الجمهورية الجزائرية الديمقراطية الشعبية

MIISTERE DE L'ENSEIGNEMENT SUPERIUR
ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITE DR. YAHIA FARES DE MEDEA

FACULTE DES SCIENCE ET DE LA TECHNOLOGIE
**Département du Génie Mécanique**



## projet de fin d'études

Présenté pour l'obtention du Diplôme
De Master en Génie Mécanique

*Spécialité : Mécatronique*

# THEME
# ETUDE ET RÉALISATION UN BRAS MANIPULATEUR AVEC UN TAPIS ROTATIF

**Proposé et dirigé par**

Pr. ABUDURA SALAM

**Préparé et présenté par**

M.ABDELLAOUI BRAHIM

Année Universitaire 2016-2017

# REMERCIMENTS

A u nom d'allah le plus grand merci lui revient de nous avoir guider vers le droit chemin, de nous avoir Aidées .

Je tiens remercie ma mère et mon père pour leurs prières accompagnées d'une affection indescriptible, et toutes la famille abdellaoui surtout mes oncles.

Je remercis
tous les enseignants de l université Dr Yahia Fares de médéa de ginie mecanique en particulier Monsieur professeur abudura salam qui guider moi en long de ce projet

chaleur remercier pour les monsieur

baamara amine et baamara ahmad

et bien sur à tous mes chères amis

## الملخص:

هدفنا في هذا المشروع هو دراسة و تحقيق روبوت ذراع آلي مع حزام دوراني, يعمل الروبوت بواسطة محركات ذاتية التحكم ف الزوايا، من خلال لوحة تحكم تدعي الاردوينو الذي يحتوي على لوحةPIC ATMEGA 328



## كلمات مفتاحية:

PIC ATMEGA 328 ، ذراع آلي، سيرفوموتور، ماتريس، برنامج



## RÉSUMÉ:

Dans ce projet notre but est l'étude et la réalisation d'un bras manipulateur avec un tapis rolon commandé par PIC ATMEGA 328 qui intégré sur la carte ARDUINO UNO, le travail de ce robot est par les servomoteur



## Mots-clés

Robot bras manipulateur, PIC ATMEGA 328, matrice programme



## ABSTRACT:

Our goal in this project is the study and the realization robot arm, our robot is controled by PIC ATMEGA 328, the robt movement is ensured by servo-motor.



## Key words:

robot arm, matrix, programme, , PIC ATMEGA 328.

**SOMMAIRE**

**1. Introduction.....1**

## Chapitre I: Introduction générale sur la robotique

**1.1 Historique de la robotique.....2**

**1.2 Buts et domaines d'applications.....2**

**1.3 Composition d'un robot.....3**

**1.3 Les domaines de la robotique.....4**

- ✓ **Mini-robots de précision.....4**
- ✓ **Robots anthropomorphes.....4**
- ✓ **Bras manipulateurs.....5**
- ✓ **Robotique terrestre et aérienne.....5**
- ✓ **Véhicules intelligents.....6**
- ✓ **Robots militaire.....6**
- ✓ **Robotique industrielle.....7**
- ✓ **Définition.....7**
- ✓ **Domaines d'utilisation.....7**
- ✓ **Robots médical.....8**

**1.4 conclusion.....8**

## Chapitre II: Généralité sur les bras manipulateurs

**II -1-Introduction.....10**

**II -2 Description des bras manipulateurs.....10**

**II-2-1 morphologie et Chaîne cinématique d'un bras manipulateur.....10**

**II-2-1-1 Configuration Articulée (RRR).....10**

**II-2-1-2 Configuration Sphérique (RRP).....11**

**II-2-1-3 Configuration SCARA (RRP).....11**

**II-2-1-4 Configuration Cylindrique (RPP).....12**

**II-2-1-5 Configuration Cartésienne (PPP).....12**

**II-2-2 Structure mécanique.....13**

**II-2-3 Espace articulaire, espace opérationnel et modèle de transformation entre les espaces.....14**

**Définition1.....14**

**Définition2.....14**

**II -2-4 degré de liberté et de mobilité.....14**

**Les articulations des robots.....14**

**SOMMAIRE**

**Les chaînes cinématiques.....15**
**Constituants d'un robot.....16**
**II -3 structure géométrique.....17**
- **Modélisation.....17**
  - **II -3-1 Le Modèle Géométrique Direct.....17**
    - **Le MGD (exemple).....17**
    - **solution.....18**
    - **modélisation systématique une chaîne cinématique.....20**
  - **II -3-2 Le Modèle Géométrique inverse.....21**
    - **Calculer le MGI.....21**
    - **Le MGI sur art1/2.....23**
    - **Le MGI sur art 2/2.....23**
    - **la résolution Algébrique 1.....23**

**II -4 conclusion.....24**

## Chapitre III: Généralité sur le microcontrôleur

**III -1-Introduction.....25**
**Schéma simplifié de la carte Arduino UNO.....26**
**Architecture interne du microcontrôleur ATMega.....27**
**III-2 Microcontrôleur ATMEL ATMega328.....27**
**III-2-1 Les Modèles ATMEGA AVR.....28**
**Les principales caractéristiques sont.....28**
**III-2-2 Structure interne de l'ATMega328 (extraits de documentations ATMEL)...29**
**Descriptions des broches.....29**
**L'Horloge Système et Option.....30**
- **L'horloge Interne.....31**
- **L'oscillateur du Timer.....31**

**Le Plan Mémoire.....31**
**La mémoire programme.....31**
**La mémoire de donnée.....32**
**La mémoire morte.....32**
**III-3 Tous les Registres.....33**
**III-4 conclusion.....35**

## Chapitre IV: Réalisation Pratique

**IV.1 Listes des matériaux utiliser.....36**
**IV.1.2 Cahier des charges fonctionnel.....36**
**IV.1.3 Budget de ce projet.....36**
**IV.2 Les Servomoteurs.....37**
**IV.2-1 Introduction.....37**

**SOMMAIRE**


**IV.2-2 Qu'est-ce qu'un servomoteur ?.....37**
**IV.2-3 Fonctionnement d'un servomoteur.....37**
**IV.2-4 L'architecture interne.....38**
**IV.2-5 Fonctionnement d'un servomoteur.....38**
**IV.2-6 Le potentiomètre.....39**
**IV.2-7 La commande du servomoteur.....39**
**IV.3 Moteur courant continu.....40**
**IV.3 -1 Introduction.....40**
**IV.3 -2 Description de la Une machine à courant continu.....41**
**IV.3 -2-1 Stator du moteur courant continu.....41**
**IV.3 -2-2 Rotor du moteur courant continu.....42**
**IV.3 -3 Constitution et principes physiques.....43**
**IV.3 -4Schéma électrique idéalisé.....44**
**IV.3 -4-1 Descriptif du fonctionnement.....45**
- **fonctionnement en moteur.....45**
- **fonctionnement en génératrice.....45**

**IV.3 -5 Avantages et inconvénients.....46**
**IV-4 L'architecture (plan) de ce projet.....47**
**IV-5 Les logiciels de programmation et de simulation.....47**
**IV-5-1 Le logiciel arduino.....47**
- **Introduction.....47**
- **Est un compilateur.....48**

**IV.5-2 Le logiciel ISIS proteus.....49**
**IV.5-2-1 Présentation générale.....49**
**IV.6 partie simulation et programmation.....51**
**IV.6-1 partie simulation.....51**
**IV.6-2 partie programmation.....51**
**IV.6-3: Les programmes.....53**
**Le programme des fonction mouvement.....53**
**Le programme principal.....54**
**IV.7 conclusion.....54**
**conclusion générale.....55**
**références.....56**

## Chapitre I: Introduction générale sur la robotique

figure 1: bras manipulateur.....1
figure 2: Organigramme d'un robot.....3
figure 2: Système de commande et de traitement de l'informatique.....3
figure 3/4:Robot de précision.....4
figure 5: Robots anthropomorphes.....4
figure 6: bras manipulateur.....5
figure 7: Robot volant.....5
figure 8: Espace robot.....5
figure 9: Système d' intelligence des véhicules..... .....6
figure 10: robot militaire forme de char.....6
figure 11 : Des robots industriel.....8
figure 12: Robot d' IRM.....8
figure 13 : Robot médicale.....8

## Chapitre II: Généralité sur les bras manipulateurs

figure II -1 Chaine cinématique d'un bras manipulateur.....10
figure II-2 bras manipulateurs type RRR.....11
Figure II-3 bras manipulateurs type RRP.....11
figure II-4 bras manipulateurs SCARA type RRP.....12
figure II-5 bras manipulateurs type RPP.....12
figure II-6 figure 1-5 bras manipulateurs type PPP.....13
figure II-7 La structure mécanique du robot manipulateur.....13
figure II-8 Vocabulaire d'un robot.....16
figure II-9 Organigramme générale.....17

## Chapitre III: Généralité sur le microcontrôleur

figure III-1: Carte arduino uno.....25
figure III-2 Schéma simplifié de carte arduino.....26
figure III-3 Architecture interne du microcontrôleur ATMega.....27
figure III-4 les broches du microcontrôleur ATMega.....29
figure III-5 Montage du Quartz.....31
figure III-6 Signale d'Horloge Externe.....31
figure III-7 Exemple de connexion d'horloge Temps Réel.....31
figure III- 8, la mémoire de donnée avec la SRAM.....32
figure III-9. Adressage de la mémoire FLASH.....33

## Chapitre IV: Réalisation Pratique

figure IV -1 Servomoteur.....37

**LISTE DES FIGURE**

**figure IV-2 : vue de l'extérieur et de l'intérieur d'un servomoteur .....38**
**figure IV-3 : Signal PWM.....40**
**figure IV-4 : Signal modulé en code d'impulsion.....40**
**figure IV-5 : Dimension du servomoteur.....40**
**figure IV-6 Moteur à courant continu.....40**
**figure IV-7/8 Rotor et stator du moteur cc.....41**
**figure IV-9 L'intérieur du moteur cc .....42**
**figure IV-10 Constitution et principes physiques d'un moteur cc.....43**
**figure IV-11 Le schéma électrique.....44**
**figure IV-12 La fenêtre du logiciel arduino.....48**
**figure IV-13 La fenêtre principale du logiciel isis proteus.....50**
**figure IV-14: Schéma électrique de bras manipulateur.....51**
**figure IV-15: Programme principal de projet.....52**
**figure IV-16: Programme des fonction mouvements.....52**

## 1. Introduction:

La Robotique fait partie des sciences des objets et des systèmes artificiels. Elle peut être vue comme la science de la perception et du mouvement et de leur intégration en une machine physique, mécanique et informatique.
Un robot est donc un système matériel possédant des capacités de perception, d'action, de décision et de communication, parfois capable d'améliorer ses propres performances par apprentissage automatique ou supervisé par des humains, pour :

- agir dans un environnement ouvert ou confiné, dynamique et imparfaitement modélisé, voire inconnu, à des échelles allant du nano monde au macro monde ;
- exécuter de façon autonome ou en relation avec un humain, des tâches d'observation, d'exploration, de modélisation, de manipulation et/ou d'intervention sur l'environnement ;
- interagir le cas échéant avec d'autres machines ou avec des humains, matériellement ou virtuellement.

Au plan scientifique, la Robotique est dotée de plusieurs grands journaux internationaux (IEEE Trans. on Robotics 3, Int. Journal of Robotics Research, Int. Journal of Robotics and Autonomous Systems) (à moindre titre : Int. Journal of Advanced Robotics Systems, Int. Journal
of Robotics and Mecatronics,) et de plusieurs conférences internationales dont les deux plus importantes: IEEE Int. Conf. on Robotics and Automation, IEEE/RSJ Int. Conf. on Intelligent Robots and Systems.
Si la Robotique non manufacturière a été longtemps considérée comme un champ scientifique très prospectif dont les applications, en dehors de la conquête spatiale, semblaient futuristes, les progrès scientifiques et technologiques importants intervenus dans les grandes disciplines invoquées par la Robotique (mécanique, électronique, automatique, informatique) ont considérablement ouvert le spectre de ses applications, de plus en plus tangibles et crédibles.



figure 1: bras manipulateur

# Introduction générale sur la robotique

## 1.1 Historique de la robotique

L'origine du mot robot provient de la langue tchèque dans laquelle sont ancêtre "**robota**" signifie **travail forcé**. Le terme de robotique quant à lui est apparu en 1942 dans le cycle universellement connu rédigé par *Isaac Asimov* et intitulé "**Les robots**".
Voici une définition concise de ce que peut être un robot :
" Machine programmable qui imite des actions d'une créature intelligente."
Au cours de l'histoire on peut distinguer 3 types de robots correspondant en quelques sorte à l'évolution de cette "espèce" créée par l'Homme.
Le **premier type** de machine que l'on peut appeler robot correspond aux "**Automates**". Ceux-ci sont généralement programmés à l'avance et permettent d'effectuer des actions répétitives.
Le **second type** de robot correspond à ceux qui sont équipés de capteurs (en fait les sens du robot). On trouve des capteurs de température, photo électronique, à ultrasons pour par exemple éviter les obstacles et/ou suivre une trajectoire. Ces capteurs vont permettent au robot une relative adaptation à son environnement afin de prendre en compte des paramètres aléatoires qui n'aurait pu être envisagés lors de leur programmation initiale. Ces robots sont donc bien plus autonomes que les automates mais nécessitent un investissement en temps de conception et en argent plus conséquent.
Enfin le **dernier type** de robot existant correspond à ceux disposant d'une intelligence dite "artificielle" et reposant sur des modèles mathématiques complexes tels que les réseaux de neurones. En plus de capteurs physiques comme leurs prédécesseurs, ces robots peuvent prendre des décisions beaucoup plus complexes et s'appuient également sur un apprentissage de leurs erreurs comme peut le faire l'être humain. Bien sûr il faudra attendre encore longtemps avant que le plus "intelligent" des robots ne soit égal, tant par sa faculté d'adaptation que par sa prise de décisions, à l'Homme.

## 1.2 Buts et domaines d'applications

Le but premier d'un robot est d'effectuer des tâches répétitives et/ou précises. Les robots permettent également d'effectuer des tâches dans des environnements de travail trop dangereux pour l'Homme.
Les robots peuvent effectuer des tâches automatiques, mais certains sont aussi dotés d'une certaine intelligence. Leurs facultés d'adaptation nécessitent des qualités de perception dans le but d'interagir avec leur environnement.
Les robots sont généralement présent dans différents domaines d'application tels que la manipulation d'objets, la soudure, l'assemblage de pièces dans l'industrie automobile par exemple.
Les robots sont donc utilisés dans les usines mais également dans les laboratoires pharmaceutiques ou encore dans les hôpitaux.
Notons toutefois que l'**industrie automobile** utilise plus de **50 % de l'ensemble des robots industriels**.

## 1.3 Composition d'un robot

Un robot est composé de **quatres parties** principales. En voici la description:
Une **structure mécanique** qui sera le squelette du robot. Une attention particulière doit être portée aux articulations car celles-ci doivent permettent un débattement assez important relatif à l'utilisation voulue.
Le **second élément** correspond aux **servomoteurs** qui vont permettent au robot d'effectuer réellement ses actions. Ces servomoteurs seront commandés par la partie commande en interaction avec les informations transmises par les capteurs. Le terme "servo" induit en effet un asservissement effectué en fonction d'une comparaison avec le résultat souhaité et la réalité extérieure.
La **troisième partie** composante d'un robot correspond aux différents **capteurs** sensoriels équipant le robot pour une application particulière.
Enfin le cerveau : La **partie commande**. C'est cette partie qui va permettent au robot d'analyser les données provenant des capteurs et d'envoyer les ordres relatifs aux servomoteurs. La partie commande est matérialisée physiquement par le **microcontrôleur**.

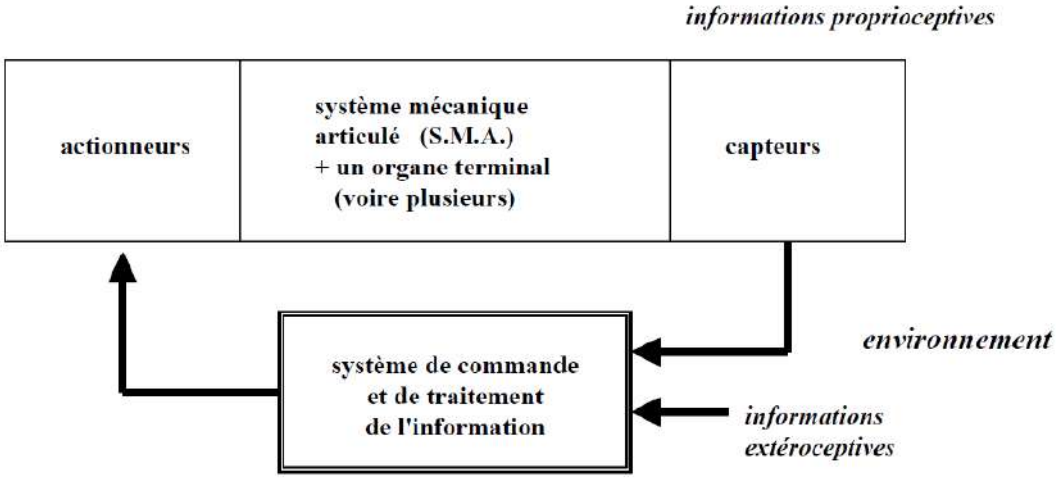



figure 2: Organigramme d'un robot

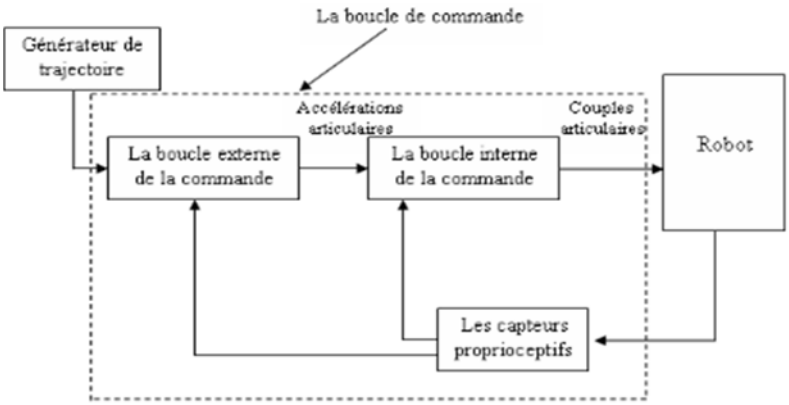



figure 2: Système de commande et de traitement de l'informatique

## 1.3 Les domaines de la robotique:

On en trouve ci-dessous un résumé d'analyses de quelques domaines particuliers ou applications de la robotique :

- ✓ **Mini-robots de précision** : ils sont conçus spécialement pour les applications de micro montage et les installations très compactes difficiles à manipuler par l'homme. Ils sont rapides, précis et très fiables destinés à répondre à toutes les contraintes posées par les opérations de précision de positionnement et d'assemblage en micromécanique





figure 3/4:Robot de précision

- ✓ **Robots anthropomorphes :**
  ils sont des robots a forme humaine pour remplacer l'homme, et très souvent autonomes. Ils sont équipés de capteurs perfectionnés; tels que caméras, télémètres laser, organes de perception de force, etc. Ils disposent de techniques avancées de reconnaissance vocale, reconnaissance de formes, traitement d'images et derniers développements en informatique (Intelligence Artificielle).



figure 5: Robots anthropomorphes

- ✓ **Bras manipulateurs :**
  ils sont les équipements les plus utilisés au milieu industriel, ils sont conçus pour saisir et manipuler des objets avec ou sans assistance humaine. Ils réalisent des tâches répétitives et souvent pénibles pour l'homme. Ils sont rapides, précis, fidèles et manipulent des objets au milieu inaccessible, ou des produits dangereux ou au milieu nocif pour l'être humain.



figure 6: bras manipulateur

- ✓ **Robotique terrestre et aérienne :**
  Les robots considérés ici se déplacent dans des environnements peu ou pas structurés, souvent naturels, et peu ou pas connus/modélisés, et se meuvent seuls ou en groupe coopératif.



figure 7: Robot volant



figure 8: Espace robot

✓ **Véhicules intelligents:**

Les véhicules considérés ici sont ceux dédiés au transport des personnes ou des marchandises. On peut distinguer les véhicules routiers (autoroutes et routes) et les véhicules urbains (avenues et rues).




figure 9: Système d' intelligence des véhicules

✓ **Robots militaire:**

Un **robot militaire**, aussi appelé **arme autonome**, est un robot, autonome ou contrôlé à distance, conçu pour des applications militaires. Les drones sont une sous-classe des robots militaires.

