الجمهورية الجزائرية الديمقراطية الشعبية

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITÉ Dr. YAHYA FARES DE MEDEA

FACULTE DE DES SCIENCES ET DE LA TECHNOLOGIE

Département du Génie Électrique et de l'informatique



Projet de Fin d'Etudes

Présenté pour l'obtention du diplôme

Master en Génie Électrique

Spécialité : Automatique des Systèmes

THEME

Contrôle d'un bras de robot manipulateur à cinq axes

Proposé et dirigé par,

Dr.Mounir Bouhedda

Préparé et présenté par,

Mlle. Karima Abdellatif

Année universitaire 2014-2015

من أجمل ما قال الإمام الشافعي

ما الفخر إلا لاهل العلم إنهـم ... على الهدى لمن استهدى أدلاء

وقدر كل امرئ ما كان يحسنه ... والجـاهـلون لاهل العلـم أعـداء

ففز بعلـم تعش حيـا بـه أبـدا ... النـاس موتـى و أهل العلم أحياء

ما تم حلم ولا علم بلا أدب ولا تجاهل في قوم حليمان وما

التجاهل إلا ثوب ذي دنس وليس يلبسه إلا سفيهان

العلم حر وطالب العلم عبد ، فإن خدم العلم ملك العلم....

وإن تجبر عليه...

فالعلم أشد تجبرًا من أن يخضع لمن لا يخضع له....



A l'astre qui s'est éteint et qui continue toujours de m'éclaircir sur la voie
de la clairvoyance, de la sagesse et du savoir.

A celle qui m'a appris le sens sacré de l'amour, la compassion,
L'humanisme ainsi que les plus belles valeurs humaines.

A celle qui a fait en sorte que je voie le jour.

A celle qui a essuyé mes larmes

A celle qui m'a donné le courage d'avancer.

A celle qui n'a pas vécu pour assister à ce jour de triomphe,

A ma chère mère

Que je dédie ce modeste travail.

Que Dieu, le Tout-Puissant,
l'accueille en son vaste paradis.

اللهم ارحمها و اغفر لها و اجعلها من سيدات أهل الجنة

آمين

Hommage respectueux

J'exprime toutes mes reconnaissances amicales à ***Dr. M.BOUHEDDA*** *, Maître de Conférences* à *l'Université de MEDEA, qui a encadré mes travaux de mémoire de fin d'études sur un sujet très intéressant dans le domaine de la robotique. Vos très nombreuses connaissances m'ont permis de bien progresser, et de mener avec succès ces travaux de mémoire. Je vous remercie, Mr, pour vos remarques et conseils précieux. Vous êtes une personne d'une très grande écoute et d'un grand dynamisme. Je vous remercie encore pour vos qualités humaines indéniables et votre disponibilité. Je suis très fière d'avoir travaillé avec vous, en Licence et en Master.*

Remerciement

Je remercie dans un premier temps, mon Dieu le Tout Puissant, qui m'a accordé le souffle de vie afin d'arriver au terme de mon cycle de formation, que l'honneur et la gloire Lui soient rendus.

J'exprime mes profonds remerciements aux membres de jury, Vous nous faites l'honneur d'accepter avec une très grande amabilité de siéger parmi notre jury de mémoire. Veuillez accepter ce travail maître, en gage de mon grand respect et ma profonde reconnaissance.

Ma gratitude va aussi à l'ensemble des enseignants du département de génie électrique et de l'informatique, pour leur encadrement universitaire.

J'exprime ma gratitude à tous les consultants et internautes rencontrés lors des recherches effectuées et qui ont accepté de répondre à mes questions avec gentillesse, surtout Mr.M.Skander et Mr.A.Tebib Chef de service de maintenance du groupe pharmaceutique SAIDAL.

Que tous ceux qui m'ont aidé de près ou de loin, trouvent ici l'expression de ma gratitude.

Karima Abdellatif

Dédicace



Je dédie ce modeste travail à mes chers parents

Je vous avoue le respect que j'ai toujours eu pour vous. Rien au monde ne vaut les efforts fournis jour et nuit pour mon éducation et mon bien être.
Ce travail est le fruit de vos sacrifices que vous avez consentis pour mon éducation et ma formation.

A mon petit frère Adel, mes sœurs Rania, Amina, Nadia et son fils Alaa-Eddine, je consacre pour vous toutes mes paroles et mes expressions fraternelles, je vous remercie pour vos soutiens quotidiens, vos prières.

Je n'oublie jamais mes proches, mes amies qui sont été toujours à mes cotés.

A toute la promotion de master 2014/2015, a mes camarades de la faculté des sciences et de technologie de Médéa.

Karima

Résumé:

Le but de ce travail est la mise en œuvre du contrôle d'un robot à 5 axes (ED-7255). Comme application, le robot a été connecté à une mini-usine afin de réaliser des taches industrielles. Une interface Homme-machine est réalisée pour donner plus de souplesse au système de contrôle global réalisé autour d'un API.

Mots Clés: API, IHM, Mini-Usine, Robot.

Abstract:

The goal of this work is the control of 5 axis robot. As an application, the robot has been connected to a mini-factory in order to achieve industrial tasks. A man-machine interface is realized to provide more flexibility to the global control system carried around the API.

Keywords: PLC, HMI, Mini-factory, Robot.

ملخص

الهدف من هذا المشروع هو تطبيق نظام تحكم في ذراع ألي ذو 5 محاور. كتطبيق لذلك تم وصل الروبوت بمصنع مصغر لأجل تحقيق مهام صناعية. كذلك تم تحقيق واجهة إنسان-آلة بهدف إضفاء مرونة لنظام تحكم كلي حول آلة مبرمجة صناعية.

كلمات مفتاحية : آلة مبرمجة صناعية , واجهة إنسان-آلة , مصنع مصغر, ذراع ألي

Liste des abréviations

MGI : Modèle Géométrique Directe.

MGD : Modèle Géométrique Indirecte.

CAO: Conception Assisté par Ordinateur.

CFAO: Conception de Fabrication Assistée par Ordinateur.

IHM: Interface Homme Machine.

API: Automate Programmable Industriel.

TCP: Tool Center Point.

SAP: Système Automatisé de Production

DDL: Degré De Liberté.

SOMMAIRE

INTRODUCTION GENERALE 1

Chapitre I **Généralités sur les Robots**

I.1.INTRODCUCTION 3
I.2. CONSTITUANTS MECANIQUE DES ROBOTS 4
I.3. LES DIFFERENTS TYPES DES ROBOTS **6**
I.4. CHOIX DU NOMBRE DE DEGRES DE LIBERTE D'UN ROBOT **11**
I.5. CARACTERISTIQUES D'UN ROBOT **12**
I.6. GENERATION DE MOUVEMENT DES ROBOT **13**
I.6.1. Espace articulaire 14
I.6.1. Espace opérationnel 14
I.7. LES DIFFERENTS MODES DE PROGRAMMATION DES ROBOTS **14**
I.7.1. Programmation en ligne 14
I.7.1. Programmation hors-ligne 15
I.8. DOMAINES D'UTILISATION DES ROBOTS 18
I.9. LES EXIGENCES DE SECURITE 21
I.10. CONCLUSION 22

Chapitre II **Présentation et modélisation de robot ED-7255**

II.1.INTRODCUCTION 23
II.2. COMPOSANTS ET CARACTERISTIQUES TECHNIQUES DU PRODUIT 24
II.2.1 Le Bras articulé 24
II.2.1.1 Point de vue mécanique 25
II.2.1.1 Modélisation des bras de robots 28
II.2.1.1. Modèle géométrique inverse du robot à 5 degrés de liberté : (ED-7255) 35
II.2.1 Contrôleur (ED-MC1) 36
II.2.1 Teach Pendant (Console de commande) 40
II.2.1 Présentation de langage de programmation ED-IRS 42
II.2. THEOREME D'ACCELERATION EN ROBOT ED-7255 45
II.2. CONCLUSION 46

Chapitre III **Présentation de la mini-usine et IHM**

III.1.INTRODCUCTION 47
III.2. DESCRIPTION GENERALE DE LA MINI - USINE A PRODUCTION 47
III.3. ARCHITECTURE D'UN SYSTEME AUTOMATISE DE PRODUCTION 48
III.4. ARCHITECTURE MATERIELLE DE L'API 52
III.5. CHOIX D'UN AUTOMATE PROGRAMMABLE INDUSTRIEL 54
III.6. LANGAGES DE PROGRAMMATION POUR 54
III.7. SOFTWARE 55
III.7.1 Découverte du logiciel GMWIN 55
III.7.2 L'interface de logiciel 55
III.8. L'INTERFACE HOMME MACHINE 58
III.8.1 Introduction 58

SOMMAIRE

III.8.2 Logiciel de conception et de configuration d'interface TouchWin 59
III.8.3 Utilisation des variables 60
III.8. CONCLUSION.....61

Chapitre IV **REALISATION PRATIQUE**
IV.1.INTRODCUCTION 62
IV.2. LE CAHIER DES CHARGES 62
IV.3. PROGRAMMATION D'UNE COMMANDE DU ROBOT 5 AXES ED-7255.....62
IV.4. CALIBRATION DES ROBOTS ET LEURS POSITIONS DE REPOS.....65
IV.5. MODE DE DEPLACEMENT DE L'ORGANE TERMINAL 65
IV.6. METHODOLOGIE DE PROGRAMMATION DE ROBOT 66
IV.7. LA MANIPULATION«PICK A N D PLACE» 66
IV.7.1 Développement de programme d'un « pick and place » classique 67
IV.7.1.1 Commande en position dans l'espace articulaire 67
IV.7.1.2 Commande en position dans l'espace opérationnel 68
IV.7.1.3. Phase de compilation et simulation 69
IV.7.2 Commande de robot ED-7255 avec une condition pour la comparaison. 70
IV.7.2.1. Description de la tâche à faire 70
IV.7.2.1. Développement du programme de commande 70
IV.8. RESOLUTION DE LA PROBLEMATIQUE DE LA TOUR D'HANOI 72
IV.9. REALISATION D'UNE CELLULE DE PRODUCTION ROBOTISEE AVEC UNE INTERFACE HOMME - MACHINE (IHM) 77
IV.9.1. Développement du projet pour la gestion de la mini-usine 77
IV.9.2. Création de la table des mnémoniques (Partie Software).....79
IV.9.3. Réalisation du programme de la gestion de cellules de Production 79
IV.9.4. Programme de la commande de Robot.....80
IV.12. ELABORATION ET CONCEPTION DE LA COMMANDE COMPLETE POUR LA GESTION DE LA CELLULE ROBOTISEE AVEC UNE INTERFACE DE CONTROLE ET DE COMMANDE HOMME MACHINE.....81
IV.10.1. La commande globale de l'API.....84
IV.10.2. Programme global de commande du Robot 87
IV.10.3. Elaboration de l'interface homme-machine configurée par ToucheWin.....90
IV.11. VUE GENERALE SUR TOUT LE PROCESSUS EN FONCTIONNEMENT 95
IV.12. CONCLUSION.....96

CONCLUSION GENERALE.....97

BIBLLOGRAPHIE

ANNEX

LISTES DES FIGURES

Chapitre I

Figure I.1. Chaine simple ouverte 4
Figure I.2. Chaine fermée. 5
Figure I.3. Robot à structure complexe arborescent 5
Figure I.5. Schéma synoptique d'un robot SCAR..... 6
Figure I.6. Robot DENSO HM..... 6
Figure I.7. Robot Adept 6
Figure I.8. Le robot cylindrique 7
Figure I.9. Exemple d'un robot cylindrique 7
Figure I.10. Robot sphérique 8
Figure I.11. Robot FANUC 8
Figure I.12. Robot Cartésien. 8
Figure I.13. Robot Toshiba 9
Figure I.14. Robot ABB. 9
Figure I.15. Robot anthropomorph..... 10
Figure I.16. Exemple d'un bras robot..... 10
Figure I.17. Robot ABB..... 11
Figure I.18. Structure d'un robot non redondant..... 11
Figure I.19. Programmation par langage robotique-RAPID (Robots ABB..... 16
FigureI .20. Programmation graphique..... 17
Figure I.21. Démarche de conception et de programmation d'un site robotisée 18
Figure I.22. Chaîne d'assemblage 19
Figure I.23. Robot soudeur a l'arc..... 19
Figure I.24. Manipulateur rapide dans le secteur Agro-alimentaire..... 19
Figure I.25 Manipulateur pour les opérations pénible et dangereuse..... 20
Figure I.26. Robot canadarm 2..... 20
Figure I.27. Robot sous-marin..... 20
Figure I.28. Drone militaire..... 20
Figure I.29. Aspirateur..... 21
Figure I.30. Robot humanoide..... 21
Figure I.31. Assistance aux opérations chirurgicales..... 21

Chapitre II

Figure II.1. Eléments fournis avec le robot ED-7255..... 23
Figure II.2. Les cinq axes du robot..... 24
Figure II.3. Dessin correspont à l'effecteur..... 24
Figure II.4.Schéma dimmensionnel du robot ED-7255 26
Figure II.5. Espace de travail du robot 27
Figure II.6. Paramètres de *Danavit -Hartenberg*..... 30
Figure II.7. Placement des repères et notation pour le robot 5ddl. 31
Figure II.8. Codeur rotatif absolu (MA3)..... 35
Figure II.9. Vue arriere du controlleur ED-MC1 de robot 36
Figure II.10. Vue de face du controleur de robot..... 37
Figure II.11. Architecture Hardware de contrôleur..... 39
Figure II.12 Les composants internes de contrôleur..... 39
Figure II.13. Console de commande (Teach pendent). 41
Figure II.14. Architecture de simulateur 43
Figure II.15. Vue d'ensemble de l'interface de logiciel de programmation ED-IRS.....43
Figure II.16.*Théorème d'accélération décélération*..... 45
Figure II.17. Point au milieu de chevauchement entre un mouvement et un mouvement 45
Figure II.18 Le mouvement réel dans la figure II.17... 45
Figure II. 19 *Point final de chevauchement entre un mouvement et un autre.* 46
Figure II.20 . *Le mouvement réel de la figure II.19*.....*46*

Chapitre III

Figure III.1.Vue de face sur la mini-usine..... 48
Figure III.2 Architecture d'un système automatisé de production SAP.....49
Figure III.3. Vérin à simple action..... 50
Figure III.4. Vérin à double action 50
Figure III.5. Capteur des pièces métalliques 51
Figure III.6. Détecteure photoélectrique de la mini-usine..... 52
Figure III.7. Architecture matérielle d'un API 53
Figure III.8. Cycle de fonctionnement de l'API..... 53
Figure III.9. Simulateur de l'API GLOFA..... 55
Figure III.10. L'interface de GMWIN 55
Figure III.11. la boite à outil. 56
Figure III.12. Structure de l'écran touchwin..... 58
Figure III.13 L'interface Homme/Machine dans un processus automatisé.....59
Figure III.14. Fenêtre de choix de l'API correspond 59
Figure III.15. Vue d'ensemble de l'interface de logiciel de conception TouchWin 60

Chapitre IV

Figure IV.1. Étapes pour la création d'un programme via ED-IRS..... 63
Figure IV.2 Boucles de génération de mouvement.....64
Figure IV.3 Choix du mode de la commande 64
Figure IV.4. Différent mode de déplacement 65
Figure IV.5. Tableau de coordonnées des positions articulaires 66
Figure IV.6. Environnement de travail virtuel. 67
Figure IV.7. Connexion avec le contrôleur de robot ED-7255 69
Figure IV.8. Sélection de choix pour transférer (download) 69
Figure IV.9 Transfert de programme en cours. 69
Figure IV.10 Connexion en cours avec le contrôleur 69
Figure IV.11. Environnement de travail virtuel 70
Figure IV.12. Image correspondante à la tour d'HANOI 72
Figure IV.13. Résolution de la tour d'Hanoi à 4 disques. 73
Figure IV.14. L'état initial de la tour 74
Figure IV.15 Des images du robot au cours de résoudre la problématique 76
Figure IV.16 Organigramme de la procédure de fonctionnement 78
Figure IV.17 La table des mnémoniques 79
Figure IV.18 Programme en GRAFCET.. 80
Figure IV.19 Le poste d'attente prise pièce par le robot (pick). 81
Figure IV.20 Les deux emplacements P1 81
Figure IV.21 Connexion et communication entre API, Contrôleuret l'IHM 82
Figure IV.22 Organigramme descriptif de fonctionnement de la synchronisation API, Robot, IHM 83
Figure IV.23. La table des mnémoniques 85
Figure IV.24. Programme LADDER correspond au quelques tâches 86
Figure IV.25. Vue d'accueil 91
Figure IV.26. Les vues de commande, production, Tours d'HANOI 92
Figure IV.27. Vue visualisation 93
Figure IV.28. Vue Robot 93
Figure IV.29. Vue « visualisation » 94
Figure IV.30. Vue « Robot » 94
Figure IV.31 Vue « Tour d'Hanoi » avec la tache choisis 94
Figure IV.32 Vue générale sur tout le processus. 95

Liste des tableaux

Chapitre I

Tableau I.1.Nombre d**e** morphologies possibles en fonction du nombre de degrés de liberté de robot.……10

Chapitre II

Tableau II.1.Caractéristiques mécaniques de robot ED-7255…….……25

Tableau. II .2 Paramètres géométriques du robot…….32

Tableau II.3 Caractéristique de contrôleur d'ED-7255.....38

Tableau II.4 Les principaux composants du contrôleur……40

Tableau II.6 Menu de ED-IRS……...44

Chapitre III

Tableau III.1. Désignation de la boite à outils……..56

Tableau III.2. Barre d'outils de logiciel GMWIN……57

Chapitre IV

Tableau IV.1. Programmation d'un pick and place passant par la MGD…….….67

Tableau IV.2. Programmation d'un pick and place passant par la MGI……...…68

Tableau IV.3. Commande conditionnée……...…70

Tableau IV.4. Développement du programme de commande……...……75

Introduction générale

Les avancées technologiques et industrielles en matière d'automatisation sont bien réelles, marquées en particulier par l'essor considérable de la robotique dans la mise en œuvre des procédés de fabrication pour accroitre la sécurité, la qualité, la répétitivité des taches et pour réduire les couts en augmentant la productivité.

Durant des années, les robots industriels se sont concentrés essentiellement sur la réalisation des tâches parfaitement définies au préalable. Ils sont destinés principalement à des utilisations industrielles de manutention et sur les longues lignes d'assemblage, de soudure ou de peinture. Ils sont parfaitement adaptés à des tâches répétitives qu'ils accomplissent avec précision du moment que les conditions de fonctionnement restent les mêmes. La plupart des robots industriels sont programmés par un opérateur expérimenté.

Notre projet à pour but d'introduire la commande de robot ED-7255 de la société sud-coréenne « ED-Corporation ». Ce nouveau dispositif qui est disponible au niveau du Laboratoire d'Automatique de l'Université de Médéa n'a pas fait l'objet de fonctionnement.

Dans cet égard, nous avons choisi d'aborder le domaine de la robotique et de toucher les robots industriels à partir de ce dispositif qui est identique à celui de l'industrie, et voir l'impact des études théoriques sur le domaine de la pratique pour mettre les point sur les i ,en appuyant sur les compétences professionnels , par la mise en œuvre sur une mini-usine équipé d'un automate programmable industriel , qui sera par la suite le cerveau gérant d'une cellule de production robotisé ,contrôlée et commandée à travers une interface Homme-machine , une synchronisation sera faite entre ces différent dispositifs qui prennent un aspect concourant industriel .

Tout au long de notre travail, on a été amené à utiliser plusieurs outils matériels et logiciels: Le robot ED-7255 muni de son contrôleur et sa console de commande, ainsi, son spécifique simulateur ED-IRS, on l'a connecté à une mini-usine contrôlé par un API programmé par le logiciel GMWIN , on a utilisé aussi une interface Homme-Machine qui sera implémenté sur une interface tactile TouchScreen.

Dans le cadre de la présente étude, Ce mémoire s'articule autour de quatre chapitres complémentaires. Premier chapitre, il y aura une généralité sur la robotique en donnant tous ce qu'on aura besoin comme étude théorique de base le long de ce projet.

Ensuite, dans le deuxième chapitre une description du robot ED-7255 est donnée , ainsi que les indications nécessaires pour le démarrer et le manipuler ainsi que sa modélisation géométrique direct et inverse. Par la suite, il y aura une brève description du logiciel de simulation ED-IRS.

Dans le troisième chapitre on donnera une description de la mini-usine ainsi qu'un aperçu sur l'automate programmable industriel et son logiciel de programmation GMWIN. Par la suite, il y aura une présentation de l'écran tactile de l'Interface-Homme –Machine et de son logiciel.

Ces trois derniers chapitres constituent une préparation au projet qui nous a été soumis à la quatrième et dernière partie qui est consacrée pour la réalisation pratique, la mise en ligne des différents systèmes, les comportements de chaque system et les différentes méthodes d'optimisation du procédé avec tous les détails utiles.

A la fin notre mémoire sera achevé par une conclusion générale et des perspectives.

CHAPITRE I

GENERALITES SUR LES ROBOTS

I.1. Introduction

D'après l'Association Française Normalisation (AFNOR), un robot est un « manipulateur commandé en position, reprogrammable, polyvalent, à plusieurs degrés de liberté, capable de manipuler des matériaux, des pièces, est programmé pour l'exécution d'une variété de tâche. Il a souvent l'apparence d'un ou plusieurs bras se terminant par un poignet. Son unité de commande utilise, notamment, un dispositif de mémoire et éventuellement de perception et d'adaptation à l'environnement et aux circonstances. Ces machines polyvalents sont généralement étudiées pour effectuer la même fonction de façon cyclique et peuvent être adaptées à d'autres fonctions sans modification permanente du matériel »

Dans cette définition, on trouve les différentes composantes d'une cellule robotisée :

- le mécanisme : ayant une structure plus ou moins proche de celle du bras humain, il permet de remplacer ou de prolonger son action (le terme 'manipulateur' exclut implicitement les robots mobiles). Sa motorisation est réalisée par des actionneurs électrique, pneumatique ou hydraulique qui transmettent leurs mouvement aux articulations par des systèmes appropriés ;
- la perception, qui permet de gérer les relations entre le robot et son environnement. Les organes de perception sont des capteurs dits proprioceptifs lorsqu'ils mesurent l'état interne du robot (position et vitesses des articulations) et extéroceptifs lorsqu'ils recueillent des informations sur l'environnement (détection de présence, mesure de distance, vision artificielle) ;
- la commande, qui synthétise les consignes des asservissements pilotant les actionneurs. A partir de la fonction de perception et des ordres de l'utilisateur, elle permet d'engendre les actions de robot ;
- l'interface homme-machine à travers laquelle l'utilisateur programme les taches que le robot doit exécuter ;
- le poste de travail et les dispositifs péri robotiques qui constituent l'environnement dans lequel évolue le robot.

La robotique est donc une science pluridisciplinaire des connaissances en mécanique, automatique, électrotechnique, traitement du signal, communication informatique, instrumentation électronique…

Nous donnons dans ce chapitre, quelques définitions nécessaires pour décrire les mécanismes des robots et leurs caractéristiques. Nous discutons ensuite sur les modes de programmations des robots industriels.

I.2. Constituants mécanique des robots

Un robot-manipulateur est constitué par deux sous-ensembles distincts, un (ou plusieurs) organe terminal et une structure mécanique articulée.

-sous le terme organe terminal, on regroupe tout dispositif destiné à manipuler ou transformer des objets (dispositif de serrage, dispositif magnétique, à dépression…)ou à les transformer (outils, torche de soudage, pistolet de peinture…). Il s'agit donc d'une interface permettant au robot d'interagir avec son environnement [1].

Un organe terminal peut être multifonctionnel, c'est-à-dire qu'il est équipé de plusieurs dispositifs ayant des fonctionnalités différentes. Il peut aussi être monofonctionnel mais interchangeable. Un robot, enfin, peut être multi bras, chacun des bras portant un organe terminal différent. Par la suite, nous utilisons indifféremment le terme organe terminal, préhenseur, outil ou effecteur pour nommer le dispositif d'interaction fixé à l'extrémité mobile de la structure mécanique [1] ;

-le rôle de la structure mécanique articulé est d'amener l'organe terminal dans une situation(position et orientation)donnée, selon des caractéristiques de vitesse et d'accélération donnée. Son architecture est une chaine cinématique de corps généralement rigide, ou supposés comme tels, assemblés par des liaisons appelée articulation [1]. Les chaines peuvent être soit ouvertes simples (Figure I.1.), soit fermées (Figure I.2.).

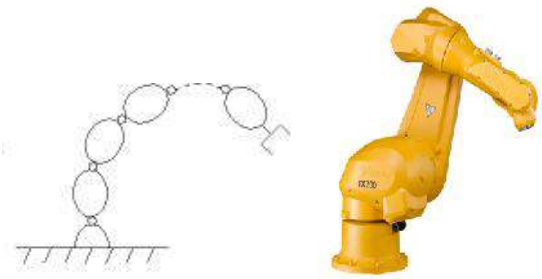


Figure I.1. Chaine simple ouverte.



Figure I.2. Chaine fermée.

Les robots manipulateurs à chaine simples sont les plus nombreux. Les structures contenant des boucles cinématique ont pour avantage essentiel d'augmenter la rigidité ainsi que la précision. Elles permettent en outre un meilleur équilibrage statique. En général, on distingue deux classes de boucles fermées

- Les chaines cinématiques composées ou complexe dont au moins l'un des corps a plus de deux liaisons (figure I.3).

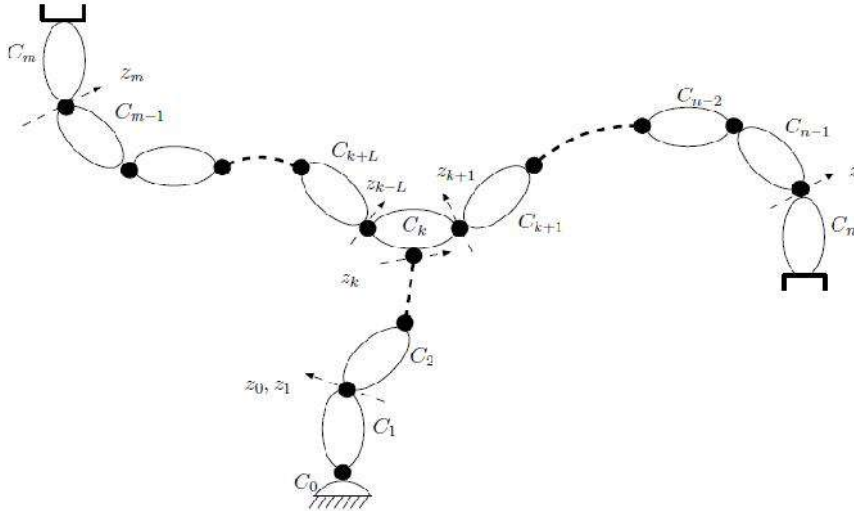


Figure I.3. Robot à structure complexe arborescente

- Des chaines cinématiques élémentaires ou simples qui possèdent des corps qui ont au plus de deux liaisons. L'étude des modèles géométriques des robots à boucles fermées complexe se ramène à l'étude d'une structure ouverte simple et d'une ou plusieurs boucles fermés simples[1].

I.3. Les différents types des robots

I.3.1. Les robots SCARA (Selective Compliance Articulated Robot for Assembly).

La figure I.5 illustre un schéma synoptique d'un robot de type SCARA.

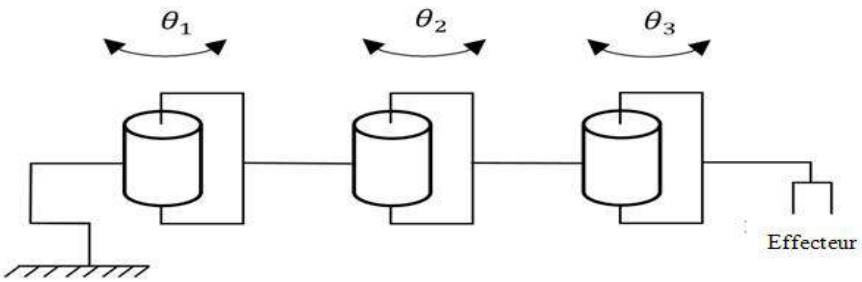


Figure I.5. Schéma synoptique d'un robot SCARA.[2]

I.3.1.1. Caractéristiques :

- 3 axes, série, RRP, 3 DDL.
- Espace de travail cylindrique.
- Précis.
- Très rapide. [2]

I.3.1.2. Exemples :



Figure I.6.Robot DENSO HM



Figure I.7. Robot Adept

I.3.2.Les robots cylindriques

La figure I.8 illustre un schéma synoptique d'un robot cylindrique et un exemple sur ça dans la figure I.9.

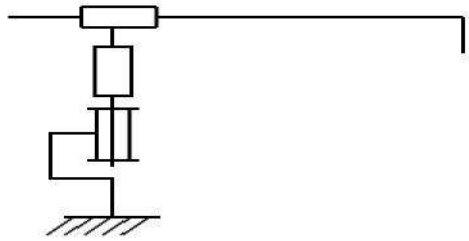


Figure I.8. Le robot cylindrique. [2]

I.3.2.1. Caractéristiques :

- 3 axes, série, RPP, 3 DDL.
- Espace de travail cylindrique.
- Très rapide. [2]

I.3.2.2. Exemples :



Figure I.9. Exemple d'unrobot cylindrique.

I.3.3.Les robots sphériques

La figure I.10 illustre un schéma synoptique d'un robot sphérique et un exemple sur ça dans la figure I.11.

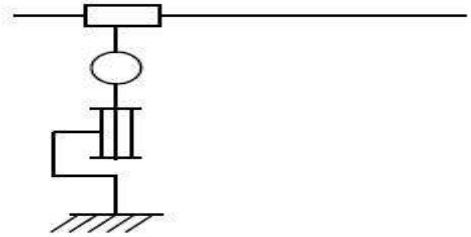


Figure I.10. Robot sphérique

I.3.3.1. Caractéristiques :

- 3 axes, série, RRT, 3 DDL.
- Espace de travail sphérique
- Grande charge utile [2].

I.3.3.2. Exemples :



Figure I.11. Robot FANUC

I.3.4.Les robots Cartésiens

La figure I.12 illustre un schéma synoptique d'un robot cartésien et un exemple sur ça dans la figure I.13.

