

N° Ordre...../FHC/UMBB/2023

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE
UNIVERSITE M'HAMED BOUGARA-BOUMERDES



Faculté des Hydrocarbures et de la Chimie
Mémoire de Fin d'Etudes
En vue de l'obtention du diplôme :

MASTER

Présenté par

CHETEHOUNA OUSSAMA
MOUCI KOTEYBA

Filière : Hydrocarbures
Option : Commande automatique

Thème

**Commande et supervision d'une unité d'élevage de poulet
de chair sous « TIA PORTAL 15.1 »**

Devant le jury :

HABBI Hacene	Pr	UMBB	Président
KESRAOUI Mohamed	Pr	UMBB	Examineur
CHAIB Ahmed	MCA	UMBB	Encadreur

Année universitaire 2022/2023



Faculté des Hydrocarbures et de la Chimie
Département : Automatisation et Electrification des procédés
Filière : Hydrocarbures
Option : Commande automatique

Mémoire de Fin d'Etudes
En vue de l'obtention du diplôme :

MASTER

Thème

**Commande et supervision d'une unité d'élevage de poulet
de chair sous « TIA PORTAL 15.1 »**

Présenté par :
CHETEHOUNA OUSSAMA
MOUICI KOTEYBA

Avis favorable de l'encadreur :

Signature
A. CHAIB

Avis favorable du Président du jury
Nom Prénom **Signature**

HABBI Hacene

Cachet et signature

Remerciement

Nous tenons à remercier Allah, le tout puissant de nous avoir donnés le courage, la volonté et la patience pour réaliser ce travail.

Nos sincères gratitudes à notre promoteur **Mr. CHAIB AHMED** pour la qualité de son enseignement, son orientation et ses conseils.

Nous adressons aussi nos remerciements à tous les enseignants et le personnel du département de « Automatisation des Procédés et Electrification » pour avoir enrichi nos connaissances scientifiques et techniques et en particulier notre spécialité commande automatique.

Nous tenons aussi à remercier chaleureusement les membres de jury pour l'honneur qu'ils nous ont fait en acceptent d'évaluer notre projet.

Enfin, nos remerciements vont à nos familles, amis, toutes et à tous nos camarades de promotion 2022-2023, et à tous ceux qui ont contribué de près ou de loin au bon déroulement de ce mémoire et notre parcours académique.

OUSSAMA & KOTEYBA

Merci!

Résumé

Les aviculteurs ont recours à l'élevage de poulets de chair afin de fournir des protéines à la consommation humaine de deux manières différentes, traditionnels et moderne. En Algérie, un grand nombre d'aviculteurs utilisent des technologies classiques et des techniques traditionnelles.

Un mauvais contrôle des conditions environnementales de l'aviculture affecte négativement la santé et la croissance de la volaille. Pour cela, nous avons conçu un système intelligent à base d'un automate programmable qui assure le contrôle de température et d'autres choses (ventilation, éclairage...) facilement, rapidement et automatiquement, afin d'améliorer le produit et accroître l'efficacité de ce secteur.

ملخص

يلجأ مزارعو الدواجن إلى تربية الفروج لتوفير البروتين للسنة الك البشرية بطريقتين مختلفتين، تقليدية وحديثة. في الجزائر، يستخدم عدد كبير من مزارعي الدواجن التكنولوجيات التقليدية والطرق التقليدية. يؤثر ضعف السيطرة على الظروف البيئية في تربية الدواجن بشكل سلبي على صحة وزمو الدواجن. لهذا الغرض، قمنا بتصميم نظام ذكي يعتمد على نظام آلي قابل للبرمجة يضمن التحكم في درجة الحرارة وأشياء أخرى (التهوئة والضاءة وما إلى ذلك) بسهولة وسرعة وثقلية من أجل تحسين المنتج وزيادة كفاءة هذا القطاع.

Abstract

Poultry farmers resort to raising broilers to provide protein for human consumption in two different ways, traditional and modern. In Algeria, a large number of poultry farmers use conventional technologies and traditional techniques.

Poor control of environmental conditions in poultry farming negatively affects the health and growth of poultry. For this, we have designed an intelligent system based on a programmable automaton which controls temperature and other things (ventilation, lighting, etc.) easily, quickly and automatically, in order to improve the product and increase efficiency of this sector.

TABLE DES MATIERES

Remerciement

Résumé

Table des matières

Liste des figures

Liste des tableaux

Liste des abréviations

Introduction générale2

CHAPITRE I

« Bâtiments d'élevage de poulet de chair »

I.1. Introduction5

I.2. Mode d'élevage du poulet en Algérie5

I.2.1. Elevage au sol5

I.2.1.1. Elevage intensif5

I.2.1.2. Elevage extensif5

I.2.2. Elevage en batterie6

I.2.3. L'élevage mixte : sol-batterie6

I.3. L'intérêt de Bâtiment d'élevage avicole6

I.4. Implantation de bâtiment6

I.5. La situation géographique de l'élevage7

I.6. Conception de l'exploitation et du bâtiment8

I.7. Dimension9

I.7.1. Surface du bâtiment9

I.7.2. La largeur du bâtiment9

I.7.3. Longueur9

I.8. Isolation9

I.9. Contrôle des conditions d’ambiances de bâtiment d’élevage.....	10
I.9.1. Qualité d’air.....	10
I.9.2. Eau	11
I.9.3. Température.....	11
I.9.4. Chauffage	11
I.9.5. Systèmes de ventilation.....	13
I.9.5.1. La ventilation naturelle : Bâtiment ouvert	13
I.9.5.2. La ventilation dynamique : bâtiments à environnement contrôlé	14
I.9.6. Système de rafraîchissement par évaporation.....	18
I.9.6.1. Rafraîchissement par panneau refroidissant	18
I.9.6.2. Brumisation/Pulvérisation	19
I.9.7. Litière et humidité.....	20
I.9.8. Eclairage.....	21
I.10. Conclusion.....	22

CHAPITRE II

« Généralité sur les automates programmables industriels »

II.1. Introduction.....	24
II.2. Définition et Historique des APIs	24
II.3. Le rôle de l’automate.....	26
II.4. Architecture d’un API.....	27
II.4.1. Aspect extérieur.....	27
II.4.2. Structure interne	30
II.4.2.1. Module d’alimentation.....	31
II.4.2.2. Unité centrale.....	31
II.4.2.3. Modules d’entrées/sorties (E/S)	34
II.4.2.4. Les liaisons de communication	35
II.4.2.5. Eléments auxiliaires	36
II.4.3. Structure d’un système automatisé.....	36
II.4.4. Les caractéristiques principales d’un API	38
II.5. Son principe de fonctionnement	38

II.6. Critères du choix d'un API	39
II.7. Avantages	40
II.8. Les domaines d'application des automates programmables	40
II.9. Comment programmons-nous un API ?	41
II.9.1. Langages d'API	41
II.9.1.1. Objets communs à tous les langages	44
II.9.1.2. Comment choisit-on la langue ?	45
II.10. Sécurité	45

CHAPITRE III

« Programmation et supervision avec TIA PORTAL et IHM »

III.1. Introduction	48
III.2. Matériels utilisés d'exemple d'un poulailler réel	48
III.3. TIA PORTAL	48
III.3.1. Description du logiciel TIA PORTAL	49
III.3.1.1. SIMATIC STEP7 (S7)	50
III.3.1.2. SIMATIC WinCC	50
III.3.2. Vues du TIA PORTAL	51
III.3.2.1. Vue du portail	51
III.3.2.2. Vue du projet	52
III.3.3. Les avantages du TIA PORTAL	53
III.4. Le programme d'automatisation de la température du poulailler	54
III.4.1. Etapes d'automatisation logicielle	54
III.4.1.1. Configuration matérielle	54
III.4.1.2. Programmation	58
III.4.1.3. Configuration IHM	67
III.5. Conclusion	71

CHAPITRE IV

« Compilation et simulation du programme »

IV.1. Introduction	73
IV.2. Simulation de l'API et l'IHM	73

IV3. Teste et résultats	76
IV3.1. En mode manuel	76
IV3.2. En mode automatique	79
IV4. Conclusion	80
Conclusion générale	82

Références bibliographiques

Annexe A

Annexe B

Annexe C

Annexe D

Liste des figures

Numéro	Titre de la figure	Page
Figure I.1	Exemple d'un bon agencement d'élevage.	8
Figure I.2	Chauffage à air pulsé.	12
Figure I.3	Exemple de ventilation naturelle.	14
Figure I.4	Flux d'air correct en ventilation minimale.	15
Figure I.5	Vue intérieure du bâtiment en mode ventilation de transition. La répartition des oiseaux indique que l'environnement est confortable.	16
Figure I.6	Exemple type d'un bâtiment équipé d'une ventilation en tunnel.	17
Figure I.7	Rafrâichissement à panneaux refroidissants avec ventilation en tunnel.	19
Figure I.8	Exemple de système de brumisation dans une poulailler.	20
Figure I.9	Exemples d'intensité lumineuse à 10 lux et 30 lux.	21
Figure II.1	Schéma généralisé d'un API.	25
Figure II.2	Exemple d'un API S7-300 (marque de SIEMENS AG).	25
Figure II.3	Automate compact (LOGO ; SIEMENS).	27
Figure II.4	Automate modulaire (Mitsubishi).	28
Figure II.5	Automate modulaire (SIEMENS).	29
Figure II.6	Structure interne d'un API.	31
Figure II.7	Schéma de principe d'un système automatisé.	36
Figure II.8	Principe de fonctionnement des APIs.	39
Figure II.9	Exemple d'un programme avec FBD.	41
Figure II.10	Exemple d'un programme avec IL.	42
Figure II.11	Exemple d'un programme avec LD.	43
Figure II.12	Exemple d'un programme avec SFC.	43
Figure II.13	Exemple d'un programme avec ST.	44
Figure III.1	Logo de TIA PORTAL (V15.1).	49
Figure III.2	Illustration de la construction de TIA PORTAL.	49
Figure III.3	Vue du TIA PORTAL V15.1.	51
Figure III.4	Vue du projet.	52
Figure III.5	Planification d'une solution d'automatisation.	54
Figure III.6	Planification de la configuration.	54
Figure III.7	Ajout d'API.	55

Figure III.8	Description de l'appareil ajouté.	55
Figure III.9	Ajout d'IHM.	56
Figure III.10	Description d'IHM ajouté.	56
Figure III.11	Liaison IHM/API.	57
Figure III.12	Partie de la table de variables.	59
Figure III.13	Les blocs de programmation dans notre système.	60
Figure III.14	Programme principal de système (MAIN OB1).	61
Figure III.15	Programme d'alarme d'incendie (FC1).	62
Figure III.16	Programme de commutation (manuel ou automatique) (FB).	63
Figure III.17	Programme en mode manuel (FB2).	64
Figure III.18	Programme en mode automatique (FB3) -1-.	65
Figure III.19	Programme en mode automatique (FB3) -2-.	66
Figure III.20	Configuration IHM.	67
Figure III.21	Hiérarchie des vues.	67
Figure III.22	Vue initiale (Poulailler).	68
Figure III.23	Partie de la table de variables IHM.	70
Figure III.24	Exemple d'animation.	70
Figure III.25	Exemple d'évènement d'un bouton.	71
Figure IV.1	Simulation de l'API.	73
Figure IV.2	Configuration de l'API et simulation de l'IHM.	74
Figure IV.3	Compilation du programme.	74
Figure IV.4	Interface de simulation PLCSIM.	75
Figure IV.5	Indication de la mise en ligne du programme et de l'IHM.	76
Figure IV.6	Démarrage en mode manuel.	76
Figure IV.7	Écran « écran de commande » à l'arrêt.	77
Figure IV.8	Écran de « Poulailler » à l'arrêt.	77
Figure IV.9	Écran « écran de commande » en fonctionnement.	78
Figure IV.10	Écran de « Poulailler » en fonctionnement.	78
Figure IV.11	Commutation en mode automatique.	79
Figure IV.12	Indication d'alarme.	79
Figure IV.13	Graphe de température.	80

Liste des tableaux

Tableau III.1	Représentation des éléments des vues.	69
----------------------	---------------------------------------	----

Liste des abréviations

O.N.A.B	Office national des aliments du bétail.
O.R.AVI	Office régional aviculture.
HR	Humidité relative.
ROM	Read Only Memory. (Anglais)
PROM	Programmable ROM. (Anglais)
EPROM	Erasable PROM. (Anglais)
EEPROM	Electrically EPROM. (Anglais)
IHM	Interface homme-machine.
TIA PORTAL	Totally Integrated Automation. (Anglais)
SCADA	Supervisory Control And Data Acquisition. (anglais)

Introduction générale

Introduction générale

Les volailles constituent une source de protéine animale appréciable et économique, notamment pour les pays en voie de développement, ce qui a justifié son développement très rapide sur l'ensemble du globe depuis une trentaine d'années. L'aviculture est généralement pratiquée en deux types de modalités : « traditionnelle » (ou familiale), et « moderne » (ou semi industrielle) [1].

Actuellement l'Algérie compte près de 140 millions de poules et une production de 350.000 tonnes à 400.000 tonnes de viandes blanches et de 6 à 7 milliards d'œufs par an, selon les déclarations de l'Association nationale des commerçants et artisans (ANCA, 2020) [1].

L'élevage de poulet de chair a connu un développement important ces dernières années au niveau international sur le plan technologique, en termes des équipements utilisés et les techniques d'élevage en général. Avec la modernisation des infrastructures avicoles, beaucoup de sociétés ont investi dans des recherches scientifiques pour améliorer l'aviculture à l'échelle mondiale. En Algérie la filière avicole connaît une perturbation malgré le nombre considérable des poulaillers, cela vient des carences enregistrées aux technologies utilisées dans ce domaine.

Aujourd'hui pour améliorer la filière de l'aviculture en Algérie, des efforts considérables devraient être déployés en vue de : encourager les investisseurs dans ce domaine à moderniser les infrastructures industrielles, éliminer les obstacles, organiser et développer cette activité en faveur de l'économie nationale.

L'objectif de cette étude est de réaliser un système automatique (en utilisant l'Automate Programmable Industriel et l'Interface Homme Machine) qui permettrait le contrôle en temps réel la température et quelque autre choses (ex : moteur de nettoyage, l'éclairage...) d'un poulailler. Car le contrôle efficace de l'environnement d'élevage de volaille assure d'une part une excellente production quantitative et qualitative et d'autre part réduire les coûts d'exploitation via l'automatisation et le contrôle automatique des activités quotidiennes dans les bâtiments d'élevage.

Ce mémoire se compose de quatre chapitres qui sont exposés de manière :

Partie théorique ;

- **Chapitre I :** Est consacré à la description de Bâtiments d'élevage de poulet de chair.
- **Chapitre II :** Sera consacré une présentation générale des automates programmables industriels.

Partie pratique ;

- **Chapitre III :** Il présentera toute la configuration matériel et logiciel de notre système (Programmation avec TIA PORTAL V15.1 et l'Interface Homme Machine...)
- **Chapitre IV :** Sera réservé aux simulations du programme crée (SIMATIC WINCC).

En fin, notre travail est terminé par une conclusion générale.

CHAPITRE I

« Bâtiments d'élevage de poulet de chair »

I.1. Introduction

Dans ce chapitre on va donner quelques notions de base sur les bâtiments d'élevage de poulet de chair et le choix technique de sites adaptés pour la pratique de cette activité, de telle sorte à être isolé et protégée des influences extérieures. Ensuite, on va parler sur les conditions de conception des élevages et aussi les normes qui on doit respecter (Surface du bâtiment, largeur, longueur, isolation). En dernier, nous citerons les technologies et les techniques utilisées dans les poulaillers pour avoir un bon contrôle des conditions d'ambiance et des paramètres zootechniques (température, humidité, qualité d'aire, l'éclairage...).

I.2. Mode d'élevage du poulet en Algérie

Il existe deux types :

I.2.1. Elevage au sol

Il peut être intensif ou extensif ;

I.2.1.1. Elevage intensif

Il se pratique pour le poulet de chair pour les grands effectifs. Il a pris sa naissance en Algérie avec l'apparition des couvoirs au sein des structures du Ministère de l'Agriculture et de la Révolution Agraire (M.A.R.A.) qui a créé l'O.N.A.B et l'O.R.AVI.

I.2.1.2. Elevage extensif

Cet élevage se pratique pour les poules pondeuses, il s'agit surtout des élevages familiaux de faibles effectifs et il s'opère en zone rurale. La production est basée sur l'exploitation de la poule locale, et les volailles issues sont la somme de rendement de chaque éleveur isolé. C'est un élevage qui est livré à lui-même, généralement géré par des femmes, l'effectif moyen de chaque élevage fermier est compris entre 15 et 20 sujets, les poules sont alimentées par du seigle, de la criblure, de l'avoine et des restes de cuisines. Elles sont élevées en liberté et complètent leur alimentation autour de la ferme. Les poules sont destinées à la consommation familiale ou élevées pour la production des œufs [2].

I.2.2. Elevage en batterie

Cet élevage qui a été introduit nouvellement en Algérie se fait pour les poules pondeuses. Il est beaucoup plus coûteux par rapport au premier. L'élevage du poulet convient très bien au climat Algérien. L'Etat dans le cadre de sa politique de relance économique, encourage au maximum les éleveurs et les coopératives à pratiquer cet élevage, pour diminuer l'importation des œufs de consommation et des protéines animales [3].

I.2.3. L'élevage mixte : sol-batterie

Il utilise les avantages des deux modes d'élevage cités précédemment. Le démarrage de 0 à 6 semaines se fait au sol. Les poussins ont une grande rusticité qui sera ressentie en deuxième phase. Finition en batterie : dans cette phase, l'éleveuse n'est plus indispensable.

Cette méthode d'élevage se justifie par l'insuffisance de locaux pour l'élevage au sol pendant 03 mois surtout pour les grands effectifs, et par l'impossibilité d'une installation complète en batteries [4].

I.3. L'intérêt de Bâtiment d'élevage avicole

Le Bâtiment est le local où les animaux s'abritent contre toute source de dérangement, c'est le local où l'animal trouve toutes les conditions de confort. Pour cette raison, il doit prendre en considération tous les facteurs internes et externes du bâtiment. La conception et la réalisation d'un élevage de poulets de chair doivent être réfléchies, car sa réussite est subordonnée à un bon habitat, une bonne alimentation, un abreuvement correct et une bonne protection sanitaire avec l'approche bio-ingénierie [2].

I.4. Implantation de bâtiment

Le choix technique d'un site adapté considère notamment les mouvements d'air et l'humidité. Ainsi, l'implantation dans une vallée peut correspondre à de l'humidité et/ou à une insuffisance de renouvellement d'air en ventilation naturelle, surtout en période chaude.

L'insuffisance de renouvellement peut aussi être la résultante de tout autre obstacle au mouvement de l'air (une autre construction par exemple). A l'opposé, l'implantation sur une colline peut causer un excès d'entrée d'air du côté des vents dominants.

Pour les bâtiments à ventilation naturelle (non forcée), dans nos régions, il est souhaitable d'éviter le balayage transversal à cause des mouvements d'air excessifs. Avec un bâtiment à lanterneau, il faut écarter l'implantation pignon plein vent pour éviter le refoulement de l'air par le lanterneau. Un compromis consiste à orienter l'axe longitudinal du bâtiment dans une limite de 30 à 45 degrés de part et d'autre de la perpendiculaire aux vents dominants.

L'implantation considère également les aspects paysagers. **La Région wallonne**¹ a édité à ce sujet, un guide de conseils pour assurer la meilleure intégration possible des bâtiments agricoles dans le paysage [5].

L'orientation des bâtiments doit être choisie en fonction de deux critères :

- ❖ Le mouvement du soleil. On a intérêt à orienter les bâtiments selon un axe Est-Ouest de façon à ce que les rayons du soleil ne pénètrent pas à l'intérieur du bâtiment.
- ❖ La direction des vents dominants. L'axe du bâtiment doit être perpendiculaire à celle-ci pour permettre une meilleure ventilation.

En Algérie l'orientation doit être Nord-Sud pour éviter l'exposition aux vents :

- du Nord froids en hiver ;
- du Sud chauds en été [5].

I.5. La situation géographique de l'élevage

Les bâtiments doivent être situés de façon à être isolés des autres élevages, de volailles et d'autres animaux ; une distance minimale d'au moins 1km des autres installations de volailles ou d'autres animaux doit être respectée, ainsi que des routes utilisées pour le transport de volaille. Il est préférable d'avoir des oiseaux de même âge sur le site afin de limiter le recyclage des agents pathogènes et des souches de vaccins vivants [6].

¹ **La Région wallonne**, communément appelée **Wallonie**, est l'une des trois régions belges ; elle est située dans le sud de la Belgique

I.6. Conception de l'exploitation et du bâtiment

Le bâtiment doit être conçu de façon à minimiser les flux de circulation, faciliter le nettoyage et la désinfection, et empêcher les oiseaux et rongeurs d'entrer. Une barrière (clôture) est nécessaire pour empêcher les accès non autorisés. Le poulailler doit disposer d'un sol en béton ; les murs et le plafond doivent pouvoir être lavés (étanches) ; les conduits de ventilation doivent être accessibles et le bâtiment doit être dépourvu de piliers internes ou de rebords.

Il est impossible de nettoyer et de désinfecter correctement les sols en terre battue. L'espace autour du bâtiment doit être dégagé et plat jusqu'à 15 m pour pouvoir tendre la pelouse rapidement et facilement. Un espace de 1 à 3 mètres de largeur en béton ou graviers, installé directement autour du bâtiment, peut décourager l'entrée des rongeurs et offrir une zone de nettoyage et de stockage des pièces amovibles des équipements. La **figure ci-dessous** illustre un bon exemple d'organisation et d'agencement de l'élevage.



Figure I.1 : Exemple d'un bon agencement d'élevage.

I.7. Dimension

I.7.1. Surface du bâtiment [2]

La surface du bâtiment est directement fonction de l'effectif de la bande à y installer. On se base sur une densité de 13 à 15 poulets au mètre carré. Le surpeuplement entraîne des conséquences graves : croissance irrégulière ; poulets griffés, litières croûteuses, coccidioses.

I.7.2. La largeur du bâtiment

La largeur du bâtiment est liée aux possibilités de ventilation. On construit couramment des poulaillers de 8 m, 12 m ou 15 m de largeur.

I.7.3. Longueur

La longueur dépend de l'effectif des bandes à y loger.

Exemple de dimensions de poulaillers : 12m de large x 100m de long pour 10.000 poulets.

I.8. Isolation

L'isolation du bâtiment doit rendre l'ambiance à l'intérieur de celui-ci, la plus indépendante possible des conditions climatiques extérieures :

- * Limiter le refroidissement en hiver ;
- * Limiter les entrées de chaleur au travers des parois en été ;
- * Limiter les écarts de température entre l'ambiance et le matériau, pour éviter la condensation. Pour cela, toutes les parois du bâtiment seront isolées. Le bon isolant présente, outre une bonne résistance aux transferts calorifiques, une résistance au feu, aux insectes, aux rongeurs et aux pressions utilisées pour le nettoyage, ainsi qu'un bon rapport qualité/prix.

L'isolation de la toiture influence largement les pertes de chaleur en hiver et l'impact du rayonnement en été. Les murs sont généralement constitués de panneaux sandwich : une couche isolante entre 2 surfaces (fibrociment ou autre). On veillera aussi à assurer l'étanchéité du bâtiment de manière à limiter les entrées d'air parasites dans l'aire de vie des volailles.