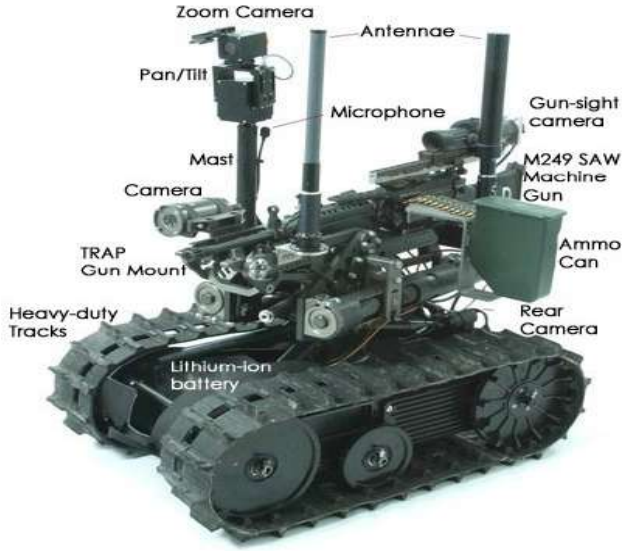



figure 10: robot militaire forme de char

✓ **Robotique industrielle :**

La **robotique industrielle** est officiellement définie par l'ISO comme un contrôle automatique, reprogrammable, polyvalent manipulateur programmable dans trois ou plusieurs axes. Les applications typiques incluent des robots de soudage, de peinture et d'assemblage. La robotique industrielle inspecte les produits, rapidement et précisément. Les robots industriels sont très utilisés en automobile. Leur conception nécessite une bonne connaissance et un très haut niveau dans le domaine de l'ingénierie.

**Définition:**

Un robot industriel est un système ayant plusieurs bras à l'image d'un bras humain souvent composé de six degrés de liberté, trois axes destinés au positionnement et trois axes à l'orientation permettant de déplacer et d'orienter un outil (organe effecteur) dans un espace de travail donné.

On peut distinguer :

- les robots de montage de dimension souvent plus réduite,
- les robots mobiles destinés à l'inspection souvent associés à de l'intelligence artificielle et capables, dans certains cas, de prendre en compte l'environnement.

Un robot se compose d'une partie mécanique, le bras lui-même, d'une armoire de commande composée d'une unité centrale qui pilote les électroniques de commande d'un ou plusieurs axes qui en assure l'asservissement, de variateurs de vitesse et d'un langage de programmation spécialisé qui permet de commander le robot (LM[1] développé par l'Ensimag Grenoble, langage Adept type basic) qui intègre un transformateur de coordonnées pour transformer une valeur cartésienne en données codeur du moteur.

Certains robots disposent d'un mode d'apprentissage qui permet de répéter les mouvements réalisés librement à la main, l'élément essentiel étant la fidélité[2] la capacité du robot à atteindre successivement la même position dans une tolérance définie, une procédure d'étalonnage permet de reprendre le zéro de chacun des axes. Ils peuvent être associés à un système de vision artificielle qui leur permet de corriger les déplacements. Pour des raisons de sécurité, ces robots sont protégés par des cages ou des carters pour interdire à l'homme de les approcher de trop près.

La diffusion et amélioration des techniques robotiques permet de faire fonctionner des usines dans le noir.

**Domaines d'utilisation:**

Les robots industriels ont d'abord été développés pour intervenir dans les milieux à risques (nucléaire, forte corrosion...) puis dans l'automobile avec le robot Unimate, le premier robot industriel de la société américaine Unimation, au début des années 1960. Ils servent aussi beaucoup dans le maniement d'objets lourds.

Le petit assemblage de précision sur des petites séries.



figure 11 : Des robots industriel

- ✓ **Robots médical:**

  Un **robot médical** est un système robotique utilisé dans le cadre d'une application thérapeutique, par exemple lors d'une chirurgie ou au cours d'un programme de réhabilitation neuromatrice. Du fait des contraintes importantes en matière de sécurité, ce type de robot est en général doté d'un faible niveau d'autonomie.



figure 12: Robot d' IRM



figure 13 : Robot médicale

## 1.4 conclusion:

Nous avons présenté dans ce chapitre l'introduction générale de la robotique, et on a découvert les différents domaines d'application et les buts de la robotique, et nous avons à connaitre les compositions d'un robot et leur système de commande et de traitement de l'informatique .

Les robots qui sont remplacent l'homme dans les taches pénible et dangereuses, les robots sont utiliser à jouer un rôle de plus en plus important dans notre vie.

# Généralité sur les bras manipulateurs

## II-1-Introduction

Quel que soit l'objectif qu'on se fixe, concevoir, simuler ou commander un système mécanique, il est nécessaire, entre autres, de disposer de modèles du mécanisme à traiter. Plusieurs niveaux de modélisation sont possibles, qui dépendent des spécifications du cahier des charges de l'application envisagée : modèles géométriques, cinématiques ou dynamiques à partir desquels peuvent être engendrés les mouvements du robot, ou bien des modèles statiques qui décrivent les interactions du mécanisme avec son environnement.

## II-2 Description des bras manipulateurs

### II-2-1 morphologie et Chaîne cinématique d'un bras manipulateur

On suppose les bras manipulateurs constitués de n corps mobiles, supposés parfaitement rigides, reliés entre eux par n liaisons rotoïdes et/ou prismatiques formant une structure de chaîne simple (voir figure I.1).

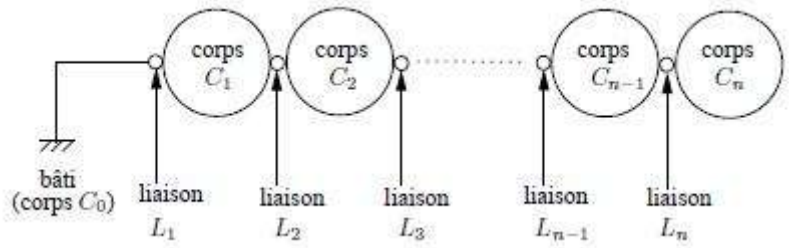



figure II-1 Chaine cinématique d'un bras manipulateur

Pour identifier la nature de la i-ème liaison du bras manipulateur, on définit le paramètre :

$$\sigma i = \begin{cases} 0, \text{ pour une liaison rotoïdes,} \\ 1, \text{ pour une liaison prismatique.} \end{cases}$$

On désigne fréquemment les bras manipulateurs en accolant les lettres R (pour rotoïdes) et P (pour prismatique) pour décrire la succession des liaisons. Par exemple un bras manipulateur de type anthropomorphe sera désigné par le sigle 6R alors qu'un bras manipulateur de type SCARA sera de type RRP ou RRPR (il existe plusieurs variantes) .

### II-2-1-1 Configuration Articulée (RRR)

Le manipulateur articulé appelé aussi le manipulateur rotoïde, ou anthropomorphe est un design des plus utilisé, la structure terminologie associé a ce type de manipulateur est montré par la figure I-2:

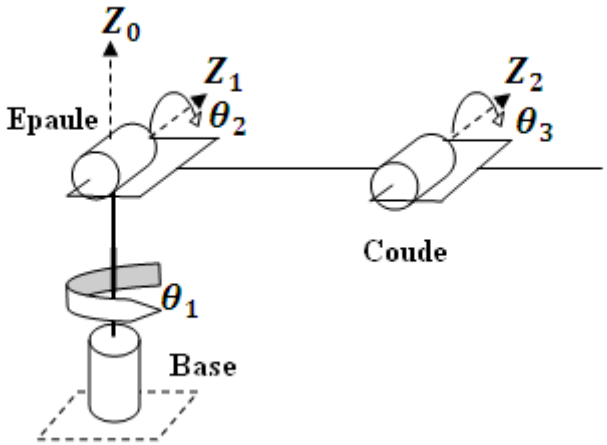



figure II-2 bras manipulateurs type RRR

### II-2-1-2 Configuration Sphérique (RRP)

En remplaçant la troisième articulation (le coude) dans la configuration rotoïde par une prismatique on obtient la configuration sphérique montrée dans la Figure 1-3. Le terme configuration sphérique dérive du fait que les coordonnées sphériques définissant la position de l'organe terminal, en respectant que l'origine de la base coïncide avec l'intersection des deux axes $Z_1$ et $Z_2$ , sont les mêmes que ceux des trois premières articulations.

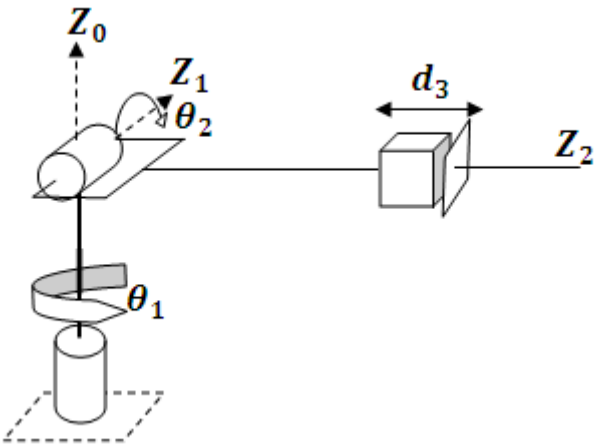



Figure II-3 bras manipulateurs type RRP

### II -2-1-3 Configuration SCARA (RRP)

SCARA (pour Selective Compliant Articulated Robot for Assembly) montré dans la figure I-4 est une configuration très connue. Comme son nom sous-entend il est spécialement conçu pour les opérations d'assemblage. Quoique le robot SCARA a une structure RRP, il est tout à fait différent de la configuration sphérique ni dans l'apparence ni dans le domaine

d'application. A la différence du design sphérique qui a $Z_0$, $Z_1$, $Z_3$ mutuellement perpendiculaires, le robot SCARA a $Z_0$, $Z_1$, $Z_3$ parallèles.

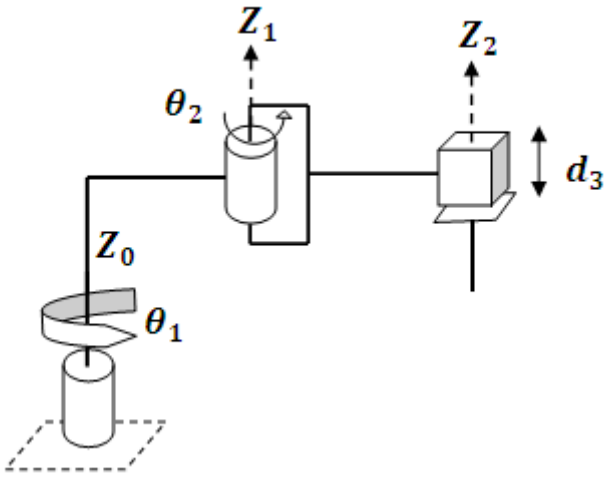



figure II-4 bras manipulateurs SCARA type RRP

### II-2-1-4 Configuration Cylindrique (RPP)

La configuration cylindrique est montrée dans la figure 1-5. La première articulation est rotoïde et produit une rotation autour de la base, alors que la seconde et la troisième sont prismatiques. Comme le nom le suggère, les variables articulaires sont les coordonnées cylindriques de l'organe terminal en prenant compte de la base.

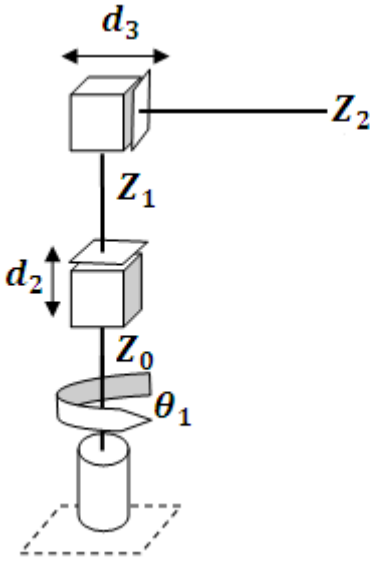



figure II-5 bras manipulateurs type RPP

### II-2-1-5 Configuration Cartésienne (PPP)

Un manipulateur dont les trois premières articulations sont prismatiques est connu sous le

nom de manipulateur cartésien. Il est montré dans la figure I-6. Pour le manipulateur cartésien les variables articulaires sont les cartésiennes de l'organe
terminal en respectant la base. Comme on peut le deviner la description cinématique de ce manipulateur est la plus simple de toutes les configurations.

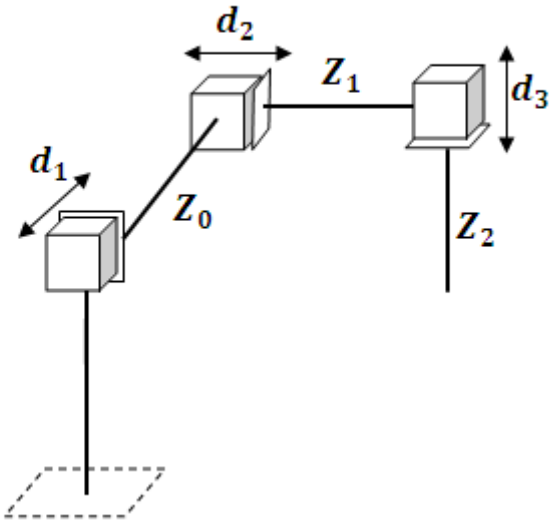



figure II-6 bras manipulateurs type PPP

## II-2-2 Structure mécanique

La structure mécanique du robot manipulateur peut être divisée en deux parties distinctes comme le montre la figure I-7 :

- **Organe terminal** : les tâches qui sont dévolues aux robots sont très variées. Pour chaque opération ou travail spécifique, l'organe terminal prend un aspect particulier.
- **Elément porteur** : il est composé d'un ensemble de corps souples ou rigides liés par des articulations, servant à déplacer l'organe terminal d'une configuration à une autre.

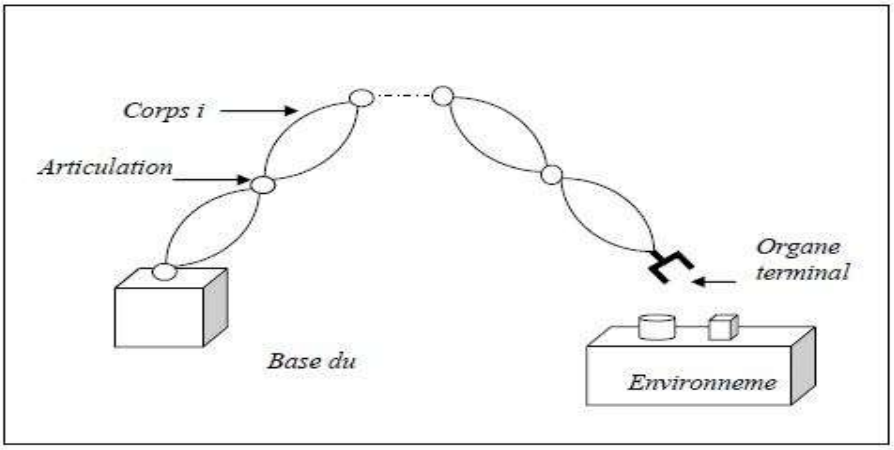



figure II-7 La structure mécanique du robot manipulateur

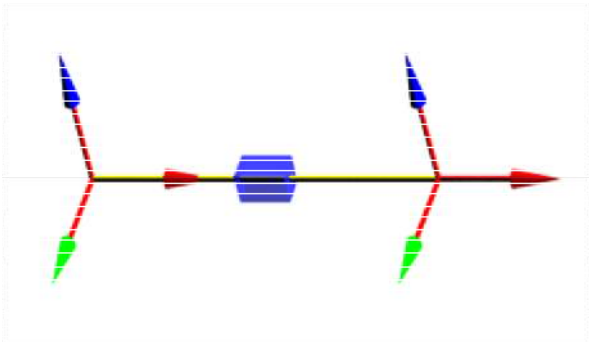
## II-2-3 Espace articulaire, espace opérationnel et modèle de transformation entre les espaces

**Définition1** : on appelle espace de configuration articulaire d'un robot manipulateur (ou simplement configuration) l'état du robot représentant la situation de ses différents corps. Les variables ou coordonnées articulaires sont aussi appelées coordonnées généralisées. L'espace de ces variables noté NB, est appelé espace articulaire ou espace des configurations.
**Définition2** : l'espace opérationnel est celui dans lequel est représentée la situation de l'organe terminal. Les variables ou coordonnées au nombre minimum permettant de définir la situation de l'organe terminale sont appelées les coordonnées opérationnelles. L'espace opérationnel de dimension m est noté MB.

