

**REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE**  
**MINISTRE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA**  
**RECHERCHE SCIENTIFIQUE**  
**UNIVERSITÉ MOHAMED SEDDIK BEN YAHIA - JIJEL**



**Faculté des Sciences et de la Technologie**  
**Département d'Automatique**  
**PROJET DE FIN D'ETUDE POUR L'OBTENTION DU DIPLOME**  
**MASTER EN AUTOMATIQUE**  
**OPTION : Automatique et Informatique industrielle**

## **Thème**

**Automatisation et supervision de deux systèmes  
industriels virtuels**

**Réalisé par :**

BOUSSADI Salim

BENDAOUY Yasser

**Encadrant :**

OUCIEF Nabil

**Co-encadrant :**

LABIOD Salim

*Année Universitaire 2020-2021*

## *Remerciements*

*Avant tout nous remercions Dieu Le tout puissant de nous avoir donné le courage, la volonté, la patience, et la santé durant toutes ces années et que grâce à lui ce travail a pu être réalisé et qui nous accordé la chance de vivre ce jour pour voir le fruit de nos études.*

*Nous tenons également à exprimer nos vifs remerciements à messieurs **Salim labiod** et **Nabil Oucief** qui nous ont proposé ce thème, et qui nous ont encadré et soutenu par leurs conseils, comprenions et encouragements.*

*Nous tenons aussi à présenter nos remerciements aux **membres de jury** qui ont accepté de juger notre modeste travail.*

*Tous les enseignants qui ont contribué de près ou de loin à notre formation sans oublier nos amis et collègues qui directement ou indirectement ont contribué à la réalisation de ce travail.*

*Mille remerciements.*

*Salim et Yasser*

# *Dédicaces*

*Chaleureusement je dédie ce modeste travail :*

*A la lumière de ma vie, mes chers parents en témoignage de leur amour et de leur sacrifices sans limites, à mon égard, je leurs souhaite une bonne santé, que dieu me les garde.*

*A mes frères et mes sœurs,  
Pour ses soutiens moraux et leurs conseils précieux tout au long de mes études.*

*A toute la famille Bousadi*

*A tous mes amis plus particulièrement:*

*A Mohamed Cherif. L  
A Amir. R*

*A tous ceux qui m'ont aidé de près ou loin pour la réalisation de ce modeste travail.*

*Salim*

## ***Dédicace***

*Je dédie ce mémoire :*

*A ma très chère mère qui a été la lumière de ma vie*

*A mes sœurs et mon frère qui m'ont encouragé et soutenu moralement.*

*A toute ma famille*

*A mon binôme, S. BOUSADI et toute sa famille*

*A mes Amis: YOUCEF, SIDI MOHAMED, HEITEM, ABD SABOUR, S.HAROUZ, HATEM... Etc.*

*A mes enseignants durant toutes les années d'études jusqu'au MASTER*

**YASSER**

## الملخص

يشكل التشغيل الآلي للنظم الصناعية أسلوباً هاماً للقدررة التنافسية بين الشركات لأنه يكفل إنتاجاً أفضل وييسر المهام أو الخدمات والرصد.

قد برمجنا في هذا العمل نظامي البرمجيات **ITS PLC** (نظام فرز افتراضي ، خليط افتراضي) كفاتنه آلية صناعية قابلة للبرمجة على وجه الخصوص **siemens S7-300** مدمجة في التطبيق **STEP 7** من **siemens** ومحاكاة على وحدة المعالجة المركزية الافتراضية مع بعض الإشراف من قبل واجهة بين الإنسان والآلة التي تسمح لنا بمراقبة و التحكم في النظام باستخدام برمجيات المرنة . ويتيح هذا العمل تطوير قدرات استخدام برمجيات محاكاة النظم الآلية والاختصاص في مجال برمجة الصناعية. وفي نهاية المطاف ، تعرض نتائج المحاة لنظم.

**الكلمات المفتاحية :** التشغيل لآلي، وحدة المعالجة المركزية الافتراضية **Siemens,S7-300, step7, wincc** , **flexible , ITS PLC, PLC SIM**

## Résumé

L'automatisation des systèmes industriel est une technique important pour la compétitivité entre les entreprises car elle assure une meilleur production et faciliter les taches ou les services et la surveillance.

Dans ce travail nous avons programmé les deux systèmes de logiciel ITS PLC (système de tri virtuel, mélangeur virtuel) assurée par un automate programmable industriel (API) en particulier le siemens S7-300 intégré à STEP 7 de siemens et simulé sur PLCSIM (CPU virtuelle) avec une partie de supervision par une interface homme-machine qui nous permet de Controller et commander le système à l'aide de logiciel wincc flexible. Ce travail permet de développer les capacités de l'utilisation des logiciels de simulation des systèmes automatisés et de la compétence dans le domaine de programmation des systèmes industriel. A la fin les résultats de simulation sont présentés.

**Mots clé :** Automatiser, API, Siemens, S7-300, step7, wincc flexible, ITS PLC, PLC SIM

## **Summary**

In this work we have programmed the two software systems ITS PLC (virtual sorting system, virtual mixer) ensured by an industrial programmable automaton (API) in particular the siemens S7-300 integrated in STEP 7 of siemens and simulated on PLCSIM (virtual CPU) with some supervision by a man-machine interface that allows us to Controller and control the system using flexible wincc software. This work allows to develop the capabilities of the use of simulation software of automated systems and competence in the field of programming of industrial systems. At the end the simulation results are presented.

**Key words:** Automate, API, Siemens, S7-300, step7, wincc flexible, ITS PLC, PLC SIM.

# Table des matières

<b>Table des matières</b> .....	<b>1</b>
<b>Table des figures</b> .....	<b>4</b>
<b>Liste des tableaux</b> .....	<b>7</b>
<b>Introduction générale</b> .....	<b>8</b>
<b>Chapitre 1: Généralité sur l'automatisation et la supervision industrielle</b> .....	<b>10</b>
1.1 Introduction.....	10
1.2 Systèmes automatisés.....	10
1.2.1 Objectifs et atouts de l'automatisation .....	10
1.2.2 C'est quoi un système automatisé ?.....	11
1.2.3 Chaines fonctionnelles.....	11
1.2.4 Structure des systèmes automatisés .....	12
1.3 Généralités sur les automates programmables industriels (API) .....	14
1.3.1 Définition.....	14
1.3.2 Principes de fonctionnement d'un API.....	15
1.3.3 Architecture d'un API .....	15
1.4 Présentation de quelques API de la gamme SIMATIC S7 de Siemens .....	17
1.4.1 SIMATIC S7 .....	17
1.5 Choix de l'API.....	19
1.5.1 Présentation de l'API choisi le S7-300 .....	20
1.5.2 Structure de l'API S7-300 .....	20
1.6 Supervision .....	23
1.6.1 Avantages de la supervision.....	24
1.6.2 Constitution d'un système de supervision .....	24
1.7 Conclusion .....	25
<b>Chapitre 2: Logiciels STEP 7 et Wincc flexible</b> .....	<b>26</b>
2.1 Introduction .....	26
2.2 Description du logiciel Step 7 .....	26
2.2.1 Présentation de Step 7.....	26
2.2.2 Gestionnaire de projets SIMATIC Manager .....	27

---

2.2.3 Langages de programmation .....	27
2.2.4 Structure générale d'un programme Step 7.....	29
2.2.5 Éditeur des mnémoniques.....	30
2.2.6 Différents types des variables des programmes Step 7 .....	31
2.2.7 Création d'un projet .....	31
2.2.8 Barres de : titre, menu et outils de SIMATIC Manager.....	34
2.2.9 Simulateur de programme PLCSIM .....	34
2.2.10 Configuration matérielle .....	35
2.3 Le logiciel e supervision WinCC flexible .....	36
2.3.1 Tâches d'un système HMI.....	37
2.3.2 Étapes de création un projet WinCC .....	37
2.4 Conclusion .....	40
<b>Chapitre 3: Automatisation et supervision d'un système de tri virtuel .....</b>	<b>41</b>
3.1 Introduction .....	41
3.2 Description du logiciel ITS PLC .....	41
3.2.1 Interface du logiciel .....	41
3.2.2 Panneaux des Systèmes .....	43
3.2.3 Système de Navigation .....	44
3.3 Description du système de tri de caisses .....	47
3.4 Cahier des charges.....	51
3.5 Modélisation par Grafcet .....	52
3.5.1 Grafkets auxiliaires .....	52
3.5.2 Grafcet principal .....	55
3.6 Réalisation du programme.....	56
3.6.1 Interface du projet .....	56
3.6.2 Table des mnémoniques.....	56
3.6.3 Contenu de bloc fonctionnel FB1 .....	58
3.6.4 Contenu des fonctions FC (FC1, FC2) .....	63
Bloc d'organisation OB1 .....	65
3.6.5 Simulation du programme par PLC-SIM.....	65
3.7 Présentation de la partie HMI.....	66
3.7.1 Variables du HMI .....	66
3.7.2 Vue principale.....	68
3.8 Simulation.....	69

3.8.1 Simulation du programme sous WinCC flexible et PLCSIM .....	69
3.8.2 Simulation générale (avec les trois logiciels en parallèles).....	70
3.9 Conclusion .....	71
<b>Chapitre 4: Automatisation et supervision d'un mélangeur de peinture .....</b>	<b>72</b>
4.1 Introduction.....	72
4.2 Description du système.....	72
4.2.1 Structure du mélangeur .....	72
4.2.2 Partie opérative.....	73
4.3 Cahier des charges.....	75
4.4 Programmation .....	76
4.4.1 Modélisation par Grafcet .....	76
4.4.2 Interface de projet.....	76
4.4.3 Table des mnémoniques.....	78
4.4.4 Contenu du bloc fonctionnel FB1 .....	79
4.4.5 Contenu des fonctions FC (FC1, FC2, FC3, FC4) .....	81
4.5 Elaboration de l'HMI.....	83
4.5.1 Table des variables .....	83
4.5.2 Vue Principale du système.....	83
4.6 Conclusion .....	84
<b>Conclusion générale .....</b>	<b>85</b>
<b>Bibliographie.....</b>	<b>86</b>

## Table des figures

Figure 1.1 : Schéma simplifié d'un système automatisé.....	11
Figure 1.2 : Chaîne d'information.....	12
Figure 1.3 : Chaîne d'énergie.....	12
Figure 1.4 : Structure d'un système automatisé.....	13
Figure 1.5 : Dialogue entre la partie commande et la partie opérative.....	14
Figure 1.6 : Cycle de fonctionnement de l'API.....	15
Figure 1.7 : Structure interne d'un API.....	17
Figure 1.8 : API Siemens S7-200.....	17
Figure 1.9 : API Siemens S7-300.....	18
Figure 1.10 : API Siemens S7-400.....	18
Figure 1.11 : API Siemens S7-1200.....	19
Figure 1.12 : API Siemens S7-1200.....	19
Figure 1.13 : L'automate S7-300 et son environnement.....	20
Figure 1.14 : Les modules de l'API S7-300.....	21
Figure 1.15 : Module CPU de S7-300 (modèle CPU 315 2PN/DP).....	21
Figure 1.16 : Modules SM d'E/S TOR de S7-300 (de gauche à droite : module d'entrées TOR SM 321, module de sorties TOR SM322).....	22
Figure 1.17 : Modules SM d'E/S analogiques de S7-300 (de gauche à droite : module d'entrées analogiques SM331, module de sorties analogiques SM332).....	22
Figure 1.18 : Module de fonction (FM) de S7-300.....	23
Figure 1.19 : Pupitre de supervision du système de pompage des eaux traitées (station de traitement des eaux de Kissir).....	23
Figure 1.20 : Pupitre de supervision du système de remplissage des eaux traitées (station de traitement des eaux de Kissir).....	24
Figure 1.21 : Schéma synoptique d'un système de supervision.....	25
Figure 2.1 : Démarrage de SIMATIC Manager.....	27
Figure 2.2 : Exemple de programme en Graph S7.....	28
Figure 2.3 : Exemple de programme CONT.....	28
Figure 2.4 : Exemple de programme LOG.....	29
Figure 2.5 : Exemple de programme LOG.....	29
Figure 2.6 : Exemple de la structure générale d'un programme STEP7.....	30
Figure 2.7 : Table des mnémoniques.....	31
Figure 2.8 : Assistant de nouveau projet.....	32
Figure 2.9 : Choix de la CPU.....	32
Figure 2.10 : Sélection des blocs et choix du langage.....	33
Figure 2.11 : Nomination et création du projet.....	33
Figure 2.12 : Station SIMATIC S7-300.....	34
Figure 2.13 : Simulateur S7-PLCSIM.....	35
Figure 2.14 : Configuration du matériel.....	36
Figure 2.15 : Un panel de supervision.....	36

Figure 2.16 : Icône de WinCC flexible. ....	37
Figure 2.17 : Icône de WinCC flexible. ....	38
Figure 2.18 : Sélection de pupitre.....	38
Figure 2.19 : Intégration avec un projet Setep 7.....	39
Figure 2.20 : Création des vues. ....	39
Figure 2.21 : Choix des objets. ....	40
Figure 2.22 : Déclaration des variables. ....	40
Figure 3.1 : Menu principal. ....	42
Figure 3.2 : Menu des systèmes.....	42
Figure 3.3 : Menu des options.....	43
Figure 3.4: Panneaux du système (1- panneau utilitaire, 2- panneau des E/S, 3- panneau choix du mode, 4- panneau de commande). 4	44
Figure 3.5: Navigation souris-clavier 1. ....	45
Figure 3.6: Navigation souris-clavier 2. ....	45
Figure 3.7 : Pointeur en forme de cible. ....	46
Figure 3.8: Exemples d'interactions avec les systèmes. ....	46
Figure 3.9: Panneau des E/S en mode manuel. ....	47
Figure 3.10: Panneau des entrées-sorties en mode automatique.....	47
Figure 3.11: Système de tri des caisses. ....	48
Figure 3.12: Exemple sur la taille de caisses.....	48
Figure 3.13 représentation des mouvements dans le système du tri. ....	49
Figure 3.14: Quelques actionneurs disponibles (1- Convoyeur à bande, 2- rouleaux du plateau tournant (chargement), 3- rouleaux du plateau tournant (déchargement), 4- plateau tournant).....	50
Figure 3.15 : Représentation de quelques capteurs disponibles (0- capteur de fin du convoyeur d'alimentation, 1,2- capteurs des tailles des caisses, 10- capteur à la fin du convoyeur de sortie). ....	51
Figure 3.16: Tapi d'alimentation (convoyeur A) _ Grafcet G10. ....	52
Figure 3.17: Convoyeur B_ Grafcet G20.....	52
Figure 3.18: Table tournante_ Grafcet G30.....	53
Figure 3.19: Tapi droit _ Grafcet G40. ....	53
Figure 3.20: Tapi gauche (Grafcet G50).....	54
Figure 3.21 : Mémorisation de la taille et le comptage des cartons rentrant (Grafcet G60).....	54
Figure 3.22 : Lecture de la taille et le comptage des cartons sortants (Grafcet G70)..	54
Figure 3.23 Interface du projet.....	56
Figure 3.24 : Table des mnémoniques.....	57
Figure 3.25: Graphe maitre.....	58
Figure 3.26: Convoyeur d'alimentation A. ....	59
Figure 3.27: Convoyeur B. ....	59
Figure 3.28: Table tournante.....	60
Figure 3.29: Convoyeur de sortie gauche.....	61
Figure 3.30: Convoyeur de sortie droite.....	61

---

Figure 3.31: Mémorisation de la taille des caisses et le comptage des caisses rentrantes.....	62
Figure 3.32: Lecture de la taille des caisses et le comptage des caisses sortantes.....	62
Figure 3.33: Contenu de FC1.....	63
Figure 3.34: Contenu de bloc FC2.....	64
Figure 3.35: Bloc d'organisation OB1.....	65
Figure 3.36: Simulation par PLC-SIM.....	65
Figure 3.37: la table de variables HMI.....	67
Figure 3.38: Animation d'une capture.....	67
Figure 3.39: Animation d'un actionneur.....	68
Figure 3.40 : Configuration d'un bouton.....	68
Figure 3.41: la vue de l'HMI de système de tri des caisses.....	68
Figure 3.42: Simulation de l'HMI avec PLCSIM.....	69
Figure 3.43: Simulation générale.....	71
Figure 4.1 : Mélangeur de peinture virtuel de l'ITS PLC.....	72
Figure 4.2 : Configuration du mélangeur de l'ITS PLC.....	73
Figure 4.3 : Emplacement des actionneurs du mélangeur.....	74
Figure 4.4 : Emplacement des capteurs disponibles.....	75
Figure 4.5 : Couleurs de sortie selon les indications des capteurs de niveaux des cuves de dosage.....	76
Figure 4.6 : Graphe maitre.....	77
Figure 4.7 : Graphe auxiliaire.....	77
Figure 4.8 : Interface du projet.....	78
Figure 4.9 : Table des mnémoniques.....	78
Figure 4.10 : Graphe maitre.....	79
Figure 4.11 : Graphe auxiliaire.....	81
Figure 4.12 : FC1 choix de remplissage.....	82
Figure 4.13 : FC2 vidange des cuves.....	82
Figure 4.14 : FC3 décharge de la peinture à la sortie.....	82
Figure 4.15 : FC4 réinitialisation du système.....	82
Figure 4.16 : Table des variables.....	83
Figure 4.17 : Vue Principale.....	83

## Liste des tableaux

Tableau 2.1 : Les types de variables dans Step 7.....	31
Tableau 3.1: Les actionneurs disponibles.....	49
Tableau 3.2 : Les entrées disponibles.....	50
Tableau 4.1 : Actionneurs et leurs adresses.....	73
Tableau 4.2 : Capteurs et boutons et leurs adresses.....	75

# Introduction générale

Depuis la nuit des temps, l'être humain a toujours cherché à économiser ses efforts en cherchant des moyens et des suppléants pour accomplir à sa place les tâches pénibles ou dangereuses. Si, dans ce but, l'homme antique s'est orienté vers l'apprivoisement des animaux et l'esclavage, l'homme moderne a créé des techniques qui réduisent voire rendre inutile son intervention dans beaucoup de tâches où cette intervention était coutumière.

L'automatisation industrielle fait partie des techniques créées pour débarrasser l'homme d'un bon nombre de travaux pénibles et de réaliser des tâches dangereuses, répétitives et fastidieuses. De nos jours, dans le monde industriel l'automatisation est devenue une nécessité pour la compétitivité entre les entreprises, car elle augmente la productivité et la qualité du produit fini et réduit les coûts de production. Ainsi, les entreprises qui cherchent à consolider leur position dans les marchés locaux ou internationaux investissent énormément dans la modernisation de leurs équipements d'automatisation.

Depuis l'invention de l'automate programmable industriel, l'automatisation industrielle a connu un intérêt de plus en plus croissant. Maintenant, on retrouve les systèmes automatisés dans presque tous les secteurs d'activités : industrie pétrolière, production d'énergie, agro-alimentaire, logistique...etc. Ceci entraîne une compétition farouche entre les fabricants de matériels d'automatisme qui ne cessent d'innover et concevoir des solutions matérielles et logicielles de plus en plus performantes.

Le but de notre travail est l'automatisation et la supervision d'un système de tri et d'un mélangeur de peinture en utilisant les logiciels Step 7 et WinCC flexible de Siemens. Pour valider les programmes développés, on utilise le logiciel éducatif ITS PLC qui permet de simuler dans la réalité virtuelle les systèmes considérés. Ceci permet de visualiser les résultats de nos programmes d'un aspect très proche à la réalité, ce qui nous a énormément aidés à comprendre facilement le fonctionnement des systèmes et déceler rapidement les erreurs de programmation.

Pour une bonne présentation de notre travail, ce mémoire a été organisé en quatre chapitres.

Le premier chapitre expose quelques notions indispensables sur les systèmes automatisés et la supervision industrielle. On présente également un état d'art sur les automates de la gamme SIMATIC S7 de Siemens. L'accent est particulièrement mis sur l'automate S7-300.

Le deuxième chapitre est consacré à la description détaillée des logiciels Step7 et WinCC flexible de Siemens qu'on a utilisés pour la programmer et nos projets.

Le troisième chapitre est destiné à présenter notre projet d'automatisation et de supervision d'un système de tri de caisses. Après avoir présenté les fonctionnalités du logiciel ITS PLC, on a donné une description du système à automatiser. Ensuite, on a exposé d'une manière détaillée les programmes et l'interface homme/machine

développés. Leur validation est effectuée en exécutant en parallèle les trois logiciels step7, WinCC flexible et ITS PLC.

Le dernier chapitre est consacré à présenter notre deuxième projet d'automatisation et de supervision d'un système d'un mélangeur de peinture.

# Chapitre 1: Généralité sur l'automatisation et la supervision industrielle

## 1.1 Introduction

L'automatisation industrielle englobe l'ensemble des techniques et équipements permettant de réduire voire éliminer l'intervention humaine directe dans le fonctionnement des installations. Cela comprend les systèmes manufacturiers, les processus industriels ainsi que les domaines de la production d'énergie, la manutention et le bâtiment.

Généralement, l'automatisation est réalisée au moyen des automates programmables industrielles (API) à cause de leur polyvalence, leur relativement faible coût, mais surtout pour l'aisance de programmation procurée par divers langages textuels et graphiques. Néanmoins, il existe d'autres dispositifs qui peuvent assurer le contrôle et la commande des installations automatisées comme le système de contrôle distribué (DCS : Distributed Control system), le contrôleur d'automatisation programmable (PAC : Programmable Automation Controller), PC industriel ...etc.

La supervision industrielle est devenue une partie intégrante des systèmes automatisés. Elle consiste à mettre à la disposition d'un opérateur un ou plusieurs pupitres du type écran/clavier lui permettant de surveiller et de contrôler l'installation localement ou à distance.

L'objectif de ce chapitre est d'exposer quelques notions indispensables sur les systèmes automatisés et la supervision industrielle pour bien appréhender la suite de ce mémoire.

## 1.2 Systèmes automatisés

### 1.2.1 Objectifs et atouts de l'automatisation

L'objectif derrière l'automatisation est de permettre aux systèmes industriels de fonctionner et de mettre au point des produits de qualité avec les coûts les plus faibles possibles et en ayant la moindre intervention humaine possible. Elle permet de :

- accroître la productivité de l'entreprise en réduisant les coûts de production ;
- améliorer la flexibilité et la souplesse de la production ;
- perfectionner la qualité du produit ;
- s'adapter à des contextes particuliers ;
- augmenter la sécurité ;
- réduire les tâches complexes, pénibles ou indésirables en les faisant exécuter par la machine ;
- faciliter les changements de fabrication en permettant de passer d'une quantité ou d'un type de production à un autre .

- réaliser des économies de matière et d'énergie

Cependant, les coûts du matériel et des logiciels utilisés dans les systèmes automatisés sont très élevés. De plus l'automatisation est tenue responsable de la suppression postes d'emploi.

### 1.2.2 C'est quoi un système automatisé ?

Un système automatisé est un ensemble d'éléments en interaction, et organisés dans un but précis : agir sur une matière d'œuvre afin de lui donner une valeur ajoutée [1]. Un automatisme ou système automatisé peut être parfois composé de plusieurs sous-systèmes automatisés.

Les API dans un système automatisé sont programmées de telle sorte que ce dernier exécute toujours le même cycle de travail sans l'intervention d'un opérateur humain. Dans les conditions de fonctionnement normales, celui-ci n'intervient que dans l'opération de programmation et le démarrage et l'arrêt du système.

Pour pouvoir fonctionner correctement, un système automatisé collecte les informations sur son environnement et sur son propre état (informations issues de capteurs) ainsi que les consignes en provenance de l'opérateur, les traite et envoie les ordres à une partie de puissance désignée par chaîne d'énergie pour actionner sa partie opérative.

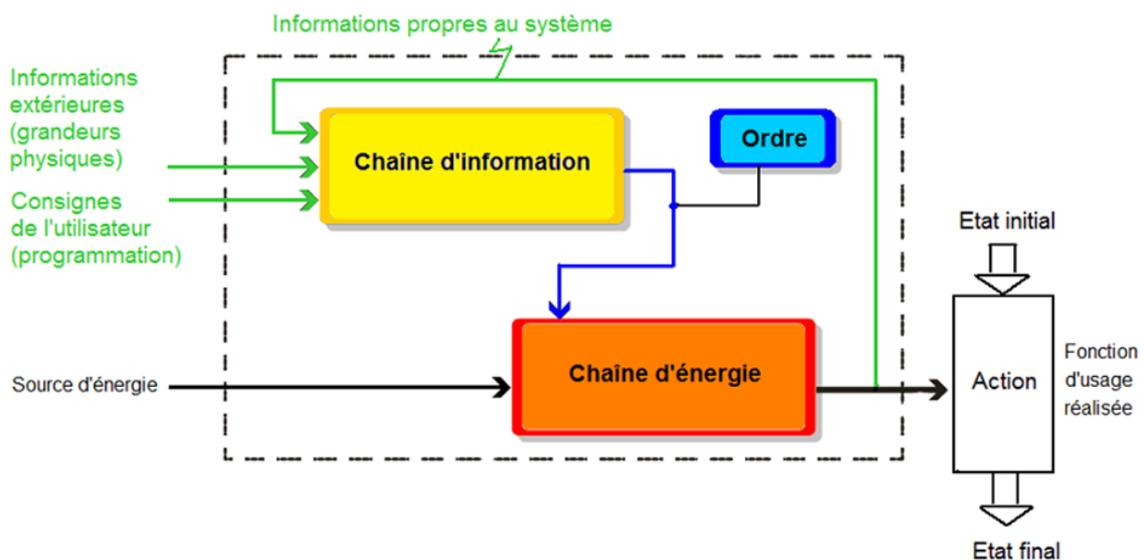


Figure 1.1 : Schéma simplifié d'un système automatisé.

### 1.2.3 Chaines fonctionnelles

#### a) Chaîne d'information

La chaîne d'information est la partie de l'automatisme responsable d'acquérir et traiter les informations issues des capteurs ou d'autres instruments et communiquer les ordres à la chaîne d'énergie. On peut subdiviser son fonctionnement en plusieurs blocs fonctionnels comme le montre la figure 1.2 [2].

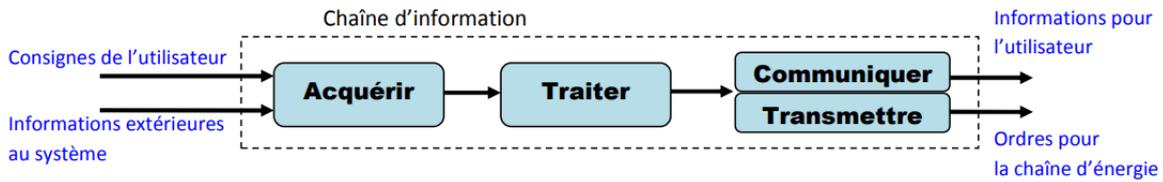


Figure 1.2 : Chaîne d'information.

- **Acquérir** : Fonction qui permet de prélever des informations à l'aide de capteurs.
- **Traiter** : C'est la partie commande composée d'un automate ou d'un microcontrôleur.
- **Communiquer** : Cette fonction assure l'interface l'utilisateur et/ou d'autres systèmes.
- **Transmettre** : Cette fonction assure l'interface avec l'environnement de la partie commande.

### b) Chaîne d'énergie

Dans un automatisme, la chaîne d'énergie représente l'ensemble des procédés impliqués dans la réalisation d'une action [2].

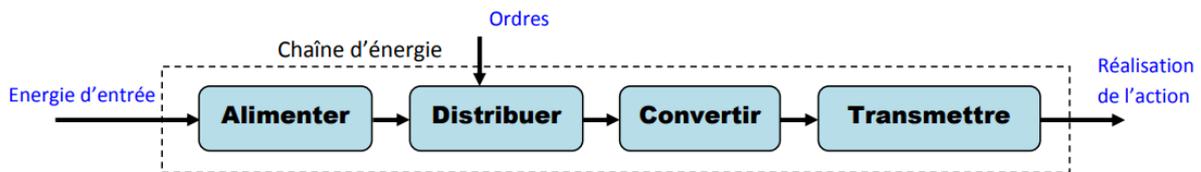


Figure 1.3 : Chaîne d'énergie.

- **Alimenter** : Mise en forme de l'énergie externe en énergie compatible pour créer une action.
- **Distribuer** : Distribution de l'énergie à l'actionneur réalisée par un distributeur ou un contacteur.
- **Convertir** : L'organe de conversion d'énergie appelé actionneur peut être un vérin, un moteur...
- **Transmettre** : Cette fonction est remplie par l'ensemble des organes mécaniques de transmission de mouvement et d'effort : engrenages, courroies, accouplement...etc.

## 1.2.4 Structure des systèmes automatisés

Communément, un système automatisé se compose de trois parties : une partie opérative et une partie commande.

### a) La partie opérative

C'est la partie visible du système. Elle comporte les éléments suivants :

- les pré-actionneurs (distributeurs, contacteur) qui reçoivent des ordres de la partie commande ;

- les actionneurs (vérins, moteurs, vannes) qui ont pour fonction d'exécuter ces ordres, ils transforment l'énergie électrique, pneumatique ou hydraulique en énergie mécanique ;
- les capteurs qui informent la partie commande de l'exécution du travail.

**b) La partie commande**

Cette partie de l'automatisme gère selon une suite logique le déroulement ordonné des opérations à réaliser. Elle reçoit les informations en provenance des capteurs de la partie opérative, et selon ceux-ci et un modèle de commande établi elle génère les ordres en direction des pré-actionneurs et actionneurs.

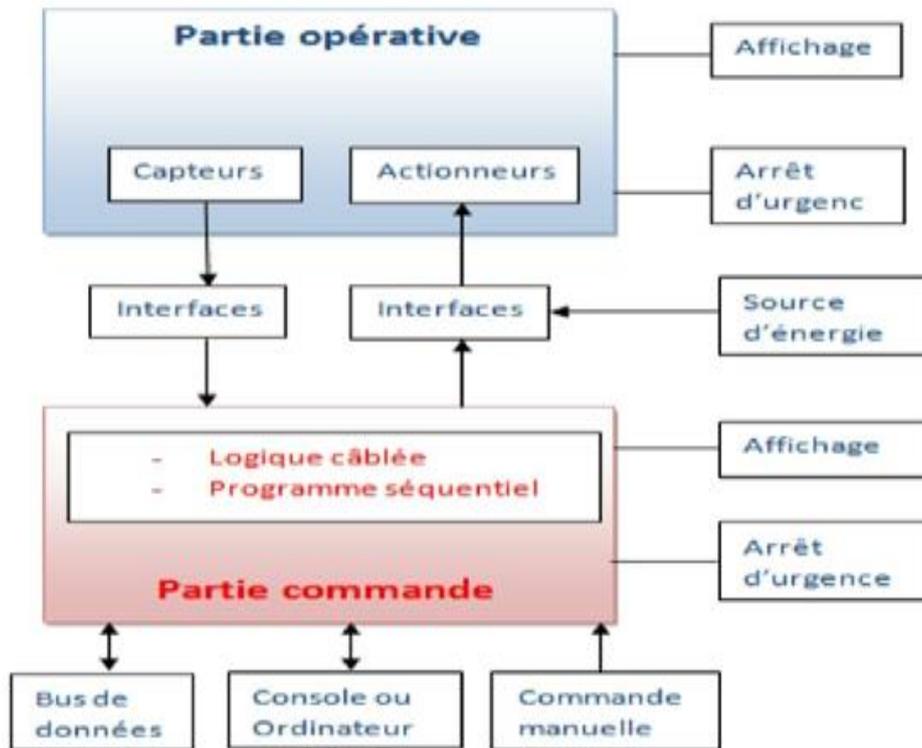
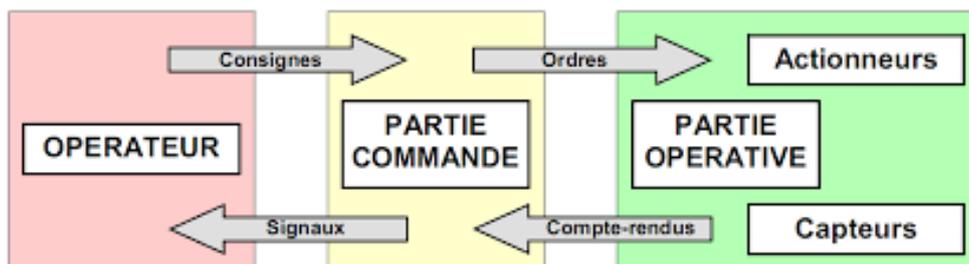


Figure 1.4 : Structure d'un système automatisé.