Une étanchéité correcte est nécessaire pour une bonne ventilation dynamique. En outre, toute fuite d'air en dessous de 1,80 m de hauteur est particulièrement dangereuse pour les poussins. Il existe une dépression à l'intérieur du bâtiment, particulièrement en ventilation dynamique, avec ventilateurs extracteurs. Cette dépression assure une bonne ventilation. Si des entrées d'air anarchiques existent, elles diminuent la dépression et la ventilation ne s'effectue pas de manière optimale. L'air froid entrant tombe directement sur les animaux avant d'être réchauffé [5].

I.9. Contrôle des conditions d'ambiances de bâtiment d'élevage

La ventilation constitue le premier moyen de contrôle de l'environnement des oiseaux. La ventilation assure une bonne qualité d'air dans le bâtiment tout en maintenant les oiseaux dans une température confortable. Elle apporte de l'air frais, évacue l'excès d'humidité et limite l'accumulation de gaz et l'émission de sous-produits potentiellement dangereux [6].

Pendant les premières phases de la vie des oiseaux, la ventilation répartit la chaleur et fournit suffisamment d'air frais pour le maintien d'un air de qualité correcte dans le bâtiment.

À mesure que les oiseaux grandissent et qu'ils produisent plus de chaleur, il est nécessaire d'augmenter les taux de ventilation pour évacuer la chaleur, ainsi que les produits de la respiration (humidité) du bâtiment.

La surveillance du comportement des oiseaux et l'ajustement de la ventilation en conséquence sont essentiels pour assurer leur confort et les rendre actifs [6].

I.9.1. Qualité d'air

Les principaux contaminants de l'air dans l'environnement intérieur d'un bâtiment sont la poussière, l'ammoniac, le dioxyde de carbone, le monoxyde de carbone et l'excès de vapeur d'eau. La concentration de ces contaminants doit rester dans les limites réglementaires en permanence. Une exposition excessive et continue à ces contaminants peut :

- * Entraîner des lésions des voies respiratoires ;
- * Réduire l'efficacité de la respiration ;
- * Provoquer des maladies (p. ex. ascite ou maladie respiratoire chronique) ;
- * Influencer la régulation de la température ;
- * Dégrader la litière ;
- * Réduire les performances des oiseaux [6].

I.9.2. Eau

Les oiseaux produisent une quantité d'eau substantielle qui est transférée vers l'environnement. L'évacuation de cette eau présente dans le bâtiment est une fonction primordiale du système de ventilation.

Un oiseau de 2,3 kg consomme en moyenne 6,3 litres d'eau au cours de sa vie et en restitue 4,9 litres dans l'atmosphère.

Pour un lot de 10 000 poulets de chair, cela représente environ 49 000 litres d'eau qui seront réémis dans l'environnement sous forme d'humidité issue de la respiration ou d'excréments. La ventilation doit éliminer ce volume d'eau du bâtiment [6].

I.9.3. Température

L'une des fonctions premières de la ventilation, notamment dans la phase initiale de production, est de maintenir le bâtiment à une température appropriée afin de permettre aux oiseaux de rester dans un bon confort thermique.

Il n'est présenté qu'à titre indicatif. La température réellement programmée dépendra de l'HR et doit toujours se fonder sur le confort visible des oiseaux. Les variations de température affectent l'efficacité de la conversion alimentaire. Ceci est particulièrement vrai lorsque la température ambiante est trop froide : l'aliment sera plus utilisé pour la production de chaleur que pour la croissance [6].

I.9.4. Chauffage

Chaque bâtiment d'élevage de poulets de chair doit disposer d'une capacité de chauffage suffisante pour fournir la ventilation nécessaire au maintien d'une qualité d'air acceptable et pour maintenir la température du bâtiment tout au long de l'année.

La chaleur doit être uniformément répartie dans tout le bâtiment. Une distribution de chaleur inégale peut avoir un effet négatif sur l'uniformité des oiseaux. Lorsque les ventilateurs de circulation sont utilisés pour véhiculer et diffuser la chaleur dans le bâtiment, il faut veiller à ne pas créer de courants d'air à hauteur des oiseaux.

Lors des premières phases du cycle de production, le chauffage doit se déclencher au plus près de la température programmée pour le bâtiment. A mesure que les oiseaux grandissent et commencent à générer plus de chaleur corporelle, l'écart entre la température de référence du bâtiment et celle à laquelle le chauffage se déclenche pourrait être augmenté.

A titre d'exemple, le chauffage peut être réglé pour démarrer seulement si la température du bâtiment est inférieure de 1 à 2 °C à la température fixée pour le bâtiment. Ces choix et ces paramètres sont dictés en fonction de la réaction et du confort observés chez les oiseaux [6].

Lors du préchauffage du bâtiment, avant la mise en place des poussins, il est conseillé de maintenir un taux de ventilation minimal. Ce taux minimum dépend du type de chauffage utilisé. L'objectif est d'évacuer tous les gaz nocifs du bâtiment et de diffuser la chaleur uniformément avant l'arrivée des poussins. Consultez les recommandations du fabricant du système de chauffage pour la ventilation minimale requise à cette étape. Ces recommandations sont normalement affichées sur le boîtier du dispositif de chauffage.

Il existe différents types des équipements de chauffage tel que l'appareil à air pulsé (**Figure I.2**) à coût de fonctionnement réduit mais aussi avec des inconvénients : investissement élevé, dessiccation de l'air (adapter des humidificateurs à la sortie), entretien hygiénique difficile [2].



Figure I.2 : Chauffage à air pulsé.

En revanche il y a les éleveuses au gaz qui sont les plus employées, en peut citer deux types de ces Equipment :

- * Eleveuse avec chapeau cloche (au fuel, au charbon, au gaz).
- * Radiants (gaz, électricité).

Ces appareils ont de gros avantages : investissement modeste, Coût de fonctionnement réduit, Réglage facile, pas de panne générale.

I.9.5. Systèmes de ventilation

Il existe deux systèmes principaux pour la ventilation :

I.9.5.1. La ventilation naturelle : Bâtiment ouvert

La ventilation naturelle se réfère à un bâtiment ouvert comprenant généralement des rideaux (bien que des volets ou des portes puissent également être utilisés) au niveau des murs latéraux (**Figure I.3**).

Dans les bâtiments ouverts, le principe consiste à ouvrir et fermer les rideaux, ou volets, pour générer des courants de convection (vent ou brise) qui vont faire circuler l'air à l'intérieur du bâtiment.

Par temps froid, les ventilateurs de circulation installés à l'intérieur peuvent servir à contrôler la température du bâtiment en faisant circuler l'air chaud qui a monté et s'est accumulé dans le faîte du toit.

Par temps chaud, à moins que le vent ne souffle, l'ouverture des rideaux peut ne pas suffire à soulager correctement les oiseaux. Les ventilateurs de circulation peuvent également aider, dans ce cas, en générant un flux d'air au-dessus des oiseaux qui contribuera à leur confort par effet de courant d'air [6].



Figure I.3 : Exemple de ventilation naturelle.

Les bâtiments à ventilation naturelle requièrent une gestion permanente, 24 heures sur 24, et un contrôle ininterrompu des conditions ambiantes extérieures (température, humidité, vitesse et direction du vent) et intérieures (température, humidité, qualité de l'air et confort des oiseaux).

D'un point de vue général, les bâtiments ouverts se gèrent le mieux lorsque les conditions ambiantes sont proches de la température fixée pour le bâtiment [6].

I.9.5.2. La ventilation dynamique : bâtiments à environnement contrôlé

La ventilation dynamique, dans les bâtiments à environnement contrôlé ou fermé, représente le système de ventilation des bâtiments d'élevage de poulets de chair le plus plébiscité, en raison de sa capacité à mieux contrôler l'environnement intérieur dans des conditions ambiantes variables. La forme la plus répandue de ventilation dans des bâtiments à environnement contrôlé fonctionne en pression négative. Ces bâtiments sont habituellement construits avec des murs latéraux et des ventilateurs d'extraction robustes, ainsi que des entrées d'air automatisées par lesquelles l'air frais est aspiré à l'intérieur du bâtiment [6].

Si l'on veut apporter le meilleur environnement aux oiseaux tout au long du cycle de production et quel que soit le moment de l'année, les bâtiments d'élevage des poulets de chair à environnement fermé doivent être équipés de trois types de ventilation. Il s'agit de :

- * la ventilation minimale ;
- * la ventilation de transition ;
- * la ventilation en tunnel.

I.9.5.2.1. La ventilation minimale

Une ventilation minimale apporte de l'air frais dans le bâtiment et évacue l'air intérieur vicié (afin d'éliminer l'humidité excessive et empêcher l'accumulation de gaz nocifs), tout en maintenant la température intérieure requise.

Un niveau minimum de ventilation doit être assuré en permanence lorsque les oiseaux sont présents dans le bâtiment, quelle que soit la température extérieure. La ventilation minimale peut être utilisée été comme hiver, à tout moment du cycle de production. Cependant, elle est largement utilisée pendant la période de démarrage et par temps frais (c-à-d lorsque la température extérieure est inférieure à la température souhaitée du bâtiment et que la température réelle du bâtiment est égale ou inférieure à la température de référence). La ventilation minimale n'est pas destinée à rafraîchir les oiseaux lorsque les températures sont élevées [6].

L'objectif de la ventilation minimale est de générer un flux d'air très léger à hauteur des oiseaux (0,15 m/s), ce qui est particulièrement important pour les jeunes oiseaux de moins de 10 jours (**figure I.4**).

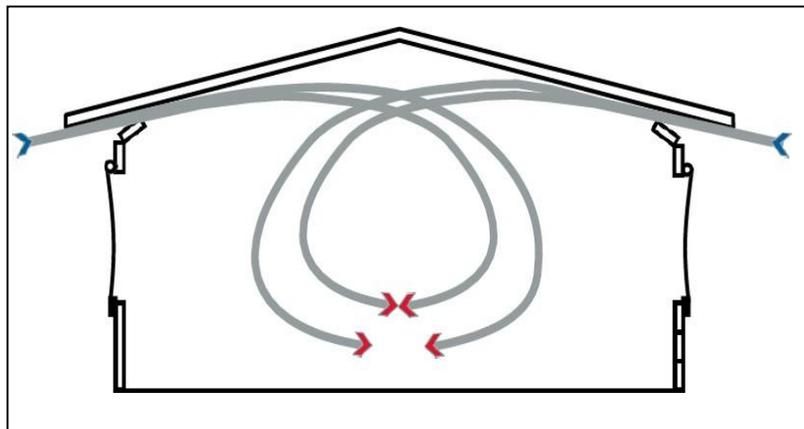


Figure I.4 : Flux d'air correct en ventilation minimale.

La ventilation minimale est réglée par minuterie. Le ventilateur fonctionne avec un programmateur, et non pas en fonction de la température. La gestion correcte du cycle du programmateur détermine la qualité de l'air à l'intérieur du bâtiment.

I.9.5.2.2. Ventilation de transition

L'objectif de la ventilation de transition est d'évacuer l'excès de chaleur du bâtiment lorsque les températures dépassent la température de référence du bâtiment. La ventilation de transition est une procédure de gestion de la température lors de laquelle les ventilateurs ne fonctionnent plus sur minuteur (ventilation minimale) mais commencent à fonctionner en permanence pour contrôler la température (**Figure I.5**).



Figure I.5 : Vue intérieure du bâtiment en mode ventilation de transition.

La répartition des oiseaux indique que l'environnement est confortable.

Lors de la ventilation de transition, un large volume d'air peut pénétrer dans le bâtiment, mais contrairement à la ventilation en tunnel, cet air n'est pas directement dirigé sur les oiseaux. La ventilation de transition est utilisée lorsque l'air extérieur est trop froid et/ou que les oiseaux sont trop jeunes pour la mise en route de la ventilation en tunnel.

I.9.5.2.3. La ventilation en tunnel

La ventilation en tunnel ne doit être actionnée que lorsque la ventilation de transition n'est plus capable de maintenir le degré de confort des oiseaux (c.-à-d. lorsque les oiseaux montrent qu'ils ont trop chaud). La ventilation en tunnel s'utilise par temps chaud à très chaud et généralement lorsque les oiseaux sont plus âgés (**Figure I.6**) [6].

Lorsque la ventilation en tunnel fonctionne, de larges volumes d'air sont aspirés vers le bas sur toute la longueur du bâtiment, permettant un renouvellement rapide de l'air du bâtiment. Ce flux d'air à grande vitesse circule sur les oiseaux et génère un effet de courant d'air qui les aide à se rafraîchir. Il est possible de moduler l'effet de rafraîchissement ressenti par les oiseaux et la vitesse de l'air traversant le bâtiment en modifiant le nombre de ventilateurs. L'effet de courant d'air varie également en fonction de :

- * L'humidité relative ;
- * La densité animale ;
- * D'autres facteurs (tels que le plumage, l'âge et le poids du lot, la température extérieure, etc.) [6].



Figure I.6 : Exemple type d'un bâtiment équipé d'une ventilation en tunnel.

Au démarrage de la ventilation en tunnel, les ventilateurs des murs latéraux doivent s'arrêter (dans le cas d'une ventilation de transition) et les entrées d'air latérales doivent se fermer. Les entrées d'air en tunnel s'ouvrent. Tout l'air qui pénètre dans le bâtiment doit uniquement passer par ces entrées.

I.9.6. Système de rafraîchissement par évaporation

Le rafraîchissement par évaporation consiste à refroidir l'air par évaporation d'eau. Il améliore les conditions ambiantes par temps chaud et optimise la ventilation en tunnel. Le rafraîchissement par évaporation ne doit être utilisé que lorsque le comportement des oiseaux indique que l'effet de courant d'air n'est plus suffisant pour assurer leur confort. Son objectif est de maintenir la température du bâtiment au dernier niveau de confort ressenti par les oiseaux lorsque tous les ventilateurs fonctionnent. Le rafraîchissement par évaporation n'a pas pour but de faire revenir la température du bâtiment au niveau (ou près) de sa température de référence.

La capacité de rafraîchissement par évaporation dépend de l'HR et de l'environnement ambiant extérieur.

- * Plus le taux d'HR est faible, plus la marge d'humidité que l'environnement ambiant peut intégrer est importante et donc, plus importante sera la capacité de rafraîchissement par évaporation.
- * Plus l'HR est élevée, moins le potentiel de rafraîchissement par évaporation de l'air sera important.

Il existe deux principaux systèmes de rafraîchissement par évaporation : par panneau refroidissant et par brumisation [6].

I.9.6.1. Rafraîchissement par panneau refroidissant

Dans les systèmes de rafraîchissement par panneau refroidissant, l'air chaud aspiré par les ventilateurs de la ventilation en tunnel est rafraîchi par son passage au travers d'un filtre imprégné d'eau (le panneau refroidissant).

Les panneaux refroidissants doivent être situés à l'extrémité opposée des ventilateurs en tunnel et également doivent être répartis de chaque côté des murs latéraux, bien que dans certains cas, certains panneaux refroidissants sont installés sur une partie du pignon. Selon les cas, les panneaux refroidissants peuvent être installés dans un local technique (**Figure I.7**) [6].

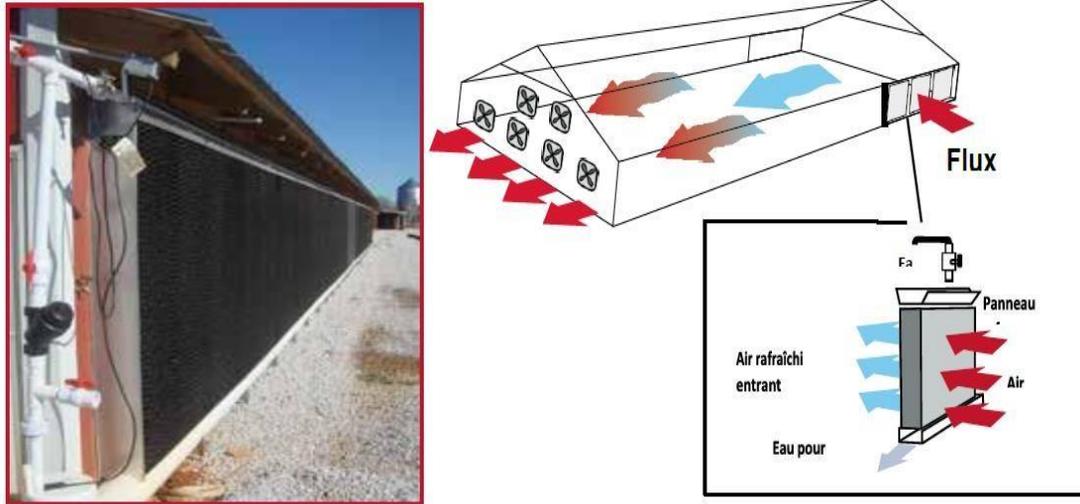


Figure I.7 : Rafrâichissement à panneaux refroidissants avec ventilation en tunnel.

Cette conception et cette configuration des panneaux refroidissants permettent à de grands volumes d'air aspirés par la ventilation en tunnel de traverser la surface des panneaux et d'être rafraîchis avant de pénétrer dans le bâtiment [6].

I.9.6.2. Brumisation/Pulvérisation

Les systèmes par brumisation rafraîchissent l'air entrant par l'évaporation d'eau obtenue par pompage à travers les buses du brumisateur/pulvérisateur.

Les lignes de brumisation doivent être installées près des entrées d'air pour optimiser la vitesse d'évaporation. Des lignes supplémentaires doivent être installées sur toute la surface du bâtiment (**Figure I.8**).



Figure I.8 : Exemple de système de brumisation dans une poulailler.

I.9.7. Litière et humidité

La litière doit toujours être propre et sèche, car une litière humide constitue un risque pour la santé des oiseaux.

Une bonne litière :

- * Est composée de sciures de bois (ripe) ou de coques de riz ;
- * Doit toujours être propre, sèche et légèrement souple. Elle ne doit pas trop coller aux mains ou aux chaussures.
- * Doit avoir une épaisseur de 8 cm la première semaine de vie des oiseaux et environ 6 cm par la suite.
- * Deviendra chaude au toucher lorsque les oiseaux atteignent 11 jours, c'est leur chaleur qui réchauffe la litière.
- * Doit être propre et sèche si une densité de 10 oiseaux / m² est respectée. - Si la litière est humide et/ou collante, dans tout le bâtiment ou une partie du bâtiment, c'est qu'il y a un problème [5].
- * Enlever la litière près des mangeoires s'il y a de la nourriture mélangée à la litière. Sinon, les poulets mangeront les grains à l'extérieur de la mangeoire et ils mangeront en même temps de la litière ce qui entraînera des problèmes digestifs et de la diarrhée [5].

L'humidité relative de l'air est également un facteur important qui influence essentiellement le développement des agents pathogènes et l'état de la litière. En revanche, l'humidité n'a pas d'action directe sur le comportement du poulet, mais peut causer indirectement des troubles.

Ainsi une atmosphère sèche conduit à l'obtention d'une litière poussiéreuse, irritant les voies respiratoires et disséminant les infections microbiennes. A l'inverse, une atmosphère saturée rend le poulet plus fragile surtout si la température est basse. Il se forme des croûtes sur le sol et les risques de microbisme et de parasitisme augmente. L'HR optimale pour l'élevage du poulet se situe entre 40 à 75%. Au-delà, les risques pathologiques peuvent apparaître (maladies respiratoires, coccidiose...) [5].

I.9.8. Eclairage

La lumière est un élément essentiel, contribuant à la croissance des animaux car elles peuvent manger toujours en présence de lumière.

La réglementation locale en matière d'intensité lumineuse doit être respectée. Cependant, une intensité lumineuse de 30-40 lux de 0 à 7 jours de vie et d'au moins 5-10 lux (**Figure I.9**) par la suite, améliorera l'activité alimentaire et la croissance.



Figure I.9 : Exemples d'intensité lumineuse à 10 lux et 30 lux.

Pendant les deux premiers jours, il est important de maintenir les poussins sur une durée d'éclairage maximum (23-24h), pour favoriser la consommation d'eau et d'aliments. Le but de l'éclairage est de permettre aux poussins de voir les mangeoires et les abreuvoirs. L'éclairage ne doit pas être d'une intensité trop forte pour éviter tout nervosisme.

En région chaude, il faut éclairer la nuit, période plus fraîche pour soutenir un niveau, de consommation correcte.

I.10. Conclusion

Jusqu'ici, nous avons présenté une vision générale des bâtiments d'élevage de poulet de chair, la conception, les techniques d'élevage appliquées dans les volailles et les technologies utilisées dans ce domaine. De ce qu'on a vu, on peut dire que les volailles modernes sont conçues pour offrir une meilleure qualité des produits avicoles, notamment le poulet de chair.

CHAPITRE II

**« Généralité sur les automates programmables
industriels »**

II.1. Introduction

L'objectif du développement de la recherche scientifique dans tous les domaines est d'aboutir à un progrès technologique voué principalement au bien-être de l'humanité. Par sa nature de chercher à avoir la quantité, la qualité, le confort avec le minimum d'effort et du coût, l'homme a commencé à réfléchir, à concevoir et à réaliser des systèmes autonomes qui peuvent lui assurer tout dont il a besoin. Par conséquent, les outils et les appareils à énergies musculaires actionnés par des opérateurs sont remplacés par leurs équivalents à énergies électriques, mécaniques ou hydrauliques. La substitution d'opérateurs par ces systèmes définit l'automatisation.

Ce chapitre présente une large formation sur les APIs.

II.2. Définition et Historique des APIs

Un automate programmable est un appareil dédié au contrôle d'une machine ou d'un processus industriel, constitué de composants électroniques, comportant une mémoire programmable par un utilisateur non informaticien, à l'aide d'un langage adapté. En d'autres termes, un automate programmable est un calculateur logique, ou ordinateur, au jeu d'instructions volontairement réduit, destiné à la conduite et la surveillance en temps réel de processus industriels.

Trois caractéristiques fondamentales distinguent totalement l'Automate Programmable Industriel (API) des outils informatiques tels que les ordinateurs (PC industriel ou autres) [7] :

- * il peut être directement connecté aux capteurs et préactionneurs grâce à ses entrées/sorties industrielles,
- * il est conçu pour fonctionner dans des ambiances industrielles sévères (température, vibrations, micro-coupures de la tension d'alimentation, parasites, etc.),
- * et enfin, sa programmation à partir de langages spécialement développés pour le traitement de fonctions d'automatisme fait en sorte que sa mise en œuvre et son exploitation ne nécessitent aucune connaissance en informatique.

Les premiers automates programmables n'effectuaient que la commutation ON/OFF (et vice-versa) avec la possibilité de temporisation, comme les relais. Leurs applications étaient limitées seulement aux procédés répétitifs ainsi qu'à certaines machines. Par contre, leurs avantages consistaient dans une installation plus facile, la visualisation des étapes ; ils possédaient des indicateurs diagnostiques permettant la localisation des pannes. C'était déjà mieux que les relais, en plus de pouvoir être reprogrammé advenant un changement de fonction ou de procédé.

De 1970 à 1974, la technologie des microprocesseurs (du moins les premiers) ajoutèrent une plus grande flexibilité et une « intelligence » à l'automate programmable. Les capacités d'interface avec l'utilisateur s'améliorent. L'automate programmable peut maintenant exécuter les opérations arithmétiques en plus des opérations logiques ; il manipule les données et les adresses ; il effectue la communication avec d'autres automates ou ordinateurs, donnant ainsi une nouvelle dimension aux applications de l'automate programmable.

Ils utilisent une mémoire non-volatile (RAM+Pile, EEPROM ou EAPROM par exemple) pour sauvegarder les instructions. Ces derniers accompliront des fonctions logiques, arithmétiques, de temporisation, de comptage et de manipulation des données. En plus, les fonctions de contrôle PID et d'autres fonctions complexes comme le contrôle numérique de processus sont présentes. Puisque les automates programmables ont été conçus pour accomplir des opérations semblables à celles des relais, la programmation est basée généralement sur la nomenclature des diagrammes en échelle (ou schéma à relais). Des langages de haut niveau ont été aussi implantés sur certains automates afin de produire une plus grande flexibilité de programmation.

