

وزارة التعليم العالي والبحث العلمي

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITE FERHAT ABBAS -SETIF

Faculté de Technologie

Département d'Electrotechnique



جامعة فرحات عباس سطيف

كلية التكنولوجيا

قسم الإلكترونيات

Mémoire de Master

No. Réf. :/...../2012

Présenté au Département d'Electrotechnique

Domaine : Sciences et Technologie
Filière : Automatique
Spécialité : Commande des processus industriels

Réalisé par :

M. SELOUANA Mounir

Thème

Automatisation du groupe de remplissage d'une chaine de production des Boissons gazeuses

Soutenu le/...../..... devant la commission d'examen composée de :

M. KHABER Farid	MCA à l'Université de Sétif	Président
M. BELKHIR Kamel Salim	MAA à l'Université de Sétif	Directeur du Mémoire
Mme. LATRECHE Samia	MAA à l'Université de Sétif	Examineur
Mme. NOUAR Embarka	MAA à l'Université de Sétif	Examineur

Dédicaces

A l'étoile de ma vie celle que je vois jour et nuit, à celle qui a su me donner amour et joie de vivre, symbole de patience et de tendresse celle qui a été toujours là pour moi, c'est grâce à elle que je suis ce que je suis, ma mère que j'aime.

A l'homme de courage et de force, à celui qui a toujours été présent, qui m'a appris les vraies valeurs de la vie, à celui qui m'a soutenu en toutes circonstances, mon père que j'aime.

*A celui qui a consacré sa vie pour m'assurer une éducation rime avec rigueur et travail, à celui qui m'a élevé avec amour et tendresse, à celui qui a toujours cru en moi, aux sacrifices qu'il a faits pour sa famille, à mon chère défunte grand-père
Ismail.*

A ceux qui m'ont donné joie et bonheur, amour et chaleur, mes frères et mes sœurs: Nawal et son mari Abd El-hak et leur adorable fille Asma, Sabah et son mari Bilal, Hanane, Hamza et Aymen.

*A ceux que j'ai eu la chance de connaître, dans les meilleurs et pires moments de ma vie, à mes amis les plus fidèles : Amine, Raouf, Samir, Hichem, Moustafa,
Nouri et djalal.*

A tous ceux que j'aime, à tous ceux qui m'aiment,

Je dédie ce modeste travail.

SELOUANA Mounir

Remerciements

Je remercie d'abord mon Dieu de m'avoir protégé, guidé et donné ce savoir depuis mes premiers mots, jusqu'à l'obtention de mon diplôme, je le prie de continuer à nous aider de plus en plus jusqu'à la mort.

J'adresse mes respects à mon encadreur Mr. BELKHIR Kamel Salim de m'avoir suivi et guidé durant le développement de ce projet.

Je remercie aussi M. GUEZATI Mabrouk et tous les responsables de l'entreprise S.P.A MAMI.

Je souhaite également remercier tous les membres du jury de soutenance : Mr. KHABER farid pour sa présidence, Mme. NOUAR Embarka et Mme. LATRECHE Samia d'avoir accepté d'examiner mon travail.

En fin, Je remercie tous ceux qui, de près ou de loin, ont contribué à la finalisation de ce travail.

SELOUANA Mounir

Sommaire

INTRODUCTION GENERALE	1
<i>Chapitre 01</i>	3
TECHNOLOGIES DES AUTOMATES.....	3
1.1. Introduction	3
1.2. Les systèmes automatisés	3
1.2.1. Description des différentes parties des systèmes automatisés.....	3
1.3. L'automate programmable industriel	5
1.3.1. Les différentes natures d'information traitée par automate	5
1.3.2. Structure interne d'un automate	5
1.3.3. Structure externe d'un automate	8
1.3.4. Programmation d'un automate	9
1.3.5. Domaines d'applications des automates	9
1.3.6. Critères de choix d'un automate.....	9
1.3.7. Condition d'emploi.....	10
1.4. Présentation de la gamme SIMATIC de SIEMENS.....	10
1.4.1. Les différentes variantes dans la gamme SIMATIC S7	10
1.4.2. Présentation des spécifications de l'API S7-300.....	12
1.5. Conclusion.....	13
<i>Chapitre 02</i>	14
DESCRIPTION DE LA MACHINE	14
2.1. Introduction	14
2.2. Présentation du groupe MAMI.....	14
2.2.1. Historique	14
2.2.2. Type des produits disponible.....	14
2.3. La chaîne de production des boissons gazeuses	16
2.4. Description Générale de la machine.....	16
2.5. Caractéristiques techniques de la machine	17
2.6. Les différents constituants de la machine	18
2.6.1. Le réservoir de remplissage.....	19
2.6.2. La trémie	22
2.6.3. L'élévateur des capsules.....	22
2.6.4. La capsuleuse	23

2.7. Cahier de charges	23
2.8. Procédure de remplissage.....	25
2.8.1. Phase 1 : Elévation bouteille.....	26
2.8.2. Phase 2 : Pressurisation de la bouteille.....	26
2.8.3. Phase 3 : Remplissage.....	26
2.8.4. Phase 4 : Fermeture	26
2.8.5. Phase 5 : Décompression.....	27
2.9. Problématique	27
2.10. Conclusion.....	27
 <i>Chapitre 03</i>	 28
LA CREATION DU PROJET, LA SIMULATION ET LA SUPERVISION.....	28
3.1. Introduction	28
3.2. Aperçu du progiciel STEP 7	28
3.3. Structuration d'un programme STEP7.....	29
3.3.1. Programmation linéaire.....	29
3.3.2. Programmation Structurée	29
3.4. Langages de programmation de STEP 7	29
3.4.1. Le schéma logique (LOG).....	29
3.4.2. Le schéma à contacts (CONT).....	29
3.4.3. Liste d'instruction (LIST)	30
3.5. SIMATIC Manager.....	30
3.6. Hiérarchisation dans un projet.....	31
3.6.1. Objet station	31
3.6.2. Objet modules programmables.....	32
3.6.3. Objet programme S7.....	32
3.6.4. Objet dossier Blocs	34
3.7. Création d'un projet S7 pour le groupe de remplissage	36
3.8. La configuration matérielle	37
3.9. Création de la table des mnémoniques	38
3.10. Procédure de la programmation	39
3.11. Structure du programme	40
3.12. Simulation du programme avec S7-PLCSIM.....	41
3.12.1. Présentation du S7-PLCSIM	41
3.12.2. Ouverture du S7-PLCSIM.....	41
3.12.3. Chargement du projet STEP 7	42
3.12.4. Exécution du programme.....	43
3.12.5. Utilisation des tables de variables pour visualiser ou forcer des données	44

3.13. Exemple de programmation.....	45
3.13.1. Gestion des vérins.....	45
3.13.2. L'élévateur des capsules.....	47
3.13.3. Le moteur secondaire.....	48
3.13.4. Exemple de simulation.....	49
3.14. La supervision.....	53
3.14.1. Description du logiciel SIMATIC WinCC.....	53
3.14.2. Pour quoi le SIMATIC WinCC.....	53
3.14.3. L'explorateur de WinCC.....	54
3.14.4. La configuration de la communication réseau.....	55
3.14.5. L'éditeur graphique.....	55
3.14.6. Lancement de la supervision.....	56
3.14.7. Supervision du groupe en arrêt.....	57
3.14.8. Supervision du groupe en marche.....	57
3.14.9. Supervision du groupe avec défaut.....	59
3.15. Conclusion.....	59
CONCLUSIONS GENERALES ET PERSPECTIVES.....	60

Liste des figures

Figure 1.1. Schéma générale d'un système automatisé.....	4
Figure 1.2. Nature des informations traitées par l'API.....	5
Figure 1.3. Schéma générale d'un API.....	6
Figure 1.4. Automate compact (Allen-bradley).....	8
Figure 1.5. Automate modulaire (Modicon).....	8
Figure 1.6. Siemens S7-200.....	10
Figure 1.7. Siemens S7-300.....	11
Figure 1.8. Siemens S7-400.....	11
Figure 1.9. Siemens S7-400H.....	11
Figure 1.10. Siemens S7-1200.....	12
Figure 1.11. Modules d'un API S7-300.....	12
Figure 2.1. Boissons gazeuses MAMI (en PET et en verre).....	15
Figure 2.2. Boissons plates MAMI (en PET et en verre).....	15
Figure 2.3. Boissons aux jus de fruits (en PET).....	15
Figure 2.4. Schéma générale de la ligne de production.....	16
Figure 2.5. La remplisseuse MONOBLOC.....	17
Figure 2.6. Vue face de la remplisseuse MONOBLOC.....	18
Figure 2.7. Vue de haut de la remplisseuse MONOBLOC.....	18
Figure 2.8. Le réservoir de remplissage.....	20
Figure 2.9. Schéma des soupapes de remplissage.....	21
Figure 2.10. La trémie.....	22
Figure 2.11. L'élévateur des capsules.....	23
Figure 2.12. La capsuleuse.....	23
Figure 2.13. Capteur photocellule d'accumulation.....	24
Figure 2.14. Capteur de sécurité d'entrée(C3).....	24
Figure 2.15. Mécanisme de nettoyage de la canule.....	25
Figure 2.16. Compteur des bouteilles remplies.....	25
Figure 2.17. Came de fermeture des robinets.....	26
Figure 2.18. Décompression du reste d'aire.....	27
Figure 3.1. Mode de représentation des langages basiques de programmation STEP7.....	30
Figure 3.2. La fenêtre SIMATIC MANAGER.....	31
Figure 3.3. Création du projet.....	36

Figure 3.4. Choix de la station	37
Figure 3.5. La configuration matérielle.	38
Figure 3.6. Editeur de mnémonique.....	38
Figure 3.7. Procédure générale de programmation.....	39
Figure 3.8. Insertion de blocs de programmation.....	40
Figure 3.9. Structure du programme	41
Figure 3.10. Le simulateur PLCSIM	42
Figure 3.11. Choix du cycle de l'exécution du programme	43
Figure 3.12. Les états du CPU dans le PLCSIM.....	43
Figure 3.13. Table des variables.....	45
Figure 3.14. Gestion de marche/arrêt du vérin d'ouverture des robinets (V1) en CONT.....	46
Figure 3.15. Gestion de marche/arrêt du vérin de nettoyage des canules (V2) en CONT	46
Figure 3.16. Gestion de marche/arrêt de l'élévateur des capsules en LIST.....	47
Figure 3.17. Gestion de marche/arrêt du moteur secondaire en LOG.	48
Figure 3.18. Simulation de fonctionnement en mode automatique du moteur secondaire.....	50
Figure 3.19. Simulation d'arrêt du moteur secondaire	51
Figure 3.20. Simulation de fonctionnement en mode manuel du moteur secondaire.....	52
Figure 3.21. Diagnostic des APIs avec WinCC et les outils de programmation Step 7	53
Figure 3.22. Interface de dialogue WinCC	54
Figure 3.23. Communication réseaux (Net Pro).....	55
Figure 3.24. L'éditeur graphique	56
Figure 3.25. Supervision du groupe en arrêt	57
Figure 3.26. Menu principal de supervision	57
Figure 3.27. Supervision d'accumulation des capsules	58
Figure 3.28. Supervision d'accumulation des bouteilles à l'entrée.....	58
Figure 3.29. Supervision d'un défaut au niveau de M1	59
Figure 3.30. Liste des alarmes.....	59

Liste des tableaux

Tableau 2.1. Caractéristiques techniques de la machine.....	17
Tableau 3.1. Types de variables manipulées dans le STEP 7	33

Listes des Acronymes

API	Automate Programmable Industriel.
CEI	Commission Electrotechnique Internationale.
CP	Communication Processor.
CPU	Central Process Unit.
EEPROM	Electrically erasable programmable read only memory.
FBD	Function Bloc Diagram.
FM	Function Modul.
HMI	Human Machine Interface.
HW Config	Hardware Configuration.
IL	Instruction List.
LD	Ladder Diagram
MPI	Multi Point Interface.
MRES	Memory Reset.
PC	Personal Computer.
PET	Polyéthylène Téréphtalate.
PLC	Programmable Logic Controller.
PLCSIM	Programmable Logic Controller Simulator.
PROFIBUS	Process Field Bus.
PROM	Programmable Read Only Memory.
PS	Power Supplie.
RAM	Random Access Memory.
ROM	Read Only Memory.
SFC	Sequential Function Chart.
SM	Signal Modul.
SNC	Société au Nom Collectif
SPA	Société Par Action.
ST	Structured Text.
TOR	Tout Ou Rien.
UAL	Unité Arithmétique et Logique.
VAT	Variable Table.
WinCC	Windows Contrôle Center.

INTRODUCTION GENERALE

Depuis la nuit des temps l'homme a su développer ses techniques de travail et commander ses machines, pour atteindre ce but la commande câblée à base de relais interconnectés par des fils à l'intérieur d'un tableau de bord a été apparue, mais cette commande était très inflexible à suivre l'évolution et le développement massif de l'industrie ce qui a exigé une augmentation de productivité.

Une des données principales de l'évolution de la production industrielle est le développement des systèmes automatisés, et par conséquent l'augmentation de la productivité. Pour fonctionner automatiquement, les systèmes industriels et les moyens de production font appel à la commande programmée à base de l'automate programmable qui permet une grande souplesse d'exploitation.

Durant ces dernières années, quelques entreprises algériennes ont pris un peu d'avance en installant des systèmes automatisés très performants, mais n'arrivent souvent pas à suivre l'évolution technologique, celle-ci est très coûteuse et évolue rapidement. Les constructeurs des machines sont toujours confrontés au défi de devoir proposer des machines plus flexibles et productives à des prix plus compétitifs, tout en assurant la maintenance ainsi qu'un bon suivi de progrès industriels.

En effet, pour accroître la production, l'entreprise MAMI doit avoir un équipement de qualité répondant à ses besoins ; la commande par logique câblée a déjà montré ses limites de fiabilité technique du point de vue panne, vitesse de traitement et défaut. Alors, pour une grande productivité avec une grande propriété aux brèves durées, cette commande de la logique câblée a été remplacée par l'automatisation qui utilise le plus souvent des automates programmables industriels qui sont apparues à la fin des années soixante, à la demande de l'industrie automobile Américaine, qui réclamait plus d'adaptabilité de leurs systèmes de commande.

Pour ce qui nous concerne, dans le cadre de notre mémoire qui censure la fin du 2^{ème} cycle d'étude universitaire et en l'occurrence pour l'obtention de grade master en automatique nous avons eu une occasion qui nous a été donnée afin d'approfondir notre connaissance en automatique pour automatiser un groupe de remplissage des boissons gazeuses au sein du SPA MAMI situé à la zone industriel de Sétif, en utilisant un automate programmable de type SIEMENS S7-300, lié avec les progiciels STEP7 et WinCC qui nous permettent de piloter et de superviser ce groupe.

Le présent travail s'articule autour de trois chapitres, dans le premier chapitre on présentera une étude générale sur les systèmes automatisés, et le moyen qui assure la commande de ces systèmes qui est l'automate programmable industriel ainsi que la gamme choisie pour effectuer l'automatisation (I'S7-300). Le deuxième chapitre sera consacré à la description du processus à automatiser ; ses différents constituants, ses différentes phases de production et la présentation du cahier de charge de son fonctionnement séquentiel. Le troisième chapitre, fera ressortir le software de la commande du processus à automatiser, édité dans le langage de programmation STEP 7, puis son déroulement sera supervisé par le progiciel WinCC. Enfin, ce modeste travail sera clôturé par une conclusion générale.

Chapitre 01

TECHNOLOGIES DES AUTOMATES

1.1. Introduction

Depuis toujours l'homme est en quête de bien être, cette réflexion qui rejoint la notion de besoin peut paraître bien éloignée d'un cours de sciences industrielles, pourtant c'est la base de l'évolution des sciences en général, et de l'automatisation en particulier. L'homme a commencé par penser, concevoir et réaliser. Lorsqu'il a fallu multiplier le nombre d'objets fabriqués, produire en plus grand nombre, l'automatisation des tâches est alors apparue pour remplacer l'homme dans des actions pénibles, délicates ou répétitives.

En effet, dans ce premier chapitre, on va vous donner quelques notions de base sur les systèmes automatisés, ensuite on va présenter le moyen fondamentale pour la commande de ces systèmes qui est l'automate programmable industriel.

1.2. Les systèmes automatisés

Un système automatisé est un système réalisant des opérations pour lequel l'homme n'intervient que dans la programmation du système et dans son réglage. Les buts d'un système automatisé sont de réaliser des tâches complexes ou dangereuses pour l'homme, effectuer des tâches pénibles ou répétitives ou encore gagner en efficacité et en précision [1].

1.2.1. Description des différentes parties des systèmes automatisés [2]

Chaque système automatisé est généralement constitué de trois parties essentielles comme le montre le schéma suivant :

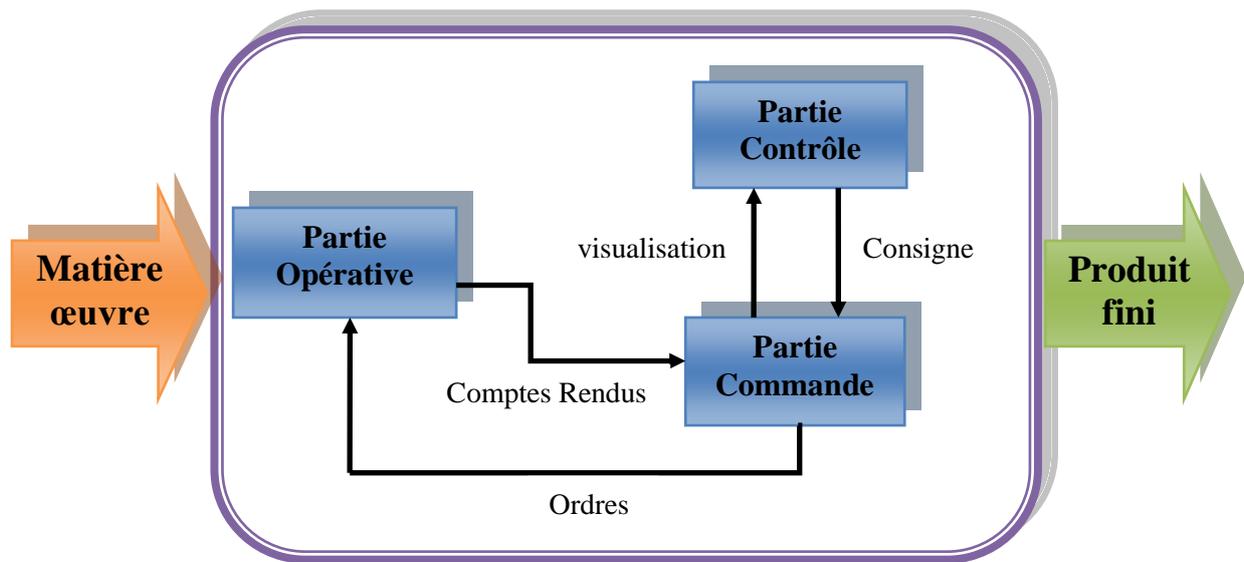


Figure 1.1. Schéma générale d'un système automatisé.

1.2.1.1. La partie opérative

C'est la partie visible du système. Elle comporte les éléments du processus, qui sont :

- Les pré-actionneurs (distributeurs, contacteurs,...) qui reçoivent des ordres de la partie commande.
- Les actionneurs (vérins, moteurs, vannes, ...) qui ont pour rôle d'exécuter ces ordres. Ils transforment l'énergie pneumatique, hydraulique ou électrique en énergie mécanique (phénomène physique).
- Les capteurs qui informent la partie commande de l'exécution du travail.

1.2.1.2. La partie commande

Ce secteur de l'automatisme gère selon une suite logique le déroulement ordonné des opérations à réaliser. Il reçoit des informations en provenance des capteurs de la partie opérative, et les restitue vers cette même partie opérative en direction des pré-actionneurs et actionneurs.

1.2.1.3. La partie contrôle (supervision)

Elle regroupe les différentes commandes nécessaires au bon fonctionnement du processus comme titre d'exemple : marche/arrêt, arrêt d'urgence, marche automatique, marche manuel etc... . Sa complexité dépend de l'importance du système.

1.3. L'automate programmable industriel

L'automate programmable industriel API (*Programmable Logic Controller PLC*) est un appareil électronique programmable, adapté à l'environnement industriel qui réalise des fonctions d'automatisme pour assurer la commande de pré-actionneurs et d'actionneurs à partir des informations logiques, analogiques ou numériques via les capteurs [3].

1.3.1. Les différentes natures d'information traitée par automate

Généralement, il existe trois types d'information qui peuvent être traitées par un automate :

- **Information analogique** : est une information qui peut prendre toutes les valeurs possibles dans un intervalle donné. Les grandeurs physiques, comme la température, la vitesse, la pression, la tension ..., sont des informations analogiques. Ce type d'information est généralement délivré par un capteur.
- **Information TOR** : Toute information peut être représentée par une proposition logique qui ne peut être que vraie ou fausse, c'est le tout ou rien. Ce type d'information est délivré par un détecteur, un bouton poussoir ...
- **Information numérique** : Information généralement issue d'un traitement d'une information analogique (échantillonnage, codage...) sous la forme d'un mot binaire, constitué de plusieurs bits. Ce type d'information est délivré par un ordinateur ou un module intelligent.

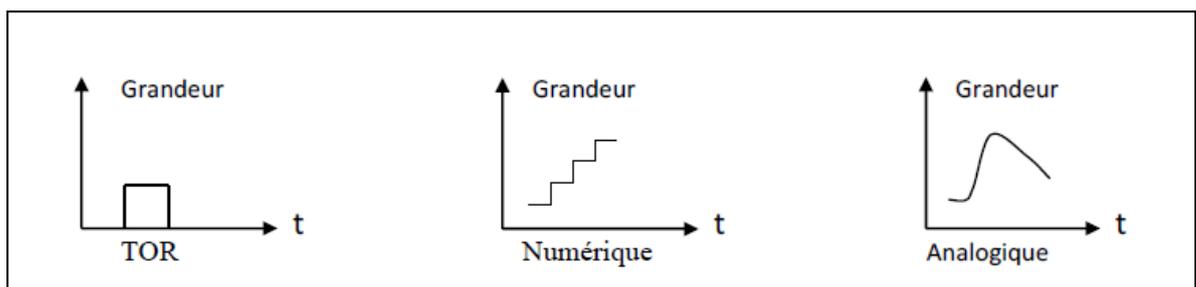


Figure 1.2. Nature des informations traitées par l'API.

