanisme de ce système enclenché, on chauffe la matière jusqu'à liquéfaction du polymère. On assure ensuite l'injection du polymère « liquéfié ». La poussée de la matière est réalisée par un arbre filtré appelé vis avec des gradations de diamètre pour chaque longueur importante, ce dernier aide l'augmentation de la force de poussée de la matière jusqu'au seuil d'injection [11].

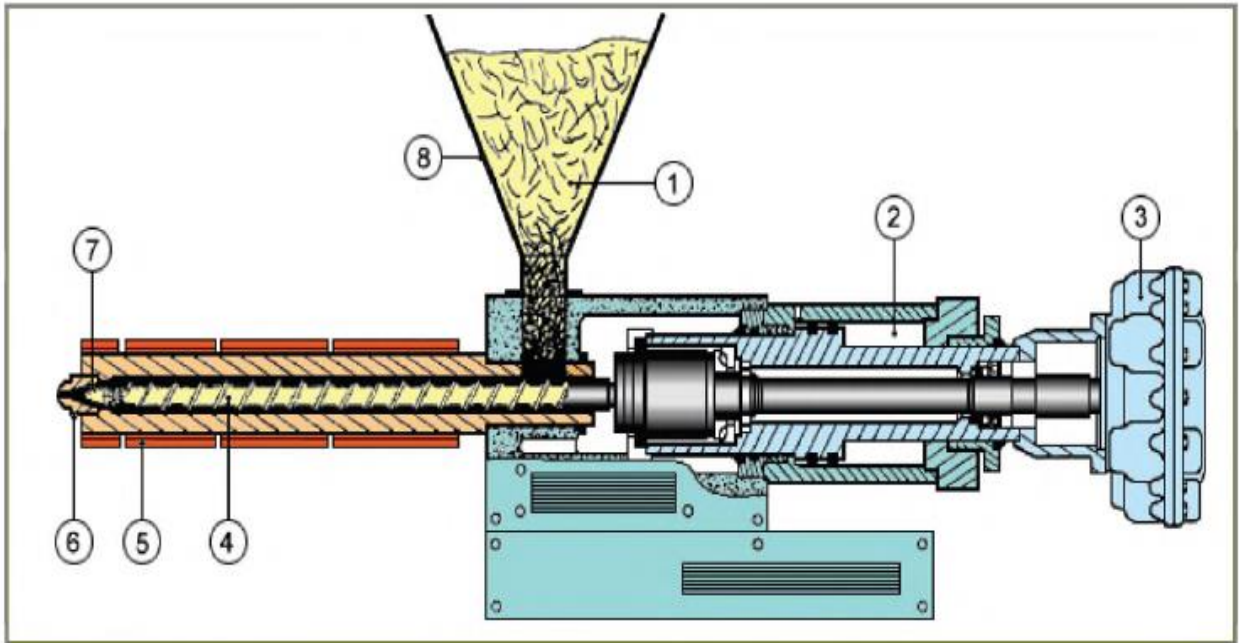


Figure1.11 Unité d'injection

Les différentes parties sont :

- | | |
|-----------------------------|------------------------|
| 1-Granulés. | 5- Fourreau chauffant. |
| 2-Vérin d'injection. | 6- Buse d'injection. |
| 3-Moteur de d'alimentation. | 7- Clapet. |
| 4-Vis sans fin. | 8- Trémie. |

A L'ensemble vis-fourreau :

La vis est située dans un fourreau cylindrique, mis en température à l'aide de résistances électriques ou colliers chauffants. Leur rôle est de transporter la matière du point d'alimentation (de la trémie) à la buse, plastifie grâce au malaxage et à la chaleur apportée de l'extérieur et enfin injecte la matière dans la moule à l'image d'une seringue [11].

La vis d'injection comprend en général trois zones :

A.1 Une zone d'alimentation : Cette zone reçoit la matière à l'état solide non compactée et la transporte vers la seconde zone. Le cœur de la vis est cylindrique : diamètre constant.

A.2 Une zone de compression : Le volume compris entre deux filets successifs diminue progressivement par augmentation du diamètre du cœur ou changement du pas. La compression expulse l'air et accélère la fusion du polymère.. Les deux premières zones d'une vis permettent de plastifier la matière.

A.3 Une zone de pompage : dans cette dernière partie la profondeur demeure constante, le polymère est mis en pression, s'homogénéise et la température de la matière y est proche de la température d'injection.

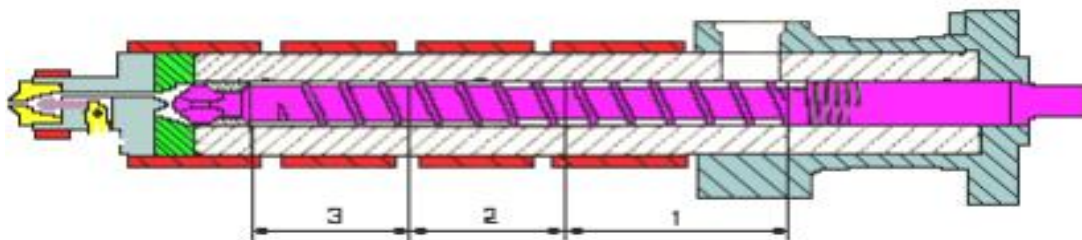


Figure1.12 Différentes zones de la vis.

B La buse :

Elle est fixée à l'avant de la vis. Elle doit assurer un contact étanche avec le moule durant l'injection, son est généralement sphérique ou tronconiques ou plans. Elle doit également présenter une surface de contact réduite avec le moule afin de limiter le refroidissement [11].

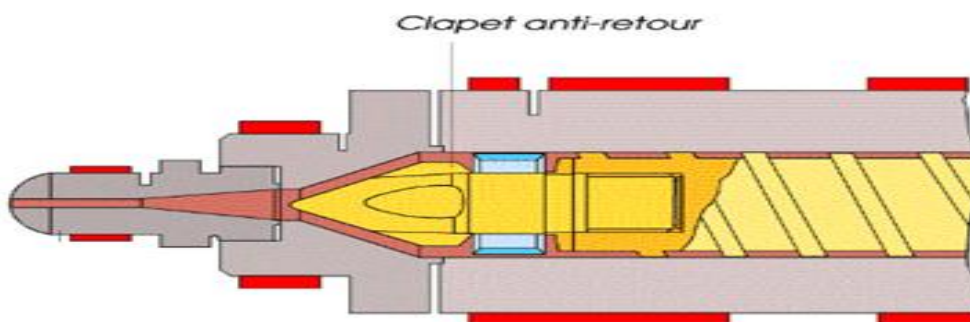
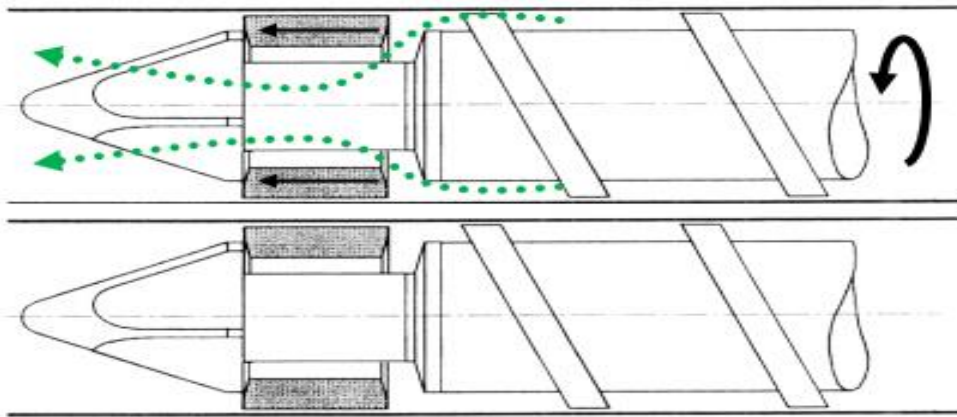


Figure1.13 La buse

C Le clapet anti-retour :

Un clapet anti-retour est vissé sur le bout de la vis côté plateau fixe pour que la matière ne retourne pas du côté trémie et que la pression soit intégralement du côté pièce[11].

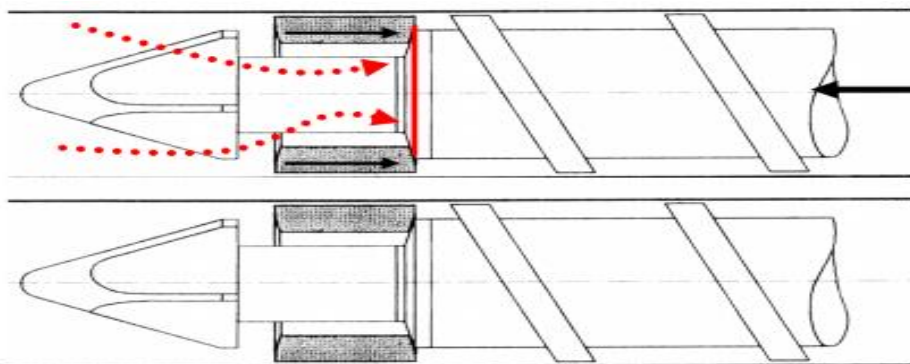
C.1 Plastification



La bague, poussée par la matière vient en appui sur la pointe

Figure1.14 clapet anti-retour en position avant.

C.2 Injection



La bague est refoulée sur le siège du clapet assurant l'étanchéité

Figure1.15 clapet anti-retour en position arrière.

1.6.2 Unité de fermeture :

Cet ensemble permet la fermeture, l'ouverture et le verrouillage de la partie mobile de la presse, sur la partie fixe. C'est un organe très important qui doit s'opposer à l'effort d'injection. Elle supporte le système d'éjection. Ainsi, cette unité peut être manœuvrée de plusieurs manières :

A Fermeture hydraulique :

Ce type de fermeture ne fait appel à aucun mouvement mécanique. Ceux-ci sont réalisés par des mouvements hydrauliques à l'aide d'un gros vérin central qui a pour but de faire l'approche du plateau mobile jusqu'au plateau fixe, et d'un vérin plus petit qui assure le verrouillage dans la phase final de la fermeture.

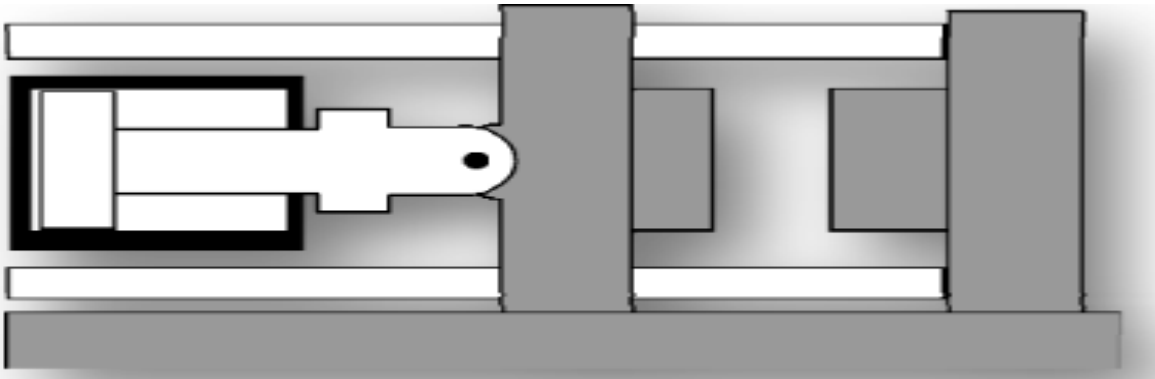


Figure1.16 Fermeture hydraulique

B Fermeture mécanique :

Bien que les mouvements sont assurés par un vérin, elle est appelée mécanique, car l'effort de verrouillage est assuré par les genouillères sur certaine presse lorsque le point d'alignement est dépassé, on peut couper la pression dans le vérin et l'ensemble reste stable.

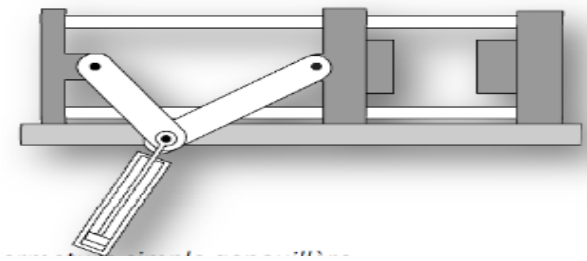


Figure 1.17 Fermeture simple genouillère

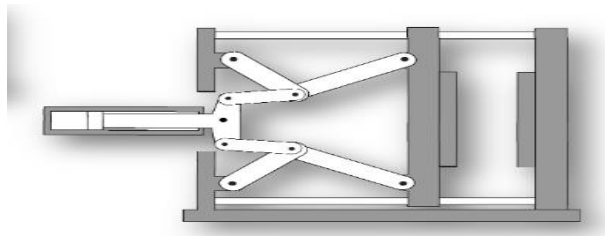


Figure 1.18 Fermeture double genouillère

C Fermeture mixte :

Ce procédé est un compromis entre la fermeture hydraulique et la fermeture mécanique, en effet les mouvements d'ouverture et de fermeture se font uniquement par des genouillères, tandis que le verrouillage est assuré par un ou des vérins hydrauliques.

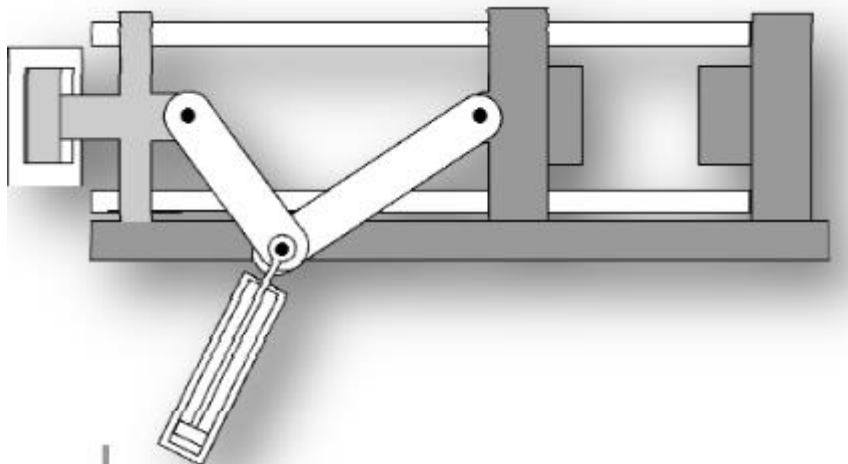


Figure 1.19 Fermeture mixte

1.6.3 Le moule :

Le moule, également appelé outillage, est un ensemble des pièces métalliques qui sont bien assemblées dans le but d'accomplir des fonctions bien déterminées, dont l'objectif est de fabriquer des pièces en plastiques d'une manière automatique. Cet outil est constitué d'une partie mobile pour la fermeture du moule et une partie fixe pour l'injection de la matière, séparée par ce qu'on appelle un plan de joint.



Figure1.20 Présentation du moule

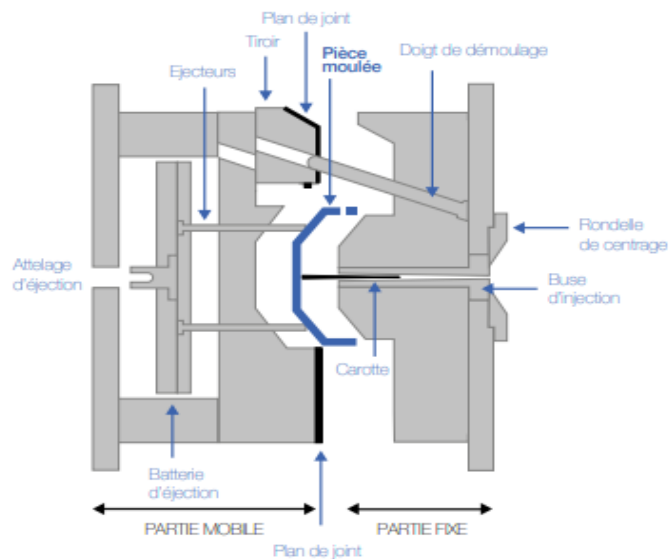


Figure1.21 Les différents éléments d'un moule d'injection.

1.7 Les composants de notre machine :

1.7.1 Les Pré-actionneurs :

Un pré-actionneur est un composant de gestion de l'énergie d'alimentation d'un actionneur, il reçoit l'information de la commande, puis, il délivre en sortie la puissance nécessaire pour l'actionner (figure 1.22).

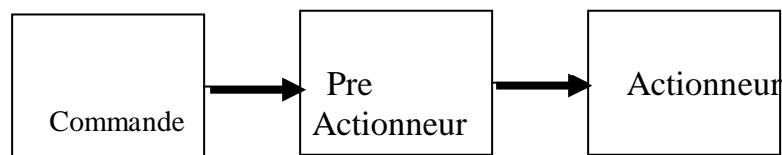


Figure1.22 Etage de puissance

A Electrovanes :

L'électrovanne est constituée d'une bobine électromagnétique et d'un robinet d'huile. L'électrovanne permet de bloquer ou de libérer la canalisation des vannes d'huile



Figure1.23 Illustration d'une Electrovanne.

B Les distributeurs :

Les distributeurs font partie de la famille des pré-actionneurs. Ils sont les constituants de la chaîne d'action qui permettent, à partir d'un ordre de la partie commande, de distribuer l'énergie de puissance aux actionneurs. Les distributeurs distribueront de l'énergie hydraulique aux vérins associés.

B.1 Principe de fonctionnement :

Ils ont pour fonction essentielle de distribuer le fluide dans des canalisations qui aboutissent aux chambres des vérins.

C Le contacteur :

Le contacteur est un appareil mécanique de connexion ayant une seule position repos, commandé autrement qu'à la main, capable d'établir, de supporter et d'interrompre des courants dans les conditions normales du circuit, y compris les conditions de surcharge en service. Un contacteur dont les contacts principaux sont fermés dans la position de repos est appelé rupteur. Il se compose d'une bobine qui est l'organe de commande, de contacts principaux et de contacts auxiliaires.

D Les contacts de puissance :

Ce sont eux qui sont chargés d'établir le circuit électrique. Ils comprennent une partie fixe et une partie mobile. La partie mobile est équipée de ressorts qui provoquent l'ouverture du contacteur à la mise hors tension. (figure 1.24) .

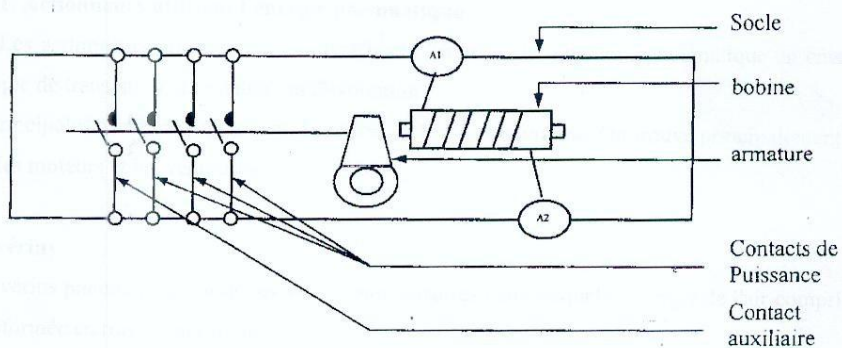


Figure1.24 Schéma d'un contacteur.

D.1 Le principe de fonctionnement :

Lorsque la bobine est alimentée, le contacteur se ferme et établit le circuit entre le réseau d'alimentation et le récepteur.

Lorsque la tension (A1.A2) est supprimée, les contacts reviennent à la position initiale par l'action d'un ressort de rappel.

1.7.2 Les actionneurs :

Dans un dispositif d'assistance ou d'un système de commande à distance (semi-automatique ou automatique), un actionneur est un organe de la partie opérative qui reçoit un ordre de la partie commande via un éventuel pré-actionneur, convertit l'énergie qui lui est fournie en un travail utile à l'exécution de tâches, éventuellement programmées d'un système automatisé

A Différents types d'actionneurs :

A.1 actionneurs utilisant l'énergie hydraulique :

Tous les appareils qui transforment l'énergie hydraulique en énergie mécanique sont des actionneurs. La plupart des usines utilisent l'énergie hydraulique pour accomplir un travail, soit, avec des outils portatifs ou avec des actionneurs.

Les vérins :

Les vérins permettent de déplacer des masses pour les positionner dans le but de réaliser des systèmes automatiques. L'énergie peut être hydraulique. Ils sont pilotés par des distributeurs. Comme vérin, on trouve principalement des vérins à simple effet ou à double effet.

Le vérin hydraulique :

C'est un récepteur linéaire dont le piston ne reçoit le débit en provenance de la pompe.

Constitution d'un vérin :

Les vérins sont généralement constitués d'un piston et d'une tige se déplaçant librement à l'intérieur d'un cylindre (figure 1.25) ;

Pour faire sortir la tige, il faut appliquer une certaine pression sur la face avant du piston, et

sur la face arrière pour faire rentrer la tige.

Dans notre système, il existe des vérins à double effet

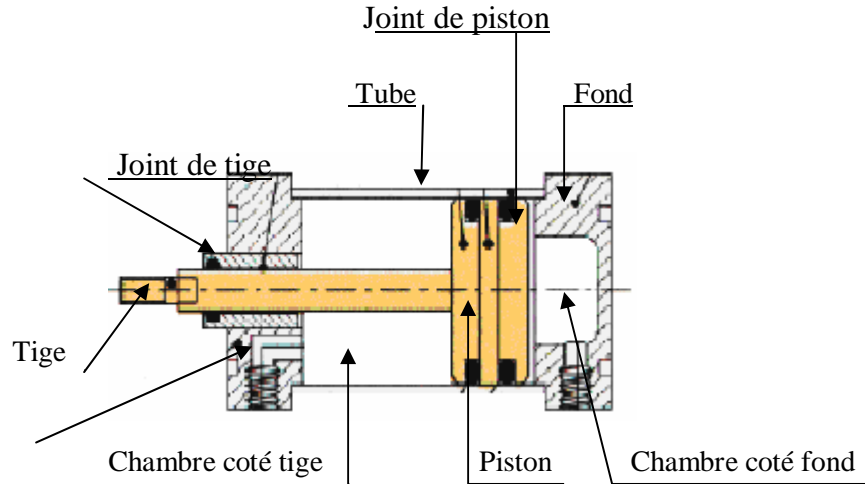


Figure 1.25 Vue en coupe d'un vérin

Vérins à double effet (VDE) :

Un vérin double effet a deux directions de travail. Il comporte deux orifices d'alimentation et la pression est appliquée alternativement de chaque côté du piston, ce qui entraîne son déplacement dans un sens, puis dans l'autre (figure 1.26) ;

L'effort en poussant (sortie de la tige) est plus grand que l'effort en tirant (entrée de la tige) car la pression n'agit pas sur la partie de surface occupée par la tige.

Les vérins sont souvent équipés d'amortisseurs de fin de course qui évitent les chocs du piston.

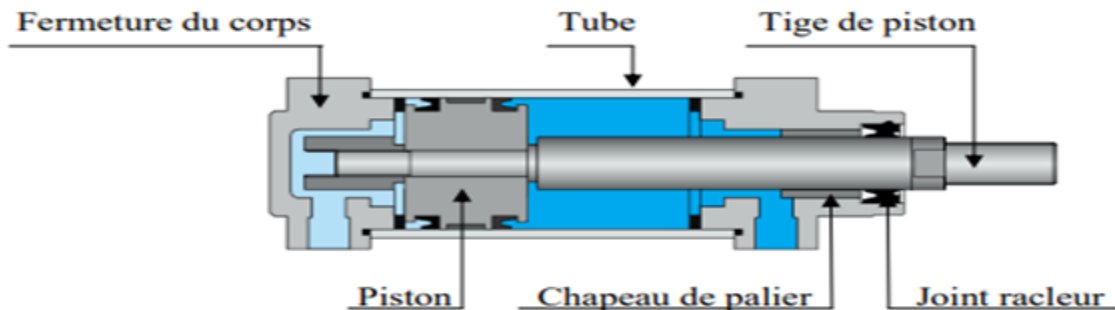


Figure 1.26 Vérins à double effet.

