

République Algérienne Démocratique et Populaire  
Ministère de l'enseignement supérieur et de la recherche scientifique



جامعة بجاية  
Tasdawit n'Bgayet  
Université de Béjaïa

Université Abderrahmane Mira Bejaia

Faculté de technologie

Département de Génie électrique



# Mémoire de fin de cycle

En vue de l'obtention du diplôme de Master en Automatique

*Option* : Automatique et système ; et

Automatique et informatique industrielle

*Thème* :

Migration d'un automate programmable S5 vers S7  
SIEMENS au sein de l'entreprise CEVITAL

Réalisé par :

- BARKOU Yanis
- SLIMANI Mira

Encadré par :

Mr YAHIAOUI Fatah

Membre de jury :

Mr A. CHARIKH

Mme M. GAGAOUA

Promotion : 2019-2020

## REMERCIEMENTS

*Nous tenons à remercier toutes les personnes qui ont contribué de près ou de loin pour que ce modeste travail puisse voire le jour.*

*Aussi nous tenons à témoigner toute notre reconnaissance, et à exprimer nos remerciements les plus distingué :*

*A notre encadreur **Mr YAHIAOUI Fatah**, enseignant à l'université de Bejaia, pour sa patience ; sa disponibilité et surtout ses judicieux conseils, qui ont contribué à alimenter notre réflexion et à l'élaboration de ce projet.*

*A messieurs les membres des jurys pour avoir consacré de leur temps à la lecture de ce manuscrit, et d'accepter de juger et d'évaluer ce travail.*

*A nos chers parents pour leur soutien constant et leur encouragement durant tout notre parcours d'études.*

# DÉDICACES

*Je tiens à dédier ce mémoire :*

*A mon Père, ma mère et à ma très chère grand-mère fatma,  
en*

*Témoignage et en gratitude de leurs dévouement, de leurs  
soutien permanent durant toutes mes années d'études, leurs  
sacrifices illimités, leurs réconfort moral, eux qui ont consenti  
tant d'effort pour mon éducation, mon instruction et pour me  
voir atteindre ce but, pour*

*Tout cela et pour ce qui ne peut être dit, mes affections sans  
limite.*

*A mes frères : Lounis, Sofiane, Ghiles,*

*A mes sœurs : Rosa, Warda, Yasmine, Djamila*

*A toute la famille Chalgou et Slimani sans oublier mes neveux  
Adam et Amir*

*A tous mes amis, en particulier Naim, Yanis, Babina, Salem,  
Taklit, Tina, Siham, Koko, Djedjiga, Hafid*

*Ceux qui m'ont soutenu pendant toute la durée de mes études.*

**SLIMANI MIRA**

# DÉDICACES

*Je dédie ce modeste travail à :*

*A mes parents, karim et Samia. Aucun hommage ne pourrait être à la hauteur de l'amour dont ils ne cessent de me combler. Je leur souhaite une bonne santé et une longue vie.*

*A ma sœur unique : Lysa*

*A mes oncle : Omar, Arezki et Mohammed.*

*A tous mes amis(e)s notamment : Lyes, Hanifa, Rabia, Jiji, Nawel, Thiziri, Mira.*

*A tous ceux qui ont contribué de près ou de loin pour que ce projet soit possible*

**BARKOU YANIS**

# Table des matières

## Table des matières

### Liste des figures

### Liste des tableaux

Introduction générale.....	1
----------------------------	---

## Chapitre 1 : Présentation de l'entreprise CEVITAL

1.1 Introduction.....	3
1.2 Historique de l'entreprise.....	3
1.3 Activités et filiales.....	3
1.3.1 Agro-industrie et distribution.....	4
1.3.2 Automotive, immobilier et sécurité.....	4
1.3.3 Industrie.....	4
1.4 Organisation de l'entreprise Cevital.....	5
1.4.1 Direction générale DG.....	7
1.4.2 Direction de finance et de comptabilité.....	7
1.4.3 Direction commerciale.....	7
1.4.4 Direction logistique.....	7
1.4.5 Direction de contrôle de qualité.....	7
1.4.6 Direction conditionnement.....	7
1.4.7 Direction systèmes informatique.....	8
1.4.8 Direction silos.....	8
1.4.9 Direction margarinerie.....	8
1.4.10 Direction de raffinerie de sucre.....	8
1.4.11 Direction de production de l'huile :.....	8
1.4.12 Direction des ressources humaines(DRH).....	8
1.4.13 Direction de projet.....	9
1.4.14 Direction media-sociale.....	9

1.5 Situation géographique .....	9
1.6 Conclusion.....	10

## **Chapitre 2 : Présentation des APIs**

2.1 Introduction .....	11
2.2 Définition d'un automate programmable.....	11
2.3 Avantages et inconvénients des API .....	12
2.4 Aspect extérieur des API .....	13
2.4.1 Automate programmable compact (centralisé).....	13
2.4.2 Automate programmable modulaire.....	13
2.5 Structure interne des API .....	15
2.5.1 Processeur .....	16
2.5.2 Mémoire.....	16
2.5.3 Alimentation électrique .....	17
2.5.4 Interfaces et cartes d'entrées/sorties.....	17
2.5.5 Bus de communication .....	17
2.6 Langages de programmations pour API .....	18
2.6.1 Langages d programmations graphiques .....	18
2.6.2 Langages de programmation textuels.....	18
2.7 Sécurité .....	19
2.8 Critères de choix des API .....	20
2.9 Conclusion .....	20

## **CHAPITRE 3 : Etude comparative entre l'API S5-95U et S7-300**

3.1 Introduction .....	21
3.2 Présentation de l'automate programmable S5-95U.....	21
3.2.1 Qualités du S5-95U .....	22
3.2.2 Alimentation .....	22
3.2.3 Structure du S5-95U.....	22
3.2.4 Unités fonctionnelles.....	23
3.2.5 Modes de fonctionnement.....	25
3.2.5.1 Mode « STOP ».....	25

3.2.5.2 Mode « RUN »	25
3.2.5.3 Mode « démarrage »	26
3.2.6 Elaboration d'un programme	26
3.2.7 Les différents modes de représentations	26
3.2.8 Cycle de vie de la série S5	27
3.2.9 Solutions pour remplacer SIMATIC S5	28
3.3 Présentation de l'automate programmable SIMENS S7 300	28
3.3.1 Constitution de l'automate S7-300	29
3.3.2 Présentation des différents CPU des automates S7-300	30
3.3.3 Élaboration d'un programme :	31
3.4 Logiciel de programmation STEP 7	31
3.4.1 Définition du logiciel STEP 7 ?	31
3.4.2 Fonctions du logiciel de base STEP 7	32
3.4.3 Langages de programmation sous STEP 7	32
3.4.4 Paramétrage des modules S5/S7	33
3.4.5 Mise en parallèle des blocs STEP 5 et blocs STEP 7	34
3.4.6 Type de données :	36
3.5 Conclusion	37

#### **CHAPITRE4 : Conversion de S5 vers S7**

4.1 Introduction	38
4.2 Explication de la méthode de conversion	38
4.2.1 Lancement de la conversion	38
4.2.2 Exécution de la conversion	40
4.2.3 Conversion de la liste d'assignation	41
4.2.4 Fichier générés	42
4.2.5 Localisation des erreurs	43
4.2.6 Interprétation des messages	44
4.2.6.1 Analyse des messages	44
4.2.6.2 Message d'erreur	44

4.2.6.3 Avertissement .....	48
4.2.7 Retouche du programme convertie .....	50
4.2.7.1 Préparation.....	50
4.2.7.2 Exécution.....	50
4.2.7.3 Modifications d'adresses.....	50
4.2.7.4 Fonctions non convertible .....	51
4. 2.8 Compilation.....	51
4.2.8.1 Vérification de la cohérence .....	51
4.2.8.2 Compilation du fichier source .....	51
4.2.8.3 corrections des erreurs : .....	52
4.3 Conclusion.....	53
Conclusion générale .....	54
Références bibliographiques.....	55

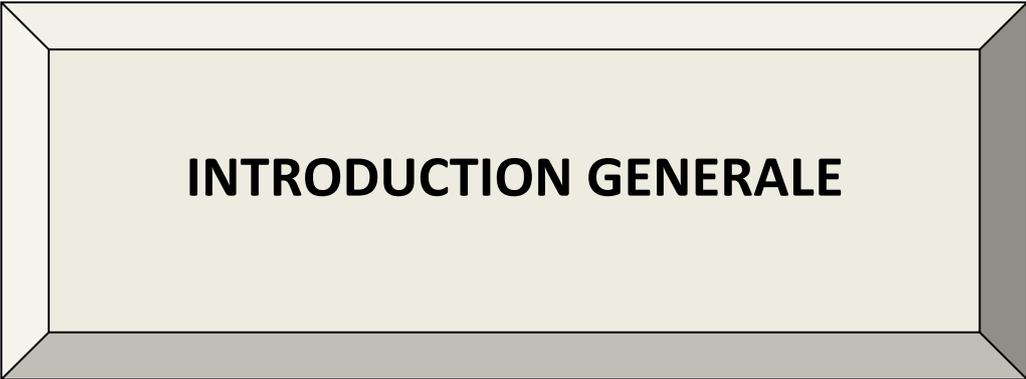
## Liste des figures

Figure 1.1: Principale activités du complexe CEVITAL.....	4
Figure 1.2: Organigramme de cevital.....	6
Figure 1.3: Situation géographique du complexe CEVITAL.....	9
Figure 2.1: Un automate programable industriel.....	12
Figure 2.2: Exemple des APIs compacts.....	13
Figure 2.3: Exepmles des APIs modulaires.....	14
Figure 2.4: Description d'un automate modulaire (siemens).....	14
Figure 2.5: Structure interne d'un API.....	15
Figure 3.1: Automate SIEMENS S5-95U.....	21
Figure 3.2 : Structure de l'API S5.....	22
Figure 3.3: Unité fonctionnelle de l'API S5-95.....	23
Figure 3.4: Cycle de vie de la série S5.....	27
Figure 3.5: Vue de l'automate S7-300.....	28
Figure 3.6: CPU de différent niveau de performe de l'automate S7-300.....	31
Figure 4.1: Image-écran initiale du convertisseur.....	39
Figure 4.2: Boite de dialogue <<conversion de fichier>>.....	40
Figure 4.3 : Première phase de la conersion.....	41
Figure 4.4: Conversion de la liste d'assignation.....	42
Figure 4.5: Message affichés par le convertisseur.....	43
Figure 4.6: Vérification de la cohérence et compilation.....	52



## Liste des tableaux

Tableau 3.1: paramètres des modules S5/S7 .....	34
Tableau 3.2: mise en parallèle des blocs STEP 5 et blocs STEP 7 .....	34
Tableau 3.3: types et données dans S5 et S7 .....	36
Tableau 4.1: messages d'erreur, signification et remède .....	44
Tableau 4.2: avertissement, significations et réactions conseillée .....	48



**INTRODUCTION GENERALE**

## Introduction générale

De nos jours, l'automatisme est le cœur de toutes les installations industrielles. Il remplace une partie ou toute les tâches de coordination, auparavant exécutées par des opérateurs humains, dans un ensemble d'objets techniques appelé partie commande.

Cette dernière est constituée principalement d'automate qui permet de mémoriser le savoir faire des opérateurs pour obtenir la suite des actions à effectuer sur les matières d'œuvre afin d'élaborer la valeur ajoutée.

L'automatisme a, principalement, comme objectif d'assurer les tâches répétitives, autrefois réalisées par les humains, de renforcer la sécurité, d'accroître la productivité, d'économiser les matières premières et l'énergie, et de maintenir la qualité. Ces objectifs peuvent être classés en deux catégories :

- ✓ Les objectifs concernant la compétitivité du produit (coût, qualité, innovation, et disponibilité).
- ✓ Les objectifs concernant l'exploitation de la machine de production (sûreté de fonctionnement, productivité, flexibilité).

Grâce à l'automatisation des systèmes de productions, l'intervention humaine est réduite à la supervision et à la surveillance des différents paramètres des machines pour assurer le bon fonctionnement de la chaîne de production et de réagir en un délai minime en cas de défaillance signalée par le système de gestion des alarmes.

La plupart des entreprises veillent à mettre à la disposition de ses clients un produit compétitif en termes de qualité et de coût, c'est pourquoi leurs ingénieurs et techniciens sont dans l'obligation d'assurer une amélioration continue de tous les équipements intégrant le processus de fabrication du produit souhaité, tout en suivant l'évolution technologique. Et c'est dans cette optique que s'inscrit notre travail, la réalisation d'une migration de l'automate S5 vers S7. Ceci est motivé d'un côté, par le rendement élevé des nouveaux équipements, grâce à la puissance du S7, et d'éviter, de l'autre côté la non disponibilité des pièces de rechange ; étant donné que la firme SIEMENS a arrêté la fabrication des automates S5 en fin d'année 2015.

# Introduction générale

---

Dans notre projet, nous nous sommes énormément intéressés au remplacement de l'automate S5 par le S7, et en d'autres termes à effectuer une migration de S5 vers S7 en exploitant l'outil de migration S5/S7 du logiciel de programmation STEP 7.

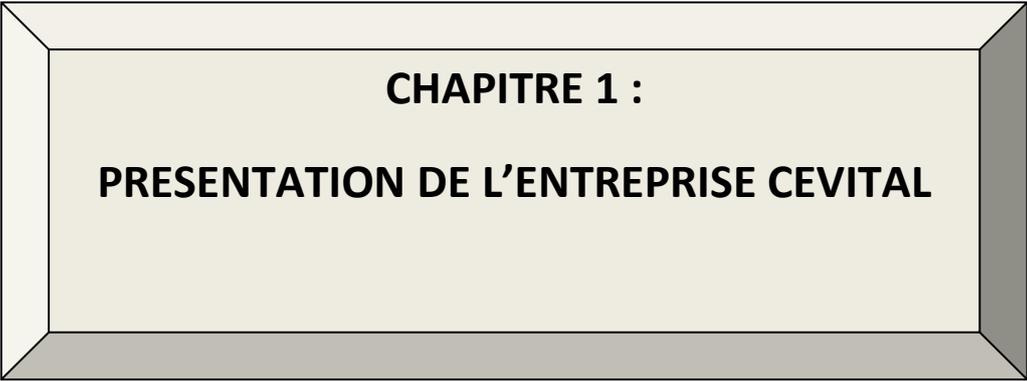
Afin de mieux élucider et éclaircir notre travail, nous avons organisé ce mémoire en une introduction, quatre chapitres et une conclusion.

Le premier chapitre est consacré à la présentation de l'entreprise, et de son organigramme.

Le second chapitre consiste en une présentation des automates programmables industriels APIs, comportant des généralités sur les APIs.

Le troisième chapitre aborde une étude comparative du S5-95U et S7-300, il introduit l'architecture de leurs équipements, leurs ressemblances et différences, et le choix de l'automate.

Le dernier chapitre, quant à lui traite de la migration S5 vers S7 où nous avons démontré avec détail la démarche à suivre pour réaliser la migration souhaitée.



**CHAPITRE 1 :**

**PRESENTATION DE L'ENTREPRISE CEVITAL**

# CHAPITRE 1 : Présentation de l'entreprise CEVITAL

---

## 1.1 Introduction

Dans ce chapitre, nous évoquerons en premier lieu l'évolution historique du complexe agroalimentaire CEVITAL, puis l'explication de ses différentes activités et filiales, ensuite nous présenterons minutieusement l'organigramme de ce gigantesque groupe avec ses nombreuses directions, enfin nous indiquerons sa situation géographique.

