

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE  
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA  
RECHERCHE SCIENTIFIQUE

---



Université de Blida -1-



Institut d'Aéronautique et des Etudes Spatiales  
Département : Navigation Aérienne

Mémoire De Fin De Cycle  
En Vue De L'obtention Du Diplôme  
De Master  
Option : CNS / ATM

**Thème**

**Etude et réalisation d'une interface à  
base d'une carte Arduino pour la  
localisation d'un objet mouvement**

**Présenté par :**  
**ERREF ISMAIL**

**Encadré par :**  
**Mr : BENOURED ABD-ELHALIM**  
**Mr : KOUIDER ELOUAHEB .B**

**Promotion: 2016**

## *Remerciements*

*Nous tenons à remercier en premier lieu et avant tout ALLAH le tout puissant, qui nous a donné la force et la patience d'accomplir notre travail dans les meilleures conditions.*

*Nous remercions nos très chers parents pour leurs soutiens et leurs*

*Patiences.*

*Nous tenons à remercier notre encadreur KOUIDER ELOUAHED.B. Sans oublier BENOURED ABD-ELHALIM Pour leurs précieux conseils et orientations, nous remercions vivement et infiniment toute la promotion M2 CNS/ATM.*

*Nous remercions tous nos enseignants durant toute notre formation ainsi le personnel administratif d'université Blida 1, Saad Dahleb.*

*Nous remercions vivement nos familles Erref et Harir pour leurs soutiens moraux et matériels.*

*Nous ne saurions oublier nos remerciements à tous nos amis SOULI Anouar, DRIFEL Abderrahman, TEMMARI Houcin, DJELAÏLI Abdelbaki, BOUHMAMA Abdeljalil, benchikhoun Arab, MAKHOUKHI Mohamed, BETIOUI Belkhir, MAHBOUB Redouane et à ceux qui ont contribué de près ou de loin à la réalisation de ce travail.*

## *Dédicace*

*Je dédié ce modeste travail à :*

*Mes très chers parents qui m'ont guidé durant les moments les plus pénibles de ce long chemin, ma mère qui a été à mes côtés et m'a soutenu durant toute ma vie, et mon père qui a sacrifié toute sa vie afin de me voir devenir ce que je suis, décédé quand j'ai eu 5 ans, que dieu le pardonne au paradis.*

*Mes frères et mes sœurs.*

*A tous mes amis.*

*Mon ami et le plus proche do ZEGHOUDI Abdelfettah.*

*Tous ceux qui m'aiment et que j'aime.*

*ERRE7 Ismail*

**Résumé ; Abstract**

ملخص

الرادار هو جهاز سلامة لقياس السرعة، المسافة وكذلك موضع شيء معين. عموما يستخدم الرادار موجات الراديو لالتقاط مواضع الأشياء في حين يستخدم الرادار الذي يعتمد على الموجات ما فوق الصوتية هذا النوع من الإشارات الموجية. النتائج المقدمة في هذا الانجاز تبين مدا صحة سير العمل مع بث الاشارات في واجهة بروسيسينغ الموصول بجهاز ارد وينو ومع أجهزة الاستشعار بالموجات فوق الصوتية والمحرك.

الكلمات الجوهرية

رادار. أمواج ما فوق الصوتية. واجهة بروسيسينغ. أردوينو. محرك.

Résumé

Le Radar est un appareil de sécurité qui sert à mesurer la vitesse, la position et la distance d'un obstacle. Un Radar basique utilise des ondes radio tandis que le radar ultrasonore utilise des ondes ultrasonores. Les résultats présentés sur l'interface processing de ce projet valide le bon fonctionnement de l'Arduino avec le capteur ultrasonore et le servomoteur.

Mots clés : Radar, onde ultrasonore, Interface processing, Arduino, servomoteur.

Abstract

Radar is a safety device for measuring the speed, position and distance of an obstacle. A basic radar uses radio waves while the ultrasonic radar uses ultrasonic waves. The results presented in the processing interface of this project validates the proper functioning of the Arduino with the ultrasonic sensor and actuator.

Key words: Radar, ultrasonic waves, processing interface, Arduino, servomotor.

CHAPITRE I GENERALITE SUR LE RADAR.....	10
I-1.Introduction .....	11
I-2. Définition .....	11
I-3. Historique du Radar .....	12
I-4. Principe du radar .....	13
I-5. Caractéristiques des systèmes radars .....	20
I-6. Technique de traitement du signal doppler .....	23
I-7.Applications radar .....	25
I-8. Conclusion.....	26
CHAPITRE II GENERALITE SUR LES MICROCONTROLLEURS .....	27
II-1.Introduction.....	28
II-2.Définition d'un microcontrôleur .....	28
II-3.Les avantages du microcontrôleur .....	28
II-4.Les inconvenient du microcontroller .....	29
II-5.Applications des microcontrollers .....	29
II-6.Architecture d'un microcontrôleur .....	29
II-7. Les familles des microcontrôleurs .....	33
II-8. Microcontrôleur ATMega2560 .....	33
II-9. Présentation générale de l'ARDUINO .....	40
II-10. Critère de choix .....	41
II-11. Carte Arduino Mega .....	43
II-12. Exemple de l'application de la carte .....	48
II-13. Conclusion.....	49
CHAPITRE III REALISATION ET MISE EN MARCHE DE RADAR.....	50
III.Introduction.....	51
III.1 Principe de fonctionnement de Radar ultrasonore .....	51
III-2.Interface de capteur ultrasonique avec le HC-SR04.....	53
III-3.fonctionnement de la communication entre le MVC et l'utilisateur .....	54
III-4.Programmation .....	58

## Sommaire

III-5.Résultat et discussion.....	59
II-6. Conclusion.....	63
Conclusion générale.....	64
Annexe A .....	65
Annexe B .....	71
Annexe C .....	74
Bibliographie.....	83

## Listes des tableaux et figures

Figure I-1 principe de la detection radar.....	14
Figure I-2 Schéma bloc d'un radar monostatique à impulsion.....	15
Figure I-3 Principe de l'émission du radar.....	18
Figure I-4 Génération d'écho.....	19
Figure I-5 Angles caractérisant une cible radar .....	20
Figure I-6 Signaux du radar continu.....	21
Figure I-7 Signaux du radar à impulsions.....	22
Figure I.8 la méthode puls paire.....	24
Figure I.9 diagramme du block doppler.....	24
Figure II.1. Architecture générale d'un microcontrôleur.....	30
Figure II.2 schéma de microcontrôleur ATmega2560.....	34
Figure II.3 les 3 registres pour les configurations du port B.....	36
Figure II.4 registre ELCRA.....	37
Figure II.5 le rapport cyclique de pwm.....	39
Figure II.6 exemple du rapport cyclique.....	40
Figure II.7 carte Arduino.....	41
Figure II.8 Quelques Shields pour Arduino.....	42
Figure II.9 Les éléments de la carte Arduino Mega.....	44
Figure II.10 Schéma de carte Arduino Mega.....	47
Figure II-11 : Carte Arduino avec servomoteur.....	49
Figure III. 1 : Schéma bloc avec circuit réalisé d'un radar ultrasonore.....	52
Figure III.2 Interface graphique de détection de la position de radar.....	53
Figure III.3 le capteur ultrasonore de radar avec illustration des ondes émis et reçus.....	54
Figure III.4 fonctionnement de capteur ultrasonore de HC-SR04.....	55
Figure III.5 Montage électronique de capteur ultrason de distance modèle HC-SR04 avec Microcontrôleur PIC 16F877.....	55
Figure III.6 Photo réelle de notre prototype de radar ultrasonore.....	56

## Listes des tableaux et figures

Figure III.7 Le schéma électronique de montage expérimental avec câblage entre les éléments de système.....	57
Figure III.8 Organigramme pour mesurer la distance entre le capteur et la cible.....	58
Figure III.9 montage expérimental avec la mesure de la distance d'un obstacle.....	60
Figure III.10 montage expérimental pour mesurer la vitesse de la cible en mouvement.....	61
Figure III.11 démonstration de la détection de la position à gauche de capteur.....	62
Tableau I. 1 : Désignation des bandes de fréquences.....	22
Tableau II.1 Le contrôle de sens d'interruptions.....	37
Tableau II.2 Synthèse des caractéristiques de la carte Arduino Mega.....	43

## *Introduction général*

Le radar (de l'anglais RAdio Detection And Ranging) est un système qui utilise généralement les ondes électromagnétiques pour détecter la présence et déterminer la position ainsi que la vitesse d'objets tels que les avions, les bateaux, ou la pluie. Les ondes envoyées par l'émetteur sont réfléchies par la cible, et les signaux de retour (appelés écho radar ou écho-radar) sont captés et analysés par le récepteur, souvent situé au même endroit que l'émetteur. La distance est obtenue grâce au temps aller/retour du signal, la direction grâce à la position angulaire de l'antenne où le signal de retour a été capté et la vitesse avec le décalage de fréquence du signal de retour généré selon l'effet Doppler. Il existe également différentes informations trouvées par le rapport entre les retours captés selon des plans de polarisation orthogonaux.

Le radar est utilisé dans de nombreux contextes : en météorologie pour détecter les orages, pour le contrôle du trafic aérien, pour la surveillance du trafic routier, par les militaires pour détecter les objets volants mais aussi les navires, en astronautique, etc.

Dans notre projet de fin d'études, nous étudierons et réaliserons un prototype d'un Radar à base d'un microcontrôleur Arduino méga avec une sonde ultrasonore et le servomoteur, dans le but de détecter la position de l'objet ainsi que sa vitesse à travers un logiciel qui nous permet de visualiser ces paramètres en temps réel.

Notre mémoire est divisé en trois chapitres :

- Le premier chapitre est une description générale du système Radar.

## Introduction générale

- Dans le deuxième chapitre, nous présentons une généralité sur les microcontrôleurs ainsi que les principales étapes qui le compose .
- Dans le chapitre trois est consacré à La conception et l'étude de l'ensemble des circuits constituant notre Radar sont détaillés dans le troisième chapitre avec la mise en marche de ce Radar.

CHAPITRE I  
GENERALITE SUR  
LE RADAR

## I-1.Introduction

De nos jours le system Radar est devenue un outil nécessaire pour réaliser l'une des fonctions les plus importantes dans le concept CNS/ATM qui est la surveillance. Cette fonction qui est un processus critique nécessite une étude sérieuse en terme de fonctionnement du system Radar et ses différentes parties, et comme ce system est basé sur un system d'émission-réception des signaux donc il nécessite une bonne maitrise des techniques de traitement de signal ce qui est le but de ce mémoire ou on va passer par les différents étages du system Radar à partir de la réception du signal jusqu'à l'affichage des cibles dans le plot radar.

## I-2. Définition

Le radar est un instrument d'alerte (détection) et de mesure (localisation). Dans cette deuxième fonction, deux caractéristiques sont essentielles :

- Précision, incertitude sur la valeur exacte de chaque paramètre de localisation.
- Pouvoir séparateur, ou résolution: possibilité de distinguer et de localiser séparément plusieurs cibles.

Détection et localisation doivent être obtenues dans un cadre espace-temps :

- Le volume surveillé (dans les quatres dimensions).
- La cadence à laquelle cette surveillance doit être renouvelée.

Les premières applications de la radioélectricité furent les télécommunications, puis la radionavigation. Le développement des moyens de transport, tant par mer que par air a fait rapidement apparaître le besoin de détecter simultanément la direction et la distance d'un objet non coopératif.

Le radar a été développé pour « détecter » la présence d'un objet sans la participation de l'objet lui-même. Le mot RADAR qui est universellement adopté pour désigner un matériel répondant à cette exigence est une abréviation de l'expression anglo-saxonne : RADIO DETECTION AND RANGING.

Les radars émettent des ondes radioélectriques, dont la longueur d'onde varie de quelques centimètres à environ 1m. Les objets passant par le faisceau Électromagnétique de l'antenne réfléchissent ces ondes et les renvoient à l'émetteur. [1]

### **I-3. Historique du Radar**

L'histoire du RADAR actuel était initiée au 19<sup>ème</sup> siècle. En 1864 James Clark maxwell décrit les lois de l'électromagnétisme, ce qui permet. En 1888, Heinrich Rudolf hertz montre que les ondes électromagnétiques sont réfléchies par les surfaces métalliques. Au début du XX siècle, le développement de la radio et de la TSF (par Marconi, entre autres) permet de développer les antennes nécessaires à l'utilisation du radar.

Plusieurs inventeurs, scientifiques, et ingénieurs ont contribué ensuite au développement du concept du radar. Les fondements théoriques datent de 1904 avec le dépôt du brevet du « Telemobiloskop » par l'Allemand Christian Hulsmeyer brouillard. Celui-ci a démontré la possibilité de détecter la présence de bateaux dans un espace très obscur.

En envoyant une onde à l'aide d'une antenne multipolaire, son système notait le retour depuis un obstacle avec une antenne dipolaire sans pouvoir cependant en définir plus qu'un azimut approximatif et aucunement sa distance. C'était donc le RAD (radio détection) mais pas l'AR (azimut et rayon). On peut considérer que l'architecture des radars était quasiment finalisée à l'aube de la Seconde Guerre mondiale.

Il manquait cependant l'expérience opérationnelle au combat qui a poussé les ingénieurs à trouver de nombreuses améliorations techniques. Ainsi, les radars aéroportés ont été développés pour donner la possibilité à l'arme aérienne de procéder aux bombardements et à la chasse de nuit. On mena également des expériences sur la polarisation.

Après les années 30, les opérateurs des radars micro-ondes des armées remarquèrent du bruit dans les images, qui s'avérèrent être des échos venant de précipitations (pluie, neige, etc.). Ceci a permis le développement d'une autre fonction du RADAR qui est le radar météorologique

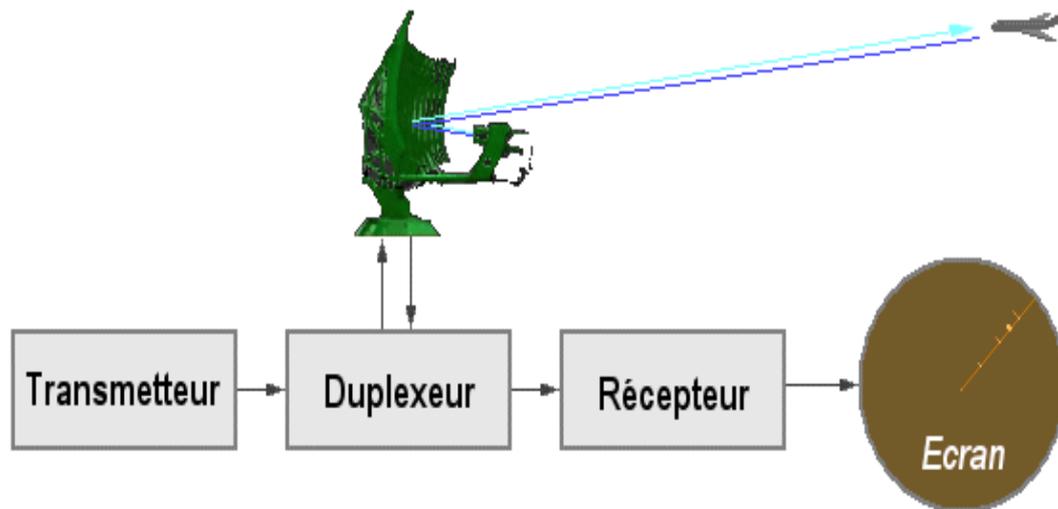
De là, ont été mis au point les premières techniques de brouillage et de contre-mesures électroniques. Après cette guerre, de nombreux domaines ont été développés pour le RADAR, par exemple : l'astrométrie, le contrôle routier et aérien, etc.

Dans les années cinquante, l'invention du radar à synthèse d'ouverture a balisé la voie vers l'obtention d'images RADAR à très haute résolution. Cooley et Tuckey redécouvrent la transformée de Fourier rapide en 1965. Son intérêt a été grandissant avec l'avènement et la progression de l'outil informatique de plus en plus puissant. Cet algorithme mathématique est à la base de la plupart des traitements RADAR numériques utilisés aujourd'hui. [1]

## **I-4. Principe du radar**

### ***I-4.1. Role du radar***

Le radar a pour rôle de détecter, localiser les cibles et d'apporter des informations les concernant. Ceci est fait en illuminant un volume de l'espace avec des ondes électromagnétiques et en recevant l'énergie réfléchiée par les objets présents dans ce volume de l'espace. La figure I.1 présente le principe de la détection radar. [2]



**Figure I-1 principe de la détection radar.**

#### ***I-4.2. Principe de fonctionnement d'un radar***

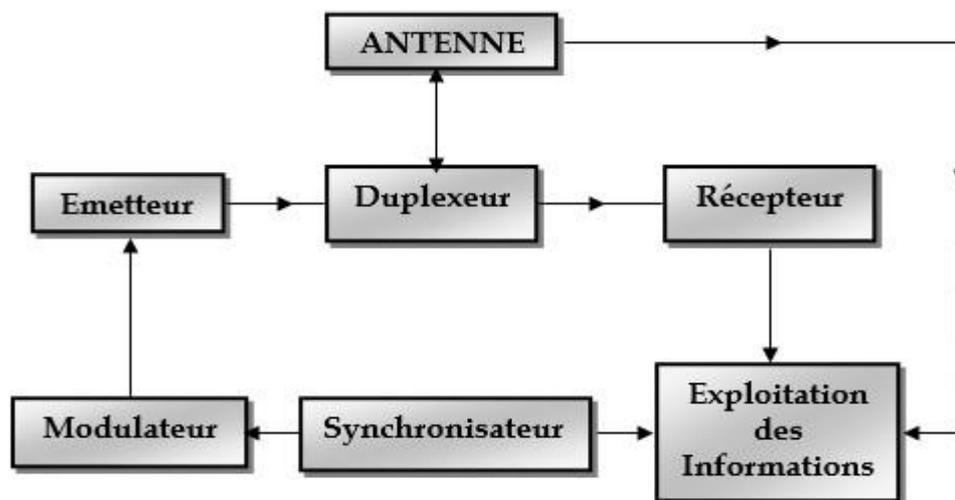
La plus grande partie de l'énergie délivrée par le radar est transmise dans la direction du lobe principal de l'antenne. Cette énergie se propage dans l'air à une vitesse proche de la vitesse de la lumière  $c$ .

Si l'onde électromagnétique rencontre dans sa direction de propagation un objet ayant une impédance caractéristique différente de celle du milieu de propagation, une partie de l'énergie électromagnétique sera diffractée.

La réflexion se fait dans plusieurs directions suivant la forme de l'objet. Le seul écho intercepté par le radar est celui réfléchi dans la direction de son récepteur.

De plus le radar peut recevoir une partie de l'énergie réfléchi par le sol, la mer ou les nuages.

A la réception, l'énergie reçue des différents échos est amplifiée et filtrée. Malheureusement, le récepteur rajoute des bruits aléatoires au signal reçu. Pour extraire le signal utile des interférences, des traitements sont effectués. Le signal résultant est comparé à un certain niveau de référence appelé *le seuil de détection*. Ce seuil doit être soigneusement choisi de manière à avoir un minimum de probabilité de fausse alarme. [3]



**Figure I-2 : Schéma bloc d'un radar monostatique à impulsion**

a) Émetteur

L'émetteur, site du radar, est un circuit oscillateur qui produit des pulsations électromagnétiques à très hautes fréquences. Il comprend un oscillateur permanent, un amplificateur et un modulateur.

Pour les radars à hyperfréquences, qui forment l'immense majorité des radars en service, la génération d'impulsions courtes et très énergétiques demande une technologie qui est différente de celle, disons, d'un émetteur radio utilisé en télécommunications. Sachant qu'une longueur d'onde courte produit un faisceau radar étroit qui distingue et sépare mieux les échos de deux cibles.

On peut également obtenir un faisceau étroit en employant une antenne plus large.

Le modulateur (générateur d'impulsions) constitue la partie active de l'émetteur. C'est un circuit électronique qui permet de fractionner l'onde continue produite par l'oscillateur en impulsions. Il permet de stocker l'énergie pendant les périodes séparant deux émissions successives et de la restituer pendant le temps très bref de l'émission radar. Un radar émet de 500 à 3000 impulsions par seconde et chaque impulsion a une durée  $\tau=0,1$  à  $0,5 \mu s$ .

L'opérateur peut varier le rythme et la durée des impulsions, en fonction de la zone à couvrir.

Des impulsions plus courtes produisent une image plus nette mais demande un rythme d'émission plus rapide. Puisque l'écho ne peut être reçu tant que l'impulsion entière n'est pas émise, la longueur de l'impulsion détermine également la portée minimale.

***b) Duplexeur***

C'est un commutateur électronique qui dirige l'onde vers l'antenne lors de l'émission ou le signal échoïque depuis l'antenne vers le récepteur lors de la réception quand on utilise un radar monostatique. Il permet donc d'utiliser la même antenne pour les deux fonctions, avec une perte minimale. Il est primordial qu'il soit bien synchronisé puisque la puissance du signal émis est de l'ordre du mega-watt ce qui est trop important pour le récepteur qui traite des signaux d'une puissance de l'ordre de quelques nano-watts. Au cas où l'impulsion émise serait dirigée vers le récepteur, celui-ci serait instantanément détruit.

***c) Antenne***

C'est l'élément le plus visible du radar et également le plus connu du profane. Son rôle est de concentrer l'énergie émise par le radar dans un angle solide déterminé. Le meilleur rendement d'une antenne radar est obtenu en la fixant à une hauteur de 3 à 6 mètres au-dessus du niveau de la mer, à l'endroit le plus dégagé possible. Placée plus haut, l'antenne devient sensible aux mouvements de roulis et de tangage. Placée trop bas, ses radiations sont dangereuses pour les personnes qui s'en approchent. L'antenne radar est dessinée de façon à concentrer l'énergie des impulsions émises en un faisceau horizontal étroit. Pour repérer les cibles, l'antenne émet sur le plan horizontal, un rayon très étroit (2 à 4°). Plus l'antenne est large, plus le faisceau est étroit. Sur le plan vertical, le rayon de balayage se situe entre 20 et 40°, et son centre est dirigé vers l'horizon de façon à tenir compte du roulis et du tangage sans perdre les cibles de vue. Les antennes radar doivent avoir une directivité élevée. La directivité d'une antenne caractérise la manière dont celle-ci concentre son rayonnement dans une

certaine direction de l'espace pour envoyer un faisceau étroit étant donné que la largeur du faisceau est proportionnelle à la longueur d'onde du rayonnement et inversement proportionnelle à la largeur de l'antenne.

Si le faisceau est trop grand, il y a gaspillage d'énergie. Par contre, s'il est trop petit, un objet se trouvant sous la portée de l'antenne, ne peut, en plus, être repéré. Par conséquent, en général, aucune antenne n'est parfaite.

