

People's Democratic Republic of Algeria
Ministry of Higher Education and Scientific Research
Mohamed Boudiaf University of M'sila
Faculty of Technology



الجمهورية الجزائرية الديمقراطية الشعبية
وزارة التعليم العالي والبحث العلمي
جامعة المسيلة
كلية التكنولوجيا

Département de Génie Mécanique

MEMOIRE DE FIN D'ETUDE

En vue de l'obtention du diplôme de :

MASTER

En Génie Mécanique

Option : Constructions mécaniques

Présenté par :

BOUZID Abdelbaset & LARBI Allouani

Thème

ETUDE D'UN ROBOT SERIEL

Devant le jury composé de :

NOM et Prénom	Grade	Qualité
BOUDILMI Aissa	MCA	Président
MAKRI Hocine	MCA	Encadreur
SLAMANI Mohamed	Professeur	Encadreur
ROUABHI Youcef	MCA	Examineur

Année Universitaire : 2020 / 2021

N° d'ordre : GM/...../2021

Remerciements

Nous remercions en premier lieu le dieu le tout puissant de nous avoir guidé et donné la force pour achever ce travail.

Nos remerciements vont en second lieu à nos chers parents, qui ont toujours été là pour nous. Nous remercions tous les ingénieurs pour leurs encouragements.

Nous tenons à exprimer notre reconnaissance à notre directeur de mémoire, le docteur MAKRI HOCINE. Nous le remercions de nous avoir aidés, dirigé, et conseillé.

Nous adressons nos sincères remerciements à tous les professeurs, intervenants et toutes les personnes qui par leurs paroles, leurs écrits, leurs conseils et leurs critiques ont guidé nos réflexions et ont accepté de nous rencontrer et de répondre à nos questions durant nos recherches.

Enfin, nous remercions nos amis qui ont toujours été là pour nous. Leur soutien inconditionnel ont été d'une grande aide.

À tous ces intervenants un grand merci.

Dédicaces

Tout d'abord, nous tenons à remercier DIEU

De m'avoir donné la force et le courage de
mener à bien ce modeste travail. Nous tenons à
dédier cet humble travail à : A ma tendre mère
et mon très cher père.

ملخص

اليوم، تتحرك التكنولوجيا في نفس الاتجاه، بالتوازي مع الزيادة السريعة في احتياجات الإنسان. العمل المنجز لتلبية هذه الاحتياجات يجعل الحياة اليومية أسهل، وتركز هذه الدراسات على دراسات الأذرع الروبوتية. تعمل أذرع الروبوت مع مستخدم خارجي أو بتنفيذ أوامر محددة مسبقاً. في الوقت الحاضر، أكثر مجالات أذرع الروبوت تطوراً في جميع المجالات هي الصناعة والطب. تم تصميم الذراع الروبوتية وثبتها في المشروع ولديه القدرة على التحرك في اتجاهات 6 محاور باستخدام 6 محركات متدرجة ويتم ذلك من خلال تطبيق الهاتف والاتصال بوحدة البلوتوث. عند القيام بذلك، يتم توفير التحكم في الروبوت عن طريق الاتصال بتطبيق أردوينو ميغا

Résumé

Aujourd'hui, la technologie évolue dans le même sens, en phase avec l'augmentation rapide des besoins humains. Le travail effectué pour répondre à ces besoins facilite la vie au quotidien, et ces études se concentrent sur des études de bras robotiques. Les bras de robot fonctionnent avec un utilisateur extérieur ou en exécutant des commandes prédéterminées. De nos jours, le domaine le plus développé des bras de robot dans tous les domaines est le secteur de l'industrie et de la médecine. Conçu et réalisé dans le projet, le bras du robot a la capacité de se déplacer dans 6 directions d'axe avec 6 moteurs pas à pas. Ce faisant, le contrôle du robot est fourni en se connectant à l'application Android via un module Bluetooth connecté au microcontrôleur Arduino Mega.

ABSTRACT

Today, technology is developing in the same direction in line with rapidly increasing human needs. The work done to meet these needs makes life easier every day, and these studies are concentrated in robotic arm studies.

Robot arms work with an outside user or by performing predetermined commands. Nowadays, the most developed field of robot arms in every field is the industry and medicine sector. Designed and realized in the project, the robot arm has the ability to move in 6 axis directions with 6 stepper motors.

While doing this, robot control is provided by connecting to the android application via Bluetooth module connected to Arduino Mega microcontroller.

TABLE DU MATIERE

INTRODUCTION GENERALE	16
CHAPITRE I : LES ROBOTS INDUSTRIELS	17
I.1. Historique du robot industriel	18
I.2. Définition d'un robot industriel.....	18
I.3. Les différents types de bras manipulateur	19
I.4. Axes de la robotique	20
I.5. Chaîne cinématique.....	21
I.5.1 Structure ouverte simple	21
I.5.2 Structure arborescente ou poly articulaire	21
I.5.3 Structure fermée	21
I.5.4 Robots parallèles	22
I.6. Domaines d'application des robots manipulateurs	23
I.6.1 Domaines industriels	23
I.6.2 Domaine médical	24
I.6.3 Domaine militaire	24
I.6.4 Autres domaines	24
I.7. Architecture des robots manipulateurs	25
I.7.1 La base	25
I.7.2 Le porteur	25
I.7.2.1. Le segment	25
I.7.2.2 Articulation	25
I.7.2.3 L'actionneur	26
I.7.2.4. Effecteur	26
I.8 Classification fonctionnelle des robots	27
I.8.1. Manipulateur à commande manuelle	27
I.8.2. Manipulateur automatique	27
I.9. Classification géométrique ou morphologie.....	28
I.9.1. La structure cartésienne (PPP)	28

I.9.2. La structure cylindrique (RPP) ou (PRP)	28
I.9.3. La structure sphérique ou polaire	25
I.9.4. La structure dite SCARA	25
I.9.5. La structure 3R (anthropomorphe)	25
I.9.6. La structure parallèle	30
I.10. Système de coordonnées en robotique	30
Conclusion.....	31
CHAPITRE II : MODELISATION DES ROBOTS	32
II.1. Modélisation d'un robot	33
II.2. Modélisation cinématique	34
II.2.1. Modélisation cinématique directe (MCD)	34
II.2.2. Modélisation cinématique Inverse	37
II.3. Robot serial et son étalonnage	39
II.3.1. Niveaux d'étalonnage	39
II.3.2. Types d'étalonnage	40
II.3.3. Classification des méthodes d'étalonnage	40
Conclusion.....	40
CHAPITRE III : CONCEPTION D'UN ROBOTS SERIEL A 6DDL.	
III.1. Justification du choix du robot	43
III.2. Présentation de la chaine Cinématique du robot	43
III.3. Présentation des croquis de conception	44
III.4. Les différentes parties du robot (groupes d'assemblages)	48
III.4.1. L'assemblage global du robot:	49
III.4.1.1 conception CAO	49
III.4.1.2 Conception réelle après réalisation	50
III.4.2. Les différents groupes d'assemblage du robot	51

III.4.2.1. Dimensionnement et Assemblage du groupe1 du robot	52
III.4.2.1.1. Conception et dimensionnement de la pièce n°1	55
III.4.2.1.2. Conception et dimensionnement de la pièce n°2	58
III.4.2.1.3. Conception et dimensionnement de la pièce n°5	61
III.4.2.1.4. Conception et dimensionnement de la pièce n°7	64
III.4.2.1.5. Conception et dimensionnement de la pièce n°9	67
III.4.2.1.6. Conception et dimensionnement de la pièce n°16	70
III.4.2.1.7. Conception et dimensionnement de la pièce n°17	73
III.4.2.1.8. Conception et dimensionnement de la pièce n°18.....	76
III.4.2.2. Le dimensionnement et l'assemblage du groupe2 du robot.....	79
III.4.2.3. Le dimensionnement et l'assemblage du groupe3 du robot	83
III.4.2.4. Le dimensionnement et l'assemblage du groupe 4 du robot.....	86
III.4.3. : Spécifications techniques des pièces préfabriquées à acheter	88
III.5. Le prototype final du robot après réalisation (UNIVERSEL 100DZ21)	90

CHAPITRE IV : COMMANDE DU ROBOTS SOUS ARDUINO 93

IV.1. La Commande du Robot	94
IV.1.1. Matériel nécessaire pour la commande	94
IV.1.2. Description et caractéristiques techniques.....	95
IV.1.2.1. La carte platine Arduino Mega	95
IV.1.2.2. Le contrôle de moteur pas-à-pas DRV8825	100

IV.1.2.3. La platine d'expérimentation (breadboard).....	101
IV.1.2.4. Câbles de liaison (jumpers)	102
IV.1.2.5. Les LEDS	102
IV.1.2.6. La résistance	103
IV.1.2.7. Condensateurs	103
IV.1.2.8. Bouton poussoir	104
IV.1.2.9. Potentiomètre	105
IV.1.2.10. Moteur pas à pas.....	106
IV.1.2.11. Module Bluetooth HC-05	108
IV.1.2.12. Boite d'Alimentation Power supply	108
IV.2. Le Câblage	109
IV.2.1. Le Câblage Avec logiciel Fritzing	109
IV.2.1.1. L'Application Fritzing.....	109
IV.2.1.2. Exécution du Câblage avec ARDUINO	109
IV.2.2 Le Câblage réel de tout groupe électronique.....	115
IV.2.2.1. Le Câblage réel des éléments électronique extérieurs.....	115
IV.2.2.2. Le Câblage réel des éléments électronique intérieurs	116
IV.2.2.3. Le Câblage réel des éléments électroniques avec les drivers DIV268N	117
IV.2.2.4. Les éléments électroniques réel	118
IV.3 .Présentation du logiciel Arduino	119
IV.3.1. Définition du module Arduino.....	119
IV.3.2. Présentation de l'interface d'Arduino	120
IV.3.3. La programmation en langage d'Arduino	123
IV.4. Programme utilisée pour pilotage du robot	124
Conclusion	126
CHAPITRE V : COMMANDE AVEC UNE APPLICATION ANDROID.....	130
Introduction	131
V.1 : L'interface d'application remotexy	131
V.2 : Création d'application	131
V.3 : Les zones de la fenêtre de l'éditeur	132

V.4 : Orientation de l'interface graphique de l'écran	133
V.5 : Liste de configuration	134
V.6 : Liste Module interface	137
V.7 : Programme de pilotage par Bluetooth	139
Conclusion.....	143
Conclusion générale.....	144
La bibliographie	145

LISTE DE FIGURE :

CHAPITRE I :

Figure. I.1: Karel Capek à gauche et L'ingénieur américain Joseph F. Engel berger qui a été à l'origine du premier robot industriel Unimate à droite	17
Figure. I.2: Structure ouverte simple	20
Figure. I.3: Structure arborescence.....	20
Figure. I.4: Structure fermée	21
Figure. I.5: Structure fermée simple.....	21
Figure. I.6: Robot parallèle.....	22
Figure. I.7: Robot de soudage	23
Figure. I.8: Robot de décontamination d'une pompe de circulation dans le nucléaire	23
Figure. I.9: Robots d'exploration spatiale à gauche et robot sou marin autonome à droite	23
Figure. I.10: Architecture d'un Robot manipulateur.....	24
Figure. I.11: Présentation d'une articulation rotoïde	25
Figure. I.12: Présentation d'une articulation prismatique	25
Figure. I.13: Représentation des différents types d'actionneurs.....	25
Figure. I.14: Représentation d'une série d'effecteurs	26
Figure. I.15: Manipulateur à commande manuelle à gauche et a commandé automatique (kuka-robot de soudage à l'arc)	26
Figure. I.16: (a) Structure cartésienne, (b) Robot cartésien (SEPRO).....	27
Figure. I.17: (a) Structure cylindrique, (b) Robot cylindrique (Plate Crane)	28
Figure. I.18: (a) Structure sphérique, (b) Robot sphérique (FANUC L1000+A).....	29

Figure. I.19: (a) Structure SCARA, (b) Robot type SCARA (ADTECH)	29
Figure. I.20: (a) Structure anthropomorphe, (b) Robot anthropomorphe (Stäubli)	29
Figure. I.21: Structure parallèle type HEXA à gauche et Robot type DELTA à droite	30
Figure. I.22: Changements de coordonnées en robotique	31
CHAPITRE II :	
Figure. II.1: Structure d'un robot sériel à 3ddl	33
Figure. II.2: Paramètres de Denavit-Hartenberg	34
Figure. II.3: les paramètres D-H	35
CHAPITRE III :	
Figure III.1 : Placement des repères des articulations de la chaîne cinématique.....	42
Figure III.2 : Vue globale de la conception du robot en CAO	48
Figure III.3 : Vue globale de la conception réelle du robot après la réalisation	49
Figure III.4 : L'assemblage des groupes du robot	50
Figure III.5 : Vue 3D de la conception CAO du groupe1 (la base du robot)	51
Figure III.6: Image de la conception réelle du groupe 1	54
Figure III.7 : Vue 3D de la conception CAO de la pièce N°1	54
Figure III.8 : Image de la conception réelle de la pièce N°1	56
Figure III.9 : Vue 3D de la conception CAO de la pièce N°2	57
Figure III.10: Image de la conception réelle de la pièce N°2	59
Figure III.11: Vue 3D de la conception CAO de la pièce N°5.....	60
Figure III.12: Image de la conception réelle de la pièce N°5	62
Figure III.13: Vue 3D de la conception CAO de la pièce N°7	63
Figure III.14: Image de la conception réelle de la pièce N°5	65
Figure III.15: Vue 3D de la conception CAO de la pièce N°9.....	66
Figure III.16: Image de la conception réelle de la pièce N°9	68
Figure III.17: Vue 3D de la conception CAO de la pièce N°16	69

Figure III.18: Image de la conception réelle de la pièce N°16.....	71
Figure III.19: Vue 3D de la conception CAO de la pièce N°17	72
Figure III.20: Image de la conception réelle de la pièce N°17 semi-fini	74
Figure III.21: Vue 3D de la conception CAO de la pièce N°5.....	75
Figure III.22: Image de la conception réelle de la pièce N°18.....	77
Figure III.23: Vue 3D de la conception CAO du groupe2.....	78
Figure III.24: Image de la conception réelle du groupe 2	81
Figure III.25: Vue 3D de la conception CAO du groupe3.....	82
Figure III.26: Image de la conception réelle du groupe 3	84
Figure III.27: Vue 3D de la conception CAO du groupe4.....	85
Figure III.28: Image de la conception réelle du groupe 4.....	87
Figure III.29 Le prototype final UNIVERSEL 100DZ21	90
CHAPITRE IV :	
Figure. IV.1 : Carte d'Arduino MEGA	92
Figure. IV.2 : Platine d'expérimentation (breadboard)	92
Figure. IV.3: Eléments électroniques utilisés	93
Figure. IV.4 : Les éléments d'alimentation et de pilotage	93
Figure. IV.5 : Description de la carte Arduino MEGA 2560	94
Figure. IV.6 : Microcontrôleur ATmega2560	98
Figure. IV.7: Le contrôle de moteur pas-à-pas DRV8825 et son schéma de montage sous Arduino	98
Figure. IV.8: Une platine d'expérimentation (breadboard).....	99
Figure. IV.9 : Le branchement sur un breadboard.....	99
Figure. IV.10: Câbles de liaison (jumpers)	100
Figure. IV.11: le schéma d'une LED	100
Figure. IV.12: la résistance électrique	101

Figure. IV.13: Condensateur électrique	101
Figure. IV.14: Chronogramme Bouton poussoir marche	102
Figure. IV.15: Schéma électrique et présentation Bouton poussoir marche	102
Figure. IV.16: Chronogramme Bouton poussoir d'arrêt	103
Figure. IV.17: Schéma électrique et présentation Bouton poussoir d'arrêt	103
Figure. IV.18: Symbole électrique du potentiomètre et piste résistive	103
Figure. IV.19: Explication type du potentiomètre	104
Figure. IV.20: Classification du potentiomètre	104
Figure. IV.21: Moteur pas à pas	105
Figure. IV.22: Principe de commande en position d'un moteur à courant continu	105
Figure. IV.23: Principe de commande d'un moteur pas à pas	105
Figure. IV.24: Module Bluetooth HC-05 avec explication d'alimentation	106
Figure. IV.25: Alimentation (power supply).....	108
Figure. IV.26: L'interface de logiciel fritzing	107
Figure. IV.27: Câblage d'un potentiomètre et une LED dans un Arduino	108
Figure. IV.28: Câblage d'une résistance et d'un condensateur	108
Figure. IV.29: Câblage d'un bouton poussoir.....	109
Figure. IV.30: Câblage d'un driver	111
Figure. IV.31: Câblage d'un alimentation et HC05	111
Figure. IV.32 : Câblage réel des éléments électronique extérieurs.....	112
Figure. IV.33 : Câblage réel des éléments électronique intérieurs.....	113
Figure. IV.34 : Câblage réel des éléments électroniques avec les drivers DIV268N.....	114
Figure. IV.35 : L'élément électronique réel	115
Figure. IV.36: Module d'Arduino	116
Figure. IV.37: Présentation du lancement de l'exécutable du logiciel d'Arduino	117

Figure. IV.38: L'interface du logiciel d'Arduino	117
Figure. IV.39: Présentation des champs relatifs au logiciel d'Arduino	118
Figure. IV.40: Présentation des commandes du menu File	119
Figure. IV.41: Présentation des boutons du menu File	119
Figure. IV.42: La bibliothèque avec définition de direction et step du moteur pas à pas	121
Figure. IV.43: Définition des boutons poussoirs	121
Figure. IV.44: La boucle bool et les boutons (marche, reverse)	121
Figure. IV.45: Fournir une définition d'une l'entrée et donner le voltage Maximin de chaque bouton poussoir.....	122
Figure. IV.46: pilotage moteur 6 à l'aide le bouton poussoir M6	123
Figure. IV.47: pilotage moteur 5 à l'aide le bouton poussoir M5	123
Figure. IV.48: pilotage moteur 4 à l'aide le bouton poussoir M4	124
Figure. IV.49: pilotage moteur 3 à l'aide le bouton poussoir M3	124
Figure. IV.50: pilotage moteur 2 à l'aide le bouton poussoir M2	125
Figure. IV.51: pilotage moteur 1 à l'aide le bouton poussoir M1	125
Figure. IV.52: pilotage robot à l'aide le bouton poussoir Start	126
CHAPITRE.V :	
Figure. IV.1: L'interface d'application remotexy	128
Figure. IV.2: Création d'application remotexy	129
Figure. IV.3: Les zones de la fenêtre de l'éditeur	130
Figure. IV.4: Orientation de l'interface graphique de l'écran	131
Figure. IV.5: Liste de configuration	131
Figure. IV.6: Liste choisie pour ce projet	132
Figure. IV.7: Connexion Bluetooth	132
Figure. IV.8: Module d'Arduino	133

Figure. IV.9: Module Bluetooth HC-05.....	133
Figure. IV.10: Logiciel de programmation	134
Figure. IV.11: Module d'interface et liste de view	134
Figure. IV.12: Liste des contrôles, d'indication et décoration	135
Figure. IV.13: L'interface de l'architecture globale	135

LISTE DES TABLEAUX

Tableau III.1 Spécifications techniques des pièces préfabriquées à acheter	88
Tableau. IV.1: Constitution de la carte Arduino Mega 2560	95
Tableau. IV.2: Caractéristiques du moteur pas à pas.....	111
Tableau. IV.3: Caractéristiques du DRV8825	110

LISTE DU FORMULES

(II.1) les variations élémentaires des coordonnées opérationnelles en fonction des variations élémentaires des coordonnées articulaire.....	33
(II.2) la matrice homogène pour définir les transformations nécessaires pour relier les référentiels successifs entre eux	34
(II.3) la matrice homogène Pour les axes parallèles consécutifs	34
(II.4) la matrice de rotation autour de l'axe x par rapport au système de coordonnées de l'articulation i-1	35
(II.5) la matrice de translation au long de l'axe x par rapport au nouveau système de coordonnées obtenu lors de la rotation précédente	35
(II.6) la matrice de rotation autour de l'axe z par rapport au nouveau système de coordonnées obtenu lors des transformations précédentes	36
(II.7) la matrice de translation au long de l'axe z par rapport au nouveau système de coordonnées obtenu lors des transformations précédentes	36
(II.8) la valeur de rotation ou translation selon un axe à appliquer	36
(II.9), (II.10), (II.11), (II.12) les matrices homogènes impliquent une quantité énorme de sinus et cosinus rendant la résolution encore plus complexe	37
(II.13) La technique de la substitution de variables	38

INTRODUCTION GENERALE

L'utilisation croissante des robots dans les domaines industriel, médical et militaire doit répondre à une demande croissante de précision. Pour répondre à cette demande, il est nécessaire de connaître le modèle géométrique exact des structures mécaniques sérielles ou parallèles utilisées. Quels que soient les modèles utilisés pour représenter la géométrie des robots, leur précision dépend de la précision des paramètres appliqués dans les ordinateurs. Les erreurs géométriques de la position réactive de l'instrument sont principalement dues aux erreurs de conversion cinématique causées par les écarts entre le comportement du modèle d'ingénierie appliqué dans le contrôle du robot et le comportement géométrique réel de la structure du robot. En fait, la structure du robot est fabriquée et façonnée, et donc le comportement technique du robot est affecté par des défauts dans la formation et l'assemblage des éléments. Ces défauts, limités par la tolérance appliquée, génèrent une différence entre les dimensions nominales du robot et les dimensions réelles de la structure assemblée. Il est alors nécessaire de définir et de contrôler les dimensions qui affectent la stabilisation de l'effecteur soit par une tolérance conditionnée, soit en développant un modèle géométrique adapté lié à la manière dont ces paramètres géométriques sont déterminés.

Notre tâche dans le traitement, d'étude et la réalisation des travaux qui nous sont confiés s'articule autour de deux étapes fondamentales, dont l'une vise principalement à fabriquer et à mettre en œuvre un robot à six degrés de liberté, et la seconde est dans laquelle l'étude théorique présentée dans la mémoire ; qu'elle a été divisé en cinq chapitres de base reliés entre eux ; Dans le premier chapitre, nous avons évoqué dans notre étude un aperçu de l'histoire et du comment de l'émergence de la technologie robotique, ses types, et son importance dans plusieurs domaines, notamment le domaine industriel.

Dans le deuxième chapitre, nous avons parlé des modèles de robots, les plus importants, et leurs avantages en termes de degrés de liberté, de dynamique et de cinétique, et la portée de leur utilisation dans l'industrie. Dans le troisième et important chapitre et sa relation avec le titre de la thèse, il a été abordé pour étudier la théorie et les étapes de la conception, nous avons commencé par des dessins préparatoires et des croquis, et la théorie du CAO conçoit à travers des programmes spéciaux dans ce domaine, tels que *CATIA* ou *SOLIDWORKS*, puis la réalisation de dessins définitions, la détermination des dimensions, les tolérances et les dimensions fonctionnelles de chaque pièce, qu'il est se présenter par des images de dessins virtuels et un autre après leur fabrication. Nous avons également discuté dans les quatrième et cinquième chapitres, qui ne sont pas moins importants que le chapitre précédent. Nous avons mentionné les médias et programmes les plus importants disponibles pour contrôler les mouvements d'un robot. Pour compléter l'unité de contrôle, la méthode de contrôle Arduino a été sélectionnée et puis connecté aux composants électroniques, suivi d'une programmation et d'un contrôle via un ordinateur ou une application Android "RemoteXY" qui envoie les données nécessaires à l'Arduino via Bluetooth, qui à son tour envoie des signaux électriques pour déplacer les organes du robot.

Nous avons travaillé dur pour livrer un travail parfait qui rassemble autant d'informations que possible pour mener à bien le projet selon les mécanismes et les conditions à notre disposition.

CHAPITRE I : LES ROBOTS INDUSTRIELS.

I.1. Historique du robot industriel:

- En 1920: Apparition du mot robot. L'origine du mot robot provient de la langue tchèque dans laquelle sont ancêtre "robota" signifie travail forcé. Il a été introduit, en 1920, par l'écrivain tchèque Karel Capek dans la pièce de théâtre Rossum's Universal Robots.
- En 1961: Unimation, le 1^{er} robot industriel Descendant direct des télémanipulateurs développés pour les besoins du nucléaire. Il est vendu à partir de 1961 par la société américaine Unimation (devenu Stäubli Unimation), créée par George Devol et Joseph Engelberger. Il est utilisé pour la première fois sur les lignes d'assemblage de General Motors. Ce robot, grâce à son bras articulé de 1,5 tonne, était capable de manipuler des pièces de fonderie pesant 150 kg.
- En 1972: 1^{ère} chaîne de production robotisée Nissan ouvre la première chaîne de production complètement robotisée, Selon une étude de l'IFR, 2142 millions de robots ont été fabriqués entre les années 60 et la fin 2010, Les analystes estiment qu'aujourd'hui, de 1 à 1,3 million de robots travaillent pour nous dans les usines dans le monde.

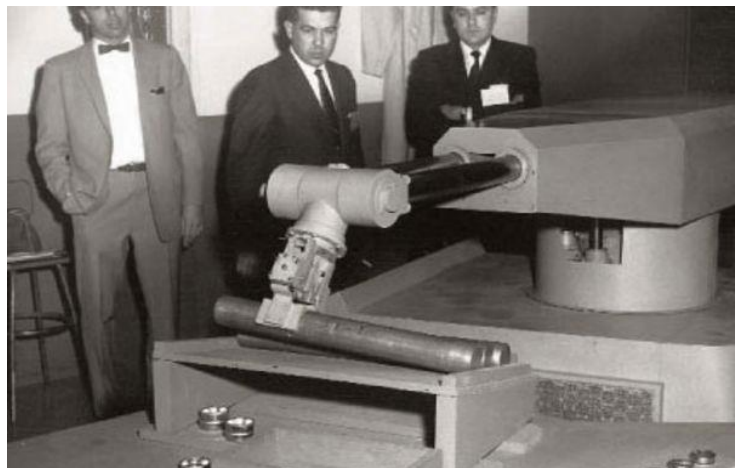


Figure. I.1: Karel Capek à gauche et L'ingénieur américain Joseph F. Engelberger qui a été à l'origine du premier robot industriel Unimate à droite. [1]

I.2. Définition d'un robot industriel :

En grec ancien, automate signifie : « machine imitant les mouvements et les fonctions d'un corps ». Le terme « robot » quant à lui vient du tchèque « robota » : travail forcé, pour désigner

des ouvriers artificiels. Il apparaît en 1924 dans la pièce « les robots universels de Rossum » de Karel Capek, auteur tchèque de nouvelles, de romans, de drames, satires des travers humains.

Le dictionnaire nous donne plusieurs définitions du mot robot : Dans les œuvres de science-fiction, machine à l'aspect humain, capable de se mouvoir, d'exécuter des opérations, de parler. Appareil automatique capable de manipuler des objets ou d'exécuter des opérations selon un programme fixe, modifiable ou adaptable. Personne qui agit de façon automatique.

De nombreux ouvrages ont déjà donné des informations sur l'origine de la robotique et du terme robot. Ce mot est tombé dans le sens commun. Pour beaucoup, il s'agit d'une machine anthropoïde qui, par sa forme, tente de ressembler à un être humain et par ses capteurs intégrés à son comportement. Depuis plus d'une décennie ces machines, qui existent aussi sous la forme de jouets donnent une fausse idée sur ce qu'elles sont réellement aujourd'hui, sur leur développement, tout particulièrement dans le domaine industriel où elles ont pris leur essor.

Dans ce premier chapitre, nous allons introduire quelques notions générales relatives aux robots manipulateurs, notamment leur définition, leur classification, leurs structures, leurs applications, ensuite, nous présenterons la fonction perception de l'environnement dans la robotique industrielle et les différents capteurs et moteurs utilisés pour l'accomplissement de cette fonction. [1].

D'une façon plus générale, la norme ISO 8373 [ISO 94b] utilise la définition du robot manipulateur donnée comme suit : « une machine, un mécanisme constitué normalement d'une série de segments qui sont reliés par une articulation assurant une rotation ou une translation relative entre segments, dont le but est de prendre et déplacer des objets (pièces ou outils) avec plusieurs degrés de liberté. Il peut être commandé par un opérateur, une unité de commande électronique ou un système logique (dispositif à cames, relais, câbles, etc.) [2].

Un robot manipulateur, est un mécanisme capable de déplacer et de situer un objet appelé « organe terminal » ou « point outil » dans une partie de l'espace appelée « volume de travail ». Un robot industriel est un manipulateur à plusieurs degrés de liberté contrôlé automatiquement, reprogrammable et multitâche qui peut être fixe ou mobile pour une application en automatisation industrielle.

I.3. Les différents types de bras manipulateur :

Les types de bras manipulateur sont regroupés selon plusieurs critères ou caractéristiques comme, les axes d'utilisation, les classes et catégories, par rapport aux degrés de liberté, aux types d'architectures, et aux différentes structures générées. On peut déjà distinguer en général deux groupes de robots [3] :

- Passif, ou dit aveugle, accomplissant son devoir sans recourir aux informations du monde extérieur, et évolue en boucle ouverte par rapport à son environnement, une fois la phase d'apprentissage terminé, il reproduit toujours les mêmes gestes, dans le même ordre, ce sont des manipulateur dépourvus de tout sens.
- Actif, ou interactif, une fois la phase d'apprentissage termine il s'adapte suivant les paramètres désirés, et les paramètres de l'environnement, dans le quelle il évolue en boucle fermée, Ce sont ces machines que l'on peut appeler robots, c'est le début de l'intelligence artificielle.

I.4. Axes de la robotique :

- Les robots de coopérations : sont sous le contrôle direct de l'homme.
- Les robots de substitution : peux intelligent, opèrent seuls après programmation.
- Les robots d'aide à la conception : conception et fabrication d'autres machines ou robot, assistés par ordinateur
- Les robots à architecture spéciale : conçus à partir d'observations de l'homme et des animaux. [1]

Selon le type on trouve, on distingue les suivants : [1]

- Le manipulateur manuel
- Le robot à séquences de travail fixes
- Le robot à séquences de travail variable, évolutive
- Le robot copieur ou play-back
- Le robot à commande programmable numérique
- Le robot intelligent capable de s'adapter aux changements des conditions de travail et de Les classifications énumérées ci-dessous est donné par les institutions suivantes :

- J.I.R.A (Japan industrial Robot – Industry Association)
- R.I.A (Robot Institute of America)
- A.F.R.I (Association Française de Robots Industriels).

I.5. Chaîne cinématique :

Un robot-manipulateur est constitué par deux sous-ensemble distinct, un (ou plusieurs) Organe terminal et une structure mécanique articulée :

- Un organe terminal est tout dispositif destiné à manipuler des objets ou a-le Transformer, il permet au robot d'interagir avec son environnement, il peut être Multifonctionnel (plusieurs dispositifs avec fonctionnalité différentes), Monofonctionnel interchangeable, ou multi-bras.

- La structure mécanique articulée amènera l'organe terminal dans une position et

Orientation donnée. Son architecture est une chaîne cinématique de corps généralement Rigides, assemblés par des articulations. Les chaînes peuvent être : [4]

I.5.1 Structure ouverte simple :

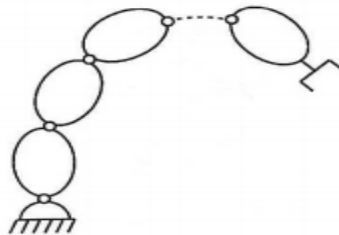


Figure. I.2: Structure ouverte simple [4]

I.5.2 Structure arborescente ou polyarticulaire:

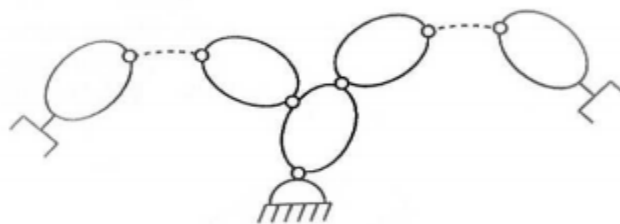


Figure. I.3: Structure arborescente [4]

I.5.3 Structure fermée :

En général, on distingue deux classes de boucles fermées [4]

- Les chaînes cinématiques composées ou complexes (au moins deux liaisons)

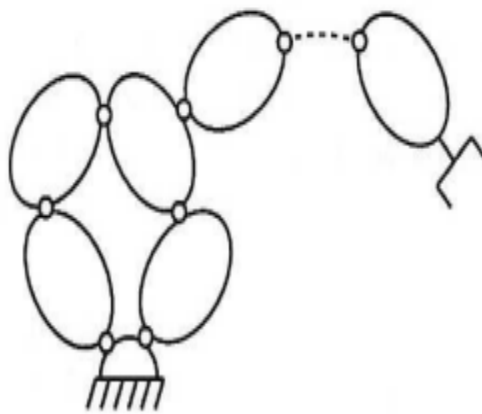


Figure. I.4: Structure fermée [4]

- Les chaînes cinématiques élémentaires ou simples (au plus deux liaisons)

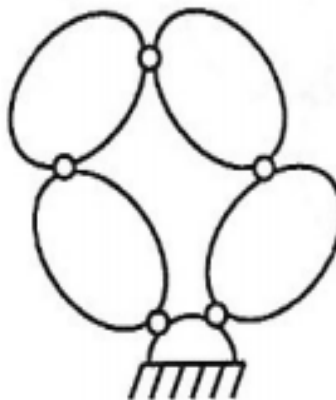


Figure. I.5: Structure fermée simple [4]

I.5.4 Robots parallèles :

Une génération différente de robots, ici l'organe terminal est relié à la base du mécanisme par plusieurs chaînes parallèles, assurant une plus grande rigidité et donc une plus grande précision. [4]

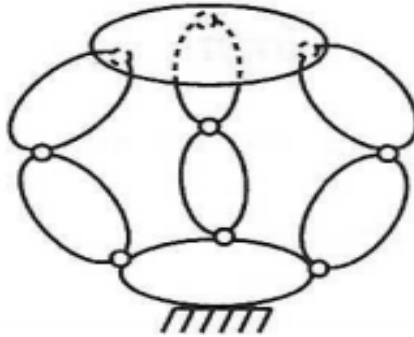


Figure. I.6: Robot parallèle [4]

I.6. Domaines d'application des robots manipulateurs:

La robotique permet de réaliser des tâches répétitives, pénibles, dangereuses et dans des milieux hostiles, avec précision et fiabilité, dans tout environnement, à des cadences élevées et continues, tout en palliant au manque de main-d'œuvre, favorisant la haute productivité, et donnant ainsi à la robotique un fort intérêt humain, technique et économique. Ce qui a permis aux robots de se diversifier dans les domaines d'utilisation sous cités.

I.6.1 Domaines industriels:

- La transformation de pièces : moulage, usinage, perçage, rectification, enlèvement de matière
- La préparation de surface : peinture, enduction, projection thermique/plasma, grenaillage
- L'assemblage : rivetage, vissage, soudage, insertion, collage
- Les manutentions : manipulation de pièces, chargement/déchargement de machine, positionnement
- La distribution

- La palettisation et le conditionnement
- L'inspection et le test
- Le nettoyage



Figure. I.7: Robot de soudage

I.6.2 Domaine médical:

- Opérations
- Diagnostique
- Radiothérapie
- Prothèses

I.6.3 Domaine militaire:

- Désamorçage
- Déminage

I.6.4 Autres domaines:

- Nucléaire : Manipulation des produits radioactifs
- Spatiale : Exploration des planètes



Figure. I.8: Robot de décontamination d'une pompe de circulation dans le nucléaire.

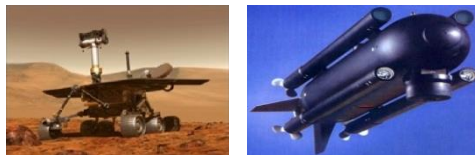


Figure. I.9: Robots d'exploration spatiale à gauche et robot sou marin autonome à droite.

I.7. Architecture des robots manipulateurs:

Un robot manipulateur est composé principalement de la base, des segments, des articulations, des actionneurs et de l'organe terminal. L'architecture générale d'un robot manipulateur est présentée dans Figure I.10 : Ci-bas ;

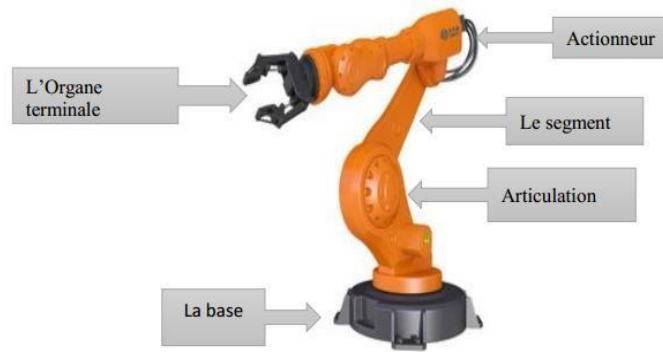


Figure I.10 : Architecture d'un Robot manipulateur

I.7.1 La base : permet la fixation du robot sur le lieu du travail. Ceci est le cas de la quasi-totalité des robots industriels.

I.7.2 Le porteur : Le porteur représente l'essentiel du système mécanique articulé, il a pour rôle d'amener l'organe terminal dans une situation donnée imposée par la tâche (la situation d'un corps peut être définie comme la position et l'orientation d'un repère attaché à ce corps par rapport à un repère de référence). Il est constitué de :

I.7.2.1. Le segment : Corps solides rigides susceptibles d'être en mouvement par rapport à la base du porteur, et les uns par rapport aux autres.

I.7.2.2 Articulation : Une articulation lie deux corps successifs en limitant le nombre de degré de liberté, de l'un par rapport à l'autre.

Soit m le nombre de degré de liberté résultant, encore appelé mobilité de l'articulation. La mobilité d'une articulation est telle que $0 \leq m \leq 6$, lorsque $m = 1$, ce qui est fréquemment le cas en robotique, l'articulation est dite simple : rotoïde (R) ou prismatique (P).

❖ **Articulation rotoïde :** Il s'agit d'une articulation de type pivot, notée R, réduisant le mouvement entre deux corps à une rotation autour d'un axe commun. La situation relative entre les deux corps est donnée par l'angle autour de cet axe.

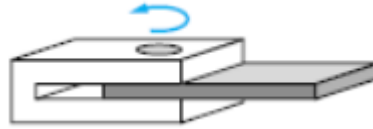


Figure.I.11: Présentation d'une articulation rotoïde. [5]

- ❖ **Articulation prismatique:** Il s'agit d'une articulation de type glissière, notée P, réduisant le mouvement entre deux corps à une translation le long d'un axe commun. La situation relative entre les deux corps est mesurée par la distance le long de cet axe.

