

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



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

Mémoire de Fin d'Etude DE MASTER ACADEMIQUE

Spécialité : INSTRUMENTATION

Mémoire dirigée par M'BENNAMANE.K

Présenté par
BENAISSA FARIZAYAHYA OUI REZIKA

Thème

ETUDE ET REALISATION D'UNE MAIN ROBOTIQUE

Mémoire soutenu publiquement le 04/07/2018 devant le jury composé de :

Président	M LAGHROUCHE
Encadreur	M BENNAMANE .K
Examineur	M ZIRMI

Dédicaces

JE DÉDIE CE TRAVAIL À MES CHER PARENTS QUI
M'ONT TOUJOURS SOUTENUE,

À MON FRÈRE AHMED ET SON ÉPOUSE LILA
AINSI QUE CES ENFANTS.

À MA CHER SŒUR GHALIA ET SON MARI
MOULOUD ET CES ENFANTS.

À MES FRÈRES MOHAND MERZOUK M'HAND.

À MES GRAND PARENTS MORT ET VIVANT.

À MES ONCLES ET TANTES PATERNELS ET MON
ONCLE ET TANTE MATERNEL.

À MES COUSINS ET COUSINES.

À MES AMIS ET MA BINÔME AINSI QUE SA
FAMILLE.

À TOUS LES ÉTUDIANT DE MASTER ÉLECTRONIQUE
TOUTES LES SPÉCIALITÉS.

2017/2018

YAHIAOUIREZIKA

Dédicaces

JE DÉDIE CE MODESTE TRAVAIL À MES CHER
PARENTS QUI M'ONT TOUJOURS SOUTENUES DURANT
MON PARCOURS.

À MES 2 FRÈRES SALIM ET HAFIDH ET LEURS
ÉPOUSES AINSI QUE LEURS ENFANTS.

À MES GRAND PARENTS ET MES TANTES ET ONCLES
AINSI QUE TOUS MES COUSINS ET COUSINES ET KHALTI
MERIAMA.

À MON ÉPOUXOUSSAMA ET MA BELLE FAMILLE.

À TOUS MES AMIS ET MA BINÔME AINSI QUE SA
FAMILLE.

À TOUS LES ÉTUDIANT DE MASTER ÉLECTRONIQUE
TOUTES LES SPÉCIALITÉS.

2017/2018

BENAISSAFARIZA



Remerciements

Nous tenons tout d'abord à remercier Dieu le tout puissant et miséricordieux, qui nous a donné la force et la patience d'accomplir ce modeste travail.

En second lieu, nous tenons à remercier chaleureusement notre encadreur Monsieur BENNAMANE pour ces précieux conseils et ses aides toute au long de notre travail ainsi que monsieur KHAZEM SALIM pour son aide précieuse pour la réussite de ce modeste travail.

Nous remercions les membres du jury de leur présence et d'avoir accepté d'évaluer et d'examiné notre modeste travail

Nous tenons à exprimer nos sincères remerciements à tous les professeurs qui nous ont enseigné et qui par leurs compétences nous ont soutenu dans la poursuite de nos études.

Enfin, nous tenons également à remercier toutes les personnes qui ont participé de près ou de loin à la réalisation de ce travail ainsi que nos respectable familles.

Merci à tous et à toutes.

Sommaire

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

Chapitre I : généralité sur la robotique

I.1 Historique	2
I.1.1 L'horloge	2
I.1.2 Les automates	3
I.1.3 Intelligence artificielle	3
I.1.4 La première main robotique.....	3
I.2 Définition de la robotique.....	4
I.2.1 Le robot.....	4
I.2.1.1 Structure générale d'un système robotique.....	4
I.2.1.2 Les différents composants d'une structure robotique.....	5
I.2.1.3 Les types de robots	8
I.2.1.4 Les caractéristiques des robots	10
I.2.1.5 domaine d'utilisation des robotiques.....	11
I.2.1.6 L'évolution de la robotique avec le temps	14

Chapitre II : conception de la main robotique

II.1 Architecture de la main robotique	15
II.2 Caractéristique de la main robotique.....	15
II.3 Microcontrôleur ATMEL ATMega328.....	34

Chapitre III : Réalisation de la main robotique

III.1 Montage de la main robotique.....	38
III.2 Les éléments nécessaires pour le fonctionnement de la main	
Robotique.....	39
III.3 Pilotage d'un servomoteur sur Arduino avec un bouton poussoir.....	40
III.4 Etape de la réalisation de la main robotique	41
III.5 Explication du programme.....	43
III.5.1 Structure du programme	44
III.6 Les actions de tests de la main.....	51
III.6.1 le montage final de la main avec tous les composants	
Utilisés	51
III.7 Organigramme de fonctionnement de la main par rapport à un	
bouton poussoir et un servomoteur.....	56
conclusion.....	57

INTRODUCTION

Introduction

La robotique touche aujourd'hui de nombreux secteurs de la vie, elle a extrêmement progressé durant le siècle dernier, notamment dans le domaine médical. La robotique est un ensemble d'études, de techniques de conception, de création et de mise en œuvre.

Le problème de manipulation d'objets est l'un des sujets de recherche qui passionnent depuis longtemps les chercheurs en robotique. Les robots assurant cette tâche et ils s'appellent les robots manipulateurs. On les retrouve dans plusieurs domaines, comme l'industrie, la médecine, l'astronomie, l'arméeetc.

Selon le type de tâches à réaliser et le secteur d'application, le robot aura une architecture mécanique et un système de commande différent.

Dans ce contexte nous nous sommes intéressés à la main robotique. Une main robotique est un robot manipulateur programmable, qui reproduit les gestes d'une main humaine.

Pour réaliser une main robotique, plusieurs outils sont utilisés (la carte arduino, les servomoteurs et les capteurs).

Ce mémoire est organisé comme suite:

- ❖ Le premier chapitre est consacré à une étude générale sur les robots et la structure robotique.
- ❖ Le deuxième chapitre présente la conception de notre propre main robotique.
- ❖ Le troisième chapitre est dédié à la réalisation de la main robotique.

CHAPITRE I

GÉNÉRALITÉS SUR LA ROBOTIQUE

1.1 Historique:

Vers 3500 avant J-C la roue est apparue et considérée comme le premier ouvrage technique significatif de l'histoire de l'humanité. Les automates ont reposé sur des ressorts, engrenage et autres mécanismes, et puis l'informatique est venue changer la donne en permettant de stocker de très large quantité d'information, et de séquences d'action dans une petite puce. Cela a entraîné une sophistication du robot et a permis de doter le robot d'une intelligence artificielle.

1.1.1 L'horloge:

Les clepsydres (horloge à eau) furent probablement inventées vers 1600 avant J-C. En Egypte. De simples bols percés, dont l'écoulement régulier permettait une mesure rudimentaire du temps. Elles furent perfectionnées en 246 avant J-C par Ctésibios, originaire d'Alexandrie, il est parvenu à créer une horloge si précise que son cadran fait exactement un tour par année scolaire. Avant de constituer de véritables automates sophistiqués sous les civilisations chinoises et arabes du premier millénaire. L'horlogerie est aussi le berceau des premiers automates, le plus ancien conservé étant le coq automate des Trois Rois de l'horloge astronomique de la cathédrale Notre-Dame de Strasbourg.

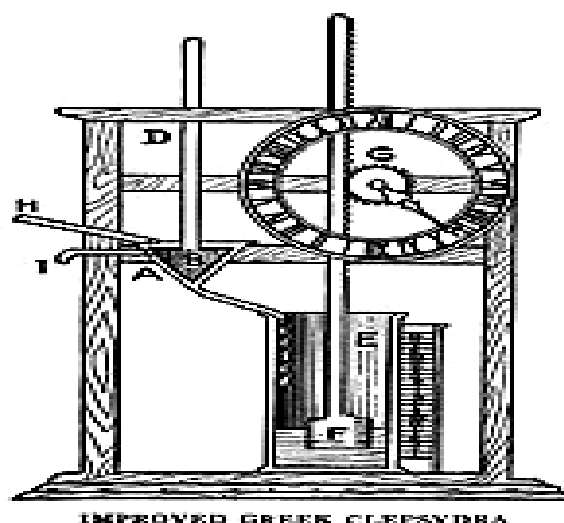


Figure 1: clepsydre.

1.1.2 Les automates:

Le 18^{ème} siècle apparaît comme l'âge d'or des automates, un des automates les plus surprenants est certainement le canard digérateur de Jacques de Vaucanson* (1738). Capable de boire, manger, cancaner et digérer comme un véritable animal, il fut malheureusement détruit dans un incendie au milieu du XIX^{ème} siècle et il n'en reste que des photographies.

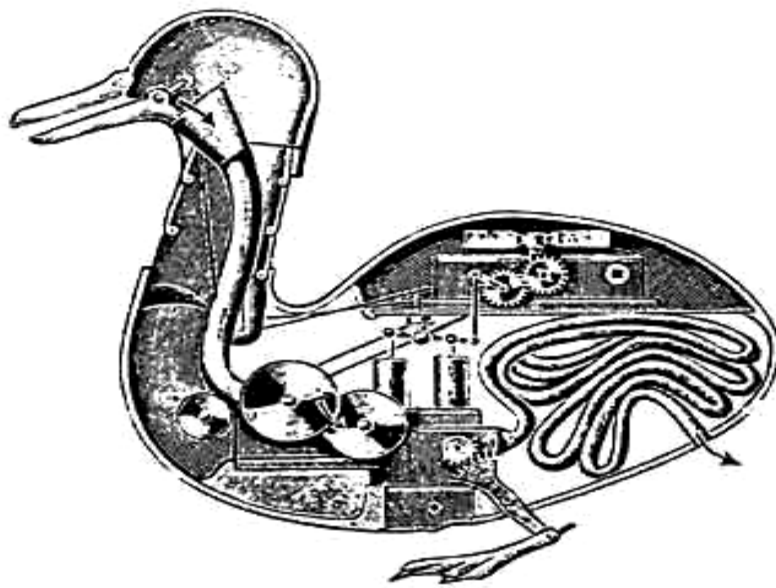


Figure 2 : le canard digérateur

1.1.3 Intelligence artificielle:

Dans la droite ligne de la cybernétique, l'objectif de rendre les machines intelligentes a mené Alan

Turing à définir en 1950 une méthode afin de tester la présence d'une conscience ou d'une intelligence au sein d'une machine

1.1.4 La première main robotique:

C'est vers les années 80 que se réalise la première main à vue le jour. Jacobsen 1986, Okada 1979, 1982 puis Salisbury et Craig 1982 ont proposé une

architecture détecteur possèdent chacune trois doigts dont la paume ne participe pas à la prise, elles ont été actionnées par des moteurs électriques en utilisant des tendons comme moyen de transmission entre les doigts et les moteurs.



Figure 3 : première main

1.2 Définition de la robotique:

La robotique est un ensemble de méthodes et de techniques (mécanique, électronique, automatique, informatique) utilisées pour la conception et la programmation de robots pour faciliter la vie humaine.

1.2.1 Le robot:

Le terme robot provient d'une pièce théâtrale de Karel Capek, il est dérivé du mot robota qui signifie travail forcé en tchèque. Un robot est une machine qui imite une créature intelligente, il peut être totalement mécanique composé de un ou plusieurs systèmes embarqués.

1.2.1.1 Structure générale d'un système robotique:

Un robot est un système alimenté en énergie évoluant dans un environnement statique ou dynamique, il est constitué d'un ensemble de composants ayant chacun un rôle spécifique.

1.2.1.2 Les différents composants d'une structure robotique:

1.2.1.2.1 Système mécanique articulé:

Un système mécanique articulé est un ensemble de solides reliés entre eux par des liaisons (rotule, pivot, glissière, sphérique...) animé avec des joints mécaniques. Dans le MSA certaines liaisons sont motorisées, on parlera de liaison active, c'est le cas des liaisons pivot. D'autres sans motorisation sont appelées passives tel que les liaisons rotule.

1.2.1.2.2 Actionneurs :

Un actionneur est un dispositif qui transforme l'énergie délivrée par l'interface de puissance en énergie utilisable, modifier le comportement ou l'état d'un système ou bien exécuter les tâches d'un système automatisé -

a. Actionneur électrique:

C'est un actionneur qui permet à partir d'un courant électrique de faire tourner un mécanisme on observe deux types :

- Moteur à courant continu (moteur DC)
- Moteur pas à pas

b. Actionneur pneumatique:

Un actionneur pneumatique convertit une énergie d'entrée pneumatique en une énergie utilisatrice mécanique, on distingue deux types :

- Vérin simple effet
- Vérin double effet

c. Actionneur hydraulique :

Dans un circuit l'actionneur hydraulique constitue l'outil indispensable pour convertir l'énergie hydraulique en énergie mécanique.

I. 2.1.2.3 Capteur (organe de perception)

Un capteur est un dispositif transformant l'état d'une grandeur physique observé en une grandeur utilisable, telle qu'une tension électrique....etc

Le capteur se distingue de l'instrument de mesure par le fait qu'il ne s'agit que d'une simple interface entre un processus physique et une information manipulable.

✓ Différents types de capteurs.