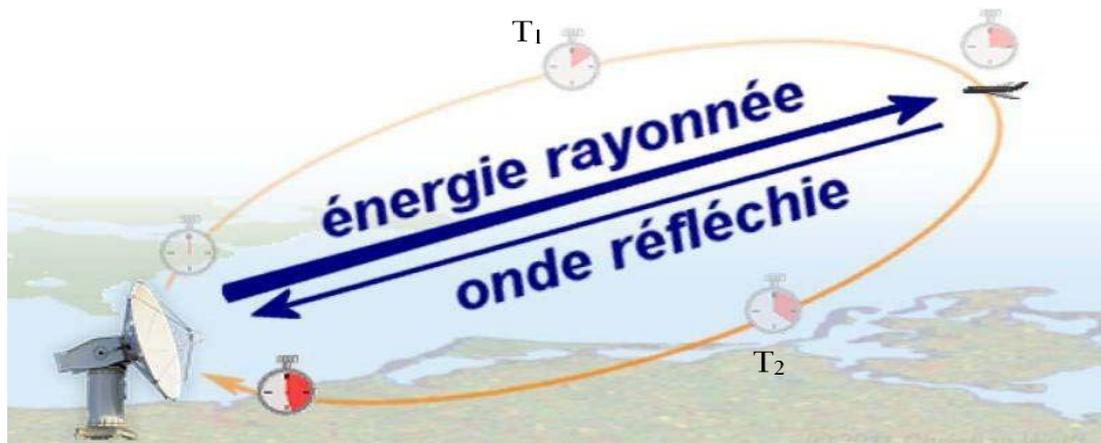
L'antenne peut être double afin de permettre indépendamment les fonctions d'émission et de réception. Dans ce cas, il est nécessaire que les deux aériens élémentaires soient orientés à chaque instant dans la même direction. Ils doivent donc être solidaires entre eux ou synchronisés. En outre, leur interférence radioélectrique doit être la plus faible possible afin qu'au moment de l'émission, le signal émis qui est de très grande puissance ne vienne perturber le fonctionnement du récepteur.

#### **d) Récepteur**

Le récepteur est l'élément le plus délicat et souvent le plus complexe du radar. Il est installé près de l'antenne. Il doit amplifier et mesurer un signal extrêmement faible à une fréquence extrêmement haute. Les amplificateurs mobiles ne pouvant remplir directement cette fonction. En effet, le signal est converti et amplifié par un circuit superhétérodyne à une fréquence intermédiaire d'environ 30 MHz. La fréquence très élevée du signal radar nécessite l'utilisation d'un oscillateur et d'un mélangeur de signaux avec une précision beaucoup plus élevée que pour les récepteurs radio ordinaires. Des circuits appropriés ont été mis au point, utilisant comme oscillateurs des tubes hauts puissances à hyperfréquences, autrement dit, les klystrons. La fréquence intermédiaire est amplifiée de manière classique. Le signal est ensuite envoyé dans un ordinateur.

Un système radar est composé généralement d'un ensemble de sous-systèmes, représentant ainsi ses constituants majeurs, dans le but de contrôler ses capacités opérationnelles. Les principaux sous-systèmes radars sont, l'antenne, l'émetteur, le récepteur et le traitement de signal. Le récepteur prépare les échos

radar pour le traitement du signal par amplification et conversion de fréquence, l'extraction des informations de cible est effectuée par le bloc de traitement du signal. Ce dernier peut être intégré avec le récepteur, mais il est usuellement considéré comme un sous-système séparé et spécialement lorsqu'il est implanté d'une façon numérique.



**Figure I-3 : Principe de l'émission du radar**

#### ***I-4.3. Mesure de la distance***

La mesure de la distance à un objet est faite d'une façon à émettre une courte impulsion de signal radio, et de mesurer le temps d'aller-retour de l'onde émise. La distance est la moitié du temps de retour de l'onde (car le signal doit aller à la cible puis revenir) multipliée par la vitesse du signal (qui est proche de la vitesse de la lumière dans le vide si le milieu traversé est l'atmosphère). [4]

$$D = \frac{C \cdot \Delta T}{2} \quad (I-1)$$

D: la distance antenne- cible[m].

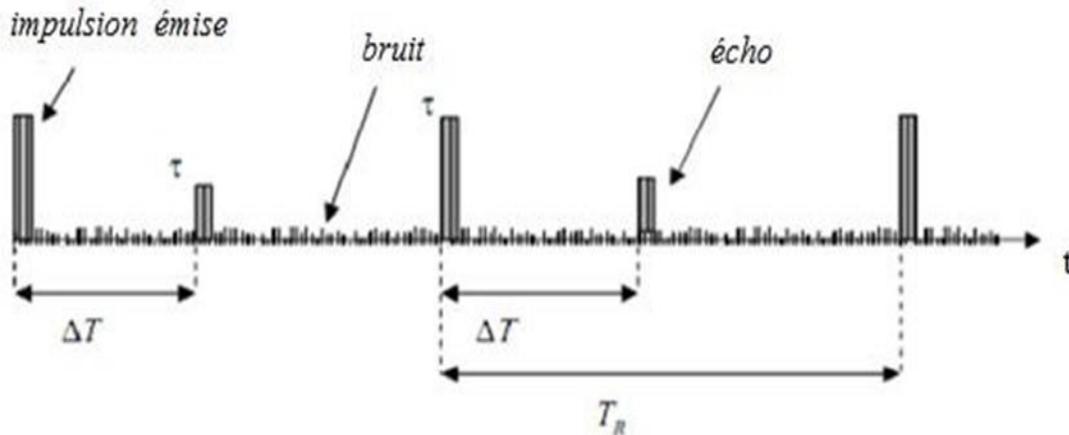
$\Delta T$  : le temps ( $\Delta T = T_2 - T_1$ ).

C: la vitesse de lumière ( $C = 3 * 10^8$ m/s).

Le signal reçu aura la même forme que le signal émis mais il sera très faible et toujours accompagné d'un bruit de fond provenant :

- Soit du bruit atmosphérique, qu'on ne peut réduire à zéro.

- Soit du bruit propre du récepteur, qu'on ne peut réduire à zéro.
- Soit même d'un brouillage du par exemple a un ennemi non coopérant (ou à un ami maladroit).



**Figure I-4 : Génération d'écho.**

$T_r$ : Période de répétition des impulsions.

$\tau$  : Durée de l'impulsion.

$\Delta T$ : Temps correspondant à un aller-retour de l'onde entre le radar et la cible.

#### **I-4.4. Mesure de la direction**

La façon qui permet de connaître la direction d'une cible est basée sur un calcul d'angle entre la direction du nord et celle de la cible (azimut). La directivité (gain directif) est la capacité de l'antenne à concentrer l'énergie rayonnée dans une direction particulière. Une antenne à forte directivité est appelée "antenne directive". En déterminant la direction dans laquelle est pointée l'antenne à l'instant où elle reçoit un écho, on peut déterminer non seulement l'azimut mais aussi le site de la cible (donc son altitude). La précision de la mesure de ces angles dépend de la directivité de l'antenne. Pour une fréquence émise donnée (ou une longueur d'onde définie), la directivité de l'antenne est fonction de ses dimensions propres. Les radars émettent normalement de très hautes fréquences pour les raisons suivantes :

- Propagation quasi rectiligne de ces ondes.

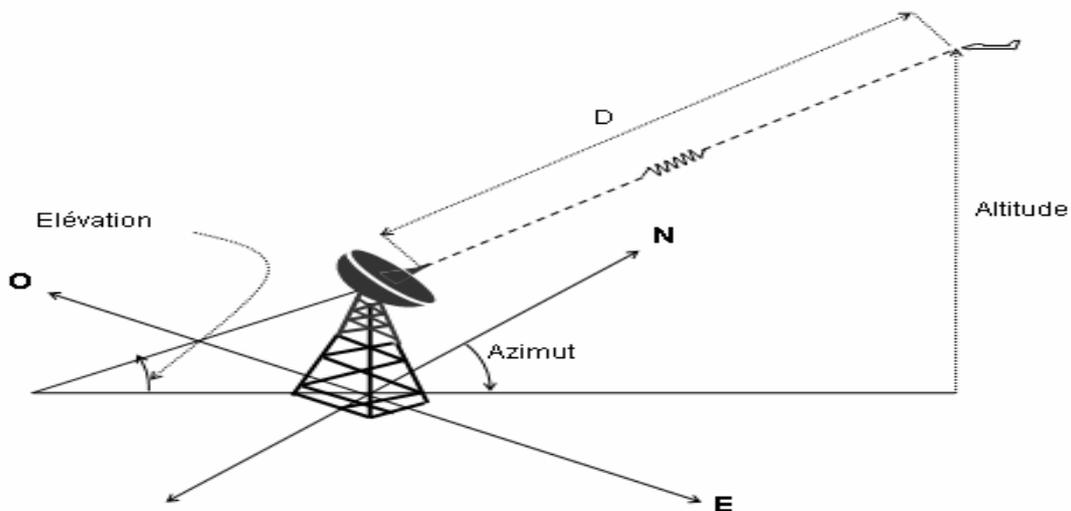
- Haute résolution (plus la longueur d'onde est courte, plus le radar est capable de détecter un petit objet).
- Encombrement réduit de l'antenne (plus on augmente la fréquence du signal rayonné plus la directivité est grande pour une antenne de taille donnée).

#### ***I-4.5. Mesure de la hauteur***

La hauteur d'une cible par rapport à la surface du globe terrestre est appelée 'Altitude'. Elle peut être calculée en multipliant la distance oblique  $D$  de la cible par le sinus de son angle d'élévation.

#### ***I-4.6. Mesure de vitesse***

Il existe différentes méthodes pour mesurer la vitesse de déplacement d'une cible. La plus ancienne consiste à noter sa position à un instant, à l'aide d'un crayon gras, sur l'affichage radar. À un autre instant, on refait la même chose et la différence des deux positions divisée par le temps écoulé entre les deux instants, donne la vitesse de déplacement [5].



**Figure I-5 :Angles caractérisant une cible radar**

### **I-5. Caractéristiques des systèmes radars . [5]**

#### ***I-5.1. Types de radars***

Les radars peuvent être classés suivant plusieurs caractéristiques (mission du radar, relation entre l'antenne émettrice et l'antenne réceptrice, la bande de

fréquences utilisée,...etc.). Cependant, la classification la plus commune est basée sur le mode de transmission du signal.

Deux grandes familles sont distinguées : le radar à impulsion et le radar à onde continue.

**Le Radar à impulsions** illumine la cible en émettant de courtes impulsions suivant une fréquence de répétition (PRF).

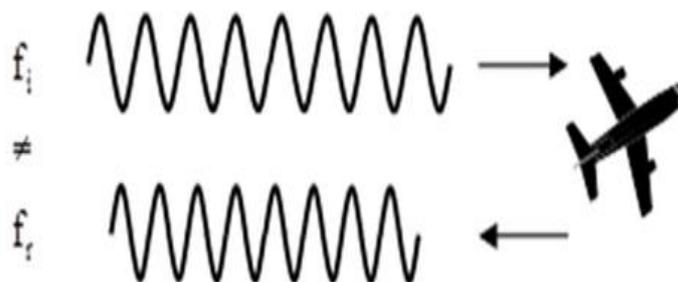
**Le Radar à ondes continues** émet un signal de façon continue.

### *I-5.2. Modes d'émission Radar*

Il existe deux principaux modes d'émission :

a) **Mode en ondes continue** : la forme la plus simple d'un signal à onde continue est la forme sinusoïdale. Il est principalement utilisé pour la mesure de la vitesse et la poursuite des cibles en mouvement. On peut distinguer deux sous modes en ondes continues : Mode en ondes continues modulé et non modulé.

La différence principale entre ces deux derniers est que le premier ne peut pas mesurer la distance de la cible tandis que le dernier le peut en introduisant une modulation de la fréquence dans le temps.



**Figure I-6 : Signaux du radar continu.**

b) **Mode à impulsion** : il travaille sur le principe de l'émission de courtes impulsions d'énergie.

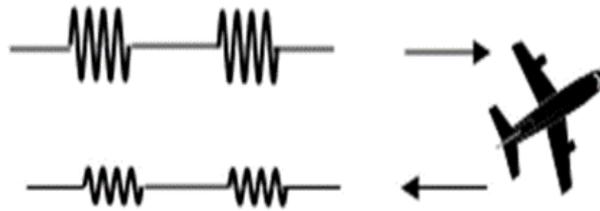


Figure I-7 :Signaux du radar à impulsions

### I-5.3. Les fréquences radar

La désignation de la bande de fréquences radar est spécifiée comme suit :

Tableau I-1 : Désignation des bandes de fréquences

Désignation de la bande	Plage des fréquences
<b>HF</b>	3-30MHz
<b>VHF</b>	30-300MHz
<b>UHF</b>	300MHz-1GHz
<b>L</b>	1-2GHz
<b>S</b>	2-4GHz
<b>C</b>	4-8GHz
<b>X</b>	8-12GHz
<b>Ku</b>	12-18GHz

Le tableau I.1, illustre l'allocation de chaque bande de fréquence avec leurs symboles correspondants.

## I-6. Technique de traitement du signal doppler

- ❖ Pulse paire
- ❖ Analyse spectral en utilisant la transformé de Fourier

### I-6.1. Pulse Pair

La vitesse de la cible peut être estimée par l'utilisation de la technique "pulse-pair". Le pulse –Pair fournit une estimation de la vitesse de la cible en déterminant la phase moyenne de changement de la cible qui a eu lieu pour cette dernière, du PRF au PRF, une moyenne sur plusieurs PRF.

Comme beaucoup PRF qui peuvent être obtenue entre 1° des limites de l'angle d'azimut sont utilisés. Plus PRF équivaut à moins de bruit, donnant des estimations plus précises. L'algorithme Pulse-Pair a été traditionnellement utilisé comme la puissance CPU requis ne soit pas excessive.

Les échantillons de données I et Q sont considérés pour décrire un vecteur complexe qui va tourner à une vitesse directement liée à la vitesse de la cible, c-à-dire l'échantillon  $[i] = I + jQ$ . Un algorithme d'autocorrélation est appliqué à chaque intervalle de distance individuelle, à travers plusieurs PRF.

### I-6.2. Calcul de la vitesse doppler avec cette méthode

- On prend deux pulsions consécutives et on mesure la phase de la pulsion reçu (Figure I-8)

- On sait que  $\frac{d\phi}{dt} = \frac{4\pi}{\lambda} \frac{dr}{dt}$  (1.1)

Ou  $df = f_2 - f_1$

$dt$  = période entre deux impulsions consécutives

$d_r/d_t$  la vitesse radial de la cible

- Donc de l'équation (1.2)

$$v = \frac{\lambda}{4\pi} \frac{d\phi}{dt} \quad (1.3)$$

- Et en réalité  $d_t/d_t$  c la vitesse angulaire

$$w = 2\pi f_d \quad (1.4)$$

• Donc l'équation (2.8) devient :

$$v = \frac{\lambda f_d}{2} \tag{1.5}$$

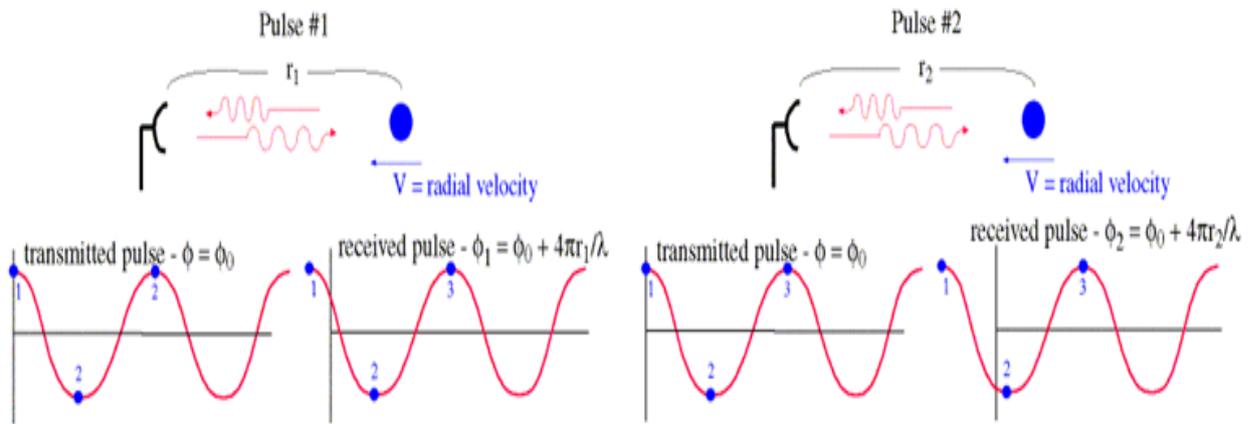


Figure I-8 : la méthode puls paire

Calcul de la fréquence doppler avec cette méthode :

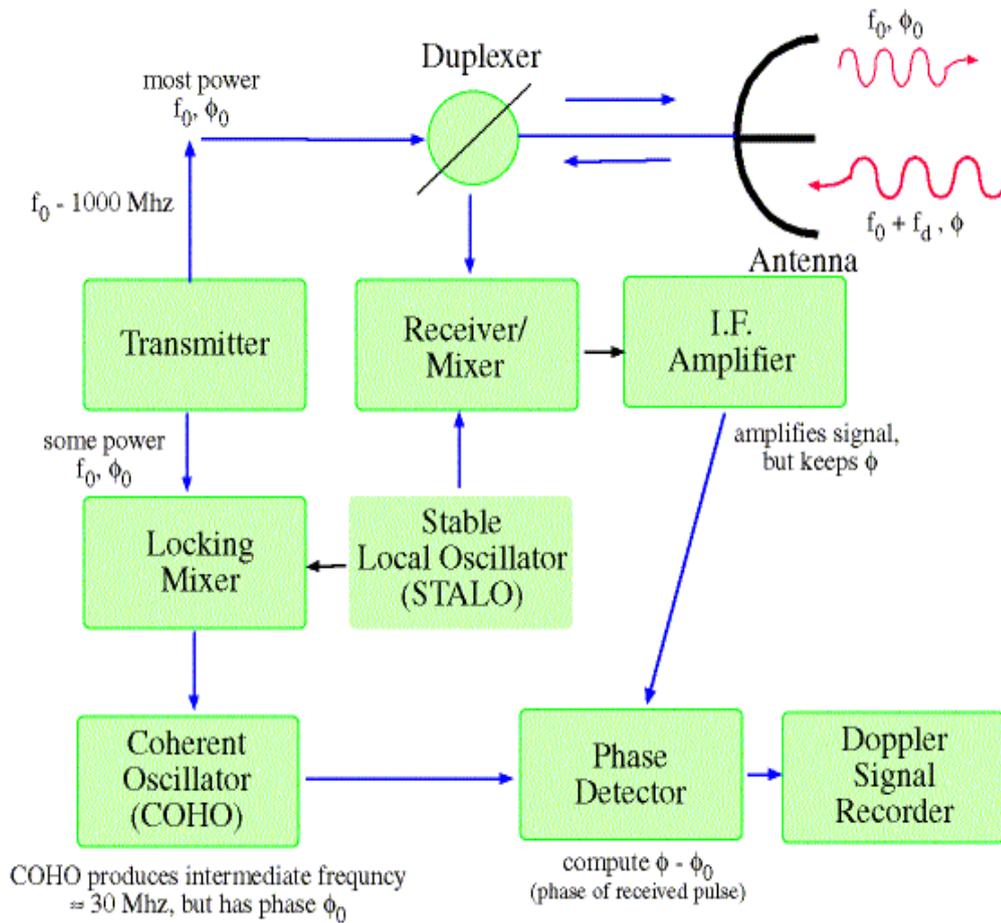


Figure I-9: diagrammed du block Doppler

- Le transmetteur produit une pulsion de fréquence  $f_0$  et de durée  $t$ .
- Une partie de la puissance avec  $f_0$  est mixé avec un signal provenant du block STALO (stable local oscillatoire) puis il est envoyé vers le block COHO (cohérent oscillatoire).
- COHO maintient  $f_0$  de l'onde transmise.
- Le block receveur/mixer mixe le signal provenant de STALO et le signal reçu
- Le signal mixé est ensuite amplifié.
- La phase du signal original et du signal reçu sont différentes c.-à-d. on calcul  $f_1 = f_0 - f$ , c'est la phase de la première impulsion.
- La répétition de ces étapes pour différentes impulsions successives va nous donner  $\frac{df}{dt}$  [6]

### I-7.Applications radar

En dehors de la navigation en mer et dans les airs, le radar a trouvé un usage quasi universel en météorologie, non seulement pour la prévision et la localisation de perturbations, telles que les ouragans et les tornades, mais également pour connaître les conditions météorologiques locales. Les radars fournissent aussi des informations sur les précipitations, permettant de signaler à l'avance les crues subites. D'autres équipements radar sont utilisés par la police pour surveiller la vitesse des véhicules.

L'une des principales utilisations du radar reste le contrôle du trafic aérien : guidage des avions vers les pistes d'atterrissage et suivi des avions en vol.

Les progrès récents, notamment les techniques rendant plus net le contraste entre les signaux radar utiles et le bruit radio aléatoire, ont encore contribué à élargir le domaine d'utilisation du radar. Son application s'est étendue à l'exploration de l'espace, à l'observation des missiles et des satellites artificiels à très haute altitude.

Le radar est un élément essentiel des systèmes de défense, pour la détection des engins balistiques intercontinentaux. Cependant, avec les récentes générations de radars, apparaissent de nouvelles parades, en particulier grâce au développement de systèmes favorisant la « furtivité » des avions de combat et des missiles.

Le développement rapide du radar qualitativement et quantitativement démontre le grand intérêt donné à ce système. Parmi ses autres applications on peut citer :

- Le contrôle du trafic aérien.
- La navigation aérienne.
- La navigation maritime.
- Les systèmes d'alerte de défense.
- Les mesures de portée et de vitesse pour les besoins de contrôle de vitesse par la police.
- L'assistance pour atterrissage des véhicules sur la lune.
- Surveillance des frontières. [7]

## **I-8. Conclusion**

Dans ce chapitre, nous avons présenté les différents paramètres de base d'un système Radar, les principaux constituants d'un radar qu'ils sont : émetteur, récepteur, antenne, détecteur et moyen de visualisation. L'équation radar permet de calculer la portée d'un radar connaissant les caractéristiques techniques de ce dernier. L'effet Doppler permet d'estimer la vitesse radiale d'une cible et de distinguer entre la cible mobile et la cible fixe. Une limitation convenable de la portée d'un radar évite l'ambiguïté dans l'estimation de la distance des cibles. Les modèles de Swerling sont des modélisations des cibles fluctuantes ; ils sont utilisés dans l'estimation des performances des détecteurs radar.

CHAPITRE II  
GENERALITE SUR LES  
MICROCONTROLLEURS

## II-1.Introduction

Afin d'étudier et présenter les principaux constituants d'un radar, il est indispensable de savoir l'élément principal qui nous permet de contrôler tous le système Radar et de traiter les signaux d'émission et de réception, dans le but d'expliquer : l'architecteur, la famille et l'application de microcontrôleur. La deuxième partie de ce chapitre présente les détails de la carte Arduino pour finir avec le montage expérimental.