**c) Dialogue entre la partie commande et la partie opérative et la partie commande et l'opérateur**

Le dialogue entre la partie opérative et la partie commande se fait au moyen des signaux de mesures fournis par les capteurs et les signaux de commande à destination des actionneurs.



*Figure 1.5 : Dialogue entre la partie commande et la partie opérative.*

Le dialogue entre l'opérateur et la partie commande se fait au biais d'interfaces homme/machine qui transforme les signaux fournis par l'API en visualisations et avertisseurs compréhensibles par l'homme et les consignes de l'opérateur en signaux exploitables par l'API.

## 1.3 Généralités sur les automates programmables industriels (API)

L'API (PLC en anglais pour Programmable Logic Controller) a été développé pour la première fois par Modicon aux États-Unis vers 1969 pour répondre au désir du constructeur d'automobiles General Motors de développer des chaînes de fabrication automatisées qui pourraient suivre l'évolution des technologies et des modèles de voitures fabriqués. Les API se sont ainsi substitués aux armoires à relais en raison de leur souplesse, mais aussi parce que dans les automatismes de commande complexe, les coûts de câblage et de mise au point devenaient trop élevés.

### 1.3.1 Définition

Un API est un appareil électronique spécialisé dans la conduite et la surveillance en temps réel de processus industriels et tertiaires. Il exécute une suite d'instructions introduites dans sa mémoire sous forme de programme et s'apparente par conséquent aux machines de traitement d'information. Trois caractéristiques fondamentales le distinguent totalement des outils d'informatiques tels que les ordinateurs utilisés dans les entreprises et tertiaires [3] :

- Connexion directe aux différents capteurs et actionneurs grâce à ces entrées/sortie.
- Fonctionnement dans des conditions industrielles sévères (température, vibrations, humidité, microcoupure de l'alimentation en énergie électrique...)
- Son aspect pratique grâce à la possibilité de sa programmation en utilisant un langage spécialement développé pour le traitement de fonctions d'automate (SIMATIC S7)



*Figure 1.6 : Exemples d'API.*

### 1.3.2 Principes de fonctionnement d'un API

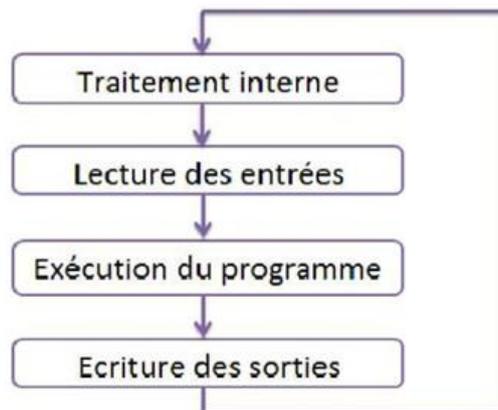
Le fonctionnement de la quasi-totalité des API utilisés dans les automatismes industriels se résume en l'exécution d'une manière cyclique des tâches suivantes [4]:

**Traitement interne** : l'automate effectue des opérations de contrôles et met à jour certains paramètres du système (détection des passages en RUN/STOP, mises à jour des valeurs de l'horodateur ...).

**Lecture des entrées** : l'automate lit les entrées (de façon synchrone) et les recopie dans la mémoire image des entrées.

**Exécution du programme** : l'automate exécute le programme instruction par instruction et écrit les sorties dans la mémoire image des sorties.

**Écriture des sorties** : l'automate bascule les différentes sorties (de façon synchrone) aux positions définies dans la mémoire image des sorties. Ces quatre opérations sont effectuées continuellement par l'automate (fonctionnement cyclique).



*Figure 1.6 : Cycle de fonctionnement de l'API.*

### 1.3.3 Architecture d'un API

De manière générale, l'automate programmable industriel est composé des éléments suivants :

#### a) Processeur (Central Processing Unit, CPU)

Son rôle consiste à :

- traiter les instructions qui constituent le programme de fonctionnement de l'application ;
- gérer les entrées et sorties ;
- surveiller et diagnostiquer l'automate (par des tests lancés régulièrement) ;
- mettre en place un dialogue avec le terminal de programmation.

**b) Mémoire**

Elle permet le stockage des instructions constituant le programme de fonctionnement ainsi que diverses informations. Il peut s'agir de mémoire vive RAM ou de mémoire morte EEPROM.

**c) Interfaces entrées/sorties**

Elles permettent au processeur de recevoir et d'envoyer des informations. Ces dispositifs d'entrée et sortie peuvent produire des signaux discrets :

- numériques (ce sont des sorties de type « tout ou rien : TOR »),
- analogiques.

L'interface d'entrée permet la connexion à l'API d'un multiple de capteurs et l'interface de sortie permet la connexion de l'API aux pré-actionneurs.

**d) Alimentation**

Elle est indispensable puisqu'elle convertit une tension alternative en une basse tension continue (3V, 5V, 12V, 24V) nécessaire au processeur et aux modules d'entrées-sorties.

L'alimentation ne fait pas toujours partie de l'automate qui sera donc directement alimenté par une base tension.

**e) Interface de communication**

Elle est utilisée pour recevoir et transmettre des données via des réseaux de communication en fonction des protocoles choisis. Elle est impliquée dans des opérations :

- acquisition de données,
- la synchronisation entre des applications,
- la gestion de la connexion.

**f) Périphérique de programmation**

Il est utilisé pour programmer l'API. Le programme est développé sur un ordinateur en se servant d'un logiciel approprié, puis on transfère l'exécutable dans la mémoire de l'API.

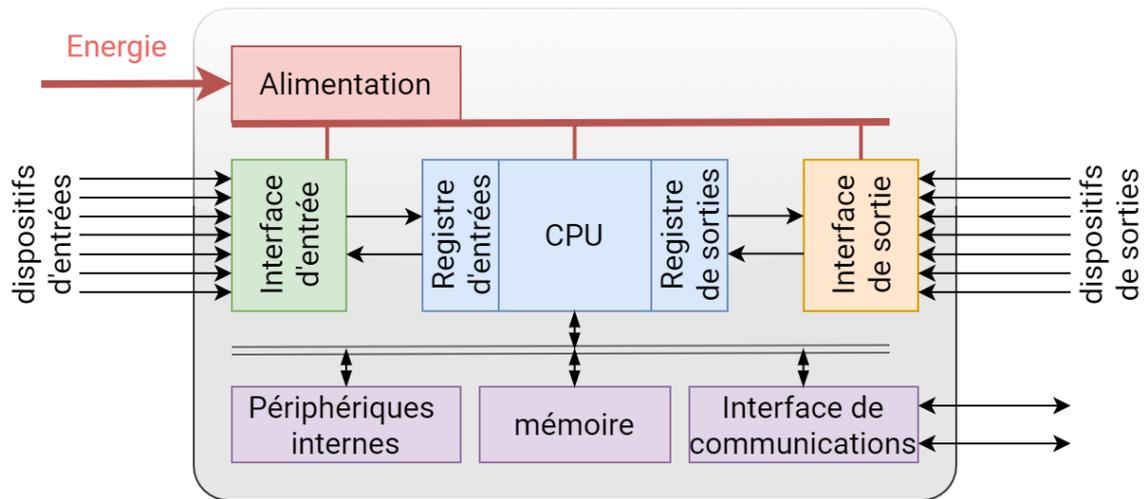


Figure 1.7 : Structure interne d'un API.

## 1.4 Présentation de quelques API de la gamme SIMATIC S7 de Siemens

### 1.4.1 SIMATIC S7

Dans cette sous-section, on décrit les cinq grandes familles des API de la gamme S7 de Siemens.

#### a) SIMATIC S7-200

La famille S7-200 est constituée de micro-automates programmables utilisables dans des applications d'automatisations variées. Son dessin compact, ses possibilités d'expansion, son faible prix et son important jeu d'opérations en font une solution idéale pour la commande de petites applications. En outre, le large choix de tailles et de tensions du CPU offre la souplesse nécessaire pour résoudre un problème d'automatisation.



Figure 1.8 : API Siemens S7-200.

Le S7-200 consiste en une CPU S7-200 seule ou complétée de divers modules d'extension facultatifs connectés à cette dernière à l'aide d'un connecteur de bus fourni avec ce module d'extension [5].

**b) SIMATIC S7-300**

La famille S7-300 est constituée d'automates programmables de conception modulaire utilisés pour des automatismes de gamme moyenne, et peuvent être connectés entre eux au moyen d'un câble-bus PROFIBUS.

Un automate S7-300 consiste en une CPU, un module d'alimentation PS, un module de comptage FM, un module de signaux SM et un processeur de communication [5].



*Figure 1.9 : API Siemens S7-300.*

**c) SIMATIC S7-400**

La famille S7-400 est également constituée d'automates programmables de conception modulaire. Pratiquement, chaque tâche d'automatisation peut être résolue par un choix approprié des constituants du S7-400 et avec la possibilité d'expansion de plusieurs modules.



*Figure 1.10 : API Siemens S7-400.*

**d) SIMATIC S7-1200**

L'API S7-1200 offre la souplesse et la puissance nécessaires pour commander une large gamme d'appareils afin de répondre à la plupart des besoins en matière d'automatisation. Sa forme compacte, sa configuration souple et son important jeu d'instructions en font une solution idéale pour la commande d'applications très variées.

La CPU combine un microprocesseur, une alimentation intégrée, des circuits d'entrée et de sortie, un PROFINET intégré, des E/S rapides de commande de mouvement, ainsi que des entrées analogiques intégrées dans un boîtier compact en vue de créer un contrôleur puissant [6].



Figure 1.11 : API Siemens S7-1200.

### e) SIMATIC S7-1500

L'API S7-1500 offre la flexibilité et la puissance nécessaires à un large éventail d'applications modernes d'automatisation. Sa structure modulaire permet d'adapter l'automate aux conditions de fonctionnement sur site. De plus, il y a des modèles qui possèdent deux CPU redondantes (une principale et l'autre en réserve) qui exécutent le programme utilisateur de manière synchrone pour pouvoir passer le relais à la CPU en réserve en cas de besoin, ce qui offre une haute disponibilité du système [7].



Figure 1.12 : API Siemens S7-1200.

## 1.5 Choix de l'API

Les API représentent le cerveau des systèmes automatisés. C'est pourquoi, avant d'entamer un projet d'automatisation industrielle, il est très important de choisir le bon modèle d'automate qui répond parfaitement au cahier de charges fixé.

Les critères de choix des API à utiliser tiennent compte évidemment de leurs coûts, mais essentiellement des caractéristiques suivantes :

- Le nombre d'entrées/sorties intégrées.
- Le type d'entrées/sorties.
- Le temps de traitement (scrutation).
- La capacité de la mémoire.
- Le nombre de compteurs.
- Le nombre de temporisateurs.
- L'alimentation de la CPU.

### 1.5.1 Présentation de l'API choisi le S7-300

Pour notre application virtuelle, on a choisi le S7-300 qui est un automate de moyenne gamme, mais qui répond largement au besoin du cahier de charge fixé pour notre application. Il a les caractéristiques suivantes :

- Une gamme diversifiée de la CPU.
- Une possibilité d'extension jusqu'à 32 modules.
- Une possibilité de mise en réseau avec :
  - Profibus.
  - L'interface multipoints.
  - Industriel Ethernet.
- Un raccordement central de la console programmation (PG) avec accès à tous les modules.
- La liberté de montage aux différents emplacements.

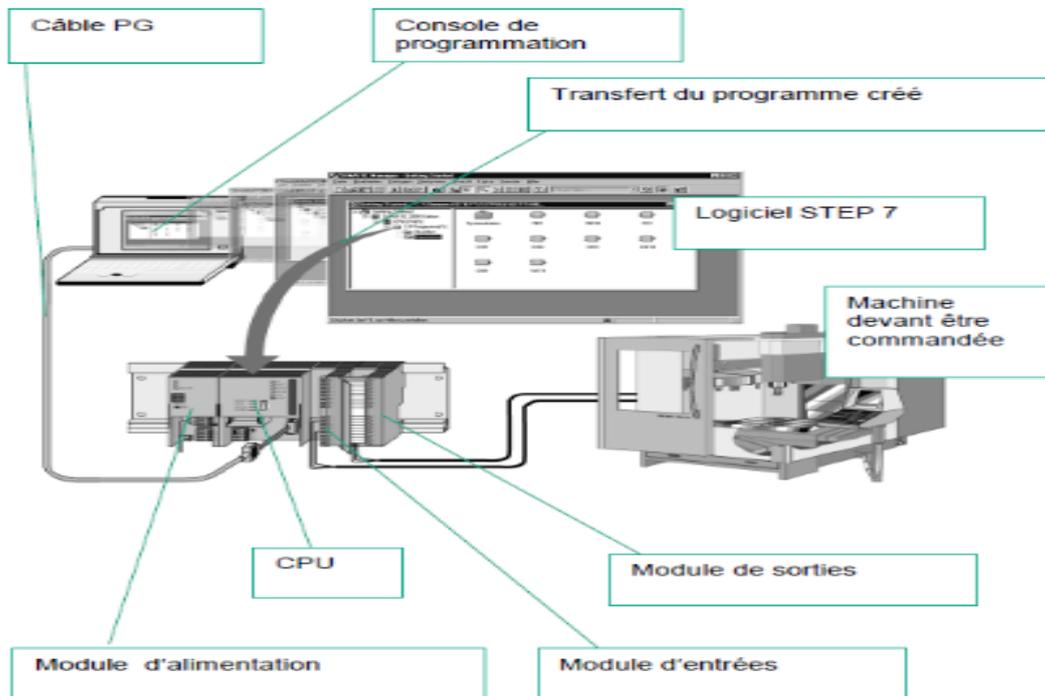


Figure 1.13 : L'automate S7-300 et son environnement.

### 1.5.2 Structure de l'API S7-300

Dans cette sous-section, on décrit les modules qui compose le S7-300

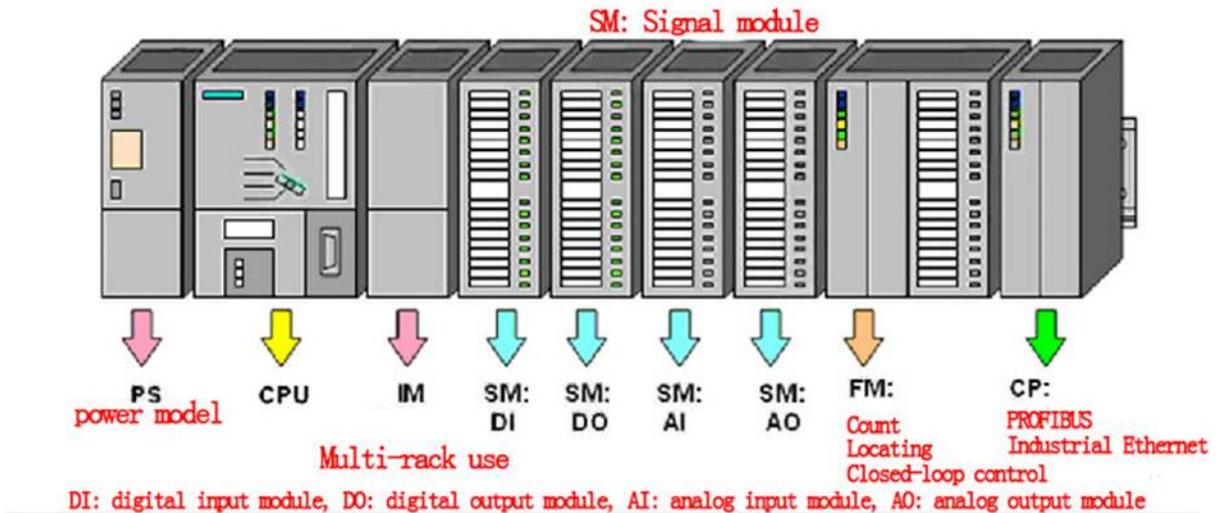


Figure 1.14 : Les modules de l'API S7-300.

**a) Module d'alimentation (PS : Power Supply)**

Ce module permet de fournir l'énergie nécessaire pour le fonctionnement des autres différents modules de l'automate à partir d'une alimentation de 220V alternative. Il délivre les tensions continues dont l'automate a besoin (5V, 12V, 24V). Un voyant est positionné sur la façade pour indiquer la mise sous tension de l'automate.

**b) Unité centrale CPU**

À base de microprocesseur, elle réalise toutes les fonctions logiques, arithmétiques et de traitement numérique (transfert, comptage, temporisation...). Comme pour les ordinateurs, la CPU tient la place la plus primordiale dans l'automate, c'est l'organe maître. L'API S7-300 dispose de plusieurs modèles de CPU, offrant ainsi un choix diversifié à l'utilisateur.



Figure 1.15 : Module CPU de S7-300 (modèle CPU 315 2PN/DP).

**c) Modules de signaux (SM)**

Les modules de signaux (SM) servent d'interface entre le processus et l'automate. Il existe des modules d'entrées et des modules de sorties TOR, ainsi que des modules d'entrées et des modules de sorties analogiques.

- **Modules entrées/sorties TOR (SM 321/SM 322) :** Les modules entrées/sorties TOR constituent les interfaces d'entrée et de sortie pour les signaux TOR de l'automate. Ces modules permettent de raccorder à l'automate S7-300 des capteurs et des actionneurs TOR en utilisant si nécessaire des équipements d'adaptation (conditionnement, conversion...etc.).



*Figure 1.16 : Modules SM d'E/S TOR de S7-300 (de gauche à droite : module d'entrées TOR SM 321, module de sorties TOR SM322).*

- **Les modules d'entrées/sorties analogiques (SM 331/ SM332) :** Ces modules permettent de raccorder à l'automate des capteurs et des actionneurs analogiques. Les modules d'entrées analogiques (SM 331) réalisent la conversion des signaux analogiques, issus du processus, aux signaux numériques pour le traitement interne dans S7-300. Les modules de sorties analogiques (SM 332) convertir les signaux numériques internes (du S7-300) aux signaux analogiques destinés aux actionneurs ou pré-actionneurs analogiques



*Figure 1.17 : Modules SM d'E/S analogiques de S7-300 (de gauche à droite : module d'entrées analogiques SM331, module de sorties analogiques SM332).*

#### d) Module de fonction (FM)

Ils réalisent les tâches du traitement des signaux critiques au niveau du temps et exigeants beaucoup de mémoire, comme par exemple le positionnement ou le réglage.



Figure 1.18 : Module de fonction (FM) de S7-300.

### e) Module de communication (CP)

Les processeurs de communication (CP) réalisent le couplage point-à-point qui relie les partenaires de communication comme les API, PC...etc.

## 1.6 Supervision

La programmation et la mise en marche d'une installation industrielle automatisée ne sont pas suffisantes, car des événements imprévus peuvent altérer son bon fonctionnement et même provoquer des dommages matériels et mettre le personnel en danger. Il est donc impératif de ne pas confier entièrement l'installation à l'API et ainsi de la superviser par des opérateurs humains.

La supervision est une technique industrielle de suivi et de pilotage informatique des procédés de fabrication automatisés. C'est une forme évoluée de dialogue Homme/Machine. Elle concerne l'acquisition de données en temps réel (mesures, alarmes, retour d'état de fonctionnement) et des paramètres de commande des processus généralement confiés à des API.

La supervision industrielle consiste généralement à mettre à la disposition d'un opérateur un système de visualisation (généralement écran plus clavier) pour lui indiquer l'état et le mode de fonctionnement de l'installation.

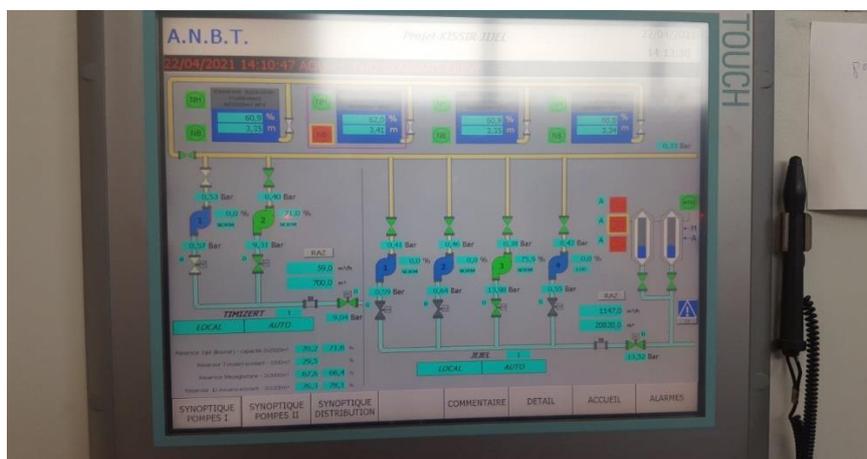


Figure 1.19 : Pupitre de supervision du système de pompage des eaux traitées (station de traitement des eaux de Kissir).



Figure 1.20 : Pupitre de supervision du système de remplissage des eaux traitées (station de traitement des eaux de Kissir).

### 1.6.1 Avantages de la supervision

Le système de supervision aide à l'opérateur dans la surveillance et la conduite du processus en lui présentant l'état de l'installation sous forme graphique claire et compréhensible. Ses avantages principaux sont :

- La possibilité de surveiller le processus à distance.
- Le diagnostic et le traitement des alarmes.
- La détection des défauts.
- Le traitement des données.

### 1.6.2 Constitution d'un système de supervision

Les systèmes de supervision se composent généralement d'un moteur central (logiciel), à qui se rattachent des données provenant des équipements (automates). Le logiciel de supervision assure l'affichage, le traitement des données, l'archivage et la communication avec d'autres périphériques. Ayant pour fonction, la mise à la disposition de l'opérateur des données instantanées du procédé. Les modules de visualisation comportent :

#### a) Un module d'archivage

Ayant comme rôle la mémorisation des données (alarmes et événements) pendant une longue période et l'exploitation des données dans des applications spécifiques pour les fins de maintenance ou de gestion de production.

#### b) Un module de traitement

Permet la mise en forme des données afin de les présenter via le module de visualisation aux opérateurs sous une forme prédéfinie.

### c) Un module de communication

Ayant pour fonctions l'acquisition, le transfert des données et la gestion de la communication avec les automates programmables industriels et autres périphériques.

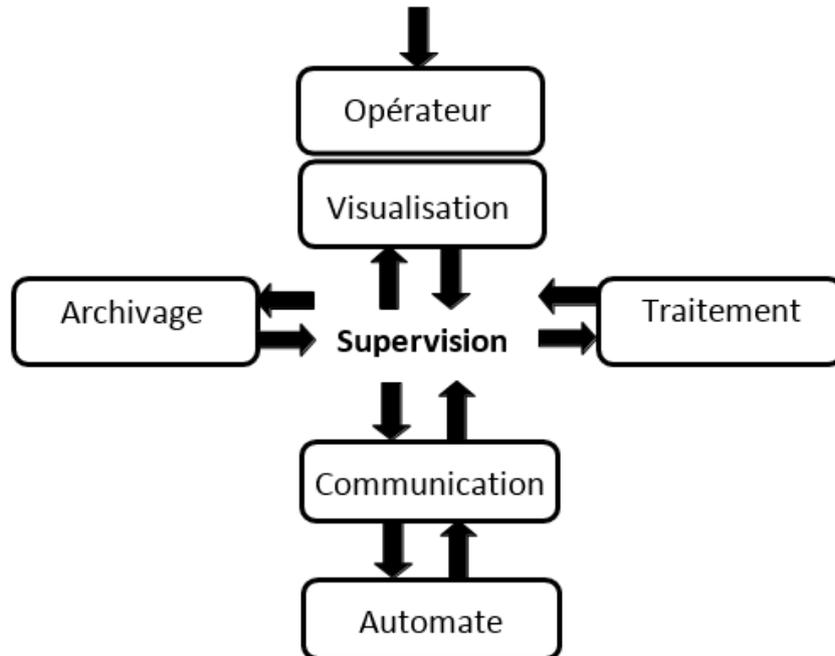


Figure 1.21 : : Schéma synoptique d'un système de supervision.

## 1.7 Conclusion

Ce premier chapitre est destiné à présenter les notions essentielles pour bien aborder les problèmes d'automatisation des systèmes industriels.

Dans un premier temps, on a abordé d'une manière non exhaustive les caractéristiques et l'architecture générale des systèmes automatisés. Dans un deuxième temps, on a présenté quelques généralités sur les API et on a décrit la gamme S7 de Siemens. L'accent est mis sur S7-300 que l'on a choisi d'utiliser dans notre application.

Enfin, on a complété le chapitre en abordant le concept de la supervision industrielle.

# Chapitre 2: Logiciels STEP 7 et Wincc flexible

## 2.1 Introduction

L'une des raisons majeures derrière le succès des API dans le monde industriel est la diversité des langages de programmation et leurs environnements de développement conviviaux. La plupart des constructeurs des API ont élaboré des langages spécifiques à leurs produits, cependant il existe des langages universels à tous les API. Parmi lesquels on distingue des langages textuels et des langages graphiques [8]:

- Le langage à contact (Ladder)
- Le langage diagramme de bloc fonctionnel (FBD).
- Le langage LIST.
- Le langage littéral structuré.
- Le langage GRAFCET.

Les premiers quatre langages listés ci-dessus sont régis par la norme industrielle IEC 61131-3 et sont donc communs à tous les fabricants [9].

Contrairement aux langages de programmation des ordinateurs et des systèmes embarqués, les environnements de développement des API sont élaborés (à quelques exceptions) par les constructeurs des automates eux-mêmes spécifiquement pour leurs produits. Ainsi, leurs popularités proviennent de celles des marques des API qu'ils supportent.

Bien entendu, puisque le leader mondial des systèmes automatisés est Siemens, ses API sont les plus utilisés dans le monde (toutes régions confondues). Par conséquent, ce sont ses logiciels de programmation des API et de supervision qui sont les plus populaires.

Ce chapitre a pour objectif de décrire les logiciels Step7 et WinCC flexible de Siemens qu'on a utilisé pour la programmer et la superviser notre projet.

## 2.2 Description du logiciel Step 7

### 2.2.1 Présentation de Step 7

Step 7 est le logiciel de base le plus populaire et le plus utilisé au monde pour la programmation des API, que ce soit dans le monde industriel ou académique. Il offre toutes les fonctions nécessaires pour la configuration et la programmation de la gamme SIMATIC S7. Son interface utilisateur commode à la plupart des utilisateurs de Windows organise ses fonctionnalités d'une manière familière et facile à apprendre.

Step7 offre les fonctions suivantes [10]:

- Création et gestion des projets.
- La création des programmes.
- Configuration et paramétrage protocoles de communication.

- Gestion des mnémoniques.
- Test de l'installation d'automatisation.
- Le diagnostic lors des perturbations dans l'installation.
- Documentation et archivage.

### 2.2.2 Gestionnaire de projets SIMATIC Manager

SIMATIC Manager constitue l'application centrale de Step7. On le trouve dans son écran d'accueil. Il présente l'arborescence de tous les objets du projet ce qui permet d'accéder rapidement à tous les outils et fonctions nécessaires pour programmer et configurer l'automate.

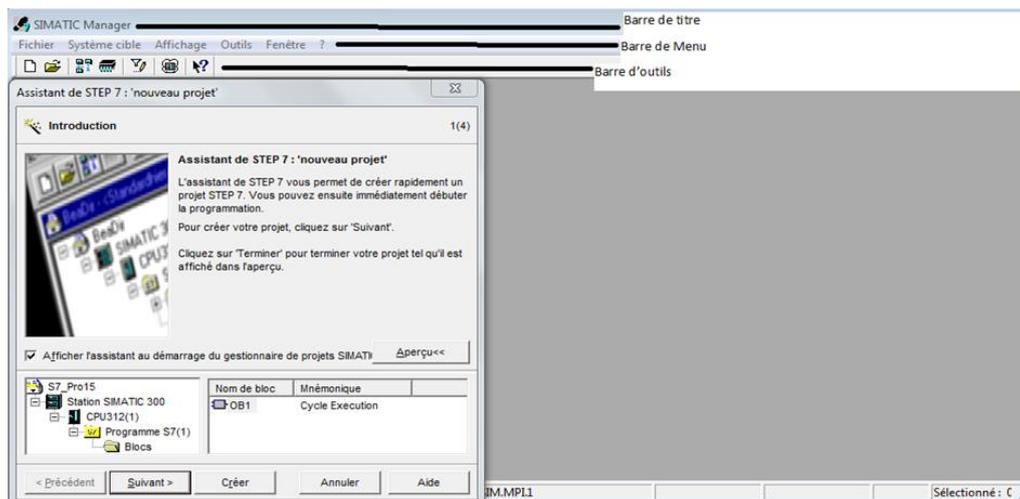


Figure 2.1 : Démarrage de SIMATIC Manager.

### 2.2.3 Langages de programmation

Step7 offre la possibilité d'écrire les programmes sous forme de plusieurs langages de programmation.

#### a) Langage de programmation graph s7 (commande séquentielle)

Le langage de programmation graphique optionnel Graph S7 permet de programmer des commandes séquentielles. Ceci implique la création d'une succession d'étapes, la détermination du contenu respectif de ces étapes, et les conditions de transfert (transitions). Pour déterminer le contenu des étapes, on utilise un langage de programmation spécial (similaire à LIST), alors que pour déterminer les transitions, on utilise une représentation sous forme de schéma à contacts ou de logigramme [11].

Graph S7 permet de représenter très clairement des séquences même complexes, ce qui favorise une programmation et une recherche d'erreurs effectives.

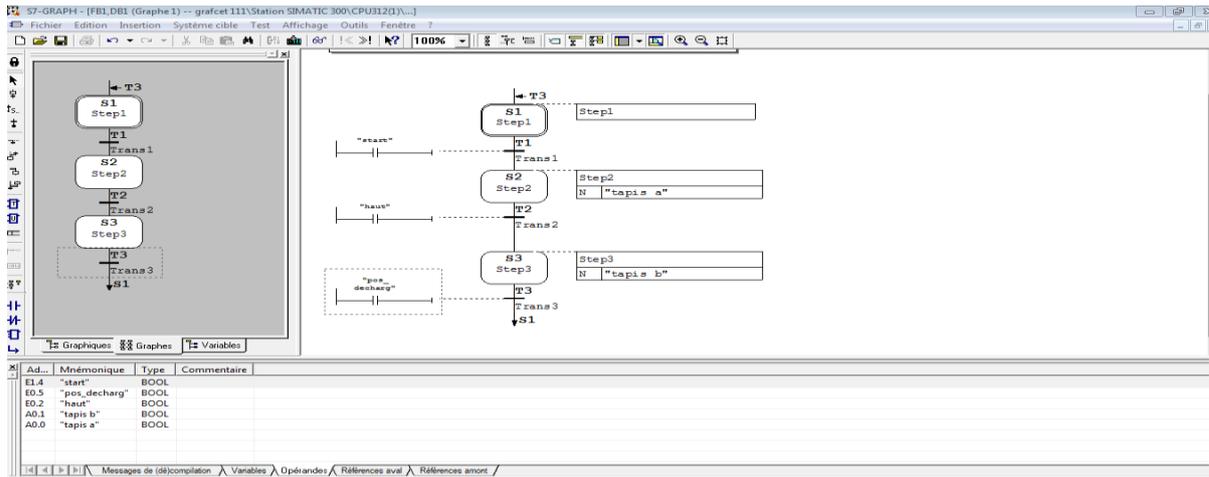


Figure 2.2 : Exemple de programme en Graph S7.