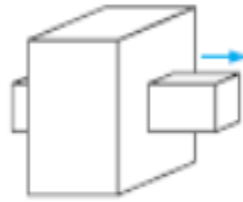


Figure. I.12. Présentation d'une articulation prismatique. [5]

I.7.2.3 L'actionneur: Les actionneurs utilisent fréquemment des moteurs électriques à aimant permanent, à courant continu, à commande par l'induit. On trouve de plus en plus de moteurs à commutation électronique (sans balais), ou, pour de petits robots, des moteurs pas à pas. Pour les robots devant manipuler de très lourdes charges (par exemple, une pelle mécanique), les actionneurs sont le plus souvent hydrauliques, agissant en translation (vérin hydraulique) ou en rotation (moteur hydraulique).



Figure. I.13: Représentation des différents types d'actionneurs.

I.7.2.4. Effecteur: Il s'agit d'une interface permettant au robot d'interagir avec son environnement. Ça regroupe les dispositifs de manipulation des objets (dispositifs de serrage, dispositifs magnétiques, à dépression, ...), ou les dispositifs de transformation (outils, torche de soudage, pistolet de peinture, ...).

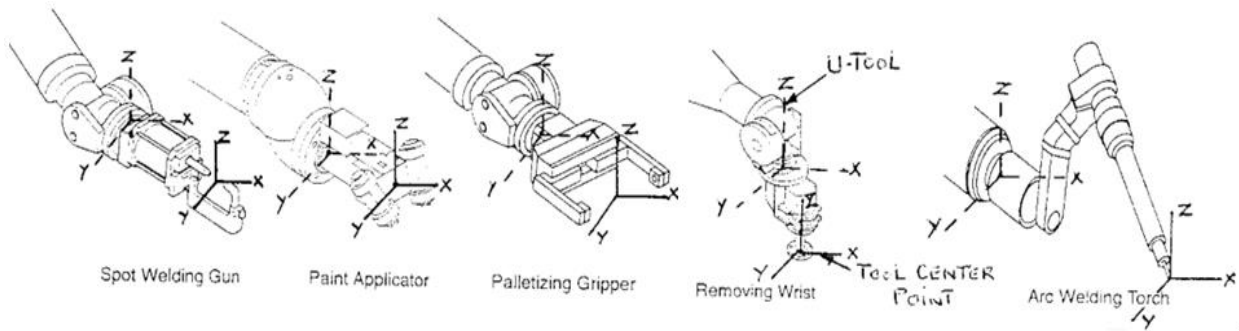


Figure. I.14: Représentation d'une série d'effecteurs.

On peut aussi classer les robots d'un point de vue fonctionnel ou d'après leur structure géométrique :

I.8 Classification fonctionnelle des robots :

I.8.1. Manipulateur à commande manuelle: Le manipulateur à commande manuelle conçu et réalisé pour les manipulations particulièrement difficiles, répond spécifiquement aux exigences liées à la manutention de charges élevées, dans tous les points du volume de travail sans effort et dans des conditions optimales d'ergonomie et de sécurité.

I.8.2 Manipulateur automatique: La commande peut se faire par automate programmable on peut distinguer entre manipulateurs à cycle fixe et manipulateurs à cycle programmable. Figure.I.15 montre un bras manipulateur qui exerce des mouvements de soudure sans l'intervention de l'homme.



Figure.I.15 : Manipulateur à commande manuelle à gauche et a commandé automatique (kuka-robot de soudage à l'arc).

- **Robots programmables:** Ces robots répètent les mouvements qu'on leur a appris ou programmés sans informations sur l'environnement ou la tâche effectuée. On peut aussi faire la distinction entre robots « Playback » qui reproduit la tâche apprise et robots à commande numérique qui peuvent être programmés hors- ligne.
- **Robots intelligents:** On trouve actuellement des robots de seconde génération qui sont capables d'acquérir et d'utiliser certaines informations sur leur environnement (systèmes de vision, détecteurs de proximité, capteurs d'efforts, ...). Les robots de troisième génération sont capables de comprendre un langage oral proche du langage naturel et de se débrouiller de façon autonome dans un environnement complexe grâce à l'utilisation de l'intelligence artificielle.

I.9. Classification géométrique ou morphologie :

On peut aussi classer les robots suivant leur architecture de leur porteur. Ci-dessous, nous soulignerons les caractéristiques les plus importantes pour chaque structure.

I.9.1. La structure cartésienne (PPP) :

- 3 axes $\perp$ 2 à 2, série, PPP, 3 DDL.
- Très bonne précision.
- Lent.

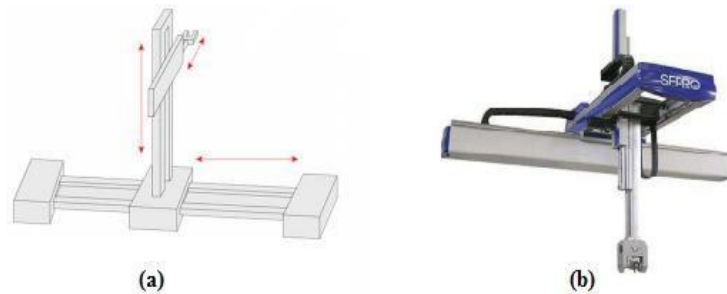


Figure.I.16 : (a) Structure cartésienne, (b) Robot cartésien (SEPRO) [1]

I.9.2. La structure cylindrique (RPP) ou (PRP) :

- 3 axes, série, RPP/ PRP, 3 DDL.
- Espace de travail cylindrique.
- Très rapide.



Figure.I.17 : (a) Structure cylindrique, (b) Robot cylindrique (Plate Crane) [1]

I.9.3. La structure sphérique ou polaire:

- 3 axes, série, RRP, 3 DDL.
- Espace de travail sphérique.
- Grande charge utile.

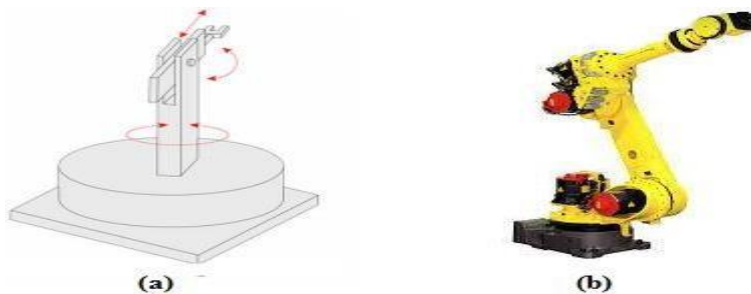


Figure.I.18 : (a) Structure sphérique, (b) Robot sphérique (FANUC L1000+A). [1]

I.9.4. La structure dite SCARA:

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

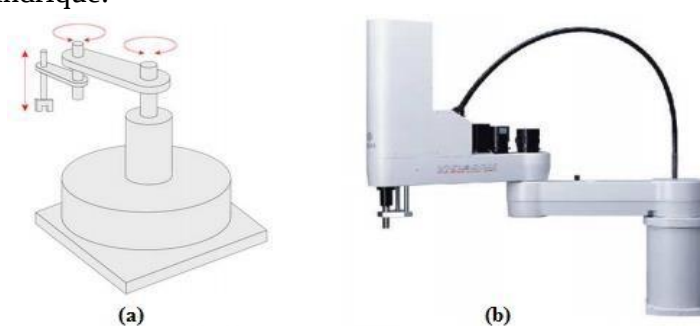


Figure.I.19 : (a) Structure SCARA, (b) Robot type SCARA (ADTECH) [1]

I.9.5. La structure 3R (anthropomorphe) :

- Reproduisent la structure d'un bras humain.
- 3 axes, série, 3R, 3DDL.

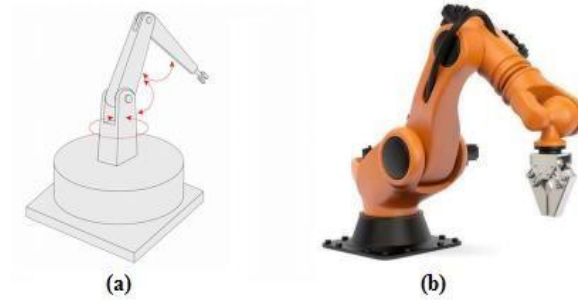


Figure.I.20: (a) Structure anthropomorphic, (b) Robot anthropomorphic (Stäubli) [1]

I.9.6. La structure parallèle:

- Plusieurs architectures parallèles sont possibles (structure Delta, structure planes, structure Hexapode) et Plusieurs chaînes cinématiques en parallèle.
- Espace de travail réduit.
- Précis (grande rigidité de la structure).

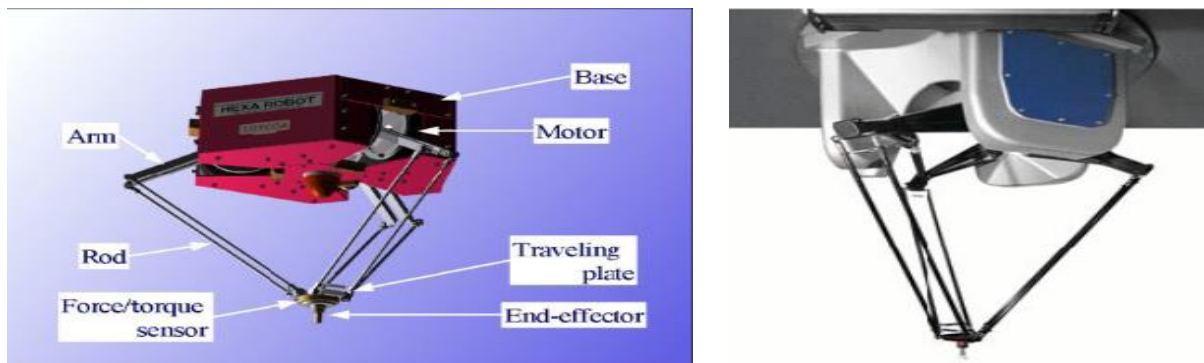


Figure I.21: Structure parallèle type HEXA à gauche et Robot type DELTA à droite. [1]

I.10. Système de coordonnées en robotique:

Pour réaliser les tâches souhaitées, l'outil doit pouvoir être déplacé dans l'espace. Ce volume, borné par les limites physiques du mécanisme, est appelé l'espace de travail. Pour identifier les points dans cet espace on utilise des repères orthonormés directs appartenant à un système de coordonnées. Ces repères sont définis à la construction du robot, mais les langages de programmation actuels permettent l'ajout de repères utilisateurs. [5]

Le repère lié au premier élément de la structure du robot est utilisé comme référence. Ce repère « base » est noté $R_0 \{X_0, Y_0, Z_0\}$, occupe une position connue dans le repère atelier (si celui-ci est

défini par l'utilisateur). Le repère lié à l'outil noté $R_N \{X_N, Y_N, Z_N\}$, occupe tout naturellement une place particulière. [5]

Chaque corps du mécanisme est activé par un actionneur. Pour un robot ne comportant que des articulations rotoïdes, on retrouvera un moteur pour chaque axe. C'est l'action de ces moteurs qui permet de situer l'outil dans l'espace de travail. On note $q_1, q_2, \dots, q_N$, les variables de position angulaire (ou articulaire) des moteurs. [5]

On appelle "configuration" le jeu de valeurs articulaires que le robot prend à un instant donné :

$q(q_1, q_2, \dots, q_N)$. Ces paramètres sont définis dans l'espace articulaire du robot. Pour passer d'un système de coordonnées à l'autre, on utilise des modèles de transformation. Ces modèles dépendent de l'architecture du robot. [5]

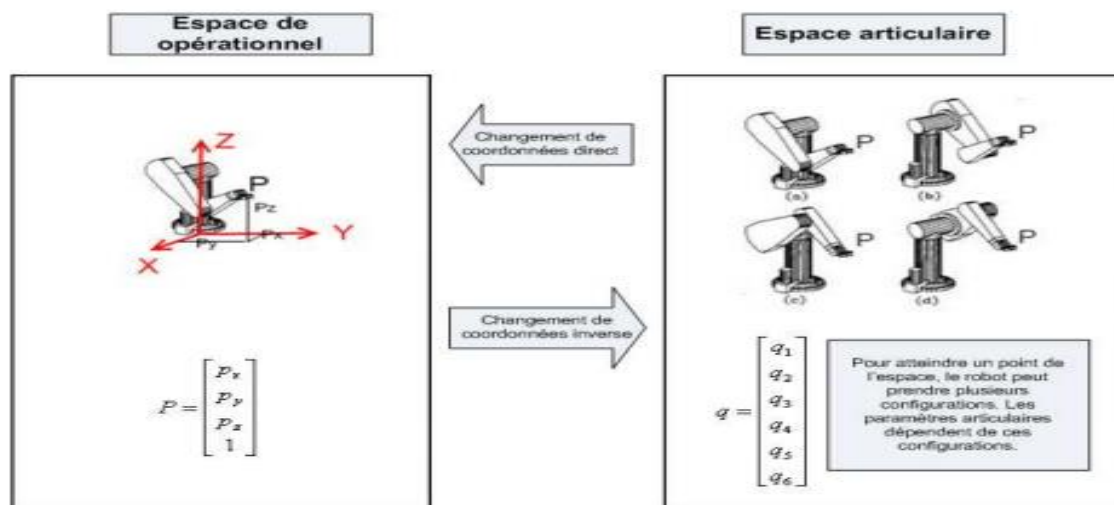


Figure. I.22 : Changements de coordonnées en robotique [5]

Conclusion:

Nous avons donné dans ce chapitre une vue général sur les bras robotisé en passant par leur définition, leurs types, classes, architectures et leur structure, anthropomorphe, cartésienne, SCARA, sphérique, cylindrique, plane, DELTA, nous avons cité les domaines d'applications les plus répandue, et décrit sa constitution, et leur Système de coordonnées.

CHAPITRE II : MODELISATION DES ROBOTS.

II.1. Modélisation d'un robot :

La conception et la commande des robots nécessitent le calcul de certains modèles mathématiques, tels que :

- Les modèles de transformation entre l'espace opérationnel (dans lequel est définie la situation de l'organe terminal) et l'espace articulaire (dans lequel est définie la configuration du robot). Parmi ces modèles, on distingue :
 - Les modèles géométriques direct et inverse qui expriment la situation de l'organe terminal en fonction de la configuration du mécanisme et inversement,
 - Les modèles cinématiques direct et inverse qui expriment la vitesse de l'organe terminal en fonction de la vitesse articulaire et inversement,
- Les modèles dynamiques définissant les équations du mouvement du robot, qui permettent d'établir les relations entre les couples ou forces exercés par les actionneurs et les positions, vitesses et accélérations des articulations.

Définir les différentes tâches d'un robot exige de pouvoir positionner l'organe terminal par rapport à un repère de référence. En effet :

- Les informations proprioceptives (issues du S.M.A.) sont généralement définies dans des repères liés aux différents solides du robot.
- La position à atteindre est souvent définie dans un repère lié au socle du robot,
- L'objet à saisir peut-être défini dans un repère mobile indépendant du robot (par exemple, des pièces à prendre sur un tapis roulant),
- Les informations extéroceptives (issues de l'environnement) sont définies dans divers repères.

Aussi, il faut un référentiel commun afin de « ramener » les diverses informations dans un même référentiel, notamment pour concevoir les consignes des actionneurs.

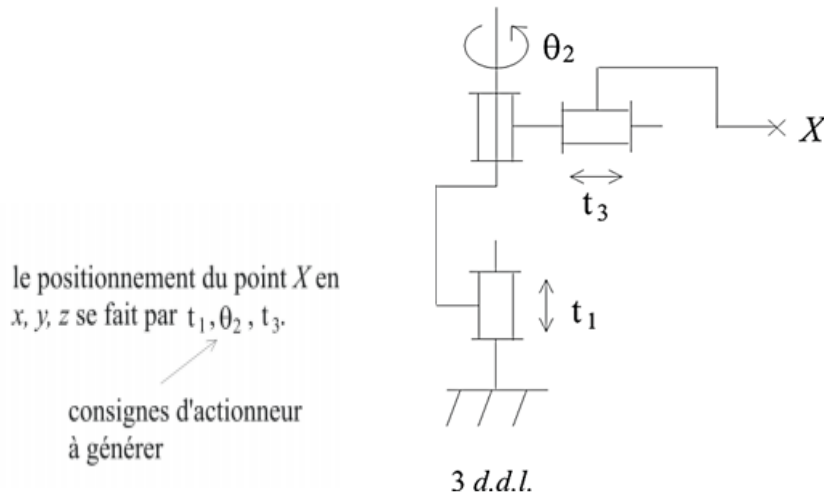


Figure II.1 : Structure d'un robot sériel à 3ddl.

II.2. Modélisation cinématique :

La première étape du processus est la modélisation du robot. Il existe plusieurs techniques de modélisation, chacune possédant leurs forces et leurs faiblesses. [9]

II.2. 1. Modélisation cinématique directe (MCD) :

Le modèle cinématique directe du robot décrit les variations élémentaires des coordonnées opérationnelles en fonction des variations élémentaires des coordonnées articulaire. Il est noté:

$$\begin{cases} dX = J(q)dq \\ X = f(q, q) \end{cases} \quad (\text{II.1})$$

$J(q)$: désigne la matrice jacobéenne de dimensions $(m \times n)$ du mécanisme, et est égale (dx/dq) en fonction de la configuration articulaire q . Elle est à la base du modèle cinématique inverse, permettent de calculer de façon initiatrice les variables articulaires q connaissant les coordonnées opérationnelles P . Le modèle cinématique directe, déterminé par les dérivées par rapport au temps $\dot{p}$ et $\dot{q}$ des coordonnées opérationnelles et des coordonnées articulaires, exprime les vitesses opérationnelles on fonction des vitesses articulaires [6].

❖ Paramètres Denavit-Hartenberg:

La deuxième étape de la méthode est d'identifier les paramètres qui relient les référentiels attribués. Selon la méthode D-H, quatre paramètres sont nécessaires pour décrire la relation complète entre deux référentiels successifs. Ces paramètres ont la signification suivante:

- a_i est la distance entre Z_i et Z_{i+1} selon X_i .
- α_i est l'angle entre Z_i et Z_{i+1} autour de X_i .
- d_i est la distance entre X_{i-1} et X_i selon Z_i .
- θ_i est l'angle entre X_{i-1} et X_i autour de Z_i .

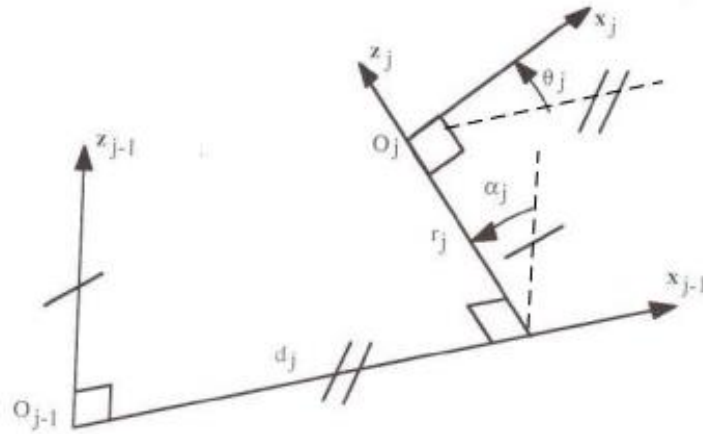


Figure II.2 : Paramètres de Denavit-Hartenberg [5]

Définir les transformations nécessaires pour relier les référentiels successifs entre eux afin de former la matrice homogène suivante :

$${}^{i-1}T_i = R_x(\alpha_{i-1}) \text{Trx}(a_{i-1}) R_z(\theta_i) \text{Trz}(d_i) \quad (\text{II.2})$$

Pour les axes parallèles consécutifs, le paramètre β_{i-1} est ajouté et forme ainsi la matrice homogène suivante :

$${}^{i-1}T_i = R_x(\alpha_{i-1}) \text{Trx}(a_{i-1}) R_y(\beta_{i-1}) R_z(\theta_i) \text{Trz}(d_i) \quad (\text{II.3})$$

Les paramètres a et α sont indexés sur une articulation en arrière (i.e. $i-1$) et d et θ sur la même articulation i . Cela va faciliter le remplissage de la table des paramètres, nécessaire pour l'obtention des matrices de transformation homogènes entre articulations. Après la multiplication de ces matrices, on obtient la matrice de transformation homogène reliant le poignet à la base du robot. Grâce à ces matrices on sera capable d'appliquer la cinématique directe et inverse.

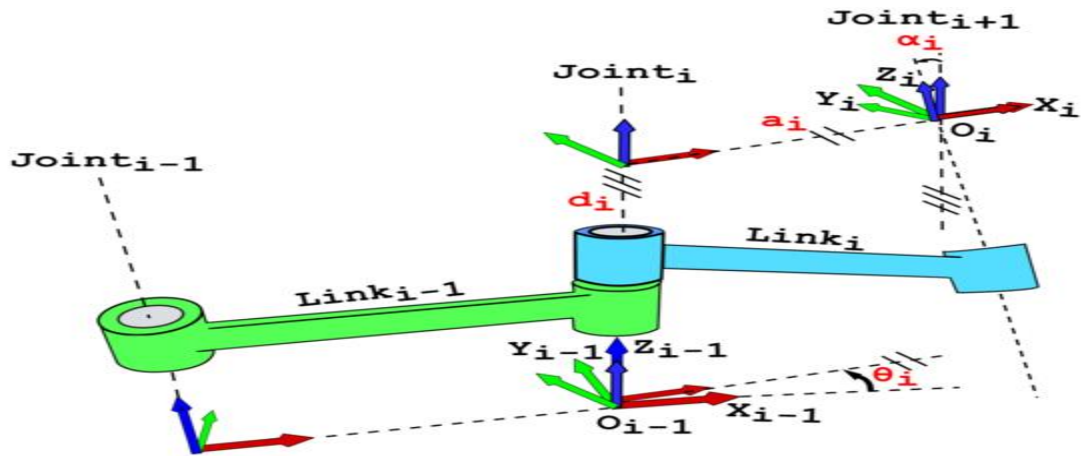


Figure II.3 : les paramètres D-H [7]

Où : R_x est la matrice de rotation autour de l'axe x par rapport au système de coordonnées de l'articulation $i-1$, donc:

$$R_x(\beta) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \beta & -\sin \beta & 0 \\ 0 & \sin \beta & \cos \beta & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (\text{II.4})$$

D_x est la matrice de translation au long de l'axe x par rapport au nouveau système de coordonnées obtenu lors de la rotation précédente,

$$D_x(\beta) = \begin{pmatrix} 1 & 0 & 0 & \beta \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (\text{II.5})$$

Rz est la matrice de rotation autour de l'axe z par rapport au nouveau système de coordonnées obtenu lors des transformations précédentes, et :

$$R_z(\beta) = \begin{pmatrix} \cos \beta & -\sin \beta & 0 & 0 \\ \sin \beta & \cos \beta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (\text{II.6})$$

Dz est la matrice de translation au long de l'axe z par rapport au nouveau système de coordonnées obtenu lors des transformations précédentes.

$$D_z(\beta) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \beta \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (\text{II.7})$$

La valeur entre parenthèses est la valeur de rotation ou translation selon un axe à appliquer. En appliquant ces transformations on obtient donc la matrice résultante suivante:

$${}^{i-1}T_i = \begin{pmatrix} \cos \theta_i & -\sin \theta_i & 0 & \alpha(i-1) \\ \sin \theta_i \cos \alpha(i-1) & \cos \theta_i \cos \alpha(i-1) & -\sin \alpha(i-1) & -\sin \alpha(i-1)di \\ \sin \theta_i \sin \alpha(i-1) & \cos \theta_i \sin \alpha(i-1) & \cos \alpha(i-1) & \cos \alpha(i-1)di \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (\text{II.8})$$

II.2. 2.Modélisation cinématique Inverse:

La cinématique décrite jusqu'ici est dite directe, c'est-à-dire, qu'à partir d'une configuration donnée du robot (connaissance des différents angles des articulations), nous sommes passés d'un référentiel (celui de l'outil) à un autre (la base). Une approche plus utile mais loin d'être aussi simple que la cinématique directe, est la cinématique inverse. C'est celle

qui intéresse le plus les utilisateurs de robots, puisqu'à partir d'un point de l'espace à atteindre, la configuration géométrique (dépendant des angles articulaires) doit être déduite. Le problème met en place n équations fortement non linéaires à n inconnues. Dans un cas linéaire, nous pouvons connaître le nombre de solutions à partir du déterminant du système. Ce n'est plus le cas ici. Avec n équations à n inconnues, nous pouvons très bien obtenir une infinité de solutions. Plusieurs techniques ont été utilisées dans le cadre du travail. Seules deux seront exposées: l'approche algébrique et l'approche optimisation.

❖ Approche algébrique :

En plus d'être fortement non linéaire, le système d'équations est transcendant. En effet, les matrices homogènes impliquent une quantité énorme de sinus et cosinus rendant la résolution encore plus complexe.

$$\begin{pmatrix} X0 \\ Y0 \\ Z0 \\ 1 \end{pmatrix} = {}^0T_1.{}^1T_2.{}^2T_3.{}^3T_4.{}^4T_5.{}^5T_6. \begin{pmatrix} X6 \\ Y6 \\ Z6 \\ 1 \end{pmatrix} \quad (II.9)$$

$${}^0T_6 = [{}^0T_6]^{-1} \quad (II.10)$$

Par conséquence:

$$T = \begin{pmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (II.11)$$

$$T^{-1} = \begin{pmatrix} n_x & n_y & n_z & -((p_x.n_x) + (p_y.n_y) + (p_z.n_z)) \\ o_x & o_y & o_z & -((p_x.o_x) + (p_y.o_y) + (p_z.o_z)) \\ a_x & a_y & a_z & -((p_x.a_x) + (p_y.a_y) + (p_z.a_z)) \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (II.12)$$

La technique de la substitution de variables peut être appliquée grâce aux relations suivantes :

$$\cos(\theta) = \frac{1-u^2}{1+u^2} \quad \text{et} \quad \sin(\theta) = \frac{2u}{1+u^2} \quad \text{avec} \quad u = \tan\left(\frac{\theta}{2}\right) \quad (\text{II.13})$$

Dans le but de faciliter les calculs gigantesques avec 6 ddl (degrés de liberté), seulement 4 des 6 articulations sont étudiées [8].

II.3. Robot serial et son étalonnage :

Le domaine de l'étalonnage des robots sériels a été exploré en profondeur par de multiples chercheurs. Certains se sont penchés sur l'élaboration de techniques de modélisation optimales. D'autres ont testé différentes techniques d'optimisation pour identifier les paramètres des modèles établis. Plusieurs méthodologies ont été explorées et différents instruments de mesure ont été utilisés, chacun présentant ses avantages et ses inconvénients. Tout cela a laissé un large éventail de connaissances pour les générations futures de chercheurs. Ainsi, ce travail présente les ouvrages jugés les plus importants à la compréhension de la problématique associée à ce travail de recherche. [9]

II.3.1. Niveaux d'étalonnage:

Le choix du niveau d'étalonnage est une étape cruciale qui doit être réalisée parallèlement à l'étape de modélisation du robot. [9]

❖ Niveau 1:

L'étalonnage de niveau 1 (c.-à-d. étalonnage des joints) permet d'augmenter la précision du robot en identifiant la relation mathématique qui existe entre les valeurs de sortie des transducteurs des joints et les valeurs angulaires obtenues réellement. [9]

❖ Niveau 2:

L'étalonnage de niveau 2 (c.-à-d. étalonnage géométrique) permet d'augmenter la précision du robot non seulement en déterminant les paramètres qui représentent les bonnes relations angulaires sur chacun des joints, mais également en identifiant les paramètres qui décrivent la géométrie du robot. [9]

❖ Niveau 3:

L'étalonnage de niveau 3 (c.-à-d. étalonnage non géométrique) permet d'augmenter la précision du robot en incluant un ou plusieurs paramètres non géométriques dans le modèle mathématique du robot. Il existe également un bon nombre de travaux sur le sujet. [9]

II.3.2. Types d'étalonnage:

Il existe deux types d'étalonnage : relatif et absolu (Whitney et al. 1986). L'étalonnage relatif utilise un modèle qui permet de localiser le robot par rapport à un référentiel arbitraire dont la localisation est définie par un ensemble de valeurs articulaires. Aucun référentiel externe n'est utilisé. L'étalonnage absolu utilise un modèle qui permet de localiser l'outil dans différentes configurations en fonction d'un repère externe. Avec ce type d'étalonnage, la pose de la base par rapport au référentiel doit être identifiée. C'est le type d'étalonnage qui sera effectué dans ce travail. [9]

II.3.3. Classification des méthodes d'étalonnage:

Il existe deux classes principales d'étalonnage : en boucle ouverte et en boucle fermée.

❖ Boucle ouverte:

L'étalonnage en boucle ouverte consiste à utiliser un système de mesure externe pour acquérir les mesures nécessaires à l'identification des paramètres. L'avantage principal de cette méthode est qu'elle permet généralement l'acquisition de mesures dans tout l'espace de travail du robot. [9]

❖ Boucle fermée:

L'étalonnage en boucle fermée, aussi connue sous le nom d'étalonnage autonome (Bennett et al. 1991), est défini comme étant le procédé qui permet de déterminer le modèle du robot en utilisant seulement l'information de ces capteurs internes. [9]

Conclusion

Nous avons donné dans ce chapitre une vue général sur la cinématique avec leurs types(la cinématique direct et la cinématique inverse) , et les équations de chaque cas, et entrée sur la définition et niveaux d'étalonnage et leurs types, avec la classification.

CHAPITRE III : CONCEPTION ET REIALISATION
D'UN ROBOTS SERIEL A 6DDL.

III.1. Justification du choix du robot:

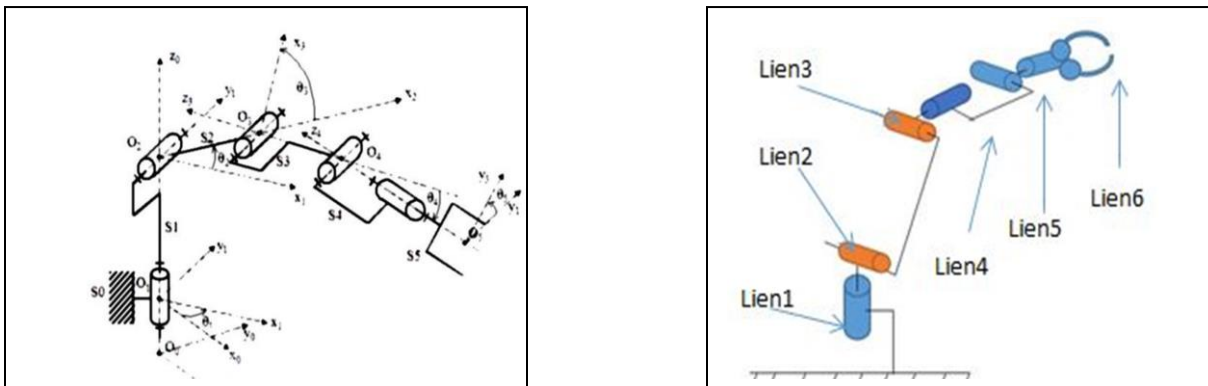
Un robot sériel à 6DDL est un outil révolutionnaire et qui une fois programmé peut accomplir diverses tâches plus ou moins complexes. De plus, il est plus utilisé et plus complet dans les usages généraux. Surtout l'industriel, car il a plus de liberté dans l'exécution des tâches requises, nous les mentionnons comme : saisir et déplacer des objets, laver la vaisselle, trier et ranger des jouets, apporter des objets.

Et il exécute des commandes importantes telles que la sculpture, l'impression normale et tridimensionnelle, la peinture et autres... à l'aide de supports et d'un outil installé dans le mandrin qui s'appelle l'organe terminale ; Ce dernier peut même être exploité de la cuisine, etc..

Pour la préparation de notre plat du jour tous ces travaux sont réalisés par programmation et simulation via un ordinateur, des appareils android ou un programme ajouté dans ses circuits électroniques (Arduino). [19].

III.2. Présentation de la chaîne Cinématique du robot:

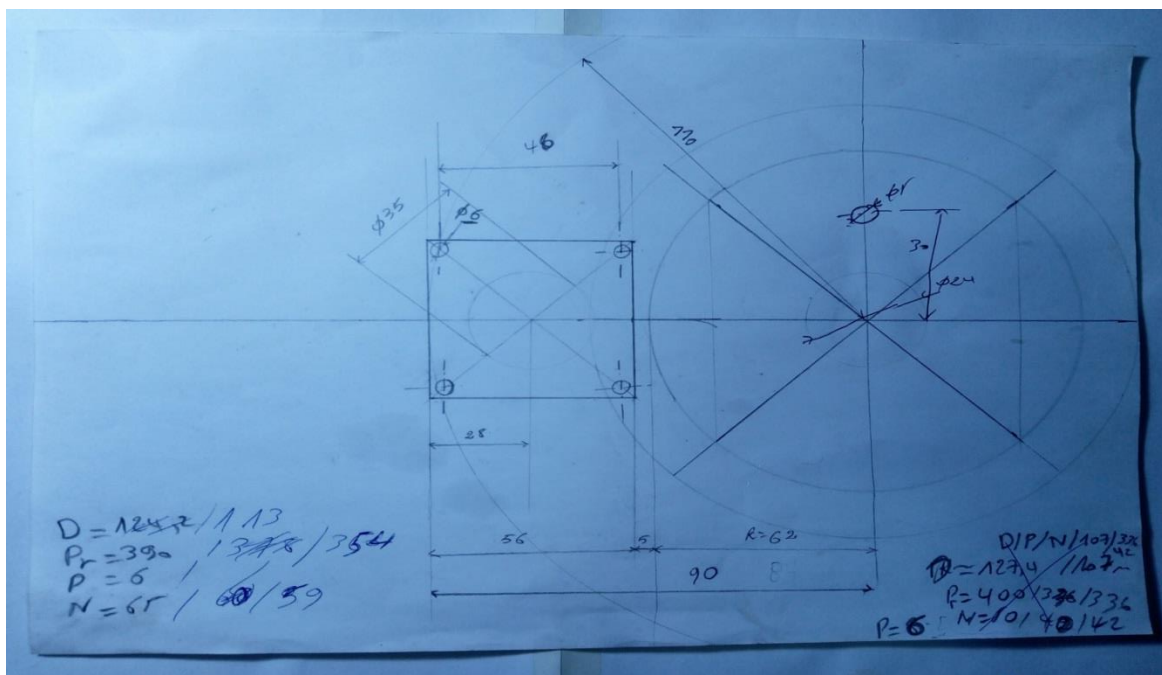
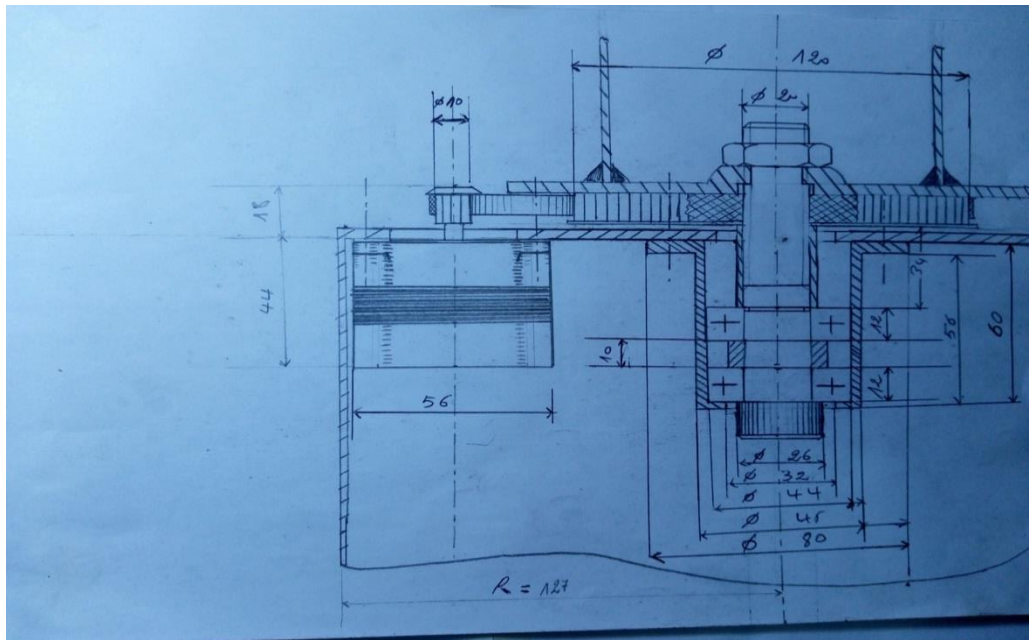
La structure de la chaîne cinématique du robot est composée des modules à une structure hybride modulaire est similaire à la plate-forme Gough-Stewart, qui se compose de plusieurs chaînes cinématiques identiques reliées entre elles et à une plate-forme mobile par des articulations de formes diverses telles que sphériques ou cylindriques et reliées à une base fixe par des axes. Un moteur pas à pas est utilisé pour modifier la longueur et l'angle des chaînes cinématiques, notre prototype visé ci-contre (**Figure III.1**). [20]

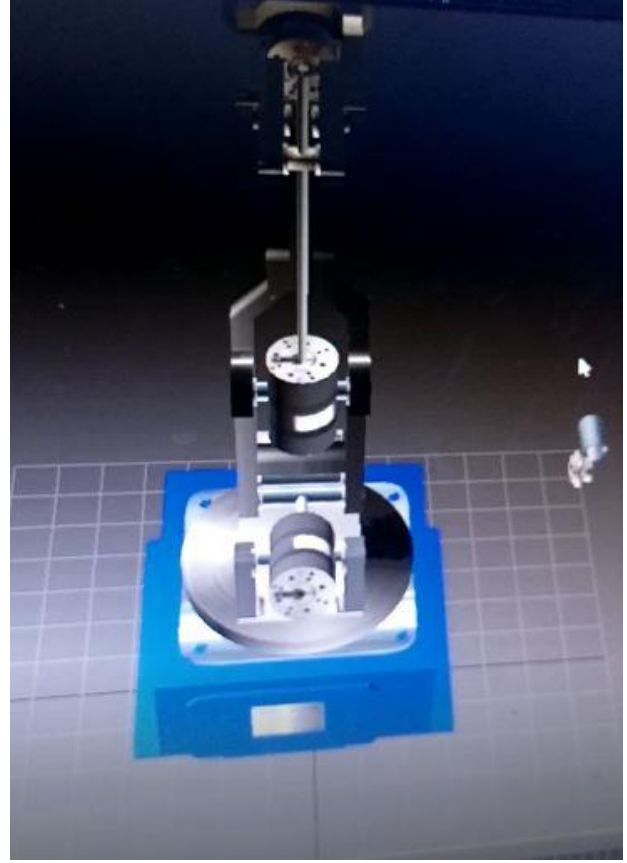
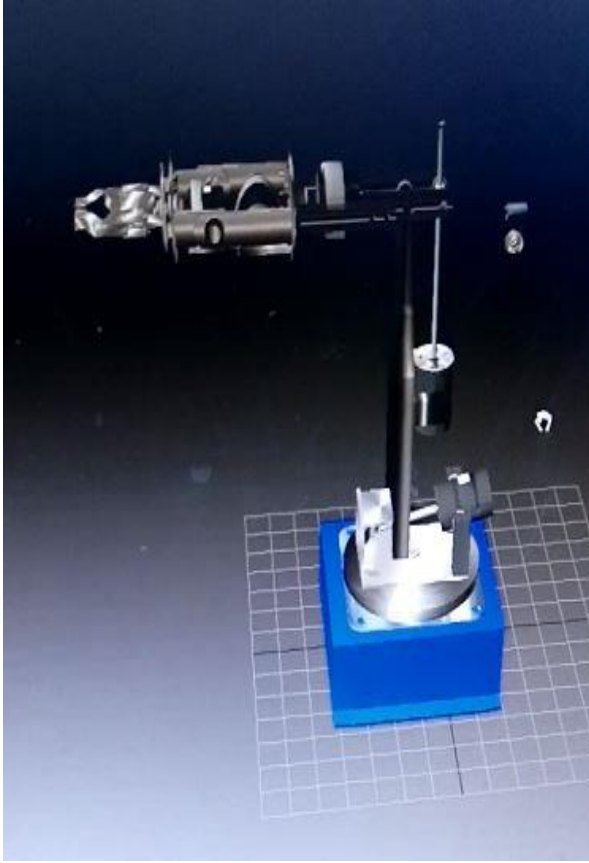


Placement des repères des articulations de la chaîne cinématique (**Figure III.1**).