## II-2.Définition d'un microcontrôleur

Un microcontrôleur (en notation abrégée  $\mu c$ , ou uc ou encore MCU en anglais) est un circuit intégré qui rassemble les éléments essentiels d'un ordinateur : processeur, mémoires (mémoire morte pour le programme, mémoire vive pour les données), unités périphériques et interfaces d'entrées-sorties. Les microcontrôleurs se caractérisent par un plus haut degré d'intégration, une plus faible consommation électrique, une vitesse de fonctionnement plus faible (de quelques mégahertz jusqu'à plus d'un gigahertz) et un coût réduit par rapport aux microprocesseurs polyvalents utilisés dans les ordinateurs personnels.

Par rapport à des systèmes électroniques à base de microprocesseurs et autres composants séparés, les microcontrôleurs permettent de diminuer la taille, la consommation électrique et le coût des produits. Ils ont ainsi permis de démocratiser l'utilisation de l'informatique dans un grand nombre de produits et de procédés. [8]

## II-3.Les avantages du microcontrôleur

- Encombrement de matériel réduit,
- Circuits imprimé peu complexe,
- Faible consommation à cause de l'intégration en technologie MOS, CMOS, ou HCMOS,
- Coût réduit de conception et montage

- Environnement de programmation et de simulation évolués. [9]

#### **II-4. Les inconvénients du microcontrôleur**

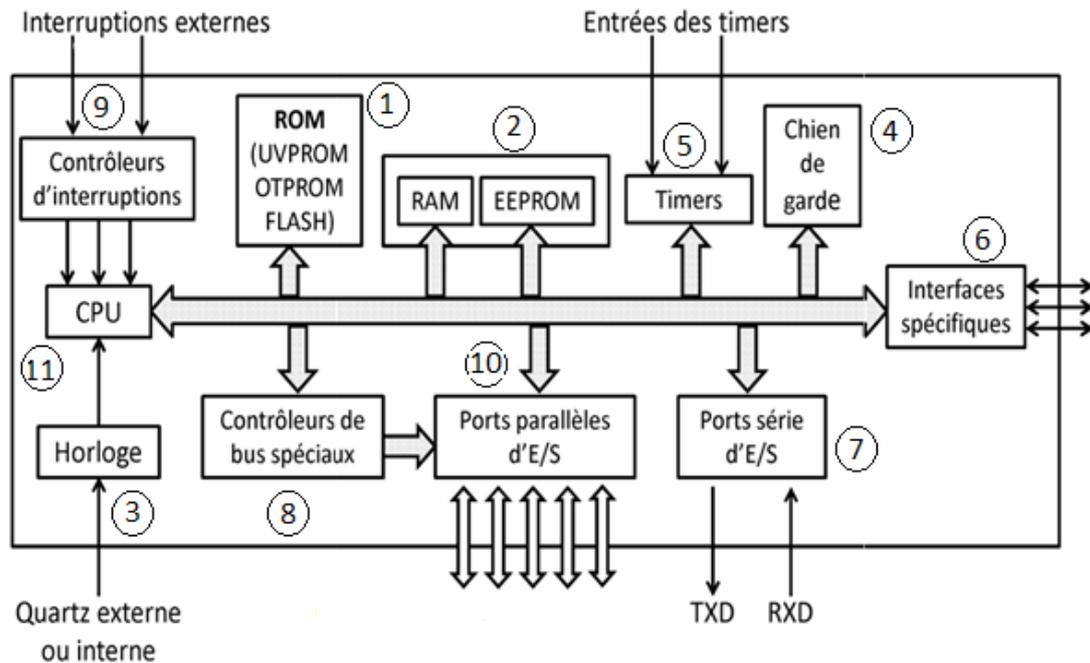
- Ne convient pas nécessairement à tous les problèmes car il n'offre pas une grande puissance de calcul.
- On ne peut pas toujours utiliser tous les périphériques simultanément : pour réduire les coûts, certaines broches sont multiplexées.
- Il faut disposer d'un outil du développement spécifique, compilateur, ou simulateur. [9]

#### **II-5. Applications des microcontrôleurs**

- Informatique (souris, modem,...),
- Vidéo (appareil photo numérique, caméra numérique, ...),
- Contrôle des processus industriels (régulation, pilotage, supervision, ...),
- Appareil de mesure (affichage, calcul statistique, mémorisation, ...),
- Automobile (ABS, injection, GPS, airbag, ...),
- Multimédia (téléviseur, carte audio, carte vidéo, MP3, magnétoscope, ...)
- Téléphone (fax, téléphone portable, modem, ...),
- Electroménager (Lave-vaisselle, lave-linge, four micro-onde, ...),  
[10]

#### **II-6. Architecture d'un microcontrôleur [11]**

On peut envisager le schéma bloc général d'un microcontrôleur de la figure II.1, en sachant que certaines fonctions peuvent être absentes d'un type à l'autre.



**Figure II-1: Architecture générale d'un microcontrôleur.**

- 1) La mémoire programme 'ROM' est de plus en plus souvent reprogrammable par différentes techniques : ultra-violet (UV PROM), une seule fois (OTP), par block mémoire (FLASH) et même si le  $\mu$ contrôleur est soudé sur le circuit imprimé en utilise la fonctionnalité de la programmation in-situ (In-System Programmions ou ISP). Des lignes d'E/S sont spécifiquement dédiées à cette fonction.
- 2) Les mémoires de données 'RAM et EEPROM' permettent de mémoriser temporairement les données générées par le microprocesseur pendant les différentes phases du traitement numérique (résultats des opérations, états de capteurs, ... etc.).
- 3) L'horloge est souvent un quartz entouré de deux condensateurs, mais il peut aussi être réalisé en interne par un circuit RC que l'on peut calibrer lors de la première utilisation du microcontrôleur.
- 4) Le chien de garde 'watchdog' est un temporisateur particulier qui génère, dans un délai préfixé ou programmé, une impulsion sur le

- reset du microcontrôleur. Son utilité est de permettre le redémarrage du microcontrôleur en cas de problème (exemple, la présence d'une boucle sans fin accidentelle).
- 5) Les Timers sont des temporisateurs ou compteurs programmables. Ils permettent la génération des impulsions calibrés ou des signaux PWM, et la mesure du temps ou d'évènements. Ils peuvent générer des interruptions et s'accompagner de pré-diviseur (prescaler).
  - 6) Les interfaces spécifiques : Certain microcontrôleurs possèdent des entrées comparateur, ou encore des convertisseurs A/D et D/A très utiles dans la gestion des processus automatisés.
  - 7) Les ports série d'entrées/sorties permettent l'échange de données entre le microcontrôleur et un périphérique bit par bit sur un fil unique, d'une manière séquentielle et suivant un ordre précis. Il y a deux types de liaison série : synchrone et asynchrone.
  - 8) Les contrôleurs de bus spéciaux de communication sont parfois présents par exemple : UART, I2C, SSP, CAN, FlexRay, USB, Ethernet, ...etc. Ils rendent le microcontrôleur plus onéreux mais nécessaires si l'application est basé sur un de ces bus.
  - 9) Le contrôleur d'interruptions est indispensable dans un microcontrôleur. Exceptionnels sont les cas où il ne faut pas envisager au moins un cas d'urgence ou tenir compte d'évènement asynchrones à l'exécution du programme. L'interruption est un programme qui permet l'interruption de l'exécution d'un programme (mais pas d'une instruction), pour réaliser une tâche prioritaire généralement de courte durée.
  - 10) Les ports parallèles d'entrées/sorties permettent l'échange de données entre le microcontrôleur et un périphérique généralement en 8 bits. Ils permettent de recueillir des informations en entrée (états de capteurs) ou d'envoyer des signaux binaires en sortie pour commander des actionneurs et piloter des modules de l'environnement extérieur.

Les broches de ces ports (Les lignes d'E/S ou encore le Pins) sont configurable individuellement en entrée ou en sortie.

La configuration ainsi que l'état logique de ces pins est effectuée par des opérations d'écriture et de lecture dans différents registres associés à chaque port. On trouve généralement :

- Un registre de direction pour une configuration en entrée ou en sortie,
- Un registre de donnée recopiant les états logique de chaque pin de port configuré en entrée,
- Un registre de donnée contrôlant le niveau logique de chaque pin de port configuré en sortie.

11) CPU le microprocesseur, noté aussi M.P.U. (Microprocessor unit) ou encore C.P.U. (Central Processing Unit) est un circuit intégré complexe appartenant à la famille des VLSI (Very large scale intégration) capable d'effectuer séquentiellement et automatiquement des suites d'opérations élémentaires.

#### Son rôle

Ce circuit remplit deux fonctions essentielles :

##### ***\*le traitement des données :***

On parle d'unité de traitement. Cette fonction est dédiée à l'U.A.L. Elle concerne la manipulation des données sous formes de transfert, opérations arithmétiques, opérations logiques....

##### ***\*le contrôle du système :***

Cette fonction se traduit par des opérations de décodage et d'exécution des ordres exprimés sous forme d'instruction. [12]

## II-7. Les familles des microcontrôleurs

- Plusieurs fabricants se partagent le marché des microcontrôleurs, citons INTEL, MOTOROLA, ATMEL, ZILOG, HITACHI, DALLAS SEMI, PHILIPS et enfin MICROCHIP. Chaque fabricant ne propose pas un seul microcontrôleur, mais des familles de microcontrôleurs. On peut citer à titre d'exemple :
- La famille MCS51 (8x31,8x51) de INTEL, la famille AVR (ATtiny 28, ATMEGA2560, AT90S8535) de ATMEL, la famille 68HCxxx de MOTOROLA (68HC11, 68HC811), la famille PIC 16Cxx (16C84,16F84) de MICROCHIP.[13]

## II-8. Microcontrôleur ATmega2560

Le modèle mega de la société ARDUINO est une carte électronique dont le cœur est un microcontrôleur ATMEL de référence ATmega2560. Ce dernier est un microcontrôleur 8bits de la famille AVR dont la programmation peut être réalisée en langage C.[14]

### II-8.1. Les principales caractéristiques du ATMEGA2560

FLASH= mémoire programme de 32Ko

RAM= 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)

Temporisateurs / compteurs : Temporisateur 0 , Temporisateur 2 ,Timer0 (comptage 8 bits) et Timer1 (comptage 16bits)

Chaque temporisateur peut être utilisé pour générer deux signaux PWM. (6 broches OCxA/OCxB)

Plusieurs broches multifonctions : certaines broches peuvent avoir plusieurs fonctions différentes choisies par programmation

PWM= 6 broches OC0A (PD6), OC0B (PD5), OC1A (PB1), OC1B (PB3), OC2A (PB3), OC2B (PD3)

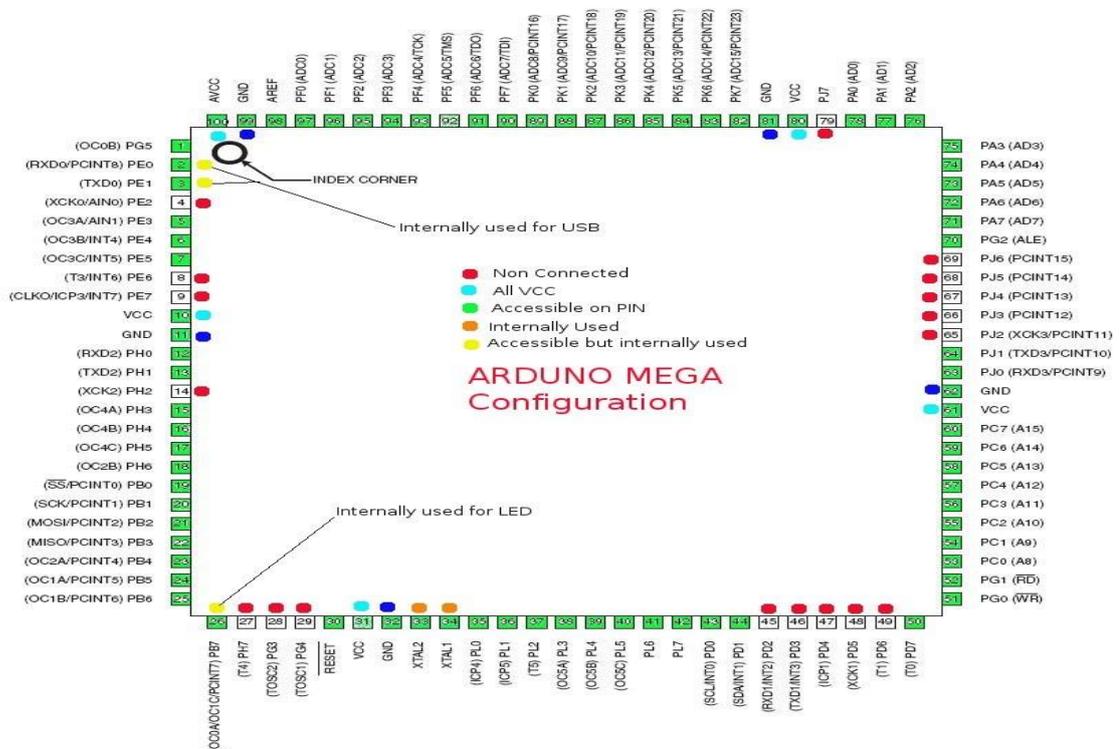


Figure II-2 : Schéma de microcontrôleur ATmega2560

Convertisseur analogique-numérique (résolution 10bits) = 6 entrées multiplexées ADC0(PC0) à ADC5(PC5)

Gestion bus I2C = le bus est exploité via les broches SDA(PC5)/SCL(PC4).

Port série = émission/réception série via les broches TXD(PD1)/RXD(PD0)

Comparateur Analogique= broches AIN0(PD6) et AIN1 (PD7) peut déclencher interruption

Gestion d'interruptions (24 sources possibles : en résumé

- Interruptions liées aux entrées INT0 (PD2) et INT1 (PD3)
- Interruptions sur changement d'état des broches PCINT0à PCINT23

- Interruptions liées aux Timers 0, 1 et 2
- Interruption liée au comparateur analogique
- Interruption de fin de conversion ADC
- Interruptions du port série USART
- Interruption du bus (I2C)

### ***II-8.2. Structure interne de l'ATMega2560***

L'utilisation des périphériques intégrés (Entrées Sorties, Timers, ...) repose sur l'exploitation (lecture/écriture) de registres internes. Ces registres (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 ATMega2560

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.

#### **II-8.2.1. Entrées / Sorties numériques**

Souvent, les microcontrôleurs disposent de broches d'entrée/sortie TOR, comme sur un automate programmable industriel. Pour placer l'état d'une sortie à 0 ou 1, ou lire l'état d'une entrée, il faut exploiter des registres internes, décrits ci-dessous. Les entrées-sorties sont réparties dans 3 groupes de broches appelés ports. Le port B regroupe les broches notées PBx, le port C les broches PCx et le port D les broches PDx. Chaque port est configuré/exploité grâce à 3 registres.

PORTx= pour l'écriture de valeurs en sortie.

DDRx = détermine la direction de chaque broche du port (1-sortie 0-entrée).

PINx= permet la lecture de la valeur en entrée.

Direction des ports : si le bit DDRB.2 vaut 1 alors la broche PB2est une sortie TOR.

Écriture des sorties TOR: si une broche est configurée en sortie ( $DDRx.n=1$ ) alors l'écriture du bit  $PORTx.n$  permet de définir l'état de la sortie (0 ou 1).

Ex:  $DDRB.5=1$  (donc PB5 en sortie) écrire 0 ou 1 dans le bit  $PORTB.5$  permet de définir l'état de la sortie PB5.

Lectures des entrées TOR: si une broche est configurée en entrée ( $DDRx.n=0$ ) alors la lecture du bit  $PINx.n$  permet de connaître l'état de l'entrée.

Ex:  $DDRB.4=0$  (donc PB4 en entrée), lire  $PINB.4$  permet de connaître l'état de l'entrée PB4.

**PORTB – The Port B Data Register**

Bit	7	6	5	4	3	2	1	0	
0x05 (0x25)	PORTB7	PORTB6	PORTB5	PORTB4	PORTB3	PORTB2	PORTB1	PORTB0	PORTB
Read/Write	R/W								
Initial Value	0	0	0	0	0	0	0	0	

**DDRB – The Port B Data Direction Register**

Bit	7	6	5	4	3	2	1	0	
0x04 (0x24)	DDB7	DDB6	DDB5	DDB4	DDB3	DDB2	DDB1	DDB0	DDRB
Read/Write	R/W								
Initial Value	0	0	0	0	0	0	0	0	

**PINB – The Port B Input Pins Address**

Bit	7	6	5	4	3	2	1	0	
0x03 (0x23)	PINB7	PINB6	PINB5	PINB4	PINB3	PINB2	PINB1	PINB0	PINB
Read/Write	R	R	R	R	R	R	R	R	
Initial Value	N/A								

**Figure II.3 : les 3 registres pour les configurations du port B**

### II-8.2.2. Interruptions Externes

Un événement d'interruption dirige le flux d'exécution du programme à un morceau totalement indépendant du code, connu sous le nom "d'interruption sous-programme". Il existe de nombreuses sources d'interruptions qui sont disponibles pour un microcontrôleur. La plupart d'entre eux sont générés par les modules internes et sont connus comme les interruptions internes. Et il y a deux broches qui acceptent les signaux externes directement et permet aux sources externes pour générer des interruptions. Les broches impliquées doivent être

configurées en entrée. Broches INT0 (PD2)/INT1(PD3) : configurables pour déclencher les interruptions (n° 2 et 3) sur niveau 0, front négatif ou positif. La cause d'interruption est choisie par le registre.

L'interruption externe 1 est activé par le INT1 broche externe. Le niveau et les arêtes sur l'axe de INT1 externe qui activent l'alarme sont définis dans le tableau ci-dessous.

#### EICRA - External Interrupt Control Register A

The External Interrupt Control Register A contains control bits for interrupt sense control.

Bit (0x69)	7	6	5	4	3	2	1	0	
	-	-	-	-	ISC11	ISC10	ISC01	ISC00	EICRA
Read/Write	R	R	R	R	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

**Figure II-4 : registre ELCRA**

Pour générer une interruption.

**Tableau 2-1 : Le contrôle de sens d'interruptions.**

ISC01	ISC00	
0	0	Le faible niveau de INTx génère une demande d'interruption
0	1	Tout changement logique sur INTx génère une demande d'interruption
1	0	Le front descendant du INTx génère une demande d'interruption
1	1	Le front montant du INTx produit dans une demande d'interruption

#### **II-8.2.3. Temporisation / comptage de ATmega2560**

Les microcontrôleurs ATmega2560 disposent de modules de temporisation/comptage internes, fonctionnant pour certains avec des registres de comptage sur 8 bits, et pour d'autres sur 16 bits. Dans tous les cas, chaque

événement de comptage conduit à une modification du registre de comptage (+1 ou -1). L'événement de comptage peut être un "tick" de l'horloge du microcontrôleur, ce qui revient à mesurer l'écoulement du temps. L'événement de comptage peut aussi être un front sur une broche d'entrée du microcontrôleur (les broches T0 et T1 peuvent servir d'entrée de comptage).

\*Fonction Temporisateur : lorsqu'on compte des "ticks" de l'horloge qui cadence le microcontrôleur, on mesure du temps. Les modules temporisation/comptage permettent de compter les ticks du signal d'horloge ou d'un signal de fréquence plus faible.

\*Fonction Compteur : lorsque l'on compte des fronts sur une entrée de comptage (broches T0 ou T1), on utilise alors la fonction "compteur" du module. Le choix entre fonction de temporisation et fonction de comptage se fait par paramétrage de registres dédiés à la gestion des modules temporisation/comptage.

\*Génération de signaux périodiques : les modules temporisation/comptage sont assez complexes et chacun de ces modules peut générer des signaux PWM dont le rapport cyclique est facilement modifiable.

Remarque : ces périphériques intégrés sont assez complexes (environ 70 pages du datasheet de L' ATmega). Seule une vision simplifiée est fournie ici.

#### **II-8.2.4. PWM (Pulse Width Modulation) [15]**

Un signal PWM est défini par son rapport cyclique (le temps au niveau haut par rapport à la période totale) mais aussi par sa fréquence. Grâce à l'environnement de programmation simplifié, une PWM s'obtient sur une Arduino avec la fonction **analogWrite** utilisé sur une patte numérique digitale 3, 5, 6, 9, 10 ou 11.

##### **II-8.2.4.1. Modulation de largeur d'impulsion PWM**

Est une technique pour fournir une puissance électrique à une charge qui a une réponse relativement lente. Le signal d'alimentation est constitué d'un

train d'impulsions de tensions de telle sorte que la largeur des impulsions individuelles de contrôler le niveau de tension efficace à la charge. Les deux signaux AC et DC peuvent être simulés avec PWM. Dans ces notes, nous allons décrire l'utilisation de PWM sur un Arduino pour les LED de contrôle et moteurs à courant continu.

La figure II-5 montre un signal de tension comprenant des impulsions de durée pour que la répétition de toutes les unités de temps  $T_c$ . La sortie d'un canal de PWM est soit  $V_s$  volts durant l'impulsion de zéro volt ou autrement. Si ce signal est fourni en entrée d'un dispositif qui a un temps beaucoup plus grande que  $T_c$  de réponse, le dispositif connaîtra le signal en entrée d'environ DC avec une tension efficace de  $V_{eff} = V_s \frac{\tau_o}{\tau_c}$ . Le rapport  $\frac{\tau_o}{\tau_c}$  est appelé le rapport cyclique des impulsions à ondes carrées. La tension continue effective fournie à la charge est contrôlée en ajustant le rapport cyclique.

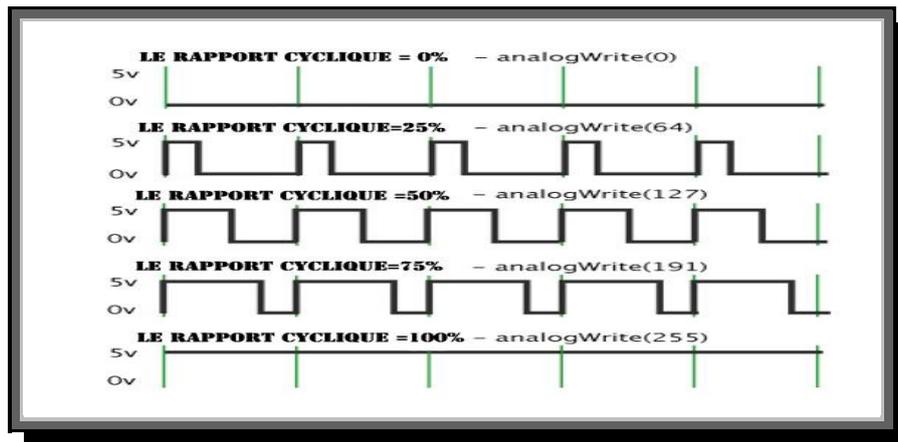
La figure II-6 montre un signal PWM avec une période impulsion  $T_0$  variable aussi la tension de signal PWM et  $V_0$  tel que  $V_s = V_{MAX} \frac{T_0}{T_c}$  et  $m = \frac{T_0}{T_c}$  et appelle rapport cyclique des impulsion carres donne la tension de ..... De la durée de l'impulsion  $T_0$

Exp :

Dans le cas d'un signal carre Te l que  $m=50\%$   $V_s \cong \frac{V_{MAX}}{2}$



**Figure II-5 : le rapport cyclique de pwm.**



**Figure II-6 : exemple du rapport cyclique**

La fréquence de la PWM est d'environ 490Hz.

