

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

Université Abderrahmane MIRA de BEJAIA  
Faculté de Technologie  
Département Automatique, Télécommunication, Electronique

# Mémoire de fin de cycle

En vue de l'obtention du Diplôme de Master en ELECTRONIQUE

Option : AUTOMATIQUE

*Thème*

*Conversion et combinaison de deux  
automates S5-95U et S7-200 vers un  
automate S7-300*

*Réalisé par :*

Mr : TIGHZER Farhat  
Mr : MESSAOUDI Slimane

*Encadrés par :*

Mr: MENDIL Boubekeur  
Mr: HAMMACHE Hakim

**Devant le jury composé de :**

Mr : HADJI  
Mme : IDJADARENE

Promotion : 2015/2016

## **Remerciements**

*Nous souhaitons tout d'abord remercier l'ensemble des membres du jury pour leur implication dans l'évaluation de notre travail de master.*

*Nous remercions notre promoteur Monsieur MENDIL Boubekour, pour le suivi continué tout le long de la réalisation de notre mémoire par ses conseils et ses remarques, et Monsieur HEMACHE Hakim, qui nous a encadrés au sein de l'entreprise CEVITAL et qui nous a aidés à trouver des solutions pour avancer. Enfin, nos remerciements vont à toutes les personnes qui nous ont soutenus de près ou de loin pendant la réalisation de notre travail, en particulier notre enseignant Monsieur HEDDAR Hocine et Monsieur TIGHZERT Lyes.*

# Sommaire

Introduction général.....	1
---------------------------	---

## Chapitre I : Présentation du processus étudié

### I.1. Présentation de l'entreprise CEVITAL

I.1.1.Introduction .....	2
I.1.2. Situation géographique du complexe.....	2
I.1.3. Bâtiments du complexe de Bejaia.....	2
I.1.4. Activités.....	3

### I.2. Présentation de l'unité de la margarinerie

I.2.1.Présentation de la margarinerie.....	5
I.2.2. Palettiseur.....	6
I.2.3. Navette.....	6
I.2.4. Banderoleuse à bras tournant.....	7

### I.3. Le système automatisé

I.3.1. introduction.....	8
I.3.2. Objectif d'un système automatisé.....	8
I.3.4. Structure d'un système automatisé.....	8
I.3.4.1. La partie opérative.....	8
I.3.4.2. La partie commande.....	9
I.3.4.3. Actionneurs.....	10
I.3.5 Les avantages et les inconvénients de l'automatisation.....	10

### I.4. L'automate Programmable Industriel

I.4.1. Introduction.....	11
I.4.2. Description des automates programmables industriels.....	11
I.4.3. Domaines d'emploi des automates.....	11
I.4.4. Principe de fonctionnement.....	11
I.4.5. Nature des informations traitées par l'automate.....	12

I.4.6. communication de l'API.....	12
I.4.7. Comportement des API.....	13
I.4.8. Critères de choix d'un automate .....	14
I.4.9. Avantages des automates programmables industriel.....	15
I.5.Conclusion.....	15

## **Chapitre II : Programmation des automates SIMATIC S5 et S7**

II.1. Introduction.....	16
-------------------------	----

### **II.2. Systèmes d'automatisations étudiées**

II.2.1. SIMATIC S5-95U.....	17
II.2.1.1. Périphérique intégré de S5-95U.....	17
II.2.1.2 Le cycle de vie de SIMATIC S5.....	17
II.2.2. SIMATIC S7-200.....	18
II.2.2.1. S7-200 automate de type compact.....	18
II.2.2.2. Utilisation de S7-200.....	18
II.2.3. SIMATIC S7-300 .....	18
II.2.3.1. S7300 automate de type modulaire.....	18
II.2.3.2. Les caractéristiques de SIMATIC S7-300.....	19
II.2.3.3. Application de S7-300.....	19

### **II.3. Programmation des automates siemens S5**

II.3.1. Structure des programmes et des données.....	20
II.3.2. La programmation structurée en S5.....	20

### **II.4. Principe de la programmation en S7**

II.4.1. Structure du programme .....	21
II.4.2. La programmation structurée en s7.....	22

### **II.5. Modules de S7 et leurs différences en S5**

II.5.1. Peu de changements par rapport à S5.....	23
--	----

II.5.2. Nouvelles performances.....	23
II.5.3. Coupleurs (IM).....	23
II.5.4. Modules de communication (CP).....	23

## **II.6. La programmation et l'organisation des blocs en S7**

II.6.1. Blocs.....	25
II.6.1.1. Fonctions et blocs fonctionnels.....	25
II.6.1.2. Blocs de données.....	25
II.6.1.3. Blocs système.....	26
II.6.1.4. Les blocs d'organisation.....	26
II.6.1.5. Classification des OB dans S5 et S7.....	27
II.7. Cycle de l' API.....	28
II.8. Transposition des blocs à la conversion.....	30
II.9. Conclusion.....	30

## **Chapitre III. Description de la navette et de la banderoleuse**

III.1. Introduction.....	31
III.2. Description de la navette.....	31
III.2.1. Principe de fonctionnement de la navette.....	33
III.2.2. Adressage de la navette.....	33
III.3. Description de la banderoleuse.....	34
III.3.1. Principe de fonctionnement.....	34
III.3.2. Parties opératives.....	35
III.3.2.1. Parties mécanique.....	35
III.3.2.2. Parties électrique.....	36
III.3.3. Principe de fonctionnement du circuit électrique.....	38
III.4. Conclusion.....	39

## **Chapitre IV. Conversion des deux programmes et leurs combinaisons dans S7-300**

IV.1.Introduction.....	40
<b>IV.2.Conversion du programme S5-95U de la navette vers S7-300</b>	
IV.2.1.Le convertisseur S5 /S7.....	40
IV.2.2.La procédure de conversion.....	41
IV.2.3.Exécution de la conversion.....	41
IV.2.4.Interprétation des messages.....	44
IV.2.5.Messages d’erreur.....	44
IV.2.6.Avertissements.....	47
IV.2.7.Retouche du programme converti.....	49
IV.2.8.Compilation.....	50
IV.2.9.Correction des erreurs.....	51
<b>IV.3. Conversion du programme S7-200 de la banderoleuse vers S7-300</b>	
IV.3.1. Introduction.....	52
IV.3.2. constitution du programme de S7-200.....	52
IV.3.3. Le choix du langage de conversion.....	52
IV.3.4. Modification des adresses.....	52
IV.3.5. Les mémentos spéciaux.....	52
<b>IV.4.Vérification de la configuration matérielle de l’API</b>	
IV.4.1.Modules pour la navette.....	55
IV.4.2.Réseau PROFIBUS pour la banderoleuse.....	57
IV.5.Conclusion.....	58
Conclusion général.....	59
Recherche bibliographique	

# Introduction général

# Introduction Générale

Le milieu industriel est caractérisé par le dynamisme des modèles et des conceptions, résultat d'une rénovation toujours en marche. Des changements majeurs ne cessent d'affecter les procédés et leurs implémentations. Dans les dernières années, de nouvelles générations d'automates ont vu jours. Ces dernières, sont beaucoup plus performantes et plus recommandées pour la réalisation des tâches dont la complexité reste croissante. Transférer ou convertir des programmes existants d'un automate à un autre s'impose alors. Cela permet aux entreprises une sorte de mise à niveau vis-à-vis du progrès et, en conséquence, devenir plus compétitives sur le marché. C'est dans ce contexte que s'inscrit notre travail. Subséquemment, l'objectif principal est d'étudier une migration de deux automates d'anciennes générations à un seul plus récent.

Notre projet de fin d'études, proposé et réalisé au sein de l'entreprise CEVITAL, traite la migration et la conversion de programmes concernant la gestion et la commande de deux machines (navette et banderoleuse) au niveau de l'unité de production de margarine. La banderoleuse est chargée de compléter la tâche réalisé par la navette. Mais il se trouve que chaque machine est gérée par son propre automate.

L'objectif principal de notre travail consiste à faire la conversion du programme de la banderoleuse, existant sur S7-200, et de faire une migration du deuxième programme de la navette, qui est programmé avec un automate ancien S5-95U, vers un seul et même automate S7-300 pour les deux machines. Cet automate a été choisi, d'une part, parce qu'il est doté d'un convertisseur intégré dont on peut se servir pour nous faciliter la migration du programme S5 et, de l'autre part, il présente de meilleurs avantages vu sa souplesse, sa fiabilité et sa capacité à répondre aux exigences.

Dans le chapitre 1, nous avons fait une présentation de l'usine CEVITAL où nous avons effectué notre stage, plus précisément l'unité de production de la margarine. Nous nous sommes focalisés sur les deux machines navette et Banderoleuse qui font objet de notre étude.

Le chapitre 2 est consacré à la présentation des automates programmables industriels en démontrant leurs avantages et leur contribution dans le développement de la production industrielle et aussi leur évolution dans le temps. La dernière partie expose les langages de programmation associés.

Le chapitre 3 présente en détails les deux machines en précisant leur rôle dans l'unité de margarinerie et leurs différentes parties : mécanique, électrique et automatique.

Le chapitre 4 décrit les démarches que nous avons suivies lors de la migration et de la conversion des programmes existants et leur combinaison dans un seul et, enfin, vérification de la configuration matérielle



# **Chapitre I**

## **Présentation du processus étudié**

## **I.1. Présentation de l'entreprise CEVITAL**

### **I.1.1. INTRODUCTION**

CEVITAL SPA a été créée avec des fonds privés en 1998. Elle est la première société privée dans l'industrie de raffinage d'huiles brutes sur le marché algérien. Elle s'est constituée autour de l'idée forte de bâtir un ensemble industriel intégré, concentré en première partie dans le secteur de l'agroalimentaire, dont le raffinage d'huile et de sucre, produits dérivés, négoce de céréales, distribution de produits destinés à l'alimentation humaine et animale. L'ensemble industriel a connu une croissance importante et a consolidé sa position de Leader dans le domaine agroalimentaire.

### **I.1.2. Situation géographique du complexe**

Le complexe CEVITAL est implanté au nouveau quai du port de Bejaia, à 3 km sud ouest de la ville, à proximité de la RN 26 ; cette situation géographique de l'entreprise lui profite bien étant donné qu'elle lui confère l'avantage de la proximité économique en effet elle se situe très proche du port de Bejaia [1].

### **I.1.1.3. Bâtiments du complexe de Bejaia**

Bâtiment Margarinerie

Bâtiment Raffinerie d'huile

Bâtiment Raffinerie de sucre

Bâtiment Soap Stock

Bâtiment Fabrication bouchons & poignées

Bâtiment de conditionnement d'huile

Bâtiment administratif et commercial

Bâtiment Raffinerie d'huile 800 T

Baraques de chantier

Hangar de stockage sucre roux

Hangar de stockage sucre blanc

Bâtiment de conditionnement de sucre

Silos de stockage de céréales



FIG.I.1. Situation géographique et les bâtiments du complexe CEVITAL

### I.1.4. Activités

Le complexe CEVITAL a commencé son activité par le conditionnement d'huile en décembre 1998. En février 1999, débutent les travaux de CEVITAL de génie civil sur la production de la raffinerie d'huile. Cette dernière est devenue fonctionnelle en Aout 1999. L'ensemble des activités de CEVITAL sont concentrées sur la production et la commercialisation des huiles végétales, margarine et sucre.

- **Raffinerie d'huile (1800t/j) :**

Elle a été mise à chantier en Mai 1998, l'adaptation d'une technologie en dernière génération lui a permis de rentrer en production en un temps record soit Aout 1999. Elle est considérée parmi les plus modernes au monde. Actuellement sa capacité de production est de 1800 tonnes par jour. Cette raffinerie est conçue pour traiter toutes les qualités d'huile comestible tel que : le colza, le tournesol, le soja, le palme...etc.

- **Production de margarine (600t /j) :**

Mise en chantier en Mars 2000, puis rentrée en production en Juillet 2001. Cette margarinerie construite par le groupe lui-même représente une offensive considérable sur le marché à grand publique. Sa capacité de production est de 100T/J pour chaque chaine de production qui sont au nombre de six.

- **Raffinerie de sucre :**

Elle est mise en chantier en octobre 2000, devenue fonctionnelle en octobre 2002.Elle est dotée d'un équipement industriel très modernisé qui répond aux besoins du marché, sa capacité de production actuelle est de 1600T/J dépassant ainsi les 500000T/an. Cette dernière couvrira les besoin nationaux en sucre blanc.

Une nouvelle raffinerie de suce d'une capacité de 3000T/J est actuellement mise en service en période de démarrage et d'essais.

- **Stockage céréales (120 000 tonnes) :**

Les silos de stockage sont opérationnels depuis Juin 2003. Ce sont de gigantesques récipients cylindriques construits en béton, destinés au stockage des céréales et des graines oléagineuses. Au nombre de 24, la capacité de stockage de chaque cellule est de 5000 tonnes. Ce qui offre une capacité de 120000 tonnes (la plus grande capacité de stockage en Afrique).

- **Fabrication d'emballage (PET) : (9600 unités par heures).**

- **Energie:**

L'unité d'énergie sert essentiellement à produire de l'électricité à l'aide d'un système d'installation qui se compose d'une chaudière dont le rôle est de produire de la vapeur sèche qui va se transformer en électricité en fonction d'autres éléments principalement l'alternateur [1].

## I.2. Présentation de l'unité de la margarinerie

### I.2.1. Présentation de la margarinerie

Cette unité est mise en production en novembre 2001, avec une Capacité de production : 600 tonnes/jour. Le bloc est composé de deux étages où se fait le parcours de la margarine à partir de la matière première aux produits finis. Il est de dimensions : 55 m de longueur, 35m de largeur et 15m de hauteur [1].

#### a) Rez-de-chaussée :

Dans ce niveau se déroule plusieurs étapes. On trouve les bacs d'émulsifiants qui contiennent la matière première (les ingrédients nécessaires à la fabrication de la margarine) et les étapes de palettisation et de conservation dans la chambre froide [2].

#### b) Le premier étage :

Dans cet étage, se fait le conditionnement de la margarine. On trouve six lignes de production :

Ligne N°1 : conditionnement de la margarine en pots de 500g.

Ligne N°2 : conditionnement de la margarine en plaquettes de 250g et 500g.

Ligne N°3 : conditionnement de la margarine de feuilletage.

Ligne N°4 : conditionnement de la margarine en plaquettes de 5Kg.

Ligne N°5 : production de graisse 100% végétale.

Ligne N°6 : production de SMEN.

Les produits qui sortent des lignes 1, 2, 3 et 6 sont déjà mis en cartons, car ces dernières contiennent des cartonneuses. Les produits qui sortent des lignes 4, 5 passent directement à l'étape suivante.

#### c) Le deuxième étage :

C'est à ce niveau que se trouve la salle de contrôle chargée de la préparation et des suivis des paramètres de fabrication de la margarine et le laboratoire pour assurer sa bonne qualité. Afin d'avoir une margarine bien conservée, des échantillons sont envoyés dans un laboratoire pour un contrôle de qualité [2].

### I.2.2. Palettiseur

L'unité de conditionnement de la margarine est constituée de trois palettiseurs qui servent à superposer, sur une palette, plusieurs couches de cartons venant de plusieurs tapis roulants. Chaque palettiseur assure deux ligne de production différentes ; donc aux totale six lignes.



**Figure I.2.** Palettiseur gérant deux lignes de production

### I.2.3. Navette

Elle se charge des trois palettiseurs. Elle les alimente en palettes vides à l'entrée et récupère les palettes pleines à la sortie de chaque palettiseur et les transporte vers la banderoleuse.



**Figure I.3.** Navette évacue une palette pleine

#### **I.2.4. Banderoleuse à bras tournant**

Elle a pour but de plastifier et stabiliser les charges palettisées avec un film extensible. Ensuite, elle fait sortir des palettes prêtes pour la manutention.



**Figure I.4.** Banderoleuse a bras tournant

### I.3. système automatisé

#### I.3.1. introduction

Les systèmes automatisés sont des mécanismes qui effectuent le travail de façon autonome. Capables de gérer des projets complexes, ils sont utilisés pour gérer les tâches répétitives. Leur utilisation permet une augmentation considérable de la production.