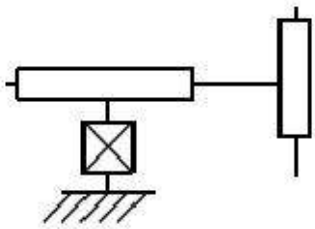


Figure I.12. Robot Cartésien. [2]

I.3.4.1. Caractéristiques :

- 3 axes ⊥2 à 2, série, PPP, 3 DDL.
- Très bonne précision.
- Lent.

I.3.4.2. Exemples :



Figure I.13. Robot Toshiba [2].

I.3.5. Les robots parallèles

I.3.5.1. Caractéristiques :

- Plusieurs chaînes cinématiques en parallèle.
- Espace de travail réduit
- Précis (grande rigidité de la structure).
- Rapide [2]

I.3.5.2. Exemples :



Figure I.14. Robot ABB.

I.3.6.Les robots anthropomorphe

La figure I.15 illustre un schéma synoptique d'un robot anthropomorphe en donnant par la suite des exemples de deux types d'architectures.

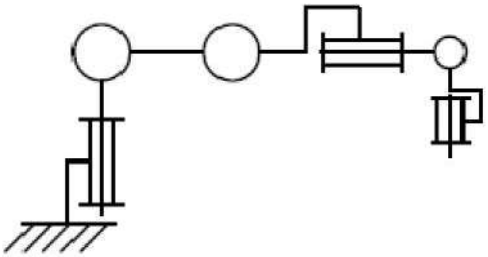


Figure I.15. Robot anthropomorphe [2]

I.3.6.1. Caractéristiques :

- Reproduisent la structure d'un bras humain.
- 6 axes, série, 6R, 6 DDL. [2]

I.3.6.2. Exemples :

- Architecture standard :

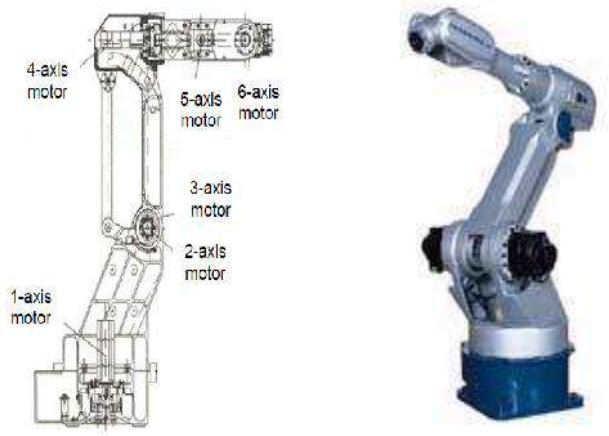


Figure I.16. Exemple d'un bras robot.

- Architecture à parallélogramme:



Figure I.17. Robot ABB

I.4. Choix du nombre de degrés de liberté d'un robot

Le robot non redondant doit disposer six degrés de liberté pour positionner et orienter un solide quelconque dans l'espace. Toutefois, si ce solide présente une symétrie de révolution, cinq degrés de liberté suffisent puisqu'il n'est pas nécessaire de spécifier la rotation autour de l'axe de révolution. [1]

De même pour situer un corps dans un plan, il ne faut que trois degrés de liberté : deux fixent les coordonnées d'un point du corps dans le plant et le troisième détermine son orientation dans ce plant (voir figure I.18).

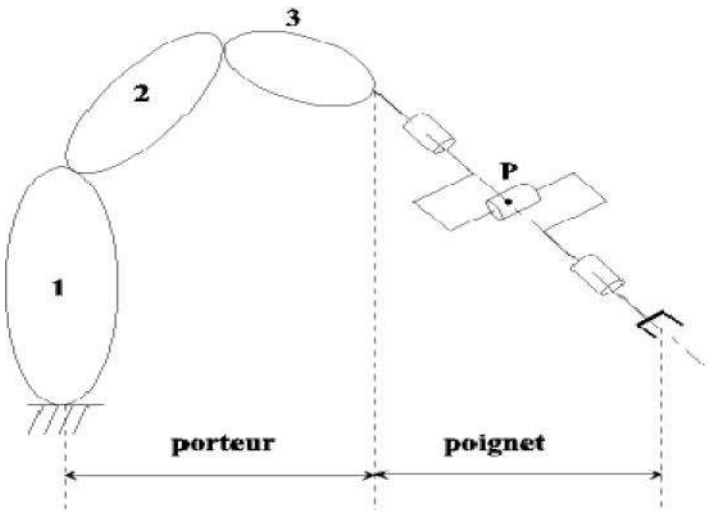


Figure I.18. Structure d'un robot non redondant.

A partir de ces constatations, on déduit que :

-Les caractéristiques des solides manipulés par le robot, donc la classe de tâche à réaliser, permettent de déterminer le nombre de degrés de liberté dont il doit disposer ;

-une condition nécessaire mais non suffisante pour qu'il y ait compatibilité entre le robot et la tâche est que le nombre de degrés de liberté de l'organe terminal de robot soit supérieure ou égale à celui de la tâche: le mécanisme peut alors placer l'organe terminal dans la situation désirer.

Tableau I.1.Nombre de morphologies possibles en fonction du nombre de degrés de liberté de robot. [1]

Nombre de degrés de liberté de robot	Nombre de structure
2	8
3	36
4	168
5	776
6	3508

I.5. Caractéristiques d'un robot

La norme ISO 9946 spécifie la caractéristique que les fabricants de robots doivent fournir. Parmi ces caractéristiques, on peut en mentionner quelques-unes qui permettent de choisir un robot en fonction de l'application envisagée.

- L'espace de travail : il s'agit de l'ensemble de situations de l'espace que l'organe terminal du robot peut atteindre. Il est défini par ses limites, imposées essentiellement par le nombre de degrés de liberté, les débattements articulaires et par la longueur des segments du manipulateur ;
- Les vitesses et accélérations maximales, qui conditionnent le temps de cycle ;
- Les performances (dont les critères sont précisées dans la norme ISO 9283) avec, les notions exactitude (écart entre une situation commandée et la moyenne des situations atteintes) et de répétabilité (dispersion des situations atteintes lorsque l'on commande successivement la même situation) ;

- La charge utile ou charge maximale transportable par le robot sans dégrader la répétabilité et les performances dynamiques. La charge utile est netement inférieure à la charge maximale que peut porter le robot.
- La résolution : c'est la plus petite modification de la configuration du robot à la fois observable et contrôlable par le système de commande. [1]

Cependant, pour une tâche donnée, d'autres caractéristiques doivent être prises en compte : techniques (énergie, commande, programmation…) et commerciales (coût, maintenance…). Les critères de choix sont donc parfois difficiles à formuler et souvent contradictoires, ce qui ne facilite pas la synthèse dont la technique moderne de la conception Assistée par ordinateur (CAO), grâce aux utiles de simulation et d'analyse qu'elles proposent, permettent dans une certaine mesure d'aider à faire un choix et de le valider. [1]

I.6. Génération de mouvement des robots

Le problème de la génération de mouvement est de calculer les consignes de référence en position, vitesse et accélération qui sont en fonction du temps et qui assurent le passage du robot par une trajectoire désirée, définie par une suite de situations de l'organe terminal ou de configurations articulaires (abusivement appelées ***points).***

On peut distinguer les classes de mouvement suivantes :

- le mouvement entre deux points avec trajectoire libre entre les points ;
- le mouvement entre deux points ***via*** des points intermédiaires, spécifiésnotamment pour éviter les obstacles, avec trajectoire libre entre les points intermédiaires.
- le mouvement entre deux points avec trajectoire contrainte entre les points (trajectoire rectiligne par exemple).
- le mouvement entre deux points ***via*** des points intermédiaires avec trajectoire contrainte entre les points intermédiaires [1].

Dans les deux premiers cas, la génération de mouvement peut se faire directement dans l'espace articulaire. Dans les deux derniers, la trajectoire étant décrite dans l'espace opérationnel, il est préférable de raisonner dans cet espace.

I.6.1. Espace articulaire

Espace articulaire d'un robot est celui dans lequel est représentée la situation de tous ses corps. La solution la plus simple consiste à utiliser les variables ou les coordonnées articulaires. L'espace de ces variables, est appelé aussi espace des configurations. Sa dimension N est égale au nombre de variables articulateurs indépendantes et correspond au nombre de degrés de liberté de la structure mécanique. Dans une structure ouverte (simple ou arborescente), les variables articulateurs sont généralement indépendantes, tandis qu'une structure fermée impose nécessairement des relations entre ces variables. [1]

I.6.2. Espace opérationnel

L'espace opérationnel représente la situation de l'organe terminal (on considère donc d'autant d'espaces opérationnels qu'il ya d'organes terminaux). La méthode la plus simple consiste à utiliser les coordonnées cartésiennes.

La dimension de l'espace opérationnel constitue le nombre de degrés de liberté maximum que peut avoir l'organe terminal et est égale au nombre de paramètres indépendants nécessaire pour décrire la situation de l'organe terminal dans l'espace. Dans l'espace tridimensionnel ce nombre est de six(trois pour placer un point du corps en un point quelconque de cet espace et trois pour orienter ce corps de façons quelconque). [1]

I.7. Les différents modes de programmation des robots industriels

Dans le processus continu, le robot est manipulé par une série de commandes et de fonctions stockées dans un fichier dit « programme robot ».Deux modes de programmation des trajectoires des robots sont couramment utilisés [3]:

- La programmation en ligne: PEL
- La programmation hors-ligne:PHL

I.7.1. Programmation en ligne

La programmation en ligne est aujourd'hui le mode de programmation la plus utilisée industriellement. Il en existe deux méthodes:L'apprentissage direct et l'appretissage indirect point à point.

I.7.1.1. L’apprentissage direct

Le principe de cette méthode de programmation consiste à mémoriser. Lors de l’apprentissage, l’ensemble des configurations articulaires du robot pendant l’exécution de la tâche ou pendant une simulation de celle-ci. L’opérateur manipule directement le robot à l’aide d’un pupitre d’apprentissage pour lui « apprendre » la tâcheà accomplir. L’acquisition de la trajectoire est continue. Ce type de programmation est simple et demande peu de formation pour le programme. Toutefois l’utilisation de cette méthode d’apprentissage est plutôt utilisée pour les applications qui exigent des mouvements de précision limitée ,ex : dans la projection sur les pièces simples ou le domaine de la peinture. [3]

I.7.1.2. L’apprentissage indirect point par point

Le principe utilisé par cette méthode est également basé sur une démonstration matérielle de la tâche à réaliser. L’opérateur déplace le robot en mode manuel, utilisant pour cela le pupitre de commande à touches. Cependant, la configuration du robot n’enregistrée que certains points caractéristiques de la trajectoire. La liaison entre ces différents points est spécifiée par les caractéristiques de la trajectoire entre ces points (linéaire, articulaire voire circulaire) et par la définition d’une vitesse de déplacement spécifié par l’opérateur. [3]

I.7.2.Programmation hors-ligne

La programmation hors-ligne permet la génération de programme avec une précision beaucoup plus importante au niveau des points de passage de l’outil (distance de projection, contrôle de l’orientation de l’outil). Le temps consacré à la programmation sur site est fortement réduit. Le positionnement des trajectoires ainsi que les conceptions des outils peuvent être optimisés.

Un des principaux intérêts de ce type de méthode est qu’il ne nécessite aucune immobilisation de l’outil de production durant la phase de programmation du robot. On distingue deux approches principales pour la programmation hors-ligne des robots: La programmation par langage et la programmation graphique à partir d’un système de CFAO (Conception de Fabrication Assistée par Ordinateur). [3]

I.7.2.1. Programmation par langage robotique

Les langages de programmation de robots sont proches des langages informatiques classiques. Ils proposent des instructions pour commander les mouvements des robots. Ils sont très performants pour définir des positions par calculs ou pour implanter des instructions conditionnelles ou des boucles.

Le principal inconvénient de ces langages est qu'ils restent généralement spécifiques à chaque marque de robot: il n'existe pas de standard reconnu. Cela oblige l'opérateur à connaitre langage pour communiquer avec le robot. Les langages suivants peuvent être cités: RAPID (ABB): VAL II (UNIMATION), RAIL (AUTOMATIX), PLAW (KOMATSU) et ED-IRS de notre robot (ED-7255) La figure I.19. montre un outil d'éditeur du programme robot chez ABB . [3]

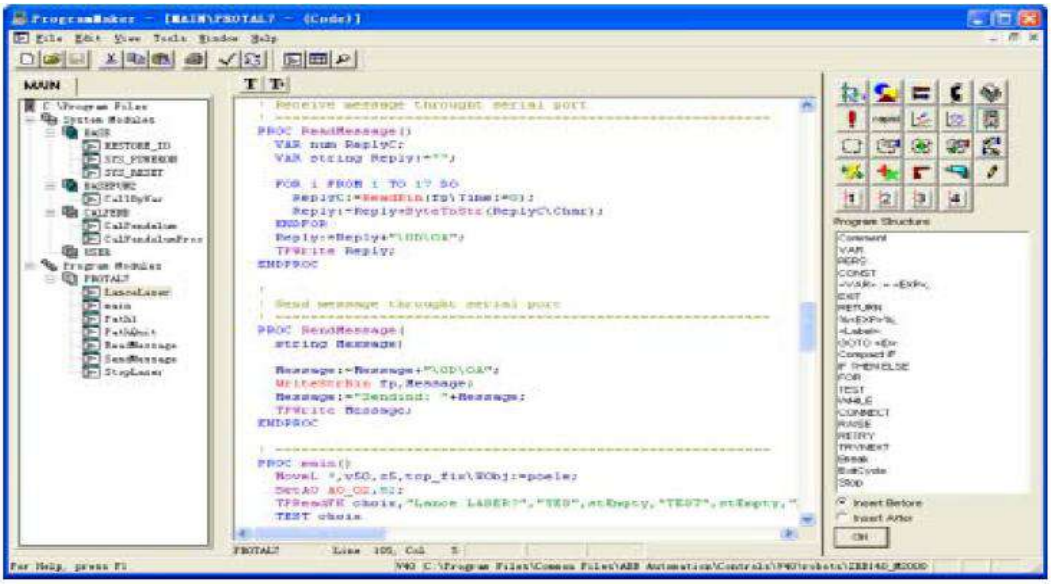


Figure I.19. Programmation par langage robotique-RAPID (Robots ABB).[3]

Ce type de programmation nécessite de connaitre à l'avance les points de passage, l'orientation de l'outil ainsi que la vitesse locale de déplacement pour chaque point. En revanche, cette solution ne peut être retenue pour les géométries complexes de par le nombre de points, les orientations et les vitesses à définir.

I.7.2.2. Programmation graphique

Cette méthode de programmation est une méthode plus pratique pour programmer des trajectoires robot sur les pièces de formes complexes. Les logicielles de CFAO actuels

permettent de représenter et de modéliser les robots, les pièces à revêtir et l'environnement de projection. La figure I.20 présente la programmation hors-ligne à l'aide d'un logiciel CFAO robotique. [3]

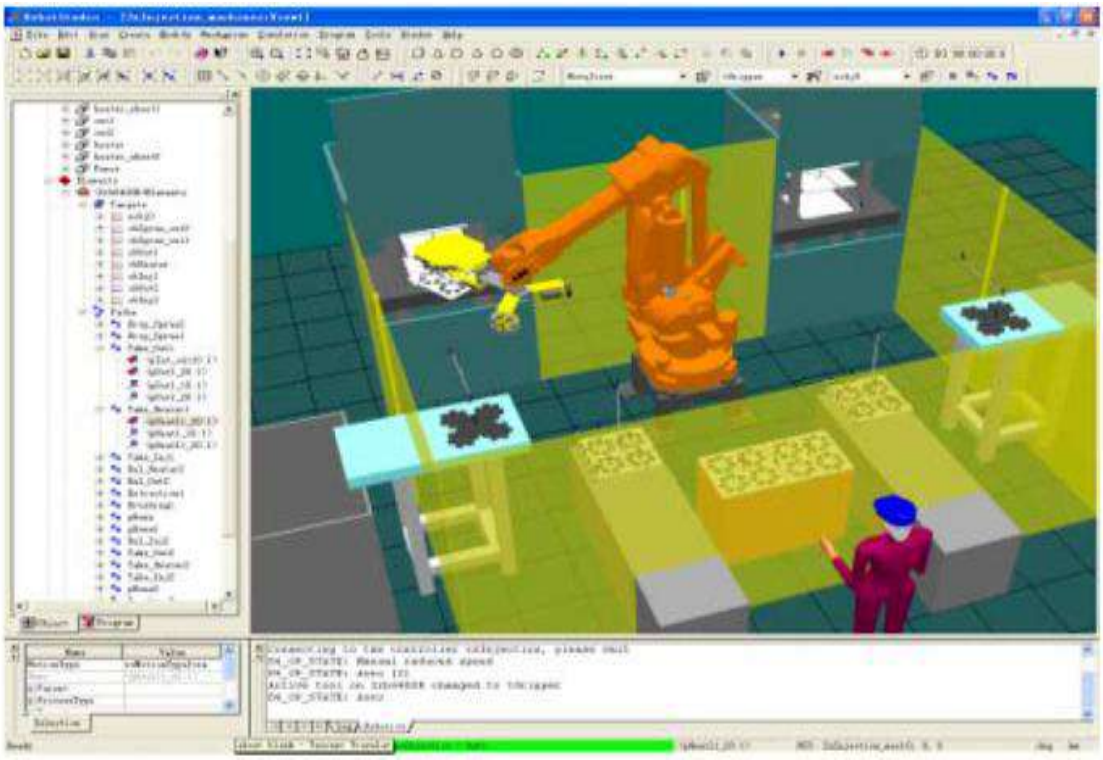


Figure I.20. Programmation graphique.[3]

Cette approche de la programmation hors-ligne présente des avantages indéniables par rapport à l'approche par langage:

-Les méthodes de visualisation en 3D utilisé par le système CFAO permettent à l'opérateur d'apprécier le travail en cours.

-L'interactivité et la convivialité de ces systèmes permettent des modifications rapides des paramètres géométriques.

-La puissance offerte par les systèmes informatiques actuels permet de résoudre certains problèmes de création de trajectoire et de détection des collisions. [3]

Les logiciels de CFAO mécanique proposent de nombreuses fonctions pour la création d'entités géométriques complexes. Ils possèdent souvent des modules dédiés à certaines applications (usinage, calculs de structures, éléments finis, etc.) et parfois même des modules spécialisés en robotique. Par ailleurs, les logiciels spécialisés en robotique possèdent toutes

les fonctionnalités requises en termes de programmation hors-ligne. En revanche, les modèles géométriques dont ils disposent sont souvent très simplifiés par rapport à ceux proposé par les logiciels de CFAO.

L'absence de standard au niveau de langages de programmation ne permet pas aux différents modules de piloter directement les robots. Les modules robotique permettent la génération de trajectoires, mais la liaison avec le contrôleur robot est confiée à des post-processeurs qui créent des programmes dans les langages des robots par discrétisation de la trajectoire. [3]

Le concept de programmation hors ligne peut se synthétiser sous la forme du schéma présenté dans la figure I.21.

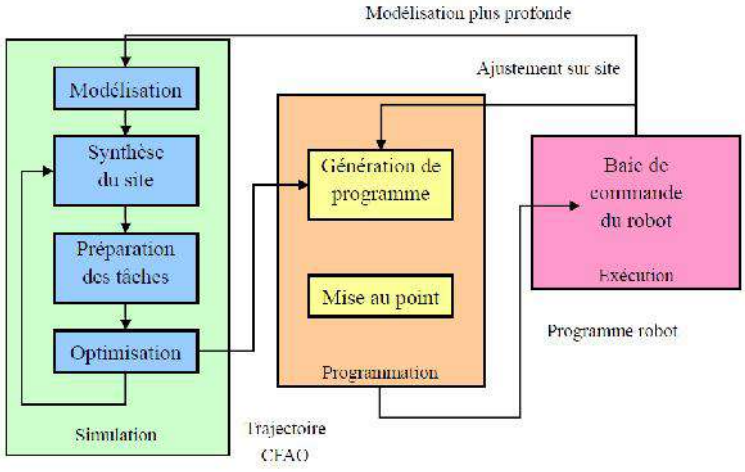


Figure I.21. Démarche de conception et de programmation d'un site robotisée.[3]

I.8. Domaine d'utilisation des robots

I.8.1. Domaine industriel

- L'industrie automobile constitue l'une des principales utilisatrices de robots industriels. Comme s'est montrée sur la figure I.22 et la figure I.23, les robots participent à l'assemblage des automobiles dans le cadre de la fabrication en série. [4] [5]



Figure I.22. Chaîne d'assemblage



Figure I.23. Robot soudeur a l'arc

- Chaine de production
  - ✓ Pour des tâches simples : une grande majorité des robots est utilisée pour des tâches simples et répétitives (figure.24).



Figure I.24.Manipulateur rapide dans le secteur Agro-alimentaire.

- ✓ Pour des tâches complexes : dans des conditions pénible et dangereuse. [4]



Figure I.25. Manipulateur pour des opérations pénible et dangereuse

I.8.2.Dmaine d'exploration



Figure I.26. Robot canadarm2.

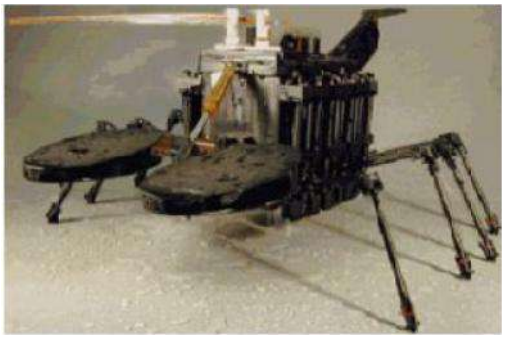


Figure I.27.Robot sous-marin. [4]

I.8.3.Domaine de Sécurité, Militaire



Figure I.28. Drone militaire.

I.8.4. Domaine de service



Figure I.29. Aspirateur [5]



Figure I.30. Robot humanoide.

I.8.5. Domaine médical



Figure I.31. Assistance aux opérations chirurgicales

I.9. Les exigences de sécurité

La prise en compte de la sécurité est absolument incontournable en robotique industrielle. La sécurité est régie par des normes très strictes, zone protégée par enceinte de protection où l'accès restreint.

Son objectif est de protéger l'utilisateur du danger potentiel du robot en mouvement. La majorité des accidents en milieu industriel se produisent pendant le démarrage, la mise au point, la programmation de l'installation (plus rarement pendant l'installation, la maintenance et la production).Chaque utilisateur doit être formé aux risques et aux consignes de sécurité. [6]

I.10. Conclusion

Le présent chapitre fait l'objet d'une description sur le domaine de la robotique en mettant l'accent sur les domaines applications des robots industriels. Ensuite la classification des robots est abordée. Une description détaillée du bras manipulateur didacticiel *ED-7255* fera l'objet du chapitre suivant.

CHAPITRE II

PRESENTATION ET MODELISATION DE ROBOT ED-7255

II.1. Introduction

Le robot ED-7255 de la société sud-coréenne ED-Corporation, est utilisé dans notre projet. Ce robot est disponible ai niveau de Hall de technologie de l'université de Médéa.

Ce produit est un robot pour un usage éducatif fait pour l'expérience et la pratique d'un robot série vertical à multi-articulation (multi-joint), qui estlargement utilisé dans les sites industriels, il comprend tous les éléments d'un robot industriel , sont illustrés par la figure II.1 ,Dont il inclut d'un bras robotisé, un contrôleur, une pince (Gripper), console de commande, logiciel pour la programmation software. On va donner dans ce chapitre une description matériel à chacun de ces éléments.

Grâce à l'énergie et aux différents systèmes de transmission (poulies courroies et engrenages), le robot ED-7255 pourra accomplir différents mouvements.

En effet, le robot ED-7255 est un robot ayant 5 DDL. Chaque articulation du robot consiste un degré de liberté. L'exécution d'une tâche exige les 5 DDL, en effet la masse portée devra pouvoir se déplacer partout, autrement dit dans l'espace de travail.

La commande de robot nécessite le calcul de certain modèle mathématique tel que la modélisation géométrique du robot qui sera établis.

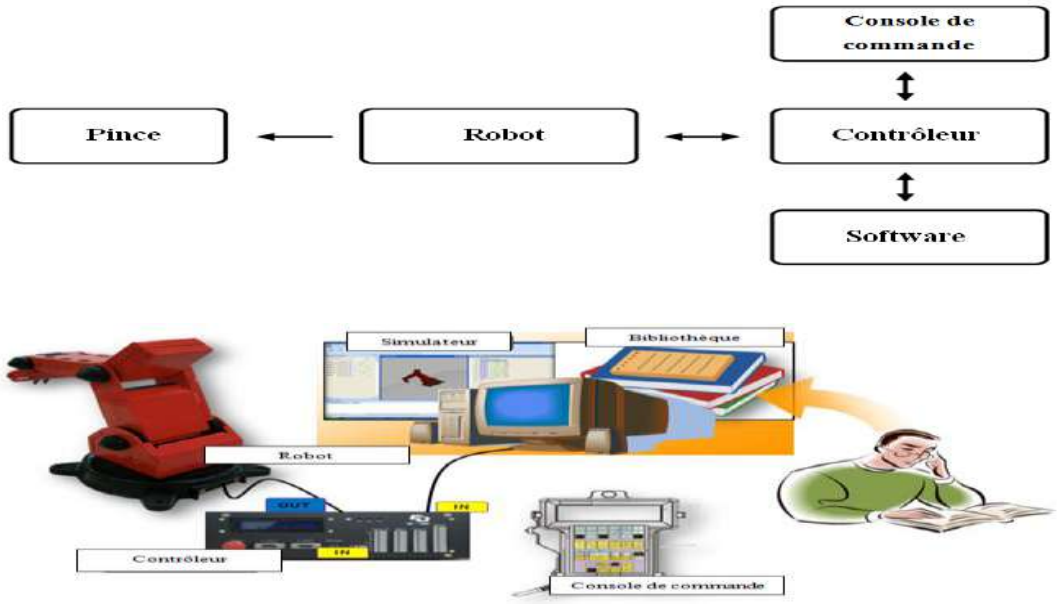


Figure II.1 Eléments fournis avec le robot ED-7255.

II.2. Composants et caractéristiques techniques du produit

II.2.1. Le Bras articulé

Ce robot dispose de 5 axes (La figure. II.2) pour une base, épaule, coude, terrain de poignet, et roulis du poignet, chacun d'euxest entraîné par un moteurs à courant.



Figure. II.2. Les cinq axes du robot.

L'effecteur : Le robot ED-7255 est doté par une pince de deux doits qui permet la manutention des objet sachant qu'elle est entrainé aussi par un moteur à courant continu.[7]

Comme s'est montrées sur la figure II.3, un pignon de 15mm est monté à l'arbre du moteur et un autre à l'encodeur .Le mouvement rotationnel du pignons est transféré à accumuler l'engrenages et il est converti en mouvement rectiligne afin d'ouvrir et fermer la pince.

Lorsque la pince est ouverte , il est de 84mm , et lorsque elle est fermées, il est de 0 mm.

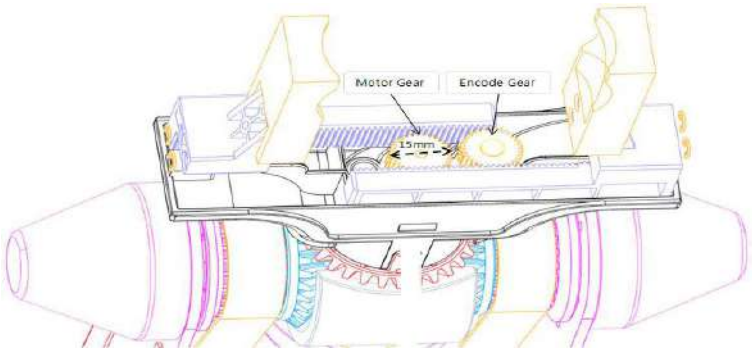


Figure. II. 3. Dessin correspont à l'effecteur.

Les six moteurs à courant continu ont respectivement un engrenage réducteur et un codeur absolu (voir la p.36)qui est attaché à chaque axe. Le réducteur sert à réduire la vitesse de rotation du moteur.[7]

II.2.1.1. Point de vue mécanique

Les caractéristiques mécaniques du robot ED-7255 sont données dans le tableau II.1.

Tableau II.1.Caractéristiques mécaniques de robot ED-7255. [7]

Sujet	Description
construction	Bras vertical articulé
Nombre d'articulation	5 axes de rotation + 1 pince
Charge utile	1.0kg
Intervalle de Mouvement	Axe1 (Rotation de base) : ±170° Axe2(Rotation de l'épaule) : -90/+30° Axe3(Rotation de coude) : 0/-135° Axe4(poignet Pitch) : -110/+90° Axe5(poignetRoll) : ±160°
Longueur du robot	max. 802mm
Rayon de fonctionnement	Max 680 mm
La taille physique	280*280*803(H)mm
La vitesse maximal	750mm/sec (32.5° /s)
Max ouverture de la pince	80mm
Precision (position-rotation)	±0.5mm -±1.0°
Capteur de position	Codeur absolu
Actionneurs	moteur DC 24V
La masse	9.5Kg
Températures d'utilisation	2°C - 40°C

Une vue sur le plan mécanique avec des mesures dimensionnelles du robotsont illustréespar la Figure II.4.

