République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Ecole Nationale Polytechnique

Département d'Automatique

Projet de fin d'études

En Vue de l'Obtention du Diplôme d'Ingénieur d'Etat en Automatique

Thème

Automatisation d'une ligne de traitement de surface

Réalisé par :

MAHOUR Saoussane

Soutenu devant le jury composé de **:**

M^{r} L. ABDELOUEL	Président
M^{r} H.ACHOUR	Examinateur
P^{r} E.M.BERKOUK	Promoteur
M^{r} B. BOUGDOUR	Co-promoteur

ENP 2015

ENP 10 Avenue Hacen Badi B.P 182 , El-Harrache 16200, Alger , ALGERIE

REMERCIEMENTS

Nous remercions Dieu le tout puissant qui nous a donné le courage et la volonté de réaliser ce travail.

Nous tenons à exprimer nos vifs remerciements à notre promoteur Pr.BERKOUK de l'Ecole Nationale Polytechnique pour nous avoir encadrer durant notre projet de fin d'études et nous conseillé tout le long de notre travail.

Nous remercions également notre co-promoteur Mr BOUGDOUR, pour son encadrement, sa générosité dans la réalisation du projet, et sa confiance au sein de l'entreprise.

Nous remercions chaleureusement les membres du jury pour l'honneur qu'ils nous ont fait en acceptant d'évaluer notre projet.

Nous souhaitons aussi remercier tous les enseignants de l'Ecole Nationale Polytechnique d'Alger, et en particulier, Nos professeurs d'Automatique qui nous ont encadrés auparavant et tous nos enseignants pour les connaissances qu'ils nous ont transmis, leur disponibilité et leurs efforts.

Enfin, tous nos remerciements à toutes les personnes qui ont contribué de près ou de loin pour l'accomplissement de ce travail.

Je dédie ce travail :

- *A mes chers parents*

Pour leur soutien, leur patience, leur sacrifice
et leur amour, vous méritez tout éloge,
vous qui avez fait de moi et mes sœurs et frères ce que nous sommes maintenant.
Nous espérons être l'image que vous êtes fait de nous, que dieu vous garde et vous bénisse.

- *A ma grand-mère, qui m'a toujours soutenue, qu'Allah la protège.*

- *A mes sœurs : Nassima, feroudja, Djamila, Saliha, Sabrina, Kahina.*

- *A mes frères : Mohamed, Islam.*

- *A mes amies et camarades : du primaire, moyen, secondaire, EPSTO et ENP d'Alger.*

Et à tous ceux qui m'aiment et j'aime

Saoussane

Résumé

Le travail présent dans ce mémoire se base essentiellement sur l'utilisation des automates programmables SIEMENS.Notre travail est l'automatisation d'une ligne de traitement de surface au niveau de l'entreprise ENIEM, par l'utilisation du logiciel de programmation STEP7, et le logiciel de conception des interfaces homme machine WinCC flexible.

Mots clés : Automates programmable SIEMENS, STEP 7, WinCC .

Abstract

The work presented in this project is essentially based on the utilization of SIEMENS Programmable Logic Controllers. Our work is the automation of a surface treatment line at ENIEM Company, by using the STEP 7 programming software and design software human machine interfaces WinCC flexible.

Keywords: Programmable Logic Controller SEIMENS, STEP7, WinCC .

ملخص

ا لعمل المنجز في هذه المذكرة يتمحور اساسا حول استخدام SIEMENS PLC.عملنا هو استخدام الالي لخط المعالجة السطحية في شركة ENIEM وذلك باستخدام برنامج البرمجةSTEP7 و برنامج تصميم واجهات الجهاز البشري WinCC مرن.

الكلمات المفتاحية :WinCC , STEP7, SIEMENS PLC.

Sommaire

Chapitre I: Automate Programmable Industriel

Chapitre II: GRAFCET et GEMMA

Chapitre III: Step7, Win CC, Logiciels de programmation et de supervision

Chapitre IV : Présentation de la chaine de traitement de surface

Chapitre V : Elaboration de GRAFCET, Programmation et Supervision de la chaine

Liste des Figures

Liste des tableaux

Symboles et abréviations

ADEPA : Agence Internationale pour le Développement de la Production Automatisée

API: Automate Programmable Industriel

CPU: Central Processing Unit

ENIEM : Entreprise Nationale des Industries de l'Electroménager

EPROM: Erasable Programmable Read Only Memory

FB : Blocs fonctionnels

FBD : Fonction Bloc Diagram

FC : Fonctions

GEMMA : Guide d'Etude des Modes de Marche et d'Arrêt

GRAFCET : GRAphe Fonctionnel de Commande Etape Transition

HMI : Interface Homme Machine

IL : Instruction List

LD : Ladder Diagram

LIST : Le langage de liste d'instructions

LOG : Le langage à base de logigramme

MPI : Multi Point Interface

OB : Blocs d'organisation

OP : Pupitre opérateur

PC : Personnel Computer

P.C : Partie Commande

PG : La console de programmation sur le terrain

PLC: Programmable Logic Controllers

P.O: partie opérative

PROFIBUS: PROcess Field Bus

RAM: Random Access Memory

ROM: Read Only Memory

SAP : Système Automatisé de Production

SFB : Bloc Fonctionnel Système

SFC : Bloc programme pour le langage évolué textuel

ST : Structured Text

TOR : Toute Ou Rien

UAL : Unité Arithmétique et Logique

WinCC : Windows Contrôle Center

Introduction générale

La compétitivité dans le secteur industriel devient plus vive du fait de la mondialisation et la globalisation des marchés. L'entreprise est désormais plongée dans un milieu fortement concurrentiel dans lequel la seule arme qu'elle possède est sa capacité à réagir efficacement et surtout rapidement à une demande de plus en plus exponentielle.

Afin de pouvoir répondre à cette contrainte de rapidité, l'entreprise industrielle est appelée à raccourcir les délais de l'ensemble des étapes de sa chaine de production .la robotisation et plus explicitement l'automatisation sont devenues aujourd'hui les parfaites solutions pour remédier à ces contraintes.

L'automatisme est une discipline importante et nécessaire dans tous les secteurs industriels. Il facilite la tâche des opérateurs intervenants dans toute installation industrielle. Il permet de développer des systèmes automatisés qui assurent des tâches dangereuses, répétitives et dans des milieux hostiles pour l'homme. L'automatisation de toute unité de production augmente la productivité et améliore la qualité du produit. Les automates programmables industriels représentent l'élément important de la chaîne automatisée, car ils gèrent des bonnes performances, meilleure flexibilité et facilite la maintenance. Une automatisation performante assure en plus d'un fonctionnement fiable de l'installation industrielle, la détection de toute anomalie éventuelle. La diversité des processus industriels nécessite des connaissances sur l'aspect processus et les différentes technologies du domaine de l'automatisme. Les automatismes câblés qui ont précédé les automates programmables industriels présentent des inconvénients majeurs relatifs à sa configuration et à la maintenance.

C'est justement la raison qui nous conduit à porter notre réflexion lors de notre stage au sein de l'entreprise ENIEM sur l'automatisation de la chaine de traitement de surface.

Le but de ce travail est l'élaboration d'une solution à base d'API pour automatiser la ligne de traitement de surface installée au niveau de l'unité cuisson. L'objectif poursuit dans ce travail est de créer à travers l'automatisation de la chaine de production un gain en énergie.

A cet effet, l'entreprise nous a remis un cahier des charges spécifique dont il fallait respecter les exigences, et sur la base duquel notre travail sera validé.

Le présent travail s'articule autour de cinq chapitres. Le premier chapitre est consacré à une étude générale sur les automates programmables industriels, ses caractéristiques propres, matérielles et logicielles.

Le deuxième chapitre est un rappel sur la méthodologie d'élaborer un GRAFCET et GEMMA.

Dans le troisième chapitre, On présente un outil de programmation des automates programmables industriels, le STEP7 qui permet au plus de programmer, de simuler le programme par S7-PLCSIM après on présente le logiciel de conception de l'interface homme machine WinCC flexible, Ce dernier se charge à la supervision du processus.

Dans le quatrième chapitre, on donne une description générale du processus industriel effectué, les différents constituants de la chaîne.

Quand le cinquième chapitre et le dernier, il est consacré à l'élaboration de GRAFCET, la programmation avec Step7 et la supervision avec le WinCC de la chaîne de traitement de surface.

Enfin, on termine par une conclusion générale et les perspectives visées en termes d'améliorer l'installation.

Chapitre I :

Automate Programmable Industriel

I.1. Introduction

L'automate programmable industriel API (ou **Programmable Logic Controller PLC**) est aujourd'hui le constituant le plus répandu des automatismes. On le trouve non seulement dans tous les secteurs de l'industrie, mais aussi dans les services. Il répond aux besoins d'adaptation et de flexibilité de nombres d'activités économiques actuelles. Cette place majeure soulève bien sûr un certain nombre de questions [1]. C'est à ces questions que nous allons essayer de répondre ici, en mettant en évidence

- Ses caractéristiques propres, matérielles et logicielles.
- Sa capacité à s'intégrer dans un ensemble plus large, et donc à répondre aux besoins d'un système automatisé de production SAP.

I.2. Historique

Les automatismes séquentiels ont été réalisés, depuis longtemps, à base de relais **électromagnétique**s. L'inconvénient c'est qu'il s'agit d'un système câblé ce qui impose la refonte complète du câblage et ceci pour la moindre modification dans l'ordonnancement des séquences. En 1966, l'apparition **des relais statiques** a permis de réaliser des divers modules supplémentaires tel que le comptage, la temporisation, le pas à pas ... Cependant cette technologie avait le même problème : technologie câblée.

En 1968 et à la demande de l'industrie automobile nord-américaine, sont apparus les premiers dispositifs de commande logique aisément modifiable : Les PLC (Programmable Logic Controller) par Allen Bradley, Modicom et Digital Equipement. Le premier dispositif français était le PB6 de Merlin Gerin en 1973.

I.3. 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 pré actionneurs et d'actionneurs à partir d'informations logique, analogique ou numérique.

Norme NFC 63-850 : « Appareil électronique qui comporte une mémoire programmable par un utilisateur automaticien (et non informaticien) à l'aide d'un langage adapté, pour le stockage interne des instructions composant les fonctions d'automatisme comme :

- Logique séquentiel et combinatoire.
- Temporisation, comptage, décomptage, comparaison.
- Calcul arithmétique.
- Réglage, asservissement, régulation, etc.
- Pour commander, mesurer et contrôler au moyen de modules d'entrées et de sorties (logiques, numériques ou analogiques) différentes sortes de machines ou de processus, en environnement industriel.

I.4. Structure d'un système automatisé

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

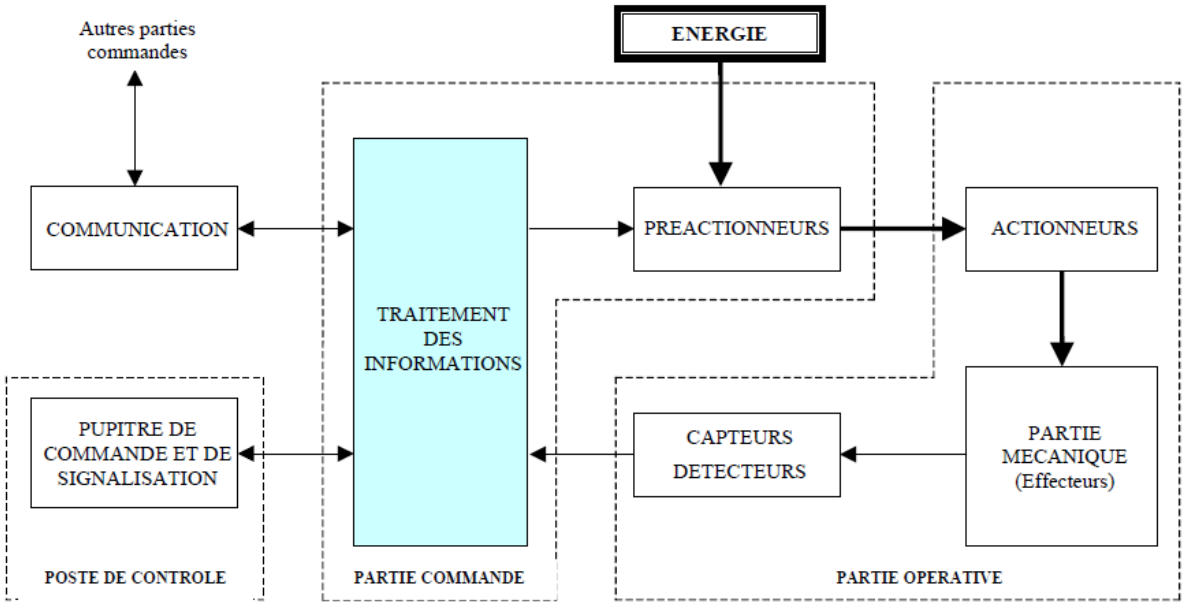


Figure I.1. Structure d'un système automatisé [2].

I.4.1. 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.

I.4.2. Partie commande

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

I.4.3. 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).

I.5. Les types des automates

Les automates peuvent être de type **micros automate**, **compact** ou **modulaire [4] :**

I.5.1. Les micros automates

Sont, comme leur nom l'indique, de toutes petites unités avec une structure fixe comprenant de 4 à 20 entrées-sorties, généralement tout-ou-rien. Ils sont utilisés pour réaliser de petits automatismes autonomes en logique combinatoire. Généralement ils se programment avec un langage simplifié qui leur est propre.

I.5.2. Les automates compacts

Ils intègrent le processeur, l'alimentation, les entrées et les sorties. Selon les modèles et les fabricants, sont des appareils avec un nombre fixe d'entrées-sorties digitales et analogiques. Ils sont cependant extensibles par blocs jusqu'à environ 250 entrées-sorties. Ils sont principalement exploités pour des applications de complexité moyenne avec de la logique séquentielle et un traitement limité des fonctions analogiques.

I.5.3. Les automates modulaires

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) .sont des machines rapides et puissantes qui travaillent avec des processeurs performants. Ce sont de véritables ordinateurs multitâches et multiprocesseurs. Une CPU peut traiter plus de 8000 entrées-sorties.

Figure I.2. Micro automate (theben)

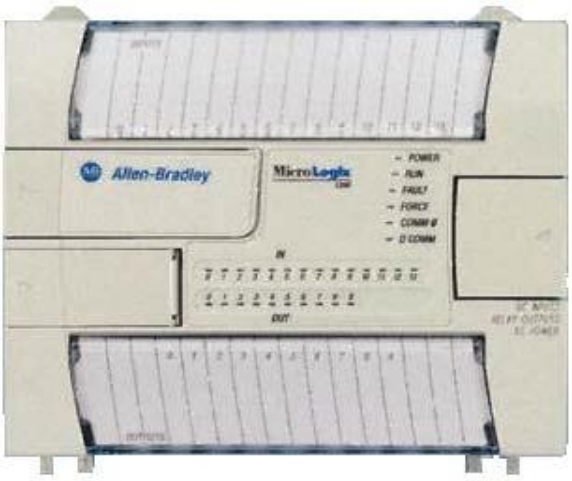


Figure I.3. Automate compact (Allen-bradley)

Figure I.4. Automate modulaire (Modicon)

I.6. Architecture des API :

I.6.1. Le Processeur

Le processeur a pour rôle principal le traitement des instructions qui Constituent le programme de fonctionnement de l'application. Mais en dehors De cette tâche de base, il réalise également d'autres fonctions :

- Gestion des entrées/sorties.
- surveillance et diagnostique de l'automate par une série de tests lancés à la mise sous tension ou cycliquement en cours de fonctionnement.
- Dialogue avec le terminal programmation aussi bien pour l'écriture et la mise au point du programme qu'en cours d'exploitation pour des réglages ou des vérifications de données.

Le processeur est organisé autour d'un certain nombre de registres, ce sont des mémoires rapides permettant la manipulation des informations qu'elles retiennent, ou leur combinaison avec des informations extérieures

Les principaux registres existants dans un processeur sont :

I.6.1.1. L'accumulateur

C'est le registre où s'effectuent les opérations du jeu d'instruction, les résultats sont contenus dans ce registre spécial.

I.6.1.2. Le registre d'instruction

Il reçoit l'instruction à exécuter et décode le code opération. Cette instruction est désignée par le pointeur .

I.6.1.3. Le registre d'adresse

Ce registre reçoit, parallèlement au registre d'instruction, la partie opérande de l'instruction. Il désigne le chemin par lequel circulera l'information lorsque le registre d'instruction validera le sens et ordonnera le transfert.

I.6.1.4. Le registre d'état

C'est un ensemble de positions binaires décrivant, à chaque instant, la situation dans laquelle se trouve précisément la machine.

I.6.1.5. Les piles

Une organisation spéciale de registres constitue une pile, ces mémoires sont utilisées pour contenir le résultat de chaque instruction après exécution.

Ce résultat sera utilisé ensuite par d'autre instruction, et cela pour faire place à la nouvelle information dans l'accumulateur.

I.6.2. Les mémoires

Un système à processeur est toujours accompagné d'un ou de plusieurs types de mémoires. Les automates programmables industriels possèdent pour la plupart les mémoires suivantes :

I.6.2.1. Mémoire de travail

La mémoire de travail (mémoire vive) contient les parties du programme significatif pour son exécution. Le traitement du programme a lieu exclusivement dans la mémoire de travail et dans la mémoire système.

I.6.2.2. Mémoire système

La mémoire système (mémoire vive) contient les éléments de mémoire que chaque CPU met à la disposition du programme utilisateur comme, par exemple, mémoire image des entrées, mémoire image des sorties, mémentos, temporisation et compteurs. La mémoire système contient, en outre, la pile des blocs et la pile des interruptions.

Elle fournit aussi la mémoire temporaire allouée au programme (pille des données locales).

I.6.2.3. Mémoire de chargement

La mémoire de chargement sert à l'enregistrement du programme utilisateurs sans affectation de **mnémonique** ni de commentaires.

La mémoire de chargement peut être soit une mémoire vive (RAM), soit une mémoire EPROM.

I.6.2.4. Mémoire RAM non volatile

Zone de mémoire configurable pour sauvegarder des données en cas de défaut d'alimentation.

I.6.2.5. Mémoire ROM

Contient le système d'exploitation qui gère la CPU.

I.6.3. Modules d'entrées/sorties

Les modules d'entrées et sorties assurent le rôle d'interface entre le procédé à commander et la CPU.

I.6.3.1. Entrées/sorties TOR (Tout ou Rien)

La gestion de ce type de variables constituant le point de départ historique des API reste une de leurs activités majeures.

Leur nombre est en général de 8,16, 24 ou 32 entrées/sorties, qui peuvent fonctionner :

- En continu : 24V, 48V.
- En alternatif : 24V, 48V, 100/120V, 210/240V

I.6.3.2. Entrées/Sorties analogique

Elles permettent l'acquisition de mesures (entrées analogique), et la commande (sorties analogiques). Ces modules comportent un ou plusieurs convertisseurs Analogique/Numérique (A/N) pour les entrées et Numérique / Analogique (N/A) pour les sorties dont la résolution est de 8 à 16 bits.

Les standards les plus utilisés sont : ± 10V, 0-10V, ±20mA, 0-20mA et 4-20mA.

Ces modules sont en général multiplexés en entrée pour n'utiliser qu'un seul convertisseur A/N, alors que les sorties exigent un convertisseur N/A par voie pour pouvoir garder la commande durant le cycle de l'API.

I.6.4. L'alimentation électrique

Elle a pour rôle de fournir les tensions continues nécessaires aux composants avec de bonnes performances, notamment face aux microcoupures du réseau électrique qui constitue la source d'énergie principale.

La tension d'alimentations peut être de 5V, 12V ou 24V.

D'autres alimentations peuvent être nécessaires pour les châssis d'extension et pour les modules entrées/sorties. Un onduleur est nécessaire pour éviter les risques de coupures non tolérées.

I.6.5. Les liaisons

Elles s'effectuent :

- avec l'extérieur par des **borniers**, sur lesquels arrivent des câbles transportant les signaux électriques.
- avec l'intérieur par des **bus**, liaisons parallèles entre les divers éléments. Il existe plusieurs types de bus, car on doit transmettre des données, des états, des adresses.

I.6.6. Eléments auxiliaires

Un ventilateur est indispensable dans les châssis comportant de nombreux modules ou dans le cas où la température ambiante est susceptible de devenir assez élevée.

Un support mécanique : il peut s'agir d'un rack, l'automate se présente alors.

Sous forme d'un ensemble de cartes, d'une armoire, d'une grille, et des fixations correspondantes.

Des indicateurs d'état : concernant la présence de tension, la charge de la batterie, le bon fonctionnement de l'automate etc.

I.7. Les langages de programmation

La norme de la Commission Électrotechnique Internationale (IEC 1131-3) définit cinq langages qui peuvent être utilisés pour la programmation des automates programmables industriels. Ces langages peuvent être divisés en deux catégories [3].

I.7.1. Langages graphiques

- SFC « Sequential Funiculite Chart » ou GRAFCET .
- LD « Ladder Diagram » ou schéma à relais .
- FBD « Function Block Diagram » ou schéma par bloc.

I.7.2. Langages textuels

- ST « structured text » ou texte structuré .
- IL « Instruction List » ou liste d'instructions.

I.8. Protection de l'automate

La protection des circuits d'entrée contre les parasites électriques est souvent résolue par découplage optoélectrique. Le passage des signaux par un stade de faisceau lumineux assure en effet une séparation entre les circuits internes et externes.

Du coté sorties, on doit assurer le même type de protection, mais amplification de puissance, avec au final un courant continu ou alternatif selon les cas.

Deux types de cartes électroniques sont utilisés :

I.8.1. Les modules à sorties statiques

Relais statiques intégrant des composants spécialisés : transistor bipolaires, thyristors. Ces composants n'ont aucune usure mécanique et leurs caractéristiques de commutation se maintiennent dans le temps.

I.8.2. Les modules à relais électromagnétique

Où le découplage résulte de l'existence de deux circuits électriques (Bobines d'excitation, circuits de puissance), ces relais électromagnétique ont l'avantage d'avoir une faible résistance de contact, une faible capacité de sortie et surtout un faible coût, mais une durée de vie et une vitesse de commutation inférieures aux sorties statiques.

I.9. Conclusion

Ce chapitre consiste à décrire d'une manière globale l'automate programmable industriel, son rôle dans un système de production automatisé et son principe de fonctionnement en présentant l'architecture interne et externe.

Chapitre II :
GRAFCET et GEMMA

II.1. Introduction

Dans ce chapitre, nous donnons une présentation générale sur le GRAFCET, un outil graphique de représentation du cahier des charges d'un automatisme séquentiel, puis GEMMMA qui permet de prévoir une hiérarchisation générale de grafcet et toutes les situations de marche et d'arrêt.

II.2. Présentation du GRAFCET

Le GRAFCET (**GRA**phe **F**onctionnel de **C**ommande **E**tape **T**ransition.) est un outil graphique. Il est basé sur les notions d'**étapes** auxquelles sont associées des **actions** et de **transitions** auxquelles sont associées des **réceptivités**. Il décrit les ordres émis par la partie commande vers la partie opérative en mettant en évidence les actions engendrées et les événements qui les déclenchent. Cette représentation est étroitement liée à la notion d'évolution du processus [5].

Pour parvenir à la réalisation de la partie commande, il est conseillé de diviser le cahier des charges en deux niveaux successifs et complémentaires [6].

Le premier niveau décrit le **comportement de la partie commande** en fonction de l'évolution de la partie opérative : c'est le rôle des spécifications fonctionnelles décrivant ce que doit faire l'automatisme. À ce niveau, les contraintes technologiques (des actionneurs, des capteurs…) n'interviennent pas, seules comptent les fonctionnalités.

Le second niveau renseigne sur la **nature technologique des actionneurs et des capteurs**; on y trouve donc leurs caractéristiques et les contraintes qui y sont associées. C'est aussi à ce niveau que les spécifications opérationnelles décrivant les conditions d'utilisation de l'application apparaissent.

L'automaticien, confronté à un problème de conception et de réalisation d'un automatisme, aborde donc l'étude en deux phases successives correspondant aux deux niveaux de spécification :

- un niveau fonctionnel.
- un niveau technologique.

II.3. Les éléments graphiques

Le GRAFCET est composé d'étapes, de réceptivités, d'actions, de transitions et de liaisons (Figure.II.1).

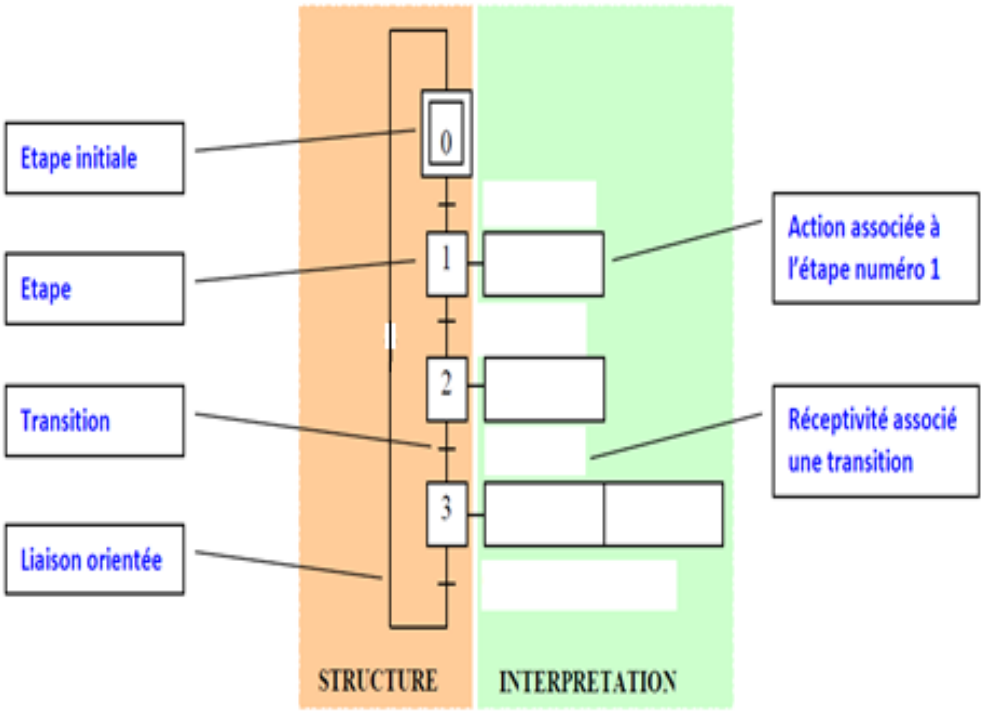


Figure II.1. Les constituants d'un grafcet

- **Une LIAISON** est un arc orienté (ne peut être parcouru que dans un sens). A une extrémité d'une liaison il y a une étape, à l'autre une transition. On la représente par un trait plein rectiligne, vertical ou horizontal.

- **Une ETAPE** correspond à une phase durant laquelle on effectue une ACTION pendant une certaine durée. On numérote chaque étape par un entier positif, mais pas nécessairement croissant par pas de 1, il faut simplement que jamais deux étapes différentes n'aient le même numéro. Une étape est dite active lorsqu'elle correspond à une phase "en fonctionnement", c'est à dire qu'elle effectue l'action qui lui est associée. On représente quelquefois une étape active à un instant donné en dessinant un point à l'intérieur.

- **Une TRANSITION** est une condition de passage d'une étape à une autre. Elle n'est que logique (dans son sens Vrai ou Faux), sans notion de durée. La condition est définie par une RECEPTIVITE qui est généralement une expression booléenne (c.à.d. avec des ET et des OU) de l'état des capteurs.

- **ACTION** : associée à une étape, une action n'est commandée que lorsque l'étape est active. On parle d'assignation sur état (en mode continu), ou d'affectation sur événement (en mode mémorisé)

- **RECEPTIVITE** : équation booléenne logique associée à une transition. C'est une fonction logique des entrées, de variables auxiliaires et/ou de l'activité d'étapes. Elle permet de distinguer parmi toutes les variables du système, celles qui sont susceptibles de faire évoluer la partie commande par franchissement d'une transition.

II.4. Les règles d'évolution

Règle 1 : Situation initiale

La situation initiale du grafcet caractérise le comportement initial de la partie commande vis-à-vis de sa partie opérative. Elle correspond aux étapes actives au début du fonctionnement, soit à la mise en énergie de la partie commande.

Règle 2 : Franchissement d'une transition

Une transition est validée si toutes les étapes immédiatement précédentes sont actives. L'évolution du grafcet correspond au franchissement d'une transition qui se produit sous deux conditions :

- si cette transition est validée
- si la réceptivité associée à cette transition est vraie

Si ces deux conditions sont réunies, la transition devient franchissable et est obligatoirement franchie.

Règle 3 : Evolution des étapes actives

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

Règle 4 : Evolutions simultanées

Plusieurs transitions simultanément franchissables sont simultanément franchies.

Règle 5 : Activations et désactivations simultanées

Si, au cours du fonctionnement, une même étape doit être désactivée et activée simultanément, elle reste active.

II.5. Les structures de base

Les structures de base sont des configurations de grafcets associées aux concepts de base des systèmes logiques. Elles permettent d'exprimer notamment des successions d'états, des sélections entre plusieurs séquences, des parallélismes de séquences, des sauts et des reprises de séquences, des partages de ressources, des couplages entre séquences [6].

II.5.1 Séquence unique

C'est une suite d'étapes pouvant être activées les unes après les autres.

II.5.2 Séquences simultanées et alternatives

La sélection de séquences exprime un choix d'évolution entre plusieurs séquences à partir d'une ou de plusieurs étapes. Cette structure se représente par autant de transitions validées simultanément qu'il peut y avoir d'évolutions possibles.

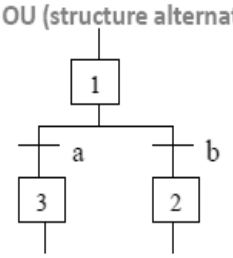
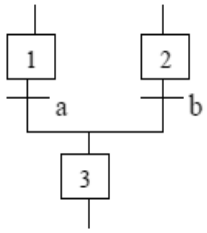
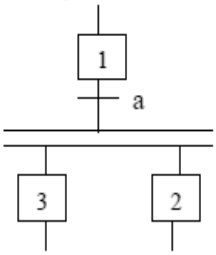
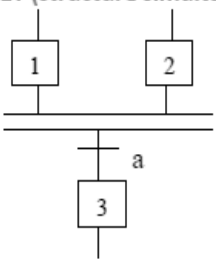
Divergence en OU (structure alternative) :  Si 1 active et si a seul, alors désactivation de 1 et activation de 3, 2 inchangé. Si a et b puis 1 active alors désactivation 1, activation 2 et 3 quel que soit leur état précédent. (règle 4)	Convergence en OU (structure alternative) :  Si 1 active et a sans b, alors activation de 3 et désactivation de 1, 2 reste inchangé Si 1 et 2 et a et b alors 3 seule active
Divergence en ET (structure simultanée) :  Si 1 active et si a, alors désactivation de 1 et activation de 2 ET 3.	Convergence en ET (structure simultanée) :  Si 1 active seule et a, alors aucun changement. Si 1 ET 2 et a, alors activation de 3 et désactivation de 1 et 2.

Figure II.2. Exemple sur les Séquences simultanées et alternatives

II.5.3. Saut d'étapes et reprise de séquence

Le saut d'étape(s) est un cas particulier de sélection de séquences. Elle permet soit de parcourir la séquence complète, soit de sauter une ou plusieurs étapes de la séquence lorsque, par exemple, les actions associées à ces étapes deviennent inutiles.