En fonction de la caractéristique électrique de la grandeur de sortie, on peut classer les capteurs en deux grandes familles : les capteurs passifs et les capteurs actifs.

Cette classification a une influence sur le conditionneur qui lui est associé.

- **Les capteurs passifs :**

Dans la plupart des cas, les capteurs passifs ont besoin d'une énergie extérieure pour fonctionner (comme dans le cas des jauges de contraintes, thermistances...), ils sont souvent modélisés par une impédance. Une variation du phénomène physique étudié (mesuré) engendre une variation de l'impédance. Il faut leur appliquer une tension pour obtenir un signal de sortie.

Le capteur se comporte en sortie comme un dipôle passif qui peut être résistif, capacitif ou inductif. En fonction du mesurande, on utilise plusieurs effets pour réaliser la mesure.

- **Les capteurs actifs :**

Lorsque le phénomène physique qui est utilisé pour la détermination du mesurande effectue directement la transformation en grandeur électrique, on est en présence d'un capteur actif.

C'est la loi physique elle-même qui relie le mesurande et grandeur électrique de sortie. La sortie du capteur est assimilée à un générateur. C'est un dipôle actif qui peut être du type courant, tension ou charge électrique Q en coulombs. Certains principes physiques peuvent être mis en jeu.

1.2.1.2.4 Effecteur (organe terminal)

En robotique c'est l'outil mis en mouvement (ressort, vérin, moteur). Certaines machines peuvent devenir des effecteurs de robotique au prix d'un travail d'adaptation permettant de

- Compenser certaine imprécision ou dispersion dans les caractéristiques des objets extérieurs ou du robot lui-même.
- Présenter une certaine flexibilité pour se prêter à des tâches diversifiées

1.2.1.2.5 Système de traitement (calculateur)

C'est l'organe qui gère l'ensemble des tâches, il admet trois rôles essentiels

- Collecte de l'information venant des capteurs
- Prise de décision en partant d'une tâche et en tenant compte des données du système et de l'environnement
- Le rôle de la communication

Le contrôle de robot incluent les fonctions qui permettent au robot d'apprendre, et d'être programmé pour une tâche spécifique puis de l'exécuter. la séquence de mouvement, le type de mouvement et l'interaction avec l'équipement externes sont toutes des parties de la fonction de contrôle

1.2.1.3 Les Types de robots

Il y'a plusieurs types de robots

➤ Robots mobiles

Un robot mobile est un système mécanique, électronique et information agissant physiquement sur son environnement en vue d'atteindre un objectif qui lui a été assignée. Cette machine est polyvalente et capable de s'adapter à certaines variations de ses conditions de fonctionnement. Elle est dotée de fonction de perception, de décision et d'actions.

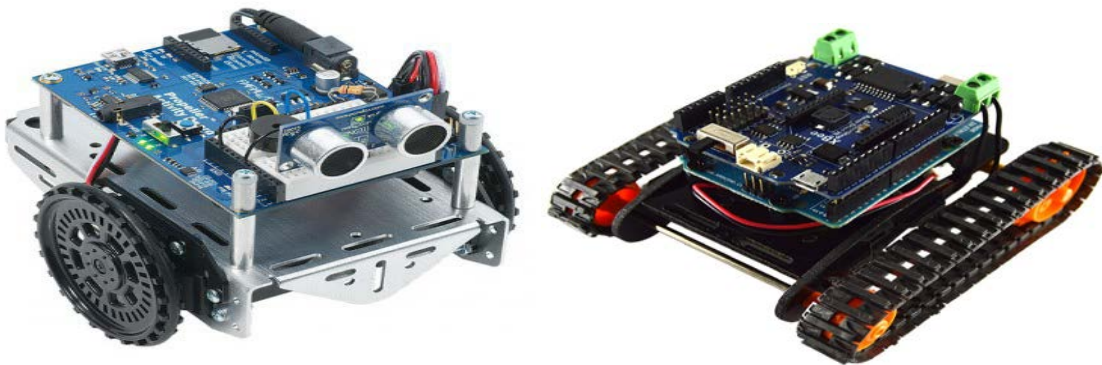


Figure 4 : robots mobiles

➤ Robots sous-marins

Un robot sous-marins autonome, est un robot qui se déplace dans l'eau de manière autonome contrairement au ROV.

ROV : on pourrait le traduire par « véhicule sous marins téléguidées ».

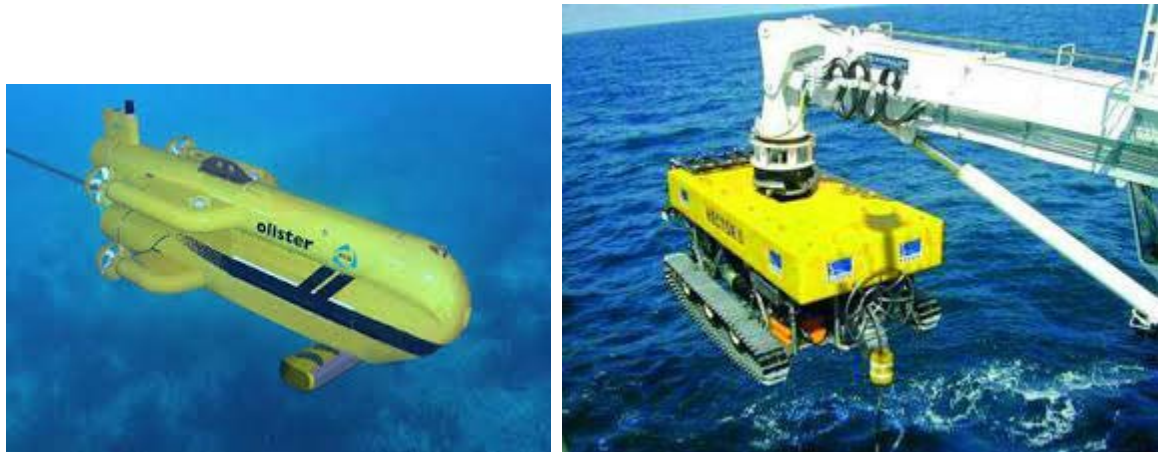


Figure 5 : ROV

➤ Robots volants

Un drone désigne un aéronef sans pilote. Il peut avoir un usage civil ou militaire. Les données sont utilisées au profit des forces armées ou de sécurités.



Figure 6 : robots volants

I.2.1.4 Les Caractéristiques des robots

Niveau d'autonomie

- **Robots télécommandés** : aucune autonomie, les commandes sont envoyées au robot par l'utilisateur
- **Robots semi-autonomes** : l'utilisateur intervient en cas de panne ou de situations non-prévues
- **Robots totalement autonomes** : l'utilisateur n'intervient jamais.

Environnement

Les agents évoluent généralement dans des environnements :

- dynamiques,
- incertains,
- partiellement observables, etc...

Les robots n'ont qu'une observabilité limitée de l'environnement dans lequel ils évoluent :

- vision locale de l'environnement,
- information sur les autres robots,
- erreurs et imprécision sur les capteurs.

Par ailleurs, les environnements peuvent être difficilement accessibles et les erreurs irrémédiables : nécessite d'avoir recours à la simulation avant l'exécution.

La simulation offre la possibilité d'étudier et de tester le comportement des robots.

Interactions

- ***Définition***

Une interaction est une mise en relation dynamique de plusieurs individus (Robots et/ou humains).

Certaines tâches impliquent qu'un groupe de robots travaille de concert, on fait alors appel à des méthodes de coordination.

- Escorte de robots
- Interaction Homme Robot (HRI)

1.2.1.5 Domaine d'utilisation des robots

- **La robotique industrielle**

La robotique industrielle est officiellement définie par l'ISO, comme un contrôle automatique, reprogrammable dans trois ou plusieurs axes. Les applications typiques incluent des robots de soudage de peinture et d'assemblage. Les robots industriels sont très utilisés en automobiles.



Figure 7 : robots industriels

➤ **La robotique domestique**

Un robot domestique est un robot de services personnels, utilisés Pour des tâches ménagères. Jusqu'a présent, il y a que rare modèle. Les robots domestiques sont utilisés par exemple en vaisselle, en nettoyage et en cuisine.



Figure 8 : robots domestiques

➤ **La robotique 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 9 : robots militaires

➤ **La robotique de transport**

Un système automatique de transport fonctionne de manière automatique, Il peut être guidé dans un parcours contraint (véhicule sur rails) ou plus libre voiture autonome.



Figure 10 : robot de transport

➤ **La robotique médicale**

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 neuromotrice. Du fait des contraintes importantes en matière de sécurité, ce type de robot est en général doté d'un faible niveau d'autonomie.



Figure 11 : robots médicales

1.2.1.6 L'évolution de la robotique avec le temps

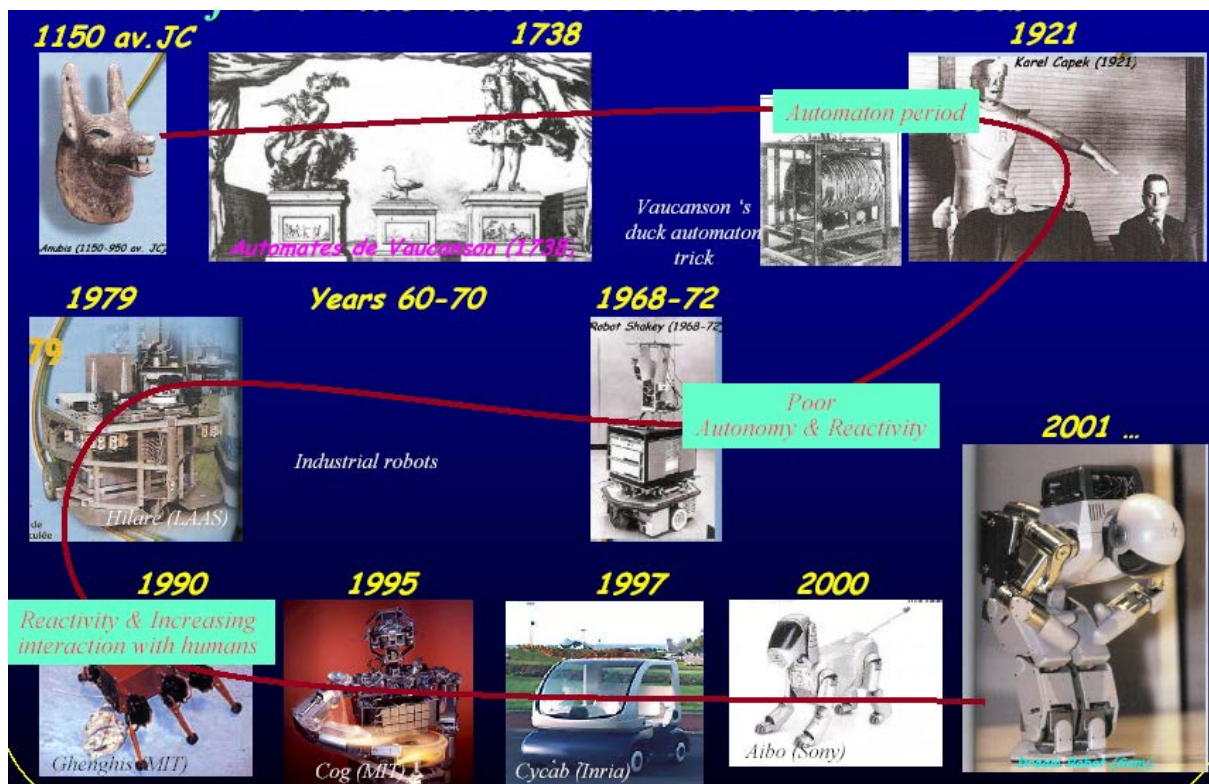


Figure 12 : l'évolution de la robotique avec le temps

CHAPITRE II

CONCEPTION DE LA MAIN ROBOTIQUE

II.1 Architecture de la main robotique

Pour faciliter l'utilisation de la main robotique nous l'avons confectionné en deux blocs communicants avec des fonctionnalités bien précises

- Le premier bloc est la partie émettrice (la commande)
- Le deuxième bloc est la partie réceptrice (l'exécution)
- La communication entre les deux blocs est sans fil

II.2 Caractéristique de la main robotique

Une main robotique possède plusieurs caractéristiques. Et ce pour objectif d'imiter une main humaine. Pour réaliser ce travail nous faut un matériel et un logiciel.

- **Matériel**
 - ✓ **partie émettrice (la commande)**
 - ❖ **Arduino nano**

La carte Arduino Nano possède des connecteurs au dos qui permettent de l'enficher facilement sur une plaque d'essais.