## 1.2 Historique de l'entreprise

CEVITAL est une entreprise d'industrie agro-alimentaire. Elle a été créée avec des fonds privés en 1998 sous forme d'une société par action (SPA). Elle est la première entreprise privée algérienne à avoir investi dans des secteurs d'activités diversifiés, elle a traversée d'importantes étapes historiques pour atteindre sa taille et sa notoriété actuelle.

CEVITAL s'est constitué autour de l'idée forte de bâtir un ensemble industriel intégré, concentré en première partie dans le secteur de l'agroalimentaire, et de tant d'autres domaines en l'occurrence : la sidérurgie, l'industrie du verre plat, l'automobile et l'électroménager.

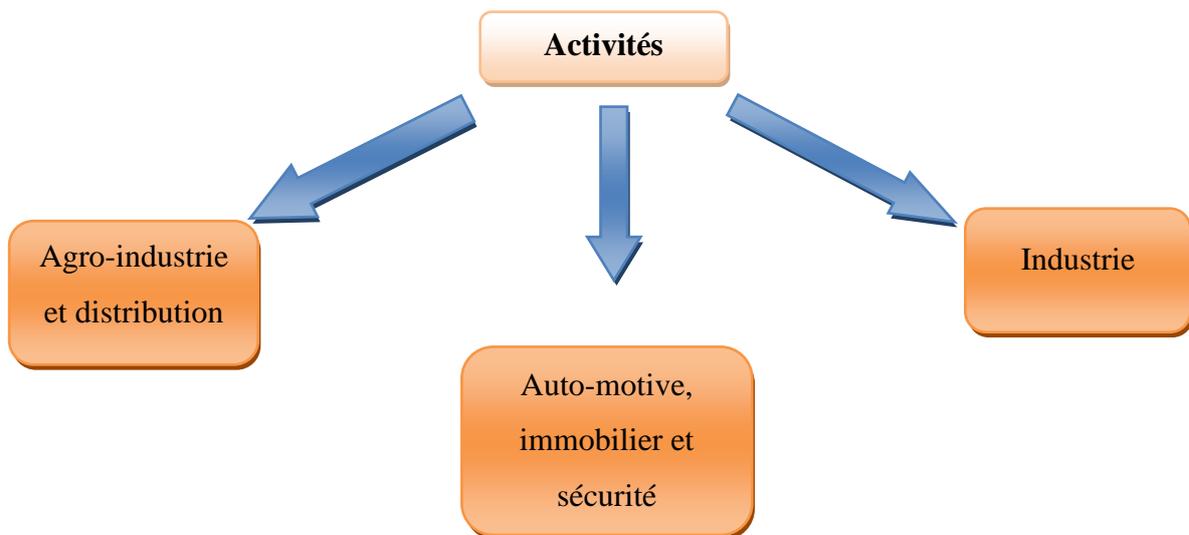
Par ailleurs, Issad Rebrab, fondateur du groupe Cevital, est né en 1944 en Kabylie. Il crée son cabinet d'expert-comptable en 1968, puis se lance des 1971 dans l'entrepreneuriat, en créant des sociétés dans la métallurgie et la sidérurgie. A la tête du groupe CEVITAL, il n'a cessé de le faire grandir en diversifiant ses activités.

ISSAD REBRAB est actuellement classé comme premier [1] milliardaire en Algérie, sixième [1] en Afrique avec un chiffre d'affaire évalué à 4.4 milliard de dollars.

## 1.3 Activités et filiales

L'entreprise CEVITAL entreprend une activité diversifiée, classifiée en trois pôles dont :

- Agro-industrie et distribution ;
- Automotive, immobilier et sécurité ;
- Industrie.



**Figure 1.1:** Principales activités du complexe CEVITAL

### 1.3.1 Agro-industrie et distribution

Ce pôle regroupe plusieurs activités dont la distribution alimentaire et non alimentaire avec la chaîne de magasin UNO, la production agricole avec Ceviagro, la gestion des centres commerciaux avec Sierra Cevital et la production agro-alimentaire qui représente 80% des activités de ce pôle [2]. Ce dernier s'est élargie avec l'intégration des activités <<Food>>, soit la production du sucre, d'huile, de graisses végétales, de céréales via Cevital Agro-Industrie, implanté au sein du port de Bejaia en Algérie.

Leader du secteur de agro-alimentaire en Algérie, la filiale qui propose des produits de qualité à des prix compétitif est composée de plusieurs unités de production qui disposent toutes d'entités de développement chargées d'expérimenter et de lancer de nouveaux produits.

### 1.3.2 Automotive, immobilier et sécurité

Ce pôle intervient dans les secteurs de l'automobile, l'immobilier et l'affichage publicitaire. Parmi ses filiales on trouve HMA (Hyundai Motors Algérie) dans l'automobile. IMMOBIS, spécialisée dans la promotion immobilière et la construction des centres commerciaux et de plates formes logistiques, dans la partie service du groupe.

### 1.3.3 Industrie

Le pôle industrie gère un portefeuille d'activités de plusieurs filiales à savoir :

- Le groupe Brand déploie ses activités dans le domaine de l'Electroménager.

# CHAPITRE 1 : Présentation de l'entreprise CEVITAL

---

- Le groupe Oxxo Algérie Spa déploie ses activités dans le domaine des fenêtres et fermetures. Acquis en 2013 et présente sur le marché Algérien en 2014. Les activités ont été réparties en deux réseaux : Oxxo Bâtiment destiné aux professionnels du Bâtiment, et Oxxo Baies destiné aux particuliers [3].
- La filiale MFG (MediterraneanFloast Glass) est une entreprise spécialisée dans la production, la transformation et la distribution du verre pour la construction, les applications solaires et certaines industries spécialisées [4].
- Baticompos a été acquise par le groupe Cevital en 2007. Héritière d'une longue tradition dans le domaine du bâtiment industrialisé [5].
- La filiale Numilog a été créée en 2012. Elle déploie ses activités autour de trois missions principales [6] :
  - ✓ Accompagner la croissance des activités du groupe Cevital au travers de prestations logistiques et de transport.
  - ✓ Proposer aux acteurs économiques et industriels en Algérie des prestations de transport et/ou logistiques à travers tout le territoire.
  - ✓ Proposer un accompagnement en conseil et solutions logistiques.

## 1.4 Organisation de l'entreprise Cevital

L'organigramme suivant donne une vue générale sur les différents organes constituant le complexe Cevital.

# CHAPITRE 1 : Présentation de l'entreprise CEVITAL

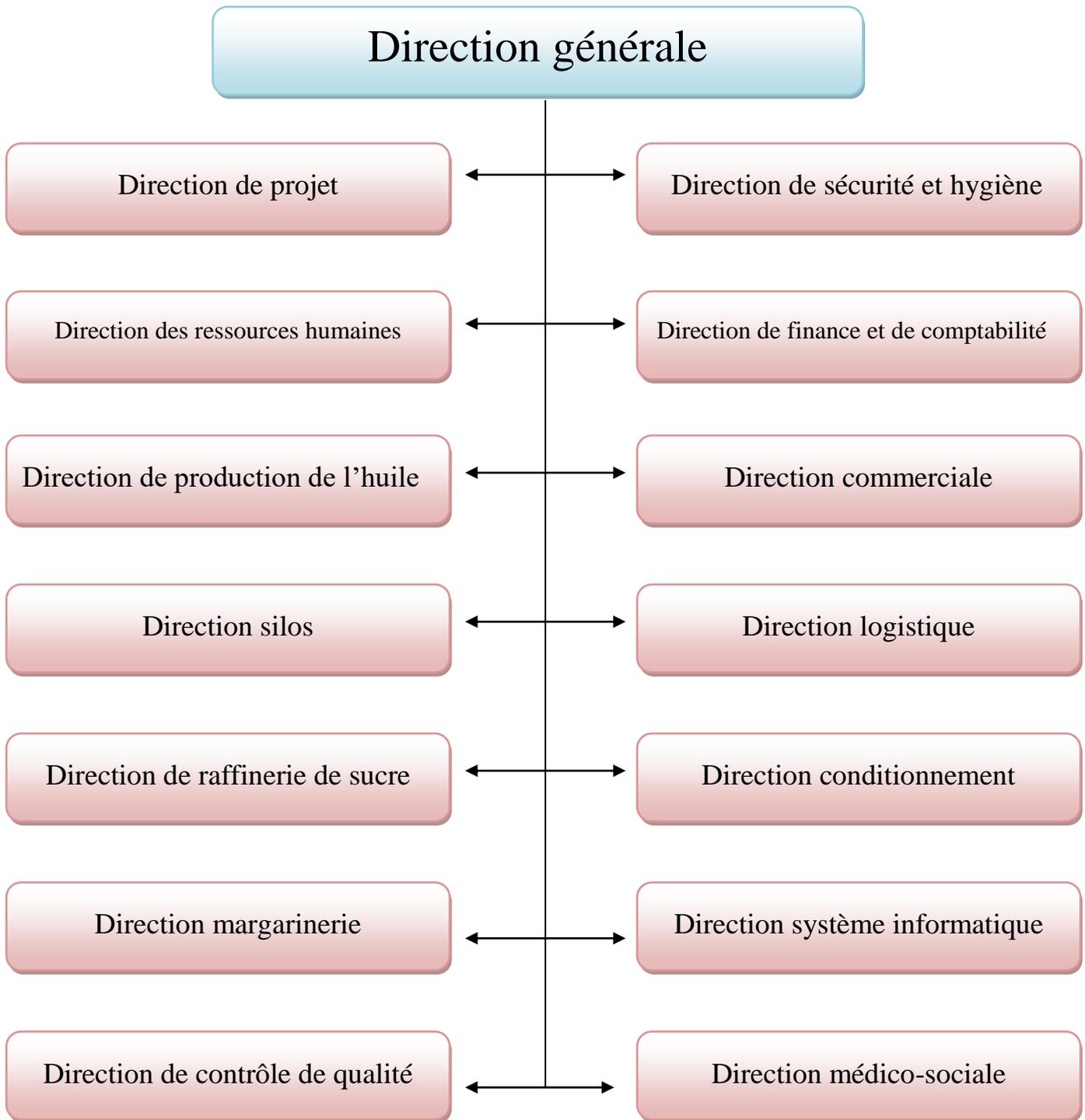


Figure 1.2: Organigramme du complexe Cevital

# CHAPITRE 1 : Présentation de l'entreprise CEVITAL

---

## 1.4.1 Direction générale DG

Elle a pour mission de superviser, contrôler et coordonner les missions des différentes directions. Elle définit la stratégie de développement et veille à la disponibilité des moyens nécessaires au bon fonctionnement de complexe.

## 1.4.2 Direction de finance et de comptabilité

Son rôle est de :

- Préparer et mettre à jour les budgets.
- Gérer les finances et tenir la comptabilité du complexe.
- Pratiquer le contrôle de gestion.
- Faire le Reporting (rapport d'activité selon un format prédéterminé) périodique.

## 1.4.3 Direction commerciale

Ce service s'occupe de :

- L'élaboration de la politique de distribution.
- L'élaboration des stratégies face à la concurrence.
- Gérer les campagnes de promotions et les opérations publicitaires.
- Etablir l'opération avec des organismes liés à l'exploitation.

## 1.4.4 Direction logistique

- Gestion des moyens matériels et humains pour garantir une bonne distribution des produits du complexe
- Assure et gère le transport de tous les produits finis.
- Gère les stocks de produits finis dans les différents dépôts locaux et régionaux.

## 1.4.5 Direction de contrôle de qualité

Elle est dotée de quatre laboratoires qui assurent le suivi permanent et continu de processus de production sous la supervision du laboratoire central qui suit la qualité microbiologique des produits.

## 1.4.6 Direction conditionnement

Chargée de la mise en bouteille de l'huile final, elle se compose de deux ateliers (atelier préforme, atelier bouchant poignets) et la production des bouteilles (1L, 2L, 5L).

# CHAPITRE 1 : Présentation de l'entreprise CEVITAL

---

## 1.4.7 Direction systèmes informatique

Son rôle est d'informatiser les systèmes administratifs et la gestion des ateliers de production.

## 1.4.8 Direction silos

Son rôle est d'assurer le stockage, notamment des céréales, dans les conditions requises des produits entrant dans les processus de production de l'entreprise. Cette direction assure la gestion de trois zones différentes :

- Zone 1 : ou zone quai où s'effectue la réception des navires (déchargement)
- Zone 2 : composé de plusieurs silos et d'un hangar destinés au stockage
- Zone 3 : ou zone des expéditions

## 1.4.9 Direction margarinerie

Chargée de la mise en œuvre du processus de production, cette direction veille au respect des paramètres de production de la margarine.

## 1.4.10 Direction de raffinerie de sucre

S'assure de la mise en œuvre et du pilotage du processus technique de raffinage du sucre. C processus consiste à raffiner le sucre roux, pour extraire du sucre blanc et un résidu dit la mélasse.

## 1.4.11 Direction de production de l'huile :

Elle consiste en exploitation des matières premières pour offrir au marché une huile de qualité.

## 1.4.12 Direction des ressources humaines (DRH)

Elle prend en charge :

- La gestion de la carrière professionnelle des employés.
- Le recrutement, la section des effectifs.
- La formation et le développement du personnel.
- Le règlement des salaires des employés.
- Gestion de la performance et des rémunérations.

# CHAPITRE 1 : Présentation de l'entreprise CEVITAL

## 1.4.13 Direction de projet

Cette direction met en place et intègre de nouveaux équipements industriels. Planifie et assure la maintenance pour l'ensemble des installations. Gère et déploie les projets d'investissement relatifs aux lignes de production, bâtiments et énergie (depuis la définition du processus jusqu'à la mise en route de la ligne ou de l'atelier). Rédige les cahiers de charges en interne. Négocie avec les fournisseurs et les intervenants extérieurs.

## 1.4.14 Direction media-sociale

Elle consiste à veiller sur le déroulement des bonnes conditions sociales et médicales des travailleurs.

## 1.5 Situation géographique

Le complexe Cevital est implanté au niveau du port de Bejaia à 3km du sud-ouest de cette ville, à proximité de la RN26. Cette situation est d'un grand profit en lui conférant l'avantage de proximité économique, en effet elle se trouve proche du port et aéroport de béjaia.



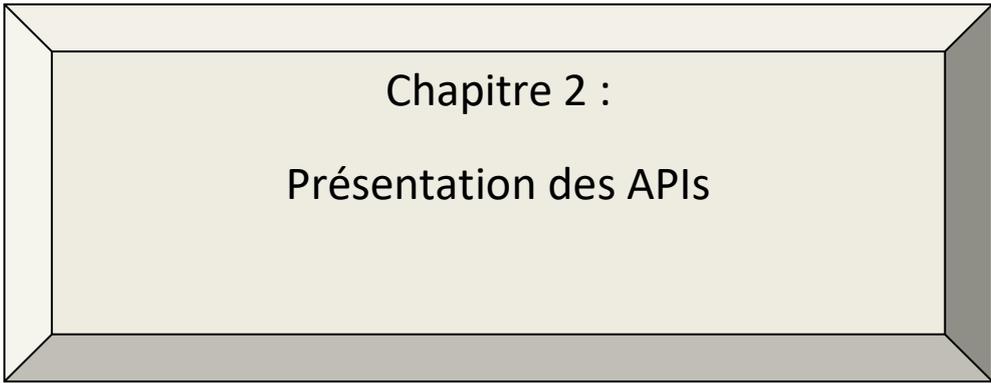
Figure 1.3: Situation géographique du complexe CEVITAL.

