

الجمهورية الجزائرية الديمقراطية الشعبية  
REPUBLIC ALGERIENNE DEMOCRATIQUE ET POPULAIRE  
وزارة التعليم العالي والبحث العلمي  
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique  
جامعة أبي بكر بلقايد - تلمسان  
Université Aboubakr Belkaïd – Tlemcen –  
Faculté de TECHNOLOGIE



MEMOIRE

Présenté pour l'obtention du **diplôme** de **MASTER**

**En** : Génie industriel

**Spécialité** : Productique

**Par** : BAHAR Mouadh Djamel  
BELABBAS Youcef

Sujet

**Commande et supervision de trois modules : Convoyeur, Pince  
et la Table**

Soutenu publiquement, le / / , devant le jury composé de :

Mr HASSAM Ahmed	MCB	Univ. Tlemcen	Président
Mr HADRI Abdelkader	MAA	Univ. Tlemcen	Examineur
Mr MELIANI Sidi Mohammed	MCA	Univ. Tlemcen	Encadrant
Mr MKEDDER Mohamed Amine	Ingénieur	Univ. Tlemcen	Co-encadrant

**Année Universitaire : 2019– 2020**

# *Dédicace*

## **A ma mère décédée**

Que Dieu ait pitié de vous et vous fasse vivre au paradis, qui serait contente d'apprendre que son fils a enfin terminé le ses études qu'il avait commencé.

## **A mon père**

## **A mes frères**

## **A mes adorables sœurs**

## **A tous mes amis**

## **A ma chère amie**

Je te remercie infiniment pour ton soutien moral et L'assistance que tu m'as apportée pour la réussite dès l'idée jusqu'à la réalisation du projet, que dieu te protège et te fasse réussir.

**Et A tous ceux que j'aime et ceux qui m'aime. Merci**

*Dédicace*

**A MES PARENTS**

**A MES FRÈRES**

**A MES ADORABLES SŒURS**

**A TOUS MES AMIS**

**ET A TOUS CEUX QUE J'AIME ET CEUX QUI**

**M'AIME. MERCI**

## *Remerciements*

Nous remercions avant tout ALLAH le tout puissant, de nous avoir guidées pendant toutes nos années d'étude et de nous avoir donné la volonté, la patience et le courage pour terminer notre travail

Nous voulons exprimer par ces quelques lignes de remerciements nos gratitude envers tout d'abord à mon encadreur, monsieur MELIANI Mohammed et notre co-encadreur MKEDDER Amine pour ses conseils et son encadrement.

Après à tous ceux en qui par leur présence, leur soutien, leur disponibilité et leurs conseils j'ai trouvé courage afin d'accomplir ce projet.

Nous voulons aussi remercier les membres de jury, madame GHOMRI et Mr HASSAM aussi, Pour étudier et évaluer notre travail.

En fin, Nous ne pouvons pas mener à bien ce projet sans exprimer notre gratitude à tous les enseignants et enseignants de l'Université Abu Bakr Belkaid à Tlemcen, pour leur dévouement et leurs assistances tout au long de nos années d'études universitaires.

# Table des matières

Liste des figures .....	iii
Liste des tableaux.....	iii
Liste d'abréviations.....	v
<b>I CHAPITRE 01 : NOTIONS GENERALES.....</b>	<b>3</b>
I.1 INTRODUCTION .....	3
I.2 GENERALITES SUR LES SYSTEMES AUTOMATISES .....	3
I.2.1 Historique.....	4
I.2.2 Systèmes automatisés.....	5
I.2.3 Principe de fonctionnement d'un système automatisé.....	7
I.2.4 Echange d'information.....	7
I.2.5 Conséquences de l'automatisation .....	8
I.3 GENERALITES SUR LE SYSTEME DE CONTROLE ET D'ACQUISITION DE DONNEES .....	9
I.3.1 Définition de SCADA .....	9
I.3.2 Evolution du système SCADA.....	10
I.3.3 Fonctionnalités d'un système de Supervision.....	13
I.3.4 Structure d'un système SCADA.....	14
I.3.5 Protocoles de communication d'un système SCADA .....	19
I.3.6 Avantages du SCADA.....	23
I.4 EVOLUTION DES IHM .....	23
I.4.1 Définition d'une IHM.....	23
I.4.2 Utilisation d'une IHM .....	24
I.4.3 Intégration d'une IHM.....	24
I.4.4 Avantages d'une IHM.....	25
I.4.5 Ergonomie des interfaces.....	26
I.4.6 Composants techniques des interfaces graphiques .....	26
I.4.7 Différence entre IHM et SCADA.....	27
I.4.8 Logiciels existant pour la création d'une IHM .....	27
I.5 CONCLUSION .....	30
<b>II CHAPITRE 02 : DESCRIPTION DU SYSTEME ETUDIE.....</b>	<b>34</b>
II.1 INTRODUCTION .....	32
II.2 GENERALITES SUR L'USINE ET LE MODE DE FONCTIONNEMENT DE LA CHAINE DE PRODUCTION .....	32
II.2.1 Présentation et objectif du laboratoire .....	32
II.2.2 Description du système étudié .....	33
II.3 GENERALITES SUR LES AUTOMATES PROGRAMMABLES.....	38
II.3.1 Historique.....	38

II.3.2	<i>Définition de l'automate programmable</i> .....	39
II.3.3	<i>Choix d'un automate : quelques critères à considérer</i> .....	39
II.3.4	<i>Structure générale des API</i> .....	40
II.3.5	<i>Structure interne d'un automate programmable industriel (API)</i> .....	41
II.3.6	<i>Domaines d'emploi des automates</i> .....	43
II.3.7	<i>Nature des informations traitées par l'automate</i> .....	44
II.3.8	<i>Langage de Programmation</i> .....	45
II.3.9	<i>Avantages et inconvénients des API</i> .....	48
II.3.10	<i>Logiciel de programmation d'API (STEP 7)</i> .....	48
II.4	CONCLUSION .....	49
<b>III</b>	<b>CHAPITRE 03 : PROGRAMMATION ET REALISATION D'UNE IHM</b> .....	<b>35</b>
III.1	INTRODUCTION .....	50
III.2	ETABLISSEMENT DU CAHIER DE CHARGE .....	50
III.3	PROGRAMMATION DU FONCTIONNEMENT DE NOTRE MINI-CHAINE SUR STEP7 .....	50
III.3.1	<i>Tâches fondamentales</i> .....	50
III.3.2	<i>Conception du programme de fonctionnement</i> .....	52
III.4	CREATION D'UNE IHM .....	57
III.4.1	<i>Création d'un projet WinCC</i> .....	58
III.4.2	<i>Établissement d'une liaison IHM-PLC</i> .....	58
III.5	CONCLUSION .....	61
	Conclusion Générale .....	62
	Références Bibliographiques .....	64
	Annexe .....	67

# Liste des figures

Figure I. 1 : Structure d'un Système automatisé. [5].....	5
Figure I. 2 : Schéma de principe d'un système automatisé. [6].....	7
Figure I. 3 : Echange d'information dans un système automatisé. [6] .....	8
Figure I. 4 : Organisation logicielle de superviseur. [7].....	10
Figure I. 5 : Architecture monolithique. [8] .....	11
Figure I. 6 : Architecture monolithique. [8] .....	12
Figure I. 7 : Architecture en réseau. [8].....	12
Figure I. 8 : Architecture en internet des objets. [8].....	13
Figure I. 9 : Schéma synoptique du SCADA. [11].....	17
Figure I. 10 : Architectures SCADA. [7] .....	18
Figure I. 11 : Combinaison des outils softs et hard des systèmes SCADA. [7] .....	22
Figure I. 12 : Interface graphique de supervision d'un générateur de vapeur. [7] .....	22
Figure I. 13 : Logo logiciel zinon. [15] .....	27
Figure I. 14 : Editeur movocon. [17] .....	27
Figure I. 15 : Logiciel myscada. [19] .....	28
Figure I. 16 : Logiciel WinCC. [21] .....	28
Figure I. 17 : Vue principale de TIA Portal. [21] .....	29
Figure I. 18 : Représentation des différentes zones trouvant dans la vue principale de TIA Portal. [21] .....	30
Figure II. 19 : Photo du module convoyeur.....	33
Figure II. 20 : Photo du module Table tournante. ....	34
Figure II. 21 : Photo du module Bras pneumatique.....	36
Figure II. 22 : Photo du module de commande mod. PLC-V7/EV. [23] .....	37
Figure II. 23 : Aspect extérieur d'un automate S7-300 CPU 314 2PN/DP.....	41
Figure II. 24 : Structure interne d'un automate programmable industriel (API). [3].....	42
Figure II. 25 : Fonctionnement cyclique d'un API. [3].....	42
Figure II. 26 : Temps de scrutation vs Temps de réponse. [3] .....	43
Figure II. 27 : Temps de réponse total. [26] .....	44
Figure II. 28 : Langage FBD. [27].....	47
Figure III. 29 : Tâches fondamentales pour crée un projet sur step7. ....	51
Figure III. 30 : Vue globale du projet et ces constituants principaux.....	52
Figure III. 31 : Table de mnémoniques obtenue de Step7.....	53
Figure III. 32 : Représentations des 04 premiers états de notre programme .....	54
Figure III. 33 : Représentation des états de déplacement de la pince.....	55
Figure III. 34 : Représentation d'OB1 après compilation. ....	55
Figure III. 35 : Simulation en représentant l'état initial et le franchissement. ....	56
Figure III. 36 : Activation de l'état S3 et mise à 1''de '7Y00' au niveau du simulateur (A0.0). .....	57
Figure III. 37 : Création du projet et son insertion à Step7. ....	58

Figure III. 38 : Représentation de la communication entre notre IHM et l'API. ....	59
Figure III. 39 : Table des variables obtenue de WinCC. ....	59
Figure III. 40 : Représentation de plusieurs figures avec des orientations différentes.....	60
Figure III. 41 : Représentation des propriétés de la figure de droite : 'p1' comme variable, 'visible' comme évènement.....	61

## Liste des tableaux

Tableau I. 1 : Type de communication entre le maitre et l'esclave. [9] .....	20
Tableau I. 2 : Quelques composants ou contrôles graphiques essentiels. ....	26

# Liste d'abréviations

**Win CC** : Windows contrôle center

**SCADA**: Supervisory Control And Data Acquisition

**API** : Automates programmables industriels

**IHM** : Interface homme/machine.

**PLC** : Programmable logic controllers

**TOR** : Tout ou rien

**CPU** : Central Processing Unit

**RTU** : Unité terminale distante

**ASCII** : American Standard Code for Information Interchange

**E/S** : Entrées/sorties.

**CONT** : Schéma à contacts

**LOG** : Logigramme.

**LIST** : La liste d'instruction.

**VAR** : Variable.

**FBD** : Function Block Diagram, Logigramme.

**SFC** : Sequential function char, Le Grafcet.

**SCL** : Structured Control Language.

**FM** : Modules fonctionnels.

**OB** : Bloc d'organisation.

**DB** : Bloc des données.

**MPI** : Multi Point Interface.

**SM** : Modules de signaux.

**SIMATIC** : Siemens Automatique.

# Introduction générale

"Depuis toujours l'homme est en quête de bien être". Cette réflexion (qui rejoint la notion de besoin) peut paraître bien éloignée d'un cours de Sciences Industrielles, pourtant c'est la base de l'évolution des sciences en général et de l'automatisation en particulier. L'homme a commencé par penser, concevoir et réaliser. Lorsqu'il a fallu multiplier le nombre d'objets fabriqués, produire en plus grand nombre, l'automatisation des tâches est alors apparue remplacer l'homme dans des actions pénibles, délicates ou répétitives.

Citons pour exemple quelques grands hommes, avec les premiers développements de l'ère industrielle au XVIIIème siècle, Watt, avec ses systèmes de régulation à vapeur, Jacquard et ses métiers à tisser automatiques. Une liste exhaustive serait bien difficile à établir.

Le développement des connaissances, et des outils mathématiques, ont conduit à un formidable essor des systèmes automatisés, et des systèmes asservis, dans la deuxième moitié du 20ème siècle. Certains se hasardent à rapprocher l'Automatique et la philosophie, observant d'étranges similitudes entre les processus propres à l'homme et l'approche technologique [1].

L'automatisation est la priorité absolue dans les industries modernes. Elle consiste à assurer la conduite par un dispositif technologique qui se substitue aux opérations manuelles. Le système ainsi conçu sait prendre en compte les situations pour lesquelles sa commande a été réalisée. Ceci prend toute son importance lorsqu'il s'agit de système dont on a l'obligation de contrôler les changements, en vue d'arriver à une précision optimale.

Un bon contrôle des processus nécessite nécessairement une transparence maximale. Ceci est possible grâce à la supervision avec l'Interface Homme-Machine (IHM), qui permet à tout moment de contrôler les différents paramètres du système, mais aussi de surveiller le fonctionnement du processus en temps réel, et de diagnostiquer avec précision toutes sortes de pannes pouvant survenir, afin d'obtenir un grand contrôle de la production.

La supervision libère les opérateurs des tâches délicates, notamment dans les environnements hostiles, elle rend également le travail moins contraignant pour ceux qui le font et améliore les conditions de travail.

Notre but est de programmer le fonctionnement de deux modules un convoyeur et une pince dans le laboratoire MELT utilisant le logiciel STEP7, puis les commander et les superviser à travers une IHM développée par le logiciel WinCC flexible 2008.

A cet effet, le présent mémoire est réparti en trois chapitres décrivant les volets principaux.

Premier chapitre porte sur des généralités des systèmes automatisés, généralités sur le système SCADA et l'évolution des IHM.

Dans le deuxième chapitre, nous apportons la description du system étudié en précisant les modules du mini chaine ainsi son fonctionnement et leurs composants, ensuite une description sur les automates programmables a été apportée pour enrichir notre étude.

Troisième chapitre est consacré à la programmation et la réalisation d'une interface homme-machine pour la partie supervision et du test du programme réalisé.

# Chapitre 01

## Notions générales

## I.1 Introduction

Les systèmes industriels sont devenus plus intelligents et plus complexes, les besoins de les surveiller et de contrôler plusieurs périphériques dans l'usine ont augmenté. C'est pour cette raison qu'il y a eu des améliorations au niveau des systèmes de supervision.

Le système SCADA (Supervisory Control And Data Acquisition) est une solution très performante pour le contrôle des systèmes industriels complexes et qui regroupe l'ensemble des fonctionnalités du superviseur [2].

C'est ainsi que nous proposons dans cette communication un modèle général pour l'analyse et la conception d'un système SCADA.

Donc dans ce chapitre nous allons suivre ce plan :

1. Généralités sur les systèmes automatisés.
2. Généralités sur le système SCADA
3. Evolution des IHM
4. Conclusion

## I.2 Généralités sur les systèmes automatisés

Automatisme est la discipline traitant d'une part la caractérisation des systèmes automatisés et d'autre part le choix de la conception et de la réalisation de la partie commande.

Il s'agit donc d'étudier les systèmes :

- Réalisant leurs fonctions en relative autonomie.
- Assurant un contrôle des performances par la mise en place possible d'une chaîne de retour.

L'automatisation de la production consiste à transférer tout ou une partie des tâches de coordination, auparavant exécutés humains et qu'est pour objectifs de :

- Eliminer les tâches répétitives
- Simplifier le travail de l'humain
- Augmenter la sécurité
- Accroître la productivité
- Economiser les matières premières et l'énergie
- S'adapter à des contextes particuliers
- Maintenir la qualité [3].

## **I.2.1 Historique**

La révolution industrielle entraîne la formation d'un nouveau mode de production et la mise en place d'une organisation scientifique du travail, le travail à la chaîne.

### **a) 1<sup>ère</sup> révolution industrielle : Production mécanique**

Elle correspond à l'exploitation du charbon et l'utilisation de la machine à vapeur créée par James Watt en 1769.

La production mécanique va remplacer l'artisanat, les manufactures et les ateliers artisanaux vont laisser place aux grandes usines. Au sein des usines, les machines vont être actionnées grâce aux machines à vapeur, ce qui va permettre d'accroître fortement les cadences de production.

Ainsi, les volumes de fabrication vont être plus importants et vont donner naissance à des produits en petites séries.

### **b) 2<sup>ème</sup> révolution industrielle : Production de masse**

Elle arrive à la fin du XIX<sup>ème</sup> siècle au travers de l'utilisation du pétrole et de l'électricité. Ces deux éléments vont permettre la modernisation des moyens de production à grandes échelles. Ce changement impacte principalement l'industrie automobile et l'industrie chimique.

Durant cette période vont être mis en place le taylorisme et le travail à la chaîne. La productivité des ouvriers non-qualifiés va être exploitée au maximum pour voir naître la production en masse de produits identiques.

### **c) 3<sup>ème</sup> révolution industrielle : Production automatisée**

La 3<sup>ème</sup> révolution va avoir lieu au milieu du XX<sup>ème</sup> siècle grâce à l'introduction de l'électronique et des technologies informatiques. Elles vont permettre la mise en place de l'automatisation de la production dans les usines.

Ces technologies viennent remplacer et soulager les ouvriers dans les tâches les plus difficiles. On verra apparaître à cette période la robotique et la production de grandes séries.

### d) 4<sup>ème</sup> révolution industrielle : la transformation numérique

De nos jours, les tendances de production à la chaîne ont été laissées de côté pour tendre vers l'ère de la personnalisation des produits et services. L'industrie 4.0 va permettre de répondre à cette problématique des produits uniques et personnalisés tout en maintenant des coûts équivalents, malgré les faibles volumes de production demandés.

Le passage à l'industrie 4.0 est associé à l'introduction de l'Internet des Objets (IoT), aux systèmes cyber-physiques et à l'essor des réseaux interconnectés à travers le monde.

Les grands industriels placent désormais la transformation numérique comme une priorité stratégique. Pour cela, plus de 900 milliards de dollars d'investissement sont prévus par an entre 2016 et 2020. Assumant une fois de plus une profonde transformation de leur mode de fonctionnement et projetant cette industrie vers un prometteur XXI<sup>e</sup> siècle [4].

### I.2.2 Systèmes automatisés

Un système automatisé est toujours composé d'une partie commande (PC) et d'une partie opérative (PO) pour faire fonctionner ce système. L'opérateur (personne qui va faire fonctionner le système) va donner des consignes à la partie (PC), celle-ci va traduire ces consignes en ordres qui vont être exécutés par la (PO).

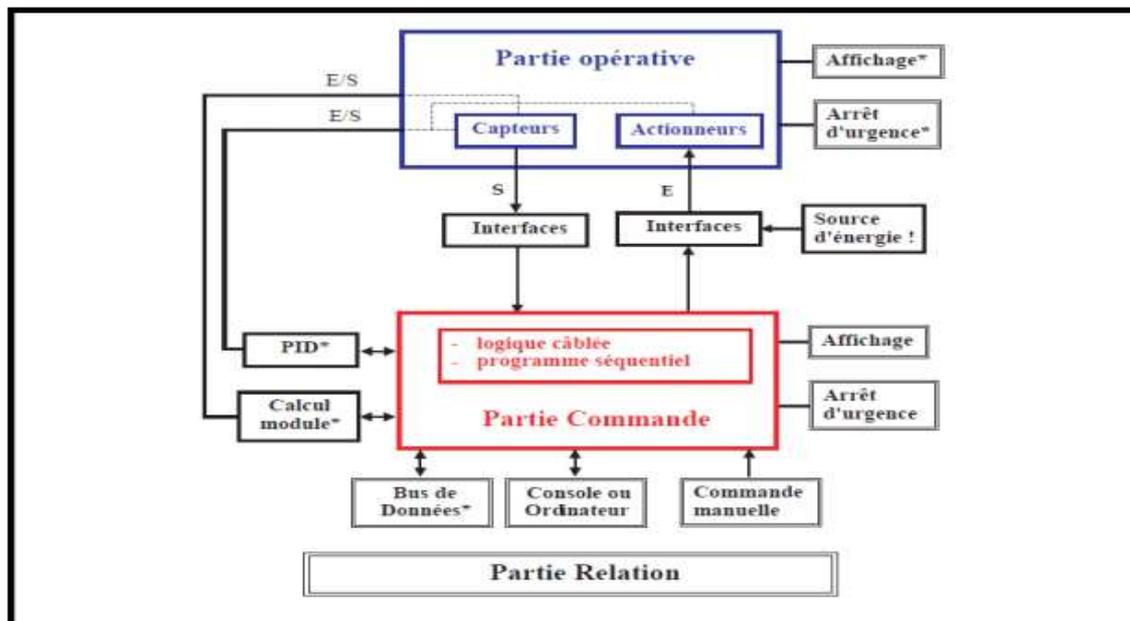


Figure I. 1 : Structure d'un Système automatisé. [5]

Une fois les ordres accomplis, la PO va le signaler à la PC (compte -rendu) qui va à son tour le signaler à l'opérateur, ce dernier pourra donc dire que le travail a bien été réalisé [5].

#### **a) Partie opérative (PO)**

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

- Du programme qu'elle contient.
- Des informations reçues par les captures.
- Des consignes données par l'utilisateur ou par l'opérateur.[5]

#### **b) Partie commande (PC)**

Ce secteur de l'automatisme gère selon une suite logique le déroulement ordonné des opérations à réaliser. Il reçoit des informations en provenance des capteurs de la Partie Opérative, et les restitue vers cette même Partie en direction des pré-actionneurs et actionneurs. Elle comporte en générale un boîtier (appelé aussi bâti) contenant :

- Des actionneurs (transformant l'énergie reçu en énergie utile : moteur, vérin, lampe).
- Des capteurs (transforment la variation des grandeurs physiques liée au fonctionnement de l'automatisme en signaux électriques : capteur de position, de température, bouton poussoir) [5].

#### **c) Partie relation (PR)**

Sa complexité dépend de l'importance du système. Elle regroupe les différentes commandes nécessaires au bon fonctionnement du procédé, c'est à dire marche/arrêt, arrêt d'urgence, marche automatique, etc.... L'outil de description s'appelle le Guide d'Etudes des Modes de Marches et d'Arrêts (GEMMA). Les outils graphiques, que sont le GRAFCET et le GEMMA, sont utilisés par les automaticiens et les techniciens de maintenance [5].

### I.2.3 Principe de fonctionnement d'un système automatisé

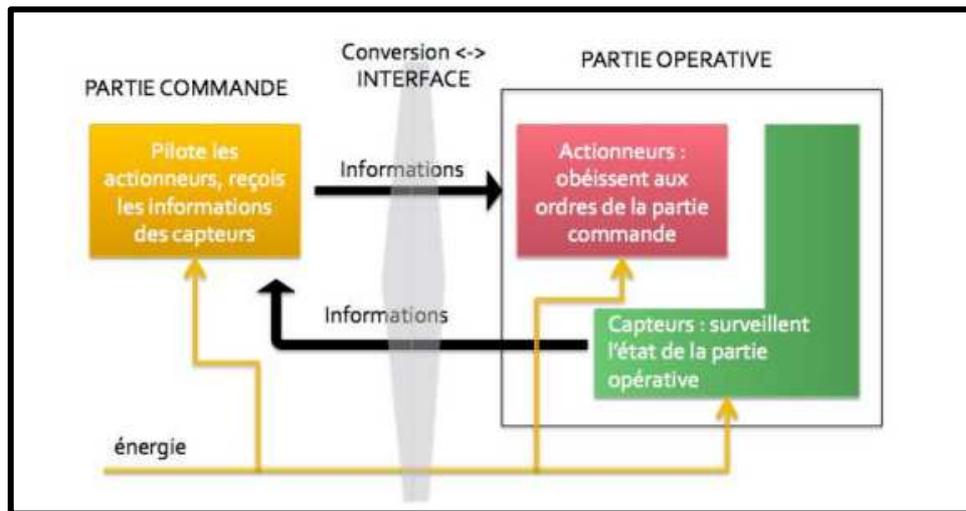


Figure I. 2 : Schéma de principe d'un système automatisé. [6]

La partie commande envoie des ordres aux actionneurs, elle reçoit des informations d'état en provenance des capteurs. Il y a donc une chaîne de transmission entre la partie opérative et la partie commande.

