

République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieure et de la Recherche Scientifique



Université Abderrahmane Mira - Béjaïa

Faculté de Technologie



Département d'Automatique, Télécommunication et d'Electronique

Projet de Fin d'Etudes

Pour l'obtention du diplôme de Master

Filière : Automatique

Spécialité : Automatique et systèmes

Thème

Réalisation d'un système automatisé de remplissage et de
fermeture de bouteilles (Entreprise MORTERO)

Réalisé par :

Mr. BOURAI Ismaïl

Mlle. MELLAK Kenza

Dirigé par :

Dr. A. OUARET

Mr. S-E. HAKEM

Examiné par :

Dr. F. TAFININE

Pr. B. MENDIL

Promotion : 2022/2023

Nous tenons à remercier,

D'abord et avant tout, le bon dieu, qui nous a donné la force, le courage et la patience d'achever et mener à bien ce modeste travail.

Notre encadrant, **Dr OUARET.A**, qui tout au long de notre travail, a su se montrer encourageant et toujours disponible. C'est un honneur pour nous de vous avoir comme encadrant pour ce mémoire de fin de cycle.

Notre encadrant, **Mr HAKEM S-E**, de nous avoir pris sous votre aile malgré toute votre charge de travail, et ce depuis le tout début. Votre expérience, vos conseils et votre savoir nous ont énormément appris et aidé durant toute cette période de stage au sein de l'entreprise. Nous vous remercions pour tout ça.

L'IT de l'entreprise MORTERO, **Mr ABDELOUHAB.A**, pour tout le temps consacré afin de nous aider. À nous avoir résolu tous les problèmes informatiques dont nous avons fait face et à constamment demander des nouvelles de nos progrès. Nous vous en sommes profondément reconnaissants.

L'ensemble de l'équipe maintenance de l'entreprise MORTERO, particulièrement à **Mr OUSALEM.M**, d'avoir énormément contribué au travail, tendu main forte à chaque petit problème rencontré et surtout à avoir pris le temps de nous transmettre vos connaissances.

Vous avez fait de notre stage une expérience unique et inoubliable.

Mr DJENNADI & Mr MAHMOUD, pour la contribution que vous avez apporté sans aucune hésitation. Ce travail n'aurait pas pu être possible sans vous et vôtres précieuses aides. Nous vous remercions sincèrement.

L'ensemble des **membres du jury**, d'accepter à examiner ce travail et d'assister à notre soutenance. Nous sommes convaincus que vos retours nous seront d'une grande aide afin de pouvoir nous améliorer et remettre un travail de meilleure qualité.

L'ensemble du **corps professoral** du département ATE. Leurs enseignements tout au long de notre cursus, les efforts fournis de leur part et la passion transmise représentent la base de tout ce travail.

Je dédie ce travail,

Avant tout, et avec beaucoup de fierté, à mes très chers parents **ZAKIA & MADJID** pour l'amour qu'ils m'ont toujours apporté, la confiance qu'ils portent en moi, le soutien dont ils ont toujours fait preuve et l'éducation qu'ils m'ont transmise, qui sans doute sont les principales raisons qui font que j'en suis arrivé là où je suis aujourd'hui. Je ne vous remercierai jamais assez pour tout ce que vous avez fait pour moi durant ces 23 dernières années, à m'avoir toujours poussé de l'avant même lorsque je doutais, à m'avoir toujours exprimé votre fierté même dans mes échecs, à tous les sacrifices que vous avez faits et que vous continuez de faire pour moi en permanence. Aucun mot ne pourrait exprimer l'amour que je vous porte et la reconnaissance que j'ai de vous avoir comme parents. Je t'aime maman, je t'aime papa, et ce, inconditionnellement.

À ma grande sœur **Yasmine** qui m'a toujours soutenu, encouragé et motivé. Une source d'admiration et d'inspiration, qui n'a jamais reculé face aux obstacles. Ta générosité et ton grand cœur font de toi une personne unique, toujours prête à tout donner pour les gens que tu chéris. Je t'aime.

À mon petit frère **Mehdi** qui me rend tellement fier. A cet âge de l'adolescence, où les enfants sont méchants et blessants, toi tu as gardé ton grand cœur, empathique, tolérant qui se soucie de ce qu'autrui peut ressentir. Cette maturité fait de toi un enfant hors du commun. Je t'aime plus que l'infini (comme tu me disais étant petit).

À toutes mes deux familles maternelle et paternelle, en particulier mes tantes, qui ont toujours été comme des mères pour moi, et à ma grand-mère. Je me considère tellement chanceux de vous avoir dans ma vie, d'être proche de vous, de partager autant de souvenirs (et pleins d'autres en route). Merci d'être toujours là pour moi, de me conseiller et de me motiver. Je vous aime.

Petit remerciement spécial pour tata **Wahiba.B** qui s'est investie pour nous aider à régler certains problèmes rencontrés avec notre matériel.

À mes cousins, plus particulièrement à ceux qui devaient faire partie de la virée à tamelaht (qui sans aucune surprise, ne s'est jamais faite), merci d'avoir fait partie de mon enfance,

mais surtout de mon présent. **Celya** la geek-militante, **Assia** la déprimasse, **Dalia** et nos 21 mois d'écart et **Evan** le problématique, je vous apprécie fortement (en vrai je vous aime un peu... beaucoup).

En hommage à mes défunts **Grands-parents**, qu'ils reposent tous en paix. Papi **Smail.B**, je ne t'ai malheureusement pas connu, mais ta mémoire perdure toujours et me rend fier de porter ton nom. Mami **Zahra.D.B**, je n'oublierai jamais les étés où tu venais nous voir, ces souvenirs me rendent mélancoliques mais je suis conscient de la chance d'avoir passé une enfance à tes côtés, tu me manque. Je vous aime. Au meilleur papi du monde, papi **Hocine.B**, j'aurais aimé que tu sois parmi nous et que tu puisses assister à la concrétisation de ce travail. Que tu sois fier de moi, même si je sais que tu l'étais déjà. Chaque souvenir passé avec toi depuis mon enfance, jusqu'à ce que tu nous quitte cette année, me rend fier de toi. Tu laisses une trace que les années n'effaceront jamais. Je t'aime et je suis fier d'être ton petit fils.

À **Adib.A** d'avoir déjeuné avec nous durant nos pauses déj. Bon il a peut-être bien réglé un ou deux problèmes majeurs de notre travail. Merci de nous avoir toujours motivé à continuer et à croire en nous.

À mes meilleurs amis **Menad.M**, **Yassine.M**, **Lydia.I**, **Lisa.A**, **Yasmine.F**, **Kenza.M** et **Melissa.B** et à d'autres amies que je n'ai pas cité mais à qui je pense, de m'avoir tant soutenu et encouragé. Particulièrement **Menad** qui m'a toujours montré que j'en étais capable, qui m'a aidé dans certains aspects de mon travail et qui m'a énormément conseillé. Petite pensée à **Lydia** avec qui on a toujours partagé nos galères ensemble, même durant notre mémoire et à **Yassine** qui joue tellement mal au foot qu'il s'est encore blessé.

À la personne qui a partagé toutes les difficultés rencontrées et qui les a réglées avec moi. Merci à ma binôme **Kenza.M** pour le travail fourni, la confiance accordée et le temps consacré. Je te souhaite plein de réussite, tu le mérites et tu le vaux.

Pour finir, je dédie ce travail à moi-même. Le fait de m'être surpassé, démêlé et n'avoir rien lâché. Ce travail m'a permis de surpasser mes limites, mes capacités à gérer mon stress et à gagner en confiance.

Ugh/Ismaïl.B

Je dédie ce travail,

À **mes parents**, qui ont toujours cru en moi, m'ont soutenu et ont été ma source d'inspiration constante tout au long de mon parcours académique. Votre amour inconditionnel et votre encouragement ont fait de ce succès une réalité.

À mon cher **grand frère**, un exemple de réussite et de persévérance pour moi, qui ne cesses de me soutenir dans toutes les épreuves de la vie.

À **mon binôme**, et meilleur ami de longue date. Pour les encouragements, la motivation et la compréhension dont tu as fait preuve tout au long de ce travail. Sans toi à mes côtés, rien n'aurait pu être possible, malgré le fait que tu sois quelqu'un de stressé et de stressant (on t'aime comme tu es ugh).

À **mes amis**, qui ont été présents durant chaque étape. Votre soutien moral, vos encouragements et vos moments de détente ont été des facteurs clés pour maintenir un équilibre pendant ces années d'études.

À toutes les personnes qui ont participé à la recherche en partageant leurs connaissances, en répondants à mes questions et en offrant de leurs temps précieux. Votre contribution a été essentielle pour l'avancement de mes travaux et la qualité de ce mémoire.

Enfin, à moi-même pour avoir relevé le défi de ce mémoire avec persévérance, discipline et détermination. Cette réalisation est le fruit de mes efforts personnels et de ma volonté à réussir malgré les difficultés rencontrées en cours de route.

Kenza.M

Liste des abréviations

API	Automate Programmable Industriel
Co	Connector
CPU	Central Processing Unit
DC	Direct Courant
FBD	Function Bloc Diagram
IL	Instruction List
LBin	List Binary
LD	Ladder Diagram
Nm	Newton-mètre
RAM	Random-access memory
ROM	Read-only memory
SFC	Sequential Function Chart
SPA	Société Par Action
ST	Structured Text

Table des matières

Liste des abréviations	i
Liste des tableaux	ii
Liste des figures	iii
Introduction générale	1
Chapitre I : Présentation de l'entreprise et généralités sur les automates programmables	
I.1. Introduction	2
I.2. Présentation de l'entreprise	2
I.2.1. MORTERO	2
I.2.2. Historique	2
I.2.3. Gamme de produits	3
I.2.4. Contacts et localisation	4
I.2.5. Evolution	4
I.2.6. Organigramme de MORTERO SPA	4
I.2.7. Dimension de MORTERO	5
I.3. Automates programmable et module de programmation LOGO Logic	6
I.3.1. L'importance de l'automatisation dans l'industrie et ses avantages	6
I.3.2. Automates programmables industrielles	6
I.3.3. Structures des API	6
I.3.4. Types d'API	7
I.3.4.1. Types compact	7
I.3.4.2. Types modulaire	8
I.3.5. Programmation utilisée pour les API	9
I.3.5.1. Liste d'instructions (IL)	10
I.3.5.2. Texte structuré (ST)	10
I.3.5.3. Schéma à contacts (LD)	10
I.3.5.4. Langage séquentiel graphique (SFC)	11
I.3.5.5. Blocs fonctionnels (FBD)	11
I.3.6. Module de programmation LOGO Logic	12
I.3.7. Particularités du module LOGO Logic	12
I.3.8. Avantages du module LOGO!	13
I.3.9. Entrées/sorties de LOGO! 8 12/24 RCE	13
I.3.10. Câblage	14

I.3.11. Programmation	14
I.3.12. Programmation sur LOGO!.....	15
I.3.13. Blocs et numéros de blocs.....	16
I.3.14. Logiciel LOGO!	18
I.4. Conclusion	18

Chapitre II : Étude, description et automatisation de la chaîne industrielle

II.1. Introduction	19
II.2. Chaîne de remplissage et fermeture de bouteilles	19
II.2.1. Définition	19
II.2.2. Principe de fonctionnement de la chaîne industrielle	20
II.2.2.1. Principe de fonctionnement de remplissage de bouteilles	20
II.2.2.2. Principe de fonctionnement de fermeture de bouteilles	20
II.2.3. Description des éléments de la chaîne industrielle	20
II.2.3.1. Convoyeur	20
II.2.3.2. Capteurs	21
II.2.3.3. Actionneurs	22
II.2.3.4. Réducteur de vitesse	23
II.2.3.5. Distributeurs	24
II.2.3.6. Contacteurs	25
II.2.3.7. Disjoncteur	26
II.3. GRAFCET	26
II.4. Description des éléments graphiques du GRAFCET	27
II.4.1. Etapes.....	27
II.4.2. Actions.....	28
II.4.3. Transitions	28
II.4.4. Réceptivités.....	28
II.4.5. Liaisons orientées.....	28
II.5. Structures de base	29
II.5.1. Saut d'étape et reprise d'étape	29
II.5.2. Séquence unique	29
II.5.3. Séquence simultanée	30
II.5.4. Séquence exclusive	30
II.6. Niveau d'emploi du GRAFCET	31

II.6.1. GRAFCET de niveau 1 du point de vue de la partie opérative	31
II.6.2. GRAFCET de niveau 2 du point de vue de la partie commande	31
II.7. Cahier de charge	32
II.8. GRAFCET de niveau 1 de la chaîne industrielle	33
II.9. Conclusion	35

Chapitre III : Programmation et simulation

III.1. Introduction	36
III.2. LOGO! Soft confort V8.3	36
III.2.1. Programmation sur LOGO! Soft confort V8.3	36
III.2.2. Création d'un projet	38
III.2.2.1. Démarrage LOGO!	38
III.2.2.2. Programme de commande	40
III.3. Simulation	42
III.3.1. Configuration de l'état du système	43
III.3.2. Configuration du tapis	43
III.3.3. Etape de présence de bouteille au remplissage	45
III.3.4. Etape de présence de bouteille à la fermeture	47
III.4. Eventuels problèmes	48
III.5. Connexion entre PC et LOGO!	48
III.5.1. Vérification de la connexion	49
III.5.2. Transfert du programme	49
III.6. Branchement de supervision et résultats	50
III.6.1. Eléments de notre branchement de supervision	50
III.6.2. Développement d'un branchement de supervision	51
III.7. Conclusion	55
 Conclusion générale	 56
Bibliographie	57

Liste des tableaux

Tableau I.1 : Principales étapes d'évolution de MORTERO SPA	4
Tableau I.2 : Données actuelles de MORTERO SPA	5
Tableau II.1 : Représentation des caractéristiques techniques du capteur photoélectrique	21
Tableau II.2 : Représentation des caractéristiques techniques du moteur	22
Tableau II.3 : Représentations de certains distributeurs	25
Tableau III.1 : Illustration des icônes de commande de la simulation	42
Tableau III.2 : Illustration des codes et désignations des éléments du branchement	51

Liste des figures

Figure I.1 : Représentation de la gamme de produit actuellement commercialisée.....	3
Figure I.2 : Illustration des emplacements géographique des régions de vente.....	4
Figure I.3 : Représentation de l’organigramme de MORTERO.....	5
Figure I.4 : Illustration par schéma d’une structure interne d’un API	6
Figure I.5 : Illustration d’un API compact	8
Figure I.6 : Illustration d’un API modulaire	8
Figure I.7 : Illustration de montage sur rail de modules	9
Figure I.8 : Exemple de syntaxe de langage IL	10
Figure I.9 : Exemple de syntaxe de langage ST	10
Figure I.10 : Représentation d’un schéma à contact LD	11
Figure I.11 : Représentation de schéma SFC	11
Figure I.12 : Représentation d’un langage à relais FBD	11
Figure I.13 : Illustration en image d’un module de programmation LOGO Logic	12
Figure I.14 : Représentation des caractéristiques du modèle LOGO! 8 12/24 RCE	13
Figure I.15 : Illustration de câblage d’un module LOGO!	14
Figure I.16 : Représentation du menu principal sur module LOGO ! avec écran	15
Figure I.17 : Représentation des éléments principaux du logiciel LOGO!	15
Figure I.18 : Illustration de mode de programmation sur écran LOGO!	16
Figure I.19 : Représentation d’un bloc sur écran LOGO!	17
Figure I.20 : Représentation de liaison entre blocs	17
Figure I.21 : Représentation des connexions des blocs	17
Figure II.1 : Représentation d’une chaine industrielle de remplissage et de fermeture	19
Figure II.2 : Illustration d’un convoyeur industriel	20

Figure II.3 : Illustration d'un capteur multifonction	21
Figure II.4 : Illustration d'un moteur	22
Figure II.5 : Représentation d'un vérin simple effet à gauche et d'un vérin double effet à droite en chaque position	23
Figure II.6 : Illustration d'un réducteur industriel	24
Figure II.7 : Illustration d'un motoréducteur de la marque BONFIGLIOLO	24
Figure II.8 : Illustration d'un contacteur	25
Figure II.9 : Illustration d'un disjoncteur	26
Figure II.10 : Illustration de la structure d'un GRAFCET	27
Figure II.11 : Représentation des deux types d'étapes	27
Figure II.12 : Représentation d'une action exécutant un ordre A associé à l'étape i	28
Figure II.13 : Représentation des deux éléments de transition et de réceptivité	28
Figure II.14 : Illustration des cas de saut d'étape et reprise d'étape	29
Figure II.15 : Représentation d'une séquence unique	30
Figure II.16 : Représentation d'une séquence simultanée	30
Figure II.17 : Représentation d'une séquence exclusive	31
Figure II.18 : Illustration des niveaux d'emploi du GRAFCET	31
Figure II.19 : Représentation des étapes importante du système étudié et du cahier de charge	32
Figure II.20 : Représentation du GRACET du système étudié	34
Figure III.1 : Illustration d'une interface d'un programme avec un éditeur CONT	36
Figure III.2 : Illustration d'une interface d'un programme avec un éditeur LOG	37
Figure III.3 : Illustration d'une interface d'un programme avec un éditeur UDF	37
Figure III.4 : Représentation d'ajout d'un diagramme dans le mode diagramme	38
Figure III.5 : Illustration du paramétrage de choix de l'éditeur	39
Figure III.6 : Représentation de la sélection du type de matériel LOGO!	40
Figure III.7 : Représentation de la liste d'instructions dans la barre d'outils	41

Figure III.8 : Représentation du menu de configuration d'un bloc	41
Figure III.9 : Illustration de l'icône simulation dans l'éditeur de diagramme	42
Figure III.10 : Illustration de la barre d'outils simulation	42
Figure III.11 : Visualisation de l'étape de configuration de l'état du système	43
Figure III.12 : Visualisation de l'étape de configuration du Tapis	43
Figure III.13 : Représentation de la simulation du tapis en marche	44
Figure III.14 : Représentation de la simulation du tapis en arrêt	45
Figure III.15 : Représentation de l'étape de remplissage de bouteilles	46
Figure III.16 : Représentation de la simulation de remplissage	46
Figure III.17 : Représentation de l'étape de fermeture de bouteilles	47
Figure III.18 : Représentation de la simulation de fermeture	48
Figure III.19 : Représentation de l'interface de transfert PC > LOGO!	50
Figure III.20 : Position "off" du disjoncteur et arrêt total du système	52
Figure III.21 : Arrêt du moteur convoyeur avec activation de la lumière rouge	52
Figure III.22 : Marche du moteur convoyeur avec activation de la lumière verte	53
Figure III.23 : Système avant et pendant détection au niveau du "DIP rempli"	54
Figure III.24 : Système avant et pendant détection au niveau du "DIP bcn"	54

Introduction générale

Introduction générale

Dans l'industrie, l'automatisation est le procédé le plus prioritaire, et ce, dans tous les domaines et secteurs quels qu'ils soient.

L'automatisation est un processus d'exécution de tâches à l'aide de technologies et de logiciels qui visent à améliorer l'efficacité, la productivité et la précision tout en éliminant les tâches manuelles répétitives [1].