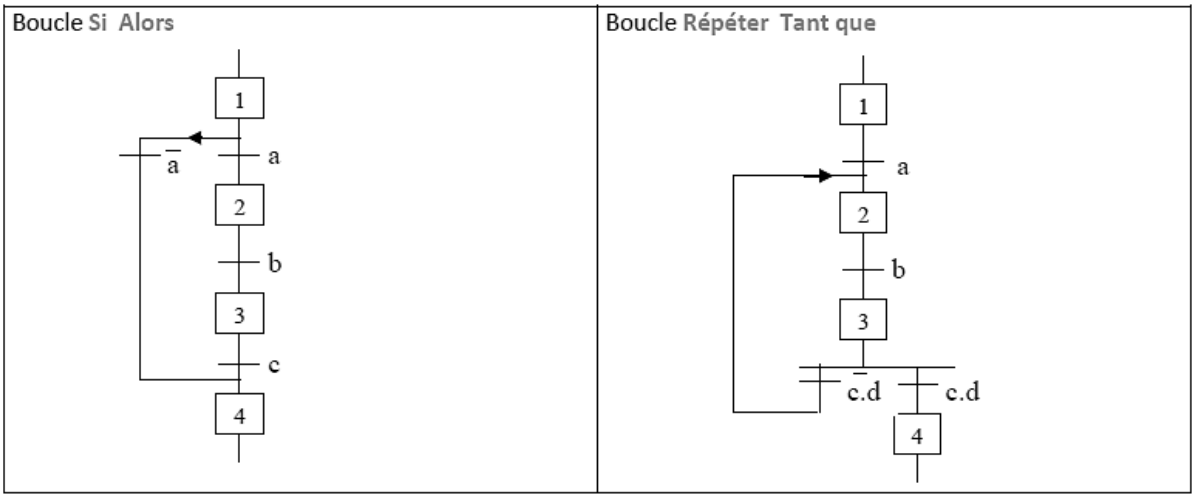


Figure II.3. Exemple sur sauts d'étapes

II.6. Présentation du GEMMA

Le GEMMA (GUIDE **D'ETUDE DES MODES DE MARCHE ET D'ARRET)** a été élaboré par l'ADEPA (Agence nationale pour le Développement de la Production Automatisée), C'est un document graphique qui facilite la conduite, la maintenance et l'évolution du système. Ce document est constitué de rectangles d'état appelés modes.

Ces rectangles sont reliés entre eux par des liaisons orientées. Le passage d'un rectangle à l'autre s'effectue un peu à la manière du franchissement d'une transition de Grafcet. Le GEMMA n'est pas un outil figé, il est modifiable à volonté en fonction des spécifications à obtenir. Les liaisons orientées présentes sur certains documents de référence ne sont là qu'à titre indicatif.

Le guide graphique GEMMA, est constitué de deux zones :

- une zone correspondant à l'état « hors énergie ».
- une zone permettant de décrire ce qui se passe lorsque la partie commande (PC) fonctionne normalement « sous énergie ».

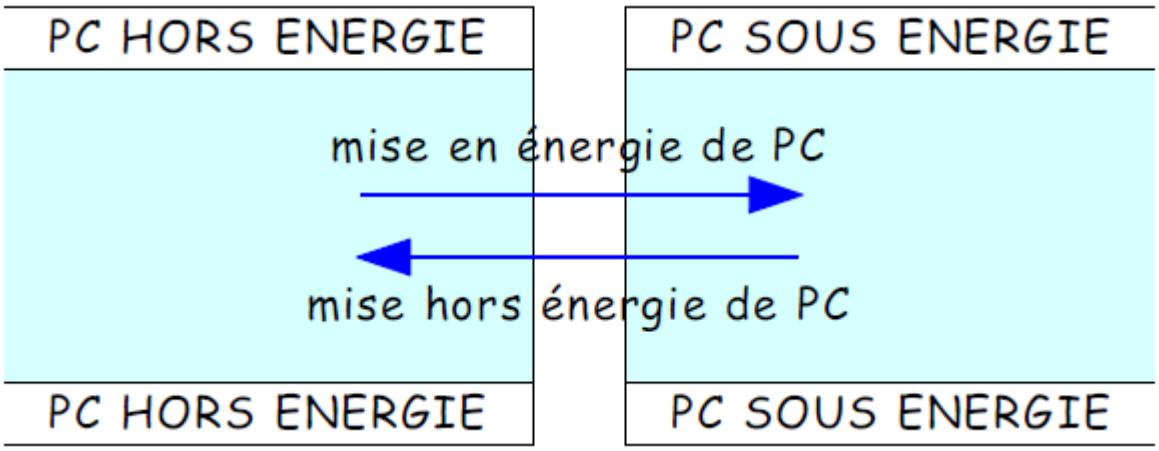


Figure 1I.4. La structure générale de GEMMA.

II.6.1. Partie commande hors énergie (PZ)

Cette zone du GEMMA, située à l'extrême gauche, correspond à l'état inopérant de la partie commande. Dans cet état la partie opérative n'est pas sous le contrôle de la partie commande. La partie opérative peut être en énergie ou hors énergie. La sécurité est garantie par les choix technologiques. En d'autres termes, dans cette partie il n'y a pas de modes traités par la partie commande.

II.6.2. Partie commande en énergie et active

C'est la partie qui va permettre de définir les différents modes de marche et d'arrêt du système ainsi que les conditions de passage d'un mode à l'autre. Cette partie est subdivisée en trois zones ou en trois familles de procédures :

II. 6.2.1. Famille A : procédures d'arrêt

On classera dans cette famille tous les modes conduisant à (ou traduisant) un état d'arrêt du système pour des raisons extérieures.

II. 6.2.1.1. A1 « Arrêt dans l'état initial »

C'est l'état « repos » de la machine. Il correspond en général à la situation initiale du GRAFCET : c'est pourquoi, comme une étape initiale, ce « rectangle-état » est entouré d'un double cadre.

Pour une étude plus facile de l'automatisme, il est recommandé de représenter la machine dans cet état initial.

II. 6.2.1.2. A2 « Arrêt demandé en fin de cycle »

Lorsque l'arrêt est demandé, la machine continue de produire jusqu'à la fin du cycle. A2 est donc un état transitoire vers l'état A1.

II. 6.2.1.3. A3 « Arrêt demandé dans un état déterminé »

La machine continue de produire jusqu'à un arrêt en une position autre que la fin de cycle : c'est un état transitoire vers A4.

II. 6.2.1.4. A4 « Arrêt obtenu »

La machine est alors arrêtée en une autre position que la fin du cycle.

II. 6.2.1.5. A5 « Préparation pour remise en route après défaillance »

C'est dans cet état que l'on procède à toutes les opérations (dégagements, nettoyages…) nécessaires à une remise en route après défaillance.

II. 6.2.1.6. A6 « Mise P.O. dans état initial »

La machine étant en A6, on remet manuellement ou automatiquement la Partie Opérative en position pour un redémarrage dans l'état initial.

II. 6.2.1.7. A7 « Mise P.O. dans état déterminé »

La machine étant en A7, on remet la P.O. en position pour un redémarrage autre que l'état initial.

II. 6.2.2. Famille F : procédures de Fonctionnement

On groupe dans cette famille tous les modes ou états qui sont indispensables à l'obtention de la valeur ajoutée, ou, autrement dit, tous ceux sans lesquels on ne peut pas techniquement ou fonctionnellement obtenir la valeur ajoutée pour laquelle la machine est prévue.

Les modes préparatoires à la production, de réglages ou, de tests, peuvent faire partie de cette famille.

II. 6.2.2.1. F1 « Production normale »

Dans cet état la machine produit normalement : c'est l'état pour lequel elle a été conçue. On peut souvent faire correspondre à cet état un **GRAFCET de production**.

II. 6.2.2.2. F2 « Marche de préparation »

Cet état est utilisé pour les machines nécessitant une préparation préalable à la production normale : préchauffage de l'outillage, remplissage de la machine, mises en routes diverses, etc.

II. 6.2.2.3. F3 « Marche de clôture »

C'est l'état nécessaire pour certaines machines devant être vidées, nettoyées, etc., en fin de journée ou en fin de série.

II. 6.2.2.4. F4 « Marche de vérification dans le désordre »

Cet état, permet de vérifier certaines fonctions ou certains mouvements sur la machine, sans respecter l'ordre du cycle.

II. 6.2.2.5. F5 « Marche de vérification dans l'ordre »

Dans cet état, le cycle de production peut être explorée au rythme voulu par la personne effectuant la vérification ; selon le cas, la machine produit ou ne produit pas.

II. 6.2.2.6. F6 « Marche de test »

Les machines de contrôle, de mesure, de tri… comportent des capteurs qui doivent être réglés ou étalonnés périodiquement : la « Marche de test » F6 permet ces opérations de réglage ou d'étalonnage.

II. 6.2.3. Famille D : procédures de Défaillance

Il est rare qu'un système fonctionne sans incident pendant toute sa vie : il est indispensable de prévoir les défaillances.

On regroupera dans cette famille tous les modes conduisant à (ou traduisant) un état d'arrêt du système pour des **raisons intérieures** au système, autrement dit, à cause de défaillances de la partie opérative

II. 6.2.3.1. D1 « Arrêt d'urgence »

C'est l'état pris lors d'un arrêt d'urgence : on y prévoit non seulement les arrêts, mais aussi les cycles de dégagement, les procédures et précautions nécessaires pour éviter ou limiter les conséquences dues à la défaillance.

II. 6.2.3.2. D2 « Diagnostic et/ou traitement de défaillance »

C'est dans cet état que la machine peut être examinée après défaillance et qu' 'il peut être apporté un traitement permettant le redémarrage.

II. 6.2.3.3. D3 « Production tout de même »

Il est parfois nécessaire de continuer la production même après défaillance de la machine : on aura alors une « production dégradée », ou une « production forcée », ou une production aidée par des opérateurs non prévue en « Production normale ».

II.7. Les « rectangles-états »

Sur le guide graphique GEMMA, chaque mode de marche ou d'arrêt désiré peut être décrit dans l'un des « rectangles-états » prévu à cette fin.

Position du « rectangle-état »

La position d'un « rectangle-état » sur le guide graphique définit :

- son appartenance à l'une des trois familles : procédures de fonctionnement, d'arrêt ou de défaillance.
- le fait qu'il soit « en » ou « hors » production.

Utilisation d'un « rectangle-état »

Un « rectangle-état » retenu se complète de façon manuscrite : en précisant l'opération exécutée propre à la machine étudiée ; en surlignant la (les) liaison(s) orientée(s) ; en indiquant la (les) condition(s) d'évolution.

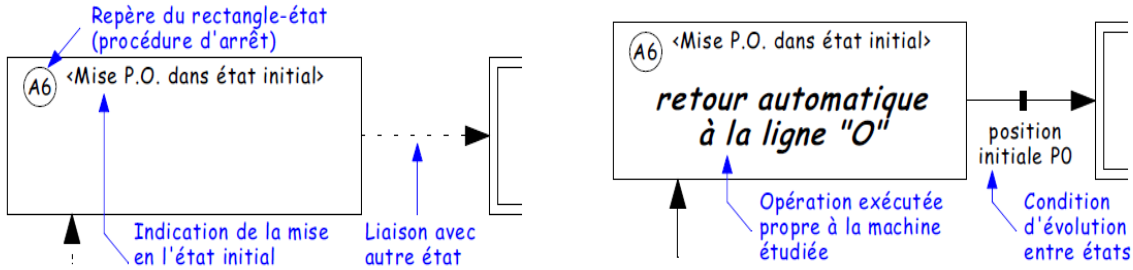


Figure. II.5. les constituants de rectangle état du GEMMA

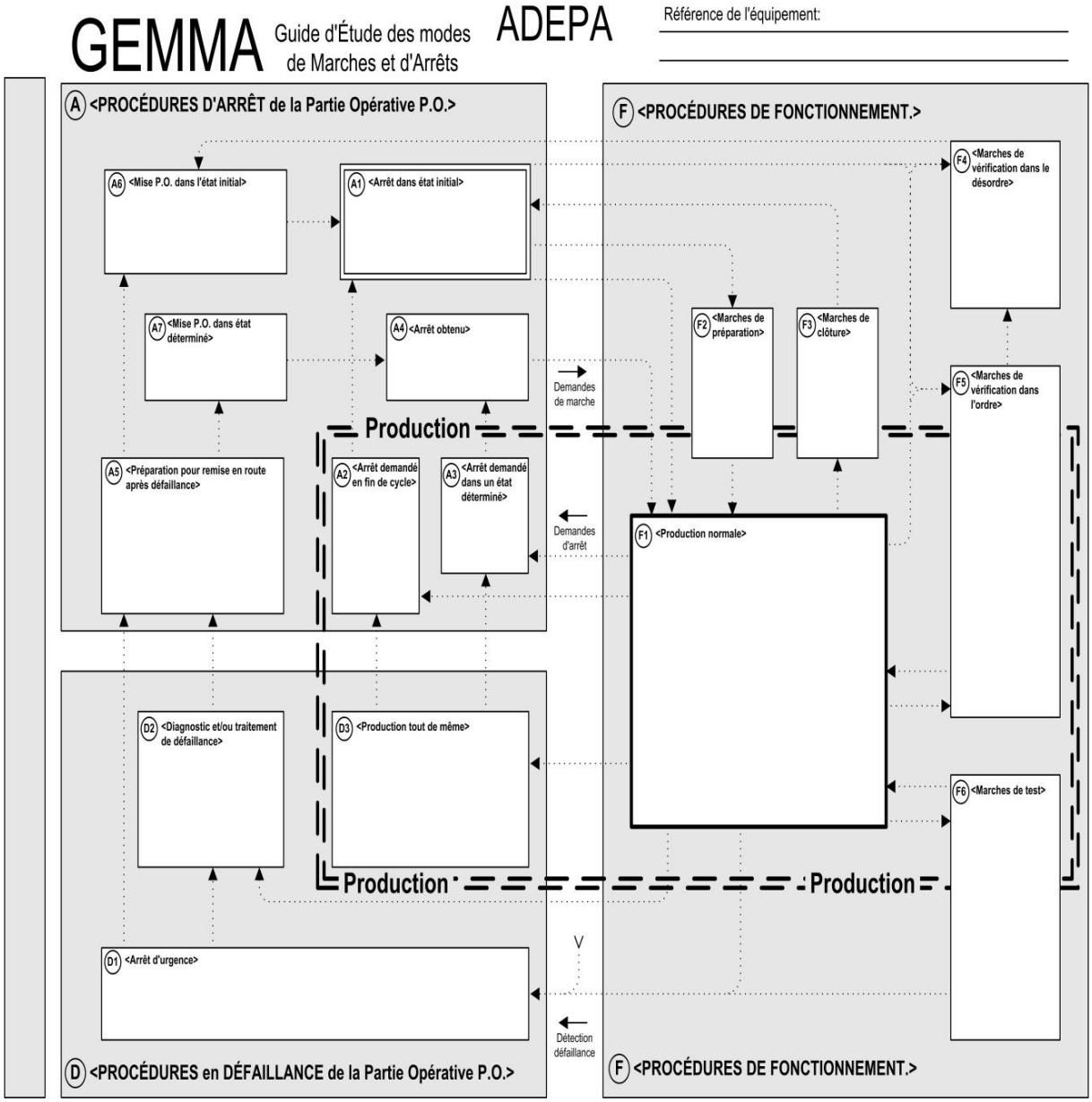


Figure II.6. Guide pratique du GEMMA (d'après ADEPA)

II.8.Conclusion

Dans ce chapitre, nous avons donné les principales règles pour traduire un cahier de charge en un GRAFCET.

En réalité, il est nécessaire de prendre en compte le comportement du système au regard des sécurités ou défaillances et d'une façon plus générale, d'étudier **comment arrêter le système puis comment le redémarrer.** Cette étude peut se réaliser à l'aide de GEMMA qui permet d'élaborer une hiérarchisation de grafcets : **grafcet de sécurité** validant le fonctionnement d'un **grafcet des modes de marches** ou **grafcet de conduite** faisant lui-même appel à un ou plusieurs **grafcets de production.**

Chapitre III :

Step7, Win CC, Logiciels de programmation et de supervision

III.1.Introduction

Dans ce chapitre, nous donnons une présentation générale sur le logiciel de programmation des automates programmables SIEMENS avec STEP7, ainsi qu'une présentation du simulateur PLC-CIM qui est une application du STEP7 et le logiciel de conception des interfaces HMI (Homme Machine) avec le WinCC flexible. Ce dernier permet la supervision de processus.

III.2.Présentation générale du logiciel STEP7

III.2.1. Définition du logiciel

Le STEP 7 est le progiciel de base pour la configuration et la programmation des systèmes d'automatisation SIMATIC et qui s'exécute sous un environnement Windows à partir d'une console de programmation PG ou d'un PC.

III.2.2. Application du logiciel STEP7

Le logiciel STEP7 met à disposition les applications de base suivantes :

III.2.2.1Gestionnaire de projets SIMATIC

Le gestionnaire de projets SIMATIC gère toutes les données relatives à un projet d'automatisation − quel que soit le système cible (S7/M7/C7) sur lequel elles ont été créées. Le gestionnaire de projets SIMATIC démarre automatiquement les applications requises pour le traitement des données sélectionnées [7].

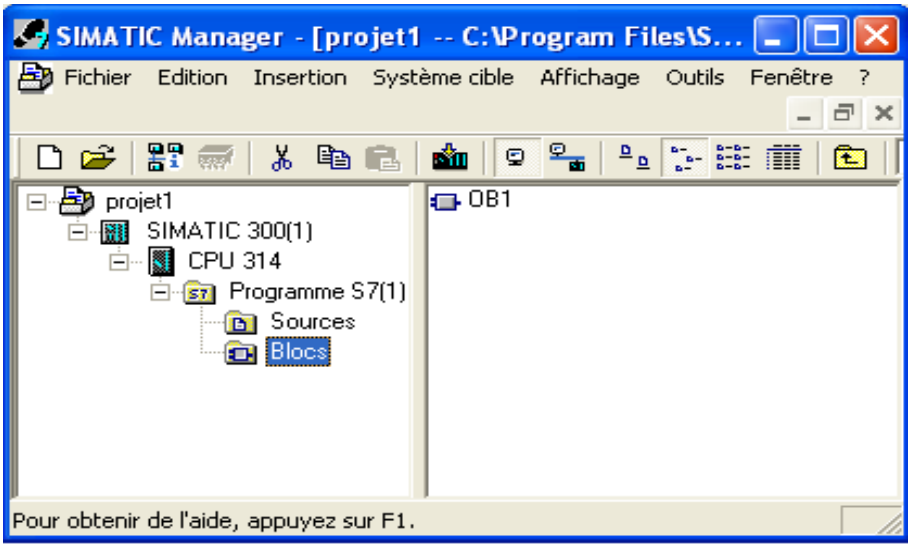


Figure III.1. Fenêtre de gestionnaire de projet SIMATIC

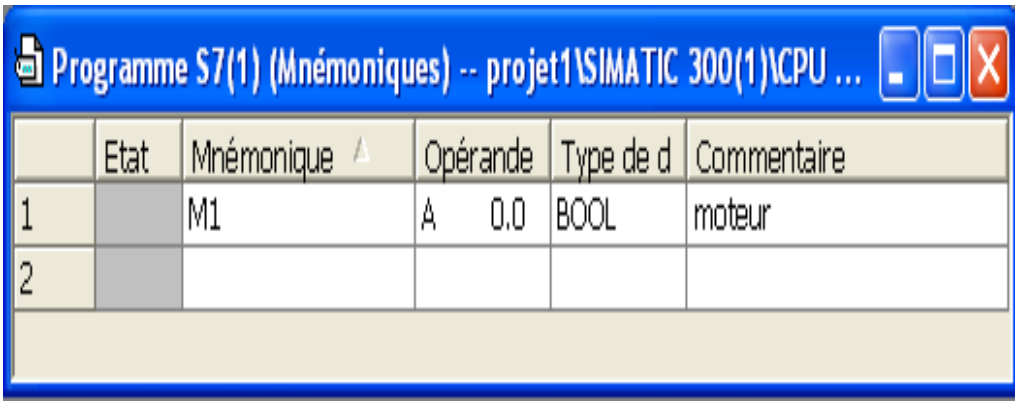
III.2.2.2. Editeur de mnémoniques

L'éditeur de mnémoniques nous permet de gérer toutes les variables globales(les signaux du processus (entrées/sorties), mémentos et blocs) ainsi leurs désignations symboliques et de commentaires.

Pour accéder à la table des mnémoniques, on click sur le dossier programme dans la fenêtre du projet, puis sur l'icône mnémoniques.

L'utilisation de cette table consiste à :

- Donner un nom à la mnémonique dans la première colonne.
- Donner la variable associée à cette mnémonique dans la seconde colonne.
- Le type de la donnée est automatiquement généré par STEP7.
- Ecrire éventuellement un commentaire dans la colonne prévue à cet effet.



	Etat	Mnémonique	Opérande	Type de d	Commentaire
1		M1	A 0.0	BOOL	moteur
2					

Figure III.2. La table de Mnémoniques

III.2.2.3. Diagnostic du matériel

Le diagnostic du matériel fournit un aperçu de l'état du système d'automatisation. Dans une représentation d'ensemble, un symbole permet de préciser pour chaque module, s'il est défaillant ou pas. Un double clic sur le module défaillant permet d'afficher des informations détaillées sur le défaut [7].

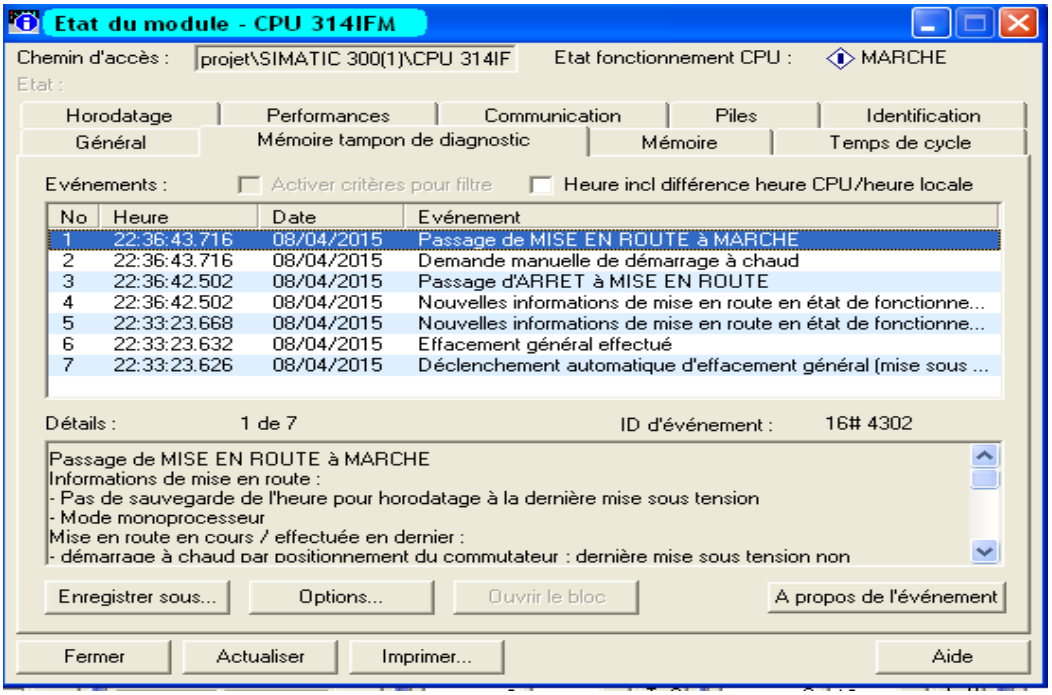


Figure III.3. La fenêtre d'état de module pour le diagnostic

III.2.2.4. Configuration matérielle

La configuration matérielle est une étape très importante, elle permet de reproduire à l'identique le système utilisé (châssis (Rack), alimentation, CPU, modules d'entrées /sorties).

III.2.2.5. Langages de programmation

Les langages de programmation CONT, LIST et LOG pour S7-300/400 font partie intégrante du logiciel de base.

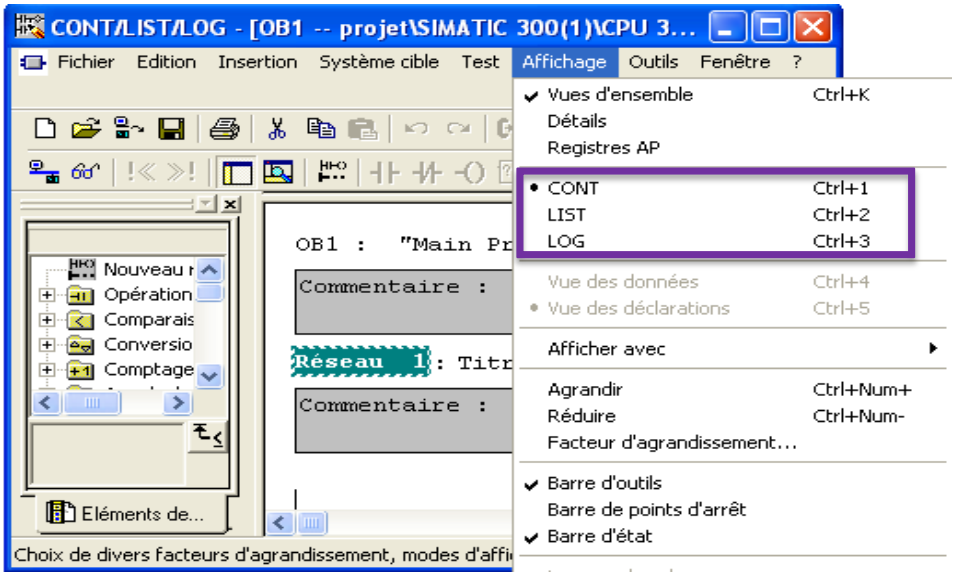


Figure III.4. Les langages de programmation intégrés dans Step7.

III.3. Elaboration du programme sous STEP7

III.3.1. Démarrage du logiciel STEP7

Pour lancer le logiciel STEP7, on localise l'icône SIMATIC Manager sur l'écran de l'ordinateur puis avec un double click sur cette icône, on se permet d'ouvrir sa fenêtre fonctionnelle.

III.3.2. Création d'un nouveau projet

Dans la fenêtre SIMATIC Manager, cliquer sur Fichier>Nouveau (ou encore CTRL+N), une fenêtre demandant un nom de projet s'ouvre. Il faut donc donner un nom au projet puis valider par OK.

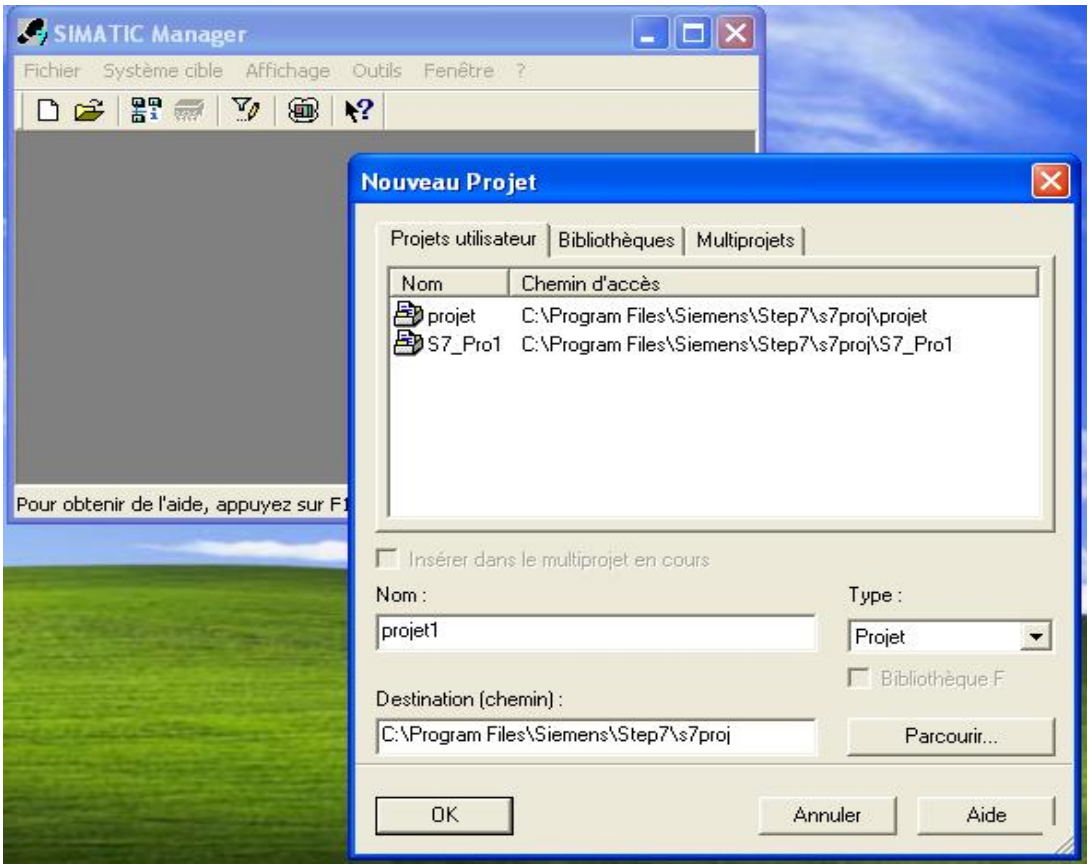


Figure III.5.création de projet

La fenêtre du projet s'ouvre et Le projet est vide, il faut lui insérer une station SIMATIC, cela est possible en cliquant sur le projet avec le bouton droit puis Insérer un nouvel objet puis station SIMATIC 300.

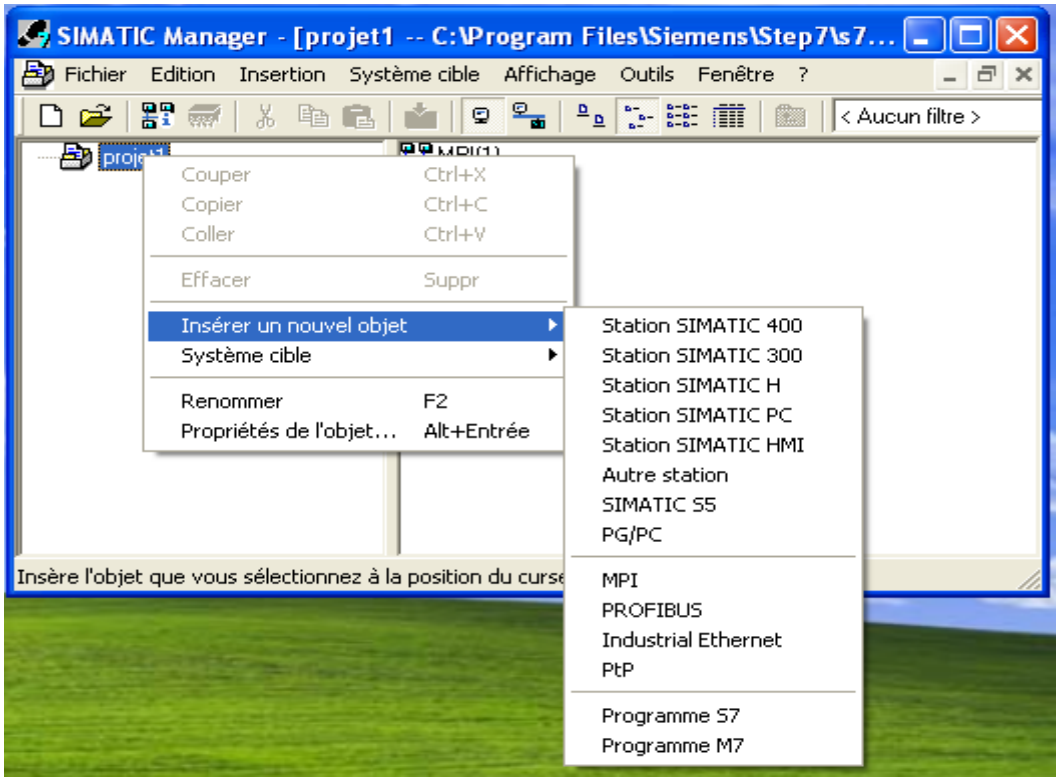


Figure III.6. Insertion d'une station.

La station SIMATIC n'est toujours pas configurée, il faut passer à l'étape de configuration matérielle.

III.3.3. Configuration du matérielle

La configuration du matériel est utilisée pour configurer et paramétrer le support matériel dans un projet d'automatisation.

On click sur l'icône <<STATION SIMATIC 300>> située dans la partie gauche qui contient l'objet <<matériel>>.la fenêtre HW Config s'ouvre

Nous avons tout d'abord besoin d'un châssis ou RACK puis on insère des modules d'alimentation, CPU, d'entrées et de sorties. .