# CHAPITRE 1 : Présentation de l'entreprise CEVITAL

---

## 1.6 Conclusion

Ce premier chapitre est consacré à la présentation générale de l'entreprise Cevital. Nous avons eu connaissance grâce à ce chapitre aux activités et aux filiales de cet important complexe et de l'organisation du groupe ainsi de sa situation géographique. Dans le second chapitre nous verrons la présentation des Automates programmables Industriels.



Chapitre 2 :  
Présentation des APIs

### 2.1 Introduction

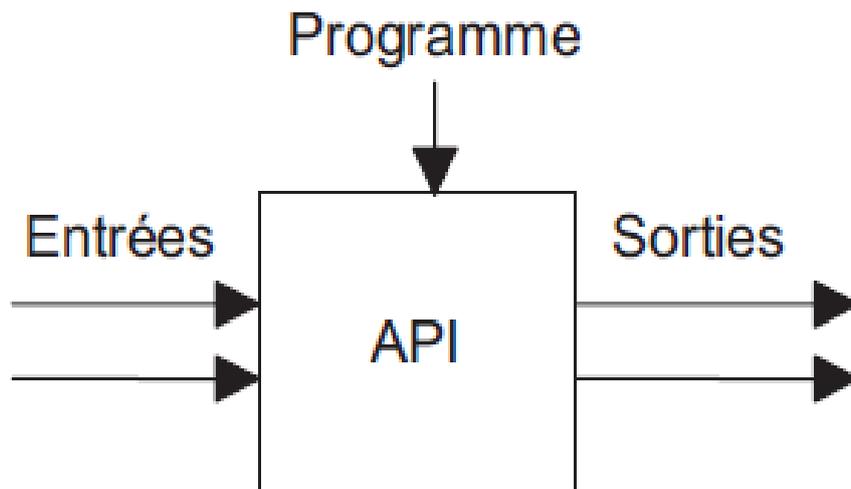
Les automates programmables industriels (APIs), ou en anglais, Programmable Logic Controller (PLC), sont apparus aux Etats-Unis en 1969 [7] pour satisfaire les besoins de l'industrie automobile. Le but recherché était de remplacer la logique à relais câblée et de développer des chaînes de fabrication automatisées qui pourraient suivre l'évolution des techniques et des modèles fabriqués.

Depuis leur apparition, les automates programmables, permettent de se passer des câblages souvent fastidieux et peu flexibles. Grâce aux automates, les opérations de modifications sur des systèmes automatisés deviennent plus faciles et ne requièrent que l'ajout de quelques lignes de code [8].

De nombreux constructeurs d'automates programmables existent comme Omron, Allen-bradley, Schneider, etc., mais la firme allemande SIMENS offre l'une des plus grandes gammes de produits avec toutes ces séries : S5-900/S5U, S7.200, S7.300... [9]. Dans ce chapitre, nous allons présenter brièvement les dispositifs d'APIs.

### 2.2 Définition d'un automate programmable

Un automate programmable industriel (API) est un système électronique fonctionnant de manière numérique, destiné à l'environnement industriel. Il utilise une mémoire de programme pour le stockage interne des instructions pour la mise en œuvre des fonctions spécifiques, telles que : des fonctions logiques, mise en séquence, temporisation, comptage et calcul arithmétique. Il peut commander, au moyen des entrées/sorties (de type tout/rien ou analogique), divers types de machines ou processus. L'API et les périphériques associés sont conçus pour pouvoir facilement s'intégrer à un système d'automatisme industriel et être facilement utilisés dans toutes leurs fonctions prévues [8].



**Figure 2.1:** Un automate programmable industriel.

### 2.3 Avantages et inconvénients des API

Les APIs, avec leur solution programmée, présentent de nombreux avantages par rapport aux dispositifs de commande câblée. Parmi ces intérêts on cite :

- La flexibilité : car le même automate de base peut être employé avec une grande diversité de système de commande. Pour modifier un système de commande et les règles appliqués, un operateur doit simplement saisir une suite d'instruction différente [8].
- La réduction de beaucoup d'espace requis pour l'installation.
- Les composants des APIs sont robustes. Cela lui permet de fonctionner dans des environnements hostiles [7].
- Plus rapides, le système réagit forcément dans le délai fixé.
- Plus compacts, à cause de leurs simplicité en ayant la possibilité de traiter une variétés d'application avec un même automate.
- Plus fiable en raison d'un nombre de composants mécaniques moindre.

En revanche, ils présentent les inconvénients suivants :

- Cherté : dépanadant du nombre d'entrée /sorties nécessaires, de la mémoire dont on veut disposer pour réaliser le programme, de la présence ou non de modules métier [7]
- Suppression d'emplois.

### 2.4 Aspect extérieur des API

Les automates programmables industriels peuvent être de type [8] :

1. Compact ; ou
2. Modulaire.

#### 2.4.1 Automate programmable compact (centralisé)

Il intègre le processeur, l'alimentation, les entrées et les sorties dans un seul boîtier. Selon les modules et les fabricants, il pourra réaliser certaines fonctions supplémentaires (comptage, E/S analogique...) et recevoir des extensions en nombre limité [7].

Ces automates de simple fonctionnement, sont souvent destinés à la commande de petits automatismes [8]



Figure 2.2: Exemples des APIs compacts

#### 2.4.2 Automate programmable modulaire

L'automate programmable se présente comme un ensemble de blocs fonctionnels. Généralement chaque bloc est physiquement réalisé par un module spécifique (coffret, rack, baie ou cartes). Ces différents modules s'articulent autour d'un canal de communication, le bus interne. L'automate programmable est de type modulaire contenant un rack, un module d'alimentation, un processeur, des modules entrées/sorties, des modules de communication et de comptage [7].

Ces automates sont intégrés dans les automatismes complexes ou puissants dont la capacité de traitement et flexibilité sont nécessaires [8].

## CHAPITRE 2 : Présentation des APIs

SIEMENS S7-300	SCHNEIDER TSX 37	MOELLER	SCHNEIDER TSX 57
			

Figure 2.3: Exemples des APIs modulaires.

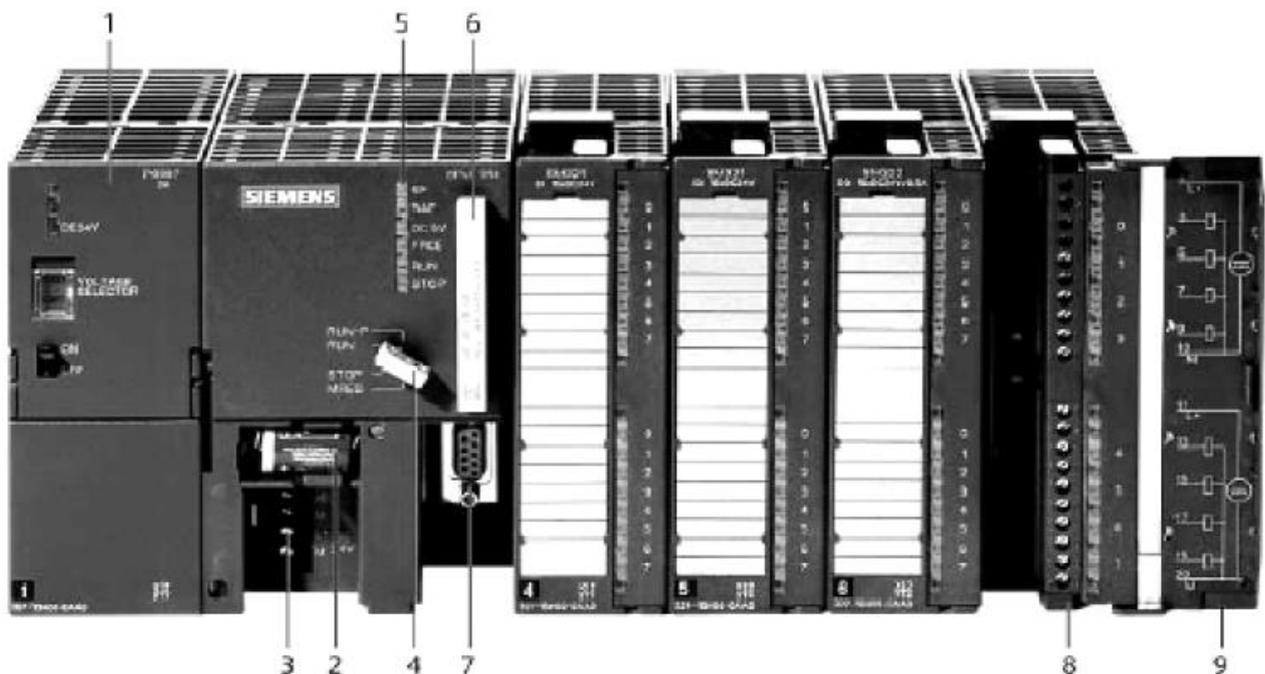


Figure 2.4: Description d'un automate modulaire (siemens) [10]

Signification des différents éléments de l'automate SIEMENS décrit dans la Figure 2.4

1-Module d'alimentation

2-Pile de sauvegarde

3-connexion au 24Vcc

4-Commutateur de mode

5-LED de signalisation d'état et de défaut

6-Carte mémoire

7-Interface multipoint (MPI)

8-Connexion frontal

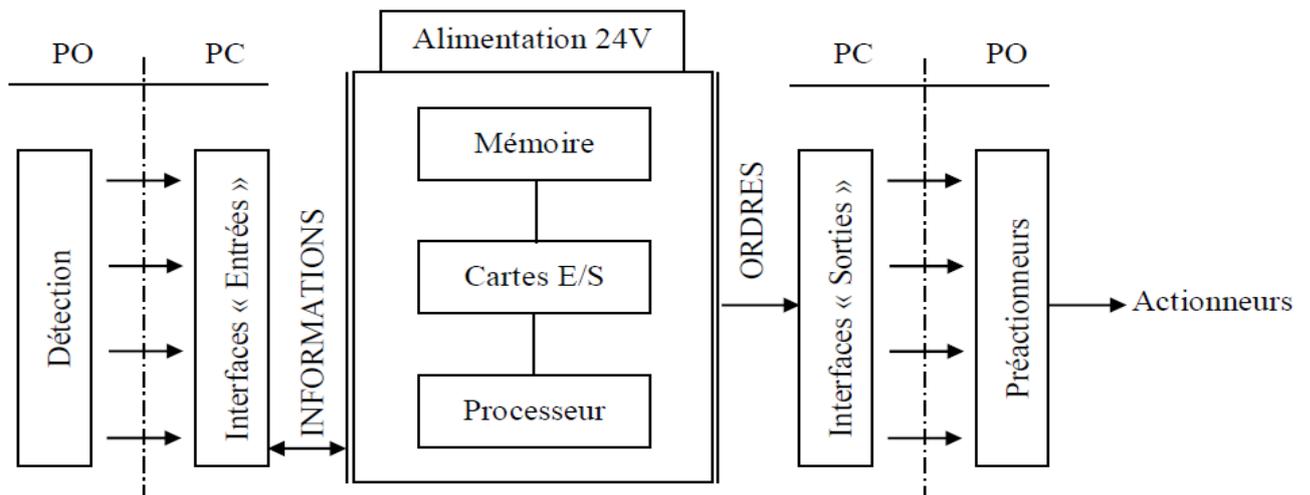
9-Volet en face avant

### 2.5 Structure interne des APIs

La structure interne d'un API comporte quatre parties principales [11]

- Une unité de traitement (processeur CPU) ;
- Une mémoire ;
- Des interfaces d'entrées/sorties ;
- Une alimentation.

Un bus interne est utilisé pour échanger les informations entre les différents éléments de l'automate (entrées, sorties, mémoires)



**Figure 2.5:** Structure interne d'un API

### 2.5.1 Processeur

Le rôle de processeur consiste d'une part à organiser les différentes relations entre la zone mémoire et les interfaces d'E/S et d'autre part à gérer les instructions du programme.

En générale les CPU sont constitués des éléments suivants [9] :

- L'unité arithmétique et logique (UAL) est responsable de la manipulation des données, ainsi que de l'exécution des opérations arithmétiques d'addition et soustraction et des opérations logique.
- La mémoire, sert à stocker les informations nécessaires à l'exécution du programme.
- Une unité de commande est utilisée pour gérer le minutage des opérations.

### 2.5.2 Mémoire

Pour que l'API effectue son travail, il doit accéder aux données à traiter et aux instructions c'est-à-dire au programme, qui lui expliquent comment traiter ces données. Ces informations sont stockées dans la mémoire des APIs qui est composé de plusieurs éléments :

- **Mémoire de programme** : cette mémoire est utilisée pour stocker le programme. Elle est en général le type EEPROM. Une mémoire morte reprogrammable (EPROM) est parfois employée pour stocker de manière permanente les programmes [12] (pour effacer une EPROM, il est nécessaire de disposer d'un effaceur d'EPROM qui projette des rayons UV, la durée d'exposition est de quelques dizaines de minutes.)
- **Mémoire système** : cette mémoire présente, dans le cas d'automates à microprocesseur, est utilisée pour stocker le système d'exploitation et elle est programmée en usine par le constructeur. Elle peut donc sans problème être réalisée en technologie PROM (c'est à dire programmer une seule fois, sans possibilité d'effacement).
- **Mémoire de données** : elle est utilisable en lecture-écriture des données pendant le fonctionnement. C'est une mémoire de type RAM (mémoire vive dans laquelle la lecture et écriture des données sont permises), utilisant une technologie de fabrication CMOS [7].

### 2.5.3 Alimentation électrique

L'alimentation électrique fournit les tensions nécessaires à l'électronique de l'automate à partir des tensions usuelles 210/220 V alternatif ou 24V continu

### 2.5.4 Interfaces et cartes d'entrées/sorties

Interface d'entrées comporte des adresses d'entrées. Chaque capteur est relié à une de ces adresses. L'interface de sortie quant à lui comporte aussi des adresses de sorties, distribuées d'une manière à assurer la liaison de chaque actionneur à l'une de ces adresses. En d'autres termes, les entrées reçoivent des informations en présence des éléments de détection (capteur) et du pupitre operateur (BP, bouton poussoir). Les sorties transmettent des informations aux pré-actionneurs et aux éléments de signalisations du pupitre [7].

Les cartes d'E/S ont une modularité de 8, 16, ou 32 voies [11].

### 2.5.5 Bus de communication

Dans les APIs, il existe quatre types de bus [9].

- ✓ **Bus d'adresse** : sert à transporter les adresses à partir de l'emplacement mémoire vers la CPU, afin que chaque mot puisse être localisé en mémoire. Chaque emplacement possède une adresse unique et chaque emplacement de mot possède une adresse que la CPU utilise pour accéder aux données enregistrées, que ce soit pour la lecture ou l'écriture.
- ✓ **Bus de contrôle** : transporte les signaux utilisé par la CPU pour le contrôle.il sert à informer les dispositifs mémoire s'ils vont recevoir des données à partir d'une entrée ou s'ils vont envoyer des données, et à transmettre les signaux de minutage qui permettent de synchroniser les opérations.
- ✓ **Bus de données** : transportent les données utilisées dans les traitements effectués par la CPU.
- ✓ **Bus système** : sert aux communications entre les ports d'entrées-sorties et l'unité d'entrées-sorties [13].

