

الجمهورية الجزائرية الديمقراطية الشعبية

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

وزارة التعليم العالي والبحث العلمي

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

جامعة أبي بكر بلقايد - تلمسان

Université Aboubakr Belkaïd – Tlemcen –

Faculté de TECHNOLOGIE



MEMOIRE

Présenté pour l'obtention du **diplôme** de **MASTER**

En : Electrotechnique

Spécialité : Commandes Electriques

Par : KHOLKHAL Oussama

Sujet

Commande d'un robot cartésien par le biais d'un API

Soutenu publiquement, le 06 / 06 / 2024, devant le jury composé de :

Mme K. GHLAM	MCA	Université de Tlemcen	Président
Mr M. A BRIXI NIGASSA	MCB	Université de Tlemcen	Examinateur
Mr S. M. MELIANI	Pr	Université de Tlemcen	Encadreur

Année universitaire : 2023 / 2024



Dédicace

Je dédie ce mémoire A mes chers parents
Ma Mère Et mon Père pour leur patience, leur amour,
Leur soutien et leurs encouragements.

A mes frères.

A mes amis et mes camarades. Sans oublier tout
Les professeurs que ce soit du primaire, du moyen,
Du secondaire et de l'enseignement supérieur.

Remerciement

Je tiens à remercier d'abord Dieu le tout puissant qui je a donné la force et la patience pour accomplir ce modeste travail.

Au terme de ce modeste travail, je tiens à exprimer mes vifs remerciements à mon encadreur Mr MELIANI Sidi Mohammed pour sa disponibilité, son aide, son orientation et ses conseils précieux.

Je tiens aussi à remercier les membres de jury d'avoir accepté de juger mon travail.

Sans oublier de remercier tous les membres de ma familles respectives pour leurs encouragements particulièrement mes chers parents.

En fin, je tiens également à remercier toutes les personnes qui ont participé de près ou de loin à la réalisation de ce travail.

Table des matières

Table des matières	i
Liste des Figures	iv
Nomenclature	vi
Introduction Générale	1
I. Chapitre I : Description des systèmes de stockage / Déstockage (AS/RS)	2
I.1 Introduction	2
I.2 Industrie 4.0	2
I.2.1 Qu'est-ce que « L'Industrie 4.0 » ?.....	2
I.2.2 Comment fonctionne une usine 4.0 ?.....	2
I.3 Systèmes stockage/déstockage AS/RS.....	3
I.3.1 Définition	3
I.3.2 Fonctionnement	3
I.3.3 Systèmes de gestion d'entrepôt intelligents.....	3
I.3.4 Composition.....	5
I.3.5 Avantages	5
I.3.6 Inconvénients	6
I.3.7 Types AS/RS	6
I.4 Différents robots	8
I.4.1 Robots manipulateurs (ou robots articulés).....	8
I.4.2 Robots SCARA	8
I.4.3 Robots cartésiens	9
I.4.4 Robots collaboratifs.....	9
I.5 Description de notre prototype	10
I.6 MCC et sa régulation de position.....	11
I.6.1 Photo d'un MCC.....	11
I.6.2 Utilisation.....	11
I.6.3 Comment réguler un moteur par un API.....	11
I.7 Description de l'API S7 1200 et son logiciel	12
I.7.1 API S7 1200	12
I.7.2 Logiciel Tia Portal.....	13
I.8 Développement d'Interface homme machine.....	13
I.8.1 Introduction.....	13

I.8.2	Logiciel WinCC	14
I.9	Conclusion	14
II.	Chapitre 2 : Description, modélisation et régulation du MCC.....	15
II.1	Introduction	15
II.2	Moteur à courant continu.....	15
II.2.1	Description.....	15
II.2.2	Pilotage de la vitesse de rotation	15
II.3	Régulation de position du MCC	17
II.3.1	Asservissement de position avec régulateur PID	17
II.3.2	Pont en H	17
II.3.3	MLI (PWM).....	17
II.4	Objet technologique : PID compact	18
II.4.1	Description.....	18
II.4.2	Insertion de l'instruction PID et de l'objet technologique	19
II.5	Conclusion	19
III.	Chapitre 3 : Développement d'IHM et régulation de position de notre station	
	AS / RS.....	20
III.1	Introduction	20
III.2	Description détaillée des composants de notre système.....	20
III.2.1	Système de déplacement horizontal.....	20
III.2.2	Système de déplacement vertical	20
III.2.3	Capteur optique odomètre.....	21
III.2.4	Système AS/RS (magasin)	21
III.2.5	Box d'interface.....	22
III.2.6	Boite postérieure.....	22
III.2.7	Clavier de commande	23
III.2.8	API S7 1200	23
III.3	Etape de création et de programmation du système.....	24
III.3.1	Création et insertion des différents appareils à notre projet	24
III.3.2	Développement de notre programme de fonctionnement	25
III.3.3	Représentation des différentes variables utilisé dans notre programme	29
III.3.4	Utilisation de l'objet Technologique PID_Compact	30
III.4	Etape de développement de notre IHM sous WinCC	31
III.4.1	Représentation des différents objets de l'IHM	31

III.4.2	Création de la table des variables de notre IHM.....	32
III.5	Fonctionnement et visualisation du système	33
III.6	Conclusion	37
	Conclusion générale	38
	Références bibliographiques	39
	Références web graphiques	39

Liste des Figures

Figure I.1 : Robot à bras articulé. [9]	8
Figure I.2 : Robot SCARA. [10]	9
Figure I.3 : Robot cartésien en L. [11]	9
Figure I.4 : Robot collaboratif cobots. [11]	10
Figure I.5 : Photo de notre station d'étude.....	10
Figure I.6 : Moteur à courant continu. [12].....	11
Figure I.7 : Fenêtre de démarrage de Tia Portal.	13
Figure I.8 : Logiciel WinCC.	14
Figure II.1 : Schéma d'un MCC à aimants permanents. [19].....	15
Figure II.2 : Fonctionnement du régulateur PID_Compact. [25].....	19
Figure III.1 : Système de déplacement horizontal.....	20
Figure III.2 : Système de déplacement vertical.	21
Figure III.3 : Capteur optique : odomètre.	21
Figure III.4 : Système AS/RS (magasin).....	22
Figure III.5 : Box d'interface.	22
Figure III.6 : Boîte postérieure.....	23
Figure III.7 : Clavier de commande.....	23
Figure III.8 : API S7 1200. [14]	23
Figure III.9 : Placement d'un nouvel appareil : API S7 1200 avec sa CPU 1214C DC/DC/Relais.....	24
Figure III.10 : Ajout du « PC-System » pour la création d'IHM avec WinCC RT Advanced.	25
Figure III.11 : Affichage des adresses IP de chaque appareil pour vérification et élimination tout conflit IP.....	25
Figure III.12 Schéma Ladder pour la marche / Arrêt et le déplacement gauche/droite de l'axe X.....	26
Figure III.13 : Schéma Ladder pour la marche / Arrêt de l'axe Z.....	26
Figure III.14 : Schéma Ladder pour l'activation et le lancement du compteur d'axe X.....	27
Figure III.15 : Schéma Ladder pour l'activation et le lancement du compteur d'axe Z.	27
Figure III.16 : Détermination de la distance utilisant le compteur avec l'instruction « MUL ».....	28
Figure III.17 : Affectation de la variable consigne par les boutons respectifs.	28
Figure III.18 : Affichage du Bloc « Main OB1 ».....	29
Figure III.19 : Table des mnémoniques des entrée / sorties physiques.....	29
Figure III.20 : Bloc de donnée DB1 pour la structure : « Commande Manuelle et Régulation ».....	30
Figure III.21 : Bloc de donnée visualisant tous les « Positions Produits » et les bouton « Position ».....	30
Figure III.22 : Instruction PID_Compact pour le déplacement horizontal.	31
Figure III.23 : Instruction PID_Compact pour le déplacement vertical.....	31
Figure III.24 : Etape de développement de notre IHM du système sous logiciel WinCC.....	32
Figure III.25 : Table de variables de notre IHM.....	32

Figure III.26 : Schéma Ladder de l'axe X du système en fonctionnement en actionnant le bouton du position « P0.0 ».	33
Figure III.27 : Déplacement du robot vers la position « 0.0 ».	33
Figure III.28 : Déplacement du robot vers la position « 1.1 ».	34
Figure III.29 : Schéma Ladder de l'axe X du système en fonctionnement à l'action du bouton du position « P2.3 ».	34
Figure III.30 : Fonctionnement des 2 « PID_Compact » pour les 2 axes à la sélection de la position « P2.3 » par le bouton.	35
Figure III.31 : Représentation des valeurs du « PID_Compact » suivant la consigne pour l'axe X.	35
Figure III.32 : Représentation des valeurs du « PID_Compact » suivant la consigne pour l'axe Z.	35
Figure III.33 : Schéma Ladder de l'axe X en fonctionnement à l'arrivée du système la position « 2.3 ».	36
Figure III.34 : Schéma Ladder de l'axe Z en fonctionnement à l'arrivée du système la position « 2.3 ».	36
Figure III.35 : Paramètres PID pour l'instruction « PID_Compact ».	37
Figure III.36 : Déplacement du robot vers la position « 2.3 ».	37

Nomenclature

API : Automate Programmable Industriel
AS/RS : Système automatisé de stockage / déstockage
CPU : Central Programming Unit
D : Coefficient dérivé
DC : Double cycle
D/L : Dépôt / Livraison
E/S : Entrée / Sortie
I : Coefficient intégral
IHM : Interface Homme Machine
IP : Internet Protocol
P : Coefficient Proportionnel
PLC : Contrôleur Logique Programmable
SC : Simple Cycle
SCARA : Selective Compliance Assembly Robot Arm
SCL : Structures Control Language
SCADA : Supervisory Control And Data Acquisition
S/D : Stockage / Déstockage

Introduction Générale

L'industrie 4.0, aussi appelée industrie du futur, est la quatrième révolution industrielle. Elle fait référence à la transformation de l'industrie et des systèmes de production, grâce à l'émergence de nouvelles technologies. [1]

Cette évolution de l'industrie garantit un fonctionnement rapide et efficace des machines, afin d'optimiser le temps lors de baisse d'activité ou de la détection d'une anomalie. L'usine devient "intelligente" pour s'adapter plus facilement aux nécessités et processus de production. [1]

De nos jours les entreprises font face à une forte concurrence qui la pousse à tout faire pour fonctionner de manière économique et efficace dans tous les domaines. Il leur faut augmenter la rapidité, l'efficacité et la rentabilité de leurs processus opérationnels donc elles doivent se concentrer beaucoup plus sur les fonctions logistiques ou elle doit se disposer du matériel de manutention, de transport, de stockage et de conditionnement ce qui justifie l'automatisation des systèmes de stockage. Parmi les solutions technologiques qui répondent aux besoins ci-dessus, on trouve les systèmes automatisés de stockage/déstockage AS/RS. Ce sont des composantes intelligentes d'une chaîne logistique et de stockage très répandus dans l'industrie. Ils peuvent fournir des charges de tous genres (outils, matériaux, palettes, produits...) avec des temps de réponse rapide pour entretenir les demandes de fabrication, d'entreposage, et des applications de distribution. En plus, La surface au sol disponible peut être exploitée au maximum. Ils assurent aussi un excellent contrôle des stocks et augmentent la sécurité. [2]

L'objectif de notre travail est de commander un robot de stockage / déstockage à contrôle cartésien avec un automate programmable industriel de type S7 1200 et de le superviser par un IHM.

Ce mémoire est divisé en trois chapitres :

Le premier chapitre décrit l'industrie 4.0 et la description des systèmes stockage déstockage AS/RS qui effectuent des opérations de stockage et de déstockage avec rapidité et précision sous un degré d'automatisation défini en décrivant les différents robots industriels.

Le deuxième chapitre est un rappel sur le moteur à courant continu et sa régulation de position avec le régulateur PID.

Le troisième chapitre représente notre station de stockage / déstockage du module d'emménagement à contrôle cartésien « Mod. MCS-620/EV » de marque ElettronicaVeneta qui se trouve au niveau du laboratoire de Productique de Tlemcen (MELT) qui est un laboratoire de recherche à la faculté de Technologie, Université de Tlemcen. Dans ce chapitre, nous allons montrer en détail les différentes étapes de programmation de notre station commandée avec un API S7 1200. Il sera suivi par le développement aussi de notre IHM sur WinCC.

Enfin, nous achèverons ce mémoire par une conclusion générale décrivant tous les points forts de ce PFE.

Chapitre I

Description des systèmes de stockage / Déstockage (AS/RS)

I.1 Introduction

Traditionnellement, les historiens font état de trois révolutions industrielles. Aujourd'hui, une quatrième prend forme et elle va encore changer notre monde. La première révolution, c'est la domestication de la force mécanique, de la machine à vapeur, avec la transformation radicale des transports et le début des fabrications en série. L'industrie atteindra les masses, et les masses commenceront à quitter les campagnes pour peupler les villes. La seconde révolution, c'est l'apparition de nouvelles formes d'énergies, le déploiement de l'électricité et l'avènement de la chimie. Le monde moderne change totalement d'apparence. L'être humain devient travailleur consommateur. La troisième révolution se nourrit des progrès incessants de l'électronique et de l'informatique. Les loisirs grignotent le temps de travail, des usages totalement nouveaux apparaissent, la communication et le virtuel prennent le pas sur le réel. Poursuivant une course exponentielle arrive la quatrième révolution, celle des objets connectés, du big data et des algorithmes intelligents. Cette quatrième révolution, qui commence à toucher notre vie quotidienne, est née des innovations de l'industrie. C'est l'industrie qui crée ces nouveaux produits et services. Cette révolution va aussi radicalement transformer l'industrie : bienvenue dans l'industrie 4.0. [3]

I.2 Industrie 4.0

I.2.1 Qu'est-ce que « L'Industrie 4.0 » ?

Le projet « Industrie 4.0 » vise à développer de nouvelles organisations de production sur toute la chaîne de valeur. « L'Industrie 4.0 » est annoncée comme la 4ème révolution industrielle. La 1ère révolution industrielle fut déclenchée dans les années 1780 par la création de la première usine de production mécanisée notamment grâce à l'invention de la machine à vapeur. La 2ème révolution industrielle débuta en 1850, avec l'invention de l'électricité puis de la production de masse. Dans les années 1970, la 3ème révolution industrielle signa l'entrée dans l'ère de la production automatisée avec le développement de l'électronique et les débuts de l'automatisme et de l'informatique industrielle. La 4ème révolution industrielle débuta, elle, avec l'Internet des objets connectés et le cloud computing, pour fabriquer des produits grâce à des systèmes intelligents, tels que les systèmes de simulation et les capteurs par exemple. [4]

I.2.2 Comment fonctionne une usine 4.0 ?

Une usine 4.0 fonctionne selon six principes clés comme :

- **Une usine est virtualisée** afin de pouvoir simuler et suivre en 3D les produits, les processus et l'environnement de production,
- **Des systèmes interopérables** ayant la capacité de communiquer et d'interagir entre eux,
- **Des systèmes décisionnels décentralisés** ou des systèmes cyber-physiques pouvant prendre des décisions de façon autonome. L'analyse et la prise de décision s'effectuent en temps réel, grâce à une communication permanente et instantanée,

- **Des systèmes avec un service orienté** pour permettre une maintenance améliorée et offrent de nouveaux services,
- **Des systèmes modulaires** qui s'adaptent rapidement à une demande changeante. [4]

I.3 Systèmes stockage/déstockage AS/RS

I.3.1 Définition

Les systèmes de stockage automatiques sont souvent utilisés pour stocker des volumes importants et des charges lourdes où la densité de stockage est élevée en raison d'un espace insuffisant. Il existe un large éventail d'applications pour les factoreries et les centres de distribution.

Les systèmes de stockage automatisés sont pour la plupart entièrement automatiques et fonctionnent sans personnel. Une capacité de stockage sensiblement élevée est obtenue en conséquence. [5]

I.3.2 Fonctionnement

Pour le stockage ou le déstockage d'un produit, la machine S/D se déplace à la fois verticalement et horizontalement vers le produit désiré, qui a été sélectionné par un opérateur, où cette sélection est transmise à la machine S/D via le système de contrôle.

On distingue deux opérations :

- a. L'opération de stockage : c'est de transporter un produit de la station D/L vers le casier déterminé,
- b. L'opération de déstockage : c'est de récupérer un produit du casier sélectionné et le transporter vers la station D/L.

Il y a deux modes de fonctionnement de la machine S/D :

- a. Fonctionnement en simple cycle (SC) : la machine S/D visite un seul casier, c'est-à-dire, elle effectue une seule opération (stockage ou déstockage),
- b. Fonctionnement en double cycle (DC) : la machine S/D visite deux casiers, elle commence par une opération de stockage puis elle passe directement vers une opération de déstockage. [6]

I.3.3 Systèmes de gestion d'entrepôt intelligents

De nos jours, beaucoup d'industrie utilise des systèmes de constructions intelligents et aussi des systèmes de gestion d'entrepôt intelligents qui sont des systèmes intelligents de stockage / déstockage complets comprenant certains logiciels, matériels et d'autres composants.

