

République Algérienne Démocratique et Populaire
Ministère De L'enseignement Supérieure et de la Recherche Scientifique
Université MOULOU D MAMMERI de tizi-ouzou



Faculté de Génie Electrique et d'Informatique
Département d'Electrotechnique

MEMOIRE DE FIN D'ETUDE

*En vue de l'obtention du Diplôme
Master Professionnel en Génie Electrique
Option Electrotechnique Industriel*

THEME

*Automatisation du système de
traitement de surface par
immersion à l'ENIEM*

Proposé par :

M^r. FEROUK Y.

Dirigé par :

M^r. FELLAG S.

Présenté par :

**M^r. HOCINI Karim
M^{elle}. BELMELLAT Kahina**

Promotion 2012/2013

REMERCIEMENTS

Nous remercions de prime abord le bon DIEU le maître suprême de temps et de circonstance, de nous avoir donné la santé, le courage et la volonté et de nous avoir permis de terminer nos études et élaborer ce modeste travail.

Nous tenons à exprimer nos sincères remerciements à notre copromoteur Mr FIROUK Yacine qui nous a proposé ce sujet et pour l'aide précieuse qu'il nous a apportée ainsi que toute l'équipe de maintenance.

Nous tenons à remercier vivement notre promoteur Mr FELLAG Sid Ali de nous avoir encadré et pour ses conseils, ainsi que la confiance qu'il a mis en nous tout au long de la préparation de ce projet.

Sans oublier Mr NOUALI Lounes pour son aide et sa disponibilité.

Nos remerciements s'adressent également à monsieur le président de jury et les membres du jury pour l'honneur d'avoir assister à notre soutenance et juger ce travail.

Nous tenons à remercier également nos amis (es) et nos familles pour leurs aides considérables.





Dédicaces

Je dédie ce modeste travail à :

La mémoire de mon très cher frère : MALEK,

Mes très chers parents;

Mes chers frères et sœurs ;

Ma belle-sœur;

Toute la famille HOCINI;

Tous mes amis (es);

Sans oublié ma binôme pour ses encouragements.

Karim





Dédicaces

Je dédie ce modeste travail à :

*La mémoire de mon très cher père : Si Mohand Larbi,
ainsi que mes grands parents ;*

Ma très chère mère : Zahra ;

Mes chers frères et sœurs ;

Mes beaux-frères et belles-sœurs ;

Mes neveux et nièces ;

Toute la famille BELMELLAT et DJOUZI ;

Tous mes amis (es) ;

En particulier ma très chère Sonia ;

Sans oublié mon binôme pour ses encouragements.

Kahina



Sommaire

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

Chapitre I : Description et présentation du système

| | |
|---|----|
| I Introduction..... | 3 |
| II Description et présentation de l'ancien système..... | 3 |
| II-1 Fonctionnement | 4 |
| III Description des différents composants de système..... | 6 |
| III-1 Thermostat 254..... | 6 |
| III-1-1 Généralités..... | 6 |
| III-1-2 Fonctionnement..... | 7 |
| III-2 Moteur asynchrone..... | 7 |
| a) Définition..... | 7 |
| b) Démarrage..... | 8 |
| III-3 Pompe à eau..... | 9 |
| III-4 Vanne manuelle..... | 10 |
| IV Nouveau système proposé..... | 11 |
| V Description des différents composants de nouveau système..... | 11 |
| V-1 Convoyeur..... | 11 |
| V-2 Les capteurs..... | 12 |
| V-2-1 Définition..... | 12 |
| V-2-2 Caractéristiques principales des capteurs..... | 12 |
| V-2-3 Choix d'un capteur..... | 12 |
| V-2-4 Capteur de température..... | 13 |
| V-2-4-1 Sonde Pt 100..... | 13 |
| a- Définition..... | 13 |
| b- Principe de la mesure..... | 13 |
| V-2-4-2 Sondes à thermocouple..... | 14 |
| a- Définition..... | 14 |
| b- Protection..... | 15 |
| V-2-5 Capteur inductif..... | 16 |
| V-2-6 Capteur de position..... | 17 |

Sommaire

| | |
|--|----|
| V-2-6-1 Les détecteurs de présence à action mécanique..... | 18 |
| V-3 Electrovanne..... | 18 |
| VI Conclusion..... | 19 |

Chapitre II : Les Automates programmables industriels (API)

| | |
|---|----|
| I introduction..... | 20 |
| II Système automatisé de production..... | 20 |
| III Structure de système automatisé de production..... | 21 |
| III-1 Partie opérative..... | 21 |
| III-2 Partie commande..... | 21 |
| III.3 Poste de contrôle..... | 22 |
| IV- de la logique câblée à l'automate programmable..... | 22 |
| V Automates programmables..... | 22 |
| V-1 Définition..... | 22 |
| V-2 Architecture d'un API..... | 23 |
| V-3 Nature des informations traitées par l'automate..... | 24 |
| V-4 Type des automates..... | 24 |
| V-5 Fonctionnement automate programmable industriel..... | 26 |
| V-6 Alimentation de l'automate programmable industriel..... | 26 |
| V-7 Fonctions réalisées..... | 26 |
| V-8 Les différentes firmes des automates..... | 27 |
| V-8-1 Automates Siemens..... | 27 |
| V-8-1-1 La famille S7..... | 27 |
| a-Automate Siemens SIMATIC S7-200..... | 27 |
| a-1 Fonctionnalités..... | 27 |
| a-2 Caractéristiques techniques..... | 27 |
| b-Automate Siemens SIMATIC S7-300..... | 27 |
| b-1 Fonctionnalités..... | 27 |
| b-2 Caractéristiques techniques..... | 27 |
| c-Automate Siemens SIMATIC S7-400..... | 28 |
| c-1 Fonctionnalités..... | 28 |

Sommaire

| | |
|--|----|
| c-2 Les caractéristiques techniques..... | 28 |
| V-8-1-2 Programmation de l'automate SIEMENS..... | 29 |
| a- Logiciel STEP7..... | 29 |
| a-1 Langage de programmation STEP7..... | 29 |
| a-2 Les opérations de STEP7..... | 31 |
| 1) Opérations combinatoires..... | 32 |
| 2) Opération de mémorisation..... | 32 |
| 3) Opération de changement et de transfert..... | 32 |
| 4) Opération de temporisation..... | 32 |
| 5) Opérations arithmétiques..... | 32 |
| 6) Opération de comptage..... | 32 |
| 7) Opération de comparaison..... | 33 |
| V-8-2 Automate Schneider..... | 33 |
| V-8-2-1 Télémécanique..... | 33 |
| a- Télémécanique Micro..... | 33 |
| b- Télémécanique Nano..... | 33 |
| V-8-2-2 Logiciel Schneider - Télémécanique PL7PRO | 33 |
| V-8-3 Automate Omron..... | 34 |
| a-Automate Omron Micro API compacts jusqu'à 320 E/S..... | 34 |
| b-Automate Omron modulaires jusqu'à 2500 E/S..... | 35 |
| IV-8-2-1 Logiciel Omron CX-One..... | 35 |
| V-9 Choix d'un automate..... | 35 |
| V-10 Avantage d'API..... | 36 |
| V-11 Inconvénients d'API..... | 36 |
| VI Conclusion..... | 36 |

Chapitre III : Le GRAFCET

| | |
|----------------------|----|
| I Introduction..... | 37 |
| II Le GRAFCET..... | 37 |
| II-1 Définition..... | 37 |

Sommaire

| | |
|--|----|
| II-2 Niveau du GRAFCET..... | 37 |
| III Les concepts de base du GRAFCET..... | 38 |
| III-1 Etape..... | 38 |
| III-2 Actions..... | 38 |
| III-3 Transition..... | 38 |
| IV Règles d'évolution du GRAFCET..... | 39 |
| IV-1 Condition initiale..... | 39 |
| IV-2 Franchissement d'une transition..... | 39 |
| IV-3 Evolution des étapes actives..... | 39 |
| IV-4 Franchissement simultané..... | 40 |
| IV-5 Conflit d'activation..... | 40 |
| V Les structures de base..... | 40 |
| V-1 Notion de Séquence..... | 40 |
| V-2 Saut d'étapes et reprise de séquence..... | 41 |
| V-3 Aiguillage entre deux ou plusieurs séquences (Divergence en OU)..... | 42 |
| V-4 Parallélisme entre deux ou plusieurs séquences (ou séquences simultanées ou divergence convergence en ET)..... | 42 |
| VI Mise en équation d'un GRAFCET..... | 43 |
| V-1 Règle générale..... | 43 |
| VII Hiérarchie des GRAFCET..... | 44 |
| VIII Le GEMMA..... | 46 |
| VIII-1 Pourquoi le GEMMA..... | 46 |
| VIII-2 les trois familles de procédures..... | 46 |
| IX GRAFCET du système (niveau 2)..... | 48 |
| X Conclusion..... | 53 |

Chapitre IV :Commende du système par API S7-300

| | |
|---|----|
| I Introduction..... | 54 |
| II Constitution de l'automate S7-300..... | 54 |
| II-1 Module d'alimentation (PS)..... | 54 |
| II-2 L'unité centrale(CPU)..... | 54 |
| II-3Modules de couplage (IM)..... | 54 |
| II-4Module de signaux (SM)..... | 55 |

Sommaire

| | |
|--|----|
| a) Modules d'entrées TOR..... | 55 |
| b) Modules de sorties TOR..... | 55 |
| c) Modules d'entrées analogiques..... | 55 |
| d) Modules de sorties analogiques..... | 55 |
| II-5 Module de fonction (FM)..... | 55 |
| II-6 Module de communication (CP)..... | 56 |
| II-7 Châssis d'extension (UR)..... | 56 |
| III Création d'un projet..... | 56 |
| III-1 Configuration et paramétrage du matériel SIMATIC300..... | 59 |
| III-1-1 Configuration..... | 59 |
| III-1-2 Paramétrage..... | 60 |
| III-2 Structure de programme..... | 61 |
| III-3 Blocs d'organisation..... | 62 |
| IV Le régulateur..... | 62 |
| IV-1 Les régulateurs continus..... | 62 |
| a- Le régulateur proportionnel (Type P)..... | 62 |
| b- Le régulateur intégral (type I)..... | 62 |
| c- Le régulateur dérivé (type D)..... | 62 |
| d- Le régulateur PID..... | 63 |
| IV-2 Réalisation d'un PID sous STEP7..... | 63 |
| a- Régulation continue avec le FB 41 « CONT_C »..... | 63 |
| b- Utilisation..... | 63 |
| c- Description..... | 64 |
| d- Schéma fonctionnel..... | 64 |
| V Présentation du S7-PLCIM..... | 89 |
| V-1 Commande de la CPU..... | 90 |
| V-1-1 Fenêtre CPU..... | 90 |
| V-2 Mise en route de logiciel S7-PLCSIM..... | 90 |
| V-3 Visualisation de l'état de programme..... | 92 |
| VI Conclusion..... | 92 |
| Conclusion générale..... | 93 |
| Bibliographie | |
| Annexe | |

Sommaire

Introduction générale

Les automates programmables industriels, sont apparus à la fin des années soixante, à la demande de l'industrie automobile américaine, qui réclamait plus d'adaptabilité de système de commande.

Depuis le début des années **80**, l'intégration des automates programmables pour le contrôle des différents processus industriels, est plus qu'indispensable. A l'origine, l'automate programmable était considéré comme une machine séquentielle, capable de remplacer des automatismes réalisés en logique traditionnelle, en apportant toutefois de profonds bouleversements dans la manière de concevoir et d'organiser le contrôle d'un processus. L'intégration de l'automate programmable renforce le degré de l'équipement, et offre une très grande adaptabilité face aux évolutions de l'environnement.

Présentation de l'entreprise

L'entreprise Nationale des Industriels de l'Electroménager (ENIEM) est issue de la restructuration de l'ex-société de fabrication et du montage du matériel électrique et électronique (SONELEC).

L'ENIEM a été créée en janvier 1983 à partir des fonctions déjà existantes au sein de l'entreprise mère, SONELEC, depuis 1974.

L'ENIEM a une gamme de produits très large. Elle produit tous types de réfrigérateurs, de congélateurs, de cuisinières, de climatiseurs, d'appareils de cuisson, de lavage, ainsi que les lampes à incandescence. Elle a aussi d'autres activités qui consistent dans le montage de petits appareils électroménagers domestiques (robots de cuisine, moulins à café, mixeurs, sèche cheveux, etc.).

L'ENIEM est organisée en trois principales unités de production qui sont l'unité de Froid, Cuisson et Climatisation, connue sous le nom de Complexe d'Appareils Ménagers (CAM), situé près de Tizi-Ouzou, l'unité Sanitaire de Miliana, et l'unité Lampes de Mohammedia.

Chacune des ces unités est elle-même organisée en différentes directions. Leader de l'électroménager en Algérie, l'ENIEM possède des grandes capacités de production et une expérience de plus de 30 ans dans la fabrication et le développement dans différentes branches de l'électroménager.

En 1987, l'ENIEM a commencé l'automatisation grâce à TOSHIBA par la mise en place de la chaîne de production réfrigérateur (R1) complètement automatique.

L'objectif de notre travail consiste à l'automatisation du système de traitement de surface par immersion ; afin de minimiser et de réduire l'intervention de l'opérateur humain et cela pour un meilleur rendement et plus de production (moins d'énergie et de temps perdus).

A cet effet, le contenu du mémoire est réparti en quatre chapitres, le premier chapitre est consacré à la présentation et description du système de traitement de surface, suivi de deuxième chapitre qui présente quelques automates programmables industriels.

La modélisation du système à l'aide de l'outil GRAFCET est l'objet de troisième chapitre ; dans le quatrième et dernier chapitre nous avons étudié la commande du système par API S7-300, et nous terminons notre travail par une conclusion générale.

I Introduction

Automatiser une tâche consiste à enchaîner les diverses opérations nécessaires à sa réalisation en limitant au maximum l'intervention d'un opérateur humain.

Les systèmes automatisés envahissent notre environnement industriel, là où ils tiennent une place très importante ; ils seraient donc difficile de concevoir un système de production sans avoir recours aux différentes technologies et composants qui forment le système automatisé de production.

II Description et présentation de l'ancien système



Figure I-1: Image du système

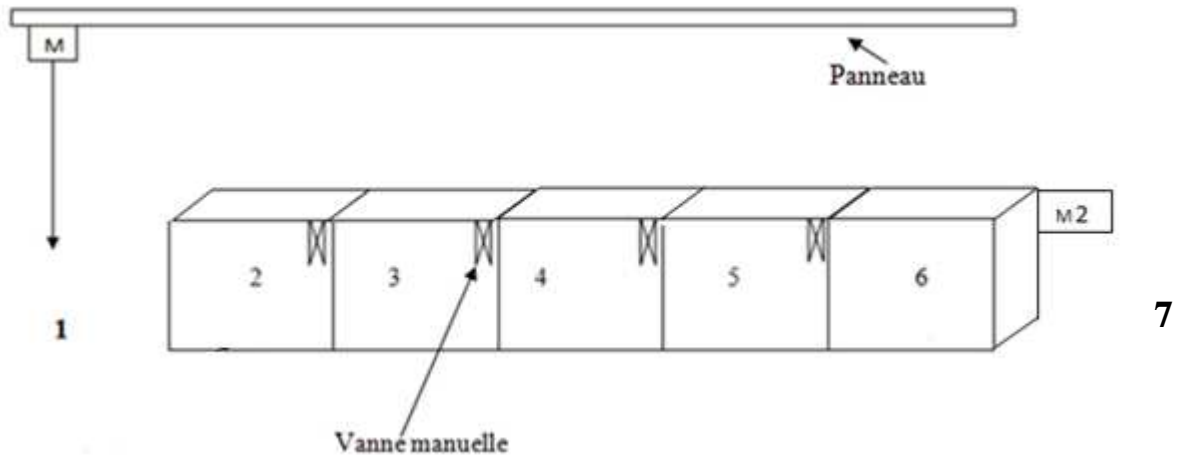


Figure I-2 : Schéma synoptique de l'ancien système

L'ensemble représente une chaîne industrielle de traitements de surfaces par immersion des pièces dans de différents bains d'eau à températures différentes. En plus de deux postes, l'un (1) situé à l'extrémité gauche de la chaîne, c'est l'endroit où le portique charge les corbeilles des pièces avant traitement, l'autre (7) situé à l'extrémité droite de la chaîne où le portique décharge les corbeilles des pièces après le traitement.

La montée ou la descente de la chaîne dans un bain est effectuée par un moteur asynchrone piloté dans les deux sens de rotation.

II-1 Fonctionnement

Poste 2 : dégraissage

Le dégraissage des pièces nécessite une régulation en température du bain. Un échangeur de chaleur par pompe assure la montée en température du bain.

- Température du bain à 80°C.
- Durée de traitement 15 minutes.

Constitution

- Thermorégulateur 254.
- Extracteur d'air.
- Agitateur.
- Pompe à eau.
- Vanne manuelle.

Poste 3 : premier rinçage

La durée de cette opération est d'environ 1 minute sans échauffer l'eau de rinçage.

Constitution

- Vanne manuelle

Poste 4 : deuxième rinçage

La durée de cette opération est d'environ 1 minute sans échauffer l'eau de rinçage.

Constitution

- Vanne manuelle.

Poste 5 : troisième rinçage

Un serpentin assure la montée en température du bain.

- Température du bain à 80°C.
- Durée de traitement 15 minutes.

Constitution

- Thermostat 254.
- Vanne manuelle.

Postes 6 : séchage

Un four assure le séchage des pièces traitées.

- Température du four à 80°C.
- Durée de séchage 15 minutes.

Constitution

- Thermostat 254.
- Moteur asynchrone.



Figure I-3 : Image de four de séchage

III Description des différents composants de système

III-1 Thermorégulateur 254

III-1-1 Généralités

Les thermorégulateurs 254 sont constitués par un thermostat et un clapet de réglage. Conçus comme modules d'assemblage, ils se prêtent au montage dans tous les types d'installations, depuis la maison individuelle jusqu'au complexe industriel, et répondent aux sévères exigences de qualité posées aux régulateurs de chauffage à distance. Grâce au mode d'assemblage par modules, l'emploi de la pièce de raccordement mixte 900/11 permet la régulation multiple avec un seul clapet. Ainsi s'offrent dans un très vaste champ d'application des solutions techniquement et économiquement pour le moins intéressantes [8].

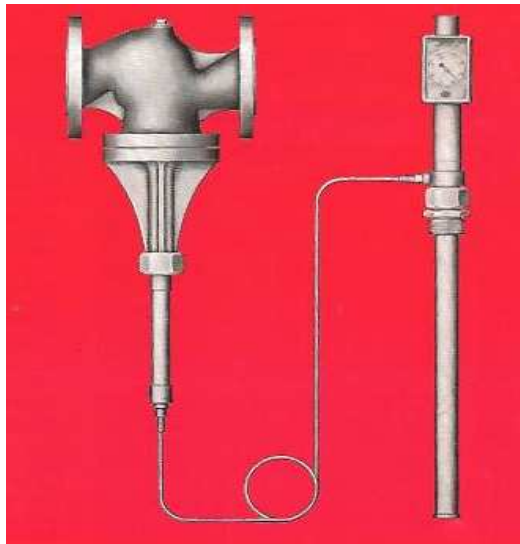


Figure I-4 : Thermorégulateur 254

III-1-2 Fonctionnement

Ces régulateurs fonctionnent en vertu de la dilatation des liquides et sont à action directe c'est-à-dire que tout accroissement de la température tend à fermer le clapet. Le bulbe, le tube capillaire et le transmetteur contiennent un liquide à coefficient de dilatation élevé. Toute variation de température enregistrée par la sonde se traduit par un changement de volume du liquide et, partant, par un mouvement qui est transmis à la tige de clapet, à ressort de rappel pour l'actionner. La bande de proportionnalité du régulateur est fonction du type de thermostat et de la levée de clapet maximale.

III-2 Moteur asynchrone

a) Définition

b) Le moteur asynchrone est une machine tournante aussi appelée moteur à induction, il fonctionne avec du courant alternatif. Sa particularité est de fonctionner avec un induit en court-circuit, sur lequel il n'y a pas de contacts électrique. Ceci permet de s'affranchir des problèmes liés aux collecteurs et aux balais, rencontrés dans le moteur à courant continu par exemple. Ce moteur peut, selon sa construction, être relié par son circuit inducteur à un réseau électrique alternatif monophasé ou polyphasé.

c) La machine asynchrone était à l'origine uniquement utilisée en moteur mais grâce à l'électronique de puissance ainsi qu'aux lois de commande, elle est de plus en plus souvent utilisée en génératrice, elle est donc une machine réversible en énergie. Bien que réversible la machine asynchrone est principalement (mais pas exclusivement) utilisée en moteur.



Figure I-5: Image moteur asynchrone

d) Démarrage

Deux sens de rotation

1 Circuit de puissance

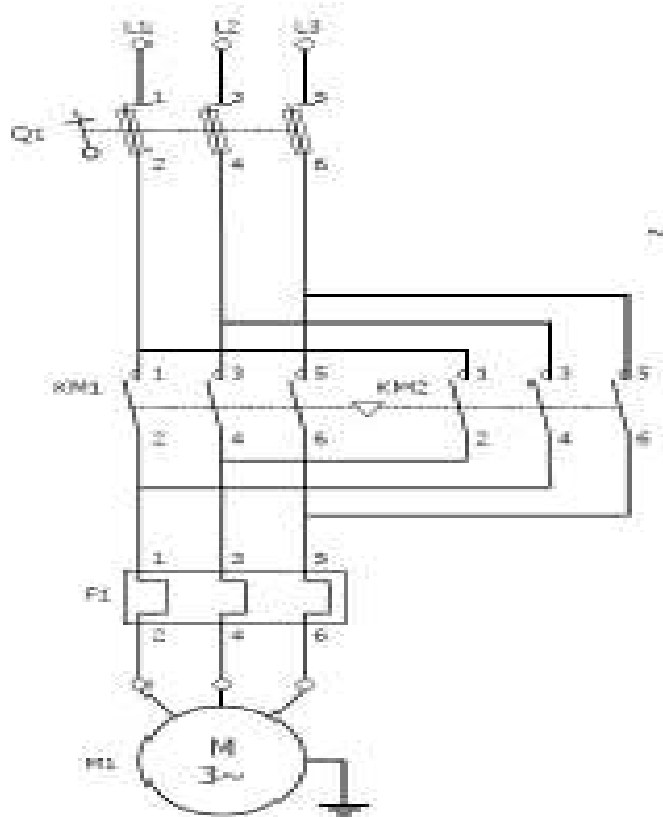


Figure I-6: Schéma circuit de puissance

2 Circuit de commande

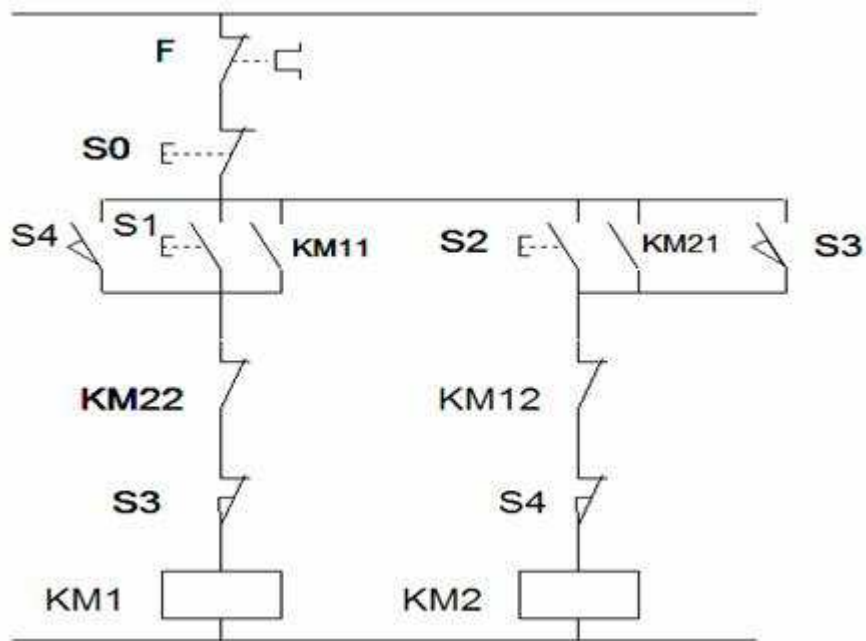


Figure I-7: Schéma circuit de commande

III-3 Pompe à eau

Une pompe est un dispositif permettant d'aspirer et de refouler un fluide. Elle est utilisée dans le premier bain pour faire circuler l'eau dans un échangeur de chaleur pour ajuster la température ; aussi elle refoule l'eau chaude dans le serpentin de quatrième bain.



Figure I-8: Image d'une Pompe à eau

III-4 Vanne manuelle

Elle est un dispositif qui sert à arrêter ou modifier le débit d'un liquide manuellement. Elle est utilisée pour remplir les bains.



Figure I-9: Image d'une vanne manuelle

IV Nouveau système proposé

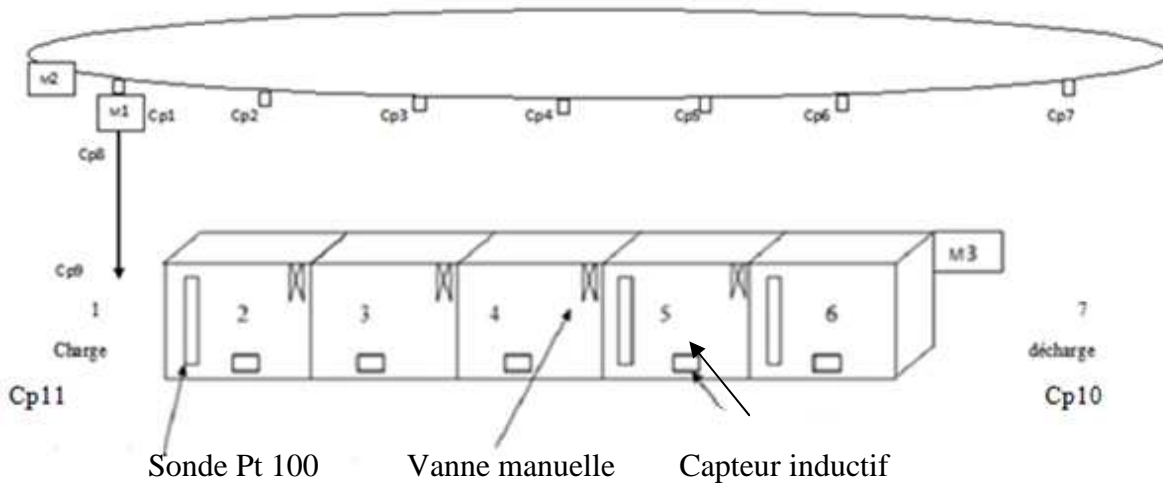


Figure I-10: Schéma synoptique du nouveau système proposé

Dans le nouveau système on utilise:

- Un convoyeur commandé par un moteur asynchrone (M1), a un seul sens de rotation à la place de panneau.
- Capteur de température (PT 100).
- Des capteurs inductifs.
- Des capteurs de position (C1.....C11).
- Electrovanne.