1.3.2. Structure interne d'un automate

La structure matérielle interne d'un API obéit au schéma donné sur la figure suivante :

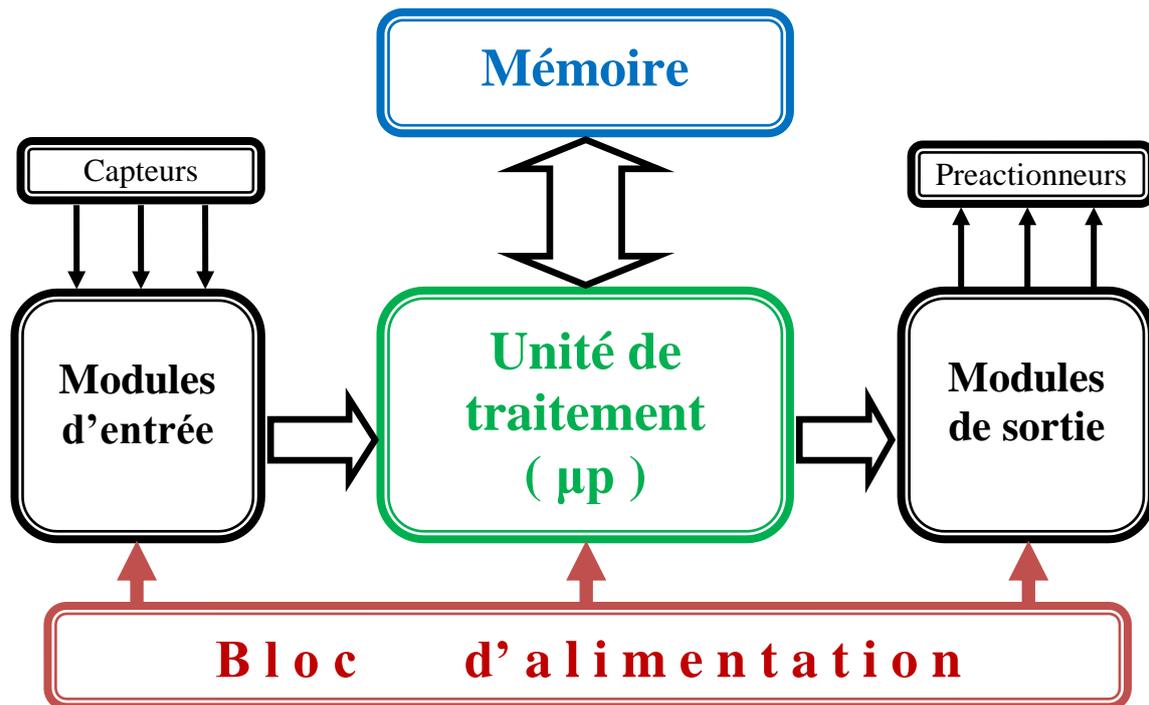


Figure 1.3. Schéma générale d'un API.

Donc, chaque API se compose généralement de cinq blocs, nous analyserons successivement chacun de ces blocs :

1.3.2.1. Bloc d'alimentation

Il assure la distribution d'énergie aux différents modules. La plus part des automates actuels sont alimentées par un réseau monophasé 220V-50 Hz mais d'autres alimentations sont possibles (110V ...etc.) et délivrant une tension dépend du module de sortie utilisée (24V, 15V, 12V...).

1.3.2.2. L'unité de traitement (Processeur)

Dans un système de traitement d'information, le processeur est l'unité fonctionnelle capable d'interpréter et d'exécuter les instructions du programme. Dans un API le processeur gère l'ensemble des échanges informationnels en assurant :

- La lecture des informations d'entrée.
- L'exécution des instructions du programme mis en mémoire.
- L'envoi des ordres à la sortie.

Suivant sa puissance de traitement le processeur peut exécuter des instructions relatives à des opérations :

- Logique : ET, OU, OR ...
- Arithmétique : addition, soustraction, multiplication...
- Temporisation, mémorisations et transfert de mémoire.
- Comptage, décomptage, comparaison, décalage... etc

Fonctionnellement, le processeur comprend :

- Une unité arithmétique et logique (UAL) : traite les opérations arithmétiques et les opérations logiques.
- Un accumulateur : registre de travail dans lequel se range le résultat d'une opération.
- Un registre d'instruction : où sont rangées durant le temps de traitement les instructions à interpréter et à exécuter.
- Un décodeur d'instruction : circuit qui transforme les instructions d'un programme en signaux de commande nécessaire à l'UAL.
- Un compteur du programme : registre qui contient l'adresse de la prochaine instruction à exécutée.

1.3.2.3. Mémoire

La mémoire est l'élément fonctionnel qui peut recevoir, conserver et restituer des données dans une zone où ces données sont écrites, effacées ou lues. Un automate est accompagné par un ou plusieurs types de mémoires qui permettent de stocker :

- Le système d'exploitation dans des **ROM** ou **PROM**.
- Le programme dans des **EEPROM**.
- Les données lors du fonctionnement dans des **RAM**. Cette dernière est généralement secourue par une pile ou une batterie.

1.3.2.4. Les modules d'Entrées / Sorties

Son rôle est l'interfaçage entre l'API et l'extérieur. Le nombre total de modules est évidemment limité pour des raisons physiques (taille du châssis et/ou de l'alimentation électrique). Selon les trois types d'information qui peuvent être traitées par un API, nous distinguons aussi trois modules Entrées / Sorties:

- **Modules TOR** : traite les informations tout ou rien.
- **Modules analogique** : traite les informations analogiques.
- **Modules spécialisées** : traite les informations numériques [4].

1.3.2.5. Bus de communication

Elles permettent la communication de l'ensemble des blocs de l'automate et des éventuelles extensions. Les liaisons s'effectuent :

- Avec l'extérieur par des borniers sur lesquels arrivent des câbles transportant le signal électrique.
- Avec l'intérieur par des bus reliant divers éléments, afin d'échanger les données et les adresses.

1.3.3. Structure externe d'un automate

Les automates peuvent être de type compact ou modulaire :

1.3.3.1. Type compact

Il intègre le processeur, l'alimentation, les entrées et les sorties. Selon les modèles et les fabricants, il pourra réaliser certaines fonctions supplémentaires (comptage rapide, E/S analogiques ...) et recevoir des extensions en nombre limité. Ces automates, de fonctionnement simple, sont généralement destinés à la commande de petits automatismes [5].

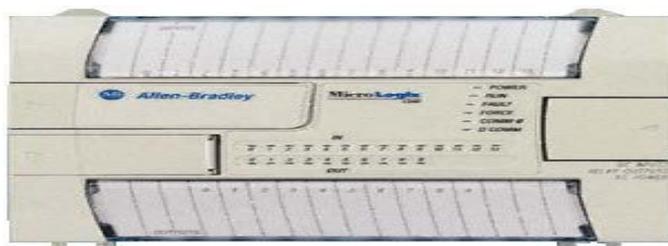


Figure 1.4. Automate compact (Allen-bradley).

1.3.3.2. Type modulaire

Le processeur, l'alimentation et les interfaces d'entrées / sorties résident dans des unités séparées (modules) et sont fixées sur un ou plusieurs racks. Ces automates sont intégrés dans les automatismes complexes où la capacité de traitement et la flexibilité sont nécessaires [5].



Figure 1.5. Automate modulaire (Modicon).

1.3.4. Programmation d'un automate

Chaque automate se programme via une console de programmation propriétaire ou par un ordinateur équipé du logiciel spécifique. La norme IEC 61131-3 a défini cinq langages pour la programmation des APIs [6] :

- **IL** (*Instruction list*) : Langage textuel de même nature que l'assembleur.
- **ST** (*Structured Text*) : Langage informatique de même nature que le Pascal.
- **LD** (*Ladder diagram*) : Langage graphique développé pour les électriciens. Il utilise les symboles tels que : contacts, relais et blocs fonctionnels et s'organise en réseaux.
- **FBD** (*Function Bloc Diagram*) : Langage graphique où des fonctions sont représentées par des rectangles avec les entrées à gauche et les sorties à droite. Les blocs sont programmés (bibliothèque) ou programmables.
- **SFC** (*Sequential Function Chart*) : est un langage graphique basé sur le GRAFCET.

1.3.5. Domaines d'applications des automates

Pour les raisons qui viennent d'être évoquées, les APIs s'adressent à des applications que l'on trouve dans les secteurs industriels comme :

- La commande des machines (convoyage, emballage ...).
- La commande des chaînes de production (automobile, agroalimentaire ...).
- La régulation des processus (métallurgie, chimie ...).
- Il est de plus en plus utilisé dans le domaine du bâtiment pour le contrôle du chauffage, de l'éclairage, de la sécurité ou des alarmes.

1.3.6. Critères de choix d'un automate

Le choix d'un automate programmable est basé sur plusieurs points, nous citons :

- Nombre d'entrées / sorties.
- Type de processeur : la taille mémoire, la vitesse de traitement et les fonctions spéciales offertes par le processeur permettront le choix dans la gamme souvent très étendue.
- Fonctions ou modules spéciaux : certaines cartes permettront de soulager le processeur et devront offrir les caractéristiques souhaitées.
- Fonctions de communication : l'automate doit pouvoir communiquer avec les autres systèmes de commande (API, supervision ...) et offrir des possibilités de communication avec des standards normalisés (Profibus, Ethernet ...).

- Le personnel de maintenance et le logiciel de programmation (achat du logiciel et formation du personnel).

1.3.7. Condition d'emploi

Le fonctionnement d'un API dans un milieu industriel risque d'être perturbé par :

- Des conditions limites d'influences externes (température, humidité, vibration...).
- Des défaillances du réseau de distribution d'énergie.
- Fluctuations des tensions et/ou de fréquence.
- Microcoupure de l'ordre de quelques dixièmes de seconde.
- Parasites industriels.

1.4. Présentation de la gamme SIMATIC de SIEMENS

Siemens reste le seul à proposer une gamme complète de produits pour l'automatisation industrielle, par le biais de sa gamme SIMATIC. L'intégration globale de tout l'environnement d'automatisation est réalisée grâce à :

- Une configuration et une programmation homogène des différentes unités du système.
- Une gestion cohérente des données.
- Une communication globale entre tous les équipements d'automatisme mis en œuvre.

1.4.1. Les différentes variantes dans la gamme SIMATIC S7

Cette gamme d'automates comporte les familles suivantes:

S7-200 : est un micro-automate modulaire pour les applications simples, avec possibilité d'extensions jusqu'à 7 modules, et une mise en réseau par MPI ou PROFIBUS [7].



Figure 1.6. Siemens S7-200.

S7-300 : est un automate modulaire pour les applications d'entrée et de milieu de gamme, avec possibilité d'extensions jusqu'à 32 modules, et une mise en réseau par MPI, PROFIBUS et Ethernet [7].



Figure 1.7. Siemens S7-300.

S7-400 : est un automate de haute performance pour les applications de milieu et haut de gamme, avec possibilité d'extension à plus de 300 modules, et une possibilité de mise en réseau MPI, PROFIBUS ou Ethernet [7].



Figure 1.8. Siemens S7-400.

S7-400H : à disponibilité élevée avec CPU redondantes. Le SIMATIC S7-400H est un automate doté de deux CPU H de même type. En cas de défaut du sous-système maître, il commute sur le sous-système réserve. Il est dédié aux processus à haute disponibilité posant des exigences de Hot-Standby (processus avec des temps de commutation inférieurs à 100 ms) [7].



Figure 1.9. Siemens S7-400H.

S7-1200 : contrôleur modulaire pour les solutions d'automatisation discrètes et autonomes. Le SIMATIC S7-1200 propose une interface PROFINET intégrée, des fonctionnalités technologiques performantes et intégrées ainsi qu'un concept ultra flexible au service d'une communication simple et de solutions efficaces pour la résolution des tâches technologiques et pour des solutions répondant exactement aux exigences spécifiques d'une grande diversité d'applications d'automatisation [7].

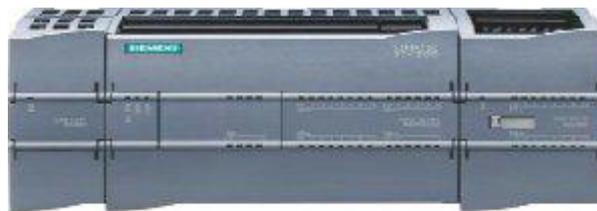


Figure 1.10. Siemens S7-1200.

1.4.2. Présentation des spécifications de l'API S7-300

L'S7-300 est l'automate conçu pour des solutions dédiées au manufacturier et constitue une plate-forme d'automatisation universelle pour des applications avec des architectures centralisées et décentralisées. L'innovation est permanente et se manifeste nouveaux modèles orientés sécurité, motion contrôle ou avec interface Ethernet intégrée. L'S7-300 traite des applications dans l'industrie comme l'automobile, la construction, l'emballage, l'agro-alimentaire, la plasturgie, etc... Il peut également s'intégrer dans des solutions compactes avec HMI ou dans des têtes de station pour traitement décentralisé.

1.4.2.1. Gamme de modules

Le SIMATIC S7-300 est un système d'automatisation modulaire offrant la gamme de modules représentés par la figure ci-dessus :

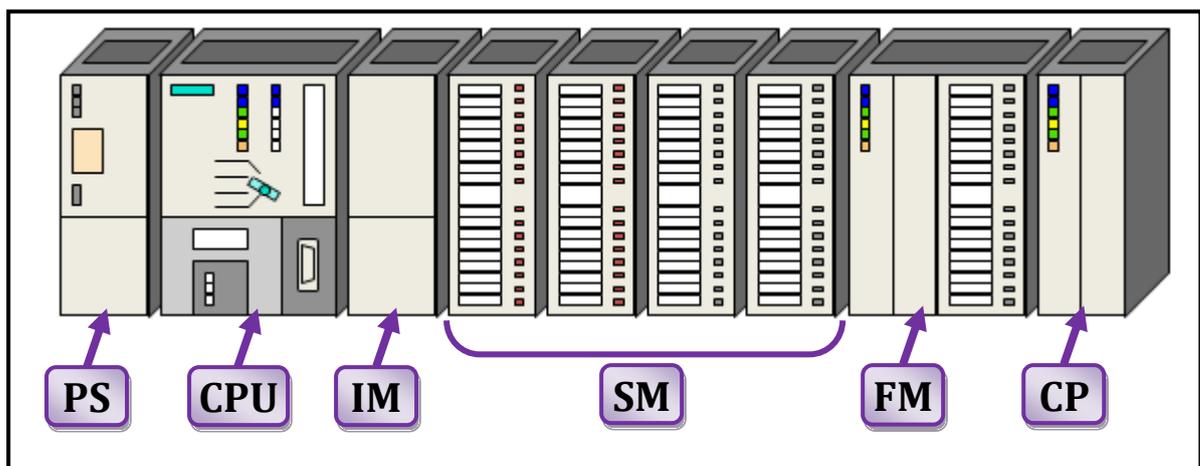


Figure 1.11. Modules d'un API S7-300

PS : (*Power Supplie*) Alimentation en tension.

CPU : (*Central process Unit*) Unités centrales pour le traitement des données.

IM : (*Interface Modul*) Modules de couplage pour raccorder les châssis d'extension.

SM : (*Signal Modul*) Modules de signaux pour les entrées et sorties numériques et analogiques.

FM : (*Function Modul*) Modules de fonction pour les opérations spéciales (régulation...).

CP : (*Communication Processor*) Processeurs de communication pour la connexion au réseau.

1.4.2.2. Extensions possibles

Si une tâche d'automatisation exige plus que 8 modules, le châssis de base du S7-300 peut être complété avec des châssis d'extension. La configuration maximale possible est de 32 modules à raison de 8 par châssis. La communication entre les différents châssis est assurée de façon totalement autonome par des coupleurs IM. Dans les installations de grande envergure, les châssis de base et d'extension peuvent être installés à plus grande distance (jusqu'à 10 m).

1.5. Conclusion

Actuellement, l'automate programmable est un produit industriel largement utilisé dans le domaine d'automatisation des systèmes de productions, il possède plusieurs gammes de CPU et de modules d'E/S ce qui le rend adaptable à plusieurs conditions d'automatisation industrielle. Pour cela on l'adopte comme un choix adéquat dans notre projet.

Chapitre 02

DESCRIPTION DE LA MACHINE

2.1. Introduction

Dans l'industrie, le terme remplissage n'est rien d'autre que l'opération qui consiste, par le biais d'une machine, à conditionner des produits liquides dans des bouteilles spécifiques.

Dans ce chapitre on va voir tous ce qui concerne la remplisseuse MONOBLOC de NOVA pour le remplissage des boissons gazeuses au sein du SPA MAMI, et comment cette opération va se faire, mais avant ça on va vous donner une vue générale sur le groupe MAMI.

2.2. Présentation du groupe MAMI

2.2.1. Historique

- **1937** : Naissance de la société
- **De 1938 à 1947** : Démarrage d'une structure de production manuelle de boissons gazeuses.
- **De 1948 à 1980** : Agrandissement de la structure, développement et production industrielle de boissons gazeuses.
- **De 1981 à 2002** : Création de la SNC MAMI, changement de site, augmentation des capacités de production et poursuite de la croissance de la société
- **De 2003 à nos jours** : Transformation juridique de la société qui devient une société par actions bicéphale (SPA à directoire et à conseil de surveillance). Une nouvelle dynamique s'est installée au sein de la société : lancement des boissons en PET et boissons au jus naturel, augmentation des capacités de production des boissons en bouteille de verre, extension géographique de la distribution au niveau national, exportation... [8]

2.2.2. Type des produits disponible

A base d'eau de forage traitée, de sirop de sucre, d'extraits aromatiques et de divers additifs alimentaires, les boissons MAMI sont fabriquées dans le strict respect de leurs recettes et

formulations originelles conformément aux règles des bonnes pratiques de fabrication et d'hygiène. Il existe trois grandes catégories des boissons disponibles :

2.2.2.1. Boissons gazeuses

Créées en même temps que la société MAMI en 1937 et fabriquées à partir d'arômes de fruits (naturels, identique naturels).



Figure 2.1. Boissons gazeuses MAMI (en PET et en verre).

2.2.2.2. Boissons plates aromatisées

Lancées à la fin de l'année 2001, les boissons plates aromatisées sont fortement appréciées pour leur douceur et leurs parfums fruités.



Figure 2.2. Boissons plates MAMI (en PET et en verre).

2.2.2.3. Boissons aux jus de fruits

Dernière gamme à être produite et mise sur le marché en 2010, elle n'est disponible qu'en PET.



Figure 2.3. Boissons aux jus de fruits (en PET).

2.3. La chaîne de production des boissons gazeuses

Cette chaîne se constitue de plusieurs machines assurant des tâches précises dans le but d'avoir un produit fini et prêt à être vendu. Le schéma suivant représente l'enchaînement et la disposition de ces machines.

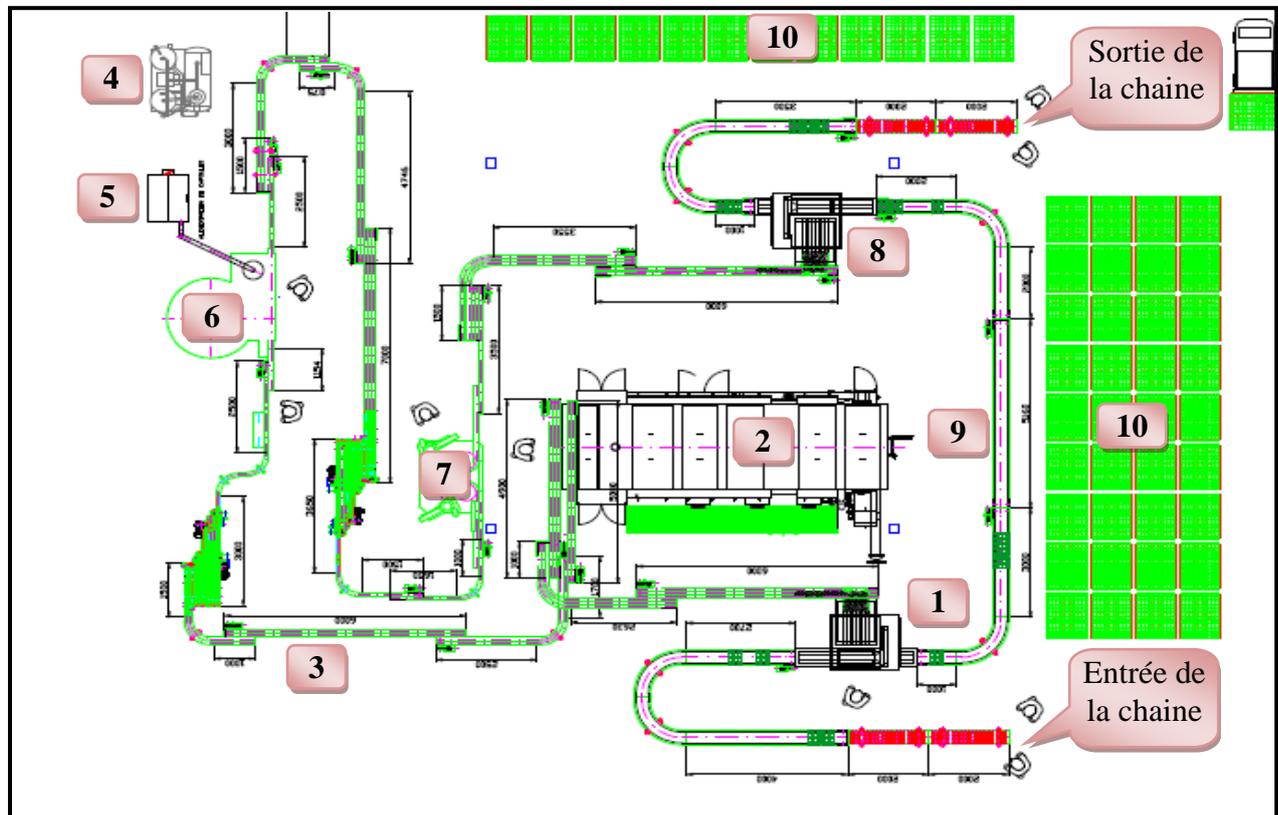


Figure 2.4. Schéma générale de la ligne de production [9]

- | | |
|-----------------------------|-----------------------------------|
| 1. Décaisseuse | 6. Remplisseuse |
| 2. Laveuse | 7. Etiqueteuse |
| 3. Transporteurs bouteilles | 8. Encaisseuse |
| 4. Prémix | 9. Transporteurs caisses |
| 5. Élévateur de capsules | 10. Zone de stockage produit fini |

2.4. Description Générale de la machine

La longue expérience et la continuelle actualisation du groupe NOVA comme un constructeur de machine depuis 1934 a conduit à construire les groupes MONOBLOC qui réunissent en une seule machine les fonctions de remplissage et de fermeture.