Chaque partie doit aussi être alimentée en énergie, on parle donc d'une chaîne d'énergie.[6]

### I.2.4 Echange d'information

L'ensemble des échanges d'informations est contrôlé par le programme de la partie commande :

- L'opérateur donne des consignes à la partie commande.
- La partie commande adresse des ordres à la partie opérative.
- Les actionneurs exécutent les ordres reçus : production d'un phénomène physique.
- Les capteurs réagissent à une variation d'état : détection d'un phénomène physique.
- La partie opérative adresse des comptes-rendus à la partie commande.
- La partie commande envoie à l'opérateur des signaux sur l'état du système ou de son environnement.

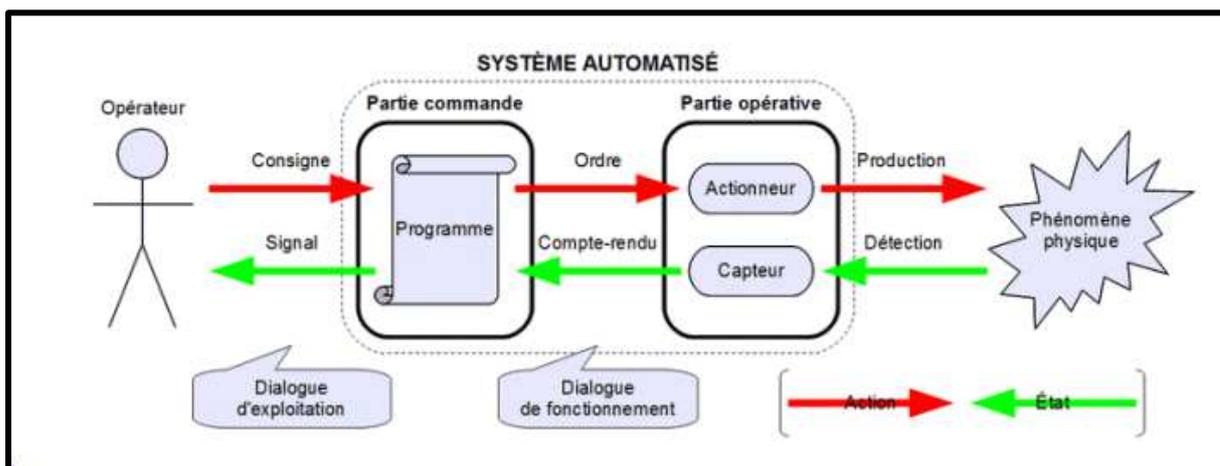


Figure I. 3 : Echange d'information dans un système automatisé. [6]

Il s'établit un dialogue d'exploitation entre l'opérateur et la partie commande, et un dialogue de fonctionnement entre la partie commande et la partie opérative. [6]

## I.2.5 Conséquences de l'automatisation

### a) Avantages de l'automatisation

- Améliorer les conditions de travail (effectuer des tâches pénibles, dangereuses et répétitives)
- Sécurité
- Précision
- Réduire les coûts de fabrications (produit plus compétitif)
- Augmenter la productivité (réduire le temps de travail nécessaire à la production, donc augmenter les cadences de travail)
- Flexibilité (une machine peut s'adapter à plusieurs productions)
- Confidentialité (une machine ne peut pas parler)
- Un S.A. peut travailler 24h sur 24h. [6]

### b) Inconvénients de l'automatisation

- Incidence sur l'emploi (licenciement –chômage : la mise en place d'une machine se substituant à 10 salariés n'aboutit pas à la création de 10 emplois)
- Investissement pour l'achat de machines
- Coût de maintenance
- Pannes
- Consommation d'énergie
- Formation d'un personnel plus qualifié (technicien de maintenance, de contrôle...) [6].

## **I.3 Généralités sur le Système de contrôle et d'acquisition de données**

La conduite d'un procédé dans le domaine industriel implique la connaissance, la surveillance et la maîtrise de certains paramètres tels que la pression, la température, le débit, ...etc.

Chaque procédé possède ses exigences propres, et chaque équipement a ses conditions de fonctionnement. Le système de contrôle-commande doit satisfaire ces besoins.

Les installations industrielles dans le domaine du pétrole et du gaz présentent des risques pour les personnes, l'environnement et les équipements d'où la nécessité de mise en œuvre des systèmes de mise en sécurité de ces installations à risque pour le respect des exigences

Les technologies nouvelles comme SCADA (Supervisory Control And Data Acquisition) nous offrent des avantages multiples qui nous garantissent un retour sur investissement particulièrement rapide, et des résultats quantitativement et qualitativement excellents.[7]

### **I.3.1 Définition de SCADA**

SCADA (Supervisory Control And Data Acquisition) est synonyme de contrôle de surveillance et d'acquisition de données. Comme son nom l'indique, désigne une catégorie de logiciels destinés au contrôle de processus et à la collecte de données en temps réel auprès de sites distants, en vue de contrôler des équipements et des conditions d'exploitation. Il ne s'agit pas d'un système de contrôle complet, mais plutôt d'un niveau de supervision.

Il s'agit d'un progiciel purement logiciel positionné sur le matériel auquel il est interfacé, en général via des automates programmables (PLC) ou d'autres modules matériels commerciaux.

On peut voir aussi que c'est un système de télégestion à grande échelle permettant de traiter en temps réel un grand nombre de télémessures et de contrôler à distance des installations techniques. C'est une technologie industrielle dans le domaine de l'instrumentation, dont les implémentations peuvent être considérées comme des Framework1 d'instrumentation incluant une couche de type middleware2. On trouve par exemple des systèmes SCADA dans les contextes suivants :

- Surveillance de processus industriels
- Transport de produits chimiques
- Réseaux municipaux d'approvisionnement en eau

- Commande de la production d'énergie électrique
- Distribution électrique, canalisations de gaz et de pétrole
- Réseaux de chaleur
- Recherche et études scientifiques et industrielles.

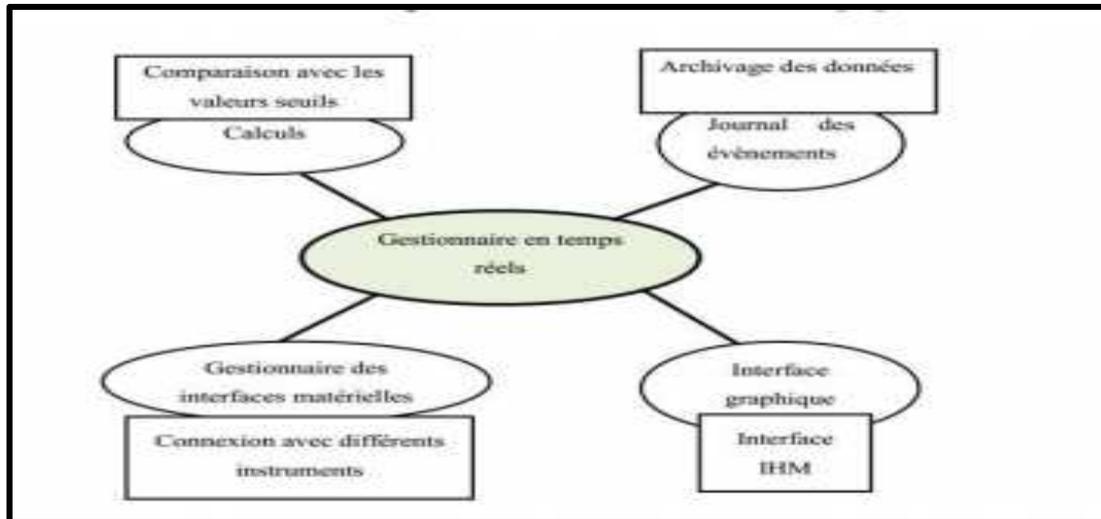


Figure I. 4 : Organisation logicielle de superviseur. [7]

Les systèmes SCADA sont largement utilisés dans l'industrie pour l'acquisition de données, le pilotage et la surveillance de processus industriels [7].

### I.3.2 Evolution du système SCADA

L'utilisation du système SCADA est devenue populaire à partir des années 1960 avec le besoin croissant de surveiller et de contrôler le matériel. Les premiers systèmes construits à l'aide d'ordinateurs centraux étaient coûteux, car ils étaient actionnés et surveillés manuellement.

Mais les progrès récents de la technologie ont permis de créer des systèmes SCADA automatisés et avancés offrant une efficacité maximale à un coût réduit, conformément aux exigences alarmantes de l'entreprise.

Dans cette partie, on s'intéresse à l'évolution des systèmes SCADA et description des quatre générations suivantes :

#### a) Première Génération : « Monolithique »

Les calculs sont réalisés avec des ordinateurs centraux. Les réseaux n'existant pas à cette époque, les systèmes SCADA sont indépendants et ne sont connectés à aucun autre système. Les réseaux étendus seront conçus par les fournisseurs de RTU (Remote Terminal Unit).

Les protocoles de communication utilisés sont le plus souvent propriétaires.

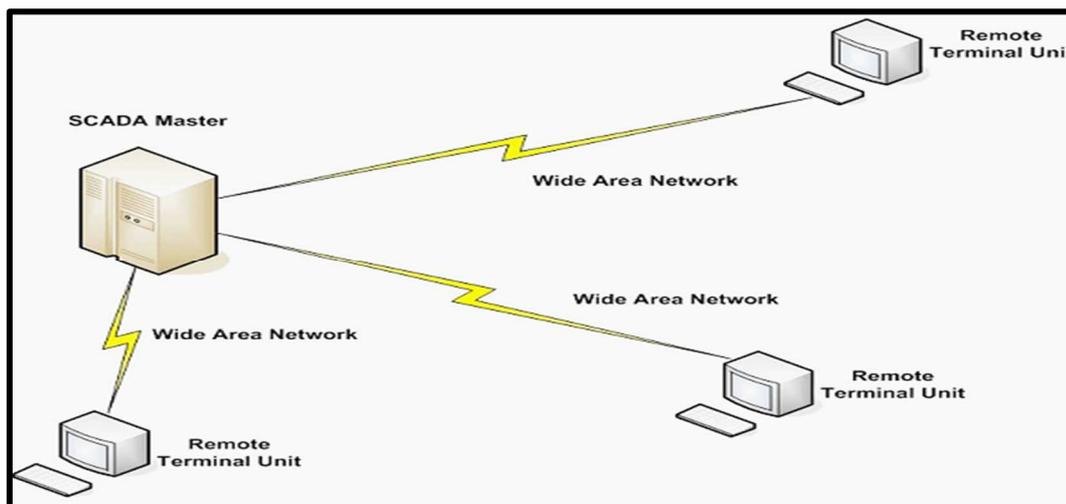


Figure I. 5 : Architecture monolithique. [8]

La première génération de systèmes SCADA est redondante car un ordinateur central de secours est connecté au niveau du bus informatique et activé en cas de panne de l'ordinateur central principal.

Certains systèmes SCADA de première génération sont des solutions «clé en main» et mis en œuvre sur des mini-ordinateurs comme le PDP-113 de Digital Equipment Corporation (DEC). Ces systèmes peuvent afficher des informations provenant des systèmes de contrôle analogiques existant aux postes de travail des opérateurs individuels, mais ils n'ont généralement pas la fonction d'envoyer des signaux de commande aux stations éloignées. Ils ont aussi des fonctions d'alarme et de journalisation et assurent des tâches de comptabilité horaires et quotidiennes. [8]

#### b) Deuxième Génération : « Distribuée »

Le traitement est réparti entre plusieurs stations reliées par un réseau local et qui échangent des informations en temps réel. Chaque station est responsable d'une tâche particulière, ce qui rend la taille et le coût de chaque poste inférieur à ceux utilisés dans la première génération. Les protocoles réseau utilisés sont encore majoritairement propriétaires, ce qui conduit à des problèmes de sécurité importants des systèmes SCADA soumis aux pirates informatiques.

Étant donné que les protocoles étaient propriétaires, très peu de gens en dehors des développeurs et des pirates en savent assez pour déterminer comment sécuriser un système SCADA. Les deux parties ayant intérêt à maintenir un certain niveau d'obscurité, la sécurité d'un système SCADA est souvent surestimée, si toutefois elle a été prise en compte. [8]

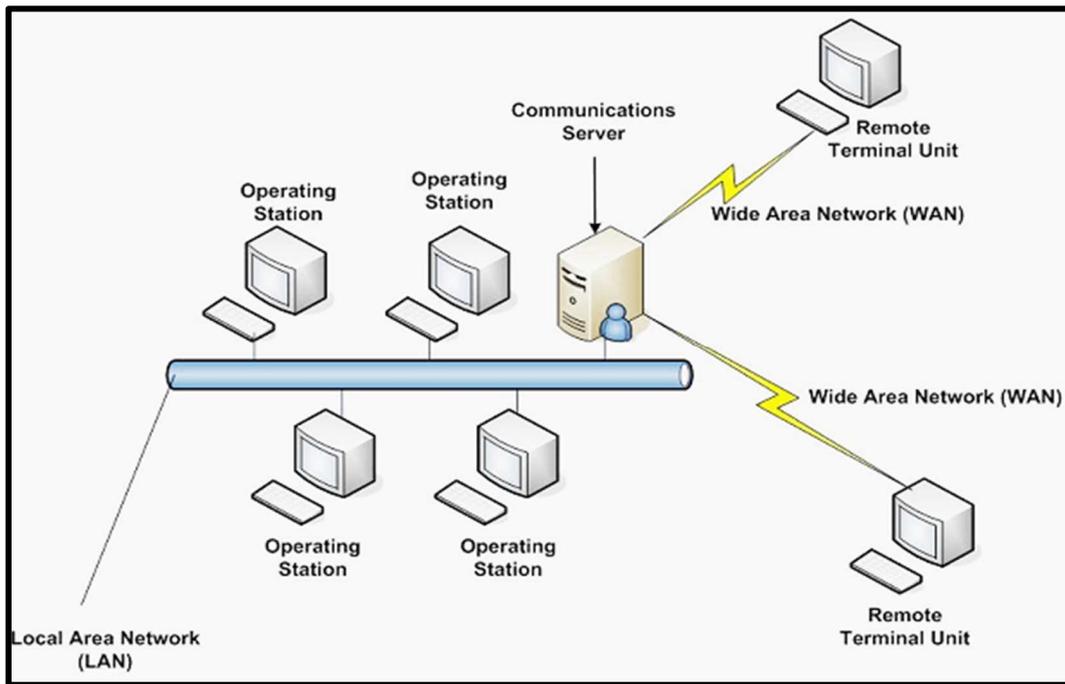


Figure I. 6 : Architecture monolithique. [8]

**c) Troisième Génération : « en réseau »**

Grâce à l'utilisation de protocoles standards et le fait que de nombreux systèmes SCADA en réseau sont accessibles à partir d'Internet, les systèmes sont potentiellement vulnérables à une attaque à distance. D'autre part, l'utilisation de protocoles standards et de techniques de sécurité signifie que les améliorations des standards de sécurité sont applicables aux systèmes SCADA, en supposant qu'ils reçoivent la maintenance et les mises à jour en temps opportun.

[7]

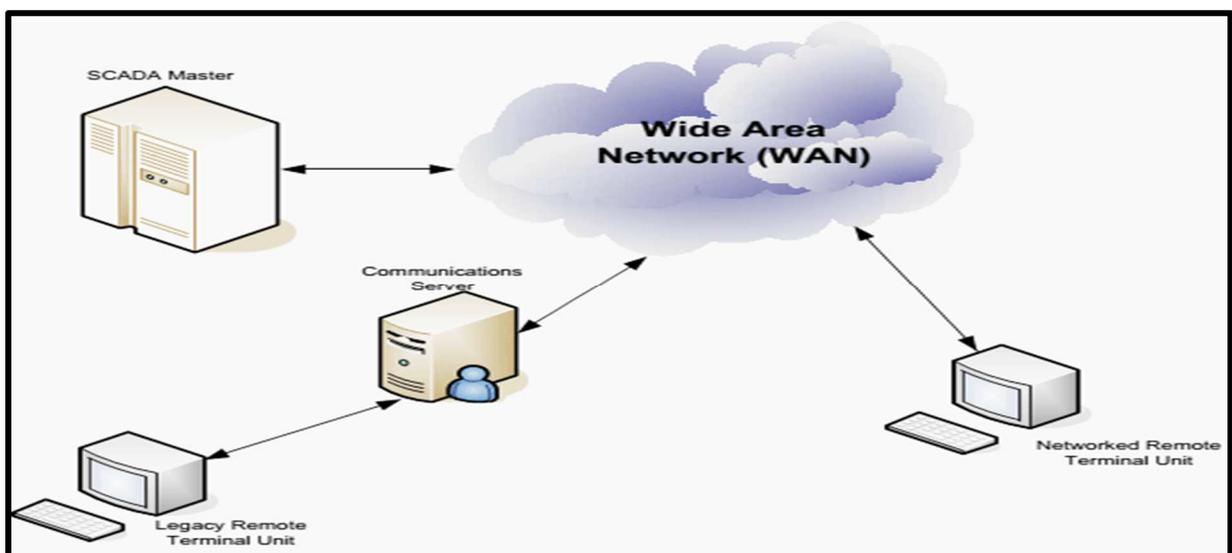


Figure I. 7 : Architecture en réseau. [8]



- Le fonctionnement en l'absence de défaillance,
- La reprise ou gestion des modes,
- Les traitements d'urgence,
- Une partie de la maintenance corrective.

Les fonctions de commande en marche normale sont :

- L'envoi de consignes vers le procédé dans le but de provoquer son évolution
- L'acquisition de mesures ou de compte-rendu permettant de vérifier que les consignes envoyées vers le procédé produisent exactement les effets escomptés.
- L'acquisition de mesures ou d'informations permettant de reconstituer l'état réel du procédé et/ou du produit.
- L'envoi vers le procédé d'ordres prioritaires permettant de déclencher des procédures de sécurité (arrêts d'urgence par exemple). [7]

### **I.3.3.2 Surveillance**

La partie surveillance d'un superviseur a pour objectifs :

- La détection d'un fonctionnement ne correspondant plus à ce qui est attendu.
- La recherche des causes et conséquences d'un fonctionnement non prévu ou non contrôlé.
- L'élaboration de solutions permettant de pallier le fonctionnement non prévu
- La modification des modèles utilisés pendant le fonctionnement prévu pour revenir à ce fonctionnement : changement de la commande, réinitialisations, etc.,
- La collaboration avec les opérateurs humains pour les prises de décision critiques, pour le recueil d'informations non accessibles directement et pour l'explication de la solution curative envisagée ou appliquée.[7]

## **I.3.4 Structure d'un système SCADA**

### **I.3.4.1 SCADA pour l'automatisation et le contrôle**

Une fois la configuration des installations du système est réalisée, à savoir : la programmation des entrées/sorties du contrôleur, la configuration du serveur, la configuration des instruments de mesure, la communication entre les différents équipements, il convient de se concentrer sur la mise en place d'un système SCADA.

En effet, le système SCADA est capable de transférer des informations entre une salle de contrôle composé d'un ou plusieurs ordinateurs « hôte » ou encore appelé MTU (Remote

Master Unit) à un certain nombre des unités terminales distantes RTU ou à des automates situés au même endroit ou à des endroits différents.

Le système SCADA permet également le contrôle du processus à travers la commande de certains appareils à distance (ouverture et la fermeture des actionneurs), l'envoi des instructions (profile de charge), etc.

De cette façon, le SCADA assure la surveillance et la sécurité de l'ensemble du système en émettant des données critiques. Il permet de prendre des décisions d'urgence qui ont pour but d'améliorer la maintenance des installations [9].

#### **a) Gestion d'alarme**

La configuration des alarmes a évolué de façon spectaculaire au cours des dernières décennies. Dans les années 1970, l'ajout d'une nouvelle alarme était complexe et présentait des coûts élevés. De nos jours, avec l'utilisation des systèmes SCADA, cela est devenu plus facile à réaliser.

Le rôle principal des alarmes est de maximiser la productivité du système de production et de fournir plus de sécurité tant aux installations qu'aux opérateurs. La gestion des alarmes ne consiste pas seulement à réduire les anomalies, mais également de rendre les opérateurs à devenir plus efficaces ceci en obtenant la bonne information au bon moment. [7]

#### **b) Archivage des données**

L'archivage des données est l'un des principaux éléments d'un système SCADA. De ce fait, il permet d'archiver les données provenant de diverses installations du système. Les informations enregistrées peuvent être des données structurées (données relationnelles...) ou des données non structurées (documents, images ...).[7]

#### **c) Interface homme-machine IHM**

L'interaction entre l'utilisateur du système SCADA et les différents équipements d'une installation se fait via une interface graphique. Cette interface contient des représentations synoptiques qui centralisent le contrôle d'un procédé sur un seul écran. L'IHM permet d'avoir un affichage clair et simple des différentes données du système (états d'entrées /sorties, alarmes, etc.) en temps réel.

### **I.3.4.2 Technologie de SCADA**

Les systèmes SCADA ont fait des progrès considérables au cours des dernières années en termes de fonctionnalités, l'évolutivité, la performance et l'ouverture.

Les systèmes SCADA ne sont pas utilisés uniquement dans la plupart des processus industriels tels que les industries de transformation, le pétrole et le gaz, la production d'énergie électrique, la distribution, les services publics, le traitement et la gestion des eaux et des déchets, l'agriculture / l'irrigation, la fabrication, les systèmes de transport, etc. mais aussi dans certaines installations expérimentales telles que la fusion nucléaire.

Les systèmes SCADA comprennent des composants matériels et logiciels. Les composants matériels collectent les données et les rassemblent sur un ordinateur équipé d'un logiciel SCADA. L'ordinateur traite alors ces données et les présente en temps opportun. En outre, le système SCADA enregistre et journalise tous les événements dans un fichier stocké sur un disque dur ou les envoie à une imprimante.

Enfin, le système SCADA émet des alarmes lorsque les conditions deviennent dangereuses ou non conforme au fonctionnement normal [10].

### **I.3.4.3 Composants des systèmes SCADA**

Avant de discuter l'architecture du SCADA et les différents types de systèmes SCADA, nous devons avant tout connaître quelques notions de base sur le SCADA. Le schéma synoptique du système SCADA est présenté dans la figure 2.6, qui comprend différents blocs, à savoir une interface homme-machine (IHM), un système de supervision, des unités terminales distantes, des automates programmables, une infrastructure de communication et des programmes SCADA [11].

