

République Algérienne Démocratique et Populaire  
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique  
Université ABDERRAHMANE MIRA de Bejaia



Faculté de technologie  
Département Génie électrique



# MEMOIRE DE FIN DE CYCLE

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

*Option : Automatismes Industriels*

*Thème*

**MIGRATION D'UN AUTOMATE PROGRAMABLE S5  
VERS S7 DU RACCORDEUR CTS  
AU SEIN DE GENERAL EMBALLAGE**

**Présenté par :**

*Mr TALA IGHIL Mohand Ou idir*

**Encadré par :**

*Mr ACHOUR.A/Y  
Mr LAOUCHICHE.L*

*Promotion 2012 - 2013*

# Remerciement

*Le grand merci s'adresse au Bon Dieu le tout puissant, qui nous a donné la patience et le courage pour achever ce travail.*

*J'adresse tous mes sincères et respectueux remerciements à mon promoteur, Mr ACHOUR A/Y pour son dévouement et sa disponibilité durant la préparation de ce mémoire.*

*Je tiens à remercier particulièrement mon encadreur à **GENERAL EMBALLAGE** Mr LAOUCHICHE LOUHAB. Je suis aussi très reconnaissant aux membres de jury qui nous feront l'honneur d'accepter de juger ce travail, d'apporter leurs réflexions et leurs critiques scientifiques.*

*Ma reconnaissance s'adresse à tous ceux qui ont contribué de près ou de loin pour que ce modeste travail puisse voir le jour.*



# Dédicaces

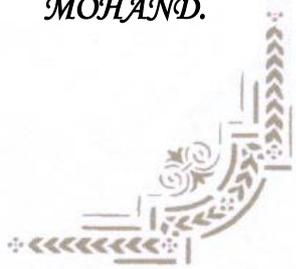


*Je tiens à dédier ce mémoire :*

*A la mémoire de mon Père et mon frère L'HASSEN, et à ma très chère Mère, 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 : belkacem elghani, Noureddine, fatah et Khaled*
- *A mes sœurs Aziza, Seltana et leurs familles*
- *A toute la famille sans oublié mes neveux Lounis et Salim*
- *A mes amis de la promotion sortante 2013*
- *A mes chers amis et copains de chambre: Azzedine, Mouloud, Samir, Bachir, et a Aissa, hamza, Adel, toufik et sa fiancé Noura, et tous mes amis du CSA et TTT de la RUTO et mes amis de la chambre H10 : Zahir, fares Toufik et Redouane*
- *A la mémoire de lounes MATOUB.*
- *Ceux qui m'ont soutenu pendant toute la durée de mes études*

**MOHAND.**



## Table des matières

## *Table des matières*

Table des matières.....	IV
Liste des figures .....	VII
Liste des tableaux.....	IIIX
Introduction général .....	1
Présentation de l'entreprise.....	3

### **Chapitre I : généralités**

Introduction.....	5
I.1.Définition d'un automate programmable .....	5
I.2.Aspect extérieur d'un API .....	5
I.3.Architecture interne d'un API .....	7
I.4. Principaux éléments d'un API.....	8
I.5. Le simulateur de programme PLCSIM .....	10
Conclusion.....	11

### **Chapitre II: Etude et présentation du Raccordeur CTS et formulation de problématique**

Introduction.....	12
II.1. Description de la machine à étudier .....	14
II.2.Elément de composition de la machine .....	15
II.3.Mode de fonctionnement.....	17
II.4.Séquence de raccordement .....	17
II.5.Régulation de la tension de système.....	19
II.6 .Fonctionnement de système de régulation .....	19
II.7 .Positionnement de problème .....	20
Conclusion.....	21

### **Chapitre III: Etude comparative entre l'API S5 et S7**

Introduction.....	22
III.1.Présentation de l'automate programmable S5-95U .....	22
III.1.1.Modules de périphérie de l'API.....	23
III.1.2.Unités fonctionnelles.....	24

III.2.Elaboration d'un programme .....	26
III.3.1.Stratégie de la firme SIMENS pour la série S5 .....	26
III.3.2 .Solution pour remplacer SIMATIC S5 .....	27
III.4.1.Présentation des automates programmable SIEMENS S7300.....	28
III.4.2.Présentation des déférentes CPU des automates S7-300 .....	29
III.4.3.Utilisation de S7-300 .....	31
III.4.4.Elaboration d'un programme .....	32
III.4.5.Fonctions du logiciel STEP 7.....	32
III.5.Interaction du logiciel et du matériel .....	33
III.5.1.Création d'un projet.. .....	34
III.5.2.Paramétrage des modules S5/S7 .....	36
III.6.Mise en parallèle des blocs STEP 5 et blocs STEP .....	37
Conclusion.....	38

## **Chapitre IV : Conversion de S5 vers S7**

Introduction.....	39
IV.1.Première approche .....	39
IV.1.1.Procédure complète de conversion .....	40
IV.1.2.Exécution de la conversion .....	41
IV.1.3.Localisation des erreurs .....	43
IV.1.4.Messages d'erreur .....	43
IV.1.5.Analyse et interprétation des messages .....	45
IV.1.6.Retouche de programme .....	46
IV. 1.7.Compilation.....	46
IV.1.8.Vérification de cohérence.....	46
IV.1.9.Correction des erreurs.....	46
IV.2.La deuxième approche.....	47
IV.2. Cahier des charges.....	47
IV.2.2.Tableau récapitulatif de l'entrées/sorties.....	48
IV.2.3.Grafcet principale de la machine.....	49
IV.2.4.Grafcet de sécurité.....	52
IV.2.5.Programmation et simulation.....	52
Conclusion.....	67

Conclusion générale .....	70
Bibliographie	
Annexes	

### Liste des figures

- Figure I.1 : Automate compact
- Figure I.2 : Automate modulaire
- Figure I.3. : Architecture interne d'un API
- Figure I.4 : Interface de simulation PLCSIM
- Figure II.1 : Processus de fabrication du carton ondulé à la ligne 'onduleuse'.
- Figure II.2 : Raccordeur CTS
- Figure II.3: Unité de raccordement (vue général).
- Figure II.4. Raccordeur CTS.
- Figure II.5 : Le système coupe-colle.
- Figure II.6 : Séquences de raccordement.
- Figure. II.7 : Rouleau de régulation
- Figure II.8 : Système d'équilibrage des forces
- Figure III.1 : Automate-SIEMENS-S5-95U
- Figure. III.2 : Unités fonctionnelles de L'API S5-95U
- Figure III.3 : Cycle de vie de série S5
- Figure. III.4 : API S7-300
- Figure. III.5 : CPU de différents niveaux de performance de l'automate S7-300
- Figure III.6 : Interaction du logiciel et du matériel
- Figure IV.1 : Image-écran initiale du convertisseur S5/S7
- Figure. IV.2 : Boîte de dialogue ' Conversion de fichiers S5'
- Figure IV.3 : Première phase de la conversion
- Figure IV.4 : Messages affichés par le convertisseur
- Figure. IV.5 : chariot du Raccordeur CTS
- Figure IV.6 : Grafcet de la station gauche
- Figure IV.7 : Grafcet de la station droite
- Figure.IV.8 : Grafcet de sécurité
- Figure. IV.9 : Fenêtre SIMATIC Manager Figure.
- Figure IV.10 : Fenêtre nouveau projet.
- Figure. IV.11 : Fenêtre pour éditer les mnémoniques.
- Figure. IV.12 : Fenêtre pour le choix du langage de programmation.



## Liste des tableaux

Tableau III.1 : Tableau explicatif d'utilisation de S7 300

Tableau III.2 : Paramétrage des modules S5/S7

Tableau III.3 : Mise en parallèle des blocs STEP 5 et blocs STEP 7

Tableau IV.1 : Messages d'erreur, signification et remède

Tableau IV.2 : Les différents entrées et sorties du '*raccordeur CTS*'

Tableau IV.3 : Editeur de mnémonique

## Introduction générale

## Introduction générale

Les premiers automates programmables ont été introduits en 1969 aux États-Unis pour satisfaire les besoins de l'industrie automobile. Le but recherché était de remplacer les armoires à relais utilisées pour l'automatisation des chaînes de production par des équipements moins coûteux et, surtout, plus faciles à modifier.

Depuis leur apparition, les automates programmables, se sont répandus très rapidement dans l'industrie. Aujourd'hui, ils représentent la moitié des équipements informatiques qui sont utilisés pour ce type d'applications. Un tel succès est dû, en grande partie, à leur faible coût, et la facilité de mise en œuvre par du personnels pas forcément formés en l'informatique [1].

Dans le monde industriel, les performances demandés ne se limitent pas à l'augmentation de la productivité, l'amélioration de la qualité du produit ou la diminution des coûts de production, mais, aussi à l'amélioration des conditions de travail, la suppression des tâches pénibles et répétitives et, surtout, une sécurité plus élevée.

Avec la progression continue de la technologie, l'automate programmable industriel (API), qui est la pierre angulaire des installations automatisées, apporte la solution attendue pour les besoins d'adaptation et de flexibilité de nombre d'activités économiques actuelles. On le trouve non seulement dans tous les secteurs de l'industrie, mais aussi dans le service tel que la gestion de parking, accès à des bâtiments, sécurité etc.

Les exigences attendues de l'automatisation ne cessent de s'accroître.

L'API placé dans un procédé industriel, fait partie intégrante de la boucle de réglage. Il a pour tâche principale de récolter des informations à partir des capteurs via ses interfaces d'entrées, de traiter ces informations pour prendre une décision en fonction du programme implémenté dans sa mémoire et ainsi commander les actionneurs avec des signaux via ses interfaces de sorties.

De nombreux constructeurs d'automates programmables existent, mais la firme allemande SIEMENS offre l'une des plus grandes gammes de produits. Telles que, SIMENS avec toutes ces séries : S5-90U/95U ; S7 200 ; S7 300 ; S7 400 ; S1200.

La plupart des anciennes installations industrielles utilisent l'automate S5-90U/95U qui est un automate conçu pour des applications de faible ou de moyenne envergure [2]. 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 du point de vue : fiabilité, performance et disponibilité des pièces de rechange. Par conséquent, une migration du modèle S5 vers S7 s'impose.

Le manuscrit est composé de quatre chapitres.

Le premier chapitre porte des généralités sur les API. Le second est consacré pour l'étude du processus industriel en question où la problématique est posée. Le troisième chapitre présente une étude comparative entre les API S5 et S7. Le dernier chapitre de ce rapport (chapitre IV) traite la partie de migration de l'API S5 vers un API S7. Les étapes de migration, qui font l'objet de notre travail sont détaillés et expliqués. En fin, on termine par une conclusion générale.

Présentation de l'entreprise

## **Présentation de l'entreprise**

SARL Général Emballage est leader confirmé au niveau national de l'industrie du carton ondulé, fondée en Aout 2000 à la zone industrielle TAHARACHT Akbou, véritable carrefour économique de Bejaia.

Actuellement, elle entre en production sur trois sites industriels (Akbou, Oran et Sétif). Elle est entrée en exploitation en 2002, et a obtenu le Trophée de la Production (Euro-Développement PME) en 2007. En raison de son développement. Général Emballage débute son exportation vers Tunis en 2008.

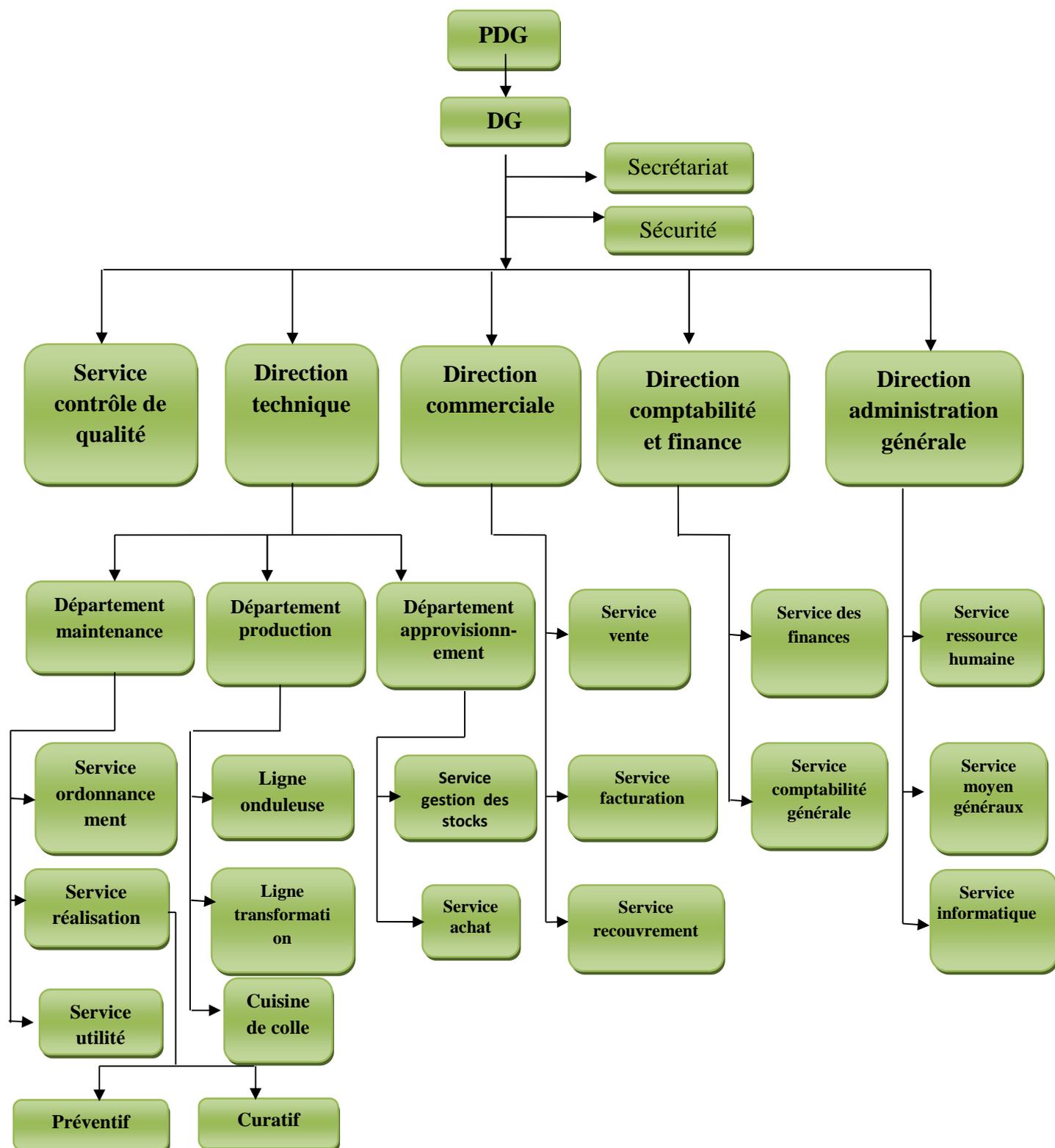
Général Emballage réalise depuis quelques années déjà une croissance à deux chiffres. Les indicateurs de performance de l'entreprise n'ont jamais été aussi mondiaux de certification en Algérie. Dans le cadre de partenariat, l'entreprise signe un accord professionnel avec l'Université de Bejaia en Juillet 2012.

Général Emballage a su au cours des années mettre son expérience au service du client grâce à une recherche permanente de l'efficience qui s'est traduite en 2013 par la certification ISO 9001 : 2008 de son système de management de la qualité.

## **Situation géographique de l'entreprise**

L'entreprise GENERAL EMBALLAGE est implantée dans une zone industrielle TAHARACHT Akbou, véritable carrefour économique de Bejaia. Située à deux Kilomètres d'une grande agglomération (Akbou) et à quelque dizaine de mètres de la voie ferrée. L'entreprise est à 60 km de Bejaïa.

## Organigramme de l'entreprise



Organisation général de Général Emballage

Chapitre : I  
Généralités

## Introduction

Les API sont particulièrement conçues pour répondre aux multiples applications dans la quasi-totalité des domaines industriels, ce sont des outils programmables universels.

En plus des fonctionnalités de la logique câblée (ET, OU...), elles permettent de traiter les fonctions particulières telles que [1].

- Comptage et calcul,
- Mesures analogiques et régulation,
- Communication - supervision.

La puissance principale d'un automate réside dans sa grande capacité de communication avec l'environnement industriel. Outre, son unité centrale et son alimentation, l'automate est constitué essentiellement de modules d'entrée/sortie, qui lui servent comme moyen de communication avec le processus industriels à conduire.

L'automate possède également de grandes capacités de dialogues homme-machine et d'autres systèmes similaires.

Il présente une grande souplesse de programmation, grâce à des méthodes directes, via une console ou un microordinateur avec des langages de programmations spéciaux très adaptés.

### I.1 Définition d'un automate programmable

Un automate programmable industriel (*API*) est une forme particulière d'automate à base de microprocesseurs qui se fonde sur une mémoire programmable pour enregistrer les instructions et mettre en œuvre des fonctions, qu'elles soient logiques, de séquençement, de temporisation, de comptage ou arithmétiques, pour contrôler des machines et des processus. Il est conçu pour être manipulé par des ingénieurs ayant, potentiellement, une connaissance limitée en informatique et en langage de programmation [1].

## I.2 Aspect extérieur d'un API

Les automates peuvent être de type compact ou modulaire [1].

- **Type compact**

On distingue les modules de programmation (*LOGO de Siemens, ZELIO de Schneider, MILLENIUM de Crozet*) ce sont des micros automates. 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 analogique . . . etc.*) et recevoir des extensions en nombre limité. Ces automates, de fonctionnement simple sont généralement destinés à la commande de petits automatismes [1].

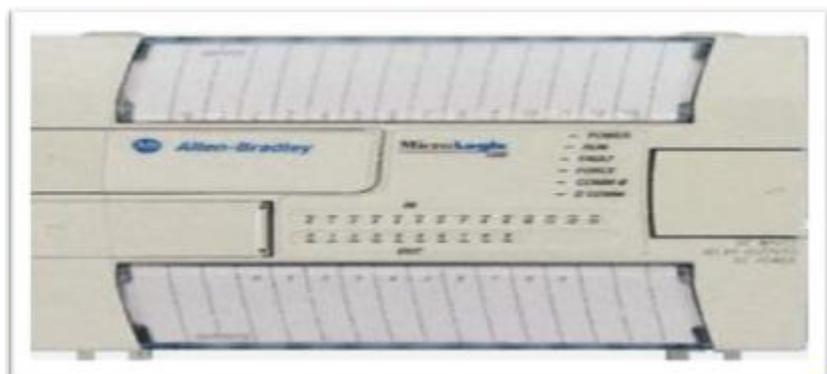


Figure I.1 : Automate compact [5]

- **Type modulaire**

On trouve le processeur, l'alimentation et les interfaces d'entrées /sorties résident dans des unités séparées (*modules*) et sont fixées sur un ou plusieurs racks

Ces automates sont intégrés dans les automatismes complexes ou puissant, nécessitants une capacité de traitement et flexibilité sont nécessaires [1].

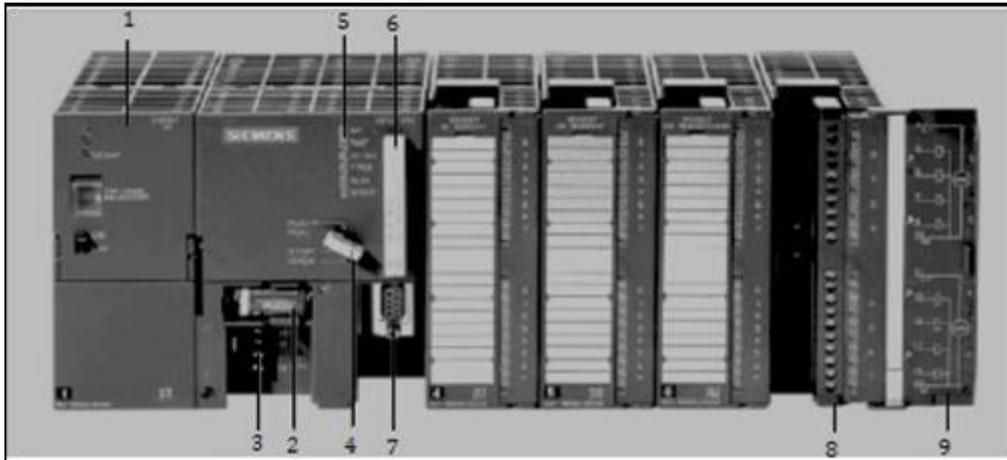


Figure I.2. Automate modulaire [9]

- 1-Module d'alimentation
- 2- Pile de sauvegarde
- 3- Connexion au  $24V_{cc}$
- 4- Commutateur de mode
- 5- LED de signalisation d'état et de défauts
- 6- carte mémoire
- 7- interface multipoint(*MPI*)
- 8- Connexion frontal
- 9- volet en face avant

### I.3. Architecture interne d'un API

En général, un API est constituée de composants fonctionnels de base suivante [3]

- Une unité de traitement,
- La mémoire,
- Une unité d'alimentation,
- Des interfaces d'entrées-sorties,
- Une interface de communication

## I.4. Principaux éléments d'un API

- **La CPU**

L'architecture interne d'un API dépend du microprocesseur employé. En générale les CPU sont constitués des éléments suivants [3] :

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

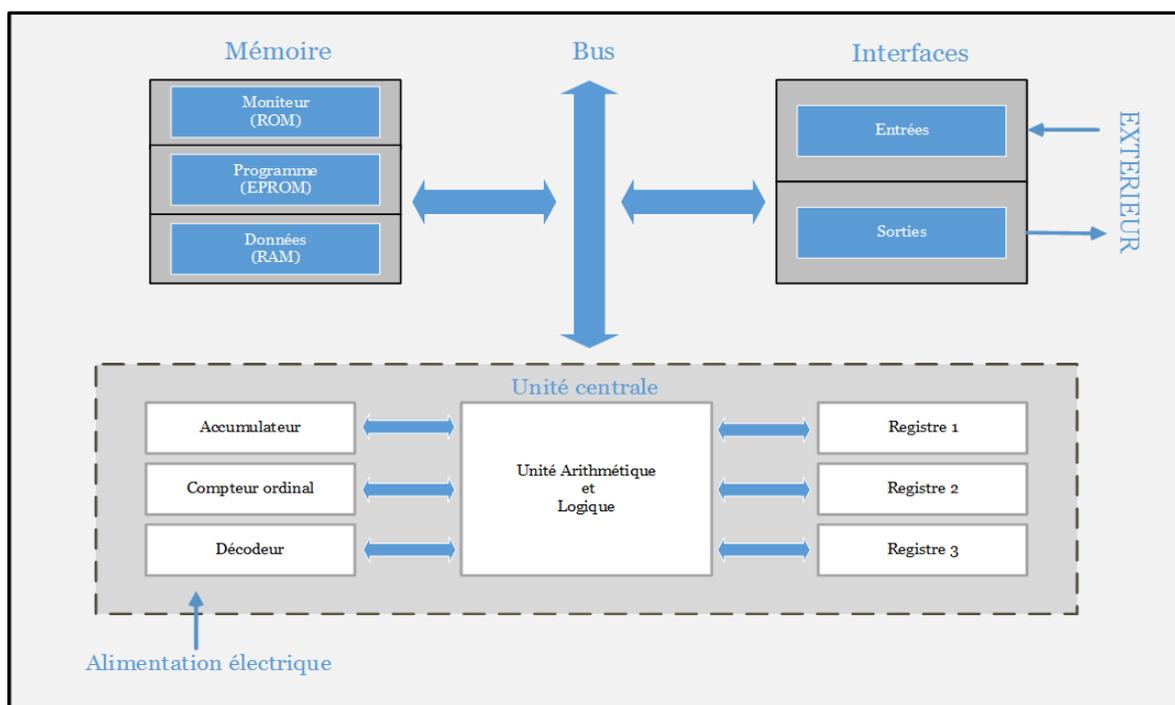


Figure I.3 : Architecture interne d'un API [4]

- **Les bus**

- **Le bus de données**

Transporte les données utilisées dans les traitements effectués par la CPU.