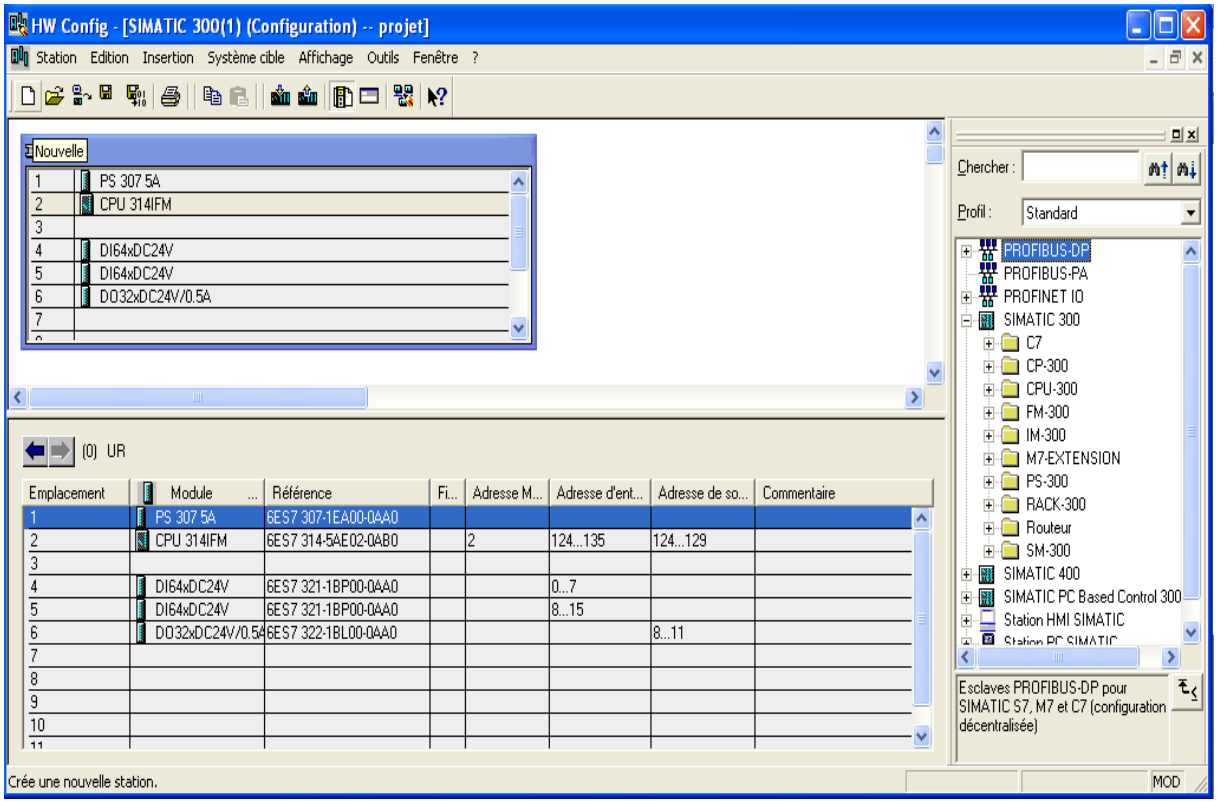


Figure III.7. La fenêtre de configuration de projet.

III.4.Projet et hiérarchie d'objets

Dans SIMATIC Manager, la hiérarchie d'objets pour les projets et bibliothèques est similaire à la structure des répertoires comportant des dossiers et fichiers dans l'explorateur de Windows.

La figure suivante donne un exemple de hiérarchie d'objets.

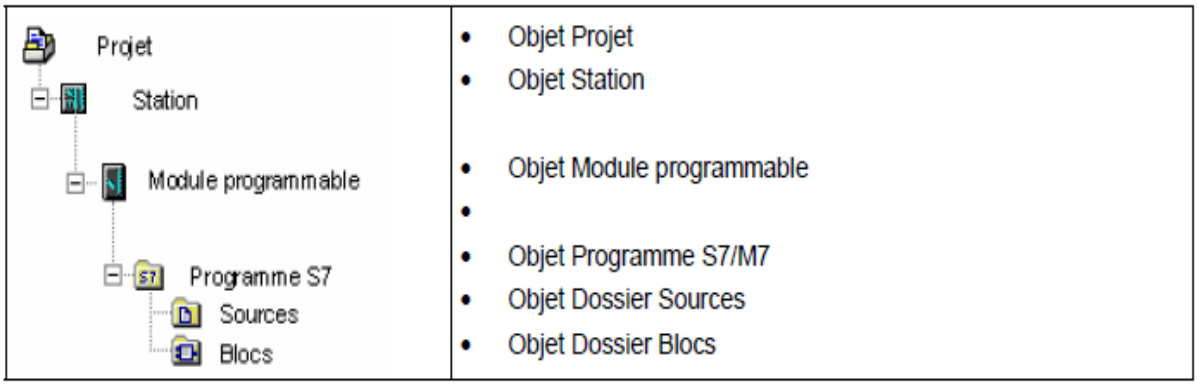


Figure III.8. La hiérarchie d'objet

III.4.1.Objet Projet

Le projet représente l'ensemble des données et programmes d'une solution, se trouve à la tête d'une hiérarchie d'objets.

III.4.2.Objet Station

Une station SIMATIC 300/400 représente une configuration matérielle S7 comportant un ou plusieurs modules programmables.

III.4.3.Objet Module programmable

Un module programmable représente les données de paramétrage d'un module programmable. Les données système de modules ne possédant pas de mémoire rémanente (par exemple CP 441) sont chargées via la CPU de la station. Aucun objet "Données système" n'est de ce fait affecté à de tels modules qui n'apparaissent pas dans la hiérarchie du projet [7].

III.4.4.Objet Programme S7/M7

Un programme (S7/M7) est un dossier contenant les logiciels pour les modules CPU S7/M7 et les logiciels pour les modules autres que les CPU (par exemple modules CP ou FM programmables).

III.4.5.Objet Dossier Sources

Un dossier Sources contient les programmes source sous forme de texte.

III.4.6.Object Dossier Blocs

Le dossier Blocs d'une vue hors ligne peut contenir : des blocs de code (OB, FB, FC, SFB, SFC), des blocs de données (DB), des types de données utilisateur (UDT) et des tables de variables. L'objet Données système représente les blocs de données système.

Le dossier Blocs d'une vue en ligne contient les éléments de programme exécutables, chargés de manière résidente dans le système cible.

III.5.Structure d’un programme utilisateur

Un programme utilisateur est composé de blocs de code et de blocs de données.

III.5.1. Bloc de code

On appelle blocs de code tous les blocs contenant une section d’instructions, c’est-à dire les blocs d’organisation, les blocs fonctionnels et les fonctions.

III.5.1.1.Blocs d'organisation

Les blocs d'organisation (OB) représentent l'interface entre le système d'exploitation et le programme utilisateur. Une tâche précise incombe à chaque bloc d'organisation.

Dans le cas le plus simple, il s'agit des blocs d'organisation destinés :

- à la mise en route (OB100, OB101),
- au programme principal cyclique (OB1)
- au traitement des erreurs (OB80 à OB87, OB121, OB122), dans le cas où une erreur ne doit pas entraîner l'arrêt de votre CPU.

Il existe d'autres blocs d'organisation vous permettant de traiter des alarmes de la

CPU ou du processus.

III.5.1.2.Fonctions et blocs fonctionnels

- **Un bloc fonctionnel (FB)** est un bloc de code avec rémanence. Dans ce cas, la mémoire est un bloc de données d'instance affecté au bloc fonctionnel, dans lequel sont sauvegardés les paramètres effectifs et les données statiques du bloc fonctionnel.
- **Une fonction (FC)** est un bloc de code sans rémanence. Après le traitement des fonctions, les paramètres de sortie contiennent les valeurs de fonction calculées. C'est ensuite à vous d'organiser l'utilisation et la sauvegarde des paramètres effectifs selon vos besoins.

III.5.2.Blocs de données

Les blocs de données mémorisent les données du programme utilisateur. On distingue les blocs de données globaux et les blocs de données d'instance.

- Vous pouvez accéder aux blocs de données globaux à partir de tout endroit du programme utilisateur.
- Les blocs de données d'instance sont affectés à un bloc fonctionnel et contiennent, en plus des données du bloc fonctionnel, aussi les données de multi-instances éventuellement définies. Aussi est-il conseillé d'accéder au bloc de données d'instance uniquement en relation avec ce bloc fonctionnel.

Le système d'exploitation met à votre disposition les données suivantes :

- Entrées et sorties de périphérie
- Mémoire image des entrées et des sorties
- Mémentos
- Temporisations
- Compteurs

Vous pouvez, en outre, définir vos propres données :

- Les données globales valables pour l'ensemble du programme utilisateur peuvent être définies dans les blocs de données.
- Les variables statiques sont uniquement valables dans le bloc fonctionnel dans lequel elles sont définies. A chaque appel de bloc fonctionnel, vous précisez un bloc de données d'instance contenant les données statiques en plus de tous les paramètres. Si des multi-instances sont définies, leurs données d'instance sont insérées avec leurs données statiques dans le bloc de données d'instance.
- Les données temporaires sont définies lors de la création de blocs de code. Elles occupent uniquement de l'espace mémoire dans la pile pendant le traitement du bloc de code.

III.6. Traitement de programme

III.6.1. Traitement de programme cyclique

Le traitement de programme cyclique constitue le traitement normal pour les automates programmables. Ceci signifie que le système d'exploitation parcourt une boucle de programme (le cycle) et appelle le bloc d'organisation OB1 dans le programme principal une fois par boucle. Le programme utilisateur dans le bloc OB1 est donc exécuté cycliquement.

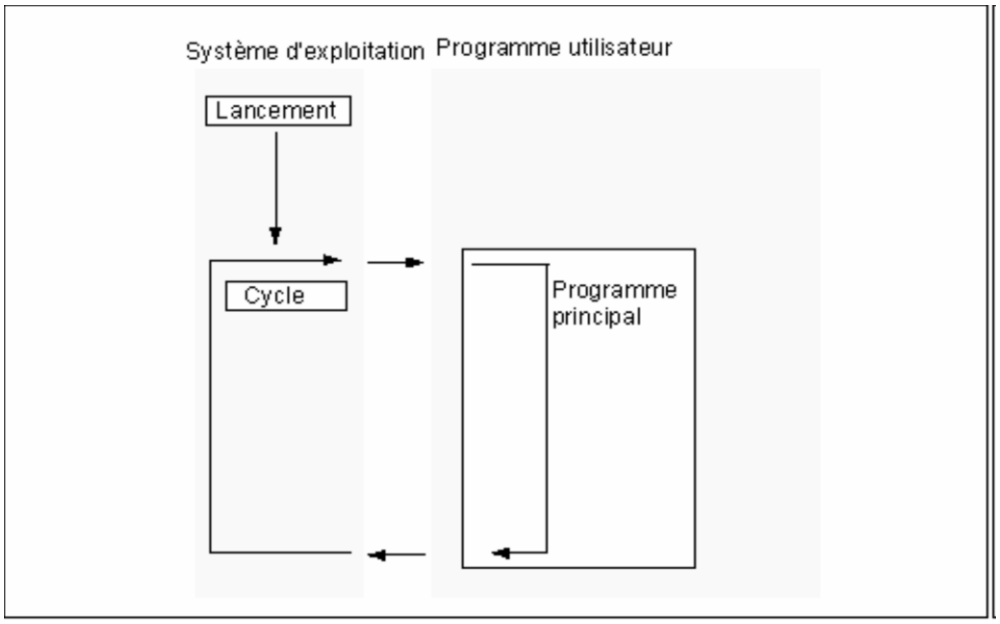


Figure III.9. Le traitement de programme cyclique.

III.6.2. Traitement de programme déclenché par événement

Le traitement de programme cyclique peut être interrompu par des événements déclencheurs précis : les alarmes. En présence d'un tel événement, le bloc en cours d'exécution est interrompu à la fin de l'instruction et le bloc d'organisation associé à l'événement déclencheur est traité. Le traitement du programme cyclique reprend ensuite au point d'interruption.

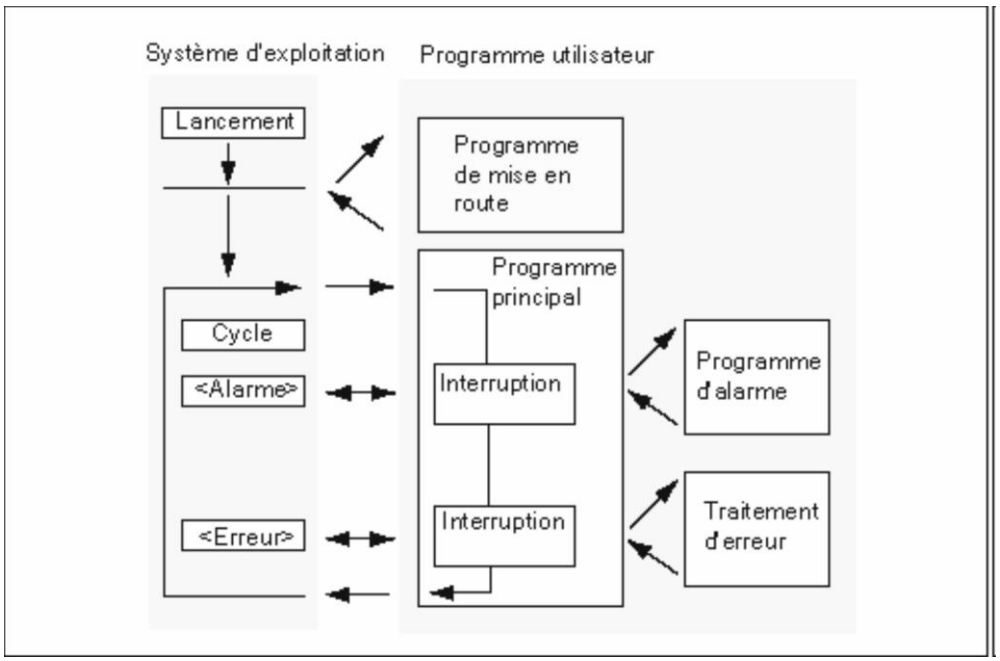


Figure III.10. Traitement de programme déclenché par événement.

III.7.Les types de programmes

On distingue deux types de programmation :

- Programmation linéaire.
- Programmation structurée.

III.7.1.Programmation linéaire

La CPU exécute le cycle habituel, en appelant le bloc OB1 dans le programme principal où les instructions s'exécutent les une après les autres jusqu'à la fin. Cela n'est toutefois recommandé que pour des programmes simples s'exécutant sur des CPU S7-300 avec une mémoire peu importante.

III.7.2.Programmation structurée

La programmation structurée consiste à subdiviser un programme complexe en sous programmes en passant par l'exécuter des fonctions spécifiques plus petites et faciles.

Le programme principal sera chargé pour gérer ces sous-programmes en les appelants autant de fois qu'il est nécessaire. Elle sert à faciliter la maintenance et l'analyse fonctionnelle.

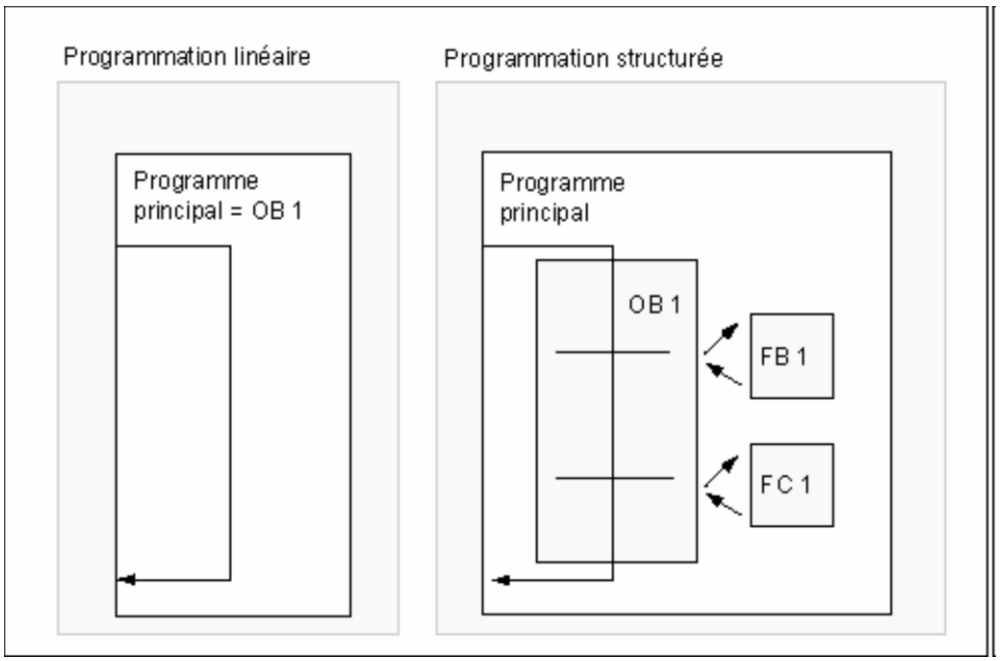


Figure III.11. Programmation linéaire et la programmation structurée

III.8.Deroulement d'un programme

III.8.1. Les programmes existants dans la CPU

Dans une CPU, s'exécutent deux programmes ; le système d'exploitation et le programme utilisateur.

III.8.1.1 Le système d'exploitation

Il regroupe toutes les fonctions et procédures qui ne sont pas liées à la tâche de programmation, ces fonctions et procédures sont :

- La mise en route.
- L'actualisation de la mémoire image des entrées MIE et l'émission de la mémoire image des sorties MIS.
- L'appel du programme utilisateur.
- L'enregistrement des alarmes et l'appel des OB d'alarme.

- La détection et le traitement des erreurs.
- La gestion des zones de mémoires.
- La communication avec les différents partenaires de communication.

La modification des paramètres par défaut du système d'exploitation permet d'influer sur le comportement de la CPU dans des cas précis. Par exemple, le changement des classes de priorités des blocs d'organisation dans les CPU S 400.

III.8.1.2. Le programme utilisateur

Il est créé par l'utilisateur puis chargé dans la CPU, il contient toutes les fonctions nécessaires au traitement de la tâche d'automatisation, il doit :

- Déterminer les conditions pour le démarrage à chaud, à froid ou pour le démarrage de la CPU.
- traiter des données du processus (par exemple, combiner des signaux binaires, lire et exploiter des valeurs analogiques, définir des signaux pour la sortie, écrire des valeurs analogiques).
- Réagir aux alarmes.
- Traiter les perturbations dans le déroulement normal du programme.

Le déroulement d'un programme commence par une mise en route qui n'est exécutée qu'une seule fois, suivie de l'exécution cyclique du programme utilisateur.

III.8.2. La mise en route

Lors de la mise en route la CPU se comporte comme suit :

- Le programme contenu dans l'OB de mise en route est exécuté, dans ce programme on peut définir des initialisations utiles pour le programme utilisateurs.
- Aucun traitement de programme déclenché par horloge n'est possible.
- Les temporisations sont mises à jour.
- Le compteur d'heure de fonctionnement est exécuté.
- Les sorties TOR des modules de signaux sont verrouillées, mais peuvent être mises à 1 par accès direct.

En plus de la mise sous tension, les causes de la mise en route peuvent être :

- Le changement de position du commutateur de mode de fonctionnement de STOP à RUN ou à RUN/P.
- A la demande d'une fonction de communication depuis la console de programmation (PG) ou par l'appel des blocs fonctionnels de communication.

Il existe 3 types de mise en route :

Démarrage à chaud, démarrage à froid et le redémarrage qui correspondent respectivement aux OB 100, OB 102, OB 101.

III.8.2.1. Démarrage à chaud

Lors du démarrage à chaud, la mémoire image et les mémentos, tempos et compteurs non rémanents sont réinitialisés. Les mémentos, tempos, compteurs conservent leur dernière valeur validée. Tous les blocs de données paramétrés avec la propriété "Non Retain" sont réinitialisés aux valeurs de chargement. Les autres blocs de données conservent leur dernière valeur validés [8].

III.8.2.2. Démarrage à froid

Lors du démarrage à froid, toutes les données (mémoire image, mémentos, temps, compteurs et blocs de données) sont ramenées aux valeurs initiales conservées dans le programme, qu'elles aient été paramétrées rémanentes ou non rémanentes.

III.8.2.3.Redémarrage

Lors du redémarrage, toutes les données, y compris la mémoire image, conservent leur dernière valeur valide.

L'exécution du programme se poursuit par la commande à laquelle l'interruption s'est produite.

Jusqu'à la fin du cycle en cours, les sorties ne sont pas modifiées.

III.9. Simulation du programme avec le S7-PLC-SIM

III.9.1. Présentation du PLC-SIM

S7-PLCSIM est une application qui permet d'exécuter et de tester le programme de l'utilisateur élaboré dans un automate programmable et simulé dans l'ordinateur ou à travers une console de programmation [9].

S7-PLCSIM dispose d'une interface simple qui permet de visualiser et de forcer les différents paramètres utilisés par le programme.

Pour ouvrir le simulateur S7-PLCSIM on Sélectionne la commande «outils» et puis «simulation de modules» ou en cliquant sur son icône qui se trouve dans la boite d'outils du gestionnaire de projet SIMATIC.

Pour charger le programme, on click sur l'icône de chargement

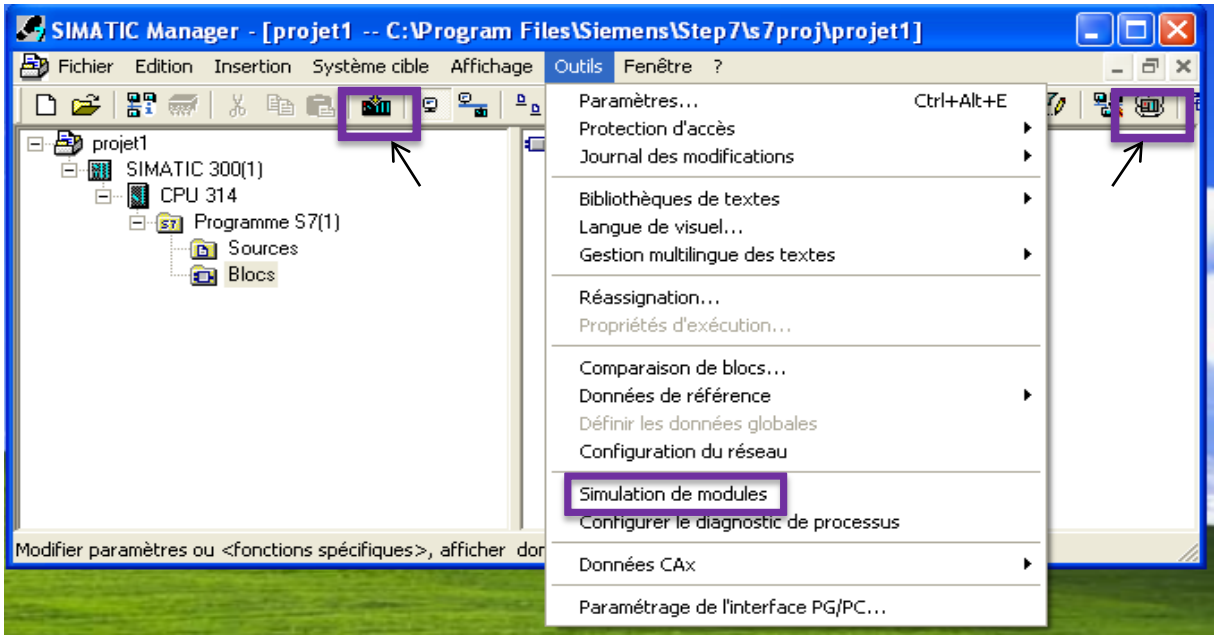


Figure III.12. L'ouverture de simulateur et chargement de projet

Dans le simulateur S7-PLCSIM, les variables d'entrées et de sorties, les mémentos, les temporisateurs, les conteurs sont sous forme de fenêtres. Pour visualiser le fonctionnement de l'automate, on suit les étapes de fonctionnement de la machine avec des cliques sur les entrées pour visualisation des sorties.

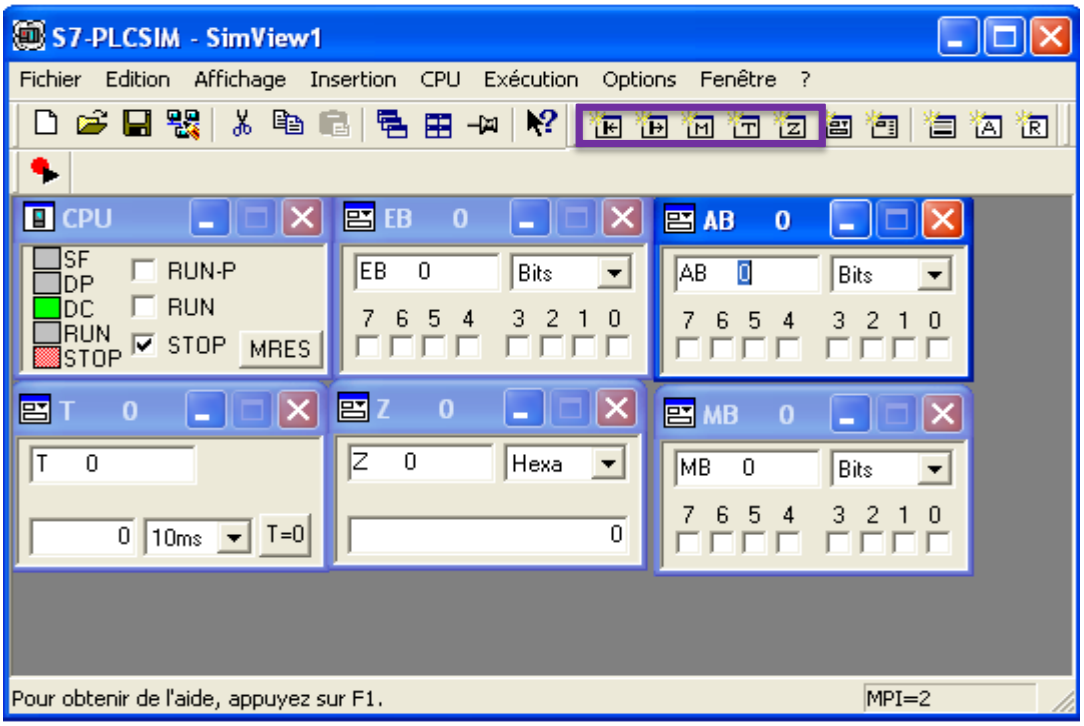


Figure III.13. La fenêtre de simulateur S7 PLCSIM

III.10. Définition de la supervision

La supervision est une forme évoluée de dialogue Homme-Machine, elle sert à représenter et surveiller l'état de fonctionnement d'un procédé.

Les fonctions de la supervision sont nombreuses, on peut citer quelques-unes :

- Elle répond à des besoins nécessitant en général une puissance de traitement importante.
- Assure la communication entre les équipements d'automatismes et les outils informatiques de gestion de la production.
- Coordonne le fonctionnement d'un ensemble de machines enchaînées constituant une ligne de production, en assurant l'exécution d'ordres communs (marche, arrêt,…) et de tâches telles que la synchronisation.
- Assiste l'opérateur dans les opérations de diagnostic et de maintenance.

III.11. Présentation du logiciel WinCC flexible

Lorsque la complexité des processus augmente et que les machines et les installations doivent répondre à des spécifications de fonctionnalité toujours plus sévères, l'opérateur a besoin d'un maximum de transparence. Cette transparence s'obtient au moyen de l'interface homme machine **(**IHM).

Le WinCC (Windows contrôle center) flexible est un logiciel qui permet de crée

l'interface HMI, cette dernière se charge à la Visualisation du processus, la Conduite du processus, l'Affichage des alarmes et l'Archivage des valeurs du processus et d'alarmes.

III.12.Elaboration du programme sous WINCC

III.12.1. Démarrage du logiciel WinCC

Pour lancer le logiciel, on localise l'icône SIMATIC WinCC flexible sur l'écran de l'ordinateur puis avec un double clic sur cette icône, on se permet d'ouvrir sa fenêtre fonctionnelle.

III.12.2. Création d'un nouveau projet

La fenêtre SIMATIC Manager s'ouvre, on crée le projet :

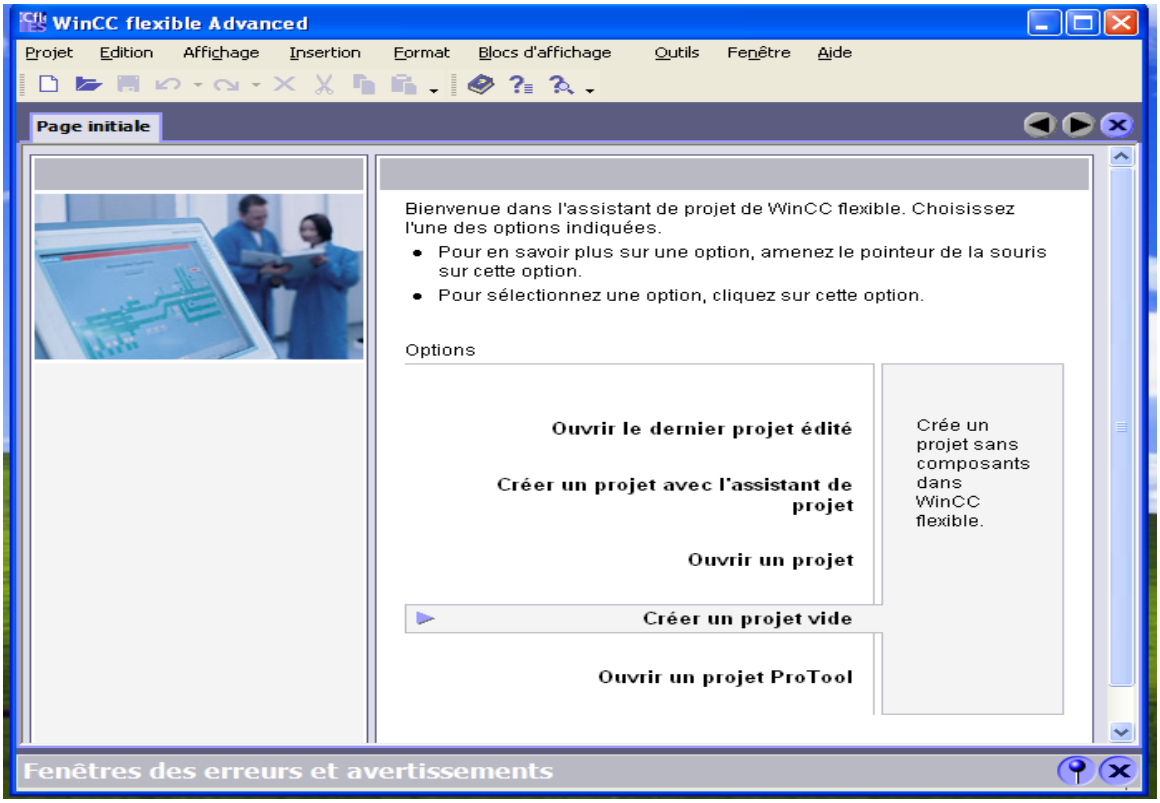


Figure III.14. La création de projet

On sélectionne le type pupitre puis on Valide les paramètres par défaut sur la page «Modèle de vue» avec «Suivant».