#### I.3.2. Objectif d'un système automatisé

L'automatisation est un moyen permettant d'accroître la compétitivité du produit élaboré. Cet accroissement de productivité exprime un gain de valeur ajoutée sous forme :

- d'une meilleure rentabilité,
- d'une meilleure compétitivité.
- Améliorer la flexibilité de la production.
- Perfectionner la qualité du produit.
- S'adapter à des contextes particuliers.
- Augmenter la sécurité. [3]

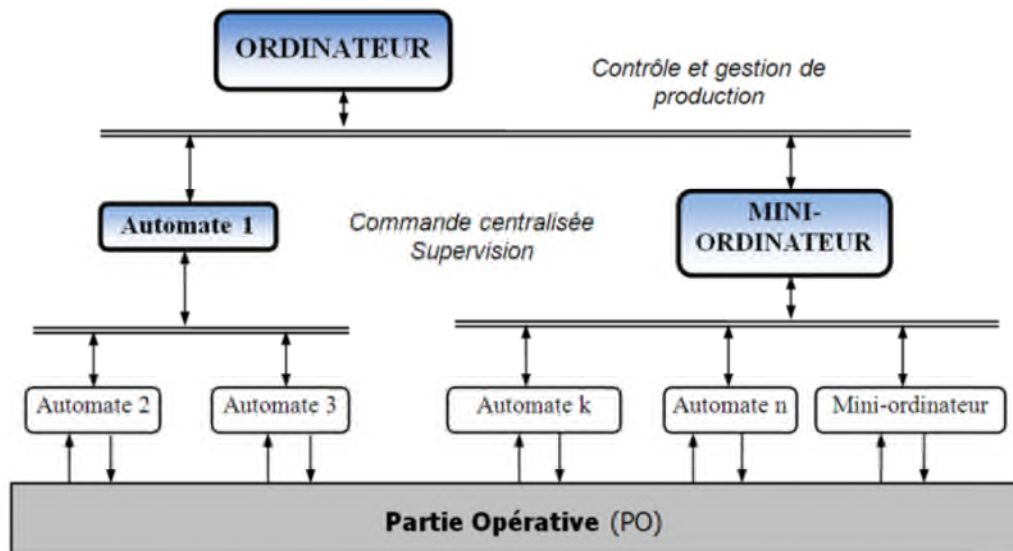
#### I.3.4. Structure d'un système automatisé

##### I.3.4.1. La partie opérative

La partie opérative est la partie visible du système. Elle comporte les éléments du procédé :

- ✓ des pré-actionneurs (distributeurs, contacteurs) qui reçoivent des ordres de la partie commande.
- ✓ des actionneurs (vérins, moteurs, vannes) qui ont pour rôle d'exécuter les ordres. Ils transforment l'énergie pneumatique (air comprimé), hydraulique (huile sous pression) ou électrique en énergie mécanique.
- ✓ des capteurs qui informent la partie commande de l'exécution du travail. Par exemple, on trouve des capteurs mécaniques, pneumatiques, électriques ou magnétiques montés sur les vérins. Le rôle des capteurs (ou détecteurs) est donc de contrôler, mesurer, surveiller et informer la partie commande sur l'évolution du système. [4]





**Figure I.5.** Schéma de la partie opérative.

### I.3.4.2. La partie commande

Elle est considérée comme le « cerveau » du système, Elle remplace l'opérateur humain. Ceci est garanti en traduisant le savoir-faire de l'opérateur sous la forme d'un programme informatique. Elle donne des ordres à la partie opérative en fonction :

- ✓ du programme qu'elle contient
- ✓ des informations reçues par les capteurs
- ✓ des consignes données par l'utilisateur

La partie commande est réalisée par :

- Une logique programmée dont les avantages sont :
  - ✓ facile à réaliser
  - ✓ multitâches
  - ✓ utilisable pour plusieurs applications (automate programmable, pc)
  - ✓ Flexibilité
- Une logique câblée : ses avantages sont le coût (un câblage électrique, un circuit électronique) et la rapidité temps réel.

Dans cette forme de système, le rôle de l'opérateur est limité à l'émission des consignes de démarrage ou de configuration, à un contrôle de sécurité et de bon fonctionnement du système. Le dialogue entre l'opérateur et le système est réalisé à travers un pupitre de commande.

Pupitre (interface de communication) : regroupe l'ensemble des éléments de communication entre le système et l'opérateur. Il peut être, constitué de boutons et de voyants mais aussi d'interface de communication plus complexe de type écran, clavier,...etc. [4]

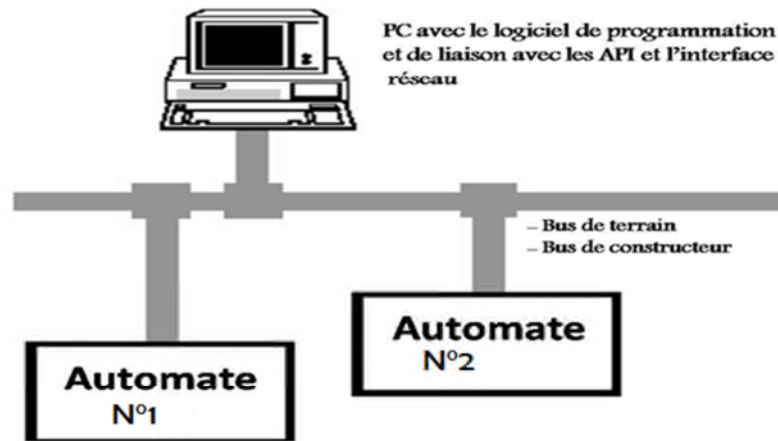


Figure I.6. La partie commande.

#### I.3.4.3. Actionneur

Est un élément de la partie opérative qui reçoit une énergie « transportable » pour la transformer en énergie « utilisable » par le système. Il exécute les ordres en reçus en agissant sur le système ou son environnement, Les actions sont les interventions physiques que le système de commande impose au processus industriel.

#### I.3.5. Les avantages et les inconvénients de l'automatisation

- **Les avantages**

- ✓ La capacité de production accélérée.
- ✓ Facilité d'intervention
- ✓ La souplesse d'utilisation.

- **Les inconvénients**

- ✓ Le coût élevé du matériel, principalement avec les systèmes hydrauliques.
- ✓ Nécessité d'une maîtrise totale du standard
- ✓ Spécialisation des programmeurs. [5]

## **I.4. Automates programmables industriels (API)**

### **I.4.1. Introduction**

Un API est un appareil électronique Programmable, adapté à l'environnement industriel, qui réalise des fonctions d'automatisme pour assurer la commande de pré-actionneurs et d'actionneurs à partir d'informations logiques, analogiques ou numériques. Ils sont apparus aux Etats-Unis vers 1969 où ils répondaient aux désirs des industries de l'automobile de développer des chaînes de fabrication automatisées qui pourraient suivre l'évolution des techniques et des modèles fabriqués, capable d'assurer, sans intervention humaine, un enchaînement d'opérations, correspondant à la réalisation d'une tâche donnée.

### **I.4.2. Description des Automates Programmables Industriels**

Un API reçoit des données. Celles-ci sont ensuite traitées par un programme. Les résultats obtenus forment des sorties. Un API est donc un instrument de calcul et de commande qui est relié physiquement, d'un côté, par une interface d'entrée à des capteurs et de l'autre côté, par une interface de sortie à des actionneurs. Il contient une unité centrale (CPU) qui gère l'ensemble du processus. Celle-ci contient le processeur, les mémoires vives et des mémoires mortes pour une taille débutant de 40K octets. Le programme contient des variables spéciales permettant de représenter les entrées et les sorties. Ces variables, comme nous le précisons dans la section suivante, sont mises à jour par l'API selon une procédure précise [4].

### **I.4.3. 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 ...). Il peut également assurer des fonctions de régulation de processus (métallurgie, chimie...). Il est de plus en plus utilisé dans le domaine du bâtiment pour le contrôle du chauffage, de l'éclairage, de la sécurité ou des alarmes [4].

### **I.4.4. Principe de fonctionnement**

Un API comporte essentiellement un microprocesseur et une mémoire programmable par un utilisateur à l'aide d'un langage adapté et des modules de communications (entrée/sortie). Matériellement, c'est un boîtier lié à l'extérieur par deux flux de signaux:

**-Les signaux d'entrée :** signaux en retour de la machine, signaux de commande issus du pupitre.

**-Les signaux de sortie :** signaux de commande émis vers la machine, signaux de visualisation vers le pupitre ... etc.

Lorsque l'automate est en service sur la machine, il émet à chaque instant les signaux de sortie nécessaires, en tenant compte de l'état des signaux d'entrée, et de l'avancement du cycle. Pour cela, il a comme référence le programme qui a été donné avant la mise en route de la machine. Chaque automate doit donc pouvoir stocker dans sa mémoire le volume d'informations nécessaire pour exécuter le programme.

#### **I.4.5. 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, etc.
- 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. [4]

#### **I.4.6. communication d'API**

L'automate doit pouvoir se connecter et dialoguer avec d'autres matériels et les agents d'exploitation. L'API ne se limite pas à communiquer avec le processus qu'il pilote via ses modules d'E/S. Parmi les autres types de relations susceptibles d'être assurées, on cite :

- ✓ La communication avec un opérateur par un pupitre ou un terminal industriel.
- ✓ L'affichage local de valeurs numériques ou de message.
- ✓ Les échanges d'informations avec d'autre API ou système de commande.
- ✓ Les échanges d'informations avec des capteurs et actionneur intelligents.
- ✓ Les échanges d'informations avec un superviseur.
- ✓ Les échanges d'informations avec un processeur maître, ou avec des esclaves, dans le cadre d'un réseau. [6]

### I.4.7. Comportement des API

Le moniteur d'exécution d'un API peut être composé de plusieurs sous-programmes, appelés tâches. Une tâche est un ensemble d'opérations programmées pour s'exécuter successivement, puis s'arrêter jusqu'au prochain lancement. Dans un API, une tâche peut être

- ✓ Cyclique : la tâche est immédiatement relancée après sa fin.
- ✓ Périodique : la tâche est relancée toutes les T unités de temps.
- ✓ Événementielle : la tâche est lancée à chaque fois qu'un événement prédéfini se produit.

L'exécution d'une tâche est un cycle composé de trois phases (figure I.3) :

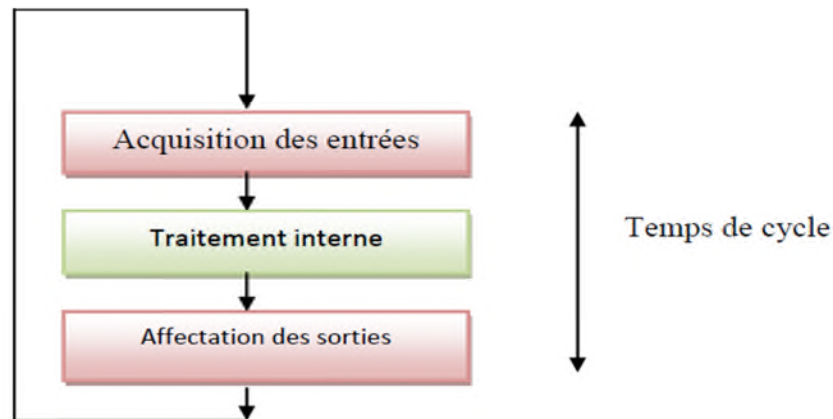
- l'acquisition des entrées : les variables d'entrées sont accessibles en lecture seule. Pendant cette première phase, leurs valeurs sont lues et ensuite stockées dans la mémoire de l'API.
- le traitement interne : c'est une phase d'exécution du programme et de calcul des valeurs de sorties à partir des valeurs stockées en mémoire dans la phase précédente. Les résultats de calculs sont ensuite à leur tour stockés en mémoire,
- l'affectation des sorties : les variables de sorties sont accessibles en écriture seule. Pendant cette phase, leurs valeurs sont mises à jour à partir des valeurs calculées dans la phase de traitement interne.

Les APIs peuvent être programmées selon deux modes différents :

- ✓ Mode mono-tâche : le moniteur d'exécution comporte une unique tâche cyclique, appelée tâche maître.
- ✓ Mode multitâches : le moniteur d'exécution comporte plusieurs tâches dont l'ordonnancement est réalisé en fonction de leurs priorités. A tout moment, une seule tâche est active et chaque tâche possède son propre cycle d'acquisition des entrées, traitement interne et affectation des sorties.

Les tâches possibles sont :

- La tâche maîtresse : elle est unique et cyclique.
- La tâche rapides: elle est optionnelle et périodique, mais peut lire un nombre limité d'entrées.
- La tâche événementielle : elle est optionnelle. Elle a accès à un nombre limité d'entrées, de variables internes et de sorties. [7]



**Figure I.7.** Un cycle de la tâche

#### **I.4.8. Critères de choix d'un automate**

Le choix d'un automate programmable est en premier lieu le choix d'une société ou d'un groupe et les contacts commerciaux et expériences vécues sont déjà un point de départ et de regarder sur le marché l'automate le mieux adapté aux besoins. Les grandes sociétés privilégieront deux fabricants pour faire jouer la concurrence et pouvoir "se retourner" en cas de "perte de vitesse" de l'une d'entre elles. Le personnel de maintenance doit toutefois être formé sur ces matériels et une trop grande diversité des matériels peut avoir de graves répercussions. Un automate utilisant des langages de programmation de type GRAFCET est également préférable pour assurer les mises au point et dépannages dans les meilleures conditions. La possession d'un logiciel de programmation est aussi source d'économies (achat du logiciel et formation du personnel). Des outils permettant une simulation des programmes sont également souhaitables.

Il faut ensuite quantifier les besoins :

- Nombre d'entrées / sorties : le nombre de cartes peut avoir une incidence sur le nombre de racks dès que le nombre d'entrées / sorties nécessaires devient élevé.
- Type de processeur : la taille mémoire, la vitesse de traitement et les fonctions spéciales offertes par le processeur permettront le choix dans la gamme souvent très étendue.
- Fonctions ou modules spéciaux : certaines cartes (commande d'axe, pesage ...) permettront de "soulager" le processeur et devront offrir les caractéristiques souhaitées (résolution, ...).

- Fonctions de communication : l'automate doit pouvoir communiquer avec les autres systèmes de commande (API, supervision ...) et offrir des possibilités de communication avec des standards normalisés (PROFIBUS ...). [4]

#### **I.4.9. Avantages de l'automate programmable industriel**

Les avantages de l'automate sont nombreux, pour l'utilisateur. Ils se situent à plusieurs niveaux :

- **Logiciel**

- ✓ Absence d'entretien.
- ✓ Robustesse.
- ✓ Economie.
- ✓ Sécurité.
- ✓ Gagne en temps.

- **Industriel**

- ✓ Cahier des charges accessible à tous (souplesse d'application).
- ✓ Sauvegarde de programmes.
- ✓ Simplicités des langages de programmation.
- ✓ Possibilité de dialogue avec son entourage.

- **Exploitation**

- ✓ Souplesse d'emploi.
- ✓ Connexions aux organes de gestion.
- ✓ Intervention sur processus en cours.
- ✓ Modification instantanée (par console) du programme.
- ✓ Suivi temps réel l'évolution de l'automatisme.
- ✓ Exploitation (fonctionnement) en conditions sévères [5].

### **I.5. Conclusion**

Dans ce chapitre, nous avons fait une présentation générale de CEVITAL, en particulier l'unité de production de la margarine. Nous nous sommes focalisés sur la banderoleuse et la navette qui sont les deux machines étudiées et notamment l'automatisation gérant leur fonctionnement.

Le prochain chapitre sera consacré à la présentation des automates et des langages de programmation faisant l'objet de notre étude de migration et de conversion.

**Chapitre II**  
**Programmation des automates**  
**SIMATIC S5 et S7**



## II.1. Introduction

Le nom de SIMATIC était hier encore associé aux automates et, plus précisément, aux automates S5. Aujourd'hui, SIMATIC est devenu synonyme de l'intégration totale.

