

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
UNIVERSITE MOULOUD MAMMERI DE TIZI-OUZOU



FACULTE DU GENIE ELECTRIQUE ET D'INFORMATIQUE
DEPARTEMENT D'ELECTRONIQUE

Mémoire de Fin d'Etudes De MASTER PROFESIONNEL

Domaine : **Sciences et Technologies**
Filière : **Electronique**
Spécialité : **Electronique industriel**

Présenté par

SADOUN Ferhat
IDDIR Massinissa

Thème

Conception et réalisation d'un bras robotique à 4 degrés de liberté

Mémoire soutenu publiquement le /07/2024 devant le jury composé de :

Mr ATTAF Youcef
President, M.C.B, UMMTO

Mr Lahdir Mourad
Encadreur, Professeur, UMMTO

Mr IDJERI Boussad
Examineur, M.C.B, UMMTO

Année universitaire 2023-2024

Remerciements

Tout d'abord, nous tenons à exprimer notre gratitude à Dieu tout-puissant pour le courage et la volonté qui nous ont permis de réaliser ce travail.

Nous souhaitons également remercier chaleureusement Monsieur M. Lahdir, notre encadreur, pour son aide précieuse et ses conseils avisés tout au long de ce projet. Nos remerciements vont également à tous nos professeurs, pour leur soutien et leur accompagnement durant notre parcours universitaire.

Un grand merci aux membres du jury pour l'honneur qu'ils nous font en acceptant d'examiner notre travail et pour leurs précieuses remarques et suggestions.

Nous adressons nos remerciements les plus sincères à toutes les personnes qui ont contribué à la réalisation de ce projet et qui nous ont aidés à surmonter les nombreuses difficultés rencontrées durant la préparation de ce mémoire.

Enfin, nous exprimons notre profonde gratitude à nos parents et à nos familles pour leurs sacrifices, leur aide, leur soutien indéfectible et leurs encouragements constants.

Dédicace

Je dédie ce modeste travail à ceux qui m'ont aidé de près ou de loin pour sa réalisation, en particulier à :

❖ **Ma famille :**

- Mes très chers parents.
- Mes trois sœurs : "Zira", "Tina" et "Fifi".
- Mon oncle, A. Si Larbi.

❖ **Mon binôme :** M. IDDIR et sa famille.

❖ **À mes deux camarades :** K. Menia et K. Mahiout.

❖ **À tous mes amis** avec qui j'ai partagé ces années d'études.

❖ **À tous mes camarades de la promotion en électronique industrielle**, sans exception.

Ferhat

Sommaire

Introduction	1
--------------------	---

Chapitre I : Généralités sur la robotique.

I.1 Préambule.....	2
I.2 Définitions de la robotique	2
I.3 L'histoire de la révolution de la robotique et des robots.....	3
I.3.1 Les trois lois de la robotique.....	4
I.4 Types de robots.....	5
I.4.1 Robots mobiles	5
I.4.2 Robots manipulateurs.....	6
I.5 Structure fonctionnelle d'un robot.....	6
I.6 Domaine d'application des robots.....	7
I.6.1 Domaine de l'industrie.....	7
I.6.2 Domaine militaire.....	9
I.6.3 Domaine de la santé.....	9
I.6.4 Domaine domestique.....	10
I.7 Structure générale d'un système robotique.....	11
I.7.1 Système mécanique articulé.....	11
I.7.2 Actionneur : (unité moteur)	12
I.7.3 Capteur (organe de détection)	16
I.7.4 Effecteurs (organes terminaux)	16
I.7.5 Système de traitement (ordinateur)	17
I.8 Classification des robots	17
I.8.1 Classification fonctionnelle.....	17
I.8.2 Classification géométrique.....	20
I.9 Avantages et Inconvénients des robots industriels.....	22
I.9.1 Les avantages.....	23
I.9.2 Les inconvénients.....	24
I.10 Discussion.....	24

Chapitre II : Conception matérielle et logiciel.

I.1 Préambule.....	25
II.1 L'architecteur d'un bras robotique.....	25
II.2 Les caractéristiques d'un bras robotique.....	26
II.3 Conception du système de commande du bras robotique.....	26
II.4 La commande manuelle.....	26
II.4.1 Joystick pour Arduino.....	27
II.4.2 La carte Arduino.....	28
II.4.3 Le servomoteur.....	32
II.4.4 Pilotage d'un servomoteur avec Arduino.....	33
II.4.5 L'articulation du bras robotique.....	34
II.4.6 Alimentation.....	36
II.5 La commande automatique.....	36
II.5.1 Séquence de commande.....	37
II.5.2 La carte ESP32.....	37
II.5.3 Pilotage d'un servomoteur avec la carte ESP32.....	41
II.6 La partie logicielle.....	41
II.6.1 Interface de logicielle.....	41
II.6.2 Le menu fichier.....	42
II.6.3 Les boutons.....	43
II.6.4 Le moniteur série.....	43
II.6.5 Transfert des programmes vers la carte Arduino.....	44
II.7 Langage HTML.....	46
II.7.1 Les Balises HTML.....	46
II.8 Serveur Web.....	47
II.8.1 Fonctionnement d'un serveur web.....	48
II.9 Discussion.....	49

Chapitre III : Réalisation pratique.

III.1 Préambule	50
III.2 Les éléments fondamentaux requis pour le bon fonctionnement du bras robotique.....	50
III.3 Montage du bras.....	51
III.4 Câblages des servomoteurs et des joysticks avec la carte Arduino.....	54
III.5 Câblages des servomoteurs et carte ESP32.....	55
III.6 La commande manuelle.....	56
III.6.1 Description.....	56
III.6.2 Structure du programme.....	56
III.7 La commande automatique.....	59
III.7.1 Description.....	59
III.7.2 Structure du programme.....	59
III.8 L'organigramme de la commande manuelle.....	64
III.9 L'organigramme de la commande automatique.....	66
III.10 Les actions de tests du bras.....	67
III.11 Discussion.....	71
Conclusion.....	72
Bibliographie.	

Liste des Figures

Figure I.1 : L’histoire de la robotique	4
Figure I.2 : Robot manipulateur	5
Figure I.3 : Bras manipulateur	6
Figure I.4: Structure fonctionnelle d’un robot	6
Figure I.5 : Robot industriel	8
Figure I.6 : Robot industriel en domaine militaire.....	9
Figure I.7 : Chirurgie robotiques.....	10
Figure I.8 : Robot domestique	11
Figure I.9: Le fonctionnement d’un moteur à courant continu	12
Figure I.10 : Réducteur en engrenage.....	13
Figure I.11 : Moteur pas à pas	13
Figure I.12 : Vérin simple effet	14
Figure I.13 : Vérin double effet	15
Figure I.14 : Actionneur hydraulique.....	15
Figure I.15 : Bras a commande manuelle	18
Figure I.16 : Rras à commande automatique	19
Figure I.17 : Robot programmable	19
Figure I.18 : Robot intelligent(humanoïde).....	20
Figure I.19 : Structure cartésienne	20
Figure I.20 : Structure cylindrique (RPP) ou (PRP)	21
Figure I.21 : Structure sphérique	21
Figure I.22 : Structure dite SCARA.....	22
Figure I.23 : Structure 3R (anthroposophe).....	22

Figure II.24 : Les éléments constitutifs d'un robot.....	25
Figure II.25 : la commande manuelle.....	27
Figure II.26 : Le Joystick pour Arduino.....	27
Figure II.27 : Les axes du joystick.....	28
Figure II.28 : La carte Arduino Uno.....	29
Figure II.29 : Structure de la carte Arduino.....	29
Figure II.30 : Microcontrôleur ATMEL ATMega328.....	31
Figure II.31 : Servomoteur 9g.....	32
Figure II.32 : Les composants du servomoteur.....	33
Figure II.33 : Pilotage d'un servomoteur par la carte Arduino.....	34
Figure II.34 : Alimentation 12 V/2A.....	36
Figure II.35 : La commande automatique.....	36
Figure II.36 : Carte de développement ESP32 Wroom.....	37
Figure II.37 : Fonctionnalités de la carte ESP32.....	38
Figure II.38 : Les Broches de la carte ESP32.....	39
Figure II.39 : Pilotage d'un servomoteur par la carte ESP32.....	41
Figure II.40 : Présentation de la fenêtre IDE Arduino.....	41
Figure II.41 : Contenu de menu fichier.....	42
Figure II.42 : Présentation des boutons.....	43
Figure II.43 : Le moniteur série.....	44
Figure II.44 : La console IDE Arduino.....	44
Figure II.45 : Le téléversement du programme vers la carte Arduino.....	45
Figure II.46 : Exemple d'un document HTML.....	46
Figure II.47 : L'architecteur client-serveur.....	48
Figure III.48 : Câblages de la commande manuelle.....	55
Figure III.49 : Câblage de la commande automatique.....	55
Figure III.50 : Algorithme de fonctionnement de la commande manuelle.....	64
Figure III.51 : Algorithme de fonctionnement de la commande automatique.....	66
Figure III.52 : Position initiale.....	67
Figure III.53 : position de bras allongé.....	68

Figure III.54 : position de la rotation du bras.....	68
Figure III.55 : Position du robot pendant qu'il lâche l'objet.....	69
Figure III.56 : Enregistrement et exécution de la commande automatique.....	70

Liste des tableaux

Tableau II.1 : Les articulations du bras robotique.....	34
Tableau II.2 : Les articulations du bras robotique.....	34
Tableau II.3 : Les balises de langage html.....	46
Tableau III.4 : Les étapes de montage du bras robotique.....	53

Introduction

Introduction

La robotique a véritablement révolutionné notre façon d'appréhender la technologie et l'automatisation. Imaginez des bras robotiques capables d'accomplir une multitude de tâches complexes, simplifiant ainsi nos vies et boostant l'efficacité de nombreuses industries. C'est dans ce contexte fascinant que s'inscrit notre projet de fin d'études, où nous plongeons au cœur de la conception et de la réalisation d'un bras robotique à 4 degrés de liberté.

Dans cette aventure captivante, nous nous attelons à comprendre les bases de la robotique, à concevoir chaque pièce de ce bras robotique et à le rendre pleinement fonctionnel. Notre objectif est d'explorer ce domaine en constante évolution, en embrassant une approche méthodique et rigoureuse. Ce projet représente bien plus qu'une simple exploration technique. C'est une immersion dans les défis et les innovations de la robotique, avec pour ambition de créer un bras robotique performant et adaptable, prêt à répondre aux besoins changeants de notre monde en constante évolution.

Pour réaliser ce travail, nous avons divisé notre mémoire en trois chapitres.

Dans le premier chapitre, nous allons détailler les fondements de base de la robotique, son histoire, ses principes de base et ses applications actuelles. Comprendre ces éléments est crucial pour saisir les défis et les opportunités que nous rencontrons dans la création d'un bras robotique.

Dans le deuxième chapitre, nous décrivons la phase de conception du bras robotique. Nous parcourons chaque étape, de l'analyse des besoins à la sélection des composants, en passant par la modélisation et la simulation, afin de créer un plan solide pour notre projet.

Dans le dernier chapitre, nous mettrons en œuvre notre plan, en réalisant concrètement le bras robotique. De la fabrication des pièces mécaniques à la programmation des contrôleurs, nous détaillerons chaque étape pour garantir le bon fonctionnement et la fiabilité de notre création.

Nous terminons notre mémoire par une conclusion ainsi que par des perspectives pour améliorer notre travail.

Chapitre I

Généralités sur la robotique

I.1 Préambule

La robotique est un domaine technique qui fusionne divers domaines tels que la mécanique, l'électronique, l'automatisation et l'informatique. Son objectif principal est de concevoir et de fabriquer des dispositifs mécaniques robotisés multifonctionnels programmables capables d'effectuer diverses tâches. Ces tâches variées vont de la fabrication industrielle aux soins médicaux, en passant par les transports, l'exploration spatiale et océanique, ainsi que la défense et la recherche scientifique. Les robots sont programmés pour exécuter des mouvements précis et variés, ainsi que des mouvements spécialisés adaptés à leur environnement et à leurs fonctions. Ils sont largement utilisés dans des situations dangereuses ou répétitives, difficiles voire impossibles à gérer pour l'homme. Leurs fonctions comprennent le déplacement d'objets, l'assemblage, le soudage, la peinture, le chargement et le déchargement de machines, etc.

En raison de leur polyvalence et de leur capacité à effectuer des mouvements répétés et précis, ils sont devenus des outils essentiels dans de nombreux secteurs industriels et dans une variété d'applications [1] [2].

Dans ce chapitre, nous débuterons en présentant plusieurs définitions et un bref historique de la robotique. Nous examinerons les différents types de robots ainsi que leurs composants. De plus, nous aborderons la classification des systèmes robotiques tout en mentionnant brièvement les domaines d'application potentiels.

I.2 Définitions de la robotique

La robotique est une discipline technologique qui englobe la création, la fabrication, le fonctionnement et l'utilisation des robots. Un robot est un système automatisé capable de manipuler des objets ou d'effectuer des opérations conformément à un programme préétabli, pouvant être ajusté ou modifié selon les besoins. Les robots industriels, tels que définis par l'Association française de la robotique industrielle, sont des machines composées de divers mécanismes, souvent équipées d'un ou plusieurs bras terminés par un poignet, capables de contenir des outils, des pièces ou des instruments de contrôle. Leur boîtier de commande comprend généralement un système de rangement et peut utiliser des accessoires sensibles et adaptables qui tiennent compte de l'environnement et de la situation. Ces machines polyvalentes sont souvent conçues pour des fonctions répétitives mais peuvent être adaptées à d'autres tâches.

Cette définition, qui intègre la perception de l'environnement par le robot, anticipe le développement potentiel des robots dits "intelligents" [3].

I.3 L'histoire de la révolution de la robotique et des robots [4]

La robotique a évolué de manière exponentielle au fil des années, comme le montre le chronogramme suivant :

- En 1801, Joseph-Marie Jacquard invente le métier à tisser programmable.
- En 1921, le mot « robot » est apparu.
- En 1942, Isaac Asimov définit les fameuses Trois Lois de la Robotique.
- Aux États-Unis, 1954 marque le début de la robotisation des usines : George C. De Vol Demande de brevet pour le premier robot appelé "Unimate" qui deviendra Officiellement Le premier robot industriel. C'est un bras qui peut bouger indépendamment Pièces plus ou moins résistantes. Pour vendre son invention, il fera équipe avec Joseph F. Engelberger et ils créeraient une société appelée Unimation Inc.
- En 1961, Le robot sera installé pour la première fois dans une usine sur la ligne de production principale Moteur électrique (usine automobile).
- En 1970, Victor Scheinman, professeur à l'Université de Stanford, a développé une Bras articulé à 4 degrés de liberté. Ce sera également le premier à pouvoir Télécommande par micro-ordinateur.
- En 1981, Au Japon, Takeao Kanade crée le premier robot avec des moteurs intégrés directement dans le robot Ses bras peuvent réaliser des mouvements plus rapides et plus précis.
- En 1987, Deux grandes industries de production de robots-Asea (Société suédoise) et Brown Boveri Ltd (société suisse) – fusion pour créer une société associée leurs connaissances et gagner plus de profits, ce groupe ABB sera nommé.
- En 1991, Un robot grand et lourd (200 kg) est construit qui va Appelé "IRB 6000". Ce sera le robot le plus rapide et le plus précis au monde.
- En 1996, Le concept de « robots collaboratifs » est apparu.
- En 2001, Première utilisation du da Vinci télécommandé : Opération Lindbergh Professeur Marescaux de New York et patient de Strasbourg.

- En 2001, FANUC a mis en œuvre le système « Dark Manufacturing », dans lequel Les robots construisent jusqu'à 30 autres robots sans surveillance humaine jours, à raison de 50 fois par 24 heures.
- En 2002, le Dr Michael Grieves a proposé le concept d'équivalents numériques
- Produit physique ou « jumeau numérique ».
- En 2003, Créer un programme et l'installer dans la mémoire du robot pour que Accélérer la production de 90 %, c'est le Teachsaver.
- En 2004, 30 ans après l'invention du robot, ABB vend plus de 100 000 machine et était le leader mondial à l'époque.
- En 2008, Universal Robots a vendu le premier robot collaboratif au monde à une entreprise. Fournisseur danois de plastique et de caoutchouc pour la surveillance Machines-outils CNC.
- En 2009, Le robot militaire « Big Dog » a effectivement été utilisé en Afghanistan.

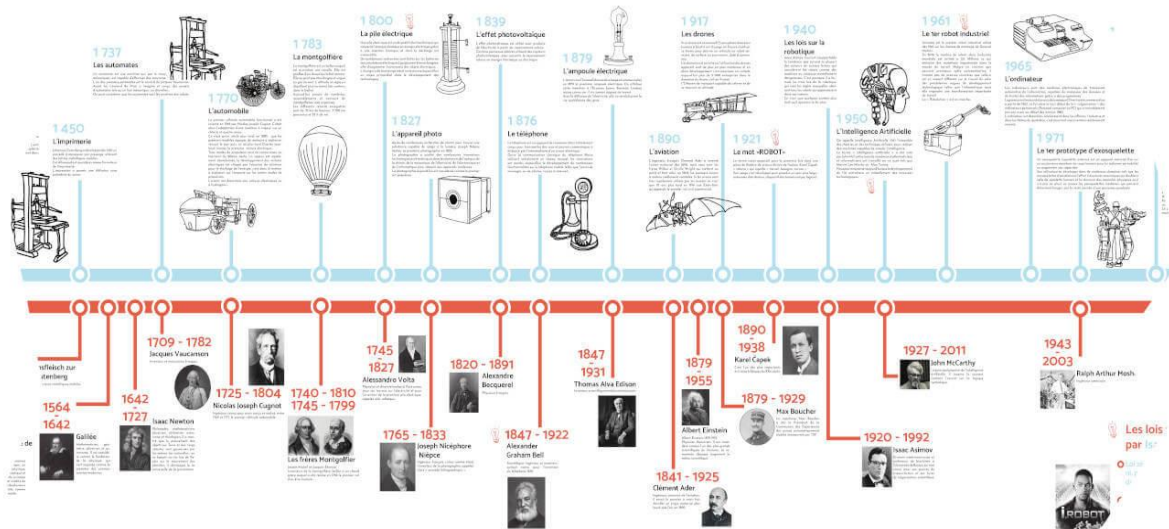


Figure I.1 : L'histoire de la robotique [5].

I.3.1 Les trois lois de la robotique

Isaac Asimov était un écrivain prolifique, principalement connu pour ses œuvres de science-fiction. Parmi ses contributions les plus célèbres figurent la série de romans "Fondation" et "Le Cycle des robots". Asimov a également formulé les "Trois lois de la robotique" dans ses histoires, ces lois sont des principes éthiques fictifs qui guident le comportement des robots dans son univers de fiction.

Les trois lois de la robotique sont :

- Un robot ne peut porter atteinte à un être humain, ni, restant passif, permettre qu'un être humain soit exposé au danger.
- Un robot doit obéir aux ordres que lui donnent les êtres humains, sauf si de tels ordres entrent en conflit avec la Première Loi.
- Un robot doit protéger son existence tant que cette protection n'entre pas en conflit avec la Première ou la Deuxième Loi.

Ces lois ont profondément influencé la réflexion éthique sur l'intelligence artificielle et les robots dans le monde réel [6].

I.4 Types de robots

Il existe deux types de robots : les robots mobiles et les robots manipulateurs. Les robots mobiles sont capables de se déplacer dans leur environnement, avec ou sans manipulateurs.

I.4.1 Robots mobiles

Les robots mobiles sont des machines qui peuvent se déplacer tout seuls dans leur environnement pour accomplir diverses tâches.



Figure I.2 : Robot mobile [7].

I.4.2 Robots manipulateurs

Ce sont des robots qui sont fixés sur le lieu de travail. Généralement configurés pour effectuer des tâches précises et répétitives, ces robots automatisés programmés peuvent remplacer les humains dans l'exécution de tâches répétitives, par exemple les bras robots, les robots médicaux, industriels, etc.



Figure I.3 : Bras manipulateur [8].

I.5 Structure fonctionnelle d'un robot

La figure I.4 illustre les différentes parties de la structure d'un robot.

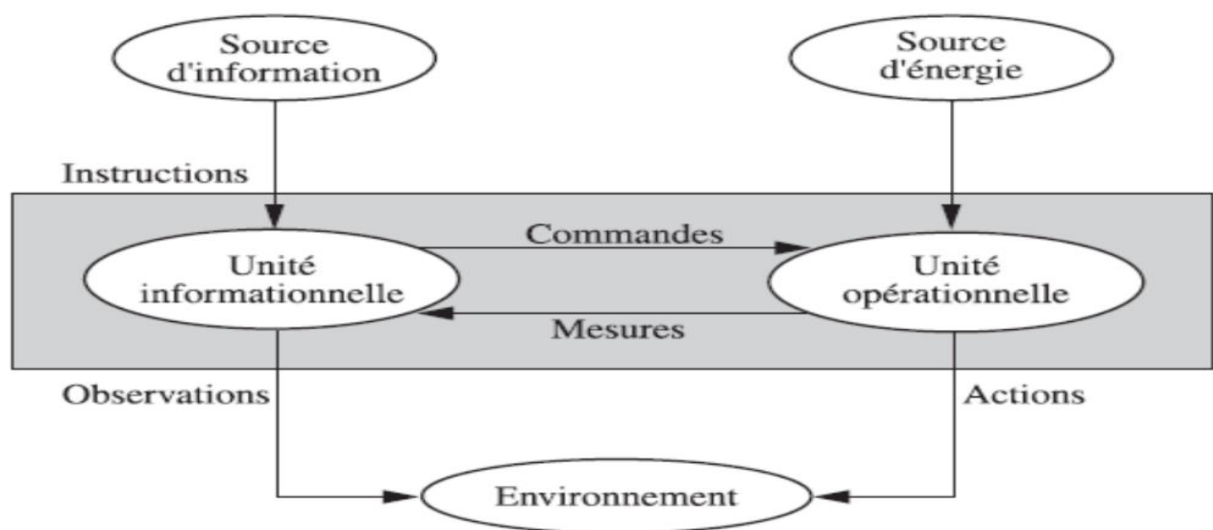


Figure I.4 : Structure fonctionnelle d'un robot [9].

- Unité d'information

Cette unité reçoit des instructions détaillant les tâches à effectuer ainsi que les mesures à prendre concernant les états internes des structures mécaniques composant le robot et les observations liées à son environnement. Elle a donc développé différents contrôles conjoints pour exécuter ces tâches. Actuellement, elle travaille sur l'interaction du système d'action permanent basé sur l'information [9].