### b) Langage de programmation CONT

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

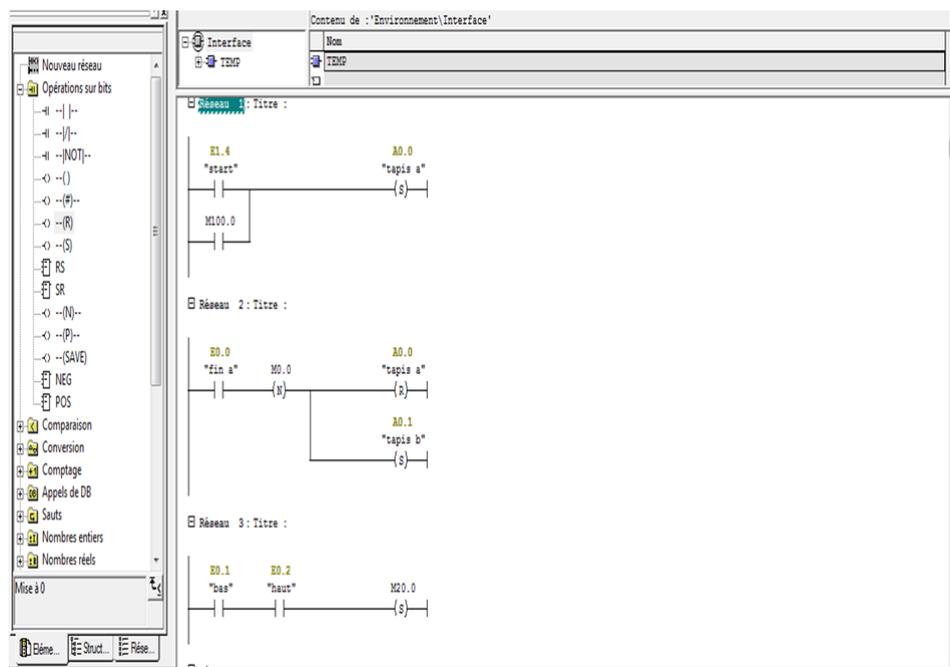


Figure 2.3 : Exemple de programme CONT.

### c) Langage de programmation LOG

Le langage de programmation LOG utilise les pavés logiques bien connus dans l'algèbre booléenne pour la représentation logique. Il permet en outre de représenter des fonctions complexes, telles que les fonctions mathématiques en les mettant directement en liaison avec ces pavés logiques.

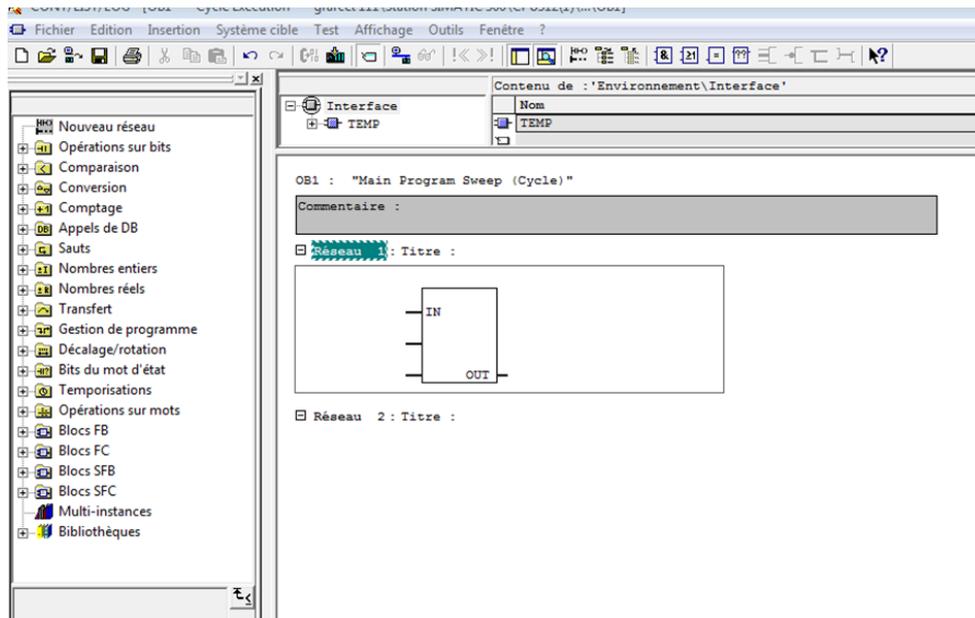


Figure 2.4 : Exemple de programme LOG.

#### d) Langage de programmation LIST

Le langage de programmation LIST (liste d'instructions) est un langage textuel proche du langage machine [12].

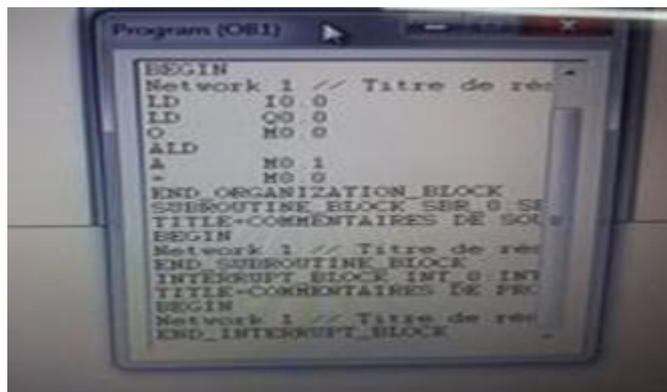


Figure 2.5 : Exemple de programme LOG.

### 2.2.4 Structure générale d'un programme Step 7

Les blocs de code sont les blocs du programme utilisateur qui contiennent les instructions à exécuter. Il existe les blocs de code suivants :

**Bloc d'organisation (OB):** c'est le chef d'orchestre du programme ; il traite cycliquement le programme.

**Fonction (FC) :** c'est un bloc de code qui ne contient pas de données statiques.

**Bloc fonctionnel (FB) :** c'est un bloc de code qui contient des données statiques (exemple programmation de graphe SFC dérivé du GRAFCET).

**Fonction système (SFC et SFB) :** c'est une fonction intégrée au système d'exploitation de la CPU qu'on peut appeler dans le programme utilisateur et elle ne peut être programmée par l'utilisateur.

**Blocs de données (DB) :** ce sont des blocs utilisés par les blocs de code du programme utilisateur pour enregistrer des valeurs.

Les blocs de code (OB, FB, FC) du programme utilisateur peuvent être chargés dans la CPU S7. Ils sont soit créés et édités directement dans des éditeurs incrémentaux, soit ils résultent de la compilation de sources.

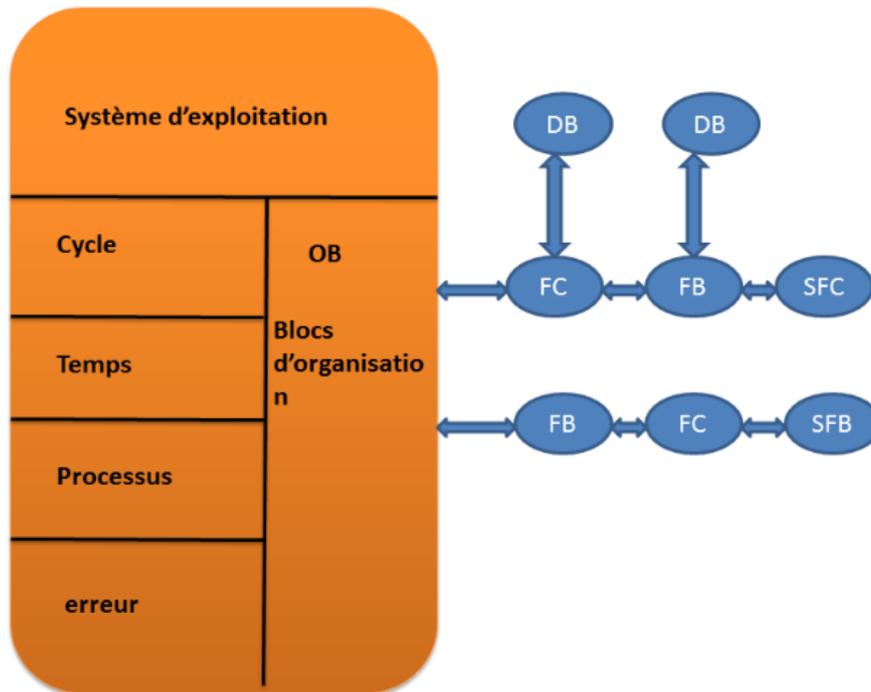
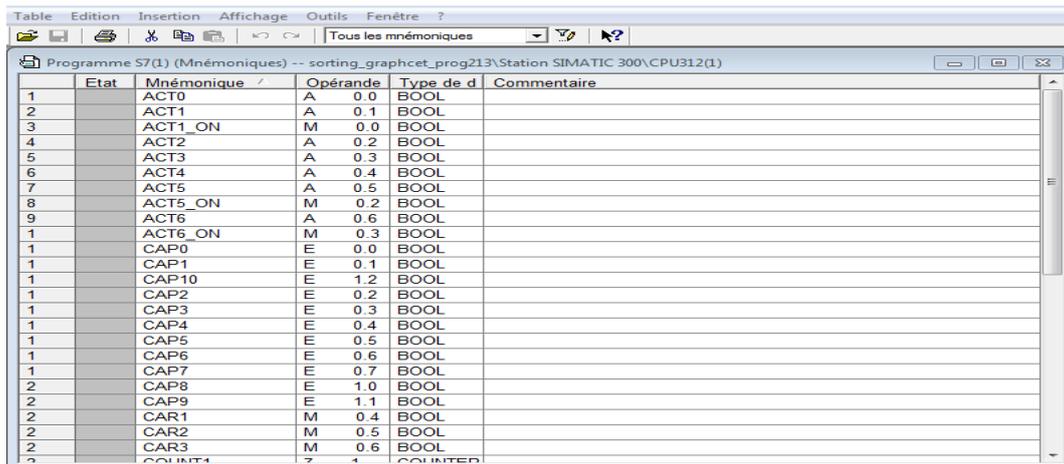


Figure 2.6 : Exemple de la structure générale d'un programme STEP7.

### 2.2.5 Éditeur des mnémoniques

Il permet la gestion de toutes les variables globales. C'est-à-dire la définition des désignations symboliques et des commentaires pour les signaux du processus (entrées/sorties), les mementos, les blocs de données, les temporisations et les compteurs [13].



Etat	Mnémonique	Opérande	Type de d	Commentaire
1	ACT0	A 0.0	BOOL	
2	ACT1	A 0.1	BOOL	
3	ACT1_ON	M 0.0	BOOL	
4	ACT2	A 0.2	BOOL	
5	ACT3	A 0.3	BOOL	
6	ACT4	A 0.4	BOOL	
7	ACT5	A 0.5	BOOL	
8	ACT5_ON	M 0.2	BOOL	
9	ACT6	A 0.6	BOOL	
1	ACT6_ON	M 0.3	BOOL	
1	CAP0	E 0.0	BOOL	
1	CAP1	E 0.1	BOOL	
1	CAP10	E 1.2	BOOL	
1	CAP2	E 0.2	BOOL	
1	CAP3	E 0.3	BOOL	
1	CAP4	E 0.4	BOOL	
1	CAP5	E 0.5	BOOL	
1	CAP6	E 0.6	BOOL	
1	CAP7	E 0.7	BOOL	
2	CAP8	E 1.0	BOOL	
2	CAP9	E 1.1	BOOL	
2	CAR1	M 0.4	BOOL	
2	CAR2	M 0.5	BOOL	
2	CAR3	M 0.6	BOOL	
2	COUNT1	7 1	COUNTER	

Figure 2.7 : Table des mnémoniques.

## 2.2.6 Différents types des variables des programmes Step 7

Dans l'environnement STEP7 on utilise plusieurs types de variables qu'on doit déclarer au préalable, le tableau suivant résume les types de variables utilisées [4].

Symbol	Nom	Taille	Plage	Valeur
Bool	Boolean	1	0 (false) et 1 (true)	Logique
Int	Integer	16	[-32768 , 32767]	Entière continue
DInt	Double integer	32	[-2 <sup>31</sup> , 2 <sup>31</sup> ]	Continue entière double
UInt	Unsigned integer	16	[0.65 , 535]	Entière continue non signée
REAL	Real Numbers	32	[-2 <sup>128</sup> , 2 <sup>128</sup> ]	Continue réelle (flottante)
Byte	Bits string	8	[0 , 255]	demi-mot
WORD	Bits string	16	[0 , 65535]	Mot
DWORD	Bits string	32	[0 , 2 <sup>32</sup> ]	double mot
Time			[0 ms , 10* 2 <sup>31</sup> ms]	Temps
Date				date
String			1 à 255 ASCII	Chaîne de caractères

Tableau 2.1 : Les types de variables dans Step 7.

## 2.2.7 Création d'un projet

Pour réaliser votre tâche d'automatisation au sein d'un gestionnaire de projets, on doit d'abord créer un nouveau projet. Il va être créé dans le répertoire que l'on a sélectionné pour les projets lorsque on a choisi la commande Outils > Paramètres et l'onglet "Général".



Figure 2.8 : Assistant de nouveau projet.

Pour la suite on procède selon les étapes suivantes :

- 1- Après avoir lancer SIMATIC Manager et choisir nouveau projet, on fait une clique sur l'icône « suivant », la fenêtre qui apparaîtra nous permettra de choisir la CPU avec laquelle on veut travailler (Pour notre application on a choisi la CPU 314).

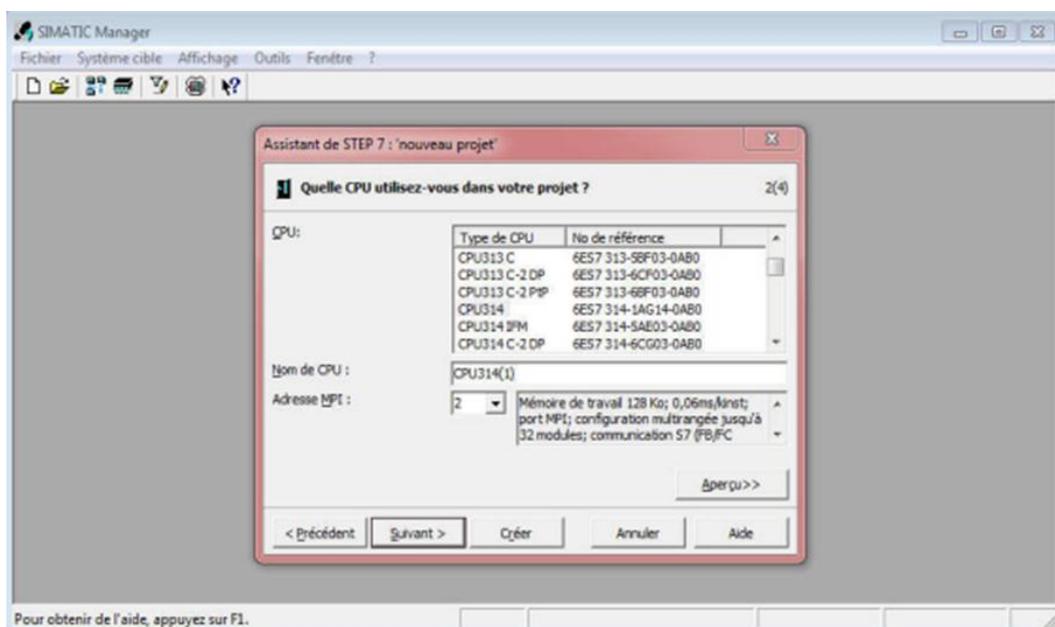


Figure 2.9 : Choix de la CPU.

- 2- Après avoir choisi la CPU, la fenêtre qui apparaît va nous permettre de choisir les blocs à insérer, ainsi que le langage de programmation (LIST, LOG ou CONT). Dans notre cas, nous avons choisi le bloc OB1 (bloc d'organisation) et le langage à contact (CONT) comme langage de programmation.

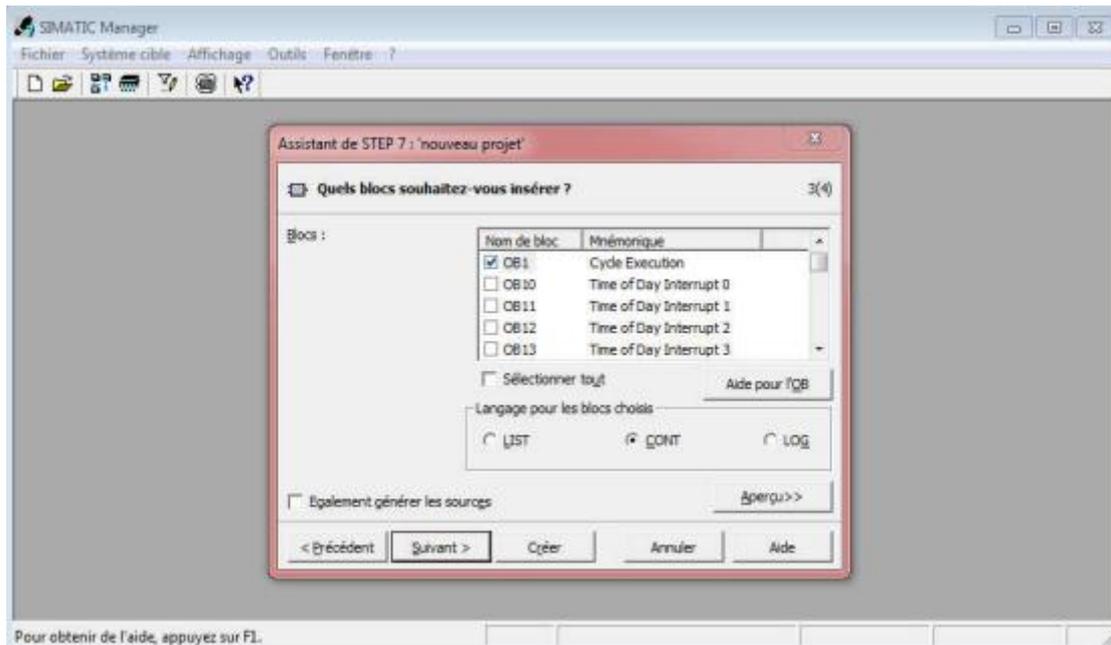


Figure 2.10 : Sélection des blocs et choix du langage.

- 3- En cliquant sur suivant, l'icône de la création de projet apparaît pour le nommer.

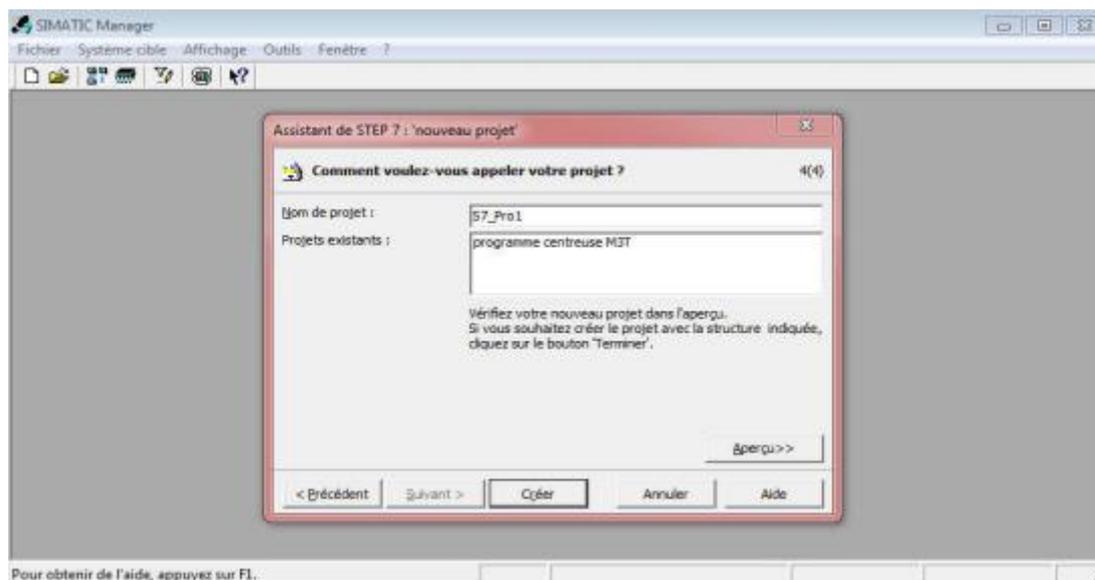


Figure 2.11 : Nomination et création du projet.

- 4- En cliquant sur créer, la fenêtre suivante apparaît.

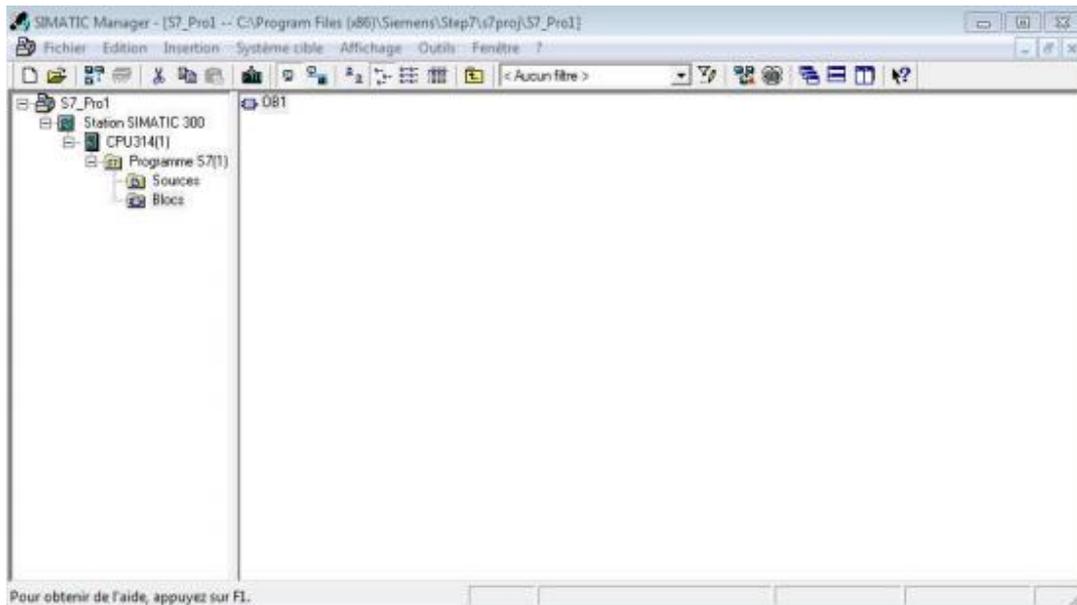


Figure 2.12 : Station SIMATIC S7-300.

### 2.2.8 Barres de : titre, menu et outils de SIMATIC Manager

La barre de titre contient le titre de la fenêtre et des boutons pour contrôler cette fenêtre. La barre de menu contient tous les menus disponibles pour cette fenêtre. Enfin, la barre d'outils contient les tâches les plus fréquemment utilisées sous forme de symboles que l'on liste ci-dessous.

-  Crée un nouveau projet ou une nouvelle bibliothèque.
-  Ouvre le projet ou la bibliothèque sélectionnée.
-  Ouvre une fenêtre affichant les partenaires accessibles.
-  Ouvre une fenêtre pour la carte mémoire.
-  Coupe les objets sélectionnés et les transpose dans le presse-papier.
-  Copie les objets sélectionnés et les transpose dans le presse-papier.
-  Insère le contenu du presse-papier à la position du curseur.
-  Charge l'objet en cours dans le système cible.
-  permet de sélectionner le mode « Hors ligne ».
-  permet de sélectionner le mode « en ligne ».
-  Affiche les objets en utilisant de grandes icônes.
-  Affiche les objets en utilisant de petites icônes.
-  Affiche les objets sous forme de liste.
-  Affiche une information sur les objets de la fenêtre.
-  Permet de passer dans le niveau de la hiérarchie supérieure.
-  Permet d'activer/désactiver la simulation.
-  Affiche une aide contextuelle.

### 2.2.9 Simulateur de programme PLCSIM

L'application de simulation de modules S7-PLCSIM permet d'exécuter et de tester le programme dans un API qu'on simule dans un ordinateur ou dans une console de programmation. La simulation étant complètement réalisée au sein du logiciel STEP7, il n'est pas nécessaire qu'une liaison soit établie avec un matériel S7 quelconque (CPU ou module de signaux). L'application S7 de simulation permet de tester des programmes destinés aux CPU S7-300 et aux CPU S7-400, et de remédier à d'éventuelles erreurs.

S7-PLCSIM dispose d'une interface simple permettant de visualiser et de forcer les différents paramètres utilisés par le programme (comme, par exemple, d'activer ou de désactiver des entrées). Tout en exécutant le programme dans l'AP de simulation, on a également la possibilité de mettre en oeuvre les diverses applications du logiciel STEP7 comme, par exemple, la table des variables afin d'y visualiser et d'y forcer des variables [14].

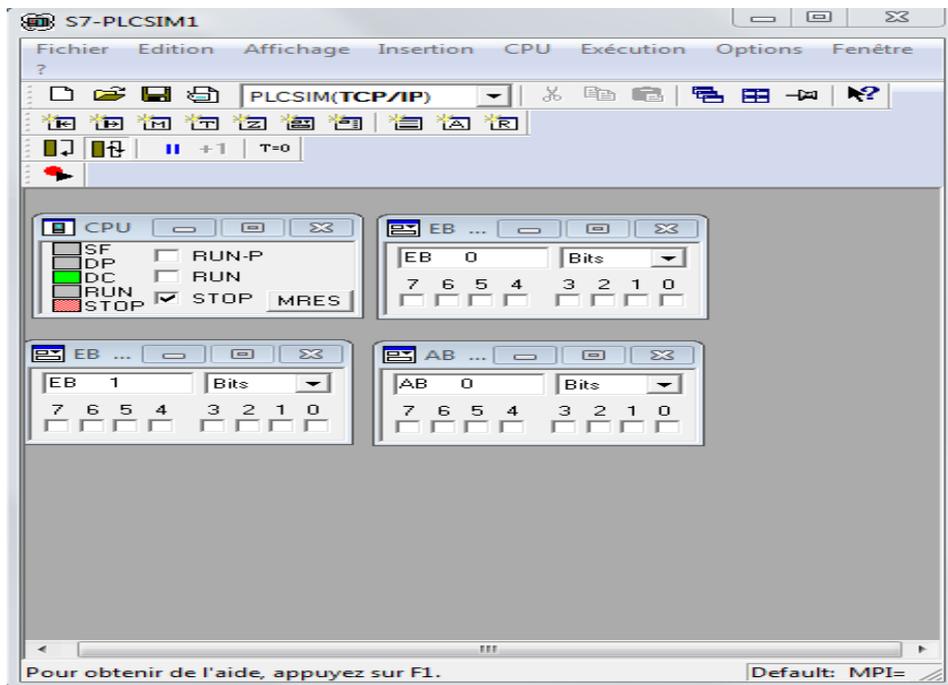


Figure 2.13 : Simulateur S7-PLCSIM.

### 2.2.10 Configuration matérielle

La configuration du matériel se fait via le logiciel dans sa propre zone d'édition. Elle est basée sur une structure en rack et permet de configurer et paramétrer le matériel d'un projet d'automatisation. Il suffit juste de sélectionner le châssis (rack) dans un catalogue électronique et leurs affecter les modules sélectionnés aux emplacements souhaités dans les racks. La figure ci-dessous montre la configuration d'un rack local et un rack distant interconnectés au système via le réseau Profibus.

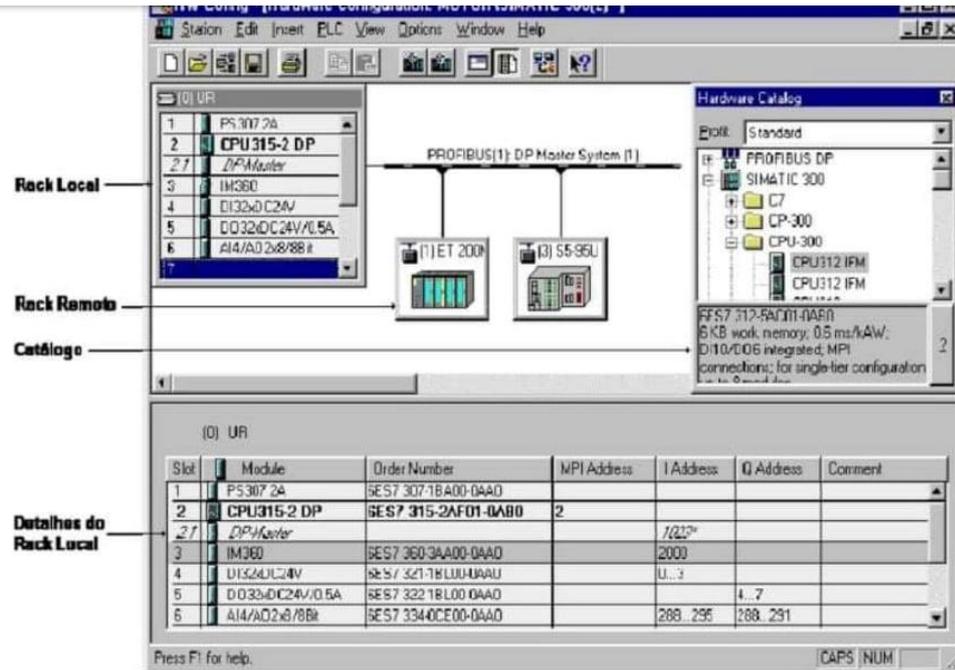


Figure 2.14 : Configuration du matériel.

## 2.3 Le logiciel e supervision WinCC flexible

WinCC est une application de supervision puissante à utiliser avec Microsoft Windows pour élaborer des HMI (Human Machine Interface ou interface homme-machine n français). WinCC permet de superviser les processus partir de différente plateformes (station de travail, PC, pupitre digital ou panel...etc.). Les informations affichées sur HMI sont transmises par les organes de commande (API par exemple).

Pour la supervision des installation industrielles et les infrastructures à grande échelle et sur de longues distances, on parle de la supervision sous l'environnement SCADA (Supervisory Control And Data Acquisition, traduit en français par Système de contrôle et d'acquisition de données).

Il existe plusieurs versions de WinCC dont la plus basic est WinCC flexible qui a été conçue pour crée des HMI qui s'exécute sur des consoles que l'on appelle Panel.

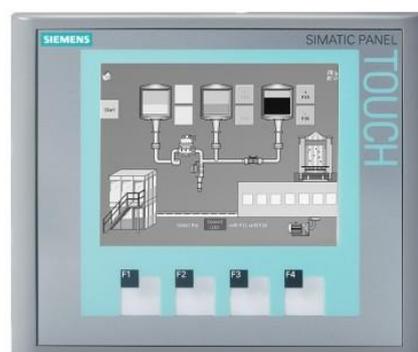


Figure 2.15 : Un panel de supervision.

WinCC flexible offre les avantages suivants :

- Amélioration de la productivité (efficacité de la configuration) pendant la construction des projets HMI
- Réaliser des concepts innovants d'HMI et d'automatisation sous plusieurs réseaux, parmi lesquels TCP/IP et Web
- Disponibilité accrue de nouvelles machines et installations.
- Accéder facilement et en toute sécurité aux données de processus à partir de partout dans le monde

### 2.3.1 Tâches d'un système HMI

Un système IHM se charge des tâches suivantes :

**Représentation du processus :** Le processus est représenté sur le pupitre opératoire. Lorsqu'un état du processus évolue par exemple, l'affichage du pupitre opératoire est mis à jour.

**Commande du processus :** L'opérateur peut commander le processus via l'interface utilisateur graphique, il peut par exemple définir une valeur de consigne pour un automate ou démarrer un moteur.

**Vue des alarmes :** Lorsque des états critiques surviennent dans le processus, une alarme est immédiatement déclenchée, par exemple, lorsqu'un défaut surgit (valeur limite est franchie).

**Archivage de valeurs processus et d'alarmes :** Les alarmes et les valeurs du processus peuvent être archivées par le système HMI, on peut ainsi documenter la marche du processus et accéder ultérieurement aux données de la production écoulée.

**Documentation de valeurs processus et d'alarmes :** Les alarmes et valeurs processus peuvent être éditées par le système HMI sous forme de journal.

**Gestion des paramètres du processus et des machines :** Les paramètres du processus et des machines peuvent être enregistrés au sein du système HMI dans des recettes.