A.2 Actionneurs utilisant l'énergie électrique :

On trouve plusieurs types d'actionneurs électriques, dans notre système étudié, Il s'agit d'un servomoteur.

Fonctionnement de servomoteur :

Les servomoteurs sont commandés par l'intermédiaire d'un câble électrique à trois fils qui permet d'alimenter le moteur et de lui transmettre des consignes de position sous forme d'un signal codé en largeur d'impulsion plus communément appelé PWM.



Figure1.27 Servomoteur.

A.3 Actionneurs utilisant l'énergie hydraulique :

Moteur hydraulique :

Un moteur hydraulique convertit l'énergie hydraulique, sous forme de pression et de débit, en énergie mécanique rotative. Ces moteurs sont avantageux de par leur puissance massique élevée et leur couple de démarrage important.



Figure 1.28 Moteur hydraulique

1.7.3 Les Capteurs :

Les capteurs sont des composants de la chaîne d'acquisition dans une chaîne fonctionnelle. Les capteurs prélèvent une information sur le comportement de la partie opérative et la transforment en une information exploitable par la partie commande. Une information est une grandeur abstraite qui précise un événement particulier parmi un ensemble d'événements possibles. Pour pouvoir être traitée, cette information sera portée par un support physique, on parlera alors de signal. Les signaux sont généralement de nature électrique. Dans les systèmes automatisés séquentiels, la partie commande traite des variables logiques ou numériques. L'information délivrée par un capteur pourra être logique (2 états), numérique (valeur discrète), ou analogique. Dans ce cas, il faudra adjoindre à la partie commande, un module de conversion analogique numérique.

On peut caractériser les capteurs selon deux critères :

- En fonction de la grandeur mesurée. On parle alors de capteur de position, de température, de vitesse, de force, de pression, etc...
- En fonction du caractère de l'information délivrée. On parle alors de capteurs logiques appelés aussi capteurs tout ou rien (TOR).

A Différents types de capteurs dans notre système :

A.1 Capteur de position :

Une capture de position (aussi appelé capteur de mouvement) permet de connaître avec précision la position physique et/ou la vitesse de l'ensemble qui lui est rattaché à un instant t . Cette information est ensuite transmise à un système de calcul qui l'utilisera à des fins d'asservissement du système ou de contrôle des opérations.

La position physique peut prendre plusieurs formes :

il peut s'agir de l'emplacement du système dans un référentiel défini, ou bien de son état par rapport à un axe (en rotation ou en inclinaison).

La vitesse s'obtient simplement en dérivant cette position par rapport au temps.



Figure 1.29 Les Capteurs de position

A.2 capteur fin de course : Un interrupteur de fin de course est un dispositif mécanique qui nécessite le contact physique d'un objet avec l'actionneur de l'interrupteur pour faire changer l'état du contact (ouvert/fermé).

Fonctionnement :

Lorsque l'objet ou la cible entre en contact avec l'opérateur de l'interrupteur, il finit par déplacer l'actionneur jusqu'à la « limite » où les contacts changent d'état.

Dans un circuit normalement fermé (NF), cette action mécanique ouvre les contacts électriques et dans un circuit normalement ouvert (NO), elle les ferme. Les contacts mettent alors en marche ou arrêtent le débit de courant dans le circuit électrique.



Figure1.30 Capteur fin de cours

1.8 Les avantages et les inconvénients de l'injection plastique :

1.8.1 Les avantages :

Ces avantages sont :

La rapidité du processus.

La précision de la pièce produite.

L'économie de matière première.

Pièces à prix de revient unitaire relativement bas.

Adapté pour des pièces de géométries complexes.

1.8.2 Les inconvénients :

Changer manuellement le moule qui prend beaucoup de temps et nécessite de la force.

Conception relativement difficile de l'outillage (prix élevé de l'outillage).

Faible résistance à la chaleur, au choc et à l'usure.

1.9 Conclusion :

A travers ce chapitre, nous avons étudié la presse d'injection avec indication de ses composants et son mode de fonctionnement. Nous avons découvert la procédure de fabrication des pièces plastiques, en notant le déroulement du cycle d'injection. Dans le prochain chapitre de ce mémoire on va voir le grafcet de la machine injection

Introduction

L'écriture d'un programme consiste à créer une liste d'instructions permettant l'exécution des opérations nécessaires au fonctionnement du système. L'API traduit le langage de programmation en langage compréhensible directement par le microprocesseur. Ce langage est propre à chaque constructeur. Il est lié au matériel mis en œuvre. Chaque automate possède son propre langage. Cependant, les constructeurs proposent tous une interface logicielle répondant à la norme CEI 61131-3 qui définit cinq langages de programmation utilisables : le Grafcet, le langage LADDER, le langage ST (Structured Text), le langage IL (Instruction List) et le langage FBD (Boîtes fonctionnelles).

2.1 Définition

Le GRAFCET (GRAphe Fonctionnel de Commande par Etapes et Transitions) ou SFC (Sequential Function Chart) est un outil graphique qui décrit les différents comportements (cahier des charges) de l'évolution d'un automatisme et établit une correspondance à caractère séquentiel et combinatoire entre :

Les entrées : c'est-à-dire les transferts d'informations de la Partie Opérative vers la Partie Commande [4].

Les sorties : transferts d'informations de la Partie Commande vers la Partie Opérative.

C'est un outil graphique puissant, directement exploitable, car c'est aussi un langage pour la plupart des API existants sur le marché. Lorsque le mot GRAFCET (*en lettre capitale*) est utilisé, il 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.

Le GRAFCET comprend :

- **Étape initiale** : représente une étape qui est active au début du fonctionnement. Elle se différencie de l'étape en doublant les côtés du carré.
- **Transition** : la transition est représentée par un trait horizontal.
- **Réceptivité** : les conditions de réceptivité sont inscrites à droite de la transition.

- **Étape** : chaque étape est représentée par un carré repéré numériquement.
- **Action(s)** : elles sont décrites littéralement ou symboliquement à l'intérieur d'un ou plusieurs rectangles reliés par un trait à la partie droite de l'étape.
- **Liaisons orientées** : indique le sens du parcours.

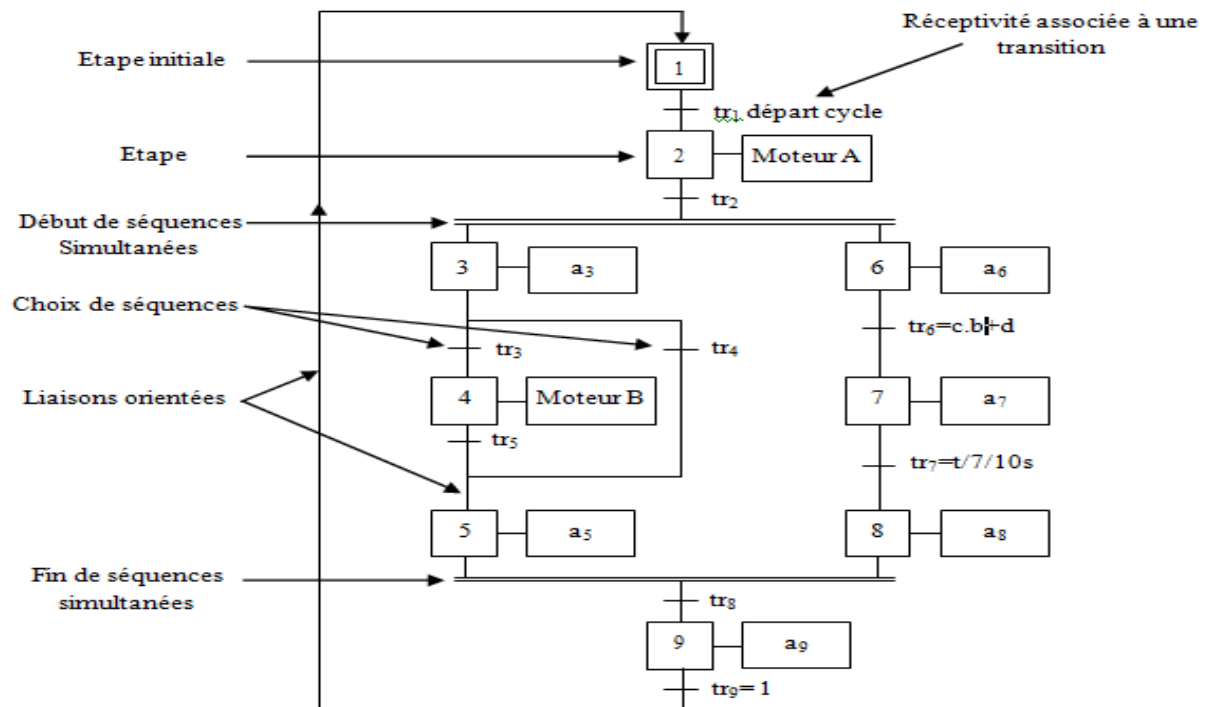


Figure2.1 Représentation d'un modèle type de GRAFCET.

2.2 Description du GRAFCET :

La description du comportement attendu d'un automate peut se représenter par un

GRAFCET d'un certain « niveau ». La caractérisation du niveau du GRAFCET nécessite de prendre en compte trois dimensions :

Le point de vue : caractérisant le point de vue selon lequel un observateur s'implique dans le fonctionnement du système pour en donner une description. On distingue trois points de vue :

Un point de vue système : C'est un graphe qui décrit le fonctionnement global du système. Il traduit le cahier des charges sans préjuger de la technologie adoptée. Il permet de dialoguer avec des personnes non spécialistes (fournisseurs, décideurs ...) Son écriture, en langage clair, permet donc sa compréhension par tout le monde.

- Un point de vue Partie Opérative : Dans ce type on spécifie la technologie de la PO ainsi que le type de ses informations reçues (ordres) et envoyées (comptes-rendus).
- Un point de vue Partie Commande : Ce type est établi la technologie des éléments de dialogue : entre PC et PO, PC et opérateur, PC et autre système. C'est la version qui permet d'établir les équations des schémas de réalisation (électrique, pneumatique). [4]

Exemple : Unité de perçage.

Les perçages sont effectués en même temps après action sur un bouton poussoir Départ Cycle.

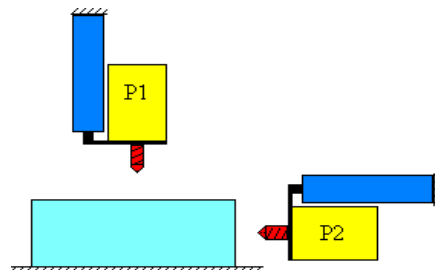


Figure2.2 unité perçage.

Point de vue "Système" :

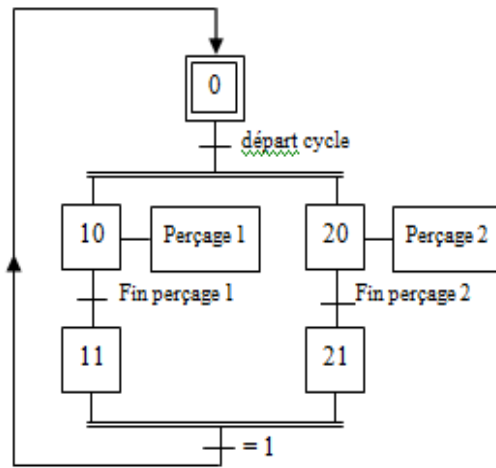


Figure 2.3 Point de vue Système.

Point de vue "Partie Opérative" (Niveau 1) :

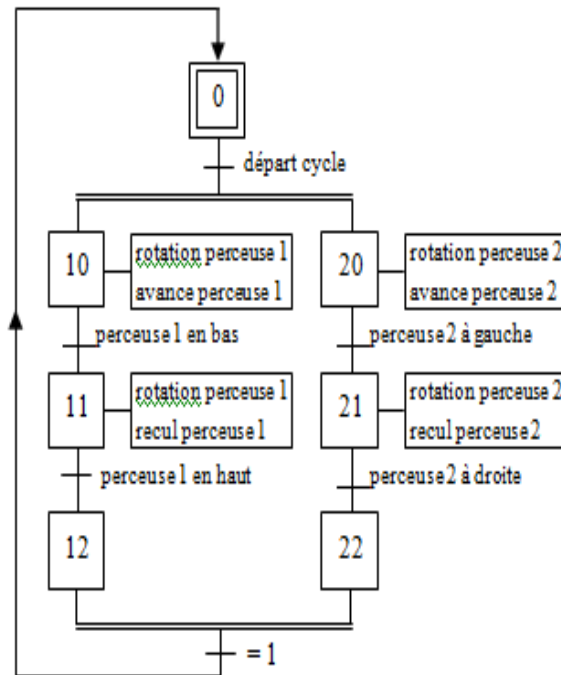


Figure 2.3 grafcet niveau 1.

Point de vue "Partie Commande" (Niveau 2) :

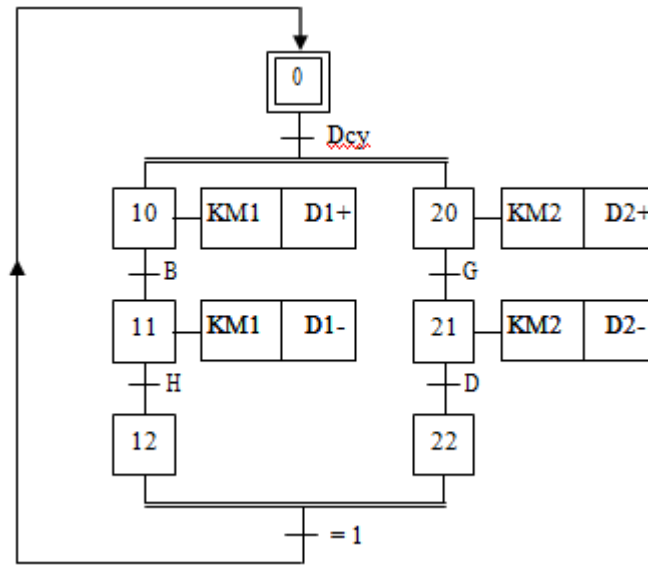


Figure 2.4 grafcet niveau 2.

- Les spécifications : caractérisant la nature des spécifications techniques auxquelles doit satisfaire la Partie Commande. On distingue trois groupes de spécifications :
 - Spécifications fonctionnelles,
 - Spécifications technologiques,
 - Spécifications opérationnelles.
- La finesse : caractérisant le niveau de détail dans la description du fonctionnement, d'un niveau global (ou macro-représentation) jusqu'au niveau de détail complet où toutes les actions et informations élémentaires sont prises en compte.

2.3 Concepts de base du GRAFCET :

2.3.1 Etapes :

Une étape caractérise un comportement invariant d'une partie ou de la totalité de la partie commande. C'est une situation dans laquelle les variables d'entrée et de sortie de la partie commande conservent leur état.

Une étape est symbolisée par un carré ou un rectangle repéré numériquement. A un instant donné et suivant l'évolution du système :

- Une étape est soit active ou inactive,
- L'ensemble des étapes actives définit la situation de la partie commande.

Si à un instant donné, il est nécessaire de préciser les étapes actives, un point (un marqueur) est placé à la partie inférieure du symbole.

Les étapes qui sont à l'état actif au moment de l'initialisation (état d'attente) sont représentées par un double rectangle et le marquage initial est défini par :

$$M0 = [m0(e1), \dots, m0(ei), \dots, m0(eN)]$$

Avec $m0(ei) = 1$ pour les étapes actives au moment de l'initialisation et $m0(ei) = 0$ pour celles qui sont inactives.

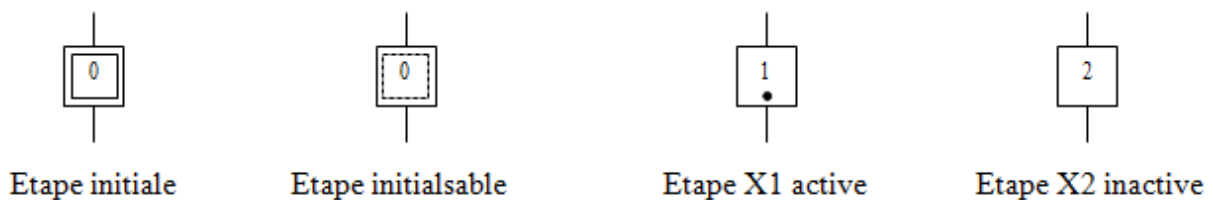


Figure2.5 Situation des étapes.

2.3.2 Actions associées à l'étape :

Une ou plusieurs actions élémentaires ou complexes peuvent être associées à une étape. Elles traduisent ce qui être fait chaque fois que l'étape à laquelle elles sont associées est active.

Les actions, qui sont le résultat du traitement logique des informations par la partie commande, peuvent être :

- Externes et correspondent aux ordres émis vers la partie opérative ou vers les éléments extérieurs
- Internes et concernent des fonctions spécifiques de l'automatisme telles que : temporisation, comptage, ..., etc.

Les actions associées sont décrites de façon littérale ou symbolique à l'intérieur d'un ou plusieurs rectangles reliés au symbole de l'étape à laquelle elles sont associées. Plusieurs rectangles peuvent être reliés à une étape ou plusieurs actions peuvent figurer dans un même cadre.

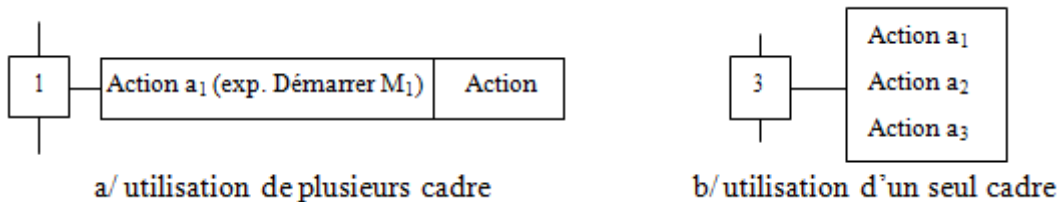


Figure 2.6 Actions associées aux étapes.

2.3.3 Transitions :

Les transitions indiquent les possibilités d'évolution entre étapes. Cette évolution s'accomplit par le franchissement de la transition.

Une transition est soit validée soit non validée. Elle est dite validée lorsque toutes les étapes immédiatement précédentes reliées à cette transition sont activées.

Une transition entre deux étapes se représente par une barre perpendiculaire aux liaisons . Lorsque plusieurs arcs arrivent ou partent d'une transition, leur regroupement est indiqué par une double ligne horizontale.

2.3.4 Réceptivité associée à la transition :

A chaque transition est associée une proposition logique appelée *réceptivité* qui peut être soit vraie soit fausse.

Parmi toutes les informations disponibles à un instant donné, la réceptivité regroupe uniquement celles qui sont nécessaires au franchissement de la transition.

La réceptivité écrite sous forme de proposition logique est une fonction combinatoire :

- D'informations externes provenant de la partie opérative ou du poste opérateur,
- D'informations internes en rapport avec les fonctions spécifiques de l'automatisme (temporisation, comptage, ...) ou les états actifs et inactifs des autres étapes.

Dans la figure (2.1), la fonction logique notée « $tr_6 = c.b + d$ » exprime la réceptivité de la transition entre l'étape 6 et l'étape 7. Elle traduit l'information de fin de l'action a_6 .

Les notations $a \uparrow$ ou $b \downarrow$ sont employées lorsque la condition de réceptivité est liée au Changement d'état de la variable logique. Dans ce cas, ces notations désignent respectivement le front montant de la variable « a » et le front descendant de la variable « b ».

Si la réceptivité est une temporisation, nous devons indiquer, après le repère t (qui signifie l'intervention du facteur temps), l'origine de la temporisation et sa durée. L'origine est l'instant du début de la dernière activation de l'étape antérieure spécifiée. A titre d'exemple, si la réceptivité tr_7 dans la figue désigne une temporisation de 10s, on écrit alors $tr_7 = t/7/10s$ où sept signifie que l'origine du temps est le début d'activation de l'étape 7.

Si la réceptivité n'est pas précisée, alors cela signifie qu'elle est toujours vraie (=1).

2.3.5 Liaisons orientées :

Elles sont de simples traits verticaux qui relient les étapes aux transitions et les transitions aux étapes. Elles sont normalement orientées de haut vers le bas. Une flèche est nécessaire dans le cas contraire.

2.4 Classification des actions associées aux étapes :

L'action associée à l'étape peut être de 3 types : continue, conditionnelle ou mémorisée.

Les actions peuvent être classées en fonction de leur durée par rapport à celle de l'étape.

2.4.1 Actions continues :

L'ordre est émis de façon continue tant que l'étape à laquelle il est associé est active.

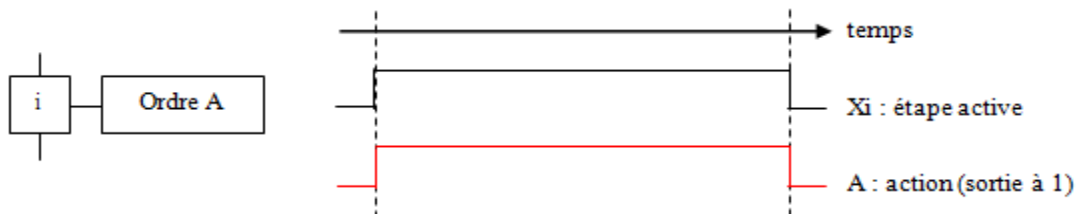


Figure2.7 Chronogramme de l'action continue.

2.4.2 Actions conditionnelles :

Une action conditionnelle n'est exécutée que si l'étape associée est active et si la condition associée est vraie. Elles peuvent être décomposées en 3 cas particuliers :

- Action conditionnelle simple : Type C

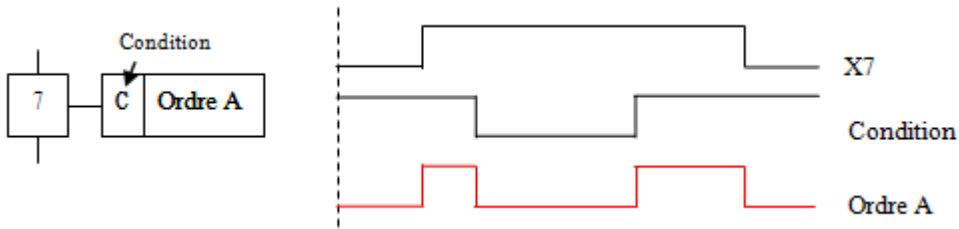


Figure 2.8 Chronogramme de l'action conditionnelle simple.

- Action retardée : Type D (Delay)

Le temps intervient dans cet ordre conditionnel comme condition logique. L'indication du temps s'effectue par la notation générale " $t/xi/D$ " dans laquelle "xi" indique l'étape prise comme origine du temps et "d" est la durée du retard.

Exemple : " $t/x6/5s$ " prendra la valeur logique 1, 5s après la dernière activation de l'étape 6.

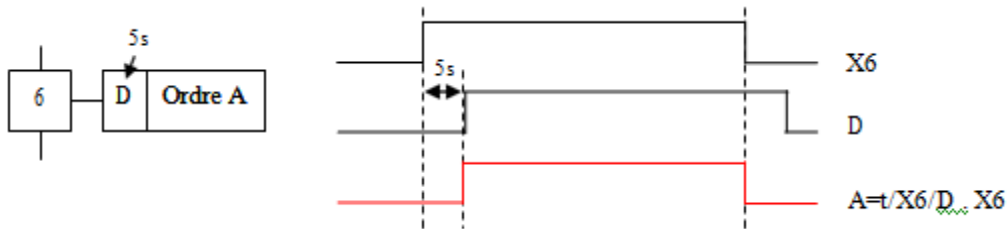


Figure 2.9 Chronogramme de l'action retardée.

- Action de durée limitée : Type L (limited)

L'ordre est émis dès l'activation de l'étape à laquelle il est associé ; mais la durée de cet ordre sera limitée à une valeur spécifiée. L'ordre "A" est limité à 2s après l'activation de l'étape 4.

