

UNIVERSITÉ IBN-KHALDOUN DE TIARET

FACULTÉ DES SCIENCES APPLIQUEES

DÉPARTEMENT DE GENIE ELECTRIQUE



MEMOIRE DE FIN D'ETUDES

Pour l'obtention du diplôme de Master

Domaine : Sciences et Technologie

Filière : Automatique

Spécialité : Automatique et informatique Industrielle

THÈME

Commande d'un bras de robot (Rob 03) par automate S7 - 1200

Préparé par :

Mokhtari Rekia

Souak Mimouna

Devant le Jury :

Nom et prénoms	Grade	Qualité
R. OURED	Pr	Président
E. BENAATIA	Pr	Examineur 1
M. KOUADRIA	Pr	Examineur 2
H. BOUMEDIENE	MCB	Encadrant
S. HASSAINE	Pr	Co-Encadrant

Dédicace

Je dédie ce travail à mes très chers parents qui ont tant prié à mon réussite. De soutien moral. Je les remercie de m'avoir encouragée Et aidé à devenir ce que Je suis. Et A mes frère, mes sœurs, et à toute la famille. Ames amis qui ont été toujours présent.

Remerciement

Tout d'abord, je tiens à remercier le grand Dieu de m'a donné la force, la patience, la volonté, le courage et la santé pour terminer ce travail.

Je remercie mes parents et ma famille qui m'aide et m'encourage durant mes longues années d'études.

Je tiens à exprimer ici tous mes respects et toutes mes reconnaissances à mon l'encadrant Mr. Boumediene et sous L'encadrant Mr. Hassaine

D'avoir accepté de m'encadrer

Zui a cru en mes capacités

Pour ses conseils et ses encouragements

Et surtout sa patience

Je n'oublie pas les membres de jury et toute l'équipe de Laboratoire

D'informatique et électrique

Enfin, Merci à toutes les membres de ma promotion et toutes les personnes qui mon aide moralement et matériellemen

SOMMAIRE

Introduction Générale.....	1
Introduction générale.....	2
Chapitre I:.....	3
Les Automates Programmable Industriels	3
I.1. Introduction.....	4
I.2. Les systèmes automatisés de production	4
I.3. Structure d'un système automatisé	4
I.3.1. Partie opérative	4
I.3.2. Partie commande.....	4
Figure I.1 Structure d'un système automatisé.....	5
I.4. Les avantages et les inconvénients d'un système automatisé.....	5
I.5. Automate programmable industriel	5
I.5.1. Historique.....	5
I.5.2. Définition d'un <i>API</i>	6
I.6. Les types des automates programmables	6
I.6.1. Automate programmable monobloc.....	6
I.6.2. Automate programmable modulaire	7
I. 7. Choix d'un <i>API</i>	7
I.8. Avantages de l' <i>API</i>	7
I.9. Description de l'Automate <i>S7-1200</i>	8
I.10. Vue du portail et vue du projet	9
I.10.1. Vue du portail	9
I.10.2. Vue du projet.....	9
I.10.3. Configuration et paramétrage du matériel	10
I.10.4. Adressage des E/S.....	11
I.10.5. Adresse Ethernet de la <i>CPU</i>	11

SOMMAIRE

I.10.6. Implantation du programme.....	12
I.12. Description générale de SIMATIC HMI	13
I.12.1. Les différents modèles de la famille SIMATIC IHM.....	14
I.12.2. Choix de l'Interface Homme-Machine (<i>IHM</i>).....	14
I.12.3. Présentation générale du produit.....	14
I.13. Conclusion	15
Chapitre II	16
Modélisation Géométriques Direct et Inverse.....	16
II.1. Introduction	17
II.2. Modélisation géométrique	17
II.2.1.Les modèles géométriques direct et inverse	17
II.2.2.Les modèles cinématiques direct et inverse	17
II.2.3.Les modèles dynamiques.....	17
II.3. Modèle Géométrique Direct	17
II.3.1.Définition.....	17
II.3.3. Règle de la main droite	18
II.4. Convention de Denavit-Hartenberg (DH) :	19
II.5. Les paramètres de Denavit-Hartenberg	19
II.6. Obtention du modèle géométrique direct	20
II.7. Tableau <i>Denavit-Hartenberg</i>	20
II.8. Calcul du MGD du bras <i>ROB3</i>	20
II.9. Modèle géométrique inverse (<i>MGI</i>)	22
II.9.1. Calcule du <i>MGI</i> du bras <i>ROB3</i>	22
II.10. Conclusion	24
Chapitre III:	25
Description et conception de <i>Rob3</i>	25
III.1. Introduction	26

SOMMAIRE

III.2. Espace du travail du robot manipulateur <i>ROB3</i>	26
III .3. Les actionneurs.....	27
III .3.1. Les moteurs à courant continu.....	27
III .3.1.1. Transmissions et systèmes réducteurs de vitesse	28
III.3.2. Les servomoteurs.....	28
III .4. La carte de puissance.....	29
III .4.1. Schéma électronique de la carte de puissance.....	29
III .5. Eléments constituant la carte de puissance.....	30
III .5.1. Le circuit <i>L298N</i>	30
III .5.1.1. Définition des ponts en H.....	31
III .5.1.2. Le principe d'un pont en <i>H</i>	32
III .5.1.3. Table de vérité du <i>L298N</i> (fonctionnement)	33
III .5.1.5. Les Avantages du <i>L298N</i>	34
III .5.2. Les diodes 1N4007.....	34
III .5.3. Le circuit d'alimentation.....	35
III.5.4. Régulateur de tension LM 7805	35
III.5.4.1. Caractéristiques du régulateur 7805	36
III.5.4.2. Application du régulateur 7805	36
III.6. Le signal de commande Pulse Train Output (<i>PTO</i>)	37
III.6.1. Concept.....	37
III.6.2. Performances	37
III.6.3. Comment fonctionne l'interface d'impulsion?.....	37
III.6.4. Relation entre le type de signal et le sens de déplacement.....	37
III.6.5. Inverser le sens	38
III.7. Conclusion.....	38
Chapitre IV	39
Partie Expérimentale	39

SOMMAIRE

V.1. Introduction	40
V.2 Cahier de charge	40
V.3 Les étapes de trie	40
V.4 Description du système	40
V.4.1 Phase de construction	41
V.4.2 Phase de programmation	42
V.4.3 Phase finale.....	49
IV.5. Conclusion	53
Conclusion générale	54

Liste des Figures

Figure I. 1 : Système automatisé	Erreur ! Signet non défini.
Figure I. 2 : Automate programmable S7-1200	6
Figure I. 3 : Automate programmable industrielle monobloc	7
Figure I. 4 : Automate programmable industrielle modulaire.....	7
Figure I. 5 : Automate programmable S7-1200	9
Figure I. 6 : Vue du portail.....	10
Figure I. 7 : Vue du projet	10
Figure I. 8 : Ajout d'un appareil.....	11
Figure I. 9 : Adressage des E/S.	11
Figure I. 10 : Adresse Ethernet de la CPU.	12
Figure I. 11 : Blocs du programme.	12
Figure I. 12 : L'interface de supervision KTP700 Basic.....	13
Figure I. 13 : La structure des appareils PROFINET KTP700 Basic.....	14
Figure II. 1 : Schéma de la main droite	
Erreur ! Signet non défini.	
Figure II. 2 : Représentation des repères sur le ROB3.....	19
Figure II. 3 : Paramètres géométriques du robot ROB3.....	20
Figure II. 4 : Paramètres géométriques du robot ROB3.....	22
Figure III. 1 Bras de robot manipulateur ROB3.....	26
Figure III. 2 : Espace du travail du robot manipulateur ROB3	26
Figure III. 3 : Les dimensions du robot ROB3 (vue de face).....	27
Figure III. 4 : Les dimensions du robot ROB3 (vue de haut).....	27
Figure III. 5 : Moteur à courant continu «Bühler».....	28
Figure III. 6 : les pignons	28
Figure III. 7 : Servomoteurs «Multiplex»	29
Figure III. 8 : Schéma de la carte de puissance (proteus).....	29
Figure III. 9 : Vue en 3D	30
Figure III. 10 : Vue de face	31
Figure III. 11 : schéma interne du L298N	31
Figure III. 12 : schéma d'un pont en H	32
Figure III. 13 : Pont en H les deux configurations possibles	32
Figure III. 14 : Diode 1N4007.....	35
Figure III. 15 : Fusible.....	35

Liste des Figures

Figure III. 16 : Régulateur de tension LM 7805	36
Figure III. 17 : Schéma représentant les deux sortie un signal PTO. [27]	38
Figure IV. 1 : Description du système	40
Figure IV. 2 : La Simulation de diviseur par Proteus ISIS	41
Figure IV. 3 : La réalisation d'un diviseur de tension.....	42
Figure IV. 4 : La réalisation de la carte automate – servo.....	42
Figure IV. 5 : Le choix de PLC	43
Figure IV. 6 : La configuration de générateur d'impulsion.	43
Figure IV. 7 : La configuration de générateur d'impulsion.	44
Figure IV. 8 : La configuration de générateur d'impulsion.	44
Figure IV. 9 : La configuration d'un objet axe.....	45
Figure IV. 10 : Configuration de moteur en sens de rotation directe (sens positif	45
Figure IV. 11 : Insérer l'instruction Motion contrôle.....	46
Figure IV. 12 : Bloc "MC_Power"	46
Figure IV. 13 : Bloc "MC_MoveRelative"	47
Figure IV. 14 : La simulation de Bloc "MC_MoveRelative".....	47
Figure IV. 15 :	Erreur ! Signet non défini.
Figure IV. 16 : Bloc TP Tim	48
Figure IV. 17 : Bloc CTU.....	48
Figure IV. 18 : La position initiale de bras.	49
Figure IV. 19 : détection de l'objet et déplacement du bas et fermeture de la pince	49
Figure IV. 20 : déplacement de l'épaule vers le haut et rotation de la bas	50
Figure IV. 21 : déplacement de l'épaule vers le bas et ouverture de la pince.....	50
Figure IV. 22 : déplacement du robot à sa position initiale pour exécuter une autre tâche	51
Figure IV. 23 : Interface HMI	51

Liste des tableaux

Tableau II. 1 : Denavit-Hartenberg	20
Tableau III. 1 : Table de vérité du L298N (fonctionnement).....	34
Tableau IV . 1 : Table de Mnémonique des entrées.....	51
Tableau IV . 2 : Table de Mnémonique des sorties.....	52
Tableau IV . 3 : Table de Mnémonique HMI.....	52

Liste Des Abréviations

API: Automate Programmable Industrielle.

CPU: Central Processing Unit

HMI: Human Machine Interface

PLC: Programmable Logic Controller

MGD : Modèle Géométrique Direct

MGI : Modèle Géométrique Inverse

PTO : Pulse Train Output

SCADA : Supervisory Control And Data Acquisition

LADDER : Ladder Diagram

OB : blocs d'organisation

FB : blocs fonctionnels

FC: Les fonctions sont des blocs de code sans mémoire.

DB: Les blocs de données .

Introduction Générale

Introduction générale

Aujourd'hui, il existe plusieurs types de robots conçus pour des tâches bien spécifiques. Dans ce travail du mémoire, nous nous sommes intéressés aux robots manipulateurs pour des applications en robotique industrielle. Utilisés dans l'industrie depuis plusieurs dizaines d'années, les robots industriels actuellement en service sont des robots de type manipulateur. Ils sont bien implantés dans les procédés de fabrication modernes et sont employés pour augmenter le volume de production et améliorer la qualité du produit. Dans les chaînes de montages de l'industrie automobile, ils remplacent les ouvriers dans les tâches pénibles, répétitives ou dangereuses (peinture, soudure...). Le bras manipulateur est un système de positionnement où les forces agissant au niveau des articulations sont produites par des actionneurs. Ces derniers peuvent être de types électrique, hydraulique ou pneumatique.

Dans notre mémoire nous intéressons particulièrement des bras de robot qui utilisent des moteurs à courant continu et des servomoteurs.

L'objectif principal de notre travail est de commander un bras manipulateur à 3 degrés de liberté commandé par un Automate Programmable Industriel (*API*). Ce dernier a pour fonction de traiter les informations entrantes pour émettre des ordres de sorties en fonction d'un programme. Il s'agit d'un automate programmable à usage industriel destiné à commander les systèmes en temps réel.

Pour aboutir à cette fin, l'étude est organisée en quatre parties:

Dans le premier chapitre on présentera les automates programmables plus précisément l'*API S7-1200* avec tout l'aspect intérieur et extérieur associés, ainsi que le logiciel de programmation **TIA PORTAL V16**. Le lecteur trouvera également un détail basic sur l'Interface Homme/Machine (**IHM**) type **KTP700** pour la supervision en temps réel.

Le chapitre 2, quant à lui, est consacré à la modélisation géométrique directe et indirecte.

Dans le chapitre 3 on présentera le fonctionnement de la carte de puissance basée sur les hacheurs en ponts à quatre quadrants (L298N). L'accent est mis sur la génération des signaux de type Pulse Train Output (**PTO**), permettant de piloter convenablement les **DC** moteurs composant le bras du robot.

Le dernier chapitre s'intéresse à la partie expérimentale dont laquelle de la carte d'interface entre le bras et l'automate est conçue. Le programme de contrôle de l'*API* est écrit sous TIA portal V16 de Siemens en langage ladder.

- Enfin, on clôture notre travail avec une conclusion générale..

Chapitre I:
Les Automates Programmable Industriels

I.1. Introduction

Les procédés industriels ont connu un grand essor au cours des dernières années grâce à l'automatisme. Cela exige des entreprises à automatiser leurs processus de production, afin d'assurer leur pérennité en offrant la compétitivité, tout en améliorant les conditions de travail de leur personnel et en supprimant les tâches pénibles et répétitives. Au début, l'Automate Programmable servait uniquement à réaliser les fonctions logiques et séquentielles et au contrôle de commande individuelle des machines-rôle de séquenceur électrique, pneumatique ou électronique. Aujourd'hui, outre ces fonctions logiques, il gère le traitement numérique, la gestion de la production, le dépannage, la messagerie et le traitement. Dans ce chapitre, nous allons décrire les systèmes automatisés, les automates programmables ainsi que la programmation associée.

I.2. Les systèmes automatisés de production

L'objectif de l'automatisation des systèmes est de produire, en ayant recours le moins possible à l'homme, des produits de qualité et ce pour un coût le plus faible possible. Un système automatisé est un ensemble d'éléments en interaction, organisés dans un but précis: agir sur une matière d'œuvre afin de lui donner une valeur ajoutée. Le système automatisé est soumis à des contraintes énergétiques, de configuration, de réglage et d'exploitation qui interviennent dans tous ces modes de marche et d'arrêt [16].

I.3. Structure d'un système automatisé

Tout système automatisé peut se décomposer en deux parties:

I.3.1. Partie opérative

Elle agit sur la matière d'œuvre afin de lui donner sa valeur ajoutée. Les actionneurs (moteurs, vérins) agissent sur les effecteurs (les outils) du système qui agit à son tour sur la matière d'œuvre. Les capteurs et détecteurs permettent d'acquérir les divers états du système.

I.3.2. Partie commande

Elle donne les ordres de fonctionnement à la partie opérative. Les pré-actionneurs permettent de commander les actionneurs; ils assurent le transfert d'énergie entre la source de puissance (réseau électrique, pneumatique ...) et les actionneurs. Exemple: contacteur, distributeur. Ces pré-actionneurs sont commandés à leur tour par le bloc de traitement des informations (*API*). Celui-ci reçoit les consignes du pupitre de commande (opérateur) et les informations de la partie opérative transmises par les capteurs et détecteurs. En fonction de ces consignes et de son programme de gestion des tâches implanté dans un *API* ou réalisé par des relais (on parle de logique câblée. Cette partie commande les pré actionneurs et renvoie

des informations au pupitre de signalisation ou à d'autres systèmes de commande ou de supervision en utilisant un réseau et un protocole de communication [17].

Un système automatisé peut être représenté comme suit :

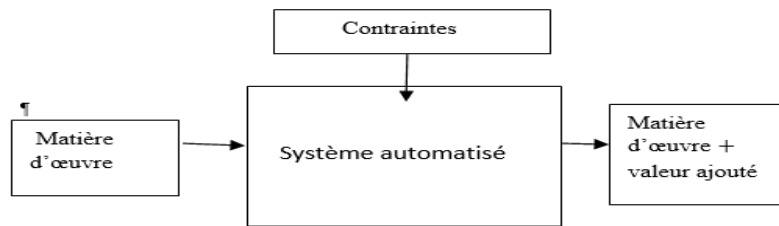


Figure I.1 Structure d'un système automatisé.

I.4. Les avantages et les inconvénients d'un système automatisé

Les avantages d'un système automatisé peuvent être résumés dans les points suivants:

- La capacité de production accélérée.
- L'aptitude à convenir à tous les milieux de production ;
- La souplesse d'utilisation.
- La création de postes d'automaticiens.

Quant aux inconvénients, on enregistre ce qui suit:

- Le coût élève du matériel, principalement avec les systèmes hydrauliques.
- La maintenance doit être structurée.
- La suppression d'emplois.
-

I.5. Automate programmable industriel

I.5.1. Historique

Les Automates Programmables Industriels (*API*) sont apparus aux Etats-Unis vers 1969 où ils répondaient aux désirs des industries de l'automobile de développer des chaînes de fabrication automatisées qui pourraient suivre l'évolution des techniques et des modèles fabriqués.

Un Automate Programmable Industriel (*API*) est une machine électronique programmable par un personnel non informaticien et destiné à piloter en ambiance industrielle et en temps réel des procédés industriels. Un automate programmable est adaptable à un maximum d'application, d'un point de vue traitement, composants, langage. C'est pour cela qu'il est de construction modulaire. Il est en général manipulé par un personnel électromécanicien. Le développement de l'industrie à entraîner une augmentation constante des fonctions électroniques présentes dans un automatisme; c'est pour ça que l'*API* s'est

substitué aux armoires à relais en raison de sa souplesse dans la mise en œuvre, mais aussi parce que dans les coûts de câblage et de maintenance devenaient trop élevés. [1]

I.5.2. Définition d'un API

L'automate programmable industriel (*API*), ou en anglais "Programmable Logic Controller (*PLC*)" est une machine électronique programmable destinée à piloter dans une ambiance industrielle et en temps réel des procédés logiques séquentielles. Autrement dit, un utilisateur (censé être un automaticien) l'utilise pour le contrôle et essentiellement la commande d'un procédé industriel en assurant l'adaptation nécessaire entre tout ce qui est de grande puissance par rapport à ce qui est de faible puissance côté commande. Son objectif principal est de rendre tout le mécanisme de type "laisser-faire-seul": le système contrôle ses sorties, décide et agit sur ses entrées afin de maintenir le fonctionnement comme prévu par l'utilisateur. C'est le principe de l'automatisme. [2]



Figure I. 1 : Automate programmable S7-1200.