III.3. Présentation des croquis de conception:

Comme nous le savons, tout projet commence par des idées clés qui sont placées à travers un croquis pour l'incarnation, après vient l'étape de la modification, puis l'amélioration, suivie de la mise en œuvre tout au long du processus de conception CAO (conception assistée par ordinateur), puis la réalisation de ce projet et voici quelques croquis et photos au cours de notre opération de conception et réalisation:





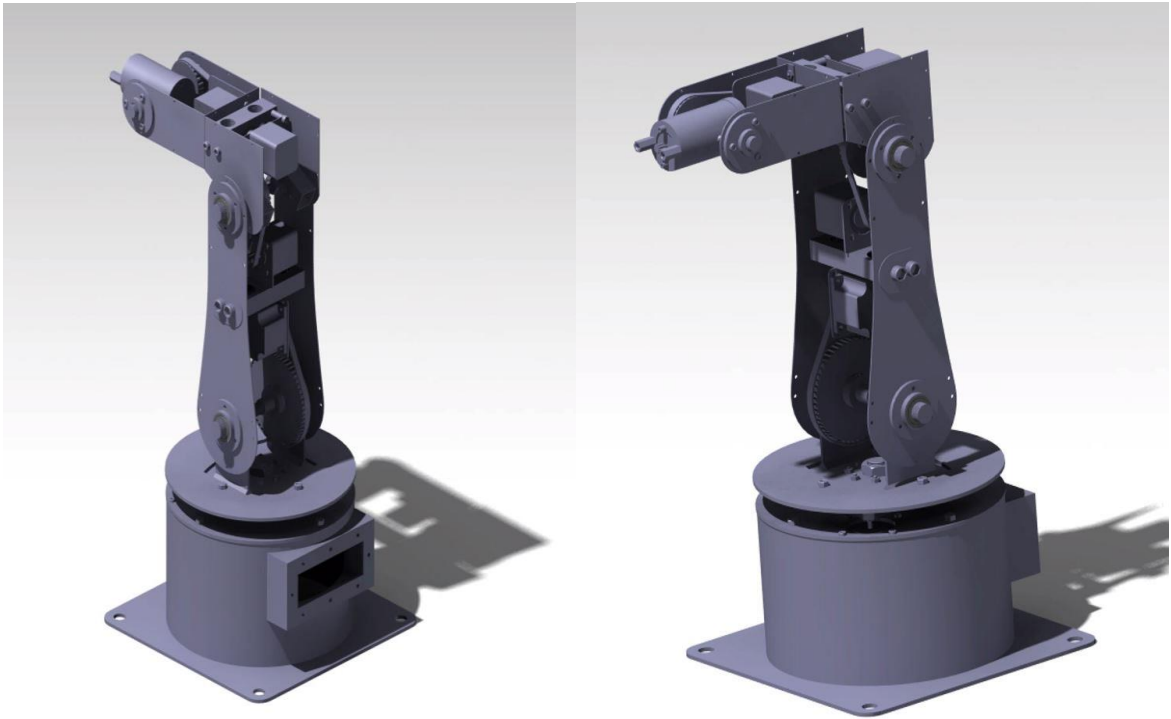
III.4. Les différentes parties du robot (groupes d'assemblages):

Dans notre projet, nous nous sommes engagés dans le système de segmentation du robot en quatre Groupes, chaque partie (groupe) formant un groupe de pièces mécaniques qui ont une relation cinétique reliées les unes aux autres, et nous vous montrerons ces pièces en détail :

- 1- Conception Cao.
- 2- Dessin définition et d'assemblage pour chaque groupe.
- 3- Dessin définition pour chaque pièce de l'ensemble.
- 4- Fabrication de la pièce avec preuve d'image.

III.4.1. L'assemblage global du robot: Après la conception par la méthode CAO de chaque pièce nous allons faire une synthèse final d'assemblage des différents pièces et partie du robot pour avoir un aspect acceptable préparant à la réalisation.

III.4.1.1 conception CAO: Voici la vue générale virtuelle du robot avant le commencement de la réalisation.



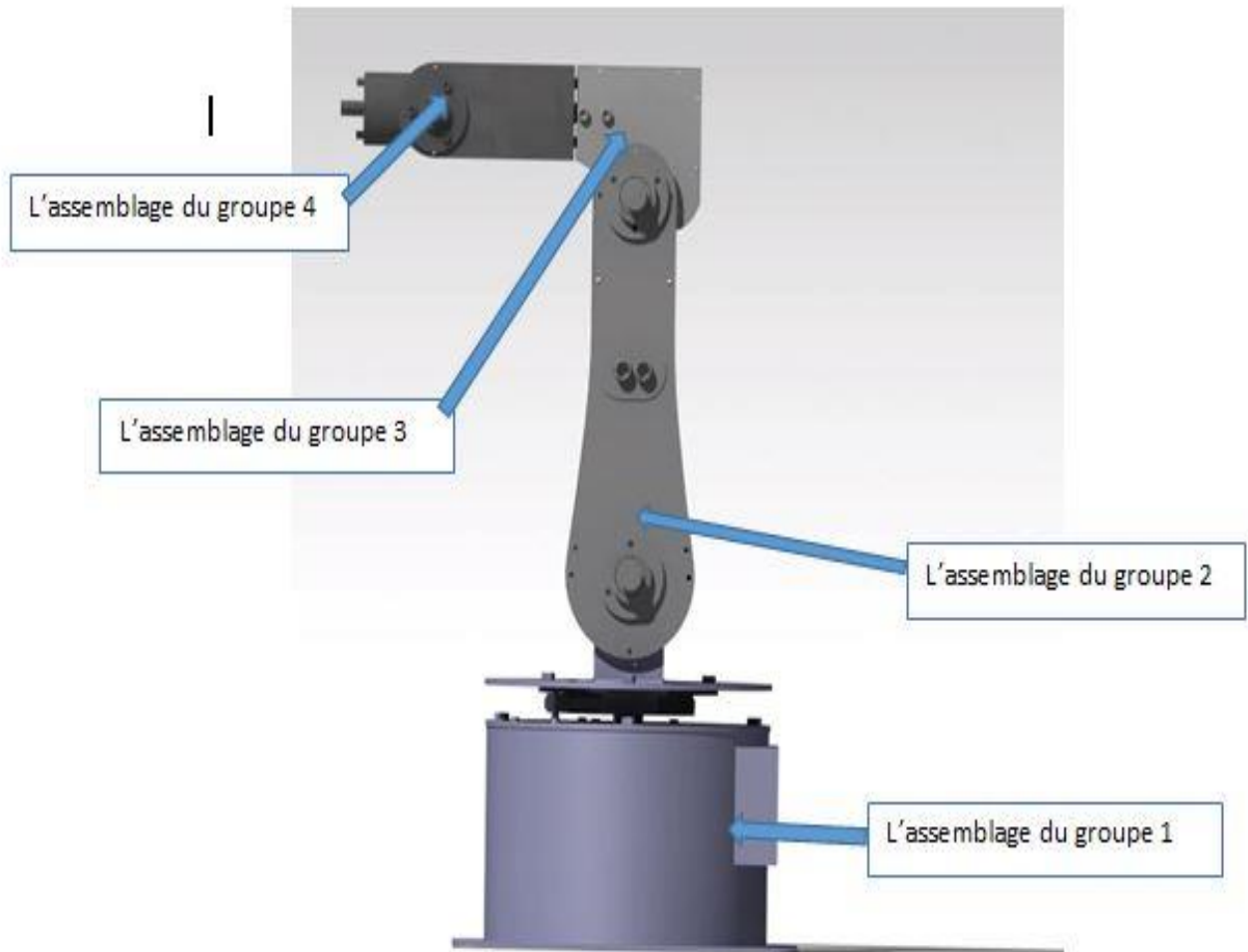
Vue globale du robot en CAO (**Figure III.2**).

III.4.1.2 Conception réelle après réalisation: les images suivantes représente la vue générale réel du robot après la réalisation.



Figure. Vue globale de la conception réelle du robot après la réalisation (**Figure III.3**)

III.4.2. Les différents groupes d'assemblage du robot : Nous avons devisé le robot à quatre parties principales chaque partie formes un groupe des pièces liés entres eux voir la figure par contre :

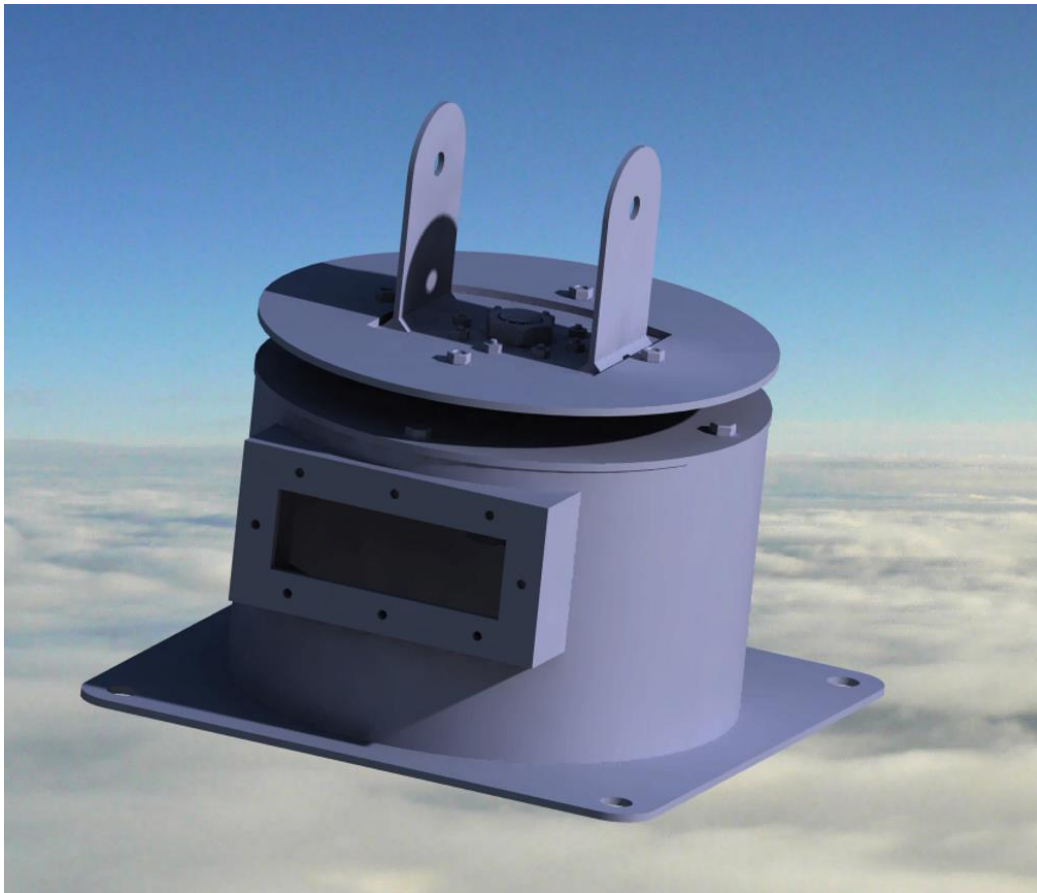


L'assemblage du groupe1 (Figure III.4).

III.4.2.1. Dimensionnement et Assemblage du groupe1 du robot:

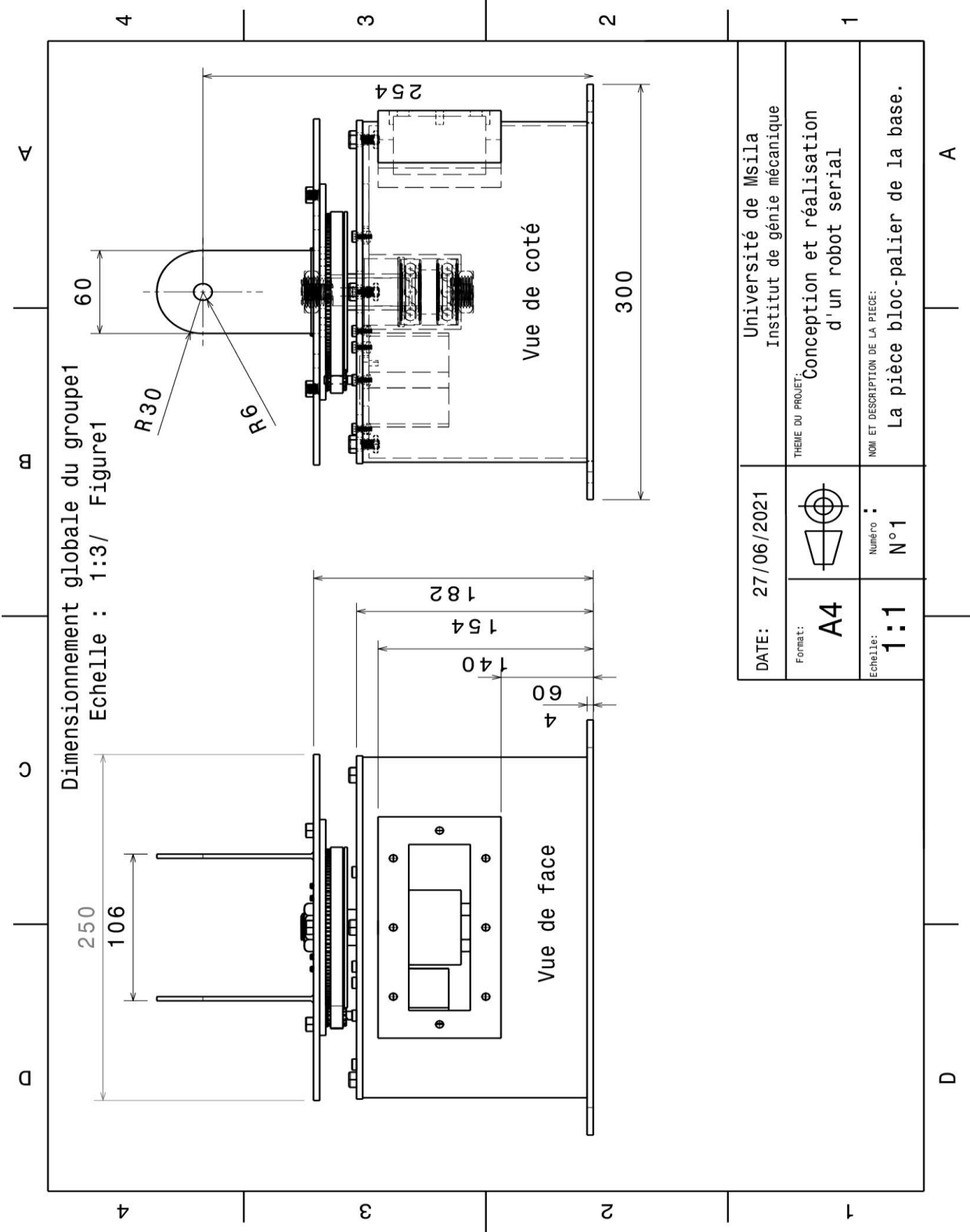
Dans cette étape, nous allons créer la forme et la composition générale des différentes pièces à travers lesquelles un processus de dessin peut être effectué pour déterminer les dimensions et un dessin définition des pièces et des groupes assemblés.

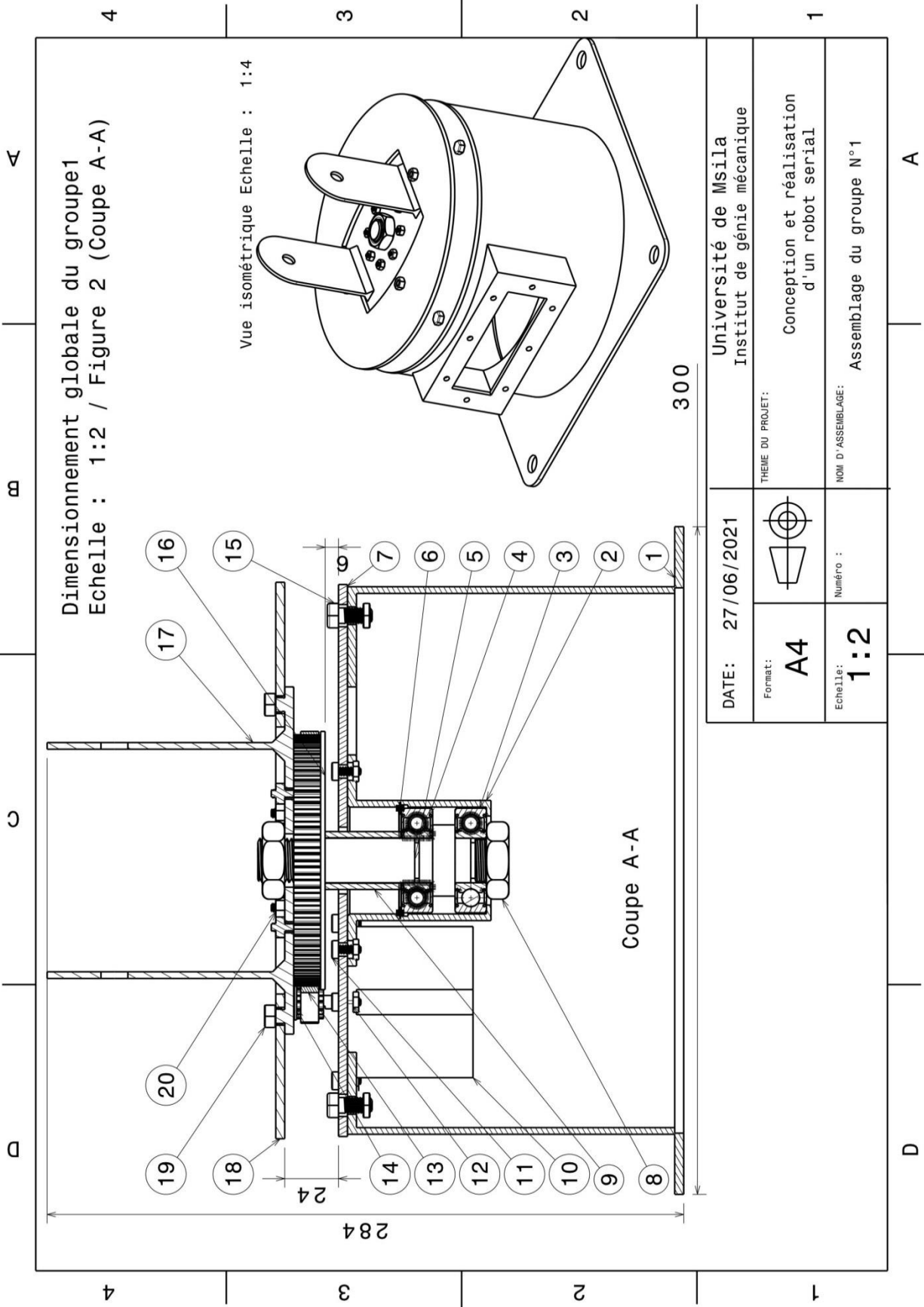
✓ **.A: Représentation CAO du groupe 1:.**



Vue 3D de la conception CAO du groupe1 (la base du robot) (**Figure III.5**).

✓ .B: Dessin d'assemblage et dimensionnement du groupe 1:





✓ **C: Conception réelle et fabrication du groupe 1:**



Figure. Image de la conception réelle du groupe 1 : (La base du robot) (**Figure III.6**).

III.4.2.1.1 : Conception et dimensionnement de la pièce n°1: Il y a trois étapes pour guider à la fin de la réalisation pour fabriquer la pièce.

✓ **.A: Conception CAO de la pièce n°1 :**

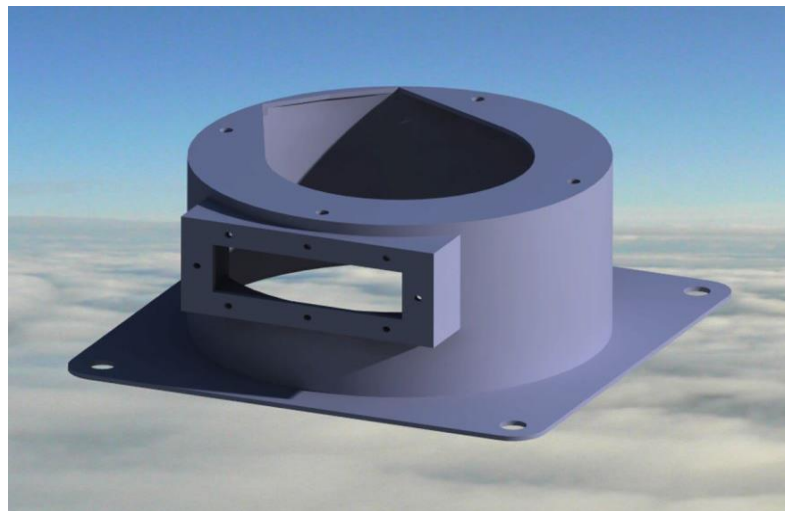
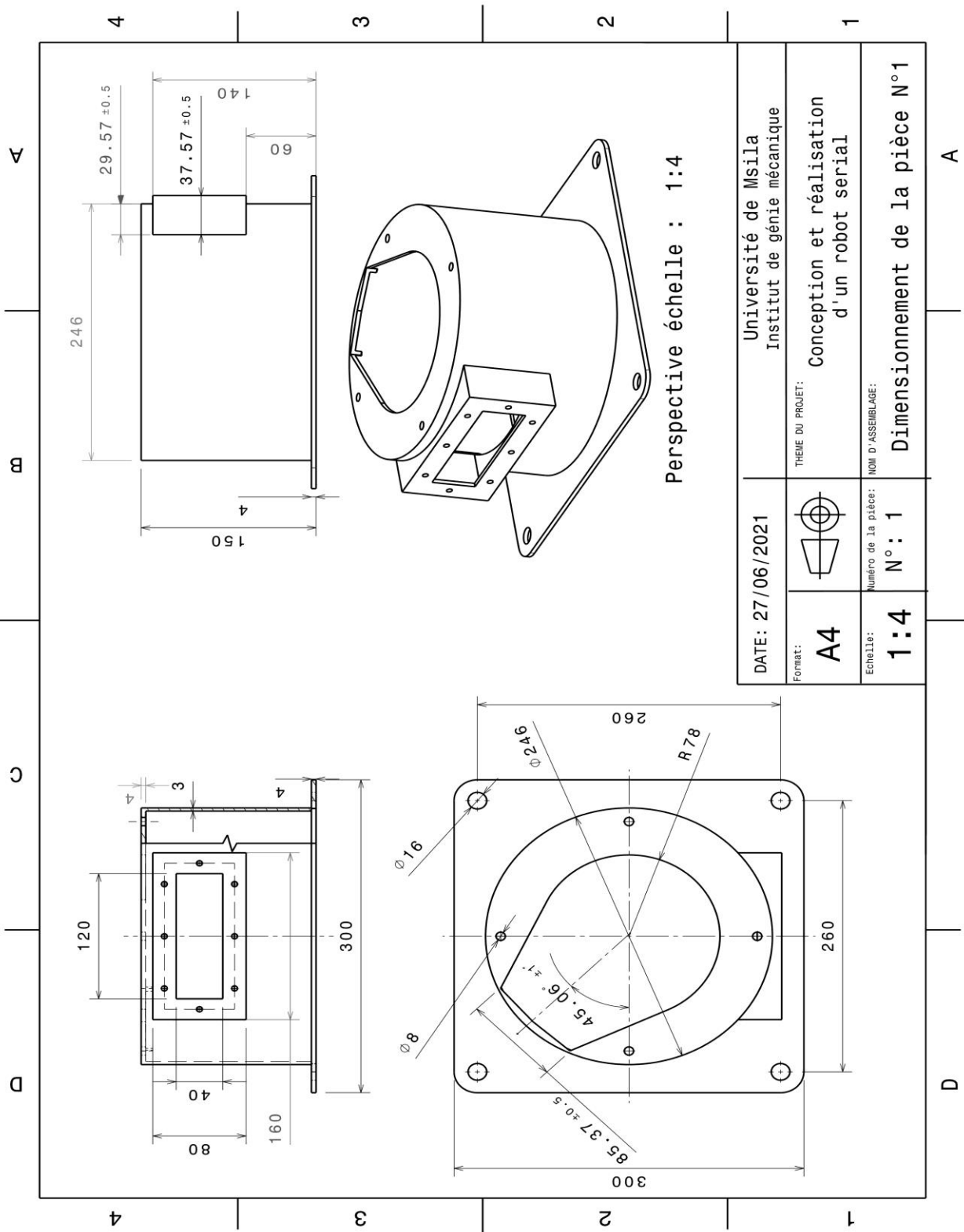


Figure. Vue 3D de la conception CAO de la pièce N°1 : (Le socle de la base du robot) (**Figure III.7**).

✓ .B: Dessin définition la pièce n°1 de la pièce n°1:



✓ .C: Conception réelle et fabrication de la pièce n°1:

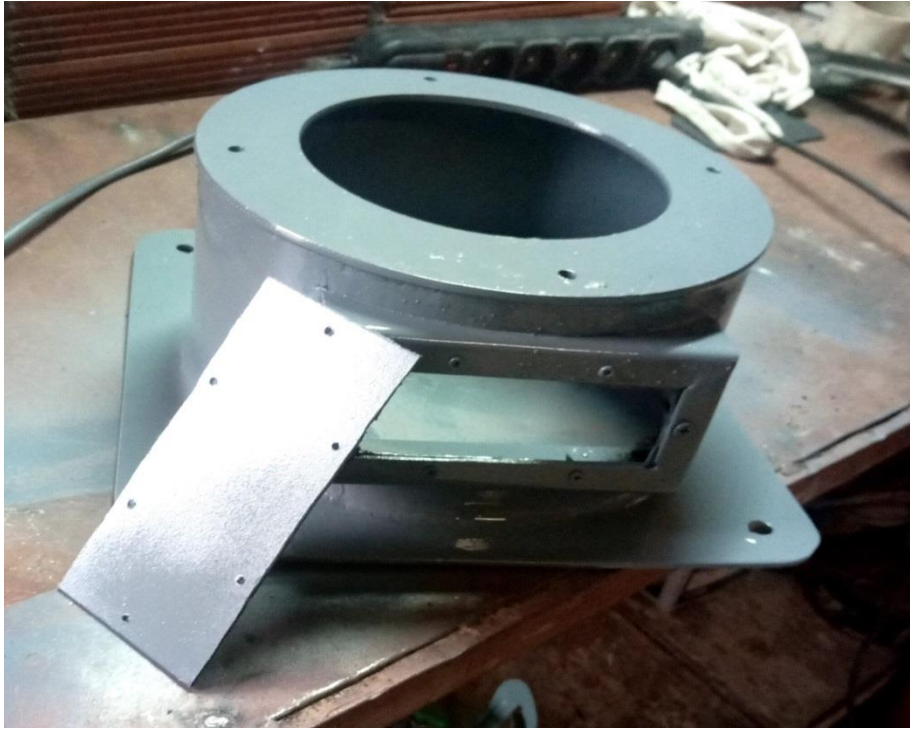


Figure. Image de la conception réelle de la pièce N°1 : (Le socle de la base du robot) (**Figure III.8**).

III.4.2.1.2 : Conception et dimensionnement de la pièce n°2: Il y a trois étapes pour guider à la fin de la réalisation pour fabriquer la pièce

✓ **A: Conception CAO de la pièce n°2 :**

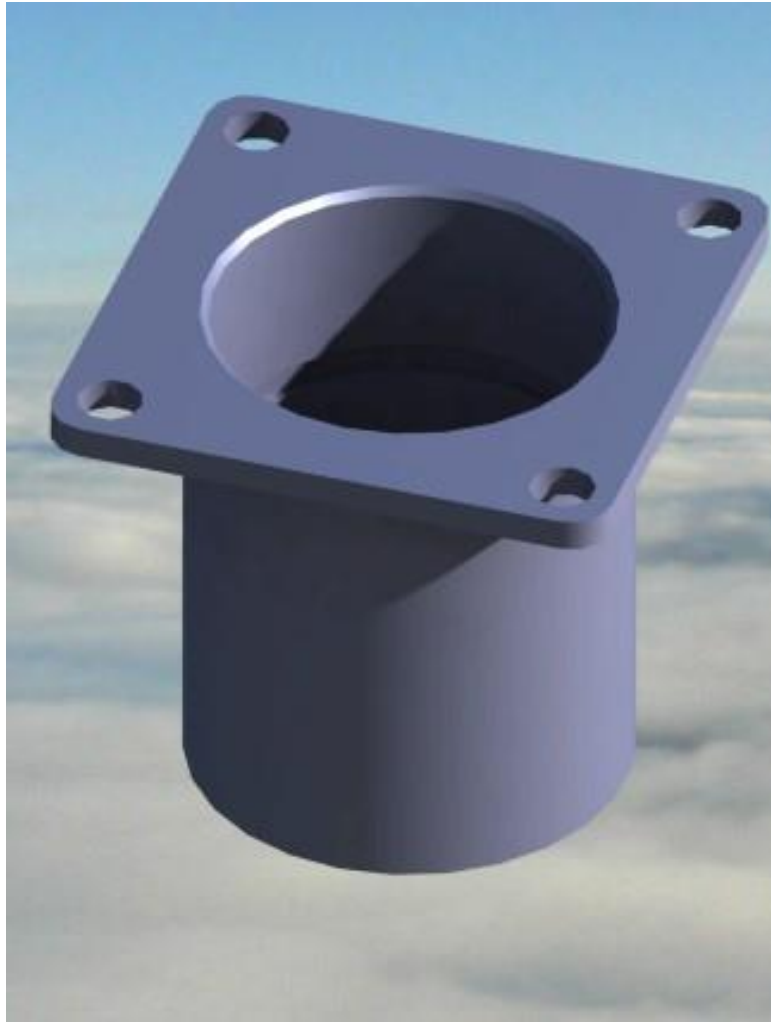
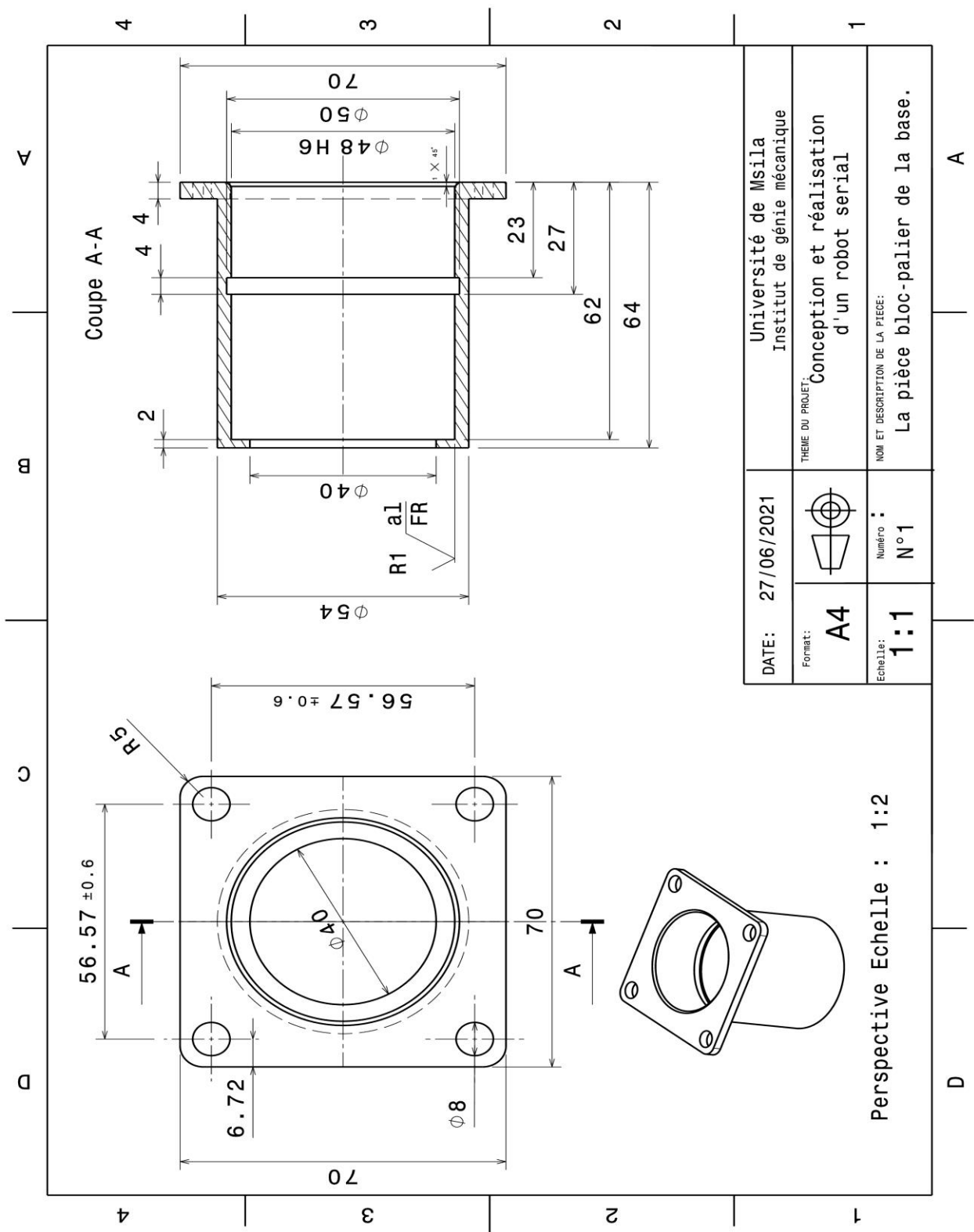


Figure. Vue 3D de la conception CAO de la pièce N°2 : (Bloc-palier de la base) (**Figure III.9**).

✓ .B: Dessin définition de la pièce n°2:



✓ .C: Conception réelle et fabrication de la pièce n°2:



Figure. Image de la conception réelle de la pièce N°2 : (Bloc-palier de la base) (**Figure III.10**).

Remarque : Les spécifications techniques des pièces n°3, n°4 se sont préfabriquées dans tableau des nomenclatures et description pièces préfabriquées à acheter ci-dessous à la fin du chapitre III.

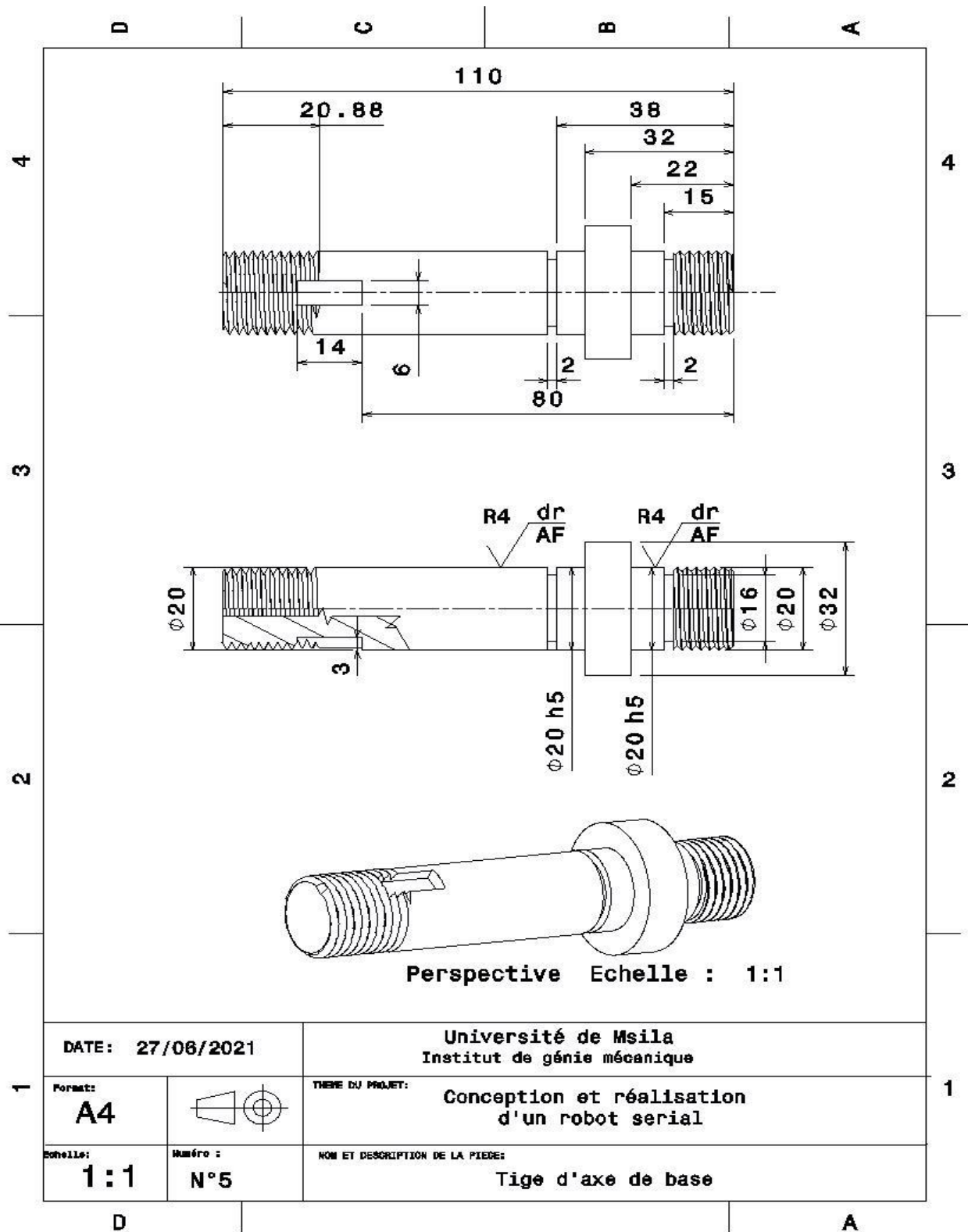
III.4.2.1.3 : Conception et dimensionnement de la pièce n°5: Il y a trois étapes pour guider à la fin de la réalisation pour fabriquer la pièce.