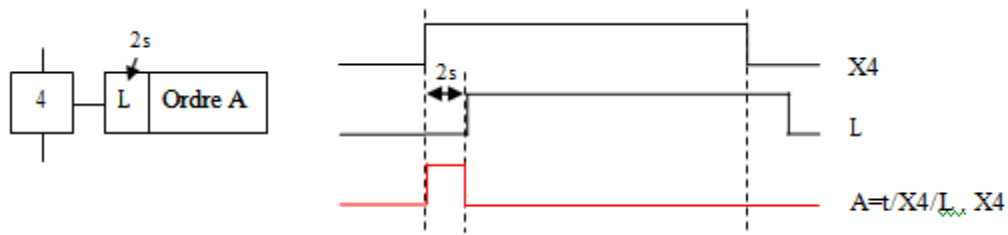


Figure 2.10 Chronogramme de l'action de durée limitée.

2.4.3 Action maintenue sur plusieurs étapes :

Afin de maintenir la continuité d'une action sur plusieurs étapes, il est possible de répéter l'ordre continu relatif à cette action, dans toutes les étapes concernées ou d'utiliser une description sous forme de séquences simultanées (Les séquences simultanées seront traitées ultérieurement).

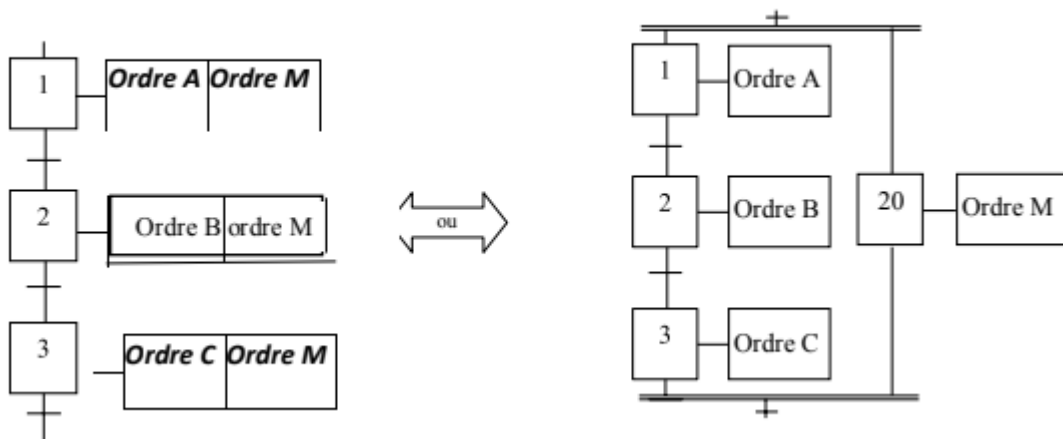


Figure 2.11 Action maintenue sur plusieurs étapes.

2.4.4 Action mémorisée :

Le maintien d'un ordre, sur la durée d'activation de plusieurs étapes consécutives, peut également être obtenu par la mémorisation de l'action, obtenue par l'utilisation d'une fonction auxiliaire appelée fonction mémoire [4] .

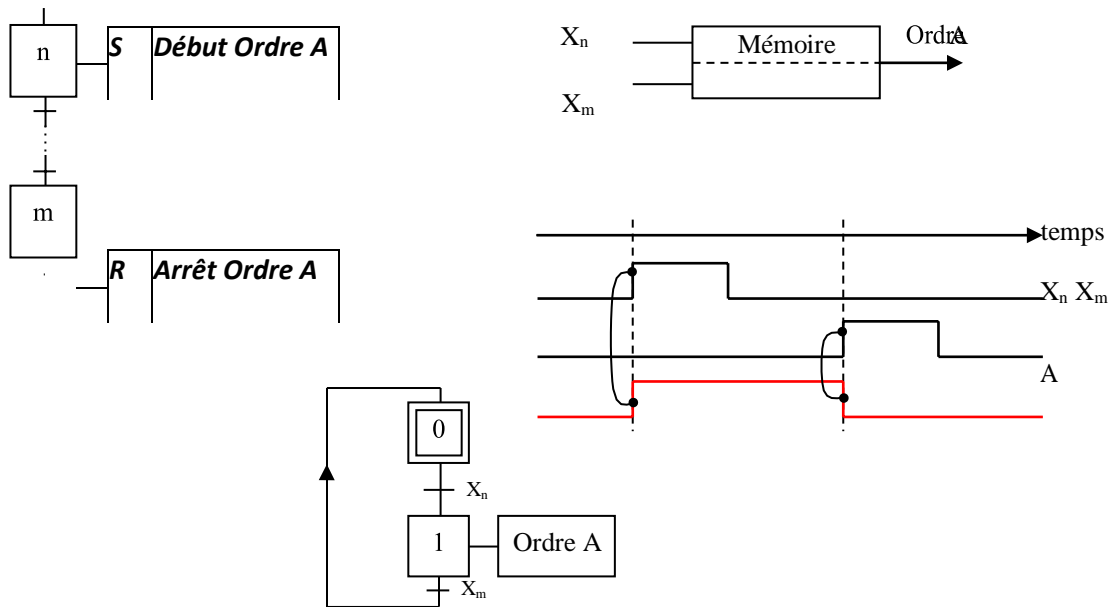


Figure 2.12 Action à effet Maintenu par une Action Mémorisée.

2.5 Règles d'évolution d'un GRAFCET :

L'évolution du réseau s'effectue par le franchissement de transitions à partir du marquage initial, selon les règles suivantes :

- **Condition initiale :** A l'instant initial, seules les étapes initiales sont actives.
- **Franchissement d'une transition :** Pour qu'une transition soit validée, il faut que toutes ses étapes amont (immédiatement précédentes reliées à cette transition) soient actives. Le franchissement d'une transition se produit lorsque la transition est validée, et seulement si

la réceptivité associée est vraie.

-- **Evolution des étapes actives** : Le franchissement d'une transition entraîne obligatoirement l'activation de toutes les étapes immédiatement suivantes et la désactivation de toutes les étapes immédiatement précédentes.

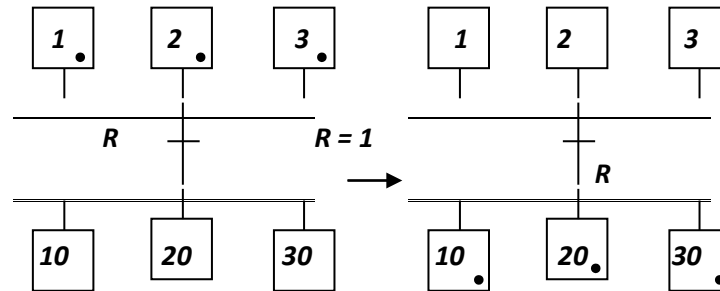


Figure2.13 franchissements d'une transition.

- **Franchissement simultané** : Toutes les transitions simultanément franchissables à un instant donné sont simultanément franchies.
- **Conflit d'activation** : Si une étape doit être simultanément désactivée par le franchissement d'une transition aval, et activée par le franchissement d'une transition amont, alors elle reste active. On évite ainsi des commandes transitoires (néfastes à la partie opérative) [4] .

2.6 Structures de base :

Une séquence, dans un Grafcet, est une suite d'étapes à exécuter l'une après l'autre. Autrement dit chaque étape ne possède qu'une seule transition AVANT et une seule transition AMONT. Les structures de base sont les suivantes :

2.6.1 Séquence unique :

Une séquence unique est composée d'une suite d'étapes pouvant être activées les unes après les autres. Chaque étape n'est suivie que par une seule transition et chaque transition n'est validée que par une seule étape.

2.6.2 Séquences simultanées :

Deux ou plusieurs séquences sont dites simultanées si elles présentent un parallélisme structural où le franchissement d'une transition conduit à leur activation au même temps.

Après l'activation simultanée de ces séquences, les évolutions des étapes actives dans chacune des séquences deviennent indépendantes.

Pour synchroniser la désactivation des séquences simultanées, il faut prévoir des étapes d'attente.

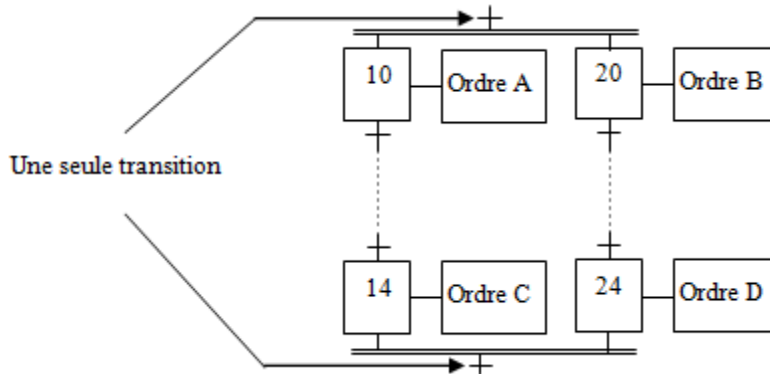


Figure2.14 Exemple de séquences simultanées.

2.6.3 Sélection de séquences :

Une sélection ou un choix d'évolution entre plusieurs séquences (ou étapes) se représente, à partir d'une ou plusieurs étapes, par autant de transitions validées qu'il y a d'évolutions possibles.

2.6.4 Séquences exclusives :

Deux séquences sont dites exclusives si les réceptivités associées aux transitions qui permettent la sélection de l'une ou l'autre séquence sont des expressions exclusives.

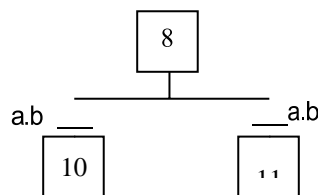


Figure2.15 Exemple de réceptivités exclusives.

2.6.5 Saut d'étapes et reprise de séquences :

Le saut d'étapes permet de sauter une ou plusieurs étapes par le franchissement d'une transition dont la réceptivité indiquant que les actions associées à ces étapes ne sont plus utiles ou sans effet, est vraie. C'est le cas de tr_4 dans la figure (2.12).

La reprise de séquence permet au contraire de recommencer plusieurs fois la même séquence tant que la condition fixée n'est pas obtenue. C'est le cas de tr_3 dans la figure (2.12).

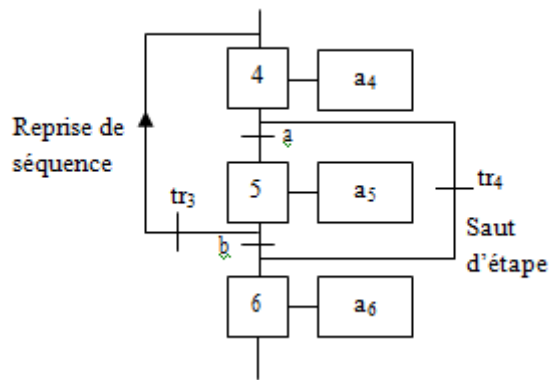


Figure 2.16 Saut et reprise de séquences.

2.6.6 Parallélisme interprété :

Dans ce type de structure, les réceptivités ne sont pas exclusives et des évolutions simultanées sont possibles (voir figure (2.13)).

Ce parallélisme est dit interprété pour le différencier du parallélisme structural des séquences simultanées.

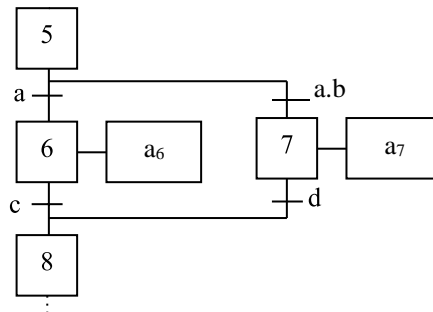


Figure 2.17 Exemple de parallélisme interprété.

2.6.7 Réutilisation d'une même séquence :

Une même séquence, répétée plusieurs fois, peut être traitée comme un sous-programme. Une telle situation peut être représentée par la figure (2.14) qui signifie : à l'étape k est associé le sous-programme $S_{k_1-k_2}$ qui comprend les étapes k_1 à k_2 . Ce sous-programme doit faire l'objet d'un diagramme séparé.

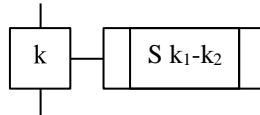


Figure 2.18 Représentation d'une séquence réutilisable.

2.6.8 Couplage entre séquences :

Le principe, représenté dans la figure (2.15), est le suivant :

A la fin d'une séquence i, l'étape k mémorise l'autorisation donnée à la séquence i+1 qui pourra commencer à s'exécuter lorsque les conditions propres à cette séquence seront réalisées.

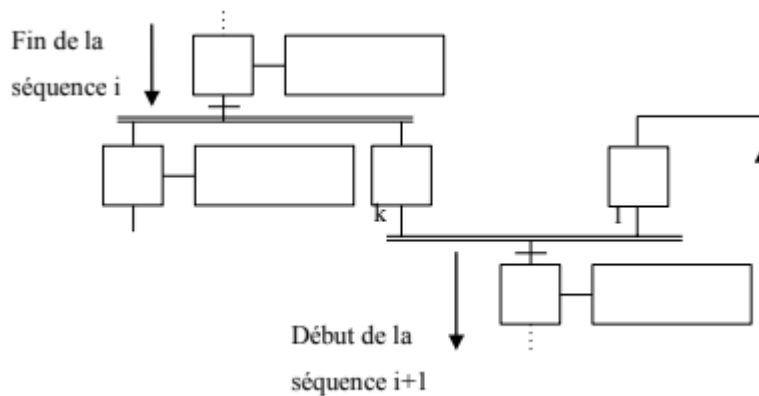


Figure 2.19 Exemple de couplage de séquences.

2.6.9 Partage de ressources ou de séquences :

Une ressource commune, physique ou logique, peut être partagée entre plusieurs séquences utilisatrices exclusives.

L'écriture des réceptivités de la première transition de chaque séquence doit traduire la priorité logique.

Dans l'exemple de la figure (2.16), la ressource commune représentée par l'étape k permet la validation des transitions tr_i , tr_j ou tr_l et la séquence correspondante S_i , S_j ou S_l sera parcourue.

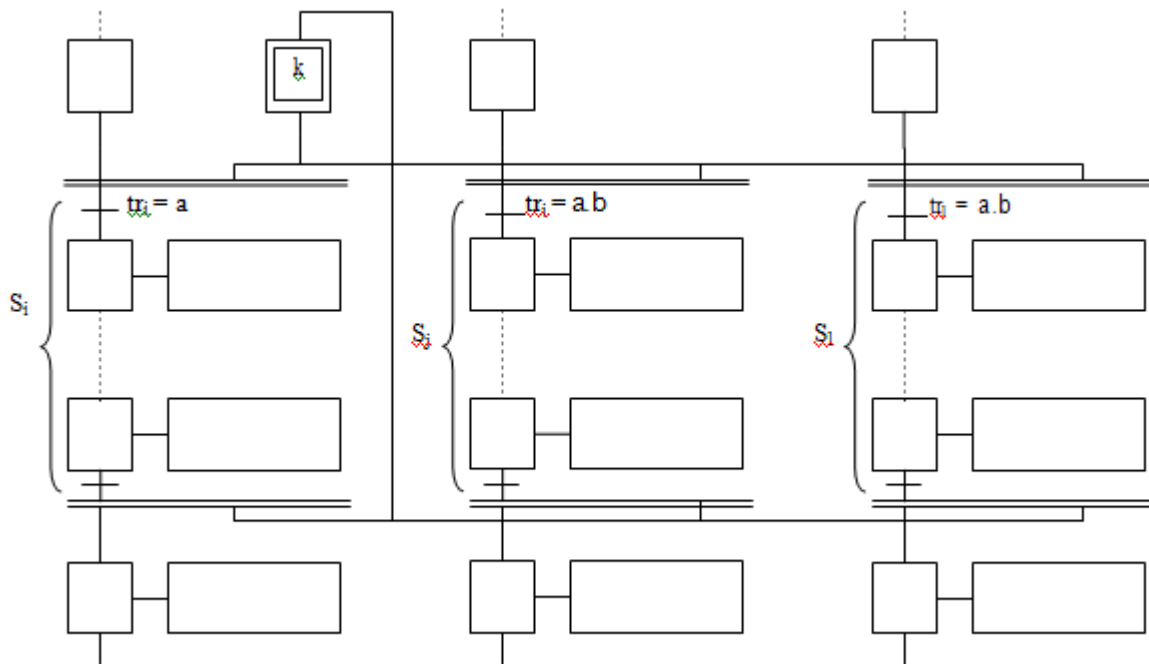


Figure 2.20 Exemple de partage de ressources.

2.7 Analyse d'un GRAFCET :

2.7.1 Méthode de réduction :

La réduction peut être effectuée par la mise en œuvre de trois opérations de simplification élémentaires :

A La suppression des redondantes :

La suppression des redondantes, c'est à dire celles qui ont toujours les mêmes conditions d'activation et de désactivations, quelque soit l'évolution du GRAFCET, consiste à les supprimer et ne laisser qu'une seule et à reporter les actions des étapes éliminées sur l'étape restante. Dans la figure (2.17), les étapes 1 et 4 sont des étapes redondantes car elles sont activées et désactivées dans les mêmes conditions, on peut donc supprimer une d'entre elles.

B La fusion d'étapes :

La fusion d'étapes qui consiste à remplacer plusieurs étapes par une seule en remplaçant les actions simples par des actions conditionnées avec les réceptivités des transitions supprimées. L'exemple ci-dessous, les étapes 2 et 3 peuvent être fusionnées en prenant soins de subordonner l'action a_3 à l'existence de la condition b_2 et en conditionnant l'action a_2 avec

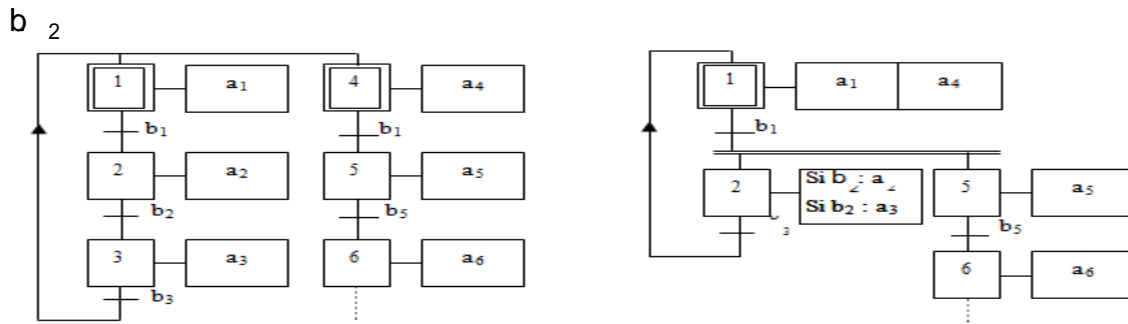


Figure2.21 Exemple de suppression et de fusion d'étapes dans un GRAFCET.

C La suppression des transitions redondantes :

La suppression des transitions redondantes, c'est à dire celles qui ont les mêmes étapes d'entrée et les mêmes étapes de sortie.

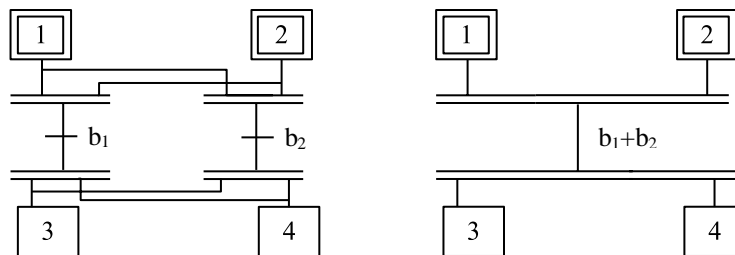


Figure2.22 Exemple de suppression de transition dans un GRAFCET.

D Graphe des états accessibles :

Le graphe des situations accessibles décrit les différentes situations du GRAFCET qui peuvent être atteintes à partir de la situation initiale en prenant en considération les événements externes (voir la figure (2.19)).

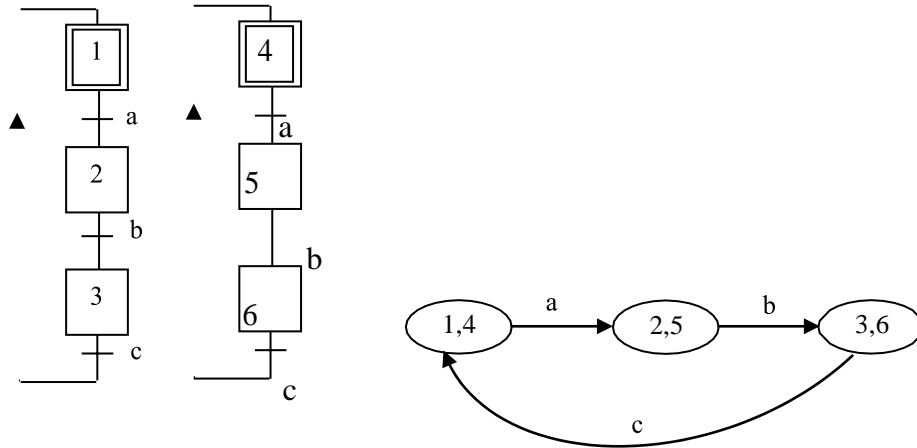


Figure 2.23 Exemple de GRAFCET et son graphe des situations accessibles.

E Liaison entre GRAFCET :

Une étape dans un GRAFCET peut servir comme réceptivité à une autre étape d'un autre GRAFCET. Cette méthode est utilisée aussi pour synchroniser deux grafquets c'est-à-dire rendre l'évolution de l'un dépendent de l'évolution de l'autre [4].

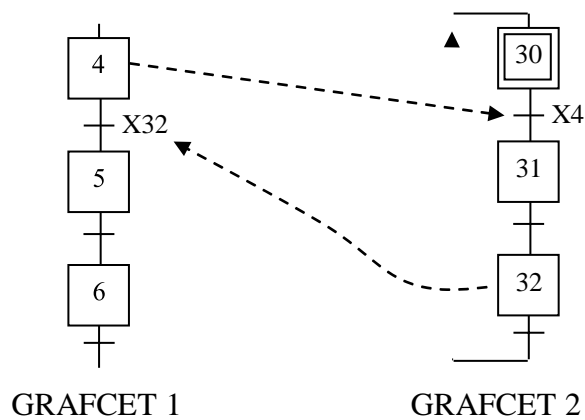


Figure 2.24 Liaison entre grafquets.

2.8 Propriétés des GRAFCET :

- Un GRAFCET est propre s'il est réinitialisable.
- Un GRAFCET est vivant si à partir de toute situation accessible, il existe des séquences de franchissement qui permettent de franchir toutes les transitions, (donc il n'admet pas de situation de blocage).
- Un GRAFCET est pseudo-vivant s'il est caractérisé par le fait qu'il existe toujours au moins une transition de sortie franchissable pour toute situation accessible.
- Un GRAFCET est sauf si aucune étape n'est réactivée immédiatement après sa désactivation.

