

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique



Université du 20 Août 1955 Skikda

Faculté de Technologie

Département de Pétrochimie

Mémoire fin d'étude

En vue de l'obtention du diplôme de **Master**

Spécialité : Automatisation en Industrie Pétrochimie

Thème

**Automatisation de trois systèmes industriels avec
API S7-300 et développement d'une interface graphique via Wincc :
Système de remplissage, portes serrures et système d'un réseau d'air comprimé**

Préparé par :

- Wassim-Zakaria Ganouche
- Ahmed Boutouil

Encadré par :

Pr. Youcef Zennir

Soutenu publiquement le : 13 / 07 / 2023

**Devant le jury
composé de :**

Président :	Noureddine Nafir	MCB	Univ-Skikda
Rapporteur :	Youcef Zennir	Pr	ISTA-Skikda
Examineur :	Saber Kared	MAA	Univ-Skikda

Année Universitaire 2022-2023

Sommaire

Sommaire
Liste des figures
Liste des abréviations
Résumé

Introduction Générale

Chapitre I : Etat de l'art des API

1. Généralité sur le domaine automatique et les API.....	3
1.1. Généralité sur le domaine automatique et les API	3
1.2. L'automatique	3
1.3. Notion système	4
1.4. Commande automatique.....	4
1.5. Intérêt de la commande	4
2. Les automates programmables industriels (API)	4
2.1 Historique.....	4
2.2. Définition.....	5
2.3. Place des automates programmables.....	6
2.4. Structure générale des automates programmables	7
2.5. Structure interne d'un automate programmable industriel API.....	7
2.6. Mode de fonctionnement.....	12
2.7. Choix d'un automate programmable industriel.....	15
2.8. Les avantages des automates programmables industriels.....	15
2.9. Les inconvénient des automates programmables industriels.....	15
3. L'automate SIEMENS S7-300.....	16
3.1. Description.....	16
3.2. Constitution de l'automate S7-300.....	16
3.3. Avantage.....	16
3.4. Caractéristique de l'automate S7-300.....	17

3.5. Composants de station S7-300.....	17
3.6. Modularité de S7-300.....	18
3.6.1. Unité central CPU.....	18
3.6.2. Interface MPI – Message Passing Interface.....	19
3.6.3. Signalisation des états.....	19
3.6.4. Commutateur de mode de fonctionnement.....	19
3.6.5. Carte mémoire.....	19
3.6.6 La pile.....	19
3.6.7 Modules d'alimentation.....	20
3.6.8. Modules des signaux (SM).....	20
3.6.9. Modules d'entrée /sortie TOR.....	21
3.6.10. Modules Entré Numériques.....	22
3.6.11. Modules sortie numériques.....	22
3.6.12. Modules entrée/sortie numériques.....	23
3.6.12. Modules analogiques.....	23
3.6.12.1. Modules d'entrée analogiques.....	23
3.6.12.2. Modules sortie analogiques.....	24
3.6.12.3. Modules entrée/sortie analogiques.....	25
3.6.13 Modules de simulation.....	26
3.6.14 Modules de communications.....	27
3.6.15. Console de programmation.....	28
3.7. Logiciel STEP-7.....	28
3.7.1. Introduction.....	28
3.7.2. Définition du STEP-7.....	28
3.7.3. Programmation de l'automate S7-300.....	29
3.7.4. Adressage des modules de S7-300.....	30
Conclusion.....	31

Chapitre II : Langages de programmation des API

Introduction.....	32
Différent langage de programmation d'un API.....	32

1. Le GRAFCET	33
1.1. Introduction au GRAFCET.....	33
1.2. Historique.....	33
1.3. Définition.....	33
1.4. Cahier de Charge.....	34
1.5. Les éléments de base du GRAFCET.....	34
1.6 Règles d'évolution du GRAFCET	38
1.7. Configuration courante.....	39
2. Langage a contact (LADDER)	41
2.1 Définition	41
2.2. Objectif.....	42
2.3. Principe.....	42
2.4 Elément de langage.....	43
3. Langage liste d'instruction (IL).....	48
3.1. Présentation.....	48
3.2. Instruction de base en langage liste.....	51
4. Le langage LOG (logigramme).....	52
4.1. Définition.....	52
4.2. Le but d'un logigramme.....	52
4.3. Symbole basique de logigramme processus.....	53
5. Langage texte structuré (ST).....	55
5.1 Définition.....	55
5.2 Syntaxe du ST.....	55

Chapitre III : Modélisation des trois systèmes industriels

1. Simatic Manager – STEP-7	57
1.1 Introduction	57
1.2. Définition de logiciel Simatic Step-7	57
1.3. Création d'un projet	58
1.4. Définition de matériel	59
1.5. Table des Mnémoniques	62
1.6. Avantages	63

1.7. Blocs du programme utilisateur	63
1.8. Structure d'un programme STEP 7	64
1.9. Programme cyclique OB 1	64
1.10. Les Fonctions et Blocs fonctionnels FC et FB.....	65
1.11. Simulation.....	65
2. L'équipement de travail au sein du laboratoire d'automatisation	67
3. Développement et programmation des systèmes industriels à l'aide de Simatic	
Step7	68
3.1. Système de remplissage contrôlé	67
3.2. Commande de deux portes de serrures	72
3.3. Système de réseau d'air comprimé	80
Conclusion	

Chapitre IV : Développement de l'interface avec le Wincc

Introduction	91
1. Le wincc explorer	91
1.1. SIMATIC HMI	91
1.2. Exposition du logiciel WinCC Explorer	91
1.3. Création d'un nouveau projet	92
1.4. Variables dans WinCC	93
1.5. Concepteur graphique.....	94
1.6. La liaison automate/IHM	95
2. Supervision avancée des systèmes industriels grâce à l'utilisation du WinCC.....	96
Explorer.....	96
2.1. Système de remplissage contrôlé.....	96
2.2. Commande de deux portes de serrures.....	98
2.3. Système de réseau d'air comprimé.....	100
Conclusion.....	106

Conclusion générale

Bibliographie

Liste des figures

Chapitre I : Etat de l'art des API

Figure 1-1 : Automate SIEMENS S7-300	6
Figure 1-2 : Architecture interne d'un API	8
Figure 1-3 : Place d'alimentation dans un API	9
Figure 1-4 : Place des modules entrées/sorties dans un API	10
Figure 1-5 : Place d'un CPU dans un API	11
Figure 1-6 : Des exemples de logiciels de programmation résidant sur l'ordinateur portable sont Studio 5000 pour les automates Allen Bradley ou SIMATIC Step 7 pour les automates Siemens	12
Figure 1-7 : La structure fonctionnelle d'un API	13
Figure 1-8 : Fonctionnement d'un API	14
Figure 1-9 : Temps de scrutation vs Temps de réponse	14
Figure 1-10 : Disposition des modules de l'API S7-300	18
Figure 1-11 : Exemple de CPU S7-300	18
Figure 1-12 : Câble MPI	19
Figure 1-13 : Alimentation PS 307	20
Figure 1-14 : SM 321 module d'entrée numérique	21
Figure 1-15 : SM 322 module de sortie numérique, DO 32	22
Figure 1-16 : SM 323 module d'entrées/sorties numérique, DI 8/DO 8	23
Figure 1-17 : SM 331 module d'entrée analogique, AI 8	24
Figure 1-18 : SM 332 module de sortie analogique, AO 4	25
Figure 1-19 : SM 334 module d'entrées/sorties analogique, AI 4, AO 2	25
Figure 1-20 : Simulateur module SM 374	26
Figure 1-21 : Connexion point à point CP 341.....	27
Figure 1-22 : console de programmation	28
Figure 1-23 : vue d'ensemble de l'automatisme	29

Chapitre II : Langages de programmation de API

Figure 2-1 : La symbolisation du GRAFCET	34
Figure 2-2 : Etape active (a) / Étape inactive (b)	35
Figure 2-3 : Etape initial	35
Figure 2-4 : Étape simple (a) / Étape a plusieurs actions (b).	36
Figure 2-5 : Réceptivité toujours vraie	37
Figure 2-6 : Réceptivité à niveau	37
Figure 2-7 : Liaison orienté	37
Figure 2-8 : Explication de la convergence et divergence dans le grafcet	40
Figure 2-9 : Explication de problème du même ordre dans le grafcet	41
Figure 2-10 : Explication d'un problème complexe du grafcet	41

Chapitre III : Modélisation des trois systèmes industriels

Figure 3-1 : création d'un projet dans step 7	58
Figure3-2 : définition du matériel sur le simatic	59
Figure 3-3, 4, 5, 6 : le catalogue des modules de l'automate s7-300.	62
Figure 3-7 : table de mnémonique.	62
Figure 3-8 : Liste des Blocs d'organisation OB disponible sur l'API S7-300.	65
Figure 3-9 : Outil de simulation STEP7 (PLCSIM).	66
Figure 3-10 : Equipement de travail dans le laboratoire	67
Figure 3-11 : Illustration de système de remplissage contrôlé	68
Figure 3-12 : Conception d'un système de remplissage contrôlé par le GRAFCET ...	71
Figure 3-13 : Tableau de mnémoniques du système de remplissage contrôlé	71
Figure 3-14 : programmation du système de remplissage contrôlé avec le langage ladder via step-7	72
Figure 3-15 : Illustration du système de commande de deux portes serrures	74
Figure 3-16 : Conception d'un système de commande de deux portes de serrures ...	76

Figure 3-17 : Tableau de mnémoniques du système de commande de deux portes serrures.....	77
Figure 3-18 : Programmation du système de commande de deux portes serrures ..	78-80
Figure 3-19 : Illustration du système du réseau d'air comprimé	81
Figure 3-20 : Conception d'un système de réseau d'air comprimé	84
Figure 3-21 : Tableau de mnémoniques du système de réseau d'air Comprimé	85
Figure 3-22 : Programmation du système de réseau d'air comprimé	86-89

Chapitre IV : Développement de l'interface avec le WinCC

Figure 4-1 : Assistant de Wincc explorer	92
Figure 4-2 : Donner les informations du projet de wincc explorer	93
Figure 4-3 : Fenêtre du travail dans le Wincc explorer	93
Figure 4-4 : Fenêtre du tag management dans le Wincc explorer	94
Figure 4-5 : Fenêtre du Graphics Designer dans le Wincc explorer	95
Figure 4-6 : liaison automate/IHM	95
Figure 4-7 : Système de remplissage déconnecté	96
Figure 4-8 : Système de remplissage connecté	97
Figure 4-9 : Les deux portes serrures se franchissent initialement	98
Figure 4-10 , 11 : Fermeture du porte 1 et ouverture de porte 2	98-99
Figure 4-12 : Etat initiale de système	100
Figure 4-13 : Démarrage du système	101
Figure 4-14 : Le pression arrive à 80%	102
Figure 4-15 : Chute a 60% de la pression maximale	103
Figure 4-16 : Chute à 50% de la pression maximale	104
Figure 4-17, 18 : La pression dépasse les normes (S19 / S20)	105

Liste des abréviations :

API: Automate Programmable Industriel.
EPROM: Erasable Programmable Read-Only Memory
EEPROM: electrically erasable programmable read-only memory
CPU: Central Processing Unit.
AC: Alternative current.
DC: Direct Current.
RAM: random access memory
ROM: read-only memory
E/S: Entrée/Sortie
PC : personal computer
TOR : Tout Ou Rien.
CEI : Commission Electrotechnique Internationale.
SCL : système control langage
MMC: Micro Memory Card
PS: Power Supply.
IM : Module d'interface
SM : Module de signaux
FM : Module de fonction
CP : Module de communication
SIL : Security integration level
LED : Signalisation d'état et de Défauts.
SD : Secure Digital
MPI : Interface Multi Points
DP : périphérie décentralisée
PtP : point à point
DI : Digital input
DO: Digital output
AI: Analog input
AO: Analog output
DM: Dummy module

GRAFCET : Graphe de Commande Etape Transition

ST : Text structuré

SFC : Séquentiel Fonction Chart

LD : Ladder Diagram

NO : Normalement ouvert

NF : Normalement fermé

IEC : Commission Electrotechnique Internationale

IL : Liste d'instructions

LOG : langage logigramme

DFD : les diagrammes de flux de données

UML : langage de modélisation unifié

TIA : totally integrated Automation

HW : Hardware

FB : Bloc fonctionnel

FC : Bloc Fonctions

SFB : Blocs fonctionnels système

DB : Bloc de données

OB : Bloc d'Organisation

LIST : Langage liste

CONT : Langage contact

PLCSIM : Programmable logic Controller Simulator

Résumé

Ce mémoire fin d'étude présente une étude approfondie sur la programmation par API des systèmes industriels, avec un accent particulier sur l'utilisation de l'automate Siemens S7-300. Le travail est divisé en quatre chapitres qui couvrent différents aspects de cette thématique.

Le premier chapitre se concentre sur la description détaillée de l'automate Siemens S7-300. Les caractéristiques, les composants et les fonctionnalités de cet automate sont exposés, permettant ainsi une compréhension approfondie de son architecture et de son fonctionnement.

Dans le deuxième chapitre, les langages de programmation associés à un API sont examinés en détail. Les différents langages disponibles, tels que le langage structuré (ST), le langage à contact (LAD) et le langage de liste d'instructions (IL), sont explorés afin de fournir aux lecteurs une connaissance approfondie des outils disponibles pour programmer efficacement les systèmes industriels.

Le troisième chapitre aborde la programmation des systèmes industriels en utilisant le logiciel Simatic Step 7. Ce logiciel offre une plateforme de développement puissante et conviviale pour la programmation et la configuration des automates Siemens S7-300. Les différentes étapes de programmation, de la création du projet à la configuration matérielle, sont décrites en détail.

Enfin, le quatrième chapitre se concentre sur la supervision des systèmes industriels à l'aide de l'interface graphique WinCC Explorer. Cet outil permet de créer des interfaces utilisateur conviviales pour surveiller et contrôler les systèmes industriels. Les fonctionnalités de WinCC Explorer, telles que la création d'écrans, la gestion des alarmes et la collecte de données, sont explorées afin de montrer comment créer une interface graphique efficace.

Ce mémoire offre une compréhension approfondie de la programmation par API des systèmes industriels, en mettant l'accent sur l'automate Siemens S7-300 et l'interface graphique WinCC Explorer. Il constitue une ressource précieuse pour les étudiants et les professionnels intéressés par le développement et la supervision des systèmes industriels.

Abstract

This final dissertation presents an in-depth study on API programming of industrial systems, with a particular focus on the use of the Siemens S7-300 controller. The work is divided into four chapters that cover different aspects of this topic.

The first chapter provides a detailed description of the Siemens S7-300 controller. The characteristics, components, and functionalities of this controller are explained, enabling a comprehensive understanding of its architecture and operation.

In the second chapter, the programming languages associated with an API are examined in detail. Different available languages, such as Structured Text (ST), Ladder Diagram (LAD), and Instruction List (IL), are explored to provide readers with an in-depth knowledge of the tools available for efficient programming of industrial systems.

The third chapter addresses the programming of industrial systems using the Simatic Step 7 software. This software provides a powerful and user-friendly development platform for programming and configuring Siemens S7-300 controllers. The various programming steps, from project creation to hardware configuration, are described in detail.

Lastly, the fourth chapter focuses on the supervision of industrial systems using the WinCC Explorer graphical interface. This tool allows the creation of user-friendly interfaces for monitoring and controlling industrial systems. WinCC Explorer features, such as screen creation, alarm management, and data collection, are explored to demonstrate how to create an effective graphical interface.

This final dissertation offers a comprehensive understanding of API programming of industrial systems, with a focus on the Siemens S7-300 controller and the WinCC Explorer graphical interface. It serves as a valuable resource for students and professionals interested in the development and supervision of industrial systems.

الموجز

يقدم هذا البحث دراسة مفصلة حول برمجة واجهة برمجة التطبيقات (API) للأنظمة الصناعية، مع التركيز الخاص على استخدام وحدة تحكم Siemens S7-300. ينقسم العمل إلى أربعة فصول تغطي جوانب مختلفة من هذا الموضوع.

يتركز الفصل الأول على شرح مفصل لوحدة التحكم Siemens S7-300. يتم شرح الخصائص والمكونات والوظائف المتاحة في هذه الوحدة، مما يتيح فهمًا شاملاً لهندستها وعملها.

في الفصل الثاني، يتم استعراض لغات البرمجة المرتبطة بواجهة برمجة التطبيقات بشكل مفصل. يتم استكشاف مختلف اللغات المتاحة، مثل النص المنظم (ST) والمخطط السلمي (LAD) وقائمة التعليمات (IL)، لتوفير معرفة عميقة للقراء حول الأدوات المتاحة لبرمجة الأنظمة الصناعية بكفاءة.

يتناول الفصل الثالث برمجة الأنظمة الصناعية باستخدام برنامج Simatic Step 7. يوفر هذا البرنامج منصة تطوير قوية وسهلة الاستخدام لبرمجة وتكوين وحدات التحكم Siemens S7-300. يتم شرح خطوات البرمجة المختلفة، من إنشاء المشروع إلى تكوين الأجهزة، بشكل مفصل.

أخيرًا، يركز الفصل الرابع على مراقبة الأنظمة الصناعية باستخدام واجهة WinCC Explorer الرسومية. يتيح هذا الأداة إنشاء واجهات مستخدم سهلة الاستخدام لمراقبة وتحكم الأنظمة الصناعية. يتم استكشاف ميزات واجهة WinCC Explorer، مثل إنشاء الشاشات وإدارة التنبيهات وجمع البيانات، لتوضيح كيفية إنشاء واجهة رسومية فعالة.

يقدم هذا البحث فهمًا شاملاً لبرمجة واجهة برمجة التطبيقات (API) للأنظمة الصناعية، مع التركيز على وحدة التحكم Siemens S7-300 وواجهة WinCC Explorer الرسومية. إنه مصدر قيم للطلاب والمهنيين المهتمين بتطوير ومراقبة الأنظمة الصناعية.

Introduction Générale :

Le recours à l'amélioration de la technique d'automatisation s'avère quelque chose d'inévitable, sachant les conséquences sur le plan industriel et recherche, c'est bien ce qui s'est produit, vers la fin des années 60, où on a assisté à un développement continu de l'industrie automobile, engendrant différents modèles de voitures, ceci a impliqué des transformations des chaînes de montage et de leurs armoires de commande, étant donné que ces dernières sont réalisées à base de relais, avec une complexité pouvant provoquer des risques élevés en cas d'erreurs de branchement. Enfin, l'idée qui s'est imposée est de créer un appareil programmable permettant de remplacer les armoires de commande, un dispositif appelé Automate Programmable Industriel. ^[1]

Un automatisme est un système qui, par le moyen de dispositifs mécaniques, pneumatiques, hydrauliques ou électriques, est capable de remplacer l'être humain pour certaines tâches simples ou complexes, les systèmes automatisés font partie de l'environnement de l'être humain. Ils se développent et prennent une place plus importante dans la manière de travailler, tant dans la production industrielle que dans les services.

Au XXI^e siècle, les appareils domestiques comportent au moins un automatisme. Dans l'industrie, ils sont indispensables pour effectuer les tâches les plus ingrates, répétitives ou dangereuses. Parfois, ces automatismes sont d'une telle rapidité et d'une telle précision qu'ils permettent de réaliser des actions impossibles pour un être humain. L'automatisme est donc aussi synonyme de productivité et de sécurité. Le développement des connaissances, et des outils mathématiques, ont conduit à un formidable essor des systèmes automatisés, et des systèmes asservis, dans la deuxième moitié du siècle. Certains se hasardent à rapprocher l'automatique et la philosophie, observant d'étranges similitudes entre les processus propres à l'homme et l'approche technologique. Mais au fait qu'est-ce qu'un système ? Bien difficile de répondre à une telle question ! Notre point de vue porte sur les systèmes de production et les systèmes pluri techniques en général, nous pouvons néanmoins en donner une définition plus large ^[1]

Système : Un système est une structure dont la fonction globale est d'ajouter de la valeur à un ensemble de matériaux de travail dans un contexte particulier. Qu'ils soient simples ou complexes, les systèmes automatisés se retrouvent partout dans notre environnement quotidien.

Ils continueront probablement d'évoluer et de jouer un rôle de plus en plus important dans notre façon de travailler, tant en atelier que dans les différents bureaux de l'entreprise. Connaître leur fonctionnement peut également nous permettre de mieux comprendre notre environnement.

Industrie : En première approximation, relèvent de l'industrie les activités économiques qui combinent des facteurs de production (installations, approvisionnements, travail, savoir) pour produire des biens matériels destinés au marché. ^{[1] [2]}

Une distinction est généralement établie entre l'industrie manufacturière et les industries d'extraction mais le contour précis de l'industrie dans chaque opération statistique est donné par la liste des items retenus de la nomenclature économique à laquelle cette opération se réfère (NAF, NES, NA, etc.).

Nous avons structuré notre travail de la manière suivante :

- **Le premier chapitre** est dédié à une description détaillée de l'automate Siemens S7-300.
- **Le deuxième chapitre** porte sur les langages de programmation utilisés dans une API.
- **Le troisième chapitre** se concentre sur la programmation des systèmes industriels à l'aide de Simatic Step 7.
- **Le quatrième chapitre** traite de la supervision des systèmes industriels à l'aide de l'outil WinCC Explorer.
- **Finalement**, nous concluons ce travail par une synthèse générale.

Chapitre I :

Etat de l'art des API

1. Généralité sur le domaine automatique et les API
 - 1.1. Généralité sur le domaine automatique et les API
 - 1.2. L'automatique
 - 1.3. Notion système
 - 1.4. Commande automatique
 - 1.5. Intérêt de la commande
2. Les automates programmables industriels (API)
 - 2.1 Historique
 - 2.2. Définition
 - 2.3. Place des automates programmables
 - 2.4. Structure générale des automates programmables
 - 2.5. Structure interne d'un automate programmable industriel API
 - 2.6. Mode de fonctionnement
 - 2.7. Choix d'un automate programmable industriel
 - 2.8. Les avantages des automates programmables industriels
 - 2.9. Les inconvénient des automates programmables industriels
3. L'automate SIEMENS S7-300
 - 3.1. Description
 - 3.2. Constitution de l'automate S7-300
 - 3.3. Avantage
 - 3.4. Caractéristique de l'automate S7-300
 - 3.5. Composants de station S7-300
 - 3.6. Modularité de S7-300
 - 3.7. Logiciel STEP-7

Conclusion

1. Généralité sur le domaine automatique et les API :

1.1 Automatisation et intervention humaine ^[3] :

L'opérateur est au cœur de la conception et de la conduite des systèmes d'automatisation industrielle.

Depuis l'avènement de l'automatisation et de l'informatique industrielle au milieu des années 60, les ingénieurs n'ont cessé de chercher à minimiser les divergences entre le modèle cognitif des tâches que l'homme souhaite réaliser et la compréhension de ces tâches par le système de contrôle-commande. Tout développement dans ce domaine débouche sur des systèmes plus performants, plus sûrs et plus fiables ainsi que sur la libération progressive de l'opérateur des tâches physiquement et mentalement contraignantes. L'homme dans la chaîne d'automatisation peut ainsi se consacrer à des activités plus stimulantes de supervision, gestion d'exceptions optimisation et maintenance. C'est ainsi qu'en 50 ans, une répartition claire des responsabilités s'est établie entre l'homme et la machine, exploitant au maximum les capacités de chacun. Or toute avancée dans cette direction est liée aux progrès continus dans trois domaines majeurs : aide à la décision ; ergonomie et technologie de visualisation de l'information utilisabilité des systèmes complexes.

La synthèse optimale de ces trois domaines créés le nec plus ultra des postes de conduite des systèmes modernes d'automatisation ^[3]

1.2. L'automatique :

L'automatique fait partie des sciences de l'ingénieur. Cette discipline traite de la modélisation, de l'analyse, de la commande et de la régulation des systèmes dynamiques. Elle a pour fondements théoriques les mathématiques, la théorie du signal et l'informatique théorique. L'automatique permet l'automatisation de tâches par des machines fonctionnant sans intervention humaine. On parle alors de système asservi ou régulé.

L'état désiré du système est nommé la consigne. Les hommes de l'art en automatique ou automatisme se nomment automaticiens. Un exemple simple, est celui du régulateur de vitesse d'une automobile, il permet de maintenir le véhicule à une vitesse constante, vitesse-consigne prédéterminée par le conducteur

1.3. Notion système :

Les systèmes sont représentés schématiquement par des blocs avec une ou plusieurs entrées et une ou plusieurs sorties. D'autres grandeurs qui ont des effets indésirables sont les perturbations (entrées erronées) :

- ✚ Système SISO : système à une entrée et une sortie, c'est un système mono variable ou SISO (Single Input Single Output)
- ✚ Système MIMO : système a plusieurs entrées et plusieurs sorties, c'est un système multi variable ou MIMO (Multiple Input Multiple Output)
- ✚ Système SIMO : système a une entrée et plusieurs sorties
- ✚ Système MISO : système a plusieurs entrées et une sortie ^[3]

1.4. Commande automatique :

Cela implique l'utilisation du dispositif approprié à la place de l'intervention humaine. Il devra utiliser les données qu'il recueille pour guider le processus afin de forcer la sortie à suivre une trajectoire prédéterminée.

1.5. Intérêt de la commande :

Gérez régulièrement les tâches ennuyeuses, répétitives ou fastidieuses. Contrôlez des systèmes complexes, erratiques ou rapides avec précision et en toute sécurité.

2. Les automates programmables industriels (API) :

2.1. Historique:

Les automates programmables industriels sont apparus à la fin des années soixante, à la demande de l'industrie automobile américaine (GM), qui réclamait plus d'adaptabilité de leurs systèmes de commande.

Les coûts de l'électronique permettant alors de remplacer avantageusement les technologies actuelles.

Avant : L'implémentation des parties de commande de la logique câblée repose sur l'utilisation de relais électromagnétiques et de systèmes pneumatiques.

Inconvénients : Cette approche, bien que précieuse, présente quelques restrictions intrinsèques, notamment une flexibilité limitée et l'absence de possibilité de communication.

Solution : L'adoption de systèmes basés sur des microprocesseurs, permettant une flexibilité accrue pour la modification des systèmes automatisés, conduit à la mise en œuvre d'une approche connue sous le nom de logique programmée.

Cette méthode se caractérise par l'utilisation de programmes informatiques pour contrôler et ajuster les opérations des systèmes automatisés de manière plus fluide. Grâce à cette transition vers la logique programmée, les restrictions inhérentes à l'utilisation de relais électromagnétiques et de systèmes pneumatiques peuvent être surmontées, ouvrant ainsi la voie à des fonctionnalités améliorées et à une communication plus efficace dans les systèmes automatisés.

Les ordinateurs de l'époque étant chers et non adaptés aux contraintes du monde industriel, les automates devaient permettre de répondre aux attentes de l'industrie. [3]

2.2. Définition:

L'Automate Programmable Industriel (API) est un appareil électronique programmable, adapté à l'environnement industriel, qui réalise des fonctions d'automatisme pour assurer la commande de pré actionneurs et d'actionneurs à partir d'informations logiques, analogiques ou numériques.

Un automate programmable industriel est un dispositif programmable dessiné à piloter des procédés ou parties opératives, et adaptable à un maximum d'applications, du point de vue traitement, composants, langages. Les API remplacent progressivement les équipements câblés, et grâce à leur structure, ils permettent de modifier simplement les séquences d'un processus, sans introduire les contraintes de câblage

Il est en général manipulé par un électricien. Le développement de l'industrie à entraîner une augmentation constante des fonctions électroniques présentes dans un automatisme c'est pour ça que l'API s'est substituée aux armoires à relais en raison de sa souplesse dans la mise en œuvre, mais aussi parce que dans les coûts de câblage et de maintenance devenaient trop élevés

La valeur du système peut être augmentée en ajoutant de nouveaux composants grâce à l'automatisation. Ces composants peuvent être décrites en termes d'objectifs par :

- ✚ Améliorer la flexibilité de production
- ✚ Améliorer la qualité du produit
- ✚ Adaptation à des contextes particuliers tels que les environnements hostiles pour l'homme (milieu toxique, dangereux. Nucléaire...),
- ✚ Adaptation à des tâches physiques ou intellectuelles pénibles pour l'homme (manipulation de lourdes charges, tâches répétitives parallélisées...),
- ✚ Augmenter la sécurité, etc...
- ✚ Accroître la productivité (rentabilité, compétitivité) du système [5]

2.3. Place des automates programmables:

Grâce à ses différents avantages, voir sa souplesse d'utilisation, son adaptation à tous les milieux industriels, ainsi que son encombrement réduit, l'API est actuellement le moyen le plus utilisé dans la commande des systèmes automatiques de production (SAP). Aujourd'hui, tous les secteurs industriels font recours aux automates [2], citons :

- ✚ L'industrie automobile
- ✚ Construction mécanique
- ✚ Transformation des matières plastiques



Figure 1-1 : Automate SIEMENS S7-300 [2]

2.4. Structure générale des API:

Les caractéristiques principales d'un automate programmable industriel (API) sont :

- ✚ Coffret, rack, baie ou cartes
- ✚ Compact ou modulaire
- ✚ Tension d'alimentation
- ✚ Taille mémoire
- ✚ Sauvegarde (EPROM, EEPROM, pile, ...)
- ✚ Nombre d'entrées / sorties
- ✚ Modules complémentaires (analogique, communication,)
- ✚ Langage de programmation ^[6]

2.5. Structure interne d'un automate programmable industriel API :

Un automate est principalement composé, du point de vue matériel, des éléments suivants :

- ✚ L'alimentation ainsi que les divers circuits annexes.
- ✚ Les entrées auxquelles sont accordés les capteurs, qui fournissent les informations relatives à l'équipement ou au procédé à commander ou à contrôler
- ✚ Les sorties qui émettent les ordres élaborés par l'unité de traitement vers les actionneurs (contacteurs, électrovannes...etc.).
- ✚ Une console de programmation, intégrée ou non à la machine, servant à introduire les instructions « utilisateur » en instructions exécutables par l'automate.
- ✚ L'unité arithmétique et logique, ou unité de traitement qui réalise les instructions logiques (ET, OU...) et arithmétiques (addition, comptage). ^[6]

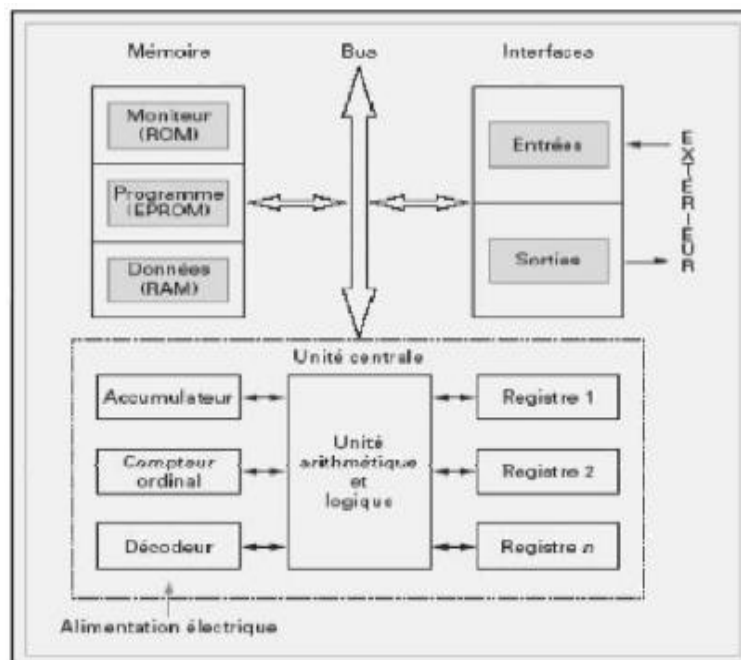


Figure 1-2 : Architecture interne d'un API [6]

- **Alimentation** : L'alimentation constitue un élément essentiel de l'automate, étant responsable de la fourniture de la tension électrique nécessaire à son fonctionnement. Elle est généralement connectée au secteur AC (courant alternatif) pour assurer une alimentation stable et fiable. L'alimentation convertit la tension secteur en une tension adaptée aux besoins spécifiques de l'automate, assurant ainsi son bon fonctionnement continu.

Permettant à l'automate de recevoir l'énergie électrique nécessaire à son fonctionnement, à partir d'une alimentation en 220V alternatif. Ce module délivre des tensions dont l'automate a besoin (5v, 12v, 24v) continu, et une mise à la terre doit être prévue. La mise sous tension de l'API est indiquée par un voyant positionné sur la façade.

La sortie de l'alimentation est une tension continue utilisée pour alimenter tous les

Autres modules associés à l'automate. L'alimentation N'alimente PAS les appareils de terrain.

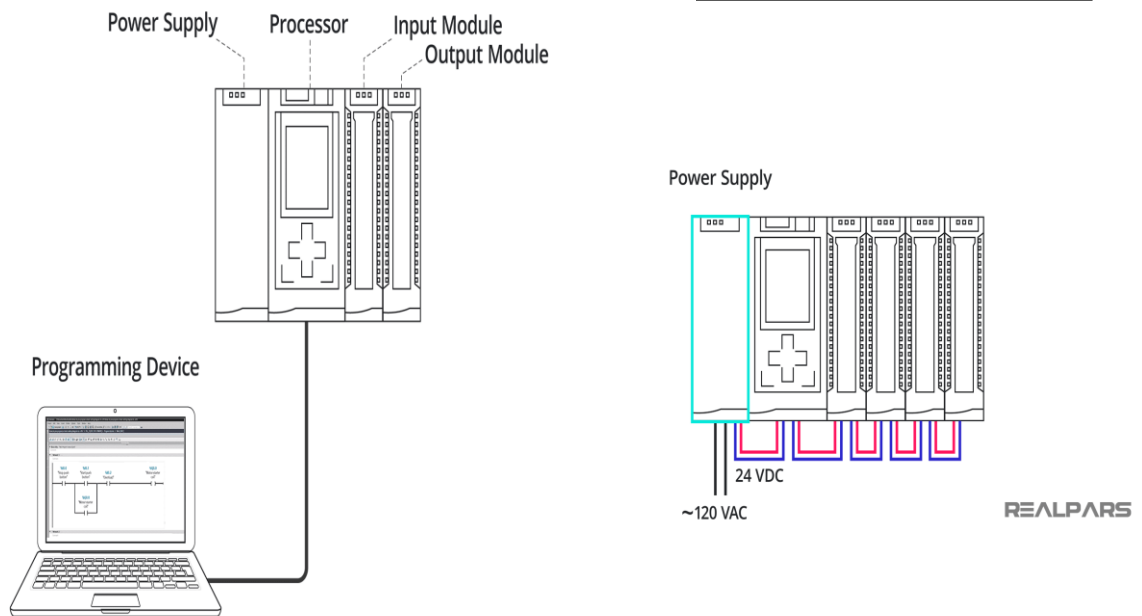


Figure 1-3 : Place d'alimentation dans un API [8]

- **Modules d'entrée/sortie** : Les modules d'entrées/sorties sont connectés à des appareils de terrain numériques ou analogiques. Les dispositifs de terrain d'entrée incluent les commutateurs, les encodeurs et les émetteurs, par exemple.

Les dispositifs de terrain de sortie typiques sont les relais, les lampes et les vannes proportionnelles.

Modules entrés : Ils Permettent à l'automate d'acquérir les informations provenant soit du pupitre de commande, ou bien des capteurs et détecteurs.

Des diodes électroluminescentes sur le module indiquent l'état de chaque entrée, et facilitent graduellement le contrôle du bon fonctionnement des dispositifs d'entrées et leur raccordement.

Modules sortis : Offrent à l'automate la possibilité de commander les divers pré-actionneurs (contacteurs, électrovanne, etc.), actionneurs (vérins, moteurs, etc.), et éléments de signalisation.