L'intégration totale est un concept révolutionnaire visant à réunir l'univers de la fabrication manufacturière et l'univers des procédés. Toutes les briques matérielles et logicielles nécessaires à la réalisation d'un projet portent désormais un seul nom : SIMATIC.

L'intégration totale est rendue possible par l'homogénéité parfaite des données, tant

- au niveau de la base de données, Les données ne sont plus saisies qu'une seule fois mais sont disponibles dans toute l'usine. Les erreurs dues à la transposition des données et les incohérences appartiennent désormais au passé.
- au niveau de la conception et de la programmation, Toutes les briques servant à la réalisation du projet sont conçues, configurées, programmées, mises en service, testées et surveillées sous une seule interface utilisateur avec l'outil qui leur est dédié.
- qu'au niveau de la communication.

## II.2. Systèmes d'automatisations étudiées

Les trois gammes des anciens SIMATIC S5 et celle des nouveaux S7 comprend les systèmes d'automatisation évolutifs avec une classification de la gamme d'entrée, de la gamme milieu, et le haut de gamme. la figure II.1 illustre bien cette répartition [8].

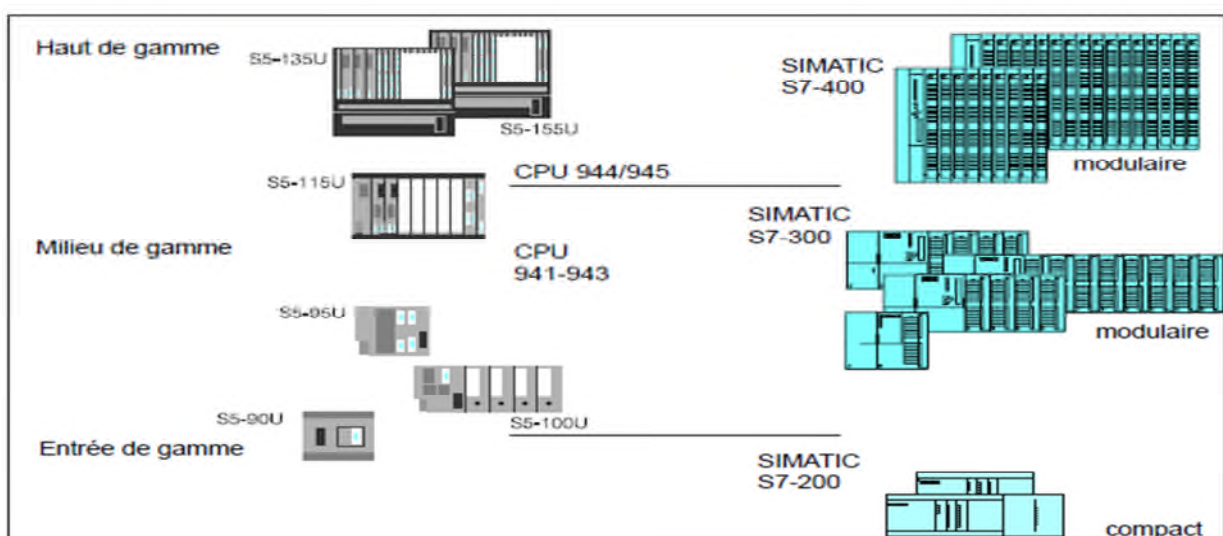


Figure II.1. Systèmes d'automatisation SIMATIC.

### II.2.1. SIMATIC S5-95U

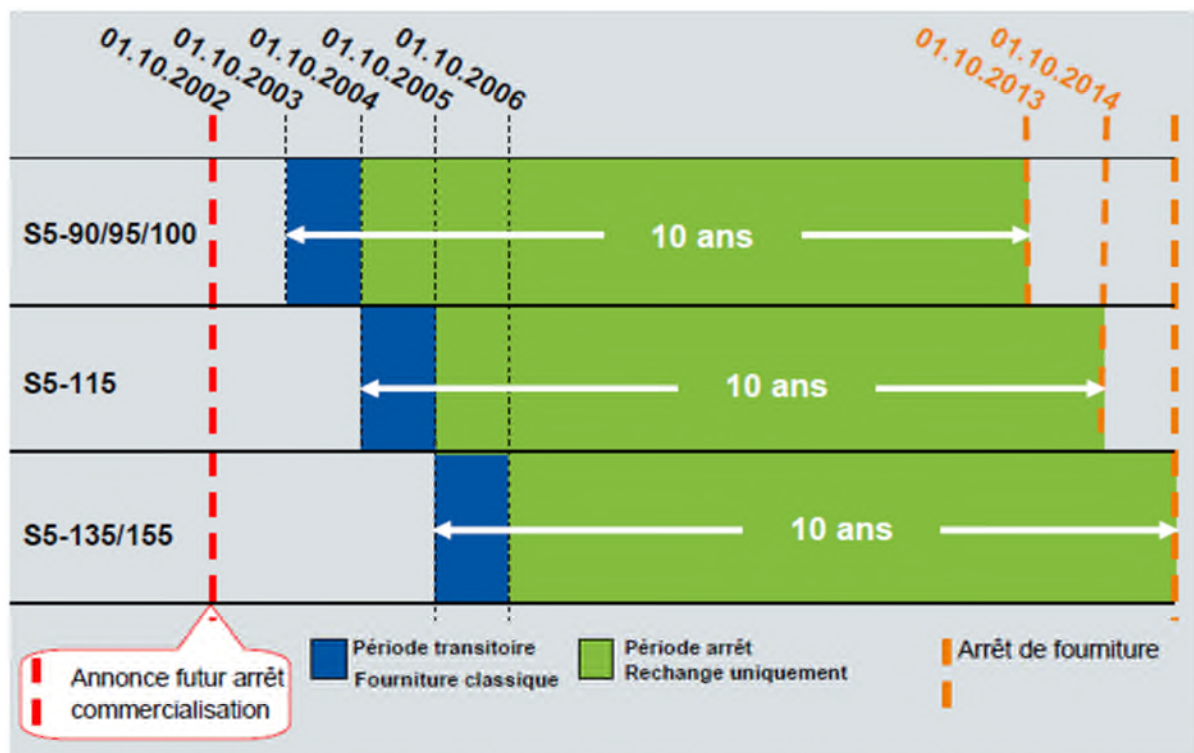
L'automate S5-95U est un appareil compact permet d'apporter des solutions à des problèmes d'automatisme simples. Il se compose d'un boîtier plastique abritant le processeur, l'alimentation interne, les entrées et les sorties. [9]

#### II.2.1.1. Périphériques intégrés de S5-95U

On peut citer les éléments suivants : 16 entrées TOR, 16 sorties TOR, 4 entrées d'alarme, 8 entrées analogique, 1 sorties analogique et 2 entrées de comptage.

#### II.2.1.2 Le cycle de vie de SIMATIC S5

En 2015, le S5 a eu 36 ans. Le cycle de vie que nous montre figure II.2 et la stratégie de siemens de stopper définitivement le SIMATIC S5 au 01/10/2002, c'est l'annonce du futur arrêt de commercialisation de S5. L'arrêt de fourniture de S5-90U/S5-95U/S5-100U sera le 01/10/2013 et 01/10/2014 pour le S5-115U, le 01/10/2015 pour S5-135U/S5-155U. [10]



Subject to change without notice / © Siemens SAS 2009 – Industry Automation and Drive Technologies

Figure II.2. Cycle de vie de S5

### **II.2.2. SIMATIC S7-200**

Le SIMATIC S7-200 est un micro-automate compact d'entrée de gamme. Il possède un progiciel qui lui est propre. Les propriétés du système S7-200 ne supportent pas une conversion logicielle des programmes S5 en programmes S7.

#### **II.2.2.1. S7-200 automate de type compact**

Le micro-automate S7-200 est de type compact. Il intègre le processeur, l'alimentation, les entrées et les sorties. Selon les modèles et les fabricants, il pourra réaliser certaines fonctions supplémentaires (comptage rapide, E/S analogiques ...) et recevoir des extensions en nombre limité.

#### **II.2.2.2. Utilisation de S7-200**

Ces automates, de fonctionnement simple, sont généralement destinés à la commande de petits automatismes [4].

### **II.2.3. SIMATIC S7-300**

Le mini-automate modulaire SIMATIC S7-300 fait encore partie de l'entrée de gamme.

#### **II.2.3.1. S7-300 automate de type modulaire**

L'automate S7-300 est de type modulaire, c'est à dire que l'on peut ajouter au CPU plusieurs autres modules qui vont gérer la communication, le comptage rapide, le pesage, etc. Le processeur, l'alimentation et les interfaces d'entrées / sorties résident dans des unités séparées (**modules**), grâce à ces modules intelligents que l'on dispose sur un ou plusieurs racks ont l'avantage de ne pas surcharger le travail de la CPU car ils disposent bien souvent de leur propre processeur [4].



**Figure II.3.** SIMATIC S7-300 de Type modulaire.

### II.2.3.2. Les caractéristiques de SIMATIC S7-300

- ✓ Il permet de réaliser la majorité des applications d'automatisme complexes où puissantes.
- ✓ Souplesse d'utilisation grâce à une configuration simple et une capacité de traitement rapide.
- ✓ Conception modulaire, nous pouvons le composer en fonction de nos besoins à partir d'un vaste éventail de modules qui comprend :
  - des CPU de différents niveaux de performances.
  - des modules de signaux pour entrées et sorties TOR et des modules analogiques.
  - des coupleurs pour configurer un automate sur plusieurs profilés-support.
  - procurer le degré de protection IP20 [4].

### II.2.3.3. Application de S7-300

- ✓ Fabrication manufacturière.
- ✓ Industrie automobile.
- ✓ Construction mécanique générale.
- ✓ Construction de machines spéciales, OEM.
- ✓ Plasturgie.
- ✓ Industrie de l'emballage.
- ✓ Industrie agro-alimentaire.
- ✓ Génie procédés [4].

## II.3. Programmation des automates Siemens S5

### II.3.1. Structure des programmes et des données

La structuration des programmes et des données est assez semblable à celle existant sur les matériels de la génération S7, mais des différences significatives (et parfois déroutantes) sont à observer. La structure du programme est schématisée par figure II.4. La programmation est organisée sous forme de *blocs*. On distingue plusieurs types de blocs. On y distingue:

#### •Blocs d'organisation OB

Le bloc d'organisation OB1 permet de définir l'ordre chronologique d'exécution des blocs de programme PB

*Remarques:* si un programme comporte des blocs autres que le bloc de programme PB1, la présence du bloc d'organisation OB1 est obligatoire. Si on veut un programme linéaire, on peut seulement utiliser le PB1. La programmation de l'OB1 est alors inutile.

#### •Blocs de programme PB

Ces blocs sont des blocs fonctionnels (nécessairement sans paramètres). Ils contiennent le programme ou des parties du programme ; c'est-à-dire que l'on va y mettre les instructions à exécuter. La numérotation est libre (de PB0 à 255 sur le 135U et PB0 à PB53 sur le 95U).

#### •Blocs de fonction FB

Il s'agit également de blocs de programme comme les PB, mais ceux-ci disposent d'un jeu d'opérations élargi par rapport à celui des blocs de programme: ils acceptent des paramètres.

#### •Blocs séquentiels SB

Les blocs séquentiels (SB) sont spécialement utilisés pour effectuer des séquences selon GRAFCET. Les paramètres d'entrées y seront les conditions d'avancement d'un pas de séquence et les paramètres de sorties, les ordres à exécuter lorsque ces conditions seront vérifiées. [9]

### II.3.2. La programmation structurée en S5

Pour mettre plus de clarté dans un programme, on le découpe en plusieurs sections affectées chacune à une fonction technologique. On est donc amené à programmer différents blocs (PB ou FB). L'ordre chronologique d'appel et de traitement des différents blocs est défini dans le bloc d'organisation (OB 1). Chaque bloc ainsi appelé peut lui-même contenir une instruction de saut vers un autre bloc. A la fin de l'exécution du bloc ainsi appelé, le traitement se poursuit automatiquement au lieu de départ du saut [9].

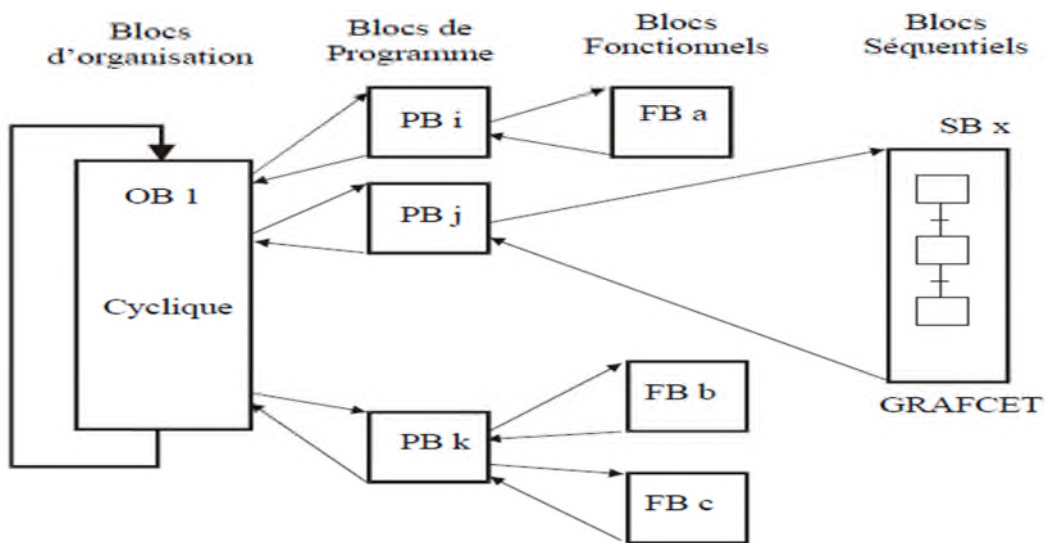


Figure II.4. Structure des programmes en STEP 5

## II.4. Principe de la programmation en S7

### II.4.1. Structure du programme

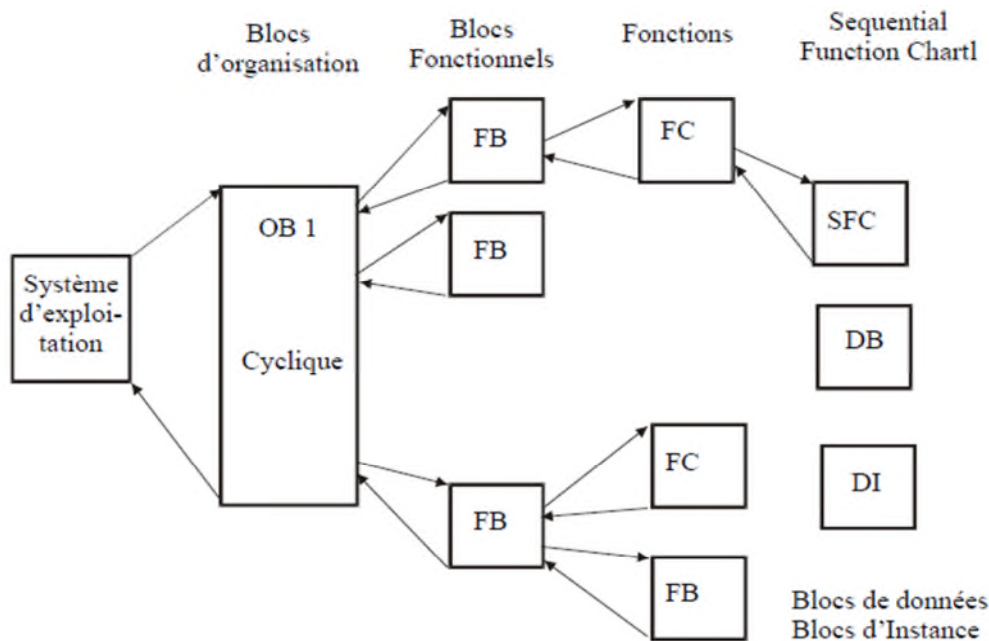
La programmation structurée permet la rédaction claire et transparente de programmes. Elle permet la construction d'un programme complet à l'aide de modules qui peuvent être échangés et/ou modifiés à volonté. Pour permettre une programmation structurée confortable, il faut prévoir plusieurs types de modules :

- Les modules de programmes (FC) servent à subdiviser le programme en parties fonctionnelles et/ou orientées vers le "processus".
- Les modules de données (DB) contiennent des données variables, textes, valeurs de temporisations ou de comptage, résultats de calculs, etc. et sont accessibles et actualisables à tout moment.