2.9 GRAFCET d'automatisation d'injection :

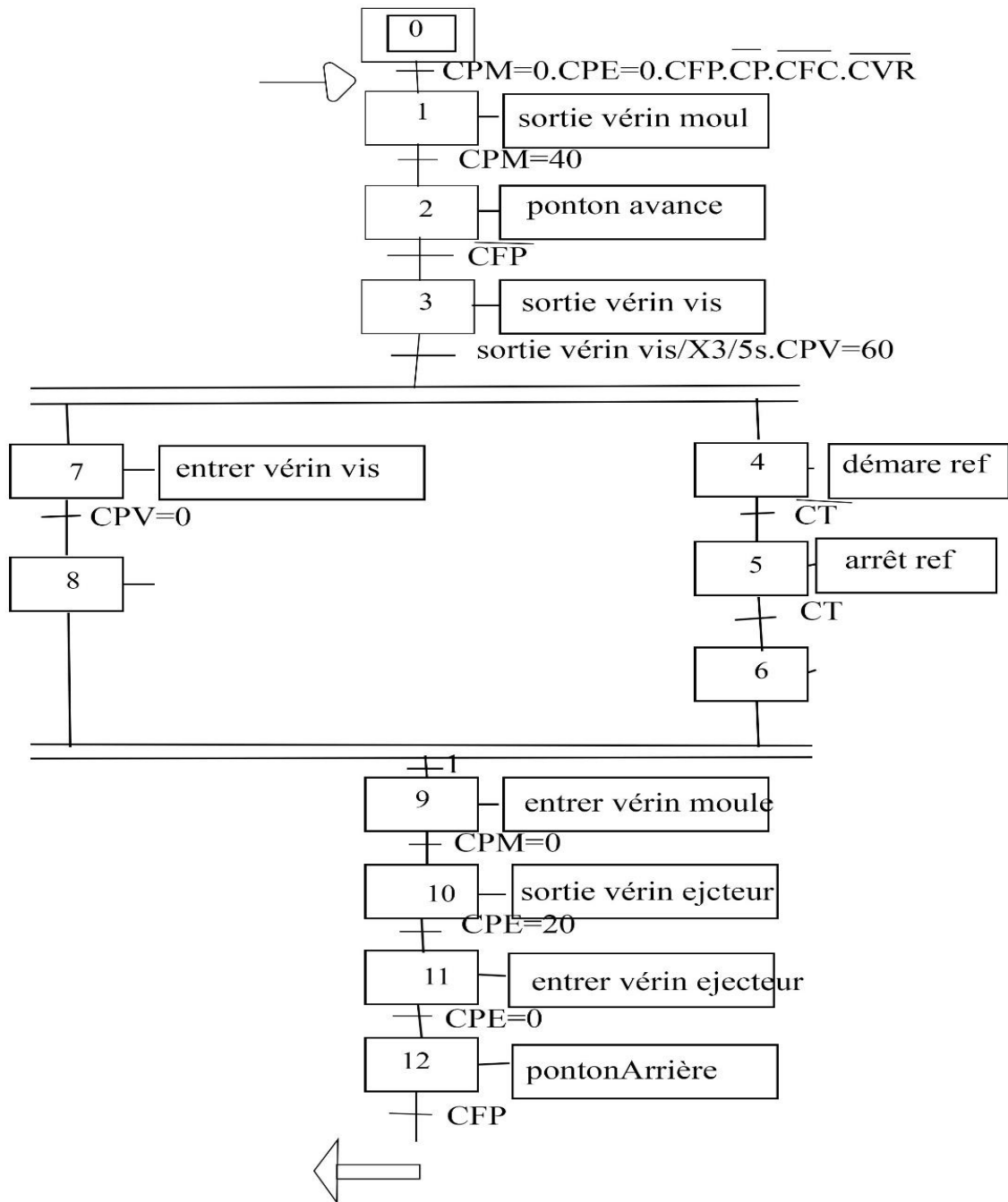


Figure 2.25 GRAFCET d'automatisation d'injection.

2.10 tableaux entrées sorties de grafcet :

Entrées	Sorties
CPM : capteur position moule	Vérin moule
CPV : capteur position vis	Vérin vis
CPE : capteur position éjecteur	Vérin éjecteur
CFP : capteur fin de cours ponton	
CP : capteur fin de cours Porte	
CFC : capteur fin de cours capots	
CT : capteur température	
CVR : capteur vérin de rotation	Vérin de rotation vis

2.11 Conclusion :

Le Grafcet est un diagramme fonctionnel dont le but est de décrire graphiquement, suivant un cahier des charges, les différents comportements de l'évolution d'un automatisme séquentiel. Il est à la fois simple à utiliser et rigoureux sur le plan formel et constitue un unique outil de dialogue entre toutes les personnes collaborant à la conception, à l'utilisation ou à la maintenance de la machine à automatiser.

Chapitre 3 **Automate Programmable industriel**

3.1 Introduction :

L'automate programmable industriel API(ou Programmable Logic Controller PLC) est l'un des appareils de commande des systèmes de production et d'automatisme les plus utilisés dans l'industrie. Son apparition pour la première fois fut aux Etats-Unis dans le secteur de l'industrie automobile. Dans ce chapitre, nous donnons un bref aperçu sur la structure des systèmes automatisés ainsi que les automates. La deuxième partie est consacrée aux différents aspects de leur programmation. Le cas de la machine injection plastique, qui fait l'objet de notre étude, est abordé dans la dernière partie du chapitre.

3.2 Systèmes automatisés :

L'automatisation d'un système consiste à transformer l'ensemble des tâches de commande et de surveillance, réalisées par des opérateurs humains, dans un ensemble d'objets techniques appelés partie commande. Cette dernière mémorise le savoir-faire des opérateurs, pour obtenir l'ensemble des actions à effectuer sur la matière d'œuvre, afin d'élaborer le produit final . Parmi les objectifs de l'automatisation on peut citer : réalisation des tâches répétitives, sécurité, économie des matières premières et l'énergie, augmentation de la productivité et plus d'adaptation à des contextes particuliers (flexibilité).

3.2.1 Structure des systèmes automatisés :

Un système automatisé est toujours composé d'une partie commande (PC) et une partie opérative (PO). Pour faire fonctionner ce système, l'opérateur va donner des consignes à la PC. Celle-ci va traduire ces consignes en ordres qui vont être exécutés par la PO. Une fois les ordres accomplis, la PO va le signaler à la PC, par un retour d'information, qui va à son tour le signaler à l'opérateur, ce dernier pourra donc dire que le travail a bien été réalisé.

A Partie commande :

C'est la partie qui gère le fonctionnement du système automatisé. Elle est, en général, composée d'un ordinateur qui contient dans sa mémoire un programme. Elle transmet les ordres aux actionneurs de la partie opérative à partir :

Du programme qu'elle contient ;

Des informations reçues par les capteurs ;

Des consignes données par l'utilisateur ou par l'opérateur.

B Partie opérative :

La partie opérative d'un système automatisé peut être décrite en une ou plusieurs chaînes fonctionnelles, ces dernières comportent une chaîne d'action et une chaîne d'acquisition. La chaîne d'action est un ensemble organisé de composants dont le rôle est de convertir un ordre émis par la partie commande, en effet, sur la matière d'oeuvre. Elle est constituée de :

Pré-actionneurs (électrovannes, distributeurs, etc.) ; Actionneurs (vérins, moteurs, etc.).

C Interface :

Elle relie la partie opérative (PO) et la partie commande (PC). C'est un système de traduction d'informations entre la PC et la PO.

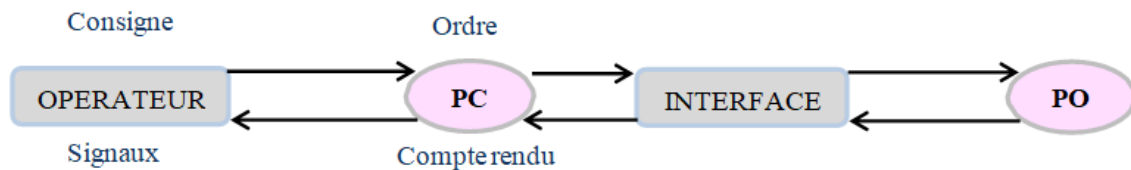


Figure 3.1 : structure d'un système automatisé.

3.3 Généralités sur les API :

3.3.1 Définition d'un automate programmable industriel (API) :

Un automate programmable est un système électronique fonctionnant de manière numérique, destiné à être utilisé dans un environnement industriel. Il utilise une mémoire programme pour le stockage interne des instructions utilisées aux fins de la mise en œuvre des fonctions spécifiques, telles que : des fonctions logique, de mise en séquence, de temporisation, de comptage et de calcul arithmétique, pour commander, au moyen des entrées/sorties (de type tout/rien ou analogiques), de 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 utilisé dans toutes leurs fonctions prévues.

Un API a trois caractéristiques fondamentales :

Il peut être directement connecté aux capteurs et pré-actionneurs grâce à ses E/S industrielles

Il est conçu pour fonctionner dans des ambiances industrielles sévères

Enfin, sa programmation à partir de langages spécialement développés pour le traitement de fonctions d'automatisme facilite son exploitation et sa mise en œuvre [5].

3.3.2 Architecture des automates :

L'architecture du processeur d'un automate programmable est fondamentalement la même que celle d'un ordinateur à usage général. Néanmoins, il existe certaines caractéristiques importantes qui les distinguent. Tout d'abord, contrairement aux ordinateurs, les automates programmables sont conçus pour résister aux conditions difficiles de l'environnement industriel. Un automate bien conçu peut être placé dans une zone avec d'importantes quantités: de bruit électrique, des interférences électromagnétiques, des vibrations et d'humidité sans condensation. Une deuxième distinction des automates est que leurs matériels et logiciels sont conçus pour une utilisation facile par les électriciens et les techniciens. Les interfaces matérielles pour la connexion d'appareils de terrain font en réalité partie de l'automate lui-même et se connectent facilement. Les circuits d'interface modulaires et autodiagnostic sont en mesure d'identifier les dysfonctionnements et, d'ailleurs, ils sont facilement enlevés et remplacés. En outre, la programmation du logiciel utilise des symboles traditionnels relais d'échelle, ou d'autres langues apprises facilement, qui sont familières au personnel de l'usine. Alors que les ordinateurs sont des machines informatiques complexes capables d'exécuter plusieurs programmes ou tâches simultanément et dans n'importe quel ordre. La norme PLC exécute un programme unique dans un mode séquentiel ordonné de la première à la dernière instruction [5].

3.3.3 Aspect extérieur des API :

Les automates peuvent être de type compact ou modulaire.

A De type compact :

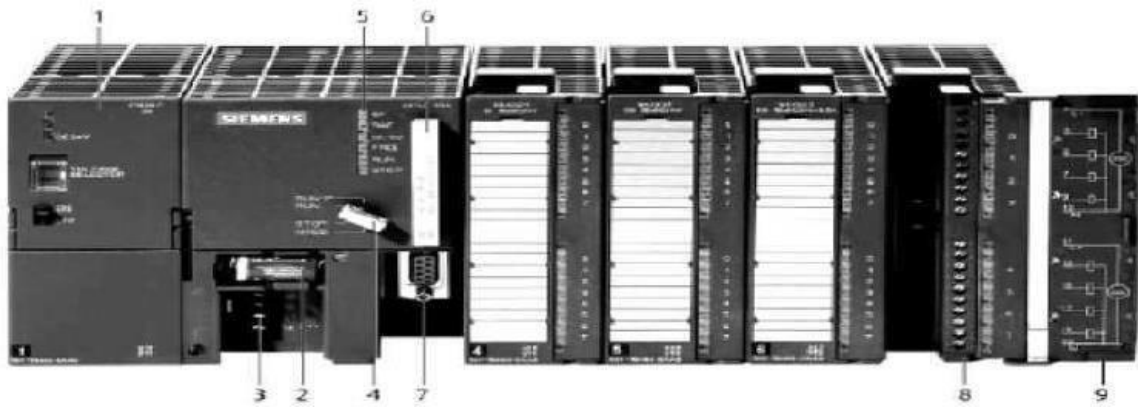
On distinguera les modules de programmation (LOGO de Siemens, ZELIO de Schneider, MILLENIUM de Crouz,...) des micros automates. Il intègre le processeur, l'alimentation, les entrées et les sorties. Selon les modèles et les fabricants, il pourra réaliser certaines fonctions supplémentaires (comptage rapide, ajout d'entrées/ sorties analogiques ...) et recevoir des extensions en nombre limité. Ces automates, de fonctionnement simple, sont généralement destinés à la commande de petits automatismes. [5]

B De type modulaire :

Le processeur, l'alimentation et les interfaces d'entrées / sorties résident dans des unités séparées (modules) et sont fixées sur un ou plusieurs racks contenant le "Fond de panier" (bus plus connecteurs). Ces automates sont intégrés dans les automatismes complexes où puissants, où la capacité de traitement et flexibilité sont nécessaires [5].



Figure 3.2 Automate compact (Allen-Bradley) Figure 3.3 Automate Modulaire (Modicon).



- | | |
|--|-------------------------------|
| 1. Module d'alimentation | 6. Carte mémoire |
| 2. Pile de sauvegarde | 7. Interface multipoint (MPI) |
| 3. Connexion au 24V cc | 8. Connecteur frontal |
| 4. Commutateur de mode (à clé) | 9. Volet en face avant |
| 5. LED de signalisation d'état et de défauts | |

Figure 3.4 Automate modulaire (Siemens).

3.3.4 Structure interne d'un automate :

Les API comportent quatre parties principales : une mémoire, un processeur, des interfaces d'entrées/sorties et d'une alimentation ($240V_{ac}$, $24V_{cc}$).

Ces quatre parties sont reliées entre elles par des bus (ensemble de câbles autorisant le passage de l'information entre ces 4 secteurs de l'API)

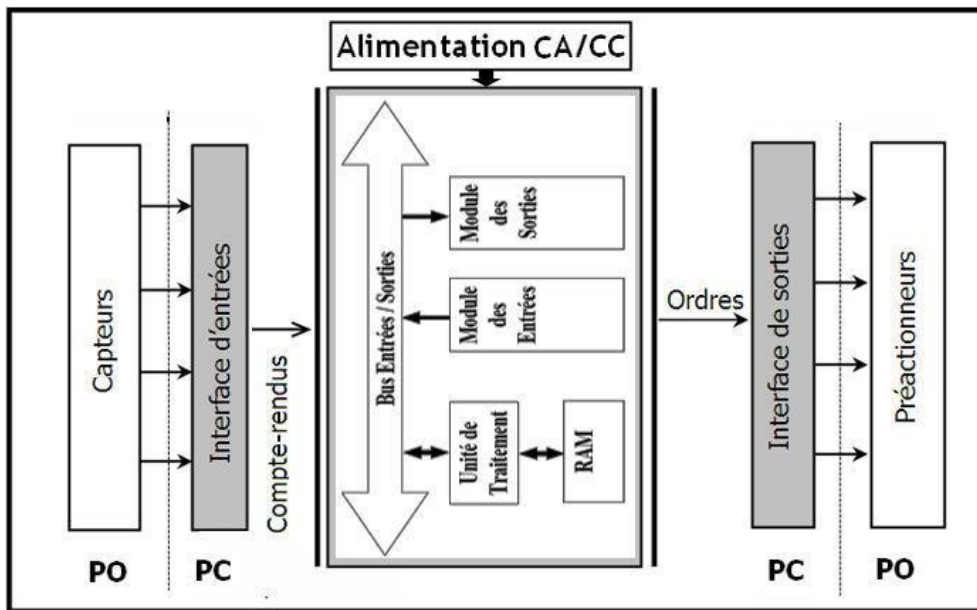


Figure 3.5 : Structure interne d'un API.

3.3.5 Description des éléments d'un API :

A L'alimentation :

L'alimentation du système joue un rôle majeur dans le fonctionnement total du système. En fait, il peut être considéré comme le "gestionnaire de premier niveau" de la fiabilité et l'intégrité du système. Sa fonction n'est pas seulement de fournir des tensions continues internes pour les composants du système, mais aussi pour surveiller et réguler les tensions fournies et prévenir la CPU si quelque chose ne va pas. Le bloc d'alimentation a pour fonction de fournir une puissance bien régulée et de protection pour les autres composants du système.

B Unité centrale ou CPU :

L'unité centrale commande l'interprétation et l'exécution des instructions programmées. Elle est aussi chargée de détecter les pannes de communication, ainsi que d'autres défaillances qui peuvent survenir pendant le fonctionnement du système. Il doit alerter l'opérateur ou le système en cas de dysfonctionnement. A base de microprocesseur, l'unité centrale réalise

toutes les fonctions logiques, arithmétiques et de traitement numérique (transfert, comptage, temporisation ...). Ce module se compose essentiellement de :

B.1 Microprocesseur : Il constitue le cœur de la CPU. Son rôle consiste, d'une part, à organiser les différentes relations entre la zone mémoire et les interfaces d'E/S et, d'autre part, à gérer les instructions du programme.

B.2 Bus : C'est un ensemble de pistes conductrices (pistes en cuivre) par lesquelles s'achemine une information binaire (suite de 0 ou 1), c'est-à-dire ensemble de fils autorisant le passage des informations entre les quatre secteurs (l'alimentation, la mémoire, le processeur et l'interface E/S) de l'automate. L'unité centrale dispose de trois bus : bus de données, bus d'adresses et bus de commandes.

B.3 Mémoire : Elle est conçue pour recevoir, gérer et stocker des informations issues des différents secteurs du système qui sont :

Le terminal de programmation.

Le processeur, qui lui gère et exécute le programme.

Elle reçoit également des informations en provenance des capteurs.

B.4 Interfaces d'entrées / sorties : Les interfaces d'entrées / sorties permettent à l'unité centrale de communiquer avec l'environnement ou les périphériques.

C Modules entrées/sorties :

Les modules d'E/S assurent le rôle d'interface entre le procédé à commander et la CPU. On distingue deux types :

C.1 Module entrées/sorties TOR (Tout ou Rien): La gestion de ce type de variables, constituant le point de départ des API, reste l'une de leurs activités majeures. Leur nombre est en générale de 8, 16, 24 ou 32 entrées/sorties, qui peuvent fonctionner:

-en continue 24V, 48V.

- en alternative 24V, 48V, 100V/120V, 200V/240V.

C.2 Module entrées/sorties analogique: Elles permettent l'acquisition de mesures (entrées analogiques) et générer des signaux de commande (sorties analogiques). Ces modules

comportent un ou plusieurs convertisseurs analogique/numériques (A/N) pour les entrées et numériques/analogiques (N/A) pour les sorties dont la résolution est de 8 à 16 bits.

Les standards les plus utilisés sont : $\pm 10V$, $0-10V$, $\pm 20mA$, $0-20mA$ et $4-20mA$. Ces modules sont, en général, multiplexés en entrée pour utiliser un seul convertisseur A/N, alors que les sorties exigent un convertisseur N/A par voie pour pouvoir garder la commande durant le cycle de l'API.

3.3.6 Cycle d'un API :

Tous les automates fonctionnent selon le même mode opératoire :

- A **Traitement interne** : l'automate effectue des opérations de contrôle et met à jour certains paramètres systèmes (détection des passages en RUN/STOP.) ;
- B **Lecture des entrées** : l'automate lit les entrées (de façon synchrone) et les recopie dans la mémoire image des entrées
- C **Exécution du programme** : l'automate exécute le programme instruction par instruction et écrit les sorties dans la mémoire image des sorties
- D **Ecriture des sorties** : l'automate bascule les différentes sorties (de façon synchrone) aux positions définies dans la mémoire image des sorties.

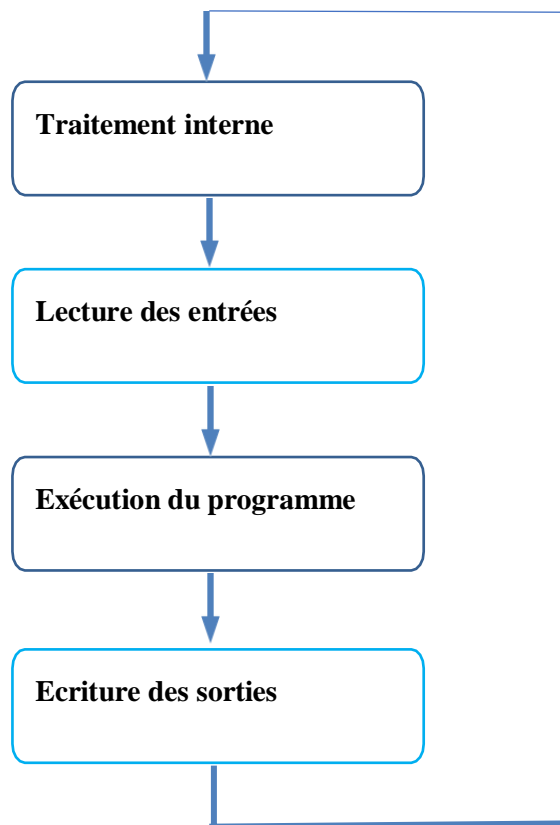


Figure 3.6 Cycle d'un API.

3.3.7 Critères de choix d'un API :

Le choix d'un automate programmable est, en premier lieu, le choix d'une société ou d'un groupe où les contacts commerciaux et expériences vécues sont déjà un point de départ. Les grandes sociétés privilégieront deux fabricants pour faire jouer la concurrence et pouvoir "se retourner" en cas de "perte de vitesse" de l'une d'entre elles.

Le personnel de maintenance doit toutefois être formé sur ces matériels et une très grande diversité des matériels peut avoir de graves répercussions. Un automate utilisant des langages de programmation de type GRAFCET est également préférable pour assurer les mises au point et dépannages dans les meilleures conditions.

La possession d'un logiciel de programmation est aussi source d'économies (achat du logiciel et formation du personnel). Des outils permettant une simulation des programmes sont également souhaitables.

Il faut ensuite quantifier les besoins :

Nombre d'entrées/sorties : le nombre de cartes peut avoir une incidence sur le nombre de racks dès que le nombre d'entrées / sorties nécessaires devient élevé .

Type de processeur : la taille mémoire, la vitesse de traitement et les fonctions spéciales offertes par le processeur permettront le choix dans la gamme souvent très étendue Fonctions ou modules spéciaux .

Fonctions de communication : l'automate doit pouvoir communiquer avec les autres systèmes de commande (API, supervision, ...) et offrir des possibilités de communication avec des standards normalisés (Profibus).

Choix de l'automate :

Après élaboration du nombre d'entrées et de sorties et l'expérience acquis dans les classes antérieures sur ce type d'automate .notre choix porte sur l'automate programmable industriel de gestion de procédé de la série S7-1200 du fabricant siemens.

Le SIMATIC S7-1200 est un système de commande modulaire pour des applications haut de gamme. Il dispose d'une gamme de modules complète pour une adaptation optimale aux tâches les plus diverses et se caractérise par la facilité de réalisation d'architectures décentralisées et la simplicité d'emploi.

3.3.8 Présentation de L'API SIEMENS S7-1200 :

Le contrôleur S7-1200 offre la souplesse et la puissance nécessaires pour commander une large gamme d'appareils afin de répondre aux besoins en matière d'automatisation. Sa forme compacte, sa configuration souple et son important jeu d'instructions forment une solution idéale pour commander une variété importante d'applications [6].

Le CPU combine un microprocesseur, une alimentation intégrée, des circuits d'entrée et de sortie, un PROFINET intégré, des E/S, ainsi que des entrées analogiques intégrées dans un boîtier compact en vue de créer un contrôleur puissant. Une fois le programme chargé, le CPU contient la donnée logique nécessaire au contrôle et à la commande des appareils concernant le processus à contrôler. Le CPU surveille les entrées et modifie-les sorties conformément à la logique du programme intégré, qui peut contenir des instructions booléennes, des instructions de comptage, des instructions de temporisation, des instructions mathématiques complexes ainsi que des commandes pour communiquer avec d'autres appareils intelligents [6].

Le CPU fournit un port PROFINET permettant de communiquer par le biais d'un réseau PROFINET. Des modules supplémentaires sont disponibles pour communiquer via les réseaux PROFIBUS, RS485 ou RS232 [6] .

PROFINET est le nouveau standard de communication créé par PROFIBUS International pour mettre en œuvre des solutions d'automatisation intégrées et cohérentes, sur Ethernet industriel [6].

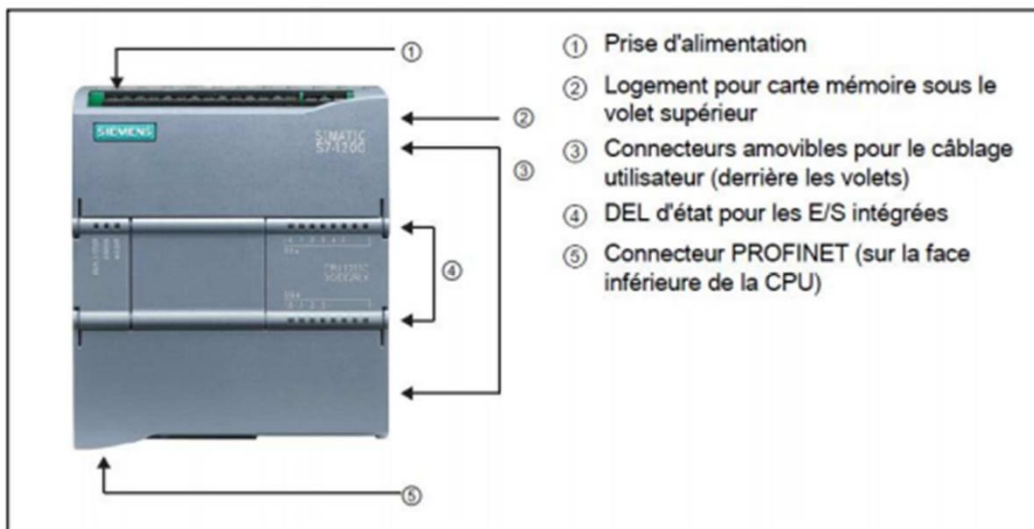


Figure 3.7 Automate S7-1200.