V Description des différents composants de nouveau système

V-1 Convoyeur

Un convoyeur est un mécanisme ou machine qui permet le transport d'une charge d'un point A à un point B.

V-2 Les capteurs

V-2-1 Définition

Un capteur est un organe de prélèvement d'information qui élabore à partir d'une grandeur physique, une autre grandeur physique de nature différente (très souvent électrique). Cette grandeur représentative de la grandeur prélevée est utilisable à des fins de mesure ou de commande.

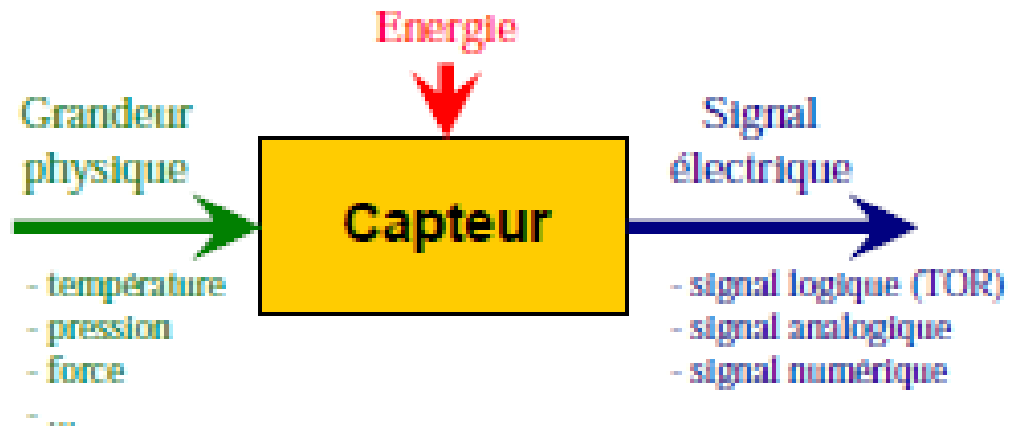


Figure I-11: schéma synoptique d'un capteur

V-2-2 Caractéristiques principales des capteurs

L'étendue de la mesure : C'est la différence entre le plus petit signal détecté et le plus grand perceptible sans risque de destruction pour le capteur.

La sensibilité : C'est la plus petite variation d'une grandeur physique que peut détecter un capteur.

La rapidité : C'est le temps de réaction d'un capteur entre la variation de la grandeur physique qu'il mesure et l'instant où l'information prise en compte par la partie commande.

V-2-3 Choix d'un capteur

Pour choisir correctement un capteur, il faudra définir tout d'abord :

- Le type et la nature de la grandeur physique à détecter.
- L'environnement de la grandeur physique.

V-2-4 Capteur de température

Les mesureurs de température analogiques transforment la température d'un fluide ou d'une enceinte en un signal 0/10V, 4-20mA.

V-2-4-1 Sonde Pt 100

a- Définition

La sonde Pt 100 est un capteur de température qui est utilisé dans le domaine industriel (agroalimentaire, chimie, raffinerie...). Ce capteur est constitué d'une résistance en Platine. La valeur initiale du Pt100 est de 100 ohms correspondant à une température de 0°C.



Figure I-12: Sonde Pt 100

b- Principe de la mesure

Le système de mesure injecte un courant de 1 mA dans la sonde Pt100 et mesure la tension à ses bornes.

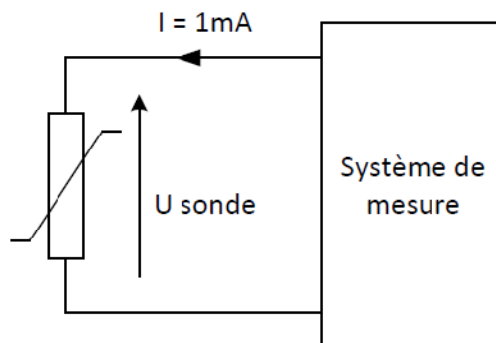
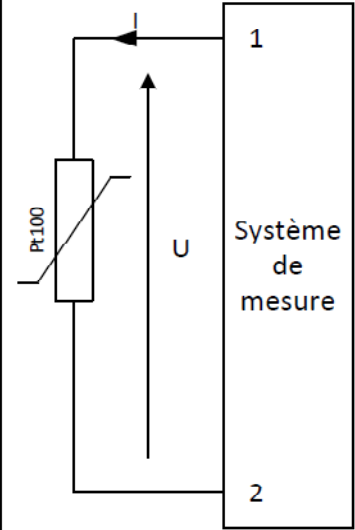
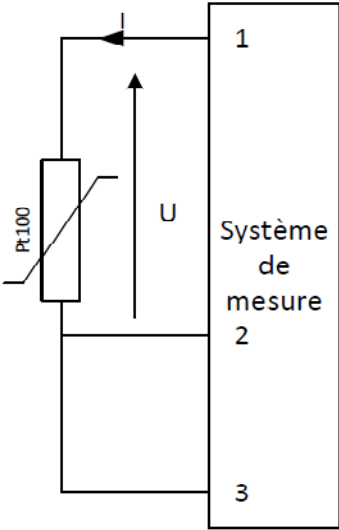
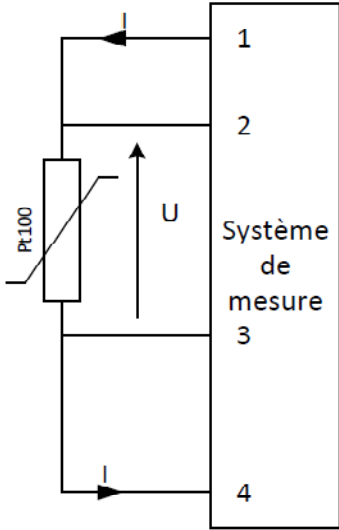


Figure I-13: Principe de la mesure Sonde PT 100

Chapitre I : Description et présentation du système

Les conducteurs de la liaison sonde – instrument de mesure, introduisent des erreurs de mesure, trois montages sont utilisés, selon la précision souhaitée.

| | Le plus simple | Le plus utilisé | Le plus précis |
|-----------|---|--|---|
| Montage | 2 fils | 3 fils | 4 fils |
| Principe |  |  |  |
| Précision | Mauvaise | Bonne | Très bonne |
| Liaison | Faible longueur | Moyenne longueur | Grande longueur |

Les Pt100 sont des sondes très utilisées dans l'industrie car elles sont simples à mettre en œuvre précises (montage 3 et 4 fils) et économiques.

V-2-4-2 Sondes à thermocouple

a- Définition

Un thermocouple est une sonde constitué par la soudure de deux conducteurs de matériaux différents. Cette soudure est mise en contact avec l'élément à mesurer.

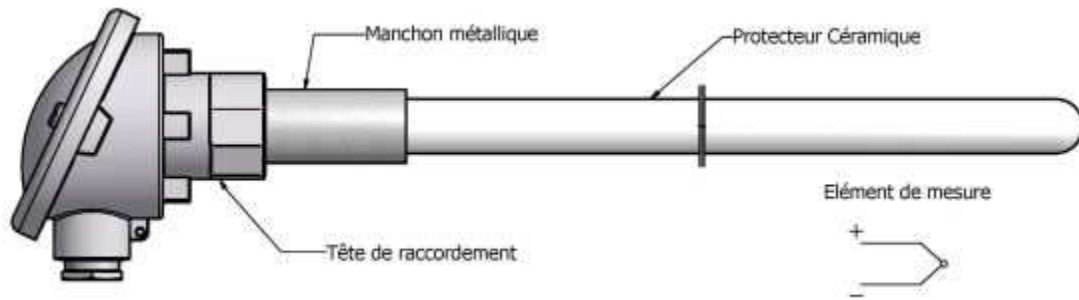
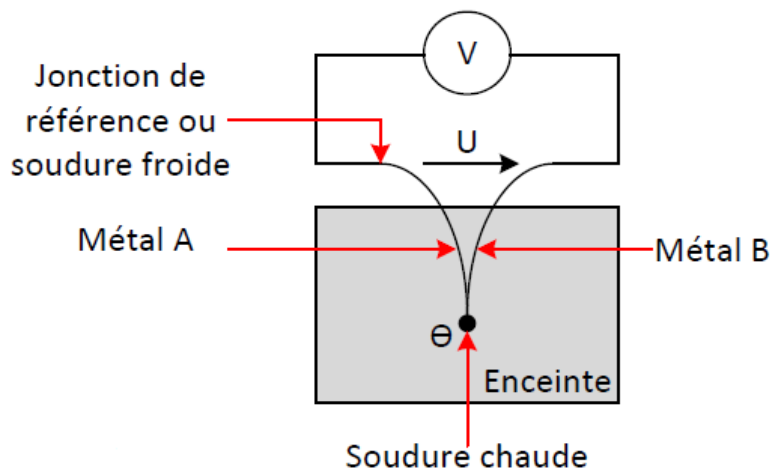


Figure I-14: Sondes à thermocouple

Un thermocouple produit une force électromotrice (fem) liée à la différence de température à laquelle sont soumises la soudure chaude et la soudure froide.



b- Protection

Les thermocouples peuvent être protégés ou non par un doigt de gant. Le thermocouple peut dans certains cas être connecté à la masse par l'intermédiaire de la protection.

| Désignation | SCA (Apparente) | SCI (Isolée) | SCM (Masse) |
|-----------------------------|------------------------------|-------------------------------|--|
| montage | | | |
| Caractéristique de la sonde | Thermocouple sans protection | Thermocouple protégé et isolé | Thermocouple protégé et en contact avec la masse |

Les thermocouples sont des sondes très utilisées dans l'industrie car ils sont robustes, précis, économiques et fonctionnent sur de larges gammes de température.

Dans notre cas on utilise une sonde à résistance (Pt100) comme élément sensible.

| Type de sonde | Avantages | Inconvénients |
|-----------------------|---|---|
| Résistance métallique | Durée de vie (6000 h) précision Pratiquement linéaire | Plage de mesure moins grande que les thermocouples |
| Thermocouple | Grande plage de mesure réaction rapide | Durée de vie (1000h) Non linéaire gestion de la soudure froide |

V-2-5 Capteur inductif

Les capteurs inductifs sont parmi les plus utilisés dans les systèmes automatisés.



Figure I-15: Image d'un capteur inductif

Ces capteurs se composent d'un oscillateur ayant pour fonction de générer un champ magnétique de fréquence 100 à 600Hz selon les modèles. Lorsqu'une pièce métallique pénètre

Chapitre I : Description et présentation du système

dans ces champs, elle est le siège de courants induits circulant qui se développent à sa périphérie. Ces courants constituent une surcharge pour le système oscillateur et entraînent de ce fait une réduction de l'amplitude des oscillations au fur et à mesure de l'approche de l'objet métallique, jusqu'à blocage complet. La détection est effective lorsque la réduction de l'amplitude des oscillations est suffisante pour provoquer un changement d'état de la sortie du détecteur.

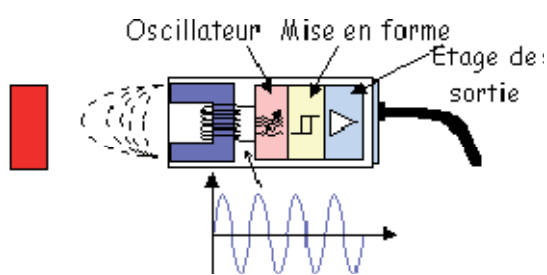


Figure a

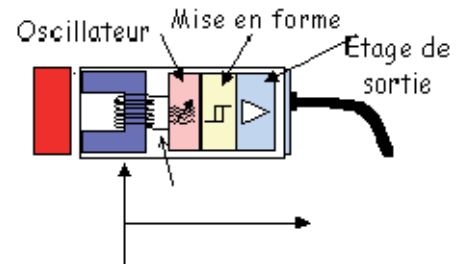


Figure b

Le branchement de ces capteurs est à 2 fils ou 3 fils:

- 2 fils : constitue directement le contact ouvert (NO) ou fermé (NC) au repos selon le cas.
- 3 fils : 2 fils d'alimentation et un fil de sortie qui peut être à collecteur ouvert (NPN) ou à émetteur ouvert (PNP).

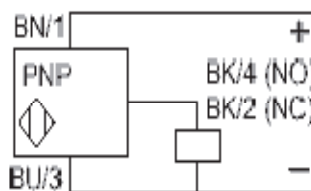


Figure c

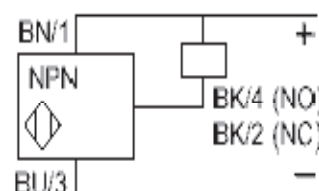


Figure d

V-2-6 Capteur de position

Les capteurs de positions sont les plus répandus dans les automatismes.

Ils sont utilisés pour détecter:

- La position précise d'un objet.
- La présence d'un objet.
- Le niveau d'un fluide.
- L'épaisseur d'une pièce.

- L'angle de rotation d'un arbre.

V-2-6-1 Les détecteurs de présence à action mécanique

Ce sont des commutateurs commandés par le déplacement d'un organe de commande (corps d'épreuve). Lorsqu'ils sont actionnés, ils ouvrent ou ferment un ou plusieurs circuits électriques ou pneumatiques. Ce sont des détecteurs Tout Ou Rien (TOR).

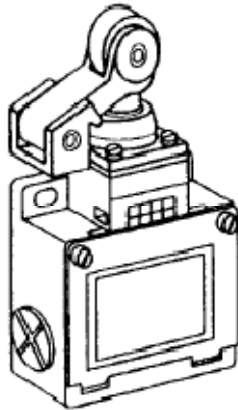


Figure I-16: Image d'un détecteur de présence à action mécanique.

V-3 Electrovanne

L'électrovanne est une vanne qui sert à contrôler l'écoulement d'un fluide, dont le fonctionnement est commandé électriquement par un électroaimant.



Figure I-17: Image d'une électrovanne

VI Conclusion

Avant l'automatisation de notre chaîne de traitement de surface, nous avons fait l'étude générale des différents composants du système. L'étude menée nous a permis de mieux connaître le système, de lui apporter quelques modifications et de mener une étude d'automatisation.

Mais ce système de traitement de surface reste semi-automatique car il ne peut pas se passer de l'intervention de l'opérateur humain pour le chargement et le déchargement des corbeilles des pièces avant et après le traitement, malgré l'industrie de nos jours qui se tourne vers une production automatisée, passant de la production manuelle à faible rendement à une production robotisée et simplifiée.

Dans la suite de notre travail, nous allons étudier les automates programmables industriels.

I introduction

L'automatisation s'impose de plus en plus, aussi bien à l'échelle de la production que de la formation. Afin d'automatiser les processus techniques, on a recourt souvent dans l'industrie à des automates programmables industriels (API) comme système directeur pour la commande de chacun des composants utilisés pour l'automatisation ainsi que le rôle de communication dans le cadre de la production.

II Système automatisé de production

L'objectif de l'automatisation des systèmes est de produire, en ayant recours le moins possible à l'homme, des produits de qualité et ce pour un coût le plus faible possible.

Un système automatisé est un ensemble d'éléments en interaction, et organisés dans un but précis : agir sur une matière d'œuvre afin de lui donner une valeur ajoutée.

Le système automatisé est soumis à des contraintes : énergétiques, de configuration, de réglage et d'exploitation qui interviennent dans tous les modes de marche et d'arrêt du système [2].

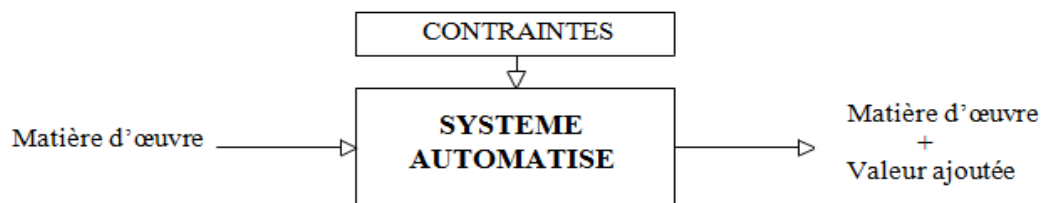


Figure II-1: Système automatisé de production

III Structure de système automatisé de production

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

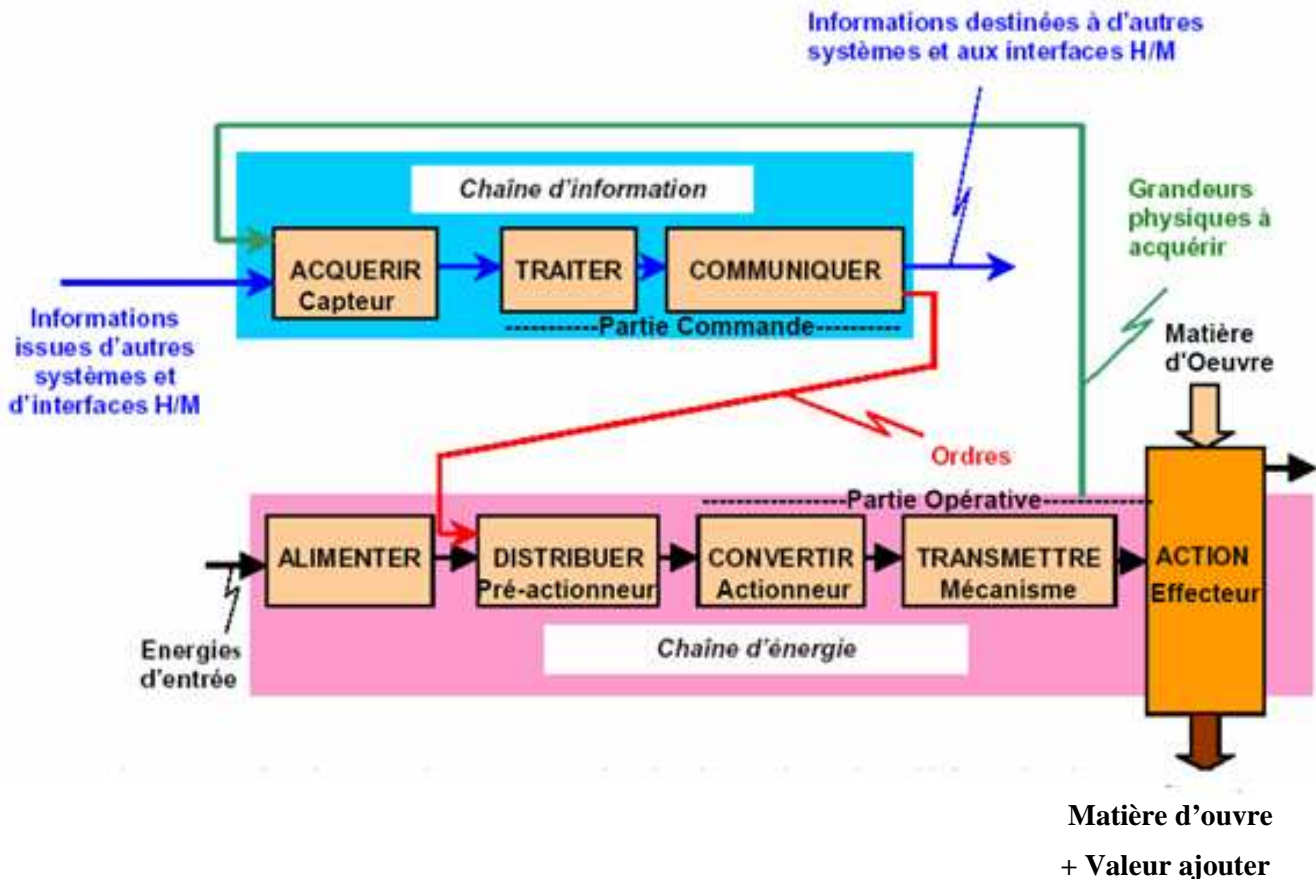


Figure II-2 : Structure de système automatisé de production

III-1 Partie opérative

Elle agit sur la matière d'œuvre afin de lui donner sa valeur ajoutée. Les actionneurs (moteurs, vérins) agissent sur la partie mécanique du système qui agit à son tour sur la matière d'œuvre. Les capteurs / détecteurs permettent d'acquérir les divers états du système.

III-2 Partie commande

Elle donne les ordres de fonctionnement à la partie opérative. Les préactionneurs permettent de commander les actionneurs ; ils assurent le transfert d'énergie entre la source de puissance (réseau électrique, pneumatique ...) et les actionneurs. Exemple : contacteur, distributeur ... Ces préactionneurs sont commandés à leur tour par le bloc traitement des

Chapitre II : Les Automates programmables industriels (API)

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

III-3 Poste de contrôle

Composé des pupitres de commande et de signalisation, il permet à l'opérateur de commander le système (marche, arrêt, départ cycle ...). Il permet également de visualiser les différents états du système à l'aide de voyants, de terminal de dialogue ou d'interface homme-machine (IHM).

IV De la logique câblée à l'automate programmable

Le programme de commande des automates à logique câblée, couramment utilisées dans le passé, était déterminé par le câblage des contacteurs et des relais, spécifique à la tâche à exécuter.

Aujourd'hui, on utilise des automates programmables pour résoudre les tâches d'automatisation. La logique stockée dans la mémoire programme du système d'automatisation est à l'aide d'une console de programmation [3].

V Automates programmables

V-1 Définition

L'Automate Programmable Industriel (API) est un appareil électronique programmable, adapté à l'environnement industriel, qui réalise des fonctions d'automatisme pour assurer la commande de préactionneurs et d'actionneurs à partir d'informations logique, analogique ou numérique [4].

V-2 Architecture d'un API

La structure interne d'un API peut se représenter comme suit :

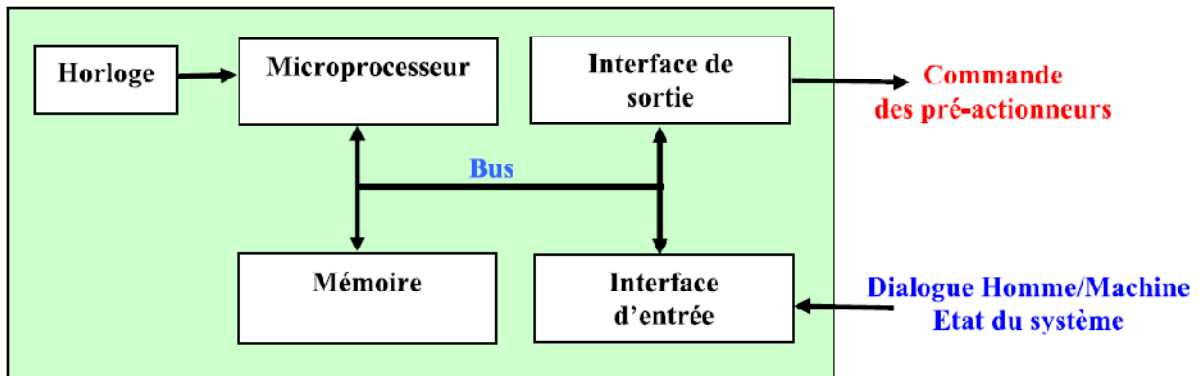


Figure II-3 : La structure interne d'un API

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

Un API se compose donc de trois grandes parties :

Le processeur

Le microprocesseur réalise toutes les fonctions logique ET, OU, les fonctions de temporisation, de comptage, de calcul... à partir d'un programme contenu dans sa mémoire.

Il est connecté aux autres éléments (mémoire et interface E/S) par des liaisons parallèles appelées BUS qui véhiculent les informations sous forme binaire.

La zone mémoire

La zone mémoire va permettre :

- De recevoir les informations issues des capteurs d'entrées.
- De recevoir les informations générées par le processeur et destiner des sorties (valeur des compteurs, des temporisations,...).
- De recevoir et conserver le programme du processus.

Chapitre II : Les Automates programmables industriels (API)

Les interfaces Entrées/Sorties

Les entrées reçoivent des informations en provenance des éléments de détection (capteurs) et du pupitre opérateur (BP).

Les sorties transmettent des informations aux pré-actionneurs (relais, électrovanne...) et aux éléments de signalisation (voyants).

V-3 Nature des informations traitées par l'automate

Les informations peuvent être de type :

Tout ou rien (T.O.R.) :

L'information ne peut prendre que deux états (vrai/faux, 0 ou 1 ...). C'est le type d'information délivrée par un détecteur, un bouton poussoir ...

Analogique :

L'information est continue et peut prendre une valeur comprise dans une plage bien déterminée. C'est le type d'information délivrée par un capteur (pression, température..).

Numérique :

L'information est contenue dans des mots codés sous forme binaire ou bien Hexadécimale. C'est le type d'information délivrée par un ordinateur ou un module intelligent.

V-4 Type des automates

Les automates peuvent être de type compact ou modulaire.

De type **compact**, intègre le processeur, l'alimentation, les entrées et les sorties. Selon les modèles et les fabricants, il pourra réaliser certaines fonctions supplémentaires (comptage rapide, E/S analogiques ...) et recevoir des extensions en nombre limité. Ces automates, de fonctionnement simple, sont généralement destinés à la commande de petits automatismes [4].



Figure II-4: Automate de type compact

De type **modulaire**, le processeur, l'alimentation et les interfaces d'entrées / sorties résident dans des unités séparées (modules) et sont fixées sur un ou plusieurs racks contenant le "fond de panier" (bus plus connecteurs). Ces automates sont implémentés dans les systèmes complexes, et dans les automatismes dont la capacité de traitement et flexibilité sont nécessaires.