Les systèmes de gestion d'entrepôts intelligents sont composés des éléments suivants : système de rayonnage, robot autocontrôlé, système informatique, matériels informatiques et logiciels.

Les composants de ce système ont la capacité de se déplacer sur les axes X, Y, Z, U et N, qui sont des armoires à commande électronique avec des systèmes de rayonnage, un ordinateur et du matériel pouvant être piloté par un PLC (contrôleur logique programmable) ou grue empileur à carte conducteur, navette (transporteur de palette), etc. Des moteurs AC et DC fonctionnant à l'énergie électrique, des machines pouvant se déplacer avec des systèmes de convoyage, des ascenseurs entre les étages et des ascenseurs indépendants à bande magnétiques.

Selon la taille du système, la coordination de ce système nécessite un ou plusieurs PC ou tablettes, un terminal portable et un serveur principal. La communication et le flux de données dans le système peuvent être effectués sans fil ou avec un câble de donnée.

La base des systèmes de stockage intelligents repose sur des systèmes de rayonnage à haute altitude. La hauteur de ces étagères peut atteindre 30 à 40 mètres.

Dans les systèmes de stockage / déstockage intelligents, il existe un transtockeur automatique capable de se déplacer simultanément horizontalement et verticalement sur un rail fixe le long d'un couloir.

Le transtockeur est équipé de capteurs qui mesurent la distance le long de la ligne et fournissent un positionnement précis. Il existe de nombreux différents capteurs pour de nombreuses fonctions telles que la détection de palette, la détection de navette, tête de casier, etc.

Outre tous ces capteurs, des capteurs supplémentaires sensibles au mouvement sont également disponible pour chaque axe. Le transtockeur assure par lui-même le control. Dans des circonstances très particulières, des contrôleurs de mouvement ou des cartes de conduite peuvent être utilisés à la place de l'unité de contrôle PLC.

Le transtockeur (robot tour) peut travailler dans un couloir d'une largeur maximale de 1500mm. Cette largeur est une mesure spécialement déterminée selon le standard EURO PALETTE. Le travail du robot tour est coordonné par une navette (radio pallet shuttle).

Le système de navette utilise sa propre source d'alimentation DC rechargeable pour se déplacer. Le système de navette dispose de son propre système de contrôle indépendants et de capteurs ainsi que le robot tour.

La navette (radio pallet shuttle) assure une communication sans fil avec le transtockeur et le serveur principal. Le programme du système de navette mesure avec précision des palettes prédéfinies de différentes dimensions, et les empile en utilisant une méthode spécialisée où les casiers sont déchargés et les déplace vers un point spécifique où le transtockeur peut les prendre.

Lorsque la navette (radio pallet shuttle) empile ou décharge des marchandises du rayon (casier), elle les transporte consécutivement ou un par un par nombre spécifié d'articles en

série. Une fois que la navette termine sa mission, elle s'arrête au point d'attente. Une fois la tâche de stockage ou de déchargement des produits spécifiés terminée, le robot tour continue de fonctionner en transférant la navette radio-palette (radio pallet shuttle) vers un autre casier.

Si l'opération de chargement ou de déchargement est terminée, le transtockeur complète l'opération en apportant la navette radio-palette (radio pallet shuttle) au point d'attente situé à l'extérieur de la zone de stockage.

Outre les systèmes de transtockeur, et le système de navette radio-palette (radio pallet shuttle), divers systèmes auxiliaires peuvent être utilisés dans des systèmes d'entrepôt intelligents. Ceux-ci incluent, des robots convoyeurs ou suiveurs des lignes, des transpalettes, des chariots élévateurs, des élévateurs de charges à étage. Diverses méthodes peuvent être utilisées pour contrôler les systèmes manuels, la plus connue est le RFID (identification par radio fréquence)

En plus de cela, un système de barre code ou QR est utilisé. Les données reçues de ces systèmes d'identification sont collectées à partir de récepteurs placés au niveau de la rampe de départ ou du couloir de passage, puis atteignent automatiquement la base de données du système.

Au moment où les coordonnées du produit ou de l'adresse identifié atteignent un point spécifique de l'entrepôt, le transtockeur ou le transpalette achève l'opération en faisant correspondre les informations des adresses et des barres codes.

Dans le même temps, ces données peuvent être directement envoyées à d'autres systèmes de logiciels d'entrepôt pour enregistrement en utilisant un logiciel approprié. [5]

1.3.4 Composition

Généralement, un système automatisé de stockage/déstockage inclut les principaux composants suivants :

- Des étagères de stockage (Racks),
- Des allées donnant accès aux racks,
- Des machines de stockage/déstockage (S/D),
- Des points d'entrées/sorties,
- Un système de contrôle gérant le tout. [6]

1.3.5 Avantages

L'apport des AS/RS par rapport aux systèmes conventionnels réside dans plusieurs facteurs, nous pouvons citer (Singh, 1996, Sari, 2003) :

- Diminution de l'espace requis : un AS/RS peut diminuer considérablement l'espace nécessaire au stockage. la place ainsi récupérée peut être utilisée pour augmenter la production.

- Gestion et contrôle des stocks améliorés : à n'importe quel instant, le système de contrôle de l'AS/RS peut donner avec précision les quantités types et qualités des produits présents en stock et celles délivrées dans l'heure, la journée, la semaine...
- Réduction des temps de réponse : les systèmes AS/RS possèdent des temps de réponse très réduits, pour localiser, stocker ou déstocker des articles, par rapport aux systèmes classiques de stockage.
- Réduction de la main d'œuvre : l'automatisation de la majorité des opérations dans les AS/RS permet de diminuer l'intervention humaine, que ce soit dans l'administration ou l'exécution.
- Réduction des détériorations : l'automatisation permet la diminution des risques de casse des produits.
- Diminution des coûts de fonctionnement : étant donné le haut degré d'automatisation, les coûts de fonctionnement sont réduits au maximum (réduction du personnel, de la lumière, du chauffage...)
- Augmentation du rendement dans la production : grâce à l'augmentation de la productivité des stations de travail, les AS/RS permettent d'augmenter le rendement du système de production.
- Augmentation de la sécurité : les AS/RS étant des aires autonomes et fermées, sans intervention humaines, la sécurité est grandement améliorée.
- Réduction des vols : les AS/RS étant des aires autonomes et fermées, sans intervention humaines, le vol de produit par le personnel, ou autres, est très réduit.
- Grande flexibilité de fonctionnement : les AS/RS peuvent s'accommoder d'une grande variété d'articles à stocker. De plus, ils sont facilement interfaçables avec des systèmes comme les AGVS, les FMS, les systèmes d'inspection.
- Production juste à temps (JIT) : les AS/RS permettent d'avoir le bon produit, outils, palette et support au bon endroit au bon moment. [7]

1.3.6 Inconvénients

Les AS/RS peuvent offrir beaucoup d'intérêts, comme nous l'avons noté précédemment, mais, d'un autre côté, ils présentent quelques inconvénients dont les plus importants sont :

- Les AS/RS, une fois installés sont très difficiles à modifier,
- L'investissement initial est très élevé. Par exemple, pour un AS/RS de 10000 casiers, 5 machines S/R et un convoyeur de 60m, l'investissement initial varie entre 2 et 3 millions de Dollars US. [8]

1.3.7 Types AS/RS

Il existe différents types d'AS/RS. Leur classification est basée sur leurs structures et leurs applications. Nous pouvons citer, parmi les plus importants :

- AS/RS à charge unitaire,
- AS/RS multi allées,
- AS/RS à racks glissants (ou racks mobiles).
- AS/RS à mini charge,
- AS/RS à carrousel,
- AS/RS à personne embarquée,
- AS/RS à étagères profondes,
- AS/RS à convoyeur gravitationnel, **[8]**

1) AS/RS à charge unitaire

Les AS/RS à charge unitaire sont utilisés pour stocker des charges palettisées ou en container standard. Les charges sont en général supérieures à 250 kg par unité. En général, le système est commandé par ordinateur. Il est équipé d'une machine de stockage/déstockage, appelée machine S/R, conçue pour traiter des containers ou palettes à charge unitaire. Chaque machine est placée sur des rails fixés au sol. La machine S/R est composée d'un cadre sur lequel coulisse, verticalement, un plateau, qui sert à transporter les charges. Ce plateau peut se déplacer horizontalement à la perpendiculaire du cadre. De ce fait, il peut charger/décharger les produits dans les casiers, ainsi qu'au niveau de la station de dépôt/livraison. Plusieurs types de mécanismes de support de la charge sont possibles :

- Le système à plateau coulissant : c'est le même principe que le transpalette.
- Le système à ventouse : la charge est accrochée en créant un vide dans des ventouses.
- Le système à aimant : approprié pour les produits en fer et acier. Les machines S/R peuvent avoir des hauteurs de 6m à 40m et supportant des charges de 50kg à 5tonnes. **[8]**

2) AS/RS multi allées

Les AS/RS multi allées sont une variation des AS/RS à charge unitaire. Leurs machines S/R capables de circuler sur deux axes horizontaux, peuvent servir plusieurs allées à la fois. Leur topologie est identique à celle des AS/RS à charge unitaire. Toutefois une allée commune reliant toutes les allées de service est aménagée pour permettre à la machine S/R de circuler. **[8]**

3) AS/RS à racks glissants

Les AS/RS à racks glissants sont une variation des AS/RS multi-allées. La particularité de ces systèmes est que les allées de service n'apparaissent que lorsqu'une opération de stockage ou déstockage est prévue dans les racks correspondants. Les racks formant le système glissent latéralement sur des rails de telle sorte que l'on puisse ouvrir une allée entre n'importe quels deux racks mitoyens. Au repos, il existe une seule allée de service, celle-ci pouvant « se déplacer » entre n'importe quel deux racks mitoyens (en vérité ce n'est pas l'allée qui se déplace mais les racks). **[8]**

4) AS/RS à convoyeur gravitationnel

Ce système est une variation des AS/RS à étagères profondes. Il est conçu pour stocker soit des produits individuels, soit des charges en palettes ou container. Le système consiste en un ensemble de casiers profonds équipés, chacun, d'un convoyeur gravitationnel, aux extrémités desquels se trouvent deux machines S/R. Le stockage se réalise sur un ensemble de convoyeurs gravitationnels accessibles aux deux extrémités. Le chargement se fait sur une face et le déchargement se fait sur l'autre face. Les deux machines S/R sont reliées par un convoyeur gravitationnel de restockage incliné en sens inverse. Ceci permet, lorsqu'on désire déstocker un produit se trouvant au milieu d'un casier, de sortir les produits qui le précèdent, de les mettre sur le convoyeur gravitationnel, puis de les restocker par l'autre face grâce à l'autre machine S/R. Dans ce cas et en considérant une seule étagère, une machine S/R sera spécialisée dans le stockage et l'autre dans le déstockage. [8]

I.4 Différents robots

Il existe plusieurs types de robots industriels conçus pour répondre à divers besoins de fabrication et de production. Voici quelques-uns des types les plus courants :

I.4.1 Robots manipulateurs (ou robots articulés)

Ces robots sont composés de bras articulés similaires à ceux d'un humain. Ils sont très polyvalents et peuvent effectuer une large gamme de tâches, comme le soudage, le montage, la peinture, la palettisation et le chargement/déchargement de pièces.



Figure I.1 : Robot à bras articulé. [9]

I.4.2 Robots SCARA

Les robots SCARA (Selective Compliance Assembly Robot Arm) sont des robots industriels qui se distinguent par leur structure en bras articulé, caractérisée par deux articulations rotatives et une articulation linéaire. Leur conception leur permet de se déplacer rapidement et avec précision dans un plan horizontal, ce qui les rend idéaux pour des applications de manipulation et d'assemblage rapides dans des environnements de production. Les robots SCARA sont largement utilisés dans des industries telles que l'électronique, l'automobile, l'industrie alimentaire et l'assemblage de petites pièces. Grâce à

leur vitesse, leur précision et leur capacité à effectuer des tâches répétitives, ils contribuent à accroître l'efficacité et la productivité des lignes de production.



Figure I.2 : Robot SCARA. [10]

I.4.3 Robots cartésiens

Les robots cartésiens, également connus sous le nom de robots à trois axes, se distinguent par leur structure linéaire le long de trois axes orthogonaux : X, Y et Z. Ils se déplacent principalement dans des mouvements rectilignes, offrant une grande précision et répétabilité. Ces robots sont souvent utilisés pour des tâches telles que le positionnement, la manipulation de pièces et l'inspection dans diverses industries, grâce à leur facilité de programmation et leur capacité à atteindre des points précis dans un espace tridimensionnel.



Figure I.3 : Robot cartésien en L. [11]

I.4.4 Robots collaboratifs

Les robots collaboratifs, également appelés cobots, sont des robots conçus pour travailler en collaboration directe avec les humains dans un même espace de travail. Contrairement aux robots industriels traditionnels, les cobots sont dotés de capteurs et de systèmes de sécurité avancés qui leur permettent de détecter la présence humaine et de réagir en conséquence pour éviter les collisions et assurer la sécurité des travailleurs.



Figure I.4 : Robot collaboratif cobots. [11]

I.5 Description de notre prototype

Notre prototype utilise un moteur à CC commandé par un API S7 1200. La Figure I.5 montre le schéma réel de prototype qui station de stockage avec 12 emplacements.

La commande adoptée pour ce type de système est la commande de position. Dans le paragraphe suivant, nous allons donner uniquement le principe de la régulation du MCC utilisant un API comme cerveau de commande.

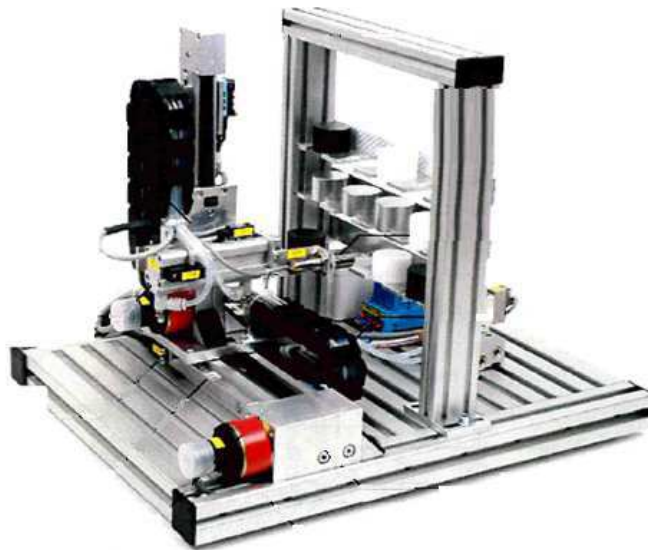


Figure I.5 : Photo de notre station d'étude.

I.6 MCC et sa régulation de position

I.6.1 Photo d'un MCC

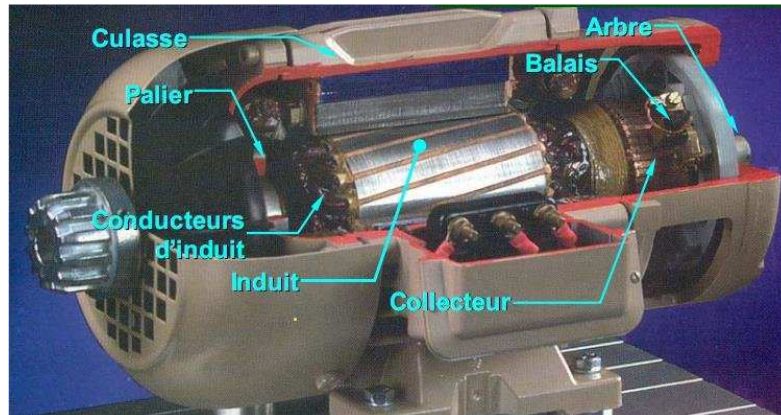


Figure I.6 : Moteur à courant continu. [12]

I.6.2 Utilisation

Ces moteurs sont idéaux pour tracter des machines nécessitant une large plage de vitesses avec précision. Cette caractéristique a fait que ces derniers temps, ces moteurs sont plus présents dans divers processus industriels.

Les moteurs à courant continu sont utilisés dans les platines, les lecteurs de CD et les lecteurs de stockage magnétiques. Ces types de mécanismes utilisent des moteurs brushless à aimants fixes. Ces moteurs offrent un contrôle efficace de la vitesse et un couple de démarrage élevé.

Dans le domaine des jouets, les moteurs électriques à courant continu sont également souvent sélectionnés. [13]

I.6.3 Comment réguler un moteur par un API

Pour réguler la position d'un moteur à l'aide d'un automate programmable industriel (API) S7-1200, vous devez suivre plusieurs étapes clés :

- 1. Configuration matérielle :** Assurez-vous que l'API S7-1200 est équipé des modules nécessaires pour le contrôle de position, tels que des modules d'entrée/sortie analogiques ou des modules de communication appropriés.
- 2. Paramétrage du logiciel :** Utilisez le logiciel TIA Portal pour configurer votre API. Vous devrez paramétrer les adresses IP pour la communication entre l'API et le moteur, ainsi que configurer les paramètres spécifiques au moteur, comme la vitesse, l'accélération, et la position désirée.
- 3. Programmation :** Écrivez un programme en utilisant le langage de programmation approprié (par exemple, Ladder, FBD, SCL, etc.) pour contrôler le moteur. Le programme doit inclure des instructions pour lire les signaux du capteur de position et ajuster la sortie du moteur en conséquence.