### 2.3.2 Étapes de création un projet WinCC

Tout d'abord, on démarre le logiciel en double cliquant sur l'icône SIMATIC WinCC flexible dans bureau Windows.



Figure 2.16 : Icône de WinCC flexible.

### a) Création d'un nouveau projet

On choisit la commande de menu, puis projet > nouveau et on clique sur aperçu et après on sélectionne créer un projet vide.

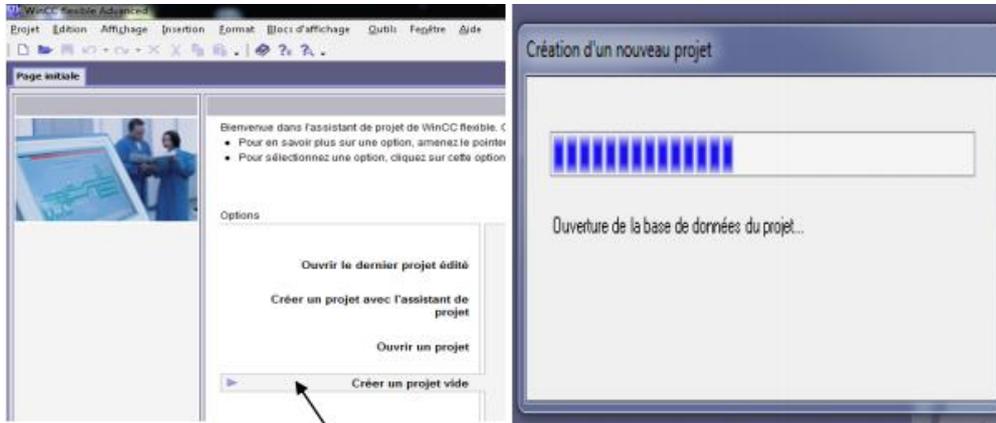


Figure 2.17 : Icône de WinCC flexible.

### b) Sélection de pupitre

On sélectionne le panel choisi et on confirme avec ok.



Figure 2.18 : Sélection de pupitre.

### c) Intégration de WinCC avec Step 7

Une intégration particulièrement étendue est proposée par WinCC flexible avec des composants appartenant aux familles des produits SIMATIC. L'intégration de Step 7 dans l'interface de configuration permet de diminuer la fréquence des erreurs et de réduire les tâches de configuration nécessaires. Durant la configuration, se mettre directement à la table mnémorique de STEP 7 ainsi que les paramètres de communication qui contiennent les adresses et les protocoles de commandes.

On clique sur aperçu puis on sélectionne Projet > Intégration dans projet STEP7 et on confirme avec OK.

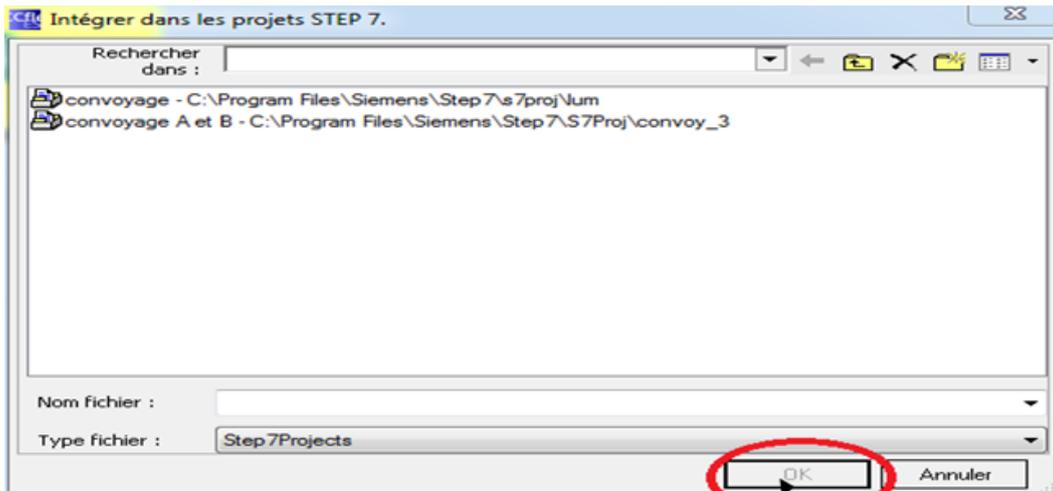


Figure 2.19 : Intégration avec un projet Setep 7.

#### d) Conception du pupitre de supervision

**Création des vues :** Les vues sont les éléments principaux du projet. Elles permettent de commander et de contrôler l'installation. Ces vues contiennent plusieurs éléments comme les objets simples qui englobent des objets graphiques fondamentaux (tel que les lignes), des éléments de commande fondamentaux, des champs d'E/S ou le bouton...etc.

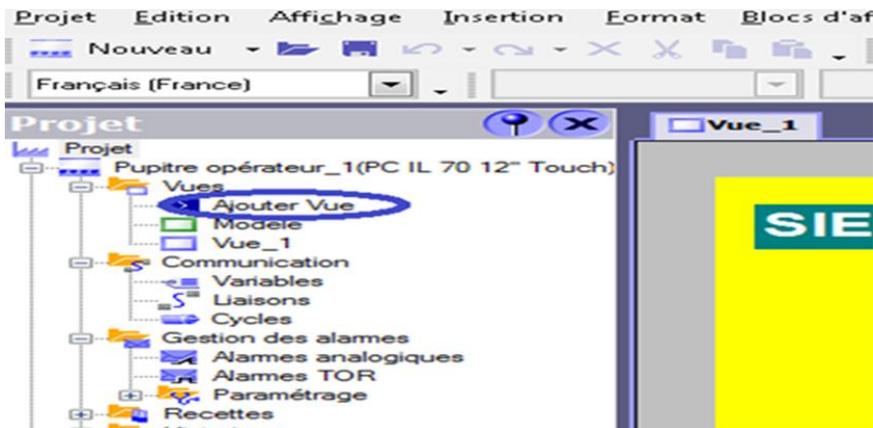


Figure 2.20 : Création des vues.

Dans la fenêtre de choix des objets, il est possible de choisir : Objets simple, Objets complexes, Mes contrôle, Graphique, Bibliothèque.

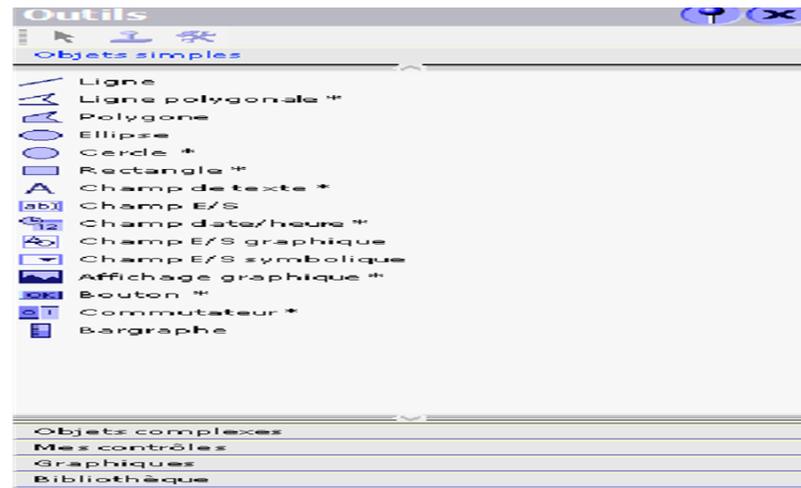


Figure 2.21 : Choix des objets.

**Déclaration des variables :** Les objets créés peuvent être associés à des variables. Pour déclarer ces variables, on clique sur aperçu, puis Communication > variable.

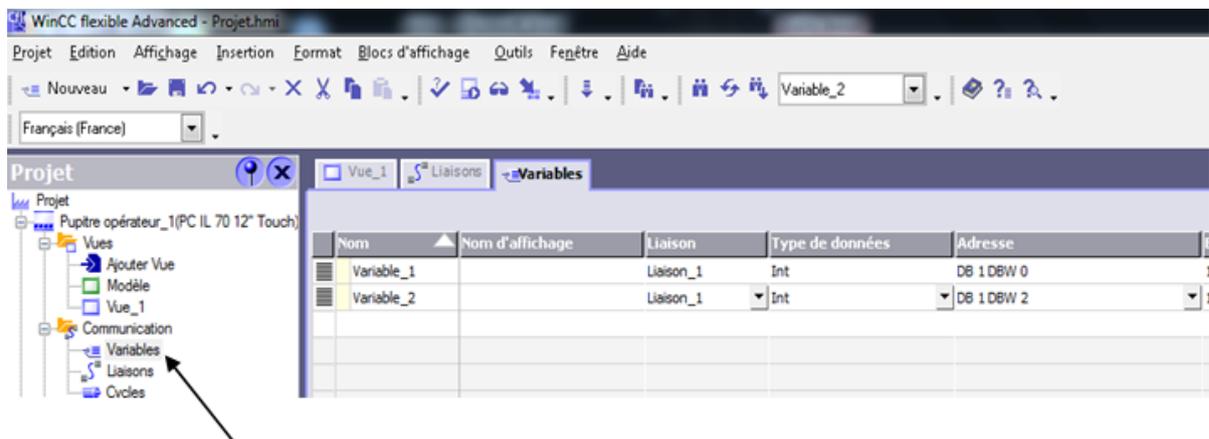


Figure 2.22 : Déclaration des variables.

**Liaison de communication :** Pour définir le protocole de communication entre l'API et le HMI, on clique sur aperçu, puis Communication > liaisons.

## 2.4 Conclusion

Dans ce chapitre, on a présenté les outils logiciels qu'on a utilisés dans la programmation et la supervision de notre projet.

Dans une première partie on a fourni description des outils de base de Step 7 ainsi que les langages de programmation des API qu'il incorpore et les étapes de création d'un projet.

Dans une deuxième partie, on a présenté brièvement le logiciel de supervision WinCC flexible.

# Chapitre 3: Automatisation et supervision d'un système de tri virtuel

## 3.1 Introduction

Dans un projet d'automatisation, la première phase consiste à spécifier un cahier des charges et concevoir le programme correspondant et la deuxième phase consiste à élaborer l'HMI.

Ce chapitre est destiné à présenter notre projet d'automatisation et de supervision d'un système de tri des caisses. La partie programmation a été réalisée en se servant du langage Graph S7 de Simatic Manager Step7 et la conception de HMI a été effectuée en utilisant WinCC flexible.

A fin tester nos programmes d'une manière plus effective à tous les scénarios et les événements qui peuvent survenir dans un système de tri réel, on les a validés sur un système tri de caisses virtuel qu'offre le logiciel ITS PLC.

## 3.2 Description du logiciel ITS PLC

C'est un logiciel éducatif adapté à l'apprentissage de la programmation des API. Basé sur les dernières technologies informatiques, le logiciel ITS PLC rend la simulation des programmes pour API très proche au monde réel au moyen de parties opératives virtuelles qu'il offre. Il contient cinq systèmes industriels virtuels très réalistes pouvant être connectés à un API réel [15].

### 3.2.1 Interface du logiciel

Dans cette partie, on décrit en détail l'interface et les fonctionnalités de l'ITS PLC édition MHJ.

#### e) Menu principal

Au lancement d'ITS PLC MHJ Edition, le menu principal s'affiche. L'interface du logiciel est assez simple et dispose de 3 menus : menu système, options et quitter pour fermer le logiciel.

En appuyant sur Systèmes il nous amène au menu des systèmes industriels ; options pour aller au menu des options.

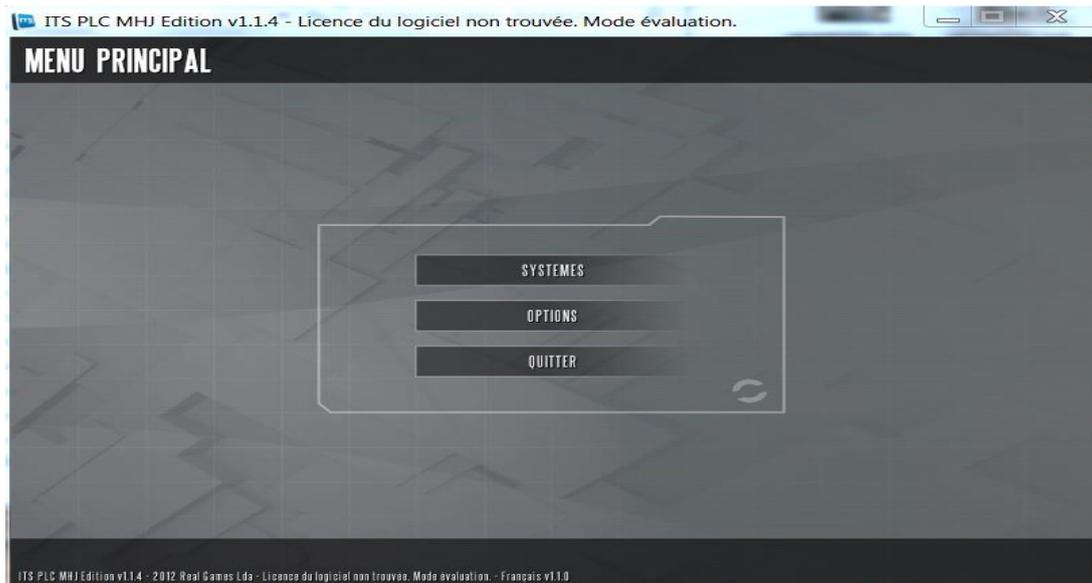


Figure 3.1 : Menu principal.

#### f) Menu des systèmes

Il nous offre cinq systèmes programmables (Figure 3.2) :

- Tri de caisses.
- Mélangeur.
- Palettiseur.
- Robot « pick & place ».
- Magasin automatique.

Pour choisir le système virtuel désiré, on appuie sur Lancer. Si on souhaite revenir au menu principal, on appuie sur Menu Principal.

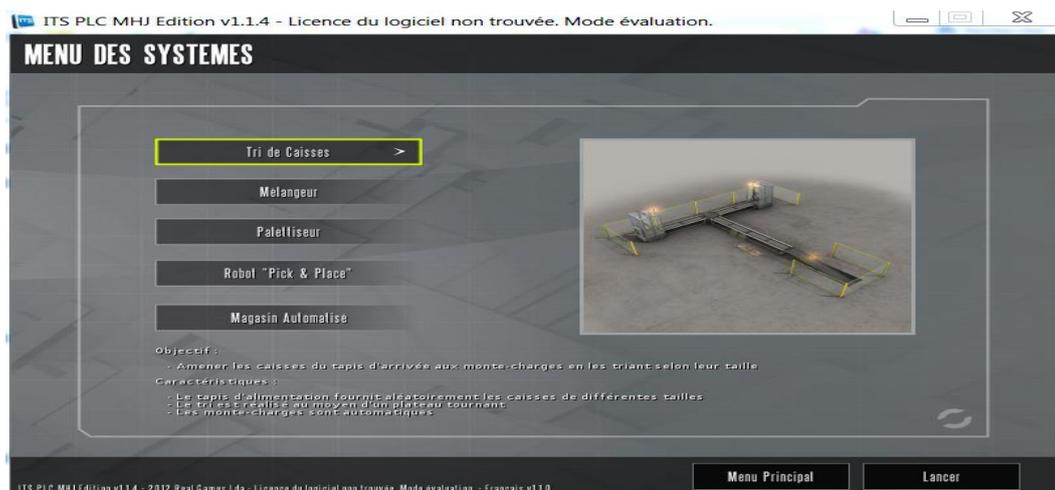


Figure 3.2 : Menu des systèmes.

#### g) Menu des options

Dans le menu des options, on peut choisir plusieurs réglages (Figure 3.3) :

- Langage : Sélectionner une langue.
- Résolution : Sélectionner une résolution compatible avec votre carte graphique.
- Plein écran : oui - mode plein écran, Non - Mode fenêtre.
- Son : Passer du mode Activé (on) au mode Désactivé (off).
- 3Dconnexion : on - pour utiliser le 3Dconnexion SpaceNavigator™ et la souris pour la navigation.
- Effet de lumière (« bloom ») : Activé (on) - pour un rendu plus réaliste de l'éclairage des scènes.

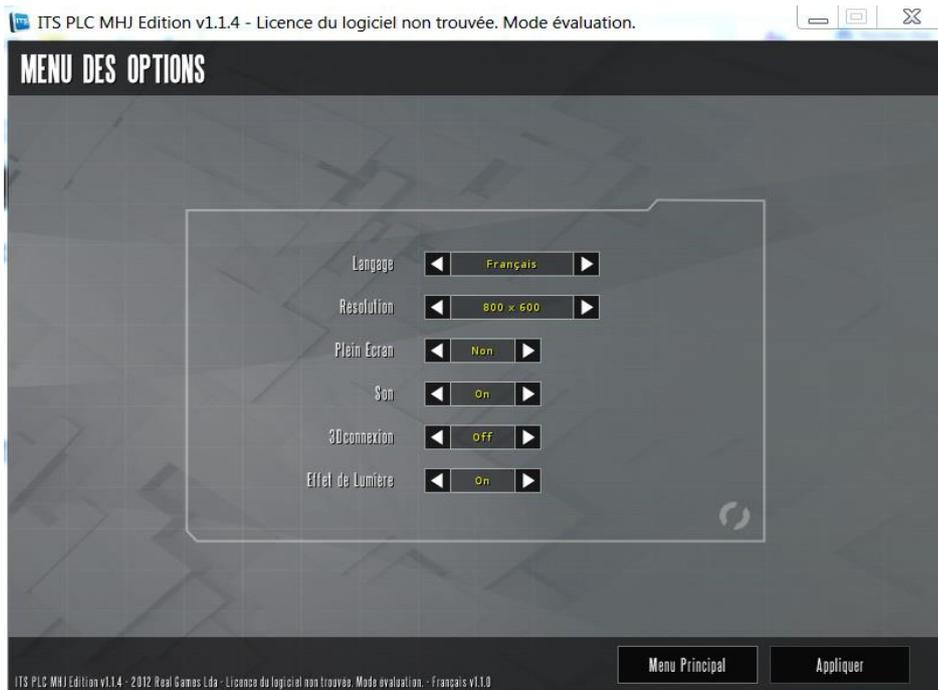


Figure 3.3 : Menu des options.

### 3.2.2 Panneaux des Systèmes

#### a) Panneau utilitaire

Il y trouve les fonctions suivantes :

- Choix d'une des positions prédéfinies des caméras, avec le sélecteur de caméra.
- Appuyer sur capteurs pour afficher les étiquettes des capteurs.
- Appuyer sur Actionneurs pour afficher les étiquettes des actionneurs.
- Appuyer sur Pannes pour afficher le panneau des pannes.
- Appuyer sur Initialisation pour réinitialiser le système.
- Appuyer sur Quitter pour retourner au menu principal.

#### b) Panneau des entrées/sorties

- Montre l'état des capteurs et des actionneurs.
- Permet le pilotage du système en mode manuel.
- L'appui sur les voyants des capteurs force leur état à 1.
- L'appui sur les voyants des actionneurs permet de les forcer à 1 (en mode automatique).

- Si ITS PLC est connecté avec WinSPS-S7, la couleur du petit logo MHJ PLC devient bleue.
- Si une licence valide est trouvée, un cadenas ouvert est affiché en haut à droite de ce panneau. Sinon, le cadenas est affiché fermé.

### c) Panneau choix du mode

Permet de choisir le mode de fonctionnement du système. En mode manuel, la partie opérative est pilotée par l'utilisateur. En mode automatique, le système est commandé par l'API.

### d) Panneau de commande

C'est pour commander le système en mode automatique.

### e) Panneau des pannes

La simulation des pannes capteurs et actionneurs se fait par ce panneau.



Figure 3.4: Panneaux du système (1- panneau utilitaire, 2- panneau des E/S, 3- panneau choix du mode, 4- panneau de commande).

## 3.2.3 Système de Navigation

### a) Souris - clavier

Le bouton droit de la souris et les touches W, S, A, D permettent de déplacer et de tourner la caméra. La molette de la souris permet de monter ou descendre la caméra.

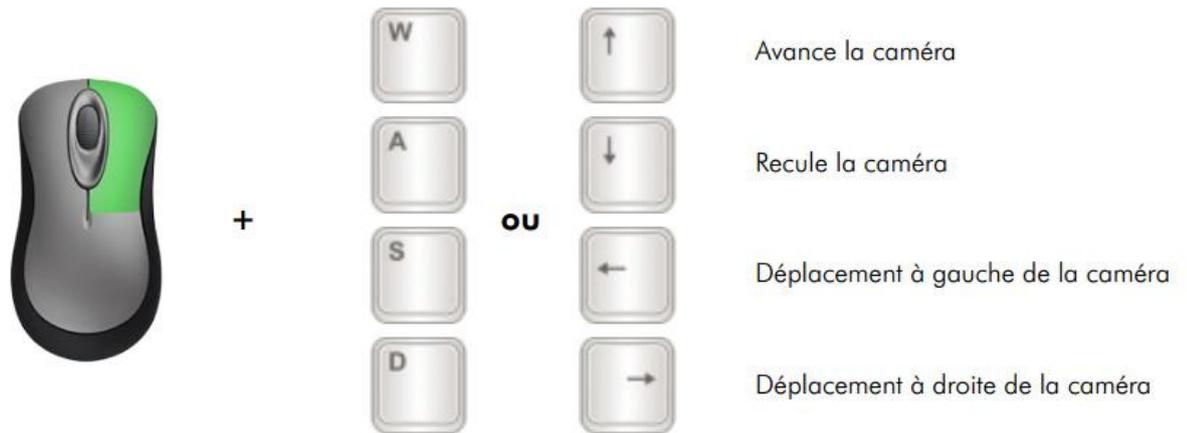


Figure 3.5: Navigation souris-clavier 1.

Pour prendre et déplacer n'importe quel objet mobile (palette, caisse, etc.), saisir l'objet avec le bouton gauche de la souris et l'objet suivra le mouvement de la souris.



Figure 3.6: Navigation souris-clavier 2.

Il est possible de déplacer l'objet et tourner la caméra en même temps en appuyant sur les deux boutons de la souris.

### b) Interactivités Avec les Systèmes

Une des principales caractéristiques d'ITS PLC MHJ Edition est la possibilité d'interagir avec les objets mobiles en temps réel et à n'importe quel moment de la simulation. Les objets mobiles sont identifiables par le pointeur de la souris en forme de cible.

Il suffit de saisir et de déplacer l'objet là où l'on souhaite. Le pointeur en forme de cible indique qu'il s'agit d'un objet mobile.

Avec cette possibilité, on peut :

- Interagir avec le système comme vous le feriez avec un système réel.
- Ajouter et retirer des objets du flux de production, à n'importe quel moment durant la simulation.



*Figure 3.7 : Pointeur en forme de cible.*

- Créer des situations d'erreur ou des embouteillages.
- Tester certaines parties du système ; par exemple : un convoyeur.



*Figure 3.8: Exemples d'interactions avec les systèmes.*

### c) Modes manuel / automatique

Toutes les parties opératives peuvent être commandées en mode manuel par l'utilisateur ou bien en mode automatique par l'API. En mode manuel, l'utilisateur agit de la même façon que la partie commande et peut ainsi appréhender la façon dont le système fonctionne. En mode automatique, la commande est assurée par l'API.

#### **Mode manuel**

- Mettre en marche ou arrêter les actionneurs.
- Forcer l'état des capteurs (signalé par une LED rouge).

Par défaut, tous les systèmes démarrent en mode manuel. À tout moment, on peut passer d'un mode à l'autre.

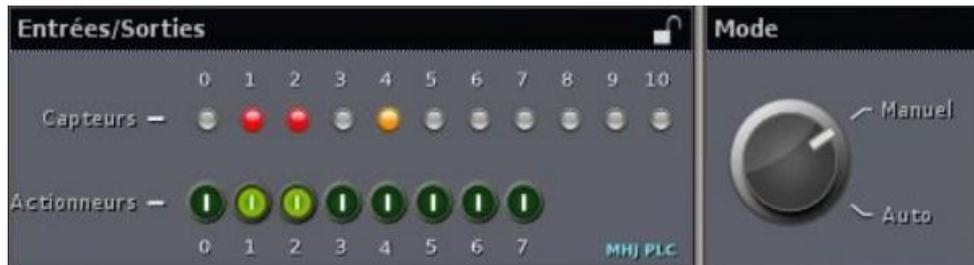


Figure 3.9: Panneau des E/S en mode manuel.

### Mode automatique

- Forcer les actionneurs (signalés par un bouton bleu), i.e. les sorties de l'API sont ignorées.
- Forcer les capteurs (signalés par une LED rouge).

En mode automatique, le panneau de commande devient visible. Il est utile pour piloter l'automate.



Figure 3.10: Panneau des entrées-sorties en mode automatique.

**Start** : démarre le système.

**Stop** : arrête le système (contact normalement fermé).

**Reset** : réinitialise le système.

**Urgence** : arrêt d'urgence (contact normalement fermé).

### 3.3 Description du système de tri de caisses

Il s'agit d'un système de tri où le but principal est de transporter des caisses d'un convoyeur d'alimentation vers des monte-charge en les triant en fonction de leur taille.

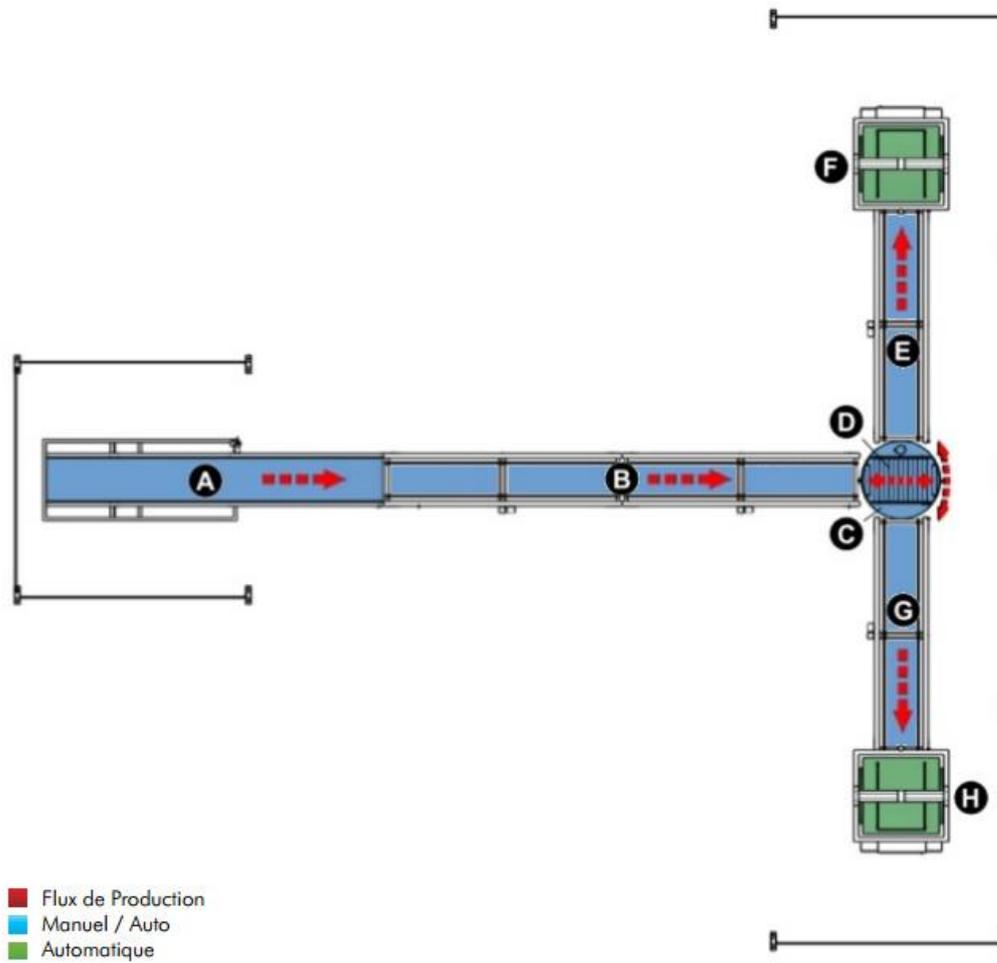


Figure 3.11: Système de tri des caisses.

**PETITE CAISSE**

**GRANDE CAISSE**



Figure 3.12: Exemple sur la taille de caisses.

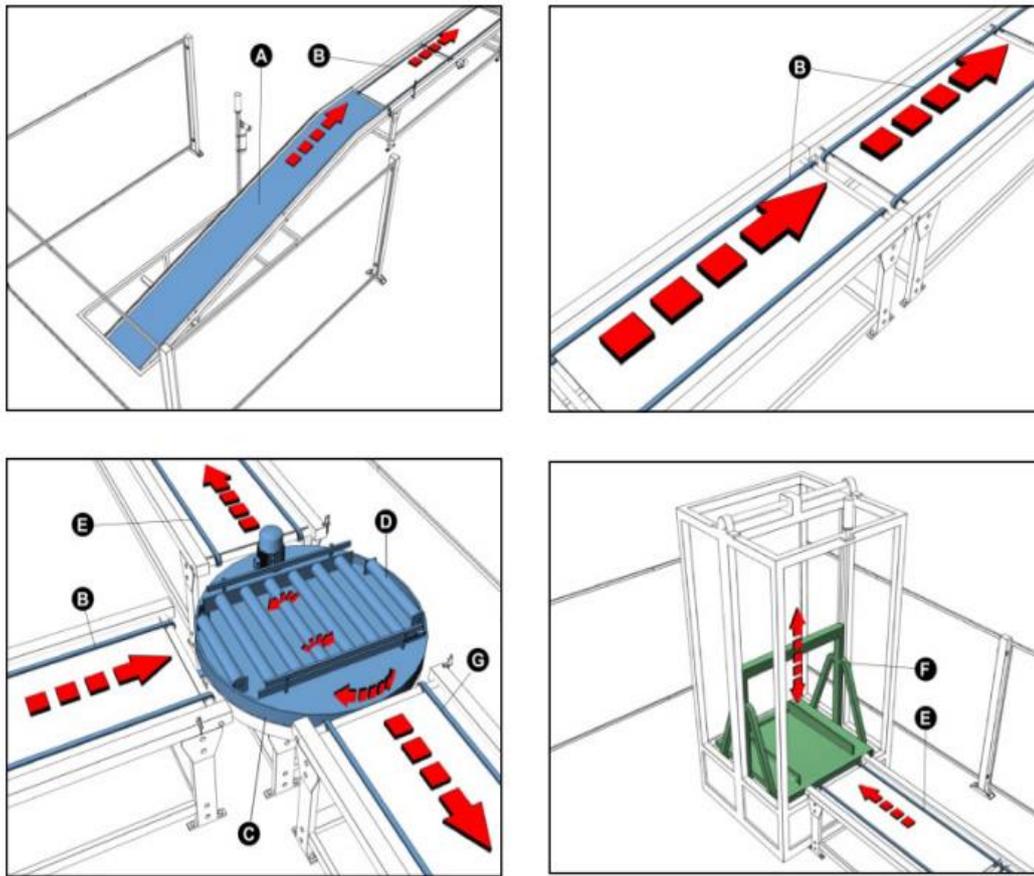


Figure 3.13 représentation des mouvements dans le système du tri.

Les actionneurs disponibles pour commander le système (sorties) sont listés dans la tableau suivant.

Adresse	Actionneur	Commentaire
Q0.0	Actionneur 0	Tapis d'alimentation
Q0.1	Actionneur 1	Convoyeur à bande
Q0.2	Actionneur 2	Rouleaux du plateau tournant (chargement)
Q0.3	Actionneur 3	Rouleaux du plateau tournant (déchargement)
Q0.4	Actionneur 4	Plateau tournant
Q0.5	Actionneur 5	Convoyeur de sortie
Q0.6	Actionneur 6	Convoyeur de sortie
Q1.0	Sortie	LED Bouton Start
Q1.1	Sortie	LED Bouton Reset

Tableau 3.1: Les actionneurs disponibles.