Ce présent mémoire a pour objet l'automatisation, suivi de la simulation d'un prototype d'une chaîne de remplissage et de fermeture de bouteilles, au sein de l'entreprise MORTERO, à l'aide du module industriel LOGO ! 8 12/24 RCE, qui est un API de type compact. Ce module présente des avantages aux vues de sa flexibilité et de sa fiabilité, répondant aux exigences actuelles en matière de commande et de communication.

Afin de réaliser ce système, nous avons dans un premier temps, rédigé un cahier de charge qui représente le fonctionnement détaillé de ce dernier, puis, présenté son GRAFCET, avant de passer à la réalisation d'un branchement qui joue le rôle d'une supervision.

Pour une meilleure présentation des travaux abordés, ce manuscrit est structuré de la manière suivante :

Le **premier chapitre** est dédié à la présentation de l'entreprise et des généralités sur les automates programmables d'une manière générale, puis d'une façon détaillée du module LOGO ! 8 12/24 RCE.

Le **deuxième chapitre** est consacré d'une part, à la description de la chaîne industrielle de remplissage et fermeture de bouteilles, ainsi qu'à l'étude détaillée de ses différents composants. D'autre part, à l'automatisation de la chaîne industrielle, ou nous présenterons de manière détaillée le GRAFCET et poursuivrons avec la réalisation de ce dernier pour notre système.

Le **troisième chapitre** traite quant à lui la partie programmation du système avec toute la partie commande et simulation. Une conclusion générale couronnera ce modeste travail.

Enfin, nous clôturons ce travail par une conclusion générale et les perspectives envisagées.

CHAPITRE I

Présentation de l'entreprise et
généralités sur les automates
programmables

I.1. Introduction

Dans ce chapitre, nous allons présenter dans un premier temps l'entreprise où nous avons effectué notre stage tout au long de cette période, en présentant certains de leurs produits et quelques chiffres représentatifs de cette dernière. Ensuite, nous allons décrire les automates et donner leurs généralités, puis exposer le module LOGO! Logic de manière générale pour enfin s'approfondir sur le modèle choisit pour notre étude avec son logiciel de programmation.

I.2. Présentation de l'entreprise

I.2.1. MORTERO

De nous jours, notre gouvernement a fortement investi dans le secteur de la construction afin de faire croitre l'économie. Il a misé sur la construction de centres commerciaux et de centres culturels, tout en s'attaquant aux problèmes de logement grâce à des projets de logements sociaux et promotionnels. Cette politique a eu un impact significatif sur l'industrie des matériaux de construction, entraînant une forte augmentation de la demande.

MORTERO est spécialisé dans le domaine des mortiers industriels et s'est concentrée sur la recherche et le développement de gammes adaptées au marché local. Elle élabore, développe et produit des matériaux de qualité pour les façades, carrelages et revêtements de sols qu'elle revend. Elle met son expertise au service du secteur de la construction.

I.2.2. Historique

Fondée en 2009 par **MADAOUI** Abdelaziz, MORTERO est lancée en 2012 avec une gamme de mortiers colles sous PAREX, un acteur de classe mondiale dans le domaine des mortiers industriels avec à son actif 35 ans d'expérience. Présente dans 23 pays et comptant 74 sites industriels et deux centres de recherche et développement à l'international, MORTERO a fait de cette association un tremplin en Algérie.

Axée principalement sur la recherche et le développement de gammes de mortiers adaptées pour l'Algérie sur la base d'intrants locaux, MORTERO a ainsi atteint 82% de taux d'intégration nationale avec plus de 18 références de produits variés. Depuis 2012, plus de 9 000 000 m² d'enduits et 8 000 000 m² de colles ont été appliqués.

La croissance importante de MORTERO lui a permis de gagner sa place au premier rang des marques commercialisées en Algérie, lui permettant ainsi d'intégrer PAREX, leader mondial des mortiers industriels, devenu actionnaire en 2017.

PAREX, racheté depuis 2019 par SIKA, apporte à MORTERO son expertise dans les gammes de produits liquides. Plus de 100 ans d'expérience et une présence dans 76 pays, elle compte 120 sites de productions et de distributions et 13 500 collaborateurs. Une association qui, encore une nouvelle fois, perpétue le développement de MORTERO.

Actuellement, MORTERO compte plus de 70 collaborateurs directs, une implémentation sur plus de la moitié des départements du pays et plus de 500 points de vente.

L'année 2020 marque une nouvelle aire chez MORTERO avec l'exportation, débutant les opérations en février vers la Tunisie et effectuée à travers la plateforme SIKA.

L'année 2022 signe l'intégration du groupe SIKA, ou du rattachement organique, par le rachat de cette dernière des actions restantes [3].

I.2.3. Gamme de produits

MORTERO réparties sa large gamme de produits en trois catégories :

- Colles et joints.
- Façades.
- Mortiers spéciaux.

Les modèles actuellement commercialisés par l'entreprise sont présentés sur la Figure I.1.



Figure I.1 : Représentation de la gamme de produit actuellement commercialisée.

I.2.4. Localisation

Les cinq régions de vente sont présentées dans la figure I.2.

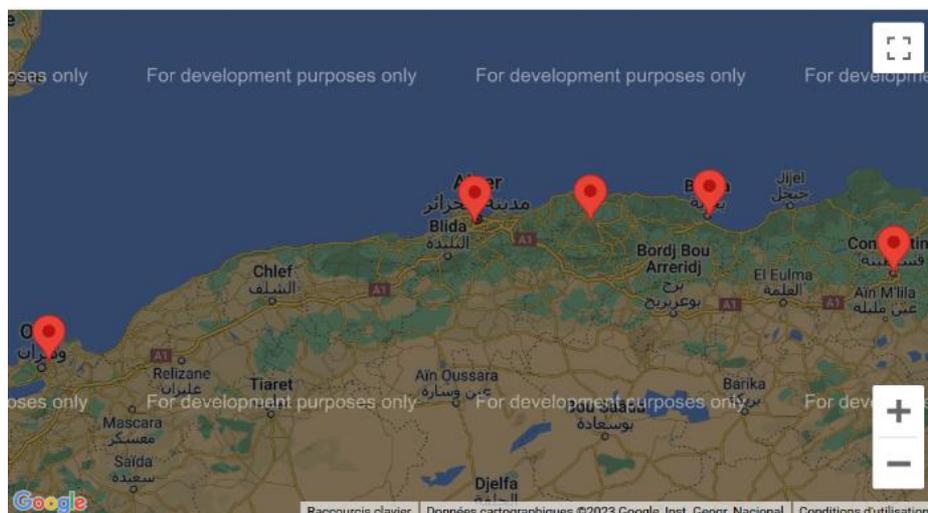


Figure I.2 : Illustration des emplacements géographique des régions de vente.

I.2.5. Evolution

Au cours des dernières années, MORTERO a connu une expansion continue et remarquable. L'évolution de cette entreprise peut être observée en consultant le Tableau I.1 :

Tableau I.1 : Principales étapes d'évolution de MORTERO SPA.

Années	Effectifs	Capacité de production
2008-2009	20	Essaie de lancement
2010-2011	50	50 tonnes/jour
2013-2014	60	100 tonnes/jour
2015-2016 (PAREX partenariat)	70	150 tonnes/jour
2020 -2022(intégration SIKA Groupe)	86	300 tonnes/jour

I.2.6. Organigramme de MORTERO SPA

La Figure I.3 représente un organigramme qui résume la structure interne complète de MORTERO.

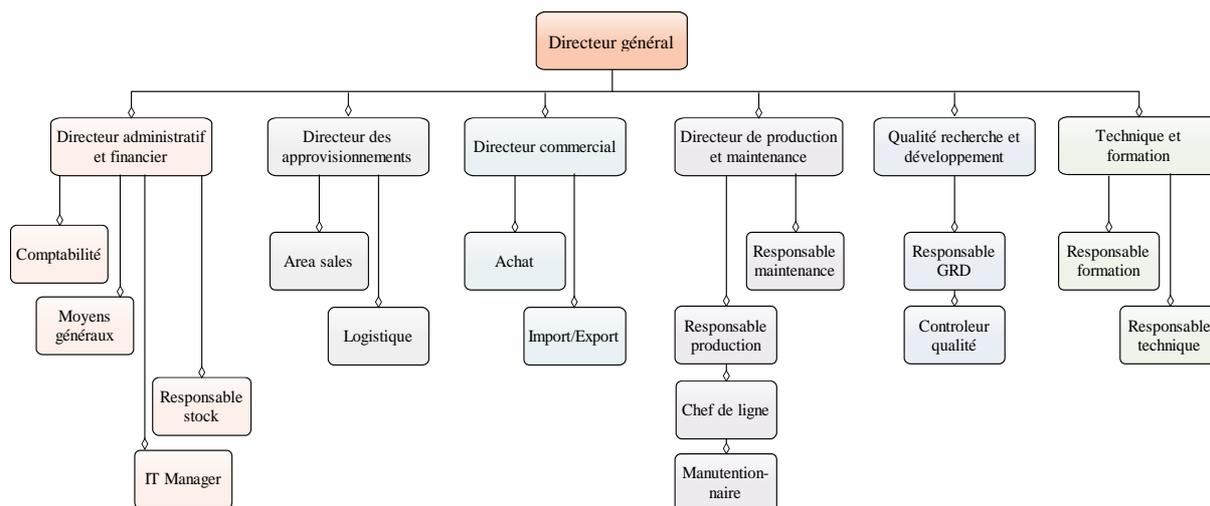


Figure I.3 : Représentation de l’organigramme de MORTERO.

I.2.7. Dimension de MORTERO

Les données de croissance de MORTERO sont présentées dans le Tableau I.2.

Tableau I.2 : Données actuelles de MORTERO SPA.

	Aujourd’hui	Avec extension
Emplois directe	60 emplois	120 emplois
Emplois indirecte	120 emplois	250 emplois
Salaires	40.000.000,00 DZD	80.000.000,00 DZD
Formation	1.000.000,00 DZD	3.000.000,00 DZD
Transfert technique français	3 Formations/An	Centre de formation local
Transfert technique et industriel	50% des installations faites localement	2 Nouvelles installations produites liquide local
Transfert recherche et développement	1 Laboratoire, 15 Formulations, mise à niveau en France	30 Formulations, recherche intégrée, accompagnement universitaire
Agents de maîtrise et cadres formes en collaboration avec PAREXGROUP	40 Personnes	100 Personnes + 20 stagiaires par an

I.3. Automates programmable et module de programmation LOGO Logic

I.3.1. L'importance de l'automatisation dans l'industrie et ses avantages

De nos jours, l'automatisation joue un rôle essentiel dans l'industrie. Elle permet de suivre l'évolution de la production et facilite l'exécution rapide et précise de certaines tâches pénibles, répétitives et dangereuses. Un système est considéré comme automatisé s'il est capable de fonctionner de manière autonome, exécutant des opérations préétablies de manière répétitive sans intervention humaine, sous réserve que les conditions requises soient remplies [4].

I.3.2. Automates programmables industrielles

Un Automate Programmable Industriel (API) est une machine électronique programmable conçue pour contrôler des processus industriels en temps réel dans un environnement industriel. Ils sont adaptés à un vaste nombre d'applications d'un point de vue traitement, composants et langage, d'où leur structure modulaire en général [2].

I.3.3. Structures des API

La structure interne d'un API est composée de plusieurs éléments reliés entre eux par des bus (câbles faisant passer l'information) comme illustré dans la Figure I.4 [4] :

- L'unité centrale de traitement (en anglais : Central Processing Unit « CPU »).
- La mémoire.
- Interface d'entrées/sorties.
- Modules d'entrées/sorties.
- Alimentation.

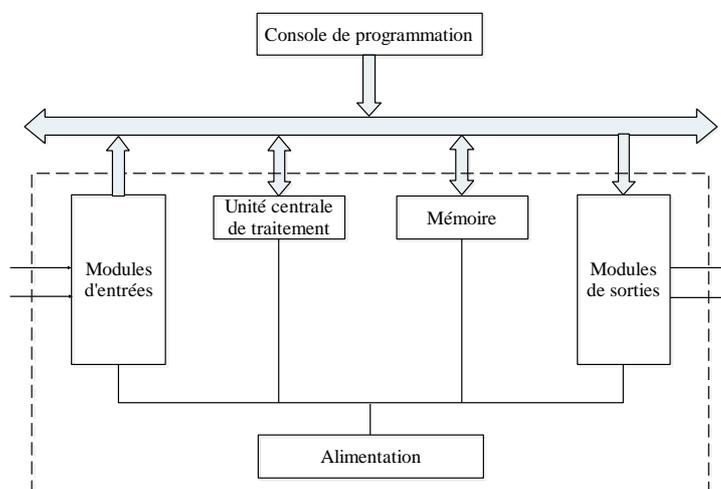


Figure I.4 : Illustration par schéma d'une structure interne d'un API.

Chaque élément de la structure de l'API remplit un rôle spécifique, qui peut être décrit comme suit [5] :

1. Unité centrale de traitement : elle consiste d'une part à organiser les différentes relations entre la mémoire et les interfaces d'entrées/sorties et d'autre part à traiter et gérer les instructions du programme.
2. La mémoire : Elle est conçue pour recevoir, gérer et stocker les informations de toutes les parties du système. À savoir le terminal de programmation (PC ou console) et le processeur qui gère et exécute le programme. Elle reçoit également des informations de capteurs.

Il existe deux types de mémoires qui remplissent des fonctions différentes :

- La mémoire où est stocké le langage de programmation (en anglais : Read-Only Memory « ROM »).
 - La mémoire utilisable en lecture-écriture pendant le fonctionnement, elle s'efface automatiquement à l'arrêt de l'automate (mémoire vive, en anglais : Random-Access Memory « RAM »).
3. L'alimentation : Les automates actuels sont tous équipés d'une alimentation 240 V 50/60 Hz, 24 V DC. Les entrées sont en 24 V DC et une mise à la terre doit également être prévue.
 4. Les interfaces et les cartes d'entrées/sorties : Les interfaces d'entrées et de sorties comportent comme suit des adresses d'entrées et de sorties. Chaque capteur est relié à une adresse d'entrée et chaque pré-actionneur est relié à une adresse de sortie. Pour chaque type d'automate, on peut trouver des nombres différents d'entrées et de sorties.

Les cartes d'entrées et de sorties ont une modularité de 8, 16 ou 32 voies qui délivrent une tension de 24V.

I.3.4. Types d'API

Il existe divers modèles et types d'API qui effectuent les mêmes tâches, avec différentes particularités, nous pouvons citer deux types de ce dernier [6] :

I.3.4.1. Type compact

Il intègre le processeur, l'alimentation, les entrées et les sorties. Selon les modèles et les fabricants, certaines fonctions supplémentaires peuvent être effectuées et des extensions en

nombre limité ajoutées. Ils sont généralement destinés à la commande de petits automatismes (voir Figure I.5).



Figure I.5 : Illustration d'un API compact.

I.3.4.2. Type modulaire

Le processeur, l'alimentation et les interfaces d'entrées/sorties résident dans des unités séparées (modules) et fixées sur des racks contenant le "fond de panier". Ces automates sont intégrés dans les automatismes complexes où puissance, capacité de traitement et flexibilité sont nécessaires (voir Figure I.6).

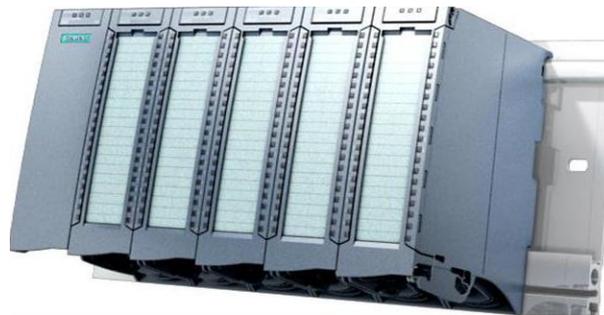


Figure I.6 : Illustration d'un API modulaire [12].

Le montage sur rail des modules basique et numérique se fait en appliquant ces étapes, représentées dans la Figure I.7 :

LOGO! Basic :

1. Accrocher le LOGO! Basic sur le rail DIN.
2. Faire pivoter le LOGO! Basic sur le rail DIN. Le coulisseau de montage se trouvant sur la face arrière doit s'enclencher.

Module numérique LOGO! :

3. Retirer le cache de la fiche de connexion sur le côté droit du LOGO! Basic/du module d'extension LOGO!.
4. Accrocher le module numérique à droite du LOGO! Basic sur le rail DIN.
5. Déplacer le module numérique vers la gauche jusqu'au LOGO! Basic.
6. A l'aide d'un tournevis, appuyer sur le coulisseau intégré et le déplacer vers la gauche. En position finale, le coulisseau s'enclenche dans le LOGO! Basic [7].

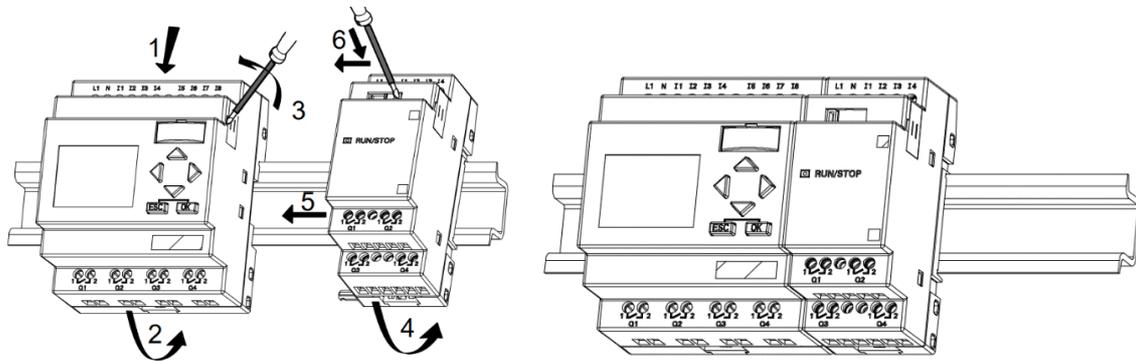


Figure I.7 : Illustration de montage sur rail de modules.

I.3.5. Programmation utilisée pour les API

Le concept de base de la programmation d'un API consiste à traduire les équations de fonctionnement du système à automatiser dans un langage spécifique. Chaque automate possède son propre langage. Les constructeurs conçoivent des interfaces logicielles selon la norme CEI 11313 (généralement des environnements logiciels graphiques) [6], pour cela nous pouvons citer les différents langages de cette norme [5] :

I.3.5.1. Liste d'instructions (IL)

C'est un langage bas niveau à une instruction par ligne. Il ressemble au langage assembleur employé pour la programmation des microprocesseurs.

Instruction list language	
LD	%I1.1
R	%C8
LD	%I1.2
AND	%M0
CU	%C8
LD	%C8.D
ST	%Q2.0

Figure I.8 : Exemple de syntaxe de langage IL.

I.3.5.2. Texte structuré (ST)

C'est un langage évolué. Il permet la programmation de tout type d'algorithme plus ou moins complexe.

```
IF #start = 1 THEN
  //comment
  "Max_nr" := #Array[0];
  FOR #i := 1 TO 10 DO
    // Statement section FOR
    IF #Array[#i] > "Max_nr" THEN
      "Max_nr" := #Array[#i];
    END_IF;
  END_FOR;
END_IF;
```

Figure I.9 : Exemple de syntaxe de langage ST

I.3.5.3. Schéma à contacts (LD)

Langage graphique essentiellement dédié à la programmation d'équations booléennes (vrai ou faux), Il utilise les symboles tels que : contacts, relais et blocs fonctionnels et s'organise en réseaux.