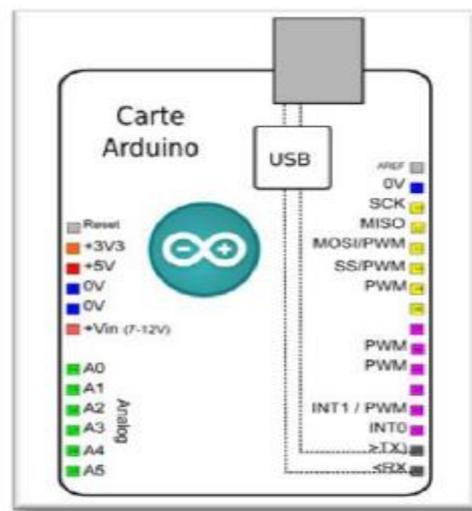
## II-9. Présentation générale de l'ARDUINO [16]

L'Arduino est une «plate-forme de prototype électronique open-source» permettant d'utiliser simplement un microcontrôleur ATMEL AVR.

Le système Arduino est une carte électronique (figure II-7) basée autour d'un microcontrôleur et de composants minimum pour réaliser des fonctions plus ou moins évoluées à bas coût. Elle possède un port USB pour la programmer. C'est une plateforme open-source qui est basée sur une carte à microcontrôleur (de la famille AVR), et un logiciel, c'est véritable environnement de développement intégré, pour écrire, compiler et transférer le programme vers la carte à microcontrôleur. Arduino peut être utilisée pour développer des applications matérielles industrielles légères ou des objets interactifs (création artistiques par exemple), et peut recevoir en entrées une très grande variété de capteurs. Arduino peut aussi contrôler une grande variété d'actionneurs (lumières, moteurs ou toutes autres sorties matériels). Les projets Arduino peuvent être autonomes, ou communiquer avec des logiciels sur un ordinateur (Flash, Processing ou MaxMSP). Les cartes électroniques peuvent être fabriquées manuellement

ou bien être achetées pré assemblées le logiciel de développement open-source est téléchargeable gratuitement.

Par sa simplicité d'utilisation, Arduino est utilisée dans beaucoup d'applications comme l'électronique industrielle et l'informatique embarquée, la modélisation, la robotique et la domotique, et bien plus encore.



**Figure II-7 : carte Arduino**

## II-10. Critère de choix [17]

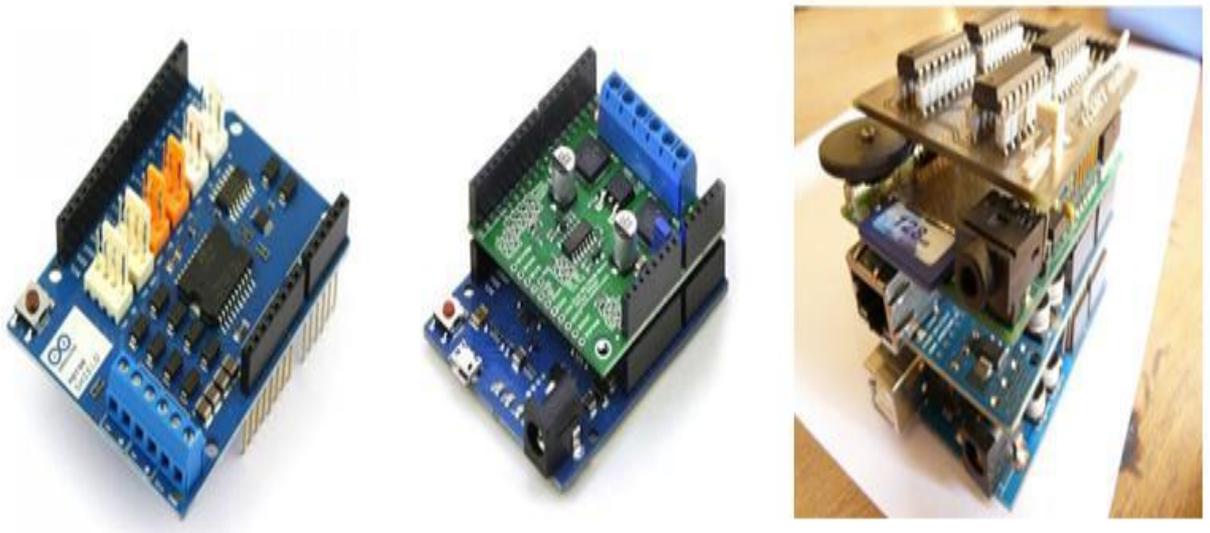
Malgré la disponibilité de nombreux microcontrôleurs et de nombreuses plateformes de développement des fonctions comparables à Arduino, la majorité d'entre-elles présente soit des limitations d'exploitation, ou des licences limitées, Il suffit d'utiliser n'importe quelle autre plateforme propriétaire pour se rendre compte des avantages indéniables et confirmés du système Arduino : l'alliance entre un hardware et un software garantissent la pérennité de ce système de développement ce qui lui donne toute sa puissance.

Donc, Arduino se distingue par rapport aux autres dans les points suivants :

- ✓ Cour faible : les cartes Arduino sont relativement peu coûteuses comparativement aux autres plateformes. La moins chère des versions du module Arduino peut être assemblée à la main, et même les cartes

Arduino pré-assemblées coûtent relativement moins de 25 Euros (microcontrôleur inclus...).

- ✓ Multiplateformes : Le logiciel Arduino, écrit en Java, tourne sous les systèmes d'exploitation Windows, Macintosh et Linux. La plupart des systèmes à microcontrôleurs sont limités à Windows.
- ✓ Un environnement de programmation clair et simple: L'environnement de programmation Arduino est facile à utiliser, tout en étant assez flexible pour que les utilisateurs avancés puissent en tirer profit également.
- ✓ Pour l'enseignement : Arduino est compatible avec l'environnement visuel basic C #, C++, JAVA... Ainsi d'autres logiciels.
- ✓ Logiciel Open-Source et extensible : Le logiciel Arduino et son langage sont publiés sous licence open source, disponible pour être complété par des programmeurs expérimentés.
- ✓ Existence de « Shields » : Ce sont des modules supplémentaires (figureII-8) qui se connectent parfaitement sur la carte Arduino pour augmenter les possibilités comme par exemple : afficheur graphique couleur, interface Wifi, Interface de puissance, etc.



**Figure II-8 : Quelques Shields pour Arduino**

## II-11. Carte Arduino Mega [17]

### II-11.1. Synthèse des caractéristiques

**Tableau II-2 : Synthèse des caractéristiques de la carte Arduino mega**

Microcontrôleur	Atmega2560
Tension de fonctionnement	5V
Tension d'alimentation (recommandée)	7-12V
Tension d'alimentation (limites)	6-20V
Broches E/S numériques	54 (dont 15 fournissent la sortie PWM)
Broches d'entrées analogiques	16
Intensité maxi disponible par broche E/S(5V)	20 mA
Intensité maxi disponible pour la sortie 3.3V	50 mA
Intensité maxi disponible pour la sortie 5V	Fonction de l'alimentation utilisée – 500 mA max si port USB utilisé seul
Mémoire Programme Flash	256 KB dont 8 KB utilisé par bootloader
Mémoire SRAM (mémoire volatile)	8 KB (Atmega2560)
Mémoire EEPROM (mémoire non volatile)	4 KB (Atmega2560)
Vitesse d'horloge	16 MHz

### II-11.2. Les éléments de la carte Mega

L'Arduino de base possède un microcontrôleur, qui rassemble la mémoire (morte et vive), le processeur et les périphériques d'entrées et sorties, des entrées et sorties USB et des prise d'alimentation pour transmettre les

informations et alimenter l'Arduino avec une tension entre 7 et 12V, permettant d'utiliser différents types d'alimentations, des LED qui servent surtout à indiquer l'état d'activité de l'Arduino, un bouton reset manuel, une entête ICSP qui permet de programmer le microcontrôleur sans passer par la «couche» Arduino; une puce FTDI USB qui convertit l'information USB et enfin des pins dans tous les sens.

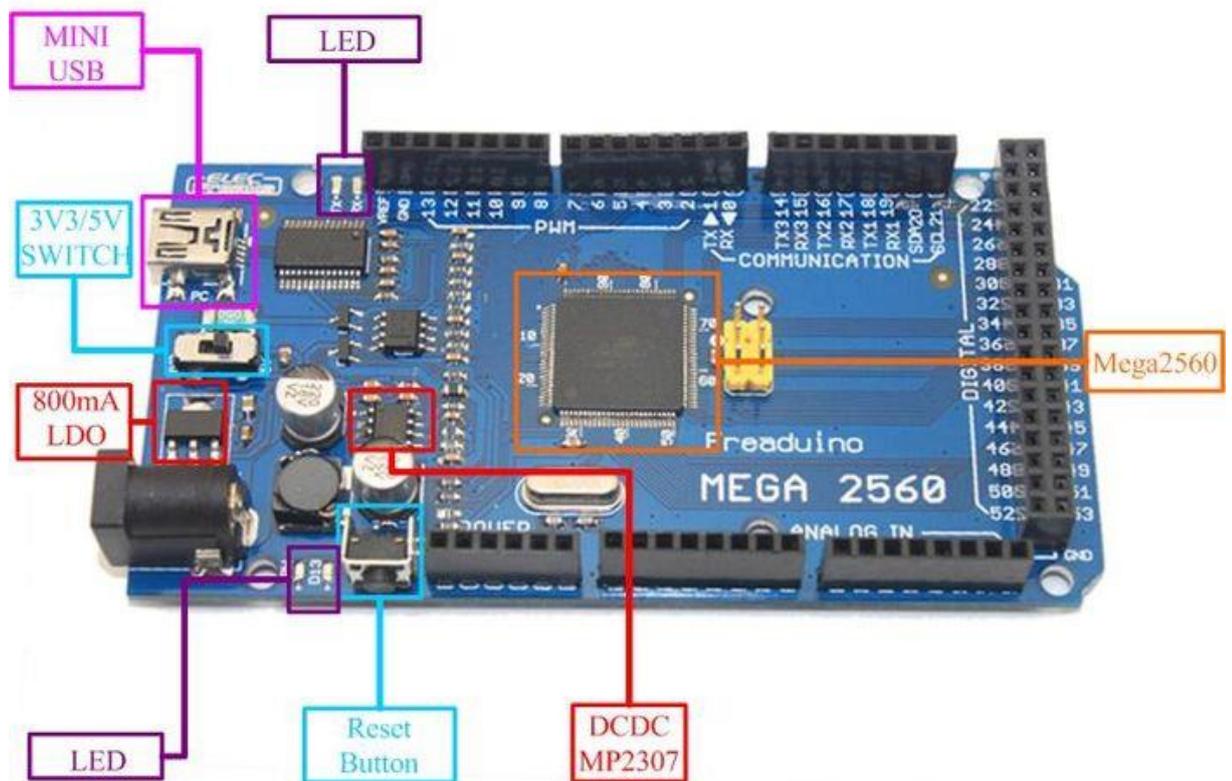


Figure II-9 : Les éléments de la carte Arduino mega

✓ **Alimentation :**

La carte Arduino mega peut être alimentée par l'USB ou par une alimentation externe. La source est sélectionnée automatiquement. La tension d'alimentation extérieure (hors USB) peut venir soit d'un adaptateur AC-DC ou de piles. L'adaptateur peut être connecté grâce à un jack (prise

d'alimentation) de 2.1mm positif au centre. Le raccordement vers un bloc de piles peut utiliser les bornes Gnd et Vin du connecteur d'alimentation (POWER).

Ce circuit inclut un régulateur de tension à 5 V mais il doit être alimenté entre 6 et 20 V. On conseille en général de l'alimenter plutôt entre 7 et 12 V pour garder une marge en basse tension et éviter que le circuit ne chauffe trop (car le régulateur de tension disperse toute surtension en chaleur).

✓ **Mémoire :**

Le ATmega2560 a 256 Ko de mémoire flash pour le stockage de code (dont 8 Ko est utilisé pour le bootloader), 8 Ko de SRAM et 4 Ko de mémoire EEPROM (qui peut être lu et écrit avec la bibliothèque de l'EEPROM).

✓ **Entrées et sorties :**

Chacune des 14 broches numériques de la mega peut être utilisée en entrée (input) ou en sortie (output) Elles fonctionnent en logique TTL (0V-5V), chacune pouvant fournir (source) ou recevoir un courant maximal de 40 mA. Certaines broches ont des fonctions spécialisées :

- Serial : broche 0 (RX) et broche1 (TX). Permet de recevoir (RX) et de transmettre (TX) des données séries TTL.
- Interruptions externes 2 et 3. Ces broches peuvent être configurées pour déclencher une interruption sur un front montant ou descendant, ou encore sur le changement de valeur.
- PWM : 3, 5, 6, 9, 10, et 11. Output 8-bit de PWM avec la fonction analog Write().
- LED : 13. Il y a une LED connectée à la broche numérique 13.

La carte mega a 6 broches d'entrée analogiques, A0 à A5, chacune avec 10 bits de résolution (1024 valeurs différentes).

- I2C : Permettent le support du bus I2C.

***II-11.3. La communication***

Le Mega 2560 dispose d'un certain nombre de moyens pour communiquer avec un ordinateur, ou avec d'autres microcontrôleurs. Le ATmega2560 fournit quatre UART sous la norme TTL (5V) de communication série. Un ATmega16U2, Le Arduino Software (IDE) comprend un moniteur de série qui permet aux données textuelles simples por' être envoyés vers et depuis la carte.

Conte à travers le port USB comme le montre la figure

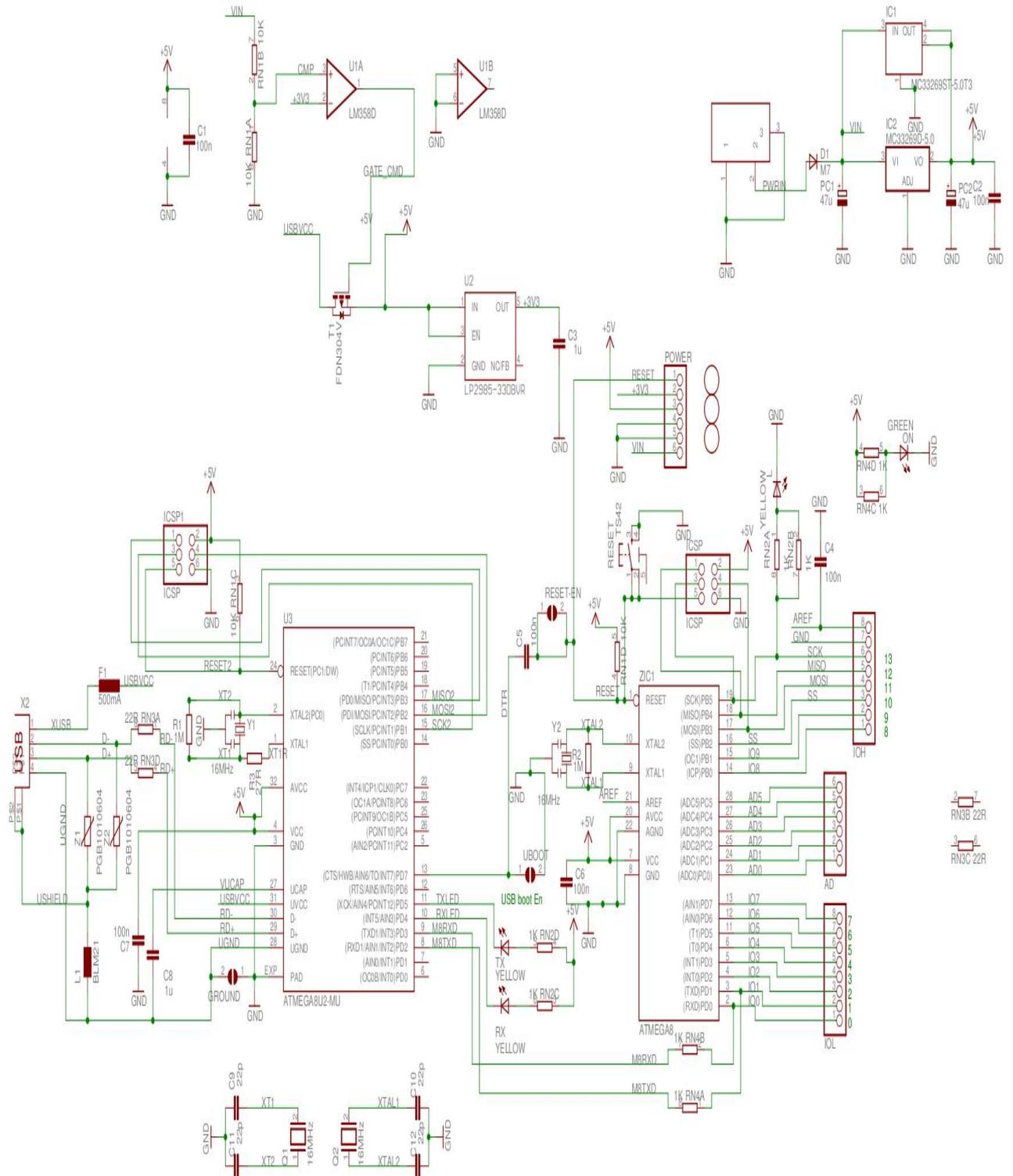


Figure II-10 : Schéma de carte Arduino Mega

## II-12. Exemple de l'application de la carte

### II-12.1 cas d'un serveur

Les servomoteurs utilisent la rétroaction pour déterminer la position de l'arbre, on peut contrôler cette position très précise. En conséquence, les servomoteurs sont utilisés pour contrôler la position des objets, à pivoter des objets, déplacer, les bras ou les mains de robots, déplacer des capteurs etc. avec une grande précision. Les servomoteurs sont de petite taille, et parce qu'ils ont des circuits intégrés pour contrôler leur mouvement, ils peuvent être connectés directement à un Arduino.

La plupart des servo-moteurs ont trois connexions suivantes :

- Noir / brun fil de terre.
- fil d'alimentation rouge (environ 5V).
- Jaune ou blanc fil de PWM.

Dans cette expérience, nous allons connecter les broches d'alimentation et de masse directement aux broches Arduino 5V et GND. L'entrée PWM sera reliée à l'une des broches de sortie numériques de l'Arduino.

#### Matériel requête

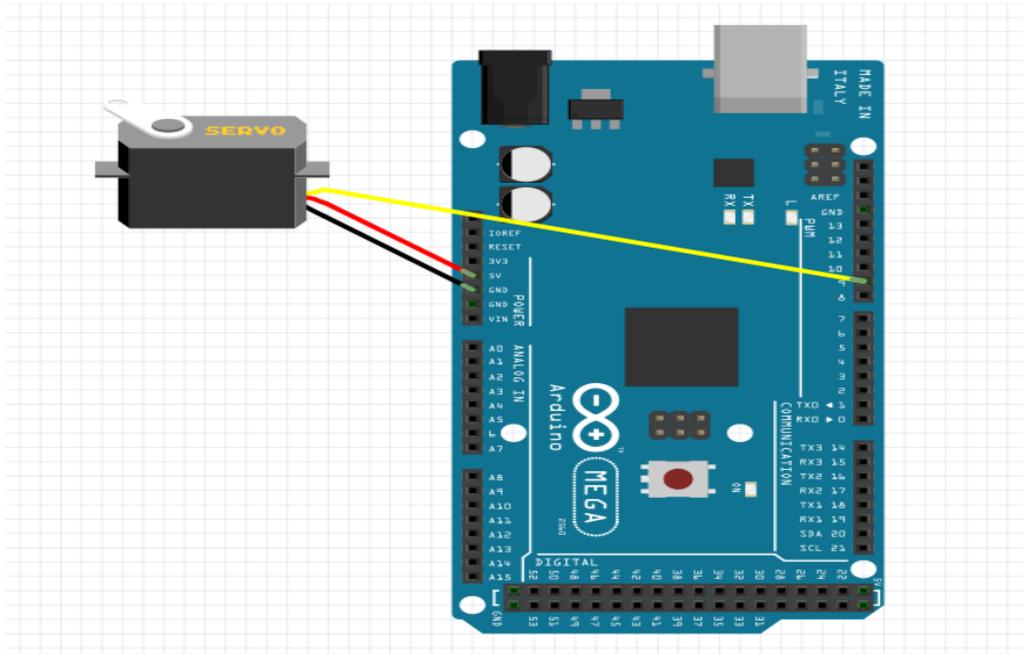
- servo-moteur 1 x TowerPro SG90
- 1 x Arduino Mega2560
- 3 x fils volants

#### Schéma de câblage

La meilleure chose à propos d'un servo-moteur est qu'il peut être connecté directement à un Arduino. Se connecter au moteur à l'Arduino comme indiqué la figure II-11 que :

- Servo fil rouge - 5V broches Arduino
- Servo fil marron - Rez-de-pin Arduino

- Servo fil jaune - PWM (9) broches Arduino



**Figure II-11 : Carte Arduino avec servomoteur**

➤ *Code*

Lorsque le programme démarre, le moteur servo tournera lentement de 0 degrés à 180 degrés, un degré à la fois. Lorsque le moteur a tourné de 180 degrés, il commence à tourner dans l'autre sens jusqu'à ce qu'elle revienne à sa position d'origine.

## II-13 Conclusion

La conception détaillée permet d'avoir une vision plus précise sur le sujet et une compréhension plus profonde des tâches à réaliser. Elle mène également à prévoir les besoins matériels et logiciels nécessaires pour atteindre l'objectif.

Ce chapitre présente les différents diagrammes conceptuels du projet. Ces diagrammes vont être la base sur laquelle sera réalisé le projet, cette phase sera détaillée dans le chapitre suivant.

CHAPITRE III  
REALISATION ET MISE EN  
MARCHE DE RADAR

### III.Introduction

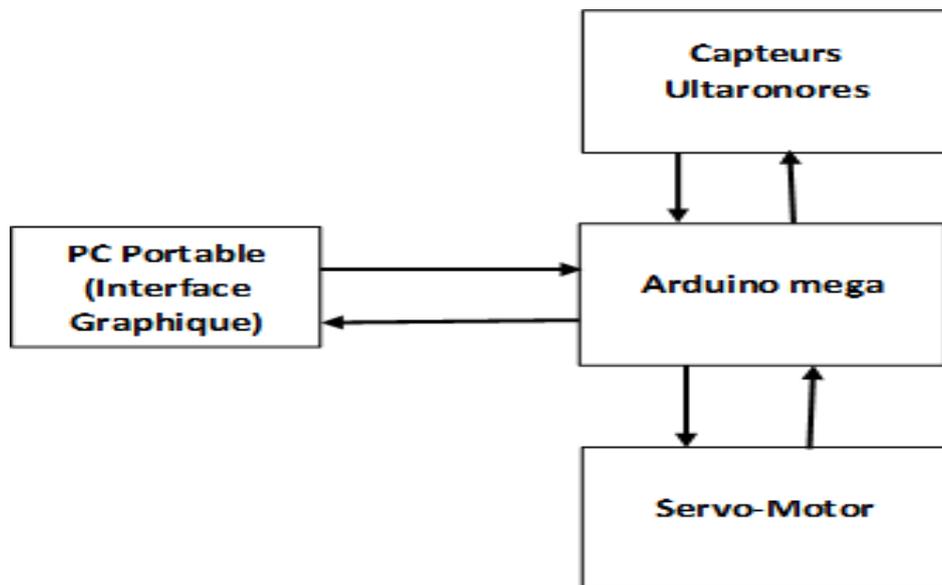
Dans ce chapitre, un ensemble des mesures et tests sont effectués sur les différents blocs de notre radar ultrasonore. Les résultats présentés ont été obtenus à travers des mesures sur des différentes parties de notre projet. Ce projet s'inscrit dans le cadre des applications Arduino . Il décrit une application pour un radar avec capteur à ultrasons monté sur un servomoteur pour créer une carte de l'environnement comme un demi-écran. Le lien à l'affichage haute résolution sur le PC se fait avec l'excellent environnement Processing.