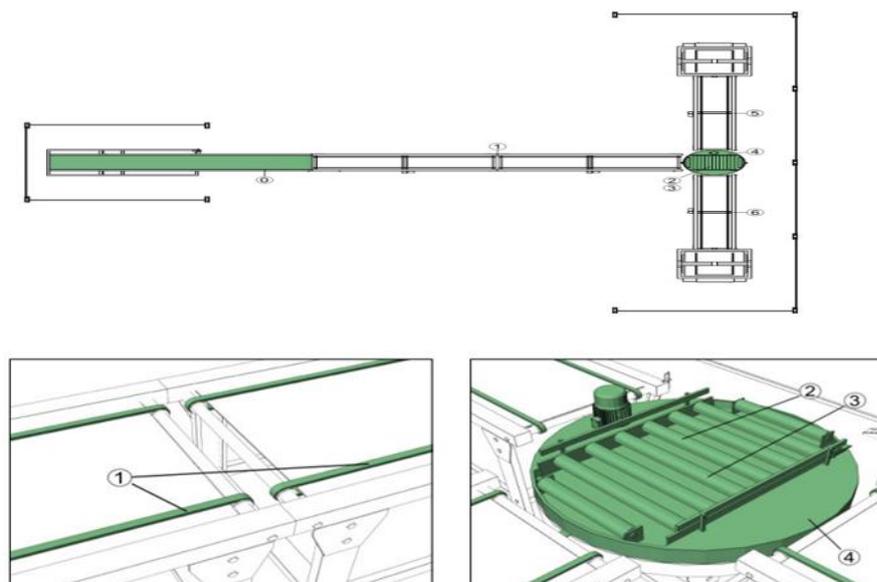


Figure 3.14: Quelques actionneurs disponibles (1- Convoyeur à bande, 2- rouleaux du plateau tournant (chargement), 3- rouleaux du plateau tournant (déchargement), 4- plateau tournant).

Les entrées disponibles (capteur et boutons) sont décrites dans le tableau 3.2.

Adresse	Capteur	Commentaire
I0.0	Capteur 0	Capteur de fin du convoyeur d'alimentation
I0.1	Capteur 1	Capteur petite caisse
I0.2	Capteur 2	Capteur grande caisse
I0.3	Capteur 3	Fin du convoyeur à bande
I0.4	Capteur 4	Plateau tournant en position chargement
I0.5	Capteur 5	Plateau tournant en position déchargement
I0.6	Capteur 6	Présence caisse sur le plateau tournant
I0.7	Capteur 7	Capteur au début du convoyeur de sortie
I1.0	Capteur 8	Capteur au début du convoyeur de sortie
I1.1	Capteur 9	Capteur à la fin du convoyeur de sortie
I1.2	Capteur 10	Capteur à la fin du convoyeur de sortie
I1.3	Mode auto	Mode automatique
I1.4	Start	Bouton Start
I1.5	Stop	Bouton stop
I1.6	Reset	Bouton reset
I1.7	Arrêt d'urgence	Bouton d'arrêt d'urgence

Tableau 3.2 : Les entrées disponibles.

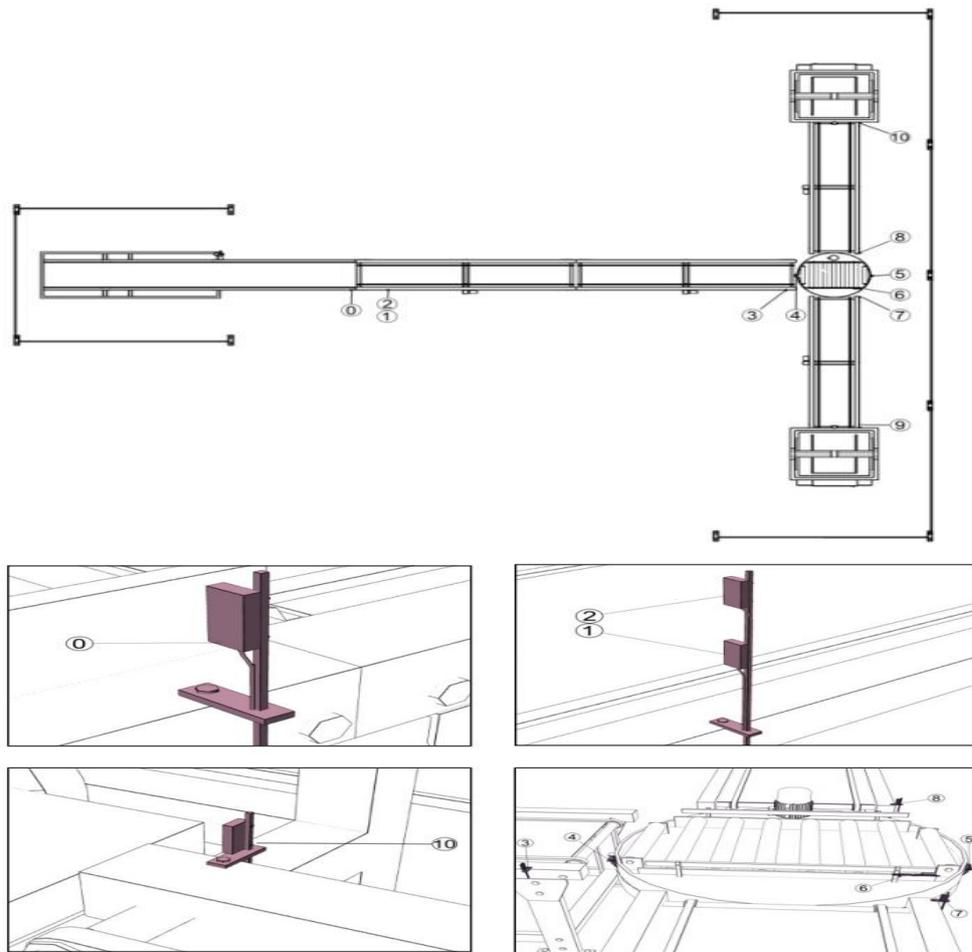


Figure 3.15 : Représentation de quelques capteurs disponibles (0- capteur de fin du convoyeur d'alimentation, 1,2- capteurs des tailles des caisses, 10- capteur à la fin du convoyeur de sortie).

### 3.4 Cahier des charges

Lorsqu'on appuie sur le bouton Start, le convoyeur d'alimentation (A) délivre aléatoirement des petites et des grandes caisses, chargées sur des palettes. Ces dernières sont transportées au moyen d'un convoyeur à bande (B) vers un plateau tournant (C) et doivent ensuite être chargées sur les rouleaux (D). Les palettes effectuent ensuite une rotation de 90°, au moyen du plateau tournant. La taille de la caisse est détectée à l'entrée du convoyeur à bande (B). Les palettes sont envoyées sur les convoyeurs à bande (E ou G) en fonction de leur taille par la mise en rotation des rouleaux dans un sens ou dans l'autre. Enfin, les palettes sont évacuées automatiquement dès qu'elles atteignent un monte-charge (F ou H).

Lorsqu'on appuie sur le bouton Stop, l'actionneur du convoyeur d'alimentation doit s'arrêter immédiatement, et l'arrêt des autres actionneurs après la fin de la tâche. En cas de panne, l'appui sur le bouton d'Arrêt d'urgence permet d'arrêter tous les actionneurs du système à l'exception de l'actionneur de la table tournante. Après le réglage de la panne, on désactive l'Arrêt d'urgence et on appuie sur le bouton Reset

pour réinitialiser tout le système. Pour redémarrer le système il faut appuyer de nouveau sur le bouton Start.

Il est également demandé de créer une partie IHM avec des boutons Start, Stop, Reset et Arrêt d'urgence permettant de poursuivre le fonctionnement du système de tri.

### 3.5 Modélisation par Grafcet

La modélisation par Grafcet de notre système de tri virtuel est composée par un grafcet maître G1 et des Grafcets axillaires (Grafcets de tâches) G10, G20, G30, G40, G50, G60 et G70. De plus, on a utilisé les variables internes suivantes :

- ACT1\_ON : Etat de l'actionneur 1.
- ACT5\_ON : Etat de l'actionneur 5.
- ACT6\_ON : Etat de l'actionneur 6.
- TAB\_ON : Etat de la table tournante.

#### 3.5.1 Grafcets auxiliaires

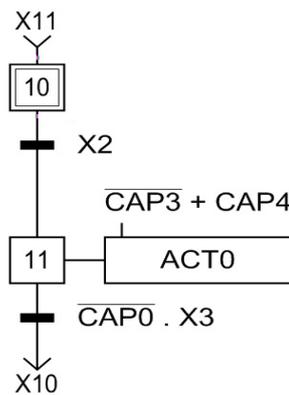


Figure 3.16: Tapi d'alimentation (convoyeur A) \_ Grafcet G10.

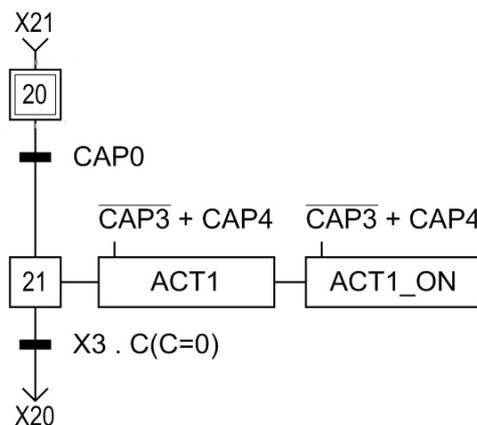


Figure 3.17: Convoyeur B\_ Grafcet G20.

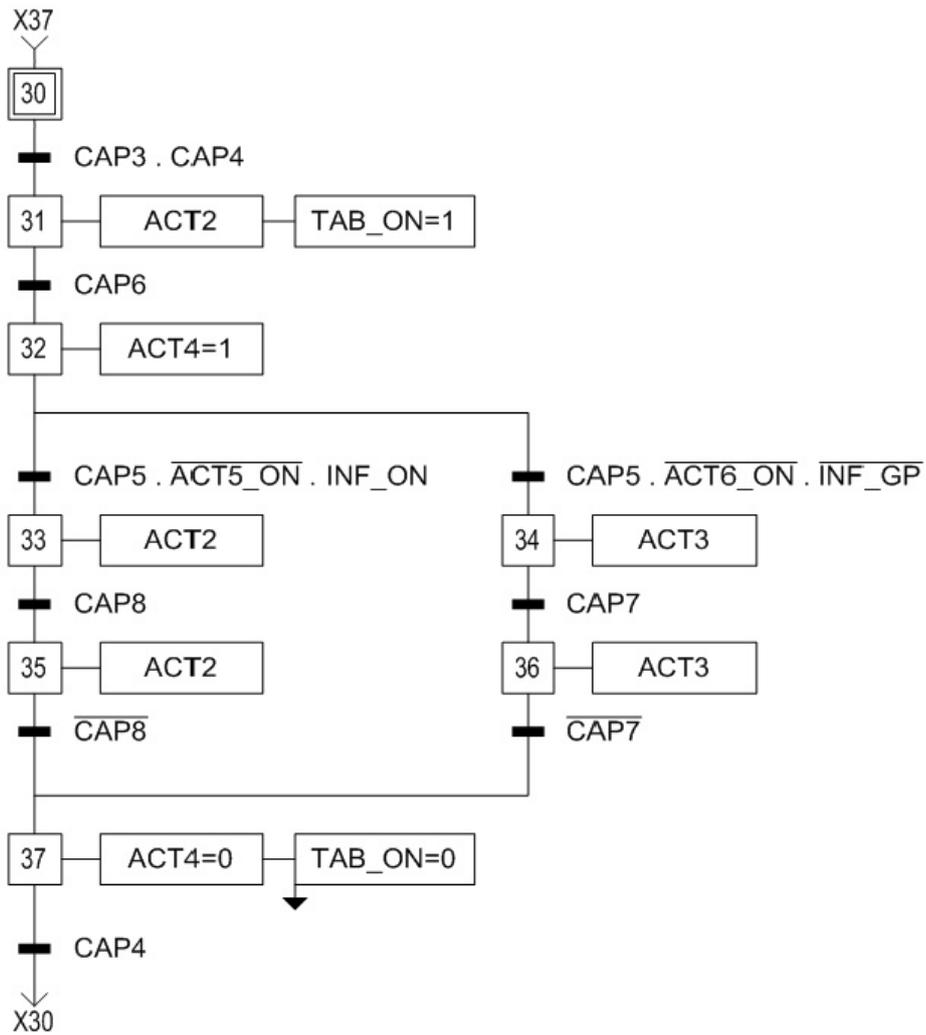


Figure 3.18: Table tournante\_Grafcet G30.

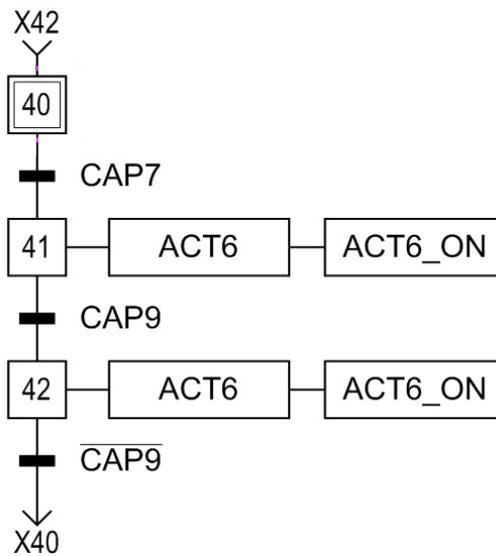


Figure 3.19: Tapi droit\_Grafcet G40.

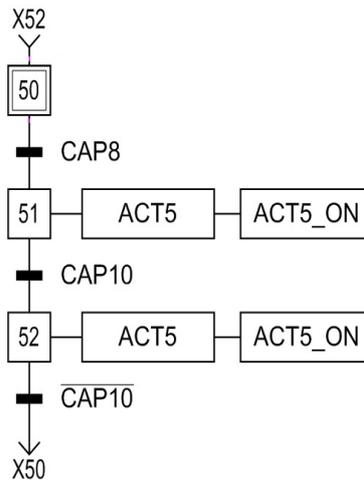


Figure 3.20: Tapi gauche (Grafcet G50).

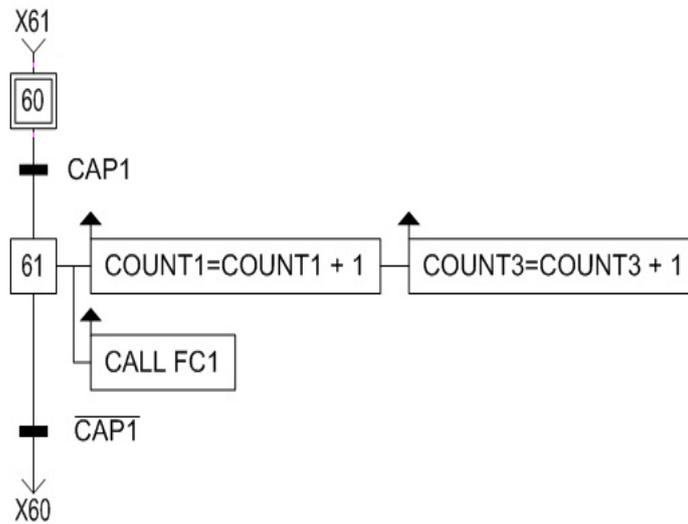


Figure 3.21 : Mémorisation de la taille et le comptage des cartons rentrant (Grafcet G60).

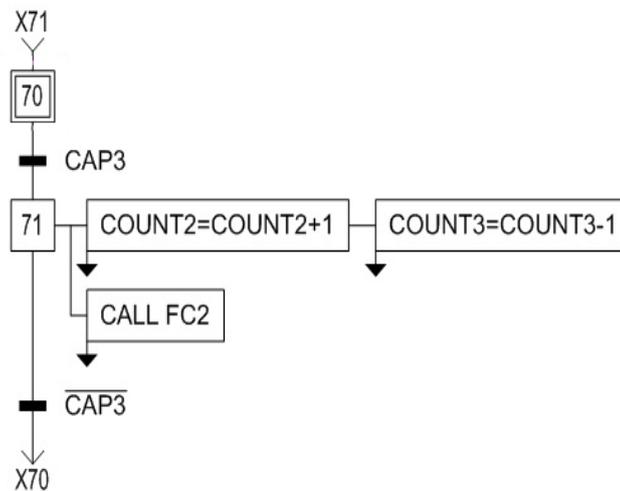


Figure 3.22 : Lecture de la taille et le comptage des cartons sortants (Grafcet G70).

La fonction FC1 contient le programme en langage Ladder de la mémorisation de la taille des cartons à l'entrée du convoyeur B (Figure 3.34). Cette fonction utilise le contenu du compteur COUNT1 et les capteurs CAP1 et CAP2 pour la mémorisation de la taille de chaque carton qui entre dans le convoyeur B. Cette information est mémorisée dans les variables internes CAR1, CAR2 et CAR3 (dans notre cas, le système fonctionne de telle sorte que le convoyeur B ne peut pas avoir plus de 3 cartons en même temps).

La fonction FC2 contient le programme en langage Ladder qui permet la lecture de la taille des cartons à la sortie du convoyeur B (Figure 3.35). Cette fonction utilise le contenu du compteur COUNT2 pour deviner la taille de chaque carton qui sort du convoyeur B. L'information est représentée par la variable interne INF\_GP.

Le compteur/décompteur COUNT3 indique le nombre de cartons dans le convoyeur B.

### 3.5.2 Grafcet principal

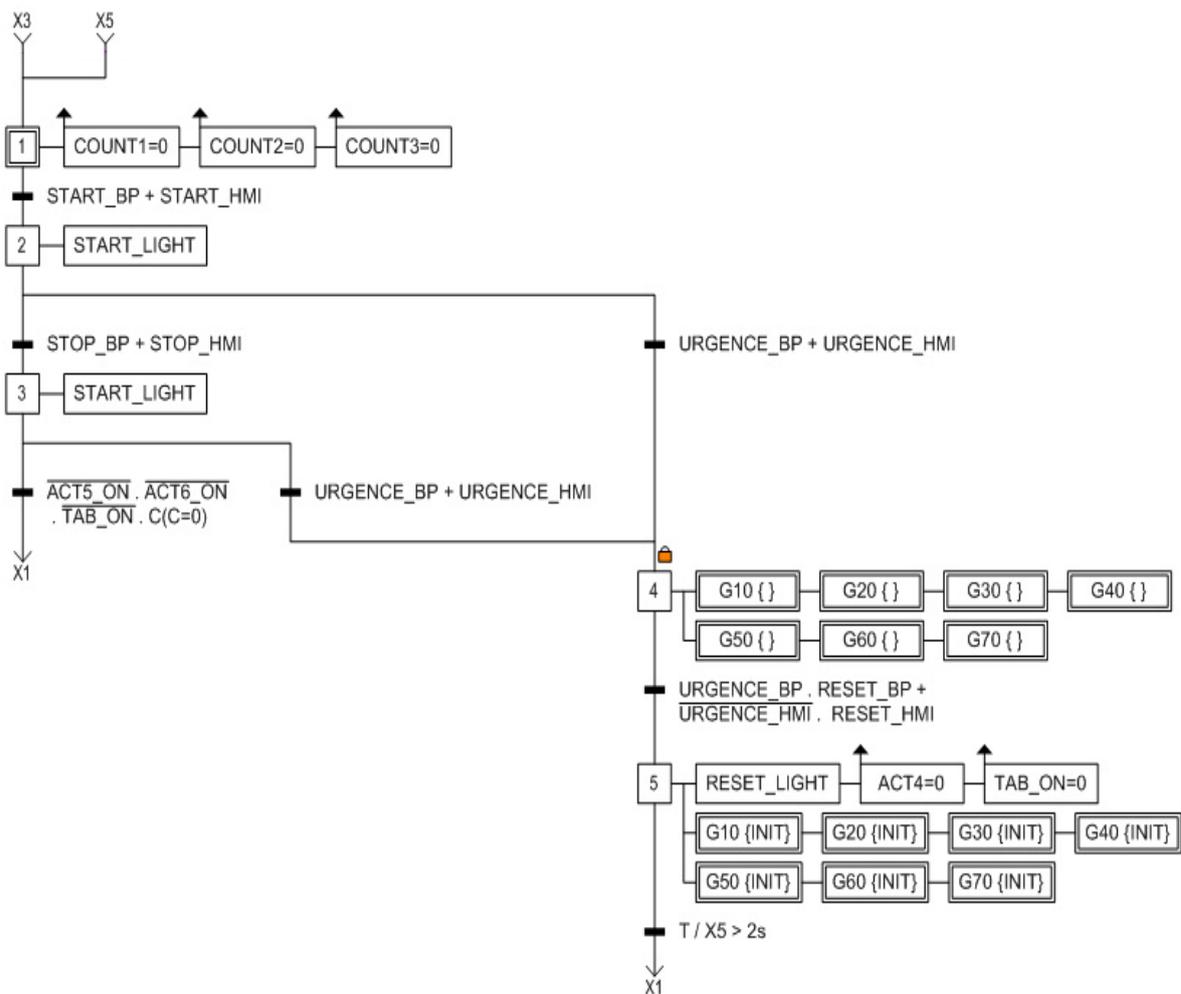


Figure 1.23 : Graphe maitre (Grafcet G1).

## 3.6 Réalisation du programme

Le logiciel SIMATIC Manager permet une programmation multi langage, c'est-à-dire, on peut programmer l'application avec plusieurs langages différents, et même mélanger plusieurs langages dans un même programme.

### 3.6.1 Interface du projet

Le programme S7 que nous avons créé se compose de : un bloc d'organisation OB1, un bloc de données DB1, un bloc fonctionnel FB1 et des fonctions FC.

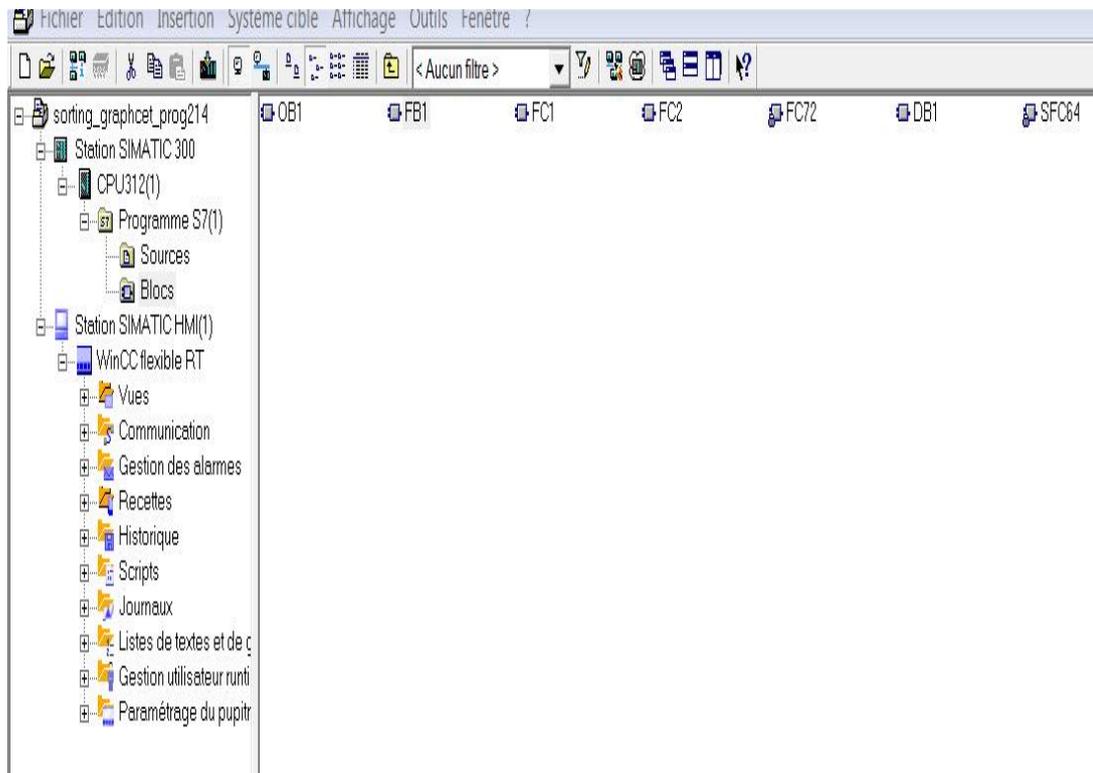


Figure 3.23 Interface du projet.

### 3.6.2 Table des mnémoniques

Au début et avant de procéder à la programmation, nous créons une table des mnémoniques pour faciliter la programmation. La Figure 3.24 représente notre table des mnémoniques.

Table Edition Insertion Affichage Outils Fenêtre ?					
Tous les mnémoniques					
	Etat	Mnémonique /	Opérande	Type de d	Commentaire
1		ACT0	Q 0.0	BOOL	Tapis d'alimentation
2		ACT1	Q 0.1	BOOL	Convoyeur à bande B
3		ACT1_ON	M 0.0	BOOL	
4		ACT2	Q 0.2	BOOL	Rouleaux du plateau tournant (chargement)
5		ACT3	Q 0.3	BOOL	Rouleaux du plateau tournant (déchargement)
6		ACT4	Q 0.4	BOOL	Plateau tournant
7		ACT5	Q 0.5	BOOL	Convoyeur de sortie droite
8		ACT5_ON	M 0.2	BOOL	
9		ACT6	Q 0.6	BOOL	Convoyeur de sortie gauche
10		ACT6_ON	M 0.3	BOOL	
11		CAP0	I 0.0	BOOL	Capteur de fin du convoyeur d'alimentation
12		CAP1	I 0.1	BOOL	Capteur petite caisse
13		CAP10	I 1.2	BOOL	Capteur à la fin du convoyeur de sortie
14		CAP2	I 0.2	BOOL	Capteur grande caisse
15		CAP3	I 0.3	BOOL	Fin du convoyeur à bande
16		CAP4	I 0.4	BOOL	Plateau tournant en position chargement
17		CAP5	I 0.5	BOOL	Plateau tournant en position déchargement
18		CAP6	I 0.6	BOOL	Présence caisse sur le plateau tournant
19		CAP7	I 0.7	BOOL	Capteur au début du convoyeur de sortie
20		CAP8	I 1.0	BOOL	Capteur au début du convoyeur de sortie
21		CAP9	I 1.1	BOOL	Capteur à la fin du convoyeur de sortie
22		CAR1	M 0.4	BOOL	
23		CAR2	M 0.5	BOOL	
24		CAR3	M 0.6	BOOL	
25		COUNT1	C 1	COUNTER	comptage des cartons rentrants
26		COUNT2	C 2	COUNTER	comptage des cartons sortants
27		COUNT3	C 3	COUNTER	comptage des cartons dans le convoyeur B
28		CU1	MW 2	INT	
29		CU2	MW 4	INT	
30		Cycle Execution	OB 1	OB 1	
31		G7_STD_3	FC 72	FC 72	
32		INF_GP	M 0.7	BOOL	
33		MODE_AUTO	I 1.3	BOOL	
34		RESET_BP	I 1.6	BOOL	
35		RESET_HMI	M 1.2	BOOL	
36		RESET_LIGHT	Q 1.1	BOOL	
37		START_BP	I 1.4	BOOL	
38		START_HMI	M 1.0	BOOL	
39		START_LIGHT	Q 1.0	BOOL	
40		STOP_BP	I 1.5	BOOL	
41		STOP_HMI	M 1.1	BOOL	

Pour obtenir de l'aide, appuyez sur F1

Figure 3.24 : Table des mnémoniques.

### 3.6.3 Contenu de bloc fonctionnel FB1

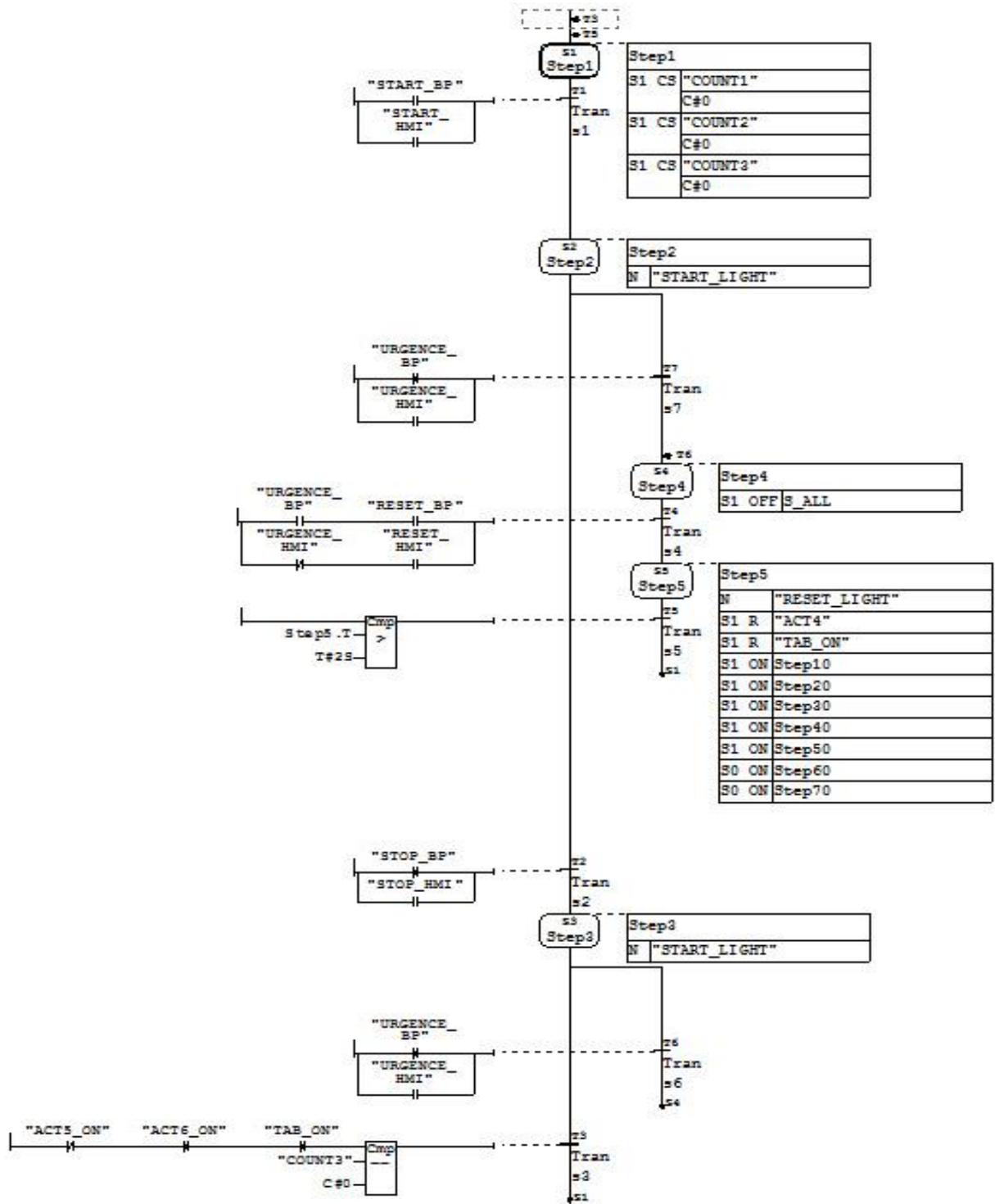


Figure 3.25: Grafcet maitre.