- **Le bus d'adresse**

Transporte les adresses de l'emplacement mémoire. Pour que chaque mot puisse être localisé en mémoire. Chaque emplacement procède une adresse unique et chaque emplacement de mot possède une adresse que la CPU utilise pour accéder aux données enregistrées, que ce soit pour les lire ou pour les écrire.

- **Le bus de control**

Transporte les signaux utilisés par la CPU pour le contrôle .Il sert, par exemple, 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.

- **Le bus système**

Sert aux communications entre les ports d'entrées-sorties et l'unité d'entrées-sorties et l'unité d'entrées-sorties [4].

- **La 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 de l'API qui est composée de plusieurs éléments :

- **Les mémoires mortes (ROM)**

Pour certaines applications, il est nécessaire de pouvoir conserver des informations de façon permanente même lorsque l'alimentation électrique est interrompue. On utilise alors des mémoires mortes ou mémoires à lecture seule (ROM : Read Only Memory). Ces mémoires sont non volatiles. Contrairement aux RAM, Ces mémoires, ne peuvent être que lue.

Suivant le type de ROM, la méthode de programmation changera. Il existe donc plusieurs types de ROM : ROM, PROM, EPROM, EEPROM, FLASH EPROM

. Une mémoire morte reprogrammable (EPROM, Erasable and Programmable Read Only Memory) est parfois employée pour stocker de manière permanente les programmes [5].

### ➤ Les mémoires vives (RAM)

La mémoire vive (RAM, Random Access Memory) est utilisée pour le programme de l'utilisateur.

### • Alimentation électrique

L'alimentation électrique qui a pour rôle de fournir les tensions continues que nécessitent les composants (5 V, 12 V) avec de bonnes performances, notamment face aux microcoupures du réseau. Sa source d'énergie est normalement le réseau électrique, parfois du 24 V continu. [5].

## I.5. Le simulateur de programme PLCSIM

L'application de simulation de modules S7-PLCSIM permet d'exécuter et de tester le programme sans automate Programmable industriel (API). La simulation étant complètement réalisée au sein du logiciel

STEP7, il n'est pas nécessaire qu'une liaison soit établie avec un matériel S7 quelconque ( CPU ou module de signaux). L'API S7 de simulation permet de tester des programmes destinés aux CPU S7-300 et aux CPU S7-400.

S7-PLCSIM dispose une interface simple permettant de visualiser et de forcer les différents paramètres utilisés par le programme (par exemple : activer ou désactiver des entrées). Tout en exécutant le programme dans l'API de simulation.

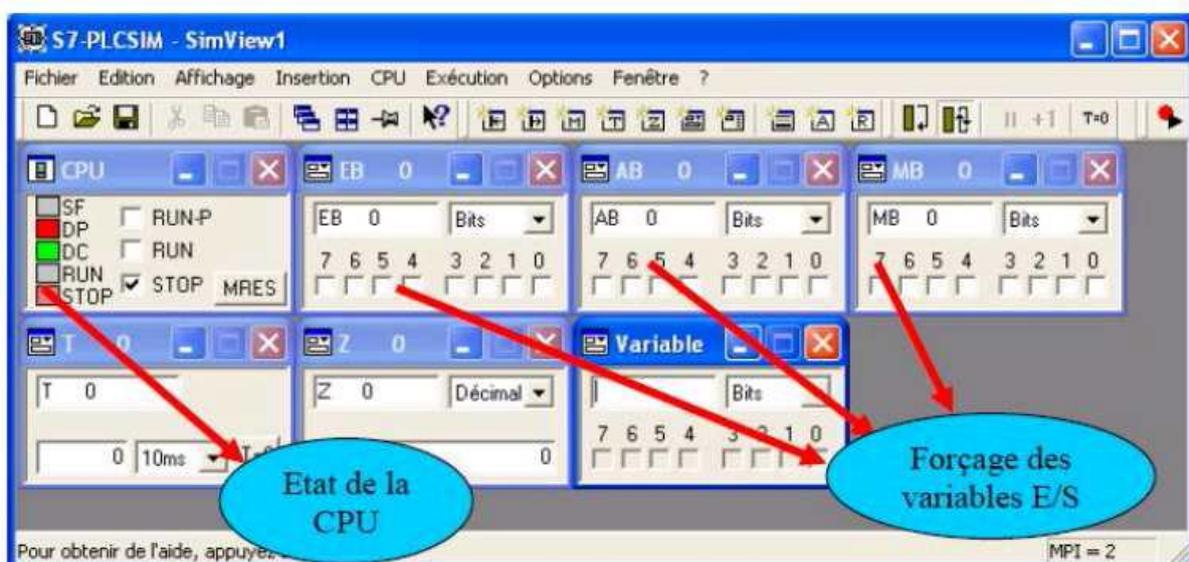


Figure I.4 Interface de simulation PLCSIM [9].

## Conclusion

L'API est une prouesse technologique, facile à programmer, à enficher et il est bien adapté aux conditions industrielles, il remplace l'homme dans des opérations ;

- Dangereuses
- Répétitives ou pénibles,

Les systèmes automatisés permettent d'augmenter

- La précision donc une meilleure qualité.
- La productivité avec une main d'œuvre réduite.

L'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 actionneurs à partir d'informations logique, analogique ou numérique.

## **Chapitre : II**

**Etude et présentation du raccordement CTS et formulation  
de la problématique**

## Introduction

Le carton ondulé est le matériau idéal pour faire un emballage très efficace grâce à son aptitude à tenir et à être gerbé. Vu l'importance de l'industrie de l'emballage, « GENERAL EMBALLAGE » qui est une entreprise spécialisée dans la fabrication du carton ondulé a opté pour des technologies nouvelles dans ce domaine.

L'Onduleuse est une machine sophistiquée d'une longueur de 125m et d'une largeur de 2 à 2.6 m (figure 2.1) qui fabrique le carton ondulé et qui consiste à assembler 3 papiers : Un papier cannelure (ondulé mécaniquement), une couverture extérieure et une couverture intérieure.

L'onduleuse est juxtaposée de plusieurs machines, synchronisées entre elles et qui constituent une chaîne cinématique parfaite [6]:

- *Les Raccordeurs* : C'est des dérouleurs des bobines équipées d'un système de freinage automatique, garantit le défilement du papier à une vitesse constante et qui raccordent le papier lors du changement de la bobine.
- *Une Nappe simple face* : Elle colle le papier sur la cannelure.
- *Une Nappe double face* : Elle colle un papier externe sur la simple face.
- *Une Partie dimensionnelle* : Elle est composée d'une traceuse longitudinale « CLM350 CL » et coupeuse transversale « CTM 350 ».

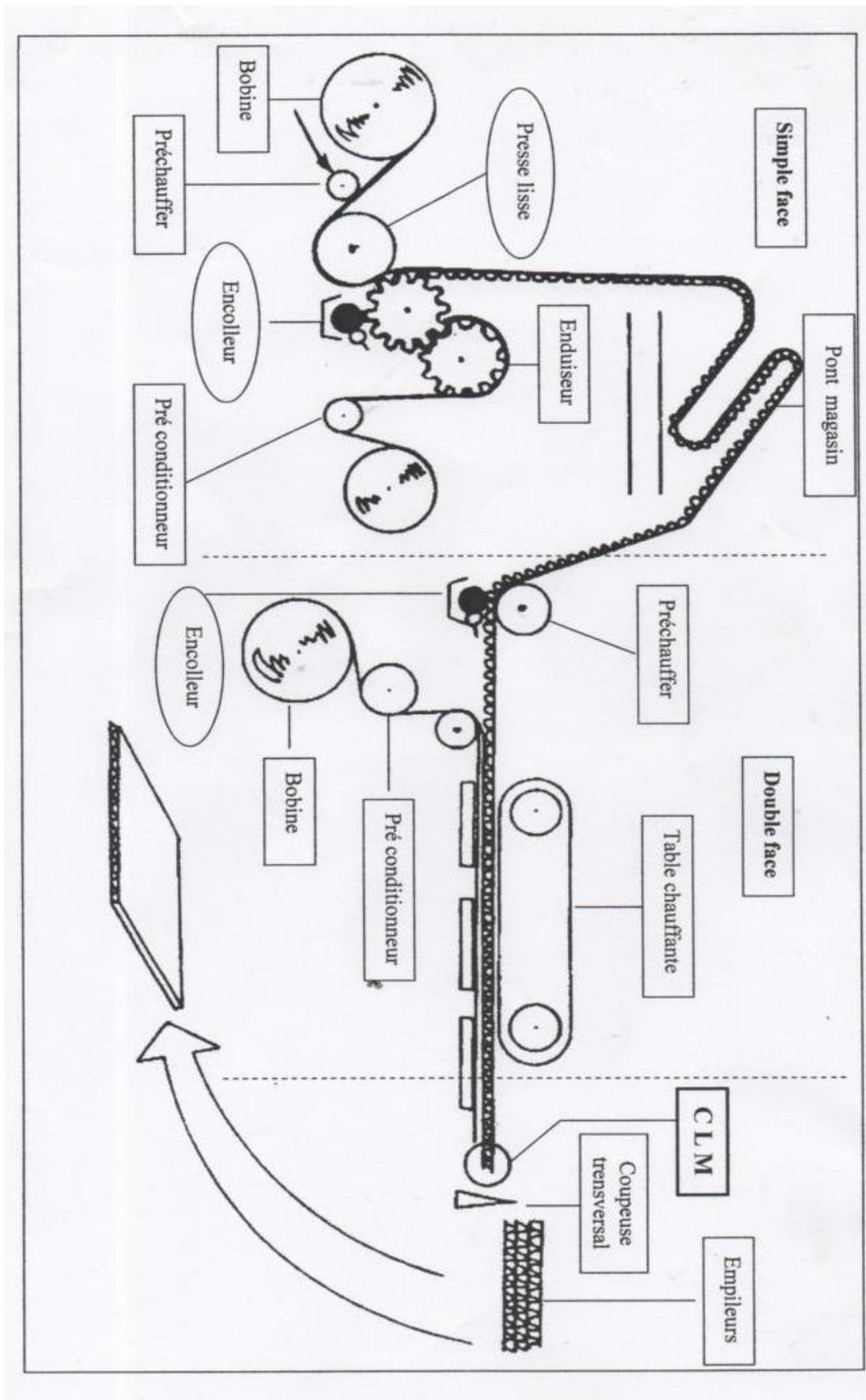


Figure II. 1: Processus de fabrication du carton ondulé à la ligne 'onduleuse'.

## II.1. Description de la machine à étudier

Le « *Raccordeur CTS* » (figure II.2) est une machine conçue pour travailler dans le débobinage continu des bandes de papier. La tension de la bande doit être maintenue, à tout moment, constante, y compris pendant les accélérations et décélérations de la ligne de production et lors de la séquence de raccordement d'une bobine avec une autre. Le processus de fabrication du carton ondulé nécessite l'assemblage de trois papiers :

- Un papier cannelure, ondulé mécaniquement
- Une couverture intérieure et une autre extérieure

Une fourniture continue du papier est impérative afin d'éviter tout éventuels arrêts de production. Le raccordement des bobines peut être réalisé à une vitesse de fonctionnement de (280 m/min).

Les avantages de l'assemblage automatique sont :

- Une exploitation quasi-totale de la bobine,
- Un déchet minime de papier dans la zone de raccord,
- Le maintien de la vitesse de production constante pendant l'assemblage.
- Un maniement facile et une rapidité de préparation [6].

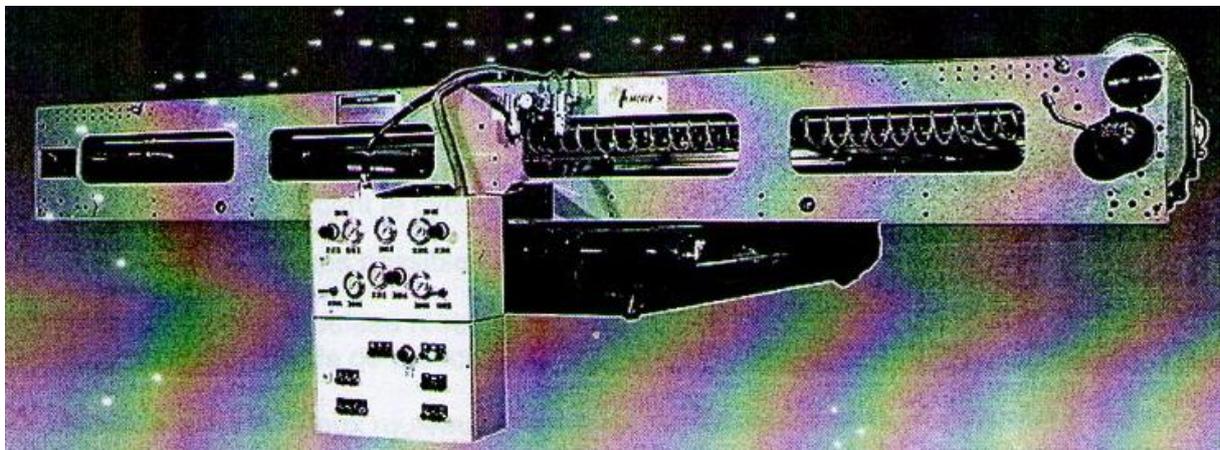


Figure II.2 : Raccordeur CTS[6].

## II.2 Elément de composition de la machine

Afin de faciliter l'identification des différentes parties de la machine, nous allons adopter les définitions suivantes :

Le côté ouvrier est celui où se situent les boutons et pupitres qui commandent la machine et le côté moteur, celui où se situent les différents moteurs de la machine. Et aussi de ce côté se situent les différents panneaux électrique et pneumatique intégrés dans le « raccordeur CTS ». On peut distinguer deux parties de ce dernier : à gauche : *station 1* et à droite : *station 2*.

Pour assembler deux bobines, dont une se trouve en mouvement et l'autre au repos, le système (figure II.4) dispose principalement de :

- Un groupe de réserve (rouleaux en déplacement)
- Unité de raccordement
- Volant d'inertie
- Régulateurs de tension de la bande
- Photocellule de raccordement
- Panneau de contrôle centralisé
- Pupitres de commande latéraux
- Armoires électrique

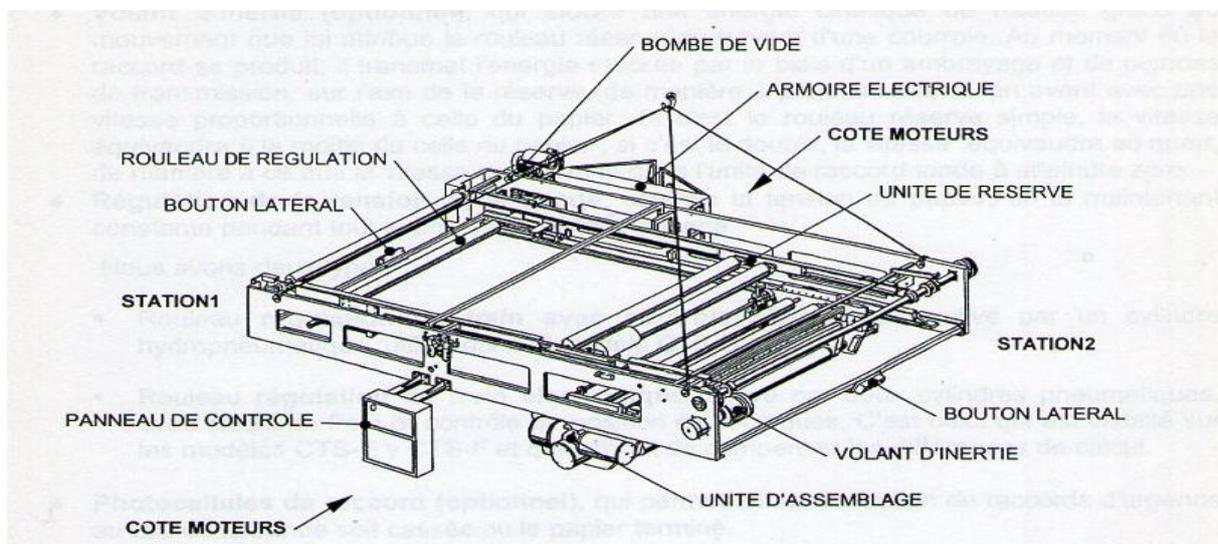


Figure II.3: Unité de raccordement (vue général)[6].

### II.3 Mode de fonctionnement

La tâche de cette machine est de fournir le papier d'une façon permanente au reste du processus, de maintenir la tension du papier constante et d'agir sur l'accélération et la décélération de la ligne.

Le groupe de réserve qui est un ensemble de rouleaux en déplacement horizontal, fournit le papier à la ligne pendant le cycle de raccordement (figure II.5).

Dans une unité de raccordement se produit l'assemblage des bobines, en coupant le vieux papier et en le collant avec un papier neuf, comme on le voit sur la figure suivante ci-dessous:

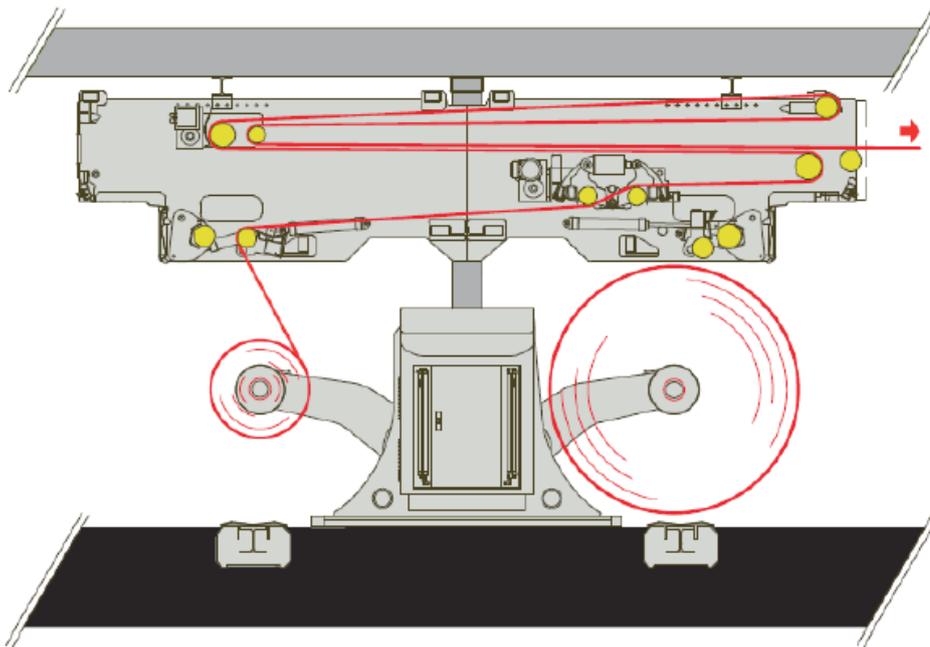


Figure II.4. Raccordeur CTS[6].

L'ensemble est monté sur deux châssis, avec un système de transmission pignon-crémaillère qui le déplace horizontalement pour accéder à la préparation du raccordement dans les deux stations.

Nous désignons le système de coupe et de collage, la partie du raccordeur CTS qui se charge de passer le vieux papier, de le couper et de le coller au nouveau. Puis, le rouleau supérieur et le rouleau en caoutchouc exercent une pression sur la zone de raccordement pour garantir un bon collage (figure II.6).

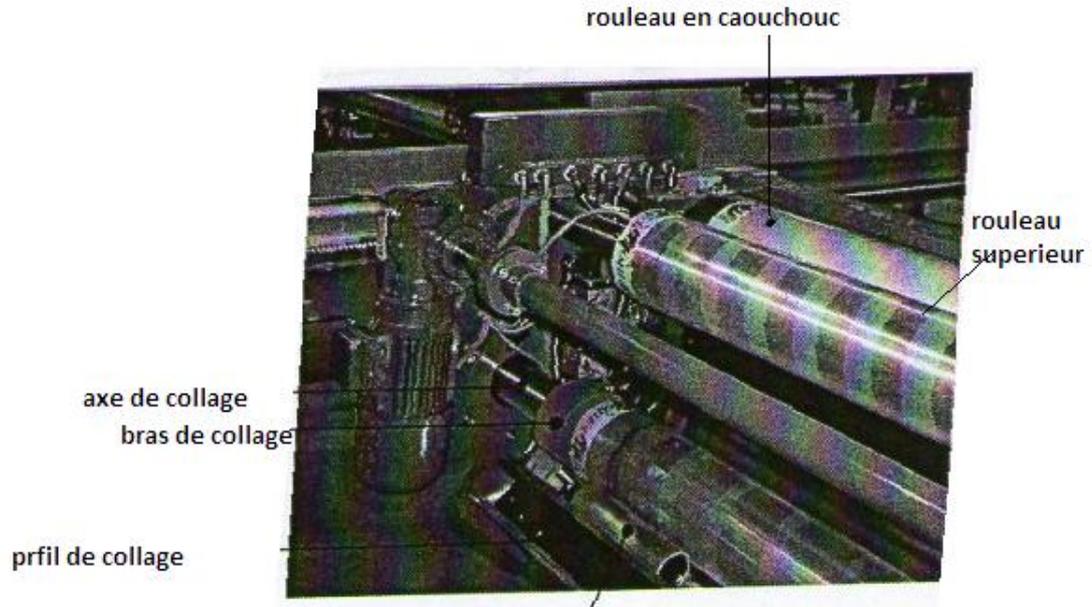


Figure II.5 : Le système coupe-colle[6].

Pour réaliser les fonctions propres au processus de raccordement et de déplacement horizontal, l'unité de raccord est dévisée en plusieurs compartiments, qui sont :

- Système de transmission, pour déplacer l'unité de raccordement de la station 1 et la station 2.
- Ensemble de préparation (un pour chaque station), où se coupe l'extrémité du papier de la bobine à raccorder et où se trouve l'adhésif.
- Ensemble d'impact et de coupe (un pour chaque station), il est chargé de coller les deux papiers sur l'adhésif et de couper le vieux papier.
- Rouleau supérieur et rouleau en caoutchouc qui assurent l'union des bandes collées à l'adhésif.