- Unité opérationnelle

Les opérations contrôlées sont exécutées en utilisant l'énergie provenant d'une source d'alimentation. Cette partie représente le robot physique qui intègre des structures mécaniques telles que des segments, des articulations, des modules énergétiques comme des amplificateurs, des transmissions, des servovalves, des convertisseurs d'énergie tels que des moteurs, des actionneurs, des cinématiques de chaînes de machines, des transmissions telles que des réducteurs, des vis à billes, des courroies crantées, etc [9].

Des capteurs corporels sont placés sur chaque axe pour mesurer en continu leur positionne leur vitesse, et ils émettent des commandes aux actionneurs ou organes de commande ainsi qu'à l'environnement.

I.6 Domaine d'application des robots

La robotique est un domaine en plein essor ces dernières années. La technologie continue de dépasser nos attentes et permet désormais des solutions techniques qui s'adaptent au moindre problème. La robotique est donc utilisée dans des domaines extrêmement stricts et exigeants. Nous allons les explorer dans différents domaines :

I.6.1 Domaine de l'industrie

Dans l'industrie, l'utilisation de la robotique est devenue une composante extrêmement importante et même incontournable. Bien que les robots industriels jouent un rôle crucial dans le processus de fabrication des chaînes d'assemblage de l'industrie automobile, ils continuent de s'améliorer grâce à l'innovation et aux progrès technologiques, devenant ainsi une partie intégrante de la chaîne de valeur industrielle. Les processus automatisés et la recherche

constante de rentabilité, d'efficacité et de durabilité garantiront que la robotique conserve son rôle fondamental à l'avenir également.

Les robots offrent une gamme très diversifiée de possibilités d'application qui varient d'un secteur à l'autre. Cependant, les domaines d'application de production les plus courants sont les systèmes de manutention pour la mise en place, l'emballage et l'équipement des machines. Les robots industriels sont également utilisés dans des processus tels que la peinture, la mesure, le meulage, la découpe et le soudage. Les exigences des robots industriels varient, en particulier en ce qui concerne leur technologie et leur conception mécanique, en fonction des tâches qu'ils doivent accomplir.

Lorsque les étapes de travail et les processus sont simples, répétitifs et ne nécessitent aucune modification, les robots utilisés sont généralement statiques et dépourvus de capteurs. Cependant, dans de nombreux domaines, la plupart des chaînes de valeur nécessitent des robots complexes. Le large éventail de possibilités offertes par les capteurs, les technologies intelligentes, la programmation et le guidage est ici crucial.

Les robots mobiles et autonomes ont également un impact considérable sur l'industrie en élargissant la gamme d'applications robotiques. L'interconnexion avec la technologie intelligente et l'autonomie de travail qui en résulte permettent à ces robots d'améliorer le transport et la logistique sur de nombreuses étapes de production. Par exemple, si des matériaux doivent être transportés pour équiper une machine, un robot spécifique se charge de ce travail. Ce dernier constitue donc un élément important dans la connexion des différentes étapes du processus et fait partie de ce que l'on appelle l'usine intelligente [10].



Figure I.5 : Robot industriel [11].

I.6.2 Domaine militaire

Les robots militaires sont des robots autonomes ou télécommandés conçus pour des applications militaires. En effet, la miniaturisation permet désormais de créer des robots discrets équipés d'un grand nombre de capteurs, idéaux pour des missions d'espionnage ou de reconnaissance [12].



Figure I.6 : Robot industriel en domaine militaire [12].

I.6.3 Domaine de la santé

Les robots sont de plus en plus utilisés en médecine, que ce soit pour des échographies simples ou pour des opérations chirurgicales complexes. Ces robots ne sont pas entièrement autonomes, mais ils assistent les médecins et les chirurgiens, et permettent même des opérations à distance grâce à la télémédecine. On parle de chirurgie assistée, intégrant les dernières technologies informatiques et robotiques dans la pratique médicale et chirurgicale. Développée pour surmonter les limites des techniques mini-invasives existantes et pour aider les chirurgiens dans les opérations ouvertes, la chirurgie robotique est en plein essor. Bien qu'elle ne soit pas encore largement disponible, elle représente l'avenir de la chirurgie [13].



Figure I.7 : Chirurgie robotique [14].

I.6.4 Domaine domestique

La robotique est devenue un élément quotidien de l'industrie et de nombreux sous-domaines de l'économie et de la science. Bien que les humains, les machines et les robots travaillent main dans la main, le champ d'application des robots à la maison reste limité à quelques tâches. Les robots domestiques, également appelés robots de service, se caractérisent par leur mobilité et leur autonomie. Ils sont capables d'effectuer des tâches de nettoyage importantes, comme tondre la pelouse et passer l'aspirateur, sans intervention ni guidage humain. Grâce à des capteurs spécifiques et à une programmation orientée tâches, ces robots peuvent s'orienter et sont très capables de réagir aux changements de leur environnement.

Cependant, les complexités fondamentales de l'utilisation de la robotique à la maison posent des défis importants à la recherche et au développement. L'environnement en question peut être considéré comme très sensible et dynamique. Les robots mobiles, en particulier, opèrent en contact direct avec les humains. C'est pourquoi il n'y a pas de place pour des systèmes ou des technologies immatures et un fonctionnement précis et parfait est crucial. En outre, de nombreuses tâches nécessitent des schémas de mouvement, des étapes de travail et des conditions de travail importants et changeant fréquemment.

Par conséquent, les robots doivent être capables de comprendre leur environnement de manière précise et dynamique et de tirer parti de leur expérience. À cet égard, le volet social représente un intérêt fondamental.

Malgré les problèmes ci-dessus, la robotique devient de plus en plus importante pour les individus et le nombre de robots utilisés dans les foyers continue d'augmenter. Le progrès technologique et l'acceptation populaire jouent avant tout un rôle décisif dans ce processus. Un élément fondamental du succès et des progrès de la robotique dans son ensemble, y compris des robots domestiques, est également dû aux changements systémiques dans l'intelligence artificielle et le flux d'informations. En raison de l'automatisation du comportement intelligent, les robots domestiques doivent être capables de gérer les problèmes individuellement, notamment les valeurs d'expérience ou les informations externes ou internes. Dans ce contexte, il est donc nécessaire d'optimiser la communication et l'interaction entre les humains et les robots [15].



Figure I.8 : Robot domestique [16].

I.7 Structure générale d'un système robotique

La structure d'un système robotique implique une série de composants interconnectés collaborant pour accomplir des missions définies.

I.7.1 Système mécanique articulé

Un Système Mécanique Articulé (SMA) est un ensemble d'entités reliées entre elles par des connexions (rotules, pivots, glissières, sphères, etc.) et animées par des articulations mécaniques. Dans un SMA, certaines connexions sont motorisées. Nous allons parler de

liaisons actives, qui se trouvent notamment dans les pivots. D'autres ne sont pas motorisées ; on les appelle articulations passives, telles que les rotules [17] ...

I.7.2 Actionneur (unité moteur)

Un actionneur est un dispositif qui convertit l'énergie délivrée par une interface en énergie utilisable pour modifier le comportement ou l'état d'un système, ou pour effectuer des tâches dans des systèmes automatisés. Nous disposons de nombreux types d'actionneurs électriques, pneumatiques et hydrauliques.

I.7.2.1 Actionneur électrique

-Moteur à courant continu

Les actionneurs utilisent souvent des moteurs à courant continu à aimant permanent. Un moteur à courant continu est un convertisseur électromécanique qui permet une conversion bidirectionnelle de l'énergie entre des appareils électriques et mécaniques transportés par courant continu. Cela signifie qu'un moteur à courant continu sera capable de convertir l'énergie électrique en électricité. Il est converti en énergie mécanique et est donc affectée par sa particularité de pouvoir fonctionner dans les deux sens en fonction du sens du courant [18].

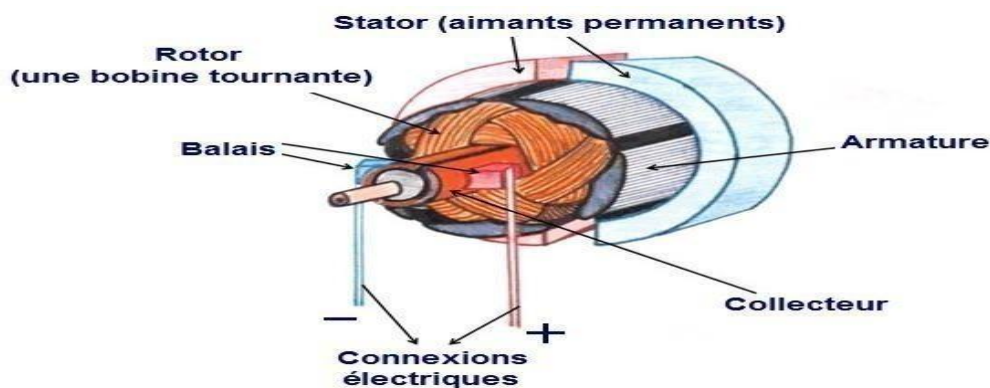


Figure I.9 : Le fonctionnement d'un moteur à courant continu [19].

Souvent, la vitesse élevée du moteur signifie qu'il y a un réducteur derrière lui, qui permet d'amplifier le couple du moteur, voir la figure I.10. On voit de plus en plus de moteurs à commutation électronique.

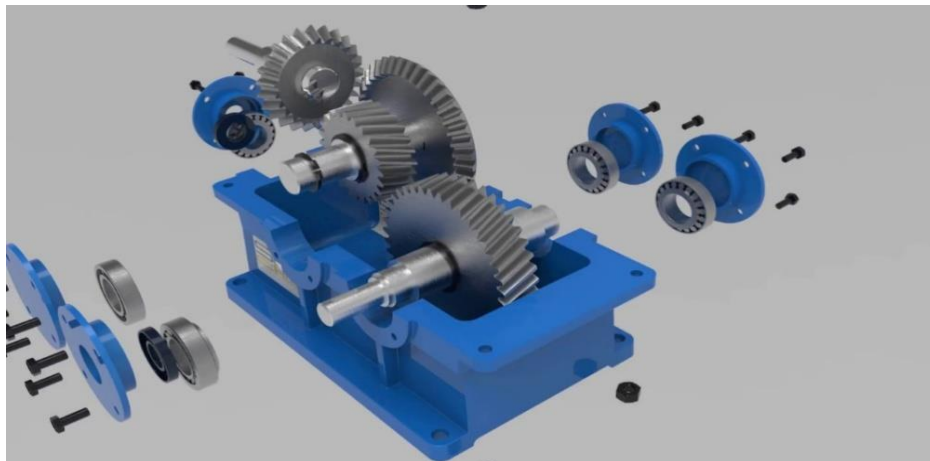


Figure I.10 : Réducteur en engrenage [20].

-Moteur pas à pas

Les moteurs pas à pas sont des moteurs spéciaux utilisés pour contrôler le mouvement et la position d'objets avec une grande précision. Comme leur nom l'indique, ces moteurs tournent par incréments discrets. Chaque incrément de rotation est provoqué par une impulsion de courant fournie à l'un des enroulements du stator.

Selon sa construction, un moteur pas à pas peut avancer de 90° , 45° , 18° ou juste une fraction de degré par impulsion. En faisant varier la fréquence d'impulsion, le moteur peut tourner très lentement, un pas à la fois, ou très rapidement à des vitesses allant jusqu'à 4 000 tr/min [21].

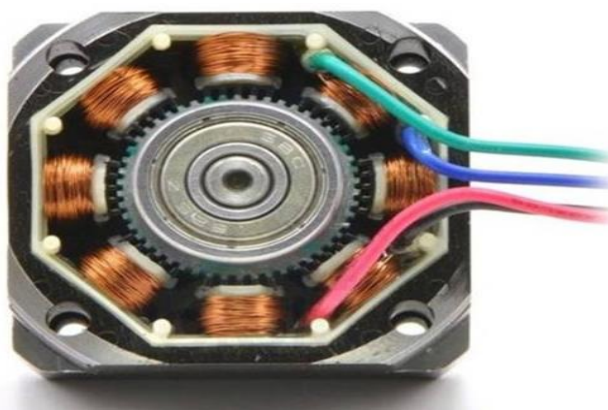


Figure I.11 : Moteur pas à pas bipolaire [22].

I.7.2.2 Actionneur pneumatique [23]

Un actionneur pneumatique est un dispositif qui transforme l'énergie de l'air comprimé en travail mécanique. Est utilisé avec de l'air comprimé entre 2 et 10 bars dans un usage courant. Simple à mettre en œuvre, il est très fréquent dans les systèmes automatisés industriels. Quand vous parlez d'actionneurs, vous devez considérer les vérins comme des actionneurs linéaires et les oscillateurs ainsi que les moteurs (à pistons et à ailettes) comme des actionneurs rotatifs.

-Vérin simple effet

Un vérin à simple effet ne peut produire une force que dans une seule direction, et la pression n'atteint qu'un seul orifice d'alimentation, qui entraîne le piston dans une seule direction.

Retour sous l'action d'un ressort ou d'une force extérieure.

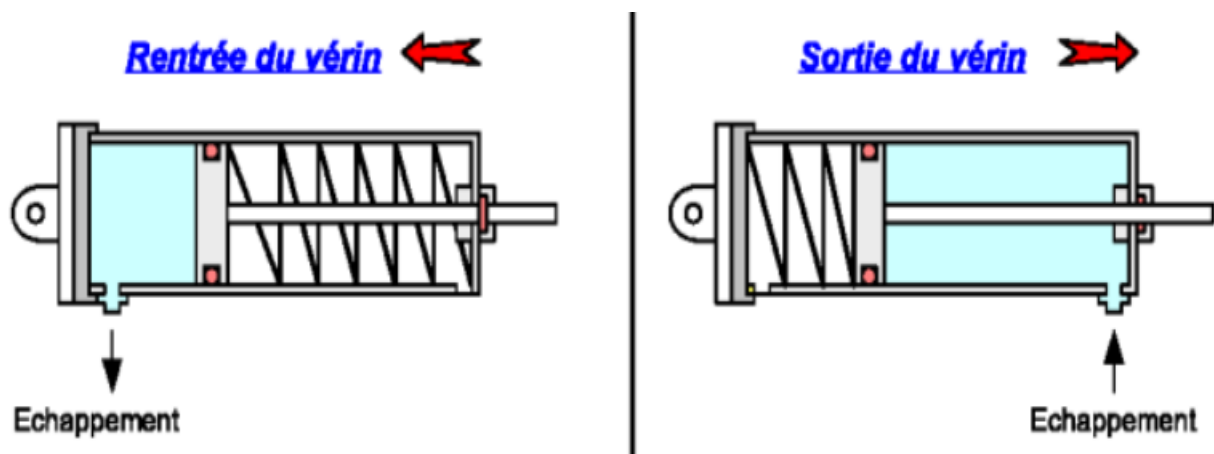


Figure I.12 : Vérin simple effet [24].

-Vérin double effet

Les vérins à double effet ont deux sens de travail. Il a deux trous

L'alimentation et la pression sont appliquées alternativement de chaque côté du piston, ce qui fait que le piston se déplace d'abord dans un sens puis dans l'autre. Nous vérifierons que le vérin n'est pas affecté par l'effet de multiplication de pression, qui pourrait provoquer une fissuration de son côté tige.

Les vérins sont généralement équipés d'amortisseurs de fin de course pour éviter les chocs provenant du piston.

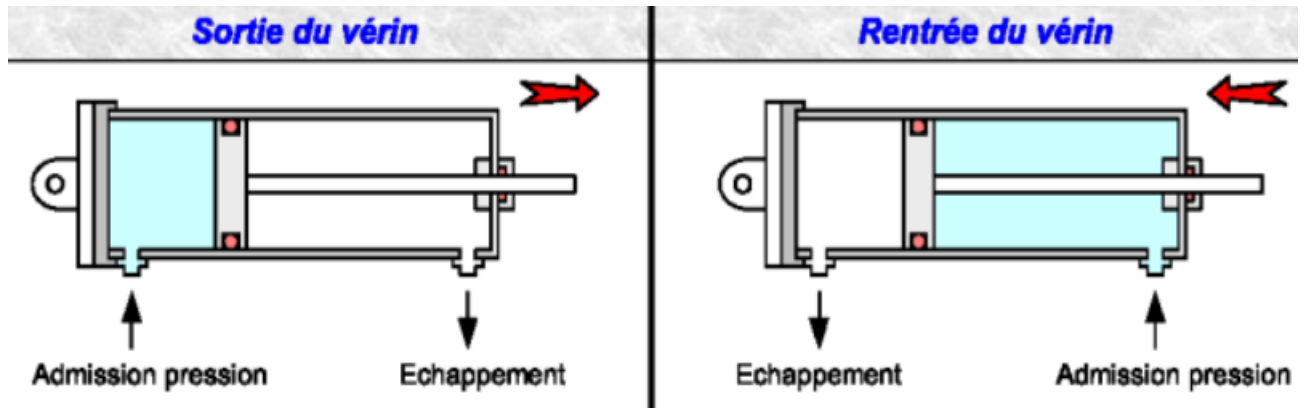


Figure I.13 : Vérin double effet [25].

I.7.2.3 Actionneur hydraulique

Pour les robots devant manipuler des charges très lourdes (comme les excavatrices) mécanique), l'actionneur est généralement hydraulique, à action de translation (vérin hydraulique) ou rotatif (moteur hydraulique).

Les vérins hydrauliques convertissent l'énergie hydraulique (pression, débit) en énergie Mécanique (effort, vitesse). Il s'utilise avec de l'huile sous pression jusqu'à 350 bar usage partagé. Plus cher, utilisé pour plus d'effort et de vitesse Il peut être développé pour être plus précis (et plus facile à ajuster) [26].

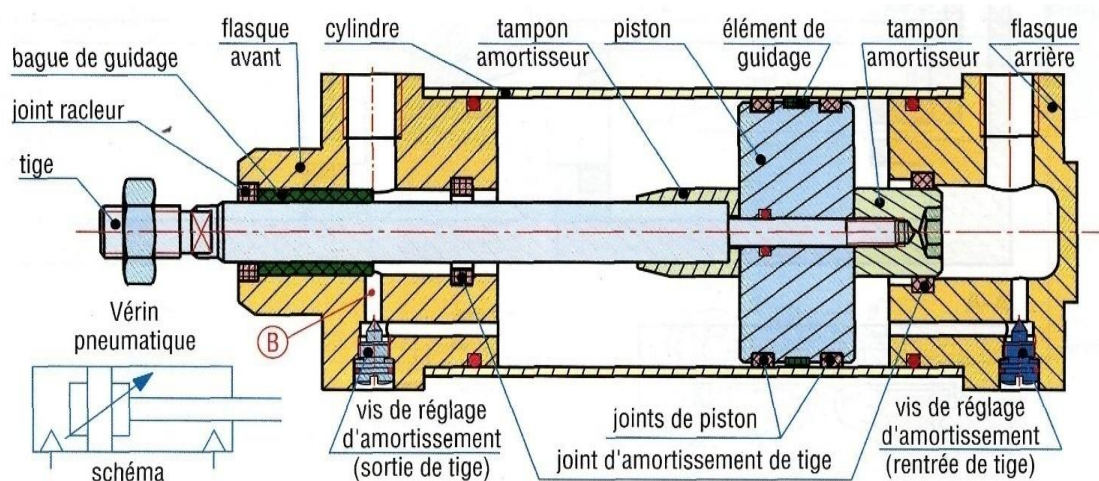


Figure I.14 : Actionneur hydraulique [27].

I.7.3 Capteur (organe de détection)

Le capteur convertit l'état observé des grandeurs physiques en une autre quantité disponible, telle que la tension, est simplement le processus physique et l'interface de traitement des données (informations Manipulable) ; c'est aussi un élément de base du système de collecte de données. Ils sont des organes sensoriels qui gèrent les relations entre les robots et son environnement [28].

Il existe deux types de capteurs :

-Capteurs proprioceptifs qui fournissent des informations sur la configuration Manipulateur (état interne du robot) tel que position, vitesse, accélération, etc... pour s'assurer que le robot manipulateur contrôle sa structure mécanique articulée [29].

-Capteurs extéroceptifs qui interviennent lorsque l'espace de travail est mal connue, afin de donner les informations sur l'environnement extérieur de manipulateur comme la température, l'image... ils permettent de modifier le comportement de manipulateur pour s'adapter aux contraintes imposées [29].

I.7.4 Effecteurs (organes terminaux)

L'actionneur est l'organe terminal du robot, fixé au poignet et doté d'une plaque à son extrémité.

La base comporte des trous filetés qui permettent de fixer différents actionneurs au robot universel, en l'adaptant à des tâches spécifiques.

En fait, la plupart des machines de production nécessitent des outils et des accessoires conçus spécifiquement pour des applications spécifiques ; les robots ne font pas exception à cet égard.

A noter également que de nombreux outils traditionnels utilisés manuellement ou sur certaines machines (pistolets à peinture ou à colle, pinces perceuses, etc.) peuvent devenir des effecteurs robotisés, mais avec beaucoup de travail. L'adaptation permet :

- Compenser certaines imprécisions ou dispersions dans les caractéristiques des objets extérieurs ou du robot lui-même.

- Présenter une certaine flexibilité pour se prêter à des tâches diversifiées.

I.7.5 Système de traitement (ordinateur)

C'est l'agence qui gère toutes les tâches. Il reconnaît trois rôles importants :

- Le rôle de l'information inclut la collecte d'informations provenant de capteurs.
- Vient ensuite le rôle décisionnel, en commençant par les tâches définies et en tenant compte

Avec les données du système et de l'environnement, il établit les opérations appropriées.

- Rôle de la communication.

Le contrôle du robot comprend des fonctionnalités qui lui permettent d'apprendre et d'apprendre

Programmez une tâche spécifique, puis effectuez cette tâche. La commande est

Mouvement, types de mouvement entre deux points et interactions avec les appareils

Extérieurement, tout cela fait partie de la fonction de contrôle.

I.8 Classification des robots

Les robots peuvent être classés d'un point de vue fonctionnel ou en fonction de leur objectif structure géométrique.