Un système SCADA se distingue par :

#### **a) Composants software**

- Les flux de communication dans les systèmes SCADA.
- Les architectures réseaux par besoin fonctionnel.
- Les protocoles de communication temps réel, PLC.
- Les langages de programmation d'automatismes industriels [11]

#### **b) Composants hardware**

- Automates, vannes, capteurs chimiques ou thermiques, système de commande et contrôle (sera bien expliqué dans le chapitre 2)
- La conception d'un système de contrôle en réponse à un cahier des charges (sera bien expliqué en chapitre 3)
- IHM (sera bien expliqué en détail dans le paragraphe 1.4 de ce chapitre)

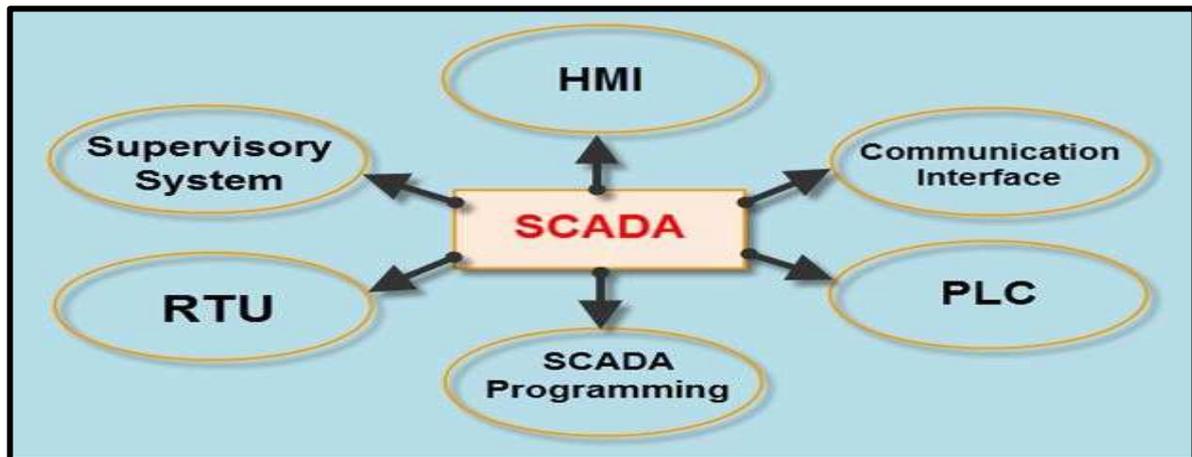


Figure I. 9 : Schéma synoptique du SCADA. [11]

### c) Composants software

En générale, la combinaison de connexions radio et filaires directes est utilisée pour les systèmes SCADA, mais dans le cas de systèmes de grande taille tels que les centrales électriques et les chemins de fer, SONET<sup>4</sup> / SDH<sup>5</sup> est fréquemment utilisé.

La programmation SCADA dans un maître ou une interface homme-machine est utilisée pour créer des plans et des diagrammes qui fourniront des informations importantes en cas de défaillance d'événement ou de défaillance de processus.

Les interfaces standard sont utilisées pour programmer la plupart des systèmes SCADA commerciaux. La programmation SCADA peut être effectuée à l'aide d'un langage de programmation dérivé ou d'un langage C. [7]

### d) Composants Hardwār

- **Système de surveillance MTU**

Le système de supervision sert de serveur pour la communication entre les équipements du système SCADA, tels que les RTU, les API et les capteurs, etc., et le logiciel IHM utilisé dans les postes de travail de la salle de contrôle

La station maîtresse ou la station de supervision comprend un seul PC dans des systèmes SCADA plus petits et, dans le cas de systèmes SCADA plus grands, le système de supervision comprend des applications logicielles distribuées, des sites de reprise après incidents et plusieurs serveurs. Ces serveurs multiples sont configurés dans une configuration de secours immédiat ou à double redondance, qui contrôle et surveille en permanence en cas de défaillance du serveur afin d'accroître l'intégrité du système.

- **Unités des Terminaux distants RTU**

Les éléments physiques dans les systèmes SCADA sont interfacés avec les dispositifs électroniques contrôlés par microprocesseur, appelés unités de terminaux distants (RTU). Ces unités sont utilisées pour transmettre des données de télémétrie au système de supervision et recevoir les messages du MTU pour contrôler les objets connectés. Par conséquent, elles sont également appelées unités de télémétrie distante.

- **Automates logiques programmables (API)**

Dans les systèmes SCADA, des automates sont connectés aux capteurs pour collecter les signaux de sortie des capteurs afin de convertir ces signaux en données numériques. Les API sont utilisées à la place des RTU en raison de leurs avantages tels que la flexibilité, la configuration, la polyvalence et l'accessibilité économique par rapport aux RTU. [9]

### I.3.4.4 Architecture du SCADA

L'évolution d'instrumentation intelligente et de terminaux distants (RTU) / contrôleurs logiques programmables (PLC) ont permis aux solutions de processus-contrôle de nombreux secteurs d'être facilement gérées et exploitées en utilisant les avantages d'un système SCADA.

Généralement, le système SCADA comprend les composants suivants : processeurs locaux, équipements d'exploitation, automates programmables, instrument, terminal distant, appareil électronique intelligent, terminal maître ou ordinateurs hôtes et un PC avec interface homme-machine. [7]

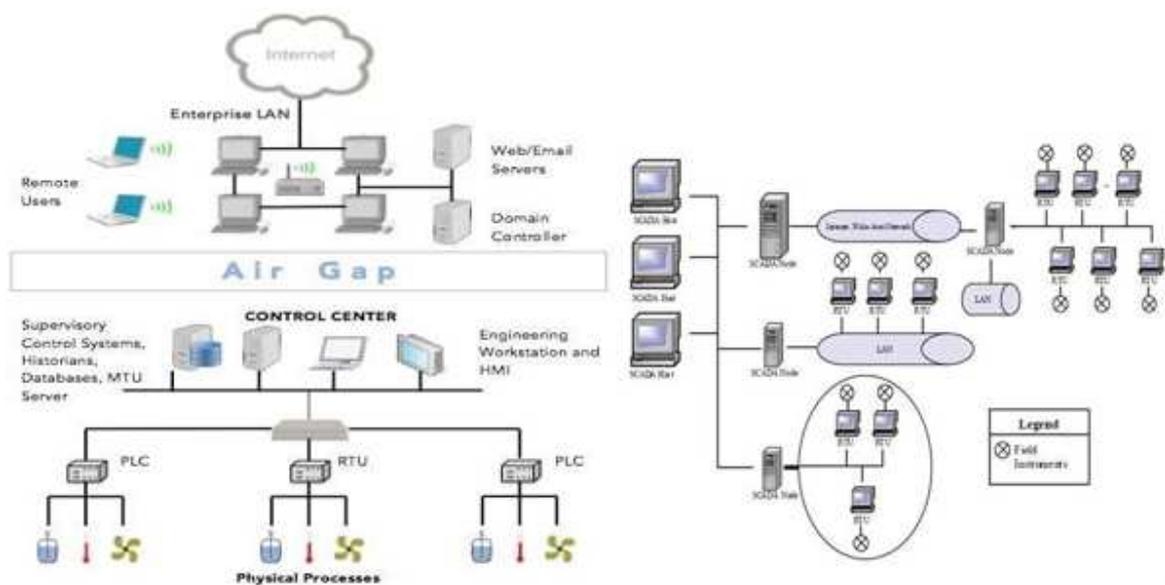


Figure I. 10 : Architectures SCADA. [7]

Le schéma fonctionnel du système SCADA présenté dans la Figure 0.10 représente l'architecture SCADA de base. Les systèmes SCADA sont différents des systèmes de contrôle distribués que l'on trouve couramment sur les sites de production. Lorsque les systèmes de contrôle distribués couvrent le site de l'usine, le système SCADA couvre des zones géographiques beaucoup plus vastes.

En général, une architecture SCADA intégrée prend en charge les protocoles de communication basés sur IP, TCP / IP, UDP et autres, ainsi que des protocoles industriels tels que Modbus TCP, Modbus sur TCP ou Modbus sur UDP<sup>6</sup>. Celles-ci fonctionnent toutes sur des réseaux cellulaires, radios privés ou par satellite.

Dans les architectures SCADA complexes, une variété de supports et de protocoles filaires et sans fil impliqués dans la restitution des données sur le site de surveillance. Cela permet la mise en œuvre de puissants réseaux SCADA basés sur IP sur ligne fixe, des systèmes cellulaires et à satellites mixtes. Les communications SCADA peuvent utiliser un large éventail de supports filaires et sans fil.

Le choix de la communication existante dépend de la caractérisation d'un certain nombre de facteurs à savoir l'éloignement, les communications disponibles sur les sites distants, l'infrastructure de communication existante, la fréquence de scrutation et les débits de données. Ces facteurs ont un impact sur la décision finale concernant l'architecture SCADA. Par conséquent, un examen de l'évolution des systèmes SCADA nous permet de mieux comprendre de nombreux problèmes de sécurité.[7]

### **I.3.5 Protocoles de communication d'un système SCADA**

La complexité de mettre en place un système SCADA se présente en grande partie dans le choix des protocoles de communication entre les différents dispositifs de l'installation. Dans la partie ci-dessous, les protocoles les plus connus sont :

#### **a) Modbus**

Modbus est un protocole de communication série développé par Modicon. C'est une méthode utilisée pour transmettre des informations en série (une après l'autre) entre les dispositifs électroniques ou des appareils d'instrumentation.

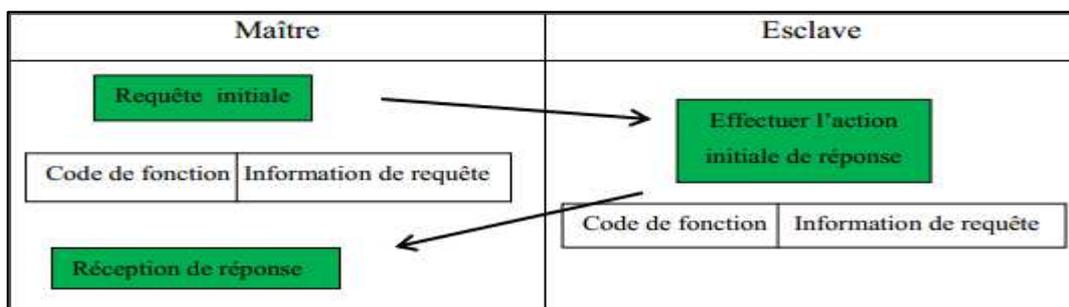
Le dispositif demandant les informations est appelé le « Maître », alors que, le dispositif qui fournit les informations est « l'esclave ».

Dans un réseau Modbus standard, un maître a jusqu'à 247 esclaves où chaque esclave a une adresse unique de 1 à 247. Le maître peut aussi écrire des informations sur les esclaves.[7]

**b) Modbus RTU**

Modbus RTU est un protocole ouvert qui transmet les données en série utilisant des câbles (RS-232 ou RS-485) dérivés de l'architecture maître/esclave. C'est un protocole largement accepté en raison de sa facilité d'utilisation. Ce protocole est utilisé dans les systèmes de gestion de bâtiment (BMS), ainsi que dans les systèmes automatiques industriels.

*Tableau I. 1 : Type de communication entre le maître et l'esclave. [9]*



Le message de Modbus RTU se compose d'une architecture 16-bits. La simplicité de cette architecture est d'assurer la fiabilité de transmission des messages.

**c) Modbus ASCII**

C'est un autre type de mode de transmission Modbus, qui utilise le même support physique de communication de mode RTU.[7]

**d) Modbus TCP/IP**

Transmission Control Protocol (TCP) et Internet Protocol (IP) sont des protocoles fiables qui utilisent la couche de transport du modèle OSI<sup>7</sup>. Ces protocoles sont utilisés conjointement.

Lorsque des informations sur Modbus sont envoyées en utilisant ces protocoles, les données sont transmises à TCP puis envoyées dans une adresse IP. Ensuite, IP place les données dans un paquet (ou datagrammes) et le transmet dans un dispositif de réception.

Ce protocole fonctionne avec le mode Client / Serveur. Les clients sont tous actifs, le serveur est complètement passif. Chaque client a le droit de lire et écrire dans le serveur.[7]

**e) Modbus Plus**

Le Modbus Plus est un protocole pair à pair (chaque client est aussi un serveur) qui transmet des données à une vitesse de 1Mb/s.

Ce protocole spécifie la couche logicielle ainsi que la couche matérielle. Les conducteurs et les raccordements doivent être utilisés avec un réseau Modbus Plus. Généralement, ce protocole n'est pas utilisé pour une communication à temps réel, car il utilise un câble. Cependant, il présente d'autres difficultés liées à la communication.[7]

#### **f) Profibus**

Profibus signifie « bus des processus de terrain ». C'est une norme de réseau en général, utilisé au sein des systèmes de contrôle industriels. Plusieurs systèmes peuvent être utilisés avec la norme Profibus, tels que : le système SCADA, le système numérique de contrôle et de commande, système de contrôle distribué (DCS), l'automate API, l'assemblage et la manipulation des appareils de terrain, le système de contrôle industriel (ICS).

Ce protocole assure la communication entre les dispositifs de champs ou entre MTU et RTU. Profibus exige quelques spécifications d'utilisation. En effet, il faut utiliser un type de connecteur D. Aussi, la distance soutenue peut aller jusqu'à 24 km au maximum.

Enfin, le débit d'accélération de transmission des données peut aller jusqu'à 12 Mb/s et la taille du message jusqu'à 244 octets par nœud.

La famille Profibus se compose de trois versions compatibles offrant de très grandes intégrités et une capacité adaptée au besoin.

- PROFIBUS DP - Périphérie décentralisée
- PROFIBUS FMS - Fieldbus Message Spécification
- PROFIBUS PA - Process Automation [9]

#### **I.3.5.1 Supervision dans un environnement SCADA**

Le système SCADA fonctionne par l'acquisition de données provenant de l'installation, ces dernières sont affichées sur une interface graphique sous un langage très proche de langage humain, ces opérations sont exécutées en temps réel, ainsi les systèmes SCADA donnent aux opérateurs le maximum d'information pour une meilleure décision, ils permettent un très haut niveau de sécurité, pour le personnel et pour l'installation.[7]

Il permet aussi la réduction des coûts des opérations ; les avantages qu'offre le SCADA sont obtenus avec la combinaison des outils softs et hard.

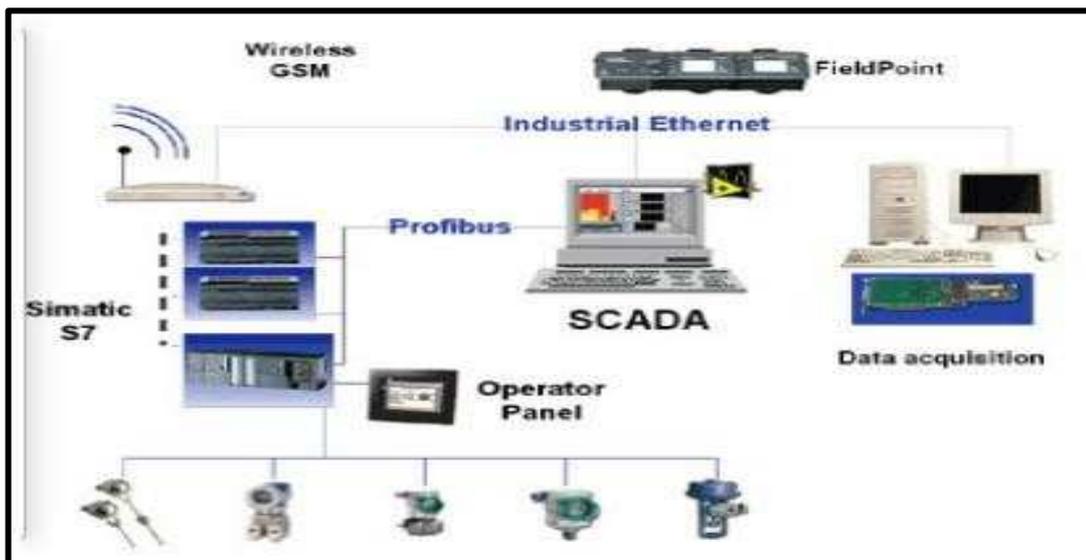


Figure I. 11 : Combinaison des outils softs et hard des systèmes SCADA. [7]

**Interfaces graphiques du SCADA**

Les interfaces graphiques sont un outil très important pour le bon déroulement de la procédure d'aide à la décision, elles sont le seul point d'interaction entre l'opérateur et les algorithmes d'aide à la décision, ainsi, elles aident l'opérateur dans sa tâche d'interprétation et de prise de décision, en lui offrant une très bonne visibilité sur l'état et l'évolution de l'installation, avec l'affichage en différentes couleurs des résidus, des alarmes et des propositions sur l'action à entreprendre [7].

La figure suivante représente une interface graphique de la supervision d'un générateur de vapeur :

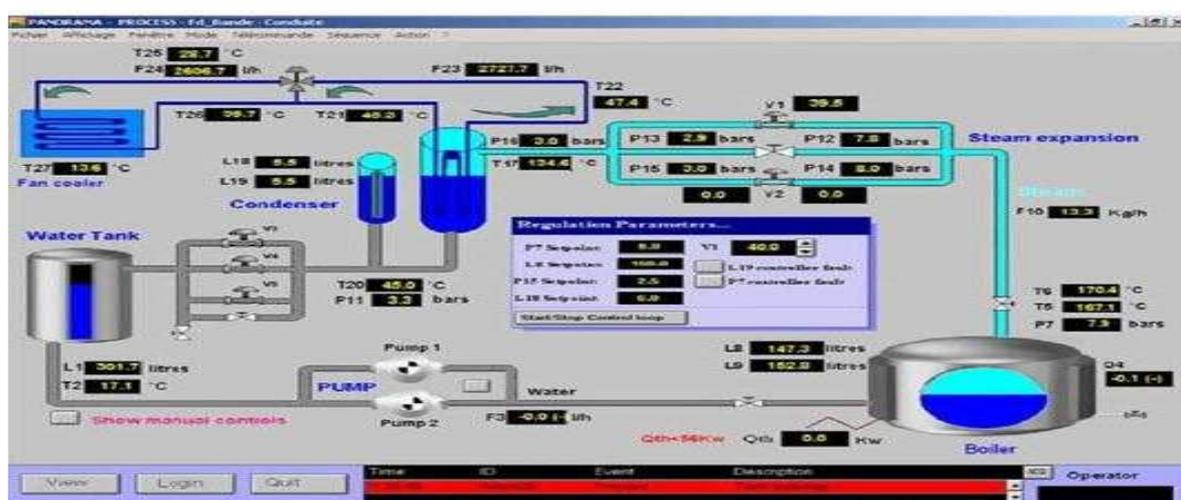


Figure I. 12 : Interface graphique de supervision d'un générateur de vapeur. [7]

### **Fonctionnement en temps réel**

La notion du temps réel est devenue très importante et indispensable dans la procédure de surveillance et de supervision en générale, elle permet de faire le rafraîchissement des signaux à chaque instant, ce qui permet de suivre l'évolution de l'état du système d'une façon continue [7].

### **I.3.6 Avantages du SCADA**

Parmi les avantages du SCADA, on retrouve :

- Le suivi de près du système ; voir l'état du fonctionnement de procédé dans des écrans même s'il se situe dans une zone lointaine.
- Le contrôle et l'assurance que toutes les performances désirées sont atteintes de visualiser les performances désirées du système à chaque instant, et s'il y aurait une perte de performance, une alarme se déclenche d'une manière automatique pour prévenir l'opérateur.
- La génération d'une alarme lorsqu'une faute se produit et visualise même la position où se situent la faute et l'élément défectueux, ce qui facilite la tâche du diagnostic et de l'intervention de l'opérateur.
- La disponibilité de plusieurs informations sur le système aide l'opérateur à prendre la bonne décision, et à ne pas se tromper dans son intervention.
- Réduction des tâches du personnel en les regroupant dans une salle de commande.
- Elimination ou réduction du nombre de visite aux sites éloignés ; avec une interface graphique, on peut suivre l'état de l'installation à chaque instant, ainsi on n'aura pas besoin de faire des visites de contrôle [7].

## **I.4 Evolution des IHM**

Les systèmes de contrôle industriels continuent d'évoluer et, dans le monde d'aujourd'hui, les tâches que les opérateurs doivent accomplir peuvent changer fréquemment. Pour gérer cette complexité, vos contrôles doivent être flexibles et ergonomiques. C'est l'avantage de l'IHM. Avec une IHM basée sur zenon, vous pouvez facilement communiquer avec les machines et obtenir des données opérationnelles sur l'ensemble de vos équipements et de vos installations.[12]

### **I.4.1 Définition d'une IHM**

IHM signifie interface homme-machine et fait référence à un tableau de bord qui permet à un utilisateur de communiquer avec une machine, un programme informatique ou un système. Techniquement, vous pourriez appliquer le terme IHM à n'importe quel écran utilisé pour

interagir avec un appareil, mais il est généralement employé pour décrire des écrans utilisés dans les environnements industriels. Les IHM affichent des données en temps réel et permettent à l'utilisateur de contrôler les machines grâce à une interface utilisateur graphique.

Dans un environnement industriel, une IHM peut prendre plusieurs formes. Il peut s'agir d'un écran autonome, d'un tableau de bord attaché à un autre équipement ou d'une tablette. Quoi qu'il en soit, son but premier est de permettre aux utilisateurs de visualiser des données quant aux opérations et de contrôler les machines. Les opérateurs pourraient, par exemple, utiliser une IHM pour voir quelles bandes transporteuses sont enclenchées ou pour ajuster la température d'un réservoir d'eau industriel.

L'IHM de pointe vous permet de communiquer avec les machines et de visualiser les données de vos équipements.[12]

### **I.4.2 Utilisation d'une IHM**

Une IHM est utilisée pour un large éventail d'industries. Elle est courante dans la fabrication de divers types de biens, de l'automobile aux aliments et boissons en passant par les produits pharmaceutiques. Les industries telles que celles de l'énergie, de l'eau, des eaux usées, du bâtiment et du transport peuvent également avoir recours à une IHM. Les IHM sont fréquemment utilisées par les intégrateurs systèmes, les opérateurs et les ingénieurs, surtout les ingénieurs de systèmes de contrôle des processus. Ces professionnels peuvent utiliser des IHM pour contrôler des machines, des véhicules, des usines ou des bâtiments.

La sophistication de l'IHM varie selon la complexité de la machine ou du système pour lequel elle est utilisée. Elle varie également selon la façon dont vous prévoyez utiliser l'IHM. Vous pouvez utiliser une IHM pour une seule fonction, par exemple la surveillance d'une machine, ou à des fins multiples, y compris la surveillance des opérations de l'usine et le contrôle des équipements.