4. **Test et mise en service** : Après la programmation, testez le système pour vous assurer que le moteur atteint la position désirée avec précision. Utilisez les fonctions de diagnostic du TIA Portal pour aider à dépanner tout problème.
5. **Optimisation** : Ajustez les paramètres du programme, si nécessaire, pour améliorer la précision et la réactivité du système de contrôle de position.

I.7 Description de l'API S7 1200 et son logiciel

I.7.1 API S7 1200

1) Introduction à l'automate S7-1200

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

La CPU combine un microprocesseur, une alimentation intégrée, des circuits d'entrée et de sortie, un PROFINET intégré, des E/S rapides de commande de mouvement, ainsi que des entrées analogiques intégrées dans un boîtier compact en vue de créer un contrôleur puissant. Une fois que vous avez chargé votre programme, la CPU contient la logique nécessaire au contrôle et à la commande des appareils dans votre application. La CPU surveille les entrées et modifie les sorties conformément à la logique de votre programme utilisateur, qui peut contenir des instructions booléennes, des instructions de comptage, des instructions de temporisation, des instructions mathématiques complexes ainsi que des commandes pour communiquer avec d'autres appareils intelligents.

La CPU fournit un port PROFINET permettant de communiquer par le biais d'un réseau PROFINET. Des modules supplémentaires sont disponibles pour communiquer via les réseaux PROFIBUS, GPRS, RS485 ou RS232. **[14]**

2) Détails du produit

SIPLUS S7-1200, CPU 1214C DC/DC/Relais basé sur 6ES7214-1HG40-0XB0 avec Conformal Coating, I/O intégrées: 14 DI 24VDC, 10 DO Relais 2A, 2 AI 0-10V DC, alimentation: DC 24V-28,8V DC, mémoire de program. **[15]**

3) Siplus S7 1200

SIPLUS S7-1200 est le contrôleur dédié aux tâches de commande et de régulation dans la construction de machines et d'installations. Des traitements spéciaux visant à accroître la résistance et l'endurance permettent un fonctionnement même dans des conditions ambiantes particulièrement exigeantes. **[15]**

I.7.2 Logiciel Tia Portal

Siemens TIA Portal, acronyme de Totally Integrated Automation Portal, est un logiciel de programmation utilisé dans le domaine de l'automatisation industrielle. Développé par la société allemande Siemens, ce logiciel est une suite intégrée d'outils de développement qui permet de programmer et de configurer différents équipements utilisés dans le secteur industriel, tels que les automates programmables, les variateurs de vitesse, les panneaux de commande, etc. [16]

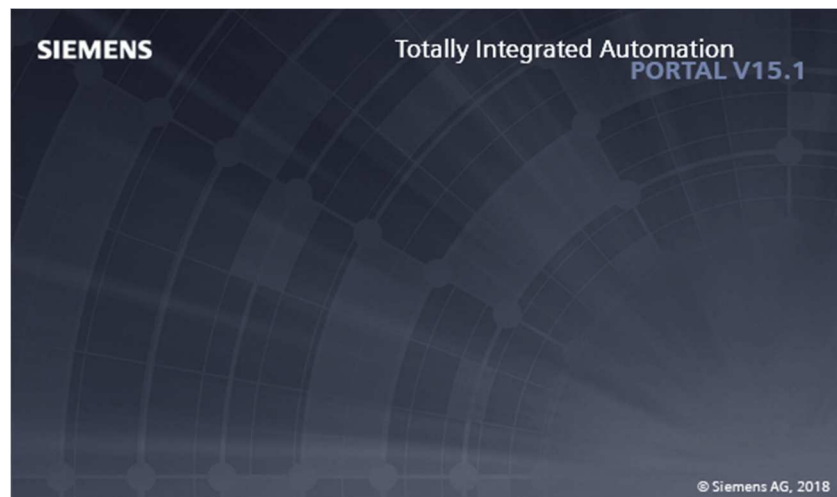


Figure I.7 : Fenêtre de démarrage de Tia Portal.

I.8 Développement d'Interface homme machine

I.8.1 Introduction

Une IHM industrielle se doit d'être robuste, fiable et résistante, surtout lorsque les opérateurs travaillent dans un environnement bruyant, poussiéreux, humide et sombre. Dans l'industrie 4.0 où l'intervention humaine se cantonne essentiellement aux prises de décisions stratégiques, la présentation des données constitue un volet très important du design. Elle ne doit pas se limiter aux affichages et aux écrans tactiles. Elle doit aussi intégrer la réalité augmentée, la reconnaissance de la parole et la synthèse vocale, et la visualisation de données intuitives. Dans les domaines de la maintenance et de la réparation, les lunettes de réalité augmentée auront un rôle important à jouer au sens où elles montreront l'impact des données issues des capteurs et les transmettront au technicien afin qu'il puisse rapidement résoudre le problème. Les données pourraient être diffusées en live streaming via des affichages visuels facilitant la compréhension. [17]

I.8.2 Logiciel WinCC

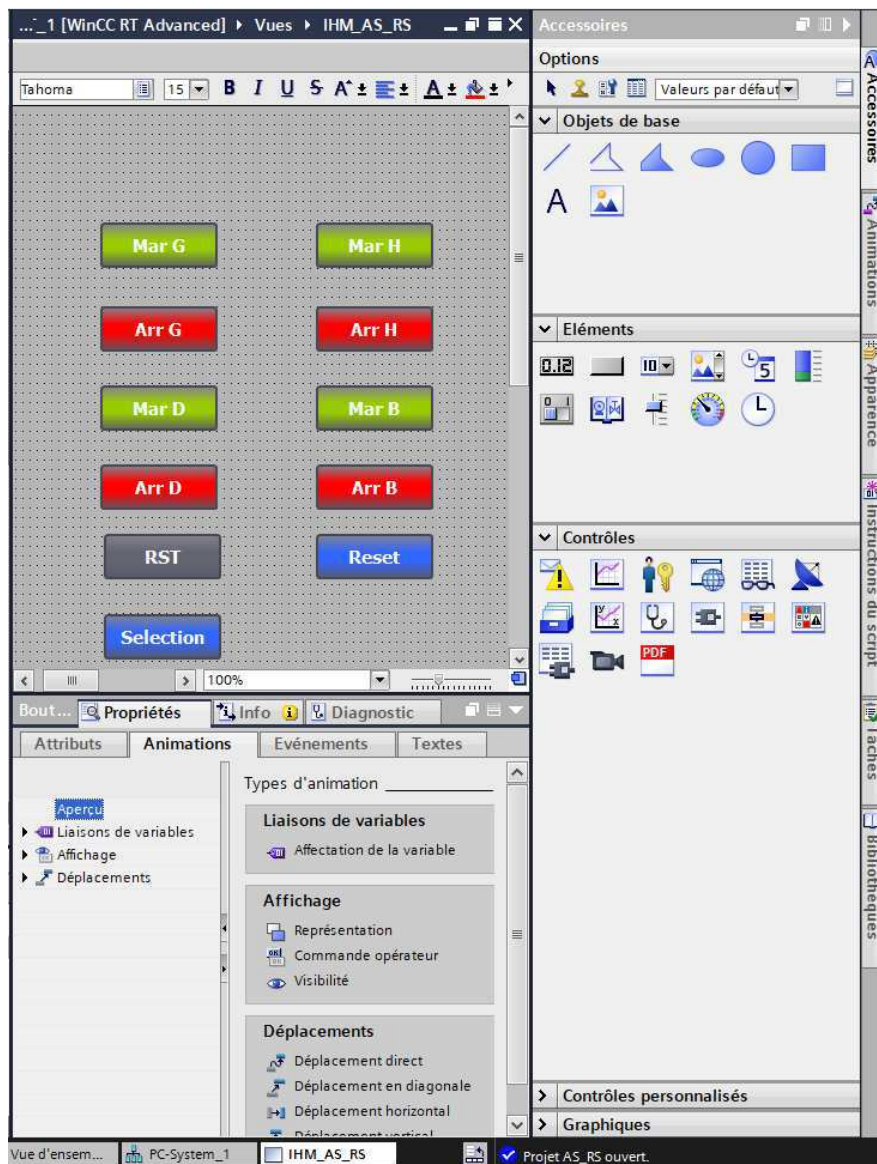


Figure I.8 : Logiciel WinCC.

SIMATIC WinCC est un système de contrôle et d'acquisition de données (SCADA) ainsi qu'une interface homme-machine développés par Siemens. Les SCADA sont particulièrement utilisés dans la surveillance des processus industriels et des infrastructures. [18]

I.9 Conclusion

Dans ce chapitre, nous avons vu la description des systèmes AS/RS avec leur fonctionnement et les différents types, et aussi les différents robots industriels. Dans le cas de notre station qui utilise un moteur à courant continu, nous avons mis l'action sur son utilisation et comment réguler ce moteur par un API. Vers la fin du chapitre, nous avons donné une description de l'API S7 1200 et son logiciel TIA PORTAL, ainsi que sur le logiciel de développement d'interface homme machine qui est le logiciel WinCC.

Chapitre 2

Régulation du MCC

II.1 Introduction

Le moteur à courant continu est un actionneur électrique ou convertisseur électromécanique. Il existe différents types de moteur qui diffèrent sur le mode d'excitation. Il existe des moteurs : à excitation indépendant, à excitation shunt, à excitation série, à excitation composée et à aimant permanent. Dans notre station, le moteur utilisé est un moteur à aimant permanent.

Dans ce chapitre, nous allons décrire en détail le MCC à aimant permanent en montrant les différents modèles mathématiques. Nous passerons ensuite sur le principe de la régulation industrielle appliquée à ce moteur par le biais d'un API S7 1200.

II.2 Moteur à courant continu

II.2.1 Description

Les moteurs à aimants permanents comportent des aimants permanents plutôt que des enroulements inducteurs pour produire le champ magnétique du stator. La Figure II.1 représente le schéma d'un tel moteur.

Ces aimants créent **une intensité de champ constante**, ce qui amène aux caractéristiques similaires à celles des moteurs à excitation en dérivation ou shunt.

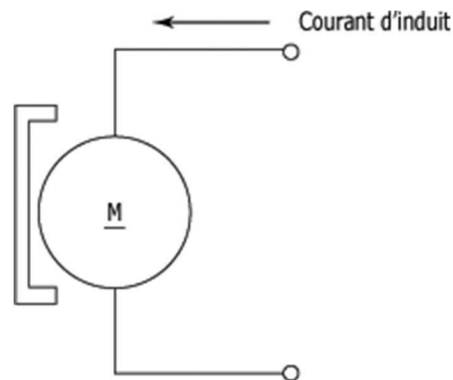


Figure II.1 : Schéma d'un MCC à aimants permanents. [19]

On se sert des moteurs à aimants permanents pour des applications **de faible et moyenne puissance**, en particulier pour **les appareils alimentés par pile**. Il est également très utilisé en robotique. [19]

II.2.2 Pilotage de la vitesse de rotation

1) Relation Vitesse et force contre-électromotrice à flux constant

Lorsque l'induit est alimenté sous une tension continue ou redressée U , il se produit en régime dynamique une force contre-électromotrice E .

$$U(t) = R_a \cdot i_a(t) + L_a \frac{di_a}{dt}(t) + e(t) \quad (2.1)$$

Par conséquent, en régime permanent, la relation qui régit cette force contre-électromotrice dévient comme suit :

$$E = U - R * I \quad (2.2)$$

Où,

- R = la résistance de l'induit [ohm].
- I = le courant dans l'induit [ampère].

Il faut noter que la force contre-électromotrice est liée à la vitesse et à l'excitation du moteur.

$$E = k_e * \omega * \Phi \quad (2.3)$$

Où,

- k_e = Constante de proportionnalité nécessaire pour la détermination de la f.e.m.
- ω = Vitesse angulaire de rotation [rad/s].
- Φ = Flux d'inducteur [Wb].

En analysant la relation ci-dessus, on voit, qu'à excitation constante Φ , la force contre-électromotrice E est proportionnelle à la vitesse de rotation. **[20]**

2) Relation Couple et flux

Quant au couple moteur, il est lié au flux inducteur et au courant de l'induit par la relation suivante.

$$C_{em} = k_m * \Phi * I \quad (2.4)$$

Avec

- k_m = Constante de proportionnalité nécessaire pour la détermination du couple.
- I = Courant d'induit [A].

En analysant la relation ci-dessus, nous voyons qu'à flux constant, le couple varie avec le courant d'induit.

Le principe fondamental de la dynamique (PFD) nous permet d'écrire :

$$J \frac{d\Omega}{dt} = C_{em} - f \cdot \Omega - Cr \quad (2.5)$$

Avec

J : moment d'inertie du moteur

f : coefficient de frottement visqueux,

Cr : couple résistant appliqué par la charge entraînée.

Pour une régulation de position, la plupart du temps, Cr très faible mais constant.

3) Variation de la vitesse

Au vu des relations existant entre la vitesse, le flux et la force contre-électromotrice, il est possible de faire varier la vitesse de notre moteur uniquement en augmentant la force contre-électromotrice E par l'augmentation de la tension d'alimentation aux bornes de l'induit tout en maintenant le flux de l'inducteur constant. On a un fonctionnement dit à "couple constant". Ce type de fonctionnement est intéressant au niveau de la conduite d'ascenseur.

II.3 Régulation de position du MCC

II.3.1 Asservissement de position avec régulateur PID

En effet, un simple asservissement PID de position modifie la tension à appliquer selon la distance du déplacement. Or il est souvent préférable de se déplacer à une vitesse globalement constante quelle que soit la distance à parcourir, en accélérant progressivement jusqu'à un palier, puis en décélérant juste avant d'atteindre la position finale afin d'éviter de la dépasser (ce que fait naturellement un conducteur de véhicule). [21]

II.3.2 Pont en H

Le pont en H permet de réaliser 2 fonctions :

- Inverser le sens de rotation du moteur en inversant le courant aux bornes du moteur,
- Faire varier la vitesse du moteur en contrôlant la tension moyenne aux bornes du moteur par hachage.

De plus, le pont en H permet d'effectuer un freinage magnétique s'il est capable d'en dissiper la puissance générée (force électromotrice). Cette opération s'effectue en actionnant en même temps les deux commutateurs, soit supérieurs ou soit inférieurs, ce qui court-circuite les bornes du moteur, et le fait par conséquent freiner. Mieux encore, il est possible avec un peu d'électronique et un contrôleur perfectionné d'effectuer un freinage régénératif. Dans le cas d'une alimentation à batterie, l'énergie est renvoyée aux batteries plutôt que dissipée dans les commutateurs du pont. [22]

II.3.3 MLI (PWM)

La modulation de largeur d'impulsion (MLI), également connue sous le nom de Pulse Width Modulation (PWM), est une technique couramment utilisée dans le contrôle des moteurs. Voici quelques points clés :

- **Principe de la MLI**
 - La MLI consiste à moduler un signal numérique pour contrôler la quantité de puissance délivrée au moteur.
 - Elle permet de générer un signal pseudo-analogique à partir d'un environnement numérique ou analogique, ce qui facilite le traitement par des composants en commutation (comme des transistors) qui agissent comme des interrupteurs ouverts ou fermés. [23]

- **Applications courantes de la MLI**

- La MLI est considérée comme une sortie analogique à partir de la conversion des signaux numériques.
- Les amplificateurs de classe D utilisent la MLI pour réguler la puissance fournie au moteur.
- La MLI est essentielle dans les alimentations à découpage.
- Elle permet de contrôler la vitesse des moteurs.
- La MLI est utilisée dans les onduleurs pour générer des signaux alternatifs.
- Elle intervient également dans les redresseurs. [23]

- **Fonctionnement de la MLI**

La MLI crée un signal logique (0 ou 1) à fréquence fixe, dont le rapport cyclique (rapport entre la durée de l'état haut et la période totale) est contrôlé numériquement.

La valeur moyenne de ce signal correspond à une grandeur analogique, calculée en multipliant le rapport cyclique par l'amplitude maximale du signal.