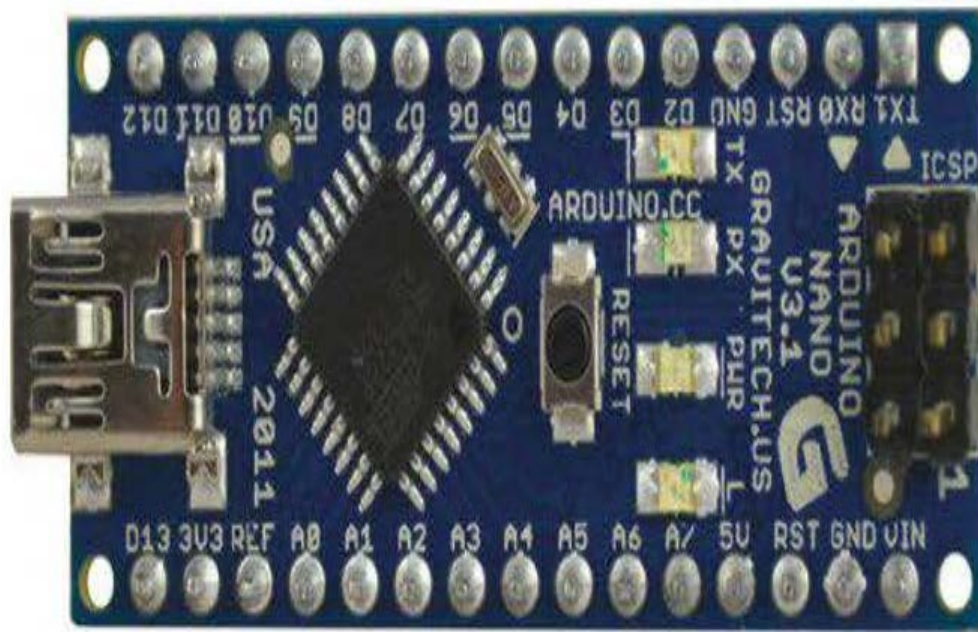


Figure 13:arduino nano

Voici un résumé des spécifications de la carte ArduinoNano:

Catégorie	Valeur
Microcontrôleur	ATmega 168 ou 328
Fréquence d'horloge	16 MHz
Tension de service	5 V
Tension d'entrée (recommandée)	7-12 V
Tension d'entrée (limites)	6-20 V
Ports numériques	14 entrées et sorties (6 sorties commutables en MLI)
Ports analogiques	8 entrées analogiques
Courant maxi. par broche d'E/S (c.c.)	40 mA
Mémoire	ATmega 168 : 16 Ko mémoire Flash 1 Ko SRAM 512 octets d'EEPROM ATmega 328 : 32 Ko mémoire Flash 2 Ko SRAM 1 Ko EEPROM
Chargeur d'amorçage	2 Ko (en mémoire Flash)
Interface	USB
Dimensions	1,9 cm × 4,3 cm
Prix (approximatif)	40 €

Figure 14: spécifications de la carte arduino

Avantages et les Inconvénients de la carte:

A. Avantages:

- encombrement réduit.
- peut être fixée directement sur la plaque d'essais.

B. Inconvénients:

- il n'est pas possible d'utiliser des shields (protection).

❖ *flexsensor*

Un capteur "Flex Sensor" est formé par deux fines couches de métal séparées par un polymère conducteur qui se déforme quand on le plie, modifiant la résistance aux bornes des deux couches métalliques.

Le capteur est souple, de taille réduite : extrêmement mince, la longueur du modèle le plus courant (12 centimètres) le rend idéal pour équiper un gant.

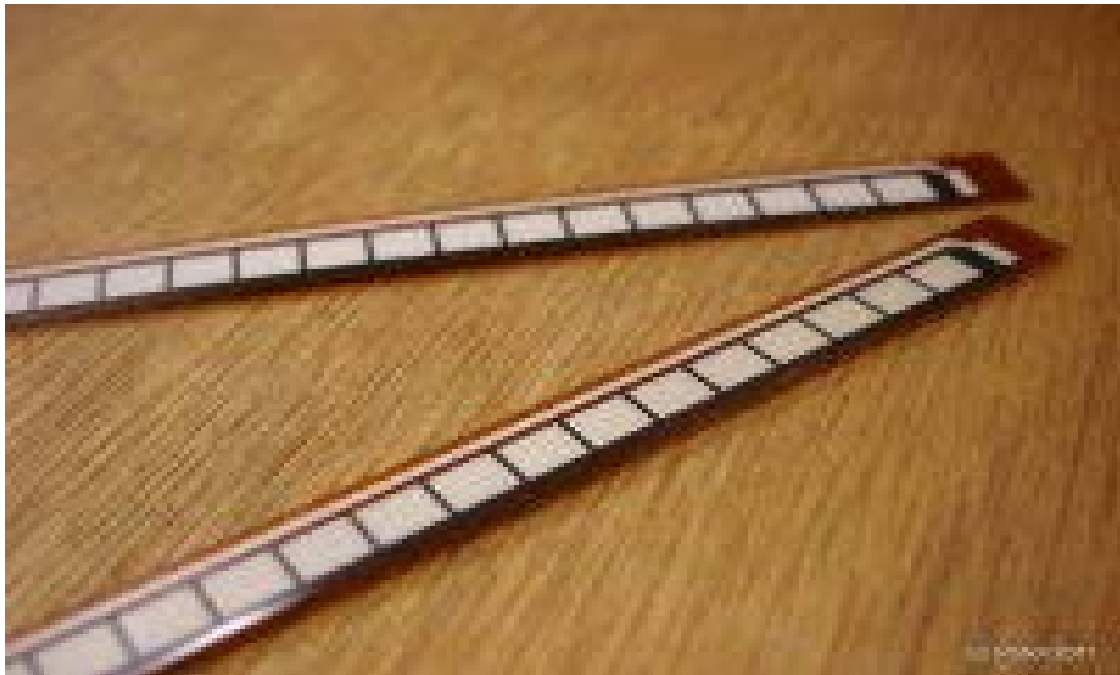


Figure 15:flexsensor

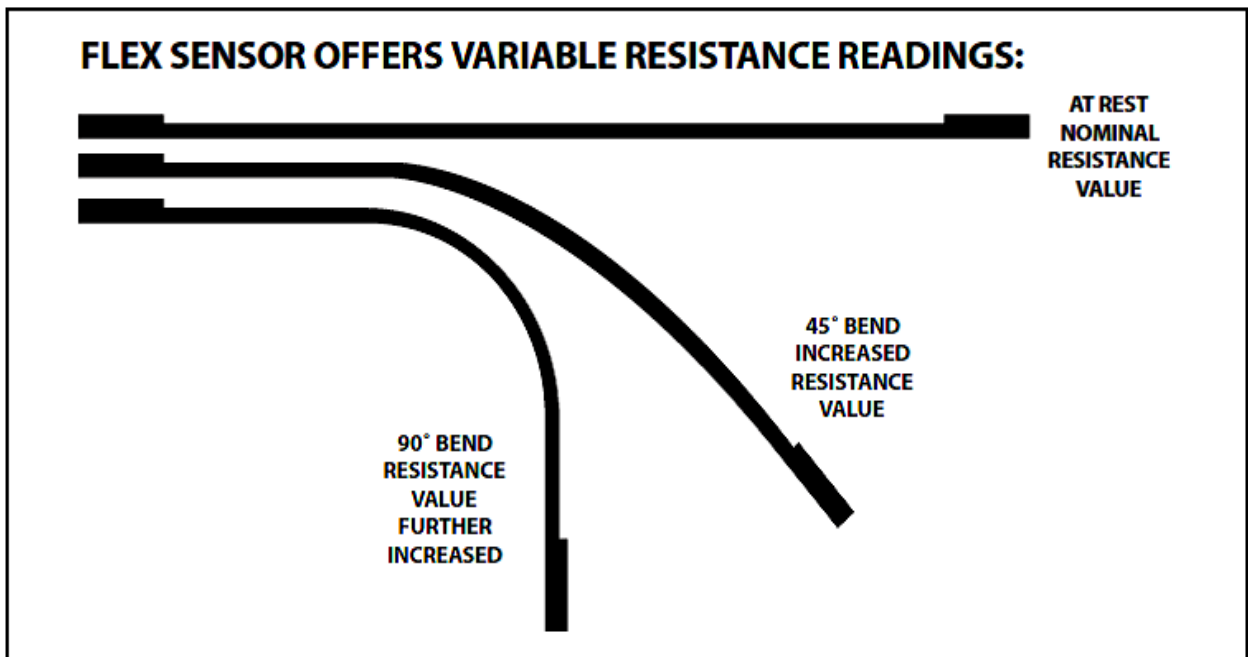


Figure 16 : détails internes du flexsensor

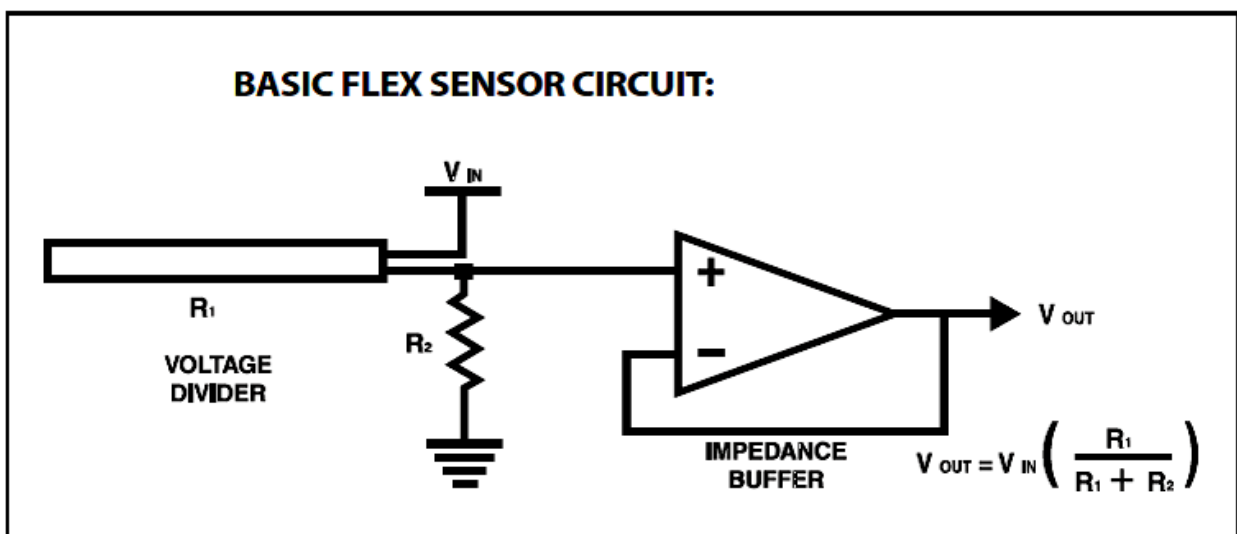


Figure17: circuit basic du flexsensor

✓ partie réceptrice du signal

❖ **Arduinouno**

la carte Arduino Uno est une carte à microcontrôleur basée sur l'ATmega 328

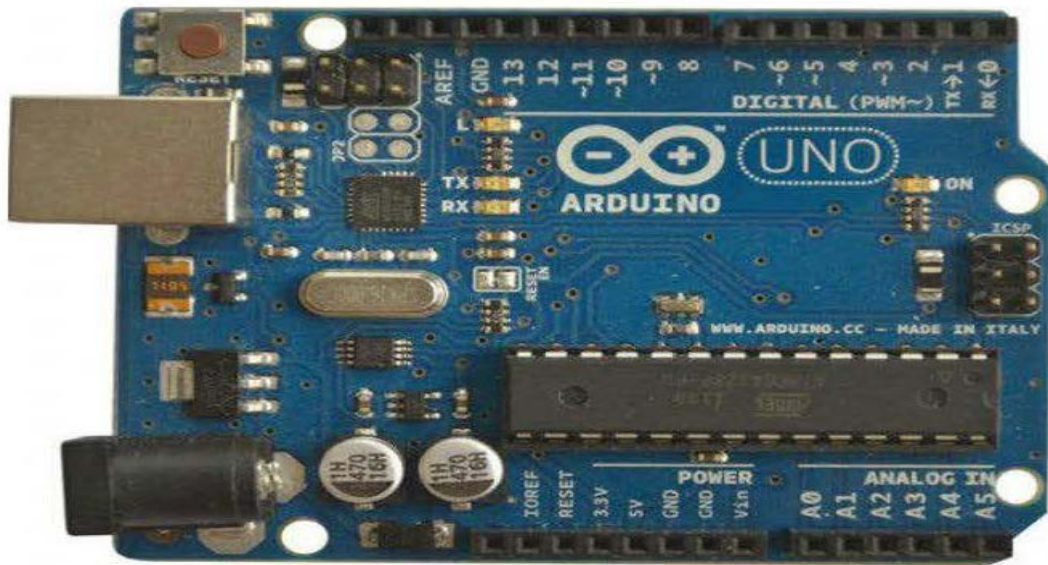


Figure 18: arduinouno

Voici un resume des specifications de la carte ArduinoUno :

Catégorie	Valeur
Microcontrôleur	ATmega 328
Fréquence d'horloge	16 MHz
Tension de service	5 V
Tension d'entrée (recommandée)	7–12 V
Tension d'entrée (limites)	6-20 V
Ports numériques	14 entrées et sorties (6 sorties commutables en MLI)
Ports-analogiques	6 entrées analogiques
Courant maxi. par broche d'E/S (c.c.)	40 mA
Courant maxi. par broche 3,3 V	50 mA
Mémoire	32 Ko Flash, 2 Ko SRAM, 1 Ko EEPROM
Chargeur d'amorçage	0,5 Ko (en mémoire Flash)
Interface	USB
Dimensions	6,86 cm × 5,3 cm
Prix (approximatif)	24 €

Figure 19 : spécifications de la carte uno

Les avantages et les inconvénients de la carte :

A. Avantages:

- nombre suffisant de broches d'entrées-sorties pour les projets élémentaires;
- vaste choix de *shields*;
- bon marché

B. Inconvénients:

- nombre insuffisant de broches d'entrées-sorties pour les projets ambitieux;
- la mémoire disponible risque d'être un peu juste pour les gros projets;
- ne peut pas être utilisée comme hôte USB pour simuler un clavier ou une souris, par exemple.