#### II.4 Séquence de raccordement

Dans un cycle de fonctionnement normal du raccordement on peut résumer cette fonction comme suit :

1. Préparation du papier,
2. Placement de l'adhésif,
3. Position d'attente,
4. Coupe et collage,
5. Changement de station.

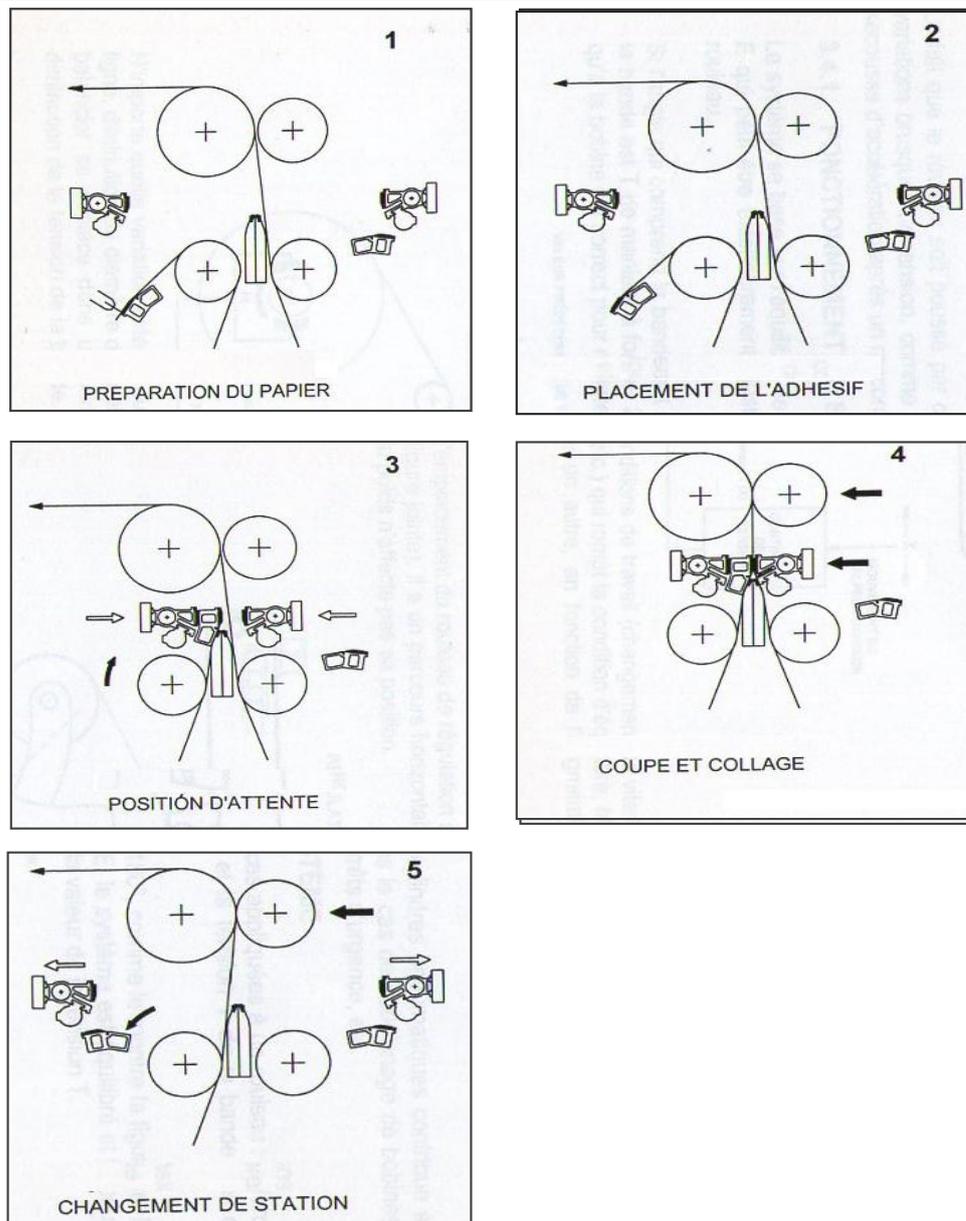


Figure II.6 : Séquences de raccordement [6].

La machine doit fonctionner dans les conditions normales d'une façon permanente sauf s'il y a une panne, alors l'assemblage s'effectue manuellement par l'opérateur [6].

## II.5 Régulation de la tension de système

L'emplacement du rouleau de régulation sur le « raccordeur CTS » vient après l'unité de raccordement (figure II.7). il a un parcours horizontal de quelques 200mm.

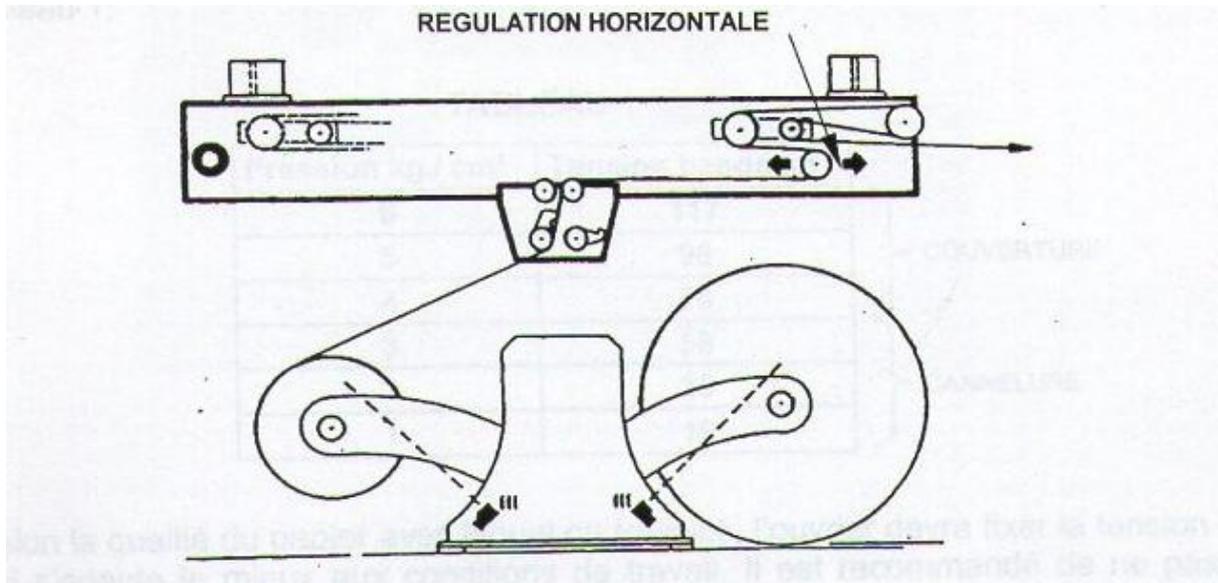


Figure. II.7 : Rouleau de régulation [6].

Le fait que le rouleau soit poussé par des cylindres pneumatiques contribue à amortir les variations brusques de tension, comme dans le cas de débobinage de bobines elliptiques, secousse d'accélération après un raccordement, arrêt d'urgence...etc.

## II.6 Fonctionnement de système de régulation

Le système se base sur l'équilibre des forces appliquées à un rouleau, la force constante  $E$  qui peut être extérieurement ajustée, et la tension  $T$  de la bande qui comprend le rouleau. Si l'angle qui comprend la bande est de  $180^{\circ}$ , comme le montre la (figure 2.8), et la tension de la bande est  $T$  de manière à former  $2T=E$ , le système est équilibré et le couple de freins qu'a la bobine pour maintenir la valeur de la tension

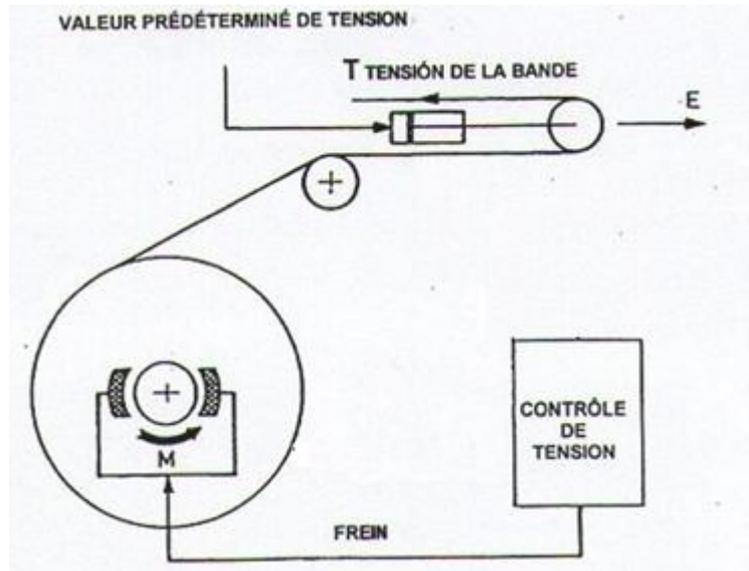


Figure II.8 : Système d'équilibrage des forces [6]

N'importe quelle variation externe aux conditions de travail (changement de vitesse de la ligne, diminution du diamètre de bobine, etc.) qui rompt la condition d'équilibre, fait que la balancier se déplace dans un sens ou un autre, en fonction de l'augmentation ou diminution de la bande.

## II.7 : Positionnement de problème

Le développement et la rénovation des équipements et matériels apportés par les concepteurs, impose aux industries de suivre le rythme de ces améliorations. En effet, *Général Emballage*

Demeure parmi les entreprises les plus modernes, alors se mettre à jour est devenu rituel dans le souci de garder sa place acquise. L'ensemble des machines constituant l'onduleuse sont automatisés avec différents automates comme *omron*, *modicon* ou *siemens*.

Le *Raccordeur CTS* est commandé par un automate de la série S5-95U. Dans le cas de dysfonctionnement de l'automate, l'entreprise doit remédier à ce problème. Pour cela il nous a été proposé de réaliser une migration de S5-95U vers S7-300 afin de surpasser cette problématique et d'apporter des améliorations au fonctionnement du processus.

## **Conclusion**

L'étude détaillée du processus industriel et l'identification des différentes parties de la machine nous a permis la compréhension de ses spécifications fonctionnelles et technologiques, ce qui constitue l'étape initiale et primordiale dans toute étude des systèmes automatisés. L'élaboration du programme est basée sur la connaissance de l'enchaînement des étapes du processus.

Comme nous avons choisi l'API SIEMENS S7-300, nous procédons dans la suite de notre travail à la présentation des API S5-95U et S7-300, et la méthode de programmation sous SIMATIC Manager.

## Chapitre : III

Etude comparative entre l'API S5 et S7

## Introduction

Actuellement la plus part des procédés utilisés au sein de l'entreprise sont automatisées en technologie plus au moins nouvelles, en outre le « Raccordeur CTS». Vu l'importance de la continuité de défilement du papier dans l'onduleuse et la tension du papier qui doit être maintenue à tout moment constante, y compris pendant les accélérations et décélérations de la ligne de production et lors de la séquence de raccordement d'une bobine avec une autre, le « *Raccordeur CTS* » est équipé d'un API SIEMENS S5-95U, et cela pour la flexibilité et l'évolution du processus, mais les avancées technologiques de SIEMENS fait que l'automate S5-95U est moins performant que les nouvelles séries S7200/300/400 ou S1200 et aussi la stratégie de la firme de SIEMENS d'arrêter définitivement de fourniture de la série S5 à partir de 01/10/2015, pour anticiper toute éventuelle insuffisance de produit de S5 (pièces de rechanges.) ,*GENERAL EMBALLAGE* a choisi de faire une migration du S5-95U vers S7-300[7].

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

### III.1 Présentation de l'automate programmable S5-95U [2]

L'API utilisé dans ce processus est S5-95U lancé en 1979, de la firme allemande SIEMENS. La caractéristique principale de cet API c'est qu'il est conçu pour des applications de faible ou de moyenne envergure et pour des applications nécessitant des entres/sorties TOR ainsi que des entres/ sorties analogiques.

Il est adapté à l'exécution des taches de commande structurées, simples, avec des vitesses de réaction élevées, et l'exécution des commandes nécessitant des fonctions supplémentaires comme la régulation.

L'automate programmable industriel S5-95U est utilisé dans notre processus d'une façon fiable, pour accomplir la tâche d'automatisation du « *Raccordeur CTS* »



Figure III.1 : Automate-SIEMENS-S5-95U [2]

### III.1.1 Modules de périphérie de l'API [2]

On peut raccorder 32 modules d'extension directement au S5-95U. Ces modules offrent une série de fonctions étendant ainsi la périphérie intégrée de l'automate. Les modules suivants d'entrée-sortie sont disponibles :

- Modules d'entrée /sorties TOR
- Modules d'entrée /sorties analogique
- Modules de fonction pour des fonctions externes de temporisateur, des compteurs.
- Modules intelligents d'entrée-sortie pour des tâches de régulation et de positionnement et pour des mécanismes de cames électroniques

- Modules programmables intelligents pour des fonctions rapides de prétraitement
- Modules de simulateur pour l'essai de programme.

### III.1.2 Unités fonctionnelles [2]

Comme tous les API le S5-95U contient :

#### ❖ Mémoire ROM

La mémoire ROM contient le système d'exploitation qui ne change pas

#### ❖ Mémoire RAM

Les variables dynamiques sont stockées dans le 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 10 ms à 9990 s avec une résolution de 10 ms.
- Des mémentos sont employés par le programme de STEP5 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

Le périphérique intégré est constituée des éléments suivants :

16 entrées TOR

16 sorties TOR

4 entrées d'alarme

8 entrées analogiques.

1 sortie analogique.

2 entrées de comptage

#### ❖ Interface PG

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

❖ **Cartouche mémoire E(E) PROM**

Les EPROM ou l'EEPROM sont des types de mémoire non-volatile pour la mémoire permanente du programme de gestion.

Après une remise globale du API ou une panne de courant sans support de batterie, le programme de gestion est automatiquement chargé du sous-module de mémoire dans la RAM quand l'API est commutée à mise sous tension.

❖ **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**

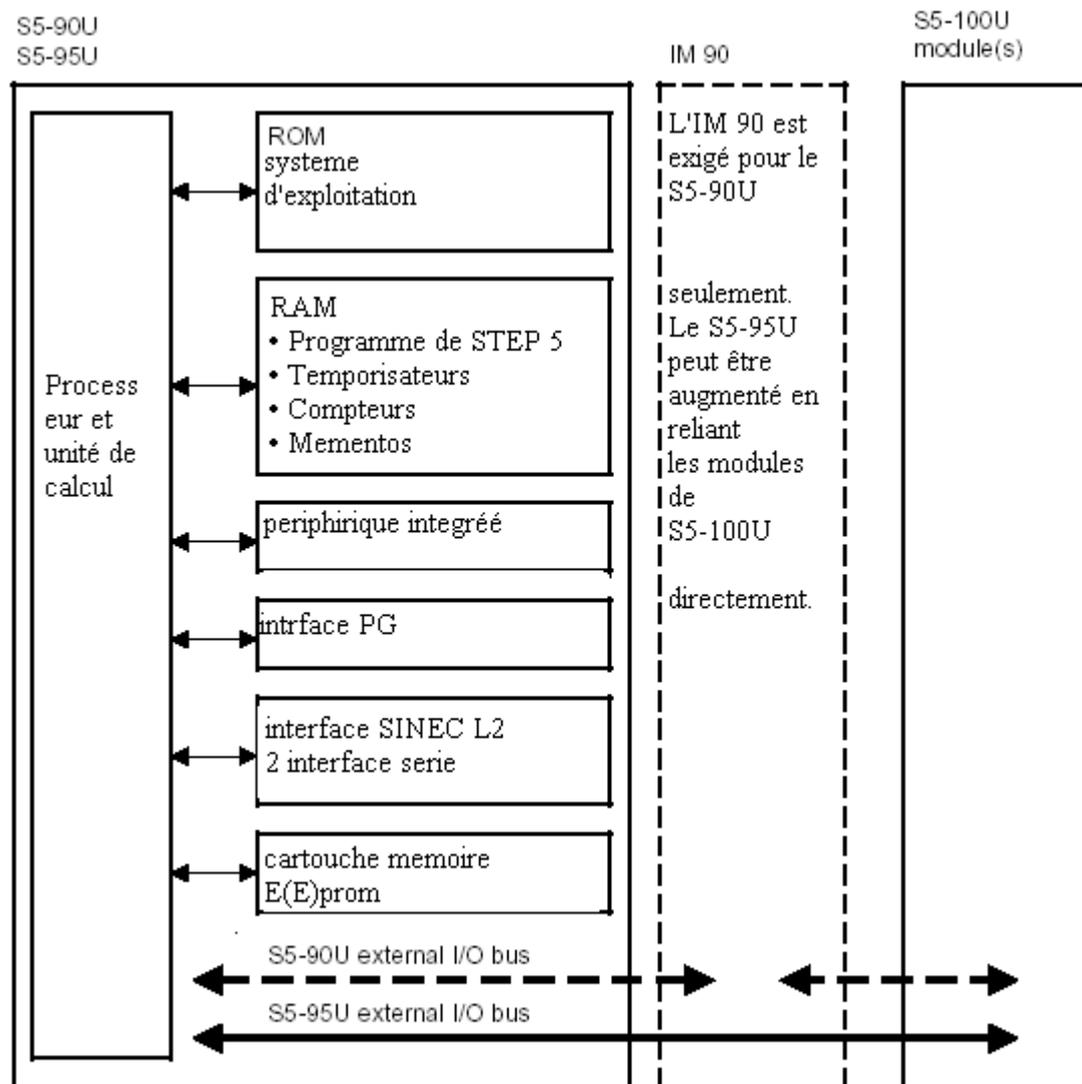


Figure. III.2 : Unités fonctionnelles de L'API S5-95U [2]

### III.2.1 Elaboration d'un programme [2]

Les automates programmables accomplissent des tâches d'automatisation traduites sous forme de programme utilisateur, pour que l'automate puisse le comprendre, ce dernier doit être écrit dans un langage déterminée et suivant des règles bien définie. La firme SIEMENS a développé le STEP5 pour la famille SIMATIC5.

### III.2.2 Les différents modes de représentation [2]

Le langage de programmation STEP5 est unique pour tous les appareils de la gamme SIMATIC5. Le programme peut être introduit sous forme de :

- **Liste d'instruction (LIST):** il est sous forme d'une suite de suite d'abréviations d'instruction.
- **Logigramme (LOG):** le logigramme représente les fonctions logiques à l'aide de symboles graphiques.
- **Schéma à contact (CON):** le schéma à contact CON représente les fonctions de commande à l'aide des symboles graphique des schémas électriques.
- **GRAPH5 :** 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

### III.3.1 Stratégie de la firme SIEMENS pour la série S5 [10]

En 2015 le S5 aura 36 ans, La figure suivante nous montre clairement la stratégie de la firme SIEMENS pour stopper définitivement l'API S5 du marché mondial. Au **01/10/2002** c'est l'annonce du futur arrêt de commercialisation de S5. L'arrêt de fourniture de S5-90/95/100 sera le **01/10/2013** et **01/10/2014** pour le S5-115, le 01/10/2015 pour S5-135/155.

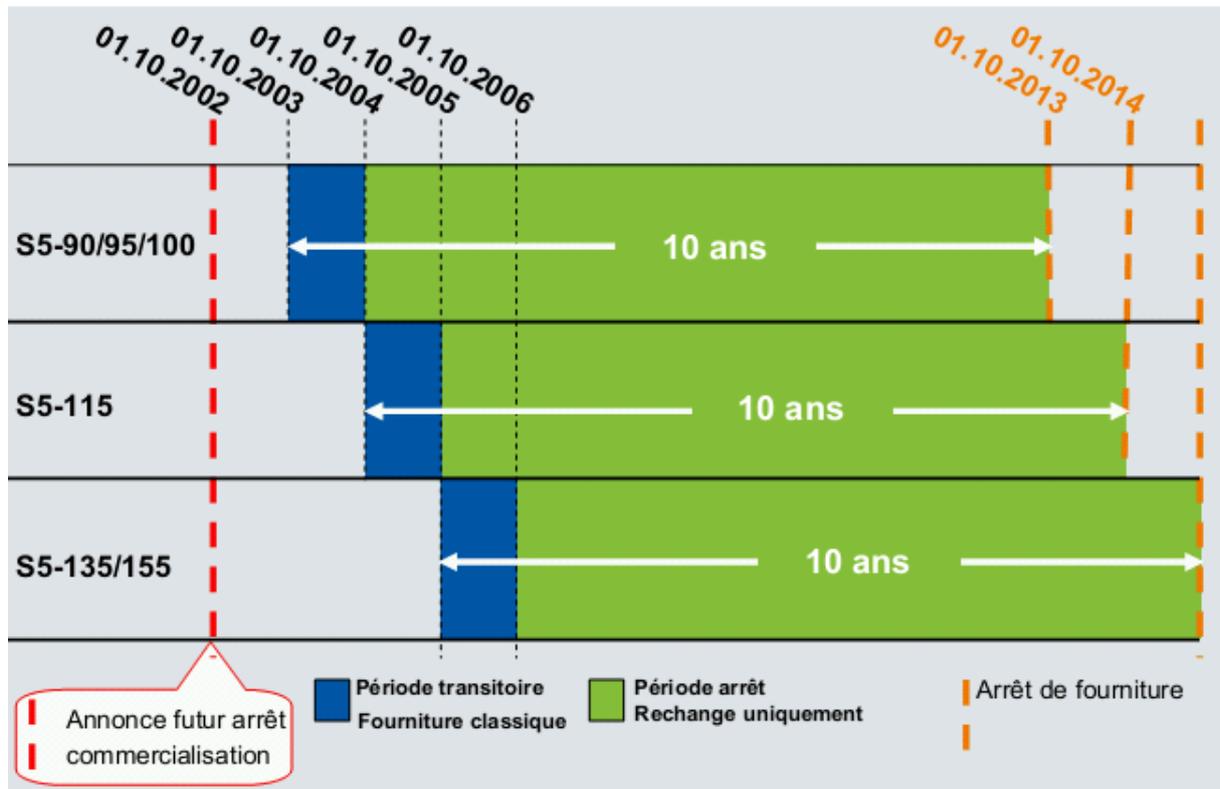


Figure III.3 : Cycle de vie de série S5 [7]

### III.3.2 Solution pour remplacer SIMATIC S5 [7]

Après 36 ans aux services des lignes de production de l'industrie, les automates SIMATIC S5 évoluent pour plus de performances. La nouvelle gamme S7, lancée en 1996, 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 (100U ou 95U), la meilleure approche serait de remplacer le système entier et réaliser une autre programmation en se basant sur le programme existant et un cahier des charge.