La MLI est souvent mise en œuvre à l'aide de microcontrôleurs modernes qui disposent de périphériques dédiés à la génération de PWM. [23]

- **Avantages de la MLI pour les moteurs sans balais :**

Dans le cas des moteurs sans balais, la MLI présente de nombreux avantages :

- Elle permet de réguler la tension d'alimentation ou le courant pour obtenir la vitesse et le couple souhaités.
- Une tension de bus de PWM adéquate et une fréquence PWM élevée réduisent l'ondulation et évitent l'utilisation d'inductances supplémentaires. [24]

II.4 Objet technologique : PID compact

II.4.1 Description

L'objet PID_Compact sert à la régulation de processus industriels à grandeurs d'entrée et de sortie continues. [25]

Cet objet PID-compact peut adapter les actions P, I et D lors la mise en route (si "l'optimisation préalable" est configurée). La configuration de l'instruction pour une "optimisation fine" permet d'optimiser les paramètres. [25]

Le régulateur PID utilise la formule suivante pour calculer la valeur de réglage pour l'instruction PID_Compact. [27]

$$y = K_p[(b \cdot w - x) + \frac{1}{T_i \cdot s}(w - x) + \frac{T_D \cdot s}{a \cdot T_D \cdot s + 1}(c \cdot w - x)] \quad (2.5)$$

y : Valeur de réglage

x : Mesure,

w : Consigne

s : Opérateur de Laplace

K_p : gain proportionnel (action P)

a : Coefficient de retard de l'action dérivée (action D)

T_i : temps d'intégration (action I)

b : Pondération de l'action proportionnelle (action P)

T_D : temps de dérivation (action D)

c : Pondération de l'action dérivée (action D)

II.4.2 Insertion de l'instruction PID et de l'objet technologique

L'objet PID_Compact fournit un régulateur PID universel optimisable. L'objet technologique contient tous les paramètres pour la boucle de régulation. [25]

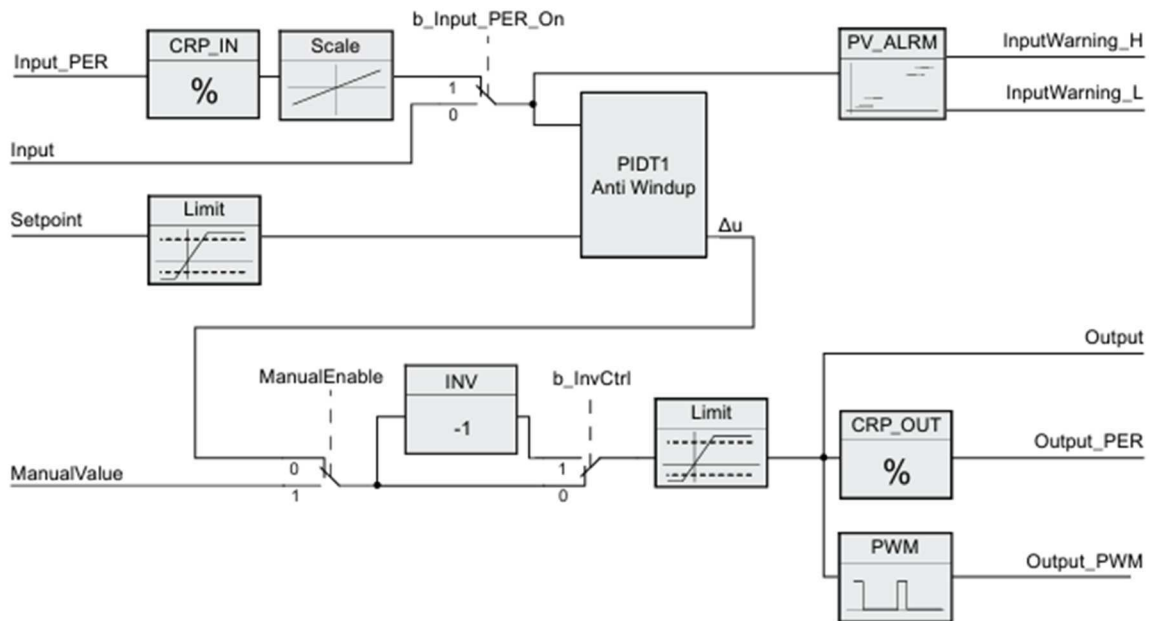


Figure II.2 : Fonctionnement du régulateur PID_Compact. [25]

La Figure II.2 donne un aperçu sur la boucle intégrée dans cet objet technologie de l'API S7 1200 qui est très efficace. Pour cette raison, il est très utilisé dans les systèmes de régulation industrielle. A la droite du schéma bloc comporte la mesure du capteur (analogique (Input) ou numérique (Input_PER)), ainsi que la consigne (Setpoint). A sa droite, nous avons la réponse du régulateur : Output, Output_PER (la plus utilisée en %) et aussi Output_PWM pour contrôler comme par exemple des convertisseurs qui fonctionne en MLI. Les autres blocs sont principalement des saturations afin de protéger notre système de mauvais fonctionnement.

II.5 Conclusion

Dans ce chapitre, nous avons fait une description puis une modélisation du moteur à courant continu à aimant permanent. Pour la partie régulation, nous avons donné un aperçu détaillé sur l'utilisation de l'objet technologique de l'API S7 1200 intégrant le régulateur PID qui sera utilisé pour réguler le moteur à courant continu.

Dans le chapitre suivant, cet objet sera étudié et configuré pour réalisation la régulation de position de notre robot cartésien en fonction de la position demandée.

Chapitre 3

**Développement d'IHM et régulation de
position de notre station AS / RS**

III.1 Introduction

Dans ce chapitre, nous allons programmer notre API S7 1200 avec le logiciel Tia portal en intégrant l'objet technologique PID_Compact pour pouvoir faire la régulation de position du MCC pour faire fonctionner notre système en déplaçant notre robot à la position sélectionnée à travers l'interface homme machine (WinCC) développée qui contrôle complètement notre système.

Nous allons commencer par une faire une description détaillée de tous les composants de notre système. Nous allons aussi montrer le fonctionnement des différents blocs de notre programme par le biais d'Imprim-écran.

III.2 Description détaillée des composants de notre système

III.2.1 Système de déplacement horizontal

Le système se déplace horizontalement sur l'axe X avec un moteur à courant continu à aimant permanent 24 V M1. Lorsque la borne (+) du moteur est connectée au relai "5K00" (Figure III.1), le moteur tourne pour déplacer le groupe pince vers la gauche. Arriver à la fin de sa course, le micro rupteur "5S00" (placé à l'extrémité gauche de l'axe X) est activé ce qui arrête le moteur M1. Inversement, lorsque la borne (+) du moteur est connectée au relai "5K01" (Figure III.2)), le moteur déplace le groupe pince vers la jusqu'au fin de course "5S06", ce qui arrête le M1. Un capteur optique "U" 5B01 (placé sur l'axe X, Figure III.3) calcule la distance de déplacement du groupe pince. La Figure III.1 représente une photo réelle du groupe pince avec son système de déplacement suivant l'axe X.

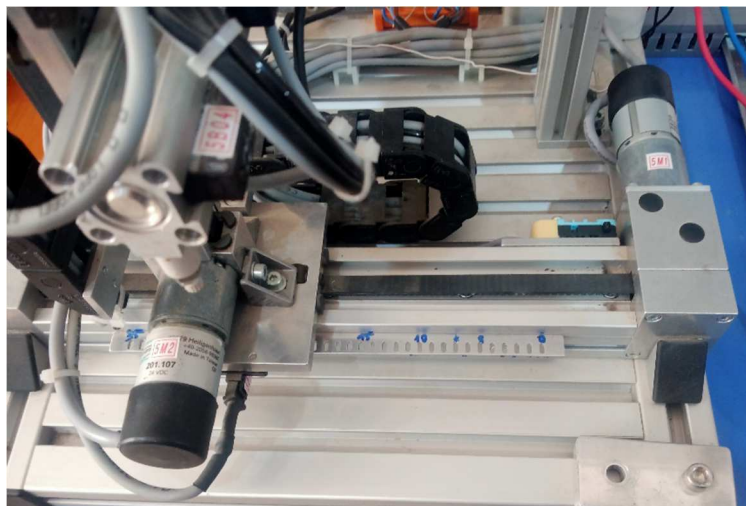


Figure III.1 : Système de déplacement horizontal.

III.2.2 Système de déplacement vertical

Le système déplace verticalement sur l'axe Z avec un moteur à courant continu à aimant permanent M2, alimenté en 24 V. C'est le même que celui utilisé pour déplacer l'axe X. Par conséquent, si la borne (+) est connectée au relai "5K03" (Figure III.3), le mouvement du groupe pince sera vers le haut jusqu'à son arrêt en actionnant le fin de course "5S10".

Inversement, si la borne (+) se connecte au relai "5K02", le mouvement du groupe pince sera descendant. Arriver au fin de course "5S02", le système s'arrête. Un capteur optique "U" 5B03 mesure le déplacement du groupe pince verticalement. La Figure III.3 montre le système de déplacement vertical du groupe pince

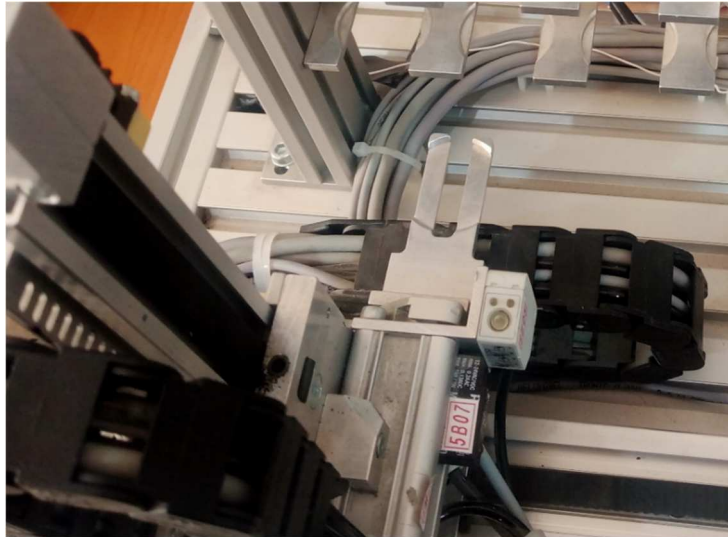


Figure III.2 : Système de déplacement vertical.

III.2.3 Capteur optique odomètre

Le capteur optique "U" 5B01 (placé sur l'axe X, Figure III.3) et 5B03 (placé sur l'axe Z) calculent le nombre d'impulsion. Il est composé d'une tige percée le long des axes X et Z et une lampe LED. Lorsque la lumière de la LED traverse un trou, le capteur est activé. L'activation du capteur incrémente ou décrémente (suivant le sens de déplacement du groupe) un compteur. En multipliant le nombre d'impulsion enregistré par le compteur par la distance entre 02 trous (qui est de 5 mm), de cette façon, nous aurons la distance déplacée par le groupe pince à tout instant suivant les 02 axes.



Figure III.3 : Capteur optique : odomètre.

III.2.4 Système AS/RS (magasin)

Le magasin présente au total 12 positions, 4 pour chaque niveau. Un capteur situé sur la pince permet d'effectuer un premier contrôle du magasin de façon à déterminer les positions libres et occupées. La Figure III.4 représente une photo réelle du système de magasin.



Figure III.4 : Système AS/RS (magasin).

III.2.5 Box d'interface

Le box d'interface de la Figure III.5 met à disposition l'état électrique des entrées / sorties du module.

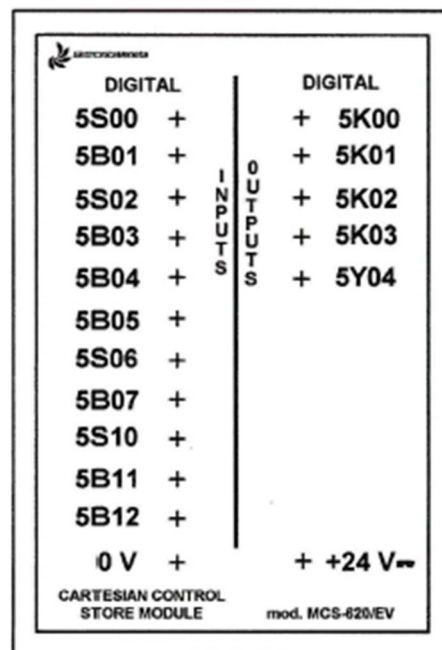


Figure III.5 : Box d'interface.

III.2.6 Boite postérieure

La boite postérieure de la Figure III.6 contient 4 boutons poussoirs, 5K00 pour le mouvement vers la gauche, 5K01 pour le mouvement vers la droite, 5K03 le mouvement vers le haut, 5K02 le mouvement vers le bas.



Figure III.6 : Boîte postérieure.

III.2.7 Clavier de commande

Il y'a dans le clavier de commande des boutons (marche, arrête et reset) et un sélecteur, utilisés pour la commande manuelle.



Figure III.7 : Clavier de commande.

III.2.8 API S7 1200

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



Figure III.8 : API S7 1200. [14]

III.3 Etape de création et de programmation du système

III.3.1 Création et insertion des différents appareils à notre projet

- 1) **Etape 1 :** Création et attribution d'un nom au projet, Ajout d'appareils et établissement de communication. Lors de la création d'un nouveau projet, nous avons choisi l'appareil de l'API S7 1200 CPU 1214C DC/DC/Relais basé sur 6ES7214-1HG40-0XB0 suivant notre besoin. La figure III.9 montre une fenêtre après d'intégration de l'appareil à notre projet.

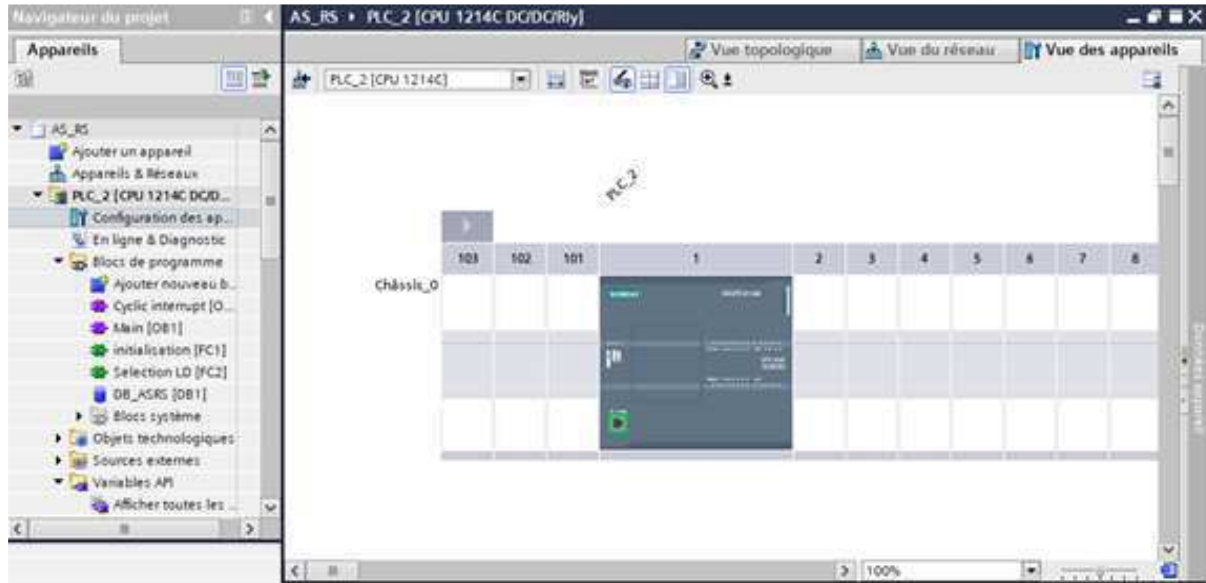


Figure III.9 : Placement d'un nouvel appareil : API S7 1200 avec sa CPU 1214C DC/DC/Relais.

- 2) **Etape 2 :** Ajout du PC System pour créer l'IHM avec WinCC RT Advanced. Il y a plusieurs façons de créer un IHM. Dans notre cas, on représente notre IHM sur notre PC, en ajoutant l'appareil PC-System. La Figure III.10 montre le PC-system avec sa carte de communication. Le WinCC RT Advanced est le logiciel de développement de notre IHM.

Dans l'industrie, on trouve des pupitres dédiés à la représentation des IHM qui sont excessivement chers.

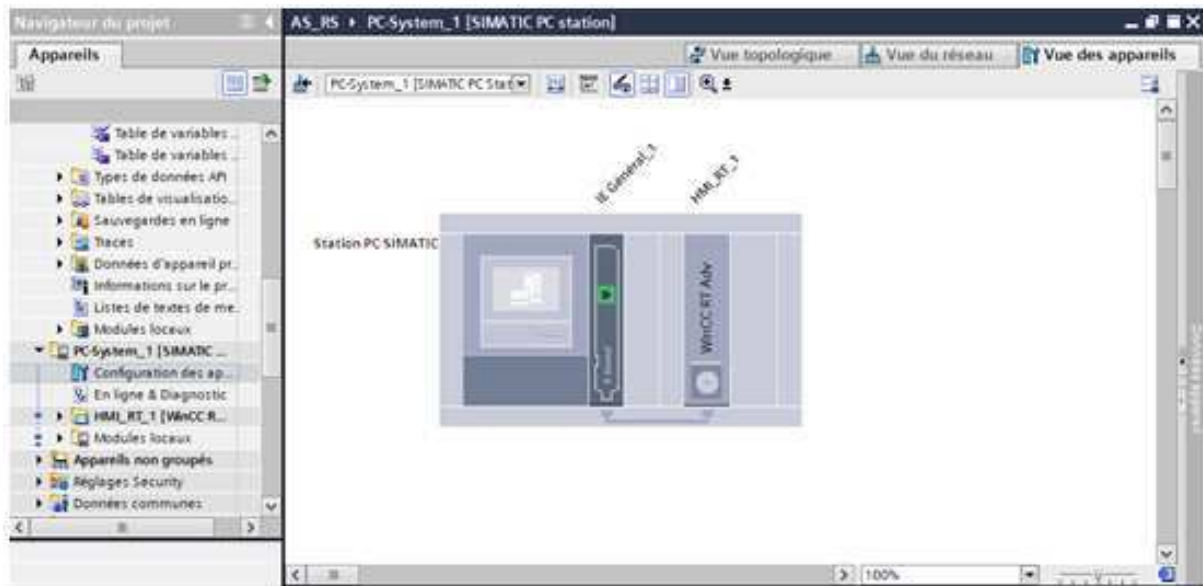


Figure III.10 : Ajout du « PC-System » pour la création d'IHM avec WinCC RT Advanced.