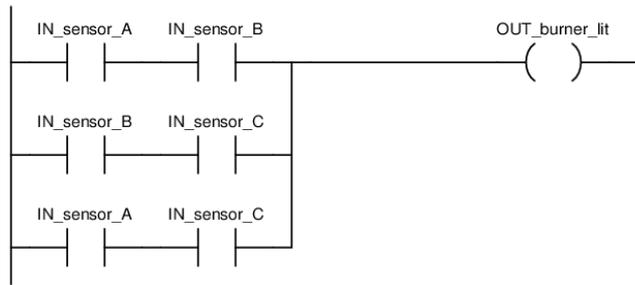


Figure I.10 : Représentation d'un schéma à contact LD.

I.3.5.4. Langage séquentiel graphique (SFC)

Résultant du langage GRAFCET, ce langage de haut niveau permet la programmation aisée de tous les procédés séquentiels. Il offre une représentation graphique des étapes et des transitions du processus, facilitant ainsi la compréhension et la conception de la logique de contrôle.

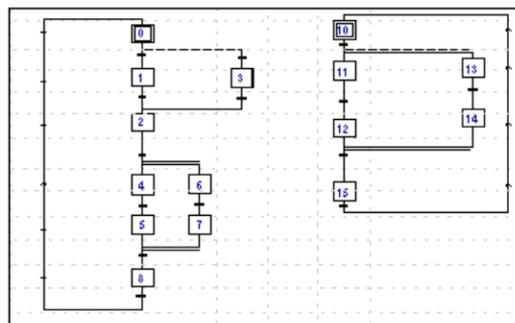


Figure I.11 : Représentation de schéma SFC.

I.3.5.5. Blocs fonctionnels (FBD)

Le langage à relais est basé sur un symbolisme très proche de celui utilisé par les schémas de câblage classiques (voir Figure I.12).

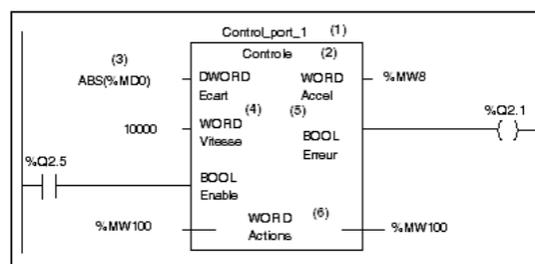


Figure I.12 : Représentation d'un langage à relais FBD.

I.3.6. Module de programmation LOGO Logic

Le module LOGO! (Illustré dans la Figure I.13) est une solution rapide, simple et compacte, idéal pour les tâches de commande de base. Il est devenu incontournable pour les petits projets d'automatisation. Sa conception modulaire lui confère une grande flexibilité, et une large gamme de modules est disponible pour créer des configurations répondant à vos besoins [8].



Figure I.13 : Illustration en image d'un module de programmation LOGO Logic.

I.3.7. Particularités du module LOGO Logic

Des "fonctions prêtes à l'emploi" ont été introduites telles que la temporisation ON/OFF, la minuterie hebdomadaire, le générateur d'impulsions, le chronomètre, etc. qui généralement, sont programmées au préalable dans un automate programmable classique. Plus de 40 fonctions sont également disponibles, il peut également être utilisé pour programmer leurs propres macro-fonctions et créer des bibliothèques à partir de sous-fonctions testées [8].

Le module LOGO ! dispose d'une interface Ethernet et comporte plusieurs versions pour différentes puissances d'alimentation DC 12V/24V RCE, AC/DC 24V RCE, DC 24V CE, AC/DC 115/240V RCE, une mémoire de programme de 400 blocs, d'un emplacement de carte micro SD et d'un serveur web intégré dans toutes les unités de base. Il offre une configuration automatique de la communication avec représentation visuelle dans la vue du réseau, l'appel de jusqu'à trois programmes côte à côte et le transfert facile des signaux d'un programme à l'autre par Glisser-Déposer. Ce ne sont que quelques-unes des nombreuses options pratiques. Il existe avec ou sans écran d'affichage intégré [7].

Parmi les nombreux modèles de modules LOGO, nous travaillerons sur le modèle **LOGO! 8 12/24 RCE** avec écran d'affichage.

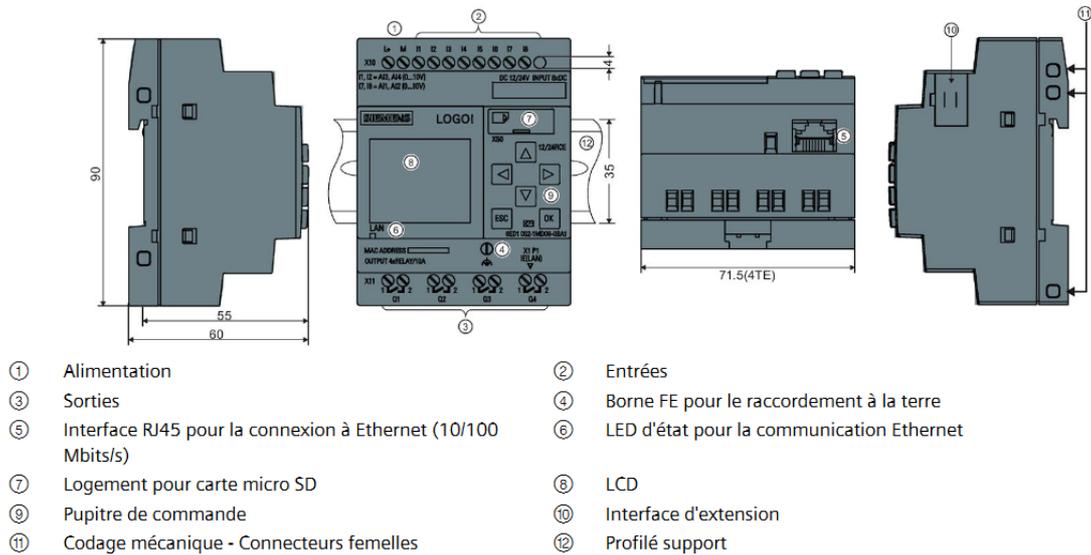


Figure I.14 : Représentation des caractéristiques du modèle LOGO! 8 12/24 RCE.

I.3.8. Avantages du module LOGO!

Le choix de l'utilisation de ce module revient à ses nombreux avantages. Il mémorise le câblage et l'installation et remplace plusieurs commutateurs auxiliaires par ses fonctions intégrées, l'ajout et la modification des fonctions ultérieure ne nécessite pas de modifier le câblage et la connexion à des installations intérieur se fait grâce à l'alimentation intégrée [9].

I.3.9. Entrées/sorties de LOGO! 8 12/24 RCE

On a pour ce module 8 entrées numériques DC 12/24 V, dont 4 peuvent être utilisées comme entrées analogiques (0...10V), 4 sorties numériques. Il est extensible de manière modulaire.

Les entrées sont désignées par la lettre I suivie d'un numéro. Lorsqu'on observe le LOGO ! de face, les bornes des entrées se trouvent en haut (I1, I2, I3, I4, I5, I6, I7 et I8). Les sorties sont quant à elles désignées par la lettre Q suivie d'un numéro (Q1, Q2, Q3 et Q4), de face, les bornes des sorties se trouvent en bas.

I.3.10. Câblage

L'utilisation d'embout n'est pas nécessaire pour les bornes, on peut utiliser des câbles électriques de section maximale de (1x2,5 mm²) et (2x1,5 mm²) pour 1 boîte de borne sur 2. Couples de raccordement : 0,4...0,5 Nm ou 3...4 LBin.

L'entrée L+ est reliée à la phase positive de l'alimentation tandis que l'entrée M est reliée à la phase négative.

On raccorde le module LOGO! au réseau comme illustré dans la Figure I.15 :

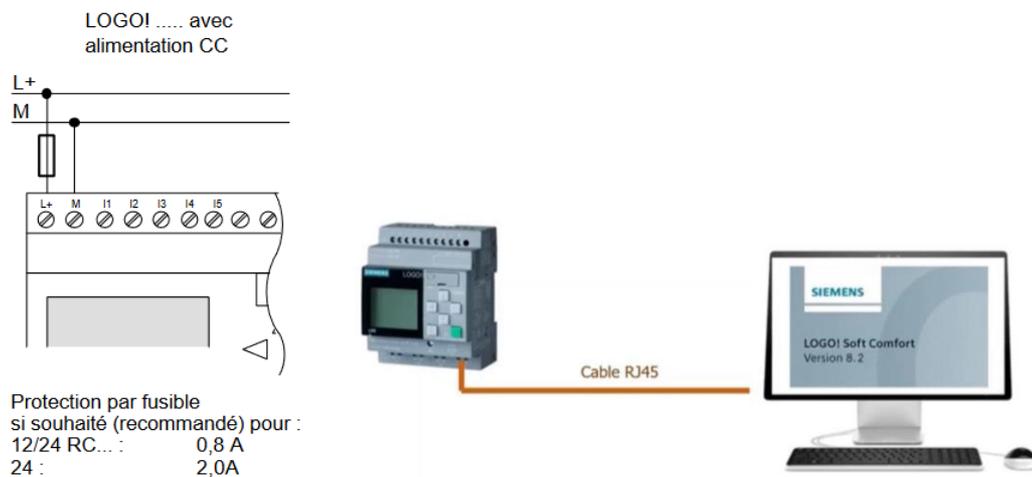


Figure I.15 : Illustration de câblage d'un module LOGO!.

I.3.11. Programmation

Le logiciel de programmation LOGO! Software est conçu pour prendre en charge les tâches d'automatisation simple. Il est utilisé sur PC afin de créer, tester, modifier, enregistrer et imprimer les programmes de commande de manière rapide et est doté d'une configuration et d'un fonctionnement simple [10].

Il est possible de créer des programmes de commutation sous forme de langage à relais ou de Ladder diagramme. La simulation, le test en ligne et l'archivage des programmes de commutation font de ce logiciel l'un des plus utiles et sa flexibilité dû à son vaste étendue de fonctions, de commentaire et d'impression n'est pas négligeable.

Sur le modèle utilisé, la programmation sur écran est possible et est très pratique d'utilisation dans certains domaines de programmation [9].

Menu principal (ESC / >STOP)

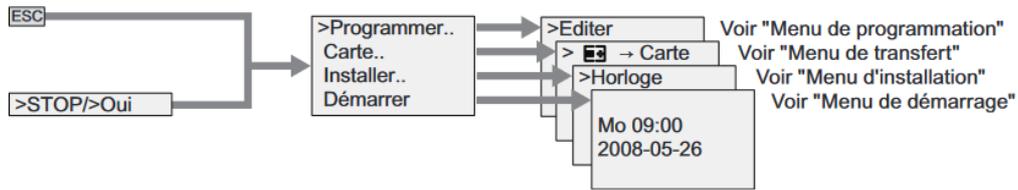


Figure I.16 : Représentation du menu principal sur module LOGO! avec écran.

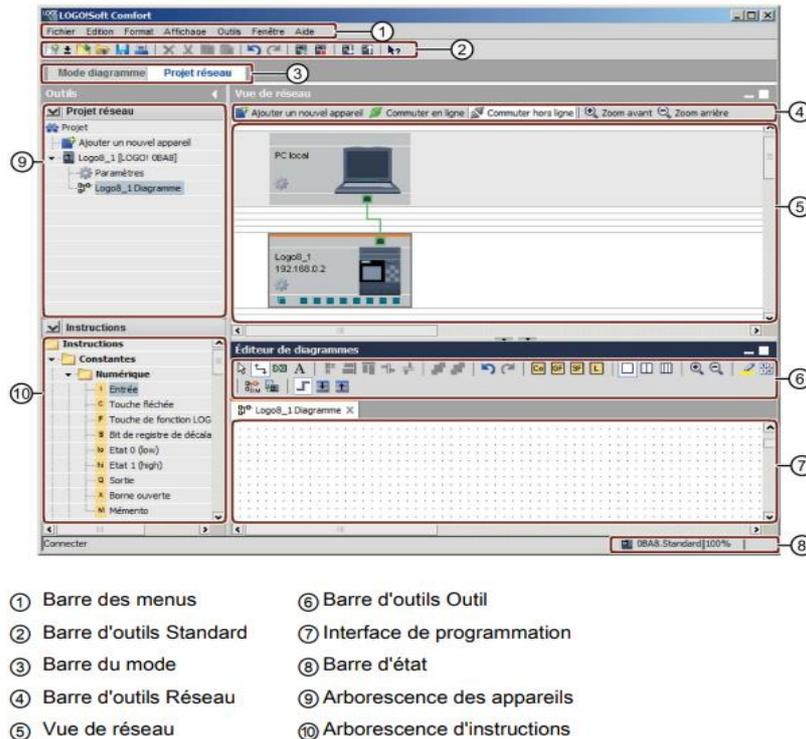


Figure I.17 : Représentation des éléments principaux du logiciel LOGO!.

I.3.12. Programmation sur LOGO!

Nous avons 4 étapes nécessaires à suivre, le changement de mode de fonctionnement, les règles d'entrées/sorties, le fonctionnement du curseur et la planification du programme.

Le mode de fonctionnement se fait par le biais des informations données sur l'écran du module illustré dans la Figure I.18, par exemple : lorsque le système est en mode **RUN**, on peut repasser en mode paramétrage en cliquant sur l'inscription de l'écran “ **ESC** ”.

Pour les entrées/sorties, on doit impérativement procéder de la sortie vers l'entrée, la sortie peut être connectée à plusieurs entrées mais pas inversement.

Les quatre boutons de direction peuvent être utilisés pour le déplacement du curseur et pour la sélection d'une borne ou d'un bloc. Le bouton OK sert à la sélection de la borne ou du bloc et le bouton ESC pour quitter le mode de programmation et pour revenir à l'étape précédente [7].

Le programme doit d'abord être complet et correct afin d'être sauvegardé.

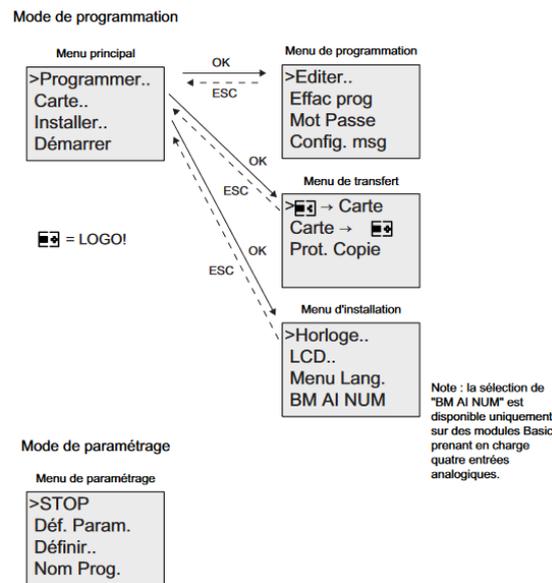


Figure I.18 : Illustration de mode de programmation sur écran LOGO!.

I.3.13. Blocs et numéros de bloc

Un bloc dans LOGO! (voir Figure I.19) est la fonction utilisée pour la conversion des informations d'entrée en informations de sortie. Il permet d'éviter la liaison physique de ces éléments dans l'armoire de commande ou dans la boîte de jonction.

Pour créer un programme de commande, les bornes d'entrées/sorties sont reliées aux blocs par le biais du connecteur adapté dans le menu Co (Connector).

Parmi ces blocs utilisés, nous retrouvons, en fonction du langage utilisés, les opérateurs logiques (AND, OR ...etc) et les fonctions spéciales tel que les relais à impulsion, les compteurs/décompteur et les touches programmables [7].

Vue de l'écran de LOGO!

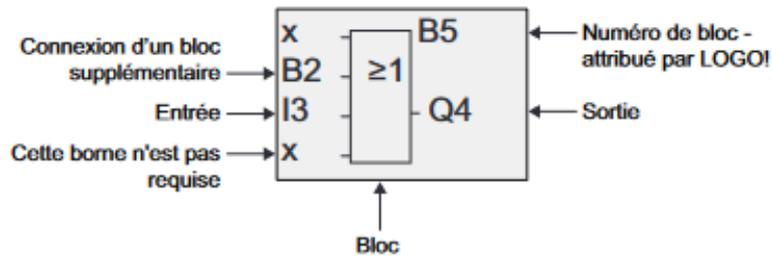


Figure I.19 : Représentation d'un bloc sur écran LOGO.

NB : Seul un bloc peut être représenté à la fois sur l'écran du LOGO!

Le module affecte un numéro de bloc à chaque nouveau bloc dans un programme de commande, il utilise ces numéros pour indiquer la connexion entre eux. Ce type d'affectation permet d'utiliser plusieurs fois les résultats intermédiaires issues de combinaisons logiques ou d'autres opérations, ce qui nous permet ainsi un gain d'espace de saisie et de mémoire [9].

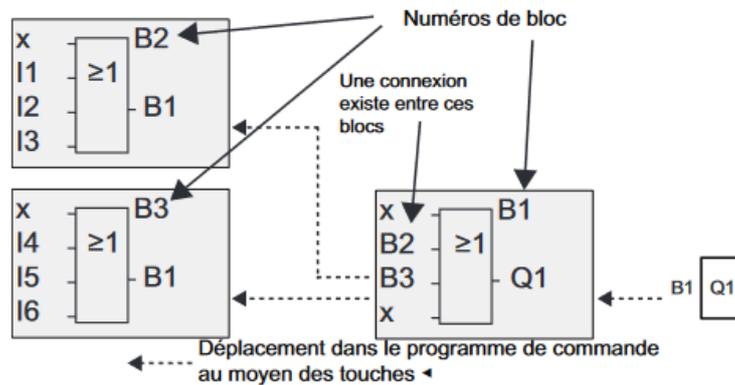


Figure I.20 : Représentation de liaison entre blocs.

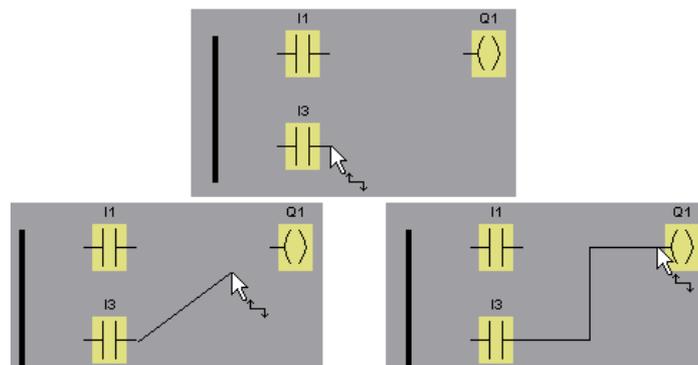


Figure I.21 : Représentation des connexions des blocs.

I.3.14. Logiciel LOGO!

