

RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE
MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITÉ DE TÉBESSA

INSTITUT DES SCIENCES ET TECHNOLOGIE

DÉPARTEMENT DE GÉNIE MINIER

OPTION ELECTROMECHANIQUE

MÉMOIRE

PRÉSENTÉ EN VUE DE L'OBTENTION DU DIPLÔME DE MAGISTER

THÈME:

AUTOMATISATION D'UNE CHAÎNE DE TRAITEMENT DE SURFACE

PRÉSENTÉ PAR:
M. ATTIA MOUSSA

SOUTENU PUBLIQUEMENT LE: 07/07/2009

DEVANT LE JURY:

PRÉSIDENT :	M. HADDOUCHE AL I	PROFESSEUR	U. ANNABA
RAPPORTEUR :	M. KARA MOHAMMED	MAÎTRE DE CONFÉRENCE (A)	U. TÉBESSA
CO-RAPPORTEUR :	M. GUEBLA ABDALLAH	MAÎTRE DE CONFÉRENCE (B)	U. TÉBESSA
EXAMINATEURS :	M. BOUZENADA LAID	MAÎTRE DE CONFÉRENCE (A)	U. TÉBESSA
	M. LABED DJAMEL	MAÎTRE DE CONFÉRENCE (A)	U. CONSTANTINE
	M. DIB DJALEL	MAÎTRE DE CONFÉRENCE (A)	U. TÉBESSA

ANNÉE : 2008/2009

AUTOMATISATION D'UNE CHAÎNE DE TRAITEMENT DE SURFACE

ATTIA MOUSSA
UNIVERSITÉ DE TÉBESSA

REMERCIEMENTS

Je remercie dieu tout puissant de m'avoir permis de mener à terme ce projet qui est pour moi le point de départ de la recherche, source de remise en cause permanent et de perfectionnement perpétuelle.

Qu'il me soit permis de rendre un vibrant hommage à mon encadreur, Mr KARA Mohammed pour avoir bien voulu superviser ce modeste travail et donné de son temps et de son intelligence à la réussite de ce projet qui pour moi représente un modèle de réussite et une source de motivation permanente, pour sa disponibilité, et son sens aigu de l'humanisme pédagogique.

Je profite de cette tribune pour remercier mon Co-encadreur Mr GUEBLA Abdallah, et les personnes qui de passage, ont pu mon apporter leur contribution, que ce soit au niveau des idées qu'à celui des conceptions. Qu'elles trouvent ici l'expression de ma sincère reconnaissance.

Enfin je remercie les membres du jury qui ont bien voulu accepter, et ce nonobstant, leur lourdes et exaltantes responsabilités pour procéder à l'évaluation de ce modeste travail.

DEDICACE

Je dédie ce modeste travail à l'esprit pur de ma mère, qui m'a appris le sens de la tolérance et l'amour, qui sont tendres, et la sécurité de la poitrine et de la pureté, tu ma bien-aimée, la miséricorde de Dieu et le paradis, nous avons recueilli le haut.

Cher à mon père qui m'a appris la patience, de lutte et de pouvoir, la ténacité, et qui donc me l'aide à la voie d'un ambitieux homme Excellence.

Pour mon épouse et compagne de mon chemin: « Samiha ».

Pour ces perles de l'amour qui éclaire mon cœur, chers soeurs et brillant: Bouchra, Zakia, et Hizia.

Pour mes chers frères: Younes et Azzddine.

J'ai appris à être accompagnée par le développement et la construction du sens de la vie.

Pour la génération d'une bonne construction.

Pour la génération de développement et la victoire souhaité.

Dédier ce modeste effort

Moussa ATTIA

RÉSUMÉ

Le traitement de surface est un procédé de fabrication qui consiste à déposer une mince couche de métal à la surface d'une pièce de métal différent pour le protéger et lui conférer certaines propriétés mécaniques.

L'objectif principal est l'exploitation de la chaîne de galvanoplastie dans de bonnes conditions de production et d'optimisation. Il s'agit de trouver un tracé optimum de cycle de pont assurant le déplacement et de conduite optimale de gestion.

Par conséquent, l'élaboration d'un logiciel STEP7 permet de répondre au cahier des charges.

Un ordinateur est nécessaire pour piloter l'ensemble.

MOTS CLES : Automate programmable, Gestion optimale, Grafset, Modélisation, Simulation du procédé.

ABSTRACT

The surface treatment is a process which is deposited a thin layer of metal on the surface of a different piece of metal to protect it with certain mechanical properties. The main objective is the exploitation of the electroplating line in conditions of production and optimization. We need to find a route optimum cycle bridge providing travel and optimal management. Therefore, the development of software STEP7 to meet the specification. A calculator is required to control the whole.

KEYWORDS: Plc, Managed, Grafset, Modeling, Simulation of the process.

ملخص

إن عملية معالجة الأسطح تتمثل في إحداث بفة رقيقة من المعدن على سطح آخر ، فنتشكل قطعة معدنية تحميه و تحمي خصائصه الميكانيكية.

والهدف الرئيسي من هذا البحث هو استغلال سلسلة وحدة معالجة السطح في أحسن الظروف والأحوال الإنتاجية؛ ونحن بحاجة إلى إيجاد الطريق الأسهل و الأمثل لتوفير إدارة أفضل.

لذلك استخدمنا أحد أحدث طرق البرمجيات الصناعية "STEP7"، لتلبية المواصفات المذشودة وفق دفتر الشروط؛ وذلك من أجل التحكم الجيد في جميع محتويات الوحدة الإنتاجية.

الكلمات الرئيسية : الحاكومات القابلة للبرمجة، الإدارة المثلى، المتامن، النمذجة، المحاكاة.

Table des matières

<i>REMERCIEMENTS</i>	<i>I</i>
<i>DEDICACE</i>	<i>II</i>
<i>RESUME</i>	<i>III</i>
<i>TABLE DES MATIERES</i>	<i>1</i>
<i>LISTE DES FIGURES</i>	<i>7</i>
<i>LISTE DES TABLEAUX</i>	<i>8</i>
<i>INTRODUCTION</i>	<i>9</i>
<i>CHAPITRE I : 13 LA CHAÎNE DE TRAITEMENT DE SURFACE</i>	<i>13</i>
<i>INTRODUCTION :</i>	<i>14</i>
<i>1. PRESENTATION GENERALE DU TRAITEMENT DE SURFACE</i>	<i>14</i>
<i>1.1. Définition générale</i>	<i>14</i>
<i>1.2 Les différents traitements</i>	<i>15</i>
<i>2. DESCRIPTION DES TRAITEMENTS DE SURFACE</i>	<i>16</i>
<i>2.1 Dégraissage</i>	<i>16</i>
<i>2.1.1 Dégraissage solvant</i>	<i>16</i>
<i>2.1.2 Dégraissage en bains alcalins</i>	<i>16</i>
<i>2.2 Décapage mécanique</i>	<i>16</i>
<i>2.3 Traitements chimiques</i>	<i>17</i>
<i>2.4 Traitements physico-chimiques</i>	<i>17</i>
<i>2.5 Primaires et promoteurs d'adhésion</i>	<i>18</i>
<i>3. DESCRIPTION D'UN ATELIER DE TRAITEMENT DE SURFACE</i>	<i>21</i>
<i>3.1 Structure d'un atelier</i>	<i>21</i>
<i>3.2 Les bains de traitement</i>	<i>22</i>

3.2.1 Les bains de rinçage	23
4. L'UNITÉ DE TRAITEMENT DE SURFACE A AUTOMATISÉE.....	24
4.1. Description du processus :.....	24
4.1.1. Le chromage :.....	24
4.1.2. Le Nickelage :.....	25
4.1.3. Opérations complémentaires :.....	25
4.1.4. Processus de chromage décoratif :.....	26
4.2. Constitution de la chaîne de traitement de surface :.....	27
4.2.1. Les bains :.....	27
4.2.2. Les chariots :.....	29
4.2.3. Chariots transversaux :.....	31
4.2.3.1. Le bain 415 (bain de rinçage) :.....	31
4.2.3.2. Chariots de chargement :.....	32
4.2.4. Armoire de commande :.....	32
CONCLUSION.....	33
CHAPITRE II: LES AUTOMATES PROGRAMMABLES INDUSTRIELS ET ENVIRONNEMENT DE PROGRAMMATION STEP7.....	34
INTRODUCTION.....	35
1. HISTORIQUE.....	35
2. DEFINITION GENERALE :	36
3. ARCHITECTURE DES API:	36
3.1. Le Processeur :.....	36
3.1.1. L'accumulateur.....	37
3.1.2. Le registre d'instruction :.....	37
3.1.3. Le registre d'adresse :	37
3.1.4. Le registre d'état :	37
3.1.5. Les piles :.....	37
3.2. Les mémoires :.....	37
3.2.1. Mémoire de travail :	38
3.2.2. Mémoire système :	38
3.2.3. Mémoire de chargement :.....	38
3.2.4. Mémoire RAM non volatile :.....	38
3.2.5. Mémoire ROM :.....	38
3.3. Les mémoires d'entrée/sortie :.....	38

3.3.1. Entrées/sorties TOR (Tout ou Rien) :	39
3.3.2. Entrées/Sorties analogique :	39
3.3.3. Module spécialisés :	39
3.4. L'alimentation électrique :	40
3.5. Les liaison :	40
3.6. Eléments auxiliaires :	40
4. PROTECTION DE L'AUTOMATE.....	40
4.1. Les modules à sorties statiques :	41
4.2. Les modules à relais électromagnétique :	41
5. ENVIRONNEMENT.....	41
6. LE STEP 7.....	41
7. APPLICATIONS DE STEP7:	42
7.1. Applications du logiciel de base STEP7 :	42
7.1.1. Gestionnaire de projets SIMATIC Manager :	42
7.1.2. Configuration du matériel HW Config :	42
7.1.3. Editeur de mnémoniques :	42
7.1.4. Editeur de programme :	43
7.1.5. Le S7Graph :	43
7.1.6. Configuration de la communication Net Pro :	43
7.1.7. Diagnostic du matériel :	43
7.2. Possibilité d'extension du logiciel de base STEP7 :	43
7.2.1. Applications techniques :	44
7.2.2. Logiciels exécutables :	44
7.2.3. Interfaces homme/machine :	44
8. PREMIER PAS VERS STEP7 :	44
8.1. Création du projet avec STEP7 :	45
8.1.1 Utilisation de l'assistant de création de projets :	45
8.1.2. Création d'un nouveau projet sans l'assistant de création de projet :	46
8.2. Configuration matérielle :	47
8.3. Définition des mnémoniques :	48
9. ORGANISATION D'UN PROGRAMME UTILISATEUR :	49

9.1. Hiérarchisation dans un projet :	50
9.1.1. Objet station :	50
9.1.2. Objet modules programmables :	50
9.1.3. Objet programme S7/M7 :	50
9.1.4. Objet dossier sources :	50
9.1.5. Objet dossier Blocs :	51
10. BLOCS D'ORGANISATION :	53
10.1. Blocs d'organisation dans la CPU S7-314 IFM :	53
10.1.1. L'OB1 (programme cyclique) :	53
10.1.2. L'OB100 (Mise en route) :	53
10.1.3. Les OB d'alarmes :	54
10.1.4. Les OB d'erreurs :	54
10.2. Classes de priorité des blocs d'organisation :	56
11. DEROULEMENT D'UN PROGRAMME :	57
11.1. Les programmes existants dans la CPU :	57
11.1.1 Le système d'exploitation :	57
11.1.2. Le programme utilisateur :	57
11.2. La mise en route :	58
11.2.1. Démarrage à chaud :	58
11.2.2. Démarrage à froid et redémarrage :	59
11.3. Exécution cyclique :	59
12. LES LANGAGES DE PROGRAMME :	60
12.1. Le cont :	60
12.1.1. Barre d'alimentations et liaisons :	60
12.1.2. Les contacts :	61
12.1.3. Les bobines :	61
12.2. Le LOG :	62
12.3. Le LIST :	63
12.3.1. Présentation :	63
12.3.2. Structure et éléments de LIST :	63
12.3.3. Structure d'une instruction :	63
13. AVANTAGES DE L'UTILISATION DU S7-GRAPH.....	64

14. CREATION D'UN PROGRAMME S7-GRAPH.....	64
14.1 Démarrer SIMATIC- Manager et créer un nouveau projet.....	64
14.2 Insérer la station SIMATIC 300 et ouvrir la configuration matérielle.....	65
14.3 Configurer le matériel et transmettre à l'automate.....	65
14.4 Créer la table des mnémoniques	65
14.5 Insérer le bloc fonctionnel du S7-GRAPH.....	66
14.6 Ouvrir le S7-GRAPH et saisir l'enchaînement	67
14.7 Le principe du langage Graph-7.....	68
14.8 Etape active	68
14.9 Elément du Graph-7.....	68
14.10 Créer le Graph-7 selon l'enchaînement choisi	69
14.10.1 Première étape	69
14.10.2 Deuxième étape.....	70
14.10.3 Troisième et quatrième étapes	70
14.10.4 Insérer un branchement.....	71
14.10.5 Insérer les actions et les transitions de l'étape 5 à 8 et insérer la dernière étape.....	72
14.10.6 Traiter la dernière étape et insérer un retour vers l'étape initiale.....	73
14.10.7 Configurer les paramètres du bloc et enregistrer le bloc fini.....	74
14.11 Ajuster les propriétés du bloc d'organisation et ouvrir OBI.....	75
14.12 Traiter le bloc d'organisation OBI et charger les blocs dans le module.....	76
CONCLUSION.....	77
CHAPITRE III : COMMANDE PAR API POUR LA CHAINE DE TRAITEMENT DE SURFACE.....	78
INTRODUCTION.....	79
1.REALISATION DU GRAFCET DE LA CHAINE.....	79
2. LE GRAFCET DU PROJET TRAITEMENT DE SURFACE :.....	81
2.1. Le Grafcet niveau de sécurité :	81
2.2. GRAFCET du projet :	83
2.2.1. GRAFCET de production:.....	83

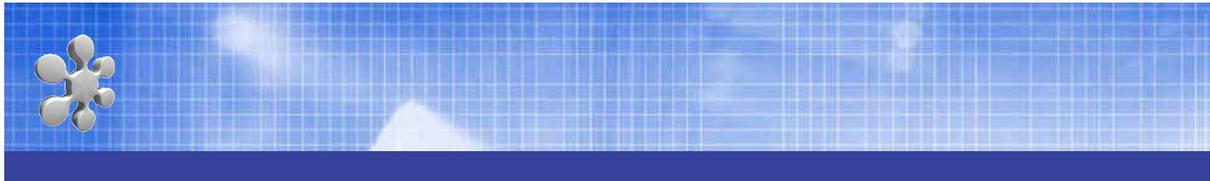
2.2.3. commande manuelle :	86
2.2.3 Mise En PO.....	87
2.2.4 Régénération.....	88
2.2.5 Organisation.....	89
2.3 Etude des modes de marche et d'arrêt.....	90
2. 3.1. - Marche de production cycle par cycle.....	90
2.3.2. - Marche de vérification dans le désordre et mise P.O. dans état initial	90
2.3.3. Marche de préparation (préchauffage des bains)	91
2.3.4. - Marche de clôture (régénération des produits et nettoyage des cuves)	92
2.3.5. Arrêt en vue d'assurer la sécurité, diagnostic et traitement de défaillance, préparation pour remise en route après défaillance.....	92
CONCLUSION.....	95
CONCLUSION GENERALE.....	96
REFERENCES BIBLIOGRAPHIES.....	99
ANNEXES.....	102
Annexe A :SCHEMAS DES ALIMENTATIONS	103
Annexe B :TABLE DE MNÉMONIQUES.....	105
Annexe C :PROGRAMMATION DE LA CHAINE DE TRAITEMENT DE SURFACE	108

Liste des Figures

<i>Figure 1 : Répartition suivant la fonction d'utilisation</i>	<i>14</i>
<i>Figure 2: Répartition par type de traitement de surface.....</i>	<i>15</i>
<i>Figure 3: la séquence des opérations de traitement de surfaces.....</i>	<i>19</i>
<i>Figure 4: Structure générale d'une chaîne de traitement de surface</i>	<i>21</i>
<i>Figure 5 : Schéma des différents types de rinçage</i>	<i>23</i>
<i>Figure 6 : La disposition des baignoires.....</i>	<i>28</i>
<i>Figure 7: Vue de face d'un chariot</i>	<i>30</i>
<i>Figure 8: Cycle de déplacement du chariot.....</i>	<i>31</i>
<i>Figure 9: L'un des deux moteurs installés sur chaque chariot</i>	<i>31</i>
<i>Figure 10: Le déplacement du bain 415.....</i>	<i>32</i>
<i>Figure 11: Fenêtre du SIMATIC manager.</i>	<i>42</i>
<i>Figure 12: Organigramme pour la création de projets sous STEP 7.....</i>	<i>45</i>
<i>Figure 13: Création d'un projet sans l'assistant.</i>	<i>47</i>
<i>Figure 14: Configuration du matériel.....</i>	<i>48</i>
<i>Figure 15: La fenêtre d'édition de mnémoniques.....</i>	<i>49</i>
<i>Figure 16: L'hierarchie d'un projet STEP 7.....</i>	<i>50</i>
<i>Figure 17: Création de blocs de données.....</i>	<i>53</i>
<i>Figure 18: Déroulement d'un programme utilisateur.</i>	<i>59</i>
<i>Figure 19: « OU » logique avec CONT.</i>	<i>61</i>
<i>Figure 20: ET logique avec CONT.</i>	<i>61</i>
<i>Figure 21: Adressage immédiat et direct avec LOG.....</i>	<i>62</i>
<i>Figure 22: Structuration du système.....</i>	<i>80</i>
<i>Figure 23: Vue du bloc fonctionnel FB1</i>	<i>82</i>
<i>Figure 24: Production normale FB2.....</i>	<i>84</i>
<i>Figure 25: Niveau conduite FB 3.....</i>	<i>85</i>
<i>Figure 26: Commande manuelle.....</i>	<i>86</i>
<i>Figure 27: Mise en P.O.....</i>	<i>87</i>

Liste des Tableaux

<i>Tableau 1: Les différents étapes de traitement de surface.....</i>	<i>20</i>
<i>Tableau 2: Récapitulatif des différents bains de traitement.....</i>	<i>22</i>
<i>Tableau 3: Récapitulatif des différents type de rinçage.....</i>	<i>24</i>
<i>Tableau 4: Les différentes opérations du processus de chaine de Traitement de surface</i>	<i>26</i>
<i>Tableau 5: Ensemble de capteurs et boutons d'un chariot.....</i>	<i>29</i>
<i>Tableau 6: Classes de priorité des blocs d'organisation.....</i>	<i>56</i>
<i>Tableau 7: Quelques types d'opérandes.....</i>	<i>63</i>
<i>Tableau 8: Les Blocs du projet traitement de surface.....</i>	<i>81</i>



INTRODUCTION



L'évolution rapide des technologies nouvelles a permis de contourner la plupart des difficultés rencontrées dans le monde industriel, et a fourni plusieurs possibilités pour satisfaire les exigences et les critères demandés tels que la productivité, la sécurité, l'optimisation des coûts de production et l'amélioration des conditions de travail.

L'automatisation des procédés industriels est actuellement l'un des axes où on fait appel, de plus en plus, aux technologies évoluées à mesure que les exigences du monde industriel ont aussi évolué. Parmi celles-ci, figurent les Automates Programmables Industriels (API), qui offrent la solution adaptée aux besoins exigés.

L'automate programmable industriel est l'organe principal de la boucle de réglage placée dans un procédé industriel, en vue de le contrôler. Il a pour tâche principale, la récolte des informations relatives à l'état du système, à partir des différents capteurs via ses interfaces d'entrées, et les traiter pour prendre une décision ; et ainsi commander les actionneurs via ses interfaces de sorties suivant une logique de fonctionnement mise en évidence, par un programme inscrit dans la mémoire.

Le but de ce travail est l'élaboration d'une solution à base d'API pour automatiser une chaîne de production industrielle traditionnelle d'une unité de production de traitement de surface.

Donc notre travail; pour simplifie considérablement le travail de l'homme qui libéré vis-à-vis de la machine, peut se consacrer à des activités plus nobles.

Elimine les taches complexes, dangereuses, pénibles ou indésirables en le faisant exécuter par la machine.

Facilite les changements de fabrication en permettant de passer d'une quantité ou d'un type de production à un autre.

Améliore la qualité des produits en asservissant la machine ou l'installation à des critères de fabrication et à des tolérances qui seront respectées dans le temps.

Accroît la production aussi que la productivité.

Permette de réaliser des économies de matière et d'énergie.

Augmente la sécurité du personnel.

Contrôle et protège les installations et les machines.

Pour réalisé sa; on décompose le travail en deux parties:

La première partie c'est la partie opérative ou partie puissance (l'installation), qui effectue les opérations en exécutant les ordres qui lui sont données par la partie commande.

La partie opérative effectue les opérations mécanique, thermique, physicochimique...etc. sur la forme des produits traités par la machine ou par l'installation et leur fait subir des transformations.

Mais ces opérations peuvent correspondre aussi à un déplacement, à un changement des positions, à des régulations, des assemblages, des montages.

La deuxième partie c'est la partie commande (Automatisme) ou automate, qui élabore les ordres nécessaire à l'exécution du processus, en fonction de consigne qu'elle Les constituants pour le traitement des données: l'automate programmable industriel (API) est un dispositif électronique programmable destiné à la commande de processus industriels par un traitement séquentiel. Il envoie des ordres vers les préactionneurs (partie opérative ou PO côté actionneur) à partir de données d'entrées (capteurs) (partie commande ou PC côté capteur), de consignes et d'un programme informatique.

reçoit à l'entrée des comptes rendus d'exécution qui lui sont fournis par la partie opérative.

La partie commande est assurée par des constituant répondant schématiquement à quatre fonctions de base: détection, traitement des données, la commande de puissance, et le dialogue.

Les constituants pour la détection: interrupteurs de position, détecteurs de proximité, cellules photoélectriques.

Les constituants pour la commande de puissance: variateur de vitesse, démarreur, distributeur.

La constituant pour le dialogue homme machine: écrans de contrôle, les éléments de signalisations sonores et lumineuses.

Le présent travail s'articule autour de trois parties. La première partie présente une description du processus industriel effectué (Traitement de surface), les différents constituants de la chaîne, et les différentes phases de production.

Dans la deuxième partie, on présentera une étude générale sur les automates programmables industriels ainsi que l'API choisi pour effectuer l'automatisation. On présentera aussi, l'interface de programmation STEP7 ainsi que les avantages apportés.

Quant à la troisième partie, elle sera consacrée à l'élaboration du programme de commande de la chaîne de traitement de surface.

Enfin, on terminera l'étude par une conclusion générale qui discutera les avantages apportés et les perspectives visées en terme de réalisation et installation.



CHAPITRE I

LA CHAÎNE DE TRAITEMENT DE SURFACE

INTRODUCTION :

L'unité de traitement de surface que nous avons à automatisée, paraît si complexe que nous ne puissions pas la décrire ici en détail. Néanmoins nous jugeons très nécessaire de décrire le processus de chromage comme étant l'élément de base, et cela sans oublier l'appareillage utilisé et les opérations et étapes que subit le produit brut avant qu'il soit une pièce finie[1].

1. PRESENTATION GENERALE DU TRAITEMENT DE SURFACE

1.1. Définition générale

La surface des matériaux peut posséder un ensemble de caractéristiques physiques (dureté, pouvoir réfléchissant, etc.), chimiques ou électriques (résistivité, potentiel, etc.). Ces caractéristiques peuvent être modifiées par une opération, le traitement de surface, qui peut intervenir dans une phase intermédiaire (25% des cas) ou finale (75% des cas) de l'élaboration du produit.

Le traitement de surface a donc pour but de modifier, transformer la surface de la pièce dans l'optique de lui conférer de nouvelles propriétés telles que, par exemples, résistance à la corrosion, à l'usure ou modification de l'aspect apparent[2].

L'objectif est le plus souvent que les pièces ou surfaces ne s'oxydent pas ou qu'elles aient une meilleure qualité esthétique. La figure 1 montre la part importante des traitements anticorrosion (45%) et anti-usure (24%). Cette activité ayant des applications très diversifiées, de nombreux industriels l'utilisent dans leur procédé de fabrication. Le traitement de surface intervient principalement dans les secteurs de l'automobile, des télécommunications, de l'électronique, de l'aérospatiale, de la bijouterie ou de la quincaillerie.

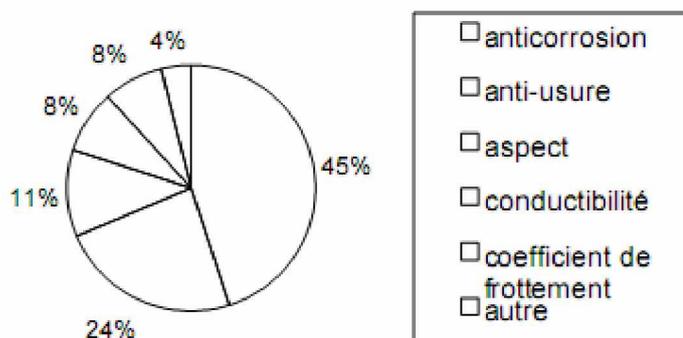


Figure 1 : Répartition suivant la fonction d'utilisation

1.2 Les différents traitements

Une trentaine de techniques de traitement de surface regroupée en cinq familles, dont la figure 2 donne la répartition, constitue l'essentiel de l'activité:

- les traitements par voie aqueuse (ex : dépôts électrolytiques) et voie sèche (ex : dépôts physiques en phase vapeur)
- les traitements de conversion (ex : oxydation anodique)
- les peintures
- les traitements thermochimiques (traitement à haute température)
- les traitements mécaniques (ex : grenaillage)

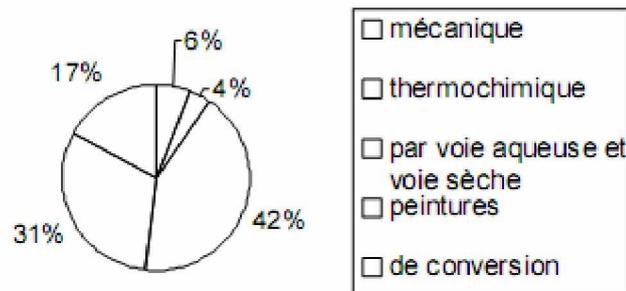


Figure 2: Répartition par type de traitement de surface

Au cours de notre étude, nous nous sommes intéressés aux traitements par voie aqueuse qui constituent l'essentiel des traitements et revêtements de surface soit 42%. Ils concernent trois fonctions[3]:

- La fonction anticorrosion
 - revêtement protecteur
 - traitement de conversion
- La fonction technique
 - préparations de surface (dégraissage, décapage, phosphatation, etc.)
 - traitements mécaniques (tréfilage, laminage, etc.)
 - dépôts
 - enlèvements de matière (usinage chimique ou électrolytique)
- La fonction décorative

- argenture, dorure, nickelage, chromage, etc.
- polissage
- brillantage chimique ou électrolytique

2. DESCRIPTION DES TRAITEMENTS DE SURFACE

2.1 Dégraissage

2.1.1 Dégraissage solvant

Le dégraissage éliminera les contaminations graisseuses sans attaquer le matériau de base. Le choix du solvant sera fortement déterminé par les conditions de sécurité sur l'environnement et la santé.

Les solvants non halogénés sont inflammables et relativement toxiques, les solvants halogénés ne sont pas inflammables mais présentent une toxicité certaine.

Ce traitement peut être réalisé selon différentes méthodes :

- manuellement, par utilisation de chiffon
- par aspersion
- par trempage en bain solvant, en phase vapeur.

2.1.2 Dégraissage en bains alcalins

En terme de sécurité, le dégraissage en bains alcalins est préférable au dégraissage solvant sans toutefois négliger l'importance des effluents liquides polluants qui émanent de ces bains.

Ces produits sont essentiellement composés des éléments suivants :

- Bases (soude, carbonate de sodium)
- Tensioactifs
- Détergents (silicates, phosphates)

Le dégraissage doit impérativement être suivi d'un rinçage à l'eau distillée ou déminéralisée.

2.2 Décapage mécanique

Le décapage mécanique peut se faire selon différentes méthodes selon les surfaces à traiter :

- Brossage
- Abrasion au papier émeri
- Sablage : la méthode consiste à projeter sur la surface à traiter des matériaux abrasifs entraînés à grande vitesse

- Grenailage

Toutes ces techniques de décapage mécanique génèrent des résidus sous forme de poussières dont l'élimination est indispensable. Elle peut se faire soit par soufflage d'air comprimé soit par rinçage à l'eau et séchage.

2.3 Traitements chimiques

L'attaque chimique en bains acides vise à éliminer les oxydes formés spontanément puis à régénérer d'autres oxydes plus adhérents et donc plus résistants.

Dans la plupart des cas, l'épaisseur du substrat modifié est très faible $< 10\text{nm}$.

- Décapage et passivation en milieu acide (acide chlorhydrique, fluorhydrique, sulfurique, nitrique)
 - Procédés de conversion : phosphatation, chromatisation. Procédés complexes, le principe consiste à créer à partir de phosphates ou de chromates acides solubles des couches insolubles qui sont fixées à la surface du métal traité.
 - Anodisation : traitement par électrolyse en milieu acide qui consiste à réaliser une oxydation forcée et contrôlée afin d'obtenir une couche protectrice passive et homogène[4].

2.4 Traitements physico-chimiques

Ils ont pour objectif d'augmenter l'aptitude au collage des substrats thermoplastiques en faisant apparaître des groupements chimiques compatibles avec les adhésifs.

Il existe plusieurs types de traitements :

- Le flammage : Ce traitement consiste à exposer à une flamme la surface du substrat à traiter. De 0,02 à 0,1 seconde sous une flamme peu oxydante de 1200°C à une distance de 6 à 10 mm. Le polymère est oxydé sur une épaisseur de l'ordre de 4 à 9 nm. On peut aussi substituer la flamme par un courant d'air chaud à 500°C ou par immersion dans un bain chaud solvanté à 40°C .
 - Traitement Plasma basse pression : Les plasmas sont des gaz sous très faible pression excités par décharge électromagnétique. Les électrons possèdent suffisamment d'énergie cinétique pour casser les liaisons covalentes et même produire d'autres ions. Les gaz les plus couramment utilisés étant l'oxygène, l'azote ou l'argon. Epaisseur du traitement de l'ordre de 100 Angstrœm.
 - Corona : Passage d'un courant alternatif de haut voltage entre deux électrodes distantes de 1 à 2 mm. L'air est ionisé et un plasma atmosphérique est ainsi créé.
 - Traitement UV : Le rayonnement UV à 243 nm combiné à l'oxygène de l'air crée de l'ozone et oxyde les substances organiques présentes sur la surface. Ce processus n'est possible que si l'énergie fournie est suffisante pour briser les doubles liaisons oxygène.

2.5 Primaires et promoteurs d'adhésion

La distinction entre primaire et promoteur d'adhésion n'est pas toujours très nette. De façon générale, le rôle du promoteur d'adhésion (silanes, zirconates, titanates) est de

créer des liens chimiques entre le substrat et l'adhésif. Le primaire joue le rôle à la fois de promoteur d'adhésion et de protection du substrat. Celui-ci possède au moins une fonction chimique compatible avec celle de l'adhésif.