- Les modules séquentiels (SB) sont spécialement utilisés pour effectuer des séquences selon GRAFCET. Les paramètres d'entrées y seront les conditions d'avancement d'un pas de séquence et les paramètres de sorties, les ordres à exécuter lorsque ces conditions seront vérifiées [11].

#### II.4.2. La programmation structurée en S7

Pour mettre plus de clarté dans un programme, on le découpe en plusieurs sections affectées, chacune à une fonction technologique. On est donc amené à programmer différents blocs (FB ou FC). L'ordre chronologique d'appel et de traitement des différents blocs est défini dans le bloc d'organisation (OB 1). Chaque bloc ainsi appelé peut lui-même contenir une instruction de saut vers un autre bloc. A la fin de l'exécution du bloc ainsi appelé, le traitement se poursuit automatiquement au lieu de départ du saut [9].



**Figure II.5.** Architecture des programmes en S7

## II.5. Modules de S7 et leurs différences avec S5

### II.5.1. Peu de changements par rapport à S5

Tout comme la gamme des SIMATIC S5 qu'elle vient compléter, la gamme SIMATIC S7 mise sur la modularité bien éprouvée :

- Unités centrales (CPU),
- Modules d'alimentation (PS),
- Coupleurs (IM),
- Processeurs de communication (CP) ; (par exemple pour le raccordement à PROFIBUS),
- Modules de fonction FM ; (par exemple de comptage, régulation, positionnement etc.),
- Les modules d'entrées/sorties TOR ou analogiques sont maintenant appelés des modules de signaux (SM) [8].

### II.5.2. Nouvelles performances

Les modules de STEP 7 se signalent par de nouvelles performances techniques.

- Il n'y a plus de cavaliers ni de commutateurs sur les modules.
- Les modules peuvent fonctionner sans ventilateur. Ils disposent comme les S5 du degré de protection IP 20.
- Vous avez un choix de modules paramétrables et dotés de fonctions de diagnostic !
- Les appareils d'extension et la périphérie décentralisée ET 200 peuvent déclencher des alarmes. [8]

### II.5.3. Coupleurs (IM)

Certains coupleurs disponibles dans S5 ont leurs équivalents dans S7. Ils peuvent être utilisés pour un couplage sur de courtes distances. Pour un couplage sur de longues distances, il est recommandé d'émettre les signaux via le bus PROFIBUS.

### II.5.4. Modules de communication (CP)

Les modules de communication S5 et S7 pouvant être employés dans les différents réseaux sont énumérés ci-dessous avec leurs fonctions :



**✓ Interface AS-i**

L'interface AS-i (*Actuator Sensor Interface*) est un système de connexion employé pour le premier niveau du processus dans les installations d'automatisation. Il permet notamment de relier des capteurs et actionneurs échangeant des données binaires. La quantité des données ne doit pas dépasser 4 bits par esclave.

**✓ MPI**

Le sous-réseau MPI est adapté au niveau du terrain et de la cellule lorsqu'ils restent de dimensions modestes. L'interface MPI est une interface multipoint pour SIMATIC S7/M7 et C7. Elle est également conçue comme interface PG et peut servir à la mise en réseau de plusieurs CPU pour l'échange de petites quantités de données (jusqu'à 70 octets).

**✓ PROFIBUS**

PROFIBUS est un système de communication ouvert acceptant en plus des SIMATIC les appareils d'autres constructeurs. C'est le réseau idéal à l'échelle de la cellule et du terrain et permet la transmission rapide de moyennes quantités de données (environ 200 octets).

**✓ Ethernet**

Ethernet est un système de communication ouvert acceptant en plus des SIMATIC les appareils d'autres constructeurs. C'est le réseau idéal pour le niveau de la conduite et de la cellule et permet la transmission rapide de grandes quantités de données.

**✓ Couplage point-à-point**

Un couplage point-à-point n'est pas un sous-réseau à proprement parler. Il est réalisé dans SIMATIC par des processeurs de communication (CP) point-à-point et relie deux partenaires de communication (automates programmables, scanner, PC, etc.). [8]

## II.6. La programmation et l'organisation des blocs en S7

La programmation avec Step7 s'articule autour de différents blocs, programmables dans 3 langages

- ✓ Le List : C'est le langage de base de step7. Il est basé sur une centaine d'instructions.
- ✓ Le Contact : C'est une représentation du List, qui s'apparente à de la logique câblée.
- ✓ Le Logigramme : C'est une autre représentation du List sous forme de bloc. C'est le langage le plus utilisé ici puisqu'il peut toujours être converti en Contact ou en List.

### II.6.1. Blocs

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

#### II.6.1.1. Fonctions et blocs fonctionnels

**Les FC :** Une fonction (FC) est un bloc de code "sans mémoire" dont les paramètres de sortie affichent en fin d'exécution les valeurs qu'elle a calculées. Le traitement ultérieur et la sauvegarde de ces résultats doivent donc être considérés par l'utilisateur lorsqu'il programme l'appel de la fonction.

- **Les FB :** Ce sont des blocs fonctionnels que l'on doit appeler avec un DB d'instance.

#### II.6.1.2. Blocs de données :

Les blocs de données servent à stocker les données du programme utilisateur. On fait la distinction entre les blocs de données globaux et les blocs de données d'instance :

- Les blocs de données globaux contiennent des informations auxquelles on peut accéder à partir de tous les blocs logiques du programme utilisateur (FB, FC, OB).ils ne sont pas affectés à un bloc précis (comme dans STEP 5).
- Les blocs de données d'instance ils sont toujours associés à un FB. Les données de ce DB ne devraient être traitées que par le FB correspondant. [8]

### II.6.1.3. Blocs système :

#### ▪ Fonctions et blocs fonctionnels système (SFC et SFB) :

Il n'est pas nécessaire de programmer chaque fonction. Vous pouvez recourir à des blocs préprogrammés intégrés au système d'exploitation des unités centrales, par exemple pour programmer les fonctions de communication. Il s'agit des blocs suivants:

- **Fonctions système (SFC)** possédant les mêmes propriétés que les fonctions (FC).

- **Blocs fonctionnels système (SFB)** possédant les mêmes propriétés que les blocs fonctionnels (FB)

#### ▪ Blocs de données système (SDB)

Nous avons jusqu'ici parlé de blocs renfermant le code ou les données du programme utilisateur. Il existe, à côté de ces blocs, des blocs qui servent à stocker les adresses ou les paramètres des modules [12].

### II.6.1.4. Les blocs d'organisation

Les blocs d'organisation (OB) servent d'interface entre le système d'exploitation et le programme utilisateur. Ces blocs d'organisation remplissent des tâches différentes bien précises.

- **Programme de démarrage OB 100** : La CPU effectue une mise en route après la mise sous tension

- **Programme cyclique OB 1**

Lors d'une exécution normale de programme, les traitements se font de façon cyclique. L'exécution du programme contenu dans l'OB 1 est démarrée une fois par cycle (quand il est fini, il recommence). On peut se servir de l'OB 1 pour appeler des blocs fonctionnels (FB, SFB) ou des fonctions (FC, SFC) [12].

### II.6.1.5. Classification des OB dans S5 et S7

**Tableau II.1.** Blocs d'organisation dans S5 et dans S7

Fonction		S5	S7
Programme principal	Cycle libre	OB1	OB1
Alarmes	Alarme temporisée (alarme de retardement)	OB6	OB20 à OB23
	Alarme horaire (alarme d'horloge à heure fixe)	OB9	OB10 à OB17
	Interruptions	OB2 à OB5	OB40 à OB47
	Alarmes de processus	OB2 à OB9 (EB 0)	remplacées par des interruptions
	Alarmes cycliques (alarmes d'horloge)	OB10 à OB18	OB30 à OB38
	Alarme multiprocesseur	–	OB60
	Démarrage manuel	OB21 (S5-115U) OB20 (à partir de S5-135U)	OB100
Mise en route	Redémarrage manuel	OB21 (à partir de S5-135U)	OB101
Fonction		S5	S7
Mise en route	Redémarrage manuel	OB21 (S5-115U) OB20 (à partir de S5-135U)	OB100
	Redémarrage automatique	OB21 (à partir de S5-135U)	OB101
Erreurs	Erreurs	OB19 à OB35	OB121, OB122, OB80 à OB87
Autres	Traitement à l'état de STOP	OB39	n'est plus possible
	Traitement en arrière plan	–	OB90

### Exemple d'alarmes sur OB

Erreur d'alimentation OB 81

Erreur de temps OB 80 : Dépassement du temps de cycle

Erreur matérielle sur CPU OB 84

Erreur d'exécution de programme OB 85 :

Erreur d'accès à un bloc

Erreur d'accès aux mémoires images E/S

Erreur de communication OB 87 : Erreur de communication par données globales

Erreur de programmation OB 121

Erreur de conversion

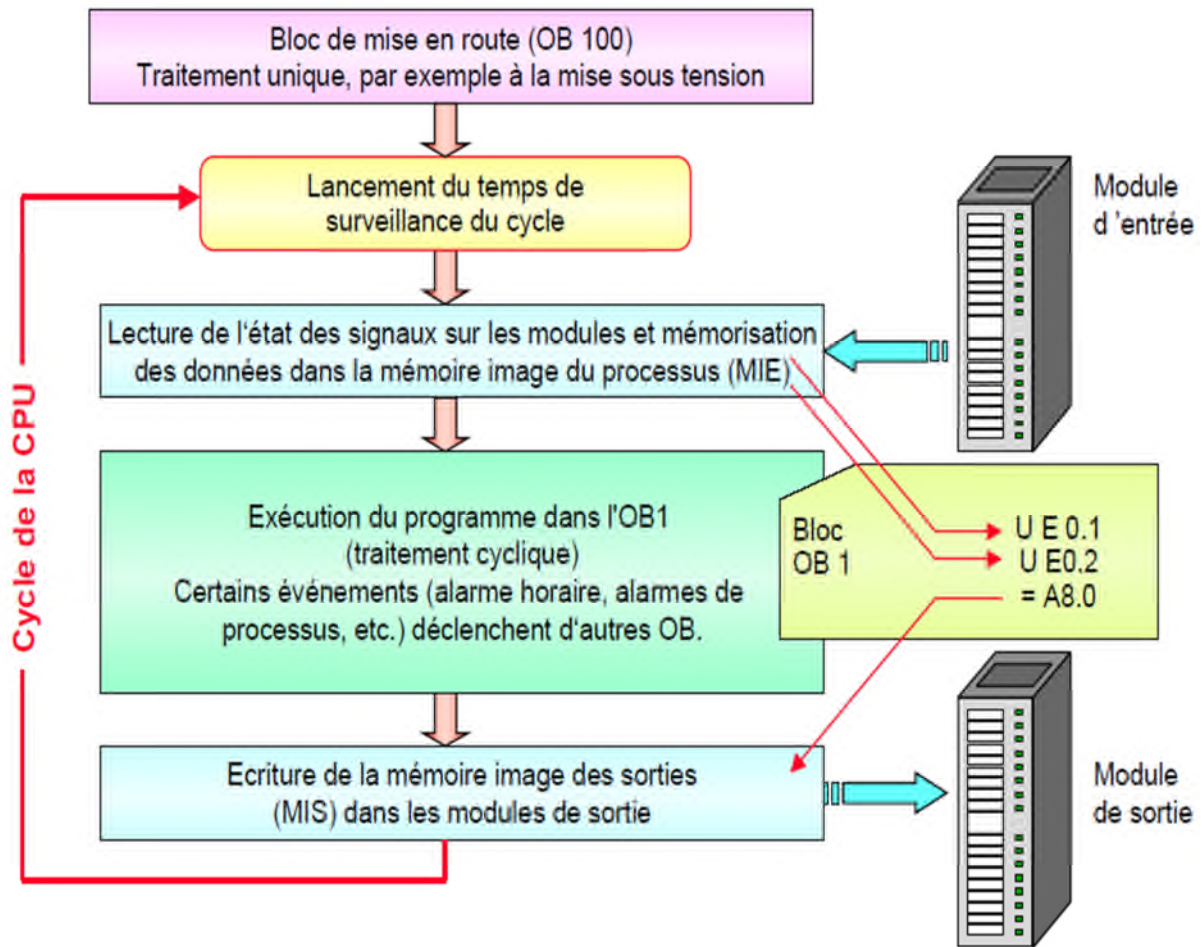
Bloc chargé

### II.7. Cycle de l'API

- ✓ STEP 7 offre une surveillance du temps de cycle maximal, ce qui garantit le temps de réaction maximal.

Par défaut, le temps de cycle maximal est de 150 ms ; vous pouvez modifier cette valeur par Paramétrage ou démarrer la surveillance de temps à n'importe quelle position de votre programme avec la fonction système SFC43 "RE\_TRIGR". Si votre programme dépasse le temps de cycle maximal défini pour l'OB1, le système d'exploitation appelle l'OB80 (erreur de temps). Si l'OB80 n'a pas été programmé, la CPU passe à l'état d'arrêt.

- Outre la surveillance du temps de cycle maximal, l'observation d'un temps de cycle minimal est garantie. Le système d'exploitation diffère le début d'un nouveau cycle (écriture de la mémoire image des sorties dans les modules de sorties) jusqu'à ce que le temps de cycle minimal soit écoulé [13].



**Figure II.6.** Traitement cyclique du programme, tiré de [14].

### Remarques

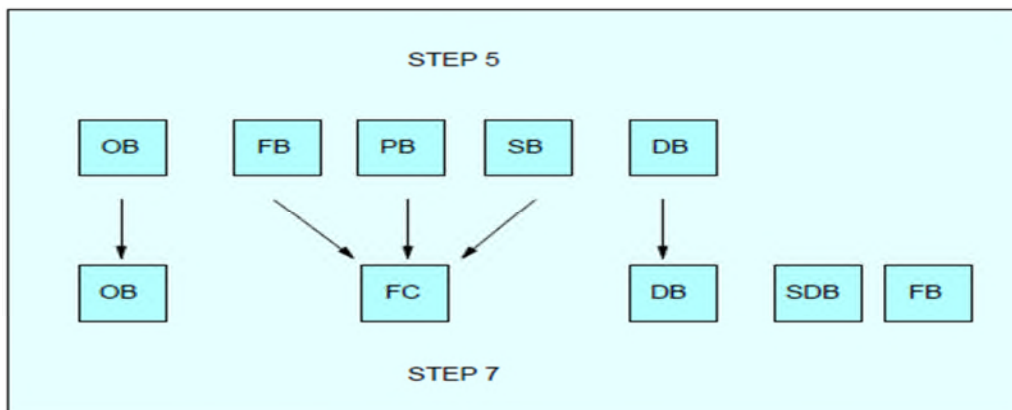
✓ Le temps effectué par l'UC pour exécuter un programme de 1K mots (210 = 1024 mots), est appelé période d'un automate programmable. Ce temps varie d'un modèle à un autre et ne se différencie qu'à la milliseconde.

✓ A la notion du temps de cycle est liée la notion du temps de réponse c'est-à-dire. l'intervalle de temps séparant la variation d'une entrée et la variation de la sortie conséquente. La pire situation se rencontre dans le cycle synchrone vis à vis des entrées et des sorties :  $TR < 2.TC$ , (TC : temps de cycle, TR : temps de réponse) [13]

## II.8. Transposition des blocs à la conversion

### Correspondance des blocs

La structure des blocs a été modifiée dans S7. Vous voyez sur figure II.7, la correspondance des blocs de STEP 5 et STEP 7. A la conversion, les blocs de STEP 5 seront remplacés de façon analogue par les blocs de STEP 7. [8]



**Figure II.7.** Blocs S5 et S7 remplissant des fonctions similaires

## II.9. Conclusion

La programmation est une tâche primordiale dans l'automatisation. Dans ce chapitre, nous avons décrit les différents automates et leurs langages de programmations qui font l'objet de notre étude. Ce qui permet une lecture et une compréhension simples des programmes existants et ainsi de procéder aux tâches de conversion et migration. Le prochain chapitre sera consacré à la présentation des deux machines étudiées (banderoleuse et navette) et les démarches que nous avons suivies pour élaborer le programme de leur commande en même temps avec un seul automate S7-300.

