

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

UNIVERSITE ABDERRAHMANE MIRA DE BEJAIA

Faculté de Technologies
Département Génie Électrique



Mémoire de fin d'étude

En vue d'obtention du diplôme Master en électrotechnique

Spécialité : AUTOMATISMES INDUSTRIELS

Thème

**Commande et simulation via TIA Portal et Factory I/O du système de
convoyage L4 au sein de Cevital**

Présenté Par :

ABDOUNE Bilal

Encadré par :

Mr MELAHI Ahmed

Université de Bejaïa

MR GOUDJIL Aboubekour

Entreprise Cevital

Année universitaire : 2024/2025

Populaire et Démocratique Algérienne République
وزارة التعليم العالي والبحث العلمي
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique



Déclaration sur l'honneur
Engagement pour respecter les règles d'authenticité scientifique dans
l'élaboration d'un travail de recherche

Arrêté ministériel n° 1082 du 27 décembre 2020 (*)
fixant les règles relatives à la prévention et la lutte contre le plagiat

Je soussigné,

Nom : ABDOUNE
Prénom : Bilal
Matricule : 202033014158
Spécialité et/ou Option : Automatismes Industriels
Département : Génie Electrique
Faculté : Technologies
Année universitaire : 2024/2025

et chargé de préparer un mémoire de (Licence, Master, Autres à préciser) : Master 2

Intitulé : commande et simulation via TIA Portal
et factory I/O du système de convoyage L4
du sein de Cevital

déclare sur l'honneur, m'engager à respecter les règles scientifiques, méthodologiques, et les normes de déontologie professionnelle et de l'authenticité académique requises dans l'élaboration du projet de fin de cycle cité ci-dessus.

Fait à Béjaïa le
09/07/2025

Signature de l'intéressé

(*) Lu et approuvé

(*) Arrêté ministériel disponible sur le site www.univ-bejaia.dz/formation (rubrique textes règlementaires)

Remerciements

Je tiens avant tout à exprimer ma profonde gratitude à Dieu, le Tout-Puissant, pour la foi et la force qu'Il m'a accordées tout au long de ce parcours, et qui m'ont permis de mener à bien ce projet jusqu'à son aboutissement.

J'adresse mes remerciements les plus sincères à mes promoteurs, **M. MELAHI** et **M. GOUDJIL**, pour l'honneur qu'ils m'ont fait en acceptant de superviser ce travail. Leur disponibilité, leurs conseils précieux et leur encadrement attentif ont été d'un soutien inestimable tout au long de cette réalisation.

Je remercie également chaleureusement **les membres du jury** pour le temps qu'ils ont bien voulu consacrer à l'étude de ce mémoire, ainsi que pour leur contribution à l'évaluation et à l'enrichissement de ce travail.

À travers ce projet, je souhaite exprimer ma reconnaissance à toutes les personnes qui, de près ou de loin, m'ont soutenu, accompagné ou encouragé durant cette période.

Je tiens aussi à adresser une pensée toute particulière à mes parents et à ma famille, pour leur soutien moral et matériel tout au long de mon parcours universitaire.

Dédicace

Je dédie ce travail à mes parents, source inépuisable d'amour, de force et d'inspiration. Leur exemple de persévérance et de dévouement a guidé chacun de mes pas, et je leur suis profondément reconnaissant pour l'éducation et les valeurs qu'ils m'ont transmises.

À mes frères, **Nadir** et **Abdel Ghani**, pour leur présence bienveillante, leurs encouragements constants et leur énergie positive.

À mes sœurs, **Dalila**, **Souria**, **Hayat** et **Farida**, pour leur soutien indéfectible tout au long de ce parcours.

Et enfin, à mes amis, pour leur présence précieuse, leur écoute et leur appui tout au long de cette étape importante de ma vie.

Sommaire

Introduction générale.....	1
Préambule : Présentation de Cevital.....	2
1 Présentation de l'entreprise Cevital :	2
2 Historique :	2
3 Situation géographique :	3
4 Stratégie de développement :	3
5 Présentation de la margarinerie :	4
5.1 Rez-de -chaussées :	4
5.2 Première étage :	4
5.3 Deuxième étage :	4
6 Activités et missions :	5

Chapitre I : Présentation du système

I.1 Introduction :	6
I.2 Description fonctionnelle :	6
I.3 Description technique du système :	7
I.3.1 Structure mécanique :	7
I.3.1.1 Les convoyeurs :	7
I.3.1.1.1 Convoyeur à bande modulaire plastique droit :	7
I.3.1.1.2 Convoyeur à bande modulaire plastique courbe :	7
I.3.1.1.3 Convoyeur à rouleau libre/motorisé :	8
I.3.1.1.4 Convoyeur à bande motorisée :	8
I.3.1.2 Les vérins :	9
I.3.2 Motorisation :	10
I.3.2.1 Moteur :	10
I.3.2.2 Structure interne du moteur :	11
I.3.3 Capteurs et automatisation :	12
I.3.3.1 Les capteurs :	12
I.3.3.2 Pupitre IHM (interface homme-machine) :	12
I.3.3.3 Automate programmable :	13
I.4 Problématique :	14
I.5 Conclusion :	14

Chapitre II : Les automates programmables et logiciels associés.

II.1 Introduction :	15
II.2 Définition de l'automatisation :	15
II.2.1 Objectif de l'automatisation :	15

II.3 Système automatisé :	15
II.3.1 Définition :	15
II.3.2 Structure d'un système automatisé :	15
II.3.2.1 La partie opérative (PO) :	16
II.3.2.2 La partie commande (PC) :	17
II.3.2.3 L'interface homme-machine (IHM) :	17
II.3.2.3.1 Rôle de pupitre IHM :	18
II.3.2.3.2 Communication avec l'automate :	18
II.3.3 Fonctionnement d'un système automatisé :	19
II.4 Automate programmable industriel :	19
II.4.1 Introduction :	19
II.4.3 Les applications de l'automate programmable industriel :	20
II.4.4 Les types des automates programmables industriel :	20
II.4.4.1 Les automates modulaires :	20
II.4.4.2 Les automates compacts :	21
II.4.5 Structure d'un automate programmable industriel API :	21
II.4.5.1 Unité centrale de traitement (CPU) :	21
II.4.5.2 Mémoire :	21
II.4.5.3 Modules d'entrées/sorties (E/S) :	21
II.4.5.4 Alimentation électrique :	22
II.4.6 Nature des informations traitées par l'automate programmable :	22
II.4.6.1 Tout ou rien (TOR) –signaux binaires :	22
II.4.6.2 Analogique- signaux continus :	22
II.4.6.3 Numérique-signaux codés :	23
II.4.7 Avantage des automates programmables :	23
II.4.8 Langage de programmation pour API :	23
II.4.8.1 Ladder (LD- Ladder Diagram/schéma à contacts) :	24
II.4.8.2 FBD (Function Block Diagram/Diagramme de blocs fonctionnels) :	24
II.4.8.3 ST (Structured Text/Text structure):	24
II.4.8.4 IL (Instruction List/liste d'instruction) :	24
II.4.8.5 SFC (Sequential Function Chart/Grafcet):	24
II.4.9 Critères de choix d'un automate programmable :	24
II.4.10 Présentation de l'automate utilisé S7-1500 :	25
II.4.10.1 Caractéristique d'API siemens SIMATIC S7-1500 :	25
II.5 TIA Portal (Totally Integrated Automation):	26
II.5.1 Définition :	26
II.5.2 Avantage de logiciel TIA Portal :	26

II.5.3 Vue de portail et vue de projet :	26
II.5.4 La vue de portail :	26
II.5.5 La vue de projet :	27
II.5.6 Création d'un projet et configuration d'une station de travail :	28
II.5.7 Configuration et paramétrage de matériel :	29
II.5.8 Adressage des E/S :	30
II.5.9 Adresse Ethernet de la CPU :	31
II.5.10 Compilation et chargement de la configuration matérielle :	31
II.5.11 Présentation des blocs de programmation :	33
II.5.11.1 Bloc d'organisation (OB) (organisation block) :	34
II.5.11.2 Blocs de fonction (FB) :	34
II.5.11.3 Fonctions (FC) :	34
II.5.11.4 Blocs de données (DB):	34
II.6 WINCC SUR TIA Portal :	35
II.7 Factory IO :	35
II.8 Conclusion :	37

Chapitre III : programmation et simulation du système.

III.1 introduction :	38
III.2 Explication de la problématique :	38
III.3 Discussion de la solution :	38
III.4 Matériels et logiciels utilisés :	38
III.5 Modélisation de système dans Factory I/O :	38
III.6 Programmation dans TIA Portal V16 :	39
III.6.1 Création de projet dans TIA Portal V16 :	39
III.6.2 Configuration matérielle :	40
III.6.3 Adressage des entrées sorties :	41
III.6.4 Création des blocs fonctionnel :	42
III.6.4.1 Bloc d'organisation (OB) :	42
III.6.4.2 Bloc fonctionnel (FC) :	48
III.6.5 Compilation et test en simulation sous TIA Portal :	51
III.6.6 Chargement du programme :	51
III.7 Connexion entre Factory IO et TIA Portal via PLCSIM :	52
III.8 Discussion à propos de la solution :	57
III.9 Conclusion :	57
Conclusion générale :	58

Liste des figures

Figure1 : Situation géographique de Cevital.	3
Figure 2 : Unité de la margarine.	5
Figure I.1 : Schéma explicatif du système de transport actuel.	6
Figure I.2 : Convoyeur de type bande modulaire plastique droit.	7
Figure I.3 : Convoyeur de type bande modulaire plastique courbe.	7
Figure I.4 : Convoyeur de type à rouleau libre/motorisé.	8
Figure I.5: Convoyeur de type bande motorisé.	8
Figure I.6 : Vérin de type pneumatique à double effet.	9
Figure I.7 : Électrovanne et distributeur électrique.	10
Figure I.8 : Moteur asynchrone triphasé.	11
Figure I.9 : Structure interne de moteur asynchrone.	11
Figure I.10 : Capteur optique et le réflecteur.	12
Figure I.11 : Pupitre IHM.	13
Figure I.12 : L'API S7-1500.	13
Figure II.1 : Structure d'un système automatisé.	16
Figure II.2 : Représentation de la partie opérative.	16
Figure II.3 : Structure de la partie opérative.	17
Figure II.4 : Présentation de la partie commande.	17
Figure II.5 : Interface homme machine IHM.	18
Figure II.6: schéma fonctionnel d'un pupitre IHM d'un système automatisé.	19
Figure II.7 : Types d'automates programmables.	21
Figure II.8 : Structure interne d'un automate programmable.	22
Figure II.9 : API Siemens SIMATIC S7-1500.	25
Figure II.10 : Vue de portail.	27
Figure II.11 : Vue de projet.	28
Figure II.12 : Création de projet.	28
Figure II.13 : Configuration et paramétrage de matériel.	29
Figure II.14 : Configuration et paramétrage de matériel.	30
Figure II.15 : Adressage des E/S.	30
Figure II.16: Adresse Ethernet du CPU.	31
Figure II.17 : Compilation et chargement.	32
Figure II.18 : Chargement de programme dans un automate virtuel.	32
Figure II.19 : Chargement de programme dans un automate virtuel.	33
Figure II.20 : Blocs de programmation.	34
Figure II.21 : Représentation de IHM.	35
Figure II.22 : Vue de Factory IO.	36
Figure II.23 : L'interface principale de Factory IO.	36
Figure III.1 : Représentation de la maquette.	39
Figure III.2 : Création de projet dans TIA Portal.	39
Figure III.3 : Vue d'ajout de CPU.	40
Figure III.4 : Représentation des caractéristiques de CPU.	40
Figure III.5 : Configuration et paramétrage du matériel.	41
Figure III.6 : Onglet de la table de variable.	42
Figure III.7 : Création de réseau dans l'OB1.	43
Figure III.8 : Réseaux du bloc OB.	43
Figure III.9 : Réseau d'appel de la fonction-S7-1500.	44
Figure III.10 : Réseau d'appel de la mise en marche.	44
Figure III.11 : Réseau d'appel de commande_M1.	44

Figure III.12 : Réseau d'appel de commande_P1.....	44
Figure III.13 : Réseau d'appel de commande_M2.	45
Figure III.14 : Réseau d'appel de commande_P2.....	45
Figure III.15 : Réseau d'appel de commande_M3.	45
Figure III.16 : Réseau d'appel de commande_M4.	45
Figure III.17 : Réseau d'appel de descenseur I.	46
Figure III.18 : Réseau d'appel de la commande_M5.	46
Figure III.19 : Réseau d'appel de la commande_M6.	46
Figure III.20 : Réseau d'appel de la commande_M7.	47
Figure III.21 : Réseau d'appel de descenseur II.	47
Figure III.22 : Réseau d'appel de FC12.	47
Figure III.23 : Réseau d'appel de la commande_M8.	47
Figure III.24 : Les blocs FC.	48
Figure III.25 : Mise en marche de système convoyage.	48
Figure III.26 : Démarrage M1.	49
Figure III.27 : Pousseur P1 avant.	49
Figure III.28 : Chargement descenseur 1.....	50
Figure III.29 : Déchargement descenseur 2.....	50
Figure III.30 : Configuration et détection de l'automate simulé.	51
Figure III.31 : Validation et préparation au chargement.	51
Figure III.32 : Démarrage l'exécution de programme.....	52
Figure III.33 : Vue de l'environnement virtuel.	52
Figure III.34 : Représentation de choix de Siemens S7-PLCSIM.....	53
Figure III.35 : Connexion entre Factory I/O et simulateur PLCSIM.	53
Figure III.36 : Démarrage fonctionnement de système.	54
Figure III.37 : Déplacement des cartons.....	55
Figure III.38 : Chargement du descenseur1.....	55
Figure III.39 : Déchargement du descenseur 1.....	56
Figure III.40 : Vue du fonctionnement de l'environnement virtuel.....	56

Introduction générale

Introduction générale

L'évolution des systèmes de production industrielle repose aujourd'hui sur l'automatisation, est largement adoptée pour améliorer la compétitivité, la qualité et la sécurité des processus. Grâce aux avancées technologiques, les entreprises cherchent à optimiser leur productivité tout en réduisant les coûts, les temps d'arrêt et les risques humains. Dans ce contexte, les automates programmables industriels (API) associés à des logiciels de simulation et de supervision jouent un rôle central dans la modernisation des lignes de production.

L'objectif de ce mémoire est de présenter une solution d'automatisation appliquée à une anomalie observée au sein de l'unité de margarinerie de l'entreprise Cevital. Il s'agit d'un système de convoyage de cartons, dont le fonctionnement initial, basé sur un déplacement par inertie, présentait des limites importantes en termes de sécurité et de pertes de produits.

Face à cette problématique, une solution innovante a été proposée : la conception et la mise en œuvre d'un système de convoyage motorisé automatisé, contrôlé via un automate programmable industriel SIMATIC S7-1500 et programmé à l'aide de l'environnement TIA Portal.

Pour valider cette solution, une simulation virtuelle a été réalisée à l'aide du logiciel Factory I/O, permettant de tester le comportement du système dans un environnement réaliste avant tout déploiement réel.

Ce mémoire se structure autour de trois principales parties : la présentation du système existant et de ses insuffisances, la modélisation de la solution automatisée, puis la programmation et la simulation du système développé. Il met en lumière l'importance de l'ingénierie de l'automatisation dans la résolution des problématiques industrielles actuelles et montre comment les outils numériques permettent de concevoir, tester et valider des solutions robustes et efficaces.

Préambule : Présentation de Cevital

1 Présentation de l'entreprise Cevital :

Cevital est un groupe familial qui s'est construit autour d'une histoire, d'un parcours et des valeurs ayant contribué à son succès et à sa notoriété.

Première entreprise privée algérienne à investir dans des secteurs d'activités variés, elle a franchi d'importantes étapes historiques pour atteindre sa dimension et sa réputation actuelles.

Industrie agroalimentaire et grand distribution, électronique et électro-ménager, sidérurgie Industrie de verre plat, construction industrielle, automobile, services, médias... Le groupe **Cevital** s'est construit, au fil des investissements, autour de l'idée forte de constituer un ensemble économique [1].

2 Historique :

Cevital est l'une des principales entreprises algériennes nées lors de l'ouverture de notre pays à l'économie du marché. Depuis ses débuts, elle joue un rôle crucial dans la dynamique industrielle algérienne, crée par des fonds privés en 1998 à Bejaia.

Le complexe est une société par action (SPA) dont les actionnaires principaux sont M. **REBRAB** et **Fils**, elle est l'un des fleurons de l'industrie agroalimentaire en Algérie qui est constituée de plusieurs unités de production équipées de la dernière technologie et poursuit son développement par divers projets en cours de réalisation. Elle a été créée en Mai 1998 avec un capital social qui est fixe 68,760 milliards de DA. Elle se situe dans le nouveau quai du port de BEJAIA et s'étend sur une superficie de 76156 m² [1].

Cevital agro-industrie regroupe plusieurs unités de production à la pointe de la technologie, parmi lesquelles :

- Deux raffineries de sucre.
- Une unité de sucre liquide.
- Une raffinerie d'huile.
- Une unité pour margarine.
- Une unité de conditionnement d'eau minérale.
- Une unité de fabrication et de conditionnement de boissons rafraichissantes.
- Une unité pour conserverie.
- Une unité de production de chaux calcinée.

L'entreprise possède également des silos portuaires ainsi un terminal de déchargement d'une capacité de 2000 tonnes par heurs, ce qui en fait l'un des terminaux les plus performants de la région méditerranéenne.

3 Situation géographique :

La position stratégique de **Cevital** offre à l'entreprise un véritable atout en matière de logistique et de compétitivité économique.

Le complexe industriel, considéré comme le plus grand complexe privé d'Algérie s'étend sur 45000 m², et se situe dans la zone sud du port délimité par :

- Au nord par l'accès au port.
- Au sud par la route menant au jetée (bougie plage).
- À l'est par la clôture de port.
- À l'ouest par la route RN 9.



Figure1 : Situation géographique de Cevital.

4 Stratégie de développement :

Cevital s'est construit autour de l'ambition et de la vision de son fondateur à bâtir un groupe industriel d'envergure mondiale, très compétitif tourné vers l'exportation et l'international. Le groupe possède des unités de production de taille mondiale, équipées des technologies les plus évoluées. La stratégie du groupe s'appuie sur une forte compétitivité en termes de prix, de qualité de volumes, de logistique, de robotisation et d'intégration verticale des unités de production.

Une place de choix est également accordée à la recherche de développement, à l'innovation et au talent des collaborateurs. Ces atouts compétitifs forment le socle d'une industrie dynamique, exportatrice, créatrice d'emplois et attractive pour la jeunesse algérienne.

Selon Mr. Issad REBRAB, fondateur de Cevital, le succès du groupe repose sur sept points clés :

- Le réinvestissement systématique des gains dans les secteurs porteurs à forte valeur ajoutée.
- La recherche et la mise en œuvre des savoir-faire technologiques les plus évolués.
- L'attention accordée au choix des hommes et des femmes, à leur formation et au transfert des compétences.
- L'esprit d'entreprise.
- Le sens de l'innovation.
- La recherche de l'excellence.
- La fierté et la passion de servir l'économie nationale [2].

5 Présentation de la margarinerie :

Cette unité a été mise en service en novembre 2001. Elle dispose d'une capacité de production de 600 tonnes par jour. Elle fabrique principalement les produits suivants : Matina, Floriale, Elio, margarine de feuilletage, Smen, ainsi que les shortenings.

Le bloc de production est structuré sur deux niveaux, permettent de réaliser l'ensemble du processus de fabrication de la margarine, depuis les matières premières jusqu'aux produits finis.

5.1 Rez-de -chaussées :

A ce niveau, plusieurs étapes clés de processus de fabrication sont mise en œuvre. On y trouve notamment les bacs d'émulsifiants contenant la matière première, composée des ingrédients essentiels à la production de la margarine. Par ailleurs les opérations de la palettisation et de stockage sont réalisées dans une chambre froide, assurant ainsi des conditions optimales de conservation des ingrédients.

5.2 Première étage :

Cette section est réservée au conditionnement de la margarine, effectué à travers plusieurs lignes de production distinctes, permettent une répartition efficace et structurée de produit fini.

5.3 Deuxième étage :

Sur cette zone, se situe la salle de supervision et de contrôle chargée de surveiller attentivement les paramètres de fabrication de la margarine, ainsi que les laboratoires dédiés à garantir sa qualité optimale.

Pour assurer une conservation adéquate de la margarine, des échantillons sont régulièrement prélevés et analysés en laboratoire pour un contrôle de qualité.

6 Activités et missions :

Les principales activités du complexe Cevital concernent la production et la commercialisation des huiles végétales, de margarine et de sucre. Les objectifs visés par l'entreprise sont la satisfaction et la fidélisation de ses clients tout en assurant le bon conditionnement des huiles, des margarines et du sucre à des prix nettement plus compétitifs.

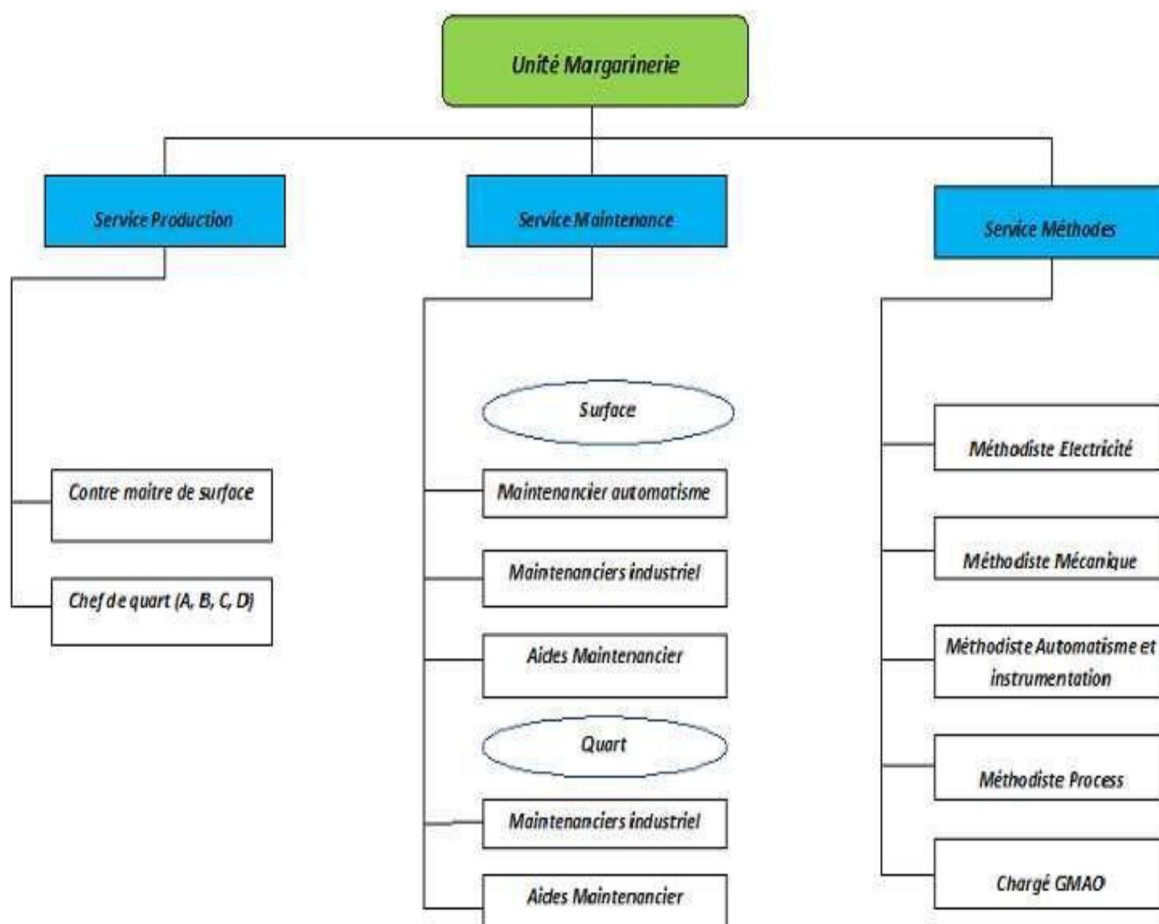


Figure 2 : Unité de la margarine.

Chapitre I :

Présentation du système

I.1 Introduction :

Dans ce chapitre, nous allons décrire le système de convoyage actuel utilisé pour le transport des cartons qui contiennent d'huile végétale de premier étage où se passe le conditionnement de la margarine, vers la zone de stockage qui se trouve au niveau rez-de-chaussée.

Cette description présente les caractéristiques générales du système de convoyage, sa structure, ses composants, ainsi que ses fonctions principales.

I.2 Description fonctionnelle :

Après la sortie des cartons de la machine encartonneuse, ce système prend en charge le déplacement de ces cartons qui contiennent un nombre de boîtes bien déterminé remplies de la graisse végétale de la zone de production qui situe au premier étage vers le rez-de-chaussée où se trouve la zone de stockage qui se trouve dans le niveau inférieur.

A la sortie de la machine encartonneuse, les cartons vont être transportés automatiquement via un convoyeur type convoyeur à bande modulaire plastique droit sur une distance de 4 mètres.

Un vérin est placé à la fin du convoyeur droit pour dévier les cartons vers un autre convoyeur à bande modulaire plastique courbe chargé de déplacer ces cartons vers un autre convoyeur de type rouleau libre incliné de niveau 1 au niveau 0. Des capteurs sont placés pour la détection des cartons et l'entraînement de ces cartons se fait par inertie.

A l'arrivée de ces cartons au niveau 0, un convoyeur va s'occuper de les déplacer vers un convoyeur de type courbe. La détection de la présence d'un carton dans l'extrémité du convoyeur via un capteur va permettre au vérin de les pousser vers un convoyeur de type rouleau motorisé qui transportera ces derniers vers le palettiseur.

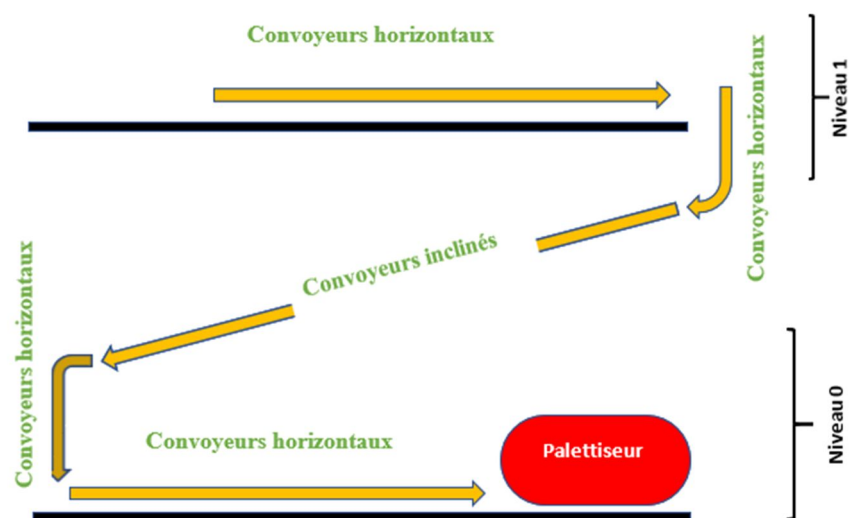


Figure I.1 : Schéma explicatif du système de transport actuel.

I.3 Description technique du système :

Le système de convoyage utilisé pour le transport des cartons entre la zone de production et la zone de stockage est composé d'une structure mécanique basée sur des convoyeurs de différents types, de moteurs, de capteurs et de l'automatisme nécessaire à sa conduite.

I.3.1 Structure mécanique :

I.3.1.1 Les convoyeurs :

I.3.1.1.1 Convoyeur à bande modulaire plastique droit :

Ce convoyeur est placé à la sortie de la machine encartonneuse et est caractérisé par sa surface lisse formée de maillons rectangulaires. Il est utilisé pour le transport des produits alimentaires ou pièces exigeant une surface continue. C'est un système de transport linéaire robuste et hygiénique et il convient au transport de produits variés.



Figure I.2 : Convoyeur de type bande modulaire plastique droit.

I.3.1.1.2 Convoyeur à bande modulaire plastique courbe :

Parmi les caractéristiques de ce convoyeur c'est que sa structure qui assure à la fois un transfert linéaire, changement de direction et accumulation sans bourrage. Pour cela, il est très utilisé dans le secteur agroalimentaire.



Figure I.3 : Convoyeur de type bande modulaire plastique courbe.

I.3.1.1.3 Convoyeur à rouleau libre/motorisé :

Ce type de convoyeurs est fréquent dans les lignes de conditionnement pour des charges rigides comme les boîtes ou cartons. Il permet le transport de cartons par roulement. Le transport des cartons se fait par motorisation ou par inertie.

Ce type de convoyeurs est utilisé dans deux emplacements différents et cela est dû à leurs caractéristiques et aux circonstances sur site.



Figure I.4 : Convoyeur de type à rouleau libre/motorisé.

I.3.1.1.4 Convoyeur à bande motorisée :

Ce type de convoyeur est utilisé pour le transfert de cartons dans une zone sécurisée associée à la précision dans le positionnement grâce à la bande adhérente et ce afin d'éviter que les cartons se glissent dans les arrêts brusques.

Ce type de convoyage est très utilisé pour le transfert précis, l'étiquetage ou le positionnement.

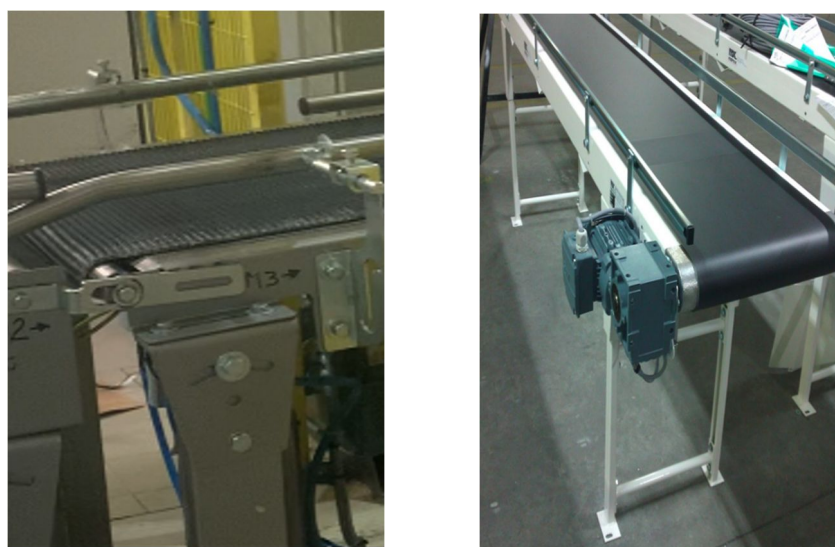


Figure I.5: Convoyeur de type bande motorisé.

I.3.1.2 Les vérins :

Dans ce système, l'utilisation des vérins est nécessaire pour un fonctionnement idéal. Le vérin utilisé est pneumatique à double effet.

Le vérin pneumatique sert à manipuler les boîtes sur la ligne de production. Son rôle est de dévier les boîtes vers un convoyeur secondaire.

Il est raccordé à un système pneumatique, les tuyaux pneumatiques visibles sur la figure alimentent les vérins pneumatiques :

Le tuyau bleu : utilisé pour l'air comprimé en pression.

Le tuyau noir : utilisé comme retour d'air.



Figure I.6 : Vérin de type pneumatique à double effet.

Ce vérin à double effet est contrôlé par une électrovanne pneumatique connectée à un module de distribution électrique et de signaux.

L'envoi d'un signal électrique à l'électrovanne permet d'ouvrir ou de fermer l'arrivée d'air comprimé vers le vérin pneumatique ce qui permet de changer la position de l'actionneur.



Figure I.7 : Électrovanne et distributeur électrique.

Sur la figure I.7, sont montrés l'électrovanne pneumatique et le module de distribution. Nous pouvons voir :

1. **Un module de distribution électrique et de signaux (WEIDMÜLLER)** : destiné à l'alimentation électrique et à transmettre les signaux vers les équipements.
2. **Une électrovanne ou distributeur électropneumatique (FESTO)** : qui permet de contrôler le passage de l'air envoyé vers le vérin.
3. **Un module d'E/S** : connecté à plusieurs câbles et permet de recevoir ou d'envoyer des signaux de commande.
4. **Un câble électrique** : pour la communication et l'alimentations des composants.
5. **Un câble pneumatique** : pour conduire l'air comprimé vers le vérin.

I.3.2 Motorisation :

Dans ce système de convoyage, certains convoyeurs sont motorisés où le transport des cartons est assuré par le mouvement de la bande transporteuse entraînée par un moteur.

I.3.2.1 Moteur :

Le type du moteur associé au convoyeur est un motoréducteur, c'est un ensemble d'un moteur électrique couplé à un réducteur de vitesse. Il permet d'entraîner mécaniquement le convoyeur en réduisant la vitesse de rotation tout en augmentant le couple, assurant ainsi un mouvement fluide et contrôlé de la bande transporteuse.

Le système de convoyage est entraîné par un motoréducteur asynchrone triphasé de type DRS71S4, de la marque SEW-EURODRIVE. Il s'agit d'un moteur asynchrone à cage d'écureuil, souvent utilisé dans les applications industrielles en raison de sa robustesse, de son faible coût et de sa facilité d'entretien.

Ce type de motorisation est particulièrement bien adapté aux systèmes de convoyage, où il assure un mouvement régulier et fiable des charges tout au long de la ligne de production.



Figure I.8 : Moteur asynchrone triphasé.

I.3.2.2 Structure interne du moteur :

La structure interne de moteur asynchrone à cage d'écureuil est composée de :

Stator : Le stator est la partie fixe du moteur. Il est formé de tôles magnétiques empilées comportant des encoches dans lesquelles sont insérés des enroulements en cuivre. En étant alimenté par du courant alternatif triphasé, ces enroulements génèrent un champ magnétique tournant, qui est à l'origine du mouvement du rotor.

Rotor : Il est la partie mobile qui tourne à l'intérieur de stator. Il est composé de barres conductrices en aluminium ou cuivre. Ce rotor est mis à un mouvement par induction électromagnétique, sans connexion direct avec le stator.

Arbre et paliers : L'arbre est solidaire du rotor et transmet le mouvement mécanique à la charge (convoyeur...). Il repose sur des enroulements à billes ou à rouleaux logés dans des paliers, qui permettent une rotation fluide et stable, en minimisant le frottement et l'usure.

Ventilateur : Elle est située à l'arrière du moteur. Le ventilateur est monté sur l'arbre. Cela prévient la surchauffe pendant les longues périodes de fonctionnement.

Borniers électriques : Ce sont les points de connexion électrique du moteur. On y raccorde les fils d'alimentation des trois phases.