### 2.6 Langages de programmations pour API

Tous les constructeurs d'automates possèdent leur propre langage de programmation. Cependant, ils proposent tous une interface logicielle répondant à la norme CEI (commission électrotechnique internationale) CEI 1131-3. Cette norme définit cinq langages de programmation [13], qui sont :

1. Les langages graphiques ; et
2. Les langages textuels.

#### 2.6.1 Langages d programmations graphiques

Les langages de programmations graphiques des automates APIs sont en nombre de trois :

- **Grafcet ou SFC (sequential function chart)** : ce langage de programmation de haut niveau permet la programmation aisée de tous les procédés séquentiels [11].
- **Schéma par blocs ou FBD (function block diagram)** ce langage permet de programmer graphiquement à l'aide de blocs, représentant des variables, des operateurs ou des fonctions. Il permet de manipuler tous les types de variable.
- **Schéma à relais ou LD (langage ladder)** : ce langage graphique est essentiellement dédié à la programmation d'équations booléennes (true/false)

#### 2.6.2 Langages de programmation textuels

Les langages de programmations textuels des automates APIs sont en nombre de deux :

- **Texte structuré ou ST (structures/textes)** : ce langage est un langage textuel de haut niveau [9]. Il permet la programmation de tout type d'algorithme plus ou moins complexe.
- **Liste d'instructions ou IL (instruction List)** : ce langage textuel de bas niveau [11], est un langage à une instruction par ligne.

### 2.7 Sécurité

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

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

-Un dysfonctionnement de celui-ci pourrait avoir de graves persécutions sur la sécurité des personnes,

-Les coûts de réparation de l'outil de production sont généralement très élevés,

-Un arrêt de la production peut avoir de lourdes conséquences sur le plan financier,

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

- ✓ Contraintes extérieures : l'automate est conçu pour supporter les différentes contraintes du monde industriel et a été l'objet de tests normalisés.
- ✓ Coupures d'alimentation : l'automate est conçu pour supporter les coupures d'alimentation et permet, par programme, d'assurer un fonctionnement correct lors de la réalimentation.
- ✓ Contrôles cycliques : procédure d'autocontrôle des mémoires, de l'horloge, de la batterie, de la tension d'alimentation et des entrées/sorties.
- ✓ Vérification du temps de scrutation à chaque cycle appelée Watchdog (chien de garde), et enclenchement d'une procédure l'alarme en cas de déplacement de celui-ci (réglé par l'utilisateur).
- ✓ Visualisation : les automates offrent un écran de visualisation où l'on peut voir l'évolution des entrées/sorties (par exemple, les capteurs sont contrôlés en permanence et ne peut indiquer que des états sûrs. Un écrasement du câble créant un court-circuit générant potentiellement un « 1 » permanant n'est pas permis)

On eut également ajouté des modules de sécurité à l'automate (sécurité de machine).

Et il existe enfin des automates dits de sécurité qui intègrent des fonctions de surveillances et de redondance accrues et garantissant la sécurité de matériels.

### 2.8 Critères de choix des APIs

Les critères de choix des APIs sont [7] :

- Le nombre et la nature des E/S
- Les capacités de traitements du processeur, en l'occurrence, la vitesse, les données ; temps réel ...
- Les moyens de dialogue et le langage de programmation.
- Fonctions ou modules spéciaux.
- Les moyens de sauvegarde du programme.

### 2.9 Conclusion

L'API est une prouesse technologique, facile à programmer et à enficher. Il est bien adapté aux conditions industrielles. Il évite à l'homme des taches pénibles et dangereuses.

Dans ce chapitre nous avons présenté d'abord les APIs, ensuite nous avons évoqué avec détails la structure interne des APIs ainsi que leurs langages de programmations, avant de finir par donner les critères de choix des API et les mesures sécuritaires à entreprendre pour protéger l'API en particulier, et tout le système automatisé en général.

### Chapitre 3 :

Etude comparative entre l'API S5-95U et S7-300

### 3.1 Introduction

Actuellement, la plus part les procédés utilisés au sein des entreprises sont automatisés en technologie plus au moins nouvelles. C'est pourquoi, les entreprises, pour des raisons de performances, s'adaptent aux avancées technologiques. La firme SIEMENS fait que l'automate S5-95U est moins performant que les nouvelles séries S7-200/300/400, et suite aussi à la stratégie entreprise par SIEMENS, d'arrêter définitivement la production de la série S5, les entreprises choisissent de faire la migration de S5-95U vers S7-300.

Nous présenterons tout au long de ce chapitre les ressemblances et les différences entre les modules S5 et les modules S7.

### 3.2 Présentation de l'automate programmable S5-95U

L'automate S5-95U est un API de faible encombrement rapide et puissant. Il est conçu pour des applications complexes nécessitant des entrées et des sorties TOR ainsi que des entrées et des sorties analogiques.

Il est adapté à l'exécution des tâches de commande structurées simples demandant des vitesses de réactions élevées et nécessitant des fonctions supplémentaires [14].



**Figure 3.1:** Automate SIEMENS S5-95U [14].

### 3.2.1 Qualités du S5-95U

Les principales qualités de l'automate programmable S5-95U sont :

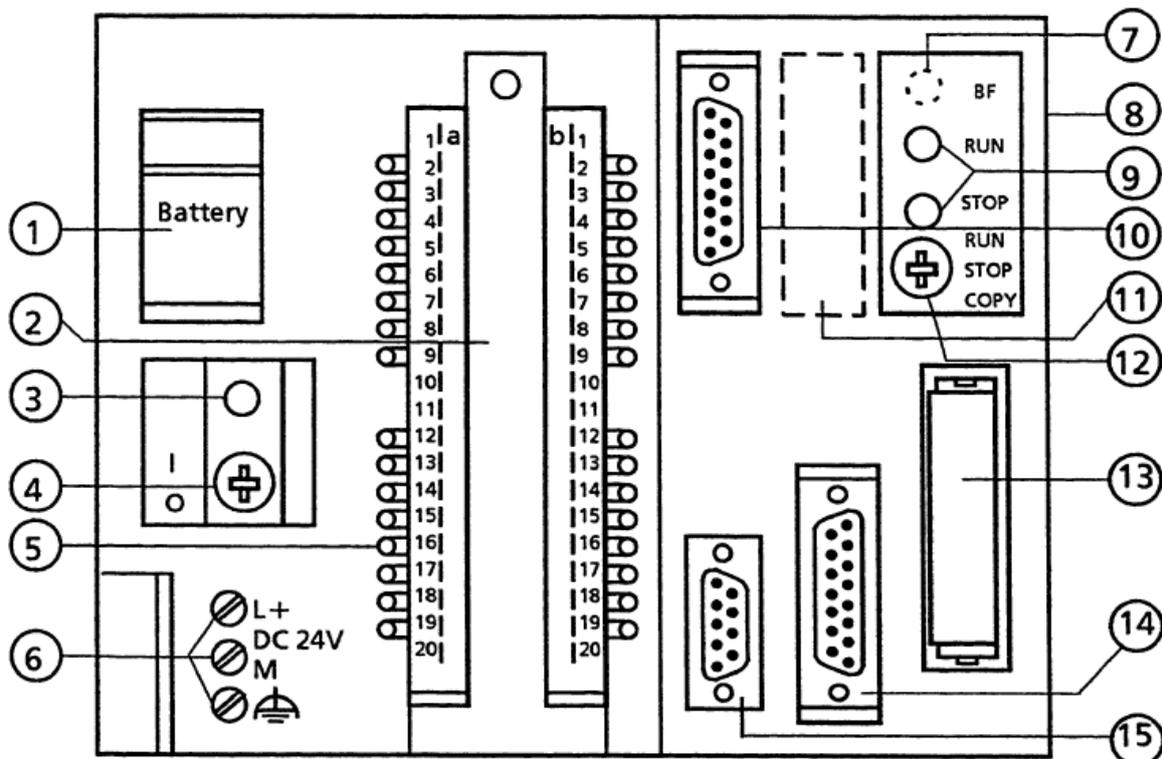
- Entrées/sorties analogiques intégrées à temps de conversion extrêmement court.
- Régulateur PID
- Mise en réseaux sur SINEC L2 en tant que station active ou passive

### 3.2.2 Alimentation

L'automate S5-95U peut être raccordé directement à une tension de 24V en continu, pour le raccordement à la tension secteur 115 /230 V en alternatif, des modules d'alimentation de 1A à 10A (sous 24v) sont placés.

### 3.2.3 Structure du S5-95U

S5-95U : interface, organes de signalisation et de commande



**Figure 3.2:** Structure de l'API S5

- 1- Emplacement de la pile
- 2- Connecteur frontal pour entrées et sorties

## CHAPITRE 3 : Etude comparative entre S5-95U et S7-300

- 3- Signalisation de défaillance de la pile
- 4- Interrupteur marche/arrêt
- 5- LED de signalisation d'état des entrées et sorties TOR
- 6- Bonnes alimentation
- 7- LED de défaut réseau SINEC L2LED de défaut réseau SINEC L2DP
- 8- Connecteurs d'extension pour modèle S5-100U
- 9- Visualisation du mode de fonctionnement LED vert ; RUN ; LED rouge ; stop.
- 10- Interface pour entrées et sorties analogique
- 11- Interface SINEC L2
- 12- Sélecteur de mode
- 13- Emplacement pour cartouche mémoire EEPROM
- 14- Interface pour PG, PC, OP ou réseau SINEC L1
- 15- Interface pour entrées d'alarme et pour entrées du comptage

### 3.2.4 Unités fonctionnelles [14]

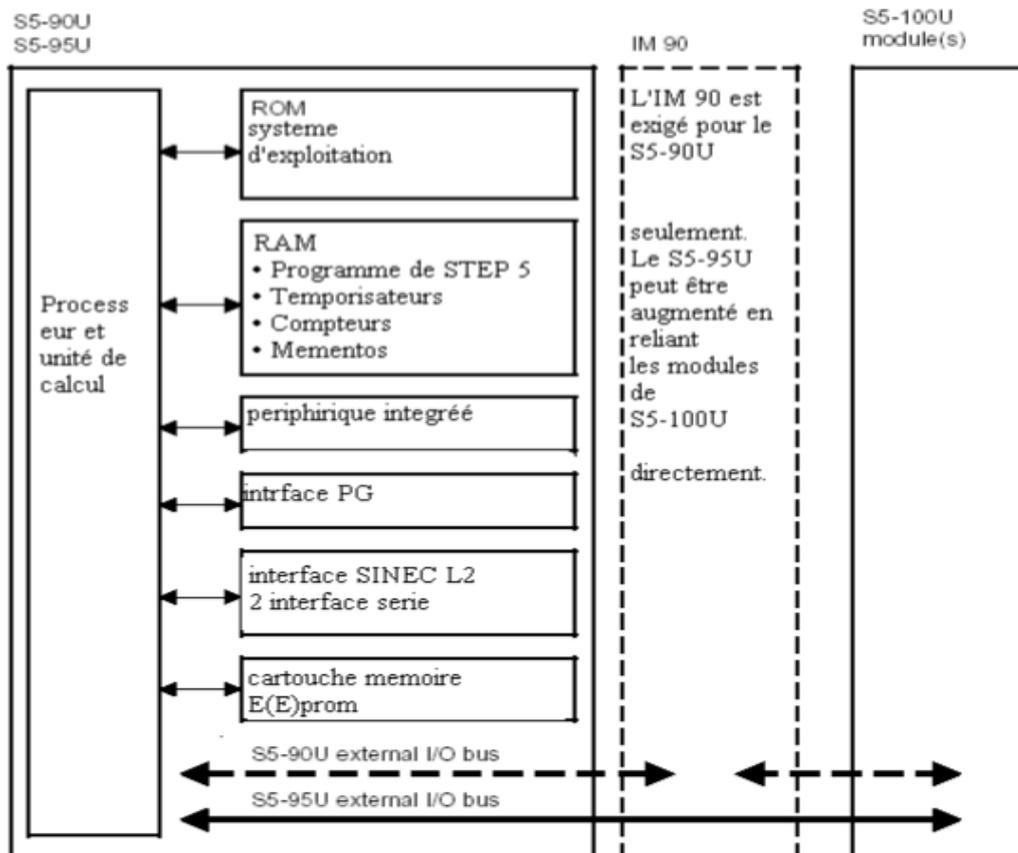


Figure 3.3 : Unité fonctionnelle de l'API S5-95U [14]

## CHAPITRE 3 : Etude comparative entre S5-95U et S7-300

---

### ➤ **Mémoire ROM :**

La mémoire ROM contient le système d'exploitation. Ce derniers est fixe et ne peut être modifié.

### ➤ **Mémoire RAM**

Les variables dynamiques sont stockées dans la RAM, on y trouve :

- Le programme de step 5 et le programme traduit
- La mémoire image de processus et image de processus d'interruption
- Temporisateurs et compteurs ; des temporisateurs sont employés pour la mesure de temps dans la gamme de 10ms à 9990s avec une résolution de 10 ms.
- Donnés système : le système d'exploitation mémorise dans les données système des résultats intermédiaires et certains paramétrages de l'automate
- Mémentos : ils sont utilisés dans le programme step 5 pour mémoriser des résultats intermédiaires. Si l'automate comporte une pile de sauvegarde, les valeurs de certains mémentos sont mémorisées dans la RAM en cas de coupure de tension ou de mise à l'arrêt de l'automate. Il s'agit des mémentos rémanents.

### ➤ **Périphérique intégrée**

La périphérie intégrée est constituée des éléments suivants :

- 16 entrées TOR
- 16 sorties TOR
- 4 entrées d'alarme
- 8 entées analogique
- 1 sortie analogique
- 2 sorties de comptage

### ➤ **Interface PG**

L'interface PG permet de raccorder une console de programmation ou un appareil de commande.

### ➤ **Interface SINEC L2 ou deuxième interface série (en option)**

L'interface permet de raccorder l'automate en tant que station active ou passive à un réseau SINEC L2. Il est ainsi possible de raccorder plusieurs S5-95U entre eux.

### ➤ **Cartouche mémoire E(E) PROM**

## CHAPITRE 3 : Etude comparative entre S5-95U et S7-300

---

L'EPROM ou L'EEPROM sont de types de mémoire non volatile pour la mémoire permanente de programme de gestion.

Si l'automate a subi un effacement général ou si la tension secteur a été coupée alors que l'automate ne contenait pas de pile de sauvegarde, le programme utilisateur sera rechargé automatiquement de la cartouche mémoire dans la mémoire RAM à la prochaine mise sous tension de l'automate.

### ➤ **Bus périphérique**

Le bus périphérique est la liaison électrique entre l'automate et les modules S5-100U servant d'extension de l'automate.

### ➤ **Unité de commande et unité arithmétique et logique**

L'unité de commande exécute les instructions de programme utilisateur. L'unité arithmétique et logique (UAL) est constituée de deux registres appelés accumulateurs (accu1 et accu2) ainsi que pas des bits de signalisation. Tous les calculs arithmétiques sont réalisés dans les accumulateurs. Les bits de signalisation fournissent des informations au sujet des résultats particuliers de certains calculs arithmétiques tels que les dépassements de seuil.

## **3.2.5 Modes de fonctionnement [14]**

### **3.2.5.1 Mode « STOP »**

- Le programme n'est pas exécuté
- Les valeurs de temporisations, compteur, mémentos et les mémoires image des entrées et sorties actives au moment du passage en mode « STOP » sont conservées.
- Les sorties intégrées et les modèles de sortie sont inhibés, c'est-à-dire que les sorties TOR sont à l'état « 0 » et les tensions ou les courants sur les sorties analogique sont nuls.
- Les compteurs intégrés ne comptent pas.
- Lors du passage de « STOP » en « RUN », les mémoires image des entrées et sorties ainsi que les mémentos et les compteurs non rémanents sont remis à zéro.

### **3.2.5.2 Mode « RUN »**

- Le programme est exécuté cycliquement.