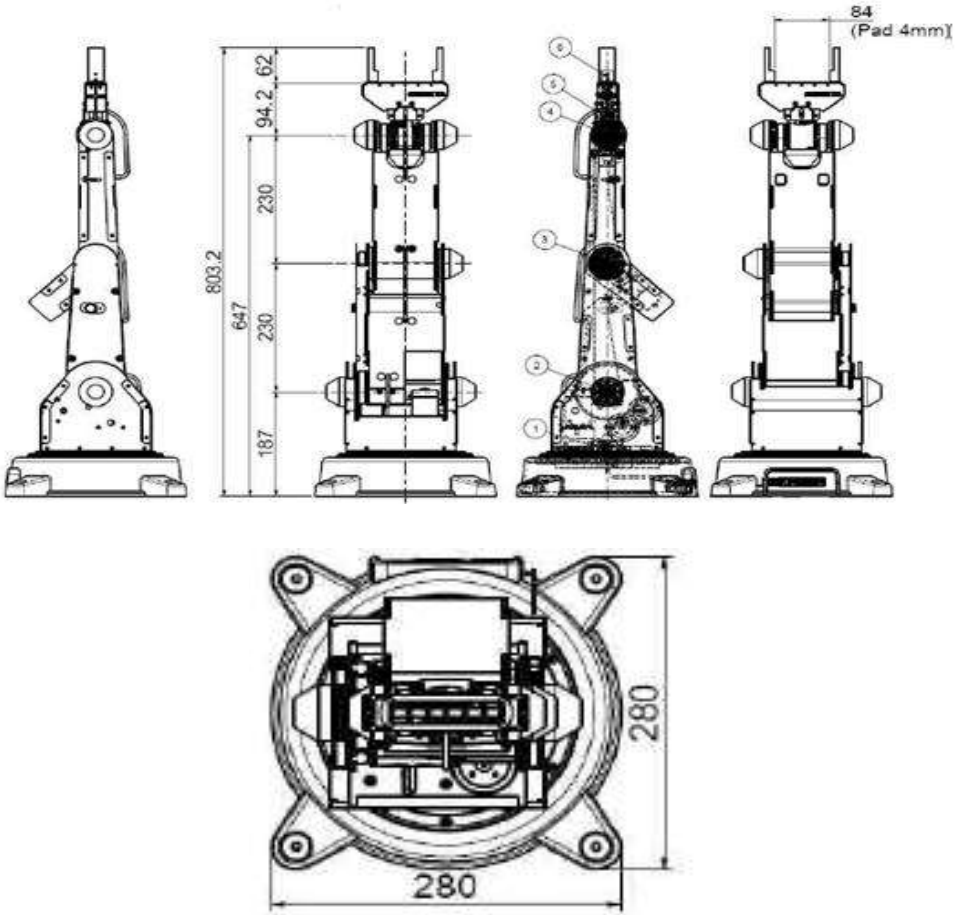


Figure. II.4. Schéma dimensionnel du Robot ED-7255.[7]

Esapce du travail de ED-7255

La figure II.5 illustre l'espace de travail du robot verticale articulé.

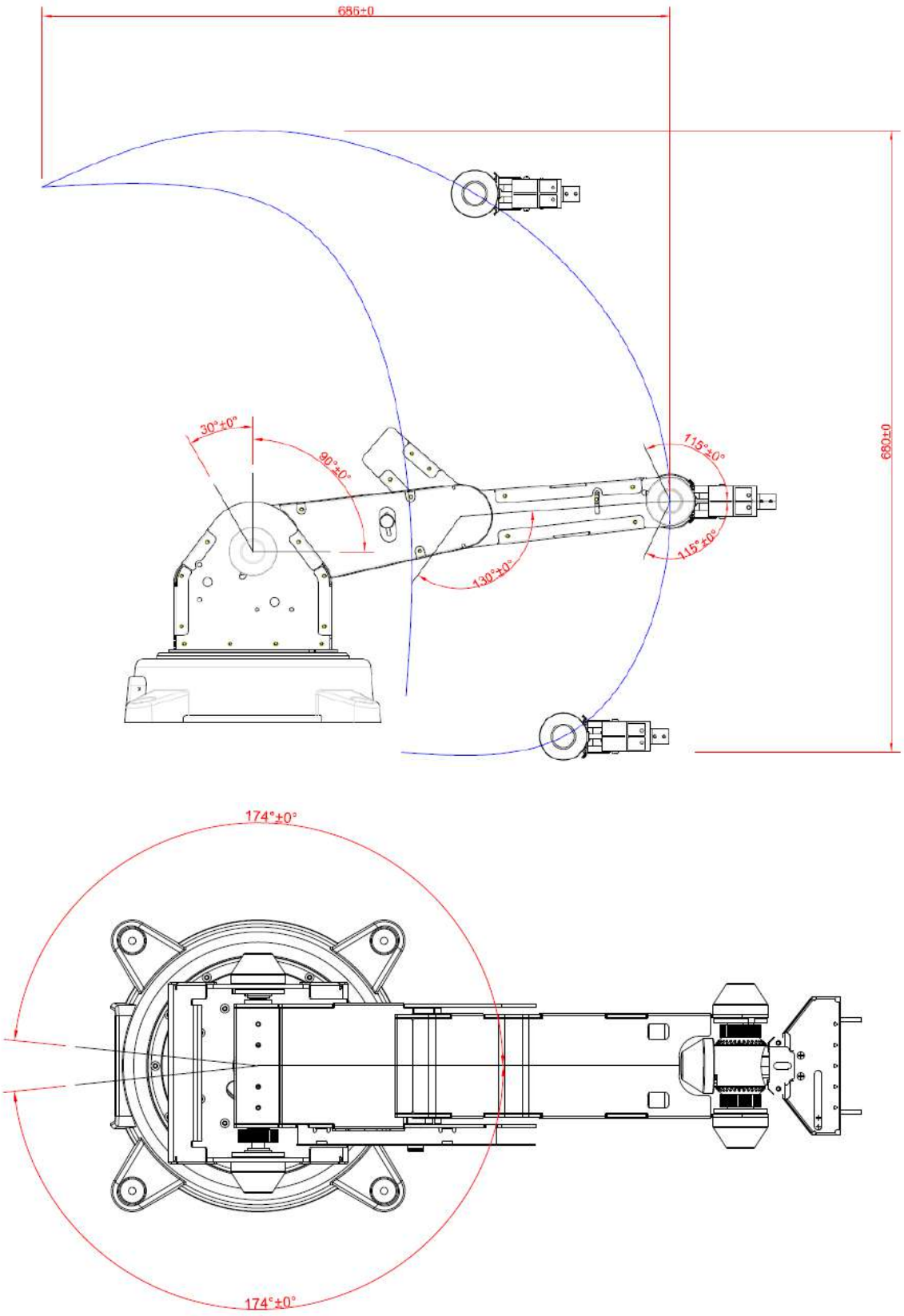


Figure 1I.5. Espace de travail du robot. [7]

II.2.1.2. Modélisation des bras de robots

Dans tous les cas où les robots manipulateurs ne sont pas directement télé-opérés, mais ils possèdent une certaine autonomie d'action, leurs systèmes de commande doivent connaitre et compenser les imprécisions éventuelles puisque l'opérateur humain n'est pas inclus directement dans la boucle d'asservissement, pour s'adapter aux caractéristiques des machines et de leurs environnement. Pour cela, il est nécessaire de passer par une modélisation mathématique précise de la géométrie, et de la cinématique des bras manipulateurs. [8],[1]

II.2.1.2.1. Modèle géométrique direct

Le modèle géométrique direct (MGD) est l'ensemble des relations qui permettent d'exprimer la situation de l'organe terminal, c'est-à-dire les coordonnées opérationnelles du robot, en fonction de ses coordonnées articulaires. Dans le cas d'une chaîne ouverte simple, il peut être représenté par la matrice de transformation $^{\circ}T_n$:

$$^{\circ}T_n = {}^{\circ}T_1 . {}^{1}T_2 \dots\dots\dots\dots\dots\dots {}^{n-1}T_n$$

Le MGD exprime la position et l'orientation du repère de référence R lié a l'outil, relativementa un repère fixe celui de la base R0 par exemple en fonction des variables articulaires motorise (asservissement électroniquement) $\theta_1, \theta_2 \dots\dots \theta_n$ du mécanisme. [8]

L'élément fondamental qui décrit la position et l'orientation d'un repèreR_i**,** d'origineO_i, dans un repèreR_j peut se mettre sous la forme d'une matrice de dimensions **(4**X**4)** :

$$^{\circ}T_n = \begin{bmatrix} S_X & n_X & a_X & P_X \\ S_y & n_y & a_y & P_y \\ S_Z & n_Z & a_Z & P_Z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad {}^{i}T_n = \begin{bmatrix} & {}^{i}\mathcal{A}_j & & {}^{i}\mathcal{P}_j \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Avec :

(S_n, n_n, a_n**) :** Désignent les vecteurs unitaire de la matrice $^{i}\mathcal{A}_j$ de rotation.

$^{i}\mathcal{P}_j$**:** désigne le vecteur de translation.

Modèle géométrique directe du robot a 5 degrés de liberté : (ED-7255)

Obtenir le modèle géométrique direct consiste a déterminer les relations de la forme suivante :

$$x = f\ (\theta)$$

Permettant de calculer X, le vecteur des coordonnées choisies pour représenter

la position du corps C5, organe terminal, par rapport au repère universel R_0 , a partir

de la connaissance de θ, le vecteur des coordonnées articulaires.

En réalisant la composition des transformations du repère universel R_0 jusqu'au repère R5, sont partitionnées en deux ensemble de matrices[9],une qui donne la position et l'orientation du porteur :

$$T_0^3 = T_0^1 T_1^1 T_2^3$$

Et l'autre qui donne la position et l'orientation de l'organe terminal

$$T_3^5 = T_3^4 T_4^5$$

On obtient :

$$T_{Base}^{Effe}(q) = T_{Base}^{Epaule} T_{Epaule}^{Coude} T_{Coude}^{Ter} T_{Ter}^{Roll} T_{Roll}^{Effe}$$

$$T_0^5 = T_0^1 T_1^2 T_2^3 T_3^4 T_4^5$$

$$T_0^5 = [T_0^1 T_1^2 T_2^3][T_3^4 T_4^5]$$

Il est usuel de représenter les matrices de transformations qui font passer d'un corps au suivant, au moyen du formalisme de Danavit et Hartenberg.[9]

Le passage du repère R_{j-1} au repère R_js'exprime en fonction des quatre paramètres géométriques de Danavit–Hartenberg suivant la figure I.5.

• α_j : angle entre les axes $\mathbf{z}_{j-1}$ et $\mathbf{z}_j$ correspondant à une rotation autour de $\mathbf{x}_{j-1}$;

• d_j : distance entre $\mathbf{z}_{j-1}$ et $\mathbf{z}_j$le long de $\mathbf{x}_{j-1}$;

• θ_j : angle entre les axes $\mathbf{x}_{j-1}$ et $\mathbf{x}_j$ correspondant à une rotation autour de $\mathbf{z}_j$;

• r_j: distance entre $\mathbf{x}_{j-1}$ et $\mathbf{x}_j$ le long de $\mathbf{z}_j$.

La Figure. II.6 montre les paramètres intervenant dans la transformation.

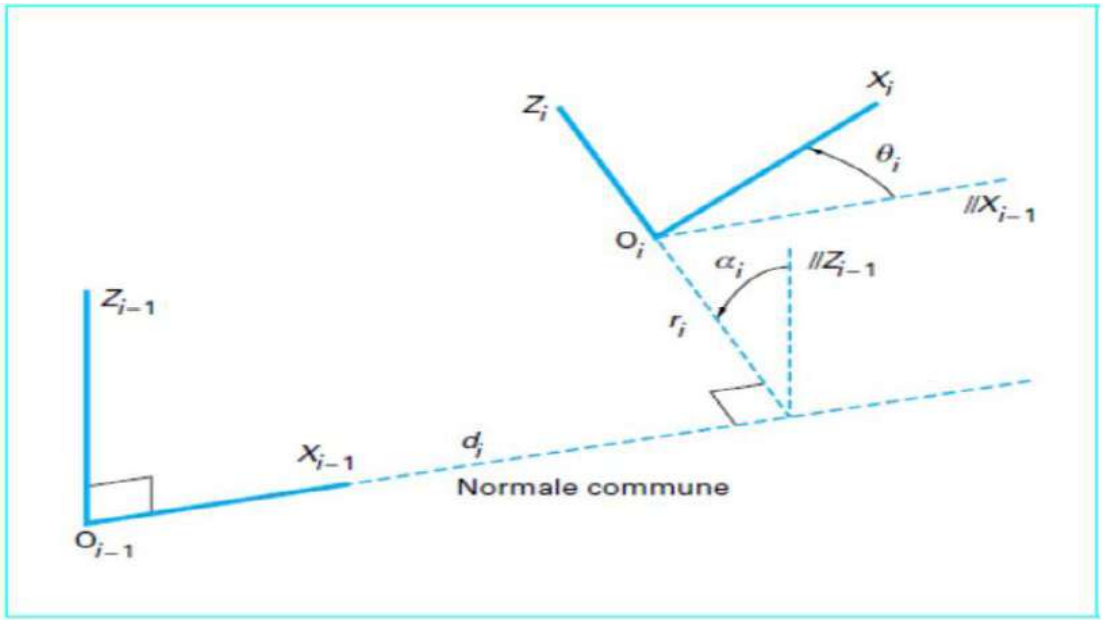


Figure. II.6 Paramètres de Danavit -Hartenberg.[1]

La matrice de transformation définissant le repère R_j dans le repère R_{j-1} est donné par (la figure I.5) : ${}^{j-1}T_j = Rot(x, \alpha j)\ Trans(x, dj)\ Rot(z, \theta j)\ Trans(z, rj)$

$$= \begin{bmatrix} C\theta_j & -S\theta_j & d_j & C\theta_j \\ C\alpha_j S\theta_j & C\alpha_j C\theta_j & -S\alpha_j & -r_j S\alpha_j \\ S\alpha_j S\theta_j & S\alpha_j C\theta_j & C\alpha_j & r_j C\alpha_j \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Compte tenu des conventions de sens de rotation retenues pour chacune des articulations (figure. II.7), on obtient les paramètres géométriques du robot, en fonction des coordonnées articulaires (Tableau. II.2) d'après les placements des repères tout en suivant les paramètres de Danavit -Hartenberg.

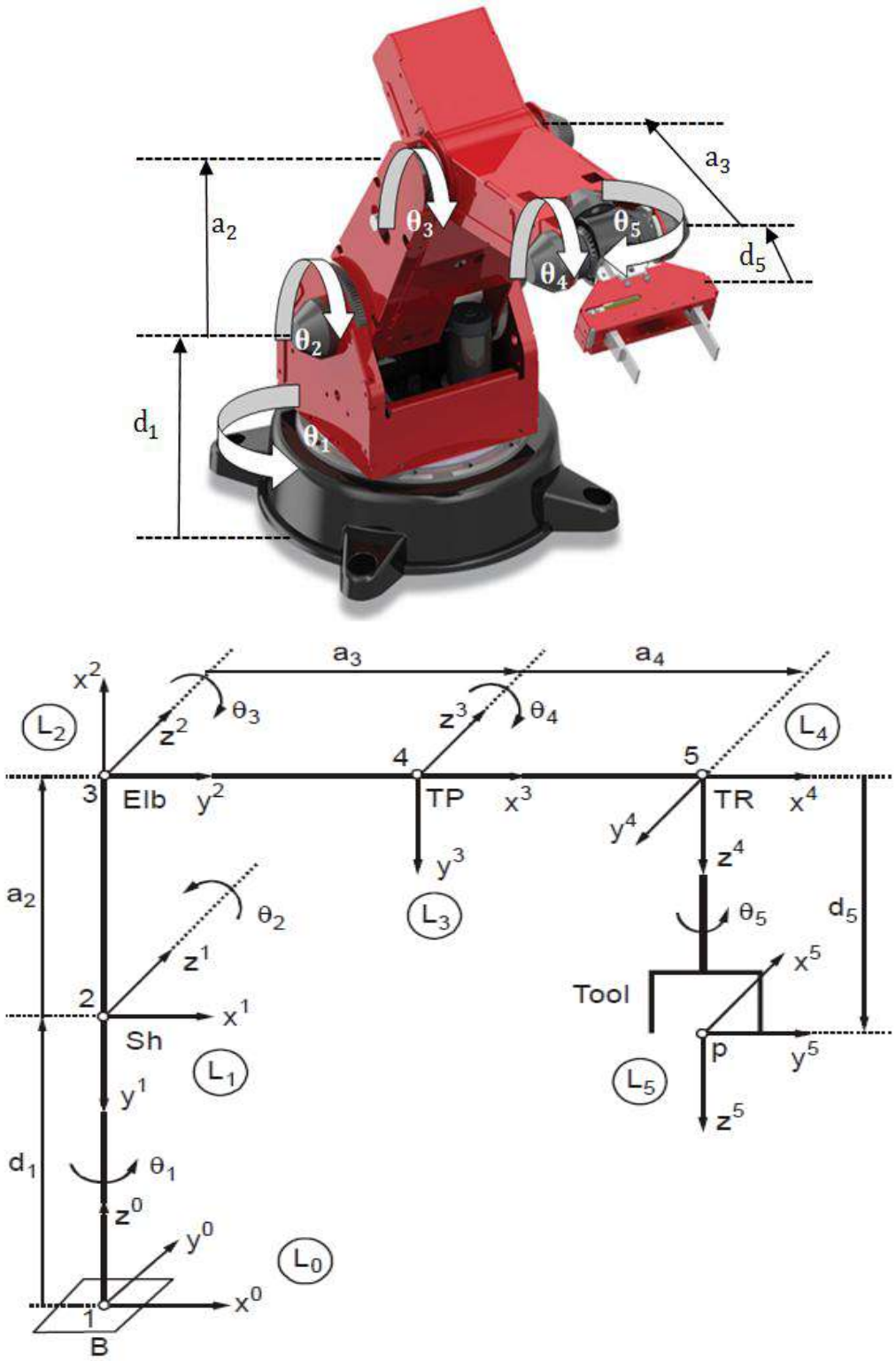


Figure. II.7 Placement des repères et notation pour le robot 5ddl.

Où

Le vecteur des variable articulaire est donné par $q = \{\theta_1, \theta_2, \theta_3, \theta_4, \theta_5\}^T$

Le vecteur des distances articulaire est donnée par

$$d = \{d_1, d_2, d_3, d_4, d_5\}^T = \{18.7,0,0,0,15.4\}$$

Le vecteur des longueur est donné par $a = \{a_1, a_2, a_3, a_4, a_5\}^T = \{0,23,23,0,0\}$

Le vecteur des angles de rotation est donné par

$$\alpha = \{\alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5\}^T = \{(-90),0,0,(-90),0\}$$

Tableau. II .2 Paramètres géométriques du robot

Axes	Type	θ_i	d_i	a_i	α_i
1	**Base**	θ_1	d_1	0	-π/2
2	**Epaule**	θ_2	0	a_2	0
3	**Coude**	θ_3	0	a_3	0
4	**Terrain de poignet**	θ_4	0	a_4	-π/2
5	**Roulis de poignet**	θ_5	d_5	0	0

Les matrices de transformations ;

$$T_0^3 = T_0^1 T_1^2 T_2^3$$

$$= \begin{bmatrix} C_1 & 0 & -S_1 & 0 \\ S_1 & 0 & C_1 & 0 \\ 0 & -1 & 0 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} C_2 & -S_2 & 0 & a_2C_2 \\ S_2 & C_2 & 0 & a_2S_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} C_3 & -S_3 & 0 & a_3C_3 \\ S_3 & C_3 & 0 & a_3S_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} C_1C_2 & -C_1S_2 & -S_1a_1C_1C_2 \\ S_1C_2 & -S_1S_2 & C_1a_1S_1S_2 \\ -S_2 & -C_2 & 0 \quad d_1 - a_2S_2 \\ 0 & 0 & 0 \quad 1 \end{bmatrix} \begin{bmatrix} C_3 & -S_3 & 0 & a_3C_3 \\ S_3 & C_3 & 0 & a_3S_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} C_1(C_2C_3 - S_2S_3) & -C_1(S_2C_3 + C_2S_3) & -S_1C_1\{a_3(C_2C_3 - S_2S_3) + a_2C_2\} \\ S_1(C_2C_3 - S_2S_3) & -S_1(S_2C_3 + C_2S_3) & C_1S_1\{a_3(C_2C_3 - S_2S_3) + a_2C_2\} \\ -(S_2C_3 - C_2S_3) & -(S_2C_3 + C_2S_3) & 1 \quad d_1 - a_2S_2a_3(S_2C_3 - C_2S_3) \\ 0 & 0 & 0 \quad 1 \end{bmatrix}$$

$$T_0^3 = \begin{bmatrix} C_1C_{23} & -C_1S_{23} & -S_1C_1(a_2C_2 + a_3C_{23}) \\ S_1C_{23} & -S_1S_{23} & C_1C_1(a_2C_2 + a_3C_{23}) \\ -S_{23} & -C_{23} & 0 \quad d_1 - a_2S_2 - a_3S_{23} \\ 0 & 0 & 0 \quad 1 \end{bmatrix}$$

$$T_3^5 = T_3^4 T_4^5$$

$$T_0^3 = \begin{bmatrix} C_4 & 0 & -S_4a_4C_4 \\ S_4 & 0 & C_4a_4C_4 \\ 0 & -1 & 0 \quad 0 \\ 0 & 0 & 0 \quad 1 \end{bmatrix} \begin{bmatrix} C_5 & -S_5 & 0 & 0 \\ S_5 & C_5 & 0 & 0 \\ 0 & 0 & 1 & d_5 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} C_4C_5 & -C_4S_5 & -S_4a_4C_4 - d_5S_4 \\ S_4C_5 & -S_4S_5 & C_4a_4S_4 - d_5C_4 \\ -S_5 & -C_5 & 0 \quad 0 \\ 0 & 0 & 0 \quad 1 \end{bmatrix}$$

$$T_0^5 = T_0^3 T_3^5$$

T_0^5

$$= \begin{bmatrix} C_1C_{234}C_5 + S_1S_5 & -C_1C_{234}S_5 + S_1C_5 & -C_1S_{234} & C_1(a_2C_2 + a_3C_{23} + a_4C_{234} + d_5C_{234}) \\ C_1C_{234}C_5 - C_1S_5 & -S_1C_{234}S_5 - C_1C_5 & -S_1S_{234} & S_1(a_2C_2 + a_3C_{23} + a_4C_{234} + d_5S_{234}) \\ -C_5S_{234} & S_{234}S_5 & C_{234} & d_1 - a_2S_2 - a_3S_{23} - a_4S_{234} - d_5S_{234} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

En identifiant le calcul final avec la forme générale de modèle géométrique on obtenir:

$$T_0^5 = \begin{bmatrix} R_{11} & R_{12} & R_{13}P_1 \\ R_{21} & R_{22} & R_{23} \quad P_2 \\ R_{31} & R_{32} & R_{33} \quad P_3 \\ 0 & 0 & 0 \quad 1 \end{bmatrix}$$

Avec:

R11 = C1C234C5 + S1S5 = 0

R21 = S1C234C5 – C1S5 = 1

R31 = – C5S234 = 0

R21 = – C1S5C234 + S1C5 = 1

R22 = – S1S5C234 – C1C5 = 0

R32 = S234S5 = 0

R31 = – C1S234 = 0

R32 = – S1S234 = 0

R33 = – C234 = – 1

p1 = C1 (a2C2 + a3C23 + a4C234 – d5S234) = a3 + a4

p2 = S1 (a2C2 + a3C23 + a4C234 – d5S234) = 0

p3 = d1 – a2S2 – a3S23 – a4S234 – d5C234 = d1 + a2 – d5

II.2.1.2.2. Modèle géométrique inverse

On a vu que le modèle géométrique direct d'un robot permettait 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. Lorsqu'elle existe, la forme explicite qui donne toutes les solutions possibles (il y a rarement unicité de solution) constituent ce que l'on appelle le modèle géométrique inverse (MGI). [1]

En supposant les asservissement parfaits, on a égalité de leurs sorties $\theta_i$et des consignes de position correspondantes (erreur statique nulle).Il s'agit ici de calculer ,si elle existe , l'application réciproque

f_{-1}de celle du modèle géométrique direct.

Aucune solution réelle n'existe en dehors de ce que l'on nomme l'espace de travail du robot. De plus, il peut exister plusieurs solutions en nombre fini ou en nombre infini dans les singularités.

En dehors des singularités, et sans tenir compte des butées mécaniques, il est impossible dans le cas général de savoir si un mécanisme donné est soluble et quel est le nombre de solution quant il l'est. [8]

II.2.1.3. Modèle géométrique inverse du robot a 5 degrés de liberté : (ED-7255)

Le modèle géométrique inverse calcule les coordonnées articulaires a partir de la donnée de la position de l'organe terminal du robot. Ainsi T_0^5 étant connue, on cherche a calculer les orientations θ_i.[10]

Des équations, on tire :

$$\theta = a\tan 2(P_y, P_x) or \theta_1 = \theta_1 + 180$$

$$\theta_2 = a\tan 2(S_2, C_2)$$

$$\theta_3 = a\tan 2(S_3, C_3)$$

$$\theta_4 = \theta_{234} - \theta_3 - \theta_2$$

$$\theta_5 = A\tan 2(S_5, C_5)$$

Point de vue éléctrique

Composants électriques du robot ED-7255 sont les suivants,

• **Moteur DC:** les 5 axes et les doigts de la pince peuvent se déplacer à droite et à gauche, de haut en bas selon le sens de rotation des moteurs à courant continu dont il varie comme étant fourni en tension positive ou négative à partir du contrôleur.

• **Encodeur:**Ce sont des capteurs rotatifs absolus miniatures du modèle MA3 est présentée par la figure II.8, ilsintègrent un convertisseur numérique. Ils codent jusqu'au 12 bits la position angulaire de l'arbre sur 360 ° ainsi que la direction et la vitesse de rotation de chaqu'un des six moteurs, sans arrêts ou des lacunes.

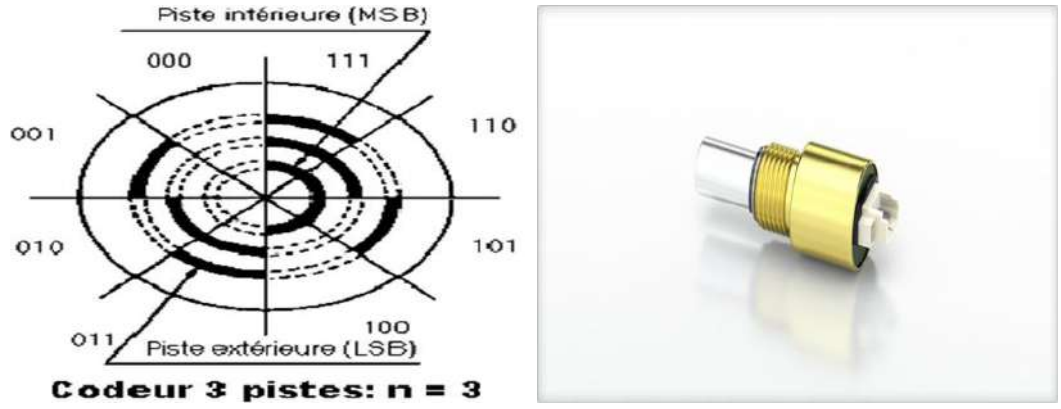


Figure. II.8 Codeur rotatif absolu (MA3).

• **Câble:** Le boîtier de commande et le bras sont reliés par un câble paralléle de moteur à courant continu et un câble parallèle d'encodeur. Le consolee de commande et le boîtier de commande sont connectés avec le câble RS-232 et le PC et le boitier de commande sont reliés avec un câble Ethernet.

II.2.2. Contrôleur (ED-MC1)

Un contrôleur de robot est un élément qui crée des signaux de commande pour faire manipuler le bras de robot .Comme le montre la Figure II.9 et la Figure II.10, il est relié à un ordinateur, un console de commande, et le bras de robot.

Ce contrôleur comporte :

- LED montrant l'état du
- LCD indiquant l'état de fonctionnement.
- Clés d'urgence pour arrêt d'urgence du robot,
- Ports pour le contrôle des moteurs externes,
- ports pour le contrôle des moteurs externes, qui permet être analogiquesou numérique pou interconnections avec les dispositifs externes (capteurs..).

Pour des informations plus détaillées, se reporter aux chiffres et description suivantes.



Figure. II.9 Vue arriere du controlleur ED-MC1 de robot.



Figure. II.10 Vue de face du controleur de robot.

1. **Bouton d'allimentation :** Activer / désactiver la source d'alimentation principale. Une LED s'allimente après le démarrage correcte de la carte principale.
2. **Alimentation principale :** Alimentation pour le contrôleur de source 220V ou 110V est utilisé.
3. **Port moteurs :** Un port parallèle pour alimenter les moteur de chaque articulation.
4. **Port Codeur :** Les valeurs des codeurs de chaque axe de bras de robot sont lus par ce port.
5. **Port Ethernet :** Une interface de communication qui est reliée au PC (simulateur) et attribue le IP pour faire télécharger le programme de commande pour le bras de robot, et pour configurer les paramètres du système du contrôleur de robot.
6. **JTAG Port :** un port de mise à jour du micrologiciel de moteur de contrôle (DSP)à l'intérieur de l'unité de commande.
7. **RS232 Port :** Connecté au PC via un câble série pour la communication série ARM noyau du le contrôleur, et il est utilisé pour la bibliothèque du commande de robot.
8. **Bouton d'urgence** : Utilisé pour l'arret d'urgence des moteurs au cours du fonctionnement du robot.
9. **Bloc d'Entrées/Sorties**: Il fournies de 8 ports numérique (entrée et sortie) pour verrouillage avec des périphériques, 4 ports analogiques (entrée et sortie respectivement) , 4 ports pour relais SPDT, et 2 ports pour source externe 12VDC.
10. **LCD:** Le statut et l'état de fonctionnement actuel du robot est affiché sur un écran LCD .
11. **Connecteur de console de commande:** connecté à un câble du « teach pendant » pour fournir la commande par l'intermédiaire d'une communication série.
12. **Ports de moteur supplémentaires**: il fournit deux ports pour lecontrôle supplémentaire des moteurs.

II.2.2.1. Description du contrôleur de robot

Le tableau II.3 résume les principales caractéristiques du contrôleur

Tableau II.3. Caractéristique de contrôleur d'ED-7255.

Sujet	Description
Type de commande	Temps-Réel , PWM ,PID
Processeur	ARM (6410), DSP(2812) , Teaching pendant 16 bits AVR
Nombre d'axes controlé	6 axes+2 axes (periphérique)
Type des trajectoirs contrôlées	Articulaire,linéaire
Source d'alimentation	110/220 VAC, 50/60Hz, 160W max.
Communication	Ethernet
Interface	Port série RS-232C, Port Ethernet , Port I/O, Port série ARM
Entrée externe	8 entrées digitales :24V 4 entrées analogiques :0-12V (bit de résolution)
Sortie externe	8 sorties digitales :24V /20 Ma 4 sorties analogiques :0-12V / 10mA 4 Relais : SPDT/250VAC-2A/30V-2A
Source externe	+12V DC

L'architecture hardware du contrôleur de robot qui effectue une série de tels processus est montrée dans la figure. II.11, et les composants internes de la carte de commande est représentée dans la Figure II-12.

