



Université Mohamed Khider de Biskra
Faculté des Sciences et de la Technologie
Département de Génie Electrique

MÉMOIRE DE MASTER

Sciences et Technologies
Filière: Télécommunication
Spécialité :Réssaux et télécommunication

Réf. :

Présenté et soutenu par :
Ratmia Mohamed Najib et Dahah Mohamed Charef Eddine

Le : dimanche 26 juin 2022

Systeme radar ultrasonnic à base d'Arduino pour la détection d'objets

Jury :

M.	ATAMNA Noura	MAA	Université de Biskra	Président
M.	AMIED Sofiane	MAA	Université de Biskra	Examineur
M.	TERGHINI Ouarda	MCA	Université de Biskra	Rapporteur

Année universitaire : 2021 - 2022



Université Mohamed Khider de Biskra
Faculté des Sciences et de la Technologie
Département de génie électrique

MÉMOIRE DE MASTER

Sciences et Technologies
Filière: Télécommunicatio
Spécialité :Réssaux et télécommunication

Réf. :

Présenté et soutenu par :
Ratmia Mohamed Najib et Dahah Mohamed Charef Eddine

Le : dimanche 26 juin 2022

Systeme radar ultrasonnic à base d'Arduino pour la détection d'objets

Jury :

M.	ATAMNA Nouara	MMA	Université de Biskra	Président
M.	AMIED Sofiane	MAA	Université de Biskra	Examineur
M.	TERGHINI Ouarda	MCA	Université de Biskra	Rapporteur

Année universitaire : 2021-2022



Université Mohamed Khider de Biskra
Faculté des Sciences et de la Technologie
Département de génie électrique

MÉMOIRE DE MASTER

Sciences et Technologies
Filière: Télécommunication
Spécialité :Réseaux et télécommunication

Réf. :

Système radar ultrasonique à base d'Arduino pour la détection d'objets

Le : dimanche 26 juin 2022

Présenté par : Ratmia Mohamed najib
Dahah Mohamed Charef Eddine

Avis favorable de l'encadreur:

Signature Avis favorable du Président du Jury

Cachet et signature

Remerciment

Nous tenons à remercier premièrement « ALLAH », le tout puissant pour la volonté, aussi nos parents qu'ils nous ont donnés le soutien durant toutes ces longues années. Ainsi, nous tenons également à exprimer nos vifs remerciements à notre encadreur Dr. Terghini Ouarda pour le suivi continuel tout au long de la réalisation de ce mémoire et qui n'a pas cessé de nous donner ses conseils. Nous tenons à remercier également les membres du jury qui ont accepté de juger notre travail. Nous tenons à remercier vivement toutes les personnes qui nous ont aidés à élaborer et réaliser ce mémoire, ainsi à tous ceux qui nous ont aidés de près ou de loin à accomplir ce travail. En fin, nous tenons à exprimer notre reconnaissance à tous nos amis et collègues pour leur soutien moral.

Résumé

Résumé

Ce projet a pour but de conception réalisation d'un radar a ultrason basé sur une carte Arduino capable de détecter des objets fixes et mobiles, le travail consiste à mener une étude générale sur la théorie des radars, sur les cartes électroniques (Arduino) d'une part et d'autre part sur le langage de programmation C et le traitement de l'affichage, à la fin de ce travail, nous avons réalisé une simulation du projet Radar à l'aide du logiciel Proteus, et nous avons également testé le dispositif RADAR sur des cibles réelles.

Mots clés : Radar, Arduino, détecteur, ultrason, servomoteurs, LCD, détection d'objet, prototypage.

Abstract

The purpose of this project is to design and build an ultrasonic radar based on an Arduino card capable of detecting fixed and mobile objects, the work consists of carrying out a general study on the theory of radars, on electronic cards (Arduino) on the one hand and on the other hand on the C programming language and the display processing, at the end of this work, we carried out a simulation of the Radar project using the Proteus software, and we also tested the RADAR device on real targets.

Keywords: Radar, Arduino, detector, ultrasound, servo motors, LCD, object detection, prototyping.

ملخص

الغرض من هذا المشروع هو تصميم وبناء رادار فوق صوتي يعتمد على بطاقة Arduino قادرة على اكتشاف الأجسام الثابتة والمتحركة ، ويتكون العمل من إجراء دراسة عامة حول نظرية الرادارات و البطاقات الإلكترونية (Arduino) من جهة ومن جهة أخرى لغة البرمجة C ومعالجة العرض ، في نهاية هذا العمل ، قمنا بعمل محاكاة لمشروع الرادار باستخدام برنامج Proteus ، كما اختبرنا جهاز RADAR على أهداف حقيقية .

الكلمات المفتاحية : الرادار ، الأردوينو ، الكاشف ، الموجات فوق الصوتية ، المحركات المؤازرة ، كشف الأشياء ، النماذج الأولية.

SOMMAIRE

Introduction générale	1
Chapitre 1 : Généralités sur les radars	
1.1-Introduction.....	4
1.2- Généralités sur les radars.....	4
1.2.1- Historique du radar.....	4
1.2.2-Radar.....	5
1.2.3- la cible.....	5
1.3- Principe de fonctionnement.....	5
1.3.1- Mesure de distance.....	6
1.3.2- Mesure de l'angle.....	7
1.3.3- Mesure de la vitesse radiale.....	7
1.3.4- L'effet Doppler.....	8
1.4- Radar ultrasons.....	8
1.4.1- Détecteurs à ultrason.....	8
1.4.2- Les caractéristiques ultrasonores	9
1.4.3- Pour quoi le choix des capteurs à ultrason.....	10
1.5- Classification des Radars selon la technologie.....	10
1.5.1- Le Radar primaire.....	11
1.5.2- Radar à impulsions.....	11
1.5.2.1- Radar à MTI (Moving Target Indicator/ Indicateur de la cible mobile).....	12
1.5.2.2- Radars Doppler à impulsions.....	12
1.5.3- Radar à ondes continues.....	12
1.5.3.1- Radar à ondes continues non modulées.....	12

SOMMAIRE

1.5.3.2-Radar à ondes continues modulées.....	12
1.5.4- Radar secondaire.....	13
1.5.5- Radar à impulsions.....	13
1.5.5.1- Description.....	13
1.5.5.2- les Composons d'un Radar à impulsion.....	14
1.6- Equation Du Radar.....	15
1.6.1- Argumentation Et Explication.....	15
1.6.2- Influence des pertes sur la portée.....	18
1.6.3- Portée maximale du Radar en présence du bruit thermique.....	18
1.6.3.1- Filtrage non optimal.....	20
1.6.3.2- Filtrage optimal (Filtre adapté).....	21
1.7- Radar à diversité de fréquences.....	21
1.8- Applications du Radar.....	22
1.9- Conclusion.....	23

Chapitre 2 : Matériel et composants

2 .1- Introduction.....	25
2.2- Etude de matériel	25
2.2.1- Matériel utilisé	25
2.2.2- Définition de module Arduino.....	25
2.2.3- La Carte Arduino Uno.....	26
2.2.3.1- Les caractéristiques de la carte Arduino Uno.....	27
2.2.3.2- les composants de la carte Arduino.....	27
2.2.3.3- Le Principe de fonctionnement d'Arduino.....	31
2.2.3.4- Les avantages de l'arduino uno.....	32

SOMMAIRE

2.2.4- Capteur Ultrasons HC-SR04.....	32
2.2.4.1- Caractéristiques et spécification du capteur.....	33
2.2.4.2- Broches de connexion.....	33
2.2.4.3- Fonctionnement du capteur Ultrasons HC-SR04.....	34
2.2.5- Module Afficheur LCD.....	35
2.2.5.1- Connecteur de l’afficheur LCD.....	36
2.2.5.2- Communication avec l’afficheur LCD.....	38
2.2.6- Servomoteur.....	38
2.2.6.1- les caractéristiques du servomoteur.....	39
2.2.6.2- Connecteur du servomoteur.....	40
2.2.6.3- Fonctionnement du servomoteur.....	40
2.3- Etude de Logiciels de programmation.....	42
2.3.1- Interface de programmation Arduino.....	42
2.3.2- Logiciel IDE de PROCESSING.....	47
2.3.3- Logiciel PROTEUS ISIS.....	48
2.3.4- Communication entre la carte Arduino, logiciel Proteus et logiciel Processing.....	50
2.3.5- Schéma d’interconnexion entre les composants.....	51
2.4- conclusion.....	51

Chapitre 3 : Simulation Et Réalisation Du Projet

3.1- Introduction.....	53
3.2- Principe de fonctionnement.....	53
3.3- Les étapes du projet.....	53

SOMMAIRE

3.4- schéma synoptique du radar.....	54
3.5- Algorithme principale de détection d'un objet.....	55
3.6- Simulation du projet sous PROTEUS.....	55
3.6.1- Bibliothèque Arduino pour Proteus.....	55
3.6.2- Bibliothèque de capteurs à ultrasons pour PROTEUS.....	56
3.6.3- Circuit global de la simulation.....	58
3.6.4- Résultats de la simulation.....	59
3.7- Réalisation de projet.....	61
3.7.1- Composants utilisés.....	61
3.7.2- Les différentes connexions et numéro des pins du circuit.....	62
3.7.3- Les étape de la réalisation.....	63
3.7.3.1- Alimentation du circuit.....	63
3.7.3.2- Test du fonctionnement de la carte Arduino.....	63
3.7.3.3- Dessiner un radar sur processing.....	64
3.8- Résultats de la réalisation.....	66
3.8.1- Visualisation de la distance et l'angle et la vitesse(speed) sur le moniteur série.....	66
3.8.2- Résultats des différents objets détectés.....	67
3.8.3- Interprétation des résultats.....	70
3.9- Conclusion.....	70
Conclusion générale.....	72

Liste des figures

Figure 1.1 : Principe d'un système radar.....	6
Figure 1.2 : Mesure de distance.....	6
Figure 1.3 : Mesure de l'angle.....	7
Figure 1.4 : Principe de fonctionnement du détecteur à ultrasonore	9
Figure 1.5 : Classification des Radars	11
Figure 1.6 : Mesure de distance avec un Radar FMCW.....	13
Figure 1.7 : Génération d'échos.....	14
Figure 1.8 : Diagramme de bloc d'un Radar à impulsion.....	14
Figure 1.9 : Relation entre les puissances émises et réfléchies.....	17
Figure 1.10 : Différents types de bruit traités par un récepteur.....	19
Figure 1.11 : Spectre du signal impulsionnel et du bruit « Blanc ».....	20
Figure 1.12 : Variation du SNR en fonction de largeur de bande.....	21
Figure 2.1 : Carte Arduino.....	26
Figure 2.2 : Schéma de carte arduino uno.....	26
Figure 2.3 : Microcontrôleur ATmega328.....	28
Figure 2.4 : Capteur Ultrasons HC-SR04.....	32
Figure 2.5 : Broches de connexion de capteur a ultrason avec arduino.....	33
Figure 2.6 : Principe de fonctionnement de HC-SR04.....	34
Figure 2.7 : Le diagramme temporel.....	35
Figure 2.9 : Afficheurs LCD (a) (16x2) (b) (20x4).....	36
Figure 2.9 : Connecteur de l'afficheur LCD.....	37

Liste des figures

Figure 2.10 : Servomoteur.....	39
Figure 2.11 : Connecteur de servomoteur	40
Figure 2.12 : impulsions de commande d'un servomoteur.....	41
Figure 2.13 : Montage servomoteur avec arduino.....	41
Figure 2.14 : Icône de programme arduino.....	42
Figure 2.15 : Interface de programme arduino.....	43
Figure 2.16 : IDE arduino	43
Figure 2.17 : Barre de bouton arduino.....	44
Figure 2.18 : Sélection de la carte arduino	45
Figure 2.19 : Sélection de porte	45
Figure 2.20 : Le processus de copier du fichier HEX.....	46
Figure 2.21 : Hyper terminale de l'arduino (moniteur série).....	46
Figure 2.22 : Icône de logiciel processing.....	47
Figure 2.23 : L'interface de logiciel processing.....	48
Figure 2.24 : Icône de programme proteus.....	48
Figure 2.25 : L'interface principale de programme proteus.....	49
Figure 2.26 : Interface de sous programme proteus ISIS.....	49
Figure 2.27 : Communication entre la carte arduino et processing.....	50
Figure 2.28 : Communication entre proteus et processing.....	51
Figure 2.29 : Schéma d'interconnexion entre les composants du projet.....	51
Figure 3.1 : Schéma synoptique de notre projet (radar).....	54
Figure 3.2 : Algorithme principale de détection d'un objet	55
Figure 3.3 : L'intégration de la carte arduino sous proteus.....	56

Liste des figures

Figure 3.4 : Capture HC-SR04 sous proteus.....	57
Figure 3.5 : Fichier ultrasonic TEB.HEX sous proteus.....	57
Figure 3.6 : Circuit électronique globale de simulation.....	58
Figure 3.7 : Résultat de simulation sur le terminale virtuel.....	59
Figure 3.8 : Signal echo et signal trigger de l'ultrason.....	61
Figure 3.9 : Le circuit globale de notre projet de réalisation radar.....	61
Figure 3.10 : Présentation final de notre projet de réalisation de radar de détection.....	64
Figure 3.11 : Dessin de l'arc et les lignes de radar.....	64
Figure 3.12 : L'objet détecté par la fonction drawObject().....	65
Figure 3.13 : écriture d'un texte à l'aide de la fonction drawText().....	65
Figure 3.14 : L'affichage de la simulation complète sur processing.....	66
Figure 3.15 : Visualisation des résultat sur le moniteur série.....	67
Figure 3.16 : la mise en marche de notre radar.....	67
Figure 3.17 : Exemple pratique de détection d'un objet fixe	68
Figure 3.18 : L'affichage de résultats de détection d'objet sur l'interface processing.....	69
Figure 3.19 : L'affichage de la distance et la vitesse (speed) de l'objet détecté sur .LCD.....	69

Liste des tableaux

Tableau 2.1 : Caractéristique de la carte arduino uno.....	27
Tableau 2.2 : Les caractéristiques de capteurs ultrason HC-SR04.....	33
Tableau 2.3 : Nomenclature de connecteur de l’afficheur LCD.....	37
Tableau 2.4 : Les caractéristiques de servomoter SG-90.....	39
Tableau 3.1 : Broches et connexion de notre circuit électronique	58
Tableau 3.2 : Résultats de détections obtenues de la simulation.....	60
Tableau 3.3 : Broches et connexion de circuit de la pratique	62
Tableau 3.4 : Le résultat des différents objets détecte.....	68

Introduction
Générale

Introduction Générale

Le radar est l'une des technologies d'observation de la Terre les plus avancées, et l'exploration planétaire. Le radar météorologique a été introduit après la Seconde Guerre mondiale. Aujourd'hui, les systèmes radar sont également largement utilisés dans Les applications civiles telles que la surveillance du trafic aérien, la météorologie et même la sécurité automobile. Par conséquent, le radar est un système essentiel et très largement utilisé aujourd'hui, pour surveiller le territoire ou éviter les collisions.

Le mot radar est l'abréviation anglaise << Radio Detection And Ranging >>, c'est un Instruments de télédétection actifs qui génèrent des impulsions d'ondes électromagnétiques Étant donné une fréquence, puis une application d'imagerie pour détecter et déterminer la distance, la vitesse ou les deux objets (par exemple, avions, bateaux, etc.). Il est utilisé dans de nombreux domaines tels que l'observation des nuages, la mesure de la vitesse du vent et la détection d'intrusion militaire.

Le radar se compose d'un émetteur et d'un récepteur. L'émetteur envoie des ondes radio réfléchies par une cible et détectées par le récepteur du radar. La position de l'objet est calculée à l'aide du temps aller-retour du signal reçu, la vitesse est mesurée selon le principe de l'effet Doppler-Fizo, qui consiste à détecter les changements de fréquence des cibles en mouvement (fréquences plus élevées pour les cibles plus proches du radar, les fréquences plus basses pour les cibles plus éloignées). En mesurant la différence de fréquence entre les ondes transmises et renvoyées, la vitesse de la cible peut être calculée.

Dans ce mémoire nous présentons la conception et la réalisation d'un radar de détection ultrason composé d'une carte à microcontrôleur Arduino Uno pour le traitement des informations reçues, d'un capteur de distance Ultrason , permettant l'acquisition des données et d'un servomoteur dont le rôle est de permettre la rotation du capteur de distance sur la zone de détection.

Le mémoire est structuré en trois chapitres plus une introduction et une conclusion générale.

- Le premier chapitre est une introduction et généralités sur les radars.
- Le deuxième chapitre présente les différents outils matériels et logiciels nécessaires à la conception du radar.

Introduction Générale

- Le troisième chapitre est consacré à la conception et la réalisation du radar et les différents tests effectués sur celui-ci.

Enfin nous allons terminer par une conclusion qui résume le travail réalisé et quelques perspectives envisagées.

Chapitre 1
Généralités sur les radars

1.1-Introduction

Ce chapitre comporte une présentation théorique des radars de détections des objets ainsi que ses conceptions et ses principes de fonctionnement et finalement présentation des différents types et classification de ces instruments.

1.2- Généralités sur les radars

1.2.1-Historique du radar

Un jour de 1884 "James Clerk Maxwell" décrit les lois de l'électromagnétisme sous forme d'équations mathématiques. Deux ans plus tard (en 1886), le physicien allemande " Heinrich Rudolf Hertz " a réalisé les premières expériences sur les ondes électromagnétiques et a montré que les ondes "Radio" pouvait être réfléchies par les corps métalliques et diélectrique, et ainsi Hertz a ouvert la voie à l'invention du radar. En effet, l'histoire du radar était lancée dès 1904 où l'allemand " Christian Hülsmeier " dépose un brevet sur un détecteur d'obstacles à ondes radio-continues à la suite de réflexions constatées sur des navires croisant sur le Rhin.

Les expériences de détection avec des antennes radio se multiplient dans les années 1920 et 1930. En 1934, la France dépose un brevet sur un système de détection par ondes courtes : les premiers radars à ondes décimétriques équipent le cargo Oregon (en 1934) puis le paquebot Normandie (en 1935). L'évolution de ce domaine a permis d'utiliser le premier réseau de radars par "Robert Wattson" .