Les opérateurs peuvent également utiliser des IHM pour contrôler les équipements et augmenter ainsi la productivité ou s'adapter aux circonstances changeantes. Ils peuvent faire des ajustements en fonction des données qu'ils voient sur l'IHM. La possibilité d'effectuer les changements requis sur l'écran rend ce processus plus rapide et plus simple. [12]

### **I.4.3 Intégration d'une IHM**

Les solutions IHM peuvent se présenter sous différentes formes, du terminal autonome à la tablette en passant par le tableau de bord attaché à une autre machine. Une installation peut avoir une unique IHM centrale ou plusieurs IHM réparties qui peuvent toutes être connectées par Internet.[12]

#### I.4.4 Avantages d'une IHM

Les IHM offrent un certain nombre d'avantages aux organisations industrielles actuelles, y compris :

- **Amélioration de la visibilité** : une IHM haute performance vous offre une meilleure visibilité de vos opérations, à tout moment. Cela vous permet de visionner la performance de votre équipement ou de votre installation sur un seul tableau de bord. Vous pouvez même visualiser ce tableau de bord à distance. Ces fonctionnalités vous aident à améliorer peu à peu votre productivité et à répondre plus rapidement aux alertes.
- **Augmentation de l'efficacité** : étant donné qu'une IHM fournit un accès constant aux données en temps réel, vous pouvez l'utiliser pour surveiller la production et vous adapter en temps réel à l'évolution de la demande. La visualisation des données, surtout lorsqu'elle est combinée avec les technologies d'analyse des données, peut vous aider à identifier les domaines dans lesquels vous pouvez améliorer l'efficacité de vos procédures.
- **Diminution des temps d'arrêt** : grâce aux alertes sur le tableau de bord central, vous pouvez réagir plus rapidement aux problèmes, ce qui réduit les temps d'arrêt. L'affichage et l'analyse des données de performance des équipements peuvent également vous aider à identifier les signes de problèmes mécaniques à venir et à les résoudre avant qu'ils ne dégénèrent et ne deviennent des problèmes capables d'entraîner des temps d'arrêt importants.
- **Amélioration de l'ergonomie** : les IHM facilitent la visualisation et la compréhension des données et des équipements de contrôle pour les utilisateurs. Elles présentent les données à l'aide de graphiques, de tableaux et d'autres visualisations, ce qui permet aux utilisateurs de les interpréter rapidement. Avec zenon, les utilisateurs peuvent également personnaliser leurs tableaux de bord en fonction de leurs besoins et de leurs préférences.
- **Unification du système** : avec zenon, vous pouvez contrôler tous les équipements à l'aide d'une seule plate-forme, ce qui permet aux opérateurs d'apprendre plus facilement comment contrôler les équipements. Vous pouvez également visualiser toutes vos données à un seul endroit, ce qui vous aide à obtenir un aperçu clair de l'ensemble de votre installation. De plus, tous les utilisateurs reçoivent des mises à jour en temps réel, donc votre équipe est toujours sur la même longueur d'onde. [12]

### I.4.5 Ergonomie des interfaces

L'ensemble du travail de conception des interfaces « homme machine » doit prendre en considération l'élément essentiel : L'ergonomie.

Cette notion d'ergonomie intègre les éléments qui vont permettre à l'utilisateur de s'approprier l'usage de l'application au travers de son interface, c'est-à-dire :

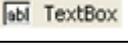
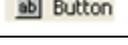
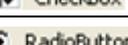
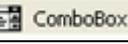
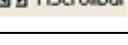
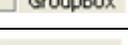
- **L'homogénéité des différents formulaires**, composant l'interface (les différentes composantes du logiciel doivent avoir une interface similaire permettant d'adaptation rapide à leur utilisation).
- **L'adaptabilité de l'interface** aux besoins fonctionnels de l'utilisateur (possibilité de paramétrage de l'environnement de l'utilisateur dans le cadre des fonctionnalités de son travail),
- **La convivialité d'usage** (l'interface doit permettre à l'utilisateur de travailler de façon simple, confortable et efficace).
- **la prise en main facile** (utilisateur doit pouvoir apprendre rapidement le fonctionnement du logiciel et disposer de fonctionnalités pouvant l'aider).

La composition ergonomique des interfaces « homme machine » repose sur un certain nombre de normes telles que la **norme ISO 9241**. [9]

### I.4.6 Composants techniques des interfaces graphiques

Les formulaires des interfaces graphiques sont structurés autour de composants graphiques appelés aussi **contrôles graphiques**.

Tableau I. 2 : Quelques composants ou contrôles graphiques essentiels.

Symbole	Désignation	Description
 Label	Etiquette	Texte constant d'informations affichées
 TextBox	Zone de texte	Zone permettant de saisir (acquisition de données) ou d'afficher des informations (restitution de résultats)
 Button	Bouton	Bouton de commande permettant l'activation de traitements principalement sur clic ou double clic
 CheckBox	Case à cocher	Permet à l'utilisation d'effectuer des choix
 RadioButton	Bouton radio	Permet à l'utilisation d'effectuer un choix
 ComboBox	Liste déroulante	Permet à l'utilisation d'effectuer un choix parmi une liste proposée
 ListBox	Zone de liste	Permet à l'utilisation d'effectuer un ou des choix (en fonction de la volonté du développeur) parmi une liste proposée
 HScrollBar	Barre de défilement	Barre de déroulement (horizontale ou verticale) permettant à l'utilisateur de se déplacer dans un document plus grand que la zone d'affichage
 GroupBox	Groupe d'options	Permet d'effectuer le regroupement de composants ou contrôles graphiques tels que les boutons radios ou des cases à cocher
 LinkLabel	Lien hypertexte	Permet une navigation hypertexte de document en document (ou de formulaire en formulaire)

Il existe de très nombreux autres composants ou contrôles graphiques pouvant être intégrés aux formulaires, qui sont fournis par les logiciels ou langages de développement d'applications. Chaque composant ou contrôle graphique (si le développeur le souhaite) peut gérer un certain nombre d'événements, qui permettront **la gestion dynamique du formulaire** (l'activation de l'événement entraîne la mise en oeuvre d'un programme).[13]

#### I.4.7 Différence entre IHM et SCADA

Les IHM sont proches des systèmes SCADA (Système de Contrôle et d'Acquisition de Données), et son souvent utilisé simultanément au sein d'un système de contrôle industriel. Cependant, ils présentent plusieurs différences.

Les IHM permettent principalement d'afficher des informations de façon visuelle pour permettre à l'utilisateur de superviser un processus industriel. Les systèmes SCADA, quant à eux, ont une capacité supérieure en termes **d'acquisition de données et d'utilisation des systèmes de contrôle**.

Contrairement aux systèmes SCADA, les IHM **ne permettent pas de collecter ou d'enregistrer d'informations**. Ils ne peuvent pas non plus se connecter à des bases de données. Il s'agit d'outils de communication fonctionnant comme une partie (ou en complément) d'un système SCADA.[14]

#### I.4.8 Logiciels existant pour la création d'une IHM

##### a) Zenon



Figure I. 13 : Logo logiciel zenon. [15]

zenon est un logiciel facile à utiliser et puissant pour l'automatisme industriel. Que vous cherchiez une interface homme machine, un outil de visualisation des procédés, un système de contrôle (SCADA), ou une interface pour la collecte de données de production.[16]

##### b) Editeur movicon



Figure I. 14 : Editeur movicon. [17]

Movicon sont des logiciels d'IHM et de supervision pour les applications d'informatique industrielle permettant à l'utilisateur de réaliser ses propres applications graphiques

### Caractéristiques

- Support de Java™
- Mise à jour vers les dernières versions du runtime
- Mise à jour du nombre de variables et de Tag
- Extension pour intégrer les dernières versions du runtime
- Serveur WEB
- Extension des capacités d'archivage et de courbes.[18]

#### c) **MySCADA**



*Figure I. 15 : Logiciel myscada. [19]*

mySCADA est un système IHM/SCADA professionnel conçu principalement pour la visualisation et le contrôle des processus industriels.[20]

### **Caractéristiques**

- Fonction: de gestion de contrôle d'accès et de sécurité, de gestion de documents, de maintenance préventive, de gestion d'énergie, HMI, de gestion des opérations, de gestion de processus de simulation, de gestion de consommation d'énergie, de gestion de données basé sur le web, statistique, de supervision, d'accès à distance, de SCADA, OPC, d'interface, de commande, MES
- Applications de process d'injection plastique, industriel, pour machine-outil, pour l'agroalimentaire, pour l'industrie automobile, pour l'application de traitement d'eaux, de connexion de serveurs OPC, pour réseau d'eau.[16]

#### d) **WinCC**



*Figure I. 16 : Logiciel WinCC. [21]*

Le WinCC (Windows Control Center) est un système IHM (Interface Homme Machine) autrement dit l'interface entre l'homme (l'opérateur) et la machine (le processus) est un logiciel de supervision développé par SIEMENS. Il permet à l'opérateur de visualiser et de surveiller le processus par un graphisme à l'écran. WinCC constitue la solution de conduite et de supervision de procédés sur un pupitre.

- Il offre une bonne solution de supervision en raison des fonctionnalités adaptées aux exigences courantes des installations industrielles qu'il met à la disposition des opérateurs.
- Il est caractérisé par sa flexibilité, c'est-à-dire qu'il peut être utilisé par un composant hors SIEMENS.
- Il offre une bonne solution de supervision, car il met à la disposition de l'opérateur des fonctionnalités adaptées aux exigences d'une installation industrielle.

Avec WinCC, il est possible de superviser une ou plusieurs machines situées à proximité ou non du poste de visualisation. Lorsque l'on lance TIA Portal, l'environnement de travail se décompose en deux types de vue :

- Vue du portail : elle est axée sur les tâches à exécuter.
- Vue du projet : comporte une arborescence avec les différents éléments du projet.

#### a) Vue du portail



Figure I. 17 : Vue principale de TIA Portal. [21]

Chaque portail permet de traiter une catégorie de tâche (actions). La fenêtre affiche la liste des actions pouvant être réalisées pour la tâche sélectionnée.

### b) Vue du projet

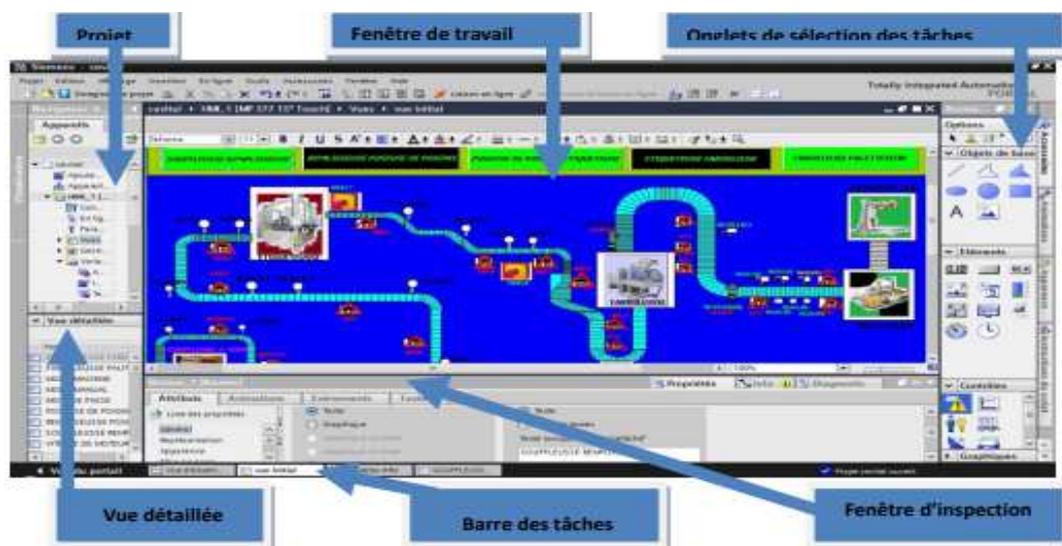


Figure I. 18 : Représentation des différentes zones trouvant dans la vue principale de TIA Portal. [21]

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

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 HMI, 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.

## I.5 Conclusion

Dans ce chapitre nous avons essayé de décrire toutes les parties qui forme un système automatisé en commençant par la partie matérielle jusqu'à la partie WinCC pour la création d'IHM. Pour enrichir ce chapitre, on a jugé utile d'intégrer la description du système étudié

Dans ce chapitre, nous avons présenté au premier lieu le système automatisé, son

historique, sa décomposition, ses avantages et ces inconvénients. Comme deuxième phase, nous avons décrit le système SCADA qui est un système de supervision industrielle qui traite en temps réel un grand nombre de mesures et contrôle à distance les installations. On a aussi décrit en détail l'interface homme machine et une présentation des automates programmables industriels a été ajoutée.

En se basant sur ces informations, nous allons présenter dans le chapitre suivant une description détaillée sur le matériel étudié et une généralité sur les automates programmable.

# Chapitre 02

## Description du Système Etudié

## **II.1 Introduction**

Pour ce deuxième chapitre, nous allons le consacrer pour la présentation de notre usine ainsi notre système étudié, Dans la première partie puis nous le consacrons a la présentation des généralités l'usine et le mode de fonctionnement de la chaine de production, ensuite dans la deuxième partie nous exposons des généralités sur les automates programmable.

## **II.2 Généralités sur l'usine et le mode de fonctionnement de la chaine de production**

Le laboratoire de Productique Tlemcen MELT (Manufacturing Engineering Laboratory Of Tlemcen) est situé à l'université Abou Bekr Belkaid Tlemcen–Chetouane-Algérie. Ce laboratoire est composé par un ensemble des systèmes de FESTO didactique qui est une entreprise familiale allemande fondée en 1925, elle est le premier prestataire mondial pour la formation dans l'automation industrielle.

Il comporte aussi une mini-chaine de production de marque EletroVeneta qu'on va contribuer durant ce PFE à le faire fonctionner.

### **II.2.1 Présentation et objectif du laboratoire**

Le Génie Industriel, est peu abordé au sein des universités, malgré qu'il a une importance capitale pour le développement de l'industrie. Cependant, il est clair que l'industrie ne peut continuer à fonctionner sans une optimisation de ses ressources et une automatisation de ses unités de production. Ainsi, l'activité de production est assujettie au changement constant des technologies et des connaissances humaines.

#### **Objectifs de ce laboratoire**

Regroupement des compétences des chercheurs venant de spécialités différentes (automatique, électronique, électrotechnique, informatique, physique...) tous formés à la productique, pour permettre le renforcement des connaissances afin de développer et valider les modèles mathématiques capables de traduire le comportement réel des systèmes industrielles.

Le développement des modèles, méthodes et techniques qui permettent l'amélioration et le développement des systèmes de production et de services par l'optimisation des ressources, la maximisation des profits et la minimisation des pertes.

Application des résultats obtenus (modélisation - optimisation - ordonnancement - pilotage...) à des cas concrets du tissu industriel et du service local, régional et national.

## II.2.2 Description du système étudié

Dans ce projet, nous allons faire une étude approfondie sur un complément d'un système qui inclura une pince et un convoyeur afin de l'intégrer à un ancien système constitué d'une table tournante, d'un convoyeur et d'une ventouse rotative.

### II.2.2.1 Module Mod. MCS-700/EV

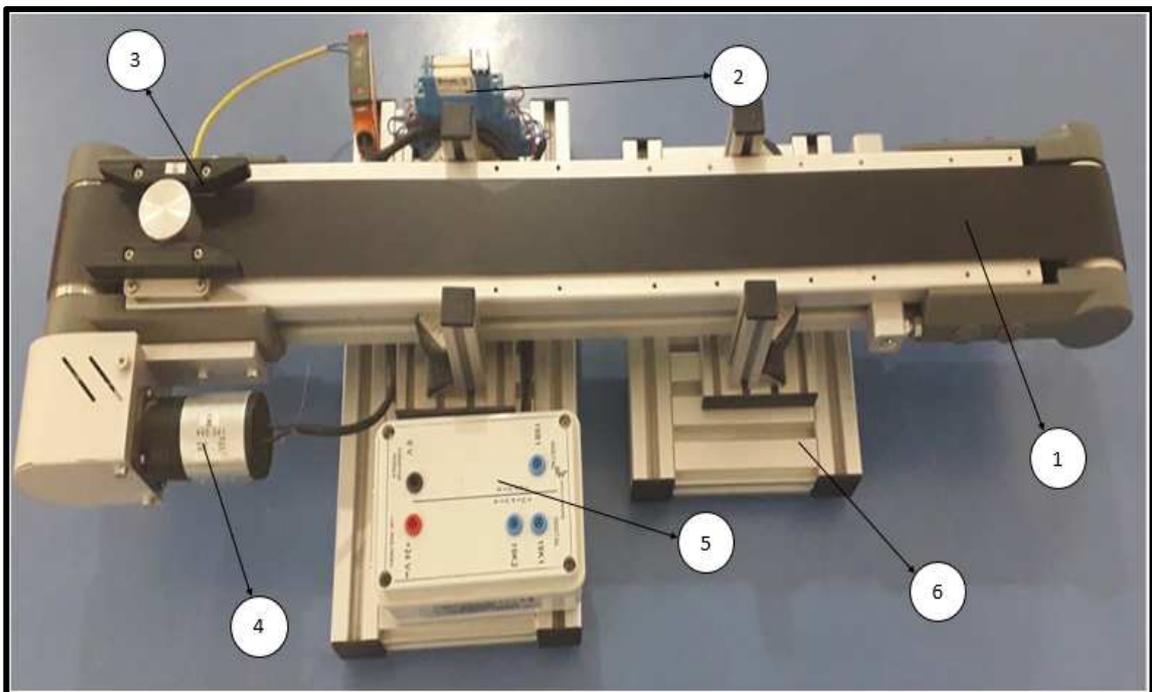


Figure II. 19 : Photo du module convoyeur.

#### a) Description

- Le module mod. MCS-700/EV a été conçu pour le transport de pièces, dans les deux directions, le long d'un axe linéaire.
- La bande transporteuse est mise en mouvement par un moteur CC commandé par un contacteur de marche directe/inverse pour le mouvement de la bande.
- Un capteur à fibre optique permet la détection du passage des pièces sur la bande transporteuse.
- L'ensemble du module est monté sur une structure profilée en aluminium pouvant être facilement transportée à l'intérieur du laboratoire.
- Le contrôle complet du module est effectué par un API relié au système par le biais d'un box d'interface E/S.

### b) Programme de formation

Le programme de formation pouvant être développé avec le module mod. MCS-700/EV comprend les thèmes suivants :

- Principes des contrôles électriques : le moteur en CC
- Opérations avec la bande transporteuse
- Capteur à fibre optique

### c) Composants du module

1. Tapis roulant
2. Contacteur de marche directe/inverse
3. Capteur fibre optique
4. Moteur à courant continu
5. Box d'interface contenant les entrées/sorties du module
6. Profilé en aluminium.

#### II.2.2.2 Module de contrôles des pièces MCS-570/EV

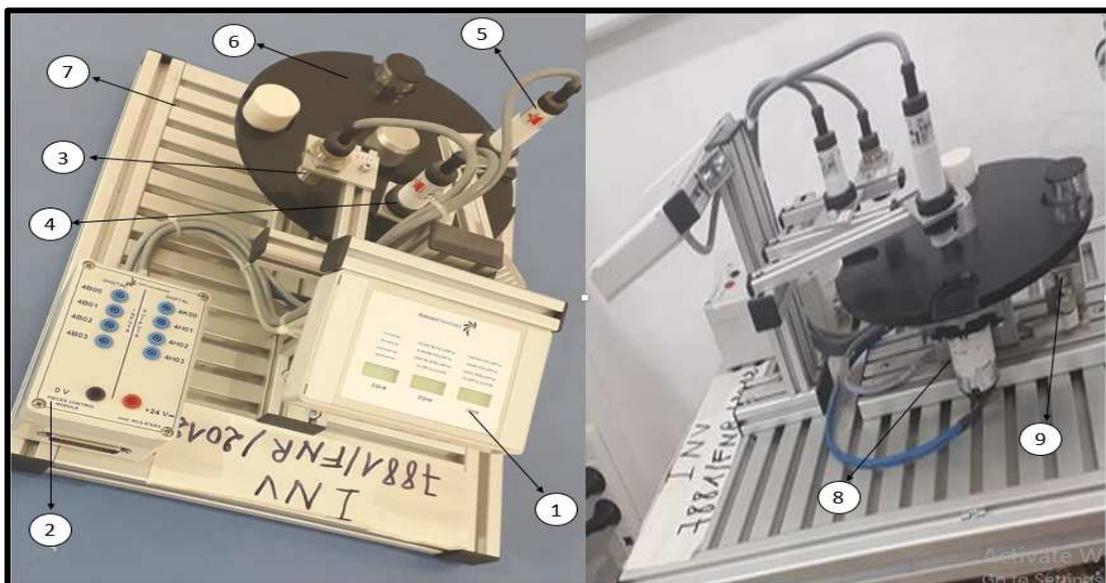


Figure II. 20 : Photo du module Table tournante.

### a) Description

- Le module de contrôle de pièces mod. MCS-570/EV est constitué par une table tournante où sont disposées les pièces à examiner. Un groupe de trois capteurs : capacitif, inductif et optique, servant pour détecter trois types de pièces présentes, noir ou blanc et métallique. Le résultat peut être observé sur l'afficheur.

- La position de la table tournante est contrôlée par un capteur de proximité inductive tandis que la rotation est gérée par un petit moteur à courant continu.
- Les positions libres pour le prélèvement et le dépôt des pièces sont au nombre de deux. Le capteur capacitif est placé dans le point de relâché des pièces par la station précédente et aussi peuvent être changée la position de capteur selon le module qui dépose la pièce sur la table
- L'ensemble du module est monté sur une structure profilée en aluminium pouvant être facilement transportée à l'intérieur du laboratoire.
- Le contrôle complet du module est effectué par un API (en option) relié au système par le biais d'un box d'interface E/S.

#### **b) Programme de formation**

Le programme de formation pouvant être développé avec le Module mod. MCS-570/EV comprend les thèmes suivants :

- Principes d'électropneumatique
- Étude des systèmes d'identification
- Étude des capteurs
- Étude des moteurs en courant continu

#### **c) Composants du module**

1. Afficheur,
2. Box d'interface,
3. Capteur inductif,
4. Capteur optique,
5. Capteur capacitif,
6. Table tournante,
7. Profile en aluminium,
8. Moteur à CC
9. Capteur inductif de proximité.

### II.2.2.3 Module Bras Pneumatique mod. MCS-590/EV

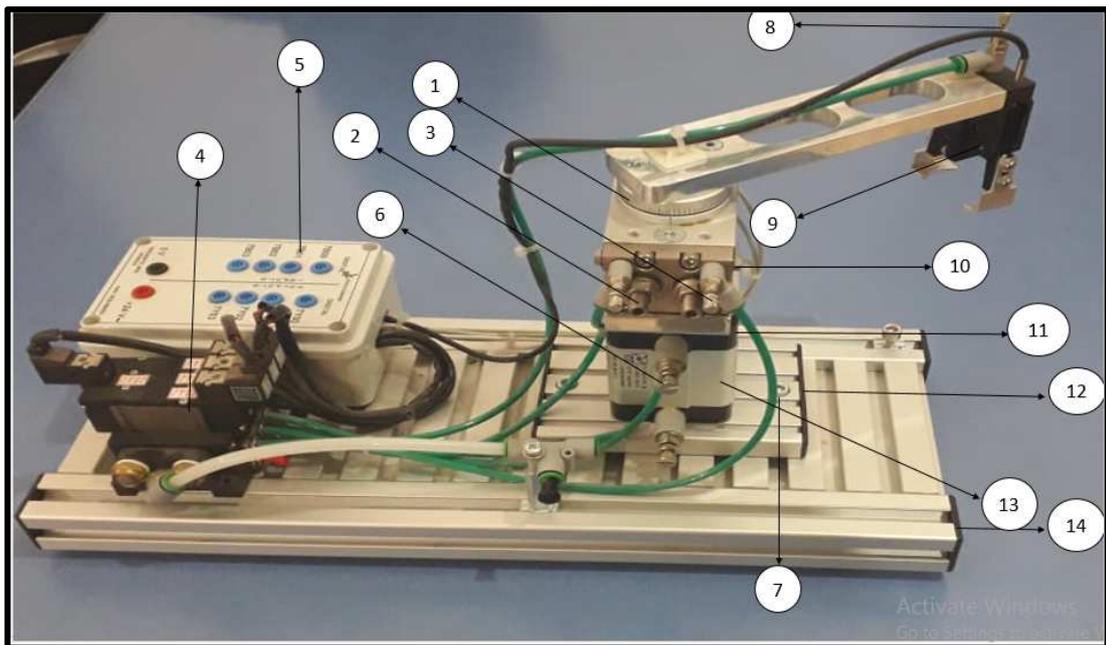


Figure II. 21 : Photo du module Bras pneumatique.