Ces unités modernes, ont été conçues comme un ensemble unique qui réalise les deux fonctions, cela les rend plus compactes, sans problème de synchronisation et de surveillance plus facile.

Ces groupes peuvent mettre en bouteille la plus part des boissons, qu'elles soient ou non avec du gaz carboniques, a froid ou a chaud, avec de la pulpe en suspension, etc...



Figure 2.5. La remplisseuse MONOBLOC

2.5. Caractéristiques techniques de la machine [10]

Les caractéristiques techniques de la remplisseuse sont résumées dans le tableau suivant :

Tableau 2.1. Caractéristiques techniques de la machine

Nom	MONOBLOC-NOVA
Année de construction	1995
Nombre des robinets de remplissage	50 robinets
Nombre des courons de la capsulation	10 courons
Capacité maximale de production	14000 B/H
Poids de la machine	6800 kg
Tension électrique principale	380 V - 50 Hz
Tension électrique de commande	24 V
Air comprimé pression	6-8 Bar
Eau pression	2.5 Bar
carbonique pression	8-10 Bar

2.6. Les différents constituants de la machine

La machine est composée des éléments principaux comme le montre les deux schémas suivants :

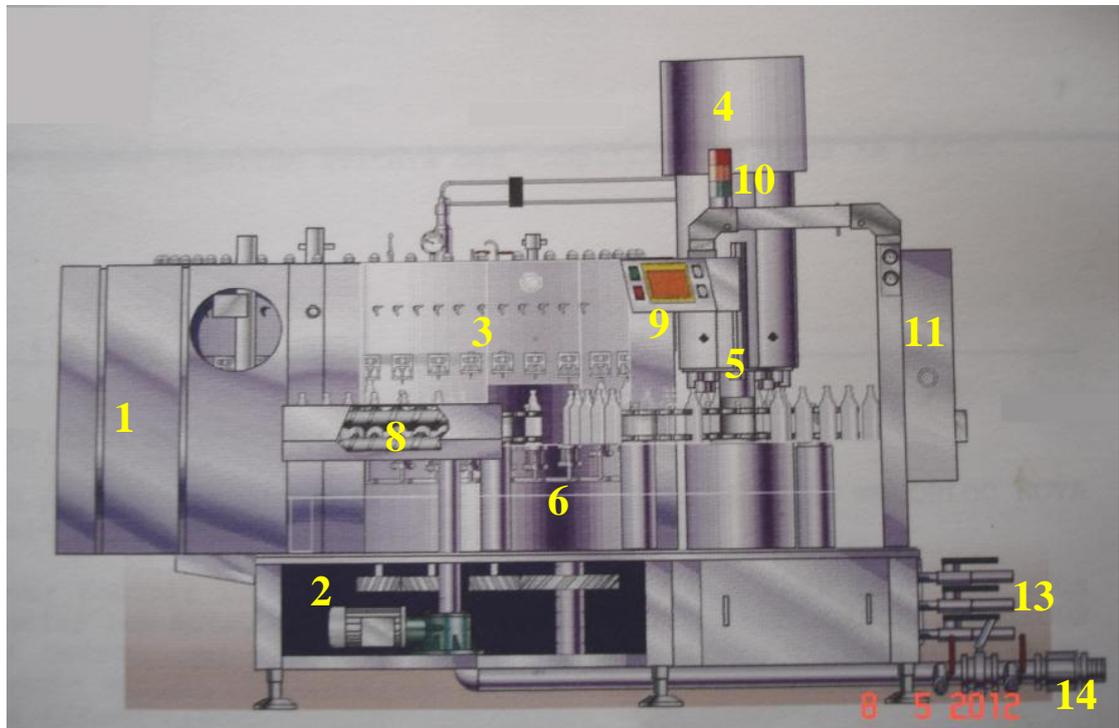


Figure 2.6. Vue face de la remplisseuse MONOBLOC [10].

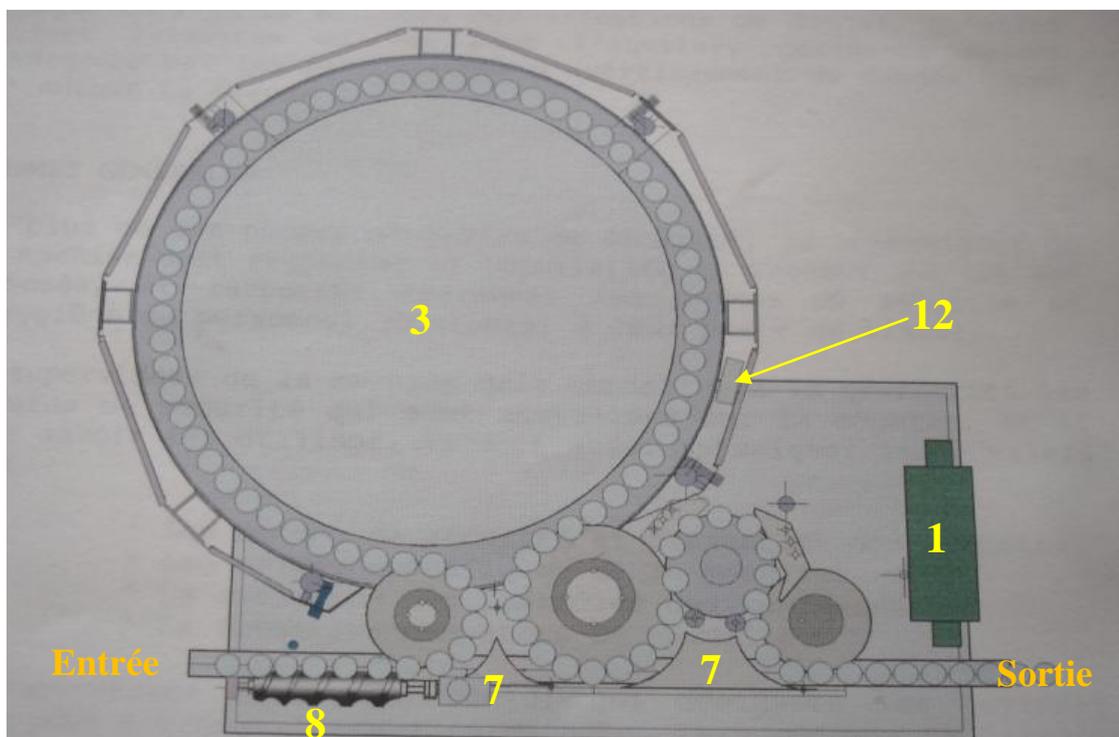


Figure 2.7. Vue de haut de la remplisseuse MONOBLOC [10].

1. Châssis principale de la machine.
2. Moteur principal.
3. Réservoir annulaire avec 50 soupapes de remplissage.
4. Trémie entraînée par un moteur secondaire.
5. Capsuleuse.
6. Plateau porte-bouteilles.
7. Colonnes étoiles, aussi bien d'entre que de sortie.
8. Vis glissière d'entrée des bouteilles pour centrer chaque bouteille sous un robinet.
9. Panneau de commande.
10. Tourelle lumineux
11. Armoire électrique.
12. Came mécanique.
13. Trois vannes pour l'alimentation de la machine avec :
 - L'air.
 - La contre-pression.
 - L'eau.
14. Entrée produit via le premix.

2.6.1. Le réservoir de remplissage

Ce réservoir représente le cœur de la machine il reçoit le produit fini depuis le premix à travers un canal situé au collecteur inférieur, la contre-pression sous forme de CO₂ s'écouler à l'intérieur de ce réservoir à travers un autre canal situé au collecteur supérieur. Cette contre-pression est un élément très essentiel dans le type de remplissage utilisé par cette machine (le remplissage iso-barométrique). Pour transférer le produit dans les bouteilles, ce réservoir contient 50 soupapes consacrées à cette opération.

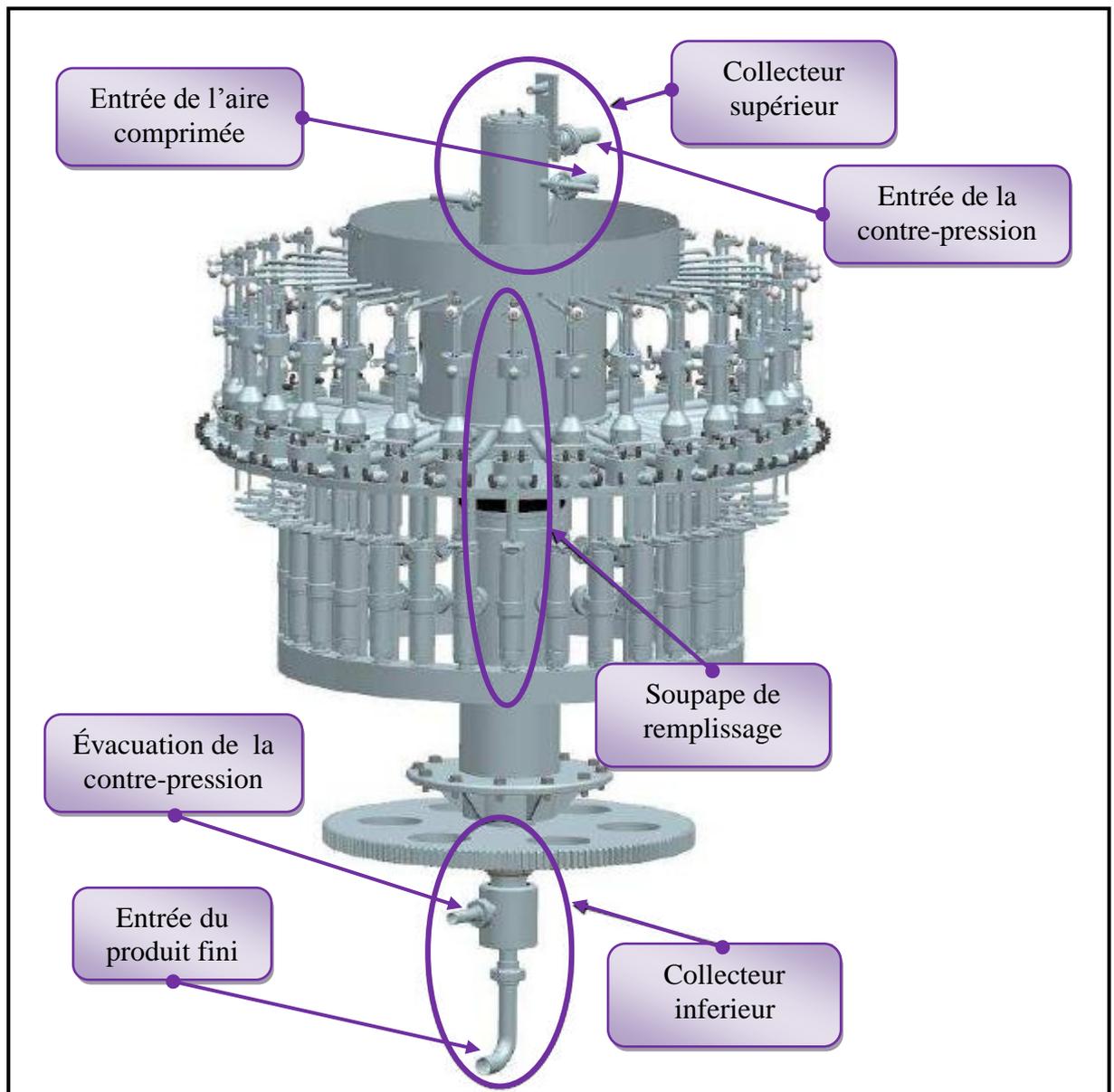


Figure 2.8. Le réservoir de remplissage.

2.6.1.1. Le collecteur supérieur

Il assure l'alimentation du réservoir avec la contre-pression et l'air comprimée, et aussi garantir la juste axialité entre le corps fixe externe et l'arbre rotatif interne.

2.6.1.2. Collecteur inférieur

Le collecteur inférieur possède la double fonction d'alimenter le réservoir avec le produit fini et d'évacuer la contre-pression.

2.6.1.3. Les soupapes de remplissage

Le but des soupapes de remplissage est de transférer la bonne quantité de produit dans la bouteille, dans le plus court temps, dans les meilleures conditions sanitaires, avec une précision extrême. Un grand avantage de ce système c'est que 50 soupapes suffisent sur la machine pour un rendement de 14.000 bouteilles/heure.

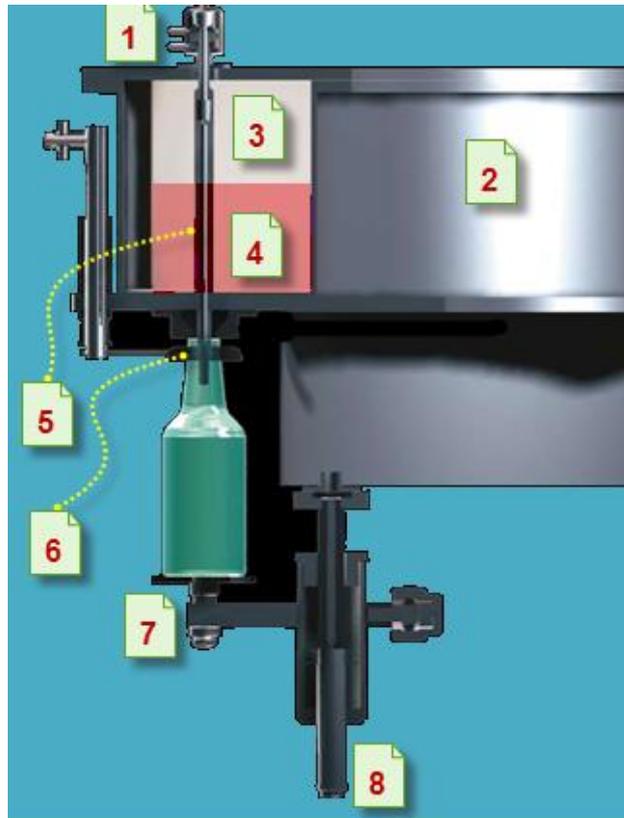


Figure 2.9. Schéma des soupapes de remplissage

1. Le robinet
2. Le réservoir
3. La contre-pression
4. Le produit fini
5. Bec de remplissage
6. Tulipe de centrage
7. Sellette
8. Vérin pneumatique (élevateur de bouteille)

2.6.2. La trémie

C'est l'élément qui reçoit et stocke les capsules venant de l'élévateur des capsules, puis elle les transfère vers la capsuleuse par l'intermédiaire d'un canal, la présence des capsules dans la trémie est contrôlée par un capteur de proximité inductif et sur le canal par deux capteurs photocellules.

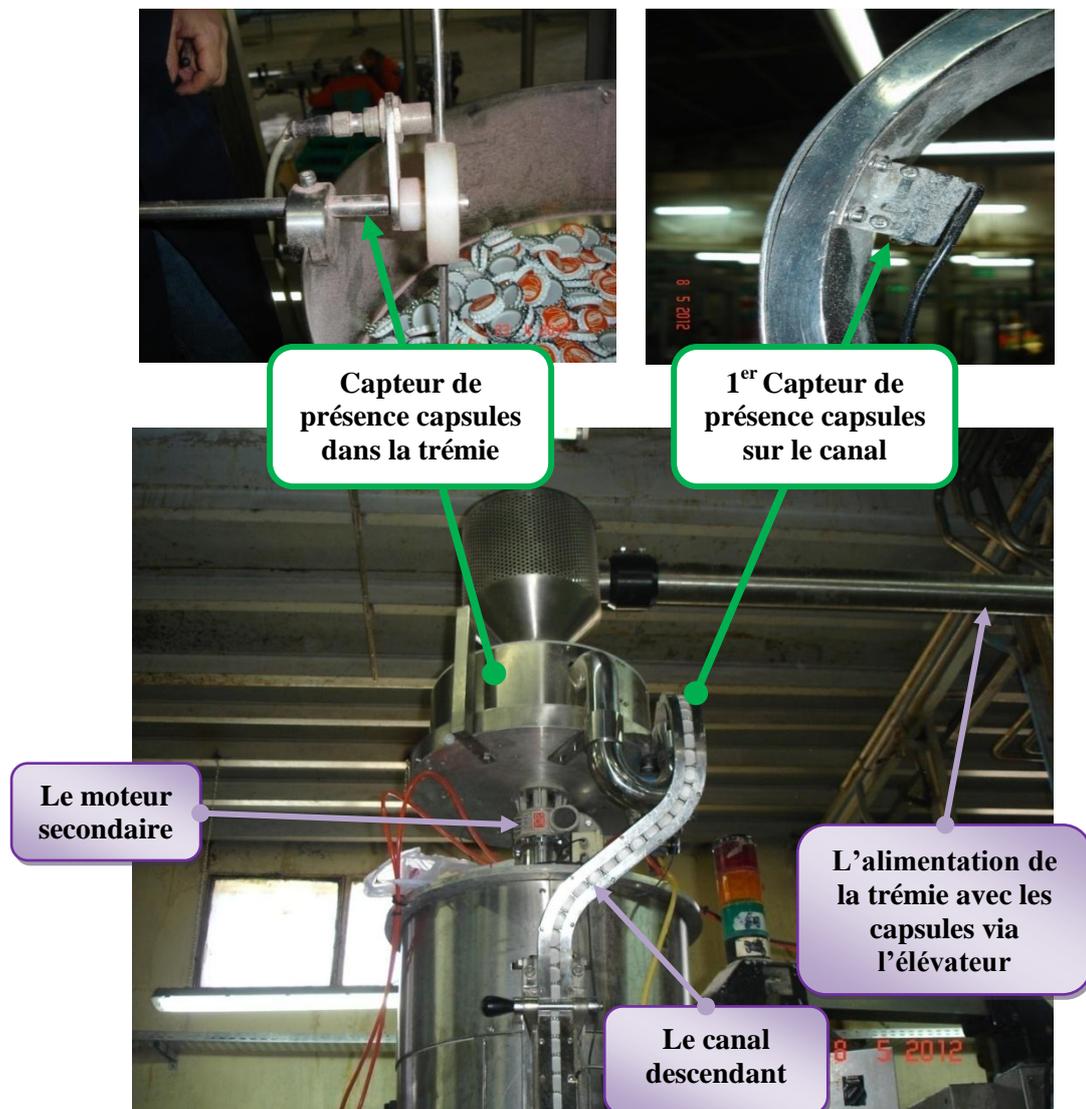


Figure 2.10. La trémie

2.6.3. L'élévateur des capsules

C'est l'élément qui assure l'alimentation de la trémie avec les capsules, le transfert de ces derniers depuis le dépôt de son stockage ce fait par un souffleur pneumatique qui les envoie dans un canal vers la trémie.



Figure 2.11. L'élevateur des capsules.

2.6.4. La capsuleuse

C'est la partie de la machine qui ferme les bouteilles après l'opération de remplissage par les capsules descendantes dans le canal.



Figure 2.12. La capsuleuse

2.7. Cahier de charges

Quand la machine aura donné deux tour à vitesse lente pour assurer que toutes les soupapes sont fermées, nous pourrons mettre la machine en marche:

- Un moteur asynchrone associé à un variateur de vitesse entraîne la machine.
- La machine règle automatiquement sa vitesse à travers le variateur à l'aide de détecteurs de présence des bouteilles à l'entrée par C1, C2 (*capteurs photocellule*) et C3 (*capteur de proximité inductif*), et à la sortie par C4, C5 et C6 (*capteurs photocellule*).



Figure 2.13. Capteur photocellule d'accumulation



Figure 2.14. Capteur de sécurité d'entrée(C3)

- S'il manque des bouteilles à l'entrée de la remplisseuse détectée par C1 et C2, la machine réduira automatiquement sa vitesse pour pouvoir récupérer les bouteilles. Si le manque de bouteilles persiste et arrive jusqu'au C3, la machine s'arrêtera totalement.
- D'une manière inversée, si C4 et C5 détectent une accumulation des bouteilles à la sortie, la machine réduira automatiquement sa vitesse, si cette accumulation arrive au C6, la machine s'arrêtera totalement.
- Dans la trémie se trouve le détecteur C7 (voir figure 2.10) qui contrôle le niveau des capsules, ce détecteur mettra en marche automatiquement l'élévateur de capsules pour que la trémie récupère le niveau correct de capsules à l'intérieur de celle-ci.
- Dans le canal descendant, se trouve le détecteur C8 qui contrôle l'accumulation des capsules et le détecteur C9 qui contrôle la présence des capsules qui alimentent les têtes de la capsuleuse. En cas de manque des capsules, la machine réduira automatiquement sa vitesse et récupéra sa vitesse normale au moment où C8 détectera de nouveau des capsules. Si le manque arrive jusqu'au C9, la machine s'arrêtera.
- Un vérin pneumatique V1 situé à l'entrée ouvrira les soupapes de remplissage quand le détecteur C10 (installé près de V1) détectera la présence des bouteilles.
- Un vérin pneumatique V2 suivi par une came permet de dégager le reste de produit dans les canules pour éviter la formation de la mousse en cours de remplissage. (le vérin ouvrira le robinet et la came le-ferme). Ce vérin s'ouvrira après 2s de la mise en marche du vérin V1.