A Modules d'extensions :

La gamme S7-1200 offre divers modules de cartes enfichables pour accroître les capacités de la CPU avec des E/S supplémentaires ou d'autres protocoles de communication Il faudrait se référer aux caractéristiques techniques pour des informations détaillées sur un module spécifique [6] :

- ① : Module communication
- ② : CPU
- ③ : Module d'entrées-sorties
- ④ : Signal Board (SB), Communication Board



Figure 3.8 Module extensions

B Pour un API S7-1200 on distingue [6] :

- Nombre de modules d'extensions E/S : 8
- Nombre Maximal des E/S : 16384
- Nombre des modules de communication : 3
- Module de signaux (SB) : 1



Figure 3.9 Siemens CPU 1214C-6ES7 214-1BG40-0XB0

Dans notre projet, on va utiliser le SIEMENS CPU 1214C 6ES7 214-1BG40-0XB0 dont certaines caractéristiques sont :

Modèle	Module CPU
Type	AC/DC/relais
Entrée numérique	14
Entrée analogique	02
Sortie numérique	10
Sortie analogique	0
Mémoire	100 kb
Tension d'entrée	24 V DC
Permissible range	20.4 28.8 VDC
Courant d'entrée	1 mA
Courant de sortie	0.5 A
Dimensions (H x D)	130 x 100 x 75 mm
Température ambiante min.	-40 °C
Température ambiante max.	70 °C
Protection	IP20
Communication	PROFINET
Software	TIA Portal V15

Tableaux de caractéristique automate siemens s7-1200 [6].

3.4 Présentation du logiciel :

3.4.1 Logiciel TIA portal :

La plateforme de développement TIA Portal (Totally Integrated Automation Portal) de Siemens permet de faire un gain important en temps lors du développement de systèmes d'automatisation.

C'est une plateforme tout en un comportant le logiciel Step 7 pour la programmation d'automates et WinCC Flexible pour les interfaces homme-machine. Cette plateforme est très architecturée proposant les sections HMI pour les interfaces, réseaux et Motion pour la commande de moteurs et variateurs. Grâce à PLCSim, on peut simuler de manière intuitive notre projet avant de le déployer sur un contrôleur.

Et par conséquent le logiciel de programmation de notre automate S7-1200 nous offre toutes les fonctionnalités nécessaires pour configurer, paramétrer et programmer notre S7-1200.

A Caractéristiques :

Le logiciel TIA portal permet d'avoir :

- La programmation d'un automate,
- Plusieurs langages de programmation,
- Une documentation facile pour l'utilisation.

B Type de bloc:

TIA portal offre pour la programmation structurée des blocs utilisateur suivants:

B.1 Bloc d'organisation (OB):

Les blocs d'organisations constituent l'interface entre le système d'exploitation et le programme utilisateur. Ils sont appelés par le système d'exploitation pour le démarrage ou pour le traitement cyclique du programme.

B.2 Bloc fonctionnel (FB):

Le FB est un bloc avec mémoire. Un bloc de données d'instance est associé à chaque appel de FB. Les paramètres transmis au FB, ainsi que les variables statiques sont sauvegardés dans le bloc de données d'instance.

B.3 Fonction (FC):

Une fonction est un bloc de code sans mémoire. Les variables temporaires d'une fonction sont sauvegardées dans la pile des données locales. Ces données sont perdues à l'achèvement de la fonction.

B.4 Bloc de données (DB):

Les blocs de données sont des blocs utilisés par les blocs de code de votre programme utilisateur pour enregistrer des valeurs. Ils y a deux catégories de bloc de données:

Les DB globaux: où tous les OB, FB et FC peuvent lire des données enregistrées et écrire eux-mêmes des données dans le BD.

Les instances DB: sont attribuées à un FB défini.

C Langages de programmation:

C.1 Langage LIST:

Le langage LIST (liste d'instruction) est un langage de programmation textuel. La syntaxe des instructions est très proche du langage machine.

C.2 Langage CONT :

CONT est l'abréviation de schéma à contact. C'est un langage de programmation graphique intégré dans TIA portal. La syntaxe des instructions rassemble à un schéma des circuits qui contient des contacts et des bobines reliés entre eux pour former un réseau.

C.3 Langage LOG:

LOG est l'abréviation de logigramme. C'est un langage graphique, utilisant les symboles de l'électronique numérique (portes logiques).

C.4 Langage grafset:

C'est un langage graphique utilisé pour décrire les opérations séquentielles. Le procédé est représenté comme une suite connue d'étapes reliées entre elles par des transitions, une condition booléenne est attachée à chaque transition

3.5 Création d'un projet :

Pour créer un projet dans la vue du portail, il faut sélectionner l'action « Créer un projet ». Nous pouvons donner un nom à ce nouveau projet, choisir un chemin où il sera enregistré, indiquer un commentaire ou encore définir l'auteur du projet.

Une fois que ces informations sont entrées, il suffit de cliquer sur le bouton « créer »

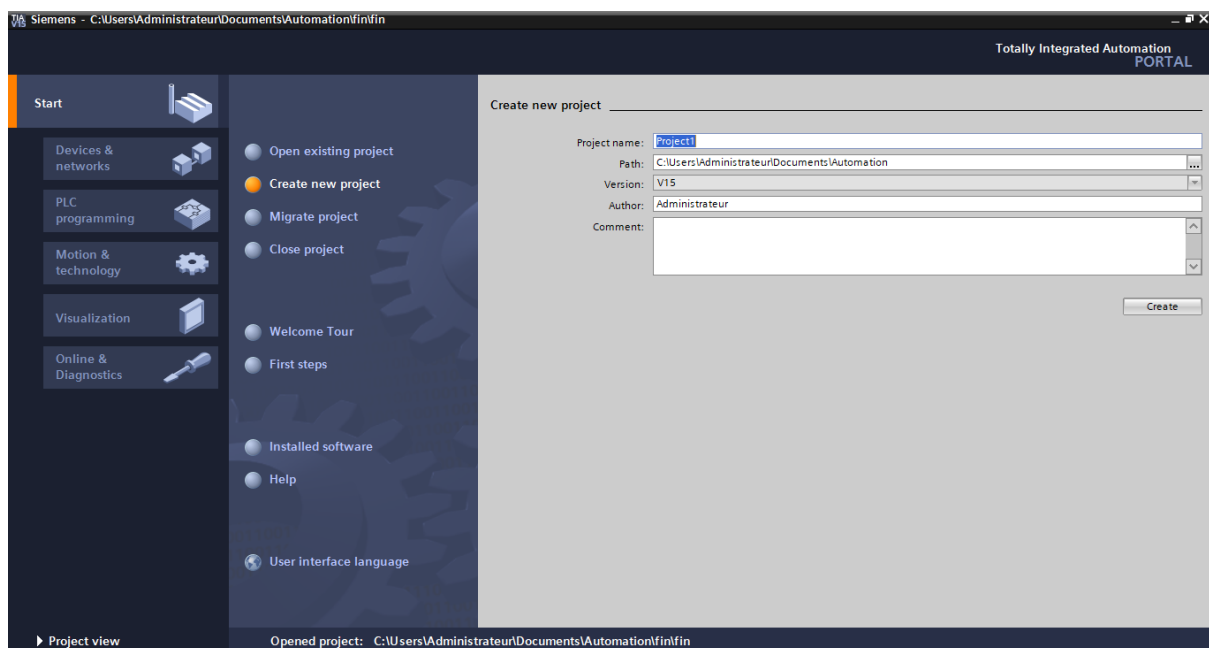


Figure 3.10 Création d'un projet.

3.6 Configuration et paramétrage du matériel :

Une fois le projet créé, la station de travail peut être configurée.

- La première étape consiste à définir le matériel existant. Pour cela, il faut passer par la vue du projet et cliquer sur « ajouter un appareil » dans le navigateur du projet.
- La liste des éléments qui peuvent être ajoutés sont (API, HMI, système PC). Tout d'abord il faut choisir le CPU pour ensuite venir ajouter les modules complémentaires (alimentation, E/S TOR ou analogiques, module de communication AS-i...).
- Les modules complémentaires de l'API peuvent être ajoutés en utilisant le catalogue.

L'ajout d'un écran ou d'un autre API se fait par la commande « ajouter un appareil » dans le navigateur du projet.

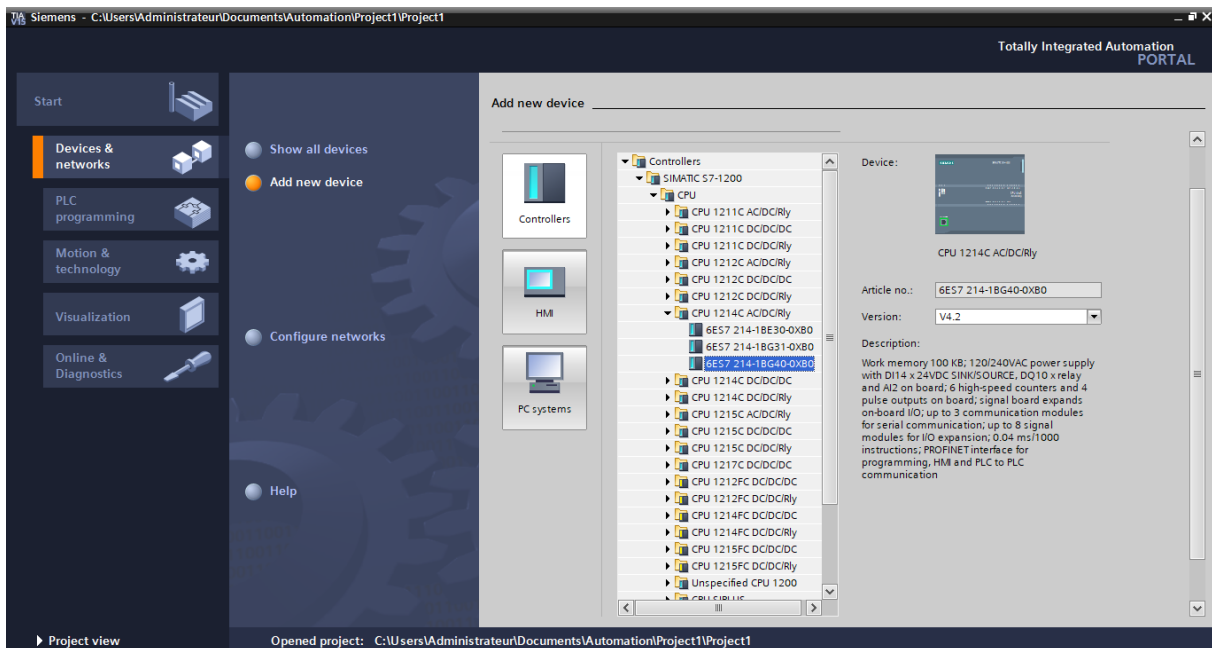


Figure 3.11 Paramétrage du matériel.

3.6.1 Adressage des E/S :

Pour connaître l'adressage des entrées et sorties présentes dans la configuration matérielle, il faut aller dans « appareil et réseau » dans le navigateur du projet.

Dans la fenêtre de travail, il est important de s'assurer d'être dans l'onglet « Vue des appareils » et de sélectionner l'appareil voulu.

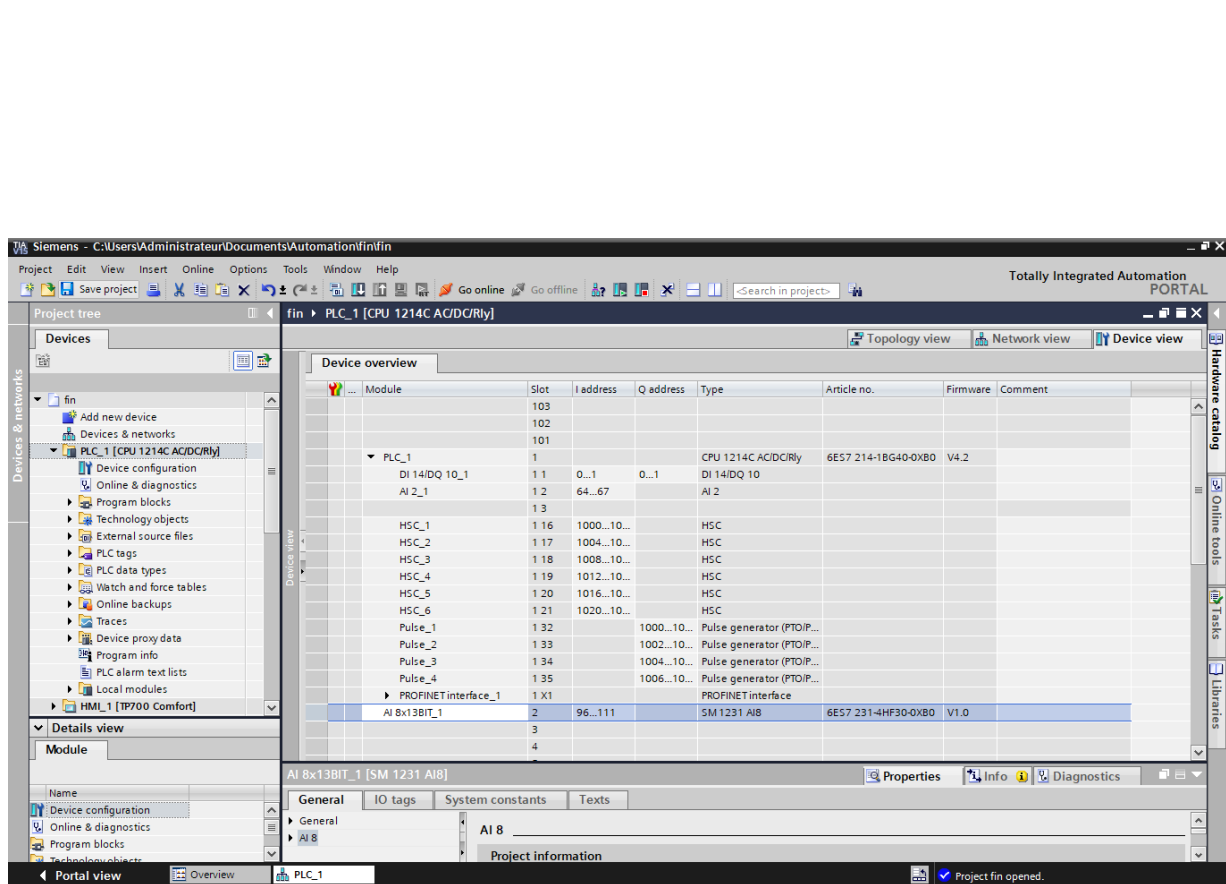


Figure 3.12 Adressage des E/S.

3.6.2 Adresse Ethernet de la CPU :

Toujours dans les propriétés de la CPU, il est possible de définir son adresse Ethernet. Un double clic sur le connecteur Ethernet de la station fait apparaître la fenêtre d'inspection permettant de définir ses propriétés.

Pour établir une liaison entre la CPU et la console de programmation, il faut affecter aux deux appareils des adresses appartenant au même réseau.

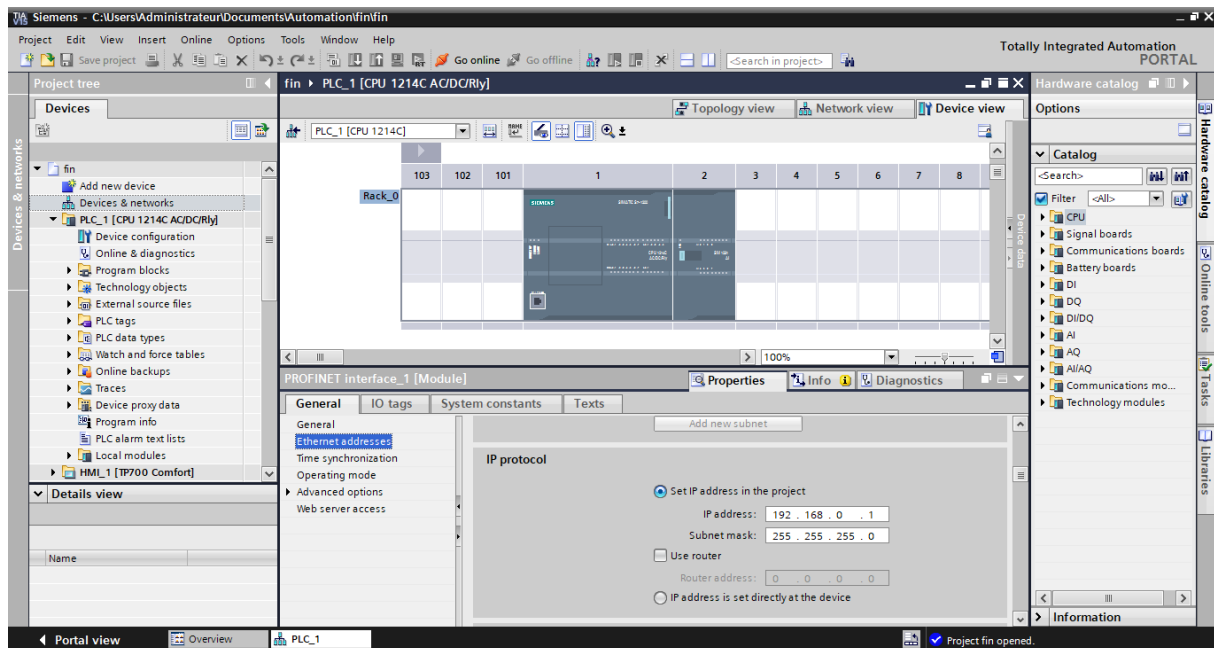


Figure 3.13 Adresse Ethernet de la CPU.

3.7 Compilation et chargement de la configuration matérielle :

Une fois la configuration matérielle réalisée, il faut la compiler et la charger dans l'automate. La compilation se fait à l'aide de l'icône « compiler » de la barre de tâche. Il faut d'abord sélectionner l'API dans le projet ensuite cliquer sur l'icône « compiler ».

En utilisant ces étapes, nous effectuons une compilation matérielle et logicielle.

Une autre solution pour compiler est de faire un clic droit sur l'API dans la fenêtre du projet et de choisir l'option Compiler

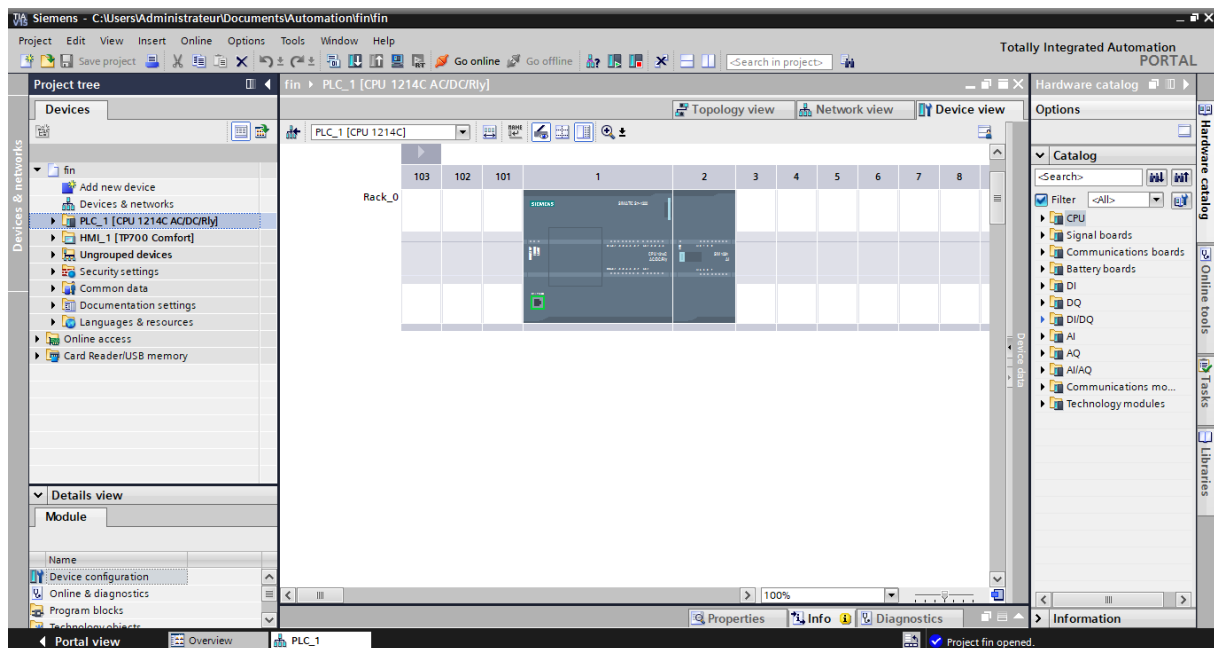


Figure 3.14 Configuration matérielle.

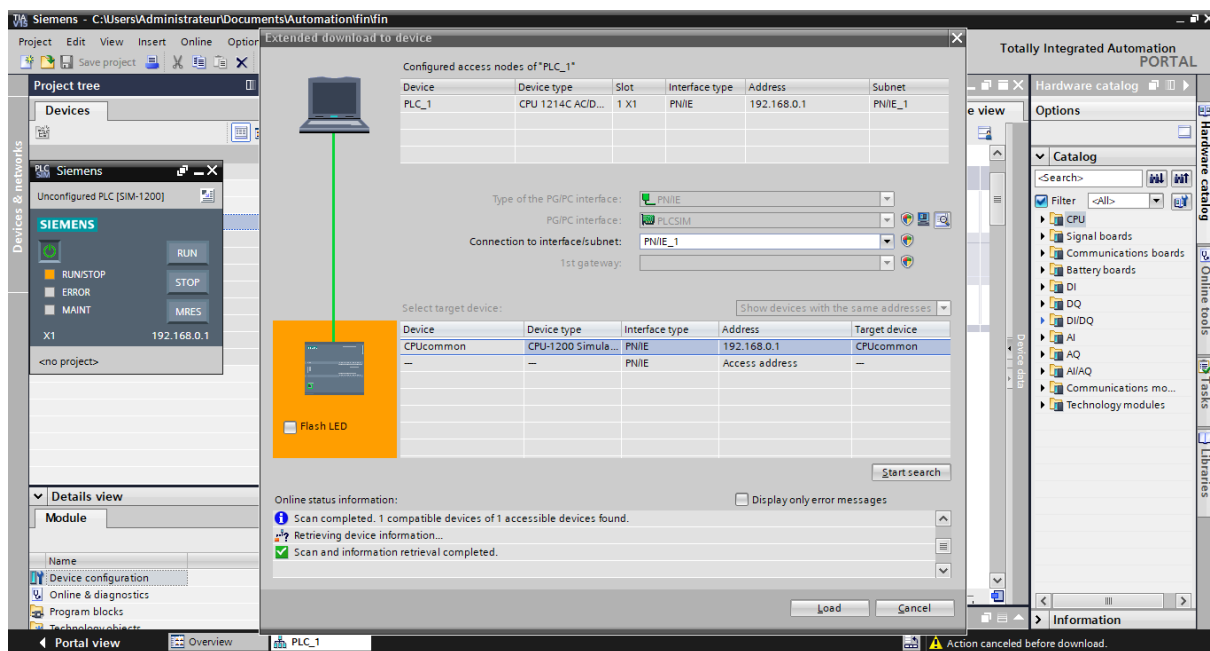


Figure 3.15 Mode de connexion.