❖ Servomoteur

Le servomoteur est un composant électrique qui nous permet des déplacements angulaires. Le pilotage du servomoteur se fait avec l'arduino. Il nous permet de nous positionner avec précision sur un angle. La plage de l'angle est de 0 à 180 degrés. Il existe différents types de servomoteur, mais ils fonctionnent de la même manière. Voici une image qui illustre quelques types



Le servomoteur est composé de plusieurs éléments visible et non visible

a. Eléments visibles

- Les fils qui sont au nombre de trois
 - Rouge pour l'alimentation positive (4.5v à 6v en général)
 - Noir ou marron pour la masse (0v)
 - Orange ou jaune ou blanc entrée du signal de commande
- L'axe de rotation sur lequel est monté un accessoire en plastique ou en métal
- Le boîtier qui le protège

b. Eléments non visibles

- Un moteur à courant continu
- Des engrenages pour former un réducteur (en plastique ou en métal)

- Un capteur de position de l'angle d'orientation de l'axe (exemple : un potentiomètre)
- Une carte électronique pour le contrôle de la position de l'axe et le pilotage du moteur a courant continu

Vu la diversité des tailles et des types de nous avons choisie d'utilisé cinq mini servomoteur

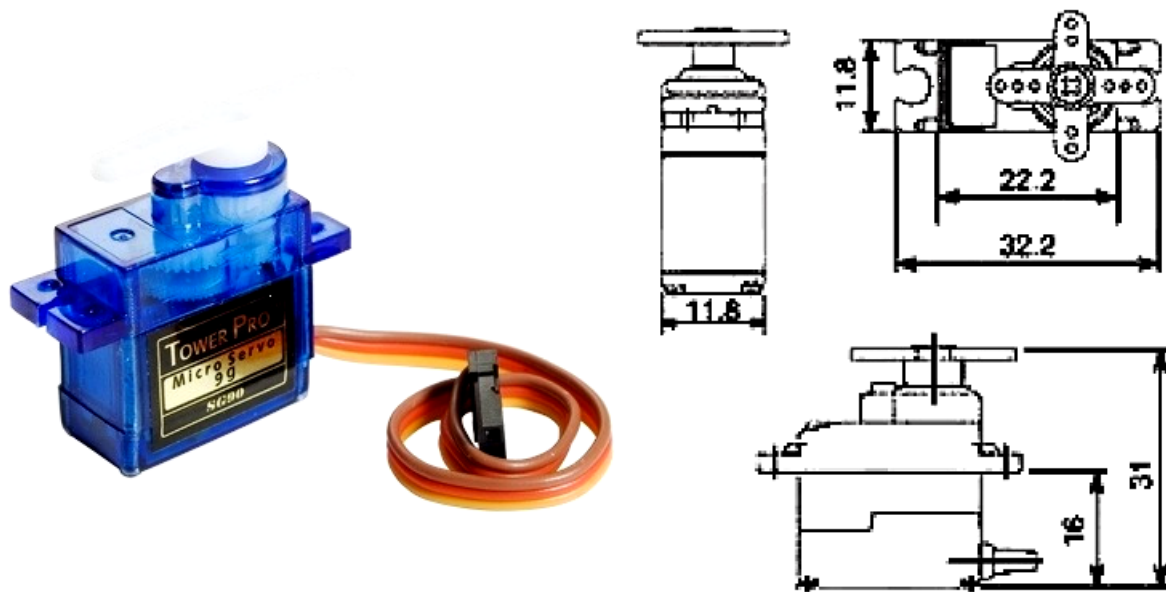


Figure 20: servomoteur

Voici laFiche technique du servomoteur tower pro sg90 <<micro servo>>:

- Dimension : 22 mm x 11.5 mm x 22.5 mm
- Poids net: 9 g
- Matériel principale plastique
- Couleur bleu
- Vitesse de fonctionnement: **0,12 secondes / 60 degrés** (4.8V à vide)
- Amplitude de rotation **180°**
- Couple de blocage (4.8V) : 17,5 oz / in (**1 kg/cm**)

- Plage de température: -30 à +60°C
- Dead band width: 7usec
- Tension de fonctionnement: 4V ~ 6V
- Convient pour tout type de jouet radio-commandé, modèle réduit, robots...
- Moteur coreless.
- Double roulement à billes
- Fil de connexion 150 mm, connecteur 3 fil standard.

✓ *La liaison des deux blocs :*

❖ **NRF24:**

Le NRF24L01 + est une solution sans fil unique pour les applications, à piles compactes avec des exigences strictes sur la vie de la batterie et le coût. L'émetteur-récepteur fonctionne dans la licence libre de la bande des 2,4GHz ISM. Il est spécialement conçu pour deux types d'applications: les périphériques PC et l'application ultra basse puissance. Pour les applications de capteurs, la consommation d'énergie, ultra faible et une gestion avancée de l'alimentation permettent des durées de vie de la batterie jusqu'à plusieurs années.



Figure 21: NRF24

Caractéristiques	
Bande de fréquence	ISM 2,4 GHz (2,4000 - 2,4835)
Sur l'air de débit de données	250 kbps, 1 Mbps ou 2 Mbps
Modulation	GFSK
Bande passante du canal	1MHz pour 2MHz mode 1 Mbps pour le mode 2 Mbps
canaux RF	126
Puissance de sortie	Programmable: 0, -6, -12 ou -18dBm
cristal externe	16MHz \pm 60ppm
Interface hôte	Haut débit SPI (jusqu'à 8 Mbps)
TX et RX FIFOs	3 32 octets séparés TX et RX FIFOs
Matériel couche de liaison	Amélioration ShockBurst TM <ul style="list-style-type: none"> - assemblage de paquets automatique (préambule, Adresse, CRC) - Détection et validation automatique emballé - longueur de paquet dynamique (1 à 32 octets) - reconnaissance automatique avec une charge utile - Retransmettre automatique - 6 canal de données MultiCilver TM
Options de l'emballage	RoHS 4x4mm 20 broches QFN

Figure 22: caractéristiques du NRF24

Caractéristiques électriques		
Paramètre	Valeur	Unité
Tension d'alimentation minimale	1.9	V
Puissance de sortie maximale	0	dBm
Débit d'air-données maximum	2	mbps
Sensibilité à 2Mbps	- 82	dBm
Sensibilité à 1Mbps	- 85	dBm
Sensibilité à 250kbps	- 94	dBm
Courant d'alimentation, TX à 0dBm	11.3	mA
Le courant d'alimentation, RX à 2Mbps	13,5	mA
Le courant d'alimentation, RX à 1 Mbps	13.1	mA
Supplu courant, RX à 250 kbps	12.6	mA
Courant d'alimentation, mode d'alimentation vers le bas	900	n / a
Courant d'alimentation, mode veille	26	pA
Température de fonctionnement	- 40-85	° C

Figure 23 : caractéristiques électrique du NRF24

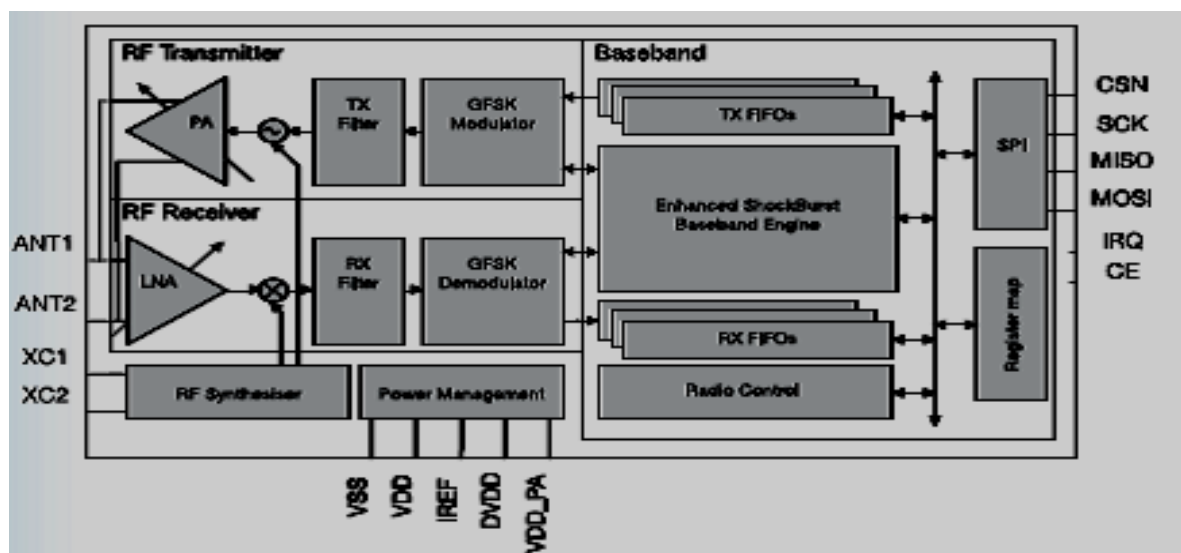


Figure 24 : schéma interne du NRF 24

- **Logiciel**

Dans nos deux blocs nous avons utilisé le logiciel arduino.

❖ ***Présentation***

Le logiciel de programmation des modules Arduino est une application Java, libre et multiplateforme, 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).

❖ ***langage de programmation***

avec arduino nous devons utiliser un code minimal lorsque l'on crée un programme ce code permet de diviser le programme en deux grosse parties « setup » et « loop »



Figure 25: interface de programmation arduino « sketch »

La fonction `setup()` est appelée au démarrage du programme. Cette fonction est utilisée pour initialiser les variables, le sens des broches, les librairies utilisées. La fonction `setup` n'est exécutée qu'une seule fois, après chaque mise sous tension ou reset (réinitialisation) de la carte Arduino.

✓ Syntaxe

```
voidsetup()

{

}
```

✓ Exemple

```
intbuttonPin = 3; // déclaration d'une variable globale

voidsetup() // fonction setup - début de l'exécution du programme
{
  Serial.begin(9600);
  pinMode(buttonPin, INPUT);
}

voidloop() // fonction loop - est exécutée en boucle
    // une fois que la fonction setup a été exécutée
{
  // ...
}
```

La fonction setup() , même vide, est obligatoire dans tout programme Arduino

Après avoir créé une fonction `setup()`, qui initialise et fixe les valeurs de démarrage du programme, la fonction `loop ()` (boucle en anglais) fait exactement ce que son nom suggère et s'exécute en boucle sans fin, permettant à votre programme de s'exécuter et de répondre. Utiliser cette fonction pour contrôler activement la carte Arduino.

✓ *Syntaxe*

```
void loop()
{
}
```

✓ *Exemple*

```
int buttonPin = 3;

// la fonction setup initialise la communication série
// et une broche utilisée avec un bouton poussoir

void setup()
{
  beginSerial(9600);
  pinMode(buttonPin, INPUT);
}

// la fonction loop teste l'état du bouton à chaque passage
// et envoie au PC une lettre H si il est appuyé, L sinon.
```

```
void loop()
{
if (digitalRead(buttonPin) == HIGH)
serialWrite('H');
else
serialWrite('L');
delay(1000);
}
```

La fonction loop() est obligatoire, même vide, dans tout programme.