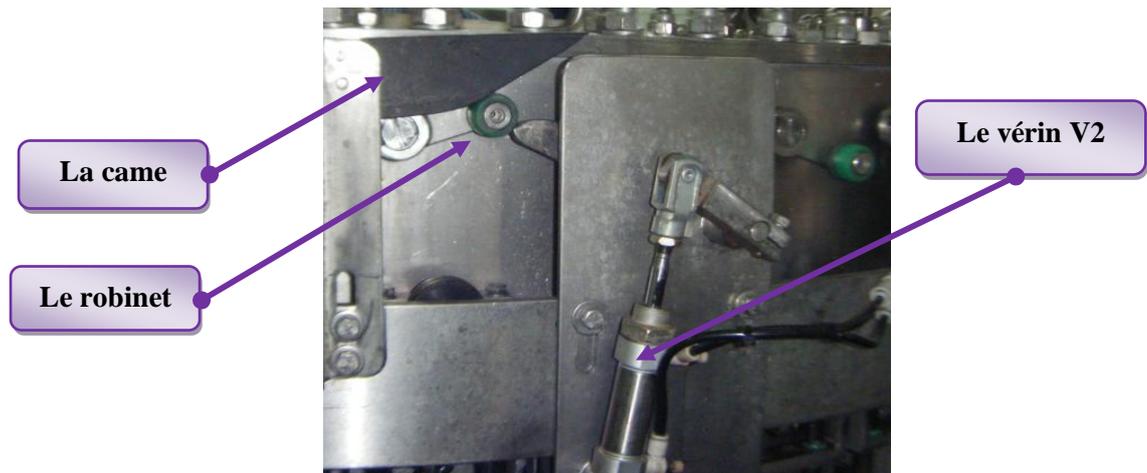


Figure 2.15. Mécanisme de nettoyage de la canule.

- Un détecteur d'explosions (C11) est installé sur la machine pour que si une quelconque explosion se produise durant l'opération de remplissage, le processus de nettoyage des restes de l'explosion se fasse automatiquement par l'intermédiaire d'une électrovanne, cette électrovanne s'ouvrira durant 30 seconds après la détection de l'explosion.
- Un capteur de proximité inductif (C12) vérifiera l'état des tulipes de centrage des bouteilles après le remplissage, si un des ces tulipes ne descende pas la machine arrêtera immédiatement.
- Un compteur est installé a la sortie de la machine pour calculer le nombre des bouteilles remplies.



Figure 2.16. Compteur des bouteilles remplies.

2.8. Procédure de remplissage

Les phases de remplissage des sont résumées comme suit :

2.8.1. Phase 1 : Elévation bouteille

Une came mécanique soulève les vérins pneumatiques pour pousser les bouteilles vers la tulipe de centrage. L'action du vérin permet à la bouteille de garder sa position sous le robinet.

2.8.2. Phase 2 : Pressurisation de la bouteille

Pour pressuriser la bouteille avec du CO₂, on agit sur la commande d'un vérin pneumatique qui ouvre les robinets ce qui permet de mettre en liaison la chambre de contre-pression avec l'intérieur de la bouteille.

2.8.3. Phase 3 : Remplissage

Le produit commence à couler à l'intérieur de la bouteille dès que celle-ci atteint la même pression que la chambre de contre-pression. L'air contenu à l'intérieur de la bouteille est évacué par un petit bec prévu à la base inférieure du robinet. Le débit du produit est contrôlé par un déflecteur monté à l'extérieur du bec afin d'éviter toute turbulence et la formation de mousse qui est un élément nuisibles à un bon remplissage. Le remplissage est terminé lorsque le niveau du produit atteint la partie inférieure du bec et bouche le retour de l'air.

2.8.4. Phase 4 : Fermeture

A la sortie, une came mécanique ferme les robinets, et une autre fait descendre les sellettes qui portent les bouteilles, donc les bouteilles sont amenées au niveau de l'étoile de transfert à la capsuleuse.



Figure 2.17. Came de fermeture des robinets.

2.8.5. Phase 5 : Décompression

La pression qui reste à l'intérieur de la bouteille après l'opération du remplissage est lentement libérée par le biais d'un purgeur.



Figure 2.18. Décompression du reste d'air

2.9. Problématique

Actuellement, le mode de marche du groupe est semi-automatique, donc les opérateurs trouvent des difficultés pour respecter les temps d'exécution des opérations en particulier lorsqu'ils sont appelés à traiter une grande quantité de produit dans un temps réduit.

En effet, nous avons constaté que ce mode entraîne une baisse substantielle de la production, d'où la nécessité d'automatiser le groupe de remplissage afin d'augmenter la productivité et d'améliorer les conditions de travail des personnels.

2.10. Conclusion

L'objectif de ce chapitre est de présenter la remplisseuse MONOBLOC ainsi que ses différents constituants et son principe de fonctionnement. Cette présentation a permis d'identifier les principaux dispositifs de commande de cette machine à travers l'automate programmable industrielle S7-300.

Chapitre 03

LA CREATION DU PROJET, LA SIMULATION ET LA SUPERVISION

3.1. Introduction

Pour conduire le groupe de remplissage, nous allons réaliser un programme qui va être chargé dans l'automate grâce au logiciel de programmation des systèmes d'automatisation SIMATIC STEP7, à base de ce programme on va construire une interface homme-machine permette de superviser le fonctionnement du groupe en utilisant le logiciel WinCC. Dans ce chapitre, on va présenter les différentes étapes nécessaires pour la création de ce programme.

3.2. Aperçu du progiciel STEP 7

STEP7 est le progiciel de base pour la configuration et la programmation des systèmes d'automatisation SIMATIC S7-300 et S7-400 et qui s'exécute sous un environnement Windows à partir d'une console de programmation ou d'un PC [11].

STEP 7 assiste l'utilisateur dans toutes les phases de création des solutions d'automatisation, par des opérations consistent à:

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

3.3. Structuration d'un programme STEP7

3.3.1. Programmation linéaire

Le programme utilisateur peut s'écrire en entier en une seule liste ou dans un seul bloc où les instructions s'exécutent les une après les autres jusqu'à la fin. Cela n'est toutefois recommandé que pour des programmes simples s'exécutant sur des CPU d'une mémoire peu importante. Le développement d'un tel programme par cette méthode devient difficilement gérable lorsque ce dernier dépasse un certain volume.

3.3.2. Programmation Structurée

La programmation structurée consiste à subdiviser un programme plus ou moins complexe en plusieurs sous-programmes où chacun des ces sous-programmes est développé pour exécuter une tâche ou fonction spécifique. Un autre programme dit programme principal sera chargé de gérer ces sous-programme.

3.4. Langages de programmation de STEP 7

STEP 7 possède trois langages de programmation possibles qui peuvent être combinés dans le même programme :

- Programmation à schéma logique (LOG).
- Programmation à schéma à contacte (CONT).
- Programmation à liste d'instruction (LIST) [12].

3.4.1. Le schéma logique (LOG)

C'est un langage de programmation graphique qui utilise les boîtes de l'algèbre de Boole pour représenter les opérations logiques. Les fonctions complexes, comme par exemple les fonctions mathématiques, peuvent être représentées directement combinées avec les boîtes logiques.

3.4.2. Le schéma à contacts (CONT)

C'est un langage de programmation graphique avec une syntaxe des instructions fait penser aux schémas de circuits électriques. CONT permet de suivre facilement le trajet du courant entre les barres d'alimentation en passant par les contacts, les éléments complexes et les bobines.

3.4.3. Liste d'instruction (LIST)

C'est un langage de programmation textuel proche de la machine. Dans un programme LIST, les différentes instructions correspondent, dans une large mesure, aux étapes par lesquelles la CPU traite le programme dans une suite littérale où chaque instruction comprend un code opération et un opérande.

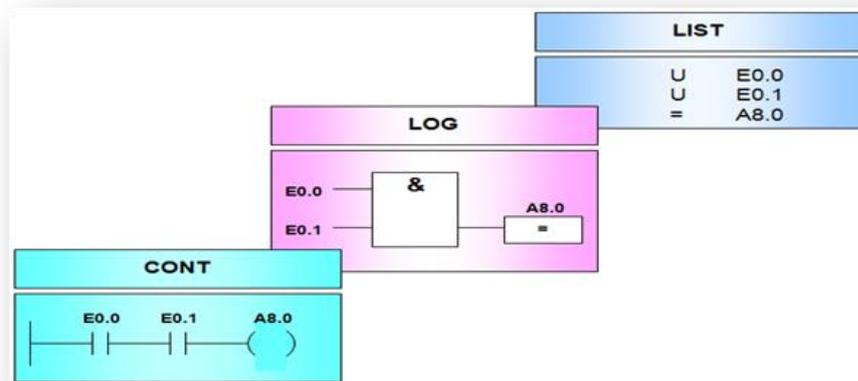


Figure 3.1. Mode de représentation des langages basiques de programmation STEP7.

3.5. SIMATIC Manager

C'est le gestionnaire qui gère toutes les données relatives à un projet d'automatisation, il constitue l'interface principale qui assure la navigation entre toutes les applications requises pour la configuration et le paramétrage du projet.

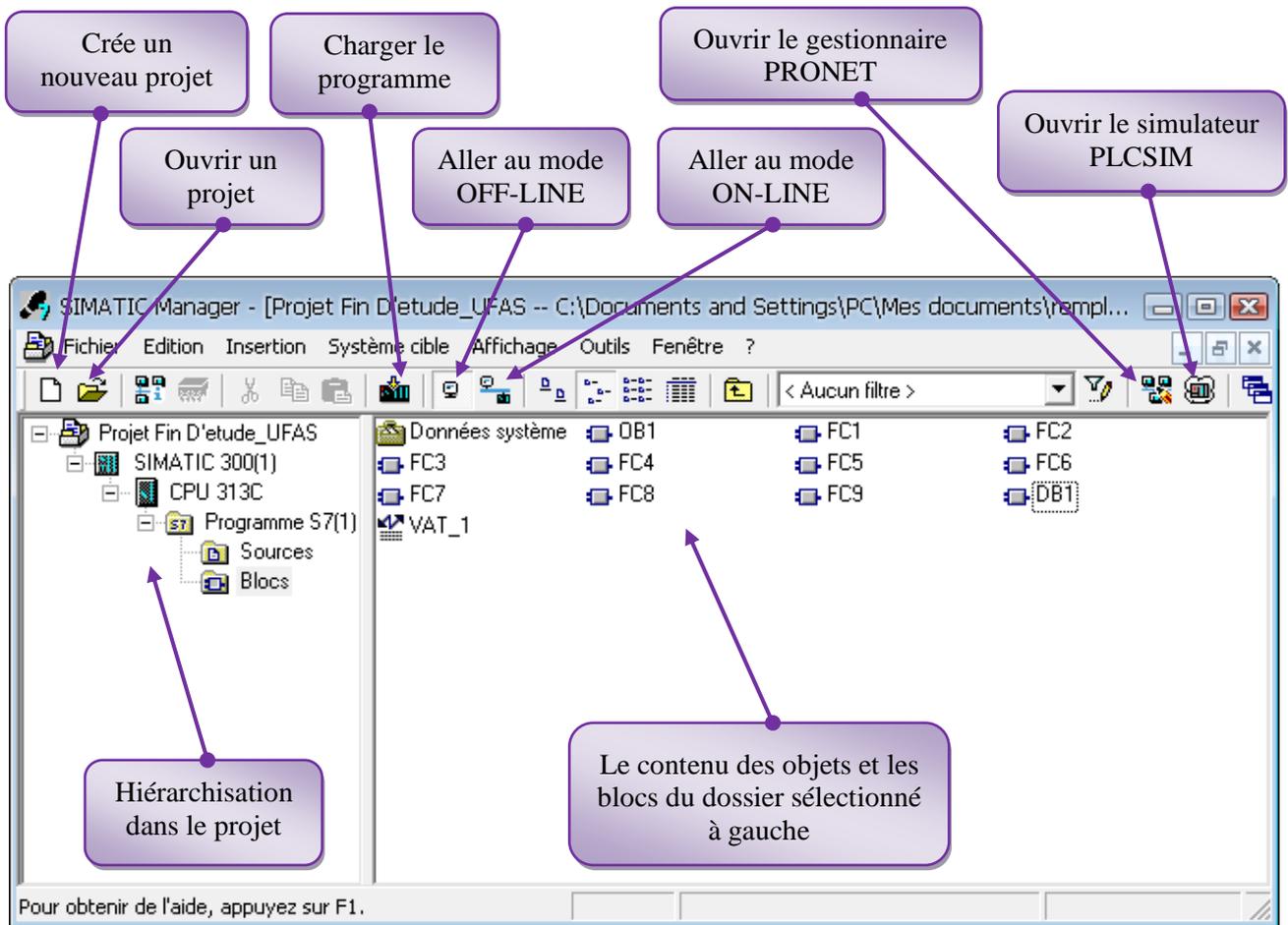


Figure 3.2. La fenêtre SIMATIC MANAGER

3.6. Hiérarchisation dans un projet

Chaque programme utilisateur est structuré sous forme de projet. Un projet représente l'ensemble des données et des programmes d'une solution d'automatisation et se trouve à la tête d'une hiérarchie d'objets comme suit :

- Objet projet.
- Objet Station.
- Objet Modules programmables.
- Objet Programme S7.
- Objet dossier Sources.
- Objet dossier Blocs.

3.6.1. Objet station

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

3.6.2. Objet modules programmables

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

3.6.3. Objet programme S7

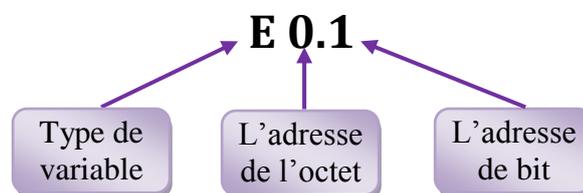
Un programme S7 est un dossier contenant les logiciels pour les modules programmables. Dans un programme S7 on trouve : une table des mnémoniques, un dossier "Blocs" et un dossier "Sources".

3.6.3.1. Table des mnémoniques

Ce sont des noms symboliques qui vont être utilisées dans la programmation. L'utilisation de noms communs est plus simple que la manipulation des adresses ou opérandes.

a. Définitions des variables

Les variables sont des emplacements mémoire fixes dans la CPU de l'automate. Ces variables sont comme l'identificateur des entrées et des sorties de l'automate dans lesquelles des valeurs sont écrites et/ou lues et qui sont de la structure suivante :



b. Types de variables

Le SIMATIC STEP 7 comporte plusieurs types de variables qui peuvent être manipulés pour la programmation des différents blocs :

- **Zone E** : Réserver aux entrées
- **Zone A** : Réserver aux sorties
- **Zone M** : Mémoire utilisateur globale
- **Zone L** : Mémoire locale, associée à un module de programme
- **Zone P** : Accès à la périphérie
- **Zone T** : Mémoire des temporisations
- **Zone Z** : Mémoire des compteurs
- **Zone DB** : Mémoire utilisateur ou système structuré dans des blocs de données

Chacune des zones précédentes est présentée sous plusieurs formes différentes comme le montre le tableau suivant

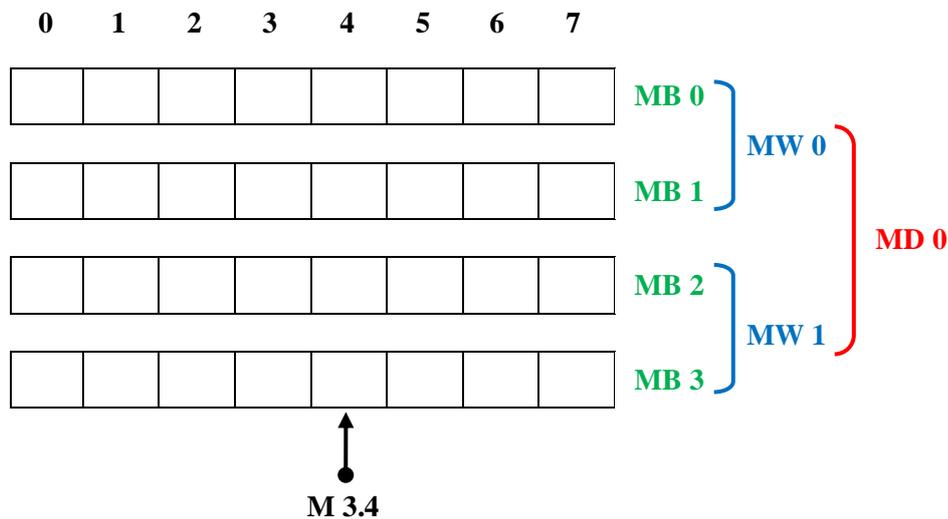
Tableau 3.1. Types de variables manipulées dans le STEP 7

Symbole SIMATIC :	Signification :
E	Bits d'entrée
EB	Octet d'entrée
EW	Mot d'entrée
ED	Double mot d'entrée
A	Bits de sortie
AB	Octet de sortie
AW	Mot de sortie
AD	Double mot de sortie
M	Bit mémoires utilisateurs
MB	Octet mémoire utilisateurs
MW	Mot mémoire utilisateurs
MD	Double mot mémoire utilisateurs
L	Bit dans la mémoire locale
LB	Octet dans la mémoire locale
LW	Mot dans la mémoire locale
LD	Double mot dans la mémoire locale
PEB	Octet de périphérie d'entrée
PAB	Octet de périphérie de sortie
PEW	Mot de périphérie d'entrée
PAW	Mot de périphérie de sortie
PED	Double mot de périphérie d'entrée
PAD	Double mot de périphérie de sortie
T	Temporisation
Z	Compteur
DBX	Bit dans un bloc de donnée
DBB	Octet dans un bloc de donnée
DBW	Mot dans un bloc de donnée
DBD	Double mot dans un bloc de donnée

c. Adressage des variables

Les zones E, A, M, DB, PE et PA sont rangés dans des octets, on peut accéder à un bit, à un octet (8bits), à un mot (16 bits) ou à un double mot (32 bits).

Exemples



- **M 3.4** : correspond au bit 4 de l'octet 3
- **MB 3** : correspond aux 8 bits de l'octet 3
- **MW1** : correspond au 16 bits constitué par les octets 2 et 3
- **MD 0** : correspond au 32 bits constitué des octets 1, 2, 3 et 4

3.6.4. 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, FC) qui contiennent les programmes qu'on doit charger dans la CPU, et le bloc de données (DB) qui contient les paramètres du programme.

3.6.4.1. Les blocs d'organisation (OB)

Les blocs d'organisation (OB) constituent l'interface entre le système d'exploitation et le programme utilisateur. Ils sont appelés par le système d'exploitation selon leur priorité et gèrent le traitement des programmes cycliques et déclenchés par alarme, ainsi que le comportement à la mise en route de l'automate programmable et le traitement des erreurs.

Les blocs d'organisation définissent l'ordre (événements de déclenchement) 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.

a. Bloc d'organisation de traitement cyclique (OB1)

Le bloc d'organisation OB1 sert à l'exécution cyclique du programme utilisateur. On programme, dans l'OB1, des appels correspondant aux blocs fonctionnels FB ou aux fonctions FC ou à d'autres types de structures. L'OB1 ne peut être appelé que par le programme système dès que l'exécution du programme de mise en route est achevée.

3.6.4.2. Bloc fonctionnel (FB)

Un bloc fonctionnel est un bloc de code avec mémoire, c'est-à-dire avec données statiques. Il permet la transmission de paramètres dans le programme utilisateur. Pour cette raison, les blocs fonctionnels conviennent à la programmation de fonctions complexes à caractère répétitif, telles que la régulation et la sélection de fonctionnement. Un bloc fonctionnel disposant d'un mémoire (le bloc de données d'instance), l'accès à ses paramètres est possible à tout moment et à toute position du programme utilisateur.

3.6.4.3. Fonction(FC)

Une fonction est un bloc de code sans mémoire, c'est-à-dire sans données statiques. Elle permet la transmission de paramètres dans le programme utilisateur.

3.6.4.4. Les blocs de données (DB)

Dans les blocs de données, sont mémorisées les données nécessaires au traitement du programme et les données affectées à chaque bloc. On distingue deux types de blocs de données :

a. Blocs de données d'instance

Un bloc de données d'instance est associé à chaque appel de bloc fonctionnel transmettant des paramètres. Ces blocs contiennent les paramètres effectifs et les données statiques du FB.

b. Blocs de données globaux

Contrairement aux blocs de données d'instance, les blocs de données globaux servent à l'enregistrement des données du programme utilisateur pouvant être utilisées par tous les autres blocs: ils contiennent des données variables que le programme utilisateur utilise.