- 3) Etape 3 :** Liaison de communication entre la CPU 1214C DC/DC/Relais et PC System. Après établissement de la communication entre le Pc-Système et le contrôleur, il faut mieux afficher les adresses IP pour vérification s'il n'y a pas de conflit IP, comme le montre la Figure III.11. Dans notre cas, c'est le réseau Profinet qui est adopté.

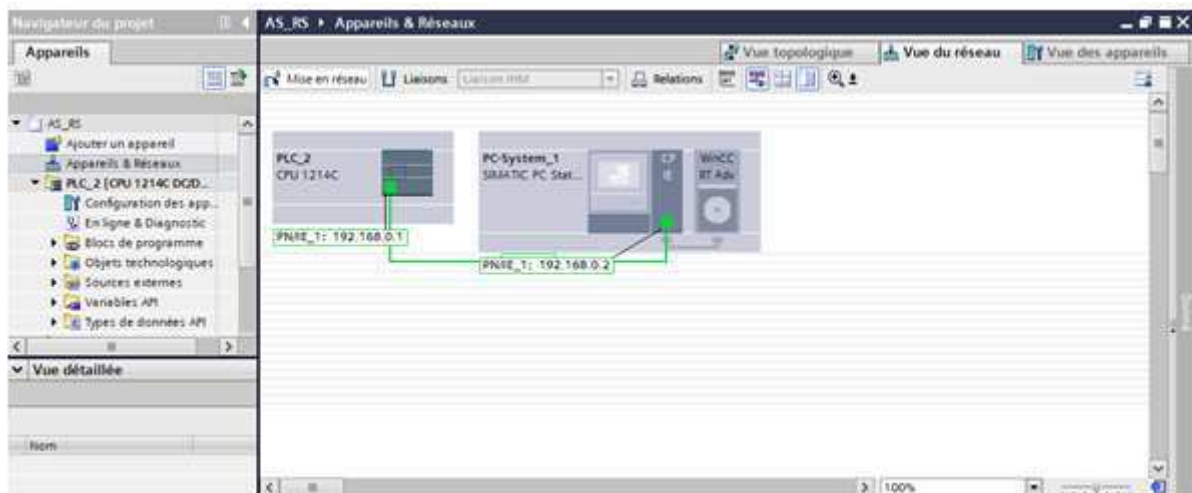


Figure III.11 : Affichage des adresses IP de chaque appareil pour vérification et élimination tout conflit IP.

III.3.2 Développement de notre programme de fonctionnement

- 1) Etape 1 :** Création des blocs FC1 et FC2. Ces blocs nous permettent d'insérer les schémas Ladder pour le fonctionnement de système et le déplacement dans les deux axes X et Z. Après avoir renommé le bloc 'Initialisation', nous avons inséré 02 schémas Ladder pour déplacer l'axe X de notre système vers la gauche et vers la droite respectivement. La Figure III.12 illustre les différents contacts insérés pour les 02 sens de déplacement.

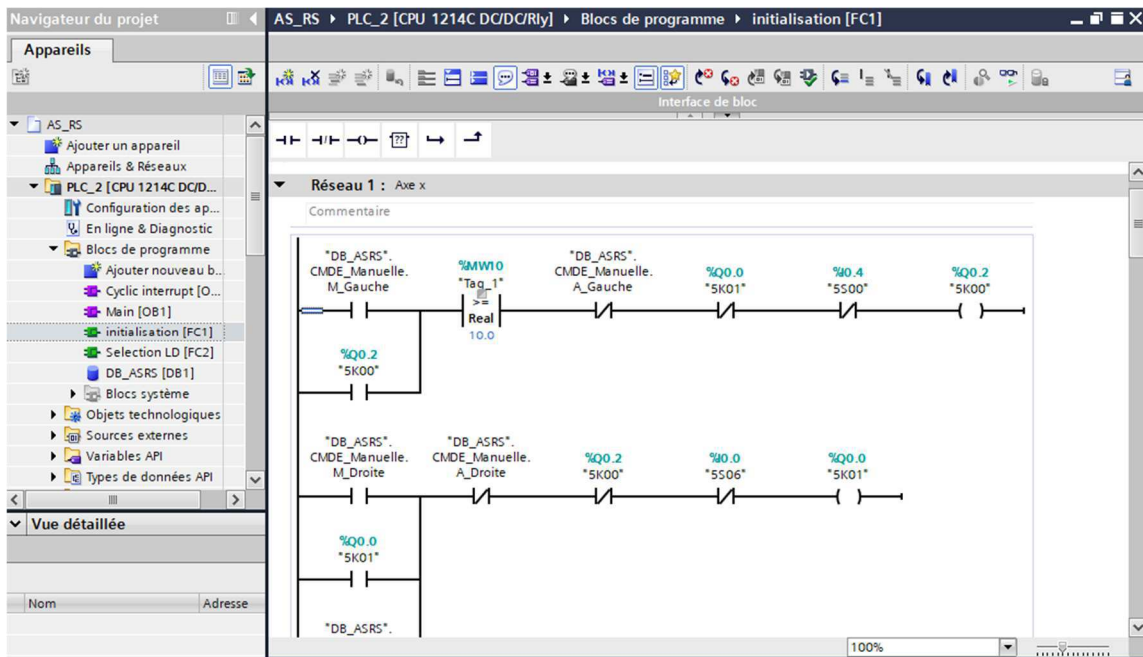


Figure III.12 : Schéma Ladder pour la marche / Arrêt et le déplacement gauche/droite de l’axe X.

La même procédure est appliquée pour faire fonctionner l’axe Z. La Figure III.13 illustre parfaitement la similitude avec les schémas Ladder du fonctionnement du système sur l’axe X.

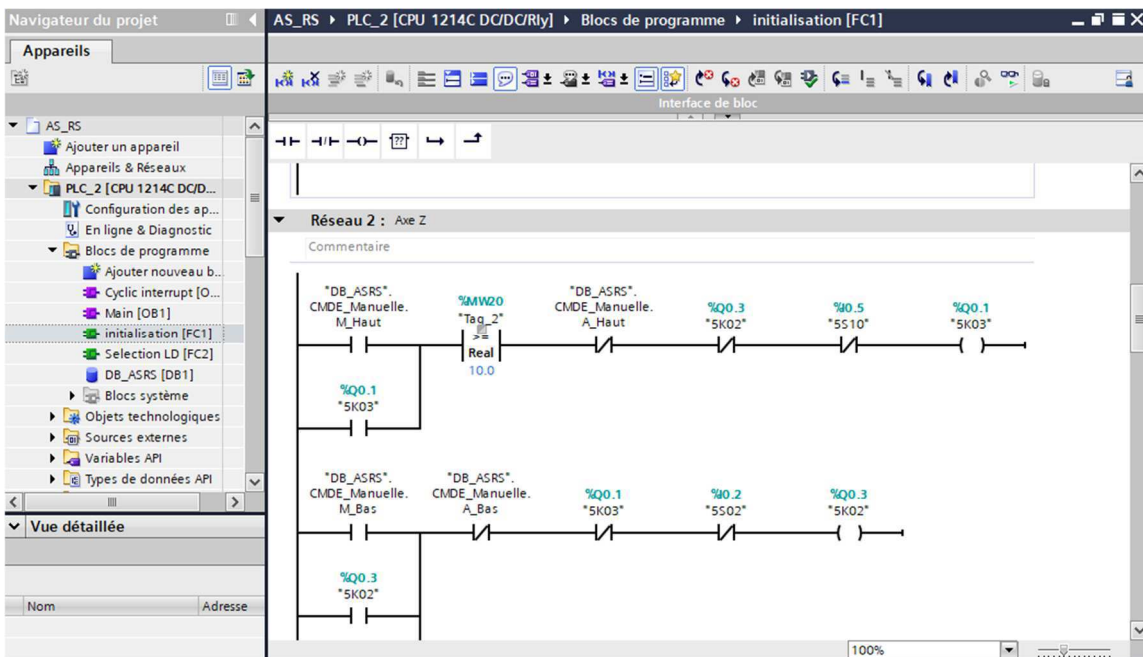


Figure III.13 : Schéma Ladder pour la marche / Arrêt de l’axe Z.

- 2) **Etape 2** : Comptage du nombre d’impulsion effectué par le déplacement sur l’axe X et Z. Dans le bloc ‘Initialisation’, nous avons inséré un réseau où nous avons introduit un compteur pour chaque axe qui nous a aidé à compter le nombre d’impulsion (par incrémentation) après chaque activation de notre capteur optique suivant l’axe et l’axe Z respectivement, comme le montre les Figures III.14 et III.15.

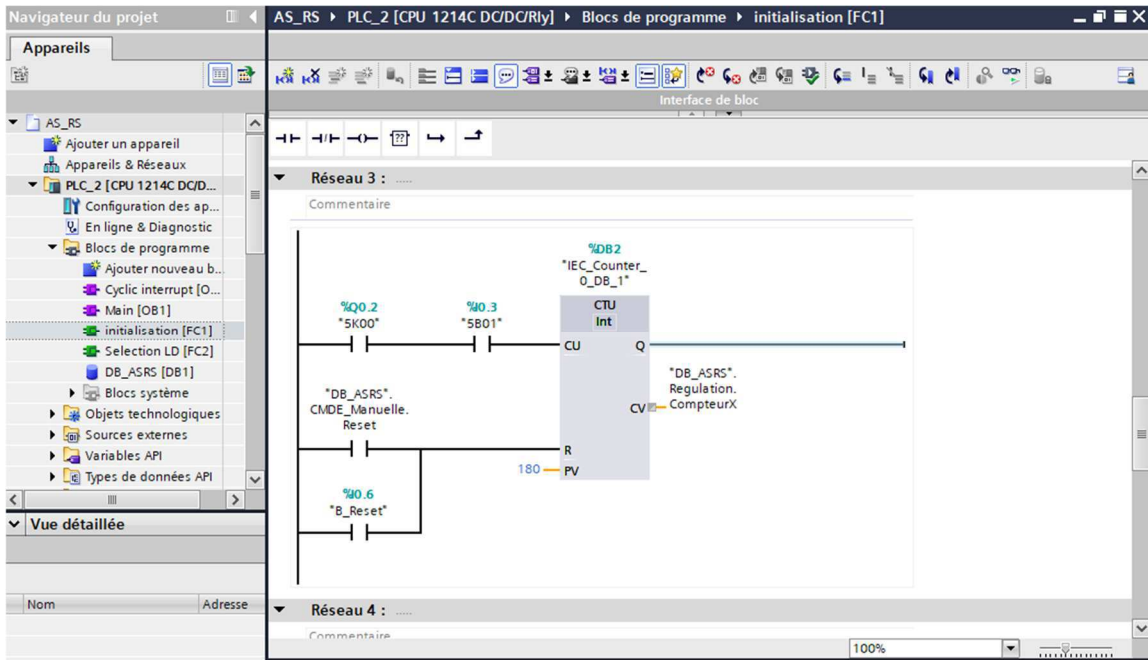


Figure III. 14 : Schéma Ladder pour l’activation et le lancement du compteur d’axe X.

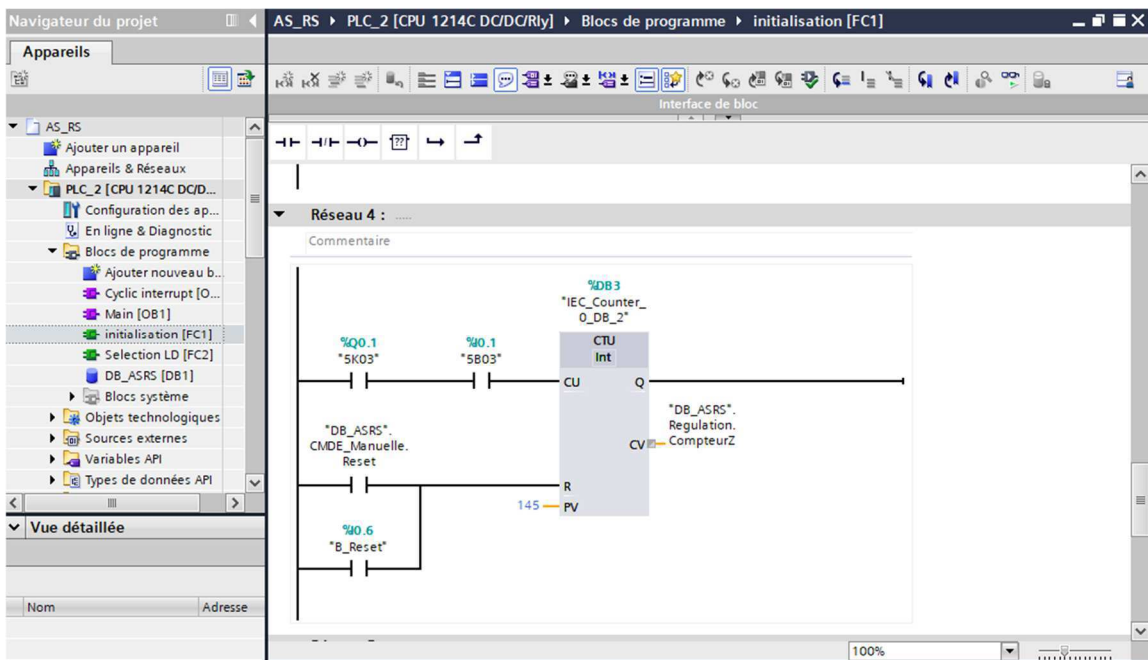


Figure III. 15 : Schéma Ladder pour l’activation et le lancement du compteur d'axe Z.

3) Etape 3 : Mesure de la distance de déplacement sur l’axe X et Z. Sur le réseau 5 et 6 de la fonction ‘Initialisation’ (Figure III.16), nous avons introduit l’instruction MUL pour obtenir la distance X et Z respectivement de déplacement qui sera égale :

$$DistanceX = CompteurX * \Delta L (mm) \tag{3.1}$$

$$DistanceZ = CompteurZ * \Delta L (mm) \tag{3.2}$$

Avec

ΔL : Distance entre deux trous de la tige percée de notre odomètre, égale à 5mm.

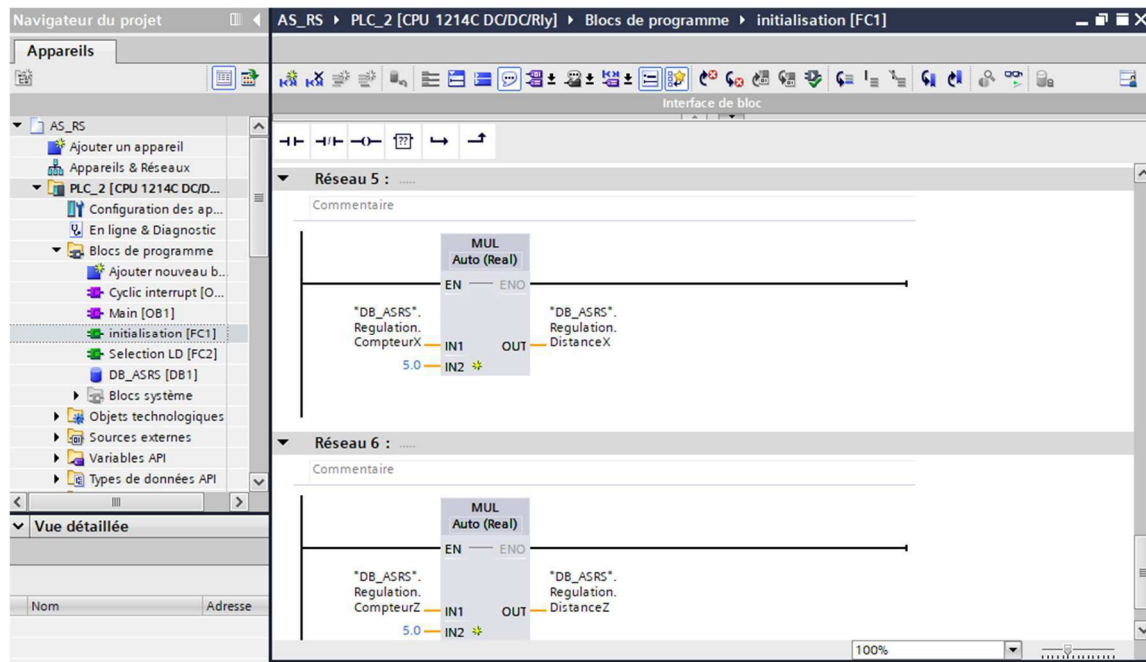


Figure III. 16 : Détermination de la distance utilisant le compteur avec l'instruction « MUL ».