II.3. Le rôle de l'automate

L'automate programmable industriel est l'élément du système de contrôle qui commande directement les processus de fabrication en fonction du programme mis en mémoire, l'API reçoit les données des dispositifs d'entrées communiquant avec lui et les utilisent pour surveiller le système de contrôle.

Lorsque le programme demande l'exécution d'une opération, l'API envoie des données aux dispositifs de sortie pour leur ordonner cette opération.

L'API peut être utilisé pour contrôler une tâche simple et répétitive ou il peut se connecter à d'autres API ou à un ordinateur pour intégrer le contrôle d'un processus complexe.

Un API est le cœur de la partie commande et il est en relation avec les autres parties du système grâce à son interface d'entrée-sortie. Il effectue des fonctions d'automatisme programmées telles que :

- * Logique combinatoire ;
- * Logique séquentielle ;
- * Temporisation ;
- * Comptage ;
- * Calculs numériques ;
- * Asservissement et régulation.

II.4. Architecture d'un API

II.4.1. Aspect extérieur

Les APIs peuvent être de type **compact** ou **modulaire** ;

- * **API compact**

Les automates type compact ou micro automates intègrent le processeur, l'alimentation, les interfaces d'entrées/sorties. Selon les modèles et les fabricants, ils peuvent réaliser certaines fonctions supplémentaires (comptage rapide, E/S analogiques ...) et recevoir des extensions en nombre limité.

Ces automates sont de fonctionnement simple et sont généralement destinés à la commande de petits automatismes.

Exemples : LOGO de Siemens, ZELIO de Schneider, S7-200 de Siemens...



Figure II.3 : Automate compact (LOGO ; SIEMENS).

* API modulaire

Les automates type modulaire ; le processeur, l'alimentation et les interfaces d'entrées/sorties résident dans des unités séparées (modules) et sont fixées sur un ou plusieurs racks contenant le "fond de panier" (bus plus connecteurs).

Ces automates sont intégrés dans les automatismes complexes où puissance, capacité de traitement et flexibilité sont nécessaires.

Exemples : S7-1200 de Siemens, Schneider Electric Modicon M580, Mitsubishi Electric iQ-Series...



Figure II.4 : Automate modulaire (Mitsubishi).

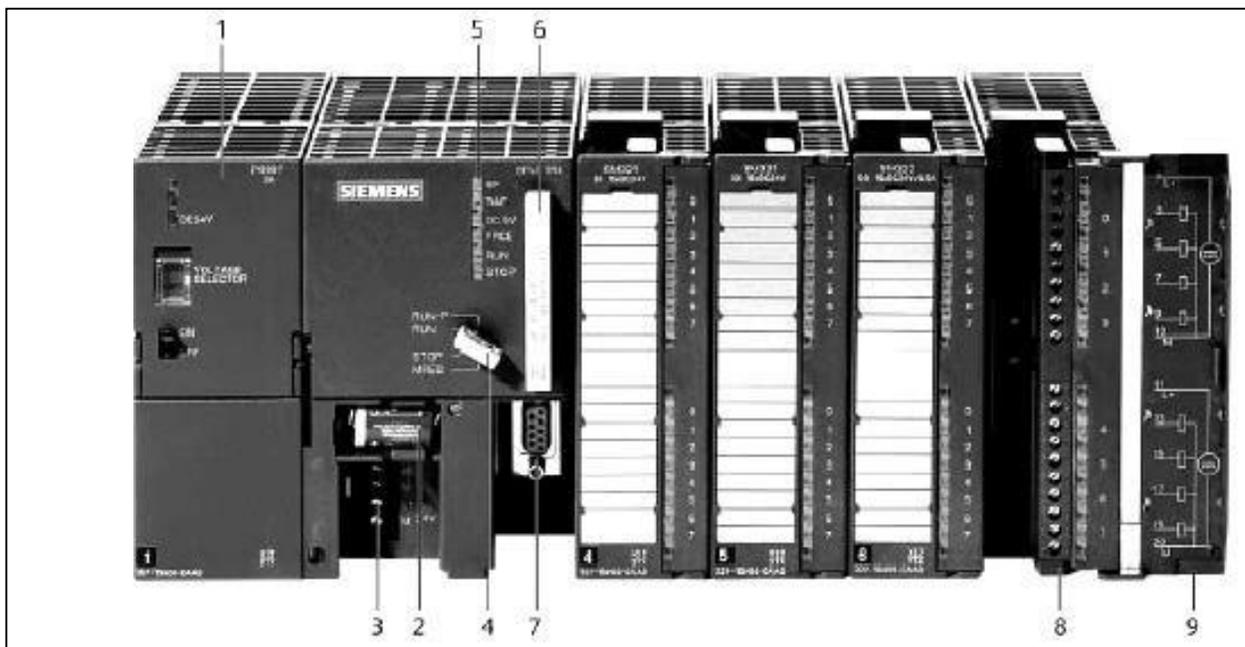


Figure II.5 : Automate modulaire (SIEMENS).

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

II.4.2. Structure interne

L'automate programmable reçoit les informations relatives à l'état du système et puis commande les préactionneurs suivant le programme inscrit dans sa mémoire.

Un API se compose de quatre grandes parties :

- * Le processeur.
- * Module de stockage (mémoire) : RAM, ROM, PROM, EPROM, EEPROM² et mémoire flash.
- * Les interfaces Entrées/Sorties (Modules E/S) :
 - TOR (Logique).
 - Analogique.
 - Numérique.
- * Module d'alimentation (110/220Vac ou 24Vcc).

² **ROM « Read Only Memory »** : Elle est programmée par le constructeur et son programme ne peut être modifié.

PROM « Programmable ROM » : Elle est livrée non enregistrée par le fabricant. Lorsque celle-ci est programmée, on ne peut pas l'effacer

EPROM « Erasable PROM » : C'est une mémoire PROM effaçable par un rayonnement ultraviolet intense.

EEPROM « Electrically EPROM » : C'est une mémoire PROM programmable plusieurs fois et effaçable électriquement

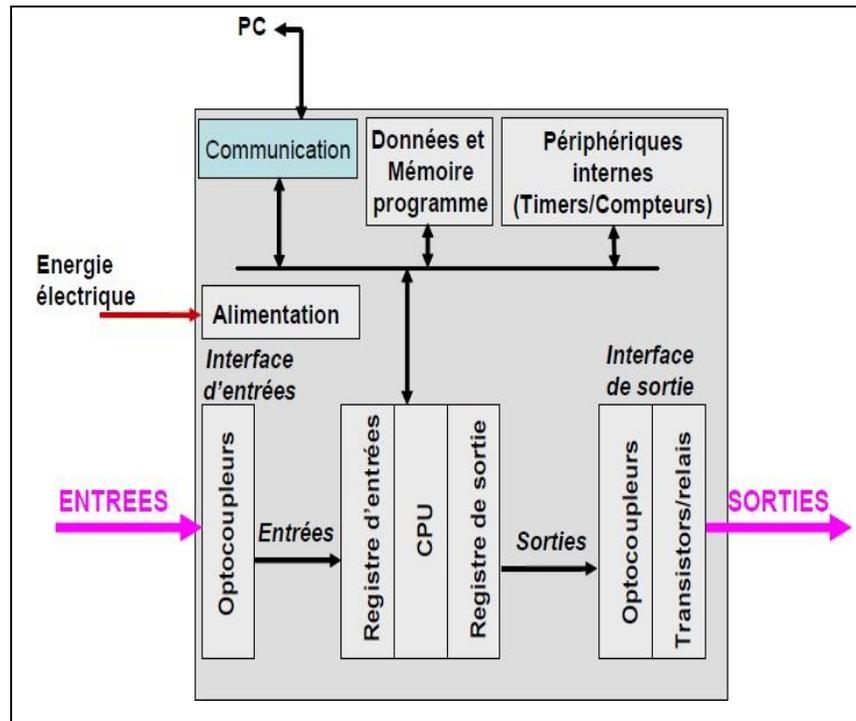


Figure II.6 : Structure interne d'un API.

II.4.2.1. Module d'alimentation

Ce module permet l'alimentation en tension continue nécessaire au bon fonctionnement de l'automate programmable ainsi que le circuit de charge. Il convertit la tension du réseau (AC 220) en tension de service (DC 24V, 15V ou 5V). Ce module doit posséder une bonne protection face aux microcoupures du réseau, ainsi qu'un transformateur d'isolement pour lutter contre les perturbations du même réseau [8].

II.4.2.2. Unité centrale

L'unité centrale (CPU) est l'élément le plus important dans l'automate programmable, elle peut être considérée comme le cerveau du système. Elle est constituée de deux composants principaux :

- * Le Processeur ;
- * Les mémoires.

II.4.2.2.1. Le Processeur

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

- * Lecture des entrées/sorties ;
- * Surveillance et diagnostique de l'automate par une série de tests lancés à la mise sous tension ou cycliquement en cours de fonctionnement ;
- * Dialogue avec le terminal programmation aussi bien pour l'écriture et la mise au point du programme qu'en cours d'exploitation pour des réglages (contrôle) ou des vérifications de données (supervision).

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

Les principaux registres existants dans un processeur sont :

➤ **L'accumulateur**

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

➤ **Le registre d'instruction**

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

➤ **Le registre d'adresse**

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

➤ **Le registre d'état**

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

➤ **Les piles**

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

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

II.4.2.2.2. Les mémoires [8]

Un système à processeur est toujours accompagné d'un ou de plusieurs types de mémoires.

Les automates programmables industriels possèdent pour la plupart les mémoires suivantes :

➤ **Mémoire de travail**

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

➤ **Mémoire système**

Elle contient les éléments de mémoire que chaque CPU met à la disposition du programme utilisateur comme, par exemple, mémoire image des entrées, mémoire image des sorties, mémentos, temporisation et compteurs. La mémoire système contient, en outre, la pile des blocs et la pile des interruptions.

Elle fournit aussi la mémoire temporaire allouée au programme (pile des données locales).

➤ **Mémoire de chargement**

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

Elle peut être soit une mémoire vive (RAM), soit une mémoire EPROM.

➤ **Mémoire RAM non volatile**

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

➤ **Mémoire ROM**

Ce sont des mémoires mortes dont l'utilisateur ne peut que lire le contenu, dans le cas des automates programmables, elle contient le système d'exploitation qui gère la CPU.

II.4.2.3. Modules d'entrées/sorties (E/S)

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

Beaucoup d'automates assurent cet interfaçage par des modules amovibles ou modulaires.

D'autres automates ont une structure monobloc, avec des modules intégrés dans un châssis de base, (cas des automates de télémécanique TSX17 et SIMATIC S7-314 IFM).

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

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

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

➤ **Entrées/sorties TOR (Tout ou Rien)**

Ces modules traitent une information qui ne peut prendre que deux états (vrai ou faux, 0 ou 1), ils constituent l'interface entre l'API et les différents capteurs et préactionneurs présents.

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

- * En continu : 24V, 48V ;
- * En alternatif : 24V, 48V, 100/120V.

➤ **Entrées/Sorties analogiques**

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

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

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

➤ **Modules spécialisés**

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

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

On peut citer :

- * **Les cartes de comptage rapide** Elles permettent de saisir des événements plus courts que la durée du cycle travaillant à des fréquences qui peuvent dépasser 10kHz ;
- * **Les coupleurs** Leur intérêt est de diminuer le câblage en réalisant la liaison avec détecteur, ou actionneurs au plus près de ceux-ci, ce qui a pour effet d'améliorer la précision de mesure.

La liaison entre le boîtier déporté et l'unité centrale s'effectue par le biais d'un réseau de terrain suivant des protocoles bien définis. L'utilisation de la fibre optique permet de porter la distance à plusieurs kilomètres.

II.4.2.4. Les liaisons de communication

Elles s'effectuent :

- * Avec l'extérieur par des borniers, sur lesquels arrivent des câbles transportant les signaux électriques, ou par liaison avec un bus de terrain.
- * Avec l'intérieur par des bus, liaisons parallèles entre les divers éléments. Il existe plusieurs types de bus, car on doit transmettre des données, des états, des adresses.

II.4.2.5. Eléments auxiliaires

Il s'agit principalement :

- * **D'un ventilateur** qui est en général indispensable dans les châssis comportant de nombreux modules ou dans le cas où la température ambiante est susceptible de devenir assez élevée (plus de 40 °C) ;
- * **De support mécanique** Il peut s'agir d'un rack (structure métallique) ;
- * **Des indicateurs d'état** concernant la présence de tension, la charge de la batterie, le bon fonctionnement de l'automate, etc.

II.4.3. Structure d'un système automatisé

Tout système automatisé peut se décomposer selon le schéma ci-dessous :

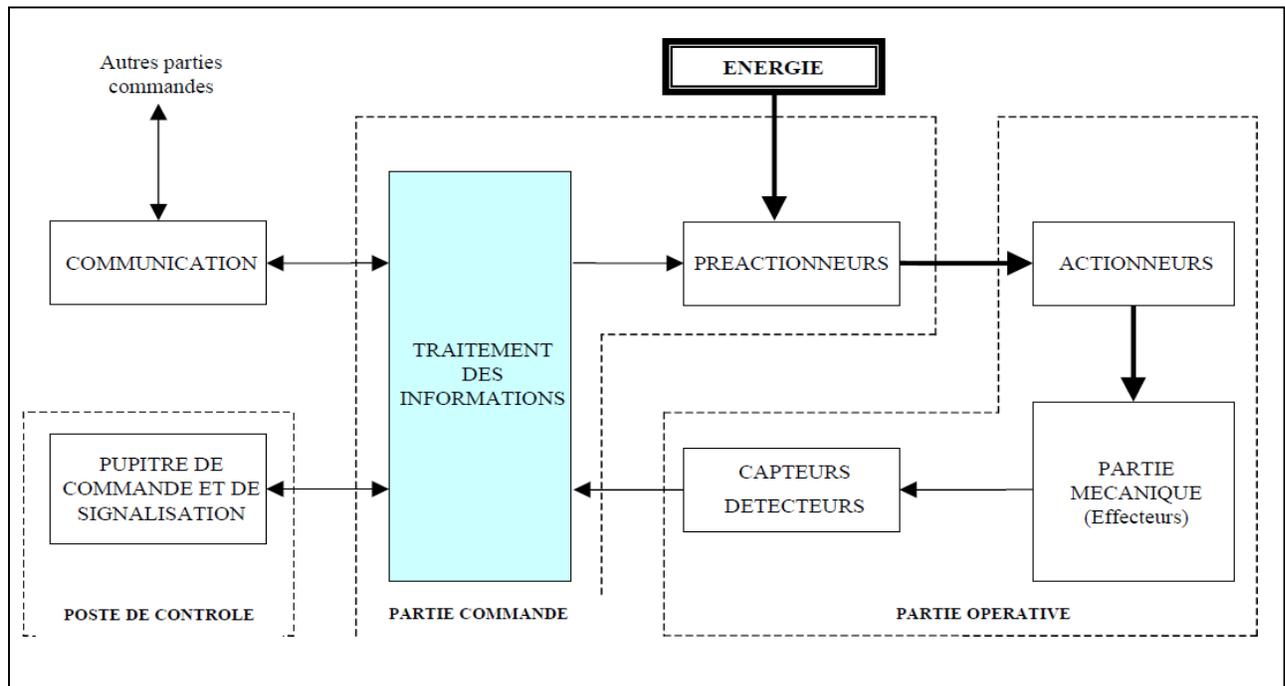


Figure II.7 : Schéma de principe d'un système automatisé.

❖ Partie opérative :

Elle agit sur la matière d'œuvre afin de lui donner sa valeur ajoutée.

Les actionneurs (moteurs, vérins) agissent sur la partie mécanique du système qui agit à son tour sur la matière d'œuvre.

Les capteurs / détecteurs permettent d'acquérir les divers états du système.

❖ Partie commande :

Elle donne les ordres de fonctionnement à la partie opérative.

Les préactionneurs permettent de commander les actionneurs ; ils assurent le transfert d'énergie entre la source de puissance (réseau électrique, pneumatique ...) et les actionneurs.

Exemple : contacteur, distributeur ...

Ces préactionneurs sont commandés à leur tour par le bloc traitement des informations.

Celui-ci reçoit les consignes du pupitre de commande (opérateur) et les informations de la partie opérative transmises par les capteurs / détecteurs.

En fonction de ces consignes et de son programme de gestion des tâches (implanté dans un automate programmable ou réalisé par des relais (on parle de logique câblée)), elle va commander les préactionneurs et renvoyer des informations au pupitre de signalisation ou à d'autres systèmes de commande et/ou de supervision en utilisant un réseau et un protocole de communication.

❖ Poste de contrôle :

Composé des pupitres de commande et de signalisation, il permet à l'opérateur de commander le système (marche, arrêt, départ cycle ...).

Il permet également de visualiser les différents états du système à l'aide de voyants, de terminal de dialogue ou d'interface homme-machine (IHM).

II.4.4. Les caractéristiques principales d'un API

Les caractéristiques principales d'un API sont :

- * Compact ou modulaire.
- * Tension d'alimentation.
- * Nombres d'entrées/sorties intégrés.
- * Taille mémoire.
- * Sauvegarde (EPROM, EEPROM, pile, ...).
- * Modules complémentaires (analogique, Communication).
- * Nombre de compteurs et de temporisateurs.
- * Langage de programmation.

II.5. Son principe de fonctionnement

Les caractéristiques d'un automate programmable interagissent et fonctionnent généralement par cycles. L'utilisateur définit les tâches requises grâce à la logique programmable de l'automate.

Un processus typique par étapes peut ressembler à ce qui suit :

- * L'automate programmable vérifie l'état des modules d'entrée et de tout dispositif connecté.
- * L'unité centrale prend en charge les entrées puis exécute la logique définie par l'utilisateur par le biais du programme.
- * Les modules de sortie de l'automate envoient le signal pour effectuer des actions physiques aux dispositifs connectés. Les commandes de sortie typiques sont la mise en marche ou l'arrêt, l'augmentation ou la diminution de la vitesse, etc.
- * L'automate programmable effectue un contrôle global pour signaler les conditions de fonctionnement des dispositifs connectés. À ce stade, l'automate exécute des protocoles de communication pour envoyer des données à un autre système ou à une interface homme machine qui traduit les données.
- * L'automate programmable met fin ou redémarre le processus tel que défini par le programme.

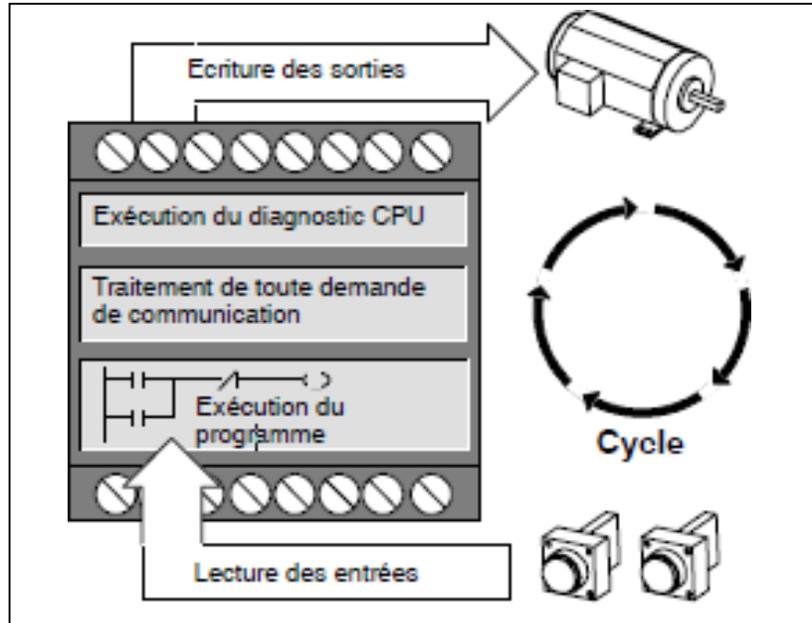


Figure II.8 : Principe de fonctionnement des APIs.

II.6. Critères du choix d'un API

Après l'établissement du cahier des charges et à part le critère de prix qui fait partie des critères les plus considérés lors du choix d'un automate, il revient à l'utilisateur de regarder sur le marché l'automate le mieux adapté aux besoins, en considérant un certain nombre de critères importants :

- * Le nombre et la nature d'entrées / sorties ;
- * Le temps de réponse (performance de processeur) ;
- * Les modules complémentaires (analogique, communication...)
- * La nature du traitement (temporisation, comptage, ...)
- * La communication avec les autres systèmes ;
- * Les moyens de sauvegarde du programme (capacité de la mémoire) ;
- * La fiabilité, robustesse, immunité aux parasites ;
- * Les moyens de dialogue et le langage de programmation ;
- * La documentation, le service après-vente, durée de la garantie, la formation.

II.7. Avantages [9]

Les avantages des APIs sont :

- * Evolutivité : Très favorable à l'évolution et très utilisé en reconstruction d'armoire.
- * Fonctions : Assure les fonctions conduites, dialogue, communication et sûreté.
- * Taille des applications : Gamme importante d'automate.
- * Vitesse : Temps de cycle de quelques ms.
- * Modularité : Haute modularité. Présentation en rack.
- * Développement d'une application et documentation : Très facile des outils de programmation de plus en plus puissant.
- * Architecture de commande : Centralisée ou décentralisée avec l'apparition d'une offre importante de choix de réseaux.
- * Mise en œuvre : Mise au point rendu plus facile avec l'apparition des outils de simulation de PO.
- * Maintenance : Echange standard et aide au diagnostic intégré.
- * Portabilité d'une application.

II.8. Les domaines d'application des automates programmables

Les APIs sont utilisés dans diverses applications dans des secteurs tels que la métallurgie et sidérurgie (sécurité), l'industrie automobile, l'industrie chimique, pétrolière, et le secteur de l'énergie. Le champ d'application des automates programmables augmente considérablement en fonction du développement de toutes les différentes technologies où ils sont appliqués.

Quelques exemples d'application [10] :

- * Fabricant industriel
- * Feu de signalisation
- * L'industrie du verre
- * Industrie du papier
- * Fabrication de ciment

II.9. Comment programmons-nous un API ?

La programmation d'un API s'effectue de trois manières différentes :

- * Sur l'API lui-même à l'aide de touches.
- * Avec une console de programmation reliée par un câble spécifique à l'API.
- * Avec un ordinateur et un logiciel approprié avec lequel la programmation est plus conviviale, communiquant avec l'automate par une liaison série RS-232, RS-485 ou câble Ethernet.

II.9.1. Langages d'API

La norme CEI 61131-3 définit cinq langages qui peuvent être utilisés pour la programmation des automates programmables industriels. Ces cinq langages sont [11] :

- * **FDB (Function Block Diagram), schéma par blocs** : ce langage permet de programmer graphiquement à l'aide des blocs, représentant des variables, des opérateurs ou des fonctions. Il permet de manipuler tous les types des variables, la construction d'équations complexes à partir des opérateurs standards de fonctions ou de blocs fonctionnels.

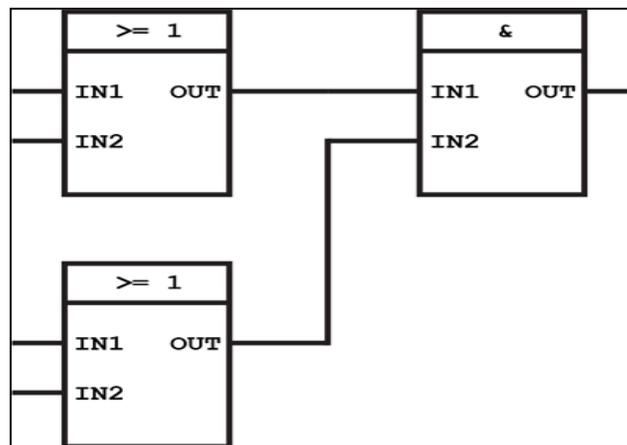


Figure II.9 : Exemple d'un programme avec FDB.