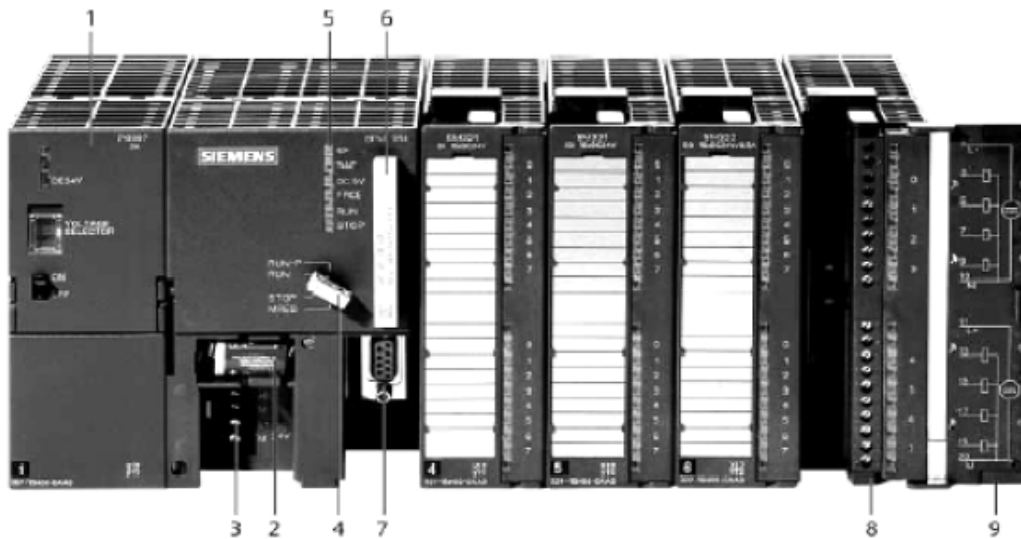


Figure II-5 : Automate modulaire (Siemens)

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

V-5 Fonctionnement des automates programmables industriels

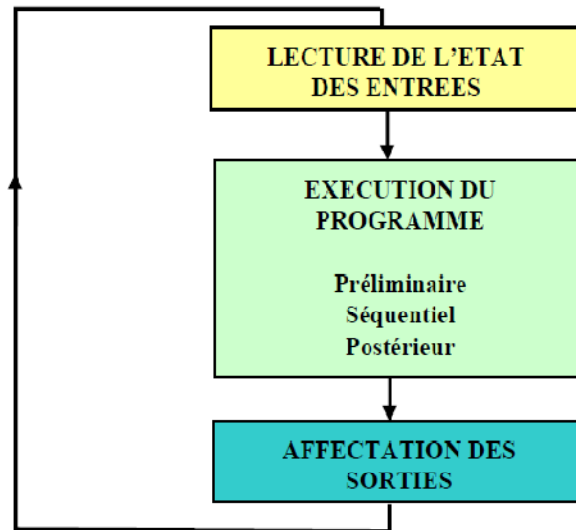


Figure II-6 : Fonctionnement automate programmable industriel

V-6 Alimentation de l'automate programmable industriel

L'alimentation intégrée dans l'API, fournit à partir des tensions usuelles des réseaux (230 V, 24 V=) les tensions continues nécessaires au fonctionnement des circuits électroniques.

V-7 Fonctions réalisées

Les **automates compacts** permettent de commander des sorties en T.O.R et gèrent parfois des fonctions de comptage et de traitement analogique.

Les **automates modulaires** permettent de réaliser de nombreuses autres fonctions grâce à des modules intelligents que l'on dispose sur un ou plusieurs racks. Ces modules ont l'avantage de ne pas surcharger le travail de la CPU car ils disposent bien souvent de leur propre processeur.

V-8 Les différentes firmes des automates

V-8-1 Automates Siemens

SIEMENS est l'une des sociétés très connue dans le domaine de la fabrication des automates programmable, a développé plusieurs types des A.P.I parmi ceux la famille S7, comme **S7 200**, **S7-300** et le **S7-400**.

V-8-1-1 La famille S7

a-Automate Siemens SIMATIC S7-200

a-1 Fonctionnalités

Le micro automate SIMATIC S7-200 est compact, performant en terme de temps réel et communicant. Le logiciel de programmation Micro/WIN contient les outils et langages nécessaires pour toute la gamme S7-200.

a-2 Caractéristiques techniques

Les configurations du SIMATIC S7-200 sont de faible encombrement avec 5 Unités Centrales et différents modules d'extension. Des extensions spécifiques permettent la mise en réseau sur PROFIBUS, Ethernet/Internet, modem et la commande de positionnement des moteurs. Les ports de communication assurent la mise en réseau d'éléments extérieurs pour la conduite et la supervision. L'automate dispose de cartouches de sauvegarde pour une grande capacité mémoire.

b-Automate Siemens SIMATIC S7-300

b-1 Fonctionnalités

Le S7-300 est une plate-forme d'automatisation universelle pour des applications avec des architectures centralisées et décentralisées, orientée sécurité, motion control ou avec interface Ethernet/PROFINet intégrée. Le S7-300 peut également s'intégrer dans des solutions compactes avec HMI ou dans des têtes de station pour traitement intelligent décentralisé.

b-2 Caractéristiques techniques

- 24 CPU standards: avec interface Ethernet/PROFINet intégrée .
- CPUS de sécurité.
- CPU technologiques pour la gestion des fonctions motion control.
- Modules d'E/S TOR et analogiques pour la quasi-totalité des signaux avec possibilité de traitement des interruptions et du diagnostic.

c-Automate Siemens SIMATIC S7-400

c-1 Fonctionnalités

Le S7-400 est un automate programmable dont lequel chaque tâche d'automatisation peut être résolue par un choix approprié des constituants d'un S7-400. Il possède de très hautes performances pour les applications de moyennes et hautes gammes, en plus la possibilité d'extension à plus de 300 modules. Les modules S7-400 se présentent sous forme de boîtiers que l'on adapte sur un châssis. Des châssis d'extension sont à disposition pour faire évoluer le système.

c-2 Les caractéristiques techniques

-Châssis

On dispose des châssis suivants pour le S7-400 :

Le Châssis universel URI/UR2 peut être utilisé comme châssis central ou châssis d'extension. Il possède 18/9 emplacements simples largeurs.

ERI/ ER2 sont des châssis d'extension.

CR2 est un châssis central pour le fonctionnement multiprocesseur asymétrique.

-Unités centrales.

-(C P U) de capacités différentes avec E/S intégrées.

-Module d'alimentation PS.

-(Power Supply) pour la conversion des tensions réseaux alternatives ou continues en tensions 5V ou 24V.

-Modules de signaux SM pour entrées et sorties numériques et analogiques.

-Modules de fonction FM assurent des fonctions de positionnement, régulation et comptage.

-Les modules CP (port de communication) permettent de raccorder une CPU aux différents réseaux.

V-8-1-2 Programmation de l'automate SIEMENS

La programmation des automates de la famille S7 se fait par la console de programmation ou par PC et sous un environnement WINDOWS, via le langage de programmation STEP7.

STEP7 présente trois langages de programmations :

- STEP7 basis: CONT, LOG, LIST.
- S7_SCL.
- S7-GRAPH.

Le langage le plus utilisé est le Step7 basis

a- Logiciel STEP7

STEP7 est le logiciel de base pour la configuration et la programmation de système d'automatisation SIMATIC. Il est formé d'un ensemble d'applications avec les quelles nous pouvons aisément réaliser des taches partielles comme:

- la configuration et le paramétrage du matériel.
- la création et le test de programme utilisateur.
- la configuration de réseau et de liaison.

Autrement dit nous permet de créer et de gérer des projets et des programmes, il permet ensuite le chargement de programmes dans des systèmes cibles et le test de l'installation d'automatisation [2].

a-1 Langage de programmation STEP7

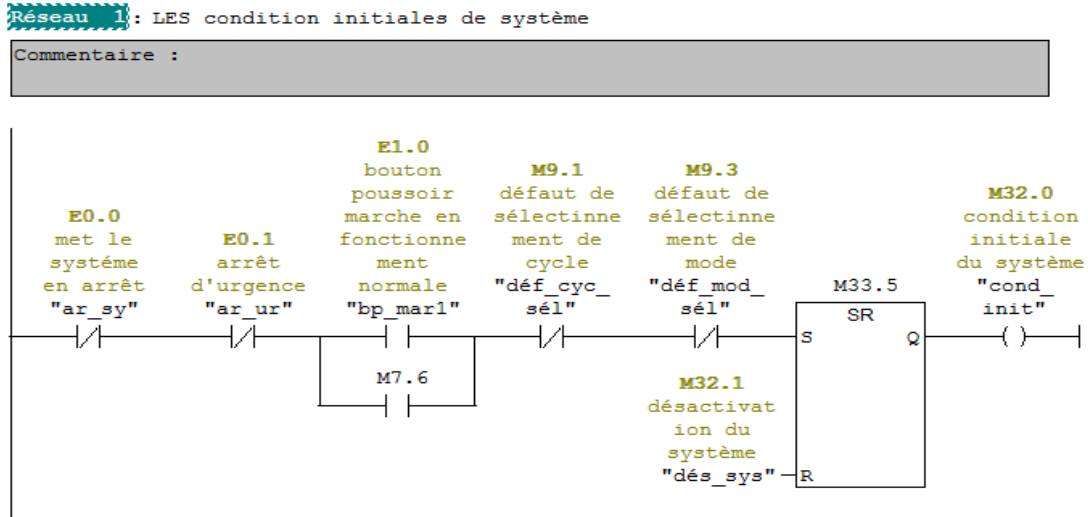
Le logiciel STEP7 met à la disposition de l'utilisateur plusieurs modes de représentation (CONT, LOG, LIST).

L'utilisateur peut donc passer d'une représentation à l'autre en sélectionnant celle adéquate pour la programmation d'un bloc.

Chapitre II : Les Automates programmables industriels (API)

- En « schéma à contact » symbolisé par (CONT)

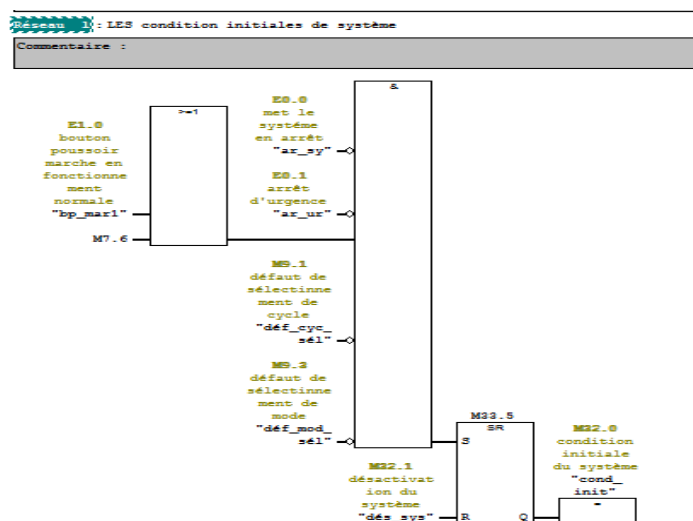
Le schéma à contact est un langage de programmation graphique, qui s'apparente aux schémas de circuit électrique. Il utilise des symboles de contacts et de bobines.



Langage contact

- En « LOGIGRAME » symbolisé par (LOG)

Ce dernier est lui aussi langage de programmation graphique qui utilise des pavés logiques (les opérateurs de l'algèbre de Boole) pour représenter les opérations logiques, les fonctions complexes, comme par exemples les fonctions mathématiques, peuvent être représentées, combinées avec les portes logiques.



Langage logigramme

Chapitre II : Les Automates programmables industriels (API)

- En « liste d'instruction » symbolisé par (LIST)

C'est un langage textuel de la machine, cette forme de représentation permet de programmer librement et de manière détaillée des opérations complexes.

Les différentes instructions correspondent, dans une large mesure, aux étapes par lesquelles la CPU traite le programme. Pour faciliter la programmation, LIST a été complété par quelques structures de langage évoluées (comme par exemple, des paramètres de blocs et accès structures aux données).

```
Réseau 1: LES condition initiales de système
```

```
Commentaire :
```

```
UN  "ar_sy"           E0.0      -- met le système en arrêt
UN  "ar_ur"           E0.1      -- arrêt d'urgence
U(
O   "bp_mar1"         E1.0      -- bouton poussoir marche en fonctionnement normale
O   M    7.6
)
UN  "déf_cyc_sél"     M9.1      -- défaut de sélectinment de cycle
UN  "déf_mod_sél"     M9.3      -- défaut de sélectinment de mode
S   M    33.5
U   "dés_sys"         M32.1     -- désactivation du système
R   M    33.5
U   M    33.5
=   "cond_init"       M32.0     -- condition initiale du système
```

Langage liste

a-2 Les opérations de STEP7

Elles regroupent les fonctions qui s'exécutent dans tous les blocs S7 programmes. On distingue :

1) Opérations combinatoires

Combinaison ET(U) : vérifie si les différentes conditions sont remplies simultanément par l'interrogation.

Combinaison OU (O) : vérifie si ou moins une condition parmi plusieurs est remplie.

2) Opération de mémorisation

Ces dernières affectent un état de signal au niveau de l'opérande concerné en fonction du résultat logique (RLG). Ces opérations sont :

S : Mise à 1.

R : Remise à 0.

3) Opération de changement et de transfert

Elles permettent :

- D'échanger des informations entre les différents types d'opérandes.

- De préparer des valeurs de comptage et de temporisation en vue de leur traitement.

- De charger des valeurs constantes nécessaires au traitement de programme.

4) Opération de temporisation

Elles permettent de réaliser et de surveiller par programme des actions faisant intervenir le facteur temps, parmi ces opérations nous avons :

SE : Temporisation retard à la montée,

SA : Temporisation retard à la retombée.

5) Opérations arithmétiques

Elles permettent la combinaison entre deux valeurs : on distingue :

1 : Addition.

2 : Soustraction.

Le système d'exploitation de la CPU contient des blocs de multiplication et de division.

6) Opération de comptage

Ces opérations sont directement réalisées par la CPU à l'aide des opérations suivantes :

ZV : Comptage.

ZR : décomptage.

7) Opération de comparaison

Ces opérations permettent de comparer entre deux valeurs. Parmi ces opérations nous avons :

=I : Comparaisons d'égalité.

> : Comparaison de supériorité.

V-8-2 Automate Schneider

V-8-2-1 Télémécanique

a-Télémécanique Micro

Jusqu'à 256 E/S peuvent être configurées avec des modules d'E/S de format pleins ou demis.

-Modèles analogiques de 16 bits multi-gammes, modules de relais de sécurité intégrés, communication multi-protocole et carte modem à l'aide de la technologie PCMCIA et la mémoire extensible.

-Mémoire jusqu'à 128 Kmots.

-Fonctions: Comptage/ positionnement, analogique/régulation, fonctions maths.

-Applicatif multitâches avec fonction sur événements.

-Protection totale du savoir-faire utilisateur, des données, des périphéries automate.

b- Télémécanique Nano

Système compacte et facile à utiliser entièrement, compatible avec la gamme télémécanique de dispositif de commande, de détecteurs, de boutons poussoirs, etc.

Les instructions incluent : les compteurs et temporisateurs ascendant/descendant, les registres, les fonctions arithmétiques, la fonction de sortie ou à impulsion, la fonction de comparaison et les contrôleurs.

V-8-2-2 Logiciel Schneider- Télémécanique PL7PRO

La conception et la mise en œuvre des applications pour automates Micro se réalisent à l'aide des logiciels PL7. Le logiciel PL7 propose 4 langages de programmation : langage à contacts, liste d'instructions, littéral structuré et Grafset. Ces langages peuvent être mixés au sein d'une même application. Une section de programme peut être écrite en langage à

Chapitre II : Les Automates programmables industriels (API)

contacts, une autre en littéral... Ces langages mettent en œuvre : des blocs fonction prédéfinis (temporisation, compteurs...), des fonctions métiers (analogique, communication, comptage...), des fonctions spécifiques (gestion du temps, chaîne de caractères.....).

Les objets du langage se symbolisent à l'aide de l'éditeur de variables ou en ligne dans les éditeurs de programme.

Langage à contacts : le langage à contacts (LD) est un langage graphique. Il permet la transcription de schémas à relais, il est adapté au traitement combinatoire. Il offre les symboles graphiques de base : contacts, bobines, blocs.

Langage liste d'instructions : Le langage liste d'instructions (IL) est un langage « machine » Booléen qui permet l'écriture de traitements logiques et numériques.

Langage littéral structuré : Le langage littéral structuré (ST) est un langage de type « informatique » permettant l'écriture structurée de traitements logiques et numériques.

Langage Grafcet : Le langage Grafcet permet de représenter graphiquement et de façon structurée le fonctionnement d'un automatisme séquentiel.

Le logiciel PL7 propose deux types de structure :

Mono tâche : c'est la structure simplifiée proposée par défaut, où une seule tâche maître composée d'un programme, constitué de plusieurs sections et de sous-programmes, est exécutée.

Multitâche : cette structure, mieux adaptée pour des applications temps réels performantes, se compose d'une tâche maître, d'une tâche rapide et de traitements prioritaires.

V-8-3 Automate Omron

a-Automate Omron Micro API compacts jusqu'à 320 E/S

Gamme d'API compacts, avec alimentation et entrées/sorties intégrées. Equipés d'un large jeu d'instructions, ces API sont parfaites pour le contrôle des petites machines. Avec leurs fonctions de compteur grande vitesse et leurs sorties d'impulsions intégrées, les applications de contrôle de vitesse ou de positionnement deviennent très faciles à gérer.

b-Automate Omron modulaires jusqu'à 2500 E/S

Avec une large gamme, de cartes réseau et de modules de contrôle d'axes, c'est le système modulaire le plus abouti. En plus d'excellentes performances de contrôle, son architecture de communication transparente en fait un point d'accès idéal pour n'importe quel périphérique intelligent en réseau, car il prend en charge Ethernet.

V-8-3-1 Logiciel Omron CX-One

Une programmation API commune à tous les API Omron, la conversion et la réutilisation de programmes API d'un modèle d'API à un autre est facile, de même que la réutilisation complète de programmes créés avec des logiciels plus anciens. Des fonctions similaires sont aussi offertes avec les logiciels de programmation pour IHM. Des IHM monochromes aux modèles les plus avancés, tous peuvent être programmés, et les étiquettes créées par l'utilisateur dans CX-One peuvent être partagées avec les autres logiciels.

La suite CX-One permet la programmation et la configuration d'une large gamme de contrôleurs d'axes, du contrôle point à point, au contrôle d'axes sur API, et à la programmation en langage basic.

Omron propose l'une des plus larges gammes pour la régulation, des régulateurs de température autonomes aux cartes de régulation API multiboucles. CX-One permet la configuration graphique, l'installation et la maintenance de réseaux d'information tels que Ethernet et Controller-Link, des réseaux de terrain PROFIBUS DP et DeviceNet, et des réseaux série comme Modbus, Compoway/F ou à protocole défini par l'utilisateur.

V-9 Choix d'un automate

Le choix des API revient à considérer certains critères importants tels que [5]:

- Les compétences/expériences de l'équipe d'automaticiens en mise en œuvre et en programmation de la gamme d'automate.
- Les capacités de traitement du processeur (vitesse, données, opérations, temps réel...).
- Le nombre et la nature des entrées/sorties.
- La fiabilité et la robustesse.
- Communication avec d'autre système.
- Le type du processeur, la taille de la mémoire, la vitesse de traitement et les fonctions spéciales offertes par le processeur.

V-10 Avantage d'API

- Il présente une grande souplesse de mise en œuvre. Sa compressibilité conduit à une économie de place et une fiabilité accrue.
- La maintenance préventive et curative des systèmes est simplifiée.
- Il facilite la documentation des applications, donc la maintenance.
- Il est bien adapté à la surveillance en ligne du fonctionnement de la loi de commande, visualisation des E/S, des traitements logiques.
- Les API permettent d'ajuster la disponibilité du système aux besoins.
- Un personnel de qualification moyenne peut manipuler facilement.
- Il offre une sécurité totale en cas d'incendie, anomalie sur le processus... [5]

V-11 Inconvénients d'API

- L'API ne supprime pas tout le reliage : il reste le câblage du circuit de puissance ;
- Les modules de recherche sont plus coûteux.
- Sa vitesse peut s'avérer insuffisante dans quelques cas.
- Le choix d'une machine nécessite une étude préalable approfondie de l'ensemble des matériels et donc la création d'une compétence API au niveau industriel.
- Le déroulement cyclique du programme peut s'avérer un facteur de complexité et limite les possibilités d'organisation des tâches [5].

VI Conclusion

Dans ce chapitre, nous avons présenté quelques firmes des automates programmables industriels (la famille SIEMENS, SCHNEIDER et OMRAN) ainsi que leurs logiciels de programmation.

Le logiciel de programmation STEP7 constitue le lien entre l'utilisateur et l'automate programmable S7-300 car ce dernier ne peut gérer ses fonctionnalités sans un programme approprié. Mais avant l'élaboration du programme il faut d'abord modéliser le procédé. Cela fera l'objet du prochain chapitre.

I Introduction

Le **GRAFCET** est un outil graphique de définition pour l'automatisme séquentiel, en tout ou rien. Mais il est également utilisé dans beaucoup de cas combinatoires, dans le cas où il y a une séquence à respecter mais où l'état des capteurs suffirait pour résoudre le problème en combinatoire. Il utilise une représentation graphique. C'est un langage clair, strict mais sans ambiguïté, permettant par exemple au réalisateur de montrer au donneur d'ordre comment il a compris le cahier des charges. Langage universel, indépendant (dans un premier temps) de la réalisation pratique (peut se "câbler" par séquenceurs, être programmé sur automate).

II Le GRAFCET

II-1 Définition

Le GRAFCET (élaboré en 1977) est l'abréviation de l'expression « **G**raphe **F**onctionnel de **C**ommande des **E**tapes/**T**ransitions ». Il permet de décrire tous les comportements attendus d'un automatisme de commande face aux événements ou aux informations issues d'un processus automatisé. En d'autres termes, c'est un modèle graphique de représentation du cahier des charges d'un automatisme logique [1].

- Les **ENTREES**, c'est-à-dire les transferts d'informations de la Partie Opérative vers la Partie Commande,
- Les **SORTIES**, transferts d'informations de la Partie Commande vers la Partie Opérative.

II-2 Niveau du GRAFCET

Le GRAFCET est réalisé selon deux niveaux de représentation, qui sont définis comme suit :

Niveau 1 : Appelé aussi GRAFCET fonctionnel. Il décrit sous forme de spécifications fonctionnelles, le comportement de la partie commande vis-à-vis de la partie opérative et du monde extérieur. Les réceptivités sont décrites sous forme littérale par des mots et non pas abréviation. A ce niveau, on ne définit pas les actionneurs ni les capteurs mais uniquement les actions à effectuer et leur enchaînement, pour permettre de comprendre l'évolution de l'automatisme.

Niveau 2 : Ce GRAFCET ajoute aux exigences fonctionnelles, les précisions indispensables aux conditions de fonctionnement, grâce aux spécifications technologiques et opérationnelles, compte tenu de la technologie de la partie commande et de la partie opérative, ainsi que la prise en compte de la technologie des actionneurs et des capteurs de l'automatisme. Il est utilisé pour la réalisation et l'exploitation des systèmes automatisés. La description des actions et des réceptivités est par abréviation.

III Les concepts de base du GRAFCET

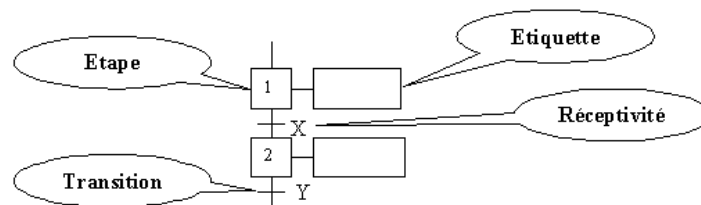


Figure III-1: Les éléments constituant le GRAFCET.

III-1 Etape

Une **étape** symbolise un état ou une partie de l'état du système automatisé. L'étape possède deux états possibles : **active** ou **inactive**.

III-2 Actions

A chaque étape est associée une **action** ou plusieurs, c'est à dire un ordre vers la partie opérative ou vers d'autres GRAFCET. Mais on peut rencontrer aussi une même action associée à plusieurs étapes ou une étape **vide** (sans action).

III-3 Transition

Une transition indique la possibilité d'évolution qui existe entre deux étapes et donc la succession de deux activités dans la partie opérative. Lors de son franchissement, elle va permettre l'évolution du système. A chaque transition est associée une condition logique appelée réceptivité qui exprime la condition nécessaire pour passer d'une étape à une autre.

IV Règles d'évolution du GRAFCET

IV-1 Condition initiale

A l'instant initial, seules les étapes initiales sont actives.

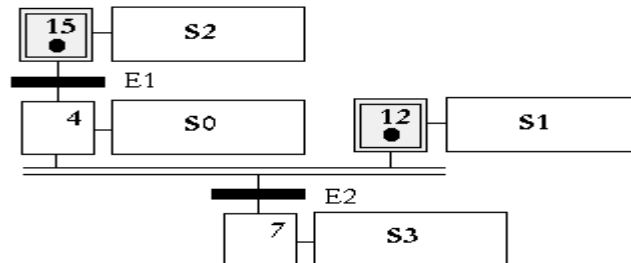


Figure III-2: Condition initiale.

IV-2 Franchissement d'une transition.

Pour qu'une transition soit validée, il faut que toutes ses étapes amont (immédiatement précédentes reliées à cette transition) soient actives. Le franchissement d'une transition se produit lorsque la transition est validée, **ET seulement si** la réceptivité associée est **vraie**.

IV-3 Evolution des étapes actives

Le franchissement d'une transition entraîne obligatoirement l'activation de toutes les étapes immédiatement suivantes et la désactivation de toutes les étapes immédiatement précédentes.

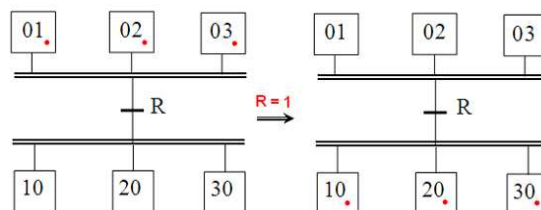


Figure III-3: Franchissement d'une transition.