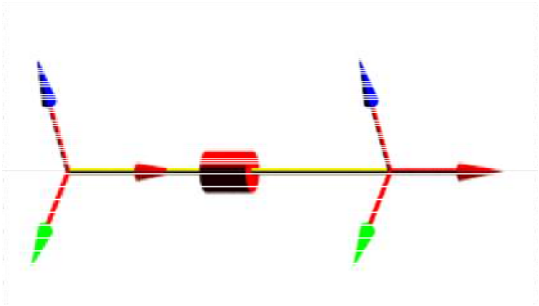
## II-2-4 degré de liberté et de mobilité

- **Degré de liberté** : nombre de paramètres utilisés pour spécifier la configuration d'un élément de la chaîne cinématique par rapport à un autre
- **Degré de mobilité** : toute articulation est caractérisée par son degré de mobilité m, c'est-à-dire le nombre de degrés de liberté entre deux corps successifs de la chaîne cinématique ($0 \leq m \leq 6$)

## Les articulations des robots

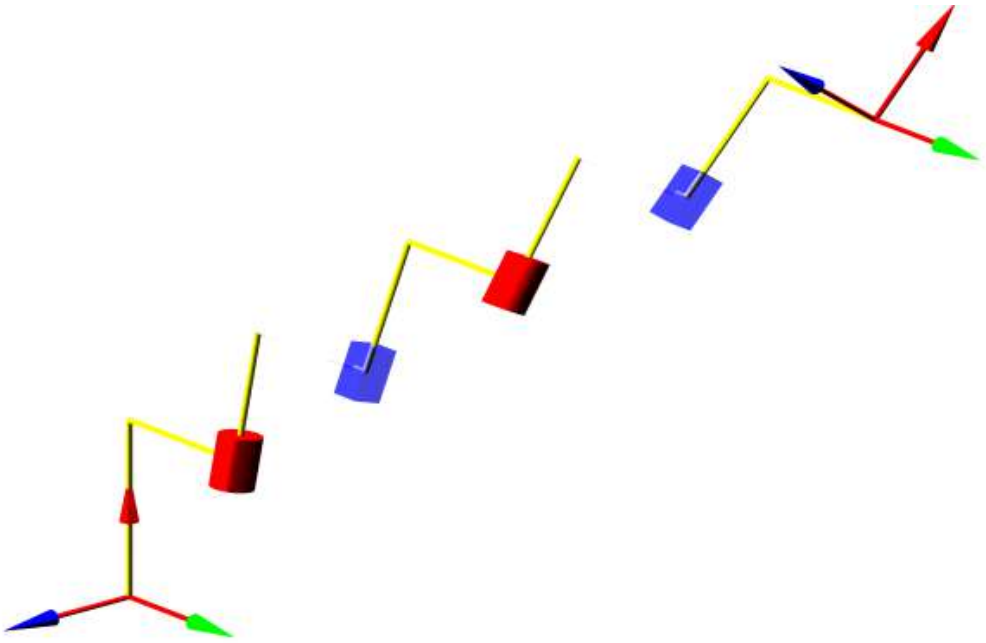


Articulation prismatique, noté **P**
1 ddl en translation T z .
Valeur articulaire q = longueur [m].



Articulation rotoïde, noté **R**
1 ddl en rotation R z .
Valeur articulaire q = angle [rad],[ ◦ ].

## Les chaînes cinématiques



Une chaîne cinématique sera définie par une succession d'articulations rotoïdes ou prismatiques.

## Constituants d'un robot

### Vocabulaire

- Actionneur, moteur
- Axe, articulation
- Corps, segment
- Organe terminal
- Effecteur, outil
- Base



figure II-8 Vocabulaire d'un robot

- Coordonnées généralisées **X = [P,R]**
  (position **P** / orientation **R** de l'organe terminal)
- Coordonnées articulaires **q**
  (consignes données aux moteurs : soit rotation autour d'un axe soit translation suivant un axe)
- Paramètres géométriques ζ qui définissent de façon statique les dimensions du robot

## II-3 structure géométrique

## Modélisation

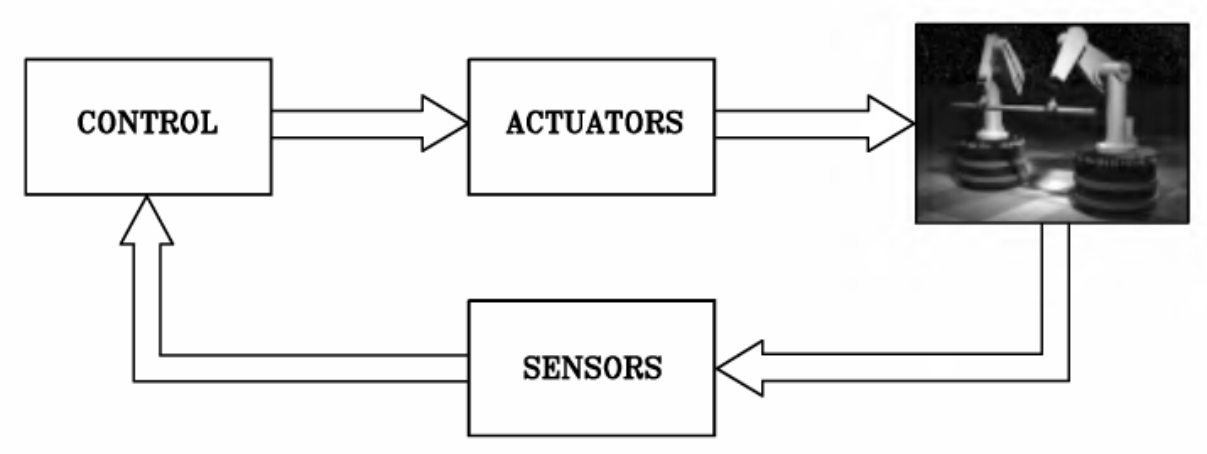



figure II-9 Organigramme générale

### II-3-1 Le Modèle Géométrique Direct

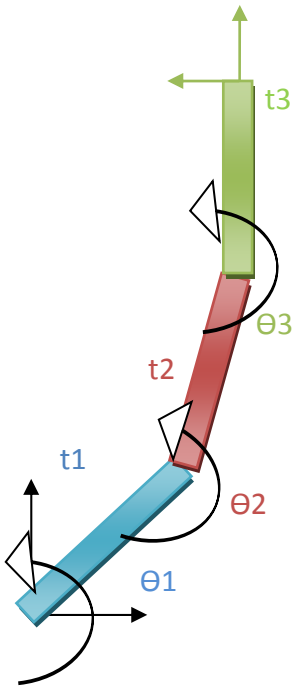
**Déterminer**: Les coordonnées généralisées **(X)** en fonction des coordonnées articulaires **(q):**

$$X = F \text{ MGD } (q_1 , q_2 , \ldots, q_n , \zeta)$$

avec ζ les paramètres géométriques (paramètres qui définissent la géométrie du robot).

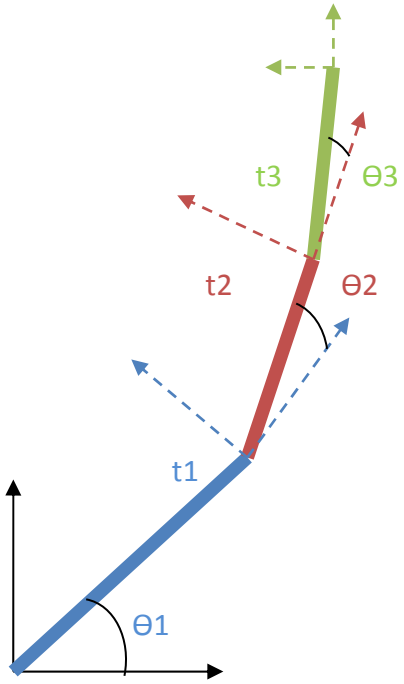
### Le MGD (exemple)

- Identifier les coordonnées articulaires
- Identifier les paramètres géométriques qui définissent le mécanisme
- Associer à chacune des articulations un repère
- Déterminer le positionnement (matrice R, vecteur P) de chaque repères par rapport au précédent.
- Mettre ces changements de repères sous
- la forme de matrices homogènes Montrer comment calculer le MGD de ce mécanisme

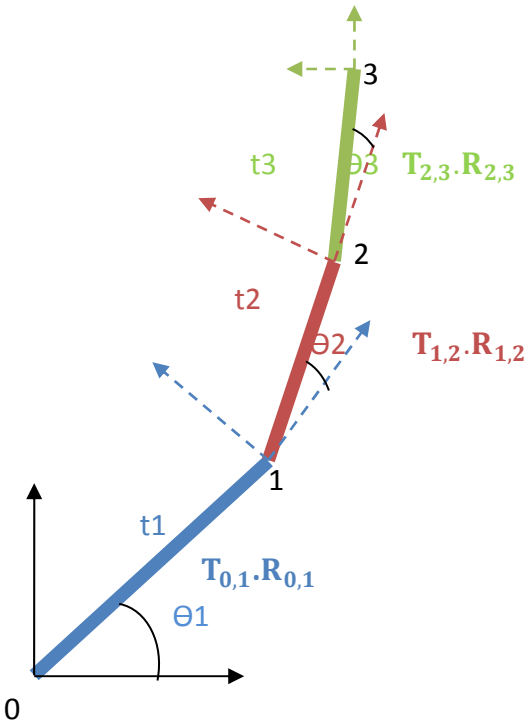


**solution**

- Identifier les coordonnées articulaires
  Solution: q 1 = θ 1 , q 2 = θ 2 , q 3 = θ 3
- Identifier les paramètres géométriques qui définissent le mécanisme
  Solution: ζ = {t 1 ,t 2 ,t 3 }
- Associer à chacune des articulations un repère



- Déterminer le positionnement (matrice R, vecteur P) de chaque repères par rapport au précedent.




$\mathbf{R_{i,j}} = \begin{pmatrix} cos\theta_j & -sin\theta_j \\ sin\theta_j & cos\theta_j \end{pmatrix}$ ; $\mathbf{T_{i,j}} = \begin{pmatrix} t_j \times cos\theta_j \\ t_j \times sin\theta_j \end{pmatrix}$ ; $\mathbf{i \in 0,1,2\ /\ j \in 1,2,3}$

- Mettre ces changements de repères sous la forme de matrice homogène

$$H_{i,j} = \begin{pmatrix} \mathbf{R_{i,j}} & \mathbf{T_{i,j}} \\ 0\ 0 & 1 \end{pmatrix}$$

- Montrer comment calculer le MGD de ce mécanisme $H_{0,3} =$

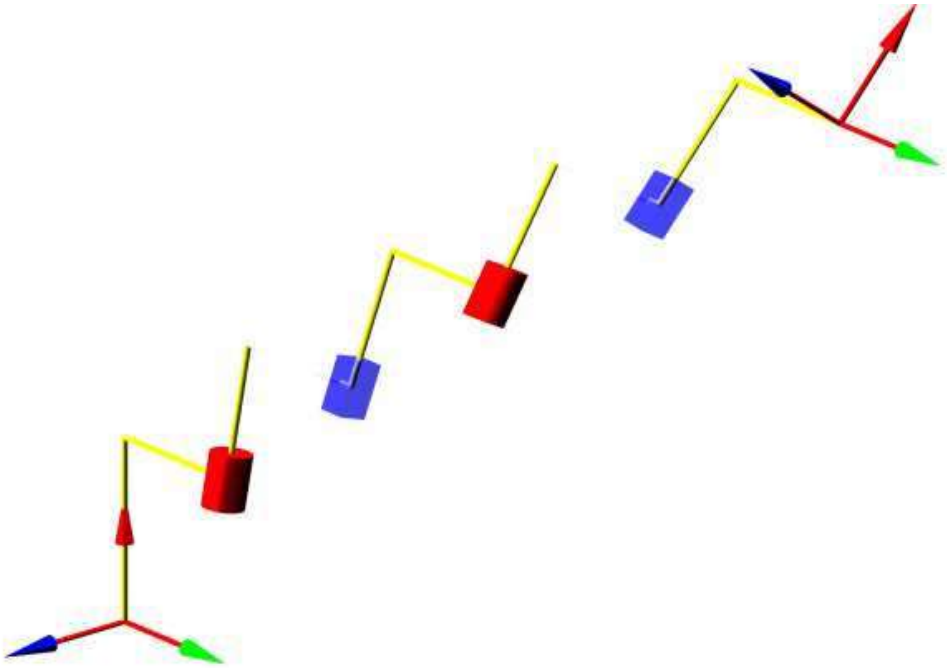
$$\begin{pmatrix} cos\theta_1 & -sin\theta_1 & t_1 \times cos\theta_1 \\ sin\theta_1 & cos\theta_1 & t_1 \times sin\theta_1 \\ 0 & 0 & 1 \end{pmatrix} \times \begin{pmatrix} cos\theta_2 & -sin\theta_2 & t_1 \times cos\theta_1 \\ sin\theta_2 & cos\theta_2 & t_2 \times sin\theta_2 \\ 0 & 0 & 1 \end{pmatrix} \times \begin{pmatrix} cos\theta_3 & -sin\theta_3 & t_3 \times cos\theta_3 \\ sin\theta_3 & cos\theta_3 & t_3 \times sin\theta_3 \\ 0 & 0 & 1 \end{pmatrix}$$

$$=\begin{pmatrix} \cos(\theta_1+\theta_2+\theta_3) & -sin(\theta_1+\theta_2+\theta_3) & t_1.cos\theta_1+t_2.\cos(\theta_1+\theta_2)+t_3.\cos(\theta_1+\theta_2+\theta_3) \\ sin(\theta_1+\theta_2+\theta_3) & \cos(\theta_1+\theta_2+\theta_3) & t_1.sin\theta_1+t_2.sin(\theta_1+\theta_2)+t_3.sin(\theta_1+\theta_2+\theta_3) \\ 0 & 0 & 1 \end{pmatrix}$$

$$\mathbf{X} = \begin{pmatrix} t_1.cos\theta_1+t_2.\cos(\theta_1+\theta_2)+t_3.\cos(\theta_1+\theta_2+\theta_3) \\ t_1.sin\theta_1+t_2.sin(\theta_1+\theta_2)+t_3.sin(\theta_1+\theta_2+\theta_3) \\ \theta_1+\theta_2+\theta_3 \end{pmatrix}$$

Repère mobile

$$X=\begin{pmatrix} R & P \\ 000 & 1 \end{pmatrix}$$



Repère de base

**modélisation systèmatique une chaîne cinématique**

Dans l'espace, nous utiliserons le formalisme de **Denavit-Hartenberg**

1 Placer les repères

2 Définir les variables articulaires et les paramètres géométriques

3 Définir les matrices de transformées homogènes

4 Multiplier ces matrices

Déterminer:

$$X = F \ \text{MGD}(q_1, q_2, \ldots, q_n, \zeta)$$

La transformation homogène entre le repère $\Omega_0$ et le repère mobile $\Omega$ n est obtenue telle que :

$$H_{DH} = H_0 . H_1 \ldots H_n$$

Il faut projeter $H_{DH}$ sur X = [T x ,T y ,T z ,R x ,R y ,R z ]

Nous souhaitons obtenir X = [T x ,T y ,T z ,R x ,R y ,R z ] en fonction des éléments de la matrice $H_{DH}$ . **Pour la position ...**

$$\begin{pmatrix} T\,x \\ T\,y \\ T\,z \end{pmatrix} = \begin{pmatrix} H_{DH_{1,4}} \\ H_{DH_{2;4}} \\ H_{DH_{3,4}} \end{pmatrix}$$

Nous souhaitons obtenir X = [T x ,T y ,T z ,R x ,R y ,R z ] en fonction des éléments de la matrice H DH . Pour l'orientation ... Sachant que :

$$R = \begin{pmatrix} cos\theta cos\varphi & -cos\theta sin\varphi & sin\theta \\ sin\varphi sin\theta\, cos\psi + cos\varphi sin\psi & cos\varphi cos\psi - sin\varphi sin\theta\, sin\psi & -sin\varphi cos\theta \\ -cos\varphi sin\theta\, cos\psi + sin\varphi sin\psi & cos\varphi sin\theta\, sin\psi + sin\varphi cos\psi & cos\varphi cos\theta \end{pmatrix}$$

$$R_x = \arctan \frac{H_{DH_{3,2}} . H_{DH_{1,1}} - H_{DH_{3,1}} . H_{DH_{1,2}}}{H_{DH_{1,1}} . H_{DH_{2,2}} - H_{DH_{1,2}} . H_{DH_{2,1}}}$$

$$R_y = \arctan \frac{H^2{}_{DH_{1,3}}}{\sqrt{H^2{}_{DH_{1,1}} + H^2{}_{DH_{1,2}} + H^2{}_{DH_{2,3}} + H^2{}_{DH_{3,3}}}}$$

$$R_z = \arctan \frac{H_{DH_{2,3}} . H_{DH_{3,1}} - H_{DH_{2,1}} . H_{DH_{3,3}}}{H_{DH_{2,3}} . H_{DH_{3,2}} - H_{DH_{2,2}} . H_{DH_{3,3}}}$$

## II-3-2 Le Modèle Géométrique inverse

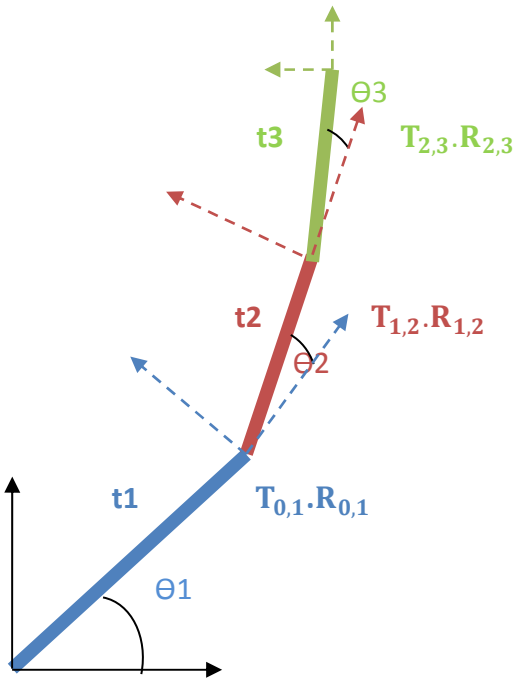
Le MGD du robot permet de calculer les coordonnées opérationnelles en fonction des coordonnées articulaires. Le problème inverse consiste à calculer les coordonnées articulaires qui amènent l'organe terminal dans une situation désirée, spécifiée par ses coordonnées opérationnelles. Cette opération est très souvent appelée transformation de coordonnée (ou changement de coordonnées). Lorsqu'elle existe, la forme explicite qui donne toutes les solutions possibles au problème inverse constitue ce que nous appelons le modèle géométrique inverse (MGI), le modèle s'écrit

$$\mathbf{q = f^{-1}(x)}$$

on a Déterminer:

$$(q_1 , q_2 , \ldots, q_n) = F \ \text{MGI}(X, \zeta)$$

avec ζ es paramètres géométriques (paramètres qui définissent la géométrie du robot série).