Boîtier : Le boîtier protège tous les composants internes contre la poussière, l'humidité et les chocs mécaniques. Il est souvent en fonte ou en aluminium et est muni d'ailettes de dissipation thermique. Il peut aussi être étanche selon les normes d'environnement [3].

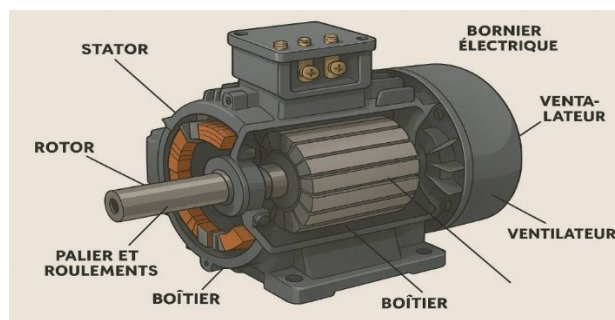


Figure I.9 : Structure interne de moteur asynchrone.

I.3.3 Capteurs et automatisation :

Pour assurer un fonctionnement automatisé, le convoyeur est équipé de capteurs, d'un pupitre IHM et d'un API.

I.3.3.1 Les capteurs :

Dans le domaine de l'automatisation des convoyeurs, la détection précise et fiable des objets est un élément clé pour garantir le bon fonctionnement de systèmes.

Les capteurs jouent un rôle central dans cette tâche, en assurant la présence, la position ou le passage des produits sur les lignes de convoyage.

Les capteurs utilisés dans le système de convoyage sont des capteurs optiques (capteurs photoélectriques à retro-réfléchissantes).

Le rôle de ces capteurs se résume sur la détection de présence d'objets. Il assure un positionnement précis des objets pour d'autres opérations. Il aide à prévenir l'accumulation excessive d'objets sur le convoyeur.

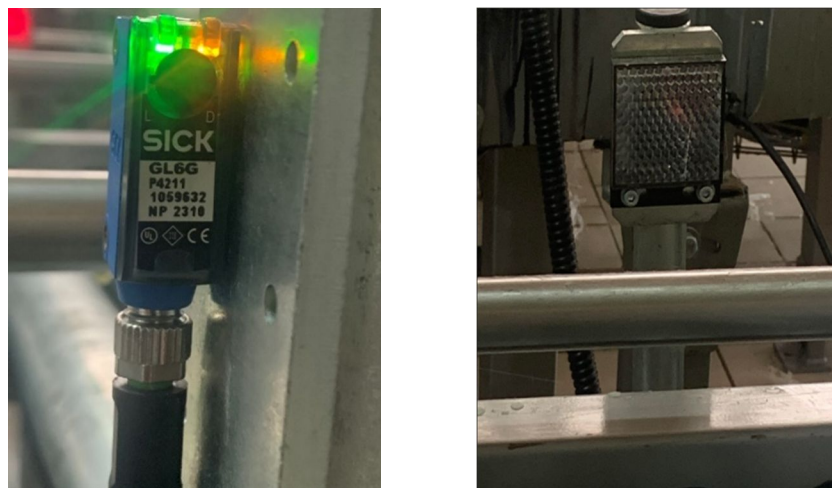


Figure I.10 : Capteur optique et le réflecteur.

I.3.3.2 Pupitre IHM (interface homme-machine) :

C'est un dispositif matériel doté d'un écran qui permet à un opérateur d'interagir avec le système.

Ce type d'interface est indispensable dans les systèmes de convoyage pour assurer la supervision, le pilotage et le diagnostic. Il permet d'afficher l'état du système comme il permet à l'opérateur de démarrer le système ou de l'arrêter, de faire les réglages ou de choisir un mode de fonctionnement, de visualiser les pannes et de saisir des consignes ou des modifications.



Figure I.11 : Pupitre IHM.

Sur la figure I.11, nous pouvons voir :

1. **Un écran tactile IHM** : qui affiche une vue synoptique de la ligne.
2. **Un bouton vert** : pour le démarrage du cycle.
3. **Un bouton bleu** : pour la commande manuelle.
4. **Un voyant** : qui indique le mode automatique actif.
5. **Un bouton d'arrêt d'urgence** : qui permet un arrêt immédiat de la machine en cas de danger.

I.3.3.3 Automate programmable :

L'automate programmable industriel (API) joue un rôle important dans le contrôle du système de convoyage dans le milieu industriel. Il permet d'assurer le pilotage automatique de différentes étapes de transport de cartons sur la ligne de production.

L'automate programmable industriel utilisé dans ce système est de type Siemens SIMATIC S7-1500.



Figure I.12 : L'API S7-1500.

I.4 Problématique :

Après la sortie des cartons de la machine encartonneuse, ceux-ci sont déplacés vers le palettiseur au niveau rez-de-chaussée par inertie.

Ce mode de transfert provoque des collisions entre les cartons, entraînant l'ouverture des boîtes et la perte du produit.

Face à cette anomalie, on était contraint de mettre en place une solution provisoire, c'est d'affecter un opérateur au premier niveau chargé de déplacer manuellement les cartons sur une palette et de les transporter vers la zone de stockage.

Cette solution est considérée comme une tâche pénible pour les opérateurs, vu que de nombreuses réclamations ont été enregistrées.

Face à cette situation, il a été décidé de rechercher à nouveau une solution durable qui permet d'éliminer les problèmes rencontrés avant.

I.5 Conclusion :

Dans ce chapitre, nous avons décrit le système de convoyage qui est destiné au transport des cartons du premier niveau, où se passe le conditionnement, vers la zone de stockage qui se trouve au niveau 0 (rez-de-chaussée), puis nous avons présenté les différents équipements et instruments qui composent ce système.

La connaissance du rôle de ces composants va nous aider à trouver une solution aux problèmes rencontrés.

Chapitre II :
Les automates programmables et
logiciels associés.

II.1 Introduction :

L'automatisation désigne l'utilisation de divers systèmes de contrôle pour le fonctionnement tels que les machines et les processus en usine.

Dans ce chapitre, nous allons présenter les automates programmables et les logiciels associés afin de connaître leurs structures et principe de fonctionnement dans le but de les utiliser dans un contexte plus précis et fiable.

II.2 Définition de l'automatisation :

L'automatisation est la technologie par laquelle un processus ou une procédure est exécutée avec une intervention humaine minimale grâce à des dispositifs technologiques ; électriques et/ou mécaniques.

C'est la technique permettant de faire fonctionner automatiquement un processus ou un système. L'automatisation touche toutes les fonctions de la quasi-totalité des secteurs d'activité : installation, maintenance, fabrication, marketing, vente, médecine, conception, approvisionnement, gestion, etc.

L'automatisation a révolutionné les domaines dans lesquels elle a été introduite, et il n'existe pratiquement aucun aspect de la vie moderne qui n'ait été touché par elle [4].

II.2.1 Objectif de l'automatisation :

L'automatisation des systèmes vise à optimiser les processus en réduisant l'intervention humaine, ce qui permet d'atteindre plusieurs objectifs tels que :

- Amélioration de l'efficacité opérationnelle.
- Réduction des coûts.
- Amélioration de la qualité des services.
- Flexibilité et évolutivité.
- Simplification des processus.
- Meilleure gestion des risques.

II.3 Système automatisé :

II.3.1 Définition :

Un système automatisé est un dispositif technique conçu pour accomplir des tâches de manière autonome, sans nécessité d'intervention humaine direct. Il fonctionne selon un cycle de travail prédéfini, exécutant les mêmes opérations de manière répétitive et fiable.

II.3.2 Structure d'un système automatisé :

Un système automatisé est constitué de plusieurs parties plus ou moins complexes reliées entre elles que l'on nomme :

- Partie opérative (PO).
- Partie commande (PC).
- Partie interface homme-machine (IHM).



Figure II.1 : Structure d'un système automatisé.

II.3.2.1 La partie opérative (PO) :

C'est la partie visible du système, elle regroupe les éléments physiques qui réalisent les actions sur la matière ou l'environnement, tels que les actionneurs et les capteurs.



Figure II.2 : Représentation de la partie opérative [5].

La partie opérative est constituée de deux sous-ensembles de constituants :

- **Chaîne d'action :** Elle désigne l'ensemble des processus et des composants qui transforment une entrée (donnée) en une sortie effective, c'est-à-dire une action physique réalisée sur la matière d'œuvre.
- **Chaîne d'acquisition :** Elle est l'ensemble des étapes et processus permettant de formuler, valider, traiter et exécuter une demande.

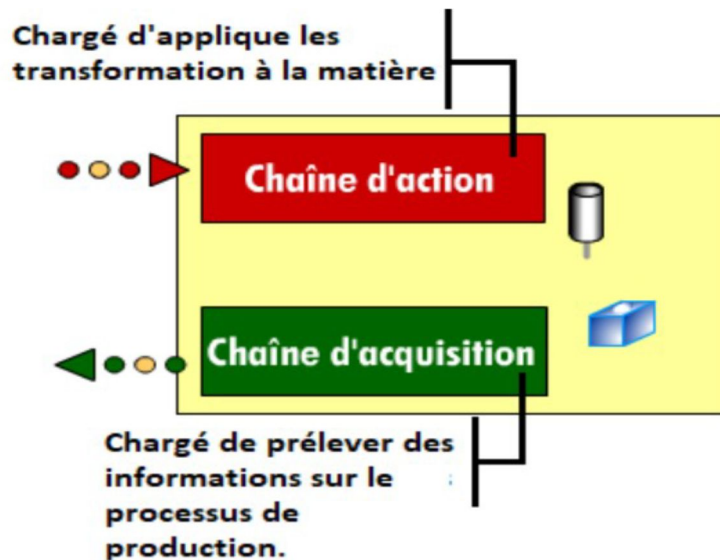


Figure II.3 : Structure de la partie opérative [5].

II.3.2.2 La partie commande (PC) :

C'est le « cerveau » du système. Elle reçoit les informations des capteurs, les compare aux consignes définies, et envoie des ordres aux actionneurs pour réaliser les actions appropriées. Cette partie est souvent constituée d'un automate programmable industriel (API), qui exécute un programme informatique préalablement défini.



Figure II.4 : Présentation de la partie commande [5].

II.3.2.3 L'interface homme-machine (IHM) :

Une interface homme machine, communément appelée IHM ou HMI, est une interface utilisateur ou un tableau de bord. Elle facilite la communication et l'interaction entre un opérateur humain et une machine ou un système. L'IHM permet aux utilisateurs de surveiller, de contrôler et de manipuler divers aspects d'une machine, d'un processus ou d'un système au moyen de représentations graphiques, de boutons, d'interrupteurs et d'autres dispositifs d'entrées/sorties [6].



Figure II.5 : Interface homme machine IHM.

II.3.2.3.1 Rôle de pupitre IHM :

Le rôle d'interface homme machine se résume en :

- Afficher l'état du système.
- Commander les actions, démarrage/arrêt et modification des consignes.
- Assure le dialogue entre l'opérateur et l'automate (API).
- Enregistrer les données.

II.3.2.3.2 Communication avec l'automate :

La communication entre l'API et IHM se fait par des protocoles tels :

- PROFINRT/PROFIBUS
- Modbus RTU/TCP
- Ethernet/IP
- CANopen

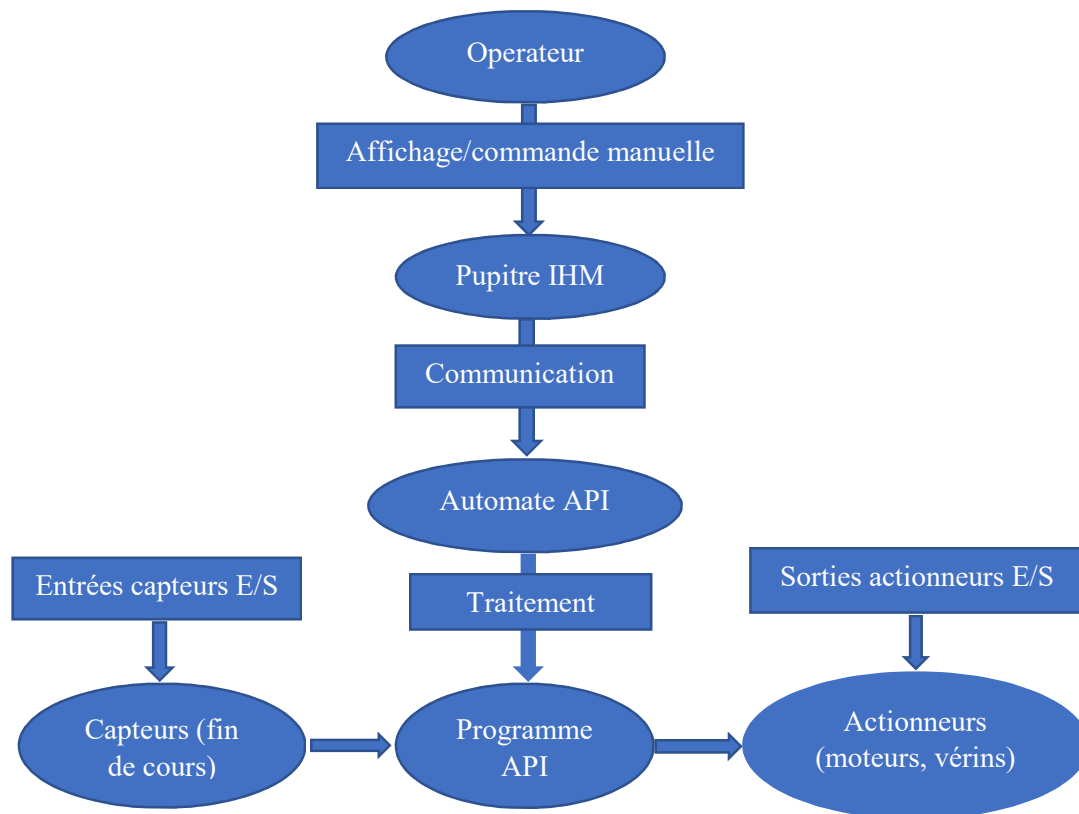


Figure II.6: schéma fonctionnel d'un pupitre IHM d'un système automatisé.

II.3.3 Fonctionnement d'un système automatisé :

Le fonctionnement du système automatisé repose sur :

- **Acquisitions des données :** les capteurs mesurent les paramètres du processus et envoient ces informations à la partie commande.
- **Traitement des informations :** L'API analyse les données reçues en fonction du programme et des consignes définies.
- **Prise de décision :** en fonction de résultats du traitement, L'API détermine les actions à entreprendre.
- **Exécution des actions :** des ordres sont envoyés aux actionneurs pour réaliser les actions nécessaires.

II.4 Automate programmable industriel :

II.4.1 Introduction :

Les automates programmables industriels (API) sont des dispositifs essentiels dans l'automatisation des procédés industriels. Ils permettent de contrôler automatiquement des machines et des systèmes grâce à des programmes configurés selon les besoins. Ces automates sont pilotés via des logiciels spécialisés, tels que TIA Portal ou Unity pro, qui facilitent leur programmation, leur diagnostic et leur supervision. Ensemble, automate et logiciel assure un contrôle fiable, flexible et efficace des installations industrielles.

II.4.2 Définition :

Un automate programmable industriel (API) est une machine programmable qui effectue des tâches répétitives à grande vitesse et avec précision. Les API sont couramment utilisés dans les industries manufacturières pour la production des composants dans différents domaines, ainsi que pour les tests de qualité et de fiabilité sur les produits finis [7].

II.4.3 Les applications de l'automate programmable industriel :

Les automates programmables industriels (API) sont utilisés dans divers industries pour effectuer ne variété de tâches.

- **Automatisation des processus de fabrication :** Les API améliorent le contrôle des processus de fabrication en suivant des instructions programmées et en surveillant les systèmes. Ils fournissent également des informations supplémentaires sur l'état du système et offrent des possibilités de surveillance et de diagnostic à distance.
- **Automatisation des processus commerciaux :** Les API automatisent et améliorent le traitement des commandes, la gestion des stocks et l'intégration des bases des données.
- **Contrôle de la qualité :** Les API automatisent les processus de contrôle de la qualité et surveille le niveau de qualité des produits pour garantir leur conformité aux normes et aux spécifications.
- **Contrôle des systèmes :** Les API facilitent le contrôle des systèmes en fournissant des commandes programmées pour automatiser les tâches, surveiller les performances du système et faciliter la maintenance.

II.4.4 Les types des automates programmables industriel :**II.4.4.1 Les automates modulaires :**

Les automates modulaires sont organisés selon l'architecture suivante :

- Un module d'unité centrale ou CPU, qui assure le traitement de l'information et la gestion de l'ensemble des unités. Ce module comporte un microprocesseur, des circuits périphériques de gestion des entrées /sorties, des mémoires RAM et EEPROM nécessaires pour stocker les programmes, les données, et les paramètres de configuration du système.
- Un module d'alimentation, généralement alimenté en 220 V / 50 Hz, qui fournit l'énergie nécessaire au fonctionnement de l'unité centrale (CPU) et des autres modules de l'automate. Dans certains cas, ce module peut également être alimenté en 24 V DC, selon les spécificités de l'installation.
- Un ou plusieurs modules d'entrées « tout ou rien » ou analogiques pour l'acquisition des informations provenant de la partie opérative.
- Un ou plusieurs modules de sorties « tout ou rien" (TOR) ou analogiques pour transmettre les signaux de commande à la partie opérative. Il y a des modules qui intègrent en même temps des entrées et des sorties [8].

II.4.4.2 Les automates compacts :

De type compact, on distinguera les modules de programmation (LOGO de Siemens, ZELIO de Schneider, MILLIENIUM de Crouzet...) des micro automate. Il intègre le processeur, l'alimentation, les entrées et les sorties. Selon les modèles et les fabricants, il pourra réaliser certaines fonctions supplémentaires (comptage rapide, ES analogiques...) et recevoir des extensions en nombre limité. Ces automates, de fonctionnement simple, sont généralement destinés à la commande de petits automatismes [9].

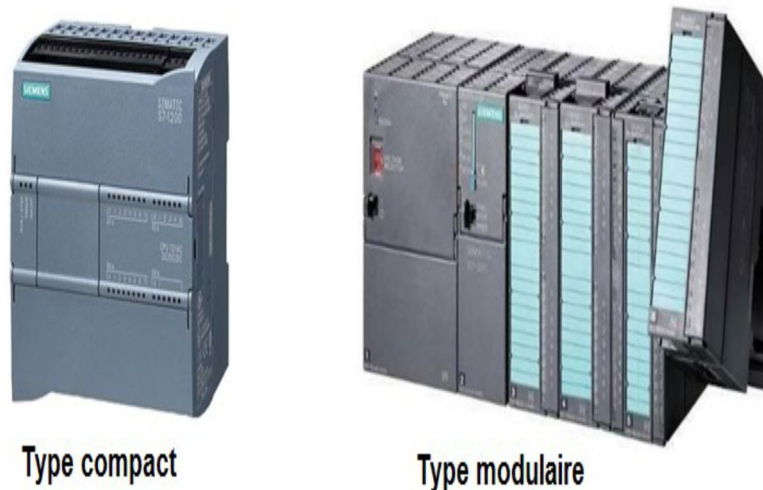


Figure II.7 : Types d'automates programmables [5].

II.4.5 Structure d'un automate programmable industriel API :

Les API sont conçus pour être robustes et modulaires, adaptés aux environnements industriels. Ils se composent généralement des éléments suivants :

II.4.5.1 Unité centrale de traitement (CPU) :

C'est la partie essentielle de l'automate, responsable de l'exécution du programme et du contrôle de différentes fonctions.

II.4.5.2 Mémoire :

- **Mémoire programme :** Elle stocke le programme d'automatisation, généralement en ROM.
- **Mémoire de données :** Elle est utilisée pour les données temporaires et les variables, souvent en RAM.

II.4.5.3 Modules d'entrées/sorties (E/S).

Ces modules permettent la communication avec le monde extérieur, en recevant des signaux des capteurs (entrées) et en envoyant des commandes aux actionneurs (sorties).

II.4.5.4 Alimentation électrique :

Elle fournit l'énergie nécessaire au fonctionnement de l'automate.

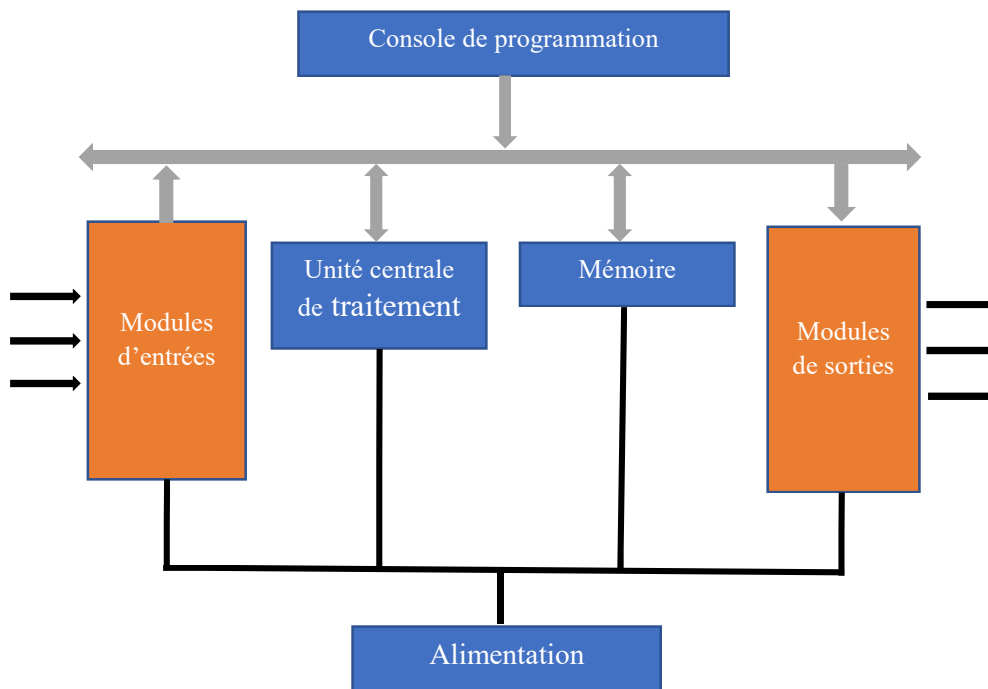


Figure II.8 : Structure interne d'un automate programmable.

II.4.6 Nature des informations traitées par l'automate programmable :

Un automate programmable industriel (API) traite différents types d'informations provenant de son environnement. Ces informations peuvent être classées en trois catégories principales :

II.4.6.1 Tout ou rien (TOR) –signaux binaires :

Les informations de type TOR sont des signaux discrets qui ne peuvent prendre que deux états possibles :

0 : absence d'événement ou état bas.

1 : présence d'événement ou état haut.

Ces signaux sont généralement fournis par des capteurs tels que des détecteurs de présence, des boutons poussoirs ou des interrupteurs.

II.4.6.2 Analogique- signaux continus :

Les informations analogiques sont des signaux continus qui peuvent prendre une infinité de valeurs dans une plage déterminée. Ces signaux sont souvent générés par des capteurs mesurant des grandeurs physiques telles que :

- Température.
- Pression.
- Niveau de liquide.
- Vitesse.

Cependant, pour être traités par l'API, ces signaux doivent être convertis en valeurs numériques à l'aide d'un convertisseur analogique-numérique (CAN). Les plages courantes de ces signaux sont 0-5 V, 0-10 V ou 4-20 mA.

II.4.6.3 Numérique-signaux codés :

Les informations numériques sont des données codées sous forme binaires ou hexadécimale. Ces données peuvent représenter des valeurs discrètes ou des séries de données complexes.

Elles sont utilisées pour des communications entre automates, interface homme-machine (IHM) ou système de supervision.

II.4.7 Avantage des automates programmables :

Les automates programmables industriels (API) sont au cœur de l'automatisation moderne, offrant différents avantages pour les processus industriels :

- **Flexibilité et adaptabilité :** Les API sont faciles à reprogrammer pour l'adapter à des nouvelles configurations
- **Fiabilité accrue :** Les API sont robustes et assurent une disponibilité élevée des installations.
- **Reduction des coûts :** L'utilisation des API conduit à diminuer les bousions en main-d'œuvre et la consommation d'énergie.
- **Amélioration de la qualité :** Avec les API, une qualité uniforme des produits est assurée.
- **Intégration et interconnexion :** Les API peuvent être intégrés dans des systèmes plus vastes et peuvent être connectés à un réseau local industriel pour échanger de l'information.
- **Maintenance facilitée :** Les API sont conçus pour une maintenance facile et rapide.
- **Evolutivité :** Les API peuvent être étendus en ajoutant des modules ou des cartes de communication.

II.4.8 Langage de programmation pour API :

Les programmes utilisés avec les API peuvent être écrits dans différents formats.

Pour que les ingénieurs ayant peu de connaissances en programmation puissent élaborer des programmes pour les API, le langage à contacts a été conçu. La plupart des fabricants d'automates ont adopté cette méthode d'écriture des programmes.

Toutefois, puisqu'on a eu tendance à développer ses propres versions, une norme internationale a été établie pour le langage à contacts et par voie de conséquence, pour toutes les méthodes de programmation employées avec les API [4].

La norme IEC 1131-3 définit cinq langages qui peuvent être utilisés pour la programmation des automates programmables industriels. Ces cinq langages sont :

II.4.8.1 Ladder (LD- Ladder Diagram/schéma à contacts) :

C'est un langage graphique inspiré des schémas électriques. Il est très utilisé en automatisme, particulièrement en maintenance et il est accessible pour les électriciens.

II.4.8.2 FBD (Function Block Diagram/Diagramme de blocs fonctionnels) :

C'est un langage graphique basé sur des blocs logiques. Il est approprié pour des traitements analogiques ou des chaînes de traitements continues.

II.4.8.3 ST (Structured Text/Text structure):

C'est un langage textuel proche du langage de programmation Pascal. Il est puissant pour les traitements complexes, les boucles, les calculs mathématiques.

II.4.8.4 IL (Instruction List/liste d'instruction) :

C'est un langage textuel bas niveau proche de l'assembleur.

II.4.8.5 SFC (Sequential Function Chart/Grafset):

C'est un langage graphique utilisé pour représenter des séquences (états, transitions). Il est idéal pour le contrôle de processus séquentiels.

Chez Cevital, la majorité des automates programmables industriels (API) utilisés appartient à la gamme SIMATIC S7 de SIMENS. On retrouve : S7-200, S7-300, S7-400, S7-150.

II.4.9 Critères de choix d'un automate programmable :

Le choix d'un automate programmable industriel doit être basé sur une analyse approfondie des besoins spécifiques de votre application, en tenant compte des critères tels que :

- **Critères techniques** : entrées/sorties, temps de cycle, mémoire et capacité de traitement, langages de programmation, fonction de communication.
- **Critères pratiques** : évolutivité, robustesse, maintenance et support.
- **Critères économiques** : coût total, renommée du fabricant.

Une sélection judicieuse garantira la performance, la fiabilité et la pérennité du système automatisé.

II.4.10 Présentation de l'automate utilisé S7-1500 :

Le S7-1500 est un automate programmable industriels de Siemens destiné aux applications d'automatisation moyennes ou complexes, avec des performances supérieures, une meilleure connectivité et des fonctions avancées.



Figure II.9 : API Siemens SIMATIC S7-1500.

II.4.10.1 Caractéristique d'API siemens SIMATIC S7-1500 :

Les principales caractéristiques de l'automate Siemens SIMATIC S7-1500 sont :

- C'est un système de commande modulaire conçu pour une multitude d'applications dans l'automatisation discrète, offrant des performances et une convivialité maximale pour les applications en milieu et haut de gamme.
- Il dispose d'une vaste gamme de modules combinables individuellement, offrant une grande flexibilité d'utilisation.
- La structure modulaire permet d'utiliser seulement les modules nécessaires à l'application, avec la possibilité d'extensions par l'ajout de modules supplémentaires selon les besoins.
- Le SIMATIC S7-1500 est adapté à une large gamme d'applications, notamment dans les secteurs de l'automatisation, des machines, des installations, de la construction mécanique, de l'industrie électronique et automobile.

II.5 TIA Portal (Totally Integrated Automation):

II.5.1 Définition :

Le TIA Portal est un environnement d'ingénierie intégré de Siemens qui permet la conception, la programmation et la mise en service d'automatisations industrielles. Il offre une plateforme unique pour gérer tous les aspects de l'automatisation, y compris les API (PLC), les IHM, les entraînements et les systèmes de sécurité. Le TIA Portal simplifie le processus d'ingénierie et améliore l'efficacité opérationnelle [10].

II.5.2 Avantage de logiciel TIA Portal :

Le TIA Portal de Siemens offre des avantages tels que l'intégration transparente des contrôleurs, E/S distribuées, IHM, variateurs, contrôle de mouvement et gestion des moteurs dans un seul environnement d'ingénierie. Il permet une réduction du temps d'ingénierie jusqu'à 30%, une gestion de données commune, un concept de bibliothèque intelligente et des fonctions logicielles et matérielles complètes pour résoudre efficacement les tâches d'automatisation [10].

II.5.3 Vue de portail et vue de projet :

Lorsqu'on lance TIA Portal, l'environnement de travail se présente sous deux types de vues principales, chacun a un rôle spécifique dans la gestion et le développement des projets d'automatisation : vue de portail et vue de projet.

II.5.4 La vue de portail :

C'est la première interface qui s'affiche au démarrage de TIA Portal. Elle est conçue pour être orientée vers les tâches.

Son objectif principal est de faciliter l'accès rapide aux fonctions essentielles du logiciel, notamment pour les débutants ou lors des premières étapes d'un projet. Cette vue est donc centrée sur la gestion générale du projet avant d'entrer dans les détails techniques de programmation ou de configuration.

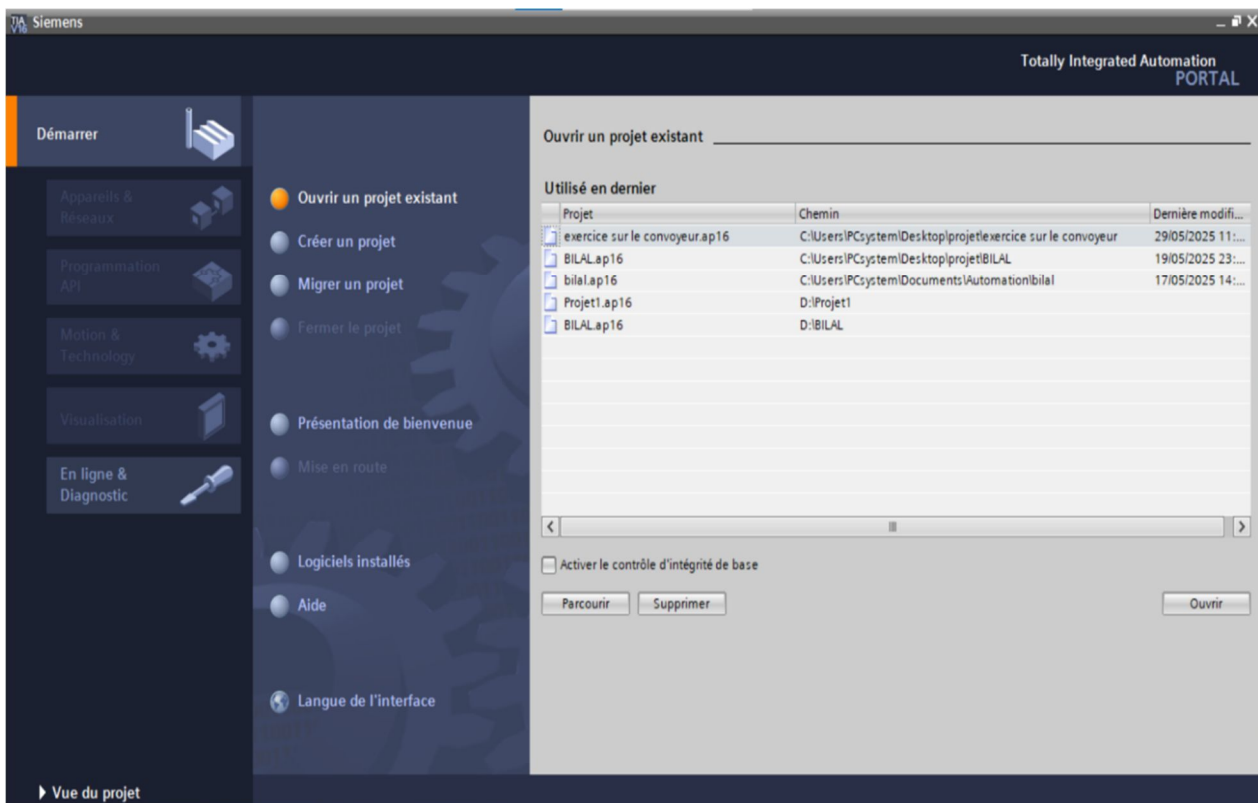


Figure II.10 : Vue de portail.

II.5.5 La vue de projet :

La vue de projet est l'environnement de travail principal pour le développement d'un projet. Elle s'ouvre dès qu'un projet est créé ou chargé à partir de la vue de portail.

Elle permet un accès rapide à tous les objets du projet, aux zones de travail correspondants et aux éditeurs. La figure suivante nous montre la structure de la vue du projet qui compose de navigateur de projet. Cela nous permet un accès à tous les composants et données du projet, et de la zone de travail. Il nous permet de visualiser les objets sélectionnés dans le projet pour être traités. Une fenêtre d'inspection affiche les informations complémentaires sur un objet sélectionné ou sur les actions en cours d'exécution.