L'organigramme ci-dessous résume la séquence des opérations de traitement de surfaces:

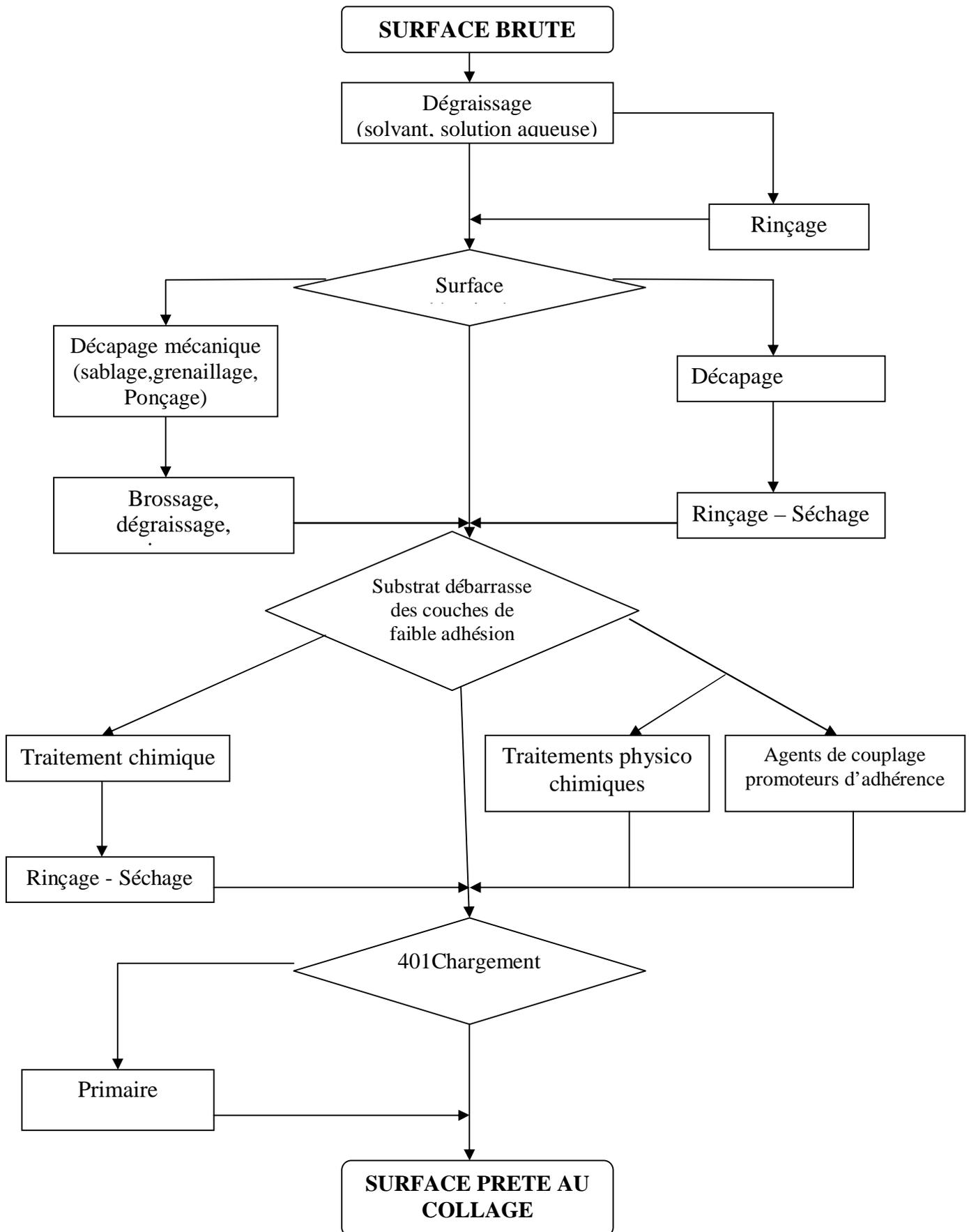


Figure 3: la séquence des opérations de traitement de surfaces

Aluminium et alliages	Forme à sa surface une couche d'oxyde protectrice et adhérente. Décapage chimique H ₂ SO ₄ Rinçage à l'eau désionisée Séchage
Alliages de titane	Décapage chimique HNO ₃ ou HF Rinçage à l'eau désionisée Séchage
Aciers doux	Les débarrasser de leur couche d'oxyde peu adhérente. Décapage chimique H ₂ SO ₄ Rinçage à l'eau désionisée Séchage
Aciers inoxydables	Forment en surface une couche d'oxyde protectrice et adhérente. Décapage chimique H ₂ SO ₄ ou HCl Rinçage à l'eau désionisée Séchage
Cuivre et Alliages	Les débarrasser de leur couche d'oxyde peu adhérente. Collage rapide après traitement. Décapage chimique H ₂ SO ₄ ou HNO ₃ Rinçage à l'eau désionisée Séchage
ABS	Dégraissage à l'acétone Immersion 15 min dans H ₂ SO ₄ Rinçage à l'eau froide et à l'eau chaude – Séchage
PTFE (Téflon)	Traitement plasma Traitement chimique (tetra-etch)
Polyamide (PA)	Dégraissage à l'acétone Traitement par flammage
Polycarbonate (PC)	Dégraissage au méthanol
PMMA	Dégraissage au méthanol Utilisation de primaire
Polypropylène (PP)	Décapage mécanique Traitement par flammage, Corona ou plasma
Polyéthylène (PE)	Décapage mécanique Traitement par flammage, Corona ou plasma
PVC	Pas de traitement particulier Ponçage Dégraissage à l'acétone ou au méthanol

Tableau 1: Les différents étapes de traitement de surface[5]

3. DESCRIPTION D'UN ATELIER DE TRAITEMENT DE SURFACE

3.1 Structure d'un atelier

Les ateliers de traitement de surface sont composés de suites de cuves formant des chaînes de production (Figure 3). Les cuves correspondent soit à des bains de traitement soit à des bains de rinçage. Les pièces subissent donc une succession de trempages dans des bains de traitement et des bains de rinçage afin de subir le traitement désiré.

L'enchaînement nécessaire au traitement complet d'une pièce constitue une gamme de Production.

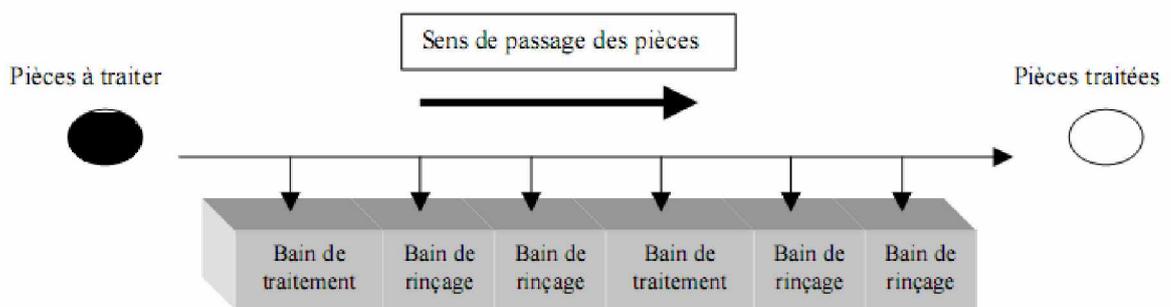


Figure 4: Structure générale d'une chaîne de traitement de surface

Le passage d'un bain à un autre s'effectue à l'aide de différents systèmes de manipulation:

- Le cadre, support sur lequel les pièces sont suspendues. Il est surtout utilisé pour les pièces volumineuses et/ou fragiles.
- Le tonneau, récipient souvent octogonal fermé et mis en rotation. Il est surtout utilisé pour des pièces de petite ou de moyenne taille supportant les frottements dus aux brassages.
- Le panier, récipient clos qui passe d'un bain à un autre sans rotation. Il est utilisé pour des pièces de petite ou de moyenne taille supportant les frottements (mouvement possible).

Ces systèmes sont manipulés soit manuellement, soit à l'aide de portants guidés automatiquement qui descendent et remontent les pièces des bains puis les font passer d'un bain à un autre.

3.2 Les bains de traitement

Les bains de traitement permettent de modifier l'état de surface de la pièce par action de différents réactifs en phase aqueuse. Un atelier de traitement de surface est souvent composé de plusieurs types de bains de traitements qui permettent une transformation complète de la pièce.

Nous distinguons les opérations éliminant les éléments gênants et des opérations ajoutant des éléments spécifiques, les premières précédant généralement les secondes, constituent des opérations de préparation de surface, les deuxièmes sont essentiellement des opérations de dépôts ou de conversion dans le but de communiquer à la pièce des fonctions qu'elle ne possède pas spontanément.

Les différents procédés de traitement regroupent les activités synthétisées dans le tableau 2:

Classe de traitement	Types de traitement	Propriétés – Descriptions
Prétraitements et intertraitements (comprennent toutes les activités d'enlèvements chimiques de la matière)	<u>Dégraissage</u>	§ Permet de nettoyer la pièce de toutes traces de graisse ou d'huile afin de la préparer au revêtement suivant § Agit à l'aide de solvants organique ou aqueux § <u>Dégraissage électrolytique, chimique ou par ultrasons</u>
	<u>Décapage</u>	§ Agit par dissolution chimique de la surface de la pièce métallique § Agit à l'aide d'acide chlorhydrique, sulfurique, nitrique, phosphorique ou fluorhydrique
	Démetallisation	§ Permet d'éliminer la couche superficielle de métal sur la pièce § Traitement chimique ou électrolytique
	Neutralisation	§ Permet de stopper la réaction chimique après traitement § S'effectue en milieu acide ou basique
Revêtement de surface	Argenture, dorure, cadmiage, etc .	§ Revêtement chimiques par utilisation de solution de solution métalliques acides ou basique § Revêtements électrolytiques par réaction de réduction d'un cation métallique $M^{2+} + zé \longrightarrow M$
Traitement de conversion	Anodisation, phosphatation, cémentation, nitruration, etc .	§ Agit sur la modification physico-chimique de la couche superficielle la pièce § Traitement électrolytique, chimique, thermo-chimique ou thermique

Tableau 2: récapitulatif des différents bains de traitement

3.2.1 Les bains de rinçage

Les bains de rinçage ont différents objectifs :

- rendre la surface de la pièce propre et apte à être traitée par les produits chimiques de l'opération suivante,
- diluer le film de liquide polluant entourant la pièce après un bain de traitement afin de limiter la pollution du bain suivant par des produits incompatibles en provenance du bain précédent,
- arrêter l'effet chimique des produits sur la surface de la pièce afin de rendre la pièce manipulable.

Il existe deux grandes classes de rinçages, les rinçages par immersion (Figure 4) et les rinçages par aspersion. Les rinçages par immersion se répartissent eux-mêmes en plusieurs catégories récapitulées dans le tableau 4 en fonction de leur mode d'alimentation (continu ou discontinu) et de vidange (continu, discontinu).

Le fonctionnement des différents rinçages sera détaillé dans la Partie II. Ils jouent un rôle essentiel dans la maîtrise des consommations d'eau et des rejets.

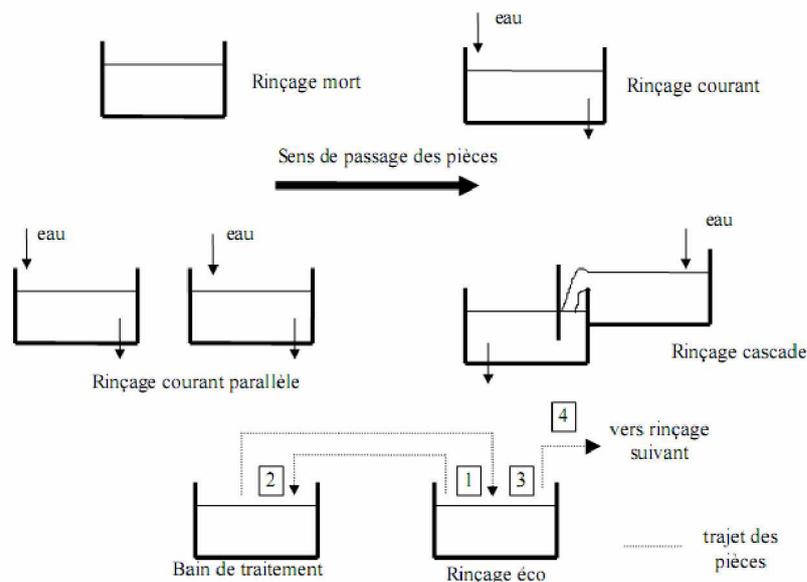


Figure 5 : Schéma des différents types de rinçage

Classe de rinçage	Types de rinçage	Propriétés
Rinçage par immersion	Rinçage statique ou rinçage mort	§ Cuve sans alimentation continue en eau propre § Vidanges périodiques § Permet de réduire les pertes par entraînement
	Rinçage éco	§ Généralement jamais vidangé § Utilisé avant et après le bain de traitement § Permet de stabiliser la concentration dans les bains de rinçage et de diminuer fortement les entraînements de polluants
	Rinçage statique de compensation	§ Placé à la suite d'un bain de traitement chaud, il permet de compenser les pertes par évaporation de ce dernier
	Rinçage courant simple	§ Cuve alimentée en continu en eau propre
	Rinçage courant parallèle	§ Association de plusieurs rinçages courants simples § Alimentation en eau individuelle
	Rinçage cascade	§ Circulation de la pièce à contre courant par rapport à l'eau
<u>Rinçage par aspersion</u>		§ Type douchette ou spray § Utilisé seul ou combiné un autre type de rinçage

Tableau 3: Récapitulatif des différents type de rinçage

4. L'UNITÉ DE TRAITEMENT DE SURFACE A AUTOMATISÉE

L'unité de traitement de surface que nous avons à automatisée, paraît si complexe que nous ne puissions pas la décrire ici en détail. Néanmoins nous jugeons très nécessaire de décrire le processus de chromage comme étant l'élément de base, et cela sans oublier l'appareillage utilisé et les opérations et étapes que subit le produit brut avant qu'il soit une pièce finie[23].

4.1. Description du processus :

4.1.1. Le chromage :

Le chromage est un procédé de revêtement par voie électrolytique sur une surface d'un métal moins résistant et oxydable, généralement un acier en fer par une couche fine et adhérente en chrome métal, dans le but de le protéger contre la corrosion et de lui procurer un aspect esthétique décoratif remarquable par sa brillance métallique éclatante, ainsi que par sa forte résistance à l'usure et à l'abrasion. De telles propriétés confèrent au chrome une grande importance industrielle et le rendent performant.

Conditions opératoires et paramètres de l'électrolyte :

Acide Chromique 180 à 300 g/l
Sulfates 0.8 à 1.1 g/l
Rapport $\text{SO}_4^{2-}/\text{CrO}_3$ 0.4 à 0.5 %
Température 36° à 46°
Densité de courant cathodique 5 à 25 A/dm²

4.1.2. Le Nickelage :

Les dépôts de nickel s'effectuent à partir de solutions de sels simples : sulfate, chlorure, fluo borate, ... Ces dépôts peuvent être mats ou brillants, durs ou tendres, ductiles ou tendus..

Actuellement le nickel brillant trouve son utilisation dans plusieurs domaines tel que l'industrie automobile, électrotechnique, l'industrie des appareils ménagers et la fabrication des appareillages divers et cela par raison d'assurer aux pièces traitées une bonne brillance en vue du chromage consécutif et en outre de les adapter aux exigences de corrosion.

Conditions Opératoires :

Température 35° à 65° optimum 50°
Densité de courant cathodique 2 à 10 A/dm² optimum 4 A/dm²
Rapport surfacique anode/cathode 1/1 à 2/1
Agitation
- air supprimé 20 à 30 m³.n⁻¹.m⁻²
- mécanique 2 à 6 m/min
Filtration continue
Tension superficielle 30 à 40 mN/m

4.1.3. Opérations complémentaires :

Préalablement à un dépôt métallique de chrome, il est important de bien débarrasser la surface à traiter de souillures et de tout corps étranger tel que graisses et oxydes, afin de permettre au dépôt final une bonne adhérence, une uniformité et une dureté remarquable, c'est pour sa qu'on remarque que la chaîne considérée contient plusieurs bains de traitement de surface, tel que le rinçage, le décapage, le dégraissage, ...

4.1.4. Processus de chromage décoratif :

Opération	Concentration (g/l)	Temps de maintien (min)	Température (°C)	Ampérage (A/dm ²)	Remarque
Dégraissage chimique	50	3 – 15	40 – 80		
Dégraissage ultrason	25 – 50	5 – 10	50 – 80		
Rinçage de récupération					
Rinçage à froid					
Dégraissage électrolytique	20 – 50	1 – 3	25 – 40	3 – 4	
Rinçage de récupération					
Rinçage à froid					
Dépassivation	50 – 100		Ambiante		Par Acide
Rinçage à froid					
Activation	10 %		Ambiante		Hcl
Nickelage	Ni ²⁺ : 85-90 NiCl ₂ : 70-80 H ₃ BO ₃ : 50-60 NiSO ₄ : 300		55	2 – 10	
Rinçage de récupération					
Rinçage à froid					
Activation	70				Acide chromique à faible concentration
Chromage	Cr ⁶⁺ : 180-220 SO ₄ ²⁻ : 0.8-1.2 Rpt : 04-0.5%				
Rinçage de récupération					
Passivation					Réduction bisulfate de sodium
Rinçage à froid					
Rinçage à chaud					
Séchage					

Tableau 4: Les différentes opérations du processus de chromage

4.2. CONSTITUTION DE LA CHAÎNE DE TRAITEMENT DE SURFACE :

La chaîne de chromage installée au sein de l'unité de production de robinetterie est constituée de plusieurs éléments dont[6] :

4.2.1. Les bains :

Elle comporte 30 bains métalliques répartis sur deux axes parallèles pour optimiser l'espace de travail. Ils peuvent contenir des solutions chimiques tel que les acides, les bases ...pour le traitement des pièces accrochées à la barre.

Il y a 19 barres qui circulent à la fois dans la chaîne, et cela pour optimiser le temps de traitement des pièces.

Il y a des bains qui sont équipés avec des capteurs de proximité pour détecter la présence de la bar (support portant les pièces a traitées), des capteurs de température, ...

Remarque :

On a constaté que les ne sont pas dotés de capteurs de position permettant de détecter la position des chariots pendant leurs déplacement. Donc un autre système est utilisé (comptage). Mais on remarque aussi que au dessus de chaque on a placé un boue de fer qui sert comme repère pour le bain désigné.

La disposition actuelle des 30 bains est représentée selon la figure qui suit :

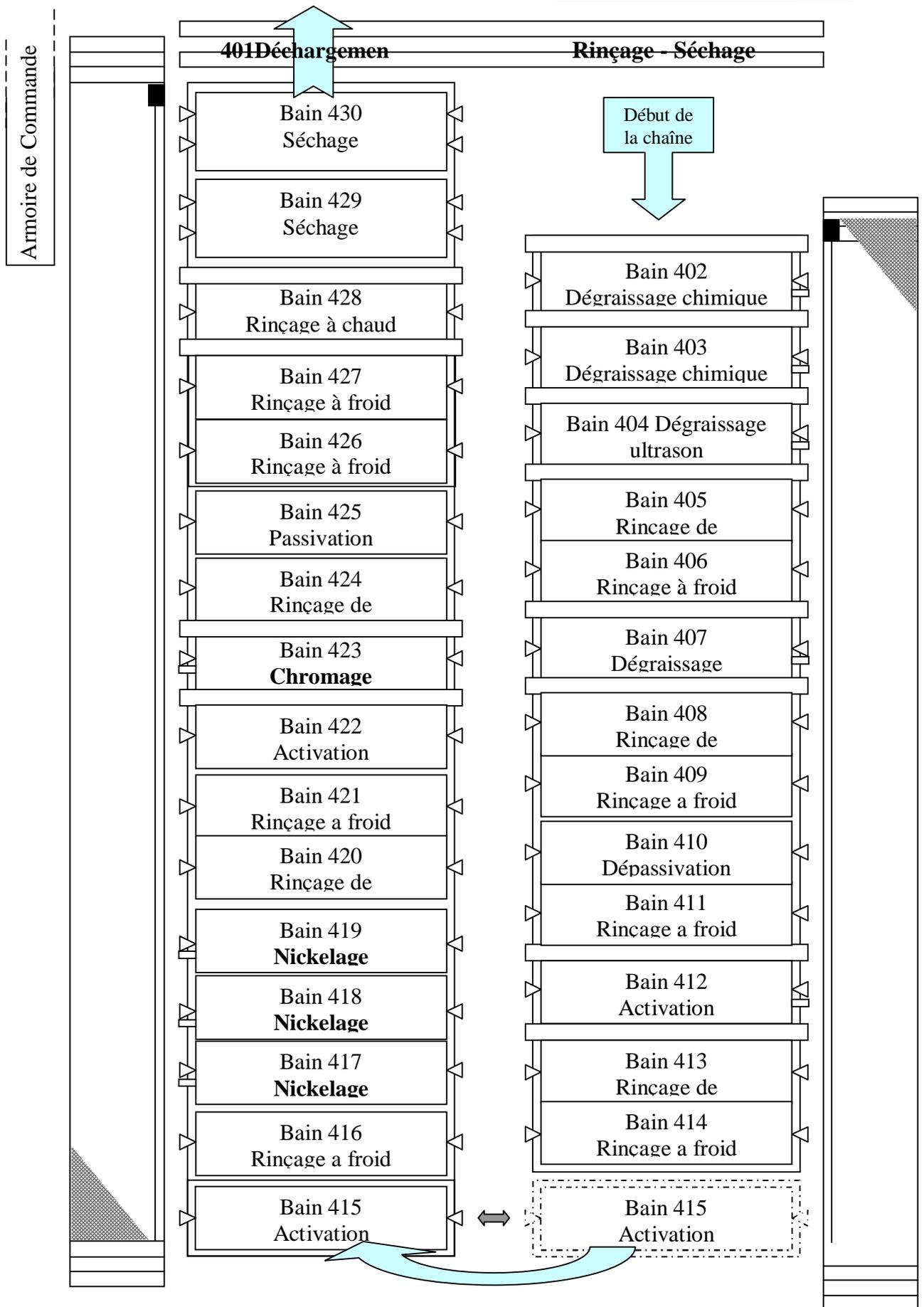


Figure 6 : La disposition des baigns

4.2.2. Les chariots :

La chaîne comporte six chariots, trois sur le premier axe et trois sur le deuxième, montés sur des rails juste au dessus des baignoires.

Chaque chariot est doté de plusieurs de capteurs, boutons, Mentionnés dans le tableau suivant :

Désignation	Type	Rôle
Xb0	Détecteur de proximité	Détection de vitesse lente (avance)
Xb1	Détecteur de proximité	Détection de vitesse lente (recule)
Xb2	Détecteur de proximité	Fin de course (pour mouvement transversal)
Xb3a	Bouton	Commande manuelle pour le levage
Xb3b	Bouton	Commande manuelle pour l'abaissement
Xb3c	Bouton	Commande manuelle pour l'avance
Xb3d	Bouton	Commande manuelle pour le recule
Xb4	Bouton poussoir	Déclenchement de secours
Xb7	Détecteur de proximité	Détection de l'état initiale
Xb8	Détecteur de proximité	Détection de la présence de la barre
Xb9	Came	Limite de levage
Xb10	Came	Limite d'abaissement
Xb11	Came	Détection de la vitesse lente (abaissement)
Xb12	Came	Détection de la vitesse lente (levage)
Xb17	Détecteur de proximité	Stationnement

Tableau 5: Ensemble de capteurs et boutons d'un chariot

(X) désigne le numéro du chariot.

Ex : pour le chariot 1 c'est on désigne : 1b0, 1b1, ...

Afin d'expliquer l'emplacement des capteurs et des boutons Sur chaque chariot, on va les présentés dans la figure suivante :

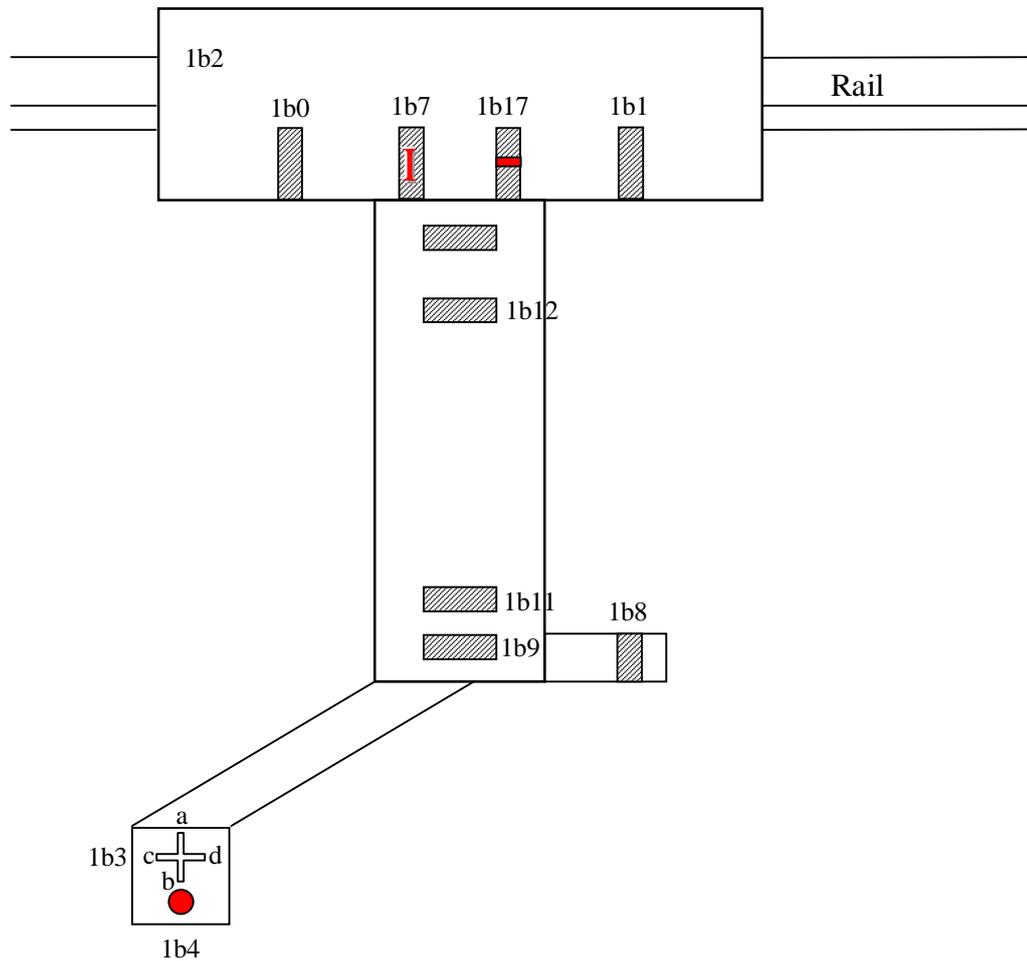


Figure 7: Vue de face d'un chariot

Chaque chariot est entraîné par deux moteurs de puissance de 20 watts (l'un pour le mouvement transversal et le deuxième pour le mouvement vertical), ces moteurs sont triphasés asynchrone qui peuvent tourner dans deux sens (à gauche et droite pour le moteur transversal, en haut et en bas pour le moteur vertical), et avoir deux vitesse (vitesse lente et vitesse rapide).

Chacun des deux moteurs est dotés aussi d'un frein mécanique alimenté par un courant continue venant d'un redresseur. Ils offrent ainsi au chariot un cycle de déplacement qu'on peu le résumer comme suit :

- Départ avec une vitesse lente (déplacement lent)
- Démarrage de la vitesse rapide (déplacement rapide)
- redémarrage de la vitesse lente et arrêt de la vitesse rapide (déplacement lent)
- Arrêt du chariot (stationnement)

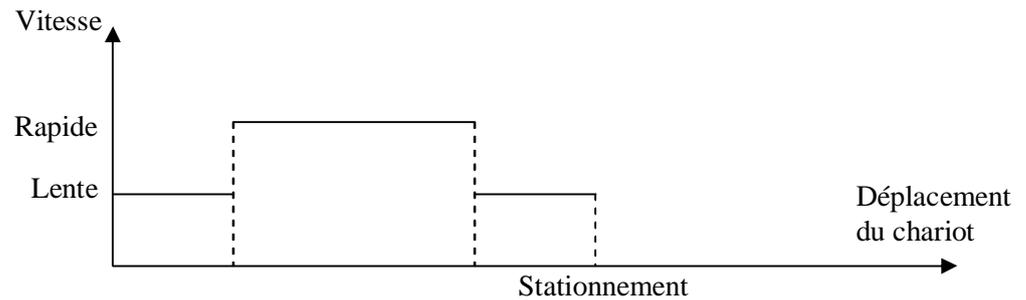


Figure 8: Cycle de déplacement du chariot

Schéma des moteurs :

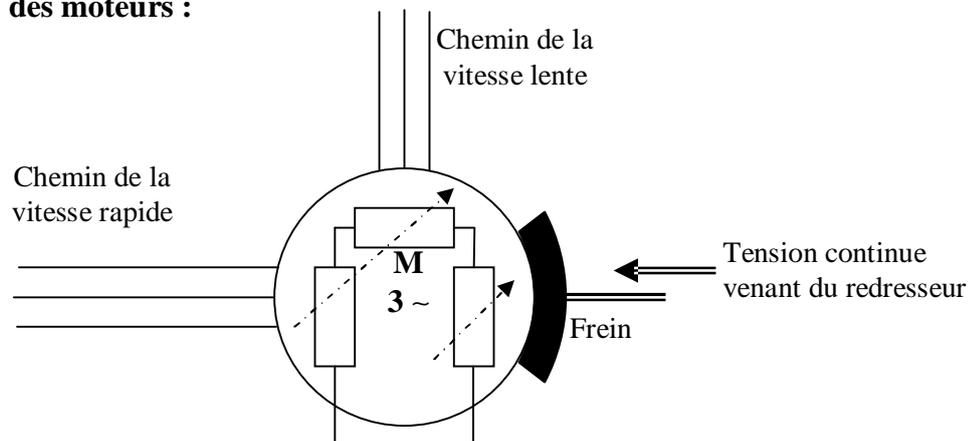


Figure 9: L'un des deux moteurs installés sur chaque chariot

4.2.3. Chariots transversaux :

4.2.3.1. Le bain 415 (bain de rinçage) :

Il se déplace entre les deux axes de la chaîne, permettant ainsi le déplacement des barres du premier axe au deuxième. On remarque aussi qu'il est doté de deux détecteurs de proximité (Un pour la détection de la barre et l'autre pour indiquer la fin de course du bain).