❖ *La structure d'un programme*

Un programme Arduino comporte trois parties :

1. la partie déclaration des variables (optionnelle)
2. la partie initialisation et configuration des entrées/sorties : la fonction **setup ()**
3. la partie principale qui s'exécute en boucle : la fonction **loop ()**

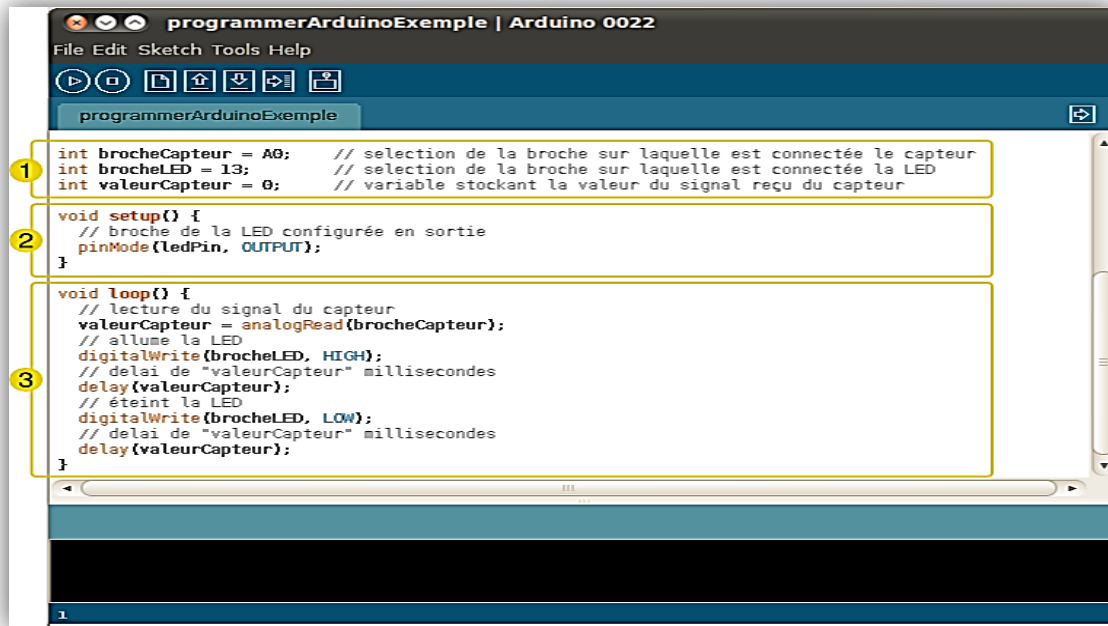


Figure 26: structure d'un programme

Dans chaque partie d'un programme sont utilisées différentes instructions issues de la syntaxe du langage Arduino.

✓ *Coloration syntaxique*

Lorsque du code est écrit dans l'interface de programmation, certains mots apparaissent en différentes couleurs qui clarifient le statut des différents éléments :

- En **orange**, apparaissent les mots-clés reconnus par le langage Arduino comme des **fonctions** existantes. Lorsqu'on sélectionne un mot coloré en orange et qu'on effectue un clic avec le bouton droit de la souris, l'on a la possibilité de choisir « Find in référence » : cette commande ouvre directement la documentation de la fonction sélectionnée.
- En **bleu**, apparaissent les mots-clés reconnus par le langage Arduino comme des **constantes**.

➤ En **gris**, apparaissent les **commentaires** qui ne seront **pas exécutés dans le programme**. Il est utile de bien commenter son code pour s'y retrouver facilement ou pour le transmettre à d'autres personnes. L'on peut déclarer un commentaire de deux manières différentes :

- dans une ligne de code, tout ce qui se trouve après « // » sera un commentaire.
- l'on peut encadrer des commentaires sur plusieurs lignes entre « /* » et « */ ».

✓ *Ponctuation*

Le code est structuré par une ponctuation stricte :

- **toute ligne** de code se termine par un point-virgule « ; »
- le contenu d'une **fonction** est délimité par des accolades « { » et « } »
- les **paramètres** d'une fonction sont contenus pas des parenthèses « (» et «) ».

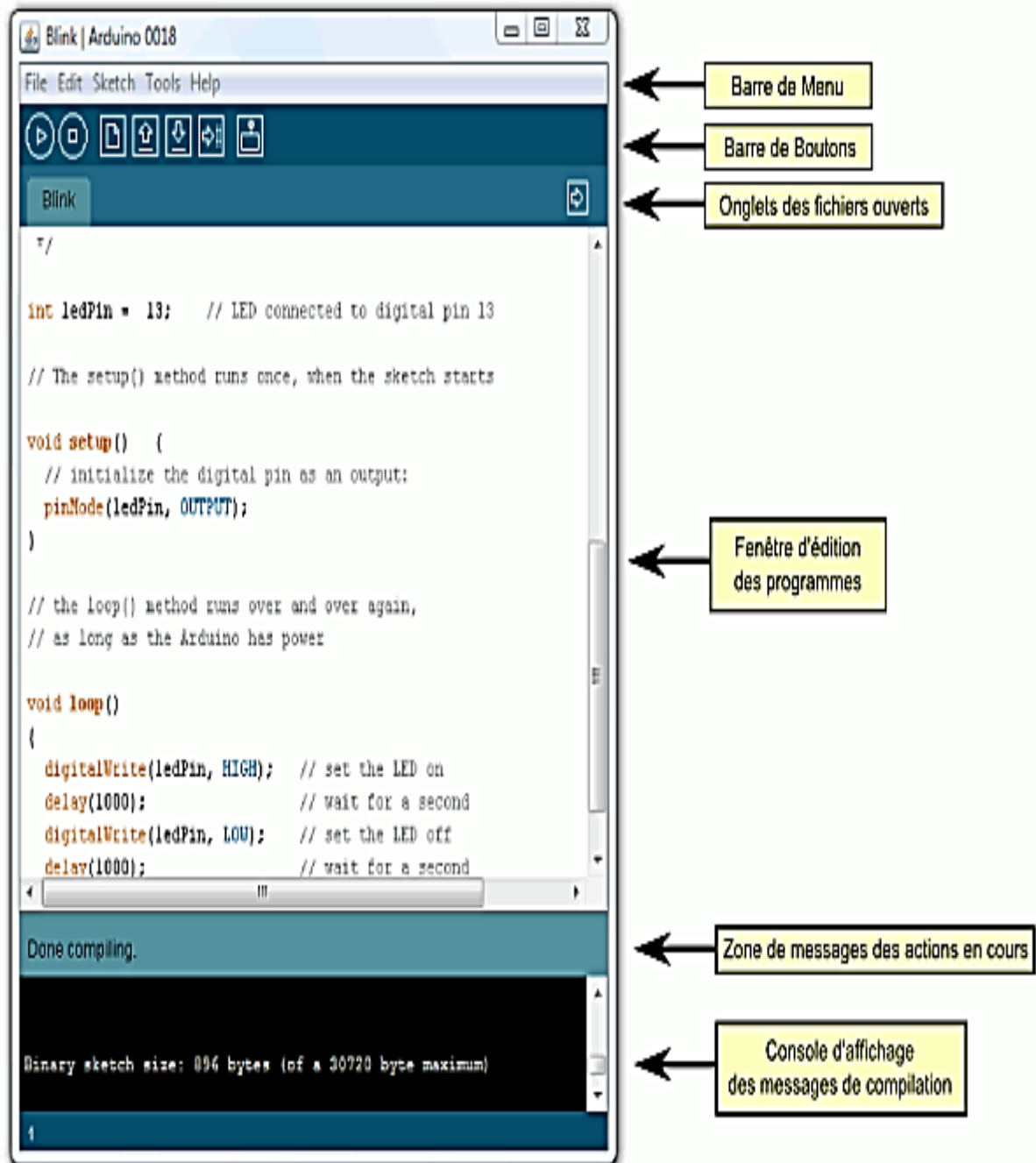


Figure 27: différentes zones de l'interface du logiciel

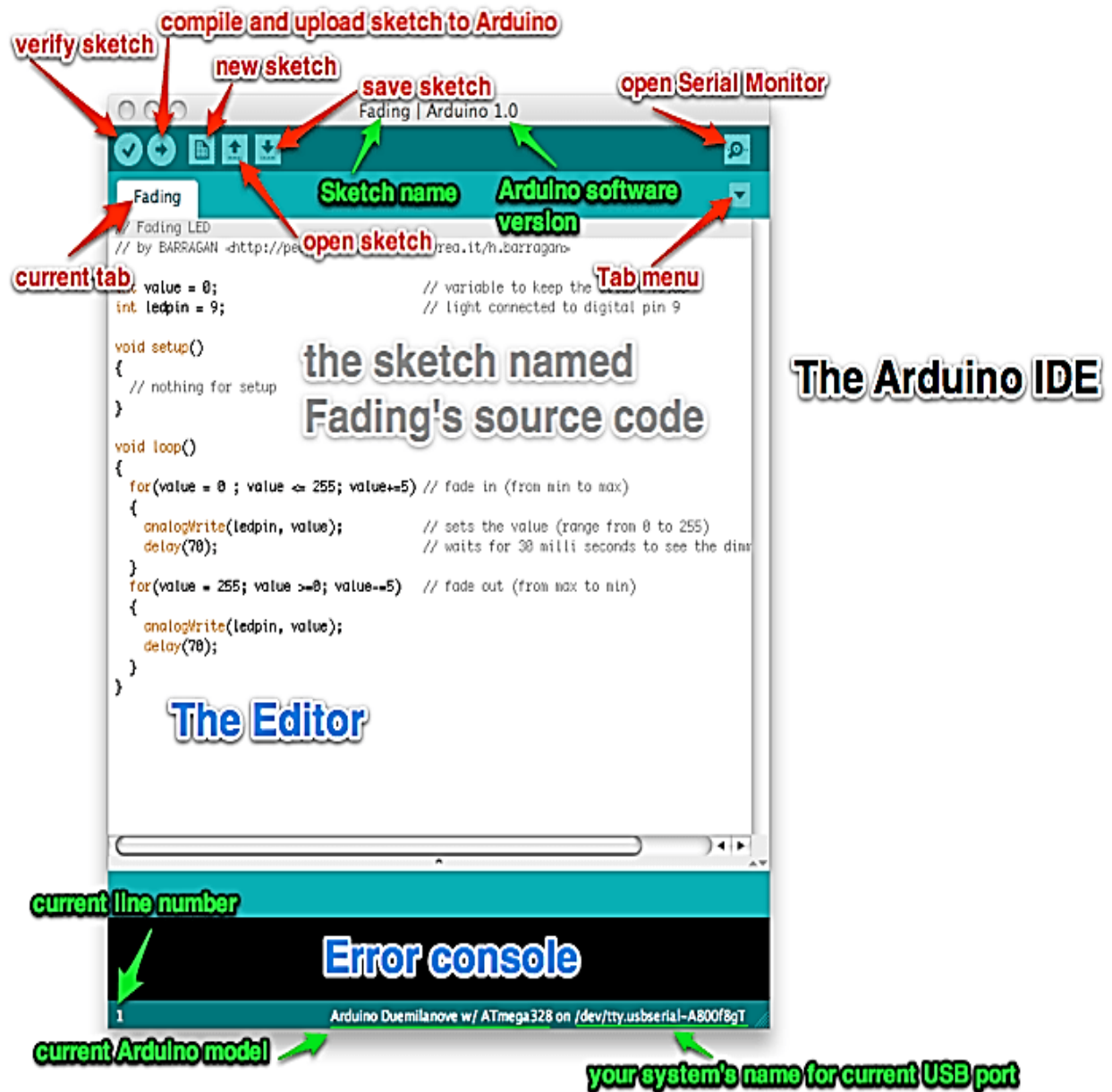


Figure 28: différent bouton et interface du logiciel

II.3 Microcontrôleur ATMEL ATmega328

Dans le premier et le deuxième blocs nous avons utilisé deux cartes Arduino qui possèdent un microcontrôleur ATmega

Les microcontrôleurs de la famille ATMEGA en technologie CMOS sont des modèles à 8 bits AVR basés sur l'architecture RISC. En exécutant des instructions dans un cycle d'horloge simple, l'ATMEGA réalise des opérations s'approchant de 1 MIPS par MHZ permettant de réaliser des systèmes à faible consommation électrique et simple au niveau électronique.

Dans notre cas nous avons utilisé le atméga 328

✓ Microcontrôleur ATMEL ATMega328

Le microcontrôleur de la carte Arduino UNO est un **ATMega328**. C'est un microcontrôleur ATMEL de la famille AVR 8bits. Les principales caractéristiques sont :