3.7. Création d'un projet S7 pour le groupe de remplissage

Les procédures qui permettent de créer un projet sous STEP 7, afin de piloter le groupe de remplissage, sont les suivantes :

1. Lancer **SIMATIC Manager** par une double clique sur son icône



2. On a vu déjà le **SIMATIC Manager**, donc après le démarrage de ce dernier cliquant directement sur **NEW** puis saisir le nom du projet dans la fenêtre de dialogue qui va être affichée

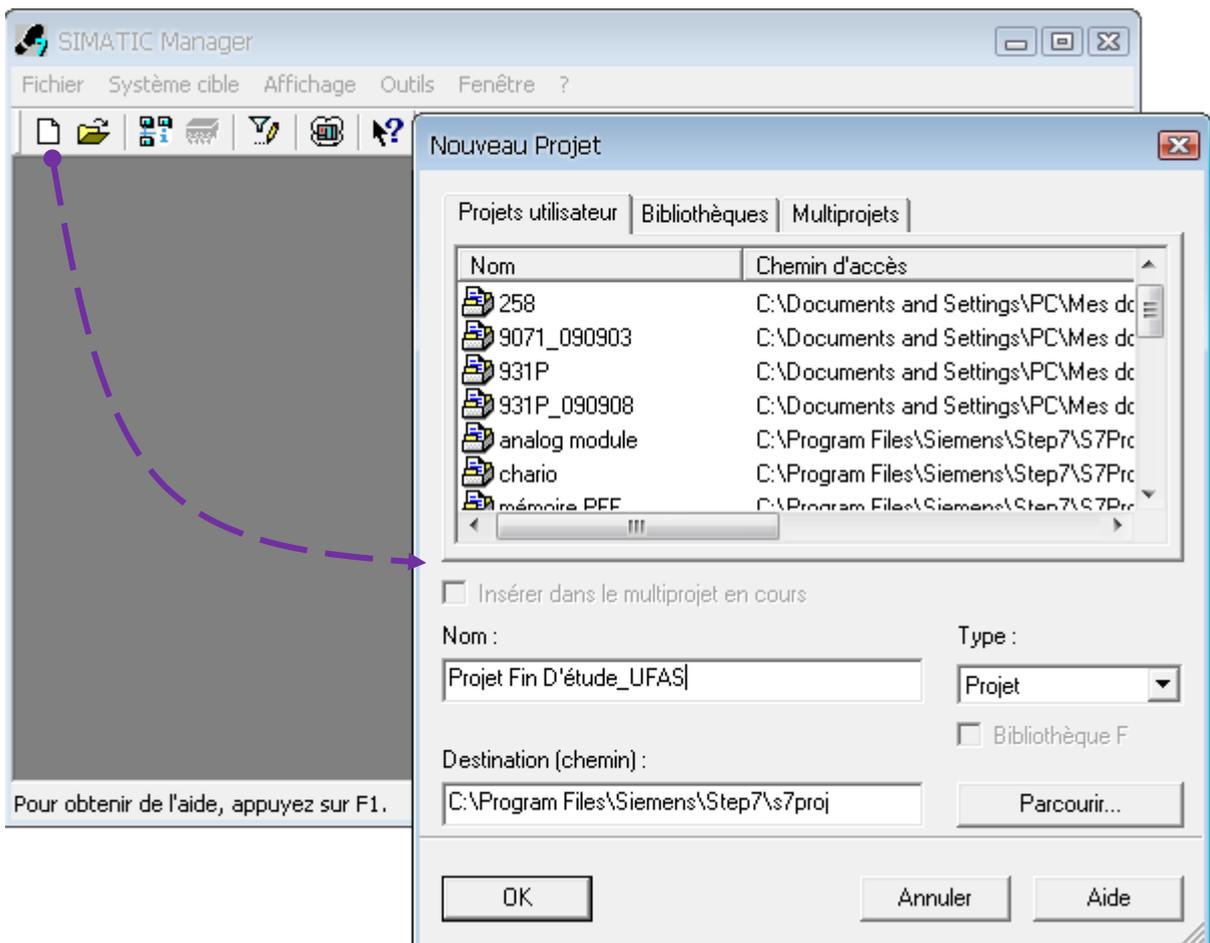


Figure 3.3. Création du projet

3. Une nouvelle fenêtre s'affichera permet d'insérer l'automate programmable requis, dans notre cas on a choisi l'S7-300 donc, de puis la barre d'outils on clique sur **Insertion** puis on choisi **Station SIMATIC 300**

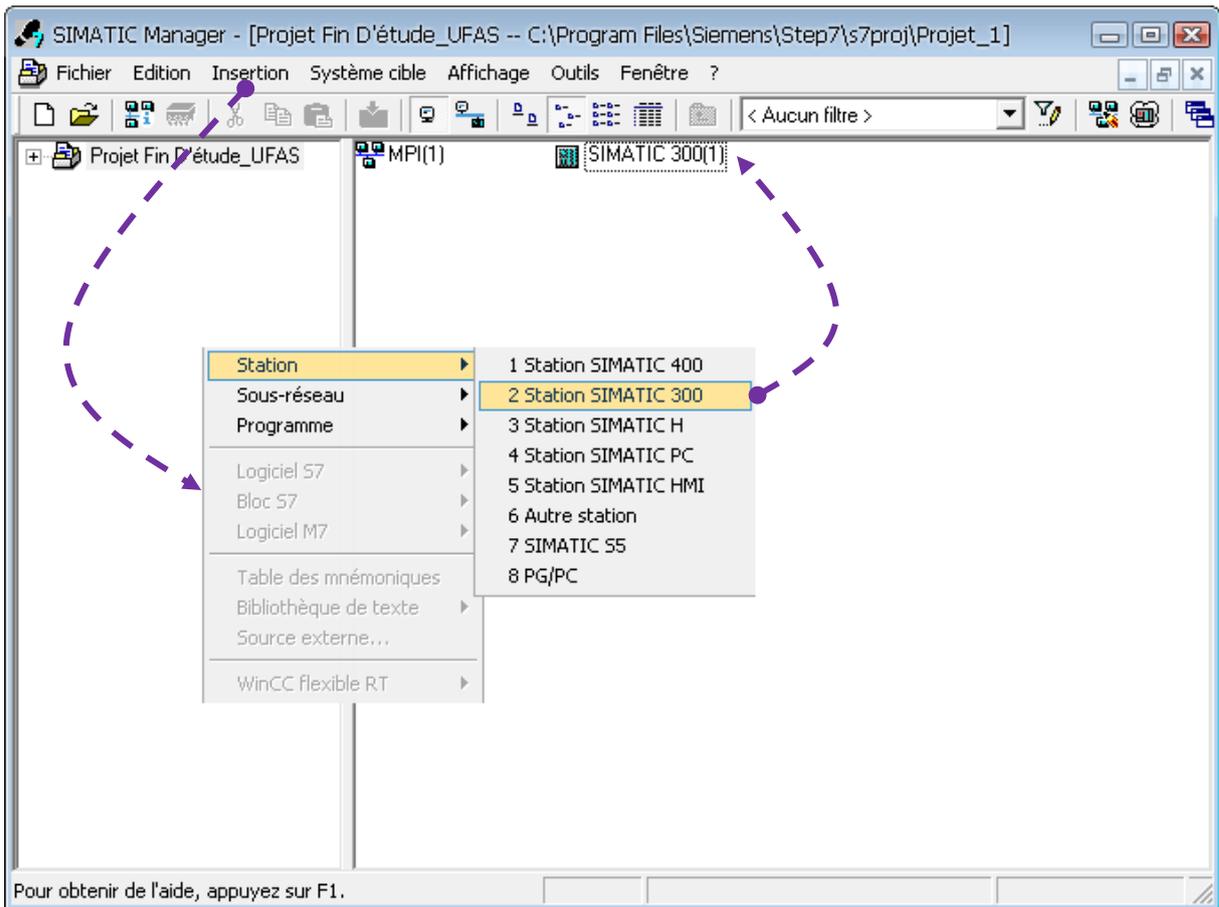


Figure 3.4. Choix de la station

3.8. La configuration matérielle

Sélectionner le dossier **Station SIMATIC 300** et double-cliquer sur **Matériel**. Ceci ouvre la fenêtre "*HW Config*" pour la configuration matérielle de la station :

On commence par le choix du châssis selon la station choisie préalablement. Pour la station SIMATIC 300, on aura à choisir le châssis « RACK-300 » qui comprend un rail profile qui va porter les modules de l'automate. Sur ce profile, l'alimentation préalablement sélectionnée se trouve dans l'emplacement n°1, parmi celles proposées notre choix s'est porté sur la « PS-307 10A ». La « CPU 313C » est impérativement mise à l'emplacement n°2. L'emplacement n°3 est réservé comme adresse logique pour un coupleur dans une configuration multi châssis. A partir de l'emplacement n°4, il est possible d'aller au choix jusqu'à 8 modules de signaux (SM), processeurs de communication (CP) ou modules fonctionnels (FM). Après cela il ne nous reste qu'à enregistrer et compiler.

La configuration matérielle étant terminée, un dossier « Programme S7 » est automatiquement inséré dans le projet.

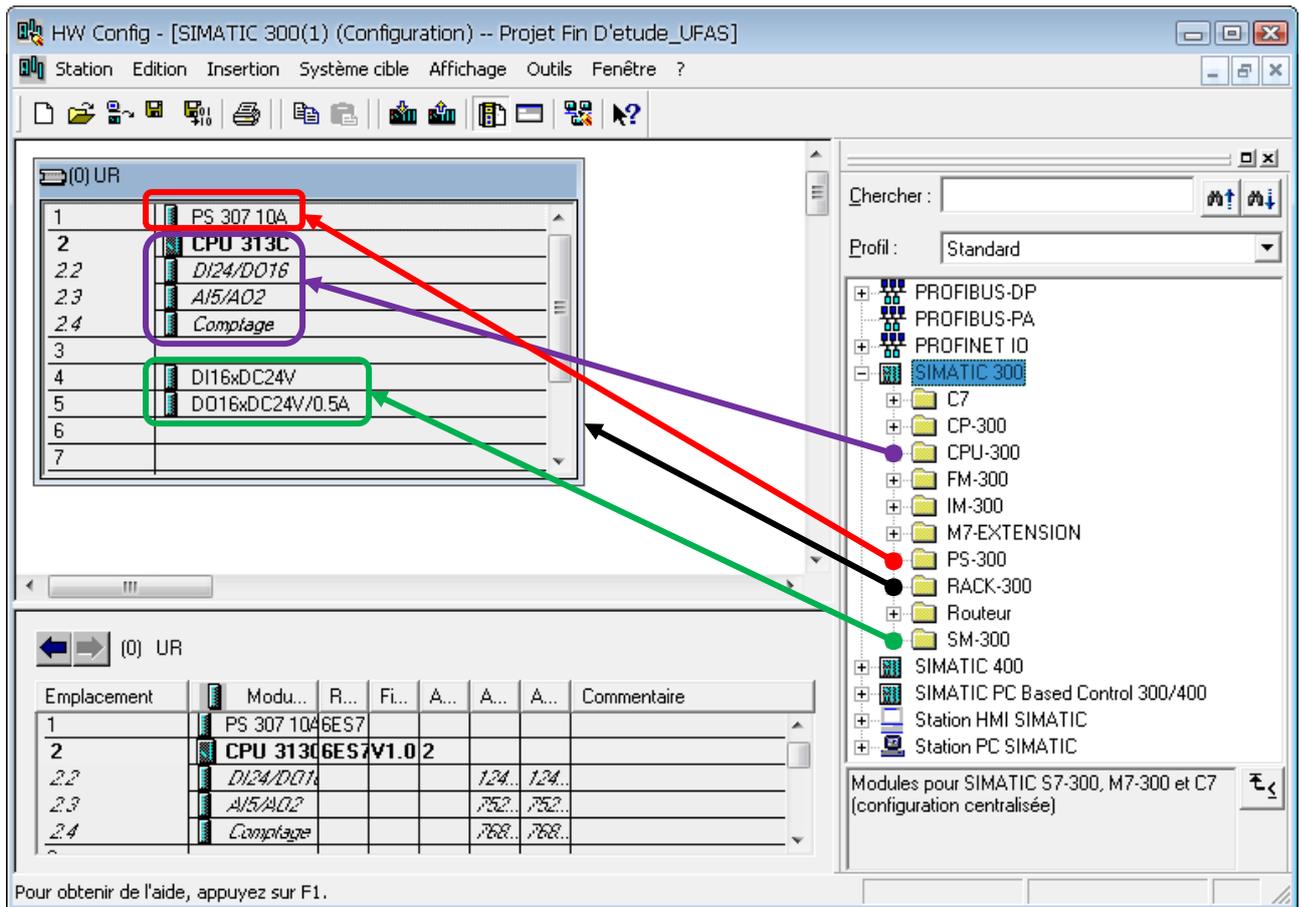


Figure 3.5. La configuration matérielle.

3.9. Création de la table des mnémoniques

Pour accéder à la table des mnémonique, on clique sur le dossier programme dans la fenêtre du projet, puis sur l'icône mnémoniques. Après avoir défini toutes les mnémoniques, il suffit d'enregistrer pour que les changements soient pris en compte dans le reste du projet.

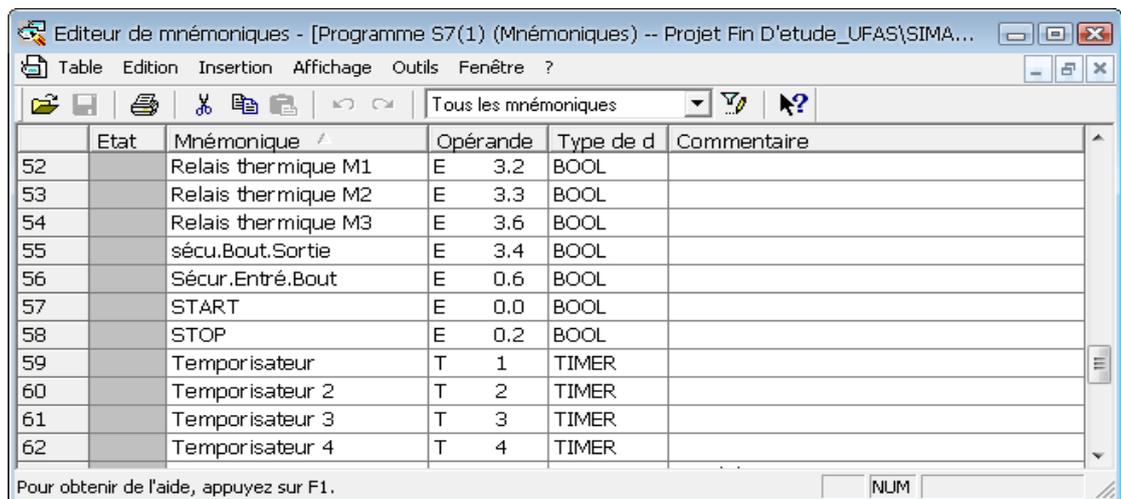


Figure 3.6. Editeur de mnémonique.

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 génère par STEP7.
- Ecrire éventuellement un commentaire dans la colonne prévue à cet effet.

3.10. Procédure de la programmation

Pour élaborer le programme qui gère l'automatisation visée, on suit la procédure suivante :

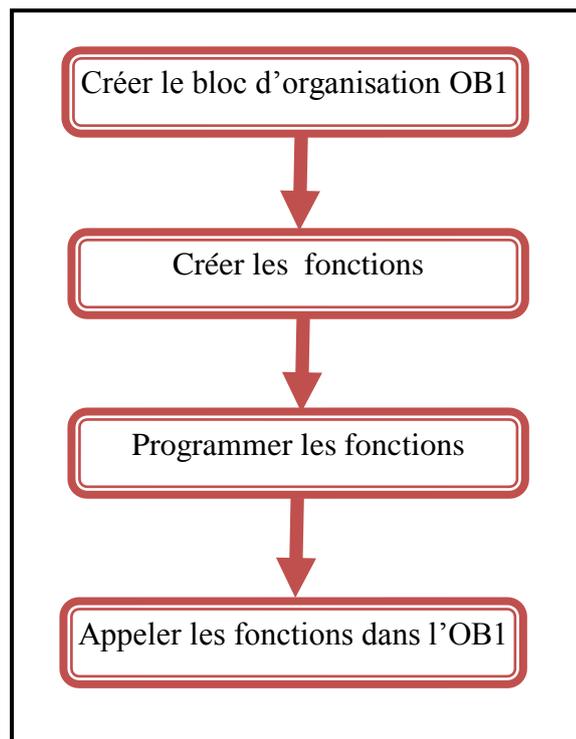


Figure 3.7. Procédure générale de programmation

Pour créer un bloc il suffi de le-choisir de puis commande **Insertion** → **Bloc S7**. Dans la boîte de dialogue qui s'ouvre sélectionnez l'éditeur de langage qui vous convient (LIST, CONT ou LOG) et confirmez avec « OK »

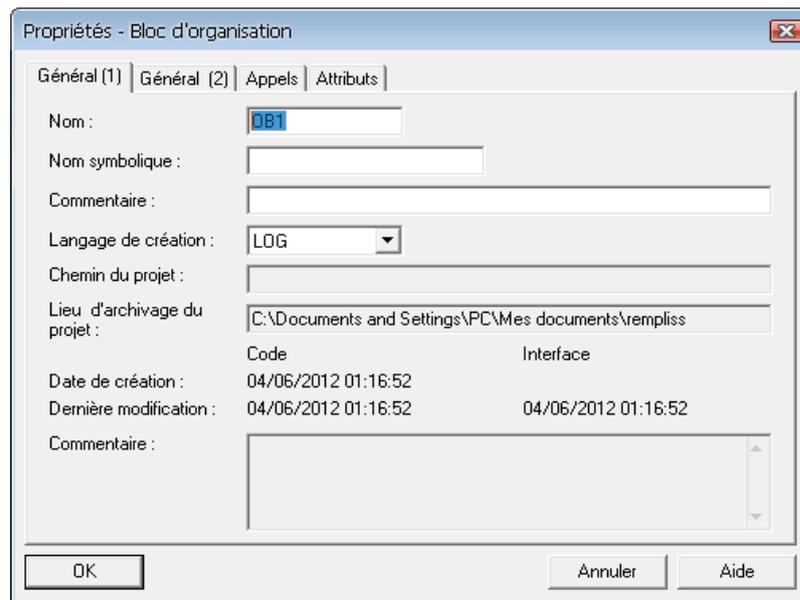


Figure 3.8. Insertion de blocs de programmation

3.11. Structure du programme

La structure du programme élaboré consiste à créer un bloc d'organisation OB1 gère l'exécution cyclique de 8 fonction FC, chacune de ces fonctions est réservée à la programmation d'un dispositif de commande ou la gestion d'une tâche :

- FC1 : réservée à la gestion du moteur principale ainsi que le variateur de vitesse
- FC2 : réservée à la gestion du moteur secondaire
- FC3 : réservée à la gestion des vérins
- FC4 : réservée à la gestion du processus de nettoyage
- FC5 : réservée au contrôle des capsules au niveau de la trémie
- FC6 : réservée au comptage des bouteilles
- FC7 : réservée à la gestion des conditions de marche et d'arrêt de la machine
- FC8 : réservée à la gestion des alarmes

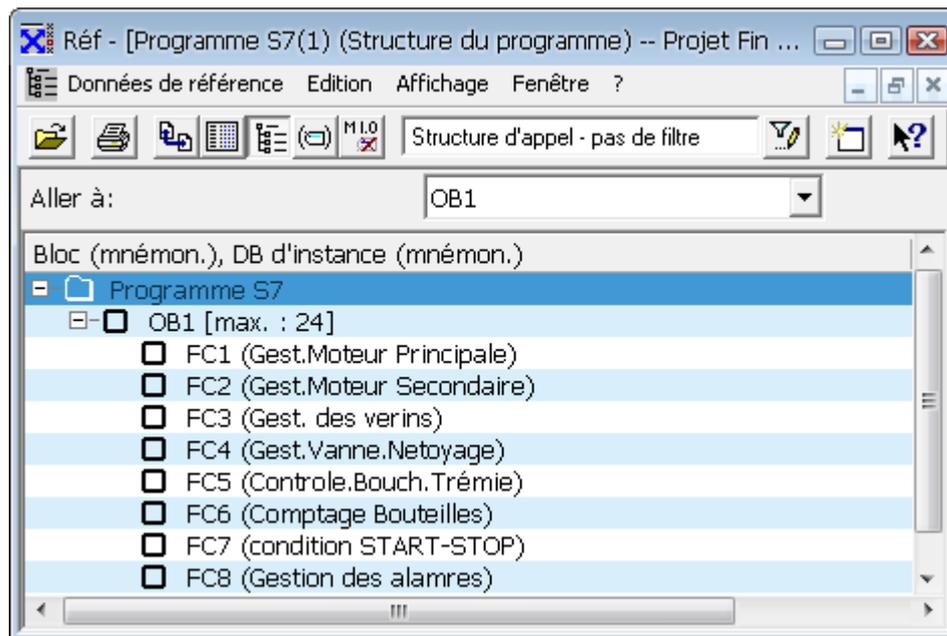


Figure 3.9. Structure du programme.

3.12. Simulation du programme avec S7-PLCSIM

3.12.1. Présentation du S7-PLCSIM

S7-PLCSIM est une application qui permet d'exécuter et de tester le programme utilisateur élaboré dans un automate programmable simulé dans l'ordinateur ou dans une console de programmation. La simulation étant complètement réalisée au sein du logiciel STEP 7, il n'est pas nécessaire qu'une liaison avec un matériel S7 quelconque (CPU ou modules de signaux) soit établie. La CPU simulée permet de tester les programmes destinés aussi bien aux CPU S7-300 qu'aux CPU S7-400. [13]

S7-PLCSIM dispose d'une interface simple permettant de visualiser et de forcer les différents paramètres utilisés par le programme. Tout en exécutant le programme dans la CPU simulée.

3.12.2. Ouverture du S7-PLCSIM

Pour ouvrir le simulateur on procède comme suit :

1. Démarrer le gestionnaire de projets SIMATIC en cliquant sur son icône.
2. Lancer l'application S7-PLCSIM en cliquant sur le bouton  qui se trouve dans la barre d'outils du gestionnaire de projets SIMATIC, ou en sélectionnant la commande **Outils Simulation des modules**.