I.8.1 Classification fonctionnelle [30]

Nombre de catégories et différences entre catégories Les cours varient selon les pays (Japon Catégorie 6, France Catégorie 4). A.F.R.I. distingue 4 classes Comme indiqué ci-dessous :

-Robot programmable Ils répètent les actions pour lesquelles on leur a appris ou programmé Informations sur l'environnement ou les tâches effectuées. Nous pouvons encore le faire Différences entre les robots

« Playback » qui reproduisent les tâches apprises et les robots avec Commande numérique programmable hors ligne. Pour certains robots, comme les robots peintres, il faut suivre Une

trajectoire complexe difficile à exprimer mathématiquement. Un opérateur humain spécialisé dans la tâche exécute la trajectoire en guidant le bras du robot à l'aide d'une "marionnette". L'intégralité de la trajectoire est mémorisée par le robot. La figure représente un robot programmable.

-Manipulateur à commande manuelle

Un bras à commande manuelle est un appareil directement manipulé par un opérateur humain pour effectuer une tâche spécifique et nécessite l'intervention directe de l'utilisateur pour contrôler son mouvement et ses actions.



Figure I.15 : Bras à commande manuelle.

-Manipulateur automatique

Les bras de contrôle autonomes sont des dispositifs robotiques utilisés dans l'industrie qui effectuent des tâches sans intervention humaine directe. Ils peuvent être équipés de capteurs et de systèmes de vision pour s'adapter à différentes situations et contribuer à optimiser les processus industriels.



Figure I.16 : Bras à commande automatique.

- Robot programmable

Ils répètent les actions pour lesquelles on leur a appris ou programmé. Informations sur l'environnement ou les tâches effectuées. Nous pouvons encore le faire. Différences entre les robots « playback » qui reproduisent les tâches apprises et les robots avec Commande numérique programmable hors ligne. Pour certains robots, comme les robots peintres, il faut suivre Trajectoires complexes difficiles à exprimer mathématiquement, un opérateur humain spécialisé dans la tâche exécute la trajectoire en utilisant une « marionnette » pour guider le bras du robot, et toute la trajectoire est mémorisée par le robot. La figure ci-dessous représente un robot programmable.



Figure I.17 : Robot programmable.

- Robot intelligent

Il existe actuellement des robots de deuxième génération capables d'obtenir et d'utiliser certaines informations sur son environnement (système vision, détecteurs de proximité, capteurs de force, etc.) comme le montre la figure Robots de 3ème génération capables de comprendre le langage parlé Proche du langage naturel et géré de manière autonome Des environnements complexes dus à l'utilisation de l'intelligence artificielle.



Figure I.18 : Robot intelligent (humanoïde).

I.8.2 Classification géométrique [31]

On peut également classer les robots en fonction de leur configuration géométrique, autrement dit, l'architecture de leur vecteur.

- Structure cartésienne (PPP) :

Il possède trois connexions prismatiques et est le plus ancien de l'histoire de la logique de conception traditionnelle des machines-outils à trois axes, le type comme une meuleuse ou une fraiseuse. Cette structure est relativement rarement utilisée, sauf pour certaines applications spécifiques, comme les robots pratiques et les robots commerciaux.

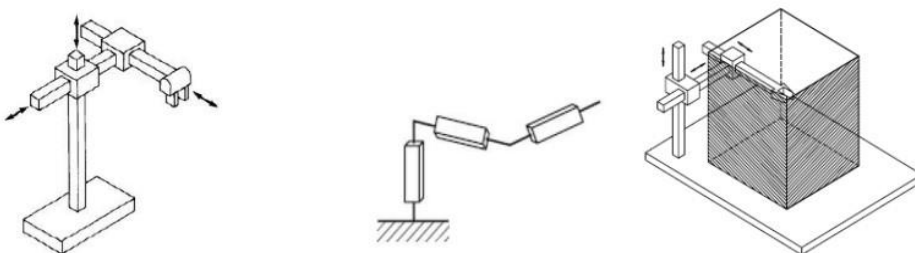


Figure I.19 : Structure cartésienne (PPP).

- Structure cylindrique (RPP) ou (PRP) :

Combine une rotation et deux traductions. L'inconvénient est une charge de travail moindre par rapport à une surface au sol totale plus grande. En réalité, il n'est plus utilisé.

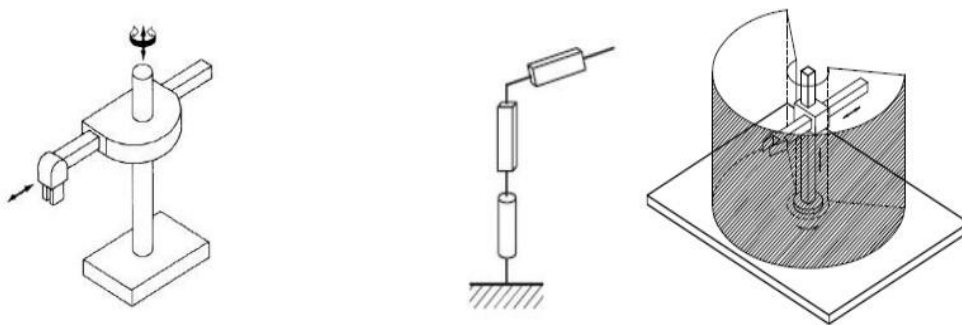


Figure I.20 : Structure cylindrique (RPP) ou (PRP).

- Structures sphériques :

La structure sphérique est une structure qui a été largement abandonnée, principalement en raison des mêmes raisons qui ont conduit à l'abandon de la structure cylindrique.

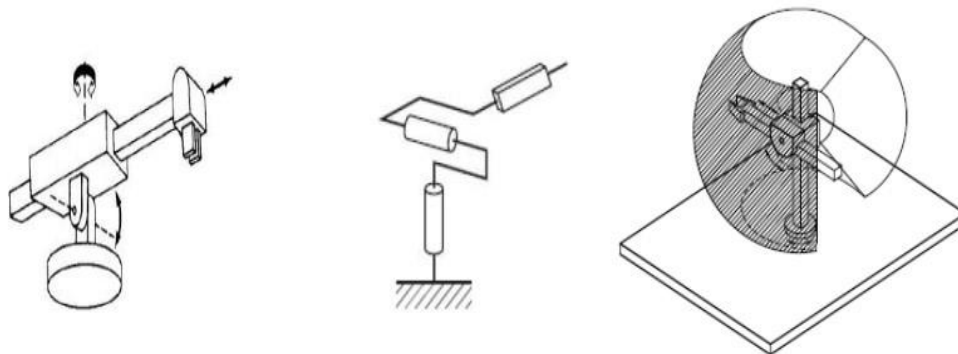


Figure I.21 : Structure sphérique.

- Structure dite SCARA :

Les axes rotatifs parallèles font partie des axes rotatifs les plus couramment utilisés, notamment pour les tâches de manipulation ou d'assemblage très fréquentes dans l'industrie. Ce succès commercial est lié à un rapport charge de travail/dimensions très favorable et à la structure SCARA bien adaptée à de telles tâches.

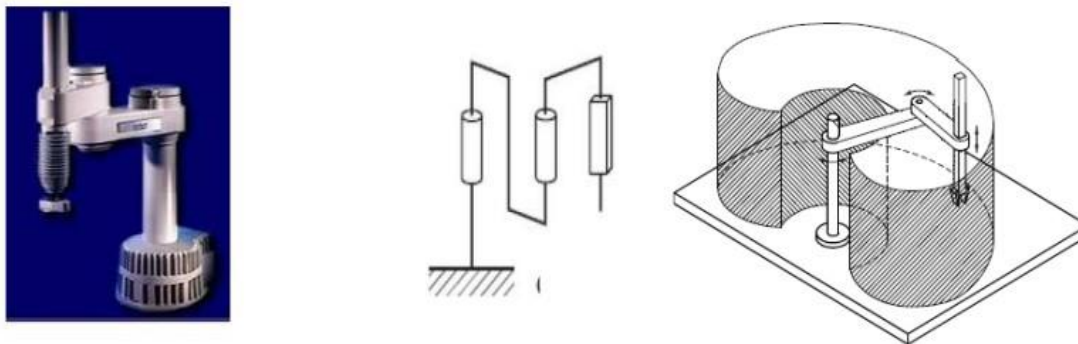


Figure I.22 : Structure dite SCARA.

-Structure 3R (anthropomorphe) :

Permet d'amener une entité à un point de l'espace grâce à trois rotations, Généralement un avec un axe vertical et deux avec des axes horizontaux et parallèles, c'est un excellent véhicule universel capable d'être facilement programmé pour différents types de tâches et avec une charge de travail importante.

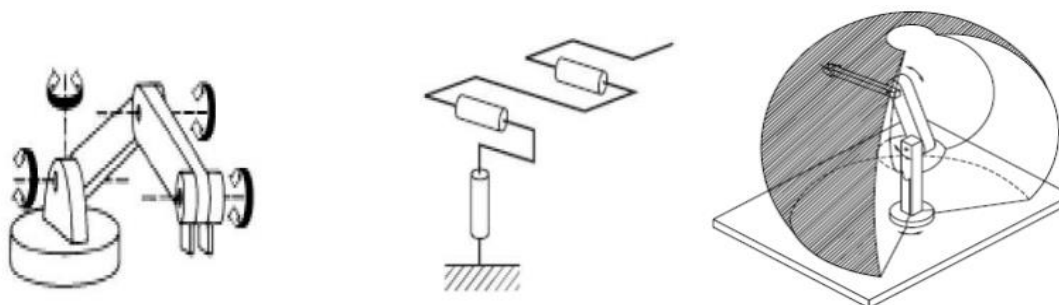


Figure I.23 : Structure 3R (anthropomorphe).

I.9 Avantages et Inconvénients des robots industriels :

Aujourd'hui, les robots industriels sont largement utilisés par les entreprises pour améliorer leurs chaînes de valeur. Ils opèrent dans plusieurs secteurs industriels de la robotique et facilitent la réalisation de grands projets. Il est important de comprendre les avantages et les inconvénients de ces machines avant de les utiliser.

I.9.1 Les avantages [32]

Premièrement, les robots sont des machines créées par les humains pour les aider à accomplir plus facilement les tâches qu'ils effectuaient auparavant avec leurs bras. De même, des robots industriels équipés de systèmes d'articulation sont créés pour aider les entreprises à réaliser leurs activités et à augmenter leur production. Ils présentent plusieurs avantages, notamment :

-La perfection dans les travaux : Lorsque vous mettez en œuvre des systèmes automatisés dans votre entreprise, vous gardez une production de qualité à un rythme plus rapide. Contrairement aux travailleurs humains qui ont besoin de pauses régulières, les robots sont capables d'effectuer des tâches avec une précision constante. Ainsi, l'utilisation de robots vous permet non seulement d'obtenir rapidement des produits de qualité et des résultats fiables, mais elle élimine également les erreurs humaines.

-La garantie d'une sécurité aux travailleurs : Les entreprises industrielles exercent des activités souvent dangereuses, exposant les travailleurs à des risques considérables dans des espaces confinés. Cependant, grâce à l'automatisation, cette exposition est désormais réduite, permettant aux travailleurs de se libérer des tâches dangereuses pour assurer leur sécurité. En effet, la fabrication de certaines pièces peut nuire aux travailleurs. La robotique peut donc prévenir de tels incidents, car les robots peuvent être contrôlés à distance sur la chaîne de production.

- La réduction du coût des mains-d'œuvre : Les entreprises industrielles exercent des activités souvent dangereuses, exposant les travailleurs à des risques considérables dans des espaces confinés. Cependant, grâce à l'automatisation, cette exposition est désormais réduite, permettant aux travailleurs de se libérer des tâches dangereuses pour assurer leur sécurité. En effet, la fabrication de certaines pièces peut nuire aux travailleurs. La robotique peut donc prévenir de tels incidents, car les robots peuvent être contrôlés à distance sur la chaîne de production.

- L'amélioration de votre productivité : La robotisation offre la possibilité d'optimiser l'efficacité de la production en contrôlant avec précision les flux d'entrée et de sortie et en établissant des normes que les robots suivent. De plus, cela évite les pertes de temps dues aux pauses des travailleurs et permet de programmer les robots pour maintenir une activité continue pendant de

courtes périodes. En intégrant différents types de robots industriels sur votre chantier, vous pouvez non seulement améliorer, mais augmenter votre productivité.

I.9.2 Les inconvénients [32]

La robotique est désormais très importante dans l'industrie car elle aide les entreprises à travailler en collaboration. Cependant, ces robots présentent également des inconvénients, notamment :

- **Le coût élevé de l'investissement** : Investir dans le secteur industriel peut présenter des défis. En effet, il est important d'installer efficacement les robots tout en planifiant les coûts associés à leur déploiement. Cette réalité peut entraver les efforts d'un réalisateur pour favoriser la collaboration au sein d'un projet. De plus, gardez à l'esprit que les coûts associés à cet investissement peuvent augmenter à mesure que la machine s'use au fil du temps.

- **Le manque de compétences techniques** : Pour maintenir des performances optimales à tout moment, les robots industriels nécessitent une supervision technique pour remédier aux dommages causés lors de leur utilisation. Ce besoin de techniciens spécialisés peut entraîner un investissement initial plus élevé et des obstacles opérationnels potentiels. De plus, la robotisation pourrait entraîner des pertes d'emplois.

I.10 Discussion

Dans ce chapitre, nous avons exploré en détail le domaine de la robotique, mettant en lumière son importance cruciale dans l'industrie. Nous avons examiné l'historique des robots, leurs structures, leurs utilisations, ainsi que les différents types et classifications, offrant ainsi une compréhension approfondie de leur fonctionnement et de leurs domaines d'application. En soulignant l'indispensabilité de la programmation pour démontrer les performances des machines, nous avons également identifié les avantages et les inconvénients inhérents à la robotique. Ces connaissances nous ont fourni une base solide pour la construction de notre bras robotique.

Dans le chapitre suivant, nous parlerons des différents composants et logiciels utilisés pour la réalisation de notre projet.

Chapitre II

Conception

matérielle et logiciel

II.1 Préambule

Dans ce chapitre, nous explorerons les fondements de la robotique, où la mécanique et l'électronique se fusionnent pour créer des systèmes complexes et fonctionnels. Cette alliance se manifeste à travers l'étude des actionneurs des chaînes cinématiques et des capteurs qui fournissent les données essentielles à leur fonctionnement. Notre objectif est de concevoir et de commander un bras manipulateur à quatre degrés de liberté en utilisant une carte Arduino comme plateforme de contrôle. Nous commencerons par une analyse approfondie de la partie mécanique de notre bras robotique, avant de plonger dans la modélisation, où nous établirons les relations mathématiques nécessaires entre les différentes grandeurs du système. Cette approche nous permettra de comprendre et de manipuler efficacement notre robot, ouvrant ainsi la voie à des applications innovantes dans le domaine de la robotique [33] [34].

II.1 L'architecture d'un bras robotique

Pour faciliter l'adaptabilité du bras robotique, nous avons opté pour une architecture modulaire. Cela implique que le bras sera constitué de plusieurs modules dédiés à des tâches spécifiques, capables de fonctionner de manière autonome et de communiquer facilement entre eux. Ainsi, nous pourrions remplacer les composants sans interrompre le contrôle du moteur. De plus, cette approche nous permettra d'ajouter de nouveaux modules en les connectant simplement au bus de communication.

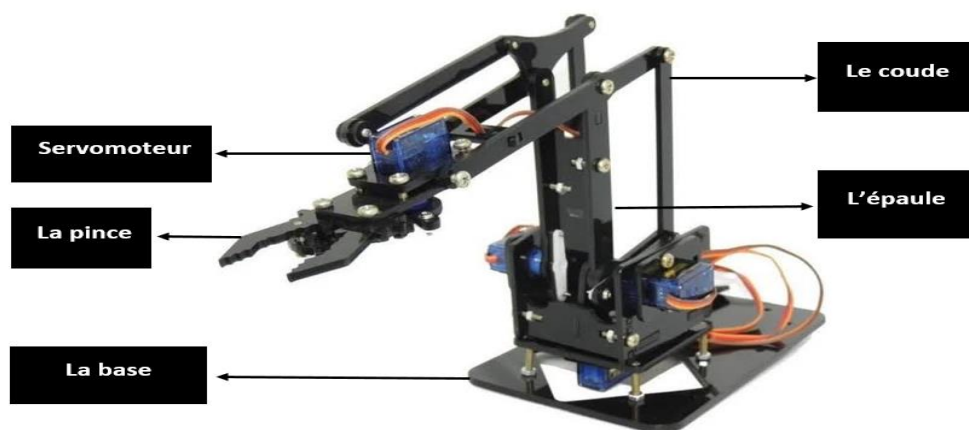


Figure II.24 : Les éléments constitutifs d'un robot.

II.2 Les Caractéristiques d'un bras robotique

Le bras robotique est doté de plusieurs fonctionnalités importantes pour réaliser des mouvements cycliques et imiter les gestes d'un bras humain. La caractéristique cruciale pour cela est le nombre de degrés de liberté (DOF), qui représentent les différents mouvements que le bras peut effectuer. Les articulations du bras robotique sont actionnées par quatre servomoteurs, des moteurs capables de maintenir leur position même sous une force statique, ce qui permet de reproduire une grande variété de mouvements.

II.3 Conception du système de commande du bras robotique

Dans notre travail, nous proposons de concevoir un système de commande du bras d'un robots en utilisant deux méthodes distinctes. Afin de prendre en considération l'aspect de sécurité, nous nous sommes intéressé dans une première approche à utiliser une commande manuelle basée sur la carte Arduino Uno. Dans une deuxième approche, nous avons utilisé une commande automatique basée sur la carte ESP32 en utilisant un smartphone. Nous décrivons ci-après en détail les deux commandes.

II.4 La commande manuelle

Dans la commande manuelle, notre système utilise une carte Arduino Uno pour générer les instructions nécessaires pour faire manipuler les articulations du bras. Ces instructions permettent de commander les servomoteurs à l'aide de deux Joysticks manipulé par l'utilisateur. La figure II.1 illustre le schéma de principe de cette commande manuelle du bras robotique.

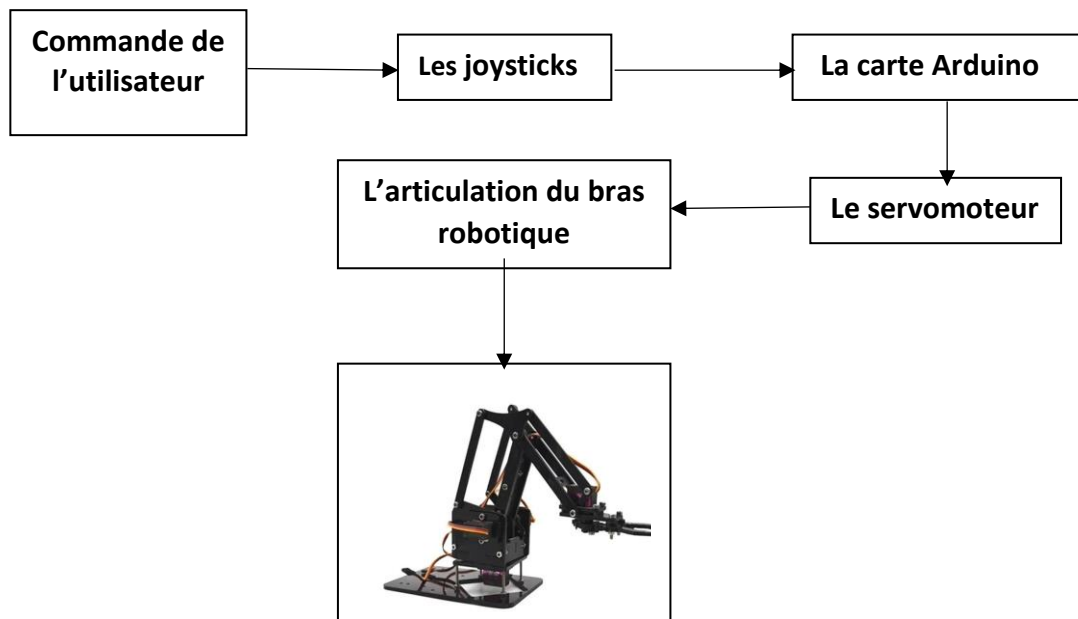


Figure II.25 : La commande manuelle.

II.4.1 Joystick pour Arduino [35]

Le joystick est un dispositif de commande qui détecte la position dans deux axes, X et Y, et fournit des valeurs analogiques correspondantes. Il est souvent utilisé comme système de navigation dans les menus ou comme interface pour contrôler la direction ou la vitesse d'un objet. On le trouve fréquemment sur les manettes de jeux vidéo, les télécommandes de modèles réduits, et même sur les tableaux de bord des machines industrielles.

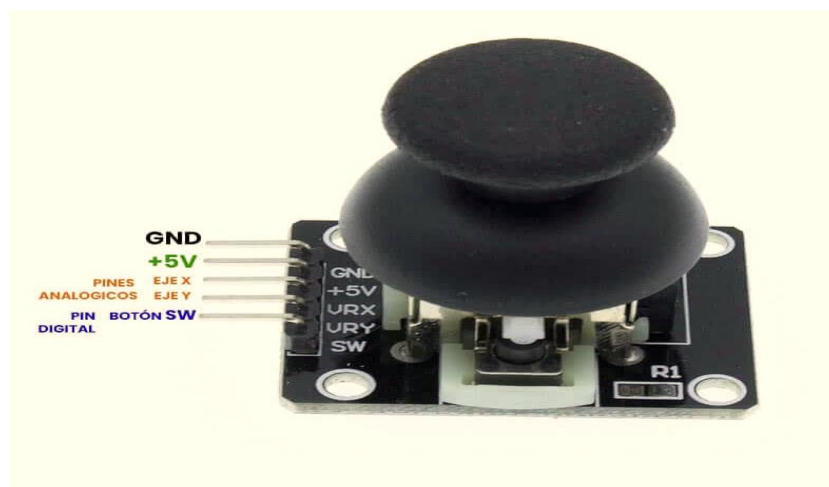


Figure II.26 : Le Joystick pour Arduino [36].

- Principe de fonctionnement :

Le joystick est composé de deux potentiomètres couplés mécaniquement qui détectent les mouvements horizontaux et verticaux du joystick. Ainsi, la valeur de résistance des potentiomètres varie indépendamment en fonction de la position du joystick. Généralement, il comporte également un bouton qui est enclenché lorsqu'on appuie sur le joystick.

- Caractéristiques du joystick :

Le joystick est doté de deux potentiomètres avec des résistances équivalentes, ce qui divise la tension de 5V en deux portions égales, soit 2,5V pour chaque potentiomètre. En déplaçant le joystick, la tension varie de 0V à 5V, couvrant ainsi toute la plage de valeurs possibles pour chaque axe.