En 1939, l'invention de la cavité résonnante, dite magnétron, permet de développer le radar à ondes centimétriques et millimétriques. À la veille de la Seconde Guerre mondiale, la Grande-Bretagne possède déjà un réseau de radars pour protéger ses côtes des attaques navales et aériennes. Durant ce conflit, d'autres nations belligérantes développent à leur tour des systèmes radar qui, après la guerre, trouvent d'innombrables applications civiles et scientifiques[1].

L'expérience acquise par les opérateurs des radars pendant la guerre a également permis le développement de nouvelles applications.

1.2.2-Radar

Le radar (acronyme issu de l'anglais radio détection and ranging) est un système qui utilise généralement les ondes électromagnétiques pour détecter la présence et la position ainsi que la vitesse d'objets tels que les avions et les bateaux. Les ondes envoyées par l'émetteur sont réfléchies par la cible, et les signaux de retour (appelés 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[2].

1.2.3- La cible

La cible est tout ce qui interfère avec les ondes émises et réfléchit une partie de l'énergie vers le radar, se comportant comme une antenne de forme complexe. L'énergie émise en direction du radar est très fluctuante et dépend largement de l'orientation de la cible par rapport au radar.

1.3- Principe de fonctionnement

Le radar utilise un principe similaire à la réflexion des ondes sonores. Lorsque vous criez en direction d'un objet qui reflète votre voix, comme dans un canyon ou une grotte, vous entendrez un écho. Si vous connaissez la vitesse du son dans l'air, vous pouvez estimer la distance et la direction approximative des objets. Le temps nécessaire à l'aller et au retour du son vers vous peut être converti en distance si vous connaissez sa vitesse.

L'appareil radar détecte ce retour et en déduit la présence d'un obstacle, c'est la fonction Détection de Radio Detection And Ranging.

Comme l'appareil radar a été construit de façon à savoir quand il a envoyé l'OEM et quand il a détecté le retour de l'OEM réfléchi, il connaît la durée séparant ces deux moments ; il peut alors calculer la distance entre lui et l'obstacle détecté en se basant sur la vitesse connue de l'OEM (300.000 km/s environ), c'est la fonction Ranging (télémétrie) de RAdar Detection And Ranging , la figure 1.1 présente le principe de système radar [2] .

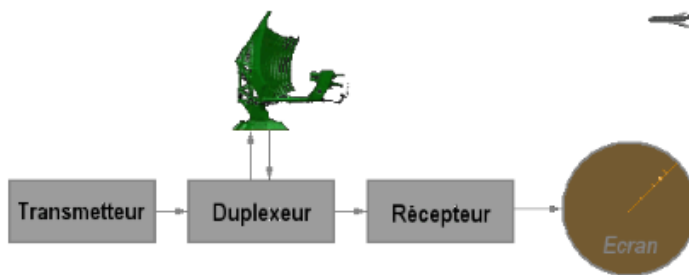


Figure 1.1 : Principe d'un système radar [2] .

1.3.1- Mesure de distance

L'onde émise par le radar parcourt la distance radar/cible R_0 à la célérité de la lumière C . Elle est rétrodiffusée par la cible dans tout l'espace. L'onde rétrodiffusée parcourt une nouvelle fois la distance R_0 comme il est illustré dans la figure 1.2 et une partie de l'onde est recueillie par le radar. La détection de l'écho reçu et la mesure du temps de propagation de l'onde nous informe sur la présence et la distance de la cible. L'onde reçue par le radar est une version atténuée et retardée de l'onde émise d'un retard [3,4].

$$\tau = \frac{R_0}{c} \quad (1)$$

Avec $C=3.10^8 m/s$

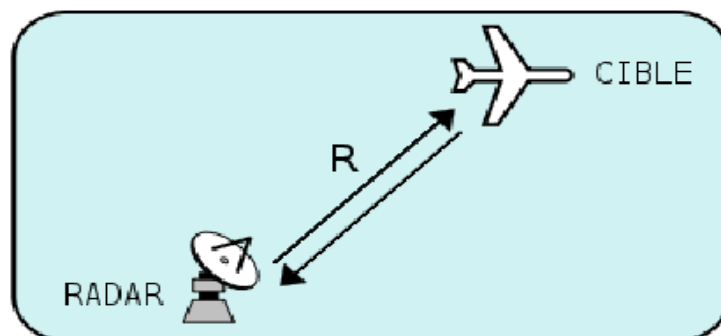


Figure 1.2 : Mesure de distance[2]

1.3.2- Mesure de l'angle

Les radars d'altimétrie rayonnent un faisceau (ou lobe) très étroit dans le plan vertical. Ce faisceau est déplacé suivant un axe vertical, mécaniquement ou électroniquement, afin de balayer en site son volume de détection et d'y positionner les cibles. Les radars d'altimétrie ayant également la capacité de déterminer l'azimut des cibles doivent utiliser un faisceau étroit tant dans le plan vertical que dans le plan horizontal.

L'Angle de site est l'angle que forme, avec sa projection sur le plan horizontal, l'axe „antenne radar - cible“, mesuré dans le plan vertical. La lettre grecque Epsilon (ϵ) désigne l'angle de site . L'angle de site est positif lorsqu'il est au-dessus de l'horizon (site 0°), et négatif en dessous, Comme le montre la figure 1.3 [5].

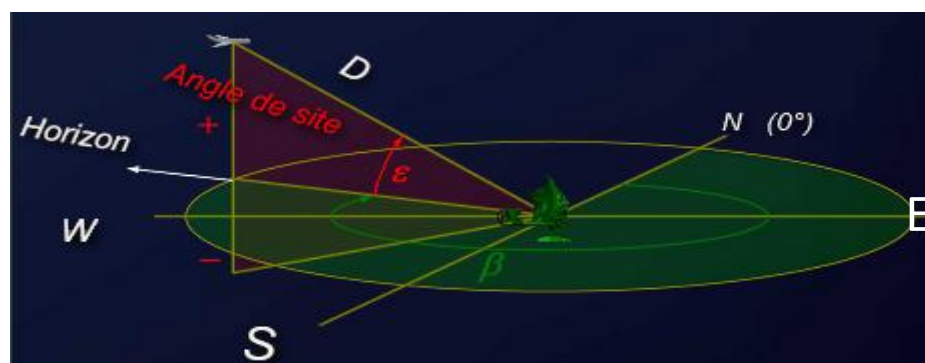


Figure 1.3 : Mesure de l'angle [5] .

1.3.3- Mesure de la vitesse radiale

La vitesse radiale de la cible peut être mesurée de deux façons : Le décalage dû à l'effet Doppler ou la mesure successive de la distance. Même si cela demande un temps de mesure relativement long pour avoir la précision adéquate, c'est la deuxième méthode qui est préférée.

1.3.4-L'effet Doppler

Lorsque la cible est mobile par rapport au radar, son retard évolue au cours du temps

$$\tau = \frac{2R_0(t)}{c} = \frac{2}{c}(R_0 + V_{rad} t) \quad (2)$$

Où V_{rad} est la vitesse radiale de la cible, vitesse de rapprochement ou d'éloignement par rapport au radar. L'onde reçue par le radar est issue d'une cible mobile comme illustrée sur la figure 2 arrive n avance ou en retard par rapport à celle d'une cible fixe. Lorsque V_{rad} est très petit devant C , ce décalage de temps d'arrivée se traduit sur l'onde reçue par un décalage sur la fréquence porteuse F_0 de l'onde [4]. Ce phénomène porte le nom d'effet Doppler. Le décalage en fréquence sur la porteuse, appelée fréquence Doppler, s'écrit :

$$f_D = \frac{2V_{rad}}{\lambda} \quad (3)$$

ou $\lambda = \frac{c}{f_0}$ désigne la longueur d'onde du radar. La mesure de f_D permet de déterminer la vitesse radiale de la cible.

1.4- Radar Ultrasons

1.4.1- Détecteurs à ultrason

L'ultrason est un son dont la fréquence est supérieure à 20 kHz. Le nom vient du fait que leur fréquence est trop élevée pour être audible pour l'homme. Deux principes physiques sont utilisables pour constituer le capteur : par effet électronique ou par effet piézoélectrique. Ces capteurs utilisent l'air comme milieu de propagation. L'émetteur et le récepteur sont situés dans le même boîtier. L'émetteur envoie un train d'ondes qui est réfléchi sur l'objet (ou le travailleur) à détecter et ensuite revient à la source. Le temps mis pour parcourir un aller- retour permet de déterminer la distance de l'objet par rapport à la source. La distance trouvée doit être divisée par deux, car le signal aura franchi le double de la distance pour revenir à l'émetteur. La capacité de détection de ce type de capteur va de quelques centimètres à environ 10 mètres.

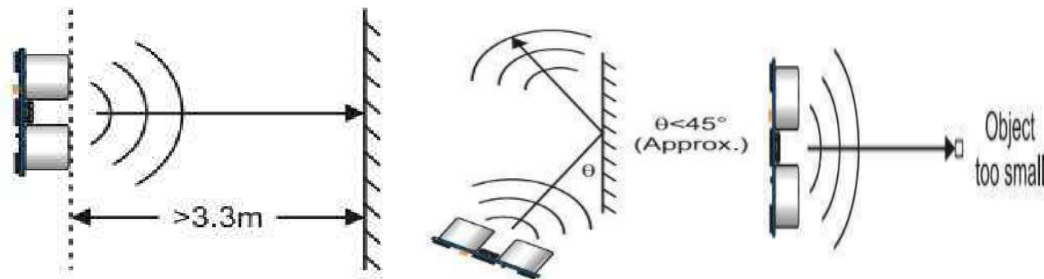


Figure 1.4 : Principe de fonctionnement du détecteur à ultrasonore.

Les capteurs ultrasonores ont de nombreux avantages. Ils ont une large gamme d'utilisation, facile à mettre en œuvre, et leur coût est très faible par rapport aux autres capteurs comme les radars. En plus, l'utilisation des ultrasons est pratiquement sans danger sur la santé humaine. D'un autre côté, les détecteurs ultrasonores présentent les inconvénients suivants :

- La vitesse de propagation des ultrasons peut être influencée de façon sensible par une variation de température. Par exemple, un écart de 30°C peut causer une erreur de 30 cm sur une mesure de 11 m.
- La propagation de l'onde sonore peut aussi être influencée par l'humidité.
- La lecture des capteurs peut être influencée par le bruit environnant et par le signal émis par d'autres capteurs sur le même système.

1.4.2- Les caractéristiques ultrasonores

Les ultrasons se déplacent à des vitesses différentes dans les différents milieux traversés. Ils détiennent les mêmes propriétés générales que les ondes élastiques, c'est-à-dire des ondes vibratoires ou des ondes de pressions dépendant du milieu de propagation. Comme les sons, les ultrasons sont dus à des mouvements de va et vient très rapides, des vibrations.

Par exemple : une corde de guitare que l'on pince, l'air d'une flûte dans laquelle on souffle, créent des vibrations audibles. Les ultrasons, inaudibles par l'oreille humaine, se différencient des sons par le nombre des vibrations effectués en une seconde, c'est-à-dire fréquence. Les sons audibles correspondent à des fréquences comprises entre 20 hertz et 16000 hertz environ.

Chapitre1 : Généralités sur les radars

Si la fréquence est supérieure à 16000 hertz, les vibrations produisent des ultrasons. Pour les détecter, il faut utiliser des récepteurs appropriés.

En échographie les fréquences utilisées vont de 2 à 10 mégahertz car seuls les ultrasons de cette gamme de fréquence se propagent bien dans les milieux liquidiens (organes pleins d'eau ou de sang) du corps humain, sans se disperser et selon une direction bien déterminée.

1 hertz (Hz) = une vibration par seconde

1 mégahertz (MHz) = 1 million de hertz.

1.4.3- Pourquoi le choix des capteurs à ultrason

- En général, la problématique commune à tous les systèmes de détection d'obstacles demeure celle de la réduction des fausses alarmes.
- Le choix du capteur utilisé est fait après avoir effectué une étude bibliographique sur les dispositifs de détection et leurs utilisations.
- Les avantages majeurs des capteurs à ultrason sont leur faible prix de revient et leur simplicité d'implantation, aussi la détection par ultrason est parfaitement efficace pour de courtes distances.

1.5- Classification des Radars selon la technologie

En fonction des informations qu'ils doivent fournir, les équipements radars utilisent des qualités et des technologies différentes. Ceci se traduit par une première classification des systèmes radars comme il est illustré dans la figure suivante (figure 1.5)

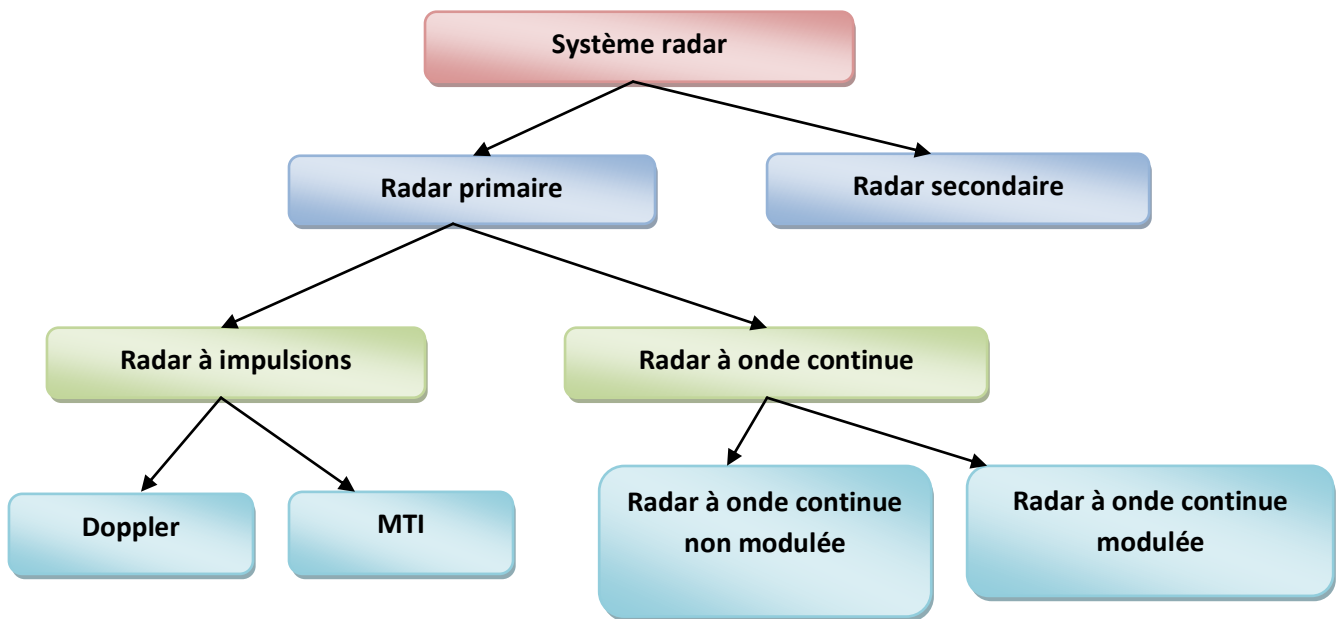


Figure 1.5 : Classification des Radars .

1.5.1- Le Radar primaire

Un Radar primaire transmet des signaux haute fréquence vers les cibles. Les impulsions transmises sont réfléchies par la cible et reçues par le même Radar. L'énergie réfléchie ou les échos sont en outre traités pour extraire des informations utiles [6].

1.5.2- Radar à impulsions

Ce type du Radar transmet des impulsions à hautes fréquence avec une puissance très élevées vers la cible. Ensuite, il attend l'écho du signal transmis pendant un certain temps avant qu'il ne transmet une nouvelle impulsion.

Le Radar à impulsions est généralement utilisé lorsque cela est nécessaire pour détecter des cibles au sein d'un certain volume de l'espace et de déterminer la distance et relèvement et dans certains cas, la vitesse de chaque cible. Le système de radar pulsé nécessite généralement l'émission de grandes puissances et peut-être très complexe et coûteux [6].

Deux grandes catégories de Radars à impulsions utilisant des décalages Doppler sont :

1.5.2.1-Radar à MTI (Moving Target Indicator/ Indicateur de la cible mobile)

Ce Radar utilise une faible fréquence de répétition d'impulsions (PRF) pour éviter les ambiguïtés de distance, mais ces Radars peuvent avoir des ambiguïtés Doppler [6].

1.5.2.2-Radars Doppler à impulsions

Contrairement au Radar MTI, le Radar Doppler à impulsions utilise un PRF élevé pour éviter les ambiguïtés Doppler, mais il peut avoir de nombreuses ambiguïtés de distance [6].

1.5.3- Radar à ondes continues

Les Radars CW(constant wave) transmettent continuellement un signal à haute fréquence et l'énergie réfléchi est également reçue et traitée en continu. Ces Radars doivent s'assurer que l'énergie transmise ne fuit pas dans le récepteur (Connexion de rétroaction). Les Radars CW sont de deux types :

1.5.3.1-Radar à ondes continues non modulées

Le signal transmis de ce type de Radar est constant en amplitude et en fréquence. Le Radar CW qui transmet une puissance non modulée, ne peut mesurer la vitesse qu'en utilisant l'effet Doppler. Il ne peut pas mesurer une distance, et il ne peut pas différencier entre deux cibles réfléchissantes [6].

1.5.3.2-Radar à ondes continues modulées

Les Radars CW non modulés ont l'inconvénient de ne pas mesurer la portée, car les mesures de temps d'exécution ne sont pas possibles dans les Radars CW non modulés. Ceci est réalisé dans les Radars CW modulés en utilisant la méthode de décalage de fréquence. Dans cette méthode, un signal qui change constamment de fréquence autour d'une référence fixe est utilisé pour détecter des objets stationnaires. La fréquence est balayée à plusieurs reprises entre f_1 et f_2 . En examinant les fréquences réfléchies reçues (et avec la connaissance de la fréquence transmise), le calcul de la distance peut être effectué par [6] :

$$R = \frac{c \cdot \Delta t}{2} \quad (4)$$

Avec R : distance.