#### III.1 Principe de fonctionnement de Radar ultrasonore

Les ultrasons se déplacent dans l'air à la vitesse de **330** mètres / seconde pour une température de **20**°c. Dans notre télémètre ultrasonique expérimental, nous faisons appel à des ultrasons caractérisés par une fréquence de **40** khz, ce qui correspond à une période de 10  $\mu$ s .En conséquence, la longueur d'onde, qui est la distance séparant par exemple deux crêtes consécutives du signal est égale à :  
$$\lambda = 330 \cdot 10^3 \cdot 10^{-6} = 8,25 \cdot 10^{-3} \text{m}$$
 ce qui correspond à environ 8 mm.

Le principe de fonctionnement de notre radar ultrasonique consiste à envoyer périodiquement une suite de 4 à 5 périodes élémentaires de 25 $\mu$ s ; et de mesurer la durée s'écoulant entre l'émission et la réception de l'écho engendré par la réflexion du signal sur un obstacle .Il est alors possible de déterminer la distance entre notre sonde ultrasonique et l'obstacle et la vitesse d'une cible en mouvement et même sa position en degré par rapport à notre capteur ultrasonore. elle correspond à la moitié de la distance parcourue par le signal, étant donné que ce dernier effectue en réalité un aller et retour.

Il suffit de se reporter au schéma bloc de la figure III-1 pour suivre les différents étages de ce Radar ultrasonique.



**Figure III- 1 : Schéma bloc avec circuit réalisé d'un radar ultrasonore**

Dans ce travail et avec Arduino on va montrer comment on peut réaliser ce radar en utilisant la carte Arduino avec le capteur ultrasonore et l'environnement

de développement Processing. On peut aussi regarder la figure suivante (figure III-2) ou lire les valeurs affichées sur l'interface.

### III-2. Interface de capteur ultrasonique avec le HC-SR04

Le capteur ultrasonore de distance HC-SR est une solution populaire et à faible coût pour la fonction de mesure de distance sans contact. Il est capable de mesurer des distances de 2cm à 400cm avec une précision d'environ 3 mm. Ce module comprend un émetteur et récepteur d'ultrasons, et son circuit de commande.

La figure III-1 suivante illustre l'interface graphique obtenir par logiciel professionnel Procissing3.2.1 :

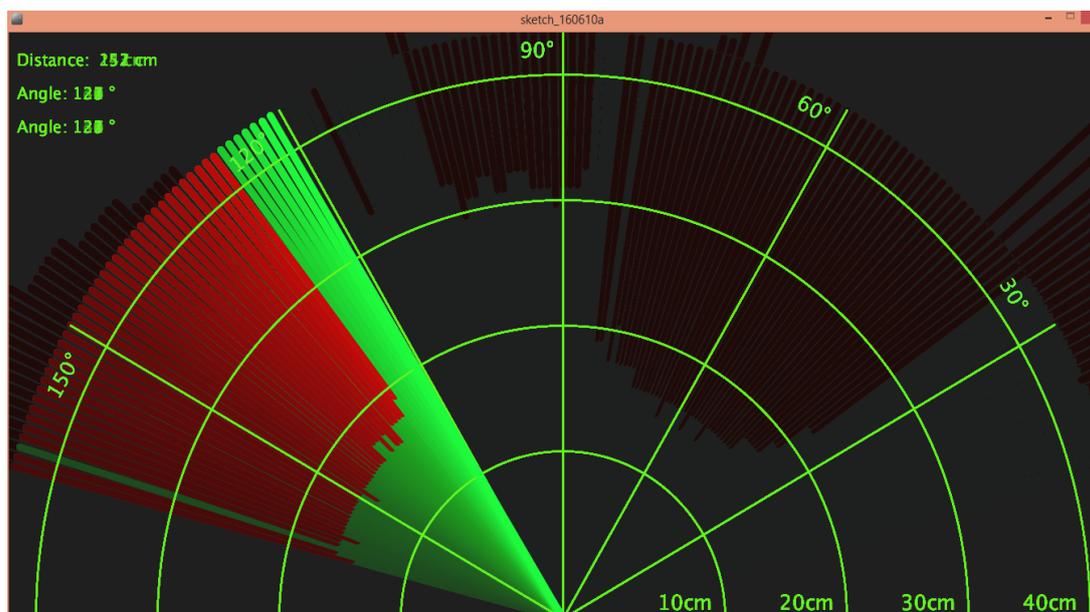


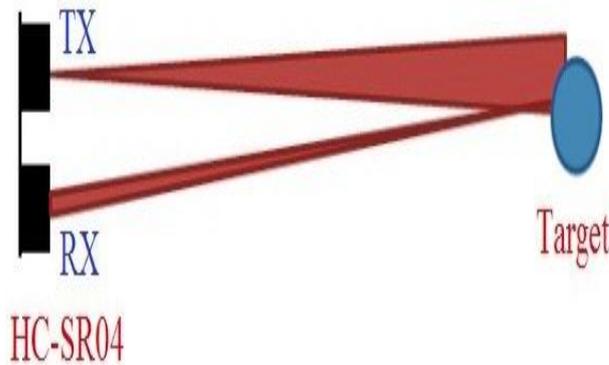
Figure III-2 :Interface graphique de détection de la position de radar

Module HC-SR04 dispose de 4 pins :

- **CCV** - 5V, de l'alimentation
- **TRIG** - Trigger Pin
- **ECHO** - Echo Pin
- **GND** - (-) de l'alimentation

TRIG et ECHO broches peuvent être utilisées pour l'interface de ce module avec une unité de microcontrôleur. Ce sont des broches de sortie d'entrée.

❖ HC-SR04 ultrasons Module de travail



**Figure III-3 : le capteur ultrasonore d radar avec illustration des ondes émis et reçus**

### III-3.fonctionnement de la communication entre le MVC et l'utilisateur

1. Fournir signal TRIGGER, atleast 10 us haut niveau (5V) impulsion.
2. Le module transmet automatiquement huit 40KHz rafale à ultrasons.
3. S'il y a un obstacle en face avant du module, il reflète l'impulsion ultrasonique.
4. Si le signal est de retour, la sortie ECHO du capteur sera en état HAUT (5V) pour une durée de temps nécessaire pour l'envoi et la réception de salve à ultrasons. La largeur d'impulsion est comprise entre environ 150µS à 25ms, et si aucun obstacle n'est détecté, la largeur d'impulsion d'écho sera d'environ 38ms.

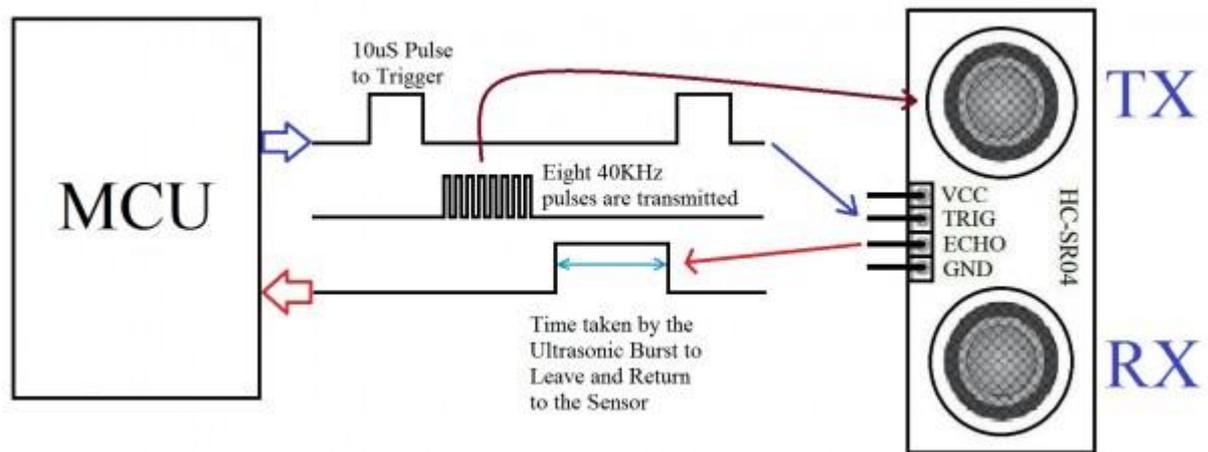


Figure III-4 : Fonctionnement de capteur ultrasonore de HC-SR04

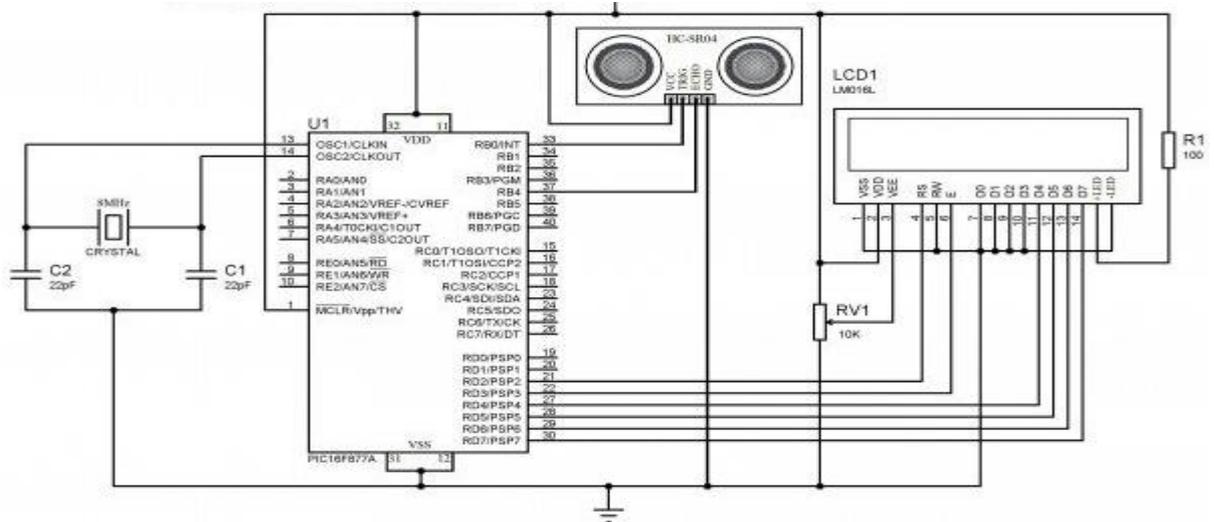


Figure III-5 : Montage électronique de capteur ultrason de distance modèle HC-SR04 avec Microcontrôleur PIC 16F877

❖ Calcul de distance

Distance = Vitesse \* Temps

Soit D la distance entre le capteur à ultrasons et Target

La distance totale parcourue par la salve d' ultrasons: 2d (avant et arrière)

Vitesse du son dans l' air: 340 m / s = 34000 cm / s

Ainsi,  $d = (34000 * \text{Time}) / 2$  , où  $\text{Time} = (\text{TMR1H} : \text{TMR1L}) / (1000000)$

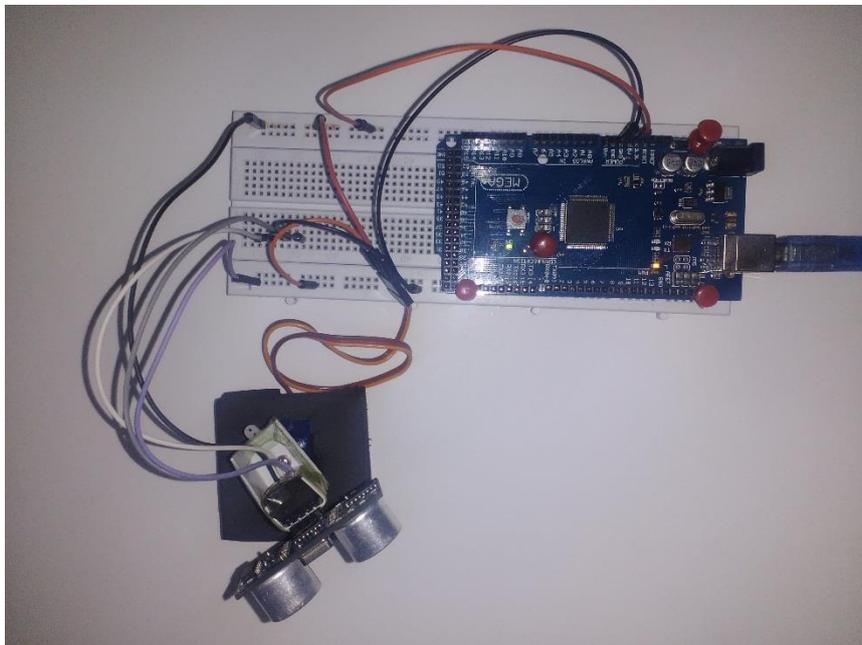
Par conséquent,  $d = (\text{TMR1H} : \text{TMR1L}) / 58.82 \text{ cm}$

$\text{TMR1H} : \text{TMR1L} = \text{TMR1L} | (\text{TMR1H} \ll 8)$

### ***III-3.1.Montage Avec Arduino***

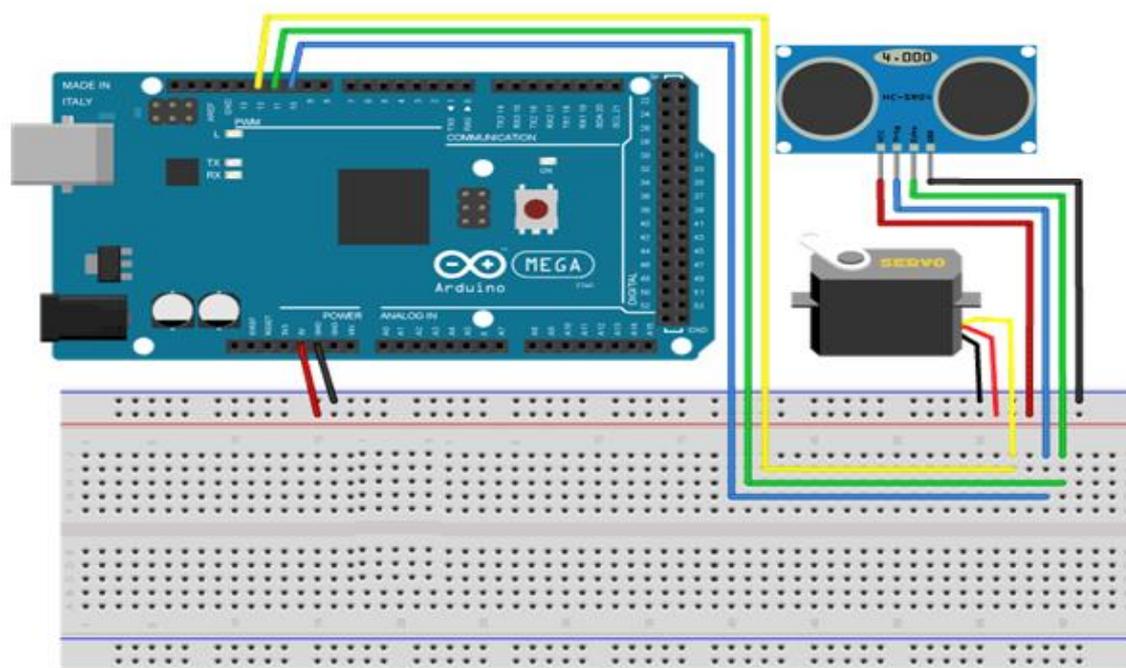
Le montage avec Arduino est généralement simple, avec un très simple circuit, on branche les fils avec la carte de test pour partager la connexion entre les trois composants principaux (Arduino, servomoteur et capteur ultrasonore).

On alimente en 5V le capteur et le servomoteur (fil rouge du servomoteur et Vcc du capteur). Le fil noir du servomoteur ainsi que le GND du capteur sont reliés au ground (GND) d'arduino. Le dernier fil du servomoteur sera relié à la broche 9 : c'est par là que les commandes arrivent au servomoteur. La broche Trig et echo du capteur sont reliées respectivement à la 12 et 11 de la carte arduino comme le montre la figure III-7



**Figure III-6 : Photo réelle de notre prototype de radar ultrasonore**

La figure suivante illustre l'animation de montage complet avec les connexions entre les fils et les trois composants principaux de Radar.



**Figure III-7 : Le schéma électronique de montage expérimental avec câblage entre les éléments de système**

### **III-3.2.Servomoteur**

Les servomoteurs sont des actionneurs. Très utilisés en modélisme et dans l'industrie, ils ont comme caractéristique principale leur « couple », c'est-à-dire la force de rotation qu'ils peuvent exercer. Plus un servomoteur aura de couple et plus il pourra actionner des « membres » lourds comme déplacer un bras qui porte une charge et même pour le balayage d'un radar ultrasonore. Pour la robotique de loisirs, les servomoteurs ont en général peu de couple et sont de taille réduite, bien adaptée à un encombrement minimal et à une énergie disponible limitée.

Le mode de commande d'un servomoteur est standardisé : on envoie sur son fil de commande une impulsion dont la durée correspond à l'angle désiré. Historiquement, cette impulsion était délivrée par un circuit oscillateur. Le circuit intégré NE555 est un exemple vedette du circuit utilisé. Avec la programmation de l'Arduino, ce circuit n'est plus nécessaire. Une bibliothèque (library) dédiée, la bibliothèque « servo », permet de piloter un servomoteur en lui transmettant simplement l'angle sur lequel il souhaite se positionner.

### III-4.Programmation

#### III-4.1. Programme pour déterminer la distance

L'organigramme de la programmation qui nous permet de mesurer la distance et la vitesse d'une cible en mouvement et organisé comme suite :

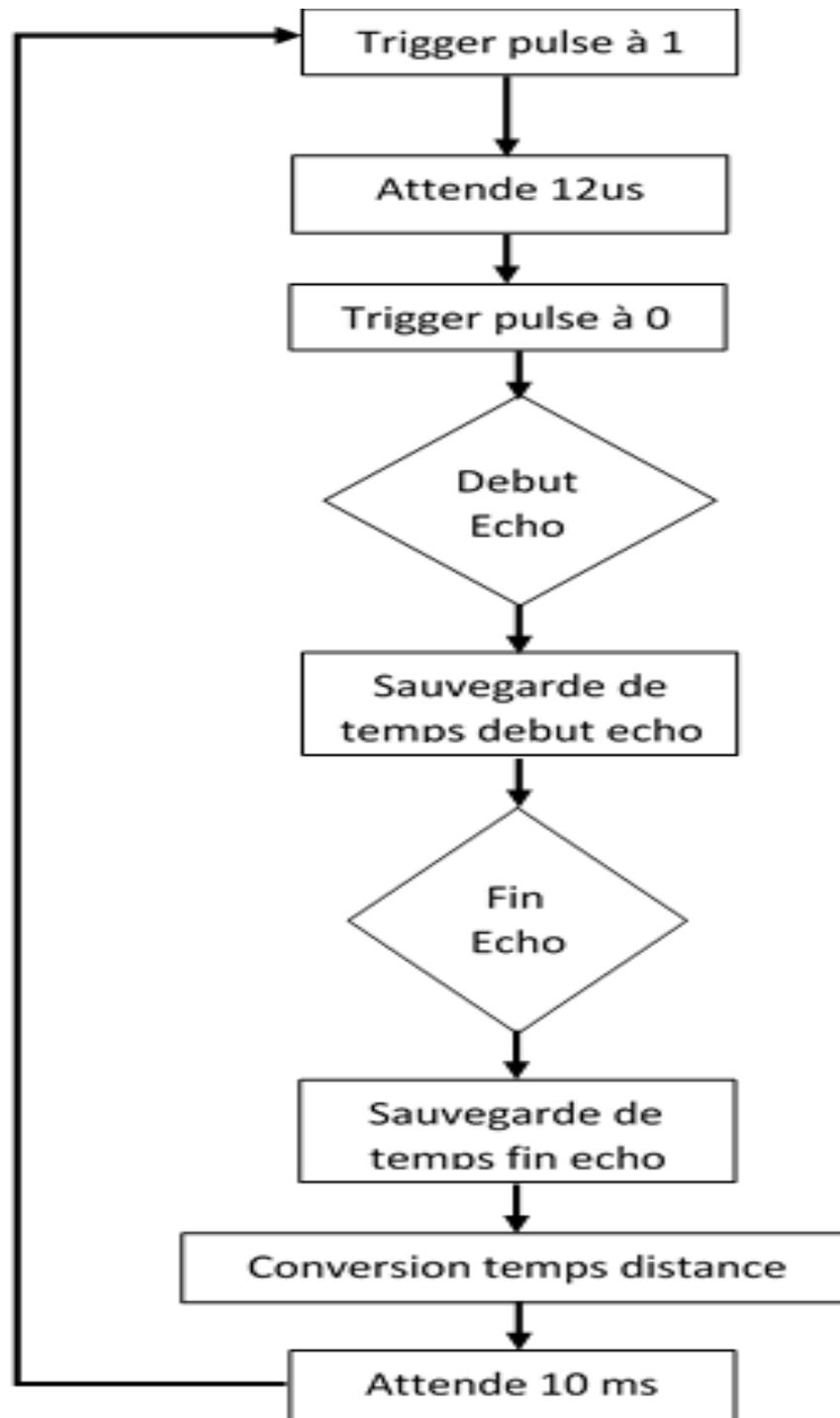


Figure III-8 : Organigramme pour mesurer la distance entre le capteur et la cible

Le programme utilisé pour déterminer la partie de la mesure vitesse de la cible en mouvement est basé sur le principe de la mesure de la distance mais avec la mesure de l'effet Doppler (voir partie 6 chapitre I).

#### ***III-4.2.Étapes de base sans servomoteur:***

- Fournir TRIGGER au module à ultrasons
- Ecoute Echo
- Démarrer la minuterie lorsque ECHO HIGH est reçu
- Arrêter la minuterie lorsque ECHO va LOW
- Lire la valeur Minuteur
- Convertir à Distance
- Affichez la distance et la vitesse de la cible

#### ***III-4.3.Programme servomoteur :***

Les étapes de programmation nécessaires au pilotage d'un servomoteur sont :

- L'utilisation de la librairie servo.h qui prend en charge la communication de bas niveau entre l'Arduino et le servomoteur.
- La création d'un objet de type Servo, appelé ici myservo.
- Le fil de commande de ce servo sera connecté au PIN 9 et l'objet myservo pilotera ce PIN.
- Demander au servomoteur de se déplacer à l'angle désiré, soit de façon absolue en lui indiquant une valeur entière (90°) dans le premier cas, soit en lui passant le contenu d'une variable (Angle) compris entre 0 et 180, ce qui peut être utile par exemple pour donner une progressivité au déplacement en faisant varier l'angle d'un pas fixe (quelques degrés) par une boucle.
- lire la valeur de l'angle du servomoteur pendant le balayage.