Lorsque nous lisons la valeur du joystick avec une carte Arduino, nous obtenons une valeur de 512 lorsque le joystick est au repos, et une valeur variant de 0 à 1023 lorsque le joystick est déplacé, en fonction de sa position. Le joystick est également équipé d'un bouton poussoir, qui peut être utilisé pour prendre le contrôle de l'ordinateur ou pour effectuer des clics sur les éléments désirés.

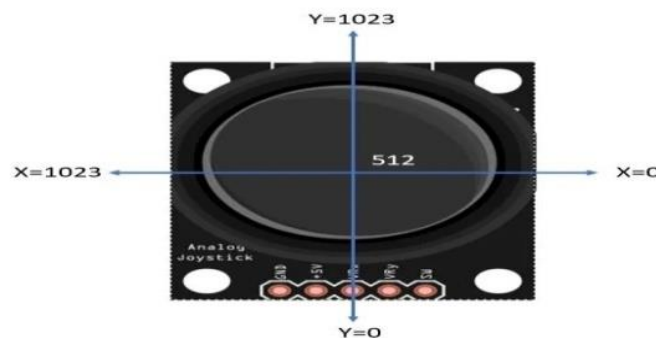


Figure II.27 : Les axes du joystick.

II.4.2 La carte Arduino [37]

Dans cette commande, nous avons utilisé une simple carte Arduino de type Uno pour générer les instructions pour commander les servomoteurs.

La carte Arduino est une petite carte électronique (typiquement de dimensions 5,33 x 6,85 cm) équipée d'un microcontrôleur. Ces microcontrôleurs permettent de programmer et de

contrôler les actionneurs en fonction des événements détectés par les capteurs. Ainsi, la carte Arduino agit comme une interface programmable.



Figure II.28 : La carte Arduino Uno [38].

- Description de la carte :

La carte Arduino Uno se compose de plusieurs composants, chacun ayant un rôle bien précis, comme le montre la figure II.5 ci-dessous.

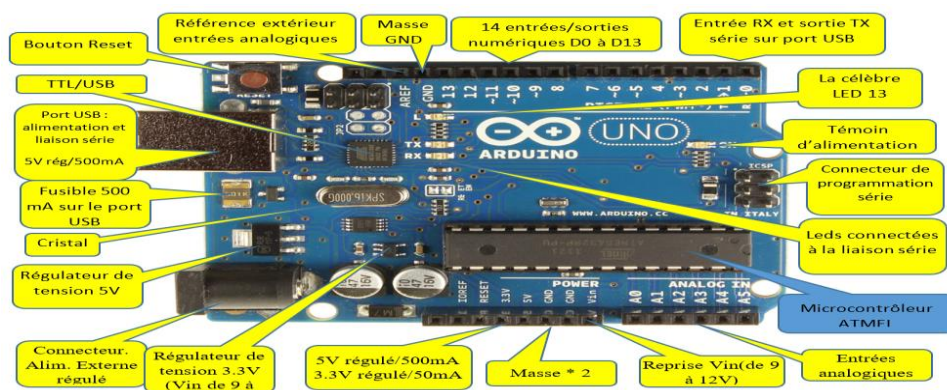


Figure II.29 : Structure de la carte Arduino [39].

- Les entrées/sorties numériques :

Les connecteurs de D0 à D13 peuvent être configurés par programmation en entrée ou en sortie. Par exemple, les connecteurs 2 et 3 peuvent être configurés comme des entrées, tandis que les connecteurs 7, 8 et 9 peuvent être configurés comme des sorties. Cela permet de connecter des capteurs logiques (comme des interrupteurs) aux connecteurs 2 et 3, et des actionneurs aux connecteurs 7, 8 et 9.

Les signaux véhiculés par ces connecteurs sont des signaux logiques, ce qui signifie qu'ils ne peuvent prendre que deux états : HAUT (5 Volts) ou BAS (0 Volt), par rapport au connecteur de masse GND, qui est toujours à 0 Volt.

Il est important de noter que les connecteurs ne peuvent pas fournir un courant en sortie supérieur à 40 mA, ce qui interdit de brancher directement un moteur sur une sortie logique. Remarquez le signe ~ sur les connecteurs 3, 5, 6, 9, 10 et 11 ; nous verrons plus tard sa signification (PWM) et son importance. Les termes « numériques », « logiques » ou « digitales » sont souvent utilisés de manière interchangeable pour qualifier ces entrées/sorties.

- Les entrées analogiques A0 à A5 :

Contrairement aux entrées/sorties numériques, ces six entrées peuvent admettre un millier de valeurs analogiques comprises entre 0 et 5 Volts (exactement 1024 valeurs). Ainsi, des valeurs de tension précises à 5 mV près peuvent être mesurées (environ $5V/1024$).

Par exemple, en faisant varier un potentiomètre, la tension de l'entrée A0 peut varier de 0 V à 5 V. Cette valeur analogique peut être utilisée par le programme pour contrôler divers dispositifs, comme la luminosité d'une DEL, la vitesse de rotation d'un moteur ou la position d'un servomoteur.

- Le terminal série :

Partie matérielle (hardware) : Arduino, tout comme la plupart des microcontrôleurs actuels, est doté d'un UART (UniversallyAsynchronousReceiver/Transmitter) utilisé pour l'émission et la réception de données via une liaison série aux niveaux TTL (Transistor-Transistor Logic), appartenant à l'une des premières familles de circuits logiques.

Électriquement parlant, en TTL, un zéro est représenté par une tension proche de 0 volt (généralement inférieure à 0,5 volt), tandis qu'un un est représenté par une tension d'environ 5 volts (comprise entre 2,4 et 5 V).

En revanche, au niveau de la sortie du PC, le protocole série RS232 est utilisé. Un niveau logique "1" correspond à une tension négative d'environ -12 V, tandis qu'un "0" correspond à une tension positive d'environ +12 V. Ainsi, une conversion entre TTL et RS232 est nécessaire.

Un autre défi réside dans le fait que les ordinateurs modernes ne sont plus équipés de connecteurs RS232, mais plutôt de ports USB. Par conséquent, les cartes Arduino intègrent un composant permettant la conversion entre le microcontrôleur et l'USB : le FTDI pour les anciennes cartes ou l'ATmega8U2 pour les nouvelles.

La liaison est de type série, plus précisément RS232. Les premières versions proposaient un connecteur série au lieu d'un port USB.

- Microcontrôleur ATMEL ATmega328 :

Le microcontrôleur utilisé sur la carte Arduino UNO est l'ATmega328 d'Atmel. Il appartient à la série AVR 8 bits d'Atmel et est présenté dans un boîtier de type DIP 28 (Dual In-line Package), comportant 28 broches.

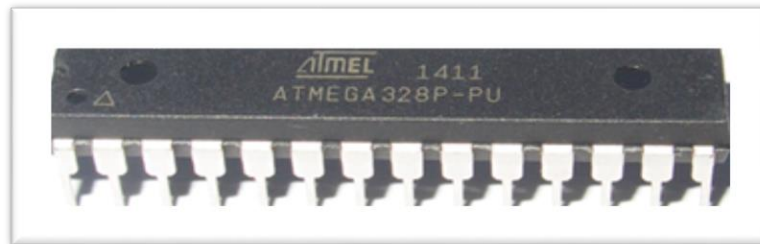


Figure II.30 : Microcontrôleur ATMEL ATmega328.

Les principales caractéristiques sont les suivantes :

- FLASH : 32 Ko de mémoire programme, capable de stocker des programmes même lorsque l'Arduino est débranché.
- SRAM : 2 Ko de mémoire vive, utilisée pour stocker les données volatiles nécessaires au programme.
- EEPROM : 1 Ko de mémoire non volatile, permettant de stocker des informations qui doivent être conservées même lorsque la carte est hors tension.
- E/S numériques (DIG In-Out) : 3 ports (PortB, PortC, PortD) totalisant 23 broches d'entrée/sortie.

- Minuteries/compteurs : Timer0 et Timer2 (comptage sur 8 bits), Timer1 (comptage sur 16 bits). Chaque minuterie peut être utilisée pour générer deux signaux PWM. (OCxA / OCxB à 6 broches).

II.4.3 Le servomoteur [40] [41]

Un servomoteur (du latin "servus", signifiant « subordonné », souvent abrégé en « servo ») est un moteur capable de résister aux forces statiques tout en maintenant une position précise, constamment vérifiée et ajustée en conséquence. Il s'agit donc d'un système dépendant. Un servomoteur est un dispositif électrique conçu pour atteindre et maintenir une position donnée.



Figure II.31 : Servomoteur 9g [42].

Comme les moteurs à courant continu, les servomoteurs possèdent un arbre rotatif freiné par un système de serrage. Cela ne signifie pas qu'il est totalement immobilisé, mais plutôt qu'il ne peut pas tourner au-delà d'une certaine limite. Par exemple, certains servomoteurs ne peuvent effectuer qu'une rotation partielle de leur arbre rotatif, tandis que d'autres peuvent réaliser une rotation complète, mais pas au-delà. Un servomoteur est donc un composant mécanique et électronique comprenant :

- Un petit moteur à courant continu.
- À la sortie du moteur, un réducteur diminue la vitesse tout en augmentant le couple.
- Un potentiomètre fonctionne comme un diviseur de tension résistif, générant une tension variable proportionnelle à l'angle de rotation de l'arbre de sortie.
- Des circuits électroniques contrôlent le fonctionnement du servomoteur.

- Des axes s'étendent depuis le boîtier, équipés de divers bras ou roues fixes.

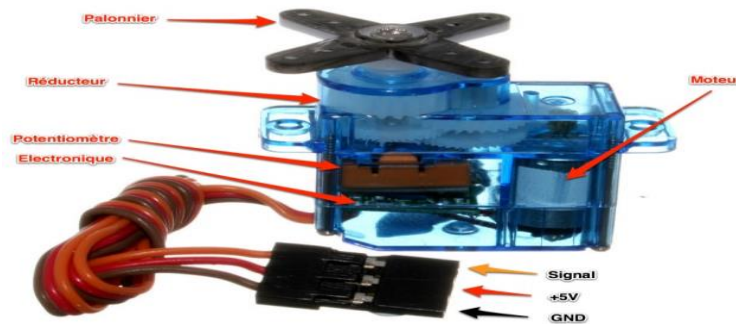


Figure II.32 : Les composants du servomoteur [43].

Dans l'écosystème Arduino, les servomoteurs sont principalement employés dans des applications robotiques. Un servomoteur est doté de trois fils de connexion distincts. Deux fils sont dédiés à l'alimentation, tandis que le troisième reçoit le signal de commande :

- Le fil rouge est utilisé pour l'alimentation positive (généralement entre 4,5V et 6V).
- Le fil noir ou marron est la masse (0V).
- Le fil orange, jaune, blanc, etc., correspond à l'entrée du signal de commande.

La puissance nécessaire au fonctionnement du servomoteur est fournie par l'alimentation, tandis que le signal de commande consomme peu voire aucun courant. Les servomoteurs peuvent être directement contrôlés par des microcontrôleurs.

Pour contrôler le servomoteur, nous utilisons un signal PWM (modulation de largeur d'impulsion) avec une fréquence fixe de 50 Hz ($T=20$ ms). La durée de l'état haut de ce signal détermine la position de l'arbre du servomoteur.

II.4.4 Pilotage d'un servomoteur avec Arduino

Autrefois, pour contrôler un servomoteur, on utilisait un circuit oscillateur pour envoyer une impulsion de durée correspondant à l'angle désiré. Mais grâce à la programmation sur Arduino, ce circuit n'est plus nécessaire. Désormais, une bibliothèque spéciale appelée "Servo" permet de contrôler le servomoteur en lui indiquant simplement l'angle où l'on souhaite qu'il se positionne. Cela facilite la réalisation de mouvements complexes et leur automatisation. La figure ci-dessous illustre ce pilotage.

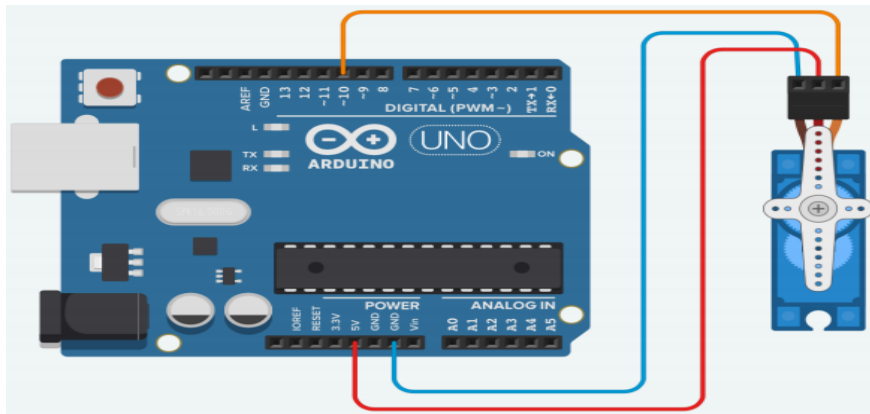


Figure II.33 : Pilotage d'un servomoteur par la carte Arduino.

II.4.5 L'articulation du bras robotique

On a réalisé un tableau avec deux colonnes, l'une pour le composant et l'autre pour sa description.

Composants	Description
	<p>La base : La base est composée de deux parties : une partie inférieure et une partie supérieure. Le plateau inférieur est immobile tandis que le plateau supérieur est rotatif.</p>



	<p>L'épaule : L'épaule est reliée à la base par une seconde articulation, contrôlée par un servomoteur, ce qui lui permet de pivoter dans une plage de mouvement allant de 0° à 180°</p>
	<p>Le coude : Le coude, relié à l'épaule par une troisième articulation, offre une amplitude de rotation de 0° à 180°.</p>
	<p>La pince : La pince est constituée d'un ensemble d'engrenages permettant son ouverture et sa fermeture, facilitant ainsi la saisie d'objets.</p>

Tableau II.2 : Les articulations du bras robotique.

II.4.6 Alimentation

Le connecteur DC Jack de la carte Arduino permet une alimentation directe, étant connecté au régulateur de tension intégré à la carte. L'alimentation via ce connecteur doit être comprise entre 5 et 12 volts pour assurer un fonctionnement stable de la carte et de ses composants.



Figure II.34 : Alimentation 12 V/2A [44].

II.5 La commande automatique

Dans la commande automatique, nous avons utilisé les mêmes composants que dans la commande manuelle, à l'exception du joystick. De plus, nous avons remplacé la carte Arduino par une carte ESP32 afin d'utiliser de générer les instructions de commande par Smartphone. Cette carte qui possède une mémoire de stockage qui permet de prendre en charge notre programme de gestion en utilisant la connexion WIFI de notre Smartphone. Le schéma de principe de fonctionnement de notre commande est illustré sur la figure II.36.

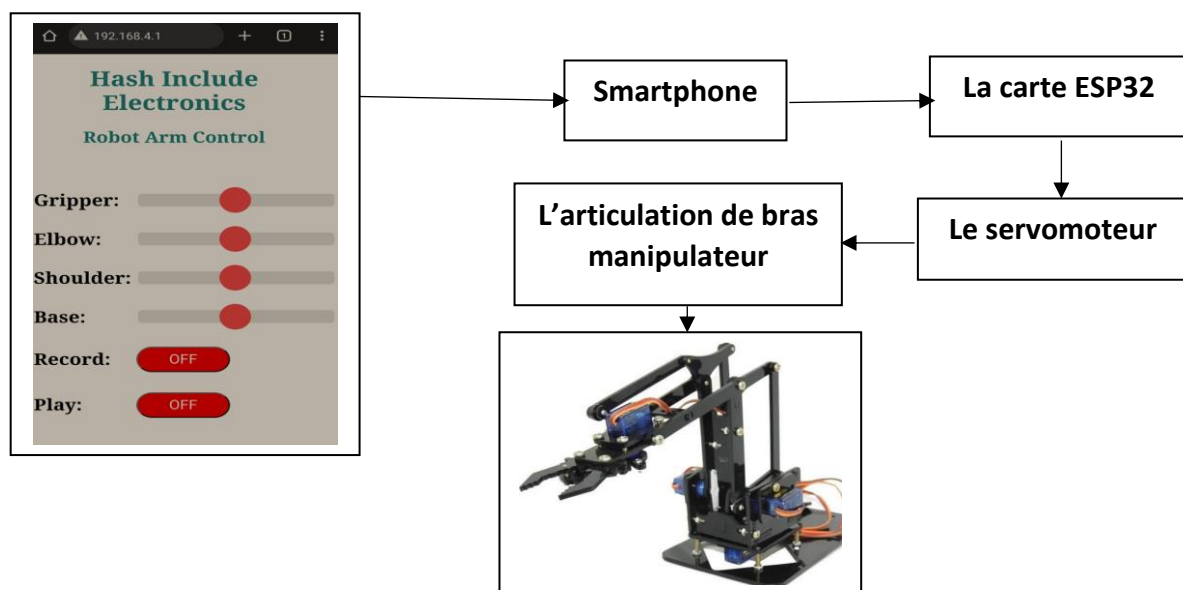


Figure II.35 : La commande automatique.

II.5.1 Séquence de commande

La séquence pour manipuler les articulations du bras sont générés par notre programme exécuté dans notre smartphone qui permet de l'enregistrer afin de l'envoyer par une connexion WIFI vers la carte ESP 32 en utilisant le protocole TCP/IP. La figure suivante illustre un exemple de séquence de programme généré par notre smartphone.

II.5.2 La carte ESP32 [45]

L'ESP32 est un microcontrôleur 32 bits polyvalent, avec des fonctionnalités Wi-Fi, Bluetooth et BLE intégrées, développé par Espressif Systems. Il cible une large gamme d'applications, offrant une faible consommation d'énergie et des solutions économiques.

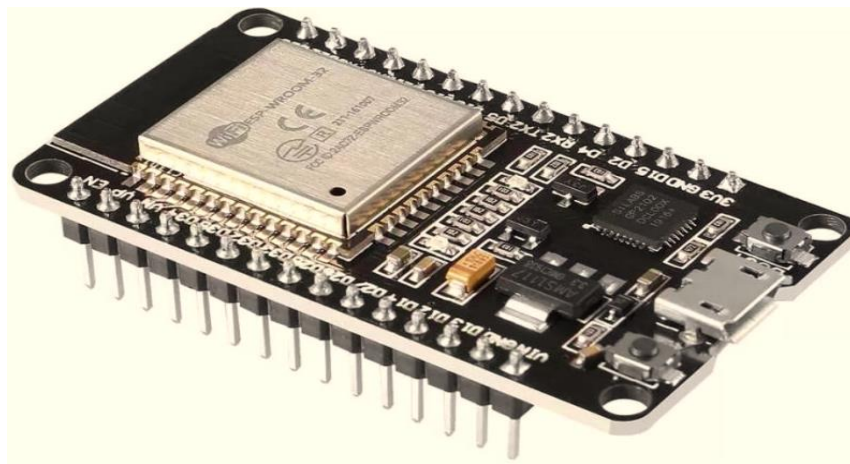


Figure II.36 : Carte de développement ESP32 Wroom [46].

- Caractéristiques de l'ESP-Wroom-32 :

- **Processeur** : Microprocesseur Dual-Core 32 bits TensilicaXtensa à faible consommation, fonctionnant à 240 MHz avec une performance de 994,26 CoreMark et 4,14 CoreMark/MHz.
- **Mémoire** :
 - 448 KB de ROM pour le démarrage et les fonctions de base.
 - 520 KB de SRAM intégrée pour les données et les instructions.
 - 4 MB de mémoire Flash.
 - 16 KB de SRAM dans le RTC.

- **Connectivité :**
- Wi-Fi 802.11b/g/n.
- Bluetooth v4.2 BR/EDR et spécifications Bluetooth LE.

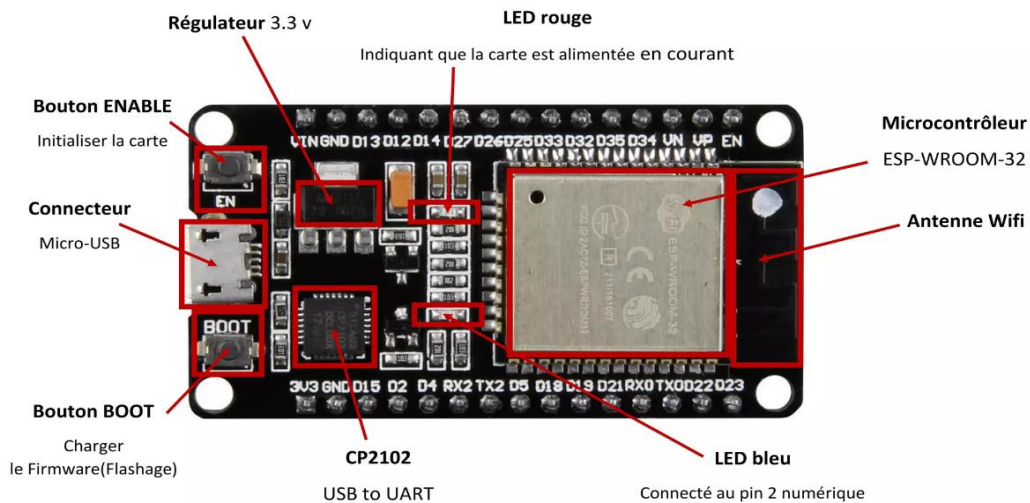


Figure II.37 : Fonctionnalités de la carte ESP32[47].

La carte de développement ESP-WROOM-32 est équipée du microprocesseur Dual-Core 32 bits TensilicaXtensa, fonctionnant à une fréquence d'horloge ajustable de 80 à 240 MHz. Les spécifications clés incluent :

- **Mémoire** : 448 KB de ROM, 520 KB de SRAM intégrée et 4 MB de mémoire Flash.
- **Connectivité sans fil** :
 - Le Wi-Fi intégré prend en charge la norme 802.11b/g/n, fonctionnant dans la bande de 2,4 GHz avec des vitesses allant jusqu'à 150 Mbps.
 - Prend en charge le Bluetooth v4.2 BR/EDR et Bluetooth LE, offrant un mode double Bluetooth pour une polyvalence accrue.
- **Boutons de réinitialisation/démarrage** :
 - **Bouton de réinitialisation** : Utilisé pour réinitialiser la puce ESP32.
 - **Bouton de démarrage** : Utilisé pour entrer en mode de démarrage pour télécharger de nouveaux croquis ou programmes.