Les tensions de sorties sont de 5v en contenu ou de 24, 48, 110, 220 volts en contenu ou en alternatif.

Les sorties peuvent êtres logiques, analogiques, ou bien numériques

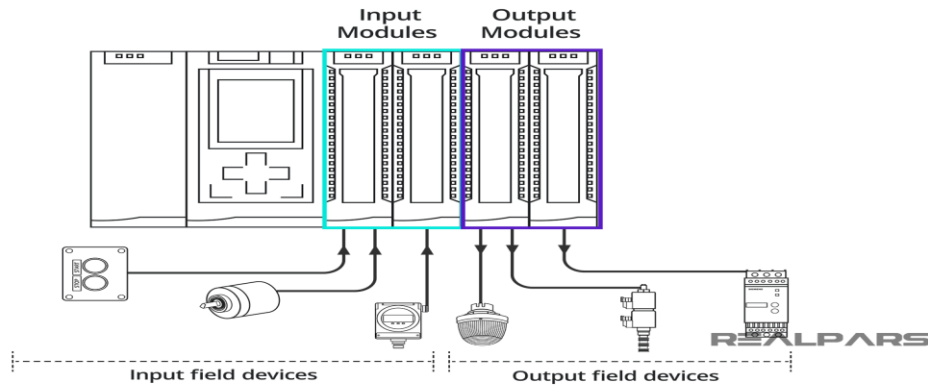


Figure 1-4 : Place des modules entrées/sorties dans un API

- **Processeur** : Le processeur se compose du CPU (unité centrale de traitement) et de la mémoire.

La section processeur prend les décisions nécessaires pour observer et faire fonctionner les dispositifs de terrain connectés aux modules d'entrée/sortie.

Les décisions sont basées sur un programme créé par l'utilisateur enregistré dans la mémoire. La mémoire stocke également des données représentant l'état de tous les dispositifs de terrain d'entrée et contient les données indiquant aux dispositifs de terrain de sortie quoi faire.

Il assure une bonne commutation entre la zone mémoire et les interfaces d'entrées et de sorties, et génère les instructions du programme.

Il est composé des éléments suivants :

- ✚ Unité logique (UL), qui traite les opérations logiques ET, OU...
- ✚ Unité arithmétique et logique (UAL) traitants les opérations de temporisation de comptage et de calcul
- ✚ Un accumulateur, représentant le registre de travail, dans lequel se range une donnée ou un résultat.
- ✚ Un registre d'instructions, qui contient durant le temps de traitement, l'instruction à exécuter.
- ✚ Un décodeur d'instructions, qui décode l'instruction à exécuter en y associant le microprogramme de traitement. ^[6]

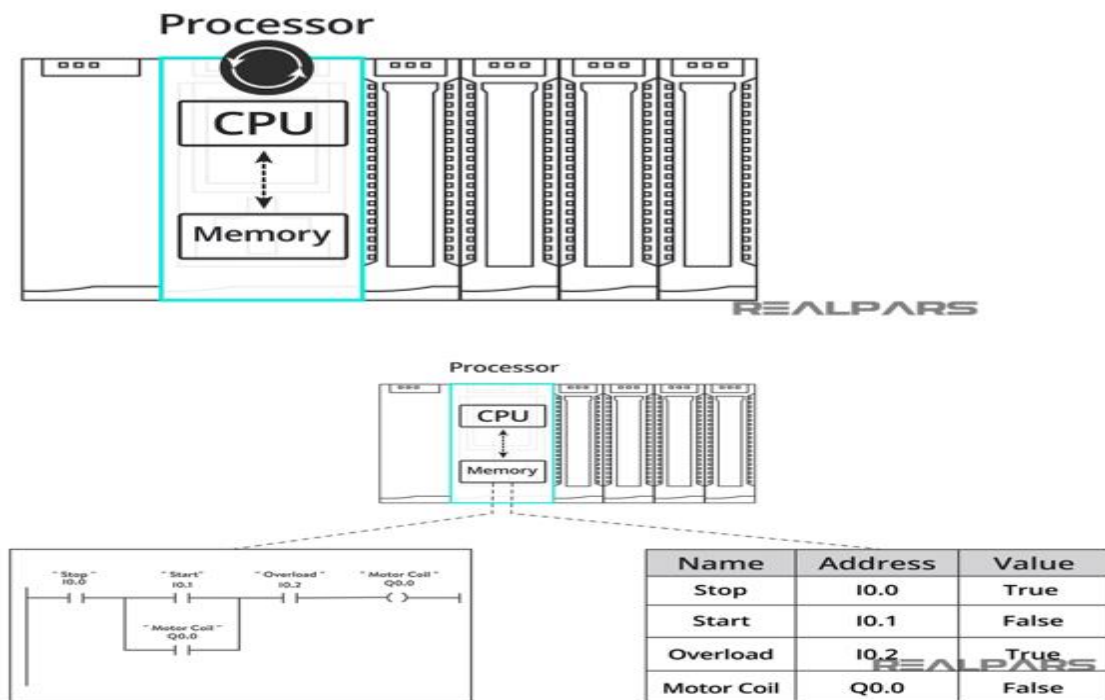


Figure 1-5 : Place de CPU dans un API [8]

- **Dispositif de programmation** : Le Dispositif de Programmation dans les applications industrielles d'aujourd'hui est généralement un ordinateur portable ou un ordinateur de bureau qui facilite la création de programmes décisionnels destinés à l'automate. Des exemples de logiciels de programmation résidant sur l'ordinateur portable sont Studio 5000 pour les automates Allen Bradley ou SIMATIC Step 7 pour les automates Siemens.

Le matériel PLC est un composant essentiel des systèmes d'automatisation industrielle modernes. L'unité centrale, les modules d'E/S, l'alimentation et les ports de communication fonctionnent ensemble pour contrôler des processus et des machines complexes.

Les progrès technologiques ont conduit à des processeurs plus rapides, à des modules d'E/S plus avancés et à des protocoles de communication améliorés, rendant les API plus puissants et polyvalents que jamais. [6]

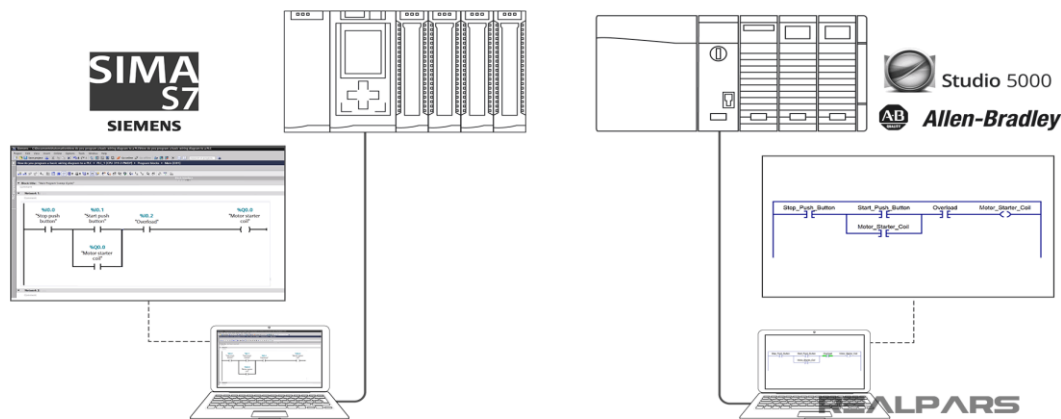


Figure 1-6 : Des exemples de logiciels de programmation résidant sur l'ordinateur portable sont Studio 5000 pour les automates Allen Bradley ou SIMATIC Step 7 pour les automates Siemens. [8]

2.6. Mode de fonctionnement :

L'API fonctionne suivant un programme, il reçoit les informations des dispositifs d'entrée, et prend des décisions sur les actions à exécuter (activer ou désactiver des sorties).

Principalement son rôle consiste à fournir à la partie opérative des instructions afin d'exécuter un travail précis :

Les 5 fonctions principales d'un automate programmable sont :

- ✚ Détection à partir de capteurs répartis sur la machine
- ✚ Commandes de mouvement des pré-actionneurs et actionneurs,
- ✚ Dialogue opérationnel, programmation et de supervision de la production.
- ✚ Ces fonctions utilisent différents modes de communication selon leur particularité.
 - ✚ Ou liaison « fil à fil » de modules d'entrées/sorties logiques (discrètes), par exemple,
 - ✚ Ou liaison « série » ou « parallèle » pour coupleur dédié et module de programmation. [6]

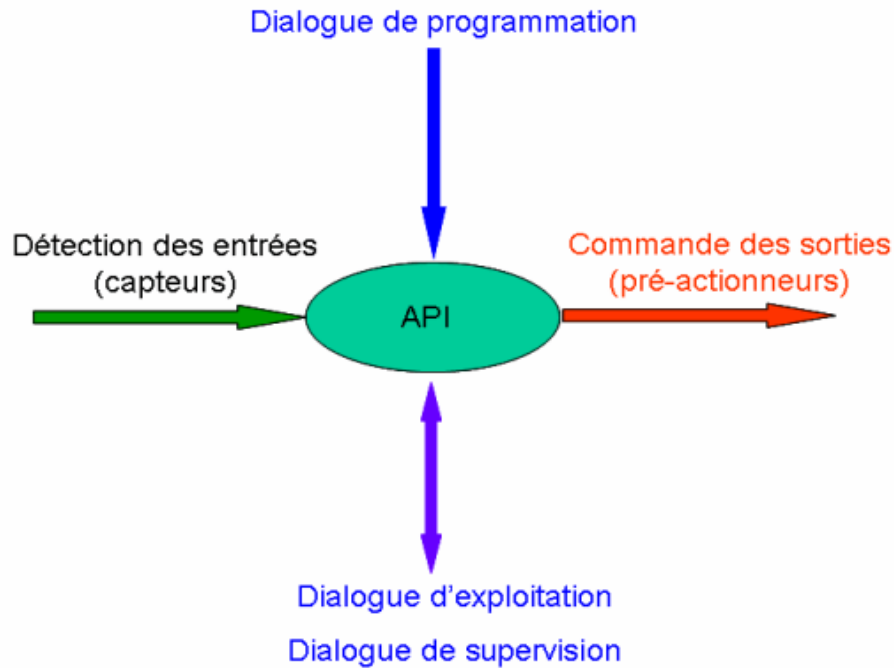


Figure 1-7 : La structure fonctionnelle d'un API ^[1]

L'automate programmable reçoit les informations relatives à l'état du système et puis **commande** les pré-actionneurs suivant le programme inscrit dans sa mémoire. Généralement les automates programmables industriels ont un fonctionnement cyclique. Le **microprocesseur** réalise toutes les fonctions logiques ET, OU, les fonctions de temporisation, de comptage, de calcul... Il est connecté aux autres éléments (mémoire et interface E/S) par des liaisons **parallèles** appelées ' **BUS** ' qui véhiculent les informations sous forme binaire... Lorsque le fonctionnement est dit synchrone par rapport aux entrées et aux sorties, le cycle de traitement commence par la prise en compte des entrées qui sont figées en mémoire pour tout le cycle.

Fonctionnement des automates

- Un automate exécute son programme de manière cyclique.
 - Lecture des entrées
 - Traitement du programme
 - Écriture des sorties
- Le temps d'exécution d'un cycle est contrôlé par une temporisation appelée **chien de garde**

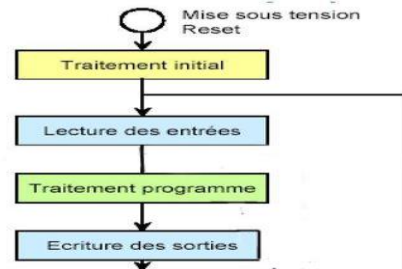


Figure 1-8 : Fonctionnement d'un API [1]

Le processeur exécute alors le programme instruction par instruction en rangeant à chaque fois les résultats en mémoire. En fin de cycle les sorties sont affectées d'un état binaire, par mise en communication avec les mémoires correspondantes. Dans ce cas, le temps de réponse à une variation d'état d'une entrée peut être compris entre un ou deux temps de cycle (durée moyenne d'un temps de cycle est de 5 à 15 ms).

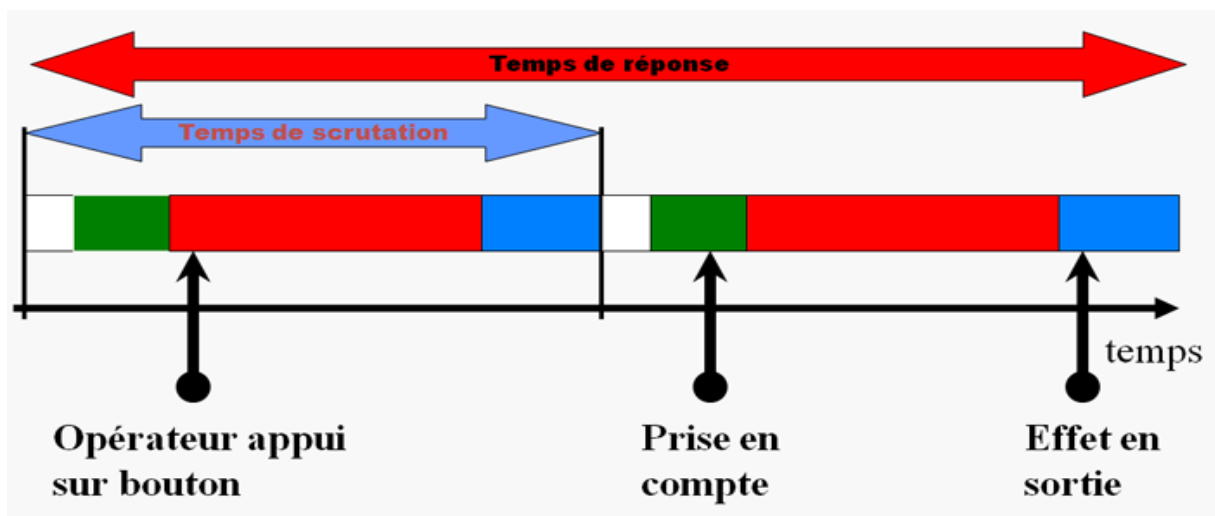


Figure 1-9 : Temps de scrutation vs Temps de réponse [6]

D'autres modes de fonctionnement moins typiques incluent :

- ✚ Uniquement en termes d'entrées.
- ✚ Asynchrone

2.7. Choix d'un API:

Afin de choisir un automate, on doit faire référence au cahier des charges et choisir un dispositif qui nous permettra de mieux suivre et gérer le procédé, et cela en considérons les éléments suivants :

Le matériel, le logiciel et les services annexes, ainsi que le nombre d'entrées et de sorties du procédé a été automatisé par l'API.

Suivant ces règles, les critères essentiels pour le choix d'un API sont :

- ✚ Les compétences de traitement du processeur (vitesse, opération...etc.)
- ✚ La qualité du service après-vente.
- ✚ Le type et le nombre d'entrées/sorties nécessaires. [2]

2.8. Les avantages des API :

- ✚ L'API est flexible, du fait de sa programmation, la modification de sa tâche est facile, contrairement au cas des systèmes de commande à relais réels, dont toute modification nécessite l'ajout ou le retrait de relais, ainsi que la modification des raccordements ; et cette opération peut engendrer des erreurs de branchements qui peuvent provoquer des risques élevés
- ✚ Il est moins encombrant que l'armoire de commande à relais, une unité centrale de traitement d'environ 0,1 mètre cube peut remplacer des centaines de relais de commande, ainsi que tout le câblage qui relie leur contact.
- ✚ Ne provoque pas de bruit (silencieux), et consomme beaucoup moins d'énergie.
- ✚ Le fait qu'il ne dispose pas de pièces mobiles, lui offre plus de fiabilité, or que la commande à relais peut voir des commandes erronées, causées par l'usage des pièces en mouvement, tel que l'oxydation et le soudage des contacts des relais [2]

2.9. Les inconvénients des API :

- ✚ L'API ne supprime pas tout le câblage, il reste celui de puissance.
- ✚ Sa vitesse d'exécution peut être insuffisante dans certains cas.
- ✚ L'ordre dans lequel on écrit le programme peut perturber le comportement de la commande, causé par des problèmes de type aléas de séquence.
- ✚ Une même fonction n'a pas nécessairement le même effet, d'un API à l'autre. [2]

3. L'automate SIEMENS S7-300 :

3.1. Description :

Le SIMATIC S7-300 est utilisé dans de nombreuses applications dans le monde entier et a fait ses preuves des millions de fois. Les contrôleurs universels SIMATIC S7-300 permettent d'économiser de l'espace d'installation et se caractérisent par une conception modulaire. Une large gamme de modules permet d'étendre le système de manière centralisée ou de créer des structures décentralisées en fonction de la tâche à accomplir, et facilite un stockage économique des pièces de rechange. SIMATIC est connu pour sa continuité et sa qualité.

Le S7-300 est l'automate conçu pour des solutions dédiées au système manufacturier et constitue à ce titre une plate-forme d'automatisation universelle pour les applications avec des architectures centralisées et décentralisées [7]

3.2. Constitution de l'automate S7-300 :

L'automate S7-300 possède :

- ✚ Des CPU de différents niveaux de performances
- ✚ Des Modules de signaux pour Entrées/Sorties « TOR » et analogique, ainsi que des Modules de fonction pour les différentes fonctions technologiques
- ✚ Une possibilité de mise à niveau par MPI
- ✚ Une largeur réduite des Modules, permettant un gain de place au montage
- ✚ Une structure compacte, lui permettant le placement aux milieux exigus. [7]

3.3. Avantage :

- ✚ Une construction compacte et modulaire, libre de contraintes de configuration.
- ✚ Une riche gamme de modules adaptés à tous les besoins du marché est utilisable en architecture centralisée ou décentralisée, qui réduit grandement le stock de pièces de rechange.
- ✚ Une large gamme de CPU adaptée à toutes les demandes de performances pour pouvoir obtenir des temps de cycle machines courts, certaines étant dotées de fonctions technologiques intégrées comme par ex. le comptage, la régulation ou le positionnement. [7]

- ✚ Une économie d'ingénierie en utilisant les outils orientés application et normalisés CEI 1131- 3 tels que les langages évolués SCL ou des logiciels exécutifs orientés technologie pour le contrôle des mouvements

3.4. Caractéristique de l'automate S7-300 :

- ✚ L'automate S7-300 est spécifié par les caractéristiques suivantes :
- ✚ Gamme diversifiée de la CPU.
- ✚ Gamme complète du module.
- ✚ Possibilité d'exécution jusqu'à 32 modules.
- ✚ Bus de fond de panier intégré en module.
- ✚ Possibilité de mise en réseau avec MPI, PROFIBUS ou INDUSTRIAL ETHERNET.
- ✚ Raccordement central de la PG avec accès à tous les modules.
- ✚ Liberté de montage au différent emplacement
- ✚ Configuration et paramétrage à l'aide de l'outil configuration matériels.
- ✚ Plusieurs automates S7-300 peuvent communiquer entre eux aux moyens d'un câble-bus PROFIBUS pour une configuration décentralisée. [7]

3.5. Composants de station S7-300 :

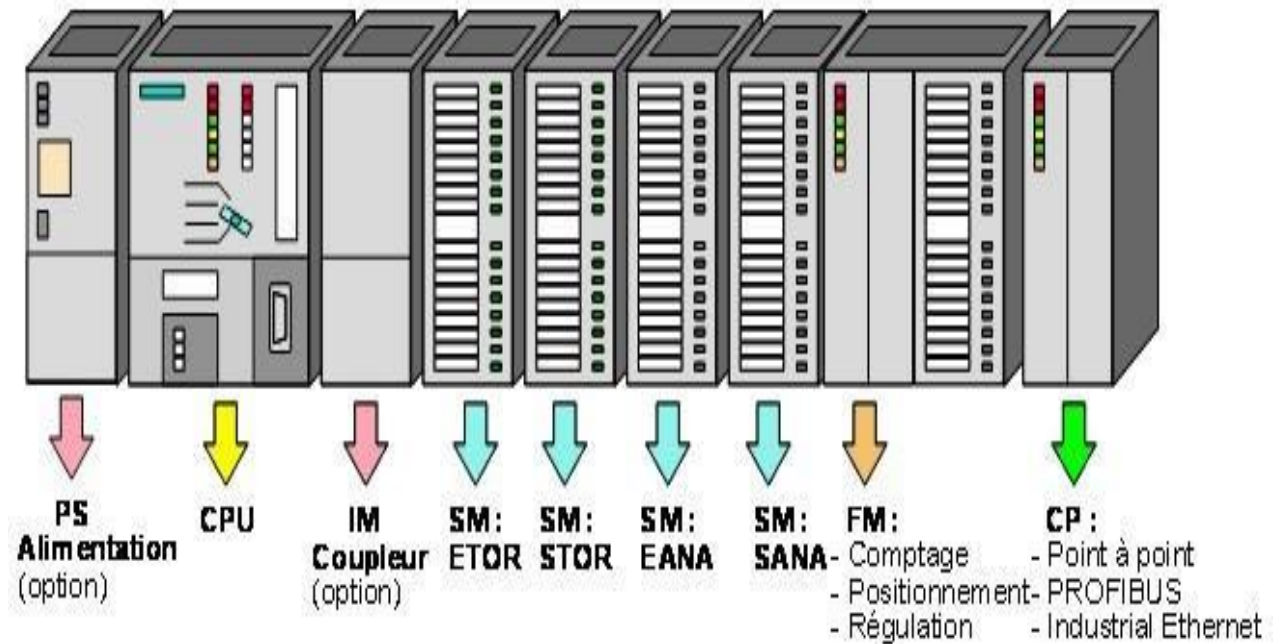
Un automate programmable comprenant tous les modules d'E/S est appelé « station ». Une station S7-300 peut contenir les composants suivants :

- Rack
 - Alimentation (PS)
 - Unité centrale de traitement (CPU)
 - Module d'interface (IM)
 - Module d'entrée/sortie (module de signal SM)
 - Module de fonction (FM)
 - Module de communication (processeurs de communication CP)
 - Module spécial tel que le module simulateur
- Une station peut également comprendre des E/S distribuées qui lui sont connectées via un système de bus.

- Certains modules SIMATIC S7-300 sont également disponibles en version SIPLUS pour des conditions environnementales difficiles. [7].

3.6. Modularité de S7-300 :

Parmi les caractéristiques essentielles du S7-300, le fait qu'il est disposé d'une vaste gamme



de Modules.

Figure 1-10 : Disposition des modules de l'API S7-300 [3]

3.6.1. Unité central CPU :

On a le choix entre plusieurs CPU, avec différentes performances, la CPU 314 représente des grandes performances.



Figure 1-11 : Exemple de CPU S7-300 [7]

3.6.2. Interface MPI – Message Passing Interface ^[10] :

Afin d'assurer une connexion de la console de programmation, ou un adaptateur PC par exemple, la CPU est équipée d'une interface MPI.

Le câble relie la PG (console de programmation) ou le PC à la CPU

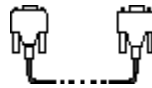


Figure 1-12 : Câble MPI

3.6.3. Signalisation des états :

Sur la face avant de la CPU, on trouve des LEDS permettant à l'automate de signaler certains états, tel que par exemple RUN :

- Etat de mise en fonctionnement
- Clignotement à la mise en route de la CPU ^[11]

3.6.4. Commutateur de mode de fonctionnement :

Il assure le changement du mode de fonctionnement, chaque position choisie autorise certaines fonctions bien particulières à la console de programmation, tel que RUN, STOP...etc.

3.6.4.1. Carte Mémoire :

La CPU peut être équipée d'une carte mémoire, permettant la sauvegarde du programme en cas de coupure du courant, même en absence de pile.

3.6.4.2. La pile :

Le contenu de la RAM peut être sauvegardé même en absence de courant, via la pile.

3.6.4.3. Modules d'alimentation :

Le S7-300 peut être alimenté avec une tension de 24vcc, cette dernière est assurée via le module d'alimentation par conversion de la tension Secteur 380/220 vac

Les capteurs, actionneurs et voyants lumineux qui demandent plus de 24 v, sont alimentés par des blocs d'alimentation ou transformateurs supplémentaires.



Figure 1-13 : Alimentation PS 307 [11]

3.6.5. Modules de signaux (SM) :

Assurant une liaison entre la CPU de l'automate S7-300, et le processus à commander. On a différents modules de signaux

3.6.6. Modules d'entrée /sortie TOR :

Permettant à l'automate de recevoir des informations sur l'état des différents capteurs, boutons poussoirs...etc.

Les têtes de stations des ET200S sont de véritables automates programmables de la gamme SIMATIC S7 donc programmées avec l'atelier logiciel STEP 7

Caractéristique technique :

- Pré câblage intégré (en IP20)
- Aucun câblage grâce à la connectique entièrement enfichable (en IP67)
- CPU 314 du S7-300

- Supporte la vitesse de 12 Mbit/s sur le réseau de terrain standard normalisé PROFIBUS-DP
- Esclave PROFIBUS-DP
- Fonctions de sécurité machine intégrées avec IM 151-7 F CPU

3.6.7. Module Entré Numérique:

Les modules d'entrées numériques sont utilisés par la CPU pour enregistrer

Les états de fonctionnement de la machine commandée ou plantée. Ces modules sont des conditionneurs de signal pour binaire traiter les signaux d'entrée. Les signaux de processus présents avec un niveau de tension continuent de 24 V à 120 V ou une tension alternative de 120 V à 230 V sont convertis en signaux avec un niveau interne.

Selon le module, les voies d'entrée sont isolées soit individuellement soit en groupe. Le module les types incluent des modules d'entrée simples, des modules avec diagnostic capacité et modules pour l'isochronisme mode. Tous les modules indiquent la présence d'un processus signal au moyen d'une LED sur le canal d'entrée. [9]



Figure 1-14 : SM 321 module d'entrée numérique [11]

3.6.8. Module sortie numérique:

Les modules de sortie numérique sont utilisés par la CPU pour contrôler La machine ou l'installation connectée. Ces modules sont des conditionneurs de signal pour les signaux de sortie de processus binaires. Les signaux internes sont amplifiés et émis en les plages de courant et de tension suivantes (valeurs nominales) :

- Avec amplificateurs électroniques de 24 V à 120 V DC et un courant de 0,3 A à 2 A
- Avec amplificateurs électroniques de 120 V à 230 V AC et un courant de 1 A ou 2 A
- Avec des contacts de relais avec une tension continue de 24 V ou un Tension alternative de 230 V et un courant jusqu'à 5 A ^[8].



Figure 1-15 : SM 322 module de sortie numérique, DO 32 [11]

3.6.9. Module entrée/sortie numérique :

Les modules d'entrées/sorties numériques sont utilisés par la CPU pour enregistrer les états de fonctionnement de la machine connectée ou de l'installation et de contrôler la machine ou l'installation. Ces modules sont des conditionneurs de signal pour le processus binaire signaux d'entrée et de sortie.

Signaux de processus présents avec un niveau de tension continue de 24V.

Sont convertis en signaux d'entrée avec un niveau interne. Les signaux de sortie internes sont amplifiés électroniquement et sortie à 24 V avec un courant jusqu'à 0,5A.

Selon le module, les voies sont isolées les unes des autres. Les sorties ont un court-circuit intégré protégé. Tous les modules indiquent un présent ou signal de processus délivré au moyen d'une LED sur l'entrée ou canal de sortie. ^[8].



Figure 1-16 : SM 323 module d'entrées/sorties numérique, DI 8/DO 8 [11]

3.6.10. Modules analogiques :

Si on à faire à des capteurs et actionneurs analogiques, on doit faire appel à des modules d'entrées analogiques, assurant la conversion des signaux analogiques du système à commander en signaux numériques, qui peuvent être traités par l'API S7-300.

Toutefois la conversion des signaux numériques en signaux analogiques destinés au processus à commander est faite, via les modules de sorties.

Les modules d'Entrées/Sorties analogiques réalisent les deux fonctions, c'est-à-dire la conversion Analogique-Numérique, et Numérique-Analogique. [8].

3.6.10.1. Module d'entrée analogique :

Le CPU peut utiliser des modules d'entrées analogiques pour traiter les variables mesurées après leur conversion en valeurs numériques par les modules. Ces modules sont conditionneurs de signaux pour signaux d'entrée de processus analogiques.

Transmetteurs de tension et de courant, thermocouples, résistances ou thermorésistances peuvent être connectées aux modules en fonction de la conception. La plage de mesure peut être réglée librement par canal ou par groupe de canaux. La valeur numérique est générée par intégration, ou avec le SM 331-7FH01 par conversion immédiate en 52 s'par voie. En fonction, dépendamment sur le module, la résolution est jusqu'à 16 bits signe compris

Une valeur analogique (une voie analogique) occupe 16 bits, soit deux octets. Analogique les modules d'entrée sont disponibles avec 2, 4 ou 8 voies, correspondant à une adresse plage de 4, 8 ou 16 octets. [8].



Figure 1-17 : SM 331 module d'entrée analogique, AI 8 [11]

3.6.10.2. Modules sortie analogique :

Le CPU peut utiliser des modules de sortie analogique pour fournir aux actionneurs des consignes analogiques. Ces modules sont des conditionneurs de signal pour les signaux de la sortie de processus analogique.

Les modules peuvent sortir une valeur de tension (0 à 10 V, 1 à 5 V, ou -10 à +10 V) ou une valeur de courant (0 à 20 mA, -20 à +20 mA, ou 4 à 20 mA). Selon les modules, la résolution est jusqu'à 16 bits, y compris le signe.

Une valeur analogique (une voie analogique) occupe 16 bits, soit deux octets. Analogique les modules de sortie sont disponibles avec 2, 4 ou 8 voies correspondant à une adresse plage de 4, 8 ou 16 octets. [8].



Figure 1-18 : SM 332 module de sortie analogique, AO 4 [11]

3.6.10.3. Modules entrée/sortie analogique :

Le CPU peut utiliser des modules d'entrées/sorties analogiques pour traiter des variables analogiques et fournir aux actionneurs des consignes analogiques. Ces modules sont des conditionneurs de signaux pour les signaux de processus analogiques.

Une valeur de tension (0 à 10 V), une valeur de courant (0 à 20 mA), ou - avec SM 334-0KE00 - également une thermo résistance Pt 100 peut être connectée à un canal d'entrée du SM 334 modules. Un canal de sortie peut sortir une valeur de tension de 0 à 10 V, avec SM 334-0CE01 aussi un courant de 0 à 20 mA. La résolution est de 8 ou 12 bits, signe compris. Dans le SM 334-0KE00, les voies de sortie sont isolées du bus de fond de panier ^[8].



Figure 1-19 : SM 334 module d'entrées/sorties analogique, AI 4, AO 2

3.6.11. Modules de simulation :

Pour valider le programme, et vérifier le bon fonctionnement de notre automatisme, on doit procéder à la simulation, ce que nous permettrons de tester le programme.

Au niveau du S7-300, le module de simulation offre différentes fonctions :

- La signalisation d'état des signaux de sorties par des LED
- La simulation des signaux des capteurs à base d'interrupteur



Figure 1-20 : Simulateur module SM 374 [11]

3.6.12. Modules de communications :

Assurant une communication par transmission série, comme ils peuvent aussi établissent des liaisons point à point avec d'autres automates SIMATIC S7-300, SIMATIC S5 ou bien d'autres constructeurs

Les tâches impliquant la communication sont déchargées de la CPU par des modules de communication (également appelés processeurs de communication ou CP). Ils établissent la liaison physique avec un partenaire de communication, prennent en charge l'établissement de

la liaison et le transport des données via celui-ci et offrent des services de communication à la CPU et au programme utilisateur

Les modules de communication suivants sont disponibles :

- ✚ CP 340
- ✚ CP 341
- ✚ CP 343-2
- ✚ CP 342-5
- ✚ CP 343-5
- ✚ CP 343-1



Figure 1-21 : Connexion point à point CP 341 [11]

3.6.13. Console de programmation :

Permet la saisie, le traitement et la sauvegarde des données machines, et les données concernant le déroulement du processus.



Figure 1-22 : console de programmation

3.7. Logiciel step 7 :

3.7.4. Introduction :

Les automates programmables effectuent des tâches d'automatisation traduites sous forme de programme d'applications. L'utilisateur définit la manière dont l'automate doit commander l'installation par une suite d'instructions. Le programme doit être écrit dans un langage déterminé avec des règles définies pour que l'automate puisse l'exécuter, pour cela le logiciel le plus performant à l'exécution de ces tâches est le STEP7. ^[13]

3.7.5. Définition du STEP7 :

LE STEP 7 est un logiciel de base pour à configuration et la programmation des systèmes d'automatisation. Il fait partie de l'industrie logicielle SIMATIC, il existe en plusieurs versions telles que : STEP micro/Dos et STEP 7 micro/Win pour les applications S7-300 et S7-400 ^[13]

3.7.6. Programmation de l'automate S7-300 :

La programmation des automates de la famille S7 se fait par la console de programmation ou par PC et sous un environnement WINDOWS. Via le langage de programmation STEP7.

Le logiciel STEP 7 offre les fonctions suivantes pour l'automatisation :

- ✚ Configuration et paramétrage du matériel et de la communication.
- ✚ La création et la gestion des projets
- ✚ La gestion des mnémoniques.
- ✚ La création des programmes pour des systèmes cibles S7.
- ✚ Le chargement des programmes dans les systèmes cibles.
- ✚ Le teste de l'installation d'automatisation
- ✚ Possibilité d'extension grâce aux applications proposées par l'industrie logicielle SIMATIC [14]

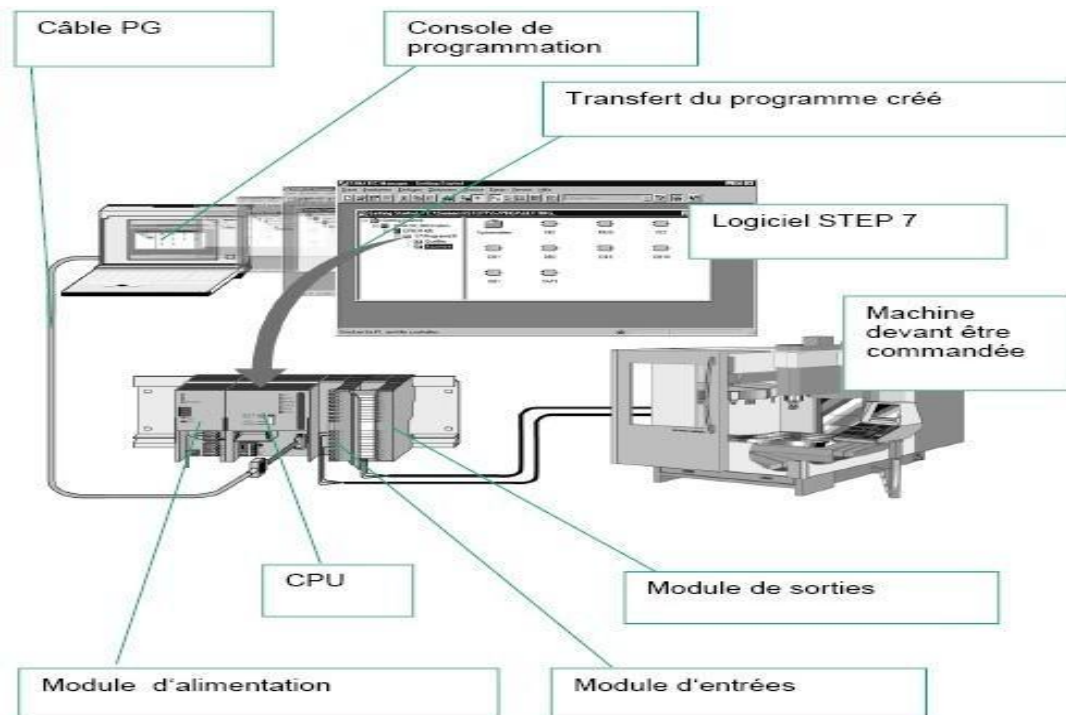


Figure 1-23 : vue d'ensemble de l'automatisme. [13]

La programmation en STEP 7 présente trois modes de représentation qui peut être combinés dans une même application :

- Le schéma logique (LOG)
- Le schéma contact (CONT)
- Liste d'instruction (LIST)

Chaque mode de représentation du programme a ses avantages mais aussi ses limites. Si les règles de programmation ont été respectées lors de la programmation, la compilation est possible dans les trois modes de représentation. Les programmes en CONT et en LOG sont en principe toujours traduisibles en LIST.