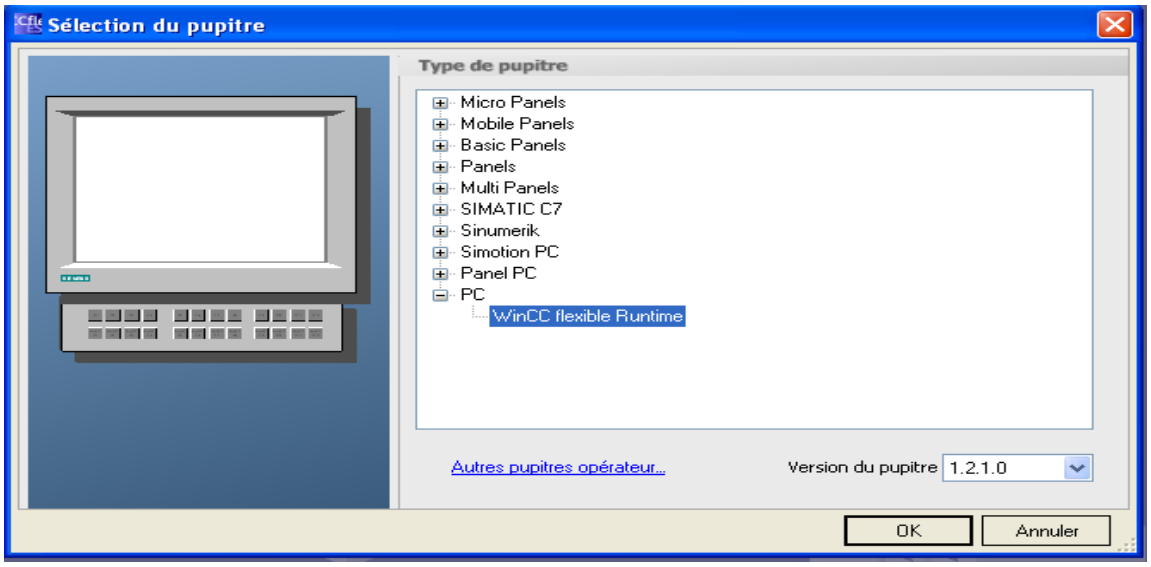


Figure III.15. Les types du pupitre

III.13.La mise en route du WinCC flexible

Après avoir lancé et configuré le logiciel WinCC flexible, ce dernier mettra à disposition une boite d'outils qui contient les différents éléments pour la réalisation d'un projet.

III.13.1.La zone de travail

Sert à éditer les objets du projet. Tous les éléments de WinCC flexible sont disposés autour de la zone de travail. A l'exception de la zone de travail, vous pouvez disposer et configurer, déplacer ou masquer.

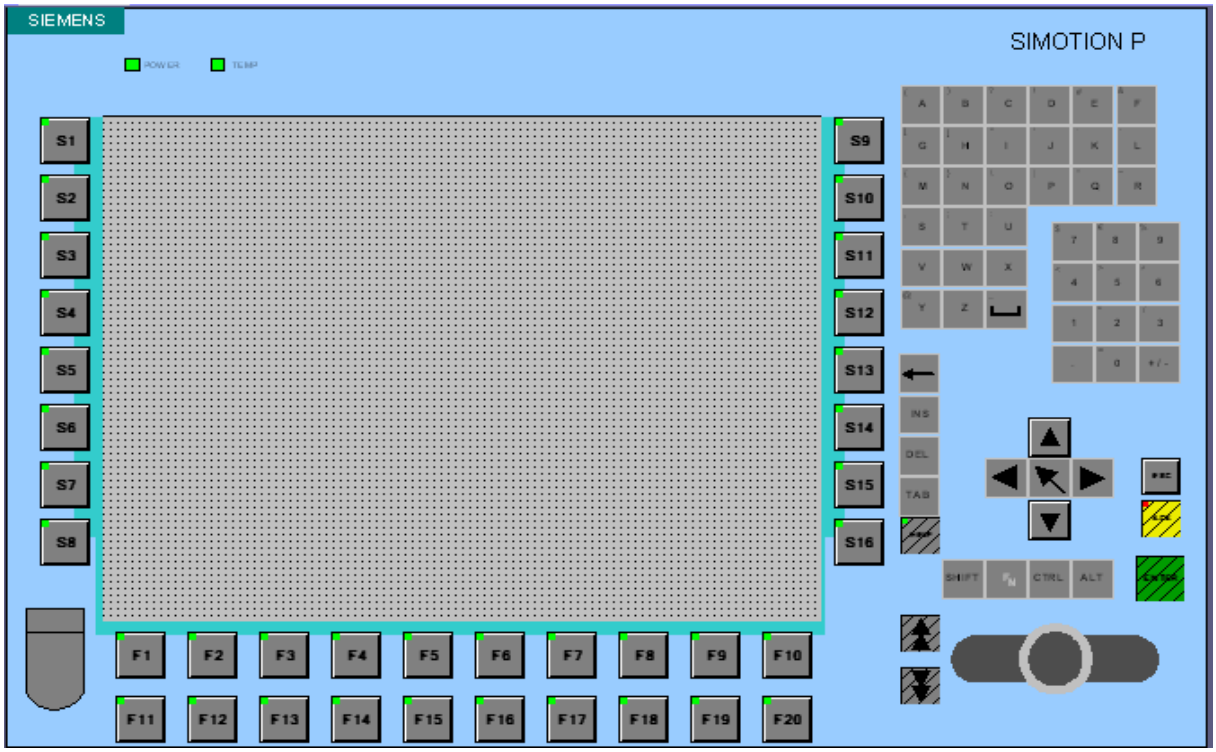


Figure III.16. La zone de travail sous le WinCC flexible

III.13.2. la fenêtre du projet

Dans la fenêtre du projet tous les éléments et tous les éditeurs disponibles d'un projet sont affichés dans l'arborescence et peuvent y être ouverts. Dans la fenêtre de projet, vous pouvez de plus accéder aux propriétés du projet et au paramétrage du pupitre utilisateur [10].

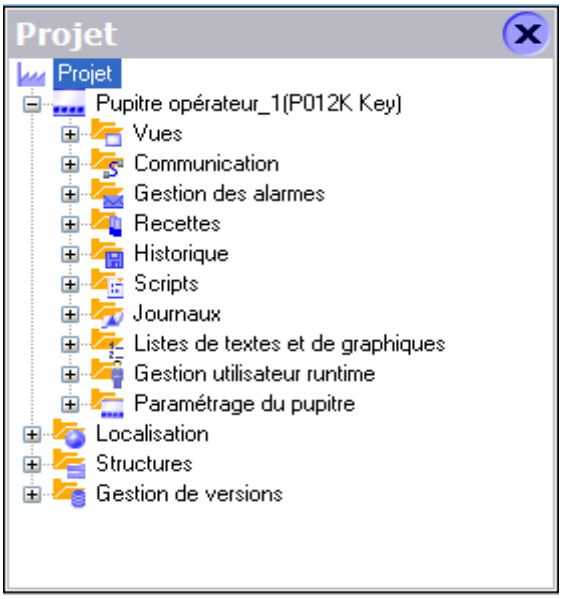


Figure III.17. La fenêtre de projet.

III.13.3. la fenêtre des propriétés

Dans la fenêtre des propriétés vous éditez les propriétés des objets.

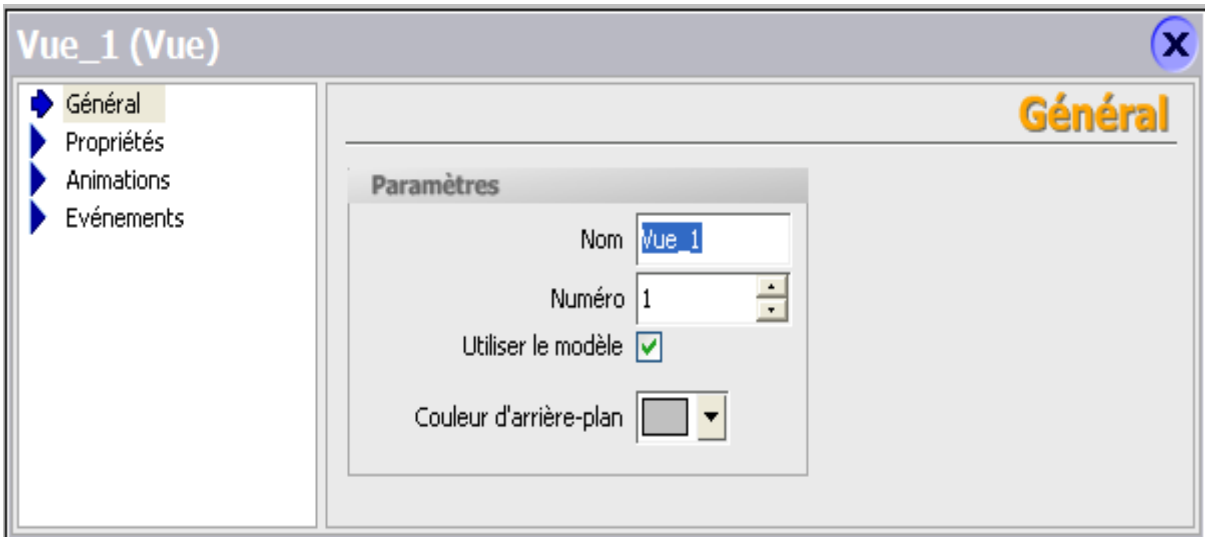


Figure III.18. La fenêtre des propriétés

III.13.4. La fenêtre d'outils

La fenêtre d'outils vous propose une sélection d'objets que vous pouvez insérer dans vos vues. La fenêtre d'outils contient en outre des bibliothèques d'objets et collections de blocs d'affichage prêts à l'emploi.

Figure III.20. La fenêtre d'outils

III.13.5. la fenêtre des erreurs et des avertissements

La fenêtre affiche les erreurs leur de la compilation

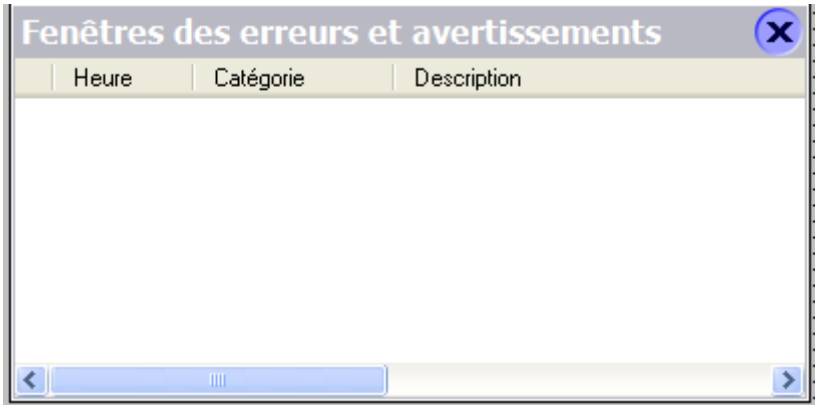


Figure III.21. Les fenêtres des erreurs et des avertissements

III.14. l'intégration de projet dans Step7

Après la sélection du projet STEP7 dans lequel on intègre le projet IHM, on click sur liaison et une autre fenêtre apparait pour choisir le type de connexion entre les pupitres et l'automate.

La communication entre les pupitres opérateurs et les automates SIMATIC S7 peut être réalisée via les réseaux MPI (Multi point Interface), PROFIBUS (process Field Bus) Ethernet…ect).

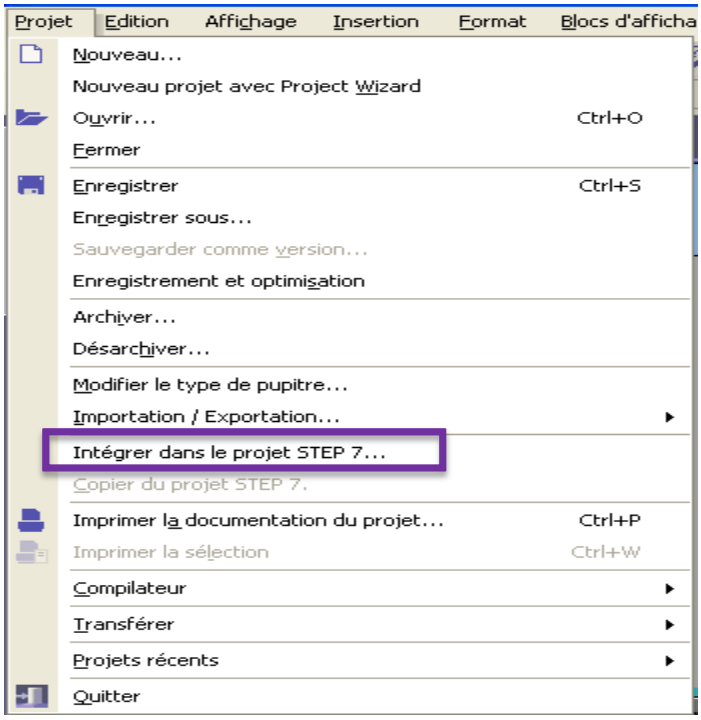


Figure III.22. Intégration de projet WinCC dans Step7

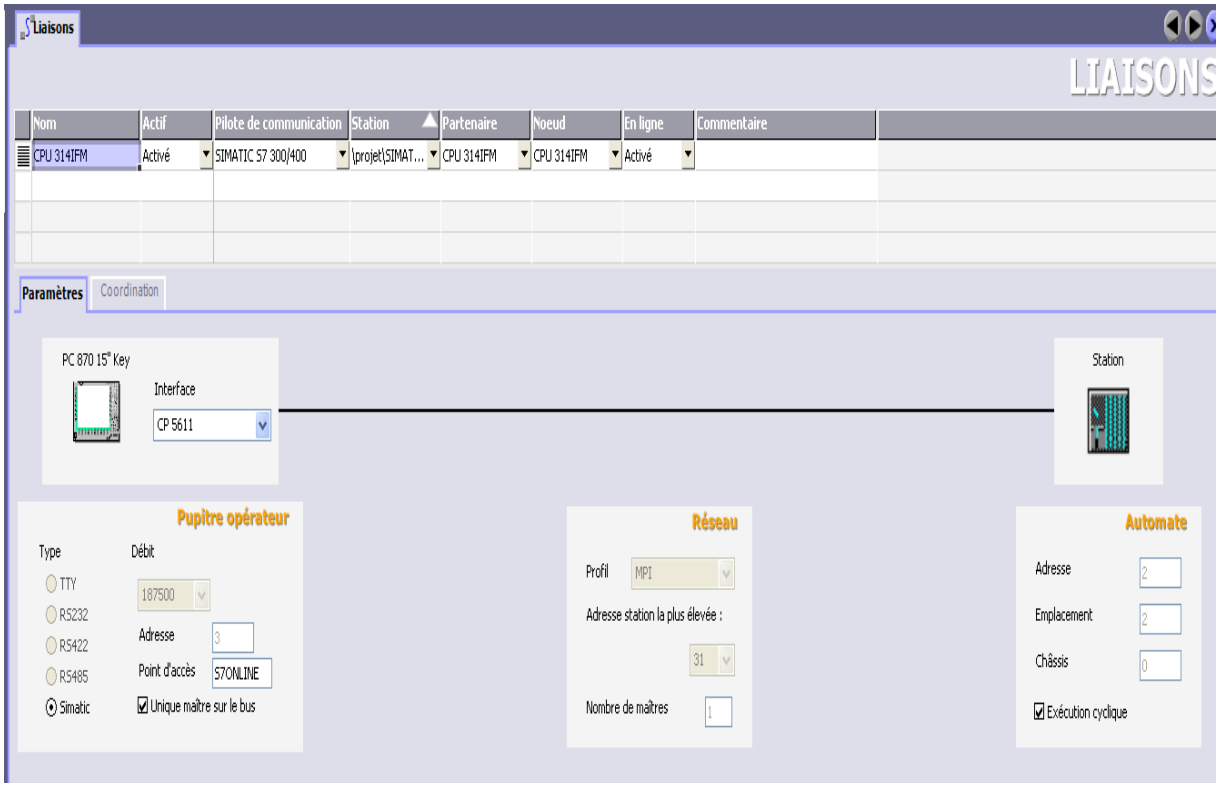


Figure III.23. La création de liaison entre le pupitre et la station

III.15. Simulation du projet sur WinCC flexible Runtime

Logiciel de visualisation de processus WinCC flexible Runtime, permet de faire fonctionner notre configuration sous Windows et de visualiser le processus. Il est également exécuté sur l'ordinateur de configuration pour tester et simuler le fichier projet compilé.

On lance la simulation par click sur l'icône situé dans la barre d'outils : 

La simulation peut être directement lancée en mode plein écran.

III.16.Conclusion

Dans ce chapitre, nous avons donné une description générale sur le Step7 et le WinCC flexible.

Le STEP7 offre la possibilité de programmer et de tests tels que la visualisation du programme afin de remédier à d'éventuelles erreurs commises et les modifications appropriées avant de passer à l'implémentation dans l'automate.

Le WinCC flexible offre le maximum de transparence est essentiel pour l'opérateur qui travail dans un environnement où les processus sont de plus en plus complexes.

Chapitre-IV-

Présentation de la chaine de traitement de surface

IV.1.Introduction

Plus de 90 % des objets d'utilisation courante ont subi un ou plusieurs traitements de surfaces avant d'être commercialisés. Ces traitements permettent de conférer aux solides des propriétés en surface que les matériaux de base ne possèdent pas ou de les améliorer. Ils sont utilisés dans des buts tels que anticorrosion, anti-usure, amélioration des coefficients de frottement, augmentation de la dureté.

La problématique de l'ingénieur qui a la charge de concevoir un objet est double : il doit assurer, d'une part, une **fonction structurale**, d'autre part, une ou des **fonctions superficielles**, le tout dans le cadre de contraintes économiques de plus en plus sévères [14].

La réponse à cette problématique conduit à des choix parfois contradictoires de sorte que la solution la plus pratique consiste à définir deux matériaux, l'un pour la structure, et l'autre pour la surface [14].

Avant de procéder au revêtement superficiel de la pièce, un traitement préalable de préparation de la surface est requis [12].

IV.2.Définition

Un **traitement de surface** est une opération mécanique, chimique, électrochimique ou physique qui a pour conséquence de modifier l'aspect ou la fonction de la surface des matériaux afin de l'adapter à des conditions d'utilisation données.

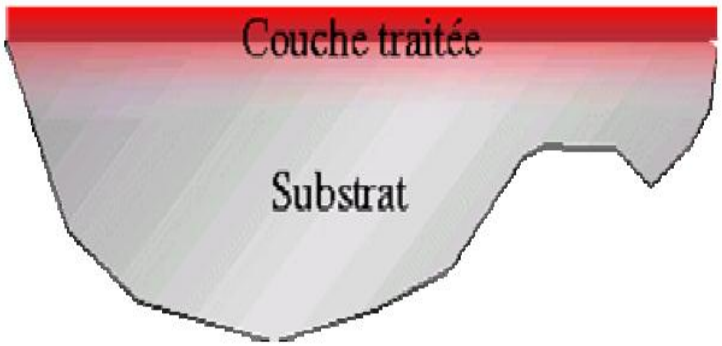


Figure IV.1.traitement superficiel de matériau.

IV.3.La description de processus de décapage par immersion

IV.3.1. Le dégraissage

Il a pour but d'enlever toutes les salissures et graisses qui empêcheraient la dissolution des oxydes de fer superficiels par une action chimique.

IV.3.2. Un rinçage

Il est effectué après le dégraissage afin de ne pas polluer les opérations suivantes en utilisant l'eau.

IV.3.3. Le décapage

Le décapage enlève toute trace de corrosion et d'oxyde qui adhèrent à la surface des pièces. Il est effectué dans une solution d'acide chlorhydrique .

IV.3.4. Rinçage

Un rinçage est également effectué après le décapage afin de laver les pièces des sels de fer et des traces d'acide qui pollueraient l'opération suivante.

S'il n'y a pas de rinçage après décapage, le PH restera bas et peu de nickel sera déposé. De même, s'il est procédé à un rinçage à l'eau, le résultat sera également une faible quantité de nickel déposée.

Le rinçage acide a donc pour but de faire remonter le PH de la pièce, mais sans dépasser la valeur optimale.

IV.3.5. Nickelage

Après avoir éliminé complètement de la surface à recouvrir toute graisse ou tout oxyde métallique, Le nickel joue un rôle important dans le cas du procédé d'émaillage blanc direct pour l'adhérence de l'émail [13].

Les conditions de nickelage ont une très grande importance sur la quantité de nickel déposée. Une faible variation peut avoir de lourdes conséquences sur l'adhérence de l'émail.

De façon à obtenir une bonne adhérence et un bon aspect de l'émail en émaillage blanc direct, il existe une combinaison optimale entre la perte en fer à obtenir et la quantité de nickel [13] :

- perte en fer de 25 à 50 g/m2 par face
- quantité de nickel déposée de 1 à 2 g/m2 par face.

L'épaisseur du revêtement est réglée par la durée du processus.

IV.3.6. Neutralisation

La surface contient des produits chimiques résultant de l'action des acides sur les oxydes. Ensuite, ces produits sont donc éliminés en procédant à leur neutralisation et en les nettoyants à l'eau courante.

IV.3.7. Séchage

A la fin de la préparation de surface, il est nécessaire que les pièces soient séchées pour éviter qu'elles ne s'oxydent avant émaillage.

IV.4.Le cycle de traitement.

Tableau. IV.1. Les différentes opérations du processus de décapage

Les bains	Le traitement	La température	Le temps	Le produit
Chargement	/	/	/	/
Bain1	Dégraissage	80°C-95°C	6 min	MEDASOL+CLEANER
Bain2	Dégraissage	80°C-95°C	6 min	MEDASOL+CLEANER
Bain3	Rinçage	60°C	3 min	L'eau
Bain4	Dégraissage	80°C-95°C	6 min	PACO415 MEDASOL
Bain5	Rinçage	60°C	6 min	L'eau
Bain6	Rinçage	Ambiante	3 min	L'eau
Bain7	Décapage	50°C	6 min	H_2SO_4 concentré
Bain8	Rinçage	Ambiante	2 min	L'eau
Bain9	Dégraissage	80°C-95°C	6 min	PACO415 MEDASOL
Bain10	Rinçage	60°C	6 min	L'eau
Bain11	Rinçage	Ambiante	3 min	L'eau
Bain12	Décapage	65°C-70°C	6 min	H_2SO_4 concentré
Bain13	Rinçage	Ambiante	4 min	L'eau
Bain14	Nickelage	65°C-70°C	5 min	$NISO_4$
Bain15	Rinçage	Ambiante	3 min	H_2SO_4 PH=2
Bain16	Rinçage	Ambiante	2 min	L'eau
Bain17	Neutralisation basique	80°C-90°C	3 min	MEPPASS ALC
Bain18	Séchage	100°C	6 min	AIR chaud
Déchargement	/	/	/	/

Le temps total d'un cycle (chargement jusqu'au déchargement) est 91mn.

IV.5. CONSTITUTION DE LA CHAINE DE TRAITEMENT DE SURFACE

La chaîne de traitement de surface installée au sein de l'unité cuisson est constituée de plusieurs éléments (figure IV.2) :

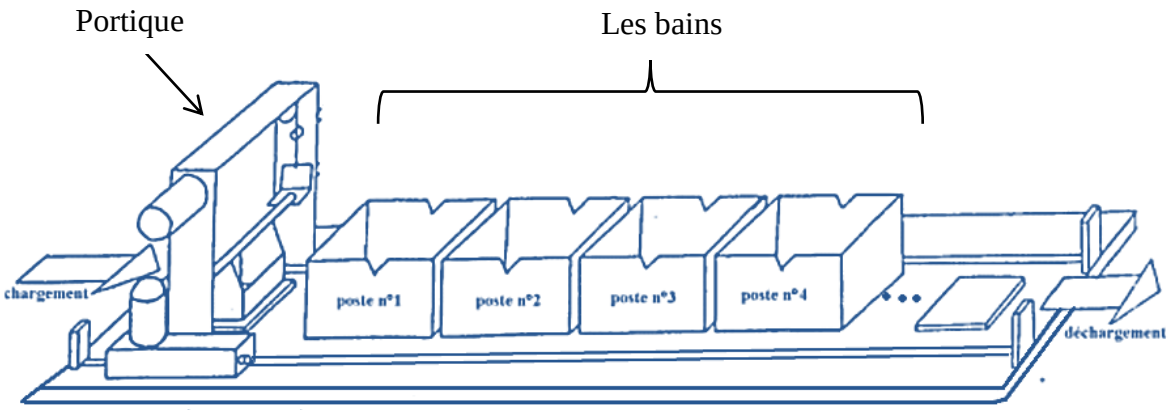


Figure IV.2.la chaine de production

IV.5.1.Poste de chargement

Il est doté d'un chariot pour mettre la corbeille qui contient les pièces non traités et deux capteurs pour détecter la position de chariot.

Les opérations de chargement et de déchargement des pièces à traiter sont effectuées manuellement par l'opérateur chargé de conduire la machine

Figure IV.3. le chariot

Figure IV.4. la corbeille

IV.5.2.Les bains

Elle comporte 18 bains métalliques alignés sur le même axe. Ils contiennent des solutions chimiques pour le traitement des pièces.

Les bains sont équipés avec des thermomètres qui permettent d'afficher la température des bains.

IV.5.3.Les portiques

La chaîne comporte 2 portiques montés sur des rails, ils ont pour rôle de transporter les pièces d'une cuve à l'autre. Chaque portique dessert une partie du tronçon, portique 1 permet de déplacer les corbeilles de poste 0 (chargement) au poste 10 et le deuxième de 10^{eme} au 19^{eme} poste une seule cuve sert de lien entre les parties.

Chaque portique est entraîné par trois moteurs (deux pour le mouvement transversal et le troisième pour le mouvement vertical), ces moteurs sont triphasés asynchrone qui peuvent tourner dans deux sens (à gauche et droite pour le moteur de translation, en haut et en bas pour le moteur de levage), et avoir deux vitesses (grande et petite) pour les moteurs de translations. Chacun des trois moteurs est dotés aussi d'un frein mécanique.

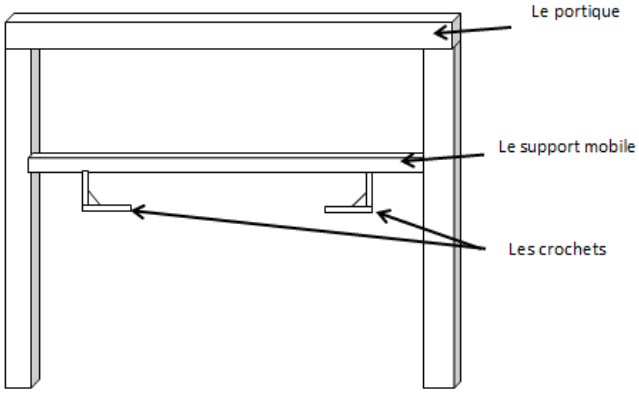


Figure IV.5.le portique

Chaque portique est doté de plusieurs fins courses.

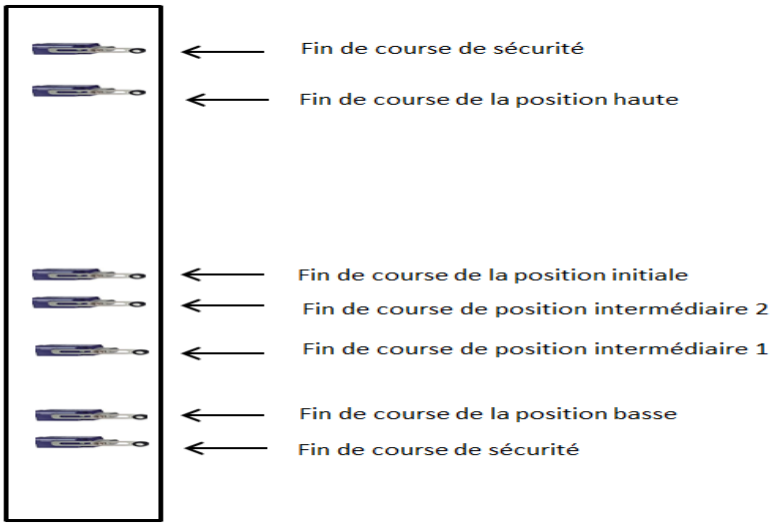


Figure IV.6. Les fins de course montés sur le portique (mouvement de levage)

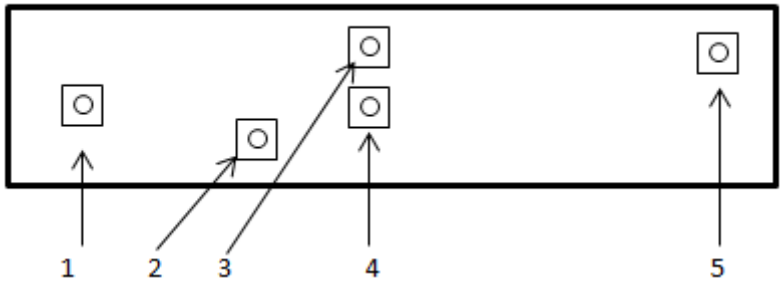


Figure IV.7. Les fins de course montés au-dessous de portique (mouvement de translation)

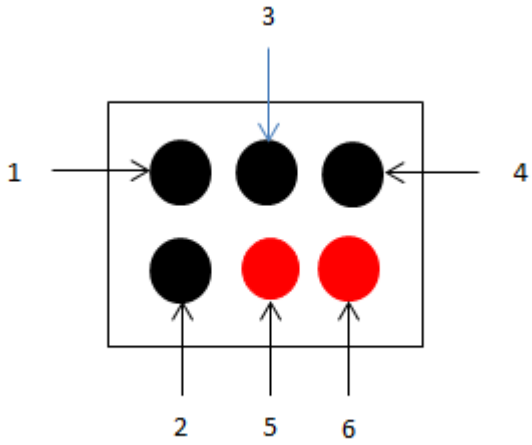
1 : fin de course pour la détection de vitesse lente (recule).

2 : fin de course de position intermédiaire.

3 : fin de course pour le freinage de moteur.

4 : fin de course pour le freinage de moteur.

5 : fin de course pour la détection de vitesse lente (avance).



1 : haut 4 : droit

2 : bas 5 : démarrer

3 : gauche 6 : stop

Figure IV.8. Les boutons de la commande manuelle

Chaque portique a un cycle de travail

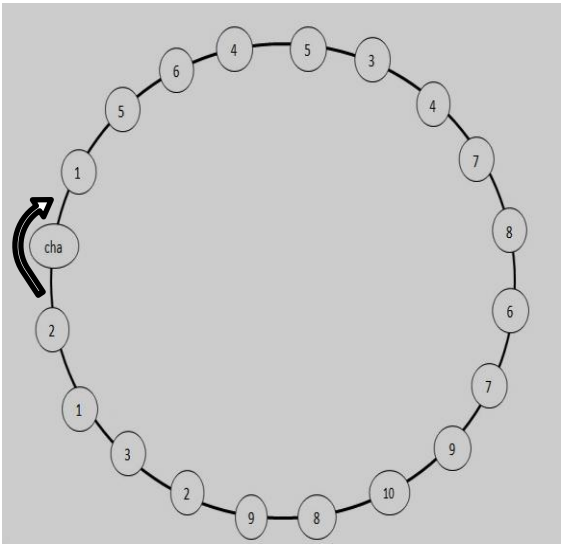


Figure IV.9.Cycle pour le portique 1

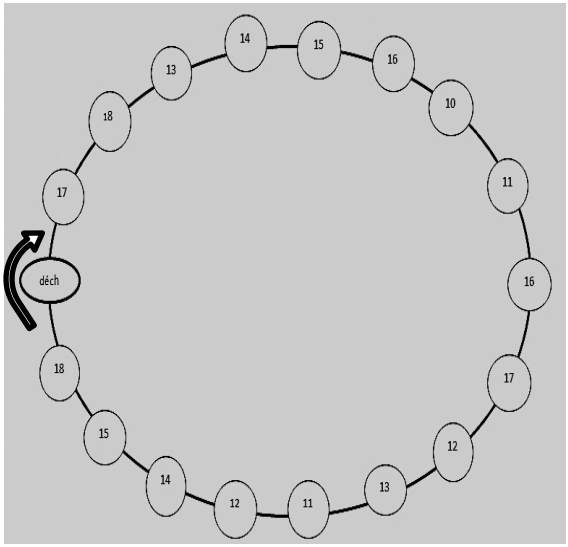


Figure IV.10. Cycle pour le portique 2

IV.5.5.Poste de déchargement

Ce poste est semblable à celui de poste de chargement, mais les pièces sont traitées.

IV.6.Fonctionnement du système

Le système a été conçu pour mettre en œuvre les différentes phases du traitement à réaliser les transferts entre chaque poste. Les pièces sont disposées dans des corbeilles qui permettent aux portiques de faire les déplacements de cuve en cuve. Les pièces à traiter doivent passer par 18 bains, selon un ordre établi.