I.6. Les types des automates programmables

Il existe deux types d'automate programmable

- Le type monobloc.
- Le type modulaire.

I.6.1. Automate programmable monobloc

L'automate programmable monobloc possède généralement un nombre d'entrées et de sorties restreint et son jeu d'instructions ne peut être augmenté. Bien qu'il soit parfois possible d'ajouter des extensions d'entrées/sorties, le type monobloc a pour fonction de résoudre des automatismes simples faisant appel à une logique séquentielle et utilisant des informations tout-ou-rien [3].



Figure I. 2: Automate programmable industriel mono bloc.

I.6.2. Automate programmable modulaire

Le type modulaire est adaptable à toutes les situations. Selon le besoin, des modules d'entrées/sorties analogiques sont disponibles en plus de modules spécialisés tels: *PID*, langage basic et Langage *C* etc.

La modularité des *API* permet un dépannage rapide et une plus grande flexibilité [3].

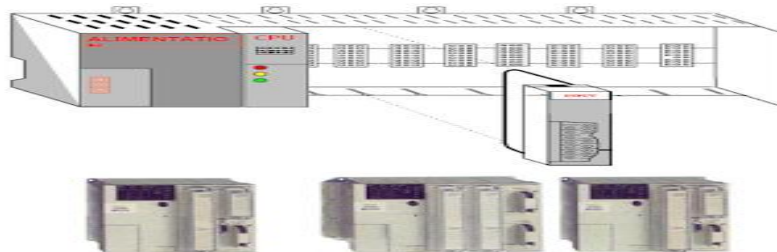


Figure I. 3 : Automate programmable industrielle modulaire.

I. 7. Choix d'un *API*

Le choix d'un *API* est fonction de la partie commande à programmer. On doit tenir compte de plusieurs critères. [4]

- Nombres d'entrées/sorties intégrés.
- Temps de traitement (scrutation).
- Capacité de la mémoire.
- Nombre de compteurs.
- Nombre de temporisateurs.

I.8. Avantages de l'*API*

L'utilisation de l'automate programmable dans le domaine industriel présente plusieurs avantages, on cite entre autres :

- **Moins de constituants**

La substitution des relais à un gain en volume, en encombrement et à la simplicité de l'emploi, particulièrement appréciées sur les machines simples.

- **Moins de câblage**

Les connections se réduisent au raccordement des capteurs aux entrées et des prés-actionneurs aux sorties. L'accès aux différents organes de l'automatisme, lots des modifications et des réglages, se trouve ainsi facile.

- **Plus de confort**

Le programme qui se substitue au câblage et l'ensemble des graphiques dont on peut le saisir, le modifier et l'archiver facilement grâce au terminal de programmation et de réglage. Ce programme peut être dupliqué pour les machines construites, ce qui se traduit par une éventuelle diminution des coûts.

- **Plus de fonctionnalités**

Pour les machines spéciales où leurs installations sont compliquées, l'automate programmable offre des fonctions d'automatisme spécifiquement intégrées.

- **Plus d'information**

La maintenance et la mise en place d'un automatisme est facile par la visualisation permanente de l'état des entrées/sorties, qui sont signalés par des voyants lumineux. Le dialogue entre l'homme et la machine est assuré par un terminal de programmation ainsi, le réglage est assuré grâce à son mode conversationnel et les messages affichés sur l'écran. [5]

I.9. Description de l'Automate S7-1200

Dans notre étude, le choix s'est articulé sur le *PLC SIEMENS S7-1200* de la famille *S7*. Le contrôleur 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 [8].

Le *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 le programme chargé, le *CPU* contient la logique nécessaire au contrôle et à la commande des appareils dans les applications.

Le *CPU* surveille les entrées et modifié les sorties conformément à la logique du programme de l'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 [6].

Le *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*, *RS4 85* ou *RS 232*. Pour ce projet, le *S7-1200* sera programmé à l'aide du logiciel *TIA PORTAL* sous Windows. [7]



Figure I. 4 : Automate programmable S7-1200.

- 1) Alimentation 24V.
- 2) Bordiers insérables pour un câblage utilisateur (derrière les caches plastiques).
- 3) Diodes électroluminescentes pour les E/S intégrées et le mode de fonctionnement de la *CPU*.
- 4) Connexion *TCP/IP* (sous le *CPU*).

I.10. Vue du portail et vue du projet

Lorsqu'on lance *TIA Portal*, l'environnement de travail se décompose en deux types de vues:

- La vue du portail: elle est axée sur les tâches à exécuter et sa prise en main est très rapide.
- La vue du projet: elle comporte une arborescence avec les différents éléments du projet, les éditeurs requis s'ouvrent en fonction des tâches à réaliser, données, paramètres et éditeurs Ils peuvent être visualisés dans une seule et même vue. [10]

I.10.1. Vue du portail

Chaque portail permet de traiter une catégorie de tâches (actions), la fenêtre affiche la liste des actions pouvant être réalisées pour la tâche sélectionnée, la figure ci-dessous représente une vue du portail. [10]

I.10.2. Vue du projet

L'élément "Projet" contient l'ensemble des éléments et des données nécessaires pour mettre en œuvre la solution d'automatisation souhaitée. La figure ci-dessous représente la vue du projet. [10]

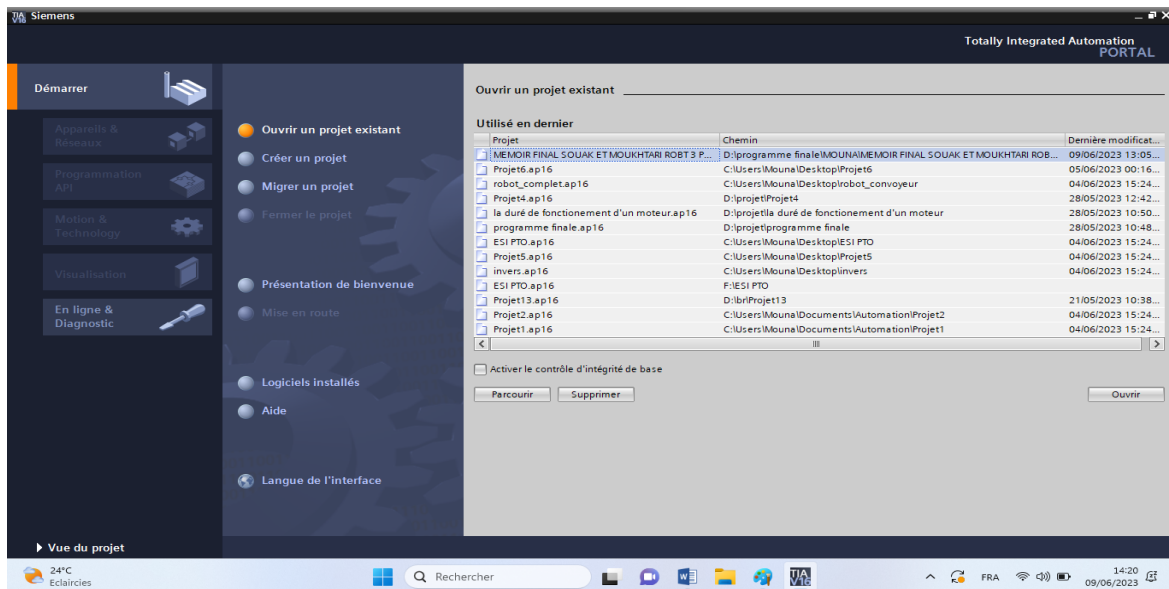


Figure I. 5 : Vue du portail.

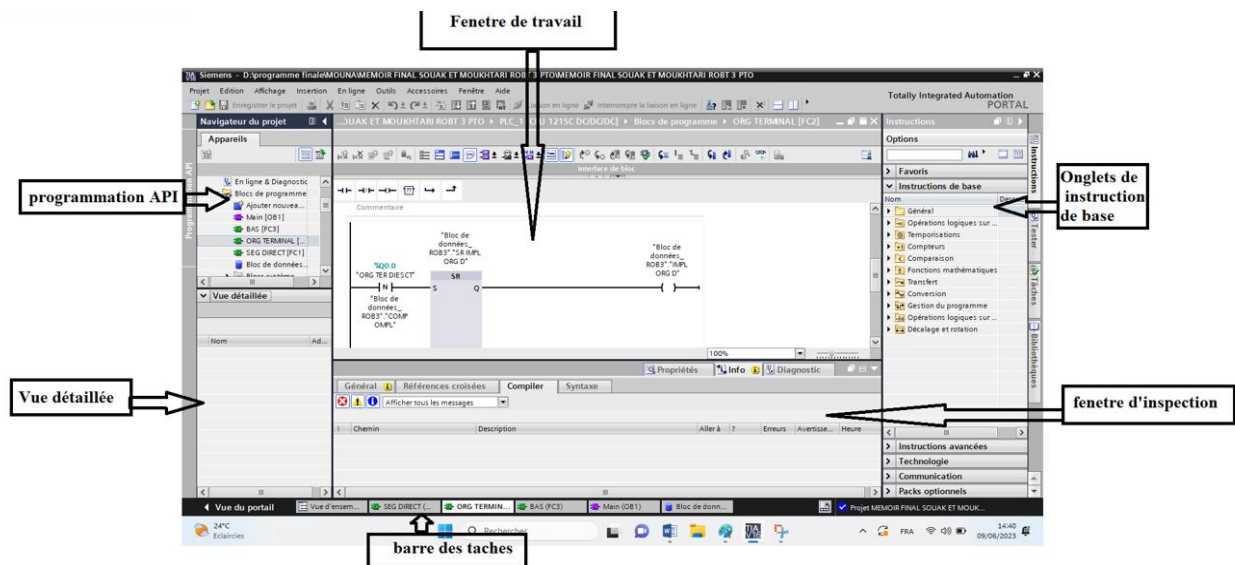


Figure I. 6 : Vue du projet.

I.10.3. Configuration et paramétrage du matériel

Pour configurer la station de travail, on suit impérativement les étapes suivantes:

Définir le matériel existant, on peut passer par la vue du projet et cliquer sur "ajouter un appareil" dans le navigateur du projet.

Lorsque la liste des éléments apparaît que l'on peut ajouter apparaît (*API*, *HMI*, *Système*, *PC*). On commencera par faire le choix de notre *CPU* pour ensuite venir ajouter les modules complémentaires (alimentation, *E/S TOR* ou analogiques, module de communication...). [10]

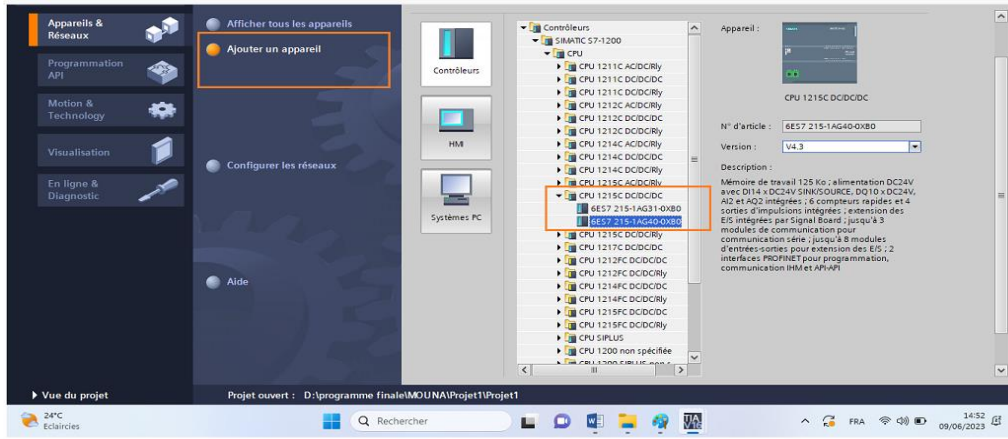


Figure I. 7 Ajout d'un appareil.

I.10.4. Adressage des E/S

Pour connaître l'adressage des entrées et sorties présentes dans la configuration matérielle, il suffit d'aller dans "**appareil et réseau**" dans le navigateur du projet, dans la fenêtre de travail, on doit s'assurer d'être dans l'onglet "**vue des appareils**", de sélectionner l'appareil voulu, la figure ci-dessous est une représentation des adressages des *Entrée/Sortie*. [9]

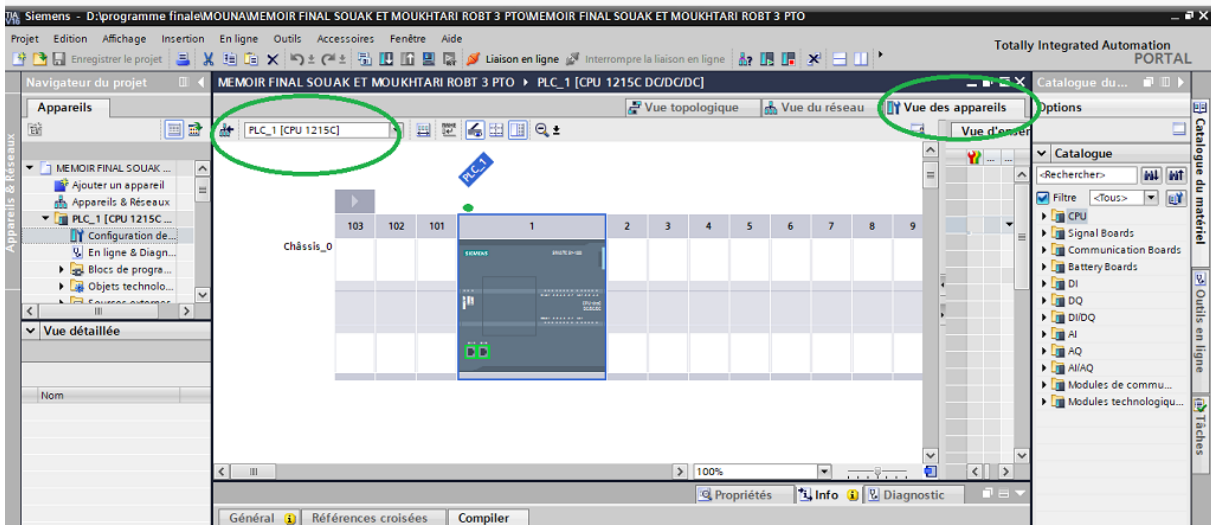


Figure I. 8 : Adressage des E/S.

I.10.5. Adresse Ethernet de la CPU

Pour définir l'adresse Ethernet, on doit procéder comme suit:

"Propriétés >Général> Interface PROFINET> Adresses Ethernet> Adresse IP:192.168.0.1 et Masq.S/res:255.255.255.0".[10]

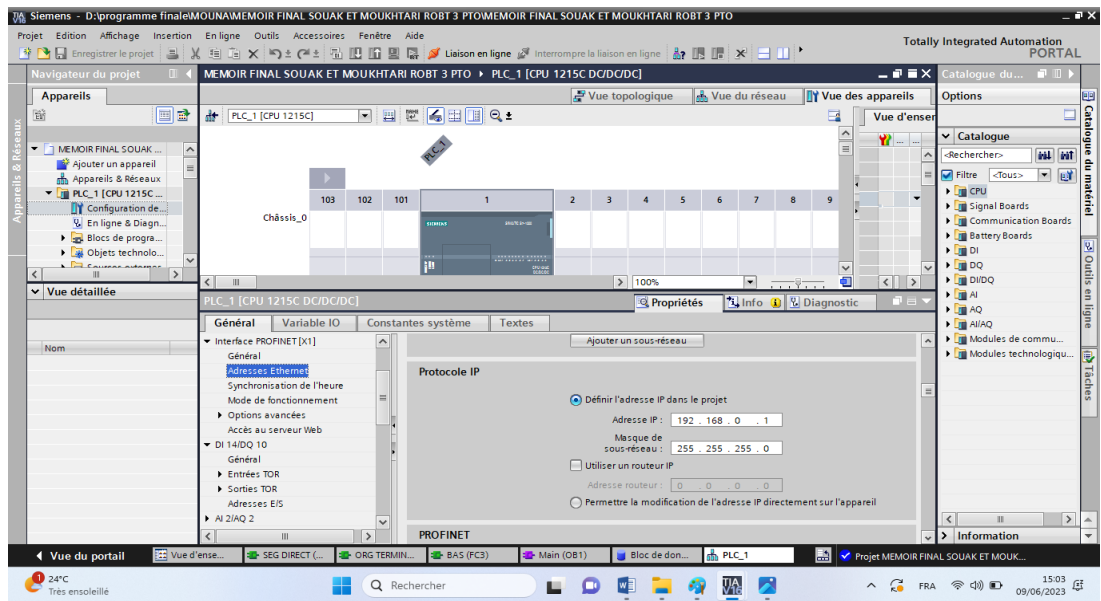


Figure I. 9 : Adresse Ethernet de le CPU.

Pour établir une liaison entre le CPU et la console de programmation, il faut affecter aux deux appareils des adresses appartenant au même réseau.

I.10.6. Implantation du programme

Après la configuration matérielle, on écrit notre programme en choisissant l'un de ces langages: *LADDER* ou *CONT*, *LOG*, *SCL*, *LIST*. On passera à choisir le bloc qui va contenir le programme, il y'a quatre blocs. [10]

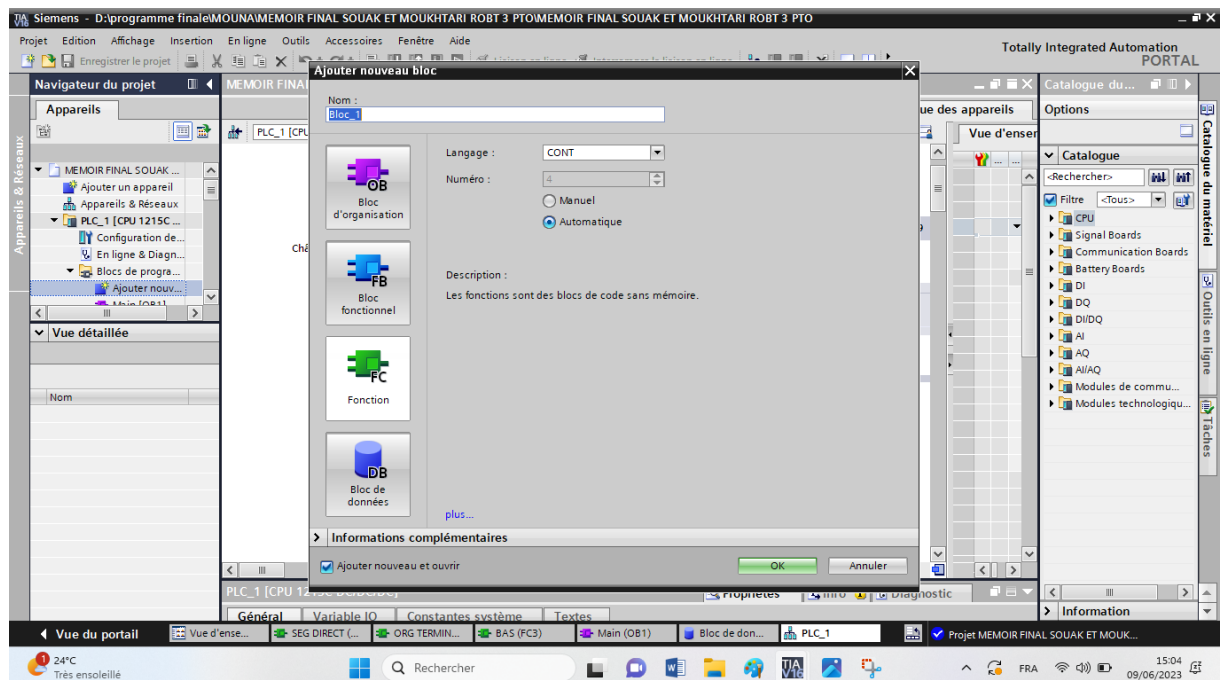


Figure I. 10 : Blocs du programme.