Dans la mémoire de programmation de l'automate, le programme est toujours stocké en LIST (plus exactement en langage machine).

3.7.7. Adressage des modules de S7-300 :

Il existe deux modalités d'adressage distinctes :

- Adressage lié à l'emplacement : il s'agit du mode d'adressage par défaut, c'est-à-dire que le step7 effectue à chaque numéro d'emplacement une adresse de défaut de modèle fixé à l'avance.
- Adressage libre : dans ce mode d'adressage, il faut effectuer à chaque mode une adresse de votre choix, pourvu qu'elle soit contenue dans la plage d'adresses possible de la CPU. **[18]**

❖ Conclusion ;

En conclusion de ce chapitre, nous pouvons affirmer que le contrôleur logique programmable (API) représente un outil adéquat pour les solutions d'automatisation.

Dans le contexte des processus industriels, cet outil se distingue comme le plus couramment utilisé.

En particulier, le logiciel S7-300 se révèle essentiel pour la simulation de programmes et de concepts de commande automatisés, offrant ainsi une étape préliminaire cruciale avant leur implémentation dans un système réel.

Cette approche permet d'optimiser les performances et la fiabilité des systèmes automatisés, tout en réduisant les risques et les coûts associés aux erreurs de mise en œuvre.

En conséquence, l'utilisation du logiciel S7-300 contribue à renforcer l'efficacité et la sécurité des processus industriels.

Chapitre II :

Langage des programmation des API

Introduction

Différent langage de programmation d'un API

1. Le GRAFCET

1.1. Introduction au GRAFCET

1.2. Historique

1.3. Définition

1.4. Cahier de Charge

1.5. Les éléments de base du GRAFCET

1.6. Règles d'évolution du GRAFCET

1.7. Configuration courante

2. Langage a contact (LADDER

2.1 Définition

2.2. Objectif

2.3. Principe

2.4 Elément de langage

3. Langage liste d'instruction (IL)

3.1. Présentation

3.2. Instruction de base en langage liste

4. Le langage LOG (logigramme)

4.1. Définition

4.2. Le but d'un logigramme

4.3. Symbole basique de logigramme processus

5. Langage texte structuré (ST)

5.1 Définition

5.2 Syntaxe du ST

Conclusion

Introduction :

Sur le marché, divers types de logiciels PLC sont disponibles, chacun présentant des caractéristiques et des capacités uniques. Parmi les types les plus répandus, on trouve le logiciel de programmation logique à relais, le logiciel de diagramme de blocs fonctionnels, le logiciel de programmation de texte structuré et le logiciel de diagramme de fonctions séquentielles.






Le logiciel de programmation logique à relais se distingue comme le plus largement utilisé, offrant une interface graphique évoquant une échelle pour la création et la modification de programmes. Le logiciel de diagramme de blocs fonctionnels, quant à lui, adopte une approche similaire, mais utilise des blocs plutôt que des échelons sur une échelle. Enfin, le logiciel de programmation de texte structuré emploie un langage textuel similaire à C ou Pascal.

Cette diversité de logiciels PLC permet aux utilisateurs de choisir celui qui correspond le mieux à leurs besoins spécifiques en matière de programmation et de manipulation des systèmes automatisés. Chaque type de logiciel offre des avantages particuliers en termes de convivialité, de flexibilité et de compatibilité avec d'autres systèmes, garantissant ainsi une mise en œuvre efficace et une gestion optimale des processus industriels.

Différent langage de programmation d'un API :

L'écriture d'un programme consiste à créer une liste d'instructions permettant l'exécution des opérations nécessaires au fonctionnement du système.

Pour la programmation des API on a plusieurs types de langage parmi lesquels :

-  Langage de programmation à partir d'un GRAFCET.
-  Langage à contact (Ladder)
-  Langage de programmation liste d'instruction.
-  Le langage booléen (Logigramme).
-  Le langage texte structuré (ST).

1. Le GRAFCET :

1.1. Introduction au GRAFCET :

Pour être analysé, communiqué et compris, le fonctionnement d'un système automatisé doit être modélisé. Selon les moments de la vie du système les modèles sont différents pour répondre à des besoins précis et variés (dessins mécanique, algorithmes, etc...).

Bien que l'utilisateur, qui doit comprendre ou modifier les modèles initiaux à des fins de conduite ET/OU de maintenance, n'ait pas le même objectif que le concepteur, il ne peut en règle générale, découvrir et appréhender le fonctionnement du système automatisé qu'à partir de documents réalisés en phase de conception, il est donc essentiel que les méthodes et outils utilisés permettant de véhiculer les informations élaborées au long du cycle de la vie du système.

Le fonctionnement d'un système automatisé peut être représenté par un ensemble appelé GRAFCET. [15]

1.2. Historique :

Le GRAFCET est né en 1977 des travaux de l'AFCEC (Association Française pour la Cybernétique Économique et Technique), en tant que synthèse théorique des différents outils existant à cette époque (organigramme, diagramme de Gérard, etc...). Mais sous sa forme graphique actuelle l'ADEPA (Agence pour le Développement de la Production Appliqué à l'industrie) en 1979 normalisé sur le plan Français en 1981 (norme NF CO3-190), il est aujourd'hui normalisé sur le plan national (norme CEI 848) [15]

1.3. Définition :

Le GRAFCET, également connu sous le nom de Graphe de Commande Etape Transition, se présente comme un schéma fonctionnel visant à représenter de manière graphique et conforme aux exigences spécifiées, les divers comportements et la progression d'un système automatisé séquentiel.

Son utilisation se distingue par sa simplicité et sa convivialité, en fournissant un outil de communication essentiel pour toutes les parties prenantes impliquées dans la conception, l'utilisation et la maintenance de la machine à automatiser.

Le GRAFCET assure une représentation visuelle précise et cohérente, permettant ainsi une compréhension commune des séquences d'actions, des états et des transitions de l'automatisme. Grâce à son approche méthodique et à sa rigueur formelle, il facilite la planification, l'optimisation et la surveillance des processus automatisés, contribuant ainsi à l'efficacité et à la fiabilité des systèmes industriels. [16]

I.4. Cahier de charge :

Le cahier des charges d'un automatisme est une description fonctionnelle faite par l'utilisateur, elle analyse le comportement de la partie commande vis-à-vis de la partie opérative permettant au concepteur de comprendre ce que l'automatisme doit faire aux différentes situations pouvant se présenter. [16]

1.5. Les éléments de base du GRAFCET :

Le GRAFCET se structure autour d'éléments fondamentaux tels que les étapes, les transitions et les liaisons.

La figure illustre les différents symboles qui caractérisent le GRAFCET

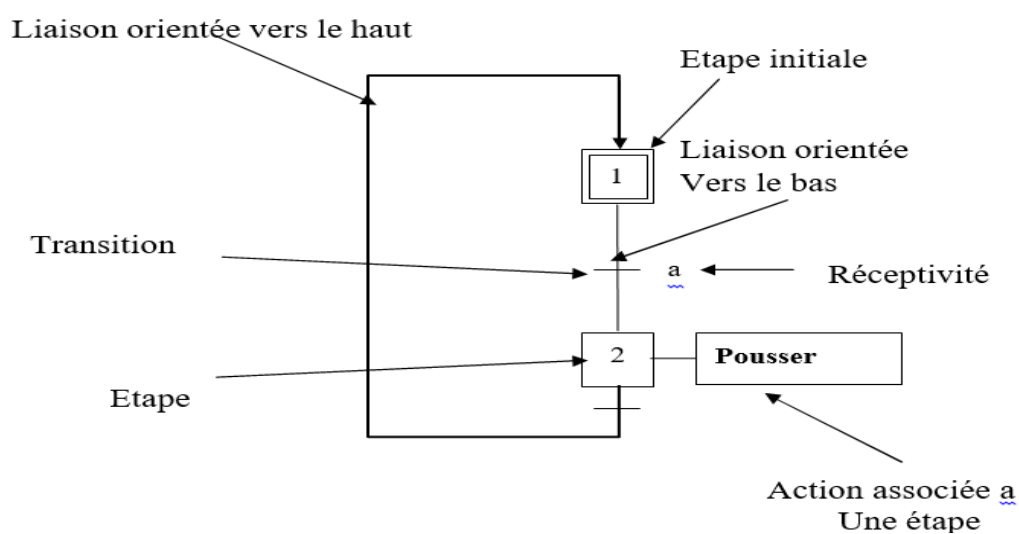


Figure 2-1 : La symbolisation du GRAFCET [15]

1.5.1. Les étapes :

Une étape correspond à une phase durant laquelle on effectue une action pendant une certaine durée. L'action doit être stable, c'est-à-dire que l'on fait la même chose pendant toute la durée de l'étape.

Lorsque l'étape est active, on le précise en y ajoutant un point.

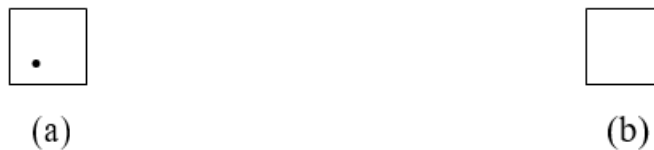


Figure II-2 : Etape active (a) / Étape inactive (b) [15]

1.5.1.1. Etape initiale :

L'étape initiale représente l'étape active du système au début du fonctionnement. Elle se différencie en doublant les côtés du carré



Figure 2-2 : Etape initial

1.5.1.2. Les actions associées à une étape :

A chaque étape sont associées une ou plusieurs actions. Ces dernières réalisent ce qui doit être fait chaque fois que l'on active les étapes auxquelles elles sont associées.

Ces étapes peuvent être externes (sortie de l'automate pour commander le processus) ou internes (temporisation, comptage, etc...).

Une étape peut n'avoir aucune action, [15]

✚ Exemple : attente d'un événement

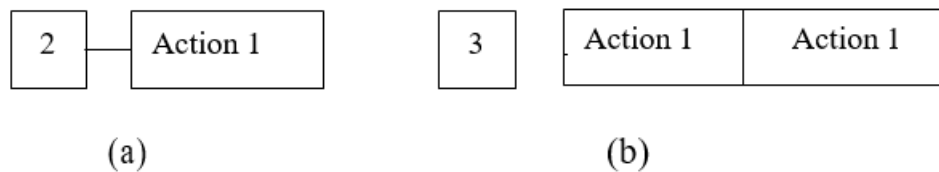


Figure 2-3 : Étape simple (a) / Étape a plusieurs actions (b). [15]

1.5.2. La transition et les réceptivités associées :

1.5.2.1. La transition :

Une transition est une condition de passage d'une étape à une autre. Elle n'est que logique (dans son sens Vrai ou Faux) et sans notion de durée. A chaque transition est associée une réceptivité.

On représente une transition par un trait vertical.

1.5.2.2. La réceptivité :

La réceptivité est une condition logique qui doit être remplie pour le franchissement de la transition. Elle est représentée à droite de la barre représentant la transition.

✚ Temporisation

La temporisation est une réceptivité qui permet une prise en compte du temps. Elle implique l'utilisation d'un temporisateur. Ce type de réceptivité est noté comme suite

$T/X_i/Q$ où i est le numéro de l'étape comportant l'action de temporisation, et Q est la durée écoulée depuis l'activation de l'étape X_i .

✚ Réceptivité toujours vraie

La réceptivité toujours vraie est une réceptivité dont la valeur logique est toujours « =1 », le franchissement de cette transition se fera dès que les étapes précédentes sont actives

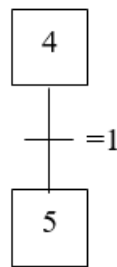


Figure 2-4 : Réceptivité toujours vraie

✚ Réceptivité à niveau

La réceptivité à niveau est une réceptivité faisant intervenir une condition logique. ^[16]

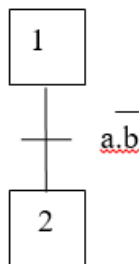


Figure 2-5 : Réceptivité à niveau

1.5.3. Liaison orientée :

Une liaison est un arc orienté (ne peut être parcouru que dans un sens). A une extrémité d'une liaison il y a une seule étape, à l'autre une transition. On la représente par un trait plein et rectiligne, vertical ou horizontal.

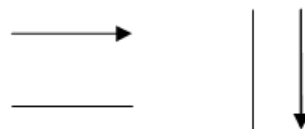


Figure 2-6 : Liaison orientée

1.6. Règles d'évolution du GRAFCET :

La modification de l'état de l'automatisme est appelée évolution, et est régie par 5 règles :

✚ Règle 1 : (situation initiale)

La situation initiale d'un GRAFCET caractérise le comportement initial de la partie commande vis-à-vis de la partie opérative, elle correspond aux étapes actives au début du fonctionnement.

✚ Règle 2 (franchissement d'une transition)

Une transition est soit validée, soit non validée Elle est validée lorsque toutes les étapes immédiatement précédentes sont actives.

Elle ne peut être franchie que lorsqu'elle est validée et que sa réceptivité est vraie. Elle est alors obligatoirement franchie.

✚ Règle 3 (évolution des étapes actives)

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

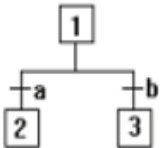
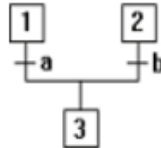
✚ Règle 4 (évolution simultanée)

Plusieurs transitions simultanément franchissables sont simultanément franchies.

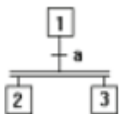
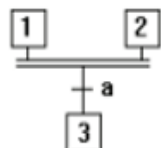
✚ Règle 5 (Maintien de l'activation en cas de nécessité simultanée)

Dans le cadre du fonctionnement, si une étape se trouve confrontée à la nécessité simultanée d'être à la fois activée et désactivée, son état demeure en position active. ^[16]

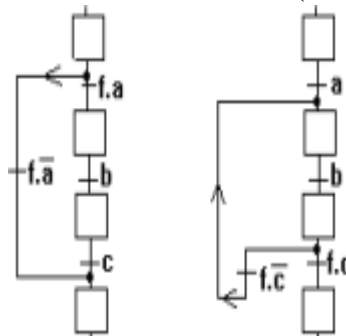
1.7. Configuration courante :

<p>divergence en OU :</p>  <p>si 1 active et si a seul, alors désactivation de 1 et activation de 2, 3 inchangé.</p> <p>si a et b puis 1 active alors désactivation 1, activation 2 et 3 quel que soit leur état précédent.</p>	<p>Convergence en OU :</p>  <p>Si 1 active et a sans b, alors activation de 3 et désactivation de 1, 2 reste inchangé</p> <p>Si 1 et 2 et a et b alors 3 seules active</p>
--	---

On appelle BARRE DE OU la barre symbolisant les entrées / sorties multiples d'étapes.

<p>Divergence en ET :</p>  <p>si 1 active et si a, alors désactivation de 1 et activation de 2 et 3.</p>	<p>Convergence en ET :</p>  <p>Si 1 active seule et a alors aucun changement. Si 1 et 2 et a, alors activation de 3 et désactivation de 1 et 2.</p>
---	--

La désignation courante de la double barre comme "BARRE DE ET" nécessite une précision : il ne s'agit pas d'une entité autonome, mais plutôt d'une composante d'une transition. De plus, nous allons maintenant examiner en détail le saut avant (condition "si a alors ...") ainsi que les



boucles (instruction "répéter ... jusqu'à c"). Ces deux éléments constituent les seules possibilités d'utilisation de l'opérateur logique « OÙ » ; il est important de souligner qu'il ne peut y avoir de divergence en amont ou en aval d'une transition.

Procédons désormais à l'examen de quelques problématiques plus complexes, puisées dans l'ouvrage intitulé "Comprendre et maîtriser le Grafcet" de Blanchard, publié par les éditions Capades.

- ✚ Soient 4 étapes 1 à 4 et deux transitions de réceptivité t1 et t2. Construire la portion de Grafcet réalisant,

Quand 1 ET 2 actifs alors :

- Si t1 passer en 3 (et désactiver 1 et 2),
- Si t2 passer en 4 (et désactivez 1 et 2), Sinon rester en 1 et 2

La solution ci -dessous est accompagnée d'une représentation de type "réseau de Petri" pour bien montrer où doivent se placer les convergences et divergences (à quoi doit être reliée 1 ? à quoi doit être reliée t1 ? ...).

En fait, on trouve la solution facilement en analysant les cas d'évolution (quand franchit-on T1 ?).

Il faut souligner que l'ajout d'une étape intermédiaire n'est pas une bonne solution car tout passage d'une étape dure un laps de temps (donc discontinuité sur les sorties = aléa technologique).

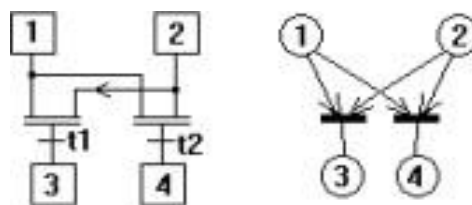


Figure 2-6 : Explication de la convergence et divergence dans le grafcet. [15]

- ✚ Problème du même ordre : Quand (étape 1 et t1) OU (étape 2 et t2) alors passer en 3 ET 4 :

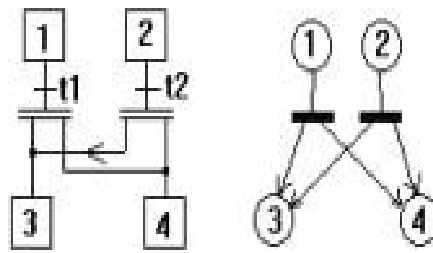


Figure 2-7 : explication de problème du même ordre dans le grafcet. [15]

- ✚ Si {étape 1 et [étape 2 ou (étapes 3 et 4)]} et transition et alors activer l'étape 5 (et désactiver les autres).

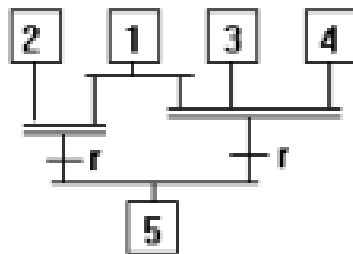


Figure 2-8 : explication d'un problème complexe du grafcet. [15]

2. Langage a contact (LADDER):

2.1. Définition :

Le langage ladder ou langage à contacts ou encore schéma à relais fait partie des 5 langages standards de la norme CEI 61131-3 défini par la commission électrotechnique internationale (CEI). Le langage ladder est un langage de programmation graphique facile à comprendre et à prendre en main. C'est sans doute le langage de programmation d'automatisme le plus couramment utilisé pour la programmation d'automates. Le langage ladder est composé d'une séquence de contacts (interrupteurs qui sont soit fermés, soit ouverts) et de bobines qui permettent de traduire les états logiques d'un système.

La logique Ladder est largement utilisée pour programmer les API, avec lesquels le contrôle séquentiel des processus de fabrication est requis. Le Ladder est utile pour les systèmes de contrôle simples mais critiques, ou pour reprendre d'anciens circuits à relais câblés. Comme les contrôleurs à logique programmable sont devenus plus sophistiqués, ils ont aussi été utilisés avec succès dans des systèmes d'automatisation très complexes. [14]

2.2. Objectif :





Définir un automatisme en utilisant les fonctions logiques ET, OU, NON, les bascules *RS*, les opérateurs à retard, les comparateurs, les compteurs. Mettre en œuvre des automatismes utilisant des fonctions logiques dans un API

2.3. Principe :





Un programme Ladder se lit de haut en bas et l'évaluation des valeurs se fait de gauche à droite. Les valeurs correspondent en fait, si on le compare à un schéma électrique, à la présence ou non d'un potentiel électrique à chaque nœud de connexion.

Le Ladder est basé sur le principe d'une alimentation en tension représentée par deux traits verticaux reliée horizontalement par des bobines, des contacts et des blocs fonctionnels, d'où le nom 'Ladder' ^[14] _

2.4. Élément de langage :**2.4.1. Les contacts :**

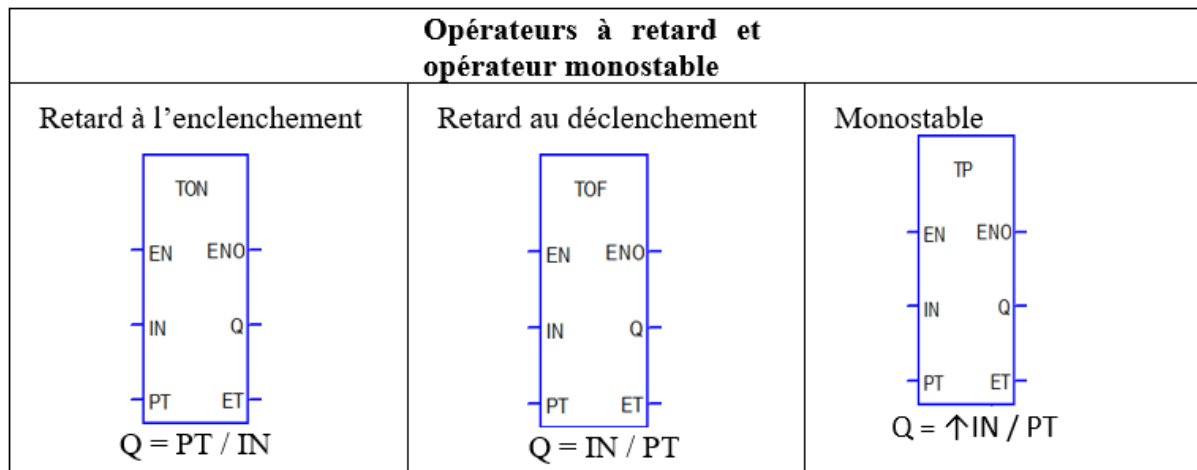
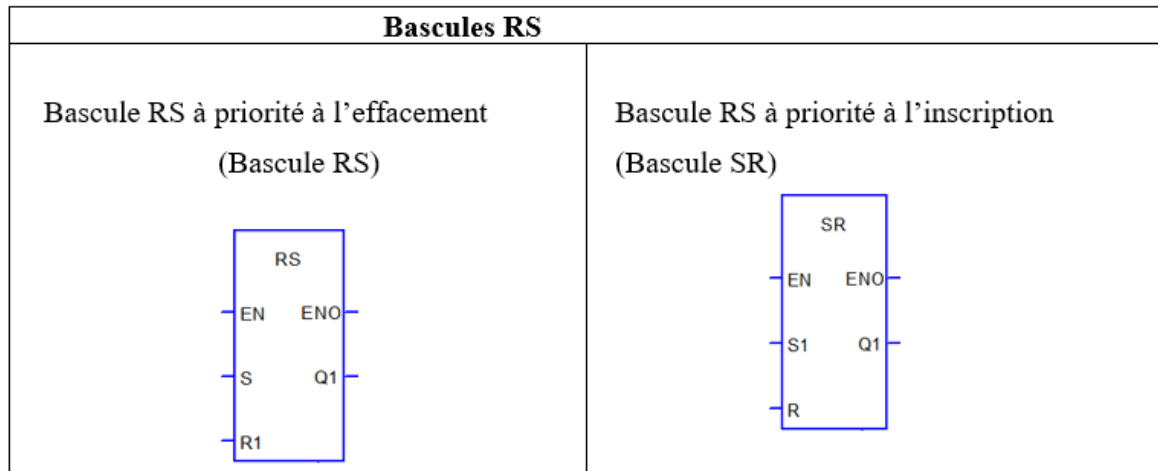
<i>Symbole</i>	<i>Désignation</i>	<i>Fonctionnement</i>
	Contact normalement fermé (NF)	Le contact est fermé lorsque la variable qui lui est associée est à l'état logique 0. Le contact est ouvert lorsque la variable qui lui est associée est à l'état logique 1.
	Contact à détection de front montant	Le contact est fermé lorsque la variable qui lui est associée passe de 0 à 1 (front montant). Le contact est ouvert lorsque la variable qui lui est associée est à l'état logique 0 ou à l'état logique 1.
	Contact à détection de front descendant	Le contact est fermé lorsque la variable qui lui est associée passe de 1 à 0 (front descendant). Le contact est ouvert lorsque la variable qui lui est associée est à l'état logique 0 ou à l'état logique 1.
	Contact normalement ouvert (NO)	Le contact est fermé lorsque la variable qui lui est associée est à l'état logique 1. Le contact est ouvert lorsque la variable qui lui est associée est à l'état logique 0

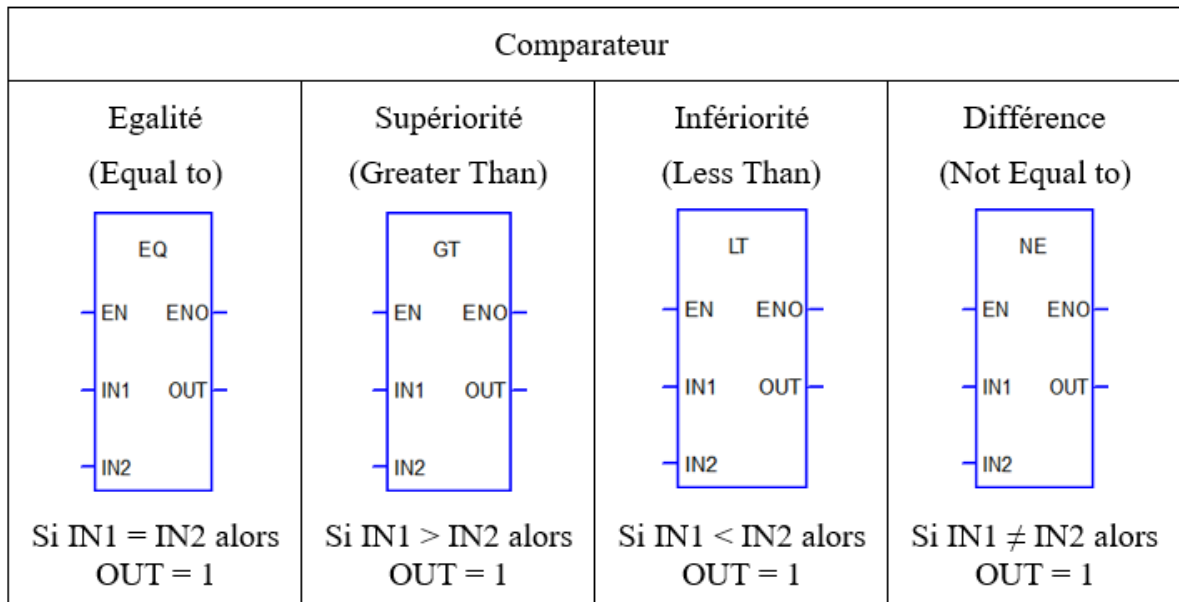
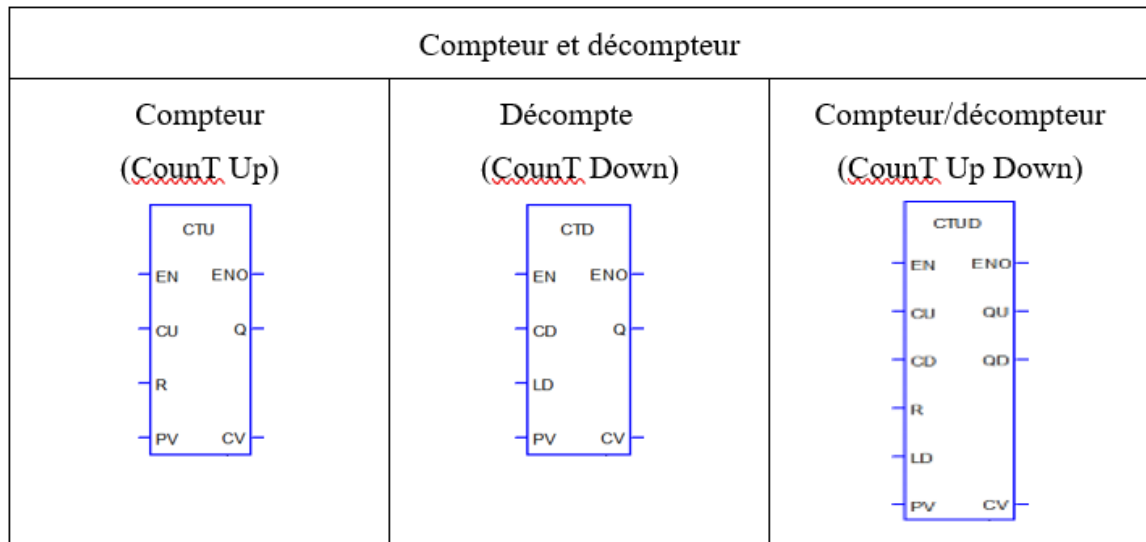
2.4.2. Les bobines :

Symbole	Désignation	Fonctionnement
	Bobine	La bobine est activée (excitée) lorsque les contacts auxquels elle est reliée sont fermés. La variable qui lui est associée est alors à l'état logique 1. Dans le cas contraire, la bobine est désactivée et la variable qui lui est associée est à l'état logique 0.
	Bobine « négative »	La bobine est activée (excitée) lorsque les contacts auxquels elle est reliée sont ouverts. La variable qui lui est associée est alors à l'état logique 1. Dans le cas contraire, la bobine est désactivée et la variable qui lui est associée est à l'état logique 0.
	Bobine d'enclenchement (Set = mise à 1)	La bobine est activée (excitée) dès que les contacts auxquels elle est reliée sont fermés. La variable qui lui est associée est alors à l'état logique 1. La bobine reste activée même si ensuite les contacts ne sont plus fermés.
	Bobine de déclenchement (Reset = mise à 0)	La bobine est désactivée dès que les contacts auxquels elle est reliée sont fermés. La variable qui lui est associée est alors à l'état logique 0. La bobine reste désactivée même si ensuite les contacts ne sont plus fermés.

2.4.3. Les blocs de fonctionnement :

Le langage LADDER permet l'utilisation de nombreuses fonctions logiques, telles que les bascules RS, les opérateurs à retard, l'opérateur monostable, le compteur et le décompteur, les comparateurs, etc. Ces fonctions logiques complexes sont appelées « blocs fonctions ».



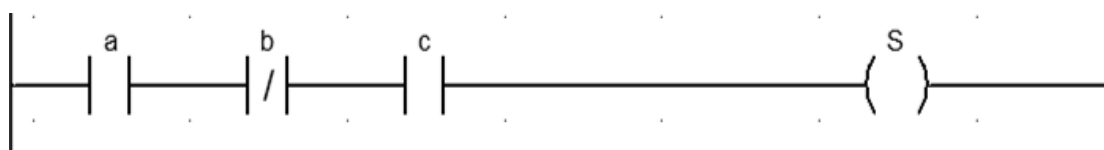


2.4.4. Association de contact et de bobine :

2.4.4.1. Contact en série :

L'association de contacts en série permet de réaliser la fonction logique ET.

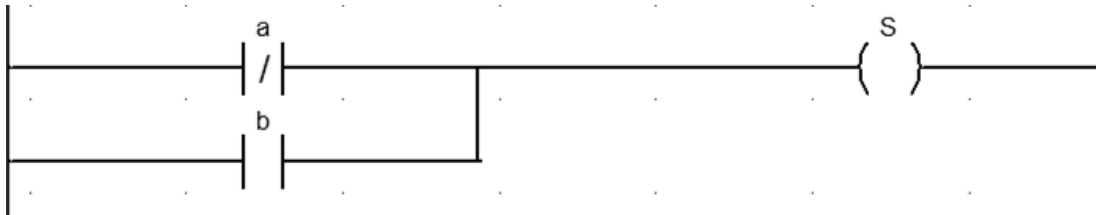
Exemple : Equation logique : $S = a\bar{b}c$



2.4.4.2. Contact en parallèle :

L'association de contacts en parallèle permet de réaliser la fonction logique OÙ.

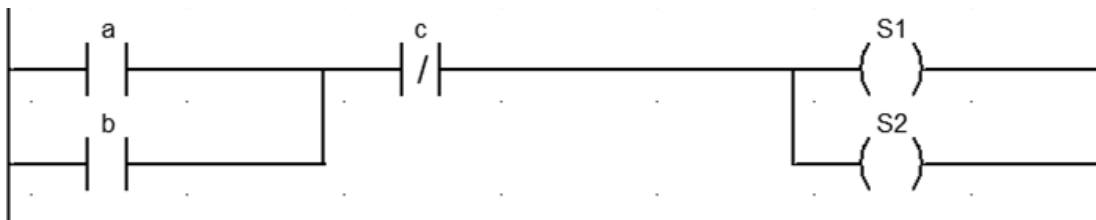
Exemple : Equation logique : $S = \bar{a} + b$



2.4.4.3. Bobine en parallèle :

L'association de bobines en parallèle permet de les commander par la même équation logique.

Exemple : Equation logique : $S1 = (ab)\bar{c}$, $S2 = (ab)c$



Remarque : l'association de bobines en série est impossible.

3. Langage liste d'instruction (IL) :

3.1. Présentation :


Les listes d'instructions constituent une méthode de programmation que l'on peut comparer à la saisie d'un schéma à contacts sous forme d'un texte. Un programme écrit selon cette méthode est constitué d'une suite d'instructions, chacune placée sur une nouvelle ligne. Chaque instruction est constituée d'un opérateur suivi d'un ou plusieurs opérandes, c'est-à-dire les objets de l'opérateur. Ainsi, nous pouvons avoir la ligne suivante :

LD A : Elle indique le chargement de l'opérande A ; LD est l'opérateur utilisé pour effectuer un chargement. Comparé au langage à contacts, un opérateur peut être vu comme un élément d'une ligne et LD équivaut à commencer une ligne par des contacts ouverts pour l'entrée A. Voici une autre instruction :

OUT Q : Elle indique qu'une sortie doit être produite sur Q.

Des codes mnémoniques sont utilisés pour les opérateurs, chaque code correspondant à un opérateur/élément d'un schéma à contacts. Ces codes diffèrent d'un fabricant à l'autre, mais la norme CEI 61131-3 a été établie et est à présent largement adoptée. Le Tableau recense quelques codes utilisés par les fabricants et les instructions normalisées Les listes d'instructions sont un langage textuel de bas niveau simple à implémenter et utilisé par nombre de fabricants d'API, principalement pour les modèles de petite et moyenne taille.

Ce langage est particulièrement adapté aux programmes simples. Certains fabricants ne prennent pas en charge les listes d'instructions, mais proposent le texte structuré, un langage de plus. ^[14]

 Le tableau ci-dessous illustre les codes mnémoniques des instructions, fournissant ainsi une représentation systématique et concise de ces dernières.