L'opérateur après avoir mis les tôles à traiter dans une corbeille qui sera posée sur un chariot au niveau de poste de chargement et si le portique 1 et portique 2 sont en position initiale, l'opérateur peut donner un ordre départ de cycle.

Le treuil de portique 1 est élevé jusqu'en position haute. Arrivé en fin de couse haut, le portique effectue une translation rapide après ralentis et se positionne au milieu du bain de dégraissage. Le treuil descend la corbeille dans le bac, Arrivé en position basse, celui-ci enclenche une temporisation. Le portique 1 recommencera les mêmes opérations jusqu'à au bain N°10 et il suit le cycle montré dans la Figure IV.9, même pour le portique 2, il suit le cycle montré dans la figure IV.10.

Remarque

- On a constaté que les portiques marchent à vide (fait l'action « translation, levage » malgré l'absence des corbeilles).
- pour la moindre modification dans l'ordonnancement des séquences impose la refonte complète de câblage.
- Divers paramètres sont à contrôler la nature du produit, la température, la concentration, le pH des bains, le temps de traitement. Un paramètre non respecté est générateur de défauts de surface, ce qui montre l'importance du contrôle instantané des conditions de ces opérations.

IV.7.Conclusion

Dans ce chapitre nous avons décrit la situation actuelle de la chaîne de production installée au sein de l'unité cuisson, ce qui nous a permis de localiser les principaux défauts, causés, en grande partie, par la technologie de contrôle adoptée.

Nous avons au cours de ce chapitre cité les différentes composantes du matériel utilisé pour l'opération de décapage par immersion. Tout cela en vue de modéliser le système pour pouvoir ensuite concevoir un système de commande moderne du processus, basé essentiellement sur l'entité **Automate Programmable Industriel**, pour lequel nous allons consacrer tout le chapitre suivant.

Chapitre-V-

Elaboration du GRAFCET, Programmation et Supervision de la chaine

V.1. Introduction

Après avoir présenté la chaine de production et décelé ses manquements, il est question maintenant pour nous d'y apporter les solutions correctives et amélioratrices qui répondre au cahier des charges donné par l'entreprise d'où l'automatisation de l'ensemble de système. Car la fonction globale de tout système automatisé est de lui conférer une valeur ajoutée.

V.2. Définition du cahier des charges

Le cahier de charge a été défini par l'entreprise. Le travail demandé consiste à élaborer un programme afin de commander les deux portiques, Développer une Interface Homme Machine afin de Visualiser le processus et afficher les défauts et les alarmes. Tout en utilisant un automate programmable pour cette gestion.

On peut diviser ce cahier de charge en trois parties :

- traduire le cahier de charge en un GRAFCET.
- la programmation avec Step 7.
- la supervision avec WinCC .

V.3. Réalisation du GRAFCET de la chaine de production

Avant de passer au GRAFCET, il est nécessaire de prendre en compte le comportement du système au regard des sécurités ou défaillances et d'une façon plus générale, d'étudier comment arrêter le système puis comment le redémarrer.

Par ailleurs, suite à un arrêt dans une situation donnée, il est nécessaire de remettre en position initiale le système pour accéder au grafcet de production normale.

Dans le ce cas il faut remplir le guide GEMMA (chapitre II), qui permet d'élaborer une hiérarchisation de grafcet. Ces différents fonctionnements peuvent être décrits de la manière suivante :

- La fermeture du sectionneur et un appuyé sur le bouton mise sous tension entraîne l'alimentation du circuit de commande, celle de l'automate et de ses entrées T.O.R. En RUN, l'automate déroule son programme et initialise le grafcet.

- Après, il vient le choix de mode de marche (auto/manu) .

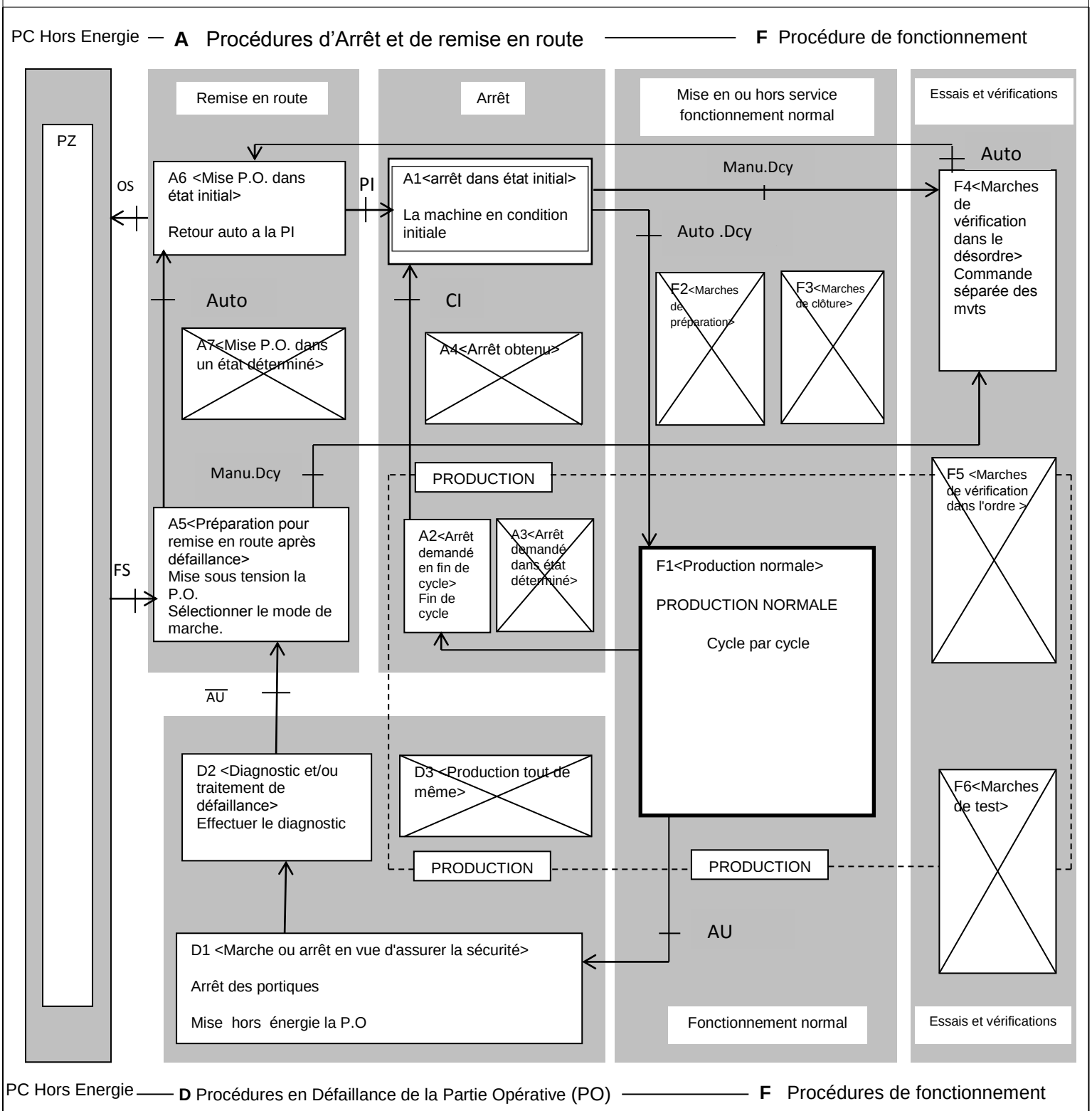
- Pour des raisons de réglage et de vérification, il peut être nécessaire de commander les actionneurs en mode manuel. Un ordre par l'operateur peut ensuite remettre le système en mode automatique.

- Si le portique1 est au poste N°0 (chargement), treuil en position initiale, et le portique 2 est au poste N°19 (déchargement), treuil est en position initiale, une action sur le bouton Dcy permet le départ du cycle de production normale. Sinon il faut appuyer sur le bouton PI pour ramener les portiques à ces positions initiales.

- En fin de cycle, retour à l'état initial en attente d'un nouvel ordre "Dcy".

- La commande d'arrêt d'urgence peut provenir d'une intervention humaine (sur-courses, source d'énergie non-conforme, la collision des portiques… etc). En ce qui concerne l'installation nous prévoirons 4 boutons d'arrêt d'urgence "AU1" et "AU2" situés à chaque extrémité de la chaine, AU3 sur l'armoire et AU4 sur le pupitre.

- L'arrêt d'urgence est un ordre prioritaire qui doit faire évoluer le système à l'état d'arrêt quelle que soit la situation dans laquelle il se trouve.



LEGENDE

P.O. = Partie Opérative	P.C. = Partie Commande
PI = bouton de la position initiale	Auto/Manu = commutateur
AU = arrêt d'urgence	CI = condition initiale
FS = fermeture du sectionneur	OS = ouverture du sectionneur
Dcy = bouton pour le démarrage de cycle	

Figure V.1. Le GEMMA de traitement de surface

V.4. Définition des entrées/sorties

D'après l'étude que on a fait sur l'installation et les exigences de cahier des charges, on peut définir le type et le nombre des entrées/sorties nécessaires à la récupération du maximum d'informations (annexe A). Donc on a dimensionné :

- 104 entrées TOR : celles qui viennent des capteurs de présences des corbeilles dans les bains, les capteurs de position, fin de course de sécurité, Les boutons (Dcy: démarrer le cycle, PI : position initiale, Arrêt d'urgence) et les boutons de la commande manuelle.
- 22 sorties : la commande des moteurs et les voyants.

V.5. La structure générale de GRAFCET

En mode manuel, les boutons poussoirs situés sur le pupitre de commande permettent d'actionner le/les portiques.

En mode automatique, un cycle programmé disponible dans l'automate permet de commander les portiques.

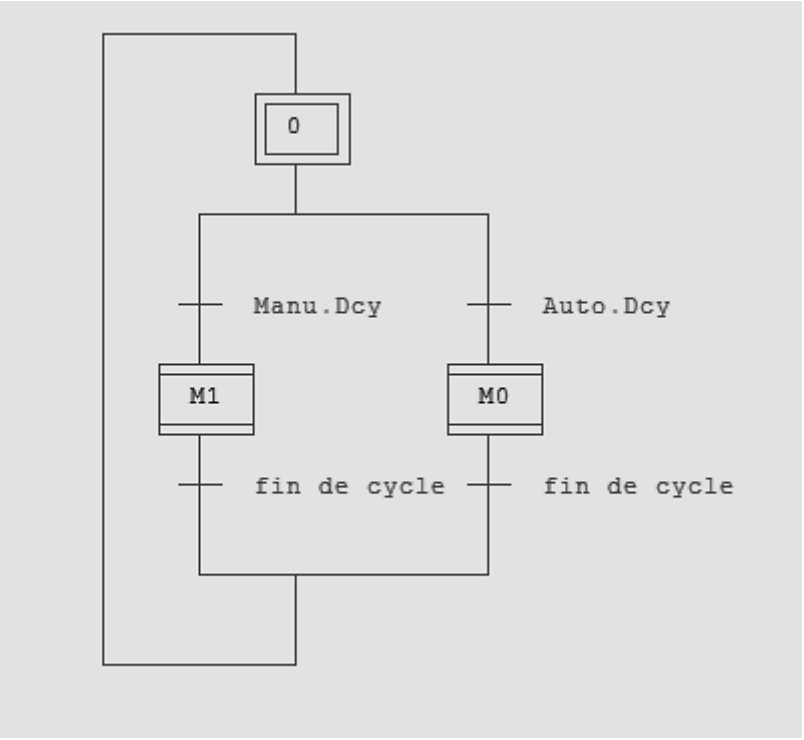
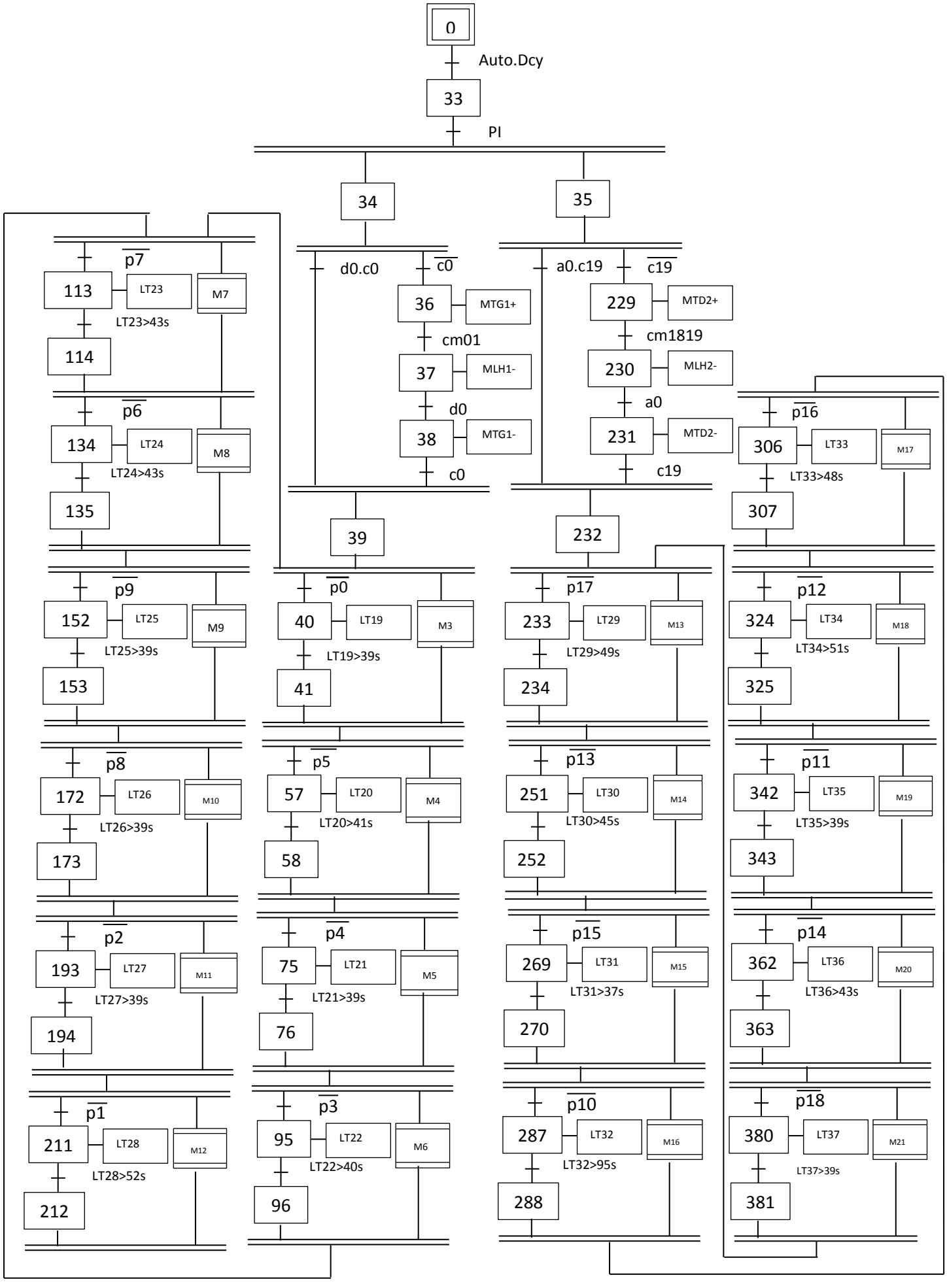


Figure V.2. La structure génerale de GRAFCET

M0 : macro etape (represente le cycle automatique)

M1 : macro etape (represente le cycle manuel)

V.6.LE GRAFCET de mode automatique



FigureV.3.LE GRAFCET de mode automatique

V.7.Le GRAFCET en mode manuel

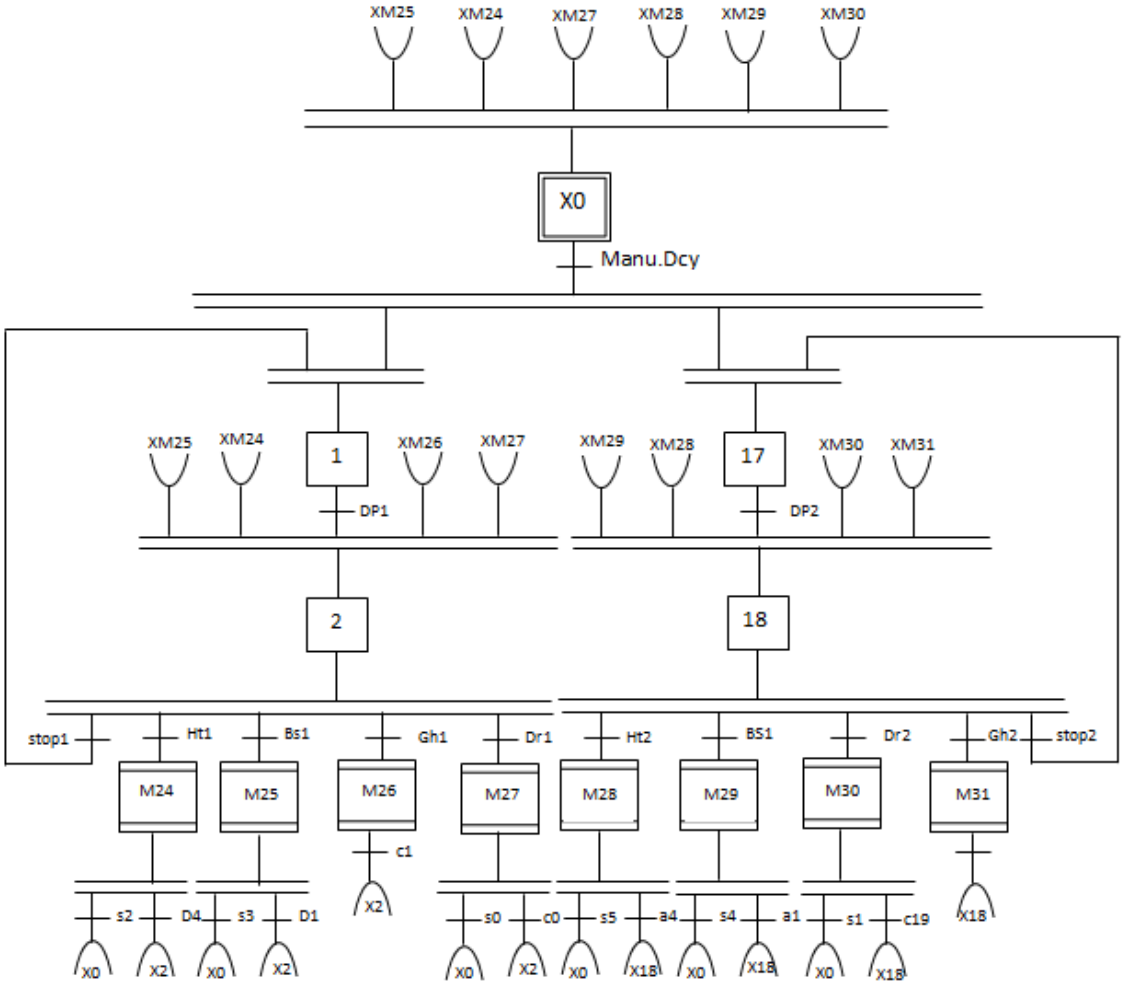


Figure.V.4. Le grafcet en mode manuel

V.8.Le choix de l'automate

Dans notre cas, le choix est porté sur l'automate programmable SIEMENS S7-300 CPU 314 IFM, et cela pour les raison suivantes :

- **Type d'entrées/sorties :** L'automate programmable S7-300 dispose de module d'entrées/sorties de type TOR et analogique.
- **Nombre d'entrées/sorties :** Le châssis de S7-300 peut prendre 8 modules de signaux de communication (analogique ou TOR) ce qui est largement suffisant pour concevoir notre application.
  - **104** entrées tout ou rien.
  - **22** sorties tout ou rien.
- **La communication :** L'automate doit pouvoir communiquer avec les autres systèmes de commande (API, supervision ...) et offrir des possibilités de communication par Profibus, MPI…etc.
- **La disponibilité du matériel (API)** : Sur le marché, l'existence de la documentation et le savoir-faire du personnel sur Le matériel. SIEMENS est une marque de renommé mondiale sur le marché des API.

V.9. Création du projet STEP 7

Cette étape consiste à la création du projet STEP 7, la configuration des matériels, Ainsi que l'écriture du programme (chapitre III).

V.10. Configuration du matériel dans STEP7

Le projet a été créé selon la procédure vue (chapitre III). Après l'insertion de la station S7-300 nous configurons le matériel utilisés :

Module ...	Référence
PS 307 5A	6ES7 307-1EA00-0AA0
CPU 314IFM	6ES7 314-5AE02-0AB0
DI64xDC24V	6ES7 321-1BP00-0AA0
DI64xDC24V	6ES7 321-1BP00-0AA0
DO32xDC24V/0.5A	6ES7 322-1BL00-0AA0

Figure V.5. La configuration matérielle

V.11. La hiérarchie de programme

On a subdivisé le système globale en plusieurs sous-systèmes, Pour mieux gérer le système global, c.-à-d. on crée des blocs fonctionnels (FB) chacun possèdent des données contenues dans le bloc de données (DB) associé, Ceci permet de mieux tester et déboguer les programmes.

Le GRAFCET global du projet est représenté par le bloc d'organisation OB1, qui est un bloc programmé qui contient des macros étapes représentant l'appel de tous les blocs fonctionnels du projet tel que le FB1 (bloc fonctionnel pour le cycle45), le FB2 (bloc fonctionnel du cycle 01) ….etc.

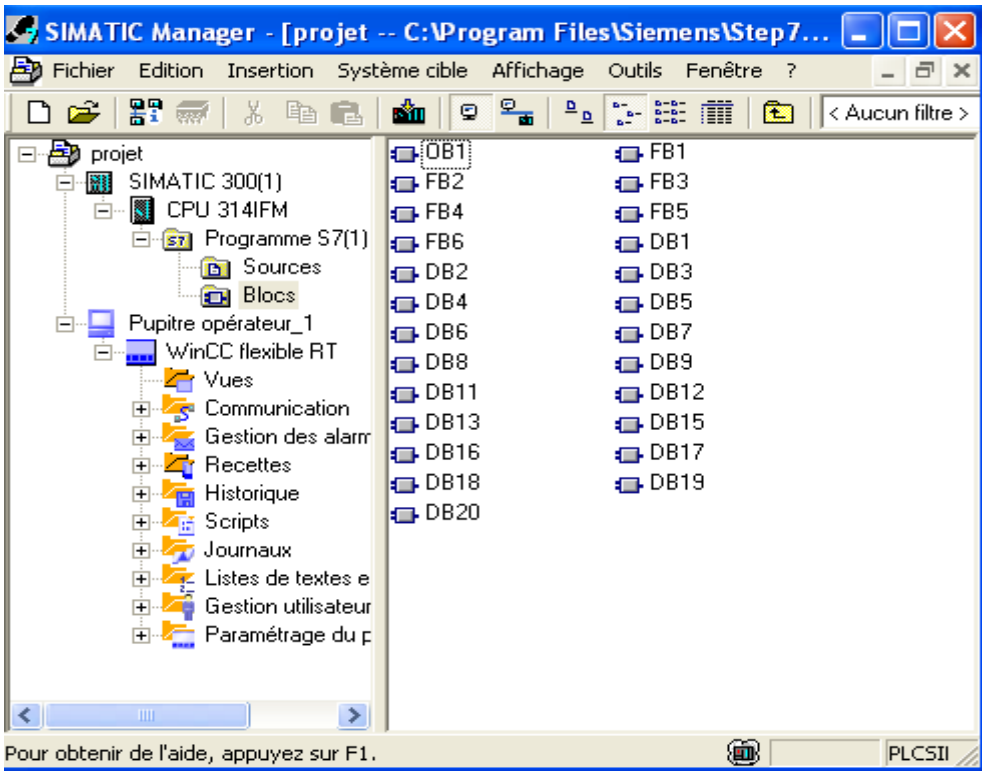


Figure.V.6. La hiérarchie de programme(1)

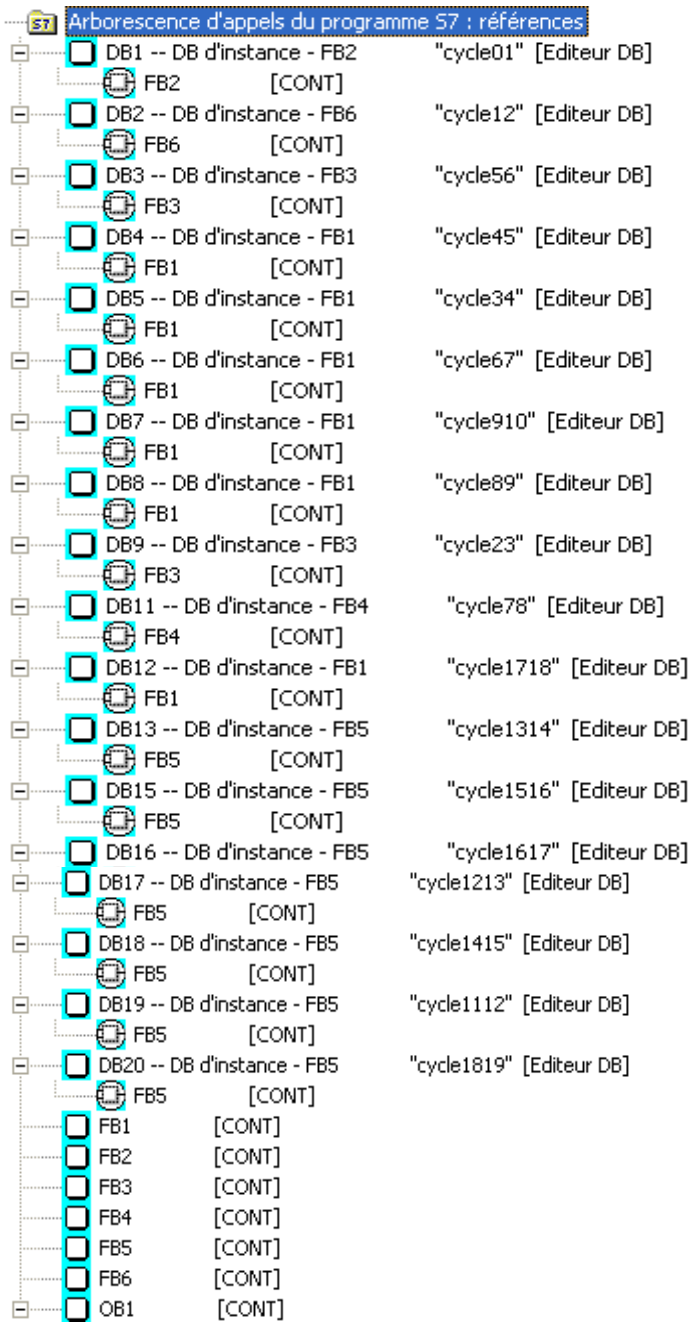


Figure V.7. La hiérarchie de programme(2)

V.12.Ecriture du programme

Avant de pouvoir commencer l'écriture du programme en sens propre, nous devons définir au préalable tous les mnémoniques (Annexe).

Dans la partie programmation on s'est basé sur le langage de programmation graphique CONT .Ce dernier permet de suivre le trajet du courant entre les barres d'alimentation.

De plus notre programme ne nécessite pas un langage de programmation trop développé.

Pour la raison de volume de programme, nous avons choisi quelques étapes, que nous avons jugé les expliquer.

V.13. Présentation les étapes et les entrées par les Bascules et les contacts.

Chaque état de grafcet est représentée par une bascule RS, un état exécute la mise à 1 (Q=1) si l'état de signal est 1 à l'entrée S et 0 à l'entrée R. Si l'état de signal est 0 à l'entrée S et 1 à l'entrée R, la bascule est mise à 0.

Les entrées et les mémentos sont présentés par des contacts.

Le contact est fermé si la valeur du bit interrogé sauvegardée en <opérande> égale 1. Dans pareil cas, le courant traverse le contact et l'opération fournit un résultat logique égal à 1.

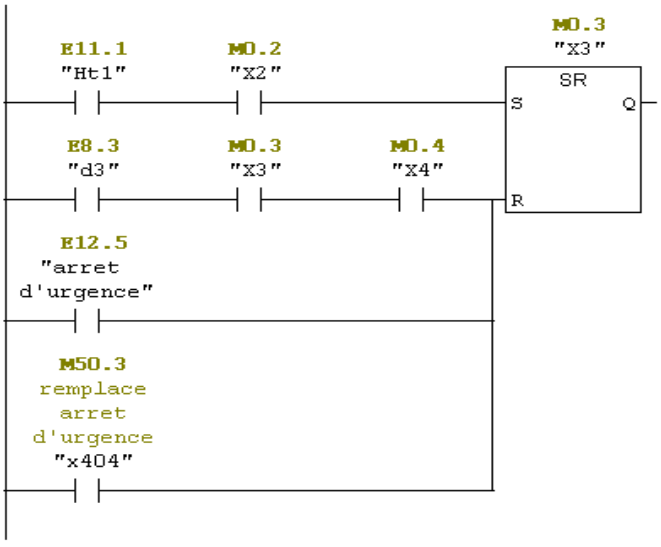


Figure V.8. Exemple sur une bascule SR

V.14. Les temporisateurs

Dans la programmation on a utilisé deux types de temporisateurs :

- S_SEVERZ : Paramétrer et démarrer temporisation sous forme de retard à la montée mémorisé.
- S_EVERZ : Paramétrer et démarrer temporisation sous forme de retard à la montée

V.14.1.le temporisateur de type S_SEVERZ

La temporisation est démarrée en cas de front montant à l'entrée de démarrage S. La valeur de temps indiquée à l'entrée TW continue à s'écouler même si l'état de signal à l'entrée S passe à 0 avant que la temporisation n'ait expiré. L'état de signal à la sortie Q égale 1 lorsque la temporisation a expiré, quel que soit l'état de signal à l'entrée S. Si l'état de signal à l'entrée S passe de 0 à 1 alors que la temporisation s'exécute, cette dernière est redémarrée avec la valeur de temps indiquée.

En cas de passage de 0 à 1 à l'entrée de remise à zéro R, la temporisation est remise à zéro quel que soit l'état à l'entrée S. L'état de signal à la sortie Q est alors 0.

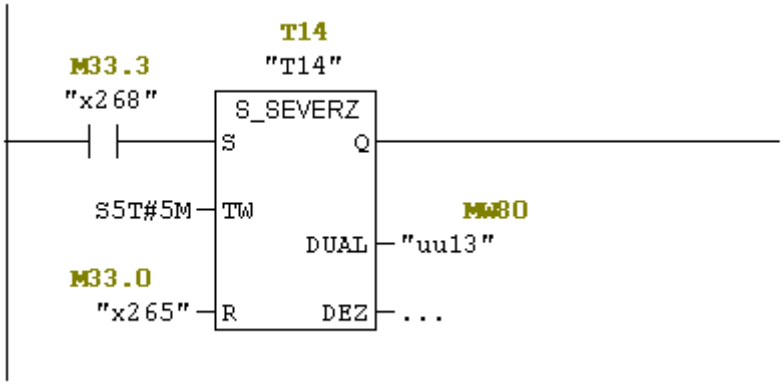


Figure V.9. Exemple sur un temporisateur SS

V.14.2.le temporisateur de type S_EVERZ

La temporisation est démarrée en cas de front montant à l'entrée de démarrage S. La valeur de temps indiquée à l'entrée TW s'écoule tant que l'état de signal à l'entrée S est à 1. L'état de signal à la sortie Q égale 1 lorsque la temporisation s'est exécutée sans erreur et que l'état de signal à l'entrée S est toujours 1. La temporisation s'arrête si l'état de signal à l'entrée S passe de 1 à 0 alors que la temporisation s'exécute. Dans ce cas, l'état de signal à la sortie Q est 0.

En cas de passage de 0 à 1 à l'entrée de remise à zéro R pendant que la temporisation s'exécute, cette dernière est remise à zéro.