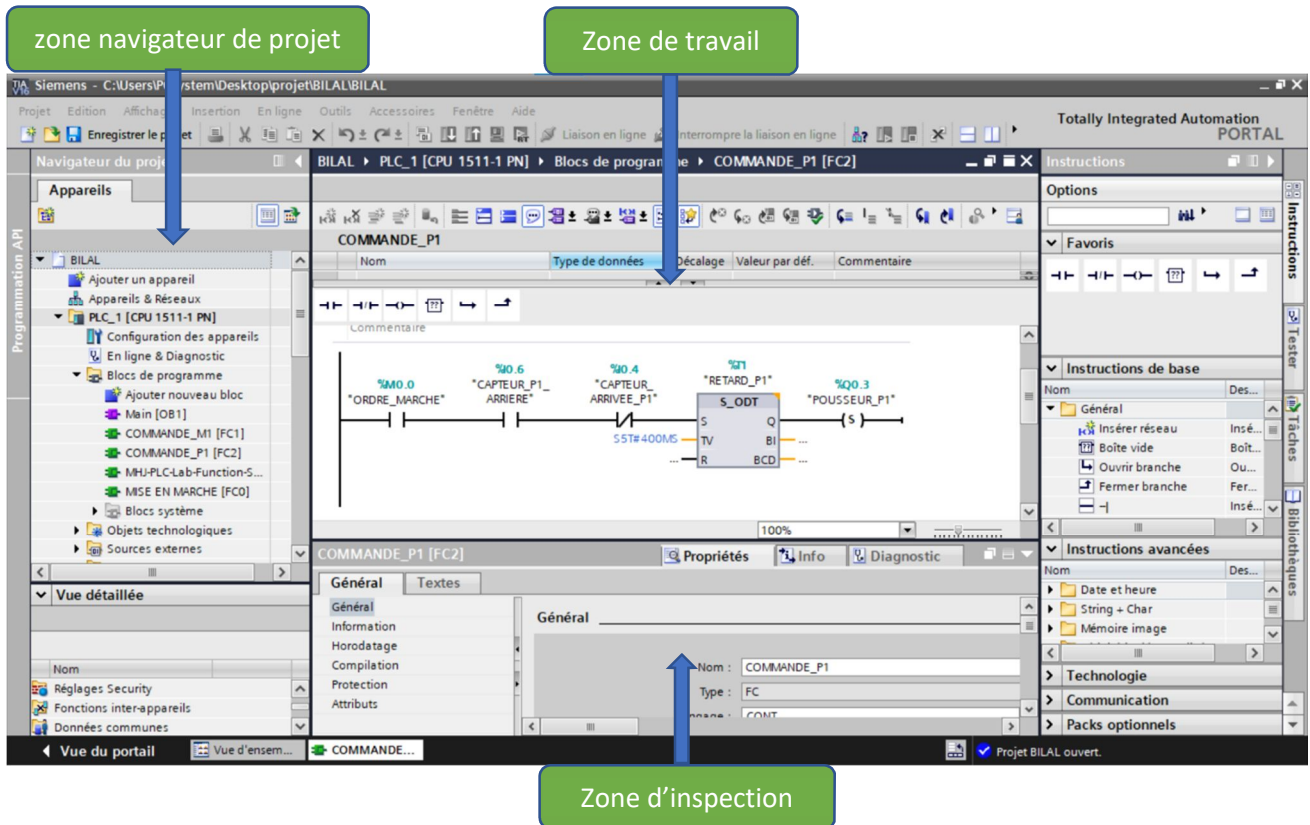


Figure II.11 : Vue de projet

II.5.6 Création d'un projet et configuration d'une station de travail :

Pour créer un projet, il faut d'abord sélectionner l'action « créer un projet » sur la vue de portail et après avoir entrées les informations, il suffit de cliquer sur le bouton « créer ». La figure suivante (figure II.12) représente la création d'un projet.

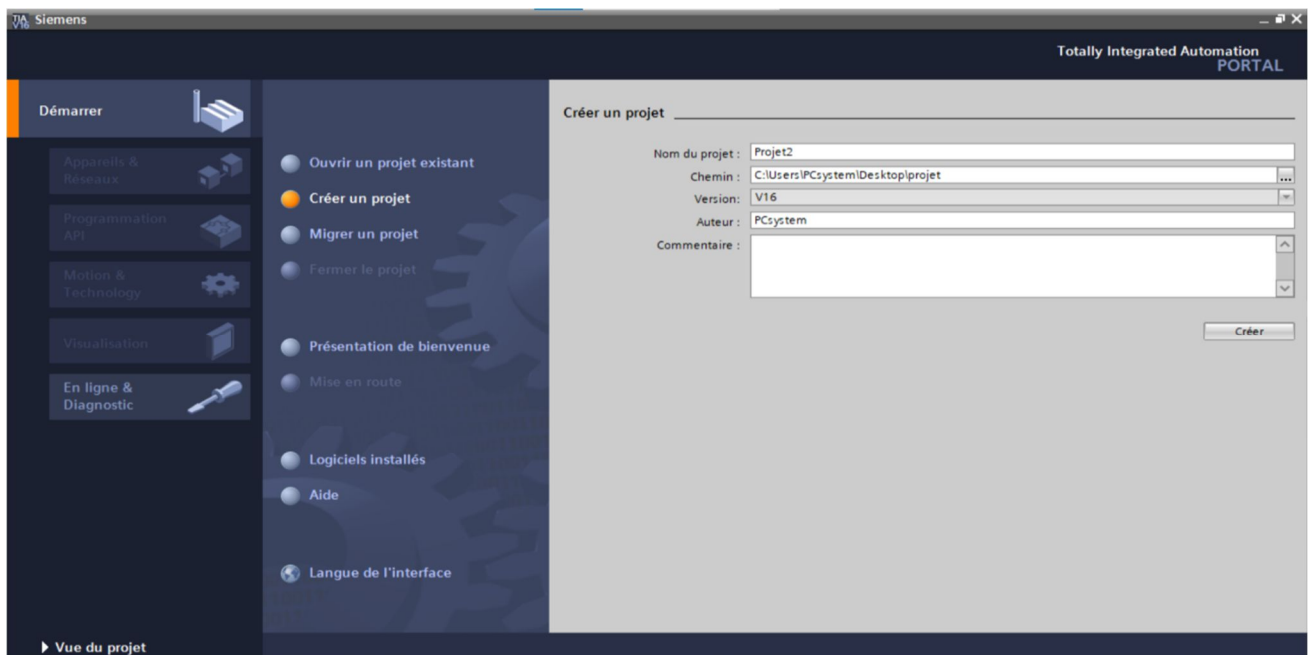


Figure II.12 : Création de projet.

II.5.7 Configuration et paramétrage de matériel :

On entame l'étape de la configuration après la création du projet.

Sur la vue de projet, on clique sur « ajouter un appareil » et cette action sur la fenêtre de navigateur de projet nous montre la liste des éléments, que on peut ajouter, apparaître (API, IHM, système PC). Le choix de la CPU permet ensuite d'ajouter les modules complémentaires (alimentation, E/S TOR ou analogiques, module de communication...etc.). La figure II.13 représente la configuration et le paramétrage de matériel.

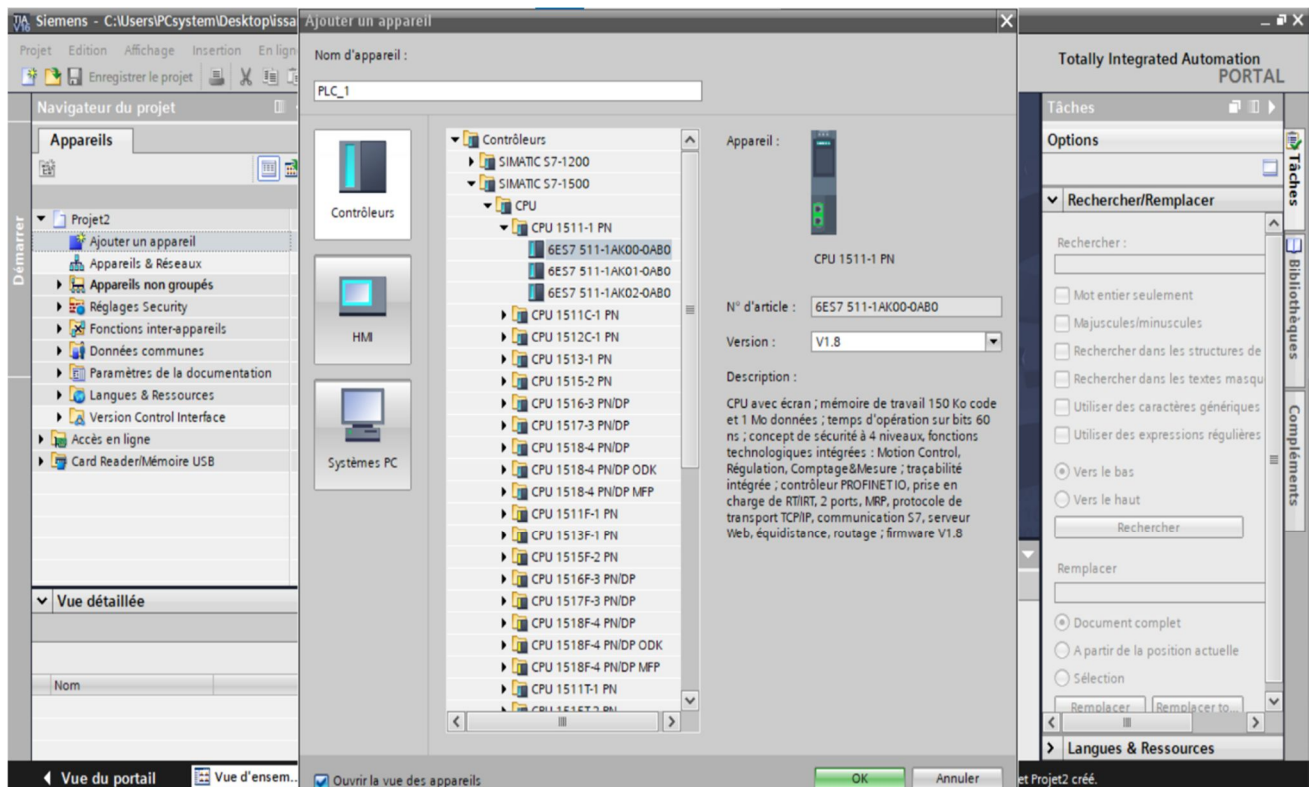


Figure II.13 : Configuration et paramétrage de matériel.

Les modules complémentaires de l'automate (API) peuvent être ajoutés à partir du catalogue. Si on souhaite intégrer un écran IHM ou un autre automate, il faut repasser par la commande « ajouter un appareil » disponible dans la fenêtre navigateur du projet.

Une description détaillée s'affiche dans l'onglet information lorsqu'un élément est sélectionné pour être inséré dans le projet, ce qui facilite son identification et sa configuration.

La figure II.14 représente une deuxième vue de la configuration et du paramétrage matériel.

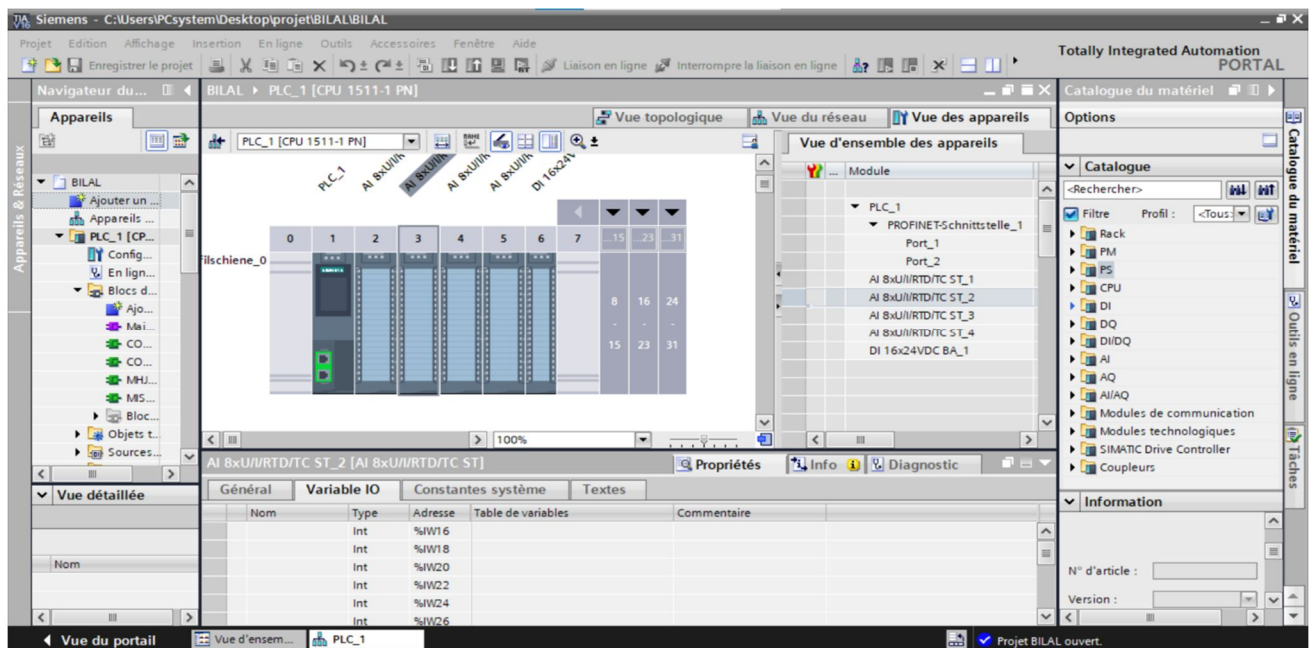


Figure II.14 : Configuration et paramétrage de matériel.

II.5.8 Adressage des E/S :

Pour consulter l'adressage des entrées et des sorties dans la configuration matérielle, il faut accéder à la section « appareil et réseau » par le navigateur du projet. Il est important de s'assurer que l'on se trouve dans l'onglet vue des appareils et de sélectionner l'appareil concerné.

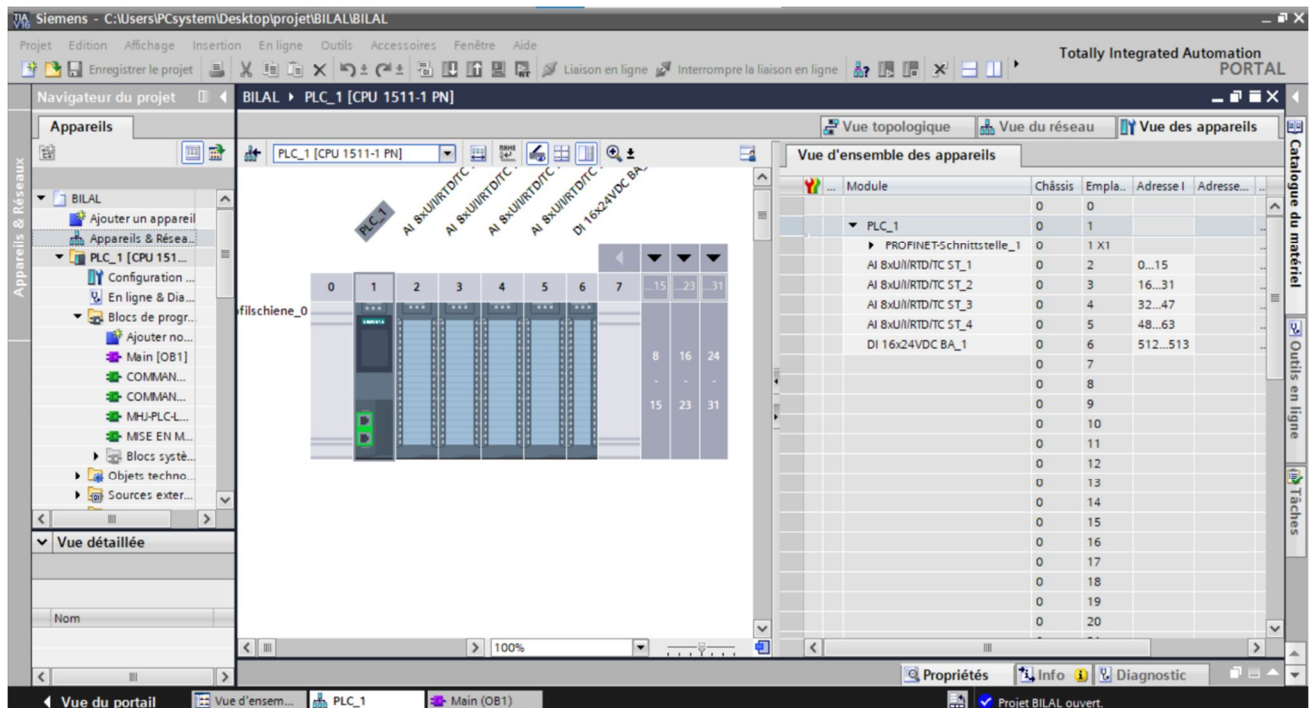


Figure II.15 : Adressage des E/S.

Pour modifier les adresses d'entrées/sorties, on sélectionne la CPU et avec l'aide des deux flèches on fait apparaître l'onglet vue d'ensemble des appareils.

II.5.9 Adresse Ethernet de la CPU :

Pour configurer une adresse Ethernet, un double clic sur le connecteur Ethernet de la station ouvre la fenêtre d'inscription, sur laquelle on peut définir les paramètres réseau de l'automate dans les propriétés de la CPU.

Il est nécessaire que les deux appareils appartiennent au même réseau IP pour faire une liaison entre la CPU et la console de programmation.

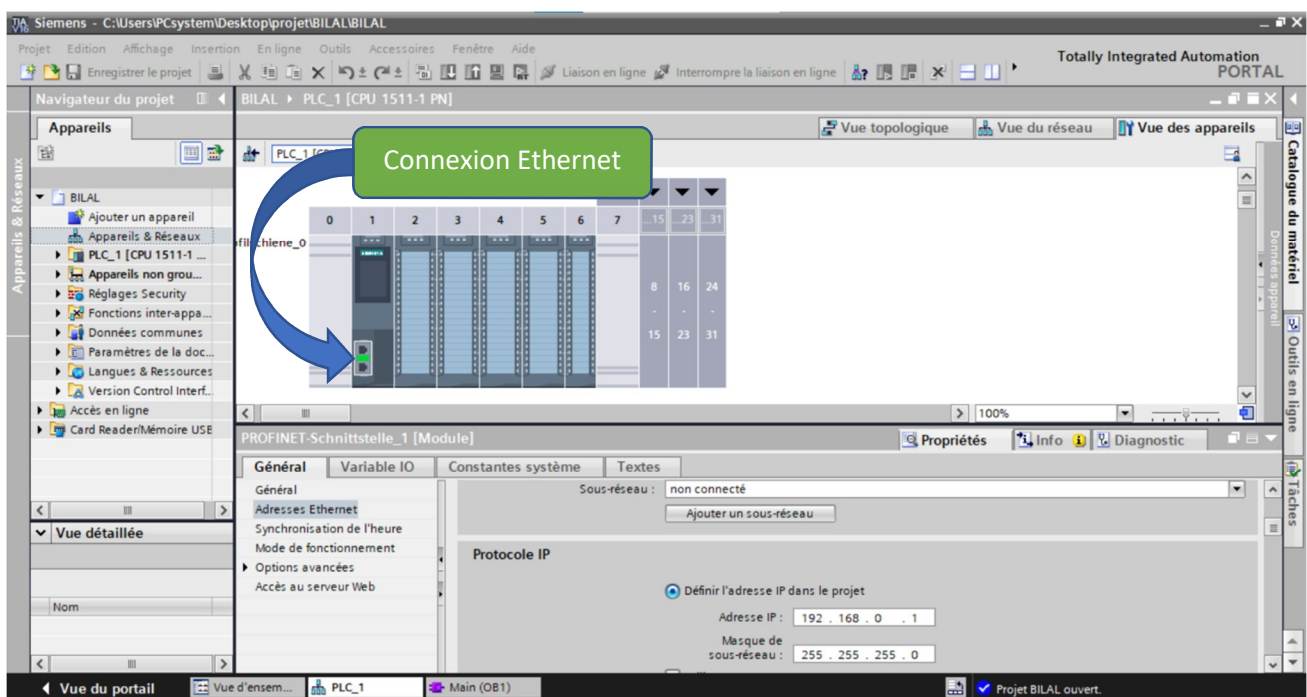


Figure II.16: Adresse Ethernet du CPU.

II.5.10 Compilation et chargement de la configuration matérielle :

Dans un projet de programmation via TIA Portal la compilation est une étape indispensable avant le téléchargement de programme.

Pour assurer que le projet ne contient pas d'erreurs de syntaxe ou de configuration après la configuration matérielle réalisé, il faut le compiler. Cette tâche nous permet de vérifier la syntaxe des blocs du programme et de contrôler la configuration matérielle.

La compilation se fait en sélectionnant l'API dans le navigateur de projet, puis on clique sur l'icône « compiler » située dans la barre des tâches en haut de l'interface, ce que nous permet de faire une compilation matérielle et logicielle.

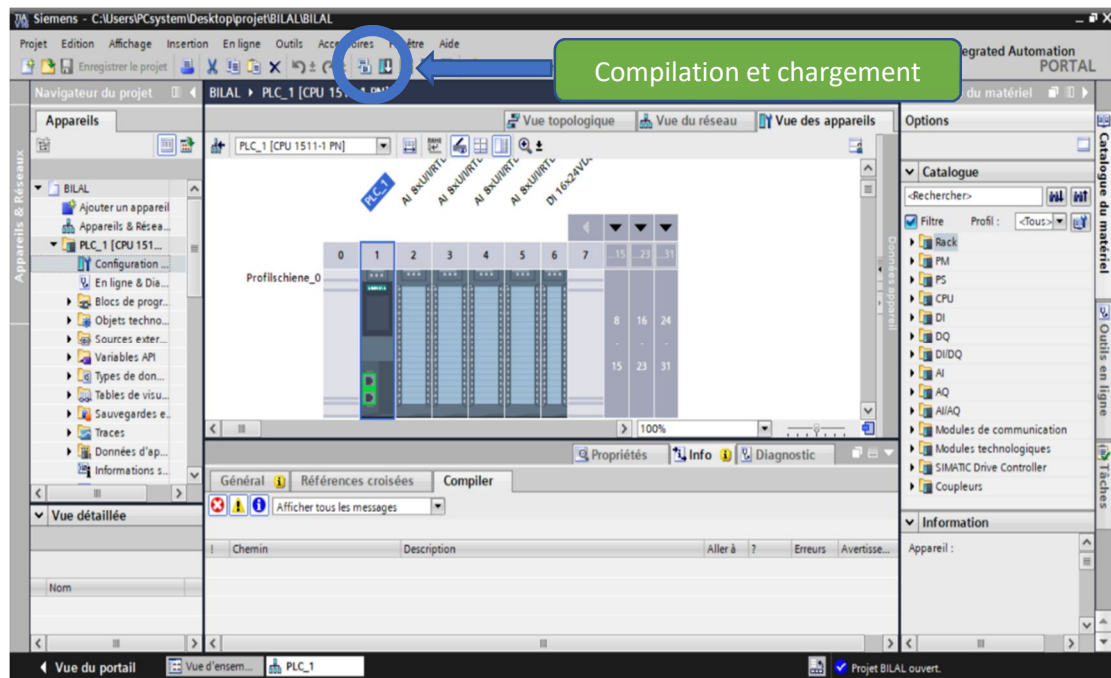


Figure II.17 : Compilation et chargement.

Après avoir compilé le projet dans TIA Portal et corrigé toute les erreurs éventuels, l'étape suivante consiste à charger la configuration dans l'automate. Pour cela on effectue un clic sur l'icône « charger dans l'appareil » et une fenêtre s'ouvre. Sur celle-ci on doit faire le choix du mode de connexion (PN/IE, Profibus, MPI). Si notre choix est sur PN/IE, l'API doit avoir une adresse IP.

Remarque : dans notre cas l'API est virtuel

Dans TIA Portal on peut cliquer sur démarrer la simulation pour l'ouvrir automatiquement.

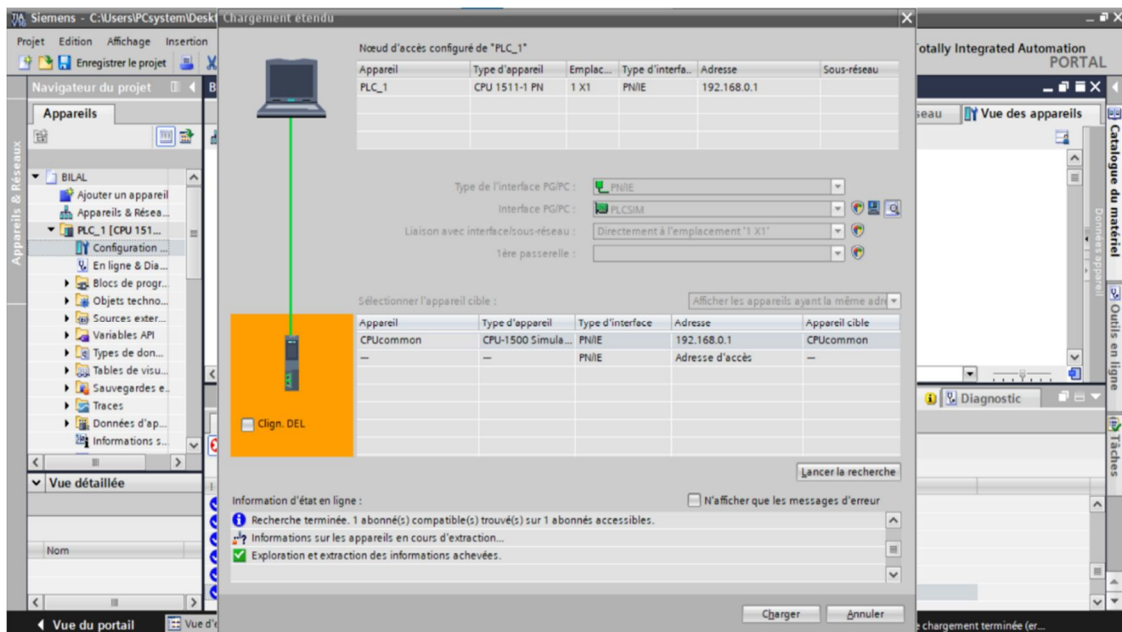


Figure II.18 : Chargement de programme dans un automate virtuel.

La figure II.18 montre que le programme est en point de se charger dans le PLC simulé, le projet est prêt et la compilation est correcte.

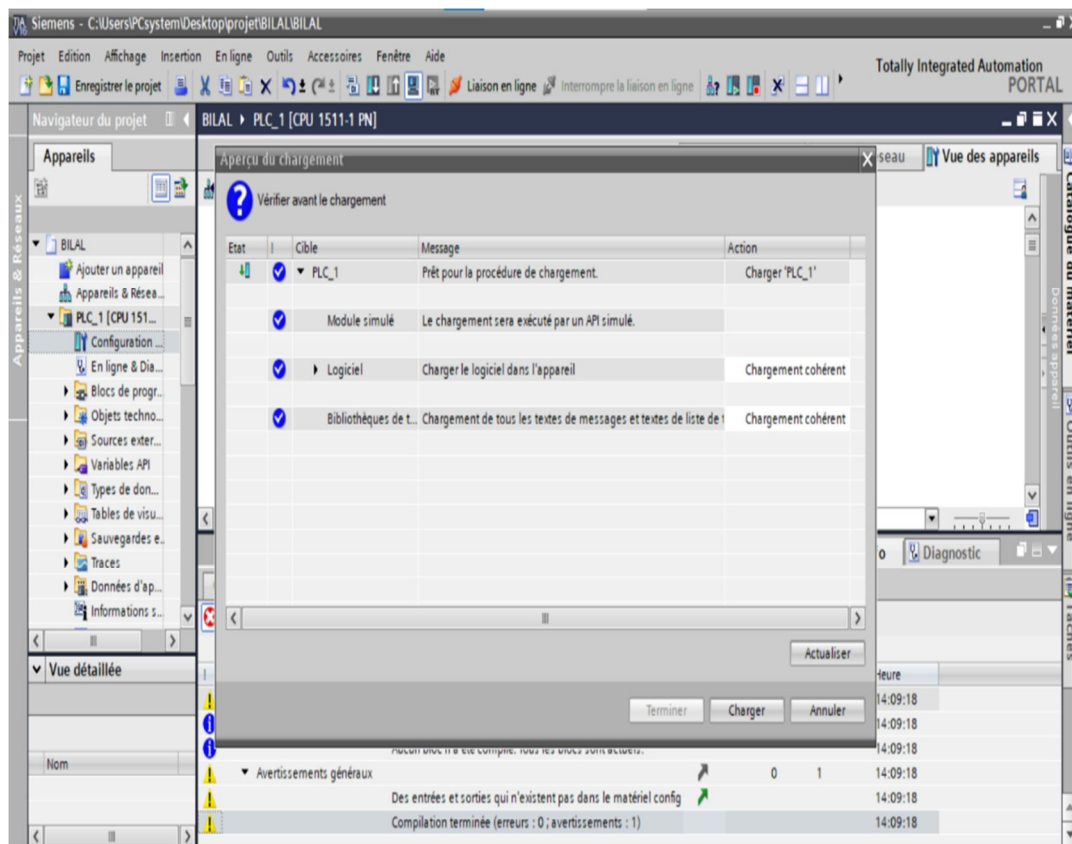


Figure II.19 : Chargement de programme dans un automate virtuel.

la figure II.19 représente un projet prêt à être chargé dans un automate simulé.

II.5.11 Présentation des blocs de programmation :

Dans TIA Portal, un ensemble des blocs de programmation est disponible pour l'application pour automate Siemens tel que S7-1200 ou S7-1500. Ces blocs permettent une gestion et un accès efficaces aux données au sein du système PLC. Ces composants sont fondamentaux dans l'organisation et l'exécution des programmes dont la structure est modulaire, claire et réutilisable.

Dans la programmation de l'automate Siemens, la structure du programme est séparée en 4 types de blocs dérivés : bloc d'organisation (OB), bloc fonctionnel (FB), bloc fonction (FC) et bloc de données (DB).

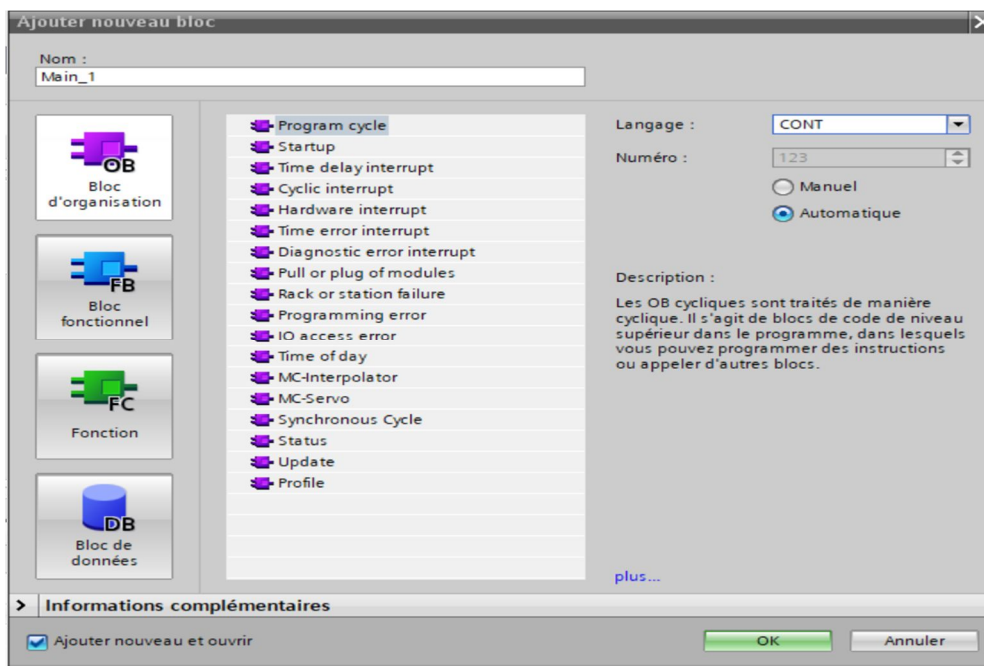


Figure II.20 : Blocs de programmation.

II.5.11.1 Bloc d'organisation (OB) (organisation block) :

Il constitue l'interface entre le système d'exploitation et le programme utilisateur. Ces blocs d'organisation (OB) définissent la structure du programme utilisateur.

Il sert à organiser et à contrôler l'exécution du programme dans le PLC. Le plus utilisé est l'OB1, qui agit comme le programme principale, s'exécutant d'une manière cyclique pendant tout le fonctionnement de l'automate. C'est dans l'OB1 que l'on appelle les autres blocs de fonction (FB, FC), ce qui permet de structurer le programme.

II.5.11.2 Blocs de fonction (FB) :

Les blocs fonctionnels sont des blocs avec mémoires qui mémorisent leurs paramètres d'entrée/sortie dans des blocs de données d'instance pour qu'il soit possible d'y accéder.

II.5.11.3 Fonctions (FC) :

Les fonctions sont des blocs de code sans mémoires. Les données des variables temporaires sont perdues dès que l'exécution de la fonction se termine. Pour conserver ces données, il est nécessaire d'utiliser des opérandes globaux. Ces blocs sont principalement utilisés pour programmer des fonctions réutilisables à plusieurs endroits dans le programme.

II.5.11.4 Blocs de données (DB):

un bloc de données (DB) est un type de bloc destiné au stockage des variables, notamment celles qui doivent être partagées entre plusieurs blocs de programme. Il permet de structurer et centraliser les données tout en les rendant accessibles aux blocs fonctionnels (FB) et aux blocs d'organisation (OB).

II.6 WINCC SUR TIA Portal :

Le SIMATIC WinCC dans le TIA Portal fait partie d'un nouveau concept d'ingénierie qui propose un environnement pour la programmation, la configuration des systèmes de commande ainsi la visualisation des processus. C'est un logiciel d'ingénierie dédié à la configuration des interface homme-machine (IHM), incluant les pupitres SIMATIC et de PC standard par le logiciel de visualisation.

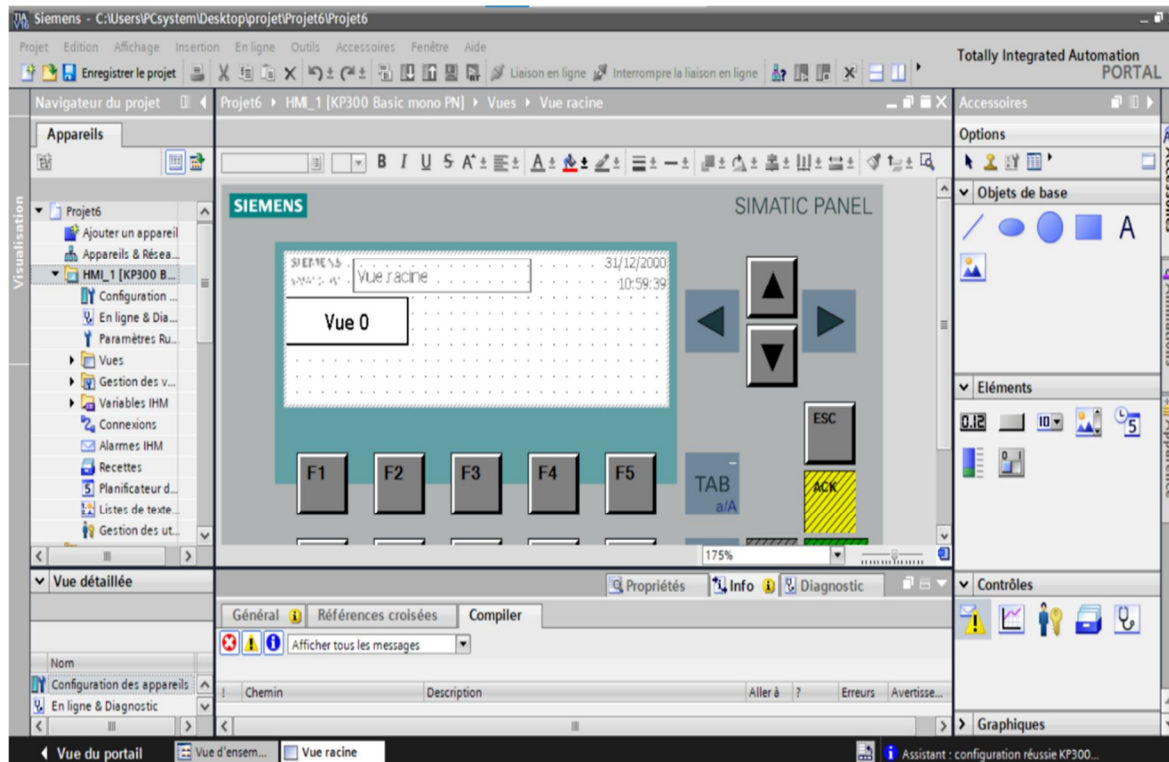


Figure II.21 : Représentation de IHM.