#### a) Description

- Les mouvements des éléments pneumatiques sont contrôlés par des capteurs.
- Le contrôle complet du module mod. MCS-590/EV est effectué par un API (en option) relié au système par le biais d'un box d'interface E/S.
- Grâce au logiciel de programmation de l'API, d'un emploi facile, il est possible de réaliser une vaste gamme d'expériences inhérentes à l'automatisation du module mod. MCS-590/EV, lequel peut être intégré dans le cycle d'usinage ou fonctionner de façon indépendante («stand alone mode»).
- Enfin, il est possible d'effectuer la supervision du module mod. MCS-590/EV en employant le logiciel mod.SV/EV.

#### b) Programme de formation

Le programme de formation pouvant être développé avec le module mod. MCS-590/EV, comprend les thèmes suivants :

- Principes d'électropneumatique
- Fonctionnement des cylindres pneumatiques
- Fonctionnement d'une pince pneumatique
- Fonctionnement des électrovannes à solénoïde
- Fonctionnement des capteurs magnétiques

### c) Composants du module

1. Bras oscillant
2. Cylindre tournant
3. Capteur reed
4. Cylindre de montée / descente
5. Base profilée en aluminium
6. Profilé en aluminium
7. Valve de réglage
8. Groupe valves
9. Groupe valves
10. Box d'interface
11. Valve de réglage du flux
12. Vis de réglage
13. Capteur pince
14. Angle de mouvement de la pince.

#### II.2.2.4 Module de Commande



Figure II. 22 : Photo du module de commande mod. PLC-V7/EV. [23]

Au sein d'un laboratoire pour les technologies de l'automatisation, le banc mod. PLC-V7/EV représente l'instrument nécessaire à la formation de très haut niveau de techniciens employés dans l'industrie moderne des processus. Réalisé totalement avec des composants

industriels, le banc mod. PLC-V7/EV permet de développer une excellente expérimentation, comportant un contenu de connaissances très élevé, sur la programmation des automates programmables industriels (API) et fournir une solution aux problèmes les plus complexes liés à l'automatisation. Dans ce contexte, le programme de formation traite un vaste spectre d'applications :

- Installations industrielles
- Robotique
- Automatisation avec transporteur
- Contrôles de processus avec techniques PID

**a) Applications typiques**

- Modules mécatroniques

**b) Régulations PID**

- Contrôles de processus (niveau, débit, pression, température)

**c) Communication entre API et ordinateur**

- Diagnostic des entrées et sorties
- Diagnostic des états internes
- Forçage E/S et variables internes
- Réseaux d'API (Maître/Esclave)

## **II.3 Généralités sur les automates programmables**

### **II.3.1 Historique**

Les Automates Programmables Industriels (API) sont apparus aux Etats-Unis vers la fin des années soixante, à la demande de l'industrie automobile américaine (General Motors) qui réclamait plus d'adaptabilité de leurs systèmes de commande.

Les ingénieurs américains ont résolu le problème en créant un nouveau type de produit nommée automates programmables. Ils n'étaient rentables que pour des installations d'une certaine complexité, mais la situation a très vite changé, ce qui a rendu les systèmes câblés obsolètes. De nombreux modèles d'automates sont aujourd'hui disponibles ; depuis les nano automates bien adaptés aux machines et aux installations simples avec un petit nombre d'entrées/sorties, jusqu'aux automates multifonctions capables de gérer plusieurs milliers d'entrées/sorties et destinés au pilotage de processus complexes. [24]

### **II.3.2 Définition de l'automate programmable**

Un automate programmable industriel (API) est une machine électronique spécialisée dans la conduite et la surveillance en temps réel de processus industriels et tertiaires. Il exécute une suite d'instructions introduites dans ses mémoires sous forme de programmes, et s'apparente par conséquent aux machines de traitement de l'information. Trois caractéristiques fondamentales le distinguent des outils informatiques tels que les ordinateurs utilisés dans les entreprises et le tertiaire :

- Il peut être directement connecté aux capteurs et pré-actionneurs grâce à ses entrées/sorties industrielles.
- Il est conçu pour fonctionner dans des ambiances industrielles sévères (Température, vibrations, microcoupures de la tension d'alimentation, parasites, etc...).
- Enfin, sa programmation à partir de langages spécialement développés pour le traitement de fonctions d'automatisme facilite son exploitation et sa mise en œuvre. [24]

### **II.3.3 Choix d'un automate : quelques critères à considérer**

#### **a) Marque**

Tenir compte de la marque est un critère à prendre en compte lorsqu'on est déjà familiarisé avec un API d'une certaine marque. En effet, un automate d'une autre marque sera peut-être plus long à maîtriser car quelque peu différent (ex : les automates de la marque Rockwell sont très répandus aux États-Unis). [25]

#### **b) Temps de cycle**

Selon les industries et les besoins, les temps d'opération peuvent être plus rapides que d'autres. Il est donc nécessaire de prêter attention au temps de cycle d'un API si l'on souhaite une bonne réactivité de la part de celui-ci. [25]

#### **c) Programmation**

Si des programmes spécifiques ont été appliqués à un API bien particulier, il convient de se tourner vers ce type d'automate à l'avenir. Les futures programmations d'automates n'en seront que facilitées. [25]

#### **d) Accessibilité**

Pour faire à nouveau référence à la marque des automates, il est utile de préciser que certaines ont plus de poids que d'autres sur le marché de l'API et proposent donc un accompagnement optimal du client (ex : service après-vente, pièces de rechange...) [21]

#### **e) Réputation**

C'est un détail mais qui pourtant pèse dans la balance : la renommée. Certains autoates sont réputés pour être de meilleure qualité que d'autres. Au client de se faire son propre avis.[25]

#### **f) Prix**

Même si l'on s'accorde à dire qu'un automate industriel constitue un bon investissement à long terme, il convient de garder en tête un prix juste et raisonnable pour le type d'API recherché. S'il s'agit d'un automate pour un petit projet, peut-être est-ce plus raisonnable d'envisager un « micro automate » qu'une artillerie lourde. [25]

#### **g) Communication**

Enfin et pas des moindres, il est utile de se renseigner sur les interfaces de communication et les bus de terrain natifs de l'API. Ce qui évite accessoirement d'avoir à racheter d'autres modules par la suite. [25]

### **II.3.4 Structure générale des API**

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

- Coffret,
- Rack,
- Baie ou cartes
- Compact ou modulaire
- Tension d'alimentation
- Taille mémoire
- Sauvegarde (EPROM, EEPROM, pile, ...)
- Nombre d'entrées / sorties
- Modules complémentaires (analogique, communication, ...)
- Langage de programmation

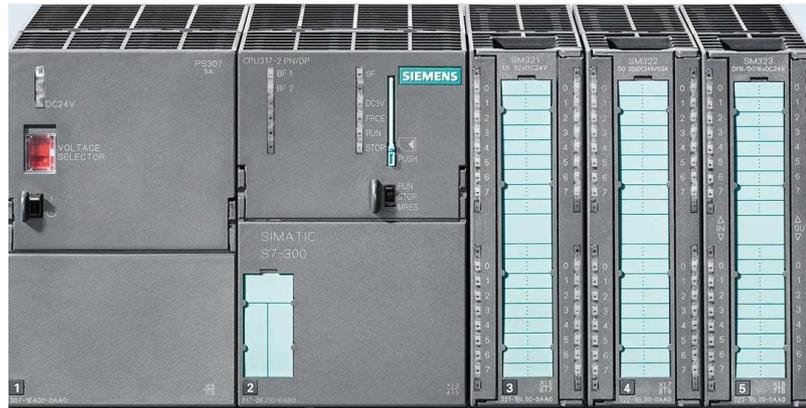


Figure II. 23 : Aspect extérieur d'un automate S7-300 CPU 314 2PN/DP.

Des API en boîtier étanche sont utilisées pour les ambiances difficiles (température, poussière, risque de projection ...) supportant ainsi une large gamme de température, humidité

L'environnement industriel se présente sous trois formes :

- Environnement physique et mécanique (poussières, température, humidité, vibrations) ;
- Pollution chimique ;
- Perturbation électrique (parasites électromagnétiques)

### II.3.5 Structure interne d'un automate programmable industriel (API)

Les API comportent quatre principales parties (Figure II.6) :

- Une unité de traitement (un processeur CPU)
- Une mémoire
- Des modules d'entrées-sorties
- Des interfaces d'entrées-sorties
- Une alimentation 230 V, 50/60 Hz (AC) - 24 V (DC) [3]

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

- La mémoire Programme où est stocké le langage de programmation. Elle est en général figée, c'est à dire en lecture seulement. (ROM : mémoire morte)

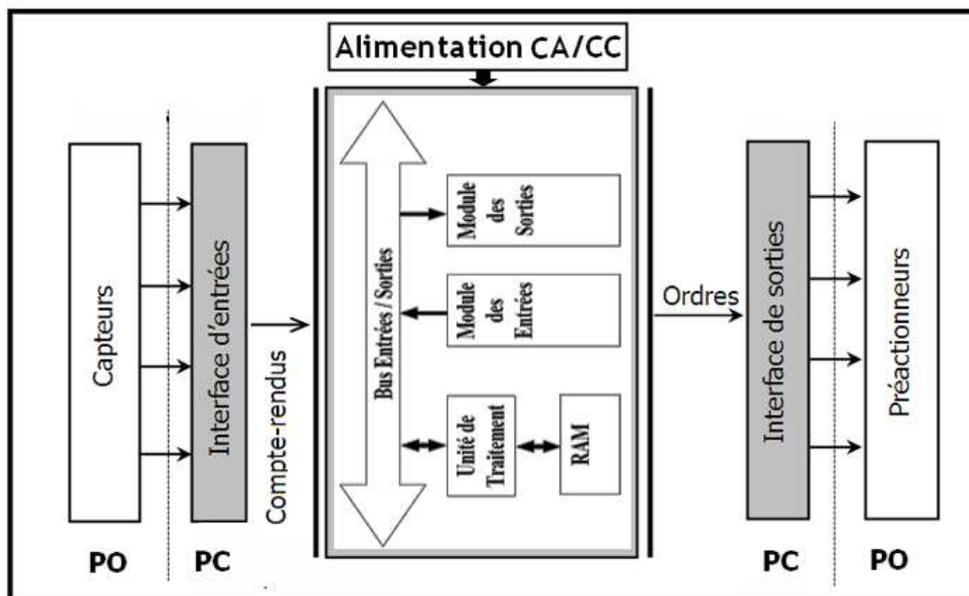


Figure II. 24 : Structure interne d'un automate programmable industriel (API). [3]

La mémoire de données utilisable en lecture-écriture pendant le fonctionnement c'est la RAM (mémoire vive). Elle fait partie du système entrées-sorties. Elle fige les valeurs (0 ou 1) présentes sur les lignes d'entrées, à chaque prise en compte cyclique de celle-ci, elle mémorise les valeurs calculées à placer sur les sorties [3]

### II.3.5.1 Fonctionnement

L'automate programmable reçoit les informations relatives à l'état du système et puis commande les pré-actionneurs suivant le programme inscrit dans sa mémoire. Généralement les automates programmables industriels ont un fonctionnement cyclique (Figure II.7). [3]

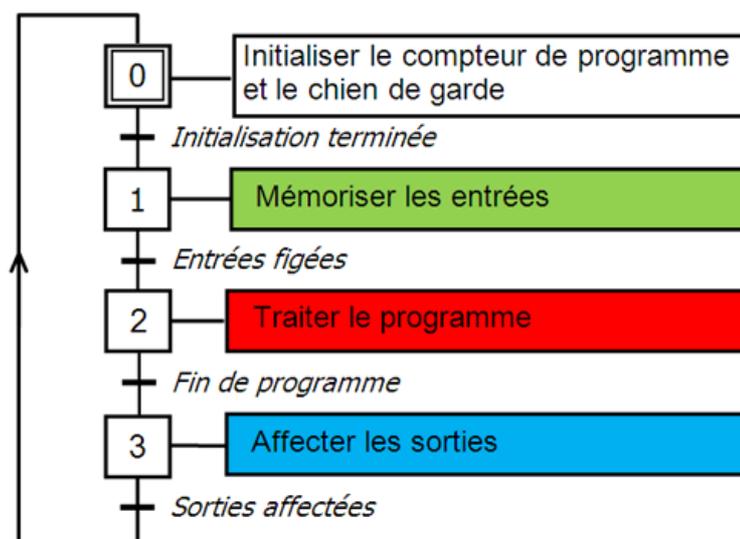


Figure II. 25 : Fonctionnement cyclique d'un API. [3]

Le microprocesseur réalise toutes les fonctions logiques ET, OU, les fonctions de temporisation, de comptage, de calcul... Il est connecté aux autres éléments (mémoire et interface E/S) par des liaisons parallèles appelées ' BUS ' qui véhiculent les informations sous forme binaire, Lorsque le fonctionnement est dit synchrone par rapport aux entrées et aux sorties, le cycle de traitement commence par la prise en compte des entrées qui sont figées en mémoire pour tout le cycle.

Le processeur exécute alors le programme instruction par instruction en rangeant à chaque fois les résultats en mémoire. En fin de cycle les sorties sont affectées d'un état binaire, par mise en communication avec les mémoires correspondantes. Dans ce cas, le temps de réponse à une variation d'état d'une entrée peut être compris entre un ou deux temps de cycle (durée moyenne d'un temps de cycle est de 5 à 15 ms. [3])

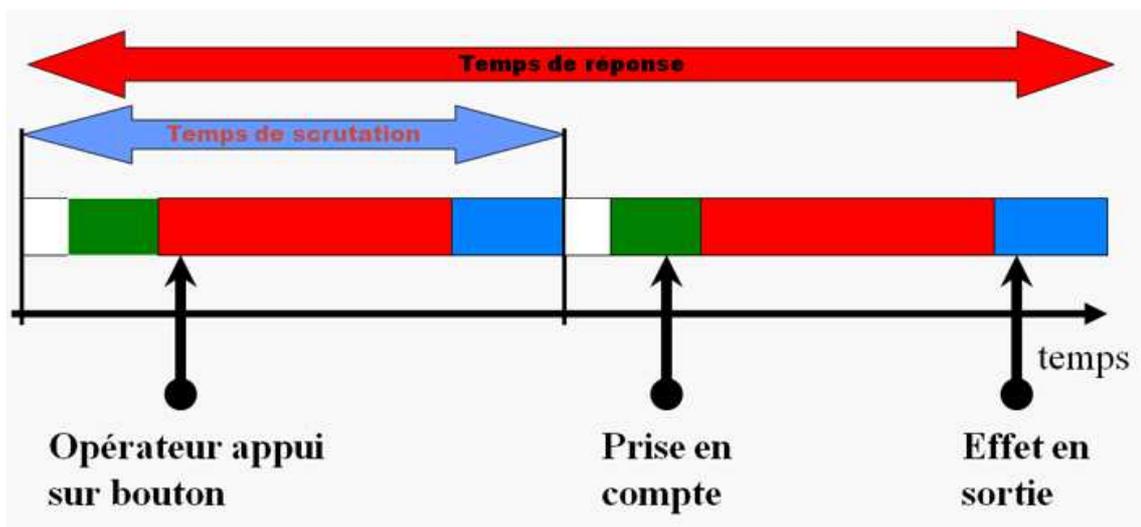


Figure II. 26 : Temps de scrutation vs Temps de réponse. [3]

Il existe d'autres modes de fonctionnement, moins courants :

- Synchrone par rapport aux entrées seulement
- Asynchrone [3]

### II.3.6 Domaines d'emploi des automates

On utilise les API dans tous les secteurs industriels pour la commande des machines (convoyage, emballage ...) ou des chaînes de production (automobile, agroalimentaire ...) ou il peut également assurer des fonctions de régulation de processus (métallurgie, chimie ...). Il est de plus en plus utilisé dans le domaine du bâtiment (tertiaire et industriel) pour le contrôle du chauffage, de l'éclairage, de la sécurité ou des alarmes. [26]

## II.3.7 Nature des informations traitées par l'automate

Les informations peuvent être de type :

- Tout ou rien (T.O.R.) : l'information ne peut prendre que deux états (vrai/faux, 0 ou 1 ...). C'est le type d'information délivrée par un détecteur, un bouton poussoir ...
- Analogique : l'information est continue et peut prendre une valeur comprise dans une plage bien déterminée. C'est le type d'information délivrée par un capteur (pression, température ...)
- Numérique : l'information est contenue dans des mots codés sous forme binaire ou bien hexadécimale. C'est le type d'information délivrée par un ordinateur ou un module intelligent [26]

### II.3.7.1 Traitement du programme automate

Tous les automates fonctionnent selon le même mode opératoire :

- Traitement interne : L'automate effectue des opérations de contrôle et met à jour certains paramètres systèmes (détection des passages en RUN / STOP, mises à jour des valeurs de l'horodateur, ...).
- Lecture des entrées : L'automate lit les entrées (de façon synchrone) et les recopie dans la mémoire image des entrées.
- Exécution du programme : L'automate exécute le programme instruction par instruction et écrit les sorties dans la mémoire image des sorties.
- Ecriture des sorties : L'automate bascule les différentes sorties (de façon synchrone) aux positions définies dans la mémoire image des sorties. [26]

Ces quatre opérations sont effectuées continuellement par l'automate (fonctionnement cyclique).

On appelle scrutation l'ensemble des quatre opérations réalisées par l'automate et le temps de scrutation est le temps mis par l'automate pour traiter la même partie de programme. Ce temps est de l'ordre de la dizaine de millisecondes pour les applications standards. [26]

Le temps de réponse total (TRT) est le temps qui s'écoule entre le changement d'état d'une entrée et le changement d'état de la sortie correspondante :

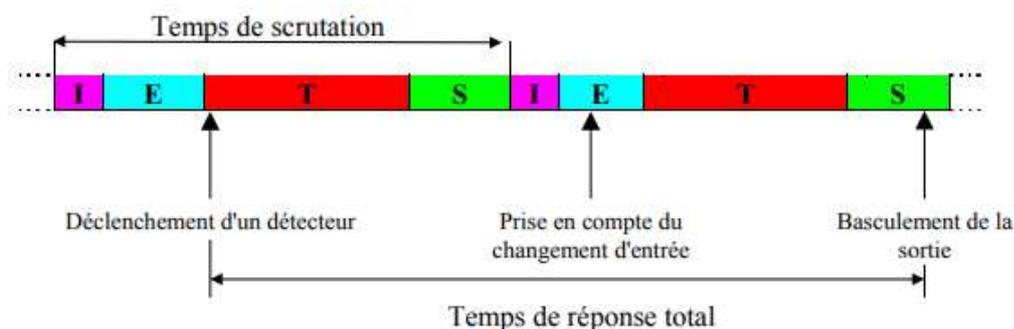


Figure II. 27 : Temps de réponse total. [26]

Le temps de réponse total est au plus égal à deux fois le temps de scrutation (sans traitement particulier).

Le temps de scrutation est directement lié au programme implanté. Ce temps peut être fixé à une valeur précise (fonctionnement périodique), le système indiquera alors tout dépassement de période.

Dans certains cas, on ne peut admettre un temps de réponse aussi long pour certaines entrées : ces entrées pourront alors être traitées par l'automate comme des événements (traitement événementiel) et prises en compte en priorité (exemples : problème de sécurité, coupure d'alimentation ...).

Certains automates sont également pour vues d'entrées rapides qui sont prises en compte avant le traitement séquentiel mais le traitement évènementiel reste prioritaire. [26]

### **II.3.8 Langage de Programmation**

La norme IEC 1131-3 définit cinq langages qui peuvent être utilisés pour la programmation des automates programmables industriels. Ces cinq langages sont :

- LD (« Ladder Diagram », ou schéma à relais): ce langage graphique est essentiellement dédié à la programmation d'équations booléennes (vraie/faux).
- IL (« Instruction List », ou liste d'instructions): ce langage textuel de bas niveau est un langage à une instruction par ligne. Il peut être comparé au langage assembleur.
- FBD (« Function Block Diagram », ou schéma par blocs): ce langage permet de programmer graphiquement à l'aide de blocs, représentant des variables, des opérateurs ou des fonctions. Il permet de manipuler tous les types de variables.
- SFC (« Sequential Function Char »): issu du langage GRAFCET, ce langage, de haut niveau, permet la programmation aisée de tous les procédés séquentiels.
- ST («Structured Text » ou texte structuré): ce langage est un langage textuel de haut niveau. Il permet la programmation de tout type d'algorithme plus ou moins complexe.[27]

#### **Objets communs**

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

- **BOOL (Booléen)** : ce type donne la valeur « Vraie » ou « Faux » qui sont équivalents à « 1 » ou « 0 »).
- **DINT (Entier)** : c'est un nombre entre -2147483647 et +2147483647. Il est exprimé dans l'une des bases suivantes : décimale, hexadécimale, octale ou binaire.
- **REAL (Réel)** il prend 1 bit de signe +23 bits de mantisse +8 bits d'exposant compris entre -37 et +37.
- **TIME (Temporisation)** : c'est une valeur strictement positive et commence par T# ou TIME#.
- **STRING (Chaîne)** : elle doit être précédée et suivie par une apostrophe, et ne doit jamais excéder 255 caractères). Le caractère spécial ('\$') est utilisée pour insérer des caractères non imprimables. [27]

#### a) Langage LD

Le langage LD (ladder diagram) 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.[27]

#### b) Langage IL

Le langage IL (instruction list), est un langage textuel de bas niveau. Il est particulièrement adapté aux applications de petite taille. Les instructions opèrent toujours sur un résultat courant (ou registre IL). L'opérateur indique le type d'opération à effectuer entre le résultat courant et l'opérande. Le résultat de l'opération est stocké à son tour dans le résultat courant.

Un programme IL est une liste d'instructions. Chaque instruction doit commencer par une nouvelle ligne, et doit contenir un opérateur, complété éventuellement par des modificateurs et, si c'est nécessaire pour l'opération, un ou plusieurs opérandes, séparés par des virgules (','). Une étiquette suivie de deux points (':') peut précéder l'instruction. 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.[27]

#### c) Langage FBD

Le langage FBD (function block diagram) est un langage graphique. Il permet la construction d'équations complexes à partir des opérateurs standards, de fonctions ou de blocs fonctionnels.

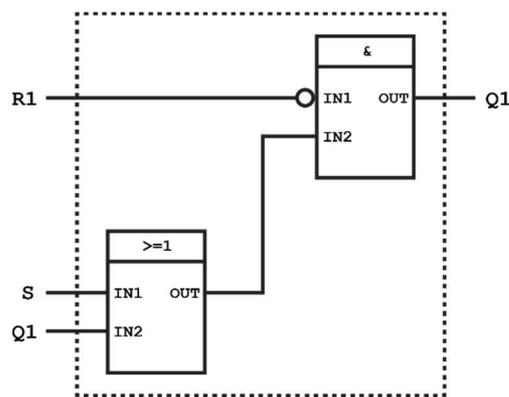


Figure II. 28 : Langage FBD. [27]

#### Les principales fonctions sont :

- L'énoncé RETURN (peut apparaître comme une sortie du diagramme, si liaison connectée prend l'état booléen TRUE, la fin du diagramme n'est pas interprétée).
- Les étiquettes et les sauts conditionnels sont utilisés pour contrôler l'exécution du diagramme. Aucune connexion ne peut être réalisée à droite d'un symbole d'étiquette ou de saut.
- Saut à une étiquette (le nom de l'étiquette est « LAB »).
- Si la liaison à gauche du symbole de saut prend l'état booléen TRUE, l'exécution du programme est dérivée après l'étiquette correspondante.
- L'inversion booléenne est représentée par un petit cercle. [23]

#### d) Langage SFC

Le langage SFC (Sequential Function Chart), ou GRAFCET, est un langage graphique utilisé pour décrire les opérations séquentielles. Le procédé est représenté comme une suite connue d'étapes (états stables), reliées entre elles par des transitions, une condition booléenne est attachée à chaque transition. Les actions dans les étapes sont décrites avec les langages ST, IL, LD ou FBD. [23]

#### e) Langage ST

Le langage ST (Structured Text) 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. C'est le langage par défaut pour la programmation des actions dans les étapes et des conditions associées aux transitions du langage SFC.