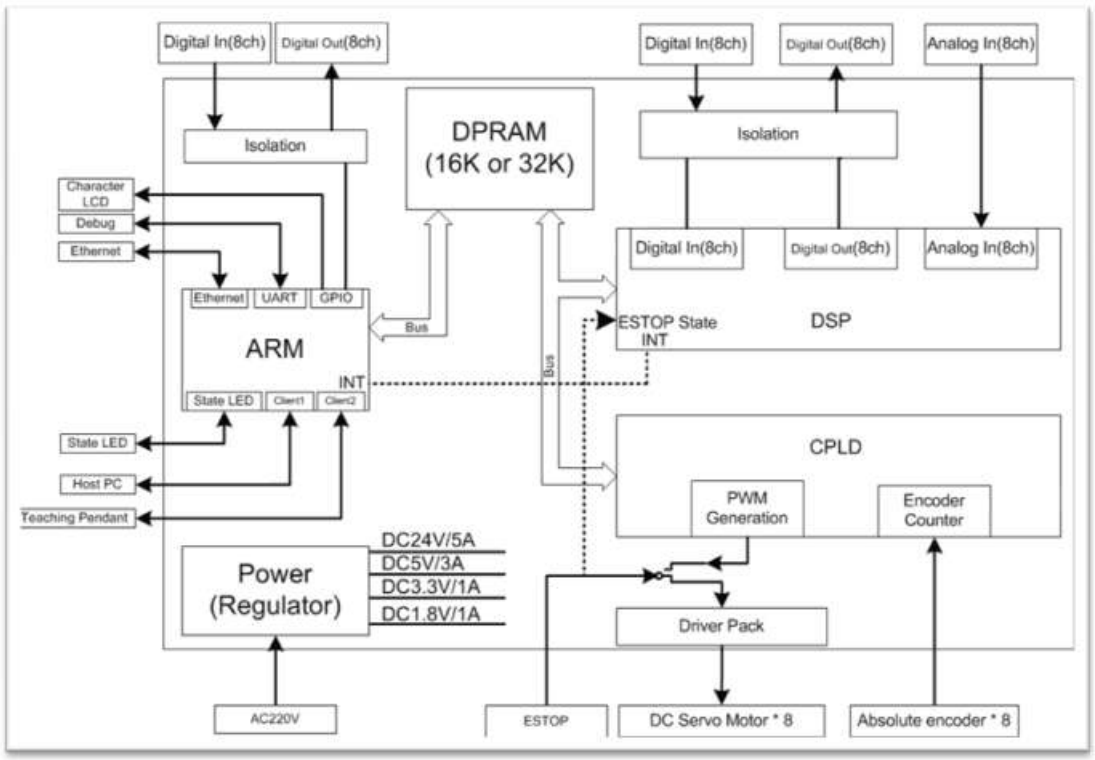


Figure II.11 Architecture Hardware de contrôleur.

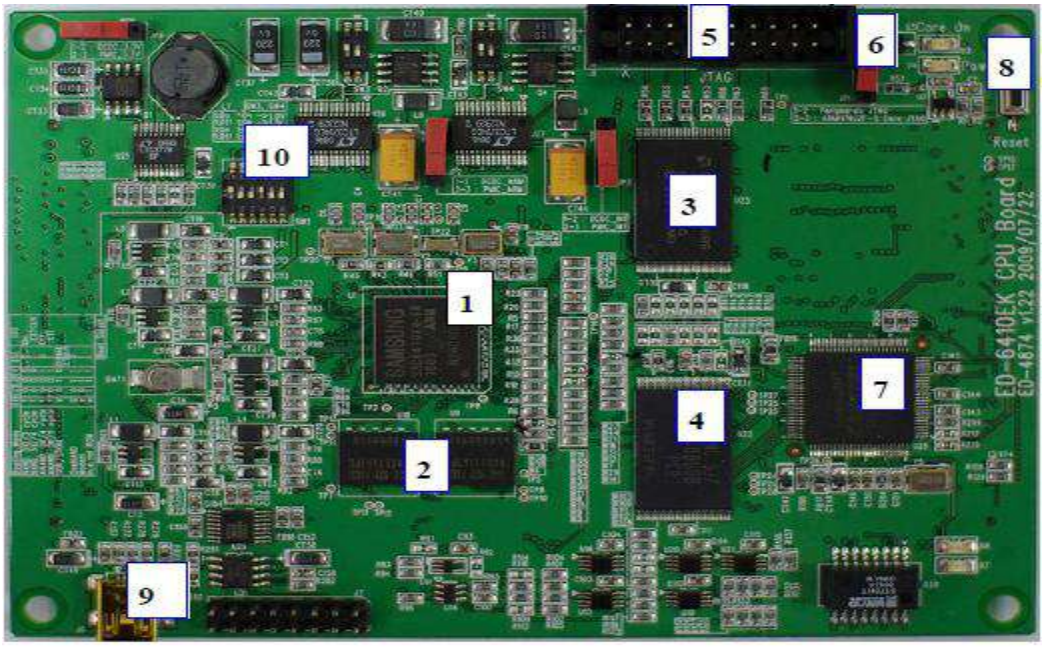


Figure II.12 Les composants internes de contrôleur.

Les principaux composants du contrôleur du robot sont énumérés dans Tableau II.3

Tableau II.4 Les principaux composants du contrôleur.

№	Nom
1	CPU(S3C6410)
2	mDDR SDRAM
3	Nor Flash
4	NAND Flash
5	JTAG Connector
6	JTAG S/W
7	Ethernet(CS8900)
8	Reset S/W
9	USB (OTG2.0)
10	Dip S/W (Bootloader)

II.2.2.1.1. Principe de fonctionnement

Une des fonctions les plus importantes de contoleur (ED-MC1) est le contrôle des mouvements du robot ED-7255 en temps réel il utilise pour cela des processeurs intégré (ARM core et DSP) .

Un programme de commande du robot écrit dans le simulateur ED-IRS qui est téléchargé vers le controleur où le programme d'interprétation de la commande (analyseur) de ARM crée un profil de mouvement en calculant la distance et la vitesse de déplacement vers la cible suivant le temps d'échantillonage et le transférer au processeur DSP. Ce dernier crée un signalde commande de moteurà l'aide d'un contrôleur de moteur de sorte que le bras de robotest entraîné.

Comme il y a d'autres fonctions , le boitier de commande effctue toute opération reliée au robot, c'est comme la liaison entre le PC, les execution externe par I/O (ex : avec API) , l'execution d'urgence etc…

II.2.3. Teach Pendant (Console de commande)

La console de commande sert d'interface entre l'opérateur et le robot , il peut ètre relier au son à travers un cable de racordement , fournit les commandes de fonctionnement au robot qui sont exécutés par 38 touches d'entrée, et ces clés d'entrée (des valeurs ou du texte) sont transmises par un microprocesseur vers le contrôleur du robot.

Pour écrire un programme de commande et exprimer l'état de robot à l'aide de ce dispositif (voir figure II.13) , ayant un écran LCD de 2 lignes, dont chacune comporte 16 caractères par ligne. Il permet de :

- Déplacer le robot en mode manuel.
- Enseigner des positions de travail.
- Vérifier un programme.
- Afficher des messages de l'usager , des messages d'erreurs , des curseurs et des menus.
- Afficher et modifier des données ,des variables, des paramétres et des entrées/sorties.

En outre, le programme ayant été écrit par un simulateur (ED-IRS) et stockée dans le contrôleur peut également être exécutée par le console de commande.

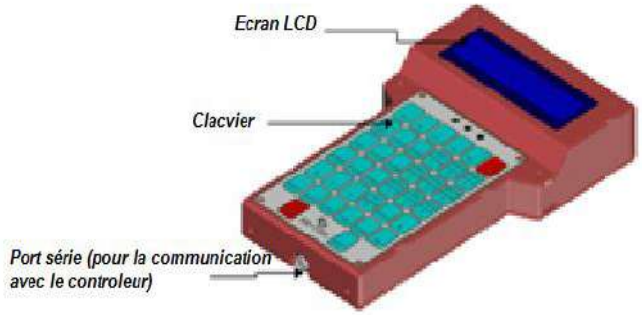


Figure II.13.Console de commande (Teach pendent).

II.2.3.1. Mode De Fonctionnement

Il dispose de quatre modes de fonctionnement

1) AUTO RUN MODE

Le mode AUTO RUN: Dans ce mode, le programme écrit est exécuté. Selon la sélection au menu d'exécution, pour, un cycle d'exécution, ligne par ligneexécution (étape Run), et l'exécution répétitive peut être fait; selon leprogramme écrit, ainsi que la commande d'entrée / sortie, le contrôle des moteurs.

2) TEACH EDIT MODE

Mode qui permet l'écriture d'un programme. Programme de contrôle de robot peutêtre écrit, édité et supprimé, et le programme déjà existé peut être importé pour créer un nouveau programme. En outre, lors de l'écriture du programme, le robot peut être déplacé pour atteindre position déployée.

3) Operate Mode

Le robot peut être déplacé par la touche HAUT / BAS avec le « teach » pendant. Le mouvement peut être effectué par rapport à l'espace articulaire ou par rapport au l'espace opérationnel.

4) PROJECT_UPLOAD

C'est un mode qui transfère le programme ayant été écrit du boîtier de commande vers le simulateur dans un ordinateur.

5) SYSTEM MODE

est un mode qui change et manipule les valeurs de réglage del'état de la vitesse et de l'utilisation / la non-utilisation d'un axe supplémentaire.

II.2.4. Présentation de langage de programmation ED-IRS :

ED-IRS (ED-Robot Industriel Simulator)est un logiciel de développement et validation des programmes opérationnels de robot ED-7255. Il permet la vérification de l'opération, des variables, et des entrées/sorties du robot à partir d'un ordinateur relié au contrôleur du robot. Il permet également d'exécuter une simulation virtuelle pour vérifier les trajectoires des mouvements avant le transfert au robot réel.

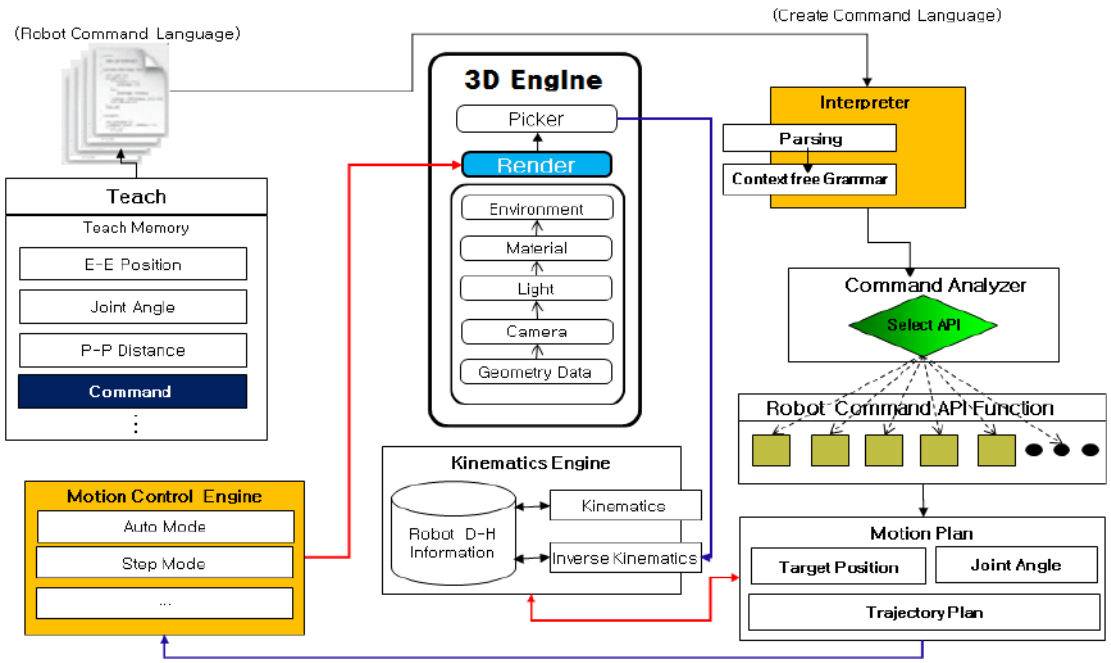


Figure II.14.Architecture de simulateur.

II.2.4.1. L'interface delogiciel ED-IRS de commande du robot ED-7255

ED-IRS offre à l'utilisateur une Flexibilité considérable à l'utilisation de ces éléments d'écran, l'environnement de travail de ED-IRS se compose de plusieurs éléments.Certains de ces éléments sont lies entre eux, on peut distinguer quatre blocs comme on peut le voir dans la figure II.15.

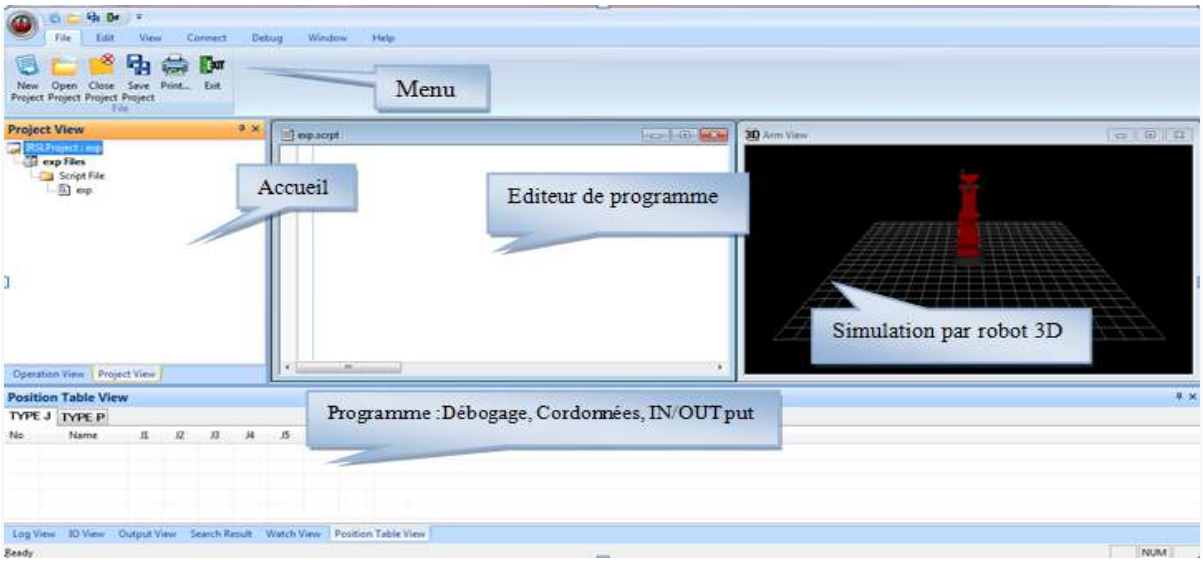


Figure II.15.Vue d'ensemble de l'interface de logiciel de programmation ED-IRS.

II.2.4.2. Menu

Le Menu se compose de 7 sous-menu. La fonction de chaque élément est illustrée dans le Tableau II.6 .

Tableau II.6 Menu de ED-IRS.

Menu fichier		Nouveau projet	**Menu connecter**		Connecter
		Ouvrir un projet			Déconnect
		Fermer un projet			Télécharger le programme au controleur
		Sauvegarder			Télécharger des modification enligne
		imprimer	**Aide**		Aide
		Sortir du programme			Info programme
Débogage		Déboguer	**Modifier**		Citer un commentaire
		Démarrer			Créer un signet
		Itération			Option
		Par Etape			chercher
		Continuer			Sélectionner tout

II.3. Théorème d'accélération en robot ED-7255

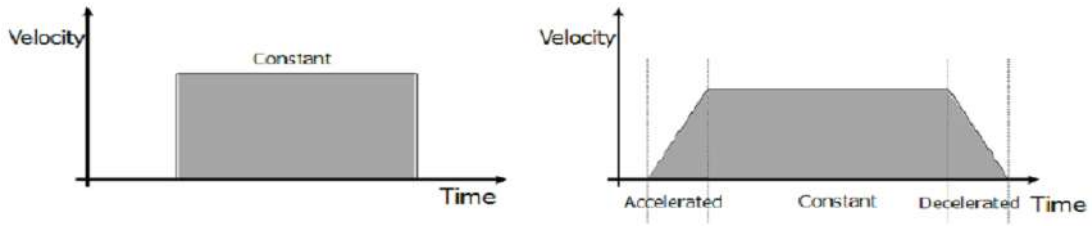


Figure II. 16. Théorème d'accélération décélération

Si une grande valeur de vitesse est entrée dans le mouvement initial d'un robot, il mettra à rude épreuve provoquant l'échec fréquent, par exemple, supposons que la vitesse d'entrée est de 30 et le temps de fonctionnement est de 2 secondes. Si un robot se déplace à une vitesse de 30 à partir de la première fois, il mettra à rude épreuve le robot. En conséquence, la zone d'accélération nécessite un ordre pour augmenter la vitesse étape par étape, comme 2, 4, 8, 15, 24, et 30 pour un temps pré réglé.

- **la différencede mouvementréel en fonction dela différencede l'intervalleentre la commandeet commande**

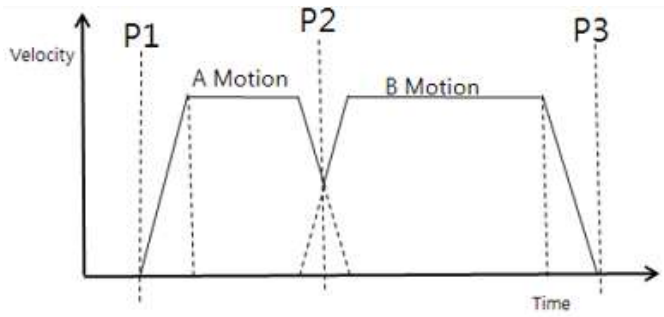


Figure II.17. Point au milieu dechevauchement entreun mouvement et un mouvement.

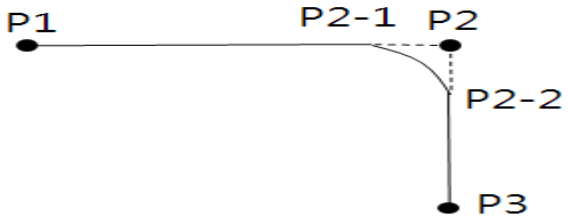


Figure II.18. Le mouvement réel dans la figure II.17.

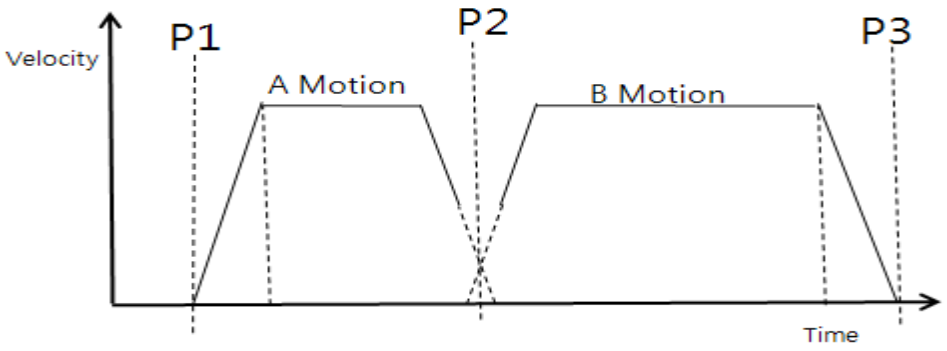


Figure II.19.Point final de chevauchement entreun mouvement etun autre.

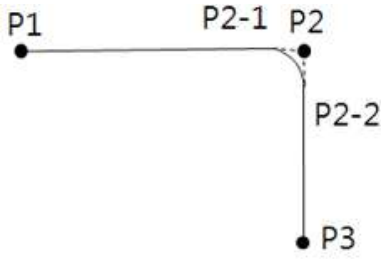


Figure II.20.Le mouvement réelde la figure II.19

Commela différence entrela figure6-76et la figure6-77, la zone de chevauchementplus important entreune commande et commande. En d'autres termes, il faitmouvementplus doux.

II.4. Conclusion

Ce chapitre est consacré pour la présentation du robot ED-7255 et ses caractéristiques matérielles (Hardware) ainsi que caractéristique logiciel (software) la modélisation de bras manipulateur a fait partie importante dans le présent chapitre.

CHAPITRE III

PRESENTATION DE LA MINI-USINE ET IHM

III.1. Introduction

L'automatisme est une discipline importante et nécessaire dans tous les secteurs industriels. Il facilite la tâche des opérateurs intervenants dans toute installation industrielle. Il permet de développer des systèmes automatisés qui assurent des tâches dangereuses, répétitives et dans des milieux hostiles pour l'homme. L'automatisation de toute unité de production augmente la productivité et améliore la qualité du produit. Les automates programmables industriels représentent l'élément important de la chaîne automatisée, car il assure de bonnes performances, meilleure flexibilité et facilite la maintenance. Une automatisation performante assure en plus d'un fonctionnement fiable de l'installation industrielle, la détection de toute anomalie éventuelle. La diversité des processus industriels nécessite des connaissances sur l'aspect processus et les différentes technologies du domaine de l'automatisme. Les automatismes câblés qui ont précédé les automates programmables industriels sont en développement en même temps que ceux des systèmes de production. Ils sont compétitifs, toute fois leur structure présente des inconvénients relatifs à leur maintenance et leur configuration.

Une combinaison entre l'API et le robot sera l'objet de cette étude, de telle sorte ils fonctionnent entre eux en synchronisation,. Une description détaillée de la mini-usine est indispensable.

III.2. Description générale de la mini-usine à production

La figure III.1 donne une vue de face de la mini-usine qui fait partie importante de notre travail. Il s'agit de l'unité ED-4031 du fabricant Sud-Coréen Sapience Corporation, ce dernier est disponible au niveau de Hall de technologie de l'université de Médéa.



Figure III.1.Vue de face sur la mini-usine.[11]

Dans la chaine de production de la mini-usine, les instruments sont fixés sur une table en aluminium, il existe un panneau de configuration pour indiquer (les entrées / sorties), c'est un système qui permet de connaitre l'état de fonctionnement de la mini-usine, et le panneau de contrôle où figure l'API qui est en quelque sorte le cerveau qui commande la mini-usine selon un programme implémenté, la table de travail possède des instruments sous forme de vérins à plusieurs types: pneumatiques à simple action sans tige et avec tige, vérin pneumatique à double action, vérin à marteau et chacun a son rôle dans la chaine de production, ces vérins marchent avec l'air comprimé qui est délivré par un compresseur d'air électrique est la régulation du débit d'air pour le fonctionnement des vérins se fait par une vanne régulatrice de débit d'air comprimé. [11]

L'unité contient également des capteurs optiques qui ont plusieurs rôles comme la détection de présence des pièces dans le magasin et la détection des pièces défectueuses. La mini-usine contient aussi un convoyeur pour le déplacement des pièces.

On peut régler la vitesse du moteur du convoyeur avec un bouton fixé en vérifiant lavitesse avec un tachymètre installé sur le panneau de commande de l'automate.

III.3. Architecture d'un système automatisé de production (SAP)

La figure III.2 permet de comprendre la structure d'un système Automatisé de production et de définir les différentes parties de ce système. Un système de production est dit automatisé

lorsqu'il peut gérer de manière autonome un cycle de travail préétabli qui se décompose en séquences et/ou en étapes. Les systèmes automatisés, utilisés dans le secteur industriel, possèdent une structure de base identique. Ils sont constitués de plusieurs parties plus ou moins complexes reliées entre elles. [12]

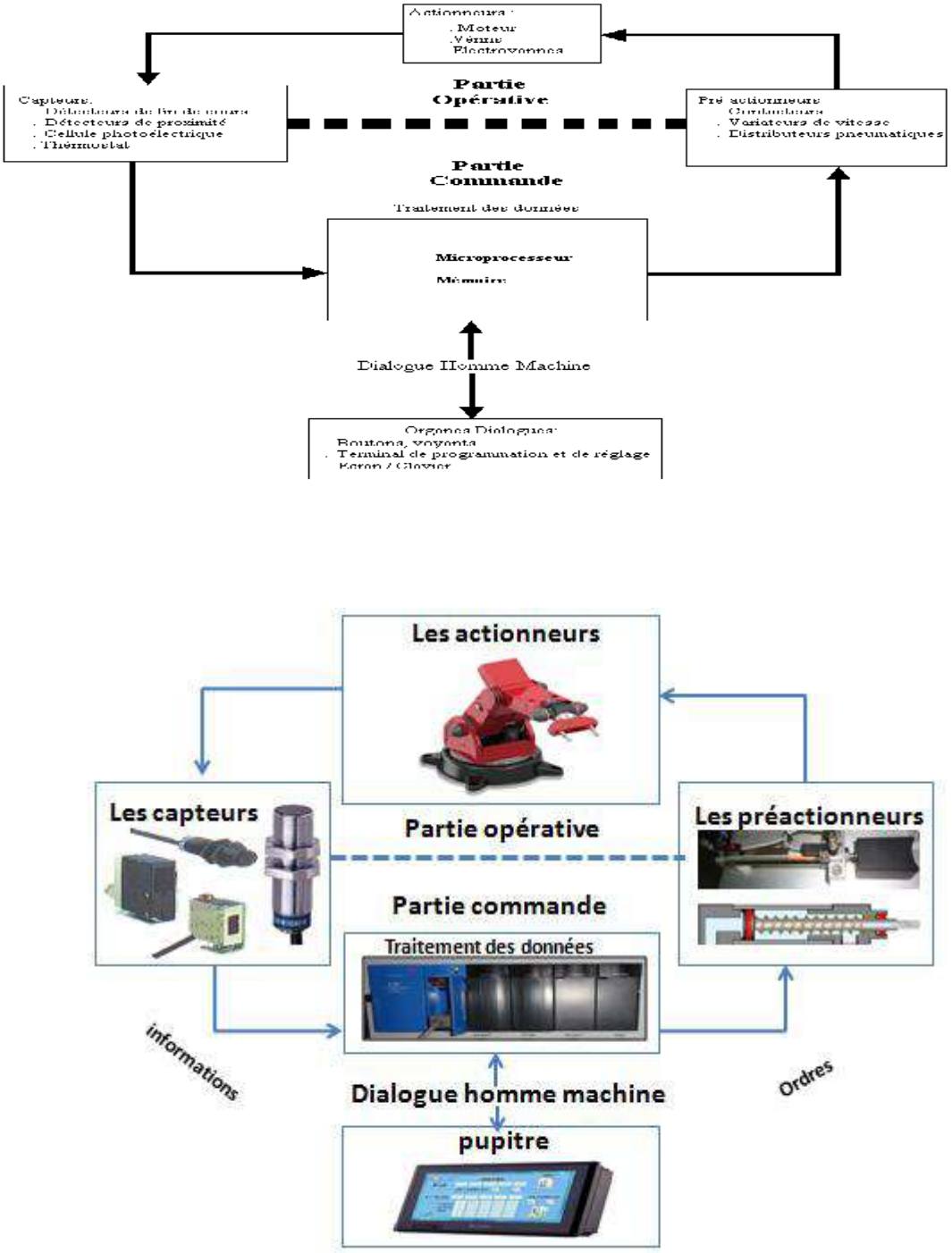


Figure III.2. Architecture d'un système automatisé de production SAP

III.3.1. Description des différentes parties

III.3.1.1. La partie opérative

C'est la partie visible du système. Elle comporte les éléments du procédé, c'est à dire :

- **Des pré-actionneurs** (distributeurs, contacteurs) qui reçoivent des ordres de la partie commande.
- **Des actionneurs** (vérins, moteurs, robots) qui ont pour role d'exécuter ces ordres. Ils transforment l'énergie pneumatique (air comprimé), hydraulique (huile sous pression) ou électrique en énergie mécanique. [12]

Dans notre chaine de production on trouve trois types de vérins :

✓ **Vérin à simple action**

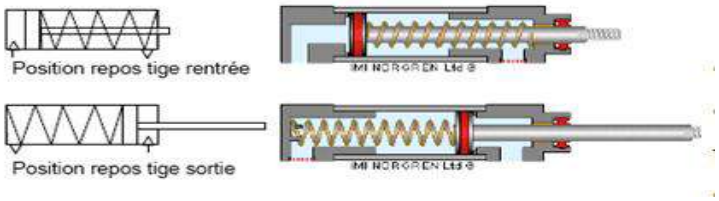


Figure III.3. Vérin à simple action

Principe de fonctionnement : Quand l'air entre dans la chambre coté fond, il pousse le piston et la tige. Le ressort contenu dans l'autre chambre est comprimée.

Mouvement : Quand la chambre est mise à l'échappement, le ressort ramène le piston et la tige rentre. Si l'échappement est ralenti, la rentrée de la tige du vérin est freinée.

✓ **Vérin à double action**

Les vérins double effet sont utilisés partout où l'aller et le retour sont à chacune une tâche contrôlée a part.

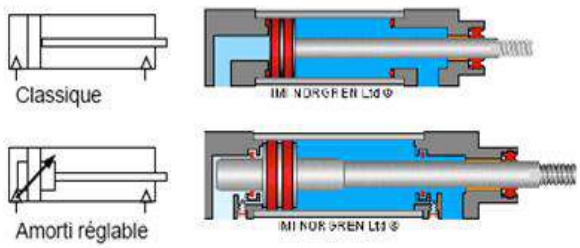


Figure III.4.Vérin à double action

Principe de fonctionnement : Quand l'air entre dans la chambre coté fond, il pousse le piston et la tige. L'air contenu dans l'autre chambre est refoulé vers l'échappement. Si l'échappement est ralenti, la sortie de la tige du vérin est freinée.

Mouvement : Quand l'air entre dans la chambre coté tige, il ramène le piston et la tige rentre, l'air de la chambre coté fond est refoulé à l'échappement. Si l'échappement est ralenti, la rentrée de la tige du vérin est freinée.

- **Des capteurs** qui informent la partie commande de l'exécution du travail, dans la mini-usine étudié on trouve deux types de détecteurs :

✓ **Détecteur de proximité capacitif (TOR)**

Son principe de fonctionnement est basé sur la variation d'un champ électrique à l'approche d'un objet quelconque.

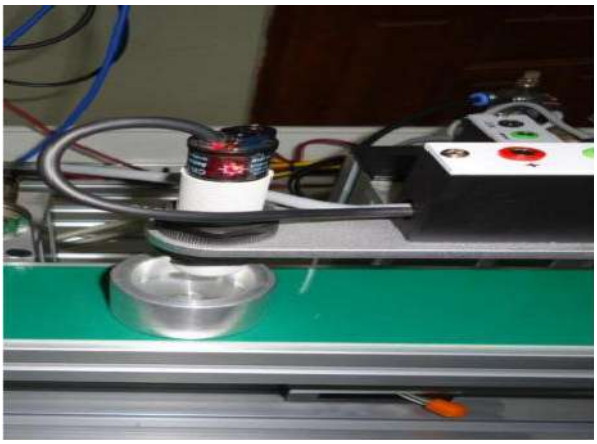


Figure III.5. Capteur des pièces métalliques.