CEI 61131-3	Mitsubishi	OMRON	Siemens	Opération	Langage à contacts
LD	LD	LD	A	Charger l'opérande dans un registre de résultat.	Commencer une ligne avec des contacts ouverts.
LDN	LDI	LD NOT	AN	Charger l'opérande inversé dans un registre de résultat.	Commencer une ligne avec des contacts fermés.
AND	AND	AND	A	ET booléen.	Éléments en série avec des contacts ouverts.
ANDN	ANI	AND NOT	AN	ET booléen avec un opérande inversé.	Éléments en série avec des contacts fermés.
OR	OR	OR	O	OU booléen.	Éléments en parallèle avec des contacts ouverts.
ORN	ORI	OR NOT	ON	OU booléen avec un opérande inversé.	Éléments en parallèle avec des contacts fermés.
ST	OUT	OUT	=	Stocker un registre de résultats dans un opérande.	Une sortie

Voici une illustration de l'utilisation des opérateurs définis par la CEI 61131-3 :

LD A (*Charger A*)

AND B (*ET B*)

ST Q (*Stocker le résultat dans Q, c.-à-d. sortie sur Q*)

Sur la première ligne du programme, LD représente l'opérateur, A représenté l'opérande et les Mots placés en fin de lignes, entre parenthèses et précédés et suivis par un caractère *, sont des commentaires qui expliquent l'opération et ne font pas partie des instructions du programme de

TAPI. LD A est ainsi une instruction qui charge A dans le registre mémoire. Il peut ensuite être rappelé pour d'autres opérations. La ligne suivante du programme effectue une opération

booléenne ET entre A et B. La dernière ligne place le résultat dans Q, c'est-à-dire sur la sortie Q.

Il est possible d'utiliser des libellés pour identifier différents points d'entrée dans un programme.

Nous le verrons plus loin, ils sont utiles pour sauter à des endroits d'un programme. Un libellé se place avant l'instruction et en est séparé par des deux-points :

```
P0MPE OK: LD C (* Charger CD*)
```

Pour les opérateurs de la CEI 6113-3, l'ajout d'un N à la fin de leur mnémonique indique une valeur inversée. Par exemple :

```
LD A (*Charger A*)
```

```
AND B (*ET NON B*)
```

Ainsi, l'opérateur AND inverse la valeur des contacts et réalise un ET avec le résultat. ^[14].

3.2. Instruction de base en langage liste :

Instructions de test	
Désignation	Fonctions
LD	Le résultat est égal à l'opérande (load : lire la valeur).
LDN	Le résultat est égal à l'inverse de l'opérande (contact ouverture).
AND	Et logique entre le résultat et précédent et l'état de l'opérande.
ANDN	Et logique entre le résultat et précédent et l'état inverse de l'opérande.
OR	OU logique entre le résultat et précédent et l'état de l'opérande.
ORN	OU logique entre le résultat et précédent et l'état inverse de l'opérande.
XOR, XOR	OU exclusif.
Instructions d'action	
ST	L'opérande associé prend la valeur de la zone de test.
STN	L'opérande associé prend la valeur inverse de la zone de test.
S	L'opérande associée est mise à 1 lorsque le résultat de la zone de test est à 1.
R	L'opérande associée est mise à 1 lorsque le résultat de la zone de test est à 1.

L'adresse ou le code opérande est précédé de %, voici un exemple d'écriture pour illustrer cette convention :

N° de ligne	Instruction	Opérande	Commentaires
00	LD	% I0, 01	Tester entrée d'adresse 0,01
01	AND	% I0, 02	ET entre l'entrée (I0, 01) et l'entrée I0, 02
02	ST	% O 0,02	Donner le résultat logique du ET à la sortie Q0, 01

4. Le langage LOG (logigramme) :

4.1. Définition :

Le logigramme s'appelle aussi ordinogramme ou abusivement organigramme. C'est est un outil utilisé en qualité qui permet de visualiser de façon séquentielle et logique les actions à mener et les décisions à prendre pour atteindre un objectif défini.

Un logigramme permet de décrire complètement une activité. Il est ainsi possible de mettre en évidence les éventuels gaspillages ou non-valeur ajoutée, dans une démarche d'amélioration continue. ^[14]

4.2. Le but d'un logigramme :






Cet outil s'utilise pour décrire une activité de façon complète. Il est notamment utilisé pour la rédaction des procédures.

Un logigramme peut avoir plusieurs objectifs :

- Vérifier si le processus est conforme
- Comprendre le rôle de chaque étape
- Améliorer certaines étapes voir les supprimer.
- Enrichir les connaissances du personnel
- Analyser un problème et apporter une solution. ^[14]

4.3. Symbole basique de logigramme processus :

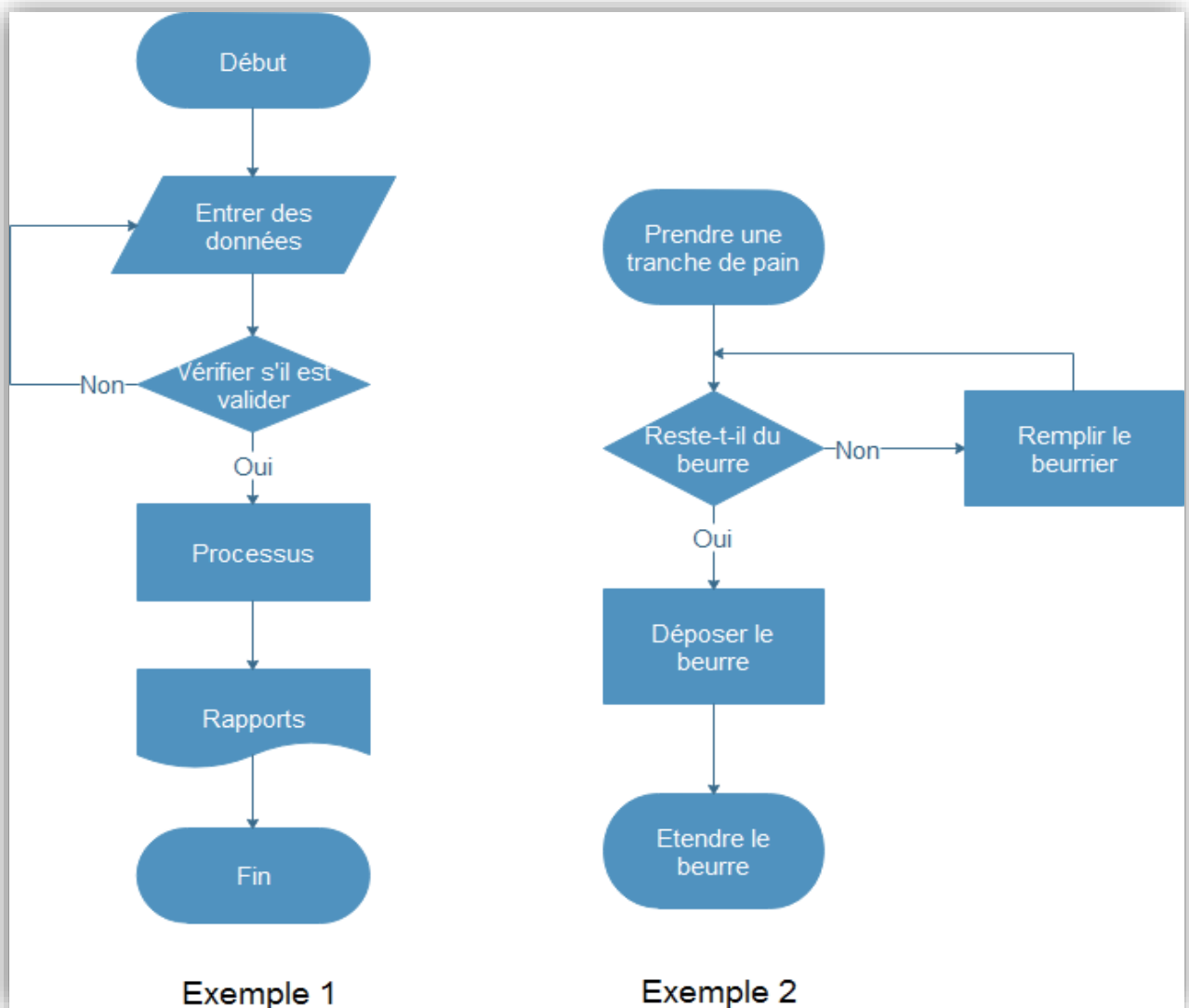
Un logigramme est principalement constitué des symboles suivants, qui servent à identifier les grandes phases du processus. Ces symboles jouent un rôle essentiel en permettant d'obtenir une vision globale du processus :

	Premier étape et dernière étape
	Autres étapes
	Un choix, une décision répond toujours par Oui ou Non
	Document lié à une étape
	Lien entre 2 activités

✚ Exemple logigramme simple :

Exemple 1 est une forme de structure de logigramme présentant le processus pour créer un logigramme ;

Exemple 2 est un exemple présentant le processus pour faire une tartine de beurre.



5. Langage texte structuré (ST) :

5.1. Définition :

Le langage ST (Structured Text) est un langage textuel de haut niveau dédié aux applications d'automatisation.

Ce langage est principalement utilisé pour implémenter des procédures complexes, difficilement modélisables avec les langages graphiques.

Le langage EST peut-être utilisé pour la programmation des actions dans les étapes et les conditions associées aux transitions du SFC ou pour décrire les actions et les tests du langage FC. ^[14]

5.2. Syntaxe du ST :

Un programme ST est une suite d'énoncés. Chaque énoncé se termine par un point-virgule (";").

Les noms utilisés par le code source (identificateurs de variables, constantes, mots-clés du langage, etc.) sont délimités par des séparateurs passifs (espace, fin de ligne ou tabulation) ou par des séparateurs actifs, qui ont une signification bien définie et qui jouent le rôle d'opérateurs (par exemple, le séparateur ">" indique une comparaison "Plus grand que").

Des commentaires peuvent être insérés librement dans le texte de programmation. Un commentaire doit commencer par les caractères "(*" et se terminer par "*)". Voici les types d'énoncés standards du ST :

- Assignment (variable :≥ expression);
- Appel de fonction ;
- Appel de bloc fonctionnel ;
- Énoncés de sélection (**IF, THEN, ELSE, CASE, etc.**) ;
- Énoncés d'itération (**FOR, WHILE, REPEAT, etc.**) ;
- Énoncés de contrôle (**RETURN, EXIT, etc.**) ;
- Énoncés spéciaux pour le lien avec d'autres langages, tels que le SFC.
- A ne pas écrire plusieurs énoncés sur la même ligne
- Utiliser les tabulations pour indenter les structures de contrôle
- Insérer des commentaires ^[14]

Chapitre III :

Modélisation des trois systèmes industriels

1. Simatic Manager – STEP-7
 - 1.1 Introduction
 - 1.2. Définition de logiciel Simatic Step-7
 - 1.3. Création d'un projet
 - 1.4. Définition de matériel
 - 1.5. Table des Mnémoniques
 - 1.6. Avantages
 - 1.7. Blocs du programme utilisateur
 - 1.8. Structure d'un programme STEP 7
 - 1.9. Programme cyclique OB 1
 2. L'équipement de travail au sein du laboratoire d'automatisation
 3. Développement et programmation des systèmes industriels à l'aide de Simatic Step7
 - 3.1. Système de remplissage contrôlé
 - 3.2. Commande de deux portes de serrures
 - 3.3. Système de réseau d'air comprimé
- Conclusion

1. Simatic Manager – STEP-7 :

1.1. Introduction :

Les automates programmables industriels se chargent de l'automatisation des tâches en exécutant des programmes d'application qui spécifient la méthode selon laquelle l'automate doit commander le système par le biais d'une série d'instructions. Ces programmes sont rédigés dans un langage spécifique, conformément à des règles définies, pour permettre à l'automate de les exécuter. Dans le cas des automates de la famille SIEMENS, leur programmation est réalisée au moyen du logiciel STEP7, utilisé avec une console de programmation ou un PC, et fonctionnant dans l'environnement Windows. [18].

1.2. Définition de logiciel Simatic Step-7 :

SIMATIC STEP7 est un logiciel d'ingénierie développé par Siemens qui offre des fonctionnalités de programmation pour les automates de la gamme Siemens. La dernière version de STEP7 est intégrée dans le logiciel d'ingénierie Siemens TIA Portal (Totally Integrated Automation). TIA Portal est une solution tout-en-un qui permet de programmer des automates, des panels et des contrôleurs d'axes Siemens.

STEP7 permet un accès de base aux automates Siemens, offrant la possibilité de programmer individuellement un automate dans différents langages. Il prend également en charge la connectivité réseau, permettant d'accéder à tous les automates du réseau pour les programmer et, éventuellement, pour les automates de s'échanger des messages entre eux.

Cependant, il ne permet pas l'intégration des ordinateurs dans l'automatisme, ce qui est possible avec des logiciels tels que PCS7 ou TIA Portal. Le logiciel, connu sous le nom de "Simatic Manager", est généralement accessible via le menu démarrer sous l'onglet "Siemens" (uniquement sur les postes du Hall Techno où le matériel est installé). [18].

1.3. Création d'un projet :

Dans le cas où l'on n'utilise pas l'assistant (Wizard) :

- ✚ Lancer SIMATIC STEP7,

Remarque: le logiciel s'ouvre sur le ou les derniers projets qui ont été fermé.

- ✚ Dans la barre d'outils sélectionner : "Fichier > Nouveau ...",
- ✚ Donner un nom au projet (ici "Test_proj"),
- ✚ Valider par le bouton OK.

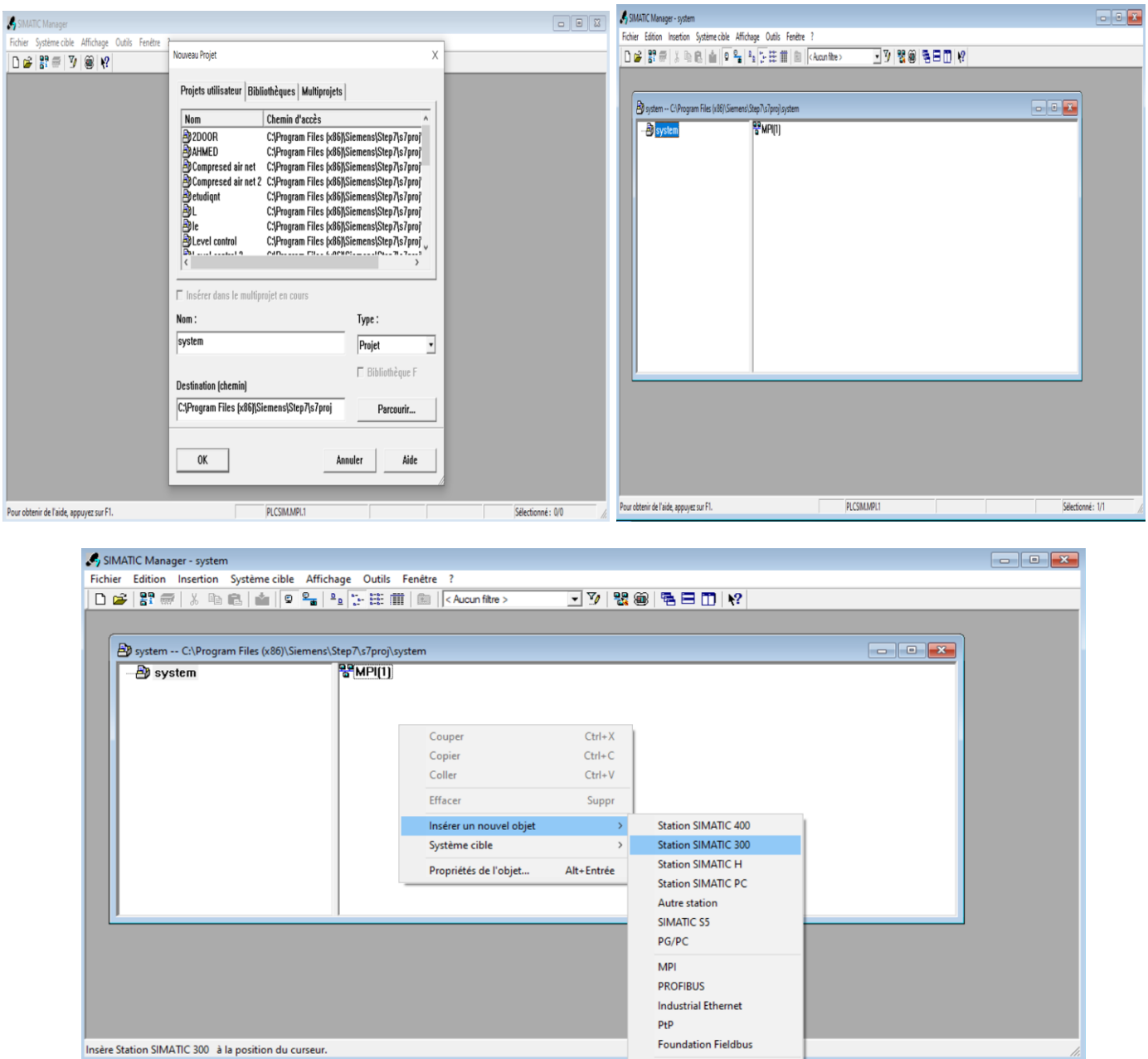


Figure 3-1 : création d'un projet dans step 7.

Le projet a été instancié avec succès, entraînant l'apparition d'une nouvelle fenêtre qui présente les éléments suivants de manière distincte:

- À gauche, une représentation arborescente du projet qui affiche tous ses composants.
- À droite, une mise en évidence des détails spécifiques du constituant actuellement sélectionné dans le projet.

1.4. Définition de matériel :

Avant d'entamer le processus de programmation, il est préalablement nécessaire de procéder à la déclaration du matériel utilisé dans le cadre du projet. Prenons l'exemple d'un système S7-300 connecté à un réseau Ethernet, auquel nous ajouterons une station distante de type ET200, reliée par un réseau PROFIBUS. À noter : en théorie, pour pouvoir programmer, il suffit d'insérer un Rack (Rail) et une CPU (contenant le programme).

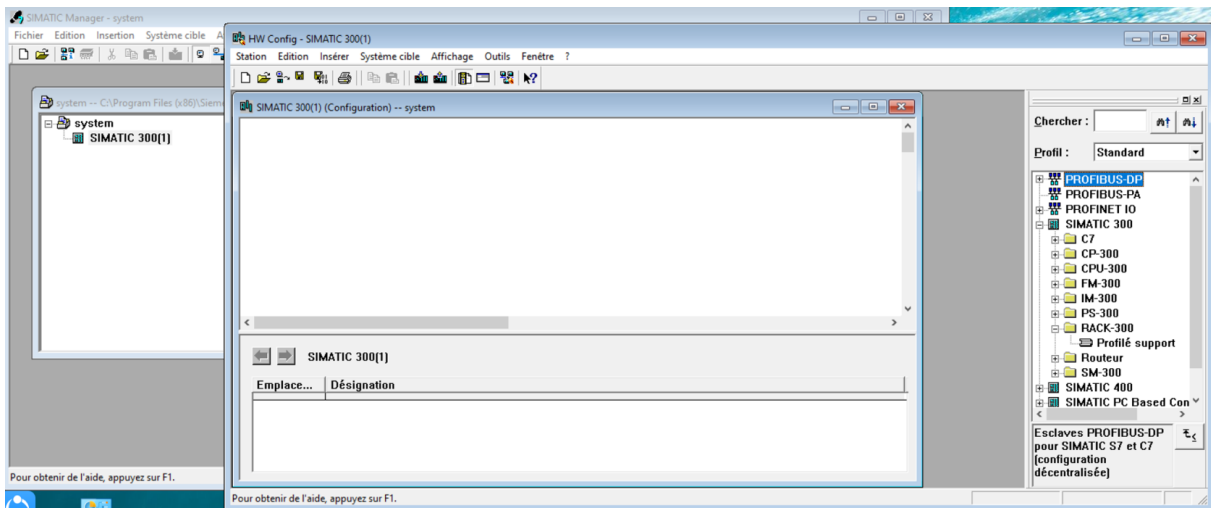


Figure3-2 : définition du matériels sur le simatic

Dans la structure hiérarchique du projet, effectuez un clic droit sur l'icône actuelle représentant le projet unique, puis sélectionnez l'option "Insérer un nouvel objet > Station SIMATIC 300". Cela aura pour effet d'ajouter un automate (ou une station) S7-300 à votre configuration, que vous pourrez renommer au besoin. [20].

➤ **Introduction et configuration du matériel:**

Pour entamer la procédure, veuillez sélectionner l'automate dans la structure hiérarchique du projet. Une fois cette étape accomplie, la fenêtre de droite affichera tous les éléments associés à l'automate sélectionné. Actuellement, seule l'icône "Matériel" est disponible. Double-cliquez sur cette icône pour ouvrir l'application "HW config" (abréviation de Hardware config), un utilitaire destiné à gérer la configuration matérielle de l'automate. [20].

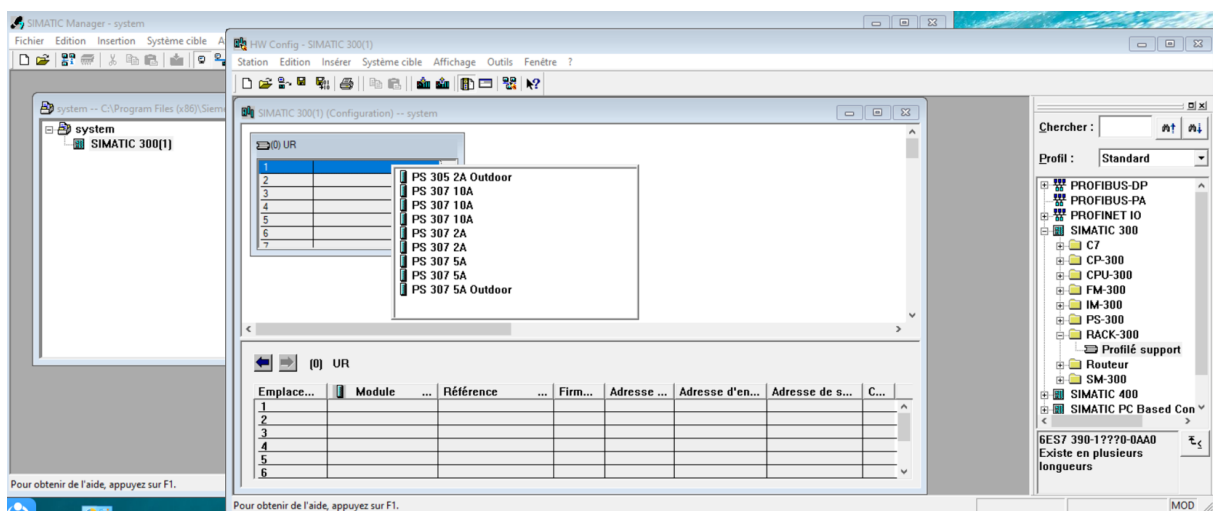
➤ **Vue d'ensemble :**

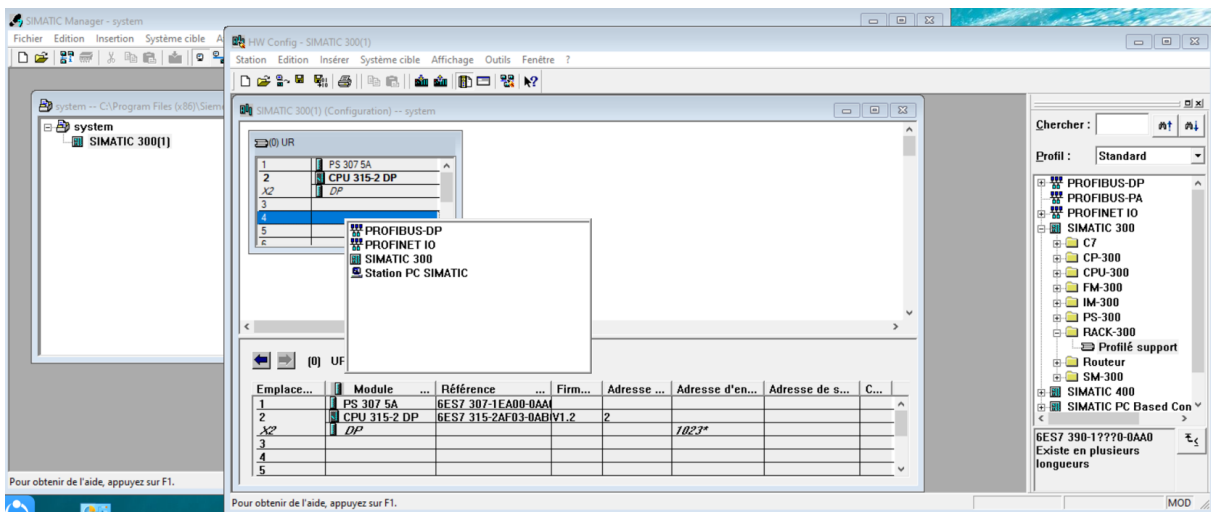
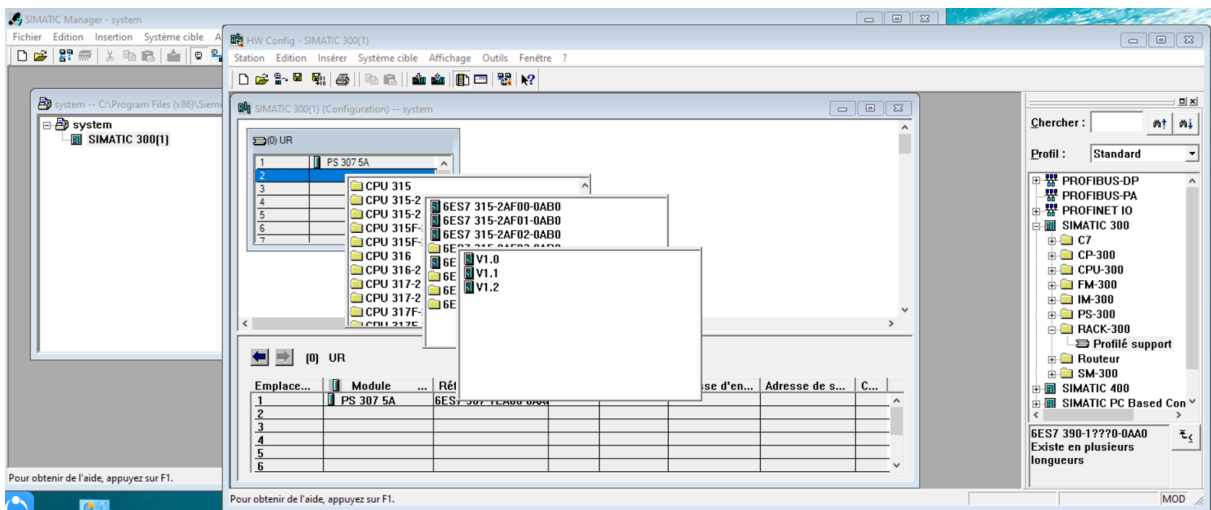
Cette application comprend trois parties:

- à gauche : l'implantation du matériel (automates, réseaux, esclaves, ...),
- à droite : le catalogue des constituants de la gamme SIMATIC (à valider à travers la barre d'outils : "Affichage > Catalogue" si elle n'est pas présente),
- en bas : un tableau comprenant les informations de l'objet sélectionné via la zone de gauche (implantation du matériel). [20].

➤ **Insertion d'un automate avec ses modules :**

Pour constituer notre automate nous allons nous servir du catalogue (droite de l'écran), et plus particulièrement la partie "SIMATIC 300" du catalogue qui contient les modules pour notre automate S7-300.





➤ Voici un exemple de configuration pour un automate S7-300 :

- Une alimentation 5A,
- Une CPU 315-2DP,
- Un coupleur Ethernet CP-343,
- Une carte de 16 Entrées TOR,
- Une carte de 16 Sorties TOR,
- Une carte de 2 Entrées Analogiques,
- Une carte de 2 Sorties Analogiques [20].

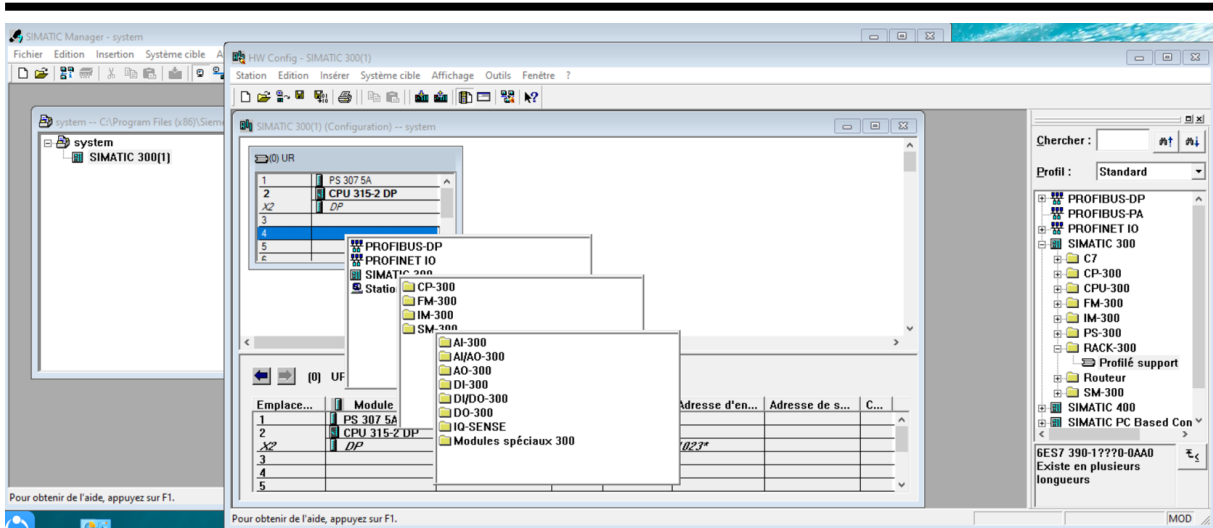


Figure 3-3, 4, 5, 6 : le catalogue des modules de l’automate s7-300.

1.5. Table des Mnémoniques :

Dans la table des mnémoniques, vous définissez les mnémoniques globales.

Une table des mnémoniques (objet "Mnémoniques") vide est automatiquement générée lorsque vous créez un programme S7 ou M7.

- **Domaine de validité :**

La table des mnémoniques vaut pour le module auquel le programme est associé. Si vous voulez vous servir des mêmes mnémoniques dans différentes CPU, vous devez vous-même faire en sorte qu’elles entrées correspondent dans les différentes tables de mnémoniques (par exemple, par copie)

- **Structure de la table de mnémoniques :**

Etat	Mnémonique	Opérande	Type de do	Commentaire
1	M=50%	E	0.7	BOOL
2	M5	A	0.2	BOOL
3	P=50%	E	1.1	BOOL
4	P=60%	E	1.0	BOOL
5	P=90%	E	0.6	BOOL
6	PS	M	0.0	BOOL
7	S1	E	0.0	BOOL
8	S19	E	0.3	BOOL
9	S19 BAR	E	0.4	BOOL
10	S2	E	0.1	BOOL
11	S2 BAR	E	0.2	BOOL
12	S20	E	0.5	BOOL
13	TR1	E	1.2	BOOL
14	TR2	E	1.3	BOOL
15	TR3	E	1.4	BOOL
16	TR4	E	1.5	BOOL
17	TR5	E	1.6	BOOL
18	TR6	E	1.7	BOOL
19	TR7	E	2.0	BOOL
20	X1	M	0.1	BOOL
21	X2	M	0.2	BOOL
22	X3	M	0.3	BOOL
23	X4	M	0.4	BOOL
24	X5	M	0.5	BOOL
25	X6	M	0.6	BOOL
26	X7	M	0.7	BOOL
27	Y1M1/Y2M2	A	0.0	BOOL

Figure 3-7 : table de mnémonique.

1.6. Avantages :

L'atelier logiciel STEP 7 Professional offre des avantages significatifs en termes de productivité tout au long des différentes phases d'un projet d'automatisation. Ces avantages comprennent notamment :

- ✚ Une configuration plus rapide des systèmes grâce à l'utilisation d'outils graphiques permettant de configurer les composants et les réseaux.
- ✚ Une programmation plus efficace en permettant la combinaison de tous les langages de programmation standard selon la norme CEI 61131-3 au sein de projets structurés et orientés objet.
- ✚ Des phases de test plus courtes grâce à l'intégration d'outils de simulation et de débogage.
- ✚ Une réduction significative des temps d'arrêt grâce à des fonctionnalités avancées de diagnostic de pannes et de maintenance à distance. [20].

1.7. Blocs du programme utilisateur :

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

- Blocs fonctionnels (FB) associés automatiquement à une 'mémoire' (DB d'instance).
- Blocs fonctions (FC) contenant des sous programmes pour les fonctions fréquemment utilisées.
- Blocs fonctionnels système (SFB) intégrés à la CPU, ils permettent de réaliser quelques fonctions système importantes.
- Blocs fonctions système (SFC) intégrés à la CPU.
- Blocs de données d'instance (DB d'instance) associés aux FB et aux SFB quand ces derniers sont appelés.
- Blocs de données globaux (DB) contenant les données utilisateurs communs à tous les blocs. [20]

1.8. Structure d'un programme STEP 7 :

Le logiciel STEP 7 met en œuvre des structures de programmation telles que les blocs d'organisation (OB) et les fonctions et blocs fonctionnels (FB et FC) pour faciliter la création de programmes modulaires.

Chaque module est dédié au traitement d'une fonction spécifique de l'automatisme, et ces modules sont appelés par le programme principal (OB 1) afin d'assurer leur exécution harmonieuse. ^[20]

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

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

1.9. Programme cyclique OB 1 :

Lors d'une exécution normale de programme, les traitements se font de façon cyclique.

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

- **Programme de démarrage OB 100 :**

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

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

Figure III-8 : Liste des Blocs d'organisation OB disponible sur l'API S7-300.

1.10. Les Fonctions et Blocs fonctionnels FC et FB :

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

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

Les FB et FC s'adaptent particulièrement bien à la programmation de fonctions récurrentes. Contrairement aux FC, les FB sont des blocs avec mémoire, les paramètres transmis aux FB sont sauvegardés dans un bloc de donnée d'instance DB. ^[20]

1.11. Simulation :

Nous avons utilisé un logiciel optionnel appelé PLCSIM, qui est intégré à l'atelier logiciel STEP 7 Professional. Ce logiciel permet d'effectuer des tests dynamiques sur les programmes conçus pour toute configuration d'automate SIMATIC S7, sans avoir besoin du matériel cible. Il offre la possibilité de simuler un automate de la famille SIEMENS avec tous ses modules.

Le simulateur présente une interface conviviale, où il suffit de cocher les cases correspondantes pour changer l'état des entrées. Les états des sorties sont automatiquement mis à jour en fonction de l'évolution du programme. Pendant la simulation, dans la fenêtre de programmation (CONT), chaque contact représentant une variable active est affiché en vert (les contacts non actifs sont affichés en pointillé). Cela permet de suivre en détail l'évolution du programme. La simulation nous a permis de tester les différentes situations auxquelles le système peut être confronté.