- 4) **Etape 4** : Sélection de la position désirée dans l'AS/RS. Cette étape est importante. Avant le déplacement de notre robot, nous devons sélectionner l'emplacement du produit à décharger. La fonction (FC) 'Selection LD' comporte différentes instructions (Figure III .17). La plus importante est l'instruction 'Move' qui a pour rôle d'affecter une position X et une position Z aux variables 'Distance_X_Desiree' et 'Distance_Z_Desiree'. Ces dernières sont utilisées dans notre objet PID_Compact.

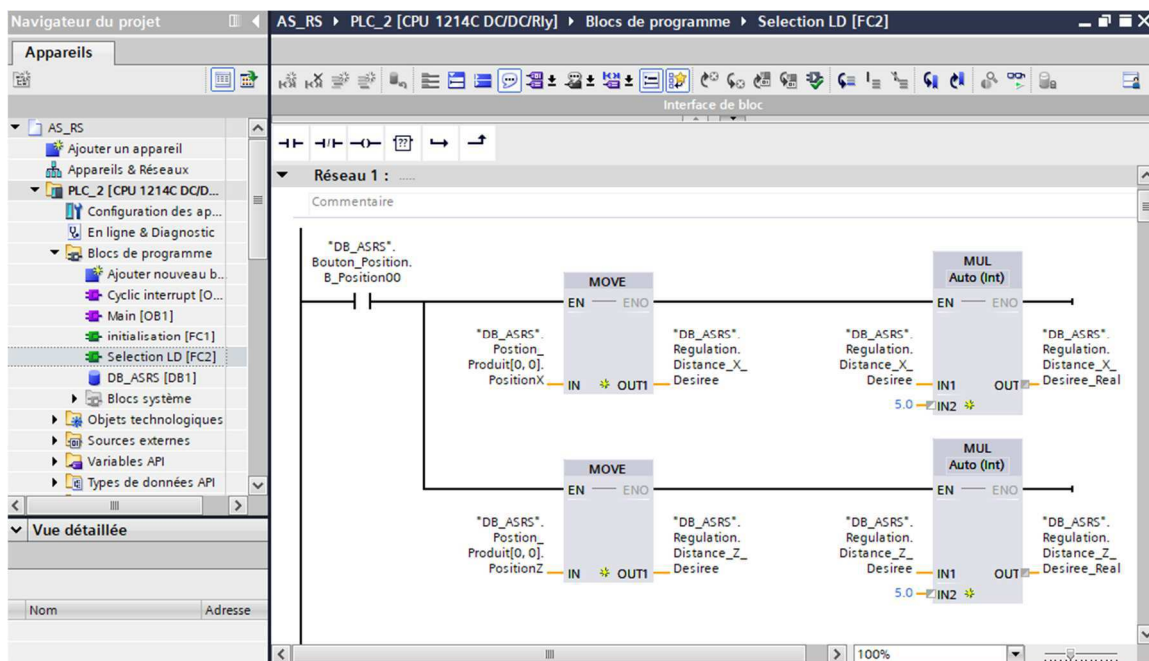


Figure III.17 : Affectation de la variable consigne par les boutons respectifs.

- 5) **Etape 5** : Placement des O2 fonction dans l'OB 'Main'. Pour que les O2 fonctions (FC) soient exécutées, il faut les placer dans le bloc organisationnel 'Main()'.

La Figure III.18 montre les deux fonctions 'Initialisation' et 'Selection LD' placées dans un réseau indépendant.

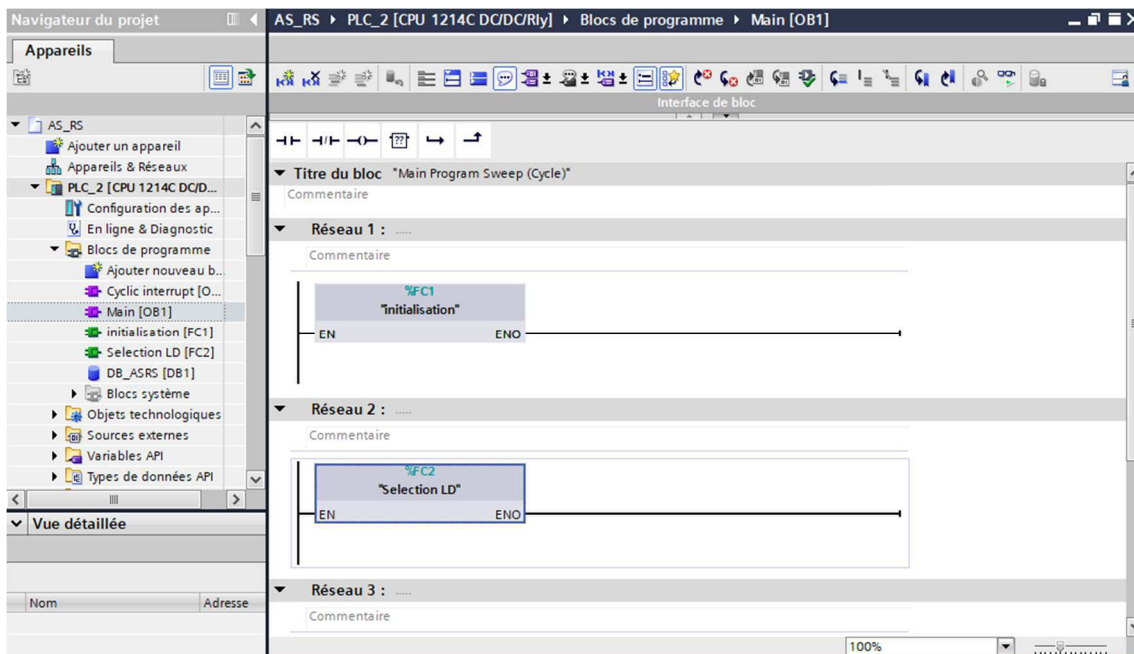


Figure III.18 : Affichage du Bloc « Main OB1 ».

III.3.3 Représentation des différentes variables utilisé dans notre programme

Pour notre programme, nous avons utilisé deux types de variable : variables Entrée / Sortie qui sont des variables physiques. Le tableau de la Figure III.19 montrent les principales variables qui se trouvent sur système réel

Nom	Type de données	Adresse	Bâma...	Acces...	Ecritu...	Visibl...	Commentaire
1	Bool	%I.4		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Fin de course gauche (o...
2	Bool	%I.0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Fin de course droite (ouv...
3	Bool	%I.2		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Fin de course bas (ouvert...
4	Bool	%I.5		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Fin de course haut (ouve...
5	Bool	%Q.2		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Relai de déplacement ga...
6	Bool	%Q.0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Relai de déplacement droi...
7	Bool	%Q.3		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Relai de déplacement Bas
8	Bool	%Q.1		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Relai de déplacement Hau...
9	Bool	%I.6		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
10	Bool	%I.3		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
11	Bool	%I.1		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
12				<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	

Figure III.19 : Table des mnémoniques des entrée / sorties physiques.

Pour améliorer la visibilité de notre programme et pour éliminer le problème d'emplacement mémoire (conflit entre les variables), nous avons choisi d'utiliser une base de données à la place d'un tableau de variable mnémoniques. Cette base de données vous donne la possibilité de bien désignée les variables en les organisant dans des structures séparées. Le

Le tableau de la Figure III.20 et III.21 montre les différentes variables qui se trouvent dans les 04 structures 'Commande_Manuelle', 'Regulation', Position_Produit' et 'Bouton_Position' de la base de données DB1. Il faut noter que, dans une base de données, nous pouvons même utiliser des matrices. Cela nous a facilité le problème de nomination des variables.

Nom	Type de données	Valeur de départ	Rémanence	Accessible...	Écritu...	Visible da...	Valeur de ...	Comme...
Static								
CMDE_Manuelle	Struct			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
M_Droite	Bool	false		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
M_Gauche	Bool	false		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
M_Haut	Bool	false		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
M_Bas	Bool	false		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
A_Droite	Bool	false		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
A_Gauche	Bool	false		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
A_Haut	Bool	false		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
A_Bas	Bool	false		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
Reset	Bool	false		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
Regulation	Struct			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
CompteurX	Real	0.0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
CompteurZ	Real	0.0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
DistanceX	Real	0.0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
DistanceZ	Real	0.0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
Distance_X_Desiree	Int	0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
Distance_Z_Desiree	Int	0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
Distance_X_Desiree_Real	Real	0.0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
Distance_Z_Desiree_Real	Real	0.0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
SortiePL_X	Real	0.0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
SortiePL_Z	Real	0.0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
Position_Produit	Array(0..2, 0..3) of ...			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
Position_Produit[0,0]	Struct			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		

Figure III.20 : Bloc de donnée DB1 pour la structure : « Commande Manuelle et Régulation ».

Nom	Type de données	Valeur de départ	Rémanence	Accessible...	Écritu...	Visible da...	Valeur de ...	Comme...
Position_Produit	Array(0..2, 0..3) of ...			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
Position_Produit[0,0]	Struct			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
Position_Produit[0,1]	Struct			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
Position_Produit[0,2]	Struct			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
Position_Produit[0,3]	Struct			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
Position_Produit[1,0]	Struct			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
Position_Produit[1,1]	Struct			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
Position_Produit[1,2]	Struct			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
Position_Produit[1,3]	Struct			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
Position_Produit[2,0]	Struct			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
Position_Produit[2,1]	Struct			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
Position_Produit[2,2]	Struct			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
Position_Produit[2,3]	Struct			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
Bouton_Position	Struct			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
B_Position00	Bool	false		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
B_Position01	Bool	false		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
B_Position02	Bool	false		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
B_Position03	Bool	false		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
B_Position04	Bool	false		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
B_Position05	Bool	false		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
B_Position06	Bool	false		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
B_Position07	Bool	false		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
B_Position08	Bool	false		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
B_Position09	Bool	false		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		

Figure III.21 : Bloc de donnée visualisant tous les « Positions Produits » et les bouton « Position ».

III.3.4 Utilisation de l'objet Technologique PID_Compact

Pour utiliser ou déposer un objet technologique PID_Compact, vous devez utiliser un OB 'Cyclic Interrupt' qui exécute cet objet à chaque temps d'échantillonnage. Dans notre cas, on a choisi $t_e = 1\text{ms}$. Les Figure III.22 et III.23 montrent l'insertion de 02 objets technologiques pour la régulation de position suivant l'axe horizontal (axe X) et suivant l'axe verticale (axe Z).

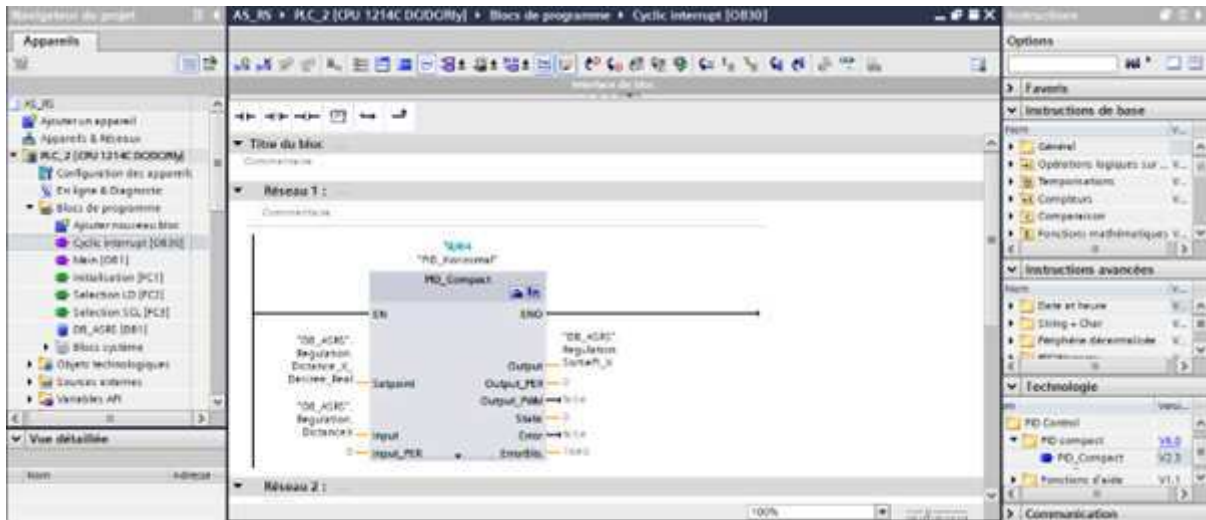


Figure III.22 : Instruction PID_Compact pour le déplacement horizontal.

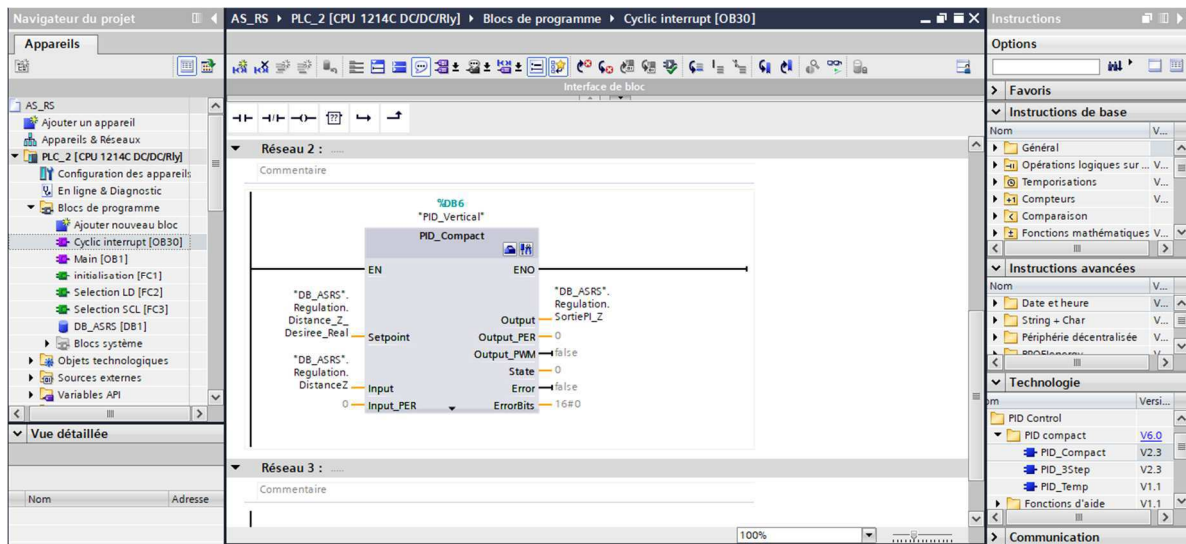


Figure III.23 : Instruction PID_Compact pour le déplacement vertical.

Avec

Setpoint : Consigne (distance en mm)

Input : Valeur mesurée par notre capteur optique (distance déplacée en mm)

Output : Sortie de notre régulateur en %.

III.4 Etape de développement de notre IHM sous WinCC

III.4.1 Représentation des différents objets de l'IHM

Notre IHM est créée sous le logiciel WinCC qui nous permet de développer notre IHM en insérant différents objets. Dans notre cas, nous avons plusieurs boutons nécessaires pour :

- Sélection de la position du produit,
- Marche / Arrêt de notre système manuellement,
- Marche automatique,
- Initialisation.

La Figure III.24 montrent la disposition de nos différents boutons sur notre IHM. Dans la partie inférieure de la figure, vous trouverez un exemple de fonctions ‘Evènements’ exécutées lors de l’action et relâchement de chaque bouton.

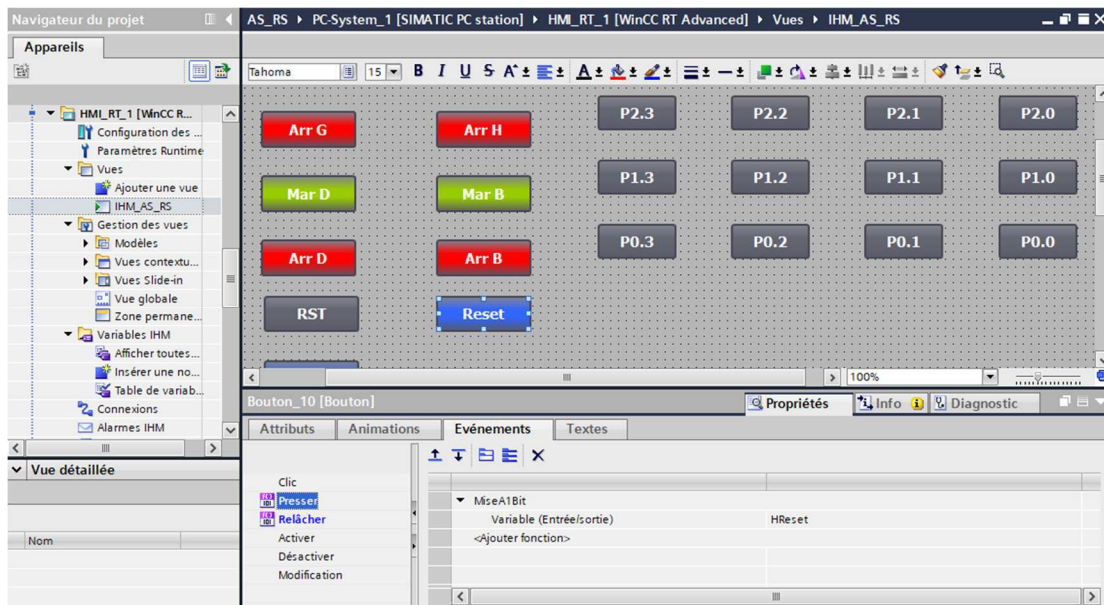


Figure III.24 : Etape de développement de notre IHM du système sous logiciel WinCC.