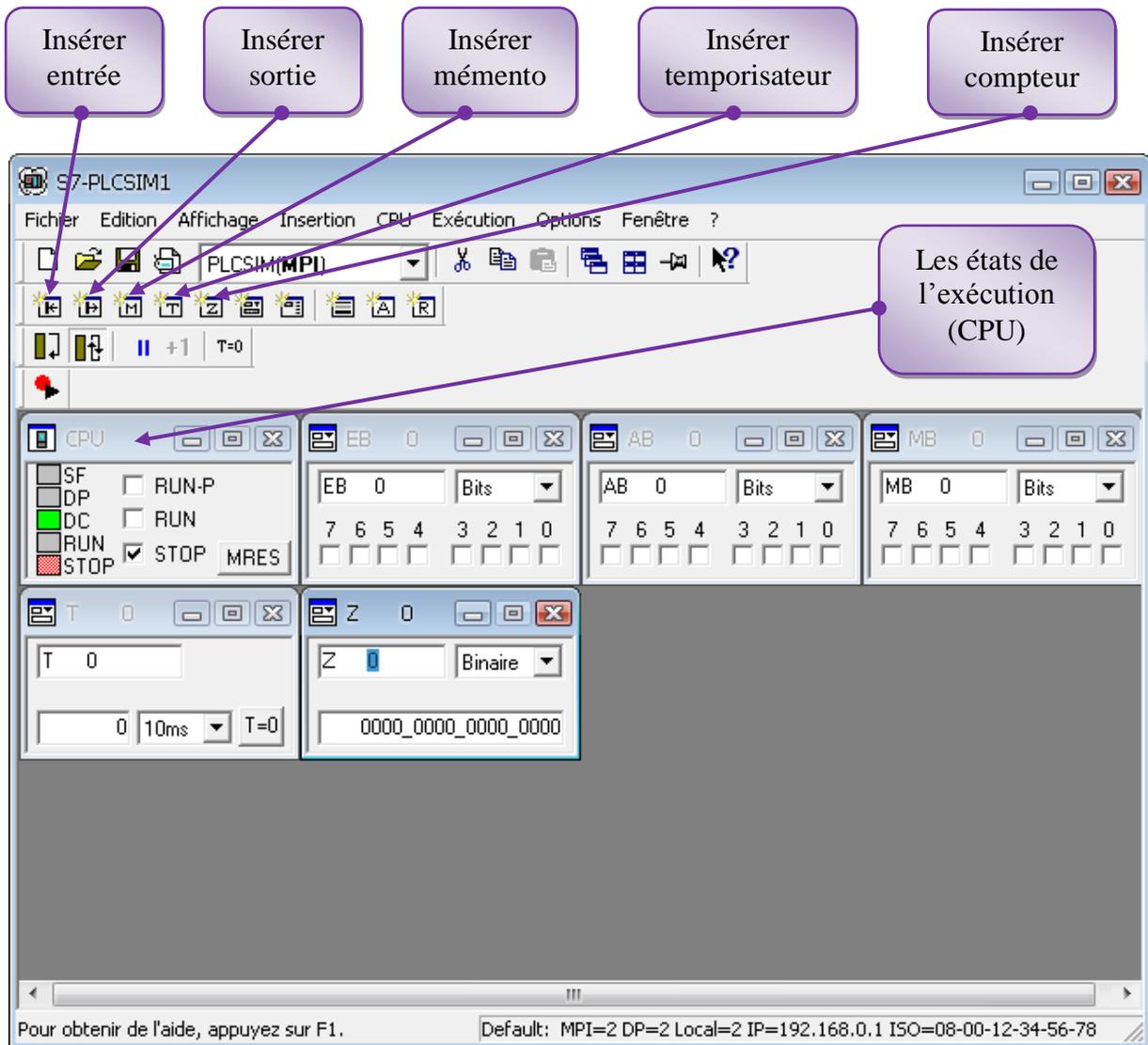


Figure 3.10. Le simulateur PLCSIM

3.12.3. Chargement du projet STEP 7

C'est l'étape qui permet de charger toutes les blocs et la configuration matérielle dans le système d'automatisation simulé. Le système de simulation reprend l'identité de la CPU chargée et toutes les données de liaison configurées. Pour charger le projet STEP 7, procédez de la manière suivante :

1. Naviguez dans SIMATIC Manager jusqu'à la station.
2. Cliquez sur l'icône  qui se trouve dans la barre d'outils du gestionnaire de projets SIMATIC ou choisissez la commande de menu **Système cible** → **Charger**.

3.12.4. Exécution du programme

3.12.4.1. Choix du mode de l'exécution

Une fois le programme soit chargé dans la CPU, on peut l'exécuter. Il faut s'assurer préalablement que le cycle continu est sélectionné comme mode d'exécution. Pour sélectionner le mode d'exécution continu du programme, choisir la commande **Exécution** → **Mode** → **Cycle continu** ou cliquer sur le bouton correspondant dans la barre d'outils.

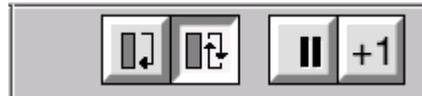


Figure 3.11. Choix du cycle de l'exécution du programme

3.12.4.2. Démarrage de l'exécution du programme

Dans le S7 PLCSIM, la CPU a trois modes de fonctionnement RUN-P, RUN et STOP et cinq indicateurs au cours de fonctionnement SF, DP, DC, RUN et STOP

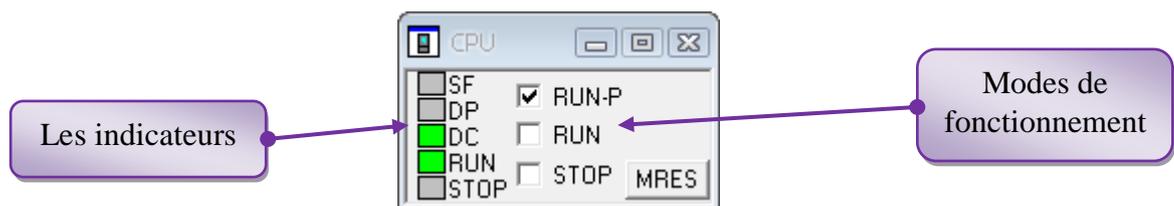


Figure 3.12. Les états du CPU dans le PLCSIM

a. Modes de fonctionnement de la CPU simulé

- **RUN-P** : La CPU exécute le programme et nous pouvons le modifier ainsi que ses paramètres à travers des fenêtres secondaires créées dans S7-PLCSIM pour modifier les données utilisées par le programme.
- **RUN** : La CPU exécute le programme en lisant les entrées, traitant le programme, puis en actualisant les sorties. Par défaut, nous ne pouvons pas charger de programme ni modifier des paramètres quand la CPU est en cour d'exécution.
- **STOP** : La CPU n'exécute pas le programme. Contrairement à ce qui se passe dans les CPU réelles en STOP, les sorties ne prennent pas de valeurs prédéfinies, mais elles conservent l'état en cours au passage de la CPU en STOP. Nous pouvons charger des programmes dans la CPU pendant qu'elle est en STOP. Quand l'état de fonctionnement passe de STOP à RUN, l'exécution du programme reprend à la première instruction.

b. Indicateurs de la CPU

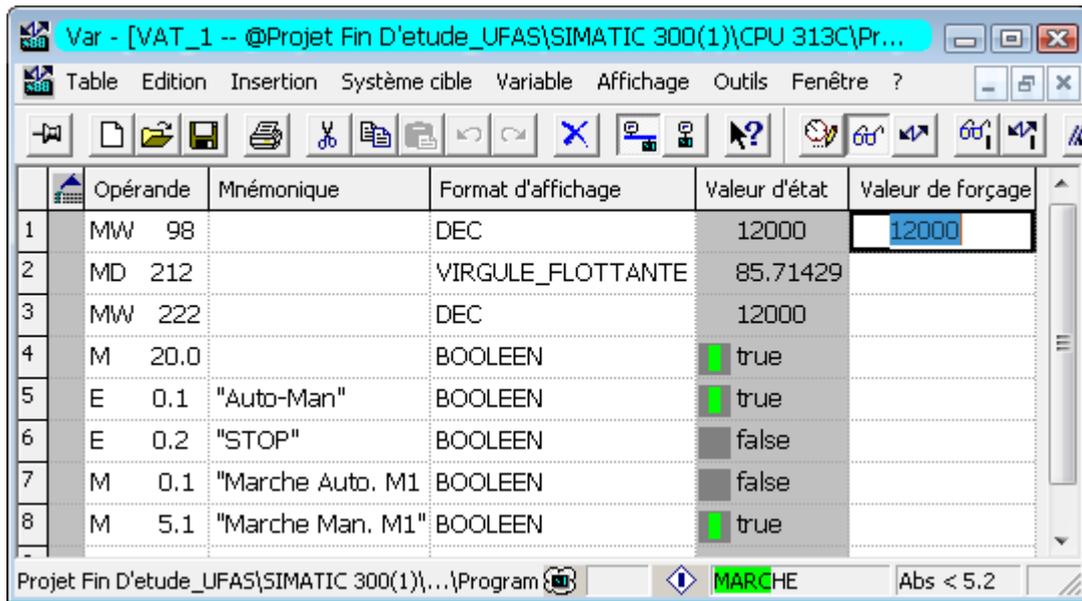
- **SF** nous signale que la CPU a détecté une erreur système qui provoque un changement de l'état de fonctionnement.
- **DP** indique l'état de la communication avec les entrées/sorties décentralisées
- **DC** (alimentation) indique si la CPU se trouve sous ou hors tension.
- **RUN** indique que la CPU est en marche.
- **STOP** indique que la CPU est en arrêt.

Remarque : Le bouton MRES (*Memory Reset*) permet d'effacer les blocs et la configuration matérielle de la station déjà chargée dans la CPU simulée et mettre toutes les zones mémoires à zéro. Alors pour une autre utilisation de cette CPU, il faut d'abord recharger la station.

3.12.5. Utilisation des tables de variables pour visualiser ou forcer des données

STEP 7 met à la disposition de l'utilisateur des tables de variables (VAT) permettant de visualiser l'état d'une variable quelconque dans le programme utilisateur. Il donne également la possibilité de forcer les variables définies dans cette table. Pour la création de cette table, on procède de la manière suivante :

1. Activer la fenêtre du gestionnaire de projets SIMATIC.
2. Naviguez dans SIMATIC Manager jusqu'à bloc
3. Effectuer un double clic sur VAT_1 pour ouvrir la table de variables du projet.
4. Pour établir une liaison en ligne avec le programme dans l'application de simulation, choisir la commande **Système cible** puis **Etablir la liaison à une CPU configurée**.
5. Choisir la commande **Variable** puis **Visualiser** pour lancer la visualisation de l'état des données dans le programme.
6. Pour forcer une variable, on clique sur le bouton gauche de la souris dans la zone **valeur de forçage** de la variable considérée, on entre la valeur de forçage appropriée au type de données et on active ces valeurs en cliquant sur le bouton de forçage des variables.



	Opérande	Mnémonique	Format d'affichage	Valeur d'état	Valeur de forçage
1	MW 98		DEC	12000	12000
2	MD 212		VIRGULE_FLOTTANTE	85.71429	
3	MW 222		DEC	12000	
4	M 20.0		BOOLEEN	<input checked="" type="checkbox"/> true	
5	E 0.1	"Auto-Man"	BOOLEEN	<input checked="" type="checkbox"/> true	
6	E 0.2	"STOP"	BOOLEEN	<input type="checkbox"/> false	
7	M 0.1	"Marche Auto. M1"	BOOLEEN	<input type="checkbox"/> false	
8	M 5.1	"Marche Man. M1"	BOOLEEN	<input checked="" type="checkbox"/> true	

Figure 3.13. Table des variables

3.13. Exemple de programmation

A fin d'illustrer notre travail, nous donnons quelques exemples sur notre programme élaboré pour l'automatisation du groupe.

3.13.1. Gestion des vérins

Dans notre système on a deux vérins pneumatiques a commandé, le premier (V1) situé a l'entrée ouvrira les soupapes de remplissage quand un détecteur photocellule détectera la présence des bouteilles. Le deuxième (V2) dégagera le reste de produit dans les canules pour éviter la formation de la mousse en cours de remplissage, ce vérin s'ouvrira après 2s de la mise en marche du vérin V1.

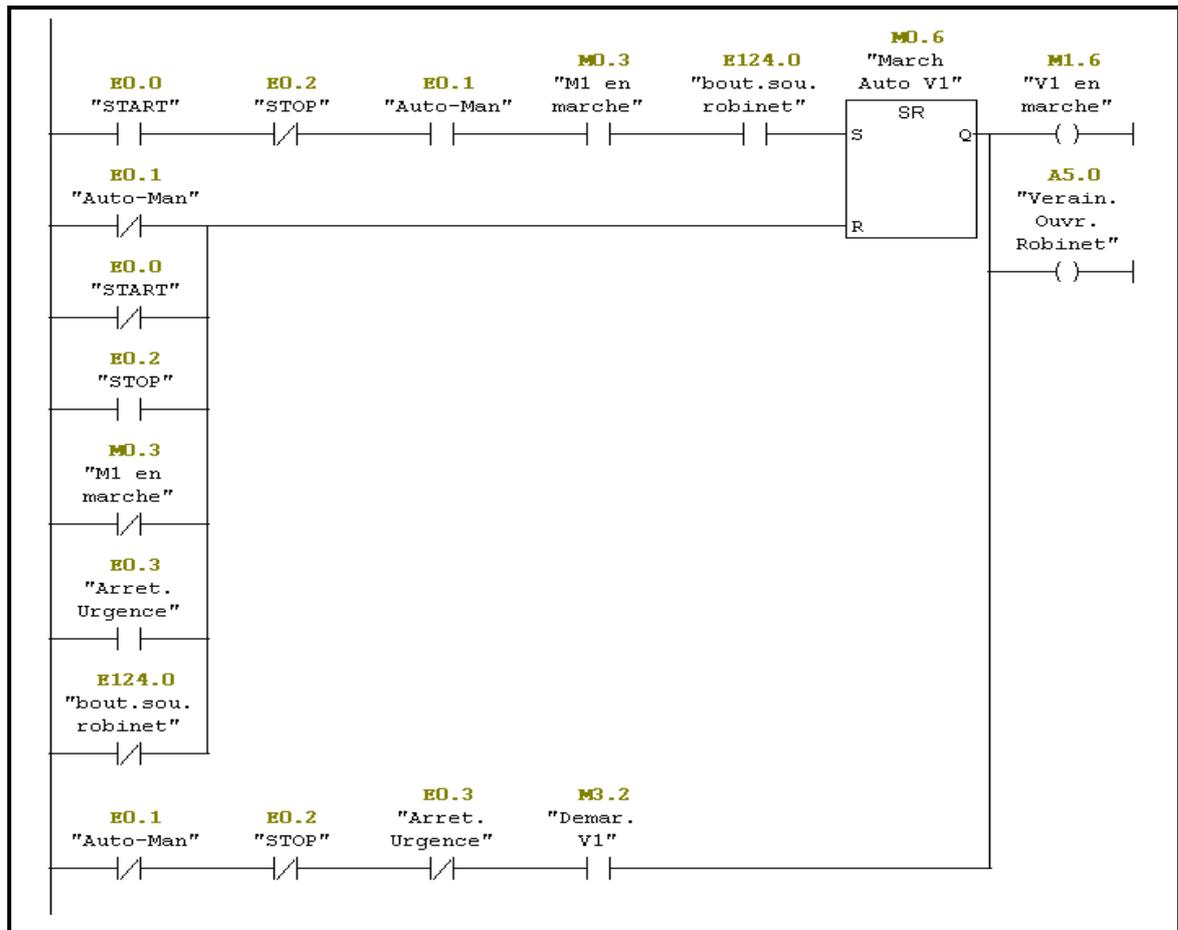


Figure 3.14. Gestion de marche/arrêt du vérin d'ouverture des robinets (V1) en CONT

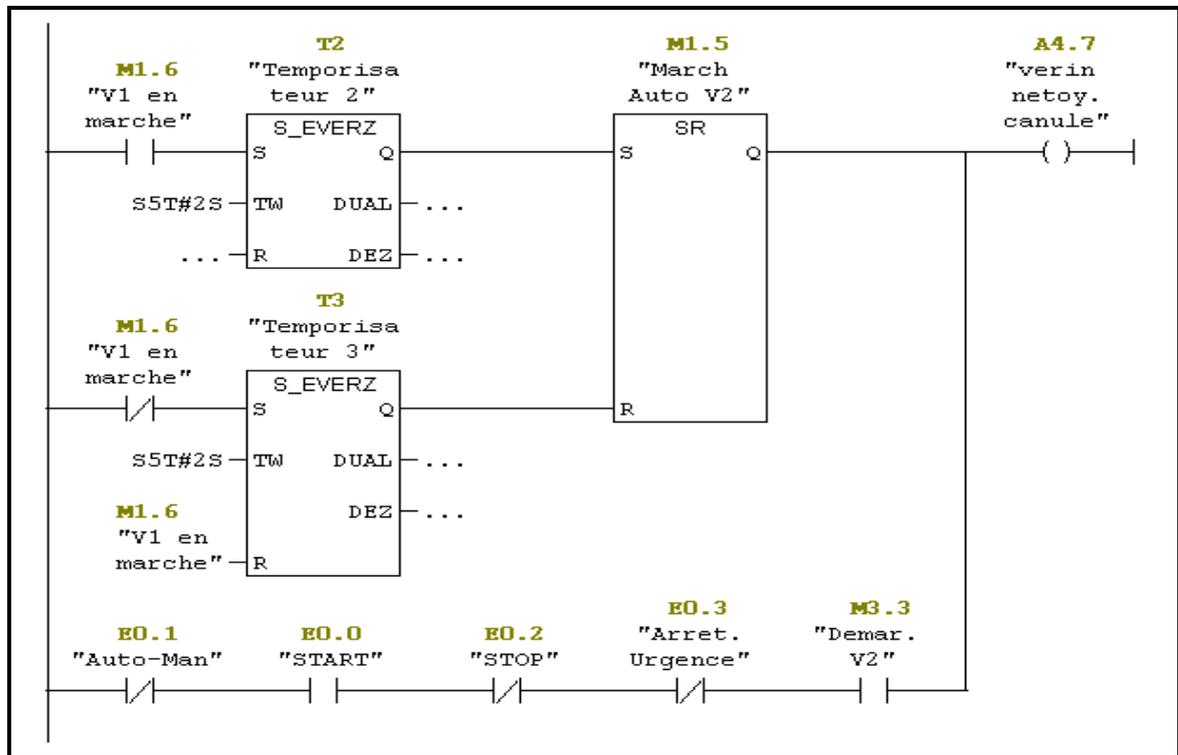


Figure 3.15. Gestion de marche/arrêt du vérin de nettoyage des canules (V2) en CONT

3.13.2. L'élévateur des capsules

Le niveau des capsules à l'intérieur de la trémie est contrôlé par un détecteur de proximité inductif, ce détecteur mettra en marche automatiquement l'élévateur de capsules pour que la trémie récupère le niveau correct de capsules à l'intérieur de celle-ci.

```

Réseau 1 : Titre :
O(
U      "START"                EO.0
U      "Auto-Man"             EO.1
UN     "STOP"                 EO.2
UN     "Arret.Urgence"        EO.3
UN     "Presence Bouch.Tremie" EO.7
S      "Marche arret elev.bouch" MO.7
U(
ON     "Auto-Man"             EO.1
O      "STOP"                 EO.2
O      "Arret.Urgence"        EO.3
O      "recup. Bouch. Tremie" M2.3
)
R      "Marche arret elev.bouch" MO.7
U      "Marche arret elev.bouch" MO.7
)
O
UN     "Auto-Man"             EO.1
UN     "STOP"                 EO.2
UN     "Arret.Urgence"        EO.3
U      "Demar.Elev. Bouchon"  M3.4
=      "Elev.Bouchon"         A4.6

Réseau 2 : Titre :
U      "Presence Bouch.Tremie" EO.7
L      S5T#10S
SE     T      16
NOP    O
NOP    O
NOP    O
U      T      16
=      "recup. Bouch. Tremie" M2.3

```

Figure 3.16. Gestion de marche/arrêt de l'élévateur des capsules en LIST.

3.13.3. Le moteur secondaire

C'est le moteur qui entraine la trémie ce qui assure le passage des capsules vers la capsuleuse à travers le canal descendant

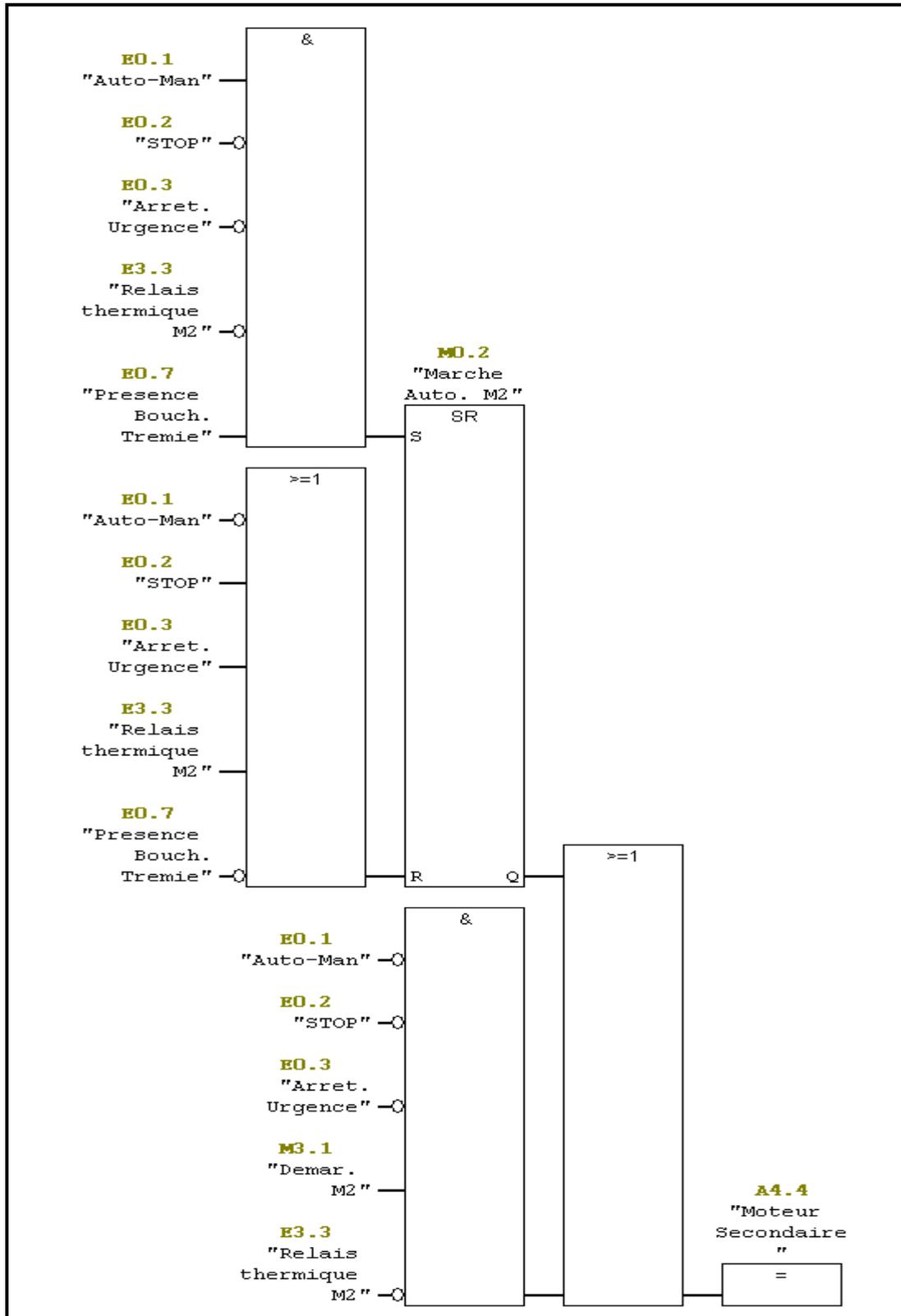


Figure 3.17. Gestion de marche/arrêt du moteur secondaire en LOG.