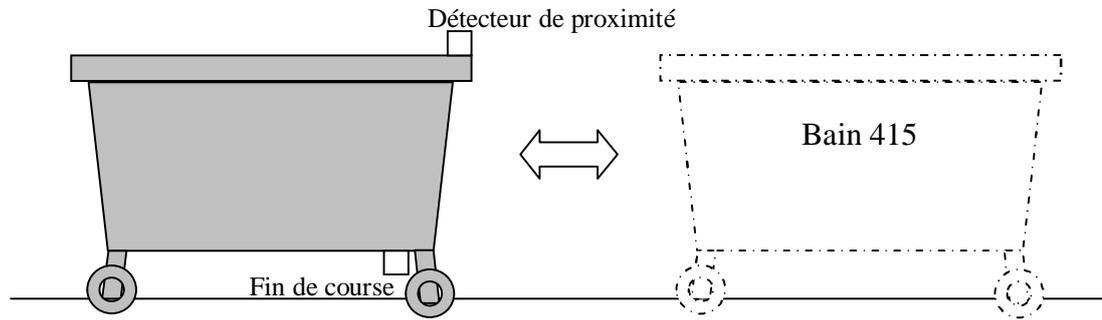


Figure 10: Le déplacement du bain 415

4.2.3.2. Chariots de chargement :

Il existe deux chariots de chargement, situés au début de la chaîne et qui travaillent en alternance.

4.2.4. Armoire de commande :

Elle contient :

La commande des six chariots :

A base de tapis, chaque tapi porte des papillons et roule avec un régime constant autour d'une succession de cames, où la configuration des papillons détermine une séquence d'étapes à suivre par le chariot à commander.

- Bouton de marche/arrêt
- Bouton de marche manuel/automatique de la chaîne
- Bouton de marche manuel/automatique de chaque chariot

Les relais :

Des relais de 24 volts assurant le lien entre la commande et la puissance

Les alimentations :

- De puissance (380 volts)
- De la commande et des relais (24 volts)

Les mesures de sécurité :

- Bouton d'arrêt de secours principal
- Bouton d'arrêt de secours de chaque chariot, ...

Des LEDs d'indication :

- Marche/arrêt de la chaîne
- Fonctionnement manuel/automatique de la chaîne
- Marche/arrêt des redresseurs, ...

CONCLUSION

Les traitements de surface sont utilisés pour modifier les caractéristiques de la surface d'une pièce dans le but de lui donner des qualités nouvelles : améliorer les propriétés mécaniques (dureté, frottement, résistance à l'usure, au grippage, à la fatigue...), la résistance à la corrosion, l'esthétique et certains comportements (conductivité électrique, réflexion de la lumière, conduction de la chaleur...).



CHAPITRE II

LES AUTOMATES PROGRAMMABLES INDUSTRIELS ET ENVIRONNEMENT DE PROGRAMMATION STEP7

INTRODUCTION

L'industrie moderne que, l'on peut qualifier d'industrie de qualité et de quantité, ne cesse d'exiger un matériel de contrôle de plus en plus performant afin de réaliser les deux objectifs, simultanément. Et c'est pour cette raison qu'on voulait remplacer les dispositifs de commande classiques avec tous les inconvénients qui en découlent (logique câblée très compliquée, encombrement, difficulté d'entretien ...) par un autre beaucoup plus performant et avantageux.

Ce serait certainement l'Automate Programmable Industriel qui devient de nos jours le cœur de toute unité industrielle moderne.

Voici donc une description plus ou moins détaillée de l'API et de tout ce qui y est lié en terme de soft et de hard.

1. HISTORIQUE

Au début des années 50, les ingénieurs étaient déjà confrontés à des problèmes d'automatismes, les composants de base de l'époque étaient les relais électromagnétique à un ou plusieurs contacts. Les circuits conçus comportaient des centaines voire des milliers de relais. Le transistor n'était connu que comme un composant d'avenir et les circuits intégrés étaient inconnus.

Vers 1960, les semi-conducteurs (transistors, diodes) sont apparus dans les automatismes sous forme de circuits digitaux. Ce n'est quelques années plus tard, que l'apparition des circuits intégrés a amorcé une révolution dans la façon de concevoir les automatismes. Ceux-ci étaient très peu encombrants et leur consommation était des plus réduite. On pouvait alors concevoir des fonctions de plus en plus complexes à des coûts toujours décroissants.

C'est en 1969 que les constructeurs américains d'automobiles (General Motors en particulier) ont demandé aux firmes fournissant le matériel d'automatisme des systèmes plus évolués et plus souples pouvant être modifiés simplement sans coût exorbitants[7].

Les ingénieurs américains ont résolu le problème en créant un nouveau type de produit nommé automates programmables. Ils n'étaient rentables que pour des installations d'une certaine complexité, mais la situation a très vite changée, ce qui a rendu les systèmes câblés obsolètes.

De nombreux modèles d'automates sont aujourd'hui disponible ; depuis les nano automates bien adaptés aux machines et aux installations simples avec un petit nombre d'entrées/sorties, jusqu'aux automates multifonctions capables de gérer plusieurs milliers d'entrées/sorties, et destinés au pilotage de processus complexes.

2. DEFINITION GENERALE :

Un Automate Programmable Industriel API dit Programmable Logic Controller PLC dans le langage industriel, est une machine électronique spécialisée dans la conduite et la surveillance en temps réel de processus industriels et tertiaires. Il exécute une suite d'instruction introduite dans ses mémoires sous forme de programme, et s'apparente par conséquent aux machines de traitement de l'information.

Trois caractéristiques fondamentales le distinguent des outils informatiques tels que les ordinateurs utilisés dans les entreprises et le tertiaire :

- § il peut être directement connecté aux capteurs et pré actionneurs grâce à ses Entrées/sorties industrielles.
- § il est conçu pour fonctionner dans des ambiances industrielles sévères, (Température, vibration, micro-coupures de la tension d'alimentation, Parasite, etc.).
- § enfin, sa programmation à partir de langages spécialement développés pour le Traitement de fonctions d'automatisme facilite son exploitation et sa mise en oeuvre¹

Selon la norme française EN 61131-1, un automate programmable est un :

« Système électronique fonctionnant de manière numérique, destiné à être utilisé dans un environnement industriel, qui utilise une mémoire programmable pour Le stockage interne des instructions orientées utilisateurs aux fins de mise en oeuvre des fonctions spécifiques, telles que des fonctions de logique, de mise en séquence, de temporisation, de comptage, et de calcul arithmétique, pour commander au moyen d'entrées et de sorties tout ou rien ou analogiques divers types de machines ou de processus. L'automate programmable et ses périphériques associés sont conçus pour pouvoir facilement s'intégrer à un système d'automatisme industriel et être facilement utilisés dans toutes leurs fonctions prévues. »²

3. ARCHITECTURE DES API:

3.1. 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 [8]:

- § Gestion des entrées/sorties.
- § surveillance et diagnostic de l'automate par une série de teste 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.³

¹ Automates Nano et Plate-forme d'automatisme micro (104) Schneider Electric 1999.

² M.Bertrand, Automates programmables industriels S 8015, Techniques de l'ingénieur.

³ Automate Nano et Plate- forme d'automatisme Micro (104) Schneider Electric 1999.

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 :

3.1.1. L'accumulateur

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

3.1.2. Le registre d'instruction :

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

3.1.3. 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.

3.1.4. Le registre d'état :

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

3.1.5. 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.

3.2. 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 :

3.2.1. 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.

3.2.2. 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, mementos, 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 (pile des données locales).

3.2.3. Mémoire de chargement :

La mémoire de chargement sert à l'enregistrement du programme utilisateurs sans affectation de mnémonique ni de commentaires (ces derniers restent dans la mémoire de la console de programmation).

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

3.2.4. Mémoire RAM non volatile :

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

3.2.5. Mémoire ROM :

Contient le système d'exploitation qui gère la CPU.

3.3. Les mémoires d'entrée/sortie :

Ils traduisent les signaux industriels en informations API réciproquement, appelées aussi coupleurs[9].

Beaucoup d'automates assurent cet interfaçage par des modules amovibles être modulaires par carte ou par rack. D'autres automates ont une structure monobloc, avec des modules intégrés dans un châssis de base, (cas des automates de télémécanique TSX17 et SIMATIC S7-314 IFM).

Le nombre total de modules est évidemment limité, pour des problèmes physiques :

- Alimentation électrique.
- Gestion informatique.
- Taille du châssis.

Différents types de modules sont disponibles sur le marché selon l'utilisation souhaitée, les plus répandus sont :

3.3.1. 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, 220/240V

3.3.2. 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 : $\pm 10V$, 0-10V, $\pm 20mA$, 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.

3.3.3. Module spécialisés :

Ils assurent non seulement une liaison avec le monde extérieur, mais aussi une partie du traitement pour soulager le processeur et donc améliorer les performances.

Ces modules peuvent posséder un processeur embarqué ou une électronique spécialisée.

On peut citer :

Les cartes de comptage rapide :

Elles permettent saisir des événements plus courts que la durée du cycle travaillant à des fréquences qui peuvent dépasser 10kHz.

Les Entrées/Sorties déportées :

Leur intérêt est de diminuer le câblage en réalisant la liaison avec détecteur, capteurs ou actionneurs au plus près de ceux-ci, ce qui a pour effet d'améliorer la précision de mesure.

La liaison entre le boîtier déporté et l'unité centrale s'effectue par le biais d'un réseau de terrain suivant des protocoles bien définis.

L'utilisation de la fibre optique permet de porter la distance à plusieurs kilomètres.

3.4. L'alimentation électrique :

Elle a pour rôle de fournir les tensions continues nécessaires aux composants avec de bonnes performances, notamment face aux micro-coupures du réseau électrique qui constitue la source d'énergie principale[10].

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.

3.5. Les liaisons :

Elles s'effectuent :

- § avec l'extérieur par des **borniers** (à vitesse, à clipser, ...), 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.

3.6. 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.

4. 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 côté 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 :

4.1. 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.

4.2. 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.

5. ENVIRONNEMENT

Dans le cadre d'une évolution conduisant à une automatisation de plus en plus globale, l'automate est de moins en moins acquis seul, et même si c'est le cas, il doit pouvoir se connecter à d'autres matériels à processeur, et dialoguer avec les agents d'exploitation[11].

Les types de communication supportés par les API modernes sont : la communication avec un opérateur par un pupitre ou un terminal industriel : ils permettent une communication homme-machine, et ce dans les deux sens (clavier, ...)

6. LE STEP 7

STEP7 fait partie de l'industrie logicielle SIMATIC. Il représente le logiciel de base pour la configuration et la programmation de systèmes d'automatisation.

Les tâches de bases qu'il offre à son utilisateur lors de la création d'une solution d'automatisation sont :

- La création et gestion de projets.
- La configuration et le paramétrage du matériel et de la communication.
- La gestion des mnémoniques.
- La création des programmes.
- Le chargement de programmes dans les systèmes cibles.
- Le test de l'installation d'automatisation.
- Le diagnostic lors des perturbations dans l'installation.

Il s'exécute sous les systèmes d'exploitation de Microsoft à partir de la version Windows 95. Et s'adapte par conséquent à l'organisation graphique orientée objet qu'offre ces systèmes d'exploitation[12].

7. APPLICATIONS DE STEP7:

Il est constitué de deux types d'application ; applications de base et applications en options.

7.1. Applications du logiciel de base STEP7 :

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

- Le gestionnaire de projets.
- La configuration du matériel.

- L'éditeur de mnémoniques.
- L'éditeur de programmes CONT, LOG, LIST.
- Le S7GRAPH pour la programmation séquentielle.
- La configuration de la communication NETPRO.
- Le diagnostic du matériel.

7.1.1. Gestionnaire de projets SIMATIC Manager :

Le gestionnaire de projets SIMATIC Manager gère toutes les données relatives à un projet d'automatisation, il démarre automatiquement les applications requises pour le traitement des données sélectionnées.

Le schéma suivant représente la fenêtre qui apparaît à l'ouverture du SIMATIC Manager

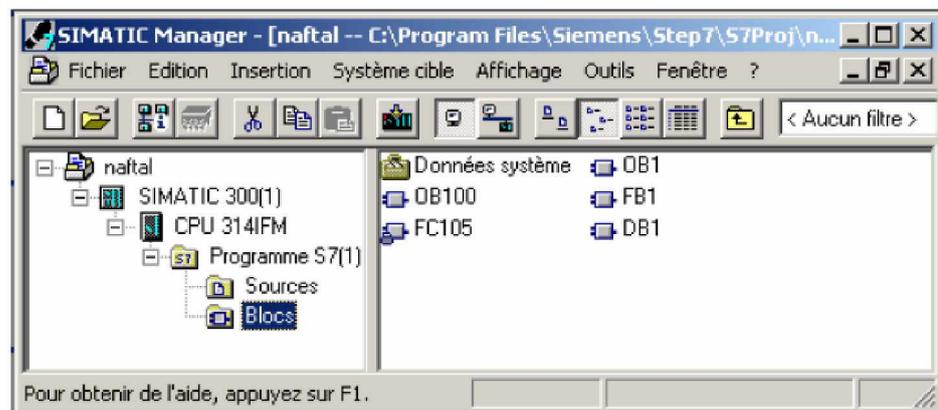


Figure 11: Fenêtre du SIMATIC manager.

7.1.2. Configuration du matériel HW Config :

HW Config est utilisé pour configurer et paramétrer le support matériel dans un projet d'automatisation.

7.1.3. Editeur de mnémoniques :

Il permet la gestion de toutes les variables globales. En effet, il définit des désignations symboliques et des commentaires pour les signaux du processus

(Entrées/Sorties), les mémentos, les blocs de données, les temporisations et les compteurs.

La table des mnémoniques qui en résulte est mise à disposition de toutes les applications. La modification de l'un des paramètres d'un mnémonique est de ce fait reconnue automatiquement par toutes les applications.

7.1.4. Editeur de programme :

Les langages de base proposés sont :

- Le schéma à contact (CONT), langage graphique similaire aux schémas de circuits à relais, il permet de suivre facilement le trajet du courant.
- La liste d'instruction (LIST), langage textuel de bas niveau, à une instruction par ligne, similaire au langage assembleur.
- Le logigramme (LOG), langage de programmation graphique qui utilise les boîtes de l'algèbre de BOOLE pour représenter les opérations logiques.

L'éditeur de programmes permet aussi la visualisation et forçage de variables,

7.1.5. Le S7Graph :

Le S7Graph est conçu pour la programmation graphique des processus, en plus des possibilités que nous offre le STEP7, il nous permet une programmation claire et rapide .

7.1.6. Configuration de la communication Net Pro :

La configuration et le paramétrage de réseaux se font à l'aide de l'application NetPro. Elle permet de :

- Créer une vue graphique du réseau en question ainsi que les sous-réseaux qui le constituent.
- Déterminer les propriétés et les paramètres de chaque sous-réseau.

7.1.7. 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.

Avec le diagnostic, on peut avoir des informations générales sur les modules, les causes des erreurs, comme on peut détecter les causes des défaillances dans un programme.

7.2. Possibilité d'extension du logiciel de base STEP7 :

Cette extension est réalisée à l'aide de logiciels optionnels, regroupés dans trois catégories[13]

7.2.1. Applications techniques :

Ensemble de langages de programmation évolués pour programmeur, des langages graphiques pour l'ingénieur en technologie et enfin des logiciels complémentaires pour le diagnostic, la simulation, la maintenance à distance, et la documentation de l'installation.

7.2.2. Logiciels exécutables :

Solutions finies programmées pouvant être appelées dans le programme utilisateur, et d'être directement intégrées dans la solution d'automatisation.

Exemple :

- Le contrôle PID qui permet l'intégration de régulateurs à action continue et de régulateurs à impulsion dans le programme utilisateur. L'application de paramétrage à laquelle la définition du régulateur est intégrée, permet le paramétrage rapide et le réglage optimal du régulateur.
- Le contrôle avec logique floue. Utilisé lorsque des processus ne peuvent pas ou peuvent difficilement être décrits mathématiquement, lorsque le déroulement de mécanismes et de processus est imprévisible, lorsque des comportements non linéaires surviennent alors que l'on dispose d'une connaissance acquise par expérience du processus.

7.2.3. Interfaces homme/machine :

Ce sont des logiciels spécifiques au contrôle-commande dans SIMATIC. On peut citer :

- Le système de visualisation du processus SIMATIC WinCC qui est un système de base indépendant des branches et technologies d'utilisation et qui comporte toutes les fonctions indispensables au contrôle-commande.
- ProAgent qui permet un diagnostic précis et rapide du processus dans les installations et les machines en fournissant des informations relatives à la localisation et à la cause des erreurs.

8. PREMIER PAS VERS STEP7 :

Pour concevoir un projet avec STEP7, il existe 2 approches[14] :

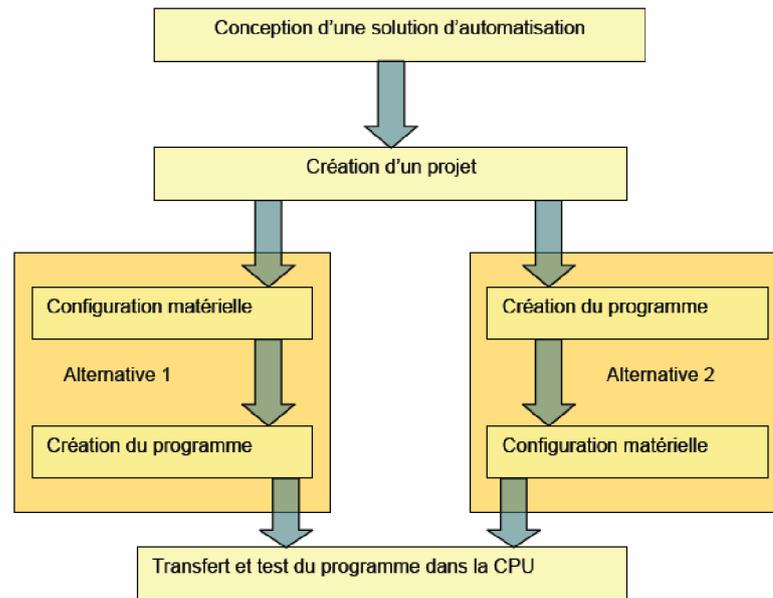


Figure 12: Organigramme pour la création de projets sous STEP 7.

Il faut noter que pour un système contenant beaucoup de variables, la seconde alternative n'est pas très pratique.

Nous optons donc pour la première approche.

8.1. Création du projet avec STEP7 :

Pour créer un projet avec STEP 7, on peut lancer l'assistant de création de projet de STEP7, ou créer directement un projet que l'on configurera soi-même.

8.1.1 Utilisation de l'assistant de création de projets :

Par défaut l'assistant de création de projets apparaît à chaque démarrage du SIMATIC Manager, si ce n'est pas le cas, son lancement se fait en passant par le menu Fichier>Assistante Nouveau projet'.

Cet assistant permet de créer avec une interface simple.

Les étapes à suivre sont les suivantes :

Etape 1 :

Cliquer sur le bouton « suivant »

Etape 2 :

§ Il faut choisir la CPU utilisée pour le projet, la liste contient normalement toutes les CPU supportées par la version de STEP7 utilisée.

§ Dans le champ « Nom de CPU », il faut donner un nom à la CPU. Cela peut

§ S'avérer utile dans le cas où l'on utilise plusieurs CPU dans un même projet.

§ Il faut aussi choisir une adresse MPI pour la CPU. Si on utilise une seule CPU

La valeur par défaut est 2.

§ Cliquer sur suivant.

Remarque :

Lors de la sélection de la CPU une brève description est disponible dans la petite fenêtre à coté du choix de l'adresse MPI.

Etape 3 :

Cet écran permet d'insérer des blocs. Ces blocs seront décrits plus loin.

§ Pour commencer on se contentera de OB1 seulement qui est le bloc principal (Équivalent du MAIN pour le langage C).

§ On doit aussi choisir un langage de programmation parmi les trois proposés (LISTE, CONT ou LOG).

§ Cliquer sur suivant.

Etape 4 :

Nommer le projet et cliquer sur créer.

Le projet est maintenant créé, on peut voir arborescence à gauche de la fenêtre Qui s'est ouverte.

8.1.2. Création d'un nouveau projet sans l'assistant de création de projet :

Cette méthode est un peu plus compliquée, mais permet de mieux gérer le 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.

La fenêtre du projet s'ouvre.

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 >station SIMATIC 300.

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

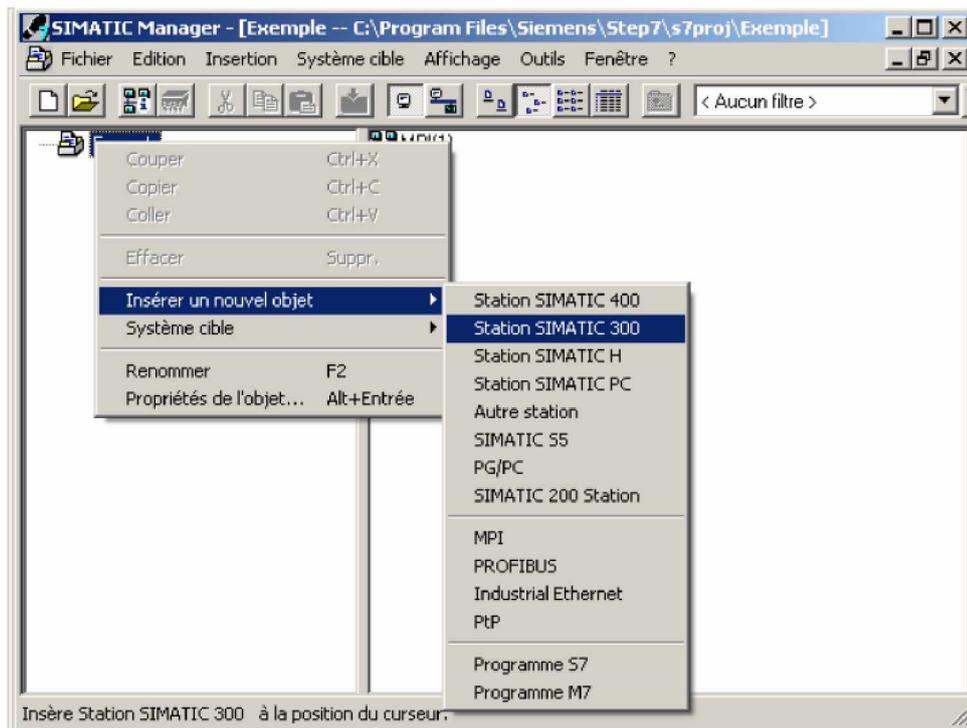


Figure 13: Création d'un projet sans l'assistant.

8.2. Configuration matérielle :

La configuration matérielle est une étape très importante, elle permet de reproduire à l'identique le système utilisé (alimentation, CPU, modules etc....)[15].

Pour effectuer cette configuration, il faut aller sur l'icône Station SIMATIC300. Sur la fenêtre de droite s'affichent deux icônes : « Matériel » et le nom de la CPU.

Il faut ouvrir l'icône matériel : la fenêtre HW Config s'ouvre.

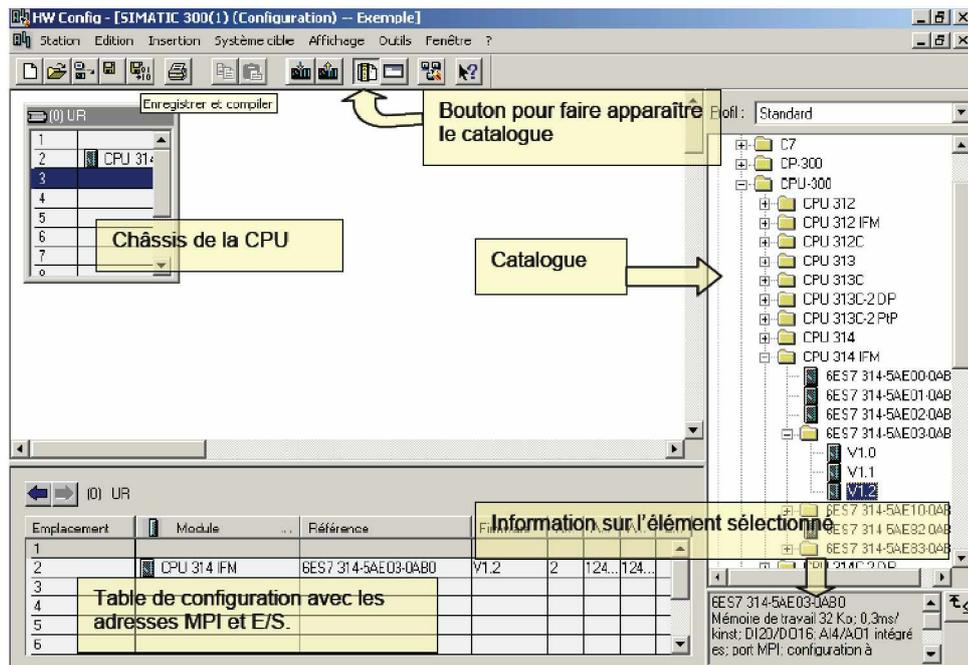


Figure 14: Configuration du matériel.

Nous avons tout d'abord besoin d'un châssis ou RACK puis d'une alimentation. Dans le catalogue ouvrir le dossier PS-300 qui se trouve dans SIMATIC-300 puis choisir un modèle PS 307 2A.

Pour l'installer sur le châssis, sélectionner l'emplacement 1 à l'aide de la souris, puis double-cliquer sur l'alimentation[16].

On peut de la même manière insérer des composants sur le châssis en fonction de la configuration réelle. Dans notre cas, l'insertion de la CPU S7 314 IFM de référence 6ES7314-5EA03-0AB0 se fera dans l'emplacement 2.

Le paramétrage de la CPU, à l'aide de menu, permet de définir des caractéristique, telles que : le comportement à la mise en route, la surveillance du temps de cycle ainsi que l'activation et la désactivation des plages de rémanences et les fonctions intégrées, ces données sont enregistrées dans les blocs de données systèmes.

Le paramétrage des modules est réalisé automatiquement au démarrage de la CPU. Ainsi, le remplacement d'un module est ainsi possible sans nouveau paramétrage.

A la fin de la configuration, il suffit de cliquer sur Station>Enregistrer et compiler pour valider les changements apportés au châssis. De cette manière les changements seront pris en compte dans le reste du projet.

8.3. Définition des mnémoniques :

Il faut définir les variables qui vont être utilisées lors des étapes de programmation. L'utilisation de noms communs est plus aisée que la manipulation de chiffres (ex : utiliser « moteur » au lieu du bit de sortie A0.0).

Pour accéder à la table des mnémoniques : cliquer 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 STEP 7.
- § Ecrire éventuellement un commentaire dans la colonne prévu à cet effet.

Après avoir défini toutes les mnémoniques, il suffit d'enregistrer pour que les changements soient pris en compte dans le reste du projet.

Etat	Mnémonique	Opérande	Type de don	Commentaire
1	A	E 124.1	BOOL	Détecteur A
2	B	E 124.2	BOOL	Détecteur B
3	C	E 124.3	BOOL	Détecteur C
4	COMPLETE RESTART	OB 100	OB 100	Complete Restart
5	D	E 124.4	BOOL	Détecteur D
6	FB1OK	A 125.5	BOOL	Sortie signalant le bon fonctionnement du bloc de fonct...
7	M	E 124.0	BOOL	Bouton de mise en marche du système
8	M1	A 125.0	BOOL	Sortie de commande du moteur M1
9	M2	A 125.1	BOOL	Sortie de commande du moteur M2
10	SENS	A 125.4	BOOL	Sortie commandant le sens de rotation des moteurs
11	TEMPO1	A 125.2	BOOL	Sortie indiquant l'état de la première temporisation
12	TEMPO2	A 125.3	BOOL	Sortie indiquant l'état de la seconde temporisation
13				

Figure 15: La fenêtre d'édition de mnémoniques.

Si on a besoin d'insérer de nouveaux objets dans le projet (ex : d'autres blocs de programme) il suffit de cliquer avec le bouton droit de la souris sur le dossier où l'on veut ajouter l'objet puis insérer un nouvel objet, et dans le menu sélectionner l'objet voulu.

9. ORGANISATION D'UN PROGRAMME UTILISATEUR :

Le logiciel de base STEP 7, permet de structurer le programme utilisateur.

Cette structuration est réalisée par la subdivision du programme en différentes parties autonomes. Il en résulte les avantages suivants [17] :

- § Ecrire des programmes importants et clairs.
- § Standardiser certaines parties du programme.
- § Simplifier l'organisation du programme.
- § Modifier facilement le programme.
- § Simplifier le test du programme, car on peut l'exécuter section par section.
- § Faciliter la mise en service.

9.1. Hiérarchisation dans un projet :

Chaque programme utilisateur est structuré sous forme de projet.

Un projet représente l'ensemble des données et programmes d'une solution d'automatisation et se trouve à la tête d'une hiérarchie d'objets qui sont :

- § objet projet.
- § objet station.
- § objet Modules programmables.
- § objet Programme S7/M7.
- § objet dossier Sources.
- § objet dossier Blocs.

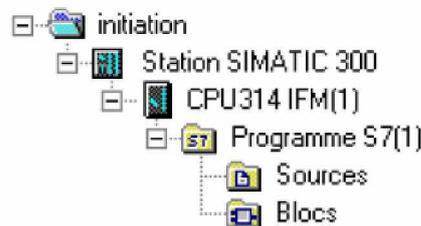


Figure 16: L'hiérarchie d'un projet STEP 7.

9.1.1. Objet station :

Une station SIMATIC 300/400 représente une configuration matérielle S7. Elle comporte un plusieurs modules programmable.

9.1.2. Objet modules programmables :

Représente les données de paramétrage d'un module comme les CPU et les modules fonctionnels.

9.1.3. Objet programme S7/M7 :

Un programme (S7/M7) est un dossier contenant les logiciels pour les modules programmable. Dans un programme S7/M7 figure déjà :

Une table de mnémoniques, un dossier « Blocs » et un dossier « Sources ».

9.1.4. Objet dossier sources :

Il contient les programmes source sous forme de texte. On peut saisir une partie ou tout le programme sous forme de source que l'on compilera ensuite en blocs.

Les avantages de la création de programme sous forme de source sont :

- § La possibilité de programmer plusieurs blocs dans une même source.

§ L'avantage d'enregistrer la source malgré la présence éventuelle d'erreurs de syntaxe, ce qui n'est pas possible lors de la création de blocs de code avec vérification de syntaxe. Cela signifie que les erreurs de syntaxe ne seront signalées que lors de la compilation de la source[18].

Elle peut être créée et traitée avec le choix de l'éditeur ASCII, puis importée et compilée en blocs individuels. La compilation entraîne la génération des différents blocs et leur sauvegarde dans le programme utilisateur S7.

La source doit être écrite en utilisant la syntaxe du langage de programmation « liste d'instructions (LIST) », et nécessite par conséquent l'application de règles précises.

9.1.5. Objet dossier Blocs :

Le dossier Blocs contient les blocs que l'on doit charger dans la CPU pour réaliser la tâche d'automatisation.

Il englobe les blocs de code (OB, FB, SFB, FC, SFC) qui contiennent les programmes qu'on doit charger dans la CPU, et les blocs de données (DB d'instance et DB globaux) qui contiennent les paramètres du programme.

Les blocs d'organisation (OB) :

Ils constituent l'interface entre le système d'exploitation et le programme utilisateur. Ils sont appelés par le système d'exploitation et gèrent le traitement de programme cyclique et déclenché par alarme, ainsi que le comportement à la mise en route de l'automate programmable et le traitement des erreurs. On peut programmer les blocs d'organisation et déterminer ainsi le comportement de la CPU.

Le bloc d'organisation **OB1** est généré automatiquement lors de la création d'un projet, il représente le programme principal (MAIN dans le langage C). En effet, c'est le programme appelé cycliquement par le système d'exploitation. Les autres blocs, existant dans le projet seront exécutés à leur appel par l'OB1.