### III.4.1 Présentation des automates programmable SIEMENS S7300 [8]

Nous avons opté pour cet API S7300, plus performant et plus puissant, facile à manier et qui a les caractéristiques suivantes :

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

L'API S7-300 est de 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 performance
- des modules de signaux pour entrées et sorties TOR et des modules analogiques
- des modules d'alimentation pour le raccordement du S7-300 sur secteur

120/230 V

- des coupleurs pour configurer un automate sur plusieurs profilés-support

Tous les modules du S7-300 sont montés sous boîtier procurant le degré de protection IP 20.

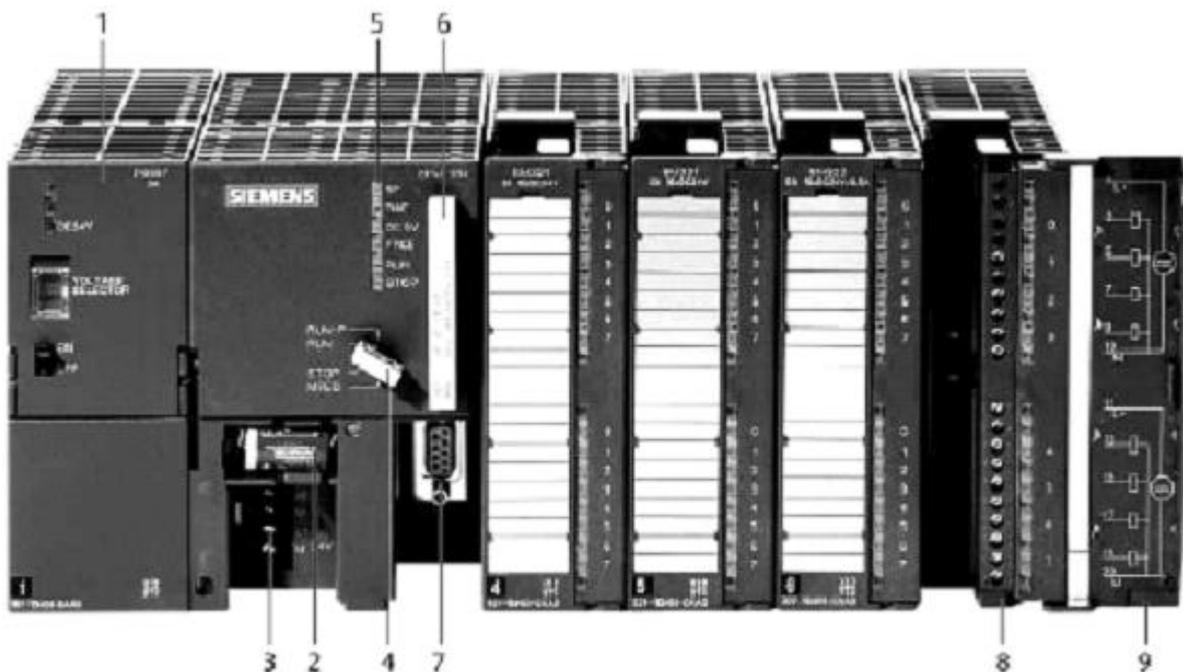


Figure. III.4 : API S7-300[9]

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

### III.4.2 Présentation des différents CPU des automates S7-300[10]

Les API S7300 sont dotés de différents CPU selon l'exigence de l'industrie (Figure III.5)  
On trouve les CPU 312 IFM, 313, 314, 314 IFM, 315, 315-2 DP et 316, 312ptp, 313-C2DP...etc.

➤ **CPUS7- 312ptp :**

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/sorties TOR intégrées pour de petites applications à exigences élevées concernant la puissance de traitement

➤ **CPU S7-313-C2DP**

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

- Pourvue de fonctions technologiques et des tâches à fonctions spéciales
- Pour le raccordement de périphérie décentralisée

➤ **CPU S7-313C**

Est une CPU compacte à entrées/sorties TOR et analogiques intégrées conçue :

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



CPUS7- 312ptp



CPU S7-312



CPU S7-313-C2DP



CPU S7-313C

Figure. III.5 : CPU de différents niveaux de performance de l'automate S7-300[9]

### III.4.3 Utilisation de S7-300

Pour pouvoir travailler avec cet automate et exécuter le programme, nous avons besoin du matériel suivants :

Constituants	Fonction
Profilé support	Il constitue le châssis du S7-300.
Module d'alimentation (PS)	Il convertit la tension de secteur (120/230 V CA) en une tension continue de 24 V pour l'alimentation du S7-300.
Module de simulation (6ES7 374...) avec 8 entrées TOR et 8 sorties TOR	Il nous donne la possibilité de tester le programme utilisateur à la mise en service de l'automate et en cours de fonctionnement : des interrupteurs simulent les signaux des capteurs ; des voyants (LED) visualisent l'état logique des signaux des sorties.
Câble MPI	Il relie la PG ou le PC à la CPU.
Console de programmation (PG) avec carte MPI et logiciel STEP 7 installé ou bien Ordinateur personnel (PC) avec carte MPI ou câble PC/MPI et logiciel STEP 7 installé	Elle sert à configurer, à paramétrer, à programmer et à tester l'automate S7-300.

Table III.1 : Tableau explicatif d'utilisation de S7 300[8]

### **III.4.4 Elaboration d'un programme**

Les automates programmables accomplissent des tâches d'automatisation traduites sous forme de programme utilisateur, pour que l'automate puisse le comprendre, ce dernier doit être écrit dans un langage déterminée et suivant des règles bien définie. La firme SIEMENS a développé le STEP7 pour la famille SIMATIC7

### **III.4.5 Utilisation de STEP7 [10]**

Le STEP 7 est le progiciel de base pour la configuration et la programmation des systèmes d'automatisation SIMATIC.

Il fait partie de l'industrie logicielle SIMATIC. Le progiciel de base STEP 7 existe en plusieurs versions :

- STEP 7-Micro/DOS et STEP 7-Micro/Win pour des applications autonomes simples sur SIMATIC S7 - 200.
- STEP 7 pour des applications sur SIMATIC S7-300/400, SIMATIC M7-300/400 et SIMATIC C7 présentant des fonctionnalités supplémentaires :
  - Possibilité d'extension grâce aux applications proposées par l'industrie logicielle
  - Possibilité de paramétrage de modules fonctionnels et de modules de communication
  - Forçage et fonctionnement multiprocesseur
  - Transfert de données commandé par événement à l'aide de blocs de communication et de blocs fonctionnels

### **III.4.5 Fonctions du logiciel STEP 7 [10]**

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 projets,
- la configuration et le paramétrage du matériel et de la communication,
- le chargement de programmes dans des systèmes cible,
- la gestion des mnémoniques,
- la création de programmes, par exemple pour les systèmes cible S7,
- le test de l'installation d'automatisation,

- le diagnostic lors de perturbations de l'installation.

### III.5 Interaction du logiciel et du matériel [5]

L'automate programmable (AP) contrôle et commande à l'aide du programme S7 notre machine. L'adressage des modules d'E/S se fait par l'intermédiaire des adresses du programme S7.

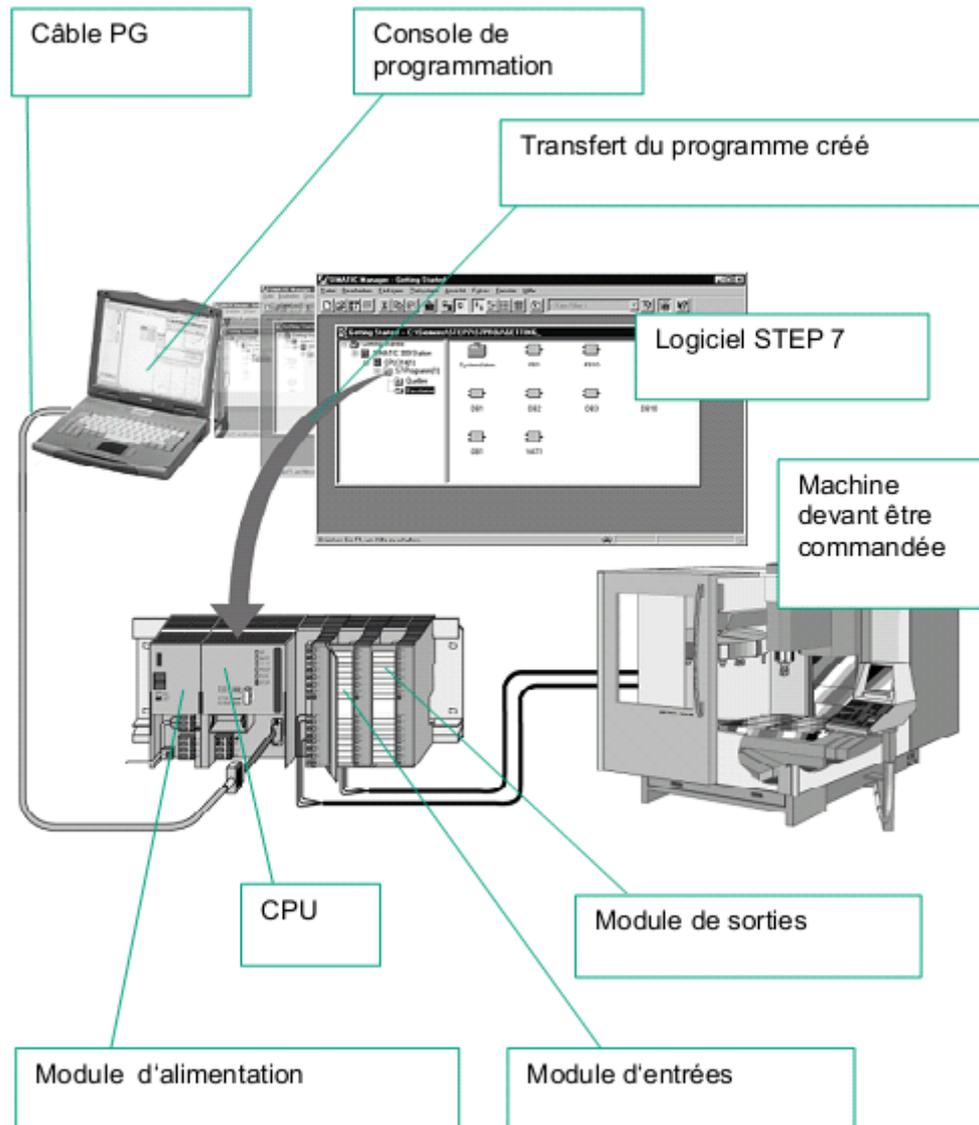


Figure III.6 : Interaction du logiciel et du matériel [5]

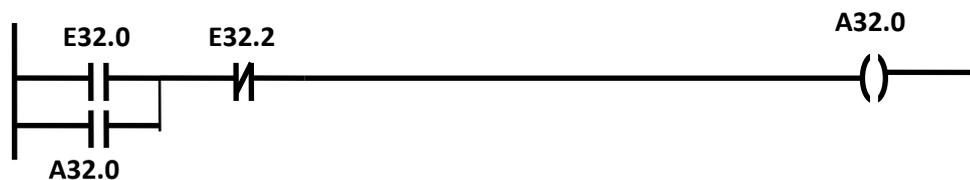
### III.5.1 Création d'un projet [9]

La programmation STEP 7 est une programmation structurée dans des blocs qui sont les blocs d'organisation, les fonctions, les blocs fonctionnels, les blocs de données. L'écriture des programmes est possible sous plusieurs langages qui sont :

- **le langage à contact (CONT) :**

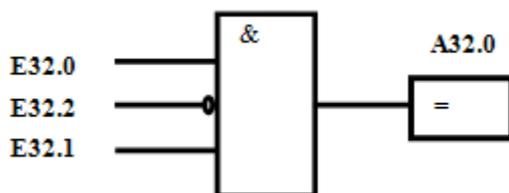
Dans le langage de programmation graphique CONT, la représentation est fondée sur des schémas à relais. Les éléments d'un tel schéma, comme par exemple les contacts à ouverture ou les contacts à fermeture sont reliés pour former des réseaux. Un ou plusieurs de ces réseaux forment la section d'instructions complète d'un bloc de code.

Le langage de programmation CONT fait partie du logiciel de base STEP 7.



- **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 ces pavés logiques.



- **le langage en liste d'instructions (LIST) :**

Le langage de programmation LIST (liste de d'instructions) est un langage textuel proche du langage machine. Chaque instruction correspond à une étape de l'exécution du programme

U (

O " Marche" // Bouton-poussoir "Marche"

O " Bobine" // Bobine de maintien

),

### **Remarque [10]**

Tout comme la gamme des SIMATIC S5 qu'elle vient compléter la gamme SIMATIC S7 mise sur la modularité bien éprouvée. Nous trouvons les types de modules suivants :

- 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).

### **Des nouvelles performances [10]**

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.
- Nous avons un choix de modules paramétrables et dotés de fonctions de diagnostic

### **III.5.2 Paramétrage des modules S5/S7 [10]**

Le paramétrage des modules SIMATIC S5 et SIMATIC S7 est expliqué dans le tableau Suivant.

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 Nous 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 zones de données système par exemple DB 1/DX 0	La 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

Tableau III.2 : Paramétrage des modules S5/S7 [10]

**III.6 Mise en parallèle des blocs STEP 5 et blocs STEP 7**

<b>Bloc STEP 5</b>	<b>Bloc STEP 7</b>	<b>Explication</b>
Bloc d'organisation (OB)	Bloc d'organisation (OB)	Interface au système d'exploitation
OB spéciaux intégrés	Fonctions système (SFC) Blocs fonctionnels système (SFB)	Les fonctions système 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 fonctions STEP 7 (FC).
Bloc séquentiel (SB)	-----	Les blocs séquentiels n'existent plus dans STEP 7
Bloc de données (DB, DX)	Bloc de données (DB)	Les blocs de données sont plus longs dans STEP 7 que leurs équivalents dans STEP 5 (jusqu'à 8 kilo-octets pour le S7-300, et jusqu'à 64 kilo-octets pour le S7-400)
Blocs de données DX0,	Blocs de données	Les nouveaux blocs de

DB1 dans leur fonction spéciale	système (SDB) (paramétrage CPU)	données système renferment toutes les données de configuration matérielle ainsi que les paramétrages de CPU nécessaires à l'exécution du programme.
Blocs de commentaire DK, DKX, FK, FKX, PK		Les blocs de commentaire n'existent plus dans STEP 7. Le commentaire est contenu dans les blocs correspondants dans la base de données hors ligne.

Tableau III.3 : Mise en parallèle des blocs STEP 5 et blocs STEP 7[10]

### Conclusion

Dans ce chapitre on a présenté le SIMTIC S5 et le SIMATIC S7, On conclut que le S7 n'est qu'une amélioration de S5 existant. Avec quelques changements de certaines fonctions et préservation de certaines d'autres. Pour passer de S5 vers S7 on utilise le convertisseur intégré dans le STEP 7 mais, il y a des fonctions non convertibles pour lesquelles une intervention manuelle est nécessaire.

**Chapitre : IV**  
**Conversion de S5 vers S7**

## Introduction

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

Des fonctions résolues dans SIMATIC S5 ont dû être repensées pour céder la place à une approche entièrement nouvelle dans SIMATIC S7.

Le logiciel de programmation STEP 7 mise sur les technologies et les idées nouvelles.

Ainsi, la surface utilisateur exécutable sous Windows 95 ou Windows XP a été développée en tenant compte des connaissances ergonomiques les plus modernes. Nous avons tenu à respecter dans nos langages de programmation la norme CEI 1131 tout en restant compatible avec STEP 5 [10]

Dans ce chapitre nous avons opté pour deux approches, la première est de faire la conversion avec un convertisseur S5/S7 et la deuxième est consacré à la conception de l'automatisation du '*Raccordeur CTS*'

### IV.1 Première approche

Conversion de S5 vers S7 en utilisant le convertisseur S5/S7 :

La programmation S7 en LIST, CONT et LOG est compatible avec, respectivement, LIST, CONT et LOG de S5. Par conséquent, si nous somme des utilisateurs de S5 et que nous souhaitons mettre en œuvre dans S7 des programmes existants, la conversion sera très facile. Nous pouvons rester fidèles à nos programmes en S5 il suffit de les convertir en programmes S7 [10].

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'instructions des programmes S5 existants en programmes S7. La conversion des programmes S5 existants se fait toujours en LIST [10].

### IV.1.1 Procédure complète de conversion

Le convertisseur S5/S7 se présente avec l'image-écran ci-après [10].

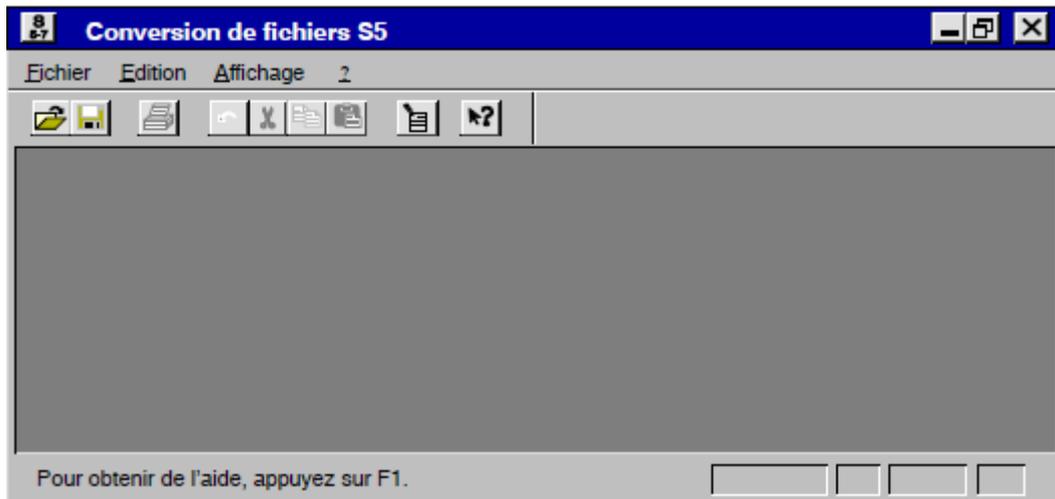


Figure IV.1 : Image-écran initiale du convertisseur S5/S7 [10].

La procédure complète de conversion comprend quatre étapes

#### ➤ Préparation des fichiers S5 à convertir

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

- <Nom>ST.S5D  
Fichier programme (le programme S5 à convertir)
- <Nom>XR.INI  
Liste des références croisées (contient la structure du programme)

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

- <Nom>Z0.SEQ  
Liste d'assignation

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

On sélectionne le fichier à convertir et cliquez sur OK pour confirmer votre choix.

## Résultat

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 (Fig. IV.2) suivante présente la boîte de dialogue "Conversion de fichiers S5 -- [<Nom>ST.S5D)".

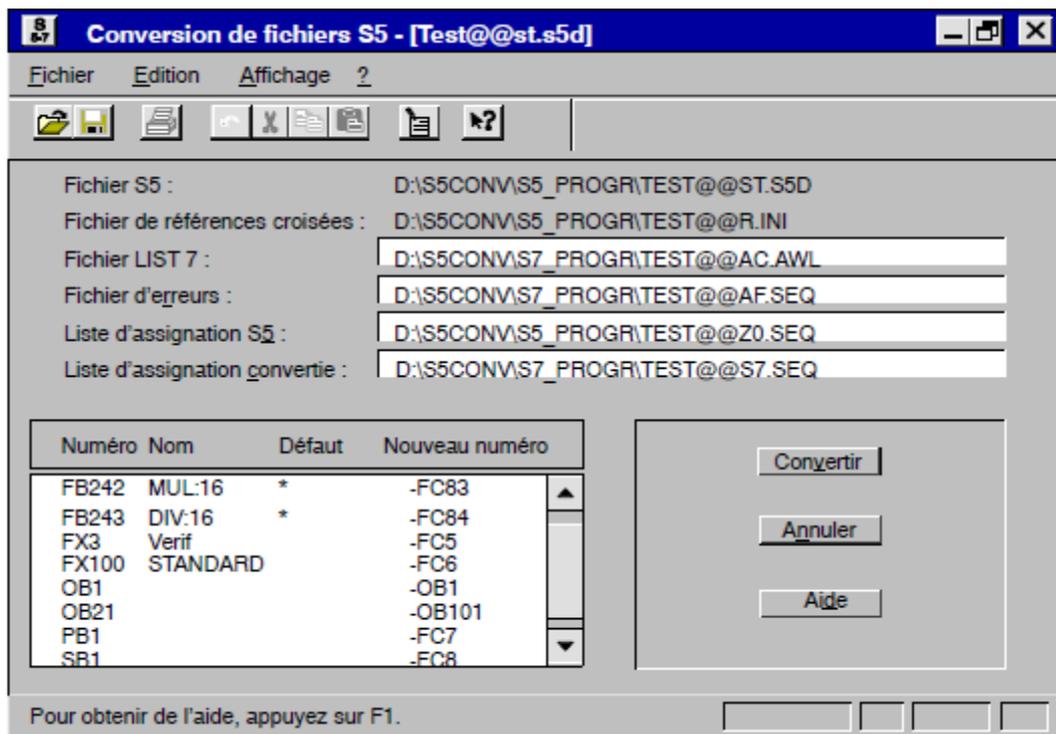


Figure. IV.2 : Boîte de dialogue ' Conversion de fichiers S5' [10]

### IV.1.2 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.

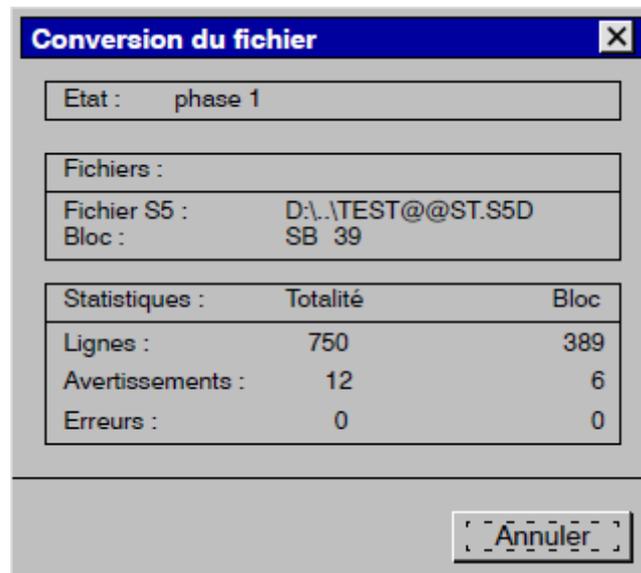


Figure IV.3 : Première phase de la conversion [10]

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

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

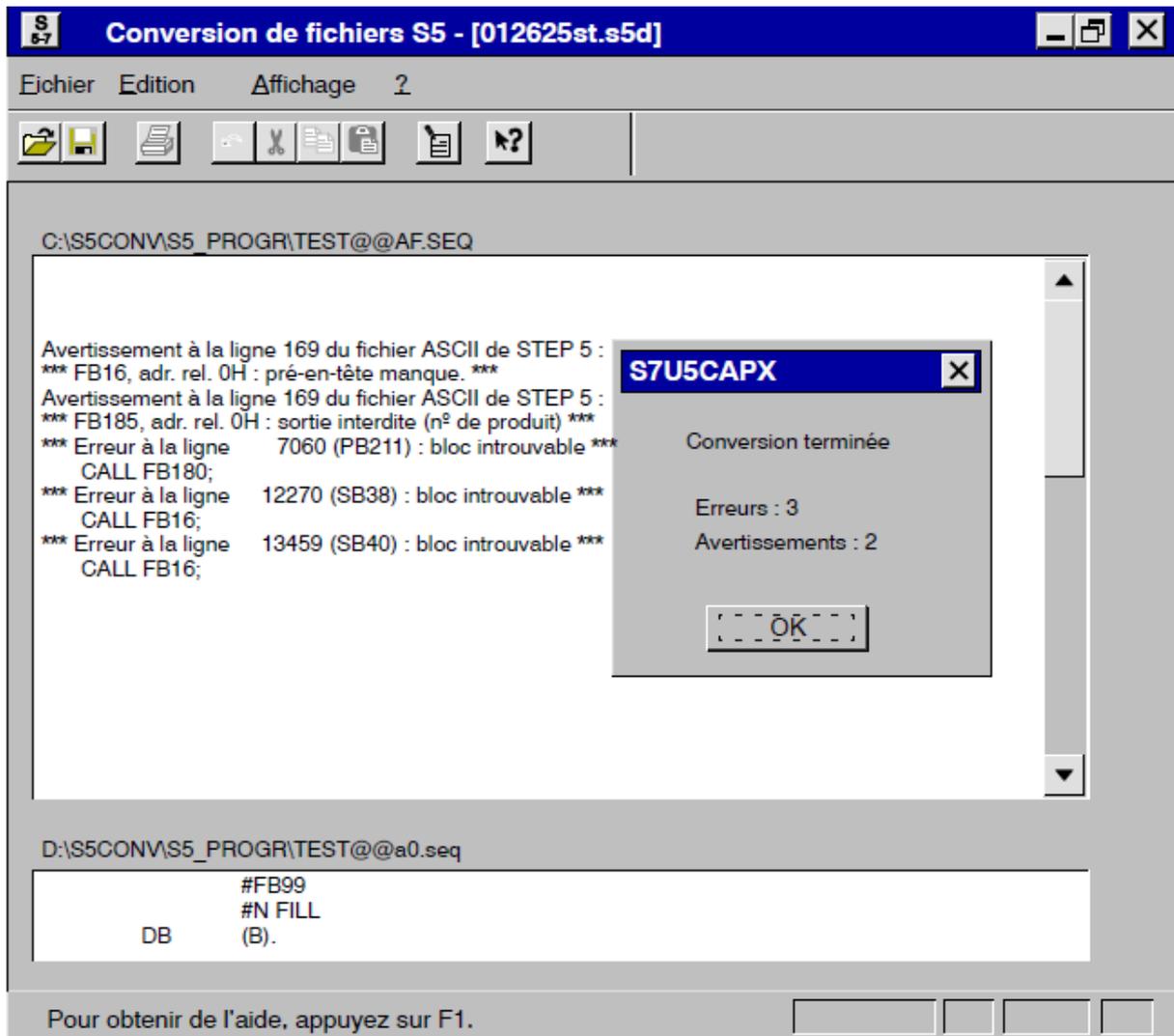


Figure IV.4 : Messages affichés par le convertisseur [10]

### IV.1.3 Localisation des erreurs

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 (par suite de modifications de la sémantique des opérations, par exemple).