- * **IL (Instruction List), liste d'instructions** : ce langage textuel de bas niveau a une instruction par ligne. Il peut être comparé au langage assembleur. Un programme IL est une liste d'instructions. Chaque instruction doit commencer par une nouvelle ligne, et doit contenir un opérateur, si un commentaire est attaché à l'instruction, il doit être le dernier élément de la ligne. Des lignes vides peuvent être insérées entre des instructions. Un commentaire peut être posé sur une ligne sans instruction.

```
! %L0: LD      %I1.0
      ANDN   %M12
      OR(    %TM4.Q
      AND    %M17
      )
      AND    %I1.7
      ST     %Q2.5
! %L5: LD      %I1.10
      ANDN   %Q2.3
      ANDN   %M27
      IN     %TM0
      LD     %TM0.Q
      AND    %M25
      AND    %M000.X5
      [ %M0015 := %M0018+500]
```

Figure II.10 : Exemple d'un programme avec IL.

- * **LD (Ladder Diagram), schéma à relais** : ce langage graphique est essentiellement dédié à la programmation des équations booléennes (vrai/faux). Le langage LD est une représentation graphique d'équations booléennes combinant des contacts (en entrée) et des relais (en sortie). Il permet la manipulation de données booléennes, à l'aide de symboles graphiques organisés dans un diagramme comme les éléments d'un schéma électrique à contacts. Les diagrammes LD sont limités à gauche et à droite par des barres d'alimentation.

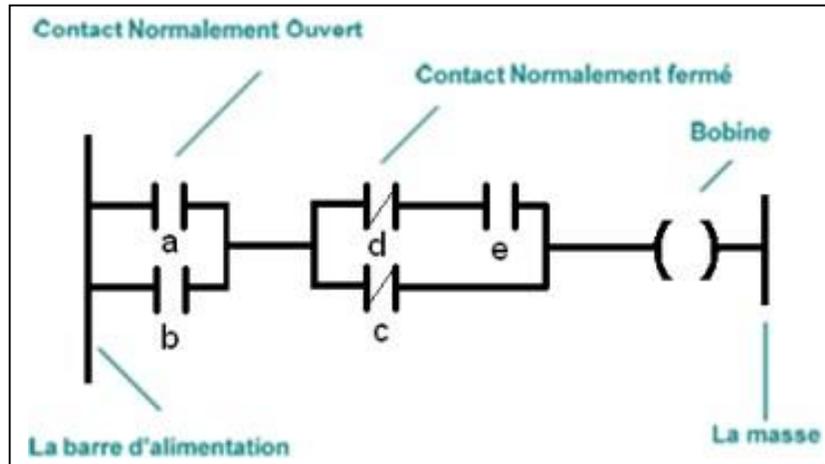


Figure II.11 : Exemple d'un programme avec LD.

- * **SFC (Sequential Function Chart), langage GRAFCET** : ce langage, de haut niveau, est un langage graphique utilisé pour décrire et programmer les procédés séquentiels. Le procédé est représenté comme une suite connue d'étapes, reliées entre elles par des transitions, une condition booléenne est attachée à chaque transition et chaque étape peut avoir une action reliée.

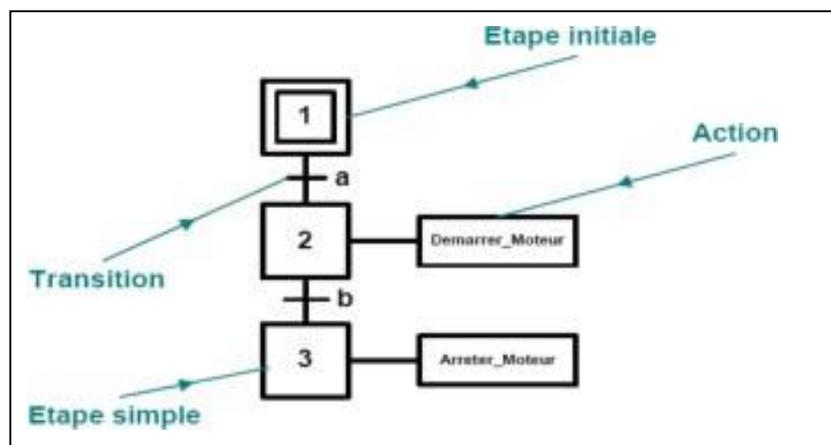


Figure II.12 : Exemple d'un programme avec SFC.

- * **ST (Structured Text), texte structuré** : c'est un langage textuel de haut niveau dédié aux applications d'automatisation, ce langage est principalement utilisé pour décrire les procédures complexes, difficilement modélisables avec les langages graphiques.

```
IF %M0 THEN
  FOR %M0099 := 0 TO %I DO
    IF %M0000 [%M0099] > 0 THEN
      %M0010 := %M0000 [%M0099];
      %M0011 := %M0099;
      %M1 := TRUE;
      EXIT;          (*Sortie de la boucle FOR*)
    ELSE
      %M1 := FALSE;
    END_IF;
  END_FOR;
ELSE
  %M1 := FALSE;
END_IF;
```

Figure II.13 : Exemple d'un programme avec ST.

II.9.1.1. Objets communs à tous les langages

Toute expression, constante ou variable, utilisée dans un programme doit être caractérisée par un type, les types de base sont :

- * Booléen : **BOOL** (Vraie ou Faux qui sont équivalent à 1 ou 0).
- * Entier : **DINT** (c'est un nombre signé entre -2147483647 et +2147483647. Il exprimé dans l'une des bases suivantes : décimale, hexadécimale, octale ou binaire.)
- * Réel : **REAL** (il prend 1 bit de signe +23 bits de mantisse +8 bits d'exposant compris entre -37 et +37.)
- * Temporisation : **TIME** (elle ne peut jamais être négative et commencer par T# ou TIME#).
- * Chaîne : **STRING** (elle doit être précédée et suivie par une apostrophe, et ne doit jamais excéder 255 caractères). Le caractère spécial ('\$') est utilisé pour insérer des caractères non imprimables.

II.9.1.2. Comment choisit-on la langue ? [12]

Les fournisseurs d'automates programmables et les sociétés de services spécialisées en automatismes, proposent un large choix de langages de programmation. Ces programmes fonctionnent sur des plates-formes matérielles adaptées aux différentes applications. Dans les bureaux d'étude, les utilisateurs ont la possibilité de réaliser leurs programmes sur un PC.

Pour les ateliers aux ambiances plus sévères, ont été conçues des consoles de programmation très fiables, adaptées aux besoins. Dans certains cas d'applications, cependant, de simples ordinateurs portables peuvent assurer ces tâches avec succès.

L'un des principaux soucis de l'automaticien est de réduire le temps de développement du programme, pas étonnant donc que l'on utilise des sous-programmes destinés à traiter les fonctions répétitives. SIEMENS, premier à avoir adopté cette démarche avec son logiciel Step7, "programmation structurée". Il s'agit d'un assemblage de modules fonctionnels (livrés par le fournisseur). La qualité du programme est garantie car les modules peuvent être testés peu à peu avant leur mise en œuvre. Sa simplicité entraîne un gain de temps. Aujourd'hui, tous les constructeurs proposent ce type de programmation.

II.10. Sécurité [13]

Les systèmes automatisés sont, par nature, source de nombreux dangers (tensions utilisées, déplacements mécaniques, jets de matière sous pression ...).

Placé au cœur du système automatisé, l'automate se doit d'être un élément fiable car :

- * Un dysfonctionnement de celui-ci pourrait avoir de graves répercussions sur la sécurité des personnes,
- * Les coûts de réparation de l'outil de production sont généralement très élevés,
- * Un arrêt de la production peut avoir de lourdes conséquences sur le plan financier.

Aussi, l'automate fait l'objet de nombreuses dispositions pour assurer la sécurité :

- **Contraintes extérieures** : l'automate est conçu pour supporter les différentes contraintes du monde industriel et a fait l'objet de nombreux tests normalisés (tenue aux vibrations, CEM ...)

- **Coupures d'alimentation** : l'automate est conçu pour supporter les coupures d'alimentation et permet, par programme, d'assurer un fonctionnement correct lors de la réalimentation (reprises à froid ou à chaud)
- **Mode RUN/STOP** : Seul un technicien peut mettre en marche ou arrêter un automate et la remise en marche se fait par une procédure d'initialisation (programmée)
- **Contrôles cycliques** :
 - Procédures d'autocontrôle des mémoires, de l'horloges, de la batterie, de la tension d'alimentation et des entrées / sorties
 - Vérification du temps de scrutation à chaque cycle appelée Watchdog (chien de garde), et enclenchement d'une procédure d'alarme en cas de dépassement de celui-ci (réglé par l'utilisateur)
- **Visualisation** : Les automates offrent un écran de visualisation où l'on peut voir l'évolution des entrées / sorties

La défaillance d'un automate programmable pouvant avoir de graves répercussions en matière de sécurité, les normes interdisent la gestion des arrêts d'urgence par l'automate ; celle-ci doit être réalisée en technologie câblée.

On peut également ajouter des modules de sécurité à l'automate (sécurité des machines).

Il existe enfin des automates dits de sécurité (APIdS) qui intègrent des fonctions de surveillance et de redondance accrues et garantissent la sécurité des matériels.

« Partie pratique »

CHAPITRE III

**« Programmation et supervision avec TIA PORTAL et
IHM »**

III.1. Introduction

Ce chapitre sera consacré à la programmation du système d'un poulailler automatisé (Température, ventilation). Nous parlerons aussi sur les outils logiciels de programmation utilisés pour programmer l'automate SIEMENS ainsi que son IHM (HMI en anglais) en général.

III.2. Matériels utilisés d'exemple d'un poulailler réel

Pour le choix du matériel, nous nous sommes inspirés sur un poulailler réel se situant à Oued-Souf. Son propriétaire utilise ;

- 1) Six ventilateurs
Deux (60cm×60cm), deux (100cm×100cm) et deux (140cm×140cm),
- 2) Deux chauffages
- 3) Deux climatiseurs
- 4) Deux lignes de lumière
- 5) Huit fenêtres

III.3. TIA PORTAL

TIA PORTAL est un environnement de développement intégré (IDE) développé par SIEMENS pour la programmation, la configuration et la mise en service de leurs automates programmables et systèmes de contrôle industriels. Il offre une interface conviviale pour la programmation et la visualisation, ainsi que des outils pour la gestion de projets, la simulation et le débogage.

Grâce à TIA PORTAL, les utilisateurs peuvent concevoir, programmer et mettre en service leurs systèmes de contrôle avec une plus grande efficacité, ce qui permet de réduire les coûts et les délais de développement. Il facilite également la maintenance et la gestion des systèmes, offrant ainsi une solution complète pour l'automatisation industrielle.

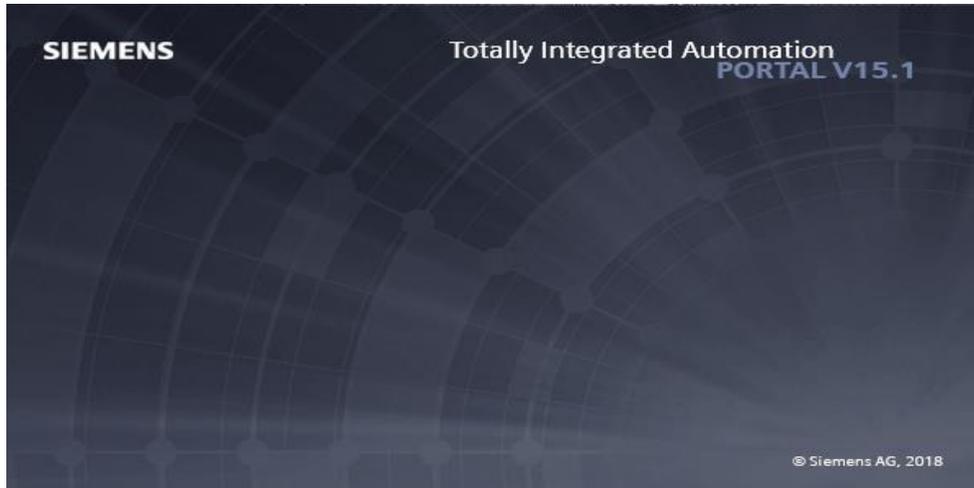


Figure III.1 : Logo de TIA PORTAL (V15.1).

III.3.1. Description du logiciel TIA PORTAL

Le logiciel TIA PORTAL est le nouvel environnement de travail SIEMENS qui permet de mettre en œuvre des solutions d'automatisation avec un système d'ingénierie intégré comprenant les logiciels SIMATIC Step7 et SIMATIC WinCC [14].

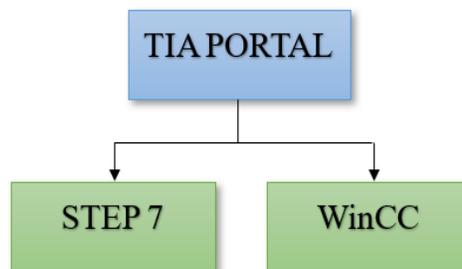


Figure III.2 : Illustration de la construction de TIA PORTAL.

III.3.1.1. SIMATIC STEP7 (S7)

SIMATIC STEP7 est le logiciel d'ingénierie le plus connu et le plus utilisé dans le monde pour l'automatisation industrielle. Il fonctionne sous le logiciel TIA PORTAL. Avec SIMATIC S7, l'utilisateur peut configurer, programmer, tester et faire le diagnostic de tous les automates SIMATIC [14].

En d'autres termes, STEP7 est une partie intégrante de TIA PORTAL. Il s'agit d'un module spécifique au développement de logiciels pour les contrôleurs SIMATIC S7 de SIEMENS.

III.3.1.2. SIMATIC WinCC

SIMATIC WinCC est un logiciel de supervision et de gestion de processus développé par Siemens. Il est utilisé pour la visualisation, le contrôle et la surveillance des systèmes d'automatisation industrielle. WinCC permet aux utilisateurs de créer des IHM conviviales pour surveiller et contrôler des processus industriels.

Avec SIMATIC WinCC, les opérateurs peuvent afficher en temps réel des données provenant de différents équipements et systèmes, tels que des APIs, des capteurs, des actionneurs, etc. Ils peuvent également interagir avec ces systèmes en effectuant des opérations de contrôle, de réglage de paramètres et de dépannage.

WinCC offre une variété de fonctionnalités, y compris la création de graphiques, de tableaux de bord, de tendances, d'alarmes et de rapports. Il prend également en charge la communication avec d'autres systèmes, tels que des bases de données, des systèmes SCADA et des applications tierces. WinCC est souvent utilisé en combinaison avec les contrôleurs SIMATIC S7 et est intégré à l'environnement de développement TIA PORTAL pour une intégration plus transparente des projets d'automatisation industrielle.

III.3.2. Vues du TIA PORTAL

Lorsqu'on lance TIA PORTAL, l'environnement de travail se décompose de deux types de vue :

- * **Vue du portail** : elle est axée sur les tâches à exécuter et sa prise en main est très rapide [14].
- * **Vue du projet** : elle comporte une arborescence avec les différents éléments du projet, les éditeurs requis s'ouvrent en fonction des tâches à réaliser. Données, paramètres et éditeurs peuvent être visualisés dans une seule et même vue [15].

III.3.2.1. Vue du portail

Chaque élément de TIA PORTAL permet de traiter une catégorie de tâche (action) la fenêtre affiche la liste des actions qui peuvent être réalisées pour la tâche sélectionnée.

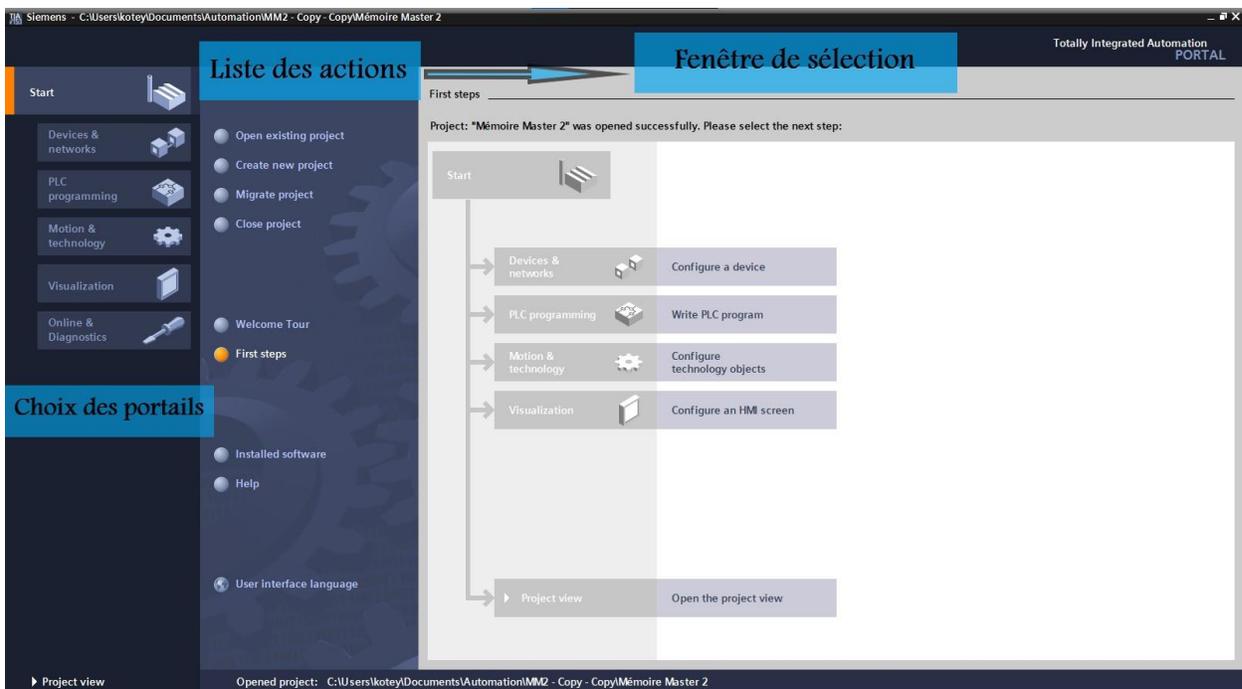


Figure III.3 : Vue du TIA PORTAL V15.1.

III.3.2.2. Vue du projet

L'élément « Projet » contient l'ensemble des éléments et des données nécessaires pour mettre en œuvre la solution d'automatisation souhaitée.

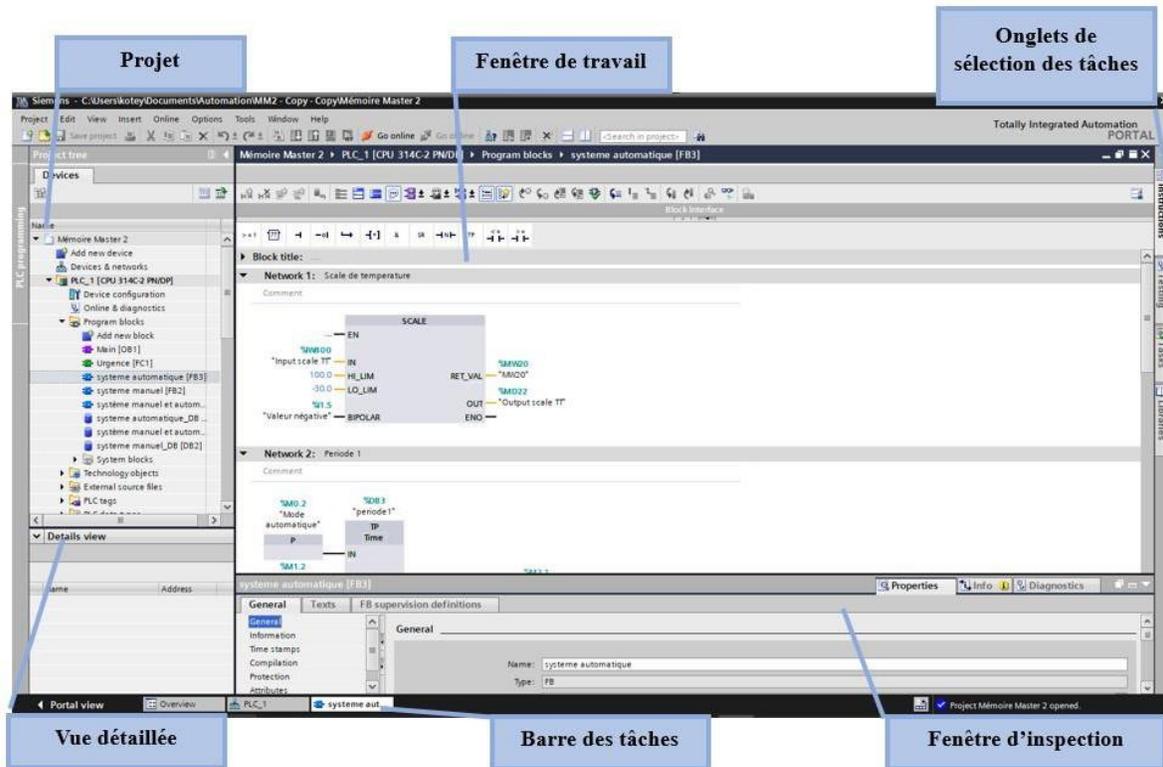


Figure III.4 : Vue du projet.

- * **La fenêtre de travail** permet de visualiser les objets sélectionnés dans le projet pour être traités. Il peut s'agir des composants matériels, des blocs de programme, des tables des variables, des IHM, ...
- * **La fenêtre d'inspection** permet de visualiser des informations complémentaires sur un objet sélectionné ou sur les actions en cours d'exécution (propriété du matériel sélectionné, messages d'erreurs lors de la compilation des blocs de programme, ...).
- * **Les onglets de sélection de tâches** ont un contenu qui varie en fonction de l'objet sélectionné (configuration matérielle, bibliothèques des composants, bloc de programme, instructions de programmation).

Cet environnement de travail contient énormément de données. Il est possible de masquer ou réduire certaines de ces fenêtres lorsque l'on ne les utilise pas.

Il est également possible de redimensionner, réorganiser, désancrer les différentes fenêtres [14].

III.3.3. Les avantages du TIA PORTAL [14]

- * Programmation intuitive et rapide : avec des éditeurs de programmation développés SCL, CONT, LOG, LIST et GRAPH.
- * Efficacité accrue grâce aux innovations linguistiques de STEP7 : programmation symbolique uniforme, Calculate Box, ajout de blocs durant le fonctionnement, et bien plus encore.
- * Performance augmentée grâce à des fonctions intégrées : simulation avec PLCSIM, télémaintenance avec TeleService et diagnostic système cohérent.
- * Technologie flexible : Fonctionnalité motion control évolutive et efficace pour les automates S7-1500 et S7-1200.
- * Sécurité accrue avec « Security Integrated » : Protection du savoir-faire, protection contre la copie, protection d'accès et protection contre la falsification.
- * Environnement de configuration commun avec pupitres IHM et entraînements dans l'environnement d'ingénierie TIA PORTAL.

Remarque :

Tout ce que nous avons mentionné n'est qu'une petite partie de ce logiciel. Mais celui qui veut bien le connaître, SIEMENS a mis sur lui des documentations très détaillées qui peuvent être consultées.

III.4. Le programme d'automatisation de la température du poulailler

III.4.1. Etapes d'automatisation logicielle

Dans cette partie, nous présentons les tâches fondamentales nécessaires à la planification d'une solution d'automatisation à base d'un automate programmable (API). Il existe de nombreuses méthodes pour concevoir une solution d'automatisation.