II.7 Factory IO :

Factory IO est la nouvelle génération de logiciels de simulation de parties opératives développés par Real Games pour l'apprentissage de l'automatisation. Factory IO permet de concevoir facilement et rapidement la ligne de fabrication par simple assemblage de sous-systèmes disponibles dans la bibliothèque fournie.

C'est un logiciel 3D utilisé pour les formations et pour les expérimentations dans l'automatisation industrielle. Il constitue un outil pédagogique performant pour arriver à comprendre, tester et valider des commandes automatisées.

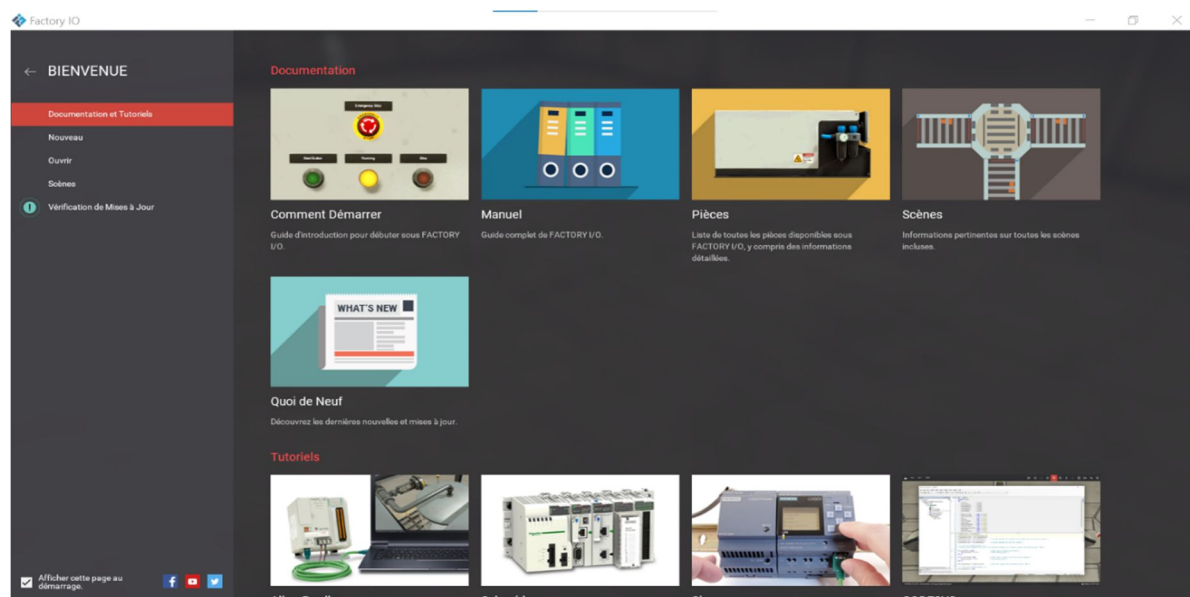


Figure II.22 : Vue de Factory IO.

La figure II.22 montre un écran d'accueil de Factory IO. Cet écran apparaît au lancement du logiciel. Il propose des ressources de démarrage pour les utilisateurs, qui permettent de créer une nouvelle scène ou de charger un projet existant.

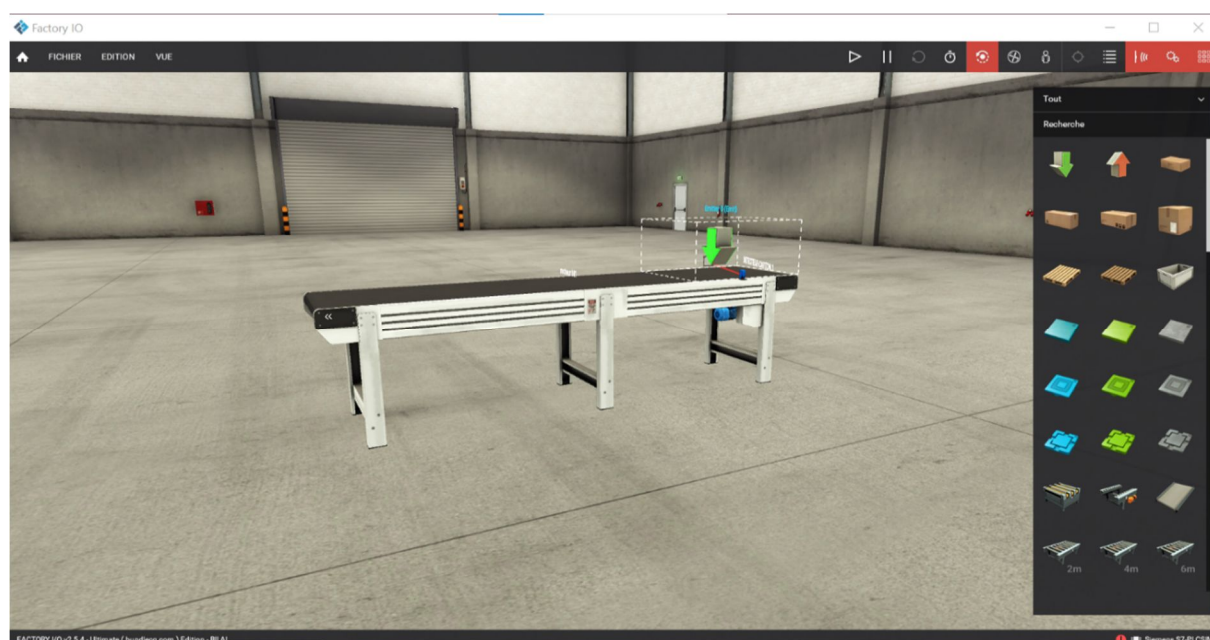


Figure II.23 : L'interface principale de Factory IO.

Dans la figure II.23, on trouve l'espace de travail (zone centrale). Elle représente le sol de l'usine virtuelle. C'est sur cet espace qu'on attribue les composants pour pouvoir construire et tester des processus industriels simulés.

Sur le panneau droit on trouve une bibliothèque de composants qui aide à construire un processus. Il suffit de glisser ces éléments dans la zone centrale pour les ajouter à la scène.

Il sert à créer un scénario industriel personnalisé pour apprendre l'automatisme industriels, tester un programme d'automate et simuler des chaines de production complètes.

II.8 Conclusion :

Tout au long de ce chapitre nous avons présenté les généralités sur l'automatisation et les différentes structures et architectures d'un automate programmable où nous nous sommes focalisés sur l'automate S7-1500. De plus, nous avons discuté en détail les logiciels Siemens TIA Portal et Factory IO.

Les automates programmables industriels, couplés aux logiciels TIA Portal et Factory IO, représentent une avancée majeure dans le domaine de l'automatisation. Leur complémentarité permet une programmation efficace et une phase de test virtuel essentielle avant toute mise en service. Ceci permet de réduire les erreurs, optimiser les performances et renforcer la sécurité des systèmes.

L'association des API avec les logiciels de simulation est essentielle aujourd'hui dans les projets d'automatisation modernes et constitue une base solide pour l'intégration des solutions.

CHAPITRE III

Programmation et simulation du système.

III.1 introduction :

Dans l'industrie, il est courant de rencontrer différentes problématiques qui peuvent ralentir la production, ce qui oblige les ingénieurs à trouver et concevoir des solutions.

Après l'exposition de notre problématique, nous allons présenter la solution proposée. Nous allons montrer la démarche à suivre pour faire le programme avec TIA Portal. Une simulation virtuelle avec le logiciel « Factory I/O » sera réalisée.

III.2 Explication de la problématique :

La problématique est présentée lors de déplacement des cartons de niveau 1 vers le niveau 0 par inertie. Ce type de déplacement entraîne l'ouverture des boîtes due à la collusion des cartons ce qui entraîne la perte des produits. Les responsables ont pris la décision de placer un operateur sur la machine qui va se charger de déplacer les cartons. Cette solution n'est pas idéale vu que cette tâche est pénible pour les opérateurs et du côté de la gestion des couts et de temps.

III.3 Discussion de la solution :

En raison de la présence de ces problèmes cumulés, nous avons suggéré de placer un nouveau système de convoyage motorisé. Ce système de convoyage va permettre de déplacer les cartons du niveau de la production jusqu'à la zone de stockage sans intervention manuelle et en toute sécurité. Ce système se compose de trois niveaux et chaque niveau comporte des convoyeurs horizontaux et entre deux niveaux un élévateur vertical est placé pour les relier.

III.4 Matériels et logiciels utilisés :

Pour pouvoir réaliser ce projet, nous avons eu besoin d'utiliser le logiciel de programmation TIA Portal et le logiciel de simulation virtuel 3D Factory I/O pour créer des convoyeurs, des moteurs, des capteurs, des élévateurs et des vérins virtuels.

III.5 Modélisation de système dans Factory I/O :

A l'aide du logiciel Factory I/O une modélisation virtuelle du système a été réalisé, et ceci nous permet de simuler un environnement industriel en 3D. L'objectif est de représenter le système automatisé de convoyage ainsi que de tester le fonctionnement du système en temps virtuel sans avoir besoin des équipements physiques et ce avant de passer à la réalisation.

- La maquette représente un système automatisé virtuel réalisé à l'aide du logiciel Factory IO qui permet de simuler des environnements industriels en 3D.
- La maquette intègre plusieurs éléments essentiels représentés sur la figure III.1, on trouve :
- Convoyeurs motorisés horizontaux : ces convoyeurs assurent le transport des cartons d'un point à un autre et ils sont entraînés par des moteurs électriques.
- Élévateur (ascenseur industriel) : son rôle principal est de transporter des objets verticalement d'un niveau à un autre.
- Capteur de détection : il est utilisé pour détecter la présence des cartons sur les convoyeurs.
- Des actionneurs (moteurs, vérins).

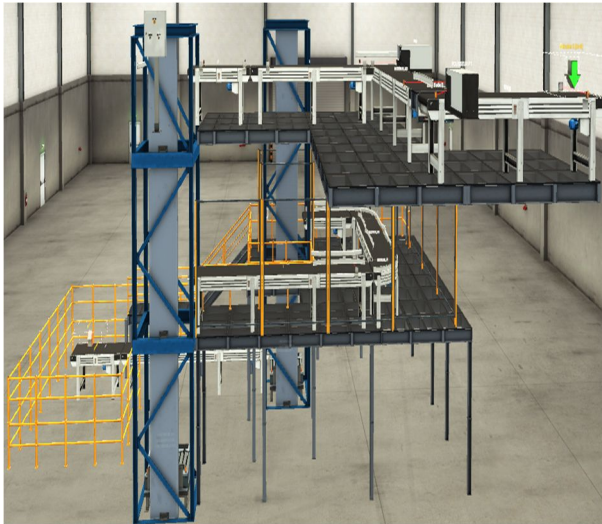


Figure III.1 : Représentation de la maquette.

Le fonctionnement souhaité par le système est le suivant :

Un carton sera détecté au niveau du convoyeur supérieur par un capteur. Cette détection déclenchera le moteur qui va prendre en charge d'entraîner le convoyeur, pour que les cartons soient dirigés vers le bas. L'élévateur est placé pour entamer cette tâche. Une fois le carton est arrivé au niveau inférieur, il continue son trajet via un convoyeur jusqu'à son arrivée à la destination souhaitée.

III.6 Programmation dans TIA Portal V16 :

III.6.1 Création de projet dans TIA Portal V16 :

La création de projet dans TIA Portal commence par le choix de l'option « Créer un projet ».

On peut donner un nom au projet, choisir un chemin où il sera enregistré, ajouter un commentaire ou encore définir l'auteur du projet.

Une fois ces informations sont entrées, il suffit de valider en cliquant sur le bouton « Créer ».

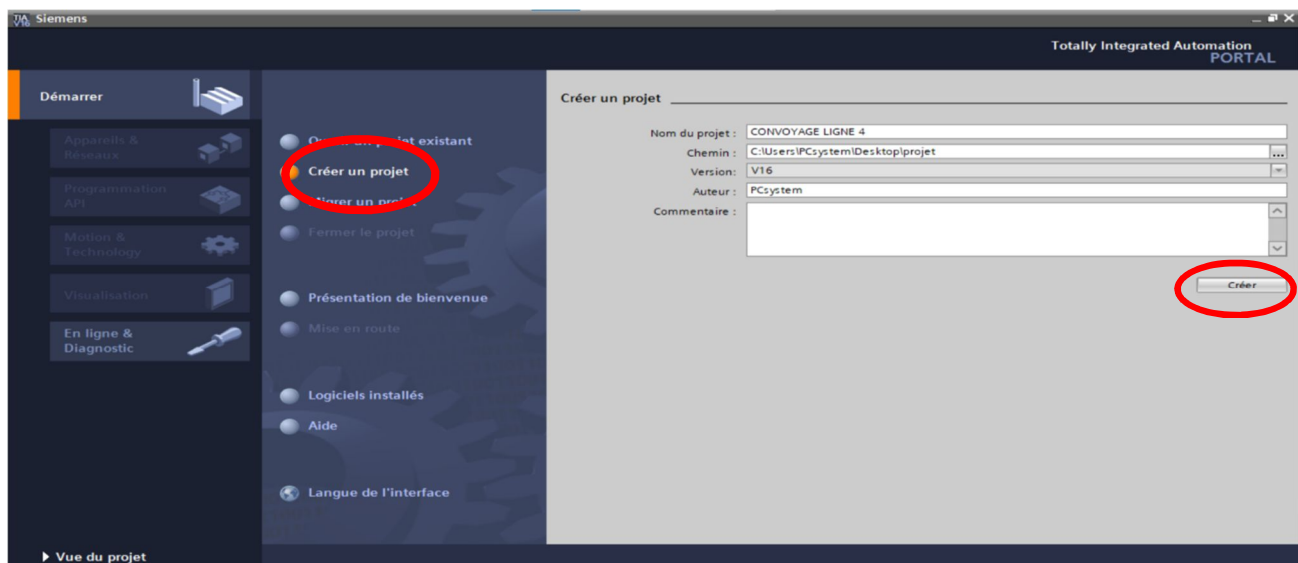


Figure III.2 : Création de projet dans TIA Portal.

III.6.2 Configuration matérielle :

Une fois le projet est créé on passe à la configuration matérielle ou nous allons choisir et configurer les modules du l'API adopté, pour cela on va passer par la vue de projet et cliquer sur ajouter un appareil dans le navigateur du projet.

Une liste des éléments que on peut ajouter s'affiche (API, HMI, Système PC). Notre choix repose sur CPU afin d'ajouter les modules complémentaires (alimentation, E/S TOR ou analogique, module de communication...). La CPU choisi est de type S7-1500.

La figure III.3 représente la vue d'ajouter de la CPU.

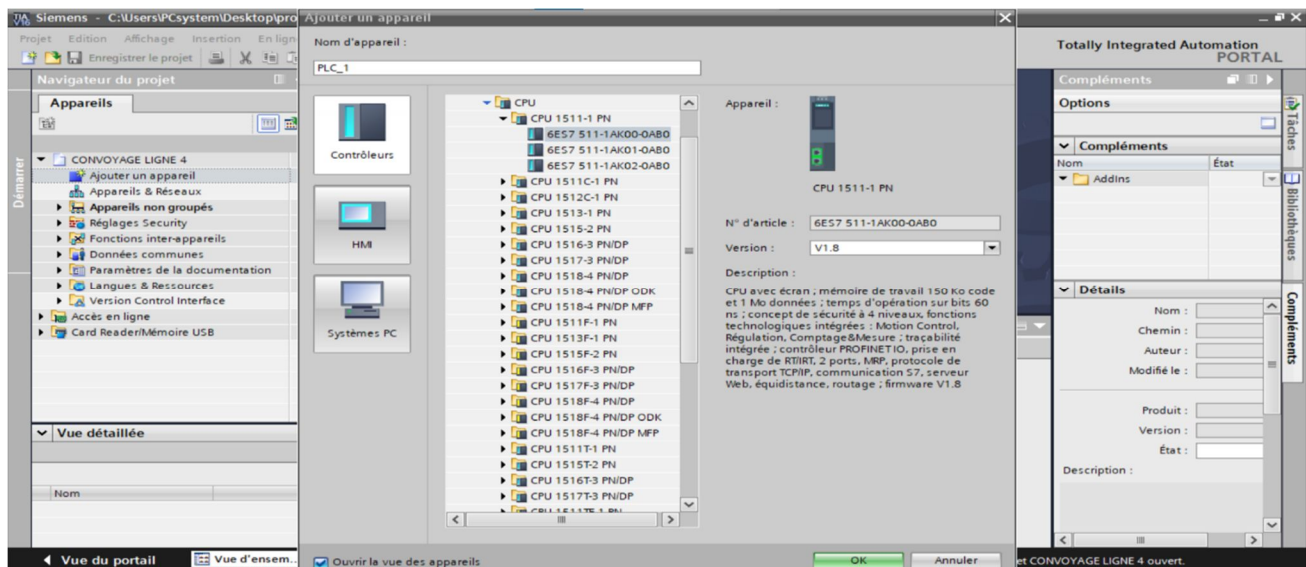


Figure III.3 : Vue d'ajouter de CPU.

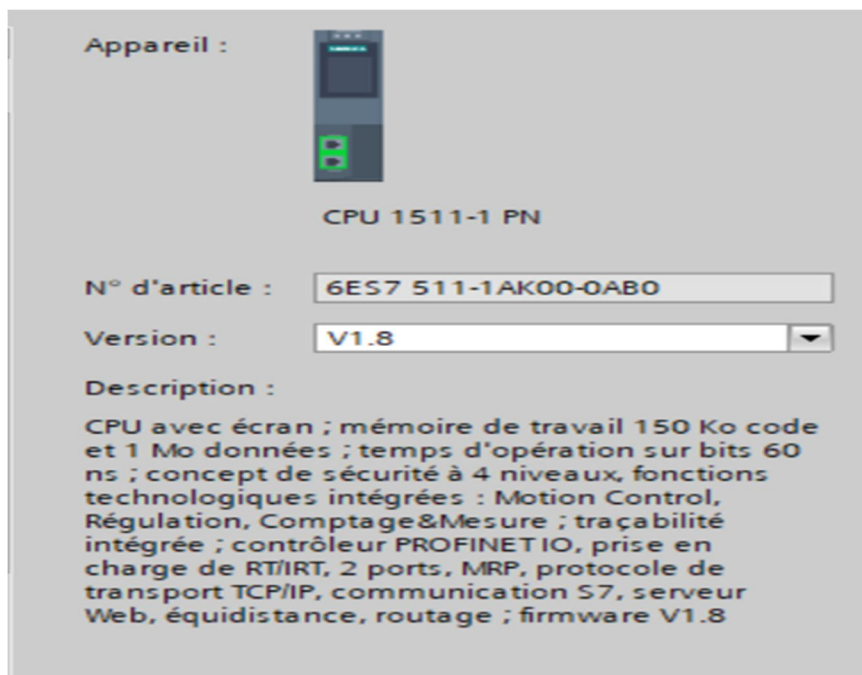


Figure III.4 : Représentation des caractéristiques de CPU.

Nous avons choisi l'API dont les caractéristiques sont données à la figure III.4. Après cela, nous avons choisi les modules d'entrées/sorties pour compléter le manque des entrées disponibles initialement.

La figure III.5 montre la fenêtre qui permet la mise en œuvre de cette étape où les modules ajoutés sont représentés.

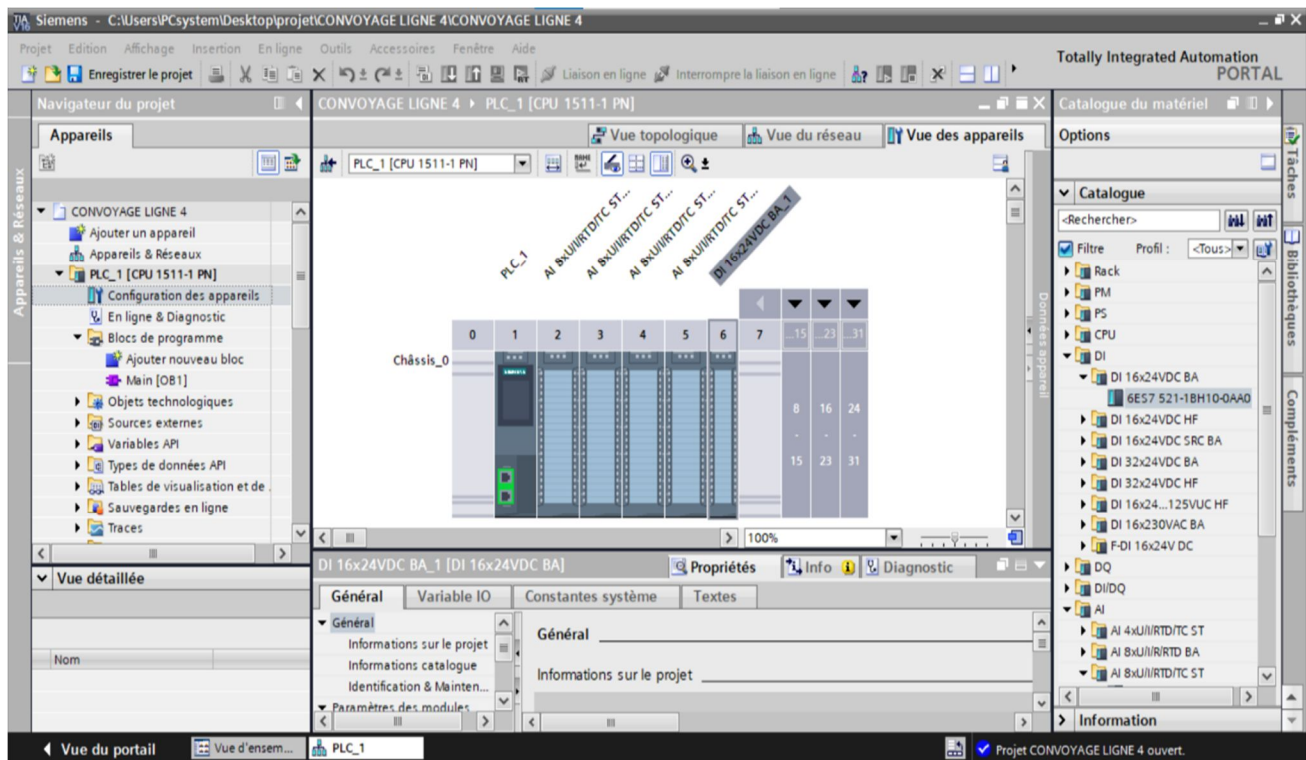


Figure III.5 : Configuration et paramétrage du matériel.

Les modules complémentaires de l'API peuvent être ajoutés en utilisant le catalogue. Il faut repasser par la commande « ajouter un appareil » dans le navigateur du projet pour pouvoir ajouter un écran ou un autre API.

III.6.3 Adressage des entrées sorties :

Après avoir ajouté les modules d'entrées/sorties, nous allons vers une autre étape où nous définissons une liste des variables qui vont être utilisées lors de la programmation.

L'utilisation des noms appropriés dans l'élaboration de la table de mnémoniques rend le programme plus compréhensible et plus facile à manipuler. Les variables peuvent être définies pendant la programmation.

La figure III.6 montre un extrait de la table des variables, L'intégralité de la table des variables est présentée en Annexe.

CONVOYAGE LIGNE 4 ▸ PLC_2 [CPU 1511-1 PN] ▸ Variables API ▸ Standard-Variablen-tabelle [121]									
Variables									
Standard-Variablen-tabelle									
	Nom	Type de données	Adresse	Réma...	Acces...	Écritu...	Visibl...	Surveilla...	Commentaire
1	FDC_DESCENSEUR_I_UP	Bool	%I2.0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
2	DESCENSEUR_I_UP	Bool	%Q2.0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
3	FDC_RALLENTIS_I_UP	Bool	%I2.1		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
4	RALENTISSEMENT_I	Bool	%Q2.1		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
5	ROULEAUX_IN_I	Bool	%Q2.2		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
6	FDC_RALENTIS_I_DOWN	Bool	%I2.3		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
7	LIMITE_DROITE_DESC_I	Bool	%I2.2		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
8	DESCENSEUR_DOWN_I	Bool	%Q2.3		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
9	FDC_DESCENSEUR_I_DOWN	Bool	%I2.4		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
10	ROULEAUX_OUT_I	Bool	%Q2.4		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
11	CAPTEUR_ARRIVEE_M2	Bool	%I0.7		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
12	POUSSEUR_P2	Bool	%Q0.5		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
13	MOTEUR_M2	Bool	%Q0.4		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
14	TEMPO_ARRIVEE_CARTON_M2	Timer	%T2		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
15	POUSSEUR_P2_ARRIERE	Bool	%I1.1		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
16	CAPTEUR_ARRIVEE_P2	Bool	%I1.0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
17	RETARD_P2	Timer	%T3		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
18	POUSSEUR_P2_AVANT	Bool	%I1.2		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
19	CAPTEUR_ARRIVEE_M3	Bool	%I1.3		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
20	MOTEUR_M3	Bool	%Q0.6		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
21	TEMPO_MOTEUR_M3	Timer	%T4		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		

Figure III.6 : Onglet de la table de variable.

III.6.4 Création des blocs fonctionnel :

Étant donné que notre convoyeur est devisé en plusieurs parties, nous allons ouvrir plusieurs FC pour que chaque partie soit appelée dans le bloc d'organisation (OB).

III.6.4.1 Bloc d'organisation (OB) :

Le dossier bloc, contient les blocs que l'on doit charger dans la CPU pour réaliser la tâche d'automatisation. Nous avons utilisé le bloc d'organisation (OB) qui fait appel aux autres blocs qui constituent le programme. Lorsqu'on appelle un bloc fonctionnel dans l'OB1, un bloc de donnée associé sera créé automatiquement.

Dans l'OB1, nous réalisons des réseaux d'appels pour nos fonctions en utilisant une boîte vide puis entrant un nom identique à celui indiqué dans la FC concernée par l'appel.

La figure III.7 illustre cette démarche.

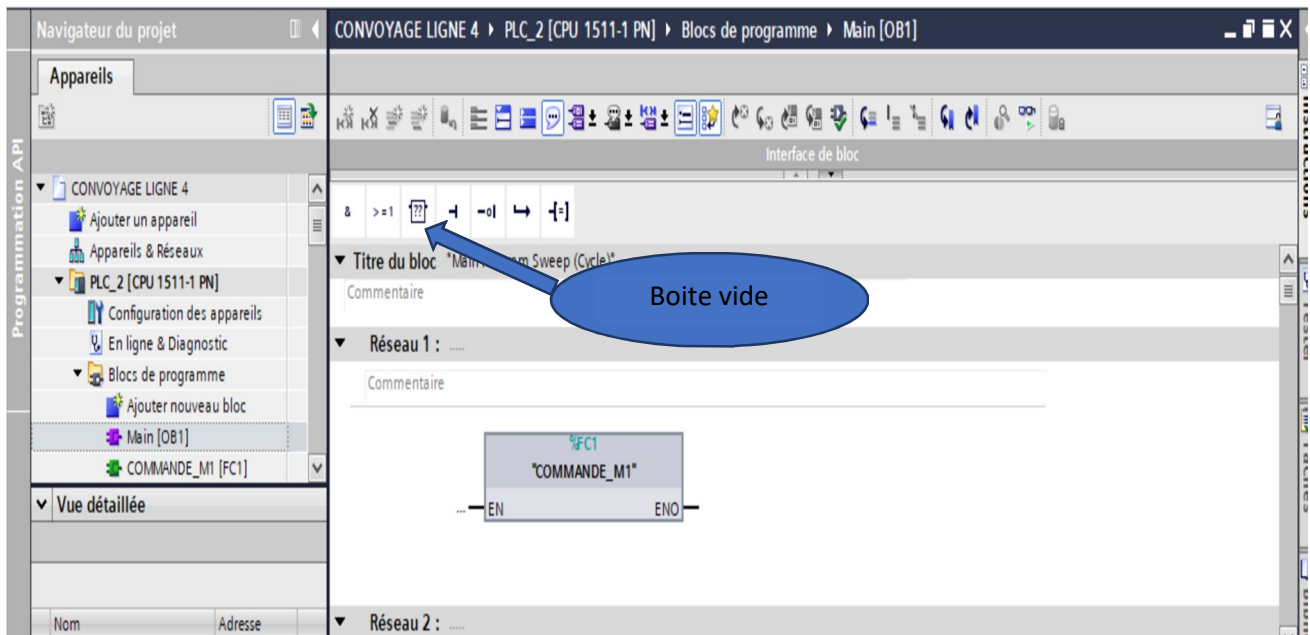


Figure III.7 : Création de réseau dans l'OB1.

La figure III.8 illustre les réseaux existant dans le bloc OB1.

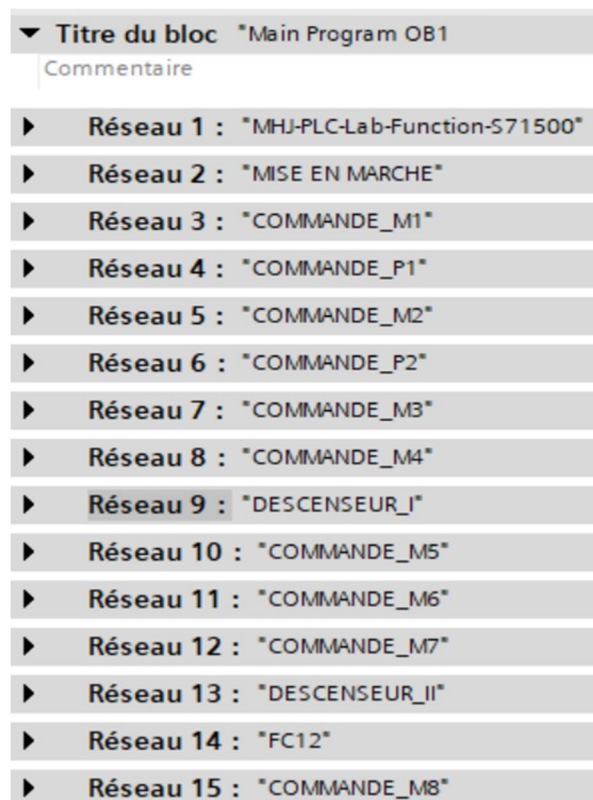


Figure III.8 : Réseaux du bloc OB.

Les réseaux d'appels existant sont présentés dans les figures suivantes (figures de III.9 à III.23).

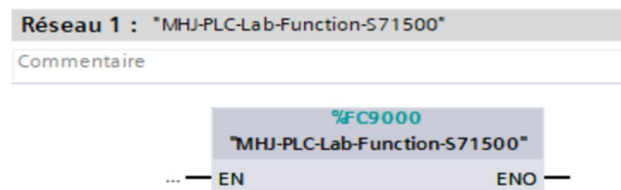


Figure III.9 : Réseau d'appel de la fonction-S7-1500.



Figure III.10 : Réseau d'appel de la mise en marche



Figure III.11 : Réseau d'appel de commande_M1.

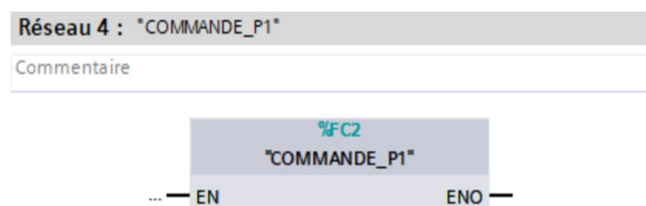


Figure III.12 : Réseau d'appel de commande_P1.

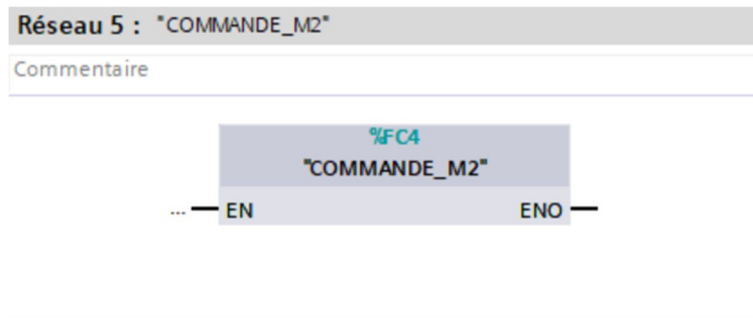


Figure III.13 : Réseau d'appel de commande_M2.



Figure III.14 : Réseau d'appel de commande_P2.

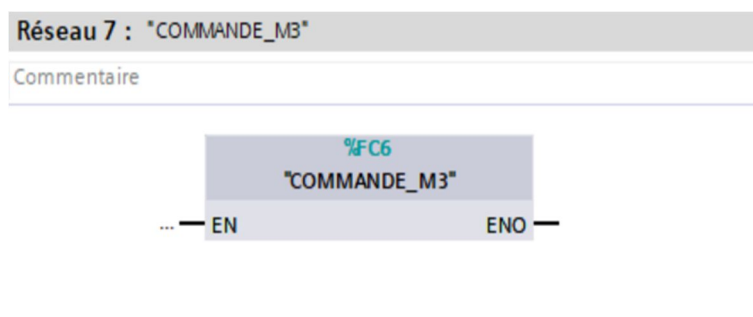


Figure III.15 : Réseau d'appel de commande_M3.