3.8 Les variables API :

3.8.1 Adresse symbolique et absolue :

Dans TIA Portal, toutes les variables globales (entrées, sorties, mémentos,..) possèdent une adresse symboliques une adresse absolue.

- A L'adresse absolue** représente l'identificateur d'opérande (I, Q, M,...) et son adresse et numéro de bit.
- B L'adresse symbolique** correspond au nom que l'utilisateur a donné à la variable (ex: Bouton_Marche).

Le lien entre les adresses symbolique et absolue se fait dans la table des variables API. Lors de la programmation.

3.9 WINCC sur TIA PORTAL :

WinCC, intégré au TIA Portal est le logiciel pour toutes les applications IHM – des simples solutions de commande par Basic Panels aux visualisations de process sur systèmes multipostes à base de PC.

Le SIMATIC WinCC dans le TIA Portal fait partie d'un nouveau concept d'ingénierie intégré qui offre un environnement d'ingénierie homogène pour la programmation et la configuration de solutions de commande, de visualisation et d'entraînement.

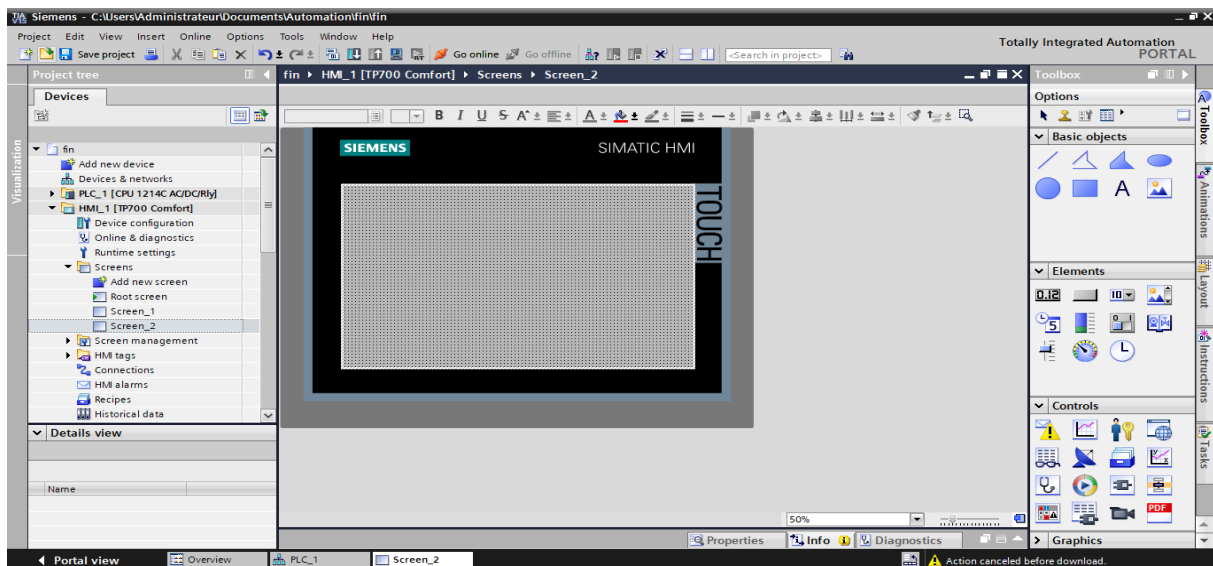


Figure 3.16 Vue de WINCC.

3.10 Création du Programme :

3.10.1 Structure de programmation TIA PORTAL:

A Programmation linéaire :

Ce type de programmation est utilisé pour des commandes simples et de volumes moins importants. Les multiples opérations et instructions de différentes fonctions sont stockées dans un seul bloc d'organisation (OB1) qui traite cycliquement le programme.

B Programmation structurée

Pour les automatismes complexes, le programme utilisateur est subdivisé en fonctions principales que l'on programme à l'aide des blocs de codes (OB, FB, FC). L'OB1 contient le programme principal qui sera exécuté par la CPU puis il fait appel aux autres blocs quand il le faut pour délivrer les données correspondantes, et dès que la CPU termine l'exécution du programme stocké dans le bloc appelé, elle reviendra pour suivre l'exécution du programme du bloc appelant. Ce genre de traitement de programme est utilisé lorsque le procédé à automatiser est complexe car il permet de simplifier l'organisation, la gestion et le test du programme.

Le TIA portal offre les blocs utilisateur suivants pour la programmation structurée :

B.1 Bloc d'organisation (OB) :

Un OB est appelé cycliquement par le système d'exploitation et constitue donc l'interface entre le programme utilisateur et le système d'exploitation. L'OB contient des instructions d'appels de bloc indiquant à l'unité de commande de l'automate l'ordre dans lequel il doit traiter les blocs.

B.2 Bloc fonctionnel (FB) :

Le FB dispose d'une zone de mémoire qui lui est affectée en propre. Il est possible d'affecter un bloc de données (DB) au FB à l'appel du bloc. Il est possible d'accéder aux données du DB d'instance via les appels contenus dans le FB. Nous pouvons affecter plusieurs DB à un FB. Il est possible d'appeler d'autres FB et FC dans un bloc fonctionnel via instructions d'appels de bloc.

B.3 Fonction (FC) :

Une FC ne possède pas une zone du mémoire propre. Les données locales d'une fonction sont perdues après l'exécution de la fonction. Il est également possible d'appeler d'autre FB et FC dans une fonction via des instructions d'appels de blocs.

B.4 Blocs de données (DB) :

Les DB sont utilisés pour la mise à disposition d'espace mémoire pour les variables types de données. Il existe deux types de blocs de données. Les DB globaux dans lesquels tous les OB, FB et FC peuvent lire les données enregistrées ou écrire des données et les DB d'instance qui sont affectées à un FB donné.

B.5 La structure du programme réalisé :

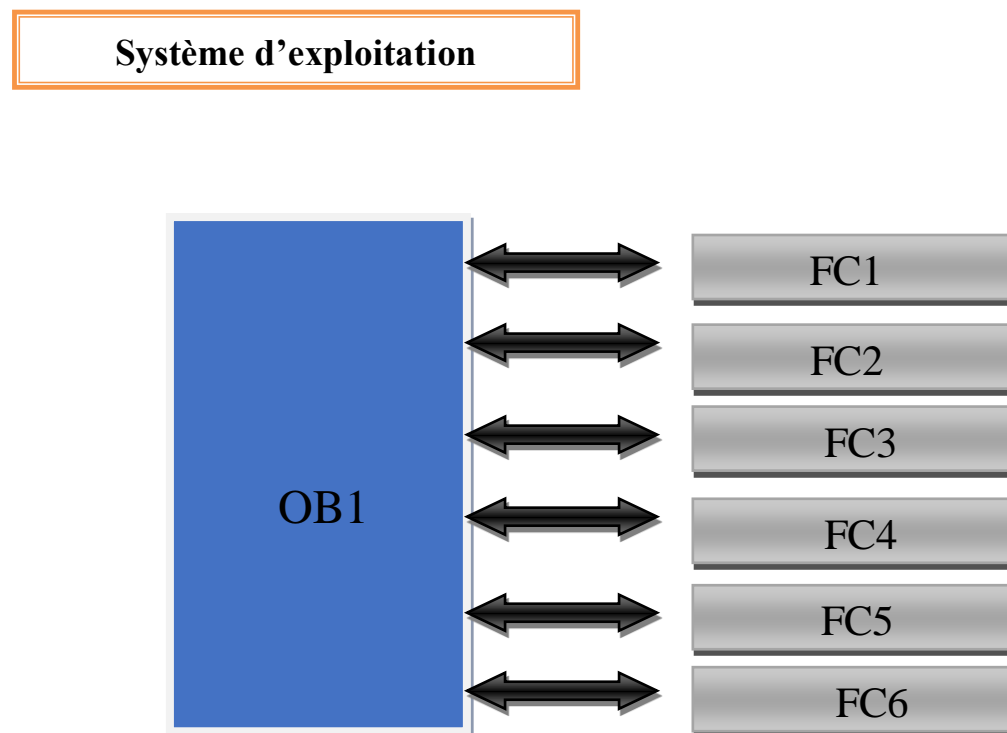


Figure 3.17 Programmation structurée de la machine injection plastique.

FC1 : Le programme de capteur position vérin de vis

FC2 : Le programme de capteur position vérin de moule

FC3 : Le programme de capteur position vérin d'éjecteur

FC4 : capteur de température zone 1

FC5 : capteur de température zone 2

FC6 : capteur de température zone 3

Nous avons réalisé notre programme avec des blocs utilisateurs. **La figure IV.17** représente la structure du programme de conduite sécurisée que nous avons développé.

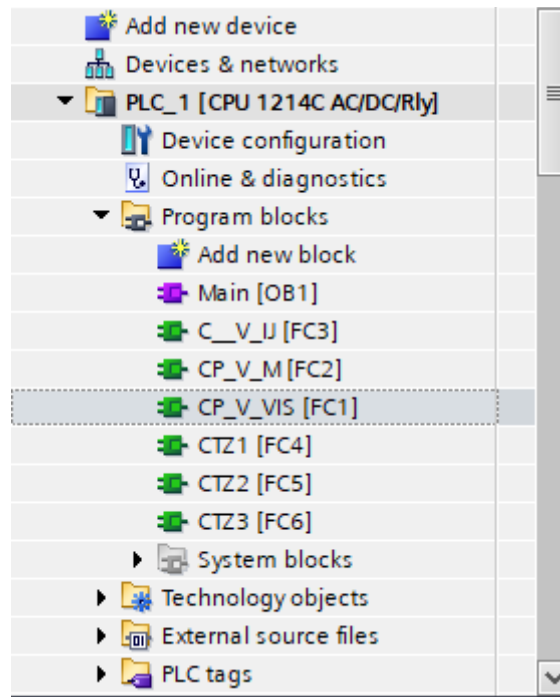


Figure 3.18 Structure du programme

C Simulation du programme réalisé :

Après l'élaboration du programme de commande de notre système à automatiser, nous arrivons à l'étape décisive du travail effectué. Cette étape est la validation du programme par simulation et vérification de son bon fonctionnement. Le rogramme comporte les étapes suivantes :

D Exemple de simulation d'une partie du programme développé:

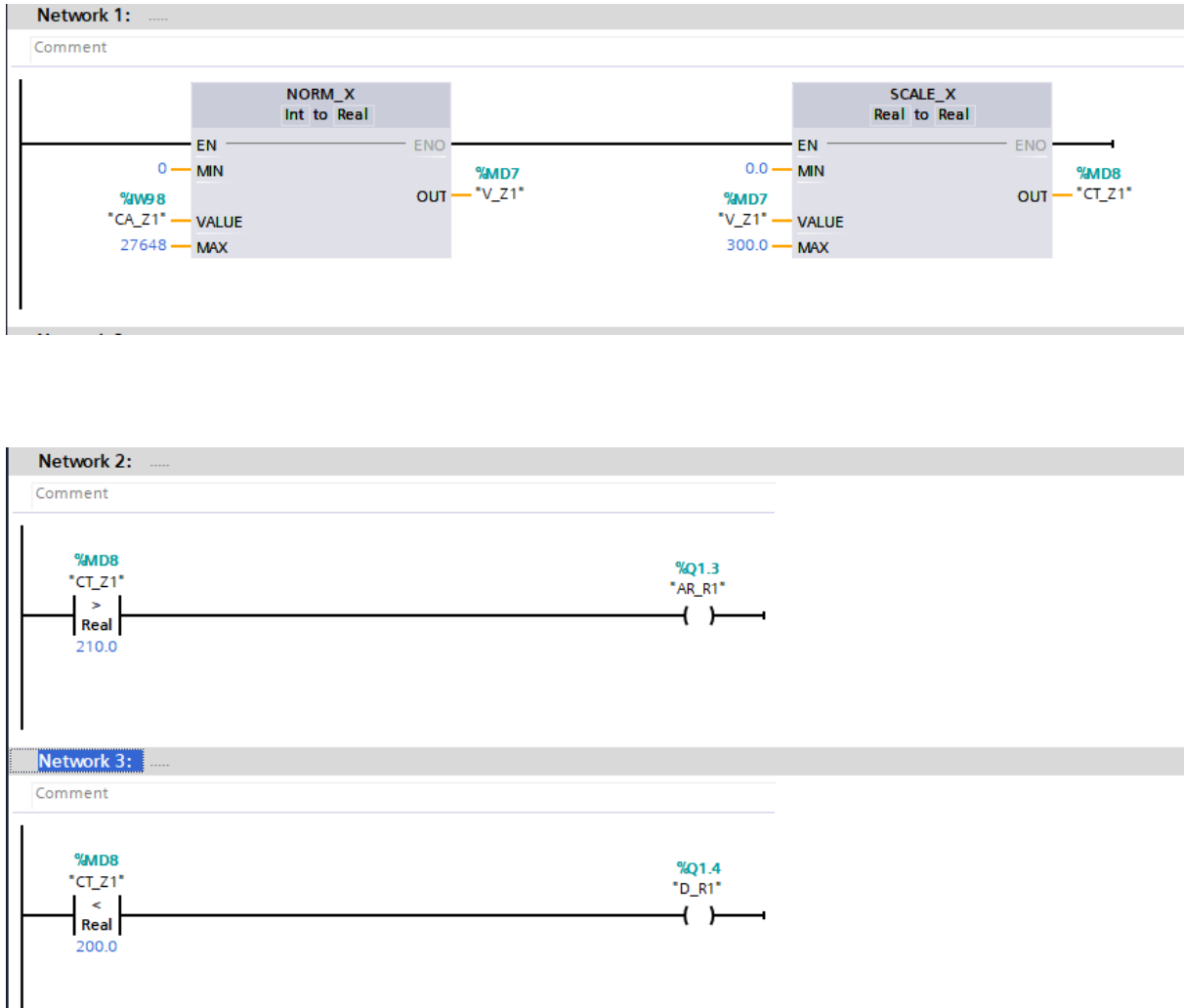


Figure 3.19 fonctionnement de résistance

La table des variables d'entrées

entre							
	Name ▲	Data type	Address	Retain	Acces...	Writa...	Visi...
1	CA_Z1	Word	%IW98	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
2	CA_Z2	Int	%IW100	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
3	CA_Z3	Int	%IW102	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
4	CAV_IJ	Int	%IW96	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
5	CAV_M	Int	%IW66	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
6	CAV_VIS	Word	%IW64	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
7	CCH_AR	Bool	%I0.7	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
8	CCH_AV	Bool	%I0.6	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
9	COPON	Bool	%I1.1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
10	CPV_IJ	Real	%MD5	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
11	CPV_M	Real	%MD2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
12	CPV_VIS	Real	%MD0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
13	CREFA_A	Bool	%I1.7	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
14	CREFA_D	Bool	%I1.6	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
15	CT_Z1	Real	%MD8	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
16	CT_Z2	Real	%MD10	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
17	CT_Z3	Real	%MD12	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
18	CVR_A	Bool	%I1.5	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
19	CVR_D	Bool	%I1.4	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
20	DCY	Bool	%I1.3	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
21	EV_IJA	Bool	%M21.5	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
22	EV_IJD	Bool	%M21.2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
23	EV_MA	Bool	%M20.3	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
24	EV_MD	Bool	%M20.2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
entre							
	Name ▲	Data type	Address	Retain	Acces...	Writa...	Visi...
25	EV_VISA	Bool	%M20.7	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
26	EV_VISD	Bool	%M20.6	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
27	maintian	Bool	%M18.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
28	maintian_1	Bool	%M18.1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
29	maintian_2	Bool	%M18.2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
30	maintian_3	Bool	%M18.3	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
31	maintian_4	Bool	%M18.4	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
32	maintian_5	Bool	%M18.5	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
33	PORT	Bool	%I1.2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
34	REF_A	Bool	%M21.4	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
35	REF_D	Bool	%M21.3	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
36	SV_IJA	Bool	%M21.1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
37	SV_IJD	Bool	%M21.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
38	SV_MA	Bool	%M20.1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
39	SV_MD	Bool	%M20.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
40	SV_VISA	Bool	%M20.5	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
41	SV_VISD	Bool	%M20.4	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
42	V_Z1	Real	%MD7	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
43	V_Z2	Real	%MD9	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
44	V_Z3	Real	%MD11	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
45	VV_IJ	Real	%MD4	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
46	VV_M	Real	%MD1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
47	VV_VIS	Real	%MD3	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
48	<Add new>			<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Figure 3.20 Table des variables E

La table des variables sorties :

	Name	Data type	Address	Retain	Acces...	Writa...	Visibl...
1	SV_VIS	Bool	%Q0.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
2	EV_VIS	Bool	%Q0.1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
3	SV_IJ	Bool	%Q0.2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
4	EV_IJ	Bool	%Q0.3	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
5	SV_M	Bool	%Q0.4	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
6	EV_M	Bool	%Q0.5	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
7	CH_AV	Bool	%Q0.6	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
8	CH_AR	Bool	%Q0.7	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
9	VR_D	Bool	%Q1.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
10	VR_A	Bool	%Q1.1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
11	REF	Bool	%Q1.2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
12	AR_R1	Bool	%Q1.3	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
13	D_R1	Bool	%Q1.4	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
14	AR_R2	Bool	%Q1.5	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
15	D_R2	Bool	%Q1.6	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
16	AR_R3	Bool	%Q1.7	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
17	D_R3	Bool	%Q2.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
18	<Add new>			<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Figure 3.21 Table des variables S

E Simulation du bloc :

The screenshot displays the 'Extended download to device' dialog in Siemens TIA Portal. The 'Configured access nodes of "PLC_1"' table is as follows:

Device	Device type	Slot	Interface type	Address	Subnet
PLC_1	CPU 1214C ACID...	1 X1	PNIE	192.168.0.1	PNIE_1

The 'Type of the PG/PC interface' is set to 'PNIE'. The 'PG/PC interface' is 'PLCSIM'. The 'Connection to interface/subnet' is 'PNIE_1'. The '1st gateway' is empty.

The 'Select target device:' table is as follows:

Device	Device type	Interface type	Address	Target device
CPUcommon	CPU-1200 Simula...	PNIE	192.168.0.1	CPUcommon
--	--	PNIE	Access address	--

The 'Flash LED' checkbox is checked. The 'Online status information' shows: 'Scan completed. 1 compatible devices of 1 accessible devices found.', 'Retrieving device information.', and 'Scan and information retrieval completed.' The status bar at the bottom indicates 'The project file was saved successfully.'

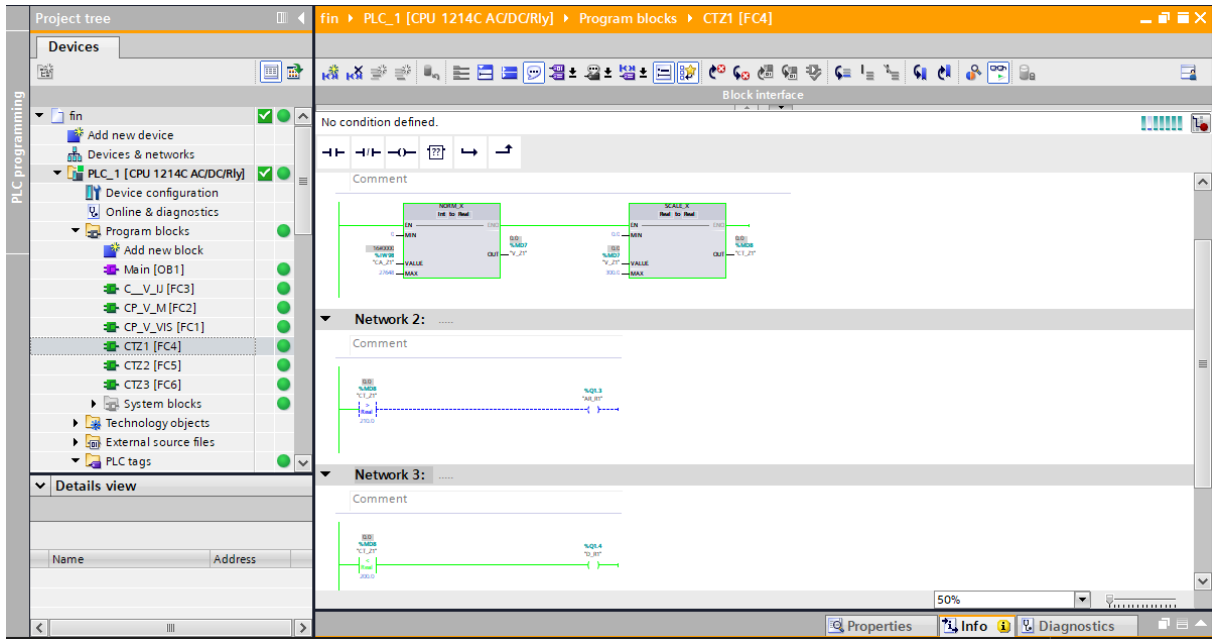


Figure 3.23 Simulation du bloc FC4

3.11 Conclusion :

Dans ce chapitre, on a présenté un aperçu sur les systèmes automatisés et les API. Notre travail est basé sur l'automate S7-1200 de SIEMENS ainsi sur le logiciel associé tia portal. La dernière partie est consacrée à la configuration du matériel de la machine injection plastique et création du programme permettant de gérer le fonctionnement de ses actionneurs.

Chapitre 4 **SUPERVISION**

Introduction :

Il est aujourd'hui vital pour un industriel d'exploiter au maximum tous les potentiels d'optimisation, que permet l'ensemble du cycle de vie d'une machine. La supervision est une technique industrielle de suivi et de pilotage informatique de procédés de fabrication automatisés, elle concerne l'acquisition de données et des paramètres de commande des processus confiés à des automates programmables

4.1 Définition de la supervision industrielle :

La supervision est une forme évoluée de dialogue Homme-Machine, elle sert à représenter et surveiller l'état de fonctionnement d'un procédé.

La supervision est donc composée d'un ou plusieurs ordinateurs en réseau équipés du logiciel adéquat.

Mais celle-ci se trouve aussi sur des consoles ou des écrans tactiles, de plus en plus de supervision se font à l'aide d'ordinateur car leur coût de revient de plus en plus faible font de ces outils plus compétitifs et pratiques.