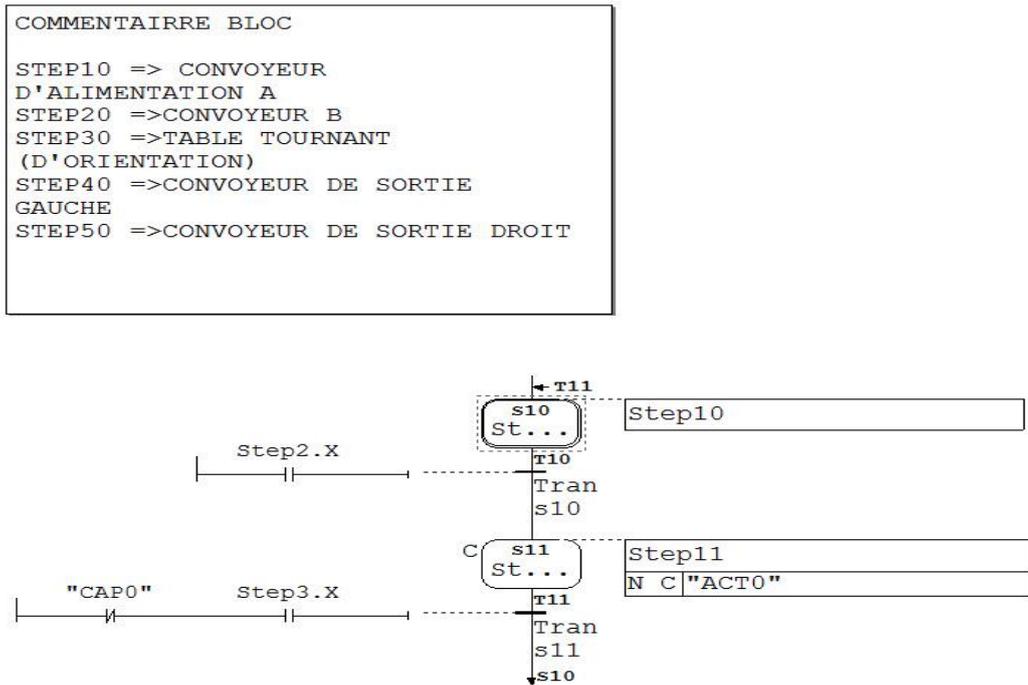


Figure 3.26: Convoyeur d'alimentation A.

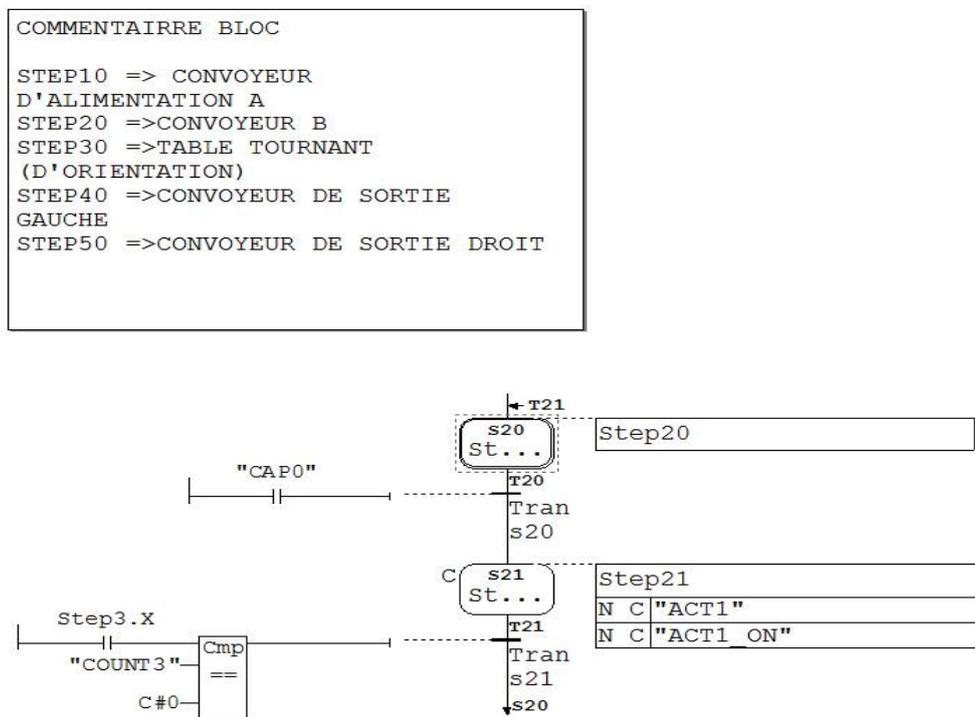


Figure 3.27: Convoyeur B.

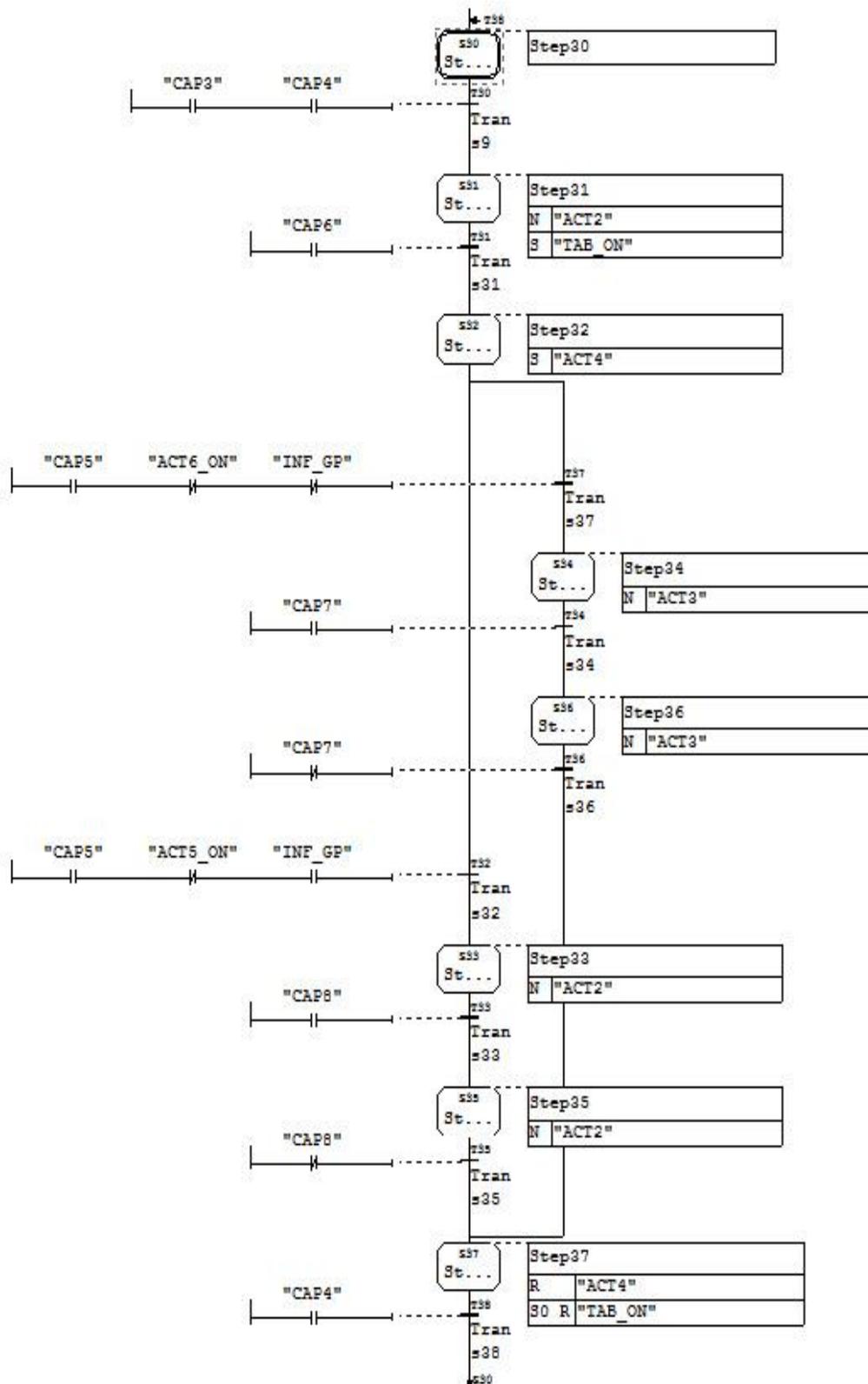


Figure 3.28: Table tournante.

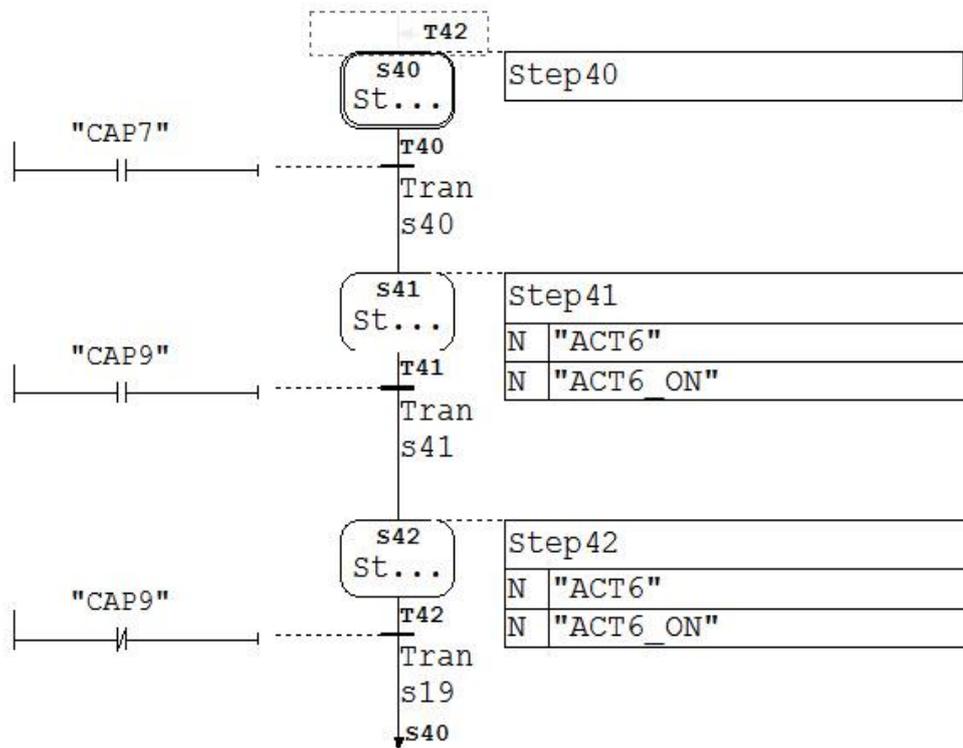


Figure 3.29: Convoyeur de sortie gauche.

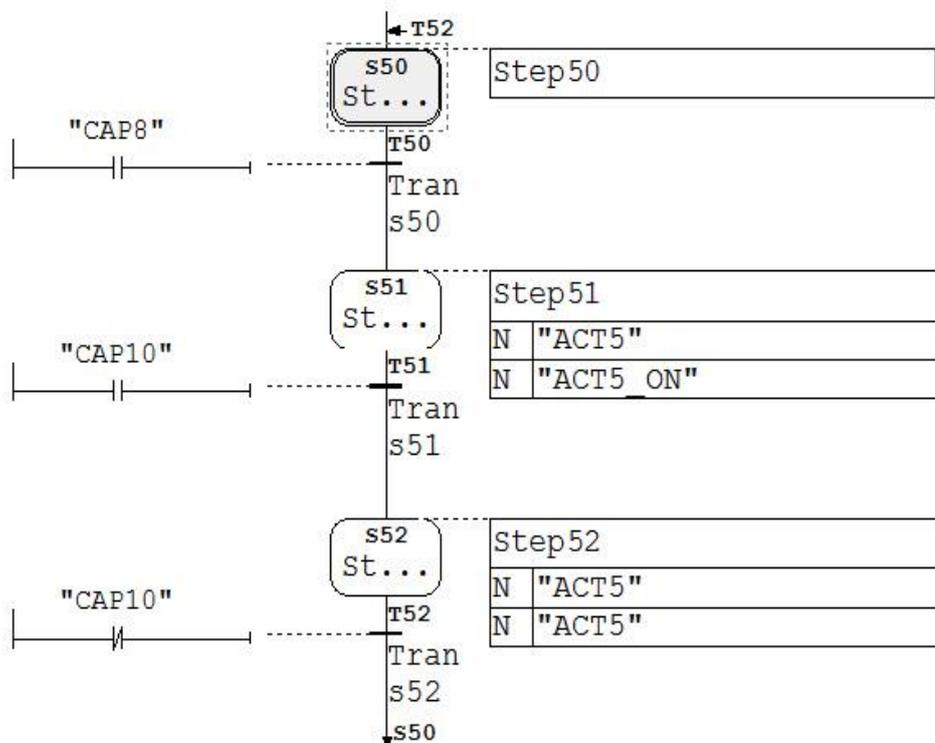


Figure 3.30: Convoyeur de sortie droite.

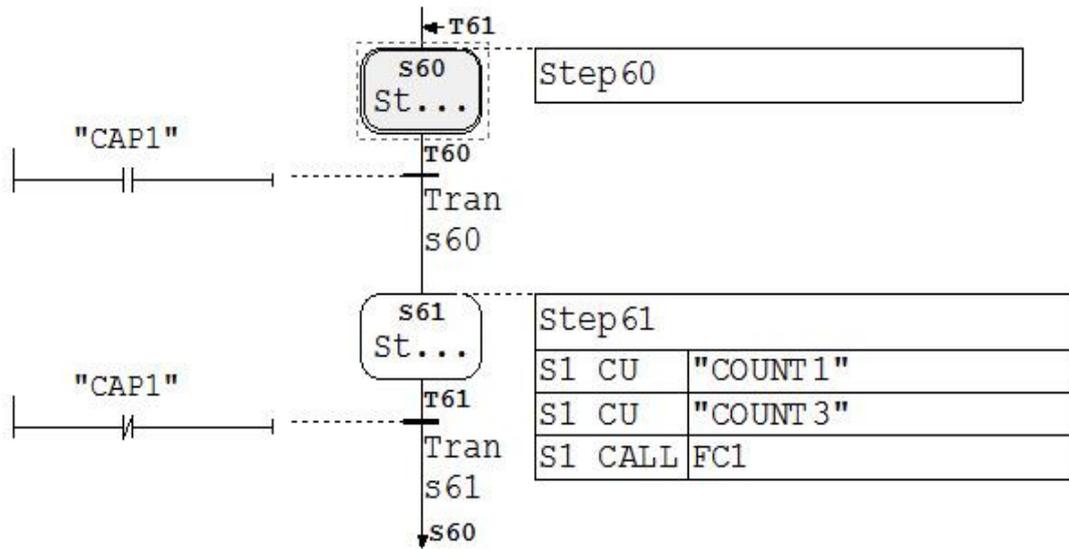


Figure 3.31: Mémorisation de la taille des caisses et le comptage des caisses rentrantes.

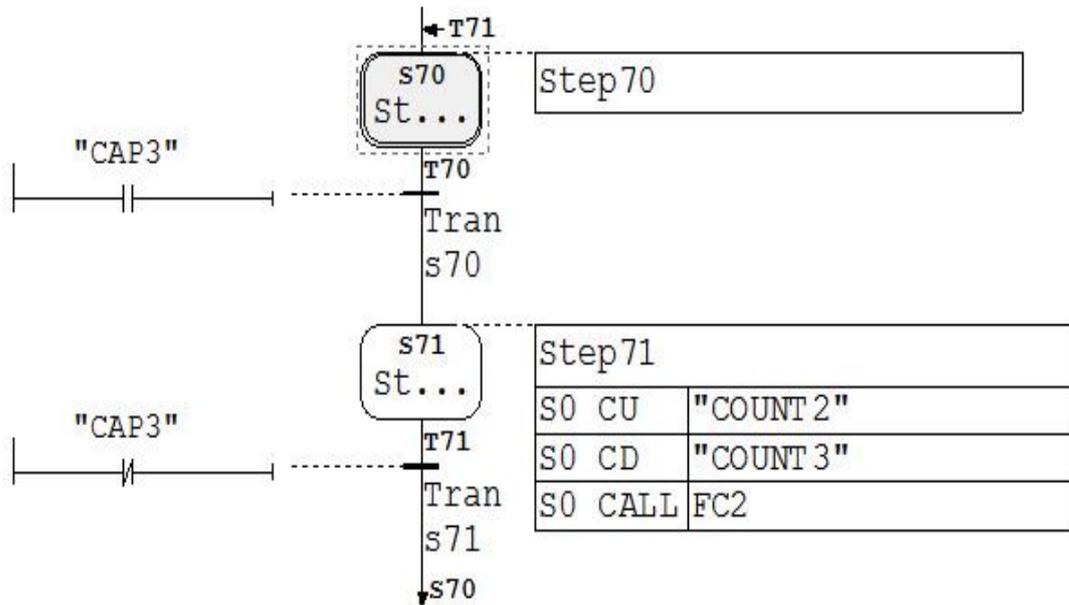


Figure 3.32: Lecture de la taille des caisses et le comptage des caisses sortantes.

### 3.6.4 Contenu des fonctions FC (FC1, FC2)

#### FC1 (figure3.33) :

- **Réseau 1** : comme on ne peut pas comparer le contenu d'un compteur avec une valeur entier on utilise alors le bloc MOVE pour copier la valeur du contenu du compteur COUNT1 dans la variable entière CU1
- **Réseau 2** : lorsque CU1= 1 en mémorise la taille du carton correspondant dans la variable car1 (car1 = cap2)
- **Réseau 3** : lorsque CU1= 2 en mémorise la taille du carton correspondant dans la variable car1 (car2 = cap2)
- **Réseau 4** : lorsque CU1= 3 en mémorise la taille du carton correspondant dans la variable car1 (car3 = cap2), et en effectue une remis à 0 du COUNT 1.

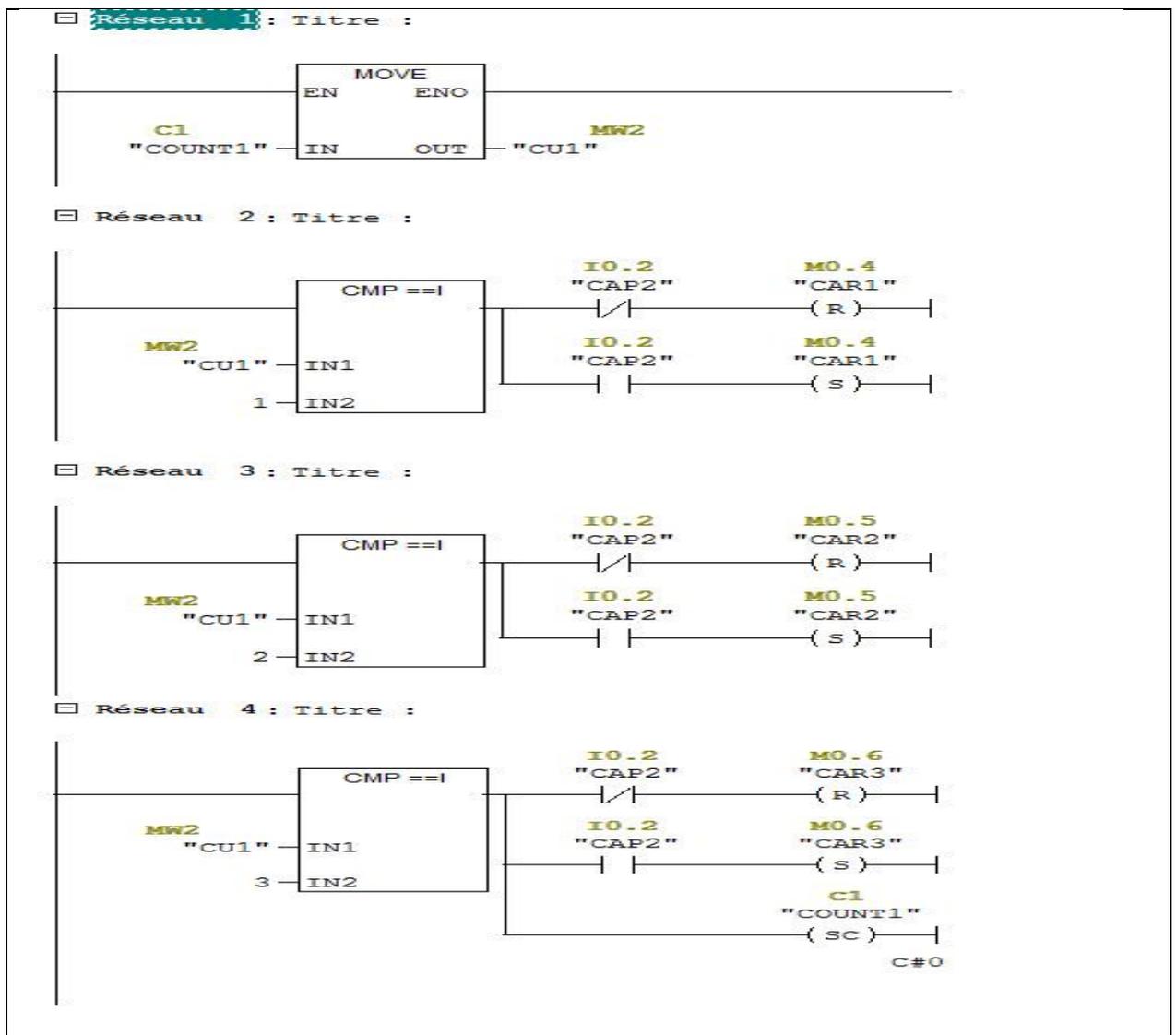


Figure 3.33: Contenu de FC1.

**FC2 (figure3.34) :**

**Réseau 1 :** on utilise le bloc MOVE pour copier la variable du contenu du compteur COUNT2 dans la variable entière CU2

**Réseau 2 :** lorsque CU2= 1 on lit la taille du carton en sortie du convoyeur B par INF-GP = CAR1.

**Réseau 3 :** lorsque CU2= 2 on lit la taille du carton en sortie du convoyeur B par INF-GP = CAR2.

**Réseau 4 :** lorsque CU2= 3 on lit la taille du carton en sortie du convoyeur B par INF-GP = CAR3, et on effectue une remise à 0 du COUNT 2.

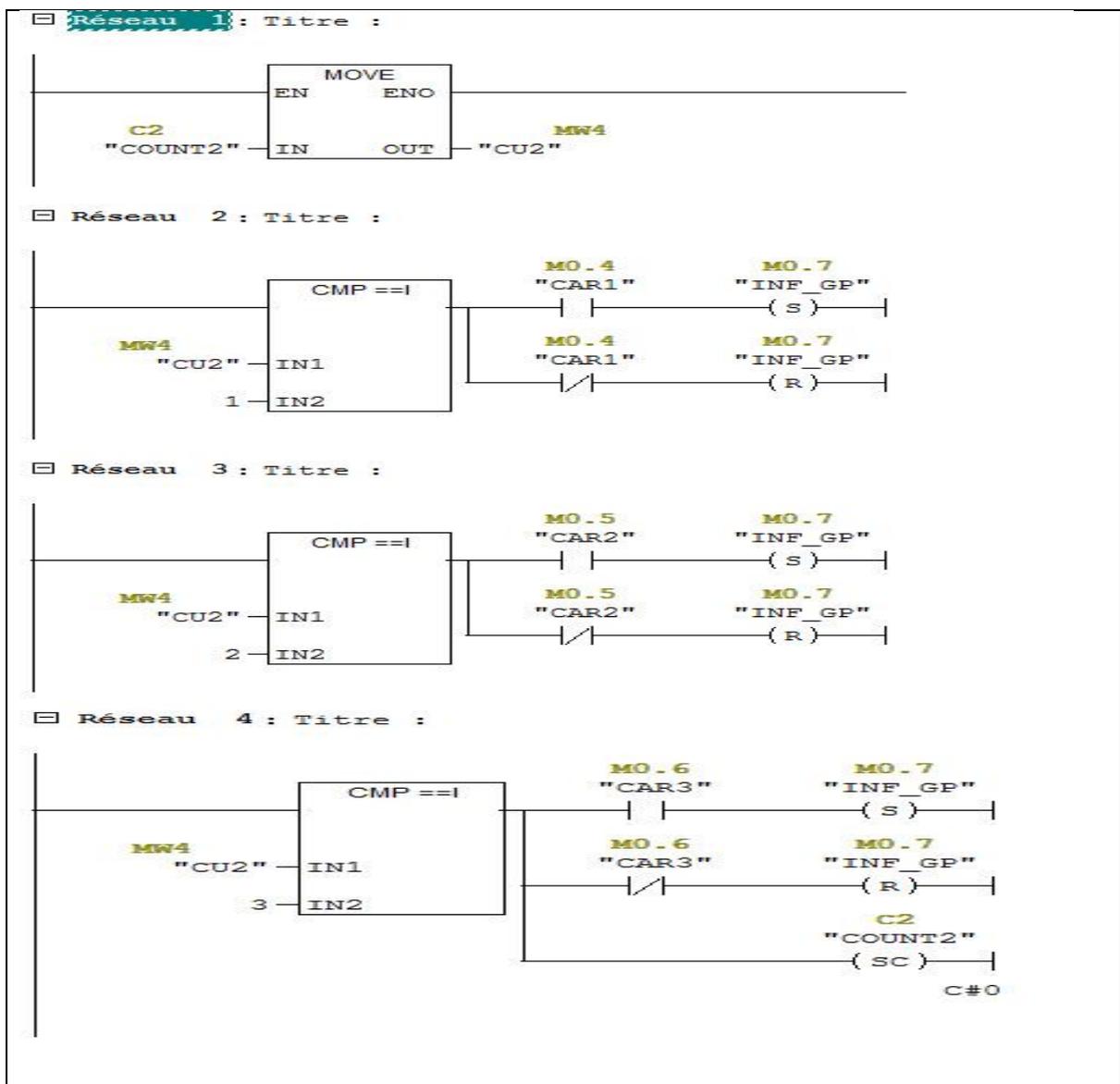


Figure 3.34: Contenu de bloc FC2.

## Bloc d'organisation OB1

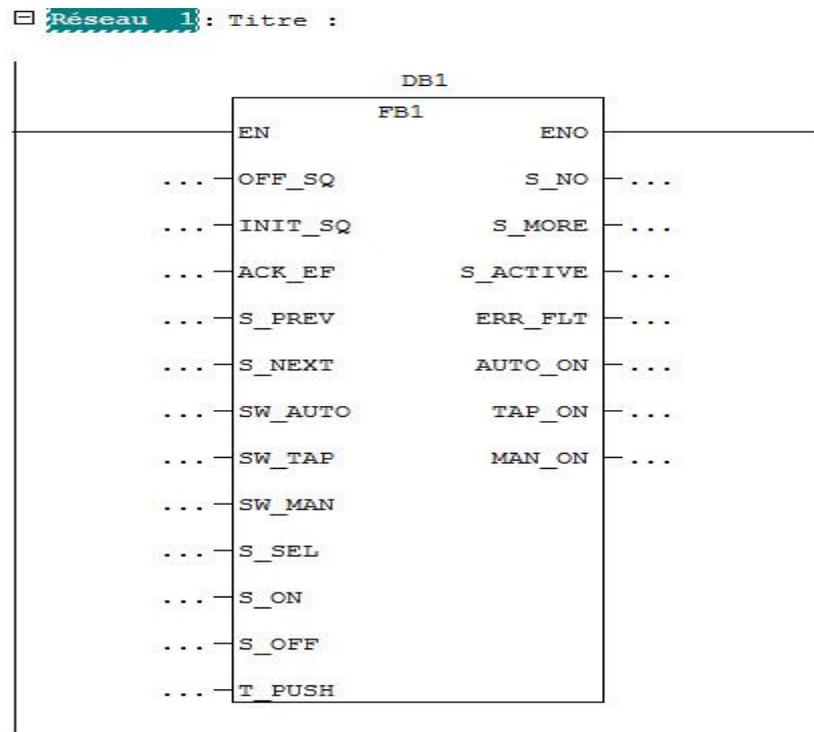


Figure 3.35: Bloc d'organisation OB1.

### 3.6.5 Simulation du programme par PLC-SIM

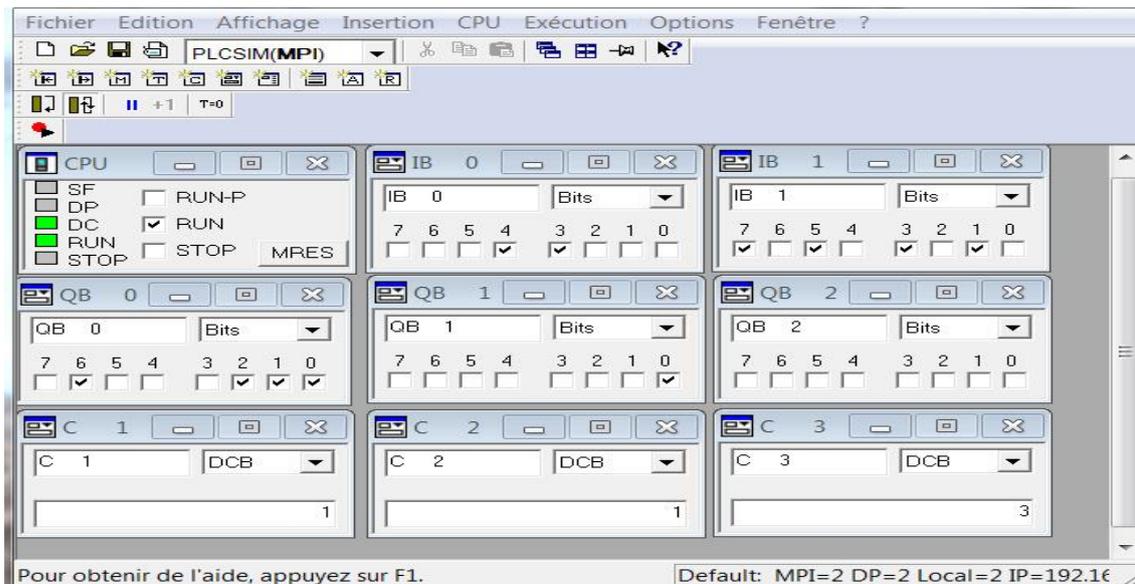


Figure 3.36: Simulation par PLC-SIM.

## 3.7 Présentation de la partie HMI

Pour bien contrôler le processus, l'opérateur a besoin d'avoir le maximum de transparence, ce qu'il lui permet de bien superviser et contrôler l'installation. Cela est possible avec l'interface homme machine (HMI). Le contrôle de processus est assuré par le système d'automatisation.

Dans notre projet la supervision permet :

- De commander et gérer le système à distance.
- De visualiser et Contrôler le fonctionnement souhaité de notre programme.
- D'afficher toutes les animations de système.

### 3.7.1 Variables du HMI

On distingue deux types de variables, les variables externes et les variables internes :

- Les variables externes permettent de communiquer et d'échanger des données entre les composants d'un processus automatisé, entre un pupitre opérateur et un automate.
- Les variables internes ne possèdent aucun lien avec l'automate, elles sont enregistrées dans la mémoire du pupitre.

#### d) Table de variables HMI

La figure 3.37 représente la table des variables HMI.

#### e) Configuration des éléments de la vue

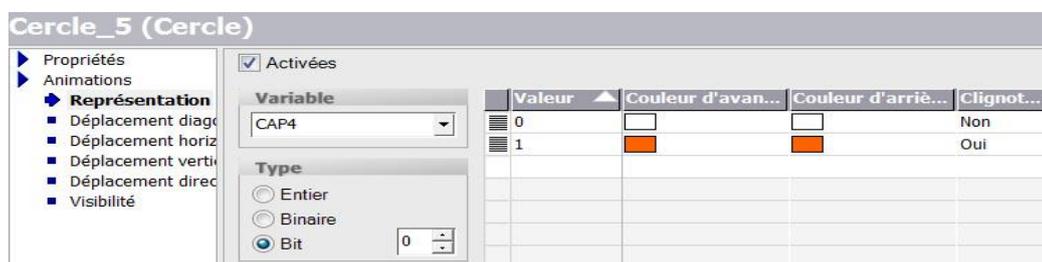
**Configuration des captures :** Chaque capture est associée à une entrée. Lorsque la variable de l'entrée est à « 0 » la capture est représentée en blanc (capture désactivée), lorsqu'elle est à « 1 » la capture est en orange (capture activée).

**Configuration des moteurs :** Chaque moteur est associé à une sortie. Lorsque la variable de sortie est à «0» le moteur est représenté en noire (moteur arrêt), lorsqu'elle est à «1» le moteur est en vert (moteur marche).

**Configuration des boutons :** Chaque bouton est animé par une couleur est associé à un memento. En appuyant sur le bouton on change l'état de ce memento dans l'automate soit à «1» ou à «0».