Le logiciel LOGO! Software fournit les fonctions suivantes [9] :

1. Une interface graphique pour la création d'un programme de commande hors ligne sous forme de diagramme CONT (schéma à contact et schéma des circuits) ou sous forme de diagramme de blocs fonctionnels.
2. Simulation du programme de commande sur ordinateur.
3. Génération et impression du schéma d'ensemble d'un programme de commande.
4. Sauvegarde des données d'un programme de commande sur un support externe.
5. Comparaison les programmes de commande.
6. Facilité de paramétrage des blocs.
7. Transfert des programmes de commande de LOGO! vers PC et inversement.
8. Affichage des changements d'états et de variables de processus de LOGO! en mode RUN.
9. Démarrage et arrêt de l'exécution via le PC.

I.4. Conclusion

Au cours de ce chapitre, nous avons débuté en abordant l'entreprise MORTERO, en mettant en lumière ses divers secteurs d'activité et en fournissant certains chiffres clés. Ensuite, nous avons exploré les différents types d'automates programmables, en mettant l'accent sur leur structure modulaire et leur architecture interne. Enfin, nous avons présenté en détail le module LOGO! 8 12/24 RCE que nous avons spécifiquement sélectionné pour notre étude de cas.

Le prochain chapitre sera consacré à l'étude, à la description et à l'automatisation de la chaîne industrielle.

CHAPITRE II

Etude, description et
automatisation de la chaîne
industrielle

II.1. Introduction

La chaîne de remplissage et de fermeture de bouteilles est un processus automatisé essentiel utilisé dans l'industrie, qui consiste à remplir et à fermer des bouteilles de manière efficace. Ce processus nécessite une représentation graphique lui permettant de modéliser son comportement, d'analyser les problèmes posés et de trouver des solutions. Le GRAFCET est le langage graphique que l'on retrouve dans l'industrie.

Dans ce chapitre, nous allons dans un premier temps décrire cette chaîne industrielle ainsi que ses composants, et étudier son fonctionnement de façon générale. Ensuite, nous allons décrire et nous approfondir sur le GRAFCET, les parties le constituant et finir avec la conception du GRAFCET de notre chaîne industrielle.

II.2. Chaîne de remplissage et fermeture de bouteilles

II.2.1. Définition

La chaîne industrielle de remplissage et de fermeture est composée de deux stations et d'un convoyeur (voir Figure II.1).

Une station de remplissage, équipée d'un dispositif permettant de remplir la bouteille avec divers types de produits. Une autre station de fermeture, équipée d'un dispositif qui permet de fermer les bouteilles de manière hermétique, et un convoyeur, qui permet le déplacement des bouteilles durant ce processus jusqu'à la fin de la chaîne.

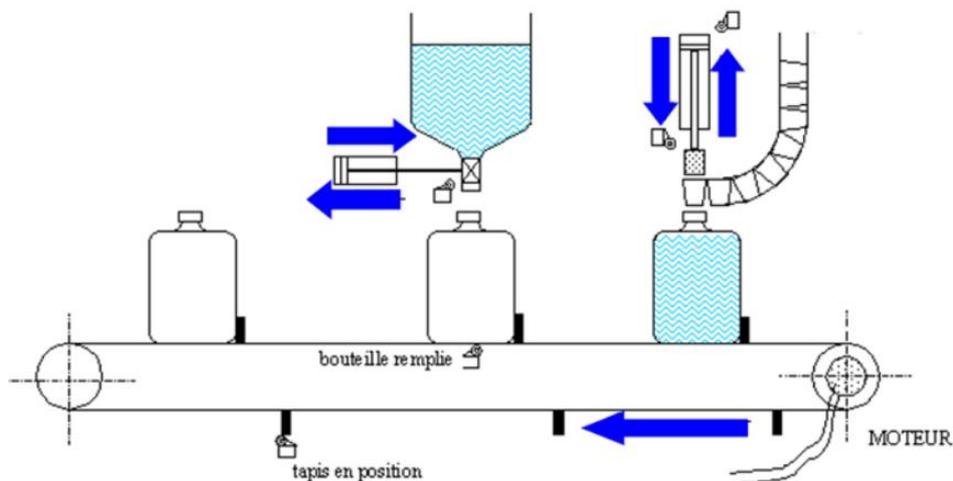


Figure II.1 : Représentation d'une chaîne industrielle de remplissage et de fermeture.

II.2.2. Principe de fonctionnement de la chaîne industrielle

Dans notre étude, la chaîne industrielle représente une chaîne à deux étapes, de remplissage et de fermeture de bouteilles. Nous allons donc présenter le principe de fonctionnement de chacune de ces étapes.

II.2.2.1. Principe de fonctionnement de remplissage de bouteilles

À l'arrivée de la bouteille vide, après sa programmation au niveau de la remplisseuse, le capteur de position de remplissage détecte la bouteille, ce qui enclenche le processus d'arrêt du convoyeur et le début du remplissage. L'électrovanne s'ouvre pour une durée déterminée afin que le liquide soit versé dans la bouteille avec une quantité voulue. Une fois le remplissage terminé, l'électrovanne se referme et la bouteille est déplacée par le convoyeur vers l'autre étape.

II.2.2.2. Principe de fonctionnement de fermeture de bouteilles

La bouteille remplie sera déplacée par le convoyeur et s'arrêtera à l'étape de fermeture avec bouchon (bouchage) quand le capteur aura détecté la bouteille. Le vérin fait descendre l'appareil à fermeture afin que le bouchon soit vissé sur le goulot de la bouteille grâce à son filetage. Une fois le bouchon inséré, le vérin fait remonter l'appareil et le tapis de convoyeur fait sortir la bouteille remplie et fermée de la chaîne.

II.2.3. Description des éléments de la chaîne industrielle

II.2.3.1. Convoyeur

Le tapis, de longueur d'un mètre, reçoit la bouteille vide, la déplace vers chaque étape de la chaîne, et la fait sortir de cette dernière. Il est commandé par un moteur (voir Figure II.2).



Figure II.2 : Illustration d'un convoyeur industriel.

II.2.3.2. Capteurs

Le capteur représenté dans la Figure II.3, est un dispositif électronique qui permet, grâce à sa capacité de conversion des informations d'une grandeur physique en un signal électrique exploitable, de transmettre des informations sur l'état d'un système à la partie commande.

Dans cette chaîne industrielle, on retrouve deux capteurs de présence : le premier servira à détecter la présence des bouteilles au niveau de la remplisseuse, le deuxième servira quant à lui, de les détecter au niveau de la boucheuse.



Figure II.3 : Illustration d'un capteur multifonction.

Tableau II.1 : Représentation des caractéristiques techniques du capteur photoélectrique.

	Caractéristiques
Capteur	Photoélectrique
Type de contact	NO
Tension d'alimentation	12-24 V CC
Connecteur	4 broches
Type de sortie	Sortie PNP
Portée	7m
Type de réflexion	Reflex

II.2.3.3. Actionneurs

L'actionneur est un dispositif de la partie opérative, qui permet d'exécuter l'ordre donné par la partie commande en traduisant le signal électrique en un mouvement physique.

On retrouve dans cette chaîne industrielle deux types d'actionneurs :

- Le moteur.
- Les vérins.

a) Moteur

Les moteurs (voir Figure II.4) sont des machines ayant le rôle d'actionneur qui permettent de convertir une forme d'énergie en un mouvement mécanique alimentant ainsi une sortie.

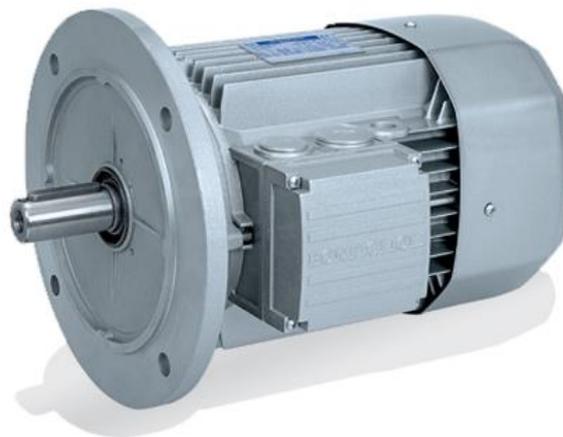


Figure II.4 : Illustration d'un moteur.

Tableau II.2 : Représentation des caractéristiques techniques du moteur.

	Spécifications
Moteur	Triphasé Asynchrone
Indice de protection	IP55 standard
Levier de freinage	Manuel
Refroidissement	Forcé indépendant

b) Vérins

Les vérins sont des actionneurs qui permettent de créer une énergie mécanique à partir de la force de pression d'une énergie hydraulique ou pneumatique. Ils sont composés de deux parties : Une partie fixe, qui est un tube de forme cylindrique et une autre partie mobile, qui représente un piston.

Il existe deux types de vérins : vérin simple effet et vérin double effet. Le vérin simple effet est composé d'un seul orifice d'alimentation, d'une chambre et possède un ressort qui sert au retour de piston, quant au vérin double effet, lui est composé de deux orifices d'alimentation, de deux chambres et le retour de piston se fait en inversant l'alimentation vers l'autre orifice (voir Figure II.5).

Dans ce système, on retrouve deux vérins. Un premier pour le silo de remplissage et le deuxième est un vérin de bouchage.

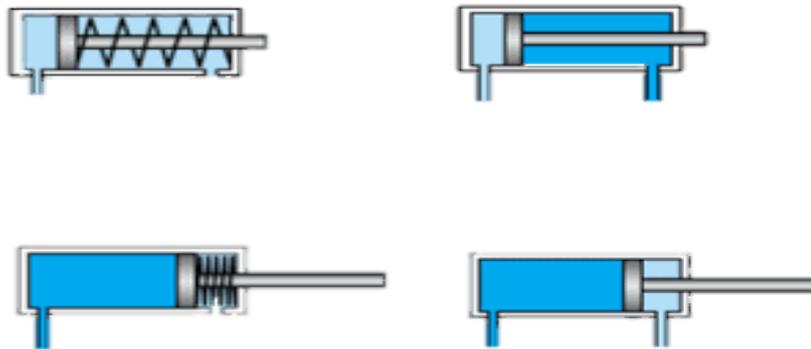


Figure II.5 : Représentation d'un vérin simple effet à gauche et d'un vérin double effet à droite en chaque position.

II.2.3.4. Réducteur de vitesse

Le réducteur de vitesse, illustré dans la Figure II.6, est doté d'un mécanisme permettant de faire baisser la vitesse d'un moteur et augmenter le couple de sortie afin d'avoir une efficacité adaptée à notre système quand le moteur à une vitesse de rotation élevée. Il existe plusieurs différents types de réducteurs en fonction de la technologie qu'ils disposent.



Figure II.6 : Illustration d'un réducteur industriel.

Il existe un dispositif qui rassemble le moteur et le réducteur de vitesse en un seul ensemble nommé le motoréducteur et représenté dans la Figure II.7. Cette chaîne industrielle en est dotée d'un qui assure le fonctionnement du tapis.

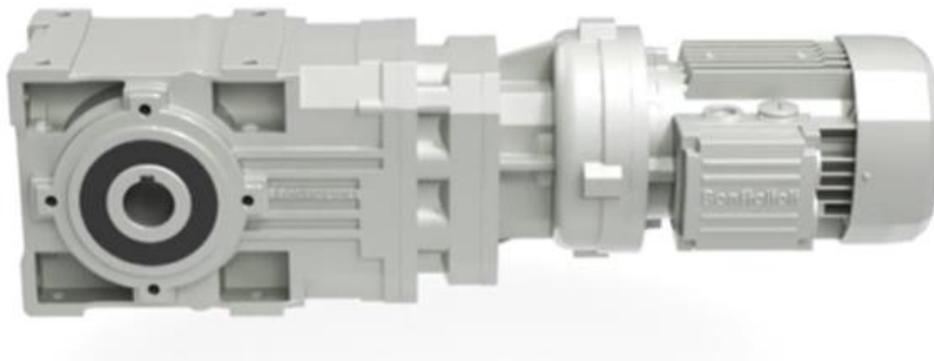


Figure II.7 : Illustration d'un motoréducteur de la marque BONFIGLIOLO.

II.2.3.5. Distributeurs

Le distributeur est un dispositif qui permet de contrôler la quantité et la direction des fluides qui se dirige vers la chambre des vérins, qu'ils soient hydrauliques ou pneumatiques. Plusieurs types existent en fonction du mode de fonctionnement, le Tableau II.1 représente certains modèles de distributeurs avec désignation, commande utilisée et symboles.

Tableau II.3 : Représentations de certains distributeurs.

Désignation	Symbole	Commande	Symbole
Distributeur 2/2		Manuelle par poussoir	
Distributeur 3/2		Par levier	
Distributeur 4/2		Par commande électrique	
Distributeur 4/3		Mécanique par poussoir	
Distributeur 5/2		Par galet	
Distributeur 5/3		Par ressort	

II.2.3.6. Contacteurs

Le contacteur, illustré dans la Figure II.8, est un appareil électrique qui peut commander des actionneurs, à distance avec une commande électrique, ou manuellement, par interrupteur. Il existe de nombreux modèles en fonction des besoins spécifique. On utilise dans ce système un contacteur pour notre moteur de tapis.

**Figure II.8** : Illustration d'un contacteur.

II.2.3.7. Disjoncteur

Le disjoncteur, illustré dans la Figure II.9, est un matériel de protection électrique qui permet d'interrompre le courant électrique dans le cas d'un court-circuit ou d'un passage d'une intensité de courant supérieur à celle que le circuit supporte (surcharge), afin de protéger les équipements électriques de certains dommages et d'assurer une sécurité. Il existe également plusieurs modèles en fonctions de différents dispositifs de protection.

Ce système comporte un disjoncteur principal et un disjoncteur pour le moteur.



Figure II.9 : Illustration d'un disjoncteur.

II.3. GRAFCET

Le GRAFCET, diminutif de « Graphe Fonctionnel de Commande Etapes-Transitions », est un langage utilisé dans le secteur de l'automatisme afin de décrire le cahier de charge d'un système [13].

Il comprend des étapes associées à des actions, des transitions associées à des réceptivités et des liaisons orientées qui relient les étapes aux transitions [11].

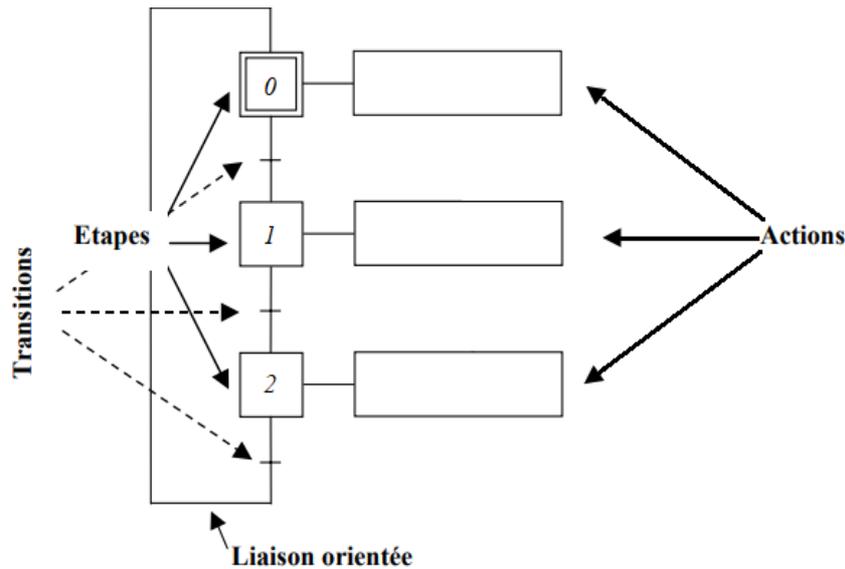


Figure II.10 : Illustration de la structure d'un GRAFCET.

II.4. Description des éléments graphiques du GRAFCET

Le GRAFCET est basé sur un ensemble d'éléments graphiques, chacun à un rôle bien déterminé que sont décrit de cette manière :

II.4.1. Etapes

Une étape représente un état particulier du système qui permet de réaliser une ou plusieurs actions. Elle est représentée par un carré numéroté et possède une variable d'état booléenne qui équivaut à 1 pour une étape active ou 0 si elle est inactive.

On retrouve deux types d'étapes : l'étape initiale, qui représente l'état au repos d'un système, et l'étape normale, associée à une action, et qui se traduit à un ordre vers la partie opérative.

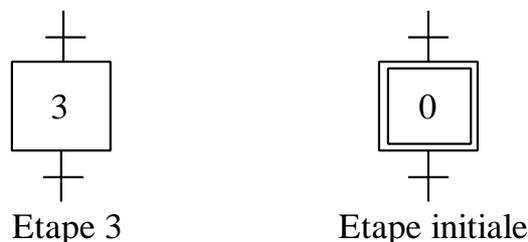


Figure II.11 : Représentation des deux types d'étapes.

II.4.2. Actions

Une action est associée à une étape afin d'exécuter un ordre, elle est représentée par un rectangle à droite de l'étape et peut être sous forme littérale ou symbolique en fonction du GRAFCET utilisé.



Figure II.12 : Représentation d'une action exécutant un ordre A associé à l'étape i.

II.4.3. Transitions

Une transition dans un GRAFCET est l'élément qui indique le passage d'une étape à une autre et représente la condition de cette évolution, elle est représentée par un petit trait.

Chaque transition est associée à une condition logique nommée réceptivité [11].

II.4.4. Réceptivités

Une réceptivité est la condition déterminant le franchissement de la transition. Elle est écrite en utilisant les opérateurs ET, OU, NON et front et est évoluée en fonction de l'état des variables ou des événements du système [11].

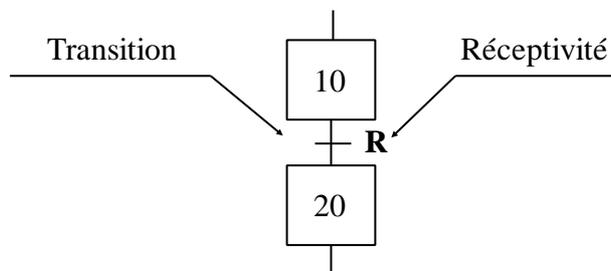


Figure II.13 : Représentation des deux éléments de transition et de réceptivité.

II.4.5. Liaisons orientées

Les liaisons orientées sont des connexions qui relient les étapes et les transitions entre elles, le sens d'évolution est du haut vers le bas sauf dans le cas où l'utilisation d'une flèche d'évolution est indiquée.

II.5. Structures de base

II.5.1. Saut d'étape et reprise d'étape

Le saut d'étape est une sélection qui permet de sauter une ou plusieurs étapes en fonction de conditions d'évolution.

La reprise d'étape est une séquence qui permet de recommencer une ou plusieurs fois une même séquence. Le sens des flèches et la position des transitions ne doit pas être négligée.