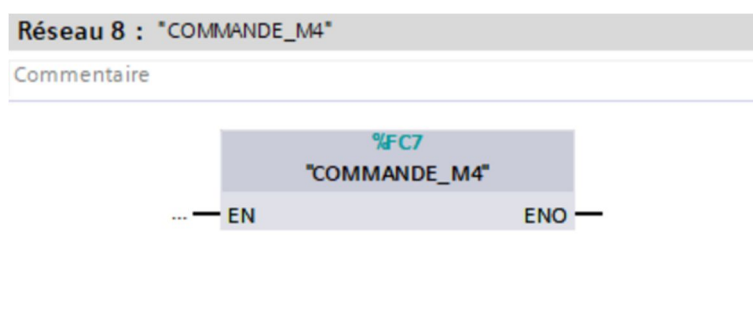


Figure III.16 : Réseau d'appel de commande_M4.

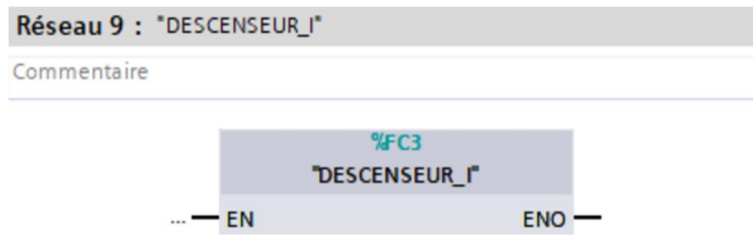


Figure III.17 : Réseau d'appel de descenseur I.

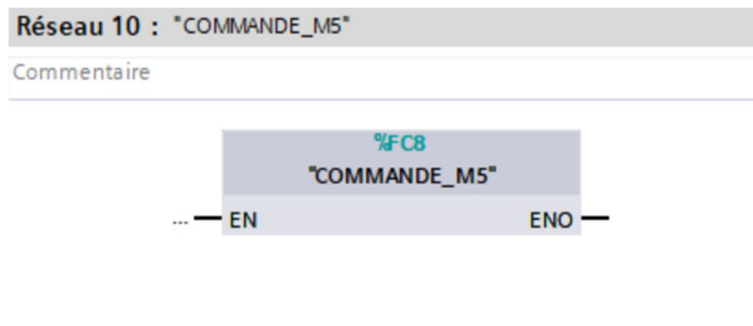


Figure III.18 : Réseau d'appel de la commande_M5.

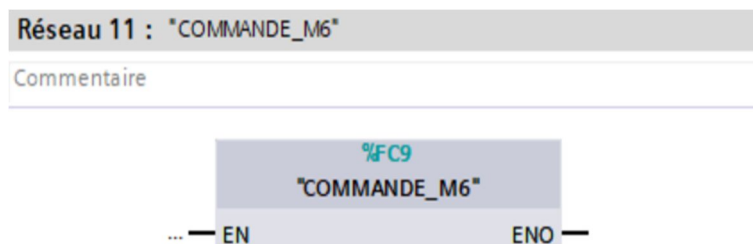


Figure III.19 : Réseau d'appel de la commande_M6.

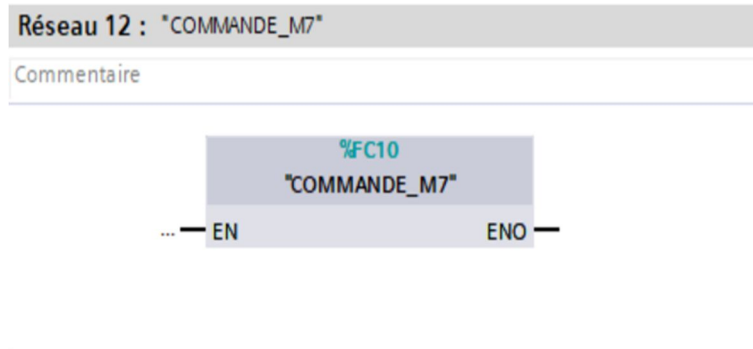


Figure III.20 : Réseau d'appel de la commande_M7.

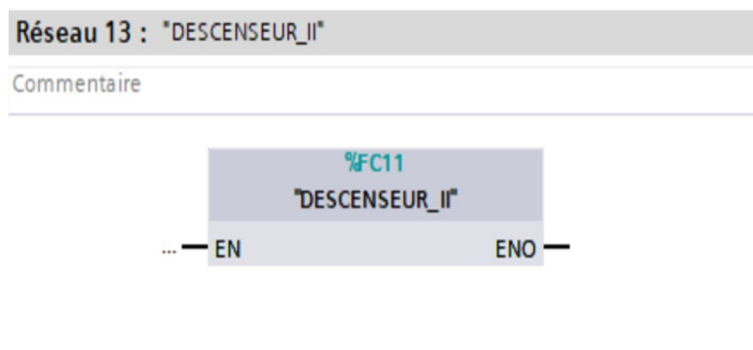


Figure III.21 : Réseau d'appel de descenseur II.

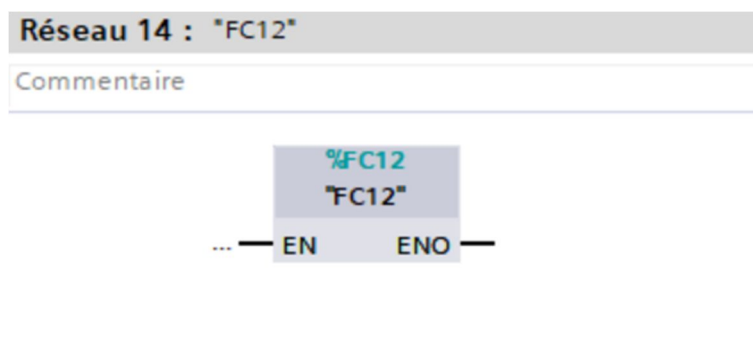


Figure III.22 : Réseau d'appel de FC12.

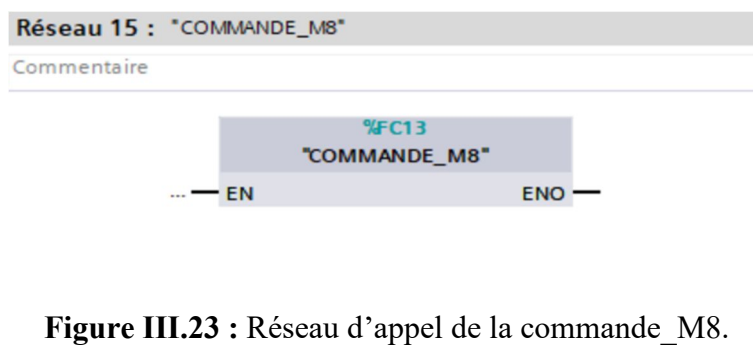


Figure III.23 : Réseau d'appel de la commande_M8.

III.6.4.2 Bloc fonctionnel (FC) :

En ouvrant le bloc fonctionnel FC, nous trouvons des réseaux vides où nous allons écrire notre programme en utilisant les contacts NF/NO. Les données des variables temporaires sont perdues après l'exécution de la fonction.

La figure III.24 représente l'ensemble de nos blocs FC.
















	COMMANDE_M1 [FC1]
	COMMANDE_M2 [FC4]
	COMMANDE_M3 [FC6]
	COMMANDE_M4 [FC7]
	COMMANDE_M5 [FC8]
	COMMANDE_M6 [FC9]
	COMMANDE_M7 [FC10]
	COMMANDE_M8 [FC13]
	COMMANDE_P1 [FC2]
	COMMANDE_P2 [FC5]
	DESCENSEUR_I [FC3]
	DESCENSEUR_II [FC11]
	FC12 [FC12]
	MHJ-PLC-Lab-Function-S71500 ...
	MISE EN MARCHÉ [FC0]

Figure III.24 : Les blocs FC.

Dans les figures qui suivent, nous présentons quelques réseaux de notre système. Les autres réseaux seront présentés en Annexe.

La figure III.25 suivante représente la mise en marche du système de convoyage.

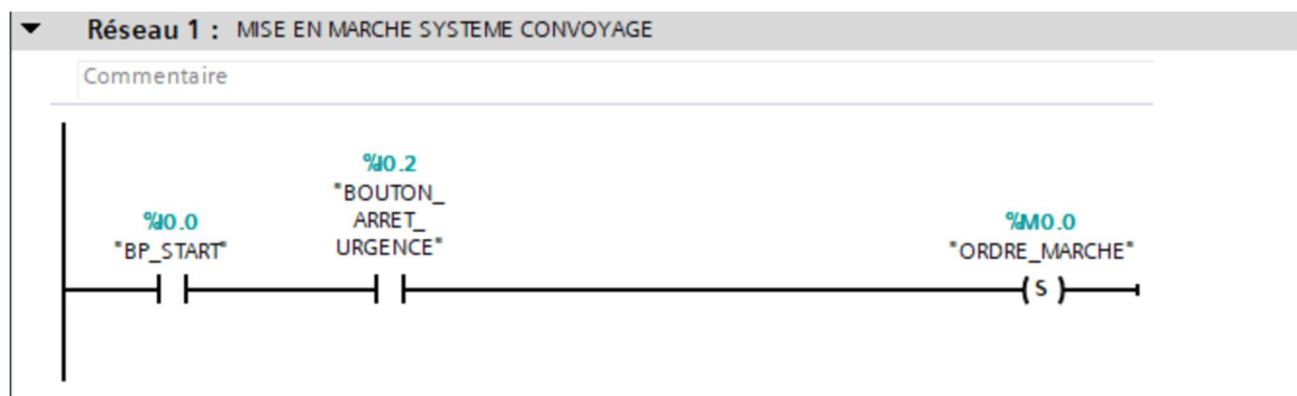


Figure III.25 : Mise en marche de système convoyage.

La figure III.26 suivante représente le démarrage du moteur 1.

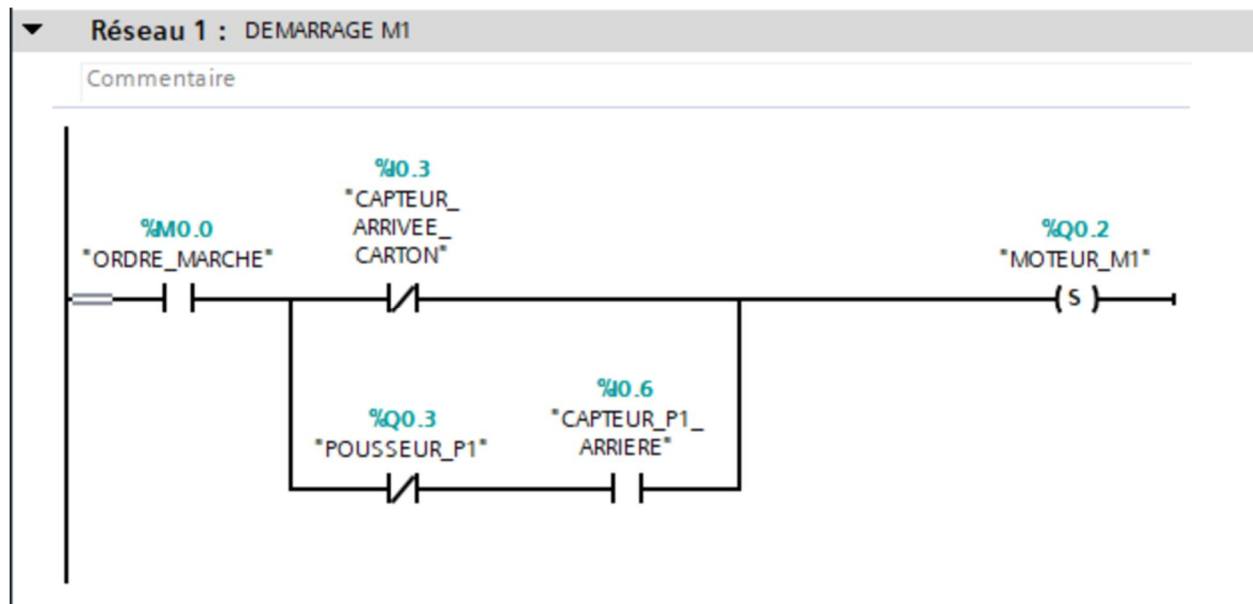


Figure III.26 : Démarrage M1.

La figure III.27 suivante représente la sortie du pousseeur P1.

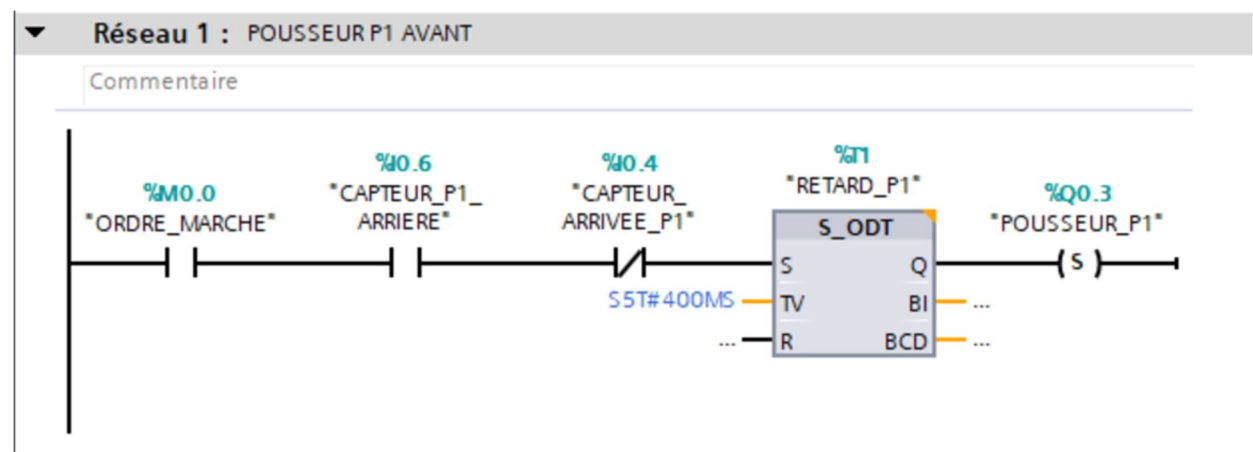


Figure III.27 : Pousseeur P1 avant.

La figure III.28 suivante représente le chargement de descenseur 1.

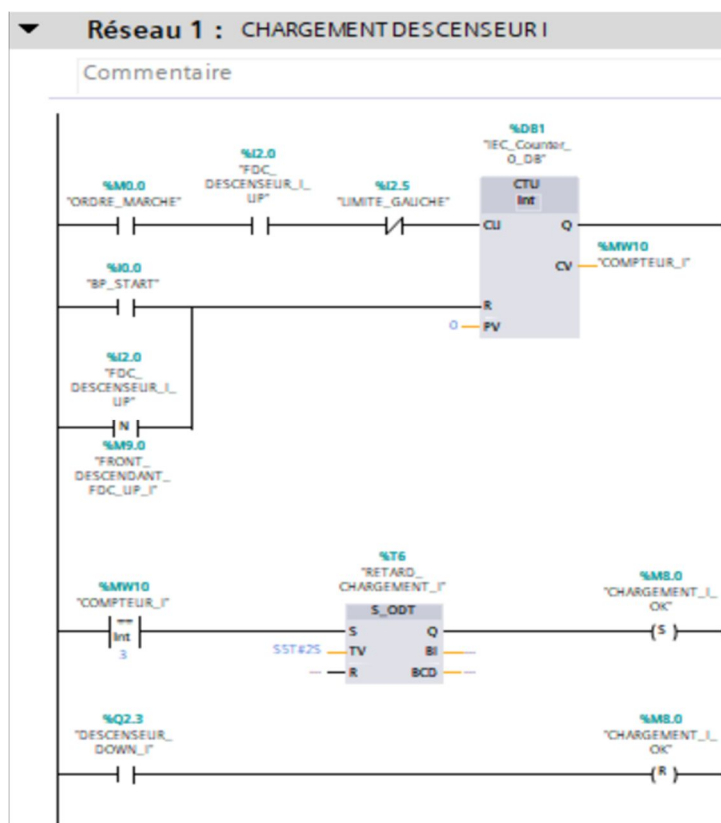


Figure III.28 : Chargement descenseur 1.

La figure III.29 suivante représente le déchargement de descenseur 2.

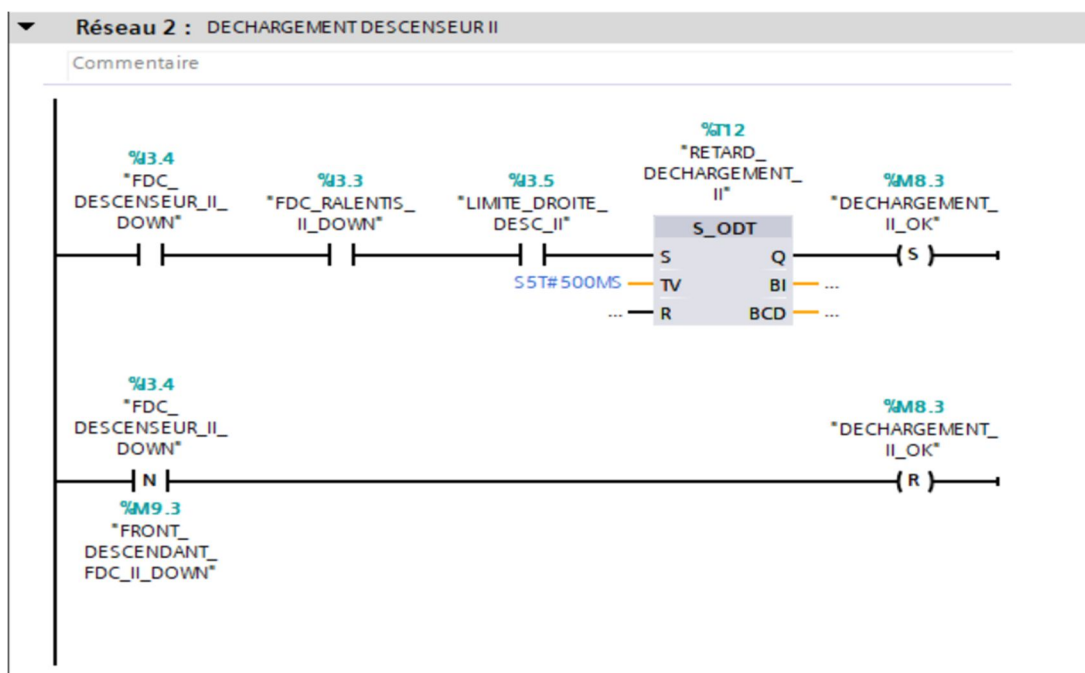


Figure III.29 : Déchargement descenseur 2.

III.6.5 Compilation et test en simulation sous TIA Portal :

Après que la programmation est finalisée dans TIA Portal, nous devons compiler et charger le programme afin de le tester s'il fonctionne sans erreurs.

III.6.6 Chargement du programme :

Le chargement de notre programme dans un API virtuel nous permet de simuler le comportement de l'automate sans avoir besoins du matériel réel.

Avant de procéder au chargement du programme vers l'automate, il est nécessaire de vérifier que la connexion entre TIA Portal et l'automate virtuel est bien établie. La figure III.30 montre que l'automate est détecté et que la connexion a été bien établie.

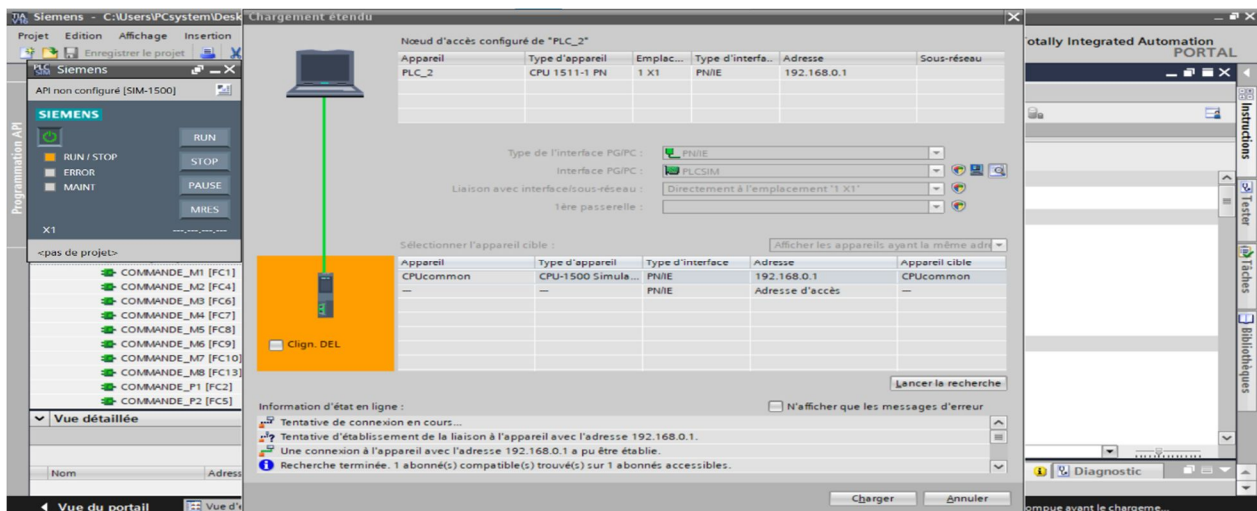


Figure III.30 : Configuration et détection de l'automate simulé.

Les deux figures qui suivent montre que le programme est prêt pour le chargement ainsi que l'API est bien simulée.

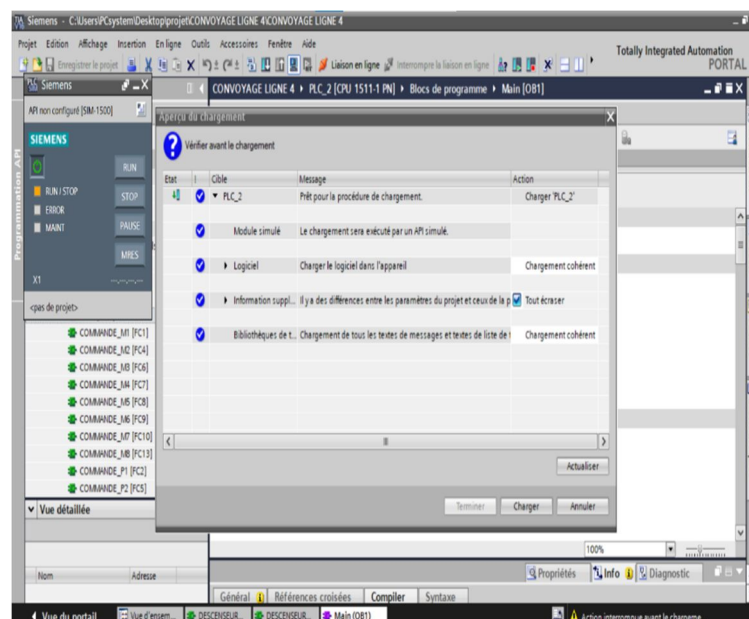


Figure III.31 : Validation et préparation au chargement.

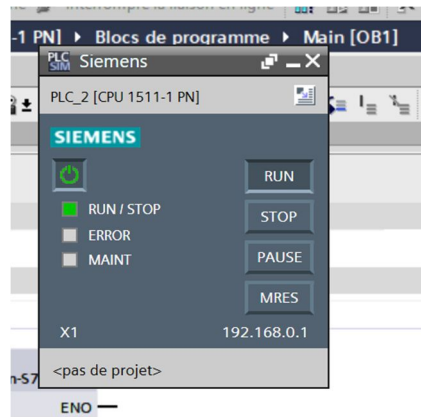


Figure III.32 : Démarrage l'exécution de programme.

Après avoir chargé le programme dans l'automate virtuel et mettre l'automate virtuel en mode « Run », l'étape suivante consiste à simuler l'interaction entre le programme automate et l'environnement virtuel créé avec le logiciel Factory I/O.

Cette simulation permet de valider le fonctionnement logique du programme dans un environnement virtuel 3D.

III.7 Connexion entre Factory IO et TIA Portal via PLCSIM :

Pour simuler le programme conçu dans TIA Portal avec l'environnement 3D Factory I/O, les étapes suivantes doivent être respectées :

1. Ouverture de Factory I/O et choix d'un scenario. La figure III.33 suivante montre la vue de notre environnement virtuel.

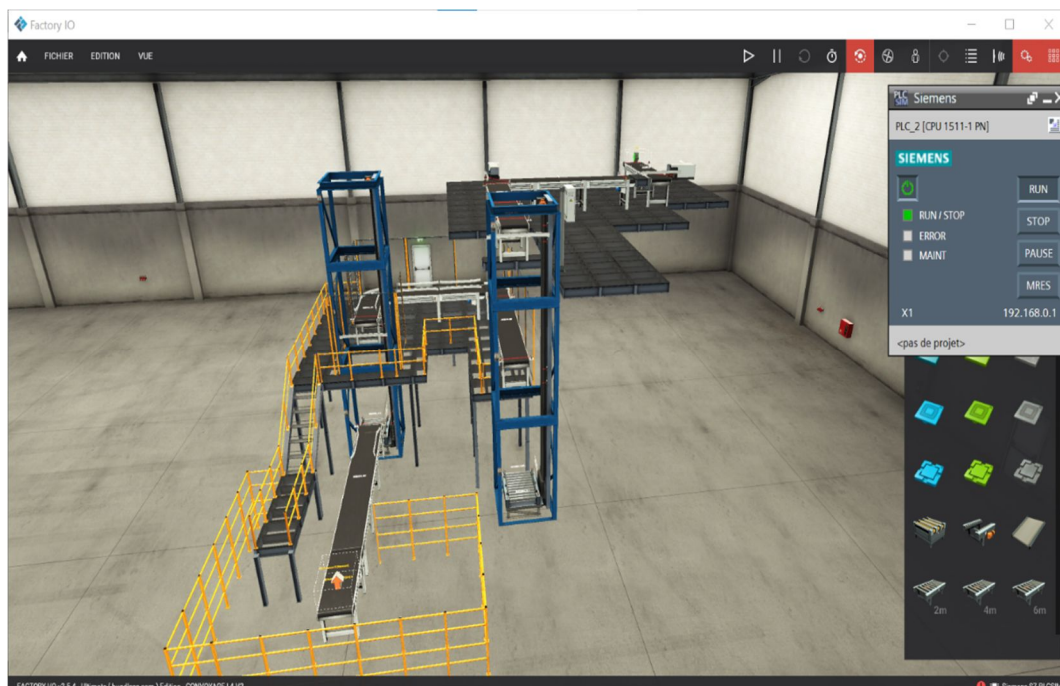


Figure III.33 : Vue de l'environnement virtuel.

2. Accès au menu (driver) et sélection de Siemens S7-PLCSIM dans la liste déroulante des pilotes comme montré par la figure III.34.

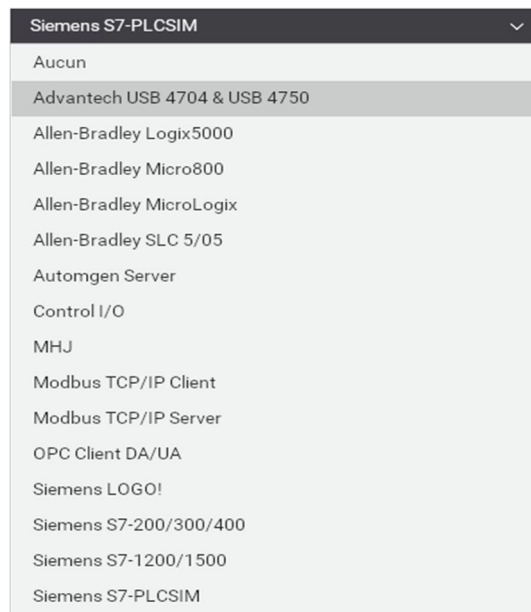


Figure III.34 : Représentation de choix de Siemens S7-PLCSIM.

3. Sur « configuration », nous faisons entrer l'adresse IP et nous vérifions le port de communication. Puis en cliquant sur « connecter » pour établir la liaison entre Factory I/O et le simulateur PLCSIM. Une fois la connexion établie, un tableau s'affiche et il nous permet d'associer les adresses des capteurs et actionneurs de Factory I/O avec les adresses d'entrées et sorties du programme développé dans TIA Portal. La figure III.35 illustre tout ce qu'a été mentionné.

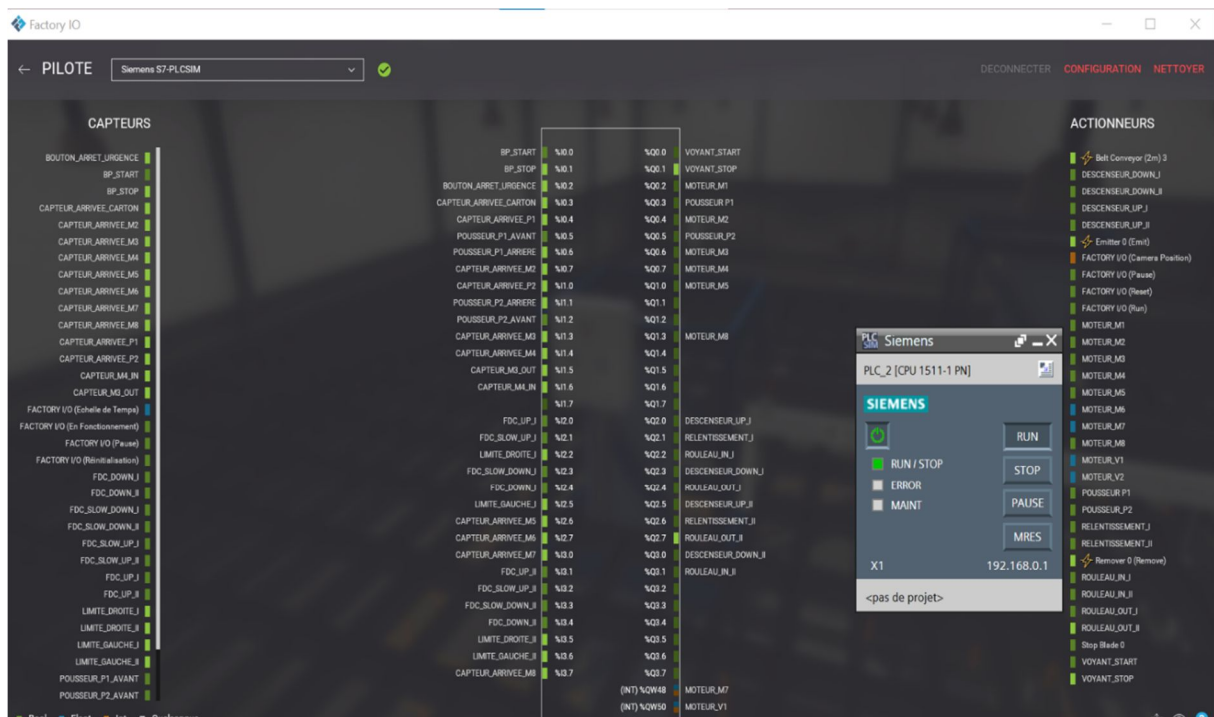


Figure III.35 : Connexion entre Factory I/O et simulateur PLCSIM.

Une fois la connexion est établie entre Factory I/O et l'automate simulé via PLCSIM, la simulation peut être lancée. Pour démarrer l'environnement virtuel il suffit de cliquer sur le bouton « Play » et les capteurs et actionneurs deviennent actifs et réagissent en fonction du programme chargé dans l'API virtuel simulé sous TIA Portal.

Cette phase permet de valider visuellement le fonctionnement du système automatisé et toute anomalie observée peut être corrigée en modifiant le programme dans TIA Portal.

Les figures III.36 suivantes représentent le démarrage du fonctionnement de notre système.

Pour que le système fonctionne, nous devons cliquer sur le bouton « Play » dans Factory IO et dans le bouton « voyant_start ».

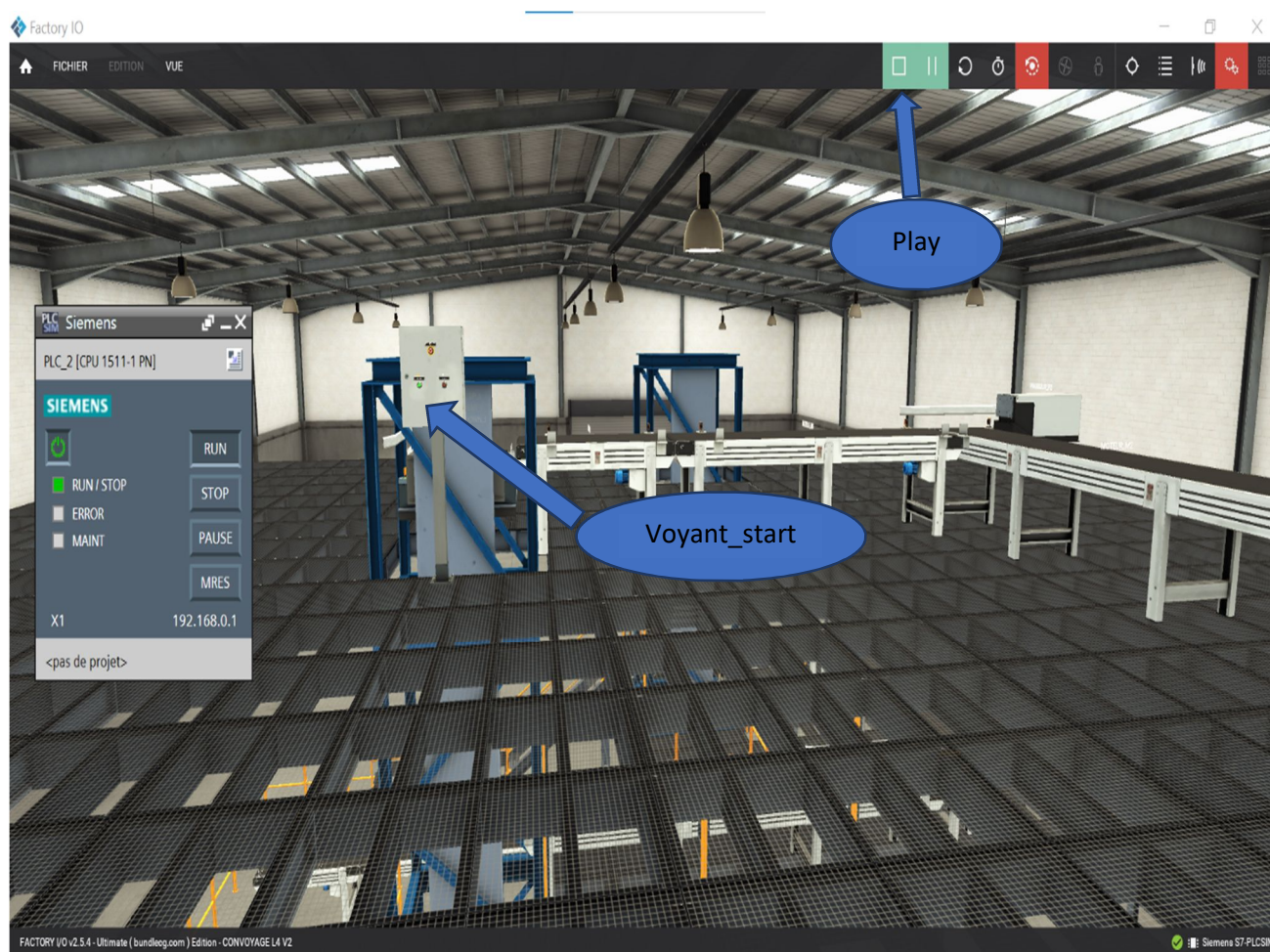


Figure III.36 : Démarrage fonctionnement de système.