## **Chapitre III.**

# **Description des la navette et de la banderoleuse**



### III.1.Introduction

La banderoleuse et la navette jouent un rôle très important dans l'unité de production de la margarine, elles effectuent les taches finales sur les palettes pleines pour enfin les stocker dans la chambre froide. Dans ce chapitre nous allons décrire en détail les deux machines (banderoleuse et navette).

### III.2.Description de la navette

Navette : machine réalisée pour transporter des palettes pleines du palettiseur à l'enrouleur.

Palette : plate-forme ou support de bois ou plastique de :

- Longueur variable (environ 1200mm)
- Largeur variable (environ 1000mm)
- Hauteur variable (environ 150mm)

Ces supports sont aptes à soutenir plusieurs couches de produits.

#### **La structure portante de la navette :**

Est formée en profilés métalliques, plis pressés de tôle à haute résistance, façonnés en forme de C et branchés par des montants transversaux.

#### **Le convoyeur à double rouleaux :**

Il est motorisé pour le transport des palettes pleines et vides, il composée d'un châssis en profilés métalliques électro-soudés, dans lesquels est montée une série de rouleaux motorisés.

#### **Groupe roues :**

Formé de 4 roues en aciers et branchés parmi eux par 2 arbres l'un desquels motorisé.

#### **Rails de glissement :**

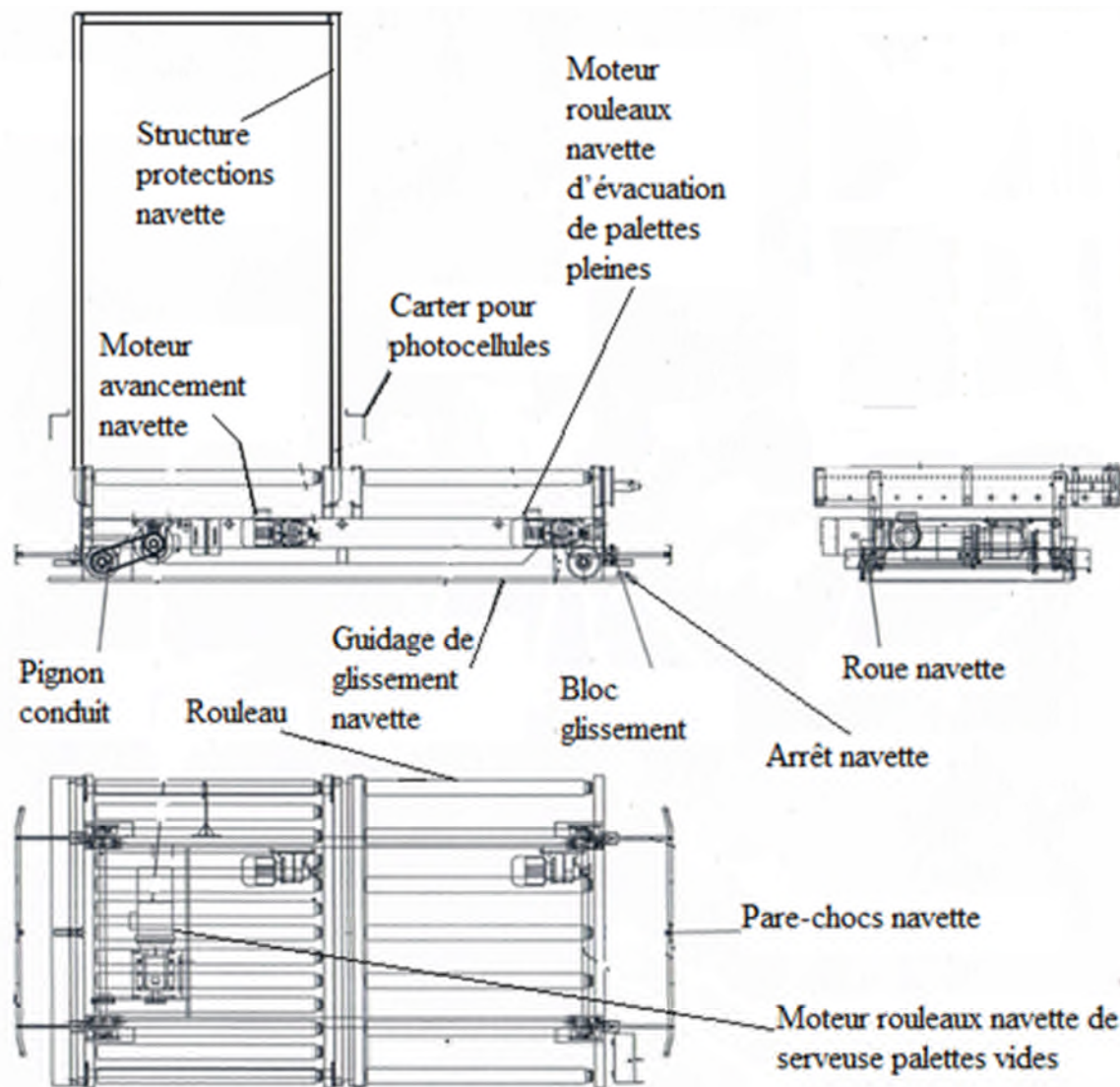
Composé de 2 tiges de fer montées avec des pièces de 2 mètres environ et branchées entre eux par des tiges transversaux.

#### **Protection contre les accidents :**

Sont composées de grillage avec châssis construit de boudins en fer. La hauteur est de 2200mm environ et elle couvre tout le périmètre de la navette, à exception des points où il y a le passage d'une palette, dans ce cas la protection est constituée d'une barrière à photocellules [15].

➤ **Magasin palettes vides pour 3 palettiseurs :**

La structure portante est exécutée en profilés métalliques branchées et soudées. Il y a 2 guidages d'invitation pour l'insertion des palettes dans le magasin réalisé en tôle de grosse épaisseur. Le mouvement du groupe positionnement des palettes est obtenu à travers le motoréducteur à bains d'huile. Le groupe de support palettes est fixé aux guidages d'invitation roulants sur un tourillon avec un mouvement qui passe par des vérins pneumatiques [15].



**Figure.III.1.** Description des pièces de la machine

### III.2.1.Principe de fonctionnement de la navette

La navette mémorise les appels des entrées et sorties des palettiseurs. Ensuite, elle fait l'ordre de priorité ;

Etat initial : magasin de la palette vide. Chaque palettiseur a deux appels :

-Appel de la demande d'une palette vide et l'appel de l'évacuation de la palette pleine sauf pour le troisième palettiseur qui constitue de deux rouleaux de sortie, il a besoin de deux appels d'évacuation de la palette pleine.



**Figure.III.2.** Palettisation de deux lignes de sortie assurées par un bras de robot

### III.2.2.Adressage de la navette

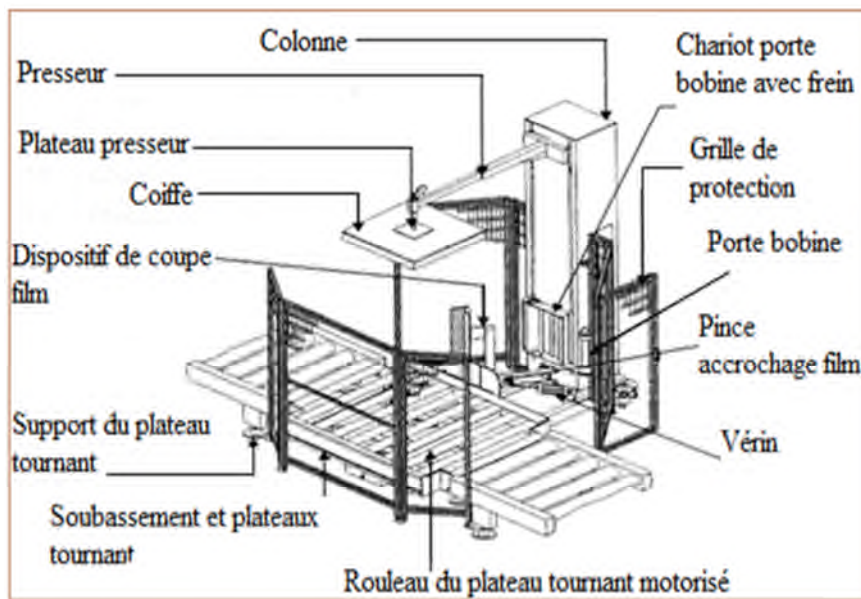
Chaque position, que ce soit une entrée ou une sortie, est constituée de quatre fins de course qui forme l'adressage de la position de la navette. Dans notre processus, en a besoin de neuf (9) adresses. Chaque adresse est formée de quatre bits (0 et 1).



**Figure.III.3.** barres de fin de course d'adressage d'une entrée et une sortie d'une palette par la navette

### III.3.Description de la banderoleuse

Il s'agit de la banderoleuse automatique avec plateau tournant MOD.095 du constructeur TMG IMPRIANTI, installé à COSSANO BELBO (CN) Italie. Elle est utilisée pour la plastification des palettes contenant plusieurs couches de cartons [2].



**Figure.III.4.**Nomenclature de la banderoleuse automatique

#### III.3.1.Principe de fonctionnement

La palette se déplace sur le convoyeur d'alimentation et s'arrête à la position de plastification au moyen d'une cellule photoélectrique qui démarre le cycle de plastification selon la séquence suivante :

- ✓ Le plateau presseur descend et s'arrête au contact de la palette.
- ✓ Le plateau tournant commence la rotation entraînant l'ensemble de plastification, avec la queue du film retenue dans la pince d'accrochage du film.
- ✓ Le chariot spirale monte et descend en entraînant le chariot porte bobine qui permet de serrer la palette et les couches de cartons de bas en haut avec du film plastique.
- ✓ Et enfin couper et souder le film plastique à l'aide d'un dispositif entraîné par des vérins [2].

### III.3.2.Parties opératives

#### III.3.2.1.Partie Pneumatique

Elle est constituée d'un vérin pneumatique qui transforme une énergie pneumatique en énergie mécanique [2].



**Figure.III.5.** Vérin pneumatique

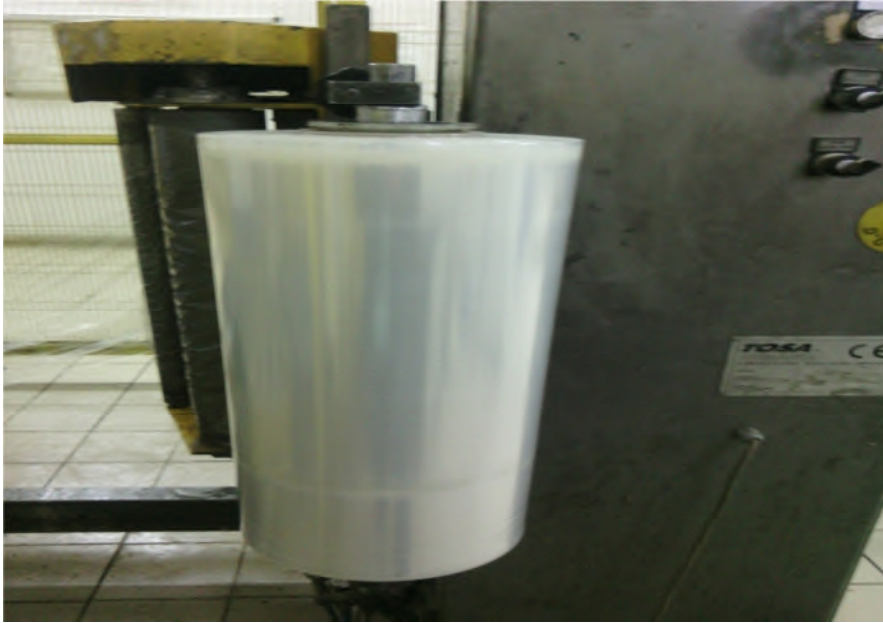
#### III.3.2.2.Parties mécanique

- **La fin de course**

Ces détecteur sont actionnés par contacte mécanique directe lorsque la tête du capteur entre en contacte avec la pièce à détecter. La tête du capteur se déplace actionnant ainsi le contacte électrique du détecteur [2].

- **Chariot porte bobine**

C'est un dispositif de déroulement et de pré-étirage de film.



**Figure.III.6.** Chariot porte bobine

### III.3.2.3.Parties électrique

La partie électrique est constituée de

- **Moteur asynchrone triphasé a cage**

Il est utilisé pour l'entraînement des machines de la navette et la banderoleuse.

- **Les détecteurs photoélectriques**

Ils permettent de détecter un objet sans contacte physique qui se situe jusqu'à plusieurs dizaines de mètres. Il est constitué d'un émetteur et récepteur de lumière.

- **Coupe film**

Il est doté de coupe à chaud avec une résistance, installé sur un bras a actionnement pneumatique.

- **Soudeuse film**

C'est un bras actionné par un cylindre pneumatique. À son extrémité se trouve un bloc de soudage à chaleur émise par un réchauffeur à cartouche.



**FIG.III.7.** Coupe film à chaud et soudeuse film.

- **Transformateurs**

Ils servent à transformer le courant. L'unité d'intervention dispose de deux transformateurs abaisseurs, le premier pour une tension de 230V et le deuxième pour une tension de 24V.

- **Contacteur**

C'est un appareil mécanique de connexion capable d'établir, de supporter et d'interrompre des courants dans des conditions normales et aussi en cas de surcharges.

- **Relais de protection**

Ce sont des appareils qui agissent sur les origines d'établissement et d'interruption des circuits par l'intermédiaire de contact auxiliaire.

- **Relais magnétothermique**

C'est un relais à maximum de courant. Il assure la protection contre les surintensités brutales sous l'effet thermique du courant, assurant ainsi la protection contre les surcharges lentes et prolongées.

- **Disjoncteur magnétothermique :** Il est utilisé pour la protection contre les surcharges du courant et les courts circuits des appareils [2].

### III.3.2.4.Principe de fonctionnement du circuit électrique

#### Exemple sur la soudeuse de film et le coup film avec :

##### **Soudeuse film :**

Fermeture manuelle du disjoncteur magnétothermique Q1350

Excitation de la bobine K342

Fermeture du contacteur K342

Alimentation du transformateur T135

Fermeture manuelle du disjoncteur magnétothermique Q1351

Alimentation des résistances R1350 et R1351

##### **Coupe film :**

Fermeture manuelle du disjoncteur magnétothermique Q1350

Excitation de la bobine K469 par la sortie A2.7

Fermeture du contacteur K469

Alimentation du transformateur T137

Fermeture manuelle du disjoncteur magnétothermique Q1371

Alimentation de la résistance R137 [2].