IV-4 Franchissement simultané

Toutes les transitions simultanément franchissables à un instant donné sont simultanément franchies.

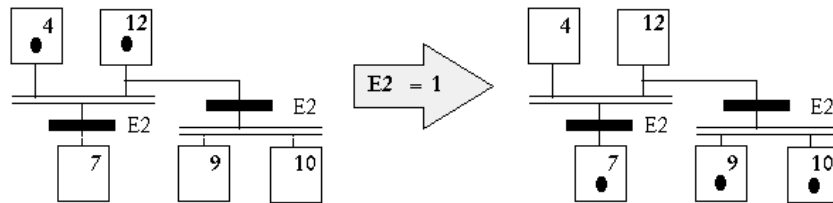


Figure III-4: Franchissements simultanés.

IV-5 Conflit d'activation

Si une étape doit être simultanément désactivée par le franchissement d'une transition aval, et activée par le franchissement d'une transition amont, alors elle reste active. On évite ainsi des commandes transitoires (néfastes à la partie opérative).

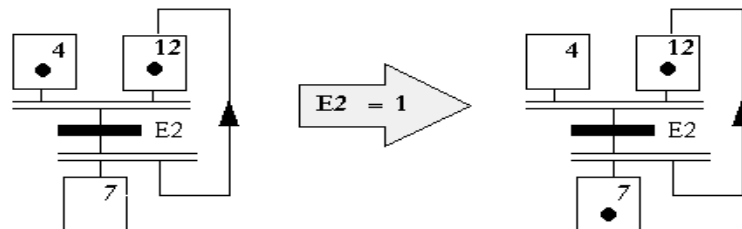
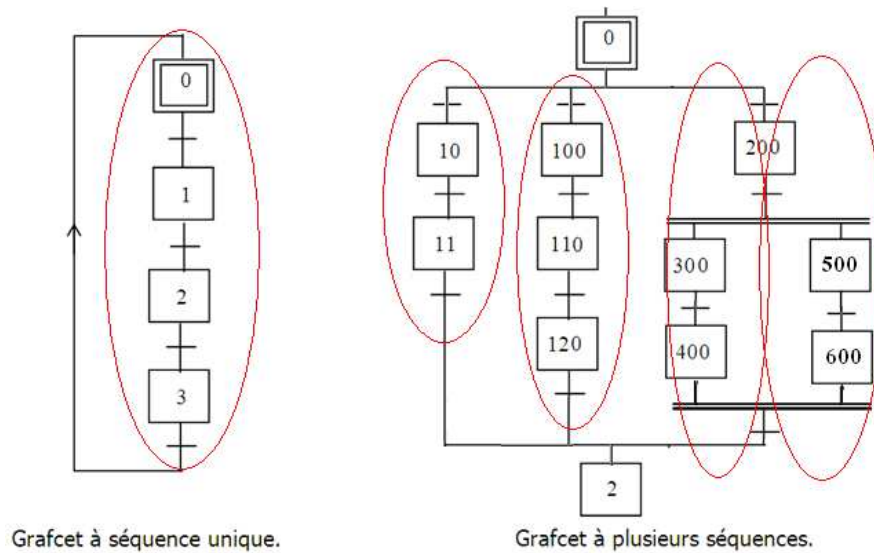


Figure III-5: Activation et désactivation simultanées d'étape (étape 12).

V - Les structures de base

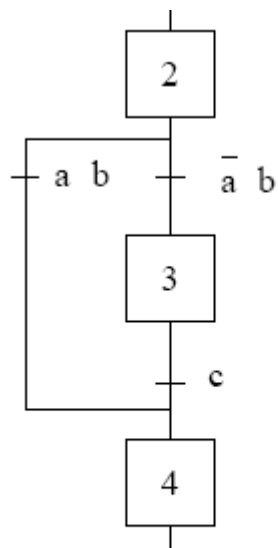
V-1 Notion de Séquence :

Une séquence, dans un GRAFCET, est une suite d'étapes à exécuter l'une après l'autre. Autrement dit chaque étape ne possède qu'une seule transition AVAL et une seule transition AMONT.

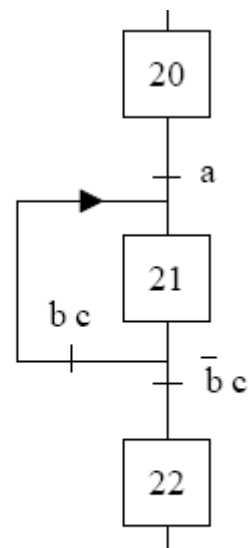


V-2 Saut d'étapes et reprise de séquence

Le saut d'étapes permet de sauter une ou plusieurs étapes lorsque les actions associées sont inutiles à réaliser, La reprise de séquence (ou boucle) permet de reprendre, une ou plusieurs fois, une séquence tant qu'une condition n'est pas obtenue.



Saut d'étape



Reprise de séquence

V-3 Aiguillage entre deux ou plusieurs séquences (Divergence en OU)

On dit qu'il y a **Aiguillage** ou **divergence en OU** lorsque le grafcet se décompose en deux ou plusieurs séquences selon un choix conditionnel. Comme la divergence en OU on rencontre aussi la convergence en OU. On dit qu'il y a convergence en OU, lorsque deux ou plusieurs séquences du grafcet converge vers une seule séquence.

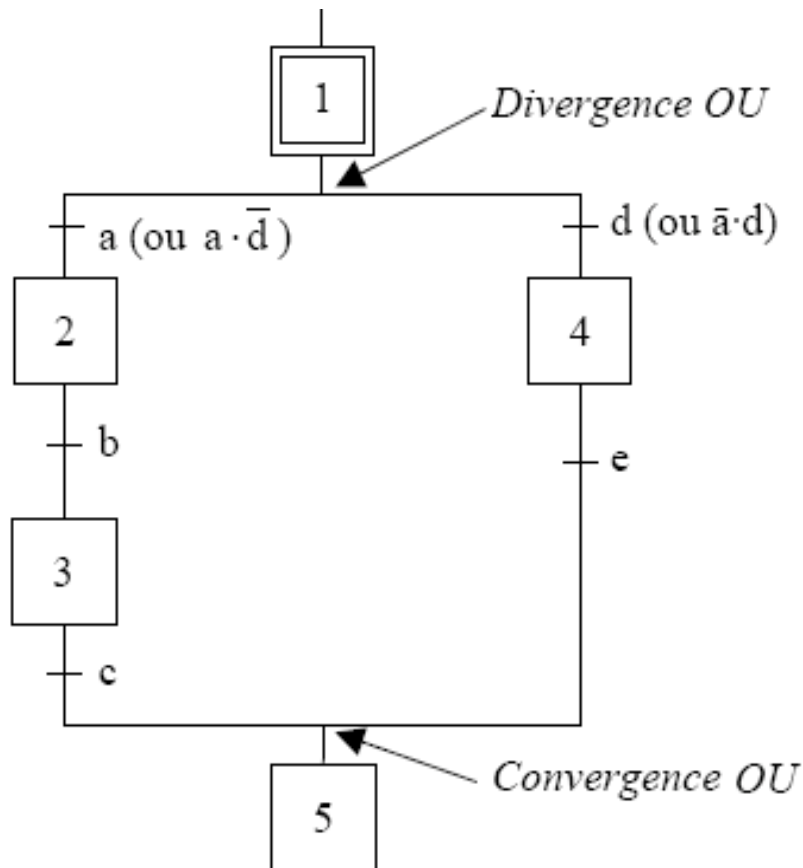


Figure III-6: Divergence et convergence en OU

Si les deux conditions a et d sont mises à 1 simultanément, les étapes 2 et 4 vont devenir actives simultanément, situation non voulue par le concepteur. Donc elles doivent être des conditions **exclusives**

V-4 Parallélisme entre deux ou plusieurs séquences (ou séquences simultanées ou divergence convergence en ET) :

Au contraire de l'aiguillage on ne peut se dérouler qu'une seule activité à la fois, On dit qu'on se trouve en présence d'un parallélisme structurel, si plusieurs activités indépendantes

Chapitre III : Modélisation du système à l'aide de GRAFCET

pouvant se dérouler en parallèle. Le début d'une divergence en ET est la fin d'une convergence en ET d'un parallélisme structurel sont représentés par deux traits parallèles.

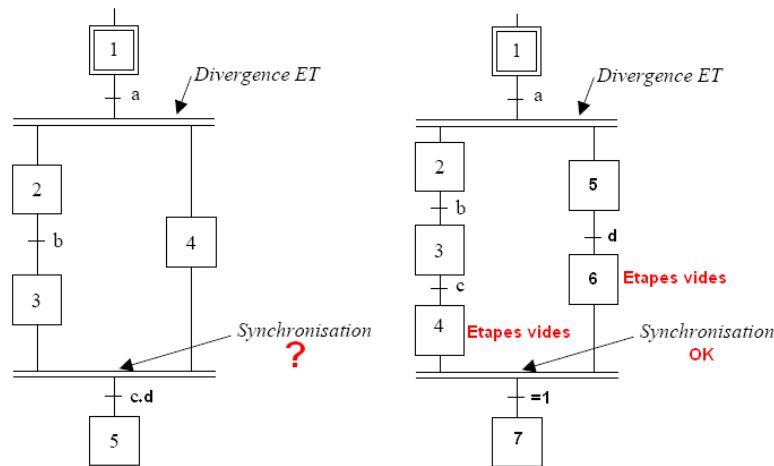


Figure III-7: Divergence et convergence en ET

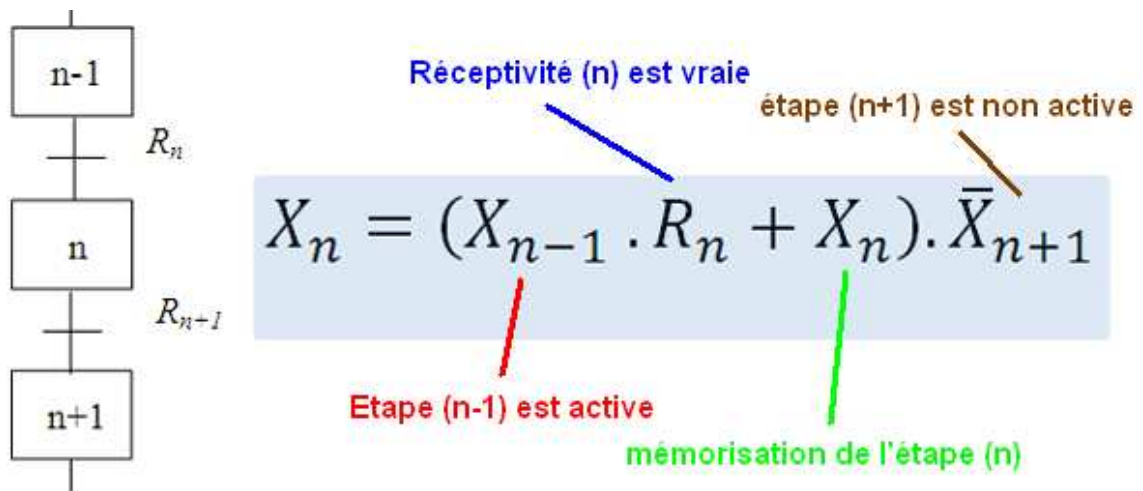
La synchronisation permet d'attendre la fin de plusieurs activités se déroulant en parallèle.

VI Mise en équation d'un GRAFCET :

VI-1 Règle générale :

Pour qu'une étape soit activée il faut que :

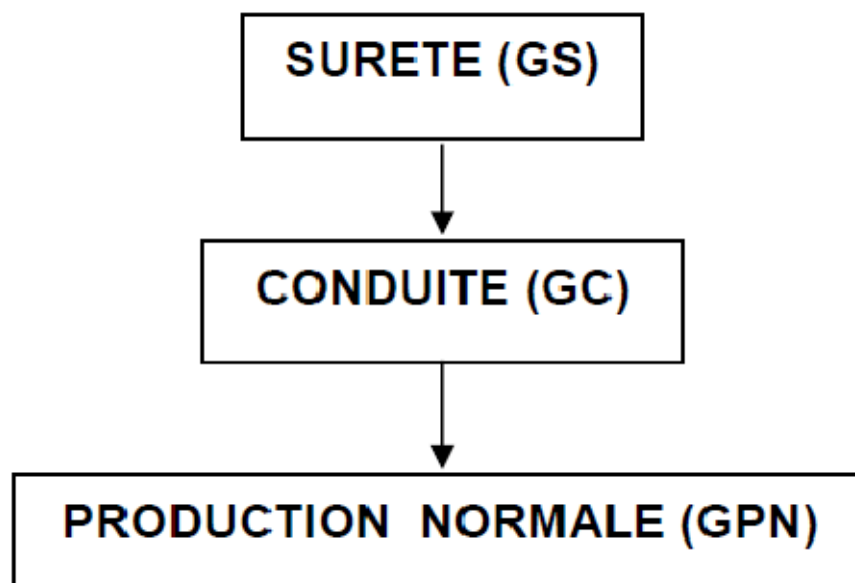
- L'étape immédiatement précédente soit activée.
- La réceptivité immédiatement précédente soit vraie.
- L'étape immédiatement suivante soit non active.
- Après activation l'étape mémorise son état.



Equation d'activation de l'étape de rang n

VII Hiérarchie des GRAFCET

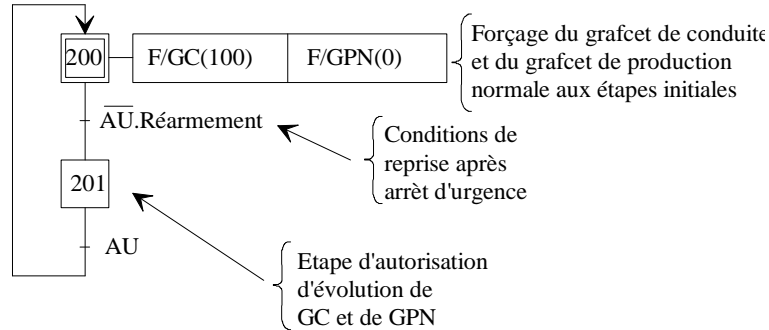
Les contraintes de SURETE et de CONDUITE induisent une priorité et une hiérarchie fonctionnelle à 3 niveaux entre les Grafcets, facilitant ainsi la compréhension et la mise en œuvre [1].



Chapitre III : Modélisation du système à l'aide de GRAFCET

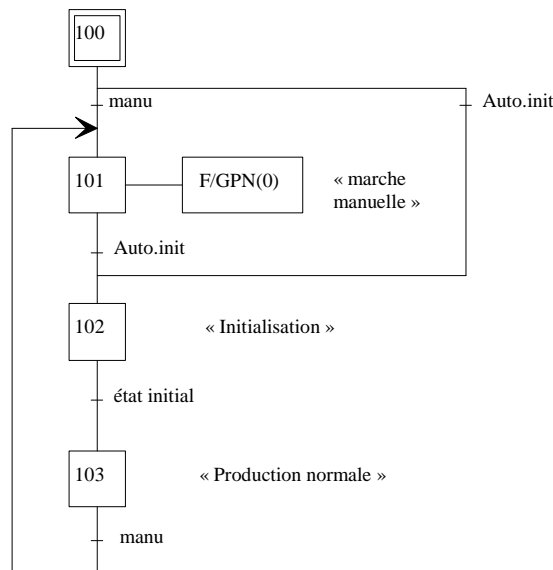
- GRAFCET de sûreté (GS)

Ce Grafcet gère les procédures de sécurité. Il est réceptif à une consigne d'arrêt d'urgence, et ce, depuis tous les états.



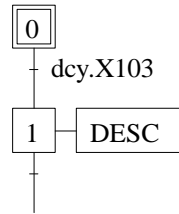
- GRAFCET de conduite (GC) ou grafcet des modes de marche et d'arrêt (GMMA)

Ce Grafcet gère les modes de marche et d'arrêt. Il associe une étape à chacun des modes de marche prévus.



- GRAFCET de production normale

Ce Grafcet gère l'évolution du cycle de production normale.



VIII Le GEMMA

Le GEMMA est le second outil méthode créé par l'AF CET (Association Française pour la Cybernétique Économique et Technique). Le mot GEMMA est l'acronyme de **Guide d'Étude des Modes de Marches et d'Arrêts**.

VIII-1 Pourquoi le GEMMA

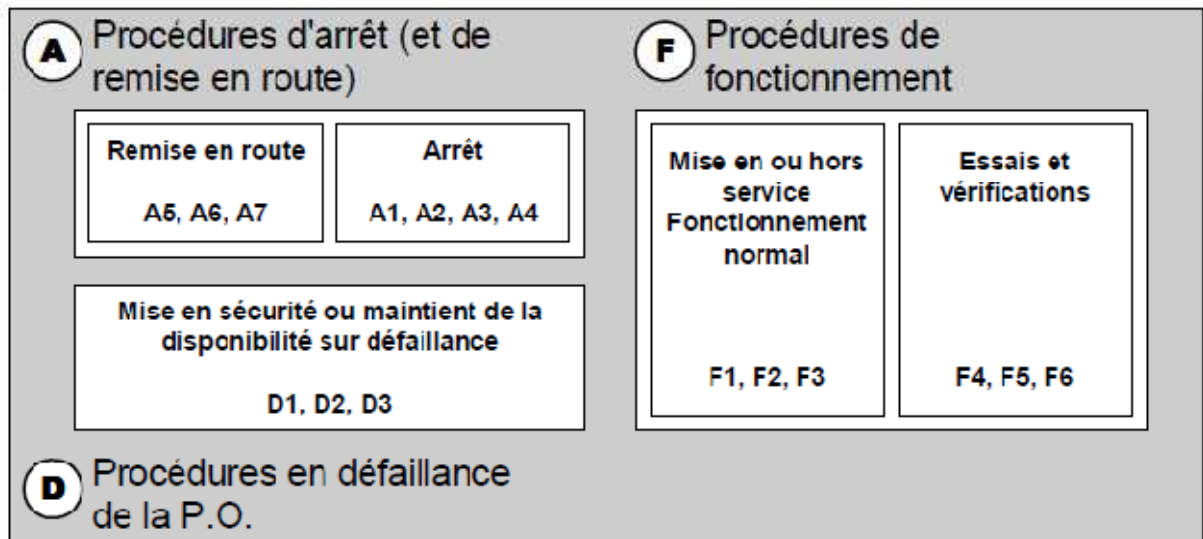
Le GEMMA est un guide graphique permettant de sélectionner et de décrire simplement les différents états de marches et d'arrêts, ainsi que les possibilités d'évoluer d'un état à un autre. Il a été créé parce qu'il y avait un grand besoin d'avoir un vocabulaire commun et précis. Le GEMMA permet d'avoir une approche guidée de l'analyse des modes de marches et d'arrêts.

Le GEMMA permet le recensement et la description des différents états du système automatisé, de la mise en route à la production normale. Il précise les procédures à mettre en œuvre après analyse d'une anomalie ou un défaut de fonctionnement.

VIII-2 les trois familles de procédures

Ces trois familles sont :

- Famille F – procédures de fonctionnement.
- Famille A – procédures d'arrêt.
- Famille D – procédures de défaillances.



-La famille « F »

La famille «F» des procédures de fonctionnement concerne tous les modes qui permettent ou aident à obtenir la valeur ajoutée. On ne produit pas dans tous les modes de cette famille.

Certains modes de la famille «F» sont nécessaires pour préparer l'automatisme avant de commencer à produire. Sans ces modes, impossible de produire correctement. D'autres modes sont nécessaires pour faire des tests, des vérifications, de la maintenance. Sans ces modes, la valeur ajoutée risque de se dégrader avec le temps. Une machine mal entretenue risque de tomber en panne ou de produire avec une qualité réduite.

- La famille « A »

La famille «A» des procédures d'arrêt concerne tous les modes ou états qui conduisent à un arrêt de l'automatisme pour des raisons extérieures. Ce peut être un arrêt volontaire à partir du panneau de commande parce que la période de travail est terminée (pause, fin de la journée, fin de semaine, vacances). Ce peut être un arrêt en fin de lot. Ce peut aussi être un arrêt par manque de matière première. C'est donc un «**arrêt normal**».

Chapitre III : Modélisation du système à l'aide de GRAFCET

Cette famille correspond aussi à certains modes de remise en route de l'automatisme. L'automatisme peut exiger d'être nettoyé suite à un incident. Il exige aussi, à l'occasion, une remise en condition initiale de tous ses actionneurs.

La famille « D »

La famille «D» des procédures de défaillance concerne tous les modes ou états conduisant à ou traduisant un arrêt du système pour des raisons intérieures. Il est très rare qu'une machine ne tombe pas en panne pendant sa vie.

Il faut donc préparer l'automatisme à ces défaillances souvent imprévisibles par l'ajout de capteurs et de boutons d'arrêt d'urgence. L'arrêt doit être immédiat. Si nécessaire, des procédures limitant les conséquences doivent être immédiatement exécutées pour protéger le personnel et le matériel.

IX GRAFCET du système (niveau 2)

A l'état initial, le premier et le quatrième bain sont à des températures quelconques, pour cela il faut mettre en marche la pompe pour ajuster les températures souhaitées (80°C) et mettre en marche le four de séchage pour atteindre la température de 80°C.

Les capteurs (Cp1, Cp2, Cp3, Cp4, Cp5, Cp6, Cp7) sont des captures de position qui arrêtent le convoyeur (M2=0) et mettent la chaîne en marche (M1=1).

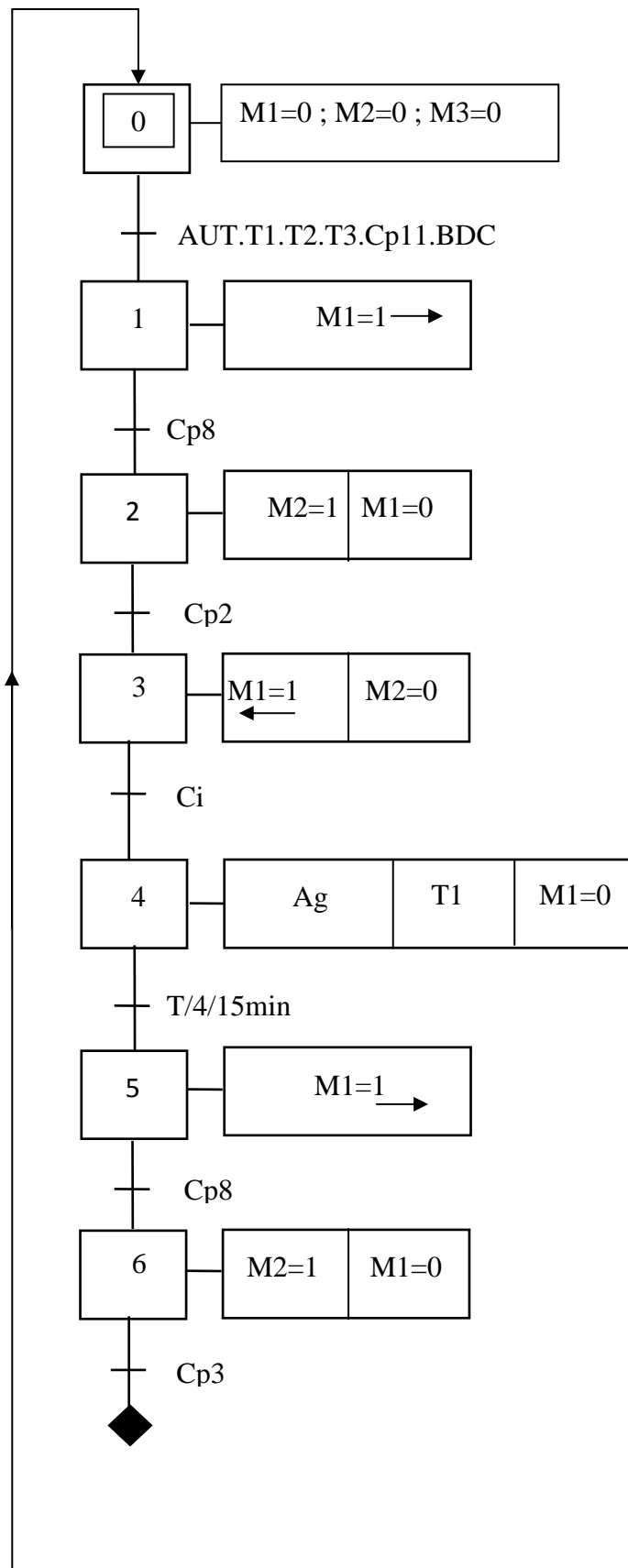
Cp8 est un capteur de position qui permet à la chaîne de s'arrêter quand elle atteint la montée souhaitée (M1=0) et mettre en marche le convoyeur (M2=1).