Son rôle dans le prototype de la mini-usine est la détection à courte distance d'objet métallique ou non (pièce défectueuse ou non).

✓ **Détecteur de proximité photoélectrique (TOR)**

Les détecteurs photoélectriques se composent essentiellement d'un émetteur de lumière associé à un récepteur photosensible, son rôle dans la chaine de production est la détection de tout objet opaque, voir figure III.6

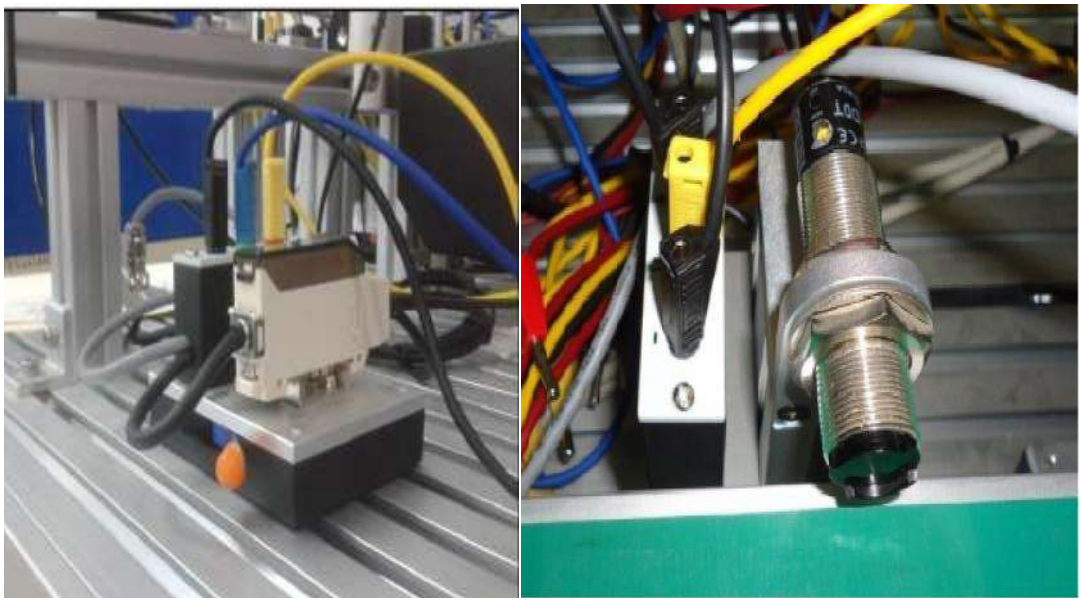


Figure III.6.Détecteur photoélectrique de la mini-usine

III.3.2. La partie commande

Cette partie de l'automatisme gère selon une suite logique le déroulement ordonné des opérations à réaliser. Il reçoit des informations en provenance des capteurs de la partie opérative, et les restitue vers cette même partie opérative en direction des pré-actionneurs et actionneurs, c'est ce qu'on appelle API (Automate Programmable Industriel).

III.3.2.1. Présentation de l'automate programmable Industriel GLOFA GM-06 et du logiciel GMWIN

C'est le cerveau le cerveau qui gère et assure la commande de la mini-usine, c'est un automate de LG type GLOFA GM-06. Il se compose comme tous les automates professionnels de plusieurs parties et notamment d'une mémoire programmable dans laquelle l'opérateur écrit, dans un langage propre à l'automate des directives concernant le déroulement du processus à automatiser.

III.4. Architecture matérielle de l'API

Les API en général comportent quatre parties principales :

Le processeur : Son rôle consiste d'une part à organiser les différentes relations entre la zone mémoire et les interfaces d'E/S et d'autre part à gérer les instructions du programme.

La mémoire : Elle est conçue pour recevoir, gérer et stocker des informations issues des différentes parties du système que sont le terminal de programmation (PC ou console) et le processeur, qui lui gère et exécute le programme. Elle reçoit également des informations en provenance des capteurs et détecteurs. [12]

Les interfaces : L'interface des entrées comporte des adresses d'entrée, une pour chaque capteur. L'interface des sorties comporte des adresses de sorties, une pour chaque pré-actionneur. Le nombre d'E/S varie suivant le type de l'automate. [12]

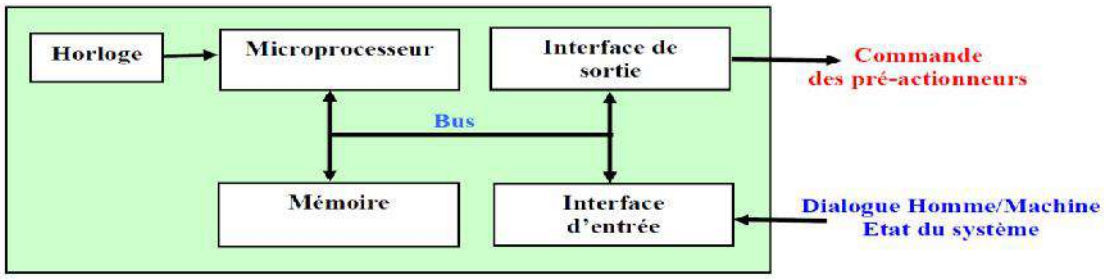


Figure III.7.*Architecture matérielle d'un API.*

Ces quatre parties sont reliées entre elles par des bus (ensemble câblé autorisant le passage de l'information entre ces 4 secteurs de l'API). Ces quatre parties réunies forment un ensemble compact appelé automate.

Un automate exécute son programme de manière cyclique (Figure III.8) .Le temps d'exécution d'un cycle est de l'ordre d'une vingtaine de millisecondes et il est contrôlé par une temporisation appelée chien de garde.

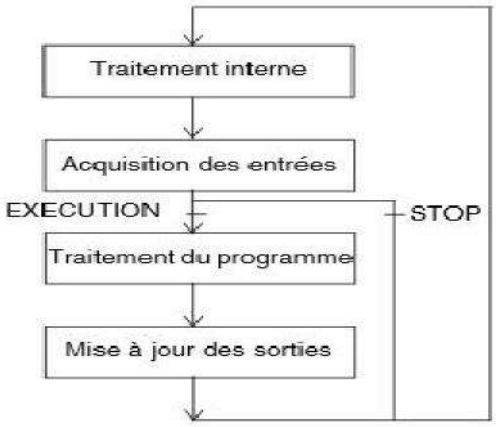


Figure III.8.*Cycle de fonctionnement de l'API.*

III.5. Choix d'un automate programmable industriel

Après l'établissement du cahier des charges, il revient à l'utilisateur de regarder sur le marché l'automate le mieux adapté aux besoins, en considérant un certain nombre de critères importants tel que :

- Le nombre et la nature des E/S
- Le type des entrées/sorties nécessaire (numérique, analogique, etc.…)
- La nature du traitement (temporisation, comptage, ...)
- Les moyens de dialogue et le langage de programmation (la console détermine le langage de programmation, IHM)
- La communication avec les autres systèmes
- Les capacités de traitement du processeur (vitesse, données, opérations, temps réel...)
- Les moyens de sauvegarde du programme
- La fiabilité, robustesse, immunité aux parasites
- La documentation, le service après-vente, durée de la garantie, la formation. [12]

III.6. Langages de programmation pour API

Chaque automate possède son propre langage. Mais par contre, les constructeurs proposent tous une interface logicielle répondant à la norme. Cette norme définit cinq langages de programmation utilisables qui sont :

- **Langages Graphiques**

  - **LD** (Ladder Diagram)Diagramme en échelle
  - **SFC** (Sequential Function Chart) GRAFCET (GRAphe Fonctionnel de Commande Étape/Transition).
  - **FBD** (Function Block Diagram) Logigramme [12].

- **Langages textuels**

  - **IL** (Instruction List) Liste d'instructions
  - **ST** (Structured Text) Texte structure.

III.7. Software

III.7.1. Découverte du logiciel GMWIN

Est un logiciel de programmation et de configuration des automates GLOFA type GM-06 de LG. Il permet également de créer des programmes avec les différents types langage cité récemment où on peut même vérifier le programme avant de le transférer à l'automate en utilisant le simulateur montré par la figure III.9.

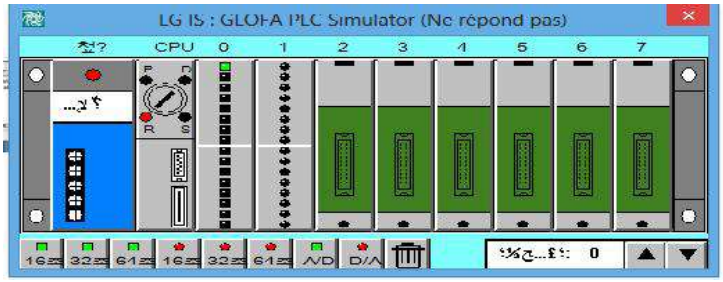


Figure III.9. Simulateur de l'API GLOFA. [14]

III.7.2. L'interface de logiciel

La figure III.10 montre l'interface de base du logiciel .

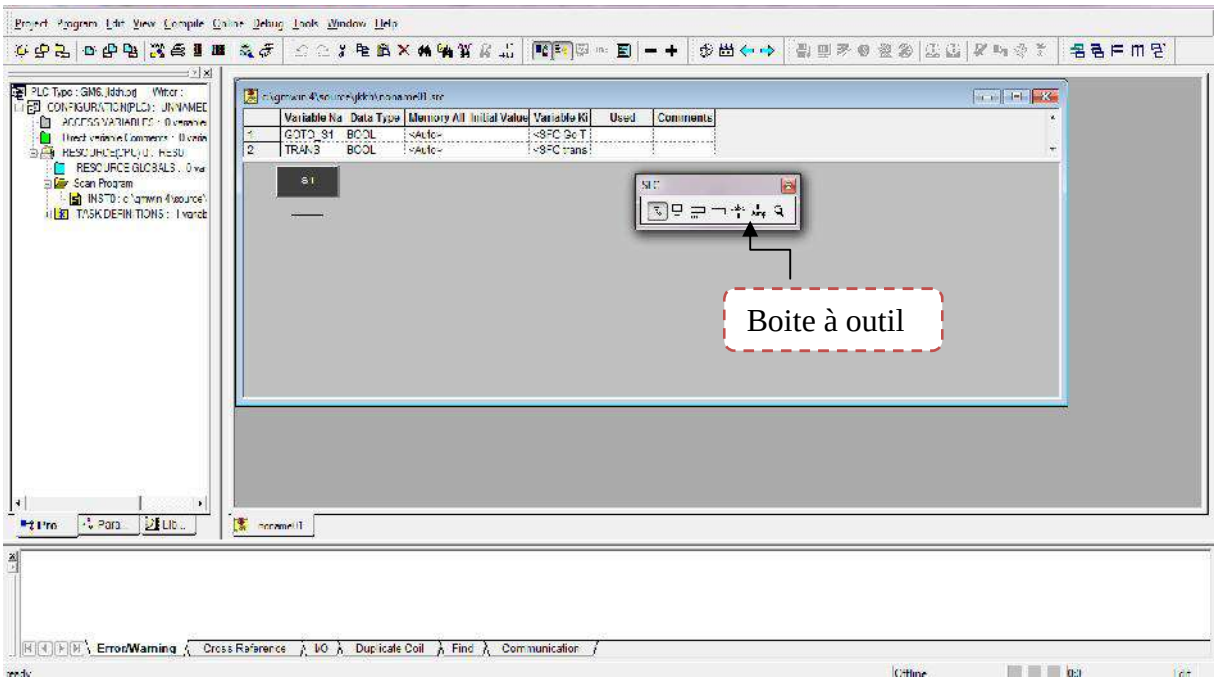


Figure III.10.L'interface de GMWIN.

La boîte à outils (Figure III.11) contient toutes les commandes utilisées pour éditer un programme, elle contient cinq outils définit dans le Tableau III.1



Figure III.11.la boite à outil.

Tableau III.1. Désignation de la boite à outils.

Symbole	désignation
	La flèche : pour le mouvement, édition et effacement en choisissant un point de contact
	Les étapes et les transitions
	Pour faire les actions dans les étapes
	Pour faire la reprise d'étape (bouclage)
	Pour faire les liaisons orientées

Le Tableau III.2 représente la barre d'outile de GMWIN ainsi que leur signification.

Tableau III.2.Barre d'outils de logiciel GMWIN.[14]

Tool	Command	Tool	Command	Tool	Command
	New project		Connect+Write+Run Monitor On		Make
	Open		Connect		Library manager
	Save		Disconnect		Start Simulation
	New program		Write		Redo
	Open		Monitor On/Off		Find from files
	Save		Run		Go To
	Local variables		Stop		Title Vertically
	Undo		Pause		Close all
	Cut		Begin Debug		Project Window
	Copy		Go		Output Window
	Paste		Step over		Variable Monitor
	Delete		Step in		I/O Monitor
	Find		Step out		Zoom Out
	Replace		Pause		Zoom In
	Find Next		Run to Cursor		Print
	Compile		Insert/Remove Breakpoint		New Window
	Full screen		Write in Online Edit		Tiered configuration
	Previous Message		System		Title Horizontally
	Next Message		I/O information		PLC history
	Online Edit Start		Data share		

III.8. L'interface homme machine

III.8.1. Introduction

Au cours de l'automatisation de l'industrie aussi que l'interface opérateur qui est habituellement une Unité Interface-Homme-Machine.

IHM joue un rôle important dans la création d'un environnement visuel amical entre l'utilisateur et la technologie. Il est considéré comme étant la fenêtre pour le système de commande d'automatisation. [15]

Le contrôle par toucher du doigt a remplacé l'utilisation des interrupteurs et des manuels énormément. Avec la demande croissante des IHM dans l'industrie, un bon nombre de logiciels sont mis en place et la compétitivité dans la conception des écrans tactiles d'IHM est en augmentation.

Pour cela j'ai pensé à faire une conception IHM à notre prototype industriel pour faciliter l'interaction entre l'utilisateur et le process. Parlant des logiciels, la disponibilité de l'écran tactile Panel « TH765-NU de WUXI XINJI ELECTRIC » s'est présenté des vue de face ,de coté , et arrière par la figure III.12 , m'as permis de concevoir mon interface en utilisant son propre logiciel touchWin qui était la solution idéale.

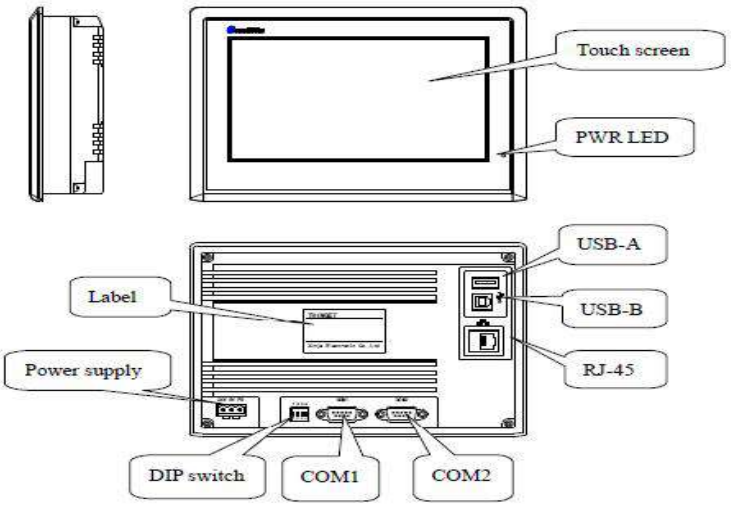


Figure III.12. Structure de l'écran touchwin. [16]

Créer l'interface graphique et les variables, c'est pouvoir lire les valeurs du processus via l'automate, les afficher pour que l'operateur puisse les interpréter, et ajuster, éventuellement. (Figure III.13).

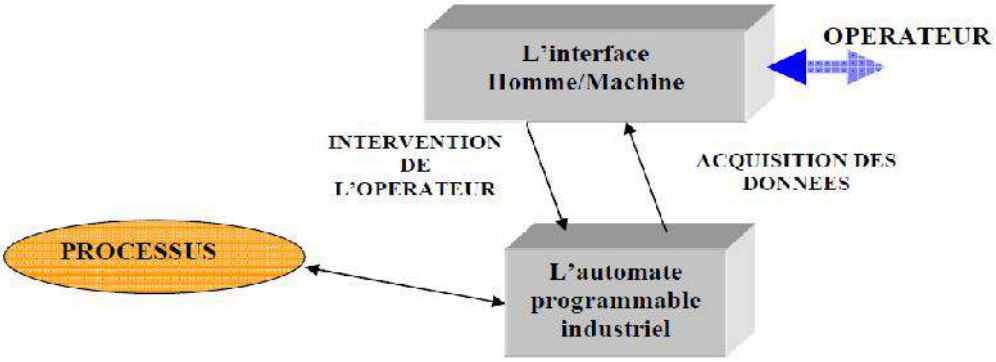


Figure III.13. L'interface Homme/Machine dans un processus automatisé.[17]

III.8.2. Logiciel de conception et de configuration d'interface TouchWin :

La série TH peut se communiquer avec la plupart des automates industriels comme s'est montrée sur la figure III.14 tels que GLOFA.

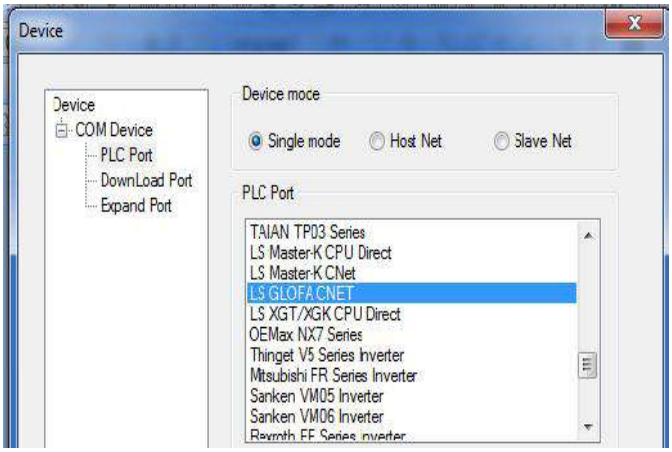


Figure III.14. Fenêtre de choix de l'API correspond.

Dans le TouchWin, chaque projet crée contient principalement des vues que l'on crée pour le contrôle-commande de machines et d'installations. Lors de la création des vues, on dispose d'objets prédéfinis permettant d'afficher des procédures et de définir des valeurs du processus. Les différents outils et barres de l'éditeur de vues sont représentes dans la figure III.15.

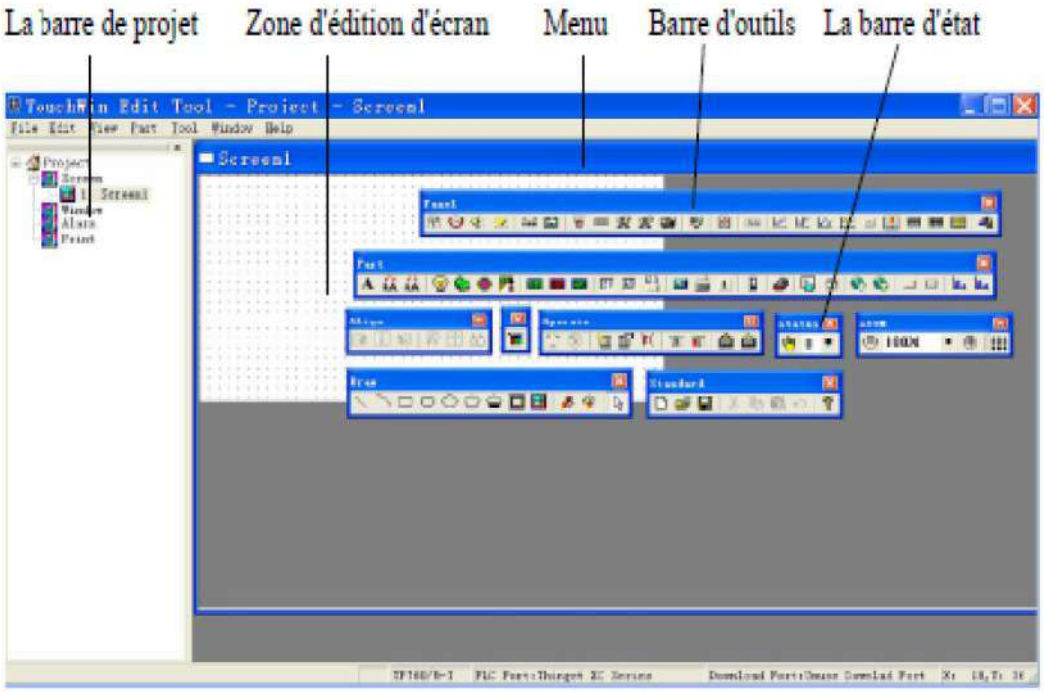


Figure III.15. Vue d'ensemble de l'interface de logiciel de conception TouchWin .

III.8.3. Utilisation des variables :

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

Les variables externes permettent de communiquer, c'est-à-dire d'échanger des données entre les composants d'un processus automatise, entre un pupitre operateur et un automate p. ex.

Une variable externe est l'image d'une cellule mémoire définie de l'automate. L'accès en lecture et en écriture a cette cellule mémoire est possible aussi bien a partir du pupitre operateur que de l'automate. Chaque variable peut être configurée selon le besoin.

III.9. Conclusion

Dans ce chapitre on a présenté la mini-usine qui fait partie essentiel dans notre projet et toute son architecture matériel et logiciel de programmation qui est un outil aussi important à la commande, en donnant des généralités sur les automates programmable industriel qui est le cerveau de cette cellule de production.

On a abordé aussi à l'interface Homme-Machine en donnant un aperçu sur logiciel de conception et de configuration TouchWin. Cette brève étude nous a facilité la pratique, et faire la liaison entre le robot ED-7255 et la chaine de production avec l'interface Homme-Machine, et ça sera l'objet essentiel du prochain chapitre et même la tâche essentiel de notre cahier des charges.

CHAPITRE IV

REALISATION PRATIQUE

IV.1. Introduction

Dans les chapitres précédents nous avons vu un aperçu sur le Robot ED-7255 et son principe de fonctionnement et nous nous somme familiarisés avec son langage de programmation en ayant une idée sur leurs instructions de base, ainsi qu'un aperçu sur la mini-usine et la programmation des APIs GLOFA de LG avec l'outil de programmation GMWIN et même on a abordé l'exploitation visuelle dans le domaine industriel avec TouchWin.

Le présent chapitre abordera le côté pratique de notre projet où trois volets seront abordés, à savoir le contrôle du robot, la connexion robot/mini-usine en utilisant un API et la réalisation d'une interface homme machine pour le contrôle du robot.

IV.2. Le cahier des charges

Le cahier des charges à réaliser dans le cadre de ce mémoire de fin d'études peut être résumé par les points suivants:

1) La maitrise du contrôle du robot ED-7255 à travers son Software et la connexion avec le milieu extérieur.
2) La connexion du robot avec un prototype d'une mini-usine.
3) Réalisation d'une IHM pour le contrôle du système robot/mini-usine.
4) Contribution à la réalisation d'un support de travaux pratiques sur le robot

IV.3. Programmation d'une commande du Robot 5 axes ED-7255

IV.3.1. Notions de base pour développement d'un programme

Pour développer un programme de commande de robot on doit passer par plusieurs étapes qui sont illustrées par la Figure IV.1.

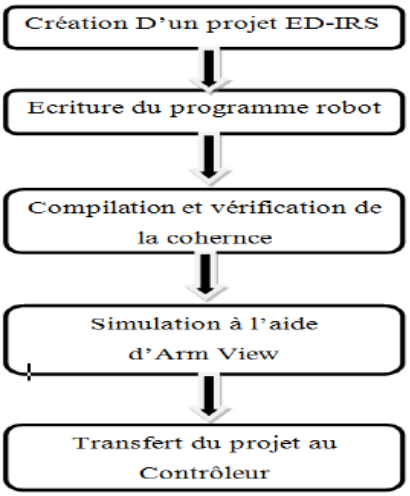


Figure IV.1.Étapes pour la création d'un programme via ED-IRS

IV.3.1.1. La commande en position

En robotique, une des tâches de base que un robot manipulateur doit accomplir consiste à se déplacer d'un point A à un point B, avec le respect éventuel d'un certain nombre de contraintes (vitesse, accélération max, évitement d'obstacle). La commande en position du robot consiste donc à calculer les différentes consignes, en fonction du temps, afin de générer le mouvement désiré. C'est pour cela que la commande en position d'un robot est aussi appelée génération de mouvement.

D'une manière classique, deux types de commande en position sont envisagée :

– une commande en position dans l'espace articulaire q (Figure IV.2. (a)). Pour assurer le passage de la position initiale à la position finale ($q_i \rightarrow q_f$ passant par la MGD) et le contrôler temporellement, une génération de mouvement est nécessaire. Par exemple, si l'on souhaite synchroniser les axes pour avoir un mouvement harmonieux et contrôler la géométrie du déplacement du robot à chaque itération.

– une commande en position dans l'espace opérationnel (Figure IV.2.(b) où le passage par la MGI est indispensable.

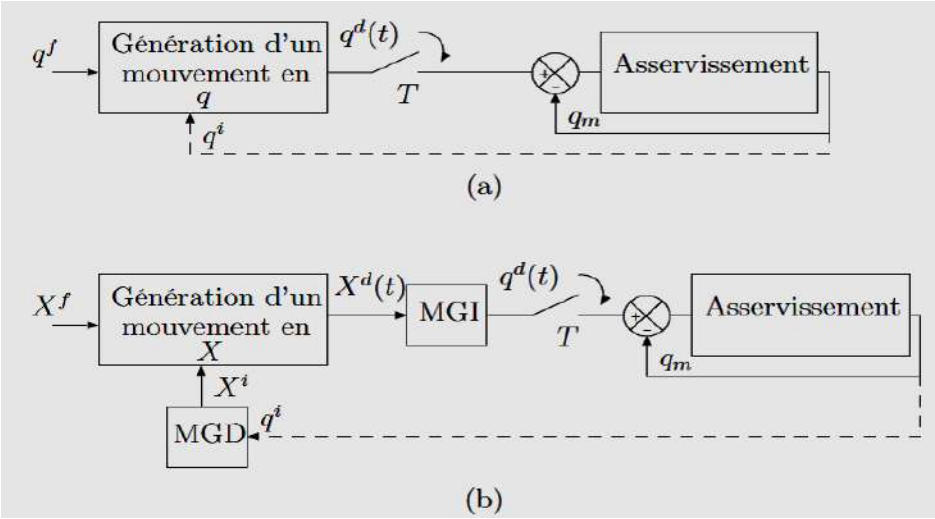


Figure IV.2 Boucles de génération de mouvement. (a) : dans l'espace articulaire - *(b) : dans l'espace opérationnel*

Le software du robot ED-7255 fournit l'accès afin de choisir le type de la commande du robot, sur la barre d'accueil (figure IV.3) où on trouve le mode cartésien (espace opérationnel) et le mode joint (espace articulaire).

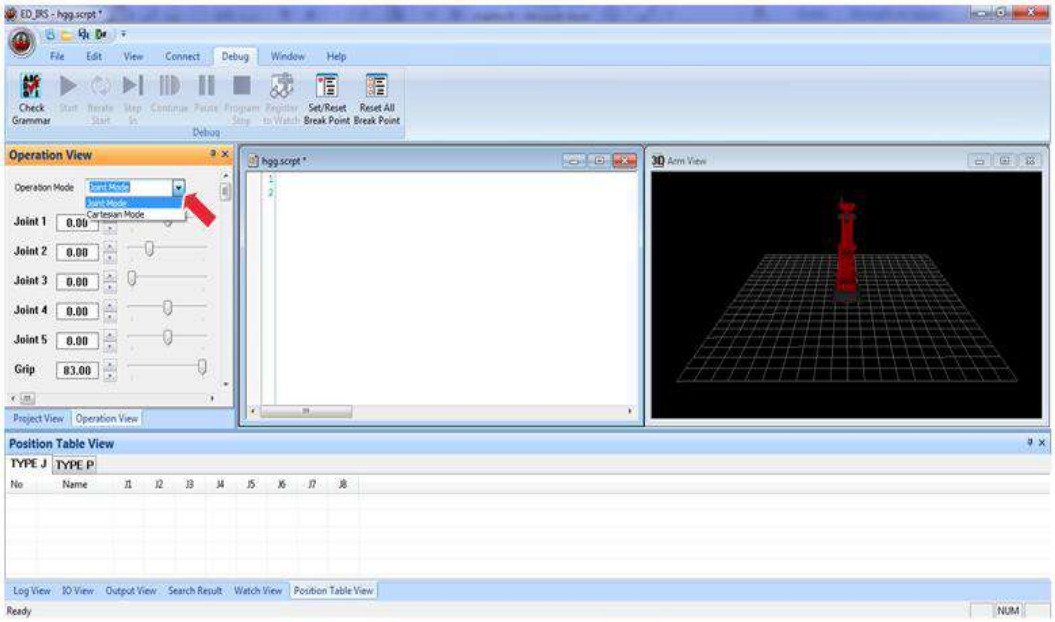


Figure IV.3. Choix du mode de la commande.

IV.4. Calibration des robots et leurs positions de repos

Tout sorte de robots a sa position de repos on bien la position sécurisée qui se définit selon son emplacement et ses contraintes de travail et ne peut pas être changé ou écrasé, c'est la position « HOME » dans le cas de Robot ED-7255 et comme les capteurs sont situés dans chaque articulation sont fonctionnels que en étant alimenté, cela veut dire que dès que le robot est hors tension tout déplacement n'est pas pris en compte, et comme les moteurs situés dans les articulations ne sont pas freinés (en étant hors tension) donc il n'y a pas de prévention et forcement les déplacements vont causer des décalages d'où la nécessité de faire une calibration à chaque démarrage du robot.

IV.5. Mode de déplacement de l'organe terminal

Le déplacement de l'organe terminal d'un point à un autre peut se faire de plusieurs maniéres,il peut étre aléatoire linéaire ou méme circulaire comme c'est éllustré par la figure IV.4, tout dépend de l'instruction utilisée, pour cela trois choix sont à notre disposition :

P : PTP (point to point) Contrôle. L'outil a libre parcours à travers les positions.