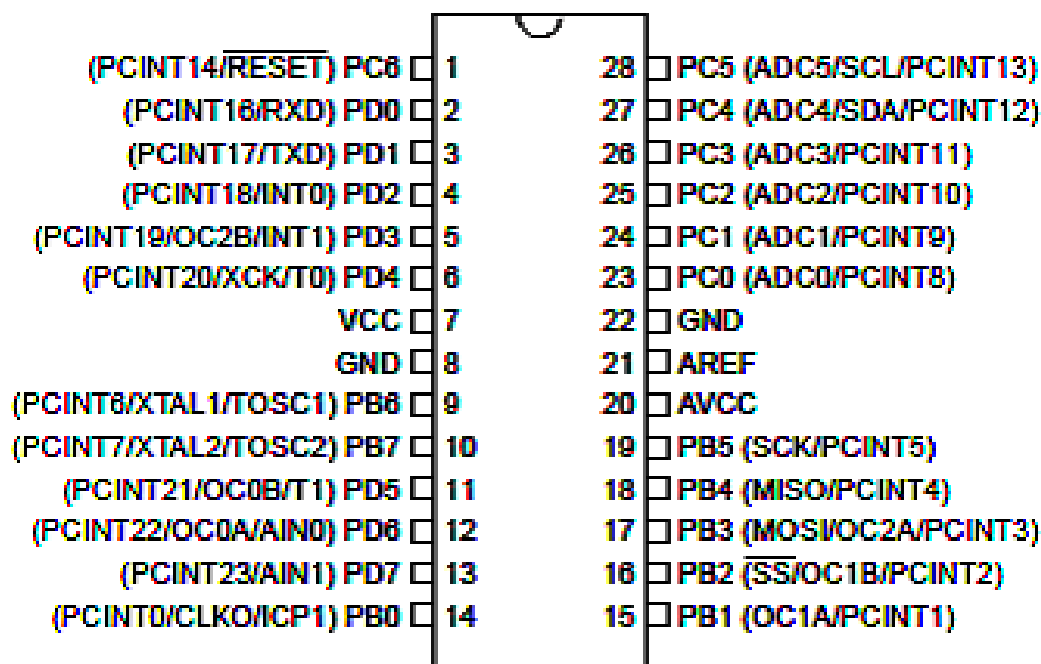


Figure 29 : Microcontrôleur ATMEL ATMega328

- **FLASH** = mémoire programme de 32Ko
- **SRAM** = données (volatiles) 2Ko
- **EEPROM** = données (non volatiles) 1Ko
- **Digital I/O (entrées-sorties Tout Ou Rien)** = 3 ports Port B, Port C, Port D (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) *Microcontrôleurs 4/29*
- **Plusieurs broches multifonctions** :certaines broches peuvent avoir plusieurs fonctions différentes, choisies par programmation. Elles ont alors plusieurs noms sur le brochage (voir ci-avant) Par exemple, les broches PB1, PB2, PB3, PD3, PD5, PD6 peuvent servir de sortie PWM (Pulse Width Modulation), c'est-à-dire des sorties qui joueront le rôle de sorties analogiques. Elles correspondent aux broches des connecteurs 3,5, 6, 9, 10 et 11. Cet autre rôle possible est lié aux timers et ces broches sont alors appelées OCxA ou OCxB dans la documentation. Ce sont les mêmes broches, mais pour une autre fonction. Si vous regardez à nouveau
- **PWM** = 6 broche **OC0A (PD6), OC0B (PD5), OC1A (PB1), OC1B (PB2), OC2A (PB3), OC2B (PD3)** Les broches du PORTC peuvent être converties par un convertisseur Analog to Digital.
- **Analog to Digital Converter**(resolution 10bits) = 6 entrées multiplexer **ADC0(PC0) à ADC5(PC5)**

- **Gestion bus I2C** (TWI TwoWire Interface) = le bus est exploiter 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
- ✓ **WatchdogTimer programmable.**
- **Gestion d'interruptions (24 sources possibles (cfinterruptvectors))** : en résumé
 - Interruptions liées aux entrées **INT0 (PD2)** et **INT1 (PD3)**
 - Interruptions sur changement d'état des broches **PCINT0** a **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)**.

CHAPITRE III

RÉALISATION DE LA MAIN ROBOTIQUE

III.1 montage de la main:

A travers les images qui vont suivre nous allons montrer les étapes de montage de notre main robotique.



Figure 30 : la pomme de la main



Figure 31: les doigts

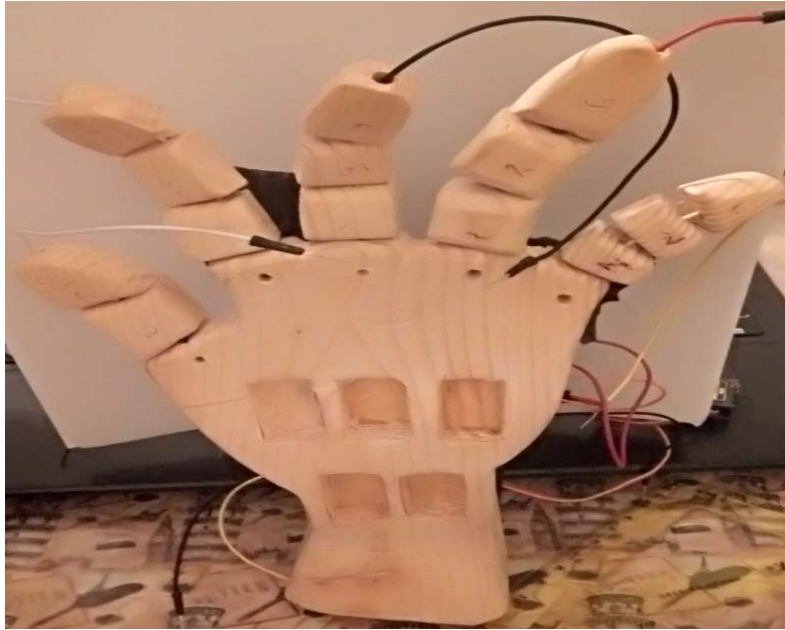


Figure 32: assemblage des doigts et de la pomme de la main

III.2 les éléments nécessaires pour le fonctionnement de la main robotique:

Pour le bon fonctionnement de la main robotique nous avons besoin des éléments suivants :

- bloc d'émission
 - ✓ Une carte arduino nano
 - ✓ 5 bouton poussoir (au lieu de 5 flexsensor que nous avons changé par faute de moyen et la non disponibilité de ces capteurs)
 - ✓ Quelquesfiles électriques
 - ✓ Nrf24 pour l'émission de l'information.
- bloc de réception
 - ✓ Une carte arduinouno
 - ✓ 5 servomoteurs
 - ✓ Nrf24 pour la réception du signal
 - ✓ Quelques files électriques

- ✓ File de pêche .
- ✓ Une pile de 9v

Vu le grand nombre de câbles nous avons utilisé des colliers de serrage pour mieux organiser le câblage

III.3 pilotage d'un servomoteur sur arduino avec un bouton poussoir:

Le pilotage d'un servomoteur avec une carte arduino se fait avec une programmation arduino, il est donc nécessaire d'appeler la bibliothèque<<servo>> et on transmettant l'angle sur le quel nous souhaitons se positionner et on vérifiant la condition mise sur le bouton poussoir.

L'appui sur le bouton poussoir déclenchera le servomoteur, ce qui va nous permettre de réaliser facilement les mouvements souhaités.

Dans la figure qui va suivre nous allons vous montrer le circuit de pilotage d'un servomoteur sur arduino avec un bouton poussoir.

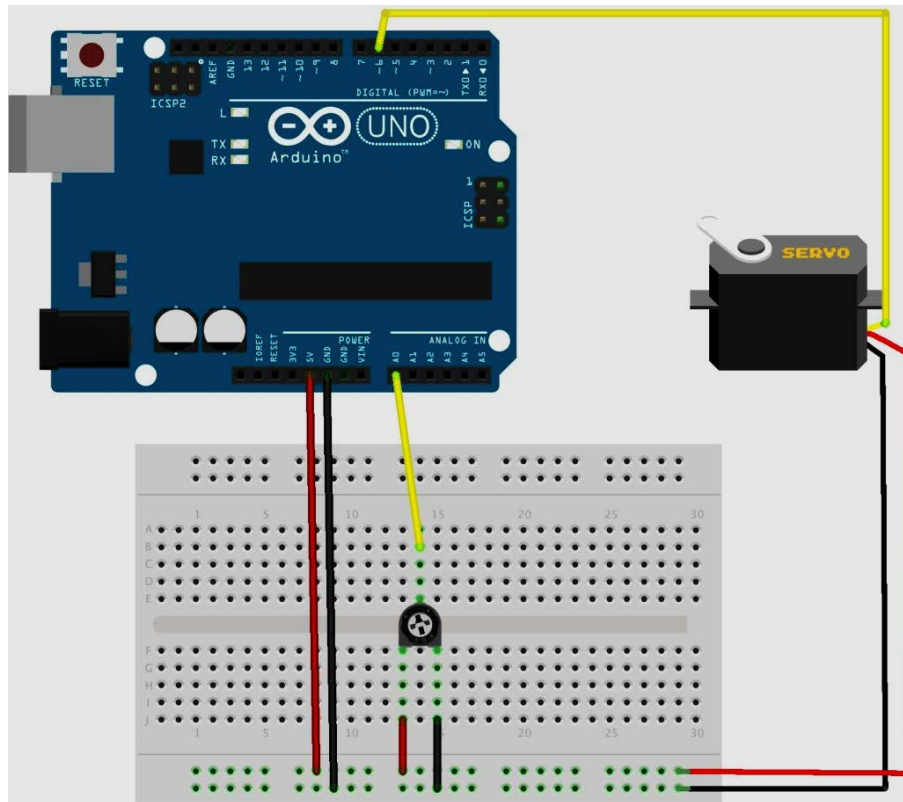


Figure 34: circuit de pilotage d'un servomoteur sur arduino avec un bouton poussoir

III.4 Etape de la réalisation de la main robotique:

Nous avons partagé notre travail en deux parties. Pour réaliser une main nous avons besoin de cinq servomoteurs et de deux NRF24 avec cinq boutons poussoirs et deux plaques d'essais et deux cartes arduino (une nano et une uno). Vu le nombre de servomoteurs nous avons utilisé une alimentation externe pour alimenter les servomoteurs (une pile de 9v)

✓ **partie émettrice**

➤ **Câblage des boutons poussoirs**

- ❖ Chaque bouton poussoir est relié au GND de la carte arduino nano
- ❖ Chaque bouton poussoir sera relié à la carte arduino nano vers les broche digital selon l'ordre suivant:

- ✓ Bouton poussoir 1: broche 2

- ✓ Bouton poussoir 2: broche 3
- ✓ Bouton poussoir 3: broche 4
- ✓ Bouton poussoir 4: broche 5
- ✓ Bouton poussoir 5: broche 6

➤ **Câblage du NRF 24:**

nous avons fait le câblage selon la figure suivante:

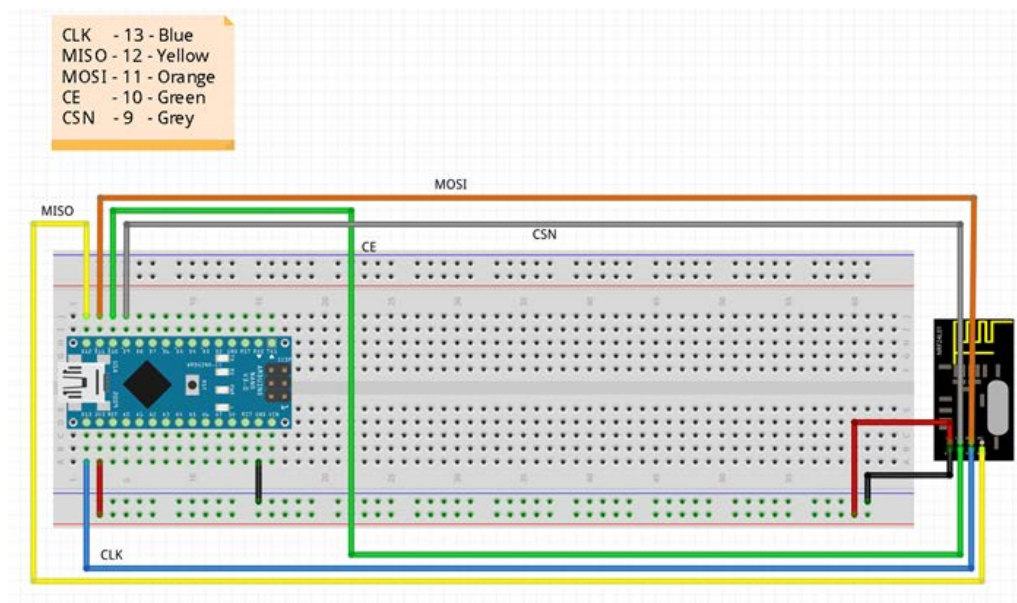


Figure 35: câblage du NRF 24 avec la carte arduino nano

✓ **partie réceptrice**

➤ **Câblage des servomoteurs:**

- ❖ Chaque servomoteur sera relié à l'alimentation externe (fil rouge).
- ❖ Chaque servomoteur sera relié à la carte arduino vers les broches digitales (fil jaune) selon l'ordre suivant:
 - ✓ Servo 1: broche 3
 - ✓ Servo 2: broche 5
 - ✓ Servo 3: broche 6
 - ✓ Servo 4: broche 9

✓ Servo 5: broche 1

❖ Le fil marron du servomoteur sera relié au GND de la carte arduino.

➤ **Câblage Du NRF24:**

Nous avons fait le câblage selon la figure suivante:

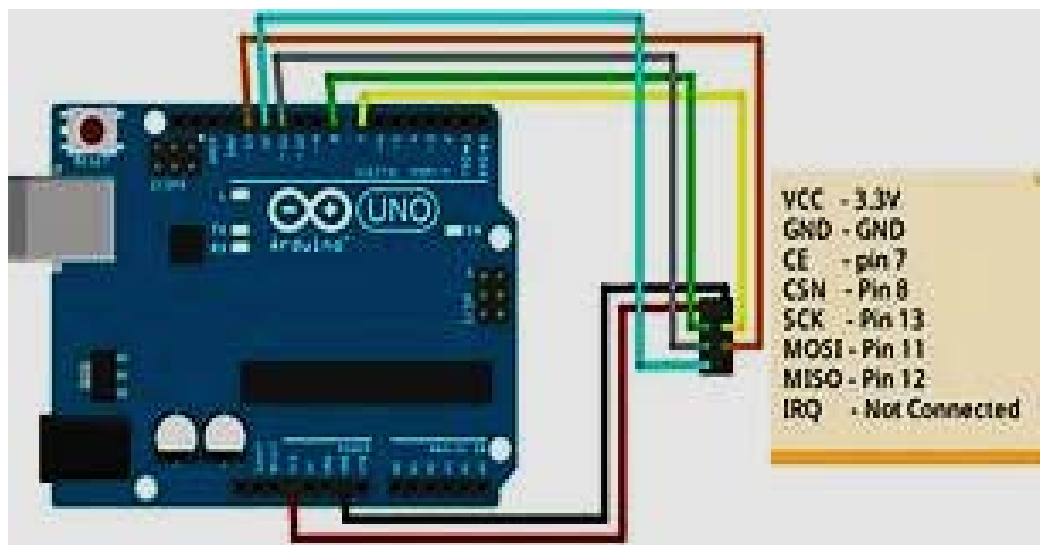


Figure 36: câblage du NRF 24 sur une carte arduinouno.