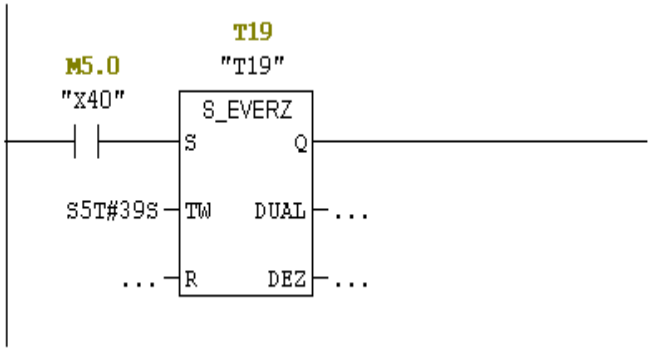


Figure V.10. Exemple sur un temporisateur SE.

V.15.Le compteur

ZAEHLER (Paramétrage et compteur d'incrémentation/décrémentation)

Un front montant à l'entrée S initialise le compteur à la valeur figurant dans l'entrée ZW. Un 1 à l'entrée R remet le compteur, et donc la valeur de comptage, à zéro.

Le compteur est incrémenté d'une unité si l'état de signal à l'entrée ZV passe de 0 à 1 – front montant – et que la valeur du compteur est inférieure à 999. L'état du signal à la sortie Q est à 1 lorsque la valeur de comptage est supérieure à 0 ; il est à 0 lorsque la valeur de comptage est égale à 0.

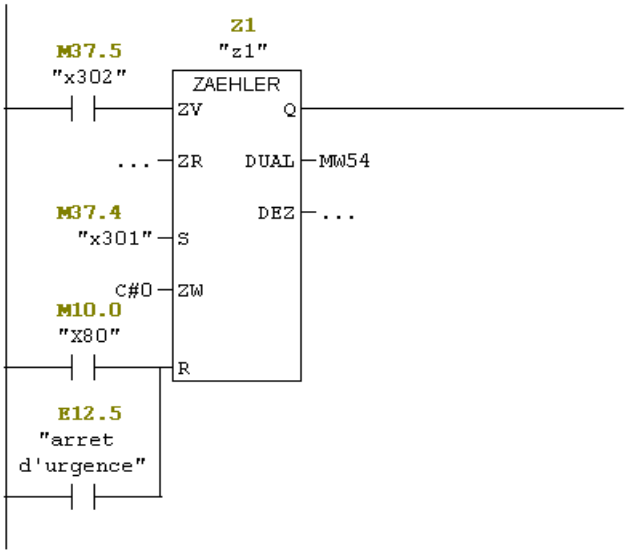


Figure V.11. Exemple sur un compteur

V.16.un comparateur

CMP I (Comparer entiers de 16 bits) .on a utilisé se comparateur pour comparer la valeur de comptage et 38(nombre de basculement leur de rinçage). Si la comparaison est vraie, le résultat logique est 1.

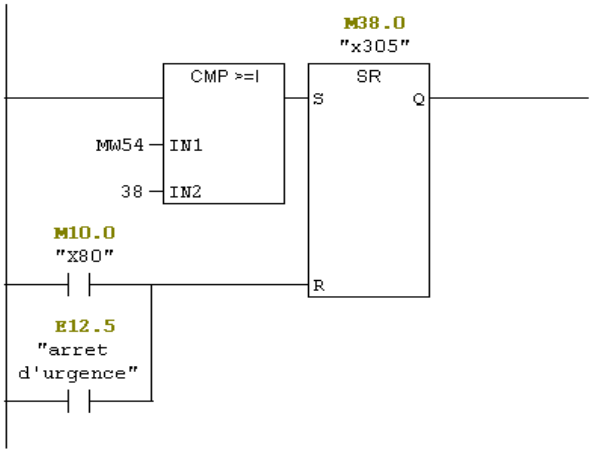


Figure V.12. Exemple sur un comparateur

V.17.Les bobines des sorties

Cette opération fonctionne comme une bobine dans un schéma à relais. Si l'énergie atteint la bobine, le bit en <opérande> est mis à 1. Si l'énergie n'atteint pas la bobine, le bit en <opérande> est mis à 0.

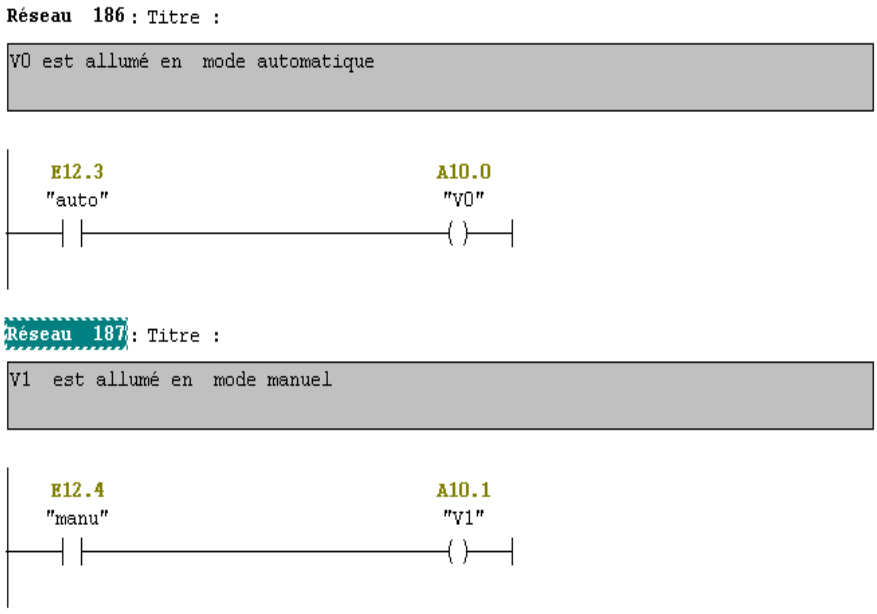


Figure V.13. Exemple sur les bobines de sorties

V.18.Les blocs FB

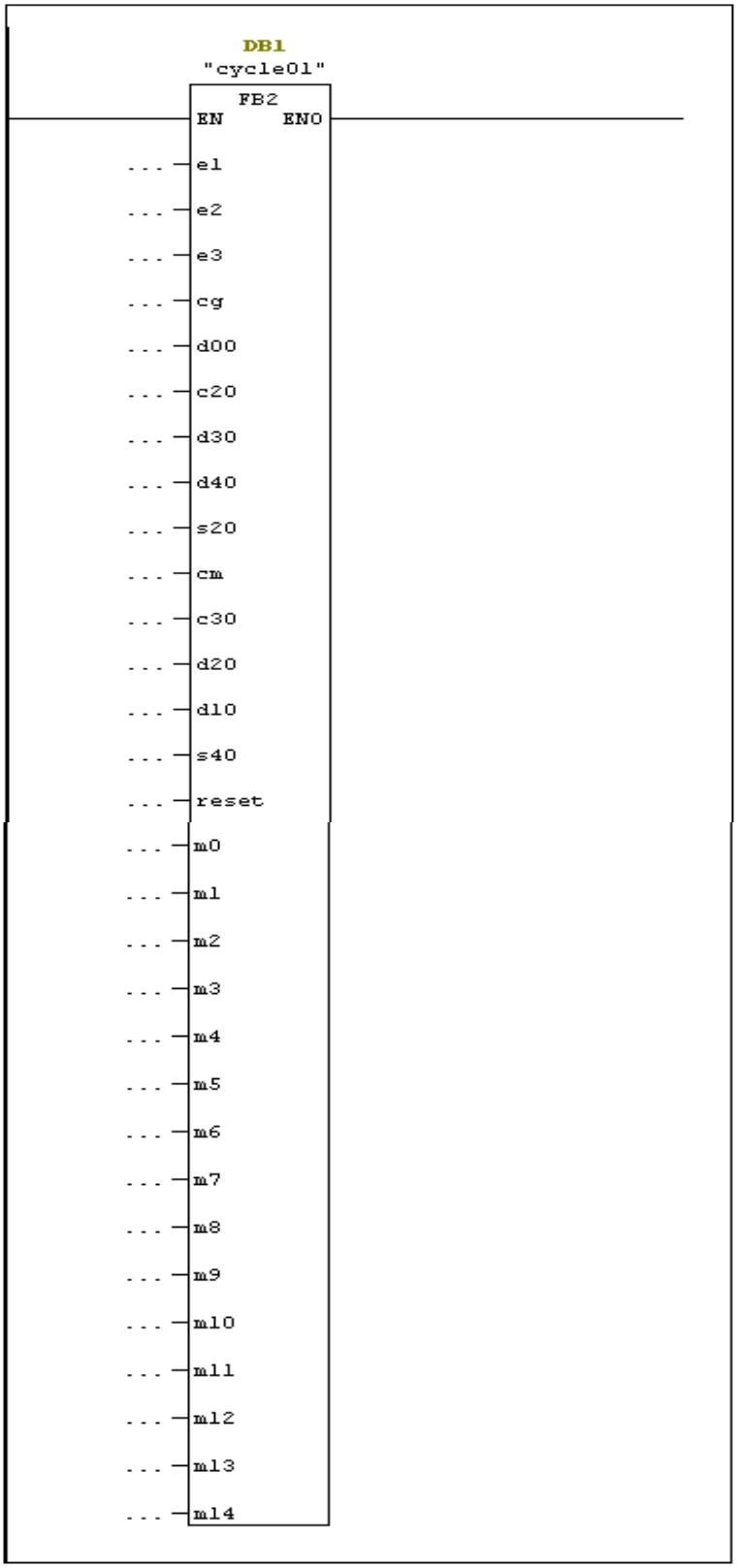


Figure V.14. Exemple sur le bloc FB

V.19. Création de l'interface HMI du projet par WinCC Flexible

D'abord nous avons défini le type du pupitre sur lequel les informations seront transmises, pour notre application nous avons utilisé un pupitre OP (poste opérateur) de type multi panels MP 370 12 Key, lié à l'automate par une liaison MPI.

V.20. La Description des vues

Notre projet de supervision est structuré de manière à pouvoir visualiser l'ensemble de la station.

V.20.1. La Vue d'accueil

C'est une vue de présentation de projet, elle permet l'accès à la navigation entres les différentes vues développées dans cette solution de supervision et cela grâce à un ensemble de boutons configurés sur celle-ci. En cliquant sur chaque bouton, on aura accès à la vue correspondante. Cette vue est représentée dans la (**Figure V.15**) suivante :

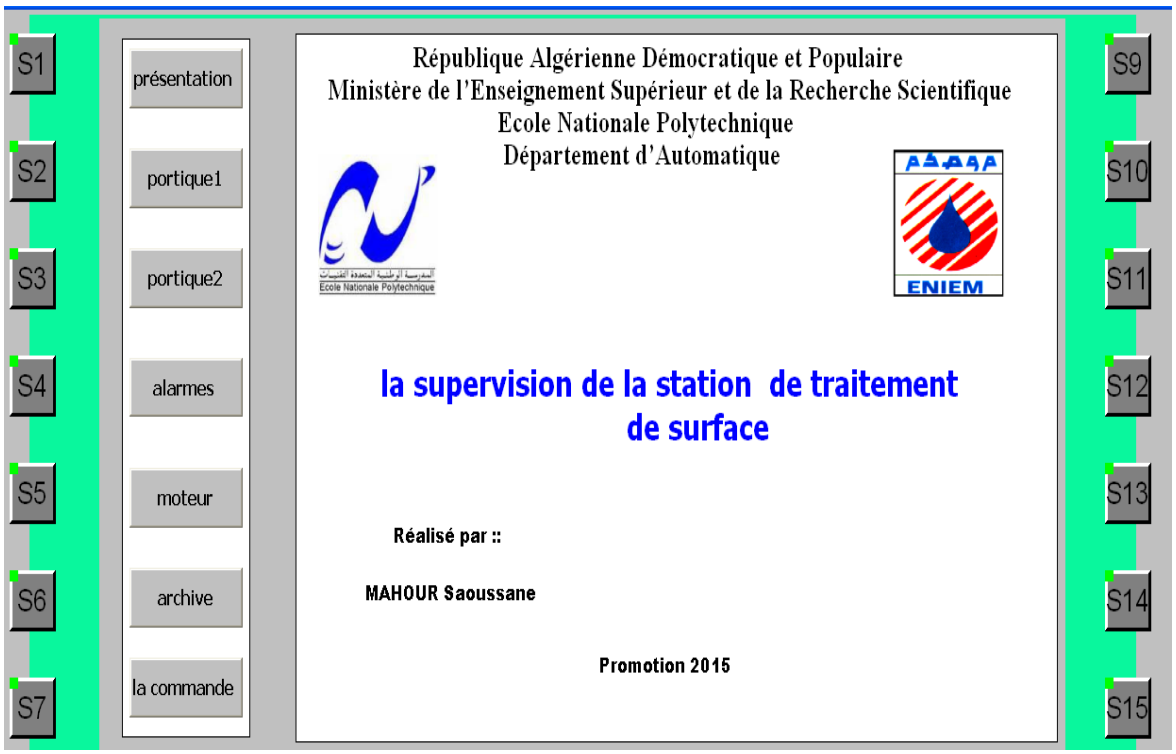


Figure V.15. La vue d'accueil

Même on a configuré les touches de pupitre pour accéder aux autres vues

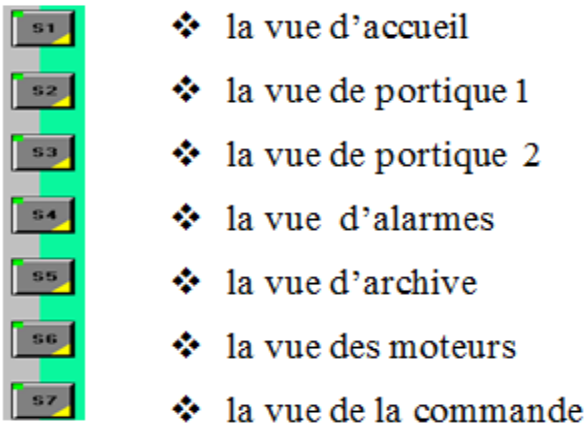


Figure V.16. La configuration les touches de pupitre

V.20.2. la Vue de portique 1

Cette vue montre la position de portique 1, les temporisations correspond au temps d'immersion des corbeilles dans les bains et les états des capteurs de présence de corbeille.

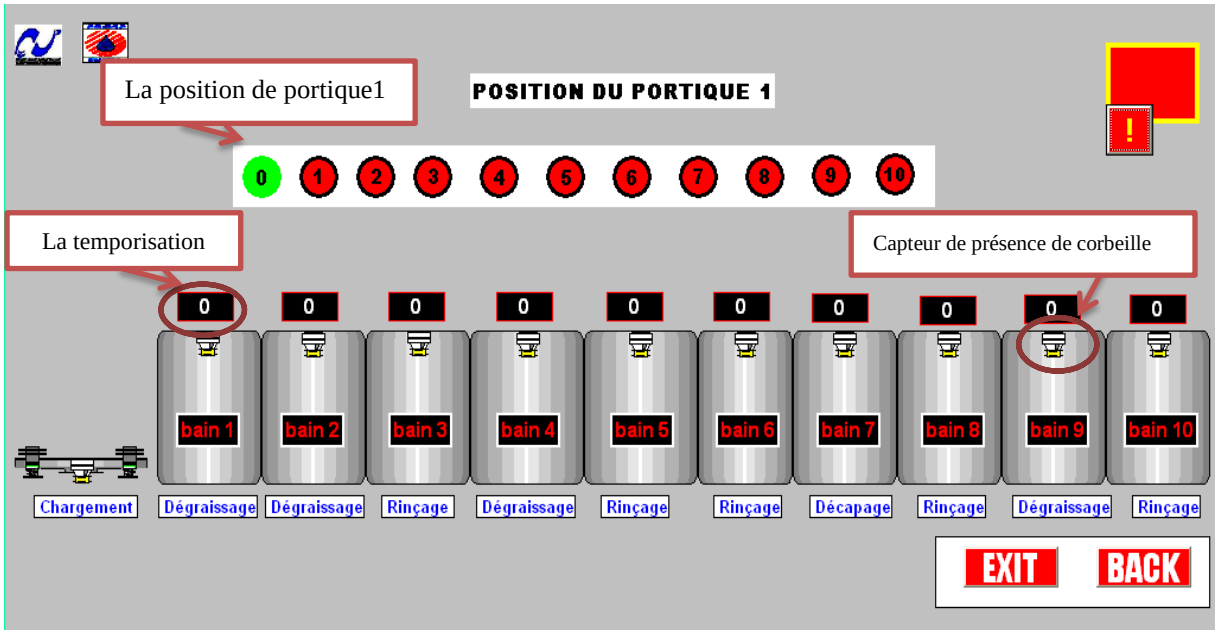


Figure V.17. la vue de portique 1

V.20.3. la Vue de portique 2

Cette vue montre la position de portique 2, les temporisations et états des capteurs de présence des corbeilles.

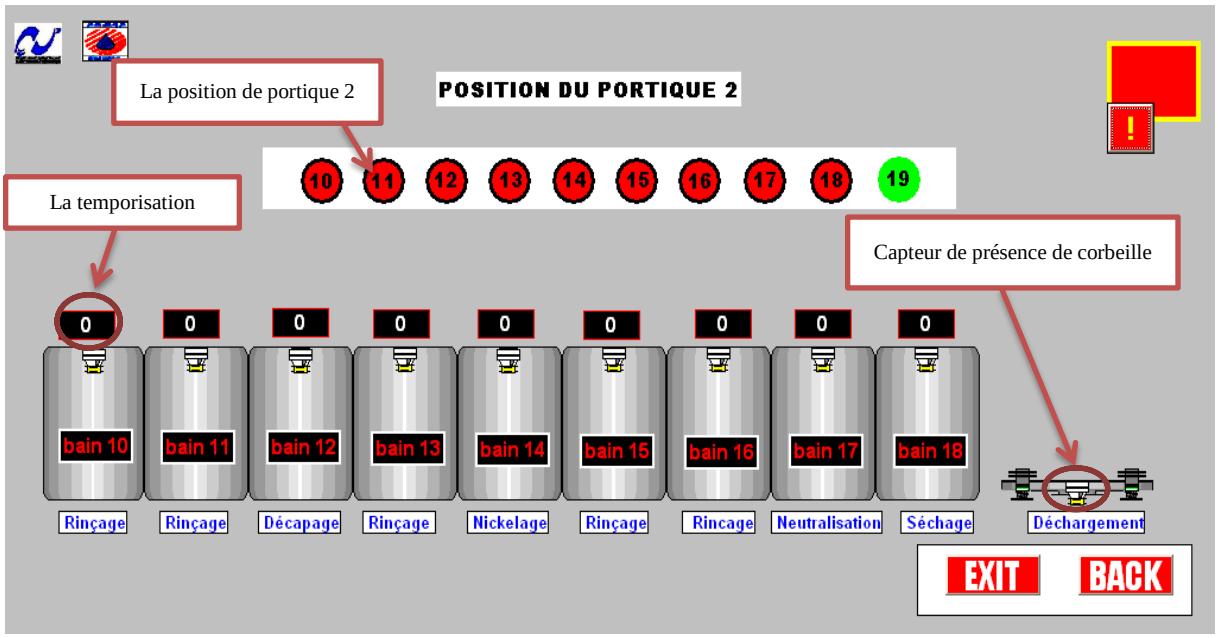


Figure V.18. La vue de portique 2

V.20.4. la Vue de commande

Elle permet de commander l'installation grâce à boutons configurés (Figure **V.19)**

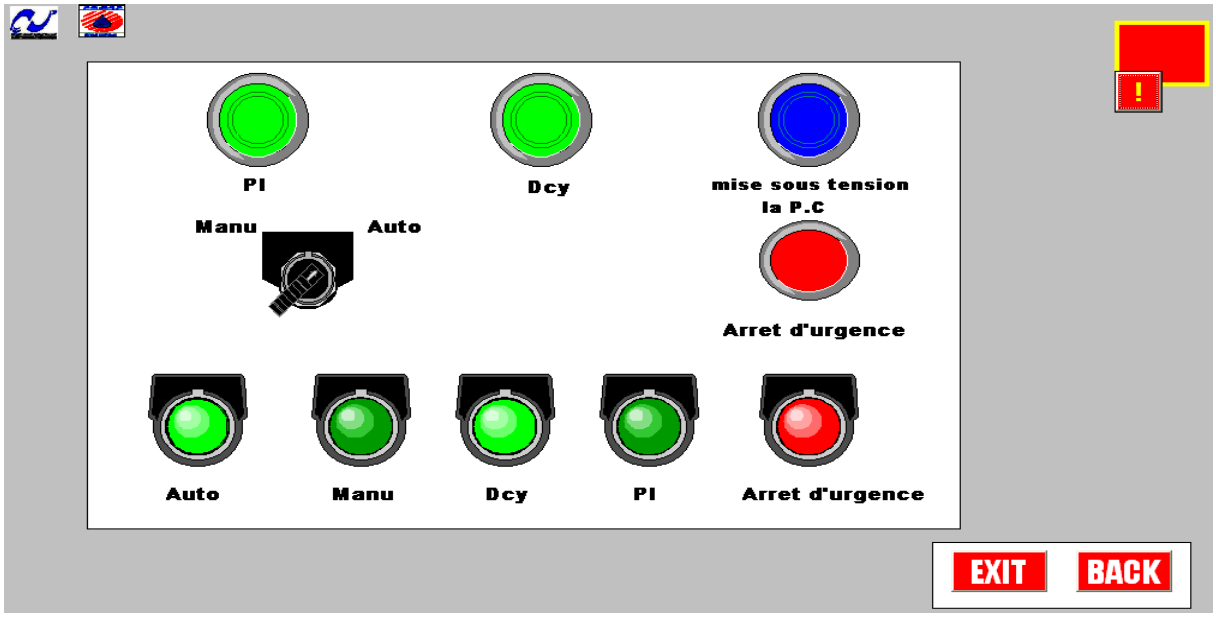


Figure V.19. La vue de commande

V.20.5.La vue des moteurs

Cette vue permet de visualiser l'etat des moteurs , marche a petite /grande vitesse, marche haut/ bas , marche droit /gauche .

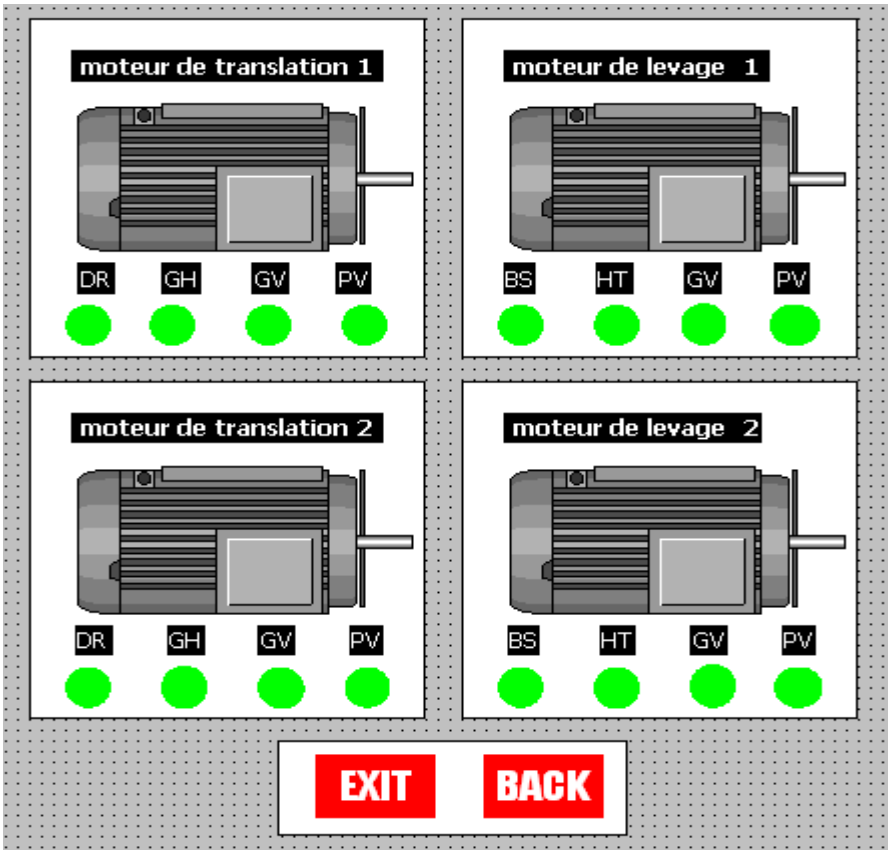


Figure V.20. La vue des moteurs

V.20.6. La Vue des alarmes

Les alarmes montrent les événements ou les états disfonctionnement qui se produisent sur l'installation. Elles peuvent servir au diagnostic des erreurs. Ces alarmes sont immédiatement déclenchées.

Les alarmes utilisées sont des alarmes de types toute ou rien (TOR), et chacune de ces alarmes est composée toujours des éléments suivants: le texte d'alarme qui donne la description de l'alarme, son numéro qui est unique et aussi son temps de déclanchement (date et l'heure).

La figure suivante (Figure V.21) montre la vue de la liste des alarmes utilisées dans le projet :

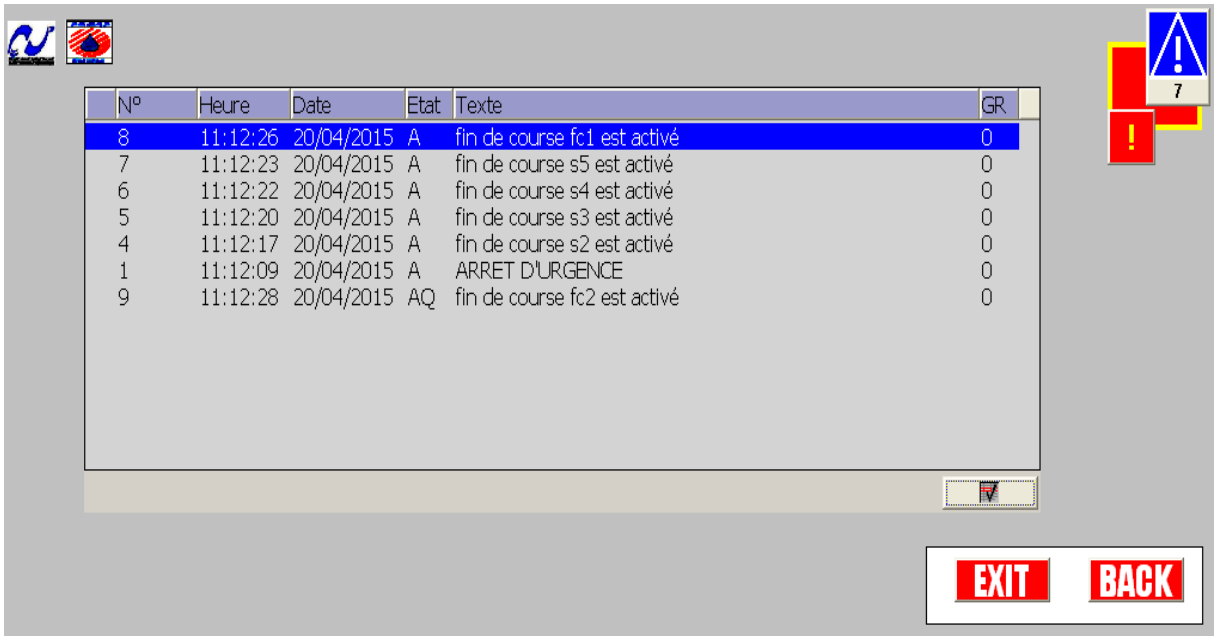


Figure V.21. La vue d'alarme

V.20.7.La Vue "Archive"

Nous avons créé une archive des alarmes dans le but de créer un historique de pannes survenues au niveau de l'installation .Cela permet d'optimiser les cycles de maintenance, améliorer la qualité des produits et assurer le respect des critères de qualité.

Nous avons créé des archives pour enregistrer le nombre de corbeilles par jour.

Figure V.22. La vue d'archive

V.21.Conclusion

Dans ce chapitre, nous avons élaboré le grafcet de l'installation puis nous avons le programmé sous Step 7 et à base WinCC flexible, nous avons fait la supervision du procédé à travers un écran, et on a créé les différentes vues qui permettent de suivre l'évolution du fonctionnement de l'installation. Ceci nous offre une grande flexibilité de contrôle.

conclusion générale

Notre travail a pour objectif d'automatiser l'installation de traitement de surface, Nous avons jugé intéressant et commode de faire appel à la technologie programmée (Automate programmable Industriel), et deux logiciels très performants, STEP7 et WinCC.

On pourrait dire que remplacer les dispositifs classiques de contrôle par les Automates Programmables Industriels se révèle comme étant la solution qui subviendrait aux besoins du monde industriel future voire même actuel qui ne cesse de trop exigé pour ce qui est de qualité et quantité des produits.

Ce projet était une occasion d'approfondir nos connaissances acquises durant notre formation et de les confronter en étude de simulation à un problème d'industrie réel. Cela nous a permis d'acquérir une expérience dans le domaine de la pratique.

Dans des travaux futurs on pourra envisager :

- Elargir le programme pour contrôler tous les paramètres influés sur la qualité de produit comme la température, la concentration ,le pH….etc .
- Automatiser le reste d'équipements tels que les systèmes de ventilation, de traitement des eaux, régénération de produit, chauffage des bains.
- Elargir la supervision pour toute l'installation afin de permettre à l'opérateur de connaitre l'état d'avancement du processus en temps réel et d'intervenir directement sur le pupitre de commande depuis la salle de contrôle.
- Optimiser le cycle de chaque portique pour gain en temps.
- Améliorer le système de sécurité par l'ajout de fonction de notification par envoi de sms et mail.
- Pour des questions de facilité de gestion, de commodité ou de gains potentiels de productivité, il peut être intéressant de ne pas se contenter de lignes contenant simplement des bacs simples avec un unique portique. On utilise alors des cuves de capacité supérieure à un (cuves multi-bacs), ou plusieurs portiques, ou plusieurs tronçons en parallèle.

La bibliographié

[1] M. BERTRAND, « **Automates programmables industriels** » Technique de l'ingénieur, Vol. S 8 015.

[2] G. MICHEL, ≪ **Les A.P.I Architecture et application des automates programmables industriels** ≫, Edition DUNOD, 1987.

[3] P.JARGOT, « **Langages de programmation pour API, Norme IEC 1131-3** », Technique de l'ingénieur, Vol. S 8 030.

[4] B. SCHNEIDER et A. BEURET, « **Automatisation Industrielle** », Yverdon-les-Bains, le 26 avril 2006.

[5] D. DUPONT, D. DUBOIS « **Réalisation technologique du GRAFCET** », Technique de l'ingénieur, Vol. S 8 032.

[6] J.DUMÉRY, « **GRAFCET - Concepts de base** », Technique de l'ingénieur,

Vol. S 7240.

[7] SIEMENS, « **Programmer avec STEP 7** », Réf. 6ES7810-4CA10-8CW0 SIMATIC, 2010.

[8] SIEMENS, « **Système d'automatisation S7-400 Caractéristiques des CPU »** », Réf. 6ES7498-8AA04-8CA0 SIMATIC, 2006.

[9] SIEMENS, «**S7-PLCSIM** », Réf. A5E00425540-01, SIMATIC 2005.

[10] SIEMNS, « **WinCC flexible 2008 Mise en route – Débutants** », Réf. A5E00279568-04, SIMATIC, 2008.

[11] SIEMENS, « **WinCC flexible Getting Started Experts », Réf .**A5E00279909-03, SIMATIC, 2006.

[12] CAR/PP, « **Alternatives de prévention de la pollution à la source dans le secteur du traitement de surfaces** » www.cema-sa.org ,2000.

[13] Guide d'utilisation, « **traitements de surface, Aspects économiques et perspectives** », Techniques de l'Ingénieur, vol.m1422.

[14] J.TERRAT, M.CARTIER, « **Comment poser un problème de traitement de surface** » Techniques de l'Ingénieur, vol. M 1 423.