Un programme ST est une suite d'énoncés. Chaque énoncé est terminé par un point-virgule (« ; »). Les noms utilisés dans le code source (identificateurs de variables, constantes, mots clés du langage...) sont délimités par des séparateurs passifs ou des séparateurs actifs, qui ont un rôle d'opérateur. Des commentaires peuvent être librement insérés dans la programmation. [27]

## II.3.9 Avantages et inconvénients des API

### II.3.9.1 Avantages des API

- Evolutivité : très favorable à l'évolution 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 quelque ms.
- Modularité : (haute modularité) présentation modulaire en rack.
- Architecture de commande : centralisée ou décentralisée avec l'apparition d'une offre importante en choix de réseaux, bus de terrain, blocs E/S déportées.
- Maintenance : échanges standards et aide au diagnostic intégré [24]

### II.3.9.2 Inconvénients des API :

- En cas de "plantage" (très rare heureusement), c'est une belle pagaille...
- Son prix qui comme nous l'avons vu plus haut ne le met pas à la portée de toutes les bourses. Mais ces équipements évoluant rapidement fait que l'on peut en récupérer quelquefois pour pas trop cher.[27]

## II.3.10 Logiciel de programmation d'API (STEP 7)

STEP 7 est le progiciel de base pour la configuration et la programmation de systèmes d'automatisation SIMATIC. Il fait partie de l'industrie logicielle SIMATIC. Le progiciel de base STEP 7 existe en plusieurs versions :

- ✓ STEP 7-Micro/DOS et STEP 7-Micro/Win pour des applications autonomes simples sur SIMATIC S7 - 200.
- ✓ STEP 7 pour des applications sur SIMATIC S7-300/400, SIMATIC M7-300/400 et SIMATIC C7 présentant des fonctionnalités supplémentaires :
  - Possibilité d'extension grâce aux applications proposées par l'industrie logicielle SIMATIC (voir aussi Possibilités d'extension du logiciel de base STEP(7))
  - Possibilité de paramétrage de modules fonctionnels et de modules de communication
  - Forçage et fonctionnement multiprocesseur
  - Communication par données globales
  - Transfert de données commandé par événement à l'aide de blocs de communication et de blocs fonctionnels
  - Configuration de liaisons STEP 7 fait l'objet du présent manuel d'utilisation, STEP 7-Micro étant décrit dans la documentation "STEP 7-Micro/DOS". [28]

## II.4 Conclusion

Dans ce chapitre nous avons décrit en détail notre système étudié qui est une mini chaîne de production composée de 3 modules, on a décrit aussi le module de commande qui a une base d'un API de Siemens.

En deuxième étape nous avons expliqué les composants principaux d'un API, les langages de programmation et aussi on a dressé les avantages et les inconvénients d'un API.

En fin, on a parlé du logiciel de programmation et le langage utilisé dans notre travail.

# Chapitre 03

## Programmation et Réalisation d'une IHM

### III.1 Introduction

Nous avons consacré ce chapitre à la programmation globale du système depuis le cahier de charge jusqu'à la supervision, en passant par plusieurs étapes. Au début nous avons commencé par l'établissement d'un cahier de charge suivant les exigences de notre système et après cela la table de mnémonique. En fin, nous sommes passés à l'étape de programmation par logiciel (STEP7) où nous avons traduit notre cahier de charge en langage GRAFCET. A partir de ce programme, nous avons essayé de développer une IHM qui va représenter principalement le déplacement de la pince. Pour concrétiser notre travail, l'ensemble a été simulé en utilisant PLCsim, côté programme en Grafcet et RunTime, côté IHM et la supervision.

### III.2 Etablissement du cahier de charge

Notre système contient des capteurs, des actionneurs et des prés actionneurs. Pour cela, il nous a été demandé de :

- Développer un programme de fonctionnement en grafcet utilisant le logiciel step7. Ce programme va nous permettre de faire la rotation d'une table tournante et le déplacement d'une pince pour récupérer les pièces de la table et les déposer sur le convoyeur et vice versa.
- Superviser le fonctionnement de notre mini chaine. Il nous a été imposé de créer une IHM utilisant le logiciel WinCC. Cette IHM doit nous permettre de commander notre mini chaine et suivre son fonctionnement en temps réel.

### III.3 Programmation du fonctionnement de notre mini-chaine sur STEP7

Comme vous le savez, le Step7 dédié à la programmation des APIs de marque Siemens. Il comporte plusieurs langages de programmation, parmi eux le langage Grafcet qui est un langage initialement développé pour être facile à utiliser et très organisé pour être exploiter par différents utilisateurs de niveaux variés. Pour expliquer son utilisation, on a donné une un organigramme qui se trouve dans le paragraphe suivant qui montre les différentes étapes nécessaires pour obtenir un programme fiable et sans erreur.

#### III.3.1 Tâches fondamentales

La mise en place d'une solution d'automatisation avec STEP 7 nécessite la réalisation de tâches fondamentales. La figure suivante indique les tâches à exécuter dans la plupart des projets qui sont classés selon la marche à suivre. Nous allons commencer par une petite introduction sur le langage Grafcet, puis on entrera directement dans la programmation du fonctionnement de notre système.

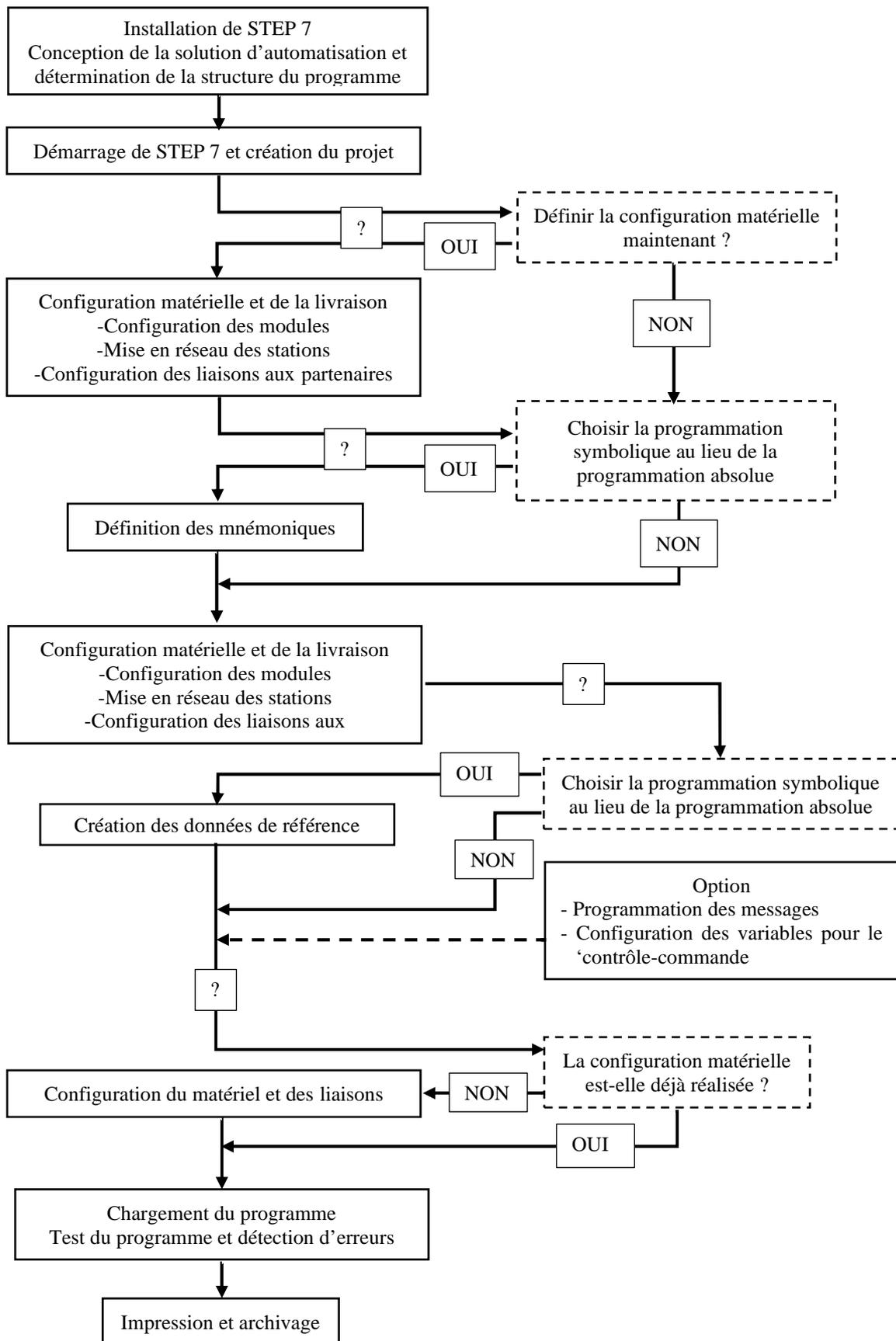


Figure III. 29 : Tâches fondamentales pour créer un projet sur step7.

### III.3.1.1 Langage de programmation GRAPH

Le langage de programmation GRAPH s'ajoute à l'éventail des fonctions de STEP 7. Il permet de programmer graphiquement les commandes séquentielles. Ces commandes séquentielles pourront être ensuite commandées à l'aide d'un automate programmable SIMATIC. Dans ce but, vous décomposez le processus en étapes ayant une fonctionnalité bien précise, vous faites une représentation graphique de son déroulement et vous pouvez imprimer une documentation graphique et textuelle. Les actions à exécuter sont associées aux étapes, tandis que des transitions régissent l'évolution entre deux étapes successives (conditions de franchissement). Pour définir les réceptivités des transitions ainsi que les verrouillages ou les surveillances d'étape, vous aurez à utiliser un nombre restreint d'éléments du langage de programmation CONT (schéma à contacts) ou LOG (logigramme).

## III.3.2 Conception du programme de fonctionnement

### III.3.2.1 Création de projet

Comme indiqué dans l'organigramme de la figure ci-dessus, un bon programme repose sur une structuration et une bonne solution. La première des choses à faire est la création d'un projet qui nous permet de choisir l'API adéquat (dans notre, c'est le S7-300) en sélectionnant la CPU et les différents modules. LA figure ci-dessus montre une fenêtre où se trouve tous instances choisies et nécessaire pour la conception de notre projet.

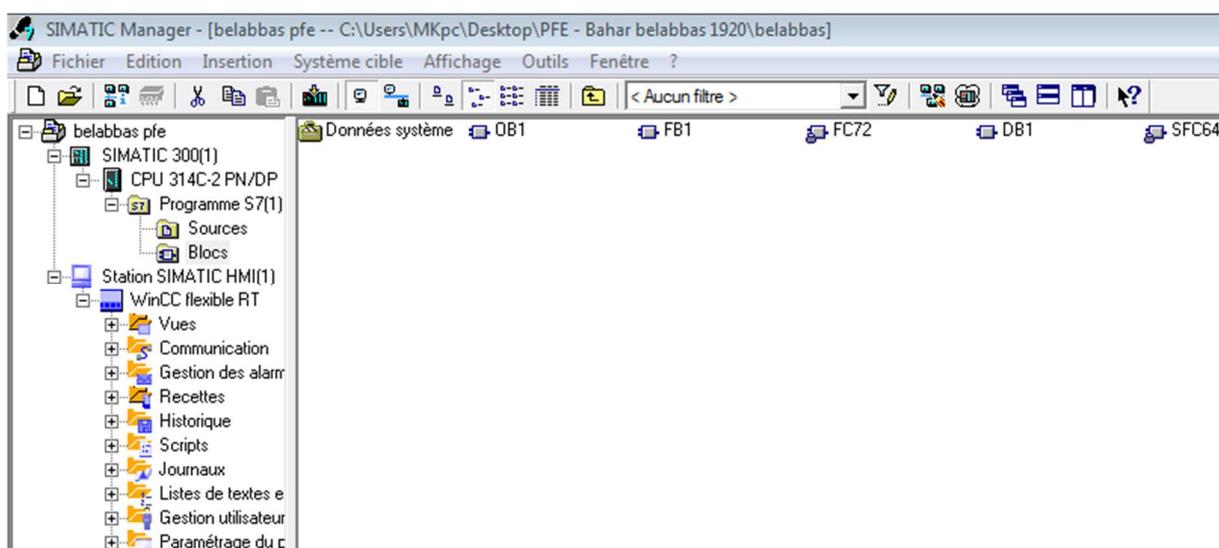


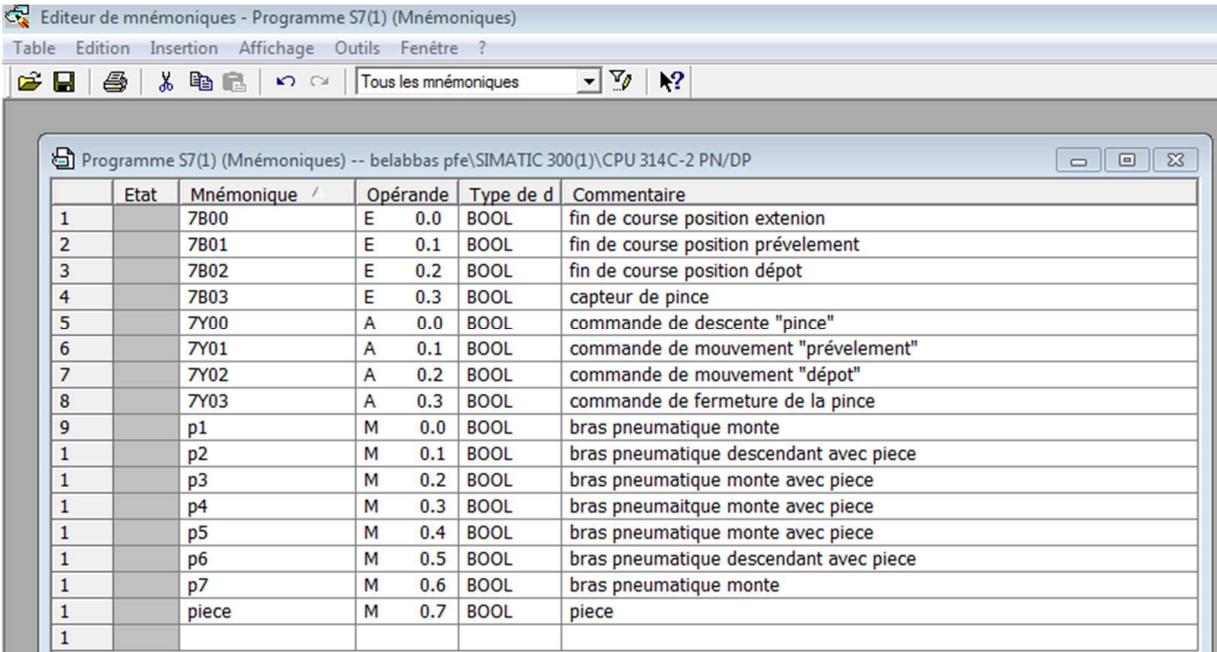
Figure III. 30 : Vue globale du projet et ses constituants principaux.

La figure montre les différents blocs créés par Step7 après avoir choisi la CPU, les différents modules et le langage utilisé qui est Grafcet, sans oublier l'étape importante qui est

la création de la table de mnémoniques. C'est la pièce maitresse dans l'enchaînement de notre programme sachant qu'il nous facilite la sélection des entrées/sorties durant la conception en affichant la désignation.

### III.3.2.2 Table de mnémoniques

Notre table comporte 04 entrées logiques (désignées par E0.0 à E0.3), 04 sorties logiques (désignées par A0.0 à A0.3) et un ensemble de mémentos introduits pour le bon fonctionnement de notre programme, fonction de la solution trouvées. La figure ci-dessous englobe les différentes déclarations avec leur commentaire, principalement les noms des capteurs et actionneurs de notre mini-chaîne.



	Etat	Mnémonique /	Opérande	Type de d	Commentaire
1		7B00	E 0.0	BOOL	fin de course position extension
2		7B01	E 0.1	BOOL	fin de course position prélevement
3		7B02	E 0.2	BOOL	fin de course position dépôt
4		7B03	E 0.3	BOOL	capteur de pince
5		7Y00	A 0.0	BOOL	commande de descente "pince"
6		7Y01	A 0.1	BOOL	commande de mouvement "prélevement"
7		7Y02	A 0.2	BOOL	commande de mouvement "dépôt"
8		7Y03	A 0.3	BOOL	commande de fermeture de la pince
9		p1	M 0.0	BOOL	bras pneumatique monte
1		p2	M 0.1	BOOL	bras pneumatique descendant avec piece
1		p3	M 0.2	BOOL	bras pneumatique monte avec piece
1		p4	M 0.3	BOOL	bras pneumatique monte avec piece
1		p5	M 0.4	BOOL	bras pneumatique monte avec piece
1		p6	M 0.5	BOOL	bras pneumatique descendant avec piece
1		p7	M 0.6	BOOL	bras pneumatique monte
1		piece	M 0.7	BOOL	piece
1					

Figure III. 31 : Table de mnémoniques obtenue de Step7.

### III.3.2.3 Configuration du réseau

Cette étape est importante car elle nous permet de tester le fonctionnement de notre programme utilisant l'API virtuel PLCsim. Nous avons choisi dans notre cas la communication par câble Ethernet pour le chargement de notre programme et sa simulation par PLCsim.

Après avoir effectué cette étape avec succès, est venu l'étape primordiale qui est la conception de notre Grafcet de fonctionnement.

### III.3.2.4 Conception de notre programme pas à pas

Comme montré dans la figure III.30, nous avons utilisé uniquement un seul bloc organisationnel pour héberger notre programme.

Le programme Grafcet repose essentiellement de l'état initial et de la transition adjacente l'état. La figure suivante montre les 04 premiers états et leurs transitions.

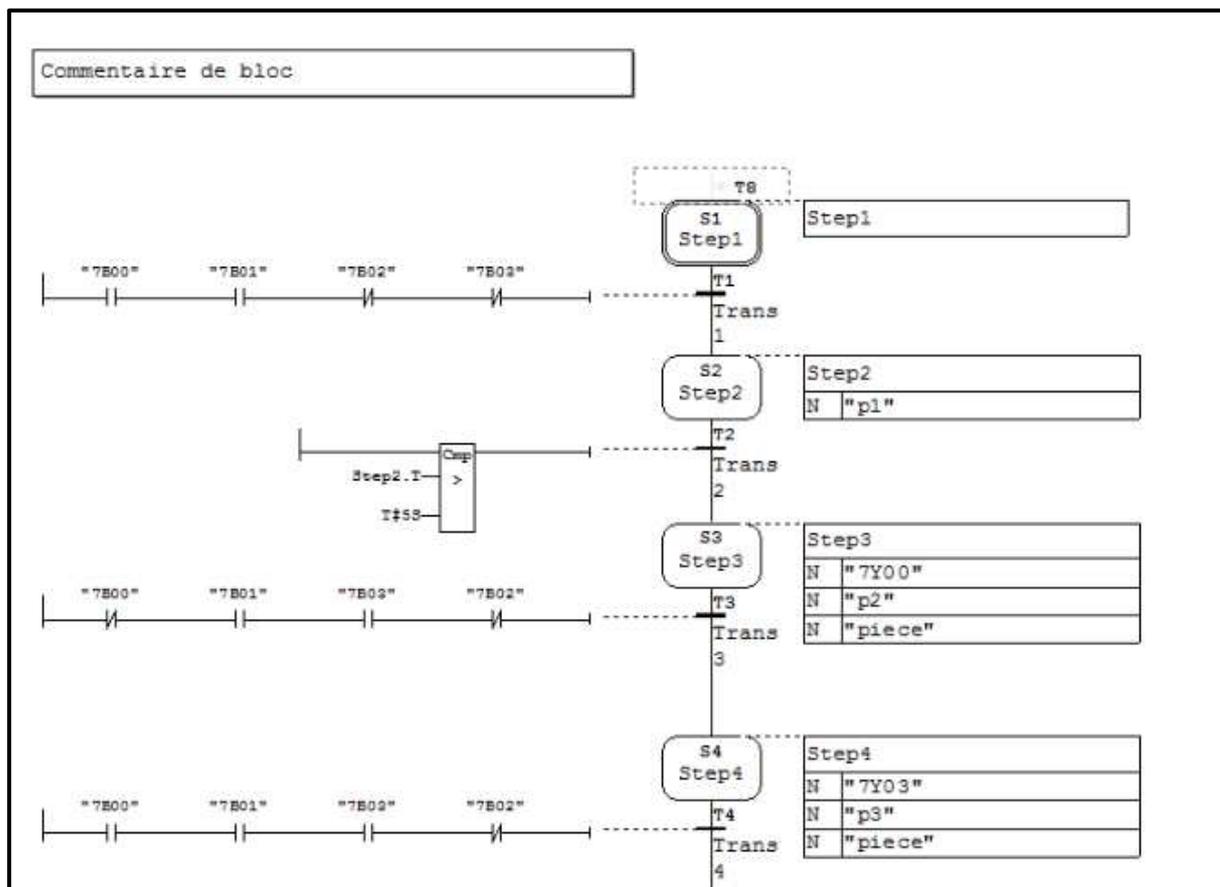


Figure III. 32 : Représentations des 04 premiers états de notre programme

D'après cette capture, on a montré que le système ne peut démarrer qu'après que la pince ne soit en position de récupération de pièce (Fin de course position prélèvement actionné 7B01) et le bras étendu (Fin de course position extension actionné 7B00). Le démarrage correspond à la descente de la pince (actionneur Commande descente de la pièce alimentée 7Y00 «'Step3)).

Après un certain instant, la pince atteint l'emplacement exact de la pièce (capteur pièce 7B03 actionné), à cet instant, on actionne la commande de fermeture de pince et le prélèvement commence (7Y03 actionné "Step4)).

Avant le déplacement, la pince doit remonter pour s'éloigner du point de récupération jusqu'à atteindre le position de dépôt (7B02 est actionné, Step5, 6, 7 et 8 de la Figure ), la pince

commence à descendre jusqu'à attendre le niveau bas ou position étendu. On commande le relâchement de la pince et le bras retourne à la position de prélèvement.