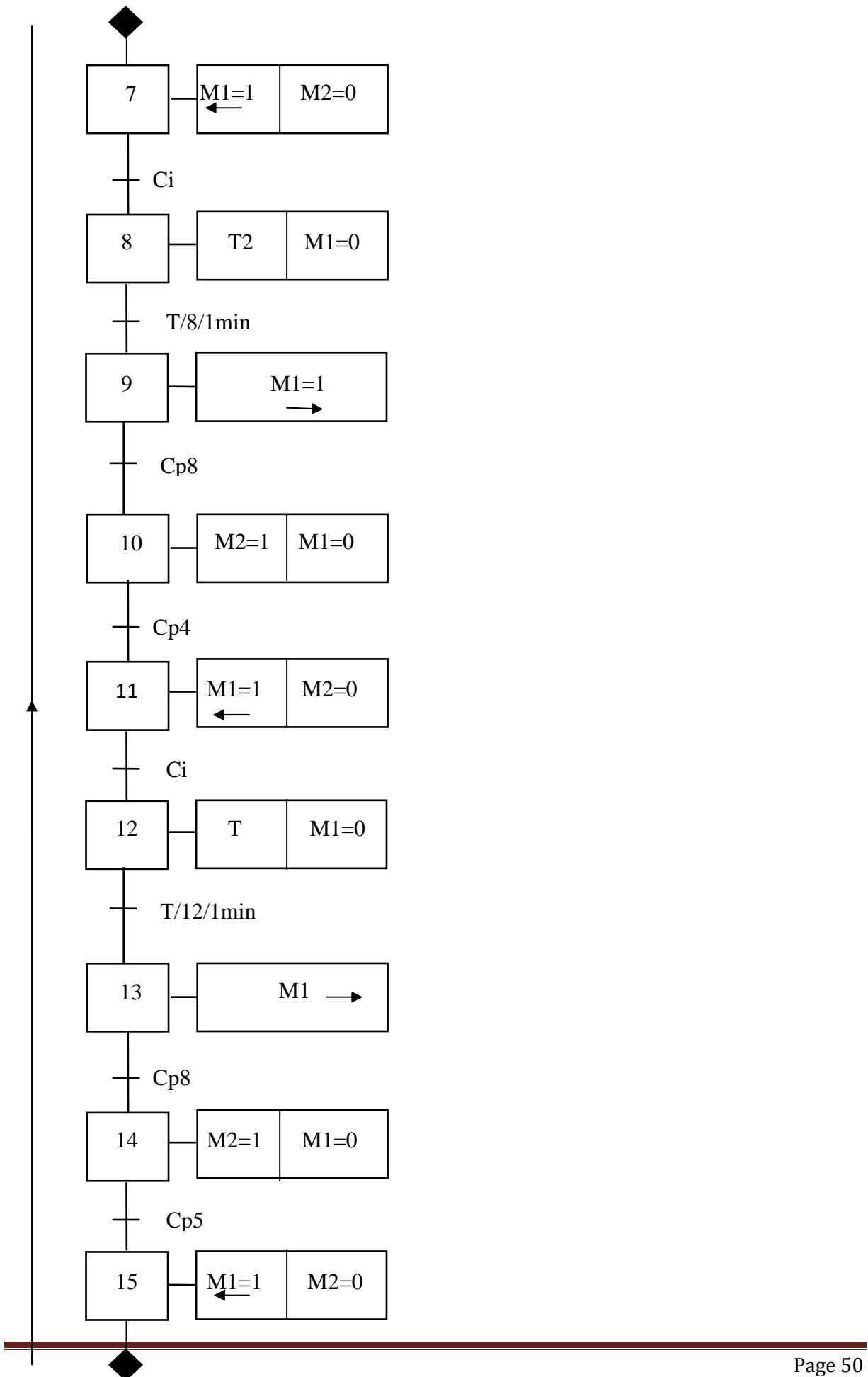
Cp9 est le capteur de position de la descente de la chaîne (M1=0).

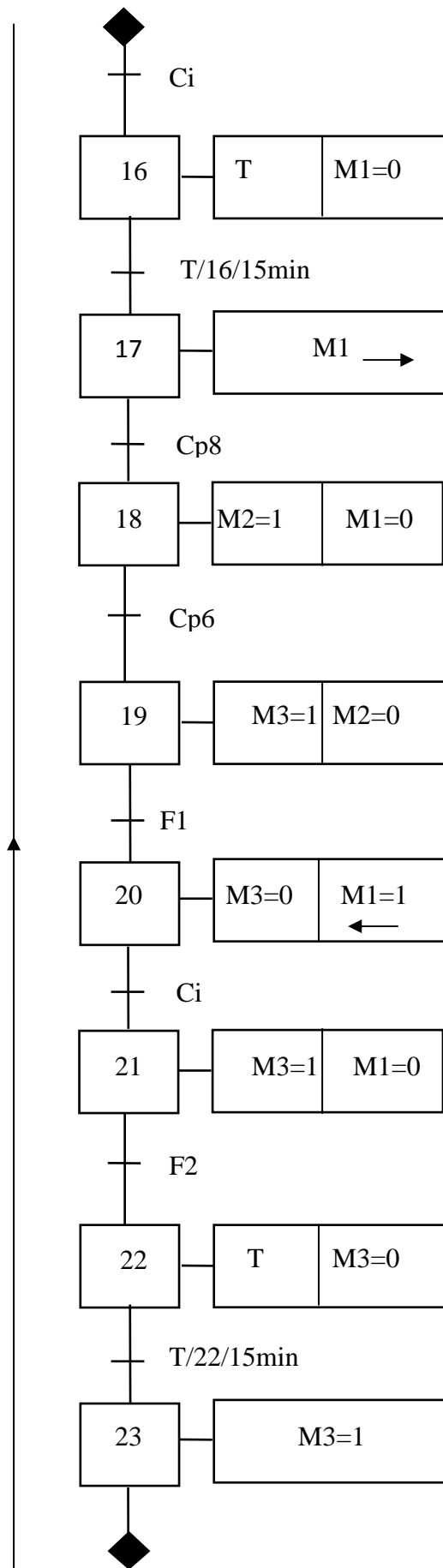
Les capteurs inductifs (Ci) mettent en arrêt la chaîne (M1=0) lorsque la corbeille des pièces est immergée dans les baignoires et lorsqu'elle est en place dans le four.

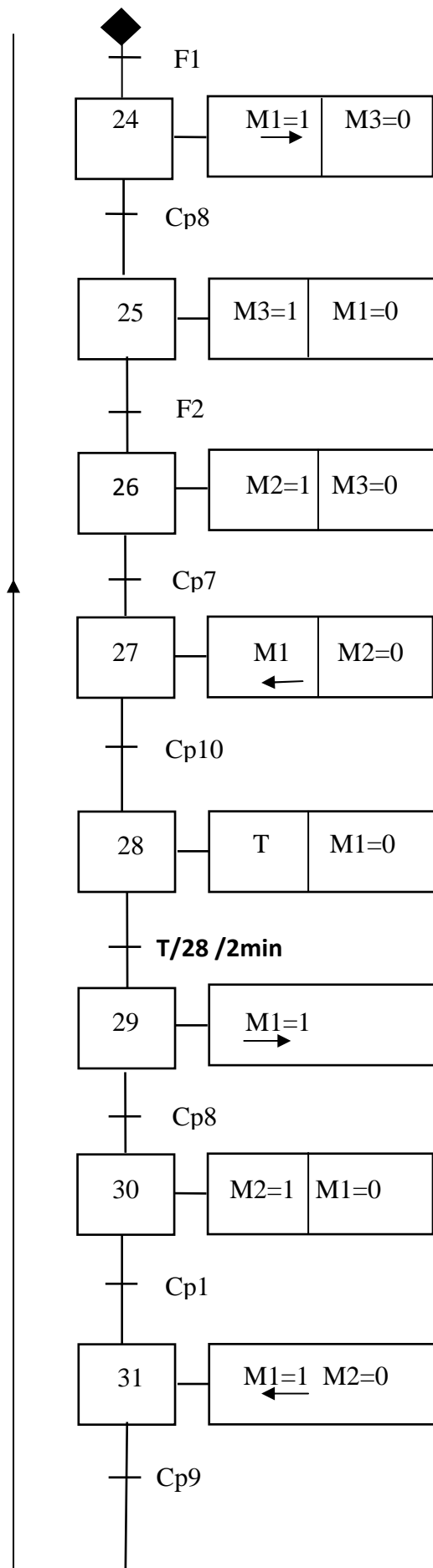
Les capteurs de positions (Cp10, Cp11) indiquent la position de la corbeille des pièces (poste 1 et 7) pour le chargement et le déchargement.

Avant démarrage du système la pompe de premier bain doit être mise en marche et cela pour toute la durée de cycle du système.









X Conclusion

Le GRAFCET est un diagramme fonctionnel dont le but est de décrire graphiquement les différents comportements d'un automatisme séquentiel ; il est l'outil adéquat pour la modélisation des systèmes industriels séquentiels pour sa souplesse et sa facilité d'utilisation. IL nous permettra au chapitre suivant d'aborder la programmation à l'aide du STEP7.

I Introduction

Après avoir étudié quelques automates programmables dans le chapitre précédent on va valider le choix de l'automate SIEMENS S7-300 pour notre système.

II Constitution de l'automate S7-300[6]

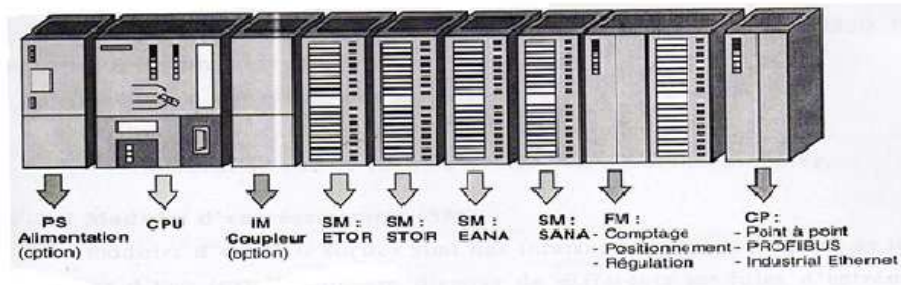


Figure IV-1: Disposition des modules de S7-300

II-1 Module d'alimentation (PS)

Occupe l'emplacement 1, le module d'alimentation (PS) convertit la tension secteur (120V/220V) en tension de service 24V DC pour alimenter le S7-300. Il délivre un courant de sortie assigné de 2,5A et 10A.

II-2 L'unité centrale (CPU)

Occupe l'emplacement 2, La CPU est le cerveau de l'automate, elle lit les états des signaux d'entrées, exécute le programme de l'utilisateur, commande les sorties. Elle permet le réglage du comportement au démarrage et le diagnostic de défaut par LED.

L'utilisateur a le choix parmi plusieurs CPU aux performances étagées. Le module de CPU est l'unité dans lequel le programme sera stocké.

II-3 Modules de couplage (IM)

Le module de couplage occupe l'emplacement 3, les coupleurs permettent de disposer d'une configuration à plusieurs châssis.

II-4 Module de signaux (SM)

Les modules de signaux établissent la liaison entre la CPU du S7-300 et le processus commandé.

On dispose de différents modules de signaux :

a) Modules d'entrées TOR

Un module d'entrée doit permettre à l'unité centrale de l'automate, d'effectuer une lecture de l'état logique des capteurs qui lui sont associés (module 4, 8, 16 ou 32 entrées).

A chaque entrée correspond une voie qui traite le signal électrique pour élaborer une information binaire. Les modules d'entrées TOR permettent de raccorder à l'automate différents capteurs logiques tels que les boutons poussoir, les fins de courses...etc.

b) Modules de sorties TOR

Un module de sortie permet à l'automate programmable d'agir sur les actionneurs. Il réalise la correspondance : état logique \longrightarrow signal électrique.

Les modules de sorties TOR permettent de raccorder à l'automate différents actionneurs tels que les électrovannes, les distributeurs, les relais...

c) Modules d'entrées analogiques

Les modules d'entrées analogiques convertissent les signaux analogiques issus du processus en valeurs numériques. La conversion est assurée par des convertisseurs analogiques numériques (CAN).

d) Modules de sorties analogiques

Les modules de sorties analogiques convertissent les signaux (valeur) numériques internes en signaux analogiques destinés au processus. La conversion est assurée par des convertisseurs numériques analogiques (CNA).

II-5 Module de fonction (FM)

Il a pour rôle l'exécution de tâches de traitement des signaux du processus à temps critique et nécessitant une importante capacité mémoire comme le comptage, le positionnement et la régulation.

II-6 Module de communication (CP)

Ils permettent d'établir des liaisons homme-machine ou machine-machine, ces liaisons sont effectuées à l'aide des interfaces de communication :

- Point à point
- PROFIBUS
- Industriel Ethernet

II-7 Châssis d'extension (UR)

Il est constitué d'un profilé support en aluminium et bus de fond de panier avec connecteur.

Il permet le montage et le raccordement électrique de divers module tels que les modules d'entrées/sorties et l'alimentation. Il est possible d'utiliser plusieurs racks en fonction de nombre d'entrées/sorties.

III Création d'un projet

Avant de commencer la programmation, il est nécessaire de créer un projet, dans lequel, les données et le programme utilisateur à créer seront structurés [7].

Le plus simple pour créer un nouveau projet, c'est d'avoir recours à l'assistant « **Nouveau projet** ».

Pour l'appeler, on choisit la commande «**fichier** » assistant « **Nouveau projet** ». L'assistant étant lancé, cette première fenêtre s'ouvre à l'écran. C'est la fenêtre d'introduction de l'assistant.

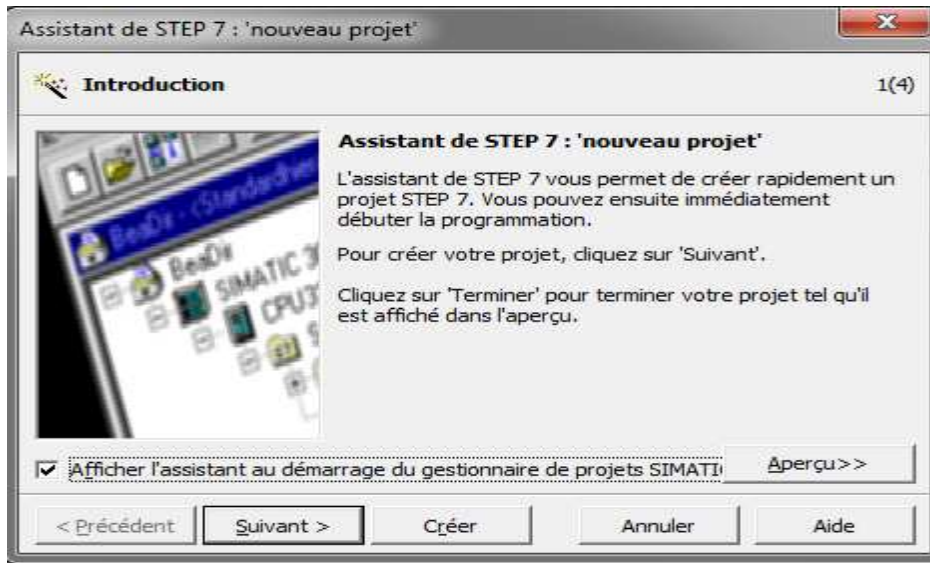


Figure IV-2: Fenêtre assistant de STEP7

Avec la commande « **suivant** » en passe à la feuille suivante dont il est possible de choisir la **CPU** et l'adresse **MPI**.

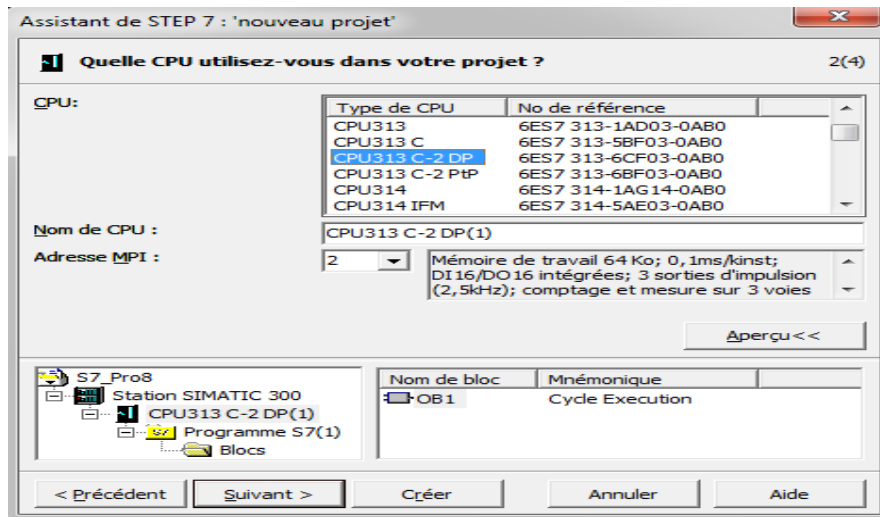


Figure IV-3: Sélection de la CPU

Il faut confirmer notre sélection par la commande « **suivant** » la fenêtre suivante va s'afficher

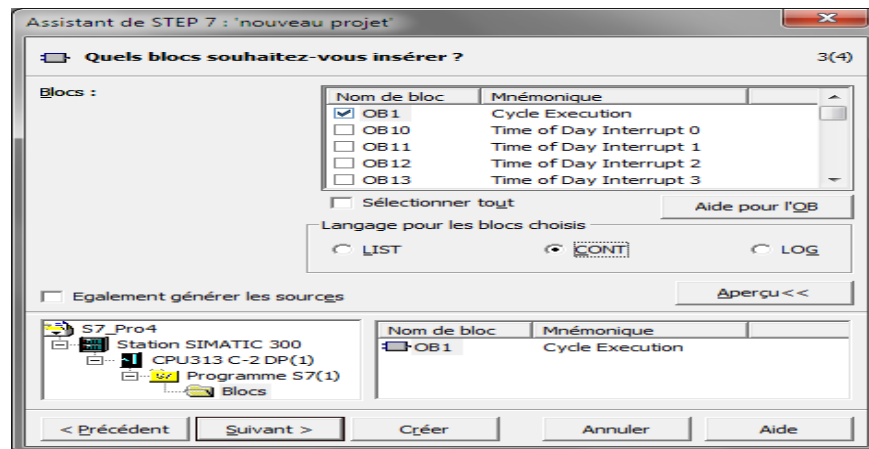


Figure IV-4: Sélection de bloc et mode de programmation.

Dans cette fenêtre il est possible de choisir les bloque qu'il faut programmer et le langage de programmation qu'il faut utiliser et confirmer la sélection avec la commande « **suivant** » une nouvelle page s'ouvre.

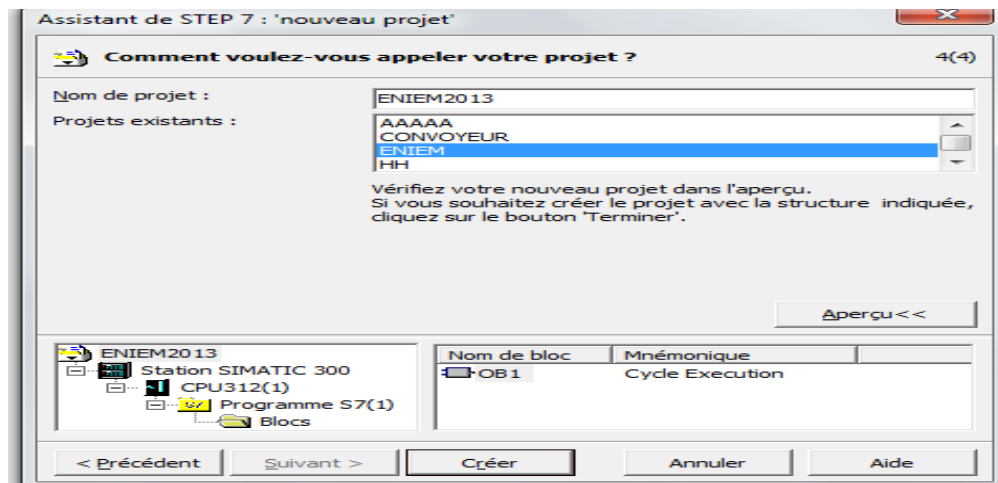


Figure IV-5: Nom du projet

Dans la zone de texte il faut donner un nom au projet, et en clique sur « créer »

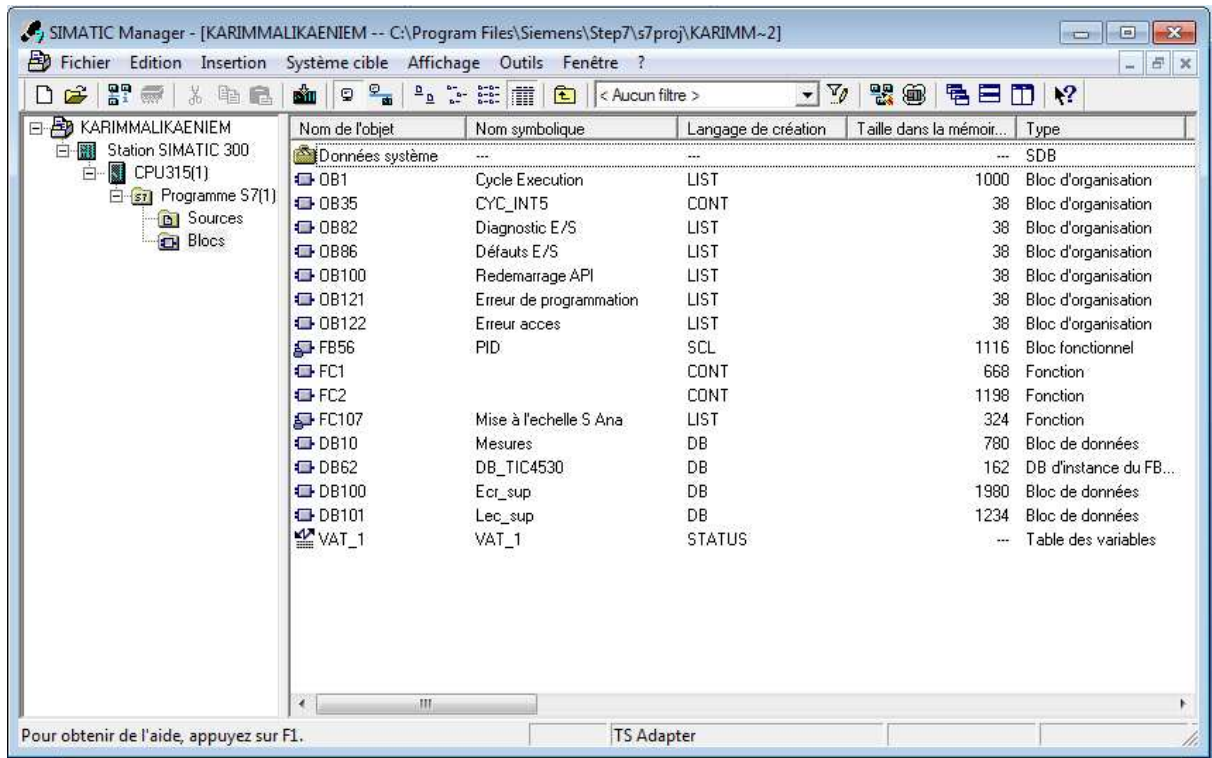


Figure IV-6: La fenêtre du projet

III-1 Configuration et paramétrage du matériel SIMATIC300

III-1-1 Configuration

Par configuration, on entend dans ce qui suit la disposition de profils support ou châssis, de module, d'appareils de la périphérie décentralisée et de cartouches interface dans une fenêtre de station.

Les profils support ou châssis sont représentés par une table de configuration, dans laquelle l'on peut enficher un nombre défini de modules tout comme dans les profils support ou châssis.

STEP7 affecte automatiquement une adresse à chaque module dans la table de configuration.

On peut modifier les adresses des modules d'une station, à condition que la CPU permette d'adressage libre.

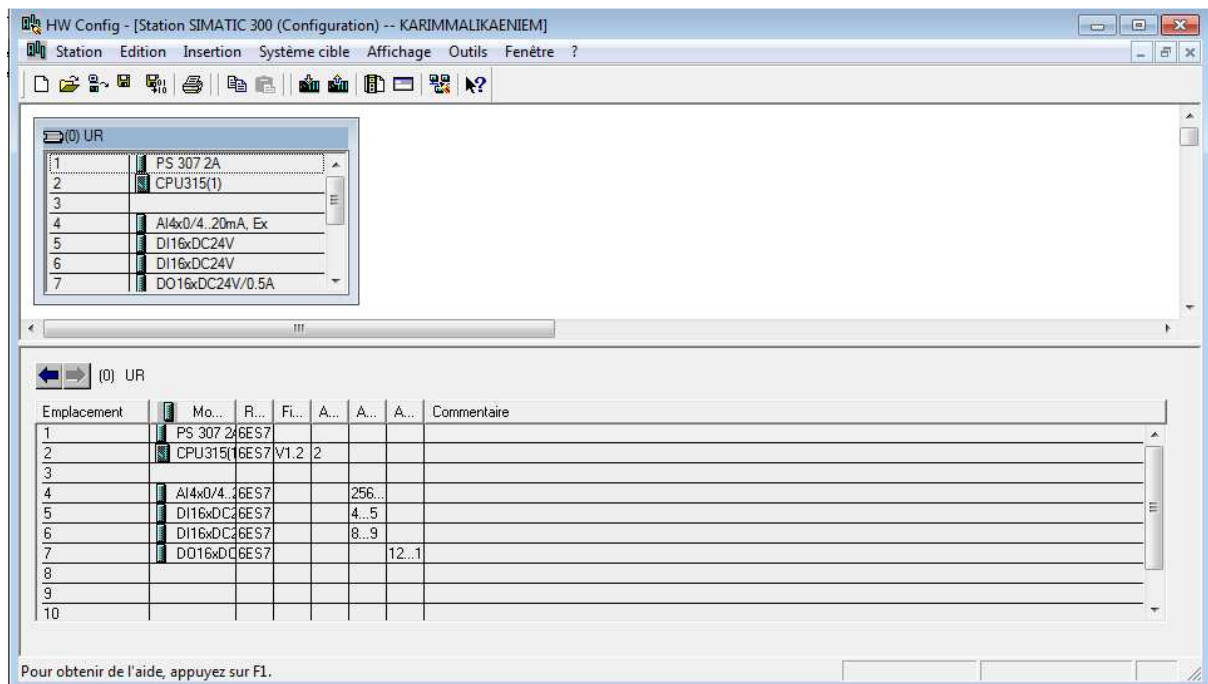


Figure IV-7: fenêtre configuration matériel

III-1-2 Paramétrage

Par « **paramétrage** », on entend dans ce qui suit :

- Le réglage des paramètres des modules paramétrable pour la configuration centralisée et pour un réseau une CPU est un module paramétrable.
- La surveillance du temps de cycle est un paramètre qu'on peut définir.
- La définition des paramètres de bus, des maître DP et des esclaves DP pour un réseau maître (PROFIBUS DP).

III-2 Structure de programme

Il existe trois types de structure de programme :

- **Programme linéaire** : toutes les opérations sont contenues dans le même bloc d'organisation (OB), qui traite cycliquement le programme.
- **Programme segmenté** : les opérations des différentes fonctions sont contenues dans des blocs isolés. L'OB1 appelle ces blocs l'un après l'autre.

- **Programme structuré** : les fonctions réutilisables sont chargées dans des différents blocs et l'OBI fait appel à ces blocs et délivre les données correspondantes. On utilise souvent le programme structuré car il simplifie l'organisation et la gestion du programme. Le test de programme peut être exécuté section par section et facilite la mise en service.