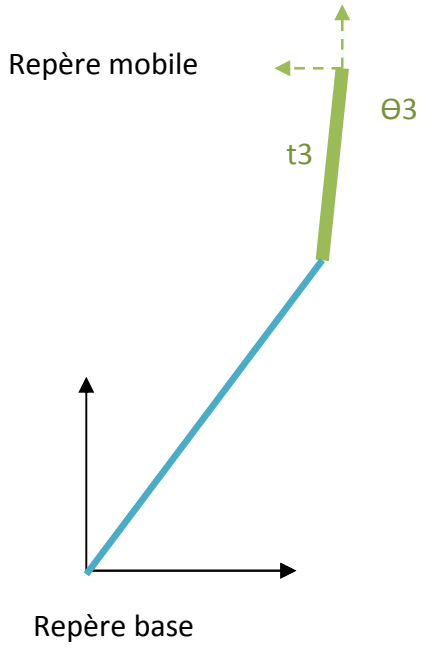

$$\mathbf{X} = \begin{pmatrix} t_1.cos\theta_1 + t_2.\cos(\theta_1 + \theta_2) + t_3.\cos(\theta_1 + \theta_2 + \theta_3) \\ t_1.sin\theta_1 + t_2.sin(\theta_1 + \theta_2) + t_3.sin(\theta_1 + \theta_2 + \theta_3) \\ \theta_1 + \theta_2 + \theta_3 \end{pmatrix}$$

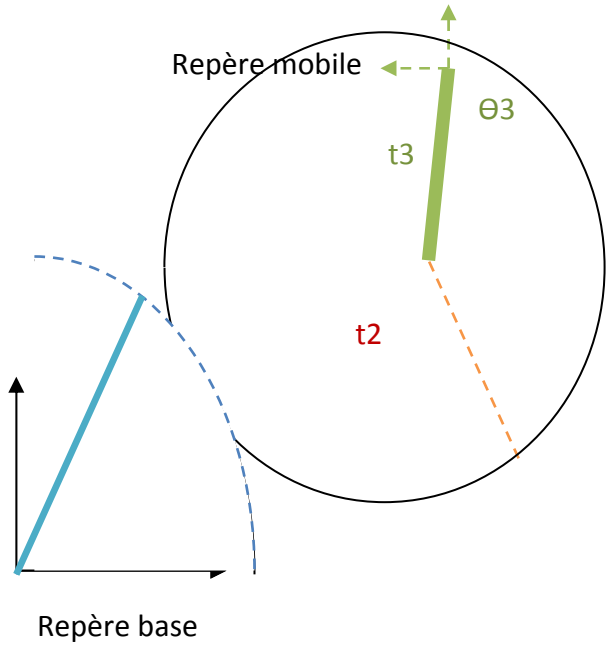
Calculer le **MGI**, c'est déterminer:

$$[\theta_1, \theta_2, \theta_2] = F \ \text{MGI}\ (X_1 , X_2 , X_3 , \zeta)$$

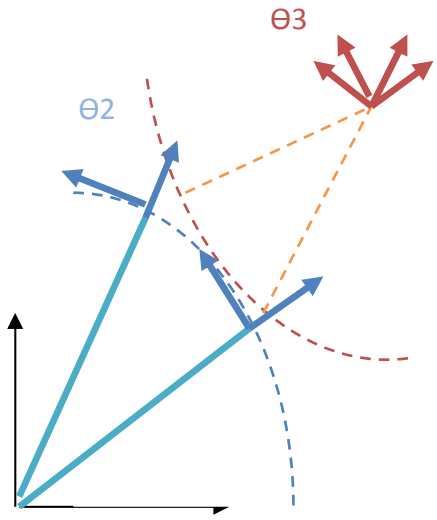
avec $\zeta = [t_1, t_2, t_3]$

**Le MGI sur art1/2**




**Le MGI sur art 2/2**




**la résolution Algébrique 1**

$$t_1.cos\theta_1 + t_2.\cos(\theta_1 + \theta_2) + t_3.\cos(\theta_1 + \theta_2 + \theta_3) - X_1 = 0$$
$$t_1.sin\theta_1 + t_2.sin(\theta_1 + \theta_2) + t_3.sin(\theta_1 + \theta_2 + \theta_3) - X_2 = 0$$
$$\theta_1 + \theta_2 + \theta_3 = X_3$$

$$t_1.cos\theta_1 + t_2.\cos(\theta_1 + \theta_2) + t_3.cosX_3 - X_1 = 0$$

$$t_1.sin\theta_1 + t_2.sin(\theta_1 + \theta_2) + t_3.sinX_3 - X_2 = 0$$

eq(1):

$$t_1.cos\theta_1 + t_2.\cos(\theta_1 + \theta_2) = \mu 1$$
$$t_1.sin\theta_1 + t_2.sin(\theta_1 + \theta_2) = \mu 2$$

On sait que:

$$\cos^2(\theta_1 + \theta_2) + \sin^2(\theta_1 + \theta_2) = 1 \text{ ..... 2}$$

En reportant, les équations 1 dans l'équation 2.

$$(\mu 1 - t_1.cos\theta_1)^2 + (\mu 2 - t_1.sin\theta_1)^2 = t_2^2$$

Nous obtenons:

$$(\mu 1.cos\theta_1 + \mu 2.sin\theta_1 = \frac{t_1^2 - t_2^2 + \mu_1^2 + \mu_2^2}{2.t_1})$$

sachant que pour l'équation

$$X.\sin\alpha + Y.\cos\alpha = Z$$

$$\cos\alpha = \frac{YZ - \varepsilon X\sqrt{X^2+Y^2-Z^2}}{X^2+Y^2}$$

$$\sin\alpha = \frac{XZ + \varepsilon Y\sqrt{X^2+Y^2-Z^2}}{X^2+Y^2}$$

avec $\varepsilon = \pm 1$

On en déduit donc $\theta_1$ puis $\theta_1 + \theta_2 \longrightarrow \theta_2$(en utilisant eq. (1)),puis $\theta_3$

## II-4 conclusion:

Nous avons abordé dans ce chapitre les différentes structures et les modèles des bras manipulateurs, on a constaté d'après cette approche que l'extraction du modèle d'un bras consiste à savoir toutes ses caractéristiques géométriques et mécaniques et nous avons vu aussi les deux méthodes essentielles pour obtenir des bras manipulateurs, **le modèle géométrique directe** et **le modèle géométrique inverse**, et encore nous avons déduit les matrice de **Denavit-Hartenberg** qui elle nous ont permis de calculer les résolutions Algébrique pour déduit les angle ($\theta_1, \theta_2, \theta_3$)

# Généralité sur le microcontrôleur

## III -1-Introduction

Le modèle UNO de la société ARDUINO est une carte électronique dont le coeur est un microcontrôleur ATMEL de référence ATMega328. Le microcontrôleur ATMega328 est un microcontrôleur 8bits de la famille AVR dont la programmation peut être réalisée en langage C. L'intérêt principal des cartes ARDUINO (d'autres modèles existent) est leur facilité de mise en œuvre. ARDUINO fournit un environnement de développement s'appuyant sur des outils open- source. Le chargement du programme dans la mémoire du microcontrôleur se fait de façon très simple par port USB. En outre, des bibliothèques de fonctions "clé en main" sont également fournies pour l'exploitation d'entrées-sorties courantes : gestion des E/S TOR, gestion des convertisseurs ADC, génération de signaux PWM, exploitation de bus TWI/I 2 C, exploitation de servomoteurs ... L'objectif de ce document est de mettre en évidence certaines informations techniques concernant l'exploitation des périphériques intégrés, en particulier lorsqu'on n'utilise pas les fonctions "clé en main" d'ARDUINO, dans l'objectif de comprendre comment ça marche !




figure III-1: Carte arduino uno

## Schéma simplifié de la carte Arduino UNO

Les signaux d'entrée-sortie du microcontrôleur sont reliés à des connecteurs selon le schéma ci- dessous.

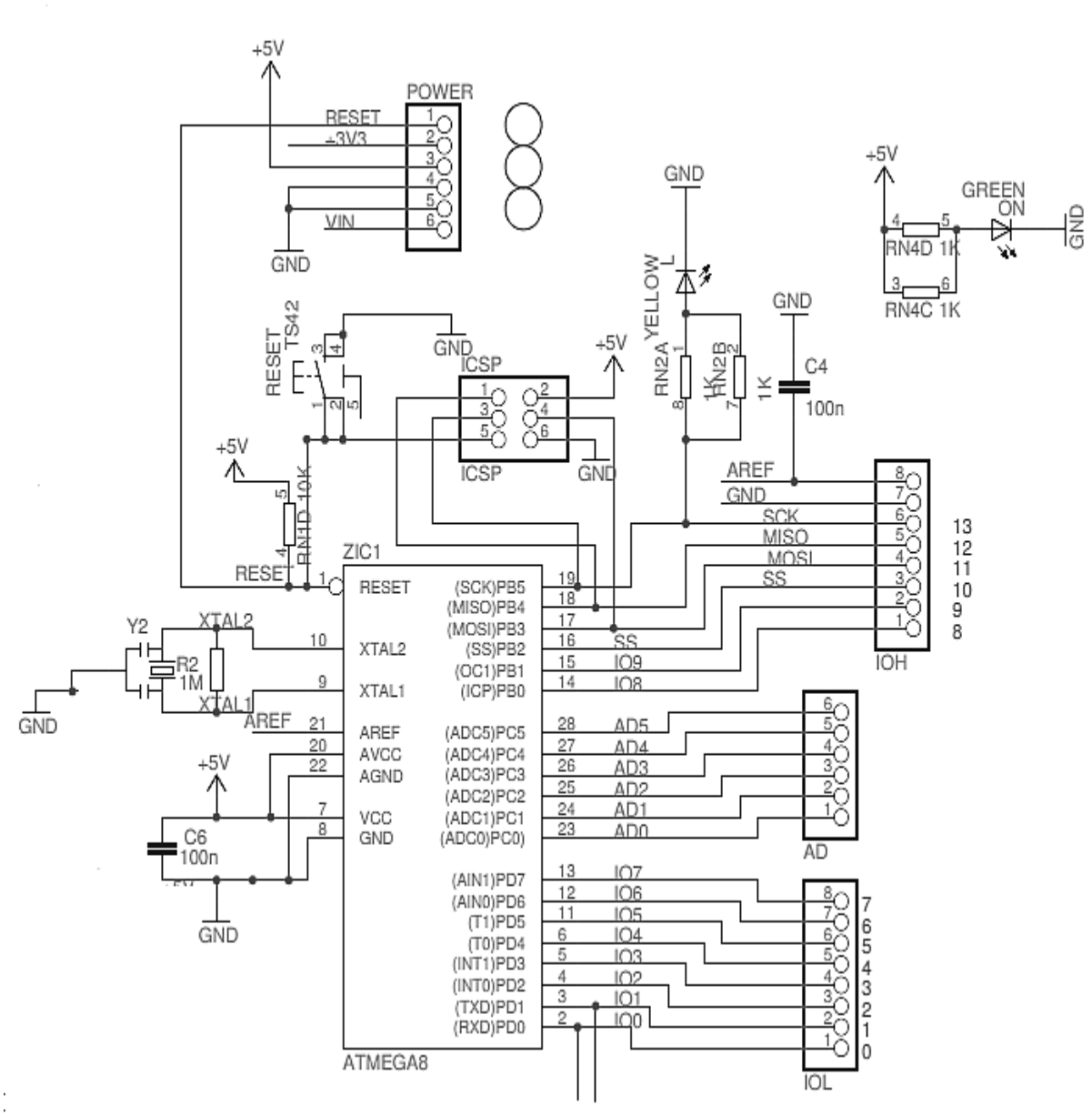



figure III-2 Schéma simplifié de carte arduino

## Architecture interne du microcontrôleur ATMega

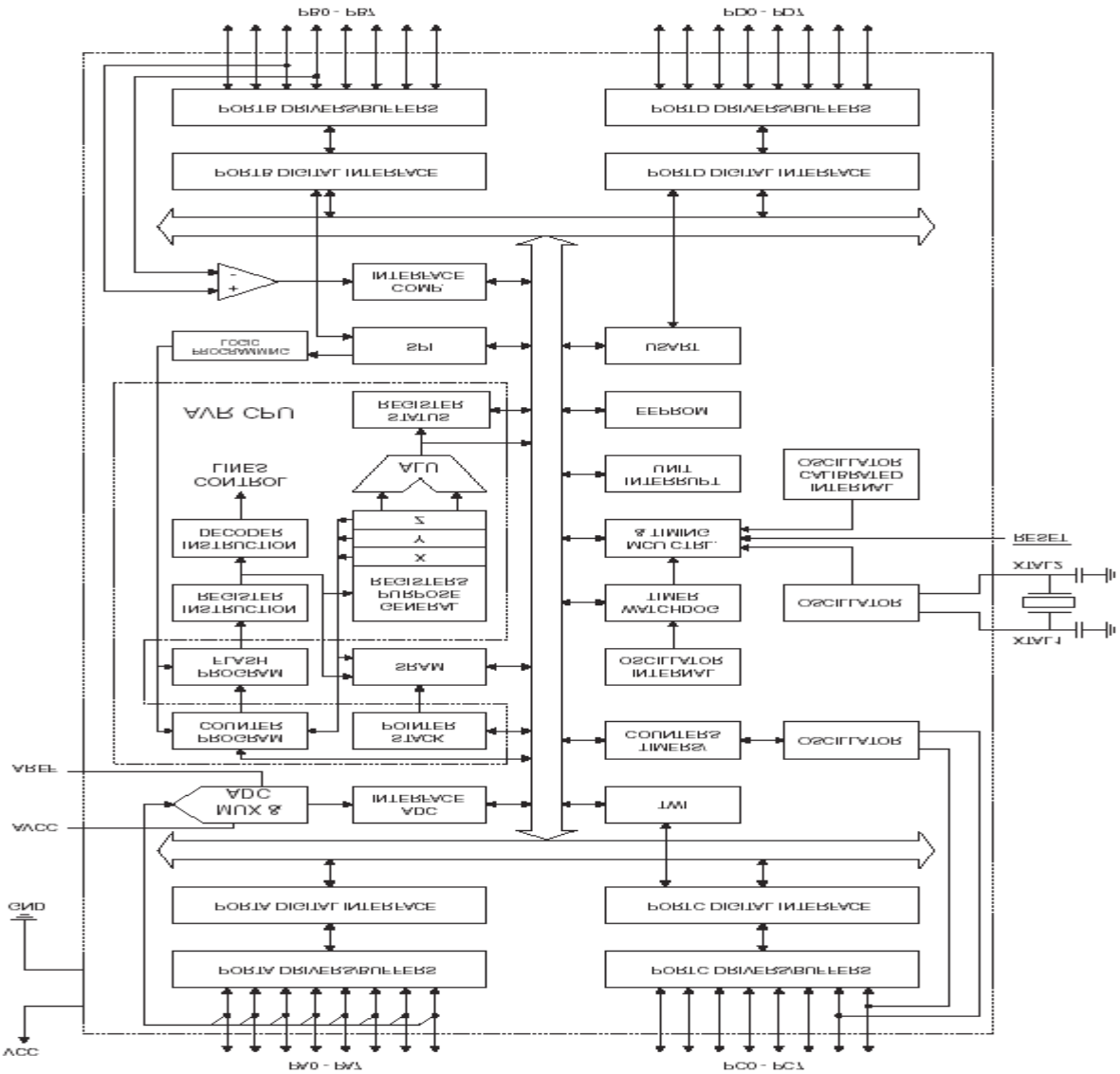


figure III-3 Architecture interne du microcontrôleur ATMega

## III -2 Microcontrôleur ATMEL ATMega328

Le microcontrôleur utilisé sur la carte Arduino UNO est un microcontrôleur ATMega328. C'est un microcontrôleur ATMEL de la famille AVR 8bits.