La pince d'épouse (un contact point r) Cette partie montre que le système ne peut démarrer que si seulement si la pince se trouve près de la pince

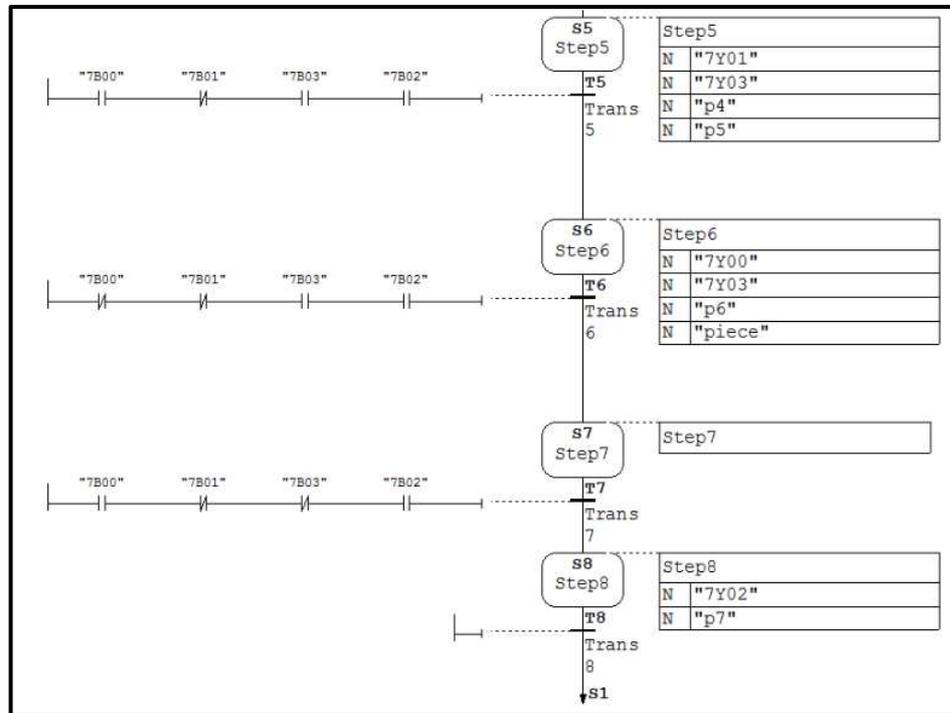


Figure III. 33 : Représentation des états de déplacement de la pince.

Le fonctionnement de la pince est cyclique tant qu'il y a présence de pièce sur le convoyeur ou la table.

La figure ci-dessous montre le bloc OB1 après compilation de notre programme sans anomalie de fonctionnement.

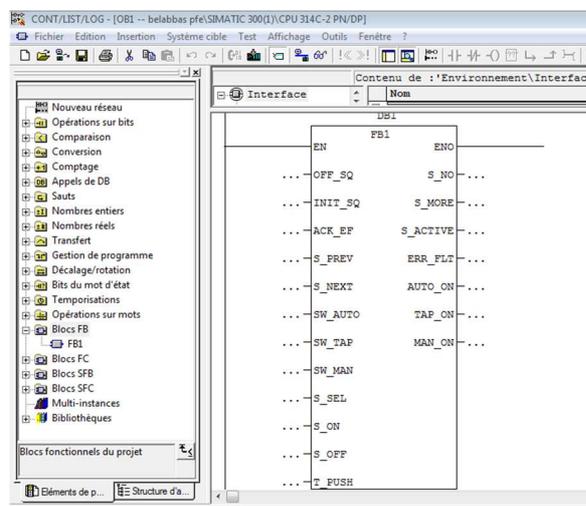


Figure III. 34 : Représentation d'OB1 après compilation.

### III.3.2.5 Simulation du programme par PLCSIM

On utilise le PLCsim pour tester notre programme. La figure ci-dessous représente une partie de fonctionnement de notre System. Il montre le bon fonctionnement du programme sans aucunes erreurs. Après avoir lancer notre simulateur et cocher la case "RUN-P" de la CPU, l'état initial devient actif (couleur "verte") et attend que les conditions de transition seront vérifiées (Figure ci-dessous).

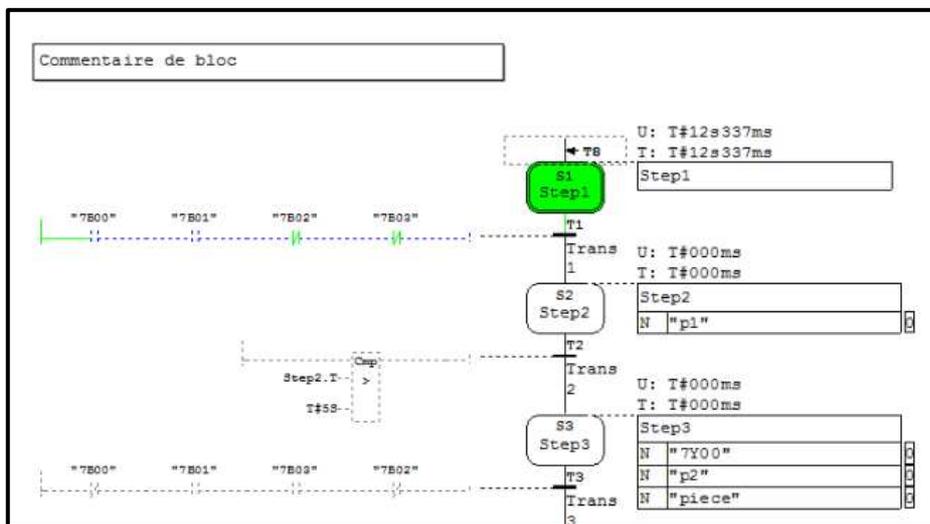
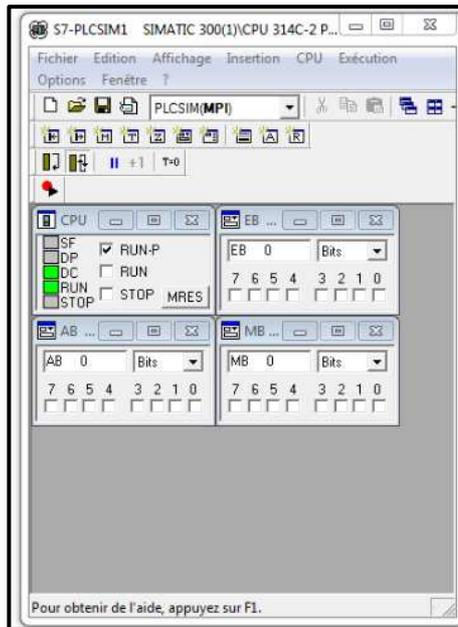


Figure III. 35 : Simulation en représentant l'état initial et le franchissement.

La figure suivante montre l'activation de l'état "Step3" après que la transition est franchie. Dans le simulateur, les actionneurs activés coïncident exactement avec ceux contenus au niveau de l'état (sur le Grafcet).

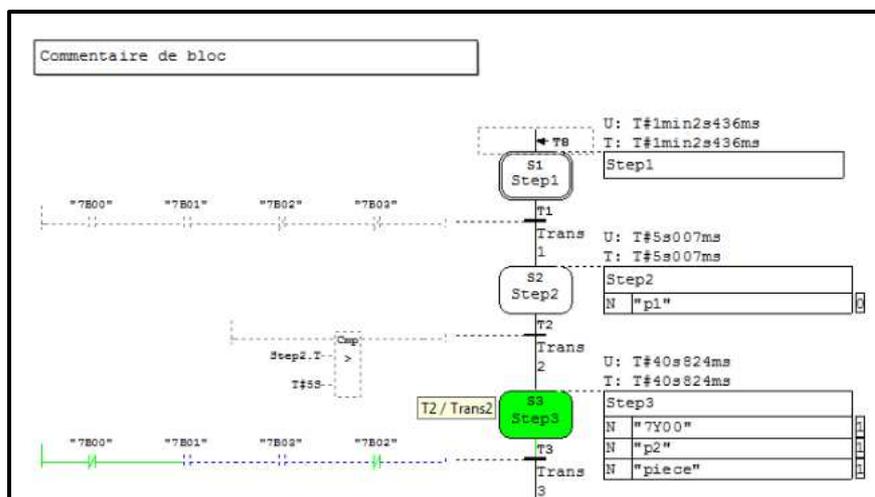
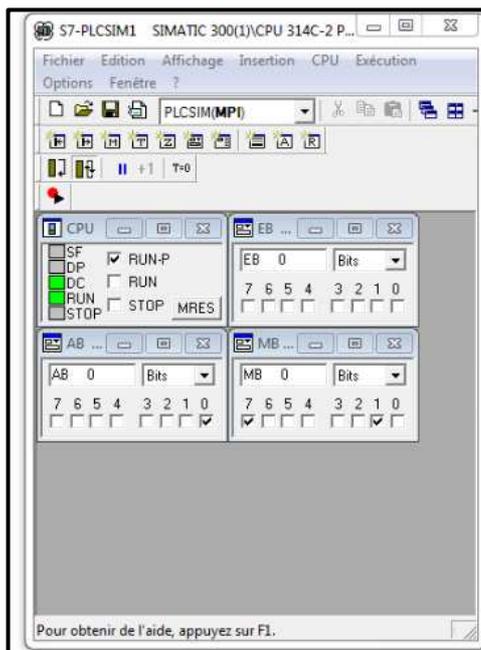


Figure III. 36 : Activation de l'état S3 et mise à 1' de '7Y00' au niveau du simulateur (A0.0).

Après avoir testé le bon fonctionnement de notre programme (comme le montre les figures III.35 et III.36), on va passer aux étapes de création de notre IHM.

Remarque : d'autres captures sont représentées dans l'annexe A

### III.4 Création d'une IHM

La création d'une IHM comporte plusieurs étapes : de la création du projet, en passant par la liaison et la communication entre IHM et l'API jusqu'au développement de notre IHM d'une façon ergonomique et claire. C'est l'étape la plus importante dans la création de l'IHM.

### III.4.1 Création d'un projet WinCC

La figure ci-dessous montre l'étape après création du projet WinCC.

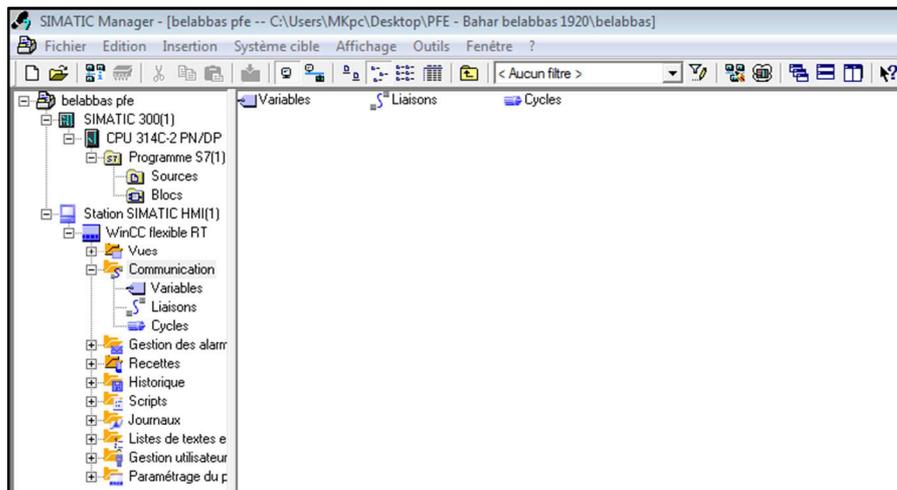
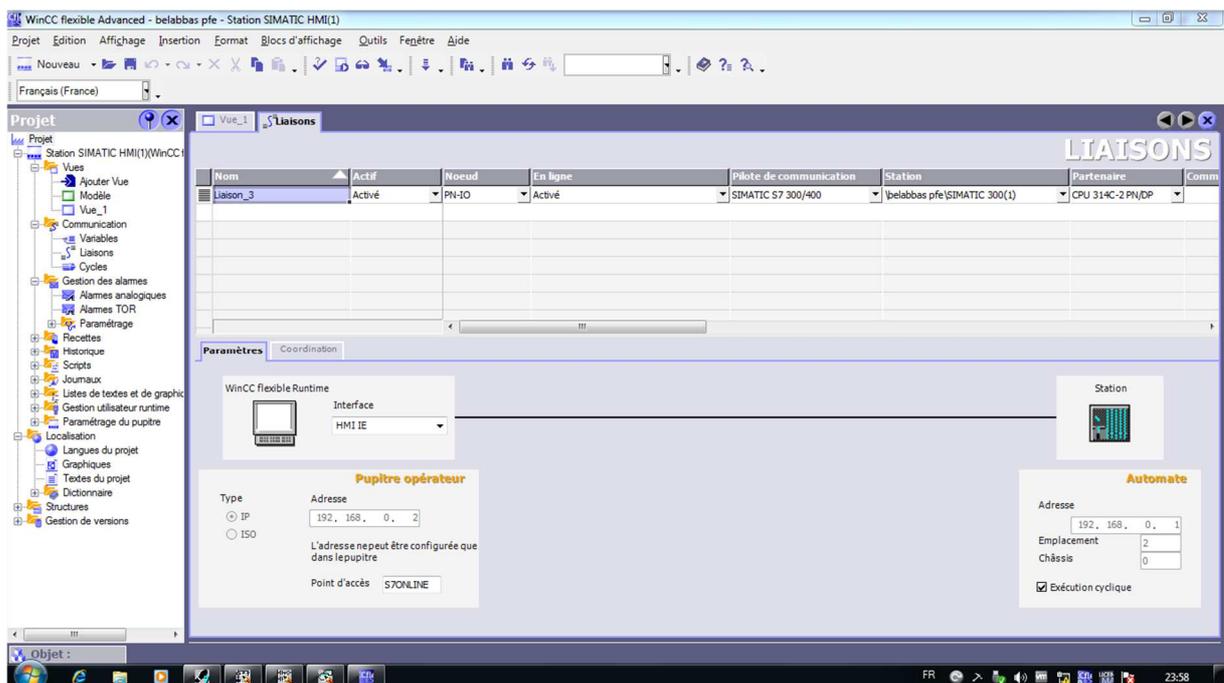


Figure III. 37 : Création du projet et son insertion à Step7.

L'étape suivante est la création de la liaison entre l'IHM avec l'API, qui est une étape primordiale.

### III.4.2 Établissement d'une liaison IHM-PLC

La liaison IHM-PLC s'est fait à travers la communication Ethernet comme le montre les deux figures suivantes. Elle montre aussi que la communication est bien établie et les IP bien choisies.



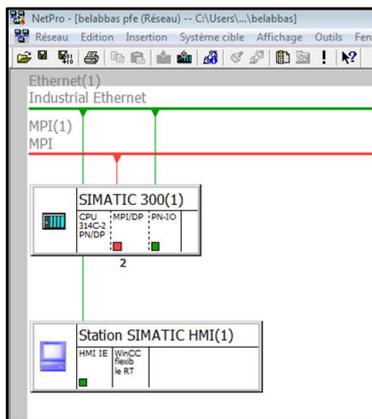


Figure III. 38 : Représentation de la communication entre notre IHM et l'API.

L'étape suivante consiste à créer une liaison entre les variables de l'IHM et de l'API. La figure III.37 montre clairement que la liaison est réussie.

### III.4.2.1 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.

La Table des variables IHM suivante montre la liaison en précisant le cycle d'acquisition qui est un paramètre très important dans la communication en temps réel.

VARIABLES									
Nom	Nom d'affichage	Liaison	Type de données	Mnémonique	Adresse	Éléments du ta...	Cycle d'acqui...	Commentair	
7800		Liaison_3	Bool	7800	I 0.0	1	100 ms		
7801		Liaison_3	Bool	7801	I 0.1	1	100 ms		
7802		Liaison_3	Bool	7802	I 0.2	1	100 ms		
7803		Liaison_3	Bool	7803	I 0.3	1	100 ms		
7700		Liaison_3	Bool	7700	Q 0.0	1	100 ms		
7701		Liaison_3	Bool	7701	Q 0.1	1	100 ms		
7702		Liaison_3	Bool	7702	Q 0.2	1	100 ms		
7703		Liaison_3	Bool	7703	Q 0.3	1	100 ms		
p1		Liaison_3	Bool	p1	M 0.0	1	100 ms		
p2		Liaison_3	Bool	p2	M 0.1	1	100 ms		
p3		Liaison_3	Bool	p3	M 0.2	1	100 ms		
p4		Liaison_3	Bool	p4	M 0.3	1	100 ms		
p5		Liaison_3	Bool	p5	M 0.4	1	100 ms		
p6		Liaison_3	Bool	p6	M 0.5	1	100 ms		
p7		Liaison_3	Bool	p7	M 0.6	1	100 ms		
pièce		Liaison_3	Bool	pièce	M 0.7	1	100 ms		

Figure III. 39 : Table des variables obtenue de WinCC.

### III.4.2.2 Stratégie de développement d'une pince en déplacement

Dans cette rubrique, nous allons vous montrer la stratégie prise pour montrer au niveau de notre IHM la pince se déplacer.

Pour déplacer la pince, l'idée de placer plusieurs figures de pinces mais d'orientation différents suivant son emplacement sur l'IHM comme le montre la figure ci-dessous

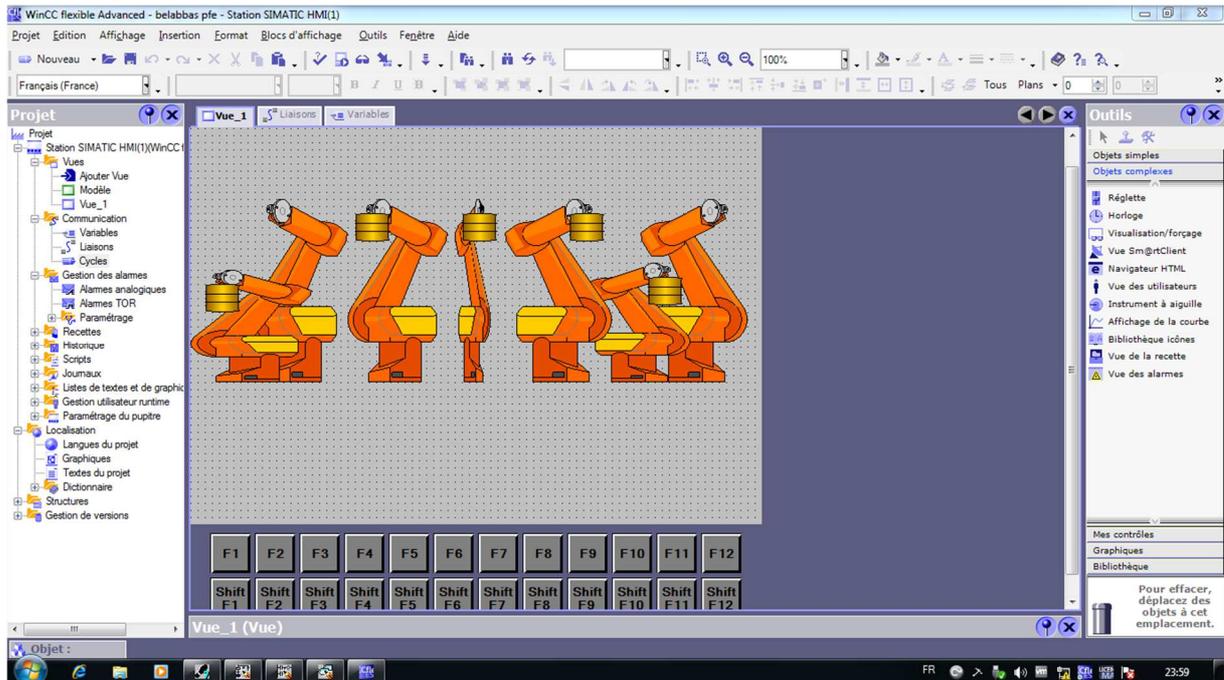


Figure III. 40 : Représentation de plusieurs figures avec des orientations différentes.

La seule différence entre les figures est sa variable déclarée (par exemple la figure de droite est liée à la variable 'p1' qui se trouve dans la table de variable de la figure III.38). En d'autres termes, chaque figure comporte une variable qui sera utilisée pour visualiser la figure (ou la masquer) suivant le temps de déplacement de la pince en temps réel. On choisira un temps de visualisation de tel façon que le déplacement devient naturel.

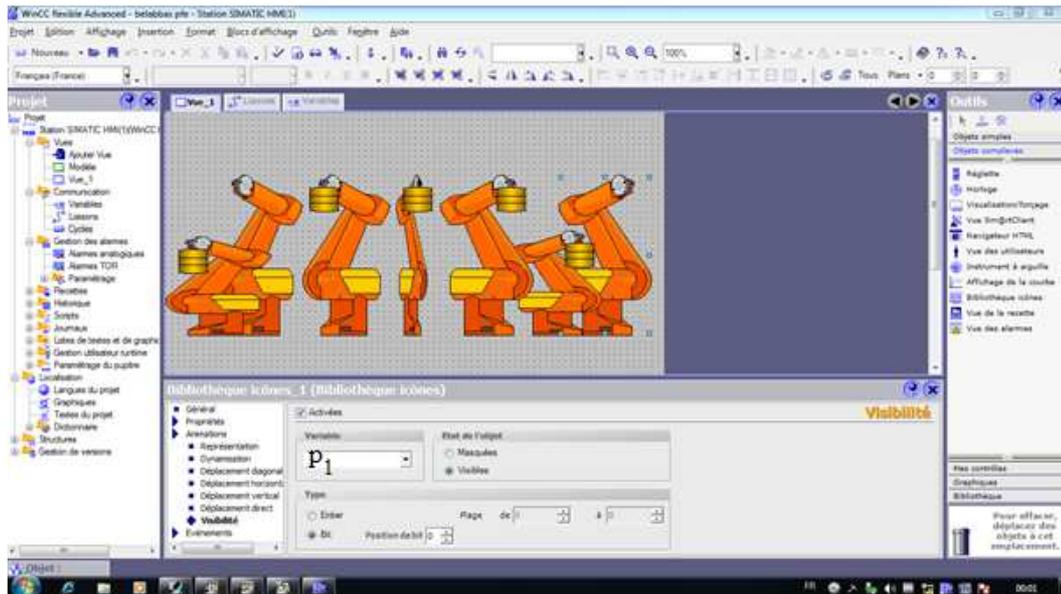


Figure III. 41 : Représentation des propriétés de la figure de droite : 'p1' comme variable, 'visible' comme évènement.

J'ai choisi cet exemple car c'est la 1<sup>ère</sup> figure qui sera affichée au lancement de Runtime.

### III.5 Conclusion

Ce chapitre est le plus important car il a permis de tester nos connaissances dans le domaine de programmation des API par le langage Grafcet qui est le langage le plus utilisé dans le monde industriel pour ces performances et son ergonomie dans sa conception.

Notre travail ne s'est pas arrêté à la programmation des API, il s'est étalé à la configuration des protocoles de communication que ce soit PC-API ou PC-API-IHM.

Le développement de notre IHM nous a permis d'étendre notre imagination pour atteindre une représentation presque réelle de notre pince en déplacement.

# Conclusion Générale

Le but de ce travail présenté dans ce projet de fin d'étude est la commande et la supervision de trois modules : Convoyeur, Pince et la Table.

Ces trois modules forment une mini-chaîne de la marque ElectroVeneta. Initialement, cette mini-chaîne était commandée par un API d'ancienne génération S7-200 qui est devenu obsolète car sa version du logiciel de programmation n'est sur le marché. Donc, nous étions contraints de modifier ou de créer notre propre programme sur STEP 7 et utilisé un récent API qui est S7 300 de Siemens.

Il faut le dire, ce travail nous a permis de s'épanouir dans le domaine de la programmation des API. Il aurait été plus bénéfique s'il n'y a pas eu cette pandémie qui nous a contraint à ne pas faire de réalisation pratique sur cette mini-chaîne.

Ce travail nous a aussi permis de s'initier dans le domaine de développement d'IHM avec création d'objet mobile comme, dans le cas de notre travail, l'objectif est de faire déplacer la pince en même temps que le déplacement de la pince de la mini-chaîne en réel. On n'a pas pu réaliser notre objectif à 100% mais on a pu le tester en utilisant le simulateur PLCsim qui est logiciel très puissant et capable de générer des événements réels et semblables comme un API en réel.