✓ **.A: Conception CAO de la pièce n°5 :**



Figure. Vue 3D de la conception CAO de la pièce N°5 : (Tige-axe de la base) (**Figure III.11**).

✓ .B: Dessin définition la pièce n°1 de la pièce n°5:



✓ .C: Conception réelle et fabrication de la pièce n°5:



Figure. Image de la conception réelle de la pièce N°5 : (Tige-axe de la base) (**Figure III.12**).

Remarque : Les spécifications techniques des pièces n°6 s'est préfabriquées dans tableau des nomenclatures et description pièces préfabriquées à acheter ci-dessous à la fin du chapitre III.

III.4.2.1.4 : Conception et dimensionnement de la pièce n°7: Il y a trois étapes pour guider à la fin de la réalisation pour fabriquer la pièce.

✓ **.A: Conception CAO de la pièce n°7 :**

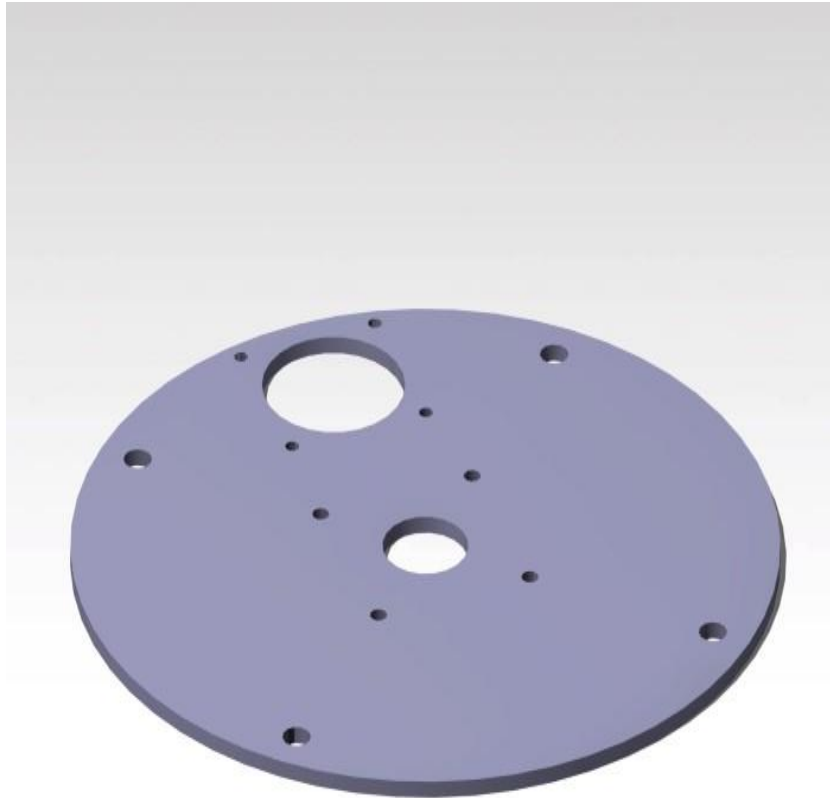
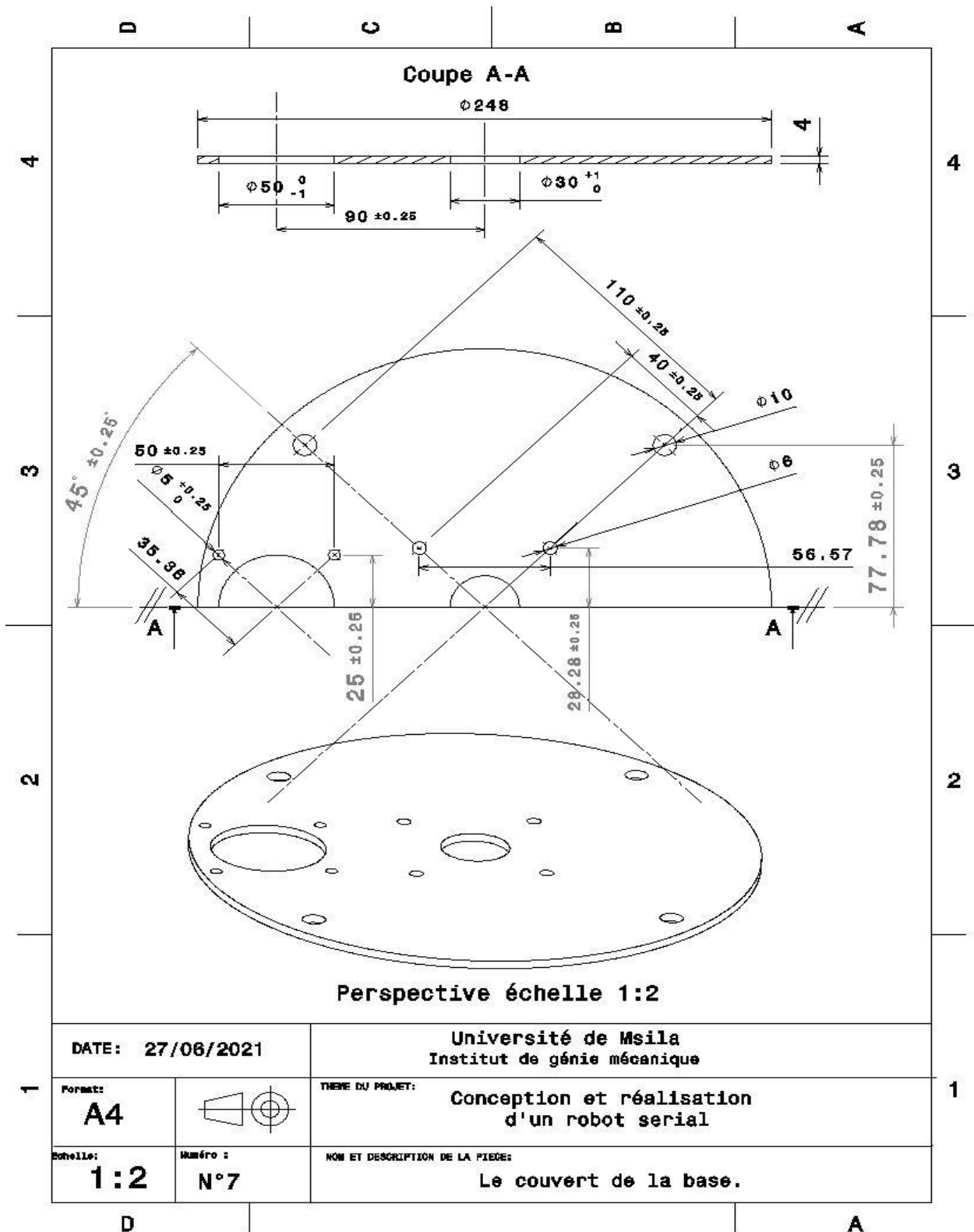


Figure. Vue 3D de la conception CAO de la pièce N°7 : (couvert-socle de base) (**Figure III.13**).

✓ .B: Dessin définition la pièce n°1 de la pièce n°7:



✓ .C: Conception réelle et fabrication de la pièce n°7:

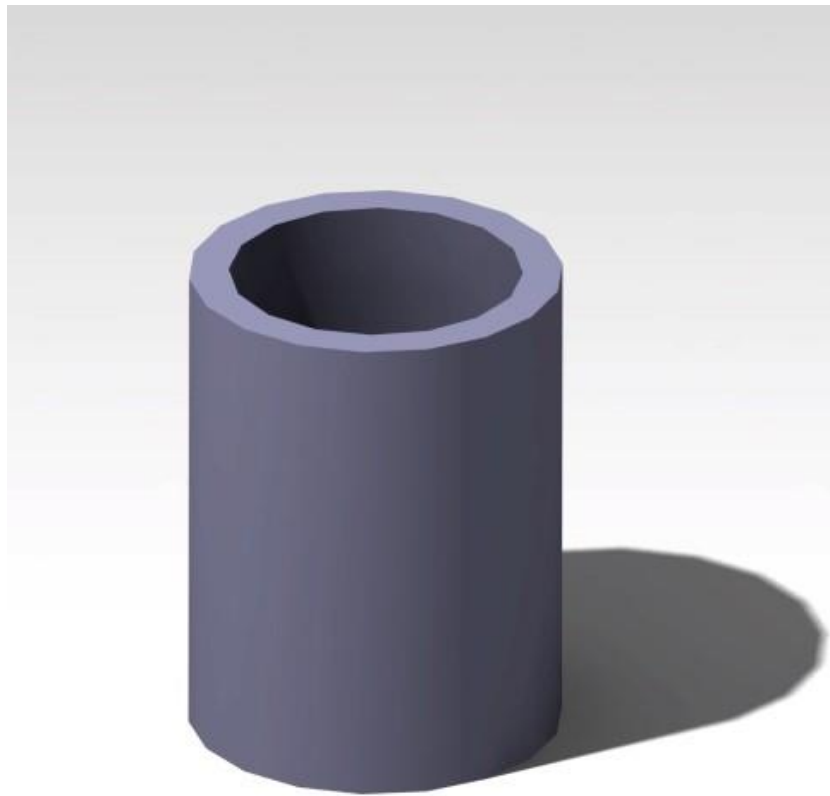


Figure. Image de la conception réelle de la pièce N°5 :(couvert-socle de base) (**Figure III.14**).

Remarque : Les spécifications techniques des pièces n°8 s'est préfabriquées dans tableau des nomenclatures et description pièces préfabriquées à acheter ci-dessous à la fin du chapitre III.

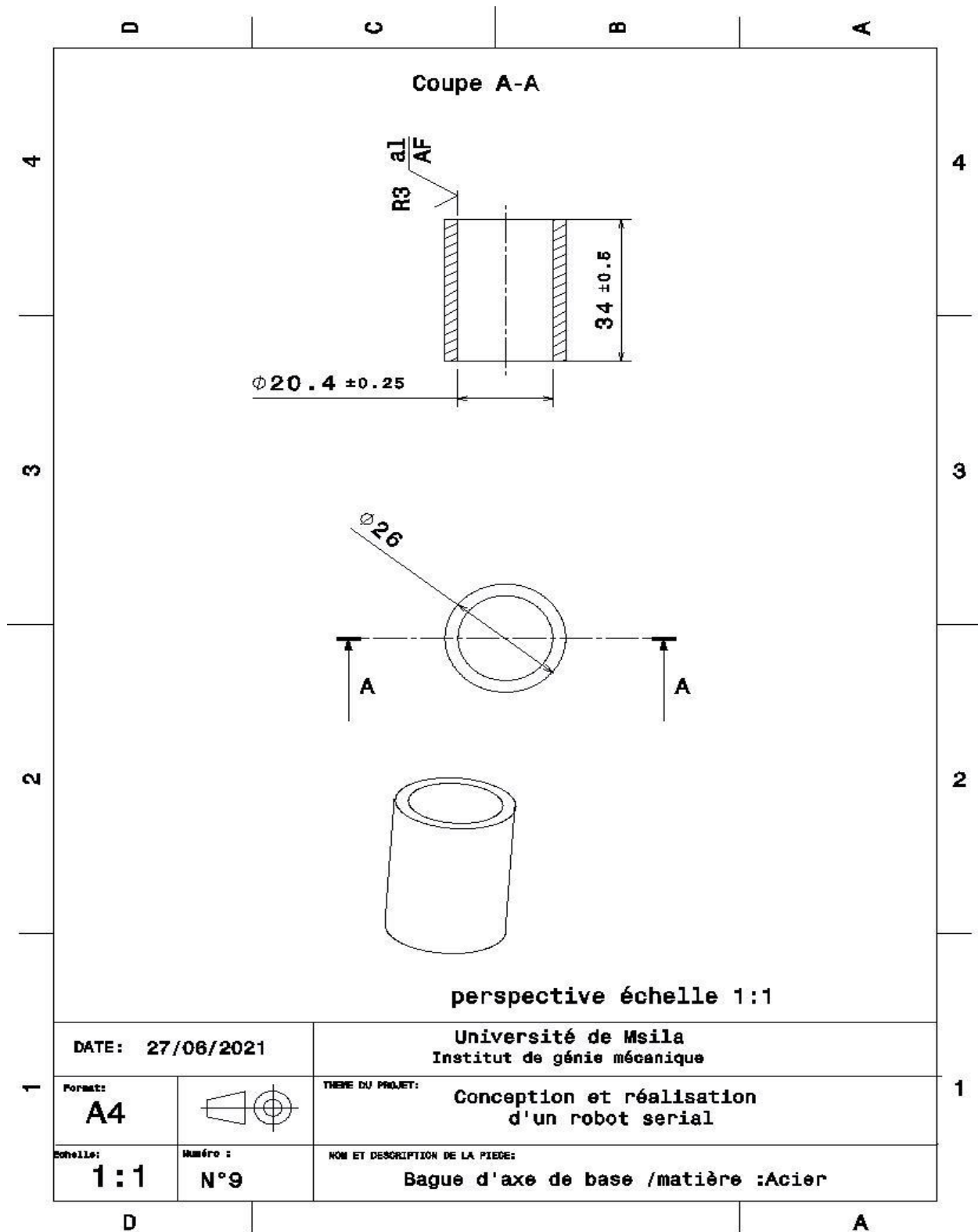
III.4.2.1.5 : Conception et dimensionnement de la pièce n°9: Il y a trois étapes pour guider à la fin de la réalisation pour fabriquer la pièce.

✓ **.A: Conception CAO de la pièce n°9 :**



Vue 3D de la conception CAO de la pièce N°9 :(Bague d'axe de la base) (**Figure III.15**).

✓ .B: Dessin définition la pièce n°1 de la pièce n°9:



✓ .C: Conception réelle et fabrication de la pièce n°9:

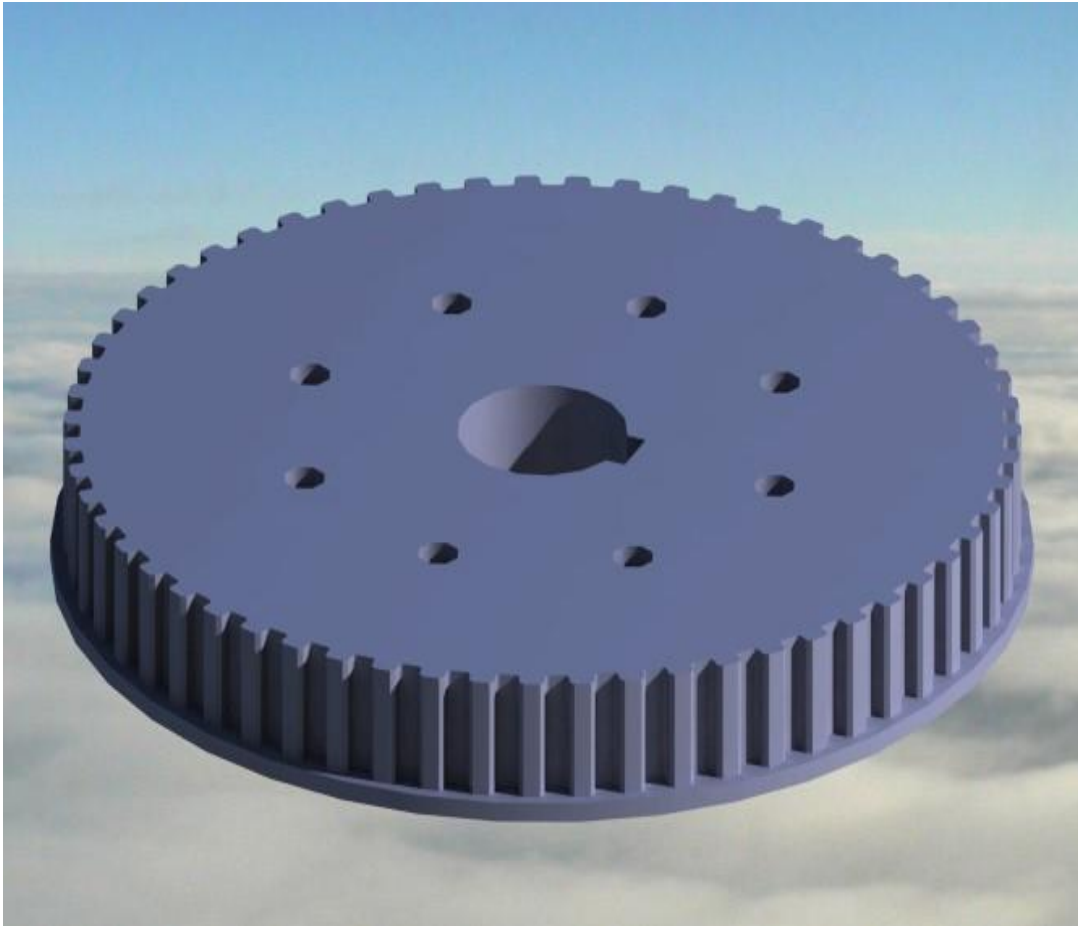


Image de la conception réelle de la pièce N°9 : (Bague d'axe de la base) (**Figure III.16**).

Remarque : Les spécifications techniques des pièces n°10, n°11, n°12, n°13 n°14, n°15 se sont préfabriquées dans tableau des nomenclatures et description pièces préfabriquées à acheter ci-dessous à la fin du chapitre III.

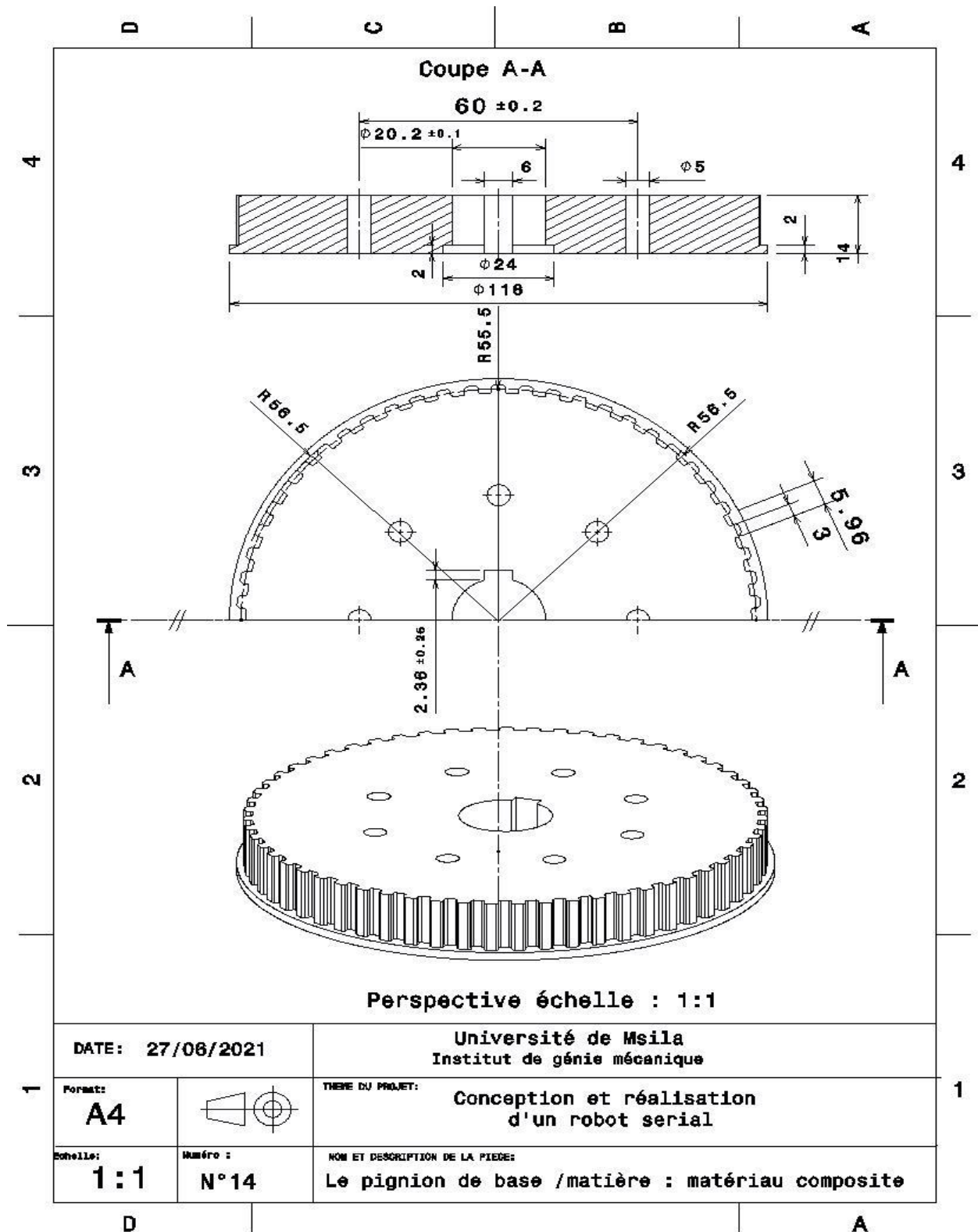
III.4.2.1. 6 : Conception et dimensionnement de la pièce n°16: Il y a trois étapes pour guider à la fin de la réalisation pour fabriquer la pièce.

✓ **.A: Conception CAO de la pièce n°16**



Vue 3D de la conception CAO de la pièce N°16 :(Pignon de base) (**Figure III.17**).

✓ .B: Dessin définition la pièce n°1 de la pièce n°16:



- ✓ .C: **Conception réelle et fabrication de la pièce n°16:** Il y a trois étapes pour guider à la fin de la réalisation pour fabriquer la pièce.



Image de la conception réelle de la pièce N°16 :

(Pignon de base) (**Figure III.18**).

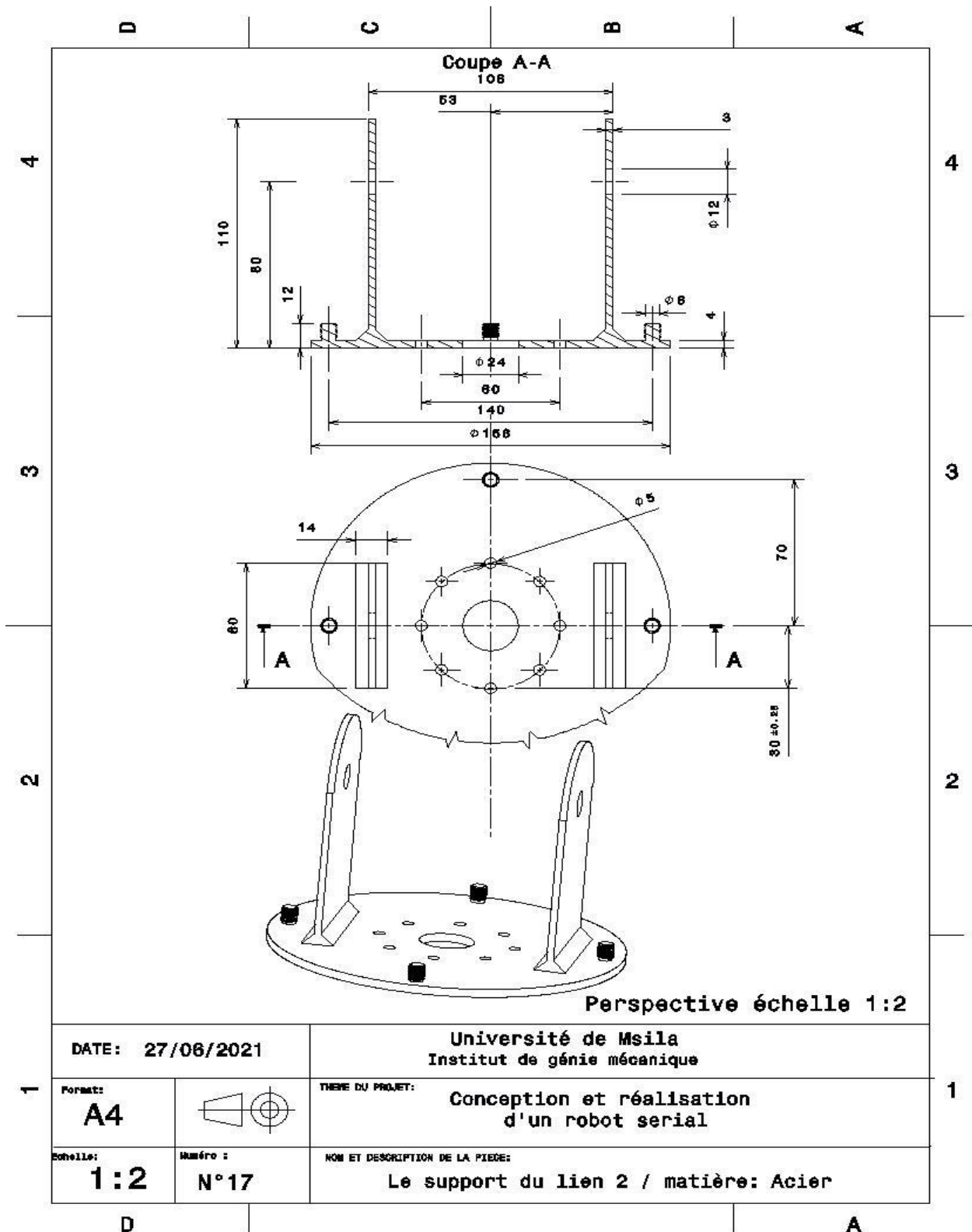
III.4.2.1.7 : Conception et dimensionnement de la pièce n°17:

✓ .A: Conception CAO de la pièce n°17 :



Vue 3D de la conception CAO de la pièce N°17 :(Support de lien 2) (**Figure III.19**).

✓ .B: Dessin définition la pièce n°1 de la pièce n°17:



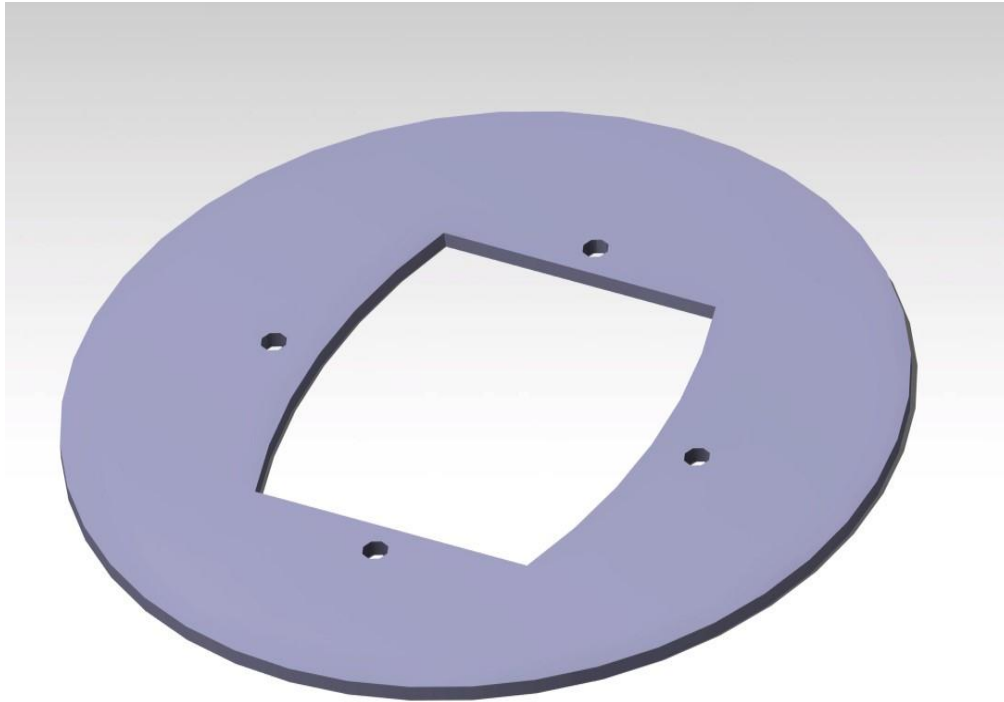
✓ .C: Conception réelle et fabrication de la pièce n°17:



Image de la conception réelle de la pièce N°17 **semi-fini** (Support de lien 2) (**Figure III.20**).

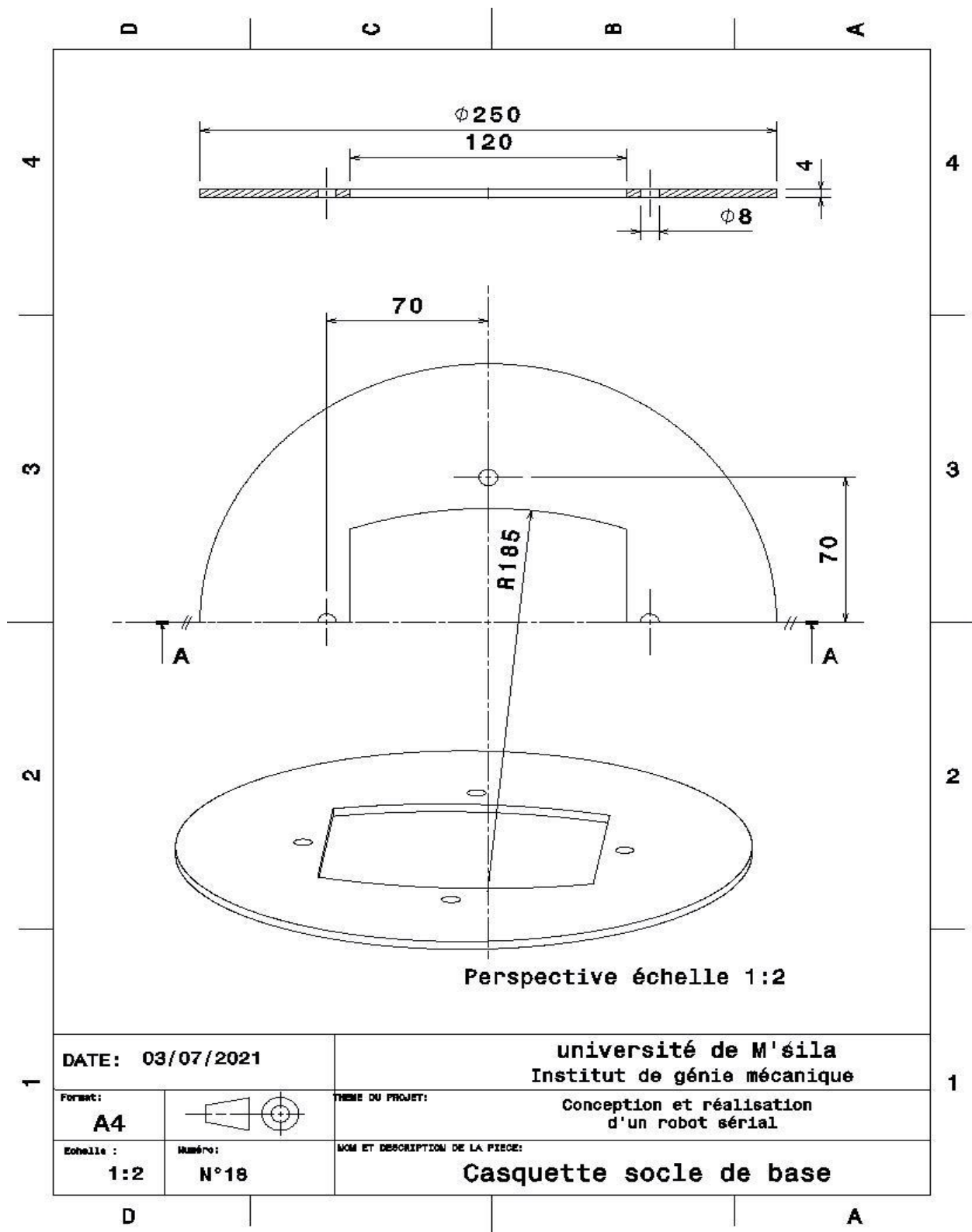
III.4.2.1. 8 : Conception et dimensionnement de la pièce n°18: Il y a trois étapes pour guider à la fin de la réalisation pour fabriquer la pièce.

✓ **.A: Conception CAO de la pièce n°18 :**



Vue 3D de la conception CAO de la pièce N°5 :(Casquette socle de base) (**Figure III.21**).

III.4.2.1. 8 .B: Dessin définition de la pièce n°18:



✓ .C: Conception réel et fabrication de la pièce n°18:

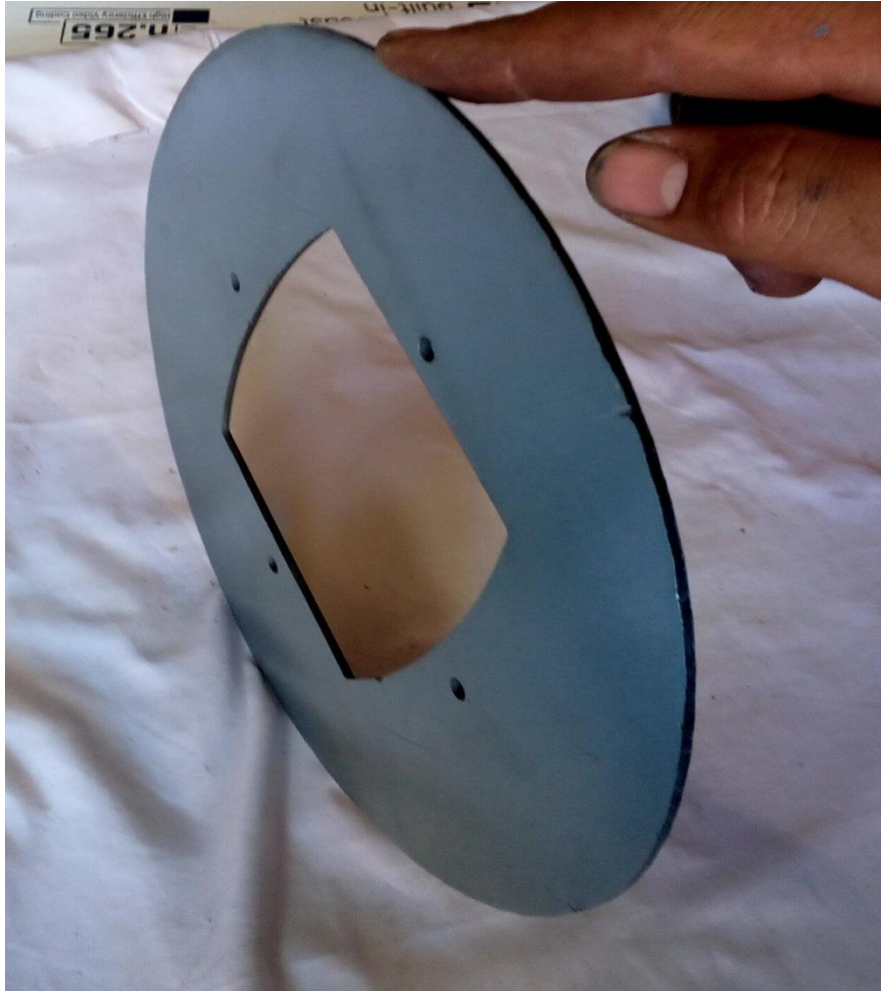


Image de la conception réelle de la pièce N°18 :(Casquette socle de base) (**Figure III.22**).

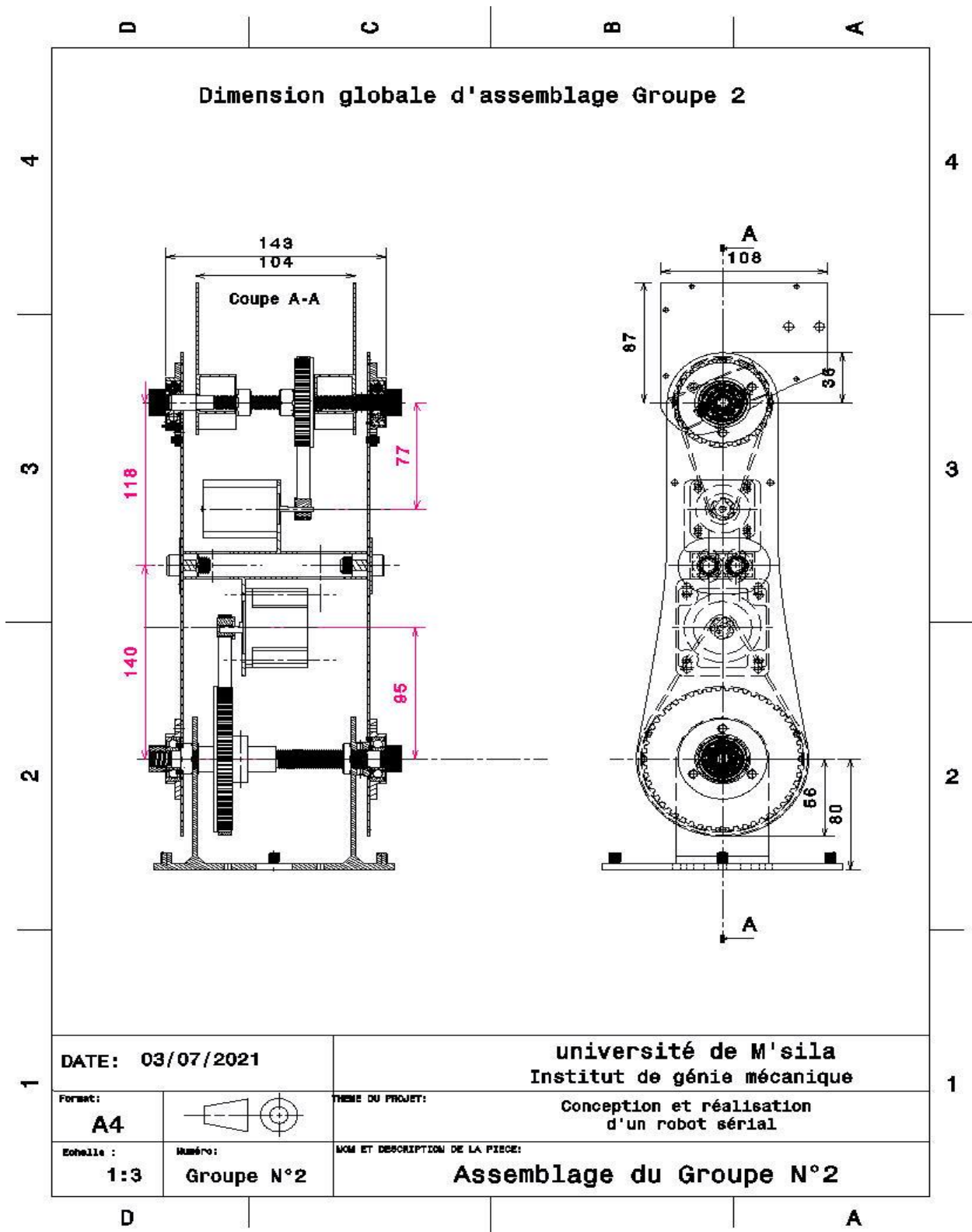
III.4.2.2. Le dimensionnement et l'assemblage du groupe2 du robot: Dans cette étape, nous allons créer la forme et la composition générale des différentes pièces à travers lesquelles un processus de dessin peut être effectué pour déterminer les dimensions et un dessin définition des pièces et des groupes assemblés.