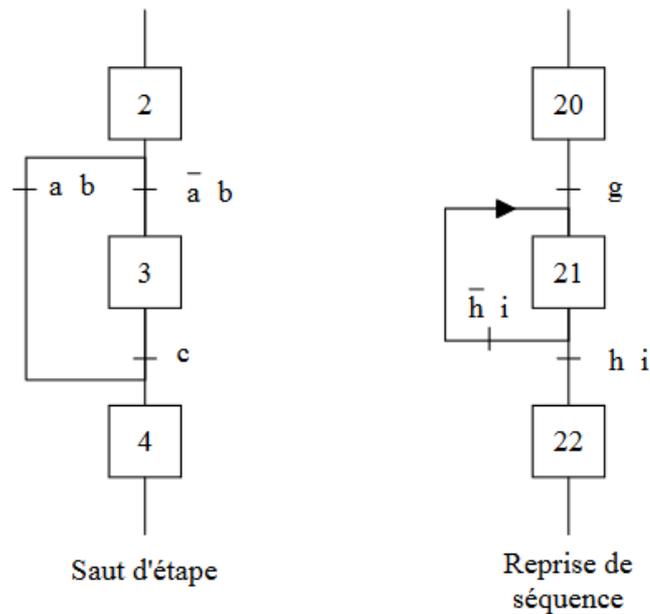


Figure II.14 : Illustration des cas de saut d'étape et reprise d'étape.

II.5.2. Séquence unique

Une séquence unique fait référence à une suite spécifique et non répétitive à travers les étapes et les transitions. En d'autres termes, chaque étape est activée et se déroule dans l'ordre défini les unes après les autres.

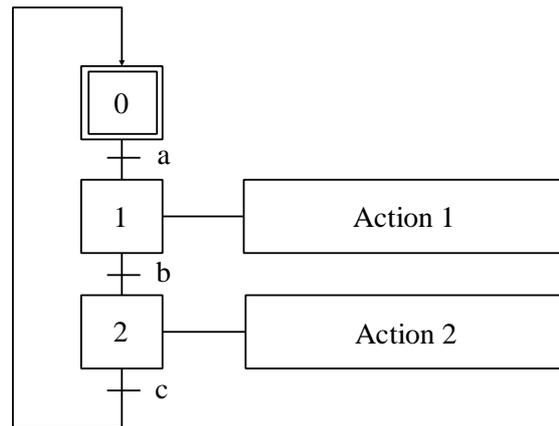


Figure II.15 : Représentation d'une séquence unique.

II.5.3. Séquence simultanée

Une séquence simultanée dans un GRAFCET se produit lorsque plusieurs étapes de deux séquences peuvent être actives et exécutées en parallèle et simultanément, sans ordre de franchissement entre elles (Divergence ET). Les étapes sont liées à une même transition qui les activent.

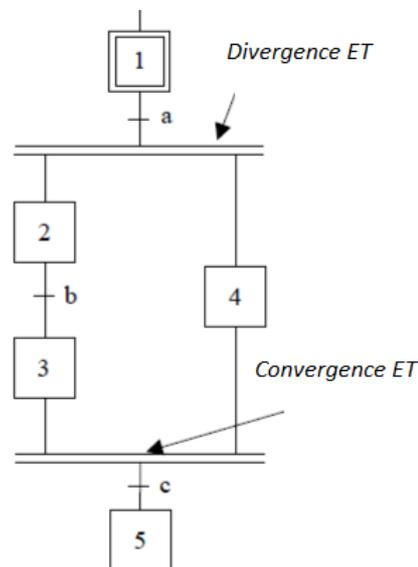


Figure II.16 : Représentation d'une séquence simultanée.

II.5.4. Séquence exclusive

Une séquence exclusive se produit lorsque les réceptivités associées aux transitions ne peuvent pas être simultanément vraies et donc ne peuvent activer qu'une seule transition (Divergence OU).

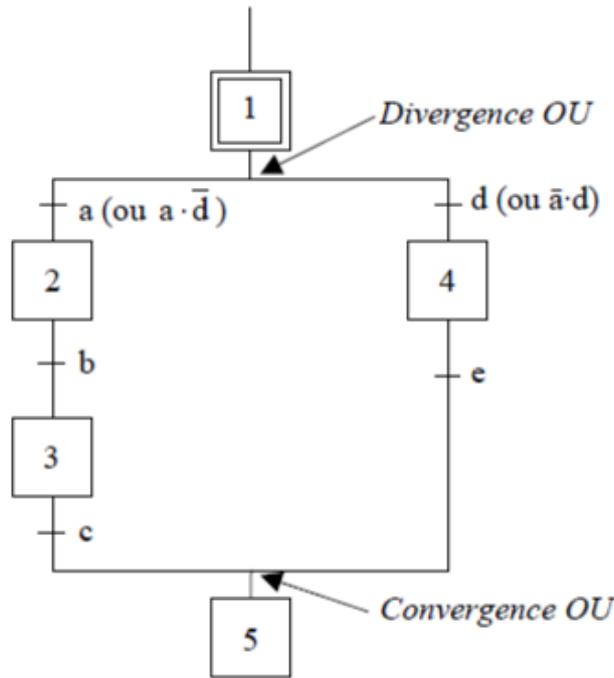


Figure II.17 : Représentation d'une séquence exclusive.

II.6. Niveau d'emploi du GRAFCET

II.6.1. GRAFCET de niveau 1 du point de vue de la partie opérative

Ce GRAFCET décrit, en général, l'enchaînement des actions et des transitions permettant de contrôler le procédé, c'est-à-dire, la description de l'automatisme seul.

II.6.2. GRAFCET de niveau 2 du point de vue de la partie commande

Ce GRAFCET donne une description complète de l'automatisme en tenant compte de toutes les contraintes du procédé et les spécifications technologiques relatifs à la partie commande, le type et la désignation des appareillages (S1, KM, Ka ...) [11].

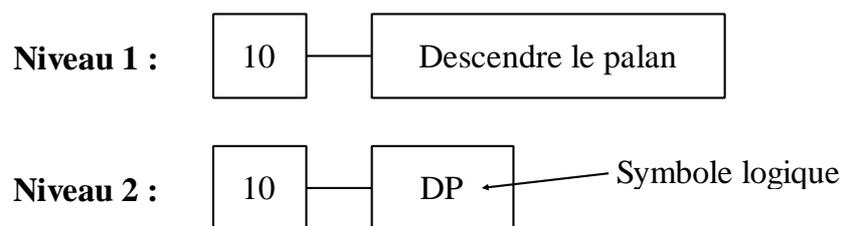


Figure II.18 : Illustration des niveaux d'emploi du GRAFCET.

II.7. Cahier de charge

A l'appui sur le bouton de démarrage, une consigne est donnée afin d'actionner le contacteur pour démarrer le moteur de tapis (état du système activé) qui va faire avancer la bouteille tout au long de la chaîne.

A la présence d'une bouteille au niveau de la zone de remplissage, un capteur de remplissage indique la présence de cette bouteille, le moteur de tapis se met en position d'arrêt, l'électrovanne de remplissage s'ouvre pendant 10 secondes afin de remplir la bouteille, une fois la temporisation écoulée, cette électrovanne se referme et le moteur de tapis se réactive.

Au moment où la bouteille remplie arrive au niveau de la zone de bouchage, le capteur de fermeture indique sa présence, le moteur de tapis se remet en arrêt, la tige du vérin de bouchage sort pendant 10 secondes afin de sceller la bouteille, après l'écoulement de cette temporisation, la tige remonte et le moteur de tapis se réactive.

Une fois la bouteille remplie et fermée, le convoyeur fait sortir la bouteille de la chaîne et le travail se répète à chaque présence de bouteille.

Dans le cas où nous appuyons sur le bouton d'arrêt, une consigne est adressée à l'automate pour forcer le GRAFCET vers la situation initiale, enclenchant l'arrêt de tout le système.

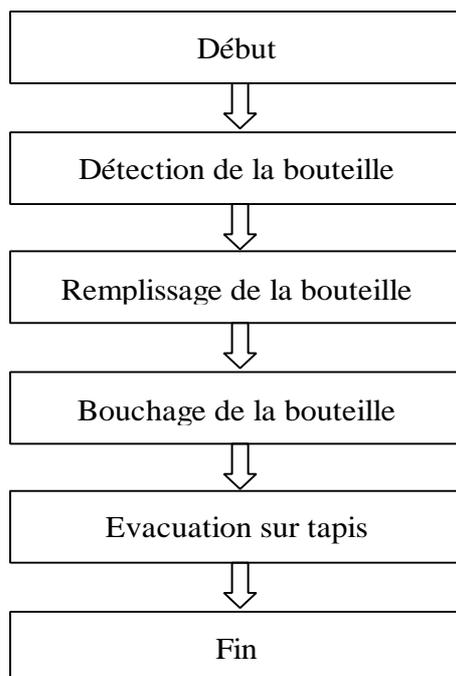


Figure II.19 : Représentation des étapes importante du système étudié et du cahier de charge.

II.8. GRAFCET de niveau 1 de la chaîne industrielle

Une conception d'un GRAFCET pour notre chaîne de remplissage et de fermeture de bouteille est donnée (voir Figure II.19) afin de mieux visualiser le travail que nous allons effectuer lors de la création de notre programme.

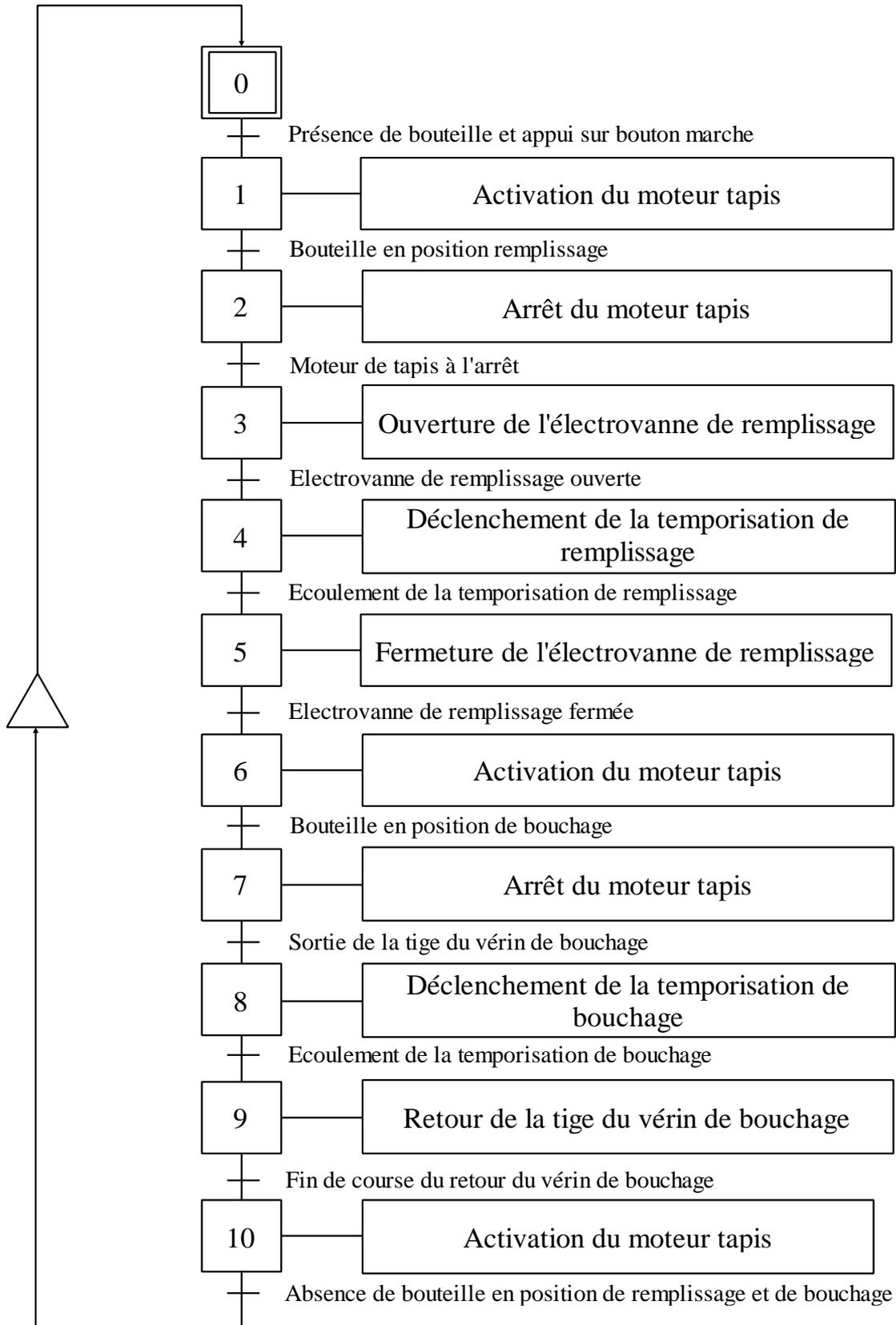


Figure II.20 : Représentation du GRACET du système étudié.

II.9. Conclusion

Dans le secteur de l'automatisme, le GRAFCET est l'outil le plus important et représente la première étape afin de rédiger le cahier de charge pour pouvoir créer un système automatisé.

Au cours de ce chapitre, nous avons commencé par exposer le rôle de notre chaîne industrielle ainsi que les principes de fonctionnement de ses différentes étapes, et avons procédé à la présentation des divers composants de notre machine. Ensuite, nous avons décrit le GRAFCET, ses différentes parties et son emploi, et avons terminé avec la présentation du GRAFCET de notre système et de son fonctionnement. Cette étape nous permettra d'enchaîner et d'avancer avec la prochaine partie.

Le prochain chapitre sera consacré à la partie programmation et simulation.

CHAPITRE III

Programmation et simulation

III.1. Introduction

L'automatisation industriel d'un système est un processus très important, où l'automate exécute des tâches à travers un programme écrit avec un langage déterminé, en prenant en considération certaines règles, afin qu'il puisse être exécuté.

Dans ce chapitre, nous allons décrire en détail le LOGO! Soft comfort V8.3 et présenter le programme de notre système utilisant le langage Ladder.

III.2. LOGO! Soft comfort V8.3

Le LOGO! Soft comfort est un logiciel de programmation des automates programmables industriels de types modulaire, plus précisément de la gamme de module LOGO! du fabricant SIEMENS, avec une utilisation sur PC.

III.2.1 Programmation sur LOGO! Soft comfort V8.3

Le LOGO! Soft comfort est un logiciel qui offre trois langages de programmation. Dans son interface de programmation, il est possible de créer un programme en langage contact, en langage logigramme (diagramme de blocs fonctionnels) et d'utiliser des blocs fonctionnels personnalisés [9].

1. Langage contact (Editeur CONT) : est un programme de commande constitué d'une suite d'instructions. Les entrées sont représentées par des interrupteurs et les sorties par des bobines.

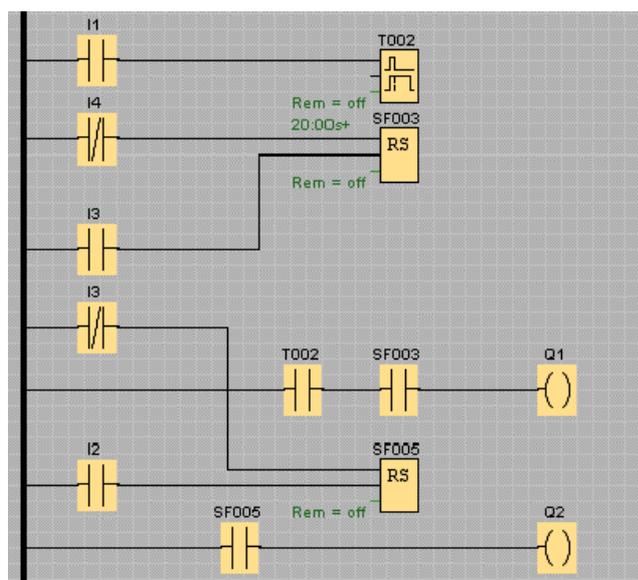


Figure III.1 : Illustration d'une interface d'un programme avec un éditeur CONT.

2. Logigramme (Editeur LOG) : Langage graphique constitué de portes logiques basé sur la manipulation des zones logiques de l'algèbre booléenne.

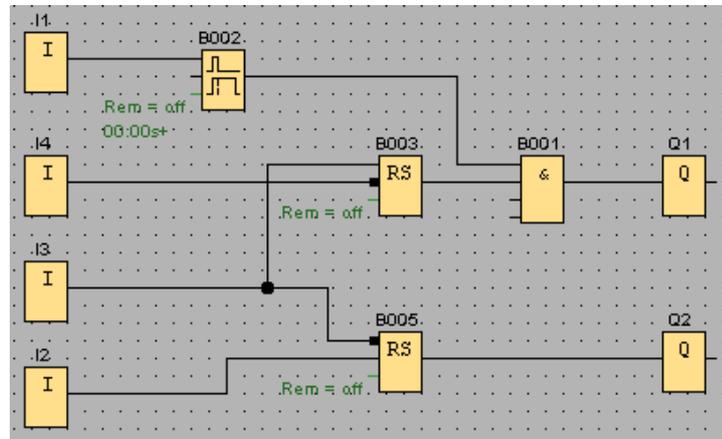


Figure III.2 : Illustration d'une interface d'un programme avec un éditeur LOG.

3. Bloc fonctionnel personnalisé (Editeur UDF) : Diagramme logique d'un groupe de blocs fonctionnels (Editeur LOG) pouvant être utilisé en tant que bloc fonctionnel dans un éditeur LOG.

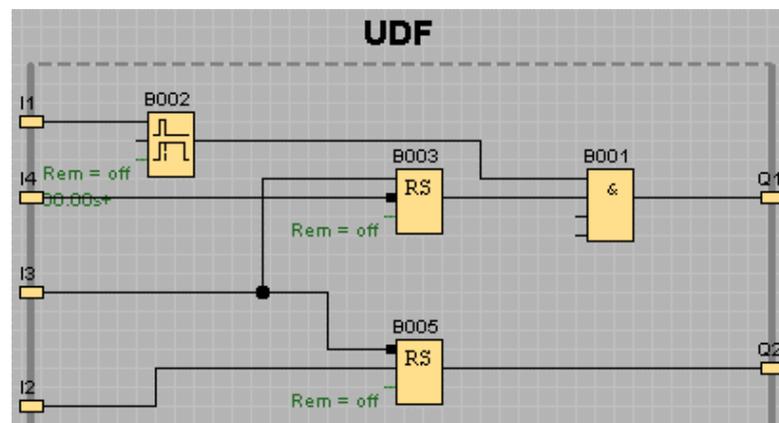


Figure III.3 : Illustration d'une interface d'un programme avec un éditeur UDF.

Durant ce travail, nous avons créé notre projet en utilisant le langage contact (Editeur CONT).

III.2.2 Création d'un projet

III.2.2.1. Démarrage LOGO!

Pour démarrer le logiciel LOGO! Soft Comfort V8.3, il suffit de double-cliquer sur l'icône correspondante. Ensuite, nous ajoutons un nouveau diagramme dans le mode diagramme en sélectionnant l'éditeur de programmation souhaité. Enfin, nous nommons le projet.

Il est important de configurer la sélection du modèle du module LOGO! avant de commencer le développement du programme de commande. Cela permet de masquer les blocs qui ne sont pas compatibles avec le modèle de LOGO! choisi, afin d'éviter toute confusion ou utilisation incorrecte.

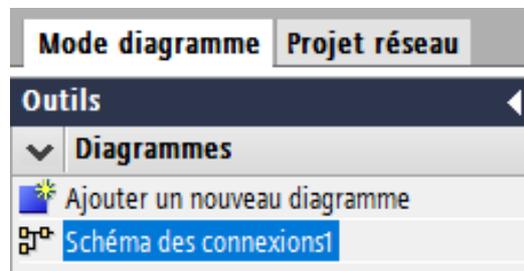


Figure III.4 : représentation d'ajout d'un diagramme dans le mode diagramme.