L : commande linéaire du TCP**(Tool Center Point)**. L'outil décrit un mouvement linéaire entre la position actuel et la position cible.

C : commande circulaire du TCP.L'outil décrit un mouvement d'un arc de cercle.

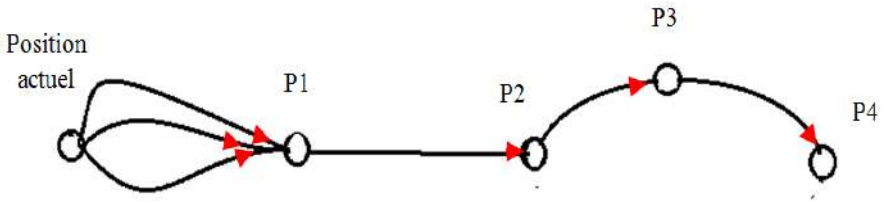


figure IV.4. Différent mode de déplacement.

IV.6. Méthodologie de programmation de robot

Partant d'une application donnée il faut définir :

- Les repères nécessaires (les points fonctionnels) on associe pour chacun de ces derniers les points d'approche et de dégagement.
- Ajouter des points intermédiaires, indispensable surtout si lespoints sont éloignés (contrôler la trajectoire), ou bien pour évitement des collisions.
- Les actions requises.
- Construire un programme textuel en enregistrant les points fonctionnels sous forme de données cartésiennes ou articulaire dans la table des positions (voir figure IV.5.) où la sélection « TYPE P » est pour les positions cartésiennes et « TYPE J » est pour les positions articulaires.

Position Table View

TYPE J | TYPE P

No	Name	J1	J2	J3	J4	J5	J6	J7	J8
1	JPOS1	-54.00	-44.00	-52.00	-86.00	0.00	302.79	0.00	0.00
2	JPOS2	-54.00	-63.00	-64.00	-55.00	0.00	0.00	0.00	0.00
3	JPOS3	54.00	-44.00	-52.00	-86.00	0.00	0.00	0.00	0.00
4	JPOS4	54.00	-63.00	-64.00	-55.00	0.00	0.00	0.00	0.00
5	JPOS5	-20.00	-44.00	-52.00	-86.00	0.00	0.00	0.00	0.00
6	JPOS6	-20.00	-63.00	-64.00	-55.00	0.00	0.00	0.00	0.00

Log View | IO View | Output View | Search Result | Watch View | Position Table View

Figure IV.5. Tableau de coordonnées des positions articulaires.

IV.7. La manipulation «Pick and Place»

La tâche la plus essentiel pour un robot série est la manutention des objets d'une position à une autre.

Cet manipulation expliquera pas à pas pour développer un programme dans un environnement virtuel qui simulera la trajectoire d'un robot lors d'une manipulation «Pick and place» pour les deux types de commande, le mode cartésien et le mode articulaire en prenant par considération la vitesse et l'accélération de robot le long des mouvements

IV.7.1. Développement de programme d'un « pick and place » classique

Nous commençons par un programme pour déplacer un objet de P1 à P2 « pick& place » simple, en utilisant les concepts cités précédemment.

IV.7.1.1. Commande en position dans l'espace articulaire

En choisissant « joint mode », Ce mode entraine chacun des cinq articulations indépendammentoù les positions sont définis par l'angle de rotation(en degré) de chaque axe (la commande passe par la MGD). Les points fonctionnels sont enregistrés préalablement.

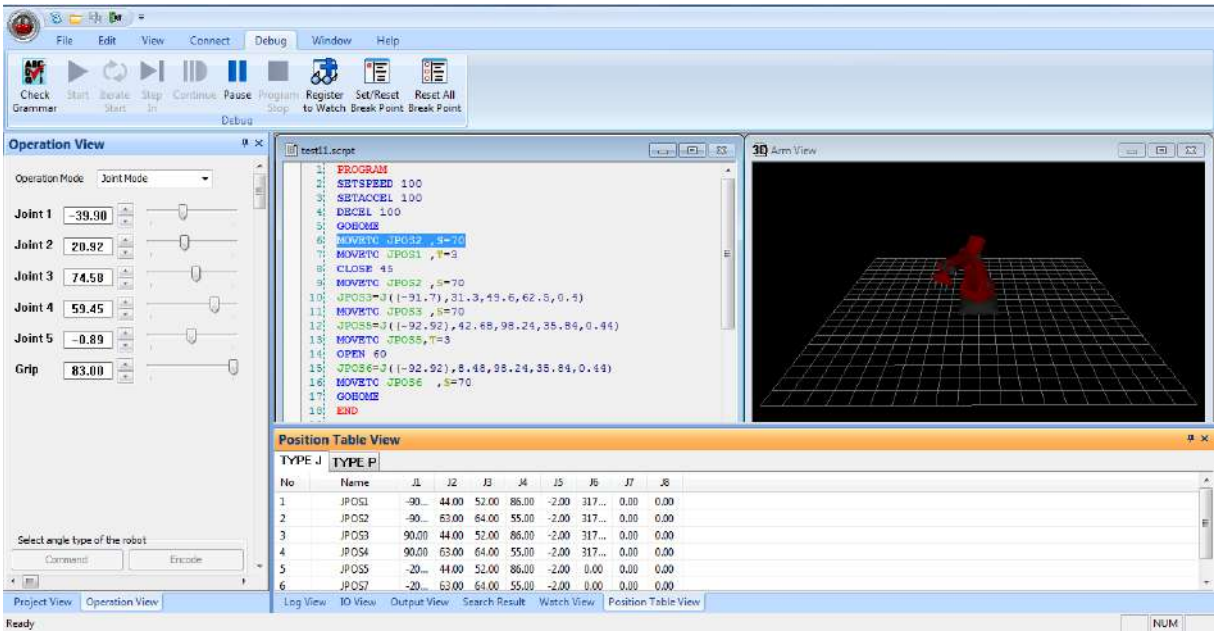


Figure IV.6.Environnement de travail virtuel.

Tableau IV.1. Programmation d'un pick and place passant par la MGD.

PROGRAMME	Commentaire
PROGRAM SETSPEED 100 SETACCEL 50 DECEL 50 DEFPOSJ JPOS1,JPOS2,JPOS3,JPOS4 GOHOME MOVETO JPOS2 ,S=70 MOVETO L, JPOS1 ,T=3	-Chaque programme doit commencer parPROGRAM et se termine par END - Sélection de 100 % de la vitesse max et 50% d'accélération et 50% de décélération du robot -Déclaration des positions. -GOHOME : c'est position référence de robot qu'elle ne peut pas changer ou écraser. -Déplacer le robot aléatoirement à JPOS2

CLOSE 45 MOVETO L, JPOS2 ,S=70 MOVETO JPOS3 ,S=70 MOVETO L, JPOS4,T=3 OPEN 60 MOVETO JPOS3 ,S=70 GOHOME END	puis linéairement à JPOS1 en précisant la vitesse de déplacement vers JPOS2 et le temps d'arrivé à JPOS1, et ainsi de suite pour les autre mouvements. -L'instruction OPEN et CLOSE sert à ouvrir ou fermer la pince (poser ou rattraper un objet) sachant que son ouverture maximale est de 80mm. -La dernière position doit être GOHOME pour qu'il revienne toujours à sa position initiale sécurisé.

IV.7.1.2. Commande en position dans l'espace opérationnel

En choisissant « cartésien mode », cette fois ci je devais définir les point fonctionnels en utilisant les coordonnées cartésiennes (X, Y, Z) pour la même cible que précédent.

Tableau IV.2. Programmation d'un pick and place passant par la MGI.

PROGRAMME	Discussion
PROGRAM SETSPEED 100 SETACCEL 50 DECEL 50 DEFPOS POS1,POS2,POS3,POS4 GOHOME MOVETO POS2 ,S=70 MOVETO L, POS1 ,T=3 CLOSE 45 MOVETO L, POS2 ,S=70 MOVETO POS3 ,S=70 MOVETO L, POS4,T=3 OPEN 60 MOVETO L,POS3 ,S=70 GOHOME END	-Chaque programme doit commencer PROGRAM et se termine par END - Sélection de 100 % de la vitesse max avec 5 d'accélération et 50% de décélération du robot -Déclaration des positions. -GOHOME : c'est position référence de robot qu'elle ne peut pas changer ou écraser. -Déplacer le robot aléatoirement à POS2 puis linéairement à POS1 en précisant la vitesse de déplacement vers POS2 et le temps d'arrivé à POS1, et ainsi de suite pour les autre mouvements. -L'instruction OPEN et CLOSE sert à ouvrir ou fermer la pince (poser ou rattraper un objet) sachant que son ouverture max=80mm. -La dernière position doit être GOHOME

IV.7.1.3. Phase de compilation et simulation :

Cette étape est importante avant de faire le transfert de programme au contrôleur, elle sert à vérifier le parcours des trajectoires et même à visualiser les mouvements du robot le long des trajets.

IV.7.1.3.1. Transfert de projet au contrôleur

Pour exporter un fichier vers le robot réel, il faut tout d'abord prendre en compte toutes les notions de sécurité citées précédemment pour ne pas causer des collisions et même des auto-collisions.

La connexion en ligne avec le robot et le transfert se faites suivant les étapes démontrées sur les figures suivantes.



Figure IV.7. Connexion avec le contrôleur de robot ED-7255



Figure IV.8. Sélection de choix pour transférer (download)

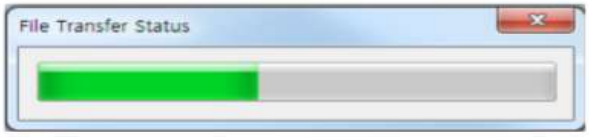


Figure IV.9. Transfert de programme en cours.

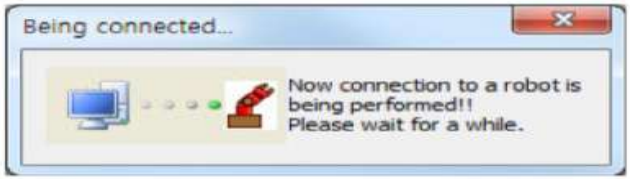


Figure IV.10 Connexion en cours avec le contrôleur.

IV.7.2. Commande de robot ED-7255 avec une condition pour la comparaison

IV.7.2.1. Description de la tâche à faire

Pour le but de faire déplacer des pièces de deux emplacements différents vers un troisième (de P1et P2 successivement à P3) de telle sorte de déplacer 10 pièces seulement et puis il s'arrête, pour cela nous avons entamé les boucles de conditionnement avec l'utilisation des sous programmes.

Nous avons utilisé le concept des nombres pairs et impairs pour atteindre la cible.

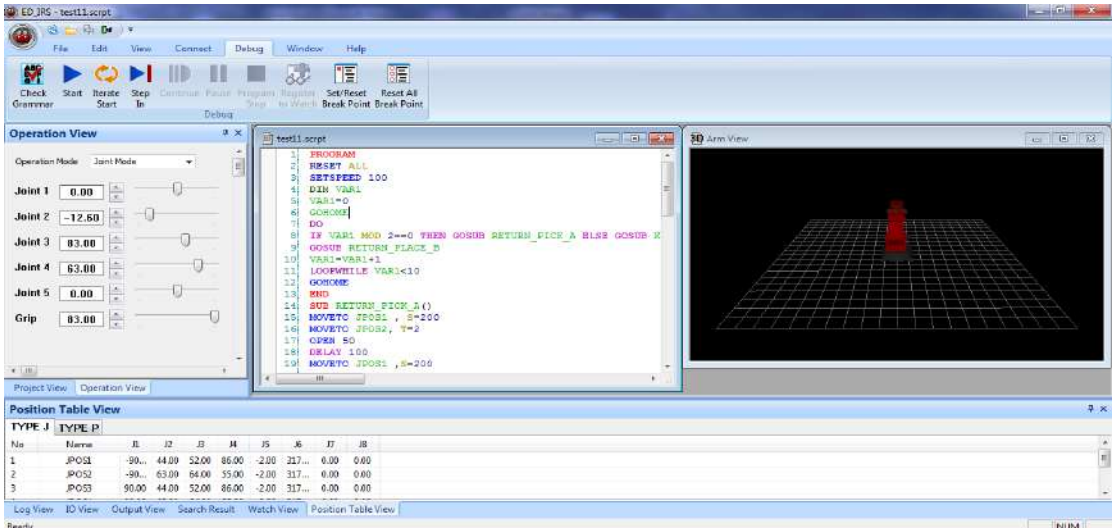


Figure IV.11. Environnement de travail virtuel

IV.7.2.2. Développement du programme de commande

Le programme correspond est donnée avec des commentaires explicatifs dans le tableau IV.3

Tableau IV.3. Commande conditionnée.

PROGRAMME	Commentaire
PROGRAM SETSPEED 100 DIM VAR1 VAR1=0 GOHOME DO IF VAR1 MOD 2==0 THEN GOSUB PICK_A ELSE GOSUB PICK_B	-Début de programme. - La vitesse -Déclaration des variables. -GOHOME : c'est position référence de robot qu'elle ne peut pas changer ou écraser. -Boucle DO…..LOOPWHILE -Condition IF ……THEN…..ELSE, pour la

GOSUB PLACE_C VAR1=VAR1+1 LOOPWHILE VAR1<10 GOHOME END SUB PICK_A() MOVETO POS1 , S=100 MOVETO POS2, T=2 OPEN MOVETO POS1 ,S=100 END SUB SUB PICK_B() MOVETO POS5 , S=100 MOVETO POS7 CLOSE MOVETO POS5 ,S=100 END SUB SUB PLACE_C() MOVETO POS3 ,S=100 MOVETO POS4 ,T=2 OPEN MOVETO POS3, S=100 END SUB	vérification de la variable si elle est de nombre paire ou impaire ,et GOSUB pour les saut au sous programme. -Incrémenter le compteur après à la fin de chaque itération. -La série d'instructions, ce sont les sous programme que chacun contient une suite d'instructions pour générer les différent mouvement en déplaçant le bras au dessus de la pièce à rattraper (ou dessus de la cible où poser) puis il décent et prend la pièce légèrement en utilisant un temps d'arriver modéré.

IV.7.3. Conclusion

Cette manipulation nous permette d'avoir des concepts assez importants pour les manipulations suivantes. On a appris à :

- Ajouter une étiquette ou label et GOTO pour permettre au programme à effectuer une boucle de routage.
- Comment déclarer une variable et la remise à zéro.
- Donnez une instruction de conditions de comparaison et faire une décision dépend de la variable.
- Faire appeler une fonction correspondante.
- Répéter des tâches.

IV.8. Résolution de la problématique de la tour d'HANOI

Dans le but de maitriser la programmation et le contrôle du robot, on s'est proposé de la résolution du problème de la tour de Hanoi par le robot.

Le problème des tours de Hanoï(Figure IV.12) est un jeu de réflexion imaginé par le mathématicien français Édouard Lucas, et consistant à déplacer des disques de diamètres différents d'une tour de « départ » à une tour d'« arrivée » en passant par une tour « intermédiaire », et ceci en un minimum de coups, tout en respectant les règles suivantes :

- on ne peut déplacer plus d'un disque à la fois, cela doit toujours être le plus petit (celui qui se trouve au sommet).
- on ne peut placer un disque que sur un autre disque plus grand que lui , ou sur un emplacement vide.

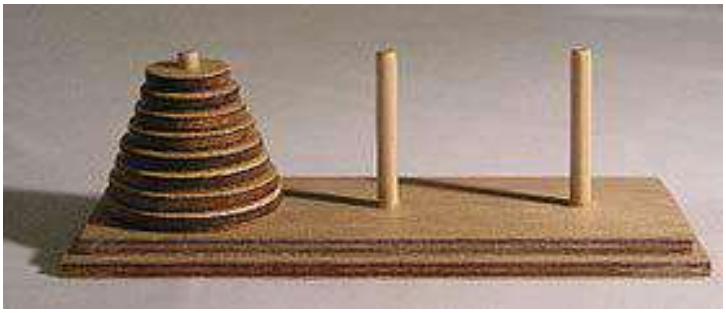


Figure IV.12. Image correspondante à la tour d'HANOI.

Origine du problème

Sous le titre « Les brahmes tombent », dans le grand temple de *Bénarès*, au-dessous du dôme qui marque le centre du monde, trois aiguilles de diamant, plantées dans une dalle d'airain, hautes d'une coudée et grosses comme le corps d'une abeille. Sur une de ces aiguilles, Dieu enfila au commencement des siècles, 64 disques d'or pur, le plus large reposant sur l'airain, et les autres, de plus en plus étroits, superposés jusqu'au sommet. C'est la tour sacrée du *Brahmâ*. Nuit et jour, les prêtres se succèdent sur les marches de l'autel, occupés à transporter la tour de la première aiguille sur la troisième, sans s'écarter des règles fixes que nous venons d'indiquer, et qui ont été imposées par Brahma. Quand tout sera fini, la tour et les brahmes tomberont, et ce sera la fin du monde ! ».

Explication

D'après l'explication du concept, on a pensé de faire la résolution de tour d'Hanoi pour 4 disques et le robot qui va dissoudre le problème posé.

Nous disposons de quatre disques et de trois tours A, B et C. Quatre disques sont ordonnés sur A,de telle sorte que chaque disque ait un diamètre inférieur à son prédécesseur. Ainsi, le plus grand disque de chaque tour se situe à leur base et le plus petit sur leur sommet, les tours de départ B et C sont vides. Comment s'y prendre pour déplacer tous les disques de A vers C s'aidant uniquement de la tour B, tout en respectant les règles cités précédemment ?

L'idée est de faire déplacer les n disques avec $2^n - 1$ déplacements, cela implique 15 déplacements pour une tour de 4 disques.

La figure IV.13 montre bien les séquences des 15 déplacements que le robot doit accomplir.

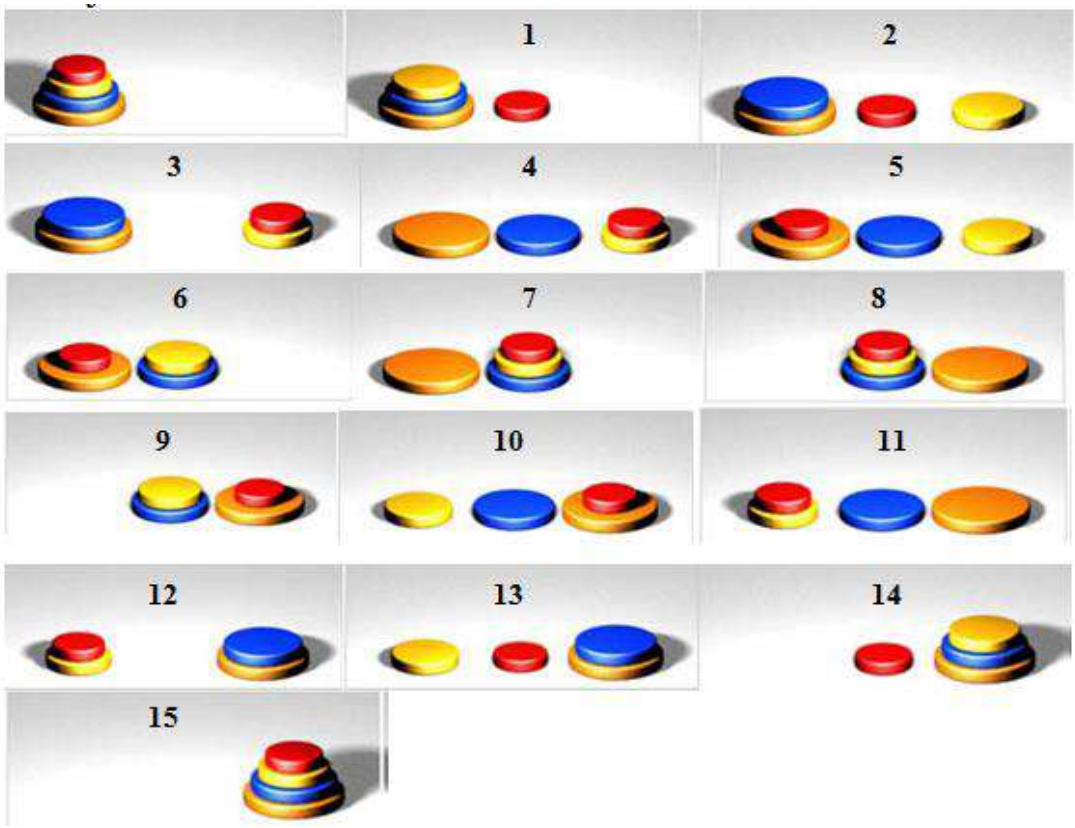


Figure IV.13. Résolution de la tour d'Hanoi à 4 disques.

IV.8.1. Principe de fonctionnement :

Il s'agit d'un problème de déplacements d'objets de différentes tailles entre trois points , pour cela nous avons placé la tour comme ci-montrée dans la figure IV.14 sur la position A. Ensuite, la fixation des repères de points fonctionnels est faite pour les trois positions afin de développer la commande. En prenant en compte les points A, B et C sont rectilignes, donc le robot doit se déplacer linéairement entre ces différentes positions. Dans ce cas là, les coordonnées (x,y) sont fixes pour toutes les positions ,contrairement l'axe z qui change à chaque point .

Pour cela la commande par les instructions « APROACH et DEPART » a été optimal, et qui peuvent être utilisés avec la programmation d'une position en hauteur.

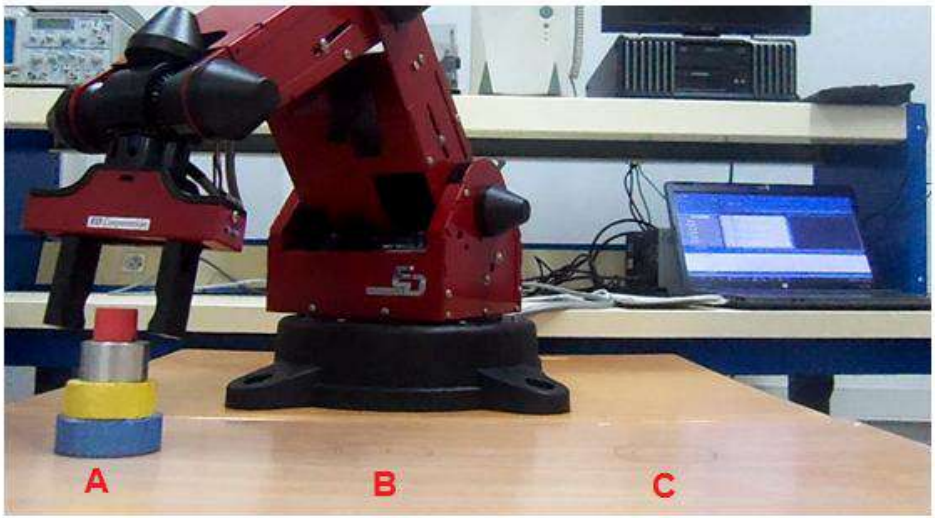


Figure IV.14. L'état initial de la tour.

IV.8.1.1. Instruction d'approche et de départ :

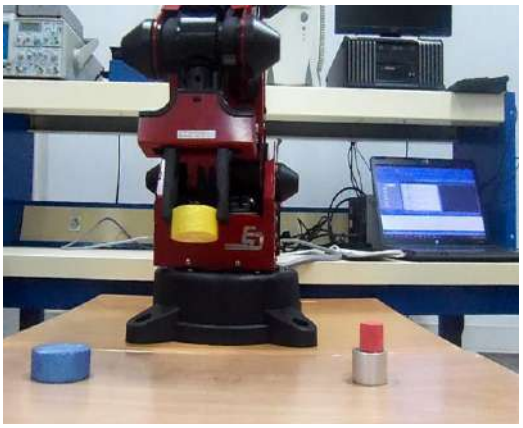
Approché une position cible et prendre un départ d'une position avec une hauteur défini avant de l'atteindre ou après.

IV.8.1.2. Développement du programme de commande

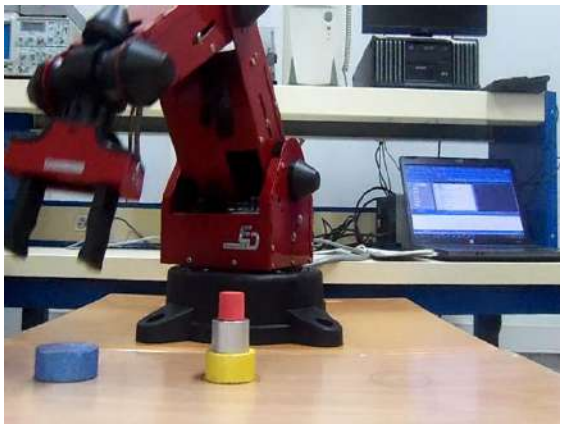
Tableau IV.4. Développement du programme de commande.

Programme	Discussion
PROGRAM	-Début de programme.
SETSPEED 100	- La vitesse
DEF VAR1,VAR2	-Déclaration des variables.
INASIG AIVAR1=AI1	-Enregistrer l'entrée analogique comme variable VAR1
VAR2= (VAR1*20)+40	
GOHOME	-GOHOME : c'est position référence de robot qu'elle ne peut pas changer ou écraser.
MOVETO JPOS1	
OPEN	-Commentaire.
REM 1-D1	
APPROACHTO L,POS1,12,S=100	-Approcher POS1 linéairement avec h=12cm
MOVETO L,POS1,T=5	-Déplacer vers la POS1 dans un temps=5s et attraper la le disque.
CLOSE 70	
DEPART L ,12,S=100	-Quitter linéairement avec h=12cm et une vitesse 100% de sa vitesse max.
APPROACHTO L,POS7,12,S=100	
MOVETO L, POS7,T=5	-Se déplacer linéairement vers la POS7 dans un temps=5s
OPEN	-Poser le disque.
DEPART L,12,S=100	-Quitter linéairement
REM 2-D2	- Commentaire.
IF AIVAR1>2 THEN GOSUB 3D ELSE GOHOME	-Si la condition VAR1>2 est vérifiée aller au sous programme 3D si non revenir à la position HOME
END	
SUB 3D ()	
.	
.	
.	
GOHOME	-En terminant toujours avec GOHOME.
END	

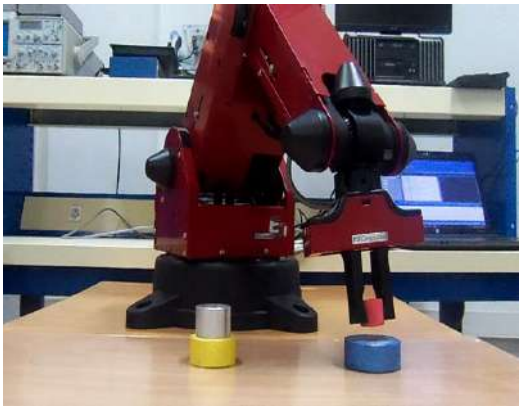
La figure IV.15 donne quelques images descriptives du Robot en cours de la résolution du problème des tours de Hanoi.



(a) :3ème déplacement.



(b) ;entrain de faire 8ème déplacement.



(c) : 10ème déplacement.



(d) :probléme résolus,Robot revenir à la position HOME

Figure IV.15. Des images du robot au cours de résoudre la problématique.

IV.8.2. Conclusion

Suite à plusieurs essais qu'on a faits, nous avons pu réaliser une commande optimale dont le robot peut résoudre la problématique de la tour d'HANOI avec n'importe quel nombre de disques donné.

IV.9. Réalisation d'une cellule de production robotisée avec une Interface Homme-Machine (IHM)

Dans cette partie, nous allons réaliser la cellule de production robotisée, ainsi que suggestion effectuée à base de l'automate programmable GLOFA. Ce projet s'articule sur trois parties distinctes mais complémentaires :

1) La programmation de l'API de gestion de la cellule de fabrication.
2) Programmation du contrôleur robot ED-7255.
3) Elaboration d'un programme pour un pupitre operateur IHM, pour répondre aux exigences et réserves qu'on s'est fixé.

IV.9.1. Développement du projet pour la gestion de la mini-usine

Commençant par l'élaboration et le développement de programme pour la mini-usine qui sera implémenté par la suite dans l'automate ou bien dans le cerveau de notre cellule, afin de gérer tout le processus après avoir faire le câblage nécessaire.

La figure IV.16 présente l'organigramme qui définit la procédure de fonctionnement de l'usine de fabrication étape par étape qui répond aux exigences de notre cahier de charges.

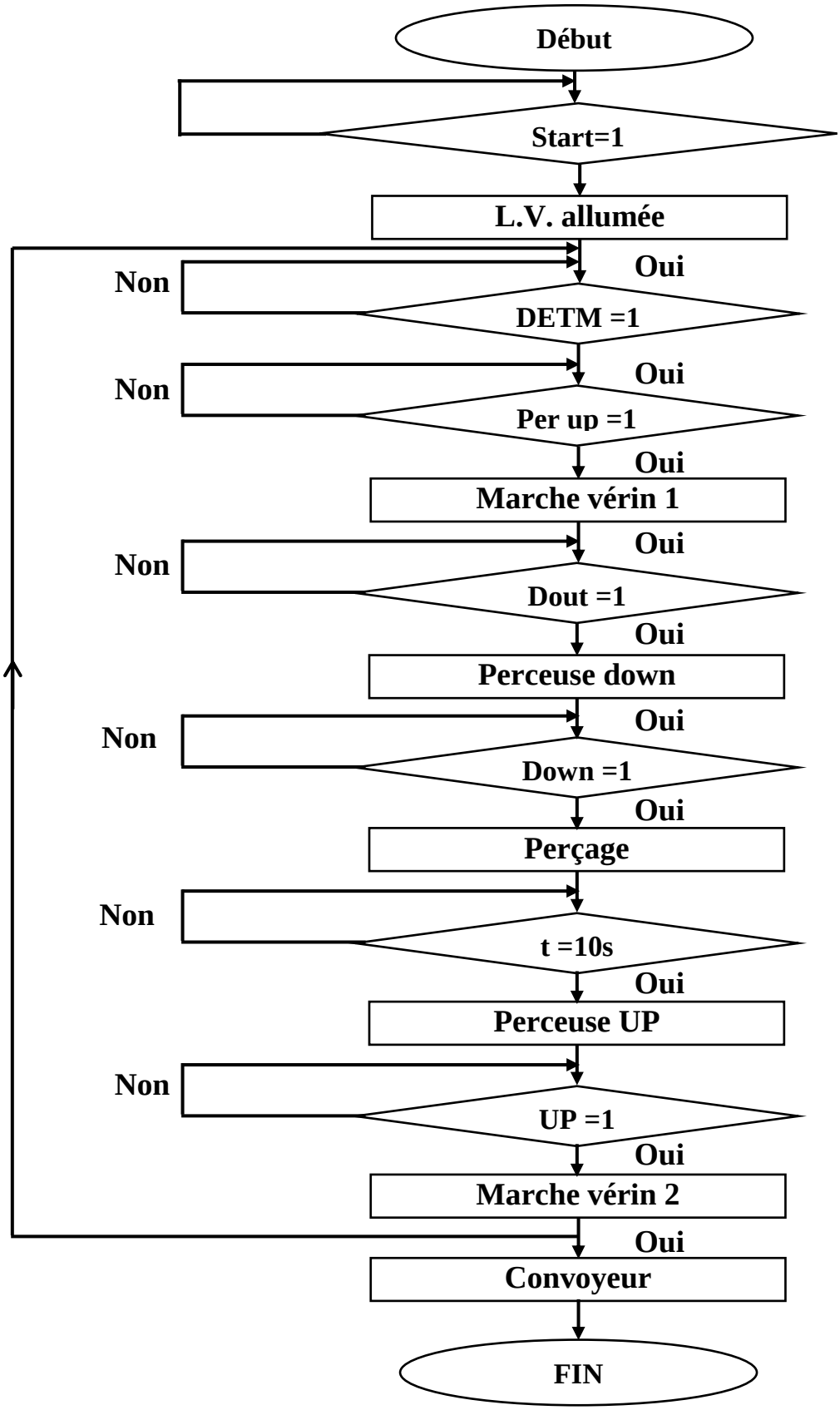


Figure IV.16. Organigramme de la procédure de fonctionnement..