III.5 Explication du programme:

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 commande⁵.

Le langage de programmation utilisé est le C++, compilé avec `avr-g++`⁶, 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++.

Pour notre projet nous avons créés deux programmes spécifiques que nous avons chargé dans les deux cartes de commande pour que cette dernière puisse positionner les servomoteur.

III.5.1 structure de la programmation:

➤ *Premier partie (émission) :*

❖ *Partie déclaration:*

Au niveau de la partie déclaration nous avons:

- ✓ Inclut la librairie « RF24.h »des fonctionnalités utilisées pour émettre le signal :

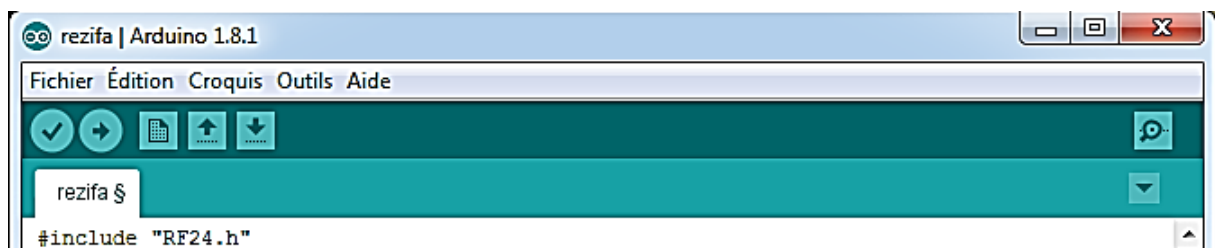


Figure 36:déclaration de la librairie « RF 24 »

- ✓ Déclaration du message a envoyé par la radio fréquence:

```
int msg [1];
```

Figure 37 : déclaration du message du RF24

- ✓ Déclaration des 5 boutons poussoirs en indiquant leur broche :

```
int push1=2;  
int push2=3;  
int push3=4;  
int push4=5;  
int push5=6;
```

Figure 38: déclaration des boutons poussoirs et leur broches

- ✓ Déclaration du statut de chaque bouton poussoir :

```
boolean stat1=0;  
boolean stat2=0;  
boolean stat3=0;  
boolean stat4=0;  
boolean stat5=0;
```

Figure 39: Déclaration du statut de chaque bouton poussoir

- ✓ Déclaration du RF24 et ses broches:

```
RF24 radio (7,8);
```

Figure 40: Déclaration du RF24 et ses broches

- ✓ Déclaration de l'adresse du RF24 :

```
const uint64_t pipe=0xE8E8F0F0E1LL;
```

Figure 41: Déclaration de l'adresse du RF24

- ✓ Déclaration de la forme du message envoyé par le RF24 (forme de caractère):

```
char x[5];
```

Figure 42: Déclaration de la forme du message envoyé par le RF24

❖ Partie de la fonction d'initiation setup():

Nous avons:

- ✓ Déterminer le fonctionnement des broches des boutons poussoirs et activé les résistances pull up internes:

```
pinMode(push1, INPUT_PULLUP) ;  
pinMode(push2, INPUT_PULLUP) ;  
pinMode(push3, INPUT_PULLUP) ;  
pinMode(push4, INPUT_PULLUP) ;  
pinMode(push5, INPUT_PULLUP) ;
```

Figure 43: Déterminer le fonctionnement des broches des boutons poussoirs et activé les résistances pull up internes

- ✓ Initier la bande passante :

```
Serial.begin(9600);
```

Figure 44: initiation de la bande passante

- ✓ Activé le RF24:

```
radio.begin();  
radio.openWritingPipe(pipe);
```

Figure 45:activation du RF24

❖ Partie de la boucle principale fonction loop ():

Nous avons:

- ✓ déclaré les valeurs des boutons poussoirs comme étant des entiers numériques :

```
int value1=! digitalRead(push1);  
int value2=!digitalRead(push2);  
int value3=! digitalRead(push3);  
int value4=!digitalRead(push4);  
int value5=! digitalRead(push5);
```

Figure 46 : déclarations des valeurs des boutons poussoirs

➤ **Deuxième partie (réception):**

❖ **Partie déclaration:**

Nous avons :

- ✓ Inclut la librairie « RF24.h » et la librairie « servo.h »:

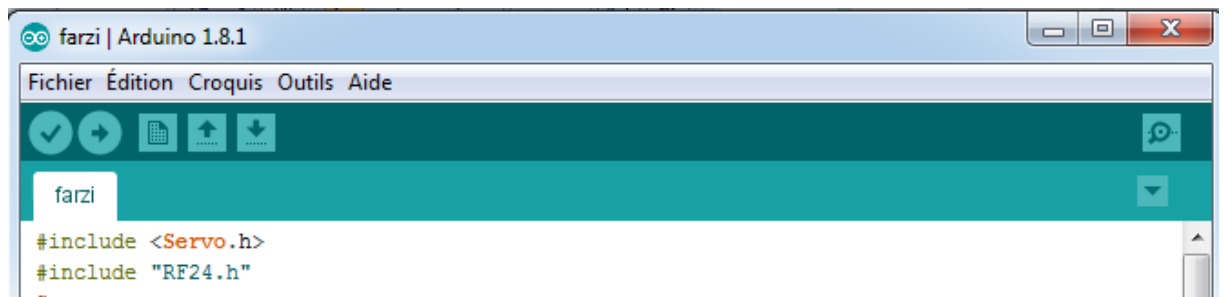


Figure 47 : déclaration de la librairie « RF24.h » et la librairie « servo.h »

- ✓ Déclaration des cinq servomoteurs :



Figure 48: Déclaration des cinq servomoteurs

- ✓ Déclaration du message ainsi que la forme du message qui sera reçu par le RF24:

```
int msg [1];  
char x[5];
```

Figure 49 : Déclaration du message ainsi que la forme du message qu'aura reçu par le RF24

- ✓ Déclaration du RF24 avec son adresse :

```
RF24 radio (7,8);  
const uint64_t pipe = 0xE8E8F0F0E1LL;
```

Figure 50: Déclaration du RF24 avec son adresse

- ✓ Déclaré les valeurs des boutons poussoirs en binaire :

```
boolean stat1;  
boolean stat2;  
boolean stat3;  
boolean stat4;  
boolean stat5;
```

Figure 51: déclaration des valeurs des boutons poussoirs en binaire

❖ Partie de la fonction d'initiation setup():

Nous avons :

- ✓ Attaché les servomoteurs avec les broche de la carte arduino:

```
a.attach(3);  
b.attach(5);  
c.attach(6);  
d.attach(9);  
e.attach(10);
```

Figure 52: attachement des servomoteurs

- ✓ Initiation de la position des servomoteurs :

```
a.write(0) ;  
b.write(0);  
c.write(0);  
d.write(0);  
e.write(0);
```

Figure 53: Initiation de la position des servomoteurs

- ✓ Activé le RF24:

```
radio.begin();  
radio.openReadingPipe(1,pipe);  
radio.startListening();
```

Figure 54: activations du RF24

- ✓ Initier la bande passante :

```
Serial.begin(9600);
```

Figure 55: initiation de la bande passante

❖ Partie de la boucle principal, la fonction loop ():

Nous avons fait:

- ✓ Une Condition du RF24 :

```
if (radio.available()){  
  while (radio.available()) {  
  
    x[0]=' '  
    radio.read(x, sizeof(x));  
    Serial.print(x[0]); Serial.print("\t"); Serial.print(x[1]); Serial.print("\t"); Serial.print(x[2]); Serial.print("\t"); Serial.print(x[3]); Serial.print("\t"); Serial.println(x[4]);  
  }  
}
```

Figure 56: Une Condition du RF24

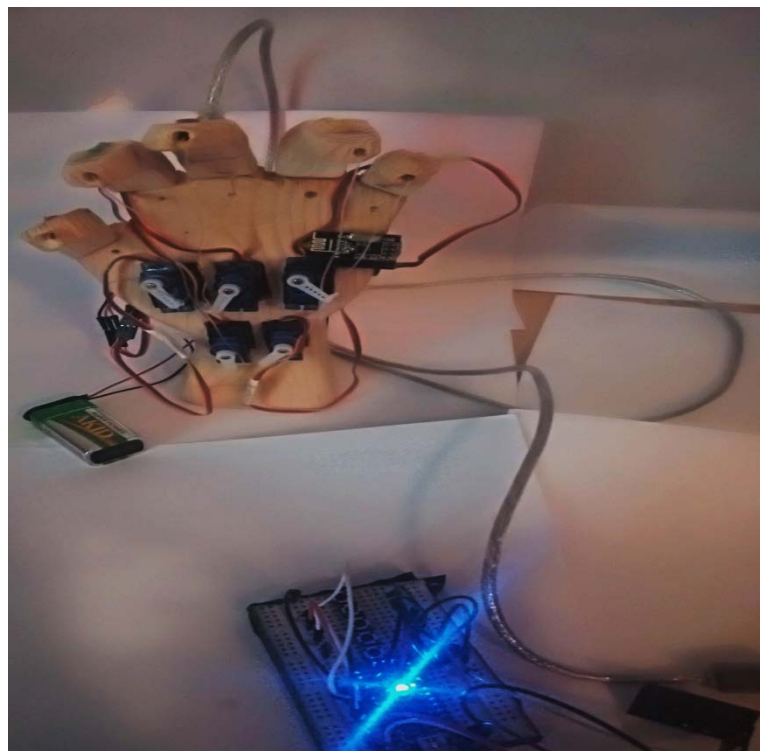
III.6 les actions de tests de la main:***III.6.1 le montage final de la main avec tous les composants utilisés***

Figure 57 : montage final de la main (vue de face)

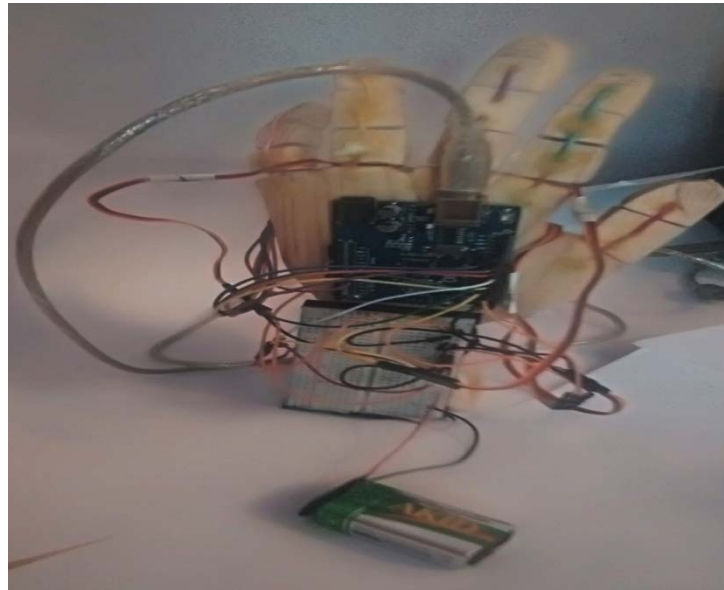


Figure 58: montage final de la main (vue de derrière)

✓ L'articulation du pousse :

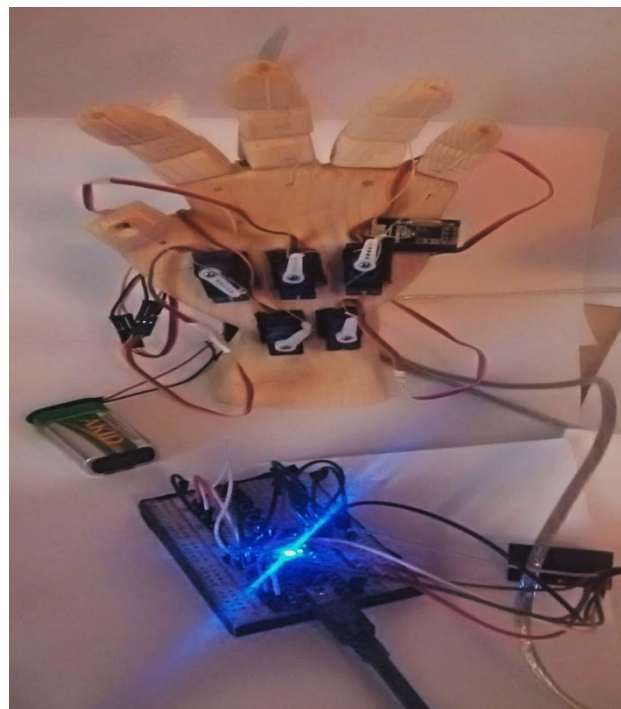


Figure 59: articulation du pousse

- ✓ Articulation de l'index :