ANNEXE

A.1.La table de mnémoniques

Les différentes entrées et sorties de notre projet sont déclarées dans le STEP 7 comme suite :

TableauA.1. la table de mnémonique

Adresse	mnémonique	Type de données	Commentaire
E 0.0	p0	BOOL	détecteur de présence de la corbeille au poste de chargement
E 0.1	p1	BOOL	détecteur de présence de la corbeille au bain N°1
E 0.2	p2	BOOL	détecteur de présence de la corbeille au bain N°2
E 0.3	p3	BOOL	détecteur de présence de la corbeille au bain N°3
E 0.4	p4	BOOL	détecteur de présence de la corbeille au bain N°4
E 0.5	p5	BOOL	détecteur de présence de la corbeille au bain N°5
E 0.6	p6	BOOL	détecteur de présence de la corbeille au bain N°6
E 0.7	p7	BOOL	détecteur de présence de la corbeille au bain N°7
E 1.0	p8	BOOL	détecteur de présence de la corbeille au bain N°8
E 1.1	p9	BOOL	détecteur de présence de la corbeille au bain N°9
E 1.2	p10	BOOL	détecteur de présence de la corbeille au bain N°10
E 1.3	p11	BOOL	détecteur de présence de la corbeille au bain N°11
E 1.4	p12	BOOL	détecteur de présence de la corbeille au bain N°12
E 1.5	p13	BOOL	détecteur de présence de la corbeille au bain N°13
E 1.6	p14	BOOL	détecteur de présence de la corbeille au bain N°14
E 1.7	p15	BOOL	détecteur de présence de la corbeille au bain N°15
E 2.0	p16	BOOL	détecteur de présence de la corbeille au bain N°16
E 2.1	p17	BOOL	détecteur de présence de la corbeille au bain N°17
E 2.2	p18	BOOL	détecteur de présence de la corbeille au bain N°18
E 2.3	p19	BOOL	détecteur de présence de la corbeille au bain N°19
E 2.4	c0	BOOL	détecteur de position de portique 1 au poste de chargement
E 2.5	c1	BOOL	détecteur de position de portique 1 au bain N°1
E 2.6	c2	BOOL	détecteur de position de portique 1 au bain N°2
E 2.7	c3	BOOL	détecteur de position de portique 1 au bain N°3
E 3.0	c4	BOOL	détecteur de position de portique 1 au bain N°4
E 3.1	c5	BOOL	détecteur de position de portique 1 au bain N°5
E 3.2	c6	BOOL	détecteur de position de portique 1 au bain N°6
E 3.3	c7	BOOL	détecteur de position de portique 1 au bain N°7
E 3.4	c8	BOOL	détecteur de position de portique 1 au bain N°8
E 3.5	c9	BOOL	détecteur de position de portique 1 au bain N°9
E 3.6	c10	BOOL	détecteur de position de portique 2 au bain N°10
E 3.7	c11	BOOL	détecteur de position de portique 2 au bain N°11
E 4.0	c12	BOOL	détecteur de position de portique 2 au bain N°12
E 4.1	c13	BOOL	détecteur de position de portique 2 au bain N°13
E 4.2	c14	BOOL	détecteur de position de portique 2 au bain N°14
E 4.3	c15	BOOL	détecteur de position de portique 2 au bain N°15
E 4.4	c16	BOOL	détecteur de position de portique 2 au bain N°16
E 4.5	c17	BOOL	détecteur de position de portique 2 au bain N°17
E 4.6	c18	BOOL	détecteur de position de portique 2 au bain N°18

E 4.7	c19	BOOL	détecteur de position de portique 2 au bain N°19
E 5.0	cm01	BOOL	détecteur de portique 1 au milieu des postes 0 1
E 5.1	cm12	BOOL	détecteur de portique 1 au milieu des bains 1 2
E 5.2	cm23	BOOL	détecteur de portique 1 au milieu des bains 2 3
E 5.3	cm34	BOOL	détecteur de portique 1 au milieu des bains 3 4
E 5.4	cm45	BOOL	détecteur de portique 1 au milieu des bains 4 5
E 5.5	cm56	BOOL	détecteur de portique 1 au milieu des bains 5 6
E 5.6	cm67	BOOL	détecteur de portique 1 au milieu des bains 6 7
E 5.7	cm78	BOOL	détecteur de portique 1 au milieu des bains 8 7
E 6.0	cm89	BOOL	détecteur de portique 1 au milieu des bains 9 8
E 6.1	cm910	BOOL	détecteur de portique 1 au milieu des bains 9 10
E 6.2	cm1011	BOOL	détecteur de portique 2 au milieu des bains 10 11
E 6.3	cm1112	BOOL	détecteur de portique 2 au milieu des bains 11 12
E 6.4	cm1213	BOOL	détecteur de portique 2 au milieu des bains 12 13
E 6.5	cm1314	BOOL	détecteur de portique 2 au milieu des bains 13 14
E 6.6	cm1415	BOOL	détecteur de portique 2 au milieu des bains 14 15
E 6.7	cm1516	BOOL	détecteur de portique 2 au milieu des bains 15 16
E 7.0	cm1617	BOOL	détecteur de portique 2 au milieu des bains 16 17
E 7.1	cm1718	BOOL	détecteur de portique 2 au milieu des bains 17 18
E 7.2	cm1819	BOOL	détecteur de portique 2 au milieu des postes 18 19
E 7.3	h0	BOOL	détecteur de la corbeille au poste de chargement
E 7.4	h1	BOOL	détecteur de la corbeille au poste de chargement
E 7.5	H2	BOOL	détecteur de la corbeille au poste déchargement
E 7.6	H3	BOOL	détecteur de la corbeille au poste déchargement
E 7.7	PI	BOOL	Bouton de la position initiale
E 8.0	d0	BOOL	détecteur de position initiale
E 8.1	d1	BOOL	détecteur de position basse (portique1)
E 8.2	d2	BOOL	détecteur de position intermédiaire 1 (portique1)
E 8.3	d3	BOOL	détecteur de position intermédiaire 2 (portique1)
E 8.4	d4	BOOL	détecteur de position haute (portique1)
E 8.5	d5	BOOL	détecteur de position pour le rinçage (portique1)
E 8.6	d6	BOOL	détecteur de position pour le rinçage (portique1)
E 8.7	a0	BOOL	détecteur de position initiale (portique2)
E 9.0	a1	BOOL	détecteur de position basse (portique2)
E 9.1	a2	BOOL	détecteur de position intermédiaire 1 (portique2)
E 9.2	a3	BOOL	détecteur de position intermédiaire 2 (portique2)
E 9.3	a4	BOOL	détecteur de position haute (portique2)
E 9.4	a5	BOOL	détecteur de position pour le rinçage (portique2)
E 9.5	a6	BOOL	détecteur de position pour le rinçage (portique2)
E 9.6	s0	BOOL	fin de course de sécurité pour la position basse de portique 1
E 9.7	s1	BOOL	fin de course de sécurité pour la position basse de portique 2
E 10.0	s2	BOOL	fin de course de sécurité pour la position haute de portique 1
E 10.1	s3	BOOL	fin de course de sécurité pour la position haute de portique 2
E 10.2	s4	BOOL	fin de course de sécurité sur l'extrémité droite du rail
E 10.3	s5	BOOL	fin de course de sécurité sur l'extrémité droite du rail
E 10.4	fc1	BOOL	fin de course de sécurité (anticollision des portiques)

E	**10.5**	fc2	BOOL	fin de course de sécurité (anticollision des portiques)
E	**10.6**	Dr1	BOOL	translation droite de portique 1
E	**10.7**	Gh1	BOOL	translation gauche de portique 1
E	**11.0**	Bs1	BOOL	levage bas de portique 1
E	**11.1**	Ht1	BOOL	levage haut de portique 1
E	**11.2**	Dp1	BOOL	Bouton de Démarrage de portique 1
E	**11.3**	stop1	BOOL	arrêt de portique 1
E	**11.4**	Dr2	BOOL	translation droite de portique 2
E	**11.5**	Gh2	BOOL	translation gauche de portique 2
E	**11.6**	Bs2	BOOL	levage bas de portique 2
E	**11.7**	Ht2	BOOL	levage haut de portique 2
E	**12.0**	Dp2	BOOL	Bouton de Démarrage de portique 2
E	**12.1**	stop2	BOOL	arrêt de portique 2
E	**12.2**	Dcy	BOOL	Bouton de démarrage de cycle
E	**12.3**	auto	BOOL	Commutateur position automatique
E	**12.4**	manu	BOOL	Commutateur position manuel
E	**12.5**	arret d'urgence	BOOL	Bouton d'Arrêt d'urgence
E	**12.6**	DPP	BOOL	bouton mise sous tension
A	**8.0**	MTD1+	BOOL	moteur de translation droit du portique 1 à grande vitesse
A	**8.1**	MTD1-	BOOL	moteur de translation droit du portique 1 à petite vitesse
A	**8.2**	MTG1+	BOOL	moteur de translation gauche du portique 1 à grande vitesse
A	**8.3**	MTG1-	BOOL	moteur de translation gauche du portique 1 à petite vitesse
A	**8.4**	MTD2+	BOOL	moteur de translation droit du portique 2 à grande vitesse
A	**8.5**	MTD2-	BOOL	moteur de translation droit du portique 2 à petite vitesse
A	**8.6**	MTG2+	BOOL	Moteur de translation gauche du portique 2 à grande vitesse
A	**8.7**	MTG2-	BOOL	moteur de translation gauche du portique 2à petite vitesse
A	**9.0**	MLH1+	BOOL	moteur de levage haut du portique 1 à grande vitesse
A	**9.1**	MLH1-	BOOL	moteur de levage haut du portique 1 à petite vitesse
A	**9.2**	MLB1+	BOOL	moteur de levage bas du portique 1 à grande vitesse
A	**9.3**	MLB1-	BOOL	moteur de levage bas du portique 1 à petite vitesse
A	**9.4**	MLH2+	BOOL	moteur de levage haut du portique 2 à grande vitesse
A	**9.5**	MLH2-	BOOL	moteur de levage haut du portique 2 à petite vitesse
A	**9.6**	MLB2+	BOOL	moteur de levage bas du portique 2 à grande vitesse
A	**9.7**	MLB2-	BOOL	moteur de levage bas du portique 2 à petite vitesse
A	**10.0**	V0	BOOL	Voyant indique que mode automatique est activé
A	**10.1**	V1	BOOL	Voyant indique que mode manuel est activé
A	**10.2**	V2	BOOL	Voyant indique que la position initiale est activée
A	**10.3**	V3	BOOL	Voyant indique que le démarrage de cycle
A	**10.4**	V4	BOOL	Voyant indique que l'arrêt d'urgence est activé

TableauA.2. les défferentes cycles de production

Le cycle	FB Associé	DB associé	Macros etape
Cycle01	FB2	DB1	M3
Cycle 12	FB6	DB2	M12
Cycle 23	FB3	DB9	M11
Cycle 34	FB1	DB5	M6
Cycle 45	FB1	DB4	M5
Cycle 56	FB3	DB3	M4
Cycle 67	FB1	DB6	M8
Cycle78	FB4	DB10	M7
Cycle89	FB1	DB8	M10
Cycle 910	FB1	DB7	M9
Cycle 1011	FB4	DB11	M16
Cycle 1112	FB5	DB19	M19
Cycle 1213	FB5	DB17	M18
Cycle 1314	FB5	DB13	M14
Cycle 1415	FB5	DB18	M20
Cycle1516	FB5	DB15	M15
Cycle1617	FB5	DB16	M17
Cycle 1718	FB5	DB12	M13
Cycle 1819	FB5	DB20	M21