La détection de la présence de carton via un capteur va permettre au moteur M1 d'entraîner le convoyeur qui est chargé de déplacer les cartons. La figure III.37 représente le déplacement des cartons via le convoyeur.

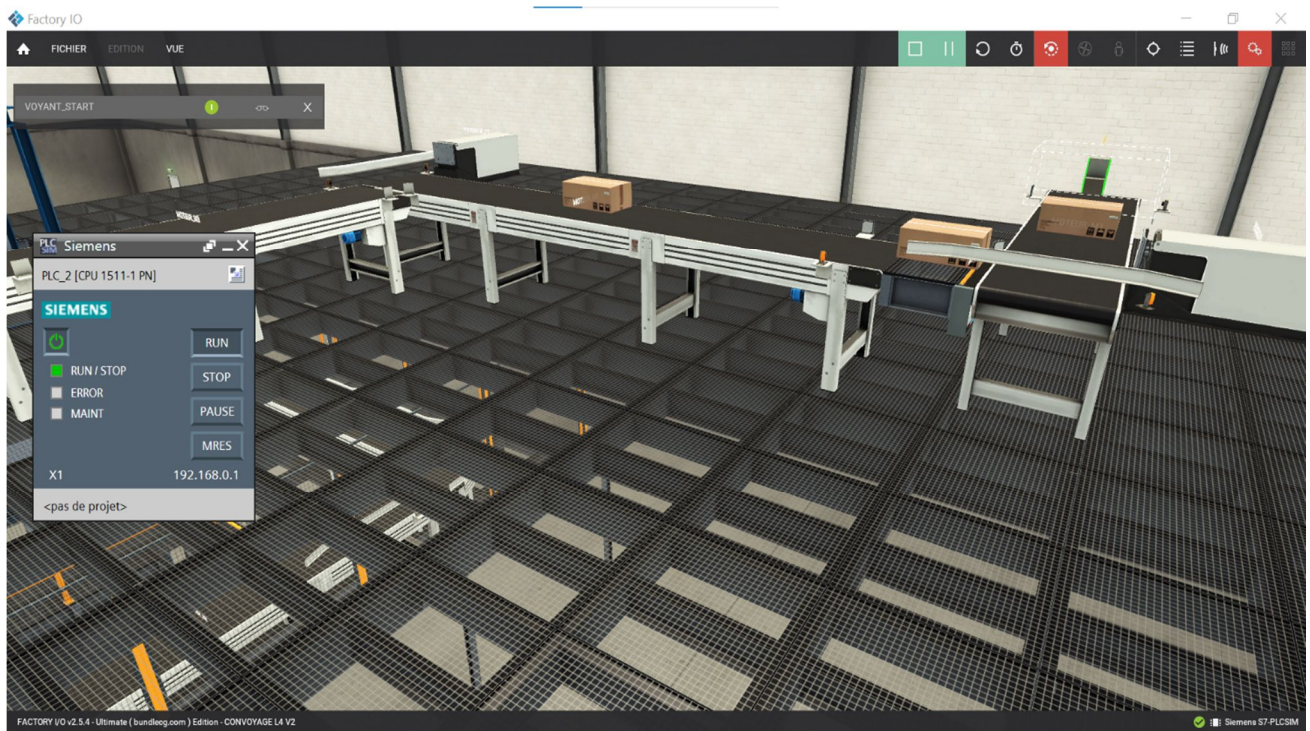


Figure III.37 : Déplacement des cartons.

La figure III.38 suivante représente le chargement et le déchargement du descenseur 1.

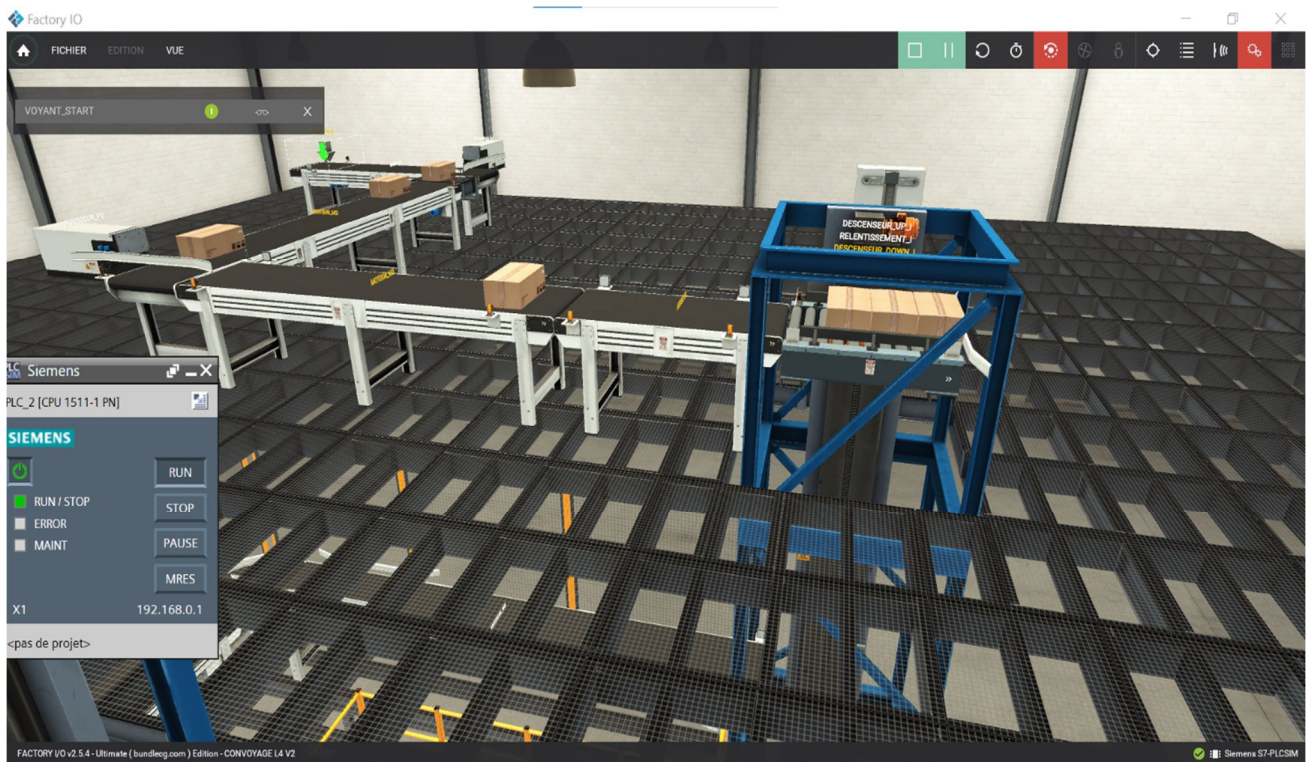


Figure III.38 : Chargement du descenseur1.

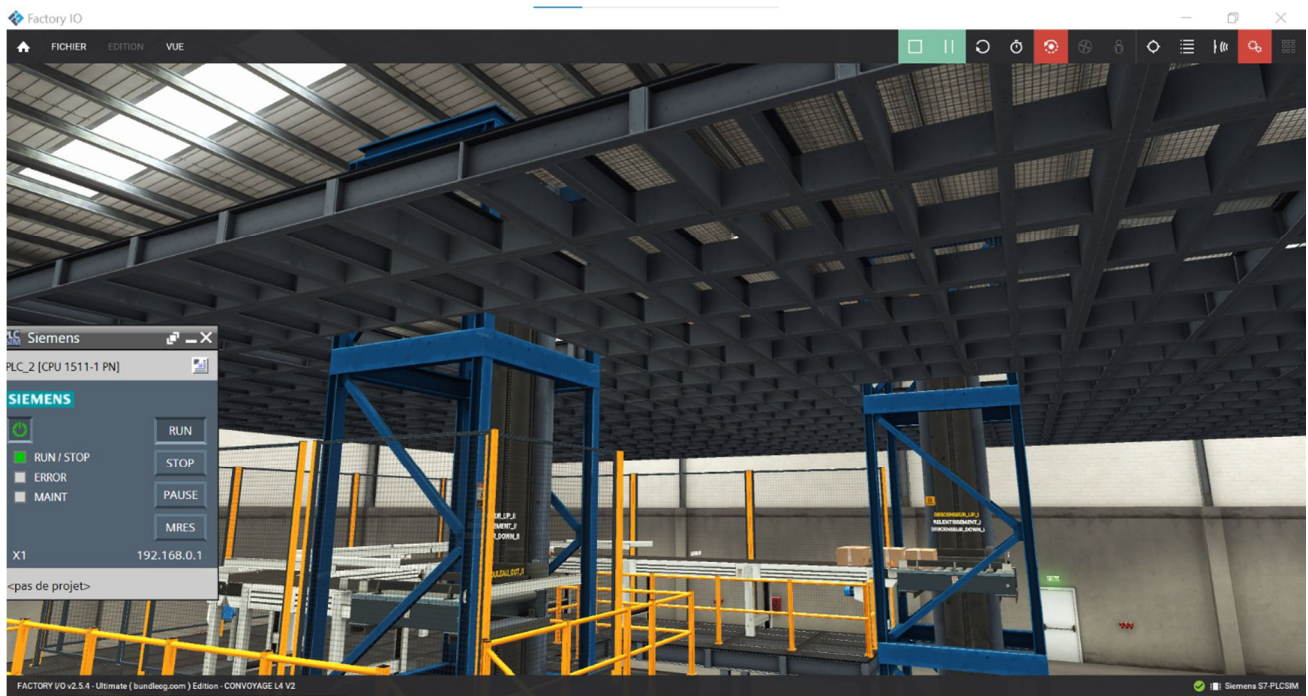


Figure III.39 : Déchargement du descenseur 1.

La figure III.40 représente une vue sur le fonctionnement de notre environnement virtuel.



Figure III.40 : Vue du fonctionnement de l'environnement virtuel.

La simulation de ce système automatisé a montré un fonctionnement cohérent et fiable. L'utilisation des capteurs ont permis d'assurer une détection précise des objets aux différents points tels que les positions d'arrêt. Leur intégration a renforcé la précision du cycle automatisé.

Grace à leurs retours d'états, les actionneurs ont été commandés de manière synchrone assurant des opérations sans blocage ni erreurs.

III.8 Discussion à propos de la solution :

Notre solution repose sur la conception d'un système de transfert automatisé, contrôlé par un automate programmable industriel API programmé via TIA Portal et simulé dans un environnement virtuel à l'aide de Factory IO.

Ce système a permis un déplacement contrôlé et progressif des cartons à l'aide des convoyeurs motorisés, élévateurs et des capteurs de détection assurant la fluidité, la sécurité et le transfert sans risque de chute ou de collision. Notre solution répond parfaitement aux objectifs souhaités cités ci-après :

- Suppression des pertes liées à l'ouverture des cartons.
- La sécurité du personnel en éliminant la tâche physique pénible.
- Continuité et régularité de la production.
- Fiabilité du système, validée par la simulation dans un environnement virtuel.

Donc, cette solution est prête à être transposée au système réel.

III.9 Conclusion :

A travers ce chapitre, nous avons présenté les différentes étapes de conception, de programmation et de simulation du système automatisé que nous avons proposé comme solution à la problématique.

La fiabilité technique du système est due à l'utilisation de TIA Portal pour le développement du programme automate, ainsi que Factory IO utilisé pour la simulation virtuelle.

Vue que les résultats obtenus montrent un fonctionnement correct et les objectifs visés ont été obtenus, ce travail confirme l'efficacité de l'automatisation dans la résolution des problèmes industriels.

Conclusion générale

Conclusion générale :

Au terme de ce mémoire, nous avons proposé et mis en œuvre une solution d'automatisation répondant à une problématique réelle rencontrée dans le système de convoyage de cartons au sein de l'usine de margarinerie de Cevital. Le transfert manuel ou par inertie, à l'origine de pertes de produits et de pénibilité pour le personnel, a été remplacé par un système automatisé fiable, sécurisé et autonome.

La solution adoptée repose sur l'utilisation de convoyeurs motorisés, d'élévateurs, de capteurs et d'un automate programmable industriel Siemens S7-1500, dont la programmation a été réalisée via le logiciel TIA Portal. La validation du fonctionnement a été assurée par une simulation virtuelle à l'aide de Factory I/O, permettant de tester l'ensemble du processus sans recourir à un déploiement physique préalable.

Les résultats obtenus ont démontré la faisabilité technique et fonctionnelle de la solution proposée, tout en mettant en évidence les avantages qu'offre l'automatisation dans le domaine industriel : amélioration de la sécurité, réduction des pertes, continuité du flux de production et fiabilité opérationnelle.

Ce projet constitue une application concrète des compétences développées en automatisation industrielle et témoigne de l'importance de la modélisation, de la programmation et de la simulation dans la conception de systèmes modernes performants. Il peut également servir de base à des extensions futures, notamment vers des systèmes supervisés (SCADA), des interconnexions multi-automates ou des améliorations en matière d'efficacité énergétique.

Références bibliographiques

[1] Document Cevital

[2] Equipe Cevital , « Stratégie de développement », <https://www.cevital.com/strategie-de-developpement>, juin 2025.

[3] Schneider Electric, « Enclosures and Motor Protection » , juin 2025.

[4] Equipe cybiant, « Définition de l'automatisation » , <https://www.cybiant.com>, juin 2025.

[5] M. Sandra, « Initiation à la programmation des API Siemens Via l'outil TIA PORTAL », mémoire de fin d'études, UFR de Sciences et Technologie d'Evry, Université Paris Saclay, 2024-2025.

[6] Equipe IP-System, « guide-ultime-de-lihm-interface-homme-machine » , <https://www.ip-systemes.com>, juin 2025.

[7] Equipe Eurotechconseil, « Automate programmable industriel » , <https://www.eurotechconseil.com/blog/>, juin 2025.

[8] B. SAMI, « Migration du système ESD de la station de pompage SP2 », mémoire de projet de fin d'étude, université de Bejaia, 2025.

[9] A. Fahem, « Automatisation et supervision d'une vanne régulatrice au sein de CEVITAL » , mémoire de fin d'études, université de Bejaia, 2024 .

[10] Equipe Siemens, « Définition et avantage de TIA Portal », <https://www.siemens.com/> , juin 2025.

Annexes

Annexe 1 : tables des variables.

CONVOYAGE LIGNE 4 ▶ PLC_2 [CPU 1511-1 PN] ▶ Variables API ▶ Table de variables standard [54]										
Variables Constantes utilisateur										
Table de variables standard										
	Nom	Type de données	Adresse	Réma...	Acces...	Écritu...	Visibl...	Surveilla...	Commentaire	
1	BP_START	Bool	%I0.0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>			
2	ORDRE_MARCHE	Bool	%M0.0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>			
3	BOUTON_ARRET_URGENCE	Bool	%I0.2		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>			
4	BP_STOP	Bool	%I0.1		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>			
5	VOYANT_START	Bool	%Q0.0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>			
6	VOYANT_STOP	Bool	%Q0.1		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>			
7	<Ajouter>				<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>			

CONVOYAGE LIGNE 4 ▶ PLC_2 [CPU 1511-1 PN] ▶ Variables API ▶ Table de variables standard_2 [8]										
Variables Constantes utilisateur										
Table de variables standard_2										
	Nom	Type de données	Adresse	Réma...	Acces...	Écritu...	Visibl...	Surveilla...	Commentaire	
1	CAPTEUR_ARRIVEE_CARTON	Bool	%I0.3		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>			
2	MOTEUR_M1	Bool	%Q0.2		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>			
3	POUSSEUR_P1	Bool	%Q0.3		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>			
4	TEMPO_ARRIVEE_CARTON	Timer	%T0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>			
5	CAPTEUR_ARRIVEE_P1	Bool	%I0.4		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>			
6	CAPTEUR_P1_AVANT	Bool	%I0.5		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>			
7	CAPTEUR_P1_ARRIERE	Bool	%I0.6		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>			
8	RETARD_P1	Timer	%T1		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>			
9	<Ajouter>				<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>			

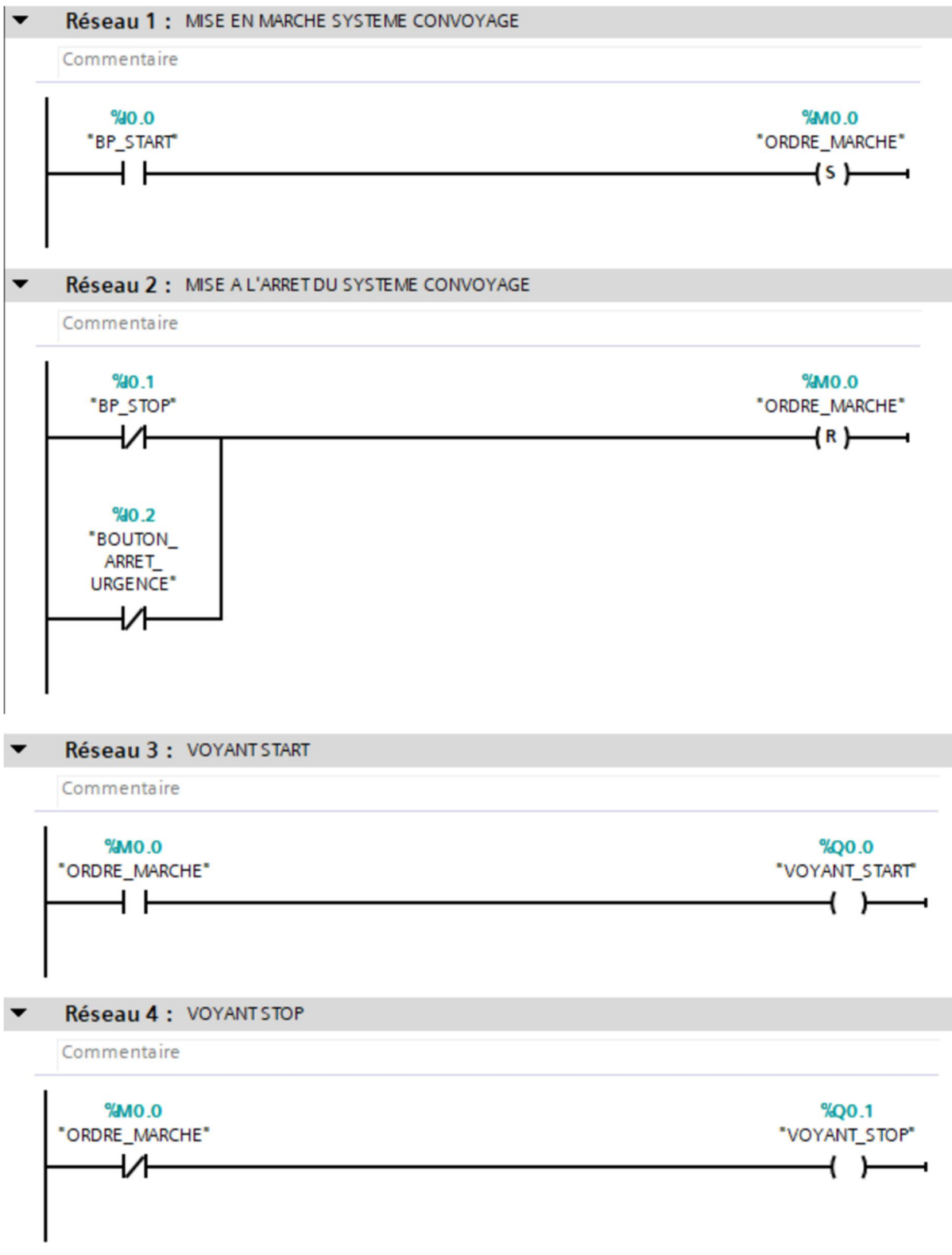
CONVOYAGE LIGNE 4 ▶ PLC_2 [CPU 1511-1 PN] ▶ Variables API ▶ Standard-Variablentabelle [121]										
Variables Constantes utilisateur Constantes système										
Standard-Variablentabelle										
	Nom	Type de données	Adresse	Réma...	Acces...	Écritu...	Visibl...	Surveilla...	Commentaire	
1	FDC_DESCENSEUR_I_UP	Bool	%I2.0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>			
2	DESCENSEUR_I_UP	Bool	%Q2.0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>			
3	FDC_RALENTIS_I_UP	Bool	%I2.1		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>			
4	RALENTISSEMENT_I	Bool	%Q2.1		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>			
5	ROULEAUX_IN_I	Bool	%Q2.2		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>			
6	FDC_RALENTIS_I_DOWN	Bool	%I2.3		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>			
7	LIMITE_DROITE_DESC_I	Bool	%I2.2		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>			
8	DESCENSEUR_DOWN_I	Bool	%Q2.3		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>			
9	FDC_DESCENSEUR_I_DOWN	Bool	%I2.4		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>			
10	ROULEAUX_OUT_I	Bool	%Q2.4		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>			
11	CAPTEUR_ARRIVEE_M2	Bool	%I0.7		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>			
12	POUSSEUR_P2	Bool	%Q0.5		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>			
13	MOTEUR_M2	Bool	%Q0.4		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>			
14	TEMPO_ARRIVEE_CARTON_M2	Timer	%T2		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>			
15	POUSSEUR_P2_ARRIERE	Bool	%I1.1		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>			
16	CAPTEUR_ARRIVEE_P2	Bool	%I1.0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>			
17	RETARD_P2	Timer	%T3		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>			
18	POUSSEUR_P2_AVANT	Bool	%I1.2		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>			
19	CAPTEUR_ARRIVEE_M3	Bool	%I1.3		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>			
20	MOTEUR_M3	Bool	%Q0.6		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>			
21	TEMPO_MOTEUR_M3	Timer	%T4		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>			

CONVOYAGE LIGNE 4 ▸ PLC_2 [CPU 1511-1 PN] ▸ Variables API ▸ Standard-Variablentabelle [121]									
Variables									
Standard-Variablentabelle									
	Nom	Type de données	Adresse	Réma...	Acces...	Écritu...	Visibl...	Surveilla...	Commentaire
22	RETARD_DEMMARAGE_M4	Timer	%T5	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
23	MOTEUR_M4(1)	Bool	%Q0.7	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
24	CAPTEUR_M3_OUT	Bool	%I1.5	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
25	CAPTEUR_ARRIVEE_M4	Bool	%I1.4	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
26	LIMITE_GAUCHE	Bool	%I2.5	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
27	COMPTEUR_I	Int	%MW10	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
28	CHARGEMENT_I_OK	Bool	%M8.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
29	FRONT_DESCENDANT_FDC_UP_I	Bool	%M9.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
30	RETARD_CHARGEMENT_I	Timer	%T6	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
31	DECHARGEMENT_I_OK	Bool	%M8.1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
32	RETARD_DECHARGEMENT_I	Timer	%T7	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
33	FRONT_DESCENDANT_FDC_I_D...	Bool	%M9.1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
34	CAPTEUR_M4_IN	Bool	%I1.6	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
35	COMPTEUR_2	Bool	%I1.7	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
36	CAPTEUR_ARRIVEE_M5	Bool	%I2.6	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
37	MOTEUR_M5	Bool	%Q1.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
38	TEMPO_ARRIVEE_CARTON_M5	Timer	%T8	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
39	CAPTEUR_ARRIVEE_M6	Bool	%I2.7	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
40	MOTEUR_M6	Bool	%Q1.1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
41	TEMPO_ARRIVEE_CARTON_M6	Timer	%T9	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
42	RETARD_DEMMARAGE_M7	Timer	%T10	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		

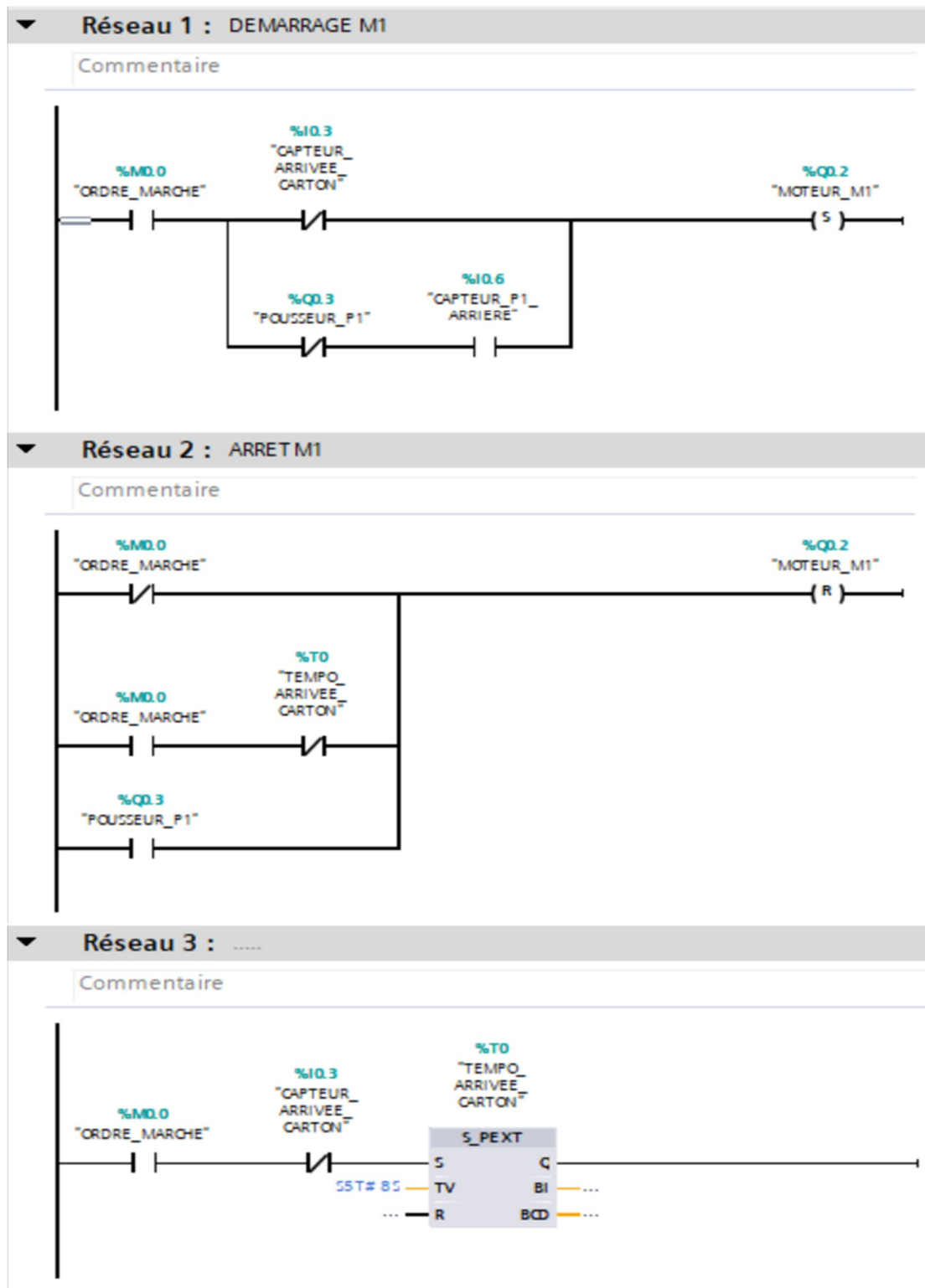
CONVOYAGE LIGNE 4 ▸ PLC_2 [CPU 1511-1 PN] ▸ Variables API ▸ Standard-Variablentabelle [121]									
Variables									
Standard-Variablentabelle									
	Nom	Type de données	Adresse	Réma...	Acces...	Écritu...	Visibl...	Surveilla...	Commentaire
43	MOTEUR_M7	Bool	%Q1.2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
44	CAPTEUR_ARRIVEE_M7	Bool	%I3.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
45	FDC_DESCENSEUR_I_UP	Bool	%I3.1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
46	LIMITE_GAUCHE_II	Bool	%I3.6	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
47	FDC_RALENTIS_I_UP	Bool	%I3.2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
48	FDC_DESCENSEUR_I_DOWN	Bool	%I3.4	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
49	FDC_RALENTIS_I_DOWN	Bool	%I3.3	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
50	LIMITE_DROITE_DESC_II	Bool	%I3.5	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
51	DESCENSEUR_I_UP	Bool	%Q2.5	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
52	DESCENSEUR_DOWN_II	Bool	%Q3.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
53	RALENTISSEMENT_II	Bool	%Q2.6	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
54	ROULEAUX_IN_II	Bool	%Q2.7	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
55	ROULEAUX_OUT_II	Bool	%Q3.1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
56	COMPTEUR_II	Int	%MW12	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
57	FRONT_DESCENDANT_FDC_UP_II	Bool	%M9.2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
58	RETARD_CHARGEMENT_II	Timer	%T11	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
59	CHARGEMENT_II_OK	Bool	%M8.2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
60	RETARD_DECHARGEMENT_II	Timer	%T12	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
61	DECHARGEMENT_II_OK	Bool	%M8.3	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
62	FRONT_DESCENDANT_FDC_II_D...	Bool	%M9.3	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
63	Tag_1	Bool	%M9.7	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		

CONVOYAGE LIGNE 4 ▸ PLC_2 [CPU 1511-1 PN] ▸ Variables API ▸ Standard-Variablentabelle [121]									
Variables									
Standard-Variablentabelle									
	Nom	Type de données	Adresse	Réma...	Acces...	Écritu...	Visibl...	Surveilla...	Commentaire
63	Tag_1	Bool	%M9.7	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
64	Tag_2	Int	%QW50	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
65	Tag_3	Int	%QW52	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
66	Tag_4	Int	%QW48	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
67	Tag_5	Bool	%Q1.7	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
68	CAPTEUR_ARRIVEE_M8	Bool	%I3.7	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
69	MOTEUR_M8	Bool	%Q1.3	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
70	TEMPO_ARRIVEE_CARTON_M8	Timer	%T13	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
71	Tag_6	Int	%QW54	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
72	<Ajouter>			<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		

Annexe 2 : mise en marche.