- **OB**: Les blocs d'organisation (*OB*) commandent le traitement du programme. Il est possible par l'intermédiaire des *OB* de réagir aux événements cycliques, temporisés ou déclenchés par alarme durant l'exécution du programme.
 - **FB**: Les blocs fonctionnels sont des blocs de code qui sauvegardent en permanence leurs valeurs dans des blocs de données d'instance afin qu'il soit possible d'y accéder même après le traitement du bloc.
 - **FC**: Les fonctions sont des blocs de code sans mémoire.
 - **DB**: Les blocs de données (*DB*) servent à sauvegarder les données du programme.
- [10]

I.12. Description générale de SIMATIC HMI

La gamme de panneaux *SIMATIC* offre la bonne solution pour chaque application, du simple panneau de clavier aux interfaces opérateur mobiles et stationnaires, en passant par un polyvalent performant - des options d'interface robustes, compactes et multiples. Des écrans d'affichage brillants et un fonctionnement ergonomique sans erreur avec un clavier ou une interface opérateur à écran tactile offrent une valeur ajoutée

Les panneaux *SIMATIC HMI* ont fait leurs preuves dans les applications les plus diverses de tous les secteurs depuis des années. Ils se caractérisent non seulement par une conception innovante et des performances élevées, mais comme point fort unique, ils peuvent être configurés via *SIMATIC Win CC* dans *TIA Portal*. Les utilisateurs peuvent ainsi bénéficier d'une efficacité d'ingénierie sans précédent.

La fonctionnalité du matériel est identique sur toute la gamme d'appareils. On sélectionne simplement la taille d'affichage optimale pour notre application et on choisit entre l'opération via l'écran tactile et/ou les touches. Comme le logiciel est évolutif. On peut commencer avec une petite solution et l'étendre à tout moment si nécessaire, par exemple en augmentant le nombre de balises ou de variables. [11]



Figure I. 11: L'interface de supervision KTP700 Basic

I.12.1. Les différents modèles de la famille SIMATIC IHM

- SIMATIC HMI confort Panels: le premier choix pour les applications exigeantes.
- SIMATIC HMI Basic Panels: fonctions de base pour les applications *IHM* simples.
- Panneaux mobiles SIMATIC HMI: puissance et sécurité entre nos mains.
- SIMATIC HMI Key Panels: pupitres opérateur innovants mise en service rapide détection d'état intuitif mise en réseau facile. [12]

I.12.2. Choix de l'Interface Homme-Machine (*IHM*)

Nous avons choisi le pupitre KTP 700 BASIC de la famille SIEMENS. Ce terminal d'exploitation (Interface Homme Machine) offre une facilité, une souplesse et surtout des fonctionnalités avancées. La mise en œuvre de cette interface de supervision nécessite le développement des programmes en utilisant des outils de programmations tel que TIA PORTAL. [10]

I.12.3. Présentation générale du produit

La nouvelle génération *HMI*, économique, répond à la tendance de la visualisation haut de gamme, y compris sur des machines et installations de petites dimensions. Avec la 2^{ème} génération des Basic Panels *SIMATIC HMI*. Siemens remplit les exigences de l'utilisateur par rapport à une visualisation et une commande haut de gamme, y compris pour les machines et installations de petite et moyenne taille. La connectivité également, via l'interface *PROFINET* ou *PROFIBUS*, au choix, et le port *USB* a pu être largement améliorée. La configuration et la commande des nouveaux panels sont plus simples grâce à la programmation facilitée par la nouvelle version logicielle de Win CC dans le portail *TIA*. [12]

La figure suivante montre la structure des appareils *PROFINET* à l'aide de l'exemple du *KTP700 Basic* :

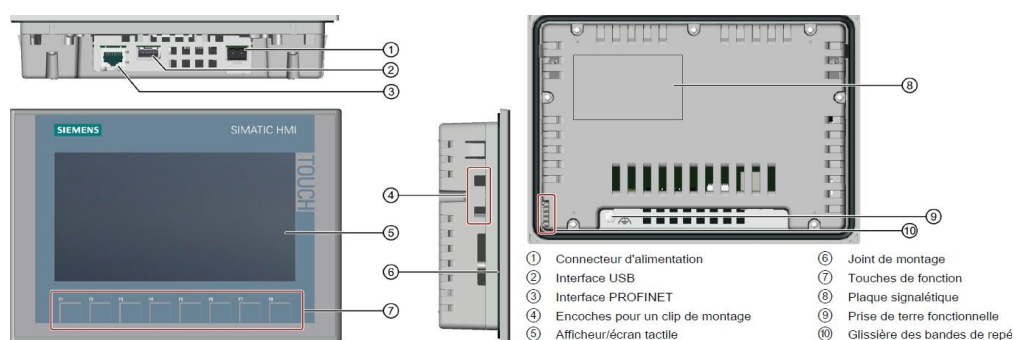


Figure I. 12 : La structure des appareils *PROFINET* KTP700 Basic.

I.13. Conclusion

Dans ce chapitre, nous avons présenté en générale les automates programmables industriels, et on a vu une description de l'automate programmable *S7-1200*, ensuite nous avons présenté le logiciel de programmation *SIMATIC TIA PORTAL* et on a mis l'accent sur l'interface de supervision *HMI*.

Chapitre II

Modélisation Géométriques Direct et Inverse

II.1. Introduction

La conception et la commande des robots nécessitent le calcul de certains modèles mathématiques tel que le modèle géométrique direct et inverse qui expriment la situation de l'organe terminal par rapport à un repère lié à la base.

Dans ce chapitre, on va voir la définition et la modélisation géométrique directe et inverse du robot type *ROB3*.

II.2. Modélisation géométrique

Quel que soit l'objectif qu'on se fixe, concevoir, simuler ou commander un système mécanique, il est nécessaire, entre autres, de disposer de modèles du mécanisme. Plusieurs niveaux de modélisation sont possibles, qui dépendent des spécifications du cahier des charges de l'application envisagée:

Les modèles de transformation entre l'espace opérationnel (dans lequel est définie la situation de l'organe terminal) et l'espace articulaire (dans lequel est définie la configuration du robot). Parmi ces modèles, on distingue plusieurs types de modèles

II.2.1. Les modèles géométriques direct et inverse

Ces types de modèles expriment la situation de l'organe terminal en fonction des variables articulaires du mécanisme et inversement.

II.2.2. Les modèles cinématiques direct et inverse

Les modèles cinématiques direct et inverse expriment la vitesse de l'organe terminal en fonction des vitesses articulaires et inversement.

II.2.3. Les modèles dynamiques

Les modèles dynamiques définissent les équations du mouvement du robot, qui permettent d'établir les relations entre les couples ou forces exercées par les actionneurs et les positions, vitesses et accélérations des articulations. [19]

II.3. Modèle Géométrique Direct

II.3.1. Définition

Le modèle géométrique direct est l'ensemble des relations qui permettent d'exprimer la situation de l'organe terminal, les coordonnées opérationnelles du robot en fonction de ses coordonnées articulaires. Dans le cas d'une chaîne simple ouverte, il peut être représenté par la matrice $T_{0,n}$ qui se calcule par :

$$T_{0n}=T_{0,1}(q_1) \times T_{1,2}(q_2) \times \dots \times T_{n-1,n}(q_n).$$

Le modèle géométrique direct du robot peut se représenter par la relation:

$$X=f(q)$$

Où X est le vecteur des coordonnées opérationnelles exprimées dans le repère de référence R_0 , et q les variables articulaires.

La matrice $T_{0,n}$ représente la position et l'orientation, exprimées dans le repère de référence R_0 , de l'organe terminal du robot.

La méthode proposée utilise les matrices de transformation homogène. [18]

• II.3.2.Principe

- Fixer des repères à chaque corps du robot.
- Calculer les matrices homogènes entre chaque corps.
- Calculer la matrice homogène entre base et organe terminal. [18]

II.3.3. Règle de la main droite

Dans la représentation partielle, la règle de la main droite nous permet de mieux placer les repères R_i de coordonnées XYZ . Comme montre figure II.11 [18]

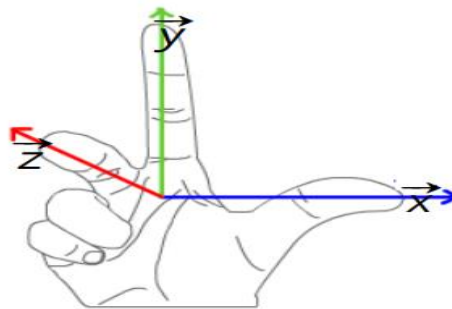


Figure II. 1 : Schéma de la main droite

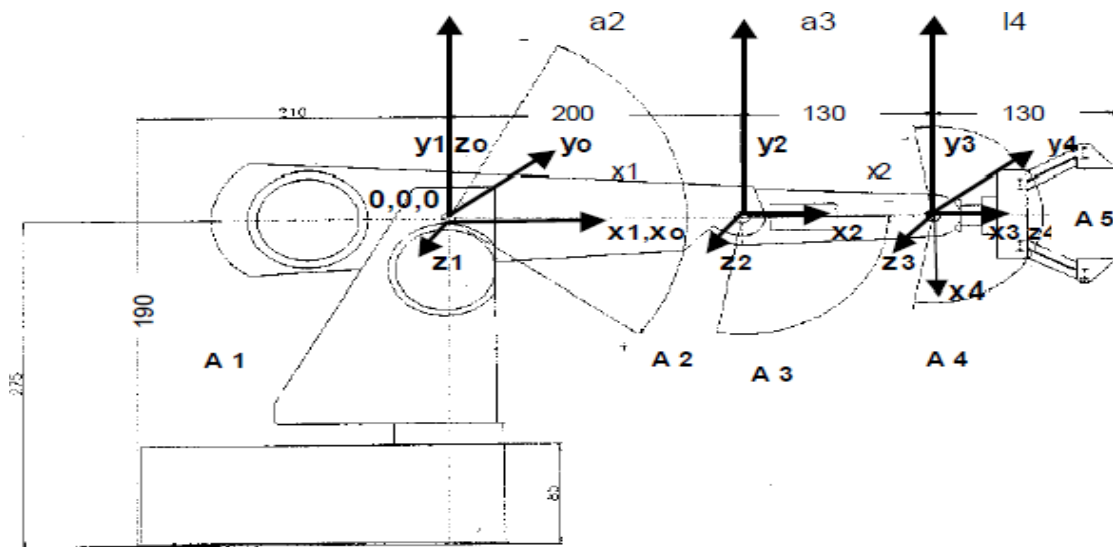


Figure II. 2 : Représentation des repères sur le *ROB3*.

II.4. Convention de Denavit-Hartenberg (DH) :

C'est une méthode destinée à systématiser la modélisation de n'importe quel type de robot série. Ses principaux avantages sont :

- Simplification maximale du modèle géométrique.
- Etablissement d'une norme reconnue par tous. [19]

II.5. Les paramètres de Denavit-Hartenberg

Décomposition en 4 transformations élémentaires:

- Rotation autour de z d'un angle θ_i
- Translation le long de z d'une longueur d_i .
- Translation le long de x d'une longueur a_i .
- Rotation autour de x d'angle α_i . [19]

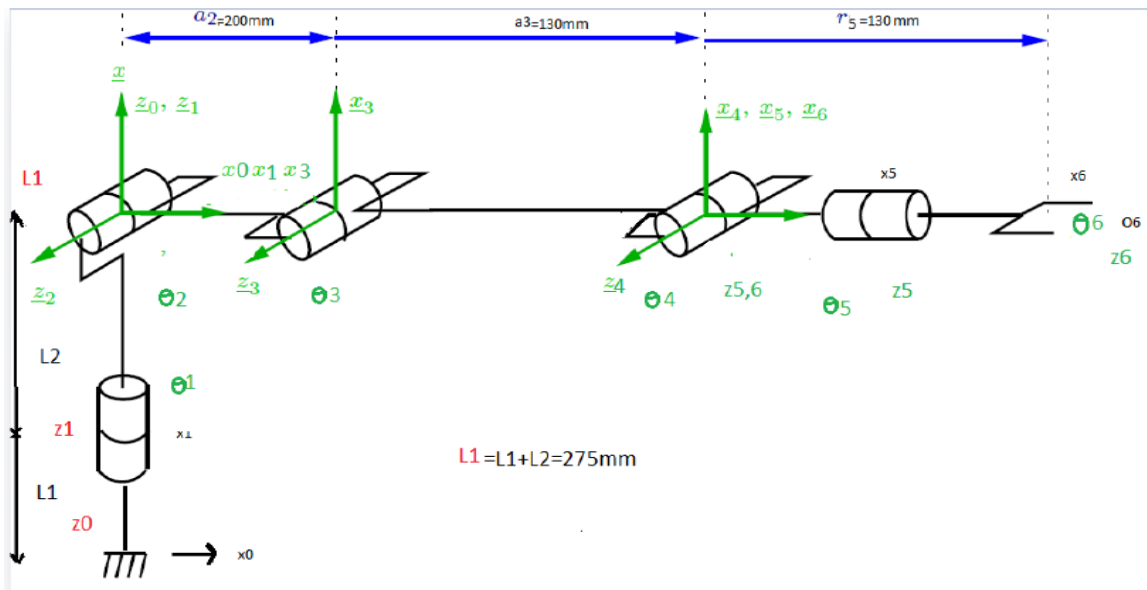


Figure II. 3 : Paramètres géométriques du robot ROB3.

II.6. Obtention du modèle géométrique direct

Il existe plusieurs méthodes pour calculer le modèle géométrique direct à savoir : Méthode de paramètres de Denavit-Hartenberg. Méthode de paramètres de Denavit-Hartenberg modifié. Méthode classique. Parmi ces méthodes, on utilise la méthode de paramètres de Denavit-Hartenberg. [18]

II.7. Tableau Denavit-Hartenberg

Le tableau ci-après résume les différentes articulations en fonction des positions du bras de robot.

Articulation	α_i	a_i	L_i	θ_i
R0 → R1	0	0	L1	θ_1
R1 → R2	$\pi/2$	0	0	θ_2
R2 → R3	0	a 2	0	θ_3
R3 → R4	0	a 3	0	θ_4
R4 → R5	$\pi/2$	0	r 5	θ_5

Tableau II. 1 : Denavit-Hartenberg.

II.8. Calcul du MGD du bras ROB3

$${}^{j-1}T_j = \text{Rot}(x, \alpha_j) \times \text{Trans}(x, d_j) \times \text{Rot}(z, \theta_j) \times \text{Tras}(z, r_j)$$

$$T_i = \begin{pmatrix} \cos\theta_i & -\sin\theta_i * \cos\alpha_i & \sin\theta_i * \sin\alpha_i & \alpha_i \cos\theta_i \\ \sin\theta_i & \cos\theta_i * \sin\alpha_i & -\cos\theta_i * \sin\alpha_i & \alpha_i * \sin\theta_i \\ \mathbf{0} & \sin\alpha_i & \cos\alpha_i & d_i \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} \end{pmatrix}$$

Les matrices de transformation homogène des six segments sont données par :

1. La matrice de transformation dans le repère R_0 vers R_1 :

$$T_{01} = \begin{pmatrix} C1 & -S1 & 0 & 0 \\ S1 & C1 & 0 & 0 \\ \mathbf{0} & 0 & 1 & L1 \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} \end{pmatrix}$$

2. La matrice de transformation dans le repère R_1 vers R_2 :

$$T_{12} = \begin{pmatrix} C2 & S2 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ \mathbf{S2} & C2 & 0 & 0 \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} \end{pmatrix}$$

3. La matrice de transformation dans le repère R_2 vers R_3 :

$$T_{23} = \begin{pmatrix} C3 & -S3 & 0 & a2 \\ S3 & C3 & 0 & 0 \\ \mathbf{0} & 0 & 1 & 0 \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} \end{pmatrix}$$

4. La matrice de transformation dans le repère R_3 vers R_4 :

$$T_{34} = \begin{pmatrix} C4 & -S4 & 0 & a3 \\ S4 & C4 & 0 & 0 \\ \mathbf{0} & 1 & 0 & 0 \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} \end{pmatrix}$$

5. La matrice de transformation dans le repère R_4 vers R_5 :

$$T_{45} = \begin{pmatrix} C5 & -S5 & 0 & r4 \\ 0 & 0 & -1 & 0 \\ \mathbf{S5} & C5 & 0 & 0 \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} \end{pmatrix}$$

Après multiplication des matrices des transformations élémentaires, nous obtenons la matrice finale qui sert à déterminer le modèle géométrique direct:

$$T_{05} = \begin{pmatrix} C1C234C5 + S1S5 & -C1C234S5 + S1S5 & C1S234 & C1(C234r5 + C23a3 + C2a2) \\ -S1C234C5 - C1S5 & -S1C234S5 - C1S5 & S1S234 & S1(C234r5 + C23a3 + C2a2) \\ S234C5 & -S234S5 & -C234 & S234r5 + S23a3 + S2a2 + L1 \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} \end{pmatrix}$$

Ainsi, on obtient la matrice $T_{0,5}$ qui résume la position et l'orientation de l'OT par rapport à la base.