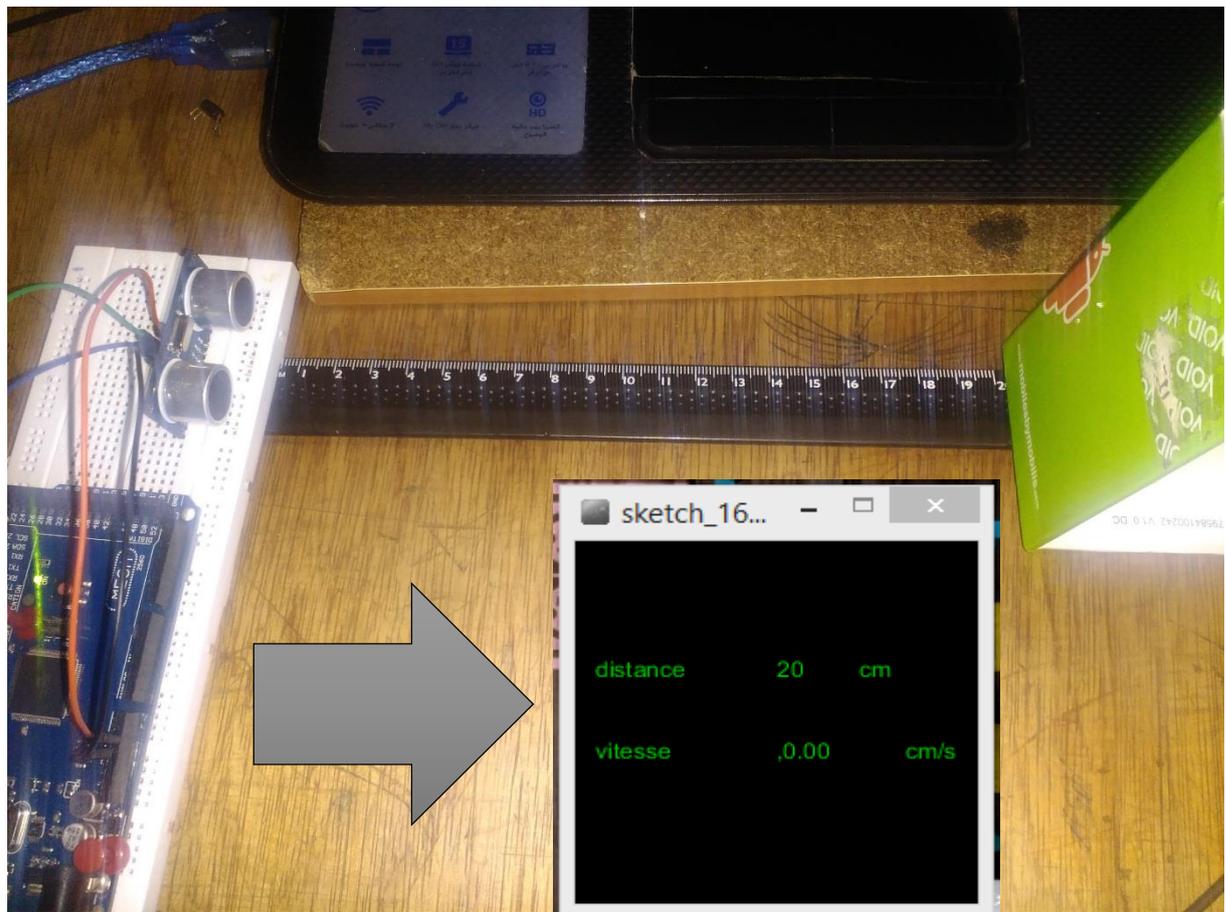
#### **III-5.Résultat et discussion**

Les résultats présentés dans cette partie se divisent en trois sections, la première consiste à déterminer la distance et la deuxième consacré à la vitesse

de la cible fixe et en mouvement et la dernière réservée pour détecter plusieurs obstacles avec sa position en degré.

### III-5.1. la méthode de mesure de la distance

Les figures suivantes valident le fonctionnement de montage.



**Figure III-9 : montage expérimental avec la mesure de la distance d'un obstacle**

#### ❖ Calibration de distance

Pour mesurer la distance exacte, on a étalonné le résultat obtenu. Ici, pour rendre la distance affichée plus précise, on ajoute 1 à la distance mesurée. Cette constante de calibration peut être trouvée en utilisant une série d'expériences pratiques avec une échelle de la règle c'est-à-dire d'une manière empirique.

Dans le programme ci - dessous il y a un petit problème car il attend infiniment pour ECHO et ECHO passe au niveau bas. La meilleure solution consiste à utiliser la fonction PORTB Sur-Change Interrompte du microcontrôleur arduino.

On note que les résultats de montage Arduino capteur ultrasonore affectent une sensibilité de distance très grande, l'erreur observée entre la valeur mesurée et affiché sur notre interface est presque nulle.

### III-5.2. méthodes mesure de la vitesse

De même les deux figures suivantes illustrent le montage avec les trois éléments principaux de notre expérience pratique pour déterminer la vitesse de main d'un être humain en mouvement.

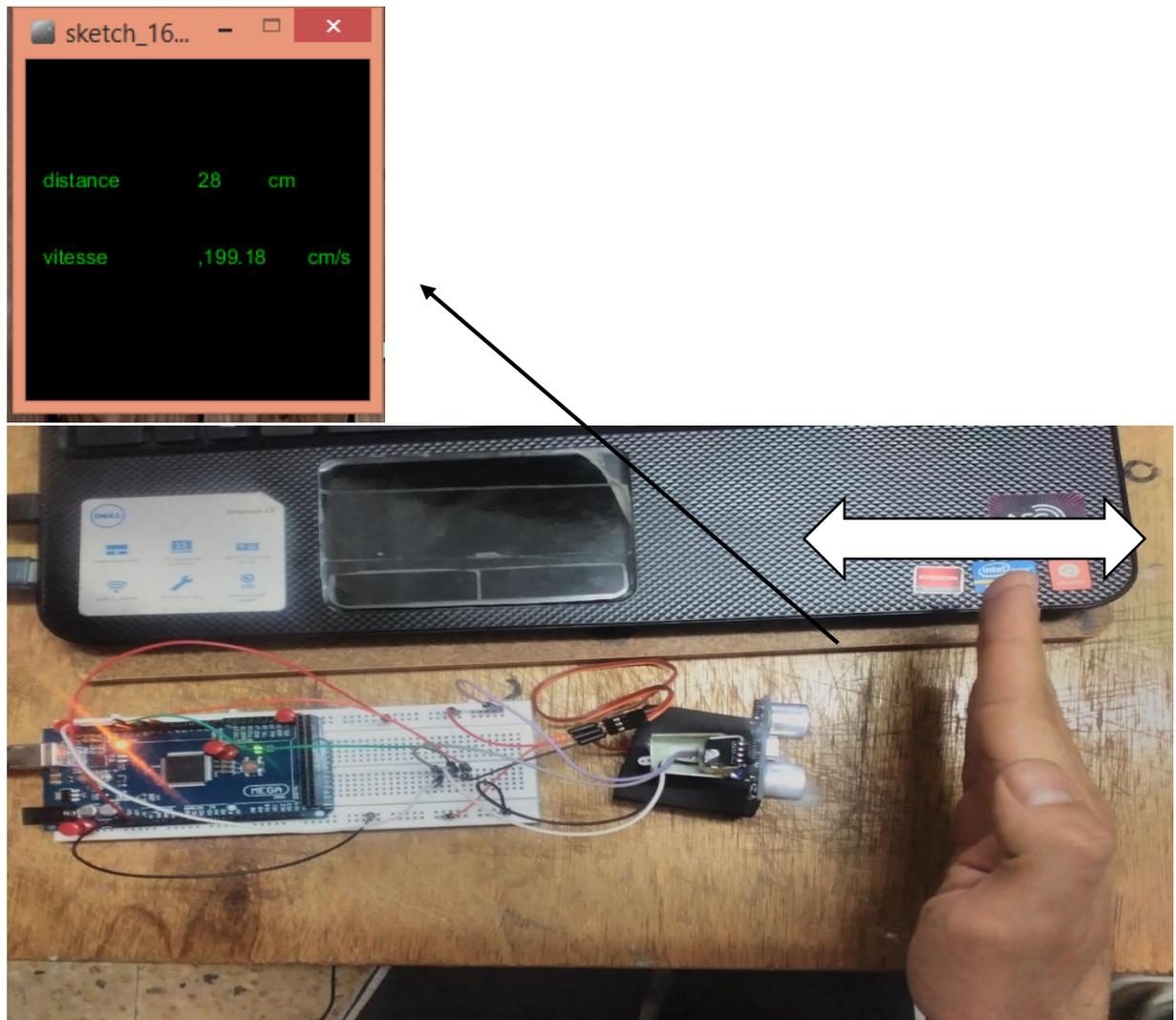


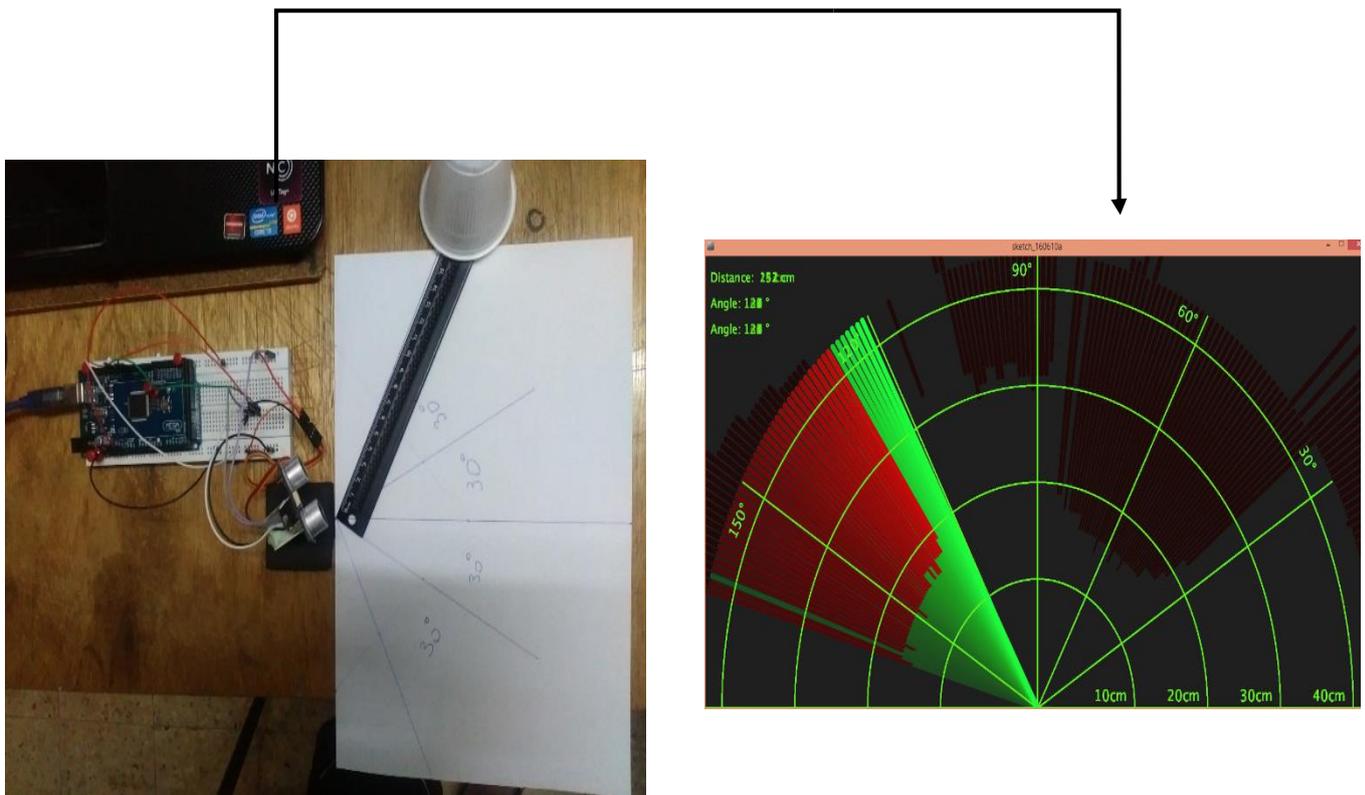
Figure III-10 : montage expérimental pour mesurer la vitesse de la cible en mouvement.

Sur ces figures, on note que le système de mesure détecte la distance de la main de l'expérimentateur et sa vitesse lorsqu'il bouge sa main aller-retour par rapport à notre capteur ultrasonore.

### III-5.3.Methode mesure de la position

Pour déterminer la position d'un obstacle on a placé deux Goblet l'un à droite et l'autre à gauche de capteur ultrasonore ; les deux figures suivante illustrent la détection de la position sur l'interface processing.

- On réalise un logiciel qui permet de visualise une portion de l'objet en utilisant langage JAVA.
- Un détecteur de balayage qui travaille avec le servo moteur.
- Un Radar qui permet de détecte l'objet et mémoriser la position en degré.



**Figure III-11 :démonstration du la détection de la position à gauche de capteur.**

Dans cette figure ,on note que le radar détecte la position de l'obstacle et montres en détaille sur l'interface avec une grande précision .

### **III-6. Conclusion :**

On a consacré ce chapitre pour la réalisation du Radar à l'aide des outille informatique (les logiciel) et électronique (le capture ultrasonore, le servomoteur, ...).

On a passé par quatre étapes essentielles : programmation des actions de système, réalisation des cartes électronique, réalisation les pièces du radar (coté mécanique) et finalement l'étape d'assemblage de tous les parties.

On à présenter Les étapes de la réalisation de radar et Les résultats des tests qui font sur toutes les mesures qui indiquent le bon fonctionnement du résultat.

### Conclusion générale

Ce projet nous a permis une bonne compréhension du système Radar, des différents mécanismes physiques et mécaniques de signaux Radar et des différentes techniques d'exploitation des ondes ultrasonores.

Nous avons réalisé notre Radar à base d'un microcontrôleur Arduino mega qui consiste à commander un servomoteur à courant continu et d'acquérir le signal ultrasonore reçu par le capteur pour afficher les résultats obtenue sur notre interface graphique.

Nous avons ensuite implémenté une sonde à base d'un émetteur et récepteur ultrasonore. Les signaux d'émissions et de réception sont générés à partir de la carte Arduino.

La conversion des niveaux en temps vers des niveaux en centimètre était possible via un algorithme de conversion programmé sous Arduino. l'affichage est assuré par l'interface graphique processing connecté à avec l'équipement de Radar via le câble USB et le microcontrolleur Arduino.

Les résultats des tests obtenus sont très satisfaisants. Ce travail nous a permis aussi de détailler des aspects pratiques sur les microcontrôleurs, les capteurs ultrasonores et les techniques de commandes d'un servomoteur.

Notre projet de fin d'études nous a offert l'opportunité d'aborder des aspects fondamentaux de l'informatique et de l'électronique analogique et numérique appliquée à l'instrumentation aéronautique.

## **Annexe A**

### **1. Applications d'Arduino**

Le système Arduino nous permet de réaliser un grand nombre de choses, qui ont une application dans tous les domaines on cite quelque exemples :

- contrôler les appareils domestiques
- fabriquer un robot
- faire un jeu de lumières
- communiquer avec l'ordinateur
- télécommander un appareil mobile (modélisme)

### **2. Les bonnes raisons de choisir Arduino**

Il existe pourtant dans le commerce, une multitude de plateformes qui permettent de faire la même chose. Notamment les microcontrôleurs « PIC ». Nous allons voir pourquoi choisir l'Arduino.

- Le prix : est moins chère
- La liberté : \* Le logiciel : gratuit et open source.
  - \* Le matériel : cartes électroniques dont les schémas sont en libre circulation sur internet.
- La compatibilité : Le logiciel, tout comme la carte, est compatible sous les plateformes les plus courantes (Windows, Linux et Mac).
- La communauté : La communauté Arduino est impressionnante et le nombre de ressources à son sujet est en constante évolution sur internet.

### 3. Les outils Arduino

À présent, rapprochons-nous de « l'utilisation » du système Arduino et voyons comment il se présente. Il est composé de deux choses principales, qui sont : le matériel et le logiciel. Ces deux outils réunis, il nous sera possible de faire n'importe quelle réalisation.

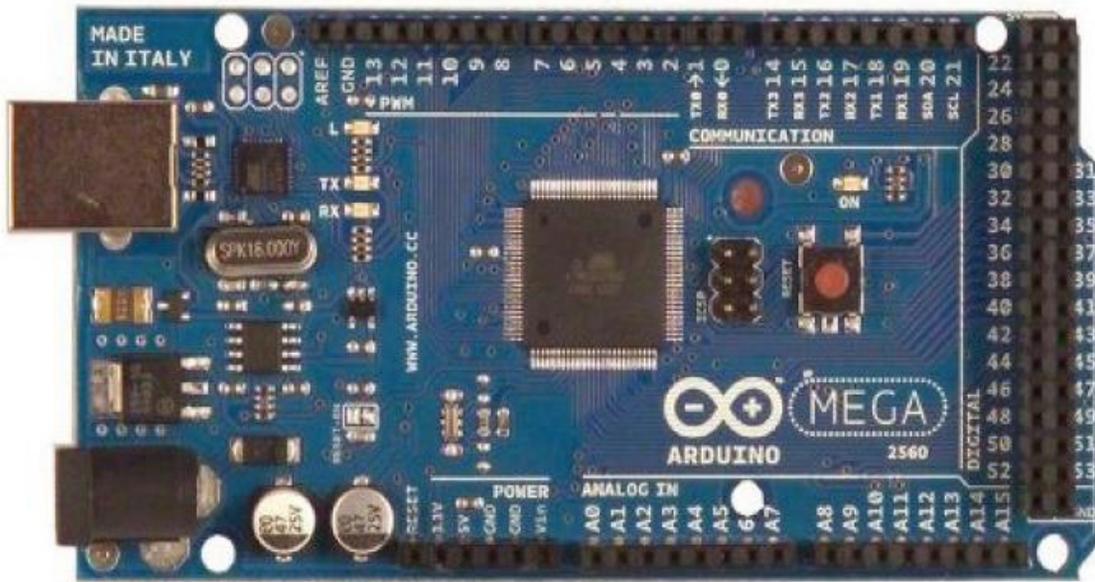


Figure a.1: carte Arduino "Mega"

➤ **Le matériel :**

Il s'agit d'une carte électronique basée autour d'un microcontrôleur Atmega du fabricant Atmel, dont le prix est relativement bas pour l'étendue possible des applications.

Voilà à quoi ressemble la carte que nous allons utiliser :

➤ **Le logiciel :**

Le logiciel va nous permettre de programmer la carte Arduino. Il nous offre une multitude de fonctionnalités.

#### 4. Les types de cartes

Il y a trois types de cartes :

- Les dites « officielles » qui sont fabriquées en Italie par le fabricant officiel : Smart Projects
- Les dites « compatibles » qui ne sont pas fabriquées par Smart Projects, mais qui sont totalement compatibles avec les Arduino officielles.
- Les « autres » fabriquées par diverse entreprise et commercialisées sous un nom différent (Freeduino, Seeduino, Femtoduino, ...).

#### 5. Description de la carte Arduino Mega 2560

La carte Arduino Mega 2560 est une carte à microcontrôleur basée sur un ATmega2560

Cette carte dispose :

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

Elle contient tout ce qui est nécessaire pour le fonctionnement du microcontrôleur; Pour pouvoir l'utiliser et se lancer, il suffit simplement de la connecter à un ordinateur à l'aide d'un câble USB (ou de l'alimenter avec un adaptateur secteur ou une pile, mais ceci n'est pas indispensable, l'alimentation étant fournie par le port USB).

## 6. Communication

La carte Arduino Mega2560 dispose de toute une série de facilités pour communiquer avec un ordinateur, une autre carte Arduino, ou avec d'autres microcontrôleurs. L'ATmega2560 dispose de quatre UARTs (Universal Asynchronous Receiver Transmitter ou émetteur-récepteur asynchrone universel) pour communication série de niveau TTL (5V) et qui est disponible sur les broches 0 (RX) et 1 (TX). Un circuit intégré ATmega8U2 sur la carte assure la connexion entre cette communication série de l'un des ports série de l'ATmega 2560 vers le port USB de l'ordinateur qui apparaît comme un port COM virtuel pour les logiciels de l'ordinateur. Le code utilisé pour programmer l'ATmega8U2 utilise le driver standard USB COM, et aucun autre driver externe n'est nécessaire

Le logiciel Arduino inclut une fenêtre terminal série (ou moniteur série) sur l'ordinateur et qui permet d'envoyer des textes simples depuis et vers la carte Arduino. Les LEDs RX et TX sur la carte clignote lorsque les données sont transmises via le circuit intégré ATmega8U2 utilisé en convertisseur USB-vers-série et la connexion USB vers l'ordinateur (mais pas pour les communications série sur les broches 0 et 1).

## 7. Caractéristiques Mécaniques

Les longueurs et largeurs maximales du circuit imprimé de la carte Mega2560 sont respectivement 10.16 cm et 5.33 cm, avec le connecteur USB et le connecteur d'alimentation Jack s'étendant au-delà des dimensions de la carte. Plusieurs trous de vis permettent à la carte d'être fixée sur une surface ou dans un boîtier. Noter que la distance entre les broches 7 et 8 est de 0.16 pouces, et non un multiple des 0.1 pouces séparant les autres broches.