La figure suivante représente ces tâches :

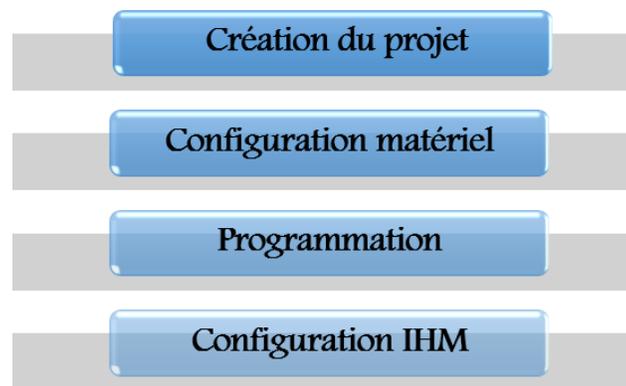


Figure III.5 : Planification d'une solution d'automatisation.

III.4.1.1. Configuration matérielle

Après la création du projet nous allons configurer les appareils que nous avons choisi parmi le large choix d'automates et d'IHM proposés par TIA PORTAL.



Figure III.6 : Planification de la configuration.

III.4.1.1.1. Ajout de l'API

On choisit un API dans la liste proposée.

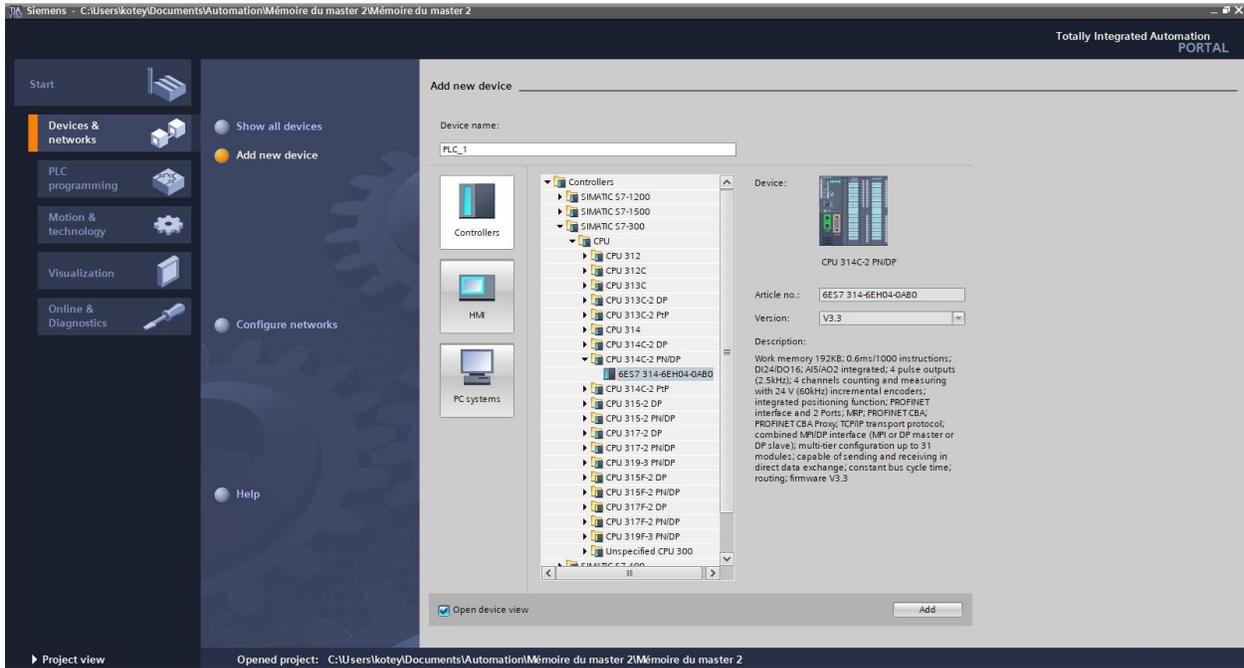


Figure III.7 : Ajout d'API.

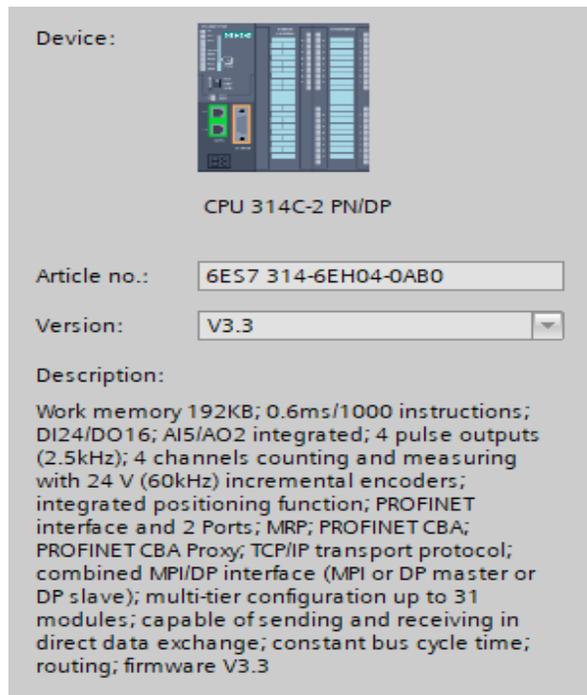


Figure III.8 : Description de l'appareil ajouté.

III.4.1.1.2. Ajout de l'IHM

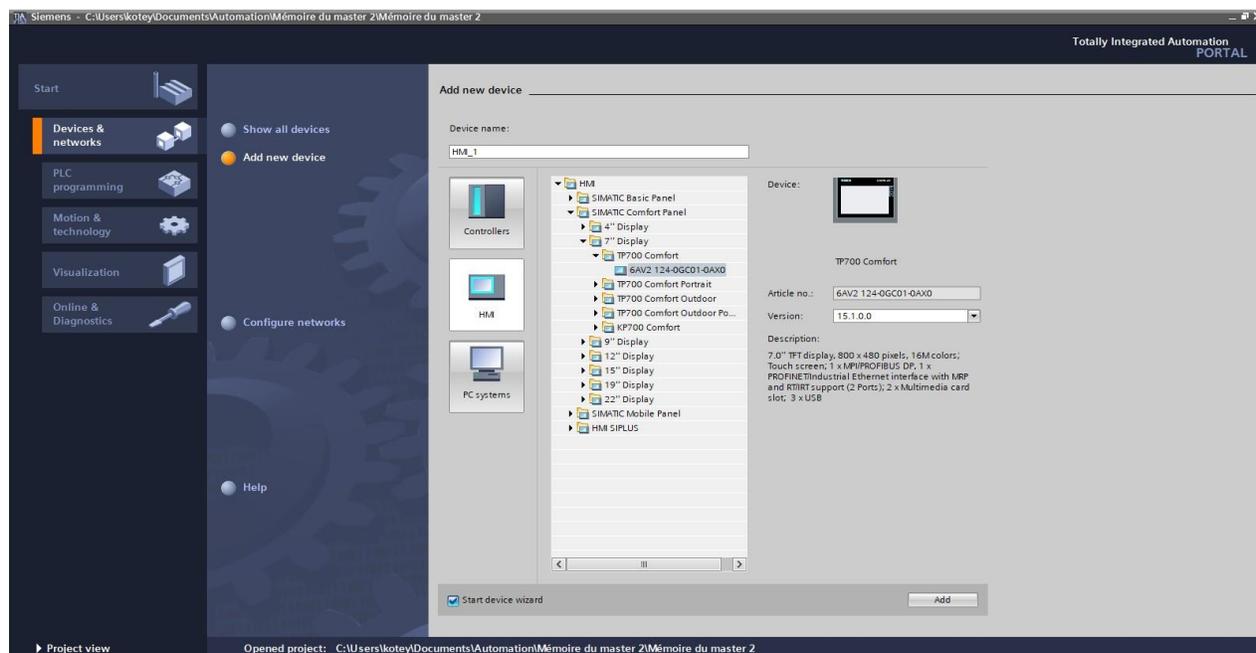


Figure III.9 : Ajout d'IHM.

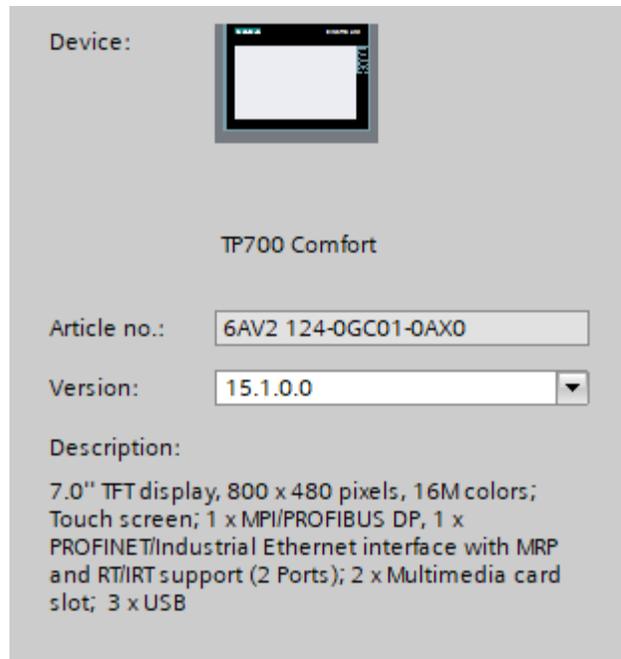


Figure III.10 : Description d'IHM ajouté.

III.4.1.1.3. Etablissement d'une liaison IHM/API

Il faut d'abord créer une liaison entre l'API et l'IHM, cela permet de lire les données qui se trouvent dans l'automate, le type de réseau PROFINET interface avec une adresse 192.168.0.1 pour l'API et une adresse 192.168.0.2 pour l'IHM.

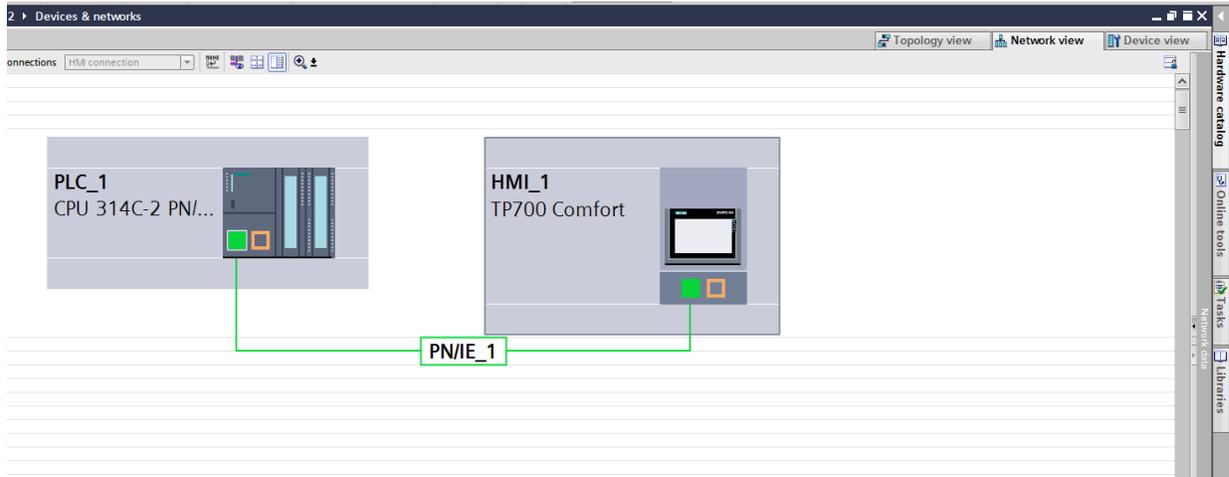


Figure III.11 : Liaison IHM/API.

III.4.1.1.4. L'adressage des entrées et sorties [16]

Les variables peuvent être de type entrée, sortie et memento :

* Entrées

Pour savoir l'état et le déroulement du processus l'automate récolte des informations sur l'état de l'installation et cela via des entrées de l'automate qui sont connectées aux différents capteurs et boutons de l'installation pour ensuite les traiter et générer la commande.

* Sorties

Après traitement des données provenant des entrées et pour commander l'installation, l'automate doit générer et envoyer des signaux via ses sorties. Les sorties automate sont connectées aux différents actionneurs de l'installation.

*** Mémento**

Zone de mémoire dans la mémoire système d'une CPU. Il est possible d'y accéder en écriture et en lecture (par bit, octet, mot et double mot). La zone des mémentos permet à l'utilisateur d'enregistrer des résultats intermédiaires.

*** Adressage**

Dans TIA portal, toutes les variables globales (entrées, sorties, mémentos) possèdent une adresse symbolique et une adresse absolue.

- **L'adresse absolue** représente l'identificateur d'opérande (I, Q, M) et son adresse et numéro de bit.

- **Adresse symbolique** correspond au nom que l'utilisateur a donné à la variable (ex : bouton marche). Le lien entre les adresses symboliques et absolues se fait dans la table des variables API.

Lors de la programmation, il est possible de choisir d'afficher les deux adresses absolues, symboliques ou encore les deux simultanément.

III.4.1.2. Programmation [16]

Une fois la configuration matérielle faite, nous arrivons sur la vue du projet. Elle comporte une arborescence avec les différents éléments du projet. Les éditeurs requis s'ouvrent en fonction des tâches à réaliser. Données, paramètres et éditeurs peuvent être visualisés dans une seule et même vue.

III.4.1.2.1. Définition des variables

Avant de commencer la programmation on déclare d'abord toutes les variables avec lesquelles on va contrôler le système.

C'est dans la table des variables API que l'on va pouvoir déclarer toutes les variables et les constantes utilisées dans le programme. Lorsque l'on définit une variable API, il faut définir :

- * Un nom : c'est l'adressage symbolique de la variable.
- * Le type de donnée : BOOL, IN.
- * L'adresse absolue : par exemple « Q1.5 ».

On peut également insérer un commentaire qui nous renseigne sur cette variable. Le commentaire peut être visible dans chaque réseau utilisant cette variable.

Name	Tag table	Data type	Address	Retain	Acces...	Visibl...	Comment
1	Bouton marche	Default tag table	Bool	%I0.0			
2	Bouton arrêt	Default tag table	Bool	%I0.1			NC
3	Commutateur automatique ...	Default tag table	Bool	%M3.6			
4	Arrêt de maintenance	Default tag table	Bool	%M3.7			
5	Start programme	Default tag table	Bool	%M0.0			
6	Chaleur augmentée	Default tag table	Bool	%Q0.0			
7	Mode manuel	Default tag table	Bool	%M0.1			
8	Mode automatique	Default tag table	Bool	%M0.2			
9	P edge manuel	Default tag table	Bool	%M0.3			
10	P edge automatique	Default tag table	Bool	%M0.4			
11	Start chauffage	Default tag table	Bool	%I0.2			
12	Stop chauffage	Default tag table	Bool	%I0.3			
13	SR chauffage	Default tag table	Bool	%M0.5			
14	Start climatiseur	Default tag table	Bool	%I0.4			
15	Stop climatiseur	Default tag table	Bool	%I0.5			
16	SR climatiseur	Default tag table	Bool	%M0.6			
17	Refrigidissement	Default tag table	Bool	%Q0.1			
18	Start ventilateurs	Default tag table	Bool	%I0.6			
19	Stop ventilateurs	Default tag table	Bool	%I0.7			
20	SR ventilation	Default tag table	Bool	%M0.7			
21	Ventilation	Default tag table	Bool	%Q0.2			
22	Start moteur sens droite	Default tag table	Bool	%I1.0			
23	Start moteur sens gauche	Default tag table	Bool	%I1.1			
24	Stop moteur de nettoyage	Default tag table	Bool	%I1.2			
25	Détecteur fin de cours: droite	Default tag table	Bool	%I1.3			NC
26	Détecteur fin de cours: gauche	Default tag table	Bool	%I1.4			NC
27	SR moteur sens droite	Default tag table	Bool	%M1.0			
28	SR moteur sens gauche	Default tag table	Bool	%M1.1			
29	Moteur sens droite	Default tag table	Bool	%Q0.3			
30	Moteur sens gauche	Default tag table	Bool	%Q0.4			
31	Input scale TT	Default tag table	Int	%W800			Transmetteur de température
32	Valeur négative	Default tag table	Bool	%I1.5			

Figure III.12 : Partie de la table de variables.

III.4.1.2.2. Les blocs de programmation [16]

L'automate met à disposition différents types des blocs qui contiennent le programme et les données correspondantes. Selon les exigences et la complexité du processus, il est possible de structurer le programme en différents blocs : OB, FB et FC.

- * **Blocs d'organisation (OB)** : Les blocs d'organisations gèrent le programme utilisateur.
- * **Blocs fonctionnels (FB)** : Les blocs fonctionnels sont des blocs de code qui sauvegardent en permanence leurs valeurs dans des blocs de données d'instance afin qu'il soit possible d'y accéder même après le traitement du bloc.
- * **Fonctions (FC)** : Les FC contiennent des routines de programmes pour les fonctions fréquemment utilisées. Les fonctions sont des blocs de code sans mémoire.
- * **Blocs de données (DB)** : Les blocs de données (DB) sont des zones de données dans le programme utilisateur qui contiennent des données utilisateur.
- * **Blocs fonctionnels système (SFB) et fonctions système (SFC)** : Les SFB et SFC sont intégrés à la CPU S7, ils permettent de réaliser quelques fonctions systèmes importantes.

Dans notre système, il y a ;

- * 69 variables divisées en quatre catégories 66 BOOL, 01 REEL, 01 INTEGER et 01 WORD
- * 05 blocs divisés en cinq ; 01 OB, 01 FC, 03 FB

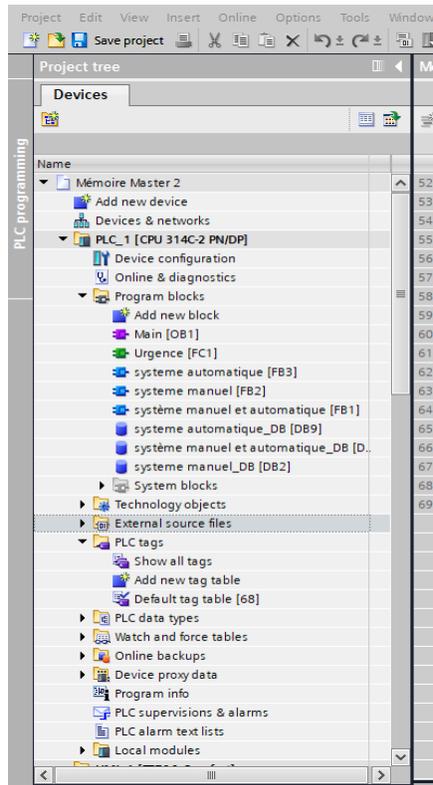


Figure III.13 : Les blocs de programmation dans notre système.

III.4.1.2.3. Le programme complet

Nous avons utilisé la langage FBD, et nous allons programmer l'automate avec les instructions disponibles dans l'onglet de sélection des tâches.

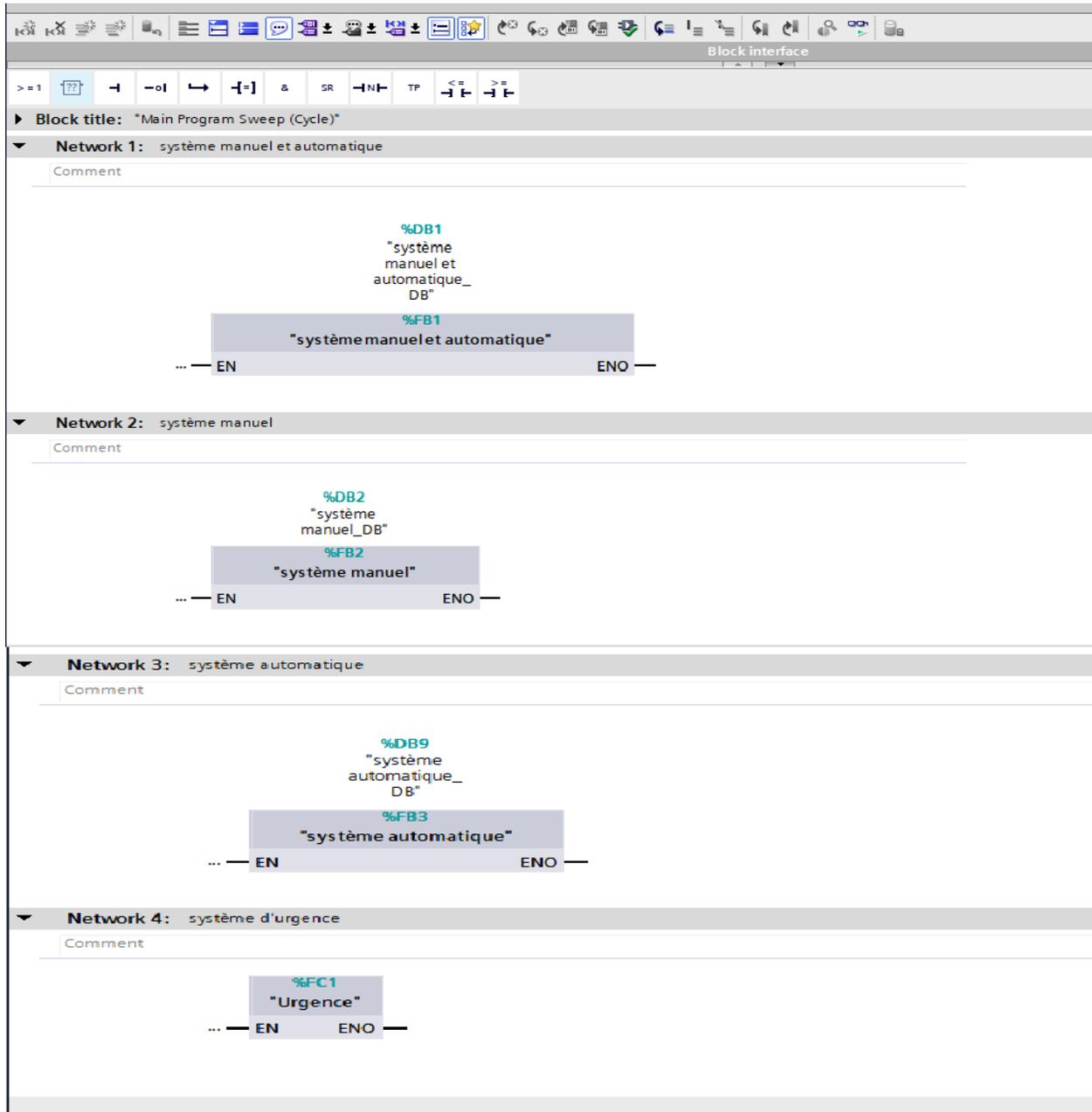


Figure III.14 : Programme principal de système (MAIN OB1).

Le bloc MAIN OB1 (**Figure III.14**) contient le programme principal et tous les autres blocs.