Les blocs fonctionnels (FB) :

§ Le FB :

C'est un sous programme écrit par l'utilisateur, il facilite la programmation de fonctions complexes souvent utilisées. Il exécute par l'appel d'autre bloc de code.

Un bloc de données d'instance, qui constitue sa mémoire, lui est associé. Ce dernier contient les paramètres transmis au FB ainsi que les variables statiques.

§ Le bloc fonctionnel système (SFB) :

C'est un bloc fonctionnel intégré à la CPU S7. Les SFB font partie du système d'exploitation, par conséquent, ils ne sont pas chargés en tant que partie du programme. Comme les FB, les SFB sont des blocs avec mémoire. On doit donc également créer pour les SFB des blocs de données d'instance que l'on charge dans la CPU en tant que partie du programme.

Ils sont utilisés pour des fonctions spéciales intégrées de la CPU 314 IFM, comme ils peuvent être utilisés pour la communication via des liaisons configurées.

Les fonctions (FC) :**§ La Fac :**

Elle contient des routines pour les fonctions fréquemment utilisées, comme le renvoi d'une valeur au bloc appelant. Elle est sans mémoire et contient uniquement des variables temporaires qui sont sauvegardées dans la pile de données locales et perdues à l'achèvement de cette fonction.

Mais elle peut faire appel à des blocs de données globaux pour la sauvegarde de ses données.

§ La fonction système (SFC) :

C'est une fonction intégrée dans la CPU S7, pré-programmée et testée. Elle est appelée à partir du programme. Comme ces fonctions font partie du système d'exploitation, elles ne sont pas chargées en tant partie du programme. Comme les FC, les SFC constituent des blocs sans mémoire.

Parmi les fonctionnalités qu'elles proposent :

Le contrôle du programme, la gestion des alarmes horaires et temporisées, la mise à jour de la mémoire image du processus, l'adressage de modules et la création de messages relatifs aux blocs.

Les blocs de données d'instance (DB d'instance) :

Associé à chaque bloc fonctionnel, il contient les paramètres effectifs et les données statiques du FB.

On peut utiliser plusieurs DB pour un même FB ; par exemple, un FB pour la commande de plusieurs moteurs, les données de chaque moteur sont sauvegardées dans différents DB.

Les blocs de données globaux (DB) :

A l'opposé des DB d'instance qui ne sont associés qu'au blocs fonctionnels, les DB globaux servent à l'enregistrement de données utilisateur pouvant être utilisées par tous les autres blocs de code.

Remarque :

A l'appel d'un FB, le DB d'instance lui correspondant est automatiquement généré une fois qu'il est inséré à son emplacement.

Mais si on veut créer un bloc de données à partir du dossier bloc, on procède comme suit :

§ On clique avec le bouton droit de la souris, puis sur Insérer un nouvel objet>Bloc de données,

§ Ou Dans la fenêtre SIMATIC Manager, on clique sur le menu Insertion> BlocS7>bloc de données.

Dans les deux cas, le type du bloc de données sera demandé (DB d'instance ou DB global).

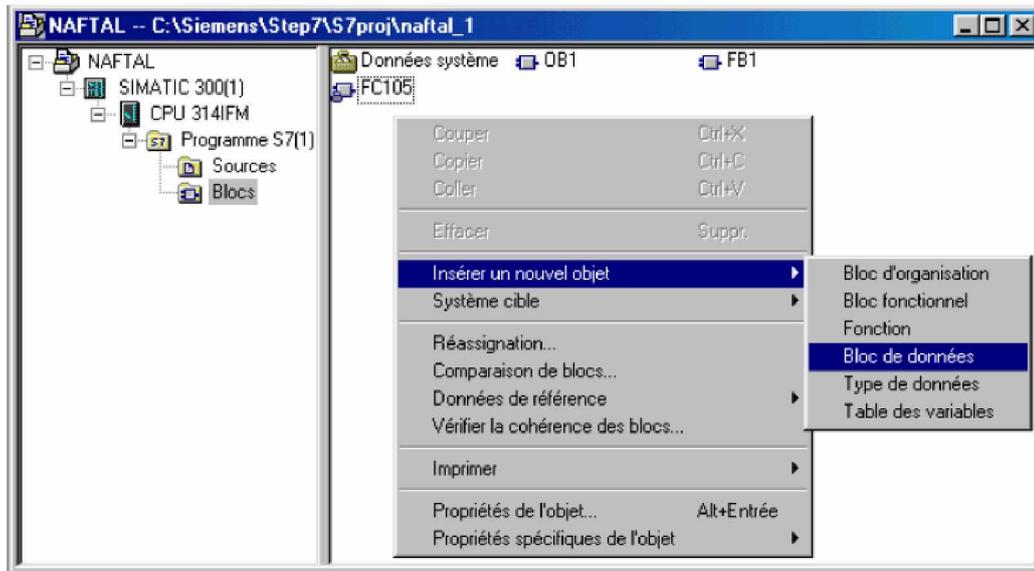


Figure 17: Création de blocs de données

10. BLOCS D'ORGANISATION :

Ils permettent l'organisation totale du programme utilisateur, en déclenchant l'exécution conditionnelle de certaines parties du programme, ceci se fait [19] :

- § A la mise en route de la CPU.
- § A des horaires précis ou cycliquement.
- § A l'apparition d'erreurs.
- § A l'apparition d'alarmes de processus.

10.1. Blocs d'organisation dans la CPU S7-314 IFM :

Les blocs d'organisation qui existe dans la CPU S7-314 IFM sont :

10.1.1. L'OB1 (programme cyclique) :

Comme dans toutes les CPU S7, l'OB1 est exécuté de manière cyclique.

Il peut être interrompu par tous les autres OB car il dispose de la classe de priorité la plus basse. Les événements suivants provoquent son appel par le système d'exploitation :

- § La fin du traitement de la mise en route.
- § La fin du traitement de son cycle précédent.

10.1.2. L'OB100 (Mise en route) :

Il contient le programme à exécuter lors d'une mise en route.

10.1.3. Les OB d'alarmes :

On appelle alarmes les événements qui déclenchent l'appel d'un OB donné.

§ **OB10 (Alarme horaire) :**

A l'appel de cet OB, le programme qui s'y trouve ne s'exécute qu'une seule fois ou de manière cyclique toutes les minutes, toutes les heures, tous les jours, toutes les semaines, tous les mois, tous les ans, ou à la fin du mois.

Pour appeler l'OB10, il faut qu'il soit activé et paramétré, ceci se fait en utilisant :

- STEP : Dans HW Config, activer et paramétré l'alarme horaire en accédant Accédant aux propriétés de la CPU.
- Le programme utilisateur : activer l'alarme en appelant le SFC30 et la Paramétrer avec la SFC28.

§ **OB20 (Alarme temporisée) :**

Cet OB est appelé après l'écoulement d'un retard paramétré. Pour ce faire, il faut procéder comme suit :

- Appeler la fonction système SFC32,
- Charger l'OB d'alarme temporisée dans la CPU comme partie du programme.

§ **OB35 (Alarme cyclique) :**

L'OB est appelé à des intervalles de temps égaux qui peuvent être de 1 ms à 60000 ms. La valeur de cycle par défaut est de 100 ms, mais on peut la changer en accédant aux propriétés de la CPU dans HW config.

§ **OB40 (Alarme processus) :**

Quand un module déclenche une alarme de processus, le système d'exploitation identifie de quel module il s'agit, et déclenche l'OB40, qui une fois exécuté, permet l'acquittement de cette voix particulière.

Si une autre alarme de processus sur le même module est détectée entre l'identification et l'acquittement de l'alarme en cours, elle n'est pas prise en considération.

Dans le cas de la CPU 314 IFM, les alarmes de processus peuvent être déclenchées par les fonctions intégrées, si elles sont validées lors de la configuration de ces fonctions.

10.1.4. Les OB d'erreurs :

§ **OB82 (diagnostic des modules) :**

Le diagnostic signale les erreurs survenant sur les modules et permet alors l'exécution de l'OB82. Cet OB contient, dans ses variables locales, l'adresse de base logique du module erroné ainsi que des informations de diagnostic de quatre octets de long.

Si l'OB82 n'a pas été programmé, la CPU passe à l'état arrêt à la survenue d'une erreur.

§ **OB80 (Erreur asynchrone) :**

Il est appelé si l'une des erreurs suivantes se produit lors de l'exécution d'un OB :

- Dépassement du temps de cycle.
- Erreur d'acquittement lors de l'exécution d'un OB.
- Saut de l'heure de déclenchement d'un OB (horloge avancée).

Si l'OB80 n'a pas été programmé, la CPU passe à l'état arrêt à la survenue d'une erreur.

§ **OB81 (Erreur d'alimentation) :**

Cet OB est exécuté si un événement provoque une erreur d'alimentation ; par exemple, absence ou rétablissement de la tension de sauvegarde dans la CPU.

Si l'OB81 n'a pas été programmé, la CPU ne passe pas à l'état arrêt à la survenue d'une erreur.

§ **OB85 (Erreur d'exécution de programme) :**

Cet OB est exécuté si l'un des événements suivants se produit :

- Absence d'OB dans la CPU lors de son appel.
- Erreur lors de l'accès du système d'exploitation à un bloc.
- Erreur d'accès à la périphérie lors de l'actualisation de la mémoire image Des entrées.
- Erreur d'accès à la périphérie lors du transfert de la mémoire image aux Modules de sorties.

Si l'OB n'a pas été programmé, la CPU passe à l'état arrêt à la survenue d'une erreur.

§ **OB87 (Erreur de communication) :**

Le programme de cet OB est exécuté, si une erreur de communication se produit. Par exemple, l'état d'ensemble des données globales ne peut pas être écrit dans le bloc de données.

Si l'OB87 n'a pas été programmé, la CPU passe à l'état arrêt si une erreur survient.

§ **OB121 (Erreur de programmation) :**

L'apparition d'erreur lors du traitement du programme utilisateur, déclenche l'appel de ce bloc. Cette erreur peut être :

- Une erreur de conversion BCD.
- Un numéro de temporisation erroné.
- Une erreur de numéro de compteur.
- Une erreur d'écriture à l'accès au DB.
- Une erreur de numéro de bloc à l'ouverture d'un DB.
- Une erreur de numéro de bloc à l'appel d'une FC.
- Une erreur de bloc à l'appel d'un FB.
- Un DB non chargé.
- Un FC non chargé.
- Un FB non chargé.

Si l'OB121 n'a pas été programmé, la CPU passe à l'état arrêt à la survenue d'une erreur.

§ OB122 (Erreur d'accès à la périphérie) :

Cet OB se déclenche à l'apparition d'erreur d'accès à la périphérie en lecture ou en écriture.

Si l'OB122 n'a pas été programmé, la CPU passe à l'état arrêt à la survenue d'une erreur.

Remarque

Les erreurs résolues par l'appel des deux blocs OB121 et OB122 peuvent être masquées, démasquées ou lues, par les fonctions systèmes suivantes :

- La SFC36 : masque certains codes d'erreurs.
- La SFC37 : démasque les codes d'erreurs qui ont été masquées par la SFC36.
- La SFC38 : lit le registre d'erreur.

10.2. Classes de priorité des blocs d'organisation :

Les blocs d'organisation définissent l'ordre dans lequel les différentes parties du programme sont traitées. L'exécution d'un OB peut être interrompue par l'appel d'un autre OB.

Cette interruption se fait selon la priorité : les OB de priorité plus élevée interrompent les OB de priorité plus faible[20].

OB	Classes de priorité
OB1	1
OB10	2
OB20	3
OB35	12
OB40	16
OB80	26
OB81	26
OB82	26
OB85	26
OB100	27
OB121	Selon l'OB responsable de l'interruption
OB122	

Tableau 6: Classes de priorité des blocs d'organisation

Remarque :

STEP7 permet la modification des classes de priorité pour les alarmes horaires, alarmes temporisées, alarmes cycliques de processus. Mais cette propriété n'est pas possible dans les CPU S7 300.

11. Déroulement d'un programme :**11.1. Les programmes existants dans la CPU :**

Dans une CPU, s'exécutent deux programmes ; le système d'exploitation et le programme utilisateur[21].

11.1.1 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.

11.1.2. 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.

11.2. 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.

La seule mise en route disponible dans la CPU 314 IFM est le démarrage à chaud.

11.2.1. Démarrage à chaud :

Quand la CPU fonctionne sans pile de sauvegarde, un effacement général est automatiquement effectué à la mise sous tension, puis un démarrage à chaud est exécuté. Le programme utilisateur doit se charger de nouveau[22].

Au démarrage à chaud, les démarches faites par la CPU dans l'ordre sont :

- § Effacer la pile des interruptions et la pile des blocs.
- § Effacer les mémentos, temporisation et compteurs non rémanents,
- § Effacer la mémoire image des sorties,
- § Effacer les sorties des modules de signaux,
- § Rejeter les alarmes de processus,
- § Rejeter les alarmes temporisées,
- § Rejeter les alarmes de diagnostic,
- § Actualiser la liste d'état système (SZL),
- § Exploiter les paramètres des modules et les transmettre aux modules ou bien leur transmettre les valeurs par défaut,
- § Traiter l'OB de mise en route concerné,
- § Actualiser la mémoire image des entrées,
- § Valider les sorties TOR (débloquent les sorties TOR) après passage à l'état de fonctionnement « marche ».

11.2.2. Démarrage à froid et redémarrage :

Au démarrage à chaud, l'effacement des mémentos, des temporisations et des compteurs concernent uniquement les zones non rémanentes, alors que le démarrage à froid les efface tous.

Le redémarrage n'efface aucune zone de mémoire ; la pile des interruptions, la pile des blocs, les mémentos, les temporisations, les compteurs.

L'effacement de la mémoire image des sorties et les sorties des modules de signaux est paramétrable. La propriété qui distingue le redémarrage des deux autres types de mise en route est le traitement du cycle restant (partie du programme utilisateur n'ayant pas pu être exécutée en raison d'une mise hors tension).

11.3. Exécution cyclique :

A la fin de la mise en route, le programme utilisateur s'exécutera de manière cyclique dans l'OB comme suit :

- § Ecriture de la mémoire image des sorties MIS dans les modules de sorties.
- § Lecture des entrées et mise à jour de la mémoire image des entrées.
- § Traitement du programme utilisateur.

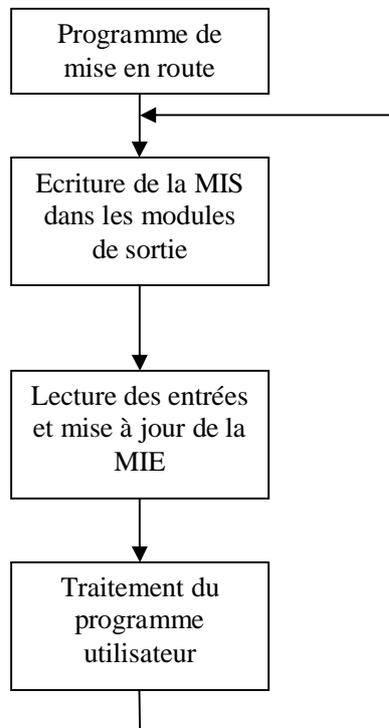


Figure 18: Déroulement d'un programme utilisateur.

12. LES LANGAGES DE PROGRAMME :

12.1. Le cont :

C'est un langage dont la logique est inspirée des réseaux électriques, ce qui en fait un langage facile pour les habitués des montages à base de relais.

Le langage à contact, est aussi appelé CONT ou LADDER. C'est un langage entièrement graphique, bien adapté au traitement logique simple de type combinatoire. Il utilise les symboles graphiques comme des contacts à ouverture ou à fermeture et des bobines. Ainsi un programme écrit en langage à contacts ne se présente pas sous la forme d'une liste d'instructions, mais comme une liste de schémas développés classique.

Des blocs d'opérations logiques ou arithmétiques pré-programmés peuvent être insérés dans les réseaux de contacts.

Les opérations combinatoires sur bits utilise deux chiffres : 1 et 0. Pour les contacts et les bobines, 1 signifie activé ou excité et 0 signifie désactivé ou désexcité.

Les opérations de combinaison sur bits évaluent les états de signal 1 et 0. et les combinent selon la logique booléenne. Le résultat de ces combinaisons est égal à 1 ou 0. Il s'agit du résultat logique (RLG).

Les éléments principaux d'un réseau en langage à contacts sont :

- ┆ Barre d'alimentation (à gauche).
- ┆ Barre représentant la masse (à droite).
- ┆┆ Contact associé à une variable.
- () Bobine de sortie.

12.1.1. Barre d'alimentations et liaisons :

Un diagramme CONT est limité à sa gauche par des barres d'alimentation et à sa droite par une barre représentant la masse.

Les différents composants sont reliés entre eux par des arcs de liaisons horizontaux ou verticaux. Chaque segment de liaison peut prendre les états TRUE ou FALSE (1 ou 0).

L'état d'une liaison est propagé à sa droite.

Toute liaison horizontale connectée à une barre d'alimentation à gauche est active.

On peut avoir des liaisons multiples en combinant plusieurs arcs de liaisons horizontaux à un arc de liaison vertical.

Avec ces quelques règles on peut déjà écrire un réseau basic :

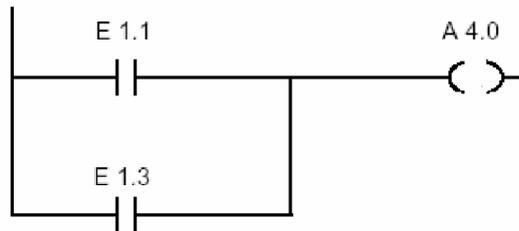


Figure 19: « OU » logique avec CONT.

A4.0 n'est mise à 1 que si : E1.1 **ou** E1.3 sont à 1.

12.1.2. Les contacts :

Les principaux contacts qui peuvent être utilisés dans un diagramme sont :

- § contact à fermeture 
- § contact à ouverture 

Remarques :

- § Le nom de la variable associée est inscrit au-dessus du symbole graphique.
- § Les contacts ne représentent pas ce qui est relié physiquement à l'automate, mais sont une interrogation à 1 ou à 0 de l'opérande associé.
- § Un contact représente une liaison entre l'état d'un arc et la variable booléenne associée à ce contact.

En effet l'état de la liaison à droite du contact n'est que le résultat du ET logique entre l'état de l'arc de gauche et :

- L'état de la variable associée au contact pour un contact à fermeture.
- L'inverse de l'état de la variable associée au contact pour un contact à ouverture.

12.1.3. Les bobines :

Les bobines standard sont une affectation du résultat logique à la variable associée.

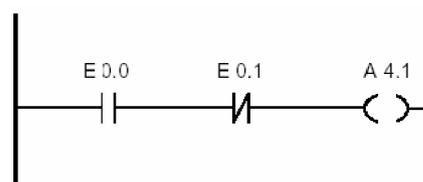


Figure 20: ET logique avec CONT.

A4.1 n'est mise à 1 que si : E0.0 **et non** (E0.1) sont à 1 simultanément.

Remarque :

Les opérations CONT possibles avec STEP 7 sont citées en annexe.

12.2. Le LOG :

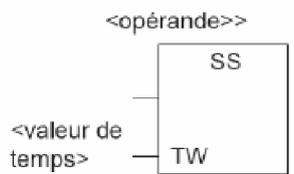
Le langage LOG est un langage graphique qui utilise des boites logiques d'algèbre de BOOL. La base de ce langage est la logique binaire, mais on peut aussi faire des opérations plus complexes telles que les opérations mathématiques à l'aide de blocs.

Les instructions LOG peuvent être sous quatre formes :

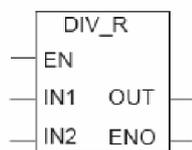
§ sous forme d'éléments : comme une inversion logique. 

§ sous forme de boite avec opérande : comme une affectation. 

§ sous forme de boite avec opérande et valeur : comme une temporisation.



§ sous forme de boite avec paramètre : comme diviser un nombre réel.



Remarque :

§ Pour l'activation d'une boite, il faut que EN soit à l'état haut.

§ Si la boite s'exécute sans erreur alors ENO est mise à 1, dans le cas contraire ENO est à 0.

Adressage :

Il n'y a que deux types d'adressage possibles avec LOG :

§ Adressage immédiat : donne une constante comme opérande.

§ Adressage direct : donne une adresse comme opérande.

Exemple :

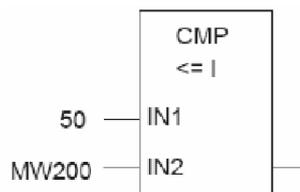


Figure 21: Adressage immédiat et direct avec LOG.

50 est la valeur effective du premier paramètre, c'est un adressage immédiat.
MW200 est la zone mémoire où est stockée la valeur du second paramètre : c'est un adressage direct.

12.3. Le LIST :

12.3.1. Présentation :

Le langage LIST figure parmi les langages de base du logiciel STEP7, sa syntaxe est similaire à celle de l'assembleur.

C'est le langage le plus proche du langage machine MC7, des CPU S7, ce qui lui donne l'avantage d'être le langage le plus adapté pour la programmation avec optimisation d'espace mémoire et de temps d'exécution.

Il dispose d'un jeu d'instruction très important permettant la création de programmes utilisateur complets.

Tout programme écrit en CONT ou en LOG peut être réécrit en LIST.

12.3.2. Structure et éléments de LIST :

Dans le langage LIST, l'éditeur de programme divise l'espace de programmation en deux parties ; partie programme et partie commentaire, ces deux parties sont séparées par //.

12.3.3. Structure d'une instruction :

Une instruction LIST est constituée soit :

- § En une opération seulement,
- § En une opération + opérande.

L'opérande peut indiquer :

- § une constante.
- § un bit du mot d'état.
- § une mnémonique.
- § un bloc de données.
- § un identificateur d'opérande d'une zone mémoire précise + une adresse.

Exemple :

Type d'opérande		Exemple	Explication
Constante	Entier	L 25	Charger l'entier 25
	Temporisation	L S5T#12s	Charger 12 secondes comme valeur initiale de la temporisation dans l'accumulateur 1
Adresse dans le mot d'état	Bit dans le mot d'état	U BIE	Interroger l'état du bit RB du mot d'état
Mnémonique	Nom symbolique	L vitesse	Charger l'octet, le mot ou le double mot dont le mnémonique est vitesse
Bloc de Données	Donnée du bloc De données.	U DB4.DBX5.1	Interroger l'état du bit 5.1 du bloc de donnée 4

Tableau 7: Quelques types d'opérandes.

13. AVANTAGES DE L'UTILISATION DU S7-Graph

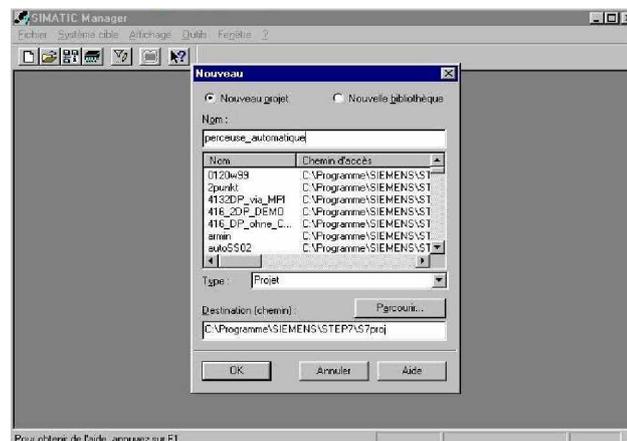
- § Avec le langage de programmation S7-GRAPH, l'étendue des fonctions de STEP7 va être élargie à une programmation graphique pour l'enchaînement des commandes[5].
- § Avec S7-GRAPH, vous pouvez programmer l'enchaînement des commandes clairement et rapidement.
- § Le processus sera pour cela décomposé en unités et représenté graphiquement.
- § Dans les unités, les actions à accomplir seront définies.
- § Les conditions de la prochaine étape (réceptivités) peuvent être programmées en CONT ou en LOG.
- § Le langage de programmation S7-GRAPH correspond à la norme DIN EN 61131-3 (IEC 61131-3) définissant le langage SFC « Sequential Function Chart ».

14. CREATION D'UN PROGRAMME S7-Graph

A partir du Graph-7, on peut maintenant créer un programme S7-GRAPH exécutable.

14.1 Démarrer SIMATIC- Manager et créer un nouveau projet

1. Cliquer sur l'icône **Nouveau**.
2. Saisir le nom du projet.
3. Cliquer sur **OK**.

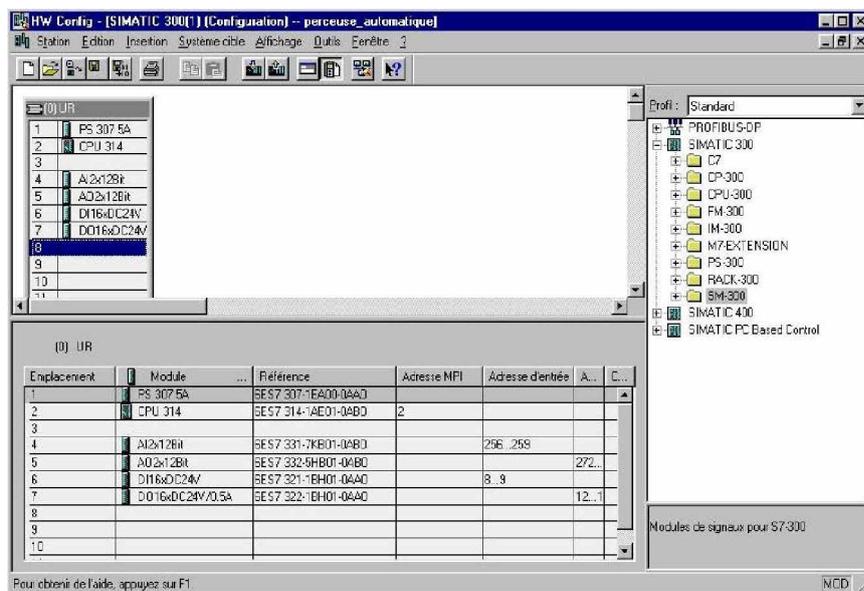


14.2 Insérer la station SIMATIC 300 et ouvrir la configuration matérielle

1. Sélectionner le projet **perceuse_automatique**.
2. Cliquer sur **Insertion**.
3. Choisir **Station**.
4. Cliquer sur **Station SIMATIC 300**.
5. Choisir **SIMATIC 300(1)**.
6. Double-cliquer sur **Matériel**.

14.3 Configurer le matériel et transmettre à l'automate

1. Entrer les composants de l'installation.
2. **Enregistrer et Compiler**.
3. **Charger dans le module** le matériel.
4. Fermer la fenêtre.



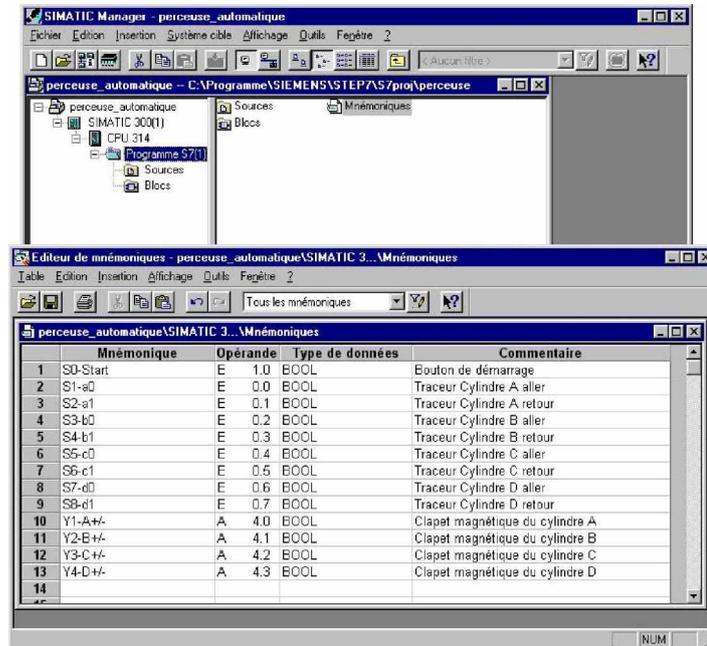
Remarque

La configuration présentée ici est un exemple.
A chaque automate correspond une configuration matérielle propre.

14.4 Créer la table des mnémoniques

1. Développer l'arborescence et cliquer sur **Programme S7 (1)**.
2. Double-cliquer sur **Mnémoniques**.
3. Saisir la table des mnémoniques.

4. Enregistrer la table des mnémoniques.
5. Fermer la fenêtre.

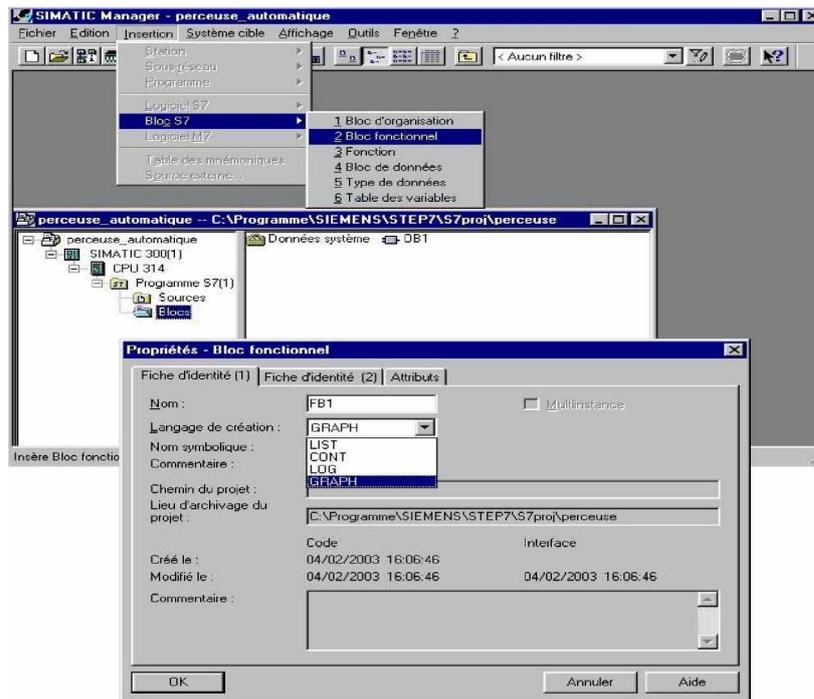


Remarque

L'adressage des opérands dépend de chaque automate.