## III-2-1 Les Modèles ATMEGA AVR

Vous trouverez dans le tableau qui suit tout les modèles d'ATMEGA existant et à venir

| Product | Flash (KB) | EEPROM (Bytes) | RAM (Bytes) | I/O | SPI | USART | USI | TWI | PWM | On-Chip Debug JTAG | On-Chip Debug debugWire | 10-bit ADC | LCD | Availability |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| megaAVR | | | | | | | | | | | | | | |
| ATmega 48 | 4 | 256 | 512 | 23 | 1 | 1 | – | 1 | 5 | – | Y | 8 | – | Q4-03 |
| ATmega8 | 8 | 512 | 1K | 23 | 1 | 1 | – | 1 | 3 | – | – | 8 | – | Now |
| ATmega88 | 8 | 512 | 1K | 23 | 1 | 1 | – | 1 | 5 | – | Y | 8 | – | Q4-03 |
| ATmega8515 | 8 | 512 | 512 | 35 | 1 | 1 | – | – | 3 | – | – | – | – | Now |
| ATmega8535 | 8 | 512 | 512 | 32 | 1 | 1 | – | 1 | 4 | – | – | 8 | – | Now |
| ATmega16 | 16 | 512 | 1K | 32 | 1 | 1 | – | 1 | 4 | Y | – | 8 | – | Now |
| ATmega162 | 16 | 512 | 1K | 35 | 1 | 2 | – | – | 6 | Y | – | – | – | Now |
| ATmega168 | 16 | 512 | 1K | 23 | 1 | 1 | – | 1 | 5 | – | Y | 8 | – | Q4-03 |
| ATmega32 | 32 | 1K | 2K | 32 | 1 | 1 | – | 1 | 4 | Y | – | 8 | – | Now |
| ATmega64 | 64 | 2K | 4K | 53 | 1 | 2 | – | 1 | 8 | Y | – | 8 | – | Now |
| ATmega128 | 128 | 4K | 4K | 53 | 1 | 2 | – | 1 | 8 | Y | – | 8 | – | Now |
| ATmega256 | 256 | 4K | 8K | 53 | 1 | 2 | – | 1 | 16 | – | Y | 8 | – | Q1-04 |
| CD AVR | | | | | | | | | | | | | | |
| ATmega169 | 16 | 512 | 1K | 53 | 1 | 1 | Y | – | 4 | Y | – | 8 | Y | Now |
| ATmega329 | 32 | 1K | 2K | 53 | 1 | 1 | Y | – | 4 | Y | – | 8 | Y | Q1-04 |

**Les principales caractéristiques sont :**

**FLASH** = mémoire programme de 32Ko
**SRAM** = données (volatiles) 2Ko
**EEPROM** = données (non volatiles) 1Ko
**Digital I/O (entrées-sorties Tout Ou Rien)** = 3 ports PortB, PortC, PortD (soit 23 broches en tout I/O)
**Timers/Counters** : Timer0 et Timer2 (comptage 8 bits), Timer1 (comptage 16bits)
Chaque timer peut être utilisé pour générer deux signaux PWM. (6 broches OCxA/OCxB)
**Plusieurs broches multi-fonctions** : certaines broches peuvent avoir plusieurs fonctions différentes choisies par programmation
**PWM** = 6 broches **OC0A(PD6)**, **OC0B(PD5)**, **0C1A(PB1)**, **OC1B(PB3)**, **OC2A(PB3)**, **OC2B(PD3)**
**Analog to Digital Converter** (résolution 10bits) = 6 entrées multiplexées **ADC0(PC0)** à **ADC5(PC5)**
**Gestion bus I2C** (TWI Two Wire Interface) = le bus est exploité via les broches **SDA(PC5)/SCL(PC4)**.
**Port série (USART)** = émission/réception série via les broches **TXD(PD1)/RXD(PD0)**
**Comparateur Analogique** = broches AIN0(PD6) et AIN1 (PD7) peut déclencher interruption
**Watchdog Timer programmable**.
**Gestion d'interruptions** (24 sources possibles (cf interrupt vectors)) : en résumé

– Interruptions liées aux entrées **INT0 (PD2)** et **INT1 (PD3)**
– Interruptions sur changement d'état des broches **PCINT0** à **PCINT23**
– Interruptions liées aux Timers 0, 1 et 2 (plusieurs causes configurables)
– Interruption liée au comparateur analogique
– Interruption de fin de conversion **ADC**
– Interruptions du port série **USART**
– Interruption du bus **TWI (I2C)**

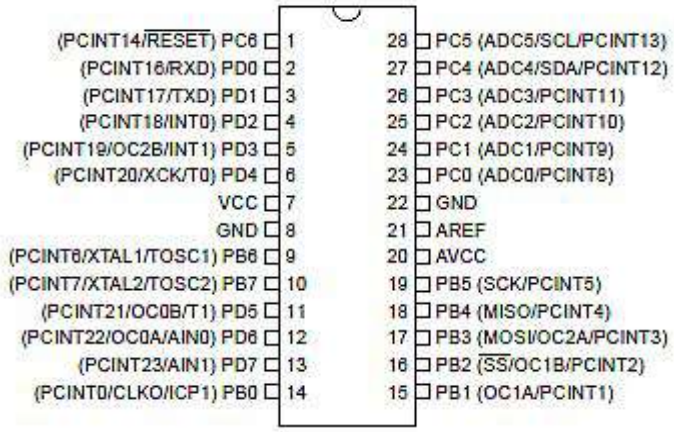



figure III-4 les broches du microcontrôleur ATMega

## III -2-2 Structure interne de l'ATMega328 (extraits de documentations ATMEL)

L'utilisation des périphériques intégrés (Entrées Sorties TOR, Timers, ...) repose sur l'exploitation (lecture/écriture) de registres internes. Ces registres, essentiellement 8 bits, sont décrits par un nom, y compris dans la programmation en C. Cette section fournit quelques détails importants sur les registres internes du microcontrôleur ATMega328.

**Notation :** par la suite, pour un registre nommé **R**, la notation **R.n** désigne le bit de rang **n** du registre **R**. Le bit **R.0** est le bit de poids faible de **R**.

**Descriptions des broches**

**Port A (PA7.. PA0)** le Port A est un port d'entrée-sortie à 8 bits bidirectionnel avec des résistances internes de tirage (choisi pour chaque bit). Il sert aussi pour les entrées analogiques du convertisseur **A/D**. Le Port A (comme le B, C et D) est en position trois états quand une condition de reset devient active, même si l'horloge ne court pas.

**Port B (PB7.. PB0)** le Port B est un port d'entrée-sortie à 8 bits bidirectionnel avec des résistances internes de tirage (choisi pour chaque bit). Il sert aussi de comparateur analogique (sortie sur **PB2, PB3**), ou de **SPI**.

**Port C (PC7.. PC0)** le Port C est un port d'entrée-sortie à 8 bits bidirectionnel avec des résistances internes de tirage (choisi pour chaque bit). Il sert aussi comme oscillateur pour le Timer/Compteur2 et d'interface I2C.

**Port D (PD7.. PD0)** le Port D est un port d'entrée-sortie à 8 bits bidirectionnel avec des résistances
internes de tirage (choisi pour chaque bit). Il sert aussi d'**USART** et d'entrées pour les interruptions externes.
**RESET** déclenché par un front descendant maintenue plus de 50 ns il produira le Reset du microcontrôleur, même si l'horloge ne court pas.
**XTAL1** Entrée de l'oscillateur externe ou libre pour l'horloge interne.
**XTAL2** Production de l'amplificateur d'oscillateur.
**AVCC** est une broche de tension d'alimentation pour le Convertisseur **A/D** qui doit être connectée à
**VCC** via un filtre passe-bas pour éviter les parasites.
**AREF** est l'entrée de référence analogue pour le Convertisseur **A/D** avec une tension dans la gamme
de **2V** à **AVCC** avec filtre passe bas.
**AGND** masse Analogique. Si la masse analogique est séparée de la masse générale, brancher cette
broche sur la masse analogique, sinon, connecter cette broche à la masse générale **GND**.
**VCC** broches d'alimentation du microcontrôleur (+3 à +5V).
**GND** masse de l'alimentation.

**L'Horloge Système et Option**

Le choix du type d'horloge est déterminé lors de la programmation de la mémoire FLASH du microcontrôleur et peux être choisi parmi les données du tableau suivant :

| Dispositif d'Option d'horloge | CKSEL3, 2, 1 & 0 |
|---|---|
| Résonateur Externe Cristal/Céramique | 1111 - 1010 |
| Cristal Basse Fréquence Externe | 1001 |
| Oscillateur Externe à RC | 1000 - 0101 |
| Oscillateur Calibré Interne à RC | 0100 - 0001 |
| Horloge Externe | 0000 |

L'oscillateur à quartz externe est connecté sur XTAL1 et XTAL2 comme le montre la figure suivante pour cadencer le microcontrôleur. La fréquence du quartz est de 4 à 16 MHz de type cristal ou un résonateur céramique :

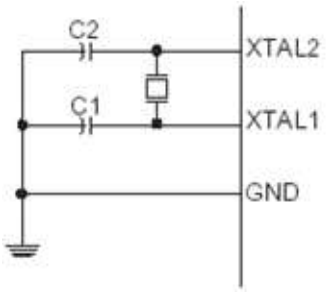



figure III-5 Montage du Quartz.

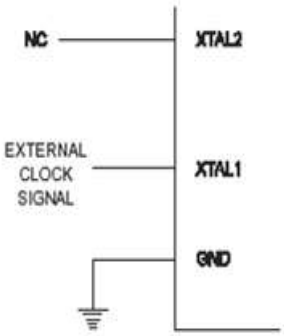



figure III-6 Signale d'Horloge Externe.

- **L'horloge Interne**

Pour conduire un dispositif externe avec la source d'horloge interne, **XTAL2** doit être laissée en l'air, tandis que l'on connecte **XTAL1** au dispositif à piloter, comme indiqué dans la Figure5.

- **L'oscillateur du Timer**

Pour l'oscillateur du Timer temps réel, les broches **TOSC1** et **TOSC2 (PC6 & PC7)** peuvent recevoir un cristal de **32 768 Hz** directement connecté entre ces broches sans condensateur externe. L'application d'une source d'horloge externe sur **TOSC1** n'est pas recommandée. Ce système permet de cadencer le Timer avec des valeurs de temps en sous-multiple de la seconde.

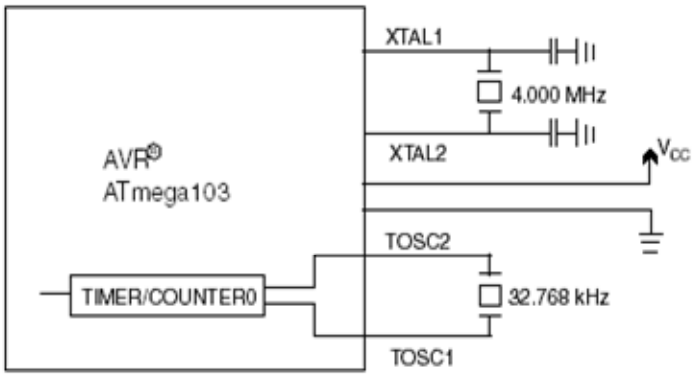



figure III-7 Exemple de connexion d'horloge Temps Réel.

**Le Plan Mémoire**

Trois types de mémoire sont utilisés dans la série **ATMEGA**, la mémoire programme **FLASH**, la mémoire de donnée **SRAM** et la mémoire morte de type **EEPROM**.

**La mémoire programme**

La mémoire programme permet de stoker et de faire fonctionner le microcontrôleur, il contient de 4 à 256 Ko de programme selon le modèle du microcontrôleur. Le nombre d'écriture sur cette mémoire est limité à 10.000, largement suffisant pour la majorité des applications. La figure 6 donne un exemple de l'adressage de la mémoire **FLASH** du modèle **ATMEGA 32**.

**La mémoire de donnée**

La mémoire de donnée contient les 32 registres de travail, les 64 registres de commande et la mémoire **SRAM** pour les variables du programme de 2048 octets pour le modèle **ATMEGA 32**. La figure 7 présente les relations entre espace physique et registre.

**La mémoire morte**

La mémoire morte est de type **EEPROM** d'accès plus complexe contiendra la configuration du programme et les données importantes qui seront sauvé pendant l'absence de courant électrique. On peut écrire jusqu'à 100.000 fois dans **l'EEPROM**. La taille de **l'EEPROM** est fonction du modèle de microcontrôleur **ATMEGA** (de 256 bits à 8 Ko).

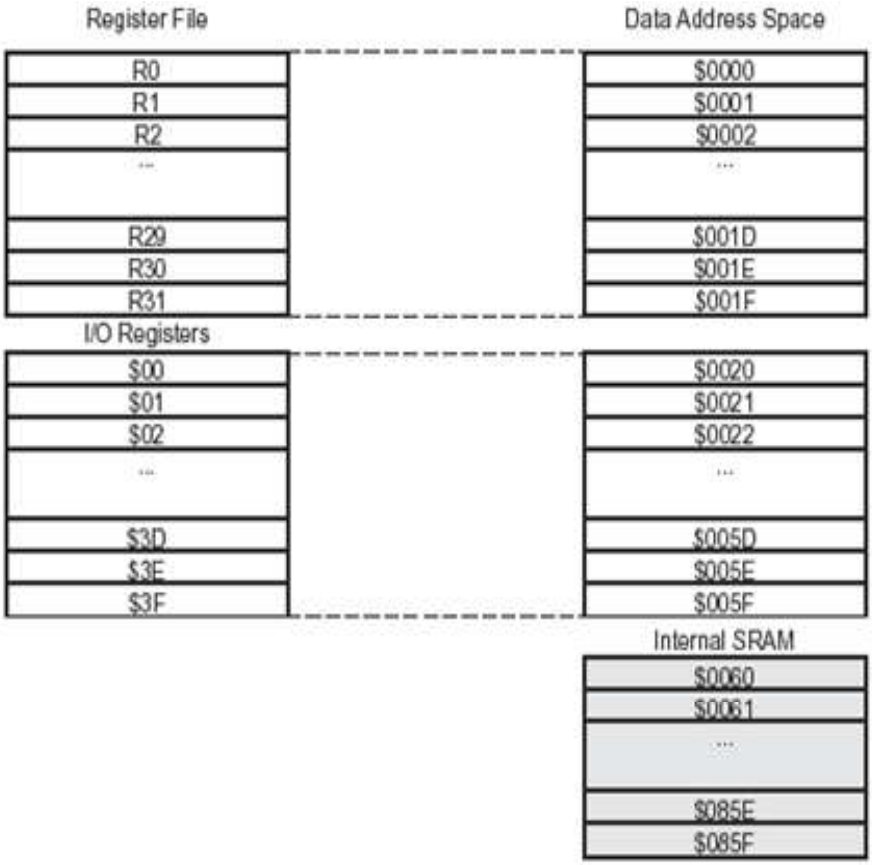



figure III- 8, la mémoire de donnée avec la **SRAM**

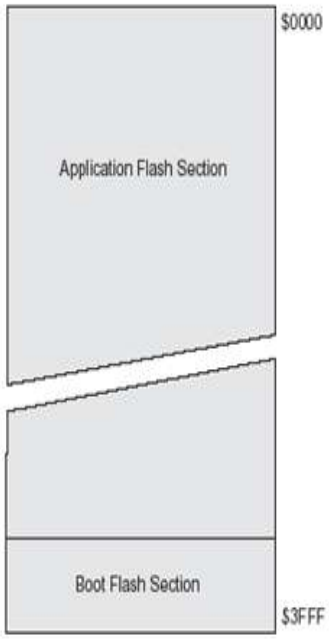



figure III-9. Adressage de la mémoire **FLASH**

## III -3 Tous les Registres

Le tableau suivant donne la liste des 66 registres (deux sont communs à d'autres registre) existant dans la gamme ATMEGA :

| Catégorie | Désignation | Registre | Adresse |
|---|---|---|---|
| **Interface deux fils I2C (TWI)** | Contrôle du Bus | TWBR | $00 |
| | Status | TWSR | $01 |
| | Adresse | TWAR | $02 |
| | Données à transmettre ou reçus | TWDR | $03 |
| **Convertisseur analogique** | Poids Faible Résultat | ADCL | $04 |
| | Poids fort Résultat | ADCH | $05 |
| | Contrôle et statut | ADCSR | $06 |
| | Sélection de la voie à échantillonner | ADMUX | $07 |
| **Comparateur analogique** | Contrôle et statut du Comparateur | ACSR | $08 |
| **USART** | Vitesse de communications (Bauds) | UBRR | $09 |
| | Contrôle | UCR | $0A |
| | Status | USR | $0B |
| | I/O donnée | UDR | $0C |
| **SPI** | Contrôle | SPCR | $0D |
| | Status | SPSR | $0E |

| | | | |
|---|---|---|---|
| | I/O donnée | SPDR | $0F |
| **Port parallèle D** | Adresse des broches (Pin) | PIND | $10 |
| | Direction des broches | DDRD | $11 |
| | Données des broches | PORTD | $12 |
| **Port parallèle C** | Adresse des broches (Pin) | PINC | $13 |
| | Direction des broches | DDRC | $14 |
| | Données des broches | PORTC | $15 |
| **Port parallèle B** | Adresse des broches (Pin) | PINB | $16 |
| | Direction des broches | DDRB | $17 |
| | Données des broches | PORTB | $18 |
| **Port parallèle A** | Adresse des broches (Pin) | PINA | $19 |
| | Direction des broches | DDRA | $1A |
| | Données des broches | PORTA | $1B |
| **EEPROM** | Contrôle | EECR | $1C |
| | Donnée | EEDR | $1D |
| | Poids faible adresse | EEARL | $1E |
| | Poids Fort adresse | EEARH | $1F |
| **USART** | Vitesse de communications (Bauds) | UBRRH/UCSRC | $20 |
| **Watchdog** | Chien de Garde | WDTCR | $21 |
| **Timer / compteur 2 (8 bits)** | Statut du mode asynchrone | ASSR | $22 |
| | Comparaison | OCR2 | $23 |
| | Compteur | TCNT2 | $24 |
| | Contrôle | TCCR2 | $25 |
| **Timer / compteur 1 (16 bits)** | Entrée poids faible de capture | ICR1L | $26 |
| | Entrée poids fort de capture | ICR1H | $27 |
| | Timer B sortie comparaison poids faible | OCR1BL | $28 |
| | Timer B sortie comparaison poids fort | OCR1BH | $29 |
| | Timer A sortie comparaison poids faible | OCR1AL | $2A |
| | Timer A sortie comparaison poids fort | OCR1AH | $2B |
| // | Timer/compteur 1 poids Faible | TCNT1L | $2C |

| | | | |
|---|---|---|---|
| | Timer/compteur 1 poids Fort | TCNT1H | $2D |
| **Port I/O Oscillateur** | Contrôle Timer/compteur 1 poids Faible | TCCR1B | $2E |
| | Contrôle Timer/compteur 1 poids Fort | TCCR1A | $2F |
| | Registre Spéciale des fonctions I/O | SFIOR | $30 |
| | Calibration d'Oscillateur | OSCCAL/OCDR | $31 |
| **Timer / compteur 0 (8 bits)** | Timer/compteur 0 | TCNT0 | $32 |
| | Contrôle Timer/compteur 0 | TCCR0 | $33 |
| **MCU** | Statuts Général | MCUSR | $34 |
| | Contrôle Général | MCUCR | $35 |
| **Interface I2C (TWI)** | Contrôle Général | TWCR | $36 |
| **MCU 'Boot Loader'** | Contrôle mémoire programme | SPMCR | $37 |
| **Interruptions** | Drapeau d'interruptionCompteur/Timer | TIFR | $38 |
| | Masque d'interruption Compteur/Timer | TIMSK | $39 |
| | Drapeau d'interruption Généraux | GIFR | $3A |
| | Masque d'interruption Généraux | GICR | $3B |
| | Timer/compteur comparaison en Sortie | OCR0 | $3C |
| **Pointeur de pile** | Pointeur de pile poids faible | SPL | $3D |
| | Pointeur de pile poids fort | SPH | $3E |
| **Registre SREG** | Registre d'Etat et de Statut | SREG | $3F |

observation: le symbole ($) indique les données en Hexadécimales pour raccourcie l'espace.