- **Alimentation :**

- La carte peut être alimentée par une source externe de 5V ou via USB.
- Fonctionne dans une plage de tension de 2,3V à 3,6V, avec un courant de sortie recommandé de 600 mA ou plus.
- Inclut un régulateur de tension 1117 intégré pour convertir l'entrée de 5V en sortie de 3,3V.

- **Indicateur de statut :** Une LED rouge indique le statut pendant le téléchargement des programmes et la communication série.

-Les Broches de la carte de développement ESP32 :

La carte de développement ESP32 comprend un total de 32 broches, mappées comme suit :

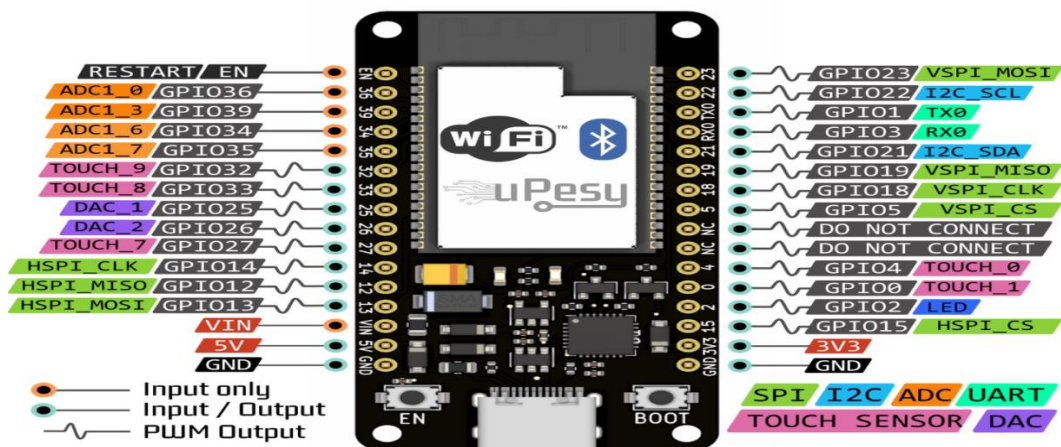


Figure II.38 : Les broches de la carte ESP32 [48].

- Interfaces périphériques avancées :

- **GPIO :** La carte dispose de 32 broches GPIO multifonctionnelles, configurables en pull-up interne, pull-down ou haute impédance. Les entrées peuvent être définies sur déclenchement par front ou par niveau pour générer des interruptions CPU.
- Remarque : Les GPIO34, GPIO35, GPIO36 et GPIO39 sont uniquement des broches d'entrée.
- **ADC :** Inclut des ADC 12 bits SAR (Registres à Approximation Successive) prenant en charge les mesures sur 16 canaux.
- **DAC :** Contient deux canaux DAC 8 bits pour convertir les signaux numériques en sorties de tension analogiques, prenant en charge les conversions indépendantes.

- **PWM** : Prend en charge 8 bits 32 canaux PWM, utile pour la gradation des LED ou le contrôle des moteurs.
- **Touch Pad** : Inclut 10 broches GPIO capacitives pour détecter le toucher physique, pouvant remplacer les boutons et interrupteurs mécaniques.
- **I2C** : Prend en charge une interface de bus I2C, configurable en tant que maître ou esclave, avec prise en charge de :
 - Mode standard (100 Kbit/s)
 - Mode rapide (400 Kbit/s)
 - Modes d'adressage 7 bits et 10 bits
 - Mode d'adressage double
- **SPI** : Inclut trois interfaces SPI (SPI, HSPI, VSPI) avec :
 - Quatre modes de format de transfert SPI.
 - Vitesses jusqu'à 80 MHz.
 - FIFO jusqu'à 64 octets.
- **UART** : Contient deux interfaces UART (UART0 et UART1) prenant en charge la communication asynchrone et les protocoles RS232, RS485 et IrDA.

-Méthodes de programmation pour l'ESP32 :

L'ESP32 prend en charge la programmation en C/C++ et MicroPython. Voici les environnements de développement compatibles avec chaque langage :

-C/C++ :

- ESP-IDF
- Arduino IDE
- VS Code

- MicroPython :

- Thonny IDE
- PyCharm
- Mu Editor
- uPyCraft IDE

II.5.3 Pilotage d'un servomoteur avec la carte ESP32

Pour câbler un servo à une carte ESP32, connectez le fil d'alimentation du servo à une source de tension appropriée, le fil de terre à la borne de terre de la carte, et le fil de signal à une broche GPIO de l'ESP32, comme illustré dans la figure II.40.

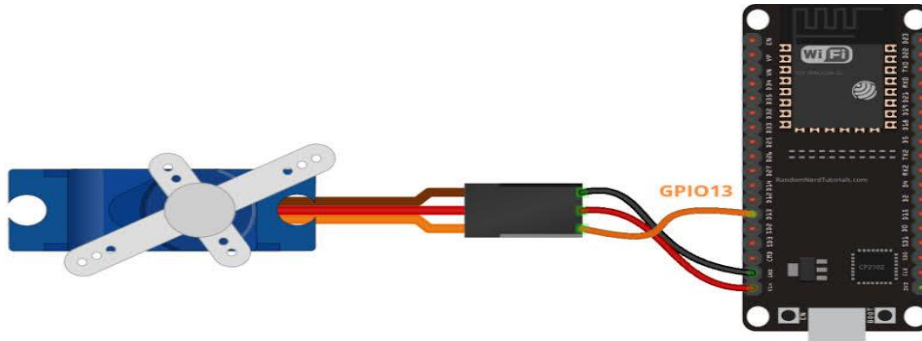


Figure II.39 : Pilotage d'un servomoteur par la carte ESP32.

II.6 Partie logicielle [49] [50]

Le logiciel utilisé pour programmer la carte Arduino est appelé un IDE, ce qui signifie Integrated Développement Environnent. En fait, cette application intègre l'édition de programmes, le téléchargement sur la carte Arduino ainsi que plusieurs bibliothèques.

II.6.1 Interface logicielle

Comme pour tout langage de programmation, cette interface flexible et simple peut être exécutée sur n'importe quel système d'exploitation, car Arduino est basé sur la programmation en langage C.

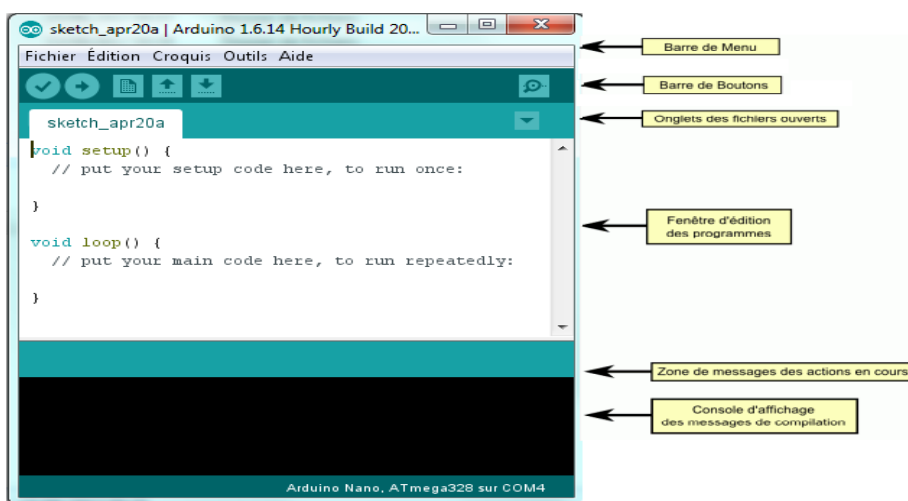


Figure II.40 : Présentation de la fenêtre IDE Arduino.

- 1 - Barre de menu (options de configuration du logiciel)
- 2 - Barre des boutons (la programmation des cartes)
- 3 - Fenêtre d'édition de programme
- 4 - débogueur (affichage des erreurs de programmation)

II.6.2 Le menu fichier

En double-cliquant sur l'icône de l'IDE, une fenêtre vide s'ouvre comme celle-ci-dessous. C'est le menu principal que nous utilisons le plus. Il contient de nombreuses fonctionnalités très utiles. La traduction en français est progressive, donc nous verrons plusieurs options inhabituelles, comme illustré dans la figure II.42.

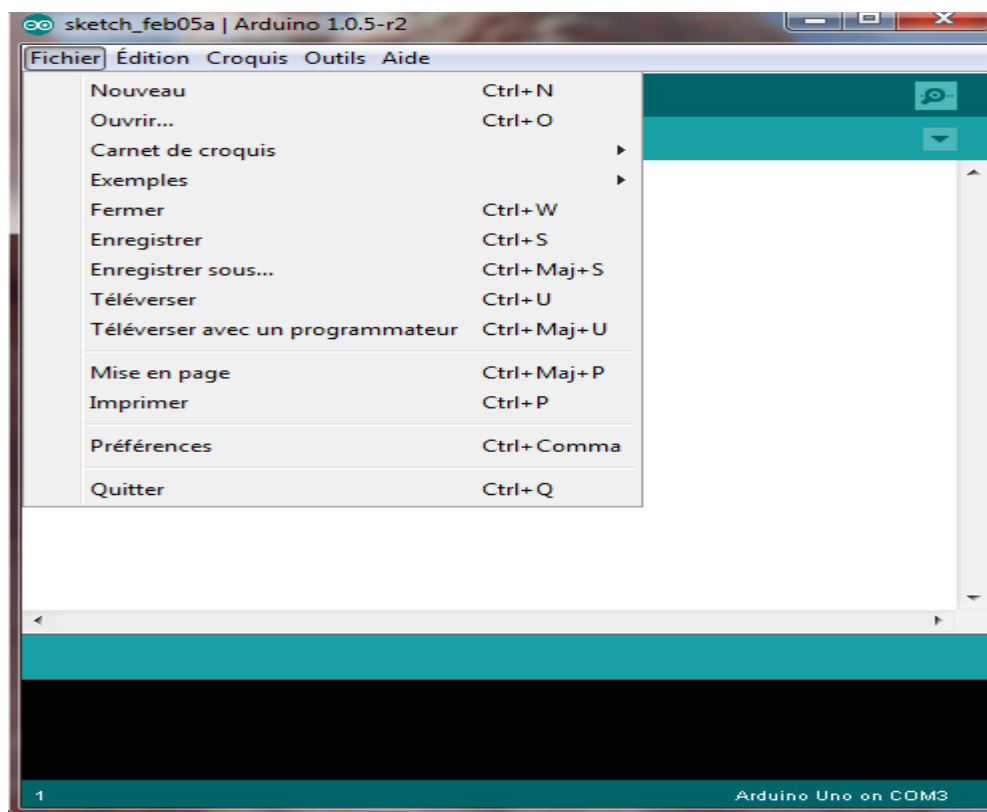


Figure II.41 : Contenu de menu fichier [51].

- **Nouveau** : Cette option vous permet de créer un nouveau programme. Lorsque vous appuyez sur ce bouton, une nouvelle fenêtre identique à celle-ci apparaît sur votre écran.
- **Ouvrir** : Avec cette commande, vous pouvez ouvrir un programme existant.
- **Enregistrer/Enregistrer sous...** : Permet d'enregistrer le document actuel ou de demander où le document actuel doit être enregistré.
- **Exemples** : Cette fonctionnalité est importante. La liste entière se développe pour afficher les noms des exemples de programmes existants.

- Sketchbook : Ce menu regroupe les fichiers que vous avez pu réaliser jusqu'à présent (s'ils sont enregistrés dans le dossier par défaut du logiciel).
- Téléverser : Permet d'envoyer votre programme vers la carte Arduino. Nous y reviendrons.
- Téléverser en utilisant le planificateur : Comme ci-dessus, mais en utilisant le planificateur.
- Préférences : Permet d'ajuster certains paramètres du logiciel.

II.6.3 Les boutons

Analysons l'utilisation du bouton rouge, comme le montre la figure II.43 :

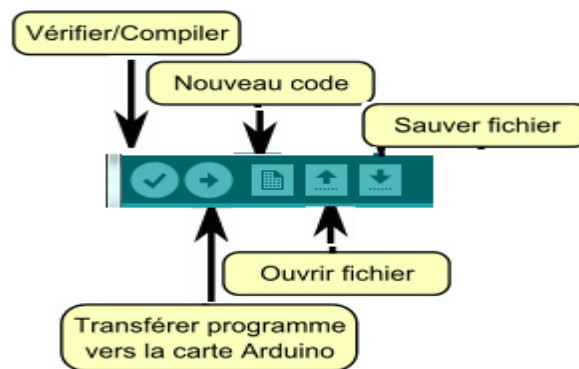


Figure II.42 : Présentation des boutons.

- Vérifier/compiler : Il active un module pour trouver les erreurs dans votre programme.
- Transférer vers la carte : Compilez le programme et envoyez-le à la carte.
- Nouveau code : Créez un nouveau fichier.
- Ouvrir fichier : Chargez le programme existant.
- Sauver fichier : Enregistrez le programme actuel.

II.6.4 Le moniteur série

Le logiciel Arduino intègre également :

- Le Terminal série (Figure II.15) permet d'afficher les messages texte reçus de la carte Arduino et d'envoyer des caractères à celle-ci. Cette fonctionnalité facilite le développement de programmes visant à afficher sur l'ordinateur l'état des variables, les résultats de calculs ou les conversions analogique-numérique : un élément essentiel pour améliorer, tester et corriger les programmes.
- Sur votre ordinateur, ouvrez la fenêtre du terminal de l'IDE Arduino en cliquant simplement sur le bouton "Serial Monitor".
- La fenêtre du "Terminal" s'ouvre ensuite.
- Réglez ensuite le débit de communication à la même valeur que celui utilisé dans le programme que vous souhaitez utiliser pour programmer la carte Arduino.

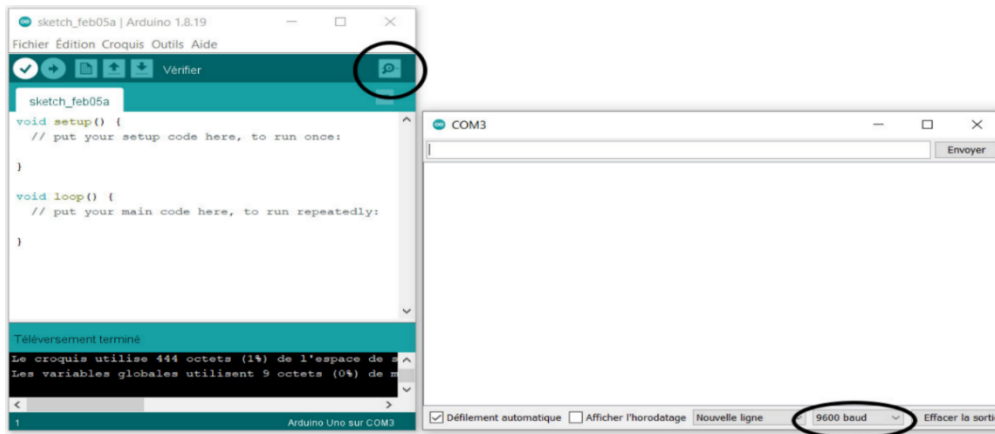


Figure II.43 : Le moniteur série [52].

II.6.5 Transfert des programmes vers la carte Arduino

- Pour une première programmation de la carte, visitez le menu Fichier > Exemples > Numéros > Clignotement : un programme s'ouvrira dans une fenêtre d'édition.
- Ensuite, appuyez sur le bouton « Vérifier » de la barre d'outils pour lancer la vérification du code.
- Si tout s'est bien passé, il ne devrait y avoir aucun message d'erreur dans la console et la boîte de message devrait indiquer "Compilation terminée", prouvant que la vérification a été effectuée correctement, comme le montre la figure II.45.

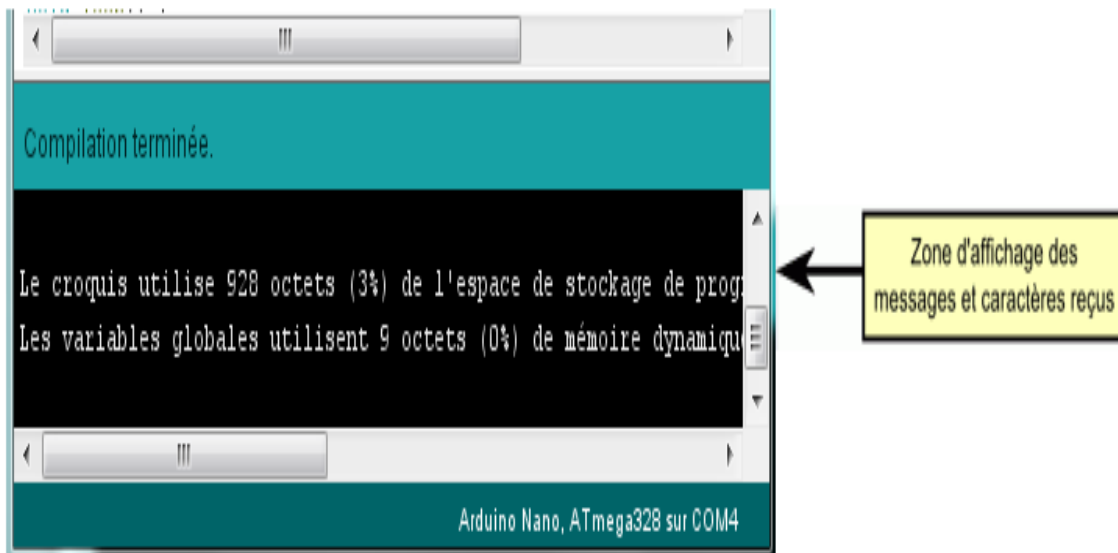


Figure II.44 : La console IDE Arduino.

Avant de transférer notre programme vers la carte Arduino, on s'assure d'avoir sélectionné la bonne carte Arduino dans le menu Outils (Tools) > Carte (Board). Les types de cartes sont décrits ci-dessous. Bien sûr, notre carte doit être connectée à l'ordinateur via un câble USB. On

doit également sélectionner le bon port série dans le menu Outils (Tools) > Port Série (Serial Port) :

Sur un Mac, le port série ressemble probablement à quelque chose comme /dev/tty.usbserial-1B1 (pour une carte USB), ou /dev/tty.USA19QW1b1P1.1 (pour une carte série connectée avec un adaptateur USB-vers-Série).

Sous Windows, il est probablement COM1 ou COM2 (pour une carte série) ou COM4, COM5, COM7 ou supérieur (pour une carte USB). Pour trouver le bon, on peut rechercher dans la section des ports série USB du panneau de configuration ou du gestionnaire de périphériques.

Sous Linux, il devrait être /dev/ttyUSB0, /dev/ttyUSB1 ou équivalent. Une fois qu'on a sélectionné le bon port série et la bonne carte Arduino, on clique sur le bouton Téléverser (Upload) dans la barre d'outils, ou on sélectionne le menu Croquis (Sketch) Téléverser (Upload) (Ctrl+U). Avec les versions récentes (notamment Duemilanove), la carte Arduino se réinitialisera automatiquement et commencera le transfert. Avec les versions antérieures qui ne disposent pas de l'auto-réinitialisation, on devra appuyer sur le bouton "reset" de la carte juste avant de commencer le transfert.

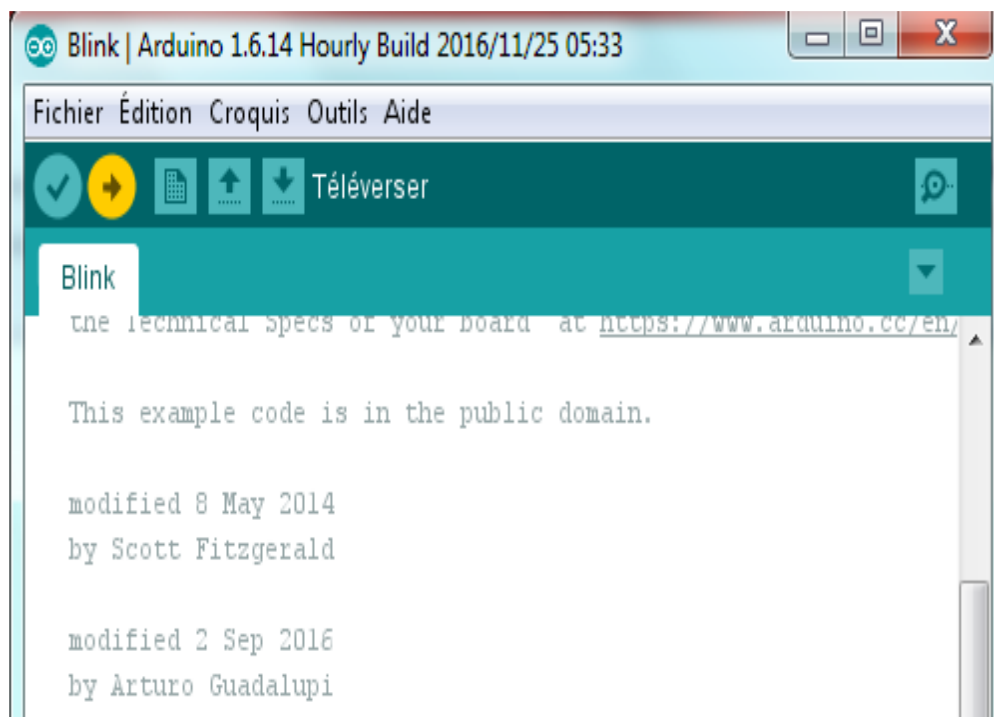


Figure II.45 : Le téléversement du programme vers la carte Arduino [53].

II.7 Langage HTML

Le HTML est un langage de balisage et non un langage de programmation car il ne permet pas les variables ou les fonctions. Il sert à créer et à structurer le contenu des pages web, il rend possible l'écriture d'hypertexte, la mise en forme du contenu, la création de champs de saisie, l'insertion d'image, vidéo. Le HTML utilise des balises entre <> pour décrire les divers éléments d'un document, dont des titres et des paragraphes. Chaque document HTML s'ouvre par une déclaration et les balises < html >, Il va organiser le contenu et les métadonnées. Il est souvent utilisé et combiné avec d'autres langages pour plus de style, dont le CSS [54].



Figure II.46 : Exemple d'un document HTML [55].