Δt : différence de temps entre le signal émis et l'écho.

Si la cible se déplace, il y a un changement de fréquence Doppler supplémentaire qui peut être utilisé pour déterminer si la cible approche ou s'éloigne.

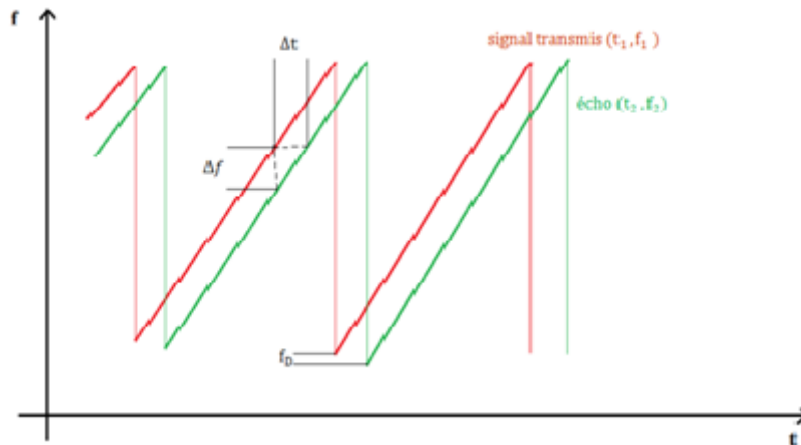


Figure 1.6 : Mesure de distance avec un Radar FMCW [8].

1.5.4- Le Radar secondaire

Les Radars secondaires fonctionnent avec des signaux de réponse actifs. En plus du Radar primaire, ce type de Radar utilise un transpondeur (Transmetteur répondeur) sur la cible où il va répondre à l'interrogation du Radar en générant un signal codé. Cette réponse peut contenir beaucoup plus d'informations que celles qu'un Radar primaire peut collecter (Par exemple l'altitude, un code d'identification, ou encore un rapport de problème à bord comme une panne totale des radiocommunications) [7].

1.5.5- Radar à impulsions

1.5.5.1- Description

Ce type de Radar utilise des impulsions à hautes fréquences, qui se réfléchissent sur toute cible, créant ainsi une onde de retour susceptible d'être détectée par un récepteur adapté à ce signal. Chaque impulsion de durée très brève de l'ordre de quelques microsecondes se propage dans l'atmosphère à la vitesse de la lumière ($C = 3 \cdot 10^8 \text{ m/s}$). Une partie de ce signal est réfléchi par la cible, nous pouvons dire que la cible est illuminée et rayonne une partie de l'énergie émise sous la forme d'une onde de faible amplitude et de caractéristiques temporelles identiques à celle du signal émis. La mesure de la distance se déduit à partir du retard entre l'émission de l'impulsion électromagnétique et sa réception. Les paramètres les plus importants pour déterminer la distance maximale, c'est-à-dire, la plus grande distance mesurable et la résolution du Radar, sont la durée τ des impulsions et la fréquence de répétition PRF [7].

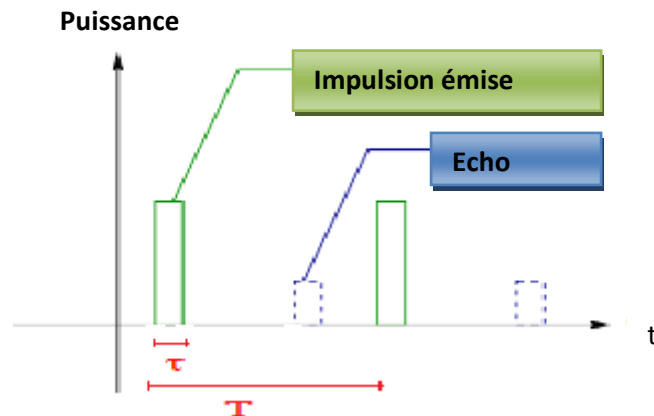


Figure 1.7 : Génération d'échos[8].

1.5.5.2- les Composons d'un Radar à impulsion

Un Radar à impulsions peut être décrit à l'aide du schéma de principe représenté sur la figure ci-dessous, il se compose d'un émetteur, récepteur, antenne et un duplexeur.

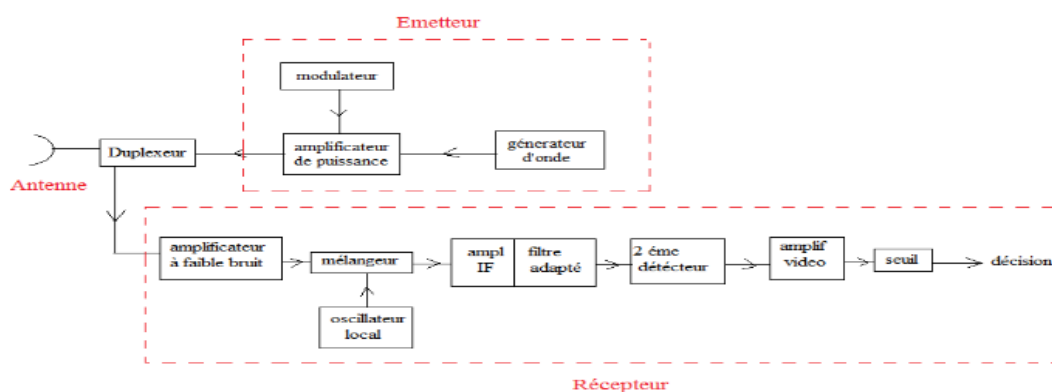


Figure 1.8 : Diagramme de bloc d'un Radar à impulsion[7].

➤ L'émetteur

L'émetteur peut être un oscillateur, tel qu'un magnétron, qui est "pulsé" (Activé et désactivé) par le modulateur pour générer un train répétitif d'impulsions. La forme d'onde générée par l'émetteur se déplace via une ligne de transmission (Guide d'onde) vers l'antenne, où elle est rayonnée dans l'espace.

Une forme commune d'antenne Radar est un réflecteur de forme parabolique, alimenté à partir d'une source ponctuelle à son foyer. Le réflecteur parabolique concentre l'énergie en un faisceau étroit, tout comme un projecteur ou un phare d'automobile.

➤ **Le duplexeur**

Le duplexeur est un commutateur électronique qui dirige l'onde vers l'antenne lors de l'émission ou le signal de retour depuis l'antenne vers le récepteur lors de la réception.

➤ **Le récepteur**

Le récepteur est généralement du type superhétérodyne. La première étape peut être un amplificateur RF à faible bruit, tel qu'un transistor à faible bruit. Cependant, il n'est pas toujours souhaitable d'utiliser un premier étage à faible bruit dans le Radar, surtout dans les Radars militaires qui doivent fonctionner dans un environnement bruyant.

Le mélangeur et l'oscillateur local (LO) convertissent le signal RF en une fréquence intermédiaire (IF). L'amplificateur FI doit être conçu comme un filtre adapté, c'est-à-dire sa fonction de réponse en fréquence $H(f)$ devrait maximiser le rapport de puissance de signal de crête à moyenne de bruit à la sortie. Cela se produit lorsque l'amplitude de la réponse fréquentielle $|H(f)|$ est égale à l'amplitude de spectre d'écho $|S(f)|$, et la partie négative de phase du spectre d'écho est la même que celle du spectre de filtre adapté.

Après avoir maximisé le rapport signal / bruit dans l'amplificateur IF, la modulation d'impulsion est extraite par le second détecteur et amplifiée par l'amplificateur vidéo à un niveau où elle peut être affichée correctement, habituellement sur un tube cathodique (CRT). Des signaux de synchronisation sont également fournis au duplexeur, l'information d'angle est obtenue à partir de la direction de pointage de l'antenne.

1.6- Equation Du Radar

L'équation du radar traduit l'influence de phénomènes physiques sur la puissance rayonnée, la propagation de l'onde, et jusqu'à la réception du signal réfléchi (écho). L'équation du radar permet de réaliser une estimation des performances d'un système radar.

1.6.1- Argumentation Et Explication

Nous supposons que les ondes électromagnétiques se propagent dans des conditions idéales, sans subir de quelconque perturbation. Lorsque l'énergie haute fréquence est rayonnée à partir d'une antenne isotrope, elle se propage de façon

Chapitre1 : Généralités sur les radars

uniforme dans toutes les directions. les zones d'égaies densité de puissance forment donc des surfaces sphériques ($A = 4 \cdot \pi \cdot R^2$) concentriques autour de l'antenne. Lorsque le rayon de la sphère augmente, une même quantité d'énergie est diffusée sur plus des surfaces sphériques. Cela revient à dire que la densité de puissance, dans une direction donnée, diminue lorsque la distance de l'émetteur augmente. [8]

La formule suivante permet de calculer la densité de puissance pour une aérienne omnidirectionnelle:

$$S_u = \frac{P_s}{4 \cdot \pi \cdot R_1^2} \quad (5)$$

S_u : Densité de puissance «omnidirectionnelle» [w/m2]

P_s : Puissance émise [w]

R_1 : Distance antenne – cible [m]

Si l'antenne n'émet que sur une portion de la surface (pour une puissance d'émission constante), la densité de puissance augmente dans la direction de l'émission, Cette caractéristique est appelée gain d'antenne, ce gain est dû à la « concentration » de la puissance émise dans une seul direction.

$$S_g = S_u \cdot G \quad (6)$$

S_g : Densité de puissance « directive » [w/m2]

S_u : Densité de puissance « omnidirectionnel » [w/m2]

G : gain d'antenne

La puissance réfléchie P_r , découle de la densité de puissance S_u , gain d'antenne G et de très fluctuante surface équivalente radar σ :

$$P_r = \frac{P_s}{4\pi R_1^2} \cdot G \cdot \sigma \quad (7)$$

P_r : puissance réfléchie [w]

P_s : puissance émise [w]

σ : surface équivalente radar [m2]

R_1 : distance antenne - cible [m]

D'une façon simplifiée nous pouvons considérer la cible comme un émetteur (du signal réfléchi).

La puissance réfléchie P_r est donc assimilable à une puissance émise (par la cible). Comme les conditions de propagation du signal sont identiques sur le trajet aller et sur le trajet retour, nous pouvons réutiliser la formule (6) pour déterminer la densité de puissance S_u atteignant l'emplacement de l'antenne du radar.

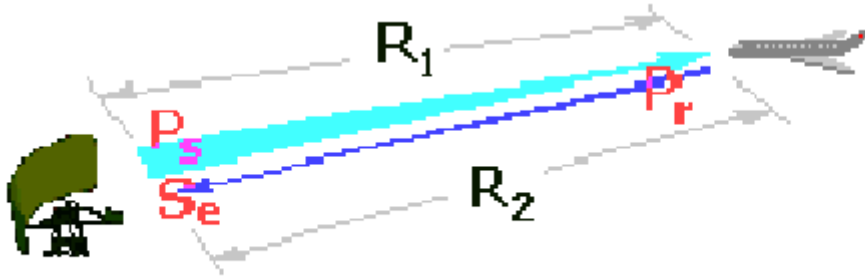


Figure 1.9 : Relation entre les puissances émises et réfléchies[7].

$$S_e = \frac{P_r}{4\pi R_2^2} \quad (8)$$

S_e : densité de puissance au niveau du radar [w/m^2]

P_r : puissance réfléchie [w]

R_2 : distance cible - antenne [m]

Nous avons jusqu'à présent considéré séparément le trajet aller (R_1 =antenne-cible) et le trajet retour (R_2 = cible-antenne) du signal. Nous allons maintenant étendre l'équation au trajet global de l'onde en prenant les définitions des équations précédentes et en notant que $R_1 = R_2 = D$ (l'émetteur et le récepteur constituent le même dispositif: on parle alors de radar monostatique), nous obtenons l'équation suivante :

$$P_e = \frac{P_r}{4\pi R_2^2} \cdot A \cdot K_a \quad (9)$$

$$P_e = \frac{P_s \cdot G}{(4\pi)^2 R_1^2 \cdot R_2^2} \cdot A \cdot K_a \quad (10)$$

A : surface réelle (géométrique) de l'antenne [m^2]

K_a : Facteur d'efficacité

Si l'on en extrait l'expression de $A \cdot K_a$, qu'on insère dans l'équation (8) ci-dessus on obtient après simplification l'équation du radar :

$$P_e = \frac{P_s \cdot G^2 \cdot \lambda^2 \cdot \sigma}{(4\pi)^3 \cdot D^4} \quad (w) \quad (11)$$

Si on suppose la puissance reçue minimum détectable par le récepteur notée, est la distance maximale de détection, et l'équation Radar devient comme suite :

$$D = \left[\frac{P_s \cdot G^2 \cdot \lambda^2 \cdot \sigma}{(4\pi)^3 \cdot P_e} \right]^{\frac{1}{4}} \quad (12)$$

1.6.2- Influence des pertes sur la portée

L'équation du Radar établie aux conditions de propagations idéales, mais en pratique, la propagation est affectée par de nombreuses pertes qui peuvent considérablement réduire l'efficacité du Radar. Ces pertes sont :

- Atténuations internes dans les circuits de l'émetteur et du récepteur, typiquement, elles sont de l'ordre de (1 à 2 dB).
- Pertes dues aux fluctuations de la surface équivalente
- Pertes dues au faisceau, dans l'équation du Radar, on a supposé le gain de l'antenne constant, or ce gain varie pendant le balayage dont il faut tenir compte. Cette perte est de l'ordre de (1,5 dB).

1.6.3 - Portée maximale du Radar en présence du bruit thermique

Puisque le bruit est le facteur principal qui limite la sensibilité du récepteur, il est nécessaire d'obtenir quelques moyens de le décrire quantitativement. Le bruit est une énergie électromagnétique indésirable qui interfère avec la capacité du récepteur à détecter l'écho.

Il peut provenir du récepteur lui-même, ou il peut entrer par l'antenne de réception avec le signal désiré. Si le Radar devait fonctionner dans un environnement parfaitement exempt de bruit, de sorte qu'aucune source externe de bruit n'entraînait le signal désiré et si le récepteur lui-même était si parfait qu'il ne générerait pas d'excès de bruit, il existerait encore une composante de Bruit généré par l'agitation thermique des électrons de conduction dans les parties ohmiques des étages d'entrée du récepteur.

Ceci est appelé bruit thermique, il est directement proportionnel à la température des composants ohmiques du circuit et de la bande passante du récepteur. La puissance thermique disponible générée par un récepteur de largeur de bande b_n (en hertz) à une température T (en Kelvin) est égale à [9]:

$$P_{th} = k \cdot T \cdot b_n \quad (13)$$

Avec K : constante de Boltzmann égal à $1.38 \times 10^{-23} \text{ J/K}$

$$b_n = \frac{\int_{-\infty}^{+\infty} |H(f)|^2 df}{|H(f_0)|^2} \quad (14)$$

Ou $H(f)$: La réponse fréquentielle du filtre IF.

f_0 : La fréquence maximale de la réponse généralement se trouve au milieu de la bande.

➤ Figure de bruit

On peut considérer le bruit total à la sortie du récepteur comme la puissance de bruit thermique obtenue à partir d'un récepteur « Idéal » multiplié par un facteur s'appelle figure de bruit F_n . La figure de bruit d'un récepteur est définie par l'équation [6]:

$$F_n = \frac{\text{bruit à la sortie d'un récepteur réel}}{\text{bruit à la sortie d'un récepteur idéal}} = \frac{S_i/N_i}{S_0/N_0} \quad (15)$$

Avec : S_i : Bruit à l'entrée d'un récepteur réel

N_i : Bruit à l'entrée d'un récepteur idéal

S_0 : Bruit à la sortie d'un récepteur réel

N_0 : Bruit à la sortie d'un récepteur idéal

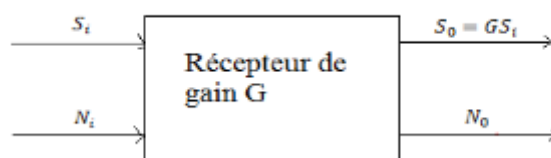


Figure 1.10 : Différents types de bruit traités par un récepteur [6].

Avec :

$$N_i = K \cdot T \cdot b_n \quad (16)$$

Le facteur de bruit devient :

$$F_n = \frac{N_0}{G \cdot K \cdot T \cdot b_n} \quad (17)$$

1.6.3.1-Filtrage non optimal

La bande passante à (-3dB) est définie comme la séparation en fréquence entre la valeur maximale de la réponse fréquentielle et un point où la réponse est réduite à (0.707 donc -3 dB). La largeur de bande à 3 dB est largement utilisée grâce à faciliter de mesure car elle nécessite une connaissance complète de la caractéristique de réponse $H(f)$ [8].

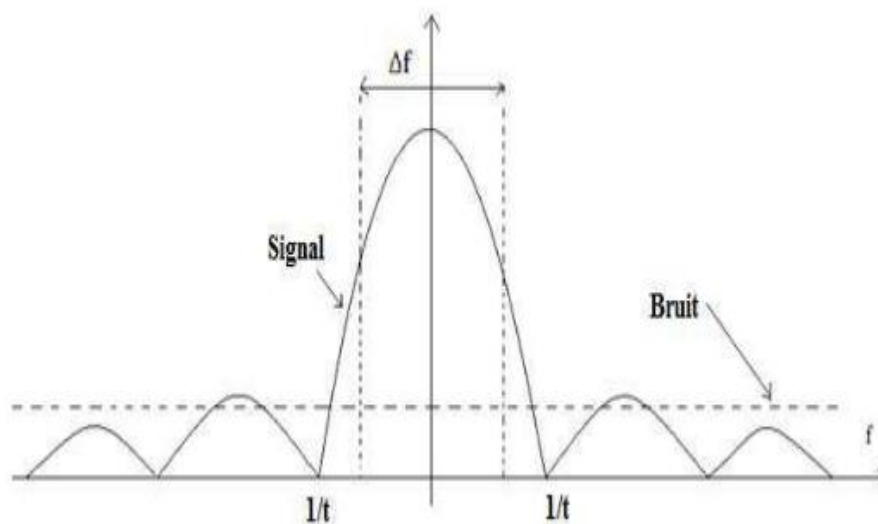


Figure 1.11 : Spectre du signal impulsionnel et du bruit « Blanc »[8].