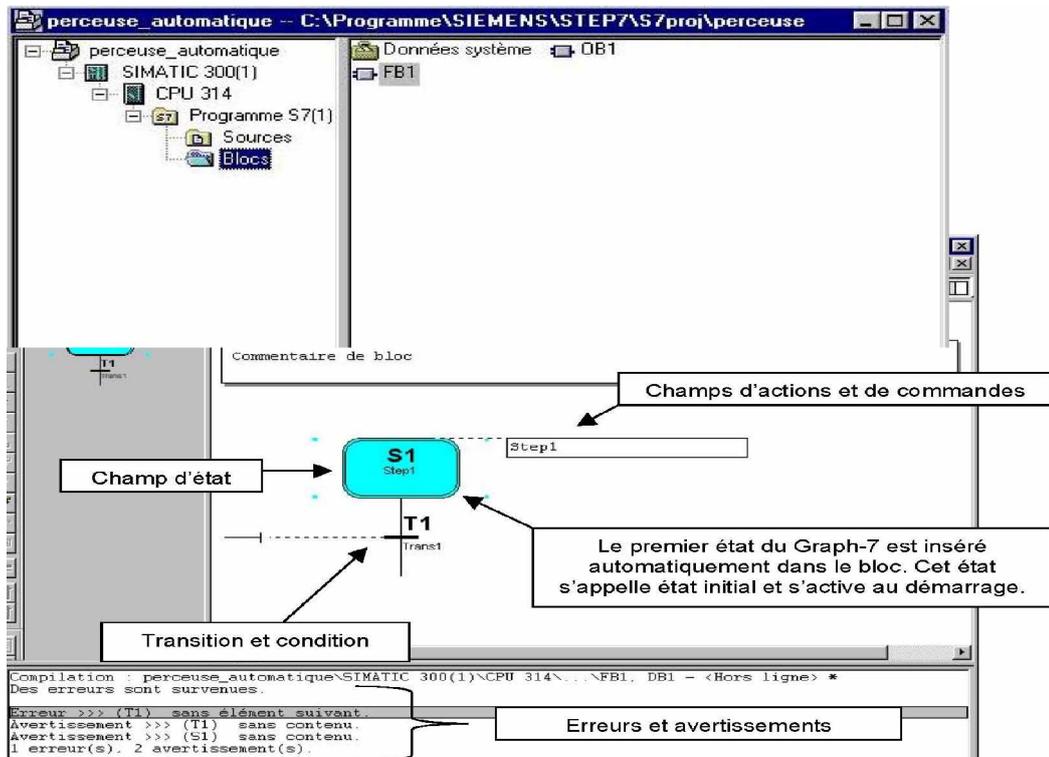
14.5 Insérer le bloc fonctionnel du S7-GRAPH

1. Ouvrir l'arborescence et cliquer sur bloc.
2. Cliquer sur **Insertion**.
3. Choisir **Bloc S7**.
4. Cliquer sur **Bloc fonctionnel**.
5. Choisir **GRAPH** en tant que langage de création.
6. Cliquer sur **OK**.



14.6 Ouvrir le S7-GRAPH et saisir l'enchaînement

1. Cliquer sur **Bloc**.
2. Double cliquer sur **FB1**. Le programme S7 GRAPH s'ouvre.



14.7 Le principe du langage Graph-7

Le langage Graph-7 est constitué par une suite d'états qui sont activés selon des conditions de transitions dans un ordre défini.

Le démarrage du Graph-7 est toujours constitué d'un ou plusieurs états initiaux qui se trouvent à des emplacements quelconques. Tant que l'action d'un état est en cours, cet état est actif. Quand plusieurs états s'exécutent simultanément, toutes ces étapes sont actives[19].

Un état est quitté, si tous les évènements sont exécutés, et si la condition de transition pour passer à l'étape suivante est remplie.

Le prochain état, qui suit cette transition, devient actif.

A la fin du Graph-7, il y a un saut vers une étape quelconque de ce Graph-7 ou vers un autre Graph-7 FB. C'est pour cette raison que le déroulement du Graph-7 peut être cyclique. A la fin du Graph-7 on peut aussi mettre une étape finale ou puits. Le déroulement se termine alors par cet état final.

14.8 Etape active

Une étape active est une étape, dont les actions sont en cours d'exécution.

Une étape est active

- si la condition de la transition en amont est remplie ou
- s'il s'agit de l'étape initiale au démarrage du Graph-7 ou
- si elle est appelée par une autre action active.

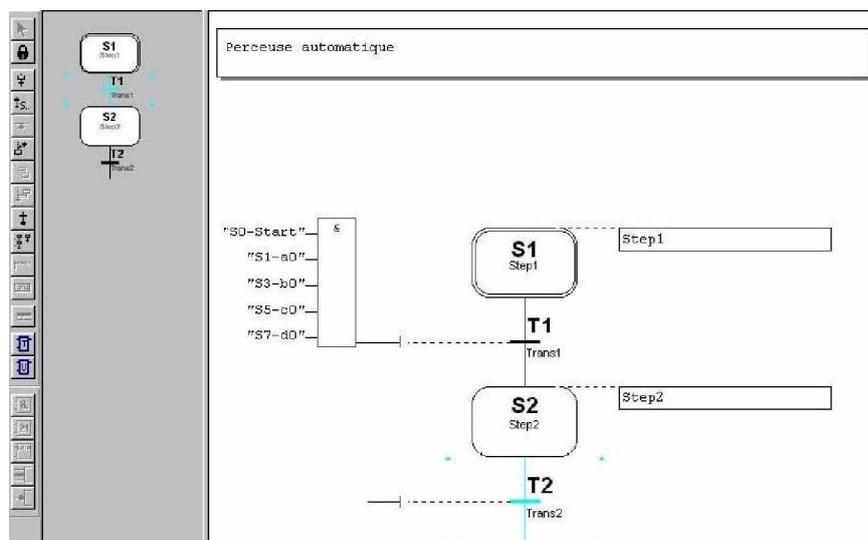
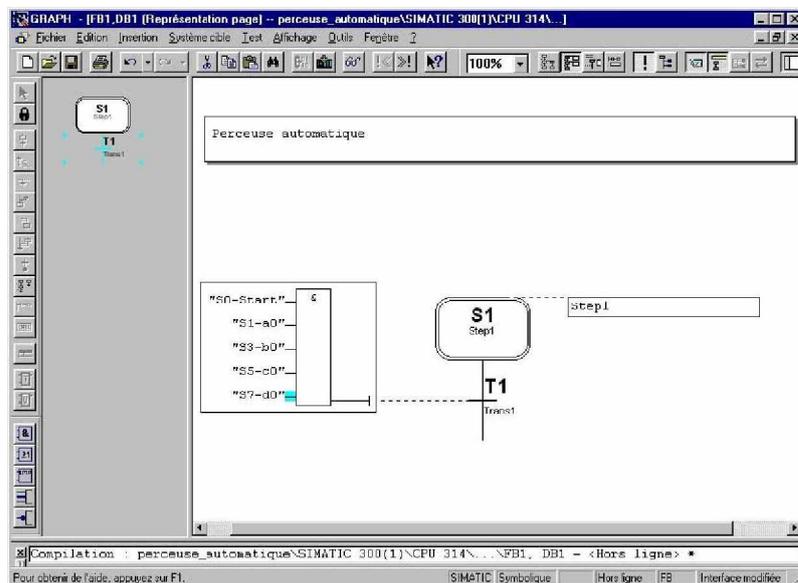
14.9 Elément du Graph-7

	Etape et transition		Ouvrir Branche ET
	Insérer un saut		Fermer Branche ET
	Puits (fin de graphe)		Ouvrir Branche OU
	Insertion d'étapes		Fermer Branche OU

14.10 Créer le Graph-7 selon l'enchaînement choisi

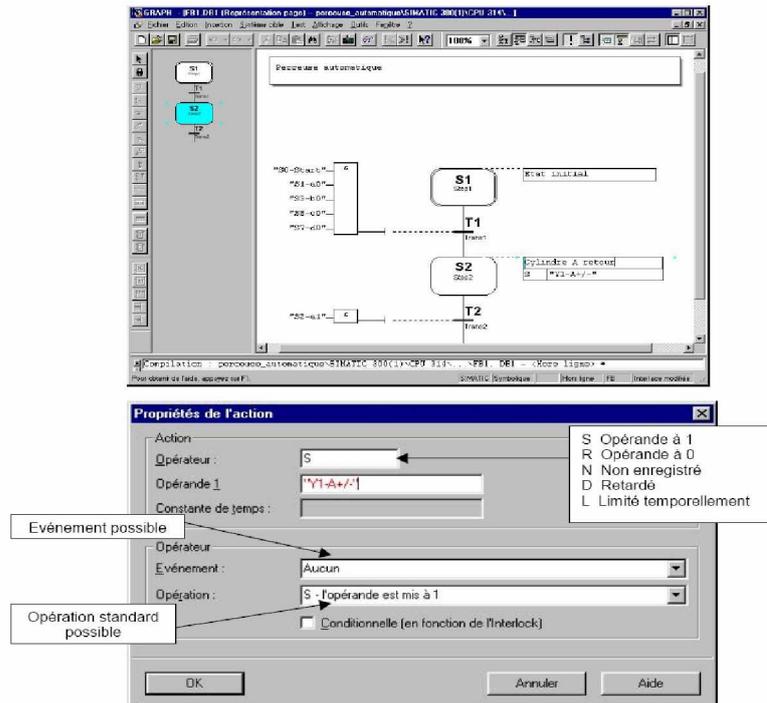
14.10.1 Première étape

1. Double-cliquer et saisir le **commentaire de bloc** et la **désignation de l'étape**
2. Cliquer sur l'**entrée de la transition**.
3. Insérer une **boîte ET**.
4. Ajouter des **entrées binaires**.
5. Saisir les **opérandes de la boîte ET**.
6. Cliquer sur la **transition T1**.
7. Insérer une **étape + transition**. La deuxième étape est insérée.



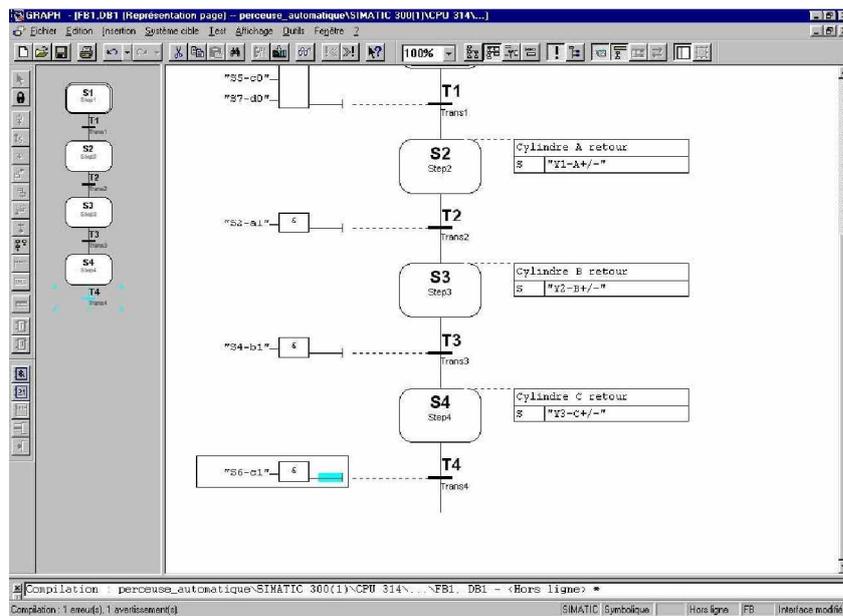
14.10.2 Deuxième étape

1. Cliquer sur le **commentaire** et saisir le nom de l'étape.
2. Insérer une **action**.
3. Saisir les actions à exécuter en double-cliquant ou avec les bouton droit de la souris en choisissant **propriétés de l'objet**.
4. Saisir la transition.



14.10.3 Troisième et quatrième étapes

1. Cliquer sur la **transition T2**.
2. Cliquer deux fois sur **étape + transition** pour insérer l'étape S3 et S4.
3. Entrer les commentaires et les actions.
4. Entrer les transitions.



Les étapes suivantes vont être insérées avec un branchement.

Il existe deux catégories de branchements

La branche OU,  va être insérée après l'étape choisie et commence avec une transition. Toutes les étapes de la branche OU vont être traitées seulement si la condition de transition est remplie.

Le branchement peut se terminer soit par une transition à gauche, soit par un puits.

 Terminer la branche OU.

 Insérer un puits.

 La branche ET, va être insérée après la transition choisie et commence avec un état.

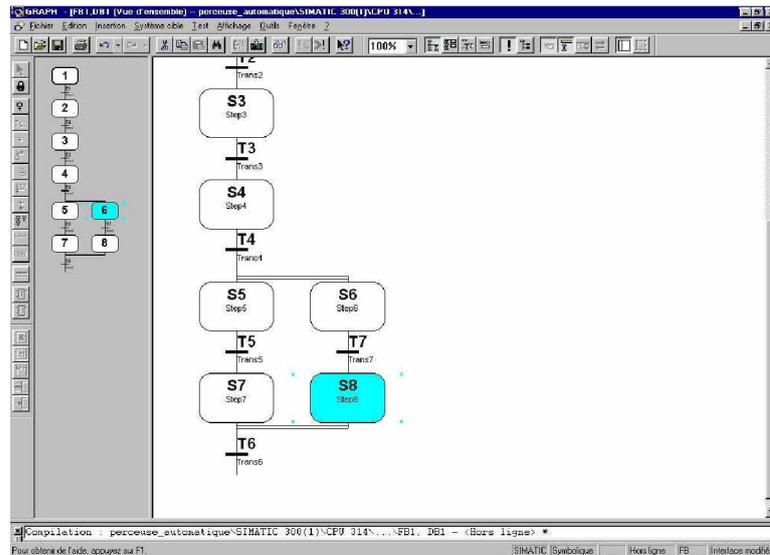
Toutes les étapes de la branche ET vont être traitées puisqu'elles s'exécutent en parallèle du branchement principal.

Le branchement se termine par une étape à gauche.

 Terminer la branche ET.

14.10.4 Insérer un branchement

Pour insérer des branchements, il est préférable de travailler en mode vue d'ensemble.



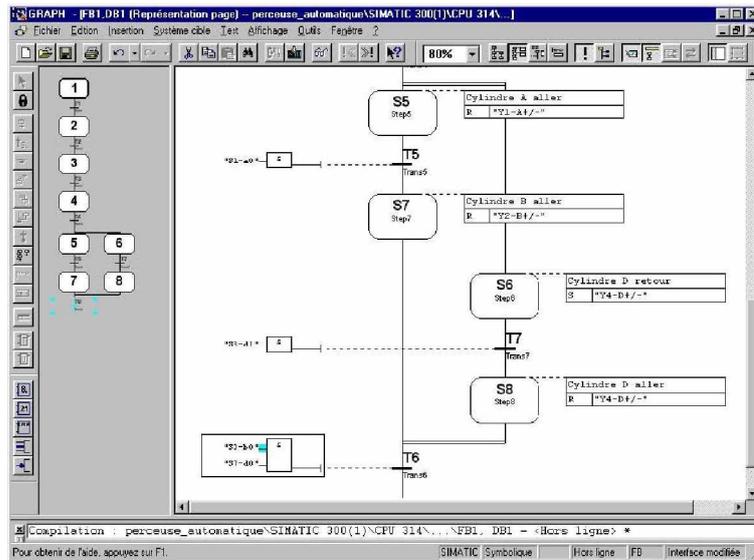
Pour créer une branche ET, procéder comme suit.

1. Cliquer sur la **transition T4**.
2. Insérer **étape + transition** (L'étape S5 et la transition T5 sont insérées).
3. Cliquer sur la **transition T4**.
4. Insérer une **branche ET** (L'étape S6 est insérée).
5. Cliquer sur la **transition T5**.
6. Insérer **étape + transition** (L'étape S7 et la transition T6 sont insérées).
7. Cliquer sur l'**étape S6**.
8. Insérer **étape + transition** (L'étape S8 et la transition T7 sont insérées).
9. Cliquer sur l'**étape S8**.
10. Cliquer sur fermeture de la **branche ET**.
11. Cliquer sur l'**étape S7**.

Pour saisir les actions et les transitions, vous devez vous mettre en représentation page.

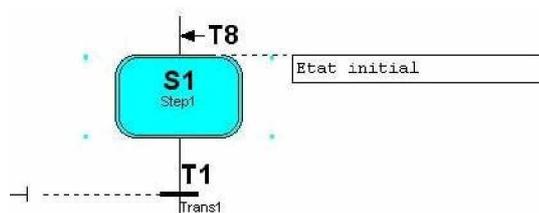
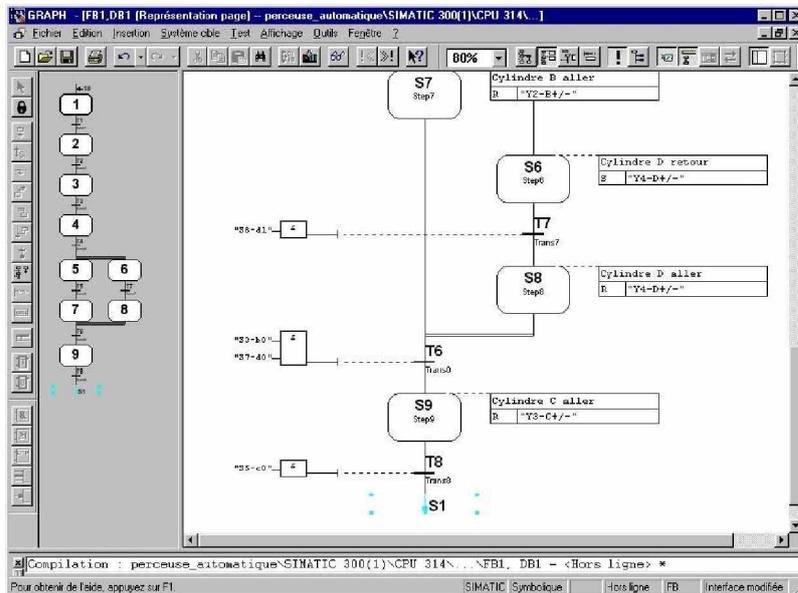
14.10.5 Insérer les actions et les transitions de l'étape 5 à 8 et insérer la dernière étape

1. Passer en **représentation page**.
2. Saisir les actions.
3. Entrer les transitions.
4. Cliquer sur la **transition T6**.
5. Insérer **étape + transition**.



14.10.6 Traiter la dernière étape et insérer un retour vers l'étape initiale

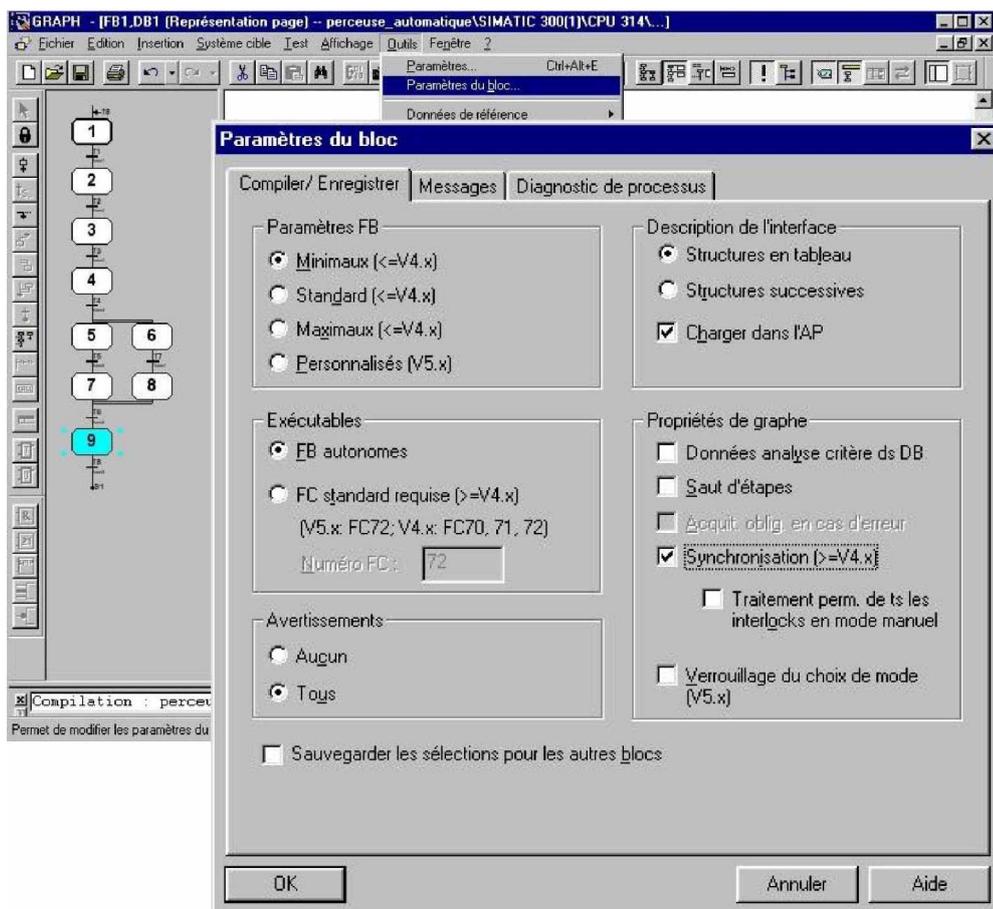
1. Saisir le commentaire et l'action.
2. Saisir la transition.
3. Cliquer sur la **transition T8**.
4. Insérer un saut.
5. Saisir S1 sur le saut ou cliquer sur l'**étape S1** afin d'insérer automatiquement le saut.



14.10.7 Configurer les paramètres du bloc et enregistrer le bloc fini

Avant l'enregistrement du bloc vous devez modifier ses propriétés.

1. Cliquer sur **Outils**.
2. Cliquer sur **paramètres du bloc**.
3. Mettre les paramètres FB à **Minimaux**.
4. Mettre exécutable à **FB autonome** afin que le FC standard soit relié au bloc fonctionnel.
5. Cocher **Synchronisation**.
6. Cliquer sur **OK**.
7. Enregistrer le bloc.
8. Fermer S7-GRAPH.

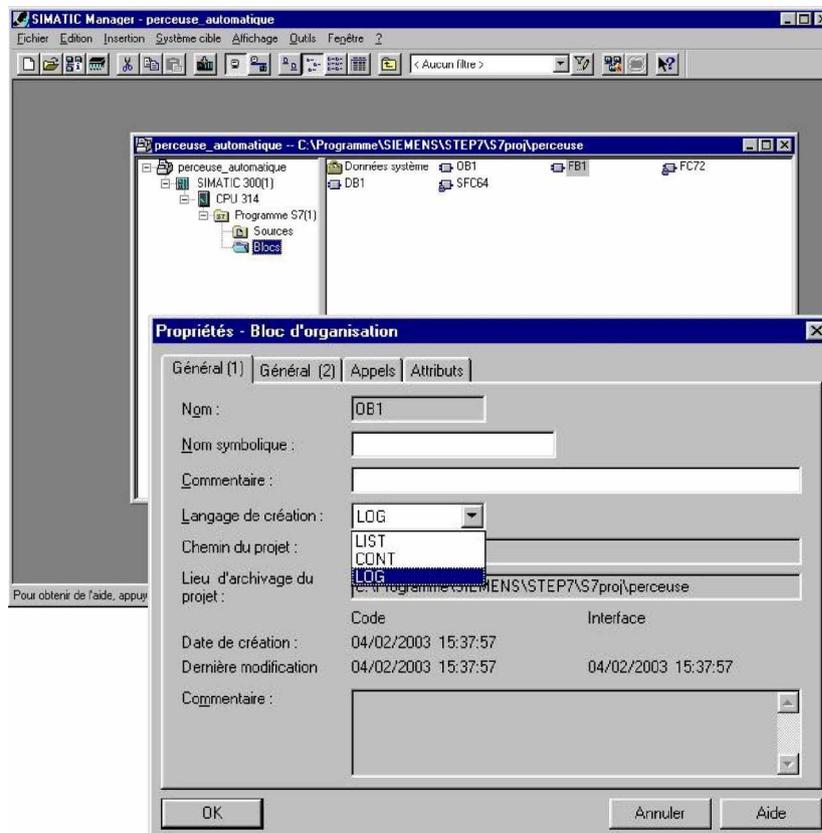


Remarque

S'il y a des erreurs dans le bloc alors celui-ci ne peut pas être enregistré. Un bloc erroné peut être généré seulement en tant que source. Avec cette sauvegarde, le bloc de données correspondant et le SFC64 sont produits. Les deux seront copiés dans le répertoire du bloc.

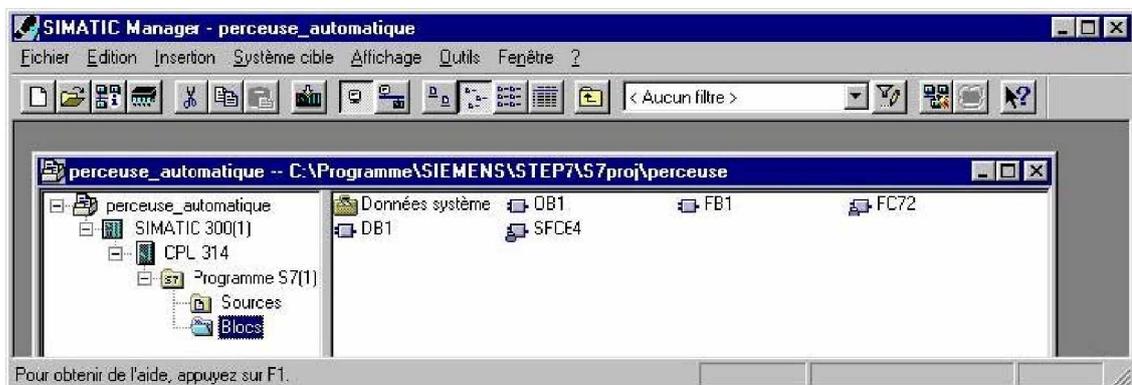
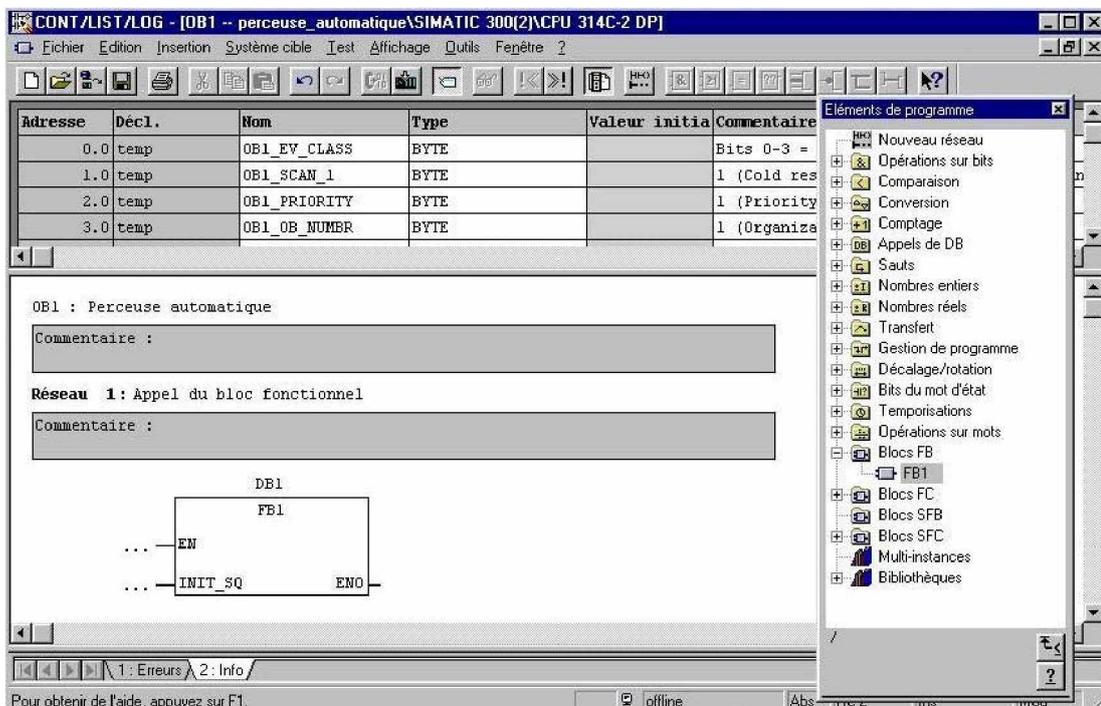
14.11 Ajuster les propriétés du bloc d'organisation et ouvrir OB1

1. Cliquer sur **Blocs**.
2. Double-cliquer sur **OB1**.
3. Choisir **LOG** en tant que langage de création dans les propriétés du bloc d'organisation.
4. Cliquer sur **OK**.



14.12 Traiter le bloc d'organisation OB1 et charger les blocs dans le module

1. Saisir la rubrique de blocs et la rubrique de réseau.
2. Cliquer dans le champ de saisie.
3. Ouvrir **Eléments de programme**.
4. Insérer **FB1** en double-cliquant.
5. Entrer DB1.
6. Enregistrer le bloc OB1.
7. Fermer CONT/LIST/LOG.
8. Cliquer sur **blocs** et **charger dans le module**.



Après le transfert du bloc dans la CPU, le programme peut être testé.

CONCLUSION

Pour situer l'objectif du présent travail, nous avons commencé par l'historique des API, ensuite nous avons exposé, et d'une façon approfondie, les deux côtés: matériel et logiciel de l'API.

Le chapitre suivant sera consacré au choix de l'API pour le processus de "TRAITEMENT DE SURFACE".



CHAPITRE III

COMMANDE PAR API POUR LA CHAÎNE DE TRAITEMENT DE SURFACE

INTRODUCTION

Au cours de ce chapitre nous allons voir un exemple de programmation des API pour l'élaboration d'un programme de commande du processus de traitement de surface. La programmation par STEP7 "Grafcet" consiste en le remplacement des différentes fonctionnalités et composants câblés par des objets fictifs simplement manipulables et ajustables.

1.REALISATION DU GRAFCET DE LA CHAINE

Les processus industriels sont, dans la plupart des cas, des systèmes séquentiels. On peut alors les traduire sous forme de GRAFCET ou de réseaux de Pétri.

Chaque événement de ces processus peut être représenté sous forme d'étape, transition ou actions, donc peut être programmé avec S7Graph.

Avant d'écrire le programme, il est tout d'abord nécessaire de créer un projet, de configurer le matériel, définir la table des mnémoniques et insérer les blocs nécessaires pour la mise en marche du projet

Remarque

Si on veut travailler avec le S7Graph les FB (Blocs Fonctionnels) insérés doivent être de type GRAPH.

Dans le STEP 7 Les OB ne peuvent pas être de type GRAPH, il sont de type CONT, LOG, ou LIST.

Dans l'exemple cité dans ce chapitre, la configuration matérielle de base est :

- § CPU S7-314 IFM 6ES7 314-5AE83-0A0B version matérielle 1.2
- § Alimentation PS 307 10A 6ES7 307-1KA00-0AA0
- § Coupleurs pour 1 châssis d'extension 6ES7 365-0BA81-0AA0
- § 3 Modules de 16 entrées TOR, AC120/230V volts ES7 321-1BL80-0AA0
- § 3Module de 16 sorties TOR 6ES7 322-1EL00-0AA0
- § 1Module de 4 entrées analogique 20 mA

La table des mnémonique a été défini à partir des entrées, sorties, mémentos (variables internes ou momentanées), tempos (variables temporelles), ..., qui sont définies par les capteurs, les actionneurs, ..., déjà existants dans l'installation initiales de la chaîne.

Pour mieux gérer le système global, on a préféré de le subdiviser en plusieurs sous-systèmes. Ceci permet de mieux tester et déboguer les programmes.

Ce choix se traduit par la réalisation de chaque partie du système soit par une sous-routine soit par une fonction.