III.4.2 Création de la table des variables de notre IHM

Il faut savoir que chaque bouton est associé à une variable IHM connectée à une variable d’API. Le tableau de la Figure III.25 comporte toutes les variables utilisées dans l’IHM avec un temps d’acquisition = 100 ms pour que la réaction soit rapide et immédiate lors de l’action sur n’importe qu’el bouton de l’IHM.

Nom	Type de données	Connexion	Nom API	Variable API	Adresse	Mode d'accès	Cycle d'acquisit.	Archivé
HMaB	Bool	HMI_Liais...	PLC_2	DB_ASRS.CMDE_Manuell...		<accès symbolique>	100 ms	
HMaD	Bool	HMI_Liais...	PLC_2	DB_ASRS.CMDE_Manuell...		<accès symbolique>	100 ms	
HMaG	Bool	HMI_Liais...	PLC_2	DB_ASRS.CMDE_Manuell...		<accès symbolique>	100 ms	
HMaH	Bool	HMI_Liais...	PLC_2	DB_ASRS.CMDE_Manuell...		<accès symbolique>	100 ms	
HMaD	Bool	HMI_Liais...	PLC_2	DB_ASRS.CMDE_Manuell...		<accès symbolique>	100 ms	
HMaG	Bool	HMI_Liais...	PLC_2	DB_ASRS.CMDE_Manuell...		<accès symbolique>	100 ms	
HMaH	Bool	HMI_Liais...	PLC_2	DB_ASRS.CMDE_Manuell...		<accès symbolique>	100 ms	
HPO.0	Bool	HMI_Liais...	PLC_2	DB_ASRS.Bouton_Positio...		<accès symbolique>	100 ms	
HPO.1	Bool	HMI_Liais...	PLC_2	DB_ASRS.Bouton_Positio...		<accès symbolique>	100 ms	
HPO.2	Bool	HMI_Liais...	PLC_2	DB_ASRS.Bouton_Positio...		<accès symbolique>	100 ms	
HPO.3	Bool	HMI_Liais...	PLC_2	DB_ASRS.Bouton_Positio...		<accès symbolique>	100 ms	
HP1.0	Bool	HMI_Liais...	PLC_2	DB_ASRS.Bouton_Positio...		<accès symbolique>	100 ms	
HP1.1	Bool	HMI_Liais...	PLC_2	DB_ASRS.Bouton_Positio...		<accès symbolique>	100 ms	
HP1.2	Bool	HMI_Liais...	PLC_2	DB_ASRS.Bouton_Positio...		<accès symbolique>	100 ms	
HP1.3	Bool	HMI_Liais...	PLC_2	DB_ASRS.Bouton_Positio...		<accès symbolique>	100 ms	
HP2.0	Bool	HMI_Liais...	PLC_2	DB_ASRS.Bouton_Positio...		<accès symbolique>	100 ms	
HP2.1	Bool	HMI_Liais...	PLC_2	DB_ASRS.Bouton_Positio...		<accès symbolique>	100 ms	
HP2.2	Bool	HMI_Liais...	PLC_2	DB_ASRS.Bouton_Positio...		<accès symbolique>	100 ms	
HP2.3	Bool	HMI_Liais...	PLC_2	DB_ASRS.Bouton_Positio...		<accès symbolique>	100 ms	
HReset	Bool	HMI_Liais...	PLC_2	DB_ASRS.CMDE_Manuell...		<accès symbolique>	100 ms	
HSelect	Bool	HMI_Liais...	PLC_2	DB_ASRS.Selection.Select...		<accès symbolique>	100 ms	

Figure III.25 : Table de variables de notre IHM.

III.5 Fonctionnement et visualisation du système

Pour montrer le bon fonctionnement de notre système en adéquation avec notre IHM, nous allons prendre un cas d'étude où nous allons voir lorsque la position « P0.0 » est sélectionnée. Nous considérerons toujours que le groupe pince est à sa position initiale. Dans le cas contraire, nous devons cliquer sur le bouton « Reset » de notre IHM (Figure III.26).

Pour sélectionner la position « P0.0 », le bouton « P0.0 » de notre IHM doit être actionné en 1^{er} (Figure III.26). Le contact « comparateur du circuit Ladder montre que la pince arrive à la position désirée en affichant même nombre en Haut et en bas (Figure III.26).

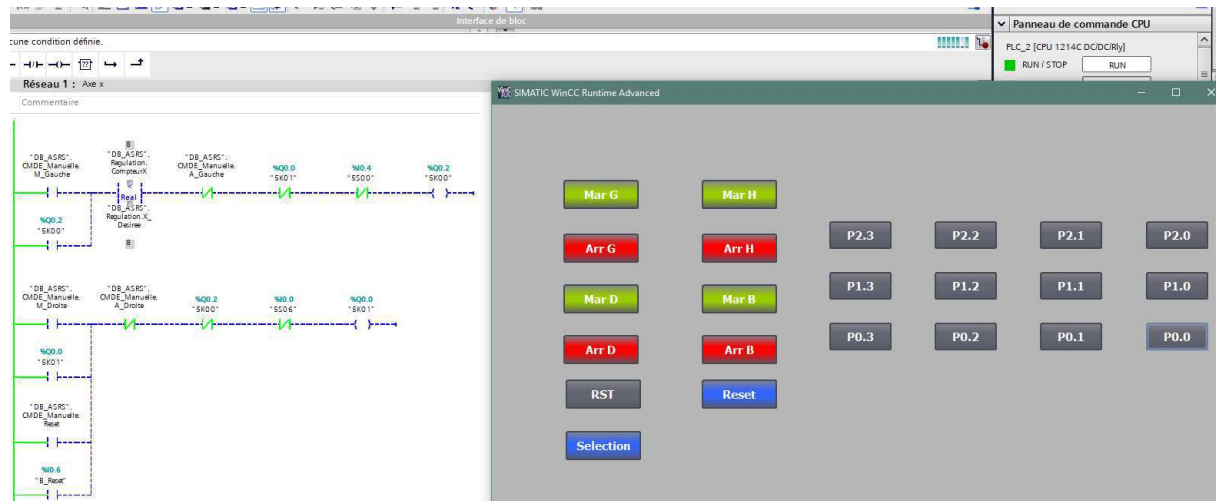


Figure III.26 : Schéma Ladder de l'axe X du système en fonctionnement en actionnant le bouton du position « P0.0 ».

Les Figures III.27 et III.28 montre que notre système fonctionne correctement. Vous constatez que le robot est arrêté juste à la position « P0.0 » et à la position « P1.1 » respectivement à notre demande.



Figure III.27 : Déplacement du robot vers la position « 0.0 ».

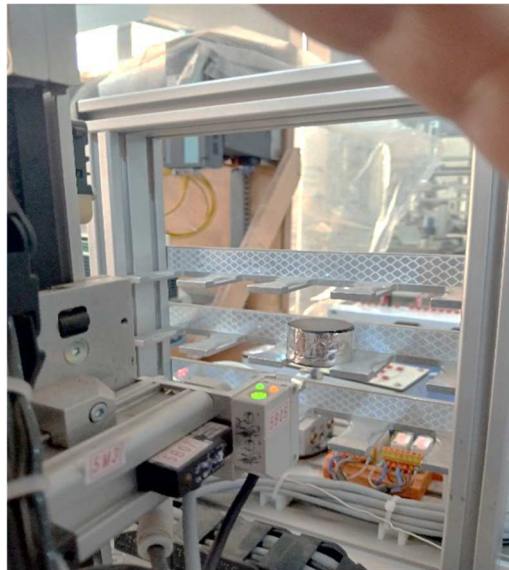


Figure III.28 : Déplacement du robot vers la position « 1.1 ».

Un autre exemple est pris dans le cas où nous avons sélectionné la position « P2.3 ». La Figure III.29 montre l'action sur le bouton « P2.3 » ainsi que le schéma Ladder qui affiche la position exacte demandée.

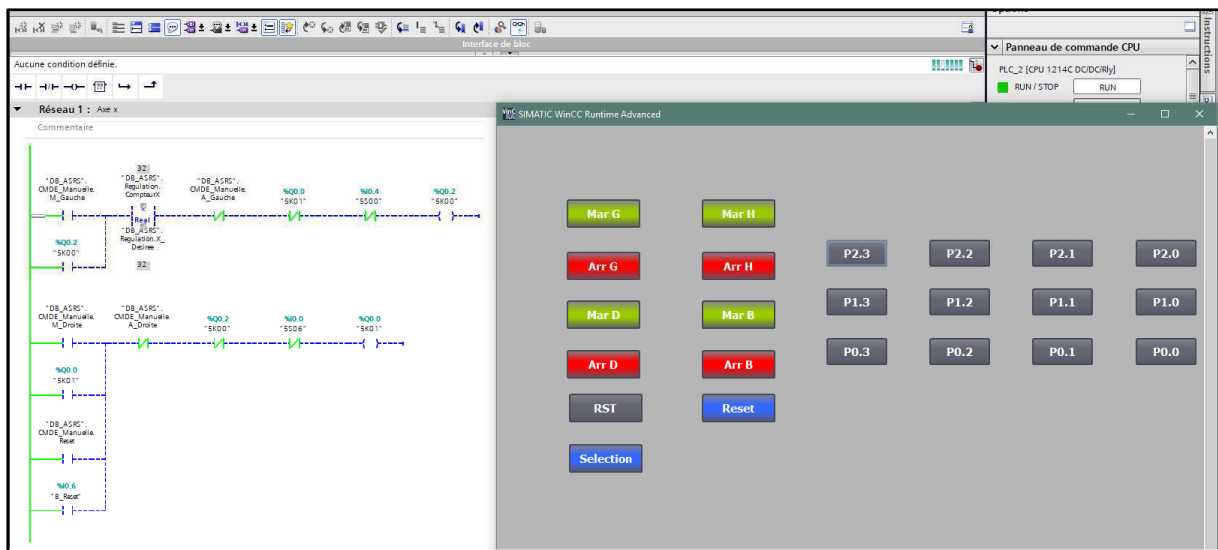


Figure III.29 : Schéma Ladder de l'axe X du système en fonctionnement à l'action du bouton du position « P2.3 ».

La Figure III.30 représente le fonctionnement des deux régulateurs PID (Compact des deux axes (PID_Horizontal et PID_Vertical) après sélection de la position « P2.3 » désirée en cliquant sur le bouton « P2.3 ». Dans les Figures III.31 et III.32 (02 PID_Compact en fonctionnement avec un zoom sur les valeurs affichées sur ces 02 PID), vous constatez que la sortie des deux PID_Compact tendent vers une valeur très faible (<11%) lorsque notre pince arrive à la position demandée. Cela montre que nos PID sont bien configurés. Il est de même pour les deux schémas Ladder de la Figure III.33 et III.34 des deux axes X et Z respectivement montrant le bon fonctionnement de notre régulation.

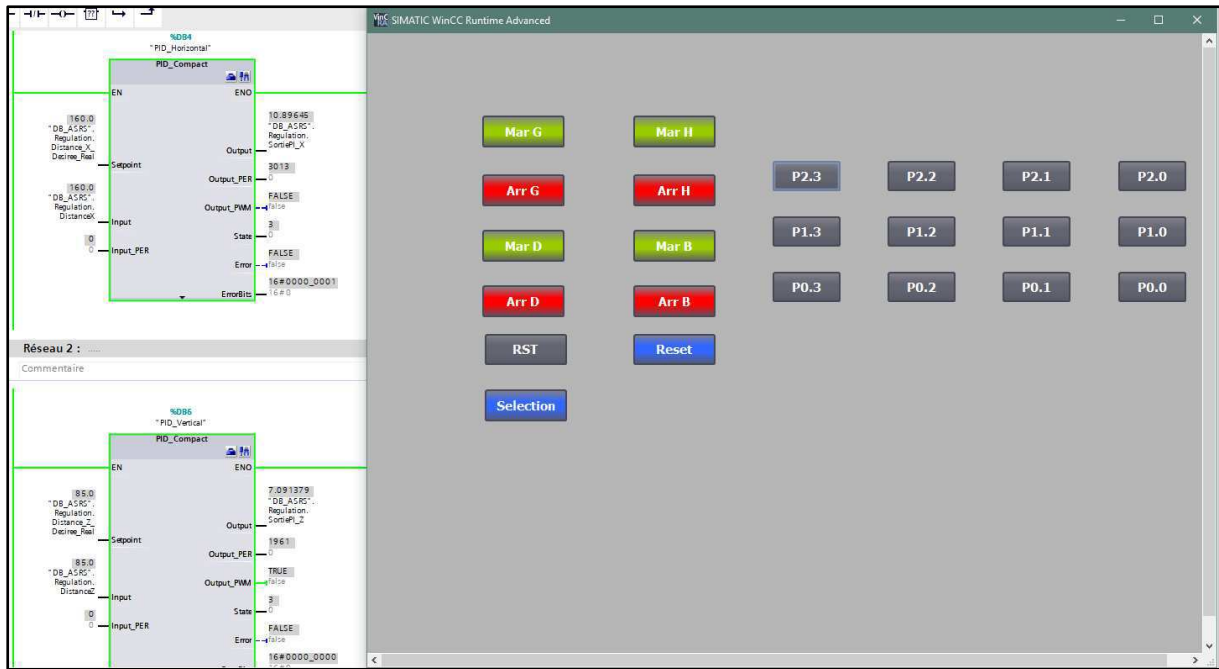


Figure III.30 : Fonctionnement des 2 « PID_Compact » pour les 2 axes à la sélection de la position « P2.3 » par le bouton.

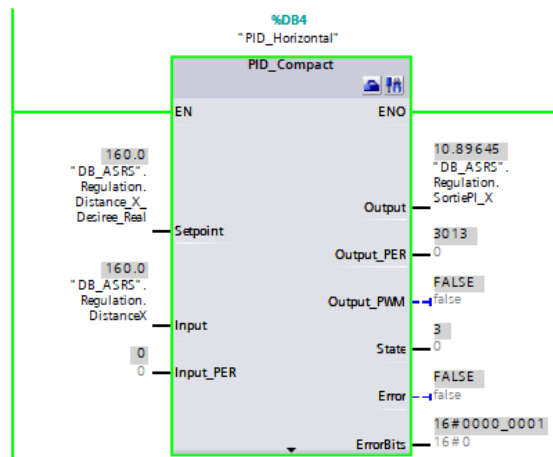


Figure III.31 : Représentation des valeurs du « PID_Compact » suivant la consigne pour l'axe X.

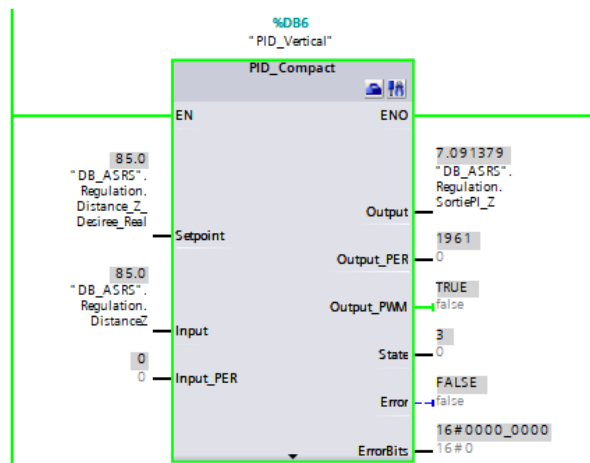


Figure III.32 : Représentation des valeurs du « PID_Compact » suivant la consigne pour l'axe Z.