III-3 Blocs d'organisation

- **Blocs d'organisation OB** : ils constituent l'interface entre le système d'exploitation et le programme utilisateur. Ils appelés par le système d'exploitation et gèrent le traitement de programme cyclique.

- **Blocs fonctionnel FB** : c'est un bloc de code avec mémoire. Lui sert de mémoire un bloc de données d'instance, qui lui associé et dans lequel les paramètres effectifs et les données statiques du bloc fonctionnel sont stockés.

- **Bloc fonctionnel FC** : c'est un bloc de code sans mémoire dont les paramètres de sortie affichent, en fin d'exécution, les valeurs qu'ile ont calculées.

- **Blocs de données DB** : ils servent à stocker les données du programme utilisateur.

IV Le régulateur

Le régulateur est l'élément central d'une régulation. Il évalue l'erreur calculée par le comparateur, c'est à dire l'écart entre la sortie et la consigne, et en déduit à partir de celle-ci une valeur régulée ou valeur de correction à transmettre au procédé, afin de corriger la sortie..

IV-1 Les régulateurs continus

Les régulateurs continus se répartissent en trois grands types différents :

a-Le régulateur proportionnel (Type P)

Dans le cas d'un régulateur P, la variable régulée est toujours proportionnelle à l'erreur calculée.

Pour corriger au mieux l'erreur, il faut choisir un facteur de gain proportionnel aussi grand que possible. Une augmentation de ce facteur entraîne une réaction plus rapide du régulateur.

Une trop grande valeur de ce facteur va apporter un risque d'oscillation et d'instabilité du système.

b- Le régulateur intégral (type I)

Un régulateur de type intégral permet de compenser une erreur de régulation constante. Tant que l'erreur n'est pas nulle, la valeur de la variable régulée est ajustée.

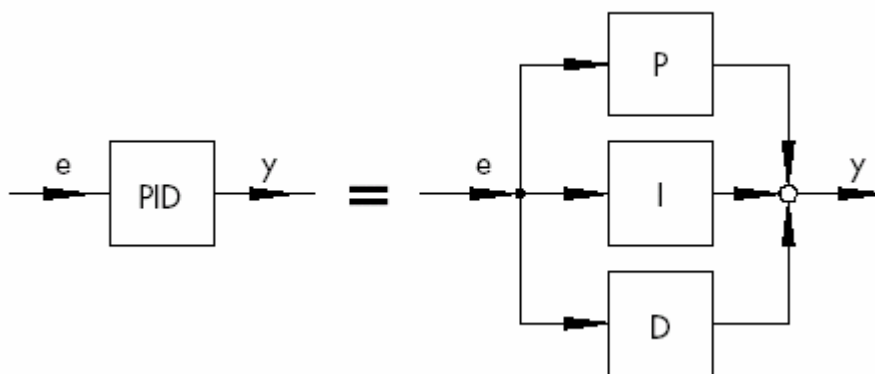
La régulation se termine lorsque la sortie a atteint la valeur de la consigne ou que la variable régulée a atteint un seuil maximal fixé par les propriétés du système (U_{max} , P_{max}).

c- Le régulateur dérivé (type D)

Le régulateur dérivé établit une valeur régulée en fonction de la vitesse de variation de l'erreur et pas en fonction de l'amplitude comme pour le régulateur P. C'est pour cette raison qu'il réagit beaucoup plus rapidement qu'un régulateur P. Même face à une petite erreur il va générer une grosse valeur régulée dès lors qu'il y a une variation d'amplitude de l'erreur. Le régulateur D sera inefficace face à une erreur résiduelle permanente, quelque soit sa valeur puisque celle-ci reste constante (pas de variation d'amplitude donc pas de réaction du régulateur). C'est pourquoi ce type de régulateur sera rarement utilisé seul dans la pratique, il est couramment associé à un régulateur de type P.

d- Le régulateur PID

Le régulateur universel PID s'obtient donc en ajoutant une composante de Type D à un régulateur PI. Comme pour un régulateur PD, la partie dérivée assure une plus grande rapidité pour atteindre la valeur de consigne en ajoutant une action anticipatrice au régulateur.



IV-2 Réalisation d'un PID sous STEP7**a-Régulation continue avec le FB 41 « CONT_C »**

Le bloc FB 41 « CONT_C » sert à régler des processus industriels à grandeurs d'entrée et de sortie continues sur les automates programmables SIMATIC S7. Le paramétrage vous permet d'activer ou de désactiver des fonctions partielles du régulateur PID et donc d'adapter ce dernier au système réglé.

b-Utilisation

On peut utiliser le régulateur comme régulateur PID de maintien autonome mais aussi comme régulateur en cascade, de mélange ou de rapport dans des régulations à plusieurs boucles. Sa méthode de travail se base sur l'algorithme PID du régulateur à échantillonnage à sortie analogique, complété le cas échéant par un étage conformateur d'impulsions assurant la formation des signaux de sortie à modulation de largeur d'impulsions pour régulations à deux ou trois échelons avec organes de réglage proportionnels.

c- Description

En plus des fonctions traitant la consigne et la mesure, le FB réalise un régulateur PID prêt à l'emploi avec sortie continue de la grandeur de réglage et possibilité d'influencer à la main la valeur de réglage.

Il propose les fonctions partielles suivantes.

Branche de consigne

La consigne est entrée en format de virgule flottante à l'entrée **SP_INT**.

Branche de mesure

La mesure peut être lue en format de périphérie ou de virgule flottante. La fonction CRP_IN convertit la valeur de périphérie PV_PER en un nombre à virgule flottante compris entre -100 et +100 % selon la formule suivante : Sortie de CPR_IN = PV_PER *100/27648

La fonction PV_NORM normalise la sortie de CRP_IN selon la formule suivante :

Sortie de PV_NORM = (sortie de CPR_IN) _ PV_FAC + PV_OFF

La valeur par défaut de PV_FAC est 1 et celle de PV_OFF est 0.

Formation du signal d'erreur

La différence entre consigne et mesure donne le signal d'erreur. Il traverse une zone morte (DEADBAND) pour supprimer la petite oscillation permanente due à la quantification de la grandeur de réglage (par exemple en cas de modulation de largeur d'impulsions avec PULSMGEN). Quand DEADB_W égale 0, la zone morte est désactivée.

Algorithme PID

L'algorithme PID travaille dans l'algorithme de position. Les actions proportionnelle, intégrale (INT) et dérivée (DIF) sont en parallèle et peuvent être activées ou désactivées séparément. Ceci permet de paramétrer des régulateurs P, PI, PD et PID, mais aussi un régulateur I pur.

d-Schéma fonctionnel

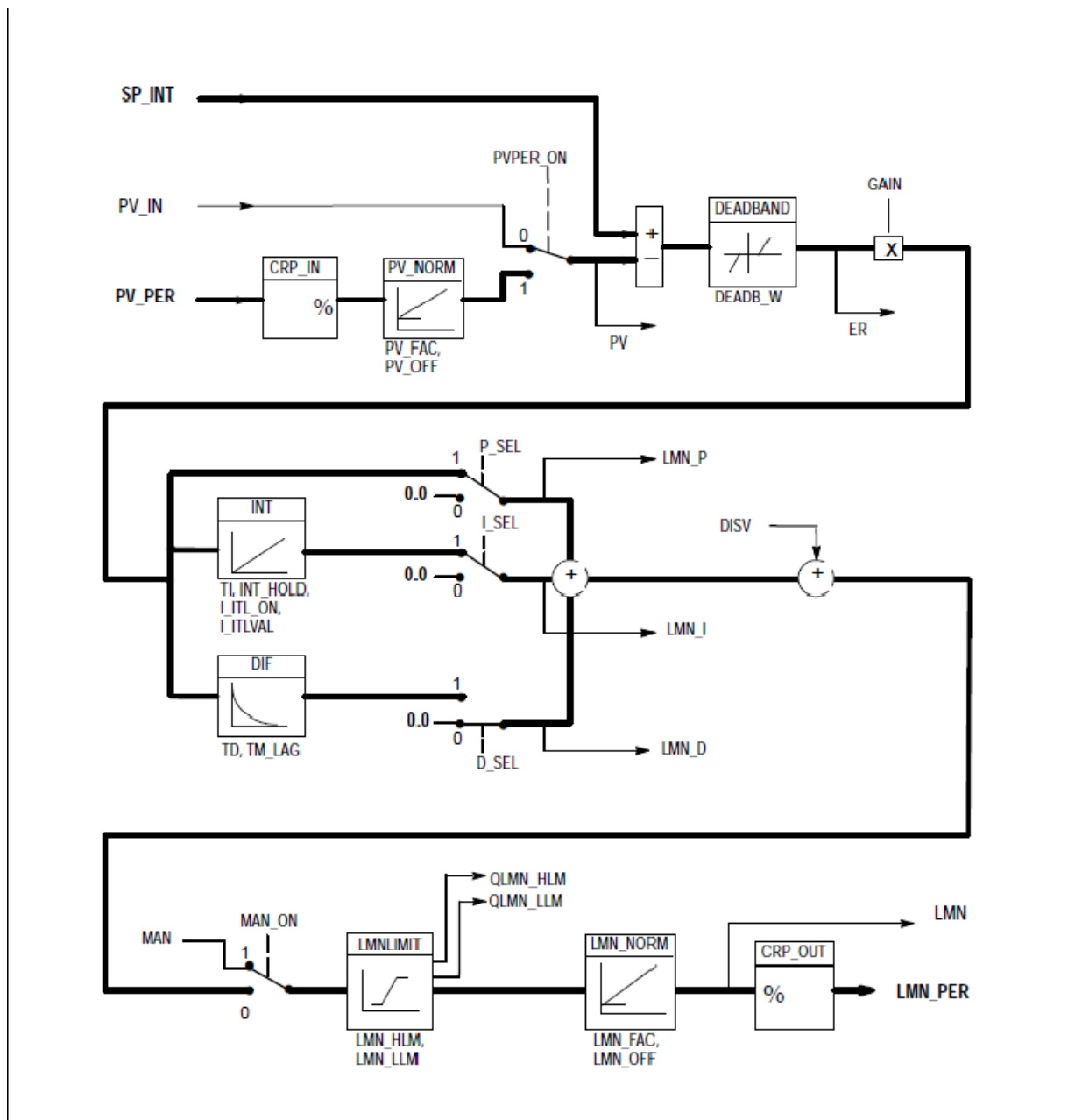


Figure IV-8: Schéma fonctionnel de CONT_C

Le Bloc FC1 : Programme de nouveau système

FC1 - <offline>

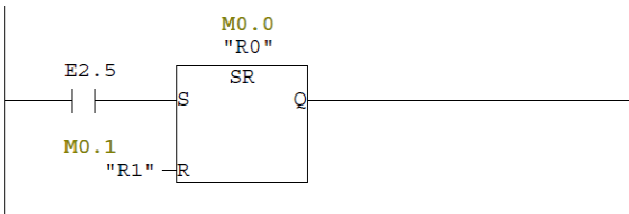
```

""
Nom :                               Famille :
Auteur :                             Version : 0.1
                                       Version de bloc : 2
Horodatage Code :                   09/06/2013 16:12:49
                                       Interface : 07/05/2013 11:46:19
Longueur (bloc/code /données locales) : 00820 00632 00000
    
```

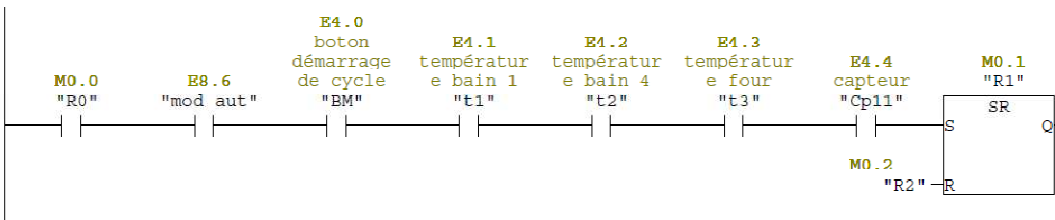
| Nom | Type de données | Adresse | Commentaire |
|---------|-----------------|---------|-------------|
| IN | | 0.0 | |
| OUT | | 0.0 | |
| IN_OUT | | 0.0 | |
| TEMP | | 0.0 | |
| RETURN | | 0.0 | |
| RET_VAL | | 0.0 | |