Figure illustrant la structuration du système global :

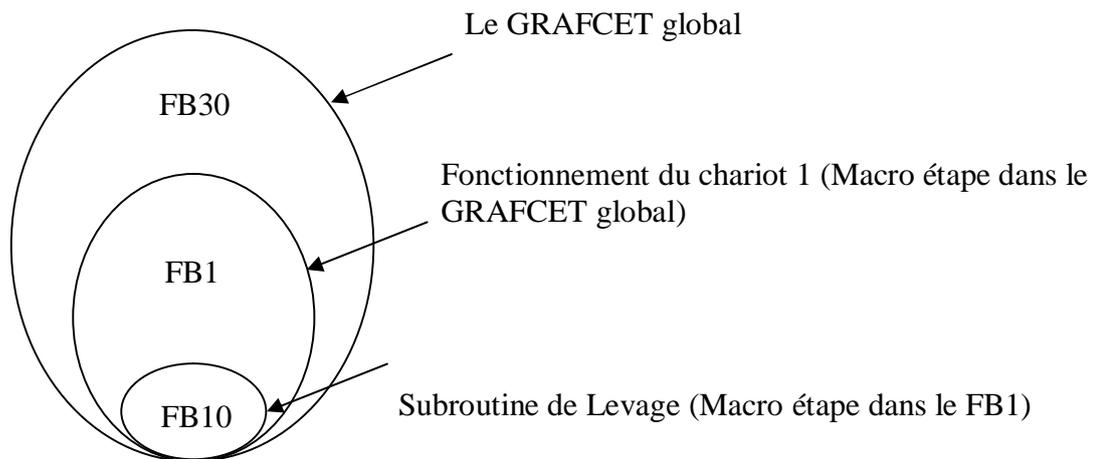


Figure 22: Structuration du système

Pour réaliser un GRAFCET, il est préférable d'utiliser une sous-routine qui, à la différence d'une fonction, possède des données statiques contenues dans le bloc de données (DB) associé. Ceci permet de sauvegarder des variables pour une utilisation dans le cycle qui suit.

Afin d'optimiser le programme, on crée différents blocs (OB : blocs d'organisations, FB : blocs fonctionnels,...).

Le tableau suivant nous présente les différents blocs créés pour le projet de traitement de surface :

Nom de l'objet	Langage de création	Type	commentaire	DB associé
Blocs d'organisations				
OB1	CONT	Bloc d'organisation	Cycle d'exécution	-
Blocs fonctionnels				
FB1	GRAPH	Bloc fonctionnel	NIVEAU DE SECURITE	DB1
FB2	GRAPH	Bloc fonctionnel	PRODUCTION NORMALE	DB2
FB3	GRAPH	Bloc fonctionnel	Niveau conduit	DB3
FB4	GRAPH	Bloc fonctionnel	commande manuelle	DB4
FB5	GRAPH	Bloc fonctionnel	MISE EN PO	DB5
FB6	GRAPH	Bloc fonctionnel	REGENERATION	DB6

Tableau 8: Les Blocs du projet traitement de surface

Remarque :

Lors de la mise en marche de l'automate, les valeurs contenues dans la RAM peuvent être quelconques (1 ou 0) selon l'historique de la CPU (démarrage à chaud, etc...).

Il est donc nécessaire d'initialiser Le GRAFCET en utilisant l'OB100 qui est le bloc d'organisation exécuté lors de la mise en RUN de la CPU.

2. LE GRAFCET DU PROJET TRAITEMENT DE SURFACE :

2.1. Le Grafcet niveau de sécurité :

Le GRAFCET NIVEAU DE SECURITE du projet est représenté par le bloc fonctionnel FB1, qui est un bloc programmé par S7Graph et qui contient les condition de démarrage

Le bloc FB1 est composé de :

- § Etape initiale pour initialiser les mnémoniques nécessaires pour le fonctionnement des autres blocs du projet.
- § Une branche **ou** pour que les blocs **FB1, FB2, FB3, FB4, FB5, FB6** puissent marcher en parallèle (momentanément).
- § Etapes **S1, S2, S3, S4, S5, S6, S7**

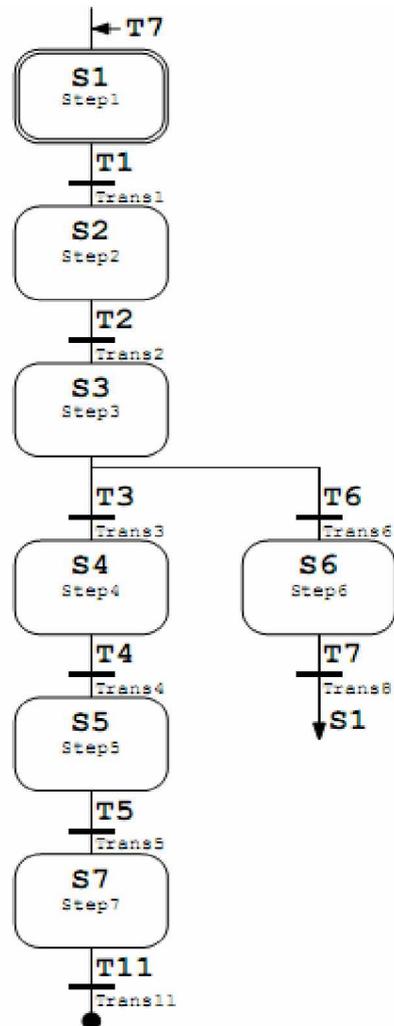
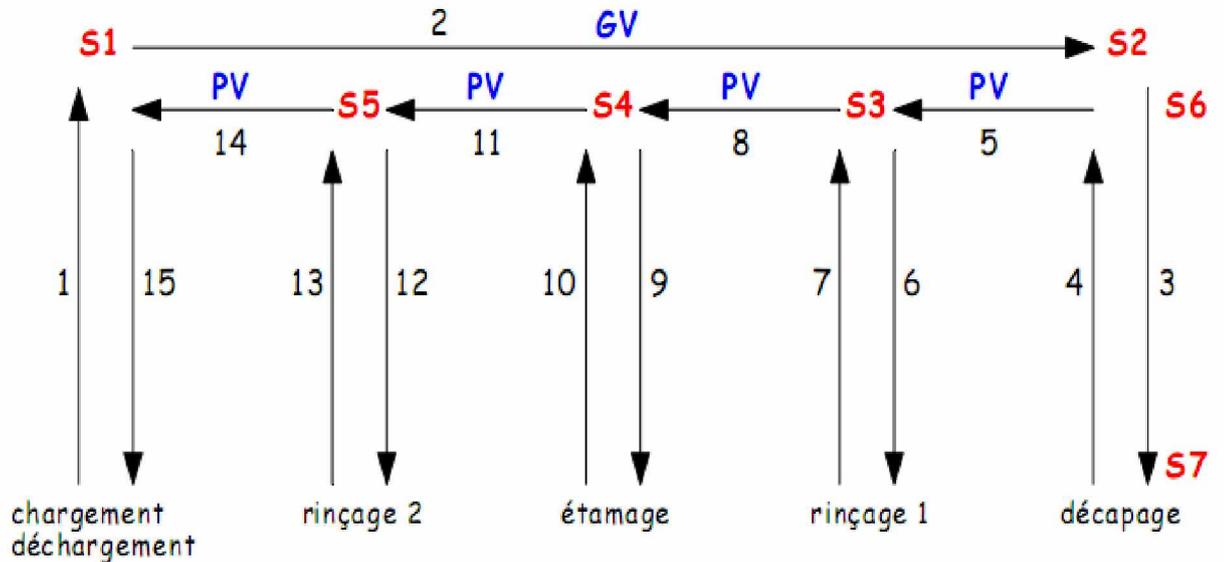


Figure 23: Vue du bloc fonctionnel FB1

2.2. GRAFCET du projet :

2.2.1. GRAFCET de production:



- **Alimentation** : 3 X 380 V + N + PE.
- **Mouvement de levage** :
 - **actionneur** : moteur frein asynchrone triphasé (ML) ;
 - **réducteur de vitesse** ;
 - **pré actionneur** : contacteur inverseur (KMDE – KMMO).
- **Mouvement de translation du chariot** :
 - **actionneur** : moteur frein asynchrone triphasé bi vitesse par bobinages séparés (MT)
 - **pré actionneurs** : contacteurs inverseurs (KMDR – KMGA et KMPV – KMGV).
- **Chauffage des bains** : réchauffeurs (EE et ED) de chez Vulcanic + régulateurs + relais statiques.
- **Capteurs** :
 - **S1 à S5** : interrupteurs de position à tige à galet (XCR A15 de chez Schneider)
 - **S6 et S7** : interrupteurs de position à tige carrée (XCR E18 de chez Schneider)
 - **S20 et S21** : interrupteurs de position à levier à galet (XC1 AC116 de chez Schneider) ;
 - **S15 et S16** : sondes PT100 de chez Vulcanic.

Remarque :

Pour mieux illustrer le GRAFCET de production (bloc fonctionnel FB2), Il sera présenté dans une annexe avec les mnémoniques utilisées pour son fonctionnement.

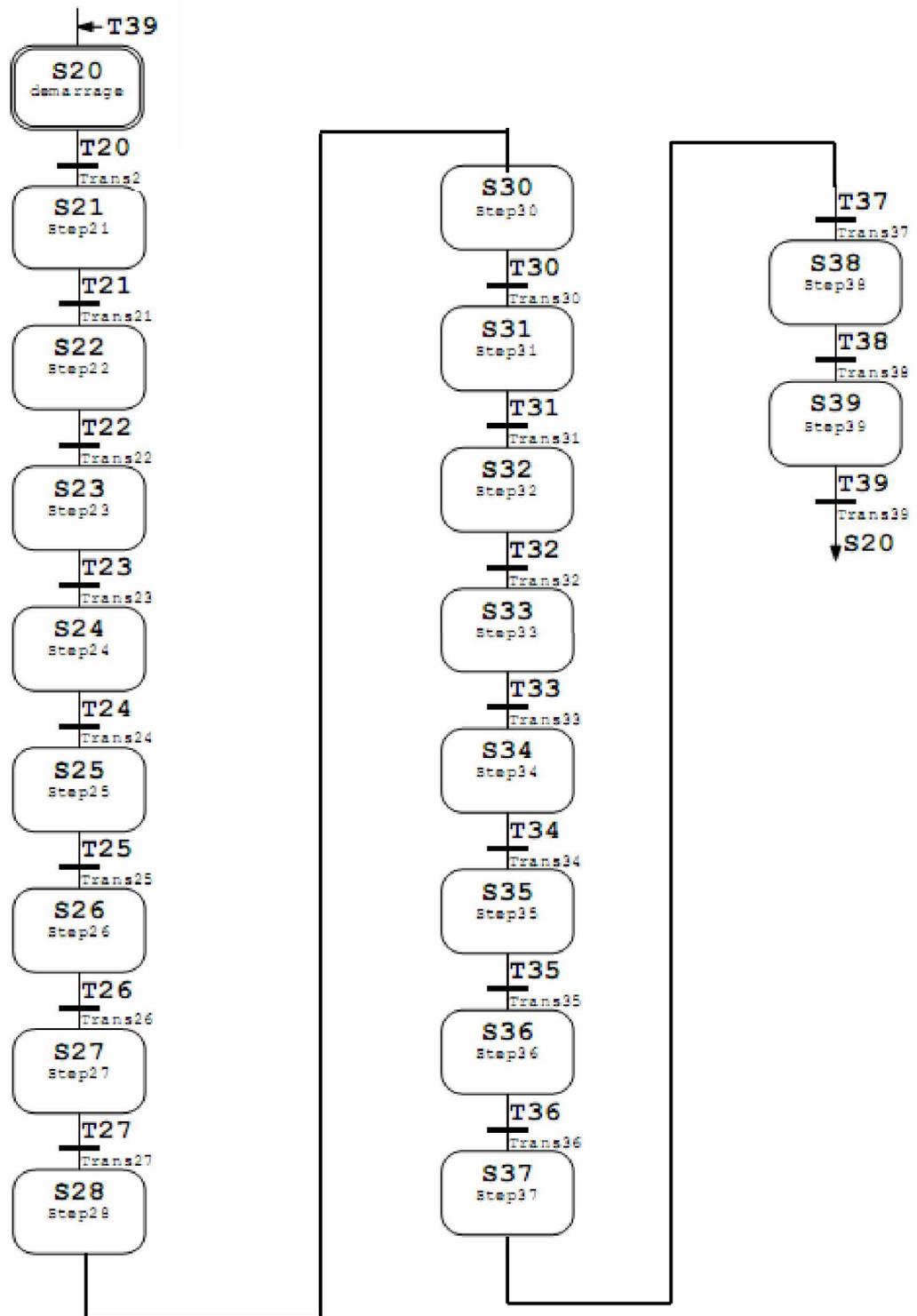


Figure 24: Production normale FB2

2.2.2. Grafcet niveau conduite :

X3 : niveau sécurité.

X46 : niveau commande (GRAF CET de MISE en REFERENCE P.O.).

X28 : niveau commande (GRAF CET de PRODUCTION NORMALE).

X52 : niveau commande (GRAF CET de REGENERATION).

N = nombre de pièces traitées, l'incrémentation du compteur s'effectue au niveau commande (GRAF CET de PRODUCTION NORMALE).

Manu/Auto : bouton tournant (S28).

Réf. P.O. : bouton-poussoir (S29).

dcy : bouton-poussoir (S12).

Machine prête : voyant (H3)

Température Atteinte : informations fournies par les régulateurs de température (raccordées en série).

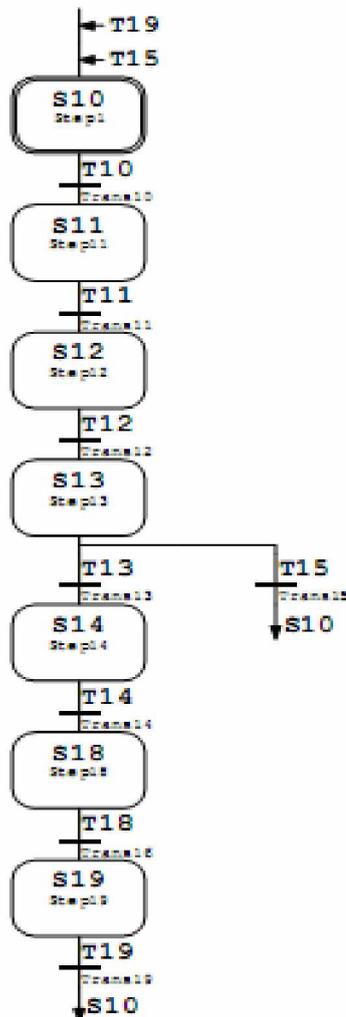


Figure 25: Niveau conduite FB 3

2.2.3. commande manuelle :

↑ (Montée panier) : bouton-poussoir (S 22).

↓ (Descente panier) : bouton-poussoir (S23).

← (Translation gauche) :

bouton-poussoir (S24). →

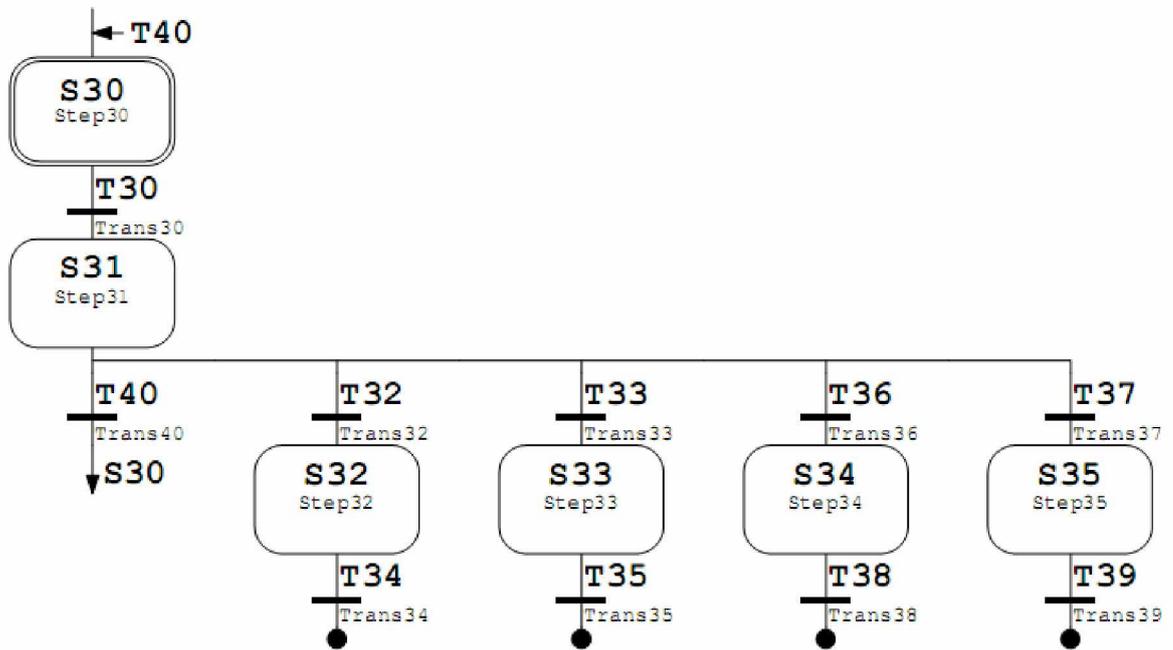


Figure 26: Commande manuelle

2.2.3 Mise En PO

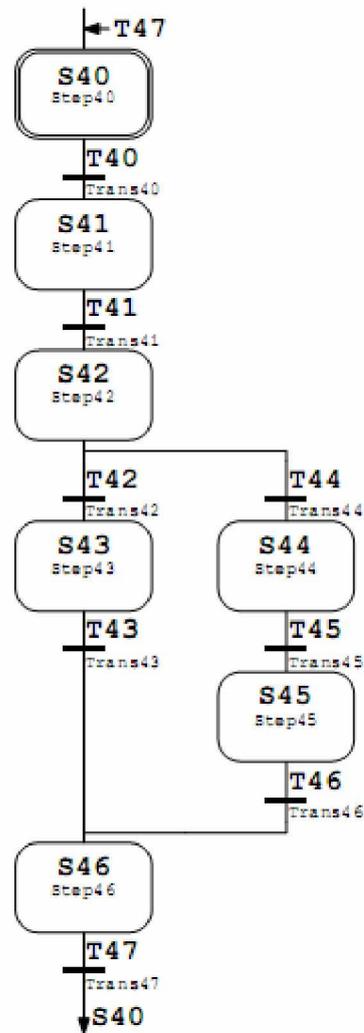
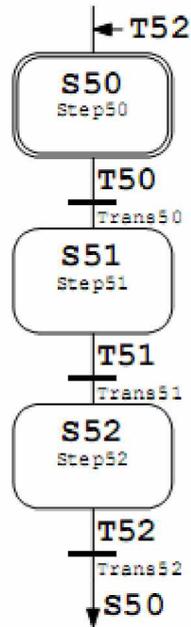
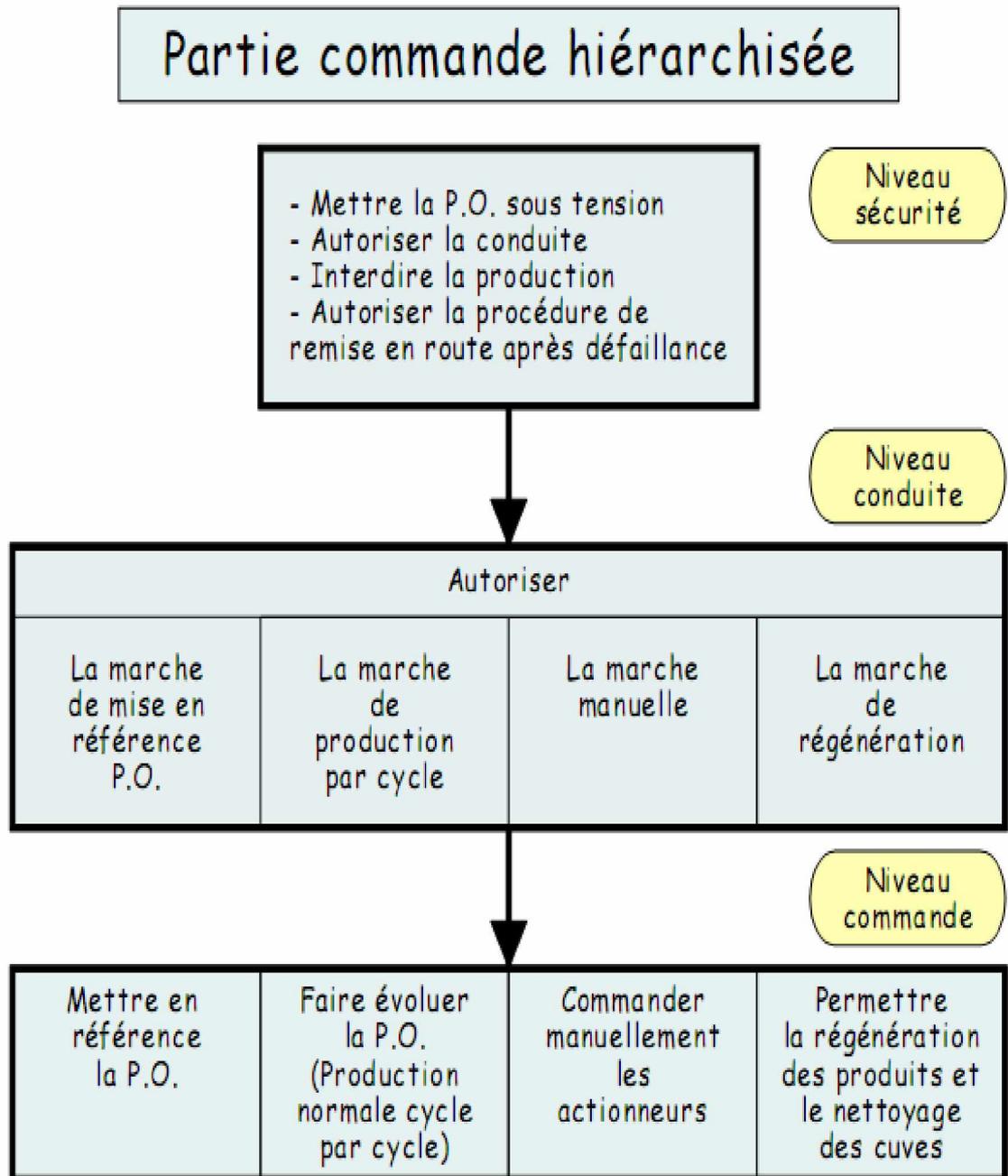


Figure 27: Mise en P.O

2.2.4 Régénération

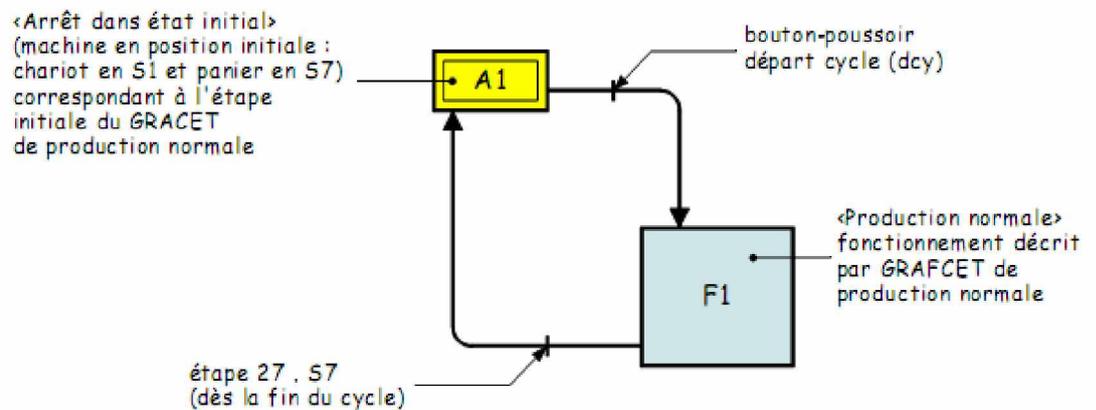


2.2.5 Organisation



2.3 Etude des modes de marche et d'arrêt

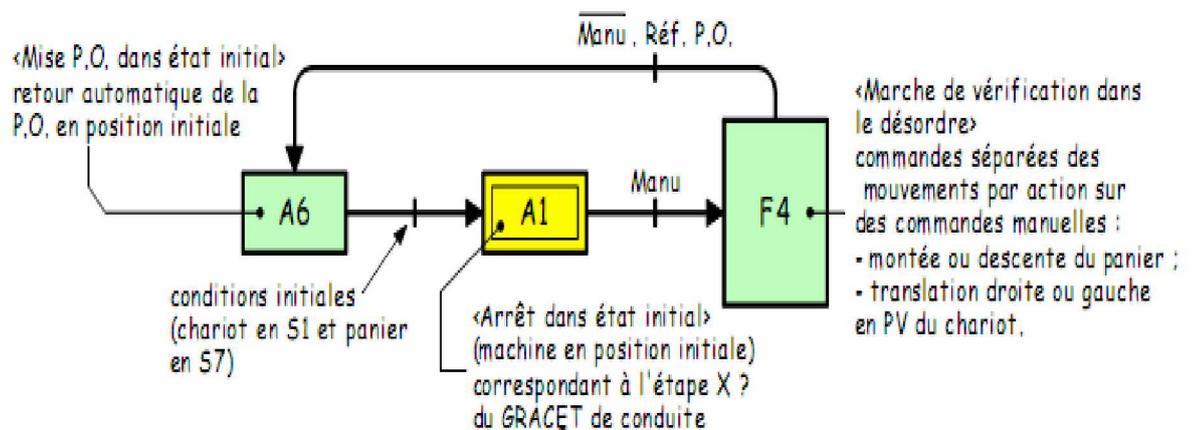
2.3.1. Marche de production cycle par cycle



Fonctionnement

- A chaque fin de cycle, l'opérateur doit appuyer sur le bouton-poussoir départ cycle (dcy).
- Un appui permanent sur celui-ci entraîne la répétition des cycles.

2.3.2. Marche de vérification dans le désordre et mise P.O. dans état initial



Fonctionnement

- La machine étant à l'arrêt en position initiale (rectangle-état A1) , l'opérateur sélectionne le mode Manu (rectangle-état F4), il lui est alors possible de commander les différents mouvements dans le désordre.

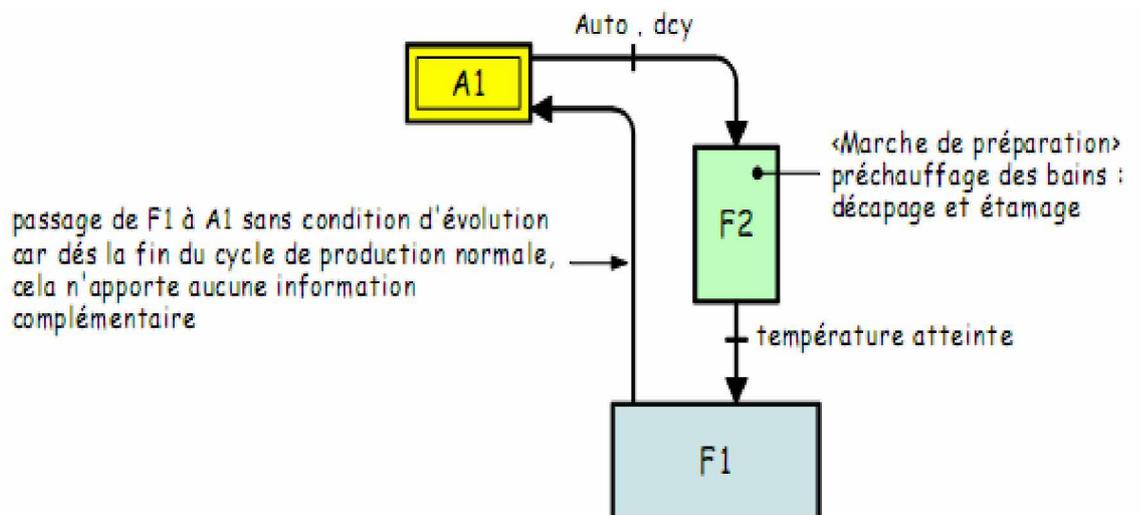
- Le retour automatique de la P.O. en position initiale s'effectue de la manière suivante : /Manu et impulsion sur un bouton-poussoir à impulsion Réf. P.O. (rectangle état A6).

- Les conditions initiales étant réunies, la machine se retrouve de nouveau en position initiale (rectangle-étatA1).

2.3.3. Marche de préparation (préchauffage des bains)

Il est nécessaire de préchauffer les bains (décapage et étamage) avant de lancer la production normale.

Cette procédure de fonctionnement sera décrite dans le rectangle état F2 <Marche de préparation>.



Remarques :

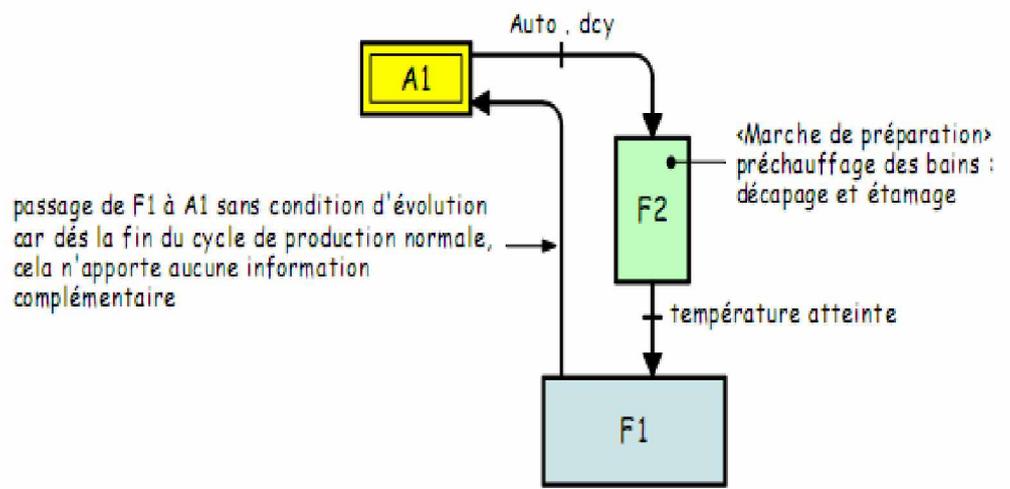
- Auto est un bouton tournant à manette à 2 positions fixes (Auto – Manu) (S28).
- L'information "température atteinte" dans les bains est fournie par les régulateurs de température

2.3.4. Marche de clôture (régénération des produits et nettoyage des cuves)

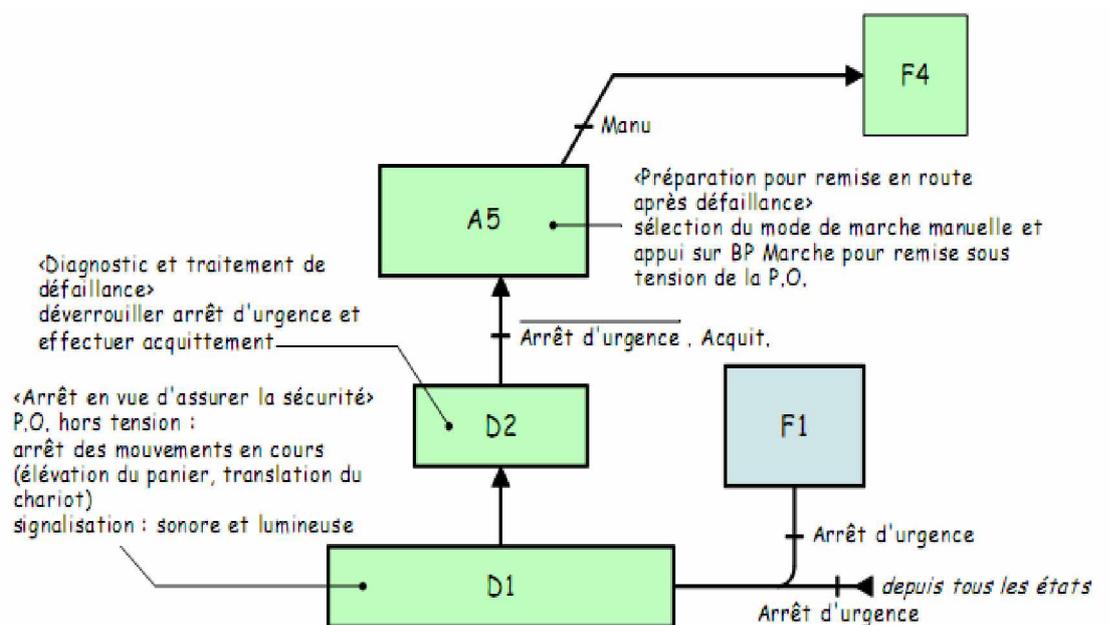
Cette opération est à réaliser au bout d'un certain nombre de pièces traitées (N = nombre de pièces), un comptage

est donc nécessaire, il sera effectué dans le GRAFCET de production normale.