II.7.1 Les Balises HTML

Voici un résumé des principales balises HTML et leur utilisation (Tableau II.2) :

Balises	Utilisation
Html	Définit le début du document html.
Head	Contient les informations de configuration pour le document html comme les titres.
Title	Définit le titre de la page ; élément de head.
Style	Définit un élément de code CSS.

Body	Définit le corps de fichier html tels que le texte, les images,...etc.
Table	Créer un tableau.
Tr/Td	Tr : Définit une ligne de tableau. Td : Définit une cellule de tableau.
P	Créer un paragraphe.
H	Les titres et les sous-titres sont indiqués par la balise h1 à h6 (six niveaux de titres).

Tableau II.3 : Les balises de langage HTML.

II.8 Serveur Web

Un serveur Web est un logiciel ou un matériel informatique qui stocke, traite et distribue des pages Web, des images, des vidéos et d'autres informations aux utilisateurs via un navigateur Internet. Un serveur HTTP est un logiciel qui peut comprendre les requêtes HTTP et les URL. Un serveur Web se connecte à Internet, ce qui lui permet de communiquer et d'échanger des données ou des fichiers avec d'autres appareils connectés. Les données peuvent avoir de nombreux formats, allant des fichiers HTML aux fichiers JavaScript aux feuilles de style CSS [56].

Il est également important de remarquer qu'un serveur Web fait généralement partie des programmes suivants :

- la création et la distribution de pages Web.
- la demande de fichiers à l'aide du protocole FTP.
- l'envoi et la réception.

II.8.1 Fonctionnement d'un serveur Web

Un serveur WEB fonctionne selon un modèle client/serveur puisque d'une part nous avons le serveur WEB et d'autre part le client du navigateur internet. Les clients sont les appareils connectés à Internet de l'utilisateur Web typiques, tel qu'un ordinateur connecté au Wifi ou un téléphone mobile connecté à un réseau mobile, et un logiciel d'accès Web disponible sur ces appareils généralement un navigateur Web tel que Firefox, ou Google Chrome.

Lorsqu'un dispositif client souhaite accéder à une page Web, une copie de la page Web est téléchargée du serveur vers la machine client pour être visualisée dans le navigateur Web de l'utilisateur. Chaque fois qu'un navigateur a besoin d'un fichier hébergé sur un serveur Web, le navigateur demande le fichier via HTTP. Lorsqu'une requête arrive sur le serveur Web, le serveur HTTP (logiciel) accepte la requête, trouve le document demandé et le renvoie au navigateur, également via HTTP.

Si le serveur ne parvient pas à livrer le document, il renvoie un code HTTP avec un numéro spécifique. Par exemple, si le document demandé est introuvable, une réponse 404 (Not Found) est renvoyée. Si le client n'a pas l'autorisation d'accéder au contenu, il renvoie une réponse 403 (forbidden) [57].

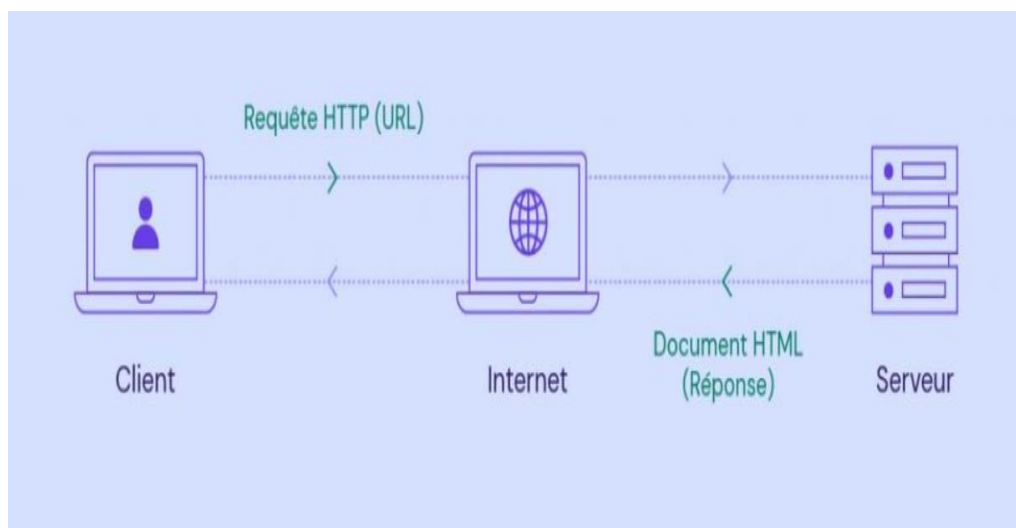


Figure II.47 : L'architecteur client-serveur [58].

II.9 Discussion

En synthèse, la modélisation de notre robot sériel à 4 degrés de liberté a impliqué plusieurs étapes. Tout d'abord, nous avons décrit mécaniquement les composants du robot. Ensuite, nous avons développé un modèle géométrique direct pour représenter le comportement du robot et décrire la position de son effecteur. Enfin, nous avons élaboré un modèle mathématique pour le moteur à courant continu. Dans cette démarche, notre objectif était de trouver le modèle le plus simple mais satisfaisant pour expliquer le comportement du processus dans son domaine d'application. Dans ce contexte, le modèle géométrique direct a été retenu pour la commande de notre bras robotique.

Dans le dernier chapitre, nous nous consacrerons à la réalisation de notre travail et à ses différentes étapes de conception.

Chapitre III

Réalisation pratique

III.1 Préambule

La réalisation d'un bras robotique demande une conception détaillée et précise pour garantir un résultat fiable. En général, un robot manipulateur vise à positionner son organe terminal dans l'espace. Pour cela, la commande des articulations du bras robotique est essentielle, nécessitant souvent un contrôle individuel pour chaque articulation afin d'accomplir une tâche donnée. Dans ce chapitre, nous nous concentrons sur la commande du bras robotique en utilisant la carte de commande Arduino et éventuellement la carte ESP32 pour contrôler les servomoteurs. Enfin, nous aborderons l'étape d'implémentation ou de tests de fonctionnement, où nous analyserons et interpréterons les résultats obtenus.

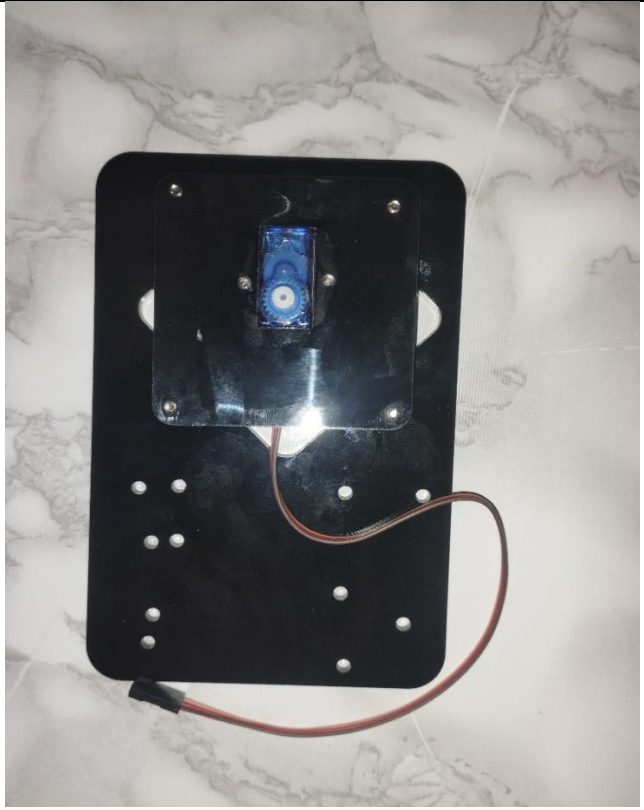
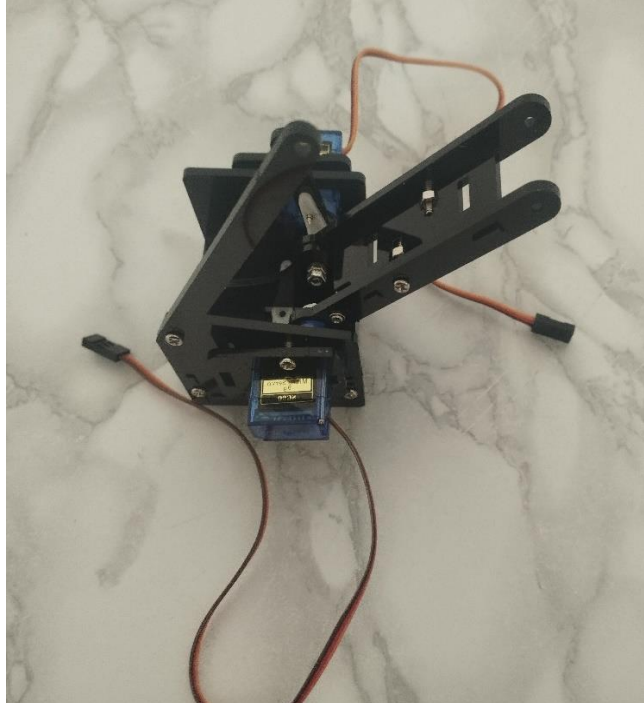
III.2 Les éléments fondamentaux requis pour le bon fonctionnement du bras robotique

Pour que le bras robotique fonctionne, plusieurs éléments sont nécessaires :

- La carte de commande (Arduino Uno, Esp32) avec son câble USB.
- Quatre servomoteurs.
- Les joysticks.
- Un ordinateur.
- Smartphone.
- Le bras que nous avons réalisé.
- Alimentation 12 V / 2A.
- Les fils de connexion

III.3 Montage du bras

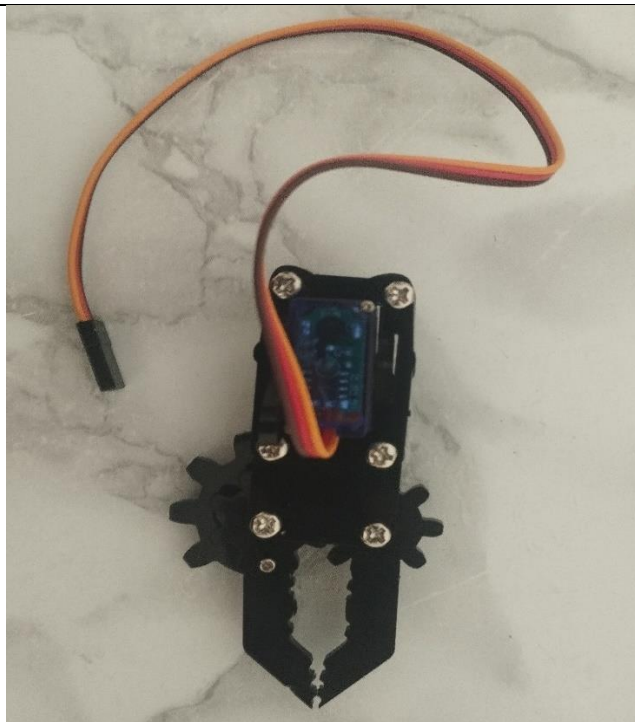
Le tableau ci-dessous illustre les étapes du montage du bras robotique.

Composants	Étapes de montage
	<p>Étape 1 : Montage de la base :</p> <ul style="list-style-type: none"> • Nous avons installé le socle pour le servomoteur, permettant ainsi au bras de pivoter à 360 degrés autour de lui-même.
	<p>Étape 2 : Assemblage de la partie centrale :</p> <ul style="list-style-type: none"> • Assemblage de pièce centrale du bras. • Fixez les servomoteurs au bras gauche. • Attachez le support du servomoteur à la plaque inférieure du bras. • Installez le deuxième servomoteur au bras droit. • Montez le joint du deuxième servomoteur. • Assemblez l'articulation du poignet avec la servomoteur du bras gauche.



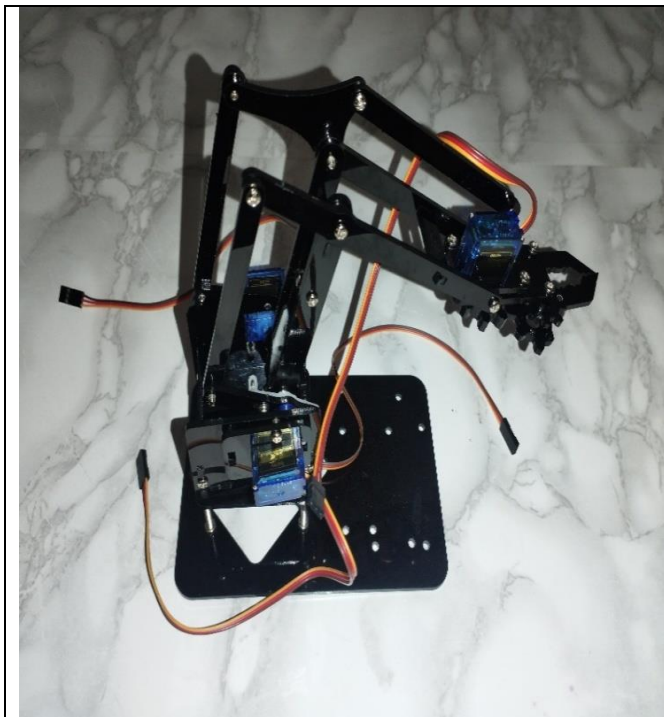
Étape 3 :

- installer le connecteur d'articulation du poignet
- Entre l'articulation du poignet gauche et droite.



Étape 4 :

- Montez les engrenages ensemble.
- Ensuite, utilisez la pince pour les fixer les engrenages entre eux.
- connectez le servomoteur avec la pince.



Étape 5 :

- Montez tous les composants ensemble en les assemblant les uns aux autres.

Tableau III.4 : Montage du bras.

III.4 Câblages des servomoteurs et des joysticks avec la carte Arduino

Nous avons besoin de quatre servomoteurs et d'une plaque d'essai. Cette dernière sera connectée à une entrée numérique. Ensuite, il suffira de mettre à l'échelle le signal du servomoteur pour obtenir un angle compris entre 0 et 180 degrés. L'objectif est de recevoir une information analogique.

Nous utilisons les sorties du microcontrôleur qui correspondent aux broches d'Arduino. La borne jaune de chaque servomoteur est connectée à ces sorties.

Les fils d'entrée PWM (jaunes) ainsi que les fils de retour pour chaque servomoteur seront fixés sur les broches de la carte selon le schéma suivant :

- Servo 01 : PWM -> la broche 3
- Servo 02 : PWM -> la broche 5
- Servo 03 : PWM -> la broche 6
- Servo 04 : PWM -> la broche 9

Pour les joysticks, nous aurons besoin de deux joysticks qui seront connectés à des entrées analogiques. L'objectif est de transformer le signal analogique en signal numérique, afin de l'envoyer au servomoteur.

Pour transformer le signal analogique des joysticks en signal numérique afin de les envoyer aux servomoteurs, voici les étapes que nous avons suivies :

- Connexion des joysticks : Les joysticks devront être connectés aux entrées analogiques du microcontrôleur. Chaque joystick aura généralement deux axes (X et Y), ce qui signifie que nous aurons besoin de quatre entrées analogiques (A0, A1, A2, A3).
- Lecture des valeurs analogiques : Nous devons utiliser le code pour lire les valeurs analogiques des joysticks à partir des entrées analogiques du microcontrôleur. Cela nous donnera des valeurs analogiques qui représentent la position des joysticks sur les axes X et Y.
- Conversion en signal numérique : Les valeurs analogiques lues devront être converties en signal numérique en les adaptant à la plage de valeurs des servomoteurs de 0 à 180 degrés.
- Boucle de contrôle : Nous devons répéter ce processus en boucle afin de lire en continu les valeurs des joysticks, les convertir en signaux numériques et contrôler les servomoteurs en fonction de ces valeurs.

Nous avons également veillé à fournir une alimentation suffisante en utilisant une alimentation externe de 12V/2A.

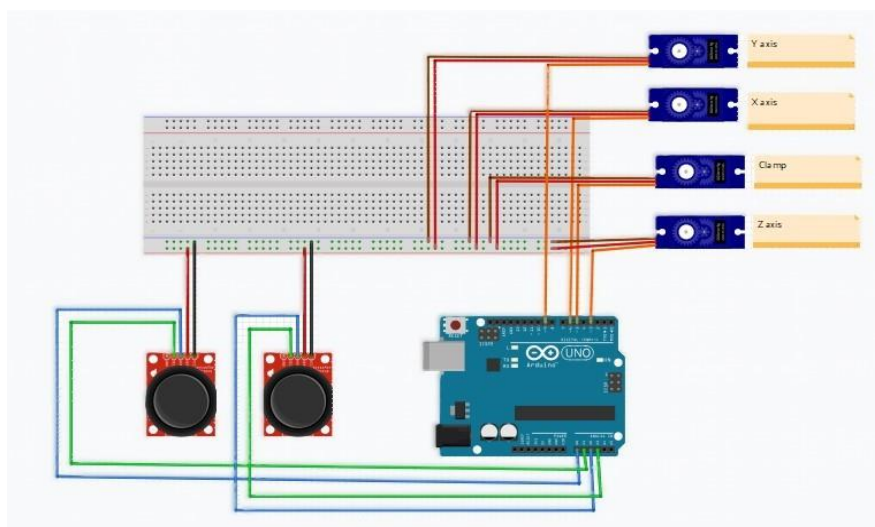


Figure III.48 : Câblages de la commande manuelle.

III.5 Câblages des servomoteurs avec la carte ESP32

Nous avons également besoin de quatre servomoteurs. Chacun d'eux sera connecté à une entrée numérique de la carte ESP32 correspondante. Chaque fil jaune du servomoteur sera branché sur une broche numérique de l'ESP32 comme suit :

- Servo 01 : PWM -> broche 25
- Servo 02 : PWM -> broche 26
- Servo 03 : PWM -> broche 27
- Servo 04 : PWM -> broche 33

Nous avons également besoin d'une alimentation de 5V/2A.

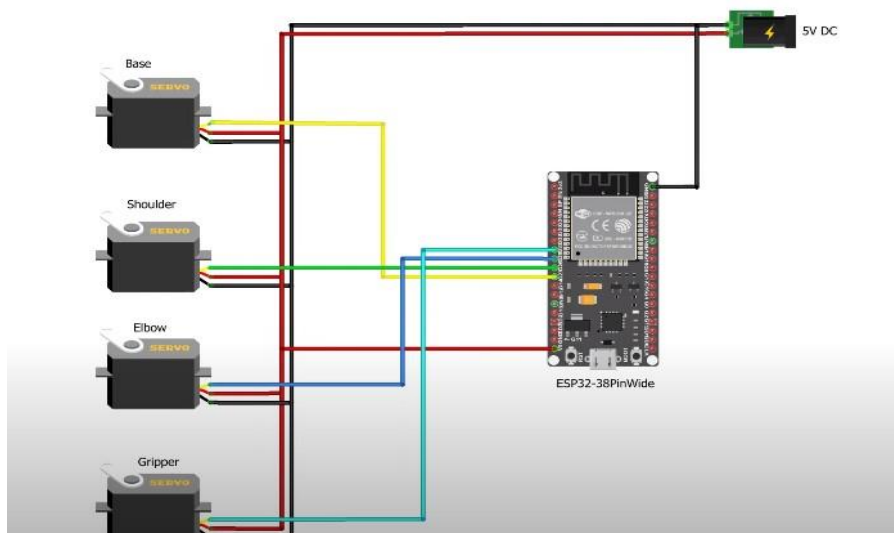


Figure II.49 : Câblage de la commande automatique.

III.6 La commande manuelle

III.6.1 Description

Le programme utilise deux joysticks analogiques connectés à un Arduino pour contrôler les mouvements d'un bras robotique. Chaque joystick contrôle deux axes du bras, permettant de déplacer les axes x, y et z, ainsi que d'ouvrir ou fermer la pince (clamp). Les positions des servos sont mises à jour en fonction des valeurs des joysticks, avec des zones mortes intégrées pour

éviter les mouvements involontaires. De plus, le programme limite les angles des servos afin de prévenir des positions dangereuses. Pour faciliter le débogage, les positions des servos sont affichées sur le moniteur série.

Pour ce faire, nous avons utilisé l'IDE Arduino, qui permet de vérifier et de compiler plusieurs programmes. L'avantage est de pouvoir tester et compiler le programme avant de le charger sur le microcontrôleur. Ce programme est écrit en langage C, un langage de programmation impératif adapté à la carte. Son principal atout est l'utilisation de fonctions préinstallées grâce à des bibliothèques, simplifiant ainsi le code.

III.6.2 Structure du programme

La structure du programme se compose de deux parties Déclarations et Initialisation.

-Inclusion de la bibliothèque Servo

```
#include <Servo.h>
```

La bibliothèque Servo est utilisée pour contrôler les servomoteurs.

-Déclaration des servos

```
Servo servo_z_axis;  
Servo servo_x_axis;  
Servo servo_y_axis;  
Servo servo_clamp;
```

Quatre objets Servos sont créés pour contrôler les différents axes et la pince du bras robotique.

-Définition des positions initiales des servos

```
int x_axis_degree = 90;  
int y_axis_degree = 90;  
int z_axis_degree = 85;  
int clamp_degree = 90;
```

Les variables `x_axis_degree`, `y_axis_degree`, `z_axis_degree` et `clamp_degree` stockent les positions initiales des servos en degrés.

-Définition des broches des joysticks

```
#define left_joystick_x A0
#define left_joystick_y A1
#define right_joystick_x A2
#define right_joystick_y A3
```

Les broches analogiques où les joysticks sont connectés sont définies à l'aide de macros.

• Fonction setup

-Initialisation de la communication série

```
void setup() {
  Serial.begin(9600);
}
```

La communication série est initialisée à 9600 baud pour le débogage.

-Attachement des servos aux broches correspondantes

```
servo_z_axis.attach(3);
servo_clamp.attach(5);
servo_x_axis.attach(6);
servo_y_axis.attach(9);
}
```

Chaque servo est attaché à une broche numérique spécifique de l'Arduino.

• Fonction Loop

-Lecture des valeurs des joysticks

```
void loop() {
  int left_joystick_x_value = analogRead(left_joystick_x);
  int left_joystick_y_value = analogRead(left_joystick_y);
  int right_joystick_x_value = analogRead(right_joystick_x);
  int right_joystick_y_value = analogRead(right_joystick_y);
}
```

Les valeurs analogiques des joysticks sont lues.