IV.9.2. Création de la table des mnémoniques (Partie Software)

D'après organigramme ci-dessus, il faut définir les variables qui vont être utilisées lors de programmation dans la table des mnémoniques (voir figure IV.17) en donnant une adresse à chacune.

Après avoir définir toutes les mnémoniques, il suffit d'enregistrer pour que les changements soient pris en compte dans le reste du projet.

	Variable Na	Data Type	Memory All	Initial Value	Variable Ki	Used	Comments
1	BPV	BOOL	%IX0.0.0		VAR		
2	DETM	BOOL	%IX0.0.11		VAR		
3	DIN	BOOL	%IX0.0.5		VAR		
4	DOUT	BOOL	%IX0.0.6		VAR		
5	DOWN	BOOL	%IX0.0.9		VAR		
6	GOTO_S1	BOOL	<Auto>		<SFC Go T		
7	LO	BOOL	%QX0.1.2		VAR		
8	LV	BOOL	%QX0.1.0		VAR		
9	PER	BOOL	%QX0.1.7		VAR		
10	PERDOWN	BOOL	%QX0.1.5		VAR		
11	ROBOT	BOOL	%IX0.0.12		VAR		
12	SOR1	BOOL	<Auto>		VAR		
13	SOUT	BOOL	%IX0.0.8		VAR		
14	TIMER	BOOL	<Auto>		VAR		
15	TRANS	BOOL	<Auto>		<SFC trans		
16	UP	BOOL	%IX0.0.10		VAR		
17	VERINDIN	BOOL	%QX0.1.4		VAR		
18	VERINDOU	BOOL	%QX0.1.3		VAR		
19	VERINSOU	BOOL	%QX0.1.6		VAR		
20	VSIN	BOOL	%IX0.0.7		VAR		

Figure IV.17. La table des mnémoniques.

IV.9.3. Réalisation du programme de la gestion de cellules de Production

Dans cette partie, nous allons donner sur la figure IV.18 le programme en langage GRAFCET réaliser afin de satisfaire nos exigences.

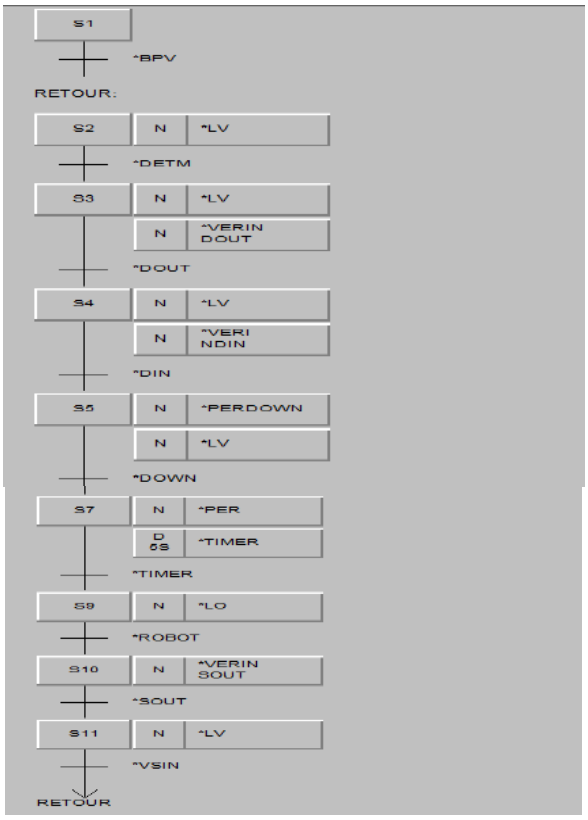


Figure IV.18. Programme en GRAFCET..

IV.9.4. Programme de la commande de Robot

Le développement du programme du robot est fait de telle sorte pour faire différencier entre deux types de pièces, des pièces bien usinée et d'autres pièces sont mal usinée (défectueuse).Ces deux types seront posées sur un poste d'attente de prise(voir figure IV.20). dont elles vont se déplacer et posées sur deux positions différentes, où les pièces défectueuse (métallique) vont être placées dans un chariot afin de les classer , contrairement aux pièces non-défectueuses (noir) qu'elles vont se déplacer et se poser dans un étui à trois emplacements rectilignes (supposé) (voir figure IV.19).

Pour cela nous devions y'avoir les repères des point fonctionnels et les points intermédiaires pour y avoir une trajectoire adapté à l'application et surtout pour éviter les collisions avec le matériel de notre processus.

Suivant la méthodologie de programmation de robot cité précédemment nous avonsdéfinit les différents coordonnées adéquates aux différents points afin de réaliser une manipulation performante. Ces coordonnées sont déjà préenregistrées dans le tableau des positions.

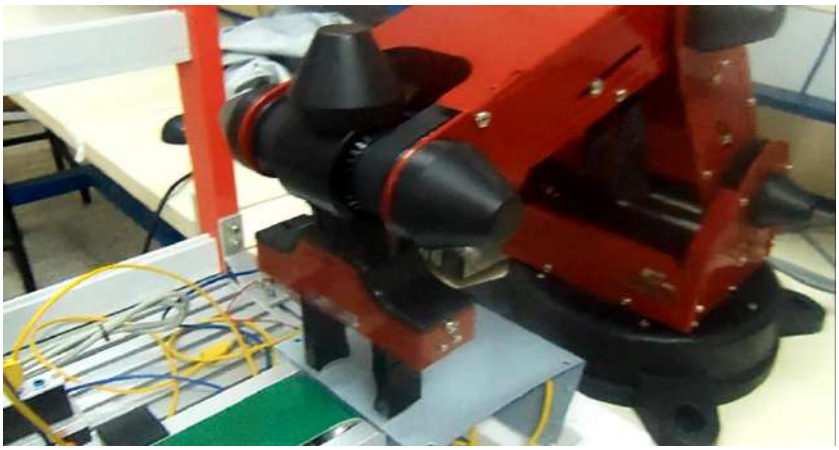


Figure IV.19.Le poste d'attente d'arrivées des pièces.



Figure IV.20.Les deux emplacements P1 (pour la piècedéf et P2 pour la non-def).

IV.10. Elaboration et conception de la commande complète pour la gestion de la cellule robotisée avec une interface de contrôle et de commande Homme-Machine

Dans cette partie nous allons aborder le programme final de notre projet qui s'articule sur la synchronisation entre API, le Robot, et l'IHM,

Comme s'est montrée sur la figure IV.21 la connexion et communication entre l'API qu'il est le noyau qui gère toute la cellule de production en communiquant avec le robot ED-7255 ainsi qu'avec le pupitre operateur.

Dans la réalisation de ce projet et pour atteindre et satisfaire le cahier des charges donné, nous avons fait une étude afin de faire une bonne synchronisation entre l'API, le Robot, et l'IHM où le robot sera fonctionnel à multitâches et l'opérateur aura l'accès de faire la commande, à partir de pupitre, selon la tâche voulue et qui sera exécutée par le robot.

En prenant en considération :

- D’éviter le bourrage des pièces.
- De Minimiser le temps de fabrication.
- D’éviter le maximum de causer des collisions et même des auto-collisions.
- De réaliser une bonne génération de mouvement.

La procédure de fonctionnement de la synchronisation entre les différents éléments est donnée sous forme d’un organigramme descriptif dans la figure IV.22.



Figure IV.21. Connexion et communication entre API ,Contrôleur et l’IHM.

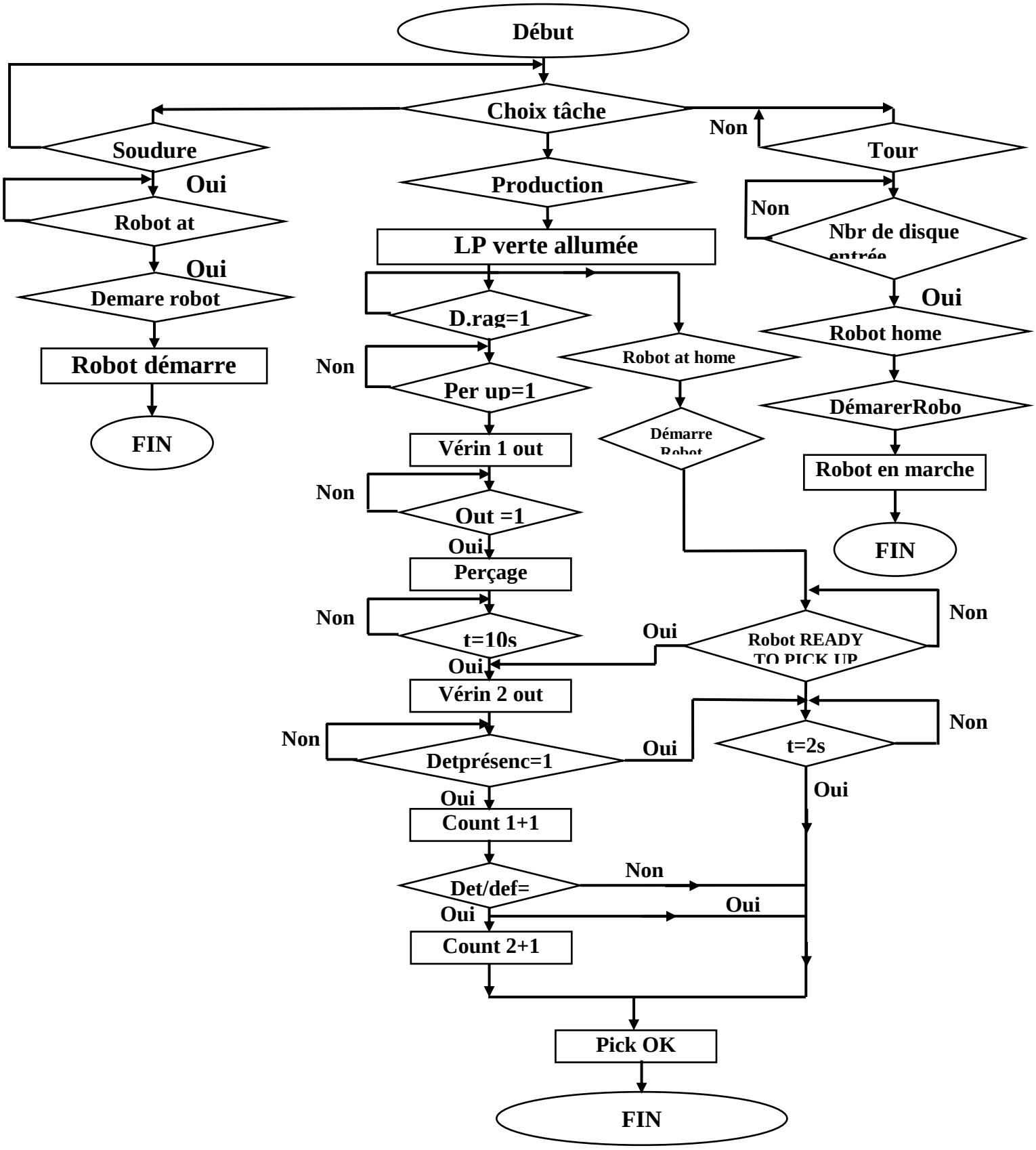


Figure IV.22. Organigramme descriptif de fonctionnement de la synchronisation API, Robot, IHM.

IV.10.1. La commande globale de l'API

Ce programme comporte toutes les conditions d'une synchronisation entre les différents éléments de la cellule de fabrication, comme j'ai dit auparavant, qu'il est le cerveau gérant de fonctionnement général où on trouve des sorties vers le robot et d'autre vers l'IHM comme on trouve des entrées du robot (les entrées/sorties de robot sont celle sur le module numérique de contrôleur) et d'autre de l'IHM, le tout est relié entre eux a pour but de réaliser un bon fonctionnement de la cellule avec l'interface et le robot.

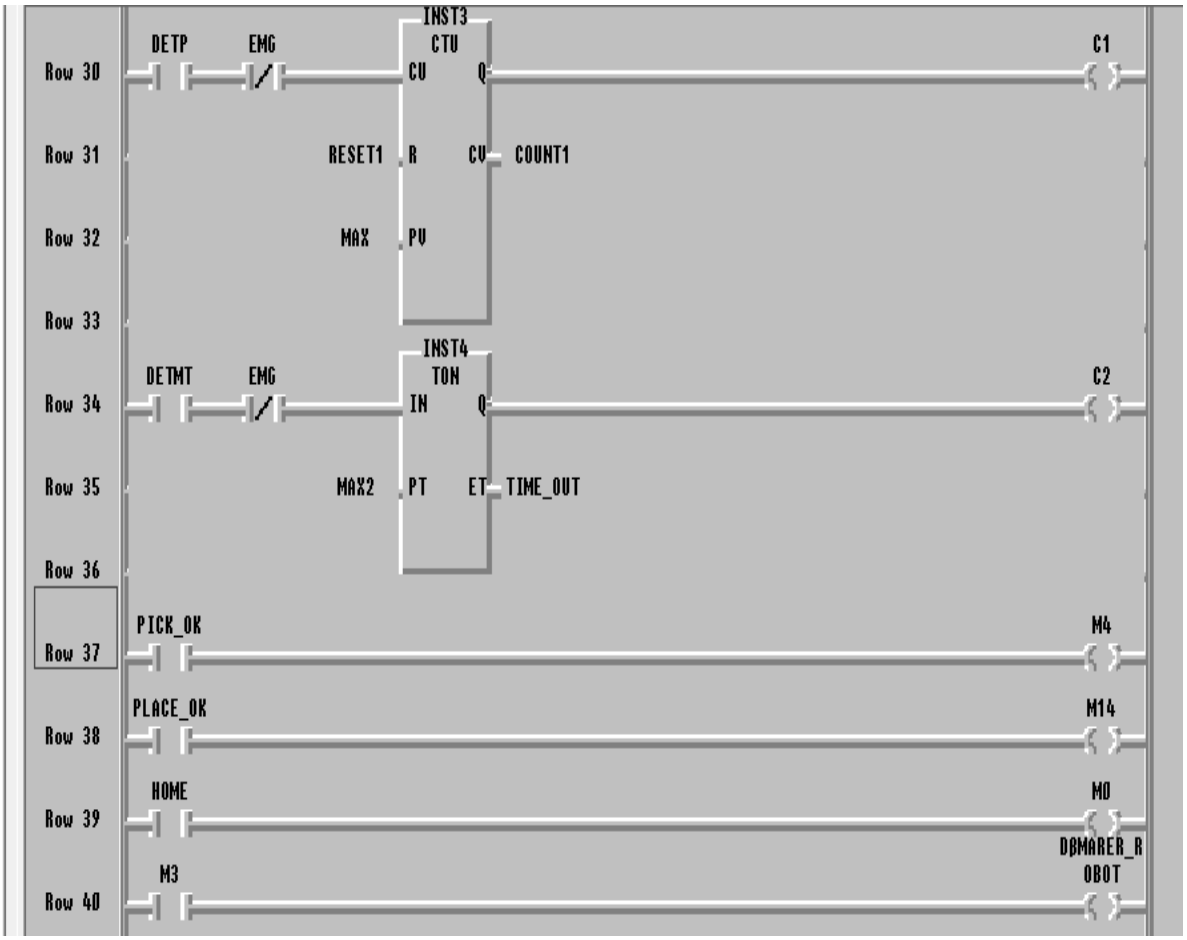
Pour cela j'ai développé ce programme sur la figure dans la suite dont on trouve plus de variables parmi eux qui font partie de la mémoire interne de l'API, pour et d'autre raisons, j'ai réalisé un programme LADDER dont on peut faire des modifications facilement et même c'est plus professionnel et adaptable.

La table des mnémoniques qui comporte toutes les variables, entrée/sortie, dans la figure IV.23 dont on trouve des variable physique et d'autre de la mémoire interne de l'API, comme on trouve des entrées/sorties mots pour l'affichage et recevoir des données à partir de L'IHM

Ensuite, nous allons donner, le programme de quelque tâche réalisées par l'API écrites par le langage de commande LADDER donnée par la figure. IV.24.

	Variable N	Data Type	Memory All	Initial Value	Variable Ki	Used	Comments
1	C1	BOOL	<Auto>		VAR		
2	C2	BOOL	<Auto>		VAR		
3	CONV	BOOL	%QX0.1.8		VAR		
4	COUNT1	INT	%MW9		VAR		
5	DAWN	BOOL	%IX0.0.9		VAR		
6	DEMARER	BOOL	%QX0.1.12		VAR		
7	DETMT	BOOL	%IX0.0.3		VAR		
8	DETP	BOOL	%IX0.0.2		VAR		
9	DISQ	BOOL	%QX0.1.13		VAR		
10	EMG	BOOL	%IX0.0.13		VAR		
11	FF	TIME	<Auto>		VAR		
12	HAN	BOOL	%QX0.1.10		VAR		
13	HN	BOOL	%MX6		VAR		
14	HOME	BOOL	%IX0.0.0		VAR		
15	INST0	FB Instanc	<Auto>		VAR		
16	INST1	FB Instanc	<Auto>		VAR		
17	INST2	FB Instanc	<Auto>		VAR		
18	INST3	FB Instanc	<Auto>		VAR		
19	INST4	FB Instanc	<Auto>		VAR		
20	LL	WORD	%MW10		VAR		
21	M0	BOOL	%MX0		VAR		
22	M1	BOOL	%MX1		VAR		
23	M11	BOOL	%MX11		VAR		
24	M14	BOOL	%MX14		VAR		
25	M15	BOOL	%MX15		VAR		
26	M3	BOOL	%IX0.0.3		VAR		
27	M4	BOOL	%MX4		VAR		
28	MAG	BOOL	%IX0.0.11		VAR		
29	MAX	INT	<Auto>		VAR		
30	MAX2	TIME	<Auto>		VAR		
31	MW15	ARRAY[5] O	%MW15		VAR		
32	MW15Q	WORD	%MW15		VAR		
33	NDISQ	WORD	<Auto>		VAR_EXTE		
34	PERCAGE	BOOL	%QX0.1.7		VAR		
19	INST4	FB Instanc	<Auto>		VAR		
20	LL	WORD	%MW10		VAR		
21	M0	BOOL	%MX0		VAR		
22	M1	BOOL	%MX1		VAR		
23	M11	BOOL	%MX11		VAR		
24	M14	BOOL	%MX14		VAR		
25	M15	BOOL	%MX15		VAR		
26	M3	BOOL	%IX0.0.3		VAR		
27	M4	BOOL	%MX4		VAR		
28	MAG	BOOL	%IX0.0.11		VAR		
29	MAX	INT	<Auto>		VAR		
30	MAX2	TIME	<Auto>		VAR		
31	MW15	ARRAY[5] O	%MW15		VAR		
32	MW15Q	WORD	%MW15		VAR		
33	NDISQ	WORD	<Auto>		VAR_EXTE		
34	PERCAGE	BOOL	%QX0.1.7		VAR		
35	PICK_OK	BOOL	%IX0.0.4		VAR		
36	PLACE_OK	BOOL	%IX0.0.14		VAR		
37	PRODUCTI	BOOL	%QX0.1.9		VAR		
38	READY_PI	BOOL	%IX0.0.12		VAR		
39	RESET1	BOOL	%MX100		VAR		
40	SOUD	BOOL	%QX0.1.11		VAR		
41	SOUDURE	BOOL	%MX7		VAR		
42	START	BOOL	%MX0		VAR		
43	T1	TIME	<Auto>	T#5s	VAR		
44	V1_IN	BOOL	%IX0.0.5		VAR		
45	V1_OUT	BOOL	%IX0.0.6		VAR		
46	V1IN	BOOL	%QX0.1.4		VAR		
47	V1OUT	BOOL	%QX0.1.3		VAR		
48	V2	BOOL	%QX0.1.5		VAR		
49	V2_DAWN	BOOL	%IX0.0.9		VAR		
50	V2_UP	BOOL	%IX0.0.10		VAR		
51	V3	BOOL	%QX0.1.6		VAR		
52	V3_IN	BOOL	%IX0.0.7		VAR		

Figure IV.23. La table des mnémoniques.



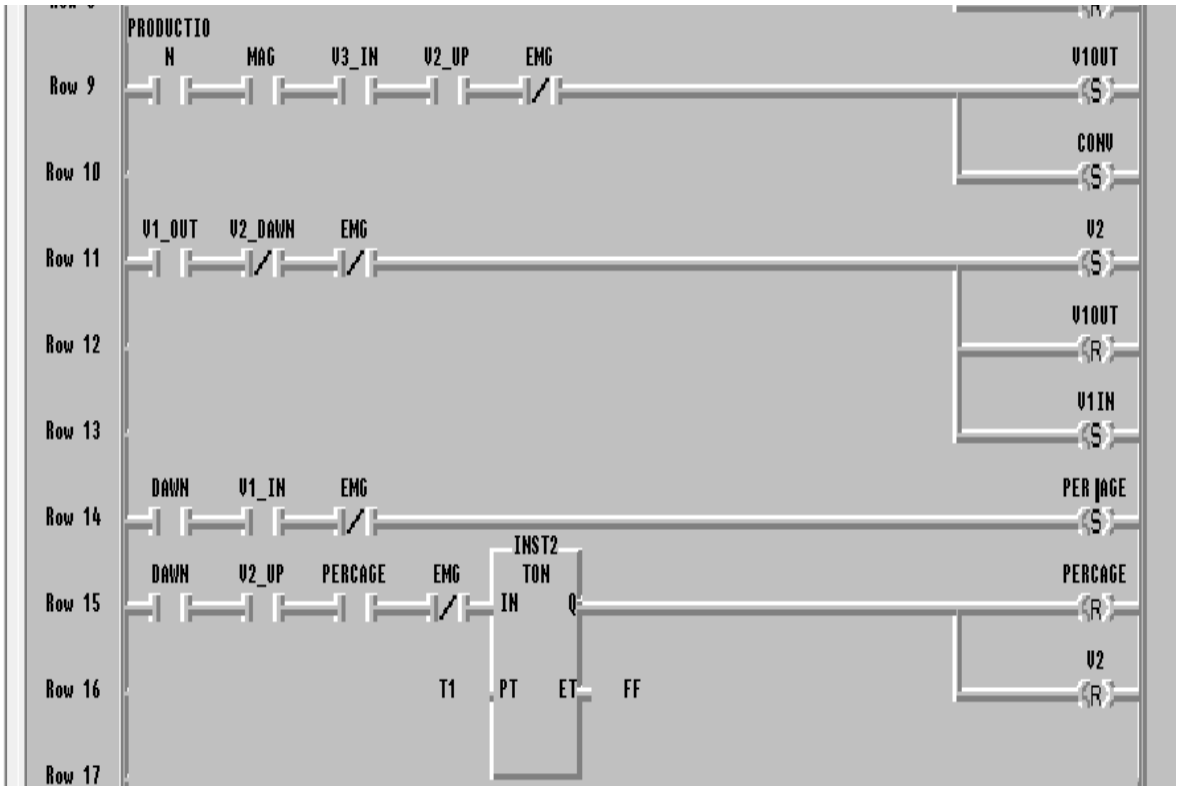


Figure IV.24. Programme LADDER correspond au quelques tâches.

IV.10.2. Programme global de commande du Robot

Le développement du programme global du robot est fait à base des précédents programmes que nous avons fait mais ceci, il est en parallèle avec la programmation del'API pour synchroniser les deux,prenant en compte le temps de perçage, le temps d'arrivé de la pièce et le temps dont le robot soit prêt pour la prise et surtout savoir la qualité de la pièce fabriquée. Ce programme est doté pour que le robot puisse fonctionner selon la commande issue du pupitre et même pour que son fonctionnement apparaisse clairement dans IHM.

Nous allons prendre l'essentiel du programme du robot et l'expliquer à l'aide de commentaire pour comprendre la logique de son fonctionnement.

```
PROGRAM
SET ALL
SETSPEED 100
SETACCEL 100
DEFPOS PO1,POS2,POS3,POS4….
GOHOME
RESET DO2
MOVETO  JPOS8

IF INSIG ON DI3 THEN GOSUB P1
ELSE IF INSIG ON DI4 THEN GOSUB P2 ELSE
IF INSIG ON DI5 THEN GOSUB P3
WAIT INSIG ON DI6
END
```

Cette série d'instruction s'active au démarrage du robot pour des questions de sécurité, on a jugé nécessaire de définir une position de sécurité qu'on a appelé HOME, attendre la commande de la tache à faire et l'ordre de démarrage depuis le pupitre. pour pouvoir après se repositionner au dessus du poste prise pièce (si le choix est pour production), quand il le fait , le robot donne un signal de sortie à l'API pour confirmation Robot prêt pour prise.

```
SUB P1 ()
X1:MOVETO JPOS8 , S=100
RESET DO1
MOVETO JPOS2, T=1
SET ALL
WAIT INSIG ON DI1
WAIT INSIG ON DI2 ,TIMEOUT=1
IFINSIG ON DI2 THEN GOSUB PIECE_DEF ELSE
GOSUB PEICE_MET
L1: MOVETO JPOS8 , S=100
RESET DO1
MOVETO JPOS2, T=1
SET ALL
```

Le robot part à la position suivant le format de mouvement entré au dessus de sa cible jusqu'à ce qu'il a une confirmation d'arrivée de pièce et l'information de type de pièce pour qu'il puisse la déplacer à emplacement correspond.

```
WAIT INSIG ON DI1
WAIT INSIG ON DI2 ,TIMEOUT=1
IFINSIG ON DI2 THEN GOSUB PIECE_DEF
ELSE GOSUB PEICE_MET2
L2:MOVETOPOS8 , S=100
RESET DO1
.
.
GOTO X1
END SUB
SUB PIECE_DEF()
RESET DO5
.
.
OPEN 55
RESET DO4
MOVETO POS5, S=70
GOTO L1
END SUB
SUB PEICE_MET ()
RESET DO5
CLOSE 60
RESET DO3
.
.
.
MOVETO POS3 ,S=100
GOTO L1
END SUB
SUB PEICE_MET2 ()
RESET DO5
.
.
OPEN 55
RESET DO4
MOVETO POS2 ,S=300
GOTO L2
END SUB
SUB PEICE_MET3 ()
RESET DO5
```

Lorsque le robot a une confirmation via API d'arrivée de pièce il l'agrippe et donne une confirmation de prise.

Déplacement du robot avec une pièce défectueuse la position de dépôt.

le robot se déplace à la position de dépôt, dépose les pièces et donne un signal à l'API pour confirmation du bon dépôt des pièceset repart en position prise pièce.

Déplacement du robot avec une pièce non-défectueuse à la position de dépôt.

```
.
.
END SUB
SUB P2 ()
GOHOME
WAIT INSIG ON DI6
INASIG VAR1=DI1
VAR2=(VAR1*20)+40
IF AIVAR1>2 THEN GOSUB 3D  ELSE GOHOME
.
.
GOHOME
END SUB
SUB P3 ()
WAIT INDISIG ON DI6
MOVETO JPOS12
CLOSE
MOVETO L ,P14
.
.
GOHOME
END SUB
```

Si le choix est le tour HANOI donc le robot attend l'entée depuis le pupitre de nombre de disque qu'il existe, et la commande démarrer robot.

Si le choix est soudure pà l'arc donc le robot attend la commande depuis le pupitre de démarrage.

IV.10.3. Elaboration de l'interface homme-machine configurée par ToucheWin

Dans ce projet, Nous avons introduit un nouvel dispositif qui est IHM (Interface-Homme-Machine) en choisissant le type de pupitre sur lequel, les informations seront transmises, pour notre prototype on utilise un panel TH765-N-MT-UT en liaison avec l'automate GLOFA.

Passant par plusieurs logiciels de plusieurs fondateurs (XP BUILDER de LS , PMU-EDITOR de LS , WinCC de Siemens …) pour avoir une compatibilité entre notre automate et l'interface , et après une profonde recherche et plusieurs essais qu'on a fait avec pas mal de configurations à chaque fois, on a trouvé finalement une compatibilité avec le TouchWin ,nous en avons fait les configurations qu'il fallait et même nous avons réalisé le câble RS232 adaptable à l'écran TouchScreen utilisé.

IV.10.3.1. Création des vues de l'interface

Les éléments représentatifs de la station sont utilisés à partir de la bibliothèque de TouchWin et chaque composant est affecté à une variable, une adresse, une représentation et un événement qui lui est assigné. Pour la gestion de ce processus j'ai créé plusieurs écrans de telle sorte l'opérateur ne peut pas seulement visualiser et contrôler le fonctionnement de la cellule robotisée mais il peut aussi faire des commandes selon le besoin.

IV.10.3.1.1. L'écran d'accueil

Cette vue permet la navigation entre les différents écrans du projet et elle contient les touches (voir figure IV.25.) : Production, Visualisation, Commande et Robot dont on a le choix pour accéder aux autres vues .Comme on trouve les deux flèches dans toutes les vues du projet pour aller à la fenêtre suivante ou précédente et l'accès sur le boutons accueil nous ramène toujours à l'écran d'accueil.