Orientation			vecteur position de l'organe
$C1C234C5 + S15$	$-C1C234S5 + S15$	$C1S234$	$C1(C234r5 + C23a3 + C2a2)$
$-S1C234C5 - C1S5$	$-S1C234S5 - C15$	$S1S234$	$S1(C234r5 + C23a3 + C2a2)$
$S234C5$	$-S234S5$	$-C234$	$S234r5 - S23a3 + S2a2 + L1$
0	0	0	1

$$Px = C1(C234r5 + C23a3 + C2a2)$$

$$Py = S1(C234r5 + C23a3 + C2a2)$$

$$Pz = S234r5 - S23a3 + S2a2 + L1$$

II.9. Modèle géométrique inverse (MGI)

Le modèle géométrique direct d'un robot permet de calculer les coordonnées opérationnelles donnant la situation de l'organe terminal en fonction des coordonnées articulaires, le problème inverse consiste à calculer les coordonnées articulaires correspondant à une situation donnée de l'organe terminal. [20]

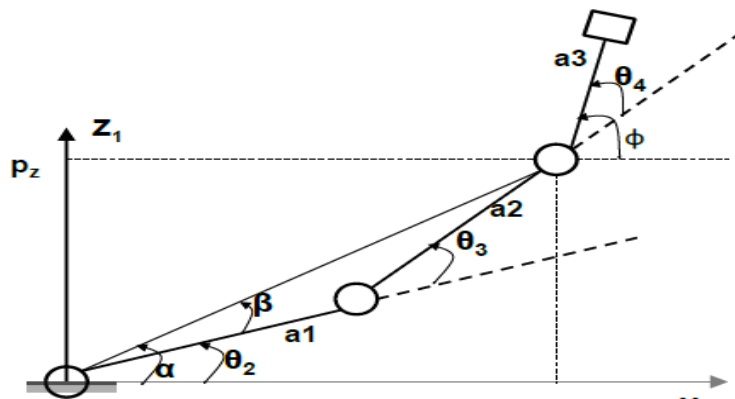


Figure II. 4 : Paramètres géométriques du robot ROB3.

II.9.1. Calcul de la MGI du bras ROB3

Il n'existe pas de méthode analytique systématique pour calculer la MGI. Le mieux est de reprendre les équations du MGD, préalablement calculé et de mener le calcul à l'envers. Le calcul se fait alors au cas par cas. Il est généralement aisé pour un bras manipulateur à moins de six axes, comme les bras manipulateur de type ROB3 le cas où n=5, l'existence d'un poignet sphérique permet de débiter la résolution.

On résout ensuite le système d'équations non-linéaires que constitue le MGD, pour exprimer les q_i , pour $i=1, 2, \dots, n$, en fonction de px, py, pz et des cosinus directeurs [20].

Calcul de θ_1 :

$$\frac{\sin(\theta_1)}{\cos(\theta_1)} = \frac{y}{x}$$

$$\theta_1 = \arctg\left(\frac{y}{x}\right) = \text{atan2}(y, x)$$

Calcul de θ_2

$$a_2c_2 + a_3c_2^3 + r_5c_2^3 - p_x = 0$$

$$a_2s_2 + a_3s_2^3 + r_5s_2^3 - z = 0 \quad (z = pz - 1)$$

Posons : $\alpha = \theta_2 + \theta_3 + \theta_4$, $\beta = \theta_2 + \theta_3$; (α : l'orientation de l'OT)

$$a_2c_2 + a_3c_2^3 + r_5c\alpha - p_x = 0$$

$$a_2s_2 + a_3s_2^3 + r_5s\alpha - z = 0$$

$$a_2c_2 + a_3c_2^3 = u_1; \quad u_1 = p_x - r_5c\alpha$$

$$a_2s_2 + a_3s_2^3 = u_2; \quad u_2 = z - r_5s\alpha$$

Sachant que :

$$\cos(\theta_2 + \theta_3)^2 + \sin(\theta_2 + \theta_3)^2 = 1$$

$$(u_1 - a_2c_2)^2 + (u_2 - a_2s_2)^2 = a_3^2$$

On obtient:

$$u_1c_2 + u_2s_2 = \frac{a_2^2 - a_3^2 + u_1^2 + u_2^2}{2 \cdot a_2}$$

Équation de type 2 de la forme :

$$X \cdot \sin\theta + Y \cdot \cos\theta = Z, \text{ avec}$$

$$X = u_2, \quad Y = u_1, \quad Z = \frac{a_2^2 - a_3^2 + u_1^2 + u_2^2}{2 \cdot a_2}$$

Après calcul, on obtient :

$$\cos\theta_2 = \frac{YZ - \epsilon X \sqrt{x^2 + y^2 - z^2}}{x^2 + y^2}$$

Avec $\epsilon = \pm 1$

$$\sin\theta_2 = \frac{XZ - \epsilon X \sqrt{x^2 + y^2 - z^2}}{x^2 + y^2}$$

Pour $\epsilon = +1$, on a :

$$\theta_{11} = \arctg\left(\frac{\sin\theta_2}{\cos\theta_2}\right)$$

Pour $\epsilon = -1$,

on a :

$$\theta_{12} = \arctg\left(\frac{\sin\theta_2}{\cos\theta_2}\right)$$

$$\theta_2 = \max(\theta_{11}, \theta_{12})$$

Calcul de θ_3

$$a_2c_2 + a_3c_2^3 = u_1; \quad \beta = \theta_2 + \theta_3$$

$$a_2^2 s_2^2 + a_3^2 s_3^2 = u_2^2$$

$$\cos \beta = (u_1 - a_2 \cos \theta_2) / a_3$$

$$\sin \beta = (u_2 - a_2 \sin \theta_2) / a_3$$

$$\beta = \arctg(\sin \beta / \cos \beta)$$

$$\theta_3 = \beta - \theta_2$$

Calcul de θ_4

$$\theta_4 = \alpha - \beta = \alpha - \theta_2 - \theta_3$$

II.10. Conclusion

Dans ce chapitre, nous avons fait la modélisation géométrique directe et inverse de notre bras de robot (*ROB3*) qui consiste à déterminer la position de l'organe terminal en fonction de ses coordonnées articulaires et vice versa.

Chapitre III:
Description et conception de *Rob3*

III.1. Introduction

Le **ROB3** est un robot de haute qualité de formation, particulièrement conçu pour l'usage dans toutes les sphères d'éducation et de formation de robotique. Le bras articulé de **ROB3** a cinq axes et une pince, qui sont actionnées par les servomoteurs à courant continu.



Figure III. 1 Bras de robot manipulateur **ROB3**.

III.2. Espace du travail du robot manipulateur **ROB3**

La figure si dessous représente la zone active du **ROB3** c'est-à-dire l'espace de travail dont tous les points sont accessibles par la pince du robot. Tout point hors de cet espace est considéré hors de portée du robot. [13]

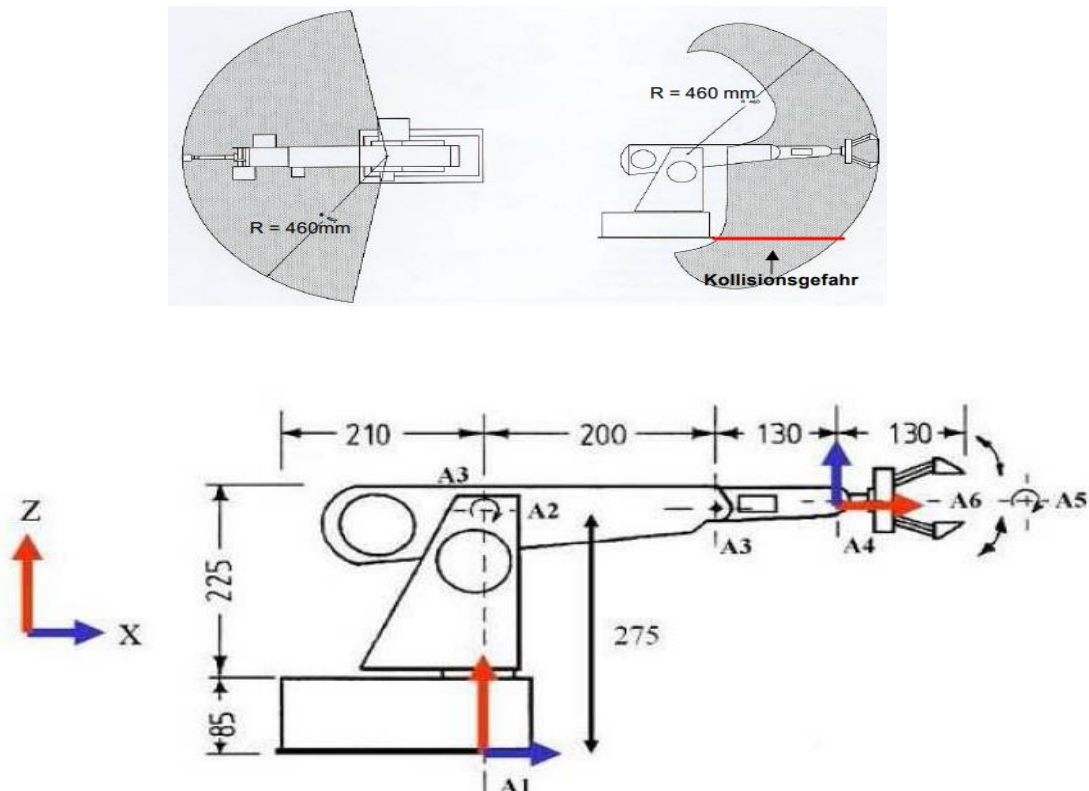
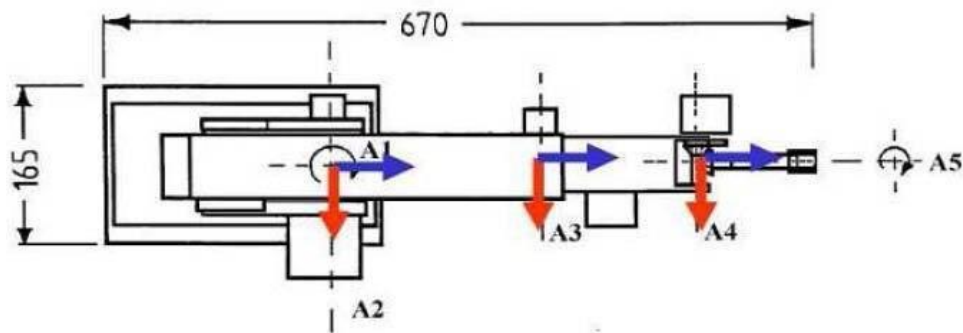


Figure III. 2 : Espace du travail du robot manipulateur **ROB3**.

Figure III. 3 : Les dimensions du robot **ROB3** (vue de face).**Figure III. 4** : Les dimensions du robot **ROB3** (vue de haut).

III .3. Les actionneurs

Le bras est muni de six petits moteurs à courant continu, trois moteurs de marque "*Bühler*" (*N° 1.61.042.303-1/9V DC*), et les trois autres sont des petits servomoteurs "*Multiplex*" (*STELL-SERVO Best. Nr.6 5089*), cinq d'entre eux sont destinés chacun à faire mouvoir une articulation. En revanche, le dernier est adapté pour ouvrir et fermer la pince.

Les moteurs utilisés sont des moteurs à aimants permanents. Ils fonctionnent avec une tension de 9V générée par une alimentation continue stabilisée et un courant max de 3A.

III .3.1. Les moteurs à courant continu

Le moteur électrique à courant continu est le muscle du bras de robot, qui est un dispositif électromécanique servant à convertir de l'énergie électrique d'entrée en énergie mécanique.

En effet, il existe plusieurs types de moteurs électriques, on a étudié un seul cas, il s'agit d'un moteur à courant continu à aimants permanents. Ces moteurs qui présentent une très faible inertie mécanique, sont très bien adaptés aux applications nécessitant des variations rapides de vitesse de rotation ou commande en position (périphériques d'ordinateurs, robotique, etc...). Ils sont toujours commandés par un ensemble électronique comportant une alimentation de puissance avec une électronique de commande réalisant un asservissement.

Classiquement, un moteur à courant continu est constitué d'un stator fixe et d'un rotor mobile. Le courant continu arrive par les cosses à l'extérieur de la carcasse sur des balais souples. Le collecteur est la partie du rotor qui touche les balais. Le rotor est constitué, quant à lui, de plusieurs bobines, toujours en nombres impairs, reliées au collecteur. Lorsque l'une des bobines reçoit le courant continu, elle crée un champ magnétique, mais

la carcasse du stator abrite aussi deux aimants de pôles opposés. Le champ magnétique créé attire les pôles des aimants, créant par cela une rotation de l'axe du moteur.

. [14]



Figure III. 5 : Moteur à courant continu "Bühler".

III .3.1.1. Transmissions et systèmes réducteurs de vitesse

La transmission du mouvement dans notre bras de robot est assurée à l'aide des courroies et des engrenages qui assurent une liaison entre les actionneurs et les articulations sur tout au niveau du poignet. Suite aux vitesses élevées des moteurs à courant continu, le système réducteur de vitesse sert tout simplement à réduire la vitesse de rotation de l'axe de sortie. [15]

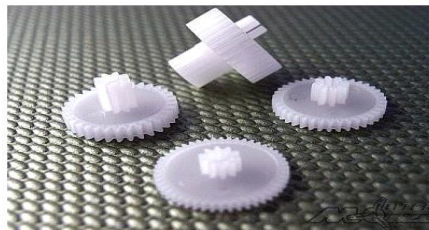


Figure III. 6: Pignons.

III.3.2. Les servomoteurs

Les servomoteurs permettent de déplacer un bras, sur lequel est fixé un objet, jusqu'à une certaine position (ou angle de rotation), puis à maintenir solidement cette position. Le terme même de servomoteur signifie qu'il s'agit d'un moteur asservi obéissant à une commande externe.



Figure III. 7 : Servomoteurs "Multiplex".

III .4. La carte de puissance

Cette partie du montage est alimentée avec une tension de $9V/3A$ fournie par une alimentation. Ce montage se concentre sur l'élément central. Le *L298N*, qui est le cœur du montage. En effet, le circuit amplificateur de puissance *L298N* de *SGS Thomson* est une référence en matière d'interface pour les moteurs *DC* et les moteurs pas à pas. Ces caractéristiques à basses tensions, lui confèrent une place incontestée dans les circuits de puissance intégrée. Il est le plus Utilisés pour les petits robots alimentés par quelques accus.

Les signaux de commande entrants du circuit considéré sont :[18]

- Un signal *PTO*
- Trois paires de signaux de sens correspondant au sens du moteur sélectionné.
- Une masse commune entre les deux parties puissance et commande.

Les signaux de mesures sortant sont :

- Un signal analogique fourni par le capteur de présence.

III .4.1. Schéma électronique de la carte de puissance

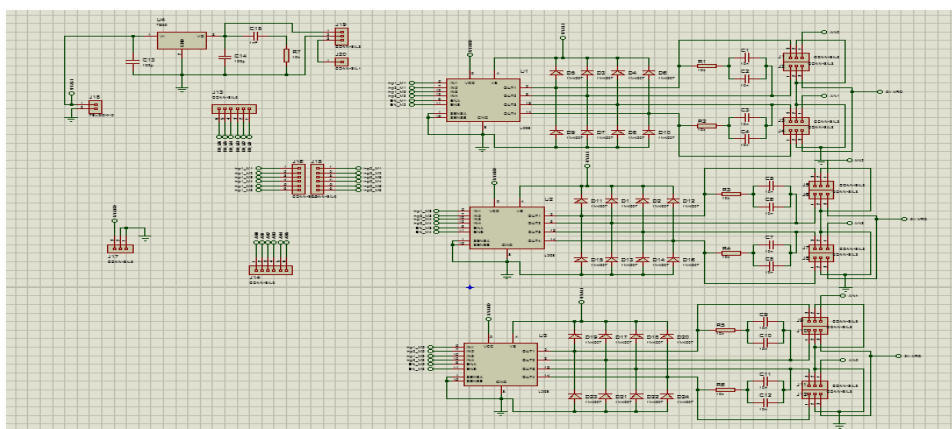


Figure III. 8 : Schéma de la carte de puissance (Proteus).

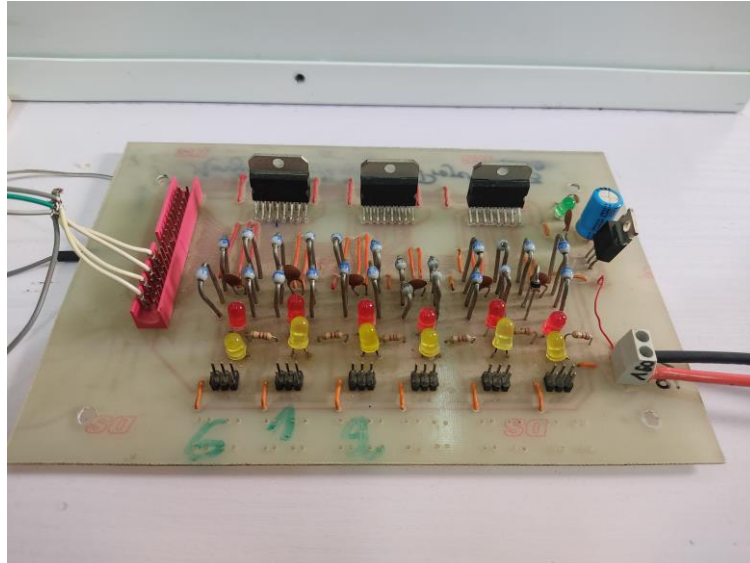


Figure III. 9 : La carte de puissance L298N.

III .5. Eléments constituant la carte de puissance

III .5.1. Le circuit *L298N*

Pour comprendre le fonctionnement du L298N, il faut tout d'abord bien comprendre qu'il y a 2 aspects à considérer :

- d'une part le fait qu'on peut faire tourner les moteurs dans un sens, ou dans l'autre (ce qui est permis par les "*Ponts en H*"),
- d'autre part, le fait qu'on peut contrôler la vitesse de rotation d'un moteur, grâce à un signal *PTO* envoyée sur sa broche de commande.

Ce sont ces 2 aspects qui font que le *L298N* peut à la fois contrôler le sens de rotation des moteurs, et, faire varier leur vitesse. Nous allons donc les voir en détail, dès à présent.



Figure III. 9 : Vue en 3D.

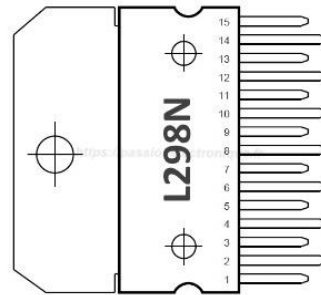


Figure III. 10 : Vue de face.

V_{ss} : Alimentation de la logique de commande (5v).

V_s : Alimentation de puissance des moteurs.