Remarque : un bouton-poussoir à impulsion changement terminé (S27) est à rajouter sur le pupitre de commande.



2.3.5. Arrêt en vue d'assurer la sécurité, diagnostic et traitement de défaillance, préparation pour remise en route après défaillance



2.3.5.1. Arrêt en vue d'assurer la sécurité des biens et des personnes (traitement de l'arrêt d'urgence). Généralités

a) Intérêt d'un arrêt d'urgence

L'arrêt d'urgence doit permettre d'assurer la sécurité dans les cas suivants :

- incident de production (exemple : mauvais positionnement d'une pièce dans le panier) ;
- incident sur la machine (exemple : défaillance mécanique d'un organe) ;
- sécurité de l'utilisateur et des tiers (remarque : l'opérateur doit avoir une vision complète sur la machine) ;
- ...

La commande d'un arrêt d'urgence peut provenir :

- d'une intervention humaine (bouton AU placé sur le pupitre de commande) ;
- de la détection d'une sur-course [remarque : cela peut être envisagé sur la machine étudiée (interrupteurs de position S20 et S21)] ;
- action sur un chasse-corps situé sur une navette d'un transbordeur de ligne ;
- d'une coupure d'énergie ;
- ...

b) Effet d'un arrêt d'urgence sur la partie opérative :

- arrêt normal en fin de mouvement ;
- cycle de dégagement (exemple : assurer le retour d'une unité de perçage puis desserrer la pièce sur une machine à percer automatique) ;
- blocage d'un mouvement (exemple : utilisation d'un bloqueur permettant l'arrêt d'un vérin en cours de course) ;
- mise hors tension de la P.O. [pour notre exemple : arrêt des mouvements en cours (élévation du panier et translation du chariot), le moteur d'élévation est équipé d'un frein à manque de courant].- ...

c) Où et comment traiter l'arrêt d'urgence ?

- L'arrêt d'urgence, « ordre prioritaire », doit faire évoluer le système quelle que soit la situation

(GRAFCET : étape) dans laquelle il se trouve.

- L'arrêt d'urgence sera traité dans le GRAFCET de SECURITE, hiérarchiquement supérieur au

GRAFCET de CONDUITE, lui même hiérarchiquement supérieur au GRAFCET de PRODUCTION NORMALE.

2.3.5.2. Diagnostic et traitement de défaillance, préparation pour remise en route après défaillance

Après avoir diagnostiqué la cause de l'arrêt d'urgence (exemple : mauvais positionnement d'une pièce pour la machine étudiée) et après avoir effectué le traitement adéquat (exemple : pièce remise correctement dans le panier pour la machine étudiée), il est alors possible de préparer la machine pour une remise en route.

Fonctionnement

- La P.O. étant hors tension (rectangle-état D2), l'opérateur doit déverrouiller l'arrêt d'urgence (AU1 + AU2).

- Un acquittement de la part de l'opérateur est obligatoire (bouton-poussoir à impulsion Acquit.) afin qu'il puisse

prendre conscience des conséquences de son geste. Cela entraîne l'évolution suivante : état D2 à état A5.

- En l'état A5, la remise sous tension de la P.O. s'effectue de la manière suivante :

- sélection du mode de marche manuelle obligatoire (Manu) ;

- impulsion sur le bouton poussoir Marche pour remise sous tension de la P.O. ;

- cela entraîne l'évolution suivante : état A5 à état F4.

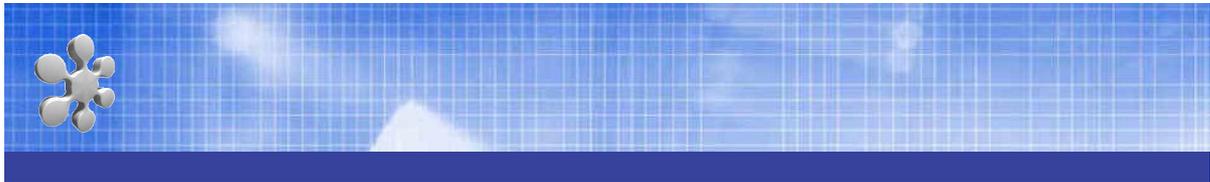
Remarque :

un bouton-poussoir à impulsion Acquit. (S30) est à rajouter sur le pupitre de commande.

CONCLUSION

Nous avons consacré ce chapitre à l'analyse et l'élaboration d'un programme pour le contrôle des différentes opérations de traitement de surface en adoptant comme langage de programmation, le Step7. L'analyse a concerné surtout, les entrées, les informations provenant des différents détecteurs et capteurs implantés partout où l'on veut s'informer du processus et des actions à déclencher pour en modifier l'état.

Nous venons, par conséquent, d'achever l'étape la plus importante du travail, et il ne reste qu'à évaluer le fonctionnement par simulation ou donc sur le terrain.



CONCLUSION GENERALE



Au cours de notre étude, et en vue de remplacer le dispositif de contrôle de la chaîne de traitement de surface, nous avons procédé par une étude descriptive du processus et de la chaîne ainsi que tous les accessoires complémentaires.

Les traitements de surface sont utilisés pour modifier les caractéristiques de la surface d'une pièce dans le but de lui donner des qualités nouvelles : améliorer les propriétés mécaniques (dureté, frottement, résistance à l'usure, au grippage, à la fatigue...), la résistance à la corrosion, l'esthétique et certains comportements (conductivité électrique, réflexion de la lumière, conduction de la chaleur...).

Nous avons jugé intéressant et commode de faire appel à la technologie programmée (Automate programmable Industriel), et ce pour des raisons de :

- Complexité de l'installation existante (logique électrique câblée),
- Diagnostic des pannes et entretien relativement difficiles,
- Problème de l'inflexibilité (installation de contrôle figée),
- Efficacité minimale et autres problèmes économiques tels que : taux de productivité, coûts d'entretien...etc.

Ensuite, dans le deuxième chapitre, l'étude théorique sur les API a été établie. Nous avons décrit leurs architectures comme : le CPU et les modules d'entrées/sorties ... en terme matériel ; l'interface de programmation ...en terme logiciel.

Et, nous avons présenté une étude assez détaillée sur le SIMATIC S7-314 IFM de SIEMENS, qui a été notre choix pour l'application visée ainsi que les raisons et les motifs du choix qui se manifestent essentiellement dans :

- Performances techniques (fréquence de traitement, type de signaux traités ... etc.),
- Capacité mémoire assez suffisante.
- Possibilité d'extension en terme du nombre élevé des entrées et des sorties du processus, exigé par certaines applications. Laquelle nous a

permis d'intégrer trois autres modules de 32 entrées TOR et deux modules de 32 sorties TOR.

- Possibilité d'intégrer des modules spéciaux (modules d'entrées ou de sorties analogiques, régulateur de température, de pression, ...etc.).
- Disposition du CPU 314 IFM de fonctions intégrés telles que : le fréquencemètre, les compteurs, ...etc.

Quant au troisième chapitre, il a été consacré à la programmation graphique dite **GRAF CET** du processus de traitement de surface.

Nous avons consacré ce chapitre à l'analyse et l'élaboration d'un programme pour le contrôle des différentes opérations de traitement de surface en adoptant comme langage de programmation, le Step7. L'analyse a concerné surtout, les entrées, les informations provenant des différents détecteurs et capteurs implantés partout où l'on veut s'informer du processus et des actions à déclencher pour en modifier l'état.

Nous venons, par conséquent, d'achever l'étape la plus importante du travail, et il ne reste qu'à évaluer le fonctionnement par simulation ou donc sur le terrain.

Enfin, on pourrait dire que remplacer les dispositifs classiques de contrôle par les Automates Programmables Industriels 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.



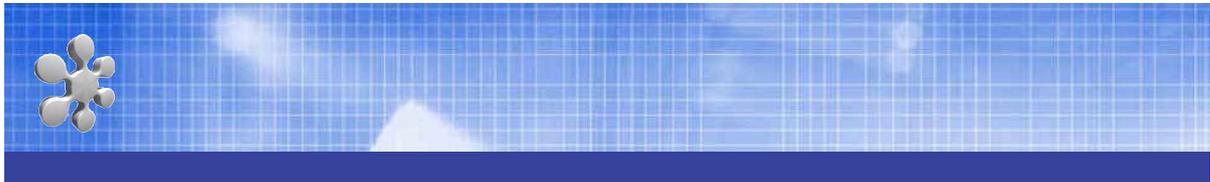
REFERENCES

BIBLIOGRAPHIQUES



- [1]. BARA J.-C., *TRAITEMENT DE SURFACE, ENVIRONNEMENT ET COMPETITIVITE, EDITIONS EYROLLES, 1988.*
- [2]. Bourgois J., Debray B., Laforest V., "Les traitements chimiques et physicochimiques des déchets", *Les Techniques de l'Ingénieur, Volume G, article 2070, soumis.*
- [3]. Catonné J.C., *Matériaux métalliques, Techniques de l'ingénieur, tome M5, n° M1427, non daté.*
- [4]. Gaïda B., *L'électroplastie en questions et réponses, Editions Librairie de traitements de surface, 1983.*
- [5]. Hiott C., *La galvanotechnique, Pourquoi et comment?; Editions Librairie de traitements de surface, 1987.*
- [6]. Leclerc O., *Traitement de surface : Techniques électrolytiques de dépollution, Les cahiers de l'ingénieurs, n°40, 1991.*
- [7]. G. MICHEL, « *Les A.P.I. Architecture et application des automates programmables industriels* », Edition DUNOD ,1987.
- [8]. A. SIMON, « *Automate Programmable Industriel, Niveau 1* », Edition L'ELAN, 1991.
- [9]. M. BERTRAND, «*Automates programmables industriels* », *Techniques de l'ingénieur, Vol. S 8 015.*
- [10]. P. JARGOT, « *Langages de programmation pour API. Norme IEC 1131-3* », *Techniques de l'ingénieur , Vol. S 8 030.*
- [11]. D. DUPONT, D. DUBOIS, « *Réalisation technologique du GRAFCET* », *Techniques de l'ingénieur , Vol. S 8 032.*
- [12]. SCHNEIDER ELECTRIC, « *Automates Nano et Plate-forme d'automatisme Micro*»,1999.
- [13]. SIEMENS, « *Mise en route STEP7 V5.2.Getting Started* », Réf. 6ES7810-4CA06-8CA0,SIMATIC, 2002.

- [14]. SIEMENS, « Programmer avec STEP7 V5.2 », Réf. 6ES7810-4CA06-8CA0, SIMATIC, 2002.
- [15]. SIEMENS , «Langage CONT pour SIMATIC S7-300/400 », Réf. 6ES7810-4CA06-8CR0, SIMATIC,2002.
- [16]. SIEMENS , «Langage LOG pour SIMATIC S7-300/400 », Réf. 6ES7810- 4CA06-8CR0, SIMATIC,2002.
- [17]. SIEMENS, «Langage LIST pour SIMATIC S7-300/400 », Réf. 6ES7810-4CA06-8CR0, SIMATIC, 2002.
- [18]. SIEMENS, «Automate programmable S7-300, Fonctions intégrées, CPU 312 IFM/314IFM », Réf. EWA 4NEB 710 6058-03a Edition 2, SIMATIC, 2000.
- [19]. SIEMENS, « Liste des opérations S7-300, CPU 312 IFM, 314 IFM, 313, 314, 315,315-2 DP, 316-2 DP, 318-2 », Réf. 6ES7 398-8AA03-8CN0 Edition 2, SIMATIC,2000.
- [20]. SIEMENS, « Automate programmable S7-300, Installation et configuration - caractéristiques techniques des CPU», Réf. 6ES7 398-8AA03-8CA0 Edition 2, SIMATIC,1999.
- [21]. SIEMENS, « Automate programmable S7-300 Caractéristiques des CPU, CPU 312IFM–318-2 DP », Réf. 6ES7398-8FA10-8CA0, SIMATIC, 2001.
- [22]. SIEMENS, «Automate programmable S7-300, Getting Started, Premières étapes pour montage et mise en service », Réf. A5E00069341 01, SIMATIC, 2000.
- [23]. <http://www.unctad.org/infocomm/francais/zinc/plan.htm>

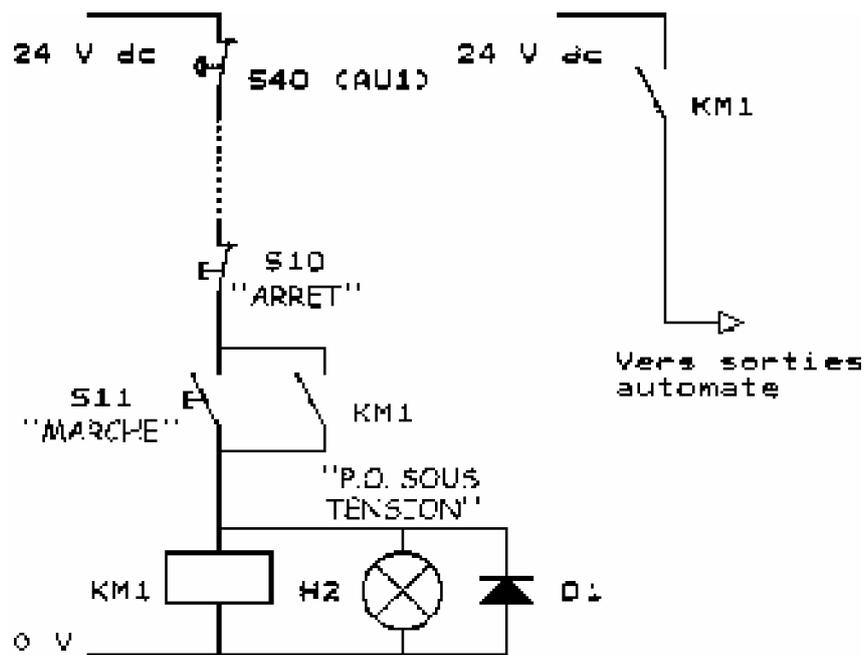
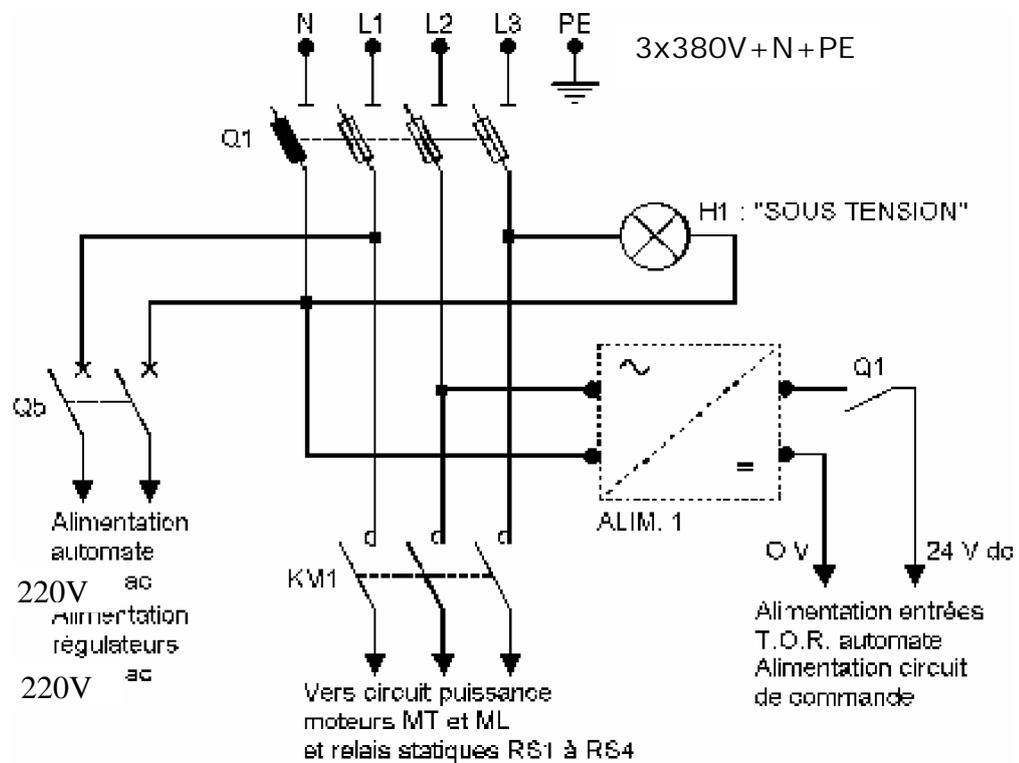


ANNEXES



Annexe A :

SCHEMAS DES ALIMENTATIONS



Annexe B :

TABLE DES MNÉMONIQUES

Propriétés de la table des mnémoniques

Nom : Mnémoniques
 Commentaire :
 Date de création : 18/04/2009 19:23:17pm
 Dernière modification : 06/06/2009 17:09:50pm
 Dernier filtre sélectionné : Tous les mnémoniques
 Nombre de mnémoniques : 50/50
 Dernier tri : Mnémonique ordre croissant

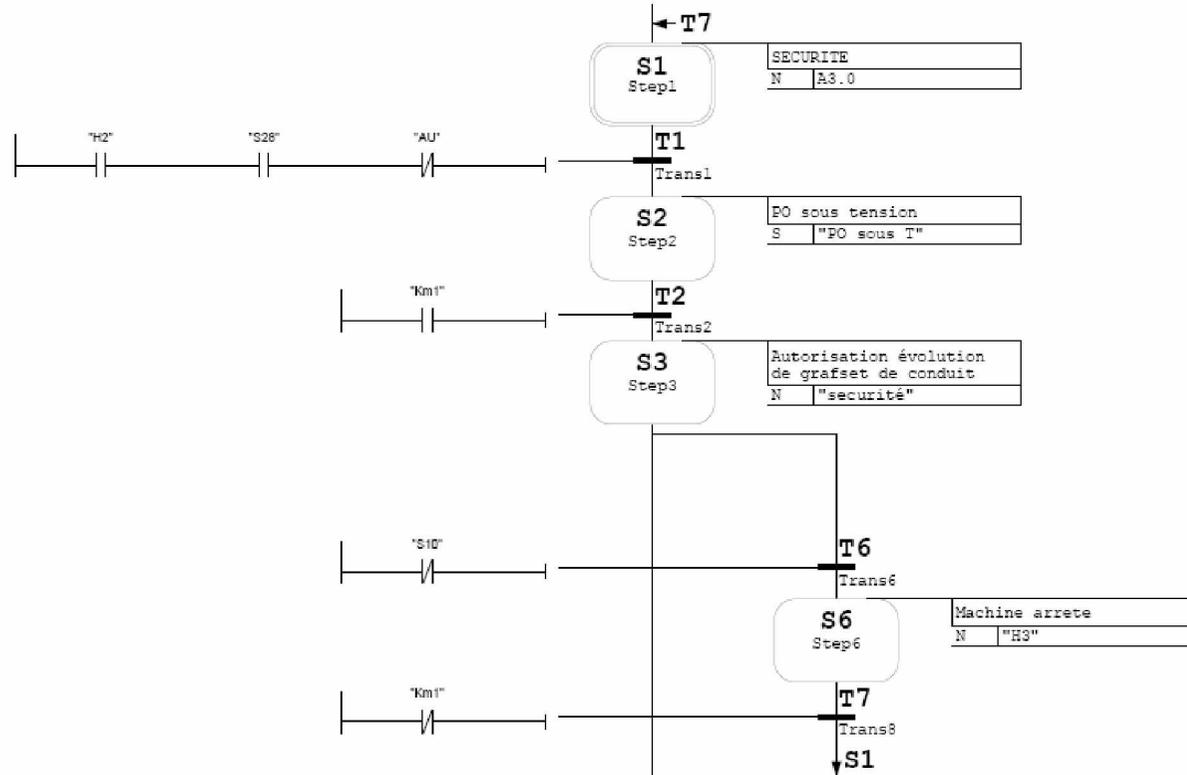
Etat	Mnémonique	Opérande	Type de données	Commentaire
	ALARM_S	SFC 18	SFC 18	Generate Permanently Acknowledged Block-Related Messages
	ALARM_SQ	SFC 17	SFC 17	Generate Block-Related Messages with Acknowledgment
	AU	E 1.6	BOOL	Au défaut
	AU COM MAN	M 1.1	BOOL	Autorisation de la commande manuel
	AUT MAN	A 1.6	BOOL	Autorisation de la commande manuel
	AUT REF P	A 1.7	BOOL	Autoriser de mise en référence P.O
	Cycle Execution	OB 1	OB 1	
	DC	E 0.2	BOOL	départ cycle
	f	VAT 1		
	G7_STD_3	FC 72	FC 72	
	G7_STD_4	FC 73	FC 73	
	H2	E 1.4	BOOL	P.O sous tension
	H3	A 1.3	BOOL	SF11
	H4	A 1.4	BOOL	
	Km1	E 1.7	BOOL	Alimentation
	M P	A 0.0	BOOL	montée panier
	Man	E 2.3	BOOL	mode manuel
	Marche pret	A 2.0	BOOL	arche pret
	MISE ST	E 2.2	BOOL	
	N COM	M 0.1	BOOL	Niveau commande
	N S	FB 2	FB 2	
	PO sous T	A 1.2	BOOL	P.O sous tension signal H2
	Préchauffage	A 2.1	BOOL	préchauffage
	Production normale	A 2.2	BOOL	Production normale
	S 30	E 2.0	BOOL	Acquit
	S1	E 0.0	BOOL	
	S10	E 2.1	BOOL	
	S2	E 0.4	BOOL	
	S20	E 3.5	BOOL	
	S21	E 3.6	BOOL	
	S22	E 3.1	BOOL	
	S23	E 3.2	BOOL	
	S24	E 3.3	BOOL	
	S25	E 3.4	BOOL	
	S28	E 1.5	BOOL	
	S29	E 2.4	BOOL	
	S3	E 0.5	BOOL	
	S4	E 0.7	BOOL	
	S5	E 1.1	BOOL	
	S6	E 0.3	BOOL	

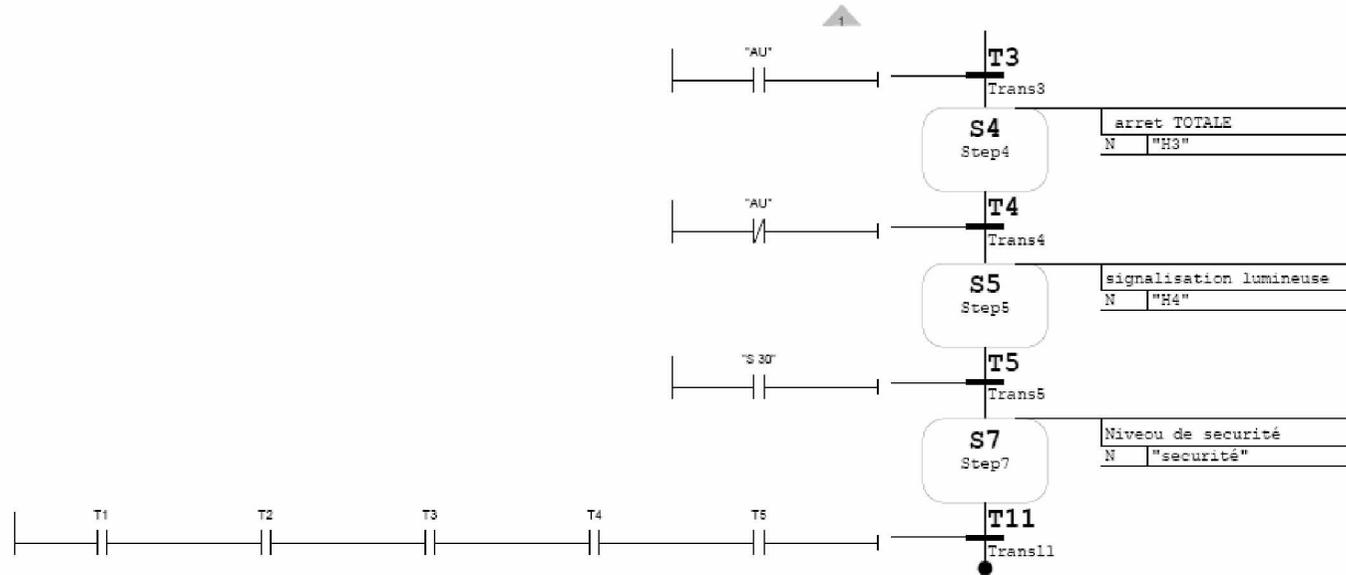
Etat	Mnémonique	Opérande	Type de données	Commentaire
	S7	E 0.1	BOOL	
	securité	M 0.0	BOOL	
	SF10	E 1.2	BOOL	
	SF11	E 1.3	BOOL	
	SF8	E 0.6	BOOL	
	SF9	E 1.0	BOOL	
	signale vers	A 0.1	BOOL	démarrage cycle
	TIME_TCK	SFC 64	SFC 64	Read the System Time
	TRA_GAU	A 0.6	BOOL	
	WR_USMSG	SFC 52	SFC 52	Write a User-Defined Diagnostic Event to the Diagnostic Buffer

Annexe C :

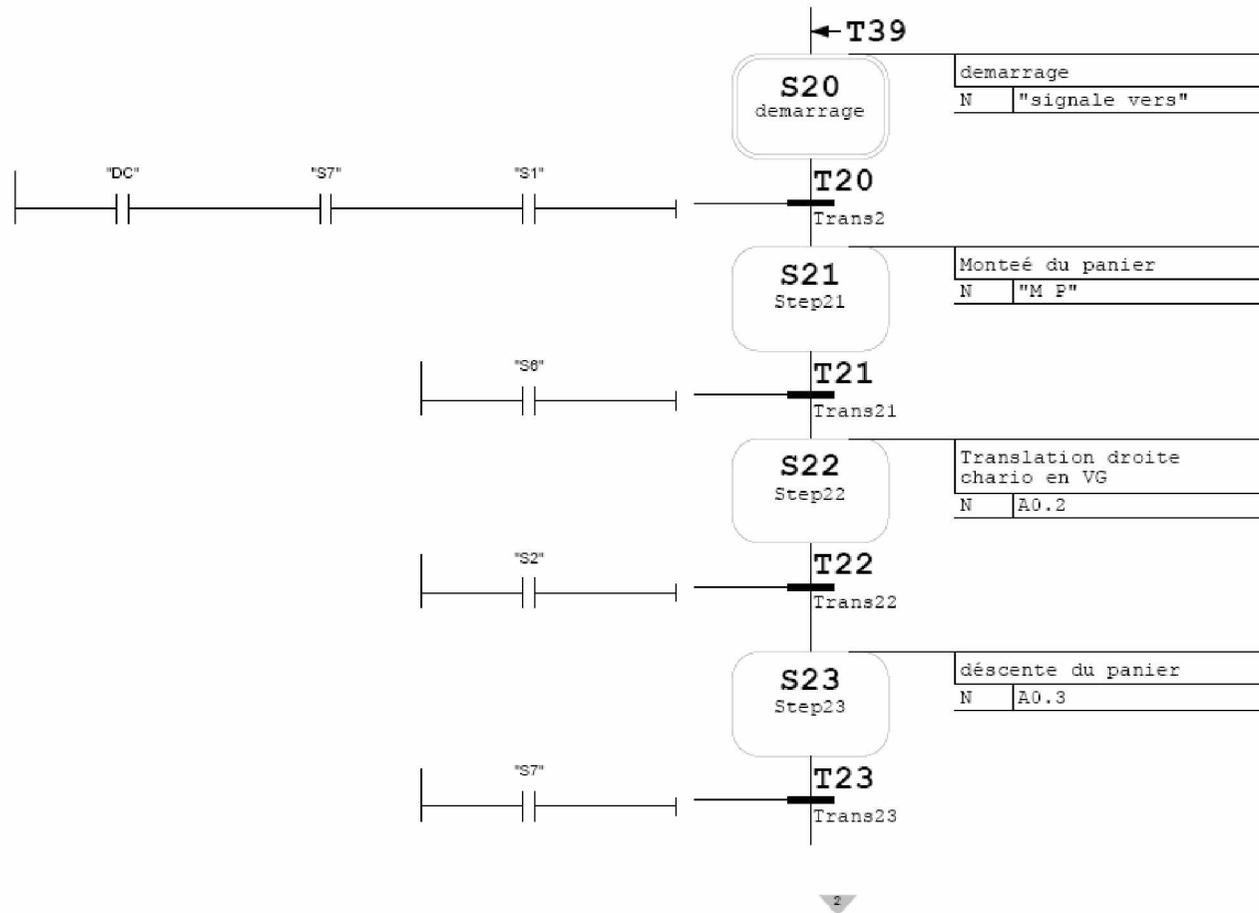
PROGRAMMATION DE LA CHAINE DE TRAITEMENT DE SURFACE