En conclusion, nous avons vérifié que notre programme répond exactement aux exigences spécifiées dans le cahier des charges, et qu'il peut donc être transféré du PC vers l'automate correspondant. [20]

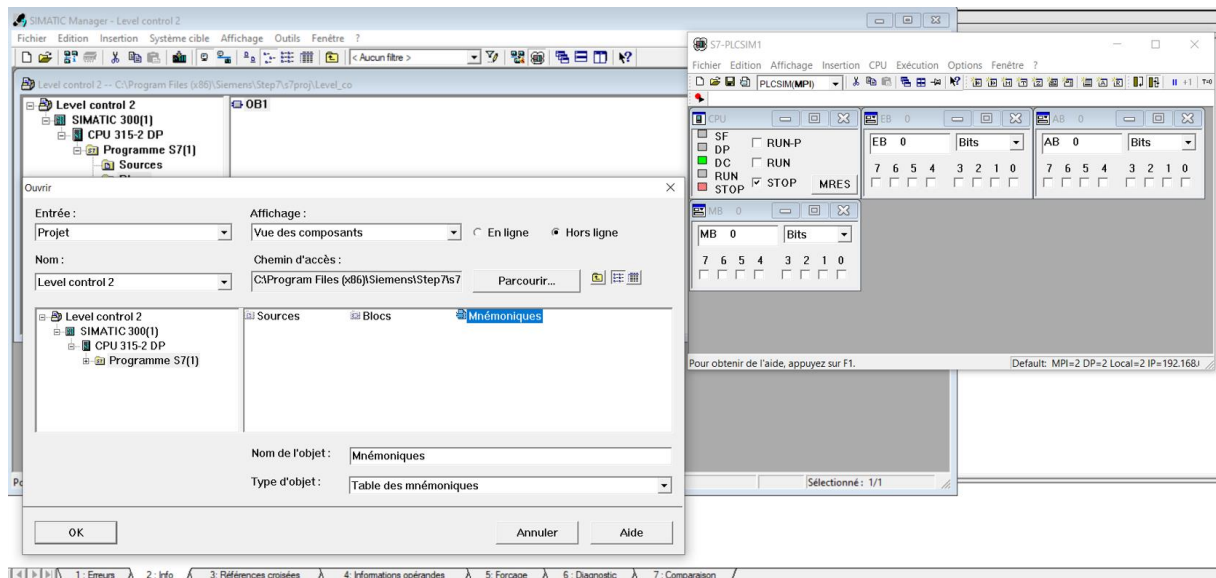


Figure 3-9 : Outil de simulation STEP7 (PLCSIM).

2. L'équipement de travail au sein du laboratoire d'automatisation:

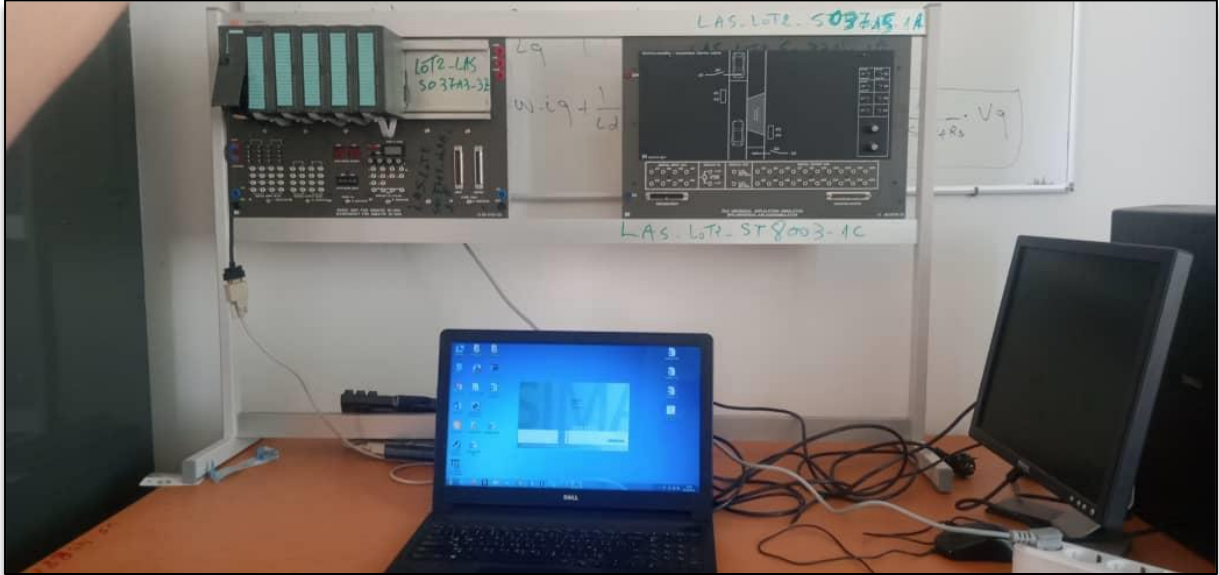


Figure 3-10 : Equipement de travail dans le laboratoire

3. Développement et programmation des systèmes industriels à l'aide de Simatic Step7 :

3.1. Système de remplissage contrôlé :

Tache requise et cahier de charge :

Un système de remplissage contrôlé doit toujours être plein à **50%** et le niveau doit fluctuer à **±10%**

Description fonctionnelle :

Après l'activation du bouton de démarrage S1, l'électrovanne ouvre l'accès, si le niveau atteint le capteur de remplissage minimum S20, la décharge est ouverte via l'électrovanne Y1.

L'évacuation est fermée si le niveau devient inférieur au niveau minimum souhaité

Le processus de remplissage, qui est affiché sur l'indicateur à barres, est fermé si le récipient est plein à 60 %. L'électrovanne Y2 se ferme au moyen du bouton rotatif "Vider", le récipient peut être vidé. Si ce faisant, le niveau du bidon atteint 40 %, il se remplit automatiquement

Si par S19 un sur remplissage du conteneur est indiqué, l'équipement est automatiquement déconnecté au moyen du bouton d'arrêt S2, l'équipement peut être déconnecté, avec lequel le conteneur est vidé automatiquement. [19]

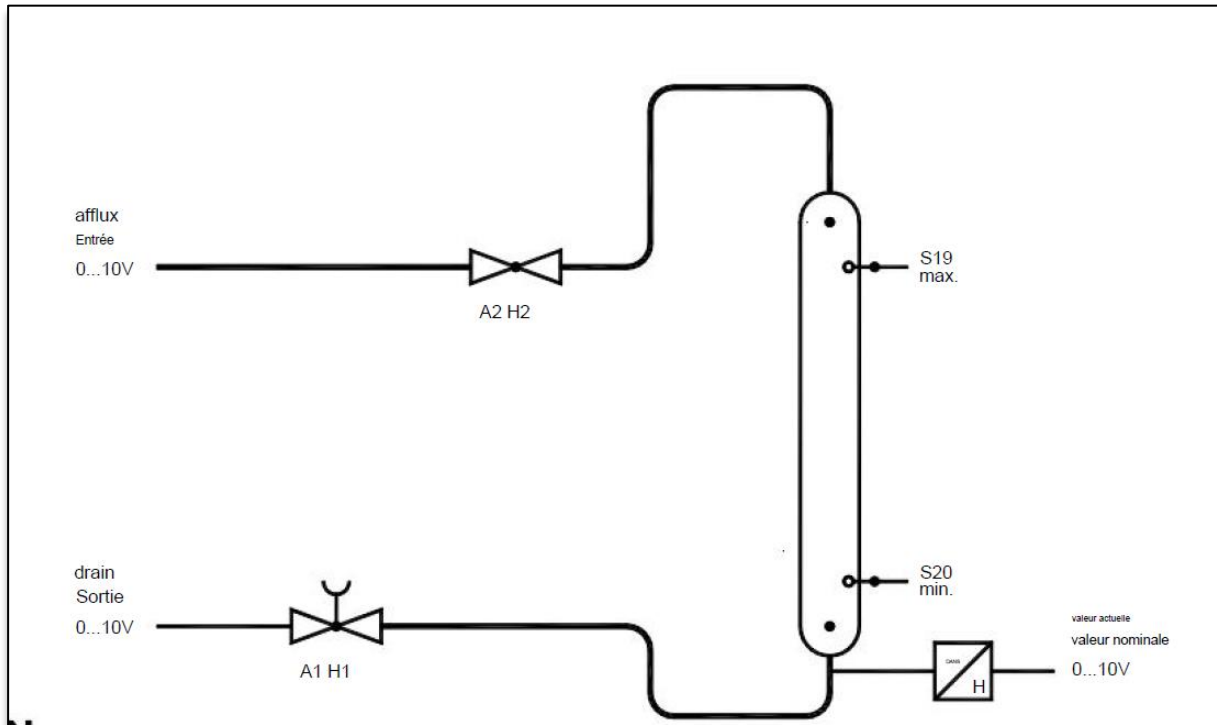


Figure 3-11 : Illustration de système de remplissage contrôlé [19]

- *Analyse approfondie et détaillée du cahier des charges - une explication complète des exigences et spécifications :*

L'objectif de ce processus est de fournir un réseau d'air comprimé alimenté par un réservoir sous pression, en maintenant la pression dans des plages de fonctionnement sûres et en ajustant la production d'air comprimé en fonction de la demande simulée.

Le système est également équipé de protections contre les surpressions et les baisses de pression, avec des fonctionnalités d'arrêt d'urgence en cas de panne ou de défaillance.

✚ *Activation du système :*

Lorsque le bouton de démarrage S1 est activé, les électrovannes Y1 et Y2 s'ouvrent. En même temps, les moteurs de compression M1 et M2 sont démarrés, ce qui permet de pomper

de l'air comprimé dans le réservoir. Les compresseurs fonctionnent à 70% de leur capacité de pompage.

✚ *Contrôle de la pression maximale du réservoir :*

Lorsque la pression dans le réservoir atteint 80% de la pression maximale, les compresseurs sont déconnectés et les vannes magnétiques sont fermées pour empêcher toute surpression.

✚ *Simulation de la demande d'air comprimé :*

L'intensité de charge du réseau d'air comprimé est simulée en utilisant le bouton rotatif "Evacuation". Cela ajuste la consommation d'air pour simuler la demande du système.

✚ *Gestion de la pression basse :*

Si la pression dans le réservoir chute à 60% de la pression maximale, l'électrovanne Y1 s'ouvre et le compresseur M2 est activé à 70% de sa capacité de pompage pour compenser la baisse de pression. Si la pression continue de chuter jusqu'à atteindre 50%, l'électrovanne Y1 s'ouvre à nouveau et le compresseur M1 est ajouté à 70% de sa capacité de pompage.

✚ *Protection contre les seuils de pression :*

Si la pression du réservoir dépasse la pression maximale (S19) ou tombe en dessous de la pression minimale (S20), tous les compresseurs sont arrêtés, toutes les électrovannes sont fermées et un message de défaut (H5) est affiché pour indiquer l'anomalie.

✚ *Arrêt du système en cas de panne :*

En cas de panne, l'appareil peut être éteint à l'aide du bouton d'arrêt S2. Une fois la panne réparée, le réseau d'air comprimé peut être redémarré en utilisant le bouton de démarrage S1.

✚ *Suppression de l'invite d'alarme au démarrage :*

Pour éviter une alarme due à une pression très basse lors du démarrage, l'invite d'alarme est supprimée pendant 5 secondes après la mise sous tension. Cela permet au système de se stabiliser avant de déclencher une alarme éventuelle.

- ✚ Visualisation graphique des divers comportements de système via le Grafcet :

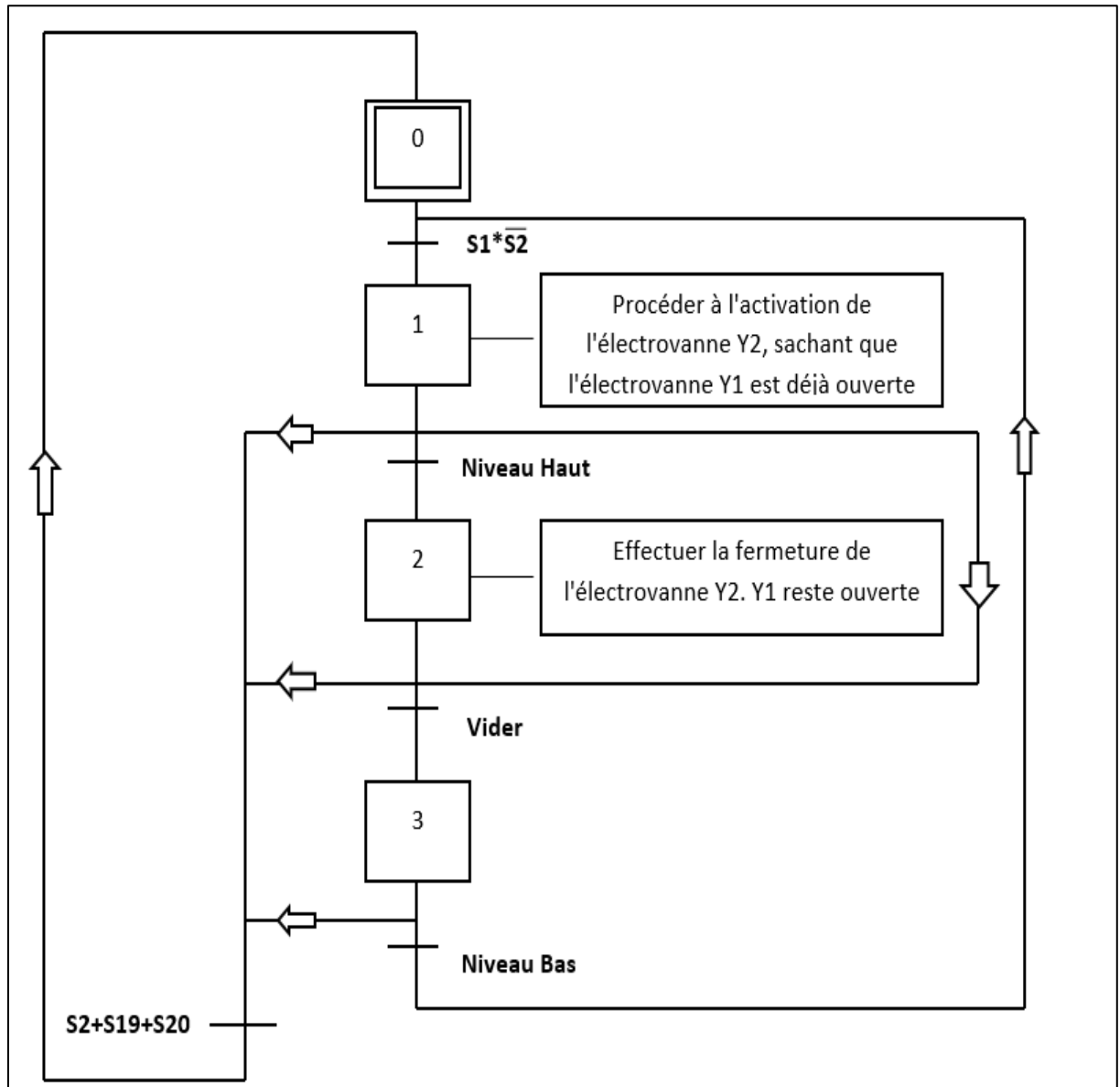
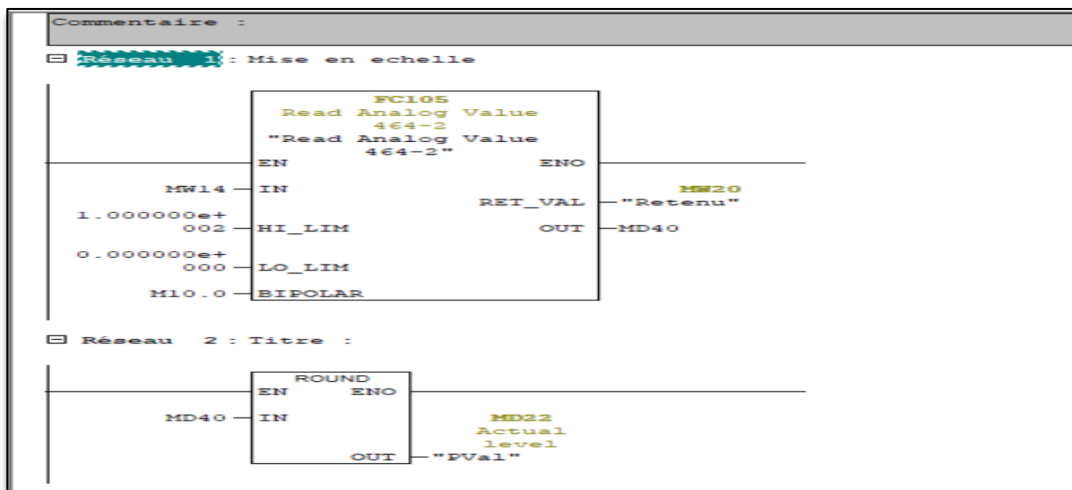


Figure 3-12 : Conception d'un système de remplissage contrôlé par le GRAFCET

❖ Développement et programmation du système via Simatic Step 7 :

	Etat	Mnémonique /	Opérande	Type de do	Commentaire
1	X				
2	X				
3	X				
4	X				
5	X				
6	X				
7		Bouton d'arrêt S2	E 0.1	BOOL	
8		Bouton démarrage S1	E 0.0	BOOL	
9		Bouton rotatif (vider)	E 0.4	BOOL	
10		capteur niveau max ...	E 0.2	BOOL	
11		capteur niveau min S...	E 0.3	BOOL	
12		Ouvert félectrova Y1...	A 0.0	BOOL	
13		Ouvert félectrovan Y1	A 0.1	BOOL	
14		Ouvert félectrovan Y2	A 0.2	BOOL	
15		THH 60%	E 1.1	BOOL	
16		THL 40%	E 1.2	BOOL	
17		x1	E 0.5	BOOL	
18		x2	E 0.6	BOOL	
19		x3	E 0.7	BOOL	
20		x4	E 1.0	BOOL	
21					

Figure 3-13 : Tableau de mnémoniques du système de remplissage contrôlé



1.1. Commande de deux portes de serrures :

Tache requise et cahier de charge :

Afin qu'une pièce reste **le moins poussiéreuse possible**, un système de portes coulissantes a été installé.

Description de l'opération :

Pour passer le verrou, le bouton de demande S1 ou S2 doit être activé. Comme il n'y a pas de boutons de sécurité à l'intérieur de la serrure pour ouvrir les portes de l'intérieur, il faut s'assurer que les deux portes ne restent jamais fermées.

Si une personne souhaite franchir la porte 1, qui est fermée, de l'extérieur vers l'intérieur, le bouton S1 est activé. La porte 2 se ferme et la porte 1 s'ouvre immédiatement après que l'interrupteur de fin de course S15 indique que la porte 2 est fermée.

Lorsque la personne franchit le seuil, la barrière lumineuse envoie une impulsion et après un délai de 3 secondes se ferme la porte 1. Immédiatement après que la fin de course S14 indique que la porte 1 est fermée, la porte 2 s'ouvre.

Le processus correspondant est également valable pour un mouvement dans le sens opposé.

Si le seuil est franchi par une porte ouverte, sans action préalable sur les boutons S1 ou S2, le verrouillage ne réagit pas chaque entrée d'écluse est gardée par une barrière lumineuse. Tant que la barrière lumineuse n'est pas interrompue, la porte ne doit pas être fermée. S'il est interrompu pendant la manœuvre de fermeture, le processus doit s'arrêter immédiatement et doit continuer instantanément une fois la barrière lumineuse activée.

Les voyants lumineux, situés à côté des boutons de demande, signalent que la commande a reconnu la pression exercée sur le bouton. Ils s'éteignent lorsque le seuil est franchi.

Si les boutons S1 ou S2 sont à nouveau activés lors du passage dans l'écluse, cela n'a aucun effet sur le processus. La demande est stockée puis traitée. Lors de la programmation, il est nécessaire de respecter la sécurité nécessaire contre la rupture de fil. ^[19] _

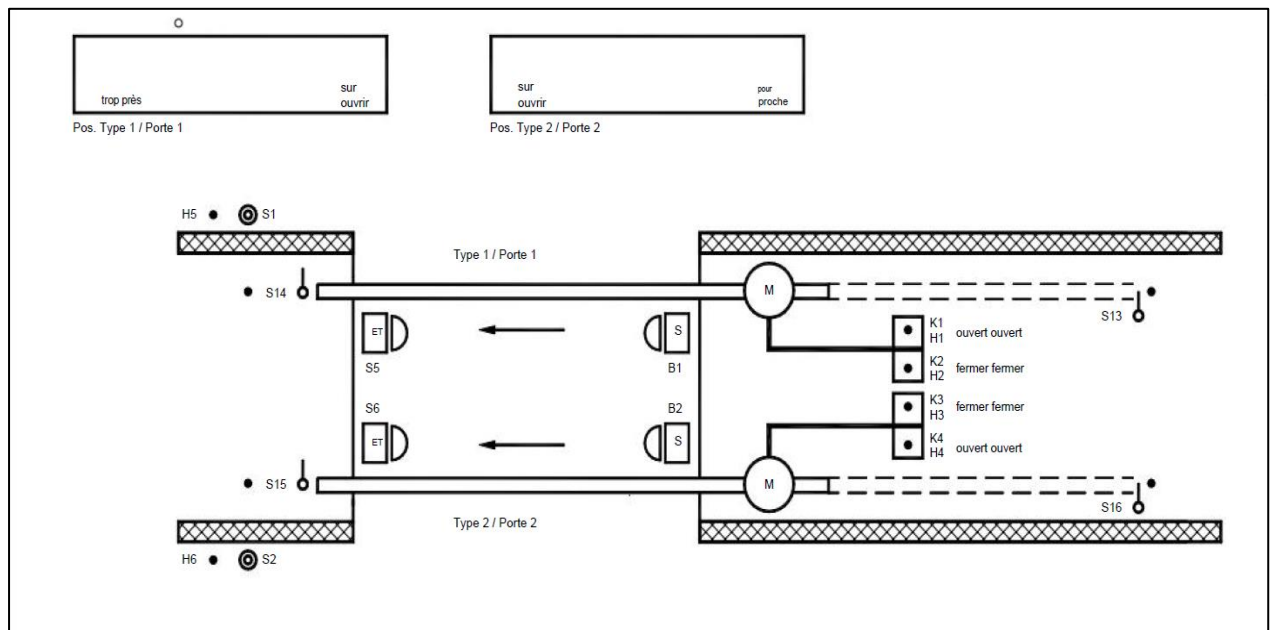


Figure 3-15 : Illustration du système de commande de deux portes serrures [19]

- *Analyse approfondie et détaillée du cahier des charges - une explication complète des exigences et spécifications :*

Ce processus assure que les portes coulissantes fonctionnent de manière sécurisée et efficace, en permettant aux personnes de passer à travers l'écluse tout en maintenant les portes dans un état approprié pour empêcher la poussière d'entrer.

✚ *Activation du système :*

Pour déverrouiller les portes, l'un des boutons de demande, S1 ou S2, doit être activé. Il n'y a pas de boutons de sécurité à l'intérieur de la serrure pour ouvrir les portes depuis l'intérieur. Par conséquent, il est essentiel de s'assurer que les deux portes ne restent jamais fermées en même temps.

✚ *Passage par la porte 1 de l'extérieur vers l'intérieur :*

Si une personne souhaite passer de l'extérieur vers l'intérieur par la porte 1, qui est fermée, le bouton S1 est activé. La porte 2 se ferme immédiatement et la porte 1 s'ouvre dès que l'interrupteur de fin de course S15 indique que la porte 2 est fermée.

✚ Passage à travers le seuil :

Une fois que la personne a traversé le seuil, une barrière lumineuse envoie une impulsion. Après un délai de 3 secondes, la porte 1 se ferme. Dès que l'interrupteur de fin de course S14 indique que la porte 1 est fermée, la porte 2 s'ouvre.

✚ Processus inverse pour un mouvement dans le sens opposé :

Le processus décrit ci-dessus est également valable pour un mouvement de l'intérieur vers l'extérieur. Les portes se comportent de la même manière, mais avec des interrupteurs de fin de course différents.

✚ Réaction aux seuils non sollicités :

Si le seuil est franchi sans action préalable sur les boutons S1 ou S2, le système de verrouillage ne réagit pas. Cela évite une fermeture ou une ouverture non intentionnelle des portes.

✚ Fonctionnement avec la barrière lumineuse :

Chaque entrée de l'écluse est équipée d'une barrière lumineuse. Tant que la barrière lumineuse n'est pas interrompue, les portes ne doivent pas se fermer. Si la barrière lumineuse est interrompue pendant le processus de fermeture, le processus s'arrête immédiatement et reprend instantanément une fois que la barrière lumineuse est activée.

✚ Voyants lumineux :

Des voyants lumineux situés à côté des boutons de demande signalent que la commande a détecté la pression exercée sur le bouton. Les voyants s'éteignent une fois que le seuil est franchi.

✚ Gestion des demandes répétées :

Si les boutons S1 ou S2 sont activés à nouveau pendant le passage dans l'écluse, cela n'a aucun effet sur le processus en cours. La demande est stockée et traitée ultérieurement selon l'ordre d'arrivée.

Visualisation graphique des divers comportements de système via le Grafset :

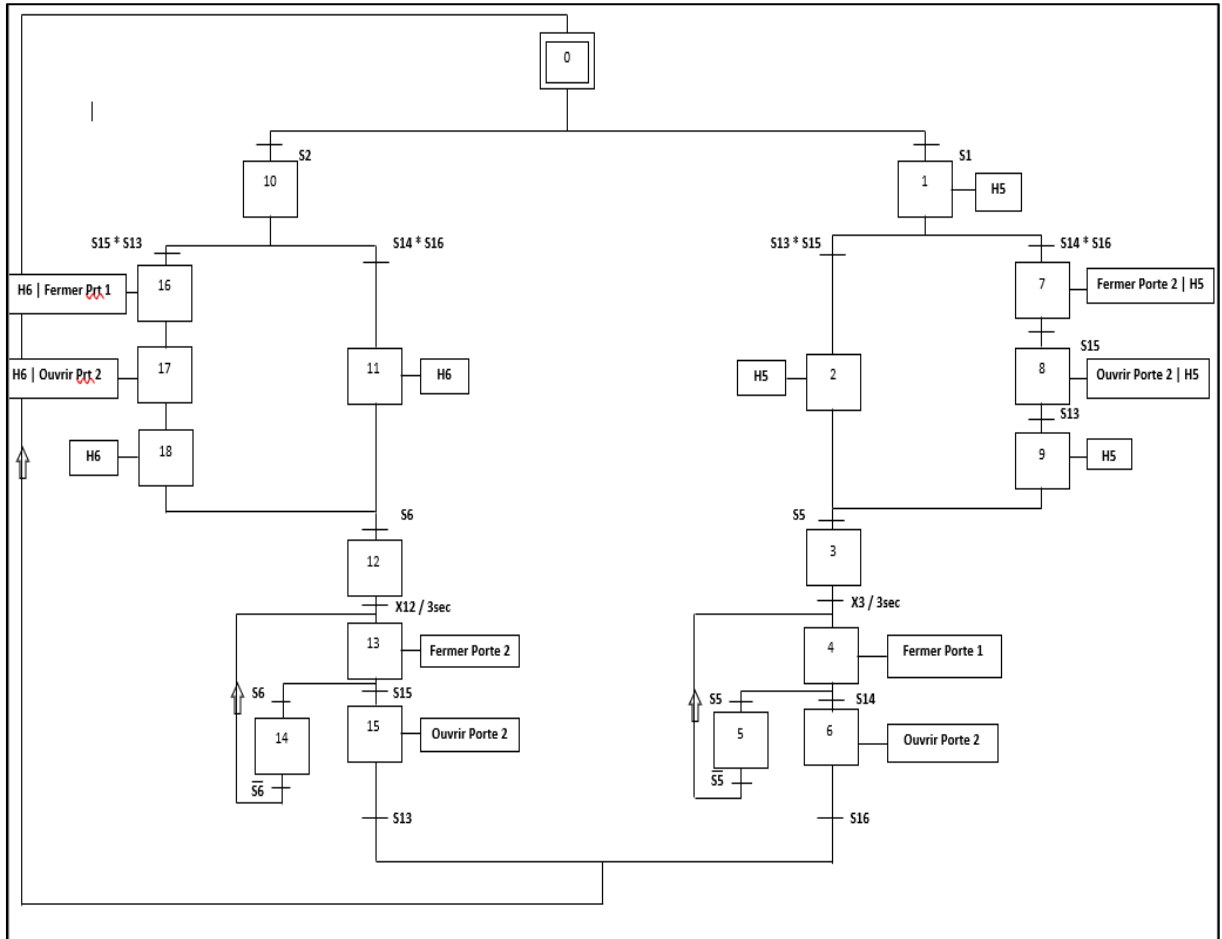


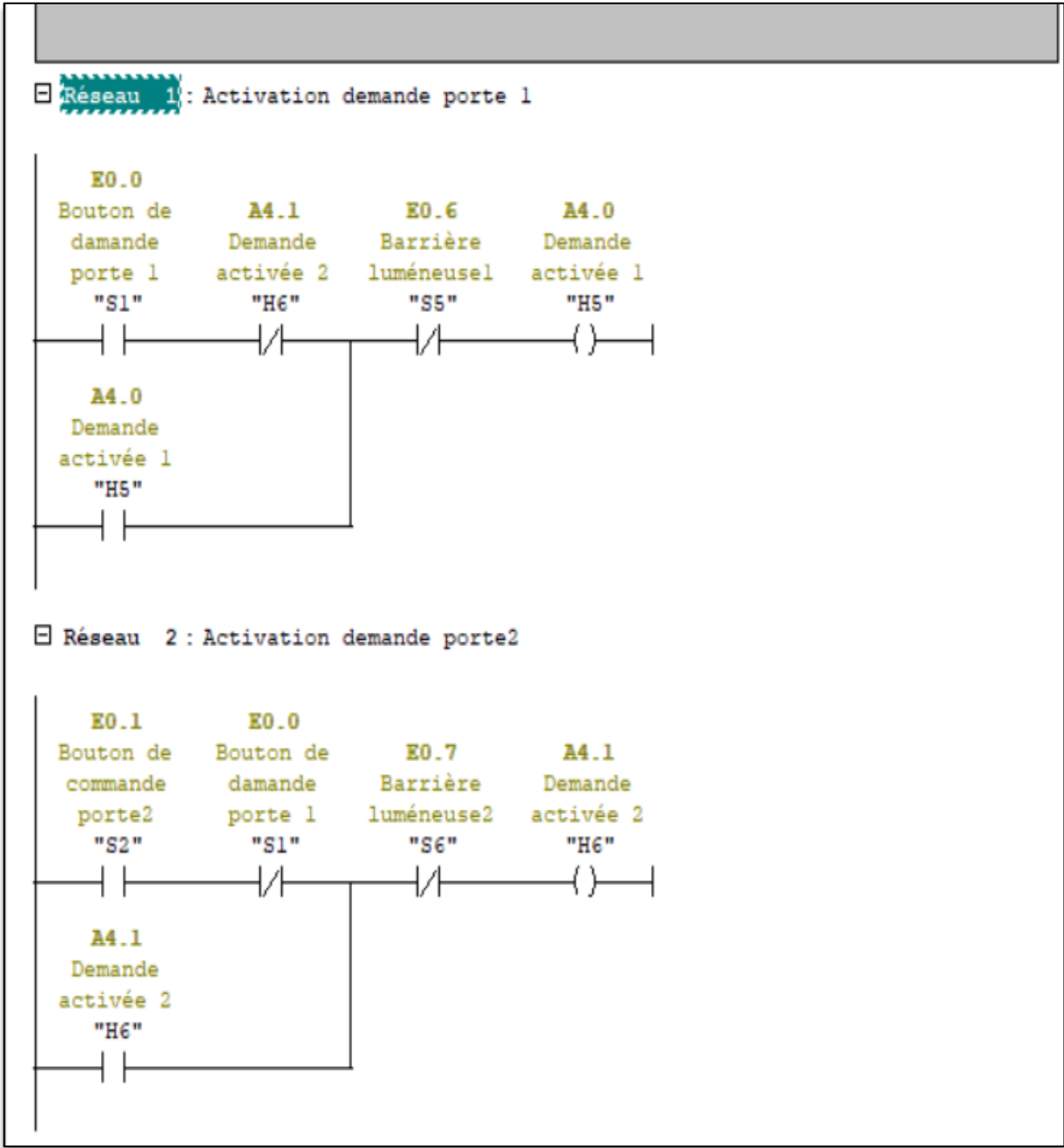
Figure 3-16 : Conception d'un système de commande de deux portes de serrures

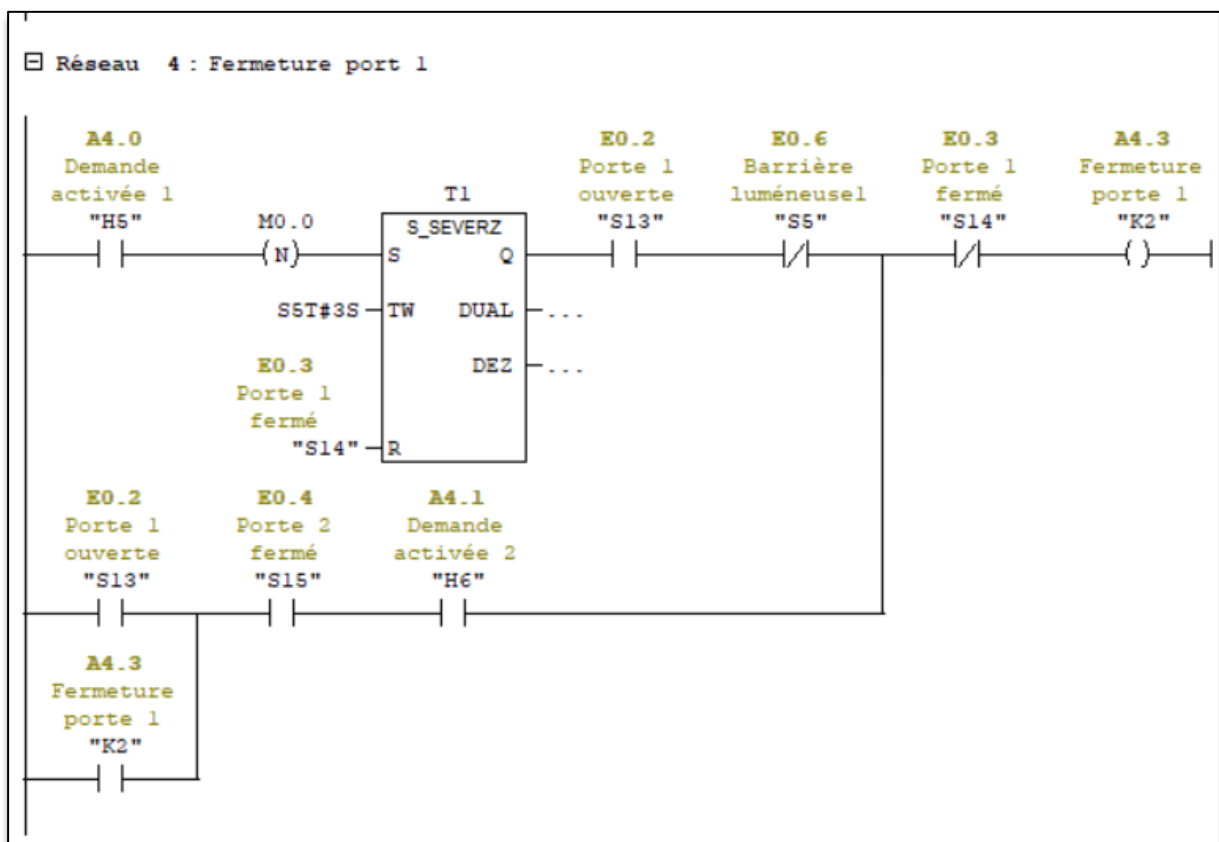
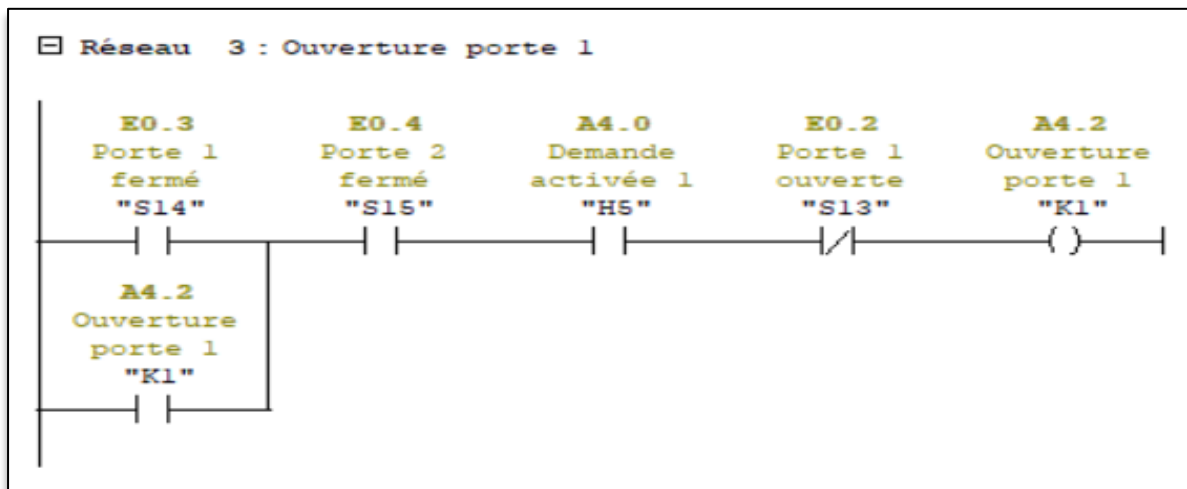
❖ *Développement et programmation du système via Simatic Step 7 :*