## II -4 conclusion:

Nous avons présenté dans ce chapitre une introduction générale sur le microcontrôleur de carte arduino uno, qu -il-t Apple ATMega328.
on a présenté aussi leur architecteur interne et leur **Registre** et comment relier ce article avec l'extérieur.

# Réalisation Pratique

## IV.1 Listes des matériaux utiliser :

- la carte ARDUINO UNO.
- servomoteur
- plaques d'essai
- moteur courant continu (cc)
- les câbles électrique
- le bois
- élastique

## IV.1.2 Cahier des charges fonctionnel :

- **Fonctions principales:**
  - ✓ Démarrer
  - ✓ Le bras manipulateur rattraper l'objet
  - ✓ Le bras manipulateur tourne certain angle (par rapport le programme )
  - ✓ Le bras manipulateur poser l'objet
  - ✓ Le tapis rolon été marche

## IV.1.3 Budget de ce projet :

| l'objet | le nombre | le prix d'unité (DA) | prix totale (DA) |
|---|---|---|---|
| la carte ARDUINO UNO | 1 | 4000 | 4000 |
| servomoteur | 4 | 2000 | 8000 |
| plaques d'essai | 1 | 500 | 500 |
| moteur cc | 1 | 150 | 150 |
| les câbles électrique | 16 | 30 | 480 |
| bande élastiqué | 1.5mètre | 300 | 450 |
| colle blanc | 1 | 250 | 250 |
| totale | | | 13830 |

**tableau de budget**

**Remarque:** Le bois est le déche approche par le menuiser

## IV.2 Les Servomoteurs:

### IV.2-1 Introduction:

Les servomoteurs sont très utilisés en robotique car ils sont à la base de nombreux mécanisme d'actionneurs, comme des pinces par exemple. Par contre, ils consomment souvent beaucoup de courant et demandent une tension souvent très bien régulée ce qui nécessite des dispositions particulières.



figure IV -1 Servomoteur

### IV.2-2 Qu'est-ce qu'un servomoteur ?

Un servomoteur est un dispositif typiquement utilisé en modélisme, par exemple pour diriger une voiture télécommandée ou contrôler à distance une caméra vidéo.
Il comprend :

- Un moteur électrique de petite taille.
- Un réducteur en sortie de ce moteur pour avoir moins de vitesse et plus de couple.
- Un capteur : un potentiomètre qui produit une tension variable en fonction de l'angle de l'axe de sortie.
- Un asservissement électronique pour contrôler la position de cet axe de sortie.

### IV.2-3 Fonctionnement d'un servomoteur :

L'utilisateur envoie à chaque instant un signal modulé représentatif d'un angle. Le servomoteur tourne vers cet angle et s'y maintient.
La duré de l'impulsion se varie entre 0.9ms qui permet à l'axe de tourner jusqu'à la position

extrême gauche et 2.1ms qui lui permet d'aller jusqu'à l'extrême droite. Et avec 1.5ms pour revenir au centre. Pour varier la vitesse d'un servomoteur en jouant sur la durée de l'impulsion. Plus la durée de l'impulsion est proche de 1.5ms plus la vitesse de rotation est faible.

## IV.2-4 L'architecture interne:

Un servomoteur contient un moteur à courant continu, un réducteur à roues dentées à axe parallèles et une électronique commande. L'alimentation et la commande se font par un câble de trois fils, un commun, un fil d'alimentation et un fil de commande. les couleurs sont conventionnelles pour un constructeur

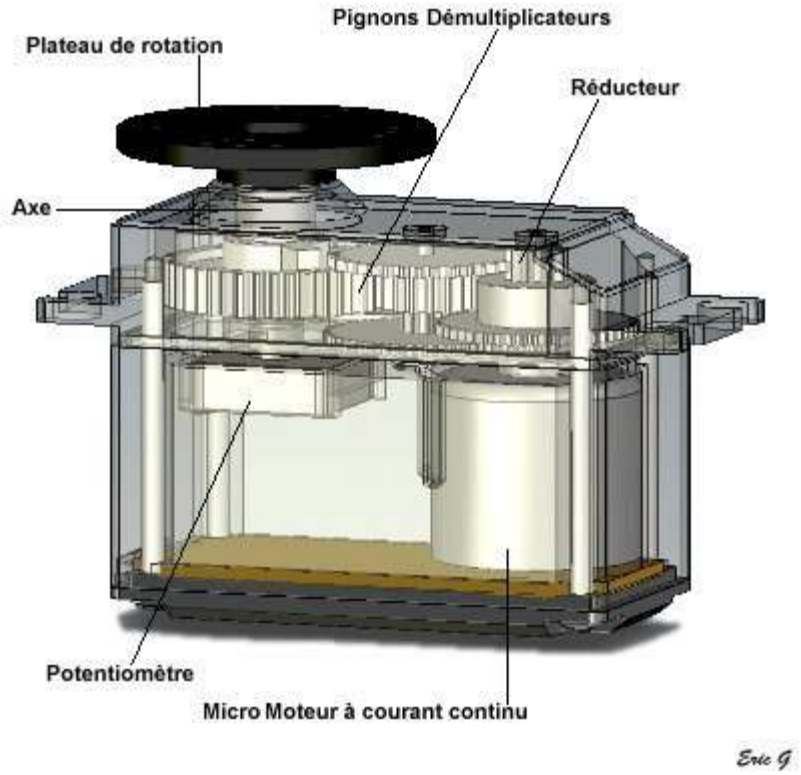



figure IV -2 : vue de l'extérieur et de l'intérieur d'un servomoteur

## IV.2-5 Fonctionnement d'un servomoteur :

L'utilisateur envoie à chaque instant un signal modulé représentatif d'un angle. Le servomoteur tourne vers cet angle et s'y maintient. La duré de l'impulsion se varie entre 0.9ms qui permet à l'axe de tourner jusqu'à la position extrême gauche et 2.1ms qui lui permet d'aller jusqu'à l'extrême droite. Et avec 1.5ms pour revenir au centre. Pour varier la vitesse d'un servomoteur en jouant sur la durée de l'impulsion. Plus la durée de l'impulsion est proche de 1.5ms plus la vitesse de rotation est faible.

## IV.2-6 Le potentiomètre :

Le potentiomètre sert à l'asservissement de position. L'axe du potentiomètre est solidaire de l'axe de sortie qui lui permet de tourner en même temps que l'axe de sortie. La résistance à la borne du potentiomètre varie en fonction de la position de l'axe. La résistance du potentiomètre est en rapport avec la durée du signal. Par exemple le servomoteur reçoit un ordre pour se déplacer à la position médiane (1.5ms). Le moteur va tourner jusqu'à ce que la résistance entre le curseur du potentiomètre et les deux extrémités soient identiques et s'il reçoit une impulsion de 2.1ms il va tourner jusqu'à ce que la résistance entre le curseur et l'extrémité du potentiomètre soit maximale. L'électronique faire la relation entre la durée de l'impulsion et la résistance aux bornes du potentiomètre.

## IV.2-7 La commande du servomoteur:

Un connecteur à trois fils permet de commander un servo-moteur. Bien sûr, il y a la masse et le +5V (à peu près). Ces deux fils sont en général noir et rouge, mais ils sont marrons et rouge pour les modèles Graupner. Le 3ème fil est le fil de commande. Il est normalement blanc, mais jaune dans le cas des Graupner. On pourrait dire sommairement qu'il faut fournir un signal PWM en tant que signal de commande, mais ce n'est pas tout à fait exact :

- Le PWM (Pulse Width Modulation : Modulation en largeur d'impulsion) est un signal dont le taux de modulation varie de 0 à 100%

$$T_M = \frac{T_{ON}}{T_{PERIODE}}$$

- Pour un servomoteur on parle de Pulse-Code Modulated Signal (Signal modulé en code d'impulsion ) où seule l'impulsion compte, pas la fréquence (en théorie).
  En clair, il faut fournir au servomoteur une impulsion à 1 (suivie d'un retour à 0).
  Le servomoteur va prendre en compte la largeur temporelle de cette impulsion qu'il va convertir de façon linéaire en un angle. La durée de retour à 0 de l'impulsion n'est pas critique, en pratique il faut éviter de dépasser 20ms entre deux fronts montants.
  A noter aussi que ce système présente un avantage par rapport au PWM : une absence de signal (toujours à 1, ou à 0) laisse le servomoteur en "roue libre" comme si il n'était pas alimenté. Le microcontrôleur programmé est capable de fournir directement ce signal de commande sur une sortie.

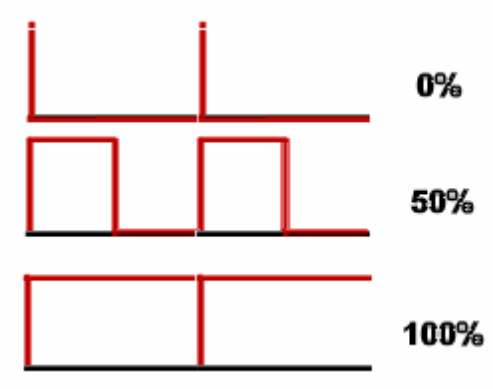



figure IV -3 : Signal PWM

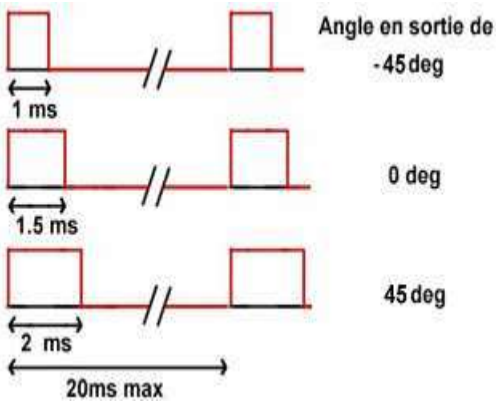



figure IV -4 : Signal modulé en code d'impulsion



figure IV -5 : Dimension du servomoteur

## IV.3 Moteur courant continu:

### IV.3 -1 Introduction:

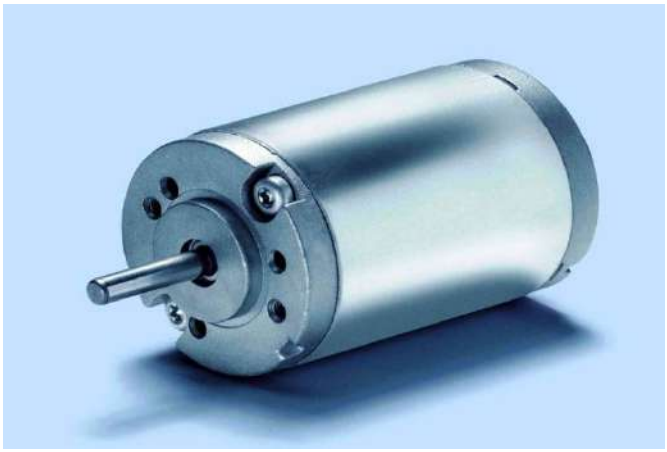


figure IV -6 Moteur à courant continu

Une **machine à courant continu** est une machine électrique. Il s'agit d'un convertisseur électromécanique permettant la conversion bidirectionnelle d'énergie entre une installation électrique parcourue par un courant continu et un dispositif mécanique; selon la source d'enregie

- En fonctionnement moteur, l'énergie électrique est transformée en énergie mécanique.
- En fonctionnement générateur, l'énergie mécanique est transformée en énergie électrique. Dans ce cas elle est aussi appelée dynamo.

Inventée par Zénobe Gramme, c'était au départ un simple générateur de courant continu (pour applications galvanoplastiques, par exemple, les accumulateurs étant onéreux). Le moteur sans balais, qui l'a supplanté, est un perfectionnement de cette machine.

## IV.3 -2 Description de la Une machine à courant continu:

Une machine électrique à courant continu est constituée de:

- Un stator qui crée un flux magnétique *longitudinal* fixe soit par des enroulements statoriques (bobinage) soit par des aimants permanents. Il est aussi appelé « inducteur » en référence au fonctionnement en génératrice de cette machine.
- Un rotor bobiné relié à un collecteur rotatif inversant la polarité de chaque enroulement rotorique, au moins une fois par tour, de façon à faire circuler un flux magnétique *transversal* en quadrature avec le flux statorique. Les enroulements rotoriques sont aussi appelés enroulements d'induits, ou communément « induit » en référence au fonctionnement en génératrice de cette machine.

## IV.3 -2-1 Stator du moteur courant continu

Le stator est constitué de la carcasse du moteur et du circuit magnétique proprement dit.
Un circuit magnétique est constitué d'une structure ferromagnétique qui canalise le flux magnétique, créé par une source de champ magnétique : aimant permanent ou électro-aimant.
Le circuit magnétique du stator crée le champ magnétique appelé « champ inducteur » (Bs).
L'inducteur magnétise le moteur en créant un flux magnétique ( ϕ ) dans l'entrefer.
L'entrefer est l'espace entre les pôles du stator et le rotor. Le flux magnétique est maximal au niveau des pôles
Il s'agit du moteur courant continu qui assure l'entraînement du Scootelec de Peugeot.




figure IV -7/8 Rotor et stator du moteur cc

Ce moteur possède 2 paires de pôles inducteurs (Nord – Sud) pour mieux répartir le flux magnétique dans la machine. Les lignes de champ magnétique vont du pôle Nord vers le pôle Sud.
La configuration des pôles est donnée pour un sens du courant inducteur. Si le courant est inv ersé, le pôle sud devient un pôle nord et inversement. C'est une possibilité pour inverser le sens de rotation du moteur, mais elle est peu utilisée.

**IV.3 -2-2 Rotor du moteur courant continu:**

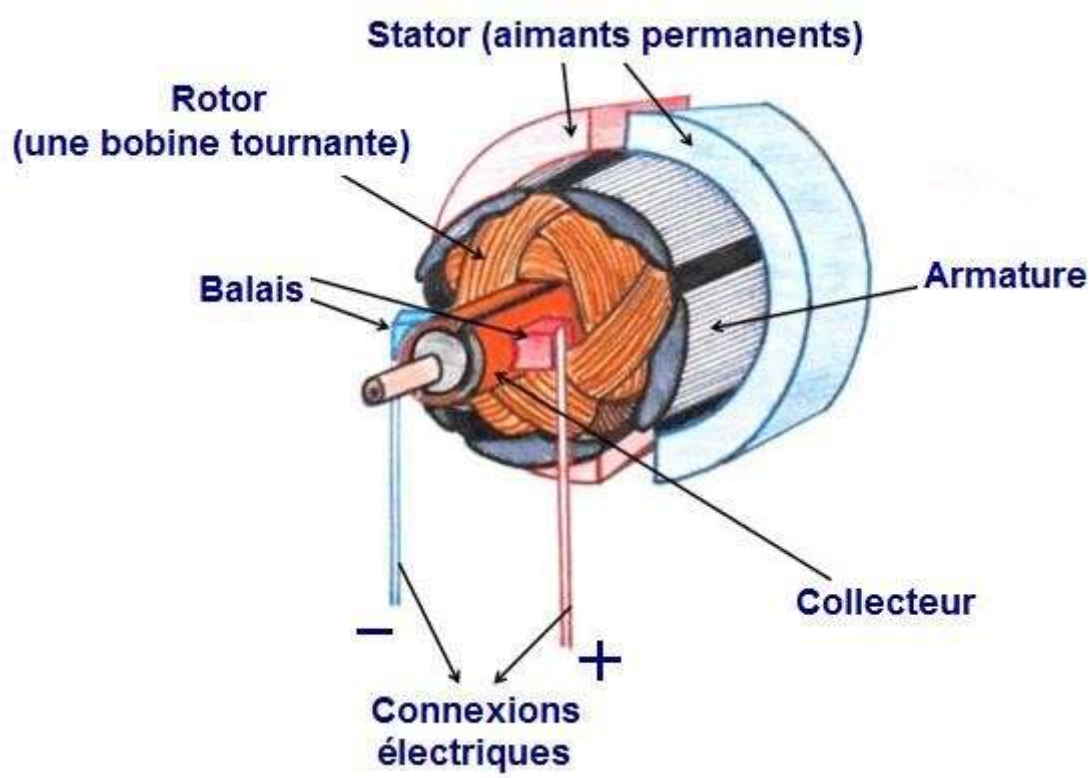



figure IV -9 L'intérieur du moteur cc

## IV.3 -3 Constitution et principes physiques

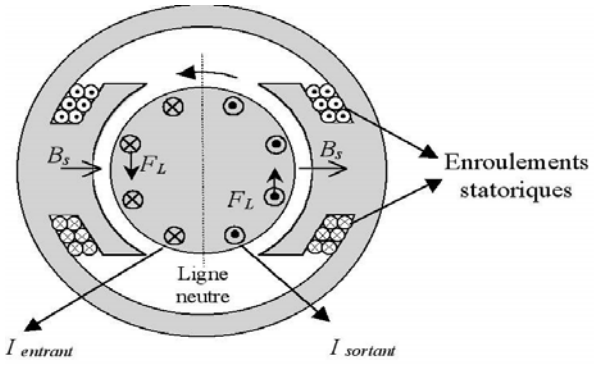



figure IV -10 Constitution et principes physiques d'un moteur cc

- Le courant $i$ injecté via les balais au collecteur, traverse un conducteur rotorique (une spire rotorique) et change de sens (commutation) au droit des balais. Ceci permet de maintenir la magnétisation du rotor perpendiculaire à celle du stator.
  La disposition des balais sur la « ligne neutre » (c'est-à-dire la zone où la densité de flux est nulle), permet d'obtenir la force contre-électromotrice (FCEM) maximum. Cette ligne peut néanmoins se déplacer par la réaction magnétique d'induit (influence du flux rotorique sur le champ inducteur) selon que la machine travaille à forte ou à faible charge. Une surtension, due en partie à la mauvaise répartition de la tension entre lames du collecteur et en partie à l'inversion rapide du courant dans les sections de conducteur lors du passage de ces lames sous les balais, risque alors d'apparaître aux bornes de la spire qui commute et de provoquer la destruction progressive du collecteur. Pour pallier cela, c'est-à-dire compenser la réaction d'induit, et aussi améliorer la commutation, on utilise des pôles auxiliaires de compensation/commutation.
- L'existence du couple s'explique par l'interaction magnétique entre stator et rotor :
  - Le champ statorique ($\boldsymbol{B_s}$ sur le schéma) est pratiquement nul sur les conducteurs logés dans des encoches et n'agit donc pas sur eux. L'origine du couple reste la magnétisation transversale du rotor, inchangée au cours de sa rotation (rôle du collecteur). Un pôle statorique agit sur un pôle rotorique et le moteur tourne.
  - Une manière classique mais simpliste de calculer le couple est de s'appuyer sur l'existence d'une force de Laplace (fictive) créée par le champ statorique ( $\boldsymbol{B_s}$ sur le schéma) et agissant sur les conducteurs rotoriques traversés par le courant d'intensité $\boldsymbol{I}$. Cette force ( $\boldsymbol{F_L}$ sur le schéma) qui résulte de cette interaction est identique en module pour deux conducteurs rotoriques diamétralement opposés, mais comme ces courants sont en sens inverse grâce au système balais-collecteur, les forces sont aussi de sens opposés.