Un signal impulsionnel a un spectre infini, donc il faut le filtrer avec un filtre passe bande caractérisé par une largeur limitée, cette largeur doit être choisie de façon à optimiser le rapport signal/bruit.

En augmentant la largeur de bande du filtre passe-bande, on augmente la qualité du signal, mais on augmente en même temps la puissance du bruit, et vice versa. D'après les calculs, le rapport signal/bruit passe par un maximum pour ($\Delta f=1.2$) donc le filtre non optimal consiste à utiliser un filtre passe bande de largeur de bande ($\Delta f=1.2$) [6].

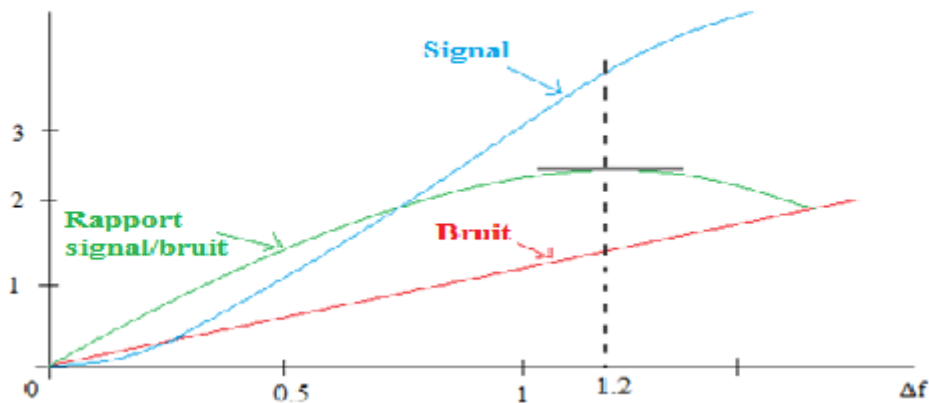


Figure 1.12 : Variation du SNR en fonction de largeur de bande[6].

1.6.3.2-Filtrage optimal (Filtre adapté)

Filtre optimal ou filtre adapté sa caractéristique la plus unique est qu'elle produit le SNR instantané maximum réalisable à sa sortie lorsqu'un signal $s(t)$ et un bruit blanc additif $n(t)$ de moyenne nulle et DSP sont $S_n(f)$ présents à l'entrée. Le bruit n'a pas besoin d'être gaussien. Le SNR à la sortie du récepteur peut être obtenu en adaptant la fonction de transfert du récepteur Radar au signal reçu [6].

On applique à l'entrée :

$$X(t) = s(t) + n(t), t \in [0, T] \quad (18)$$

Où E est l'énergie du signal. On voit donc que le rapport signal sur bruit maximal ne dépend pas de la forme du signal mais uniquement de son énergie [6]:

$$SNR(t_0)^{max} = \frac{2 \cdot E}{N_0} \quad (19)$$

1.7- Radar à diversité de fréquences

La fluctuation de la taille de la cible cause un problème de puissance, de nombreux Radars utilisent deux fréquences ont plus d'illumination.

La diversité de fréquences est réalisée grâce à l'emploi de deux émetteurs fonctionnant en tandem permettant d'éclairer la cible avec deux signaux de fréquences suffisamment éloignées pour que la réception être indépendante [6].

1.8- Applications du Radar

Les radars ont aujourd'hui une très grande variété d'applications dans de nombreux domaines :

- **militaire** : radars de détection et de surveillance aérienne au sol ou embarqués (sur chasseurs pour le combat aérien et sur SDCA pour Système de Détection et de Commandement Aéroporté), radars de veille de surface sur les navires de guerre, radars d'atterrissage ou d'appontage, identification ami-en-nemi (IFF pour Identification Friend or Foe) ,autodirecteurs de missiles, radars de détection terrestre, radars d'artillerie, brouilleurs radar, satellites radar d'observation de la terre.
- **Aéronautique** : contrôle du trafic aérien, guidage d'approche d'aéroport, radars d'altimétrie, radars de navigation.
- **Maritime** : radar de navigation, anticollision, balises radars transpondeur.
- **Météorologie** : détection de précipitations (pluie, neige, grésil, grêle, etc.) et de formations nuageuses. Les radars les plus récents utilisent l'effet Doppler et sont donc capables d'évaluer la vitesse de ces particules. Certains radars utilisent les polarisations verticale et horizontale pour donner une idée du mélange de formes des particules sondées ce qui, associé à leur intensité, peut indiquer le type de précipitation.
- **Circulation et sécurité routière** : contrôle de la vitesse des automobiles (voir cinémomètre), le modèle classique sur les routes de France est le Miradop (mini radar doppler) utilisé par les brigades de gendarmerie. Ils sont placés sur les autoroutes, dans les zones où les véhicules peuvent rouler à une vitesse supérieure à la vitesse maximale autorisée. Dans le même esprit, le radar de régulation de distance (ACC Adaptive Cruise Control) sur automobile permet de contrôler l'espace entre les voitures en utilisant les ondes électromagnétiques ou laser.
- **Systèmes de sécurité** : (anti intrusion) fonctionnant par l'évaluation de la présence d'un objet en mouvement par effet doppler. Les encombrants radars à cavité des années 1970-1980 ont été remplacés par de plus petits, nettement moins gourmands en énergie, généralement associés à un détecteur infrarouge (on parle alors de « double technologie »). En extérieur, on utilise également des couples émetteurs récepteurs disposés face à face

- **Radars de recul sur automobile** : cette appellation « radar » est utilisée de manière abusive car on utilise des transducteurs acoustiques pour émettre et recevoir des ondes ultrasonores, et non des ondes électromagnétiques. Il s'agit donc plutôt d'un sonar que d'un radar à proprement parler.
- **Scientifiques** : embarqués sur satellite pour l'observation de la Terre ou du niveau des océans.

1.9-Conclusion :

Au terme de cette étude théorique, nous avons vu des généralités sur les systèmes radar, tel que la définition du radar, son principe de fonctionnement et les différentes parties des radars, et différents types de radar ainsi que leurs domaines d'applications. Dans le prochain chapitre, nous présenterons les moyens et les outils nécessaires pour la réalisation de notre projet.

Chapitre 2
Matériel Et Composants

Chapitre 2 :Matériels et composants

2.1- Introduction

Avant d'aborder la réalisation du système radar, nous allons présenter les outils nécessaires à sa conception. Dans ce contexte ce chapitre porte sur la présentation des outils logiciels et matériels tels la carte arduino utilisé pour le traitement des données et le capteur de distance HC-SR04 et les autres composants périphériques.

2.2- Etude de matérielle

2.2.1- Matériel utilisée :

Pour réaliser le Radar de détection nous avons besoin de :

- Carte Arduino Uno.
- Capteur ultrasons HC-SR04.
- Servomoteur.
- Afficheur LCD.

2.2.2- Définition du module Arduino

Arduino est la marque d'une plateforme de prototypage open-source qui permet aux utilisateurs de créer des objets électroniques interactifs à partir de cartes électroniques matériellement libres sur lesquelles se trouve un microcontrôleur (D'architecture Atmel comme l'Atmega328p, et d'architecture ARM comme le Cortex-M3 pour l'Arduino Due) . Les schémas de ces cartes électroniques sont publiés en licence libre [10].

Une carte Arduino est une petite carte électronique (5,33 x 6,85 cm) équipée d'un micro-contrôleur. Le micro-contrôleur permet, à partir d'événements détectés par des capteurs, de programmer et commander des actionneurs ; la carte Arduino est donc une interface programmable [11].la carte Arduino la plus utilisée est la carte Arduino Uno.

L'intérêt principal des cartes ARDUINO (d'autres modèles existent) est leur facilité de mise en œuvre. ARDUINO fournit un environnement de développement s'appuyant sur des outils open- source. Le chargement du programme dans la mémoire du microcontrôleur se fait de façon très simple par port USB.

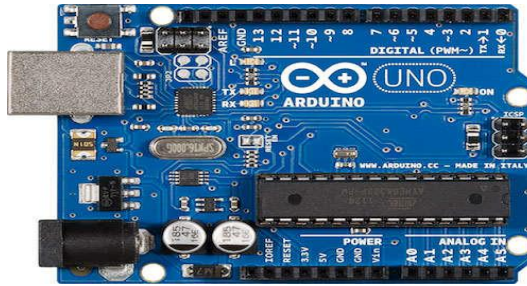


Figure 2.1 : Carte Arduino [12] .

2.2.3- La Carte Arduino Uno :

L'Arduino est une famille de cartes électronique à microcontrôleur open-source née en Italie en 2005. Ces cartes basées sur une interface entrée/sortie simple et sur un environnement de développement proche du langage C.

La carte Arduino uno est la première version stable de carte Arduino. Elle possède toutes les fonctionnalités d'un microcontrôleur classique en plus de sa simplicité d'utilisation. Elle utilise une puce ATmega328P cadencée à 16 Mhz. Elle possède 32ko de mémoire flash destinée à recevoir le programme, 2ko de SRAM (mémoire vive) et 1 ko d'EEPROM (mémoire morte destinée aux données). Elle offre 14 pins (broches) d'entrée/sortie numérique (donnée acceptée 0 et 1) dont 6 pouvant générer des PWM (Pulse width Modulation). Elle permet aussi de mesurer des grandeurs analogiques grâce à ces 6 entrées analogiques. Chaque broche est capable de délivrer un courant de 40 mA pour une tension de 5V. Cette carte Arduino peut aussi s'alimenter et communiquer avec un ordinateur grâce à son port USB. On peut aussi l'alimenter avec une alimentation comprise en 7V et 12V grâce à son connecteur Power Jack [13].

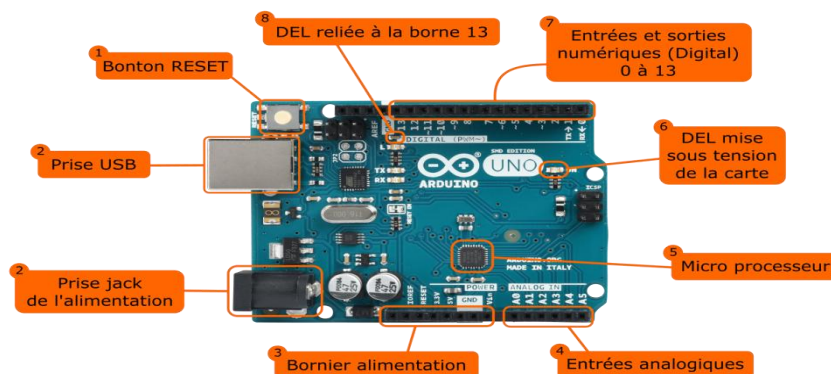


Figure 2.2: schéma de carte arduino uno[14] .

Chapitre 2 : Matériels et composants

2.2.3.1- Les caractéristiques de la carte Arduino Uno

Les caractéristiques techniques de la carte Arduino Uno sont présentées dans le tableau suivante (tableau 2.1) :

Tableau 2.1 : Caractéristiques de la Carte Arduino UNO

Microcontrôleur	ATMega328
Tension de fonctionnement	5V
Tension d'alimentation – entrée - (Recommandée)	7-12V
Tension d'alimentation – entrée - (Limites)	6-20V
Broches E/S numériques	14 (Dont 6 disposent d'une sortie PWM)
Broches d'entrées analogiques	6 (Utilisables en broches E/S numériques)
Intensité maximum disponible par broche E/S (5V)	40 mA (ATTENTION : 200mA cumulé pour l'ensemble des broches E/S)
Intensité maximum disponible pour la sortie 3.3V	50 mA
Intensité maximum disponible pour la sortie 5V	Fonction de l'alimentation utilisée 500 mA max si port USB utilisé seul
Mémoire Programme Flash	32 KB (ATmega328) dont 0.5 KB sont utilisés par le bootloader
Mémoire SRAM (Mémoire volatile)	2 KB (ATmega328)

2.2.3.2- les composants de la carte Arduino

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

Un microcontrôleur ATMega328 est un circuit intégré qui rassemble sur une puce plusieurs éléments complexes dans un espace réduit en soudant un grand nombre de composants encombrants; tels que les transistors; les résistances et les condensateurs tout peut être logé dans un petit boîtier en plastique noir muni d'un certain nombre de broches dont la programmation peut être réalisée en langage C [15 ,16]. La figure (2.3) montre deux types de microcontrôleur ATmega328, qu'on trouve sur les cartes Arduino.

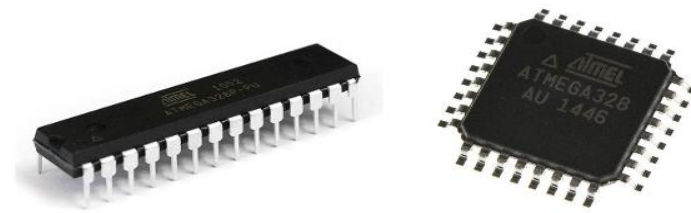


Figure 2.3 : Microcontrôleur ATmega328[17].

Le microcontrôleur ATmega328 est constitué par un ensemble d'éléments qui ont chacun une fonction bien déterminée. Il est en fait constitué des mêmes éléments que la carte mère d'un ordinateur. Globalement, l'architecture interne de ce circuit programmable se compose essentiellement de :

- **Mémoire Flash :** C'est celle qui contiendra le programme à exécuter. Cette mémoire effaçable et réinscriptible est une mémoire programmée de 32Ko (dont bootloader de 0.5 ko).
- **RAM :** c'est la mémoire dite "vive", elle va contenir les variables du programme. Elle est dite "volatile" car elle s'efface si on coupe l'alimentation du microcontrôleur. Sa capacité est 2 ko.
- **EEPROM :** C'est le disque dur du microcontrôleur. On y enregistre des infos qui ont besoin de survivre dans le temps, même si la carte doit être arrêtée. Cette mémoire ne s'efface pas lorsque l'on éteint le microcontrôleur ou lorsqu'on le reprogramme [18].
- **Sources de l'Alimentation :** La carte Arduino Uno peut-être alimentée soit via la connexion USB (qui fournit 5V jusqu'à 500mA) ou à l'aide d'une alimentation externe. La source d'alimentation est sélectionnée automatiquement par la carte [19].

La source d'alimentation est sélectionnée automatiquement par la carte [20]. L'alimentation externe peut être soit un adaptateur secteur (Pouvant fournir typiquement de 3V à 12V sous 500mA) ou des piles, la plage idéale recommandée pour alimenter la carte Uno est entre 7V et 12V.

Les broches d'alimentation sont les suivantes :

- **VIN :** La tension d'entrée positive lorsque la carte Arduino est utilisée avec une source de tension externe (à distinguer du 5V de la connexion USB ou autre source 5V régulée). Vous pouvez alimenter la carte à l'aide de cette broche, ou, si

Chapitre 2 : Matériels et composants

l'alimentation est fournie par le jack d'alimentation, accéder à la tension d'alimentation sur cette broche.

- **5V** : La tension régulée utilisée pour faire fonctionner le microcontrôleur et les autres composants de la carte (pour info : les circuits électroniques numériques nécessitent une tension d'alimentation parfaitement stable dite "tension régulée" obtenue à l'aide d'un composant appelé un régulateur et qui est intégré à la carte Arduino). Le 5V régulé fourni par cette broche peut donc provenir soit de la tension d'alimentation VIN via le régulateur de la carte, ou bien de la connexion USB (qui fournit du 5V régulé) ou de toute autre source d'alimentation régulée.
- **3.3V** : Une alimentation de 3.3V fournie par le circuit intégré FTDI (circuit intégré faisant l'adaptation du signal entre le port USB de votre ordinateur et le port série de l'ATmega) de la carte est disponible : ceci est intéressant pour certains circuits externes nécessitant cette tension au lieu du 5V). L'intensité maximale disponible sur cette broche est de 50mA.
- **GND** : Broche de masse (ou 0V)[21].

▪ **Entrée sortie de la carte ARDUINO** : La carte Arduino Uno est dotée de :

6 entrées analogiques, 14 entrées/sorties numériques dont 6 peuvent assurer une sortie PWM, chacune des entrées/sorties de la carte ne peut pas délivrer plus de 20 mA[21].

➤ **Les entrées analogiques (Broches analogiques) :**

La carte Uno dispose de 6 entrées analogiques (numérotées de 0 à 5), chacune pouvant fournir une mesure d'une résolution de 10 bits (càd sur 1024 niveaux soit de 0 à 1023) à l'aide de la très utile fonction [analogRead\(\)](#) du langage Arduino. Par défaut, ces broches mesurent entre le 0V (valeur 0) et le 5V (valeur 1023), mais il est possible de modifier la référence supérieure de la plage de mesure en utilisant la broche AREF et l'instruction [analogReference\(\)](#) du langage Arduino.

Note : les broches analogiques peuvent être utilisées en tant que broches numériques : elles sont numérotées en tant que broches numériques de 14 à 19[21].

➤ **Les entrées/sorties numériques (broches numériques) :**

Chacune des 14 broches numériques de la carte UNO (numérotées des 0 à 13) peut être utilisée soit comme une entrée numérique, soit comme une sortie numérique, en utilisant les instructions [pinMode\(\)](#), [digitalWrite\(\)](#) et [digitalRead\(\)](#) du langage Arduino. Ces broches fonctionnent en 5V. Chaque broche peut fournir ou recevoir un maximum de 40mA d'intensité et dispose d'une résistance interne de "rappel au plus"

Chapitre 2 : Matériels et composants

(pull-up) (déconnectée par défaut) de 20-50 KOhms. Cette résistance interne s'active sur une broche en entrée à l'aide de l'instruction [digitalWrite](#)(broche, [HIGH](#)).