## CHAPITRE 3 : Etude comparative entre S5-95U et S7-300

---

- Les temporisations démarrées dans le programme s'écoulent.
- Les états des signaux sont lus sur les entrées intégrés et sur les modules d'entrées (acquisition des entrées).
- Des états des signaux sont affectés aux sorties intégrées et aux modules et aux modules de sorties (émission des sortie).
- Le mode « RUN » peut être choisi après effacement général, c'est-à-dire lorsque la mémoire de programme est vide.

### 3.2.5.3 Mode « démarrage »

- Le système d'exploitation traite le DB1, les paramètres sont validés.
- Les OB (blocs d'organisations) de démarrage OB21 ou OB22 sont traités.
- Le chien de garde n'était pas activé, la durée du démarrage n'est pas limitée.
- Des programmes d'alarme et d'horloge ne peuvent être traités.
- Durant le démarrage, il est possible de lire les entrées intégrées et d'affecter des états aux sorties intégrées à l'aide des opérations L PB /L PW ou T PB/T PW.
- Les modules d'entrées et sorties extrêmes sont inhibés durant le démarrage

### 3.2.6 Elaboration d'un programme

Les automates programmables accomplissent des taches d'automatisation traduites sous forme de programme utilisateur, pour que l'automate puisse le comprendre, ce dernier doit être écrit dans un langage déterminé et suivant des règles bien définie. la firme SIEMENS a développé le STEP 5 pour la famille SIMATIC 5. STEP 5 est un langage facilement compréhensible permettant d'écrire aisément, de manière flexible et peu couteuse des programmes [14].

### 3.2.7 Les différents modes de représentations [14]

Le langage de programmation STEP 5 est unique pour tous les appareils de la gamme SIMACTIC S5. Le programme peut être introduit sous forme de :

- **Liste d'instruction (LIST)** : il est sous forme de suite d'abréviation d'instruction
- **Logigramme (LOG)** : il représente les fonctions logiques à l'aide de symboles graphiques

## CHAPITRE 3 : Etude comparative entre S5-95U et S7-300

- **Schéma à contacts (CONT)** : il représente les fonctions logiques de commande à l'aide des symboles graphiques des schémas électriques.
- **Graph 5** : ce mode de représentation sert à décrire la structure des automatismes séquentiels. Mais ce mode de représentation ne peut être utilisé que de manière limitée dans le S5-95U [14].

### 3.2.8 Cycle de vie de la série S5

La figure suivante nous montre clairement la stratégie de la firme SIMENS pour stopper définitivement l'API S5 du marché mondial. L'annonce de l'arrêt de commercialisation des automates de série S5 était au 01/10/2002. L'arrêt de fourniture de S5-90/95/100 était le 01/10/2013 et 01/10/2014 pour le S5-115/, le 01/10/2015 pour S5-135/155 [16].

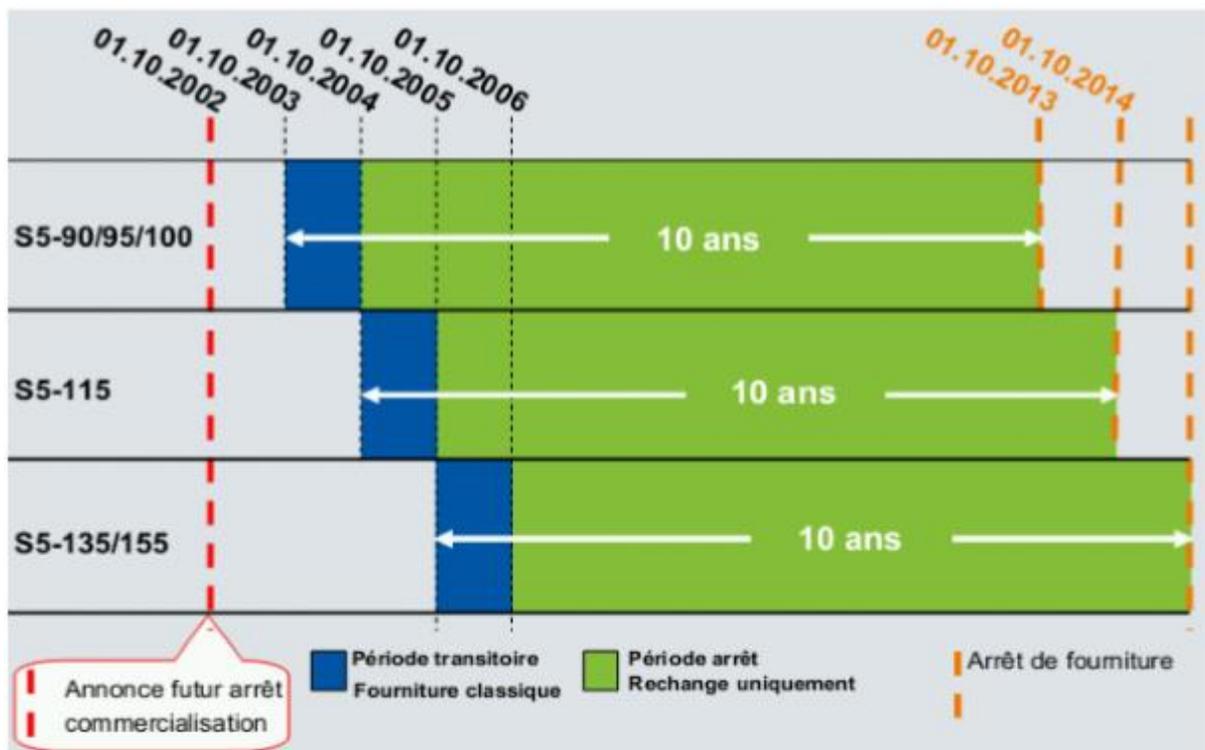


Figure 3.4: Cycle de vie de la série S5 [17].

### 3.2.9 Solutions pour remplacer SIMATIC S5

Après plusieurs années aux services des lignes de production industrielle, les automates SIMATIC S5 évoluent pour plus de performances. La nouvelle gamme S7, lancée en 1996 [17], apporte une intégration idéale avec les interfaces homme-machine, ainsi qu'un atelier logiciel beaucoup plus convivial.

Dans la plupart des cas, la possibilité serait de changer le module d'entrée-sortie, tout en gardant l'unité centrale de traitement de S5 puisque habituellement l'entrée/sortie est le plus grand investissement sur une machine, mais cette stratégie est à écarter, pour un système réduit (100 U ou 95U), la meilleure approche serait de remplacer le système entier et réaliser une programmation en se basant sur le programme existant et un cahier des charges.

### 3.3 Présentation de l'automate programmable SIMENS S7-300

L'API S7-300 est de conception modulaire, il est composé de plusieurs éléments, comme l'indique la figure ci-dessous

- CPU
- Un module d'alimentation
- Modules de signaux
- Coupleurs IM
- Modules de communication (CP)
- Module de fonction (FM)
- Module de simulation [18]

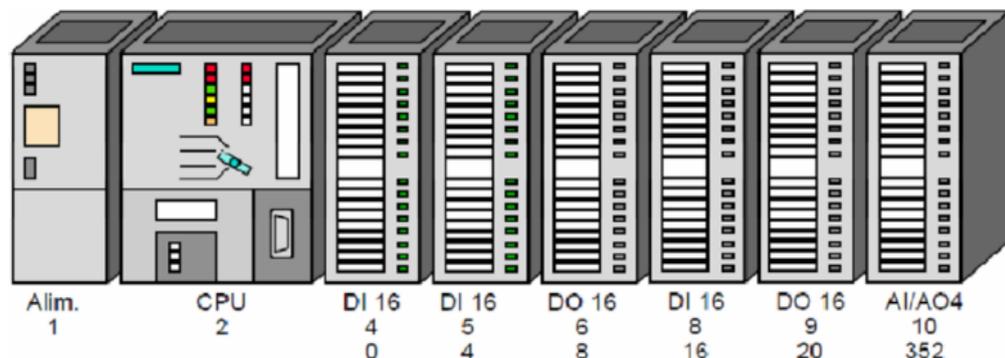


Figure 3.5 : Vue de l'automate S7-300

L'automate S7-300, plus performant et plus puissant, facile à manier, a les caractéristiques suivantes : [19]

- Souplesse d'utilisation grâce à des architectures décentralisées simple et aux multiples possibilités de mise en réseau.
- Facilité et confort d'utilisation grâce à une configuration simple
- Evolutivité permettant l'intégration des nouvelles tâches
- Haut niveau de performance procuré par les nombreuses fonctions intégrées

### 3.3.1 Constitution de l'automate S7-300 [20]

Il est constitué des éléments suivants :

- **Module d'alimentation** : délivré sous une tension de 24V, un courant de sortie assigné de 2, 5,10 A
- **Unité centrale** : c'est le cerveau de l'automate qui exécute le programme utilisateur et qui commande les sorties
- **La pile** : permet de sauvegarder le contenu de la RAM en cas de coupure de courant
- **Carte mémoire** : la plupart des CPU possèdent une carte mémoire. Son rôle est de sauvegarder le programme utilisateur, le système d'exploitation et les paramètres qui déterminent le comportement de la CPU et des modules en cas de coupure du courant
- **Module d'entrées** : provenant soit de la part des capteurs ou du pupitre de commande.
- **Module de sortie** : permettant de raccorder avec les différents pré-actionneurs, ainsi qu'avec les actionneurs.
- **Module coupleur(IM)** : c'est un coupleur qui permet la configuration multi rangée des S7-300 et assuré la liaison entre les châssis et le couplage entre les différentes unités.
- **Modules de fonction(F1)** : assure des tâches lourdes en calcul ainsi des fonctions spéciale comme le positionnement, la régulation, le comptage, la commande numérique ... etc.
- **Processeur de communication(CP)** : permet la communication entre plusieurs automates. Ils permettent d'établir les liaisons homme-machine qui sont effectuées

## CHAPITRE 3 : Etude comparative entre S5-95U et S7-300

par les interfaces de communications point à point, profibus , et l'Ethernet industriel.

- **Le rais profilé** : constitue le châssis de S7-300
- **Interface(MPI)** : une liaison MPI (multi point interface) est nécessaire pour programmer un SIMATIC S7-300 depuis le PC.
- **interface PROFIBUS DP** : les CPU possèdent deux interfaces disposent de l'interface PROFIBUS DP pour la connexion au réseau PROFIBUS DP.

### 3.3.2 Présentation des différents CPU des automates S7-300 [16]

Les API S7-300 sont dotés de différentes CPU selon l'exigence de l'industrie. On trouve les CPU 312 IFM, 313,314, 314 IFM, 315,315-2 DP et 316,312 ptp , 313-C2DP...etc.

- **CPU S7-312 ptp :**

C'est une CPU compacte disposant d'entrées/sorties TOR intégrées et d'une deuxième interface série, elles sont conçues pour des installations ayant des exigences élevées concernant la puissance de traitement et la vitesse de réaction.

- **CPU S7-312 :**

C'est une CPU compacte à entrées/sortie TOR intégrées pour de petites applications à exigences élevées concernant la puissance de traitement.

- **CPU S7-313 -C2 DP :**

C'est une CPU compacte disposant d'entrées/sorties TOR intégrées et d'une interface maitre/esclave PROFIBUS DP.

- Pourvue de fonction technologique des taches à fonction spéciales.
- Pour le raccordement de périphérie décentralisée.

- **CPU S7-313C :** est une CPU compacte à entrées/sorties TOR et analogiques intégrées conçue :

- Pour les installations ayant des exigences élevées concernant les puissances de traitement et la vitesse de la réaction.
- Pourvue de fonctions technologiques.



CPU S7-312 ptp



CPU S7-312



CPU S7-313C



CPU S7-313-C2DP

**Figure 3.6:** CPU de différent niveau de performe de l'automate S7-300 [20]

### 3.3.3 Élaboration d'un programme :

Les automates S7-300 doivent être écrits dans un langage déterminé et suivant des règles bien définies. La firme SIEMENS a développé le STEP 7 pour la famille SIMATIC7.

## 3.4 Logiciel de programmation STEP 7

### 3.4.1 Définition du logiciel STEP 7

STEP 7 est le logiciel de base pour la configuration et de la programmation des systèmes d'automatisation SIMATIC. Il fait partie de l'industrie logicielle SIMATIC [21].

Le logiciel de base STEP 7 existe en plusieurs versions :

- STEP 7-micro / DDS et STEP 7-micro / Win pour des applications autonomes simples sur SIMATIC S7-200

## CHAPITRE 3 : Etude comparative entre S5-95U et S7-300

- STEP 7 pour des applications sur SIMATIC S7-300/400, SIMATIC M7-300/400 et SIMATIC C7 présentant des fonctionnalités supplémentaires [21] :
  - Possibilité d'extension grâce aux applications proposées par l'industrie logicielle SIMATIC.
  - Possibilité de paramétrage de modules fonctionnels et de modules de communication
  - Forçage et fonctionnement multiprocesseur
  - Communication par données globales.
  - Transfert de données commandées par événement à l'aide de blocs de communication et de blocs fonctionnels.
  - Configuration de liaisons

### 3.4.2 Fonctions du logiciel de base STEP 7 [21] :

Le logiciel nous assiste dans toutes les phases du processus de création de nos solutions d'automatisation, comme par exemple :

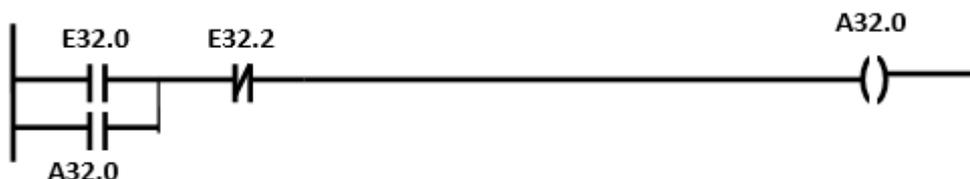
- ✓ La création et la gestion de projet
- ✓ La configuration et le paramétrage du matériel et de la communication
- ✓ La gestion de mnémoniques,
- ✓ La création de programme, par exemple pour les systèmes cible S7,
- ✓ Le changement de programme dans des systèmes cible,
- ✓ Le diagnostic lors de perturbations de l'installation,
- ✓ Le test d'installation d'automatisation.

### 3.4.3 Langages de programmation sous STEP 7

L'écriture des programmes sous STEP 7 est possible avec différents langages qui sont :

- **Le langage à contact (CONT) :**

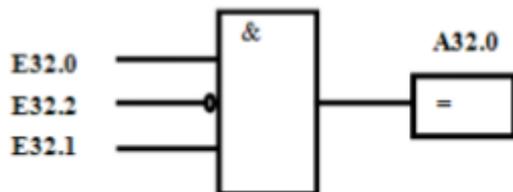
Dans ce langage, la représentation est fondée sur des schémas à relais. Les éléments d'un tel schéma, par exemple les contacts à ouverture ou fermeture sont reliés pour former des réseaux.



## CHAPITRE 3 : Etude comparative entre S5-95U et S7-300

- **Le logigramme (LOG) :**

Le langage de programmation LOG utilise les pavés logiques bien connus dans l'algèbre booléenne pour la représentation logique. Il permet en outre de représenter des fonctions complexes, telles que les fonctions mathématiques en les mettant directement en liaison avec les pavés logiques.



- **Le langage en liste d'instruction (LIST) :**

C'est un langage de programmation textuel proche de la machine. Dans cette programmation, les différentes instructions correspondent, dans une large mesure aux étapes par lesquelles la CPU traite le programme. Pour faciliter la programmation, LIST a été complété par quelques structures de langage évoluées (comme par exemple, des paramètres de blocs et accès structuré aux données) [21].