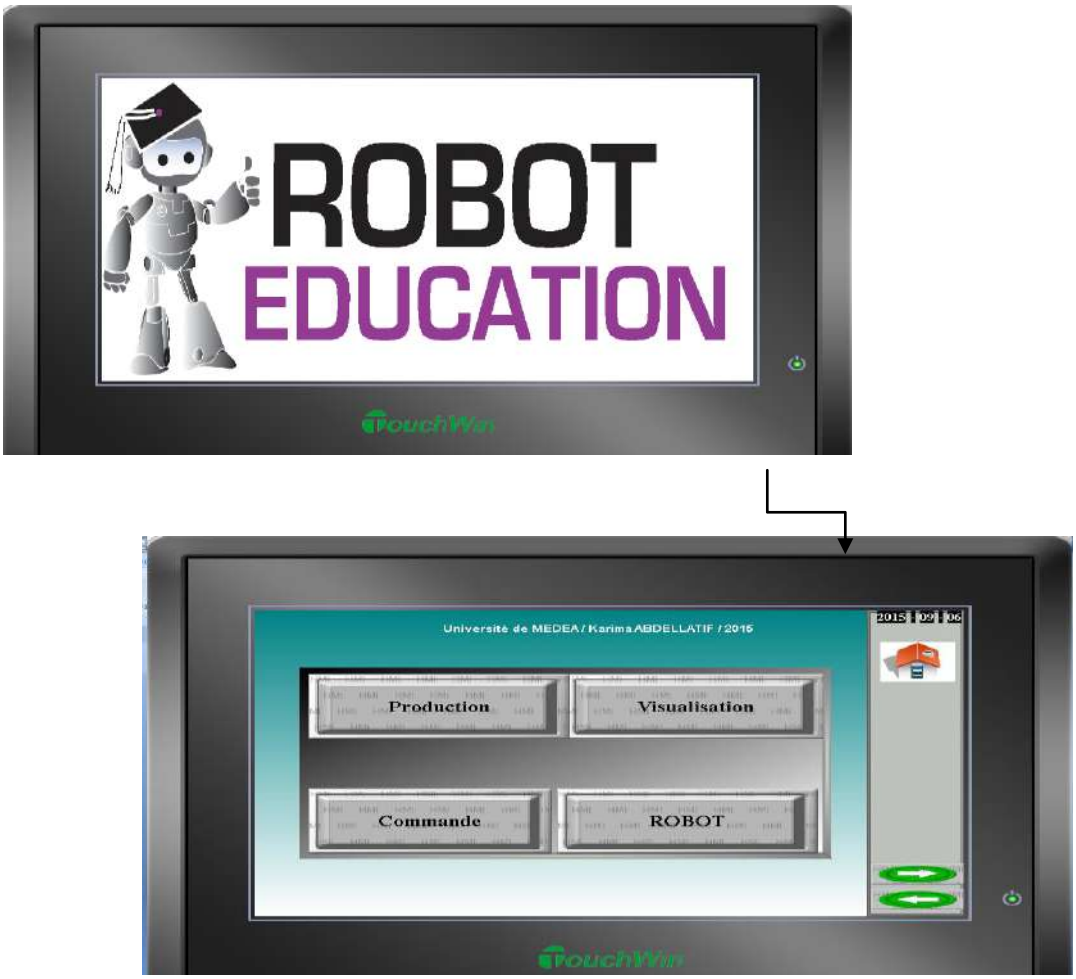
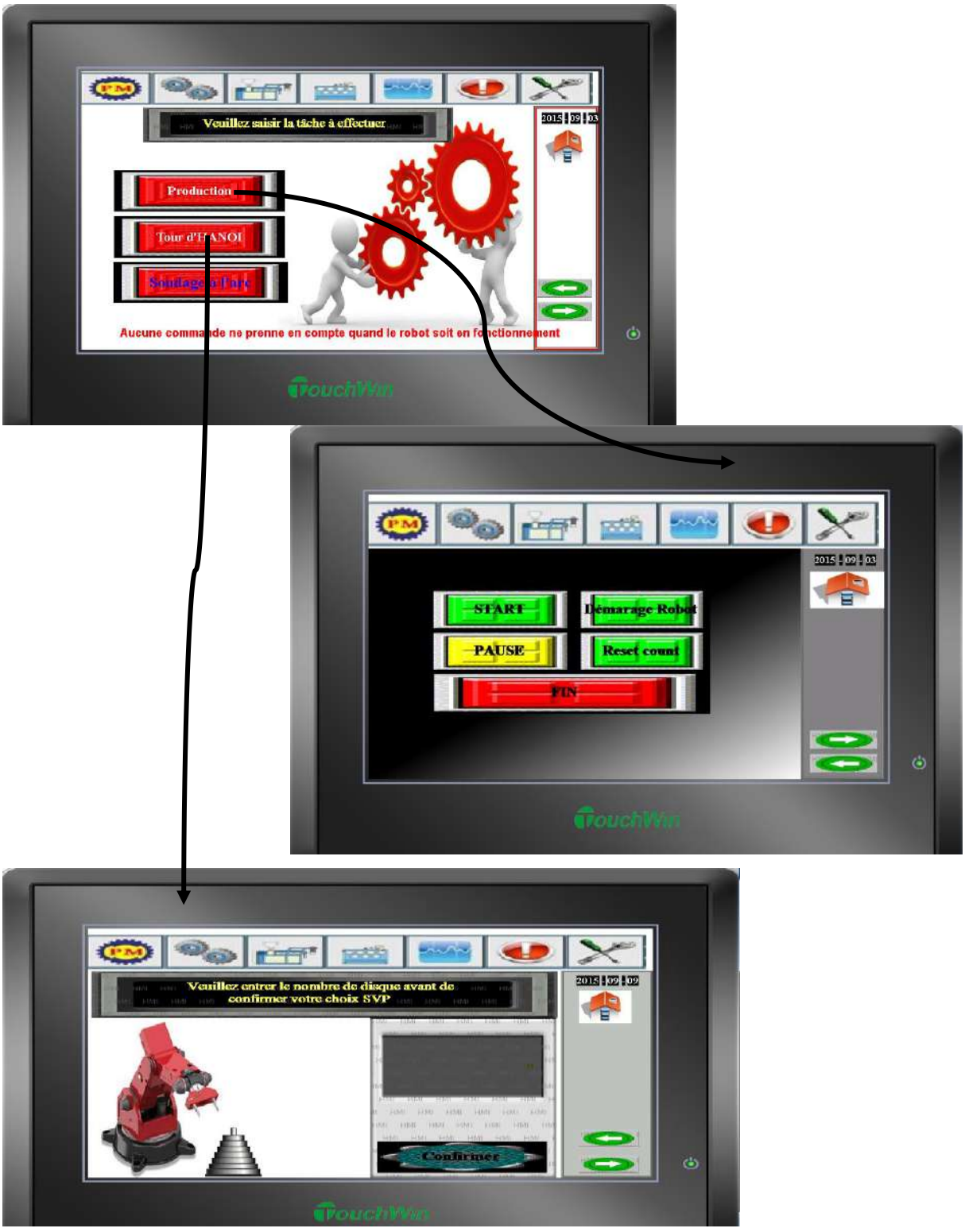


Figure IV.25 Vue d'accueil.

IV.10.3.1.2. La Vue « commande »

En appuyant sur la touche commande, où l'accès à cet opération n'est pas possible à tout utilisateurs que à l'opérateur spécialiste, cetteétape nous donne 3 différents choix pour réaliser une tâche voulueet réalisée par le robot (voir figures IV.26).Sachant que ce dernier est fonctionnel à multi tâches, on trouve sur cet écran les trois manipulations qu'on a réalisé sur le robot: La production qui fait la mise en marche de l'usine, la résolution du problème de la tour d'HANOI , et la soudure à l'arc qui a le même principe que le dessin ,nous avons essayer de valider notre programme pour dessiner mais le problème étais toujours à l'organe terminal (pince)qu'il est destinée à des application limités donc ça n'arrivait pas à agripper un stylo , contrairement que ça marche parfaitement en simulation virtuelle donc l'opérateur aura l'accès de choisir l'une des 3 tâches pour le robot ED-7255, ce qui permet d'obtenir la vue de l'IHM correspondante.



Figures IV.26. Les vues de commande, production, Tours d'HANOI

IV.10.3.1.3. La Vue «Visualisation»

Cette vue est pour la visualisation des états de tous les éléments de la chaine de production, la figure IV.27 montre les différer élément qu'on doit contrôler et même deux afficheurs de compteurs pour voir le nombre des pièces passé.

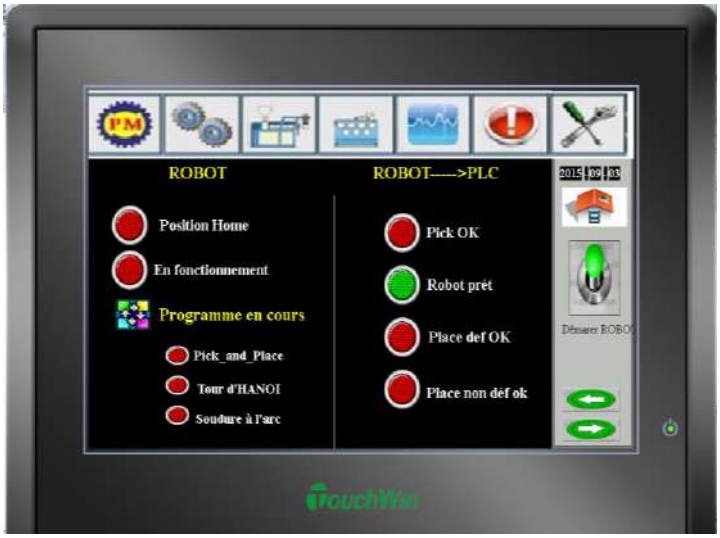


La communication entre l'API et le panel operateur est interactive, par exemple perçage en marche qui correspond a l'entrée QX0.1.7 dans notre API et aussi représenté dans notre pupitre operateur par une lampe qui est rouge lorsque QX0.1.7 est à l'état 0 et verte si elle est à l'état 1.

Figure IV.27. Vue visualisation.

IV.10.3.1.4. La Vue «ROBOT »

Cette vue est pour la visualisation des états des positions de robot au cours de fonctionnement et même on trouve un bouton marche de robot (Figure IV.28).



La communication entre l'API et le panel operateur est interactive, par exemple place def OK qui correspond a la sortie DO4 du contrôleur et une entrée à l'API IX0.0.14 est aussi représentée dans notre pupitre operateur par une lampe qui est rouge lorsque IX0.0.14 à l'état 0 et verte si elle est à l'état 1.

Figure IV.28. Vue Robot.

Voiçi quelques images sur l'ètat de fonctionnement réel de l'IHM après la mise en marche de la producion.



Figure IV.29. Vue « visualisation »



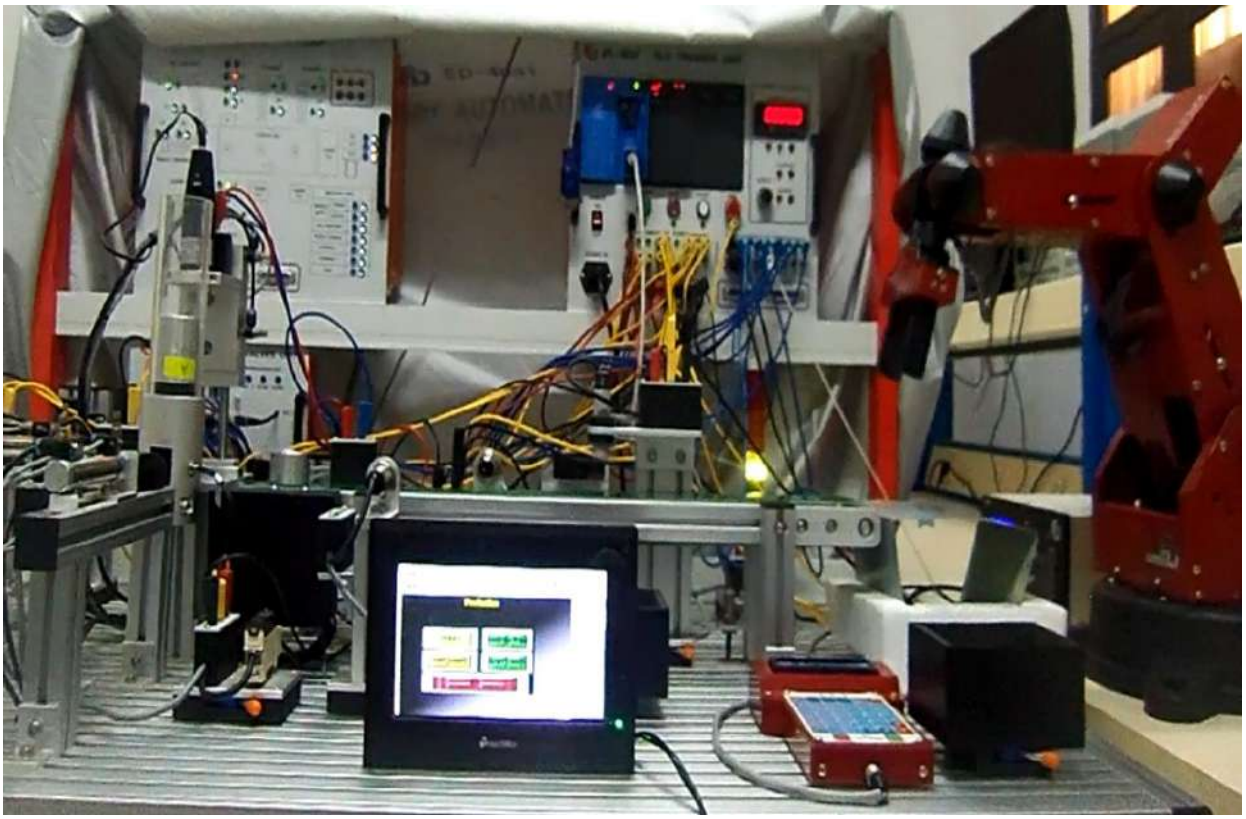
Figure IV.30. Vue « Robot »

au cours de fonctionnement usine/Robot



Figure IV.31. Vue « Tour d'HANOI » avec latâche choisis.

IV.11. Vue générale sur tout le processus en fonctionnement



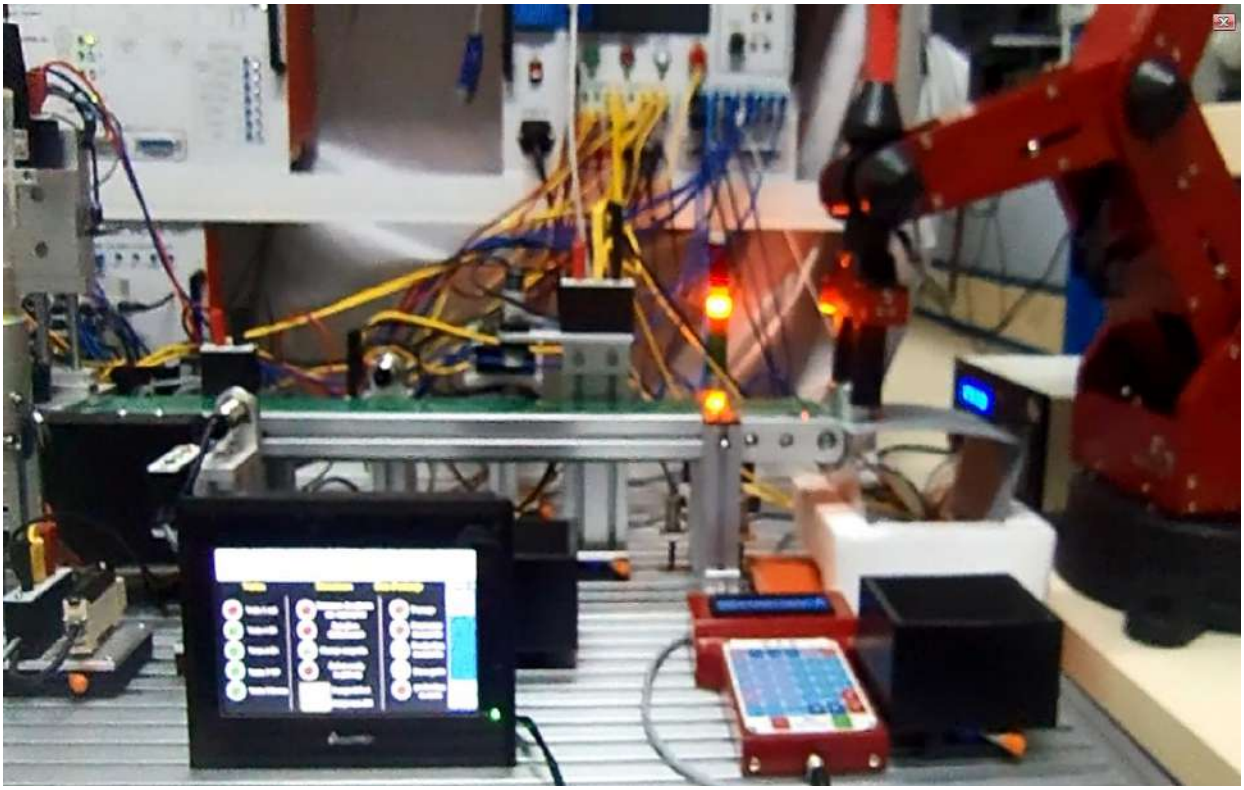


Figure IV.32. Vue générale sur tout le processus.

IV.12. Conclusion

Un contrôle industriel d'un robot et d'une mini-usine est réalisé autour d'une IHM représentée et commandé par un pupitre TouchWin.

Une visualisation de l'état de fonctionnement de la cellule robotisée peut être faite à partir de l'IHM ce qui facilite l'intervention et la communication de l'opérateur au système.

Le choix du type de commande peut être effectué en fonction de l'espace dans lequel est décrite la trajectoire à suivre. La génération d'un mouvement directement dans l'espace articulaire présente des avantages :

- Le mouvement est minimal sur chaque articulation,
- Elle nécessite moins de calculs (pas de passage MGI) alors que le temps d'exécution.
- Le mouvement n'est pas affecté par le passage sur les configurations singulières,
- Les contraintes de couples maximum et de vitesse maximum sont connues car on les fixe aux limites physiques des actionneurs.

En contrepartie, la géométrie du déplacement du robot dans l'espace opérationnel n'est pas contrôlée (bien qu'elle soit répétitive). Il y a donc risque de collision si l'environnement est encombré. On en déduit que ce type de mouvement conviendra d'avantage lorsque l'espace sera dégagé, et pour des déplacements rapides.

Lorsque la géométrie de la trajectoire doit être contrôlée, la génération de trajectoire dans l'espace opérationnel sera préférée. Par contre, elle comporte un certain nombre d'inconvénients :

- Elle demande d'appliquer le MGI en chaque point de la trajectoire,
- Elle peut être mise en défaut lorsque la trajectoire calculée passe par une position singulière,
- Elle peut être mise en défaut lorsque la trajectoire calculée fait passer une articulation hors de ces limites de variation $q[q_{min}; q_{max}]$,
- Les limites C_{max}, V_{max} étant définies dans l'espace opérationnel ne permettent pas d'utiliser au mieux les actionneurs.

Le choix dépend essentiellement de la tâche à effectuer et de l'espace de travail.

Conclusion générale

Des connaissances théoriques et pratiques dans le domaine de la robotique étaient nécessaires pour aborder le thème à étudier.

En effet, ce projet de réalisation nous a été une expérience très fructueuse, bénéfique et enrichissante. Il nous a permis d'exploiter les différentes notions théoriques vues durant notre période d'étude. Il nous permet également la maitrise de la commande des robots à travers leurs propres langages , et ça nous a permis aussi bien de mettre le premier pas dans le monde pratique de la robotique industriel et faire des interconnections entre l'événement extérieur.

Sur le plan software, ce projet nous a mis directement en contact avec des langages de programmation puissants, ainsi que différents logiciels spécialisés dans le domaine industriel.

D'après les différentes manipulations réalisées, on a pu constater que Le logiciel de simulation robotique permet de programmer une trajectoire complexe hors-ligne, mais il ne représente pas parfaitement le comportement réel du robot. En effet, d'une façon générale, les phénomènes sont modélisés de façon approchée pour accélérer les calculs.

Des divergences peuvent donc être constaté entre le temps cycle observé et celui attendu. Les points suivants peuvent être cités comme phénomènes constatés

- Les modèles CAO des robots sont théoriques, des écarts entre le robot réel et le virtuel sont inévitables (axes, positionnement, initiation des moteurs, etc.).
- Les frottements, les flexibilités des bras de robots, les non linéarités des transmissions, etc. Ces erreurs non géométriques dépendent de la charge, de la vitesse et des accélérations mises en jeu lors de déplacement. Elles peuvent causées des divergences dynamique entre la simulation et la réalité.

Il est donc recommendé de controler les trajectoires de robot en temps réel pour garantir la qualité d'élaboration du dépôt.

En fin nous espérons que notre mémoire soit une bonne référence aux étudiants qui vont utiliser ce robot et espérons que notre travail va être amélioré en futur. Comme perspectives nous proposons :

- Commander le robot par automate programmable
- La commande du bras à distance via *Internet*.
- La commande en 3D par des caméras.

Bibliographie

[1] W.Khalil ,E.Dombre , Modélisation, identification et commande des robots,2eme édition ,France , Hermés Science , 1999.

[2]N.Salhi , R.Salhi , Réalisation de carte à microcontrôleur pour le contrôle de bras manipulateur via un pc , mémoire de fin d'étude , université de Biskra ,2010

[3] Sihao Deng, Programmation robotique hors ligne et controle en temps reel des trajectories: Développement d'une extention logiciel de RobotStudio.TM pour la projection thermique , thése de doctorat , université de technologie de Belfort-Montbéliare , Ecole doctoral « Sciences Physiques pour l'ingénieur et Microtechniques », 2006

[4] O.Kollache , Commande de robot manipulateur « SEIKO-200 » par langage de programmation robotique. Application au chargement /déchargement des machines-outils à C.N , mémoire de fin d'étude , université de Médéa , 1996

[5] : O.Ahcene, NEMMICHE .B, BELHAMITI .H.A , BERRAHO .I , « Commande en 3D d'un bras robot à 5 DL par positionnement en temps réels par deux caméras », Université ABDELHAMID IBN BADIS , Mostaganem 2013.

[6] T.Chateau , élément de robotique ,vol. 107, Université Blaise Pascal ,2013

[8] Hayet Iddir, Etude et réalisation d'un kit électronique du type système embarqué à base de microcontrôleur application au control d'un robot à 4 degrés de libertés , mémoire de fin d'étude, université de M'hamed Bouguerra , boumerdes , 2013.

[9] T.C. Manjunath , Kinemaic modeling and manauvring of A5-axes articulated robot arm, International Journal of Mechanical, Aerospace, Industrial, Mechatronic and Manufacturing Engineering Vol:1, No:4, 2007

[10] Mus7tapha Jabbar Hayawi, analutical inverse kinimatics algorithm of A5-dof robot arm, Journal of education of college,vol. 1 , Thi-Qar University . no.4 vol.1 march./2011

[11] M.A.Helal,A.Belhadj , controle par automate programmable industriel d'un prototype d'une mini-usine , mémoire de fin d'étude , université de médéa , 2014

[12] L. BERGOUGNOUX , A.P.I. Automates Programmables Industriels , POLYTECH' Marseille , 2004–2005

[13] H.Khelil Cherfi , A.Yousfi , Conception et réalisation d'un robot (Bras manipulateur) 3 degrés de liberté minimum , mémoire de fin d'étude , université de Médéa , 1995.

[15] Prof. Akram Hossain, Hmi design: an analysis of a good display for seamless integration between user ub=nserstandind and automatic controls. Purdue University, Calumet ,2012.

[17] A.C.Razem ,Conception et realisation d'une ligne de conditionnement robotisé , mémoire de fin d'étude ,université de Médéa, 2013

Rapports techniques:

[7] ED-Corporation , 5-axis arm robot trainer , Sanngdaemon-Dong , Seongnam-City, KOREA , 517-15

[16] TounchWin , TH series Touch Screen ,Xinje Automation Co,

[14] GMWIN, LS Indestriel systeme, T solution leader and electrics and automation , 1994-2002

ANNEXE

- **Caractéristiques technique des principaux composants du controleur.**

Function	Item	Spec	수량
ARM Process	CPU(6410)	400MHz, Min	1
	FLASH	32MByte	
	SRAM	1MByte(32bits Access)	
	DPRAM	16K or 32k	
DSP	TMS320F2811	Digital In(8 Points)	1
		Digital Out(8 Points)	
		Analog In(8-Ch)	
	Motion Control	Closed loop operation	
		Available motor(max 8-axis)	
CPLD	Available PWM Generation	8 axis	1
	Absolute encoder counter	8-ch(PWM method)	
Driver Pack	Operating environment	24V, 120W Max	8
Interface	USB	USB 1,1 A-Type	1
		USB 1,1 B-Type(Host PC)	1
	Ethernet	10Base-T	1
	RS-232		1
	RTC	RTC 4513 Real Time Clock	1
	State LED	Power(3 color), Alarm, Start/stop	3

ANNEXE

- **Data sheet de Robot ED-7255**

> SPECIFICATIONS

ARM PROCESSOR

- CPU
  - » 666MHz, Min : 1ea
- NAND FLASH
  - » 128MByte : 1ea
- NOR FLASH
  - » 1MByte : 1ea
- DRAM
  - » 128MByte : 1ea
- DPRAM
  - » 128k : 1ea

DSP PROCESSOR

- TMS320F2811
  - » 150MHz : 1ea
- Motion Control
  - » Closed loop operation : 1ea
- Avaliable motor 8-axis
  - » CPLD : 1ea
- Avaliable PWM Generation
  - » 8 axis : 1ea
- Absolute encoder counter
  - » 8 channels : 1ea

ROBOT BODY

- Width : 280mm
- Depth : 280mm
- Height : Max. 786mm
- Weight : 12.7kg
- Operating Part

Link Number	Operating Range	Link Length	Weight
No. 0(Base)	0	80mm	5.9kg
No. 1	-170~+170°	107mm	3.4kg
No. 2	-30~+94°	230mm	3.3kg
No. 3	-0~+130°	230mm	1.9kg
No. 4	-110~+110°	0mm	0.6kg
No. 5	-110~+110°	0mm	0.05kg
No. 6(Gripper)	0 ~ 80mm (Rubber Pad)	139mm	0.35kg

TEACH PENDANT

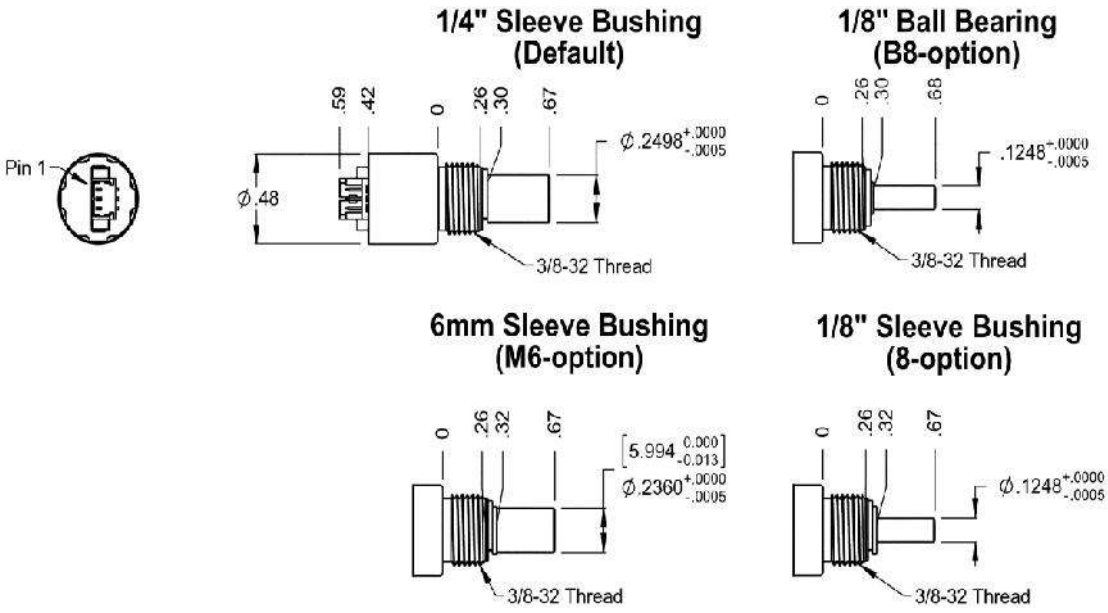
- Display : 20 x 2 Character LCD
- Interface : USB 1.1
- Key : 37 user key(Emergency switch included)

ROBOT CONTROLLER DIMENSION

- Width : 250mm

ANNEXE

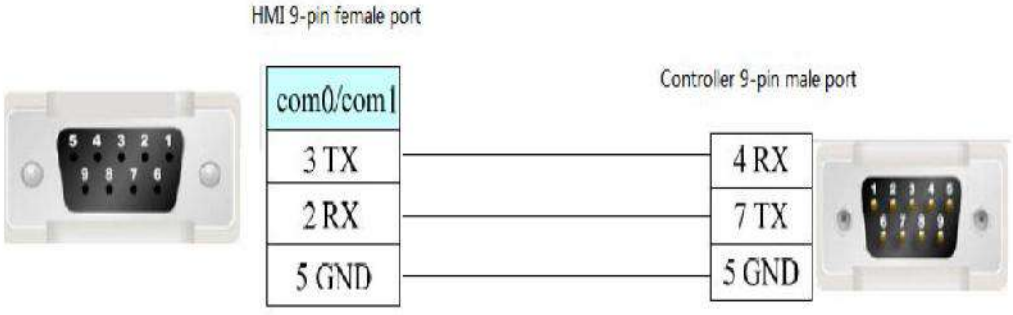
- **Data sheet de l'encodeur**



- **Configurations pour la compatibilité Interface-API**

Parameters	Recommend settings	Choices of settings	Note
PLC type	LG **Glofa (cnet)**		
Port	RS232	RS232	
Data bit	8	7 or 8	
Stop bit	1	1 or 2	
Parity	No parity	Even/odd/no parity	
Baud rate	19200	4800/38400/9600/115200/19200/187500	
Station no.	0	0~255	

- **Réalisation de câble pour la compaptibilité**



ANNEXE