- La force ainsi créée est proportionnelle à $\boldsymbol{I}$ et à $\boldsymbol{B_s}$. Le couple moteur $\boldsymbol{T}$ est donc lui aussi proportionnel à ces deux grandeurs.
- La tige conductrice traversée au rotor par le courant $\boldsymbol{I}$ se déplace soumise au champ statorique $\boldsymbol{B_s}$. Elle est donc le siège d'une FCEM induite (loi de Lenz) proportionnelle à $\boldsymbol{B_s}$ et à sa vitesse de déplacement, donc à la fréquence de rotation. L'ensemble de ces forces contre-électromotrices a pour conséquence l'apparition d'une FCEM

globale $E'$ aux bornes de l'enroulement rotorique qui est proportionnelle à $B_s$ et à la vitesse de rotation du moteur.

- Pour permettre au courant $I$ de continuer à circuler, il faudra que l'alimentation électrique du moteur délivre une tension supérieure à la force contre-électromotrice $E$ induite au rotor.

## IV.3 -4Schéma électrique idéalisé:

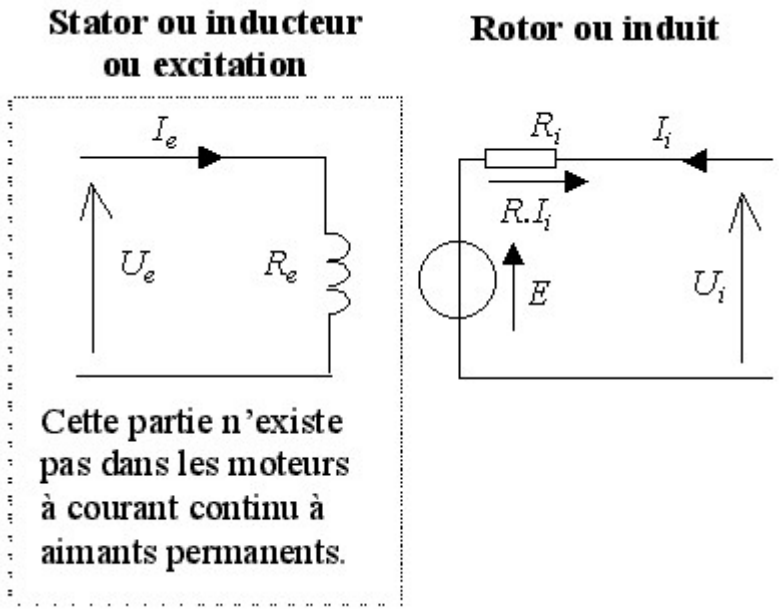



figure IV -11 Le schéma électrique

Ce schéma rudimentaire n'est pas valable en régime transitoire. $R_i$ et $R_e$ sont respectivement les résistances du rotor (résistance de l'induit) et du stator (résistance du circuit d'excitation)
Ce schéma correspond aux équations électriques suivantes :

- au stator : $U_e = R_e \cdot I_e$ (loi d'ohm) et le champ statorique vaut $B_s = k \cdot I_e$ (*la moins exacte des formules de ce paragraphe, car on ne tient pas compte des non-linéarités qui sont importantes et, de plus, on suppose que la machine comporte des enroulements de compensation/commutation qui rendent ce champ indépendant des courants rotoriques. En fait, on fait passer dans ces enroulements de compensation/commutation un courant tel qu'il crée un champ annulant le champ induit au niveau des balais. Ce courant est le courant passant dans l'enroulement*

*d'armature car le champ de commutation doit varier de la même manière que le champ induit.*)

- au rotor : $U_i = E' + R_i \cdot I_i$

D'autre part on a deux équations électromécaniques :

- La force contre électromotrice : $E = k' \cdot B_s \cdot \Omega$
  où $\Omega$ correspond à la vitesse de rotation, ou pulsation, en rad/s reliée à la fréquence (nombre de tours par seconde) par la relation

$$f - \frac{\Omega}{2\pi}$$

- Le couple électromécanique (moteur ou résistant) : $T = k' \cdot B_s \cdot I_i$

On peut montrer que c'est la même constante k' qui intervient dans les deux relations précédentes ce qui implique :

- $E \cdot I_i = T \cdot \Omega$ ou « Puissance électrique utile » = « Puissance mécanique ».

**IV.3 -4-1 Descriptif du fonctionnement:**

Imaginons une machine électrique alimentée par une source de tension $U$ constante. Lorsque le moteur tourne à vide (il ne fait pas d'effort) il n'y a pas besoin de fournir de couple, $I_i$ est très faible et $U \approx E$ La vitesse de rotation est proportionnelle à $U$.

- **fonctionnement en moteur**

Lorsque l'on veut la faire travailler, en appliquant un couple résistant sur son axe, cela la freine donc $E$ diminue.
Comme $U$ reste constante, le produit $R_i . I_i$ augmente donc $I_i$ augmente, donc le couple $T$ augmente lui aussi et lutte contre la diminution de vitesse : c'est un couple moteur.
Plus on le freine, plus le courant augmente pour lutter contre la diminution de vitesse. C'est pourquoi les moteurs à courant continu peuvent « griller » lorsque le rotor est bloqué, si le courant de la source n'est pas limité à une valeur correcte.

- **fonctionnement en génératrice**

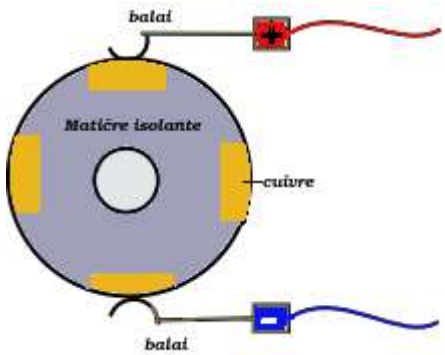
Si une source d'énergie mécanique essaie d'augmenter la vitesse de machine, (la charge est entraînante : ascenseur par exemple), $\Omega$ augmente donc $E$ augmente.
Comme $U$ reste constante, le produit $R_i . I_i$ devient négatif et augmente en valeur absolue, donc $'I_i$ augmente, donc le couple $'T$ augmente lui aussi et lutte contre l'augmentation de vitesse : c'est un **couple frein**.
Le signe du courant ayant changé, le signe de la puissance consommée change lui aussi. La machine consomme une *puissance négative*, donc elle fournit de la puissance au circuit. Elle est devenue génératrice.
Ces deux modes de fonctionnement existent pour les deux sens de rotation de la machine. Celle-ci pouvant passer sans discontinuité d'un sens de rotation ou de couple à l'autre. On dit alors qu'elle fonctionne dans les quatre quadrants du plan couple-vitesse.

## IV.3 -5 Avantages et inconvénients

L'avantage principal des machines à courant continu réside dans leur adaptation simple aux moyens permettant de régler ou de faire varier leur vitesse, leur couple et leur sens de rotation : les variateurs de vitesse, voire leur raccordement direct à la source d'énergie : batteries d'accumulateur, piles, etc.
Le principal problème de ces machines vient de la liaison entre les balais, ou « charbons » et le collecteur rotatif. Ainsi que le collecteur lui-même comme indiqué plus haut et la complexité de sa réalisation. De plus il faut signaler que :

- plus la vitesse de rotation est élevée, plus la pression des balais doit augmenter pour rester en contact avec le collecteur donc plus le frottement est important ;
- aux vitesses élevées les balais doivent donc être remplacés très régulièrement ;
- le collecteur imposant des ruptures de contact provoque des arcs, qui usent rapidement le commutateur et génèrent des parasites dans le circuit d'alimentation, ainsi que par rayonnement électromagnétique.

Un autre problème limite les vitesses d'utilisation élevées de ces moteurs lorsque le rotor est bobiné, c'est le phénomène de « défrettage », la force centrifuge finissant par casser les liens assurant la tenue des ensembles de spires (le frettage).
La température est limitée au niveau du collecteur par l'alliage utilisé pour braser les conducteurs du rotor aux lames du collecteur. Un alliage à base d'argent doit être utilisé lorsque la température de fonctionnement dépasse la température de fusion de l'alliage classique à base d'étain.
Un certain nombre de ces inconvénients ont partiellement été résolus par des réalisations de moteurs sans fer au rotor, comme les moteurs « disques » ou les moteurs « cloches », qui néanmoins possèdent toujours des balais.
Les inconvénients ci-dessus ont été radicalement éliminés grâce à la technologie du moteur *brushless*, aussi dénommé « moteur à courant continu sans balais », ou moteur sans balais.



## IV -4 L'architecture (plan) de ce projet:

Dans ce projet on va utiliser 4 **servomoteur** à chaque articulation :

- ✓ Le 1$^{er}$ servo pour la base.
- ✓ Le 2$^{ème}$ servo pour l'articulation une.
- ✓ Le 3$^{ème}$ servo pour l'articulation deux.
- ✓ Le 4$^{ème}$ servo pour le pince.

et on a utiliser un moteur **à courant continu** pour un tapis rolon, on va faire ça par une bande élastiqué relier autour 4 roue placé sur une morceau d'une baguette à bois.

## IV -5 Les logiciels de programmation et de simulation:

Dans notre étude nous avons choisi les deux ce logiciels :

- ✓ Le logiciel arduino pour la programmation du microcontrôleur ATMega328,ce compilateur est très simple à manipuler.
- ✓ le logiciel ISIS de proteus pour la smulation du programme dévloppe.

## IV -5-1 Le logiciel arduino:




- **Introduction:**

Le logiciel de programmation des modules Arduino est une application Java, libre et multi-plateforme, servant d'éditeur de code et de compilateur, et qui peut transférer le firmware et le programme au travers de la liaison série (RS-232, Bluetooth ou USB selon le module). Il est également possible de se passer de l'interface Arduino, et de compiler et uploader les programmes via l'interface en ligne de commande. Le langage de programmation utilisé est le C++, compilé avec avr-g++ , et lié à la bibliothèque de développement Arduino, permettant l'utilisation de la carte et de ses entrées/sorties. La mise en place de ce langage standard rend

aisé le développement de programmes sur les plates-formes Arduino, à toute personne maîtrisant le C ou le C++.

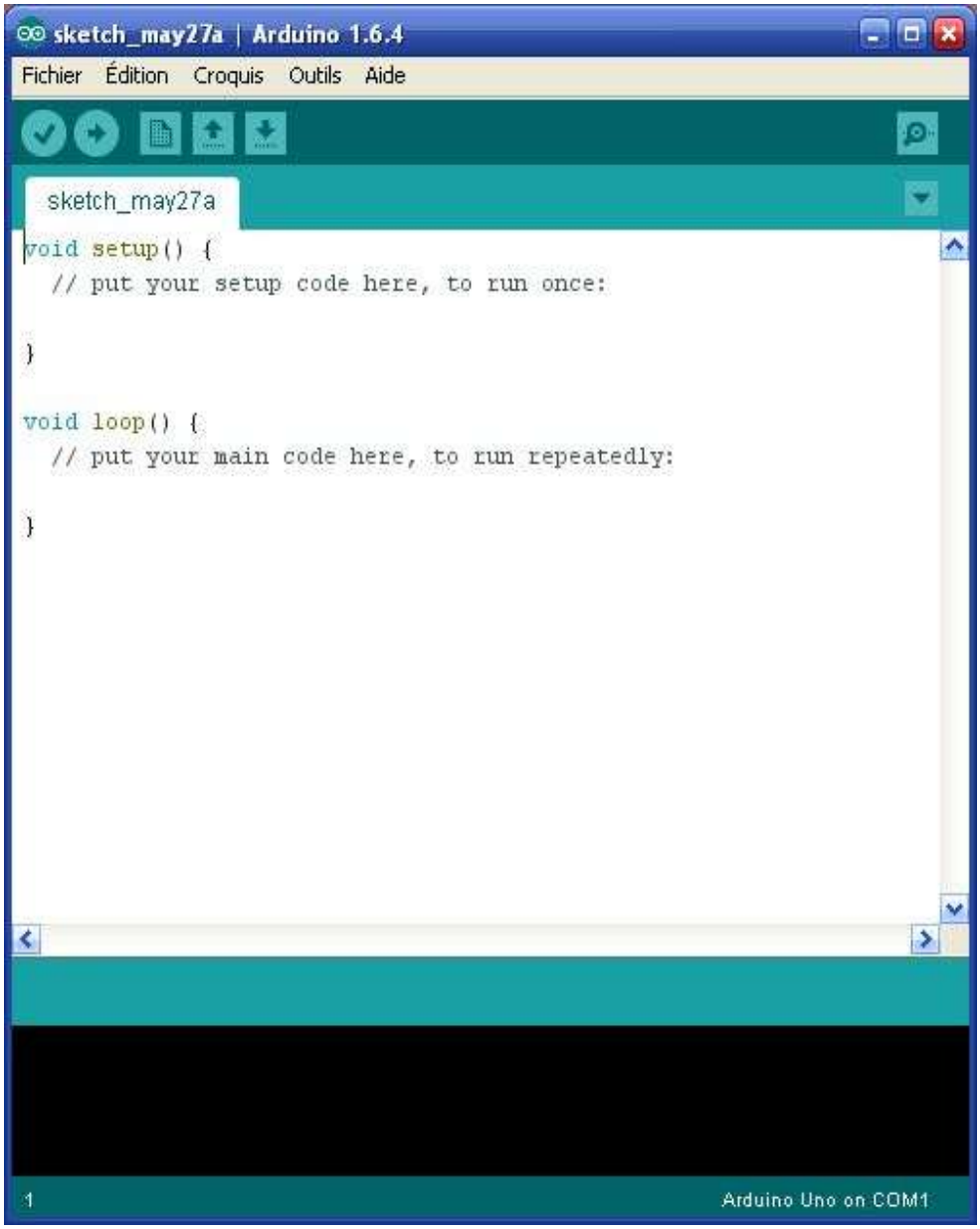



figure IV -12 La fenêtre du logiciel arduino

- **Est un compilateur:**

En informatique, ce terme désigne un logiciel qui est capable de traduire un langage informatique, ou plutôt un programme utilisant un langage informatique, vers un langage plus approprié afin que la machine qui va le lire puisse le comprendre. C'est un peu comme si le patron anglais d'une firme Chinoise donnait des instructions en anglais à un de ses ouvriers chinois. L'ouvrier ne pourrait comprendre ce qu'il doit faire. Pour cela, il a besoin que l'on traduise ce que lui dit son patron. C'est le rôle du **traducteur**. Le compilateur va donc

traduire les instructions du programme précédent, écrites en langage texte, vers un langage dit « machine ». Ce langage utilise uniquement des 0 et des 1. Nous verrons plus tard pourquoi. Cela pourrait être imagé de la façon suivante :




Donc, pour traduire le langage texte vers le langage machine (avec des 0 et des 1), nous aurons besoin de ce fameux compilateur. Et pas n'importe lequel, il faut celui qui soit capable de traduire le langage texte Arduino vers le langage machine Arduino. Et oui, sinon rien ne va fonctionner. Si vous mettez un traducteur Français vers Allemand entre notre patron anglais et son ouvrier chinois, ça ne fonctionnera pas mieux que s'ils discutaient directement.

## IV.5-2 Le logiciel ISIS proteus:

**Proteus** est une suite logicielle destinée à l'électronique. Développé par la société Labcenter Electronics, les logiciels incluent dans Proteus permettent la CAO dans le domaine électronique. Deux logiciels principaux composent cette suite logicielle: ISIS, ARES, PROSPICE et VSM.

### IV.5-2-1 Présentation générale:

Cette suite logicielle est très connue dans le domaine de l'électronique. De nombreuses entreprises et organismes de formation (incluant lycée et université) utilisent cette suite logicielle. Outre la popularité de l'outil, Proteus possède d'autres avantages

- Pack contenant des logiciels facile et rapide à comprendre et utiliser
- Le support technique est performant
- L'outil de création de prototype virtuel permet de réduire les coûts matériel et logiciel lors de la conception d'un projet

**ISIS**

Le logiciel ISIS de Proteus est principalement connu pour éditer des schémas électriques. Par ailleurs, le logiciel permet également de simuler ces schémas ce qui permet de déceler certaines erreurs dès l'étape de conception. Indirectement, les circuits électriques conçus grâce à ce logiciel peuvent être utilisé dans des documentations car le logiciel permet de contrôler la majorité de l'aspect graphique des circuits.