GND : doit être raccordée à la masse.

OUTPUT 1, OUTPUT 2 : broche à raccorder à la charge (le moteur dans notre cas).

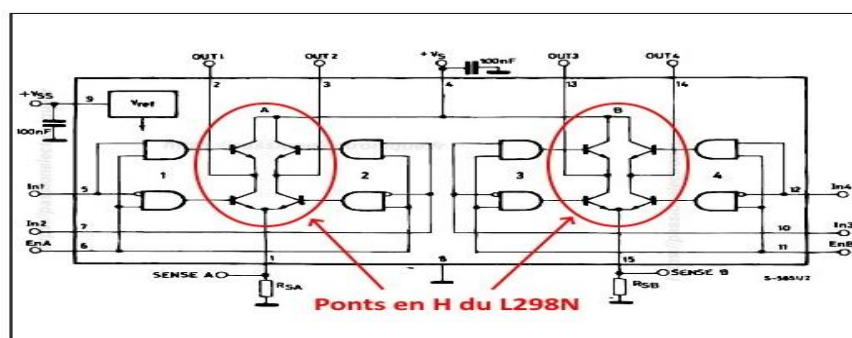
INPUT 1, INPUT 2: Broche de commande du Pont-H.

ENABLE A : commande l'activation du premier Pont-H, permet d'envoyer la tension sur les sorties du moteur via **OUTPUT 1**.

ENABLE B commande l'activation du premier Pont-H, permet d'envoyer la tension sur les sorties du moteur via (**OUTPUT 2**).

Si **ENABL A=GND**, le pont-H est déconnecté et le moteur ne fonctionne pas.

Si **ENABL A= V_{ss}** , le pont-H est connecté aux sorties et le moteur fonctionne dans **CURRENT SENSING A** et **CURRENT SENSING B**-permet de faire une mesure du courant dans le circuit de puissance, à placer impérativement sur **GND** si cette fonctionnalité n'est pas utile.

Figure III. 11 : Schéma interne du **L298N**.

III .5.1.1. Définition des ponts en H

Les Ponts en H ne sont en fait qu'un nom donné à une configuration particulière, de composants électroniques. Classiquement, ceux-ci sont composés de 4 transistors, disposés d'une certaine manière, formant visuellement la lettre "H" (ou presque !). En voici un exemple, avec quatre transistors bipolaires, comme ceux qu'on trouve en interne, dans le

L298N :[21]

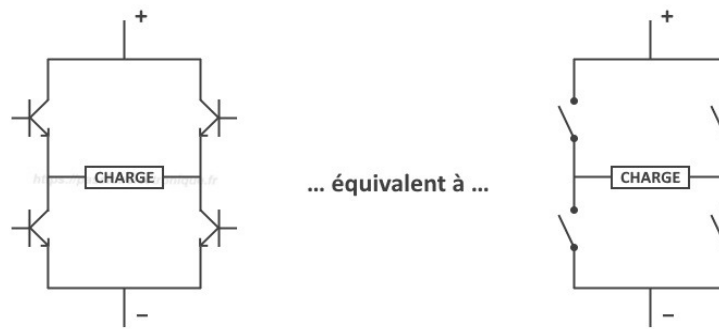


Figure III. 12 : Schéma d'un pont en H.

III .5.1.2. Le principe d'un pont en H

En activant les transistors deux par deux (ceux de sens opposés), on peut contrôler le sens du passage du courant dans la charge, branchée "au milieu du H". C'est ce changement de sens de courant, qui permet aux moteurs à courant continu de pouvoir tourner "à l'endroit", ou "à l'envers". Visuellement, voici comment cela se passe: [21]

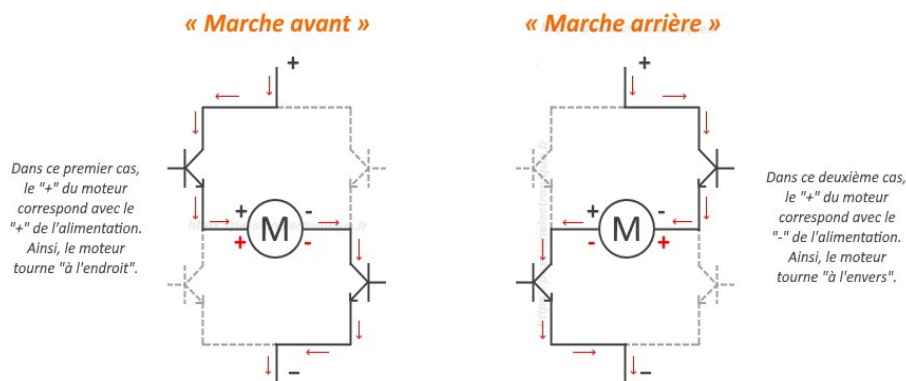


Figure III. 13 : Pont en H les deux configurations possibles.

Comme vous le voyez, cette configuration à 4 transistors permet :

- -Dans le 1^{er} cas : le "+" du moteur sur le "+" de l'alimentation et le "-" du moteur sur le "-" de l'alimentation
- Dans le 2^{ème} cas : le "+" du moteur sur le "-" de l'alimentation et le "-" du moteur sur le "+" de l'alimentation (faisant ainsi tourner le moteur en sens inverse).

En bref, en commandant simplement 2 des 4 transistors de manière symétrique, on peut facilement faire passer le courant dans un sens, ou bien dans l'autre, et c'est d'ailleurs exactement le principe utilisé dans le L298N, pour faire changer le sens de rotation des moteurs électriques. [21]

III .5.1.3. Table de vérité du *L298N* (fonctionnement)

Le *L298N* possède plusieurs entrées logiques, qui sont :

- *ENA* et *ENB* : pour activer respectivement ou désactiver les ponts A et B (les deux ponts en H de notre *L298N*, constitués de 4 transistors bipolaires chacun),
- *IN1* et *IN2* : pour définir le sens du courant en "sortie" du pont A (pour le 1^{er} moteur),
- *IN3* et *IN4* : pour définir le sens du courant en "sortie" du pont B (pour le 2^{ème} moteur).

Ces entrées s'alimentent avec:

- Une tension basse (0 volts, par exemple), pour signifier un état bas (un "0" logique).

Une tension haute (+5 volts, par exemple), pour signifier un état haut (un "1" logique), et suivant comment on alimente chacune de ces 6 broches (à l'état bas, ou à l'état haut), le comportement des moteurs branchés sur le *L298N* sera modifié en conséquence. Voici d'ailleurs tous les cas possibles, pour le pont A (*ENA/IN1/IN2*) et le pont B (*ENB/IN3/IN4*) :

[21]

Entrées		Sorties	Fonction
ENA = 1	IN1= 1 IN2=0	Out1 = 1/ Out2 = 0	Marche arrière
	IN1= 0 IN2= 1	Out1 = 0/ Out2 = 1	Marche avant
	IN1 =IN2	Out1=1; Out2= 1/ Out1=1 ; Out2=1	Blocage du moteur le (arrêt rapide, freinage fort)
ENA = 0	IN1= X IN2= X	Rien	Moteur en roue libre (à l'arrêt, sans frein)
ENB = 1	IN3 = 1 IN4 = 0	Out3 = 1 / Out4 = 0	Marche arrière
	IN3= 0 IN4= 1	Out3 = 0 / Out4 = 1	Marche avant
	IN3= IN4	Out3 = 1 ; Out4 = 1 Out3 = 1 ; Out4 = 1	Blocage du moteur le (arrêt rapide, freinage fort)
ENB = 0	IN3 = X IN4 = X	Rien	Moteur en roue libre (à l'arrêt, sans frein)

Tableau III. 1 : Table de vérité du L298N (fonctionnement).

III .5.1.4.Caractéristiques du *L298N*

Les caractéristiques du L298N sont les suivantes

- Intensité maximale: 2A par pont.
- Alimentation de puissance de 5.5V à50V.
- Type de boîtier : Multi watt 15.
- Dissipation puissance total: 25w.
- Trois entrées par pont: In1, In2 et ENABLE [18]

III .5.1.5.Les Avantages du *L298N*

- Supporte une tension de puissance jusqu'à 36 volts maximum.
- Un courant de service de 2A.
- Dispose d'un dispositif de mesure du courant (sens or/sens pins).
- A raccorder à la masse si on ne l'utilise pas.

III .5.2.Les diodes 1N4007

Les diodes (*1N4007*) sont des diodes de puissance présentent un temps de commutation

faibles et capables de laisser passer un courant important. Elles servent de protection "roue libre", afin de compenser le courant inverse qui se fera pour chaque déclenchement **ON/OFF** du moteur. Donc nous avons câblé huit (08) diodes en antiparallèle sur chaque L298N pour la protection de ce circuit parce que contrairement à la majorité des circuits, le **L298N** n'intègre pas les diodes de protections des transistors internes.

Ces diodes vont récupérer le courant et ainsi assurer la protection de notre étage de puissance. [22]



Figure III. 14 : Diode 1N4007.

III .5.3.Le circuit d'alimentation

Cette partie est très simple. Nous avons besoin de produire une tension stabilisée. Il existe pour cela des composants dédiés, les 78xx. Ces composants sont des composants à trois pattes. Leur fonctionnement est très simple. Ils ont entre les deux premières pattes une tension d'alimentation quelconque (supérieure à la tension de sortie désirée). Ils fournissent alors en sortie une tension stabilisée égale à xx (05, 09,12,...). On utilise donc ces composants en parallèle avec des condensateurs de découplage comme conseillé dans le *Datasheet*. Pour notre montage on a besoin de tension stabilisée de 5v donc on utilise le régulateur **LM7805**. Pour la protection contre le court-circuit ou les sur intensités de courant, un fusible est disposé directement après l'alimentation, suivi d'un groupe résistance-led de contrôle qui confirme la mise sous tension du circuit et le sens de rotation du chaque moteur.[23]

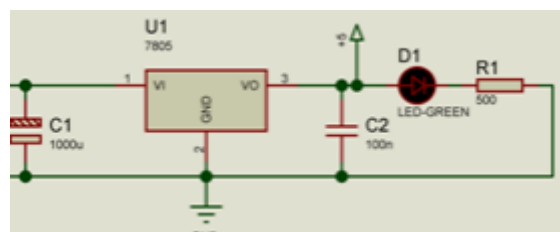


Figure III. 15 : Circuit d'alimentation

III.5.4.Régulateur de tension LM 7805

Il s'agit d'un régulateur de tension de la série **LM78xx**. Le **LM7805** est un régulateur positif à trois bornes avec une tension de sortie fixe de 5V. Ce régulateur fixe fournit une régulation locale, une limitation de courant interne, un contrôle d'arrêt thermique et une

protection de zone sûre pour votre projet. Chacun de ces régulateurs de tension peut produire un courant maximum de 1.5A.

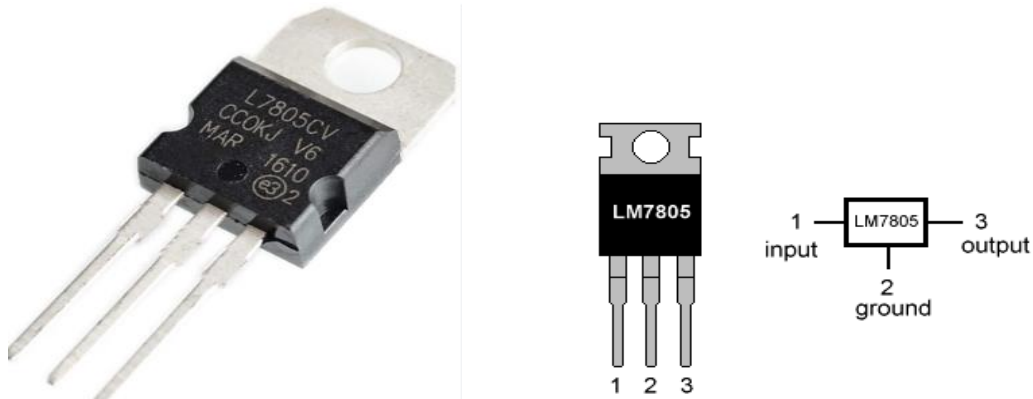


Figure III. 16 : Régulateur de tension LM 7805.

III.5.4.1. Caractéristiques du régulateur 7805

- Courant de sortie maximal 1.5 A
- Tensions de sortie de 5V
- tension d'entrée minimale est de 7 V
- Tensions maximale à l'entrée: 30v
- Protection contre les surcharges thermiques
- Protection contre court-circuite courant de fonctionnement (I Q) est de 5mA
- Une protection interne contre les surcharges thermiques et le court-circuit est disponible
- Température de jonction maximum 125 degrés Celsius
- Disponible en boîtier TO-220 et KTE.[24]

III.5.4.2. Application du régulateur 7805

- Régulateur de tension,
- Projets de bricolage nécessitant une tension spécifique à partir d'une alimentation existante plus élevée,
- Protection pour les appareils ayant une sensibilité spécifique à la tension. [25]

III.6. Le signal de commande Pulse Train Output (PTO)

La fonction *PTO* (Pulse Train Output) met en œuvre une technologie numérique permettant d'assurer un positionnement précis pour la commande en boucle ouverte des variateurs de moteur.

La fonction *PTO* peut être configurée sur la voie 0.

III.6.1. Concept

La fonction *PTO* produit une sortie d'onde carrée pour un nombre d'impulsions et une fréquence spécifiée. Elle permet de contrôler le positionnement et la vitesse de l'axe d'un équipement rotatif.

III.6.2. Performances

La fréquence maximale générée est de 50 kHz pour *PTO*. La fonction *PTO* intégrée peut être utilisée pour un déplacement d'axe simple d'un point à un autre, mais pas pour des applications nécessitant les mouvements suivants:

- Mouvement point à point simultané sur deux axes (chaque axe étant géré de manière indépendante),
- Mouvement point à point synchronisé sur deux axes,
- Mouvement d'interpolation sur deux axes. [26].

III.6.3. Comment fonctionne l'interface d'impulsion?

Selon les réglages du servomoteur, chaque impulsion affecte le mouvement du servomoteur d'un angle spécifique. Si le servomoteur est réglé sur 1000 impulsions par tour, par exemple, il se déplace de 0...36° par impulsion, la vitesse du servomoteur est déterminée par le nombre d'impulsions par unité de temps.[27]

III.6.4. Relation entre le type de signal et le sens de déplacement

Le CPU fournit la vitesse et le sens de déplacement via deux sorties, les relations entre la configuration et le sens de déplacement diffèrent selon les Type de signal sélectionné. Vous pouvez configurer les types de signaux suivants dans la configuration de l'axe sous "Basic paramètres > General": "PTO (impulsion A et sens B)".

PTO (impulsion A et sens B):

Spécifie le sens de rotation de l'entraînement:

- 5 V/24 V à la sortie de sens ⇒ sens de rotation positif
- 0 V à la sortie de sens ⇒ sens de rotation négatif

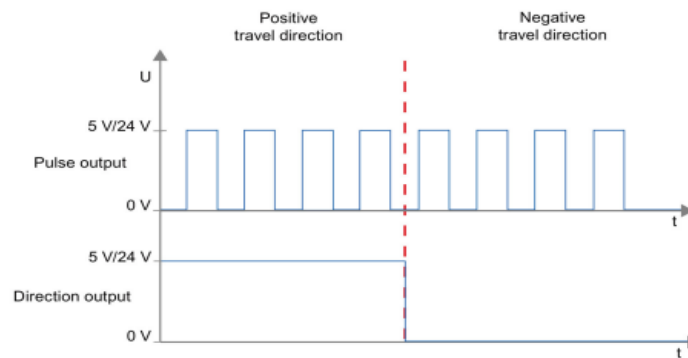


Figure III. 17 : Schéma représentant les deux sorties d'un signal PTO. [27]

III.6.5. Inverser le sens

Si l'on sélectionne l'option "Inverser le signal de rotation", la logique de sens est alors inversée:

PTO (impulsion A et sens B):

- 0 V à la sortie de sens (niveau bas) \Rightarrow sens de rotation positif.
- 5 V/24 V à la sortie de sens (niveau haut) \Rightarrow sens de rotation négatif. [27]

III.7. Conclusion

Dans ce chapitre nous avons présenté les différents éléments constitutifs de notre bras manipulateur *ROB3* et la carte de puissance de façon détaillée, et nous avons expliqué le signal de commande Pulse Train Output (*PTO*) qui commande le bras manipulateur.

Chapitre IV
Partie Expérimentale

V.1. Introduction

Dans le présent chapitre, nous présentons la raison et le but de notre projet, la commande d'un bras manipulateur par un automate programmable industriel. Cette réalisation consiste à créer une carte d'interface entre la carte driver *L298N* des moteurs et *API*, pour lancer le déplacement des bras manipulateur de *ROB3*. Nous allons créer un programme sur logiciel *TIA PORTAL V16* par langage Ladder, pour faire marcher l'ensemble des équipements reliés dans ce système.

V.2 Cahier de charge

Notre cahier de charge a pour objectif de concevoir un système pour déplacer une pièce placée près d'un capteur par un automate programmable S7-1200.

V.3 Les étapes de trie

Les étapes de trie sont:

1. Le capteur détecte la présence de la pièce et ordonne le bras robot pour se déplacer et le mettre dans la position désirée.
2. L'automate déplace le bras robot à sa position initiale pour exécuter une autre tâche.

V.4 Description du système

Comme la montre la **Figure IV.1** ci-dessous le système est complètement automatisé pour déplacer une pièce par un bras manipulateur.

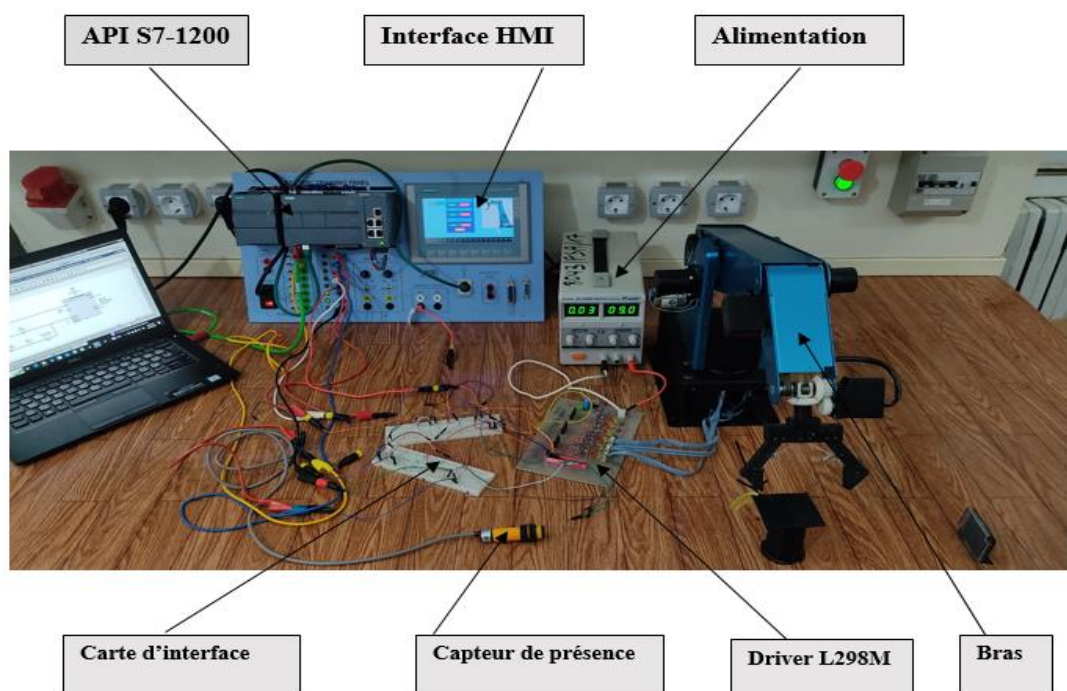


Figure IV. 2 : Description du système.