Le choix de l'éditeur se fait en suivant les étapes suivantes :

1. Cliquez sur "Outils" dans la barre de menu principale du logiciel.
2. Dans le menu déroulant, sélectionnez "Options".
3. Dans la fenêtre des options, sélectionnez la section "Général".
4. Dans la zone de liste déroulante de la partie "Editeur par défaut", vous avez le choix entre "Editeur LOG" et "Editeur CONT" pour l'éditeur standard lors de la création d'un nouveau programme de commande, comme indiqué dans la Figure III.5.
5. Sélectionnez l'éditeur souhaité en fonction de vos besoins.
6. Cliquez sur "OK" pour enregistrer vos modifications.

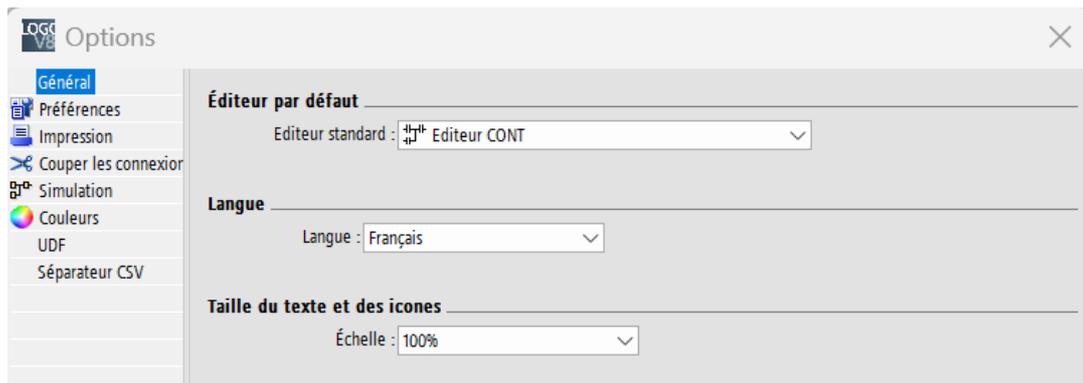


Figure III.5 : Illustration du paramétrage de choix de l'éditeur.

La sélection du modèle se fait en suivant les étapes suivantes :

1. Cliquez avec le bouton droit de la souris sur le nouveau diagramme que vous avez nommé.
2. Dans le menu contextuel, sélectionnez "Propriétés".
3. Sélectionnez l'onglet "Paramètres hors ligne" dans la fenêtre des propriétés.
4. Dans cet onglet, cliquez sur "Type de matériel".
5. Dans la liste déroulante, sélectionnez le modèle approprié. Dans notre cas, on sélectionne : "LOGO! 8.1 & 8.2 (LOGO! 8.FS4)", comme indiqué dans la Figure III.6.
6. Une fois le modèle sélectionné, cliquez sur "OK" pour enregistrer vos modifications.

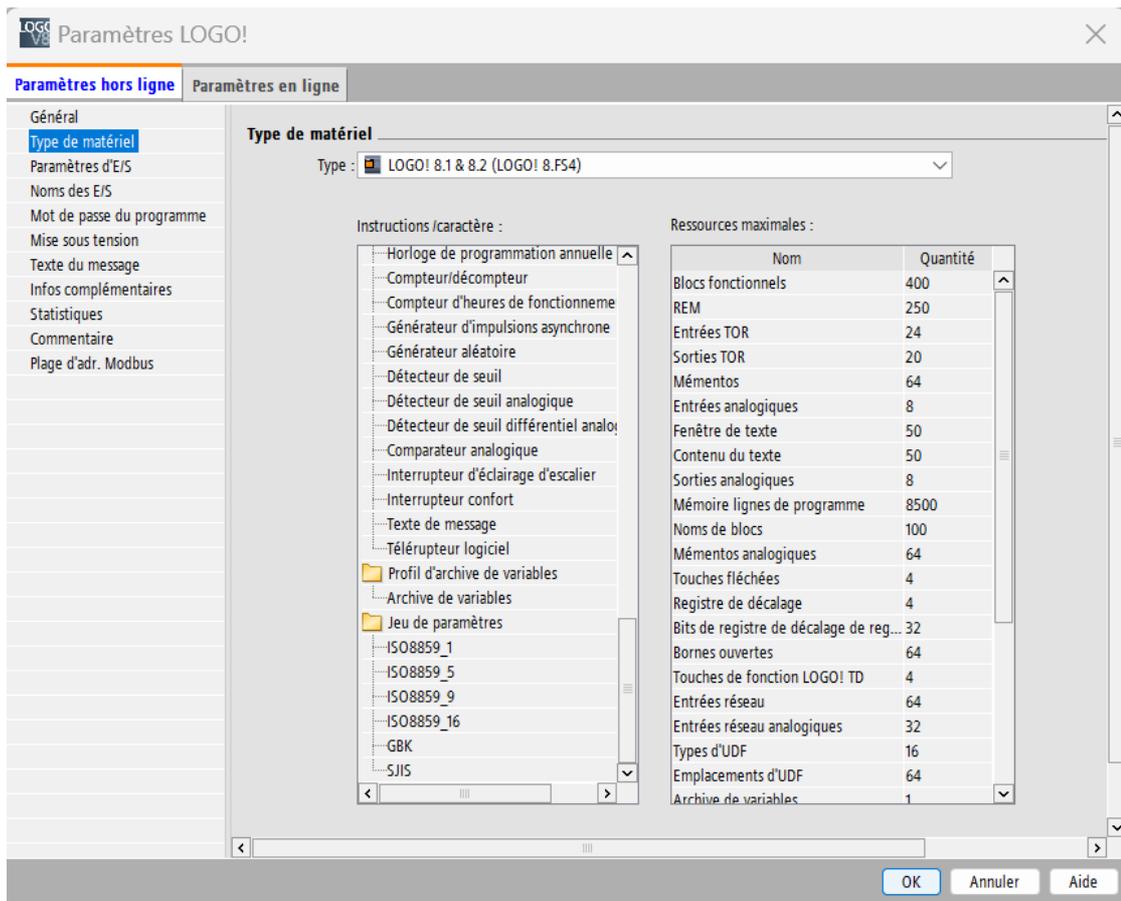


Figure III.6 : Représentation de la sélection du type de matériel LOGO!.

III.2.2.2. Programme de commande

Le développement d'un programme de commande se fait de manière très simple. Dans un premier temps, nous sélectionnons les blocs requis pour ce programme dans la liste d'instructions et les plaçons sur l'interface de programmation. Ensuite, nous procédons à la configuration de chaque bloc en effectuant un double-clic, ce qui affiche des onglets dédiés aux commentaires, aux paramètres et à la simulation, en fonction des blocs. Les interconnexions entre les blocs sont réalisées en dernier afin de finaliser le programme de commande (voir Figure III.7 et Figure III.8).

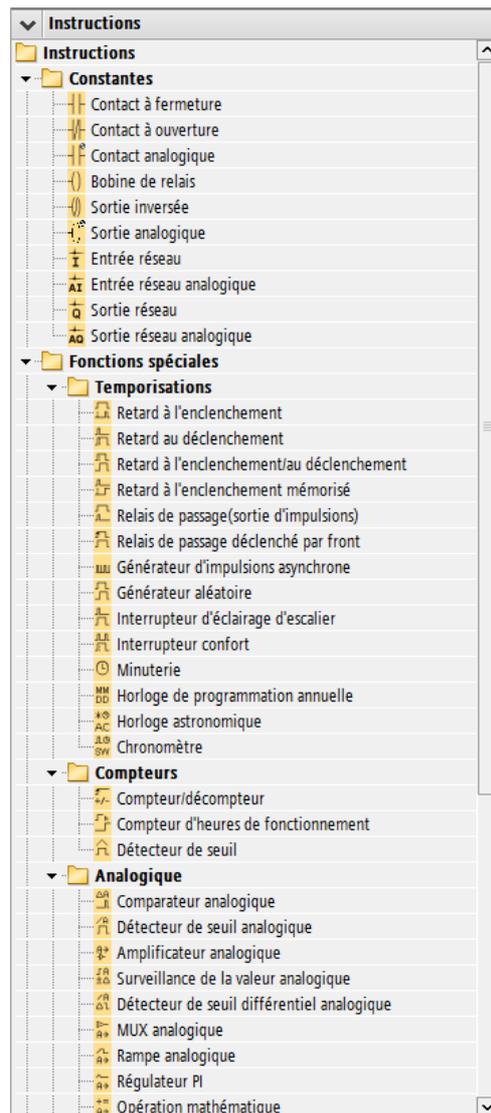


Figure III.7 : Représentation de la liste d'instructions dans la barre d'outils.

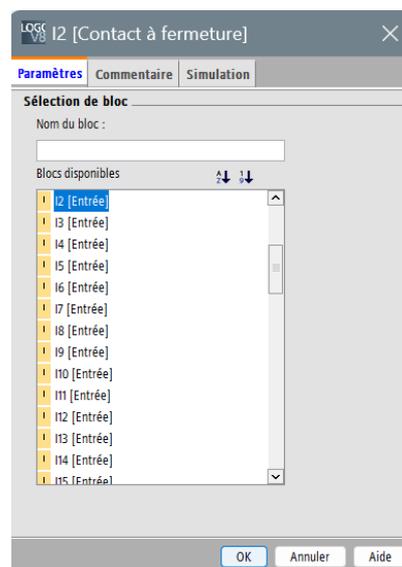


Figure III.8 : Représentation du menu de configuration d'un bloc.

III.3. Simulation

Une fois que le programme de commande de notre système est élaboré, nous passons à l'étape de vérification du bon fonctionnement du système. Cela se fait grâce à la simulation, qui nous permet de valider le programme. La simulation sur le logiciel LOGO! Soft Comfort se fait de manière simple et ne nécessite pas l'intervention d'un autre logiciel. Il suffit de cliquer sur l'icône "simulation" dans l'éditeur de diagrammes (voir Figure III.9 et Figure III.10).



Figure III.9 : Illustration de l'icône simulation dans l'éditeur de diagramme.



Figure III.10 : Illustration de la barre d'outils simulation.

Le tableau III.1 illustre les différentes icônes de commande de la simulation et leurs descriptions respectives.

Tableau III.1 : Illustration des icônes de commande de la simulation.

	Démarrage de la simulation
	Arrêt de la simulation
	Interruption de la simulation
	Simuler réseau. Si aucune adresse IP n'est configurée pour un programme de commande dans LOGO! Soft Comfort, l'icône de simulation de réseau est inactive (grisée)
	Afficher la fenêtre de message
	Démarrage de la simulation pour un intervalle de temps donné ou un nombre défini de cycles. Vous définissez l'intervalle de temps et le nombre de cycles avec les icônes suivantes
	Modification de la temporisation actuelle dans LOGO! Soft Comfort

III.3.1. Configuration de l'état du système

Pour que le système puisse s'enclencher, nous devons dans un premier temps donner les deux états du système. Il est considéré actif lorsque le bouton "Marche" est enfoncé et inactif lorsque le bouton "Arrêt" est enfoncé. Dans le cas où les deux signaux sont émis, le système est impérativement en arrêt par mesure de sécurité.



Figure III.11 : Visualisation de l'étape de configuration de l'état du système.

III.3.2. Configuration du tapis

Le tapis se met en position "Marche" lorsque l'état du système est activé (bouton "Marche") et que tous les autres états (état de remplissage et état de bouchage) sont inactifs.

Le tapis se met en position "Arrêt" si au moins l'une des conditions est satisfaite : si l'état du système est inactif (bouton "Arrêt"), si l'état de remplissage est activé ou alors si l'état de bouchage est activé.

Voici ci-dessous les équations du tapis :

$$\text{Tapis en marche} = (\text{état du système} \wedge \overline{\text{état de remplissage}} \wedge \overline{\text{état de bouchage}})$$

$$\text{Tapis en arrêt} = (\overline{\text{état du système}} \vee \text{état de remplissage} \vee \text{état de bouchage})$$

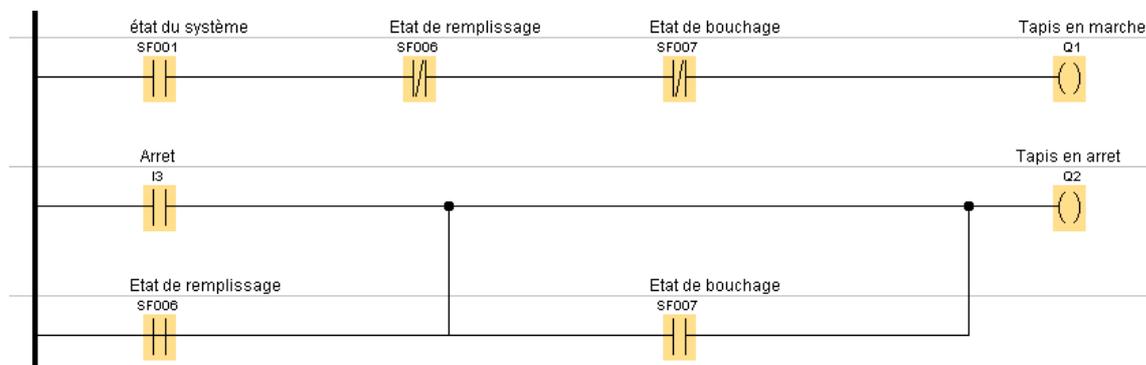


Figure III.12 : Visualisation de l'étape de configuration du Tapis.

Après simulation, nous obtenons les résultats présentés dans la Figure III.13 avec l'appui du bouton Marche (I2) et l'activation du tapis (Q1), et la Figure III.14 13 avec l'appui du bouton Arrêt (I3) et l'arrêt du tapis (Q2).

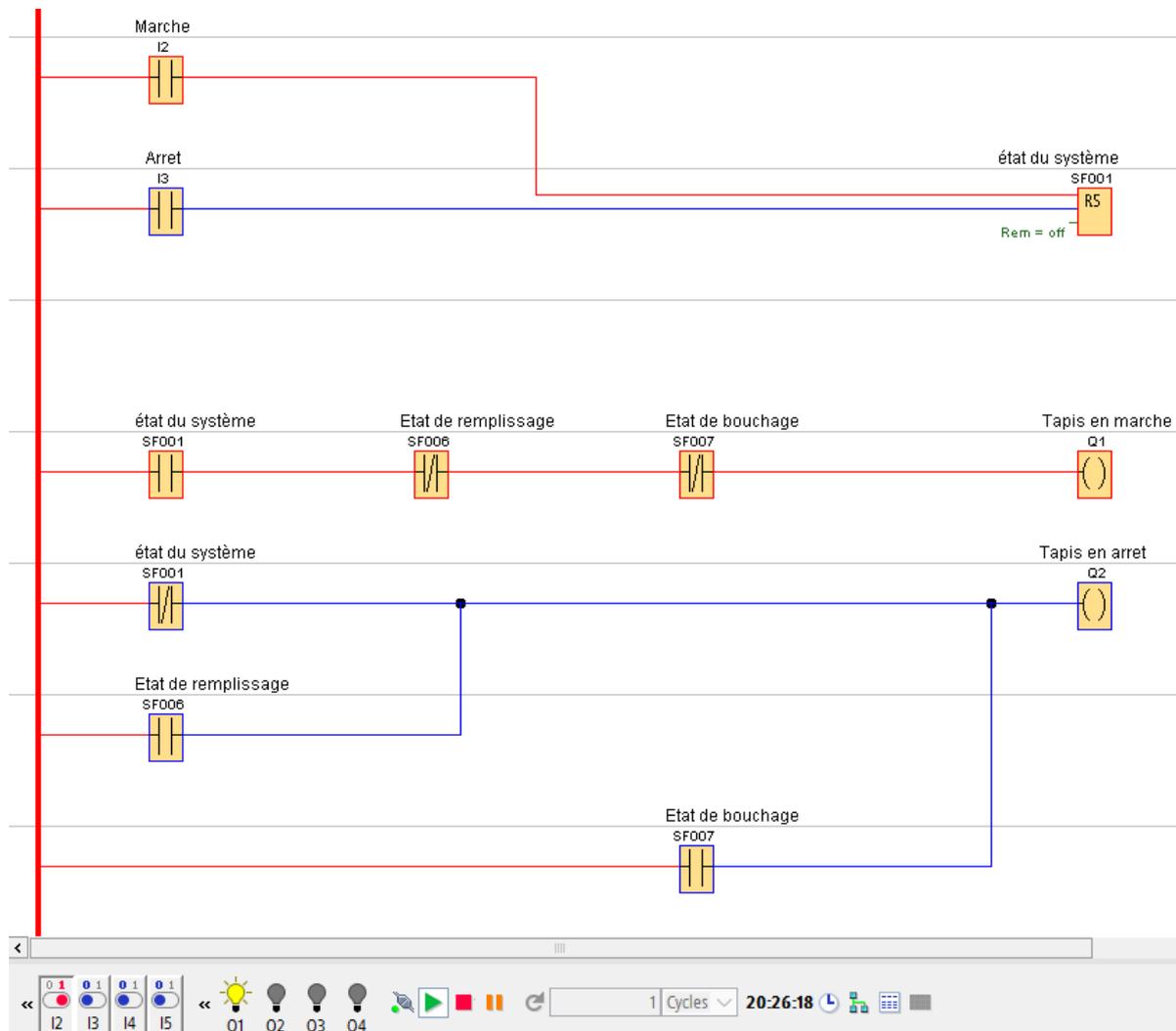


Figure III.13 : Représentation de la simulation du tapis en marche.