De plus, certaines broches ont des fonctions spécialisées :

- **Communication Série** : Broches 0 (RX) et 1 (TX). Utilisées pour recevoir (RX) et transmettre (TX) les données séries de niveau TTL. Ces broches sont connectées aux broches correspondantes du circuit intégré ATmega8U2 programmé en convertisseur USB-vers-série de la carte, composant qui assure l'interface entre les niveaux TTL et le port USB de l'ordinateur.
 - **Interruptions Externes**: Broches 2 et 3. Ces broches peuvent être configurées pour déclencher une interruption sur une valeur basse, sur un front montant ou descendant, ou sur un changement de valeur. Voir l'instruction [attachInterrupt\(\)](#) pour plus de détails.
 - **Impulsion PWM (largeur d'impulsion modulée)**: Broches 3, 5, 6, 9, 10, et 11. Fournissent une impulsion PWM 8-bits à l'aide de l'instruction [analogWrite\(\)](#).
 - **SPI (Interface Série Périphérique)**: Broches 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). Ces broches supportent la communication SPI (Interface Série Périphérique) disponible avec la [librairie pour communication SPI](#). Les broches SPI sont également connectées sur le connecteur ICSP qui est mécaniquement compatible avec les cartes Mega.
 - **I2C**: Broches 4 (SDA) et 5 (SCL). Supportent les communications de protocole I2C (ou interface TWI (Two Wire Interface - Interface "2 fils"), disponible en utilisant [la librairie Wire/I2C \(ou TWI - Two-Wire interface - interface "2 fils"\)](#)).
 - **LED**: Broche 13. Il y a une LED incluse dans la carte connectée à la broche 13. Lorsque la broche est au niveau HAUT, la LED est allumée, lorsque la broche est au niveau BAS, la LED est éteinte [21].
 - **Puce d'interface USB** : En haut à gauche de la carte à côté de la prise USB est l'interface USB puce. Celle ci convertit les niveaux du signal utilisé par le standard USB à des niveaux qui peuvent être utilisés directement par l'Arduino[20].
- **Oscillateur à quartz**
 - **Interrupteur de réinitialisation**
 - **Ports de communications** : La carte Arduino Uno dispose de toute une série de facilités pour communiquer avec un ordinateur, une autre carte Arduino, ou avec

Chapitre 2 : Matériels et composants

d'autres microcontrôleurs. L'ATmega328 dispose d'une UART (Universal Asynchronous Receiver Transmitter ou émetteur-récepteur asynchrone universel) pour la 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 vers le port USB de l'ordinateur et 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. Cependant, sous Windows, un fichier inf est requis [22,23].

Le logiciel Arduino inclut une fenêtre terminal série (ou moniteur série) sur l'ordinateur 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 USB- vers-série et la connexion USB vers l'ordinateur. Une librairie Série Logicielle permet également la communication série sur n'importe quelle broche numérique de la carte UNO.

L'ATmega 328 supporte également la communication par protocole I2C (ou interface TWI, TwoWire Interface - Interface "2 fils") et SPI :

- Le logiciel Arduino inclut la librairie Wire qui simplifie l'utilisation du bus I2C.
- Pour utiliser la communication SPI (Interface Série Périphérique).
- communication SPI est disponible [20].

2.2.3.3- Le Principe de fonctionnement d'Arduino

Les différentes versions des Arduins fonctionnent sous le même principe général :

- On conçoit ou on ouvre un programme existant avec le logiciel Arduino.
- On vérifie ce programme avec le logiciel Arduino (compilation).
- Si des erreurs sont signalées, on modifie le programme.
- On charge le programme sur la carte.
- On câble le montage électronique.
- L'exécution de programme est automatique.
- On alimente la carte soit par le port USB. Soit par une source d'alimentation autonome.
- On vérifie que notre montage fonctionne.

Chapitre 2 : Matériels et composants

2.2.3.4- Les avantages de l'arduino uno

- Pas cher.
- Le point positif de cette carte, c'est qu'elle est open-source, leur
- Environnement de programmation clair et simple.
- Multiplateforme tourne sous Windows, Macintosh et Linux.

2.2.4- Capteur Ultrasons HC-SR04

Le HC-SR04 est un type de capteur à ultrasons qui utilise un sonar pour connaître la distance entre l'objet et le capteur. Il fournit cette information grâce à une électronique à laquelle il est associé voir figure 2.4. Il fournit une gamme exceptionnelle de détection sans contact avec une précision élevée et des lectures stables. Il comprend deux modules comme un émetteur et un récepteur à ultrasons. Ce capteur est utilisé dans une variété d'applications telles que la mesure de la direction et de la vitesse, les alarmes antivol, le médical, le sonar, les humidificateurs, la charge sans fil, les tests non destructifs et l'échographie[24].

Ce capteur de distance à ultrasons permet des mesures de distance allant de (2cm à 500cm) avec une précision pouvant aller jusqu'à (3mm). L'angle du cône de mesure est d'environ (15°). 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 [25].



Figure 2.4: Capteur Ultrason HC-SR04 [25].

Chapitre 2 : Matériels et composants

2.2.4.1- Caractéristiques et spécification du capteur

Les caractéristiques en détail du Capteur sonar à Ultrasons HC-SR04 sont présentées dans le tableau ci-dessous [26].

Tableau 2.2 : Les Caractéristiques de capteurs ultrason HC-SR04.

Distance de captation	2 cm à 5 m
Résolution (précision)	3mm
Tension d'exploitation (Voltage d'entrée)	5 V
Courant (Ampérage d'entrée)	15 mA
Fréquence d'opération	40 Hz
Angle de mesure	30 degrés
Angle efficace	15 degrés
Signal d'entrée trigger	10 μ s TTL impulsion
Dimensions L x W x H	45 mm x 20 mm x 15mm
Poids	8.5g

2.2.4.2- Broches de connexion

Les broches du Capteur sonar à Ultrasons HC-SR04 sont :

- **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[27].

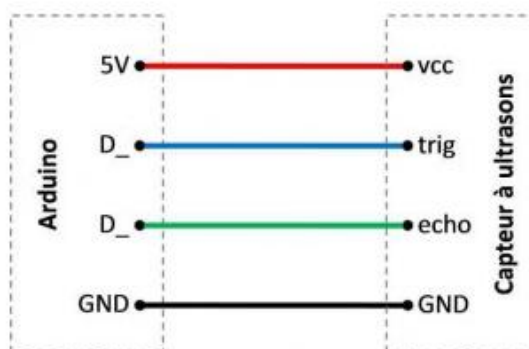


Figure 2.5: Broches de connexion de capteur a ultrason avec arduino[28].

2.2.4.3-Fonctionnement du capteur Ultrason HC-SR04

Pour déclencher une mesure, il faut présenter une impulsion "High" (5 V) d'au moins 10 μ 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 [19].

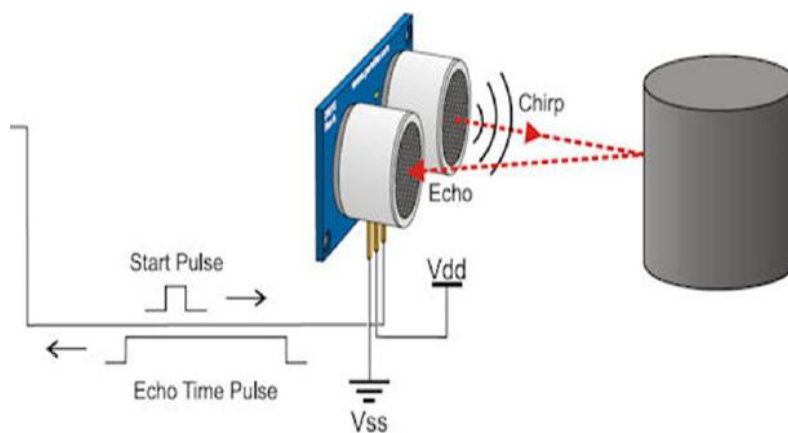


Figure 2.6: Principe de fonctionnement de HC-SR04 [28].

➤ Mesure de distance de la cible

La distance parcourue par un son se calcule en multipliant la vitesse du son, environ 340 m/s (ou 34'000 cm/1'000'000 μ s) par le temps de propagation, soit :

$$d = v \cdot t \quad (2.1)$$

d : distance en mètre , **v** : vitesse de son (m/s) mètre par seconde , **t** : temps (s) en seconde.

Le HC-SR04 donne une durée d'impulsion en dizaines de μ s. Il faut donc multiplier la valeur obtenue par 10 μ s pour obtenir le temps t. On sait aussi que le son fait un aller-retour. La distance vaut donc la moitié.

$$d = \left(\frac{34000}{1000000} \cdot 10 \cdot \text{Valeur} \right) \cdot \frac{1}{2} \quad (\text{cm}) \quad (2.2)$$

Chapitre 2 : Matériels et composants

En simplifiant :

$$d = \frac{17000}{1000000} \cdot \text{Valeur} \quad (\text{cm}) \quad (2.3)$$

Finalement :

$$d = \frac{17}{100} \cdot \text{Valeur} \quad (\text{cm}) \quad (2.4)$$

$$d = \frac{\text{durée}}{58} \quad (\text{cm}) \quad (2.5)$$

La formule (2.5) figure aussi dans le manuel d'utilisation du HC-SR04 car la fraction $17/1000$ est égale à $1/58.8235$. Elle donne cependant des résultats moins précis.

Note : À grande distance, la surface de l'objet à détecter doit mesurer au moins 0.5 m^2 [19].

➤ Le diagramme temporel

Le diagramme temporel est affiché ci-dessous (figure 2.7). Il faut fournir un signal à l'état haut pendant $10\mu\text{s}$ à l'entrée de l'émetteur pour déclencher la salve d'ultrasons. Le module envoie ensuite 8 cycles d'ultrasons à 40 kHz et attend le signal reçu. Le récepteur reçoit un signal à l'état haut dont la durée est proportionnelle à la distance de l'objet perçu. Le calcul de la distance peut se faire grâce au temps mesuré entre le signal émis et le signal reçu.

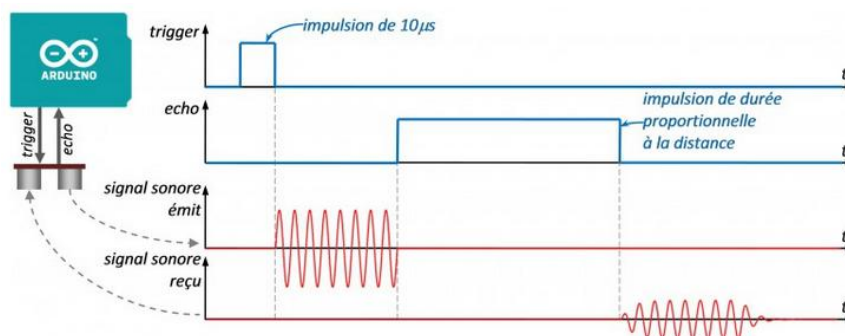


Figure 2.7: Le diagramme temporel [28].

2.2.5- Module Afficheur LCD

Les afficheurs à cristaux liquides, autrement appelés afficheurs LCD (Liquid Crystal Display), sont des modules compacts intelligents et nécessitent peu de composants externes pour un bon fonctionnement. Ils consomment relativement

Chapitre 2 : Matériels et composants

peu (de 1 à 5 mA), sont relativement bons marchés et s'utilisent avec beaucoup de facilité.

Plusieurs afficheurs sont disponibles sur le marché et diffèrent les uns des autres, non seulement par leurs dimensions, (de 1 à 4 lignes de 6 à 80 caractères), mais aussi par leurs caractéristiques techniques et leur tension de service. Certains sont dotés d'un rétroéclairage de l'affichage. Cette fonction fait appel à des LED montées derrière l'écran du module, cependant, cet éclairage est gourmand en intensité (de 80 à 250 mA), Deux modèles se rencontrent très fréquemment et sont les meilleurs marché, celui ayant 2 lignes et 16 colonnes d'affichage et celui ayant 4 lignes et 20 colonnes d'affichage voir figure 2.8. Ils sont très utilisés dans les montages à microcontrôleur, et permettent une grande convivialité. Ils peuvent aussi être utilisés lors de la phase de développement d'un programme, car on peut facilement y afficher les valeurs de différentes variables.



Figure 2.8 : Afficheurs LCD (a) (16x2) (b) (20x4) [25].

2.2.5.1- Connecteur de l'afficheur LCD

Ces deux écrans ont exactement la même nomenclature des broches, un connecteur 16 broches véhicule plusieurs signaux dont une partie forme un bus de communication parallèle 4 ou 8 bits selon la configuration choisie ainsi que les signaux permettant de contrôler la communication entre l'Arduino et l'écran [12].

La Figure 2.9 donne la nomenclature des broches de ce connecteur :

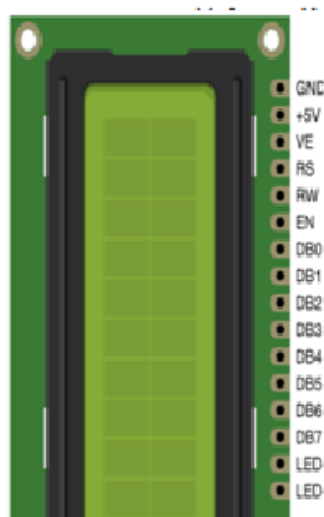


Figure 2.9 : Connecteur de l'afficheur LCD [22].

Le tableau (2.3) présente les spécifications des broches et leur rôle [8] :

Tableau 2.3 : Nomenclature du connecteur de l'afficheur LCD

Numéro de la broche	Nom	Rôle
1	GND	Masse 0V
2	VDD	Alimentation +5V
3	VE	Tension de réglage du contraste
4	RS	Sélection du registre donnée ou commande
5	RW	Lecture ou écriture
6	EN	Activation pour un transfert (enable)
7	DB0	Bit 0 de la donnée/commande
8	DB1	Bit 1 de la donnée/commande
9	DB2	Bit 2 de la donnée/commande
10	DB3	Bit 3 de la donnée/commande
11	DB4	Bit 4 de la donnée/commande
12	DB5	Bit 5 de la donnée/commande
13	DB6	Bit 6 de la donnée/commande
14	DB7	Bit 7 de la donnée/commande
15	LED+	Anode (+) du rétro-éclairage
16	LED-	Cathode (-) du rétro-éclairage

Chapitre 2 : Matériels et composants

2.2.5.2- Communication avec l'afficheur LCD

L'afficheur LCD peut fonctionner en mode 4 bits ou en mode 8 bits. En mode 8 bits, les octets sont transférés sur les lignes DB0 à DB7. En mode 4 bits les octets sont transférés en deux fois sur les lignes DB4 à DB7. Le LCD dispose de 3 registres internes, le registre de données permettant entre autre l'envoi des codes des caractères à afficher, le registre de commande permettant d'envoyer des commandes d'effacement de l'écran, de positionnement du curseur, etc. et le registre d'état qui permet de consulter notamment la disponibilité du LCD pour recevoir des commandes ou des données.

La sélection de l'un ou l'autre de ces registres est effectuée via les états, LOW ou HIGH, des lignes RS et RW. Une fois l'état de ces deux lignes établi, EN est placé à HIGH, la donnée ou la commande est placée sur les lignes DBx puis EN est placé à LOW [19].

Piloter directement un afficheur LCD est un processus relativement compliqué. Évidemment, comme c'est très souvent le cas avec l'Arduino comme dans notre situation, il existe des bibliothèques pour ça, ce qui permet de les utiliser aisément sans avoir à plonger dans la data sheet en mode (4 bits), les broches à connecter à l'Arduino sont donc RS, EN, DB4, DB5, DB6 et DB7 ainsi que, de façon optionnelle le RW [19].

2.2.6- Servomoteur

Un servomoteur est un système qui vise à produire un mouvement précis en réponse à une commande externe. C'est un actionneur (système produisant une action) qui mélange l'électronique, la mécanique et l'automatique. Il est composé d'un moteur à courant continu avec deux caractéristiques spéciales: au lieu d'avoir une rotation constante, il est asservi en position angulaire, cela signifie que l'axe de sortie du servomoteur respectera la consigne d'instruction que vous lui avez envoyée et maintiendra sa position fixe permettant de contrôler la vitesse de rotation[22].

Le modèle SG90 est un des plus populaires à cause de son prix réduit. Il s'agit d'un servo très petit (26.7×11.8×22.5 mm) et il est souvent utilisé dans la construction de robots ou à niveau éducationnel. Le système d'engrenages de ce servo est en plastique, raison pour laquelle il se dégrade rapidement. Il est idéal

Chapitre 2 : Matériels et composants

pour des petits projets ou pour faire des test. Il est tout à fait compatible avec Arduino.

Un servomoteur est composé :

- D'un moteur à courant continu.
- D'un axe de rotation et potentiomètre.
- D'une carte électronique pour le contrôle de la position de l'axe et le pilotage du moteur.



Figure 2.10 : Servomoteur.

2.2.6.1- les caractéristiques du servomoteur

Les principales caractéristiques d'un servomoteur sont: le couple et la vitesse. Le couple est la force qui est capable de faire sur son axe, et se mesure en Kg/cm. De façon proportionnel si le couple augmente, la consommation d'énergie augmente également. La vitesse est la vitesse angulaire ou de rotation[22].

Les caractéristiques du SG90 sont les suivantes :

Tableau 2.4 : Les caractéristiques du servomoteur SG90.

Dimensions	22 x 11.5 x 27 mm
Poids	9 g
Tension d'alimentation	4.8v à 6v
Vitesse	0.12 s / 60° sous 4.8v
Couple	1.2 Kg / cm sous 4.8v
Amplitude	de 0 à 180°

Chapitre 2 : Matériels et composants

2.2.6.2- Connecteur du servomoteur

Le servo est équipé d'une prise de type Graupner à 3 fils. Cette prise permet à la fois de l'alimenter et de lui transmettre des consignes de position par le fil de signal [29].

La correspondance des fils est la suivante :

- Le noir ou marron : La masse (-)
- Le rouge : La tension d'alimentation continue (+5 V)
- Le jaune, orange, blanc ou bleu : Le signal de commande PWM



Figure 2.11 : Connecteur du servomoteur.