-Mise à jour des angles des servos en fonction des joysticks

```
if(left_joystick_x_value < 340) y_axis_degree -=7;
else if(left_joystick_x_value > 680) y_axis_degree +=7;

if(left_joystick_y_value < 340) clamp_degree -=5;
else if(left_joystick_y_value > 680) clamp_degree +=5;

if(right_joystick_x_value < 340) x_axis_degree -=7;
else if(right_joystick_x_value > 680) x_axis_degree +=7;

if(right_joystick_y_value < 340) z_axis_degree -=7;
else if(right_joystick_y_value > 680) z_axis_degree +=7;
```

Les angles des servos sont ajustés en fonction des valeurs des joysticks. Si une valeur est inférieure à 340, l'angle correspondant est diminué ; si elle est supérieure à 680, l'angle est augmenté. Cela crée une zone morte (Dead zone) pour éviter les mouvements involontaires lorsque les joysticks sont proches de la position centrale.

-Limitation des angles des servos

```
z_axis_degree = min(145, max(15, z_axis_degree));
x_axis_degree = min(175, max(40, x_axis_degree));
y_axis_degree = min(150, max(5, y_axis_degree));
clamp_degree = min(90, max(75, clamp_degree));
```

Les angles des servos sont limités à des valeurs maximales et minimales pour éviter des mouvements dangereux ou des positions impossibles.

-Affichage des positions des servos sur le moniteur série

```
Serial.print("x_axis_degree : ");
Serial.print(x_axis_degree);
Serial.print(", y_axis_degree : ");
Serial.print(y_axis_degree);
Serial.print(", z_axis_degree 4 : ");
Serial.print(z_axis_degree);
Serial.print(", clamp_degree : ");
Serial.println(clamp_degree);
```

Les positions actuelles des servos sont affichées sur le moniteur série pour le débogage.

-Mouvement des servos aux nouvelles positions

```
servo_clamp.write(clamp_degree);  
servo_x_axis.write(x_axis_degree);  
servo_y_axis.write(y_axis_degree);  
servo_z_axis.write(z_axis_degree);  
}
```

Les servos sont déplacés vers les nouvelles positions calculées.

III.7 La commande automatique

III.7.1 Description

Ce programme contrôle un bras robotique à l'aide d'un ESP32, utilisant des servomoteurs et une interface web pour recevoir les commandes. Il permet également l'enregistrement et la reproduction des mouvements du bras.

Pour cela, le programme est écrit en langage C pour la partie commande du bras et en langage HTML pour la partie web, rendant la connexion accessible à partir d'un smartphone ou d'un PC.

III.7.2 Structure du programme

-Déclaration des Bibliothèques

```
#include <Arduino.h>  
#include <WiFi.h>  
#include <AsyncTCP.h>  
#include <ESPAsyncWebServer.h>  
  
#include <ESP32Servo.h>  
#include <iostream>  
#include <sstream>
```

Ces lignes incluent les bibliothèques nécessaires pour gérer les servos, la connexion WiFi et le serveur web asynchrone.

-Déclaration des Variables :

Définition des Servos

```
struct ServoPins
{
  Servo servo;
  int servoPin;
  String servoName;
  int initialPosition;
};
std::vector<ServoPins> servoPins =
{
  { Servo(), 27 , "Base", 90},
  { Servo(), 26 , "Shoulder", 90},
  { Servo(), 25 , "Elbow", 90},
  { Servo(), 33 , "Gripper", 90},
};
```

Cette structure stocke les informations pour chaque servo, y compris l'objet servo, le pin auquel il est connecté, son nom et sa position initiale.

-Enregistrement des Étapes

```
struct RecordedStep
{
  int servoIndex;
  int value;
  int delayInStep;
};
std::vector<RecordedStep> recordedSteps;

bool recordSteps = false;
bool playRecordedSteps = false;

unsigned long previousTimeInMilli = millis();
```

Ces variables et structures permettent d'enregistrer et de reproduire les mouvements des servos.

-Connexion Wifi

```
const char* ssid    = "RobotArm";
const char* password = "12345678";

AsyncWebServer server(80);
AsyncWebSocket wsRobotArmInput("/RobotArmInput");
```

Ces lignes définissent les informations de connexion Wifi et initialisent le serveur web et le WebSocket.

-Interface Web

```
const char* htmlHomePage PROGMEM = R"HTMLHOMEPAGE(
<!DOCTYPE html>
<html>
  <head>
    <meta name="viewport" content="width=device-width, initial-scale=1, maximum-scale=1, user-scalable=no">
    <style>
      input[type=button]
      {
        background-color:red;color:white;border-radius:30px;width:100%;height:40px;font-size:20px;text-align:center;
      }
      .noselect {
        -webkit-touch-callout: none; /* iOS Safari */
        -webkit-user-select: none; /* Safari */
        -khtml-user-select: none; /* Konqueror HTML */
        -moz-user-select: none; /* Firefox */
        -ms-user-select: none; /* Internet Explorer/Edge */
        user-select: none; /* Non-prefixed version, currently
        supported by Chrome and Opera */
      }
    )
```

Ce bloc de code contient le HTML de l'interface web, qui permet à l'utilisateur de contrôler les servos via des curseurs et des boutons. Il utilise JavaScript pour envoyer les commandes via WebSocket.

• Fonctions de Gestion des Requêtes**- Gestion des Requêtes http**

```
void handleRoot(AsyncWebServerRequest *request)
{
  request->send_P(200, "text/html", htmlHomePage);
}

void handleNotFound(AsyncWebServerRequest *request)
{
  request->send(404, "text/plain", "File Not Found");
}
```

Ces fonctions gèrent les requêtes HTTP pour la page principale et les requêtes non trouvées.

- Gestion des Événements WebSocket

```
void onRobotArmInputWebSocketEvent(AsyncWebSocket *server,
                                   AsyncWebSocketClient *client,
                                   AwsEventType type,
                                   void *arg,
                                   uint8_t *data,
                                   size_t len)
{
```

Cette fonction gère les différents types d'événements WebSocket, comme les connexions, les déconnexions et la réception de données.

- **Mise à Jour des Servos**

- **Envoi de l'État Actuel des Servos**

```
void sendCurrentRobotArmState()
{
    for (int i = 0; i < servoPins.size(); i++)
    {
        wsRobotArmInput.textAll(servoPins[i].servoName + "," + servoPins[i].servo.read());
    }
    wsRobotArmInput.textAll(String("Record,") + (recordSteps ? "ON" : "OFF"));
    wsRobotArmInput.textAll(String("Play,") + (playRecordedSteps ? "ON" : "OFF"));
}
```

Cette fonction envoie l'état actuel de chaque servo via WebSocket.

- **Écriture des Valeurs des Servos**

```
void writeServoValues(int servoIndex, int value)
{
    if (recordSteps)
    {
        RecordedStep recordedStep;
        if (recordedSteps.size() == 0) // We will first record initial position of all servos.
        {
            for (int i = 0; i < servoPins.size(); i++)
            {
                recordedStep.servoIndex = i;
                recordedStep.value = servoPins[i].servo.read();
                recordedStep.delayInStep = 0;
                recordedSteps.push_back(recordedStep);
            }
        }
        unsigned long currentTime = millis();
        recordedStep.servoIndex = servoIndex;
        recordedStep.value = value;
        recordedStep.delayInStep = currentTime - previousTimeInMilli;
        recordedSteps.push_back(recordedStep);
        previousTimeInMilli = currentTime;
    }
    servoPins[servoIndex].servo.write(value);
}
```

Cette fonction écrit les valeurs des servos et enregistre les mouvements si l'enregistrement est activé.

-Lecture des Étapes Enregistrées

```

void playRecordedRobotArmSteps()
{
  if (recordedSteps.size() == 0)
  {
    return;
  }
  //This is to move servo to initial position slowly. First 4 steps are initial position
  for (int i = 0; i < 4 && playRecordedSteps; i++)
  {
    RecordedStep &recordedStep = recordedSteps[i];
    int currentServoPosition = servoPins[recordedStep.servoIndex].servo.read();
    while (currentServoPosition != recordedStep.value && playRecordedSteps)
    {
      currentServoPosition = (currentServoPosition > recordedStep.value ? currentServoPosition - 1 : currentServoPosition + 1);
      servoPins[recordedStep.servoIndex].servo.write(currentServoPosition);
      wsRobotArmInput.textAll(servoPins[recordedStep.servoIndex].servoName + ", " + currentServoPosition);
      delay(50);
    }
  }
}

```

Cette fonction reproduit les mouvements enregistrés des servos.

• Configuration et Boucle Principale**- Configuration Initiale**

```

void setUpPinModes()
{
  for (int i = 0; i < servoPins.size(); i++)
  {
    servoPins[i].servo.attach(servoPins[i].servoPin);
    servoPins[i].servo.write(servoPins[i].initialPosition);
  }
}

void setup(void)
{
  setUpPinModes();
  Serial.begin(115200);

  WiFi.softAP(ssid, password);
  IPAddress IP = WiFi.softAPIP();
  Serial.print("AP IP address: ");
  Serial.println(IP);

  server.on("/", HTTP_GET, handleRoot);
  server.onNotFound(handleNotFound);

  wsRobotArmInput.onEvent(onRobotArmInputWebSocketEvent);
  server.addHandler(&wsRobotArmInput);

  server.begin();
  Serial.println("HTTP server started");
}

```

- Boucle Principale

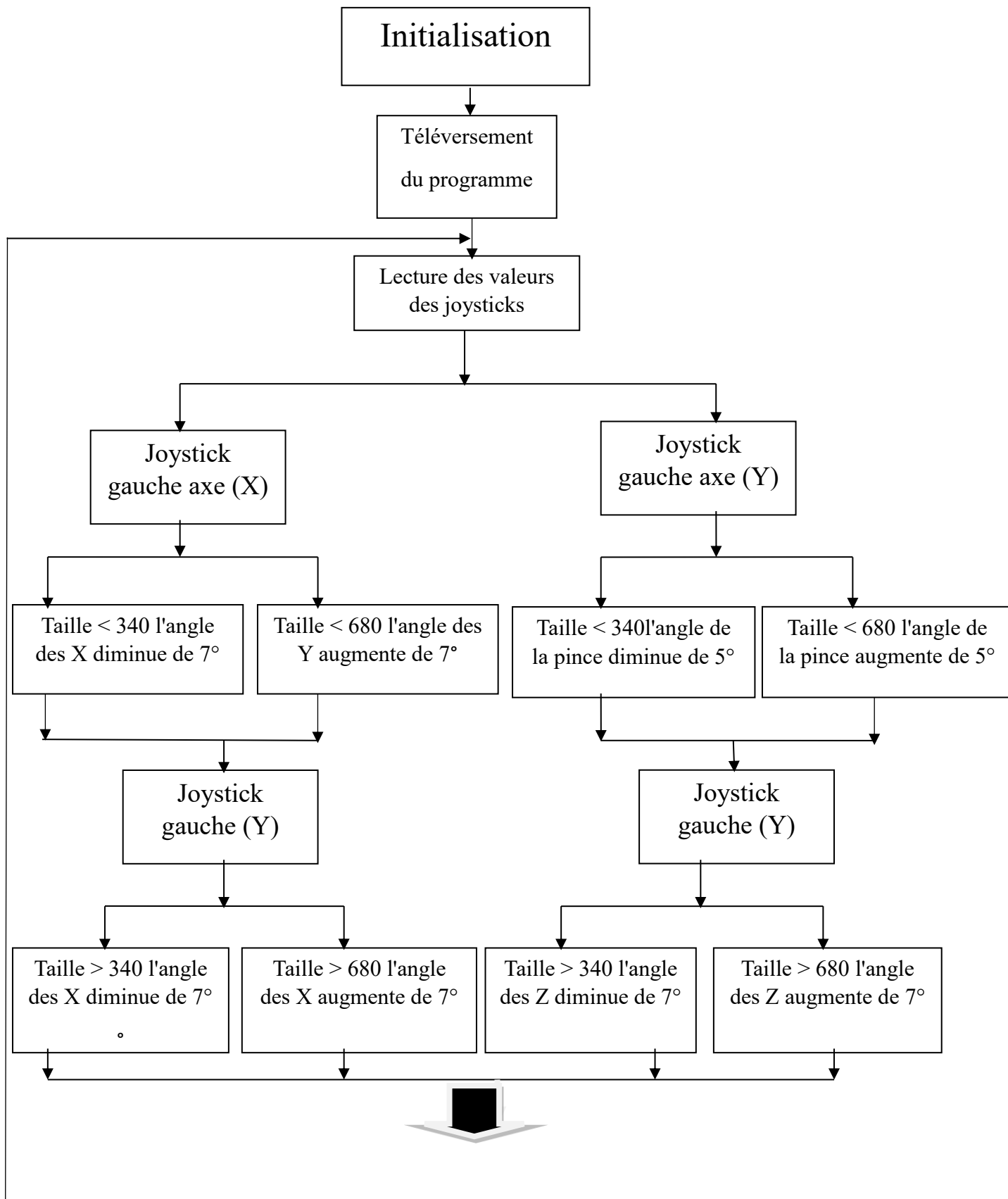
```
void loop()
{
  wsRobotArmInput.cleanupClients();
  if (playRecordedSteps)
  {
    playRecordedRobotArmSteps();
  }
}
```

- La méthode `cleanupClients()` est appelée pour nettoyer les clients WebSocket.
- Si la variable `playRecordedSteps` est vraie, la fonction `playRecordedRobotArmSteps ()` est appelée pour reproduire les mouvements enregistrés du bras robotique.

Nous donnons ci-après, l'algorithme de fonctionnement de la commande manuelle des articulations de notre bras manipulateur.

III.8 L'organigramme de la commande manuelle

La figure III.50 illustre l'organigramme de la commande manuelle.



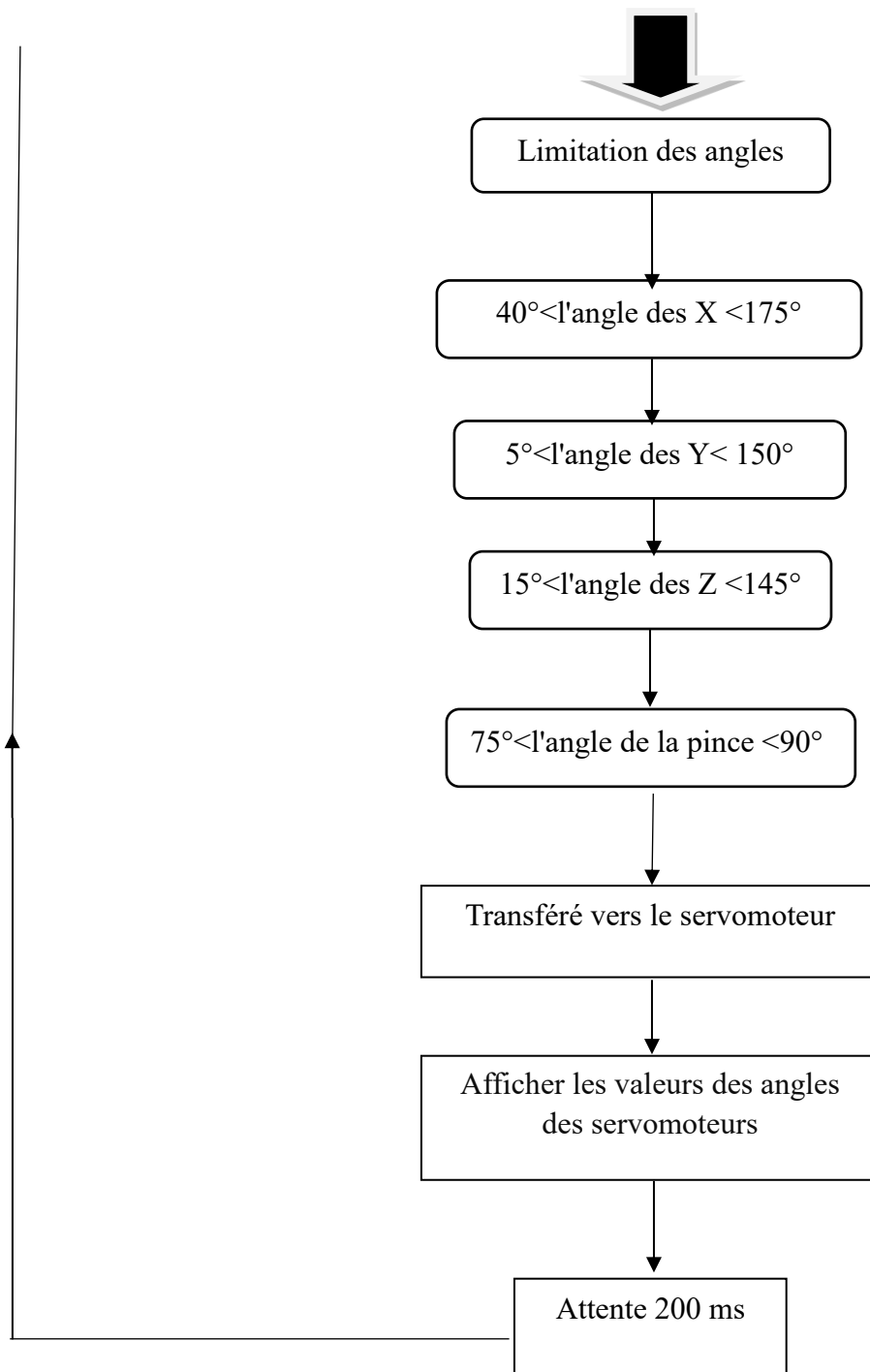


Figure III.50 : L’organigramme de fonctionnement de la commande manuelle.

III.9 L'organigramme de la commande automatique

La figure III.51 illustre l'organigramme de la commande automatique.

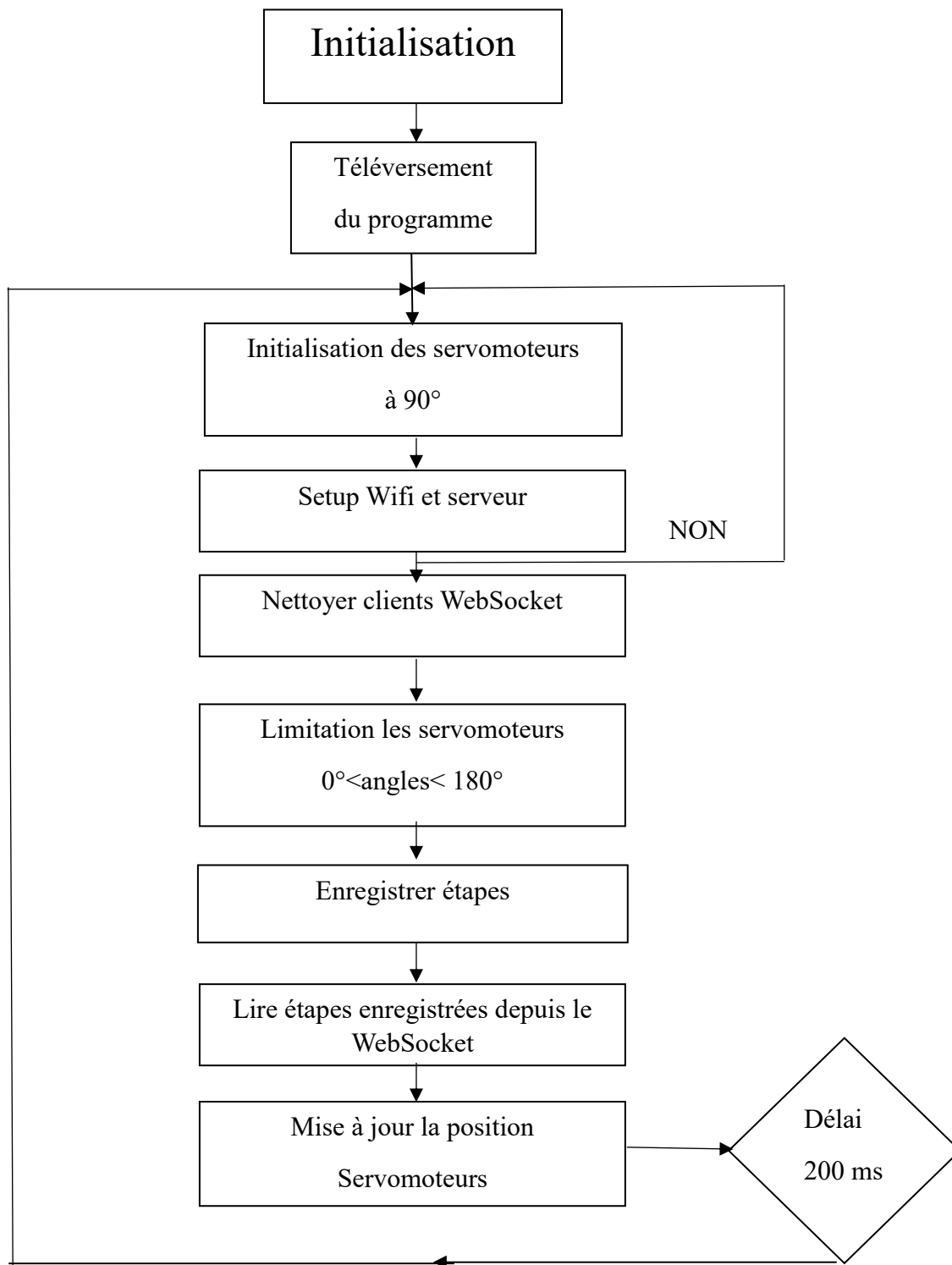


Figure III.51 : L'organigramme de fonctionnement de la commande automatique.

III.10 Les actions de tests du bras :

Pour la partie test, nous avons organisé deux rubriques : à gauche, c'est la commande manuelle avec les joysticks, et à droite, c'est la partie automatique commandée à l'aide du smartphone. La figure ci-dessus représente la position initiale.