### 3.4.4 Paramétrage des modules S5/S7

Le paramétrage des modules SIMATIC S5 et SIMATIC S7 est expliqué dans le tableau suivant :

## CHAPITRE 3 : Etude comparative entre S5-95U et S7-300

**Tableau 3.1:** Paramètres des modules S5/S7 [16]

SIMATIC S5	SIMATIC S7
/	Disposition des modules dans l'application de configuration matérielle (HW config) de step 7
Réglage des adresses à l'aide des commutateurs DIL	STEP 7 vous assiste dans l'enfichage des modules et choisit automatiquement les adresses
Réglage du mode de fonctionnement à l'aide de commutateurs	L'application de configuration matérielle de STEP 7 se charge du paramétrage des modules
Paramétrage du mode de fonctionnement des unités centrales par les zone de données système par exemple DB1/DX0	Le CPU est paramétrée dans HW config
/	Chargement après compilation des données de configuration dans la CPU et transmission automatique des paramètres aux modules à la mise en route

### 3.4.5 Mise en parallèle des blocs STEP 5 et blocs STEP 7

Le tableau suivant énumère les blocs STEP 5 et les blocs STEP 7 en parallèle montrant à chaque fois la correspondance entre ceux indiqués quand un bloc S5 doit être remplacé par un bloc ou une fonction STEP 7 équivalente.

**Tableau 3.2:** Mise en parallèle des blocs STEP 5 et blocs STEP 7 [16]

Bloc STEP 5	Bloc STEP 7	Explication
Bloc d'organisation(OB)	Bloc d'organisation(OB)	Interface aux systèmes d'exploitation

## CHAPITRE 3 : Etude comparative entre S5-95U et S7-300

OB spéciaux intégrés	Fonction système(SFC) bloc fonctionnels système(SFB)	Les fonctions systèmes de STEP 7 remplacent les blocs d'organisation spéciaux de STEP 5 pouvant être appelés dans le programme utilisateur
Bloc fonctionnel (FB, FX)	Fonction(FC)	Les fonctions(FC) de step 7 ont les mêmes propriétés que les blocs fonctionnels de STEP 5
Bloc de programme(PB)	Bloc fonctionnel(FB)	Les blocs de programme ont leur équivalent dans STEP 7 : on les appelle des blocs fonctionnels. A la différence de leurs homologues dans STEP 5, ils possèdent des propriétés nouvelles et ouvrent de nouvelles perspectives en matière de programmation.  Attention : les blocs de programme sont convertis en fonction STEP 7(FC)
Bloc séquentiel(SB)	/	Les blocs séquentiels n'existent plus dans STEP 7
Bloc des données DX0, DB1 dans leur fonction spéciale	Blocs de données systèmes (SDB)  Paramétrage(CPU)	Les nouveaux blocs de données de configuration matérielle ainsi que les paramétrages de CPU nécessaires à l'exécution du programme

## CHAPITRE 3 : Etude comparative entre S5-95U et S7-300

Bloc de données (DB,DX)	Blocs de données (DB)	Les blocs de données sont plus longs dans STEP 5 (jusqu'à 8 kilo octets pour le S7-300)
Blocs de commentaire DK,DKX ,FK,FKX ,PK	/	Les blocs de commentaire n'existent plus dans STEP7.le commentaire est contenu dans les blocs correspondants dans la base de données hors ligne

Les possibilités qu'offre le nouvel univers de blocs ne pouvant toutes être traitées ici, ce tableau a uniquement énumérés quelques blocs les plus importants et est destiné à nous aider dans notre choix de blocs STEP7

### 3.4.6 Type de données :

STEP 7 utilise de nouveaux types de données. On peut voir les types de données S5 et S7 en parallèle dans le tableau ci-après

**Tableau 3.3:** types et données dans S5 et S7 [16]

Types de données dans S5	Types des données S7	Classe
BOOL, BYTE, WORD, DWORD,	BOOL, BYTE, WORD, DWORD, INT, DINT, REAL, SSTIME, TIME,   DATE ;TIME-OF- DAY, CHAR,	Types de données simples

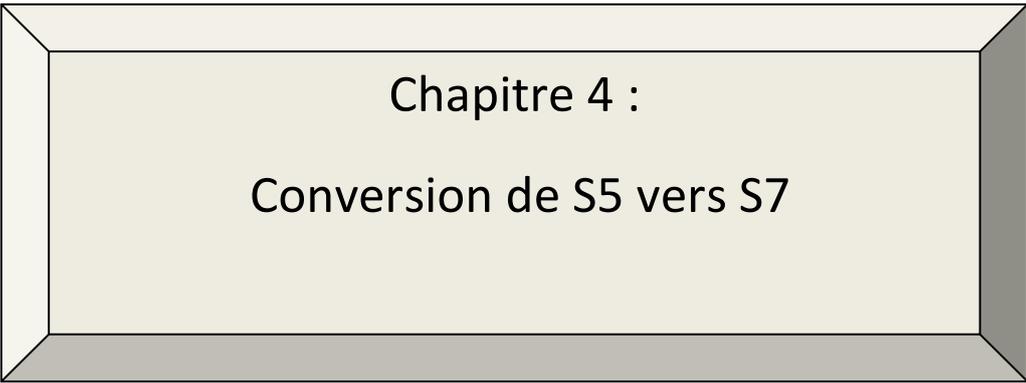
## CHAPITRE 3 : Etude comparative entre S5-95U et S7-300

/	DATE.END-TIME, STRING, ARRAY, STRUCT,	Types de données complexes
Temporisation, Compteur, Blocs	TIMER, COUNTER, BLOCK-FC, BLOCK-FP, BLOCK-DP, BLOCK-SDP POINTER, ANY	Types de paramètre

### 3.5 Conclusion

Dans ce chapitre, on a focalisé notre étude sur l'API S5-95U, le S7-300 et les différents logiciels, step5 et step7. En premier lieu, nous avons étudié avec attention les qualités et les performances de l'automate S5-95U, ensuite celles de l'automate S7-300, et nous avons constaté que ce dernier est plus performant et plus flexible que le premier automate.

Nous démontrerons dans le chapitre suivant comment effectué la migration de S5 vers S7.



Chapitre 4 :  
Conversion de S5 vers S7

### 4.1 Introduction

Dans le monde industriel, la plupart des anciennes installations utilise l'automate S5-90U /S5-95U qui est un automate conçu pour des applications de faible ou de moyenne envergure [14]. Il répond aux exigences que l'on peut attendre d'un automate programmable. Cependant, l'arrivée des nouvelles séries de SIEMENS (S7 200/300/400, S1200) a fait que les anciens automates tels que SIEMENS S5 ne répondent plus à l'évolution des besoins des industries de point de vue : fiabilité, performance et disponibilité des pièces de rechange. Par conséquent, une migration du modèle S5 vers S7 s'impose.

### 4.2 Explication de la méthode de conversion

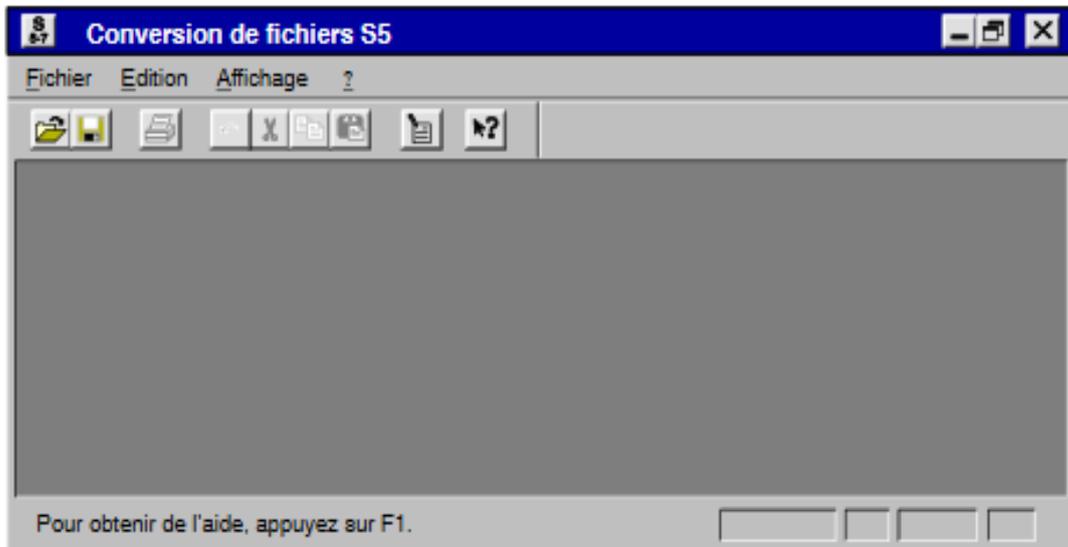
Dans ce chapitre nous avons opté, pour faire la conversion de S5 vers S7, d'utiliser le convertisseur S5/S7. La programmation S7 en LIST, CONT, LOG est compatible avec, respectivement, LIST, CONT et LOG de S5. Par conséquent, si nous sommes des utilisateurs de S5 et que nous souhaitons mettre en œuvre dans S7 des programmes existants, la conversion sera très facile. Nous pouvons rester fideles à nos programmes en S5, il suffit de les convertir en programmes S7 [16].

Le programme existant avec S5 peut être converti au S7 avec un convertisseur intégré dans le STEP 7. L'application convertisseur S5/S7 nous propose de convertir, si possible de façon complète, le jeu d'instruction des programmes S5 existants en programmes S7. La conversion des programmes S5 existants se fait toujours en LIST [16].

#### 4.2.1 Lancement de la conversion

Avant de lancer la conversion, il faut s'assurer que le fichier S5 à convertir, la liste des références croisées et éventuellement la liste d'assignation se retrouve bien dans le même répertoire.

Après avoir sélectionné l'option « Simatic, STEP7, convention de fichier S5 ». Le convertisseur S5/S7 se présente avec l'image-écran ci-après.



**Figure 4.1:** Image-écran initiale du convertisseur

La procédure complète de la conversion est la suivante :

➤ **Préparation des fichiers S5 à convertir**

Pour préparer la conversion, il faut copier les fichiers suivants dans un répertoire DOS :

- Fichier programme : <Nom>ST.S5D (le programme S5 à convertir)
- List de références croisées : <Nom>XR.INI

La liste croisée est nécessaire afin de conserver la structure et la hiérarchie d'appel du programme S5.

Si nous désirons convertir également le fichier des mnémoniques nous avons besoin, en outre, du fichier

- Liste d'assignation : <Nom>ZO.SEQ.

Tous les fichiers générés lors de la conversion sont archivés dans ce répertoire.

➤ **Sélection d'un fichier programme**

On procède comme suit pour sélectionner un fichier programme :

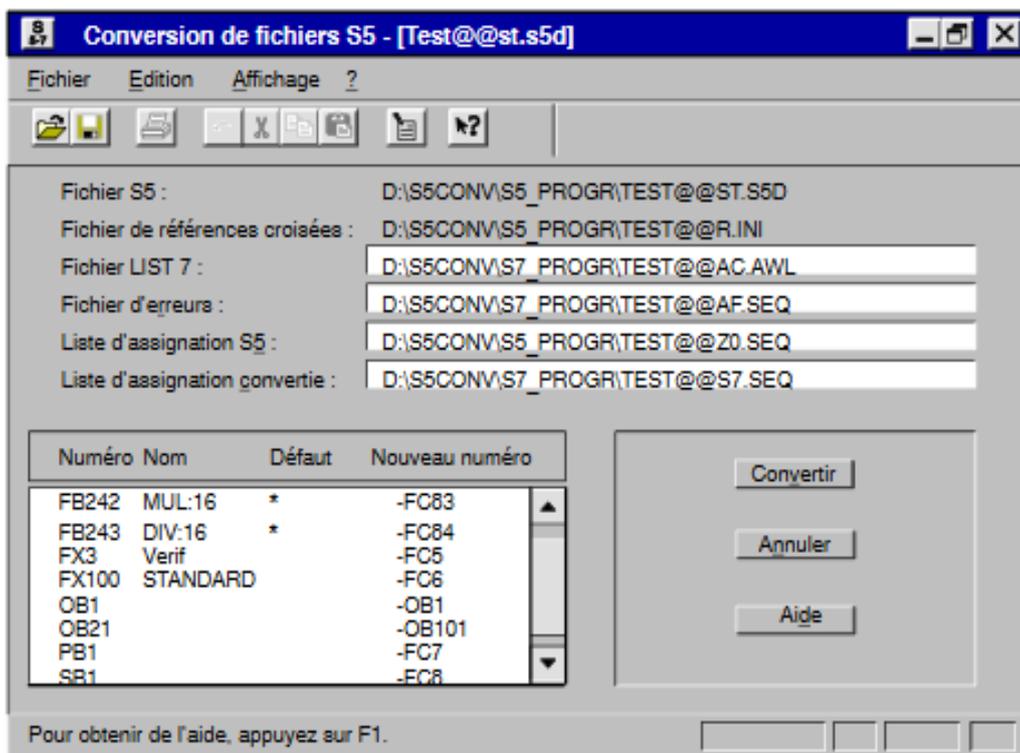
- 1- Exécuter la commande **Fichier>Ouvrir**

## CHAPITRE 4 : Conversion de S5 vers S7

- 2- Sélectionner le lecteur et le répertoire dans lesquels se trouvent les fichiers à convertir
- 3- Sélectionner le fichier à convertir et cliquer sur **OK**.

**Résultat :** le convertisseur S5/S7 affiche les fichiers source et cible ainsi qu'une table de correspondance des anciens et nouveaux numéros des blocs.

La figure suivante présente la boîte de dialogue « Conversion de fichier S5 [<nom> ST.S5D] »



**Figure 4.2:** Boîte de dialogue«<conversion de fichier>>

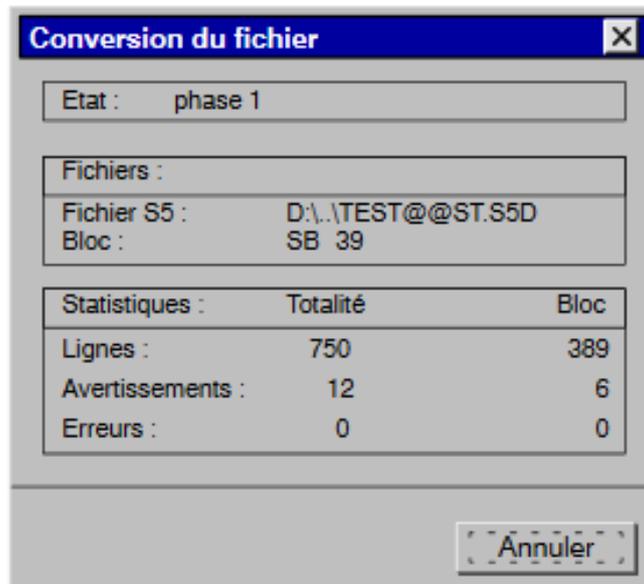
### 4.2.2 Exécution de la conversion

L'exécution de la conversion est lancée via le bouton « Convertir », nous mettons en route la procédure de conversion. Elle se compose de deux phases :

- 1- La première est celle de la conversion
- 2- La seconde celle de la transposition de la liste d'assignation [12]

## CHAPITRE 4 : Conversion de S5 vers S7

Au cours de la première phase, dite phase de conversion, le programme S5 est converti en un fichier source S5 avec tous les blocs et tous les commentaires source S5.



**Figure 4.3:** Première phase de la conversion

Au cours de la seconde phase, le fichier source S5 est converti en un fichier source LIST avec les nouveaux numéros de blocs et la syntaxe de S7.