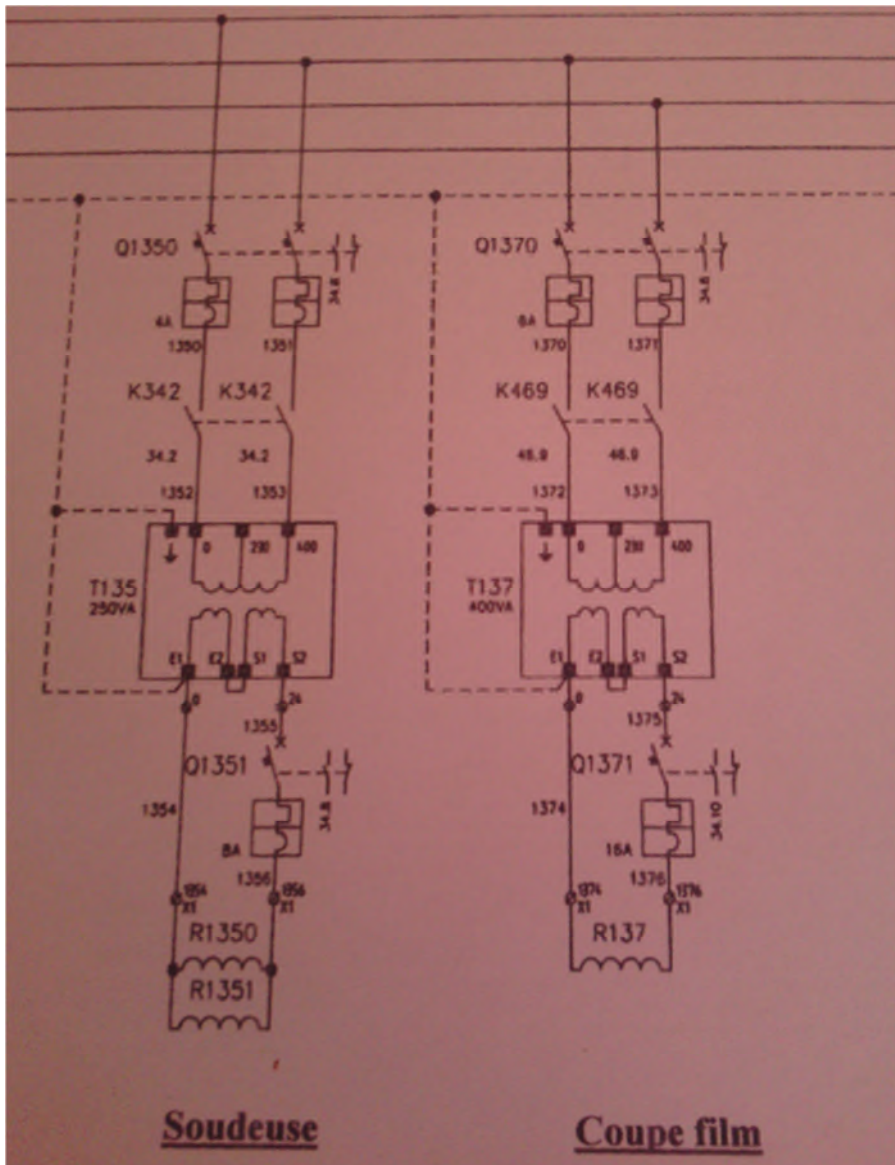


Figure.III.8. Schémas électrique de soudeuse et coupe de film

### III.7.Conclusion

Ce chapitre nous a permis de bien étudier les deux machines en précisant leur rôle dans l'unité de margarinerie et leurs différentes parties ( mécanique, électrique et automatique).Le prochain chapitre est consacré au fond du travail qui concerne la migration et la conversion de leurs programmes en utilisant un seul automate S7-300.

## **Chapitre IV.**

# **Conversion des deux programmes et leurs combinaisons dans S7-300**

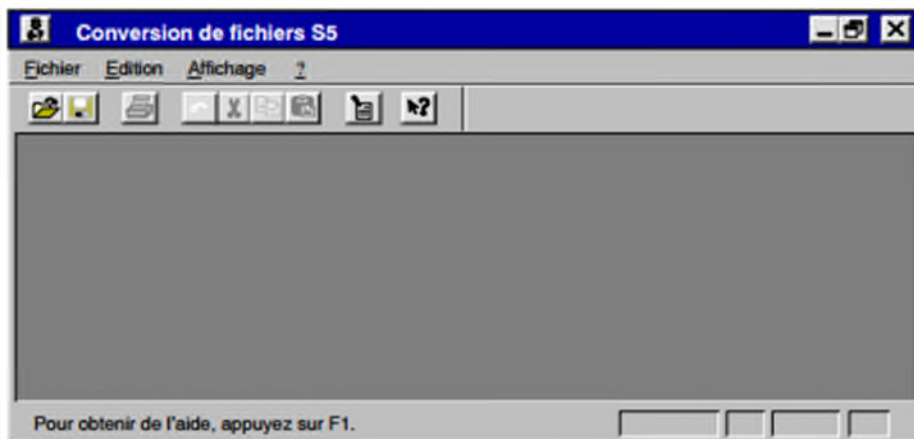
## IV.1. Introduction

Dans ce chapitre, nous allons entamer la migration et la conversion vers le S7-300 des programmes existants sur le S5-95U et le S7-200 qui font fonctionner respectivement les deux machines, navette et banderoleuse, afin d'avoir un seul automate qui commande les deux à la fois.

## IV.2. Conversion du programme S5-95U de la navette vers S7-300

### IV.2.1. Le convertisseur S5 /S7

Le convertisseur S5 /S7 est une application intégrée dans le step7. Elle nous permet, si possible, de convertir des programmes S5 existants en S7 de façon complète. Ils seront convertis entièrement. Il sera nécessaire de corriger les erreurs générées par cette conversion. Le convertisseur S5/S7 se présente avec l'image-écran de Figure IV-1. [8]



**Figure IV.1.** Convertisseur S5/S7

Les fichiers suivants sont nécessaires à la conversion de Notre programme S5 :

- Fichier programme <nom > ST.S5D
- La liste des références croisées <nom> XR.INI. Elle est nécessaire afin de conserver la structure et la hiérarchie d'appel du programme S5.

Si on désire employer des mnémoniques (noms symboliques) plutôt que des adresses absolues dans notre programme, on aura besoin de :

- La liste d'assignation S5 <nom>Z0.SEQ pour créer la liste d'assignation convertie.

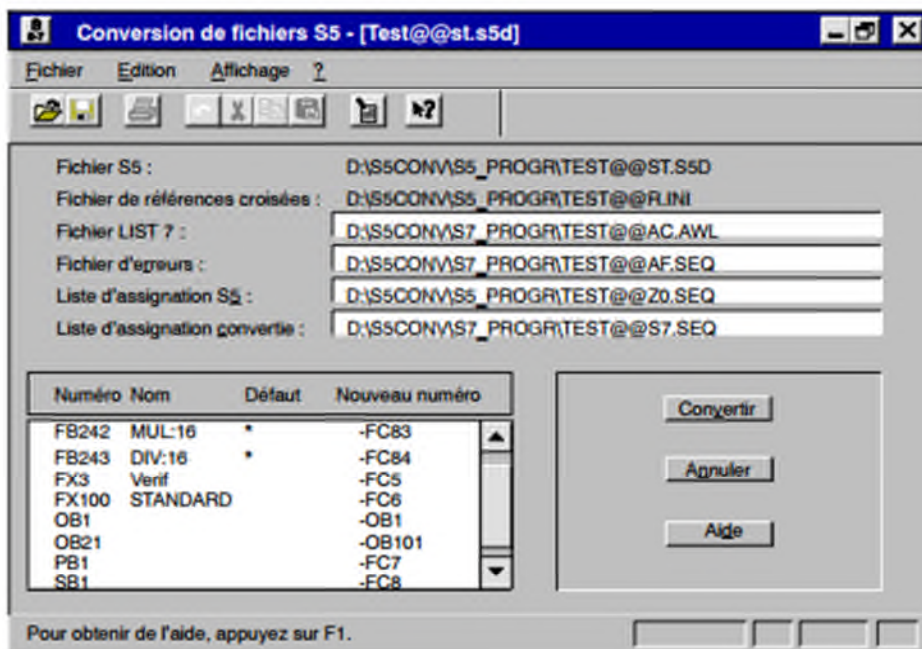
**IV.2.2.La procédure de conversion**

Sélection d'un fichier programme :

Pour sélectionner un fichier programme, il faut suivre les étapes suivantes:

1. Exécuter la commande Fichier > Ouvrir.
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 pour confirmer votre choix.

Le convertisseur S5/S7 affiche les fichiers source et cible ainsi qu'une table de correspondance des anciens et nouveaux numéros de blocs. La figure VI-2 présente la boîte de dialogue "Conversion de fichiers S5 -- [ST.S5D]". [8]



**Figure IV.2** conversion du fichier S5

**IV.2.3.Exécution de la conversion**

En cliquant sur le bouton "Convertir", nous mettons en route la procédure de conversion. Elle se compose de deux phases. La conversion et la transposition de la liste d'assignation. Au cours de la première phase de conversion, le programme S5 est converti en un fichier source S5 avec tous les blocs et tous les commentaires. Au cours de la deuxième phase de conversion, le fichier source S5 est converti en un fichier source LIST avec les nouveaux numéros de bloc et la syntaxe de S7.

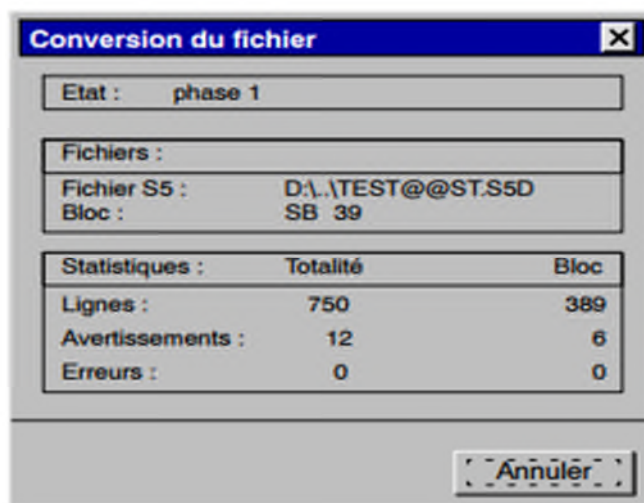


Figure IV.3. Phase 1 de la conversion.

#### ❖ Conversion de la liste d'assignation :

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

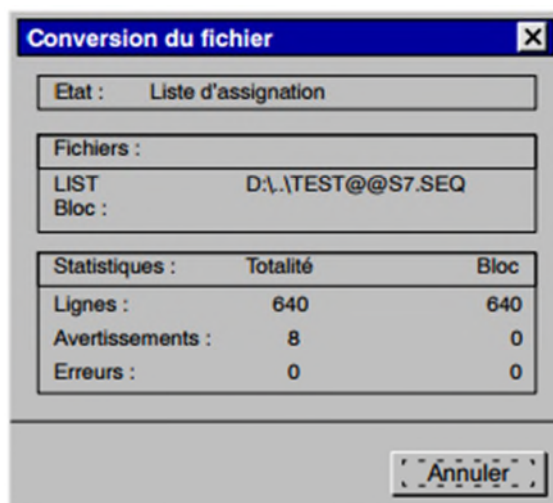


Figure IV.4 liste d'assignation de la conversion.

#### ❖ Fichiers générés

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

- Fichier <nom>Z0.SEQ : Ce fichier est créé pendant la première phase de 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.

- 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 les première et seconde phases de conversion et pendant la transposition de la liste d'assignation.

La conversion une fois terminée, une boîte de dialogue indiquant le nombre d'erreurs et d'avertissements s'affiche donc, dans notre programme de la navette en a 21 avertissements et une erreur (le programme dans L'OB doit être reformulé). [8]

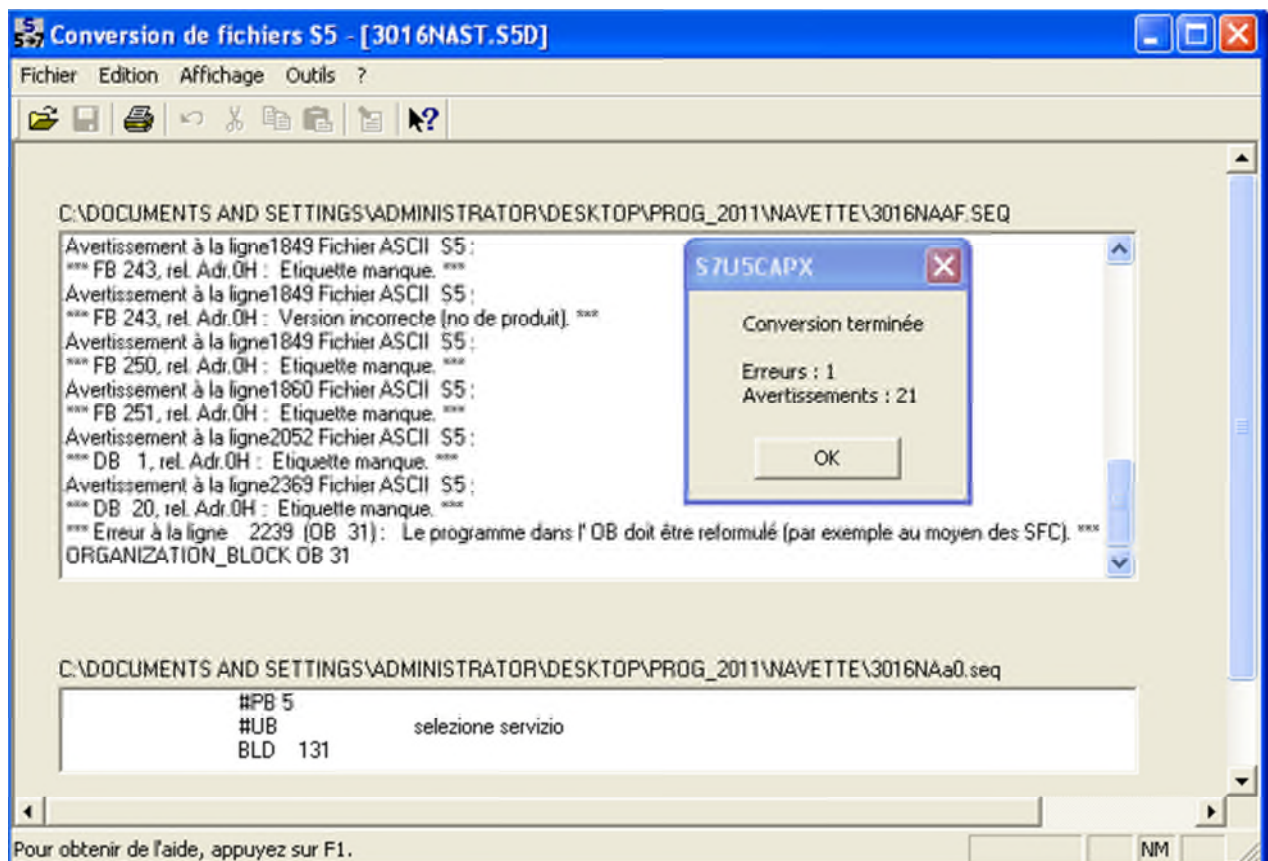


Figure IV.5. Erreurs et avertissement de la conversion.

Dans la zone inférieure de la fenêtre “Conversion de fichiers S5“, nous pouvons visualiser dans le fichier concerné la position à laquelle 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.

#### IV.2.4. Interprétation des messages

Parmi les messages du convertisseur, on distingue les messages d’erreur et les avertissements.

Pour pouvoir analyser ces messages nous devons procéder comme suite :

- Visualisez dans la zone inférieure de la fenêtre “Messages“ le fichier dans lequel l’erreur s’est produite.
- Consultez l’aide en ligne pour comprendre la signification du message.
- Corrigez l’erreur comme il est proposé.

#### IV.2.5. Messages 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 IV-1 dresse la liste de tous les messages. [8]

**Tableau .IV.1.** Messages d’erreur avec leurs significations et remède.

Message d’erreur	Origine	Signification	Remède
Paramètre absolu diverge de l’identificateur d’opérande	phase 1	L’identificateur d’opérande n’est pas correct.	Vérifiez 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.	Vérifiez la structure du programme.

	phase 2	Un bloc est appelé qui n'existe pas dans le fichier de programme.	Vérifiez que la liste de références croisées a bien été indiquée lors de la conversion ou contrôlez 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	Vérifiez l'instruction.
Commande non définie	phase 1	L'instruction MC5/LIST n'est pas valable	Corrigez le fichier programme S5.
	phase 2	L'instruction n'existe pas dans S7.	Editez 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 bit à des temporisations et à des compteurs.	Vérifiez 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, utilisez l'instruction CALL SFC.
Profondeur d'imbrication incorrecte	phase 1	Toutes les parenthèses ne sont pas correctement fermées	Respectez les niveaux de parenthèse, éliminez l'erreur de programmation.
Opérande incorrect	phase 1	L'opérande ne convient pas à l'opération.	Vérifiez la source S5.
	phase 2	L'opérande ne convient pas à l'opération	Modifiez le fichier LIST.
Erreur de conversion	phase 2	Opération BI sans constante	Complétez l'opération de chargement par une constante.
Erreur dans le fichier macro, xy	phase 2	Erreur de macro-instruction	Vérifiez la macro-instruction



non pris en compte			
Paramètre formel non défini	phase 1	Il y a plus de paramètres que dans le bloc appelant.	Vérifiez le fichier programme S5
Fichier ou répertoire introuvable	phase 1	Le fichier programme ne contient aucun bloc.	Vérifiez le fichier programme.
Longueur de commentaire incorrecte	phase 1	Erreur dans le fichier S5	Vérifiez le fichier programme.
Commentaire trop long	phase 1	Erreur dans le fichier S5	Vérifiez le fichier programme
Aucun nom de bloc spécifié	phase 1	Le nom de bloc ne comporte que des espaces.	Entrez un nom de bloc.
Droits d'accès manquants	globale	Le fichier est protégé en écriture.	Supprimez la protection en écriture.
Marque non définie	phase 1	Le repère de saut n'est pas défini dans l'étiquette.	Vérifiez le fichier S5
Marque incorrecte	phase 1	Le repère de saut contient des caractères non valables	Vérifiez le fichier S5
Opérateur incorrect	phase 1	L'opérateur dans le fichier S5 est inconnu ou impossible à convertir	Remplacez 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.	Utilisez éventuellement l'instruction indiquée.
Nombre de paramètres incorrect	phase 1	Erreur dans le programme S5	Vérifiez le fichier programme.