Dans ce travail, avant d'entamer notre objectif, une description complète des systèmes automatisés a été présentée. On a trouvé aussi judicieux de discuter du protocole SCADA qui prend une ampleur très importante dans le domaine industriel en général et dans l'industrie 4.0 en particulier.

La compréhension du fonctionnement de notre mini-chaîne était une phase importante, malheureusement obturé par cette pandémie mais le résultat obtenu donne beaucoup de perspectives. Parmi ces perspectives :

- ✓ Programmation directement sur Tia Portal avec une version récente. Je trouve que c'est plus bénéfique et mieux structuré que STEP 7.
- ✓ Améliorer notre IHM en introduisant des objets conçus qui reflète réellement l'objet qu'on veut représenter comme dans le cas de CIROS.

## Conclusion Générale

- ✓ La difficulté de cette mini-chaîne se trouve dans le réseau électrique où on a besoin énormément de fil électrique pour l'alimentation et la connexion API-capteurs, pré actionneurs-actionneurs.

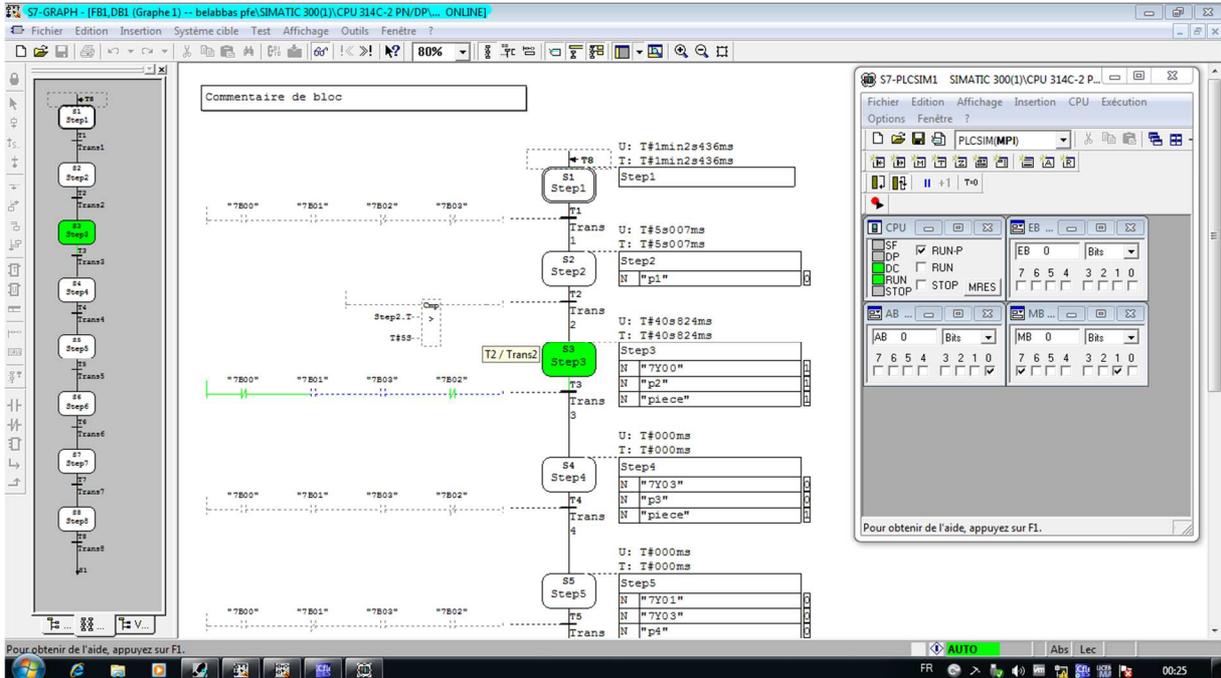
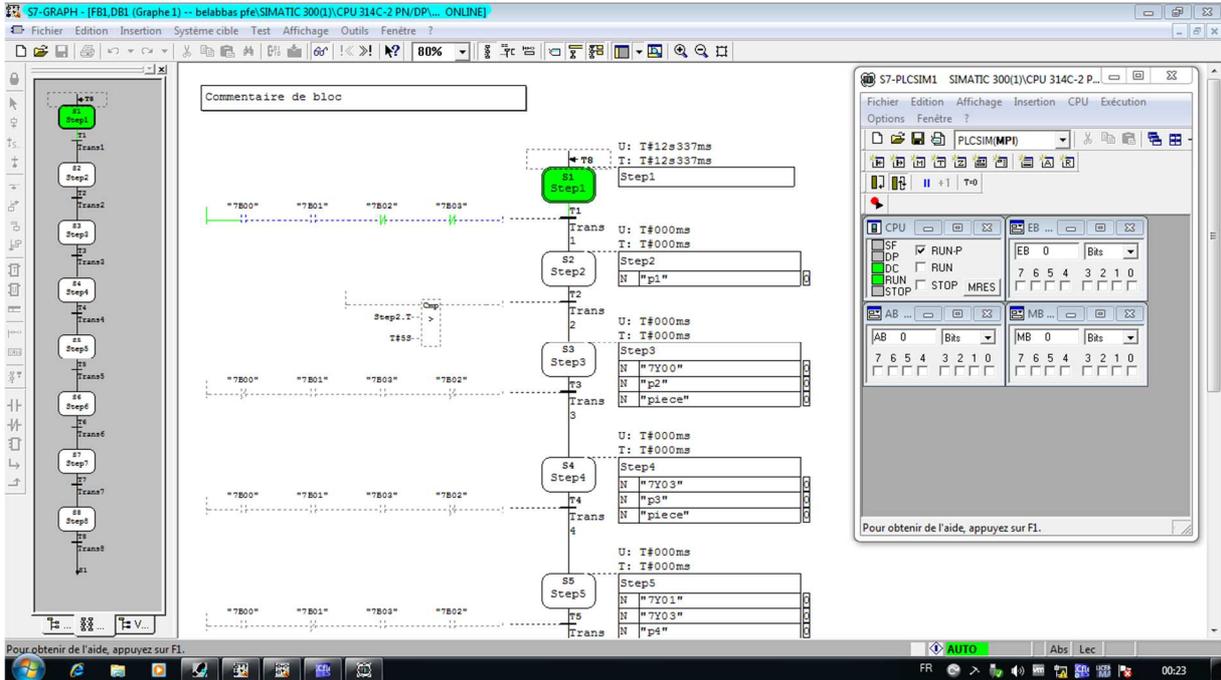
# Références Bibliographiques

- [1] <http://lyc58-renardfollereau.ac-dijon.fr/ljr/cours/supsi/si/ commande-système /ch1-SystBoucles.pdf>
- [2] R. Glaa, Mohamed Najeh Lakhoua. « Methodology of analysis and design of a SCADA system. », Conference: 2014 International Conference on Electrical Sciences and Technologies in Maghreb, CISTEM 2014
- [3] <https://www.technologuepro.com/cours-automate-programmable-industriel /Les-automates-programmables-industriels-API.htm>
- [4] <https://www.intesens.com/de-lindustrie-1-0-a-lindustrie-4-0-trois -siecles - de-transformation/>
- [5] [http://technologieaucollege.free.fr/ressources\\_web/ressources2.techno.free.fr /mecanique /systemes/descript.htm#:~:text=La%20Partie%20Commande%20%3A,dans%20ses%20m%C3%A9moires%20un%20programme.](http://technologieaucollege.free.fr/ressources_web/ressources2.techno.free.fr /mecanique /systemes/descript.htm#:~:text=La%20Partie%20Commande%20%3A,dans%20ses%20m%C3%A9moires%20un%20programme.)
- [6] M. L. Dilmi. « Contribution à la modélisation des système. », Mémoire de Master, Université Ferhat Abbas-Sétif, faculté de Technologie, No. Réf. : MAC21/JUIN/2014
- [7] H. Mezdoor, M. Ayab. « Etude et réalisation d'un système de supervision sous YOKOGAWA CS3000, Application à l'unité de production d'air de l'entreprise nationale SONATRACH. », Mémoire de Master, Université 8 Mai 1945 – Guelma, Faculté des Sciences et de la Technologie, 2019
- [8] <https://crushtymks.com/fr/scada/1459-3-generations-of-scada-system-architectures-you-should-know-about.html>
- [9] R. Debibi. « Développement d'un système de supervision, d'acquisition et de transport de données d'un système hybride de production électrique'', Mémoire d'Ingénieur Université Québec, 2016.
- [10] H. MOUSS. « Supervision et sûreté des systèmes industriels'', Cours Master II en Automatique, Université de Batna
- [11] P. Nord. « SCADA, La sécurité des systèmes industriels'', Cours de Synthèse, Paris, 2019.
- [12] <https://www.copadata.com/fr/produits/zenon-software-platform/visualisation-contrôle/quest-ce-qu-une-ihm-interface-homme-machine-copa-data/#:~:text=IHM%20signifie%20interface%20homme%20machine,programme%20informatique%20ou%20un%20système%20me.>

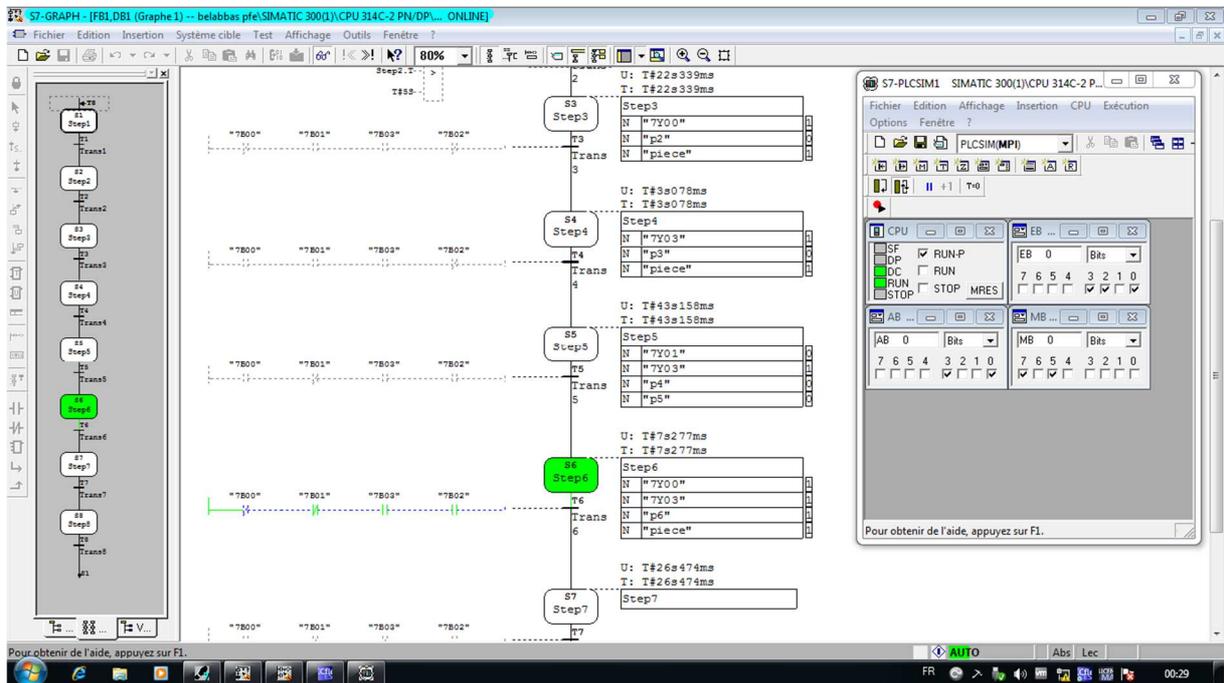
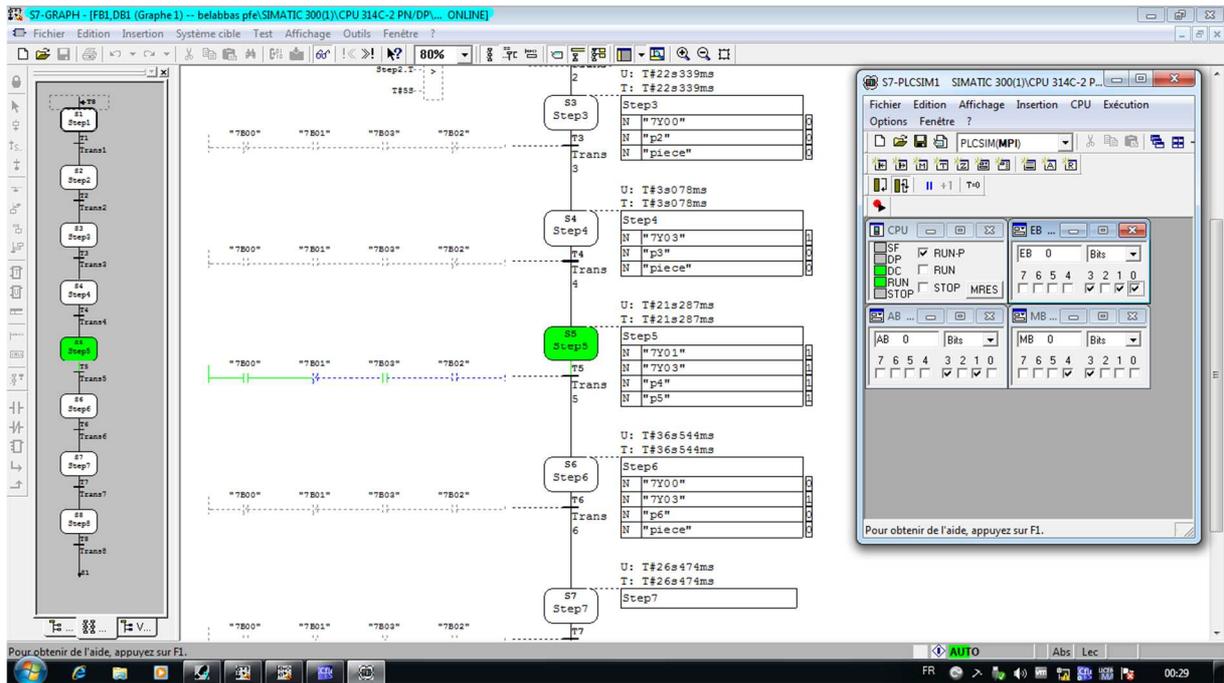
- [13] <https://www.maxicours.com/se/cours/l-interface-homme-machine-et-contrôles-graphiques/#:~:text=L'ensemble%20du%20travail%20de,%C3%A9%20essentiell%20%3A%20l'ergonomie.>
- [14] [http://tjdm33.free.fr/index.php?article155/interface-homme-machine-ihm-ou-scada#:~:text=Les%20IHM%20sont%20proches%20des,un%20syst%C3%A8me%20de%20contr%C3%B4le%20industriel.&text=Il%20s'agit%20d'outils,\)%20d'un%20syst%C3%A8me%20SCADA.](http://tjdm33.free.fr/index.php?article155/interface-homme-machine-ihm-ou-scada#:~:text=Les%20IHM%20sont%20proches%20des,un%20syst%C3%A8me%20de%20contr%C3%B4le%20industriel.&text=Il%20s'agit%20d'outils,)%20d'un%20syst%C3%A8me%20SCADA.)
- [15] <https://www.hellopro.fr/logiciel-de-supervision-contrôle-commande-automatisation-scada-ihm-zenon-1001305-11937-produit.html>
- [16] <https://www.jsautomation.fr/Produit/zenon-logiciel-hmi-scada/#:~:text=zenon%20est%20un%20logiciel%20facile,produit%20qu'il%20vous%20faut.>
- [17] <https://www.movicon.fr/>
- [18] <https://www.vipa.com/fr/produits/afficheurs/logiciels-pour-hmi/#:~:text=Movicon%20sont%20des%20logiciels%20d,directement%20utilisable%20par%20l'op%C3%A9rateur.>
- [19] <https://www.google.com/url?sa=i&url=https%3A%2F%2Fwww.myscada.org%2Fdownloadsde%2F&psig=AOvVaw1ITKdZqS4GxDLFWuW08RpY&ust=1600207328161000&source=images&cd=vfe&ved=0CAIQjRxqFwoTCLiSnP7S6esCFQAAAAAdAAAAABAD>
- [20] <https://www.directindustry.fr/prod/myscada/product-213769-2211865.html>
- [21] <https://www.google.com/url?sa=i&url=https%3A%2F%2Fdreamreport.net%2Fblog%2Fsiemens-simatic-wincc-report-generator-dream-report%2Fsiemens-simatic-wincc-logo%2F&psig=AOvVaw0TXgpyQmkYISQhmJJV18NY&ust=1600207543218000&source=images&cd=vfe&ved=0CAIQjRxqFwoTCLiL3uTT6esCFQAAAAAdAAAAABAX>
- [22] <https://fr.wikipedia.org/wiki/WinCC#:~:text=SIMATIC%20WinCC%20est%20un%20syst%C3%A8me,processus%20industriels%20et%20des%20infrastructures.>
- [23] [https://www.google.com/url?sa=i&url=https%3A%2F%2Fwww.elettronicaveneta.com%2Ffr%2Fprodotto%2Fbanc-pour-letude-de-lapi-2%2F&psig=AOvVaw2\\_D357MIOv-Hq3tuw6uMYs&ust=1600209114599000&source=images&cd=vfe&ved=0CAIQjRxqFwoTCKCc29LZ6esCFQAAAAAdAAAAABAO](https://www.google.com/url?sa=i&url=https%3A%2F%2Fwww.elettronicaveneta.com%2Ffr%2Fprodotto%2Fbanc-pour-letude-de-lapi-2%2F&psig=AOvVaw2_D357MIOv-Hq3tuw6uMYs&ust=1600209114599000&source=images&cd=vfe&ved=0CAIQjRxqFwoTCKCc29LZ6esCFQAAAAAdAAAAABAO)
- [24] S. Mellali, L. Yousfi. « Etude de l'automatisation et de la supervision d'un procédé de lavage de filtres Niagara à CEVITAL- TIA PORTAL V12. », Mémoire de Master, Université Abderrahmane MIRA de Bejaia, Faculté de Technologie, 2017.

- [25] <https://www.achoisir.fr/comment-choisir-un-automate-programmable-industriel/#:~:text=%2D%20La%20marque%20%3A%20Tenir%20compte%20de,API%20d'une%20certaine%20marque.&text=Il%20est%20donc%20n%3%A9cessaire%20de,la%20part%20de%20celui%2Dci>.
- [26] [https://conceptec.net/fr/techniques-de-base/automatisme/l-automate-programmable-industrielapi#:~:text=La%20norme%20IEC%201131%2D3,Ces%20cinq%20langages%20sont%20%3A&text=SFC%20\(%C2%AB%20Sequential%20Function%20Char%20%C2%BB,de%20tous%20les%20proc%3%A9d%3%A9s%20s%C3%A9quentiels](https://conceptec.net/fr/techniques-de-base/automatisme/l-automate-programmable-industrielapi#:~:text=La%20norme%20IEC%201131%2D3,Ces%20cinq%20langages%20sont%20%3A&text=SFC%20(%C2%AB%20Sequential%20Function%20Char%20%C2%BB,de%20tous%20les%20proc%3%A9d%3%A9s%20s%C3%A9quentiels).
- [27] <https://www.courstechpro.com/2020/07/les-avantages-et-inconvenients-de-l-api.html>
- [28] [https://cache.industry.siemens.com/dl/files/107/45531107/att\\_91662/v1/S7pr\\_c.pdf](https://cache.industry.siemens.com/dl/files/107/45531107/att_91662/v1/S7pr_c.pdf).

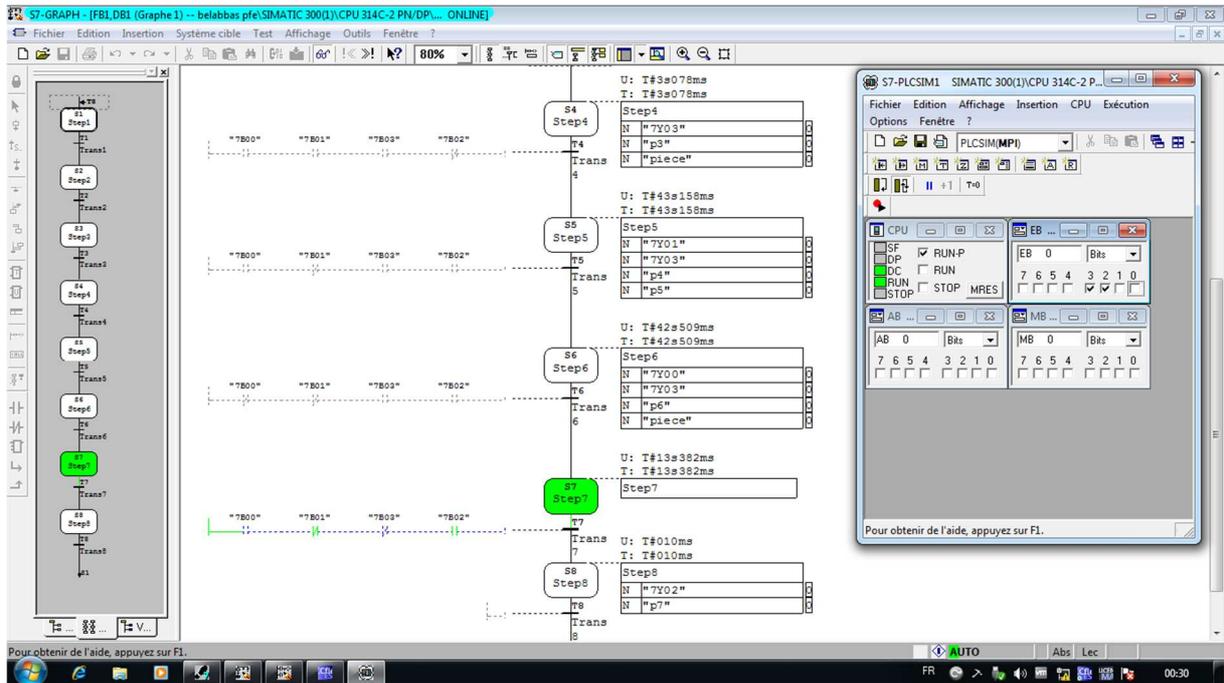
# Annexe



# Annexe A



# Annexe A



# ملخص

هذا العمل تم إجرائه في مختبر الأبحاث بجامعة أبو بكر بلقايد بتلمسان، من أجل وضع برنامج جديد للتحكم في النظام المدرّس، لهذا أجرينا دراسة محاكاة حول هذا الموضوع، لقد وجدنا حلاً لاستبدال اليد العاملة بواسطة التحكم الآلي لقد قمنا بدمج تقنية التحكم التي هي الإشراف و تم تنفيذ بواسطة برنامجين و هما

Simatic manager Step 7 و WinCC Flexible 2008

بعدها قمنا باتباع خطوات إنشاء المشروع

# Résumé

Ce travail a été réalisé au sein de labo de recherche de l'université ABOU BEKR BELKAID de Tlemcen. Afin de mettre en place un nouveau programme de contrôle de système étude, nous avons mené une étude sur ce sujet. Nous avons trouvé une solution pour remplacer complètement l'opérateur humain par un automate programmable industriel, nous avons intégré la technologie de contrôle qui constitue la supervision et la programmation sous logiciel Simatic Manager Step7 et WinCC flexible 2008.

# Abstact

This work was carried out within the research laboratory of ABOU BEKR BELKAID University of Tlemcen. In order to set up a new study system control program, we conducted a simulation study on this topic. We have found a solution to completely replace the human operator by an industrial programmable logic controller: we have integrated the control technology that is the supervision and programming software Simatic Manager Step7 and WinCC flexible 2008.