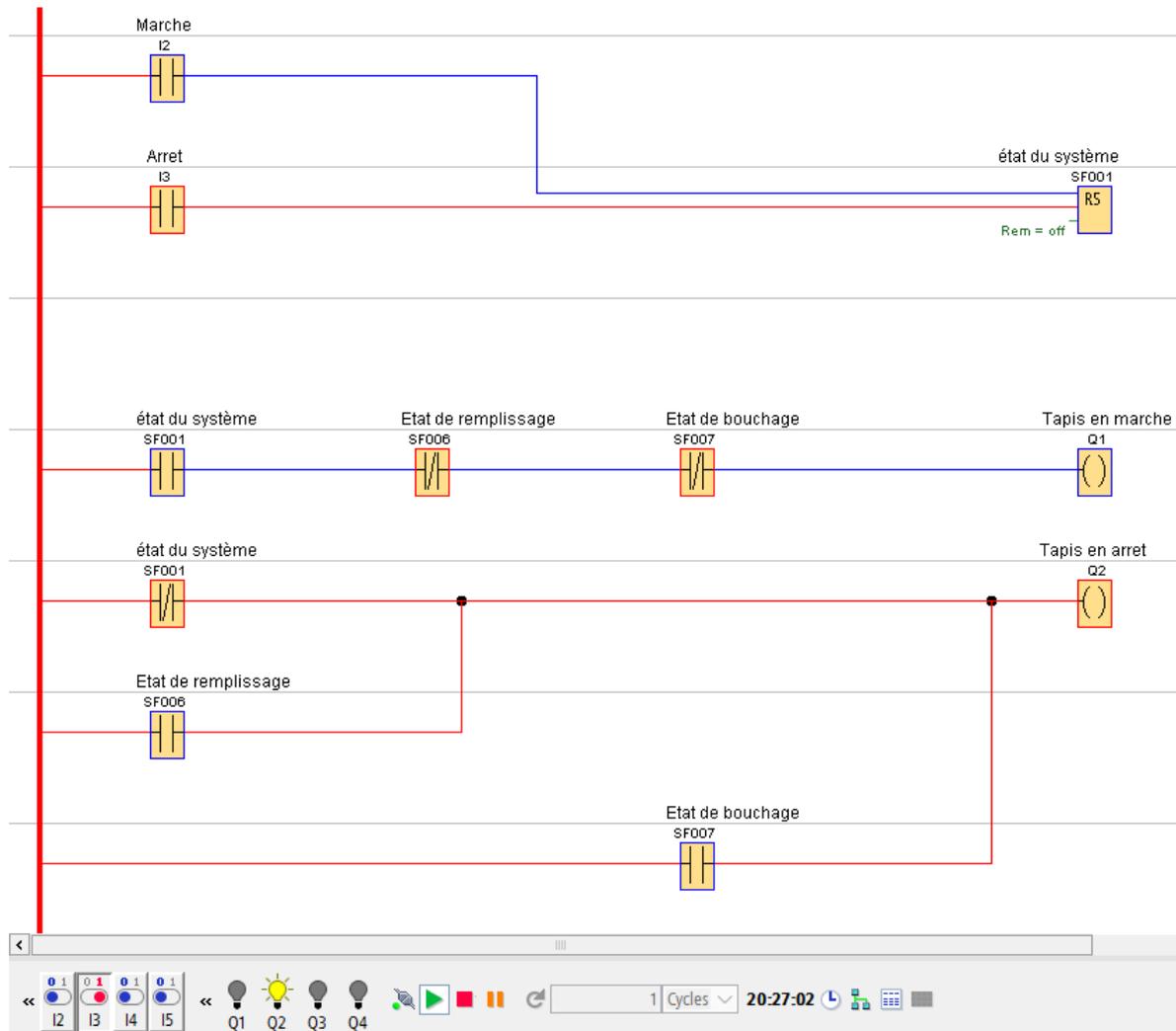


Figure III.14 : Représentation de la simulation du tapis en arrêt.

III.3.3. Etape de présence de bouteille au remplissage

Le capteur de remplissage détecte une bouteille à sa position, ce qui implique l'arrêt du tapis, l'ouverture de l'électrovanne de remplissage avec le décompte de la temporisation de remplissage. Une fois que la temporisation écoulee, l'état de remplissage redevient inactif, l'électrovanne se referme et le tapis se remet en marche.

$$\text{Electrovanne} = (\text{Tapis en arrêt} \wedge \text{position de remplissage} \wedge \text{temporisation})$$

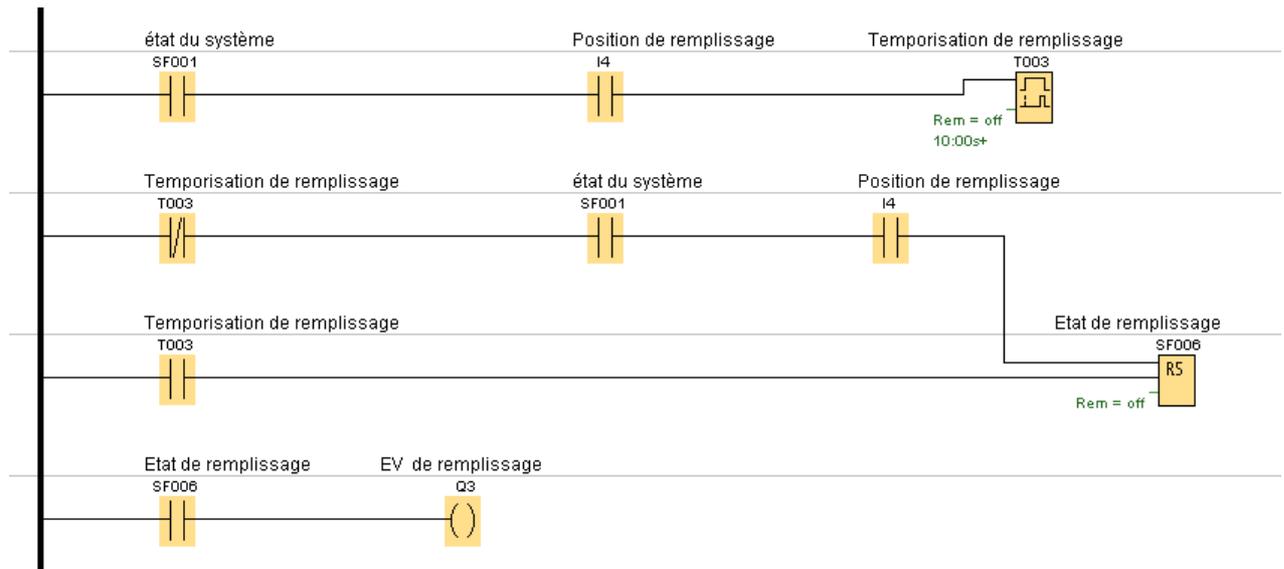


Figure III.15 : Représentation de l'étape de remplissage de bouteilles.

Après simulation, les résultats obtenus correspondent à ceux présentés dans la Figure III.16. On observe la détection de la bouteille au niveau de la position de remplissage (I4), ce qui entraîne l'arrêt du tapis (Q2) et le remplissage de la bouteille (Q3).

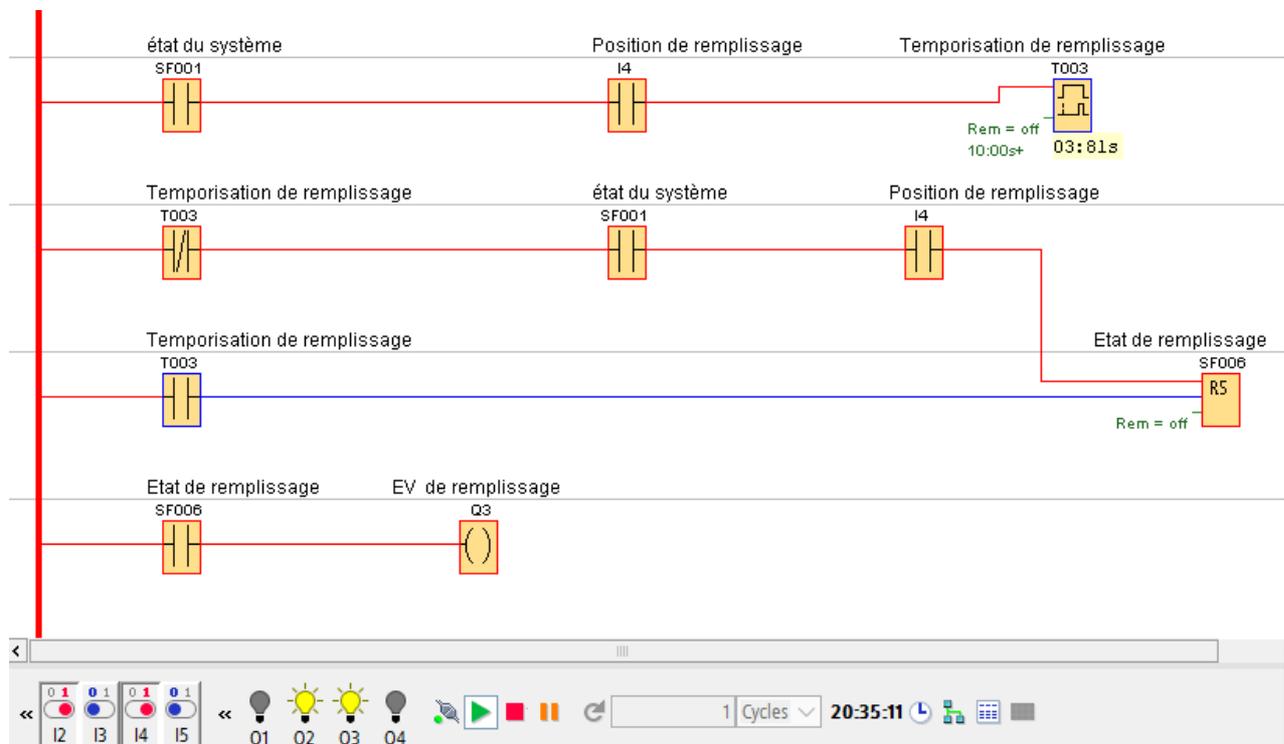


Figure III.16 : Représentation de la simulation de remplissage.

III.3.4. Etape de présence de bouteille à la fermeture

Le capteur de bouchage détecte la présence d'une bouteille à sa position, ce qui provoque l'arrêt du tapis. En même temps, la tige du vérin de bouchage sort et la temporisation de bouchage est enclenchée. Une fois la temporisation écoulée, l'état de bouchage redevient inactif, la tige du vérin remonte et le tapis se réenclenche.

$$\text{Vérin de bouchage} = (\text{Tapis en arrêt} \wedge \text{position de bouchage} \wedge \text{temporisation})$$

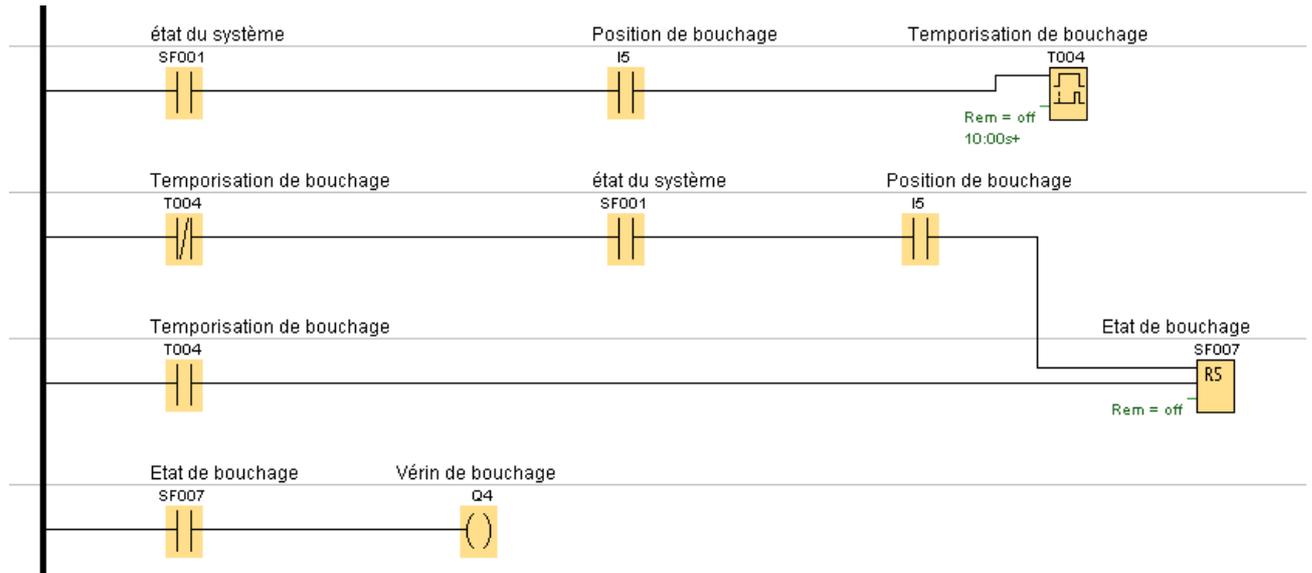


Figure III.17 : Représentation de l'étape de fermeture de bouteilles.

Après la simulation, les résultats obtenus, comme indiqué dans la Figure III.18, montrent la détection d'une bouteille à la position de bouchage (I5), ce qui entraîne l'arrêt du tapis (Q2) ainsi que la fermeture de la bouteille (Q4).

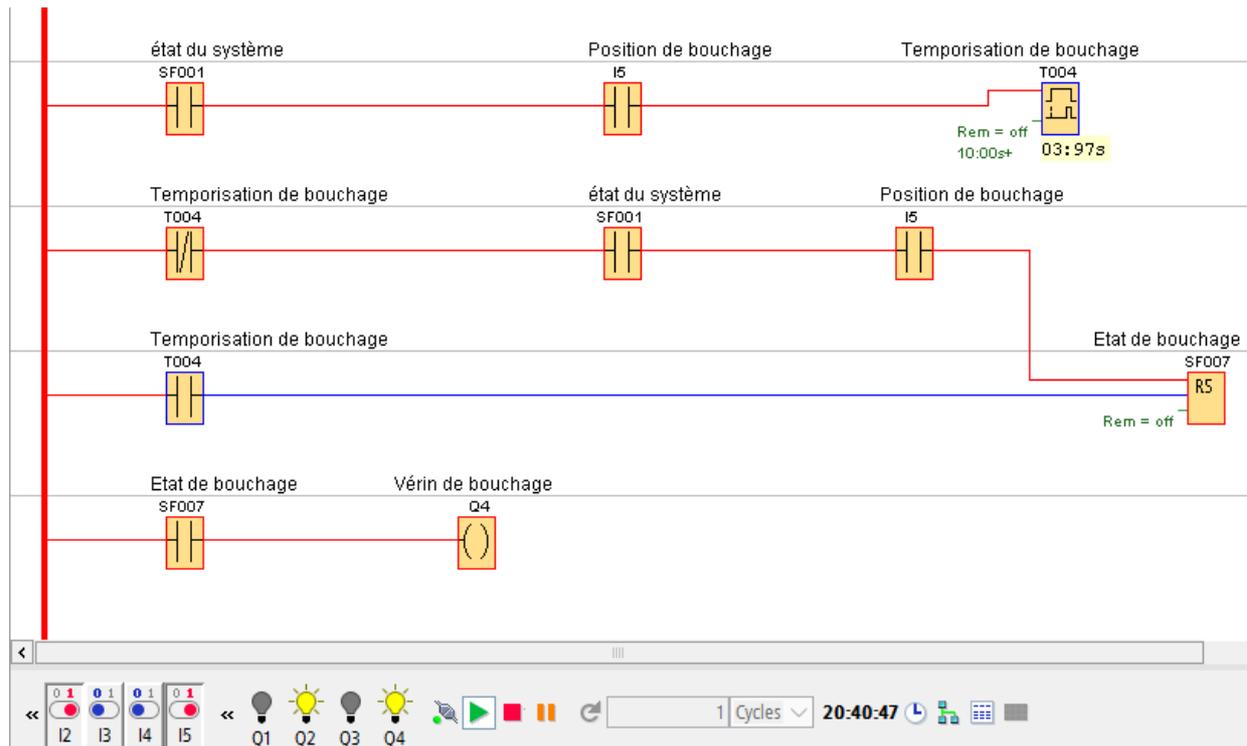


Figure III.18 : Représentation de la simulation de fermeture.

III.4. Eventuels problèmes

Dans un système industriel, plusieurs défauts peuvent survenir. Nous pouvons faire face à un convoyeur faisant déplacer les bouteilles trop rapidement, ce qui revient à la vitesse élevée de son moteur, pouvant causer un entassement. Ce problème peut être réglé grâce à un réducteur de vitesse.

D'autres problèmes, tels que la non-conformité de certaines bouteilles, la défaillance d'un capteur, ou même des problèmes d'usures peuvent avoir lieu. Pour cela, il est essentiel d'analyser attentivement le système, d'identifier les zones problématiques et de mettre en œuvre les mesures correctives appropriées.

III.5. Connexion entre PC et LOGO!

Une fois le programme rédigé, nous passons à l'étape de connexion entre le PC utilisé et notre module LOGO! afin d'établir le transfert du programme. Cette étape nous permet de visualiser le fonctionnement du programme en utilisant les entrées/sorties du module, et surtout d'effectuer notre branchement final avec tous les éléments de notre système (tapis, capteurs, vérins... etc.) pour exécuter les tâches du programme.

III.5.1. Vérification de la connexion

La connexion entre le module et le PC se fait en utilisant un câble RJ45 (liaison Ethernet). Pour cela, nous devons d'abord nous assurer que les deux adresses IP sont compatibles. L'adresse IP du LOGO! peut être retrouvée dans la section "réseau" du menu principal, avec le masque de sous-réseau et la passerelle. Quant à l'adresse IP du PC, nous devons ouvrir l'invite de commande et entrer la commande "ipconfig". Afin que la connexion soit possible, il est impératif que dans chaque chiffre 255 du masque de sous-réseau, les valeurs de l'adresse IP du LOGO! et du PC soient identiques.

III.5.2. Transfert du programme

Une fois que la connexion est établie, nous pouvons procéder au transfert du programme du PC vers le module. Pour cela, nous devons suivre les étapes suivantes :

1. Cliquez sur "Outils" dans la barre de menu principale du logiciel.
2. Dans le menu déroulant, sélectionnez "Transférer".
3. Sélectionnez la section "PC > LOGO!".
4. Dans la section "interface", sélectionnez "se connecter via Ethernet" en choisissant la liaison Ethernet dans la liste déroulante.
5. Indiquez l'adresse IP du LOGO! dans "Adr. IP cible".
6. Effectuez un test pour vérifier la bonne liaison entre le PC et le module LOGO!.
7. Cliquez sur "OK" pour enregistrer les modifications et procéder au transfert du programme.

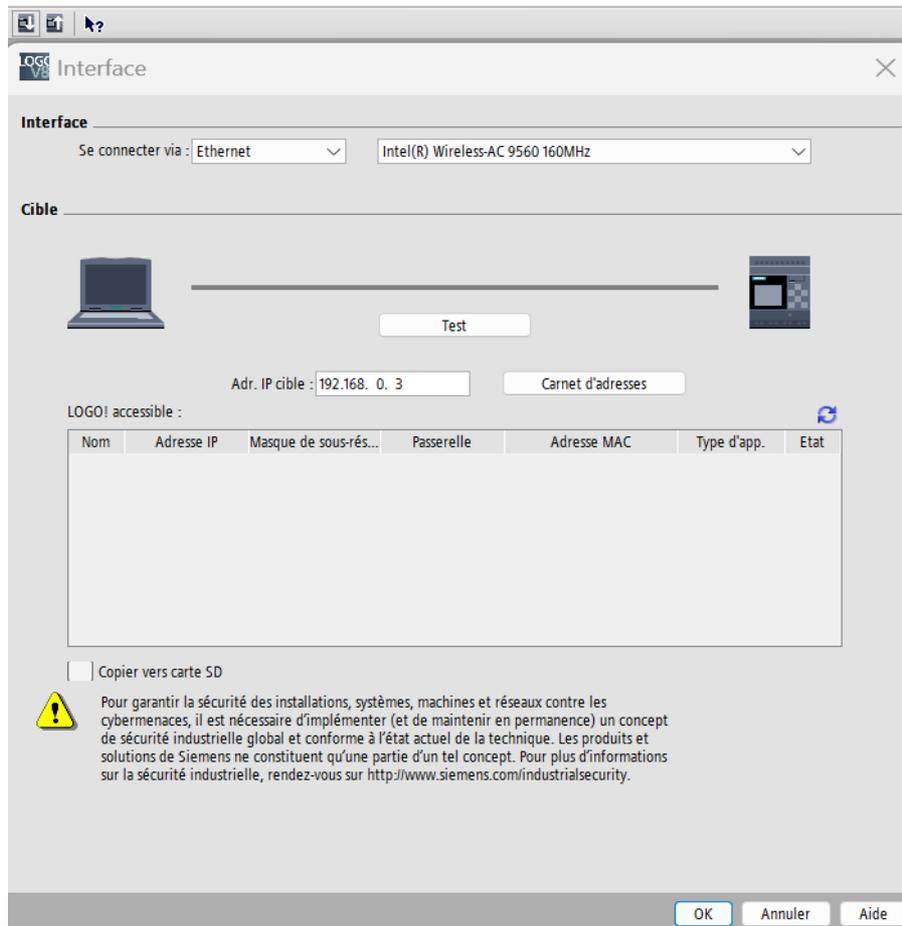


Figure III.19 : représentation de l'interface de transfert PC > LOGO!.