Paramètre erroné	phase 1	Erreur dans le programme S5	Vérifiez le fichier programme.
Type de paramètre incorrect.	phase 1	Erreur dans le programme S5	Vérifiez 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.	Supprimez 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.	Effacez de 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.	Eliminez l'erreur dans le programme S5.
Un code MC5 incorrect a été converti.	phase 1	Conversion d'une ancienne opération de S5.	Aucun

#### IV.2.6. Avertissements

Un avertissement est émis quand une partie du programme S5 est certes convertie, mais qu'il faudrait en vérifier la validité. Le tableau IV-2 présente les avertissements qui peuvent y avoir, leur signification et la réaction conseillée. [8]

**Tableau .IV.2.** Avertissements et leurs significations.

Version incorrecte (no de produit)	phase 1	Un bloc fonctionnel standard de S5 doit être remplacé par une FC de S7.	Aucune
Version incorrecte (bloc GRAPH 5)	phase 1	Les blocs GRAPH 5 ne sont pas convertibles.	Utilisez éventuellement un bloc créé avec GRAPH pour S7.

Vérifiez la base de temps choisie.	phase 2	Dans S7, la base de temps peut être plus serrée que dans S5.	Paramétrez la base de temps à l'aide de l'application « Configuration matérielle ».
I/D n'influence que l'accum 1L, qui est maintenant l'accum 1.	phase 2	Les accumulateurs de S7 sont étendus à 32 bits.	Examinez les conséquences d'une opération indirecte d'incrément ou de décrémentation dans le programme LIST.
Tenez 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 est prélevé dans le mot de memento ou de données approprié).	Modifiez la logique dans S5 ou utilisez des appels de bloc fixes.
OB23 et OB24 sont convertis en OB 122.	phase 2	Les OB23 et OB24 sont remplacés tous deux par l'OB122 dans S7.	Regroupez le contenu des OB23 et 24 dans un OB122 et effacez l'autre OB122.
L'OB a été interprété par l'AG115U comme un OB34.	phase 2	Selon la CPU employée, l'OB34 peut avoir des significations différentes.	Vérifiez que cet OB convient à votre programme.
'Masques DB S5' n'est plus utilisé pour le paramétrage de S7.	phase 1	Il y a MASK dans DW0 et dans DW1.	Paramétrez l'AP avec STEP 7.
L'opération de saut suivant l'opération 'B' ne	phase 2	Une opération de substitution suivie d'une	Remplacez l'opération par SPL

peut être convertie (utilisez SPL).		opération SPA ne peut être convertie automatiquement.	dans le fichier LIST et revoyez 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.	Effectuez le paramétrage du système dans la table de configuration.
Tenez compte des différentes opérations d'arrêt.	phase 2	Il n'est pas fait de différence entre STP, STS et STW.	Vérifiez 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.	Ajoutez 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 sont les formats de données qui manquent.	Examinez si les étiquettes se trouvent dans un autre fichier.
Dans le cas d'un AG115U, changez-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.

#### IV.2.7.Retouche du programme converti

- ✓ **Préparation**
- ✓ Les étapes de préparation suivantes sont nécessaires avant de retoucher le fichier source LIST généré:
  - Etudier 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 "Sources" du programme S7 que nous avons créé.
- Ouvrir le fichier converti.

#### ✓ Exécution

Pour pouvoir retoucher le fichier source LIST généré nous devons parcourir le programme en mode interactif et modifier ou compléter les blocs d'organisation et les opérations S5 non convertibles en nous aidant des messages émis.

#### ✓ Fonctions non convertibles

Les opérandes et opérations non convertibles figurent sous forme de commentaires dans le programme S7 généré, nous devons alors procéder à leur retouche.

Deux méthodes de conversion nous sont proposées :

- définir pour ces opérations et opérandes s'ils apparaissent dans le programme utilisateur des macro-instructions spécifiques en LIST S7 qui seront utilisées lors de la conversion.
- éditer les séquences d'instructions correspondantes dans le programme S7 résultant.

Le choix de la méthode dépend de la fréquence d'apparition de telles commandes dans notre programme utilisateur.

### IV.2.8.Compilation

Nous devons compiler le programme converti et retouché avec le compilateur LIST afin de le rendre exécutable.

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
- et l'existence des blocs appelés dans le programme.

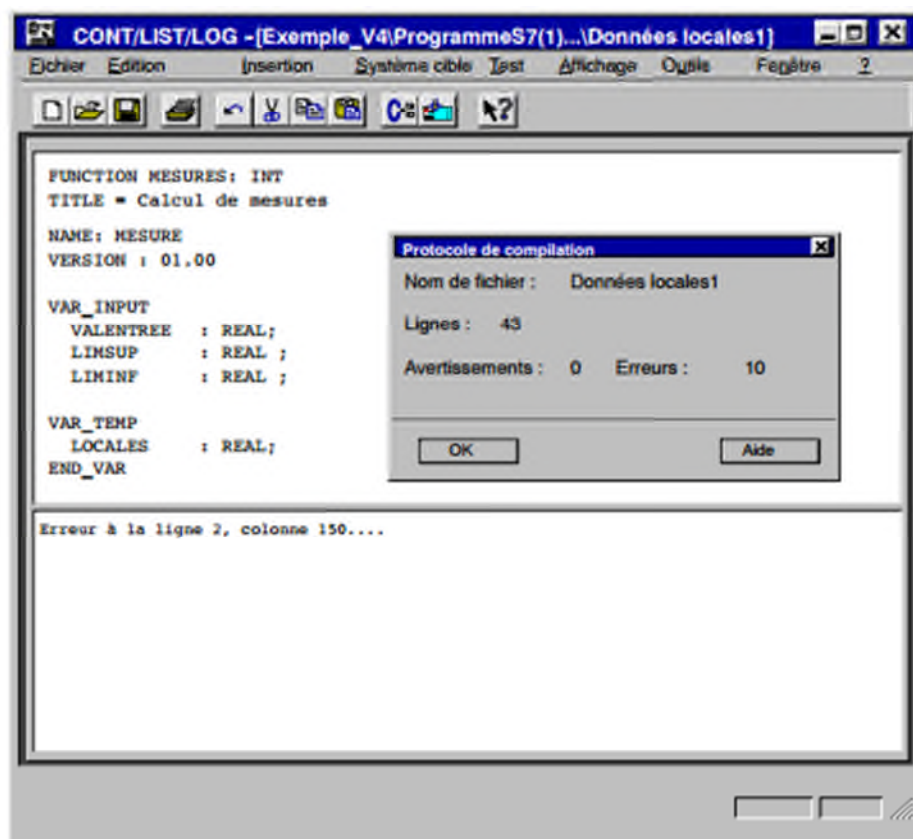
Nous obtenons ensuite un protocole indiquant le nom du fichier compilé, le nombre de lignes compilées ainsi que le nombre d'erreurs et d'avertissements.

#### Compilation du fichier source

La commande Fichier > Compiler nous permet de compiler notre fichier source afin de générer les blocs correspondants. Un protocole s'affiche après la compilation. Il indique les erreurs éventuelles comme après la vérification de cohérence.

En cas de programmation de plusieurs blocs dans un fichier source, sauf ceux sans erreur seront compilés et sauvegardés.

La figure IV-6 présente la fenêtre de vérification de la cohérence et compilation de fichiers source dans S7. [7]



**Figure IV.6.** Cohérence et compilation du fichier source dans S7.

#### IV.2.9. Correction des erreurs

Les erreurs ou les avertissements que pourrait contenir notre programme converti 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 précisée.

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. [8]

### **IV.3. conversion du programme S7-200 de la banderoleuse vers S7-300**

#### **IV.3.1. Introduction**

Pour avoir un seul automate qui gère les deux machines (banderoleuse et navette), il est nécessaire de faire un seul programme. Pour se faire, on garde le programme converti de la navette tel qu'il est et on fait des modifications des adresses des mémentos et les entrées et sorties de celui de la banderoleuse. Pour éviter un conflit entre les données, le bloc d'organisation OB1 sera complété par les réseaux de la banderoleuse et les fonctions doivent être créés.

#### **IV.3.2. Constitution du programme de S7-200**

Le programme de la navette contient 18 fonctions (*FC0 à FC17*) et celui de la banderoleuse existant dans le S7-200 contient 12 fonctions (*SRB\_0 à SRB\_11*) qu'on doit convertir à S7-300. Pour combiner les deux programmes, on est donc obligé de les créer en commençant par la fonction FC18.

#### **IV.3.3. Le choix du langage de conversion**

Le logiciel step7 est doté de trois langages de programmation : logigramme, liste et contact qui évoluent d'une gamme à une autre. Nous avons fait le choix de convertir notre programme en utilisant le langage à contacte pour nous faciliter la tâche. Car, on peut garder la même structure et faire la modification des instructions.

#### **IV.3.4. Modification des adresses**

Pour réussir à intégrer le programme de la banderoleuse dans celui de la navette, il est important de modifier les adresses des mémentos, entrées et sorties du programme pour éviter les conflits. Nous avons additionné 300 à chaque adresse des entrées et sorties et les mémentos de type M. Pour les mémentos de type V, chaque adresse est additionnée par 800. Ce qui va nous faciliter par la suite la réalisation de notre configuration matérielle

**IV.3.5. Les mémentos spéciaux**

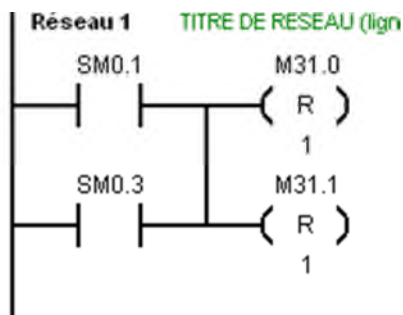
Le logiciel micro Win de S7-200 contient une série d’octets de mémentos prédéfinis en lecture seule appelés mémentos spéciaux SMB (0 à 29). On s’en sert généralement pour appeler des fonctions dans le programme.

Notre programme de la banderoleuse utilise les deux bits SM0.1 et SM0.3 de l’octet SMB0 qui est mis à jour par la CPU S7-200 à la fin de chaque cycle.

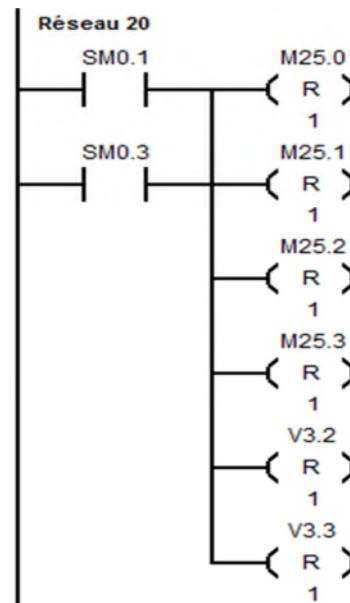
**SM0.1** Ce bit est à 1 au premier cycle. Il sert, entre autres, à l'appel d'un sous-programme d'initialisation.

**SM0.3** Ce bit est mis à 1 pour la durée d'un cycle si une mise sous tension entraîne le passage à l'état de marche (RUN). Il permet, par exemple, de fournir un temps de chauffe de l'installation avant de commencer l'exploitation.

Ils sont utilisés dans le réseau 1 du bloc d’organisation et le réseau 20 de la fonction SRB\_11 :



**Figure IV.7.** Réseau 1 de l’OB



**Figure IV.8.** Réseaux 20 de la fonction SRB 11

**Remarque :**

Au cours de la conversion, nous avons constaté que le S7-300 n’est pas doté de ces mémentos. Nous étions alors dans l’obligation de trouver une solution. On a mis dans l’OB100 deux réseaux. Le premier pour le **SM0.1** est illustré par la Figure IV-9. Le deuxième réseau pour **SM0.3** est donné par Figure IV-10



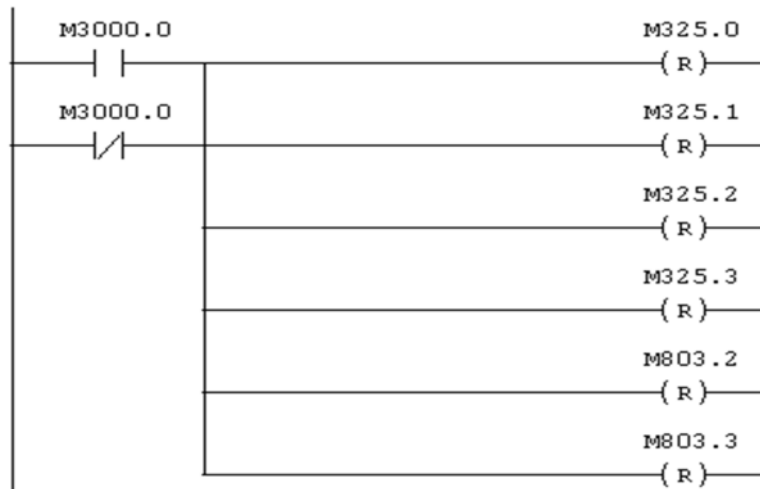


Figure IV.9. Equivalence de SM0.1 dans l'OB100.



Figure IV.10. Equivalence de SM0.3 dans l'OB100.

Le T2 qu'on a récupéré dans la fonction 25, réseaux 20 de Figure IV-11.

Réseau 20 : Titre :

Commentaire :

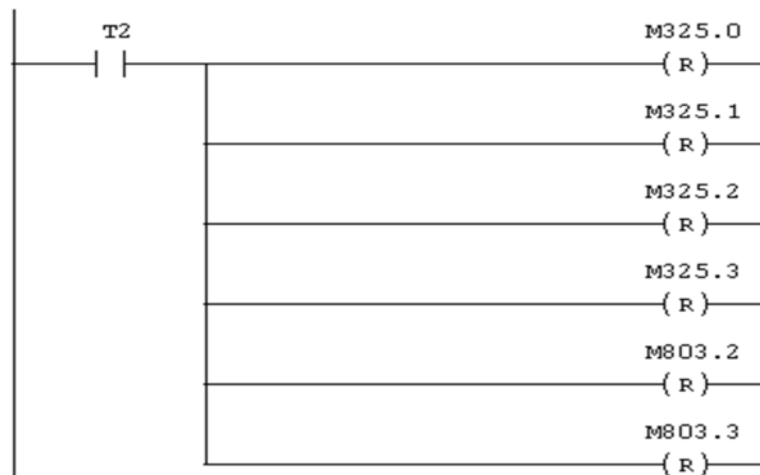


Figure IV.11. Utilisation de T2 dans le réseau 20 de la fonction 25.