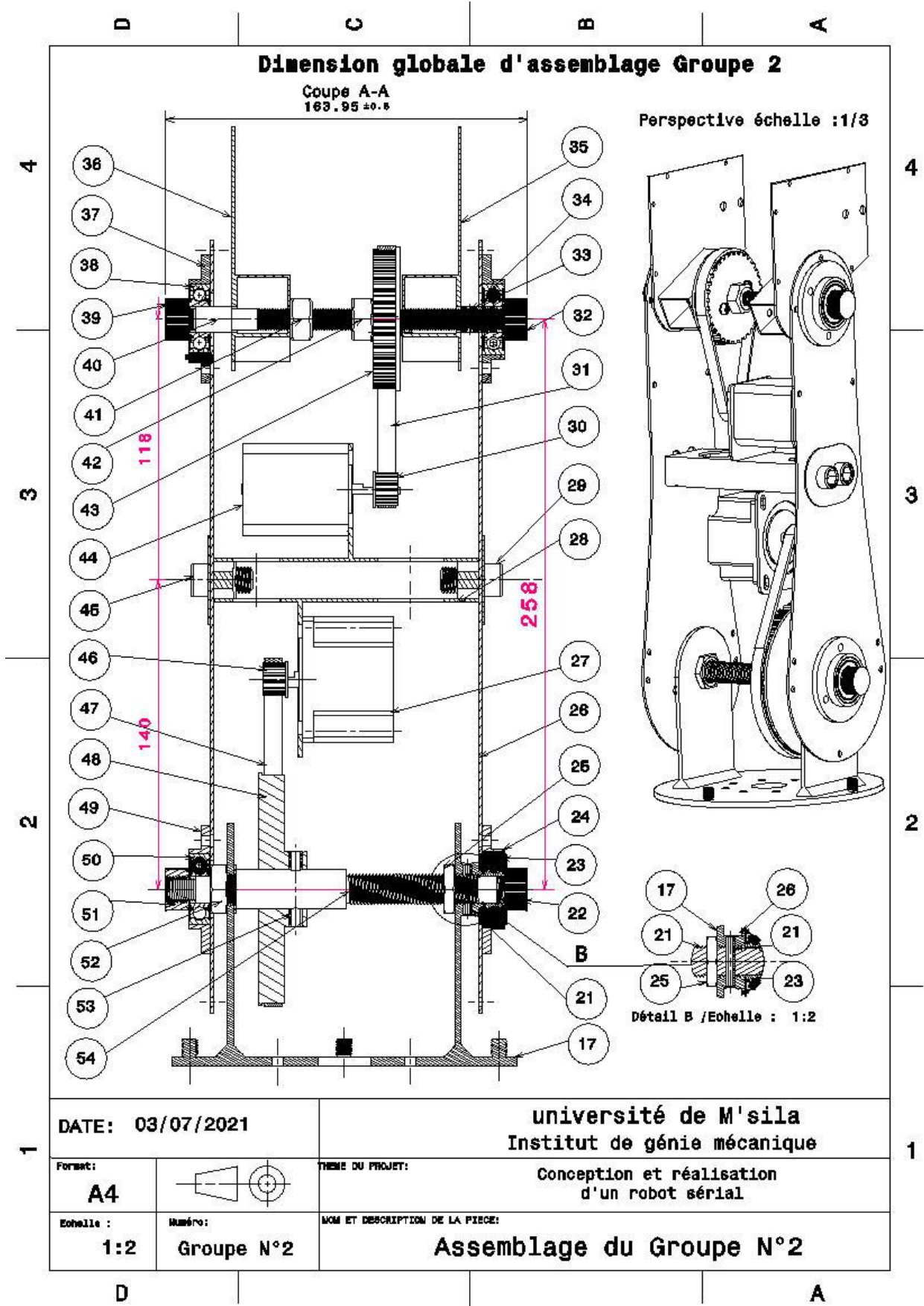
✓ **.A: Conception CAO du groupe 2:**



Vue 3D de la conception CAO du groupe2 :(Le segment-bras du robot) (**Figure III.23**).

✓ .B: Dessin d'assemblage et dimensionnement du groupe 2:





✓ .C: Conception réel et fabrication du groupe 2:

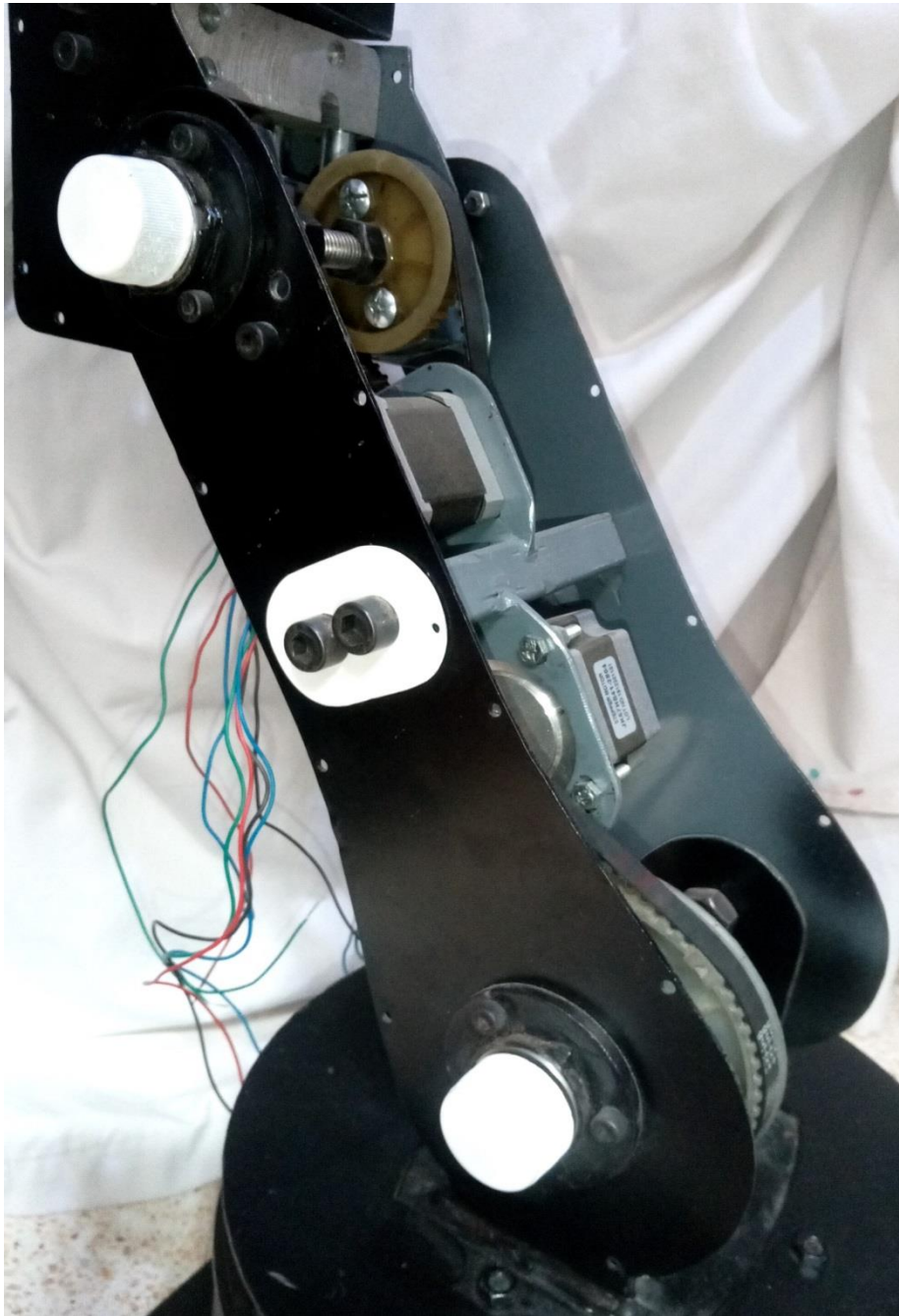
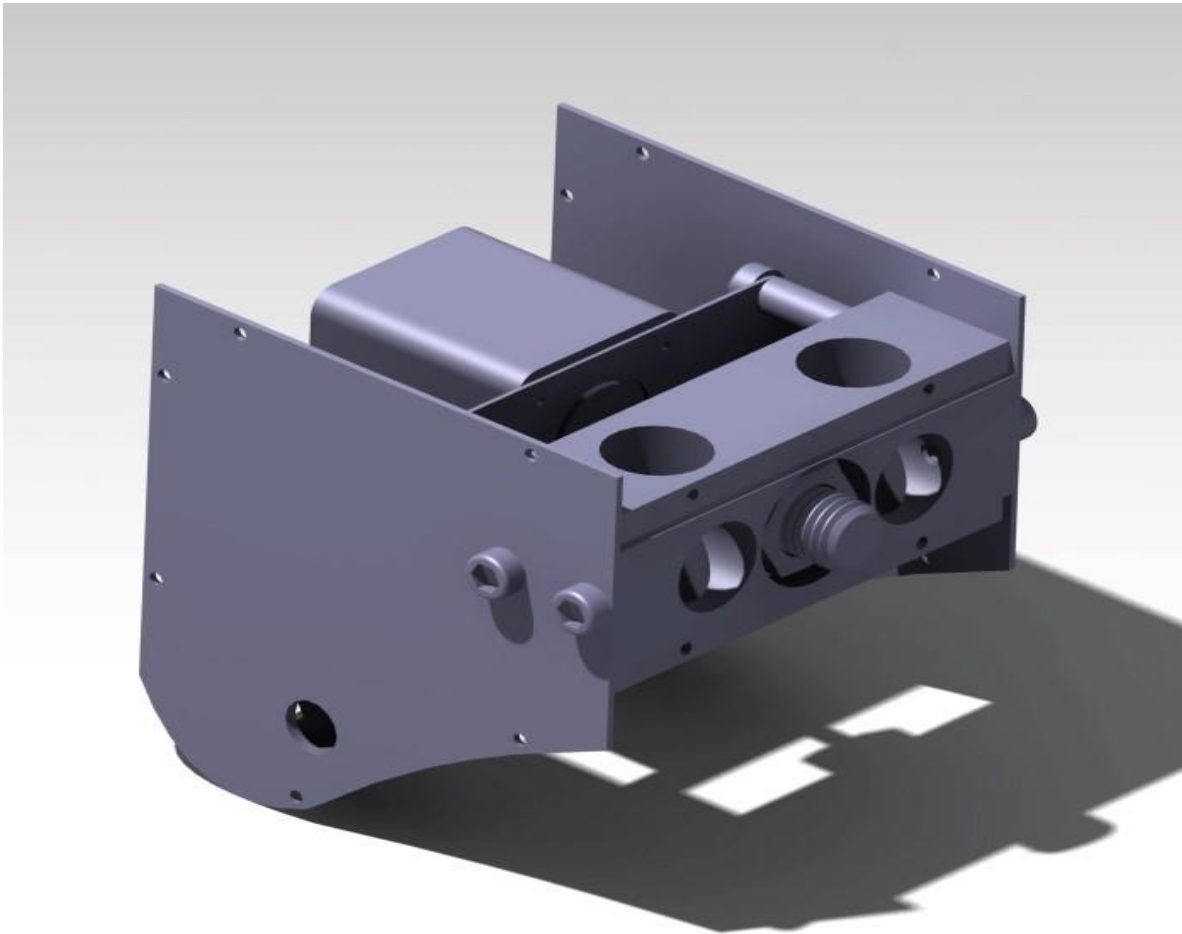


Image de la conception réelle du groupe 2 : (Le segment-bras du robot) (**Figure III.24**)

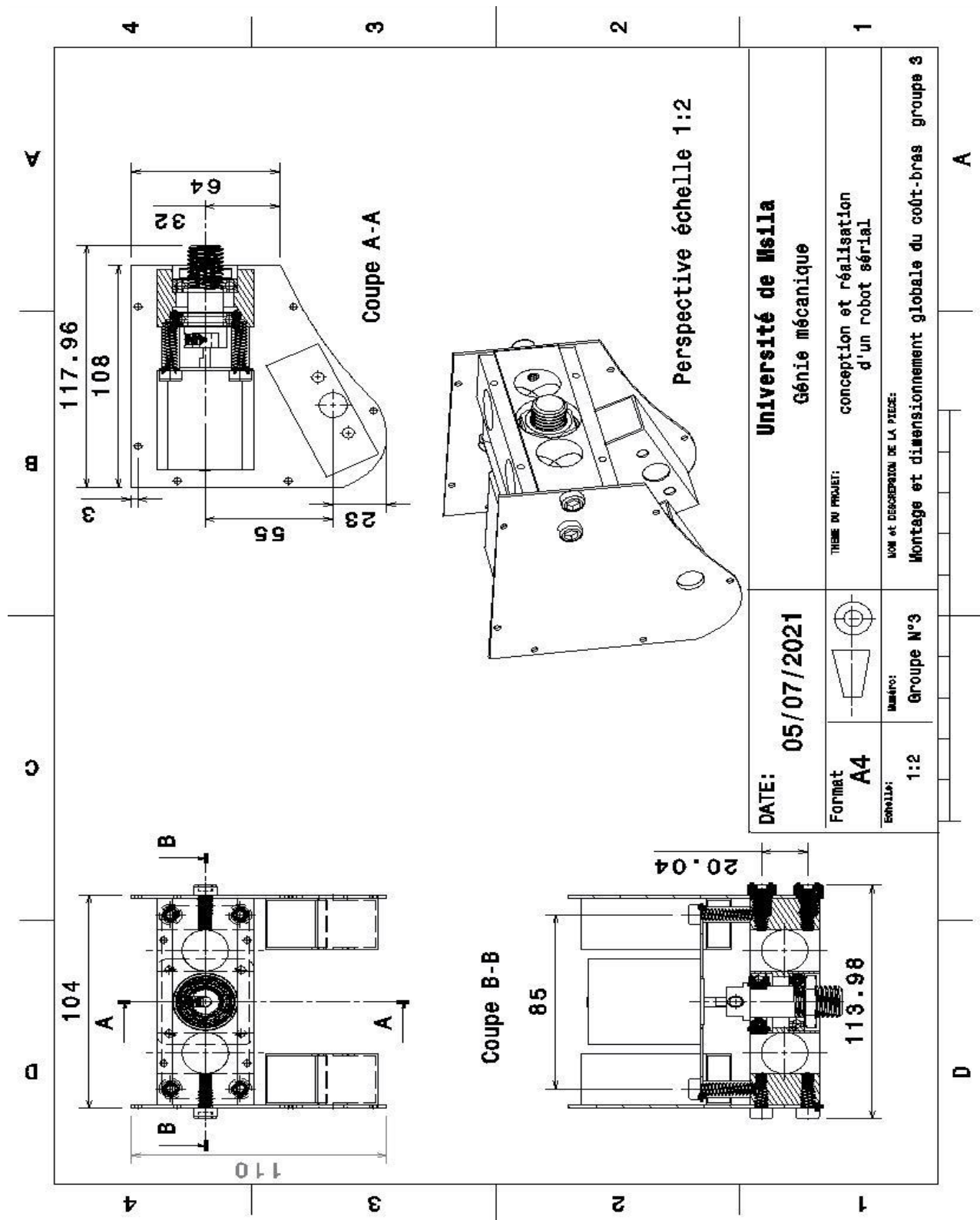
III.4.2.3. Le dimensionnement et l'assemblage du groupe3 du robot: Dans cette étape, nous allons créer la forme et la composition générale des différentes pièces à travers lesquelles un processus de dessin peut être effectué pour déterminer les dimensions et un dessin définition des pièces et des groupes assemblés.

✓ **.A: Conception CAO du groupe 3:**



Vue 3D de la conception CAO du groupe3 : (coude de l'avant-bras du robot) (**Figure III.25**).

✓ .B: Dessin d'assemblage et dimensionnement du groupe 3:



✓ .C: Conception réelle et fabrication du groupe 3:

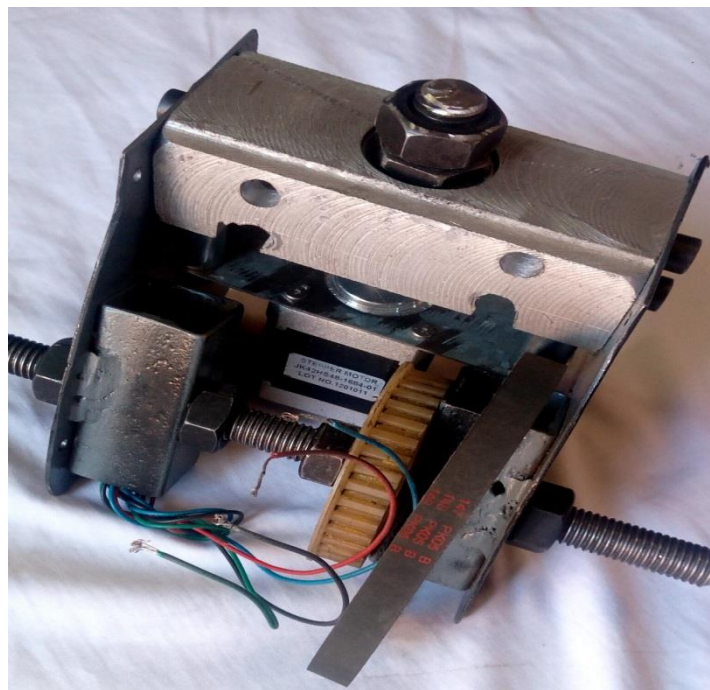
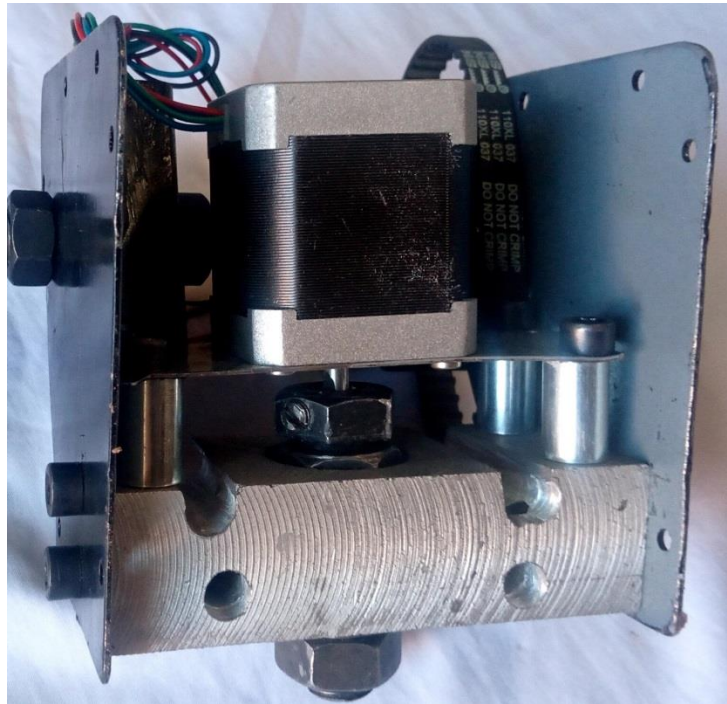
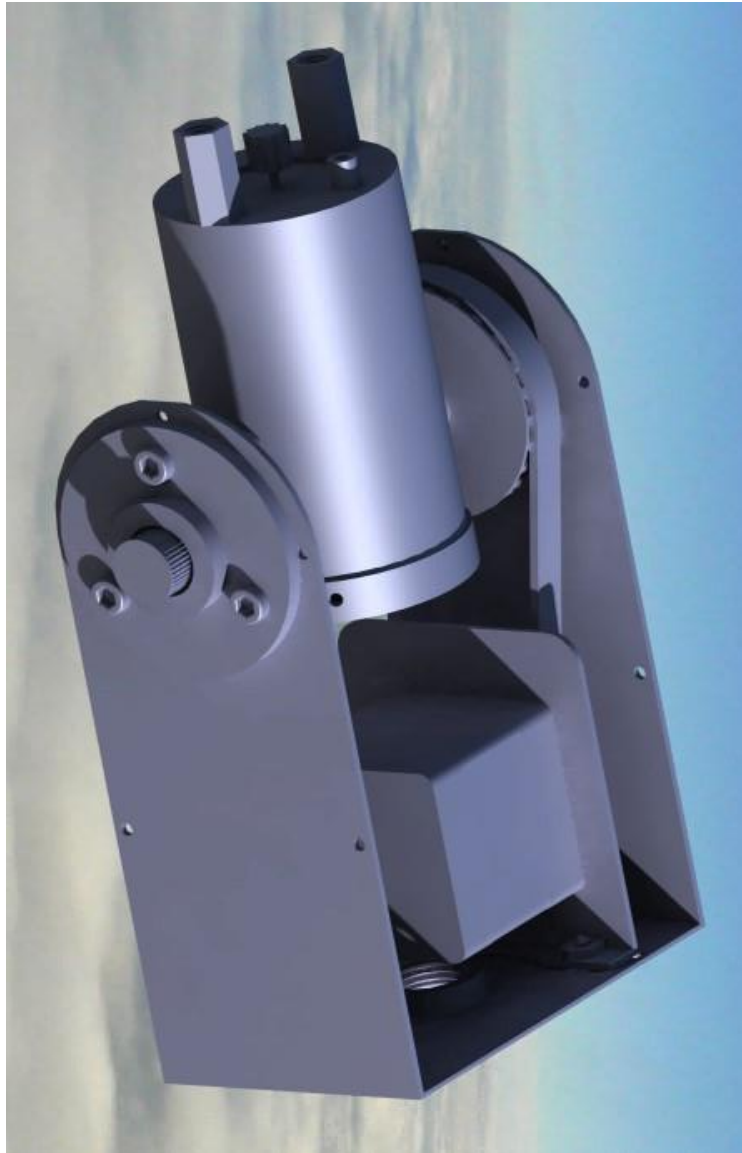


Image de la conception réelle du groupe 3 :(Coude de l'avant-bras du robot) (**Figure III.26**).

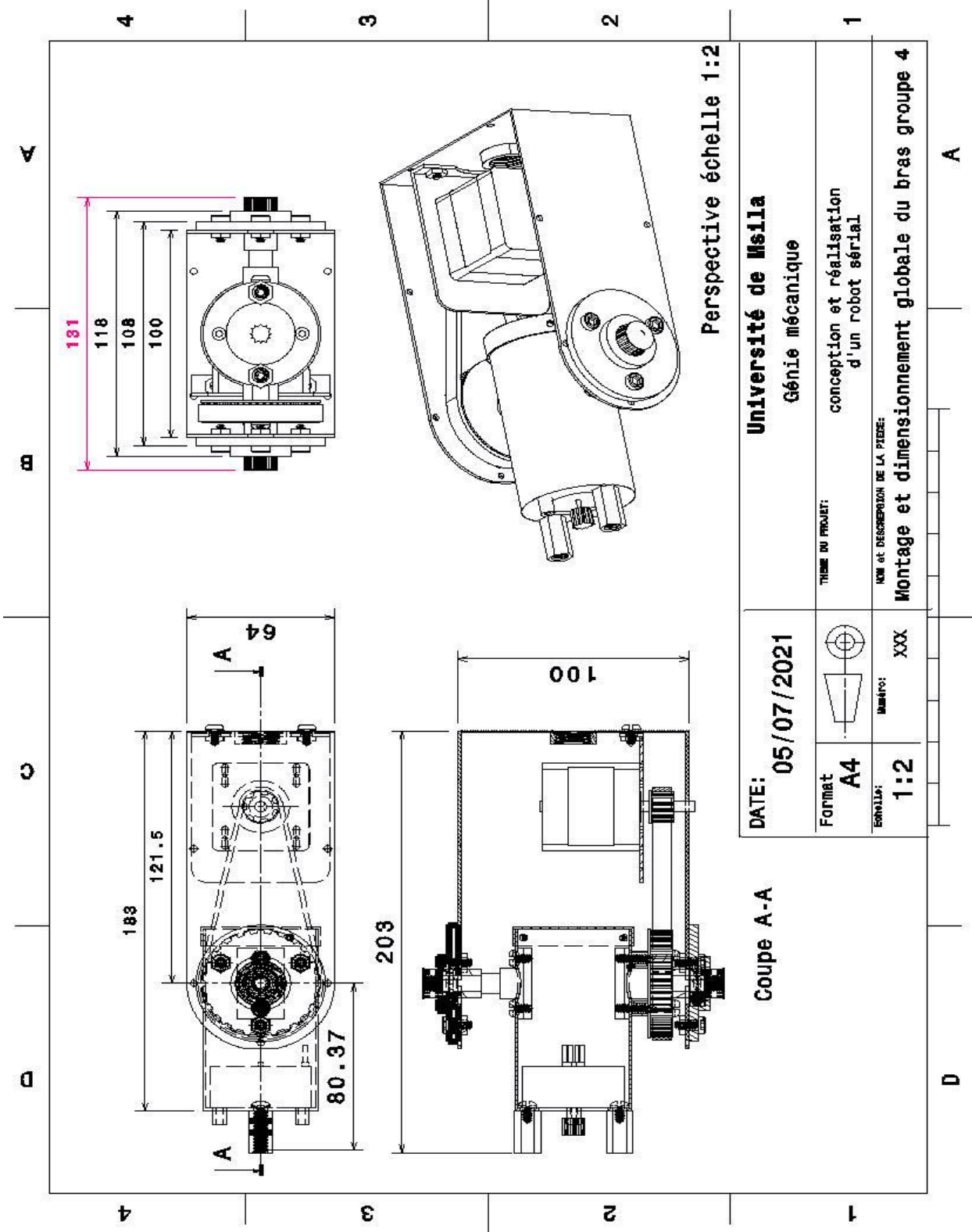
III.4.2.4. Le dimensionnement et l'assemblage du groupe 4 du robot: Dans cette étape, nous allons créer la forme et la composition générale des différentes pièces à travers lesquelles un processus de dessin peut être effectué pour déterminer les dimensions et un dessin définition des pièces et des groupes assemblés.

✓ **.A: Conception CAO du groupe 4:**



Vue 3D de la conception CAO du groupe4 :(L'avant-bras du robot) (**Figure III.27**).

✓ .B: Dessin d'assemblage et dimensionnement du groupe 4:



✓ .C: Conception réelle et fabrication du groupe 4:

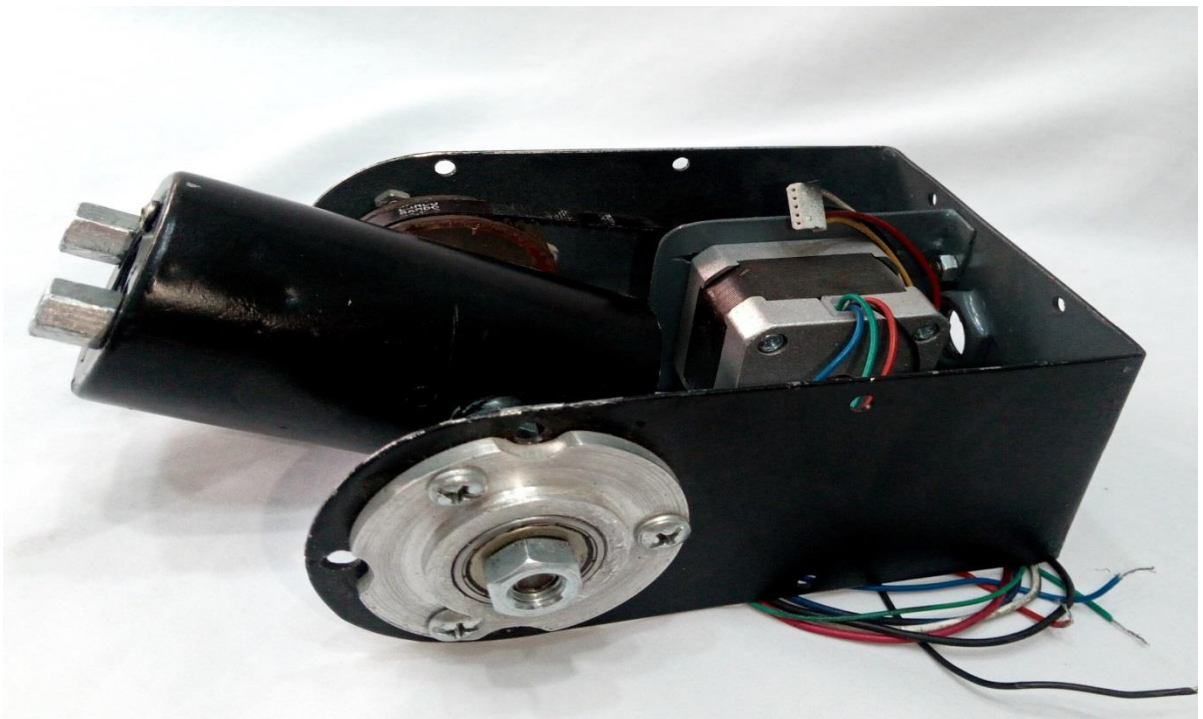
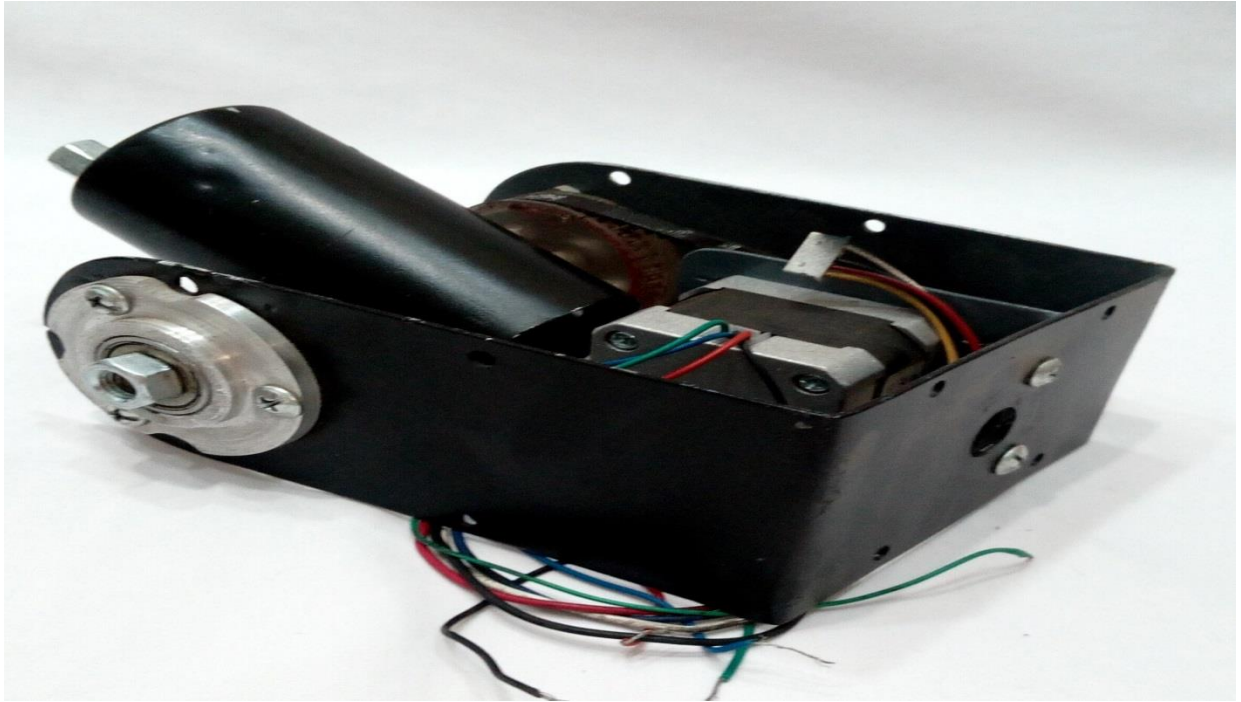


Image de la conception réelle du groupe 4 : (L'avant-bras du robot) (**Figure III.28**).

III.4..3: Spécifications techniques des pièces préfabriquées à acheter : Il existe des pièces prêtes à l'emploi et fabriquées qui sont vendues sur les marchés, leur achat est donc plus économique et permet d'économiser de l'argent, des efforts et de la force, et elles sont répertoriées dans le tableau suivant: Tableau des nomenclatures et description pièces préfabriquées

N° de la pièce dans l'assemblage	Noms des pièces	Dimensions	Spécifications techniques	Description et autres détails
Assemblage groupe1 pièces : N°3 et N°4	Roulement BC	d = 20 mm D = 47 mm B = 14 mm $d_2 \approx 26.3$ mm $D_2 \approx 40.6$ mm $r_{1,2}$ min. 1 mm		
Assemblage groupe1 pièces : N°6	circlips intérieur	d=70mm b≈ 6mm a≈8mm e≈2mm		
Assemblage groupe1 pièces : N°8	Écrou hexagonal	S = 30 mm e= 32.95mm M=M20 m=10mm		
Assemblage groupe1 pièces : N°10	Moteur pas à pas Model 1 Model NO. JK57HS41-2804	Dimensions: (Unit:mm) 		Current /Phase 2.8A Holding Torque 0.55n.M Motor Length 41mm Step Angle 1.8 Degree
Assemblage groupe1 pièces : N°11	Boulon et écrou hexagonal (M6)	L > 12 ≤14 D= 6 mm(M6) H=3-4 mm F=10mm		

III.5. Le prototype final du robot après réalisation :

On a nommé ce robot **UNIVERSEL 100DZ21**

Où :

100 : Version 100.

DZ : Origine paye Algérie.

21 : l'année de fabrication

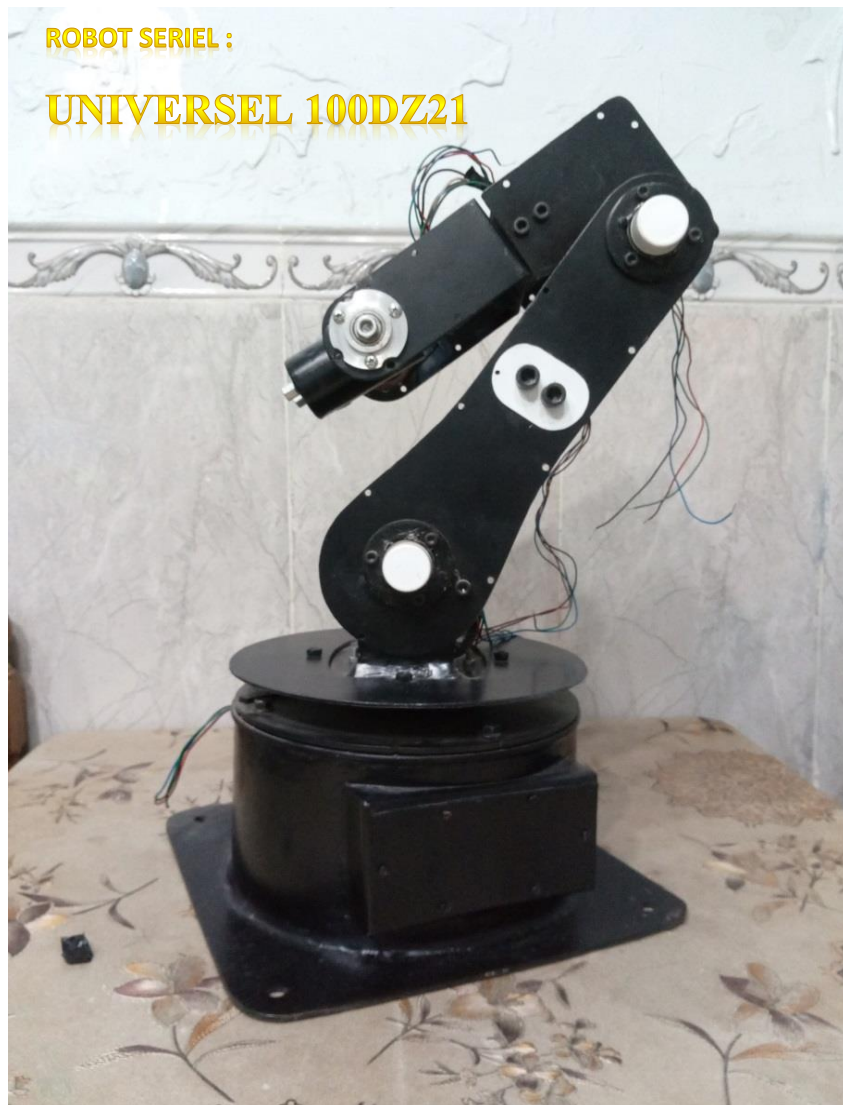


Figure III.29: Le prototype final UNIVERSEL 100DZ21

Excuse et suggestion

Nous nous excusons pour certaines lacunes qui imprègnent ce chapitre Et certain dessin définition des pièces qui ne sont pas disponibles pour les raisons suivantes:

- Pas assez de temps.
- Manque de soutien financier.
- Manque de moyens et de matières premières.
- La situation particulière et la situation exceptionnelle due à l'épidémie de Coronavirus.
- Dispersion de membre d'équipe de travail, manque de réunions... et autres raisons.

Par conséquent, nous espérons que vous apprécierez notre effort, qui a duré plus de 4 mois de travail continu, et nous espérons continuer et améliorer notre projet à l'avenir, dans l'espoir d'obtenir motivation et privilège de la part de l'administration universitaire.

CHAPITRE IV : COMMANDE DU ROBOTS SOUS ARDUINO.