### IV.1.4 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

La figure suivante est générée automatiquement lors de la conversion :

(Tab.IV.1) suivant dresse la liste de tous les messages d'erreur avec leur signification et les mesures à prendre pour remédier à l'erreur.

Message d'erreur	Origine	Signification	Remède
Paramètre absolu diverge de l'identificateur d'opérande	phase 1	L'identificateur d'opérande n'est pas correct.	Vérification de 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	phase 2	Le programme S5 contient des accès	Vérifiez le programme LIST.

possible. Veuillez vérifier		par bit à des temporisations et à des compteurs.	
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.
CALL SFCxy a été généré, veuillez compléter la liste des paramètres	phase 2	Des paramètres SFC manquent	On complète la liste des paramètres SFC
Opérande incorrect	phase 1	L'opérande ne convient pas à l'opération.	Vérification de la source S5.
	phase 2	L'opérande ne convient pas à l'opération.	Modification du fichier LIST.
Erreur de conversion	phase 2	Opération BI sans constante	On complète l'opération de chargement par une constante.
Erreur dans le fichier macro, xy non pris en compte	phase 2	Erreur de macro-instruction	Vérification de la macro-instruction.
Paramètre formel non défini	phase 1	Il y a plus de paramètres que dans le bloc appelant.	Vérification du fichier programme S5.
Fichier ou répertoire introuvable	phase 1	Le fichier programme ne contient aucun bloc.	Vérification du fichier programme.

Tableau IV.1 : Messages d'erreur, signification et remède [11]

#### IV.1.5 Analyse et interprétation des messages

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

On procède comme suit pour analyser ces messages

- ✓ La visualisation dans la zone inférieure de la fenêtre "Messages" le fichier dans lequel l'erreur s'est produite.
- ✓ Correction de l'erreur comme il est proposé.

#### IV.1.6 Retouche de programme

À la fin de la conversion, une fenêtre de messages séparée affiche la localisation des erreurs de conversion dans le programme. Nous devons alors prendre les mesures nécessaires pour remédier à ces erreurs.

#### IV.1.7 Compilation

Nous devons compiler le programme converti et éventuellement retouché avec le compilateur PLCSIM afin de le rendre exécutable. Procédons pour ce faire exactement comme pour un fichier de texte que nous venons de créer.

#### IV.1.8 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.

#### IV.1.9 Correction 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 précisée. Lorsque nous sélectionnons un message d'erreur, l'emplacement correspondant du fichier source s'affiche dans la fenêtre supérieure. Cela vous 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.

**Remarque 1**

Dans S5, les blocs d'organisation n'ont pas les mêmes fonctions que dans S7. Lors de la retouche du programme converti, nous devons remplacer les OB qui ne sont pas convertis automatiquement par [10]

- ✓ des blocs d'organisation avec d'autres fonctions,
- ✓ de nouvelles opérations S7
- ✓ ou des paramètres système que nous définissons lors du paramétrage du matériel.

**Remarque 2**

Un exemple de conversion (voir annexe 2)

**IV.2 La deuxième approche**

Conception d'un automatisme en utilisant le cahier de charges

**IV.2.1 Cahier des charges**

Le processus est composé de trois parties

- a) Pendant le cycle de débobinage des bandes, on doit les préparer sur les bras de dérouloir

En appuyant sur le bouton poussoir 'S02' si 'S04' est activé, le chariot se déplace jusqu'à la *position 1* et le bras gauche du *poste 1* peut être levés 'K2' pour introduire une bobine neuve.

On appuyant sur le bouton poussoir 'S03', le chariot se déplace jusqu'à la *position 2*, le bras droit du *poste 2* peut être levés 'K1' pour introduire une bobine neuve.

- b) Le chariot se déplace si les deux bras ne sont pas levés, et si l'un des bras est levé, le chariot ne peut pas s'avancer dans la direction de ce bras.
- c) Le processus de raccordement, c'est-à-dire, la coupe du vieux papier et le collage du nouveau, se réalise comme suit :

En appuyant sur 'BP', La bonde du papier qu'on veut remplacer est arrêtée pendant approximativement une seconde.

Après avoir placé l'adhésif sur le papier neuf (manuellement), et le préparer on réalise le raccordement en appuyant sur le bouton *BP2* (Figure II.6)

Au moment où se produit le raccordement, à l' instant même où s'annule le blocage des rouleaux de réserve qui garantissent le défilement du papier vers l'onduleuse afin de gardé la vitesse de ligne constante pendant l'arrêt de la bobine.

La tension du papier doit être maintenue constante, et cela est garanti par un système de régulation (Figure : II.8).

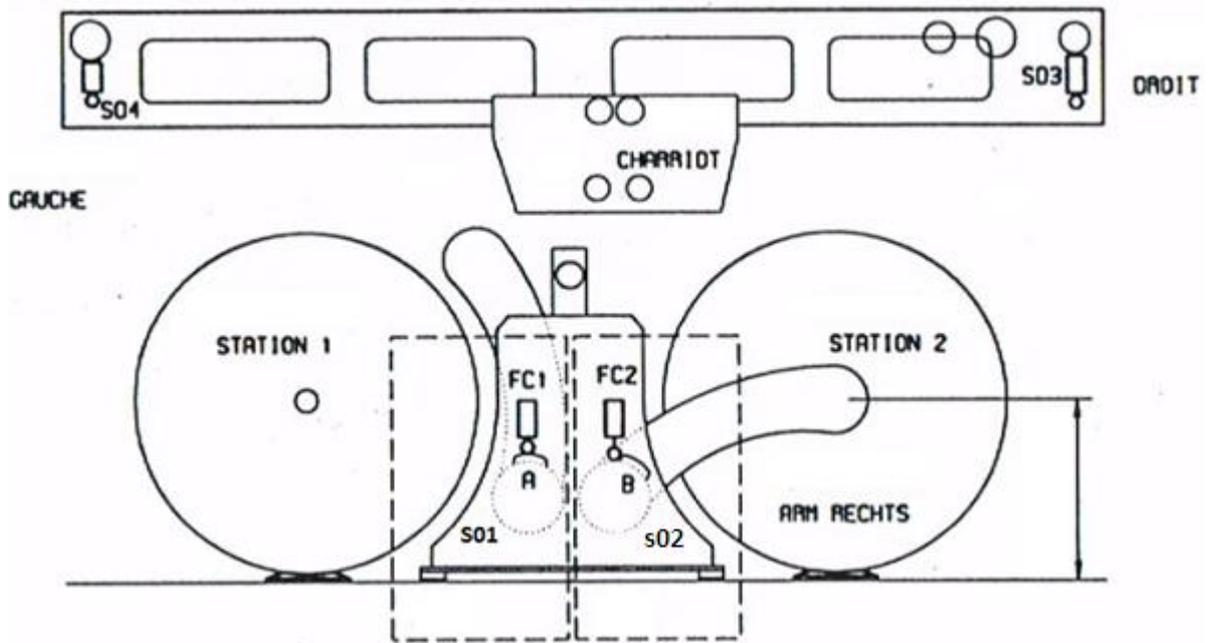


Figure. IV.5 : chariot du Raccordeur CTS [6]

#### IV.2.2 Tableau récapitulatif des entrées/soties

Nom de la variable	définition
S04	interrupteur de fin de course gauche
S03	interrupteur de fin de course droite
S01	bouton poussoir 'levage de bras de st1'
S02	bouton poussoir 'levage de bras de st2'
K1	contacteur 'levage bras droit'
K2	contacteur 'levage bras gauche'
BL1	levage de bras 1

BL2	levage de bras 2
attente	introduire la bobine la station1
BOB2INT	introduire la bobine la station2
BP1	arrêt de rotation de bobine 'freinage'
BP2	coupe colle
T1	temporisateur 1 seconde
LF	libération de frein
ROT BOB	rotation de la bobine normalement
STOP BOB	Arrêt de rotation
BP0	arrêt d'urgence
Q	capteur gauche de présence de papier
L	capteur droit de présence de papier
F	freinage
LED	Alarme de default
resrv	Rouleaux de réserve fonctionne
CP	Coupé -collé
C	Capteur de seuil
START	Mise en marche

Tableau IV.2 : Les différents entrées et sorties du '*Raccordeur CTS*'

### IV.2.3 Grafcet principale de la machine

Nous avons élaboré le **GRAFCET** principale du fonctionnement automatique de '*Raccordeur CTS*' des bobines à partir du cahier des charges.

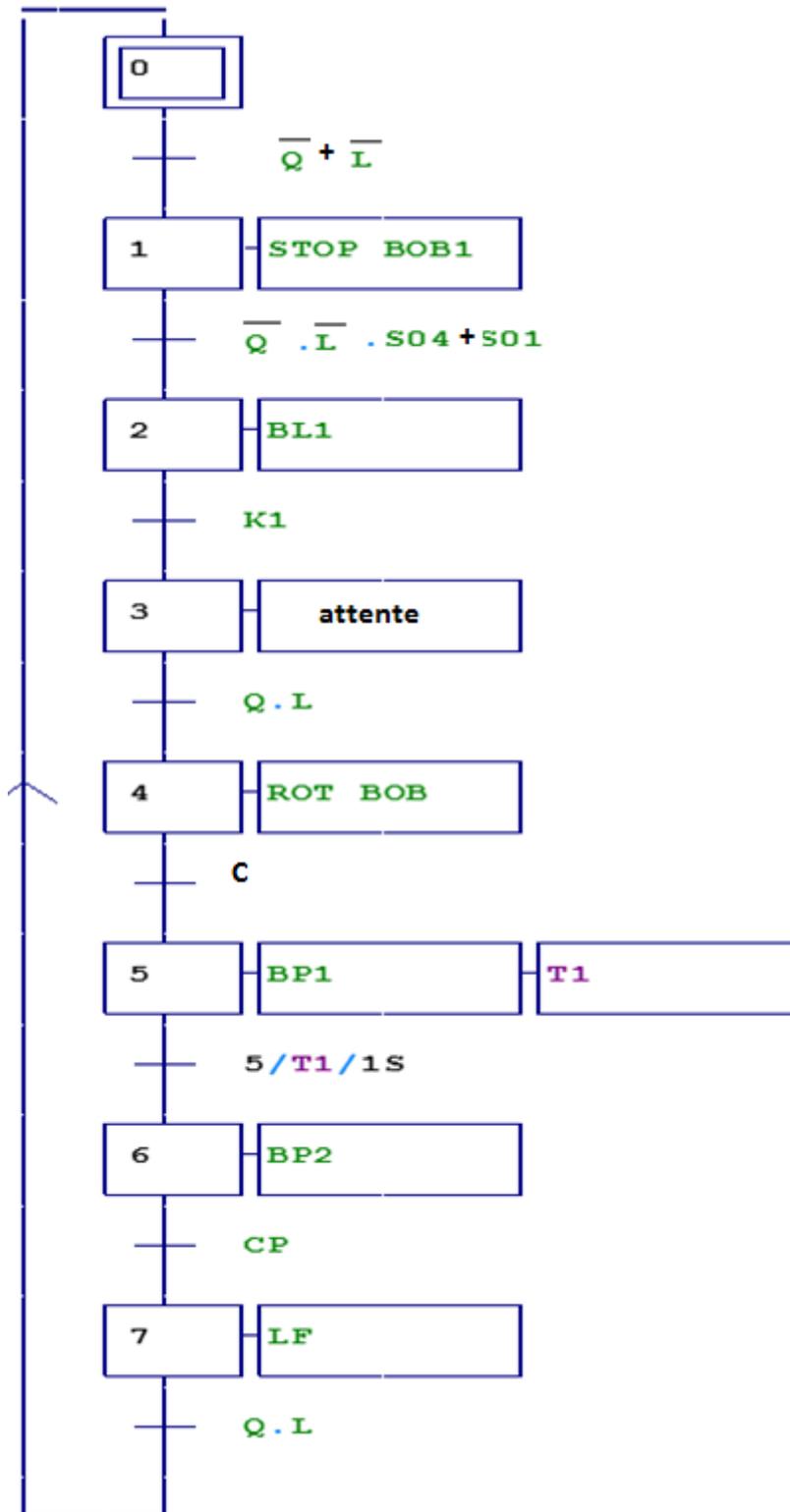


Figure IV.6 : Grafcet de la station gauche

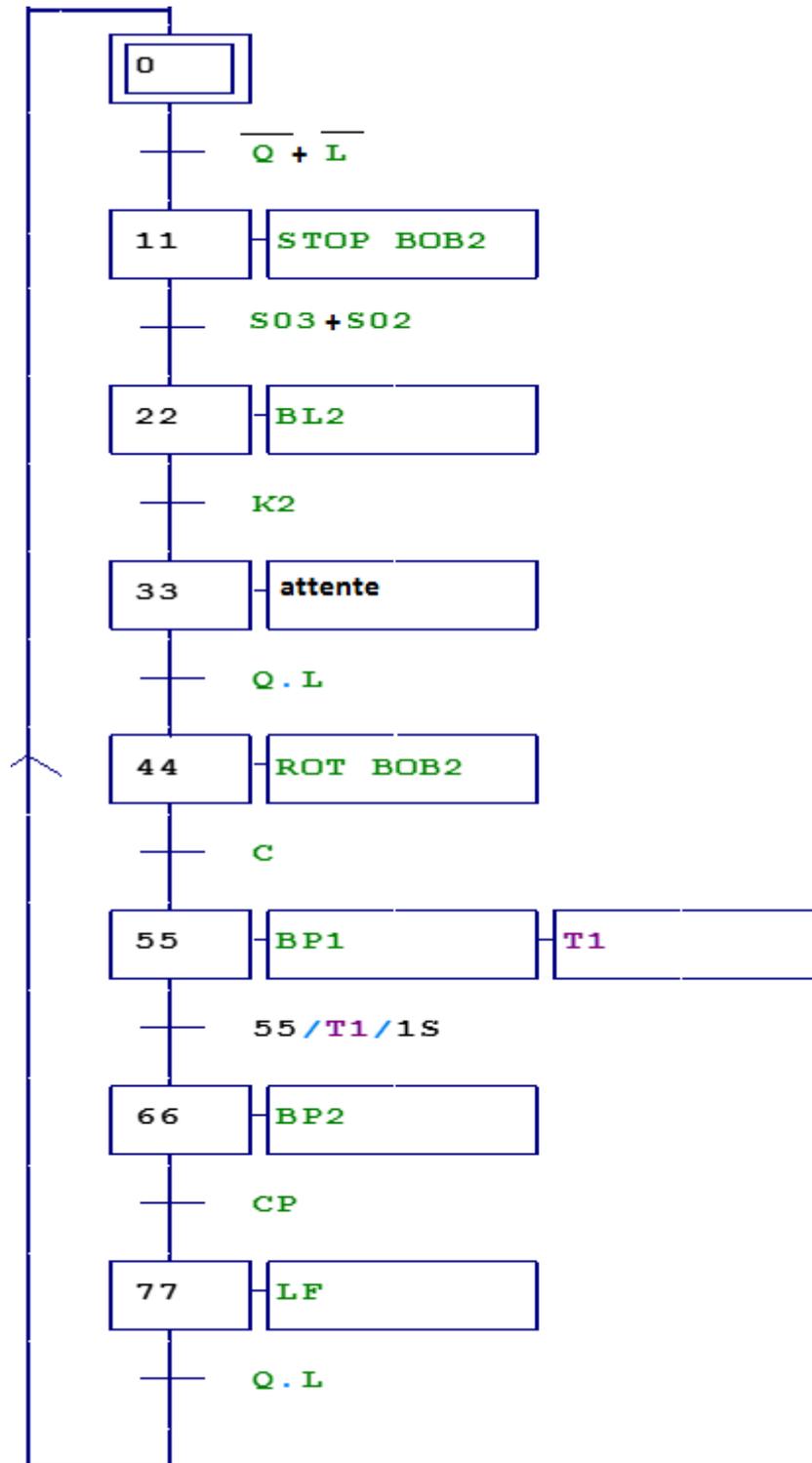


Figure IV.7 : Grafcet de la station droite

#### IV.2.4 Grafcet de sécurité

L'action sur le bouton d'arrêt d'urgence permet de figer tous les autres Grafcets. Une fois le contacteur BP0 est fermée un voyant rouge(LED) s'allume et les GRAFCETs sont remis à l'état initial, et ils restent dans cet état tant que le bouton d'arrêt d'urgence n'est pas désactivé.

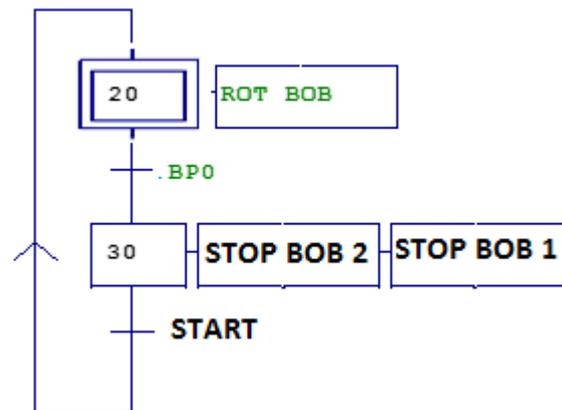


Figure.IV.8 : Grafcet de sécurité

#### IV.2.5 Programmation et simulation

Pour concevoir le programme à l'aide du logiciel STEP 7 nous avons utilisé le langage à Contact (LADDER).

##### ➤ Démarrage du STEP 7

Pour le démarrage de STEP 7, on effectue un double clic sur l'icône « SIMATIC Manager ».

De là, on peut accéder à toutes les fonctions qu'on a installées aussi bien du logiciel de base que des logiciels optionnels.

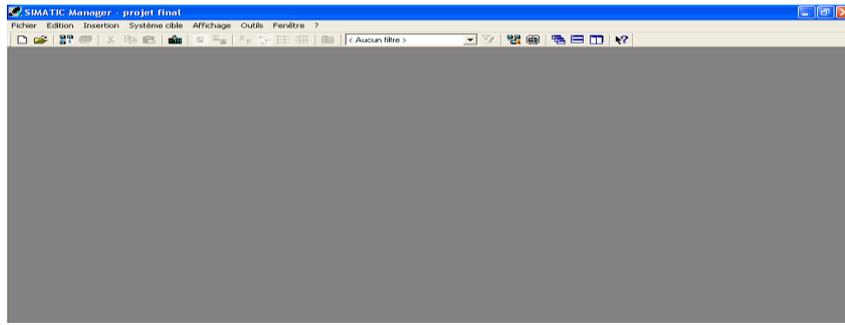


Figure. IV.9 : Fenêtre SIMATIC Manager [11].