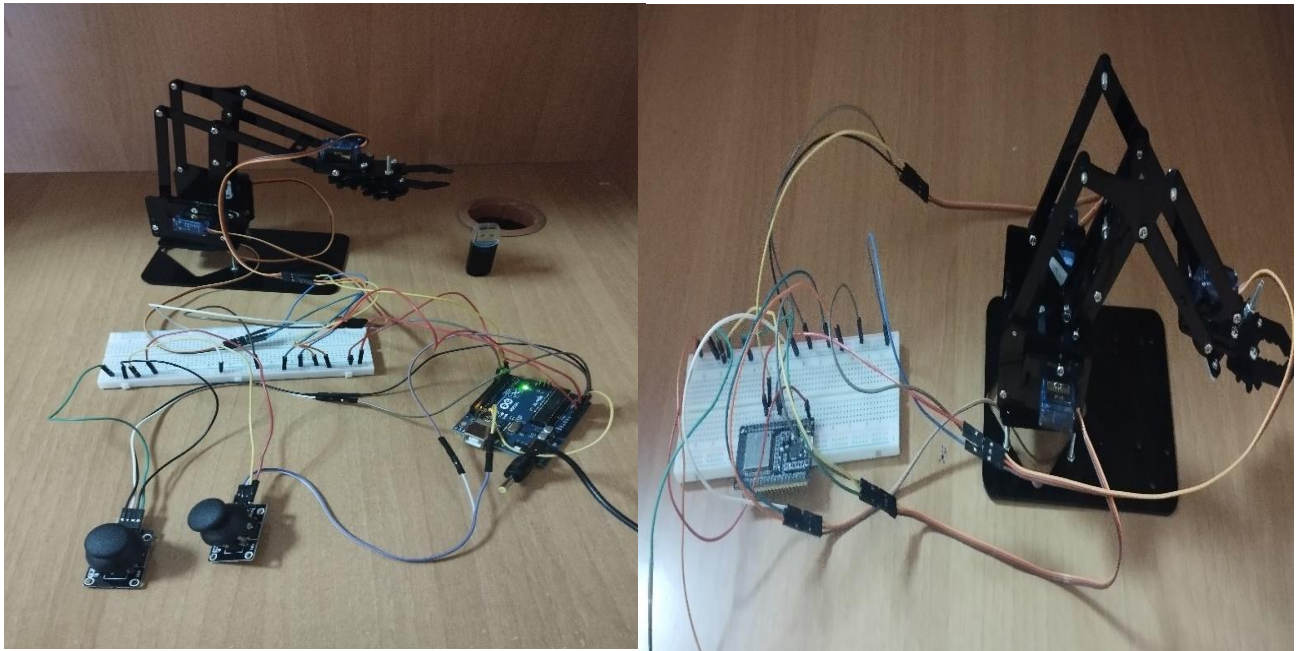


Figure III.52 : Position initiale.

Nous avons numéroté les servomoteurs de 1 à 4 :

- Servomoteur 1 : Pour la rotation horizontale du disque rotatif.
- Servomoteur numéro 2,3 : Pour la rotation verticale du bras inférieur.
- Servomoteur numéro 4 : Pour l'ouverture et la fermeture de la pince.

Pour contrôler les quatre servomoteurs, nous avons testé tous les mouvements du bras et nous allons les présenter comme suit :

• position bras allongé

Les servomoteurs numéro 2 et 3 sont actionnés pour faire descendre le bras.

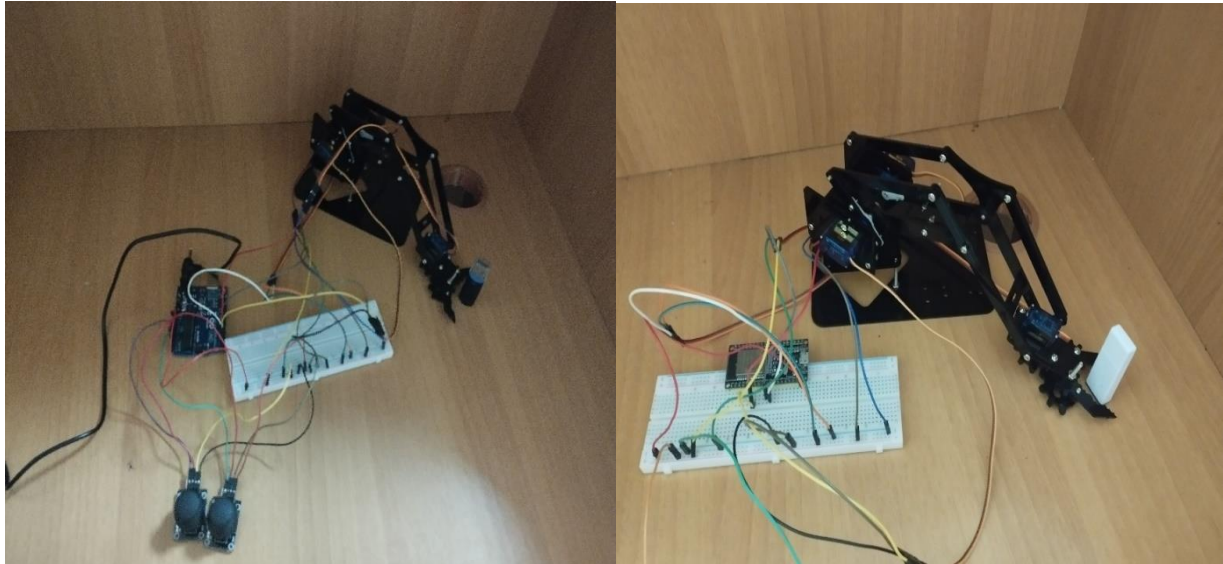


Figure III.53 : Position du bras allongé.

• position rotation

Le servomoteur 1 est actionné pour exécuter la rotation du bras, la pince (servo 4) est fermée pour agripper l'objet. La figure ci-dessous représente cette étape.

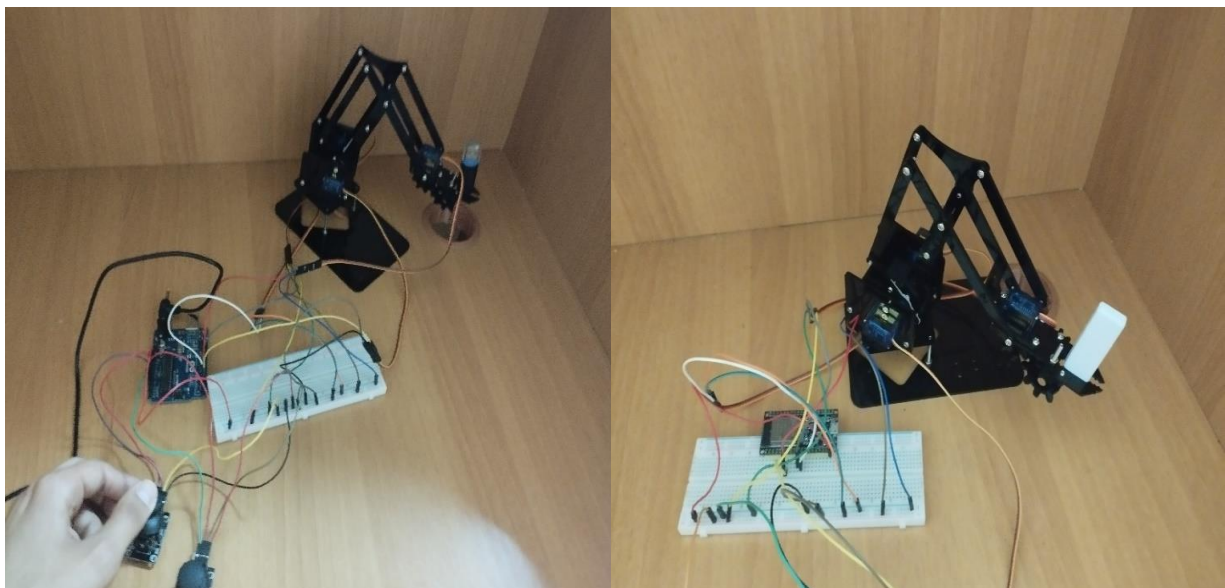
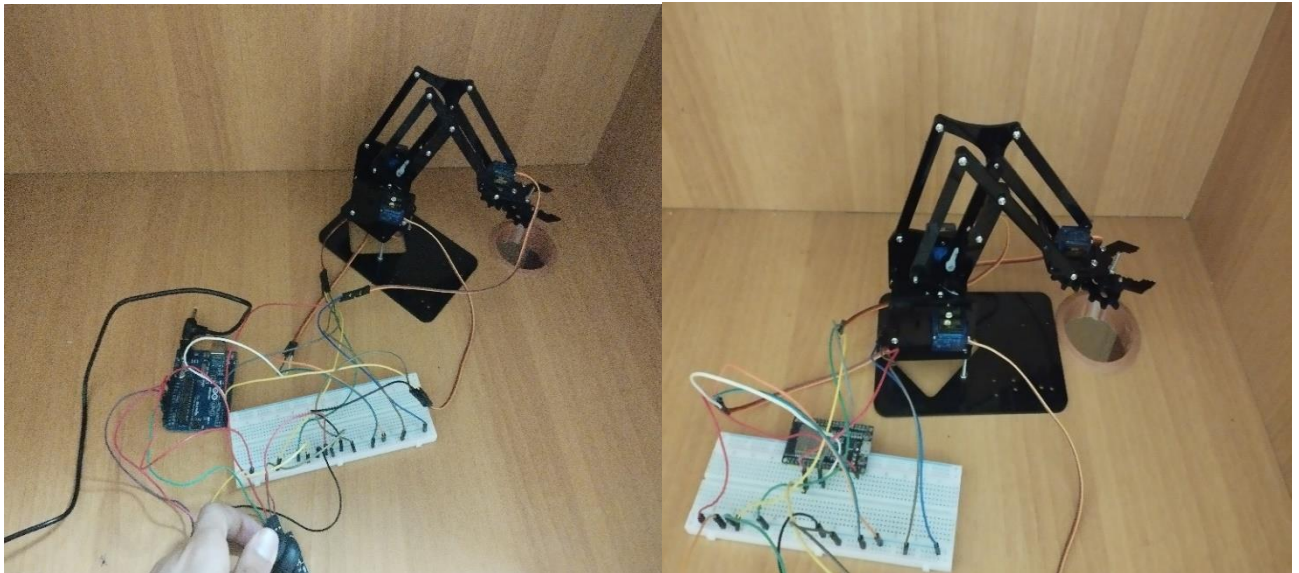


Figure III.54 : Position de la rotation du bras.

• position de relâchement

La pince (servo 4) est relâchée et l'objet est tombé dans le trou, comme le montre la figure.



La figure III.55 : Position du robot pendant qu'il lâche l'objet.

Pour la commande automatique, on appuie sur le bouton Record (ON) pour enregistrer les étapes, puis sur Play (ON) pour répéter les étapes automatiquement. La figure III.75 montre les boutons de l'application.



Figure III.56 : Enregistrement et exécution de la commande automatique.

III.11 discussion

Ce chapitre explique comment mettre en place le bras robotique en s'assurant que tout le matériel et les logiciels fonctionnent bien ensemble. Pour rendre le bras fonctionnel, il est crucial que tous les équipements matériels et logiciels collaborent efficacement. À cet effet, nous avons décrit en détail le processus d'assemblage du bras, les composants essentiels pour son fonctionnement, ainsi que les différents mouvements qu'il peut réaliser.

Conclusion

Conclusion

Les bras robotiques sont devenus de plus en plus indispensables dans l'industrie, car ils peuvent être utilisés à divers niveaux de la chaîne de production. Ces équipements sont largement employés pour effectuer des tâches répétitives avec une grande précision, ce qui se traduit par une augmentation de la production.

Notre objectif principal était de concevoir et de réaliser un bras robotique capable de saisir des objets et de les déplacer d'un point à un autre de son environnement de travail sans changer leur position. Pour ce faire, nous avons suivi deux étapes principales : la conception du bras, où nous avons sélectionné les composants appropriés et étudié leur structure, et l'assemblage du bras, qui nous a permis de tester différentes rotations et modes de contrôle.

Nous avons exploré l'impact de diverses combinaisons entre le bras et la programmation de deux cartes de contrôle (Arduino Uno et ESP32) tout en présentant les résultats de ces expériences. Dans le cadre de ce projet, nous avons réussi à automatiser les mouvements du bras, en enregistrant les coordonnées pour un contrôle initial puis en le laissant fonctionner de manière autonome.

Le contrôle sans fil Wi-Fi des bras robotiques programmés avec l'ESP32 offre des solutions innovantes pour les situations où l'intervention humaine directe est limitée ou risquée, ouvrant la voie à des applications dans tous les domaines, du domaine médical à la recherche scientifique.

Il existe de nombreuses possibilités d'amélioration du robot, car une fois le robot opérationnel, nous pouvons modifier son algorithme de fonctionnement. Dans une perspective future, nous proposons :

Utiliser des applications mobiles pour contrôler le robot à distance et recevoir des informations en temps réel.

Changer l'unité centrale du robot par une carte Raspberry Pi, ou une carte de développement plus performante que l'Arduino Uno, avec une caméra pour le tri des objets.

Commander le robot avec une reconnaissance vocale.

En résumé, notre mémoire sur la conception et la réalisation d'un bras robotique à 4 degrés de liberté met en évidence l'importance de ces technologies dans l'industrie et fournit des informations précieuses sur les processus de conception et de mise en œuvre de tels équipements.

Bibliographies

- [1] Laumond, Jean-Paul. "La robotique." Annales des Mines-Réalités industrielles. No. 4. Cairn/Softwin, 2016.
- [2] <https://www.hubertprocess.com/nos-solutions/robotique-industrielle/>
- [3] [robot du tchèque robot a travail forcé mot créé en 1920 par K Capek - LAROUSSE](#)
- [4] Mr : L. HADJ BEKLAOUZ « Contrôle à distance d'un bras robot à 4 DDL à l'aide d'un ESP32 » mémoire de master en électronique, Université de Mostaganem, 2022/2023
- [5] <https://educabot.fr/wp-content/uploads/2020/09/WEB-frise.jpg>
- [6] ASIMOV, Isaac. Les robots (I, robot), 1967
- [7] <https://www.generationrobots.com/blog/wp-content/uploads/2022/12/robot-mobile-scoutsan-scout-20-ugv-bras-robotique-a0509.jpg>
- [8] https://static.turbosquid.com/Preview/2018/11/07__14_13_15/001.jpg8F362404-A8F5-442B-B741-2086CEAE97BDZoom.jpg
- [9] Mr : Z. AOUES et B. KARI « Commande dynamique d'un robot manipulateur de deux degrés de liberté » mémoire de master Génie Electrique Université Djilali Bounaama - Khemis Miliana, 2015/2016.
- [10] <https://bcnvision.es/blog-vision-artificial/robotique-vision-industrielle-ia/>
- [11] <https://www.robots-et-compagnie.com/wp-content/uploads/2016/01/robots-industriels.jpg>
- [12] <https://www.opex360.com/2014/05/13/les-nations-unies-se-penchent-sur-le-cas-des-robots-soldats/>, par LAURENT LAGNEAU , 13 MAI 2014
- [13] https://en.wikipedia.org/wiki/Robot-assisted_surgery
- [14] <https://intelligence-artificielle.com/wp-content/uploads/2017/05/chirurgie-robot-1-1152x765.jpg>
- [15] <https://harmonicdrive.de/fr/glossaire/la-robotique>
- [16] <https://www.robotnettoyeur.fr/wp-content/uploads/2020/10/xxlaveur-sol-fond.jpg>
- [17] https://elearning.univ-eloued.dz/pluginfile.php/40193/mod_resource/content/3/Chapitre-1%20Description-des-robots.pdf

- [18] <https://www.astuces-pratiques.fr/electronique/le-moteur-a-courant-continu-principe>, 23 juillet 2015
- [19] https://www.astuces-pratiques.fr/imagesarticles/24/1_le-moteur-a-courant-continu-principe.jpg
- [20] <https://i.ytimg.com/vi/CJUYKErCY5E/maxresdefault.jpg>
- [21] <https://www.omega.fr/prodinfo/Moteur-pas-a-pas.html>
- [22] https://shop.mchobby.be/1977-thickbox_default/moteur-pas-a-pas-12v-200-pas.jpg
- [23] <https://www.technologuepro.com/cours-capteurs-actionneurs-instrumentation-industrielle/ch23-les-actionneurs-pneumatiques.pdf>
- [24]
https://lefrancoisjj.fr/BTS_ET/Schematisation_pneumatique/Activite_schematisation_pneumatique_he lpndoc/lib/Images_010.png
- [25]
https://lefrancoisjj.fr/BTS_ET/Schematisation_pneumatique/Activite_schematisation_pneumatique_he lpndoc/lib/Images_011.png
- [26] <https://www.ummtto.dz/fgei/wp-content/uploads/2020/04/Chapitre-4-V%C3%A9rins.pdf>
- [27]
https://www.unilim.fr/pages_perso/thierry.cortier/Hydraulique_cours/res/8_14_verin_amortissement.png
- [28] <https://fr.wikipedia.org/wiki/Capteur>
- [29] <https://www.evsint.com/fr/types-of-sensors-in-robotics/>
- [30] Mr M. Laribi, et M. Limani. « Commande gestuelle d'un bras manipulateur à 4 degrés de liberté » Diss. Mémoire de master en Commande des systèmes, Université Mouloud Mammeri, 2015.
- [31] Mr : H. TALEB et Z. SIFODIL « Modélisation et commande d'un robot manipulateur à 3 degrés de liberté de type SCARA » mémoire de master en électromécanique et système de commande, Université Aki Mohand Oulhadj Bouira, 2016.
- [32] <https://www.hubertprocess.com/avantages-inconvenients-robots-industriels/>

- [33] Mr S. Khachouche « Conception et Réalisation d'un Bras Manipulateur De Type PUMA quatre Degré de Liberté » mémoire de master en Mention Électronique, Université SAAD DAHLAB de BLIDA, 2017/2018.
- [34] Mr : Z. HAMDACHE et F. TOUIL « Réalisation et commande d'un bras robotique à quatre degrés de liberté » master en électromécanique, université Akli Mohand Oulhadje-Bouira, 2018/2019.
- [35] Mr : Cédric et MESTRE, Daniel « Conception d'un joystick à 4 degrés de liberté pour la conduite de chaise roulante ».
- [36] <https://uelectronics.com/wp-content/uploads/2017/06/ky-023-v8.jpg>
- [37] B. Cotteceau « Carte ARDUINO UNO Microcontrôleur ATmega328 » univ-angers, 2019/2020.
- [38] <https://www.arduino-france.com/wp-content/uploads/2019/02/arduino-uno.jpg>
- [39] https://www.framboiseetcompagnie.fr/wp-content/uploads/2021/02/img_6036bb37279ca.png
- [40] DE LIMA TELES, Diego « Modélisation semi-analytique 2D non linéaire et optimisation d'un moteur électrique : application à un servomoteur pour la robotique » Thèse de doctorat, Université Grenoble Alpes, 2020.
- [41] <https://arduino-france.site/servo-arduino/>
- [42] <https://www.railwayscenics.com/images/electrical/9g-full-servo-motor-close.jpg>
- [43] <https://cdn.shopify.com/s/files/1/0609/6011/2892/files/doc-sg90-servo-motor-internal-components-overview.jpg?width=538>
- [44] <https://diydomo.com/wp-content/uploads/2017/08/1912-Alimentation-12V-2A-BananaPi-BPI-R2.jpg>
- [45] Mme : K. Oukil, et A. Zerbout « Contrôle d'accès à base d'empreinte digitale » mémoire en mathématique et informatique, Université Mouloud Mammeri, 2019.
- [46] https://2btrading.tn/8233-large_default/carte-de-developpement-wifi-esp32-esp-wroom-32-nodemcup2102.jpg
- [47] <https://image.slidesharecdn.com/fonctionnalitsdelacarteesp32-220720104022-242b8515/85/fonctionnalits-de-la-carte-esp32pdf-1-320.jpg?cb=1670783424>
- [48] <https://cdn.shopify.com/s/files/1/0609/6011/2892/files/doc-esp32-wrover-board-simplified-pinout.png>
- [49] DJEBIRI, Amir et BAKHALED, Hadjer « Etude et réalisation d'un compteur d'énergie intelligent » mémoire en Electrotechnique Industrielle, univ-ouargla, 2020.
- [50] Mr. H. KHOUIDMI « TP : Systèmes Embarqués & Systèmes Temps Réel », Université Hassiba Benbouali de Chlef, 2017/2018.

- [51] https://zestedesavoir.com/media/galleries/954/026d3e88-8282-48b9-8355-062de4081471.png.960x960_q85.jpg
- [52] <https://robdomo.com/wp-content/uploads/2022/02/Afficher-Moniteur-serie-regler-communication-9600-baud-1.jpg>
- [53] <https://www.researchgate.net/profile/Khouidmi-Houari/publication/336903004/figure/fig3/AS:819808669020161@1572469026009/La-televersement-du-programme-vers-la-carte-Arduino.ppm>
- [54] https://developer.mozilla.org/fr/docs/learn/html/introduction_to_html/getting_started
- [55] https://www.udacity.com/blog/wp-content/uploads/2020/06/HTML_Blog-scaled.jpeg
- [56] LE SERVEUR, W. E. B. LE SERVEUR, 2009.
- [57] <https://www.hostinger.fr/tutoriels/serveur-web>
- [58] <https://www.hostinger.fr/tutoriels/wp-content/uploads/sites/9/2023/03/comment-fonctionne-un-serveur-web.jpg>

Résumé

La robotique combine plusieurs domaines techniques, tels que la mécanique, l'électronique, l'automatique et l'informatique, pour créer des dispositifs polyvalents et programmables.

Les robots sont conçus pour accomplir des mouvements précis et déplacer des objets, des pièces, des outils ou des dispositifs spécialisés grâce à des mouvements programmés.

Nous allons vous montrer notre conception de bras robotique, en détaillant les différentes étapes de sa construction et en expliquant les termes clés tels que le microcontrôleur, le servomoteur, la base, l'épaule, le coude et la pince.

Pour créer un bras robotique et sa carte de commande, il est essentiel de concevoir chaque pièce avec précision pour obtenir un résultat fiable. Les pièces du bras ont été conçues pour maintenir un bon équilibre et assurer le bon fonctionnement ainsi que la maîtrise des mouvements de notre système robotique.

Mots clés : Logiciel Arduino, bras robot, la robotique, dispositifs mécaniques, Microcontrôleur, Servomoteur, base, épaule, coude et pince.

Abstract

Robotics combines several technical fields, such as mechanics, electronics, automation, and computer science, to create versatile and programmable devices. Robots are designed to perform precise movements and manipulate objects, parts, tools, or specialized devices through programmed actions.

We will present our robotic arm design, detailing the various stages of its construction and explaining key terms such as microcontroller, servomotor, base, shoulder, elbow, and gripper. To create a robotic arm and its control board, it is essential to design each component with precision to achieve a reliable outcome. The arm components have been engineered to maintain proper balance and ensure the effective functioning and control of our robotic system's movements.

Keywords : Arduino software, robot arm, robotics, mechanical devices, microcontroller, servo motor, base, shoulder, elbow, and gripper.