Le système il est constitué de :

- Une alimentation de 9V DC.
- Une carte de puissance contient de 3 ponts en H (L298N).
- Un capteur de présence OMRON E3F-R2C4.
- Un bras robot qui possède 3 degrés de libertés de rotation.
- Un API s7 1200 1215/DC/DC/DC.
- Une interface HMI de supervision KTP700 Basic.
- Une interface entre les servos moteurs et l'automate.
- PC (logiciel TIA PORTAL V 16).
- Câble PROFINET (6XV1870-3RH60).

La réalisation du projet est devisée en trois phases:

V.4.1 Phase de construction

- La réalisation de la carte adaptation

Nous allons commencer par la construction de la carte automate –servo

Carte interface de sortie (Automate-servo moteur) :

La tension de sortie de la PTO de l'automate est 24 V et la tension de commande PTO des servo est de 5 V d'où la nécessité d'utilisé d'un diviseur de tension pour chaque servo

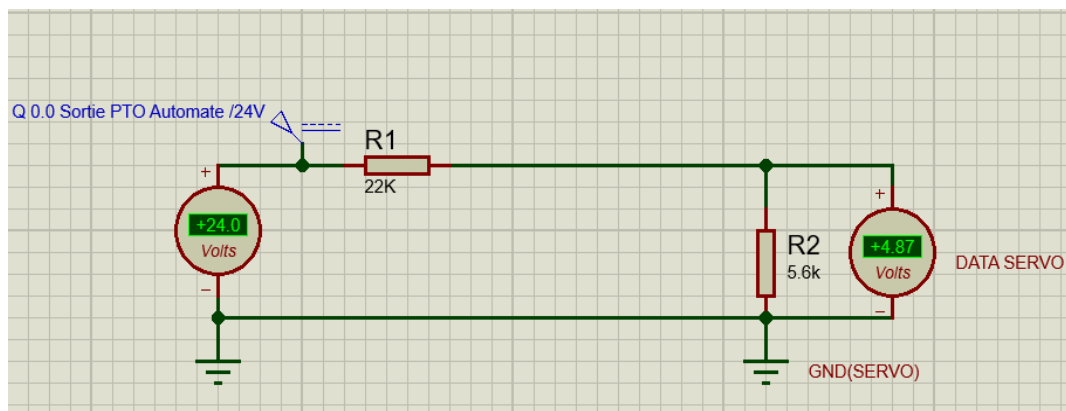


Figure IV. 3 : La Simulation de diviseur par Proteus ISIS.

Dans ce type de montage, les résistances R_1 et R_2 sont en parallèle et elles sont soumises à la tension d'alimentation du circuit. On souhaite mesurer la tension aux bornes d'une des résistances (ici R_2)

On obtient l'équation suivante :

$$V_s = V_{\text{int}} \times \frac{R_2}{R_1 + R_2}$$

$$V_s = 24 \times \frac{5.6}{22+5.6} = 4.876 \text{ V}$$



Figure IV. 4 : La réalisation d'un diviseur de tension.

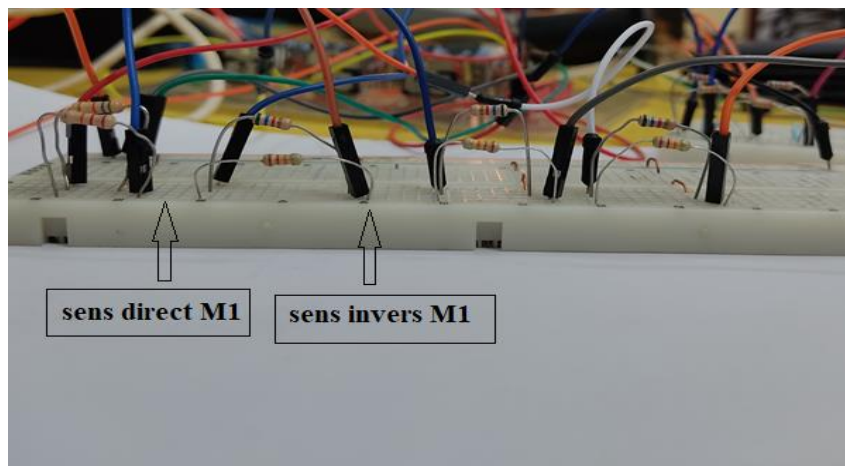


Figure IV. 5 : La réalisation de la carte adaptation.

V.4.2 Phase de programmation

- Réalisation du programme de l'automate S71200,
- Réalisation du programme de la supervision.

Le programme de l'automate est écrit sous TIA portal V16 de Siemens en langage Ladder (contact), il est devisé en blocs essentiels

1. bloc MC_POWER
2. bloc MC_MoveRelative
3. bloc TP (Timer)
4. bloc CTU

Configuration matérielle :

Nous allons d'abord choisir notre PLC

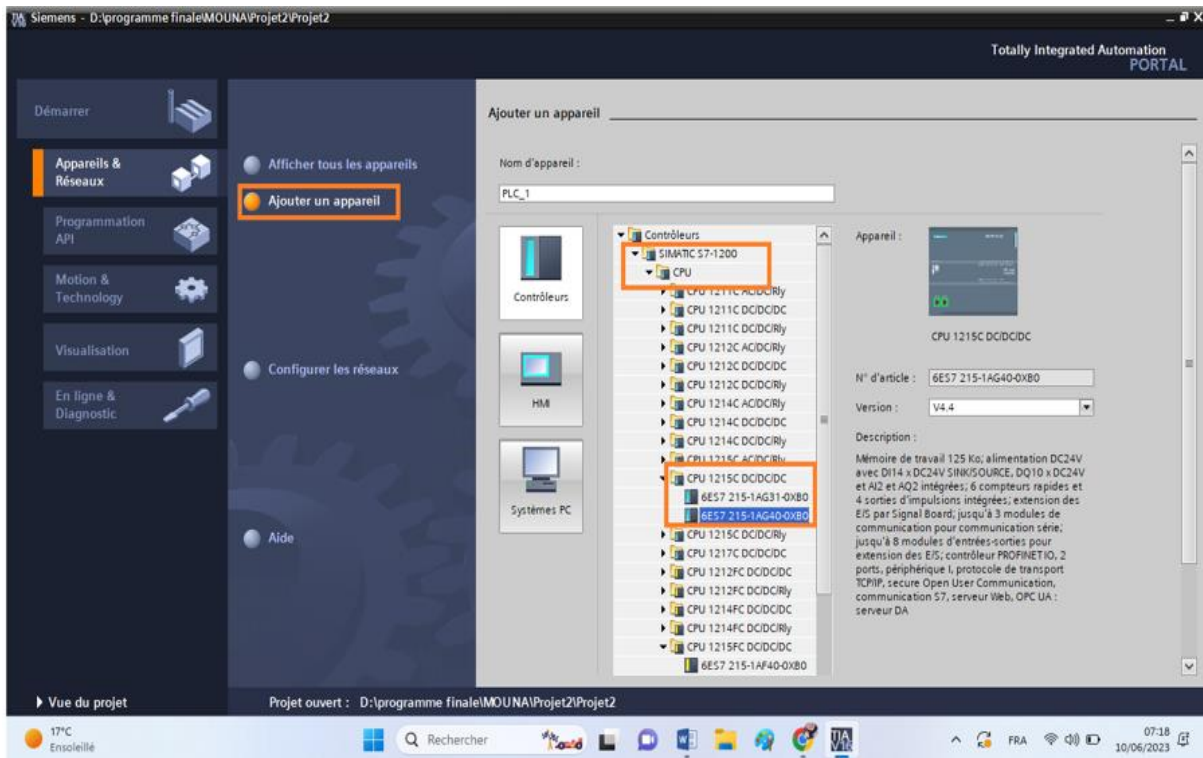


Figure IV. 6 : Le choix de PLC.

La configuration d'un générateur d'impulsions :

Maintenant, nous allons faire une configuration de générateur d'impulsion.

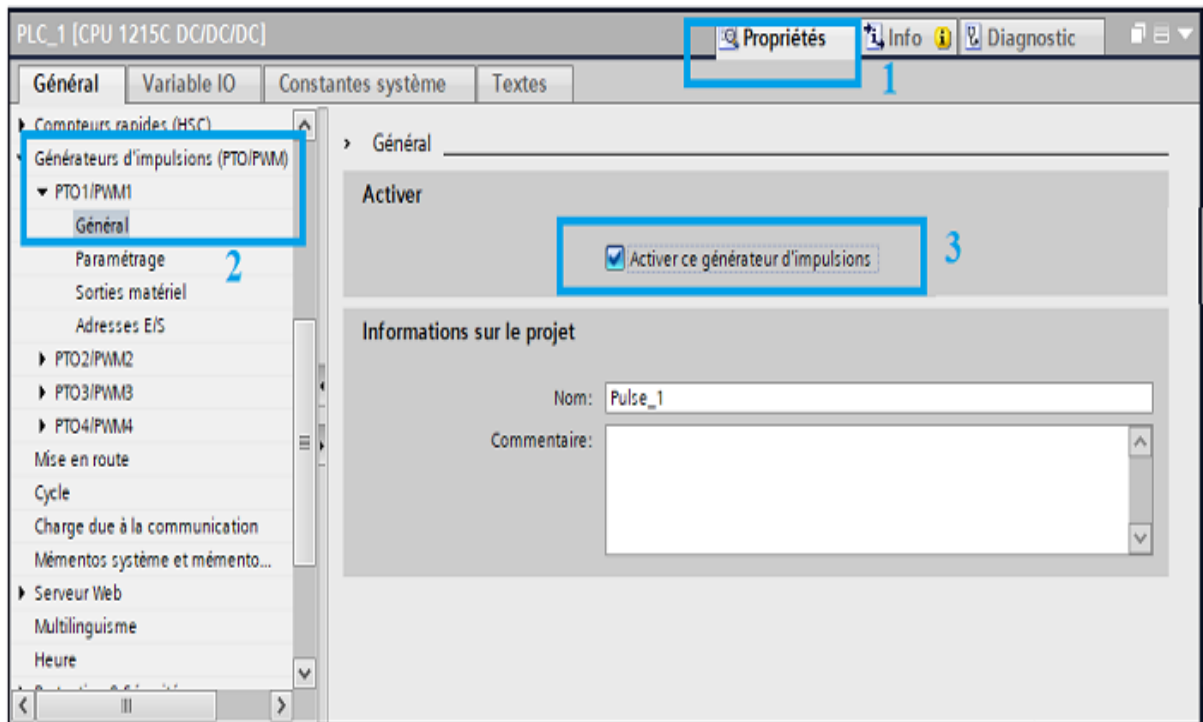


Figure IV. 7 : La configuration de générateur d'impulsions.

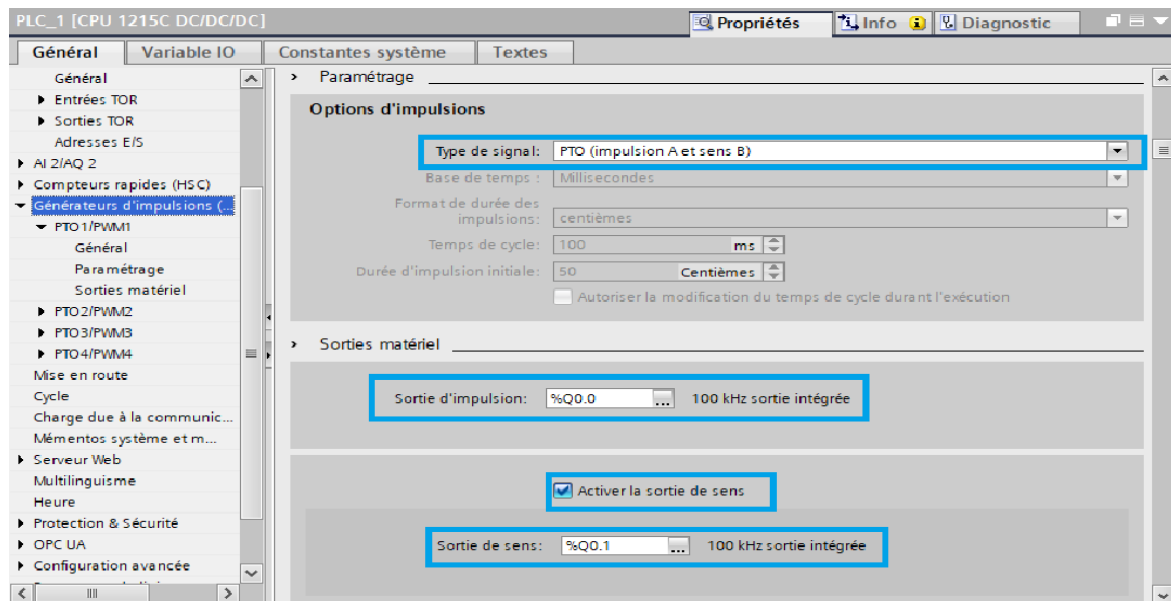


Figure IV. 8 : La configuration de générateur d'impulsions.

L'adresse Q_{0.0} est la sortie d'impulsions *PTO1*.

Maintenant, cet automate est prêt à être utilisé comme générateur d'impulsions,

La configuration d'un axe (technologie objet) :

Pour commencer la programmation, on ajoute "Objets Technologiques"

Nous choisisons

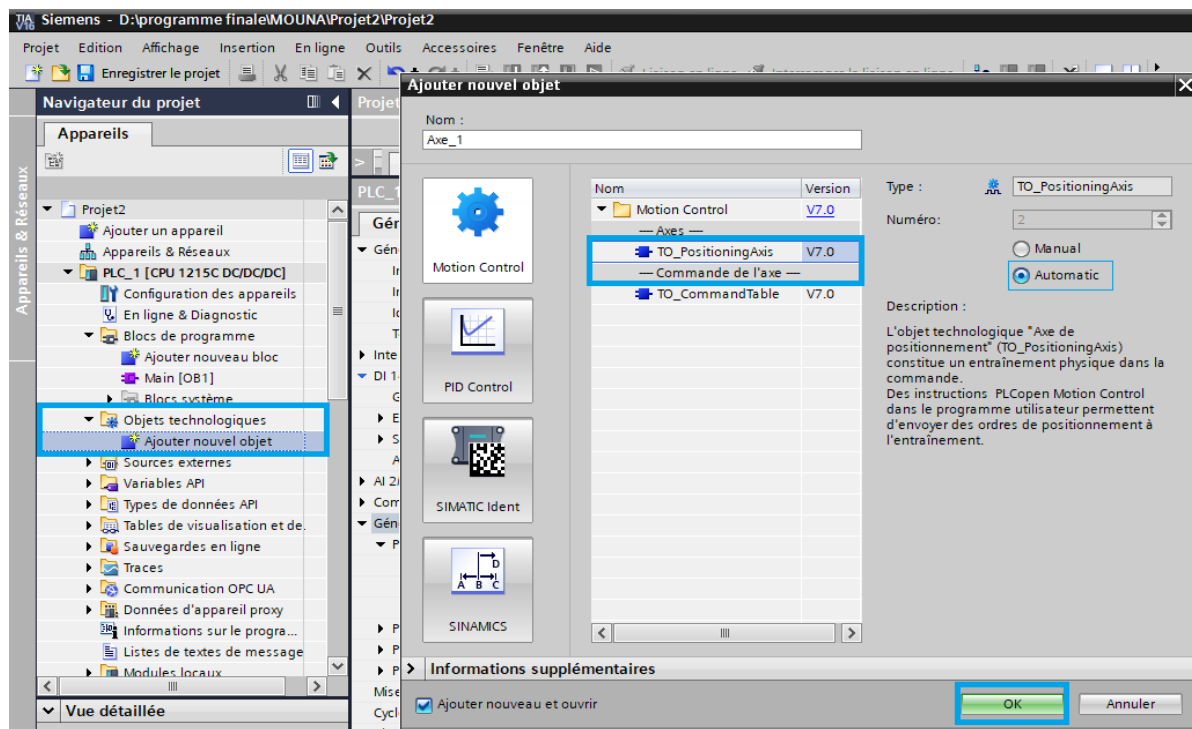


Figure IV. 9 : La configuration de générateur d'impulsions.

Voici les étapes à suivre pour configurer un objet axe :

- Sélectionnez le générateur d'impulsions que vous configurez et la sortie d'impulsions

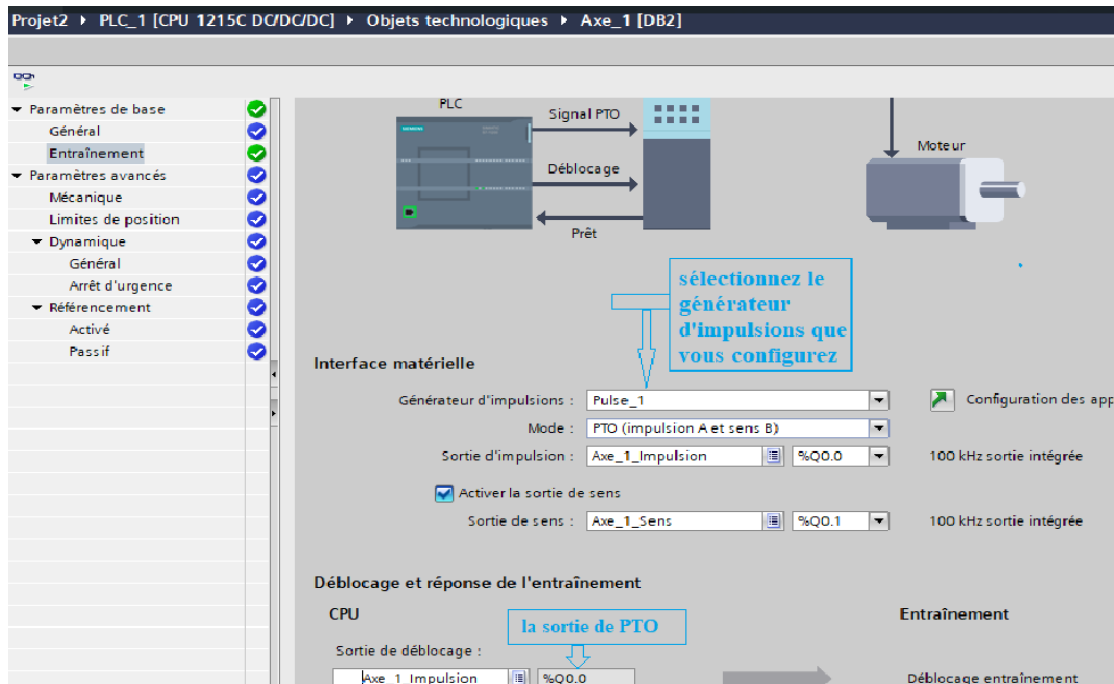


Figure IV. 10 : La configuration d'un objet axe.