IV.1. La Commande du Robot:

IV.1.1. Matériel nécessaire pour la commande :

Pour la réalisation du circuit de commande nous avons utilisé un matériel en dessous :

- Une carte Arduino Mega
- Une platine d'expérimentation (breadboard)
- Drivers) DRV8825)
- HC05 Bluetooth
- Moteurs pas à pas
- Des câbles de liaison (jumpers)
- LEDS
- Résistances de 220 à 470 Ω
- Condensateurs
- Boutons- poussoirs
- Potentiomètre (résistance variable)
- Power supply (un Transformateur)

❖ Carte Arduino MEGA :

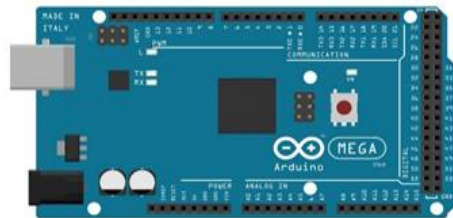


Figure. IV.1 : Carte d'Arduino MEGA.

❖ La platine d'expérimentation:

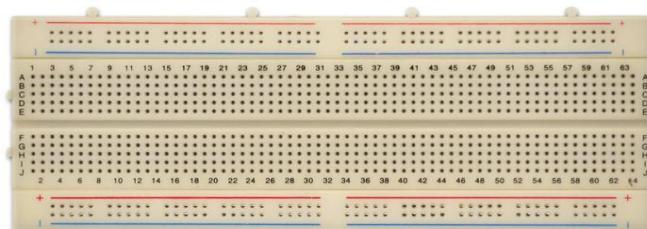


Figure. IV.2 : Platine d'expérimentation (breadboard).

❖ Les composants électroniques :

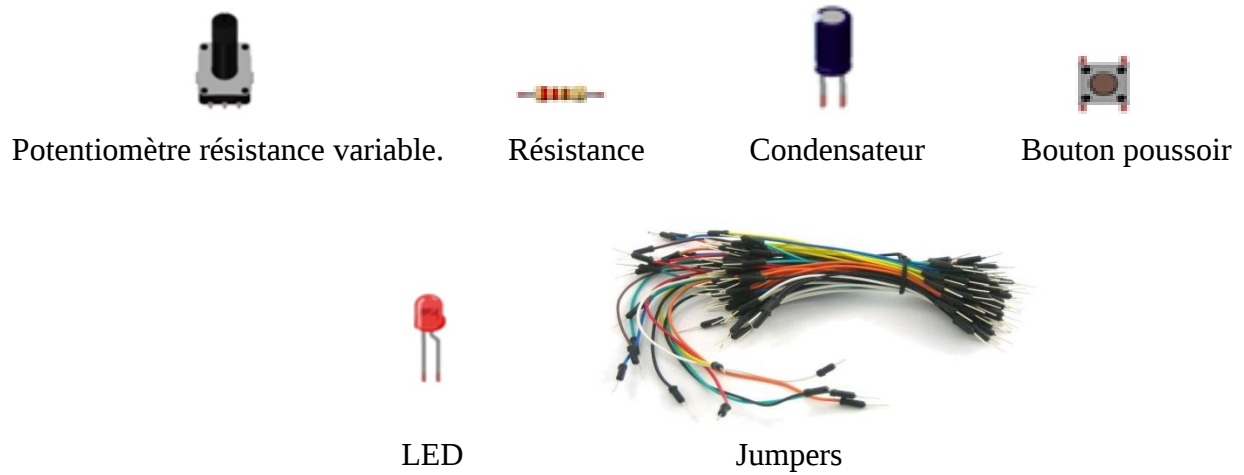


Figure. IV.3: Eléments électroniques utilisés.

❖ Les éléments d'alimentation et de pilotage :

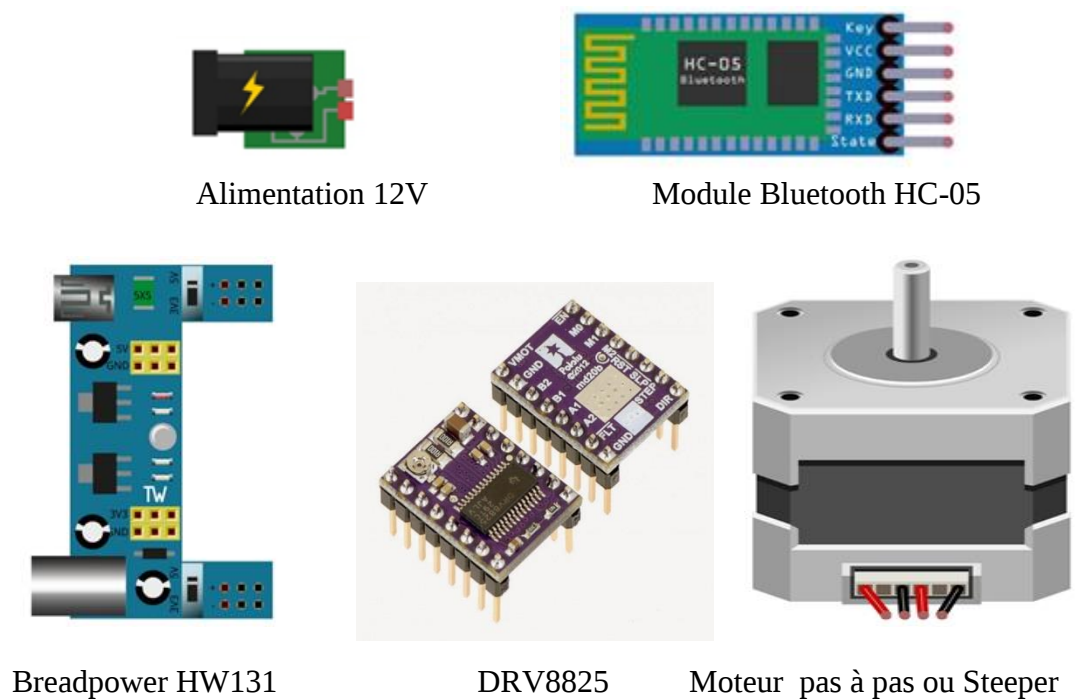


Figure. IV.4 : Les éléments d'alimentation et de pilotage :

IV.1.2. Description et caractéristiques techniques:

IV.1.2.1. La carte platine Arduino Mega:

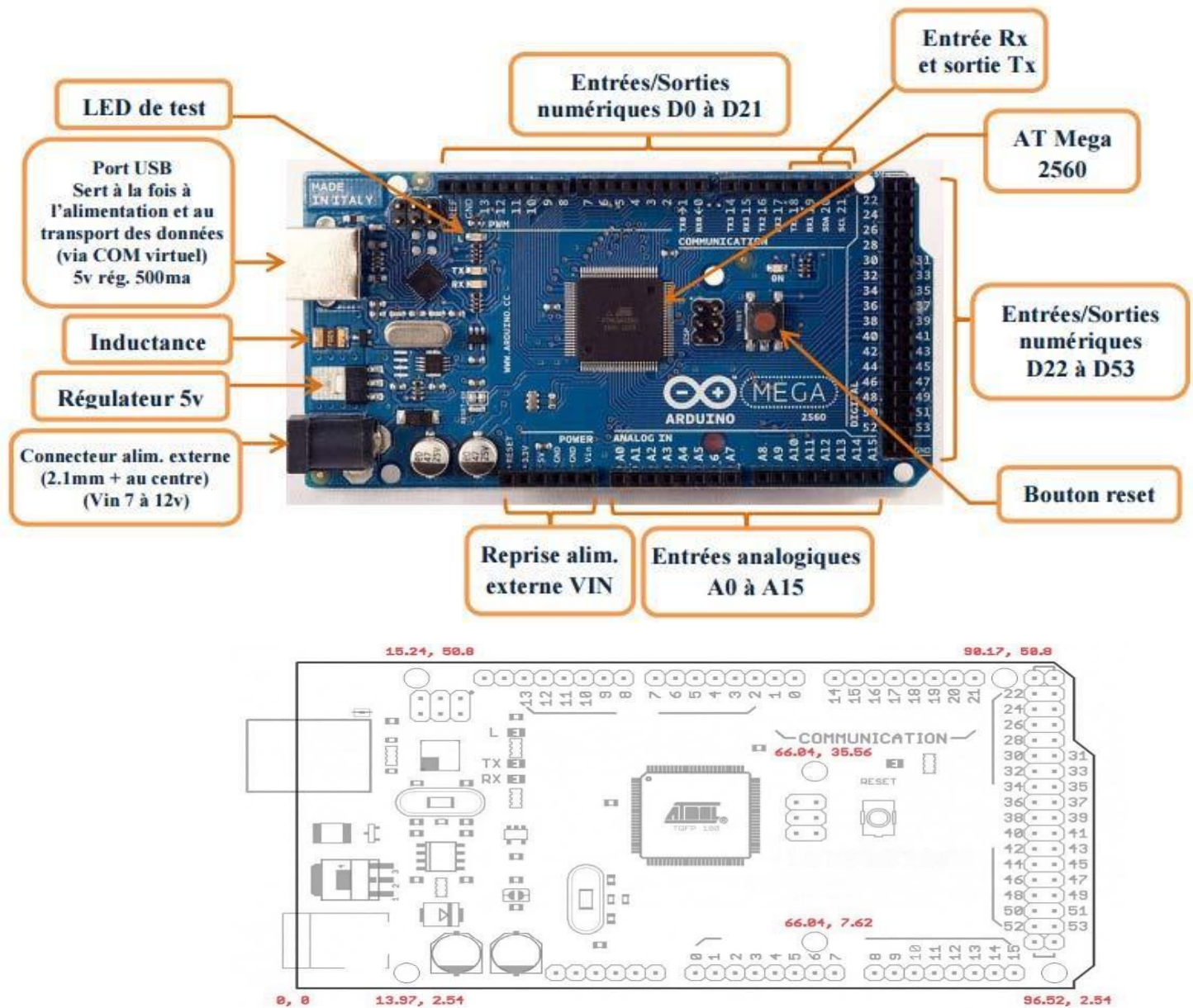


Figure IV.5 : Description de la carte Arduino MEGA 2560[10]

La carte Arduino Mega 2560 est une carte à microcontrôleur basée sur un ATmega2560. Cette Carte dispose :

- de 54 broches numériques d'entrées/sorties (dont 14 peuvent être utilisées en sorties. PWM (MLI : Modulation de largeur d'impulsion).
- de 16 entrées analogiques (qui peuvent être utilisées en broches entrées/sorties Numériques).
- de 4 UART (port série matériel).
- d'un quartz de 16Mhz.
- d'une connexion USB.
- d'un connecteur d'alimentation jack.
- d'un connecteur ICSP (programmation "in-circuit").
- d'un bouton de réinitialisation (reset).

Elle contient tout ce qui est nécessaire pour le fonctionnement du microcontrôleur ; Pour Pouvoir l'utiliser, il suffit simplement de la connecter à un ordinateur à l'aide d'un câble USB (Ou de l'alimenter avec un adaptateur secteur ou une pile, mais ceci n'est pas indispensable, l'alimentation étant fournie par le port USB). La carte Arduino Mega 2560 est compatible avec les circuits imprimés prévus pour les cartes Arduino Uno, Duemilanove ou Diecimila [10].

Partie matérielle Généralement tout module électronique qui possède une interface de programmation est basé toujours dans sa construction sur un circuit programmable ou plus [10].

Microcontrôleur	ATMEGA2560
Tension de fonctionnement	5V
Tension d'alimentation	7 à 12V
Broches E/S numérique	54 (dont 14 disposent de sortie PWM)
Broches d'entrées analogiques	16
Vitesse d'horloge	16 MHz
Mémoire programme Flash	25 6KB dont 8 KB utilisés en bootloader
Mémoire SRAM	8 KB
Mémoire EEPROM	4 KB

Tableau IV.1 : Constitution de la carte Arduino Mega 2560[10]

❖ Les sources de l'alimentation de la carte:

La carte Arduino Mega 2560 peut être alimentée soit via la connexion USB (qui fournit

5V jusqu'à 500mA) ou à l'aide d'une alimentation externe. La source d'alimentation est Sélectionnée automatiquement par la carte. [10]

La carte peut fonctionner avec une alimentation externe de 6 à 20 volts. Cependant, si la carte est alimentée avec moins de 7V, la broche 5V pourrait fournir moins de 5V, et la carte pourrait être instable. Si on utilise plus de 12V, le régulateur de tension de la carte pourrait chauffer ce dernier. Les broches d'alimentation sont les suivantes : [10]

- **VIN** : La tension d'entrée positive lorsque la carte Arduino est utilisée avec une source de tension externe (à distinguer du 5V de la connexion USB ou autre source 5V régulée).
- **5V** : La tension régulée utilisée pour faire fonctionner le microcontrôleur et les autres composants de la carte. Le 5V régulé fourni par cette broche peut donc provenir soit de la tension d'alimentation VIN via le régulateur de la carte, ou bien de la connexion USB ou de tout autre source d'alimentation régulée.
- **3.3V** : Une alimentation de 3.3V fournie par le circuit intégré FTDI (circuit intégré faisant l'adaptation du signal entre le port USB de votre ordinateur et le port série de l'ATmega) de la carte. L'intensité maximale disponible sur cette broche est de 50Ma.
- **GND** : Broche de masse (0V).

❖ Entrées et sorties numériques:

Chacune des 54 broches numériques de la carte Mega peut être utilisée soit comme une entrée numérique, soit comme une sortie numérique. Ces broches fonctionnent en 5V. Chaque broche peut fournir ou recevoir un maximum de 40mA d'intensité et dispose d'une résistance interne de "rappel au plus" (pull-up) (déconnectée par défaut) de 20-50 KOhms. Cette résistance interne s'active sur une broche en entrée à l'aide de l'instruction digitalWrite (broche, HIGH).

De plus, certaines broches ont des fonctions spécialisées : [10]

- **Communication Série:** Port Série Serial : 0 (RX) et 1 (TX); Port Série Serial 1: 19 (RX) and 18 (TX); Port Série Serial 2: 17 (RX) and 16 (TX); Port Série Serial 3: 15 (RX) and 14 (TX).

- **Interruptions Externes:** Broches 2 (interrupt 0), 3 (interrupt 1), 18 (interrupt 5), 19 (interrupt 4).
- **20 (interrupt 3), et 21 (interrupt 2).** Ces broches peuvent être configurées pour déclencher une interruption sur une valeur basse, sur un front montant ou descendant, ou sur un changement de valeur.
- **Impulsion PWM (largeur d'impulsion modulée):** Broches 0 à 13. Fournissent une impulsion PWM 8-bits à l'aide de l'instruction `analogWrite ()`.
- **SPI (Interface Série Périphérique):** Broches 50 (MISO), 51 (MOSI), 52 (SCK), 53 (SS). Ces broches supportent la communication SPI (Interface Série Périphérique).
- **I2C: Broches 20 (SDA) et 21 (SCL).** Supportent les communications de protocole I2C.
- **LED: Broche 13.** Il y a une LED incluse dans la carte connectée à la broche 13.

❖ Broches analogiques:

La carte Mega2560 dispose de 16 entrées analogiques, chacune pouvant fournir une mesure d'une résolution de 10 bits (1024 niveaux soit de 0 à 1023) à l'aide de la très utile fonction `analogRead ()` du langage Arduino. Par défaut, ces broches mesurent entre le 0V (valeur 0) et le 5V (valeur 1023). Les broches analogiques peuvent être utilisées en tant que broches numériques. [10]

❖ Autres broches:

Il y a deux autres broches disponibles sur la carte :

- **AREF :** Tension de référence pour les entrées analogiques (si différent du 5V). Utilisée avec l'instruction `analogReference ()`.
- **Reset :** Mettre cette broche au niveau BAS entraîne la réinitialisation du microcontrôleur. Comme un port de communication virtuel pour le logiciel sur l'ordinateur, La connexion série de l'Arduino est très pratique pour communiquer avec un PC, mais son inconvénient est le câble USB, pour éviter cela, il existe différentes méthodes pour utiliser ce dernier sans fil. [10]

❖ Le microcontrôleur:

Un microcontrôleur ATmega2560 est un circuit intégré qui rassemble sur une puce plusieurs éléments complexes dans un espace réduit, c'est le processeur de la carte, s'occupe de tout ce qui est calculs, exécution des instructions du programme et gestion des ports d'entrée/sortie [11].



Figure IV.6 : Microcontrôleur ATmega2560 [10]

IV.1.2.2. Le contrôle de moteur pas-à-pas DRV8825 :

C'est un breakout de contrôle moteur pas-à-pas. le DRV8825 est basé sur le DRV8825 de Texas Instrument pouvant piloter un moteur pas-à-pas en micro-stepping (micro pas) jusqu'au 1/32 pas. Le DRV8825 est surtout connu pour pouvoir contrôler des moteurs plus puissants. Il intègre un limiteur de courant actif et réglable. Le DRV8825 dispose également d'une protection contre les sur courants, la surchauffe avec 6 résolutions de micro-stepping (jusqu'au 1/32 de pas). Il fonctionne entre 8.2 et 45 V et sais délivrer apprimativement 1.5 A par phase sans refroidissement (par refroidisseur ou air forcé - conçu pour 2.2A par bobine en utilisant un système de refroidissement efficace/adéquat).

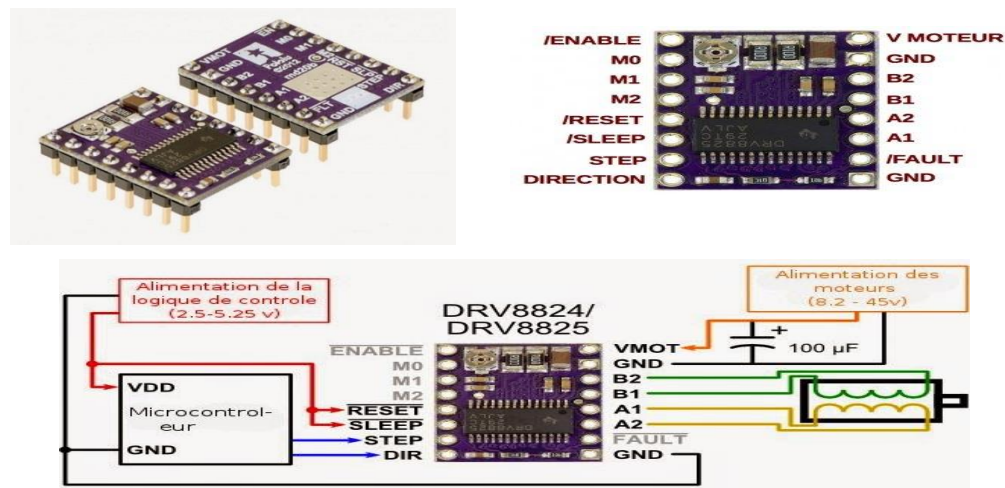


Figure. IV.7 : Le contrôle de moteur pas-à-pas DRV8825 et son schéma de montage sous Arduino

IV.1.2.3. La platine d'expérimentation (breadboard) :

Une platine d'expérimentation (appelée breadboard) permet de réaliser des prototypes de montages électroniques sans soudure et donc permet de pouvoir réutiliser les composants.

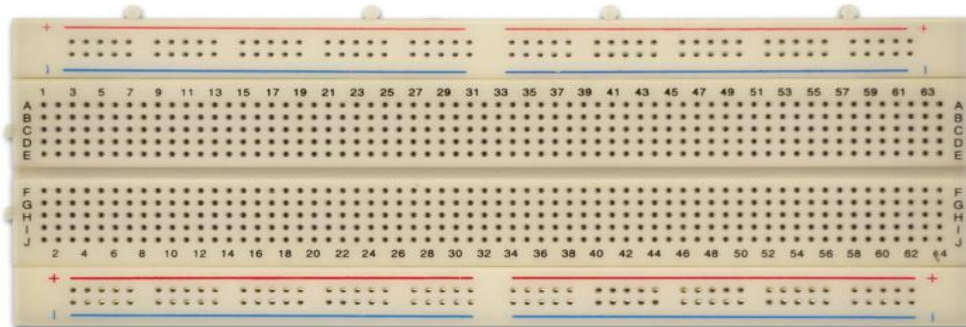


Figure IV.8 : Une platine d'expérimentation (breadboard).

Tous les connecteurs dans une rangée de 5 sont reliés entre eux. Si l'on branche deux éléments dans un groupe de cinq connecteurs, ils seront reliés entre eux. Il en est de même des alignements de connecteurs rouges (pour l'alimentation) et bleus (pour la terre). Ainsi, les liens peuvent être schématisés ainsi:

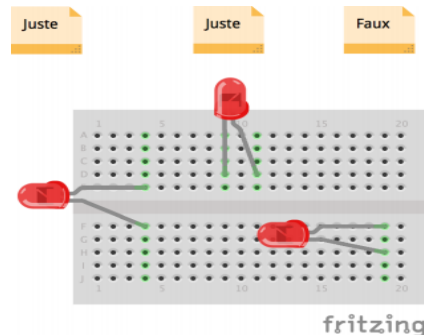
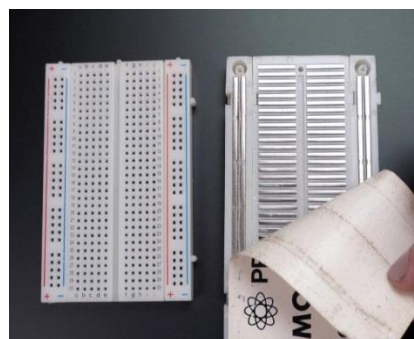
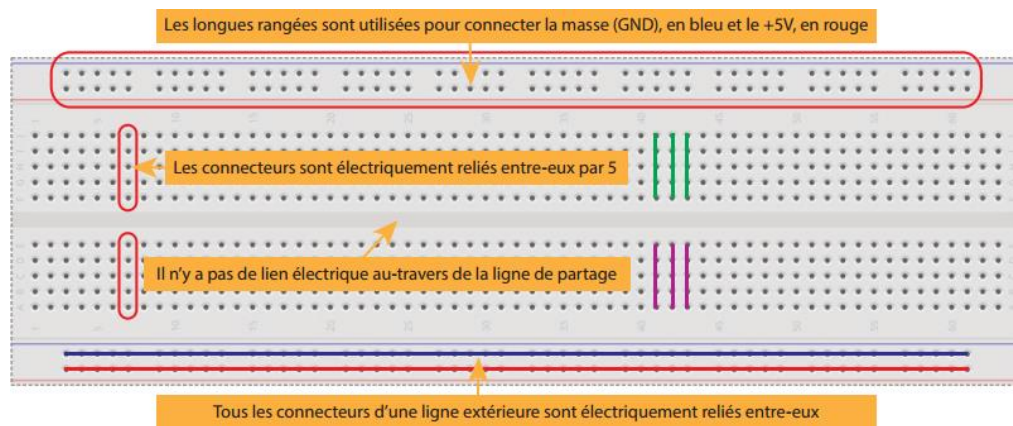


Figure. IV.9 : Le branchement sur un breadboard

Les composants doivent ainsi être placés à cheval sur des connecteurs qui n'ont pas de liens électriques entre eux, comme sur le schéma de la Figure. IV.9 ci-dessus.

IV.1.2.4. Câbles de liaison (jumpers):

Les connexions entre les composants sont réalisées par des jumpers, sortes de petits câbles.

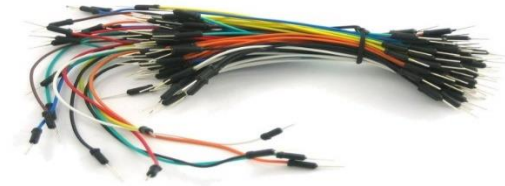


Figure. IV.10 : Câbles de liaison (jumpers)

IV.1.2.5. Les LEDs:

Les LEDs se sont beaucoup améliorées ces dernières années. Elles sont beaucoup plus lumineuses pour le même courant, ce qui veut dire que l'on peut mettre des résistances série plus élevées. Les diodes sont moulées dans un plastique transparent qui fait lentille et converge plus ou moins la lumière; les diodes visibles sont souvent encapsulées dans un plastique de la même couleur, ce qui diffuse mieux de côté. Le choix Bricoshop est « standard » et suffit pour s'amuser avec des Leds. Une LED est polarisée. Le courant entre par la patte longue et sort par la patte plus courte, du côté où le plastique a un replat. La chute de tension est de 1.4 à 1,9V, selon la couleur et le courant. Les Leds bleues et blanches ont une chute de tension de 2 à 3.6V. Il faut une résistance pour limiter le courant, et pour les diodes de 3 et 5mm de diamètre, 20mA est le maximum usuel. La valeur de la résistance dépend de la tension d'alimentation. [12]

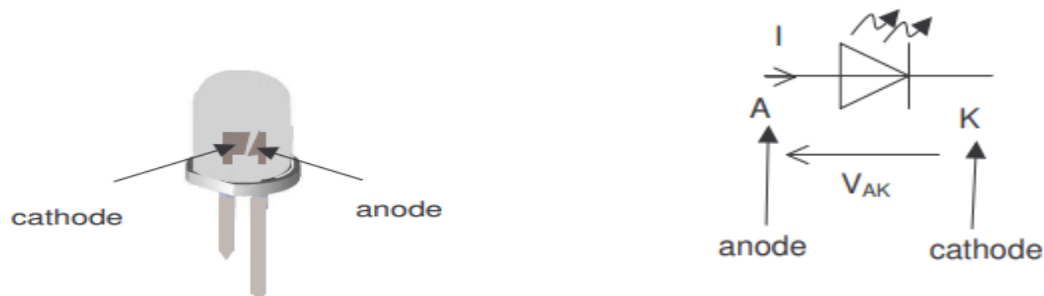


Figure IV.11 : le schéma d'une LED [13]

IV.1.2.6. La résistance:



Figure IV.12 : la résistance électrique [12]

Une résistance est un bipède, qui n'a pas de sens de branchement, en théorie, mais aussi en pratique. Sa grandeur principale se mesure en ohms (Ω). Elle va de quelques dixièmes d'ohms ($m\Omega$) à quelques giga ohms ($G\Omega$). La petite équation bien connue est $U = R \times I$. Elle décrit une relation entre la tension (qui est un potentiel comme la gravité) qui traverse la résistance, et le courant (qui est un flux, un débit). C'est comparable à une chute d'eau sur un torrent. Si la rivière est large (résistance faible), le courant est fort, si la rivière est étroite (résistance forte), le courant est faible. La hauteur de la chute représente la tension aux bornes de la résistance. Une résistance transforme de l'énergie électrique en énergie thermique. Ce mode de fonctionnement [12].

IV.1.2.7. Condensateurs:

Un condensateur est un bipède, qui a un sens de branchement la plupart du temps, selon la technologie employée, contrairement à la théorie. Sa grandeur principale se mesure en farads (F). Elle va de quelques picofarads (pF) à quelques centaines de farads (F). La petite équation est $I = C \cdot \frac{dU}{dt}$, l'intensité est proportionnelle aux variations de tension, ou $Q = C \cdot U$, la quantité de particule électrique stocké (en Coulomb), est proportionnel à la tension et la capacité. Les capacités, les capas, les condensateurs, sont des petites réserves d'énergie incontournables, sur une carte d'électronique numérique. [12].



Figure IV.13 : Condensateur électrique. [12]

IV.1.2.8. Bouton poussoir :

Ce sont des interfaces homme machine très simples, ils permettent de Lancer un fonctionnement par l'action « appuyer / relâcher ». Ces dispositifs sont des astables [14]

❖ Bouton poussoir marche et Chronogramme:

C'est un organe électrique de commande permettant de démarrer une ou plusieurs action, voir un cycle par une impulsion [14]

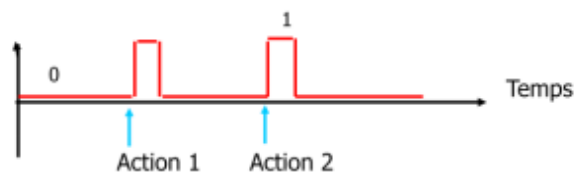


Figure IV.14 : Chronogramme Bouton poussoir marche [14]

❖ Schéma électrique et présentation



Figure IV.15 : Schéma électrique et présentation Bouton poussoir marche [14]

❖ Bouton poussoir d'arrêt et Chronogramme:

Il permet d'arrêter un fonctionnement par la simple action « Appuyer / Relâcher ». C'est un bouton complémentaire du bouton marche.

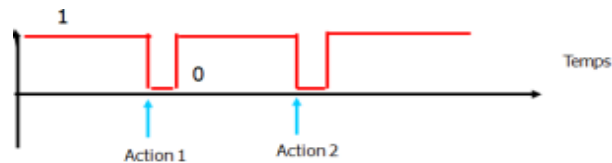


Figure IV.16 : Chronogramme Bouton poussoir d'arrêt [14]

❖ Schéma électrique et présentation:



Figure IV.17 : Schéma électrique et présentation Bouton poussoir d'arrêt [14]

IV.1.2.9. Potentiomètre:

Lorsqu'il est nécessaire de faire varier la valeur de la résistance, on utilise un système à curseur qui frotte sur celle-ci, faisant intervenir ainsi dans le circuit une portion variable de la résistance totale ; on réalise de la sorte un potentiomètre. [15]

Dans tous les cas la patte centrale est connectée au curseur comme le montre le symbole.

Symbole associé :

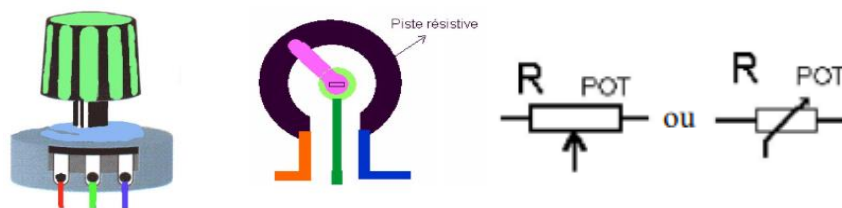


Figure IV.18 : Symbole électrique du potentiomètre et piste résistive [15]

Dans sa forme boîtier, le potentiomètre peut être de type **rotatif**, à forme circulaire, ou de type **linéaire à glissière**. Le trou de perçage pour la fixation ou dimension du canon est de 10 mm avec un axe de diamètre 6 mm pour le bouton en général.

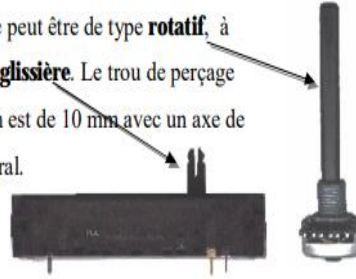
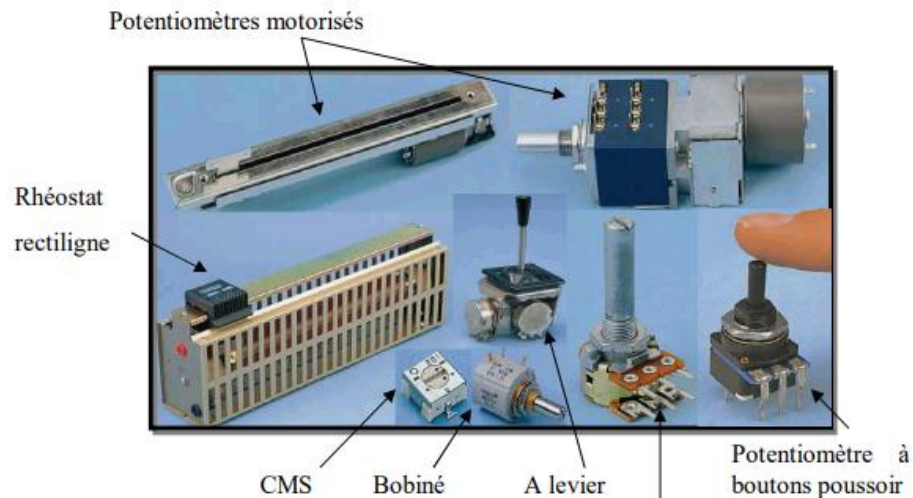


Figure IV.19 : Explication type du potentiomètre [15]



Il existe des potentiomètres double pour les applications audio stéréo et aussi avec interrupteur. Les pattes sont soit à souder ou déportées (sortie sur cosses avec un trou permettant une liaison filaire).

Figure IV.20 : Classification du potentiomètre [15]

IV.1.2.10. Moteur pas à pas:

Un moteur pas à pas transforme des impulsions de commande en une rotation de n° pas du rotor: il permet donc un positionnement précis sans boucle d'asservissement (via potentiomètre, codeur ...). [16]



Figure IV.21 : Moteur pas à pas [16]

❖ Principe de commande en position d'un moteur à courant continu:

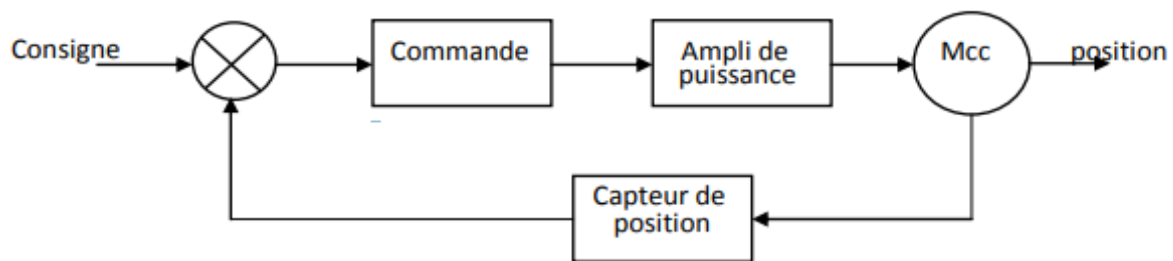


Figure IV.22 : Principe de commande en position d'un moteur à courant continu [16]

❖ Principe de commande d'un moteur pas à pas:

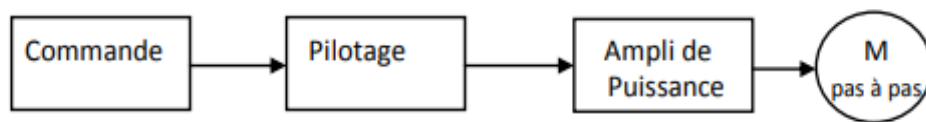


Figure IV.23 : Principe de commande d'un moteur pas à pas [16]

On constate que le système est beaucoup plus simple. En effet, à chaque impulsion du signal de commande correspond au niveau du rotor un déplacement angulaire bien défini appelé **pas**. [16]

Un moteur pas à pas est caractérisé par sa résolution ou encore son nombre de pas par tour. Il peut avoir une valeur comprise entre 0.9° et 90° . [16]

Les valeurs les plus couramment rencontrées sont :

- 0.9° : soit 400 pas par tour
- 1.8° : soit 200 pas par tour
- 3.6° : soit 100 pas par tour
- 7.5° : soit 48 pas par tour
- 15° : soit 24 pas par tour

La vitesse de rotation est fonction de la fréquence des impulsions. On distingue 3 groupes de moteurs pas à pas : [16]

- Les moteurs à aimant permanent
- Les moteurs à reluctance variable
- Les moteurs hybrides

IV.1.2.11. Module Bluetooth HC-05:



Figure IV.24 : Module Bluetooth HC-05 avec explication d'alimentation [17]

IV.1.2.12. Boîte d'Alimentation Power supply:

à quatre Channels 12 volt DC avec les caractéristiques suivantes:

- Input 88-264 V AC
- Output 12V DC / 4 channels

- Supply Current: 2.91A
- Integral Surge Protection
- Individual LED Power Indication
- Max Amp: 0.72 Amp/Channel.



Figure. IV.25 : Alimentation (power supply)

IV.2. Le Câblage

IV.2.1. Le Câblage Avec logiciel Fritzing :

IV.2.1.1 L'Application Fritzing:

❖ **Fritzing**: est un de conception de circuit imprimé qui permet de concevoir de façon entièrement graphique le circuit et d'en imprimer le typon.

❖ **Présentation de FRITZING:**

Se voulant dans la ligne d'Arduino et de processing, Fritzing est un projet de logiciel libre, destiné aux non-professionnels de l'électronique. Il a notamment pour vocation de favoriser l'échange de circuits électroniques libres et d'accompagner l'apprentissage de la conception de circuits.

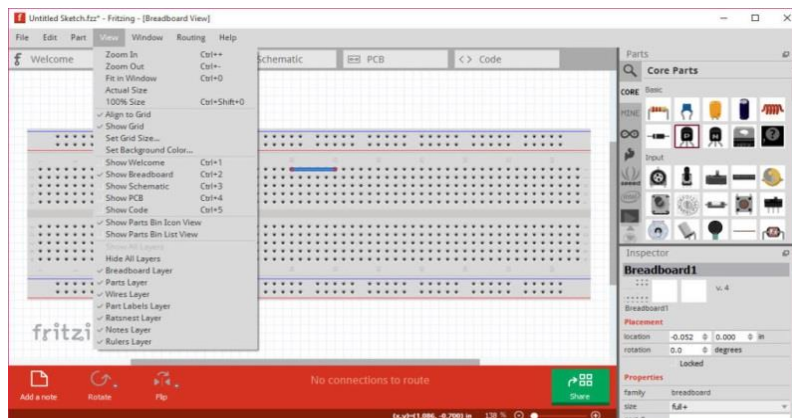


Figure.IV.26 :L'interface de logiciel fritzing

IV.2.1.2. Exécution du Câblage avec ARDUINO:

- ❖ **Le câblage d'un potentiomètre** : le positif (rouge) du potentiomètre peut être lié avec le 5V d'Arduino et le négatif (jaune) peut-être relié au GND, et le curseur peut être lié avec le pin A8 (bleu) d'analogue.
- ❖ **Le câblage du LED** : le négatif connecté avec GND et positif lié avec le pin digital 50

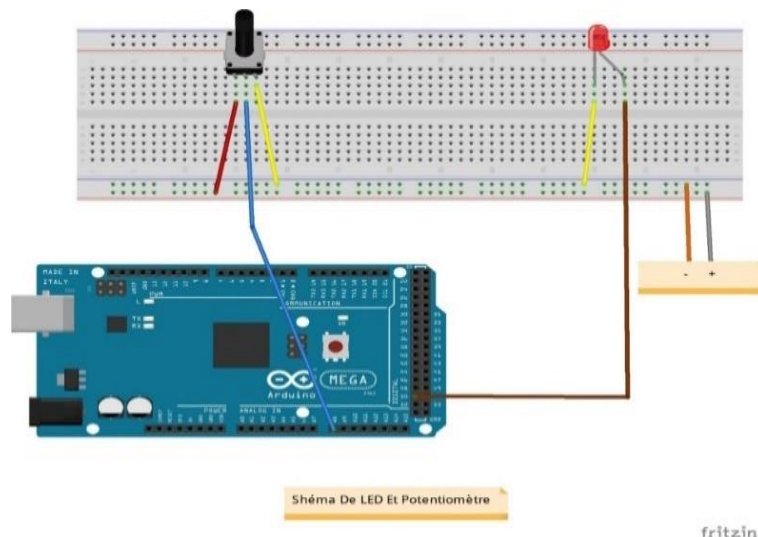


Figure.IV.27 : Câblage d'un potentiomètre et une LED dans un Arduino

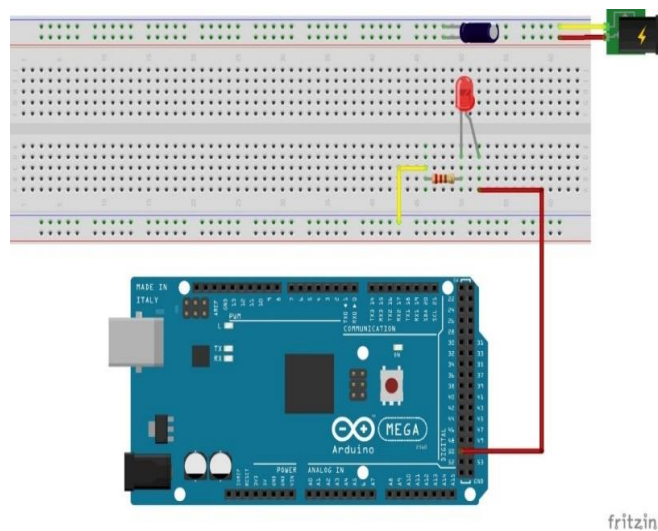


Figure. IV. 28: Câblage d'une résistance et d'un condensateur.