2.2.6.3-Fonctionnement du servomoteur

Le servomoteur est commandé par l'intermédiaire d'un câble électrique à 3 fils qui permettent d'alimenter le moteur et de lui transmettre des ordres de positions sous forme d'un signal codé en largeur d'impulsion plus communément appelés PWM (Pulse Width Modulation ou Modulation de Largeur d'Impulsion) ou RCO (Rapport Cyclique d'Ouverture)

Cela signifie que c'est la durée des impulsions qui détermine l'angle absolu de l'axe de sortie et donc la position du bras de commande du servomoteur. Le signal est répété périodiquement, en général toutes les 20 millisecondes, ce qui permet de contrôler et de corriger continuellement la position angulaire de l'axe de sortie, cette dernière étant mesurée par le potentiomètre [26].

Et à partir de la :

- pour une impulsion de 1 ms, le servomoteur se met en position 0° ;
- pour une impulsion de 1,5 ms, le servomoteur se met en position 90° ;
- pour une impulsion de 2 ms, le servomoteur se met en position 180° .

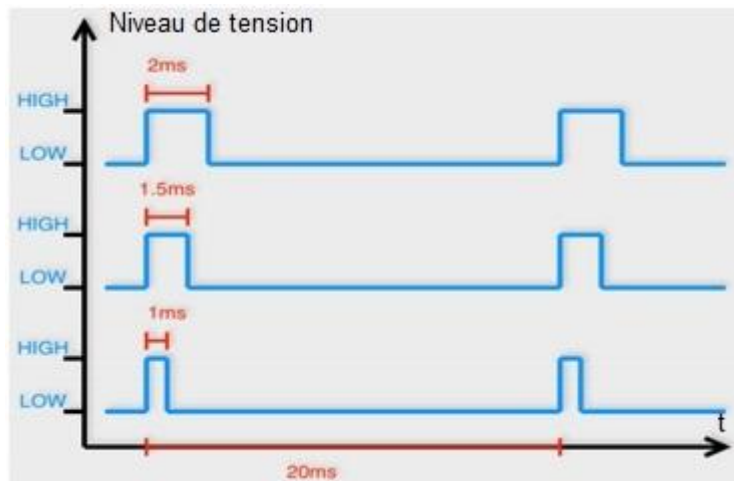


Figure 2.12 : Impulsions de commande d'un servomoteur[30].

➤ Montage du servomoteur avec Arduino

Ce type de servomoteur peut être branché directement sur l'Arduino. Par exemple le branchement pourra être le suivant :

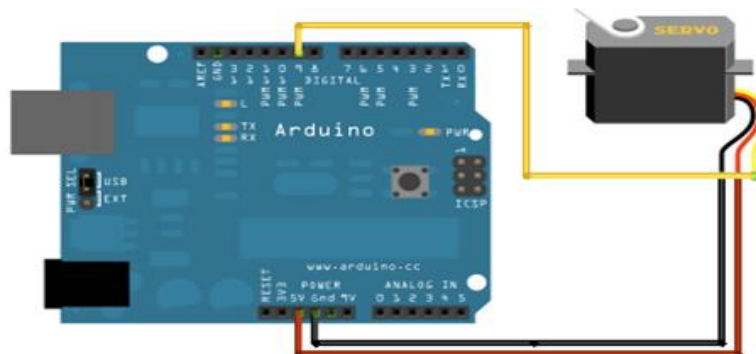


Figure 2.13 : montage servomoteur avec Arduino [29].

➤ La librairie Servo.h

Pour piloter vos servomoteurs, vous aurez besoin de la **librairie servo.h** qui fait partie des bibliothèques de base de l'environnement Arduino. Parmi les fonctions proposées par la librairie servo.h, deux sont les plus utilisées :

```
#include <Servo.h> //Librairie  
  
Servo myServo; // création d'un objet Servo
```

- attach(pin) : permet d'attacher le servomoteur au Pin d'asservissement
- write(angle) : envoi le servo à l'angle spécifié

Chapitre 2 : Matériels et composants

Ces quatre autres fonctions pourront être utiles pour gérer plus finement les servomoteurs :

- writeMicroseconds(us) : permet d'ajuster les positions limites si votre servomoteur n'est pas standard.
- read(pin) : permet de lire l'angle sur lequel est actuellement positionné le servo
- attached(pin) : permet de savoir sur un servomoteur est attaché au Pin spécifié
- detach(pin) : pour détacher votre servomoteur du Pin spécifié.

La librairie servo.h supporte jusqu'à 12 servomoteurs. Sachez toutefois que l'utilisation de servos entraîne des limitations sur votre Arduino. En effet, sur les cartes autres que la Mega, il n'est plus possible d'écrire avec l'instruction analogWrite() sur les broches 9 et 10 (même s'il n'y a aucun servo de branché dessus). Si votre projet nécessite plus de 12 servos, il faudra passer à l'Arduino Mega qui permet d'en piloter jusqu'à 48. Sur la Mega, à partir du 13ème servo, les impulsions PWM seront **désactivées** sur les broches 11 et 13.

Les servomoteurs de positionnement angulaire permettent de déplacer précisément un objet dans une plage de 0° à 180°.

2.3- Etude de Logiciels de programmation

2.3.1- Interface de programmation Arduino

➤ Programmation Arduino

L'environnement de développement Arduino IDE est l'outil utilisé pour écrire du code dans n'importe quel langage de programmation, puis le convertir en un format exécutable qui peut être placé sur le microcontrôleur du circuit [31].



Figure 2.14 : Icône de programme Arduino.

Chapitre 2 : Matériels et composants

- Pour connaître l'interface graphique de l'environnement de développement Arduino :

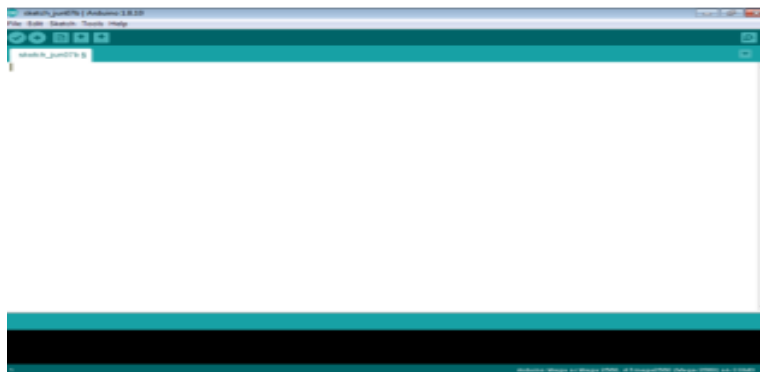


Figure 2.15 : Interface de programme Arduino.

- **Description de l'IDE:**

L'IDE est un logiciel de programmation qui permet d'écrire, de modifier un programme et de le convertir en une série d'instructions compréhensibles pour la carte. Il programme par code, contenant une cinquantaine de commandes différentes. A l'ouverture, l'interface visuelle du logiciel contient le menu, des boutons de commande en haut, une page blanche vierge, une bande noire en bas, comme ceci [31] :

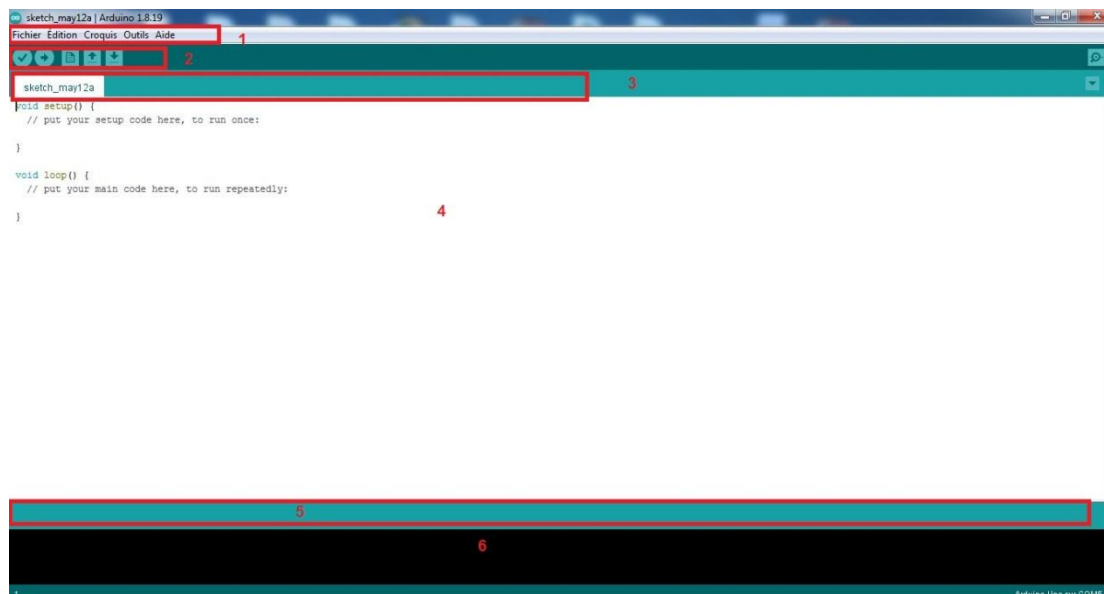


Figure 2.16 : IDE Arduino.

- L'environnement de développement logiciel se compose d'une interface simple et est divisé en quatre parties principales :
 - 1- La barre de menu
 - 2- Barre de boutons

Chapitre 2 :Matériels et composants

- 3- Onglets des fichiers ouverts
- 4- Fenêtre des éditions des Programmation
- 5- Zones des messages des actions en cours
- 6- Consoles d'affichages des messages de complications.

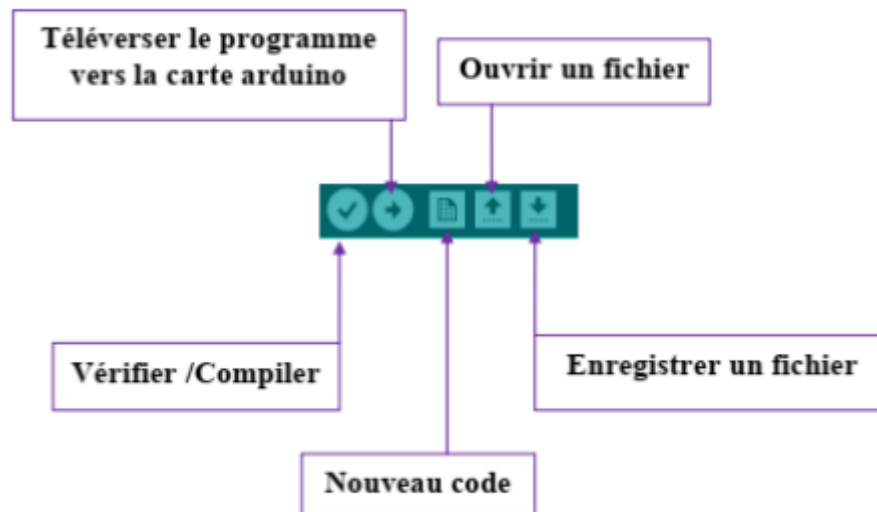


Figure 2.17: Barre de boutons Arduino.

- Pour préparer l'environnement de développement, nous devons choisir la carte avec laquelle nous allons traiter, et cela se fait en allant dans le menu Outils et en choisissant Carte, puis nous choisissons le type de carte à notre disposition et dans ce cas nous choisirons Arduino Uno.

Chapitre 2 : Matériels et composants

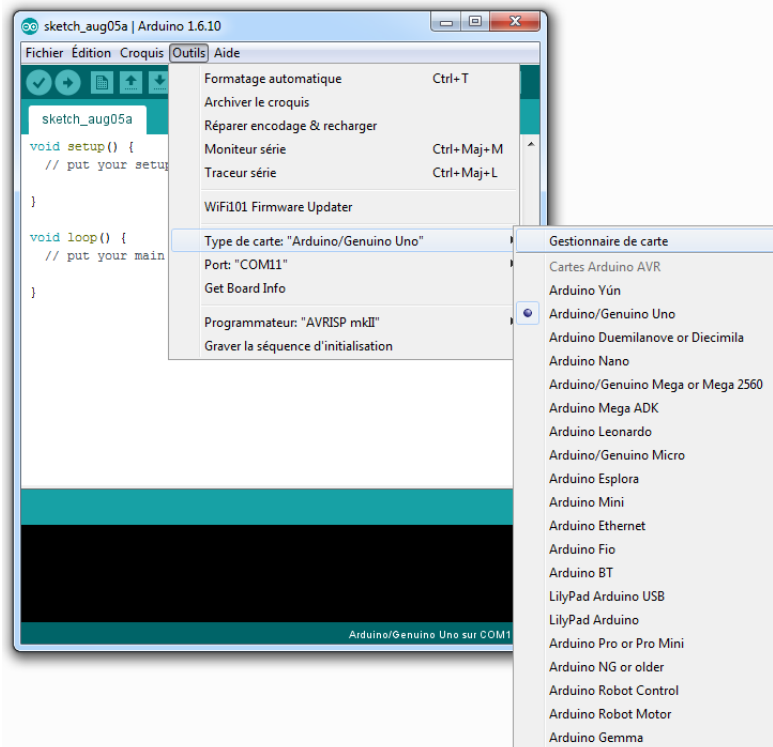


Figure 2.18 : élection de la carte Arduino.

- Ensuite, le numéro de port utilisé dans l'arduino doit être spécifié comme dans l'image suivante (numéro de port COM6) :

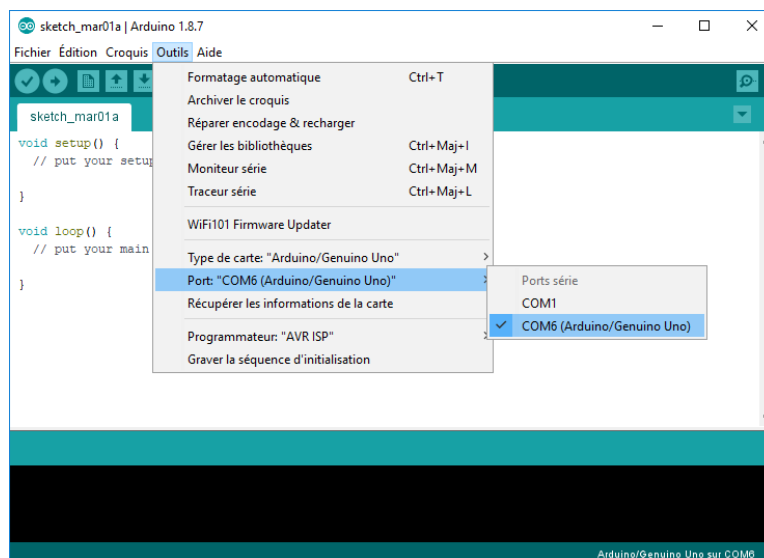


Figure 2.19 : Sélection du port.

Après avoir terminé les processus de vérification et de téléchargement, nous extrayons le fichier HEX en suivant le chemin donné :

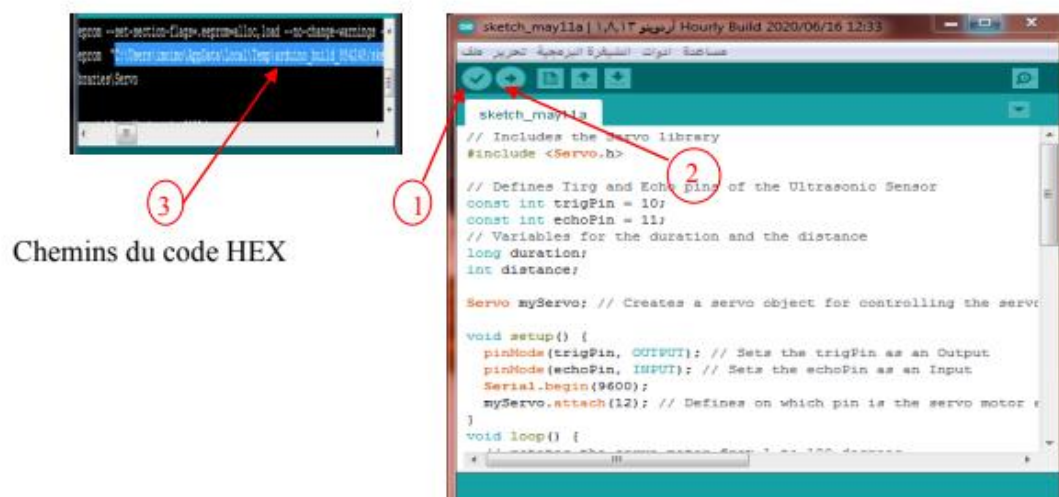


Figure 2.20: Le processus de copie du fichier HEX[31].

- Le logiciel comprend aussi un moniteur série (Equivalent à HyperTerminal) qui permet de d'afficher des messages textes émis par la carte Arduino et d'envoyer des caractères vers la carte Arduino (en phase de fonctionnement) :

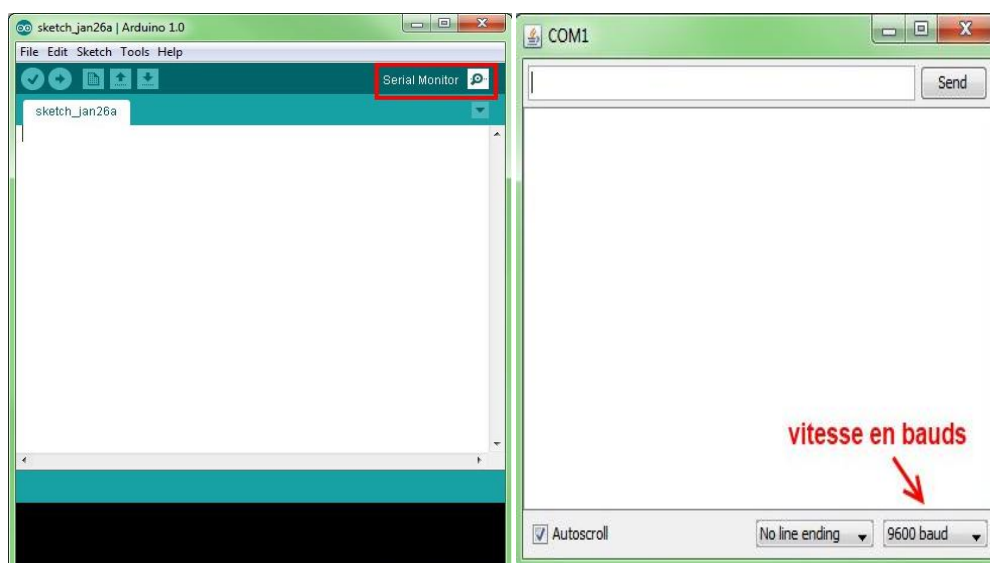


Figure 2.21: Hyper terminal de l'Arduino (Moniteur Série)

Chapitre 2 :Matériels et composants

2.3.2- Logiciel IDE de PROCESSING

Processing est un langage de programmation et une interface de développement intégré (IDE) écrite en Java. Les programmes réalisés avec Processing peuvent donc fonctionner sur toute machine possédant une machine virtuelle Java ainsi que dans les navigateurs Web équipé du plugin Java.