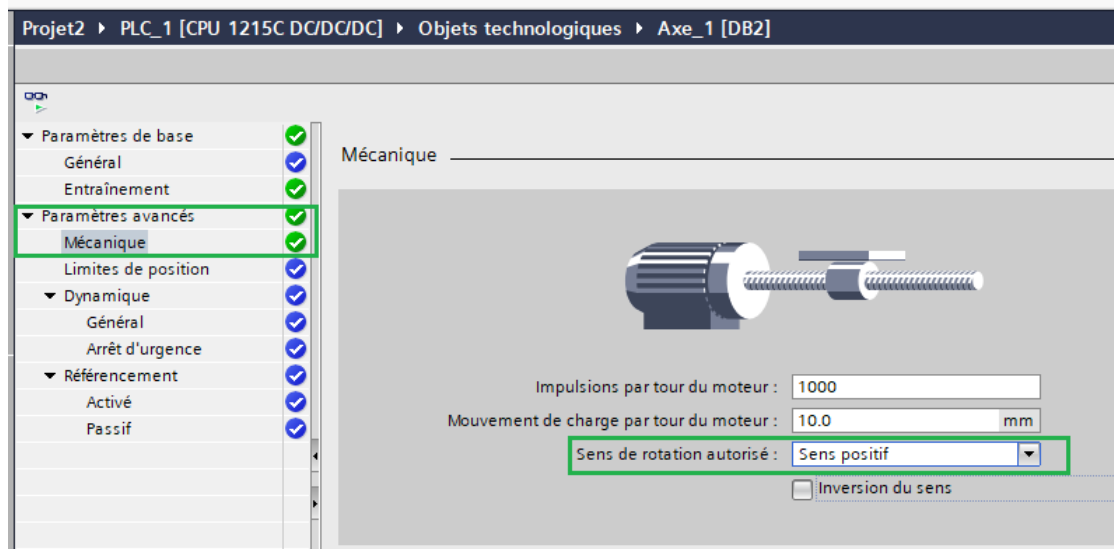


Figure IV. 11 : Configuration de moteur en sens de rotation directe (sens positif)

Motion contrôle

Ceci signifie que l'automate contrôle le mouvement du moteur (position et vitesse) en utilisant un générateur de train d'impulsions pour contrôler l'entraînement du moteur et le mouvement du moteur.

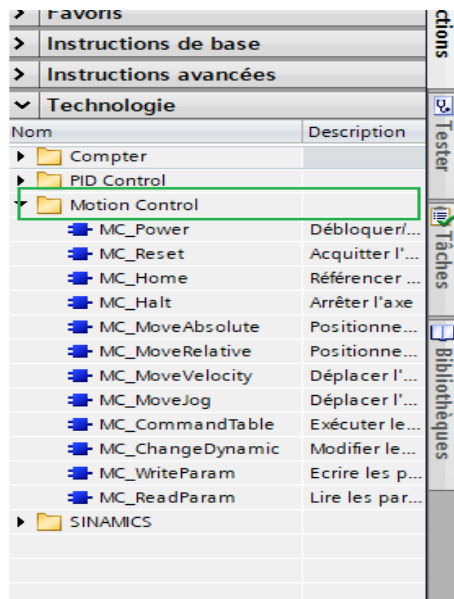


Figure IV. 12 : l'insertion de l'instruction Motion contrôle.

1. Comment l'utiliser le bloc MC_power

L'instruction Motion Control "MC_Power" permet de débloquent ou de bloquer un axe et une impulsion.

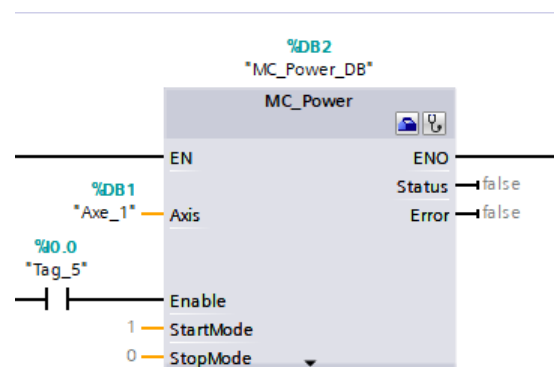


Figure IV. 13 : Bloc "MC_Power".

- EN: Activation de bloc "MC_Power"
- Axis: On commence par sélectionner les AXES pour lesquels la configuration a été faite
- ENABLE: ACTIVE (enable 1) OU DESACTIVE (enable 0) AXIS.

2. Comment utiliser Le Bloc MC_MoveRelative

L'instruction Motion Control "MC_MoveRelative" démarre un mouvement de positionnement par rapport à la position de départ. (déplacement jusqu'à une position d'axe précisée).

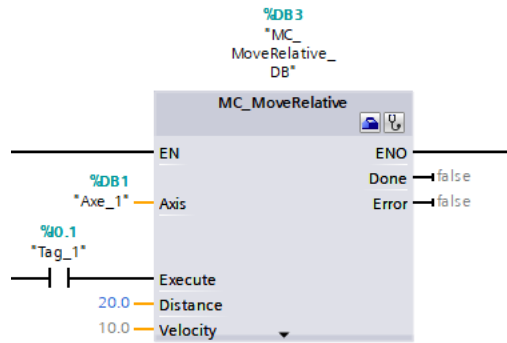


Figure IV. 14 : Bloc "MC_MoveRelative".

- EN : Activation de Bloc "MC_MoveRelative"
- Axis: On commence par sélectionner les axes pour lesquels la configuration a été faite.
- Execute: On met l'entrée que si le signal lui vient, les impulsions sortent (I0.1).
- Distance: Nous définissons la distance de rotation du servomoteur.
- Velocity: Détermine la vitesse de rotation du servomoteur.
- Done: Indique si nous avons fini d'envoyer l'impulsion.
- Error: Indique s'il y a une erreur et précise l'erreur.

Il est maintenant prêt à le simuler

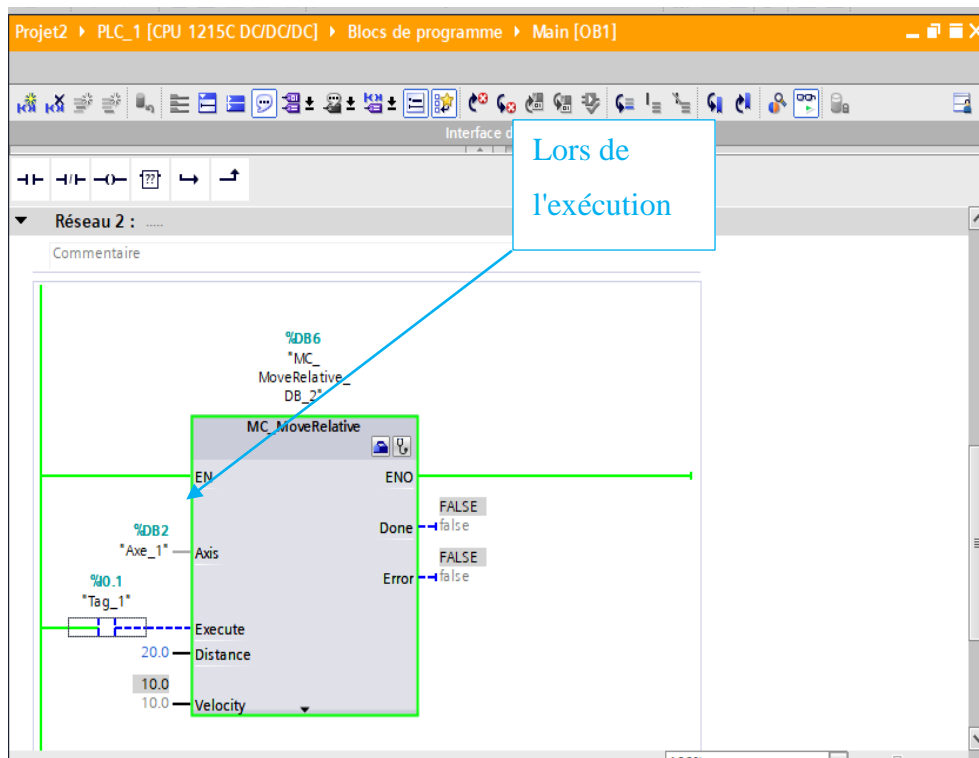


Figure IV. 15 : La simulation de Bloc "MC_MoveRelative"

3. Bloc TP Time

Le temps programmé (*PT*) commence lorsque l'instruction démarre. Dans ce temporisateur, même si un nouveau front positif est détecté, l'état du signal à la sortie *Q* n'est pas affecté tant que la durée *PT* s'écoule.

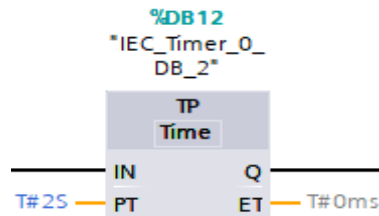


Figure IV. 16 : Bloc TP Time.

4. Bloc CTU

L'instruction "Comptage" permet d'incrémenter la valeur à la sortie *CV*.

- *CU* => Entrée du compteur. La valeur du compteur est incrémentée de un lorsque l'opérande reliée à cette entrée passe de "0" à "1". Le compteur peut être incrémenté jusqu'à ce que la valeur supérieure du type de donnée (INT) soit atteinte.
- *R* => Entrée de réinitialisation. La valeur de sortie *CV* (valeur de comptage en cours) est remise à zéro lorsque l'entrée *R* passe à "1". Tant que *R* est à "1", l'entrée *CU* n'a pas d'effet sur l'instruction.
- *PV* => Valeur à laquelle la sortie *Q* est mise à "1". Cette entrée doit être de type INT.
- *Q* => Etat du compteur. Cette sortie est mise à "1" lorsque la valeur actuelle de comptage est supérieure ou égale à la valeur *PV*.
- *CV* =< Valeur de comptage en cours. Cette donnée doit être de type INT.

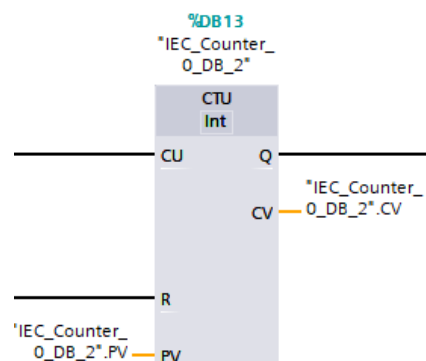


Figure IV. 17 : Bloc CTU.

V.4.3 Phase finale

- Étape finale pour les tests.

Après le chargement du programme dans l'automate, on procède à plusieurs séries de tests

Essai en mode automatique

Dans ce mode l'automate détecte l'objet et le place dans la position desserrée. Les figures suivantes résument les différentes étapes:

Étape 0

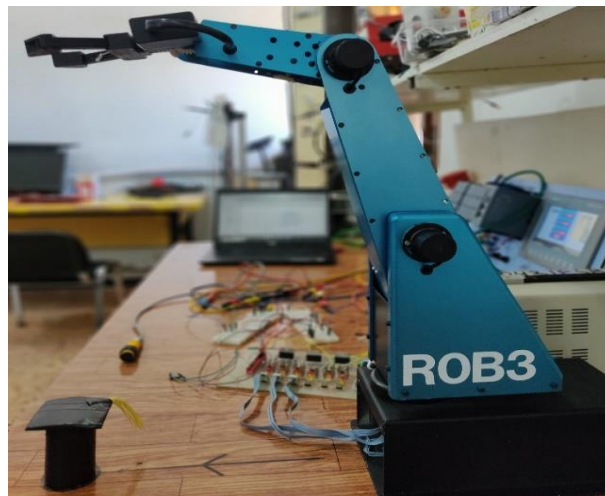


Figure IV. 18 : La position initiale de bras.

Étape 1 et 2

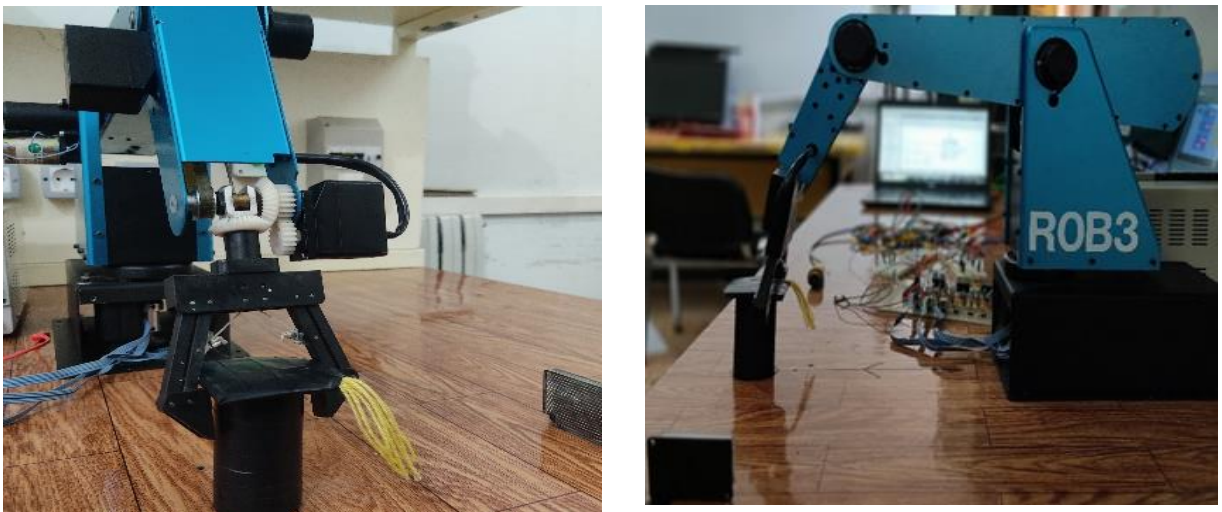


Figure IV. 19 : Détection de l'objet et déplacement du bas et fermeture de la pince.

Etape 3 et 4

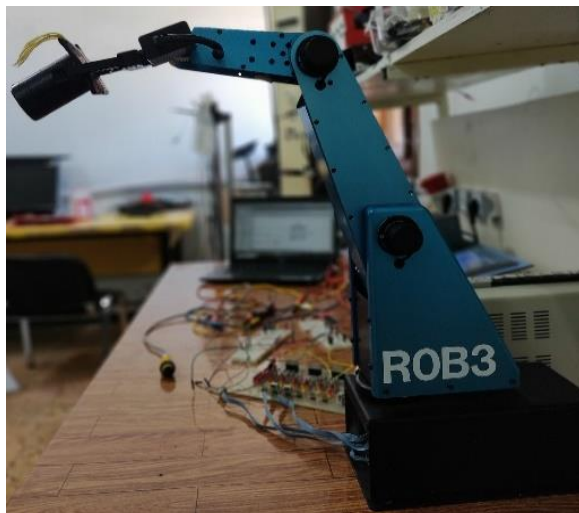


Figure IV. 20 : Déplacement de l'épaule vers le haut et rotation du bas.

Etape 5 et 6

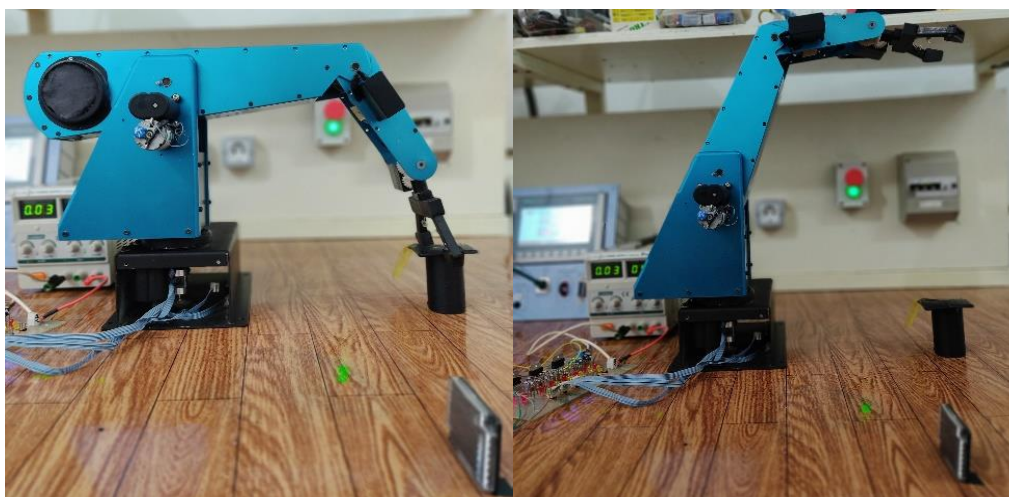


Figure IV. 21 : Déplacement de l'épaule vers le bas et ouverture de la pince.

Etape 7



Figure IV. 22 : Déplacement du robot à sa position initiale pour exécuter une autre tâche

Supervision HMI

Programme de la supervision pour le contrôle et le bon fonctionnement du système.