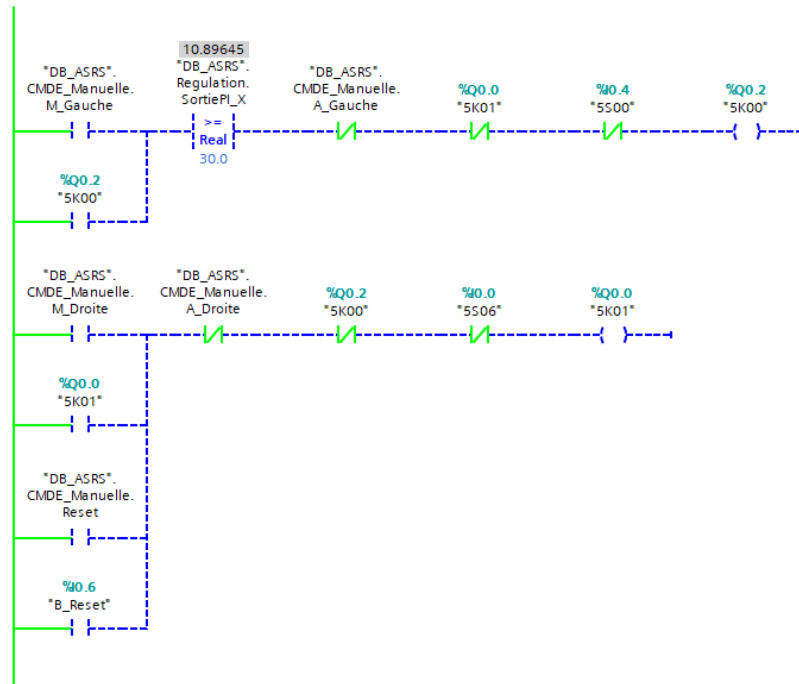


Figure III.33 : Schéma Ladder de l'axe X en fonctionnement à l'arrivée du système la position « 2.3 ».

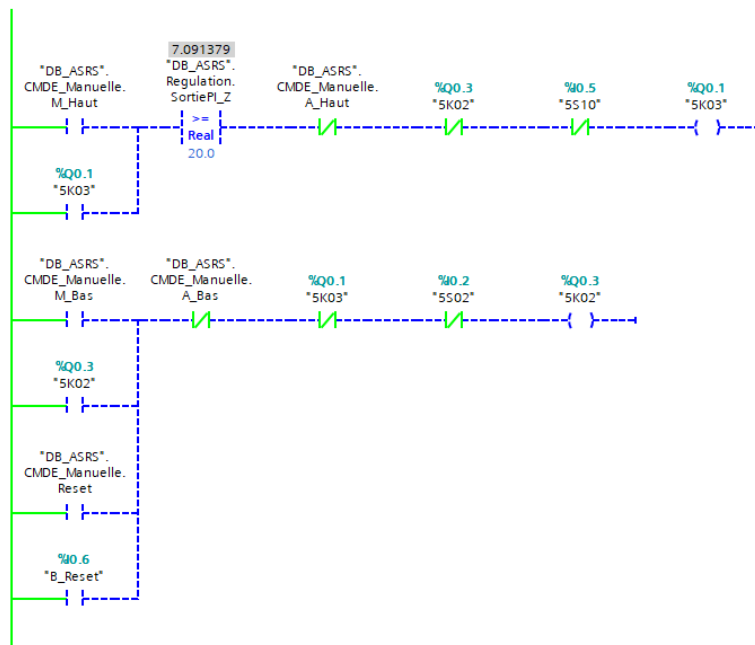


Figure III.34 : Schéma Ladder de l'axe Z en fonctionnement à l'arrivée du système la position « 2.3 ».

La Figure III.35 représente les paramètres des 02 PID de l'instruction PID_Compact pour les deux axes (X et Z). Nous constatons que ces paramètres sont importants car nous avons constaté que le système se déplace rapidement d'où la nécessité des valeurs du gain proportionnel élevé. Par contre, pour une régulation de position, il est naturel de choisir un temps d'intégration élevé pour réduire ou éliminer le dépassement.

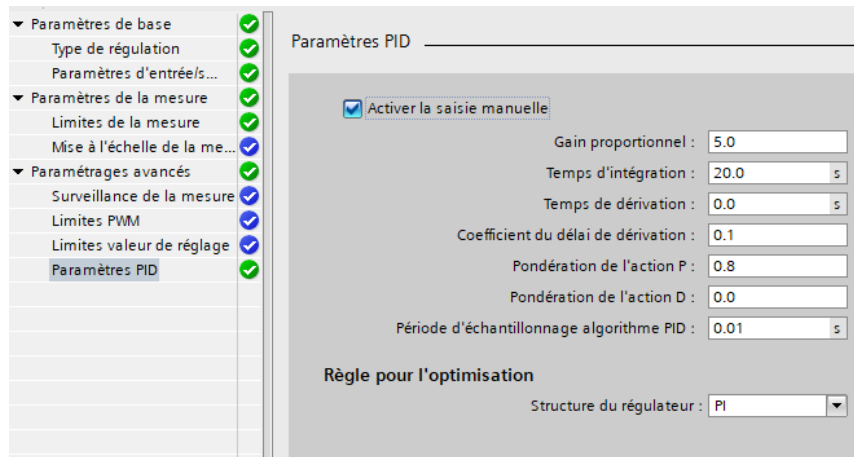


Figure III.35 : Paramètres PID pour l'instruction « PID_Compact ».

Enfin, la Figure III.36 représente la position exacte (très faible erreur ou erreur négligeable) de notre robot face à la position sélectionnée « P2.3. Cela montre que les 02 PID_Compact de notre API sont très performants, fiables et précis.



Figure III.36 : Déplacement du robot vers la position « 2.3 ».

III.6 Conclusion

Dans ce chapitre, nous avons décrit en détail notre contribution pour faire fonctionner notre station de stockage / déstockage AS / RS dans les meilleures conditions en utilisant une IHM développée et en introduisant le régulateur PID_Compact de l'API S7 1200. Il faut savoir que ce type d'API a été fabriqué pour améliorer les performances du régulateur PID, très utilisé dans l'industrie.

Nous avons introduit toutes les étapes de conception de notre IHM et aussi celles de notre programme Ladder. Nous avons montré pas à pas l'exécution de notre programme dans diverses figures. Les résultats montrés témoignent du bon fonctionnement de notre IHM utilisant un système PC-System avec WinCC comme logiciel et ainsi que notre programme de fonctionnement utilisant Step7 de Tia Portal v.15.1.

Conclusion générale

Cette étude a mis en évidence divers aspects fondamentaux et défis associés au contrôle d'un robot cartésien par API S7 1200. En étudiant différentes méthodes de contrôle et technologies émergentes, nous avons constaté le développement important dans ce domaine et les opportunités qu'il offre pour une automatisation efficace dans divers secteurs industriels.

Le travail présent est accentué autour de la commande du robot cartésien d'un système stockage et déstockage AS/RS du module d'emmagasinement à contrôle cartésien « Mod. MCS-620/EV, ElettronicaVeneta » avec l'API S7 1200 qui se trouve au niveau du laboratoire de recherche MELT de la faculté de Technologie, Université de Tlemcen.

Dans ce travail, nous avons fait la programmation sur logiciel Tia Portal de Siemens qui commande l'API S7 1200 pour contrôler et déplacer un robot cartésien de stockage et déstockage AS/RS vers une position sélectionnée par le biais d'une IHM développée sous le logiciel WinCC.

La programmation de ce système est faite en utilisant le langage Ladder où le programme est fait sur différents réseaux pour que le système puisse se déplacer horizontalement et verticalement en fonction de la position sélectionnée. Nous avons aussi utilisé une base de données pour améliorer la visibilité et faciliter sa programmation.

La régulation de position est faite avec l'instruction PID_Compact et son objet technologie de Tia Portal l'API S7 1200, cette instruction est optimisable. Ce choix est important pour donner une valeur à notre PFE car il touche essentiellement notre formation en Electrotechnique, spécialité Commandes Electriques.

La création d'IHM est réalisée avec le logiciel WinCC qui nous a permis de faire fonctionner notre système dans de meilleures conditions avec aucun apport physique extérieur. Ce type de fonctionnement rejoint la philosophie de l'Industrie 4.0.

En conclusion, cette étude met en évidence le potentiel considérable de l'utilisation d'API pour la commande d'un robot cartésien. En combinant la puissance de la programmation logicielle avec la précision et la polyvalence des robots cartésiens, cette approche ouvre de nouvelles perspectives pour l'automatisation industrielle et la robotique avancée.

Références bibliographiques

- [3] Pegon, M., & Riah, K. (2019). LIVRE BLANC : Industrie 4.0 : pour une transformation réussie. France:EVAGroup.
<https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&cad=rja&uact=8&ved=2ahUKEwiltKFlv3uAhWmRhUIHYmBWQQFjAAegQIAxAD&url=https%3A%2F%2Fblogevagroup.azureedge.net%2Fwp-content%2Fuploads%2F2020%2F01%2FLivre-Blanc-Industrie-4.0.pdf&usg=AOvVaw2HdQTGOUN7POYrOQfPrvyl>
- [4] Rüßmann, M., & Lorene, L. Bost. Consult (2015). Industry 4.0: The Future of Productivity and Growth in Manufacturing Industries (bcg.com).
- [5] temesist.systeme automatisés de stockage et de récupération. Systèmes automatisés de stockage et de récupérations AS/RS | temesist depo ve raf sistemleri (temesist.com) .
- [6] Metahri, D. (2019). « Contribution à l'évaluation des performances d'un flow-rack AS/RS à chute libre ». Université d'Oran 2 Mohammed Ben Ahmed, Oran.
- [7] Khalid HACHEMI, 2008. « Pilotage dynamique d'un système automatisé de stockage/déstockage à convoyeur gravitationnel », Université Abou Bekr Belkaïd, Tlemcen
- [8] Sari Z. 2003. « Modélisation, Analyse et Evaluation des Performances d'un AS/RS à Convoyeur Gravitationnel » Thèse de Doctorat d'état, Université de Tlemcen, Algérie.

Références web graphiques

- [1] Introduction aux systèmes automatisés de stockage/déstockage AS/RS (clicours.com)
- [2] [https://www.google.com/imgres?q=Robots%20manipulateurs%20\(ou%20robots%20articul%C3%A9s\)&imgurl=https%3A%2F%2Fwww.axesindustries.com%2Fimages%2Fnp%2F9591FR.JPG&imgrefurl=https%3A%2F%2Fwww.axesindustries.com%2Fdetails.php%2Fid%2F9591%2Ftous-les-fabricants-de-robots-a-bras-articles.html&docid=rfaLr0Jf5ceAwM&tbnid=Iz9g3I9_I3NYTM&vet=12ahUKEwj1u4W736GGAxUuQfEDHbhXC6MQM3oECB8QAA..i&w=400&h=358&hcb=2&ved=2ahUKEwj1u4W736GGAxUuQfEDHbhXC6MQM3oECB8QAA](https://www.google.com/imgres?q=Robots%20manipulateurs%20(ou%20robots%20articul%C3%A9s)&imgurl=https%3A%2F%2Fwww.axesindustries.com%2Fimages%2Fnp%2F9591FR.JPG&imgrefurl=https%3A%2F%2Fwww.axesindustries.com%2Fdetails.php%2Fid%2F9591%2Ftous-les-fabricants-de-robots-a-bras-articles.html&docid=rfaLr0Jf5ceAwM&tbnid=Iz9g3I9_I3NYTM&vet=12ahUKEwj1u4W736GGAxUuQfEDHbhXC6MQM3oECB8QAA..i&w=400&h=358&hcb=2&ved=2ahUKEwj1u4W736GGAxUuQfEDHbhXC6MQM3oECB8QAA)
- [9] https://www.google.com/imgres?q=Robots%20SCARA%20&imgurl=https%3A%2F%2Fupload.wikimedia.org%2Fwikipedia%2Fcommons%2F0%2F09%2FSCARA_robot_2R.png&imgrefurl=https%3A%2F%2Fen.wikipedia.org%2Fwiki%2FSCARA&docid=SMa-8LJnAapE-M&tbnid=2ZtHniGlvdUDWM&vet=12ahUKEwjU-Nr436GGAxW1cKQEHUaODSgQM3oFCIQBEAA..i&w=1080&h=1037&hcb=2&ved=2ahUKEwjU-Nr436GGAxW1cKQEHUaODSgQM3oFCIQBEAA

- [10] <https://www.google.com/imgres?q=Robots%20cart%C3%A9siens&imgurl=https%3A%2F%2Fwww.kinetic-systems.fr%2Fwp-content%2Fuploads%2F2018%2F12%2FCartesien-L.jpg&imgrefurl=https%3A%2F%2Fwww.kinetic-systems.fr%2Fproduit%2Frobot-cartesien-manipulateur-portique-en-l%2F&docid=FzJGwSEAi836PM&tbid=0hTbxpPI52l8cM&vet=12ahUKEwi77uC24KGGAxUdUqQEHUdgAwEQM3oFCIEBEAA..i&w=768&h=768&hcb=2&ved=2ahUKEwi77uC24KGGAxUdUqQEHUdgAwEQM3oFCIEBEAA>
- [11] https://www.google.com/imgres?q=Robots%20collaboratifs&imgurl=https%3A%2F%2Fwww.axesindustries.com%2Fimages%2Fnp%2F10447FR.JPG&imgrefurl=https%3A%2F%2Fwww.axesindustries.com%2Fdetails.php%2Fid%2F10447%2Fcobots-industriels-lourdes-charges-universal-robots.html&docid=ILKEtQohCtil_M&tbid=TI9IDlyS65eyhM&vet=12ahUKEwjwiNmI4qGGAxXzRaQEHZnkDfEQM3oECHKQAA..i&w=400&h=457&hcb=2&ved=2ahUKEwjwiNmI4qGGAxXzRaQEHZnkDfEQM3oECHKQAA
- [12] <https://fr.demotor.net/moteurs-electriques/moteur-courant-continu>
- [13] https://media.automation24.com/manual/fr/61777246_s71200_System_Manual_fr-FR_fr-FR.pdf
- [14] Siemens sas 6AG1214-1HG40-4XB0 | SIPLUS S7-1200 CPU 1214C DC/DC/RLY | Rexel France
- [15] Siemens TIA Portal : tout savoir sur ce logiciel de programmation (umvie.com)
- [16] L'industrie 4.0 bouleverse les interactions homme-machine (journaldunet.com)
- [17] <https://fr.wikipedia.org/wiki/WinCC#:~:text=SIMATIC%20WinCC%20est%20un%20syst%C3%A8me,avec%20Siemens%20PCS7%20et%20Teleperm.>
- [18] <https://www.maxicours.com/se/cours/types-de-moteur-a-courant-continu/>.
- [19] <https://energieplus-lesite.be/techniques/ascenseurs7/moteur-a-courant-continu/>
- [20] <https://www.abcelectronique.com/annuaire/montages/cache/3074/rapport.pdf>
- [21] https://fr.wikipedia.org/wiki/Pont_en_H#:~:text=Utilisation%20avec%20les%20moteurs%20%C3%A0%20courant%20continu,-Pont%20en%20H&text=Le%20pont%20en%20H%20permet,bornes%20du%20moteur%20par%20hachage.
- [22] Modulation de largeur d'impulsion — Wikipédia (wikipedia.org)
- [23] Comprendre L'effet D'une Électronique DE Commande PWM Sur Un Moteur Cc Sans Balais (portescap.com) Comprendre L'effet D'une Électronique DE Commande PWM Sur Un Moteur Cc Sans Balais (portescap.com)
- [24] https://media.automation24.com/manual/fr/61777246_s71200_System_Manual_fr-FR_fr-FR.pdf

[25] <https://www.mabeo-industries.com/A-509587-siemens-unite-centrale-automate-compacte-simatic-s7-1200>

الملخص

الهدف هو التحكم في الروبوت الديكارتي لنظام استرجاع التخزين AS/RS باستخدام S7 1200 API، حيث يتم برنامج التحكم في برنامج Tia Portal مع لغة Ladder ويتم تنظيم موضع النظام من خلال التعليمات PID_Compact وملحقاته التكنولوجي. ثم تم تطوير HMI ضمن برنامج WinCC لقيادة النظام والتحكم فيه باستخدام أزرار مختلفة. سيتم استخدام هذه الأزرار لاختيار المنتج أو بشكل خاص موضع المنتج. يتم حفظ التصور لعدة حالات اختيار.

الكلمات المفتاحية : التخزين، الإزالة، S7 1200 API، (واجهة الآلة البشرية) HMI، وحدة التحكم PID، تنظيم الموضع.

Abstract

The objective is to control a Cartesian robot of the AS/RS storage retrieval system with the S7 1200 API, where the control program is done in Tia Portal software with Ladder language and the regulation of the position of the system is done with the PID_Compact instruction and its technology object, and created the HMI supervision tool under WinCC software to command and control the system with the buttons, and then load the program into the S7 1200 API and select the position with the HMI to move the system to the selected position, and finally visualize the movement of the robot.

Keywords : Storage, Removal, PLC S7 1200, HMI, PID controller, Regulation of Position.

Résumé :

L'objectif est de commander un robot cartésien du système de stockage / déstockage AS/RS avec l'API S7 1200, où le programme de commande se fait dans logiciel Tia Portal avec langage Ladder. Aussi, la régulation de position du système se fait par l'objet technologie du S7 1200 qui est le PID_Compact. Un IHM a été développé sous le logiciel WinCC pour commander et contrôler le système avec différents boutons. Ces boutons seront utilisés pour la sélection du produit ou particulièrement de la position du produit. Une visualisation est enregistrée pour plusieurs cas de sélection.

Mots clés : Stockage, Déstockage, API S7 1200, IHM, Régulateur PID, Régulation de Position.