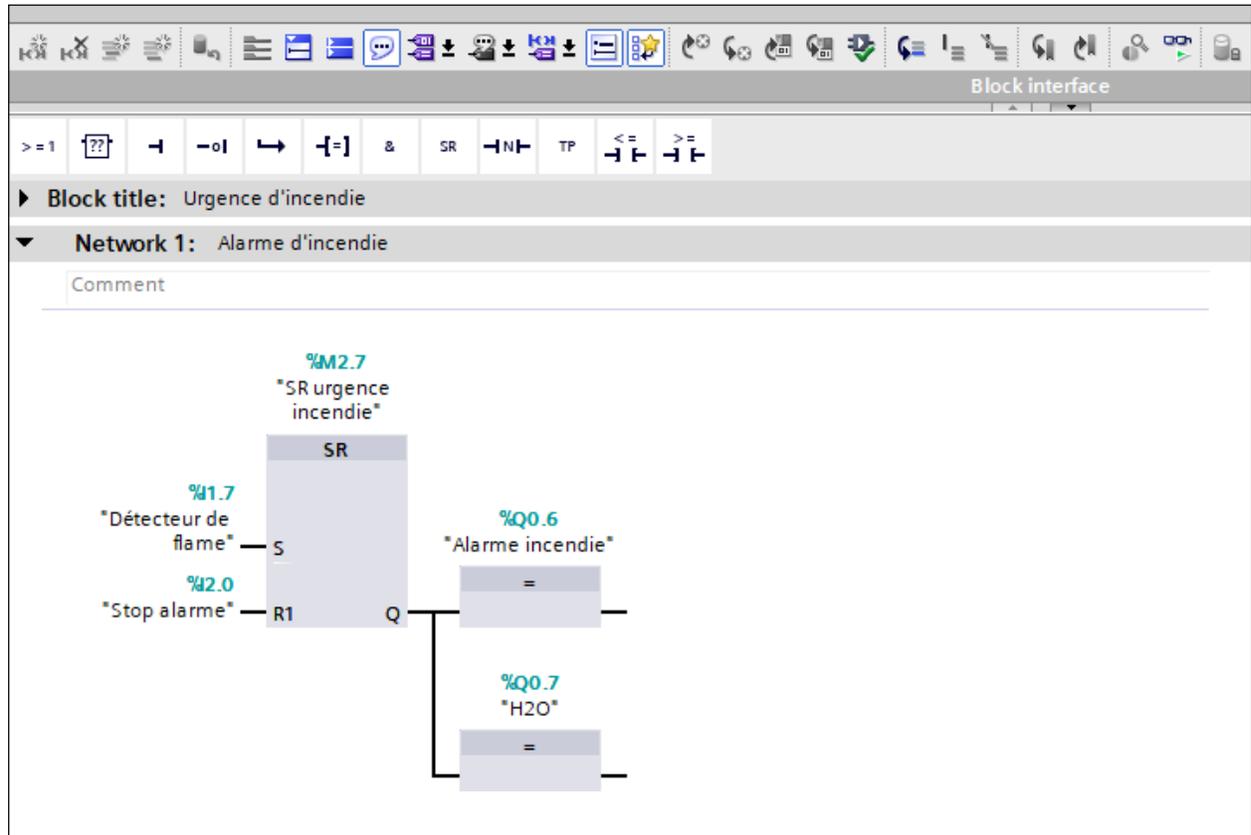


Figure III.15 : Programme d'alarme d'incendie (FC1).



Figure III.16 : Programme de commutation (manuel ou automatique) (FB).

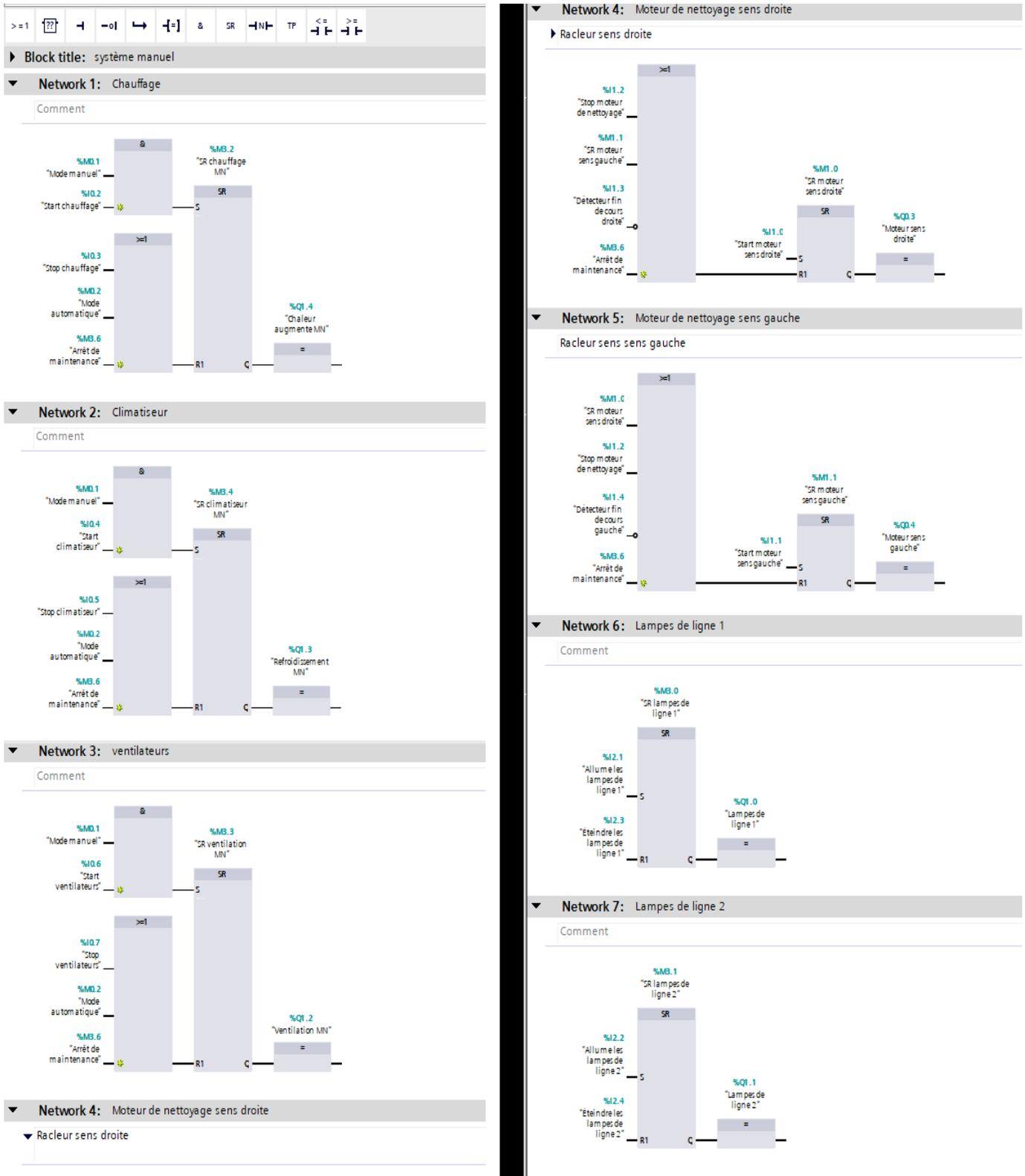


Figure III.17 : Programme en mode manuel (FB2).

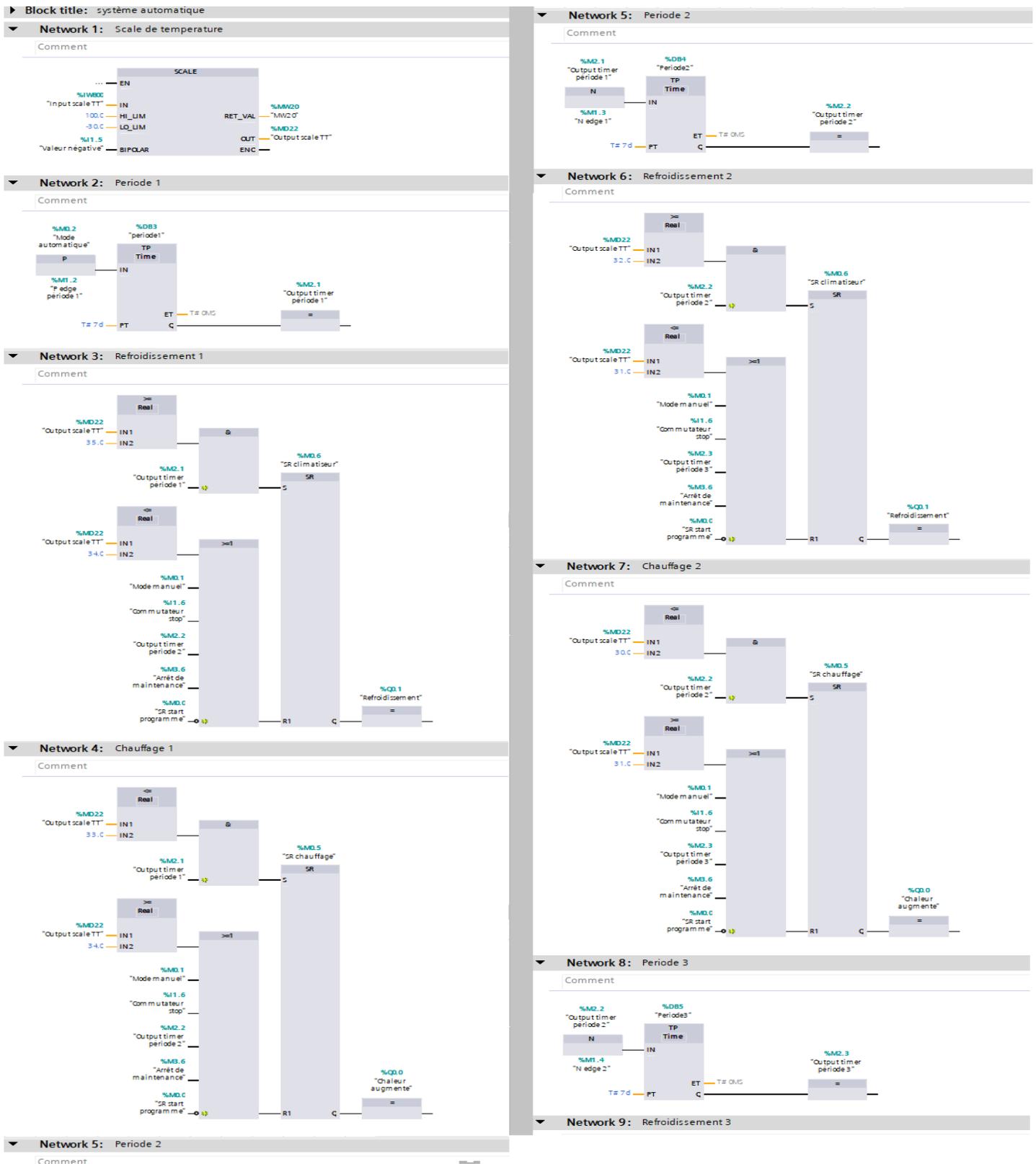


Figure III.18 : Programme en mode automatique (FB3) -1-

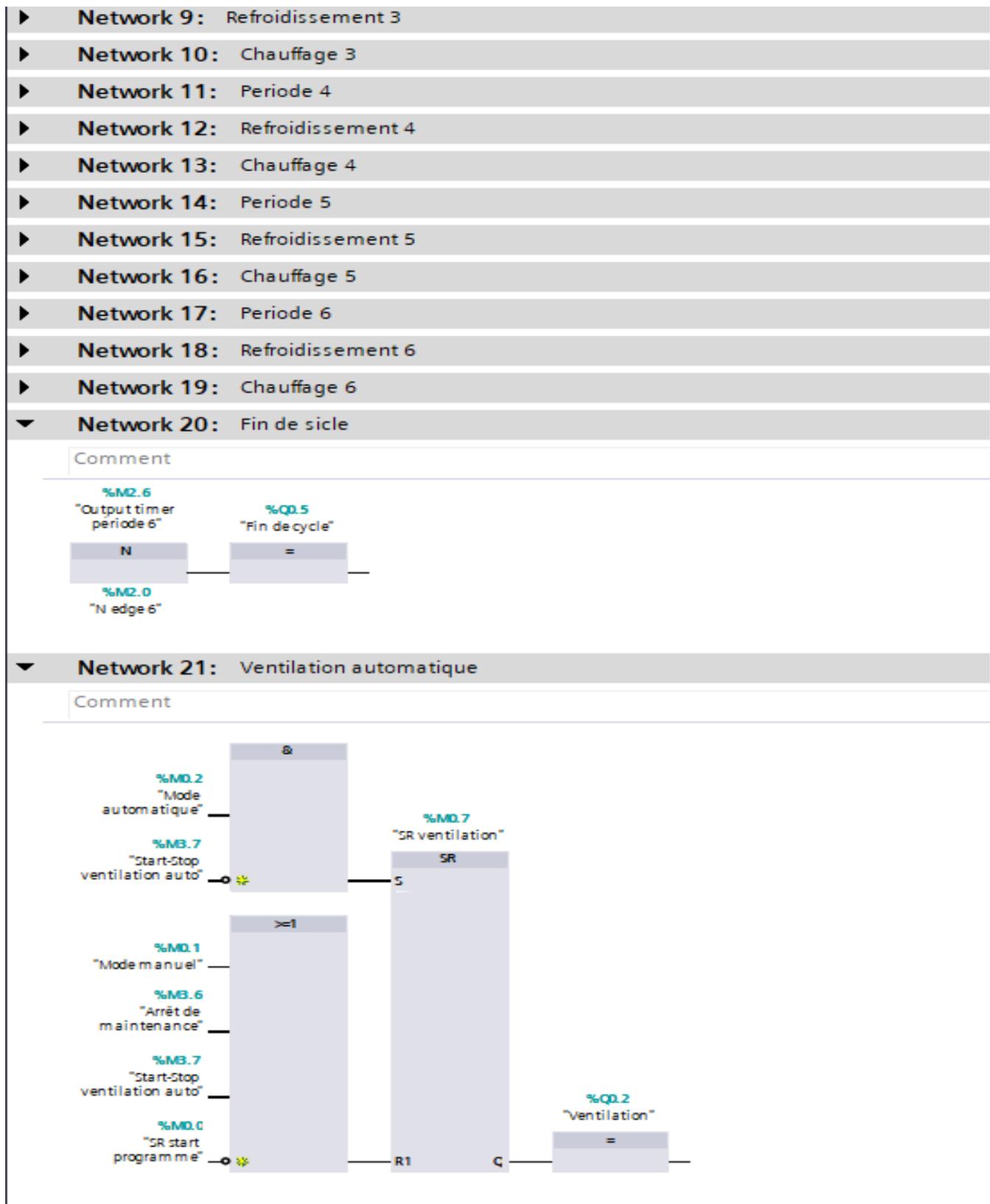


Figure III.19 : Programme en mode automatique (FB3) -2-.

III.4.1.3. Configuration IHM [14]

Pour bien contrôler le processus, l'opérateur a besoin d'avoir le maximum de transparence, ce qui lui permet de bien superviser et contrôler l'installation, ceci est possible avec l'IHM. Le contrôle du processus est assuré par le système d'automatisation.

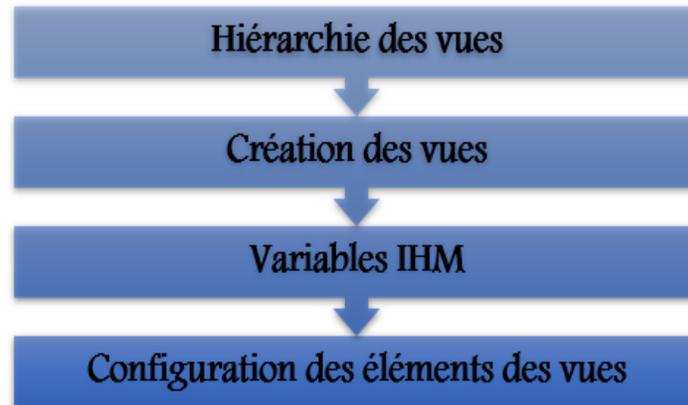


Figure III.20 : Configuration IHM.

III.4.1.3.1. Hiérarchie des vues

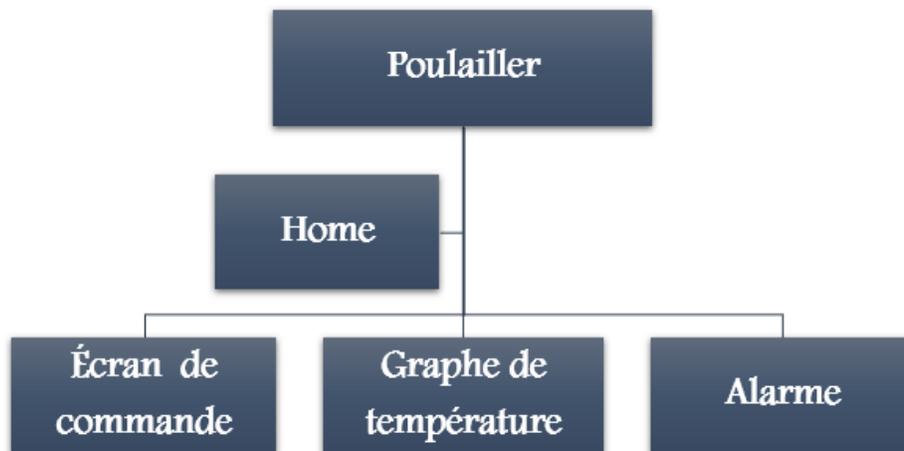


Figure III.21 : Hiérarchie des vues.

III.4.1.3.2. Création des vues

Après avoir choisi l'appareil, nous pouvons maintenant créer les vues synoptiques à partir de la grande bibliothèque de WINCC RT de SIEMENS. L'image suivante représente la vue initiale (Poulailler) de l'IHM qui affiche le système en général avec les états des machines (chauffage, climatiseur, ...).

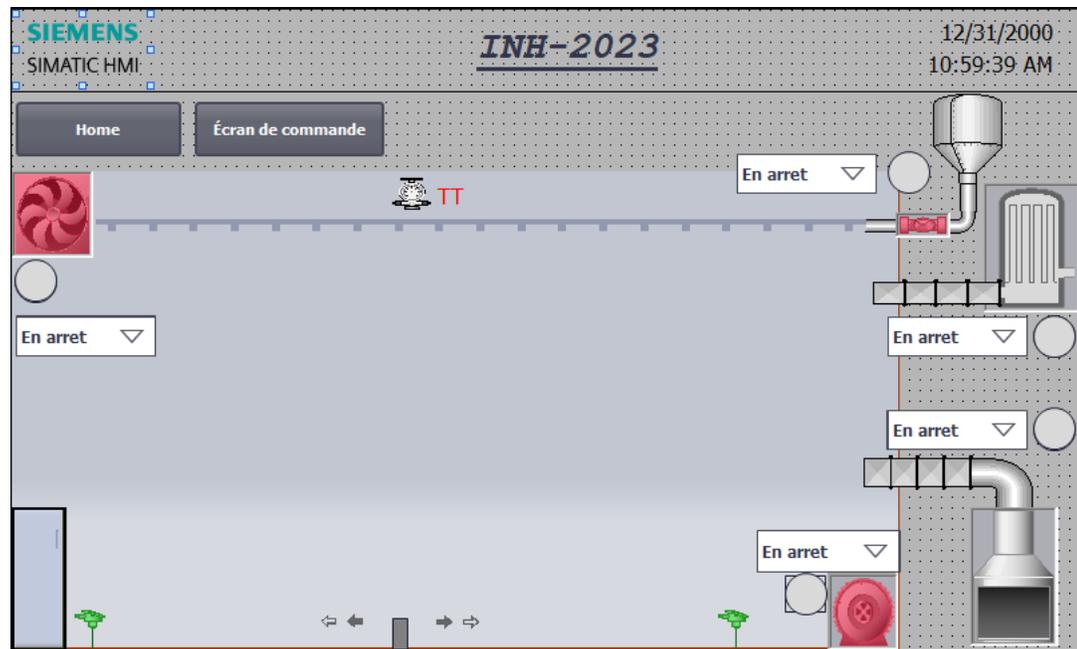


Figure III.22 : Vue initiale (Poulailler).

Les autres vues :

- * Home
- * Écran de commande
- * Graphe de température
- * Alarme

Elles seront présentées dans le prochain chapitre !

✚ Tableau des différents éléments des vues :

Composants	Non actif	Actif
Chauffages		
Climatiseurs		
Ventilateurs		
Moteur double sens		
Vanne du H2O		

Tableau III.1: Représentation des éléments des vues.

III.4.1.3.3. Variables IHM

On distingue deux types de variables, les variables externes et les variables internes :

- * Les variables externes permettent de communiquer et d'échanger des données entre les composants d'un processus automatisé, entre un pupitre opérateur et un automate.
- * Les variables internes ne possèdent aucun lien avec l'automate, elles sont enregistrées dans la mémoire du pupitre.

Name	Tag table	Data type	Connection	PLC name	PLC tag	Address	Ac...
Bouton arrêt	Default tag table	Bool	HMI_Connectio...	PLC_1	"Bouton arrêt"	%I0.1	<B... ^
Start chauffage	Default tag table	Bool	HMI_Connectio...	PLC_1	"Start chauffage"	%I0.2	<B... ^
Stop chauffage	Default tag table	Bool	HMI_Connectio...	PLC_1	"Stop chauffage"	%I0.3	<B... ^
Start moteur sens droite	Default tag table	Bool	HMI_Connectio...	PLC_1	"Start moteur sens droite"	%I1.0	<B... ^
Stop moteur de nettoyage	Default tag table	Bool	HMI_Connectio...	PLC_1	"Stop moteur de nettoya..."	%I1.2	<B... ^
Input scale TT	Default tag table	Int	HMI_Connectio...	PLC_1	"Input scale TT"	%IW800	<B... ^
Start programme	Default tag table	Bool	HMI_Connectio...	PLC_1	"Start programme"	%M0.0	<B... ^
Mode manuel	Default tag table	Bool	HMI_Connectio...	PLC_1	"Mode manuel"	%M0.1	<B... ^
Mode automatique	Default tag table	Bool	HMI_Connectio...	PLC_1	"Mode automatique"	%M0.2	<B... ^
SR chauffage	Default tag table	Bool	HMI_Connectio...	PLC_1	"SR chauffage"	%M0.5	<B... ^
SR climatiseur	Default tag table	Bool	HMI_Connectio...	PLC_1	"SR climatiseur"	%M0.6	<B... ^
SR ventilation	Default tag table	Bool	HMI_Connectio...	PLC_1	"SR ventilation"	%M0.7	<B... ^
SR moteur sens droite	Default tag table	Bool	HMI_Connectio...	PLC_1	"SR moteur sens droite"	%M1.0	<B... ^
SR moteur sens gauche	Default tag table	Bool	HMI_Connectio...	PLC_1	"SR moteur sens gauche"	%M1.1	<B... ^
SR urgence incendie	Default tag table	Bool	HMI_Connectio...	PLC_1	"SR urgence incendie"	%M2.7	<B... ^
Alarme incendie	Default tag table	Bool	HMI_Connectio...	PLC_1	"SR urgence incendie"	%M2.7	<B... ^
SR lampes ligne 1	Default tag table	Bool	HMI_Connectio...	PLC_1	"SR lampes de ligne 1"	%M3.1	<B... ^
SR lampes ligne 2	Default tag table	Bool	HMI_Connectio...	PLC_1	"SR lampes de ligne 2"	%M3.2	<B... ^
SR chauffage MN	Default tag table	Bool	HMI_Connectio...	PLC_1	"SR chauffage MN"	%M3.3	<B... ^

Figure III.23 : Partie de la table de variables IHM.

III.4.1.3.4. Configuration des éléments des vues

Nous avons configuré les éléments de vues (les boutons de commandes, les textes, les graphiques, la courbe de température, les couleurs, ...) soit avec ;

- * Animation
- * Evènement
- * Visibilité

Ou par certains d'entre eux.