❖ Câblage d'un bouton poussoir :

Le câblage est du même principe que la LED

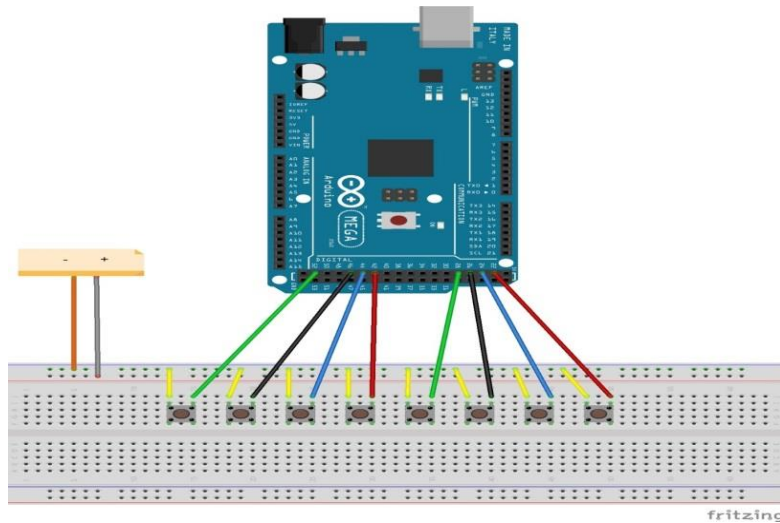


Figure. IV. 29 : Câblage d'un bouton poussoir

❖ Câblage d'un driver DRV8825:

Dans ce câblage, nous utilisons un moteur pas à pas. Les caractéristiques qui nous importent ici, sont la valeur de courant par phase, la résistance par phase, et le voltage recommandé :

Caractéristiques Moteur	Quantité	Unités
Courant par Phase	2.0	Ampères
Résistance par Phase	1.4	Ohms
Voltage Recommandé	12-24	Volts

Tableau.IV.2: Caractéristiques du moteur pas à pas

Le driver choisi est le Pololu DRV8825. Les caractéristiques de ce driver sont

Caractéristiques Driver	Quantité	Unités
Courant par Phase	2.2 Max	Ampères
Voltage Recommandé	8.2 - 45	Volts

Tableau.IV.3 : Caractéristiques du DRV8825

Pour réaliser ce circuit, installez votre driver sur la platine de test, et effectuez les diverses liaisons comme indiqué.

Attention : ne pas brancher l'alimentation moteur avant que tout ne soit installé et prêt, y compris les branchements avec l'Arduino. Il vaut mieux aussi que le programme de test soit déjà flashé.

Sur Lee schéma, les 4fils du moteur sont identifiés B2, B1, A1, A2. Sur notre moteur, nous avons des fils avec 4 couleurs différentes. La correspondance est :

Noir = A+, Vert = A-, Rouge = B+, Bleu = B- et Pour Les entrées et les sorties DRV8825 voir la partie théorique.

La figure suivante présente le câblage des steppers et des DRV8825 avec Arduino:

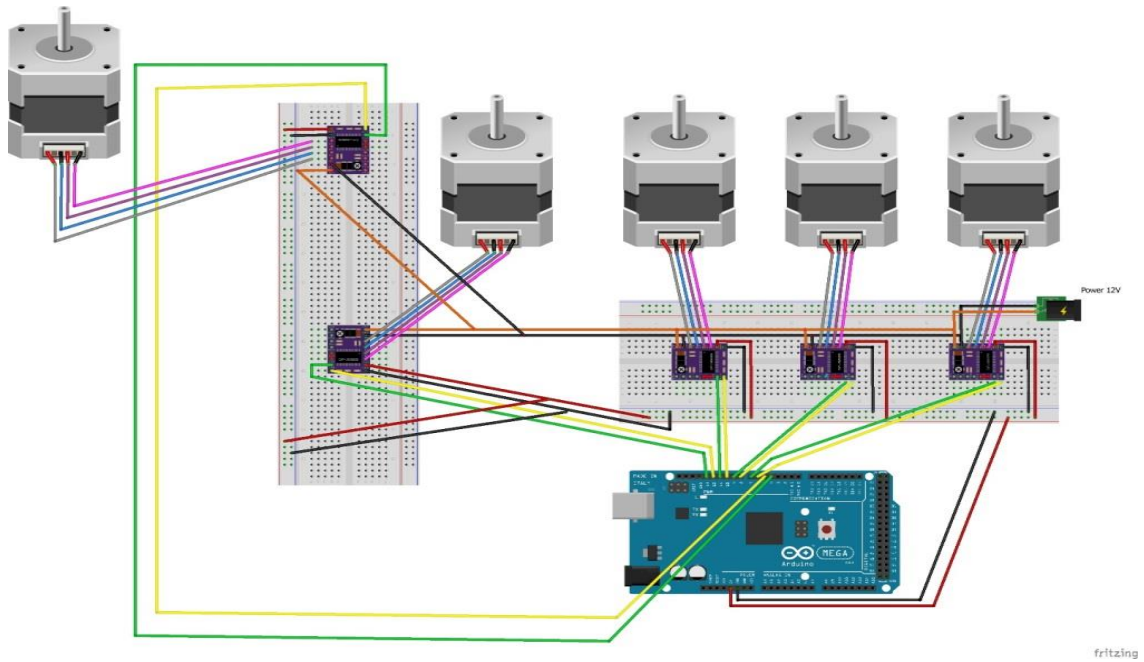


Figure IV.30: Câblage d'un driver

❖ **Le Câblage d'une alimentation et un HC-05 :**

Toutes les informations sont sur la partie théorique.

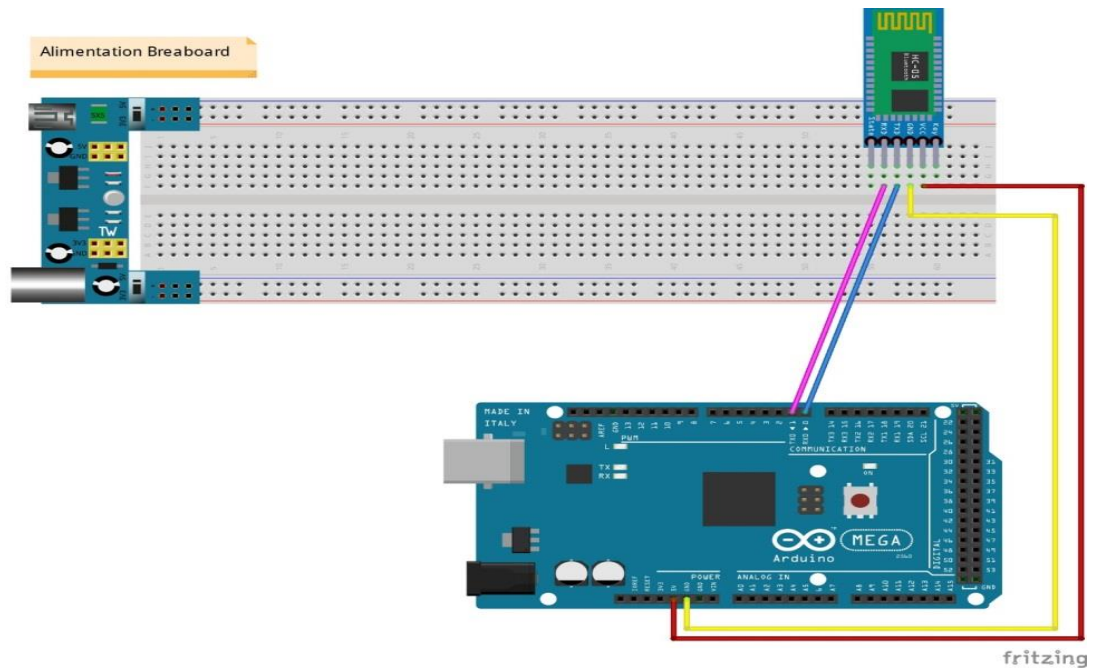


Figure.IV.31 : Câblage d'un alimentation et HC05.

IV.2.2 Le Câblage réel de tout groupe électronique :

IV.2.2.1 Le Câblage réel des éléments électronique extérieurs :

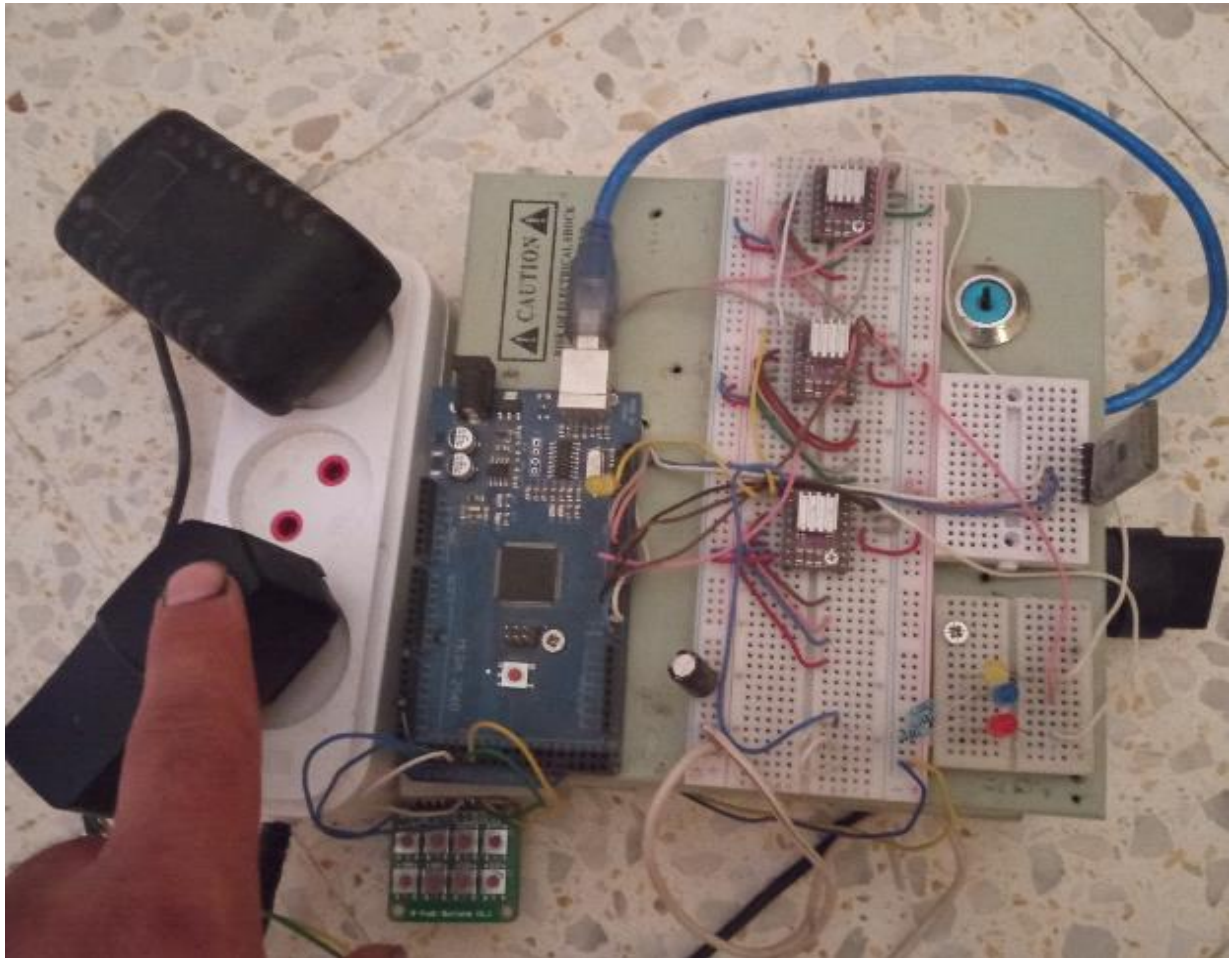


Figure.IV.32 : Câblage réel des éléments électronique extérieurs

Remarque 1:

- ✓ Toute les drivers utilisée dans cette câblage obligatoires ≥ 12 volts.
- ✓ Le condensateur est toujours après l'alimentation de source.
- ✓ Le LED est toujours (+) sur l'Arduino (pilotage) et le (-) après la résistance pour sécurité.
- ✓ Sur le HC-05 toujours Rx (HC-05) relie par Tx (Arduino) et Tx (HC-05) relie par Rx (Arduino).

IV.2.2.2 Le Câblage réel des éléments électronique intérieurs :

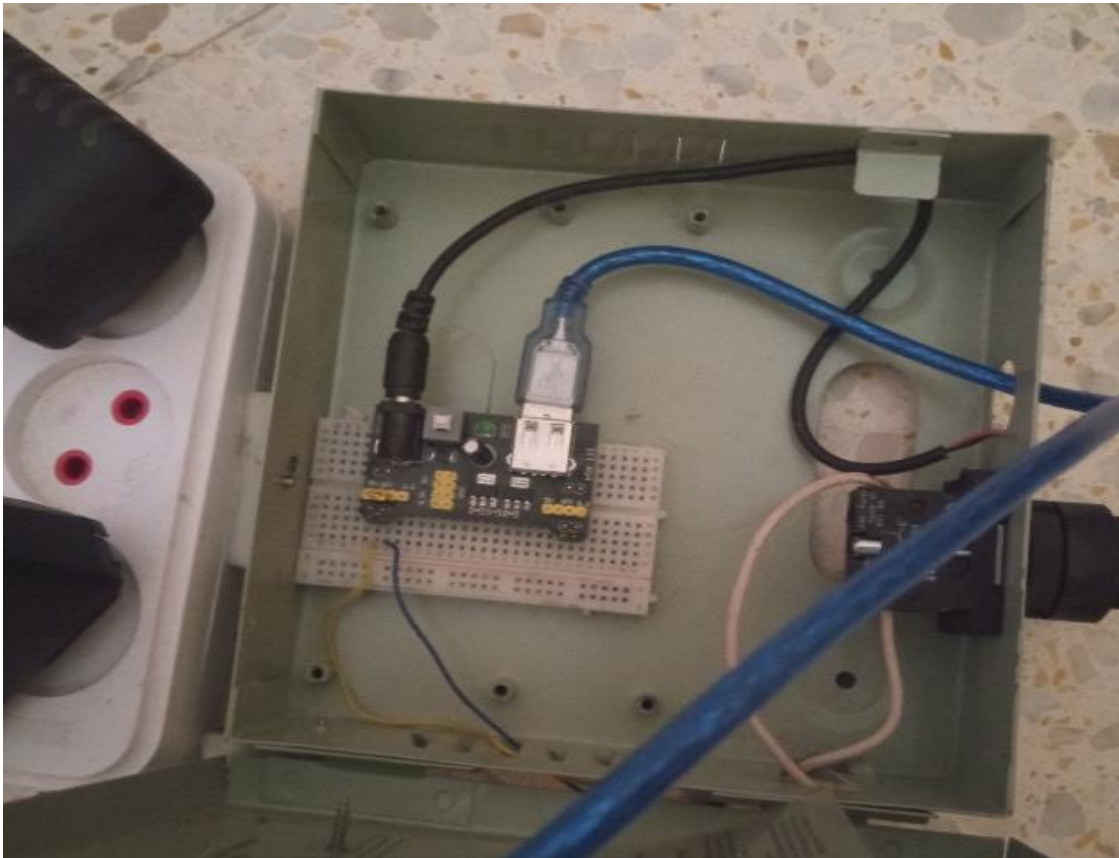
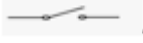


Figure.IV.33 : Câblage réel des éléments électronique intérieurs

Remarque 2:



- ✓ Pour câblage un sélecteur On peut connecter le fil en partant de la source avec un signe positif vers le numéro 3 Comme le montre l'image de symbole 3  4 et la deuxième borne positive portant le numéro 4 vers La platine d'expérimentation (breadboard) pour alimenter les drivers utilisé.

IV.2.2.3 Le Câblage réel des éléments électroniques avec les drivers DIV268N:



Figure.IV.34 : Câblage réel des éléments électroniques avec les drivers DIV268N

Remarque 3:

La caractéristique d'un driver DIV268N-5A pour Pilote de Moteur Pas à Pas Hybride 2 Phases

- ✓ Produit de qualité supérieure.
- ✓ Réduction automatique du courant d'inactivité.
- ✓ Tension d'alimentation jusqu'à 48 V DC.
- ✓ Courant de sortie jusqu'à 5 A.
- ✓ Entrée de signal d'isolation optoélectronique haute vitesse.

IV.2.2.4 Les éléments électroniques réels :



Figure.IV.35 : L'élément électronique réel

Remarque 4:

Ces éléments sont constitués d'un groupe électronique, qui est représenté dans

- ✓ Shield V3 Arduino Mega 2560
- ✓ Les Adafruits
- ✓ Transformateur
- ✓ DRIVERS ULN 2003
- ✓ Les relais
- ✓ Module wifi et module Bluetooth
- ✓ Potentiomètre et manette

IV.3. Présentation du logiciel Arduino :

IV.3.1. Définition du module Arduino:

Le module Arduino est un circuit imprimé en matériel libre, issu du travail d'une équipe d'enseignants et d'étudiants de l'école de Désigne d'interaction d'IVEREA, les plans de la carte elle-même sont publiés en licence libre dont certains composants de la carte, comme le microcontrôleur et les composants complémentaires qui ne sont pas en licence libre. Un microcontrôleur programmé peut analyser et produire des signaux électriques de manière à effectuer des tâches très diverses. Arduino est utilisé dans beaucoup d'applications comme l'électrotechnique industrielle et embarquée ; le modélisme, la domotique mais aussi dans des domaines différents comme l'art contemporain et le pilotage d'un robot, commande des moteurs et faire des jeux de lumières, communiquer avec l'ordinateur, commander des appareils mobiles (modélisme). Chaque module d'Arduino possède un régulateur de tension +5 V et un oscillateur à quarte 16 MHz (ou un résonateur céramique dans certains modèles). Pour programmer cette carte, on utilise l'logiciel IDE Arduino.

Les projets Arduino peuvent être autonomes, comme ils peuvent communiquer avec d'autres logiciels installés sur l'ordinateur tel que Flash, processing ou MaxMPS, Matlab. Ces cartes sont faites à base d'une interface entrée/sortie simple et d'un environnement de développement proche du langage.

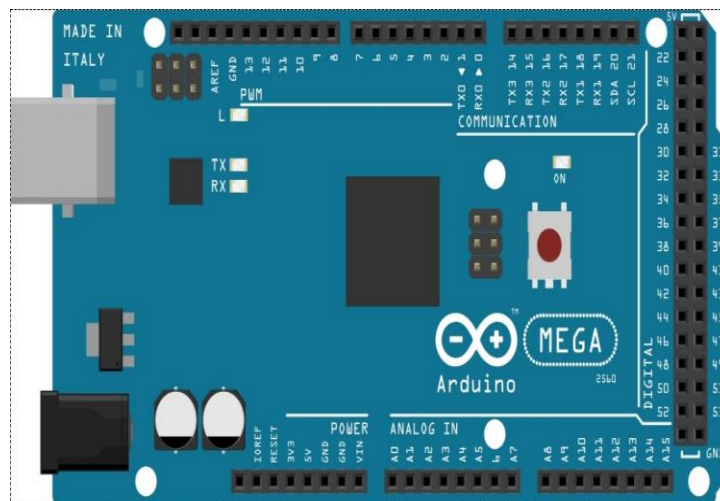


Figure IV.36 : Module d'Arduino.

IV.3.2. Présentation de l'interface d'Arduino:

❖ Lancement du logiciel :

Le lancement du logiciel se fait en double-cliquant sur l'icône avec le symbole "infinie" en vert. C'est l'exécutable du logiciel. Après un temps court, une image s'affiche :

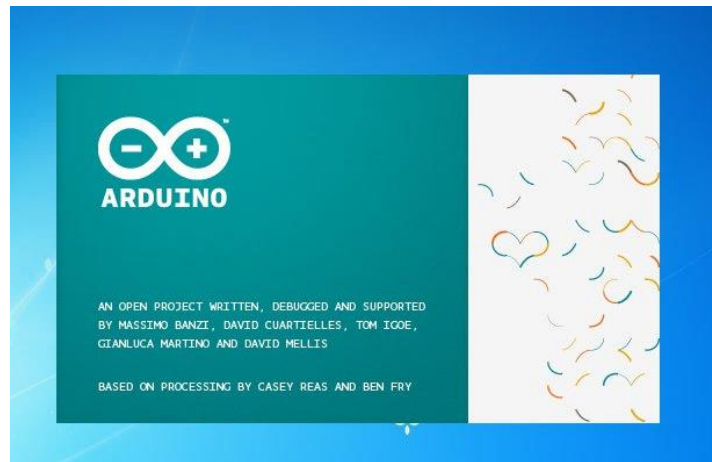


Figure IV.37 : Présentation du lancement de l'exécutable du logiciel d'Arduino.

Après quelques secondes, le logiciel s'exécute et Une fenêtre se présente :

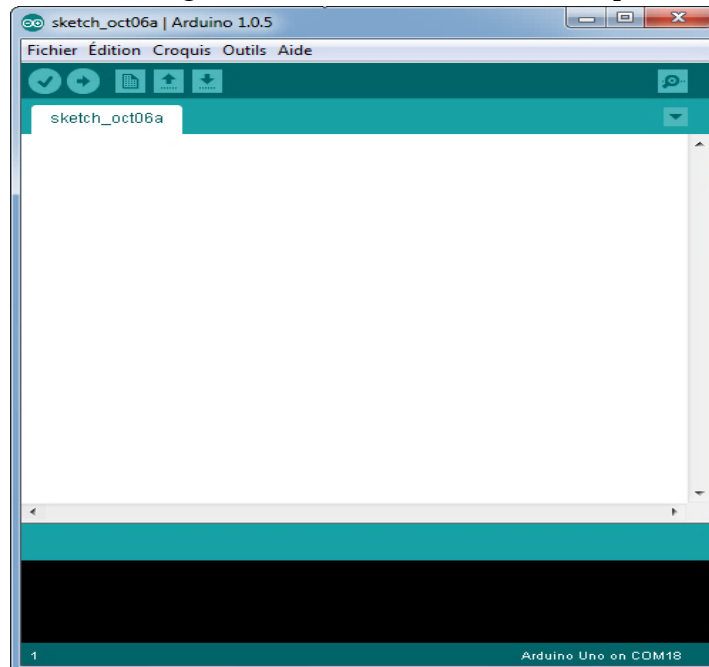


Figure IV.38 : L'interface du logiciel d'Arduino

Cette interface se compose des champs suivants :

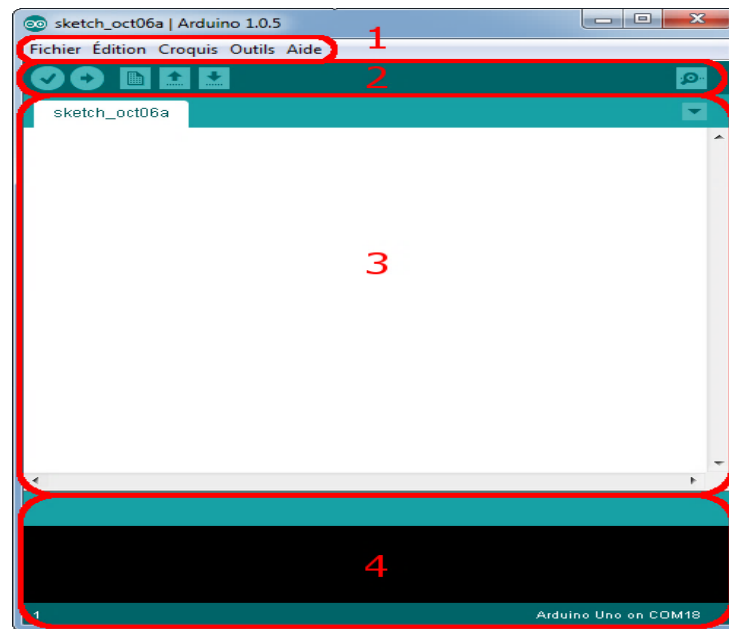


Figure.IV.39: Présentation des champs relatifs au logiciel d'Arduino.

- Le champ 1 : les options de configuration du logiciel.
- Le champ 2 : Contient les boutons servant à la programmation.
- Le champ 3 : Contient le programme crée.
- Le champ 4 : C'est le débogueur, c'est le plus important, car il permet de corriger les fautes dans le programme.

❖ Les options de configuration du logiciel.

I. Le menu File ou (Fichier):

C'est principalement le menu que l'on utilise le plus. Il dispose d'un certain nombre de commandes très utiles présentés comme suit :

- ✓ **Carnet de croquis** : C'est un menu qui regroupe les fichiers que l'on a pu faire jusqu'à maintenant (et s'ils sont enregistrés par défaut dans le dossier du logiciel).
- ✓ **Exemples (exemples)** : Englobe toute une liste déroulante qui affiche les noms des exemples de programmes existants ; avec lesquels on peut s'aider ou s'inspirer pour en créer d'autres programmes ou pour tester des composants.

- ✓ **Téléverser** : Permet d'envoyer le programme sur la carte Arduino
- ✓ **Téléverser avec un programmeur** : Idem que celle dessus, mais avec l'utilisation d'un programmeur (besoin très rarement de cette commande).
- ✓ **Préférences** : Pour le réglage des paramètres du logiciel.

Le reste des menus sera détaillé plus tard,

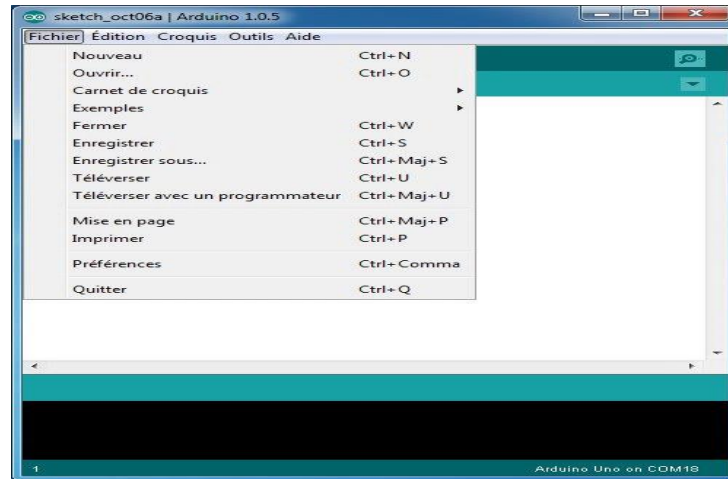


Figure.IV.40: Présentation des commandes du menu File.

II. Les boutons :

Les boutons du menu file sont présentés à la figure ci-dessous :

- ✓ **Bouton 1** : Ce bouton permet de vérifier le programme, il actionne un module qui cherche les erreurs dans le programme.
- ✓ **Bouton 2** : Charge (téléverse) le programme dans la carte Arduino.
- ✓ **Bouton 3** : Crée un nouveau fichier.
- ✓ **Bouton 4** : Ouvre un fichier.
- ✓ **Bouton 5** : Enregistre le fichier.
- ✓ **Bouton 6** : Ouvre le moniteur série.

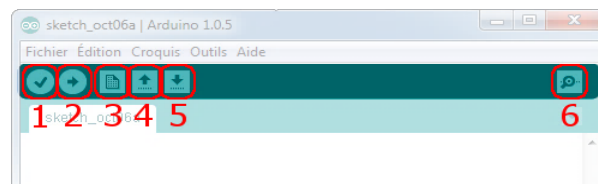


Figure.IV.41 : Présentation des boutons du menu File.

IV.3.3. La programmation en langage d'Arduino:

❖ Prise en main du langage Arduino :

De nombreux exemples pré-enregistrés peuvent être chargés et testés : ils permettent de se familiariser avec le langage plus rapidement. Testons par exemple :

Fichier > Exemples > 01 Basics > Blink.

Dans un premier temps, Le logiciel demande la sauvegarder le programme dans un dossier qui porte le même nom à la première compilation. Il faut observer les points suivants :

- Les majuscules et les minuscules sont prises en compte comme pour les fonctions et variables.
- La plupart des lignes se terminent par un point-virgule;
- Les blocs, boucles, fonctions sont encadrées par des accolades { }, qu'il faut apprendre à bien les repérer !
- Toutes les variables doivent être déclarées (nom et type).
- L'indentation (décalage) n'est pas obligatoire mais il est conseillé pour bien se repérer dans le programme.
- Les commentaires sont définis par // en début de ligne.
- Les blocs de commentaires par /* au début et */ à la fin.

❖ Le code minimal :

Avec Arduino, un code minimal doit être utilisé lorsque l'on crée un programme. Ce code permet de diviser le programme crée en deux grosses parties :

```
// Fonction d'initialisation de la carte
```

```
void setup ()
```

```
{ // Contenu de l'initialisation}
```

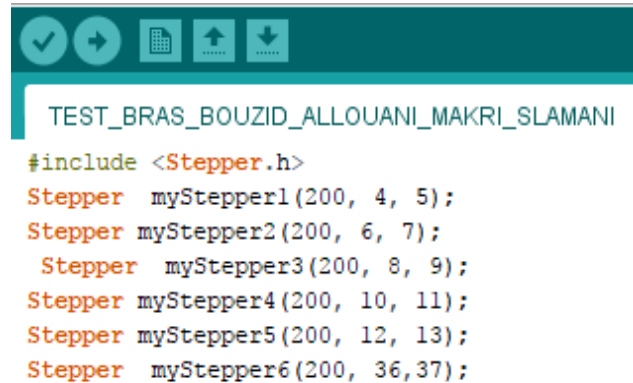
```
// Fonction principale, elle se répète (s'exécute) à l'infini
```

```
void loop ()
```

```
{ // Contenu de votre programme}
```

IV.4. Programme utilisée pour pilotage du robot :

- ❖ **La première partie:** On choisit la bibliothèque du moteur pas à pas avec une définition de direction et step de chaque moteur.



```
#include <Stepper.h>
Stepper myStepper1(200, 4, 5);
Stepper myStepper2(200, 6, 7);
Stepper myStepper3(200, 8, 9);
Stepper myStepper4(200, 10, 11);
Stepper myStepper5(200, 12, 13);
Stepper myStepper6(200, 36, 37);
```

Figure.IV.42 : La bibliothèque avec définition de direction et step du moteur pas à pas

- ❖ **Deuxième partie :** Définition des boutons poussoirs et choisit les entrées dans un Arduino.

```
const int pushbotonREV = 46;
const int ledpushbotonREV = 50;
const int pushbotonM6 = 44; //S6
const int pushbotonM5 = 42; //S5
const int pushbotonM4 = 28; //S4
const int pushbotonM3 = 26; //S3
const int pushbotonM2 = 24; //S2
const int pushbotonM1 = 22; //S1
const int pushbotonStart = 52;
```

Figure.IV.43 : Définition des boutons poussoirs

- ❖ **Troisième partie :** On utilise la boucle **bool** pour définition des boutons **marche** et **reverse**, (c'est-à-dire en **marche = 1** avec appuyer sans relâcher en même temps **reverse = 0** c'est la direction en même direction aigle d'heure, si non **reverse = 1** avec **marche = 1** sans relâcher c'est la direction contre direction aigle d'heure).

```
bool enMarche = 0, boutonAvant = 1, boutonMaintenant = 1;
bool reverse = 0, boutRevAvant = 1, boutonRevMaintenant = 1;
```

Figure.IV.44 : La boucle **bool** et les boutons (marche, reverse)

- ❖ **quatrième partie :** Dans cette partie on peut changer la vitesse de rotation de chaque moteur avec fournir une définition d'une l'entrée de chaque bouton poussoir par la boucle **pinMode** et la boucle **digitalWrite** pour donner 5V par la boucle **HIGH**.

```
void setup() {  
  
  myStepper1.setSpeed(100);  
  myStepper2.setSpeed(100);  
  myStepper3.setSpeed(100);  
  myStepper4.setSpeed(100);  
  myStepper5.setSpeed(100);  
  myStepper6.setSpeed(100);  
  pinMode(pushbotonREV , INPUT);  
  digitalWrite(pushbotonREV , HIGH);  
  pinMode(ledpushbotonREV , OUTPUT);  
  pinMode(pushbotonM6, INPUT);  
  digitalWrite(pushbotonM6, HIGH);  
  pinMode(pushbotonM5, INPUT);  
  digitalWrite(pushbotonM5, HIGH);  
  pinMode(pushbotonM4, INPUT);  
  digitalWrite(pushbotonM4, HIGH);  
  pinMode(pushbotonM3, INPUT);  
  digitalWrite(pushbotonM3, HIGH);  
  pinMode(pushbotonM2, INPUT);  
  digitalWrite(pushbotonM2, HIGH);  
  pinMode(pushbotonM1, INPUT);  
  digitalWrite(pushbotonM1, HIGH);  
  pinMode(pushbotonStart, INPUT);  
  digitalWrite(pushbotonStart , HIGH);  
  
}
```

Figure.IV.45 : Fournir une définition d'une l'entrée et donner le voltage Maximin de chaque bouton poussoir

- ❖ **Cinquième partie:** La rotation dans le sens spécifié à l'aide le bouton poussoir

```

void loop() {
    boutonRevMaintenant = digitalRead(pushbotonREV );
    digitalWrite(ledpushbotonREV , reverse);

    if (boutRevAvant && !boutonRevMaintenant)
    {
        reverse = !reverse;
    }

    if (!digitalRead(pushbotonM6)) {
        if (reverse) {
            myStepper6.step(-100);
        }
        else {
            myStepper6.step(100);
        }
    }
}

```

Figure.IV.46 : pilotage moteur 6 à l'aide le bouton poussoir M6

```

    boutonRevMaintenant = digitalRead(pushbotonREV );
    digitalWrite(ledpushbotonREV , reverse);

    if (boutRevAvant && !boutonRevMaintenant)
    {
        reverse = !reverse;
    }

    if (!digitalRead(pushbotonM5)) {
        if (reverse) {
            myStepper5.step(-100);
        }
        else {
            myStepper5.step(100);
        }
    }
}

```

Figure.IV.47 : pilotage moteur 5 à l'aide le bouton poussoir M5

```

boutonRevMaintenant = digitalRead(pushbotonREV );
digitalWrite(ledpushbotonREV , reverse);

if (boutRevAvant && !boutonRevMaintenant)
{
    reverse = !reverse;
}

if (!digitalRead(pushbotonM4)) {
    if (reverse) {
        myStepper4.step(-100);
    }
    else {
        myStepper4.step(100);
    }
}

```

Figure.IV.48 : pilotage moteur 4 à l'aide le bouton poussoir M4

```

boutonRevMaintenant = digitalRead(pushbotonREV );
digitalWrite(ledpushbotonREV , reverse);

if (boutRevAvant && !boutonRevMaintenant)
{
    reverse = !reverse;
}

if (!digitalRead(pushbotonM3)) {
    if (reverse) {
        myStepper3.step(-100);
    }
    else {
        myStepper3.step(100);
    }
}

```

Figure.IV.49 : pilotage moteur 3 à l'aide le bouton poussoir M3

```

boutonRevMaintenant = digitalRead(pushbotonREV );
digitalWrite(ledpushbotonREV , reverse);

if (boutRevAvant && !boutonRevMaintenant)
{
    reverse = !reverse;
}

if (!digitalRead(pushbotonM2)) {
    if (reverse) {
        myStepper2.step(-100);
    }
    else {
        myStepper2.step(100);
    }
}

```

Figure.IV.50 : pilotage moteur 2 à l'aide le bouton poussoir M2

```

boutonRevMaintenant = digitalRead(pushbotonREV );
digitalWrite(ledpushbotonREV , reverse);

if (boutRevAvant && !boutonRevMaintenant)
{
    reverse = !reverse;
}

if (!digitalRead(pushbotonM1)) {
    if (reverse) {
        myStepper1.step(100);
    }
    else {
        myStepper1.step(100);
    }
}

```

Figure.IV.51 : pilotage moteur 1 à l'aide le bouton poussoir M1

```

boutonRevMaintenant = digitalRead(pushbotonREV );
digitalWrite(ledpushbotonREV , reverse);

if (boutRevAvant && !boutonRevMaintenant)
{
    reverse = !reverse;
}

if (!digitalRead(pushbotonStart)) {
    if (reverse) {
        myStepper1.step(100);
        myStepper2.step(-100);
        myStepper3.step(-100);
        myStepper4.step(-100);
        myStepper5.step(-100);
        myStepper6.step(100);
    }
    else {
        myStepper1.step(100);
        myStepper2.step(100);
        myStepper3.step(100);
        myStepper4.step(100);
        myStepper5.step(100);
        myStepper6.step(100);
    }
}
}
}

```

Figure.IV.52 : pilotage robot à l'aide le bouton poussoir Start

Conclusion

Ce chapitre consiste à connecter l'électronique, pour ce projet nous aurons besoin d'une carte Arduino Mega pour la commande, d'un module Bluetooth HC-05 pour la communication avec le smartphone, et d'une source d'alimentation de 12V. Dans ce qui suit, nous allons décrire le matériel utilisé, et montrer comment nous pouvons associer les différents composants en eux.

Avec l'explication exacte de boucle utilisée dans ce programme.