The screenshot shows the 'Editeur de mnémoniques' window in Simatic Manager. The title bar reads 'Editeur de mnémoniques - [Programme S7(1) (Mnémoniques) -- Deux portes\Station SIMAT]'. The menu bar includes 'Table', 'Edition', 'Insertion', 'Affichage', 'Outils', and 'Fenêtre ?'. The toolbar contains icons for file operations and a search box containing 'Tous les mnémoniques'. Below the toolbar is a table with the following data:

	Etat	Mnémonique /	Opérande	Type de do	Commentaire
1		H5	A 4.0	BOOL	Demande activée 1
2		H6	A 4.1	BOOL	Demande activée 2
3		K1	A 4.2	BOOL	Ouverture porte 1
4		K2	A 4.3	BOOL	Fermeture porte 1
5		K3	A 4.4	BOOL	Fermeture porte 2
6		K4	A 4.5	BOOL	Ouverture porte 2
7		S1	E 0.0	BOOL	Bouton de demande porte 1
8		S13	E 0.2	BOOL	Porte 1 ouverte
9		S14	E 0.3	BOOL	Porte 1 fermé
10		S15	E 0.4	BOOL	Porte 2 fermé
11		S16	E 0.5	BOOL	Porte2 ouverte
12		S2	E 0.1	BOOL	Bouton de commande porte2
13		S5	E 0.6	BOOL	Barrière lumineuse1
14		S6	E 0.7	BOOL	Barrière lumineuse2
15					

Figure 3-17 : Tableau de mnémoniques du système de commande de deux portes serrures





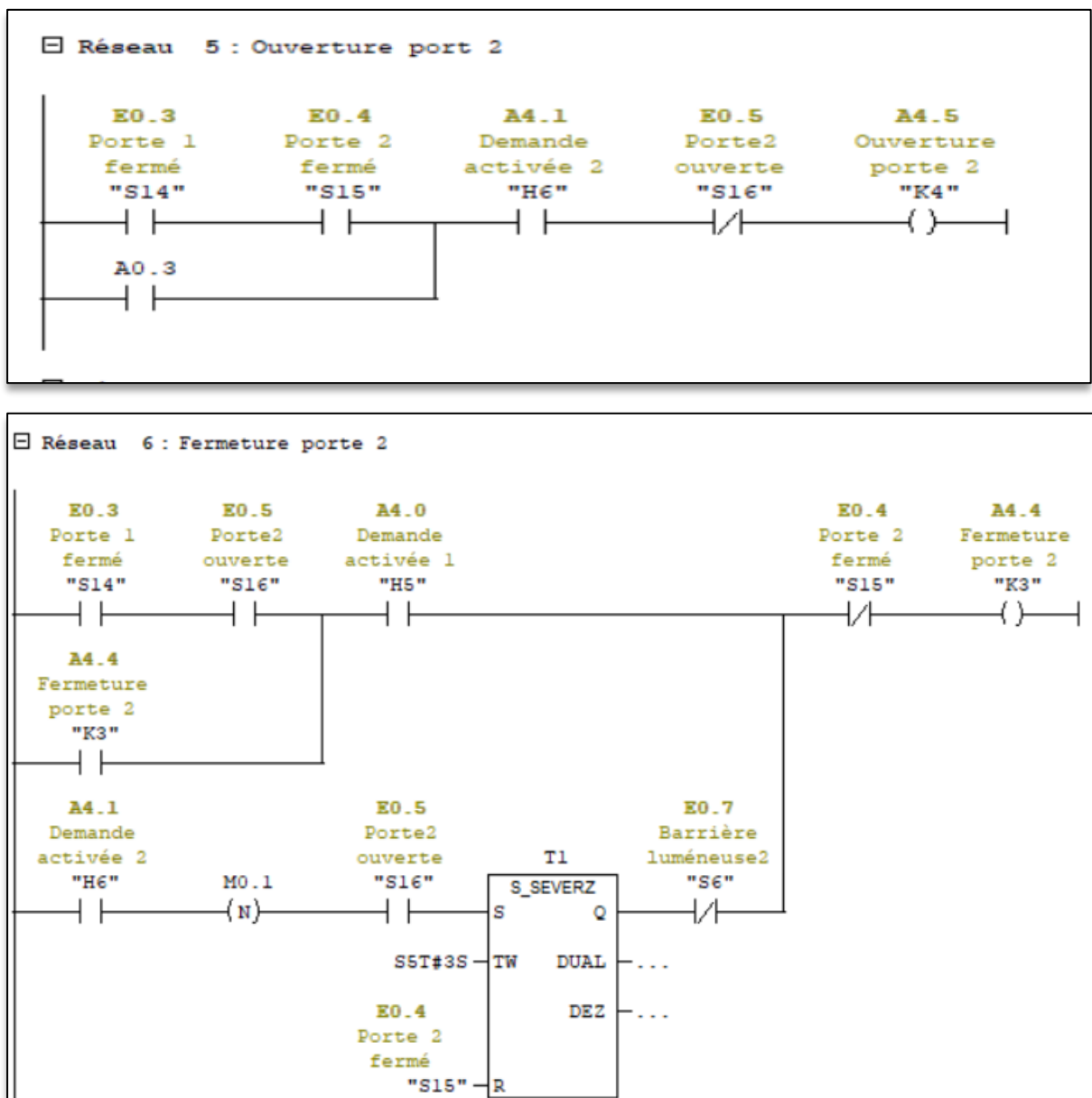


Figure 3-18 : Programmation du système de commande de deux portes serrures

1.2. Système de réseau d'air comprimé :

Tache requise et cahier de charge :

Un réseau d'air comprimé est alimenté par son réservoir sous pression, qui reçoit sa surpression de deux compresseurs

Description fonctionnelle :

L'activation du bouton de démarrage S1 ouvre les électrovannes Y1 et Y2. En même temps, les moteurs de compression M1 et M2 démarrent et pompent de l'air comprimé dans le réservoir.

Les compresseurs fonctionnent à **70 %** de la capacité de pompage

Avec une pression de 80% de la pression maximale du conteneur, les compresseurs sont déconnectés et les vannes magnétiques sont fermées

L'intensité de charge du réseau d'air comprimé est simulée à l'aide du consommateur via le bouton rotatif « Evacuation ».

Si la pression du réservoir chute à 60 % de la pression maximale du réservoir, l'électrovanne Y1 s'ouvre et le compresseur M2 est activé avec une capacité de pompe de 70 %.

Si la pression continue à chuter jusqu'à une valeur de 50 %, l'électrovanne Y1 s'ouvre et le compresseur M1 avec une capacité de pompe de 70 % est ajouté.

Si la pression du réservoir dépasse la pression maximale (S19) ou tombe en dessous de la pression minimale (S20), tous les compresseurs doivent être arrêtés, toutes les électrovannes fermées et un message de défaut (H5) doit être affiché.

En cas de panne, l'appareil doit être éteint avec le bouton d'arrêt S2. Le défaut peut alors être réparé et le réseau d'air comprimé peut être redémarré avec le bouton de démarrage S1.

Pour éviter une invite d'alarme due à une pression très basse lors du démarrage, l'invite d'alarme doit être supprimée pendant 5 secondes après la mise sous tension.

En activant le bouton d'arrêt, tous les moteurs sont arrêtés et toutes les électrovannes sont fermées. [19]

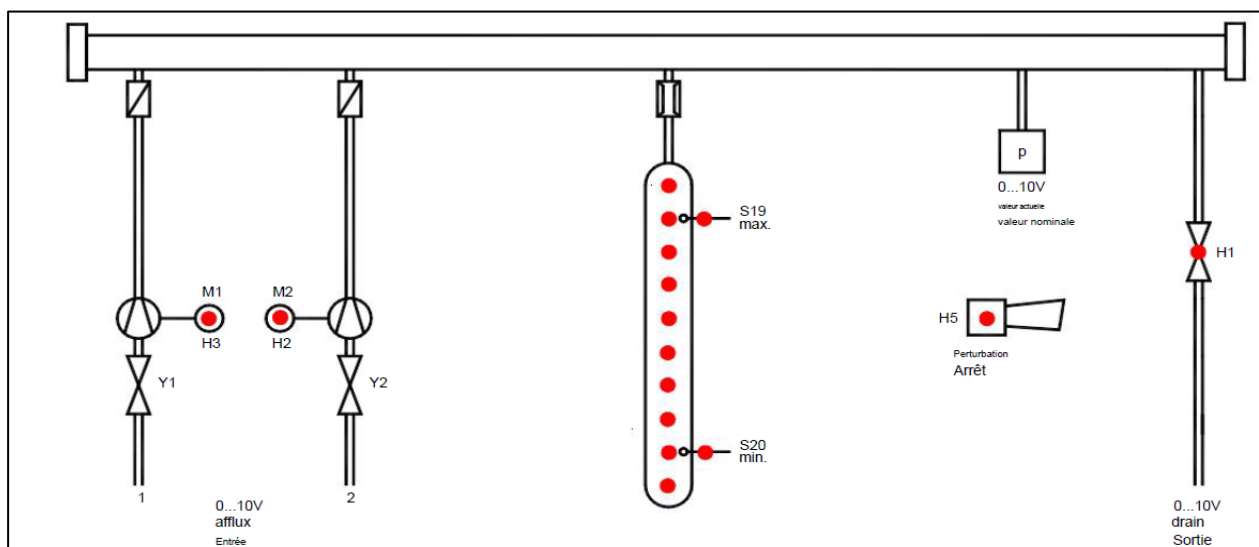


Figure 3-19 : Illustration du système du réseau d'air comprimé [19]

- *Analyse approfondie et détaillée du cahier des charges - une explication complète des exigences et spécifications :*

L'objectif de ce processus est de fournir un réseau d'air comprimé alimenté par un réservoir sous pression, en maintenant la pression dans des plages de fonctionnement sûres et en ajustant la production d'air comprimé en fonction de la demande simulée. Le système est également équipé de protections contre les surpressions et les baisses de pression, avec des fonctionnalités d'arrêt d'urgence en cas de panne ou de défaillance.

 **Activation du système :**

Lorsque le bouton de démarrage S1 est activé, les électrovannes Y1 et Y2 s'ouvrent. En même temps, les moteurs de compression M1 et M2 sont démarrés, ce qui permet de pomper de l'air comprimé dans le réservoir. Les compresseurs fonctionnent à 70% de leur capacité de pompage.

 **Contrôle de la pression maximale du réservoir :**

Lorsque la pression dans le réservoir atteint 80% de la pression maximale, les compresseurs sont déconnectés et les vannes magnétiques sont fermées pour empêcher toute surpression.

 **Simulation de la demande d'air comprimé :**

L'intensité de charge du réseau d'air comprimé est simulée en utilisant le bouton rotatif "Evacuation". Cela ajuste la consommation d'air pour simuler la demande du système.

 **Gestion de la pression basse :**

Si la pression dans le réservoir chute à 60% de la pression maximale, l'électrovanne Y1 s'ouvre et le compresseur M2 est activé à 70% de sa capacité de pompage pour compenser la baisse de pression. Si la pression continue de chuter jusqu'à atteindre 50%, l'électrovanne Y1 s'ouvre à nouveau et le compresseur M1 est ajouté à 70% de sa capacité de pompage.

 **Protection contre les seuils de pression :**

Si la pression du réservoir dépasse la pression maximale (S19) ou tombe en dessous de la pression minimale (S20), tous les compresseurs sont arrêtés, toutes les électrovannes sont fermées et un message de défaut (H5) est affiché pour indiquer l'anomalie.

✚ Arrêt du système en cas de panne :

En cas de panne, l'appareil peut être éteint à l'aide du bouton d'arrêt S2. Une fois la panne réparée, le réseau d'air comprimé peut être redémarré en utilisant le bouton de démarrage S1.

✚ Suppression de l'invite d'alarme au démarrage :

Pour éviter une alarme due à une pression très basse lors du démarrage, l'invite d'alarme est supprimée pendant 5 secondes après la mise sous tension. Cela permet au système de se stabiliser avant de déclencher une alarme éventuelle.

✚ Visualisation graphique des divers comportements de système via le Grafcet :

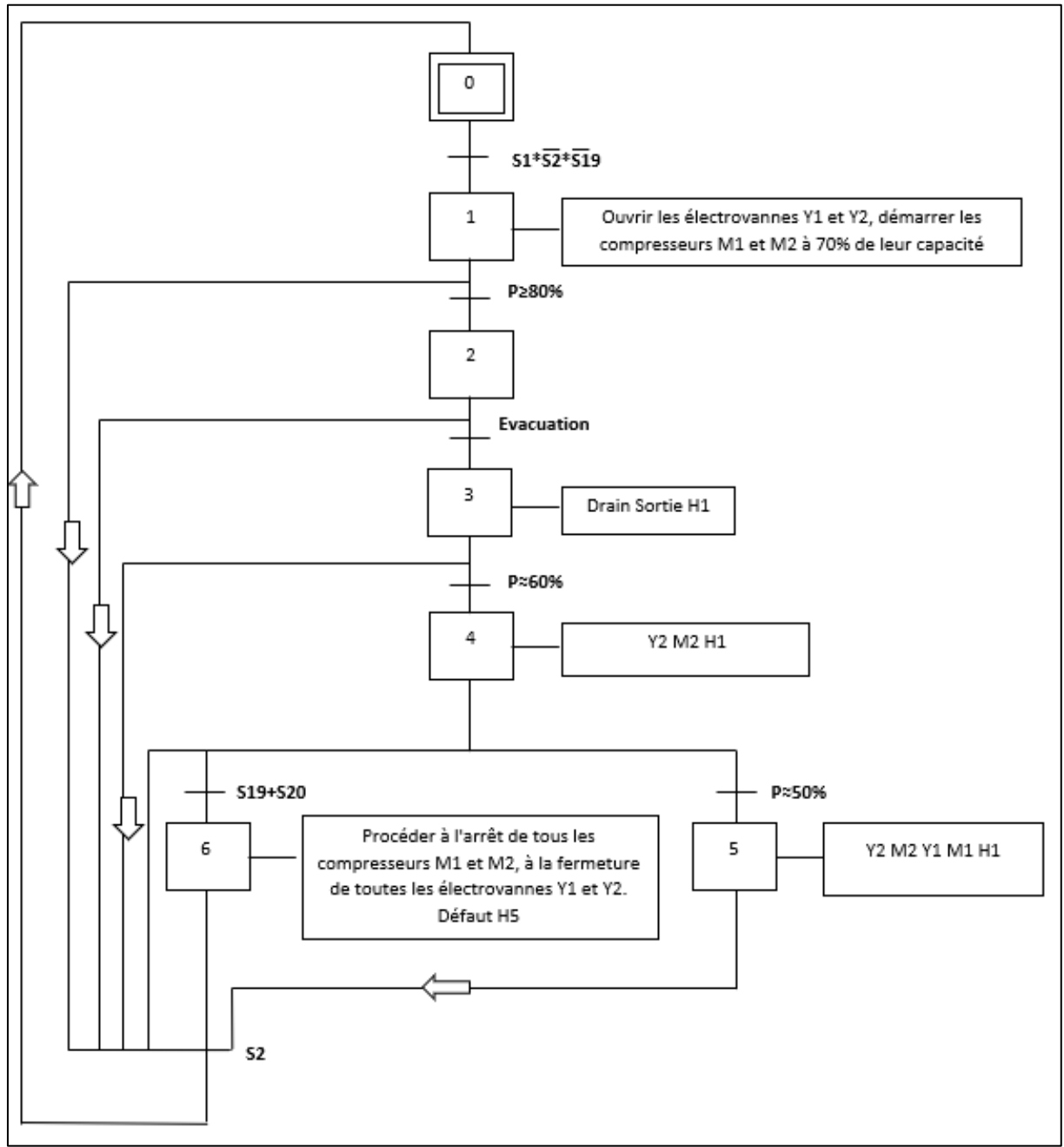
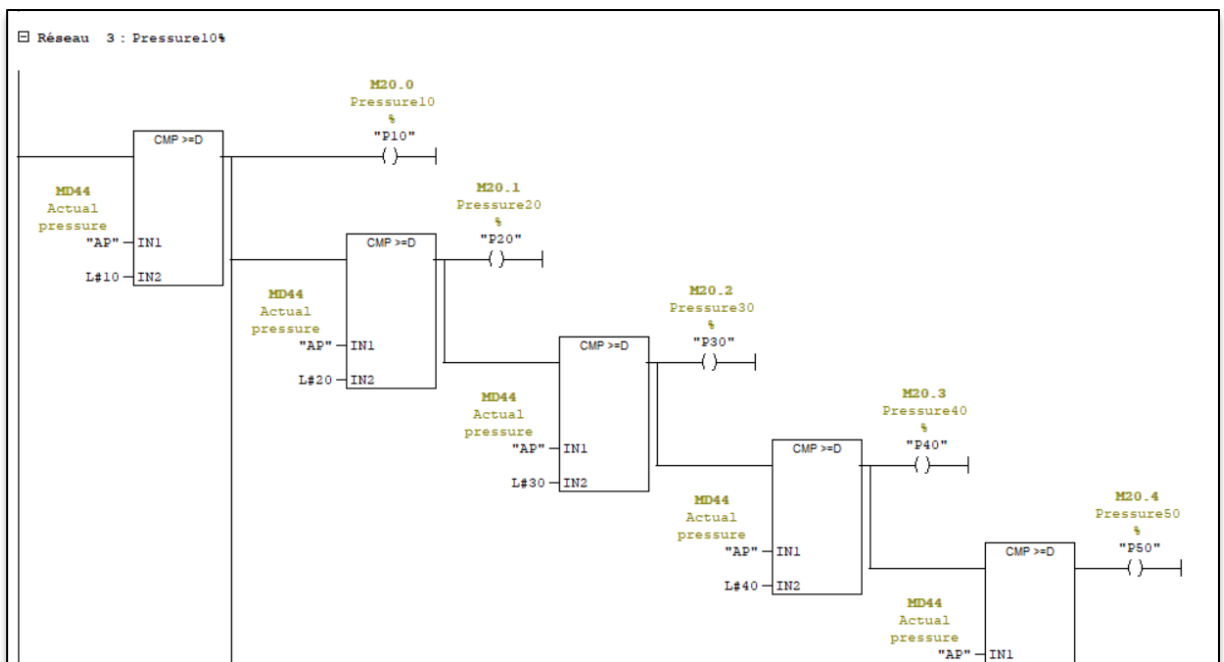
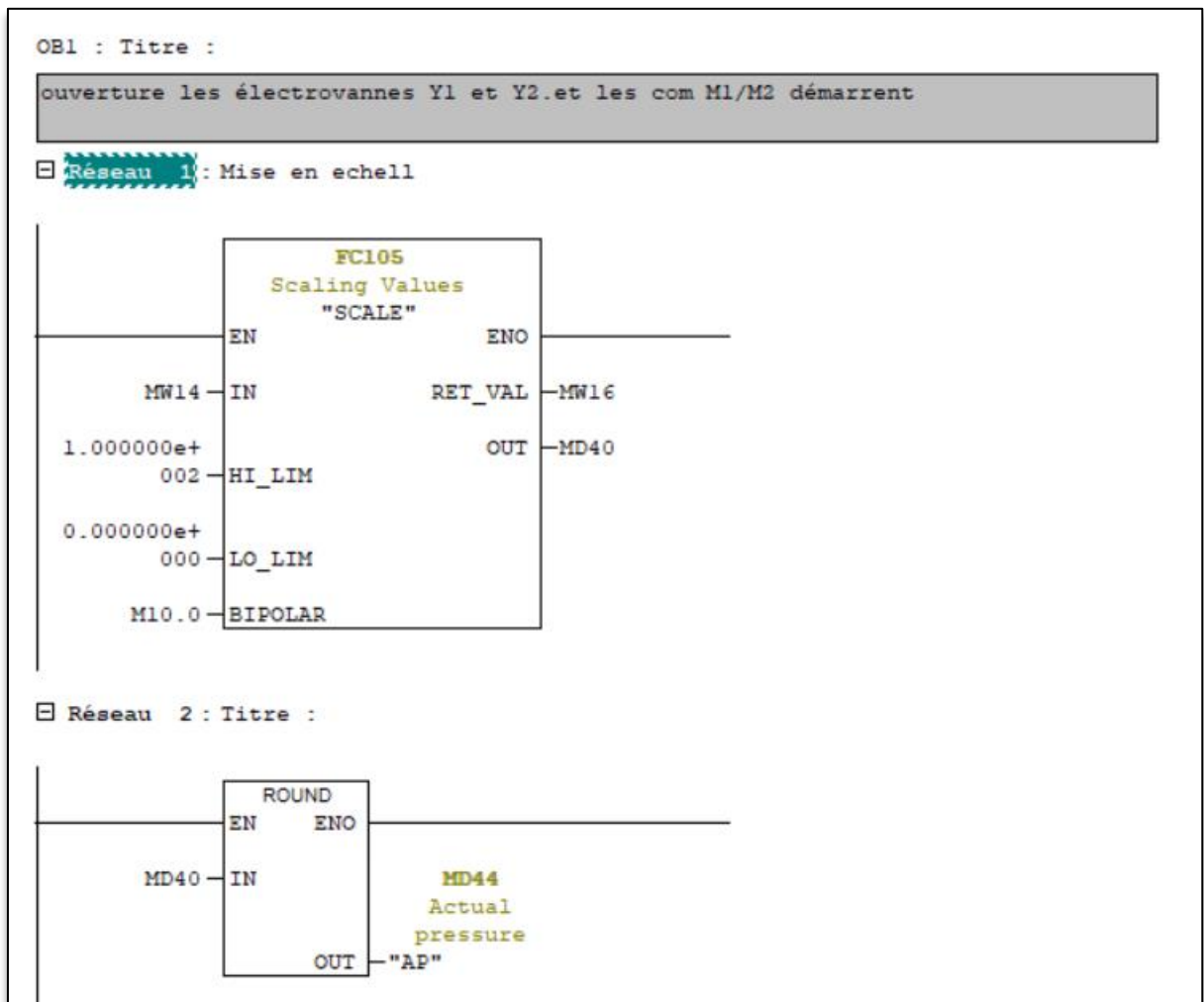


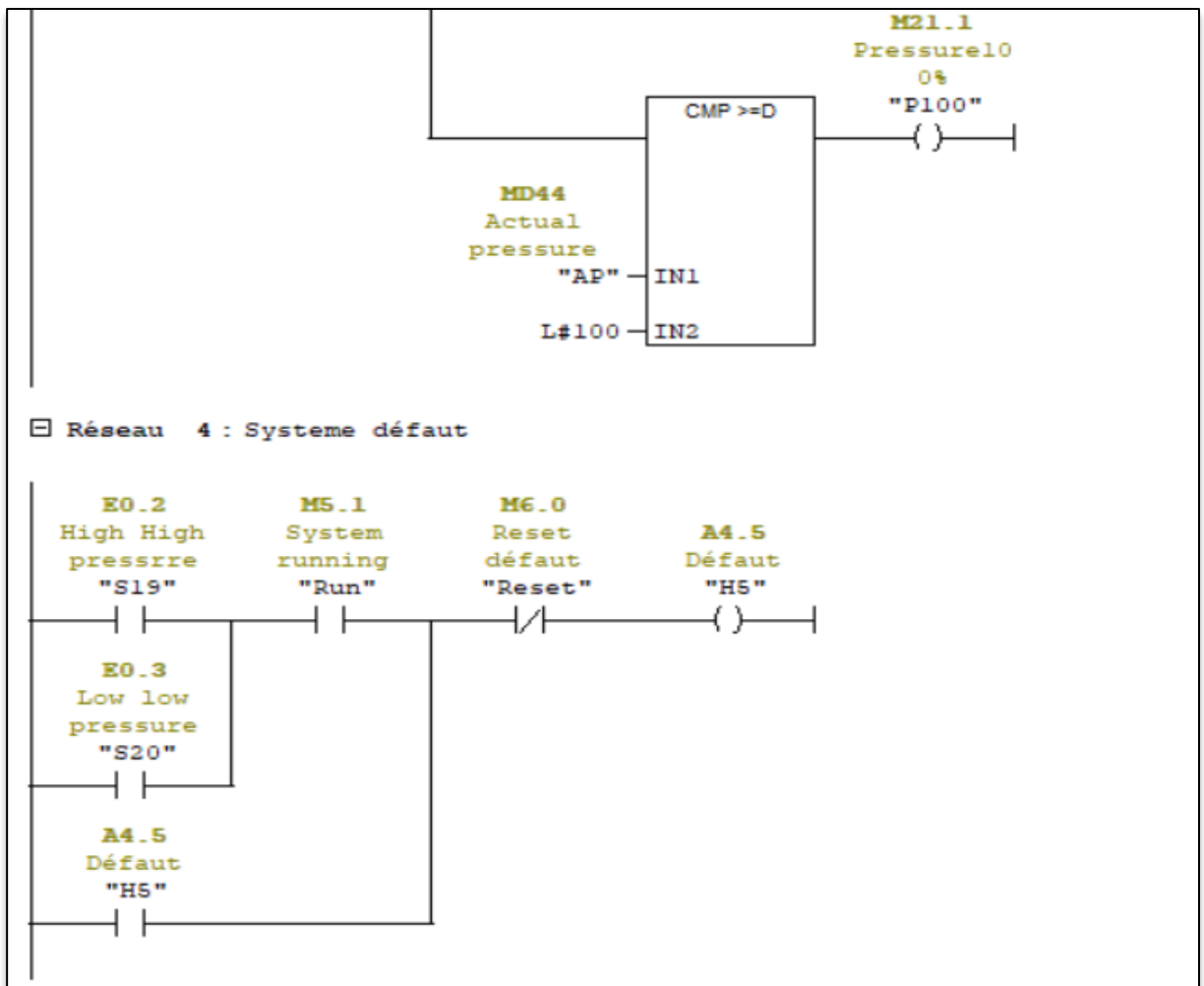
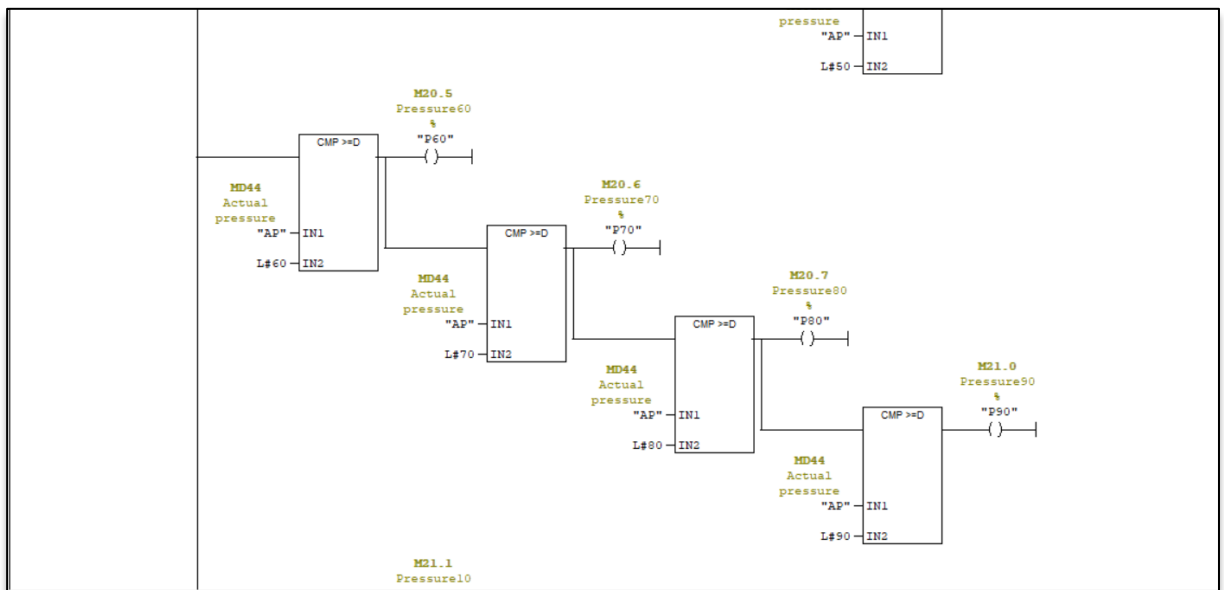
Figure 3-20 : Conception d'un système de réseau d'air comprimé

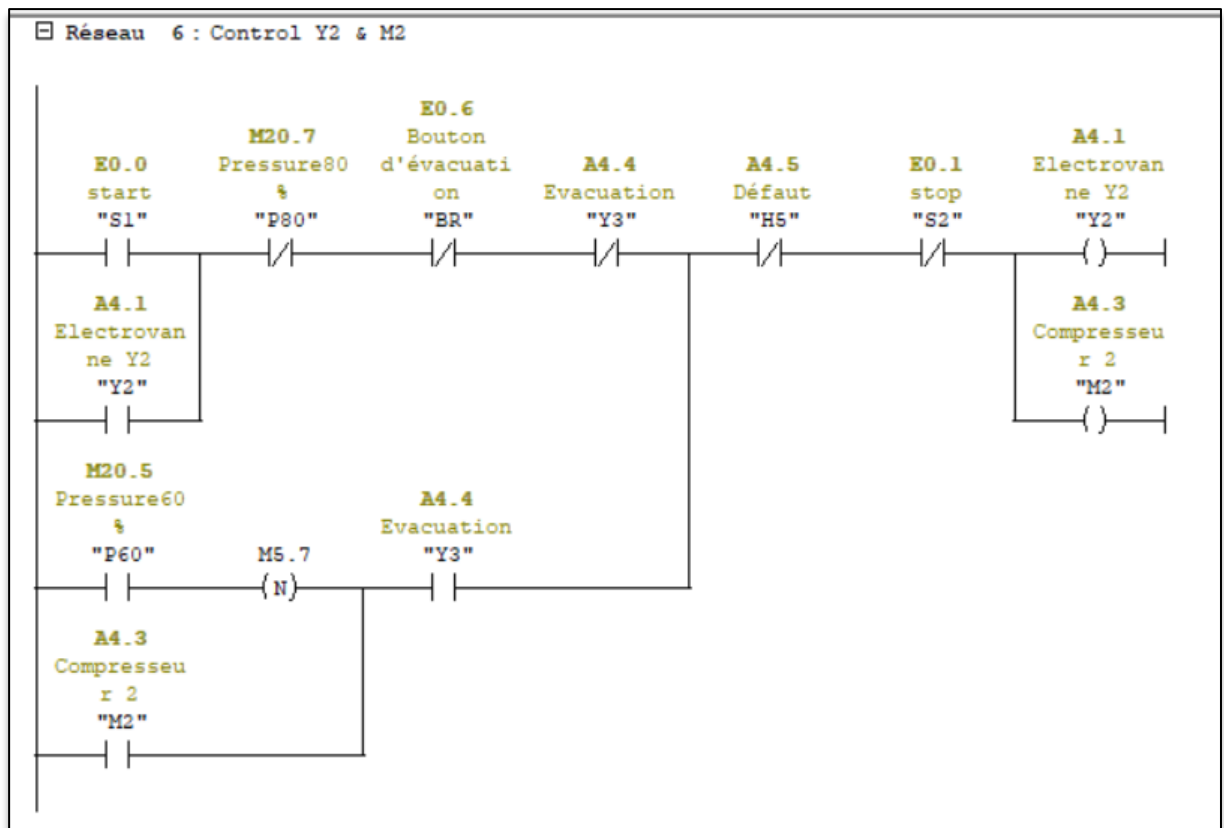
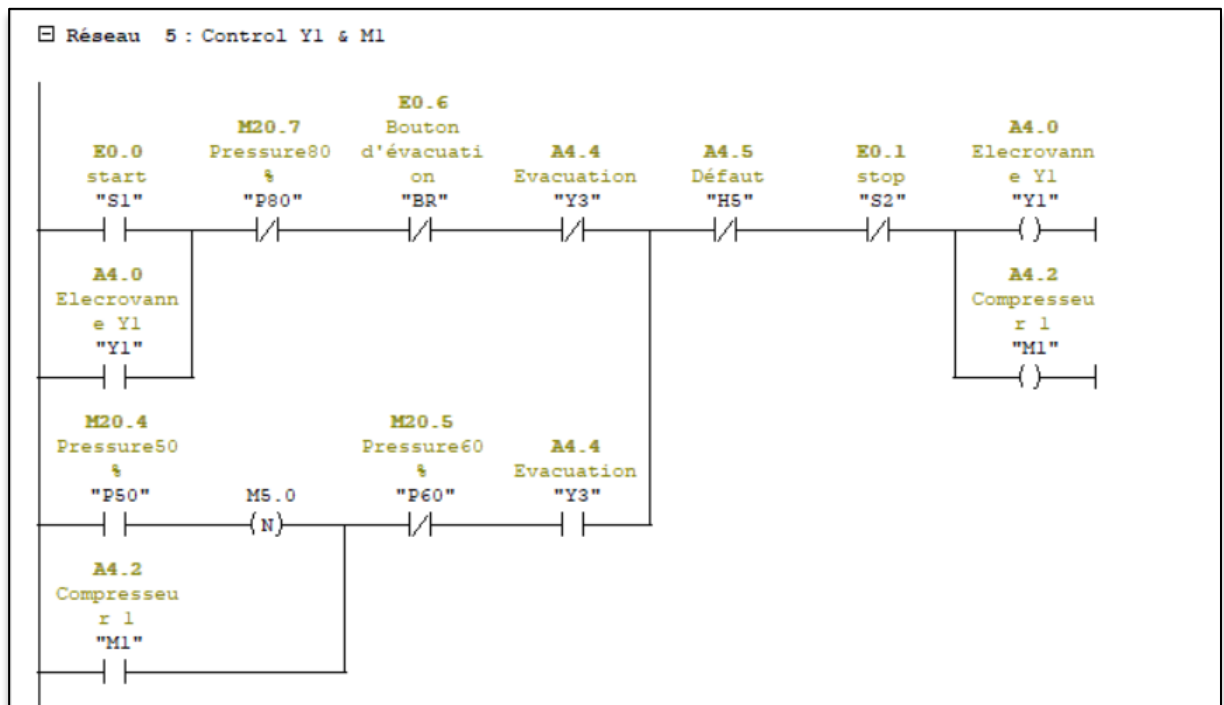
❖ *Développement et programmation du système via Simatic Step 7 :*

	Etat	Mnémonique /	Opérande	Type de do	Commentaire
1		AP	MD 44	DWORD	Actual pressure
2		BR	E 0.6	BOOL	Bouton d'évacuation
3		H5	A 4.5	BOOL	Défaut
4		M1	A 4.2	BOOL	Compresseur 1
5		M2	A 4.3	BOOL	Compresseur 2
6		P10	M 20.0	BOOL	Pressure10%
7		P100	M 21.1	BOOL	Pressure100%
8		P20	M 20.1	BOOL	Pressure20%
9		P30	M 20.2	BOOL	Pressure30%
10		P40	M 20.3	BOOL	Pressure40%
11		P50	M 20.4	BOOL	Pressure50%
12		P60	M 20.5	BOOL	Pressure60%
13		P70	M 20.6	BOOL	Pressure70%
14		P80	M 20.7	BOOL	Pressure80%
15		P90	M 21.0	BOOL	Pressure90%
16		Reset	M 6.0	BOOL	Reset défaut
17		Run	M 5.1	BOOL	System running
18		S1	E 0.0	BOOL	start
19		S19	E 0.2	BOOL	High High pressrre
20		S2	E 0.1	BOOL	stop
21		S20	E 0.3	BOOL	Low low pressure
22		S9	M 0.0	BOOL	Seuil max
23		SCALE	FC 105	FC 105	Scaling Values
24		Str	M 5.5	BOOL	First start
25		Y1	A 4.0	BOOL	Electrovanne Y1
26		Y2	A 4.1	BOOL	Electrovanne Y2
27		Y3	A 4.4	BOOL	Evacuation
28					

Figure 3-21 : Tableau de mnémoniques du système de réseau d'air comprimé







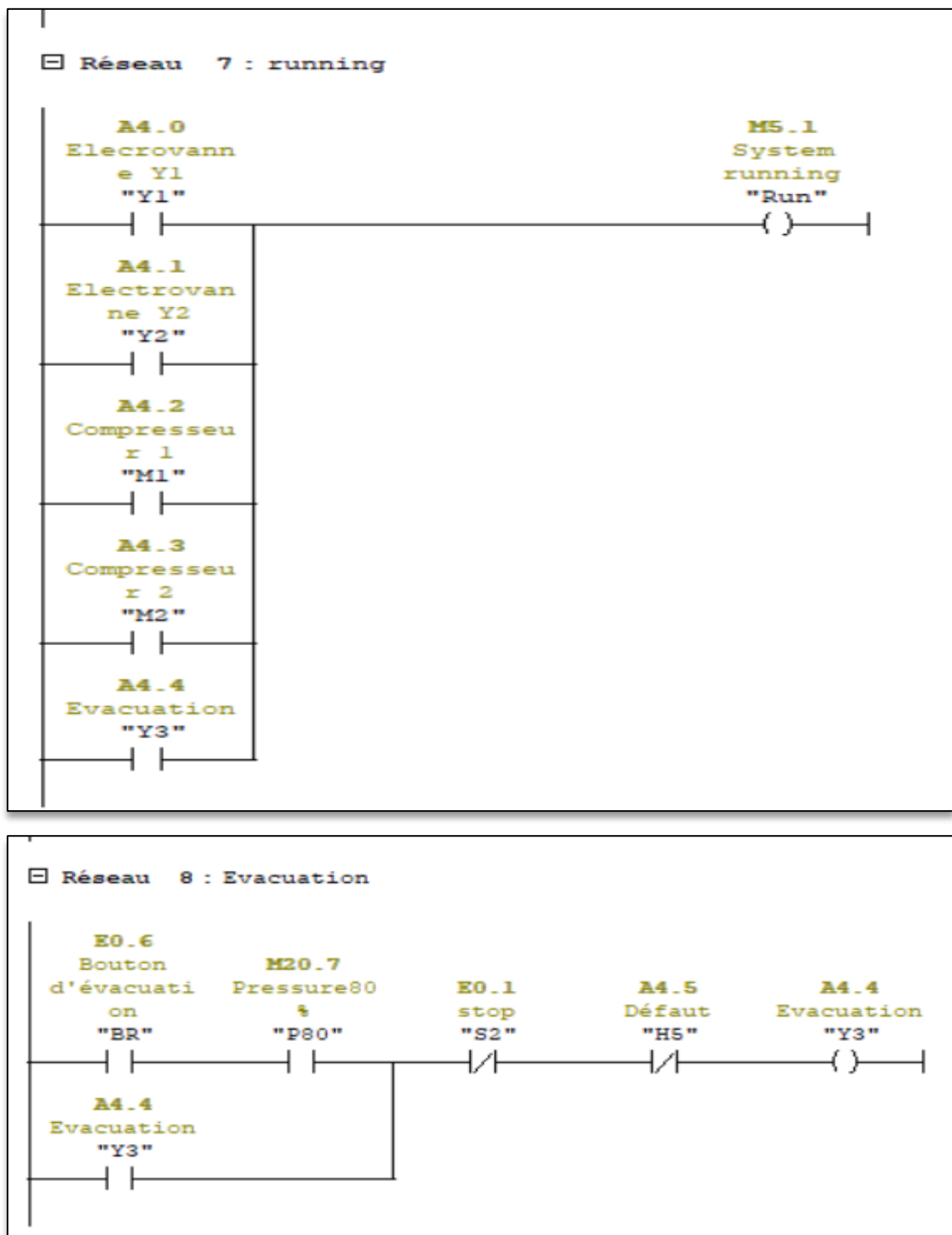


Figure 3-22 : Programmation du système de réseau d'air comprimé

Conclusion :

En conclusion de ce chapitre nous avons exploré trois systèmes clés dans le domaine industriel.