### ➤ Création d'un nouveau projet

Afin de créer un nouveau projet STEP 7, il nous est possible d'utiliser « l'assistant de création de projet », ou bien créer le projet soi-même et le configurer directement. Cette dernière méthode est un peu plus complexe, mais nous permet aisément de gérer notre projet. En sélectionnant l'icône SIMATIC Manager, on affiche la fenêtre principale, pour sélectionner un nouveau projet et le valider

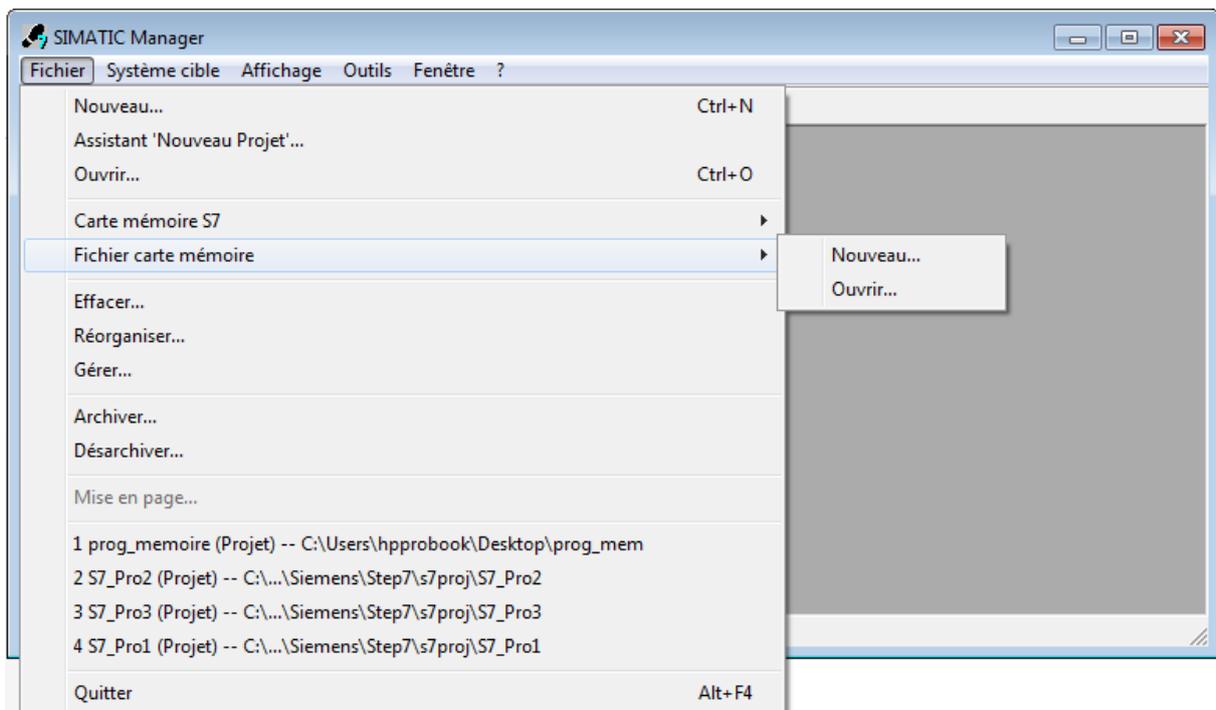


Figure IV.10 : Fenêtre nouveau projet.

### ➤ Création de la table des mnémoniques

Dans tout programme, il faut définir la liste des variables qui vont être utilisées lors de la programmation. Pour cela la table des mnémoniques est créée. L'utilisation des noms appropriés rend le programme plus compréhensible et plus facile à manipuler. Ce type d'adressage est appelé « relatif ».

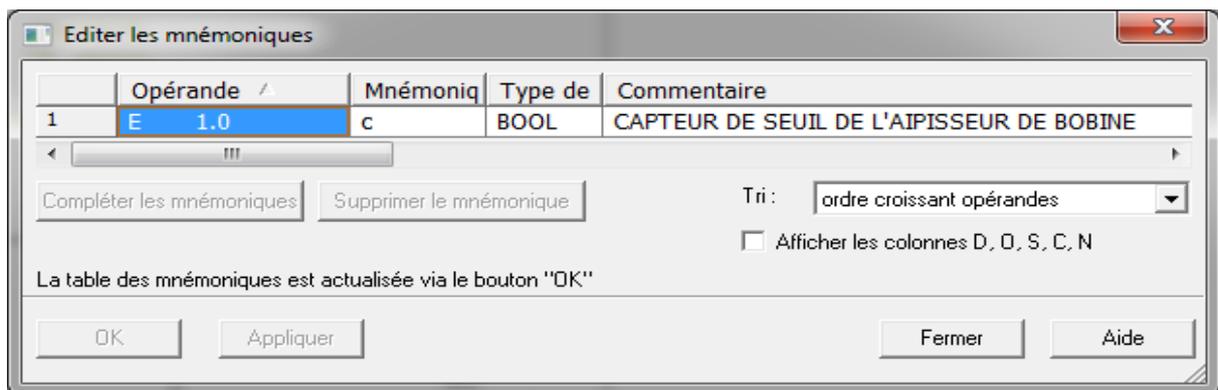


Figure. IV.11 : Fenêtre pour éditer les mnémoniques.

### ➤ Création d'un programme dans le bloc d'organisation « OB »

Le bloc OB1 est généré automatiquement lors de la création d'un projet. C'est le programme cyclique appelé par le système d'exploitation. Ensuite on clique sur l'item « OB1 » et la fenêtre de l'éditeur de programme apparaît.

#### ➤ Editeur de programme et langages de programmation

Avant de commencer le programme, il faut sélectionner le langage de programmation par la commande « affichage »

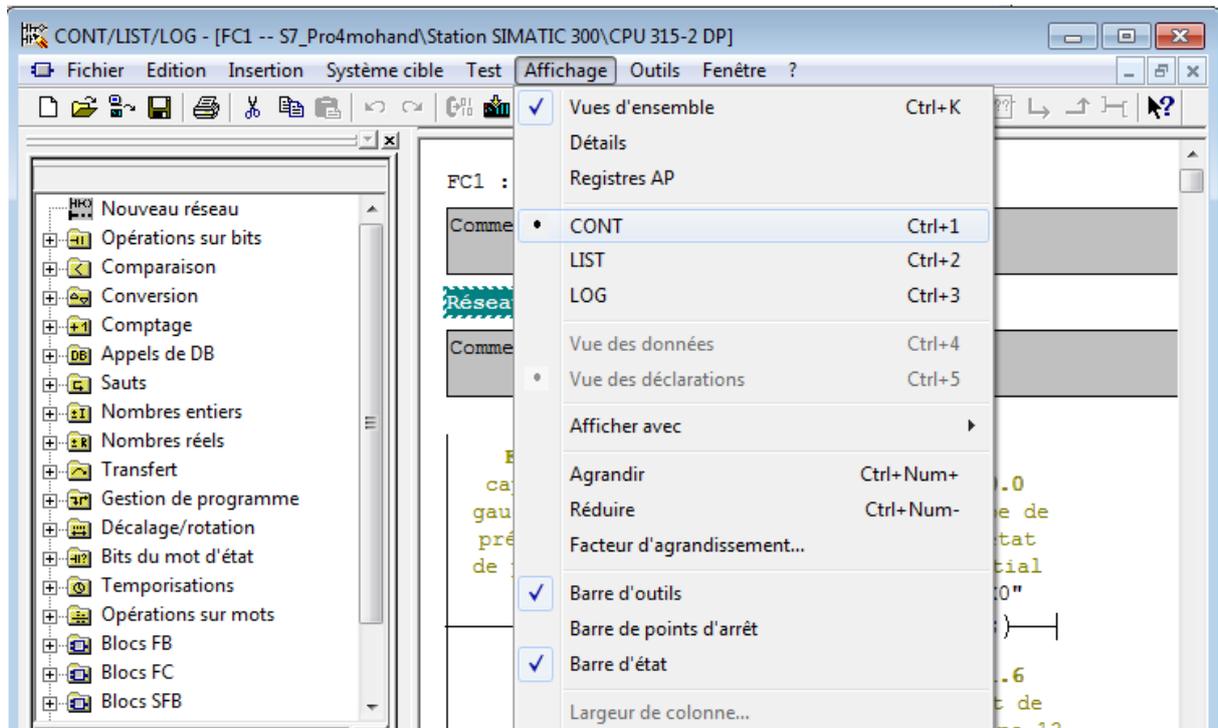


Figure. IV.12 : Fenêtre pour le choix du langage de programmation.

#### ➤ Exemple de programmation en LADDER

Comme exemple on choisit la programmation du GRAFCET de fonctionnement de la machine. En négligeant toutes la communication entre les autres automates.

**Propriétés de la table des mnémoniques**

Nom : Mnémoniques  
 Auteur :  
 Commentaire :  
 Date de création : 01/06/2013 22:51:44  
 Dernière modification : 14/06/2013 13:31:00  
 Dernier filtre sélectionné : Tous les mnémoniques  
 Nombre de mnémoniques : 40/40  
 Dernier tri : Mnémonique ordre croissant

Etat	Mnémonique	Opérande	Type de données	Commentaire
	ALARME	FC 3	FC 3	
	BL1	A 0.0	BOOL	levage de bras gauche
	BL2	A 1.0	BOOL	levage de bras 2
	BP1	A 0.2	BOOL	arrêt de rotation de la bobine
	BP2	A 0.4	BOOL	coupe colle
	C	E 0.4	BOOL	capteur de seul de l'épaisseur de la bonde
	CP	E 0.6	BOOL	coupé collé
	fc1	FC 1	FC 1	station gauche
	fc2	FC 2	FC 2	station droite
	init	E 0.0	BOOL	initialisation
	int bob1	A 0.1	BOOL	introduction de bobine 1
	INT BOB2	A 0.7	BOOL	INTRODUCTION DE BOBINE2
	K1	E 0.7	BOOL	bras levé
	k2	E 1.3	BOOL	bras 2 levé
	L	E 1.0	BOOL	capteur droit de présence du papier
	LF	A 0.5	BOOL	libération de frein
	M0.0	M 0.0	BOOL	etat de l'etat de X0
	Q	E 0.1	BOOL	capteur gauche de présence du papier
	resrv	E 1.1	BOOL	rouleau de reserve
	rot bob	A 0.3	BOOL	rotation de la bobine normalemet
	ROT BOB2	A 1.1	BOOL	
	s01	E 0.5	BOOL	bouton poussoir 'levage du bras de st1'
	s02	E 1.4	BOOL	bouton poussoir 'levage de bras droit'
	s03	E 1.2	BOOL	fin de course droit
	s04	E 0.3	BOOL	fin de course gauche
	stop bob	A 4.4	BOOL	stop bobine
	stop bob 2	A 0.6	BOOL	
	T1	T 1	TIMER	temp d'arrêt
	X1	M 0.1	BOOL	etat de l'etape1
	x11	M 1.0	BOOL	etat de l'etat 11
	x2	M 0.2	BOOL	etat de l'etape 2

SIMATIC

S7\_Pro4mohand\Station SIMATIC  
300\CPU 315-2 DP\Programme S7(5)\Mnémoniques

14/06/2013 13:43:48

Etat	Mnémonique	Opérande	Type de données	Commentaire
	x22	M 1.2	BOOL	etat de l'etape 22
	X3	M 0.3	BOOL	etat de l'etape 3
	x33	M 1.3	BOOL	etat de l'etape 33
	X4	M 0.4	BOOL	fonctionnement normal
	X44	M 1.4	BOOL	etat de l'etape 44
	x5	M 0.5	BOOL	freinage
	x6	M 0.6	BOOL	etat de l'etape 6
	x66	M 1.5	BOOL	etat de l'etape66
	X7	M 0.7	BOOL	etat de l'etape 7

Tableau IV.3 : Editeur de mnémonique

SIMATIC S7\_Pro4mohand\Station 02/07/2013 19:55:48  
 SIMATIC 300\CPU 315-2 DP\...\FC2 - <offline>

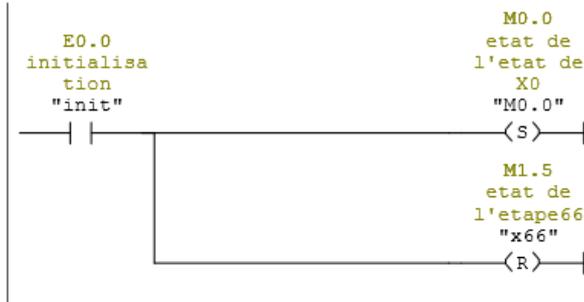
**FC2 - <offline>**

"fc2" station droite  
 Nom : Famille :  
 Auteur : Version : 0.1  
 Version de bloc : 2  
 Horodatage Code : 02/07/2013 19:41:55  
 Interface : 14/06/2013 11:21:12  
 Longueur (bloc/code /données locales) : 00224 00108 00000

Nom	Type de données	Adresse	Commentaire
IN		0.0	
OUT		0.0	
IN_OUT		0.0	
TEMP		0.0	
RETURN		0.0	
RET_VAL		0.0	

Bloc : FC2

Réseau : 1      etat de l'etat de X0

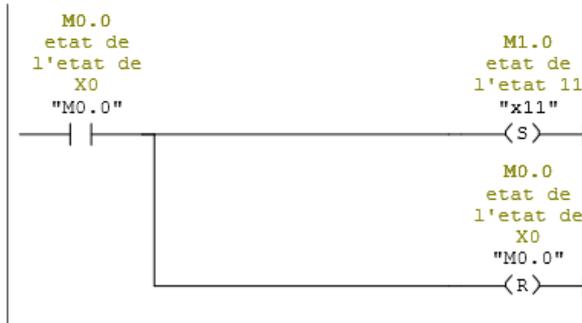


SIMATIC

S7\_Pro4mohand\Station  
SIMATIC 300\CPU 315-2 DP\...\FC2 - <offline>

02/07/2013 19:55:48

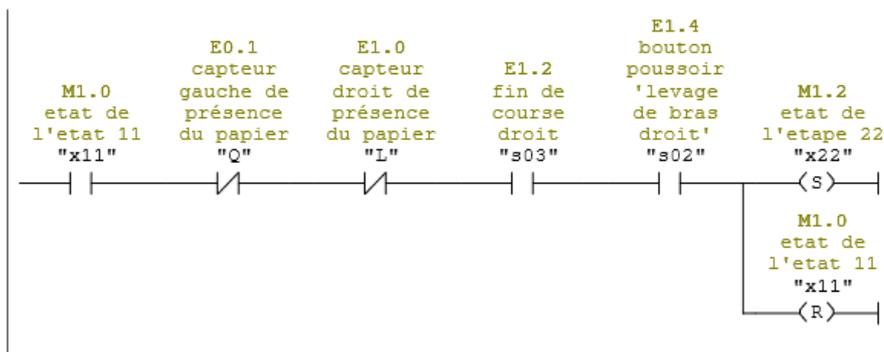
Réseau : 2      etat de l'etat 11



Réseau : 3      coupe colle



Réseau : 4      etat de l'etape 22



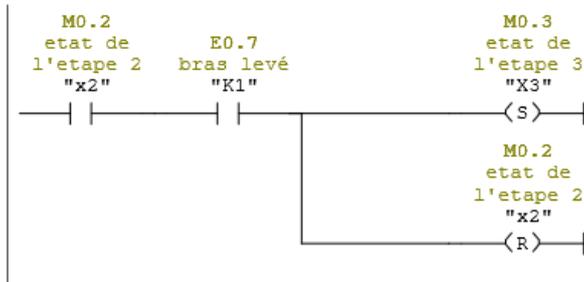
SIMATIC

S7\_Pro4mohand\Station

14/06/2013 13:48:47

SIMATIC 300\CPU 315-2 DP\...\FC1 - <offline>

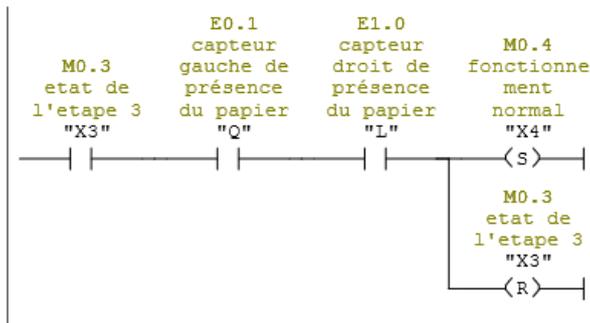
Réseau : 6      etat de l'etape 3



Réseau : 7      introduction de bobine 1



Réseau : 8      fonctionnement normal



SIMATIC

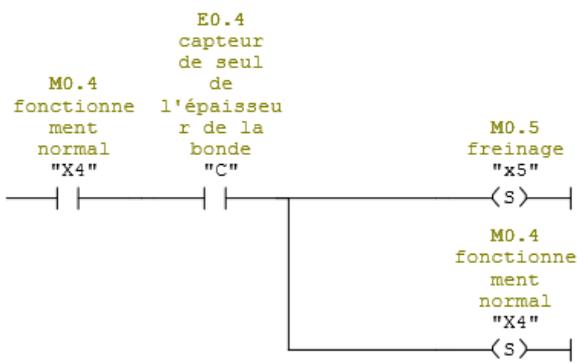
S7\_Pro4mohand\Station  
SIMATIC 300\CPU 315-2 DP\...\FC1 - <offline>

14/06/2013 13:48:47

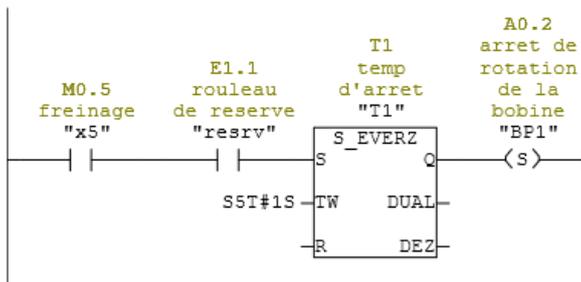
Réseau : 9 rotation de la bobine normalemet



Réseau : 10 freinage



Réseau : 11 arrêt de rotation de la bobine



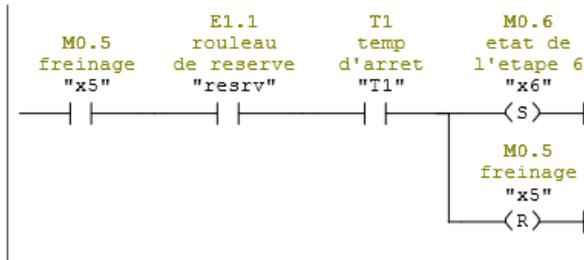
SIMATIC

S7\_Pro4mohand\Station

14/06/2013 13:48:47

SIMATIC 300\CPU 315-2 DP\...\FC1 - <offline>

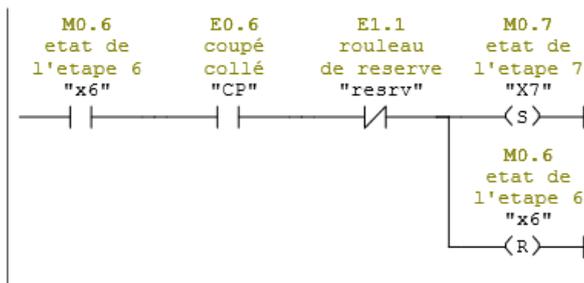
Réseau : 12 temp d'arrêt



Réseau : 13 rotation de la bobine normalemet



Réseau : 14 etat de l'etape 7



Réseau : 15 coupe colle



SIMATIC S7\_Pro4mohand\Station 14/06/2013 13:58:31  
 SIMATIC 300\CPU 315-2 DP\...\FC2 - <offline>

**FC2 - <offline>**

"fc2" station droite  
 Nom : Famille :  
 Auteur : Version : 0.1  
 Version de bloc : 2  
 Horodatage Code : 14/06/2013 13:22:45  
 Interface : 14/06/2013 11:21:12  
 Longueur (bloc/code /données locales) : 00222 00106 00000

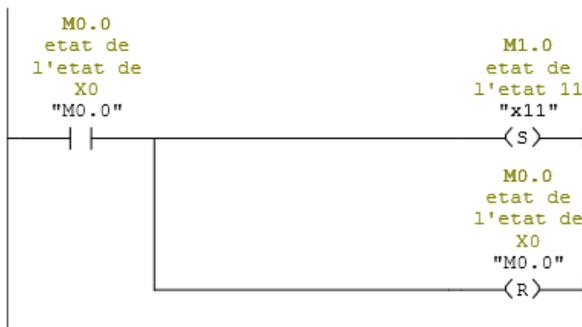
Nom	Type de données	Adresse	Commentaire
IN		0.0	
OUT		0.0	
IN_OUT		0.0	
TEMP		0.0	
RETURN		0.0	
RET_VAL		0.0	

Bloc : FC2

Réseau : 1 etat de l'etat de X0



Réseau : 2 etat de l'etat 11

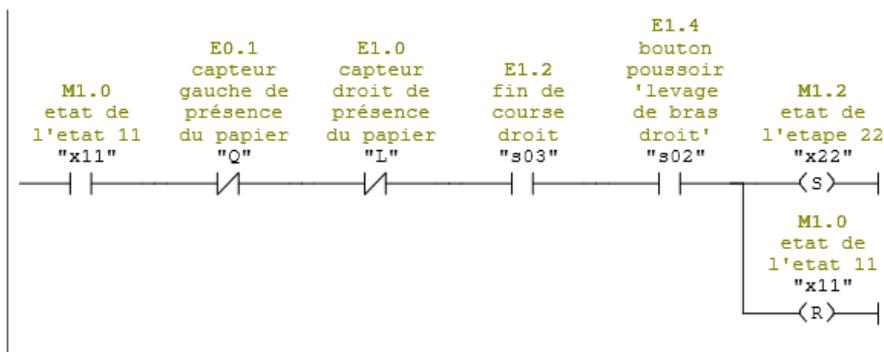


SIMATIC S7\_Pro4mohand\Station 14/06/2013 13:58:31  
 SIMATIC 300\CPU 315-2 DP\...\FC2 - <offline>

Réseau : 3 coupe colle



Réseau : 4 etat de l'etape 22



Réseau : 5 levage de bras 2

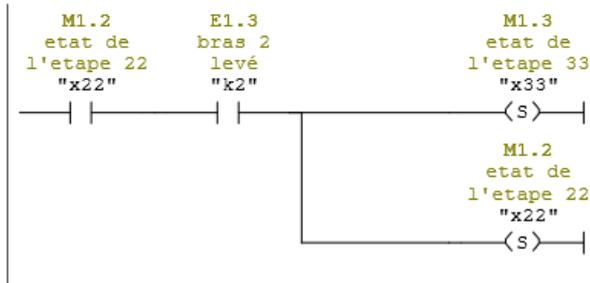


SIMATIC

S7\_Pro4mohand\Station  
SIMATIC 300\CPU 315-2 DP\...\FC2 - <offline>

14/06/2013 13:58:31

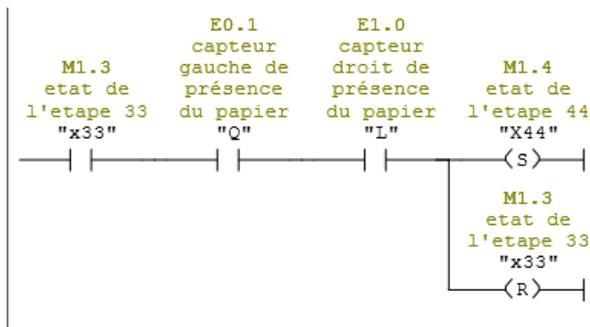
Réseau : 6      état de l'étape 22



Réseau : 7      INTRODUCTION DE BOBINE2



Réseau : 8      état de l'étape 33



SIMATIC

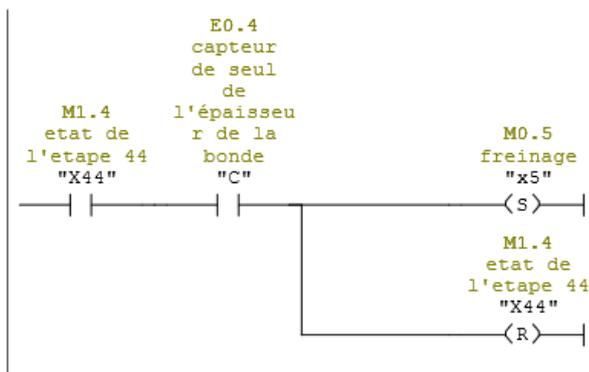
S7\_Pro4mohand\Station  
SIMATIC 300\CPU 315-2 DP\...\FC2 - <offline>