### 4.2.3 Conversion de la liste d'assignation

Au cours de la conversion de la liste d'assignation, les mnémoniques de la liste d'assignation de S5 sont converties en un format que l'éditeur mnémonique peut importer.

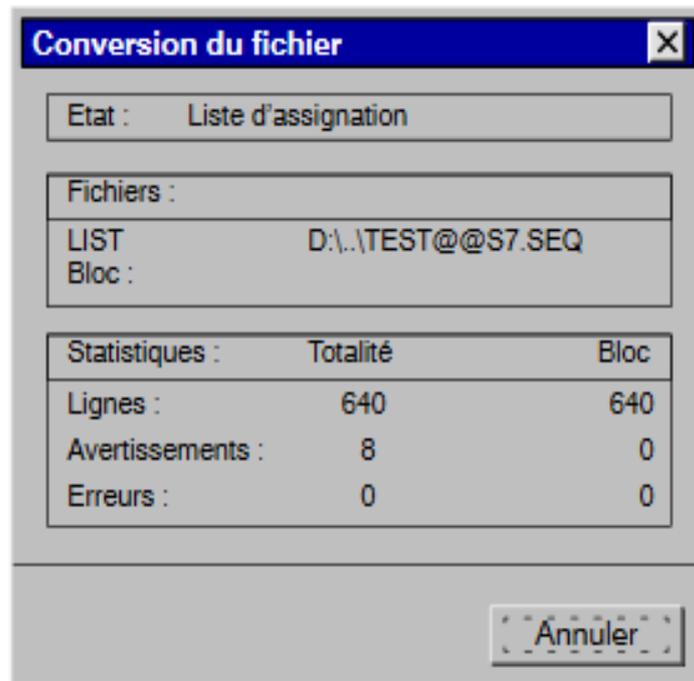


Figure 4.4 : Conversion de la liste d'assignation

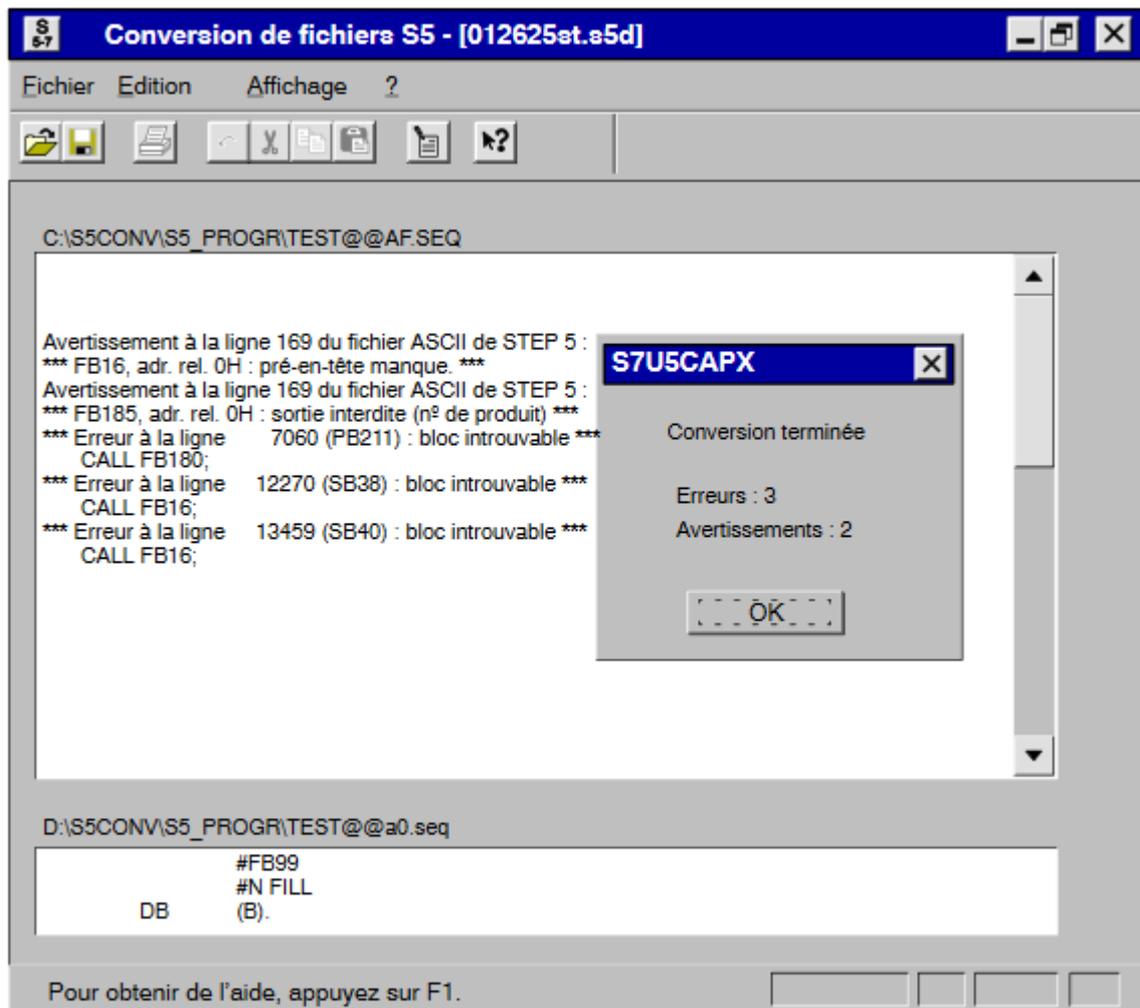
### 4.2.4 Fichier générés [16]

Le convertisseur S5/S7 génère les fichiers suivants lors de la conversion :

- **Fichier<nom>AD.SEQ** : ce fichier est créé pendant la première phase de la conversion. Il contient le fichier <nom>ST.S5D sous forme ASCII.
- **Fichier<nom>AC.AWL** : ce fichier est créé pendant la seconde phase de conversion. Il contient le programme LIST. De cette seconde phase peuvent également provenir des messages résultants de définitions incorrectes de macro-instruction.
- **Fichier<nom>S7.SEQ** : ce fichier est créé lors de la transposition de la liste d'assignation. Il contient la liste d'assignation convertie en un format que l'éditeur de mnémoniques peut importer.
- **Fichier d'erreurs<nom>AF.SEQ** : ce fichier, affiché dans la partie supérieure de la fenêtre « conversion de fichiers S5 », contient les erreurs et les avertissements figurant dans le programme converti. Ces messages sont générés pendant la première et la seconde phase de conversion et pendant la transposition de la liste d'assignation.

## CHAPITRE 4 : Conversion de S5 vers S7

La conversion une fois terminée, une boîte indiquant le nombre d'erreurs et d'avertissement s'affiche.



**Figure 4.5** : Message affichés par le convertisseur

### 4.2.5 Localisation des erreurs

Dans la zone inférieure de la fenêtre « conversion de fichiers S5 ».

Nous pouvons visualiser dans le fichier concerné la position à la quelle l'erreur s'est produite.

Le fichier source LIST mentionne aussi les messages du convertisseur aux endroits du programme où des erreurs ont été constatées. En outre, il contient des avertissements ou des observations au sujet des problèmes qui peuvent se présenter (par suite de modifications de la sémantique des opérations, par exemple)

### 4.2.6 Interprétation des messages

#### 4.2.6.1 Analyse des messages

Parmi les messages du convertisseur, on distingue les messages d'erreurs et les avertissements. On procède comme suit pour analyser ces messages :

- Visualiser dans la zone inférieure de la fenêtre « Messages » le fichier dans lequel l'erreur s'est produite
- Consulter l'aide en ligne pour comprendre la signification du message
- Corriger l'erreur comme il est proposé.

#### 4.2.6.2 Message d'erreur

Un message d'erreur est émis quand une partie du programme S5 n'est pas convertible et ne peut figurer qu'en tant que commentaire dans le programme S7.

Le tableau suivant dresse la liste de tous les messages d'erreur avec les significations et les mesures à prendre pour remédier à l'erreur.

**Tableau 4.1:** Messages d'erreur, signification et remède [16]

Message d'erreur	Origine	signification	Remède
Paramètre absolue diverge de l'identificateur d'opérande	Phase 1	L'identification d'opérande n'est pas correcte	Vérifier l'instruction
Bloc introuvable	Phase 1	Le bloc appelé (FB,FX) manque ou il figure dans la liste des blocs mais n'existe pas dans le fichier programme	Verifier la structure du programme

## CHAPITRE 4 : Conversion de S5 vers S7

	Phase 2	Un bloc est appelé qui n'existe pas dans le fichier programme	Vérifier que la liste de références croisées a bien été indiquée lors de la conversion ou contrôler la structure du programme
La commande n'est pas autorisée dans ce bloc	Phase 1	Saut à l'intérieur d'un bloc de programme, par exemple	Vérifier l'instruction
Commande non définie	Phase 1	L'instruction MC5/LIST n'est pas valable	Corriger le fichier programme S5
	Phase 2	L'instruction n'existe pas dans S7	Etudier une macro-instruction ou remplacez l'instruction par la séquence d'instructions de S7 appropriée.
L'accès par bits au compteur / à la temporisation n'est pas possible. veuillez vérifier	Phase 2	Le programme S5 contient des accès par bits à des temporisations et à des compteurs	Vérifier le programme LIST
CALL OB n'est pas autorisé	Phase 2	L'appel de blocs d'organisation n'est pas autorisé dans S7	Le cas échéant, utiliser l'instruction CALL SFC
CALL SFCxy a été généré, veuillez compléter la liste	Phase 2	Des paramètres SFC manquent	Compléter la liste des paramètres SFC

## CHAPITRE 4 : Conversion de S5 vers S7

des paramètres			
Fichier introuvable	Globale	Le fichier sélectionné n'existe pas	Vérifier le fichier de programme
Profondeur d'imbrication incorrecte	Phase 1	Toutes les parenthèses ne sont pas correctement fermées	Respecter les niveaux de parenthèses, éliminer l'erreur de programmation
Opérande incorrect	Phase 1	L'opérande ne convient pas à l'opération	Vérifier la source S5
	Phase 2	L'opérande ne convient pas à l'opération	Modifier le fichier LIST
Erreur de conversion	Phase 2	Opération BI sans constate	Compléter l'opération de changement par une constante
Erreur dans le fichier macro, XY non pris en compte	Phase 2	Erreur de macro-instruction	Vérifier la macro-instruction
Paramètre formel non défini	Phase 1	Il y a plus de paramètres que dans le bloc appelant	Vérifier le fichier programme S5
Fichier ou répertoire introuvable	Phase 1	Le fichier programme ne contient aucun bloc	Vérifier le fichier programme
Longueur de commentaire incorrecte	Phase 1	Erreur dans le fichier S5	Vérifier le fichier programme
Commentaire trop long	Phase 1	Erreur dans le fichier S5	Vérifier le fichier programme
Aucun nom de bloc spécifié	Phase 1	Le nom de bloc ne comporte que des espaces	Entrer un nom de bloc
Droits d'accès manquants	Globale	Le fichier est protégé en écriture	Supprimer la protection en écriture
Marque non définie	Phase 1	Le repère de saut n'est pas défini dans l'étiquette	Vérifier le fichier S5

## CHAPITRE 4 : Conversion de S5 vers S7

Marque incorrecte	Phase 1	Le repère de saut contient des caractères non valables	Vérifier le fichier S5
Opérateur incorrect	Phase 1	L'opérateur dans le fichier S5 est inconnu ou impossible à convertir	Remplacer l'opérateur par l'opération S7 appropriée
Opérateur incorrect, peut éventuellement être remplacé par l'instruction : <code>\\ ""L P# paramètre formel \\ ""</code>	Phase 2	L'opérateur ne peut pas être chargé sous cette forme dans S7	Utiliser éventuellement l'instruction indiquée
Nombre de paramètres incorrect	Phase 1	Erreur dans le programme S5	Vérifier le fichier programme
Paramètre erroné	Phase 1	Erreur dans le programme S5	Vérifier le fichier programme
Type de paramètre incorrect	Phase 1	Erreur dans le programme S5	Vérifier le fichier programme
Erreur d'écriture disquette	Globale	Le fichier est protégé en écriture ou il n'y a plus de place sur la disquette	Supprimer la protection en écriture ou effacez les données dont vous n'avez pas besoin
Débordement de la mémoire dans la PG (problèmes de place)	Phase 1	La mémoire centrale est insuffisante	Effacer la mémoire centrale les fichiers dont vous n'avez plus besoin
Le repère de saut ne peut être généré	Phase 2	L'opération SPR dépasse la limite du bloc	Éliminer l'erreur dans le programme S5
Un code MC5 incorrect a été converti	Phase 1	Conversion d'une ancienne opération de S5	Aucun

## CHAPITRE 4 : Conversion de S5 vers S7

### 4.2.6.3 Avertissement

Un avertissement est émis quand une partie du programme S5 est certes convertie, mais qu'il faudrait en vérifier la validité.

Le tableau suivant dresse la liste des avertissements avec les significations et les réactions conseillées.

**Tableau 4.2:** Avertissement, significations et réactions conseillée [16]

Avertissement	Origine	signification	Réaction
Version incorrecte (no de produit)	Phase 1	Un bloc fonctionnel standard de S5 doit être remplacé par un FC de S7	Aucune
Version incorrecte (bloc GRAPH 5)	Phase 1	Les blocs GRAPH 5 ne sont pas convertibles	Utiliser éventuellement un bloc créé avec GRAPH 7 pour S7
Vérifier la base de temps choisie	Phase 2	Dans S7, la base de temps peut être plus serrée que dans S5	Paramétrer la base de temps à l'aide de l'application « configuration matérielle »
I/D n'influence que l'accu 1 L qui est maintenant l'accu 1	Phase 2	Les accumulateurs de S7 sont étendus à 32 bits	Examiner les conséquences d'une opération indirecte d'incrémentatation ou de décrémentation dans le programme LIST
Tenir compte de la nouvelle numérotation de blocs	Phase 2	L'appel indirect de bloc ne tient pas compte des nouveaux numéros de bloc (le numéro et prélève dans le mot de memento ou de données appropriées)	Modifier la logique dans S5 ou utiliser des appels de blocs fixes
OB 23 et OB 24 sont convertis	Phase 2	Les OB 23 et OB 24 sont remplacés tout deux par	Regrouper le contenu des OB23 et OB24 dans

## CHAPITRE 4 : Conversion de S5 vers S7

en OB 122		L'OB 122 dans S7	un OB122 et effacer l'autre OB122
L'OB a été interprété par l'AG115U comme un OB23	phase 2	Selon la CPU employée, L'OB34 peut avoir des significations différentes	Vérifier que cet OB convient ou programme
Masque DB S5 n'est plus utilisé pour le paramétrage de S7	Phase 1	Il y a MASK dans DWO et dans DW1	Paramétrer l'AP avec step 7
L'opération du saut suivant l'opération "B" ne peut être convertie (utilisé, SPL)	Phase 2	Une opération de substitution suivie d'une opération SPA ne peut être convertie automatiquement	Remplacer l'opération par SPL dans le fichier LIST et renvoyer le saut
Le convertisseur ne définit pas les paramètres système	Phase 2	DB1 et DX0 sont convertis mais ils n'ont plus la même signification que dans S5	Effectuer le paramétrage du système dans la table de configuration
Tenir compte des différentes opérations d'arrêt	Phase 2	Il n'est pas fait de différence entre STP, STS, et STW	Verifier le fichier programme
RLG est mis à 1	Phase 2	Pour les opérations SU et RU de S5, le RLG est mis à 1 dans S7	Ajouter au besoin l'opération CLEAR
Pré-en-tête manque	Phase 1	Pour les blocs FB et FX, les désignations des repères de saut manquent. Pour les blocs DB et DX, ce	Examiner si les étiquettes se trouvent dans un autre fichier

## CHAPITRE 4 : Conversion de S5 vers S7

		sont les formats de données qui manquent	
Dans le cas d'un AG115U, changer le en OB 100	Phase 2	L'OB21 de mise en route de S5 est converti automatiquement en OB101	Si le programme S5 était exécuté sur un automate S5-115U, il faut transformer l'OB101 en OB100

### 4.2.7 Retouche du programme convertie

#### 4.2.7.1 Préparation

Les étapes de préparations suivantes sont nécessaires avant de retoucher le fichier source LIST généré.