La mise en place d'un système de supervision permet de visualiser en temps réel la bonne marche de l'installation et d'être alerte immédiatement en cas de défaut ou d'alarme [9]

Les fonctions de la supervision sont nombreuses, on peut citer quelques-unes :

- Assurer la communication entre les équipements d'automatismes et les outils informatiques d'ordonnancement et de gestion de production.
- Coordonner le fonctionnement d'un ensemble de machines enchaînées constituant une ligne de production, en assurant l'exécution d'ordres communs (marche, arrêt, etc.)
- Répondre à des besoins nécessitant en général une puissance de traitement importante.
- Assiste l'opérateur dans les opérations de diagnostic et de maintenance

4.2 Les avantages de la supervision :

- Contrôler la disponibilité des services/fonctions
- Contrôler l'utilisation des ressources
- Vérifier qu'elles sont suffisantes (dynamique)
- Détecter et localiser des défauts
- Diagnostic des pannes
- Prévenir les pannes/défauts/débordements (pannes latentes)
- Prévoir les évolutions
- Suivi des variables

4.3 Structure d'un système de supervision :

Un réseau de supervision est souvent constitué de :

- Un ou plusieurs appareils de terrain d'interfaçage de données i.e. les API
- Un système de communications pour transférer les données entre les interfaces de données, et entre les api et les interfaces de supervision
- PC master terminal unit (MTU), dédié à l'administration du système et au paramétrage de l'application. [9]

4.4 Présentation WINCC :

WinCC, intégré au TIA Portal est un système HMI performant, il nous permet de visualiser le processus, ce qui facilite la surveillance par un graphisme à l'écran, dès qu'un état évolue, l'affichage est mis à jour

4.5 les vues de supervision de la machine injection plastique :

On a 3 vues :

MODE automatique :

La vue principale permet la supervision de mode automatique et comporte les différents boutons de navigation vers les autres vues

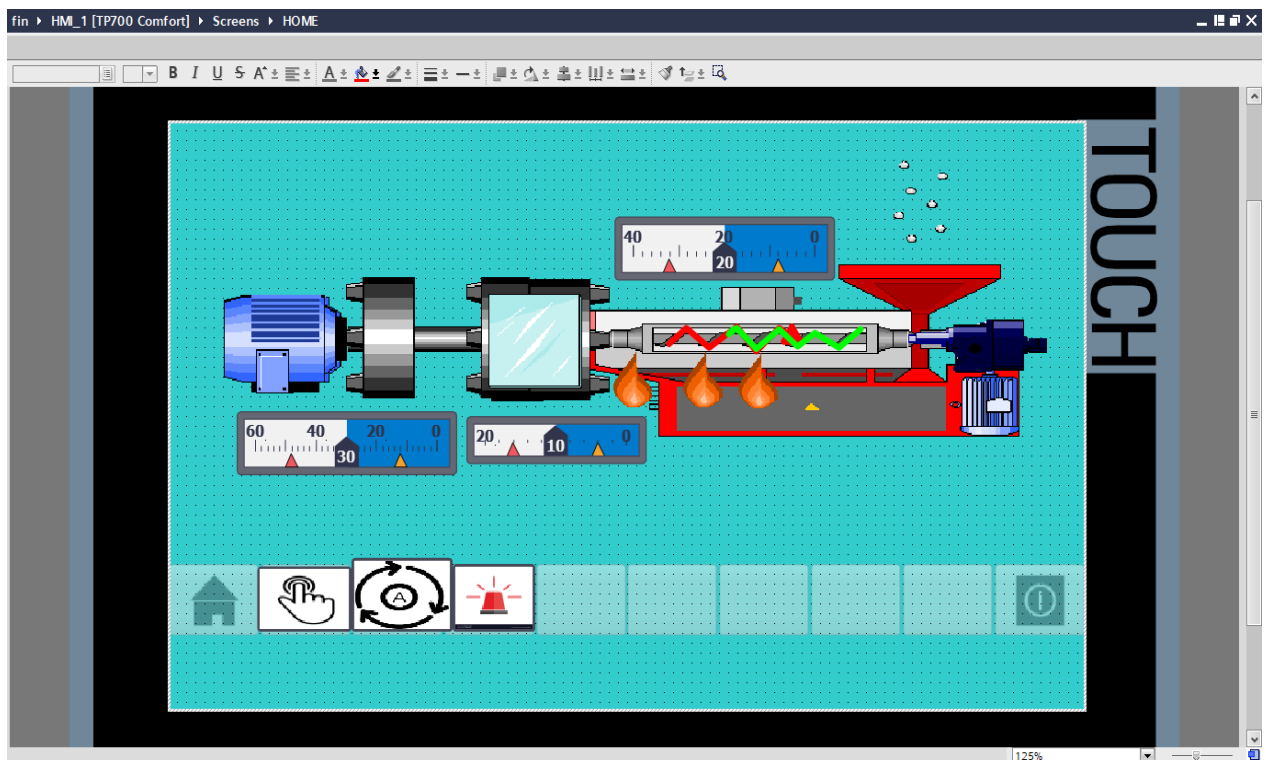


Figure 4.1 VUE MODE AUTOMATIQUE .

MODE manuel :

La vue mode manuel permet de voir le fonctionnement de la machine en mode manuel et de navigation vers les autres vues

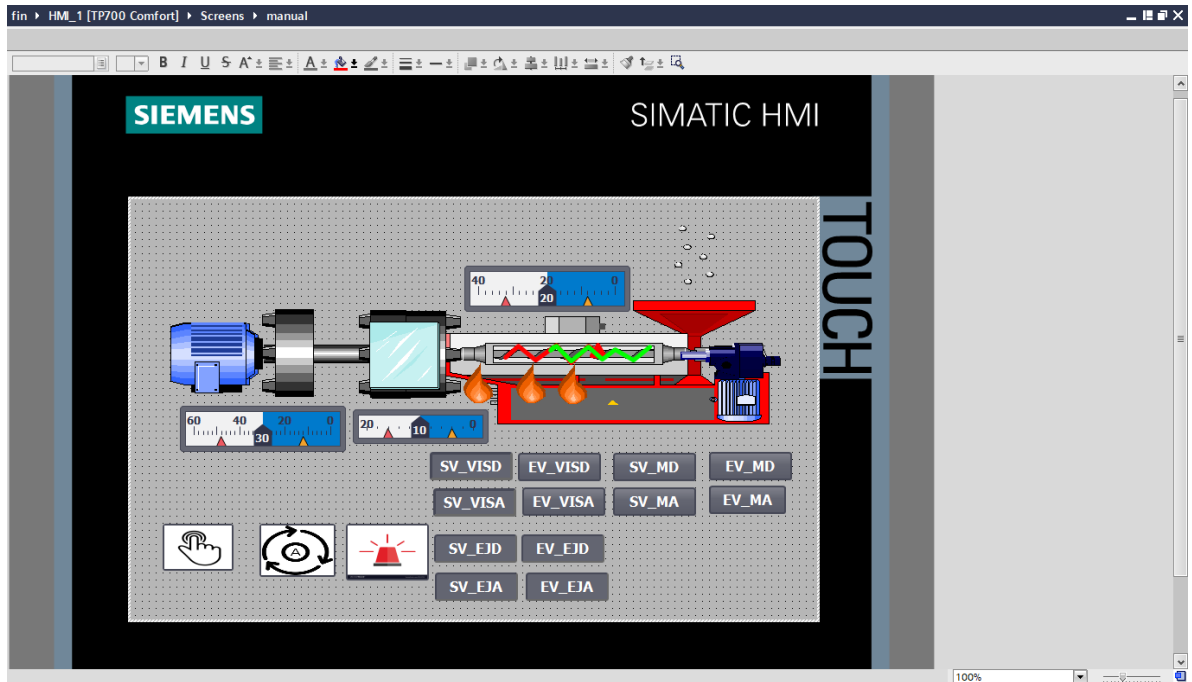


Figure 4.2 Vue mode manuel.

Vue des alarmes :

Cette vue permet de visualiser l'état de marche de tous les capteurs et de nous indiquer une alarme en cas de défaillance des capteurs

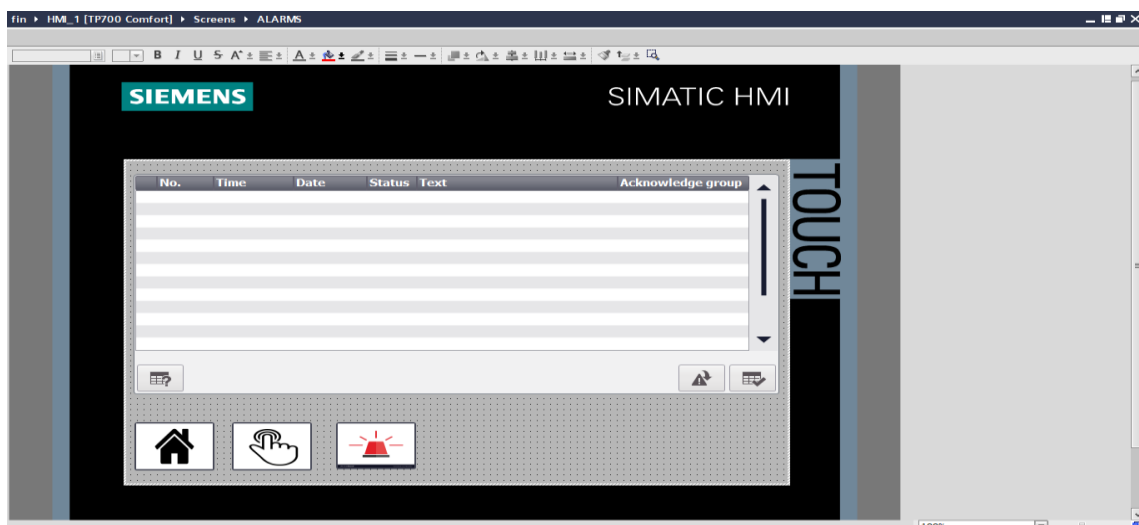


Figure 4.3 vue des alarmes .

Voici la figure de la configuration des alarmes :

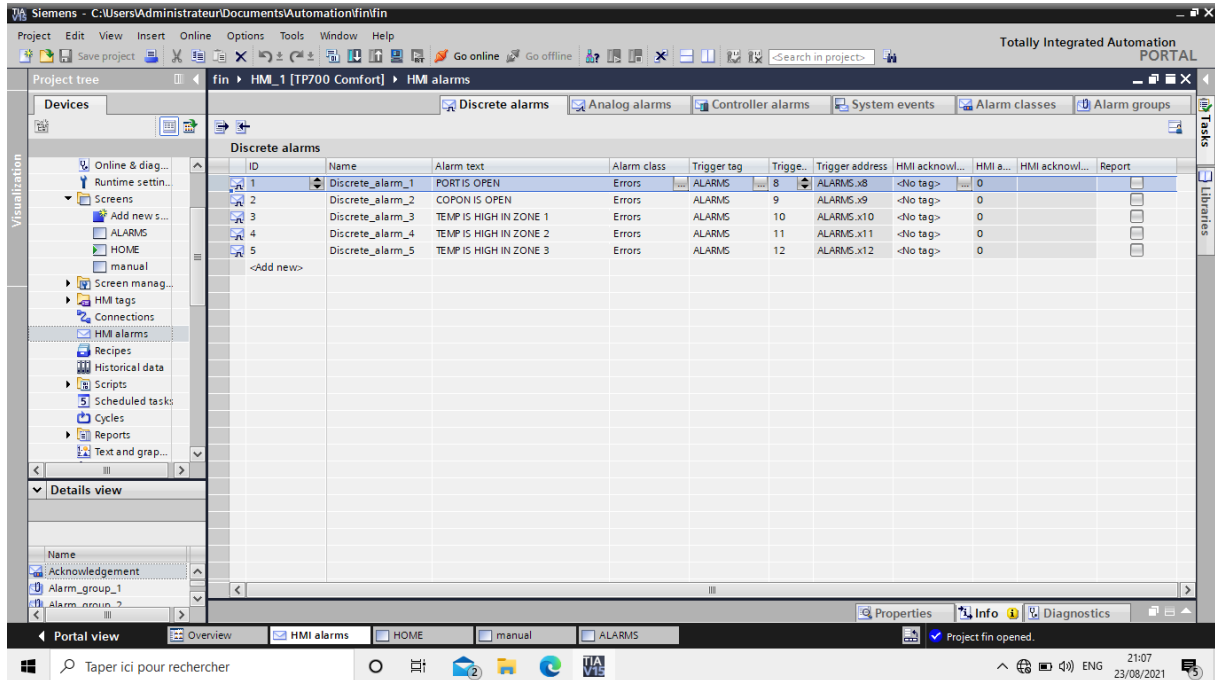


Figure 4.4 vue des alarmes .

4.5 CONCLUSION :

Dans ce chapitre, nous avons donné un aperçu général sur la supervision, ainsi que la présentation de logiciel WinCC flexible et la création d'une IHM laquelle permettra de gérer toutes les opérations assignées de la machine, par des moyens d'ingénierie simple et efficaces.

On a terminé par la présentation des principales vues de supervision de notre machine.

Conclusion générale :

Notre travail consistait à l'étude la programmation et la supervision d'une machine d'injection plastique au sein de l'entreprise SLR.

Pendant notre stage pratique, nous avons, en premier lieu, étudié le fonctionnement de la machine ainsi que la présentation de tous les composants constituant sa partie opérative.

Une fois le fonctionnement décrit, nous avons élaboré un GRAFCET. Ce dernier est la solution proposée pour l'automatisation de notre machine.

Le GRAFCET présente une série de tâches correspondantes aux différentes étapes du fonctionnement de la machine.

Côté automates programmables, on a porté notre choix sur l'API S7-1200 avec la CPU 1214C qui répond au cahier de charges, et qui sont des automates superpuissants disposant d'une structure modulaire extensible.

Le programme a été élaboré sous le logiciel TIA PORTAL V15, ce dernier est réalisé avec le langage à contacts (CONT), tout en se basant sur le GRAFCET.

En dernier lieu, nous avons exploité les performances du WinCC flexible intégré dans TIA PORTAL, permettant à l'opérateur de connaître l'état d'avancement du processus en temps réel et d'intervenir directement sur le pupitre de commande depuis la salle de contrôle.

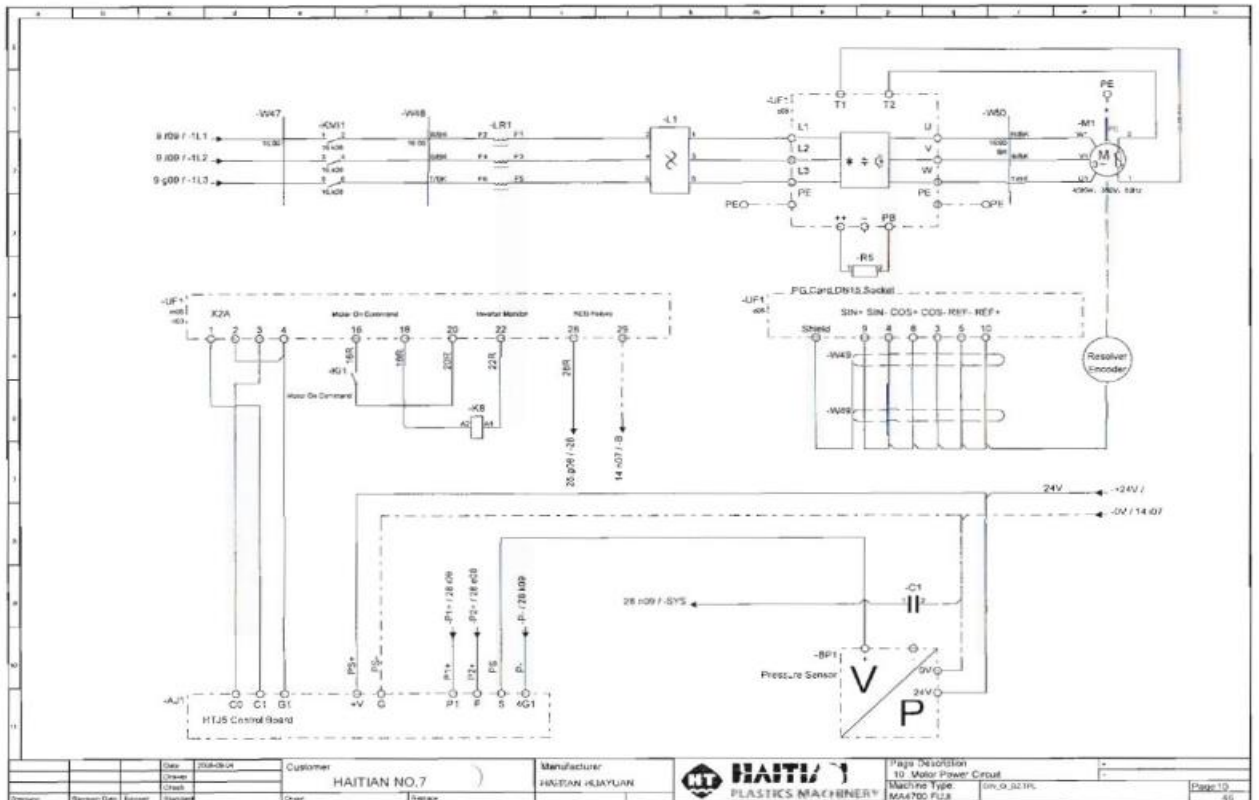
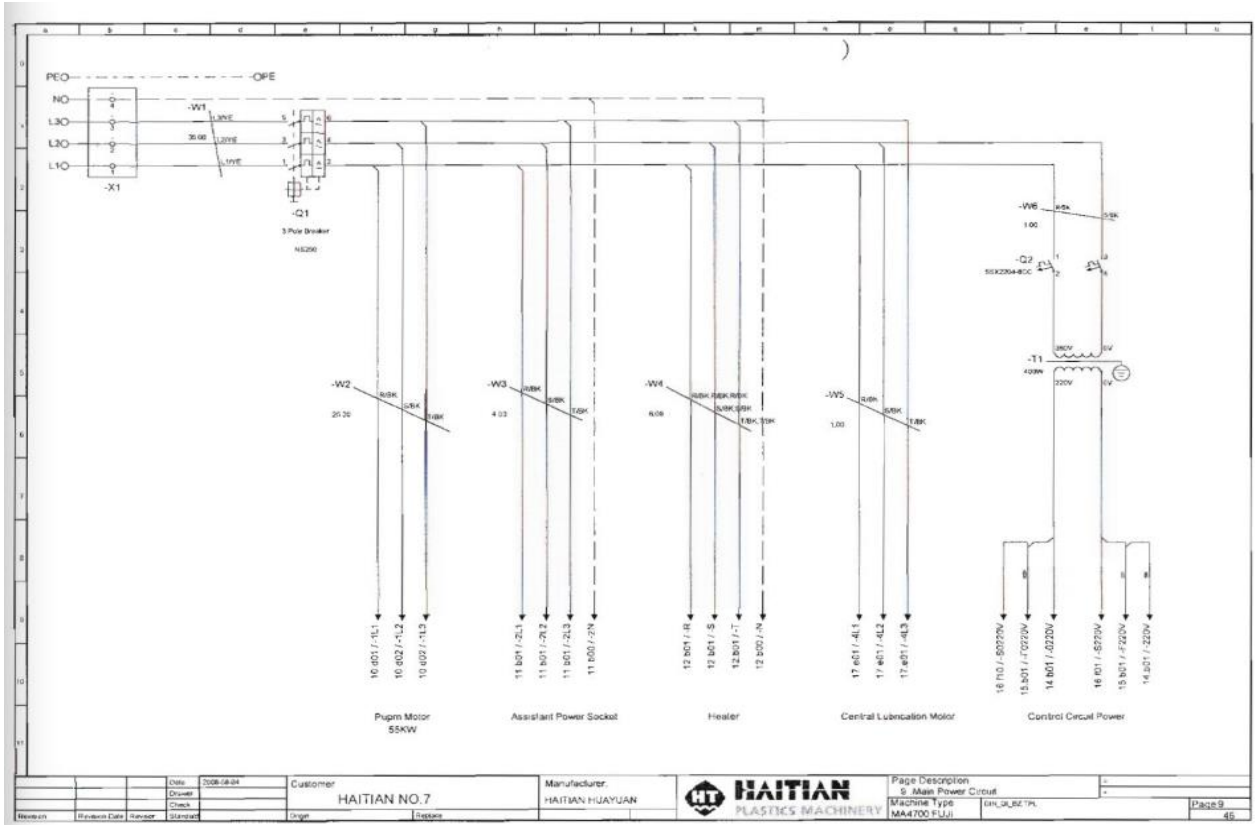
Ce projet nous a permis de bénéficier de plusieurs avantages :

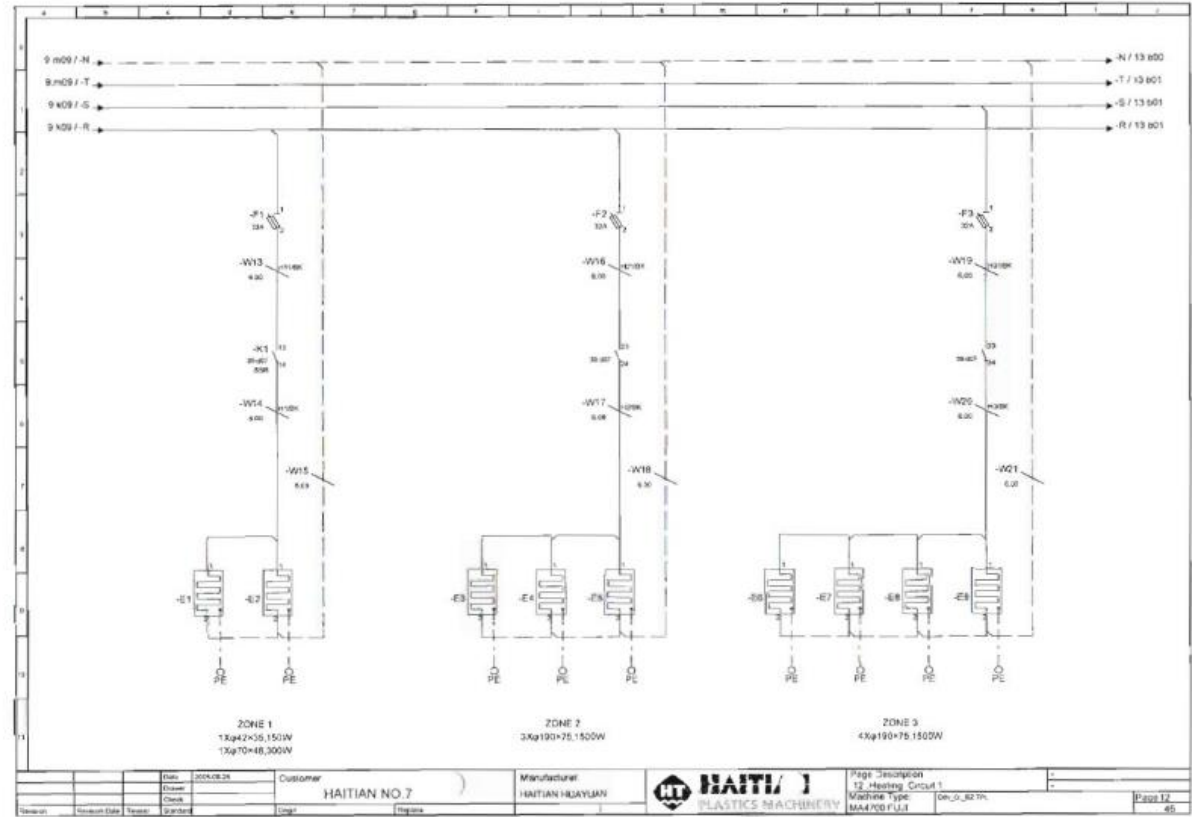
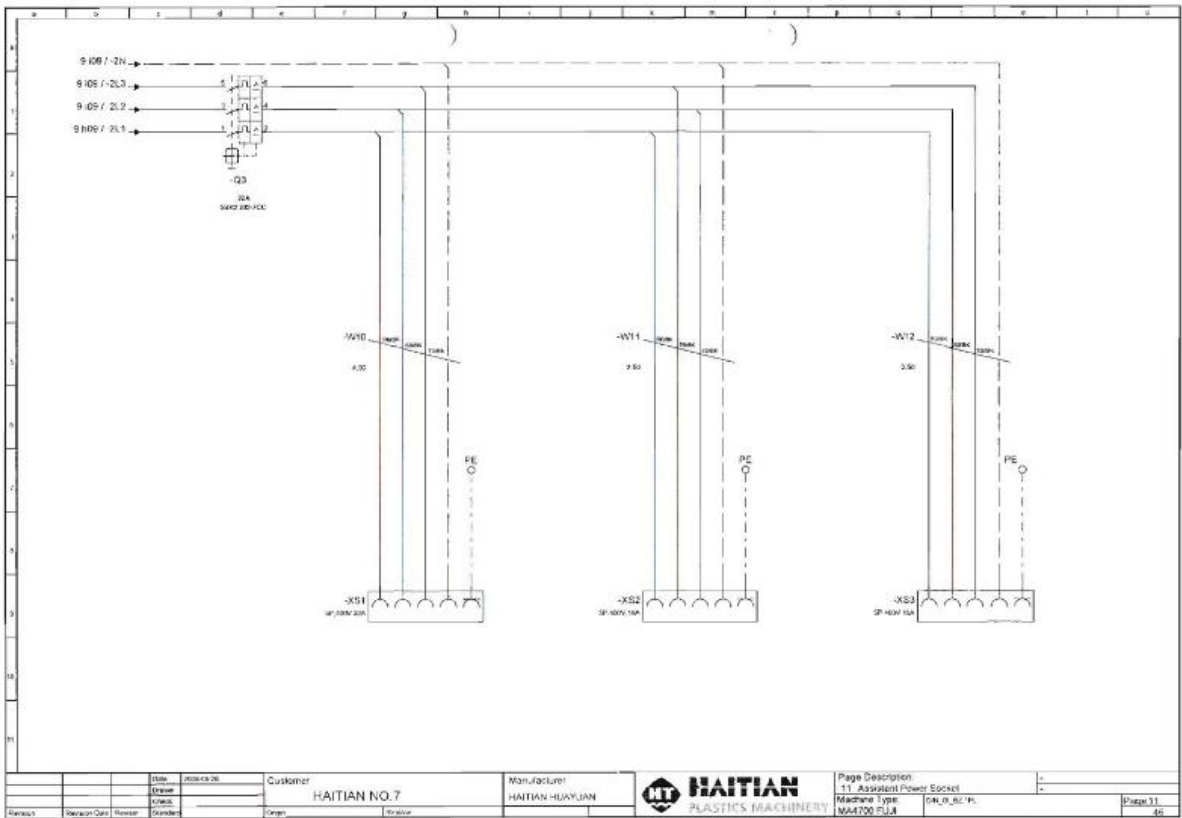
- Il a permis de nous familiariser avec les automates programmables s7-1200 et de nous initier encore plus sur leurs langages de programmation
- Et aussi de renforcer nos connaissances théoriques, par une expérience pratique non négligeable dans le domaine de l'automatisation
- A la fin on propose d'ajouter un convoyeur pour faciliter le déplacement de la pièce éjectée dans le futur .