14/06/2013 13:58:31

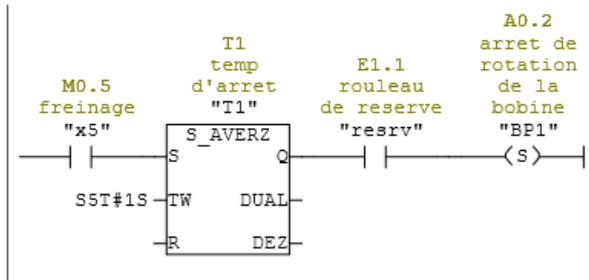
Réseau : 9 levage de bras 2



Réseau : 10 freinage



Réseau : 11 temp d'arrêt

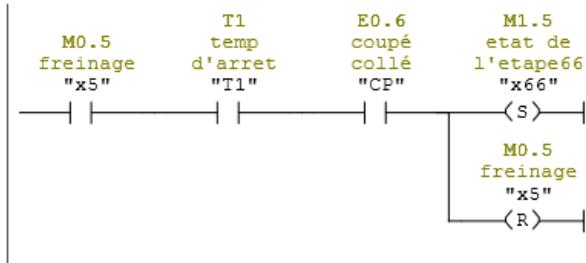


SIMATIC

S7\_Pro4mohand\Station  
SIMATIC 300\CPU 315-2 DP\...\FC2 - <offline>

14/06/2013 13:58:31

Réseau : 12      état de l'etape66



Réseau : 13      libération de frein



SIMATIC S7\_Pro4mohand\Station 02/07/2013 20:00:04  
 SIMATIC 300\CPU 315-2 DP\...\FC3 - <offline>

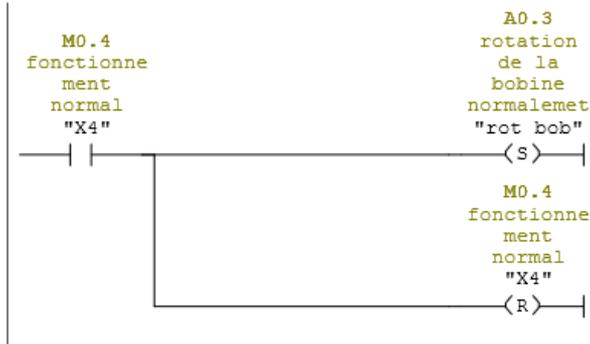
**FC3 - <offline>**

"ALARME"  
 Nom : Famille :  
 Auteur : Version : 0.1  
 Version de bloc : 2  
 Horodatage Code : 02/07/2013 19:44:20  
 Interface : 27/05/2013 23:09:45  
 Longueur (bloc/code /données locales) : 00114 00018 00000

Nom	Type de données	Adresse	Commentaire
IN		0.0	
OUT		0.0	
IN_OUT		0.0	
TEMP		0.0	
RETURN		0.0	
RET_VAL		0.0	

Bloc : FC3

Réseau : 1 rotation de la bobine normalemet

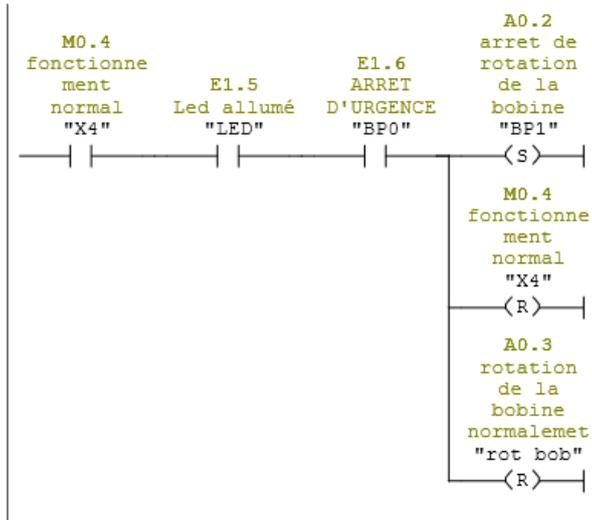


SIMATIC

S7\_Pro4mohand\Station  
SIMATIC 300\CPU 315-2 DP\...\FC3 - <offline>

14/06/2013 16:16:02

Réseau : 2      arret de rotation de la bobine



## **Conclusion**

Dans ce chapitre, nous avons présenté deux approches pour résoudre la problématique.

La première, c'est la méthode directe qui utilise le convertisseur S5/S7. Cette méthode nécessite la correction des erreurs générées par ce dernier.

La deuxième, consiste à faire une programmation avec le logiciel de programmation STEP 7 en se basant sur l'étude de mode de fonctionnement de la machine (GRAFCET). Après simulation, les résultats obtenus montrent une bonne cohérence entre l'état des sorties en fonction des entrées et le principe de fonctionnement de notre système.

Conclusion générale

## Conclusion générale

Nous avons effectué une migration de l'automate SIEMENS S5-90 vers l'automate SIEMENS S7-300 pour le pilotage du « *raccordeur CTS* » à l'entreprise *GENERAL EMBALLAGE*, puis, une généralisation de cette action sur toutes les machines de l'unité qui sont commandé par la série S5. La conception du programme en STEP7 pour S7300 est basée principalement sur le programme existant avec STEP5 ainsi que le cahier des charges du « *raccordeur CTS* », afin de corriger les erreurs générées par la migration.

Contrairement aux automates S5 Les automates programmables de la série S7-300 sont des appareils d'une grande flexibilité, ce qui permet leur utilisation dans de nombreux systèmes automatisés.

La firme SIEMENS offre un logiciel (SIMATIC MANAGER), très puissant pour la mise en œuvre de solutions à base d'API.

En effet, STEP 7 est un logiciel qui permet d'exploiter d'une manière optimale les fonctionnalités des différentes CPU de la gamme S7.

Ce projet nous a permis de nous familiariser avec les automates programmables S5 et S7 ainsi que leurs langages de programmation qui obéissent à des règles spécifiques.

Vu la complexité de travail réalisé, nous avons proposé deux approches. La première c'est la conversion directe avec un convertisseur intégrés dans le 'Step7'. Et la deuxième c'est l'élaboration d'un GRAFCET de fonctionnement de la machine qui nous donne une vision claire sur tous les paramètres rentrant dans son fonctionnement.

Nous avons à la fin, un programme convertis avec le convertisseur S5/S7 qui nécessite des corrections, et en se basant sur le cahier des charge nous avons proposé un programme édité à l'aide de logiciel 'Step7' pour l'implémenter dans le nouvel automate S7-300 qui va remplacer l'automate existant S5.

Le travail que nous avons réalisé, nous a permis de compléter notre formation théorique par un complément de connaissance du *STEP5* et *STEP7*, afin de faciliter le passage de l'ancien automate S5 vers S7.

## Bibliographie

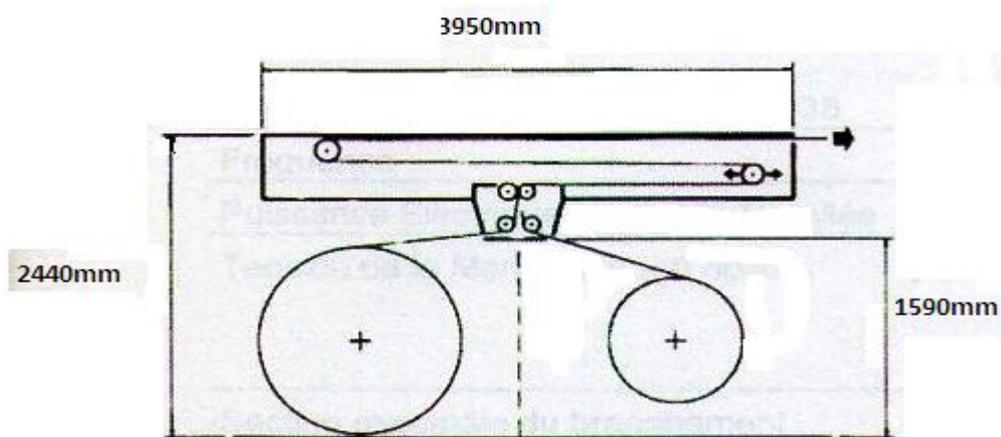
## Références bibliographiques

- [1] W.bolton, 'Les automates programmables industriels', Dunod, 2010.
- [2] Manuel système N°6ES5998-8MA32 Edition 01.1994
- [3] 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 ingénieur université de Bejaia, 2006.
- [4] P.JARGOT. 'Langages de programmation pour les API .norme CIE 11313, Technique de l'ingénieur ', S 8030 ,23 pages ,1999.
- [5] Bertrand. M. 'Automate programmable industriels'. Technique de l'ingénieur, S8015, page 14, 2001.
- [6] Document interne de General emballage 'Torres Medesa machine n°51.2309/2313'
- [7] 'Siemens Automation Innovation' 2009 site de siemens.
- [8] Manuel SIEMENS SIMATIC S7-300 Edition02 2003
- [9] Manuel Automate programmable S7-300. Installation et configuration –Caractéristiques des CPU Edition 2003.
- [10] Manuel SIEMENS STEP 7 « Pour une transition facile de S5 à S7 » Edition 05/2010
- [11] SIEMENS, « Mise en route STEP7 V5.2.Getting Started », Réf. 6ES7810-4CA06-8C A0, SIMATIC, 2002

## Annexes

## Données techniques

- Dimensions générales



## Spécifications

- Données mécaniques

Vitesse maximale du papier		450	m/min
Largeur maximale de la bobine		2.500	mm
Grammage Maximal		700	g/m <sup>2</sup>
Grammage Minimal		40	g/m <sup>2</sup>
Tension maximale de la bonde	Réserve simple	1000	N/m
	Double Réserve	1600	N/m

- Données électriques

Tension d'alimentation selon IEC38	400	V
Fréquence	50	Hz
Puissance électrique maximale installée	6	KVA

Tension de la Manœuvre (en option)	220	VAC
	110	VAC
	24	VDC
Section maximale de branchement	6	Mm <sup>2</sup>

- **Données pneumatiques**

Pression Maximale d'alimentation	10/145	bar/psi
Débit Minimal	3600(*)	Nm
Pression Maximale du travail	6/87	bar/psi
Pression Minimale du travail	4.5/65	bar/psi

(\*) : Données vrais, seulement pendant le cycle de travail.

- **Emission de bruits**

- Niveau de pression acoustique mesuré au poste de travail : 96 dB
- Pression acoustique maximale mesuré dans l'unité d'assemblage : 108 dB
- Les mesures ont été effectués avec un sonomètre digital Neurtek 840029, N<sup>0</sup> de série 0107940, conforme à l'échelle de décibels mesurée dans la machine. [6]



- Exemple de conversion de S5 vers S7 (arrêt d'urgence)

SIMATIC MOHAND1oiuytrez\SIMATIC 12/06/2013 00:38:01  
300(1)\CPU 314 IFM\Programme S7(1)\Mnémoniques

**Propriétés de la table des mnémoniques**

Nom : Mnémoniques  
 Auteur :  
 Commentaire :  
 Date de création : 09/06/2013 11:59:04  
 Dernière modification : 13/05/2013 13:14:26  
 Dernier filtre sélectionné : Tous les mnémoniques  
 Nombre de mnémoniques : 182/182  
 Dernier tri : Mnémonique ordre croissant

Etat	Mnémonique	Opérande	Type de données	Commentaire
	A32.0	A 32.0	BOOL	station 1 fonctionne
	A5.2	A 5.2	BOOL	FESTON EN AVANT
	B01 E02	A 32.3	BOOL	SIGNAL DE BASSE PRESSION
	B01 E05	E 34.1	BOOL	Détecteur de retour de frein
	B01E02	E 32.3	BOOL	signal DE basse pression
	B04 ROLL	E 34.0	BOOL	station sur le travail de détecteur de bras de cage de laminoir
	BATTRIE	E 35.3	BOOL	BATTERIE DE PLC FAIBLE
	CYCL_EXC	OB 1	OB 1	Cycle Execution
	E2.4	E 2.4	BOOL	arrêt normal
	E2.5	E 2.5	BOOL	arrêt rapide
	E2.6	E 2.6	BOOL	la décélération de la vitesse de ligne
	E2.7	E 2.7	BOOL	accélération de vitesse de ligne
	FC1	FC 1	FC 1	généralité
	FC2	FC 2	FC 2	arrêt d'urgence
	FF	A 0.1	BOOL	ACTION
	H06 E02	A 5.5	BOOL	ALARME
	k	E 32.0	BOOL	bouton poussoir pour l'arrêt d'urgence
	K04	A 32.1	BOOL	ÉPISSURE PRÊTE
	K05E01	A 4.6	BOOL	épissures
	KM01A	A 5.0	BOOL	DEPLACEMENT DR UNITÉ D'ÉPISSURE VERS LA GAUCHE
	KM02A	A 5.1	BOOL	FESTON EN AVANT
	KM02B	A 5.3	BOOL	FESTON VERS L'ARRIÈRE
	M	M 10.0	BOOL	SIGNAL DE CLIGNOTEMENT
	M 12.6	M 12.6	BOOL	épissure- impact
	M0.0	M 0.0	BOOL	memento d'arrêt d'urgence
	M0.1	M 0.1	BOOL	urgence
	M0.2	M 0.2	BOOL	station 1 fonctionne
	M0.3	M 0.3	BOOL	changement de station au-dessus d'impact
	M0.4	M 0.4	BOOL	le changement de station plus de rentrent
	M0.7	M 0.7	BOOL	changement de station

SIMATIC MOHANDloiytrez\ 12/06/2013 00:28:00  
SIMATIC 300(1)\CPU 314 IFM...\FC1 - <offline>

**FC1 - <offline>**

"FC1" généralité  
Nom : Famille :  
Auteur : Version : 0.1  
Version de bloc : 2  
Horodatage Code : 11/06/2013 22:44:32  
Interface : 03/05/2013 12:45:42  
Longueur (bloc/code /données locales) : 00974 00786 00000

Nom	Type de données	Adresse	Commentaire
IN		0.0	
OUT		0.0	
IN_OUT		0.0	
TEMP		0.0	
RETURN		0.0	
RET_VAL		0.0	

Bloc : FC1 généralité

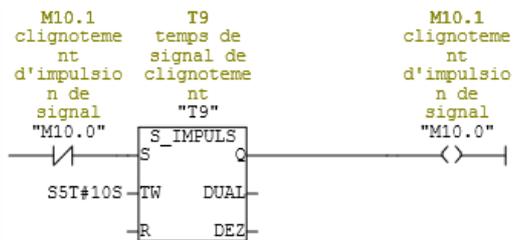
Réseau : 1 memento d'arrrt d'rgence

arret d'urgence



Réseau : 2 temps de signal de clignotement

clignatemet



**Programme convertis en S7**

SIMATIC MOHANDloiuuytrez\ 12/06/2013 00:28:00  
 SIMATIC 300(1)\CPU 314 IFM\...\FC1 - <offline>

**FC1 - <offline>**

"FC1" généralité  
 Nom : Famille :  
 Auteur : Version : 0.1  
 Version de bloc : 2  
 Horodatage Code : 11/06/2013 22:44:32  
 Interface : 03/05/2013 12:45:42  
 Longueur (bloc/code /données locales) : 00974 00786 00000

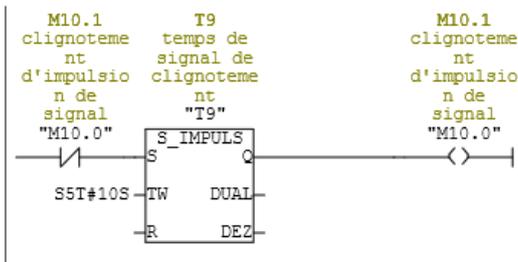
Nom	Type de données	Adresse	Commentaire
IN		0.0	
OUT		0.0	
IN_OUT		0.0	
TEMP		0.0	
RETURN		0.0	
RET_VAL		0.0	

Bloc : FC1 généralité

Réseau : 1 memento d'arrrt d'rgence  
 arret d'urgence



Réseau : 2 temps de signal de clignotement  
 clignatemet



SIMATIC

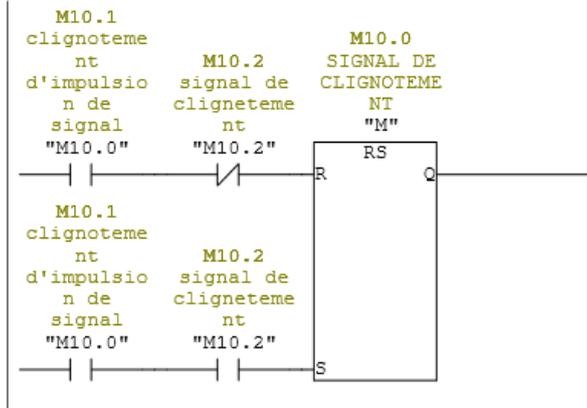
MOHAND1oiuytrez\

12/06/2013 00:28:01

SIMATIC 300(1)\CPU 314 IFM\...\FC1 - &lt;offline&gt;

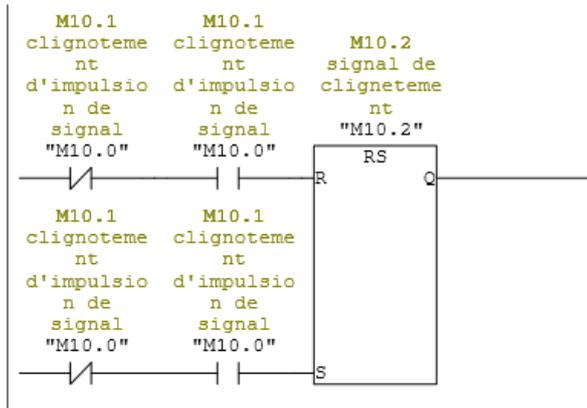
Réseau : 3

clignotement



Réseau : 4

signal de clignetement



SIMATIC

MOHAND1oiuytrez\

12/06/2013 00:28:01

SIMATIC 300(1)\CPU 314 IFM\...\FC1 - &lt;offline&gt;

Réseau : 5      changement de station

```

E32.4
station
sur le
changement
de
travail      M0.6      T16      T17      M0.5
avec         changement rétracter rétracter changement
bouton      de          l'impulsio l'impulsio de
poussoir   station    n de      n de      de
"s03 E02"  "M06"      "T16"    "T17"    "M05"
|-----|-----|-----|-----|-----|
|-----|-----|-----|-----|-----|

```

Réseau : 6      changement de station

```

E32.4
station
sur le
changement
de
travail      M0.6
avec         changement
bouton      de
poussoir   station
"s03 E02"  "M06"
|-----|-----|
|-----|-----|

```

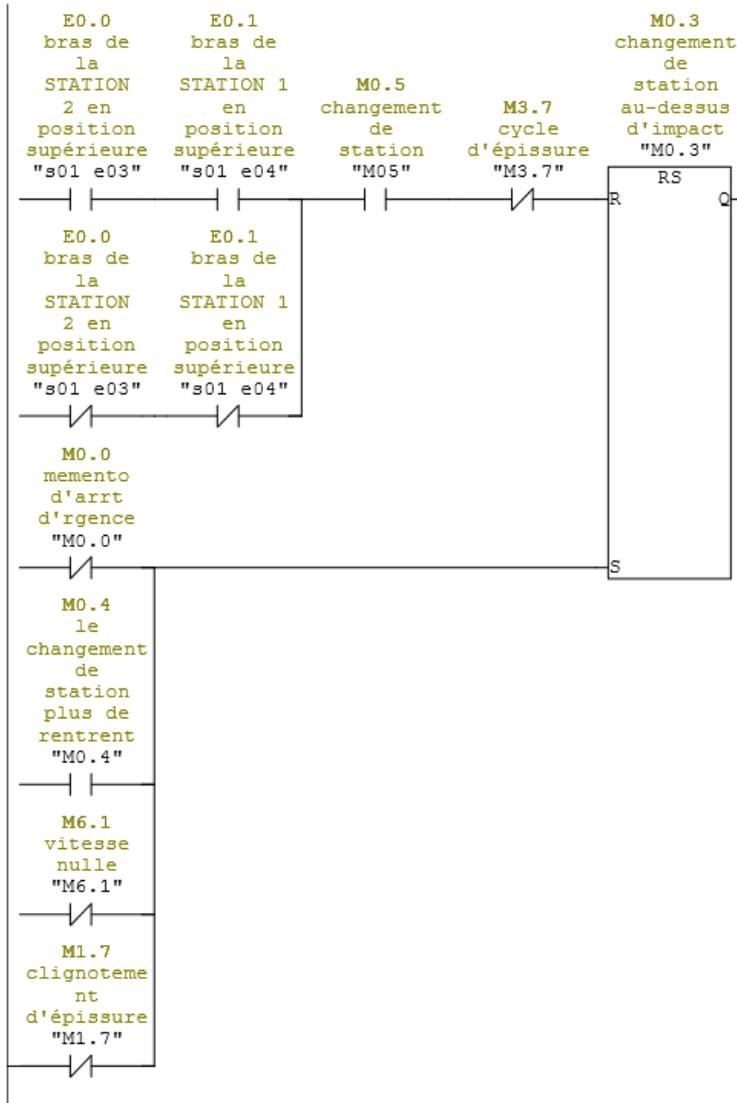
SIMATIC

MOHAND1oiuytrez\

12/06/2013 00:28:01

SIMATIC 300(1)\CPU 314 IFM\...\FC1 - &lt;offline&gt;

Réseau : 7      changement de station au-dessus d'impact



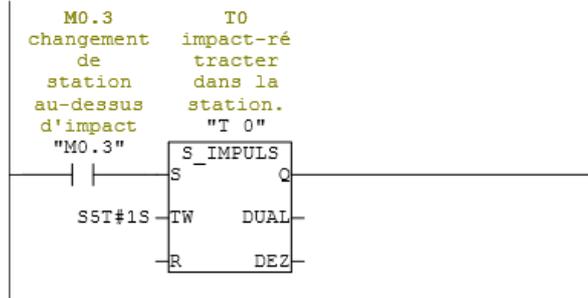
SIMATIC

MOHAND1oiuytrez\

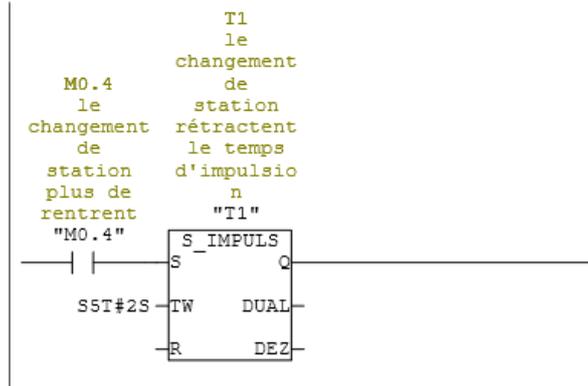
12/06/2013 00:28:01

SIMATIC 300(1)\CPU 314 IFM\...\FC1 - &lt;offline&gt;

Réseau : 8      impact-rétracter dans la station.