A.2.les macros étapes pour le mode manuel

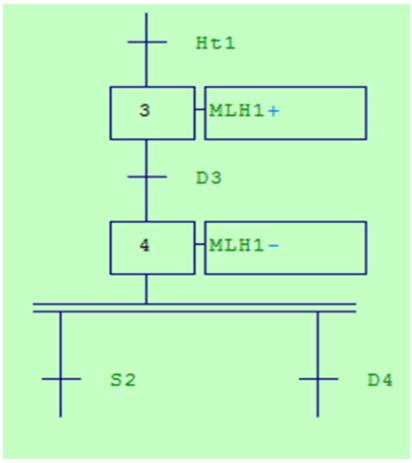


Figure .A.1.Macro étape M24

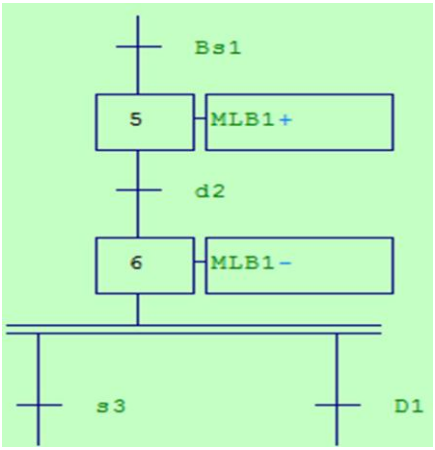


Figure .A.2. Macro étape M25

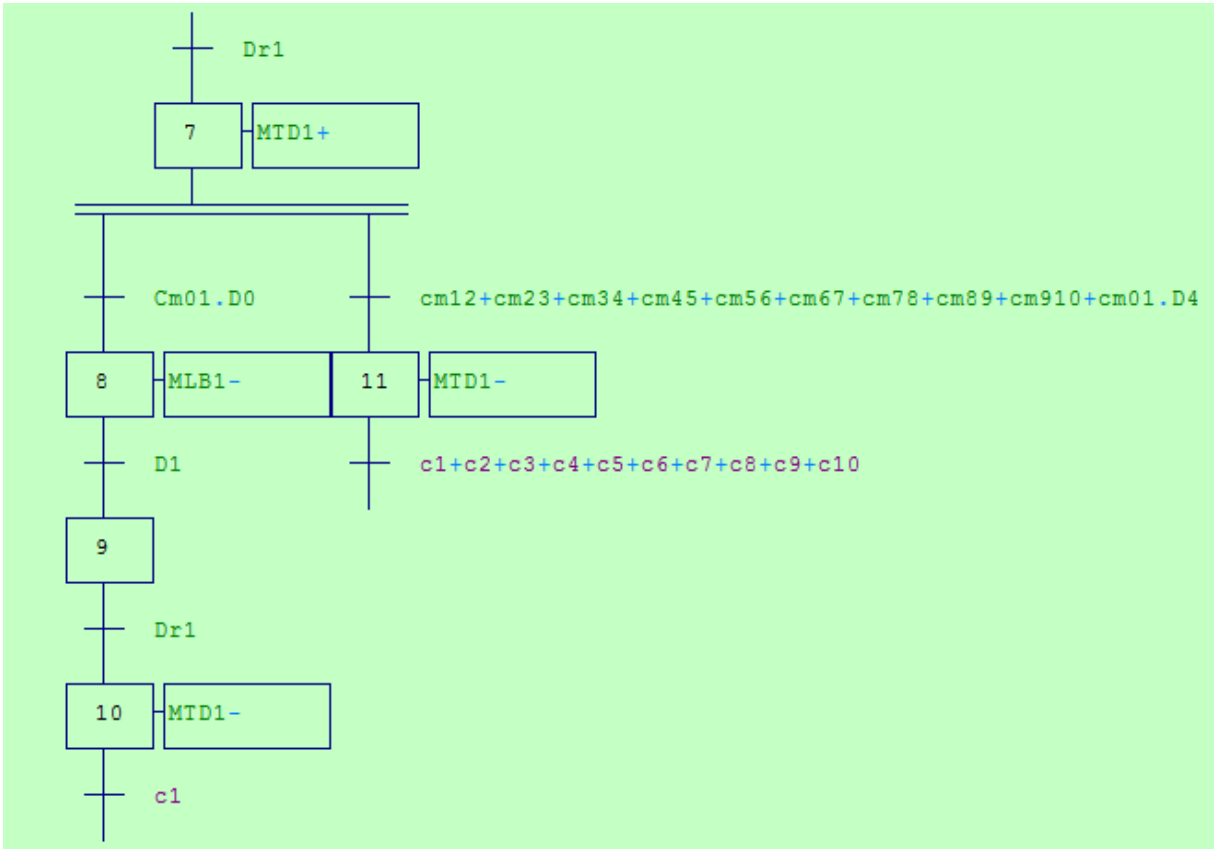


Figure .A.3. Macro étape M26

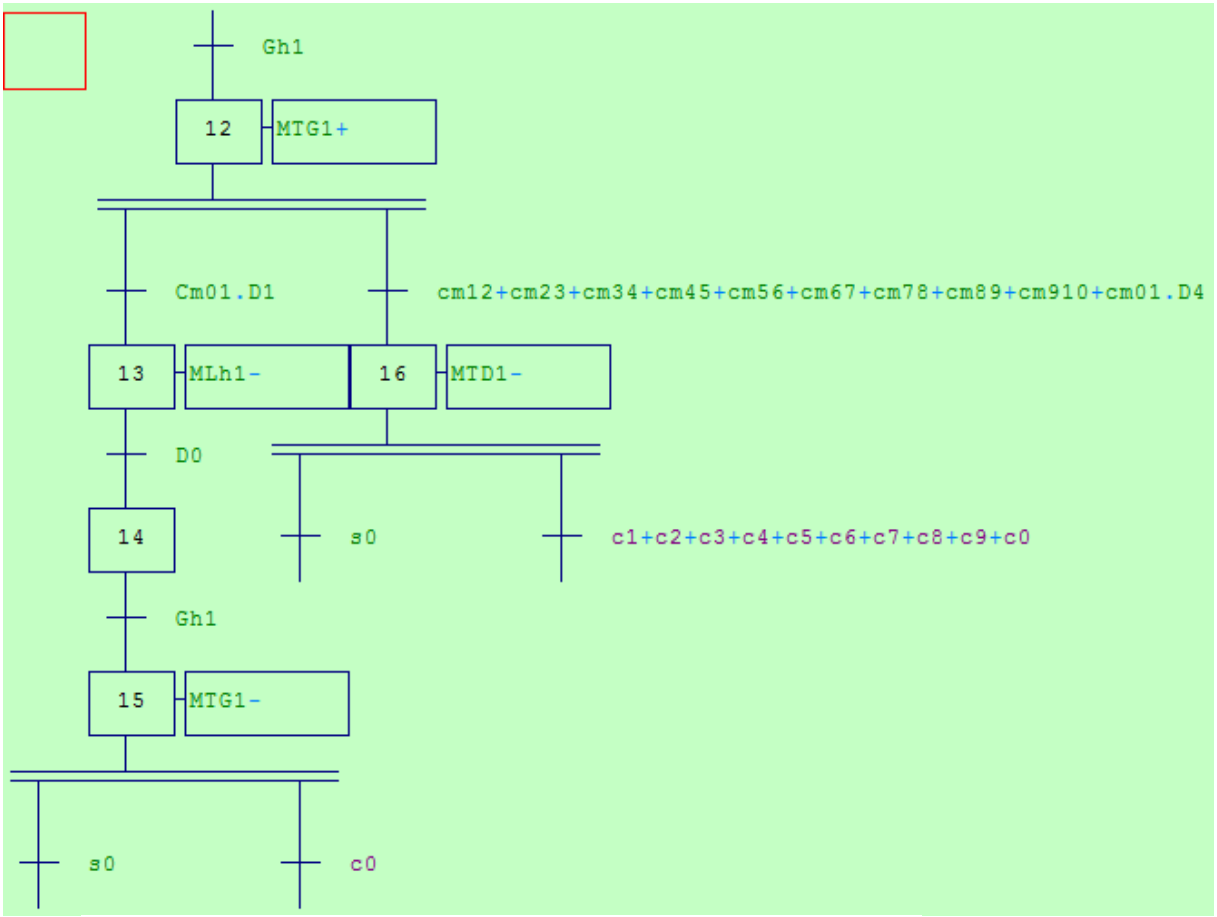


Figure .A.4. Macro étape M27

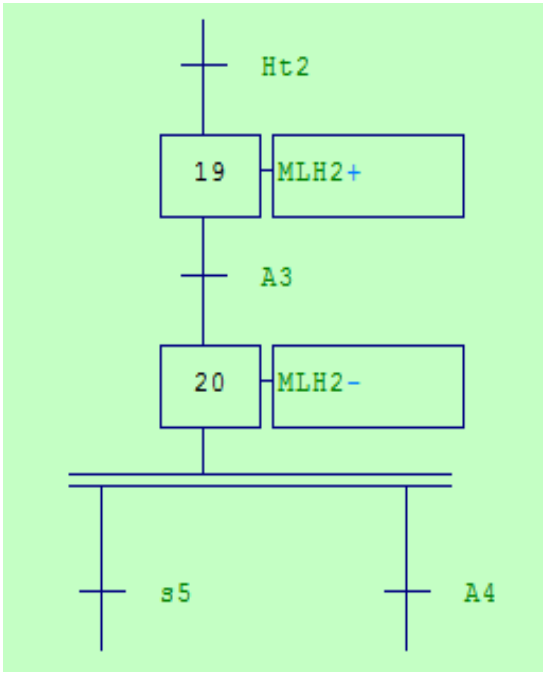


Figure .A.5. Macro étape M28

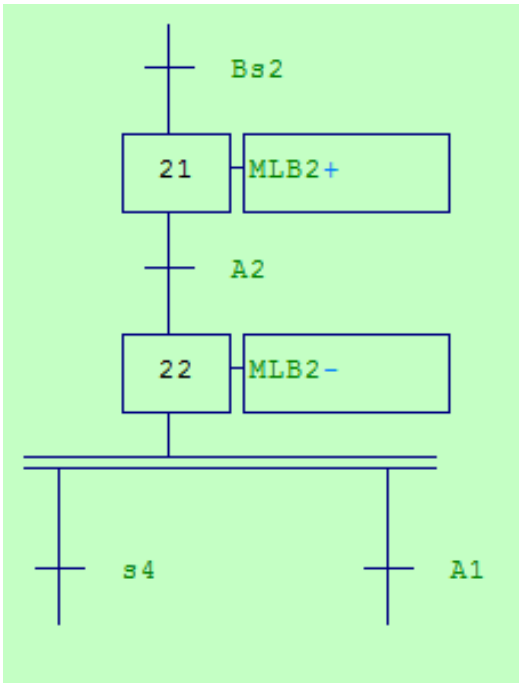


Figure .A.6. Macro étape M29

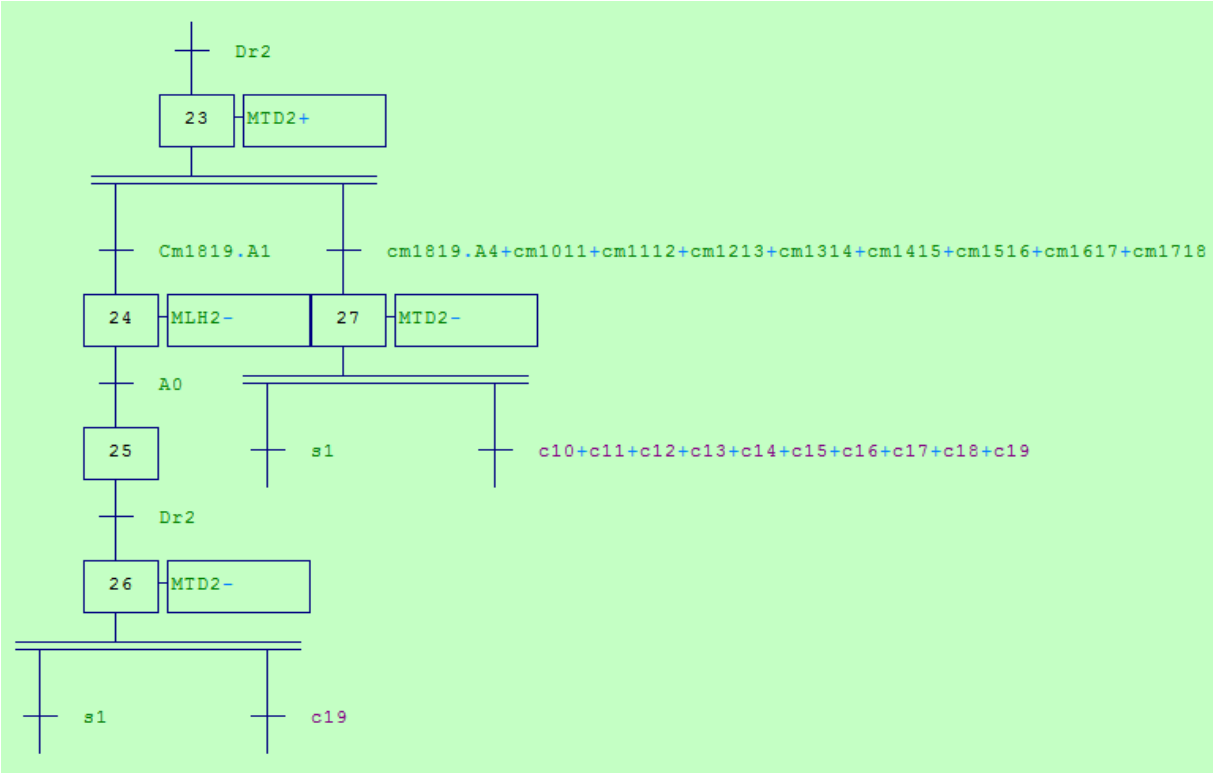


Figure .A.7 Macro étape M30

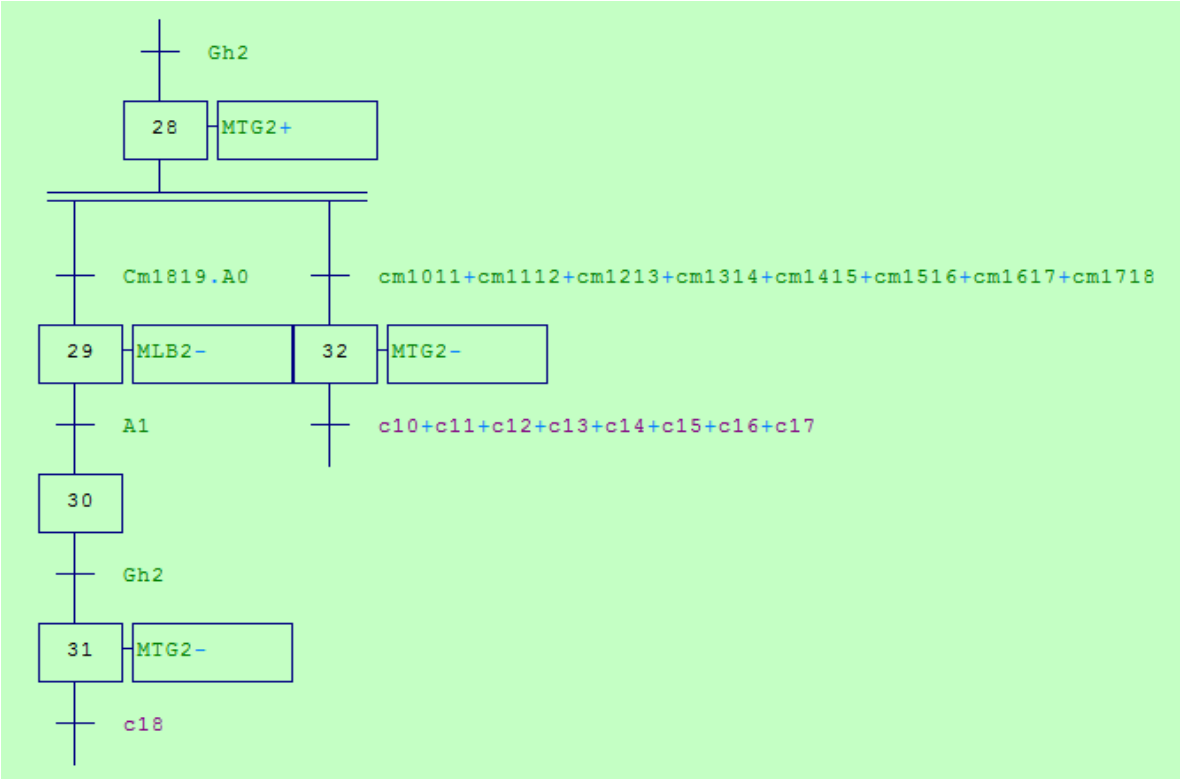


Figure .A.8 .Macro étape M31

A.3.les macros étapes pour le mode automatique

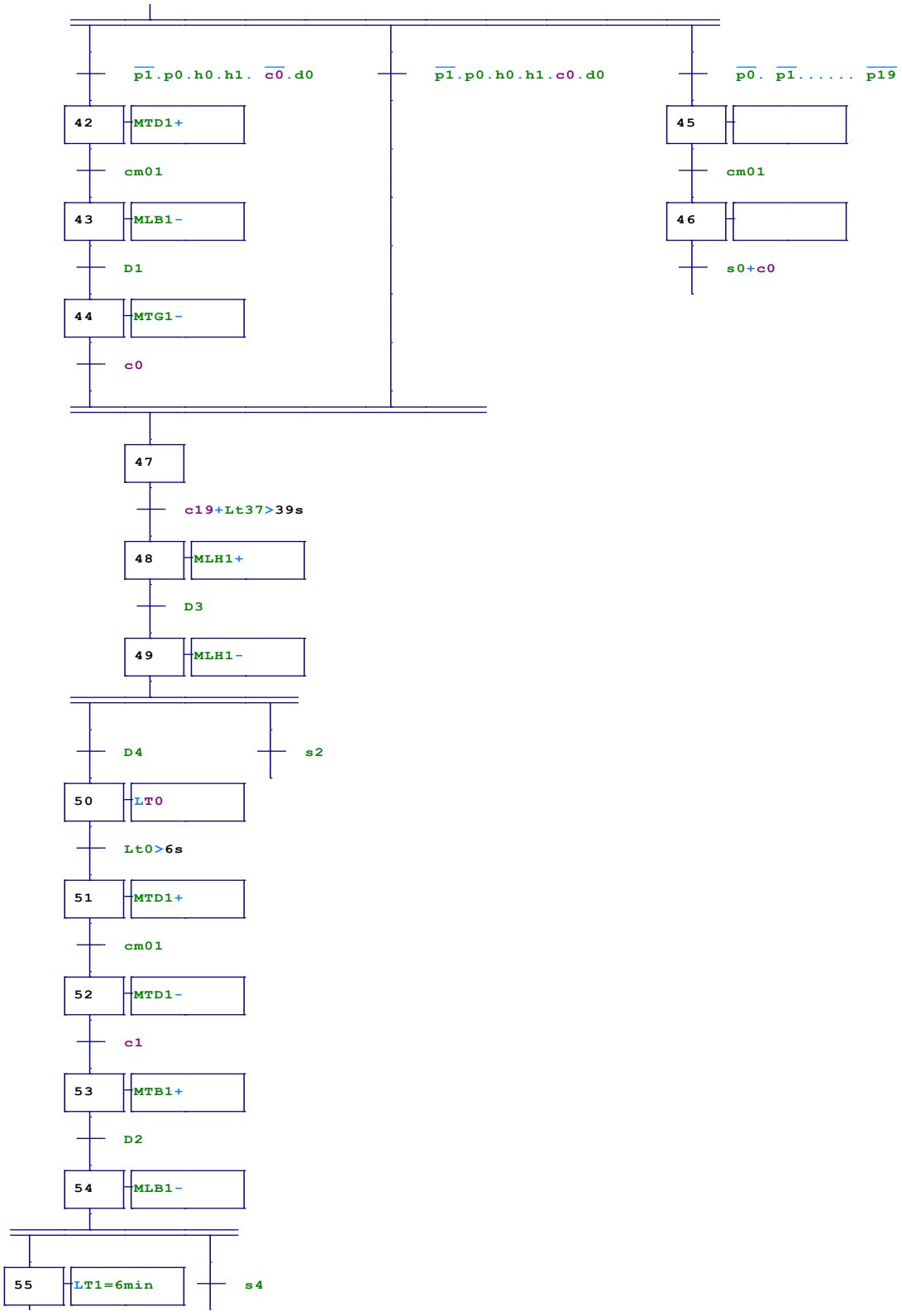


Figure .A.9. Macro étape M3

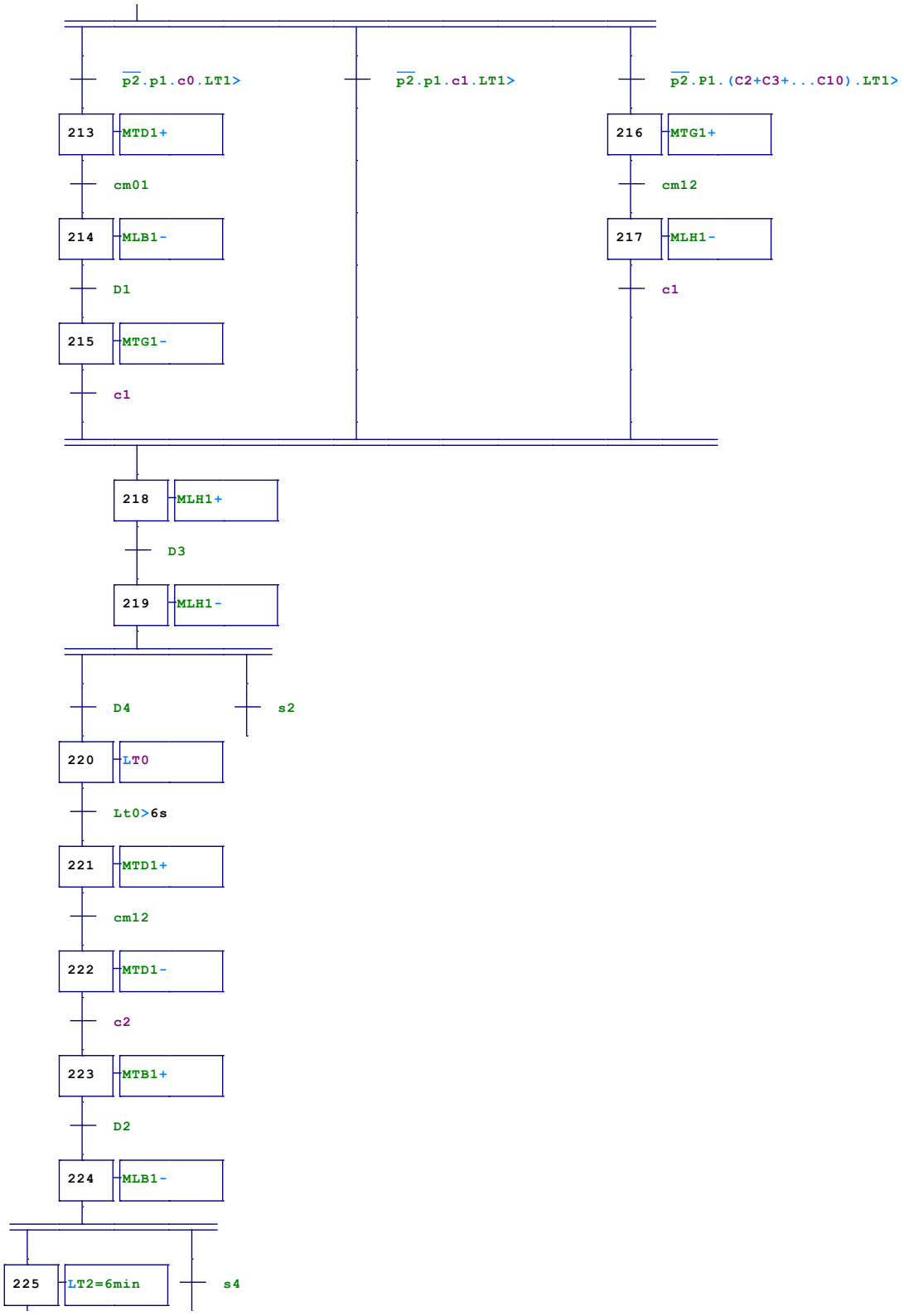


Figure .A.9. Macro étape M12

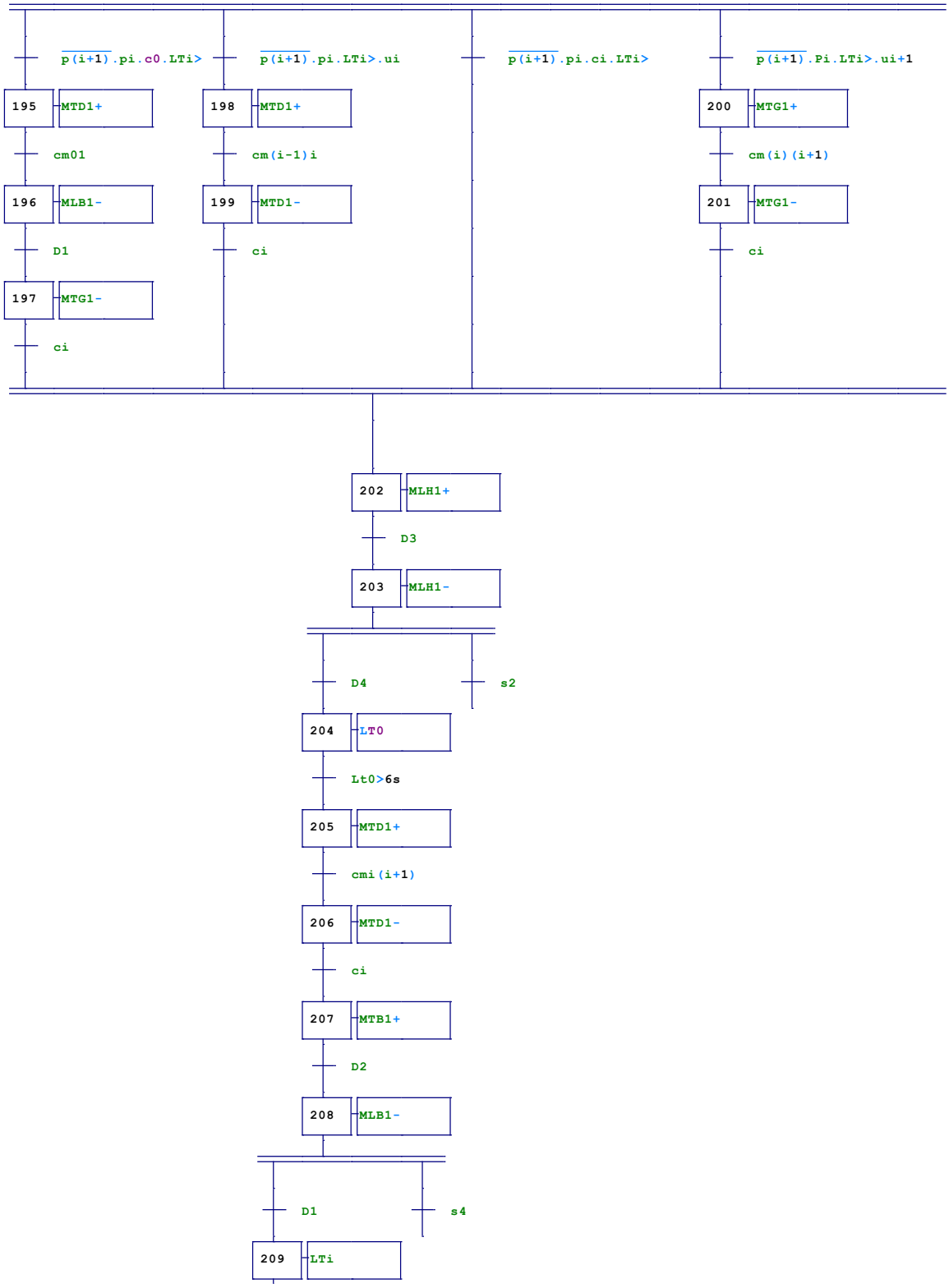


Figure .A.11. Macro étape Mi (i=4,5,6,8,9,10,11,12,13,14,15,17,18,19,20,21)

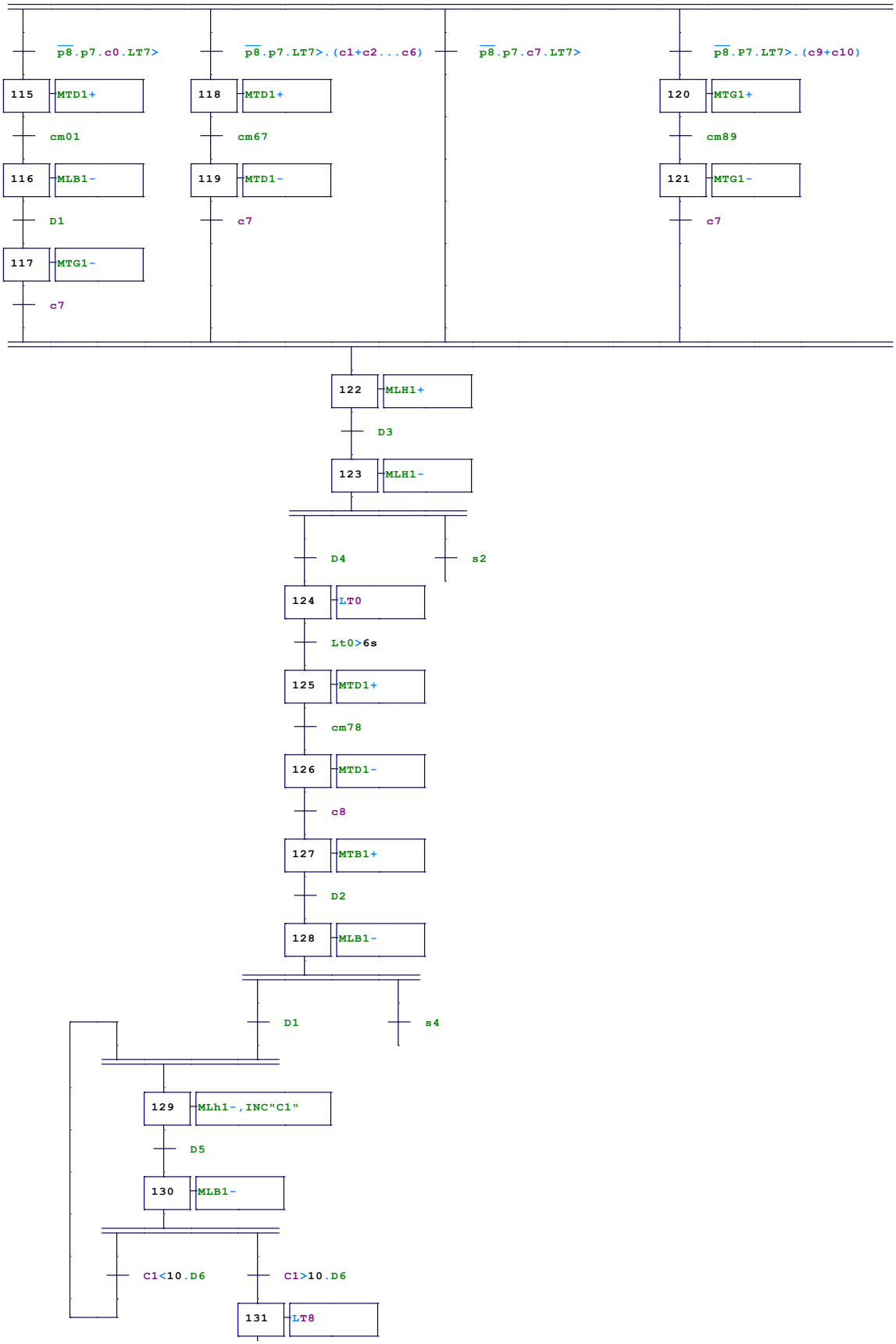


Figure .A.12.Macro étape M7

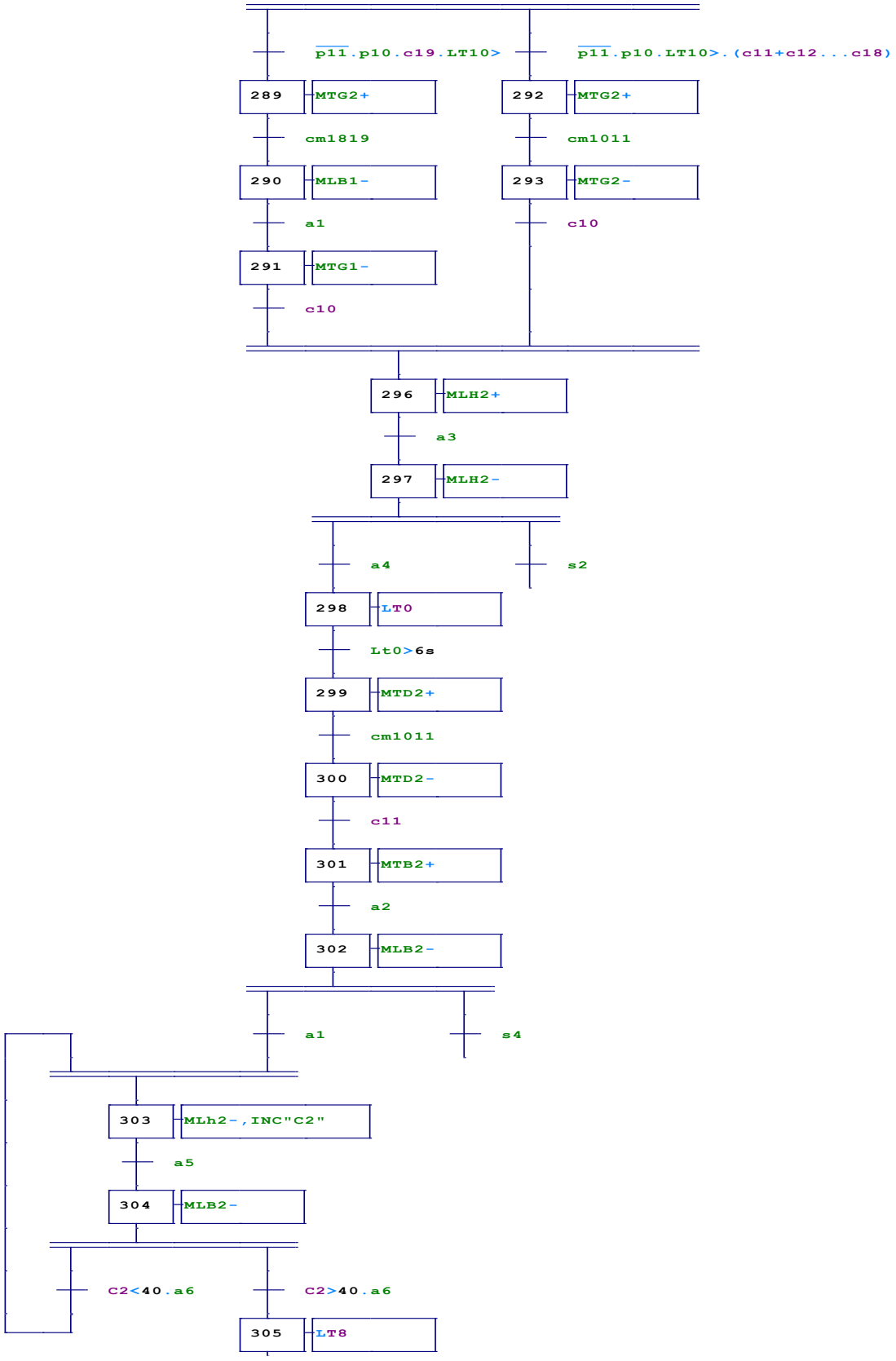


Figure .A.13. Macro étape M16

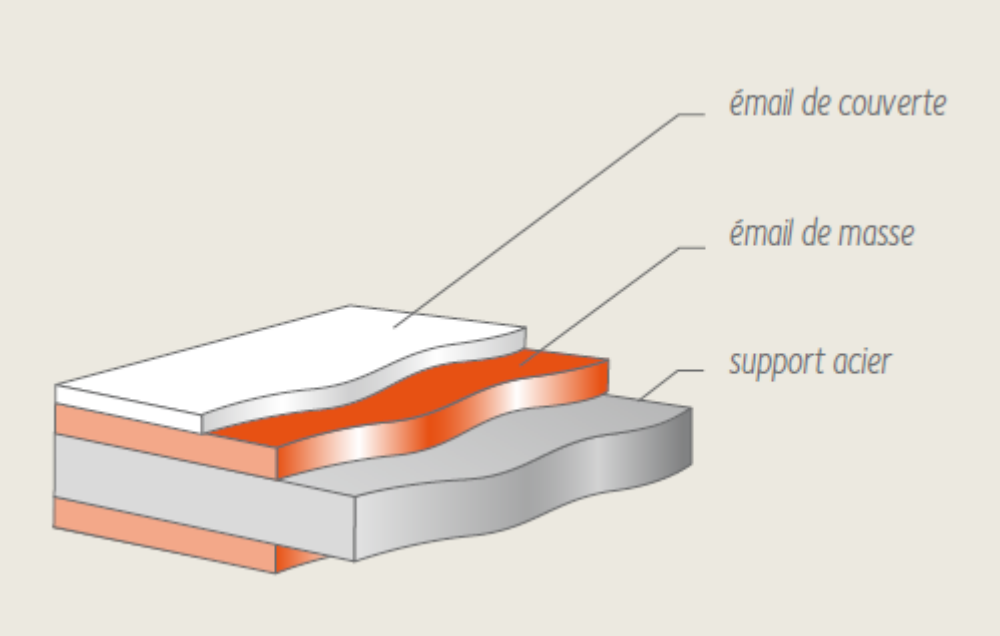
B.1.L'émail

B.1.1.Définition

L'émail est un mélange de matières non organiques naturelles de la famille des verres. Le principal constituant en est la silice, SiO2, qui est le constituant le plus important de l'écorce terrestre. Ce mélange est fusionné à haute température vers 1300 °C, coulé, puis refroidi brutalement et concassé de façon à obtenir la «fritte d'émail», qui sera elle-même transformée en émail par broyage.

B.1.2.L'opération d'émaillage

L'opération d'émaillage consiste à appliquer, puis à cuire une ou plusieurs couches d'émail sur une ou deux faces d'un support acier adapté.



Figue B.1.les différents couches d'email

L'email de masse

L'émail de masse contient des oxydes métalliques (oxydes de Ni, Co, Cu) qui vont permettre son adhérence sur l'acier en créant des alliages avec le fer qui y est contenu. Les oxydes métalliques étant de couleur foncée, il ne peut pas exister d'émail de masse blanc.

Les émaux de couverte

Les émaux de couverte vont donner à la pièce émaillée ses qualités esthétiques. De plus, ils contribuent à améliorer la résistance chimique de la pièce émaillée. Ne contenant absolument aucun agent d'adhérence, ils ne peuvent en aucun cas être utilisés seuls sur un support métallique

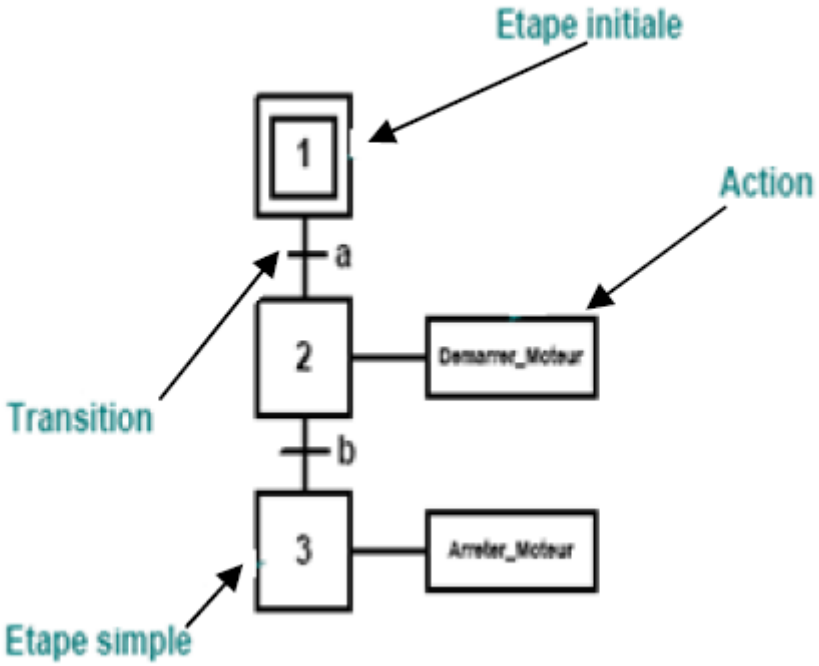
L’opération d’émaillage comprend généralement plusieurs étapes :

- préparation de la surface de la pièce après mise en forme.
- préparation de l’émail.
- application de l’émail sur l’acier.
- séchage.

C.1.Les langages de programmations

C.1.1.Le GRAFCET (SFC) :

Le langage (sequential function chart), ou GRAFCET, est un langage graphique utilisé pour décrire les opérations séquentielles. Le procédé est représenté comme une suite connue d'étapes (états stables), reliées entre elles par des transitions. Une condition booléenne est attachée à chaque transition. Les actions dans les étapes sont décrites avec les langages ST, IL, LD ou FBD.
Le format graphique d'un programme GRAFCET est le suivant :



C1.2.Le Langage FBD :

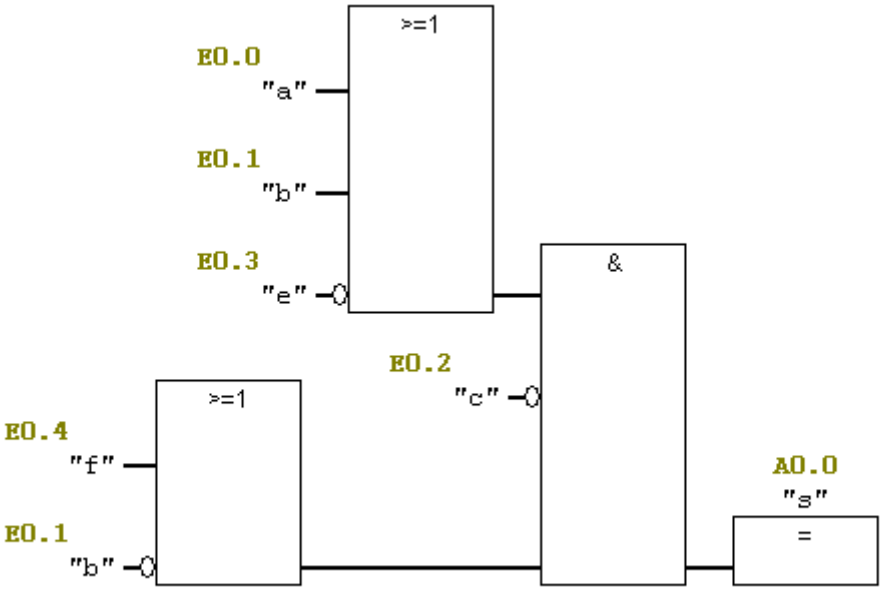
Le diagramme FBD décrit une fonction entre des variables d'entrée et des variables de sortie. Une fonction est décrite comme un réseau de fonctions élémentaires. Les variables d'entrée et de sortie sont connectées aux boîtes fonctions par des arcs de liaison. Une sortie d'une boîte peut être connectée sur une entrée d'une autre boîte.

Les variables d'entrée du diagramme FBD doivent être connectées aux entrées des boîtes fonctions. Le type de chaque variable doit être cohérent avec le type de l'entrée de la boîte correspondante. Une entrée peut être une expression constante, une variable interne ou externe.

Les variables de sortie du diagramme FBD doivent être connectées aux sorties des boîtes fonctions. Le type de chaque variable doit être cohérent avec le type de la sortie de la boîte correspondante.

Une sortie peut être une variable interne ou de sortie, ou le nom du programme édité (pour les fonctions seulement). Quand le nom de la fonction éditée est utilisé comme sortie du diagramme, il représente une assignation de la valeur retournée par la fonction.

On peut réaliser comme exemple l'équation logique : s = (a+b+e).c.(f+b)



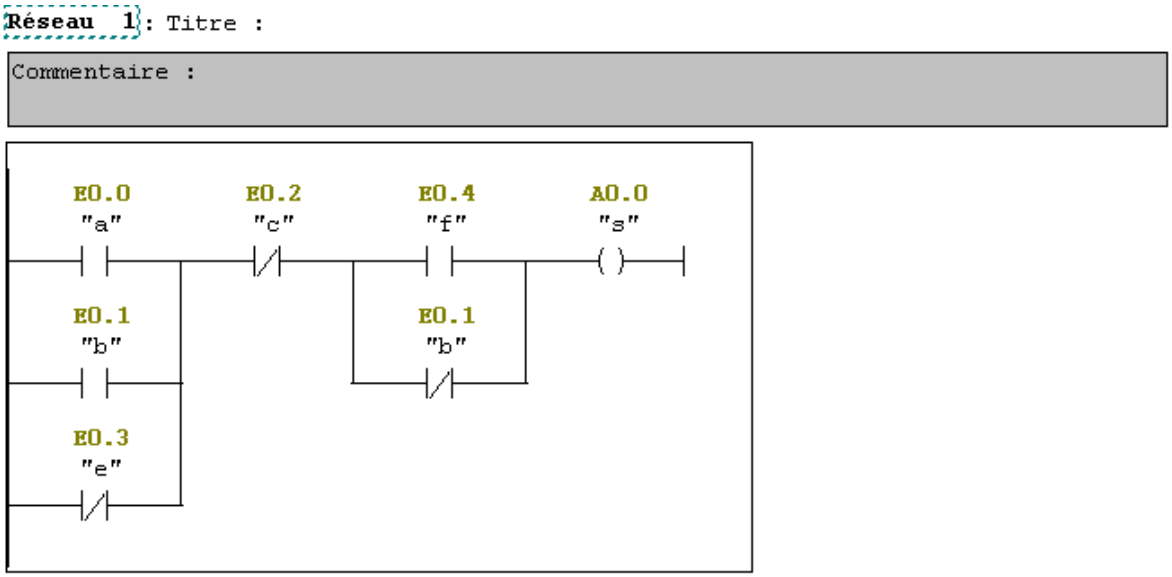
C.1.3.Le langage LD

Le langage Ladder Diagram (LD) est une représentation graphique d'équations booléennes combinant des contacts (arguments d'entrée) avec des bobines (résultats de sortie). Il permet la description de tests et la modification de données booléennes à l'aide des symboles graphiques placés dans un diagramme.

Les symboles graphiques LD sont organisés comme les éléments d'un schéma électrique à contacts. Et sont connectés à gauche et à droite aux barres d'alimentation verticales par des segments de liens.

Chaque segment de lien peut prendre l'état booléen FALSE ou TRUE. Cet état est le même pour tous les segments directement connectés ensembles.

On peut réaliser comme exemple l'équation logique : s = (a+b+e).c.(f+b).



C.1.4.Le langage ST

Le langage ST (StructuredText) est un langage textuel de haut niveau dédié aux applications d'automatisation. Ce langage est principalement utilisé pour implémenter des procédures complexes, difficilement modélisables avec les langages graphiques. Le langage ST peut être utiliser pour la programmation des actions dans les étapes et les conditions associées aux transitions du SFC.

Un programme ST est une suite d'énoncés. Chaque énoncé se termine par un point-virgule (";"). Les noms utilisés par le code source (identificateurs de variables, constantes, mots-clés du langage, etc.) sont délimités par des séparateurs passifs (espace, fin de ligne ou tabulation) ou par des séparateurs actifs, qui ont une signification bien définie et qui jouent le rôle d'opérateurs (par exemple, le séparateur ">" indique une comparaison "Plus grand que").
Des commentaires peuvent être insérés librement dans le texte de programmation. On peut citer plusieurs types d'énoncés standards du ST tels que:

• Assignation (variable := expression;) ;
• Appel de fonction ;
• Appel de bloc fonctionnel ;
• Enoncés de sélection (IF, THEN, ELSE, CASE, etc.) ;
• Enoncés d'itération (FOR, WHILE, REPEAT, etc.) ;
• Enoncés de contrôle (RETURN, EXIT, etc.) ;
• Enoncés spéciaux pour le lien avec d'autres langages, tels que le SFC

Pour la résolution d'une équation de deuxième ordre AX2 + BX+C =0, on écrit le programme suivant en langage ST :

```
Delta := B * B – 4 * A * C;
IF   Delta < 0   THEN
     NSolution : = 0;
ELSE   IF   Delta = 0   THEN
     NSolution := 1;
     X1 := (-B) / (2 * A);
ELSE
     NSolution := 2;
     X1 := ( -B + sqrt (Delta) )  / (2 * A);
     X2 := ( -B – sqrt (Delta) )  / (2 * A);
END_IF
```

C1.5.Le langage IL

Le langage IL, ou Instruction List, est un langage textuel de bas niveau. Les instructions travaillent toujours sur un résultat courant(ou registre IL). L'opérateur indique le type d'opération à effectue rentre le résultat courant et l'opérande. Le résultat de l'opération est à son tour stocké dans le registre.

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

Vue d'ensemble des opérations combinatoires sur bits	
U	ET
UN	ET NON
O	OU
ON	OU NON
X	OU exclusif
XN	OU NON exclusif
O	ET avant OU
U(	ET d'une expression
UN(	ET NON d'une expression
O(	OU d'une expression
ON(	OU NON d'une expression
X(	OU exclusif d'une expression
XN(	OU NON exclusif d'une expression
)	Fermer la parenthèse d'une expression
Affectation	
R	Mettre à 0
S	Mettre à 1
NOT	Négation du RLG
SET	Mettre RLG à 1
CLR	Mettre RLG à 0
SAVE	Sauvegarder RLG dans le bit RB
FN	Front descendant
FP	Front montant

Réseau 1: Titre :

Commentaire :

```
U(
O     "a"                    E0.0
O     "b"                    E0.1
ON    "e"                    E0.3
)
UN    "c"                    E0.2
U(
O     "f"                    E0.4
ON    "b"                    E0.1
)
=     "s"                    A0.0
```

C.2.Tableaux techniques

Les tableaux suivants résument les principales caractéristiques de la CPU S7- 314 IFM

Mémoires	
Mémoire de travail intégrée uniquement	32ko
Mémoire de chargement intégrée	48Ko de RAM 48Ko de FEPROM
Impossibilité d'extension pour la mémoire de travail ainsi que la mémoire de chargement	
Mémentos	
Nombre	2048 bits
Rémanence : réglable par défaut	de MB 0 à MB 143 de MB 0 à MB 15
Mémentos de cadence	Un octet de mémento
Blocs de données	
Nombre	Maximum 127 (DB 0 réservé)
Taille	Maximum 8 Ko
Rémanence : réglable par défaut	Maximum 2 DB, 144 octets de données. Pas de rémanence
Blocs	
Blocs d'organisation (OB)	13
Taille	Maximum 8 Ko
Profondeur d'imbrication : Par classe de priorité Supplémentaire à l'intérieur d'un OB d'erreur	 8 4
Blocs fonctionnels (FB)	128
Taille	Maximum 8 Ko
Fonctions (FC)	128
Taille	Maximum 8Ko
Temporisations /compteurs	
Compteurs S7 Rémanence par défaut Rémanence réglable Plage de comptage	64 Z0 àZ7 Z0 à Z63 0 à 999
Temporisations S7 Rémanence par défaut Rémanence réglable Plage de temps	128 Aucune temporisation rémanente T0 à T7 10 ms à 9990 s

Zones d'adresses (entrées/sorties)	
Numériques	0.0 à 123.7/0.0 à 123.7
Numériques intégrées	124.0 à 125.7/124.0 à 125.7
Spéciales	126.0 à 126.3/124.0 et 124.1
Analogiques	256 à 751/256 à 751
Analogiques intégrées	128 à 135/128 à 129
Mémoire image (non réglable)	128 octets/128 octets
Sauvegarde	
Avec pile	Toutes les données
Sans pile	144 octets

Zones de mémoire et de périphérie de la CPU*

Fonction de test et de diagnostic	
Etat/forçage de variables	Oui
Variable	Entrées, sorties, DB, temporisations, compteurs, mémentos.
Nombre	
Etat de variables	Maximum 30
Forçage de variables	Maximum 14
Forçage permanent	Oui
Variables	Entrées, sorties
Nombre	Max 10
Nombre de points d'arrêt	2
Tampon de diagnostic	Oui
Nombre d'entrées (non réglables)	100

Fonction de test et de diagnostic*

Interface de communication MPI	
Vitesse de transmission	19.2 , 187.5 k bauds

Interface de communication MPI*

Tensions, Courants	
Tension d'alimentation	24V cc
Plage admissible	20,4 à 28,8 V
Consommation (en marche à vide)	Typique 1.0 A

Tensions et courants*

Fonctions intégrées	
Compteur	1 ou 2 selon la configuration utilisateur
Fréquencemètre	Maximum 10 KHz
Positionnement	1 voie

Fonctions intégrées de la CPU*