III.6. Branchement de supervision et résultats

Nous concevons la réalisation d'un branchement, remplaçant la partie supervision, qui n'est pas prise en charge par notre logiciel. Ce branchement viendra finaliser notre travail, nous donner une vision en temps réel du fonctionnement de notre programme et nous donner des résultats concrets.

III.6.1. Eléments de notre branchement de supervision

Les éléments utilisés dans cette partie sont représentés dans le Tableau III.2. Nous retrouverons leurs codes ainsi que les éléments qu'ils désignent.

Tableau III.2 : Illustration des codes et désignations des éléments du branchement.

Code	Désignation
DJ AL 220VAL	Disjoncteur Alimentation 220 volts alternative courant
R 220	Répartiteur 220 volts courant alternatif
R 24	Répartiteur 24 volts courant continu
AL 24 VCC	Alimentation 24 volts courant continu
Marche G	Sélecteur marche générale
Contacteur M	Contacteur moteur convoyeur
CPU	Module LOGO!
RL rempli	Relai de remplissage
RL bcn	Relai de bouchon
RL VMM	Relai voyant marche moteur convoyeur
RL VAM	Relai voyant arrêt moteur convoyeur
DIP rempli	Détecteur inductif de proximité de remplissage
DIP bcn	Détecteur inductif de proximité de bouchon
Lumière verte	Marche du moteur convoyeur
Lumière rouge	Arrêt du moteur convoyeur
Lumière violette	Enclenchement du processus de remplissage
Lumière bleue	Enclenchement du processus de bouchage

III.6.2. Développement d'un branchement de supervision

Nous visualisons dans cette partie le fonctionnement du branchement. Lorsque notre disjoncteur "DJ AL 220VAL" est éteint, tout le système est en arrêt et aucune lumière ne s'allume.



Figure III.20 : Position "off" du disjoncteur et arrêt total du système.

Lorsque le disjoncteur "DJ AL 220VAL" est placé, le système se met en marche. La position du sélecteur marche générale "Marche G" est prise en considération. Si le sélecteur est en position 0, le moteur du convoyeur s'arrête et la lumière rouge s'active. En revanche, si le sélecteur est en position 1, le moteur du convoyeur démarre et la lumière verte s'active.



Figure III.21 : Arrêt du moteur convoyeur avec activation de la lumière rouge.

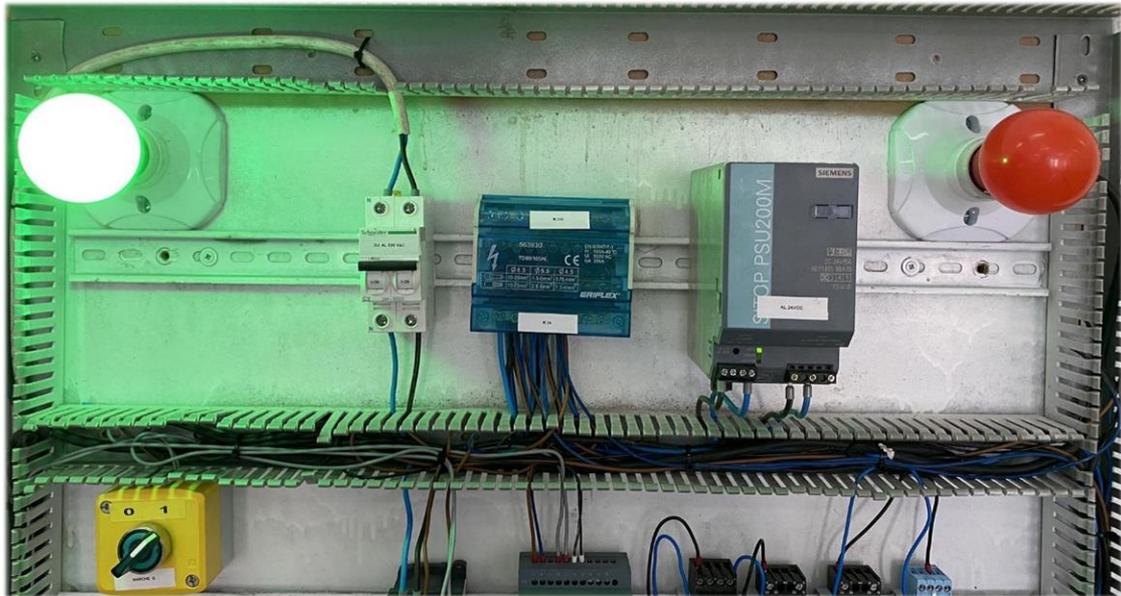


Figure III.22 : Marche du moteur convoyeur avec activation de la lumière verte.

Une fois que le tapis est en marche, les capteurs de remplissage "DIP rempli" et de bouchage "DIP bcn" donnent les instructions et les actions suivantes :

- Détection d'un objet au niveau de "DIP rempli", ce qui implique l'arrêt du moteur convoyeur (lumière rouge) et l'activation de la remplisseuse (lumière violette), suivi d'un retour à l'état initial après écoulement de la temporisation (activation de la lumière verte et arrêt de la lumière violette).
- Détection d'un objet au niveau de "DIP bcn", ce qui implique l'arrêt du moteur convoyeur (lumière rouge) et l'activation de la boucheuse (lumière bleue), suivi d'un retour à l'état initial après écoulement de la temporisation (activation de la lumière verte et arrêt de la lumière bleue).

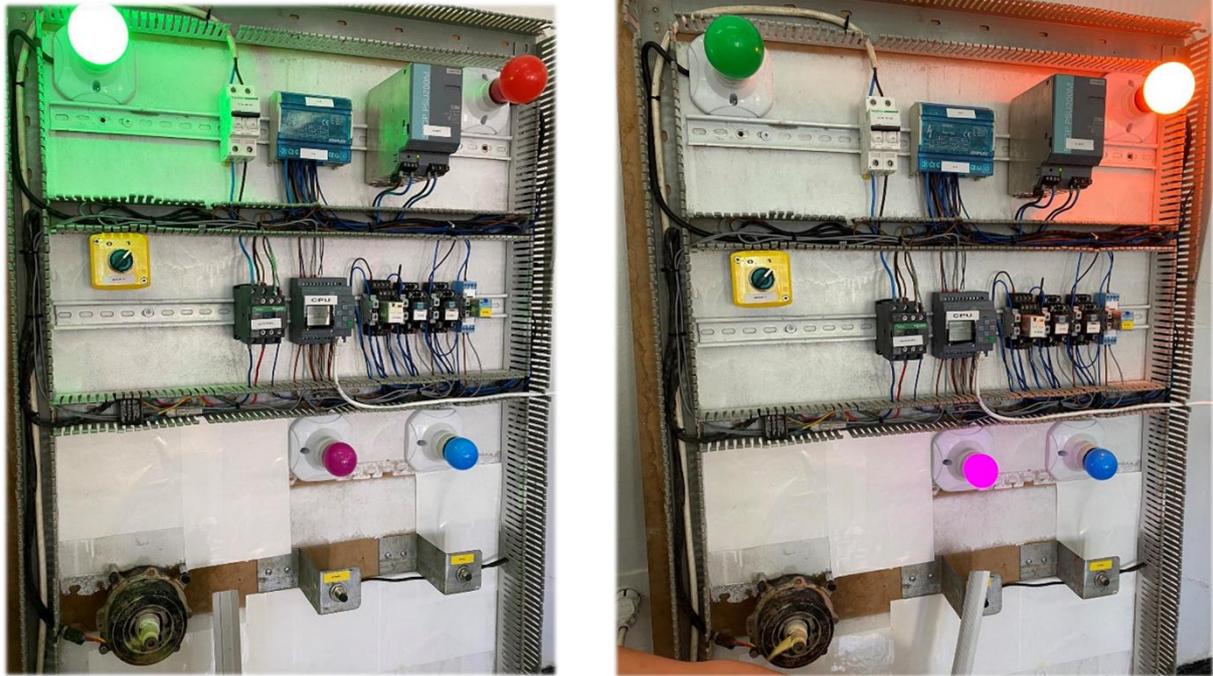


Figure III.23 : Système avant et pendant détection au niveau du "DIP rempli".

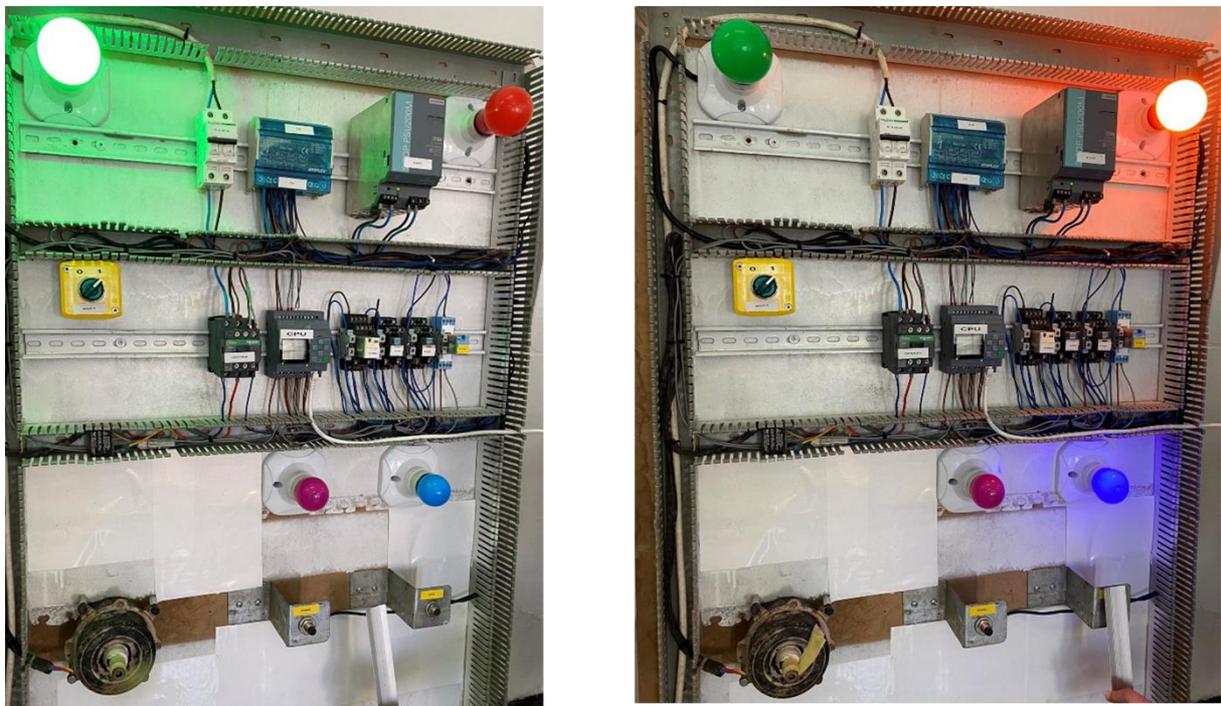


Figure III.24 : Système avant et pendant détection au niveau du "DIP bcn".

III.7. Conclusion

Dans ce chapitre, nous avons présenté le logiciel LOGO! Soft Comfort afin de comprendre ses éléments de base, nous avons ensuite abordé les différents programmes qui nous ont permis de réaliser l'automatisation de notre chaîne de remplissage et de fermeture de bouteilles, en incluant la simulation et le branchement de supervision afin de mieux visualiser son fonctionnement.

Conclusion générale

Conclusion générale

Les automates programmables industriels sont, de nos jours, les constituants les plus répondu quant à la réalisation d'un système automatisé. Ce sont des dispositifs électroniques programmables, utilisés pour automatiser les processus industriels et contrôler les équipements, tout en améliorant l'efficacité des opérations dans différents secteurs industriels. On les retrouve dans des domaines aussi divers que multiples.

Notre modeste travail a traité l'automatisation d'un prototype d'une chaîne industrielle de remplissage et de fermeture de bouteilles. Nous avons entamé par une analyse fonctionnelle du système envisagé et cela nous a permis de décrire le GRAFCET ainsi que le fonctionnement de notre système.

Ce travail nous a permis, d'une part, de nous familiariser avec le module LOGO! 8, idéal pour les tâches de commande de base et est un incontournable pour les petits projets d'automatisations, et son logiciel de programmation LOGO soft confort version 8.3, de découvrir les différences existantes avec d'autres API tels que le SIEMENS S7-1200, et les différences avec d'autres logiciels tels que Tia Portal, ainsi que d'acquérir une certaine expérience professionnelle en côtoyant le monde du travail. D'autre part, il nous a permis de concevoir un branchement au niveau de l'entreprise sur lequel des améliorations pourront être faites, et de travailler concrètement sur un concept qui de base, était un prototype.

Nous espérons que ce modeste travail puisse apporter un plus et constituer un apport supplémentaire et de qualité aux promotions futures.

Pour les perspectives, nous envisageons les extensions suivantes :

- Rendre notre système plus complexe, en ajoutant des sécurités et des conditions supplémentaires.
- Concevoir la machine de ce système afin d'essayer son fonctionnement.

Bibliographie

Bibliographie

- [1] B. Massinissa, D. Brahim. « Automatisation d'une remplisseuse bouchonneuse à l'unité de conditionnement d'huile 5/10L de Cévitall », Mémoire de Master, Université de Béjaïa, Juin 2019.
- [2] M.Gilles « Les A.P.I. Architecture et applications des automates programmables industriels » édition bordas, Paris 1988. Consulté le 13 mai 2023.
- [3] <https://mortero-dz.com/a-propos/>, consulté le 23 mars 2023.
- [4] <https://www.technologuepro.com/cours-automate-programmable-industriel/Les-automates-programmables-industriels-API.htm>, consulté le 25 mars 2023.
- [5] https://fasoeducation.net/espace_eleves/secondaire/eftp/bac_technologique/automates_programmables_industriels/co/grainStructure.html, consulté le 11 avril 2023.
- [6] https://www.geea.org/IMG/pdf/LES_AUTOMATES_PROGRAMMABLES_INDUSTRIELS_pour_GEEA.pdf, consulté le 3 avril 2023.
- [7] https://cache.industry.siemens.com/dl/files/461/16527461/att_82565/v1/Logo_f.pdf, manuel consulté le 26 mars 2023.
- [8] <https://new.siemens.com/be/fr/products/automatisation/systems/industrial/simatic-controller/logo.html>, consulté le 15 mars 2023.
- [9] https://cache.industry.siemens.com/dl/files/807/100782807/att_892881/v1/Help_fr-FR_fr-FR.pdf, manuel consulté le 28 mars 2023.
- [10] <https://new.siemens.com/global/en/products/automation/systems/industrial/plc/logo/logo-software.html>, consulté le 5 avril 2023.
- [11] <https://sabtex.files.wordpress.com/2011/11/cours-api-automatique.pdf>, consulté le 4 mai 2023.
- [12] M. Mohamed, O. Farouk. « Conception d'un système de remplissage d'une installation d'incendie », Mémoire de Master, Université de Boumerdes, Juin 2015.

- [13] C. ABDELJAOUED. « Elaboration d'un système d'automatisme et de régulation d'une unité d'aérocondenseur de vapeur d'eau », Mémoire fin de cycle, université de Carthage, Janvier 2010.

Réalisation d'un système automatisé de remplissage et de fermeture de bouteilles

Résumé : Ce manuscrit présente l'étude d'un système automatisé de remplissage et de fermeture de bouteilles à l'aide d'un automate programmable industriel de type compact. Dans ce travail, nous avons dans un premier temps essayé de se familiariser un peu plus avec les automates programmables industriels, qui sont des dispositifs électroniques programmables, ainsi que leurs concepts de base, leurs architectures et leurs programmations. Ils sont utilisés dans l'industrie afin de contrôler, superviser et automatiser un processus industriel. Après avoir choisi le module LOGO! 8 12/24 RCE, nous nous sommes intéressés à l'étude du système automatisé de remplissage et de fermeture de bouteilles. Nous avons pu découvrir les diverses composantes principales d'une chaîne industrielle, présenter le fonctionnement de notre système, puis décrire et élaborer un GRAFCET qui nous a permis d'entamer la réalisation de notre programme. Tout ceci a abouti au développement du programme à l'aide du logiciel du module LOGO! et à la visualisation des résultats à l'aide de la simulation et d'un branchement faisant office d'une supervision.

Mots-clés : Automate Programmable Industriel, LOGO!, automatisation, module programmable, siemens.

Development of an automated bottle filling and capping system

Abstract : This manuscript presents a study of an automated bottle filling and closing system using a compact-type programmable logic controllers. In this work, we first tried to familiarize ourselves a little more with programmable logic controllers, which are programmable electronic devices, as well as their basic concepts, architectures and programming. They are used in industry to control, supervise and automate an industrial process. Once we had chosen the LOGO! 8 12/24 RCE module, we turned our attention to studying the automated bottle filling and closing system. We were able to discover the various main components of an industrial chain, present the operation of our system, and then describe and draw up a GRAFCET which enabled us to start creating our program. All this led to the development of the program using the LOGO! module software and the visualization of the results using simulation and a connection acting as a supervisor.

Keywords : Programmable Logic Controller, LOGO!, automation, programmable module, siemens.

تنفيذ نظام آلي لتعبئة وإغلاق الزجاجات

هذا المخطوط يقدم دراسة لنظام تعبئة وإغلاق آلي للزجاجات باستخدام وحدة تحكم صناعية قابلة للبرمجة من النوع الموديولار. في هذا العمل، حاولنا في البداية التعرف بشكل أفضل على وحدات التحكم الصناعية القابلة للبرمجة، وهي أجهزة إلكترونية قابلة للبرمجة، بالإضافة إلى مفاهيمها الأساسية وهندستها وبرمجتها. يتم استخدامها في الصناعة للتحكم والرقابة والتشغيل التلقائي لعملية صناعية. بعد اختيار وحدة لوقو! 8 "LOGO! 8 12/24 RCE"، اهتمنا بدراسة نظام التعبئة والإغلاق التلقائي للزجاجات. تمكنا من اكتشاف المكونات الرئيسية لسلسلة الإنتاج الصناعي، وعرض عملية عمل نظامنا، ثم وصفنا ووضعنا مخططاً "GRAFCET" الذي سمح لنا بالبدء في تنفيذ برنامجنا. وقد أدى كل هذا إلى تطوير البرنامج باستخدام برمجيات وحدة لوقو! "LOGO!" وتصور النتائج باستخدام المحاكاة والربط مع نظام مراقبة.

كلمات مفتاحية : وحدة تحكم صناعية قابلة للبرمجة، LOGO!، التشغيل التلقائي، وحدة برمجية، سيمنز.