3.13.4. Exemple de simulation

On passe maintenant à la simulation d'un programme à l'aide du simulateur PLCSIM, en prend comme un exemple le fonctionnement du moteur secondaire en mode automatique, en mode manuel et en arrêt :

a. Condition de marche en mode automatique :

- La sélection de mode automatique est activée.
- Le bouton 'STOP' est désactivé
- Le bouton d'arrêt d'urgence est désactivé
- Le relais thermique du moteur est désactivé
- Il existe des capsules au niveau de la trémie.

b. Condition d'arrêt du moteur:

- Le bouton 'STOP' est activé
- Le bouton d'arrêt d'urgence est activé
- Le relais thermique du moteur est activé.
- Manque des capsules au niveau de la trémie.

c. Condition de marche en mode manuel :

- La sélection de mode manuel est activée
- Le bouton 'STOP' est activé
- La commande 'Demarrer M1' est activée
- Le bouton d'arrêt d'urgence est désactivé
- Le relais thermique du moteur est activé

3.13.4.2. La simulation de fonctionnement en mode automatique

Il faut que toutes les conditions de marche en mode automatique soient satisfaites

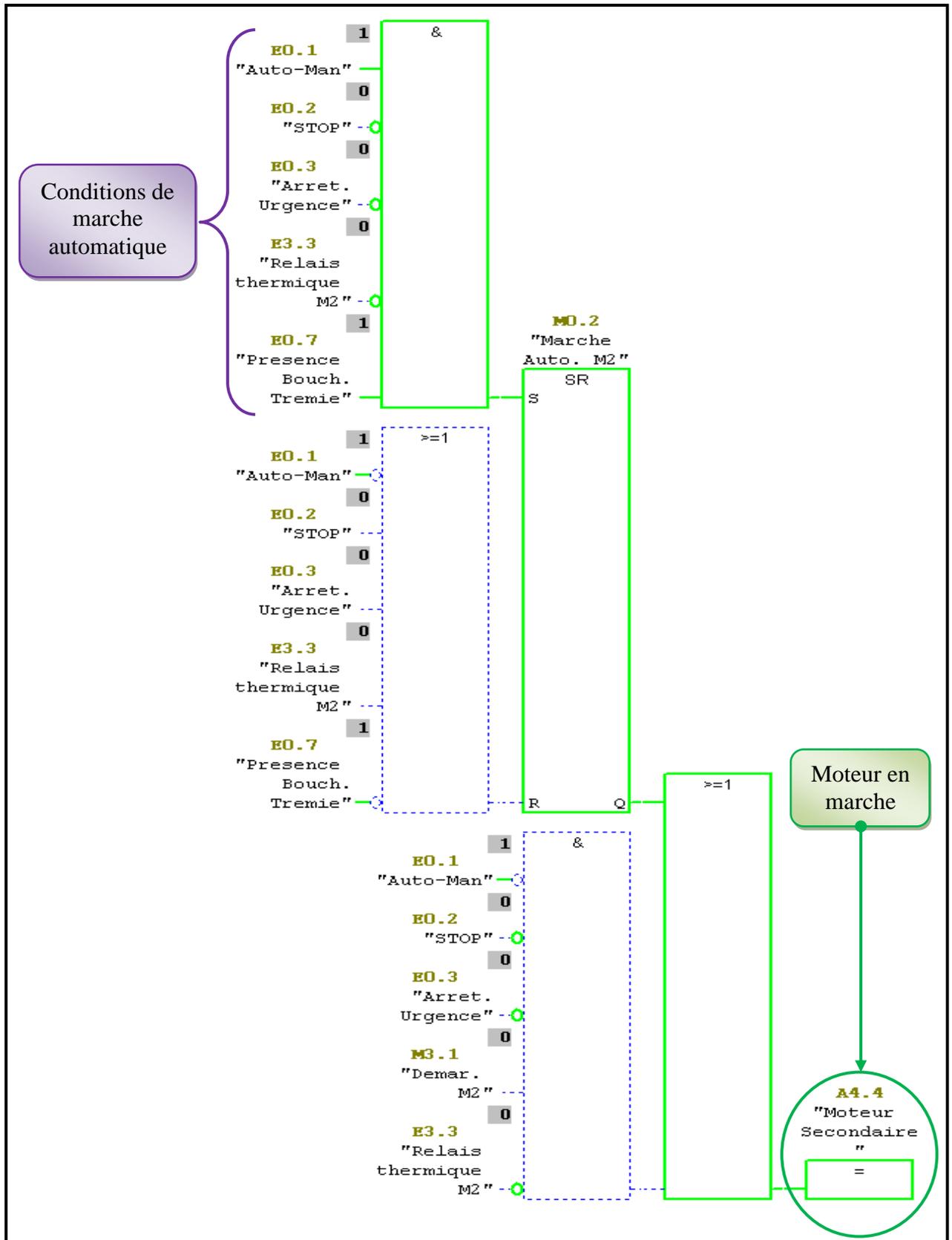


Figure 3.18. Simulation de fonctionnement en mode automatique du moteur secondaire

3.13.4.3. La simulation d'arrêt du moteur

Il suffit de satisfaire une des conditions d'arrêt comme par exemple s'il n'y a pas des capsules dans la trémie, le moteur s'arrêta immédiatement:

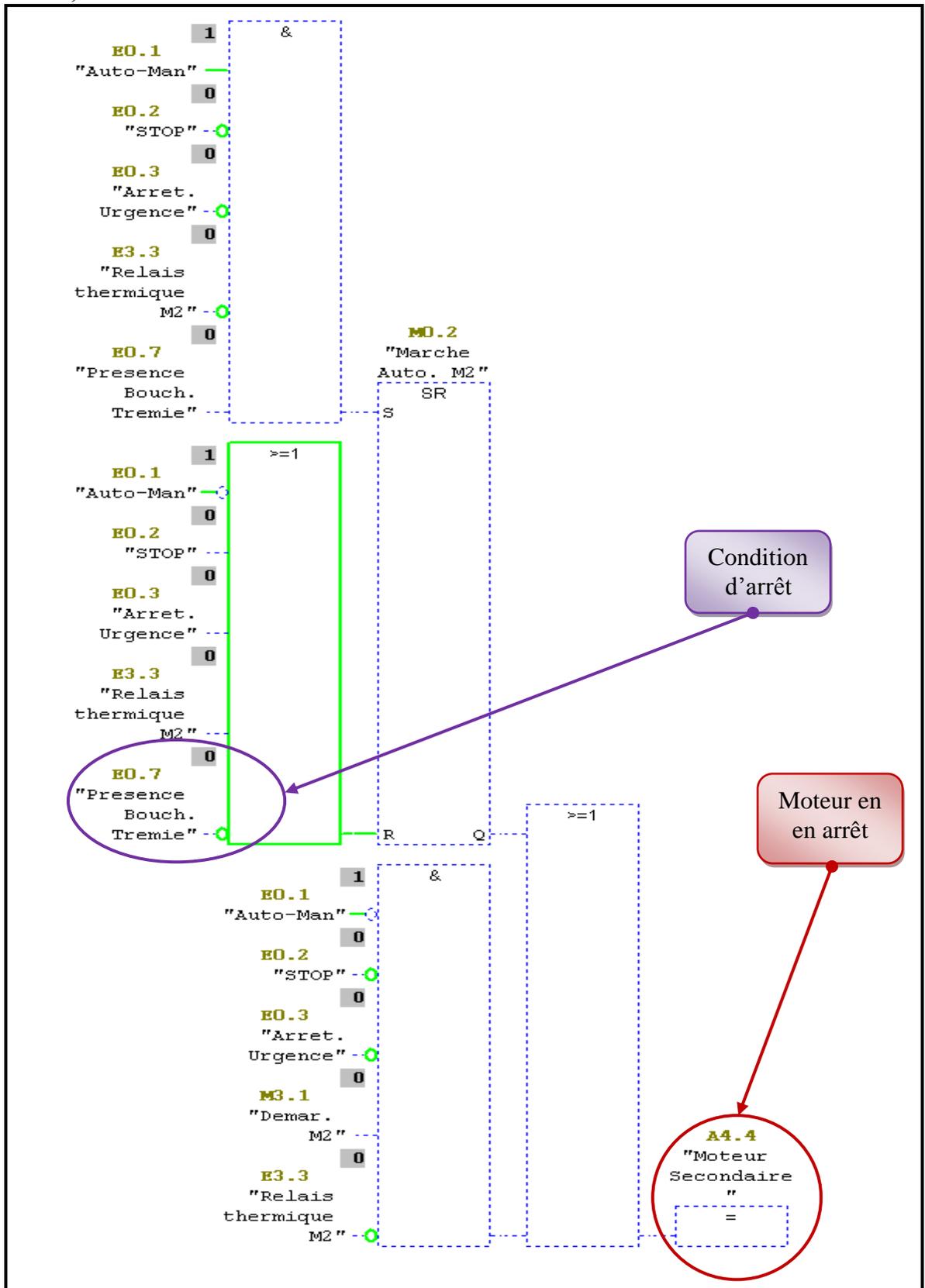


Figure 3.19. Simulation d'arrêt du moteur secondaire

3.13.4.4. La simulation de fonctionnement en mode manuel

Il faut que toutes les conditions de marche en mode manuel soient satisfaites :

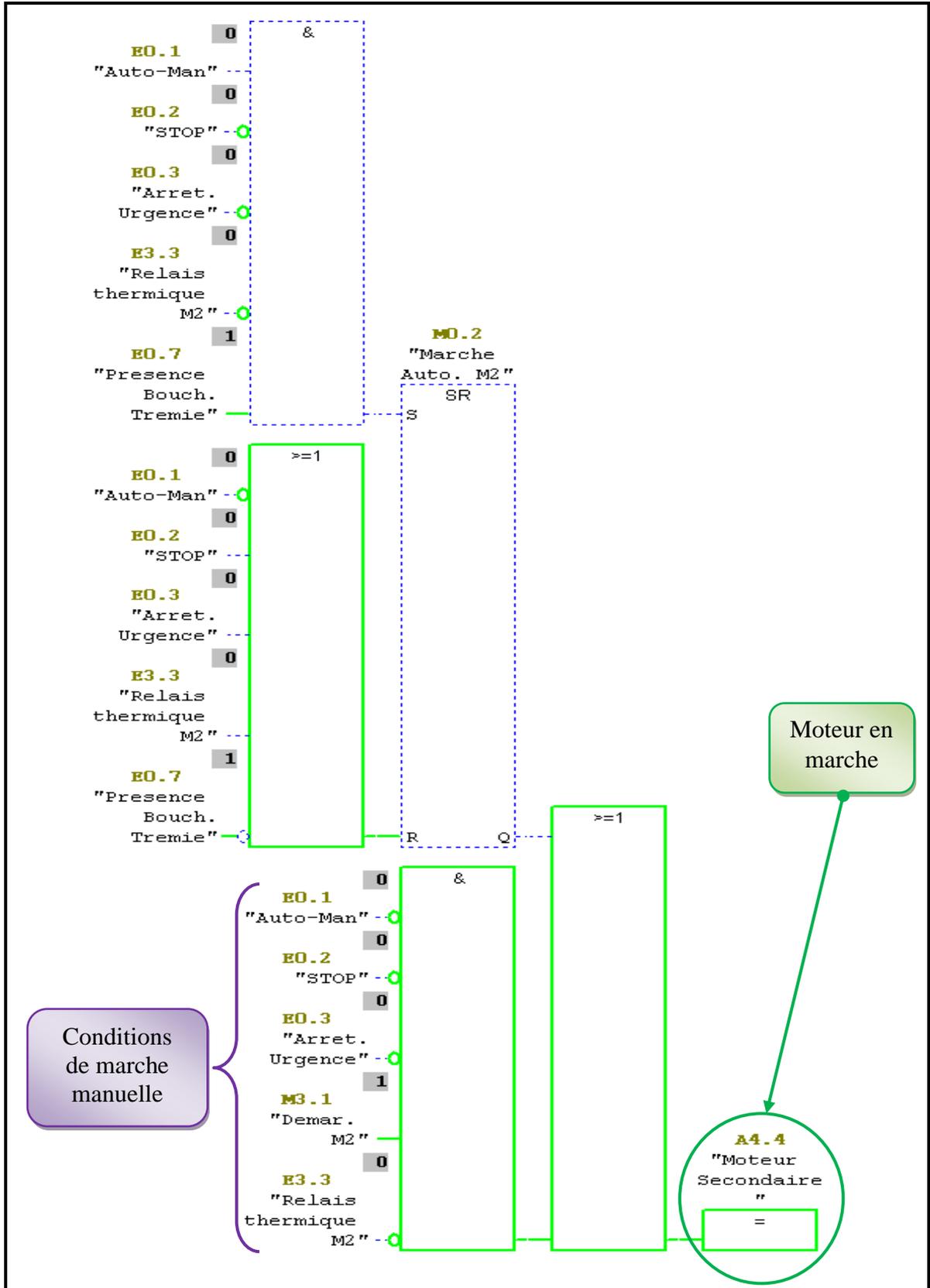


Figure 3.20. Simulation de fonctionnement en mode manuel du moteur secondaire.

3.14. La supervision

3.14.1. Description du logiciel SIMATIC WinCC

SIMATIC WinCC flexible est un logiciel HMI innovant sous Windows qui offre des fonctions performantes de supervision et de surveillance des systèmes automatisés, il convient pour toutes les applications au pied de la machine dans le domaine de la construction et de l'installation.

WinCC flexible permet de disposer d'un logiciel d'ingénierie pour tous les pupitres opérateur SIMATIC HMI, du plus petit Micro Panel jusqu'au Multi Panel ainsi que d'un logiciel de supervision runtime pour solutions monoposte basées sur PC et tournant sous Windows.

WinCC flexible apporte une efficacité de configuration maximale avec des bibliothèques contenant des objets préconfigurés et des blocs d'affichage réutilisables [14].

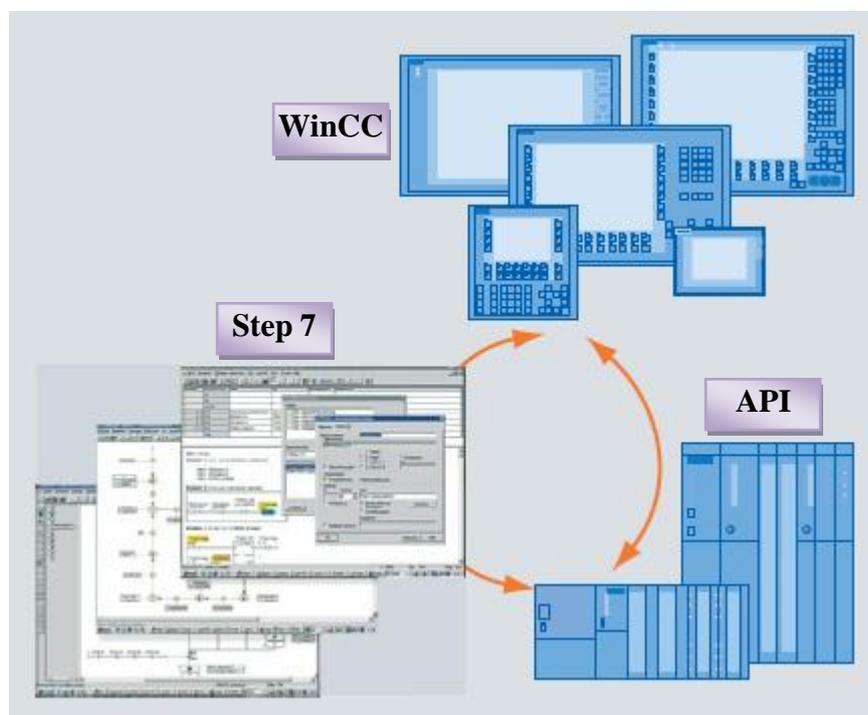


Figure 3.21. Diagnostic des APIs avec WinCC et les outils de programmation Step 7

3.14.2. Pour quoi le SIMATIC WinCC

SIMATIC WinCC est aujourd'hui considéré comme le numéro un mondial de l'automatisation. Ceci s'explique dans une large mesure par le fait qu'il possède les propriétés suivantes :

- Visualisation entièrement graphique des processus et des états des processus.
- Conduite de la machine ou de l'installation par le biais d'une interface utilisateur personnalisable avec ses propres menus et barres d'outils.

- Signalisation d'événements.
- Archivage des valeurs de mesure et des messages dans une base de données processus.
- Gestion des utilisateurs ainsi que de leurs droits d'accès.
- Visibilité grâce à une liste de références croisées.

3.14.3. L'explorateur de WinCC

C'est l'interface qui permet à l'utilisateur de choisir une des opérations supportables sur SIMATIC WinCC, ces opérations sont résumées comme suit :

- Ouvrir les projets les plus récents.
- Créer étape par étape un projet WinCC avec l'assistant de projet qui donne la main à l'utilisateur dans chaque étape de choisir la configuration souhaité (API, type de liaison, le programme step7, le pupitre ...).
- Ouvrir un projet déjà créé.
- Créer un projet WinCC sans configuration.
- Ouvrir un projet ProTool c'est à dire la migration WinCC- ProTool.

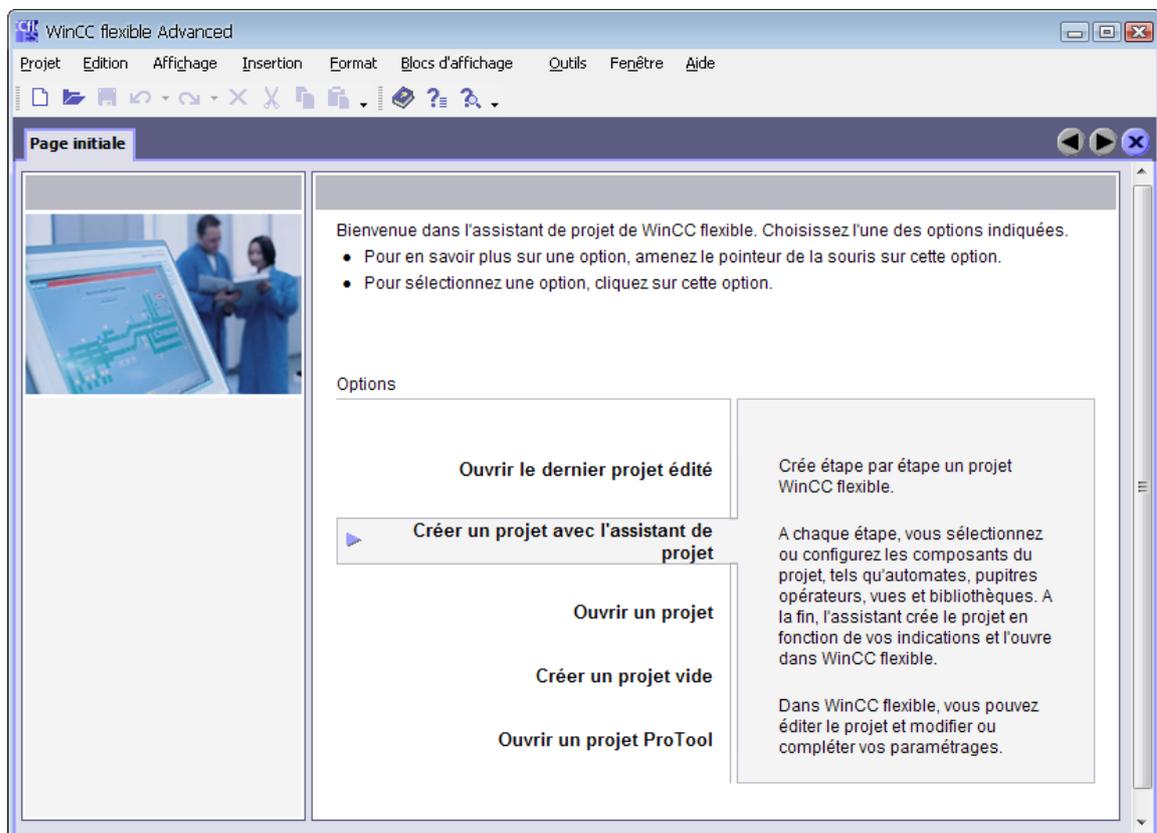


Figure 3.22. Interface de dialogue WinCC

3.14.4. La configuration de la communication réseau

Le gestionnaire ProNet nous permet de configurer le réseau de communication utilisé afin de mettre en liaison l'automate programmable S7-300 avec le serveur de supervision (pupitre opérateur). On a choisit le réseau MPI comme réseau de communication

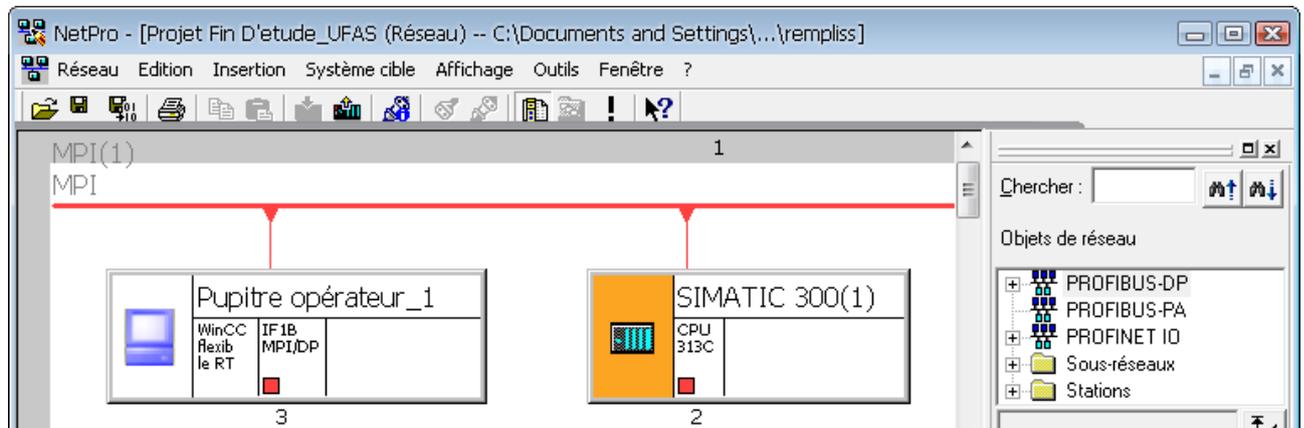


Figure 3.23. Communication réseaux (Net Pro)

3.14.4.1. Le réseau communication MPI

L'interface multipoint (MPI) est la solution économique pour la communication avec des PG/PC, des systèmes HMI ainsi que d'autres systèmes d'automatisation comme SIMATIC S7, afin d'assuré l'échange des données entre différents composants. L'interface MPI autorise la connexion d'un maximum de 125 abonnés MPI, avec des vitesses de transmission jusqu'à 19,2 kbit/s.