## 8. Le logiciel Arduino

C'est un logiciel de programmation par code, code qui contient une cinquantaine de commandes différentes. A l'ouverture, l'interface visuelle du logiciel ressemble à ceci:

des boutons de commande en haut, une page blanche vierge, une bande noire en bas

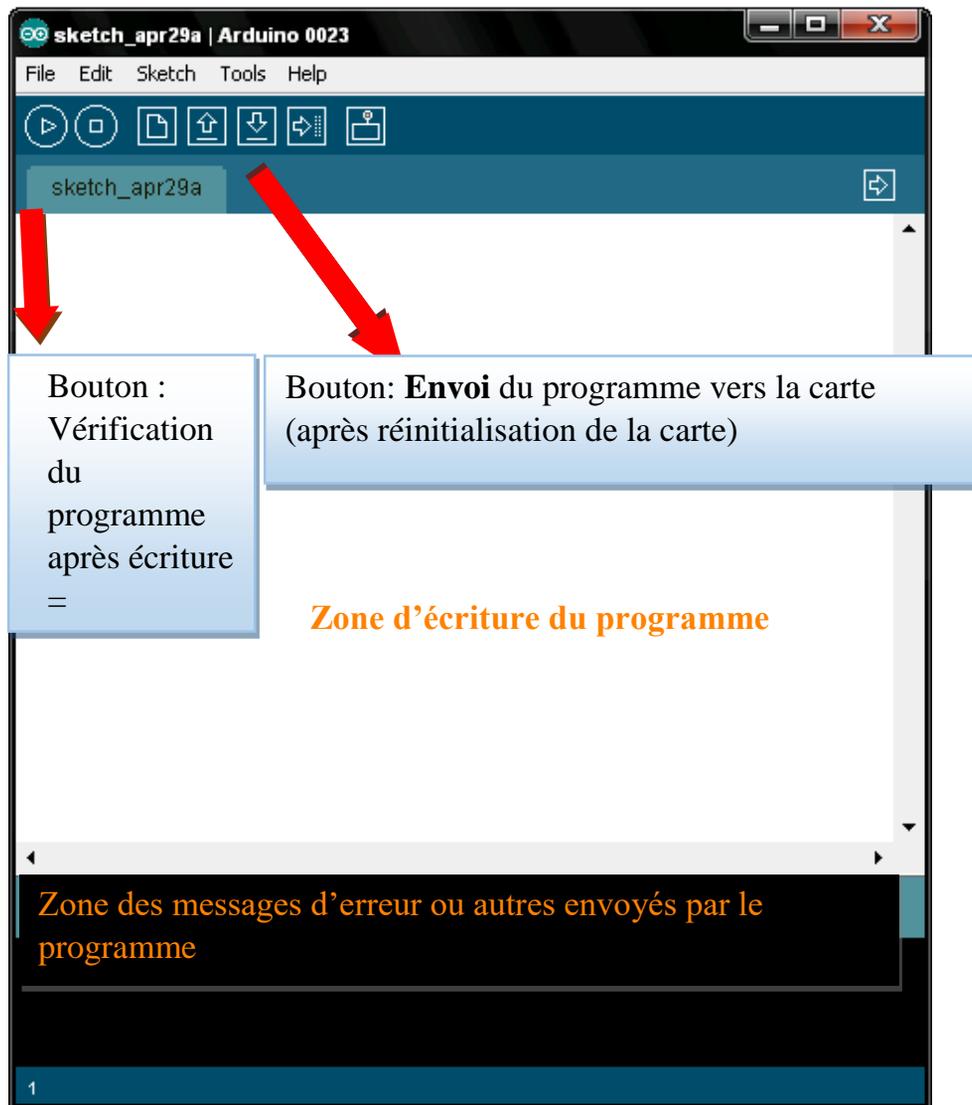


Figure a.2 : Logiciel Arduino

## 9. Structure d'un programme

Il y a trois phases consécutives

**Commentaires**

**Commentaires multilignes pour se souvenir du patch ==>**

```

/* Ce programme fait clignoter une LED branchée sur la broche 13
 * et fait également clignoter la diode de test de la carte
 */
int ledPin = 13; // LED connectée à la broche 13

void setup()
{
  pinMode(ledPin, OUTPUT); // configure ledPin comme une sortie
}

void loop()
{
  digitalWrite(ledPin, HIGH); // met la sortie à l'état haut (led allumée)
  delay(3000); // attente de 3 secondes
  digitalWrite(ledPin, LOW); // met la sortie à l'état bas (led éteinte)
  delay(1000); // attente de 1 seconde
}

```

**1/La définition des constantes et des variables**

**2/La configuration des entrées et sorties**  
**void setup()**

**3/La programmation des interactions et comportements**  
**void loop()**

Une fois la dernière ligne exécutée, la carte revient au début de la troisième phase et recommence sa lecture et son exécution des instructions successives. Et ainsi de suite.

Cette **boucle** se déroule des milliers de fois par seconde et anime la carte.

Done compiling.

22

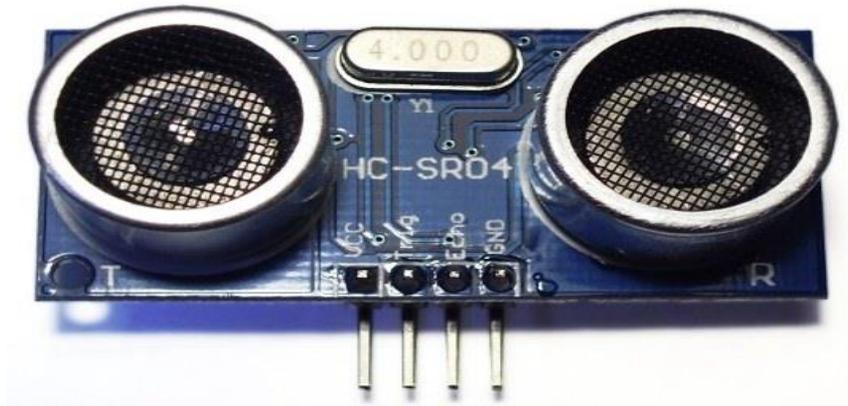
Figure a-3 : programme clignoter une LED

## Annexe B

### HC-SR04 - Module de détection aux ultrasons

Le capteur HC-SR04 utilise les ultrasons pour déterminer la distance d'un objet. Il offre une excellente plage de détection sans contact, avec des mesures de haute précision et stables. Son fonctionnement n'est pas influencé par la lumière du soleil ou des matériaux sombres, bien que des matériaux comme les vêtements puissent être difficiles à détecter.

Le brochage du module est le suivant :



### Caractéristiques

- Dimensions : 45 mm x 20 mm x 15 mm
- Plage de mesure : 2 cm à 400 cm
- Résolution de la mesure : 0.3 cm
- Angle de mesure efficace : 15 °
- Largeur d'impulsion sur l'entrée de déclenchement : 10  $\mu$ s  
(Trigger Input Pulse width)

### Broches de connection

- Vcc = Alimentation +5 V DC

- Trig = Entrée de déclenchement de la mesure (Trigger input)
- Echo = Sortie de mesure donnée en écho (Echo output)
- GND = Masse de l'alimentation

## Spécifications et limites

Paramètre Min Type Max Unité

Tension d'alimentation 4.5 5.0 5.5 V

Courant de repos 1.5 2.0 2.5 mA

Courant de fonctionnement 10 15 20 mA

Fréquence des ultrasons - 40 - kHz

Attention : la borne GND doit être connectée en premier, avant l'alimentation sur Vcc.

## Fonctionnement

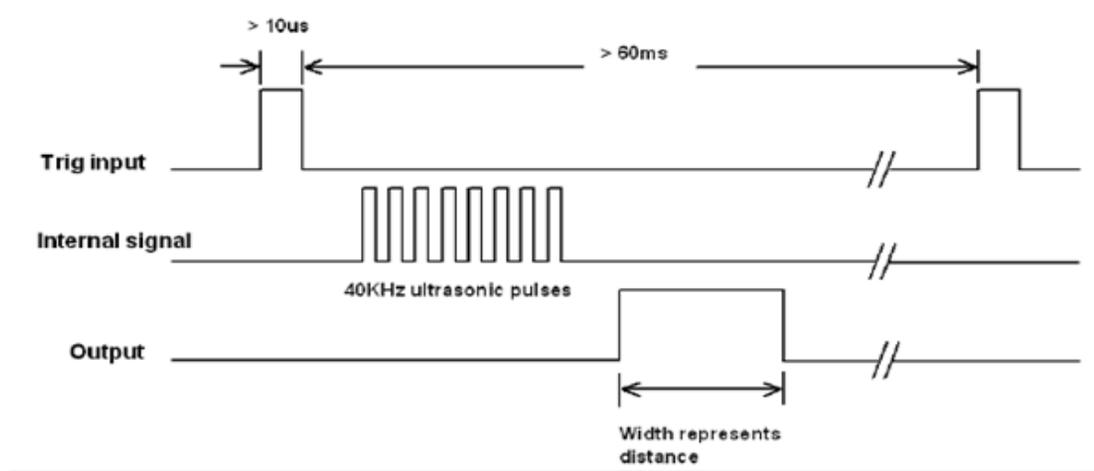
Pour déclencher une mesure, il faut présenter une impulsion "high" (5 V) d'au moins 10  $\mu$ s sur l'entrée "Trig". Le capteur émet alors une série de 8 impulsions ultrasoniques à 40 kHz, puis il attend le signal réfléchi. Lorsque celui-ci est détecté, il envoie un signal "high" sur la sortie "Echo", dont la durée est proportionnelle à la distance mesurée.

Le fonctionnement du module est le suivant :

Il faut envoyer une impulsion niveau haut (à + 5v) pendant au moins 10  $\mu$ s sur la broche '**Trig Input**'; cela déclenche la mesure. En retour la sortie '**Output**' ou '**Echo**', va fournir une impulsion + 5v dont la durée est proportionnelle à la distance si le module détecte un objet. Afin de pouvoir calculer la distance en cm, on utilisera la formule suivante :

**distance = (durée de l'impulsion (en  $\mu$ s) / 58**

Voici une représentation graphique de la séquence de fonctionnement du module :



Après la théorie passons à la pratique; le câblage du module à l'Arduino sera le suivant :

– broche 12 de l'Arduino vers Trig.

– broche 11 de l'Arduino vers Echo.

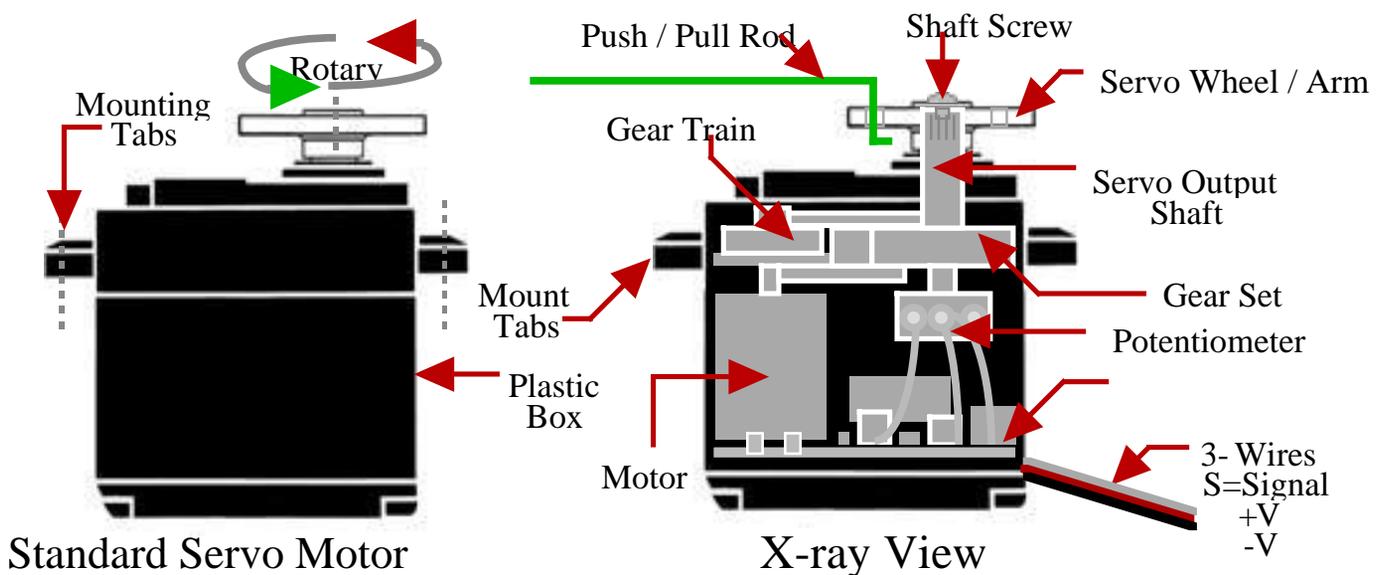
Le capteur HC-SR04 est intéressant. Pour un coût très bas, il donne des résultats étonnants de précision.

L'écart est d'environ 3 cm avec un objet placé à 2 m, ce qui représente une erreur inférieure à 2 %.

## Annexe C

### Servomotor Information

A servo is a mechanical motorized device that can be instructed to move the output shaft attached to a servo wheel or arm to a specified position. Inside the servo box is a DC motor mechanically linked to a position feedback potentiometer, gearbox, electronic feedback control loop circuitry and motor drive electronic circuit.



A typical R/C servo looks like a plastic rectangular box with a rotary shaft coming up and out the top of the box and three electrical wires out of the servo side to a plastic 3 pin connector. Attached to the output shaft out the top of the box is a servo wheel or Arm. These wheels or arms are usually a plastic part with holes in it for attaching push / pull rods, ball joints or other mechanical linkage devices to the servo. The three electrical connection wires out of the side are V- (Ground), V+ (Plus voltage) and S Control (Signal). The control S (Signal) wire receives Pulse Width Modulation (PWM) signals sent from an external controller and is converted by the servo on board circuitry to operate the servo.

A pulse width signal (PWM) of approximately 1.5 mS (1500 uS) is the "neutral" position for the servo. The servo, neutral is defined to be the point where the servomotor has exactly the

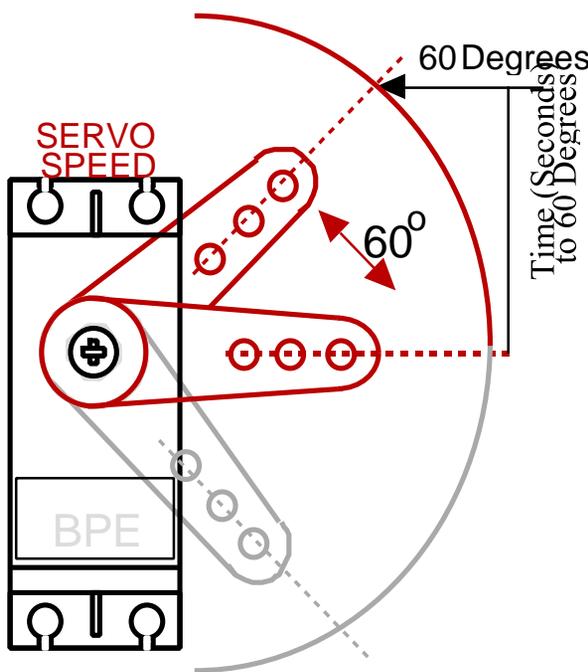
same amount of potential rotation in the counter clockwise direction as it does in the clockwise direction. When the pulse width signal (PWM) sent to a servo is less than 1.5 mS. the servo moves some number of degrees counterclockwise from the neutral point. When the pulse is greater than 1.5mS the servo moves some number of degrees clockwise from the neutral point. Generally the minimum pulse will be about 1.0 mS and the maximum pulse will be 2.0 ms with neutral ( Stop) movement at 1.5 mS

R/C servos run on 5 volts DC but they often work with voltages V-, V+ between 4 and 6 volts DC power, near 1 Amp of current. (Torque load on the servo arm determines amps and can be from 200 mA to 1 Amp depending on moving or holding force the servo needs for position)

### SERVO RATINGS

The most common details available on a servo are its speed and torque rating. Nearly all servo packages are listed with brand name, model name/ number, speed, and torque output at 4.8 volts

and 6.0 volts. Some information about metal, plastic gears or ball bearings may also be listed.



### SERVO SPEED

Servo Speed is defined as the amount of time ( in seconds) that a servo arm attached to the servo output shaft will move from 0 to 60 degrees.

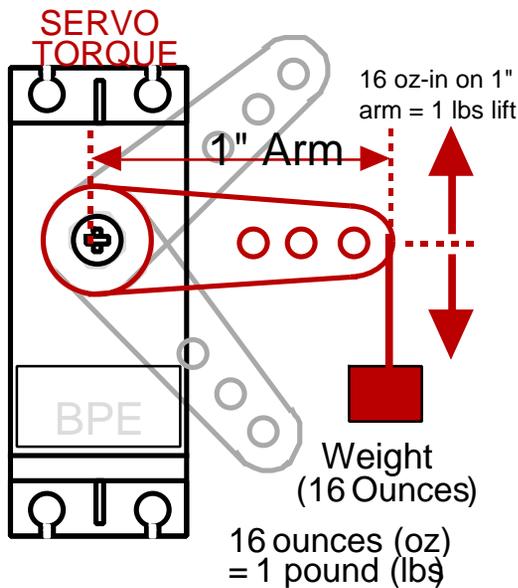
Note: The lower the time (Seconds) the faster the servo can move an attached wheel or arm.

Servo Speed is measured by the amount of time (in seconds) it takes a 1 inch servo arm to sweep left or right through a 60 degree arc at either 4.8 or 6.0 volts.

A servo rated at 0.22 seconds/60 degrees takes 0.22 seconds to sweep through a 60 degree arc. Some of the fastest servos available move in the 0.06 to 0.09 second range. In some servos, faster speeds may lower torque available.

### SERVO TORQUE (Power)

Servo Torque is defined as ounce-inch (oz-in) The total push / pull power a servo can apply on a 1" servo arm when moving.



Servos have a certain amount of torque (strength) that is generally proportional to their size. Servos come in all kinds of sizes, strengths and weight. Torque is the measurement of force given over a distance. For most servos in the USA, torque is measured in oz-in (force in ounces times inches, or ounce-inch). Servo Torque is measured by the amount of weight (in ounces) that a servo can hold at 1-inch out on the servo output arm in the horizontal plane, again at either 5.0 or 6.0 volts to see when the servo stalls as it tries to lift the weight horizontally. The reported result is a measurement like

this: Servo XYZ = 100 oz/in. @ 6.0 V. That means that Servo XYZ is capable of holding 100 ounces using a 1 inch output arm

without excessive deflection at 6.0 input volts. To convert oz-in to kilogram-centimeters (kg-cm) just divide by 13.9

**Examples:** Servo-A has 42 oz-in of torque

42 divided by 16 = 2.63 Lb-In. of force on a 1" servo arm

Servo-B has 2.5 oz-in of torque

2.5 divided by 2 = 1.25 Oz.-In. of force on a 2" servo arm

Servo-C has 36 oz-in of torque

36 divided by 4 = 9 Oz.-In. of force on a 4" servo arm

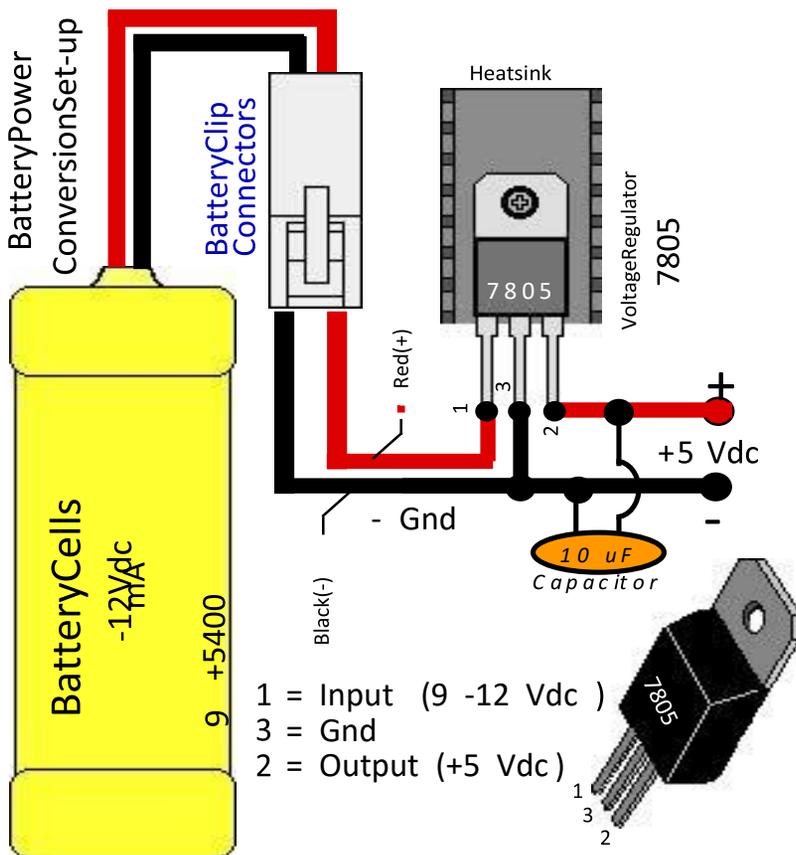
Servo Information



R/C Servo

Servo Connector from the power source

Servo operate from 4.5 to 6.0 volts DC. At the higher voltage servos tend to be faster and sometimes stronger, but can heat up faster when stalled or in a hold position with stress forces against the servo output shaft. Some servo controllers require a separate

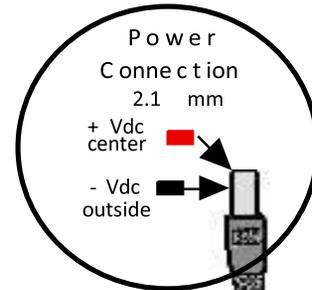


Vdc. The current drain (Amps required) depends on the torque being put out by the servo motor and can be in excess of one amp if the servo is stalled under load.

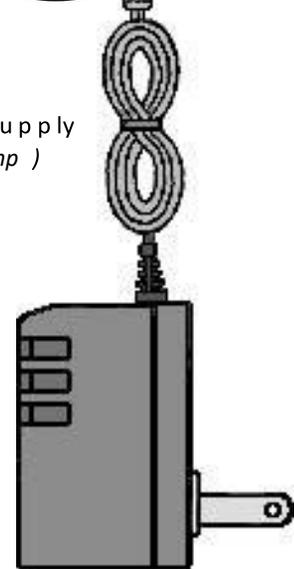
It is best to calculate 1 Amp per servo when figuring power supply needs for most servos.