#### IV.4. Vérification de la configuration matérielle de l'API

Afin de vérifier la configuration matérielle, on double-clique sur l'icône « matériel » dans la partie droite de l'écran :

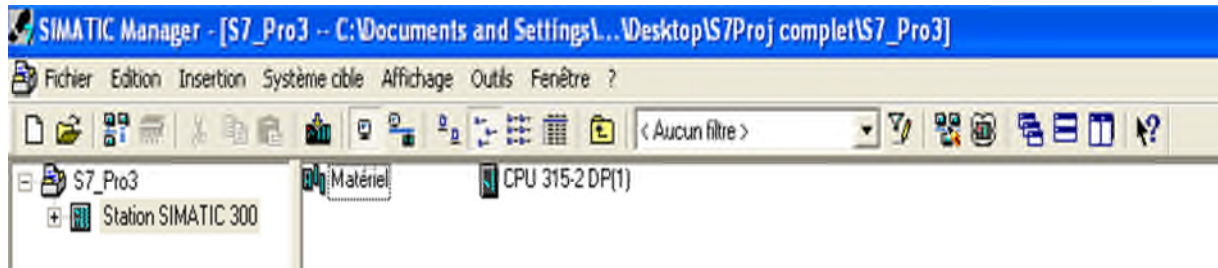


Figure IV.12. Accès à la configuration matérielle

La nouvelle fenêtre qui s'ouvre va nous permettre de configurer l'automate. Si la liste des composants (sur la droite) n'apparaît pas, cliquer sur l'icône « catalogue » (ou menu « Affichage », puis cocher « catalogue ») :

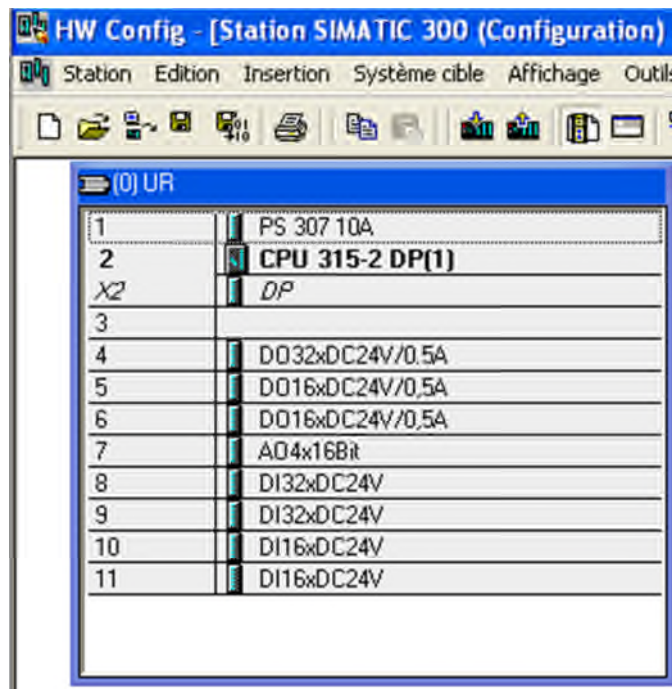


Figure IV.13. Configuration de l'automate.

- Le module alimentation **PS 307 10A** est placé sur le premier emplacement du rack.
- Le module Unité de Traitement **CPU 315 2 DP** occupe deux emplacements 2 et 3.

#### **IV.4.1. Modules pour la navette**

Le programme de la navette est constitué de :

- 3 modules de sorties TOR ou DO (digital output).
- 1 module de sorties analogiques ou AO (analogique output).
- 4 modules d'entrées TOR ou DI (digital input).

-Le module de sorties TOR ou DO (digital output) occupe l'emplacement 4. Ce module comporte **32 entrées TOR à 24VDC**

-Les modules de sorties TOR ou DO (digital output) occupent les emplacements 5 et 6. Ces modules comportent **16 sorties TOR** chacun.

- Le module de sorties analogiques ou AO (analogique output) occupe l'emplacement 7. Ce module comporte 4 entrées de 16 bits

-Les modules d'entrées TOR ou DI (digital input) occupent les emplacements 8, 9. Ces modules comportent 32 entrées TOR chacun.

-Les modules d'entrées TOR ou DI (digital input) occupent les emplacements 10, 11. Ces modules comportent 16 entrées TOR chacun

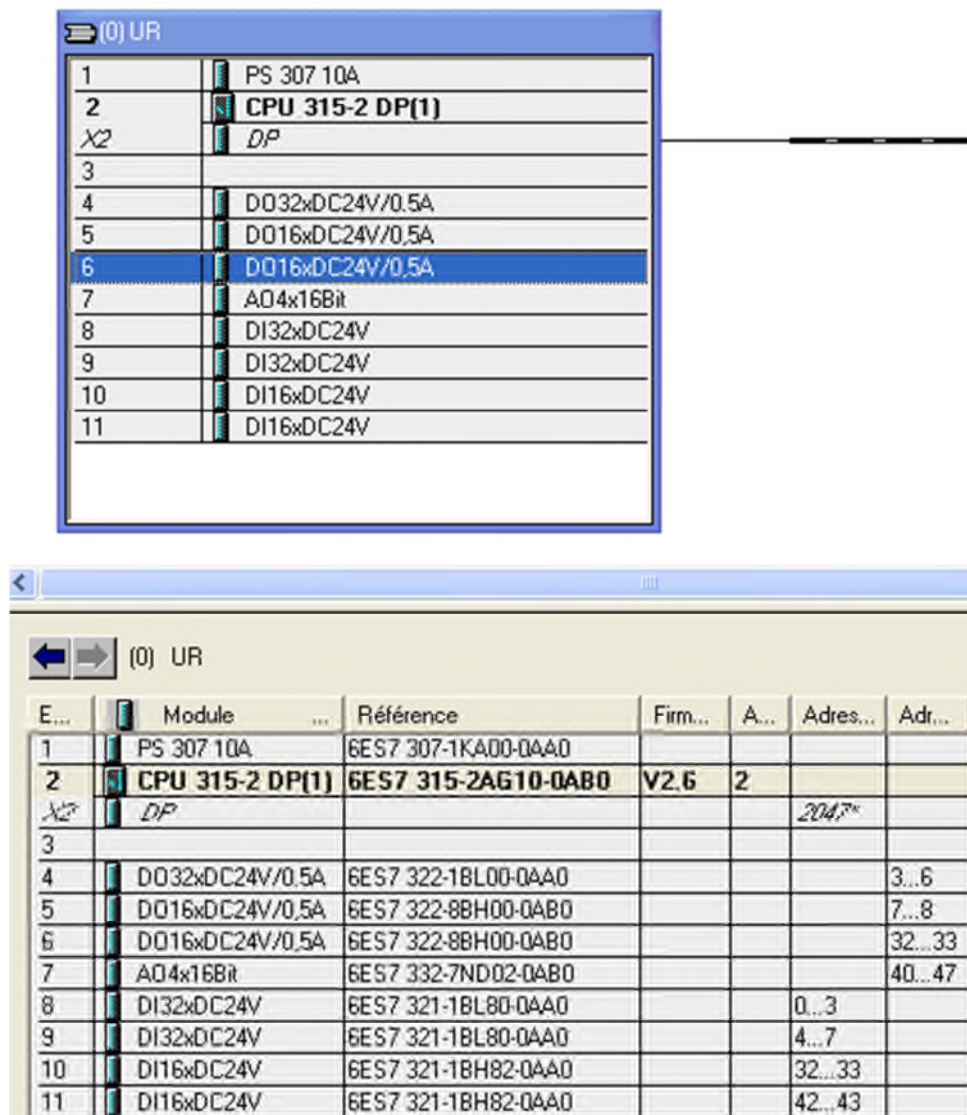


Figure IV.14. Modules d’entrées et sorties pour la navette.

#### IV.4.2.Réseau PROFIBUS pour la banderoleuse

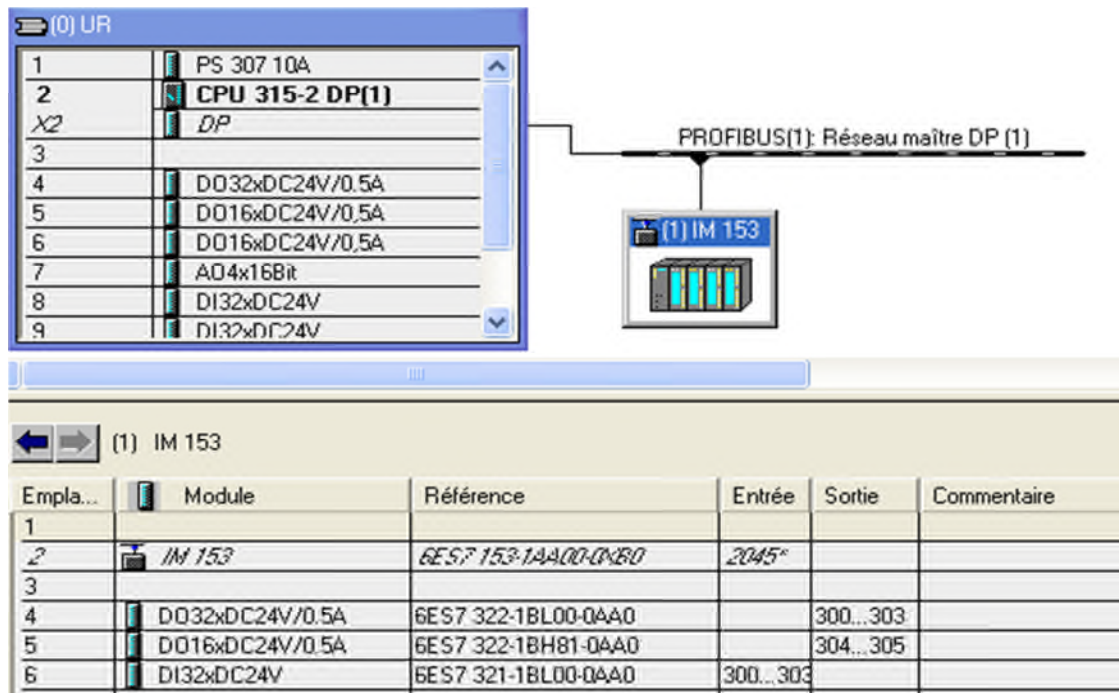
##### ➤ PROFIBUS

Bus de terrain pour la mise en réseau d’équipements de production industrielle, qui assure l’interconnexion entre des automates industriels et des équipements de terrain (banderoleuse).

- Vitesse de transmission: 9,6 Kbits/s à 12 Mbits/s
- Long. max. du bus 100 m à 12 Mbits/s, 200 m à 1 500 Kbits/s, 1 200 m à 93,75 Kbits/s. [16]

Le réseau PROFIBUS qu'on a ajouté est constitué de 2 modules de sorties et 1 module d'entrées ;

- Le module de sorties TOR ou DO (digital output) comportant **32 sorties TOR**
- Le module de sorties TOR ou DO (digital output) comportant **16 sorties TOR**
- Le module d'entrées TOR ou DI (digital input) comportant **32 entrées TOR**



**Figure IV.15.** Module d'entrées et sorties pour la banderoleuse.

## IV.5.Conclusion

Ce dernier chapitre expose d'une manière détaillée le travail réalisé de migration et de conversion de deux programmes, de la banderoleuse et de la navette, afin d'utiliser un seul automate S7-300 géré par un seul programme pour commander les deux tâches à la fois, de l'évacuation des palettes par la navette à la plastification par la banderoleuse des palettes évacuées.

# Conclusion général

## Conclusion générale

Ce travail de master est dédié à la migration et la conversion de deux automates programmables de générations différentes. Pour atteindre cet objectif, une recherche bibliographique intensive et élargie nous a permis à nous familiariser avec les automates s5 et s7. La seconde étape consistait à réaliser la migration S5/S7. L'application en question revient à reprogrammer un procédé sous S7-300, dont les deux programmes originales sont écrits et implémentés sous S5-95U et S7-200. Le procédé est un système de plastification des palettes pleine par la banderoleuse après leur évacuation par la navette.

Nous avons effectué une migration du programme de l'automate siemens S5-95U vers le S7-300 pour la navette et la conversion du programme de la banderoleuse de l'automate S7-200 vers le même automate S7-300. Comme le programme S7-300 ne possède pas des mémentos spéciaux, une solution a été proposée par l'utilisation du bloc d'organisation de démarrage. Cette opération peut être vue comme une création (programmation d'un mémento spéciale) pour S7-300. Le programme de la navette, existant dans l'automate S5-95U, a été converti à l'aide du convertisseur S5/S7 intégré dans le S7-300. La conversion du deuxième programme de la banderoleuse, programmé avec l'automate S7-200, a été effectuée directement (sans convertisseur). Par la suite, nous avons procédé à la correction des erreurs et la configuration matérielle.

Ce travail nous a permis, entre autres, d'enrichir nos connaissances et notre savoir-faire dans l'automatisme industriel. La thématique abordée, nous a confrontés à plusieurs problèmes interdisciplinaires toujours en relation avec l'automatisme.

## Recherche Bibliographique

- [1] Notice d'information, Visa COSOB N° 05-07 du 21 décembre 2005.
- [2] MERSEL Riadh, SACI Djilali: « étude technologique d'une banderoleuse automatique avec plateau tournant à rouleaux Mod.095 », techniciens supérieurs de l'institut National spécialisés de la formation professionnelle de oued semar Alger, juin 2007.
- [3] Philippe LE BRUN, Automates programmables industriels.
- [4] Alain GONZAGA, les automates programmables industriels.
- [5] Yoann GAUTHIER : « programmation d'une cellule robotisée de soudage », ingénieur mécatronicien de l'institut National des Sciences Appliquées de Strasbourg, septembre 2012.
- [6] UGERGAZI Khaled, SAHRAOUI Hamza « étude de la tréfileuse et sa commande automatique», thèse d'ingénieur de Electrotechnique, Biskra 2006.
- [7] BEL MOKADEM Houda, « vérification des propriétés temporisées des automates programmable industriel», thèse de docteur de l'école normale supérieure de CACHAN, septembre 2006.
- [8] SIEMENS STEP 7 Pour une transition facile de S5 à S7, référence : 6ES7810-4CA07-8CW0, Edition 01/2004.
- [9] automates programmables siemens : SIMATIC S5 niveau 1.
- [10] Siemens Automation Innovation, Subject to change without notice / © Siemens SAS 2009.
- [11] Pierre Duysinx, Geoffray Hutsemekers, Henri Lecocq, automatisation et robotisation de la production, Année académique 2009-2010.
- [12] Alain MALVOISIN, notice de programmation d'automates siemens S7-300 S7-400
- [13] siemens logiciel système pour SIMATIC S7-300/400 - Fonctions standard et fonctions système –Volume ½ , référencée: 6ES7810-4CA10-8CW1.
- [14] Cours S7MA2, SITRAIN Formation, Automatisation et entraînements, SIMATIC S7, Siemens, 22.07.03, Fichier : 03\_MA2\_FR.1
- [15] Manuel d'utilisation de la navette, document TMGimpianti, Novembre 2000.
- [16] Présentation du PROFIBUS, SITRAIN Formation, Automatisation et entraînements, SIMATIC S7, Siemens, 06.05.2009, Fichier : PROFIBUS\_01fr.1



## **Résumé**

Ce travail de master, réalisé au sein de CEVITAL, est dédié à la conversion et la combinaison de deux programmes différents de deux machines (banderoleuse et navette). Chacune est gérée par son propre automate d'anciennes générations. L'objectif consiste à réaliser un programme unique qui assure la mise en marche des deux machines à la fois sur un automate plus récent.

Pour la navette, nous avons effectué une migration du programme de l'ancien automate siemens S5-95U vers le S7- 300. Le programme de la banderoleuse, existant sur le S7-200, a été aussi converti en S7- 300. Enfin, les deux ont été combinés dans un seul et unique programme implémenté sur l'automate S7-300.

## **Abstract**

This Master dissertation, performed within CEVITAL company, is dedicated to the conversion and the combination of two different programs of two machines (banderoler and shuttle). Each one is managed by its own automaton of former generations. The aim is to develop a single program that insures the operation of the two machines, using only one recent automaton.

We made a migration of the S5 program of the old automaton Siemens S5-95U, towards S7-300 for the shuttle. The same work has done for the banderoler S7-200 program. Both resulted programs have been gathered in a single code implemented on S7-300 automaton.