Processing est le prolongement d'un environnement de programmation graphique développé pour la création plastique et graphique interactive. Il dispose de nombreuses bibliothèques, principalement orientées vers les interfaces graphiques, le traitement d'images 2D et 3D et du son. Il relie les concepts de logiciel aux principes de la forme visuelle, du mouvement et de l'interaction. Il intègre un langage de programmation, un environnement de développement et une méthodologie d'enseignement dans un système unique. Le Processing a été créé pour enseigner les fondamentaux de la programmation informatique dans un contexte visuel, être utilisé comme outil de production pour des contextes spécifiques. Les étudiants, les artistes, les professionnels du design et les chercheurs l'utilisent pour l'apprentissage, le prototypage et la production [32].



Figure 2.22 : Icône de logiciel Processing[33] .

➤ **L'interface de logiciel**

L'interface d'utilisation de Processing est composée de deux fenêtres distinctes : la fenêtre principale dans laquelle vous allez créer votre projet et la fenêtre de visualisation dans laquelle vos créations (dessins, animations, vidéos) apparaissent[33].

On trouve plus précisément les éléments suivants dans l'interface :

1. Barre d'actions
2. Barre d'onglets
3. Zone d'édition (pour y saisir votre programme)

Chapitre 2 :Matériels et composants

4. Console, qui comprend un onglet les messages affichés par programme un onglet pour les erreurs. Cette console indique aussi si des mises à jour (« updates ») sont disponibles pour les bibliothèques et les modes.
5. Fenêtre de visualisation (espace de dessin)
6. Liste déroulante pour les modes
7. Bouton pour activer le mode debug (pas à pas)

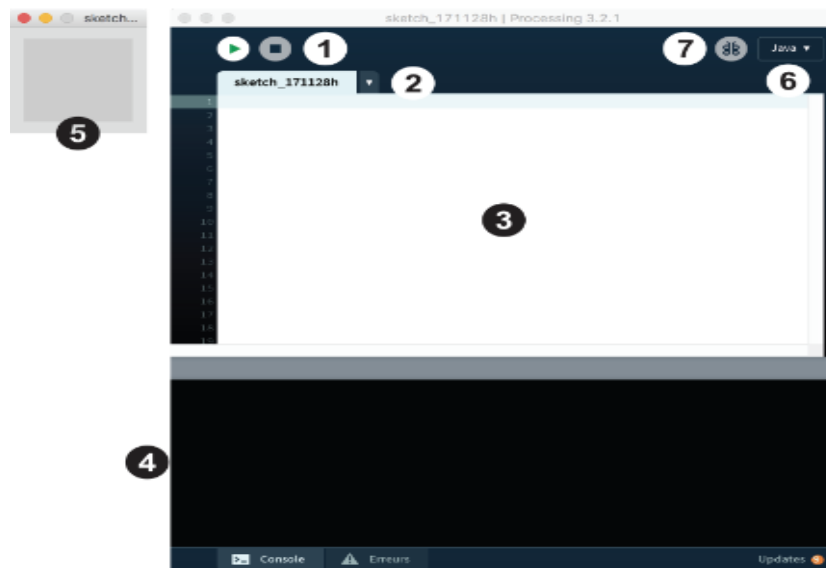


Figure 2.23 : L'interface de logiciel processing[33] .

2.3.3- Logiciel PROTEUS ISIS

➤ Programmation Proteus

Proteus est un programme de modélisation de systèmes virtuels est l'un des meilleurs logiciels dans le domaine de la simulation car il combine des systèmes SPICE pour simuler des circuits et des éléments électroniques (les illustre sous forme de clips d'images mobiles pour une manipulation et une précision plus précises) et des modèles de microprocesseurs pour faciliter une étape des étapes de simulation de systèmes électroniques basés sur des microcontrôleurs [31].

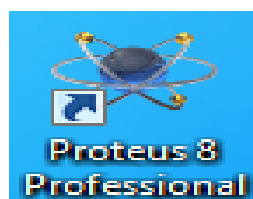


Figure 2.24 : Icône de programme Proteus.

Chapitre 2 : Matériels et composants

- Pour connaître l'interface graphique de l'environnement de développement Proteus:

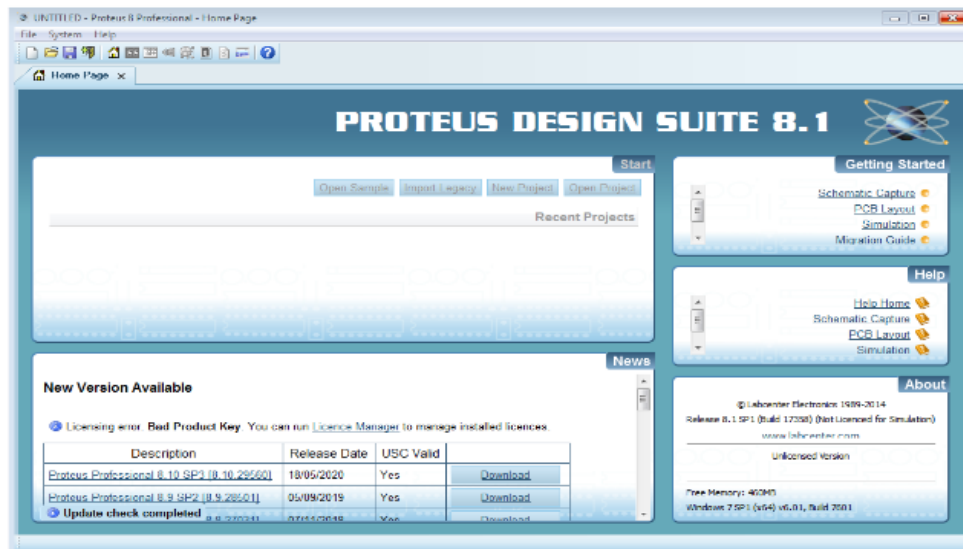


Figure 2.25 : Interface principale de programme Proteus .

- L'environnement de développement logiciel se compose d'une interface simple :

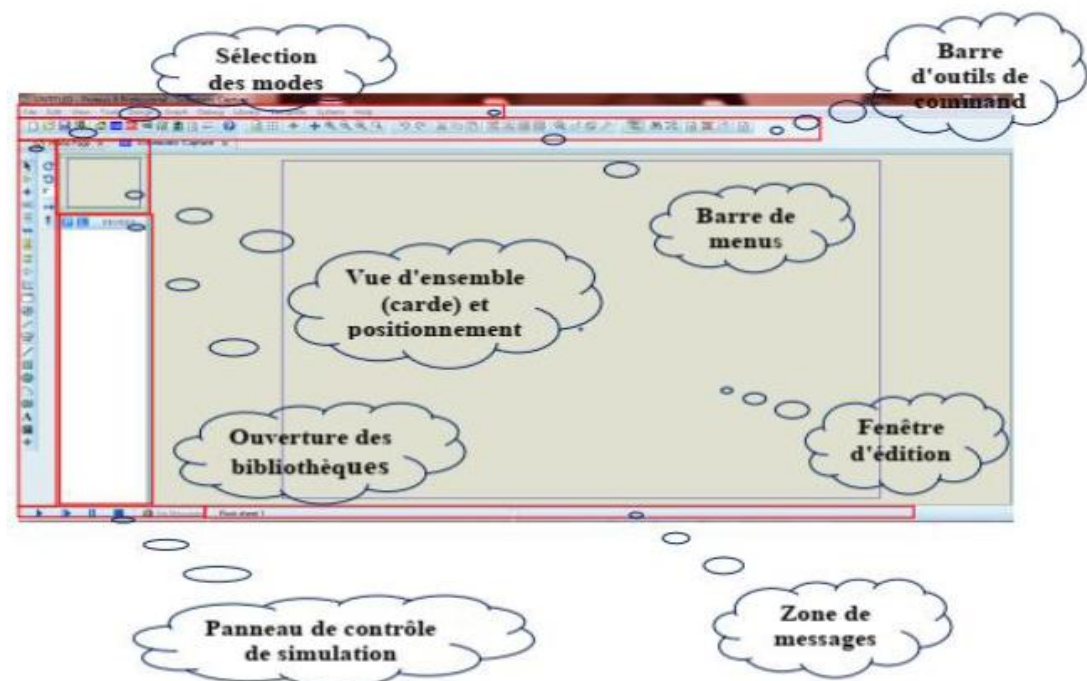


Figure 2.26 : interface du sous-programme Proteus isis[31].

Chapitre 2 :Matériels et composants

2.3.4 - Communication entre la carte Arduino, logiciel Proteus et logiciel Processing

L'IDE de Processing est similaire à Arduino en termes de structure. Il a des fonctions de configuration et des fonctions de dessin comme un Arduino IDE a une fonction de configuration et de boucle. L'IDE de traitement peut communiquer avec l'IDE Arduino via une communication en série. De cette façon, nous pouvons envoyer des données de l'Arduino à l'IDE de processing et aussi de l'IDE de processing à l'Arduino.

Le déroulement de la communication entre le logiciel Arduino IDE et Processing, se fait à travers une liaison série half duplex, les données sont téléversés vers la carte Arduino où se passe le traitement ensuite les résultats de traitement seront transférés vers le logiciel Processing sous la même liaison, voir figure 2.27.

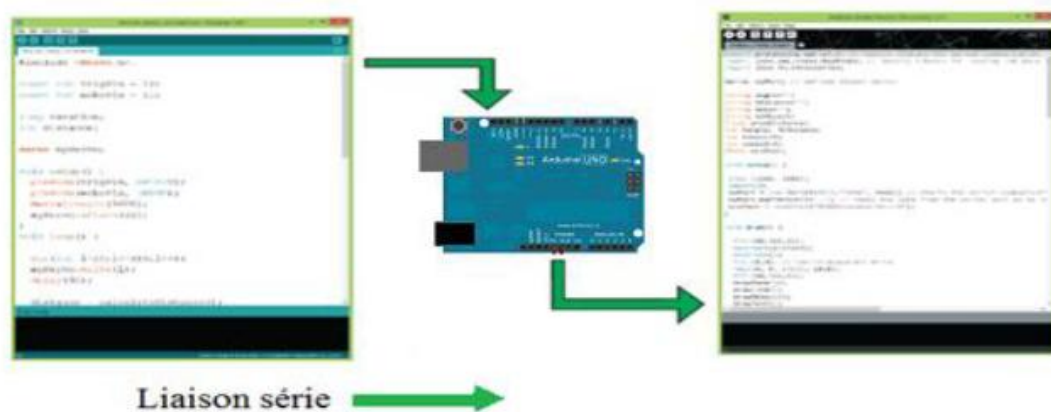


Figure 2.27 : Communication entre la carte Arduino et Processing[6].

Dans le cas de la simulation, et après la compilation de code Arduino, il faut exporter les données de type binaire en hexadécimale pour l'exécuter par la carte Arduino au niveau de simulateur ISIS, cette dernière va traiter les données reçues puis elle va orienter les résultats obtenues vers les processing mais il n'existe pas une liaison entre eux, pour cela il faut créer une liaison virtuelle à l'aide de logiciel « Virtual Serial Port Driver » en identifiant deux ports l'un à côté de processing et l'autre pour la carte Arduino d'une part et d'autre part il faut rajouter un adaptateur pin-port à côté de la carte Arduino pour être prêt d'émettre des données à travers cette liaison. La communication entre Proteus et Processing est présentée par la figure (2.28) :

Chapitre 2 : Matériels et composants

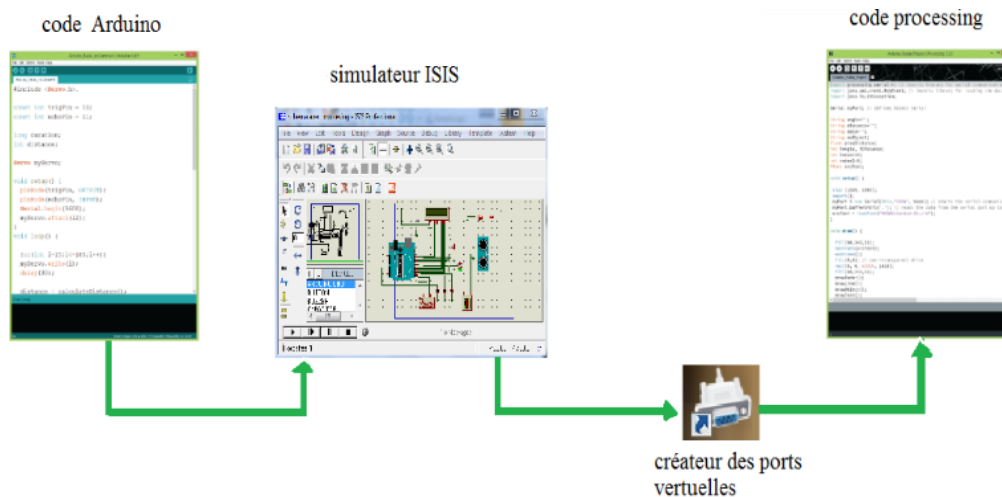


Figure 2.28 : Communication entre PROTEUS et PROCESSING[6].

2.3.5- Schéma d'interconnexion entre les composants

La carte Arduino effectue des lectures à travers les ports numériques, ensuite détecte l'angle du servomoteur et calcule les coordonnées cartésiennes de la cible pour les afficher le sur l'écran.

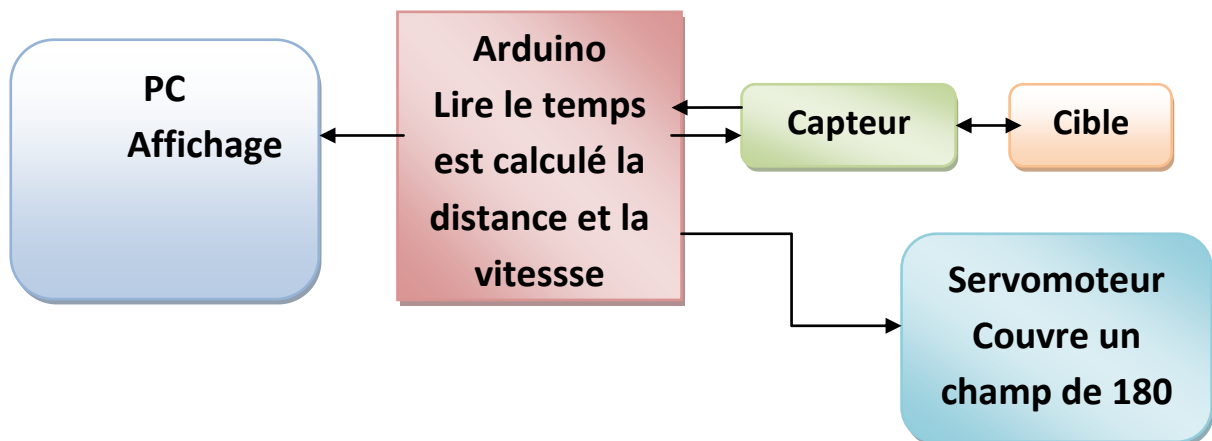


Figure 2.29 : Schéma d'interconnexion entre les composants du projet.

2.4- conclusion

A travers ce chapitre nous avons présenté les outils matériels et logiciels tels la carte Arduino Uno, le capteur de distance ultrasons, l'afficheur LCD et le servomoteur ainsi que les logiciels Arduino et processing et ISIS Proteus que nous allons utiliser pour concevoir un radar de détection. Cette partie sera développée dans le chapitre suivant.

Chapitre 3
Simulation Et Réalisation

Chapitre 3: Simulation Et Réalisation

3.1-Introduction

Dans ce chapitre, nous allons présenter la mise en place du dispositif expérimental "Radar" détection". Après une description théorique des cartes Arduino, servomoteurs, capteurs à ultrasons, écran LCD qui a été présenté dans le chapitre précédent et nous continuons pour les applications expérimentales, pour cette raison, plusieurs blocs sont nécessaires pour réaliser cette combinaison avec le module Arduino et son environnement de développement d'IDE.

3.2- Principe de fonctionnement

Le Radar est un système de détection d'objet qui utilise des ondes radio ou des micro-ondes pour déterminer la portée, l'altitude, la direction et la vitesse des objets. L'antenne Radar transmet des impulsions d'ondes radio ou de micro-ondes qui rebondissent sur tout objet sur leur chemin. L'objet renvoie une partie de l'onde reçue par le récepteur qui est en ligne de vue avec l'émetteur.

3.3-Les étapes du projet

Notre projet de réalisation a été fait en deux parties:

- **La 1^{er} partie** : La simulation avec PROTEUS.
- **La 2^{ème} partie** : la réalisation pratique.

Pour les deux parties de simulation et de réalisation, on passe par :

- Programmation orientée d'objet (programmation par bloc).
- Visualisation des résultats trouvés sur l'afficheur LCD et sur l'écran d'ordinateur à l'aide de logiciel Processing.