Interface HMI

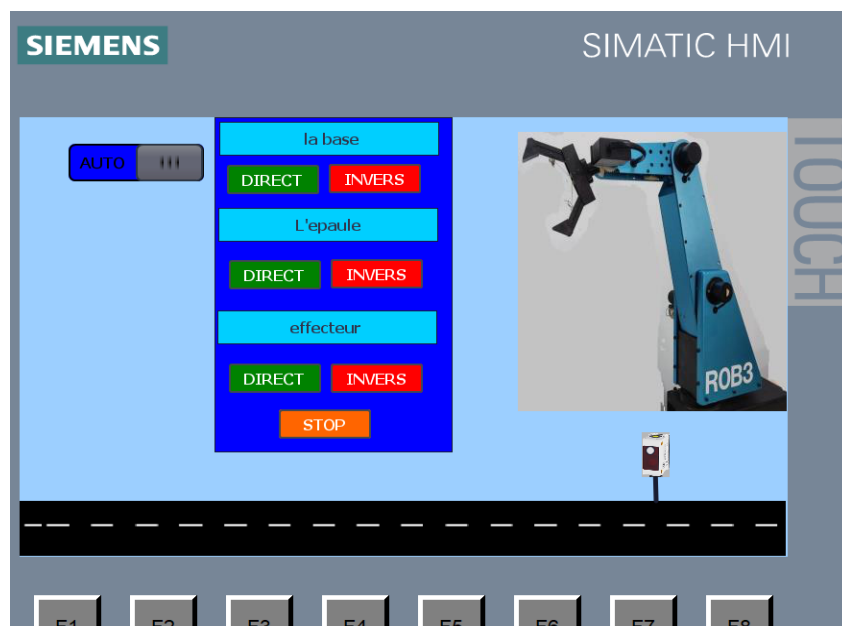


Figure IV. 23 : Interface HMI.

Table de mnémoniques

Les adresses utilisées dans notre programme "table de mnémoniques"

- Les entrées

Nom	Type de données	Adresse
START SYS	Bool	%I0.0
P PIECE	Bool	%I0.1

Tableau IV . 1 : Table de Mnémotique des entrées.

- Les sorties:

Nom	Type de données	Adresse
ORG TER DIRECT	Bool	%Q0.0
BAS DIRECT	Bool	%Q0.2
BAS INVERS	Bool	%Q0.3
Axe_1_Impulsion	Bool	%Q0.4

Tableau IV . 2 : Table de Mnémonique des sorties.

Les Mnémoniques du HMI

Mnémonique	Type de données	Connexion	commentaire
Axe_1_Impulsion	Bool	HMI_Liaison_1	Animation
Axe_2_Impulsion	Bool	HMI_Liaison_1	Animation
Bas direct	Bool	HMI_Liaison_1	Affichage de la sortie du bas dans le sens direct dans l'HMI
Bas inverse	Bool	HMI_Liaison_1	Affichage de la sortie du bas dans le sens inverse dans l'HMI
ORG TER DIRECT	Bool	HMI_Liaison_1	Affichage de la sortie de l'effecteur dans le sens direct dans l'HMI
ORG TER INVERSE	Bool	HMI_Liaison_1	Affichage de la sortie de l'effecteur dans le sens inverse dans l'HMI
P PIECE	Bool	HMI_Liaison_1	Affichage de la sortie de capture dans l'HMI
STOP	Bool	HMI_Liaison_1	Arrêt de système

Tableau IV. 3 : Table de Mnémonique HMI.

IV.5. Conclusion

Dans ce chapitre, nous avons présenté un détail sur la programmation de l'automate *s7 1200*. Nous avons présenté aussi le test de déplacement d'objet puis le retour à la position initiale. La commande de bras est réalisée en employant une interface de supervision *HMI*. Les différents tests que nous avons effectués ont confirmé le bon travail du prototype et la tâche demandée par le cahier de charges.

Conclusion générale

Conclusion générale

Finalement, le *ROB3* est en état de marche. Ceci est réalisé grâce à nos études dans plusieurs domaines de l'informatique, de l'électronique et de l'automatique.

Ce projet nous a été une expérience très bénéfique. Il nous a permis d'exploiter les différentes notions théoriques vues durant notre période de formation. Il nous a permis également de mettre le premier pas dans le monde pratique de la commande robotique. Ce projet nous a permis une maîtrise du logiciel *TIA PORTAL*.


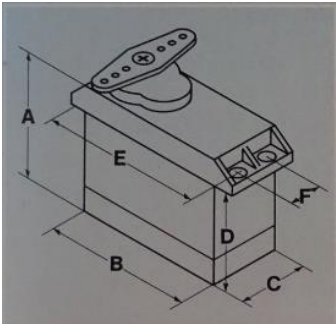
Ce projet de fin d'études a été pour nous l'occasion d'expérimenter enfin nos études théoriques, de porter notre savoir dans ce terrain. Il été une expérience riche dans divers domaines et plain d'échanges avec l'équipe du laboratoire, nos professeurs, ainsi que les autres groupes d'étudiants.

Enfin nous espérons que notre travail va être amélioré dans le futur par d'autres étudiants

Les annexes


Annexe 1

Caractéristiques techniques du servo moteur :

MULTIPLEX STELL-SERVO	
Dimensions 	A=41mm B=46mm C=23mm D=26mm E=54mm F=12mm
Poids	65gr
Tension d'alimentation	4.3V à 5.8V
Vitesse	2.5Tr/min
Couple	80cmN
Amplitude	De 0 à 180°.
Numéro de sérier	Nr.6 5089

Annexe 2

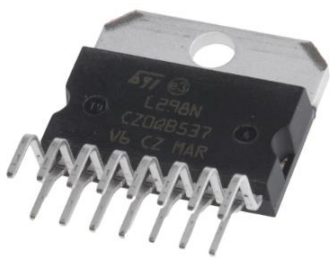
Caractéristiques techniques capteur de présence

Capteur de présence OMRON E3F-R2C4	
Type	Photoélectrique
Configuration de sortie	NPN

Courant de sortie	Max.300 mA
Fréquence max	400Hz
Température de travail	-25...55°C
Tension d'alimentation	10...36V DC
Poids brut	99.757g
Mode d'activation	Reflex avec projecteur 4
Temps de réponse	30ms
La longueur de câble	1.5M
Objet de détection standard	5×5cm objet


Annexe 3

Caractéristiques techniques du Module de commande de moteur L298N

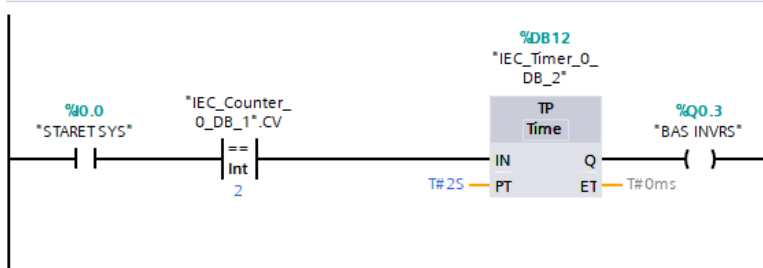
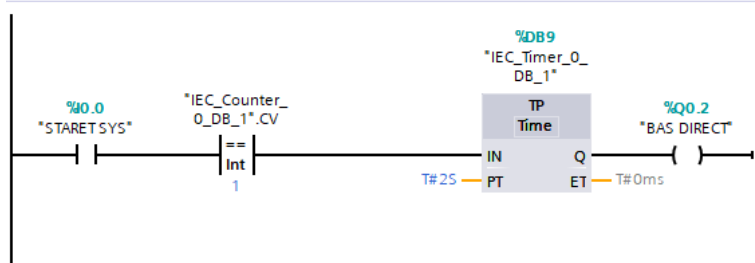
L298N	
• Modèle de pilote	L298N 2A
• Puce du pilote	double pont en H L298N
• Tension d'alimentation du moteur	46 V
• Courant d'alimentation du moteur	2 A
• Tension logique	5 V
• Tension du pilote	5-35 V
• Courant du pilote	2 A
• Courant logique	0-36 mA
• Puissance maximale (W)	25 W

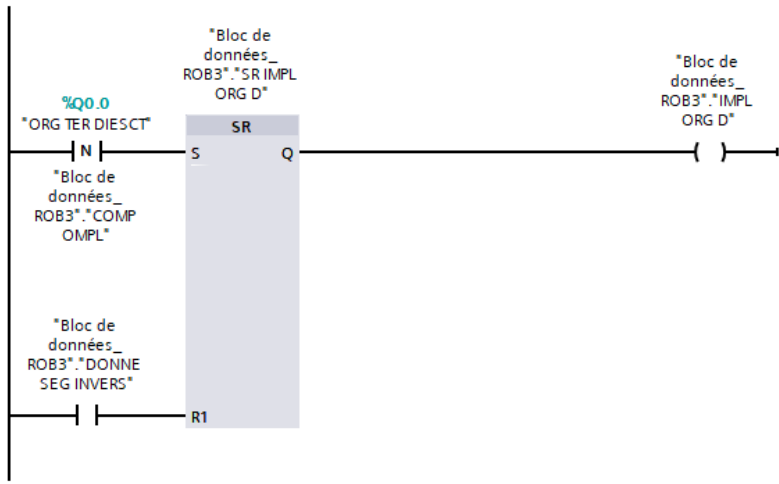
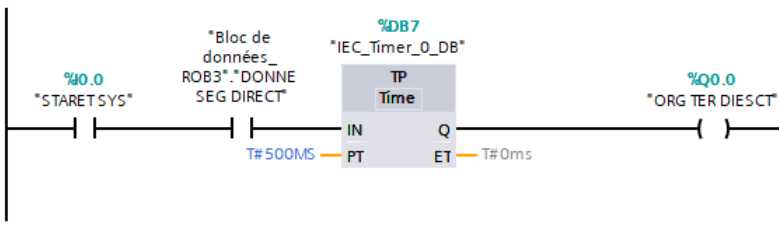
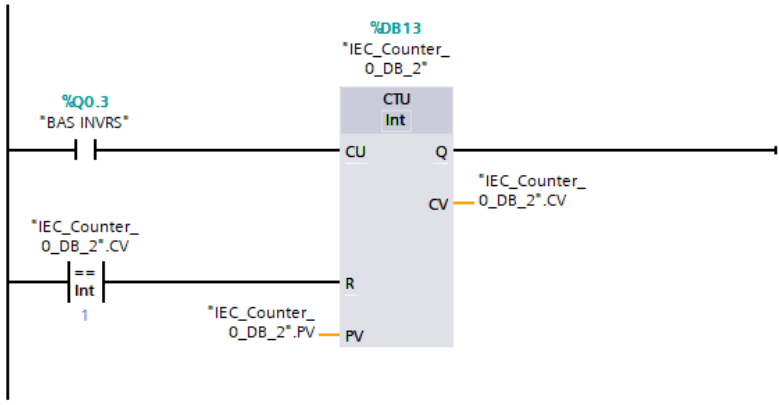
Annexe 4

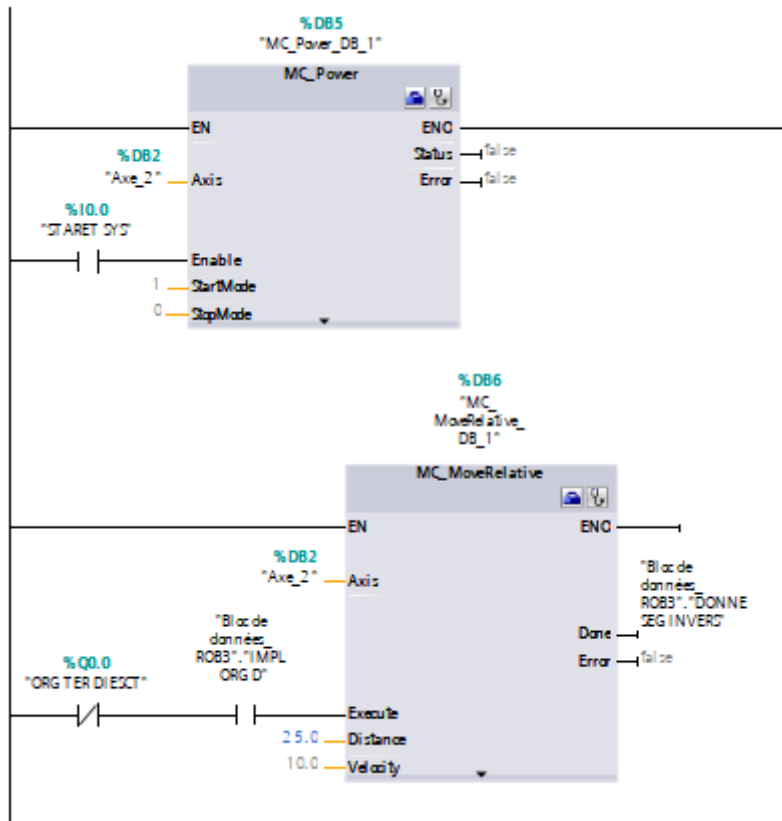
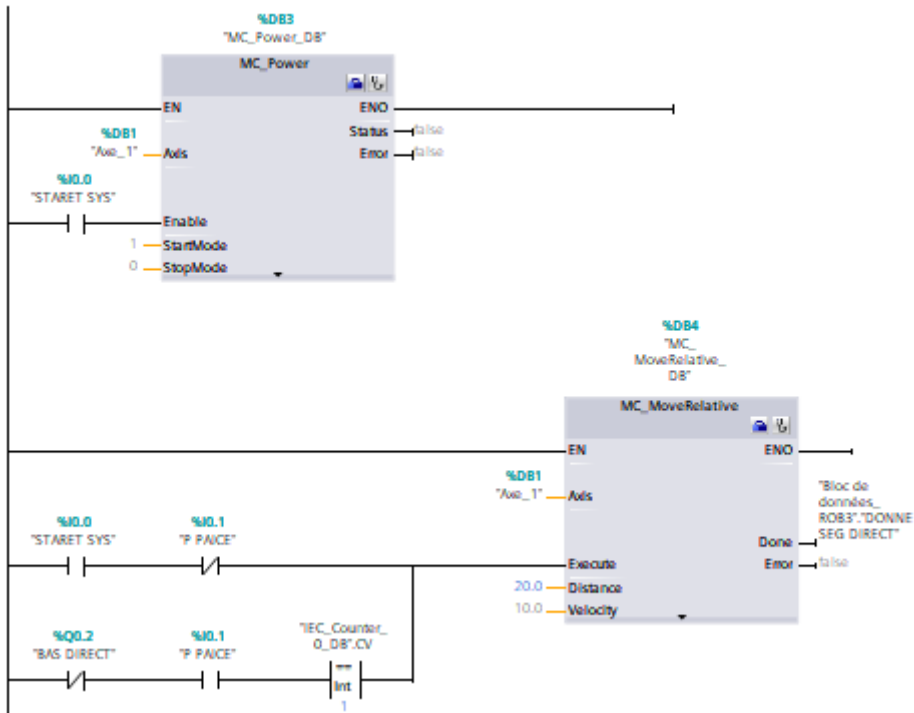
Caractéristiques techniques de moteur cc

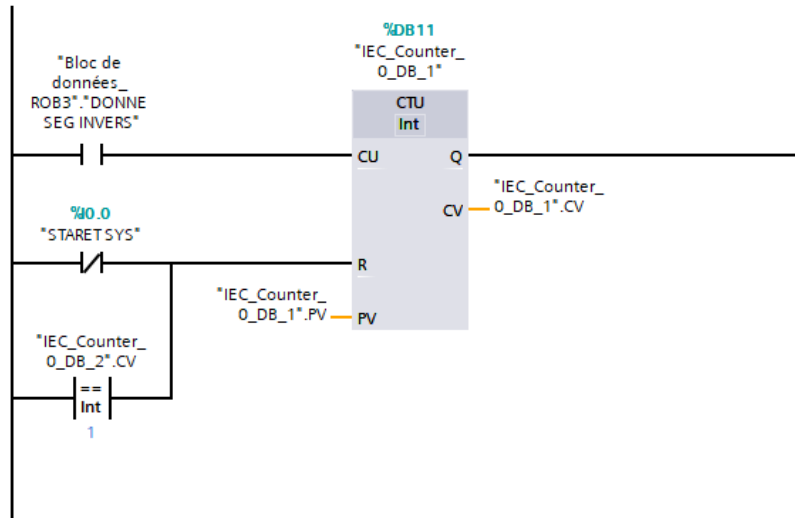
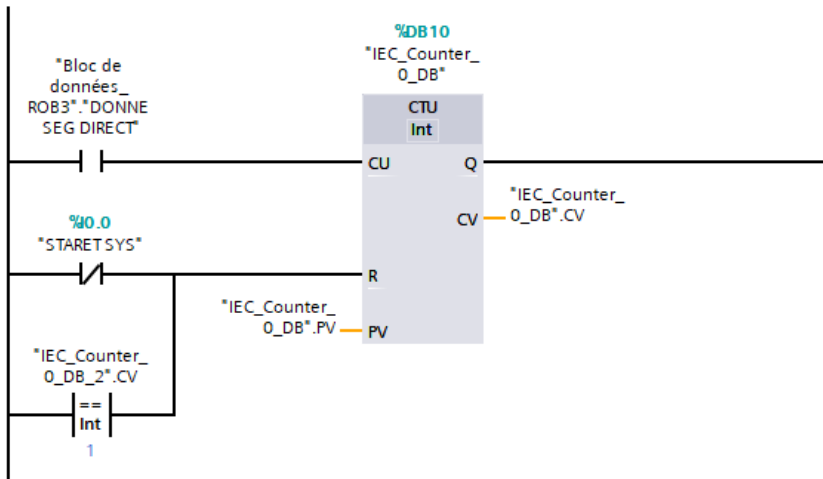
moteur à courant continu «Bühler»	
N° de série	1.61.042303-1
Tension d'alimentation	9 V
Partie	Moteur à engrenages
La vitesse de rotation	2,5 tr/min
intensité du courant d'induit	3 A

Annexe 5









Résumé

Depuis la révolution industrielle, une discipline a marqué l'évolution du monde technologique: la Robotique. L'avènement des robots dans l'industrie a permis de soulager l'homme des travaux répétitifs et difficiles tels que: le déplacement d'objets lourds, les tâches d'assemblages, les microsoudures... etc.

Ce projet a pour but de commander un bras manipulateur *ROB3* qui contient des servomoteurs commandés par un signal PTO.

Ce robot aura pour tâches de faire déplacer un objet en utilisant un *API* pour le contrôler et une interface *HMI* pour superviser en temps réel et un capteur de présence pour détecter la pièce.

Abstract

Since the industrial revolution, a discipline has marked the evolution of the technological world: Robotics. The advent of robots in industry has made it possible to relieve humans of repetitive and difficult work such as: moving heavy objects, assembly tasks, micro-welding, etc.

This project aims to control a manipulator arm *ROB3* which contains servomotors controlled by a PTO signal.

This robot will have the tasks of moving an object using an *API* to control it and an *HMI* interface to supervise in real time and a presence sensor to detect the part.

Mots clé:

API . PLC . CPU . LADDER . TIA PORTAL . L298N . LM7805 . PTO . SERVOSMOTEUR
3ddl . HMI .MGI.