Réseau : 9      le changement de station rétractent le temps d'impulsion



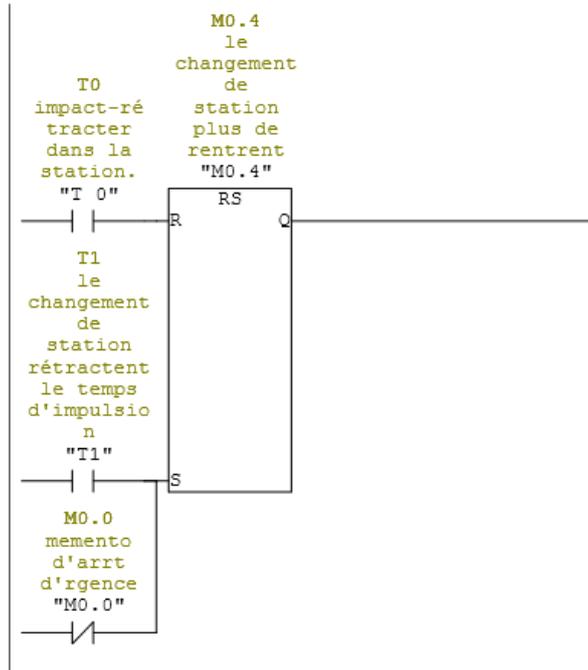
SIMATIC

MOHAND1oiuytrez\

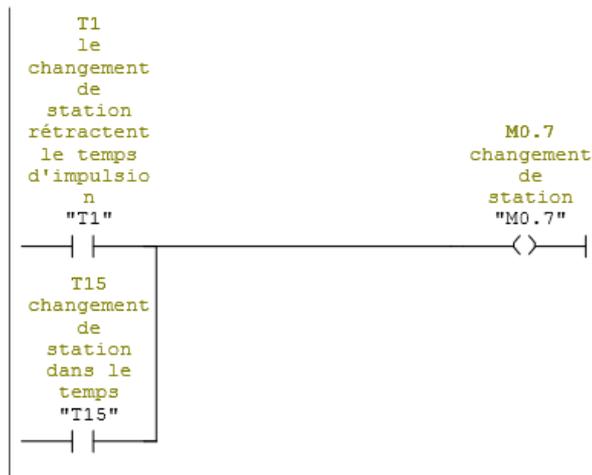
12/06/2013 00:28:01

SIMATIC 300(1)\CPU 314 IFM\...\FC1 - &lt;offline&gt;

Réseau : 10 le changement de station plus de rentrent



Réseau : 11 changement de station



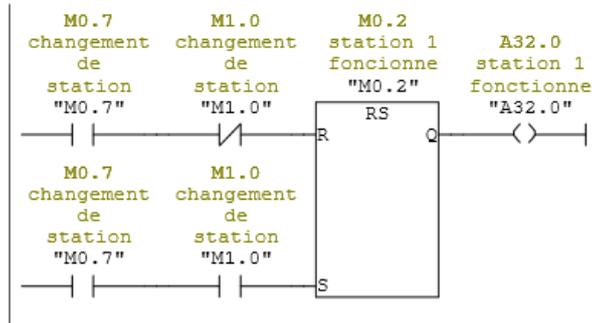
SIMATIC

MOHAND1oiuytrez\

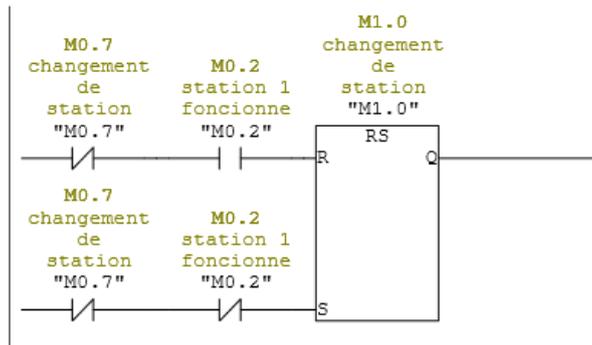
12/06/2013 00:28:01

SIMATIC 300(1)\CPU 314 IFM\...\FC1 - &lt;offline&gt;

Réseau : 12 station 1 fonctionne



Réseau : 13 changement de station



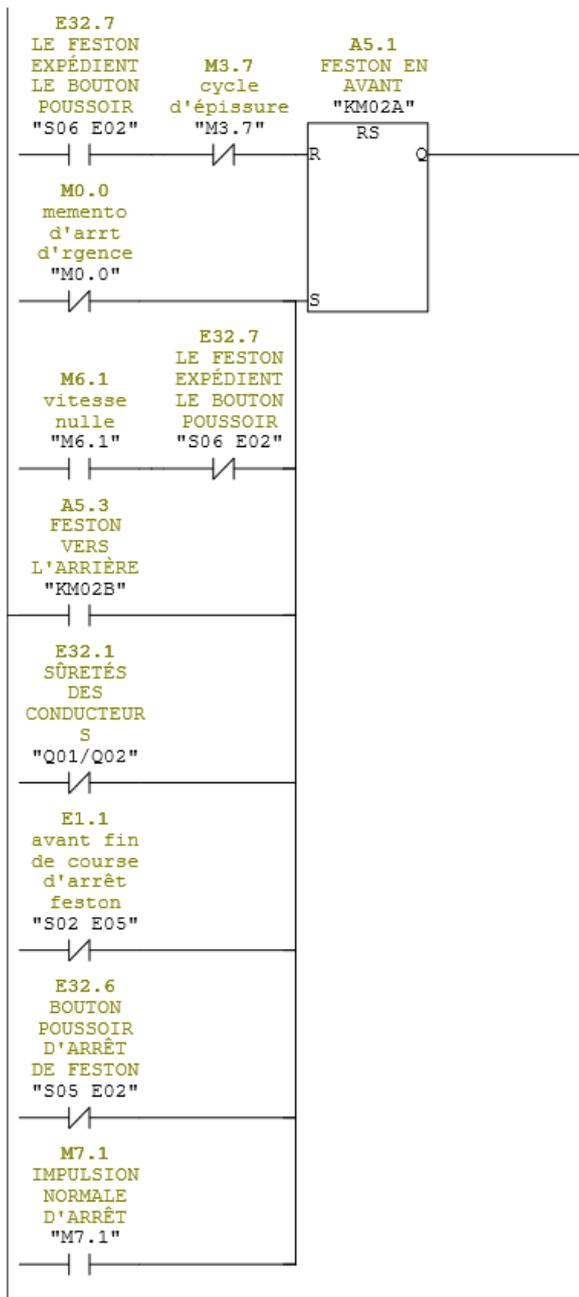
SIMATIC

MOHAND1oiuytrez\

12/06/2013 00:28:01

SIMATIC 300(1)\CPU 314 IFM\...\FC1 - &lt;offline&gt;

Réseau : 14 FESTON EN AVANT



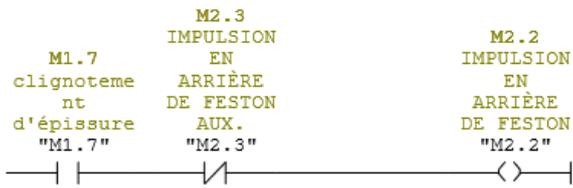
SIMATIC

MOHANDloiytrez\

12/06/2013 00:28:01

SIMATIC 300(1)\CPU 314 IFM\...\FC1 - &lt;offline&gt;

Réseau : 15      IMPULSION EN ARRIÈRE DE FESTON



Réseau : 16      IMPULSION EN ARRIÈRE DE FESTON AUX.



```

1wPB 1                                R: MED_ARST.S5D                                LAE=
Segmento 14                            FESTOON FORWARD
!                                     A 5.2
!E 32.7      M 3.7      +-----+
+---] [---+---]/[---+!S  !
!                                     !
!M 0.0                                     !
+---]/[---+-----+!R  Q!-
!                                     !
!M 6.1      E 32.7      !
+---] [---+---]/[---+
!                                     !
!A 5.3                                     !
+---] [---+-----+
!                                     !
!E 32.1                                     !
+---]/[---+-----+
!                                     !
!E 1.1                                     !
+---]/[---+-----+
!                                     !
!E 32.6                                     !
+---]/[---+-----+
!                                     !
!M 7.1                                     !
+---] [---+-----+

E   32.7 = S06 E02                FESTOON FORWARD PUSHBUTTON
M   3.7 = M3.7                    SPLICE CYCLE
A   5.2 = KM02A                   FESTOON FORWARD
M   0.0 = M0.0                    SAFETIES
M   6.1 = M6.1                    ZERO SPEED
A   5.3 = KM02B                   FESTOON BACKWARD
E   32.1 = Q01/Q02                DRIVES SAFETIES
E   1.1 = S02 E05                 FORWARD FESTOON STOP LIMIT SWITCH
E   32.6 = S05 E02                FESTOON STOP PUSHBUTTON
M   7.1 = M7.1                    NORMAL STOP PULSE

Segmento 15
!
!M 1.7      M 2.3                                     M 2.2
+---] [---+---]/[---+-----+-----+-----+-----+---( )

M   1.7 = M1.7                SPLICE READY BLINKING
M   2.3 = M2.3                FESTOON BACKWARD PULSE AUX.
M   2.2 = M2.2                FESTOON BACKWARD PULSE

Segmento 16
!
!M 1.7                                     M 2.3
+---] [---+-----+-----+-----+-----+---( )

4w+-----
!FECHA:          !M. TORRES DISENOS INDUSTRIALES !          ! CLIENTE: MEDE
! 20-08-2001    !          ! SIEMENS SA          ! MAQUINA N.:
!-----!CTRA.HUESCA KM9 TORRES DE ELORZ!          ! REFERENCIA: E
!          ! NAVARRA-SPAIN TFN:948/317811 ! SIMATIC 95U    ! PROYECTO N.:
!V. B.:F.LATASA !-----!
!          !REALIZADO: J. M. GONZALEZ    ! SIMATIC S5    !
+-----+

```





```

1wPB 1
R:MED_ARST.S5D
LAE=

Segmento 8
! T 0
!M 0.3 +-----+
+----] [----+!T!-10!
!KT 040.0 --!TW DU!-
! ! ! DE!-
! ! ! !
! ! ! !
! +-!R Q!-
! +-----+

M 0.3 = M0.3
T 0 = T0

STATION CHANGE OVER IMPACT
IMPACT - RETACT IN ST. CHANGE OVER

Segmento 9
! T 1
!M 0.4 +-----+
+----] [----+!T!-10!
!KT 020.0 --!TW DU!-
! ! ! DE!-
! ! ! !
! ! ! !
! +-!R Q!-
! +-----+

M 0.4 = M0.4
T 1 = T1

STATION CHANGE OVER RETRACT AUX.
STATION CHANGE RETRACT PULSE TIME

Segmento 10
! M 0.4
!T 0 +-----+
+----] [----+!S !
! ! ! !
!T 1 ! !
+----] [----+!R Q!-
! ! ! +-----+
!M 0.0 !
+----]/[----+

T 0 = T0
M 0.4 = M0.4
T 1 = T1
M 0.0 = M0.0

IMPACT - RETACT IN ST. CHANGE OVER
STATION CHANGE OVER RETRACT AUX.
STATION CHANGE RETRACT PULSE TIME
SAFETIES

Segmento 11
! T 1
!T 15 !
+----] [----+

STATION CHANGE OVER COMMON
M 0.7 ( )

4w+-----+
!FECHA: !M. TORRES DISENOS INDUSTRIALES ! ! CLIENTE: MEDE
! 20-08-2001 ! ! SIEMENS SA ! MAQUINA N.:
!-----+!CTRA.HUESCA KM9 TORRES DE ELORZ! ! REFERENCIA: 6
! ! NAVARRA-SPAIN TFN:948/317811 ! SIMATIC 95U ! PROYECTO N.:
!V. B.:F.LATASA !-----+
! ! REALIZADO: J. M. GONZALEZ ! SIMATIC S5 !
+-----+

```

```

1wPB 1                                R:MED_ARST.S5D                                LAE=
Segmento 14                            FESTOON FORWARD
!                                     A 5.2
!E 32.7   M 3.7   +-----+
+---] [---+---]/[---+!S   !
!                                     !
!M 0.0                                     !
+---]/[---+-----+!R   Q!-
!                                     !
!M 6.1     E 32.7   !
+---] [---+---]/[---+
!                                     !
!A 5.3                                     !
+---] [---+-----+
!                                     !
!E 32.1                                     !
+---]/[---+-----+
!                                     !
!E 1.1                                     !
+---]/[---+-----+
!                                     !
!E 32.6                                     !
+---]/[---+-----+
!                                     !
!M 7.1                                     !
+---] [---+-----+

E   32.7 = S06 E02                FESTOON FORWARD PUSHBUTTON
M   3.7 = M3.7                    SPLICE CYCLE
A   5.2 = KM02A                  FESTOON FORWARD
M   0.0 = M0.0                   SAFETIES
M   6.1 = M6.1                   ZERO SPEED
A   5.3 = KM02B                  FESTOON BACKWARD
E   32.1 = Q01/Q02               DRIVES SAFETIES
E   1.1 = S02 E05               FORWARD FESTOON STOP LIMIT SWITCH
E   32.6 = S05 E02              FESTOON STOP PUSHBUTTON
M   7.1 = M7.1                  NORMAL STOP PULSE

Segmento 15
!
!M 1.7     M 2.3                                     M 2.2
+---] [---+---]/[---+-----+-----+-----+-----+---( )

M   1.7 = M1.7                SPLICE READY BLINKING
M   2.3 = M2.3                FESTOON BACKWARD PULSE AUX.
M   2.2 = M2.2                FESTOON BACKWARD PULSE

Segmento 16
!
!M 1.7                                     M 2.3
+---] [---+-----+-----+-----+-----+---( )

4w+-----+-----+-----+-----+-----+-----+-----+-----+-----+
!FECHA:          !M. TORRES DISENOS INDUSTRIALES !          ! CLIENTE: MEDE
! 20-08-2001    !          !          ! SIEMENS SA    ! MAQUINA N.:
!-----+-----+-----+-----+-----+-----+-----+-----+-----+
!          !CTRA.HUESCA KM9 TORRES DE ELORZ!          ! REFERENCIA: E
!          ! NAVARRA-SPAIN TFN:948/317811 !          ! PROYECTO N.:
!V. B.:F.LATASA !-----+-----+-----+-----+-----+-----+-----+-----+-----+
!          !REALIZADO: J. M. GONZALEZ    ! SIMATIC S5  !
+-----+-----+-----+-----+-----+-----+-----+-----+-----+

```

```

PB 1                                R:MED_ARST.S5D                                LAE=47
Segmento 1                          EMERGENCY
!E 32.0                                M 0.0
+---] [---+-----+-----+-----+-----+-----+-----+-----+
E 32.0 = K                                SPLICER IN SERVICE
M 0.0 = M0.0                             SAFETIES
Segmento 2                          BLINKING DESTROYED
! T 9
!M 10.1 +-----+
+---]/[---+!T!-!0!
!DW 162 --!TW DU!-
! DE!-
! !
! !
! !
! +!R Q!+-----+-----+-----+-----+-----+-----+-----+
! +-----+
M 10.1 = M10.1                          BLINKING SIGNAL PULSE
DW 162 = DW162                          BLINKING TIME
T 9 = T9                                BLINKING SIGNAL TIME
Segmento 3                          BLINKING
! M 10.0
!M 10.1 M 10.2 +-----+
+---] [---+---]/[---+!S !
! !
!M 10.1 M 10.2 ! !
+---] [---+---] [---+!R Q!-
! +-----+
M 10.1 = M10.1                          BLINKING SIGNAL PULSE
M 10.2 = M10.2                          BLINKING SIGNAL AUX.
M 10.0 = M10.0                          BLINKING SIGNAL
Segmento 4                          M 10.2
!M 10.1 M 10.0 +-----+
+---]/[---+---] [---+!S !
! !
!M 10.1 M 10.0 ! !
+---]/[---+---]/[---+!R Q!-
! +-----+
M 10.1 = M10.1                          BLINKING SIGNAL PULSE
M 10.0 = M10.0                          BLINKING SIGNAL
M 10.2 = M10.2                          BLINKING SIGNAL AUX.
Segmento 5                          STATION CHANGE OVER PULSE
!E 32.4 M 0.6 T 16 T 17 M 0.5
+---] [---+---]/[---+---]/[---+---]/[---+---+-----+-----+-----+
4w+
!FECHA: !M. TORRES DISENOS INDUSTRIALES ! ! CLIENTE: MEDI
! 20-08-2001 ! ! SIEMENS SA ! MAQUINA N.:
!-----!CTRA.HUESCA KM9 TORRES DE ELORZ! ! REFERENCIA:
! ! NAVARRA-SPAIN TFN:948/317811 ! SIMATIC 95U ! PROYECTO N.:
!V. B.:F.LATASA !-----!
! !REALIZADO: J. M. GONZALEZ ! SIMATIC S5 !
+-----+

```

Convertisseur S5/S7

12/06/2013 00:44:12

```

// =====
//
// Automatiquement g n r par le convertisseur S5/S7, version : S7_S7BAS_P05.05.0
// 0.99 99.02.00.01,S7_S7BAS_FLOAT_P5.5.0.99-RELEASE-0505, 0099, 9902, 0001 12/06/2
// 013 00:43:25
// =====
//
// Nota :
// -----
// BLOCS : Il existe ds STEP 7 les types de blocs suiv. :
// OB : comme ds STEP 5
// FB : Function Block (par ex. r gulateur), est appel avec diff r
ents DB d'instance
// FC : Function Call, comparable au FB dans STEP 5
// Types blocs PB, FX, DX et SB n'existent plus.
// OB : Les OB ne peuvent pas tre appel s ds STEP 7 via programme utilis
ateur.
// Les OB aux fonct. sp c. comp. aux instructions sp ciales comme E
DB
// sont rempl. par de nouv. instr. ou p. SFC.
// ADRESSAGE : Ds S7, toutes les zones de m moire (E, A, M, D, etc) sont adress e
s
// par octets. L DW 2 dans STEP 5 devient ainsi L DBW 4 dans ST
EP 7.
// NOMBRE D'ACCU : CPU STEP 7 poss dent comme CPU STEP 5 2 ou 4 ACCU.
// Si un progr. STEP 5 d'une CPU S5 avec 2 ACCU (115U) est converti
pr une CPU S7
// avec 4 ACCU, il faut inclure l'instruction ENT (+,-,*,/) avant
// chaque op ration arithm tique, si le contenu de l'accumulateur 2
doit tre
// ult rieurement trait .
// LARG. ACCU : Les accus des CPU S 7 ont une larg. de 32 bits.
// Op. arith. sur n. entiers ds STEP 7 :
// - mot de pds fort reste inchang en cas de d b. haut/ bas de l'a
ccu 1 (135U : signe).
// - une conversion implicite au format 32 bits n'est ps effectu e
(155U).
// BIT DM : Le bit DM (d bord. m m.) reste dans STEP 7 inchang
// dans les op rations suivantes (il tait remis 0 dans STEP 5) :
//
// T, >F, <F, ><F, !=F, >=F, <=F, >D, <D, ><D, !=D, >=D, <=D,
// >G, <G, ><G, !=G, >=G, <=G, SLW, SRW, SLD, SVW, SVD, RLD, RRD.
// BIT DEB : Les op rations de conversion suiv. ne modifient pas l'accu 1 e
n cas de d b.
// (dans STEP 5 : valeur nomb. entier). Un d bord. est affich ds l
es bits DEB et DM (STEP 5 : uniq. DM).
// DUF, DUD, GFD.
// LIR, TIR : Pas d'instructions avec adress. absolu dans STEP 7.
// Elles sont en part. remplac. par de nouv. instruct. (par ex. acc
s mots > 255),
// et en partie par des fonctions syst me (SFC).
// Il en est de m me pr le registre de d but et de fin du DB et reg
istre base.
// ZONES DONNEES BS,BT,BA,BB,Q:
// Une interv. manuelle est n c. en cas d'util. de ces zones donn e
s.
// Solutions poss. :
// - Utiliser une zone m mentos largie.
// - D finir des DB plus longs.
// - Utiliser fonctions syst me S7.
// INSTRUC. ARRET: Il n'y a ds STEP 7 plus qu'une instruction d'arr t via appel S
FC .
// Elle correspond l'instruc. STS de STEP 5.
// Pour toute question, voir manuel ou aide en ligne.
//
FUNCTION FC 0 : VOID
NAME: GENERALI
BEGIN
NETWORK

```

E:\DESCENSEUR\_L3\_11\_2003\3011XXAC.AWL

[ 1]

## ***Résumé***

De nos jours, les industries utilisant les automates programmables S5-90U/95U de la firme allemande SIEMENS souffrent du manque flagrant de pièces de rechanges. Cependant, les automates S7-300 ont la particularité d'intégrer un module d'entrées/sorties TOR, un module d'entrées/sorties analogiques et un module fonctionnel. De plus, du point de vue efficacité ces dernières présentent plusieurs avantages à savoir :

- La rapidité d'exécution des tâches assez complexes,
- La localisation de défauts en temps réel,
- L'augmentation de la fiabilité de l'instrumentation.

Notre travail consiste à faire une migration de l'automate S5-90U/95U vers l'automate siemens S7300 pour un '*Raccordeur CTS*' (machine d'une chaîne de façonnement de carton), puis, généraliser cette action sur toutes les machines commandées par S5-90U/95U. La conception du programme en STEP7 pour S7300 est basée principalement sur le programme existant en STEP5 ainsi que le cahier de charge du *Raccordeur*:

***Mots clefs:*** automates programmable S5-90U/S5-95U/S7300, STEP7/STEP5, Raccordeur CTS.

## ***Abstract***

Today, industries using S5-90U/95U PLCs of the German firm Siemens suffer from serious lack of spare parts. However, the S7-300 are unique to integrate a digital input / output module, a module analog inputs / outputs and a functional module. In addition, the efficiency point of view the latter have several advantages including:

- Timeliness fairly complex tasks,
- The location of defects in real time,
- Increased reliability of instrumentation.

Our job is to make migration S5-90U/95U PLC to PLC Siemens S7300 for 'connector CTS' (string machine shaping cardboard), and then generalize this action on all controlled machinery by S5-90U/95U. The design of the program in STEP 7 for S7300 is based primarily on the existing program in STEP 5 and the terms of reference of the splicer.

***Key words:*** automats programmable S5-90U/S5-95U/S7300, STEP7/STEP5, CTS Connector.