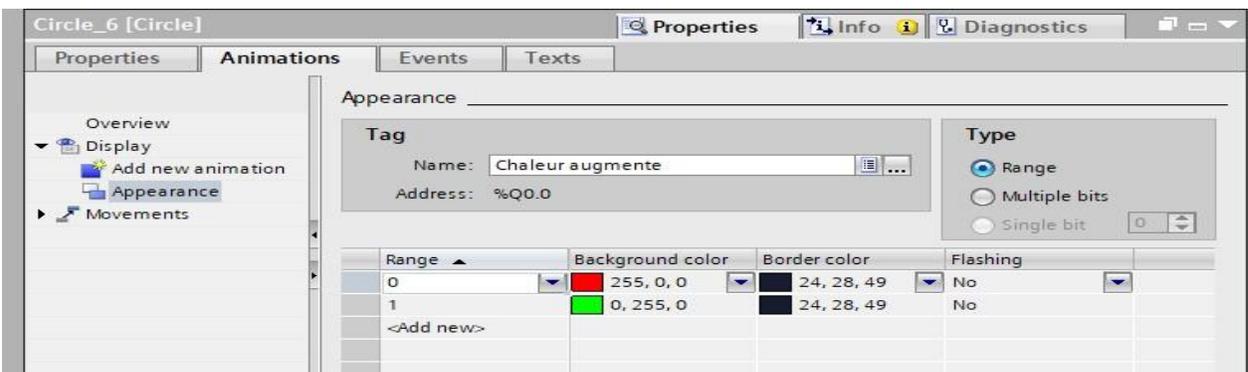


Figure III.24 : Exemple d'animation.

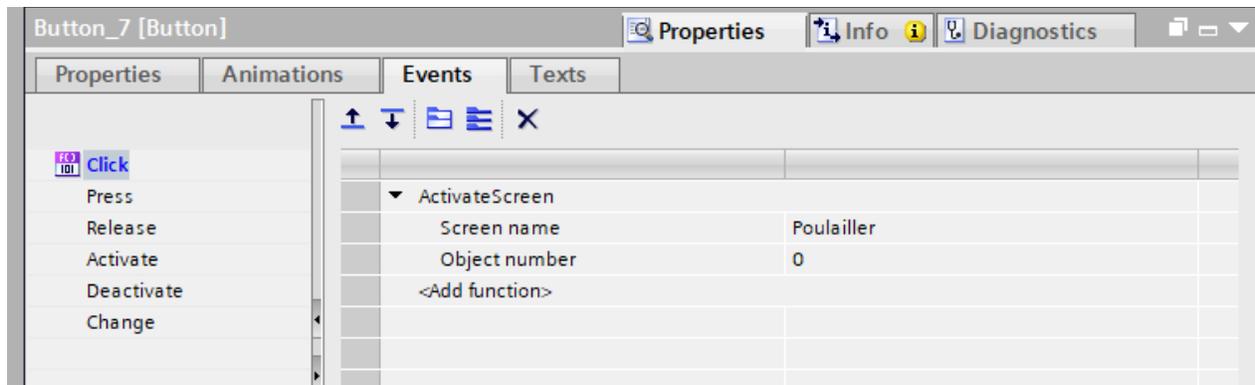


Figure III.25 : Exemple d'évènement d'un bouton.

III.5. Conclusion

Dans ce chapitre, nous avons exposé la programmation sur le logiciel TIA PORTAL V15.1. Après, nous avons défini et configuré les appareils choisis selon nos besoins. Puis, on a démarré la réalisation de notre projet en utilisant des blocs d'organisation avec le langage FBD, aussi nous avons configuré dans ce chapitre l'appareil d'interface IHM.

CHAPITRE IV

« Compilation et simulation du programme »

IV.1. Introduction

Dans ce chapitre nous allons simuler notre processus dans différentes situations et nous suivrons son fonctionnement et ses réactions dans toutes les situations à l'aide de S7-PLCSIM³ qui nous permet de simuler et de tester notre programme.

La programmation et la simulation ont été complètement réalisées avec logiciel TIA PORTAL V15.1.

IV.2 Simulation de l'API et l'IHM

Une fois le programme réalisé dans TIA PORTAL, nous le chargeons dans l'automate simulé, et procédons aux tests grâce à son extension PLCSIM.

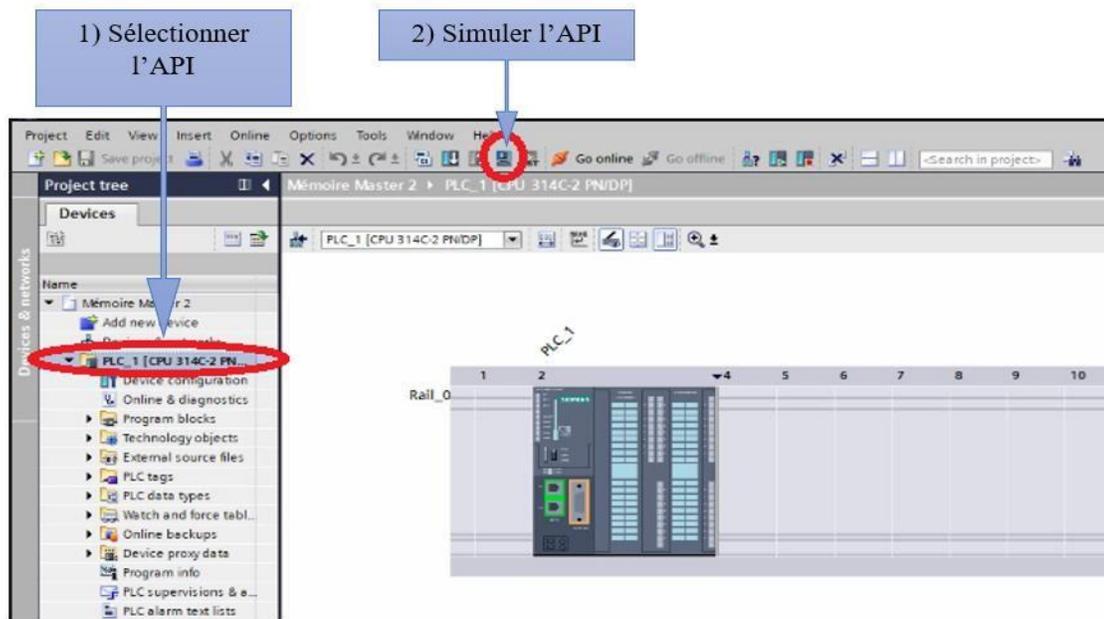


Figure IV.1 : Simulation de l'API.

³ Application de simulation (dans notre cas, nous utilisons la version 5.4)

La même pour l'IHM.

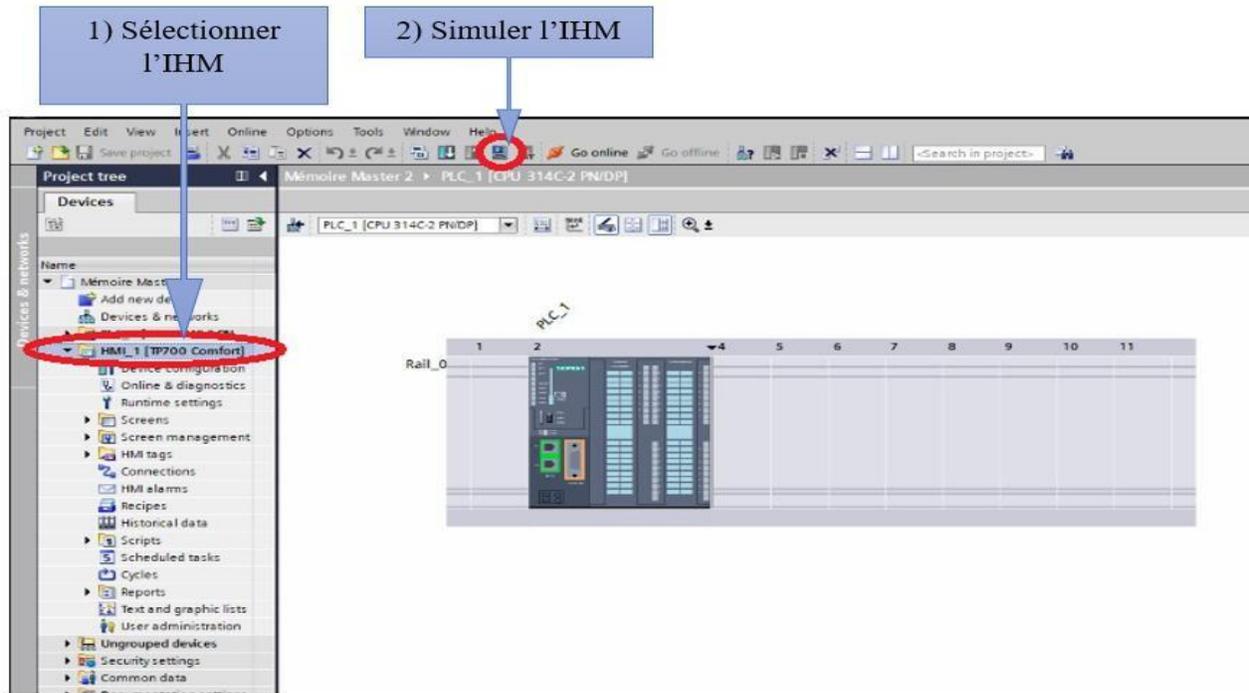


Figure IV.2 : Configuration de l'API et simulation de l'IHM.

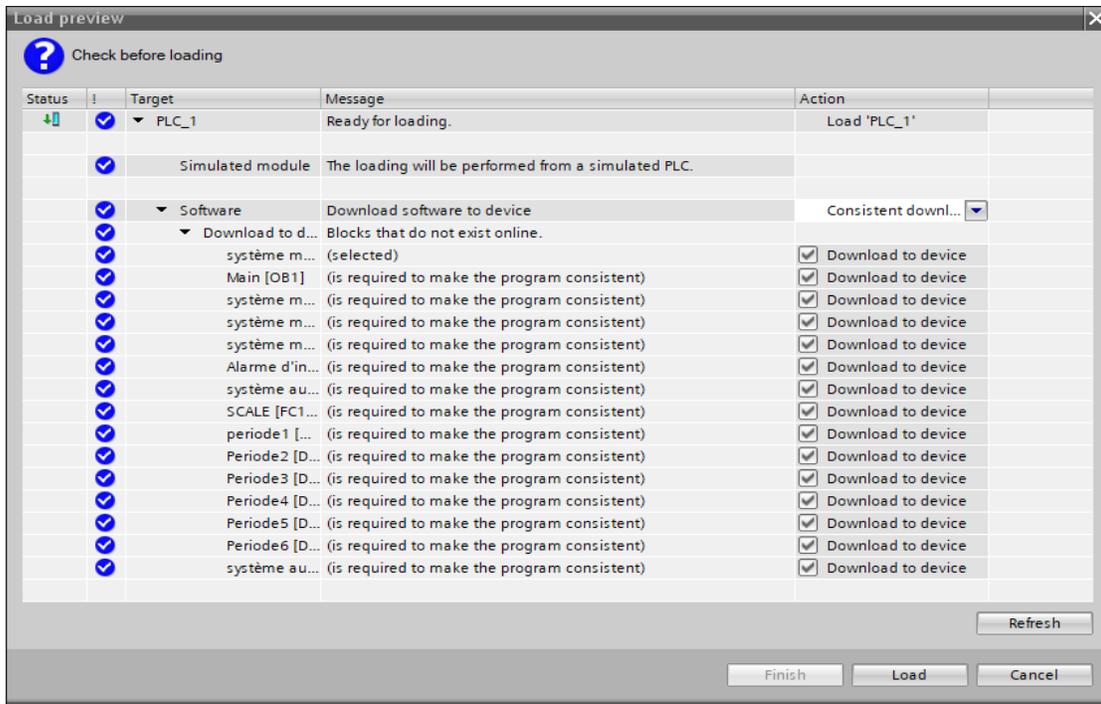


Figure IV.3 : Compilation du programme.

Après la compilation du programme l'interface de simulation PLCSIM est ouverte directement.

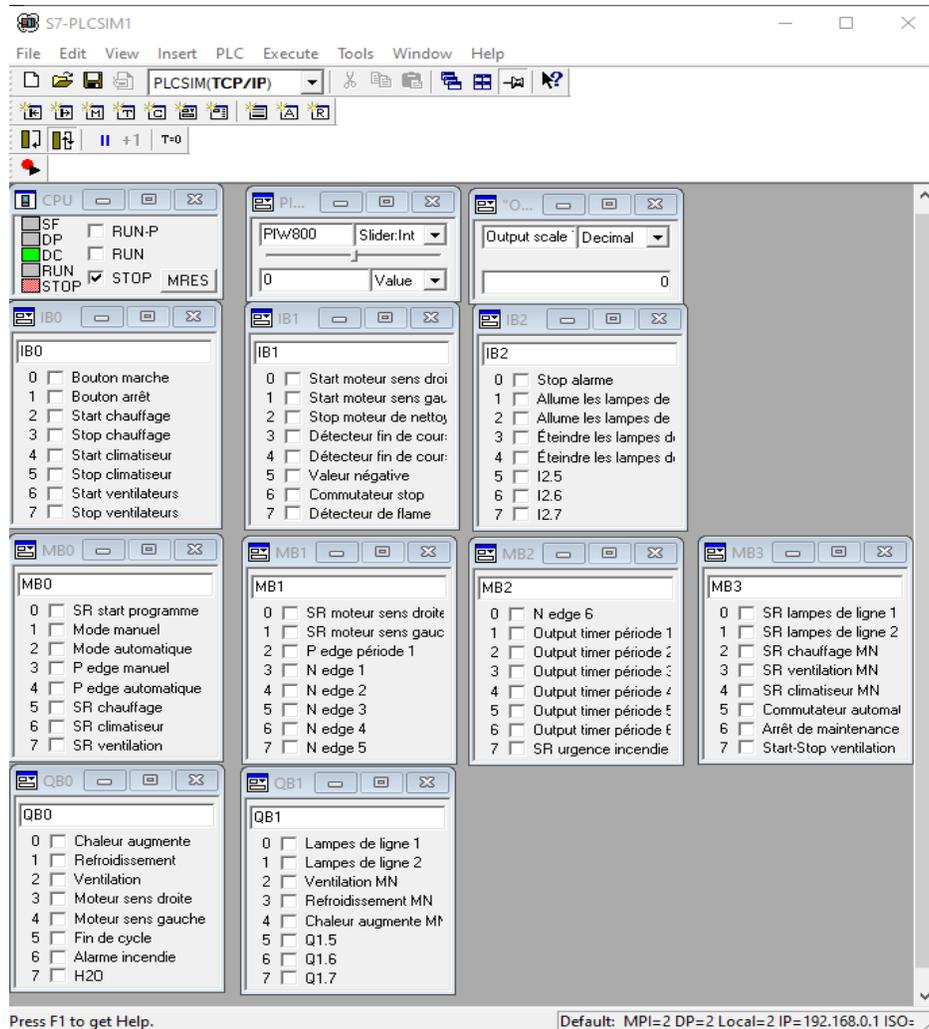


Figure IV.4 : Interface de simulation PLCSIM.

Une fois le programme et l'IHM chargés et la mise en ligne des appareils effectuée, des voyants verts indiquent que tout est prêt et fonctionnel sans erreurs et toute la procédure de chargement est correcte.

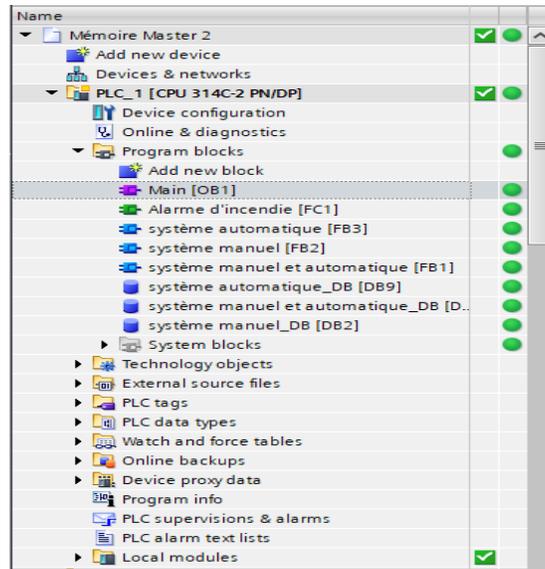


Figure IV.5 : Indication de la mise en ligne du programme et de l'IHM.

On lance ensuite la simulation sur PLCSIM, IHM et on commence à modifier l'état des entrées pour simuler différentes situations.

IV.3. Teste et résultats

IV.3.1. En mode manuel

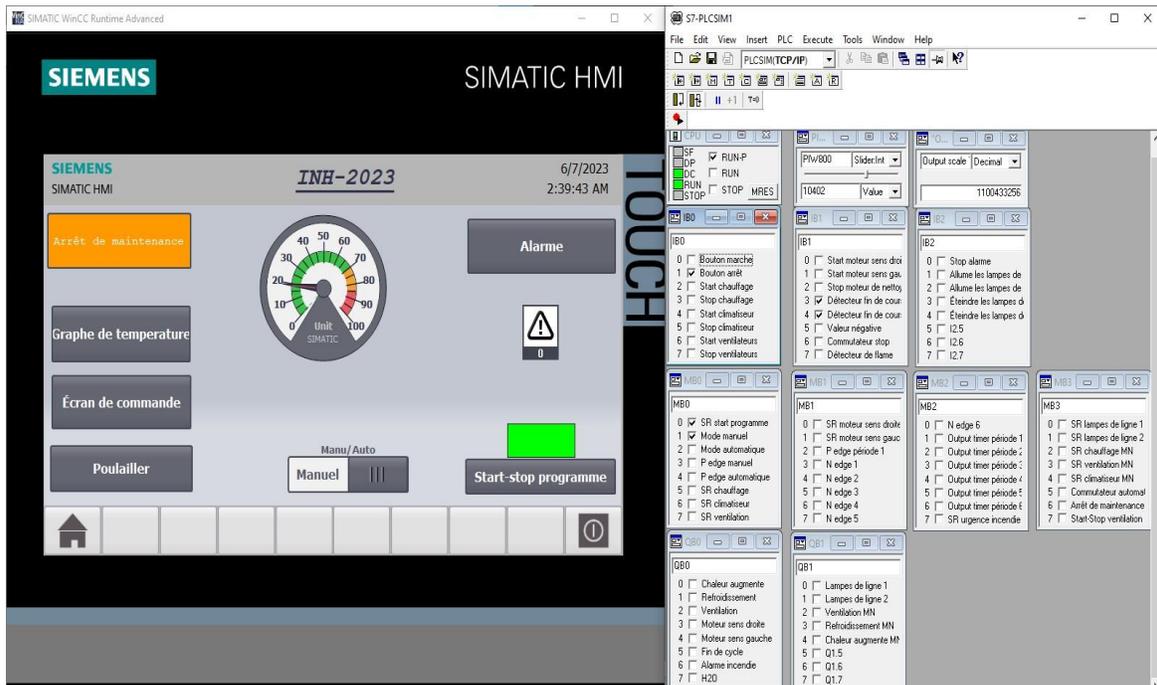


Figure IV.6 : Démarrage en mode manuel.

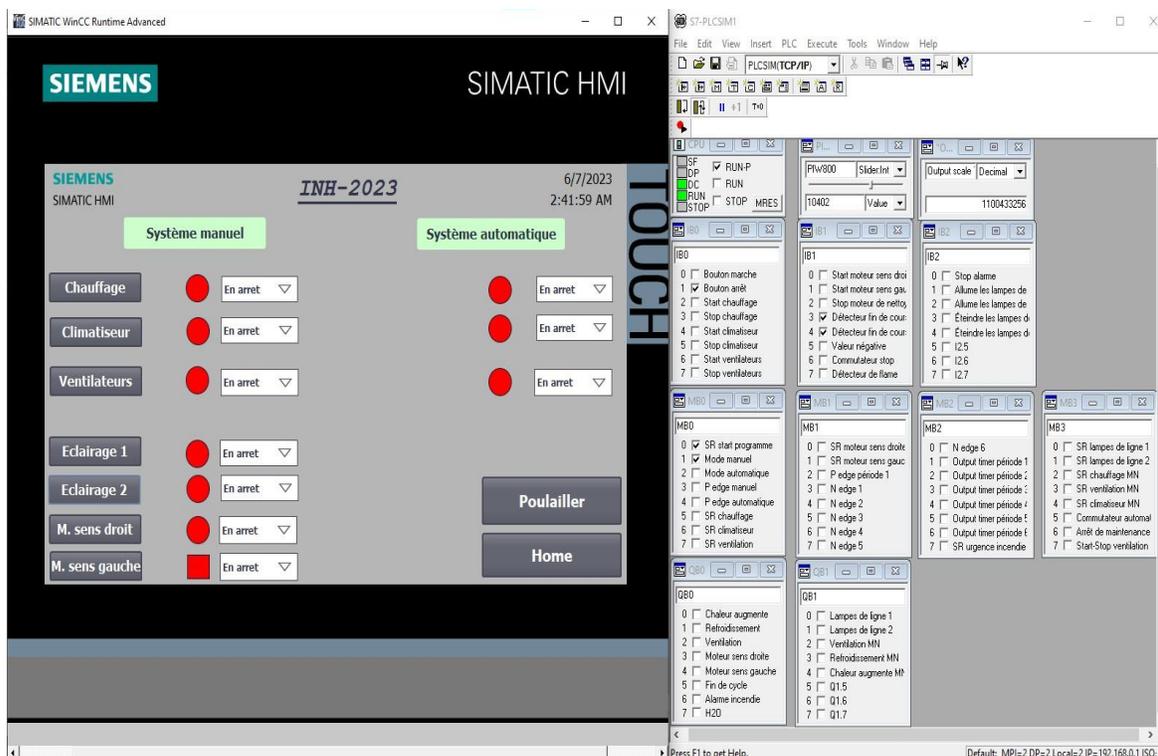


Figure IV.7 : Écran « écran de commande » à l'arrêt.

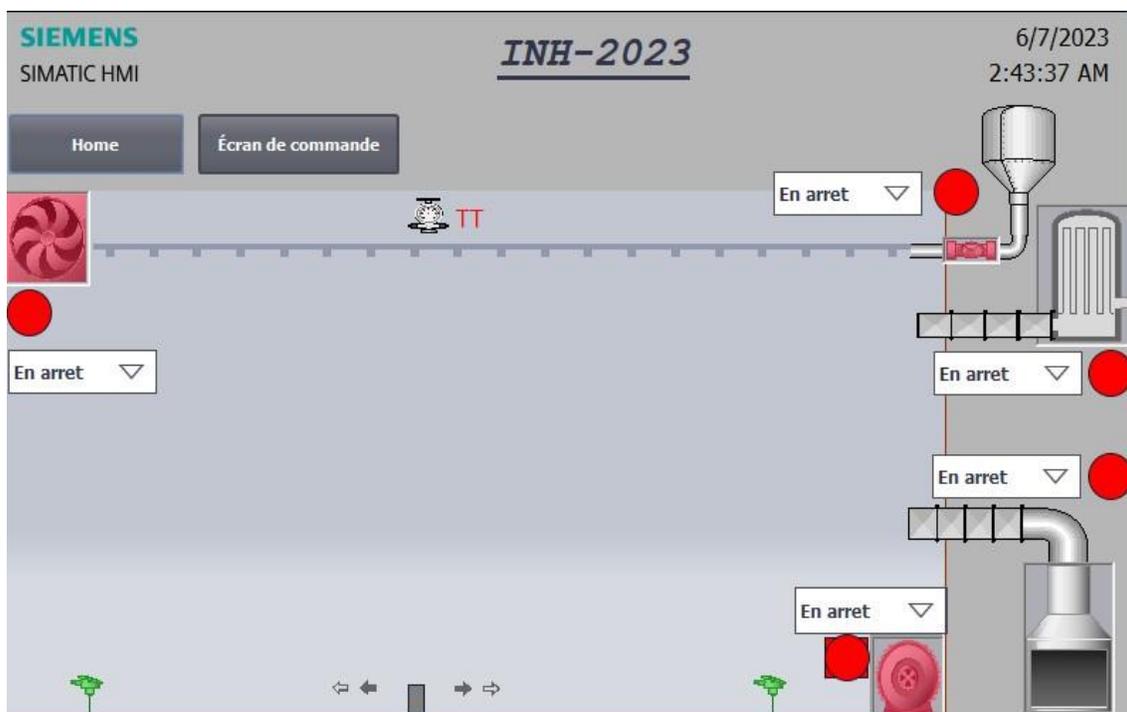


Figure IV.8 : Écran de « Poulailler » à l'arrêt.