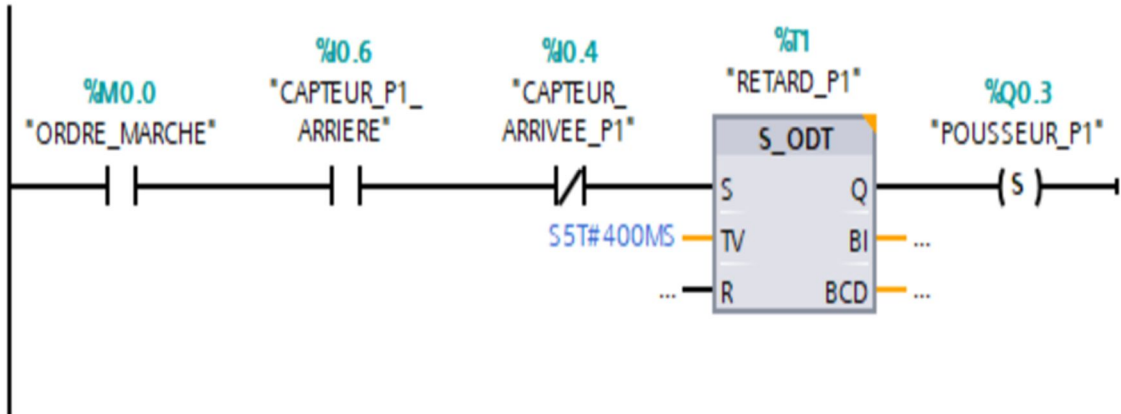
Annexe 3 : commande moteur 1



Annexe 4 : pousseur 1

▼ Réseau 1 : POUSSEUR P1 AVANT

Commentaire

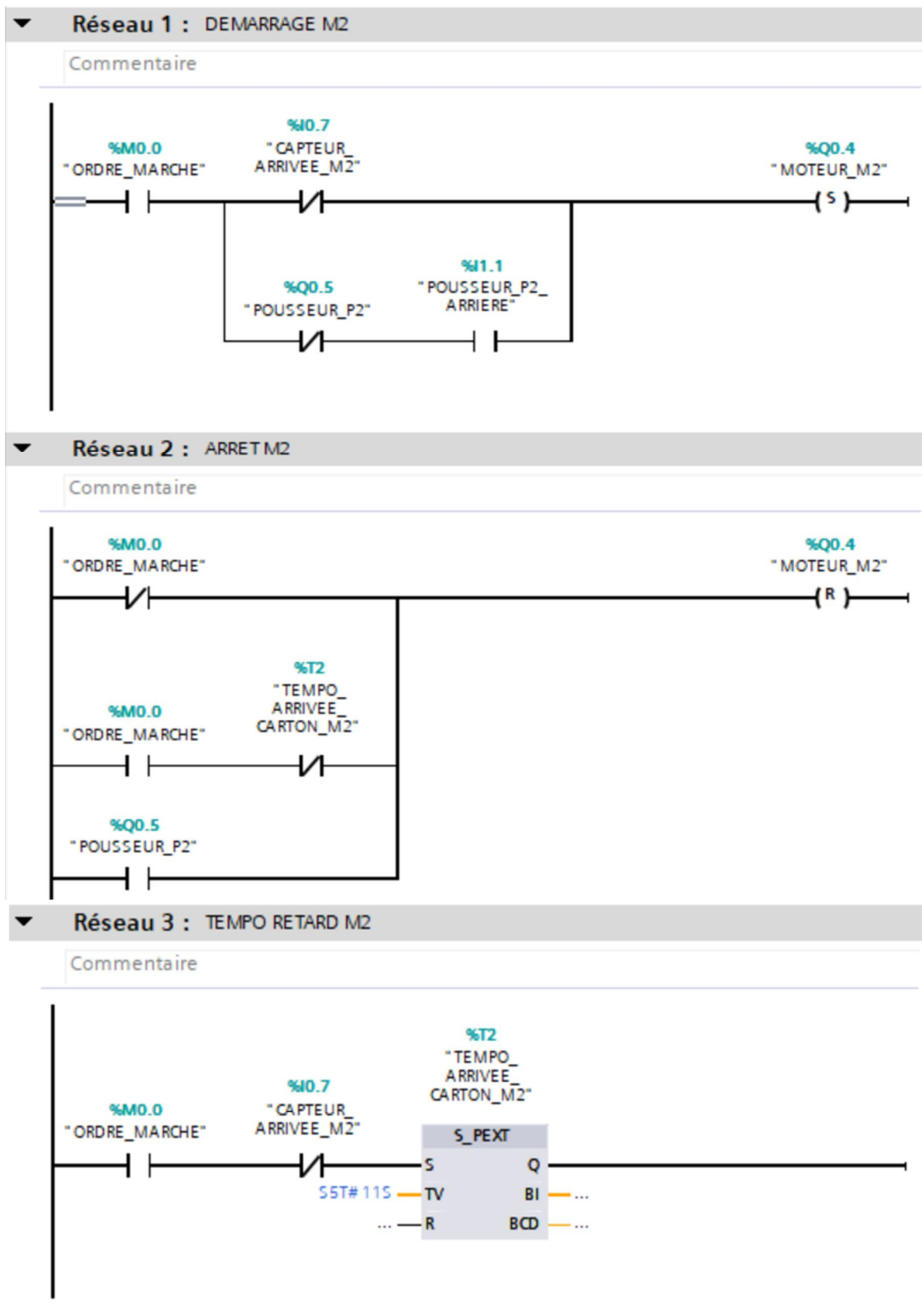


▼ Réseau 2 : POUSSEUR P1 ARRIERE

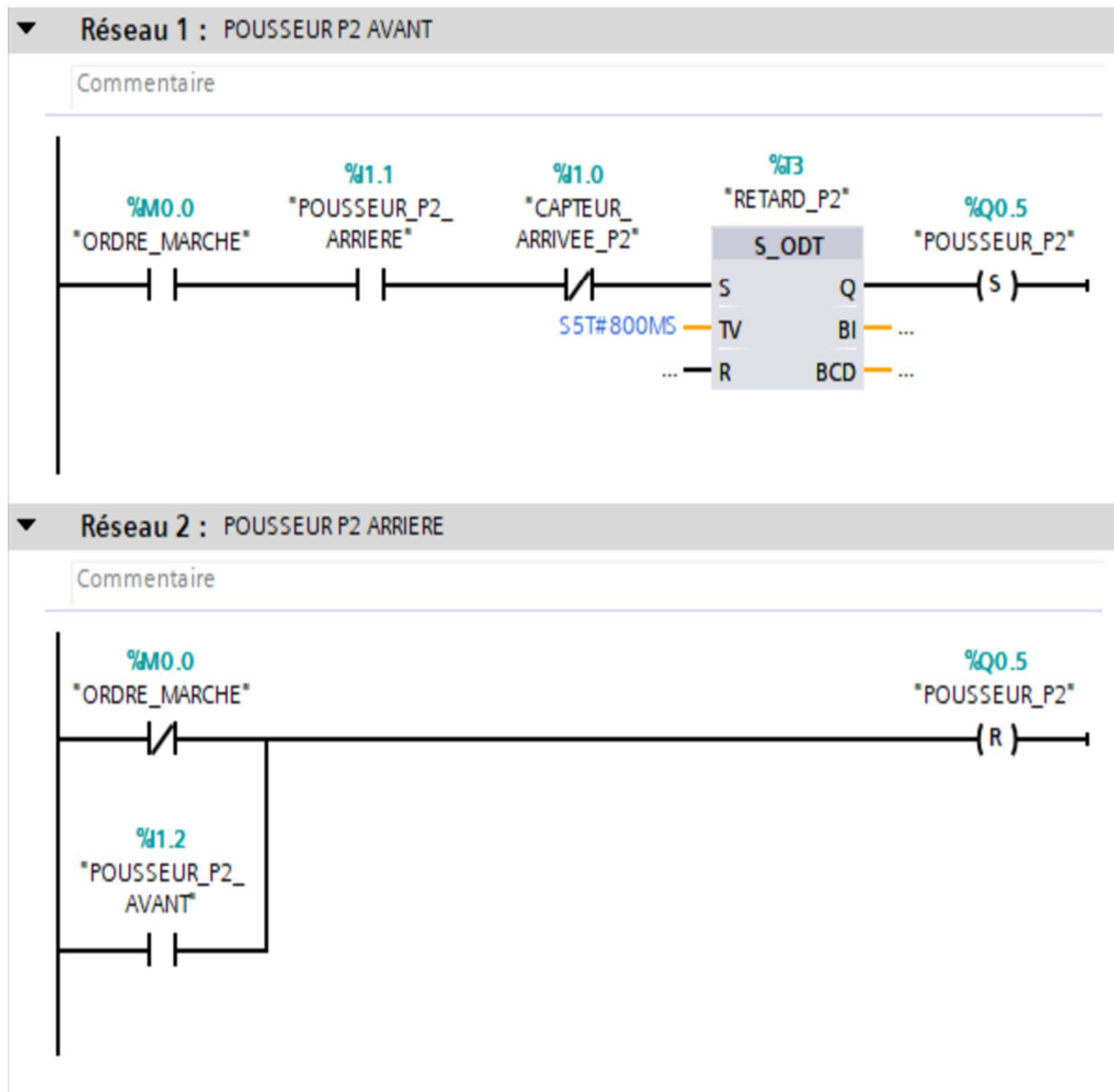
Commentaire



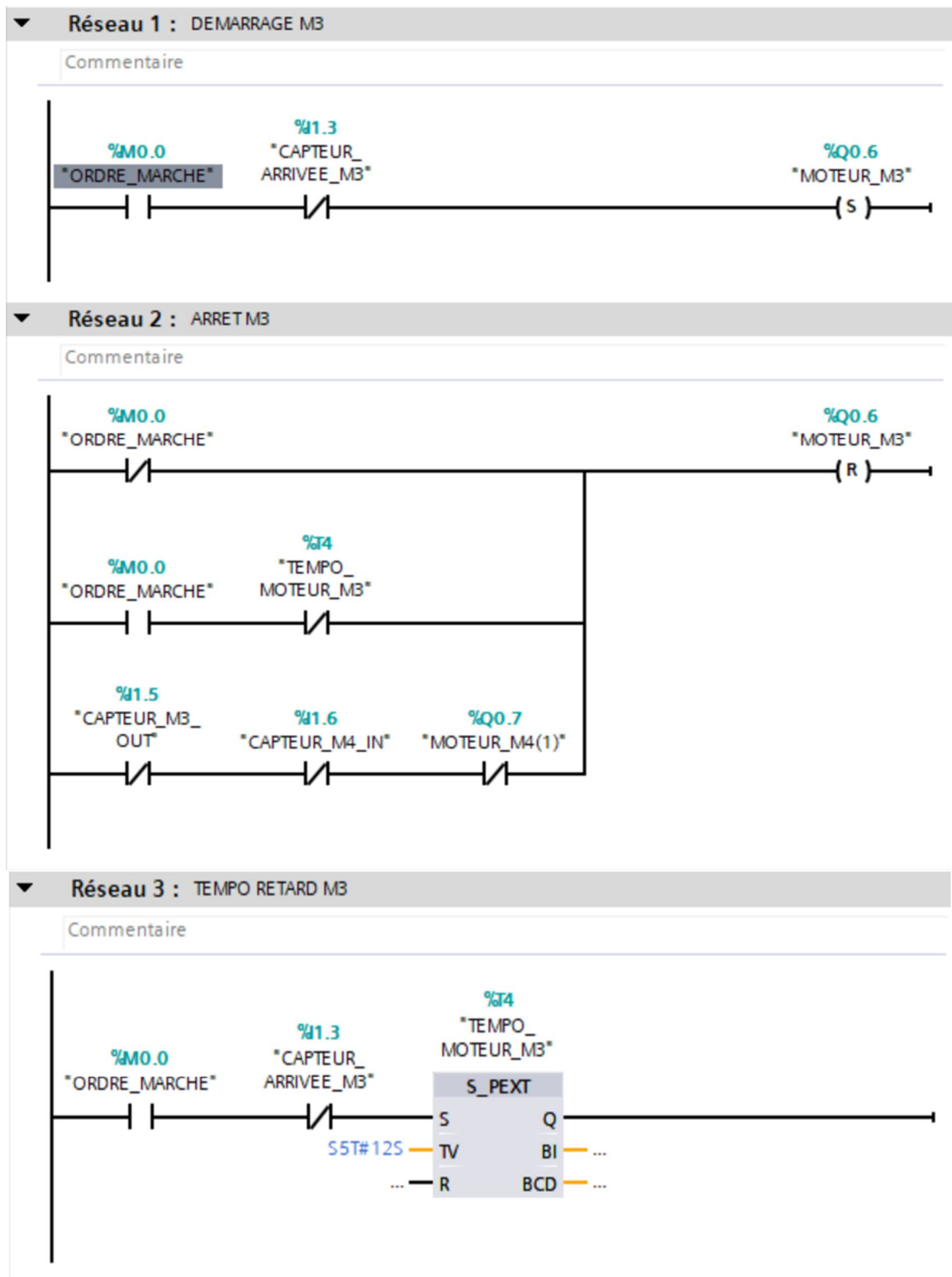
Annexe 5 : commande M2.



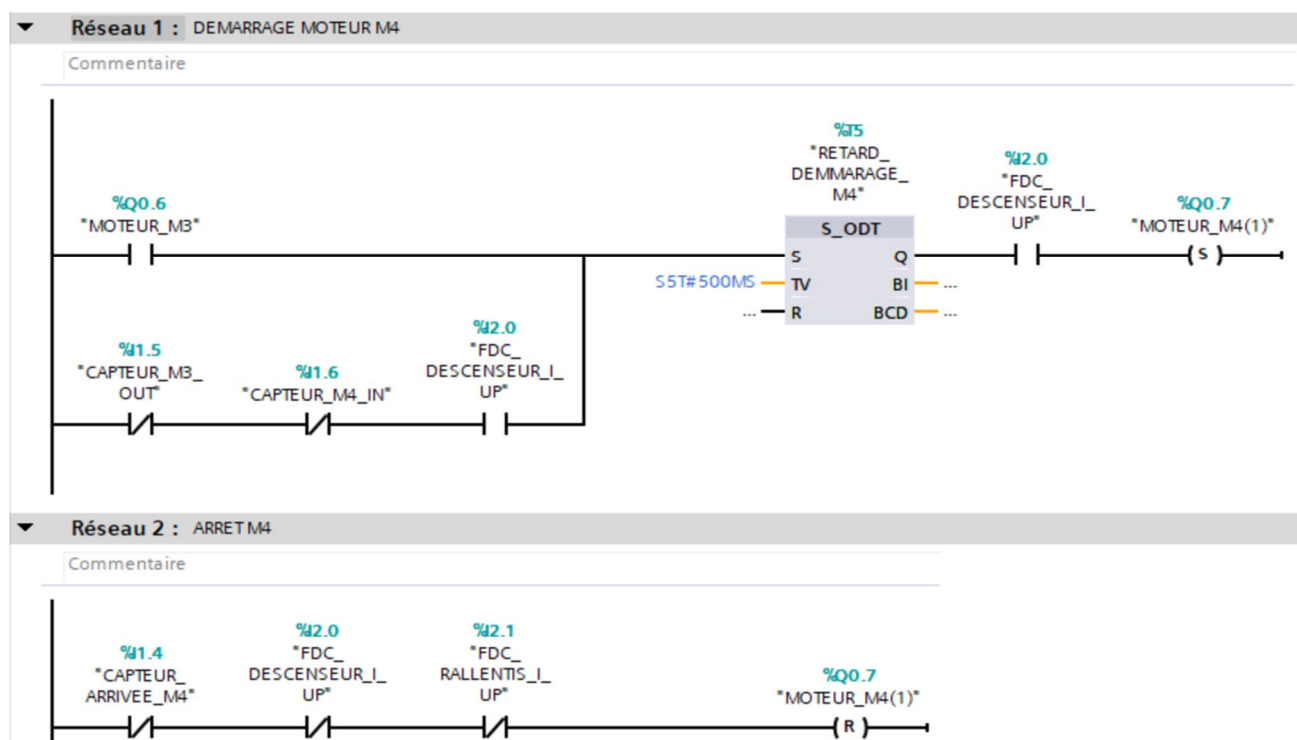
Annexe 6 : commande pousseur 2 .



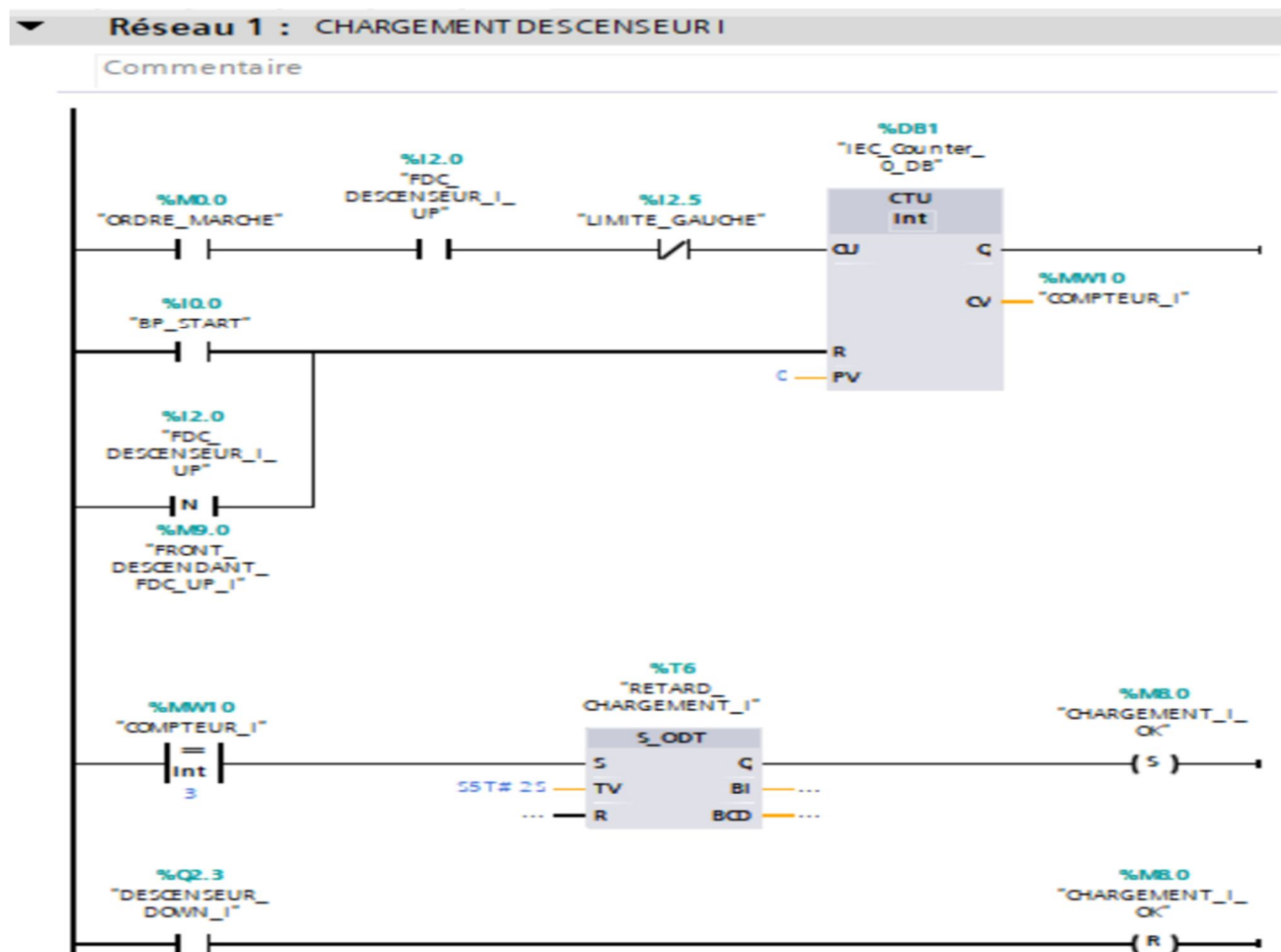
Annexe 7 : commande M3.



Annexe 8 : commande M4.

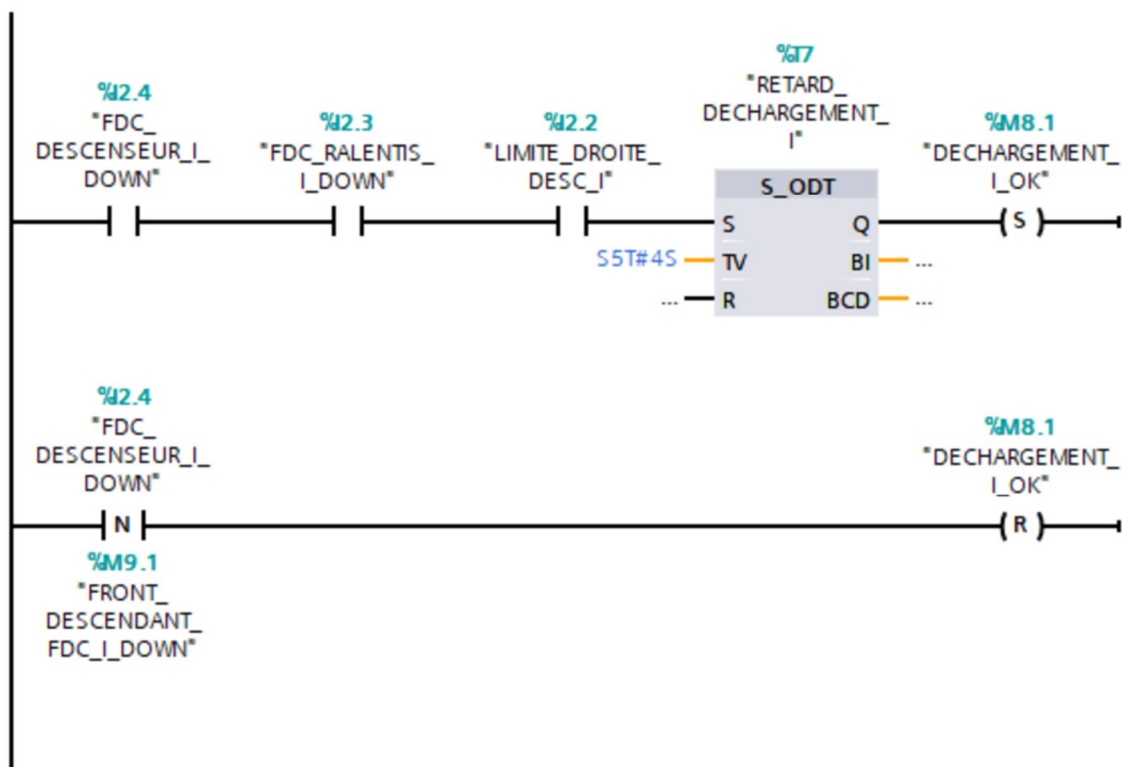


Annex 9 : descenseur 1.



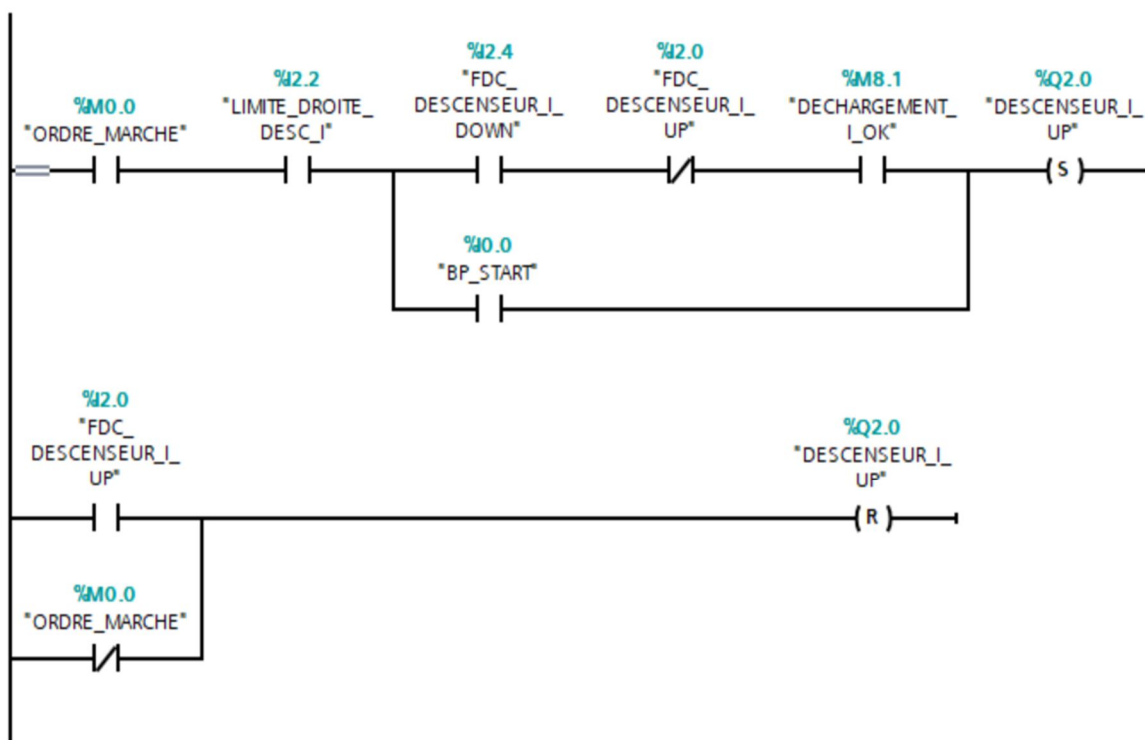
▼ Réseau 2 : DECHARGEMENT DESCENSEUR I

Commentaire



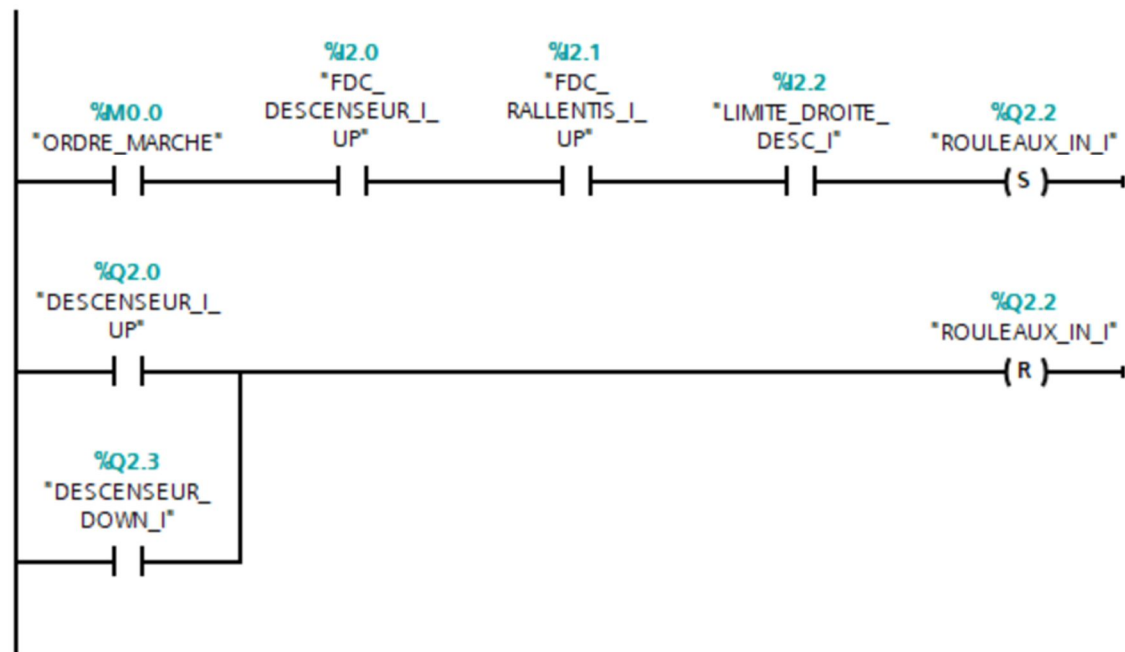
▼ Réseau 3 : DESCENSEUR I UP

Commentaire



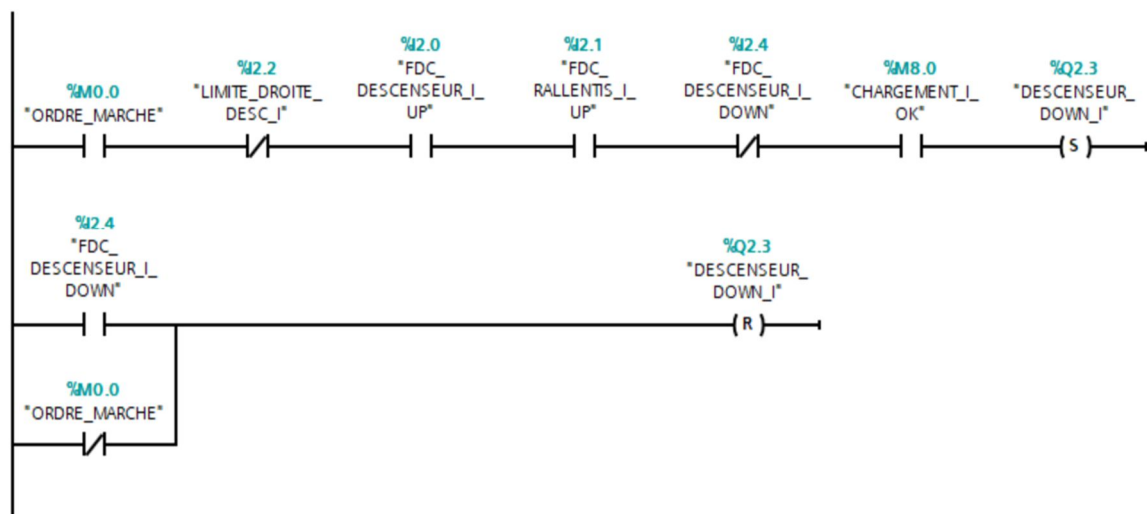
▼ Réseau 4 : ROULEAUX IN I

Commentaire



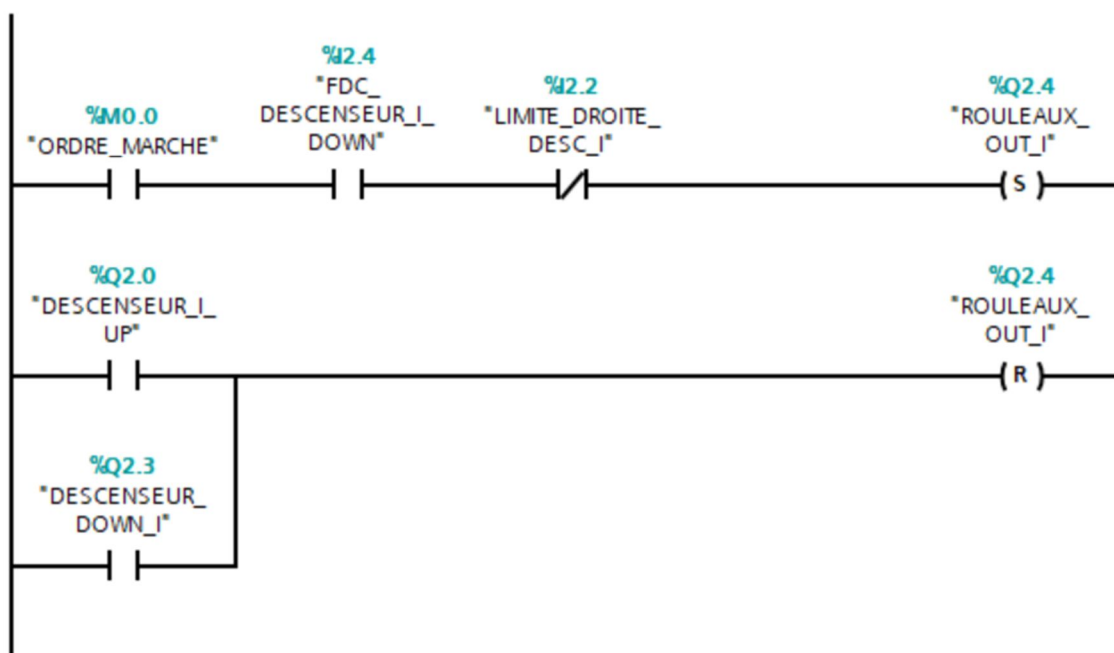
▼ Réseau 5 : DESCENSEUR I DOWN

Commentaire



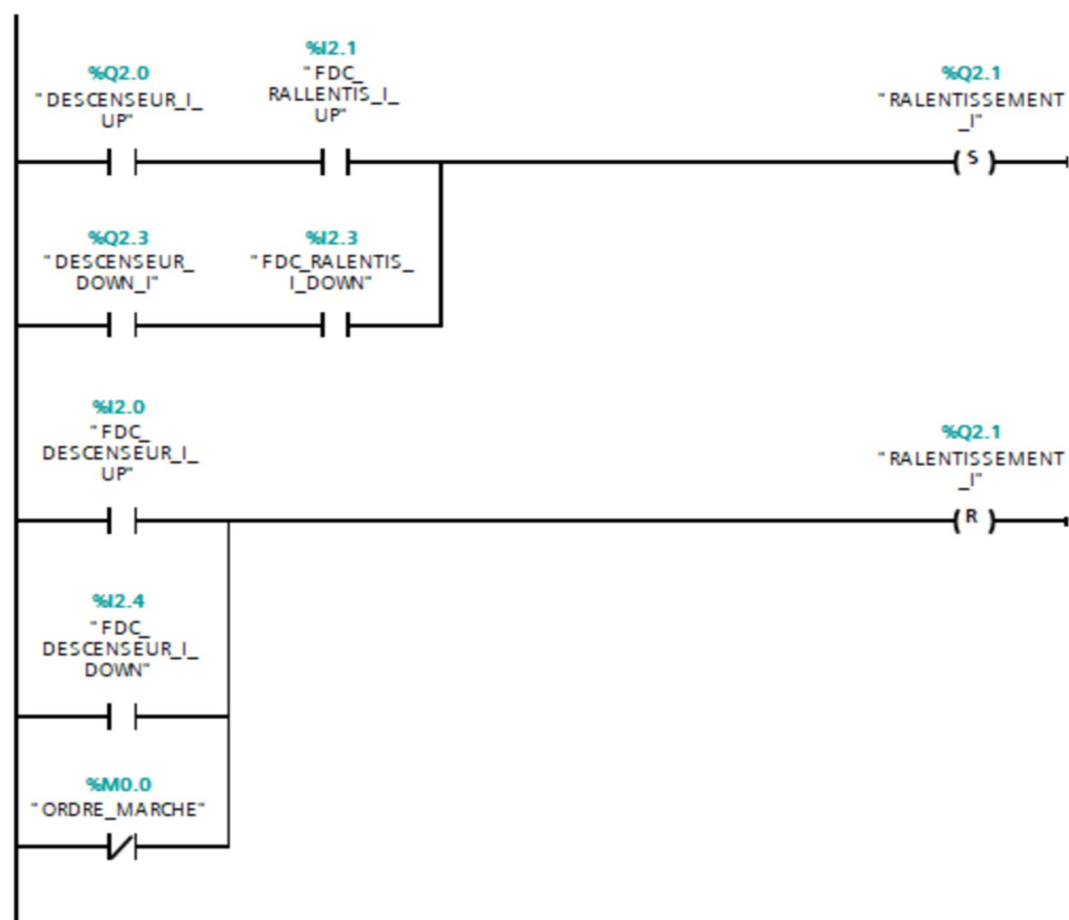
▼ Réseau 6 : ROULEAUX OUT I

Commentaire



▼ Réseau 7 : RALENTISSEMENT I

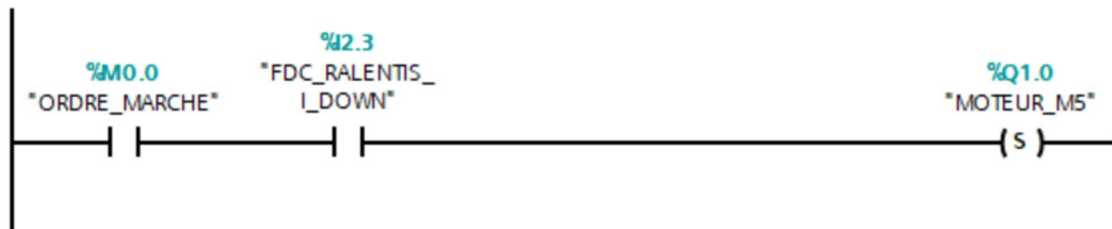
Commentaire



Annexe 10 : commande M5.