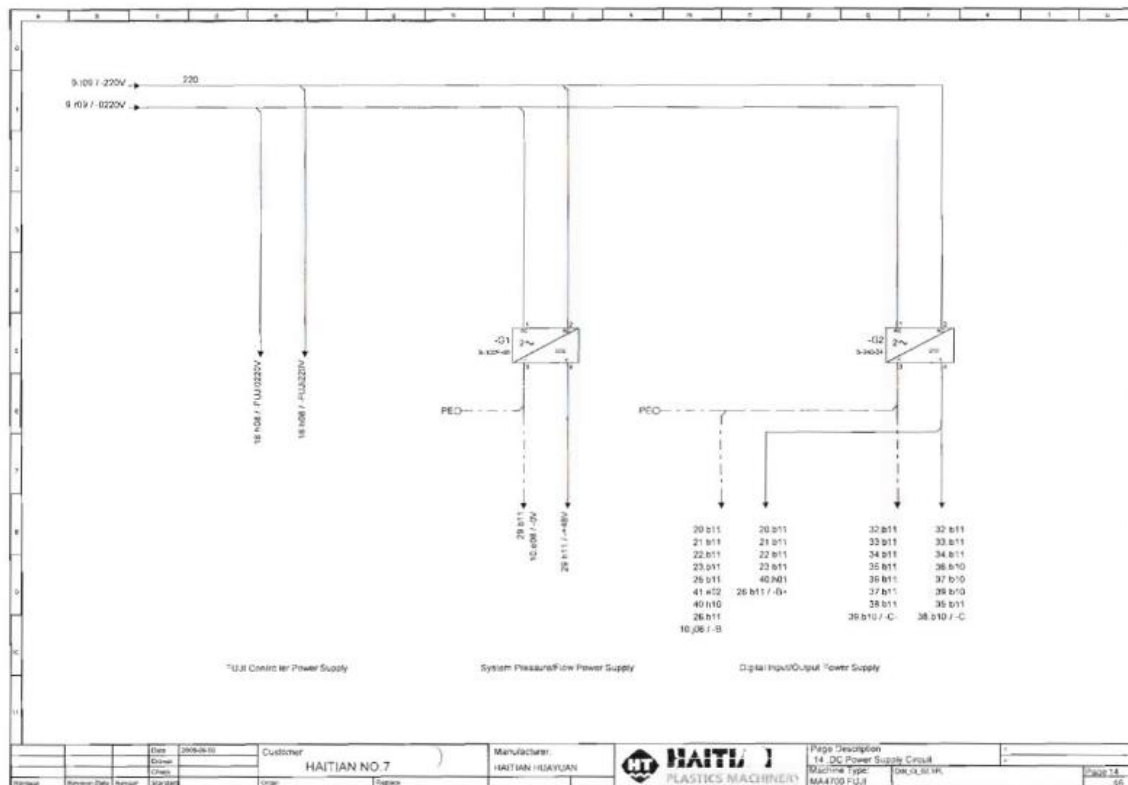
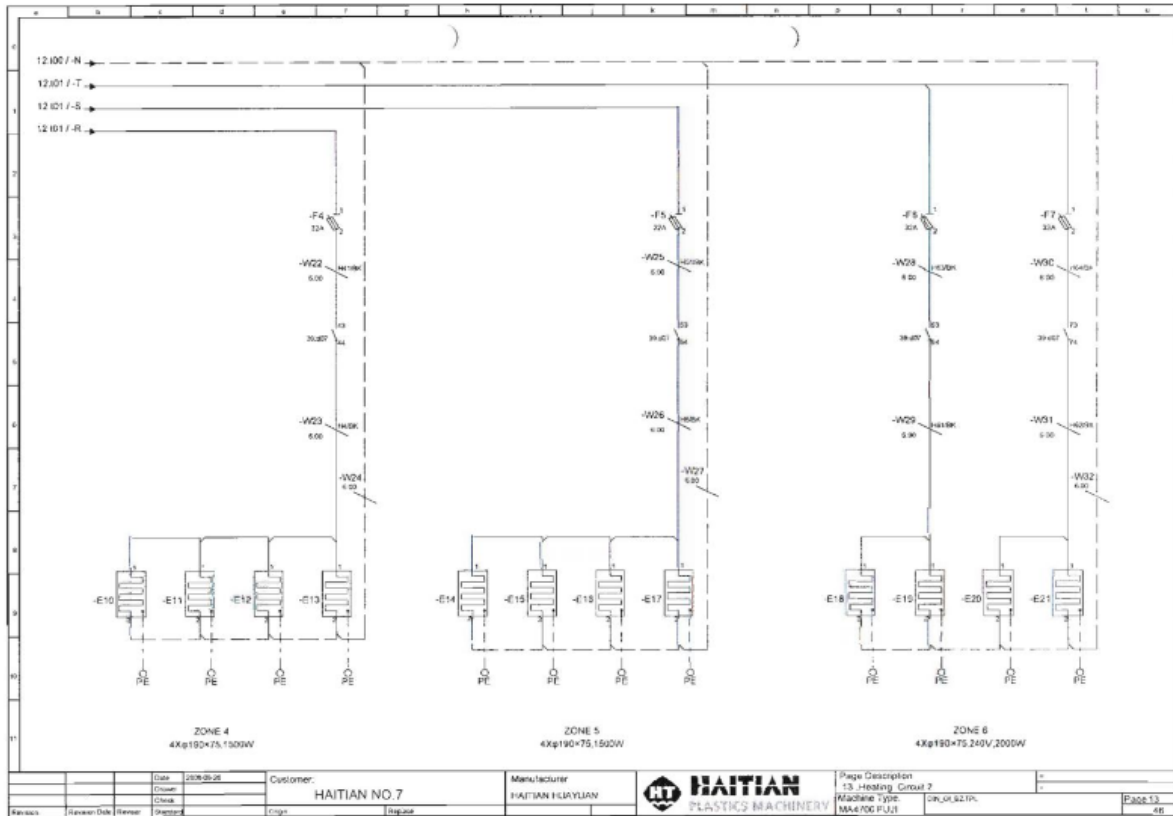
Bibliographie :

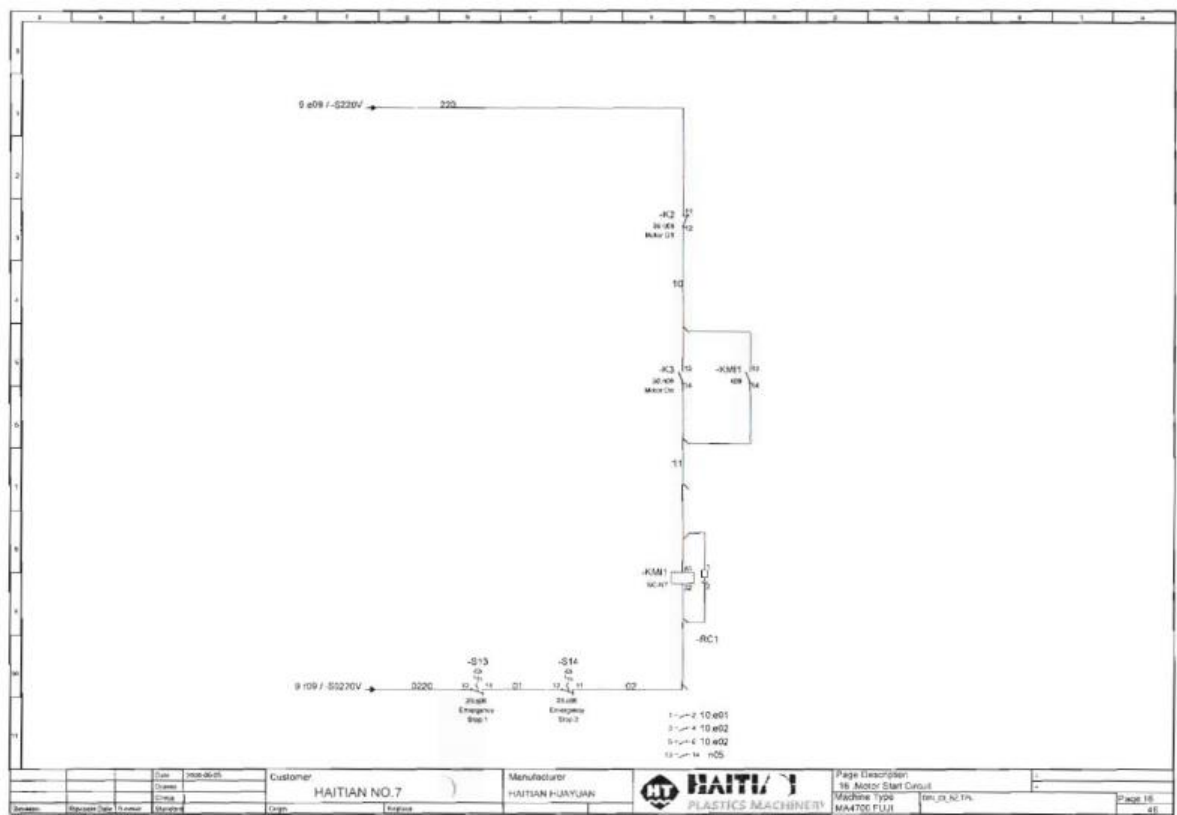
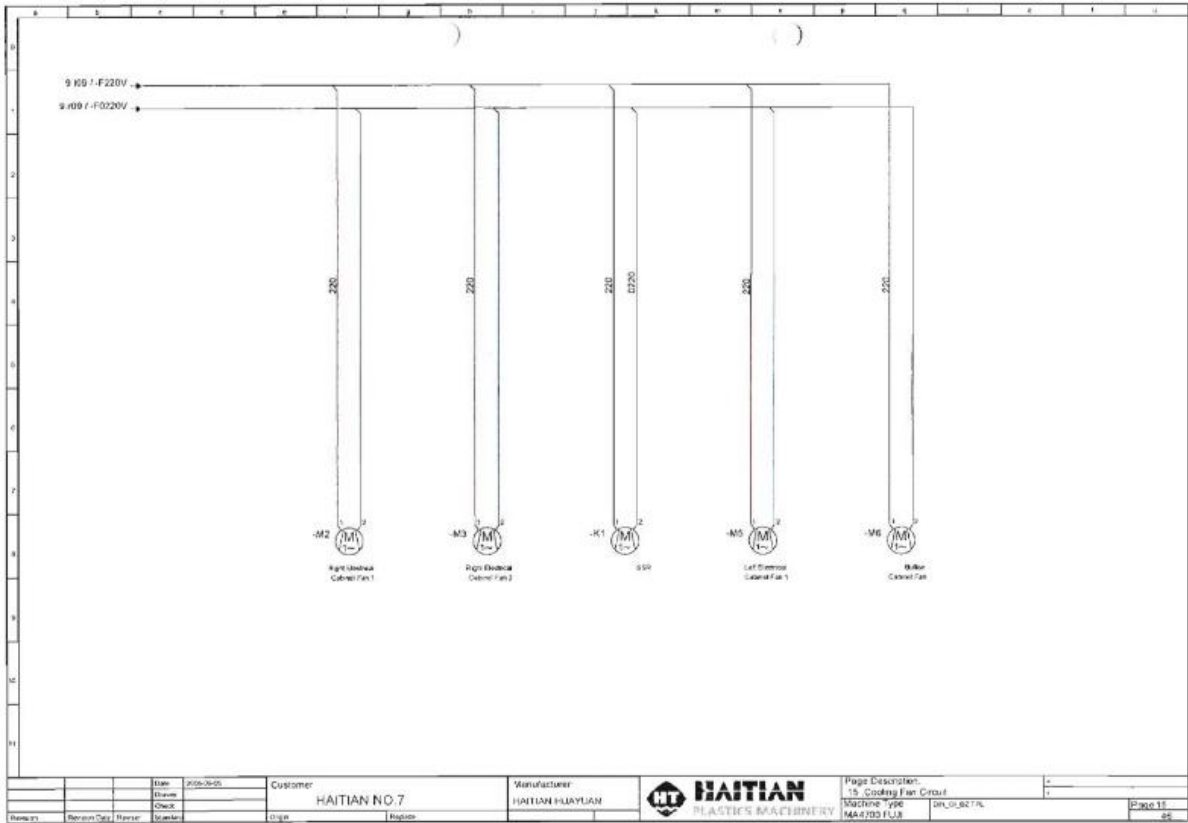
- [1] Explication sur la Presse à injection en Injection thermoplastique (prototechasia.com)
- [2] Figure 5 from Pilotage en Injection Plastique – Etat de l’Art | Semantic Scholar
- [3] Schwendiplasturgie injection 2008-2009
- [4] Cours GRAFCET
<http://fst.univ-guelma.dz/sites/default/files/field/Chapitre%202.GRAFCET.pdf>
- [5] Mme EL HAMMOUMI Cours Automatismes –GE1:
<http://www.est-usmba.ac.ma/coursenligne/GE-S2-M8.1-Automatismes%20logiques%20Industriels-CRS-EI%20Hammoumi.pdf>
- [6] Manual automate programmable s7-1200
https://support.industry.siemens.com/cs/attachments/109764129/s71200_system_manual_fr-FR_fr-FR.pdf
- [7] Manuel haitain machine injection
- [8] Gonei Kasmir. Développement d'un outil de design pour les buse thermique des moules a canaux chauds , Thèse de doctora,Ecole polytechniques de montréal, 1999.
- [9] Master ASE SMaRT – Introduction à la Supervision – P. Bonnet
- [10]Fiche de Connaissances Injection Innovation Technologique et Eco Conception académie Lyon
- [11] SAADA BECHIR cours MATIERE : CONCEPTION DE MOULE
CHAPITRE 3 : CONCEPTION DE MOULE D’INJECTION PLASTIQUE

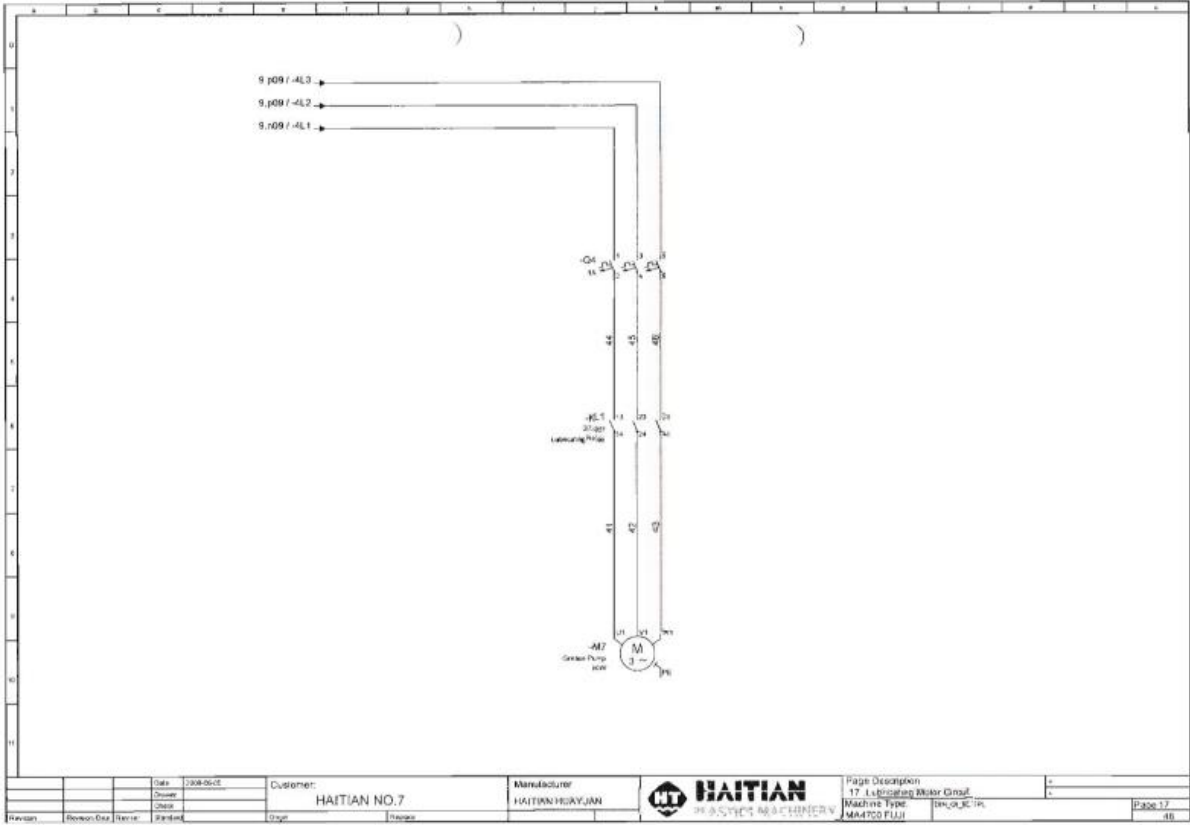
ANNEXE A





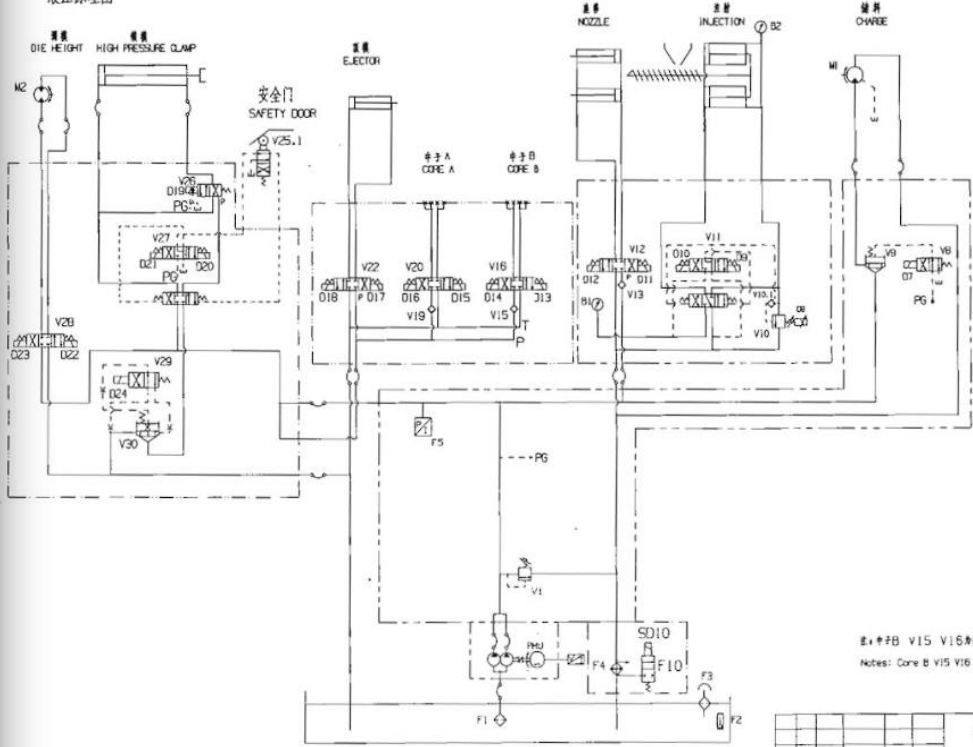






Les schémas électrique [7]

7.1 Hydraulic Circuit
 液压原理图



Ex 中#B V15 V16 为选件。
 Notes: Core B V15 V16 is optional.

				原 理 图	海天机械有限公司
				Circuit	
图号	分区	图号	图名	图例	图例
MA4700-04-00			液压原理图		
				MA4700-04-00	
				1/1	1/1
				共 1 页	第 1 页

7.3 Hydraulic Components List

No	Symbol	Code	Name	Model	Description	Remark
1.	PMU	E06363	J5 dynamic unit	HTF530(600)W1XJ5-55KW	Drive pump	
2.	M1	110093	Hydraulic Motor	ITM16-1800-000-D47	Drive screw charge	STF
3.	M2	112002	Hydraulic Motor	500300A3812ZAAAA	Drive mold adj. system	WHITE
4.	F1	130041	Oil filter	SF2-504-AF1S-M90-E1	Filter oil	
5.	F2	132018	Oil level	SNK127V-O-O-12-R	Oil level display	
6.	F3	131000	Suction	FSB 25 BM	Filter air	
7.	F3	131001	Suction	EF5-65	Filter air	
8.	F4	133003	Heat exchanger	SL-5.3M ²	Lower oil temperature	
9.	F5	134060	Pres. sensor	MSP400-200-B-4-B-1	Check system pressure	
10.	F10	136041	Valve	5281 (RC1 1/4), DC24V	Control oil temperature	
11.	V1	120016	Valve	CG2V-8FW-10	System safety valve	VICKERS
12.	V8	122001	Valve	DG4V-3-0AL-MUH5-60	Control charge	VICKERS
13.	V9	844154 844155	Valve cover Valve core	25-AM T-09-112-00 D25-BS T-09-113-00	Control charge	
14.	V10	125102	Valve	EBG-06-A-T-5007	Adjust charge and back pressure	YUKEN
15.	V10.1	126064	Valve	COBAXAN	Adjust charge and back pressure	
16.	V11	122032	Valve	DG5S-H8-3C-T-MUH5-60	Control inject and suck back	VICKERS
17.	V12	122004	Valve	DG4V-3-2C-MUH5-60	Control nozzle	VICKERS
18.	V13	124010	Valve	MC-02P-05-20	Nozzle adv. and hold pressure	
19.	V15	124010	Valve	MC-02P-05-20	Core B hold pressure	OPTIONAL
20.	V16	122004	Valve	DG4V-3-2C-MUH5-60	Control Core B	OPTIONAL
21.	V19	124010	Valve	MC-02P-05-20	Core A hold pres.	
22.	V20	122004	Valve	DG4V-3-2C-MUH5-60	Control Core A	VICKERS
23.	V22	122215	Valve	SWH-G03-C22-D24-20-M3	Eject motion	
24.	V25	848943	Valve	T-09-181-00	Hydraulic safety	
25.	V26	122033	Valve	DG5S-H8-6AL-E-MUH5-60	Mold close and differential	VICKERS
26.	V27	122276	Vane	DG5S-H8-2C-E-MUH5-60	Mold open/close	VICKERS
27.	V28	122004	Valve	DG4V-3-2C-MUH5-60	Control mold adj. motor	VICKERS
28.	V29	122000	Valve	DG4V-3-0A-MUH5-60	Mold open and back pressure	VICKERS
29.	V30	848068 848079	Valve cover Valve core	T-09-178-00 32-AM T-09-179	Mold open/close and back pressure	
30.	B1	135018	Pressure gauge	1453/0-25MPa, panel	System pressure display	IMT
31.	B2	135018	Pressure gauge	1453/0-25MPa, panel	Charge and back pressure display	IMT
32.		176A006	Motor assembly of oil pump	SMVP20-01-380/50AN 5bar	Servo motor cooling	

Note: We have the right of changing hydraulic components model in case of special requests or further technical development. Model in detail should be found on the machine you order.

Schéma hydraulique[7]