Le premier système abordé était celui du remplissage contrôlé, où nous avons étudié comment utiliser Simatic Step 7 pour programmer un processus de remplissage précis et automatisé. Grâce à cette approche avancée, nous avons pu développer des stratégies de contrôle efficaces pour assurer un remplissage optimal, réduisant ainsi les erreurs humaines et augmentant l'efficacité de la production.

Le deuxième système que nous avons examiné était la commande de deux portes sécurisées. Nous avons pu utiliser Simatic Step 7 pour mettre en œuvre des algorithmes de sécurité sophistiqués afin de contrôler l'accès aux zones sensibles. Grâce à cette approche avancée, nous avons pu garantir la sécurité des opérateurs et minimiser les risques d'accidents liés à l'utilisation des portes.

Enfin, nous avons abordé le système de réseau d'air comprimé. En utilisant Simatic Step 7, nous avons pu développer une solution robuste pour contrôler et superviser efficacement les compresseurs d'air et les vannes du réseau. Cette approche avancée nous a permis d'optimiser les performances du réseau d'air comprimé, d'améliorer la fiabilité de l'alimentation en air et de réduire les coûts d'exploitation.

Chapitre IV :

Développement de l'interface avec le WinCC

Introduction

1. Le Wincc Explorer
 - 1.1. SIMATIC HMI
 - 1.2. Présentation du logiciel Wincc Explorer
 - 1.3. Création d'un nouveau projet
 - 1.4. Variables dans le Wincc
 - 1.5. Concepteur graphique
 - 1.6. La liaison automate/HMI

2. Supervision et des systèmes industriels et réalisation de l'interfaces graphique avec le WinCC Explorer
 - 2.1. Système de remplissage contrôlé
 - 2.2. Système de commande de deux portes serrures
 - 2.3. Système de réseau d'air comprimé

Conclusion

Introduction :

À mesure que la complexité des processus s'accroît et que les machines et les installations doivent satisfaire à des exigences de fonctionnalité de plus en plus rigoureuses, l'opérateur est confronté à la nécessité d'obtenir un niveau d'informations maximal pour surveiller l'état actuel du système.

C'est à travers l'utilisation de l'Interface Homme-Machine (IHM) que ces informations peuvent être obtenues.

1. Le wincc explorer :

1.1. SIMATIC HMI :


Les pupitres SIMATIC HMI offrent une gamme complète de fonctionnalités hautement performantes.

Dotés d'écrans larges de résolution élevée allant de 4" à 22", avec la possibilité de choisir entre une commande tactile ou un clavier, ils s'adaptent de manière optimale à toutes les applications.

Parmi les nombreuses innovations par rapport aux pupitres SIMATIC existants, une fonctionnalité remarquable consiste à pouvoir coordonner et centraliser la mise en veille des écrans des appareils pendant les pauses grâce à PROF Energy, ce qui permet de réduire la consommation d'énergie de manière efficace. ^[21]

1.2. Exposition du logiciel WinCC explorer :

WinCC est un système modulaire. Il permet de visualiser le processus et de configurer un graphique interface utilisateur. Vous utiliserez l'interface utilisateur pour opérer et observer le processus. ^[21]

 *Fonctionnalités* : Le WinCC explorer offre, pour les applications au niveau machine (couvert jusqu'à présent par la famille Protocol), un considérable gain d'efficacité dans la configuration ainsi que des concepts d'automatisation innovateurs. Dans les secteurs proches du process, de la construction d'installations et de machines ainsi que de la construction de machines de série, SIMATIC WinCC explorer permet en outre ^[21] :

- D'améliorer la productivité (efficacité de la configuration) lors de la création de projets IHM de réaliser des concepts d'IHM et d'automatisation innovants dans le cadre de réseaux TCP/IP et du Web
- D'accroître la disponibilité des machines et installations par de nouveaux concepts de maintenance
- D'accéder facilement, en toute sécurité aux données de process à partir de n'importe quel endroit du globe

1.3. Création d'un nouveau projet :

- ✓ Démarrez WinCC explorer.
 - ✓ Créer un nouveau projet.
-
- L'assistant projet WinCC explorer est ouvert.
 - L'assistant projet vous aide au projet en vous conduisant étape par étape tout au long de la configuration.
 - Pour cela, l'assistant projet dispose de différents scénarios pour les configurations les plus courantes.
 - Vous effectuez votre configuration sur la base des scénarios sélectionnés. ^[21]

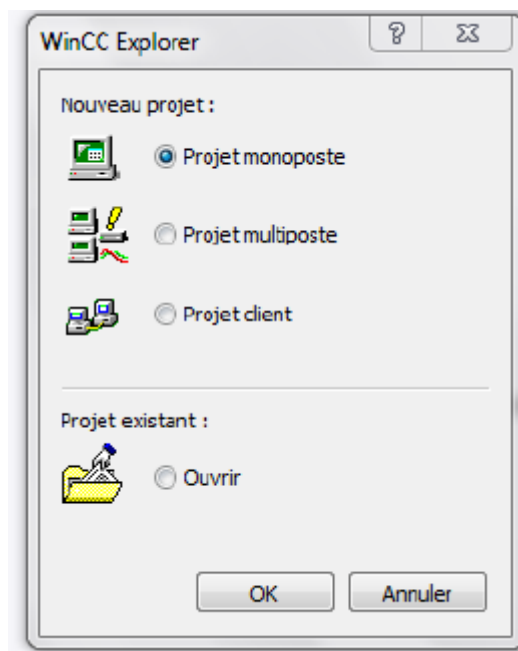


Figure 4-1 : Assistant de Wincc explorer.[21]

- Donner les informations du projet

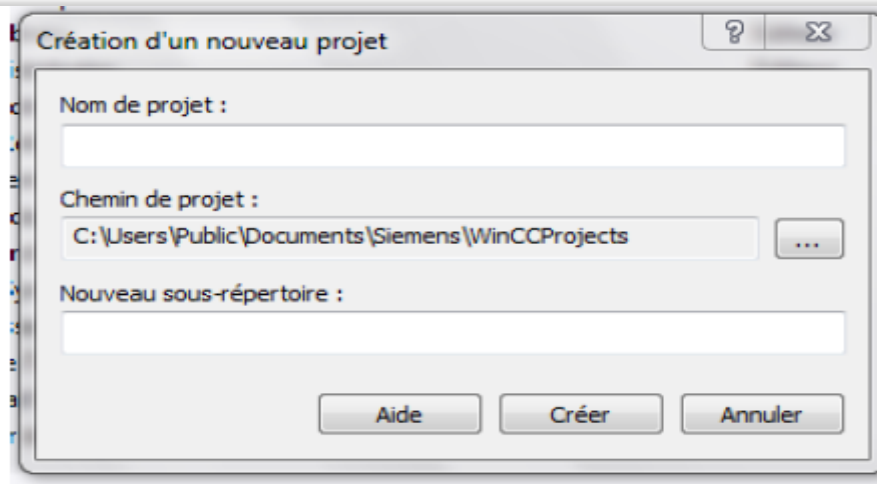


Figure 4-2 : Donner les informations du projet de wincc explorer.[21]

Vous avez créé le projet "Quick Start". Le projet est ouvert dans WinCC Explorer. La structure du projet avec les éditeurs et répertoires nécessaires est affichée dans le volet gauche d'Explorateur WinCC. Le volet de droite affiche les éléments appartenant à un éditeur ou à un répertoire.[21]

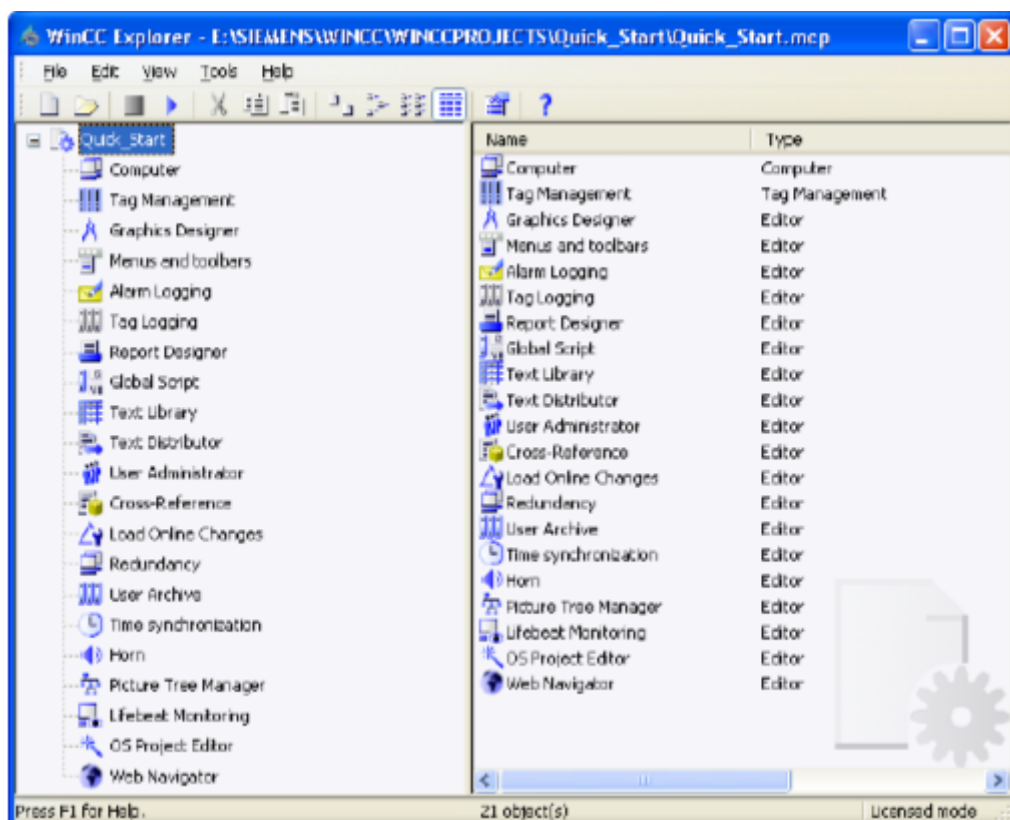


Figure 4-3 : Fenêtre du travail dans le Wincc explorer.[21]

1.4. Variables dans WinCC :

Les variables dans WinCC représentent soit des valeurs réelles, soit des valeurs internes. Les valeurs internes sont calculées ou simulées dans WinCC. WinCC gère toutes les variables dans le "Configuration Studio" éditeur.^[21]

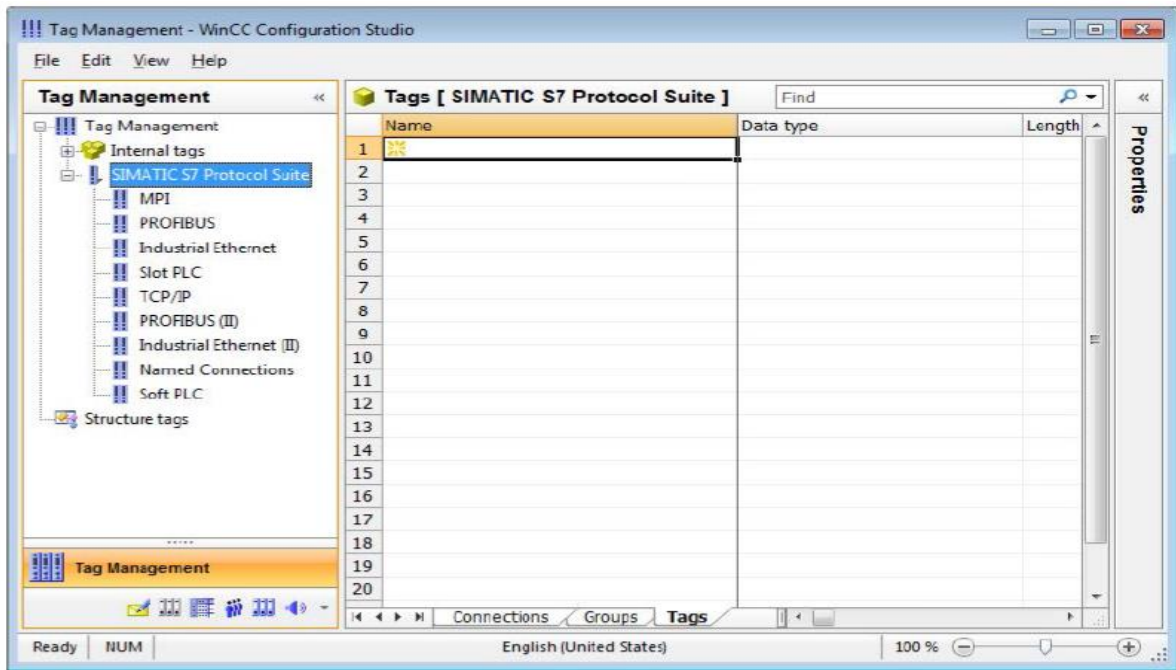


Figure 4-4 : Fenêtre du tag management dans le Wincc explorer.^[21]

1.5. Concepteur graphique :

Vous pouvez utiliser l'éditeur "Graphics Designer" pour configurer les vues de process. La structure de cet éditeur est similaire à un programme de dessin et fonctionne également de la même manière.

Le "Graphics Designer" met à disposition des objets et des outils pour configurer des vues de process. Pour le Projet "Quick Start" vous utilisez principalement la palette d'objets et la bibliothèque de l'éditeur "Graphics Designer".^[21]

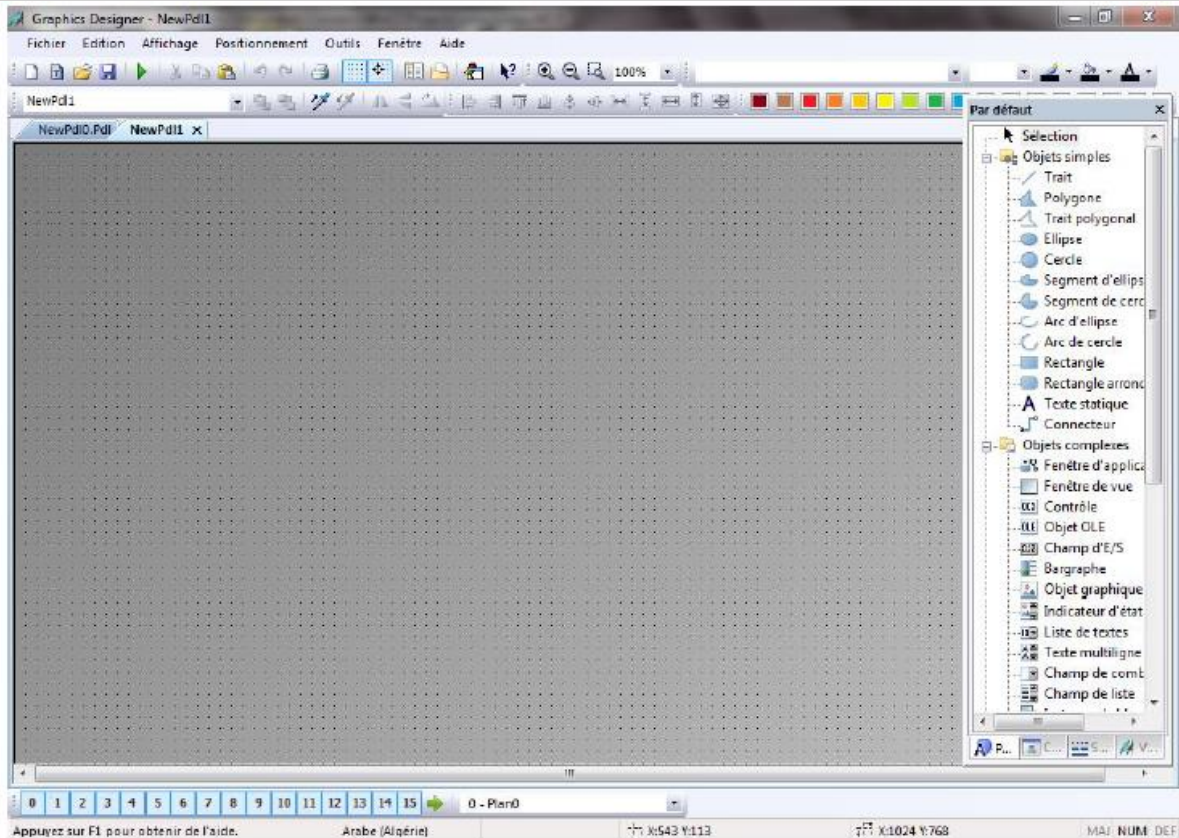


Figure 4-5 : Fenêtre du Graphics Designer dans le Wincc explorer.[21]

1.6. La liaison automate/IHM :

La liaison est établie en choisissant le protocole de communication par exemple l'interface multi points (MPI), Profibus ...Etc. [22]

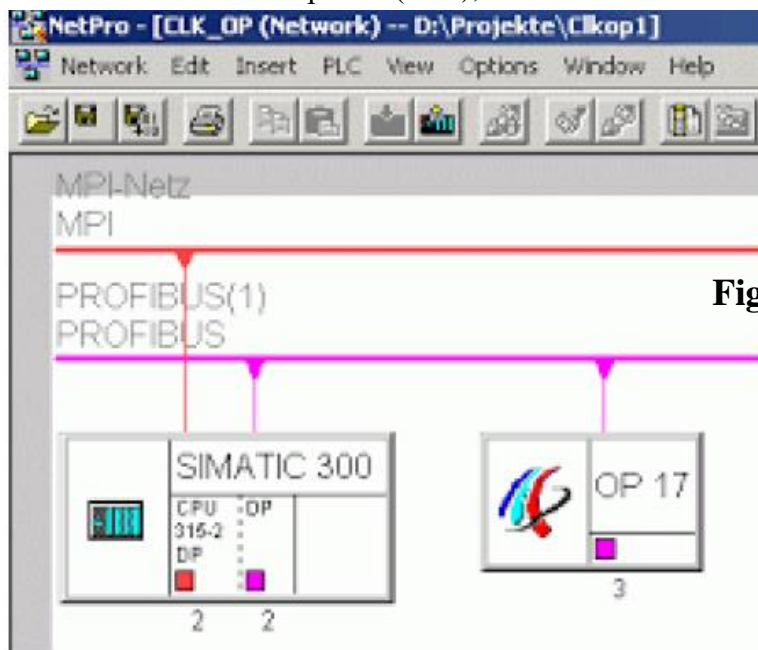


Figure 4-6 : liaison automate/IHM.[22]

2. Supervision et des systèmes industriels et réalisation de l'interfaces graphique avec le wincc explorer :

2.1. Système de remplissage contrôlé :

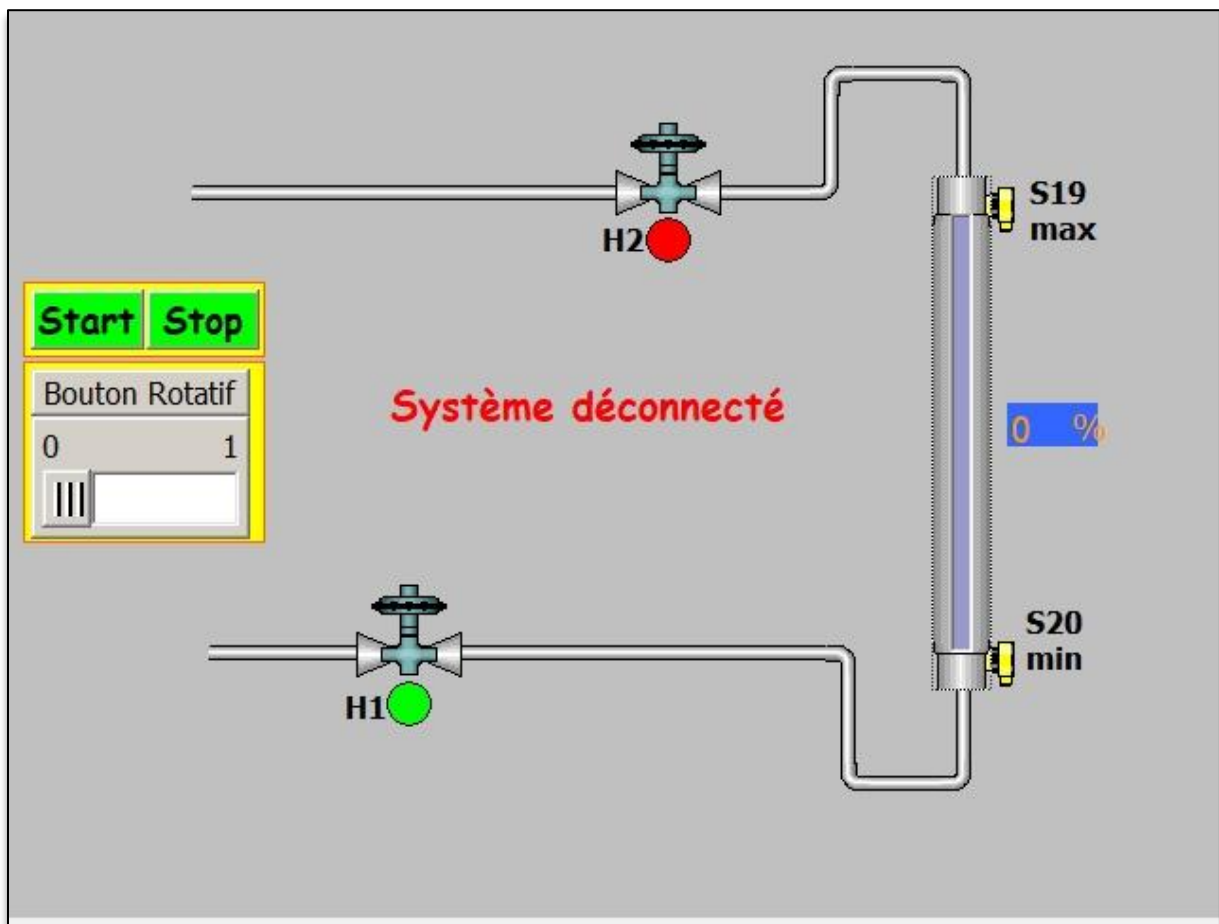


Figure 4-7 : Système de remplissage déconnecté

Le système démarre le processus, avec une ouverture d'électrovanne pour permettre l'entrée de fluide.

Il surveille les niveaux de liquide, arrête le processus si le niveau minimum est atteint et s'arrête automatiquement si le conteneur est trop rempli.

En utilisant le bouton rotatif "Vider", l'opérateur peut déclencher manuellement la vidange du récipient.

Si, lors de la vidange, le niveau du récipient atteint 40% de sa capacité, le système se met automatiquement en marche et commence à le remplir à nouveau.

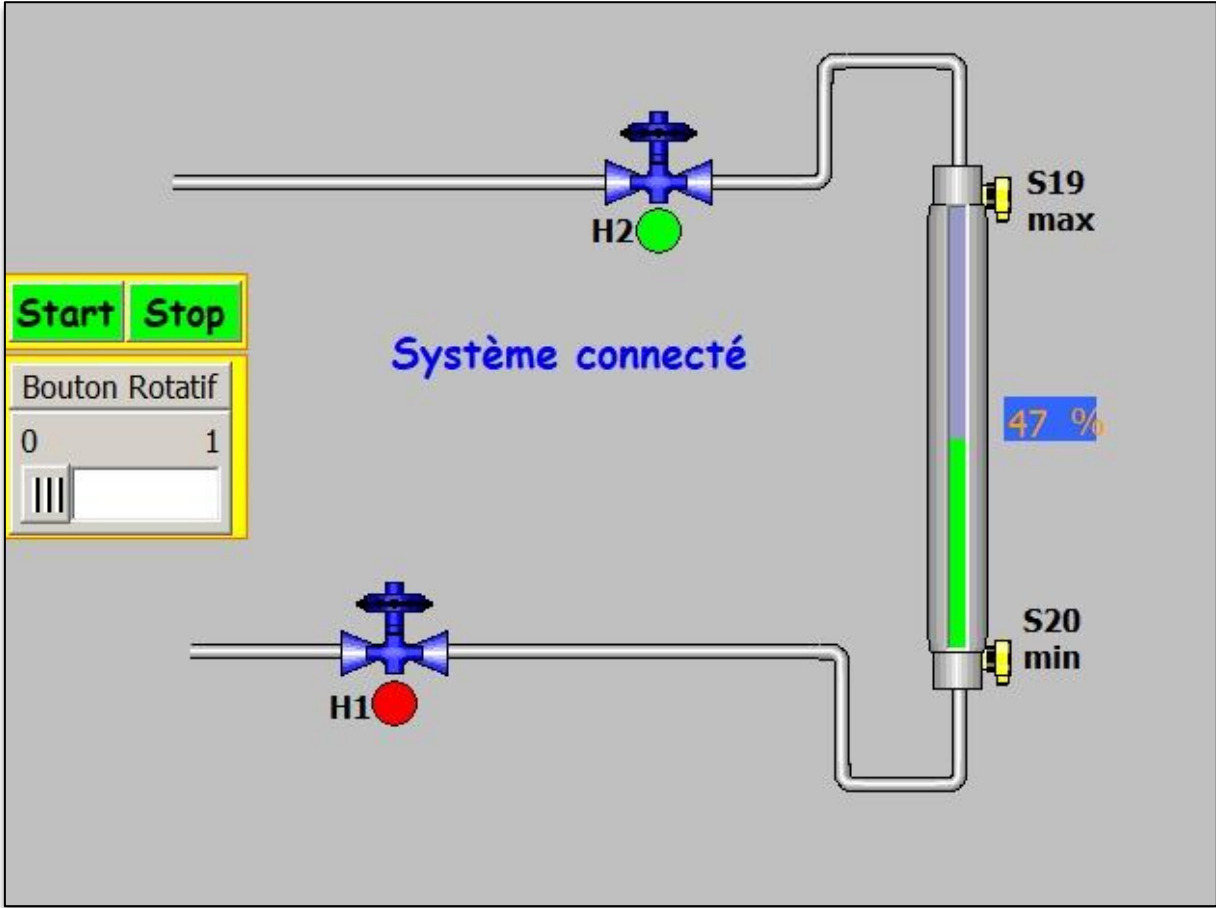


Figure 4-8 : Système de remplissage connecté

2.2. Système de commande de deux portes serrures :

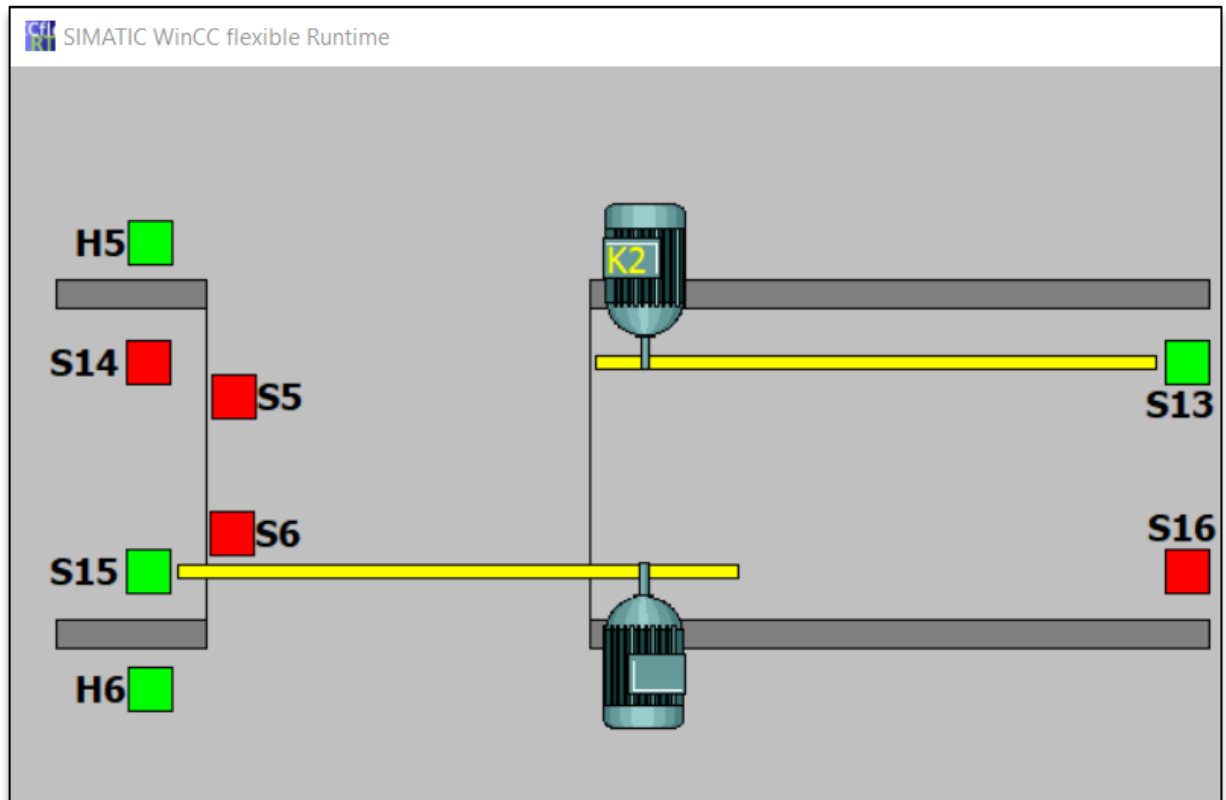
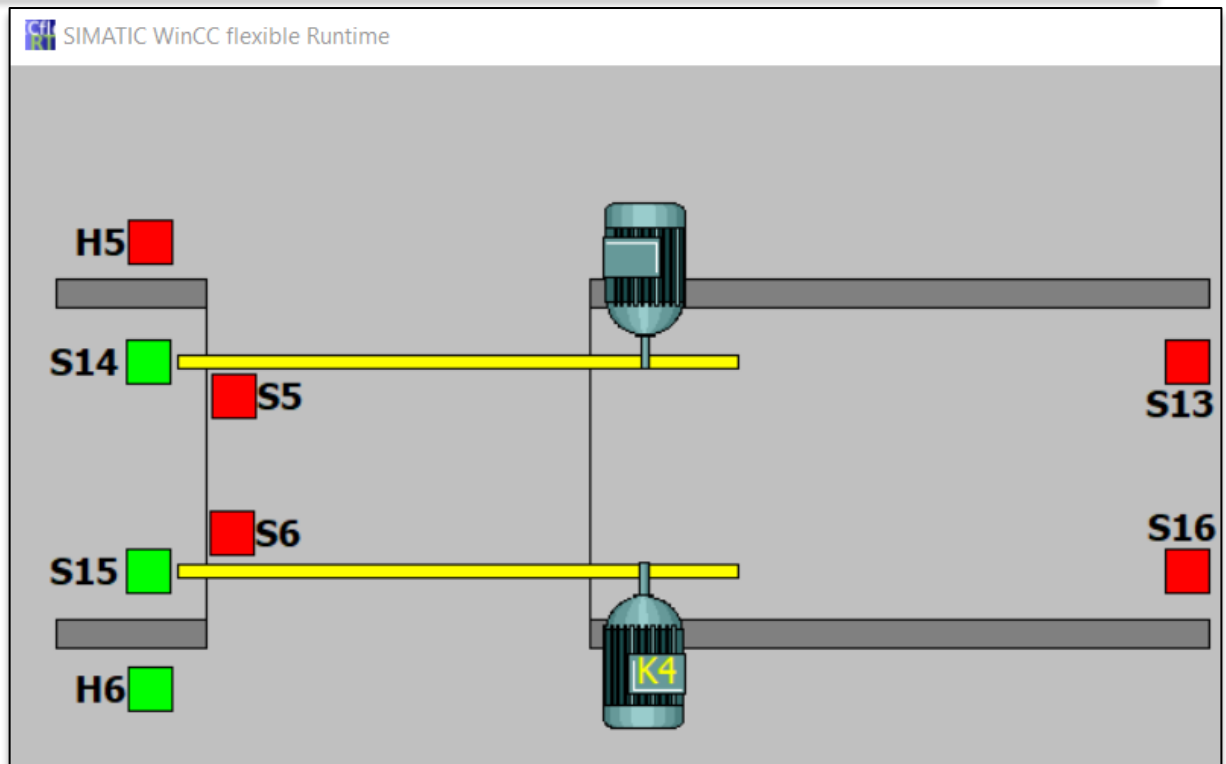


Figure 4-9 : Les deux portes serrures se franchissent initialement



Pour déverrouiller les portes, activez un bouton de demande, S1 ou S2, et assurez -vous que les deux portes ne restent jamais fermées en même temps.