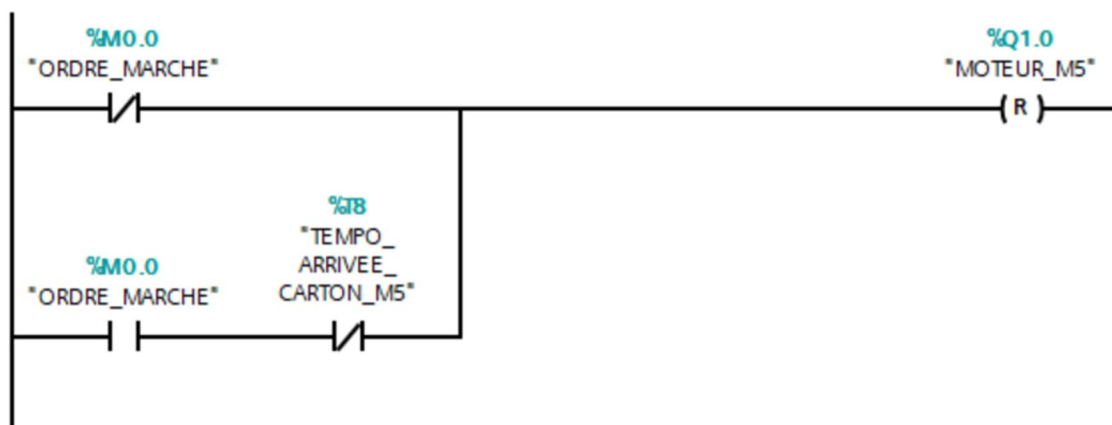
▼ Réseau 1 : DEMARRAGE M5

Commentaire



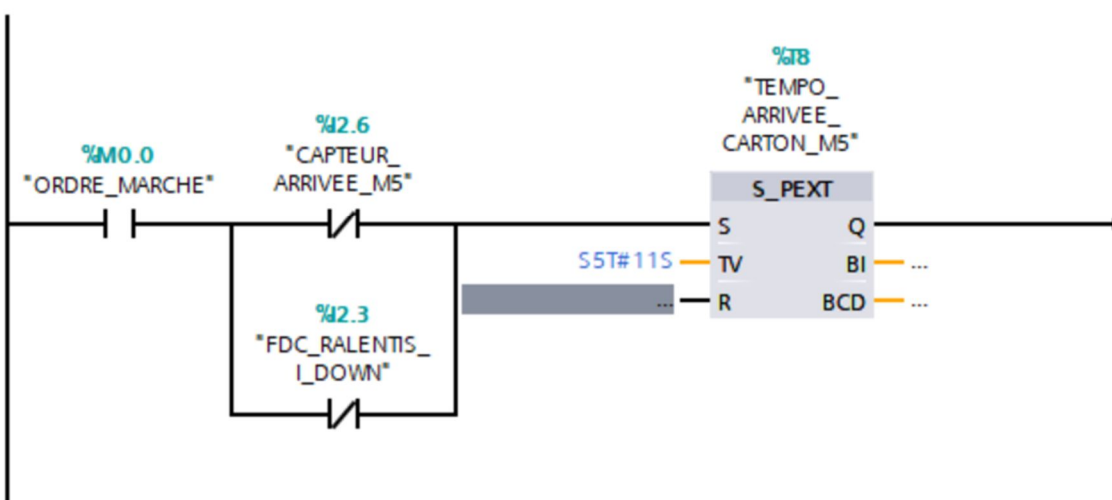
▼ Réseau 2 : ARRET M5

Commentaire

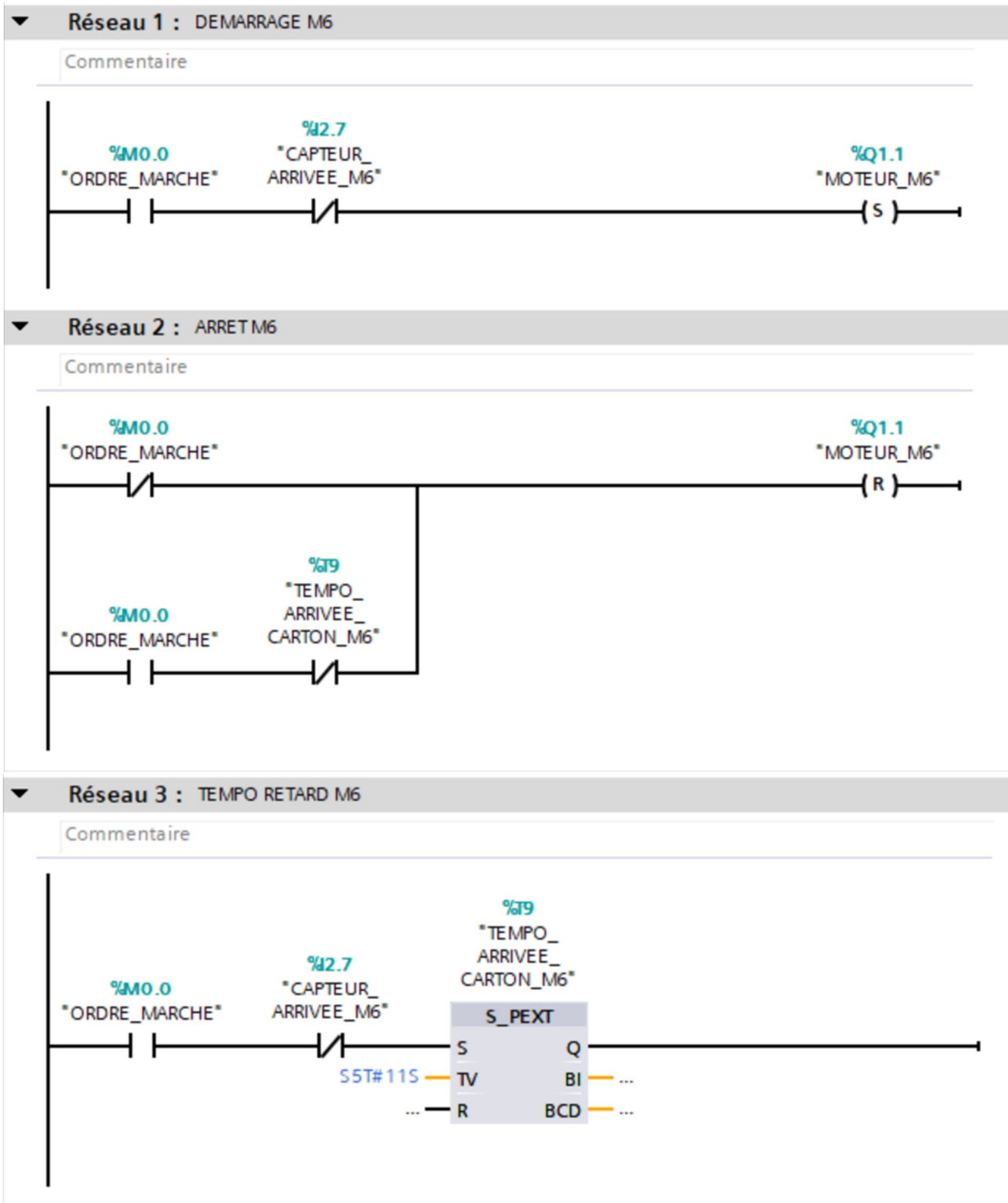


▼ Réseau 3 : TEMPO RETARD M5

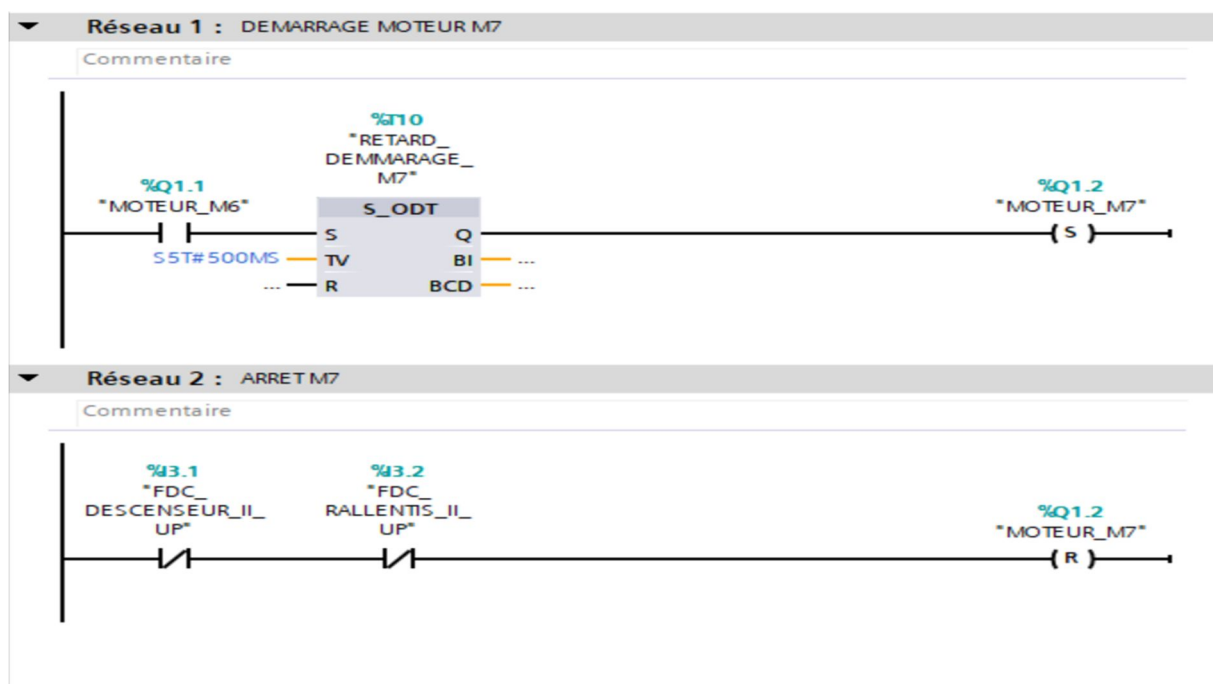
Commentaire



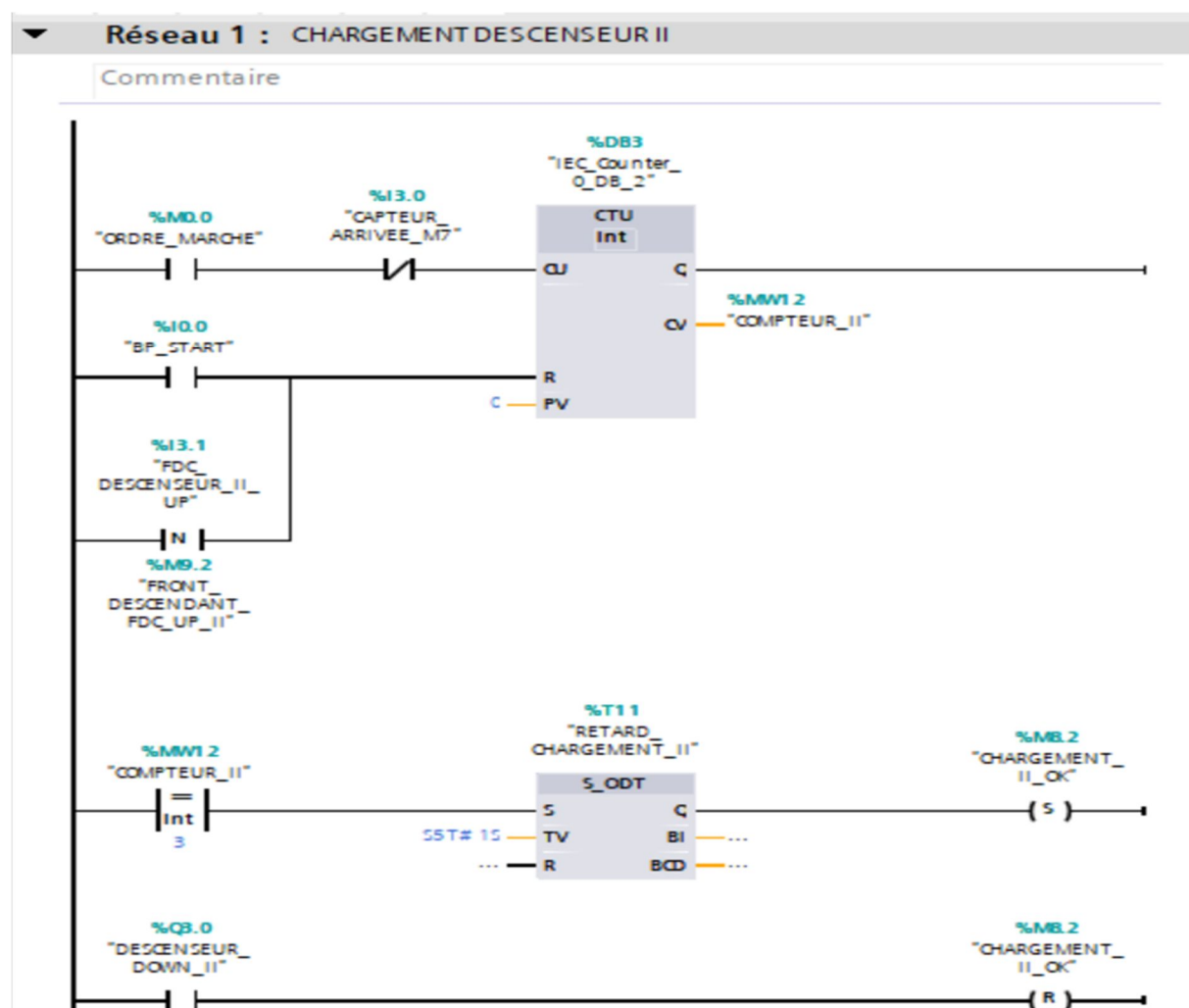
Annexe 11 : commande M6.



Annexe 12 : commande M7.

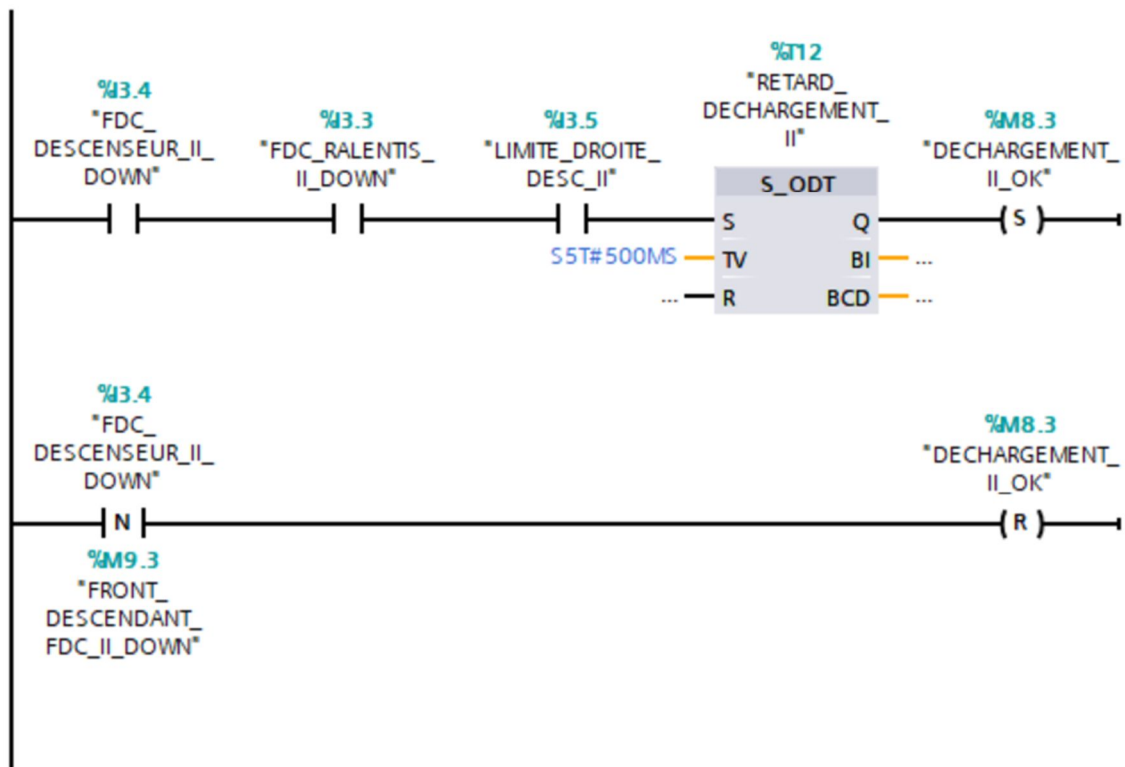


Annex 13 : descenseur 2.



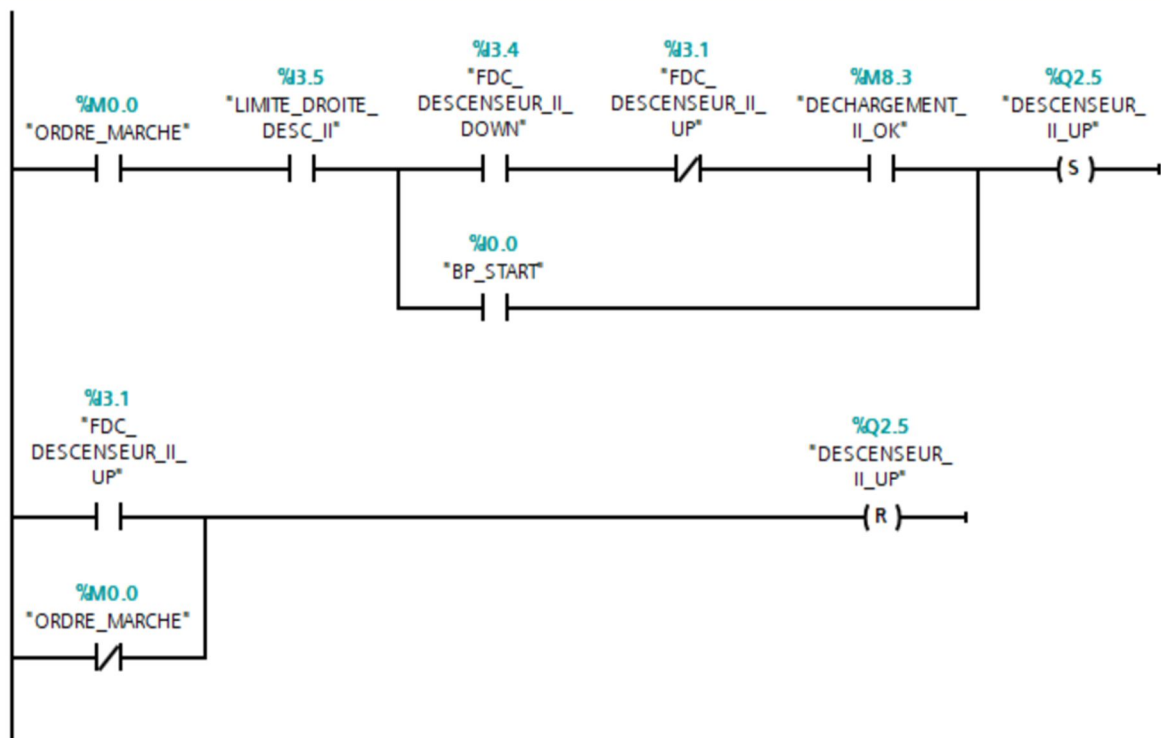
▼ Réseau 2 : DECHARGEMENT DESCENSEUR II

Commentaire



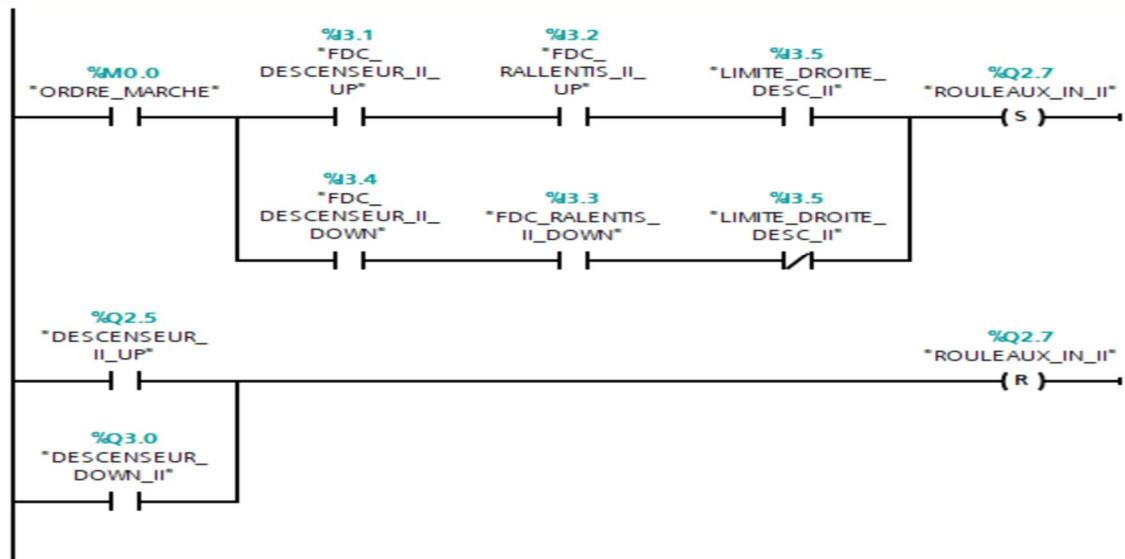
▼ Réseau 3 : DESCENSEUR II UP

Commentaire



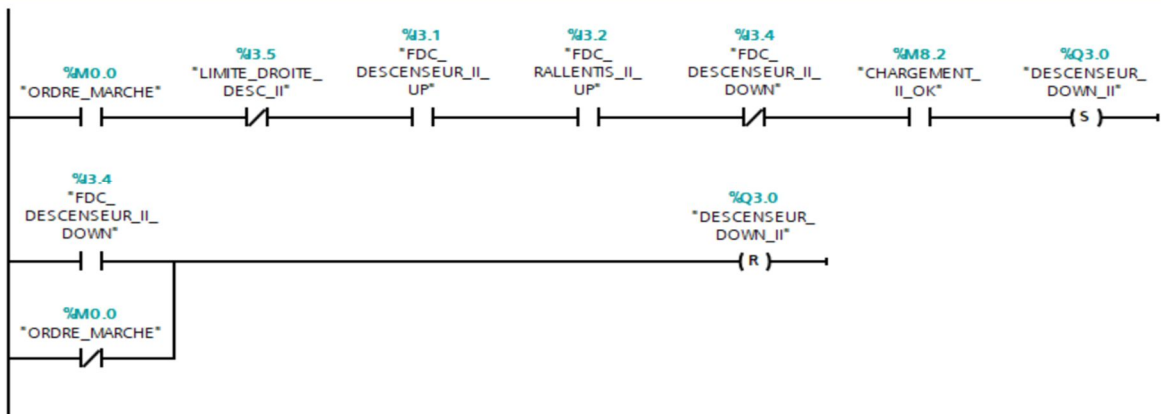
▼ Réseau 4 : ROULEAUX IN II

Commentaire



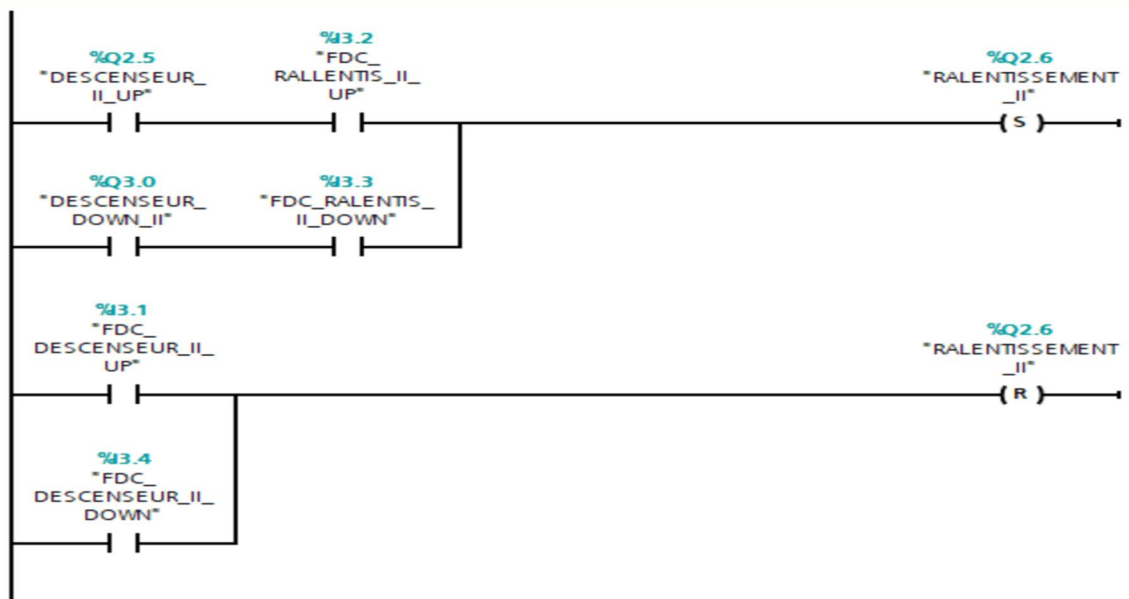
▼ Réseau 5 : DESCENSEUR II DOWN

Commentaire

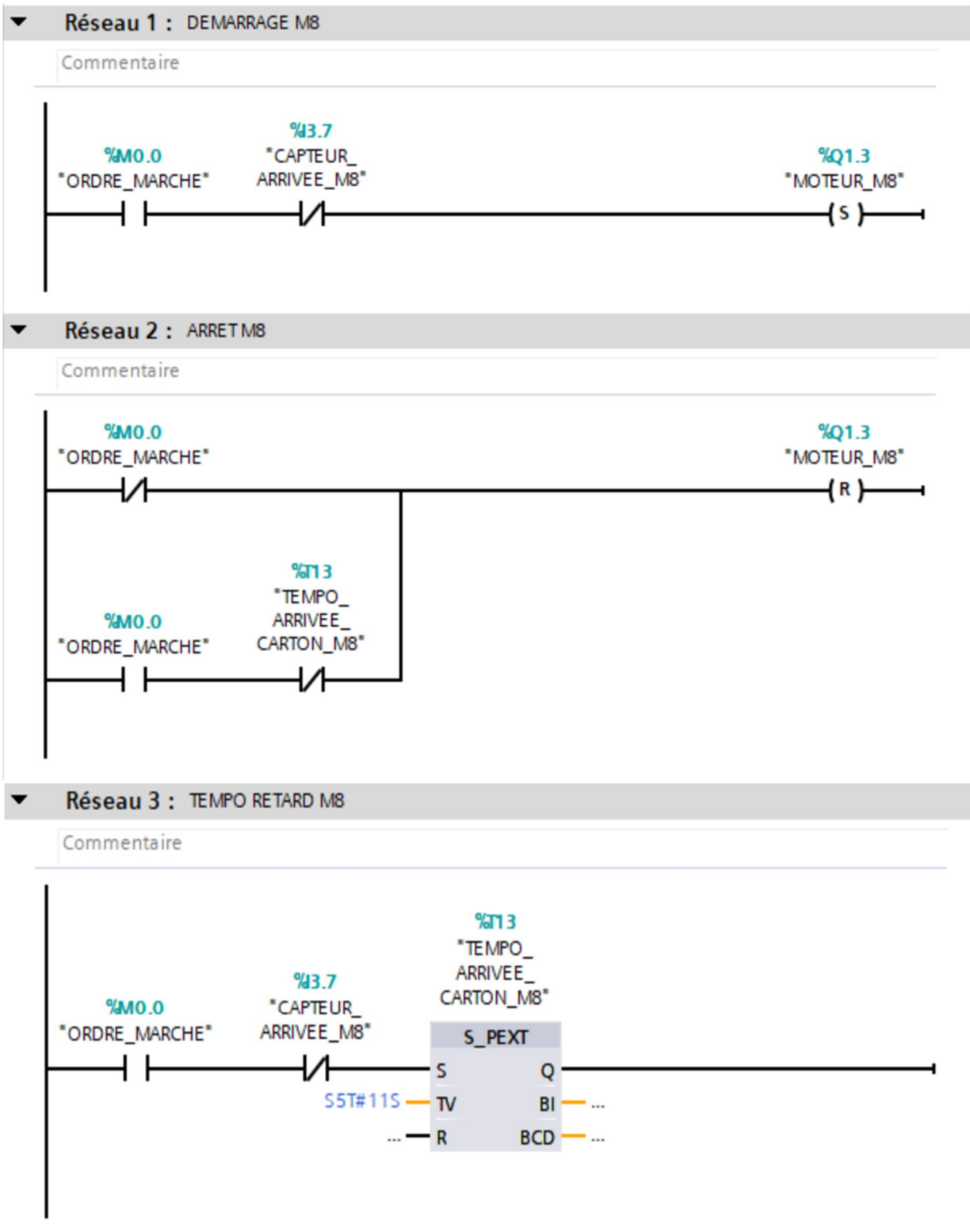


▼ Réseau 6 : RALENTISSEMENT II

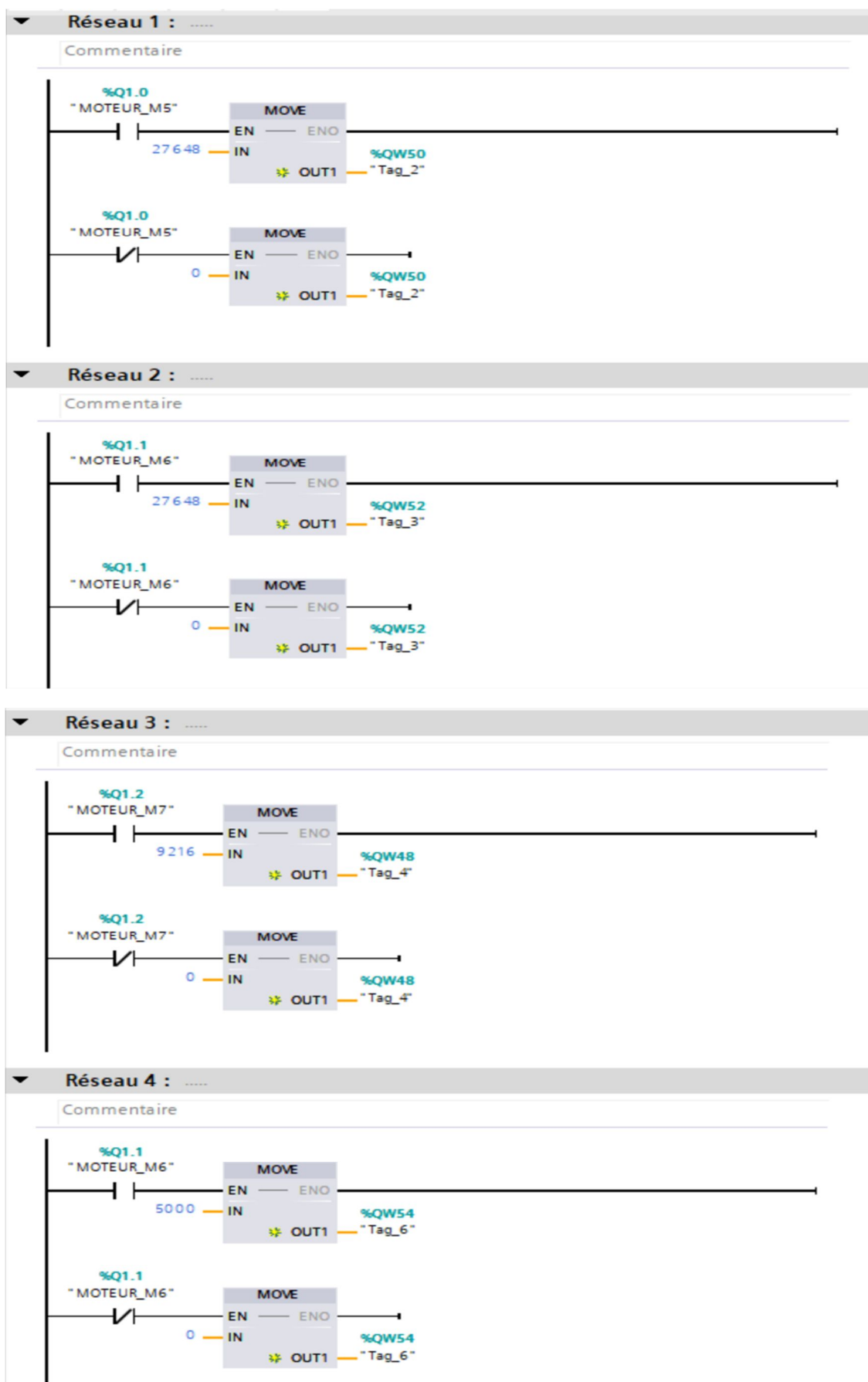
Commentaire



Annexe 14 : commande M8.



Annexe 15 : commande FC12.



Annexe 16 : commande function-S7-1500.



```
2  #Value:=PEEK(area := 16#82,  
3      dbNumber := 0,  
4      byteOffset := 511);  
5  #Value := #Value + 1;  
6  
7  POKE(area := 16#82,  
8      dbNumber := 0,  
9      byteOffset := 511,  
10     value := #Value);  
11  
12 POKE(area := 16#82,  
13     dbNumber := 0,  
14     byteOffset := 1016,  
15     value := #Value_01_DW);  
16 POKE(area := 16#82,  
17     dbNumber := 0,  
18     byteOffset := 1020,  
19     value := #Value_02_DW);  
20  
21 FOR #ForCounter := 0 TO 63 DO  
22     #Value:=PEEK(area := 16#1,  
23         dbNumber := 0,  
24         byteOffset := #ForCounter);  
25     POKE(area := 16#81,  
26         dbNumber := 0,  
27         byteOffset := #ForCounter,  
28         value := #Value);  
29 END_FOR;  
30 #Value := PEEK(area := 16#1,  
31     dbNumber := 0,  
32     byteOffset := 512);  
33 POKE(area := 16#82,  
34     dbNumber := 0,  
35     byteOffset := 512,  
36     value := #Value);
```

Résumé :

Ce mémoire traite de la conception, de la programmation et de la simulation d'un système de convoyage automatisé, destiné à améliorer le transfert de cartons au sein de l'unité de conditionnement de margarine de l'entreprise Cevital. Le système initial, basé sur un déplacement par inertie, provoquait des collisions entre cartons, des pertes de produits et imposait une tâche pénible aux opérateurs.

Afin de résoudre ces problèmes, une solution automatisée a été développée à l'aide du logiciel TIA Portal pour la programmation d'un API Siemens S7-1500, et du logiciel Factory I/O pour la simulation 3D du système. Le projet intègre des convoyeurs motorisés, des élévateurs, des capteurs de détection et des actionneurs, formant un système autonome et sécurisé.

Les résultats de la simulation ont confirmé la fiabilité du système conçu et sa capacité à répondre aux exigences de continuité, de sécurité et d'efficacité du processus industriel. Ce travail illustre l'intérêt des outils modernes d'automatisation et de simulation dans l'amélioration des performances industrielles.

المخلص

يتناول هذا المذكرة تصميم وبرمجة ومحاكاة نظام نقل آلي، يهدف إلى تحسين عملية نقل الصناديق داخل وحدة تعبئة السمن التابعة لشركة كان النظام السابق يعتمد على الحركة بالقصور الذاتي، مما تسبب في تصادم الصناديق، وفقدان المنتجات، وأدى إلى مهام مرهقة. سيفيتال للعمال.

، Siemens S7-1500 لبرمجة جهاز التحكم المنطقي المبرمج TIA Portal ولحل هذه المشاكل، تم تطوير حل آلي باستخدام برنامج لمحاكاة النظام في بيئة ثلاثية الأبعاد. يتضمن المشروع سيورًا ناقلة مزودة بمحركات، ورافعات، Factory I/O بالإضافة إلى برنامج وحساسات كشف، ومشغلات، تشكل نظامًا ذاتيًا وآمنًا.

أكدت نتائج المحاكاة موثوقية النظام المصمم وقدرته على تحقيق متطلبات الاستمرارية، والسلامة، والكفاءة في العملية الصناعية. ويوضح هذا العمل أهمية أدوات الأتمتة والمحاكاة الحديثة في تحسين الأداء الصناعي.

Abstract:

This thesis focuses on the design, programming, and simulation of an automated conveyor system aimed at improving the transport of boxes within the margarine packaging unit of the company Cevital. The previous system relied on inertial motion, which caused collisions between boxes, product loss, and resulted in physically demanding tasks for operators.

To solve these issues, an automated PLC solution was developed using the TIA Portal software to program the Siemens S7-1500 PLC, and the Factory I/O software to simulate the system in a 3D virtual environment.

The project includes motorized conveyors, elevators, detection sensors, and actuators that form a fully automated and safe system.

Simulation results confirmed the reliability of the designed system and its ability to meet the requirements of continuity, safety, and efficiency in the industrial process. This work highlights the importance of modern automation and simulation tools in enhancing industrial performance.