Nom	Nom d'affichage	Connexion	Type de données	Mnémonique	Adresse
CAP7		CPU312(1)	Bool	CAP7	I 0.7
ACT0_0		CPU312(1)	Bool	ACT0	Q 0.0
CAP3		CPU312(1)	Bool	CAP3	I 0.3
START_BP		CPU312(1)	Bool	START_BP	I 1.4
ACT4		CPU312(1)	Bool	ACT4	Q 0.4
START_HMI		CPU312(1)	Bool	START_HMI	M 1.0
CAP8		CPU312(1)	Bool	CAP8	I 1.0
URGENCE_BP		CPU312(1)	Bool	URGENCE_BP	I 1.7
ACT2		CPU312(1)	Bool	ACT2	Q 0.2
ACT6		CPU312(1)	Bool	ACT6	Q 0.6
ACT3		CPU312(1)	Bool	ACT3	Q 0.3
URGENCE_HMI		CPU312(1)	Bool	URGENCE_HMI	M 1.3
CAP5		CPU312(1)	Bool	CAP5	I 0.5
RESET_HMI		CPU312(1)	Bool	RESET_HMI	M 1.2
Variable_1		Liaison_1	Bool	<indéfini>	DB 1 DBX 0.0
CAP6		CPU312(1)	Bool	CAP6	I 0.6
STOP_BP		CPU312(1)	Bool	STOP_BP	I 1.5
RESET_BP		CPU312(1)	Bool	RESET_BP	I 1.6
ACT1		CPU312(1)	Bool	ACT1	Q 0.1
STOP_HMI		CPU312(1)	Bool	STOP_HMI	M 1.1
CAP4		CPU312(1)	Bool	CAP4	I 0.4
ACT0		CPU312(1)	Bool	ACT0	Q 0.0
CAP9		CPU312(1)	Bool	CAP9	I 1.1
CAP10		CPU312(1)	Bool	CAP10	I 1.2
CAP1		CPU312(1)	Bool	CAP1	I 0.1
ACT5		CPU312(1)	Bool	ACT5	Q 0.5
CAP2		CPU312(1)	Bool	CAP2	I 0.2

*Figure 3.37: la table de variables HMI.*



*Figure 3.38: Animation d'une capture.*

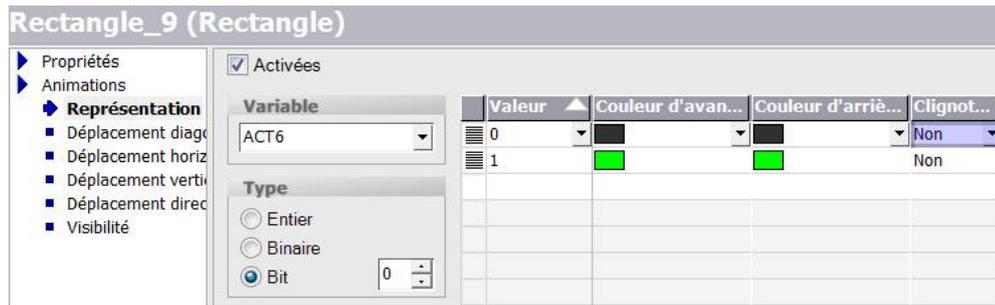


Figure 3.39: Animation d'un actionneur.

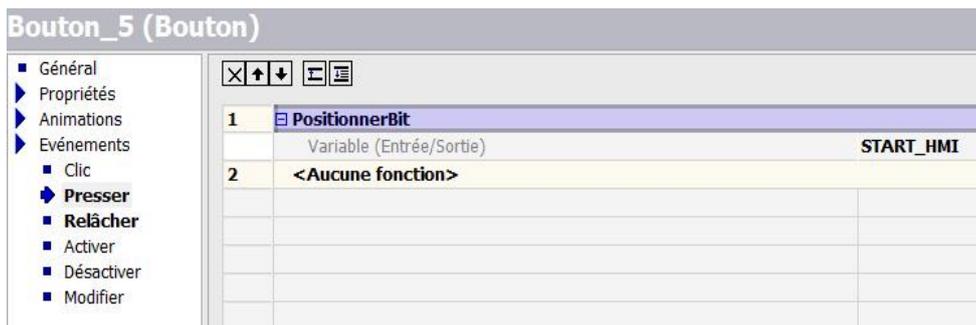


Figure 3.40 : Configuration d'un bouton

### 3.7.2 Vue principale

C'est une vue détaillée qui montre le fonctionnement du système de tri de caisses et permet à l'opérateur de commander le système (Marche/Arrêt/Arrêt d'urgence/Reset) par des boutons associés aux variables de l'automate.

Pour bien montrer le fonctionnement du système, on a effectué des animations pour présenter l'état des capteurs et les convoyeurs (active/d'active), le déplacement des caisses dans les convoyeurs et le tri des caisses.

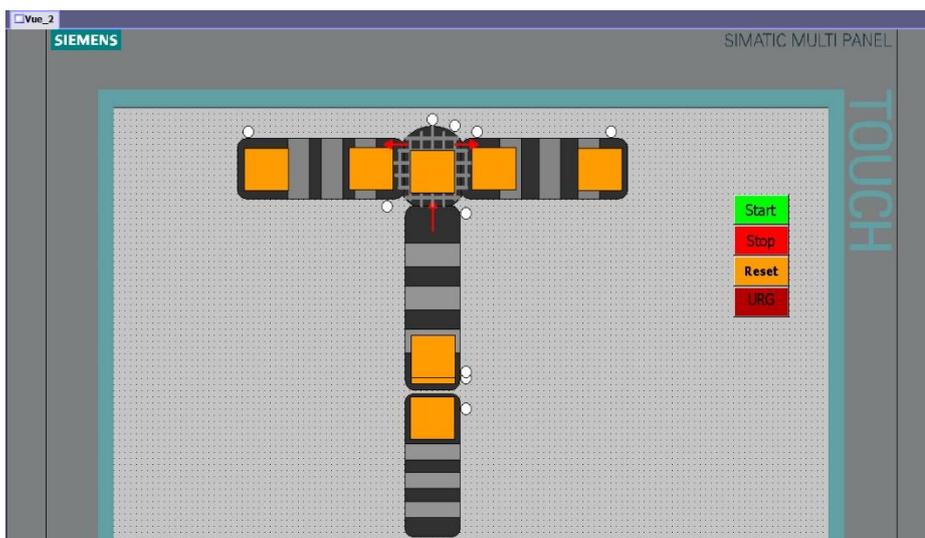


Figure 3.41: la vue de l'HMI de système de tri des caisses.

## 3.8 Simulation

### 3.8.1 Simulation du programme sous WinCC flexible et PLCSIM

Les premiers tests qu'on a effectués concernent l'exécution de notre programme HMI avec le simulateur PLCSIM.

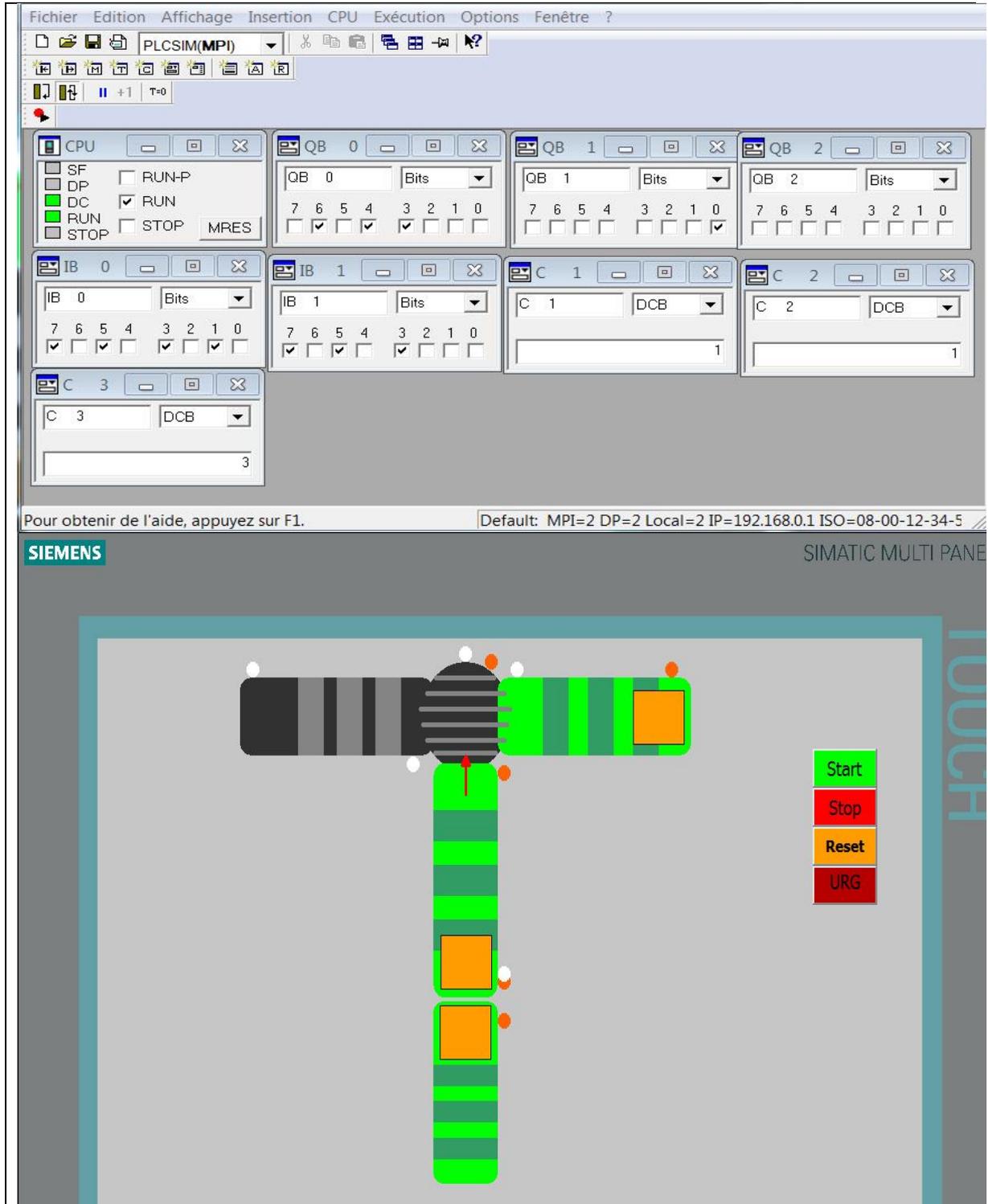
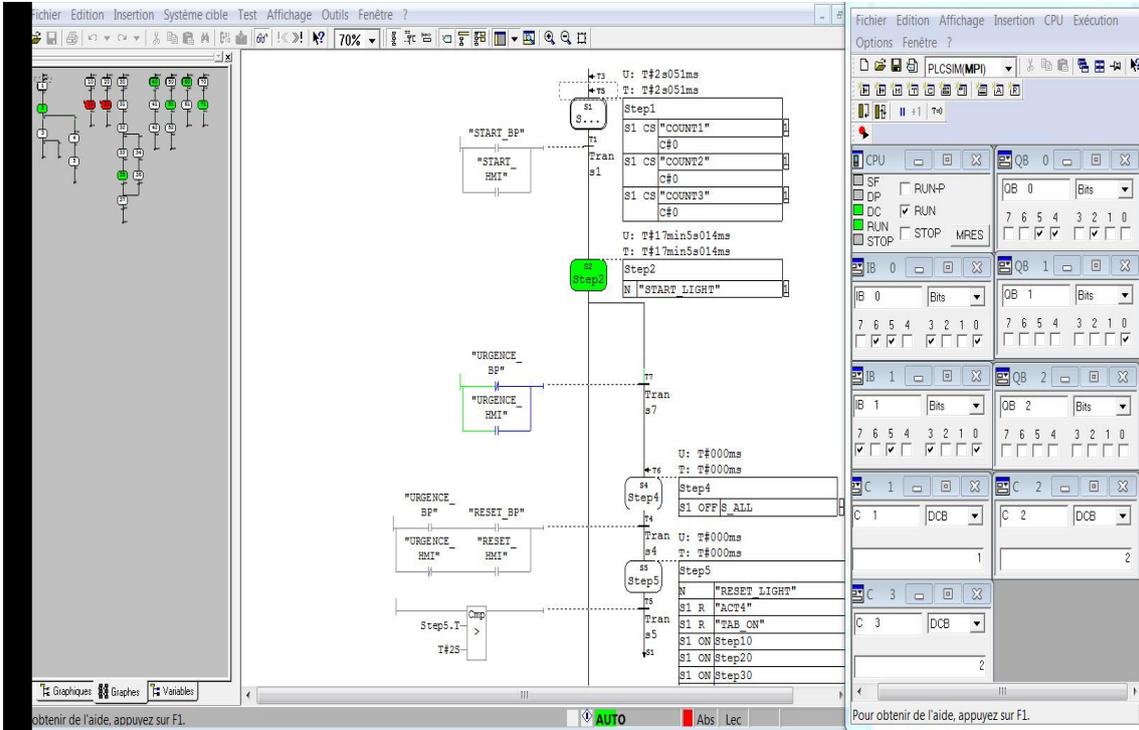
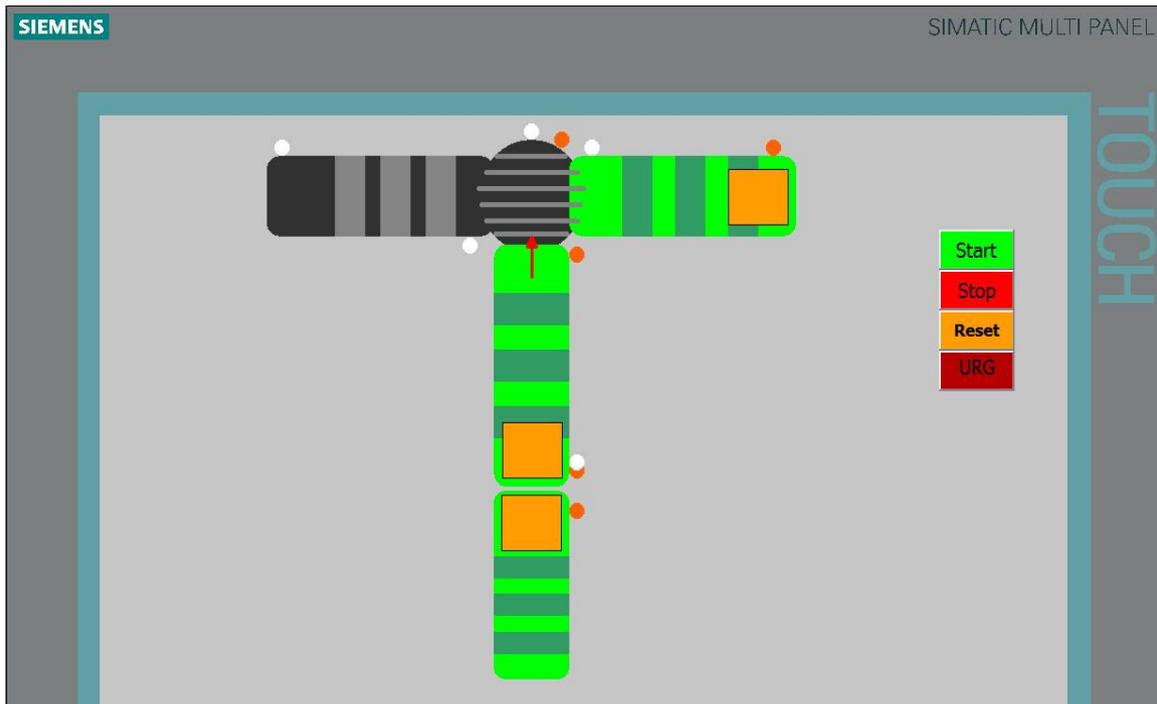


Figure 3.42: Simulation de l'HMI avec PLCSIM.

### 3.8.2 Simulation générale (avec les trois logiciels en parallèles)

Cette simulation permet d'exécuter et de tester la coordination entre les deux programmes (programme step7, interface homme-machine) qu'on a programmé sur l'ordinateur et de visionner le résultat en temps réel sur le système de tri de l'ITS PLC.





*Figure 3.43: Simulation générale.*

Les résultats de simulation que nous avons obtenus montrent bien que les programmes proposés respectent toutes les spécifications du cahier des charges.

### 3.9 Conclusion

Dans ce chapitre, nous avons développé un programme Grafcet complet sur le SIMATIC Manager S7 et un HMI à l'aide du logiciel WinCC flexible pour la commande et la supervision du système de tri virtuel de l'ITS PLC.

Les résultats de simulation obtenus confirment que les programmes développés respectent fidèlement le cahier de charge imposé.

# Chapitre 4: Automatisation et supervision d'un mélangeur de peinture

## 4.1 Introduction

Dans ce dernier chapitre, on présente les résultats d'automatisation et de supervision d'un mélangeur de peinture. Le but est d'obtenir la couleur souhaitée en mélangeant les proportions correctes des trois couleurs : le rouge, le vert et le bleu.

Les programmes et l'HMI développés sont validés en utilisant le modèle virtuel du mélangeur de peinture du logiciel ITS PLC.



*Figure 4.1 : Mélangeur de peinture virtuel de l'ITS PLC.*

## 4.2 Description du système

### 4.2.1 Structure du mélangeur

Le système mélangeur de l'ITS PLC se compose des éléments suivants :

- Trois réservoirs de peinture (A : rouge, B : vert, C : bleue).
- Trois vannes (D, E, F) pour remplir les cuves.
- Trois cuves (G, H, I) du dosage.
- Trois vannes (J, K, L) pour envoyer les peintures vers mélangeur.
- Un mélangeur de peinture (M).
- Tuyau de trop plein (O).
- Une vanne pour décharger la peinture (N).
- Des tuyaux de sorite (P).
- Un ensemble de capteurs et de voyants.

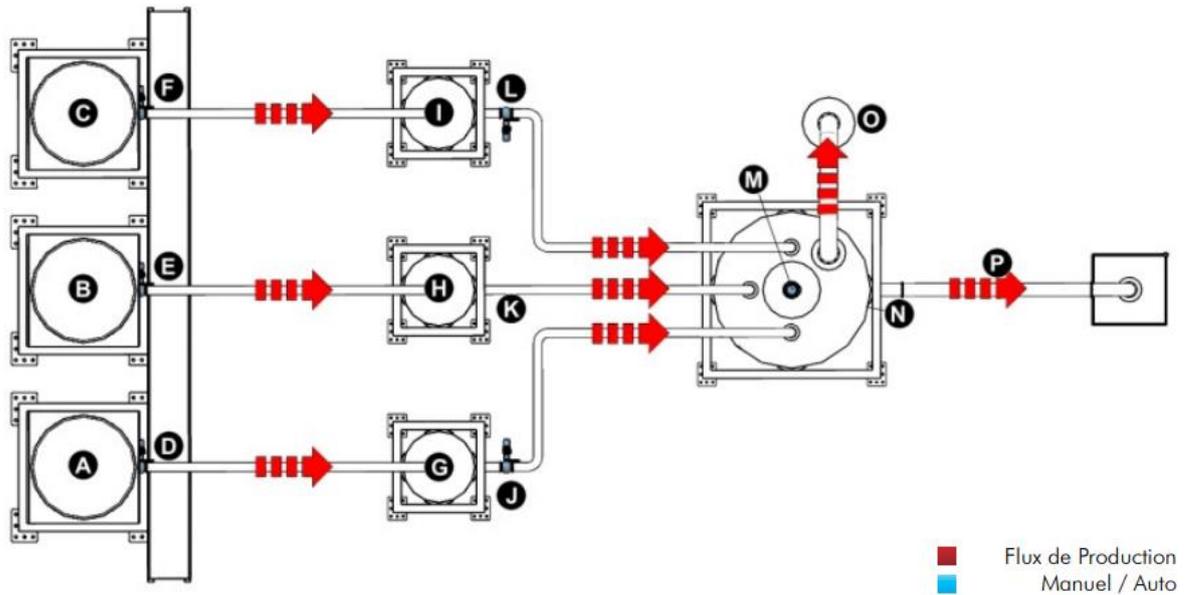


Figure 4.2 : Configuration du mélangeur de l'ITS PLC.

#### 4.2.2 Partie opérative

##### f) Actionneurs dont dispose le mélangeur

Le tableau suivant liste tous les actionneurs dont dispose le mélangeur avec leurs adresses.

Adresse	Actionneur	Description
Q0.0	Actionneur 0	Vanne de vidange du réservoir de peinture rouge
Q0.1	Actionneur 1	Vanne de vidange de la cuve de dosage de peinture rouge
Q0.2	Actionneur 2	Vanne de vidange du réservoir de peinture verte
Q0.3	Actionneur 3	Vanne de vidange de la cuve de dosage de peinture verte
Q0.4	Actionneur 4	Vanne de vidange du réservoir de peinture bleue
Q0.5	Actionneur 5	Vanne de vidange de la cuve de dosage de peinture bleue
Q0.6	Actionneur 6	Mélangeur
Q0.7	Actionneur 7	Vanne de vidange de la cuve du mélange
Q1.0		start_light
Q1.1		reset_light

Tableau 4.1 : Actionneurs et leurs adresses.

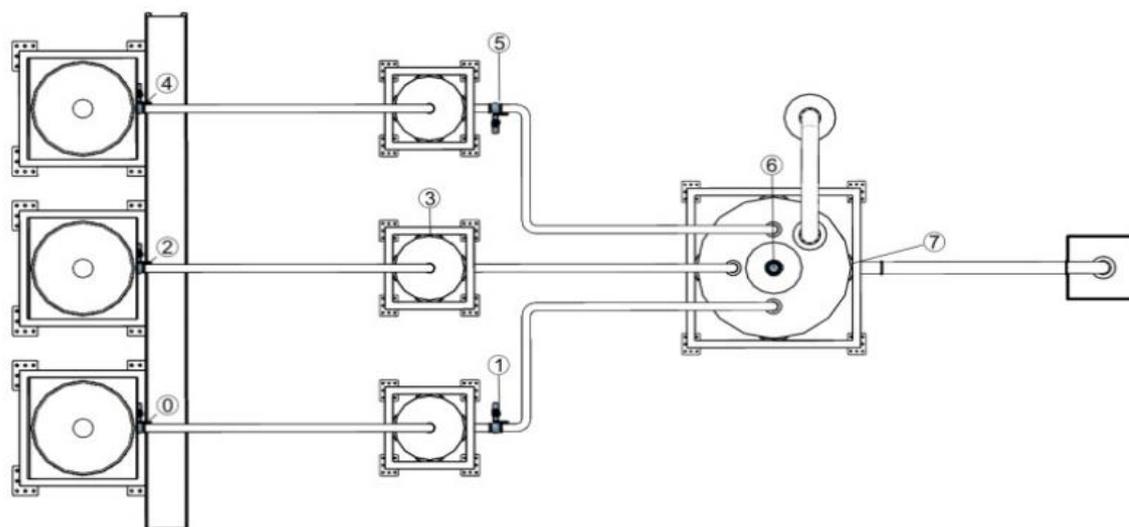


Figure 4.3 : Emplacement des actionneurs du mélangeur.

#### g) Capteurs et boutons dont dispose le mélangeur

Le tableau suivant recense les 15 capteurs et boutons que comporte le mélangeur ainsi que leurs adresses.

Adresse	Capteur	Description
I0.0	Capteur 0	Capteur niveau bas de la cuve de dosage de peinture rouge (cuve vide)
I0.1	Capteur 1	Capteur niveau milieu de la cuve de dosage de peinture rouge
I0.2	Capteur 2	Capteur niveau haut de la cuve de dosage de peinture rouge (cuve pleine)
I0.3	Capteur 3	Capteur niveau bas de la cuve de dosage de peinture verte (cuve vide)
I0.4	Capteur 4	Capteur niveau milieu de la cuve de dosage de peinture verte
I0.5	Capteur 5	Capteur niveau haut de la cuve de dosage de peinture verte (cuve pleine)
I0.6	Capteur 6	Capteur niveau bas de la cuve de dosage de peinture bleue (cuve vide)
I0.7	Capteur 7	Capteur niveau milieu de la cuve de dosage de peinture bleue
I1.0	Capteur 8	Capteur niveau haut de la cuve de dosage de peinture bleue (cuve pleine)

I1.1	Capteur 9	Capteur niveau bas de la cuve de mélange
I1.2	Capteur 10	Capteur niveau haut de la cuve de mélange
I1.3		Mode automatique
I1.4		Start
I1.5		Stop
I1.6		Reset
I1.7		Urgence

Tableau 4.2 : Capteurs et boutons et leurs adresses.



Figure 4.4 : Emplacement des capteurs disponibles.

### 4.3 Cahier des charges

Ce système de mélangeur est composé de trois réservoirs de peinture, trois cuves de dosage et une cuve pour faire le mélange. Les réservoirs de peinture (A, B et C) contiennent respectivement de la peinture rouge, verte et bleue. Les vannes (D, E, F) permettent de remplir les cuves de dosage (G, H, I) à partir des réservoirs. Chaque cuve de dosage a trois niveaux de mesure. La peinture des cuves de dosage est envoyée à la cuve de mélange (M) via les vannes (J, K, L). Si le volume de peinture ainsi envoyé est plus important que la capacité de la cuve de mélange, le surplus passe par le tuyau de trop plein (O). Le mélange doit durer un minimum de cinq secondes. La peinture obtenue est déchargée par la vanne (N) dans le tuyau de sortie (P).

La couleur de sortie obtenue résulte des niveaux de colorants dans les cuves de dosage. C'est-à-dire en fonction des indications des capteurs de niveaux 0...8 des trois cuves comme le montre la figure 4.5.

Après avoir choisi la couleur de sortie à partir de l'HMI, l'appui sur le bouton Start met le système en marche et l'appui sur Stop l'arrête. Un autre appui sur Start permet de continuer l'opération.

Capteurs de Niveau		
1	4	7
1	5	8
1	4	8
2	4	8
1		8
2	4	7
2		7
1		7
1		
2	5	7
1	4	
2	4	
1	5	7
	5	7
	4	
1	5	
	4	8
	4	7
		7

Figure 4.5 : Couleurs de sortie selon les indications des capteurs de niveaux des cuves de dosage.

En appuyant sur le bouton d'arrêt d'urgence le système s'arrête et ne peut se remettre en marche qu'après avoir appuyé sur Reset.

L'appui sur Reset permet de vidanger le mélangeur et les cuves de dosage à travers le tuyau du trop-plein. Cependant, dans le cas où le mélangeur est vide les cuve sont vidangées à travers le tuyau de sortie.

## 4.4 Programmation

### 4.4.1 Modélisation par Grafcet

Le modèle Grafcet composé par le graphe maitre et un graphe auxiliaire illustrés dans les figures 4.6 et 4.7 respectivement.

### 4.4.2 Interface de projet

Le programme créé se compose de :

- Bloc d'organisation OB1.
- Bloc de données DB1.
- Bloc fonctionnel FB1.

- Fonctions FC.

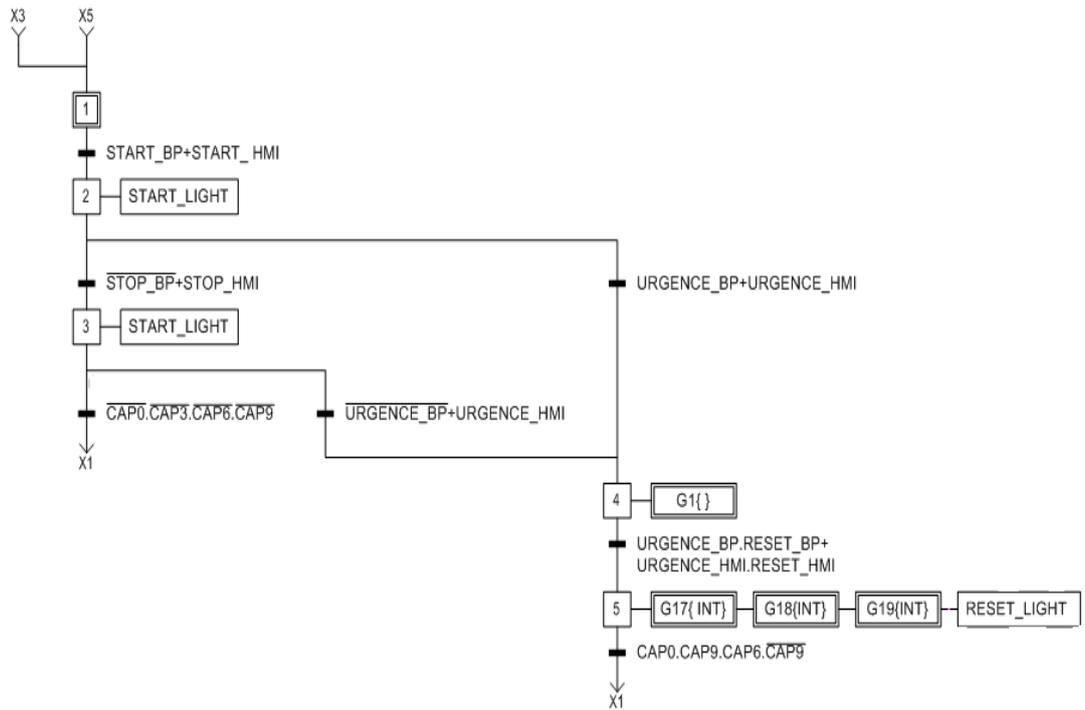


Figure 4.6 : Grafcet maitre.

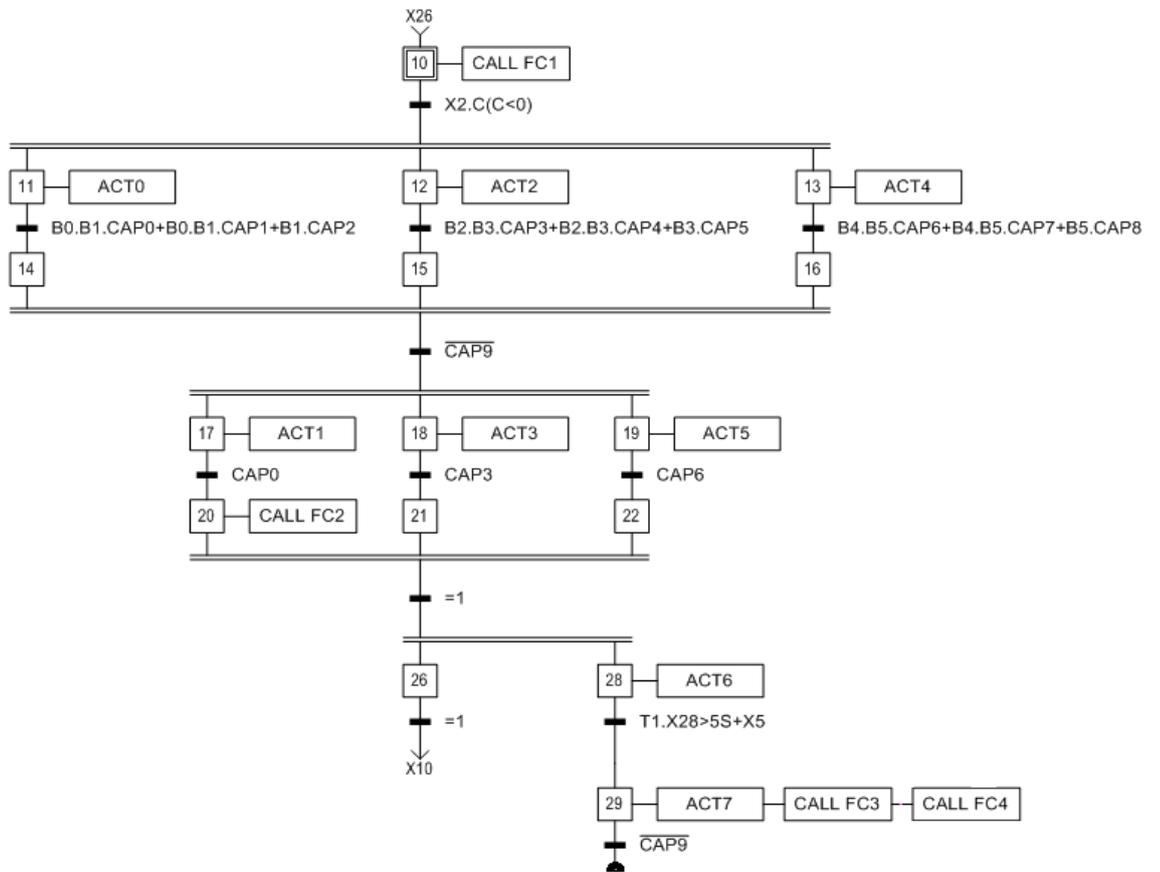


Figure 4.7 : Grafcet auxiliaire.