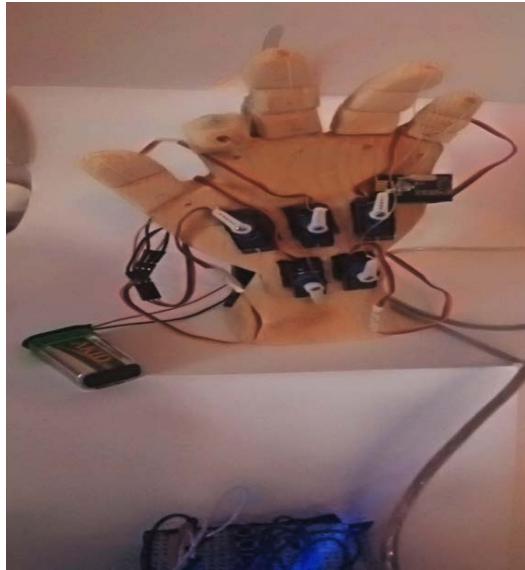


Figure 60: articulation de l'index

- ✓ Articulation du majeur :

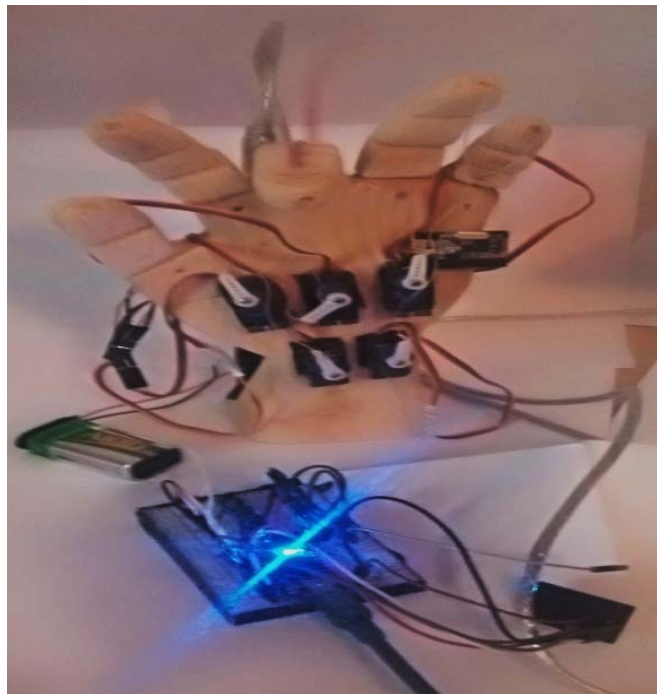


Figure 61 : articulation du majeur

- ✓ Articulation de l'annulaire:

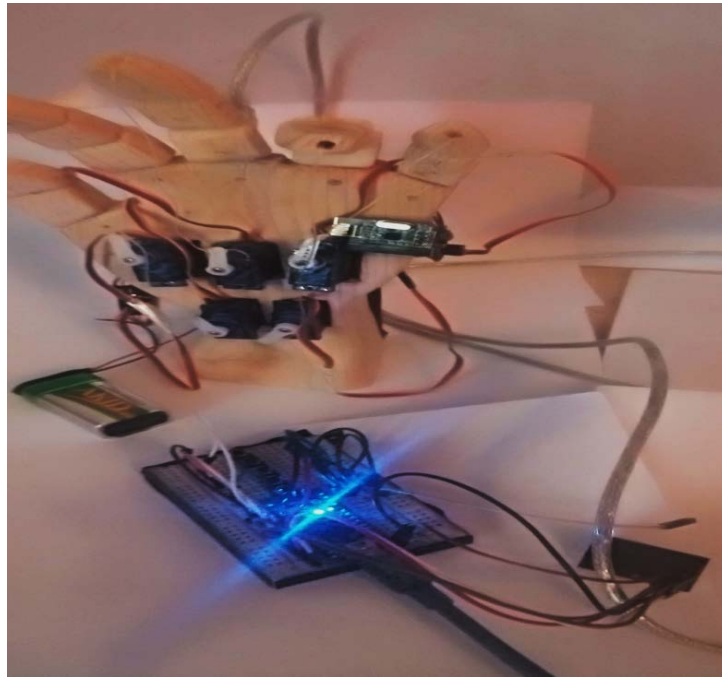


Figure 62 : articulation de l'annulaire

✓ Articulation de l'auriculaire :

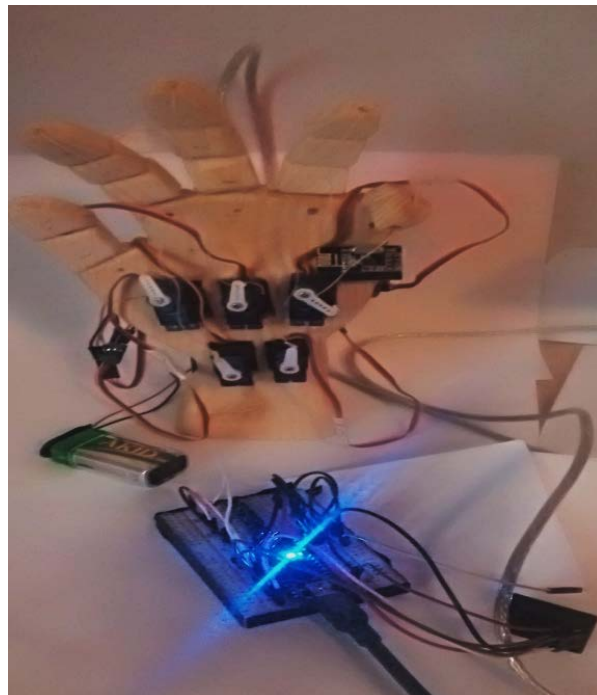


Figure 63 : articulation de l'auriculaire

- ✓ Articulation de l'ensemble des doigts :

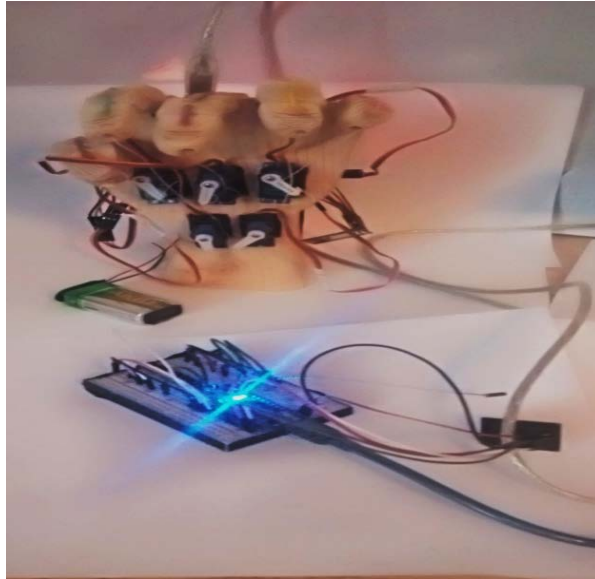
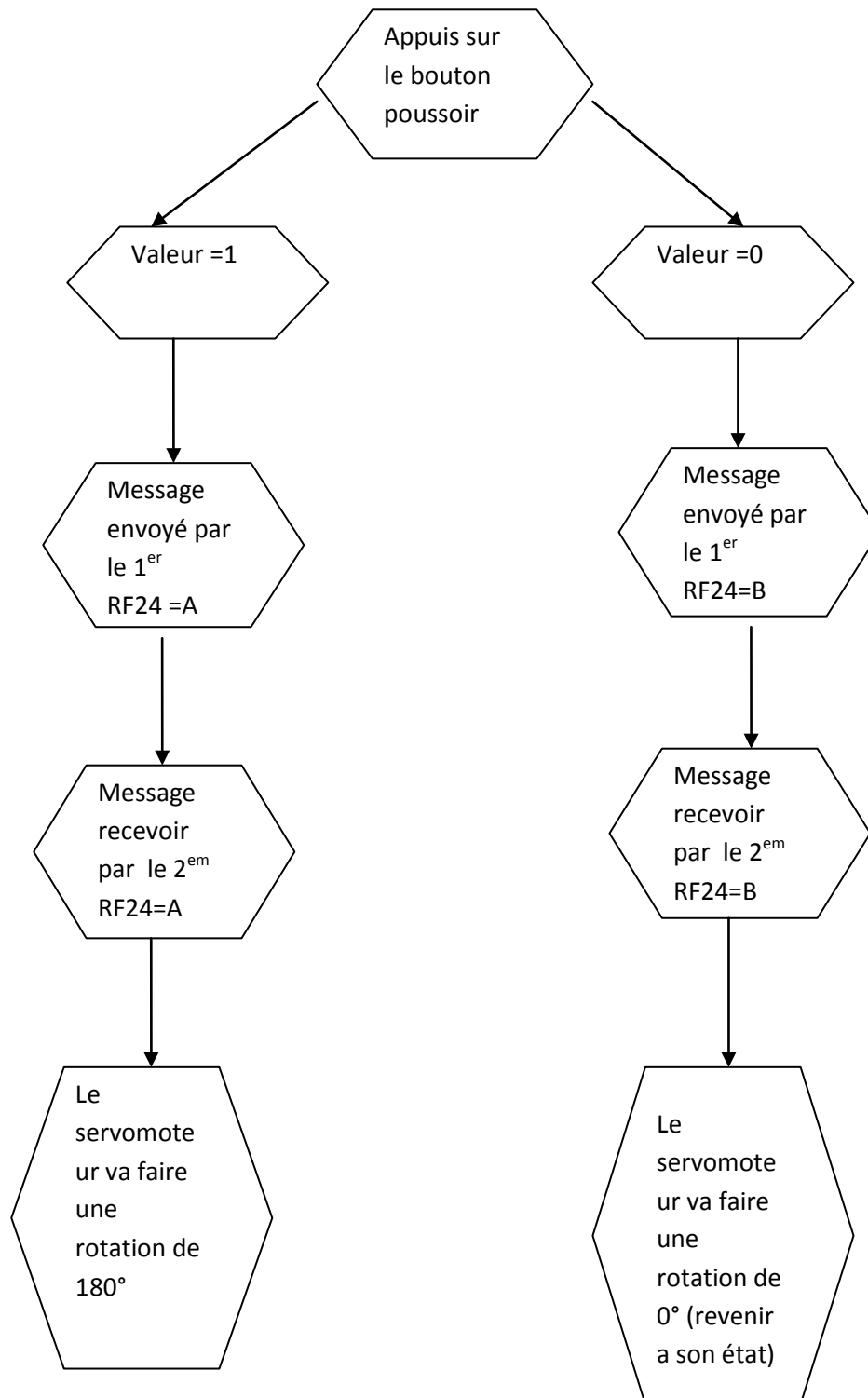


Figure 64: articulation des doigts

III.7 organigramme de fonctionnement de la main par rapport à un bouton poussoir et un servomoteur :



CONCLUSION

Conclusion

Ce mémoire est le résultat d'un travail de recherche dans le domaine de la robotique. Il est consacré à la commande des servomoteurs et des capteur de force par la carte arduino.

Nous avons exposé quelques généralités sur les robots tout en se basant sur les robots manipulateurs et ce pour objectif de réaliser une main robotique et de contrôler les mouvements de la main de manièrea reproduire les gestes d'une main humaine.

Pour aboutir à cela nous avons partagé le travail en deux étapes. La conception de la main, ou nous avons spécifié les différents composants nécessaires pour notre main et étudié leur structure et leur architecture. La réalisation, dans cette partie nous avons étudiéles spécifications de chaque composant utilisé.

Nous avons testé la main et inscrit les résultats au cours de ce mémoire. Comme perspectives futures, nous voulons.

1. Automatiser les mouvements de cette main pour que son fonctionnement soit autonome on rajoutant des capteurs de pression et de présence pour que la main contrôle son articulation.
2. Commandé la main à distance et la faire transporté sur un robot mobile pour un fonctionnement en zones dangereuses.
3. Utilisé des interfaces neurone pour commander la main par des personnes handicapés ou atteint d'un AVC.

BIBLIOGRAPHIE

Bibliographie

Liens internet

<https://www.sparkfun.com>

<http://robotix.ah-oui.org>

<http://lmatigno.perso.info.unicaen.fr>

<http://www.robot24.fr>

<https://fr.wikipedia.org>

<http://physiquemangin.pagesperso-orange.fr>

Livres:

BARTMAN, E.2015, le grand livre d'arduino.2eme édition

EYROLLES.

Les mémoires:

IAMRACHE, M /BENTOUMI,N. étude et réalisation d'un bras robotique .Université Mouloud Mammeri ,2017.

OUKHARFALLAH,K /IHADJADENE,F. Modélisation et conception d'une pince a deux doigts et à deux degrés e liberté d'un bras manipulateur, Université Mouloud Mammeri,2015.