Bloc : FC1

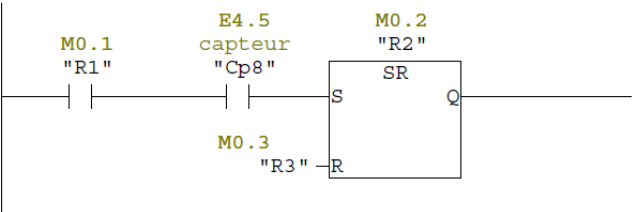
Réseau : 1



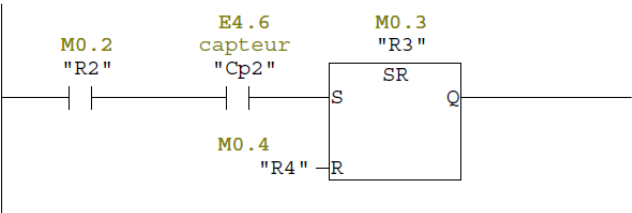
Réseau : 2



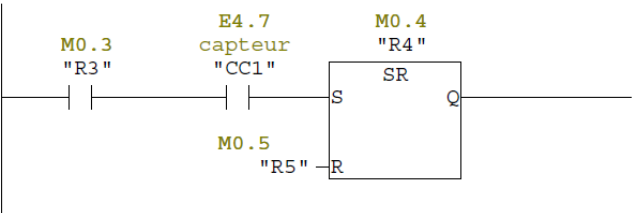
Réseau : 3



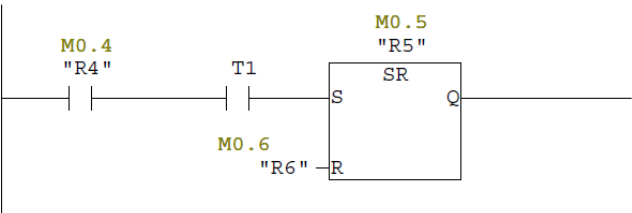
Réseau : 4



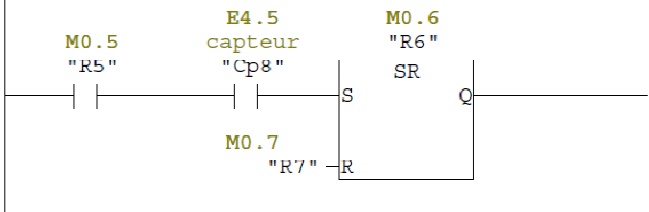
Réseau : 5



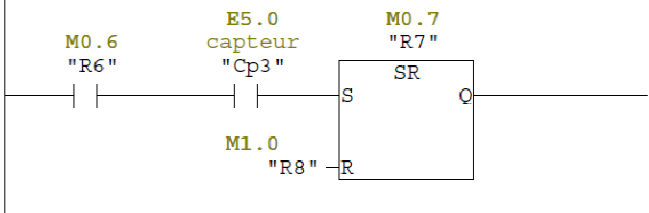
Réseau : 6



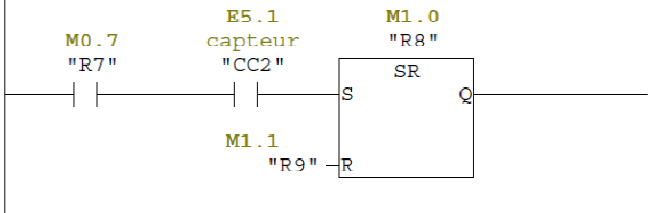
Réseau : 7



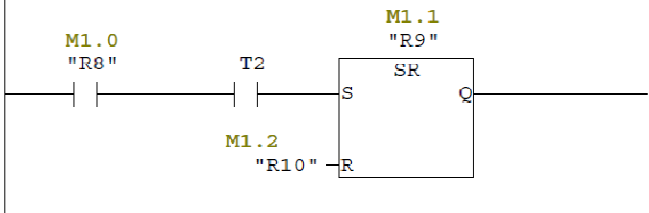
Réseau : 8



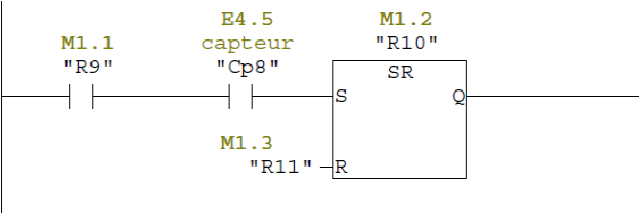
Réseau : 9



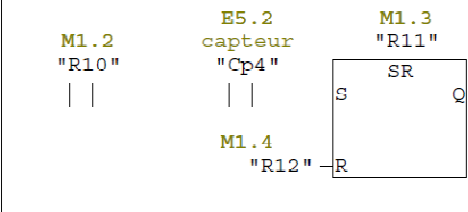
Réseau : 10



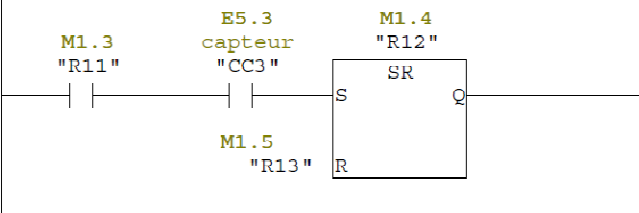
Réseau : 11



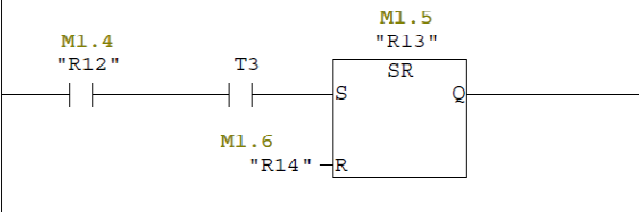
Réseau : 12



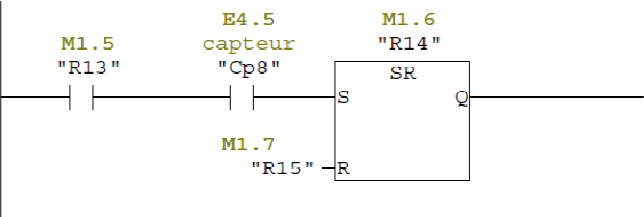
Réseau : 13



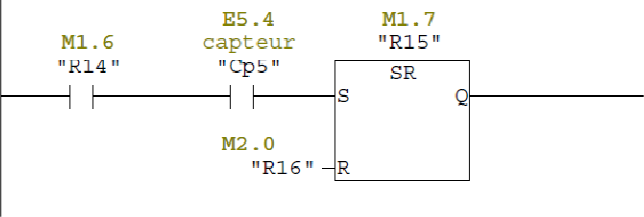
Réseau : 14



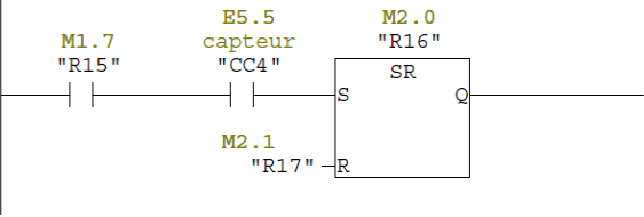
Réseau : 15



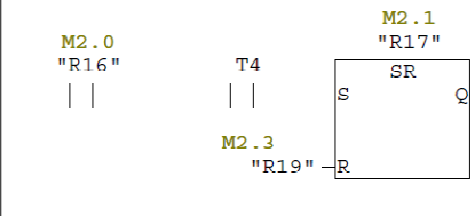
Réseau : 16



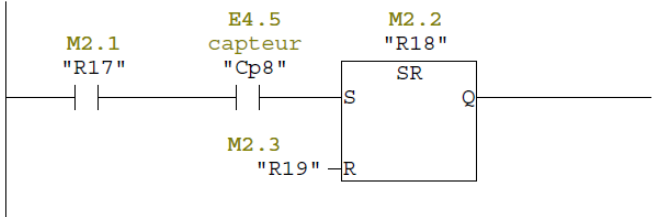
Réseau : 17



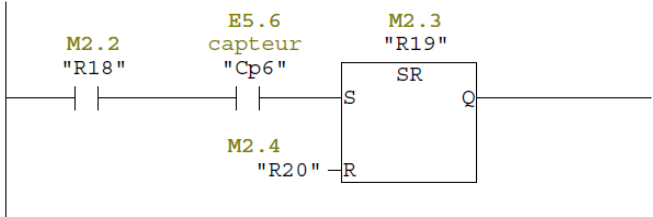
Réseau : 18



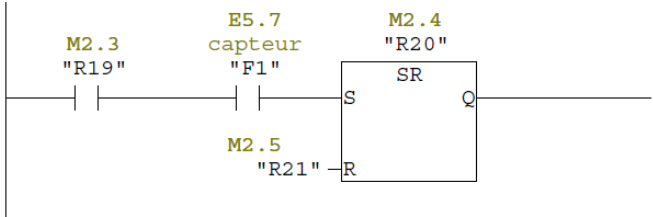
Réseau : 19



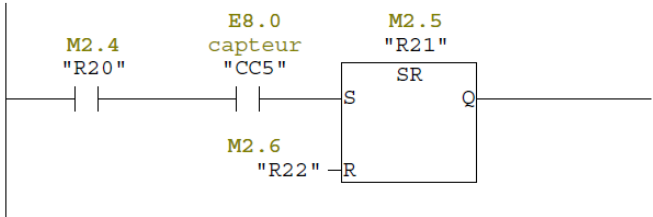
Réseau : 20



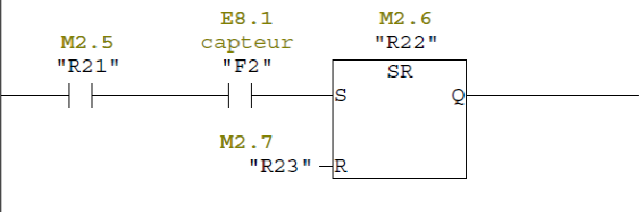
Réseau : 21



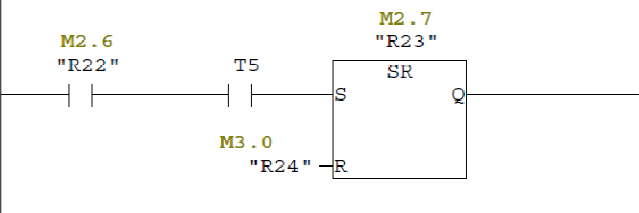
Réseau : 22



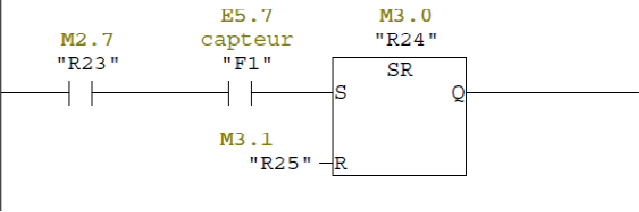
Réseau : 23



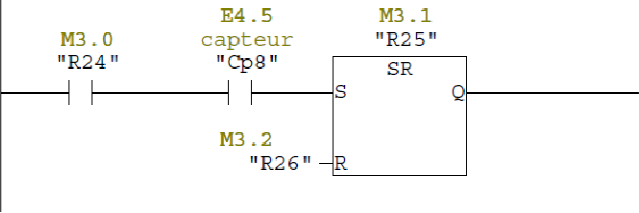
Réseau : 24



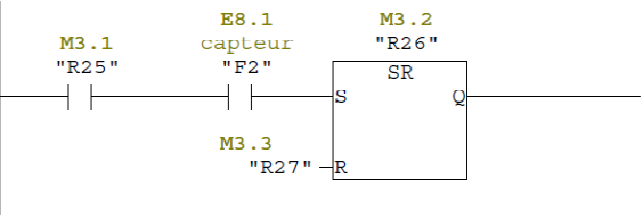
Réseau : 25



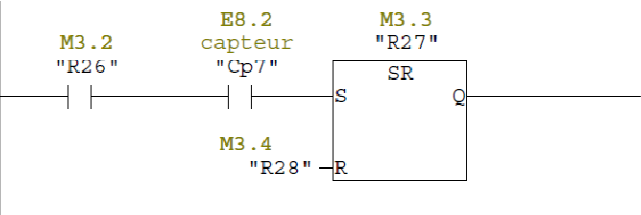
Réseau : 26



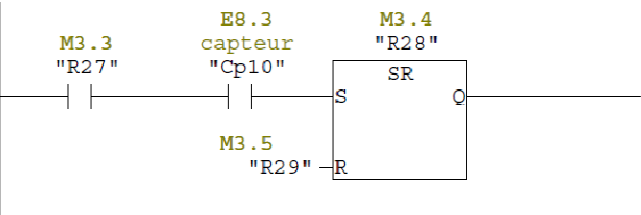
Réseau : 27



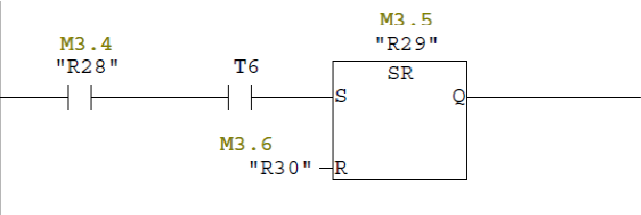
Réseau : 28



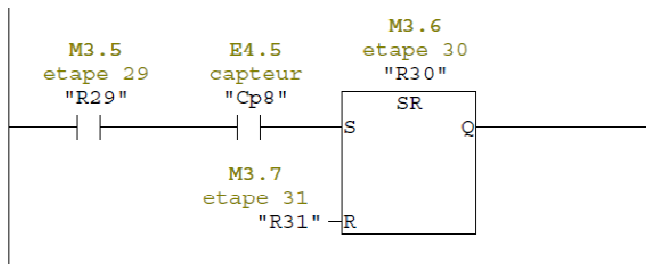
Réseau : 29



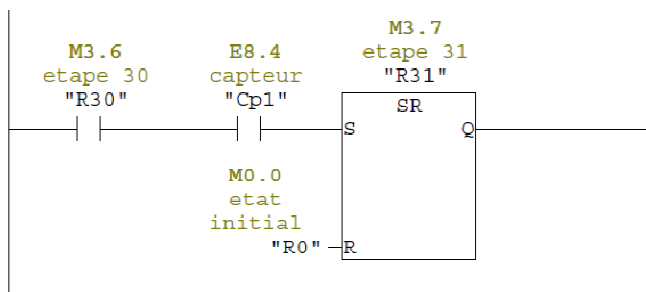
Réseau : 30



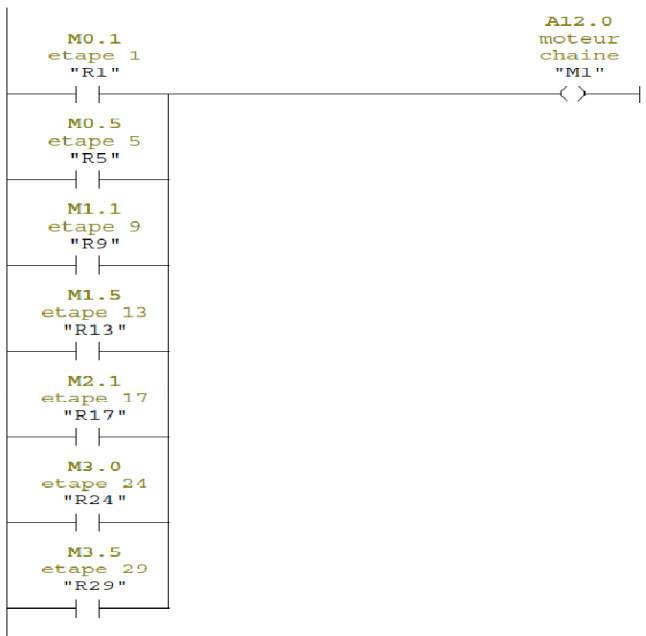
Réseau : 31



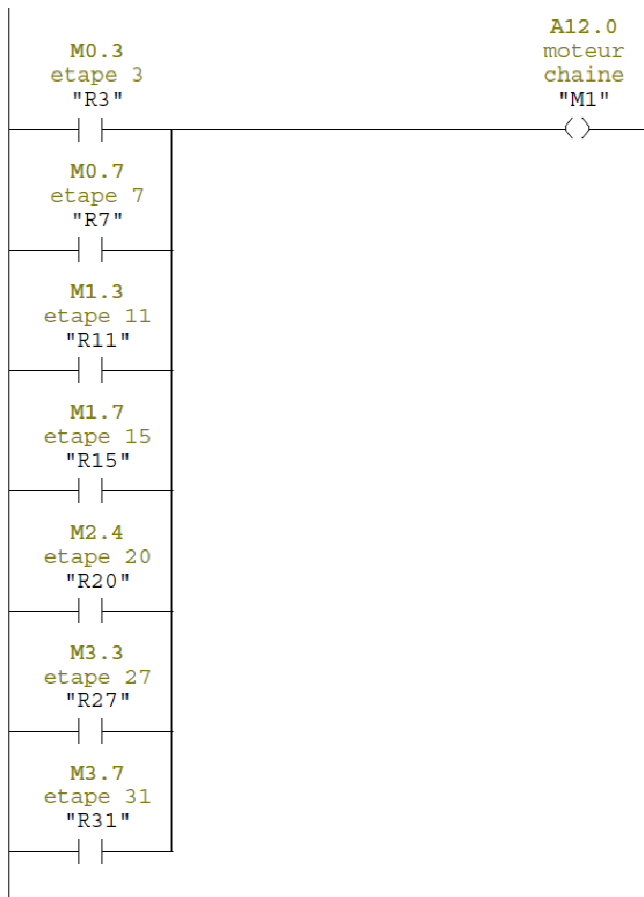
Réseau : 32



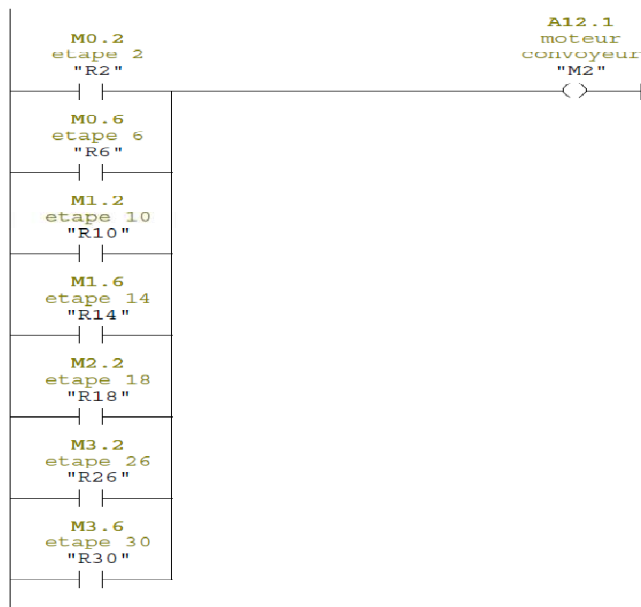
Réseau : 33 moteur chaine
 moteur de la chaine en haut, agitateur en marche



| | |
|---|---------------|
| Réseau : 34 | moteur chaine |
| moteur de la chaine en bas, agitateur en marche | |



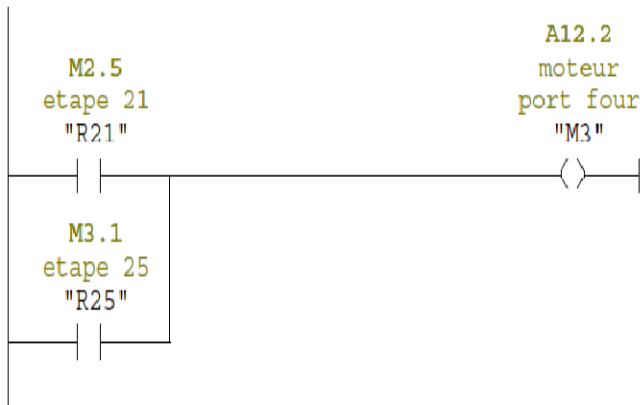
| | |
|---|------------------|
| Réseau : 35 | moteur convoyeur |
| moteur convoyeur, pompe agitateur en marche | |



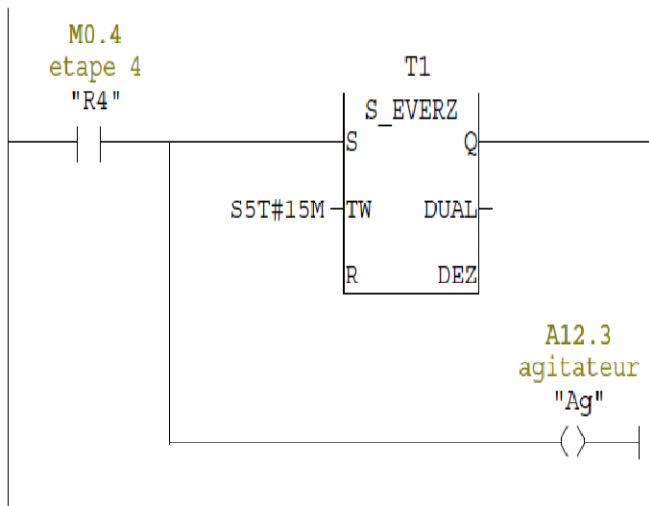
Réseau : 36 moteur port four
 Ouverture porte de four, pompe et agitateur en marche



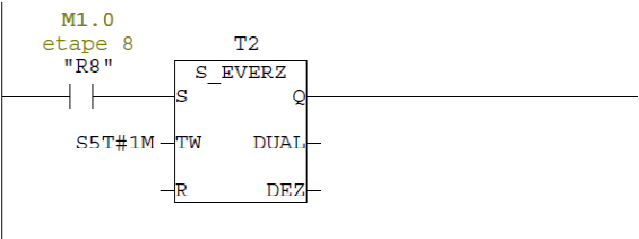
Réseau : 37 moteur port four
 fermeture porte four, pompe et agitateur en marche



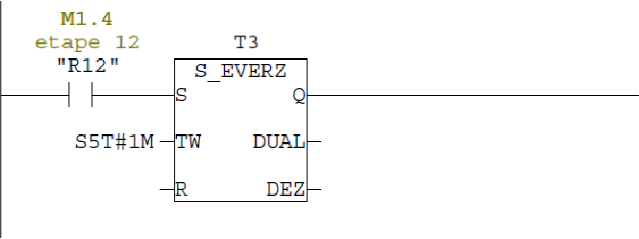
Réseau : 38



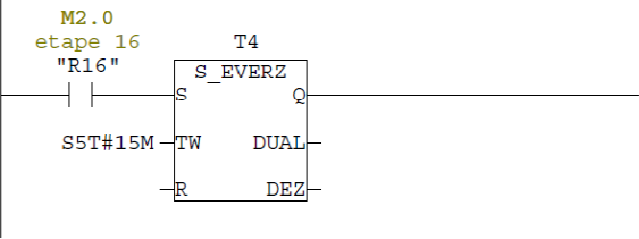
Réseau : 39



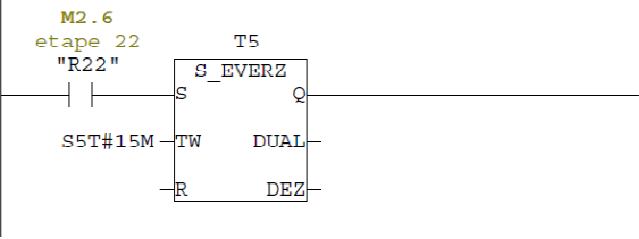
Réseau : 40



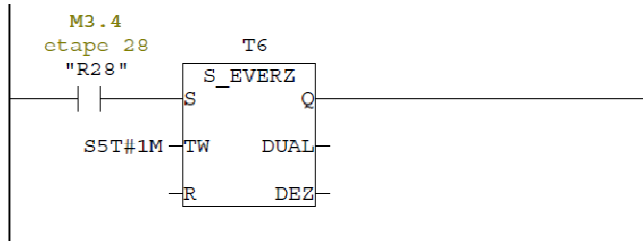
Réseau : 41



Réseau : 42

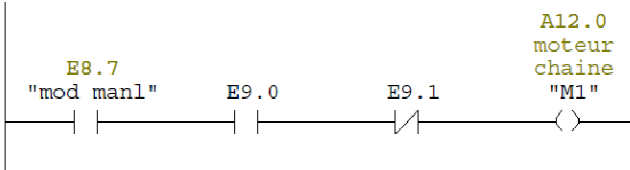


Réseau : 43



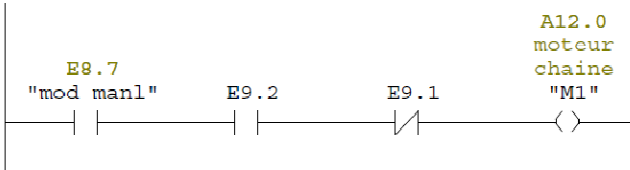
Réseau : 44

moteur chaine en haut



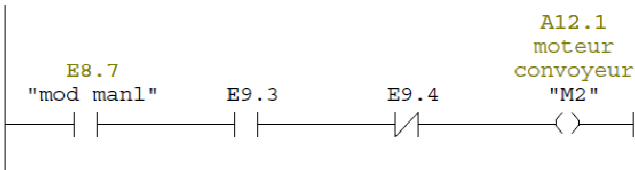
Réseau : 45

moteur chaine en bas



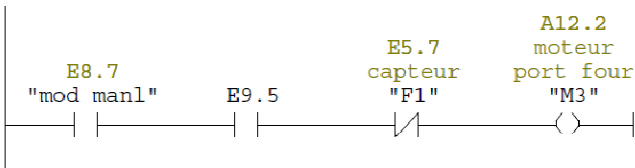
Réseau : 46

moteur convoyeur



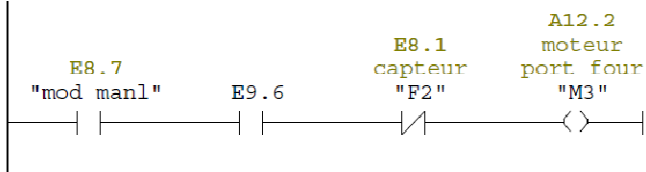
Réseau : 47

ouverture porte four

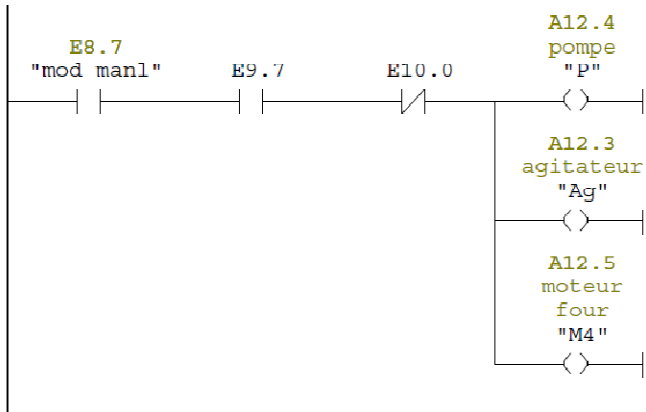


Réseau : 48

fermeture porte four



Réseau : 49

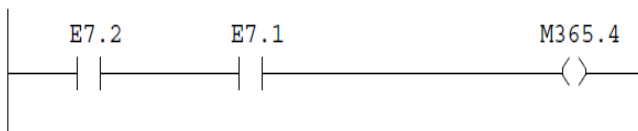


Chapitre IV : Commande du système par API S7 300

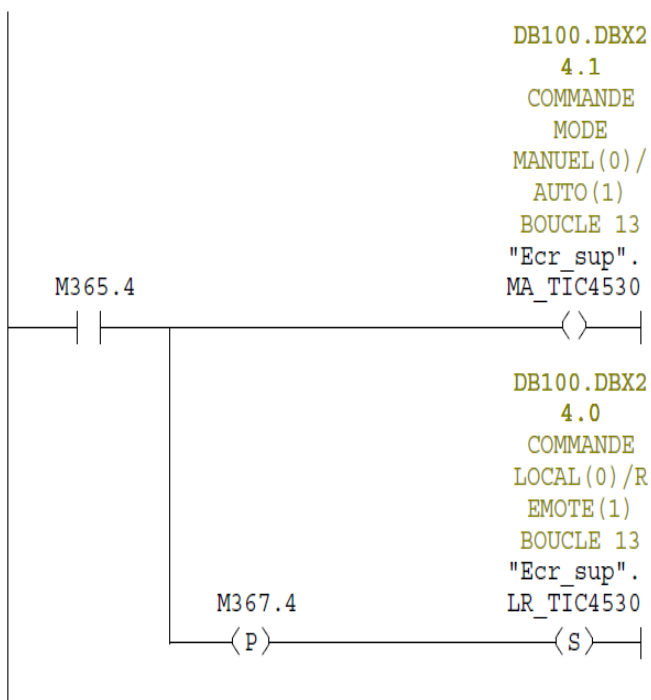
Programme Bloc FC2 : régulation de la température

Bloc : FC2

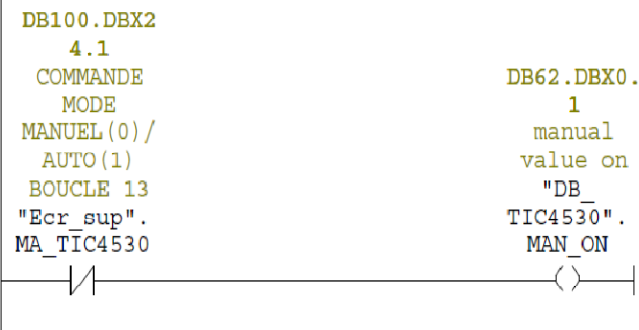
Réseau : 1 Validation régulation TIC4530



Réseau : 2 COMMANDE MODE MANUEL(0)/AUTO(1) BOUCLE 13



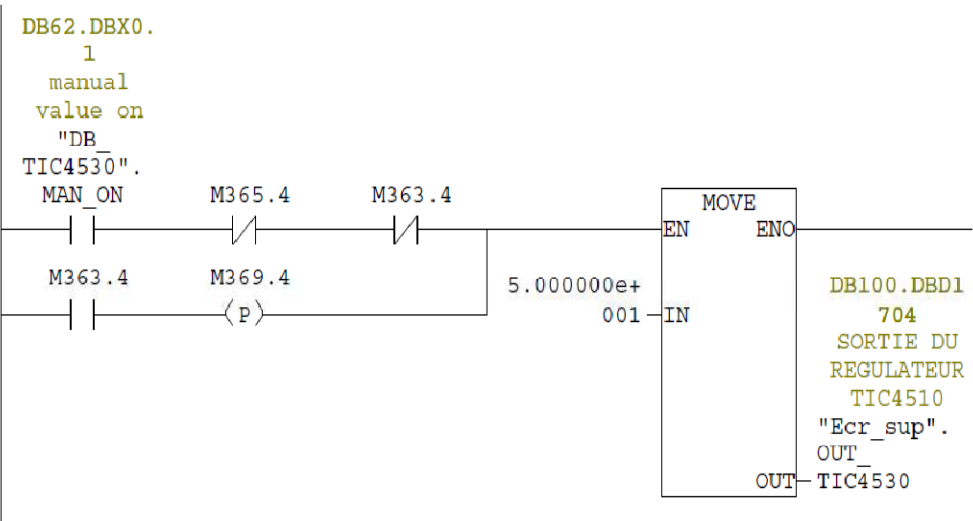
Réseau : 3 manual value on



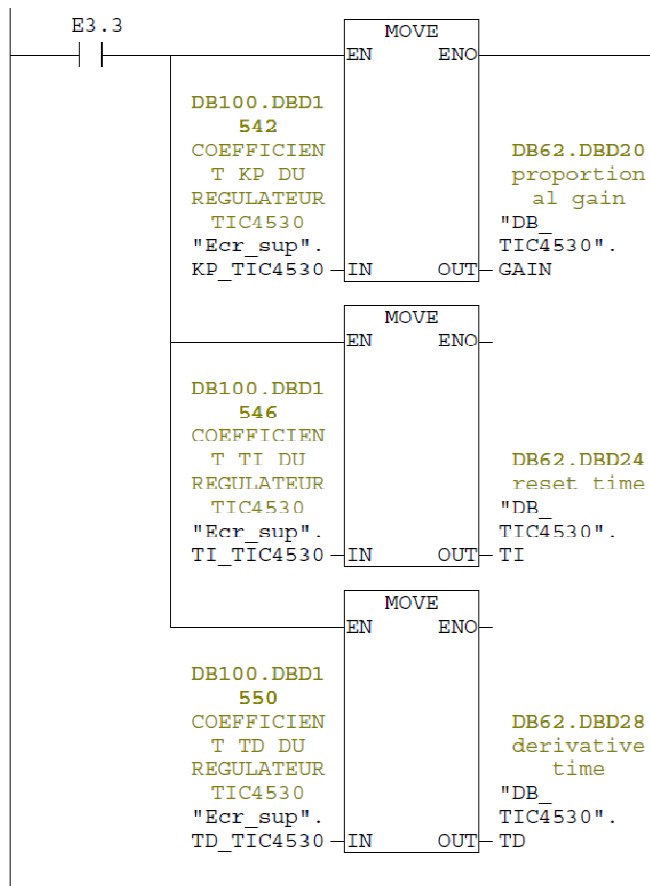
Réseau : 4 Liberation forçage sortie régulateur si étape initiale et arrêt:



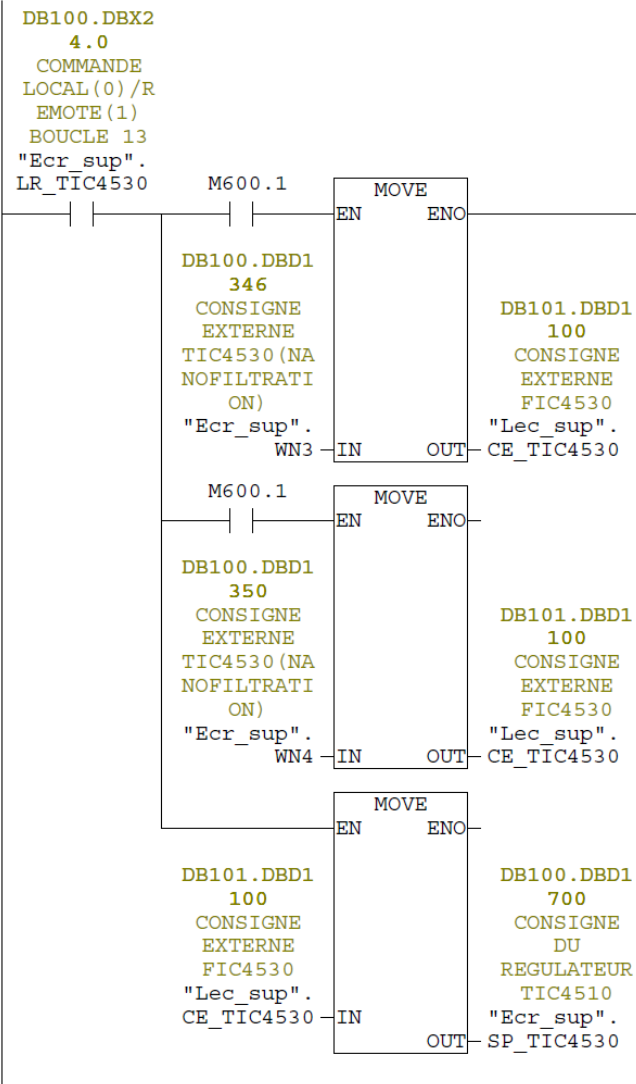
Réseau : 5



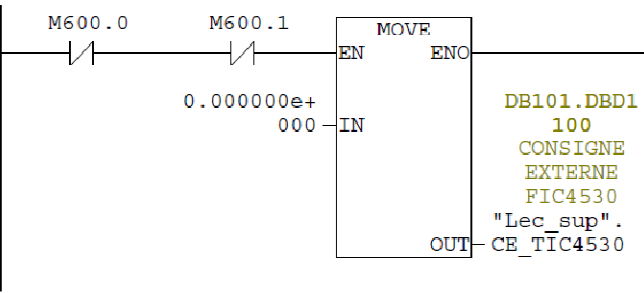
Réseau : 9



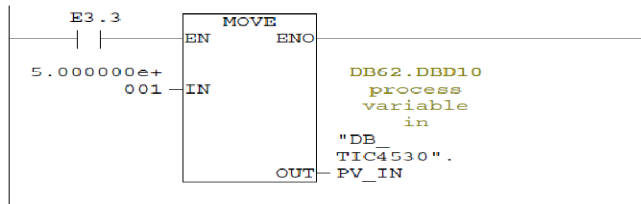
Réseau : 10



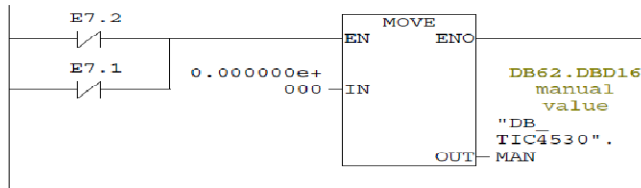
Réseau : 11



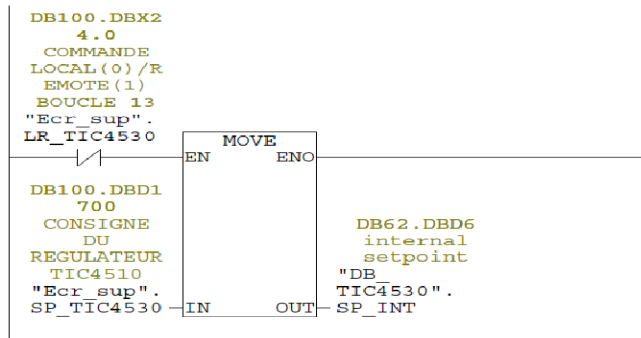
Réseau : 12



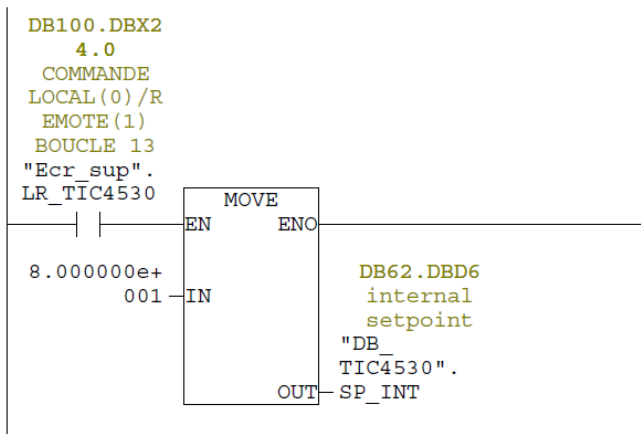
Réseau : 13



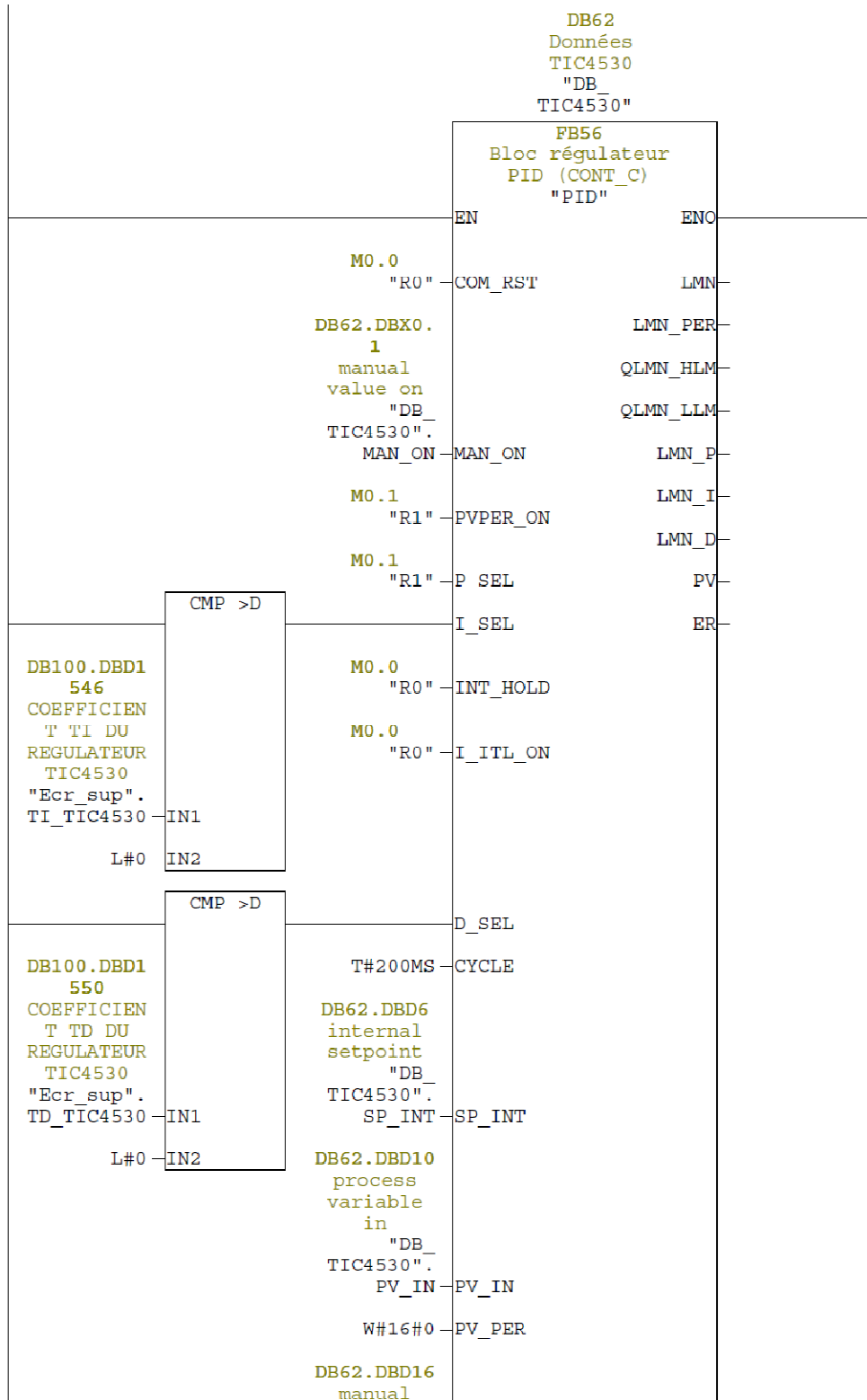
Réseau : 14



Réseau : 15

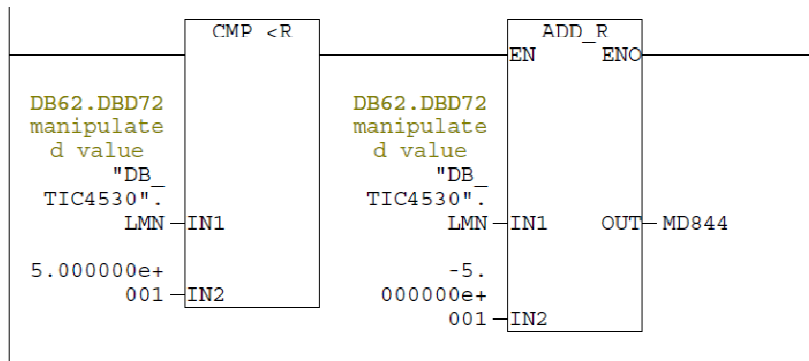


Réseau : 16

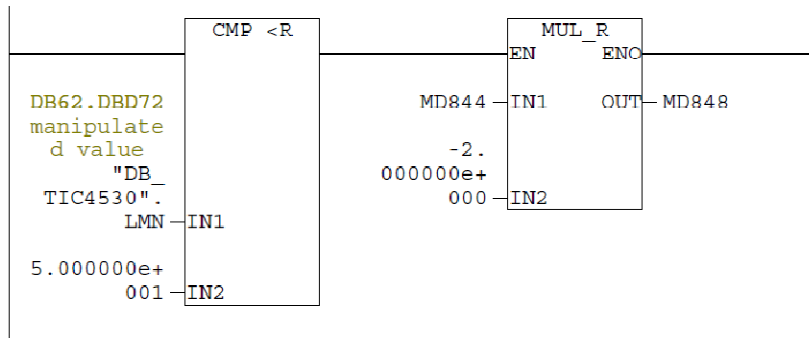


| | |
|------------|----------|
| value | |
| "DB_ | |
| TIC4530". | |
| MAN | MAN |
| DB62.DBD20 | |
| proportion | |
| al gain | |
| "DB_ | |
| TIC4530". | |
| GAIN | GAIN |
| DB62.DBD24 | |
| reset time | |
| "DB_ | |
| TIC4530". | |
| TI | TI |
| DB62.DBD28 | |
| derivative | |
| time | |
| "DB_ | |
| TIC4530". | |
| TD | TD |
| T#0MS | TM_LAG |
| 0.000000e+ | |
| 000 | DEADB_W |
| 1.000000e+ | |
| 002 | LMN_HLM |
| 0.000000e+ | |
| 000 | LMN_LLM |
| 1.000000e+ | |
| 000 | PV_FAC |
| 0.000000e+ | |
| 000 | PV_OFF |
| 1.000000e+ | |
| 000 | LMN_FAC |
| 0.000000e+ | |
| 000 | LMN_OFF |
| 0.000000e+ | |
| 000 | I_ITLVAL |
| 5.000000e+ | |
| 001 | DISV |

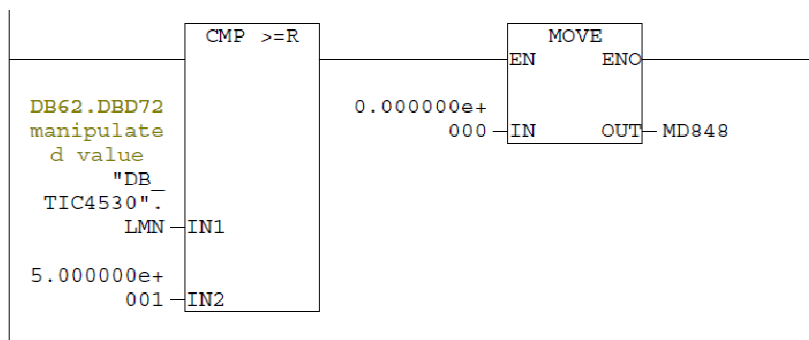
Réseau : 17



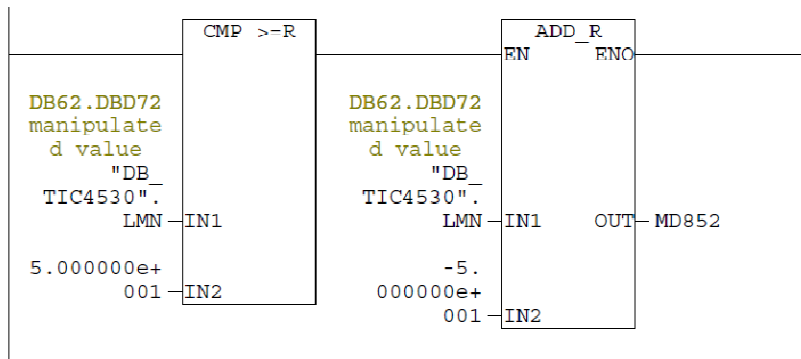
Réseau : 18



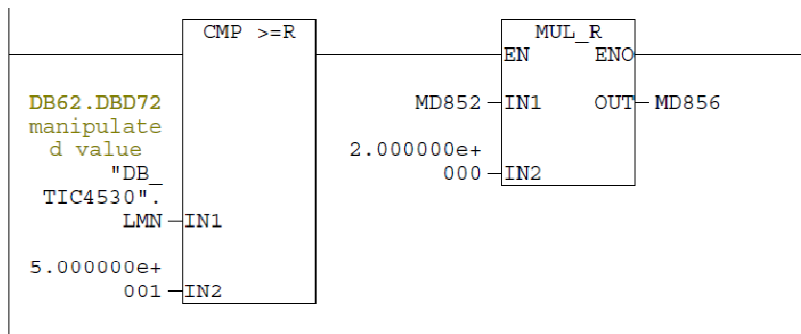
Réseau : 19



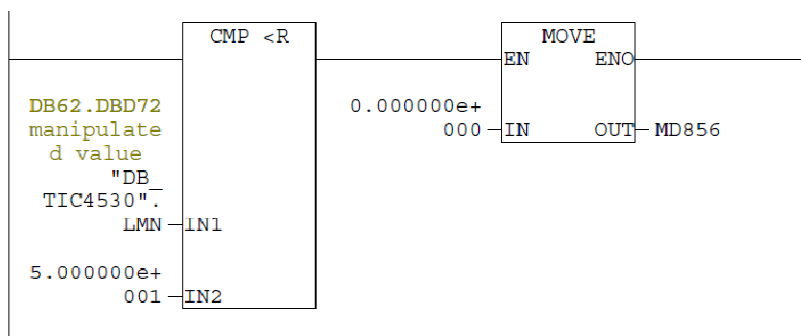
Réseau : 20



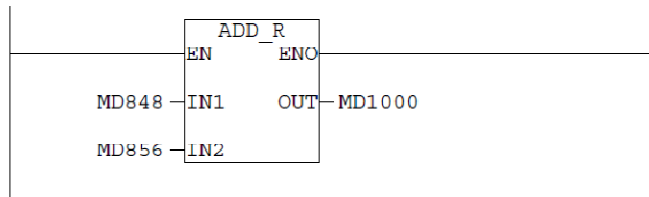
Réseau : 21



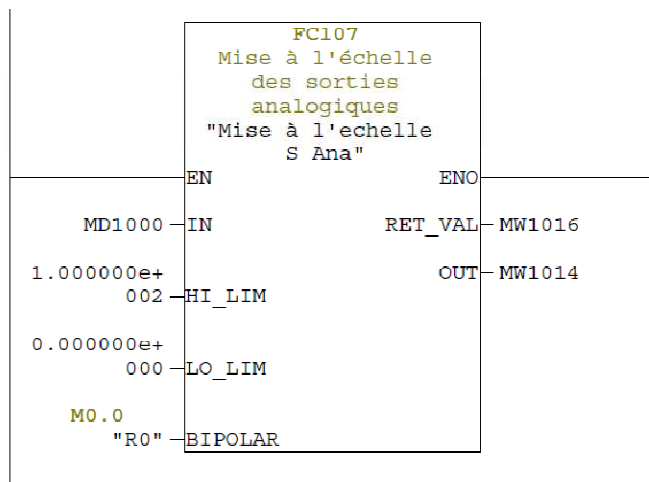
Réseau : 22



Réseau : 23



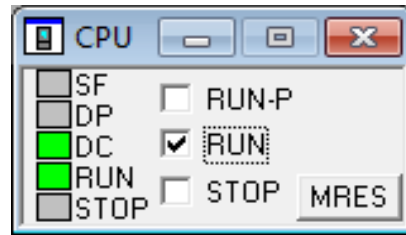
Réseau : 24



V Présentation du S7-PLCIM

L'utilisation de simulateur de module physique S7-PLCSIM nous permet d'exécuter et de tester le programme dans un automate de simulation que nous simulons dans un ordinateur. La simulation étant complètement réalisée au sein du logiciel STEP7.

Le S7-PLCSIM dispose d'une interface simple nous permettent de visualisé et de forcer les différentes paramètres utilisé par le programme (comme activé ou désactivé des entrées). Tout en exécutant le programme dans l'API de simulation, nous avons également la possibilité de mettre en œuvre les diverses applications du logiciel STEP 7 comme, par exemple, le teste de bloc afin de visualisé les variable d'entrés et de sorties.

V-1 Commande de la CPU**V-1-1 Fenêtre CPU**

- **SF** : signal une erreur dans le système.
- **DP** : indique l'état de communication avec les E/S décentralisées.
- **DC** : indique si la CPU est mise sous la tension ou non.
- **RUN** : indique si la CPU en mode de visualisation.
- **STOP** : indique que la visualisation est arrêtée.
- **RUN-P, RUN**: c'est pour sélectionner le mode de fonctionnement de la CPU.
- **MRES** : c'est pour effectuer un effacement général de la mémoire de la CPU.

Les modes de fonctionnement de la CPU

- **Mode STOP** : dans ce mode, la CPU n'exécute pas le programme et pour pouvoir charger un programme, la CPU doit être obligatoirement dans ce mode.
- **Mode RUN** : dans ce mode, la CPU exécute le programme mais on ne peut ni charger de nouveaux programmes ou blocs ni forcer les variables.
- **Mode RUN-P** : le programme est exécuté dans la CPU, et il est possible de charger des nouveaux programmes ou blocs et de forcer les variables en utilisant les applications de STEP7.

V-2 Mise en route de logiciel S7-PLCSIM


Le mode de simulation est disponible à partir du gestionnaire de projet SIMATIC à condition qu'aucune liaison à des API réels ne soit établie. On peut suivre la procédure suivante pour la mise en route de logiciel S7-PLCSIM.

La procédure à suivre est :


- Ouvrir le gestionnaire de projet SIMATIC.


- Cliquer sur  ou sélectionner la commande **Outils > Simulation du module**

Cela lance l'application S7-PLCSIM et ouvre une fenêtre CPU.

- Parcourir la boîte de recherche jusqu'au projet.
- Parcourir la boîte de recherche jusqu'au classeur des blocs.
- Cliquer sur  charger ou choisir la commande **Système cible> Charger** pour charger le dossier bloc dans l'API de simulation

Dans l'application S7-PLCSIM, on crée de nouvelles fenêtres pour visualiser les informations provenant de l'API simulation.

- Cliquez sur  ou choisissez la commande **Insertion> Entrée** pour créer une fenêtre dans laquelle vous pouvez visualiser et forcer des variables dans la zone de mémoire des entrées (zone E). Cette fenêtre s'ouvre avec l'adresse de mémoire par défaut EB0. Mais on peut modifier l'adresse (EB1, EB2...).


- Cliquez sur  ou choisissez la commande **Insertion> Sortie** pour créer une fenêtre dans laquelle vous pouvez visualiser et forcer des variables dans la zone de mémoire des sorties (zone A). Cette fenêtre s'ouvre avec l'adresse de mémoire par défaut AB0. Mais on peut modifier l'adresse (AB1, AB2...).

- Ça sera la même procédure pour la création de fenêtre de mémentos. Temporisations et de conteurs.

- Choisir le menu CPU dans la fenêtre du S7-PLCSIM et vérifier que la commande mètre sous tension est activée.


- Choisir la commande **exécution > mode d'exécution** et vérifier que la commande cycle continue est activée.

- Mettre la CPU de la simulation en marche en cliquant sur l'une des cases à cocher RUN ou RUN-P.

Pour sauvegarder la version actuelle de la simulation d'API, cliquer sur  ou choisissez la commande **Fichier>Enregistrer CPU**.

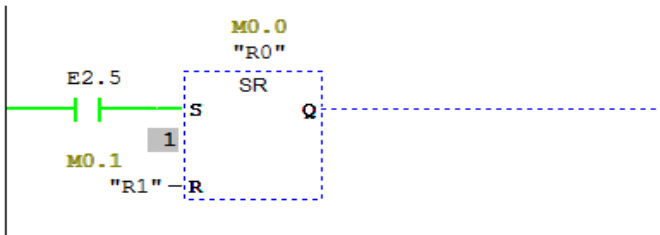
Une fois toutes les fenêtres d'entrées et de sorties sont prêtes, nous activons les entrées voulues pour lire l'état des sorties.

V-3 Visualisation de l'état de programme

Après le chargement du programme dans la CPU du simulateur et la mise de cette dernière en mode « RUN-P », le STEP7 nous permet de visualiser l'état du programme soit en cliquant sur l'icône  ou en sélectionnant la commande **Test> Visualiser**.

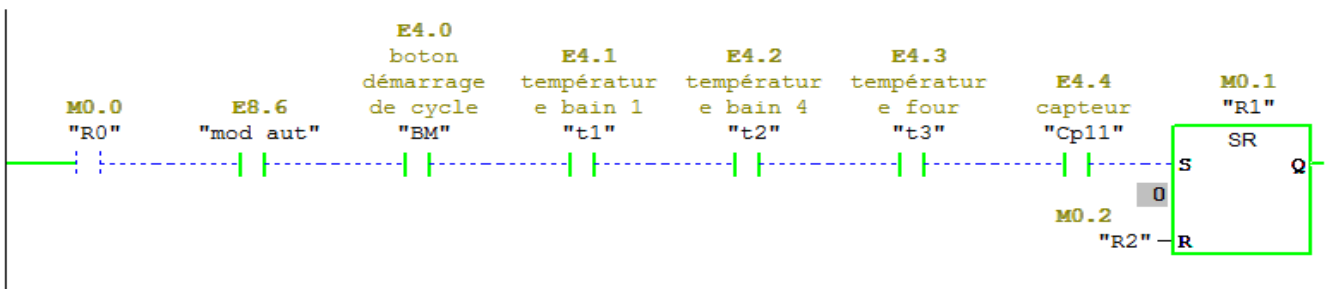
Réseau 1: Titre :

Commentaire :



Réseau 2: Titre :

Commentaire :



VI Conclusion

Dans ce chapitre, nous avons présenté une vue d'ensemble de l'automate programmable industriel S7-300 que nous avons adopté pour le système et comment créer un projet avec le logiciel STEP7.

La validation du programme de conduite que nous avons développé à été réalisé grâce au logiciel de simulation de modules S7-PLSIM. Cette procédure nous a permis d'apporter les modifications nécessaires pour la concrétisation de nos programmes d'automatisation.

Conclusion générale

Dans le cadre de notre projet de fin d'études, nous nous sommes intéressés à l'automatisation du système de traitement de surface par immersion au sein de l'Entreprise d'Electroménager (ENIEM).

L'automatisation est un domaine qui a rapport à plusieurs disciplines scientifiques ou autres qui associe les notions de la mécanique, de l'électronique et de l'informatique ; elle permet d'exécuter des tâches industrielles avec une intervention humaine très réduite.

La commande des processus par des API est une solution de plus en plus recherchée dans l'industrie, en raison de l'exactitude et la justesse des traitements qu'ils effectuent pour générer la commande adéquate à tout moment et dans toutes les conditions.

Dans notre travail, nous avons approfondi nos connaissances sur le logiciel de programmation STEP7 et l'automate programmable S7-300 de la famille SIEMENS.

L'API S7-300 procure plusieurs avantages tels que la flexibilité, la facilité d'extension de ses modules et la possibilité de visualiser et de valider le programme établi avant son implantation grâce à son logiciel de simulation S7-PLCSIM.

Ce que nous retenons du travail que nous avons effectué à l'ENIEM est primordial dans notre vie professionnelle.

Ce stage pratique nous est bénéfique en tenant compte des nombreux avantages qu'il présente ; Cela nous a permis d'améliorer et d'enrichir nos connaissances théoriques et de les mettre en pratique et de se familiariser avec les automates programmables industriels.

Nous espérons que notre travail sera d'une utilité pour d'autres promotions à venir, qui veulent travailler dans ce domaine et qu'il sera réalisé au sein de l'entreprise.

BIBLIOGRAPHIE

- [1] **PATRICK TRAU** « Le GRAFCET et sa mise en service » ULP 1997
Edition Paris.
- [2] **N.KEBDI** et **M.LAMRI**, « Automatisation et supervision de système de gestion des eaux à **FRUITAL Coca-Cola** (Rouiba) » **Mémoire du master professionnel en électrotechnique, U.M.M.Tizi-Ouzou 2012.**
- [3] **A.HAMDAD** et **Y.DAHMANI**, « Etude d'une station de pompage pilotée par des automates programmables SIMATIC S7 300» **Thèse d'ingénieur électrotechnique, U.M.M.Tizi-Ouzou 2006.**
- [4] **Alain GONZAGA** Manuel d'utilisation / document français/ les automates programmables industriels.
- [5] **K.oumezzaouche** et **R.STITI**, « **Adaptation d'un automate programmable S7 300 pour une chaîne de production des portes réfrigérateurs** » Mémoire master professionnel en électrotechnique, U.M.M.Tizi-Ouzou 2012.
- [6] **S.AKROUR** et **N.ARAB**, « Automatisation et supervision d'une station de transport du sucre à l'usine **Coca-Cola**» **Thèse d'ingénieur automatique, U.M.M.Tizi-Ouzou 2009.**
- [7] Manuels SIMATIC français/ STEP7/ programmation avec STEP7 de Logiciel STEP7 version 5.5.
- [8] Documentation ENIEM.

Propriétés de la table des mnémoriques

Nom : Mnémoriques
 Auteur :
 Commentaire :
 Date de création : 07/05/2013 12:43:32
 Dernière modification : 12/09/2013 11:48:52
 Dernier filtre sélectionné : Tous les mnémoriques
 Nombre de mnémoriques : 83/83
 Dernier tri : Mnémorique ordre décroissant

| Etat | Mnémorique | Opérande | Type de données | Commentaire |
|------|------------------------|----------|-----------------|--|
| | VAT_1 | VAT 1 | | |
| | TIC4530 | FC 57 | FC 57 | Régulation chaud/froid TIC4530 |
| | t3 | E 4.3 | BOOL | température four |
| | t2 | E 4.2 | BOOL | température bain 4 |
| | t1 | E 4.1 | BOOL | température bain 1 |
| | SCALE | FC 105 | FC 105 | Scaling Values |
| | Redemarrage API | OB 100 | OB 100 | Redemarrage API (Froid ou chaud) |
| | R9 | M 1.1 | BOOL | etape 9 |
| | R8 | M 1.0 | BOOL | etape 8 |
| | R7 | M 0.7 | BOOL | etape 7 |
| | R6 | M 0.6 | BOOL | etape 6 |
| | R5 | M 0.5 | BOOL | etape 5 |
| | R4 | M 0.4 | BOOL | etape 4 |
| | R31 | M 3.7 | BOOL | etape 31 |
| | R30 | M 3.6 | BOOL | etape 30 |
| | R3 | M 0.3 | BOOL | etape 3 |
| | R29 | M 3.5 | BOOL | etape 29 |
| | R28 | M 3.4 | BOOL | etape 28 |
| | R27 | M 3.3 | BOOL | etape 27 |
| | R26 | M 3.2 | BOOL | etape 26 |
| | R25 | M 3.1 | BOOL | etape 25 |
| | R24 | M 3.0 | BOOL | etape 24 |
| | R23 | M 2.7 | BOOL | etape 23 |
| | R22 | M 2.6 | BOOL | etape 22 |
| | R21 | M 2.5 | BOOL | etape 21 |
| | R20 | M 2.4 | BOOL | etape 20 |
| | R2 | M 0.2 | BOOL | etape 2 |
| | R19 | M 2.3 | BOOL | etape 19 |
| | R18 | M 2.2 | BOOL | etape 18 |
| | R17 | M 2.1 | BOOL | etape 17 |
| | R16 | M 2.0 | BOOL | etape 16 |
| | R15 | M 1.7 | BOOL | etape 15 |
| | R14 | M 1.6 | BOOL | etape 14 |
| | R13 | M 1.5 | BOOL | etape 13 |
| | R12 | M 1.4 | BOOL | etape 12 |
| | R11 | M 1.3 | BOOL | etape 11 |
| | R10 | M 1.2 | BOOL | etape 10 |
| | R1 | M 0.1 | BOOL | etape 1 |
| | R0 | M 0.0 | BOOL | etat initial |
| | PULSEGEN | FB 43 | FB 43 | Pulse Generation |
| | PID | FB 56 | FB 56 | Bloc régulateur PID (CONT_C) |
| | P | A 12.4 | BOOL | pompe |
| | mod manl | E 8.7 | BOOL | |
| | mod aut | E 8.6 | BOOL | |
| | Mise à l'échelle S Ana | FC 107 | FC 107 | Mise à l'échelle des sorties analogiques |

| Etat | Mnémonique | Opérande | Type de données | Commentaire |
|------|-------------------------|----------|-----------------|---|
| | Mesures | DB 10 | DB 10 | Table d'acquisition anamlogique et autre mesure |
| | M4 | A 12.5 | BOOL | moteur four |
| | M3 | A 12.2 | BOOL | moteur port four |
| | M2 | A 12.1 | BOOL | moteur convoyeur |
| | M1 | A 12.0 | BOOL | moteur chaine |
| | Lec_sup | DB 101 | DB 101 | Gestion infos en lec. avec supervision |
| | FB1 | FB 1 | FB 1 | |
| | F2 | E 8.1 | BOOL | capteur |
| | F1 | E 5.7 | BOOL | capteur |
| | ew 256 | EW 256 | WORD | temp capti |
| | Erreur de programmation | OB 121 | OB 121 | |
| | Erreur acces | OB 122 | OB 122 | Erreur d'accés au module |
| | Ecr_sup | DB 100 | DB 100 | Gestion infos en lec./ecr. avec supervision |
| | Diagnostic E/S | OB 82 | OB 82 | |
| | Défauts E/S | OB 86 | OB 86 | |
| | DB_TIC4530 | DB 62 | FB 56 | Données TIC4530 |
| | Cycle Execution | OB 1 | OB 1 | |
| | CYC_INT5 | OB 35 | OB 35 | Cyclic Interrupt 5 |
| | Cp9 | E 8.5 | BOOL | capteur |
| | Cp8 | E 4.5 | BOOL | capteur |
| | Cp7 | E 8.2 | BOOL | capteur |
| | Cp6 | E 5.6 | BOOL | capteur |
| | Cp5 | E 5.4 | BOOL | capteur |
| | Cp4 | E 5.2 | BOOL | capteur |
| | Cp3 | E 5.0 | BOOL | capteur |
| | Cp2 | E 4.6 | BOOL | capteur |
| | Cp11 | E 4.4 | BOOL | capteur |
| | Cp10 | E 8.3 | BOOL | capteur |
| | Cp1 | E 8.4 | BOOL | capteur |
| | CONT_S | FB 42 | FB 42 | Step Control |
| | CONT_C | FB 41 | FB 41 | Continuous Control |
| | Ci5 | E 8.0 | BOOL | capteur |
| | Ci4 | E 5.5 | BOOL | capteur |
| | Ci3 | E 5.3 | BOOL | capteur |
| | Ci2 | E 5.1 | BOOL | capteur |
| | Ci1 | E 4.7 | BOOL | capteur |
| | BM | E 4.0 | BOOL | boton démarrage de cycle |
| | Ag | A 12.3 | BOOL | agitateur |