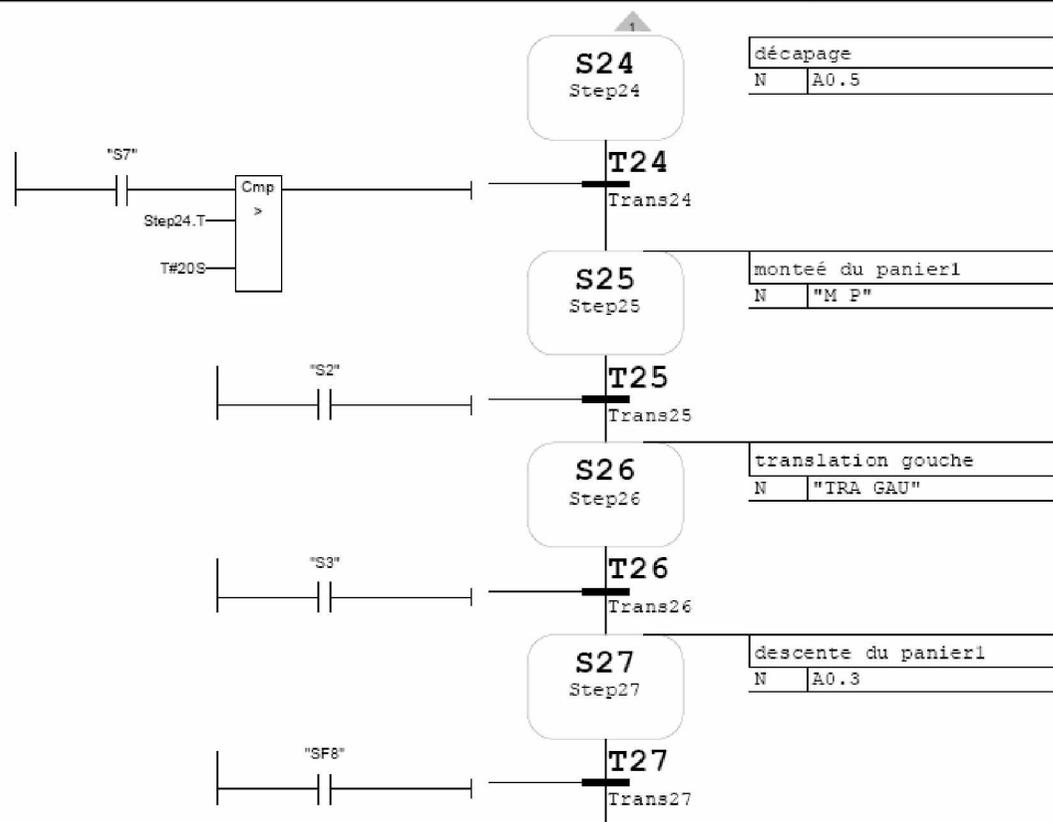
NIVEAU DE SECURITE

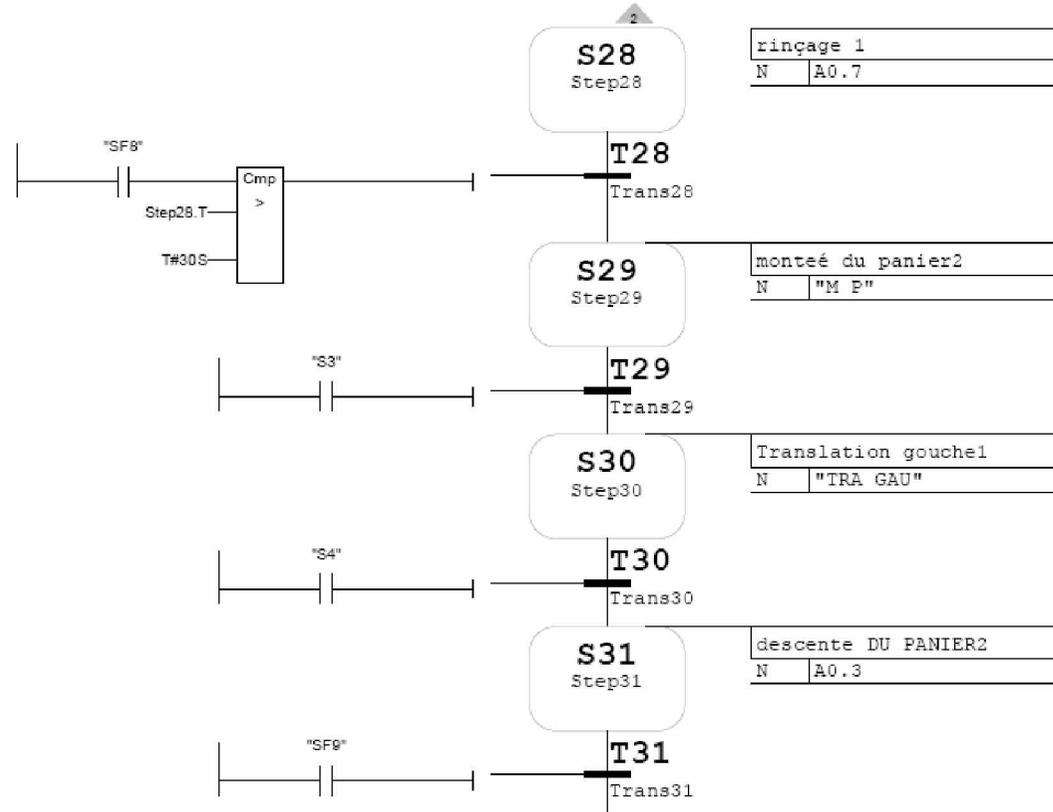




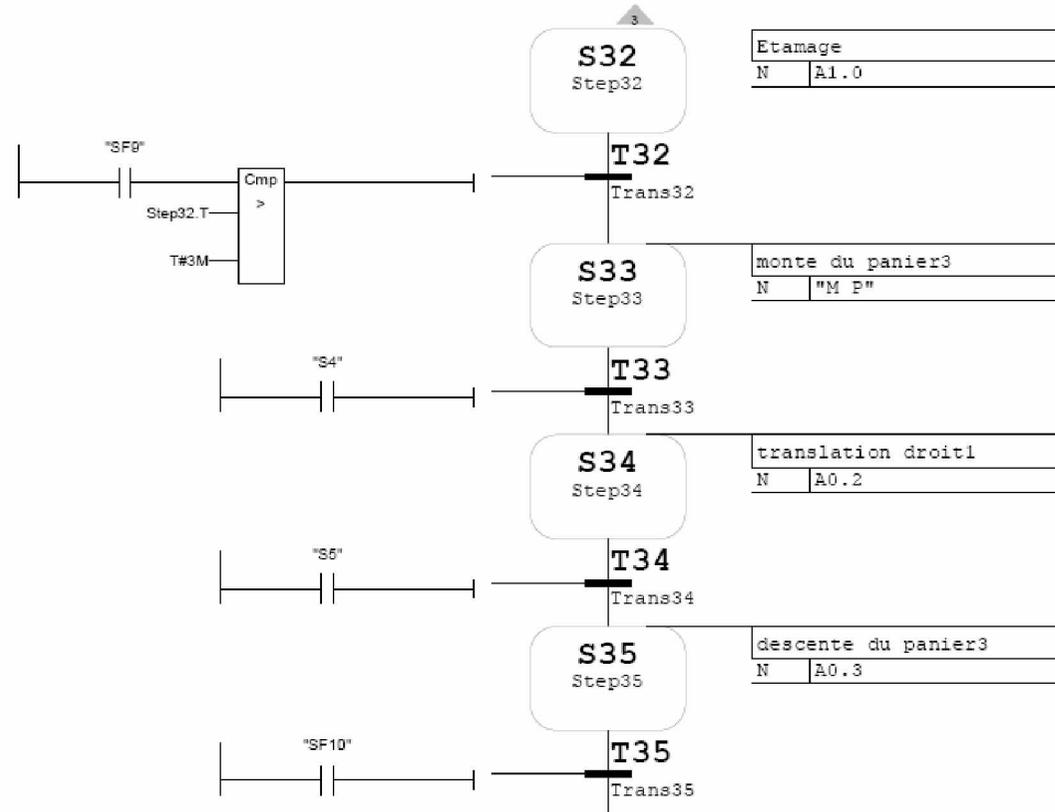
PRODUCTION NORMALE

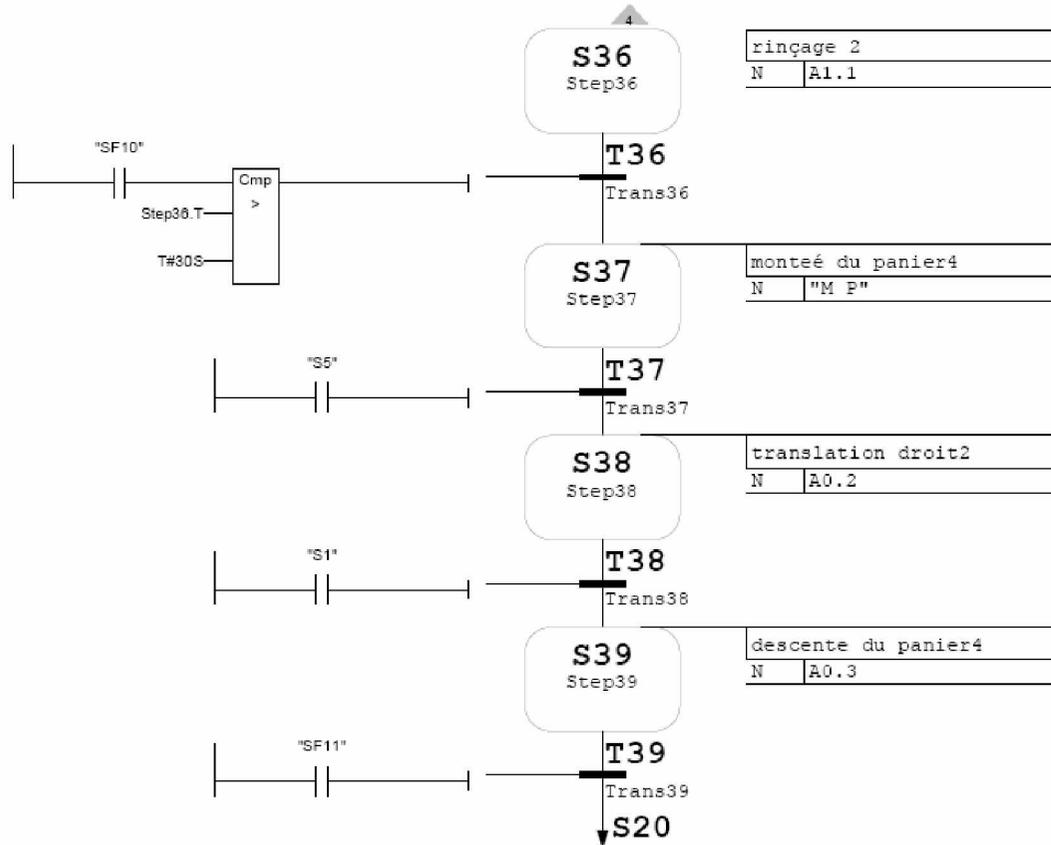




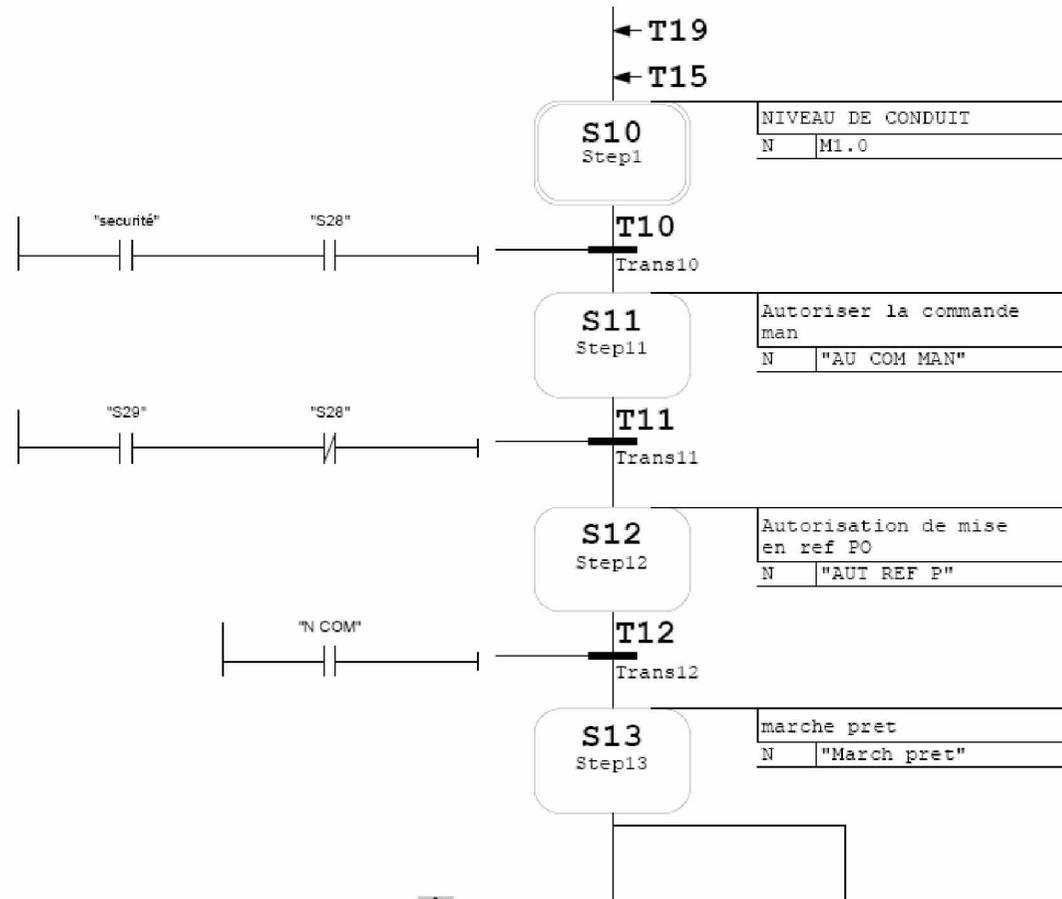


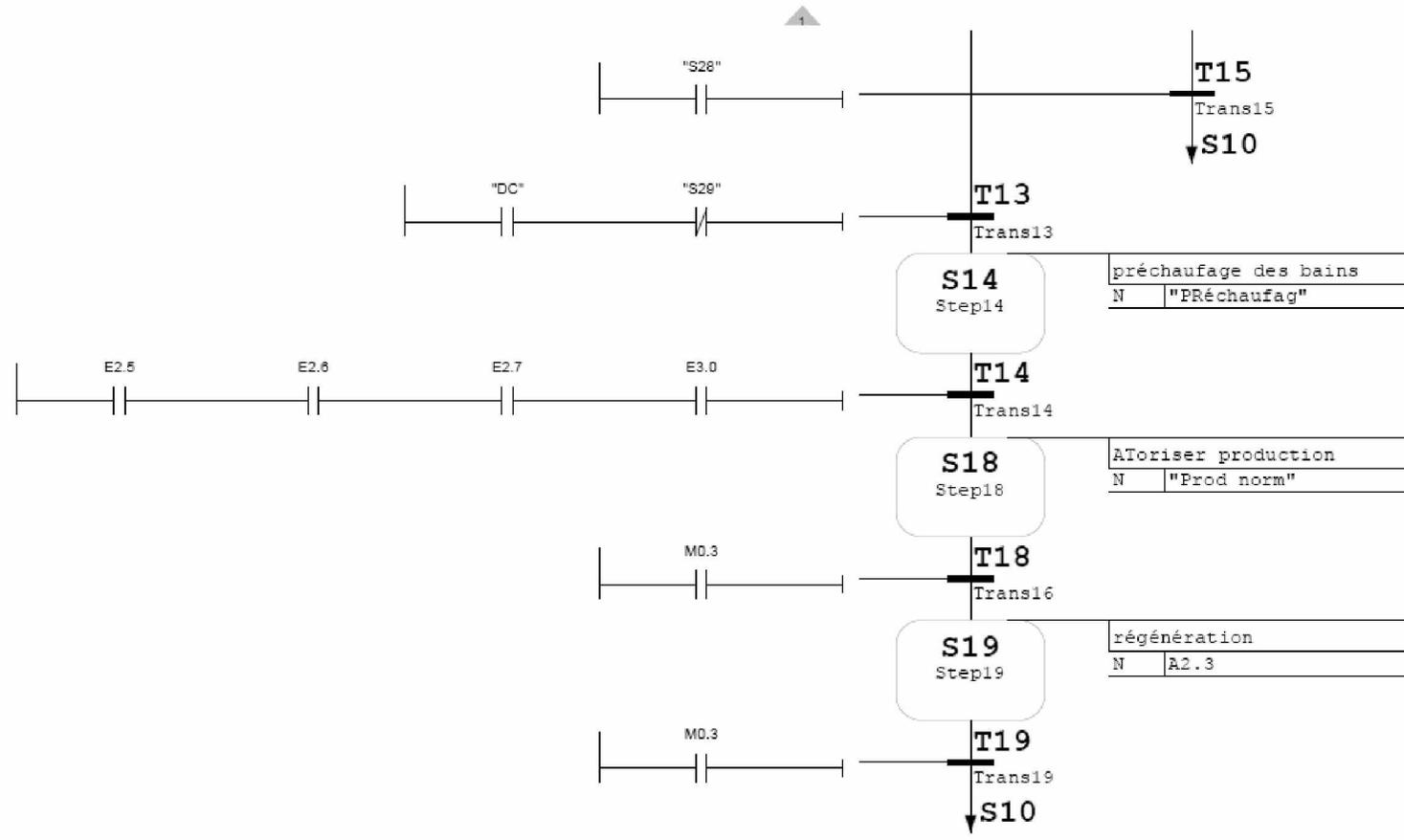
4



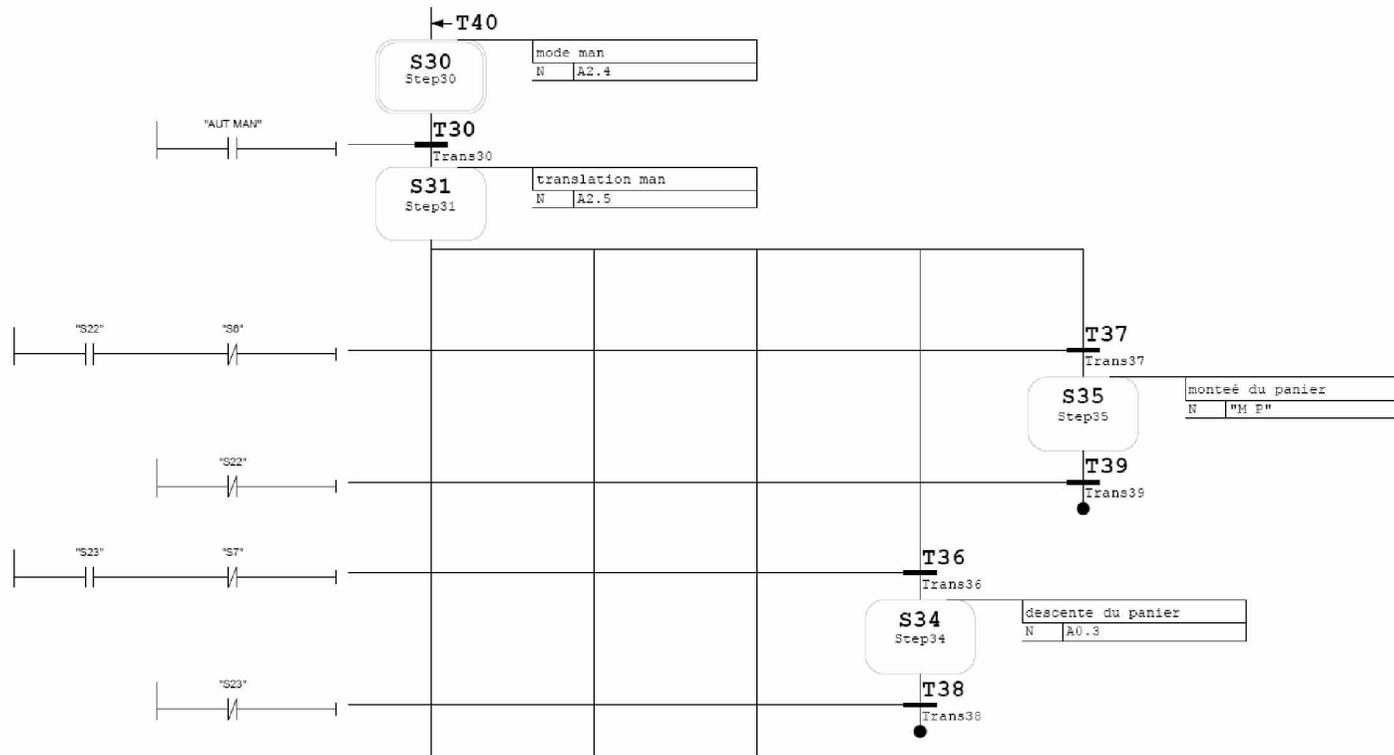


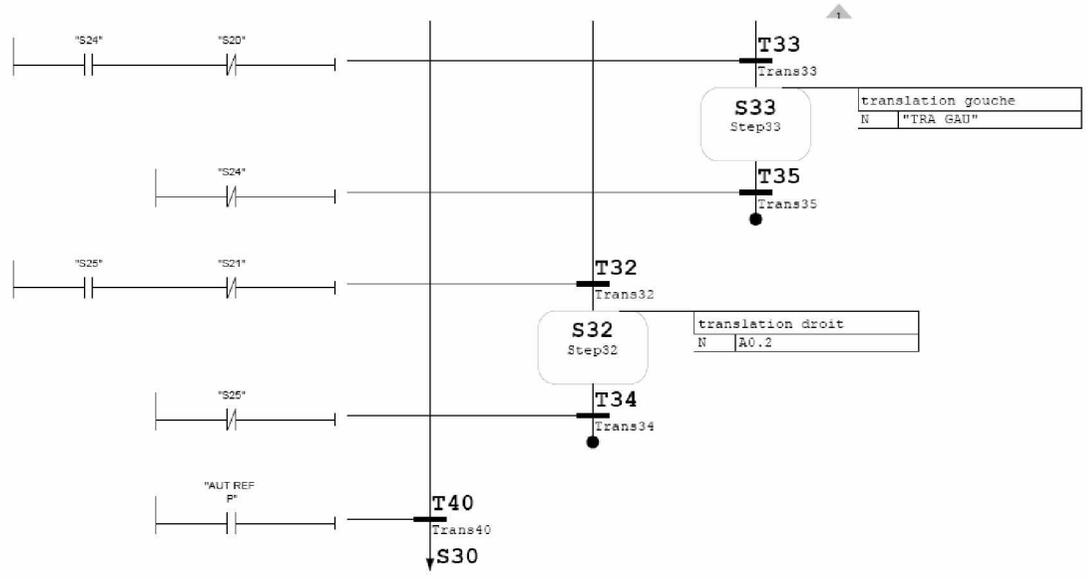
Niveau conduit



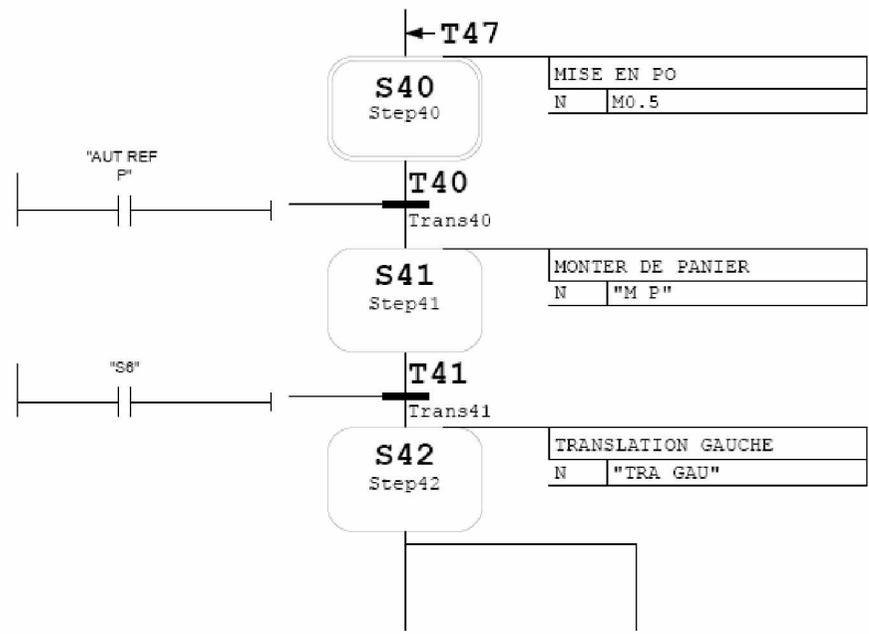


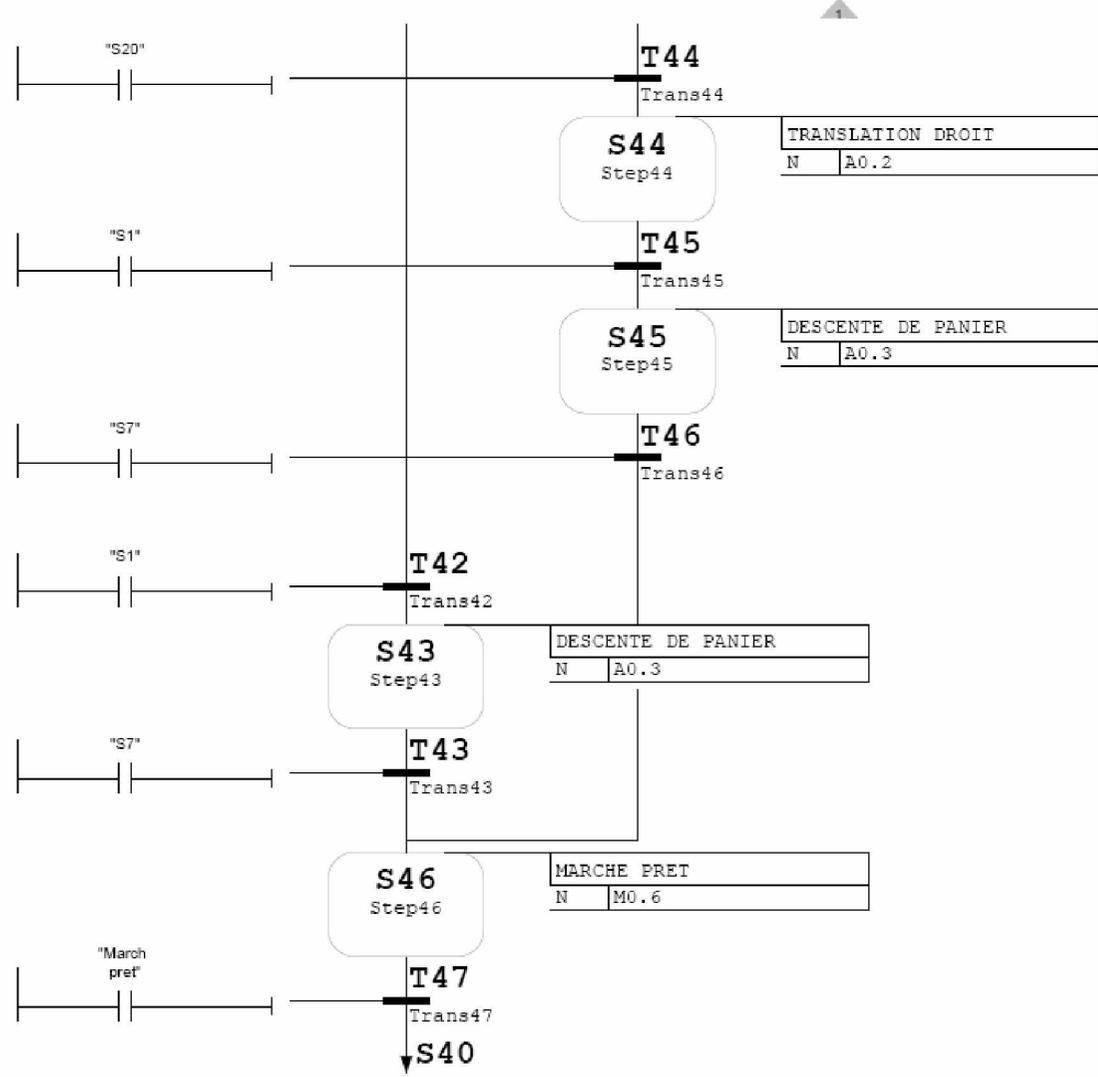
commande manuelle



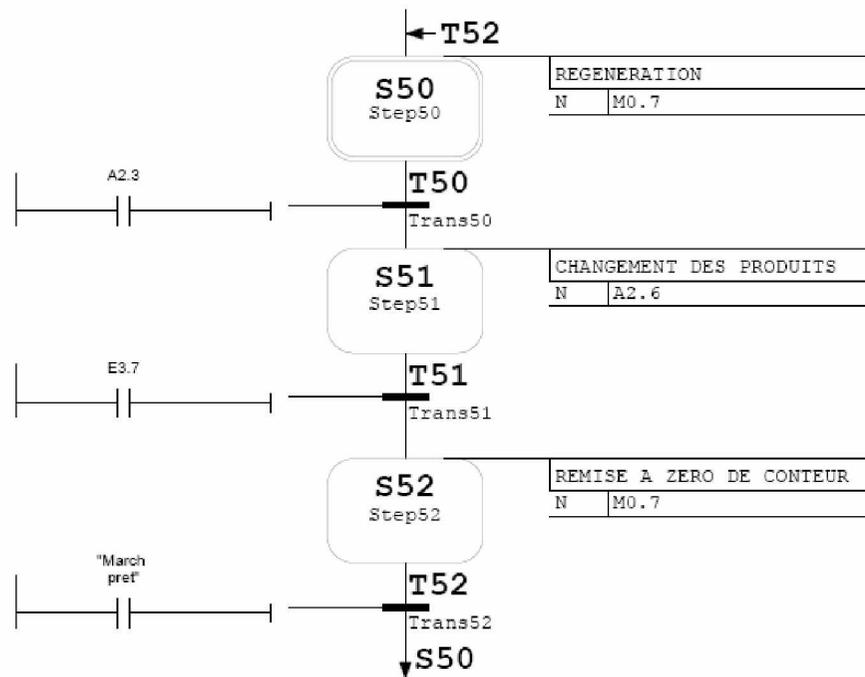


MISE EN PO





REGENERATION



RESUME

Le traitement de surface est un procédé de fabrication qui consiste à déposer une mince couche de métal à la surface d'une pièce de métal différent pour le protéger lui conférer certaines propriétés mécanique.

L'objectif principal est l'exploitation de la chaîne de galvanoplastie dans de bonnes conditions de production et d'optimisation. Il s'agit de trouver un tracé optimum de cycle de pont assurant le déplacement et de conduite optimale de gestion.

Par conséquent, l'élaboration d'un logiciel STEP7 permettent de répondre au cahier des charges.

Un calculateur est nécessaire pour piloter l'ensemble.

MOTS CLES : Automate programmable, Gestion optimale, Grafset, Modélisation, Simulation du procédé.

ABSTRACT

The surface treatment is a process which is deposited a thin layer of metal on the surface of a different piece of metal to protect it with certain mechanical properties.

The main objective is the exploitation of the electroplating line in conditions of production and optimization. We need to find a route optimum cycle bridge providing travel and optimal management.

Therefore, the development of software STEP7 to meet the specification.

A calculator is required to control the whole.

KEYWORDS: Plc, Managed, Grafset, Modeling, Simulation of the process.

ملخص

إن عملية معالجة الأسطح تتمثل في إحداث طبقة رقيقة من المعدن على سطح آخر، فنتشكل قطعة معدنية تحميه و تحمي خصائصه الميكانيكية.

والهدف الرئيسي من هذا البحث هو استغلال سلسلة وحدة معالجة السطح في أحسن الظروف والأحوال الإنتاجية؛ ونحن بحاجة إلى إيجاد الطريق الأسهل و الأمثل لتوفير إدارة أفضل.

لذلك استخدمنا أحد أحدث طرق البرمجيات الصناعية "STEP7"، لتلبية المواصفات المذشودة وفق دفتر الشروط؛ وذلك من أجل التحكم الجيد في جميع محتويات الوحدة الإنتاجية.

الكلمات الرئيسية : الحاكمت القابلة للبرمجة، الإدارة المثلى، المتامن، النمذجة، المحاكاة.