- Imprimer les messages
- Créer un programme S7 dans un projet via le gestionnaire de projets SIMATIC
- Importer à l'aide de la commande **Insertion>Source externe** le programme source LIST généré dans le classeur "source" du programme S7 qu'on a créé.
- ouvrir le fichier converti.

#### 4.2.7.2 Exécution

On retouche le fichier source LIST généré de la manière suivante :

- Parcourir le programme en mode interactif et modifier ou compléter les blocs d'organisation et les opérations S5 non convertibles en s'aidant des messages émis.

#### 4.2.7.3 Modifications d'adresses

Ce sont surtout les modules d'entrées et de sorties qui sont touchés par les modifications d'adresses. On trouve l'adresse des modules avec l'application config

On a la possibilité de modifier les adresses suivantes :

- Réassignation dans S5
- Réassignation dans S7
- Modifications d'adresses dans le fichier source

- Création d'un nouveau fichier source

### 4.2.7.4 Fonctions non convertible

Les opérande et les opérations non convertibles figurent sous forme de commentaires dans le programme S7 généré, on doit procéder a leur retouche.

Il existe deux méthodes différentes de conversion qui sont proposées :

- On définit pour ces opérations, s'ils apparaissent dans le programme utilisation des macro-instructions spécifiques en LIST S7 qui seront utilisées lors de la conversion
- On édite les séquences d'instructions correspondantes dans le programme S7 résultant

le choix de la méthode dépend de la fréquence d'apparition de telle commande dans notre programme utilisateur

### 4.2.8 Compilation

On doit compiler le programme converti et éventuellement retouché avec le compilateur LIST afin de le rendre exécutable. On procède comme suit :

#### 4.2.8.1 Vérification de la cohérence

La commande **Fichier>Vérifier** la cohérence nous permet de vérifier à tout moment la syntaxe et la cohérence du fichier source sans toutefois déclencher la génération des blocs.

La vérification porte sur :

- La syntaxe ;
- Les mnémoniques ;
- L'existence des blocs appelés dans le programme.

On obtient ensuite un protocole indiquant le nom du fichier compilé, le nombre de lignes compilés ainsi que le nombre d'erreurs et d'avertissement

#### 4.2.8.2 Compilation du fichier source

La commande **Fichier>Compile** nous permet de compiler notre fichier source afin de générer les blocs correspondants.

## CHAPITRE 4 : Conversion de S5 vers S7

Un protocole s'affiche après la compilation, il indique les erreurs éventuelles comme après la vérification de cohérence. Si nous avons programmé plusieurs blocs dans un fichier source, seuls ceux sans erreur sont compilés et sauvegardés.

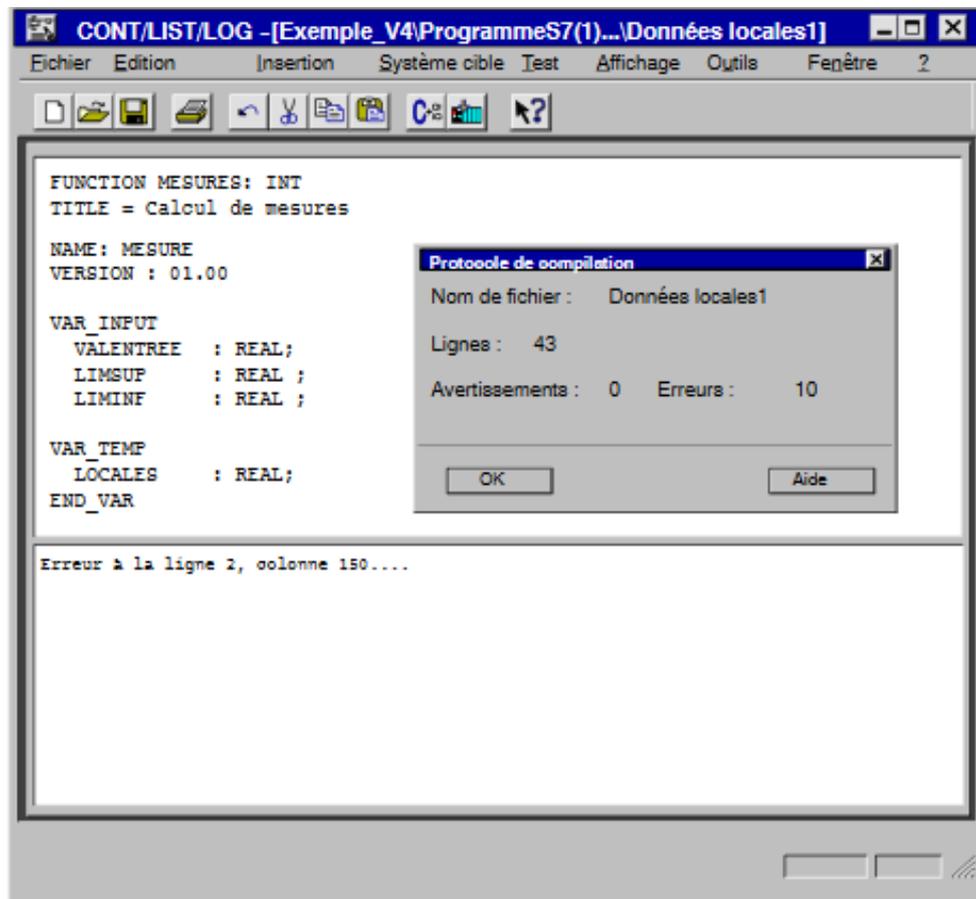


Figure 4.6 : vérification de la cohérence et compilation de fichier source

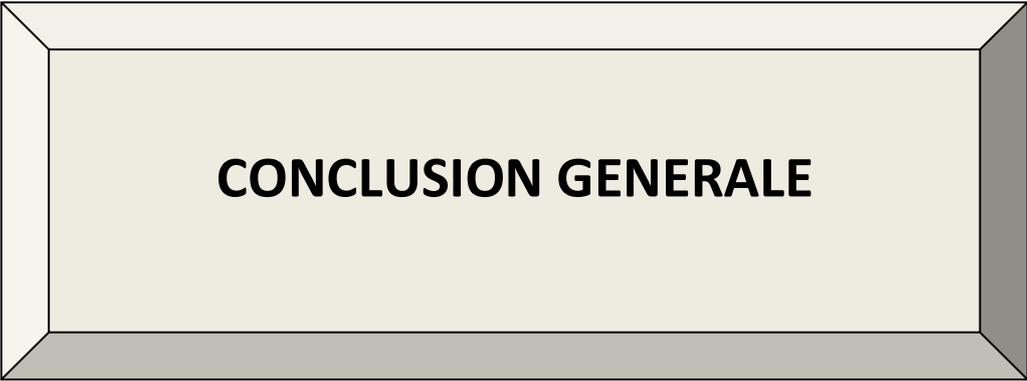
### 4.2.8.3 corrections des erreurs :

Si notre programme converti contient des erreurs ou des avertissements, ceux-ci sont énumérés après la vérification de cohérence ou la compilation dans une sous-fenêtre en dessous du fichier source. La cause de l'erreur est également précise. Lorsque nous sélectionnons un message d'erreur l'emplacement correspondant du fichier source s'affiche dans la fenêtre supérieure. Cela nous permet de remédier rapidement aux erreurs éventuelles

Nous pouvons procéder aux corrections et aux modifications en mode de substitution que nous activons à l'aide de la touche d'insertion

### 4.3 Conclusion

La nécessité de suivre l'évolution technologique nous oblige à envisager une modernisation pour garder la compétitivité. La migration vers la technologie SIMATIC S7 engendre des avantages décisifs. Durant ce chapitre nous avons démontré la démarche à suivre pour réaliser une migration réussite.



**CONCLUSION GENERALE**

## Conclusion générale

En guise de conclusion, nous souhaitant mettre le point sur les différentes étapes de réalisation de notre projet.

Notre mission consistait à étudier, en premier lieu, les automates S5-95U et S7-300 aussi que les logiciels STEP 5 et STEP 7, et de procéder par la suite à la migration de l'automate S5 vers S7 par le biais du logiciels STEP7, qui fait l'objet de notre étude.

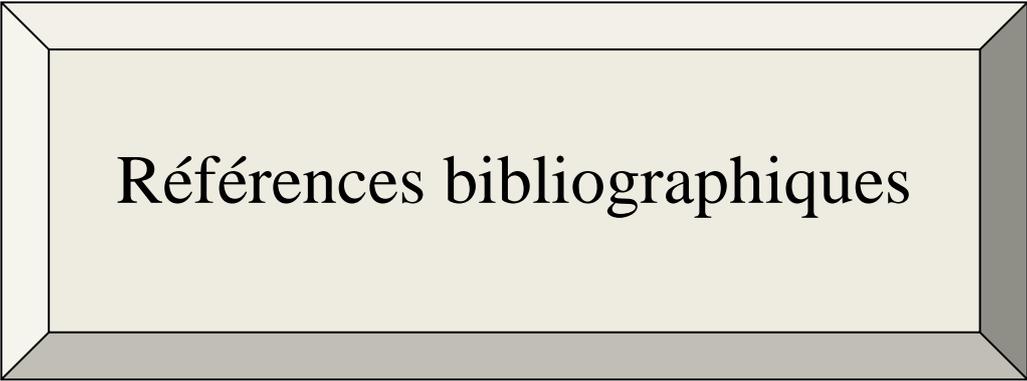
Nous avons débuté ce projet par la présentation de l'entreprise CEVITAL, son historique, ses activités et filiales et l'organisation du groupe.

En suite, nous avons abordé brièvement les dispositifs d'API, en étalant sur leurs aspects extérieurs, leurs structures internes, ainsi que les langages de programmation qui leur sied.

Par ailleurs, nous avons réalisé une étude comparative entre l'API S5 et S7, qui consiste à étudier les caractéristiques des automates S5-95U et S7-300, et les logiciels STEP 5 et STEP 7.

Et pour les besoin des industries de points de vue : fiabilité, performance, et disponibilité des pièces de rechange, il est nécessaire de changer l'automate S5-95U par un autre automate S7-300, plus performant. Nous avons opté pour l'approche, basée sur la conversion avec un convertisseur S5/S7.

Ce projet nous a permis de nous familiariser avec les automates S5 et S7 ainsi que leurs langages de programmation qui obéissent à des règles spécifique. En outre, il nous a aussi permis de consolider et d'approfondir nos connaissances dans ce domaine.



Références bibliographiques

## Références bibliographiques

- [1] selon le classement des vingt milliardaires africain pour l'année 2020, par LE MAGAZINE FORBES.
- [2] [www.Cevital.com](http://www.Cevital.com)
- [3] <https://www.Cevital.com/oxxo/>
- [4] <https://www.Cevital.com/mfg/>
- [5] <https://www.Cevital.com/baticompos.fr/>
- [6] <https://www.Cevital.com/numilog/>
- [7] H. Nussbaumer, «Informatique industrielle », presses polytechniques de Romandes, 1987.
- [8] W. Bolton, « les automates programmables industriels », Dunod, 2010
- [9] Y. Felfoul & N. Bekkouche, « Etude de la commande de la station de préparation de colle de la chaine de fabrication du panneau de particules à base d'automate SIEMENS », PFE, Université de Bejaia, 2006.
- [10] Alain Gonzaga, « les automates programmables industriels », cours.
- [11] L. Bergougoux, « automates programmables industriels »,Polytech Marseille, cours.
- [12] M. Bertrand, « automates programmables industriels, technique pour l'ingénieur »,2001
- [13] P. Jargot, « Langages de programmation pour API. Norme CEI 11313, technique de l'ingénieur », 1999.
- [14] Manuel système « SIMATIC S5, automates programmables S5-90U/S5-95U », référence 6ES5998-8MA32.
- [15] J. Hassaoui & K. khales, « migration d'un automate S5 vers S7, d'une remplisseuse de bouteilles d'eau au sien de l'entreprise IFRI », PFE, université Béjaia, 2017.
- [16] Manuel SIEMENS STEP 7 « pour une transition facile de S5 vers S7 », édition 05/2010
- [17] « SIEMENS AUTOMATION INNOVATION », 2009, site de siemens

## Références Bibliographiques

---

[18] Fiche technique, « Automate programmable S7-300 caractéristique électrique, technique des CPU, SIMATIC », 2001.

[19] Manuel SIEMENS SIMATIC S7-300, édition 02-2009

[20] Manuel « automate programmable S7-300. Installation et configuration, caractéristiques des CPU », édition 2003.

[21] Manuel « SIMATIC, programmer avec STEP 7 », référence 6ES7810-4CA10-8CW0, édition 05/2010.

## Résumé

Dans une politique visant à répondre à l'évolution des besoins des industries du point de vue : flexibilité, performance et disponibilité des pièces de rechange, les entreprises industrielles ont identifié un ensemble d'opportunités d'amélioration des performances de leurs processus de fabrication. Dans ce contexte, il nous a été utile, dans le cadre de notre travail, d'effectuer d'abord, une étude détaillée sur les automates S5-95U et S7-300 de la firme allemande SIEMENS, ainsi que les logiciels STEP 5 et STEP7, et de procéder par la suite à la migration de SIMATIC S5 vers SIMATIC S7, et pour cela nous avons opté pour l'approche basée sur la conversion de S5 vers S7 avec un convertisseur S5/S7, intégré dans le STEP 7.

Mots clés : S5-95U, S7-300, STEP 5, STEP 7, Convertisseur S5/S7

## Abstract

In a policy aimed at responding more to the changing needs of industries from the point of view of flexibility, performance and availability of spare parts, industrial companies have identified a set of opportunities for improving the performance of their manufacturing processes. . In this context, it was useful for us, in the context of our work, to first carry out a detailed study on the S5-95U and S7-300 PLCs from the German firm SIEMENS, as well as the STEP 5 software and STEP7, and then proceed with the migration from SIMATIC S5 to SIMATIC S7, and for this we have opted for the approach based on the conversion from S5 to S7 with an S5 / S7 converter, integrated in STEP 7.

Key words : S5-95U, STEP 5, STEP7, Converter S5/S7.S7-300.

## ملخص

في سياسة تهدف إلى تلبية الاحتياجات المتغيرة للصناعات من حيث المرونة والأداء وتوافر قطع الغيار ، حددت الشركات الصناعية مجموعة من الفرص لتحسين أداء عملياتها التصنيعية. في هذا السياق ، كان من المفيد لنا ، في سياق عملنا ، إجراء دراسة تفصيلية أولاً على S5-95U و S7-300 PLCs من شركة SIEMENS الألمانية ، بالإضافة إلى برنامج STEP 5 و STEP 7 ، ثم تابع الترحيل من SIMATIC S5 إلى SIMATIC S7 ، ولهذا اخترنا النهج القائم على التحويل من S5 إلى S7 باستخدام محول S5 / S7 ، مدمج في الخطوة 7.

الكلمات الرئيسية: S5 / S7 ، STEP 7 ، STEP 5 ، S7-300 ، S5-95U المحول