(S) Signal =Yellow (PWM Signal)

(+) 5 Vdc = Red



Regulated Wall Power Supply (5 Vdc @ 2 Amp )

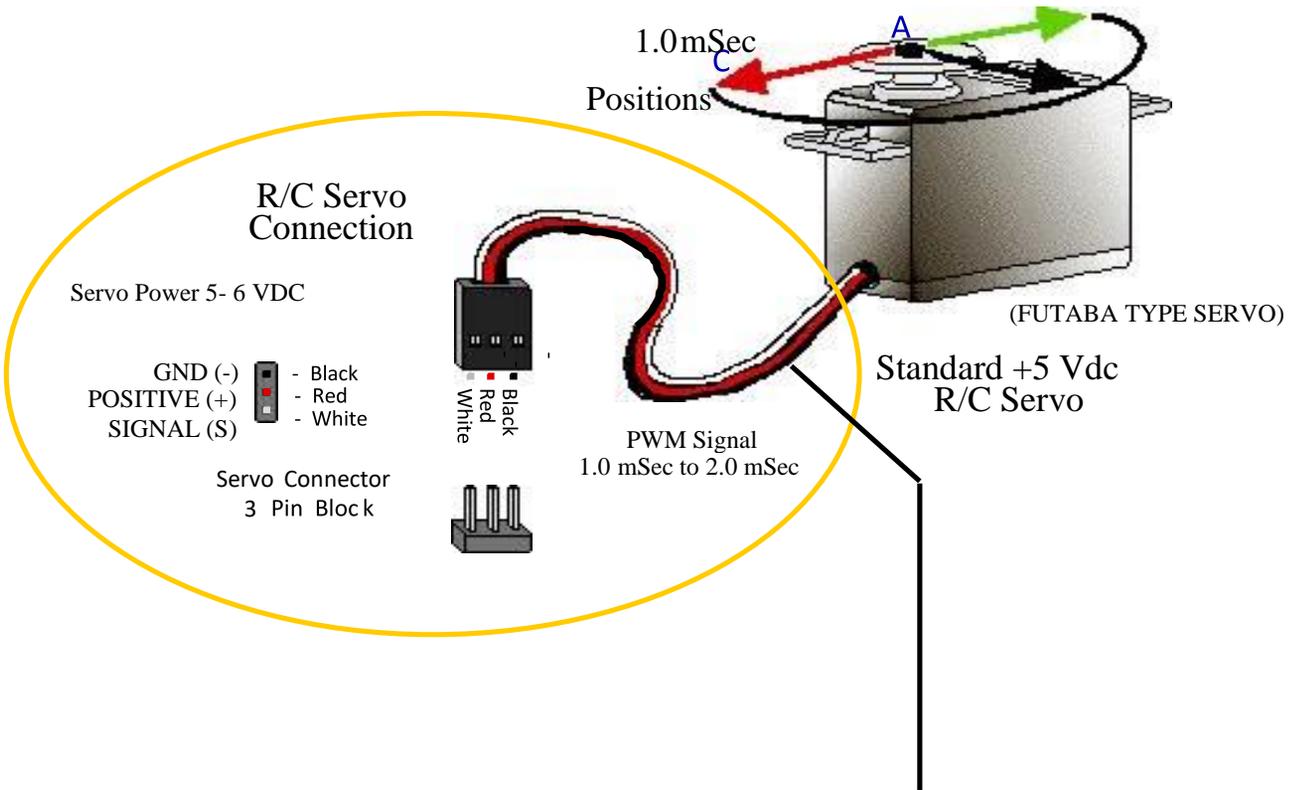


control source to deliver the higher 6.0

9-12 Vdc to +5 Vdc Converter	
7805	Voltage Regulator
10 uF	Capacitor
9-12 Vdc Supply	Battery Supply
Heatsink	

## Servo Wire Information

1.0 mSec to 2.0 mSec 1.5 mSec B

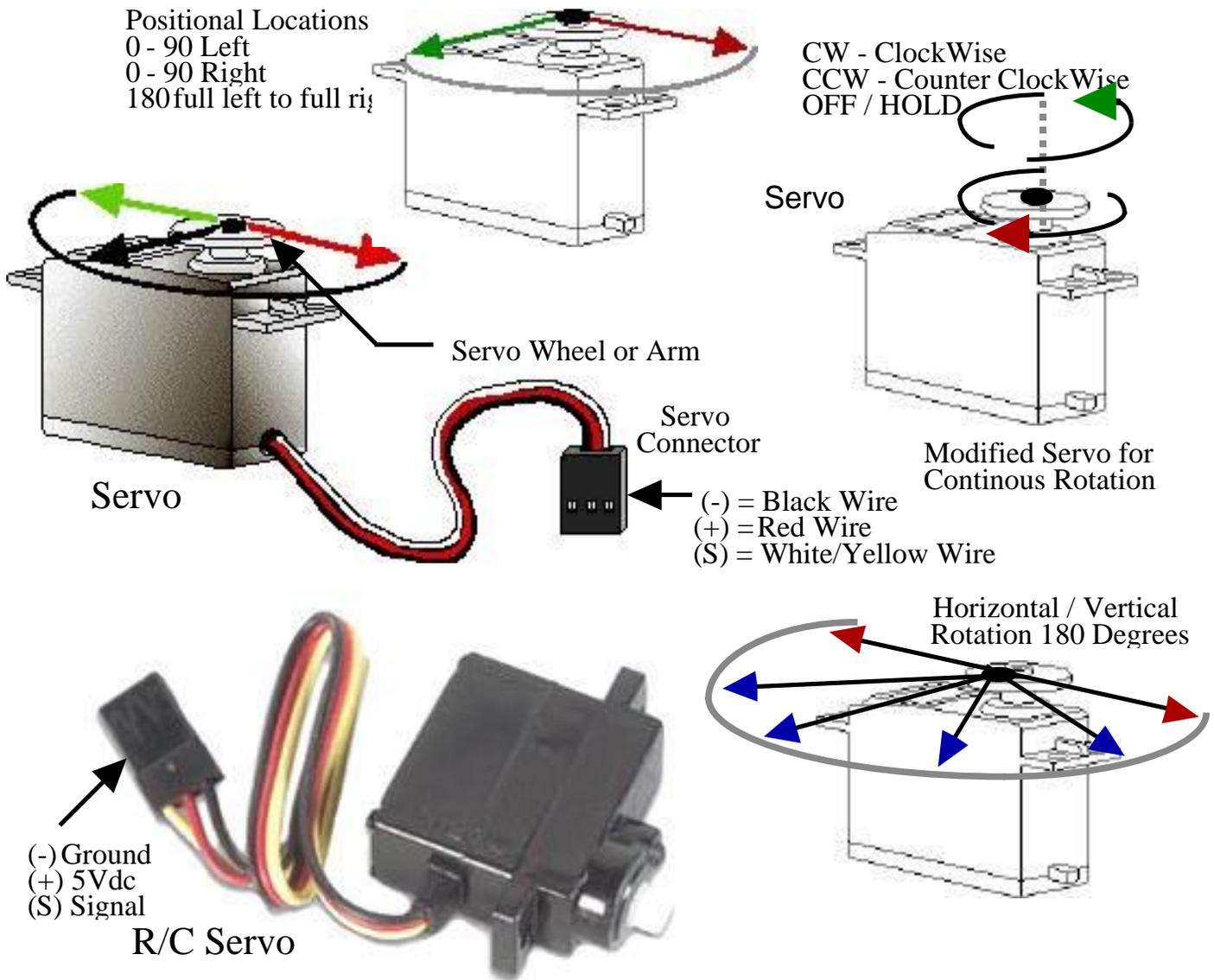


### Servo Wire Code

Servo Type	Positive (+)	Signal (S)	Negative (-)
Cirrus	Red	White	Black
Futaba - J	Red	White	Black
JR	Red	Orange	Brown
Hitec	Red	Yellow	Black
Airtronics	Red	Orange	Black
	Red	White	Black
	Red	Black	Black
Airtronics - Z	Red	Blue	Black
Fleet	Red	White	Black
GW	Red	Orange	Brown

## Servo Centering

R/C servos are usually mechanically stopped from moving at full rotation. They have limited rotation through a mechanical, plastic block on the internal gearing and can rotate about 90 to 180 degrees or less only. Servos are unable to continually rotate and usually can't be used for driving rotating wheels. A servos precision positioning makes them ideal for robotics and animatronics, since servos are self contained with control loop circuitry, drive circuits, servo position, speed control, and are very easy to control by an external device such as a electronic servo controller board used in animatronic character and robotic applications.



## Servo Info and Centering

The best and easiest way to set a servo's center position is to connect the servo to a servo checker or tester. There are several low cost servo checkers worth purchasing and having in your toolbox if you plan to work with servos. To use a Servo Checker simply set the Servo Checker to its automated center position and let the controller move the connected servo to center. If the servo arm is pre-attached, you may have to remove the servo arm on top of the servo shaft and re-position it back onto the servo shaft center point. The servo is now ready for installation. Once centered place the servo arm on top of the servo shaft, secure it with servo screw and install the servo as needed or carefully remove the servo arm from the servo if not needed for installation. Do not move the shaft at this point, if it moves, simply repeat the above procedure to find the servo center again.

### Manual Servo Center Position - Steps

Carefully place one of the servo arms or wheel onto the servo shaft, mark a reference line through (across) the servo arm or wheel center point.

Slowly and carefully rotate the servo arm or wheel by hand as far to one side as it will go, do not force servo arm / wheel.

Mark a line on the servo base where the servo arm or wheel reference line stops.

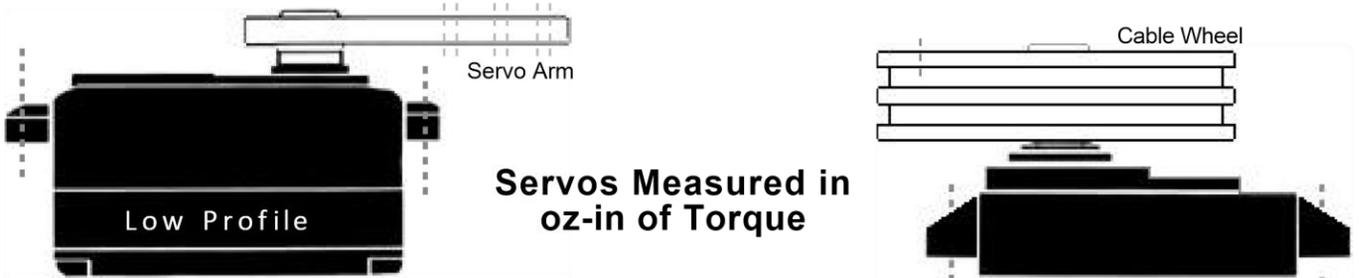
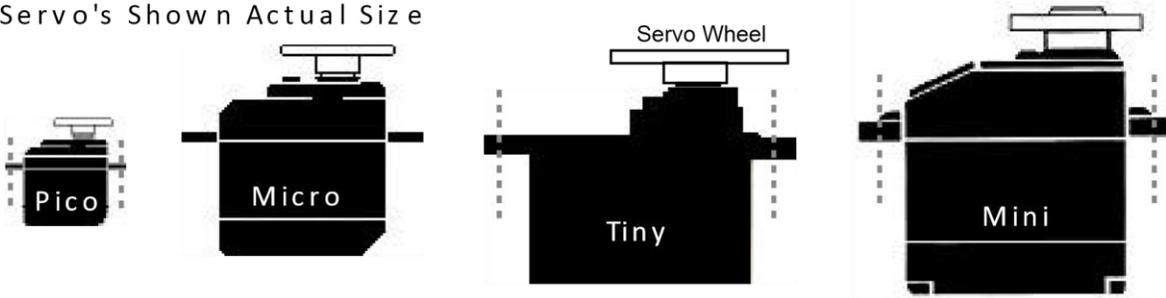
Slowly and carefully rotate the servo arm or wheel by hand to the opposite position, as far to one side as it will go, do not force servo arm / wheel. Should travel about 180 degrees. Mark a line on the servo base where the servo arm reference line stops.

Rotate the servo arm back to 90 degrees between the two end reference line positions (A,B) marked on the servo base. You may have to remove the servo arm on top of the servo shaft and re-position it back onto the output shaft to get the center point if the arm is off center position. This should put the servo arm close to center position.

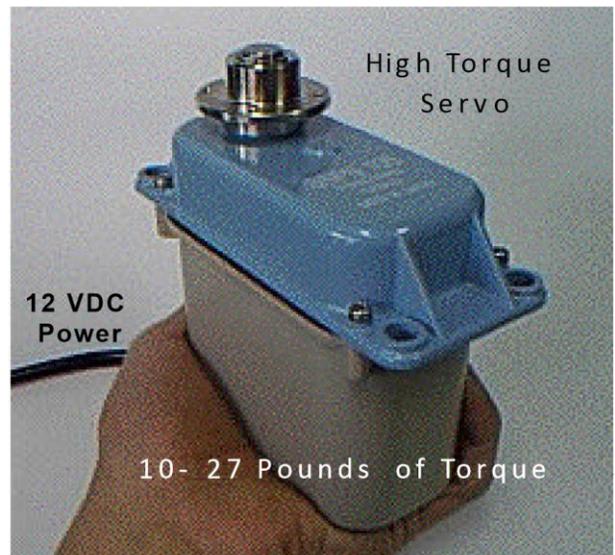
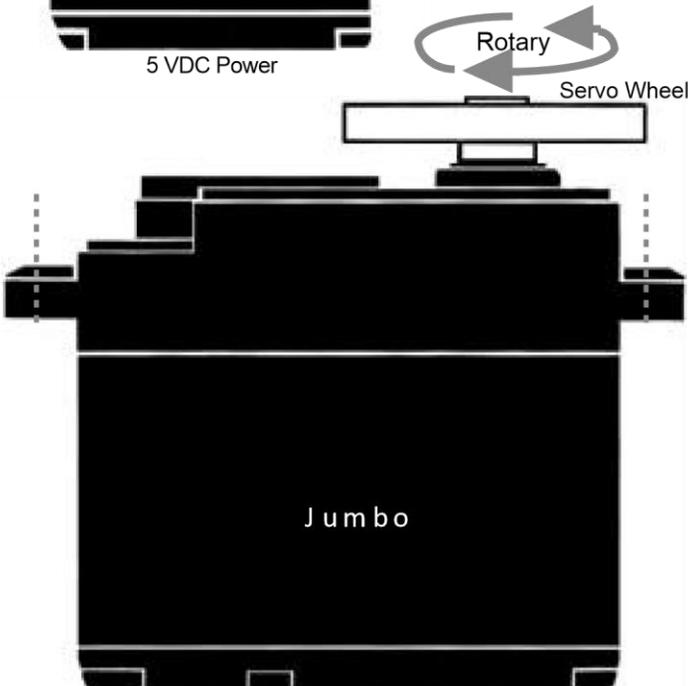
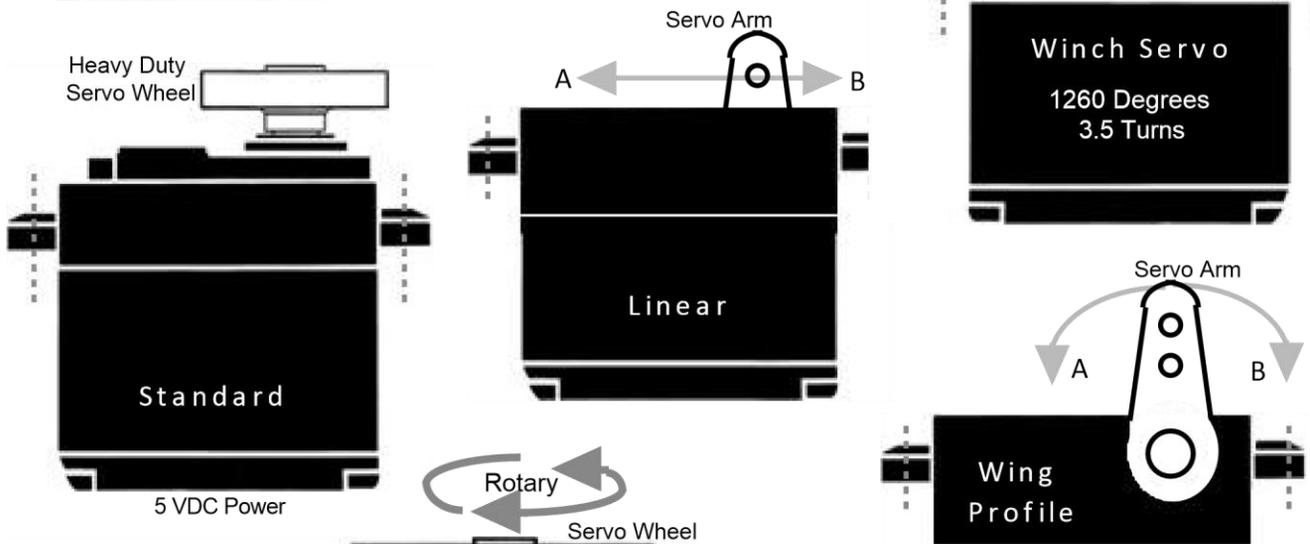
Carefully remove the servo arm from the servo if not needed for installation or place the servo arms on top of the servo shaft,

# R/C Servo Sizes / Types

Servo's Shown Actual Size

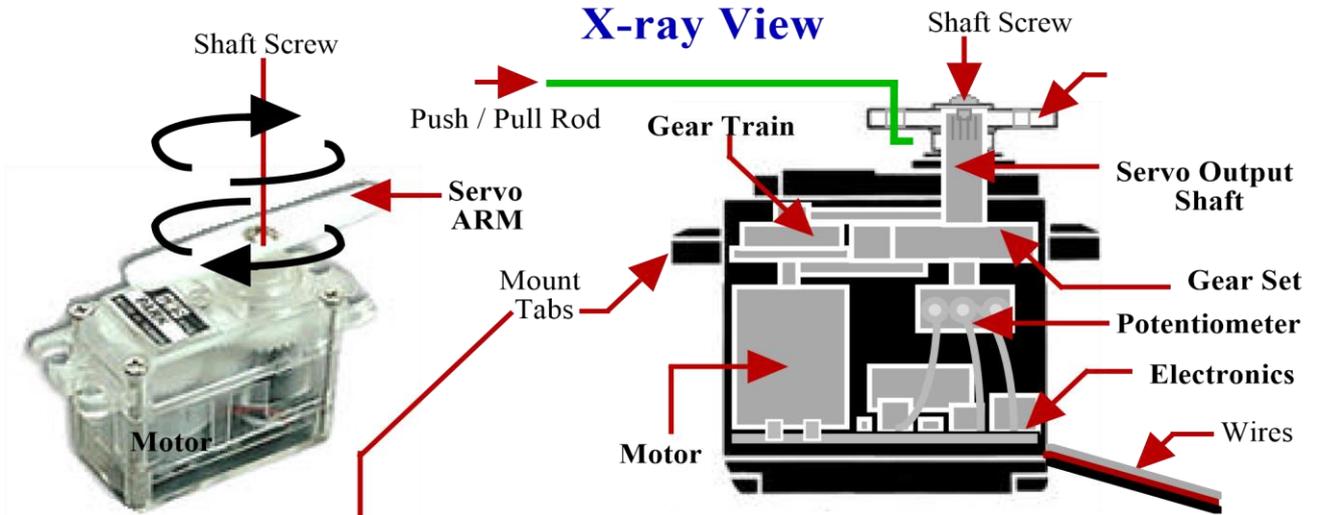


Servos Measured in oz-in of Torque



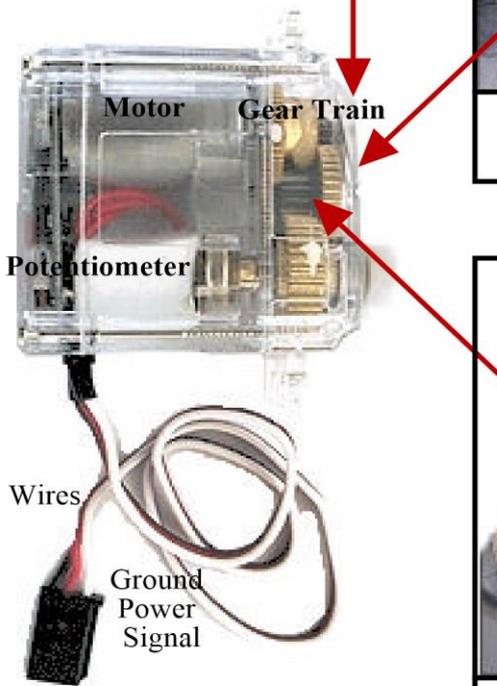
**Servomotor Information**

Ver 2.5



Note: A Servo stability and longevity is determined by the quality of the electronics, mechanical parts. Any servo under stress in pushing or pulling a mechanical device will fail eventually. Never let a servo Hum, shake, or push / pull beyond its limits.

**Servo Components: Best -vs- Good Parts**



## Bibliographie

---

- [1] B. Atrouz, 'Les Systèmes Radar', Ecole Militaire Polytechnique.
- [2] Mr Benachenhou, 'Cours Radar', Université De Blida, 5<sup>ème</sup> Année Ingénieur, Département Aéronautique, Année: 2014/2015.
- [3] C.Fougiyas and C.Menychtas, 'Netted LPI radars', thesis naval postgraduate school, Montrey California, September 2011.
- [4] Latifa Abdou "Application Des Algorithmes Évolutionnaires À L'optimisation Du Seuil De Détection Dans Les Systèmes Distribués Cfar",Thèse De Doctorat En Science En Électronique,Université De Batna, 08 Avril 2009.
- [5] Souâd Chabbi, "Detection Adaptative Cfar A Censure Automatique Basse Sur Les Statistiques D'ordre En Milieux Non Gaussiens" Université De Constantine, Département D'électronique, Promotion 2008.
- [6] N.Hamdad, 'Transformée de Huang-Hilbert appliquée à la détection des défauts', mémoire de magister, université Mouloud Mammeri Tiziouzou, mars 2013.
- [7] Mr S.Meguelati,"Cours Radar",Université De Blida,5<sup>ème</sup> Année Ingénieur,Département Aéronautique(Option:Installation),Année:2009/2010.
- [8] Mr Gilles MAURIS ,CAPTEURS ULTRASONORES "INTELLIGENTS" Application à la représentation symbolique de mesures de distance par codage flou
- [9] Ieee standard glossary of software engineering terminology. *IEEE Std 610.12-1990*, pages 1–84, 1990. (Citée en page 4.)
- [10] R. Abreu, P. Zoetewij, and A.J.C. van Gemund. On the accuracy of spectrum-based fault localization. In *Testing : Academic and Industrial Conference Practice and Research Techniques - MUTATION. TAICPART-*

- MUTATION 2007*, pages 89 –98, sept. 2007. (Citée en pages [viii](#), [9](#), [80](#), [83](#), [84](#), [89](#), [104](#) et [110](#).)
- [11] Rui Abreu, Peter Zoeteweyj, and Arjan J. C. van Gemund. An evaluation of similarity coefficients for software fault localization. In *Proceedings of the 12th Pacific Rim International Symposium on Dependable Computing*, PRDC '06, pages 39–46, Washington, DC, USA, 2006. IEEE Computer Society. (Citée en pages [80](#) et [82](#).)
- [12] Rui Abreu, Peter Zoeteweyj, and Arjan J. C. van Gemund. Spectrumbased multiple fault localization. In *Proceedings of the 2009 IEEE/ACM International Conference on Automated Software Engineering*, ASE '09, pages 88–99, Washington, DC, USA, 2009. IEEE Computer Society. (Citée en pages [88](#) et [153](#).)
- [13] Rui Abreu, Peter Zoeteweyj, Rob Golsteijn, and Arjan J. C. van Gemund. A practical evaluation of spectrum-based fault localization. *Journal of Systems and Software*, June 2009. (Citée en pages [88](#) et [153](#).)
- [14] Mr TIMIN J-Louis, Cours Info Indus n°3a : Fonctionnement d'un microprocesseur, Année 2009-2010
- [15] Mr Hernando Barrag , pour sa thèse de fin etudes , avait entrepris le développement d'une cart Arduino, année 2005
- [16] Mr Jean-Noël Montagné ,Centre de Ressources Art Sensitif, novembre 2006
- [17] Mr Simon Landrault et Hippolyte Weisslinger, Arduino : Premiers pas en informatique embarquée , Édition du 01 juin 2014.

### **Site internet :**

<http://perso.wanadoo.fr/grc/acqueririnfolog04.htm>

<http://www.sinaptec.fr/>

[www.datasheetcatalog.com](http://www.datasheetcatalog.com)

[www.l'electronique.com](http://www.l'electronique.com)

<http://www.arduino.cc/en/Main/Donate>

<http://fr.slideshare.net/housseml10901/robot-base-dandroid-prsentation-pfe>