CHAPITRE.V: COMMANDE AVEC UNE APPLICATION ANDROID

Introduction :

RemoteXY est un moyen facile de créer et d'utiliser une interface utilisateur graphique mobile pour les cartes contrôleurs à contrôler via un smartphone ou une tablette. Le système comprend éditeur d'interfaces graphiques mobiles pour cartes contrôleurs, situé sur le site remotxy application mobile RemoteXY qui permet de se connecter au contrôleur et de le contrôler via une interface graphique. [18]

V.1 : L'interface d'application remotxy:

Pour utiliser l'application Android, nous suivons les étapes suivantes :

- On va entrer sur le site de <http://remotxy.com/>
- On clique sur START NOW

On va voir la fenêtre suivante :



Figure.V.1 : L'interface d'application remotxy [18]

V.2 : Création d'application :

L'éditeur d'interfaces est environnement de développement en ligne, héberge sur ce site. L'éditeur est conçu pour le développement de l'interface utilisateur graphique et la génération de code source du microcontrôleur.

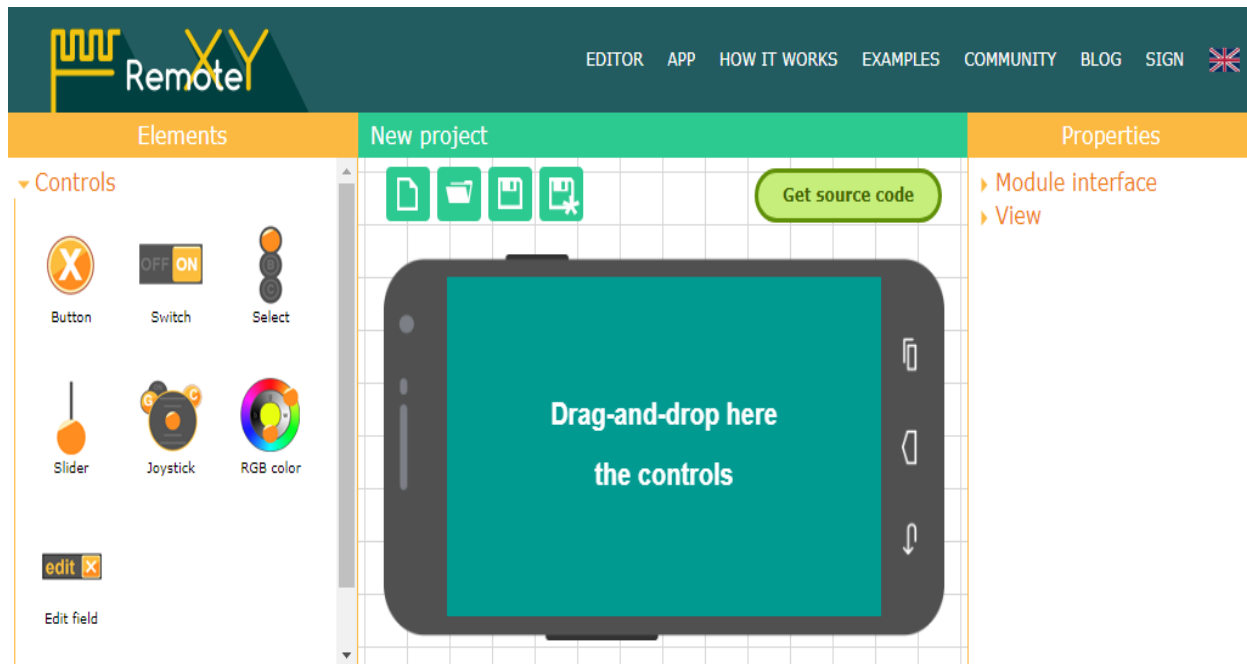


Figure.V.2 : Création d'application remotexy [18]

V.3 : Les zones de la fenêtre de l'éditeur:

La fenêtre de l'éditeur comporte trois zones, divisées par fonction:

❖ La zone des commandes:

La zone des commandes est à gauche. Tous les contrôles disponibles regroupés. Pour la navigation, vous pouvez ouvrir et fermer des groupes de commandes. Pour placer l'élément dans la boîte de l'éditeur, faites un clic droit sur l'élément et sans relâcher le bouton pour le déplacer dans la boîte de l'éditeur (glisser-déposer).

❖ La zone de l'éditeur:

La zone de l'éditeur est au centre. La zone contient la boîte de l'éditeur principal en tant que champ Smartphone. Dans cette zone, vous définissez des éléments et créez une interface. De plus, la zone contenant les boutons de contrôle vous permet de créer un nouveau projet, d'enregistrer le projet, d'ouvrir le projet. En haut à droite se trouve le bouton qui permet d'obtenir le code source de l'interface du microcontrôleur.

❖ La zone des propriétés et des paramètres:

À droite se trouve la zone des propriétés et des paramètres. Les propriétés et les paramètres sont regroupés. Pour la navigation, vous pouvez ouvrir et fermer les propriétés et les paramètres du groupe. Les paramètres permettent de définir la configuration du projet, de définir les paramètres de connexion du module de communication. Les propriétés vous permettent de spécifier les paramètres d'affichage et les paramètres des éléments de contrôle.

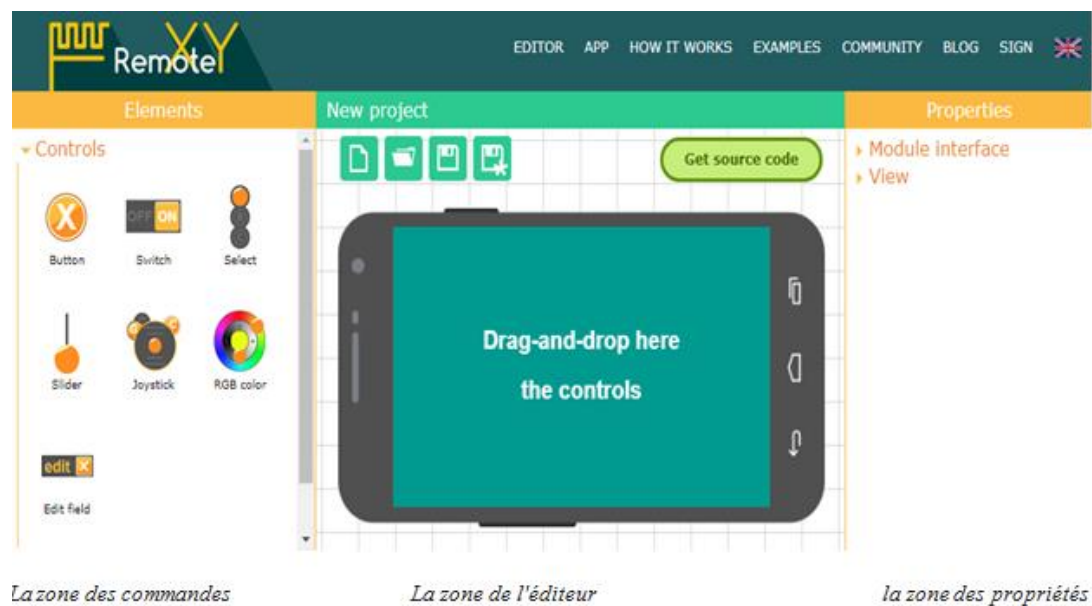


Figure.V.3 : Les zones de la fenêtre de l'éditeur [18]

V.4 : Orientation de l'interface graphique de l'écran:

RemoteXY prend en charge le réglage de l'orientation de l'écran de l'interface graphique vers l'application mobile. Établi l'orientation horizontale de l'écran par défaut.

Vous pouvez sélectionner l'un des trois modes d'orientation :

- Horizontal
- Verticale
- Les deux (horizontal et vertical)

Le mode détermine le comportement de l'interface dans une application mobile. S'il est réglé sur le mode d'orientation horizontale ou verticale, l'écran de l'application mobile accepte fermement

cette orientation. Si elle est définie sur "Les deux", l'application mobile changera l'orientation en fonction du capteur, comme c'est le cas dans ce téléphone ou cette tablette.

Le mode d'orientation du projet en cours peut être sélectionné dans les paramètres "**Affichage**" du panneau des propriétés.

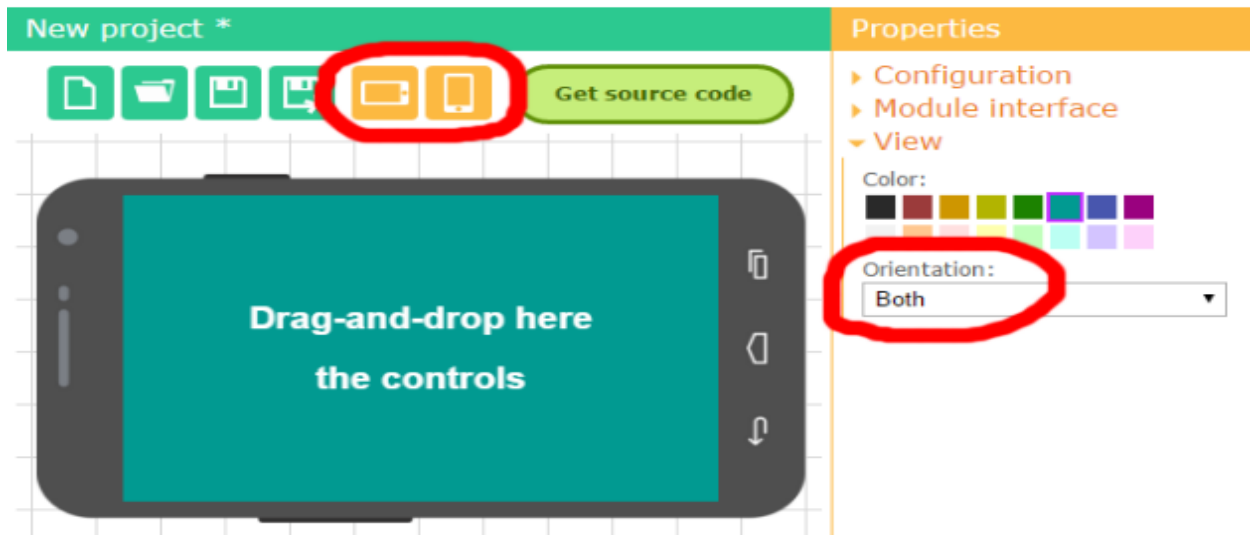


Figure.V.4 : Orientation de l'interface graphique de l'écran [18]

V.5 : Liste de configuration :

Cette liste pour définir la configuration du projet, définir les paramètres de connexion du module de communication

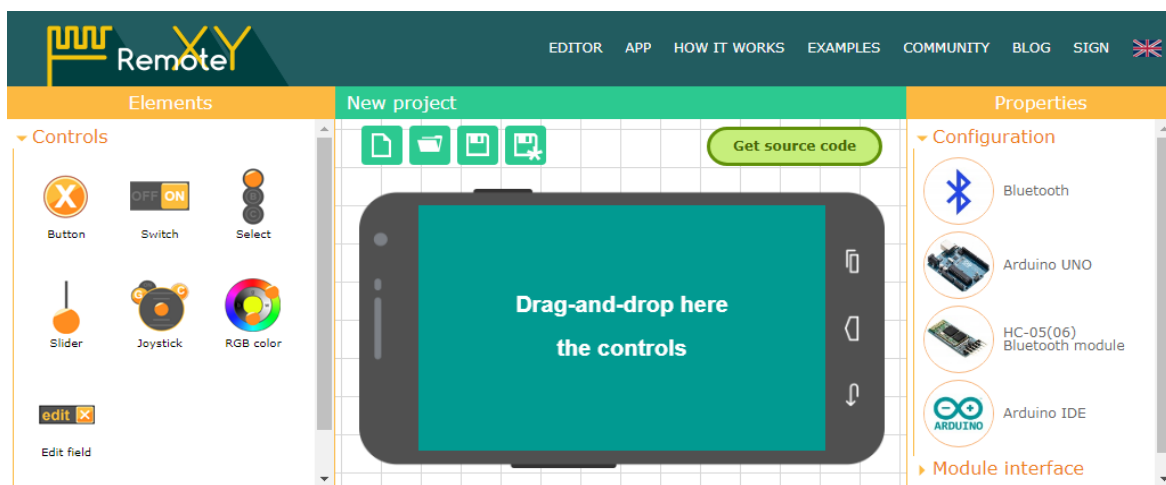


Figure.V.5 : Liste de configuration [18]

Dans le volet de droite, sélectionnez les paramètres suivants sous l'onglet "**Configuration**".



Figure.V.6 : Liste choisie pour ce projet [18]

Premièrement : On choisit Connexion Bluetooth

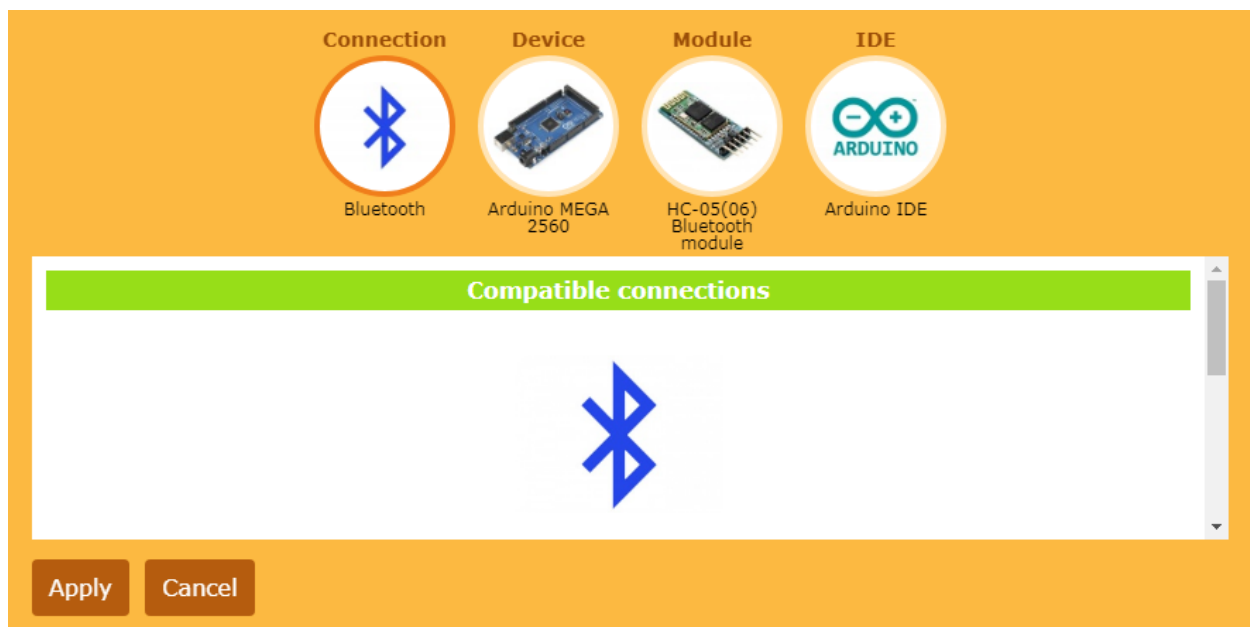


Figure.V.7 : Connexion Bluetooth [18]

Deuxièmement : on choisit Arduino MEGA 2560



Figure.V.8 : Module d'Arduino[18]

Troisièmement : On choisit le module HC-05



Figure.V.9: Module Bluetooth HC-05[18]

Quatrièmement : On choisit logiciel de la programmation



Figure.V.10 : Logiciel de programmation [18]

V.6 : Liste Module interface:

Entrée sur **Module interface**

On sélectionne la connexion interface et on choisit Hardware Serial

❖ Liste View:

Dans une liste View on choisit la couleur et l'orientation



Figure.V.11 : Module d'interface et liste de view [18]

❖ Liste la zone des commandes:

Dans la zone des commandes on observe les éléments principales pour le déplacer dans la boîte de l'éditeur).

❖ Liste de contrôle, d'indication et décoration

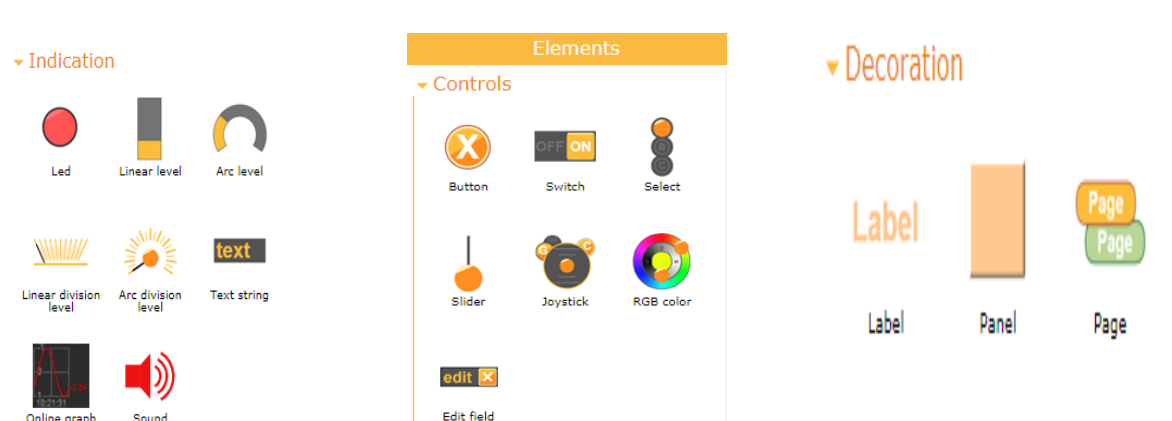


Figure.V.12 : Liste des contrôles, d'indication et décoration [18]

On finalement l'interface suivante résume toute les étapes précédentes



Figure.V.13 : L'interface de l'architecture globale

V.7 : Programme de pilotage par Bluetooth:

```
Memoire_Bouid_Allouani_Makri_Slamani $
#define REMOTEXY_MODE __HARDSERIAL

#include <RemoteXY.h>
#include <Stepper.h>
Stepper myStepper1(200, 4, 5);
Stepper myStepper2(200, 6, 7);
Stepper myStepper3(200, 8, 9);
Stepper myStepper4(200, 10, 11);
Stepper myStepper5(200, 12, 13);
Stepper myStepper6(200, 36, 37);
// RemoteXY connection settings
#define REMOTEXY_SERIAL Serial
#define REMOTEXY_SERIAL_SPEED 9600

// RemoteXY configurate
#pragma pack(push, 1)
uint8_t RemoteXY_CONF[] =
{ 255, 9, 0, 0, 0, 245, 0, 11, 7, 0,
  1, 0, 8, 0, 16, 16, 36, 31, 112, 117,
  115, 104, 98, 111, 116, 111, 110, 77, 49, 0,
  1, 0, 29, 0, 16, 16, 14, 31, 112, 117,
  115, 104, 98, 111, 116, 111, 110, 77, 50, 0,
  1, 0, 49, 0, 16, 16, 13, 31, 112, 117,
  115, 104, 98, 111, 116, 111, 110, 77, 51, 0,
  1, 0, 8, 30, 16, 16, 247, 31, 112, 117,
  115, 104, 98, 111, 116, 111, 110, 77, 52, 0,
  1, 0, 29, 30, 16, 16, 53, 31, 112, 117,
  115, 104, 98, 111, 116, 111, 110, 77, 53, 0,
  1, 0, 49, 30, 16, 16, 5, 31, 112, 117,
  115, 104, 98, 111, 116, 111, 110, 77, 54, 0,
  1, 1, 72, 23, 28, 23, 108, 31, 112, 117,
  115, 104, 98, 111, 116, 111, 110, 82, 69, 86,
  0, 1, 1, 0, 46, 100, 17, 2, 31, 85,

  0, 1, 1, 0, 46, 100, 17, 2, 31, 85,
  78, 73, 86, 69, 82, 83, 73, 84, 69, 32,
  68, 69, 32, 77, 39, 83, 73, 76, 65, 32,
  32, 32, 32, 32, 32, 32, 32, 32, 32, 32,
  32, 32, 32, 66, 79, 85, 90, 73, 68, 32,
  65, 66, 68, 69, 76, 66, 65, 83, 69, 84,
  32, 69, 84, 32, 76, 65, 82, 66, 73, 32,
  65, 76, 76, 79, 85, 65, 78, 73, 0, 1,
  1, 72, 0, 28, 23, 13, 31, 112, 117, 115,
  104, 98, 111, 116, 111, 110, 83, 116, 97, 114,
  116, 0 };

```

```

// this structure defines all the variables and events of your control interface
struct {

    // input variables
    uint8_t pushbotonM1; // =1 if button pressed, else =0
    uint8_t pushbotonM2; // =1 if button pressed, else =0
    uint8_t pushbotonM3; // =1 if button pressed, else =0
    uint8_t pushbotonM4; // =1 if button pressed, else =0
    uint8_t pushbotonM5; // =1 if button pressed, else =0
    uint8_t pushbotonM6; // =1 if button pressed, else =0
    uint8_t pushbotonREV; // =1 if button pressed, else =0
    uint8_t button_1; // =1 if button pressed, else =0
    uint8_t pushbotonStart; // =1 if button pressed, else =0

    // other variable
    uint8_t connect_flag; // =1 if wire connected, else =0

} RemoteXY;

#pragma pack(pop)

//////////////////////
//          END RemoteXY include          //
//////////////////////

#define PIN_PUSHBOTONM1 22
#define PIN_PUSHBOTONM2 24
#define PIN_PUSHBOTONM3 26
#define PIN_PUSHBOTONM4 28
#define PIN_PUSHBOTONM5 42
#define PIN_PUSHBOTONM6 44
#define PIN_PUSHBOTONREV 46
#define PIN_PUSHBOTONSTART 52
const int ledpushbotonREV = 50;
bool enMarche = 0, boutonAvant = 1, boutonMaintenant = 1;
bool reverse = 0, boutRevAvant = 1, boutonRevMaintenant = 1;

void setup()
{
    myStepper1.setSpeed(140);
    myStepper2.setSpeed(100);
    myStepper3.setSpeed(100);
    myStepper4.setSpeed(100);
    myStepper5.setSpeed(100);
    myStepper6.setSpeed(100);

    RemoteXY_Init ();

    pinMode (PIN_PUSHBOTONM1, OUTPUT);
    pinMode (PIN_PUSHBOTONM2, OUTPUT);
    pinMode (PIN_PUSHBOTONM3, OUTPUT);
    pinMode (PIN_PUSHBOTONM4, OUTPUT);
    pinMode (PIN_PUSHBOTONM5, OUTPUT);
    pinMode (PIN_PUSHBOTONM6, OUTPUT);
    pinMode (PIN_PUSHBOTONREV, OUTPUT);
    pinMode (PIN_PUSHBOTONSTART, OUTPUT);
    pinMode (ledpushbotonREV , OUTPUT);
    // TODO you setup code
}

```

```

void loop()
{
    RemoteXY_Handler ();

    digitalWrite(PIN_PUSHBOTONM1, (RemoteXY.pushbotonM1==0)?HIGH:LOW);
    digitalWrite(PIN_PUSHBOTONM2, (RemoteXY.pushbotonM2==0)?HIGH:LOW);
    digitalWrite(PIN_PUSHBOTONM3, (RemoteXY.pushbotonM3==0)?HIGH:LOW);
    digitalWrite(PIN_PUSHBOTONM4, (RemoteXY.pushbotonM4==0)?HIGH:LOW);
    digitalWrite(PIN_PUSHBOTONM5, (RemoteXY.pushbotonM5==0)?HIGH:LOW);
    digitalWrite(PIN_PUSHBOTONM6, (RemoteXY.pushbotonM6==0)?HIGH:LOW);
    digitalWrite(PIN_PUSHBOTONREV, (RemoteXY.pushbotonREV==0)?HIGH:LOW);
    digitalWrite(PIN_PUSHBOTONSTART, (RemoteXY.pushbotonStart==0)?HIGH:LOW);

    boutonRevMaintenant = digitalRead(pushbotonREV );
    digitalWrite(ledpushbotonREV , reverse);

    if (boutRevAvant == !boutonRevMaintenant)
    {
        reverse = !reverse;
    }

    if (!digitalRead(pushbotonM6)) {
        if (reverse) {
            myStepper6.step(-100);
        }
        else {
            myStepper6.step(100);
        }
    }

    boutonRevMaintenant = digitalRead(pushbotonREV );
    digitalWrite(ledpushbotonREV , reverse);

    if (boutRevAvant == !boutonRevMaintenant)
    {
        reverse = !reverse;
    }

    if (!digitalRead(pushbotonM5)) {
        if (reverse) {
            myStepper5.step(-100);
        }
        else {
            myStepper5.step(100);
        }
    }

    boutonRevMaintenant = digitalRead(pushbotonREV );
    digitalWrite(ledpushbotonREV , reverse);

    if (boutRevAvant == !boutonRevMaintenant)
    {
        reverse = !reverse;
    }

    if (!digitalRead(pushbotonM4)) {
        if (reverse) {
            myStepper4.step(-100);
        }
        else {
            myStepper4.step(100);
        }
    }
}

```

```

boutonRevMaintenant = digitalRead(pushbotonREV );
digitalWrite(ledpushbotonREV , reverse);

if (boutRevAvant && !boutonRevMaintenant)
{
    reverse = !reverse;
}

if (!digitalRead(pushbotonM3)) {
    if (reverse) {
        myStepper3.step(-100);
    }
    else {
        myStepper3.step(100);
    }
}

boutonRevMaintenant = digitalRead(pushbotonREV );
digitalWrite(ledpushbotonREV , reverse);

if (boutRevAvant && !boutonRevMaintenant)
{
    reverse = !reverse;
}

if (!digitalRead(pushbotonM2)) {
    if (reverse) {
        myStepper2.step(-100);
    }
    else {
        myStepper2.step(100);
    }
}

boutonRevMaintenant = digitalRead(pushbotonREV );
digitalWrite(ledpushbotonREV , reverse);

if (boutRevAvant && !boutonRevMaintenant)
{
    reverse = !reverse;
}

boutonRevMaintenant = digitalRead(pushbotonREV );
digitalWrite(ledpushbotonREV , reverse);

if (boutRevAvant && !boutonRevMaintenant)
{
    reverse = !reverse;
}

if (!digitalRead(pushbotonM1)) {
    if (reverse) {
        myStepper1.step(100);
    }
    else {
        myStepper1.step(100);
    }
}

```

```

boutonRevMaintenant = digitalRead(pushbotonREV );
digitalWrite(ledpushbotonREV , reverse);

if (boutRevAvant && !boutonRevMaintenant)
{
    reverse = !reverse;
}

if (!digitalRead(pushbotonStart)) {
    if (reverse) {
        myStepper1.step(100);
        myStepper2.step(-100);
        myStepper3.step(-100);
        myStepper4.step(-100);
        myStepper5.step(-100);
        myStepper6.step(100);
    }

    boutonRevMaintenant = digitalRead(pushbotonREV );
    digitalWrite(ledpushbotonREV , reverse);

    if (boutRevAvant && !boutonRevMaintenant)
    {
        reverse = !reverse;
    }

    if (!digitalRead(pushbotonM1)) {
        if (reverse) {
            myStepper1.step(100);
        }
        else {
            myStepper1.step(100);
        }
    }
}

```

Conclusion

Dans ce chapitre nous avons cité quelque étape (Connexion Bluetooth, Module d'Arduino ,Module Bluetooth HC-05, Logiciel de programmation) pour utiliser L'application Android et piloter le robot par programme Bluetooth précédent

Conclusion générale:

Lors de ce projet intitulé « Conception d'un robot serial », nous avons dans un premier temps, dans le premier chapitre, introduit quelques notions générales relatives aux robots manipulateurs, notamment leur définition, leur classification, leurs structures, à savoir les structures, anthropomorphe, cartésienne, SCARA, sphérique, cylindrique, plane, DELTA, ainsi que leurs domaines d'application, en finissant par la langue de programmation d'Arduino et autre aussi la problématique et la solution adopté .

Dans le deuxième chapitre nous résumons les équations direct et inverse de cinématique.

Le troisième chapitre nous somme entré dans la modélisation, la conception et le montage mécanique de notre bras manipulateur de six degré avec dessin technique de chaque élément.

Dans le troisième chapitre, Il y a deux parties(théorique, pratique) ,dans la première partie on définit toutes notions générales des éléments utilisée, on deuxième partie on réalisit du circuit électronique composé d'un Arduino.

Dans le quatrième chapitre on utilise l'application Android avec module Bluetooth, et en. Dans ce projet, j'ai essayé de toucher à un maximum de domaines possible, la modélisation, la mécanique, l'électronique, le développement mobile. Je me suis efforcé de tirer le maximum du potentiel de l'Arduino en l'utilisant de différentes façons. J'ai énormément appris durant la réalisation de ce projet, tout en m'amusant, et tout en essayant de faire le plus de manipulation possible. Néanmoins j'aimerais approfondir le sujet de la robotique des bras manipulateurs.

En perspective, pour aller plus loin dans ce domaine :

- Conception d'un nouveau modèle de bras a six ou sept degrés de liberté
- Utiliser des moteurs pas à pas, avec asservissement en vitesse et position
- Utiliser les équations de rotation
- Choix de composants électroniques plus adaptés

La bibliographie

- [1] COMMANDE D'UN BRAS ROBOTISE AVEC UNE APPLICATION ANDROID
Université de Lorraine Faculté des sciences et technologies de Nancy 2018/2019 YACINE
TAZEROUT
- [2] Patrick Maurine, Jean-François Quinet. (2013). L'étalonnage des robots manipulateurs
industriels (Hermes Science publications). Paris/France.
- [3] Charles BOP. (2010). Traité de robotique 1 « Les architectures ». (Edition ellipses).
Paris/France.
- [4] Wisama Khalil, Etienne Dombre. (1999). Modélisation identification et commande des
robots. (Hermes Science publications). Paris/France.
- [5] palettisation et contrôle robotisés 15 septembre 2010 Ali YASAR CONSERVATOIRE
NATIONAL DES ARTS ET METIERS CENTRE REGIONAL ASSOCIE DE RHONE ALPES
- [6] HAMISSI Ayoub, Modélisation d'un Robot SCARA à structure complexe (Parallèle) destiné
à manipuler une sonde échographique, UNIVERSITE FERHAT ABBAS - SETIF
UFAS(ALGERIE),
- [7] CHIKH Aissa Etalonnage Géométrique d'un Robot 2018 /2019 UNIVERSITE MOHAMED
BOUDIAF - M'SILA
- [8] T. LATOUR, Modélisation cinématique et dynamique du robot industriel IRB2400 de
ECAM – Bruxelles, 2006
- [9] MARTIN GAUDREAU, ÉTALONNAGE AUTOMATISÉ D'UN ROBOT SÉRIEL,
AVEC UN NOUVEAU DISPOSITIF DE MESURE TRIDIMENSIONNELLE À FAIBLE
COÛT, ÉCOLE DE TECHNOLOGIE SUPÉRIEURE 2016
- [10] Réalisation d'un module de distribution d'énergie à base d'une carte Arduino
Méga 2560 IABBADEN Zinedine et LAHLOU Farid Promotion 2016-2017[10]

- [11] C. Tavernier, « Arduino applications avancées ». Version Dunod.
- [12] 4_elec_chap3-1.pdf
- [13] F.MANDIN/C VRILLEAUD – Lycée Mireille GRENET – COMPIEGNE
- [14] GTE 1 S2 M 7 3 COURS_schéma_elect CHALH
- [15] CHAPITRE VI. Les potentiomètres SYSTEMES TECHNIQUES BEP ELECTRONIQUE 1
ER ANNEE. A. Le potentiomètre
- [16] ch33-les-moteurs-pas-a-pas.pdf
- [17] 08_Arduino_BlueTooth.pdf
- [18] <http://remotexy.com/>
- [19] Article posté dans un blog sous le lien/
<https://www.eagle-robotics.com/blog/4-bras-robot-6-axes-arduino---montage-et-programmation>

Thème : Bras Robot 6 Axes Arduino - Montage et programmation.

*

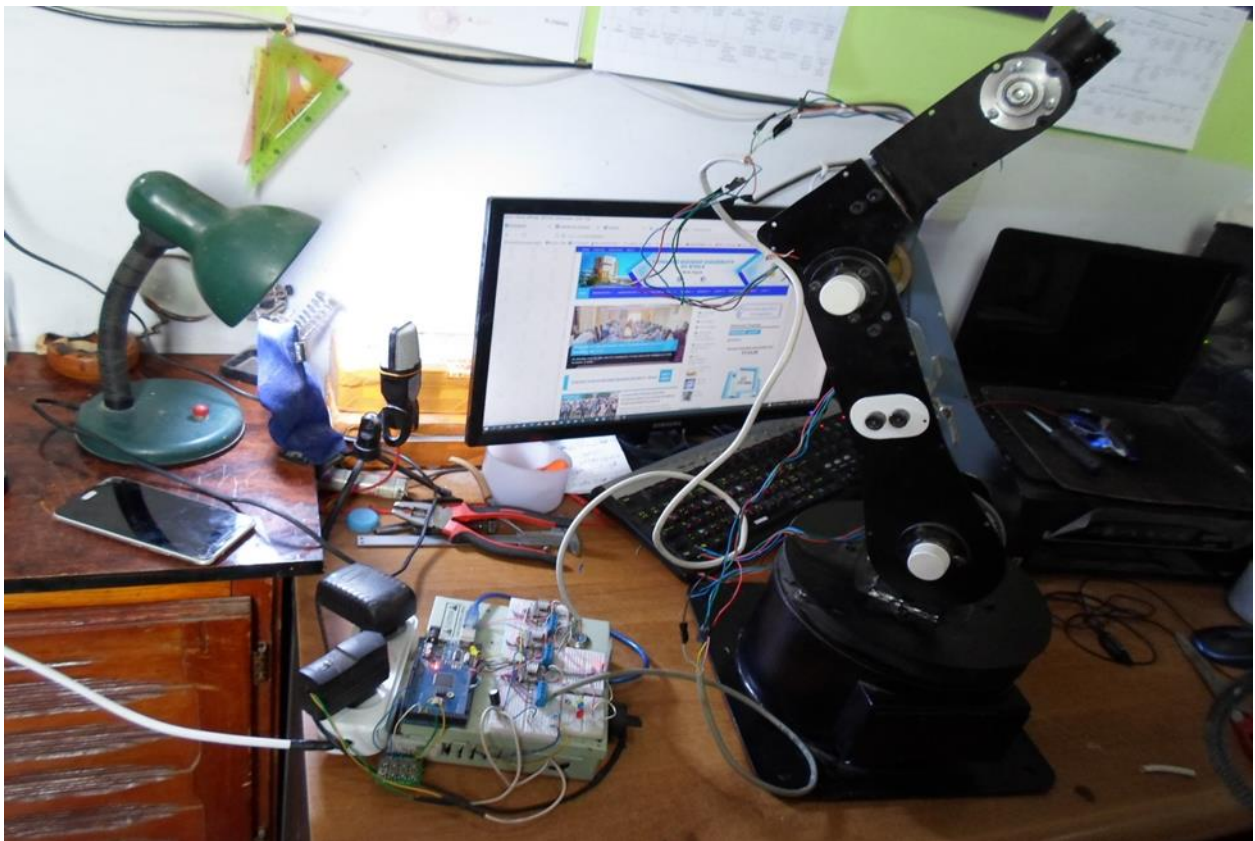
Annexe















les coûts ,dépenses et les frais

Nom d'éléments	Nombre	états	prix
Moteurs à pas G	2	Neuf	840000
Moteurs à pasM	2	Neuf	560000
Moteurs à pas P	1	Neuf	220000
Moteurs à pas Plus P	1	Neuf	120000
Bouteille à gaz	2	ocasion	120000
Pignon de perseuse Type 1 /P	1	ocasion	60000
Tige pignon Type 1/ P	1	ocasion	50000
roulement BC D= 24/d=8	10	Neuf	80000
roulement BC D=40/d=10 pour link2	5	Neuf	200000
roulement BC D= 44/d=12 pour tige de base	2	Neuf	90000
roulement BC D= 28	2	Neuf	70000
roulement BC D= 34	2	Neuf	90000
aliminum 10 caracas disq dur	10	Neuf	200000
baguette soudure	40	Neuf	60000
foret perseuse D= 50	1	Neuf	120000
foret perseuse D= 25	1	Neuf	80000
foret perseuse D = 7	1	Neuf	20000
foret perseuse D = 5	1	Neuf	18000
foret perseuse D= 14	1	Neuf	80000
foret perseuse D= 8		Neuf	20000
Pignon de perseuse Type 2 /G	2	Neuf	200000
Pignon de perseuse type 2/P	1	Neuf	150000
Tige pignon perseuse Type2 /P	2	ocasion	50000
amortisseur occasion	1		50000
Pignon plastique D=62	1		140000
pignon tres petit	1		40000
Coût des éléments pièce de la base (barika)			350000
boite de peinture noir	1		25000
boite de peinture gris	1		25000
Tige filetée G + 2 écroux	1		55000
coût de voyage a setif			200000
disque de tronçonneuse 2 type 1	4		80000
disque de tronçonneuse 2 type 2	1		45000
3 pigons plastiques grands	3		320000
3 courroies	3		180000
3 pignon petits fers	3		150000
12 boulons 08	12		24000
4 boulons 011 pour cap	4		8000
foret perseuse D= 35	1		60000
4 long croux	4		12000
4 boulons 011 pour base	4		10000
4 boulons 011 pour rotor de base	4		10000
6 boulon 10	6		12000
3 boulons 8 pour cage R	3		6000
usinage de tige pignon	1		100000
papier meule	2		20000
foret de filtage 10	1		35000
foret de filtage 11	1		40000
10 divers rondelles	10		80000

Coût de disques N.systeme(2 disques)	2	120000
3 rondelles Grandes pour cages r. (la base)	3	36000
3 rondelles Petites pour cages r.(la base)	3	27000
tige filté 19	1	60000
8 boulons 8 palats	8	20000
Disque couvrt	1	80000
4 boulons 10 pour disque Couvrt	4	10000
4 boulons 10 plats Pour rotor (remédier)	4	10000
4 colle pattes(mastique)	4	40000
bande aluminun	1	15000
Epoxy 1kg		100000
charbon Perseuse	1	20000
6 boulons 10 pliens+8 croux 19	14	30000
2 Support externes link2		100000
Sudure cuiuvre tige D=20		30000
2eme courroie De base 2eme		80000
baguettes soudure d=2	10	8000
4 rondelles de link 2 pour cage de relements	4	40000
petit croie link4+2 pignios	3	100000
12 vis+ecroues 6	12	12000
soudure cuivre supports moteur L(link)2+L3		80000
soudure cuivre pignion moteurs L2+L3		50000
mastic blanche 2	2	20000
boullon 22 +3 croues	2	18000
meche perseuse 3 mm + 4 mm	2	25000
boulons 10 pleins	4	10000
bouteille peinture noir	1	25000
boulons 6 pleines	12	25000
10 bagutttes soudure 2 mm +3 mm	10	15000
meches 6 mm (2eme)	6	15000
bouteille peinture gris	1	25000
bouteuille peinture blanche	1	30000
électrisité		200000
form usiné link5		60000
Totale		711000

Plus les coûts des partie comande électronique

Autour de 30000.00 DA

La somme totl du prix autour de

100000.00DA