3.14.5. L'éditeur graphique

Il dispose la zone de travail principal pour la création et la dynamisation de différentes vues reliées à l'état du processus. Il rassemble les différentes opérations qui permettent d'élaborer l'interface homme-machine désirée.

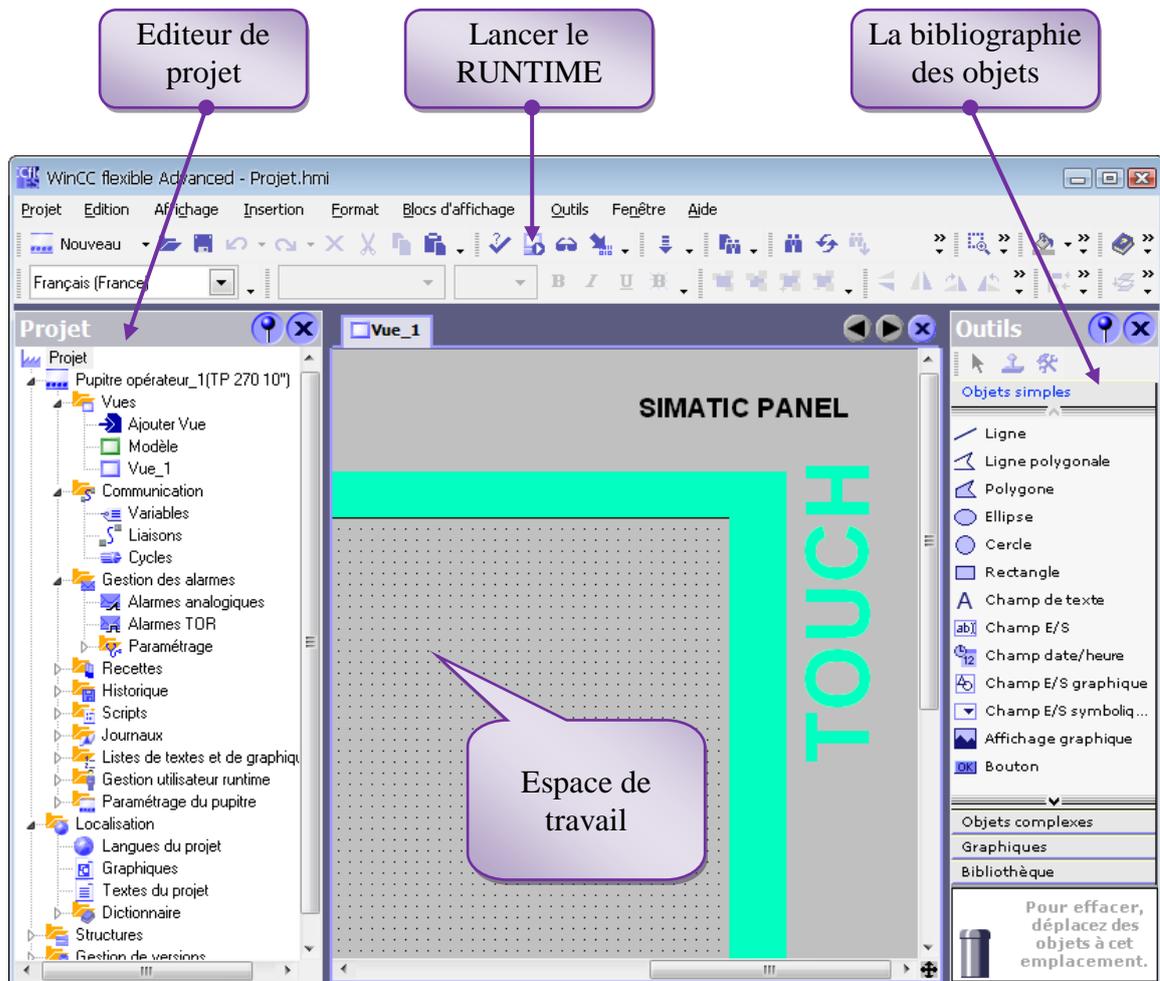


Figure 3.24. L'éditeur graphique

3.14.6. Lancement de la supervision

La conduite et la surveillance du processus en temps réel se font à l'aide de l'application RUNTIME, cette application permet d'assurer la réalisation de différentes tâches nécessaires pour la supervision visée comme :

- Lecture des données.
- Affichage des vues à l'écran.
- Communication avec les automates programmables.

3.14.7. Supervision du groupe en arrêt

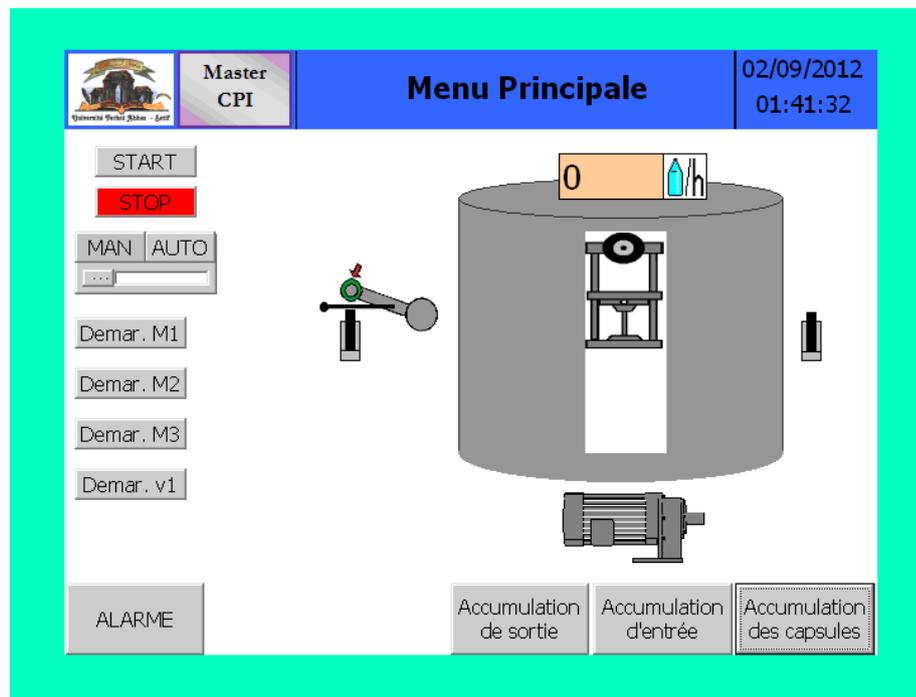


Figure 3.25. Supervision du groupe en arrêt

3.14.8. Supervision du groupe en marche

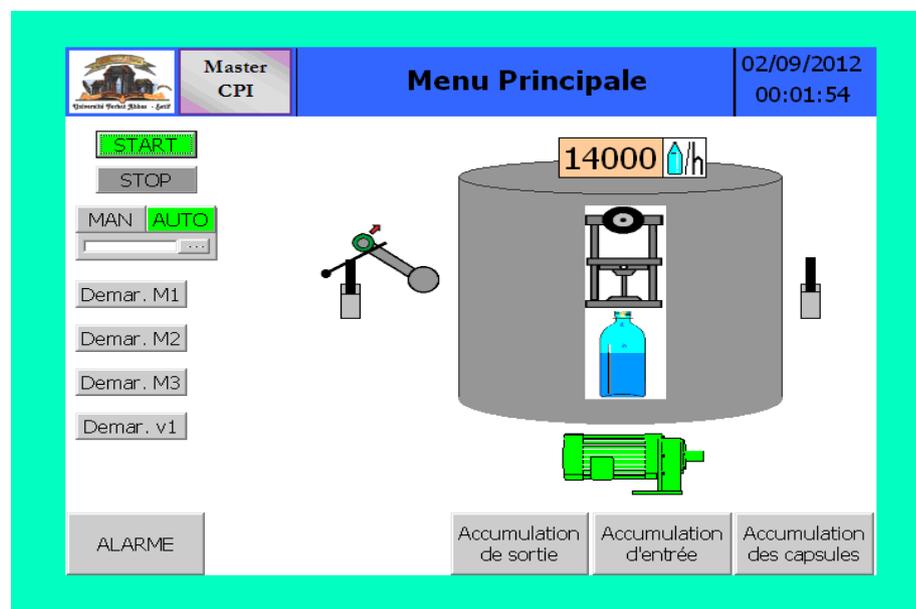


Figure 3.26. Menu principal de supervision

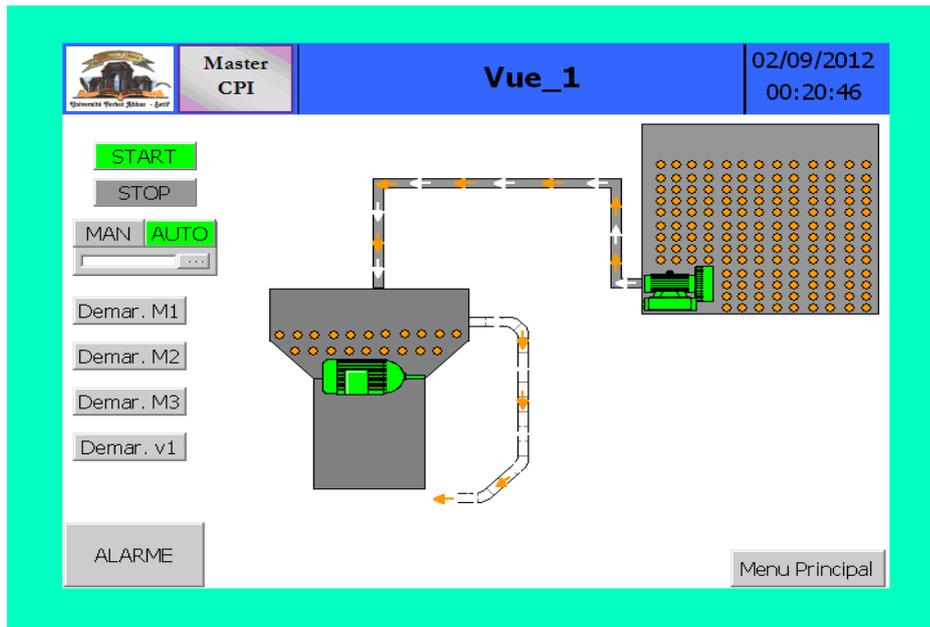


Figure 3.27. Supervision d'accumulation des capsules

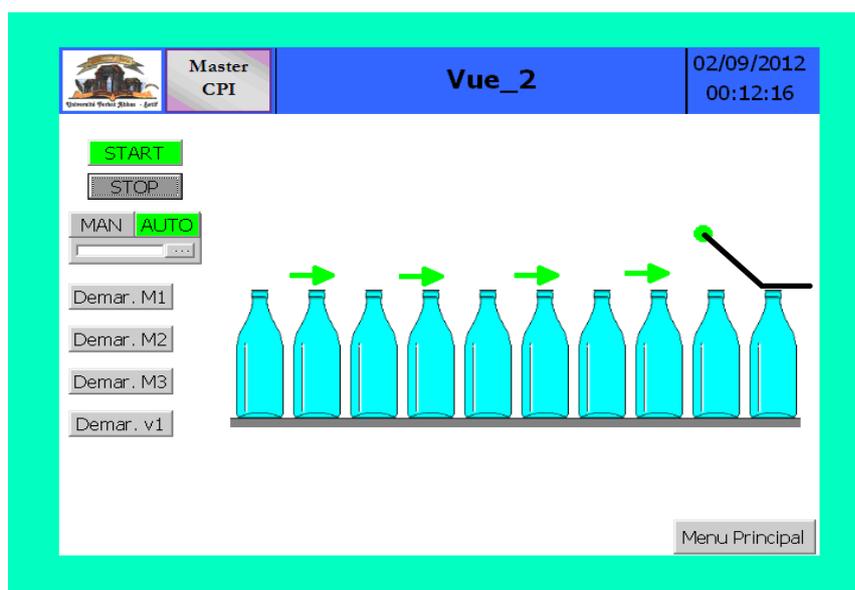


Figure 3.28. Supervision d'accumulation des bouteilles à l'entrée

3.14.9. Supervision du groupe avec défaut

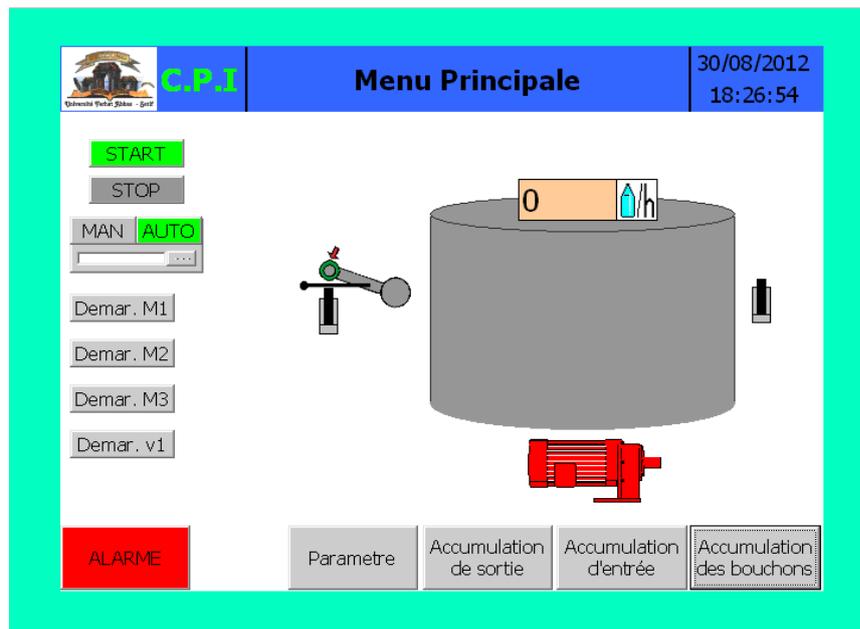


Figure 3.29. Supervision d'un défaut au niveau de M1

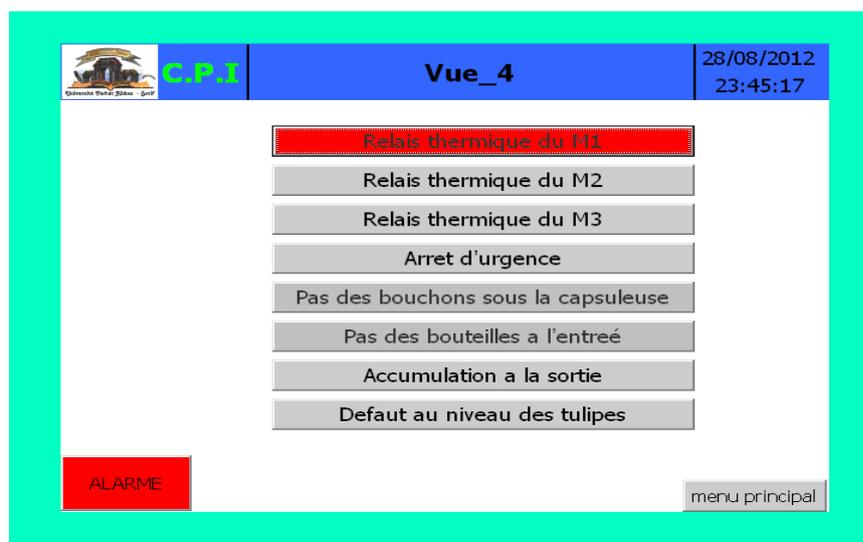


Figure 3.30. Liste des alarmes.

3.15. Conclusion

Dans ce chapitre nous avons présenté les étapes nécessaires pour la création d'un programme avec SIMATIC step7, puis on a présenté quelques exemples sur notre programme, pour tester ce programme afin de corriger d'éventuelles erreurs, le logiciel S7-PLCSIM a été utilisé, en fin on a utilisé le logiciel SIMATIC WinCC pour élaborer une interface de supervision afin d'évoluer l'état du groupe et localiser les pannes ou les défauts possibles.

CONCLUSIONS GENERALES ET PERSPECTIVES

Le travail présenté s'inscrit dans le cadre de notre projet de fin d'études pour l'automatisation d'un groupe de remplissage des boissons gazeuses au sein de l'entreprise SPA MAMI de Sétif, ce travail de caractère grossièrement pratique nous a permis de découvrir le monde de l'industrie et de participer au renforcement des relations entre l'université et les entreprises industrielles.

Cette expérience est une occasion d'appliquer nos connaissances acquises durant notre formation au sein de la filière Commande des Processus Industriels, elle nous a permis aussi d'acquérir d'autres connaissances dans le domaine de la pratique, de tirer profit de l'expérience des personnes du domaine et d'apprendre une méthodologie rationnelle à suivre pour l'élaboration des projets d'automatisation.

Pour automatiser le groupe de remplissage, nous avons travaillé sur l'automate S7-300 de la gamme SIEMENS, qui a été notre choix pour l'application visée. Celui-ci qui se manifeste essentiellement dans :

- Performances techniques (fréquence de traitement, type de signaux traités ... etc.).
- Capacité mémoire suffisante.
- Possibilité d'élever le nombre des entrées et des sorties par des extensions.
- Possibilité d'intégrer des modules spéciaux (modules d'entrées ou de sorties analogiques, régulateur de température, de pression, ...etc.).

Concernant la programmation, on a utilisé le logiciel STEP7 qui nous a permis une très grande souplesse avec la possibilité de visualiser et de tester les états des programmes à l'aide du simulateur PLCSIM.

Grace à la présentation graphique offerte par le logiciel WinCC, on a pu suivre l'évolution de l'état du processus, basé sur le déroulement du programme STEP7, ainsi que la localisation de la panne ou du défaut possible.

Finalement, nous espérons que notre travail sera un apport pour les promotions à venir.

Références Bibliographiques

- [1] Professeur M. TAHIRI, *Support de Cours sur les Automatismes et l'Automatisation*, Rabat (Maroc).
 - [2] L. Bergougnoux, *Cours : Automates Programmables Industriels*, d'département de Mécanique énergétique, POLYTECH Marseille, 2005.
 - [3] Génie électrique électrotechnique automatisme [en ligne]. Disponible sur le lien : <http://www.geea.org/IMG/pdf/>
 - [4] G. Michel, *Architecture et Application des Automates Programmables Industriels*, édition DUNOD, Paris, 1987
 - [5] Alain GONZAGA, *LES AUTOMATES PROGRAMMABLES INDUSTRIELS*, 2008, page 5
 - [6] William BOLTON, *Les automates programmables industriels*, 2010, page18
 - [7] www.SIEMENS.com
 - [8] www.SPAMAMI.com
 - [9] L'architecture de la chaine de production livrée par l'entreprise SPA MAMI, fichier AUTOCAD
 - [10] Manuel de référence de la remplisseuse MONOBLOC livrée par l'entreprise SPA MAMI
 - [11] Manuel SIEMENS, *Programmer avec STEP 7*, n° de référence : A5E02789667-01, 05/2010
 - [12] P. Jargot, *Langages de Programmation pour API. Norme IEC 1131-3-Techniques de l'Ingénieur*, S8030, 2006
 - [13] Manuel SIEMENS, *S7-PLCSIM - Tester les programmes S7*, n° de référence : A5E00992426-02, 03/2009
 - [14] Brochure SIMATIC WinCC flexible, N° de référence : 6ZB5370-1CD03-0BB0, mars 2010.
-

ملخص:

الهدف من هذا العمل هو التشغيل الآلي والإشراف على مراقبة عمل وحدة تعبئة المشروبات الغازية الخاصة بشركة MAMI وهذا باستخدام S7-300 من شركة سيمنس لتعزيز الإنتاجية وتحسين ظروف عمل الموظفين نظرا لأن لوضع التشغيل الحالي للوحدة هو شبه التلقائي مما يتسبب في انخفاض كبير في الإنتاج ويخلق صعوبات في استيفاء وقت التنفيذ. لتحقيق الهدف المنشود، فمننا بإعداد برنامج باستخدام SIMATIC STEP7 كما قمنا ببناء واجهة لمراقبة عمل الوحدة باستخدام برنامج SIMATIC WinCC استنادا على البرنامج المطور.

كلمات مفتاحية: أنظمة آلية ، سيمنس S7-300 ، STEP7 ، WinCC

Résumé :

L'objectif de ce travail est l'automatisation et la supervision de fonctionnement d'un groupe de remplissage des boissons gazeuses au sein de SPA MAMI en utilisant l'automate programmable S7-300 de Siemens afin d'augmenter la productivité et d'améliorer les conditions de travail des personnels car le mode de marche actuel du groupe est semi-automatique ce qui entraîne une baisse substantielle de la production et crée des difficultés pour respecter les temps d'exécution des opérations. Pour atteindre l'objectif désiré, nous avons élaboré un programme en utilisant le logiciel SIMATIC STEP7 et nous avons construit une interface-homme machine grâce au logiciel SIMATIC WinCC a base du programme élaboré.

Mots Clés : Systèmes automatisés, API, Step7, WinCC.

Abstract:

The objective of this work is the automation and supervision of functioning of a group used for filling carbonated drinks in SPA MAMI using the PLC S7-300 of Siemens in order to increase the productivity and improve working conditions of staff because the current operating mode of the group is semi-automatic which causes a substantial decrease in production and creates difficulties in complying with time execution. To achieve the desired objective, we have developed a program using the SIMATIC STEP7 and we have built a man-machine interface with the software SIMATIC WinCC based on the program developed.

Key Words : Automation systems, PLC, Step 7, WinCC.