Le système de verrouillage ne réagit pas si le seuil est franchi, et chaque entrée d'écluse est équipée d'une barrière lumineuse.

Les voyants lumineux situés à côté des boutons de demande signalent la pression maintenue sur le bouton.

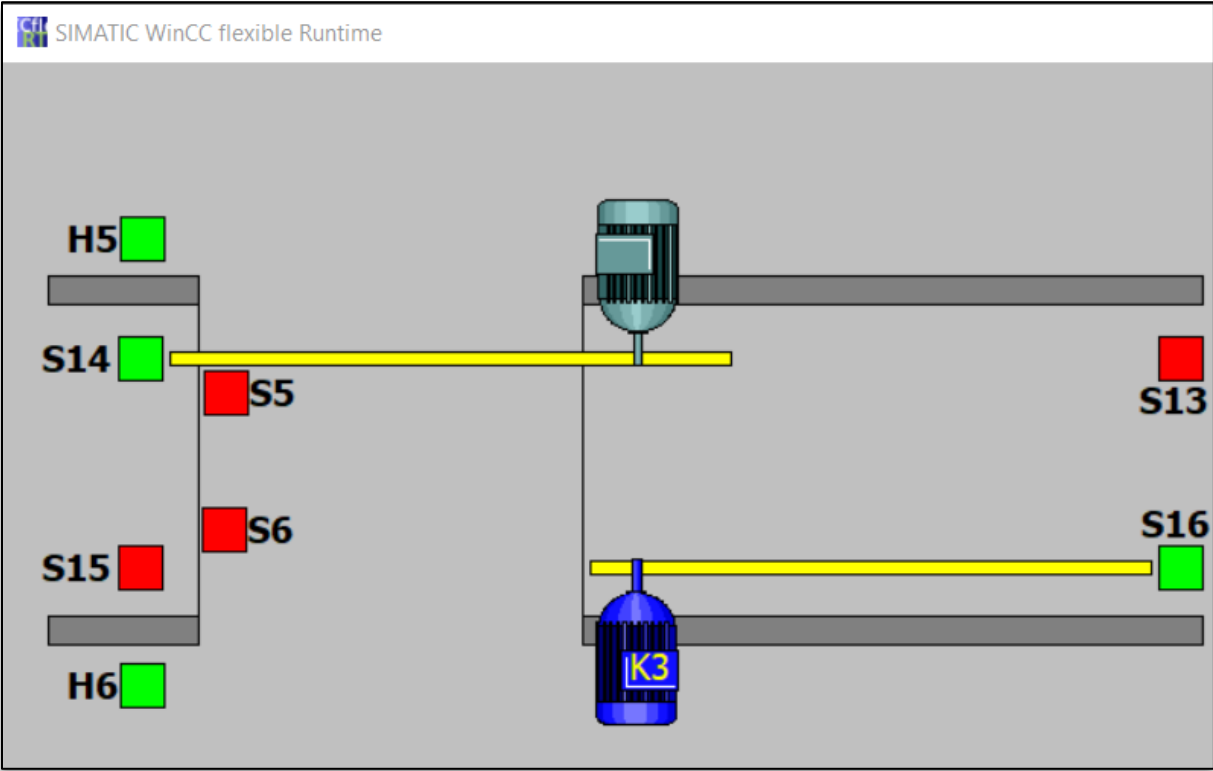


Figure 4-10, 11 : Fermeture du porte 1 et ouverture de porte 2

2.3. Système de réseau d'air comprimé :

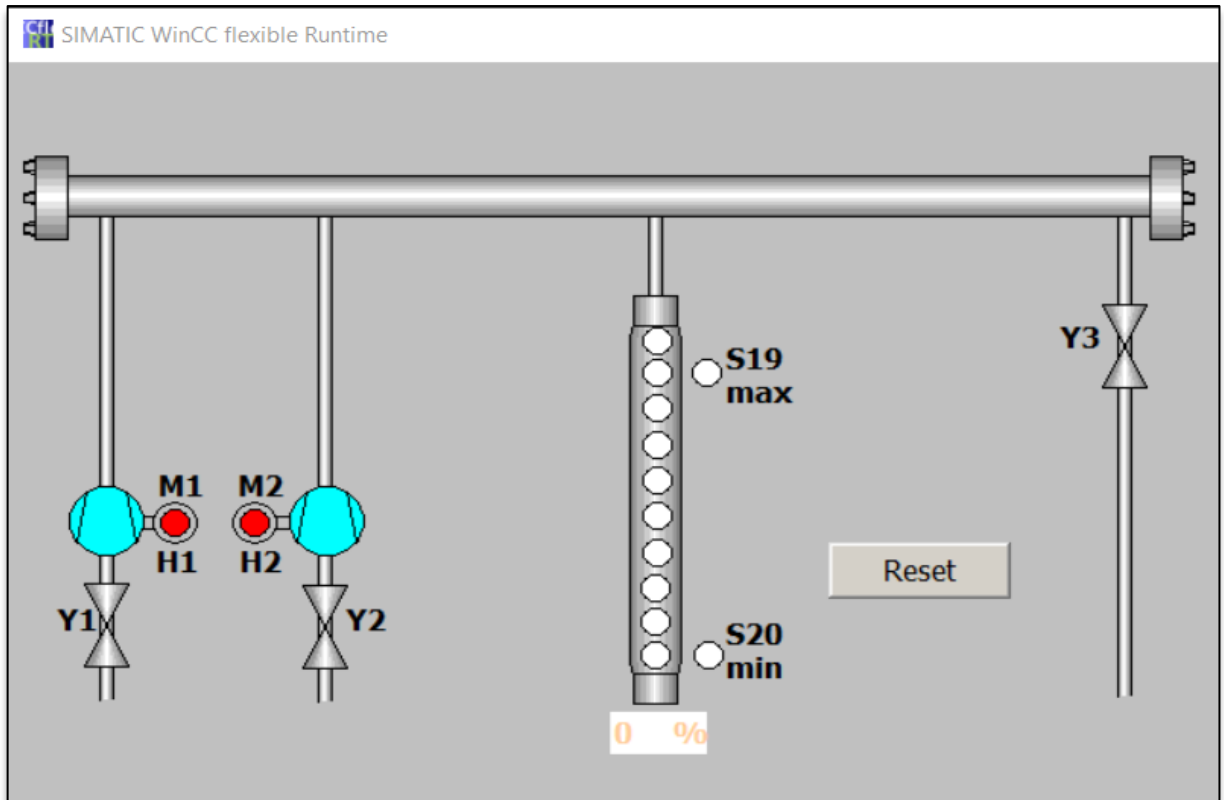


Figure 4-12 : Etat initiale de système

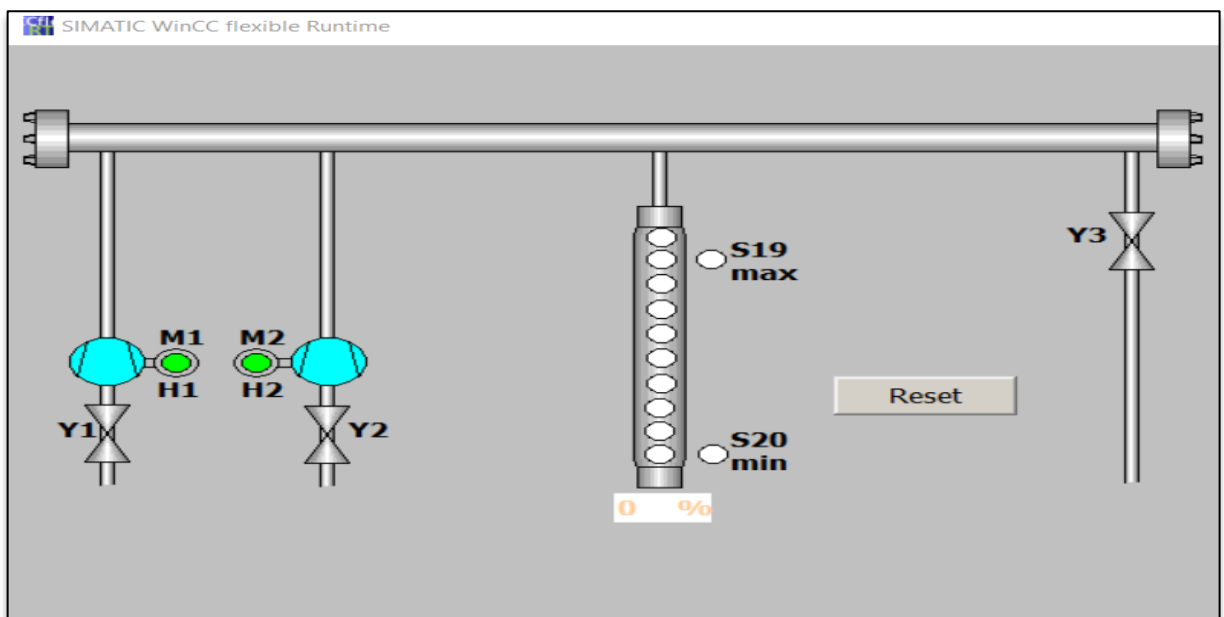


Figure 4-13 : Démarrage du système

L'activation du bouton de démarrage S1 ouvre les électrovannes Y1 et Y2.

En même temps, les moteurs de compression M1 et M2 démarrent et pompent de l'air comprimé dans le réservoir.

Les compresseurs fonctionnent à 70 % de la capacité de pompage

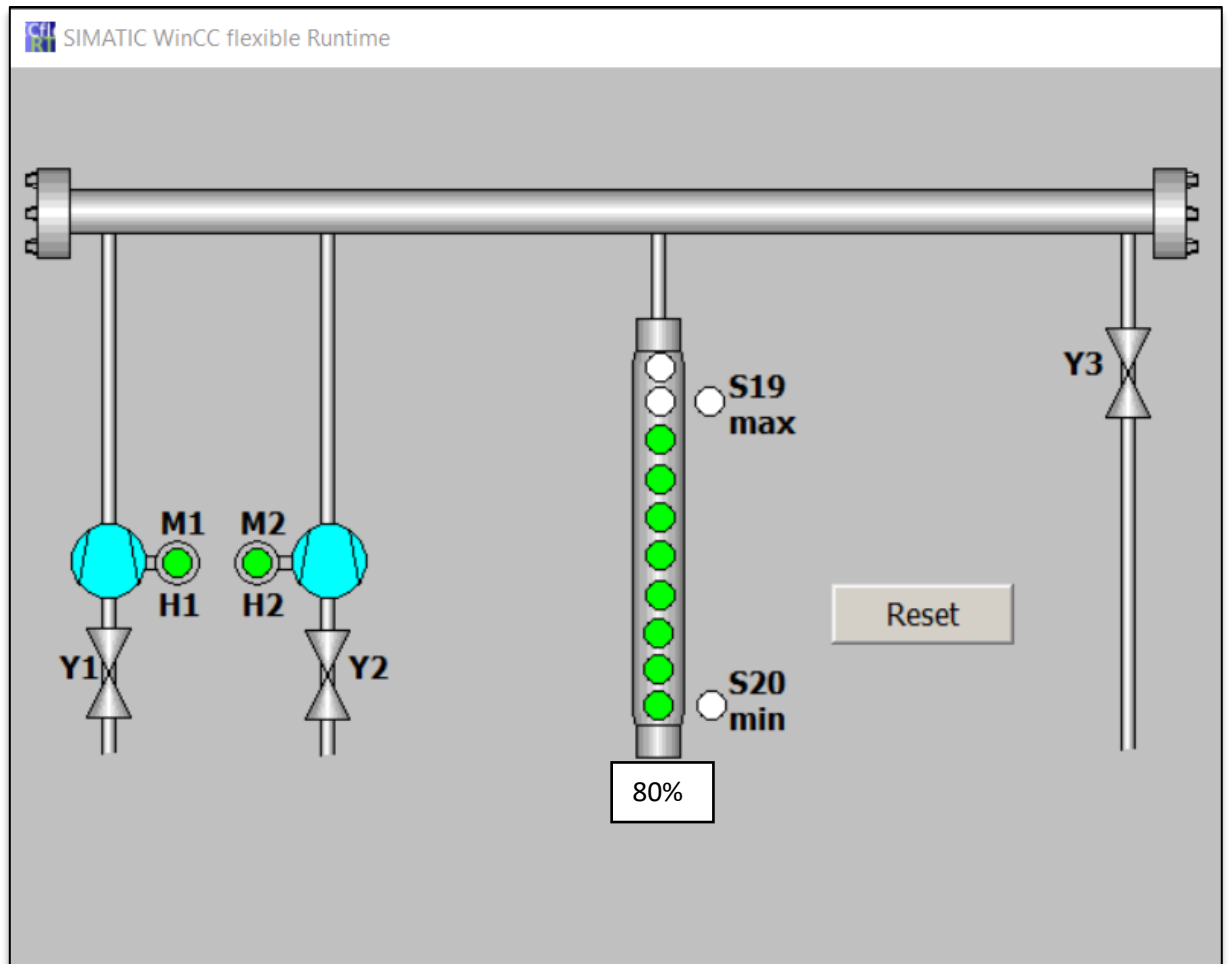


Figure 4-14 : Le pression arrive à 80%

Avec une pression de 80% de la pression maximale du conteneur, les compresseurs sont déconnectés et les vannes magnétiques sont fermées.

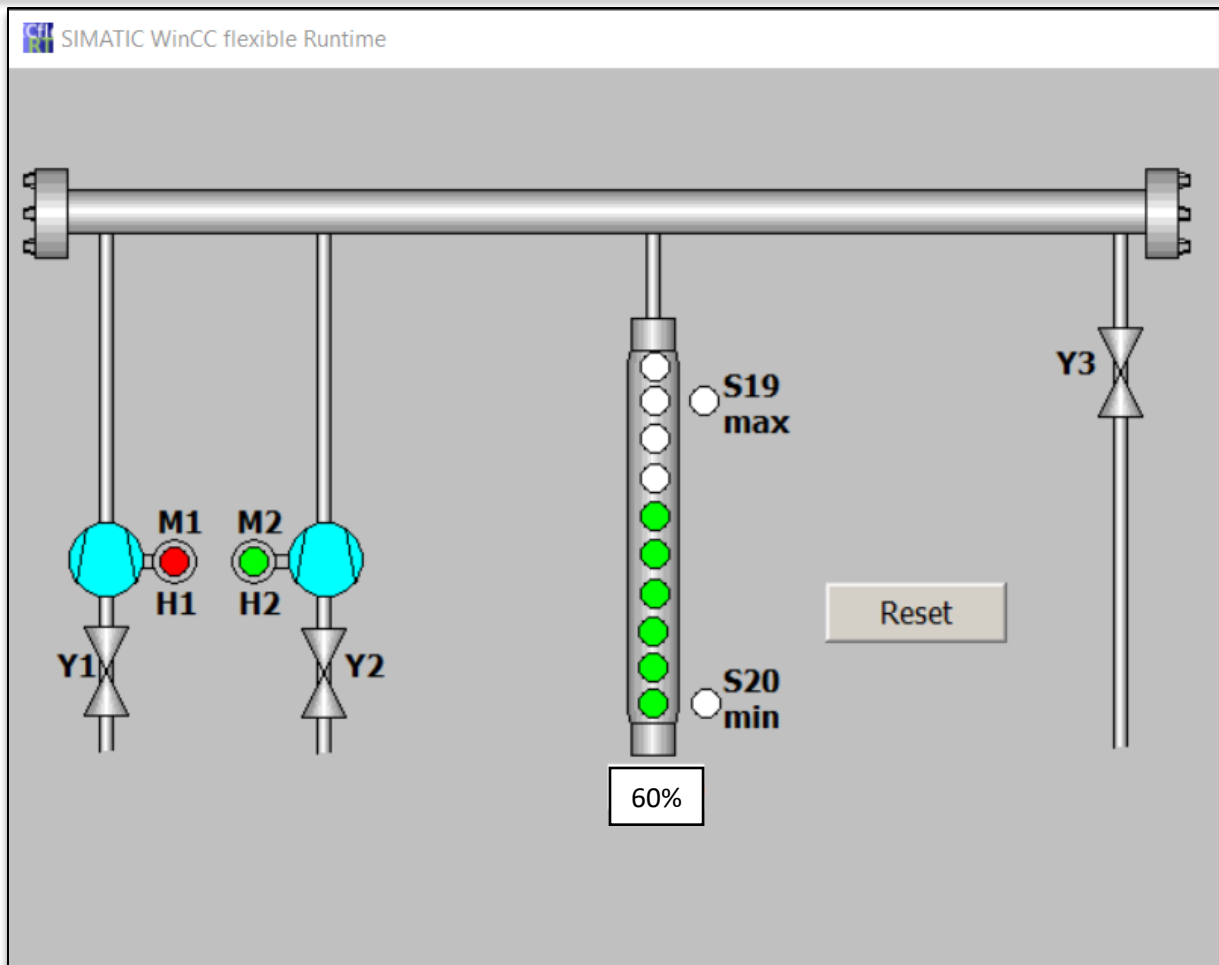


Figure 4-15 : Chute a 60% de la pression maximale

Si la pression du réservoir chute à 60 % de la pression maximale du réservoir, l'électrovanne Y1 s'ouvre et le compresseur M2 est activé avec une capacité de pompe de 70 %.

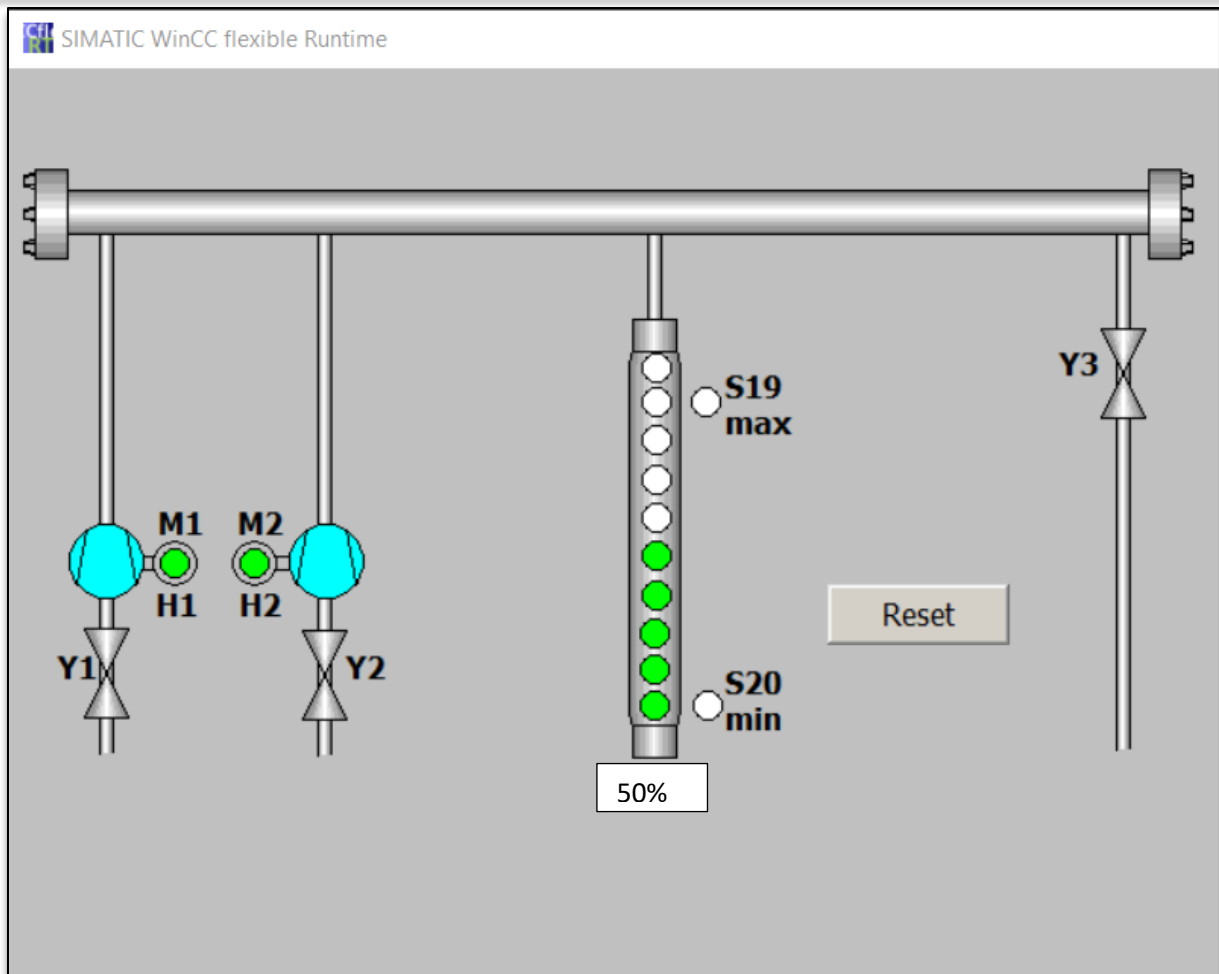


Figure 4-16 : Chute à 50% de la pression maximale

La pression continue à chuter jusqu'à une valeur de 50 %, l'électrovanne Y1 s'ouvre et le compresseur M1 avec une capacité de pompe de 70 % est ajouté.

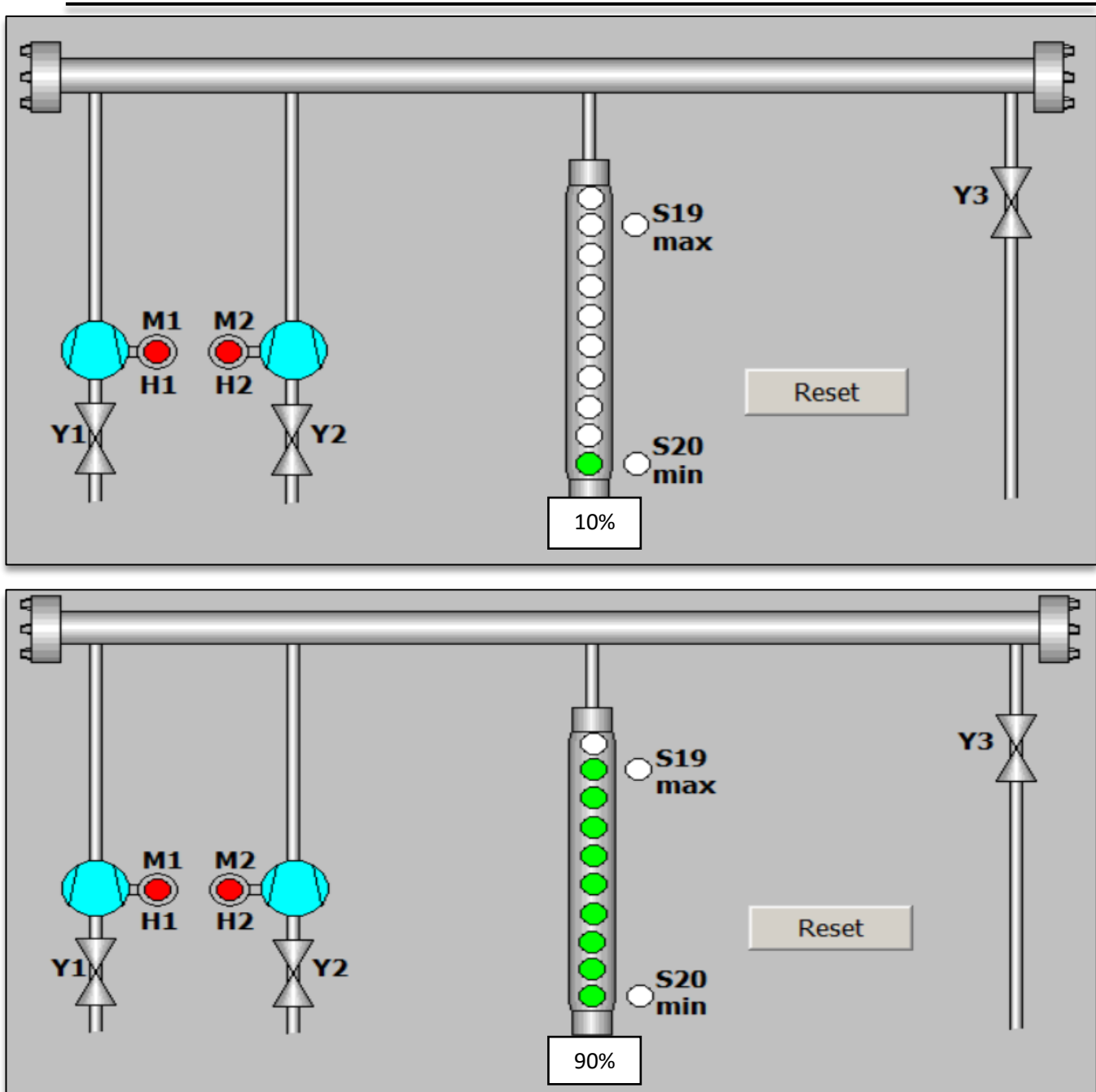
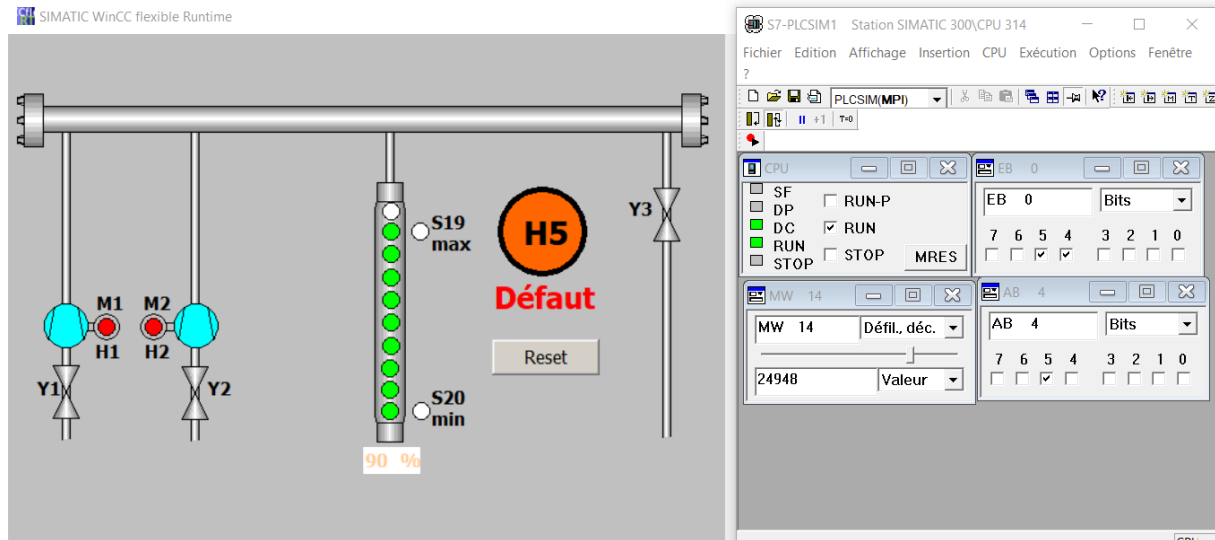


Figure 4-17, 18 : La pression dépasse les normes (S19 / S20)

La pression du réservoir dépasse la pression maximale (S19) ou tombe en dessous de la pression minimale (S20), tous les compresseurs doivent être arrêtés, toutes les électrovannes

fermées et un message de défaut (H5) doit être affiché.



Conclusion :

La surveillance des systèmes industriels à l'aide du WinCC Explorer présente une solution efficace pour assurer le bon fonctionnement et la sécurité des processus industriels. Dans ce chapitre, nous avons examiné trois systèmes spécifiques : le système de remplissage contrôlé, le système de commande de deux portes serrures et le système de réseau d'air comprimé.

Le système de remplissage contrôlé permet un remplissage précis des récipients en surveillant en permanence le niveau de liquide et en ajustant automatiquement le débit d'entrée. Cela garantit un niveau de remplissage optimal et évite les débordements ou les sous-remplissages.

Le système de commande de deux portes serrures assure la sécurité dans les environnements industriels en contrôlant l'accès aux zones restreintes. En utilisant des dispositifs de verrouillage électromécaniques et des capteurs, le système garantit que les portes ne peuvent être ouvertes que lorsque certaines conditions sont remplies, réduisant ainsi les risques d'accidents et de dommages matériels.

Le système de réseau d'air comprimé est crucial dans de nombreux processus industriels, fournissant de l'air comprimé pour alimenter des équipements et des outils. La surveillance de ce système permet de détecter les fuites, les baisses de pression ou d'autres problèmes, assurant ainsi un fonctionnement continu et efficace.

Grâce au WinCC Explorer, ces systèmes peuvent être surveillés en temps réel, avec la possibilité de visualiser les données, de recevoir des alertes en cas d'anomalies et de prendre des mesures correctives rapidement. Cela améliore la fiabilité, la sécurité et l'efficacité globale des processus industriels.

En conclusion, la surveillance des systèmes industriels à l'aide du WinCC Explorer offre une solution puissante pour optimiser les opérations industrielles. En intégrant ces systèmes de surveillance avancés, les entreprises peuvent améliorer la gestion des processus, minimiser les risques et assurer une productivité maximale.

Conclusion générale

En conclusion de ce mémoire de fin d'études intitulé "Automatisation de trois systèmes industriels avec API S7-300 et développement d'une interface graphique via Wincc: Système de remplissage, portes serrures et système d'un réseau d'air comprimé", il est clair que l'utilisation de l'API S7-300 et du logiciel Wincc Explorer a permis une automatisation efficace de ces systèmes industriels spécifiques.

Le chapitre 1 a fourni une compréhension approfondie de l'API S7-300, soulignant son importance dans le contrôle et la surveillance des processus industriels. Les connaissances acquises dans ce chapitre ont jeté les bases nécessaires pour la réalisation de ce projet.

Le chapitre 2 a introduit l'utilisation des logigrammes pour la modélisation et la programmation des processus. Les logigrammes ont permis une représentation visuelle claire et concise des différentes étapes impliquées dans chaque système, facilitant ainsi la conception et la compréhension du projet.

Le chapitre 3 s'est concentré sur le développement du langage Ladder, un langage de programmation couramment utilisé dans les automates programmables. Le développement du Ladder pour les trois systèmes a permis de mettre en œuvre les fonctionnalités souhaitées, garantissant un contrôle précis et sécurisé.

Enfin, le chapitre 4 a abordé la réalisation de l'interface graphique à l'aide du logiciel Wincc Explorer. Cette interface a fourni une interface utilisateur conviviale et interactive, permettant aux opérateurs de surveiller et de contrôler facilement les systèmes automatisés.

En conclusion, ce mémoire démontre l'efficacité de l'API S7-300 et du logiciel Wincc Explorer dans l'automatisation de systèmes industriels spécifiques tels que le système de remplissage, les portes serrures et le système d'un réseau d'air comprimé. L'utilisation de logigrammes et du langage Ladder a permis une programmation précise et cohérente, tandis que l'interface graphique développée a amélioré l'expérience utilisateur et la facilité de contrôle. Ce projet contribue ainsi à l'optimisation des processus industriels, à l'amélioration de la sécurité et à l'augmentation de l'efficacité des opérations industrielles.

Bibliographies

- [1] : A. Zinne Eddine et Yahia : " Commande par API d'un système industriel " PFE, Département de Pétrochimie, Option Automatisation, Univ de SKIKDA 2021/2022
- [2] : Programmable Controllers: Theory and Implementations 2nd edition – L.A. Bryan & E.A. Brayn (Georgia, USA)
- [3] : Dr. K. Labadi cour d'automatique 2008/2009
- [4] : Slim BEN SAOUD " Les Automates Programmables Industriels (API) " .
- [5] : Alain Gonzaga " Les Automates Programmables Industriels " .
- [6] : L. Bergougnoux " A.P.I. Automates Programmables Industriels". Polythèque Marseille, 2004/2005.
- [7] : A. Anissa et B. Salah-Eddine : " Réalisation et gestion d'un prototype de station de pompage à base d'automates programmables industriels SIEMENS " . PFE, Département du Génie Electrique, Option Automatique, Ecole Nationale Polytechnique, Alger 2006/2007.
- [8] : Jake Watkins – Introduction to PLCs - Realpars.com
- [9] : Programmable Controllers: Theory and Implementations 2nd edition – L.A. Bryan & E.A. Brayn (Georgia, USA)
- [10] : Les Automates Programmables Industriels – Alain Gonzaga
Automation Manufacturing Systems with PLCs – Hugh Jack v4.2, 03/04/2003
- [11] : Hans Berger " Automating with simatic s7-300 inside Tia portal "
- [12] : Réalisation technologique du GRAFCET par Daniel Dupont et David Dubois.
- [13] : Mémoire de Master "automatisation et réalisation à petite échelle (maquette) d'une chaine transporteuse de briques ", Université Hassiba Benbouali De Chlef, Juin 2016.
- [14] : Automatisme, Reproduction Interdite Association Ouvriere Des Compagnons Du Devoir Du Tour De France, Fevrier 2004.

- [15] : Cours Complet sur le Grafcet & Exercices Corrigés.
- [16] : Programmation des Automates : Utilisation du GRAFCET et du GEMMA – Philippe le Brun
- [17] : Les Automates Programmables Industriels – Alain Gonzaga GRAFCET : Notion de Base – D. Dubois
- [18] : Formation step7 Totally Integrated Automation (T.I.A.), 2001.
- [19] : Simulateur API universel d'installations : Ensemble de modelés de base – Lucas-Nülle
- [20] : Pierre Duysinx, Geoffray Hutsemekers, Henri Lecocq " Automatisation Et Robotisation De La Production " Université De Liège 2009-2010.
- [21] : Jean Caelen, Définition et caractérisation des IHM, Laboratoire CLIPS-IMAG, Grenoble .
- [22] : Formation WinCC Al Asri Industrial.
- [23] : Simatic HMI Wincc V7.2 getting started