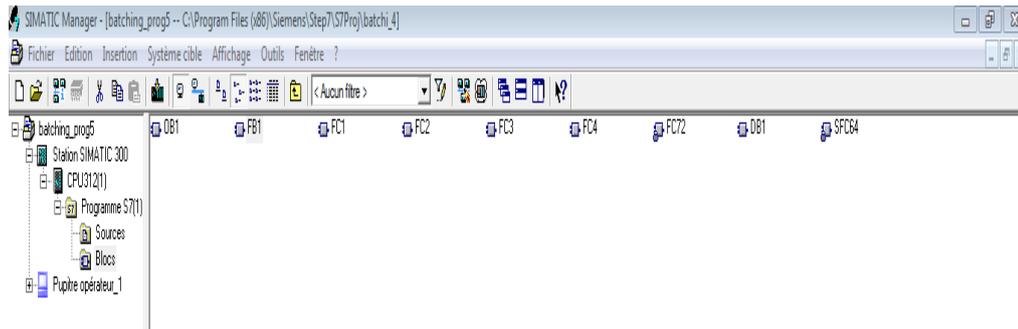


Figure 4.8 : Interface du projet.

### 4.4.3 Table des mnémoniques

La table mnémétique est représentée dans figure ci-dessous.

	Etat	Mnémoni	Opéran	Type d	Commentaire
1		ACT0	A ...	BOOL	Vanne de vidange du réservoir de peinture rouge
2		ACT1	A ...	BOOL	Vanne de vidange de la cuve de dosage de peinture rouge
3		ACT2	A ...	BOOL	Vanne de vidange du réservoir de peinture verte
4		ACT3	A ...	BOOL	Vanne de vidange de la cuve de dosage de peinture verte
5		ACT4	A ...	BOOL	Vanne de vidange du réservoir de peinture bleue
6		ACT5	A ...	BOOL	Vanne de vidange de la cuve de dosage de peinture bleue
7		ACT6	A ...	BOOL	Mélangeur
8		ACT7	A ...	BOOL	Vanne de vidange de la cuve de mélange
9		B0	M ...	BOOL	
10		B1	M ...	BOOL	
11		B2	M ...	BOOL	
12		B3	M ...	BOOL	
13		B4	M ...	BOOL	
14		B5	M ...	BOOL	
15		CAP0	E ...	BOOL	Capteur niveau bas de la cuve de dosage de peinture rouge (cuve vide)
16		CAP1	E ...	BOOL	Capteur niveau milieu de la cuve de dosage de peinture rouge
17		CAP10	E ...	BOOL	Capteur niveau haut de la cuve de mélange
18		CAP2	E ...	BOOL	Capteur niveau haut de la cuve de dosage de peinture rouge (cuve pleine)
19		CAP3	E ...	BOOL	Capteur niveau bas de la cuve de dosage de peinture verte (cuve vide)
20		CAP4	E ...	BOOL	Capteur niveau milieu de la cuve de dosage de peinture verte
21		CAP5	E ...	BOOL	Capteur niveau haut de la cuve de dosage de peinture verte (cuve pleine)
22		CAP6	E ...	BOOL	Capteur niveau bas de la cuve de dosage de peinture bleue (cuve vide)...
23		CAP7	E ...	BOOL	Capteur niveau milieu de la cuve de dosage de peinture bleue
24		CAP8	E ...	BOOL	Capteur niveau haut de la cuve de dosage de peinture bleue (cuve pleine)
25		CAP9	E ...	BOOL	Capteur niveau bas de la cuve de mélange
26		COULEUR1	MW ...	INT	
27		COULEUR2	MW ...	INT	
28		COULEUR3	MW ...	INT	
29		COULEUR4	MW ...	INT	
30		Cycle Execut...	OB ...	OB ...	
31		G7_STD_3	FC ...	FC ...	
32		MODE_AUTO	E ...	BOOL	mode auto
33		RESET_BP	E ...	BOOL	reset
34		RESET_HMI	M ...	BOOL	
35		RESET_LIGHT	A ...	BOOL	
36		START_BP	E ...	BOOL	Start
37		START_HMI	M ...	BOOL	
38		START_LIGHT	A ...	BOOL	
39		STOP_BP	E ...	BOOL	stop
40		STOP_HMI	M ...	BOOL	
41		TIME_TCK	SFC ...	SFC ...	Read the System Time
42		URGENCE_BP	E ...	BOOL	urgence
43		URGENCE...	M ...	BOOL	

Figure 4.9 : Table des mnémoniques.

#### 4.4.4 Contenu du bloc fonctionnel FB1

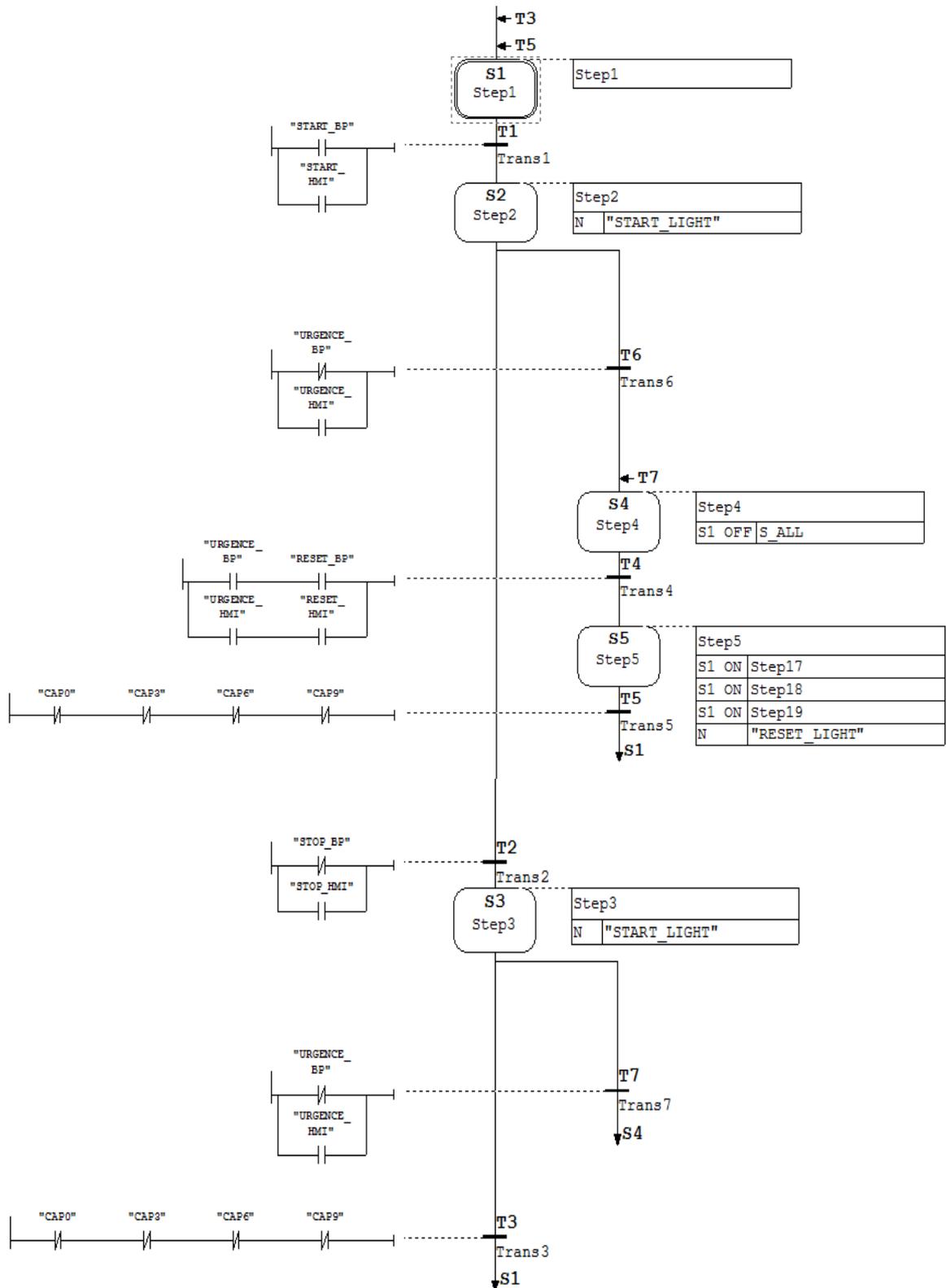
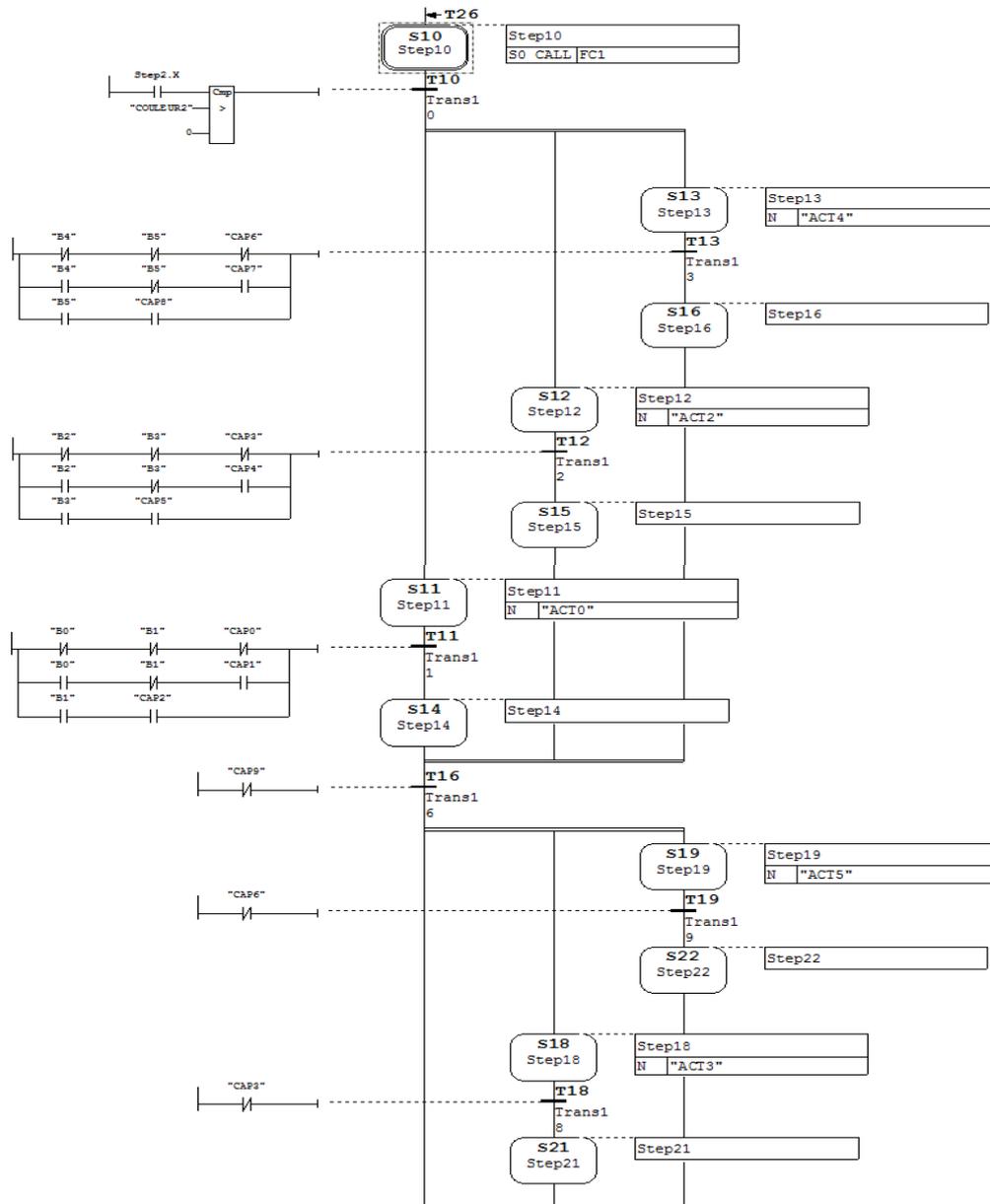


Figure 4.10 : Grafset maitre.



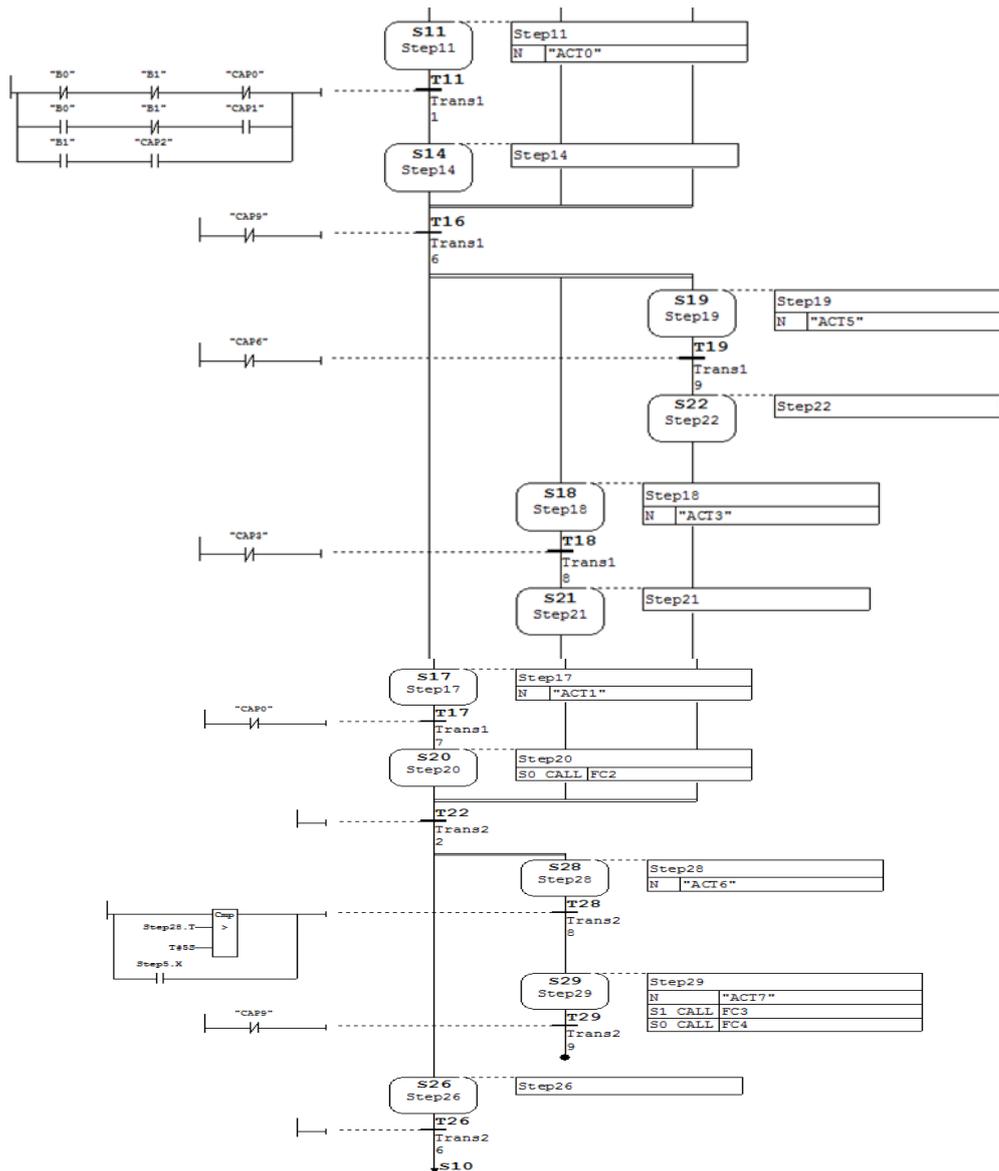


Figure 4.11 : Grafcet auxiliaire.

#### 4.4.5 Contenu des fonctions FC (FC1, FC2, FC3, FC4)

**FC1 (Figure4.12) :** Au début de chaque cycle de remplissage, le code de la couleur choisie (COULEUR2 est copié dans la variable COULEUR1).

**FC2 (Figure4.13) :** A la fin de la vidange des 3 cuves, on mémorise le code de la couleur à produire dans le mélangeur dans la variable COULEUR3.

**FC 3 (Figure4.14) :** lors de la vidange du mélangeur en copier le code de la couleur de la peinture produite COULEUR3 dans la variable COULEUR4, cette variable été utilisé pour afficher la couleur de la peinture produite en sortie du mélangeur dans la partie HMI.

**FC 4 (Figure4.15) :** elle permet de réinitialiser la couleur de la conduite en sortie du mélangeur après la fin de la vidange du mélangeur.

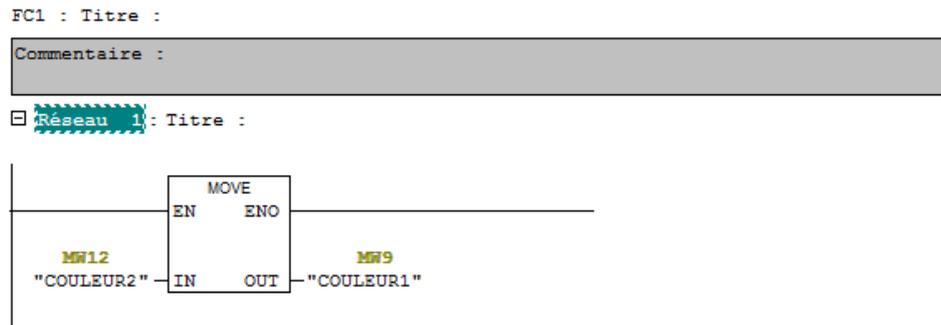


Figure 4.12 : FC1 choix de la peinture.

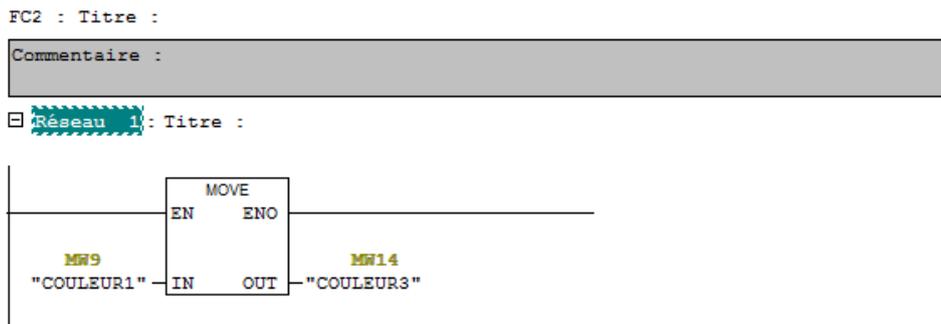


Figure 4.13 : FC2 enregistrement du choix de la peinture.

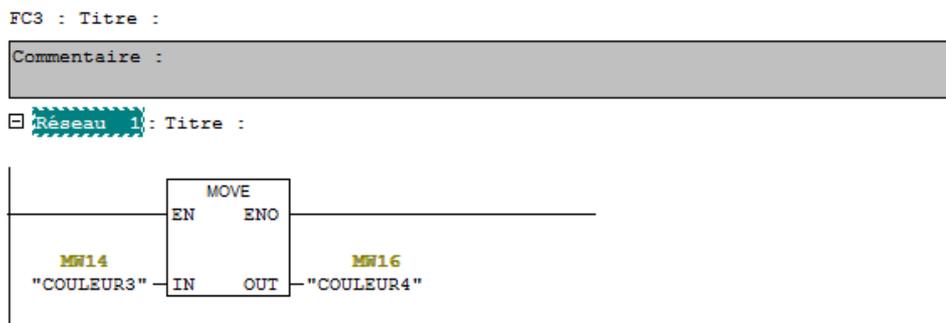


Figure 4.14 : FC3 affichage de la peinture produite à la sortie.

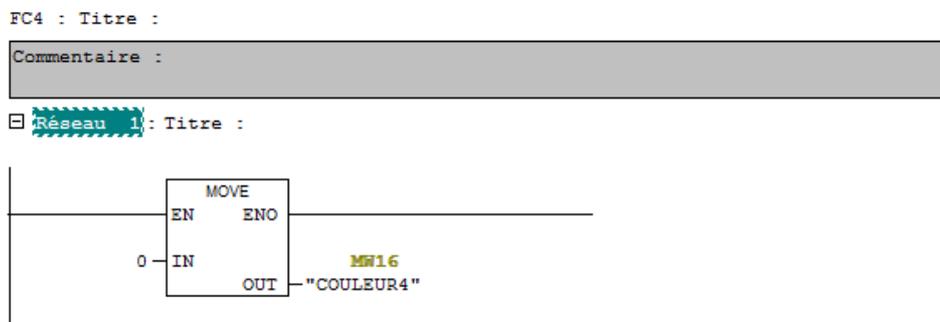


Figure 4.15 : FC4 effacer l'information de la peinture produite.

## 4.5 Elaboration de l'HMI

### 4.5.1 Table des variables

Nom	Nom d'affichage	Connexion	Type de données	Mnémonique	Adresse	Éléments du ta...	Cycle d'acq...
ACT4		CPU312(1)	Bool	ACT4	Q 0.4	1	100 ms
URGENCE_HMI		CPU312(1)	Bool	URGENCE_HMI	M 20.3	1	100 ms
CAP3		CPU312(1)	Bool	CAP3	I 0.3	1	100 ms
ACT6		CPU312(1)	Bool	ACT6	Q 0.6	1	100 ms
ACT2		CPU312(1)	Bool	ACT2	Q 0.2	1	100 ms
COULEUR2		CPU312(1)	Int	COULEUR2	MW 12	1	100 ms
CAP2		CPU312(1)	Bool	CAP2	I 0.2	1	100 ms
CAP1		CPU312(1)	Bool	CAP1	I 0.1	1	100 ms
CAP8		CPU312(1)	Bool	CAP8	I 1.0	1	100 ms
CAP5		CPU312(1)	Bool	CAP5	I 0.5	1	100 ms
CAP7		CPU312(1)	Bool	CAP7	I 0.7	1	100 ms
ACT3		CPU312(1)	Bool	ACT3	Q 0.3	1	100 ms
ACT7		CPU312(1)	Bool	ACT7	Q 0.7	1	100 ms
START_HMI		CPU312(1)	Bool	START_HMI	M 20.0	1	100 ms
START_BP		CPU312(1)	Bool	START_BP	I 1.4	1	100 ms
CAP4		CPU312(1)	Bool	CAP4	I 0.4	1	100 ms
ACT5		CPU312(1)	Bool	ACT5	Q 0.5	1	100 ms
STOP_HMI		CPU312(1)	Bool	STOP_HMI	M 20.1	1	100 ms
ACT0		CPU312(1)	Bool	ACT0	Q 0.0	1	100 ms
STOP_HMI_0		CPU312(1)	Bool	<indéfini>	M 20.2	1	100 ms
MODE_AUTO		CPU312(1)	Bool	MODE_AUTO	I 1.3	1	100 ms
RESET_HMI		CPU312(1)	Bool	RESET_HMI	M 20.2	1	100 ms
COULEUR4		CPU312(1)	Int	COULEUR4	MW 16	1	1 s
COULEUR3		CPU312(1)	Int	COULEUR3	MW 14	1	1 s

Figure 4.16 : Table des variables.

### 4.5.2 Vue Principale du système

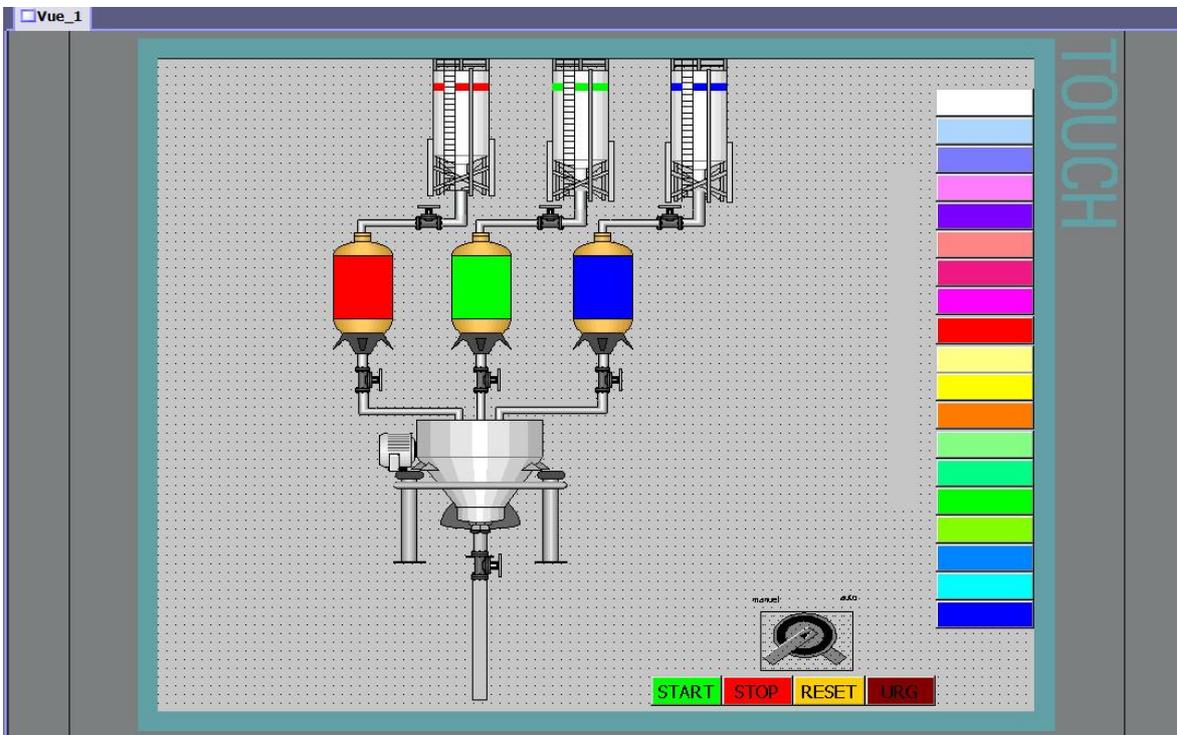


Figure 4.17 : Vue Principale.

## 4.6 Conclusion

Dans ce chapitre, après avoir décrit le mélangeur de peinture virtuel dont dispose le logiciel ITS PLC, on a présenté les étapes que l'on a suivi dans l'automatisation et la supervision de ce système. On a utilisé SIMATIC Manager S7 pour développer les programmes et WinCC flexible pour élaborer l'HMI.

La simulation des différents scénarios de fonctionnement a montré que les programmes développés suivent littéralement le cahier des charges.

## Conclusion générale

Le travail présenté dans ce mémoire concerne l'automatisation et la supervision de deux systèmes virtuels qu'offre le logiciel ITS PLC : un système de tri de caisses et un mélangeur de peinture. Tout d'abord, on a défini les cahiers des charges, puis on a modélisé le problème en utilisant le Grafcet. Dans l'étape suivante, on a utilisé Step 7 pour programmer de l'automatisme en utilisant un mélange de langages graphiques, et ensuite on a élaboré l'interface homme/machine pour la supervision du système en se servant du logiciel WinCC flexible.

La simulation des programmes développés a été effectuée en exécutant en parallèle les trois logiciels step7, WinCC flexible et ITS PLC.

La difficulté majeure qu'on a rencontrée dans l'automatisation du système de tri était comment savoir la taille du carton qui vient d'arriver sur le plateau tournant pour l'aiguiller vers la direction correcte. Pour résoudre ce problème, on a créé des sous-programmes (les fonctions FC1, FC2) pour la mémorisation et la lecture successivement. Pour le deuxième système l'ennui était comment programmer l'HMI pour qu'il affiche la même couleur à la sortie que celle du mélangeur.

Le logiciel ITS PLC avec sa réalité virtuelle 3D nous a été d'une très grande utilité. En effet, il permet d'appréhender rapidement le principe de fonctionnement du système à automatiser et par conséquent de définir un cahier des charges qui tient compte de tous les scénarios de fonctionnement possibles. De plus, il permet déceler facilement les déficiences dans les programmes développés et rend l'apprentissage des automatismes plus amusant que jamais.

On espère que ce modeste travail servira de base de départ pour notre vie professionnelle et puisse apporter aussi un plus ainsi être bénéfique aux promotions.

Enfin, pour des développements ultérieurs en perspective du travail effectué dans ce mémoire, on propose :

- Système de tri : Utilisation d'une file d'attente FIFO pour la mémorisation de la taille des caisses en entrée du convoyeur B pour savoir comment les aiguiller en sortie de la table tournante. On peut aussi envisager l'amélioration des performances du système en considérant un fonctionnement où le convoyeur B peut contenir jusqu'à 4 cartons en même temps.
- Système mélangeur : Utilisation du code RGB (Red, Green, Blue) des couleurs et des temporisations pour doser chacune des couleurs primaires et obtenir ainsi la couleur souhaitée.

---

# Bibliographie

- [1] S. Soumya et O. Kawtar, «Etude d'une boucle de régulation de niveau implémentation du régulateur et réglage du procédé,» Rapport de projet de fin d'étude, Ecole supérieure de technologie de Fés, 2011.
- [2] [En ligne]. Available: <https://sites.google.com/site/technologieprevert/techno-3eme>. [Accès le 09 05 2021].
- [3] M. Khelouat, «Automatisation et supervision d'une thermolaquage par un automate S7-1200,» Projet de fin d'étude ,Université m'hamed bougara, Boumerdes, 2007.
- [4] L. Bergougroux, «API automate programmable industriel,» Support de cours, Poly Tech Marseille, 2004.
- [5] A. Tensaout et Y. Khoudja, «Conception d'une régulation de niveau avec un automate programmable,» Projet de fin d'étude, Université de Bejaia, 2015.
- [6] «Automate programmable S7-1200,» Manuel Siemens, 2015.
- [7] «S7-1500, ET 200MP système d'automatisation,» Manuel siemens, 2019.
- [8] E. Maalem et I. Touadji, «Les langages de programmation de l'automate programmable industriel (application au pilotage d'un ascenseur),» Université Ahmed Draia, Adrar, 2017.
- [9] [En ligne]. Available: [https://fr.wikipedia.org/wiki/CEI\\_61131-3](https://fr.wikipedia.org/wiki/CEI_61131-3). [Accès le 06 06 2021].
- [10] B. Chahrazed, «Automatisation & Supervision d'une porte coulissante,» Projet de fin d'étude, Université de Annaba, 2018.
- [11] C. Jaunay. [En ligne]. Available: [http://christophe.jaunay.free.fr/IMG/etudes\\_techniques\\_bts2/tp\\_langage\\_contact](http://christophe.jaunay.free.fr/IMG/etudes_techniques_bts2/tp_langage_contact). [Accès le 01 07 2021].
- [12] «Programmer avec STEP 7,» Manuel, Industry Support Siemens, 2006.
- [13] A. Latifa, T. Nadjiba et M. Asia, «Etude et automatisation de système du,» Mémoire de Master, Université Mohamed Khider de Biskra, 2018.
- [14] C. Abdeljaoued, «Elaboration d'un système d'automatisme et de régulation d'une unité d'aérocondenseur de vapeur d'eau en remplacement d'un condenseur de vapeur à eau de mer,» Mémoire d'ingénieur , Université du 7 novembre à Carthage, 2015.

- [15] « l'ITS PLC user guide,» Real games Lda, 2008.
- [16] H. Bousri et M. Hammoud, «Automatisation et supervision du four de la SCS d'El-Milia par S7-300 et SIMATIC,» Projet de fin d'étude, Université de Jijel, 2020.