## ARES

Le logiciel ARES est un outil d'édition et de routage qui complètement parfaitement ISIS. Un schéma électrique réalisé sur ISIS peut alors être importé facilement sur ARES pour réaliser le PCB de la carte électronique. Bien que l'édition d'un circuit imprimé soit plus efficiente lorsqu'elle est réalisée manuellement, ce logiciel permet de placer automatiquement les composants et de réaliser le routage automatiquement.

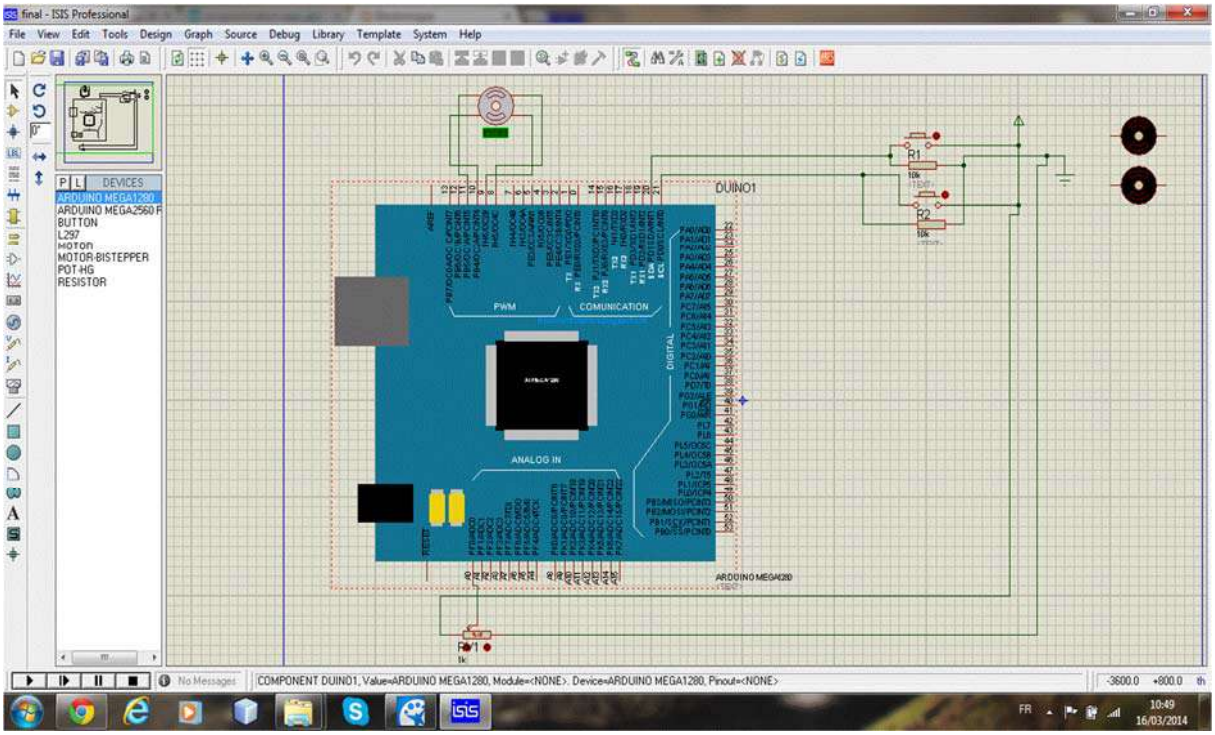



figure IV -13 La fenêtre principale du logiciel isis proteus

## IV.6 partie simulation et programmation:

### IV.6-1 partie simulation:

La figure III -14 présenté le schéma électrique sur logiciel ISIS PROTEUS de ce projet comment on a déjà présenté dans cette mémoire.

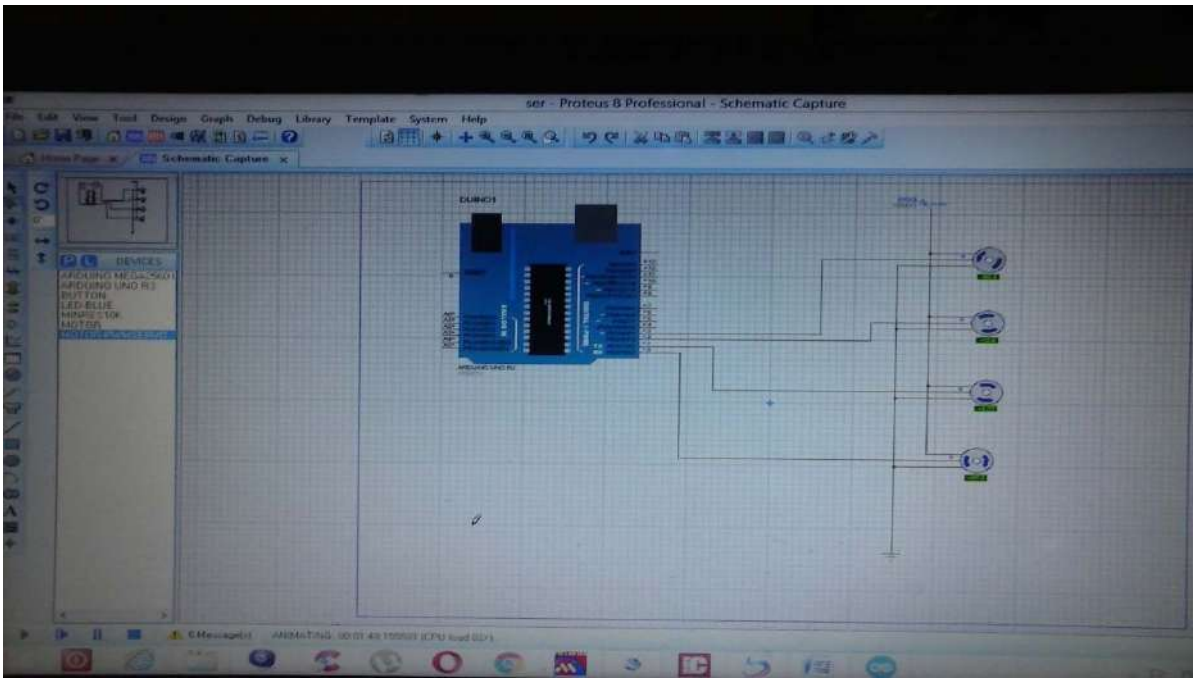



figure IV -14: Schéma électrique de bras manipulateur

### IV.6-2 partie programmation:

La figure III -15 et la figure III -16 le programme principale qui commande l'arduino uno :
Le programme devise en deux partie, **prog principale** et **prog mouvement**.
**Prog mouvement**: indique les fonction qui commande les mouvements des articulation on a six (6) fonction:

- ✓ ouvrir pince
- ✓ fermer pince
- ✓ descendre
- ✓ monter
- ✓ tourne droite
- ✓ tourne gauche

**prog principale:** indique le programme de commande finale la configuration des pin et des fonction.....

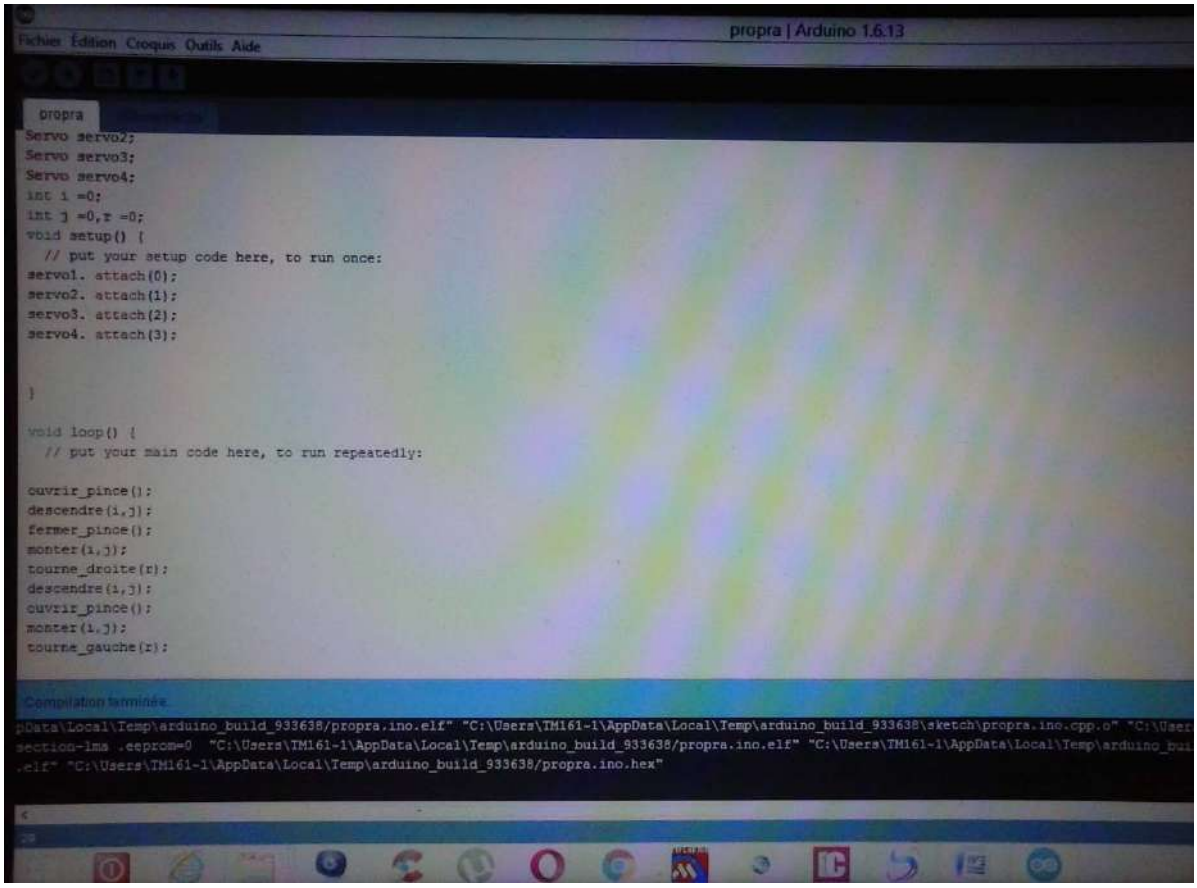



figure IV -15: Programme principal de projet

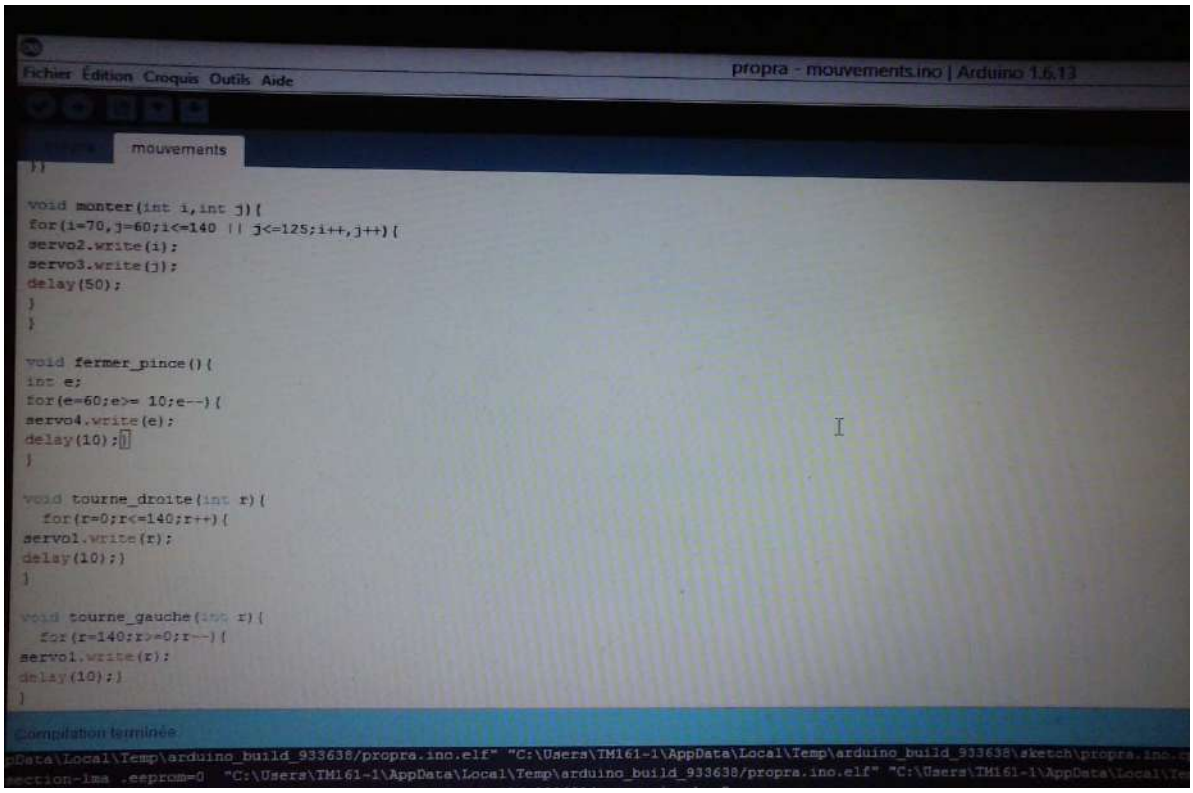



figure IV -16: Programme des fonction mouvements

## IV.6-3: Les programmes

### Le programme des fonction mouvement:

```
void descendre(int i,int j){                    //indique fonction descendre
for(i=140,j=130;i>=70 || j>=80;i--,j--){
servo2.write(i);
servo3.write(j);
delay(50);
}}

void monter(int i,int j){                       // indique fonction monter
for(i=70,j=60;i<=140 || j<=125;i++,j++){
servo2.write(i);
servo3.write(j);
delay(50);
}
}

void fermer_pince(){                            // indique fonction fermer le pince
int e;
for(e=60;e>= 10;e--){
servo4.write(e);
delay(10);}
}

void tourne_droite(int r){                      // indique fonction tourne le bras manipulateur à droite
  for(r=0;r<=140;r++){
servo1.write(r);
delay(10);}
}

void tourne_gauche(int r){                      // indique fonction tourne le bras manipulateur à gauche
  for(r=140;r>=0;r--){
servo1.write(r);
delay(10);}
}

void ouvrir_pince(){                            // indique fonction ouvrir le pince
int e;
for(e=10;e<=60;e++){
servo4.write(e);
delay(10);}}
```

**Le programme principal:**

```
#include <Servo.h>
Servo servo1;                    // création de l'objet servo1
Servo servo2;                    // création de l'objet servo2
Servo servo3;                    // création de l'objet servo3
Servo servo4;                    // création de l'objet servo4
int i =0;
int j =0,r =0;
void setup() {
  // put your setup code here, to run once:
servo1. attach(0);               // attache le servo au pin 1
servo2. attach(1);               // attache le servo au pin 2
servo3. attach(2);               // attache le servo au pin 3
servo4. attach(3);               // attache le servo au pin 4


}

void loop() {
  // put your main code here, to run repeatedly:

ouvrir_pince();
descendre(i,j);
fermer_pince();
monter(i,j);
tourne_droite(r);
descendre(i,j);
ouvrir_pince();
monter(i,j);
tourne_gauche(r);

}
```

## IV.7 conclusion

Dans cette partie nous avons réaliser qu'est ce que on a étudier précédant, on a commencer par les matériaux utiliser et la commenter à chacun de ces matériaux. Nous avons découvert la partie simulation et programmation pour commandé le projet (le bras manipulateur).

## Conclusion générale

L'objectif de notre projet est la réalisation d'un bras manipulateur avec un tapis rotatif. Notre robot se travail sous un programme commander par la carte arduino uno et l'ensemble des servomoteur.

Pour réaliser ce projet, nous avons passé par différentes étapes :

- On a donné une introduction générale sur la robotique.
- Après on a faire des études se notre bras manipulateur.
- Des circuit de commande son utiliser a fin de contrôler les moteur associer. L'ensemble de système perception et commandé par un microcontrôleur qui doit en utilisant les information actuelle, décider l'action à prendre. Pour notre cas, on a utilisé le microcontrôleur ATMega328 la famille de AVR qui intégrer sur l'arduino uno, qui généré le système PWM aux les servomoteur.
- A la fin de travail on a fait une petite réalisation par le boit et les servomoteurs pour:
  - ✓ appliqué la théorie en pratique.
  - ✓ la bien comprendre des mécanisme de ce théorie.
  - ✓ La connaissance sur la programmation

On a déduire après ces études les points suivantes :

- ➢ On a détecté plusieurs types des robot et des bras manipulateurs.
- ➢ Il-y-a une relation entre la partie mécanique et la partie électronique.
- ➢ Il-y-a plusieurs logiciels de programmation et de simulation avec différent langage.
- ➢ La programmation a beaucoup évolué sur le passé et plus souple dans le traitement Nous a permis d'études sur l'utilisation l'arduino uno et comment il fonctionne
- ➢ Nous avons conclu qu'il existe plusieurs méthodes de travail du logiciel qui contrôle le système
- ➢ Nous avons utilisé le langage C, ce qui est plus facile à traiter l'excrétion de code
- ➢ Pour la fabrication d'un bras manipulateur nous avons utilisé les servomoteurs qui contrôle l'angle de rotation par PWM, Et nous avons appris au sujet de ses composants et pièces

A la fin de ce projet grâce au travail contenu on a respect d' amélioré ce travail à travers d' augmentation de la vitesse et bien sur la précision par d' addition les capteur....

# références


- Magistère en électronique option: contrôle industriel (université de Batna)
- Cours de Robotique Fondamentale La cinématique des robots séries (Julien Alexandre dit Sandretto COPRININRIA Sophia-Antipolis ).
- Introduction à la Robotique (Bernard BAYLE).
- Arduino : Premiers pas en informatique embarquée
- Moteur à courant continu (pdf)
- par internet (introduction sur les servomoteurs, wikipedia,..... )
- Microcontrôleur ATML ATMEGA
- Carte ARDUINO UNO Microcontrôleur atmega 328