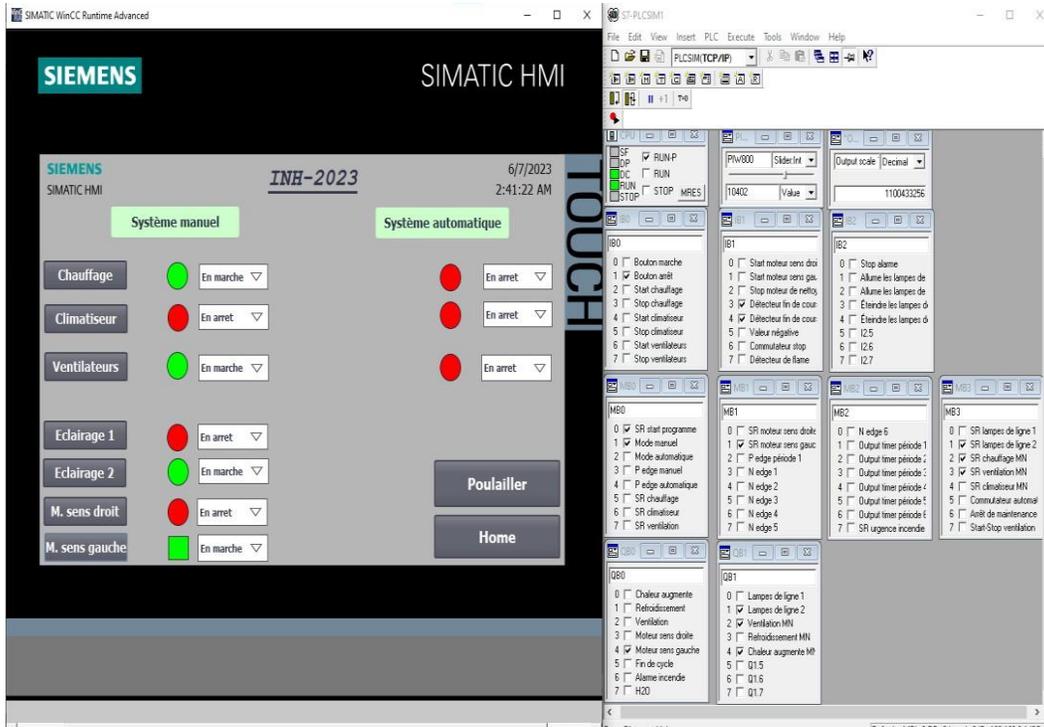


Figure IV.9 : Écran « écran de commande » en fonctionnement.

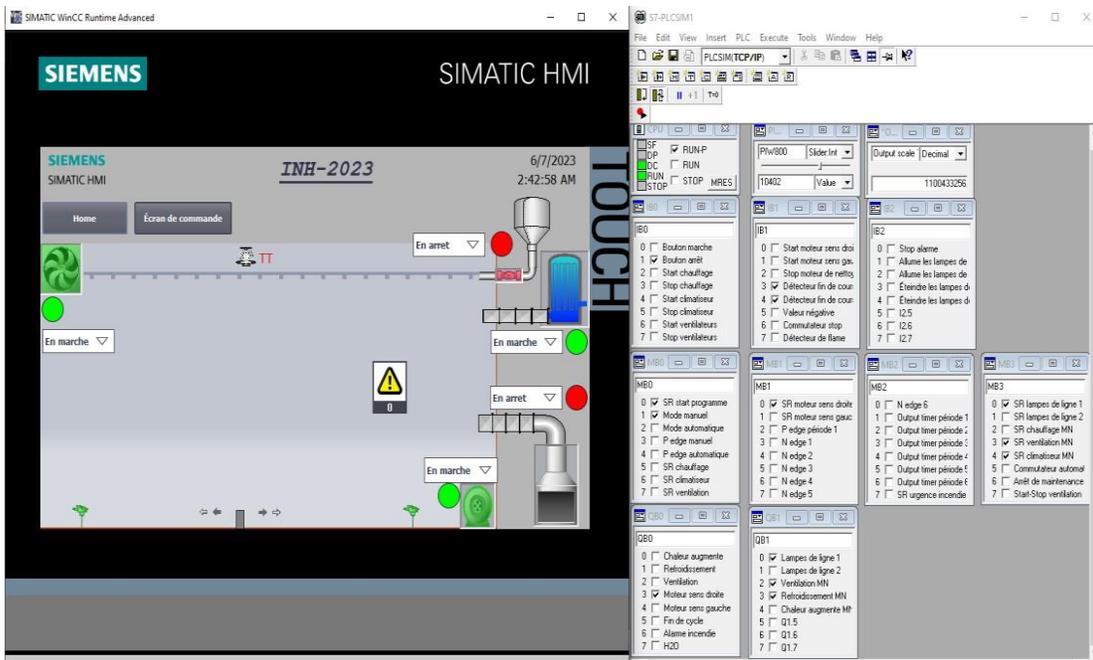


Figure IV.10 : Écran de « Poulailler » en fonctionnement.

IV32 En mode automatique

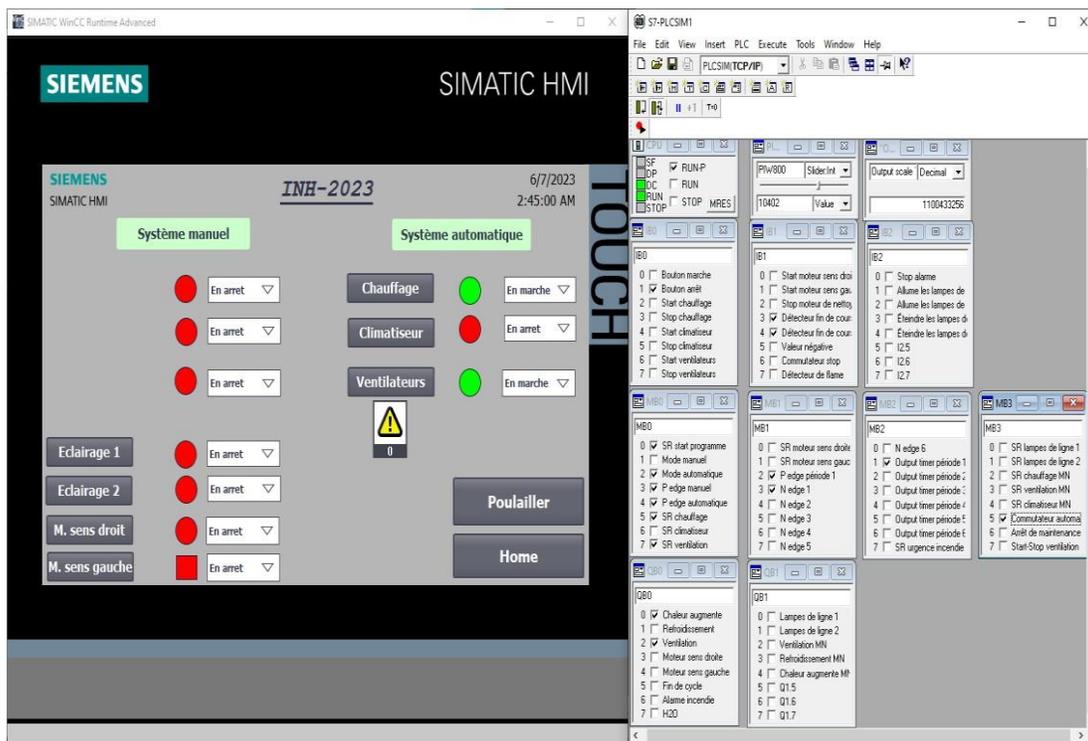


Figure IV.11 : Commutation en mode automatique.

Alarme :

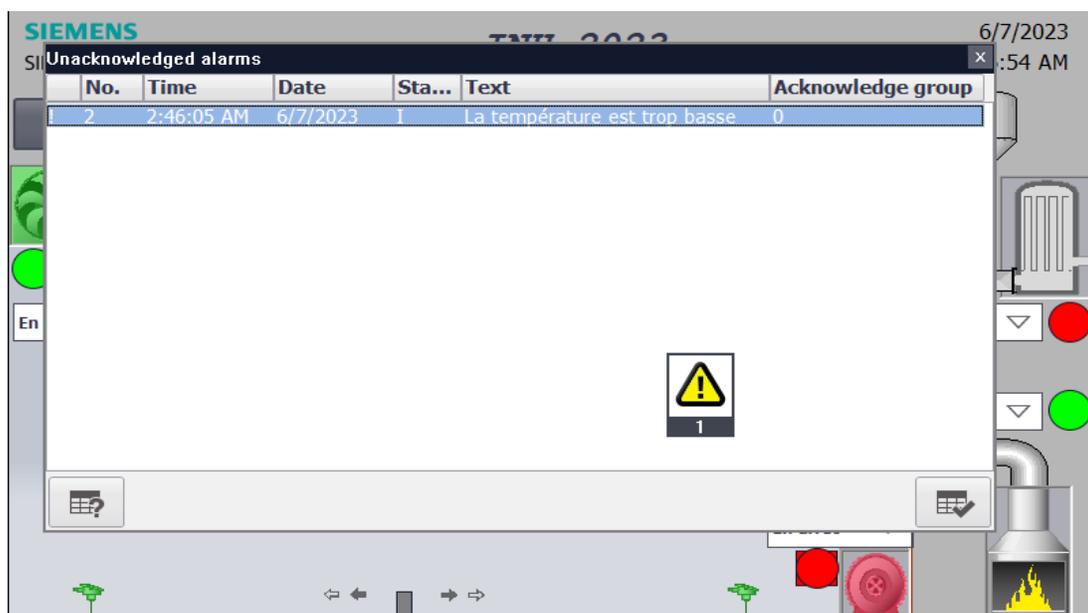
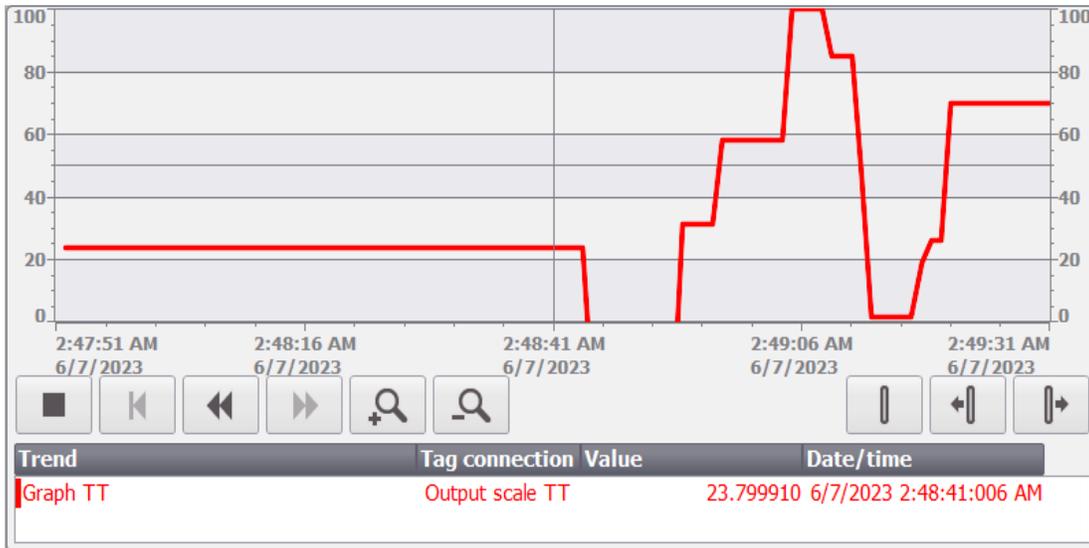


Figure IV.12 : Indication d'alarme.

+ Graphe de température :**Figure IV.13** : Graphe de température.

IV.4 Conclusion

Ce dernier chapitre nous a permis de tester notre système en utilisant le programme de l'API et de l'IHM dans différentes situations afin de s'assurer de son bon fonctionnement et prévoir des modifications ou des ajouts à effectuer plus tard. Et enfin, nous remarquons que la réponse du programme est compatible avec le travail proposé.

Conclusion générale

Conclusion générale

Ce travail de fin d'étude s'inscrit dans le cadre de l'automatisation et supervision d'un bâtiment d'élevage de volaille (Poulailler dans notre cas). Afin d'atteindre cet objectif, nous avons commencé par prendre connaissance sur les poulets de chair et les meilleures conditions de leur vie (température, humidité, qualité d'air, l'éclairage, ventilation...). Nous avons aussi étudié l'élevage du poulet, ainsi que les bâtiments d'élevage avicole et leur caractéristiques (surface, largeur, longueur, implantation...).

Nous avons passé en revue les automates programmables industriels, leurs caractéristiques, critères de choix, avantages et inconvénients, principe de fonctionnement, ainsi que les langages de programmation utilisables et tous les éléments qui leurs sont liées.

La prise de connaissance du SIMATIC STEP7 intégré à TIA PORTAL, nous a permis de programmer le fonctionnement du poulailler et d'en récupérer les états des variables qui nous intéressent pour créer notre IHM. Pour la conception de l'IHM en vue de la supervision du système, nous avons exploité les performances de SIMATIC WinCC (TIA PORTAL), qui est un logiciel permettant de gérer les interfaces graphiques avec des visualisations et des animations actualisées.

Enfin, nous espérons que notre travail servira comme base de départ pour notre vie professionnelle, et être bénéfique, utile aux promotions et générations futures.

Références bibliographiques

- [1] S. BAALI & M. MAHMOUDI, « Régulateur de température et d'humidité d'un poulailler ». Mémoire fin d'étude, 2021-2022.
- [2] S. KERMIA & A. OUACHEM, « Elevage de poulet de chair dans la région de Bouira (Enquête et suivi) ». Mémoire de master, 2020.
- [3] R. BELALA & A. TALAH, « Suivi d'élevage de poulet de chair au niveau de la Daïra de Khemis Miliana ». Mémoire fin d'étude, université Saad Dahleb-Blida, 2019.
- [4] Z. DJEROU « Influence des conditions d'élevage sur les performances chez le poulet de chair ». Mémoire fin d'étude (de Magister en médecine vétérinaire), université Mentouri de Constantine, 2006.
- [5] S. BOUCHERB, « Enquête sur des bâtiments d'élevage de poulet de chair durant la période de démarrage dans la région de Biskra ». Mémoire de master, 2019.
- [6] Arbor Acres, « Guide d'élevage de poulet de chair ». Aviagen, 2018.
- [7] A. SALLEMI, « Automates Programmables industriels ». Université Virtuelle de Tunis (UVT).
URL : https://www.uvt.rnu.tn/resources-uvt/cours/Automates/chap2/co/Module_chap2.html.
- [8] Documentation SIEMENS « SIMATIC S7-300 Système d'automatisation S7-300 Caractéristiques des modules ». Version : 06/2017.
- [9] P. RAYMOND, « Mécanique et automatismes industriels ». Cours, 2005.
- [10] Site web ; URL :
www.ip-systemes.com/details-qu+est+ce+qu+un+automate+programmable+industriel+-772.html#
- [11] W. BOLTON, « Automates Programmables industriels ». DUNOD, Traduction de Herve Soulard 2^e édition 2015.

Références bibliographiques

- [12] A. GONZAGA (Enseignant Automatismes, Toulouse, Occitanie, France), « Les Automates Programmables Industriels ». PDF.
- [13] Mme EL HAMMOUMI, « Cours Automatismes Logiques & Industriels –GE1 ». Université MOHAMED BEN ABDELLAH de FES.
- [14] S. MELLALI & L. YOUSFI, « Etude de l'automatisation et de la supervision d'un procédé de lavage de filtres Niagara à CEVITAL ». Mémoire de master, université de Bejaia, 2017.
- [15] Siemens AG, « SIMATIC STEP7 dans le portail Totally Integrated Automation Portal ». Livre, 2013.
- [16] B. BEKKARI & O. BARKA, « Automatisation d'une station de pompage à l'aide d'un API S7-1200 ». Mémoire fin d'étude, 2017-2018.

Annexe A : Table de variables de notre système.

PLC tags								
	Name ▲	Tag table	Data type	Address	Retain	Acces...	Visibl...	Comment
1	Alarme incendie	Default tag table	Bool	%Q0.6		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	sonore + lumière
2	Allume les lampes de ligne 1	Default tag table	Bool	%I2.1		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
3	Allume les lampes de ligne 2	Default tag table	Bool	%I2.2		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
4	Arrêt de maintenance	Default tag table	Bool	%M3.6		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
5	Bouton arrêt	Default tag table	Bool	%I0.1		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	NC
6	Bouton marche	Default tag table	Bool	%I0.0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
7	Chaleur augmente	Default tag table	Bool	%Q0.0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
8	Chaleur augmente MN	Default tag table	Bool	%Q1.4		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
9	Commutateur automatique - ...	Default tag table	Bool	%M3.5		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
10	Commutateur stop	Default tag table	Bool	%I1.6		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
11	Détecteur de flame	Default tag table	Bool	%I1.7		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
12	Détecteur fin de cours droite	Default tag table	Bool	%I1.3		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	NC
13	Détecteur fin de cours gauche	Default tag table	Bool	%I1.4		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	NC
14	Éteindre les lampes de ligne 1	Default tag table	Bool	%I2.3		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
15	Éteindre les lampes de ligne 2	Default tag table	Bool	%I2.4		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
16	Fin de cycle	Default tag table	Bool	%Q0.5		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
17	H2O	Default tag table	Bool	%Q0.7		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
18	Input scale TT	Default tag table	Int	%IW800		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Transmeteur de température
19	Lampes de ligne 1	Default tag table	Bool	%Q1.0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
20	Lampes de ligne 2	Default tag table	Bool	%Q1.1		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
21	Mode automatique	Default tag table	Bool	%M0.2		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
22	Mode manuel	Default tag table	Bool	%M0.1		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
23	Moteur sens droite	Default tag table	Bool	%Q0.3		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
24	Moteur sens gauche	Default tag table	Bool	%Q0.4		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
25	MW20	Default tag table	Word	%MW20		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
26	N edge 1	Default tag table	Bool	%M1.3		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
27	N edge 2	Default tag table	Bool	%M1.4		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
28	N edge 3	Default tag table	Bool	%M1.5		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
29	N edge 4	Default tag table	Bool	%M1.6		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
30	N edge 5	Default tag table	Bool	%M1.7		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
31	N edge 6	Default tag table	Bool	%M2.0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
32	Output scale TT	Default tag table	Real	%MD22		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
33	Output timer période 1	Default tag table	Bool	%M2.1		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
34	Output timer période 2	Default tag table	Bool	%M2.2		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
35	Output timer période 3	Default tag table	Bool	%M2.3		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	

Annexes

35		Output timer période 3	Default tag table	Bool	%M2.3		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
36		Output timer période 4	Default tag table	Bool	%M2.4		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
37		Output timer période 5	Default tag table	Bool	%M2.5		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
38		Output timer période 6	Default tag table	Bool	%M2.6		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
39		P edge automatique	Default tag table	Bool	%M0.4		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
40		P edge manuel	Default tag table	Bool	%M0.3		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
41		P edge période 1	Default tag table	Bool	%M1.2		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
42		Refroidissement	Default tag table	Bool	%Q0.1		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
43		Refroidissement MN	Default tag table	Bool	%Q1.3		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
44		SR chauffage	Default tag table	Bool	%M0.5		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
45		SR chauffage MN	Default tag table	Bool	%M3.2		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
46		SR climatiseur	Default tag table	Bool	%M0.6		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
47		SR climatiseur MN	Default tag table	Bool	%M3.4		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
48		SR lampes de ligne 1	Default tag table	Bool	%M3.0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
49		SR lampes de ligne 2	Default tag table	Bool	%M3.1		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
50		SR moteur sens droite	Default tag table	Bool	%M1.0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
51		SR moteur sens gauche	Default tag table	Bool	%M1.1		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
52		SR start programme	Default tag table	Bool	%M0.0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
53		SR urgence incendie	Default tag table	Bool	%M2.7		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
54		SR ventilation	Default tag table	Bool	%M0.7		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
55		SR ventilation MN	Default tag table	Bool	%M3.3		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
56		Start chauffage	Default tag table	Bool	%I0.2		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
57		Start climatiseur	Default tag table	Bool	%I0.4		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
58		Start moteur sens droite	Default tag table	Bool	%I1.0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
59		Start moteur sens gauche	Default tag table	Bool	%I1.1		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
60		Start ventilateurs	Default tag table	Bool	%I0.6		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
61		Start-Stop ventilation auto	Default tag table	Bool	%M3.7		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Juste pour le mode automatique
62		Stop alarme	Default tag table	Bool	%I2.0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
63		Stop chauffage	Default tag table	Bool	%I0.3		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
64		Stop climatiseur	Default tag table	Bool	%I0.5		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
65		Stop moteur de nettoyage	Default tag table	Bool	%I1.2		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
66		Stop ventilateurs	Default tag table	Bool	%I0.7		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
67		Valeur négative	Default tag table	Bool	%I1.5		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
68		Ventilation	Default tag table	Bool	%Q0.2		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
69		Ventilation MN	Default tag table	Bool	%Q1.2		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
70		<Add new>					<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	

Annexe B : Normes pour les gaz nocifs (ITAVI, 2001).

Gaz	Source	Dose	Effet
Hydrogène sulfuré H ₂ S	Décomposition des substances organiques des matières fécales	De 7ppm 20 à 150ppm 500ppm (30 minutes) 800 à 1000ppm Irritation des yeux, de l'appareil respiratoire, asphyxie. Action sur le système nerveux Coma-mort
Méthane CH ₄ (gaz de fumier)	Fermentation anaérobie des matières fécales	+ 1000ppm	Atmosphère asphyxiante Caractère inflammable
Gaz carbonique CO ₂	Respiration des animaux, mauvaise combustion d'appareil de chauffage à gaz propane		Asphyxiant Remarque : pour les pondeuses il permet d'améliorer la solidité de la coquille
Ammoniac NH ₃	Décomposition des matières fécales	20ppm 60 à 70ppm + 70ppm (en pratique ne pas dépasser 15ppm)	Irritation des voies respiratoires Lésions oculaires Réduction du gain de poids. Retard de maturité sexuelle et réduction de la production d'œufs chez les pondeuses

(ppm : mg de gaz /kg d'air)

Annexe C : Influence de la durée d'éclairage sur les performances du poulet de chair.

Durée de la photophobie	24 h	12 h	6 h	3 h
Poids à 9 semaines (grammes)	1850	1831	1804	1816
Indice de consommation	2,248	2,235	2,227	2,225

(d'après Skoglund et ses collaborateurs, 1966)

Annexe D : Température et humidité appropriée pour le poulet de chair d'après l'article du «
Misr Journal of Agricultural Engineering ». Octobre 2018 (Cobb, 2012)

Age (jours)	Humidité relative (%)	La température appropriée °C (F)
0	30-50	34 (93)
7	40-60	31 (88)
14	40-60	27 (81)
21	40-60	24 (75)
28	50-70	21 (70)
35	50-70	19 (66)
42	50-70	18 (64)