3.4- Schéma synoptique du radar

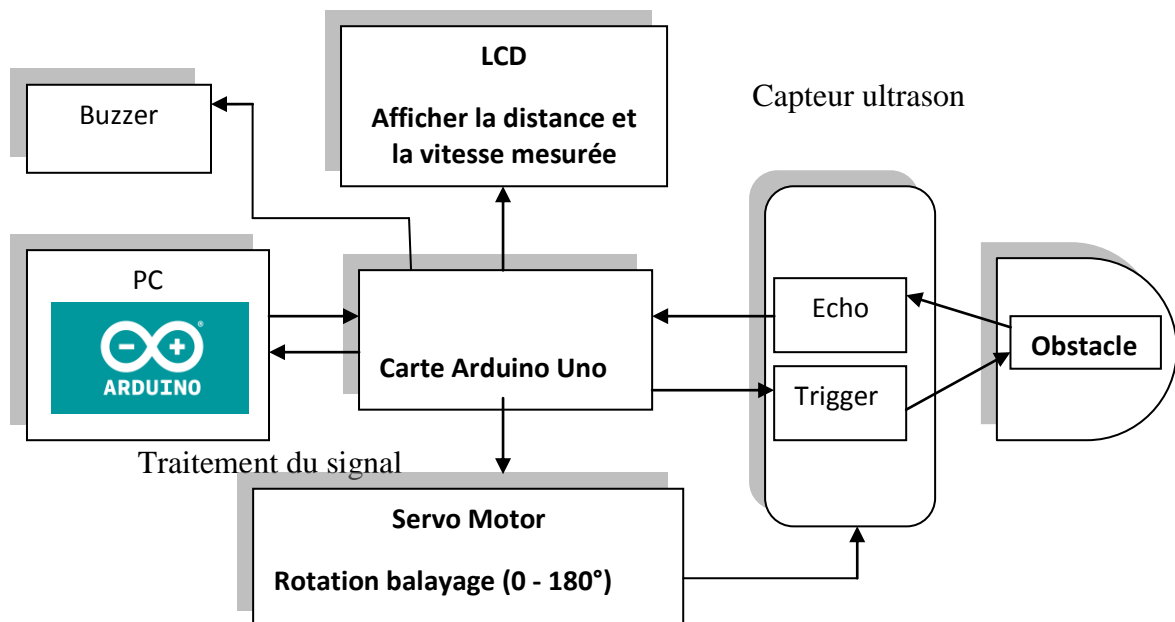


Figure 3.1 : Schéma synoptique de notre projet (Radar)[34].

Chapitre 3: Simulation Et Réalisation

3.5- Algorithme principale de détection d'un objet

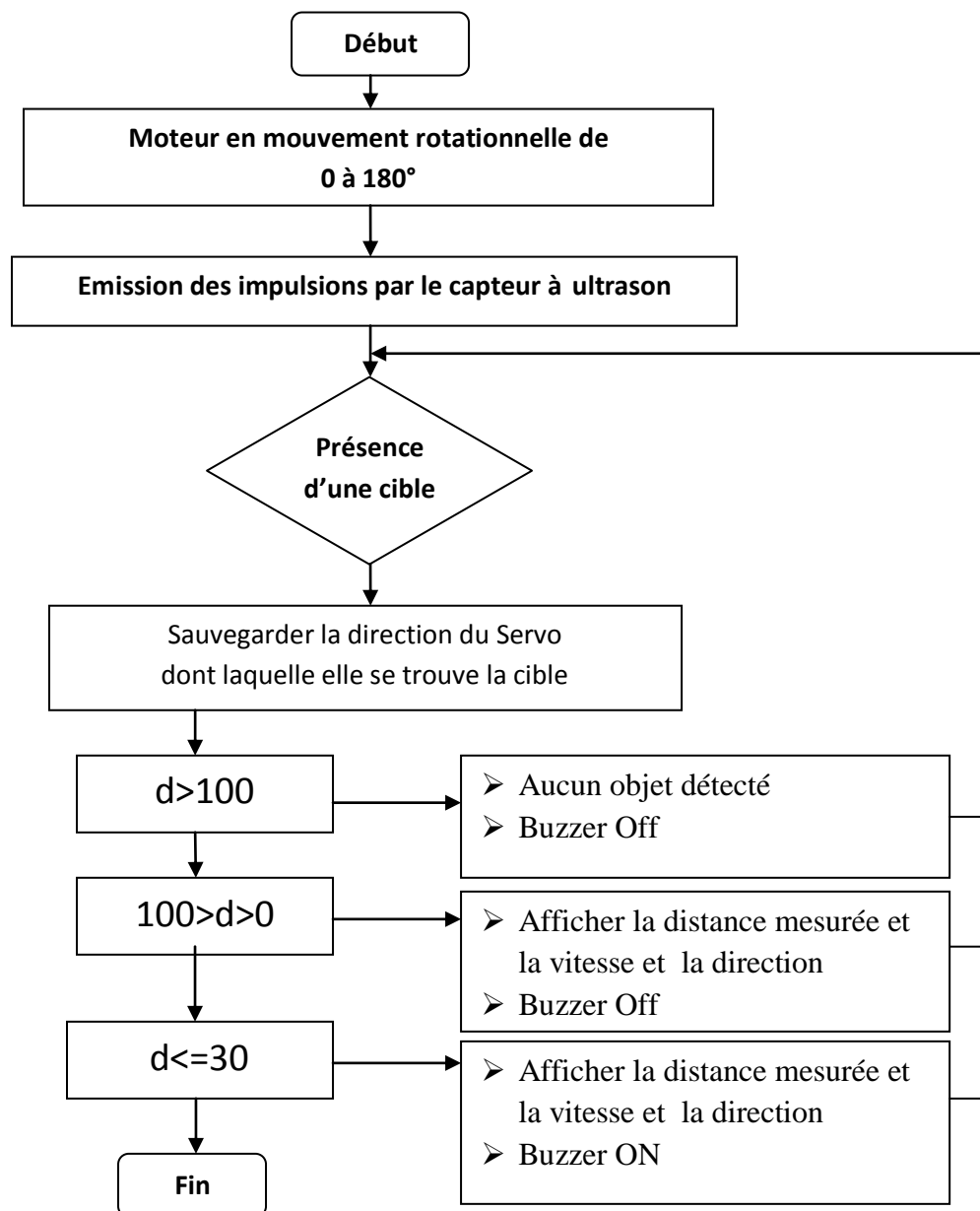


Figure 3.2 : Algorithme principale de détection d'un objet.

3.6- Simulation du projet sous PROTEUS

3.6.1- Bibliothèque Arduino pour Proteus

Pour simuler une carte Arduino sur Proteus, il faut d'abord télécharger la bibliothèque Arduino sur Internet, et décompresser et copier les deux fichiers nommés "ArduinoTEP.LIB et ArduinoTEP.IDX" et les placer dans le dossier de la bibliothèque de notre logiciel Proteus.

Chapitre 3: Simulation Et Réalisation

Après, on redémarre le logiciel Proteus et dans la recherche de sections de composants pour ArduinoTEP, on choisit Arduino Uno comme indiqué sur la figure 3.3.

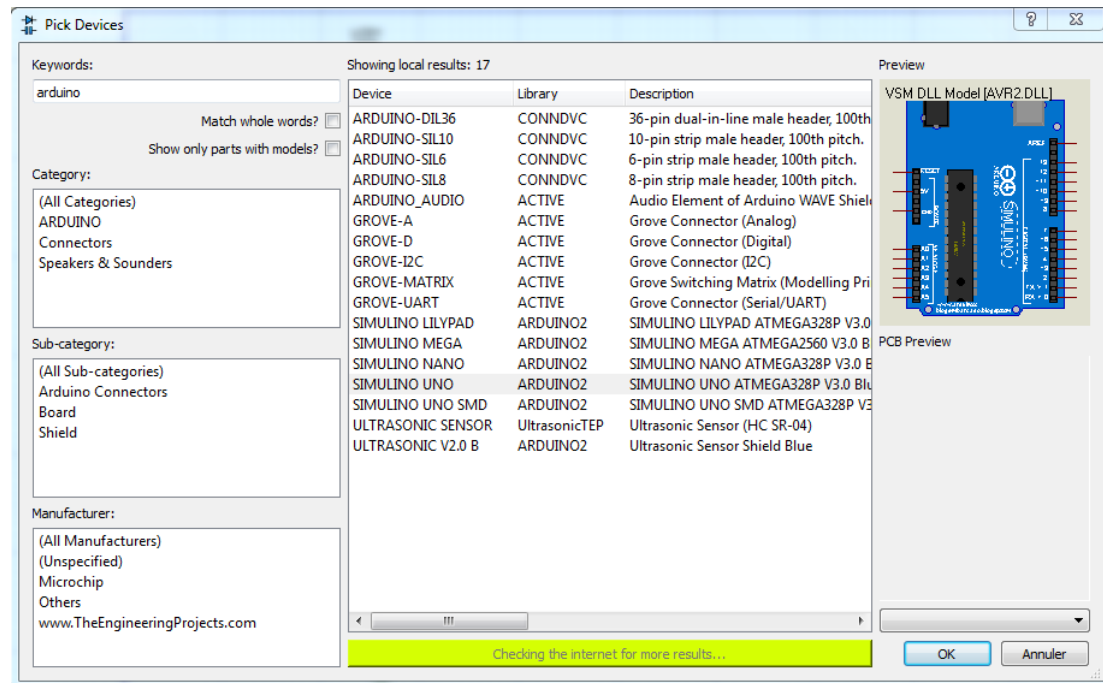


Figure 3.3 : L'intégration de la carte Arduino sous Proteus.

Maintenant, notre carte Arduino Uno est prête à être utilisée.

3.6.2- Bibliothèque de capteurs à ultrasons pour PROTEUS

Pour faire interagir le capteur à ultrasons avec l'Arduino, nous avons téléchargé une bibliothèque Capteur à ultrasons pour Proteus, disponible sur l'internet avec trois fichiers nommé "UltrasonicTEP.IDX, UltrasonicTEP.LIB, UltrasonicTEP.HEX" Maintenant, nous Mettons-les dans le référentiel Proteus.

Initialement le capteur à ultrason est hors service. Afin d'ajouter ses fonctionnalités, on double-clique sur ce capteur à ultrasons (figure 3.4) et ouvre ses propriétés et on sélectionne la section fichier du programme et on choisit le fichier UltrasonicTEP.HEX comme indique la figure.

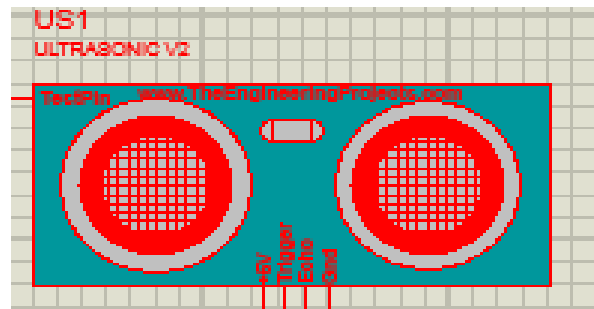


Figure 3.4 : Capteur HC-SR04 sous Proteus.

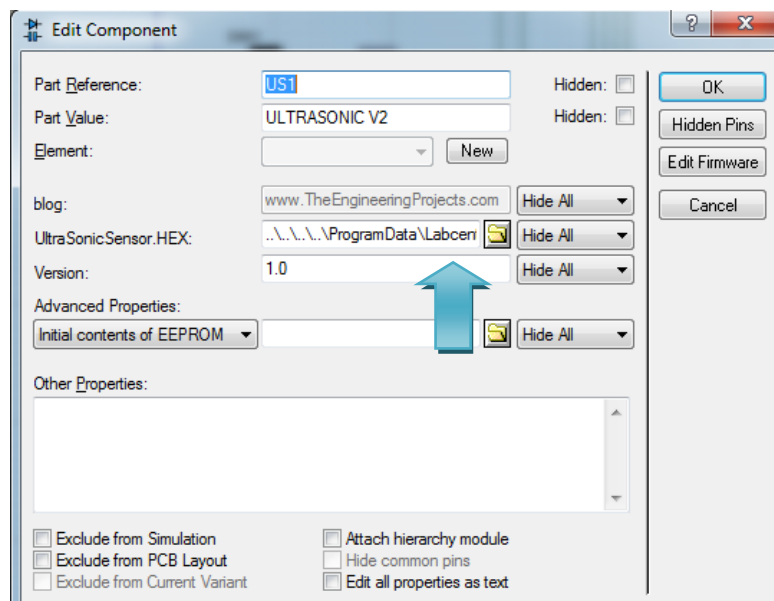


Figure 3.5 : Fichier UltrasonicTEP.HEX sous Proteus.

Maintenant, notre capteur à ultrasons est prêt à être utilisé.

Chapitre 3: Simulation Et Réalisation

Afficheur LCD	D6	3
	D7	2
	R/W	GND
	VSS pin	GND
	VCC	+5V
Servomoteur	Marron	GND
	Rouge	VCC
	Orange	9
Buzzer	VCC	10
	GND	GND

3.6.4- Résultats de la simulation

Pour démarrer la simulation, nous compilons le code du programme Arduino afin de récupérer le fichier .Hex et injectezr-le dans la carte Arduino , puisque c'est une simulation, On représente donc la cible avec une source de tension connectés avec le capteur au pin de test, d'où la résistance variable sert à indiquer la présence ou l'absence de la cible. Pour avoir les résultats de simulation nous appuyons sur le bouton démarrer et résultats seront affichés comme sons représentons sur la figure 3.7.

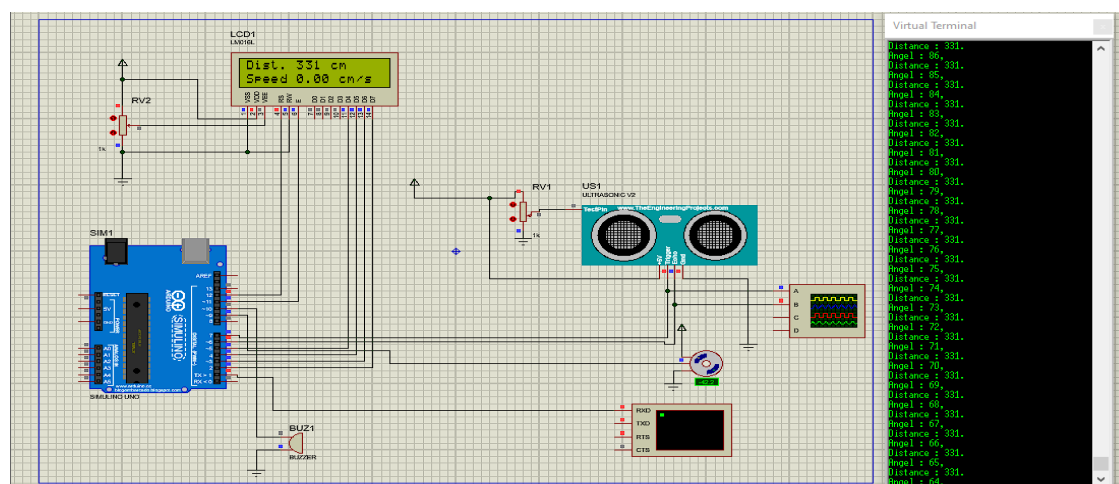


Figure 3.7 : Résultats de simulation sur le terminal virtuel.

Chapitre 3: Simulation Et Réalisation

Le terminal virtuel affiche la valeur de distance , la valeur de l'angle, et La vitesse (speed) et s'affiche aussi dans l'écran de l'LCD dans chaque durée de 1 seconde , ces valeur dépend de la résistance variable associée au capteur à ultrasons, lorsque nous augmentons ou nous diminuons la tension de la résistance variable, la valeur de la distance , l'angle et la vitesse change aussi.

Le tableau suivant (tableau 3.2) présente un résumé de différents résultats de détections obtenues dans la simulation :

Tableau 3.2 : Résultats de détection obtenues de la simulation

Numéro de détection	Distance (Cm)	L'angle (°)	Vitesse/Speed(Cm/S)
1	1	25°	0
2	110	143°	109
3	221	60°	111
4	331	79°	110
5	441	94°	0
6	551	164°	110
7	663	90°	110
8	777	1°	0
9	883	58°	0
10	994	119°	111

Un oscilloscope a été connecté au capteur pour visualiser les deux signaux Trigger et Echo (figure 3.8).

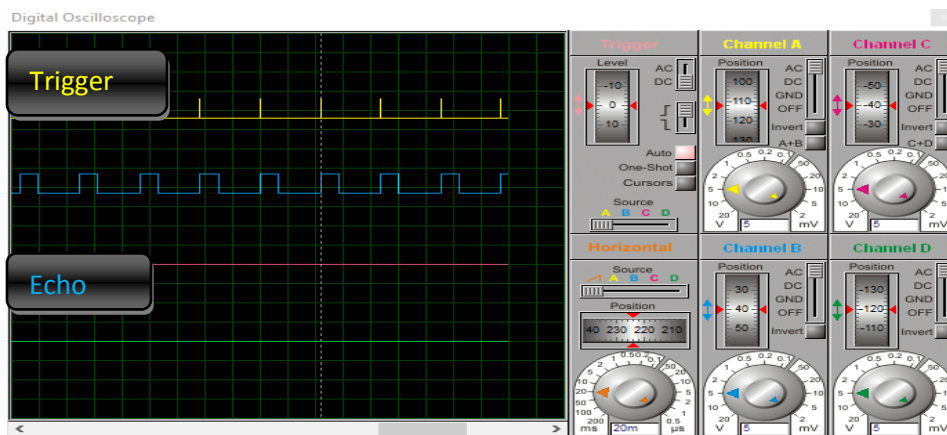


Figure 3.8 : Signal écho et signal trigger de l'ultrason

3.7- Réalisation pratique de projet

3.7.1- Composants utilisés

- (01) Carte Arduino Uno ;
- (01) Capteur sonar à Ultrasons HC-SR04 ;
- (01) SERVO Moteur ;
- (01) LCD 2*16 ;
- (01) Résistances variable ;
- (01) Buzzer .

Le circuit global de notre projet de réalisation Radar est présenté par la figure 3.9.

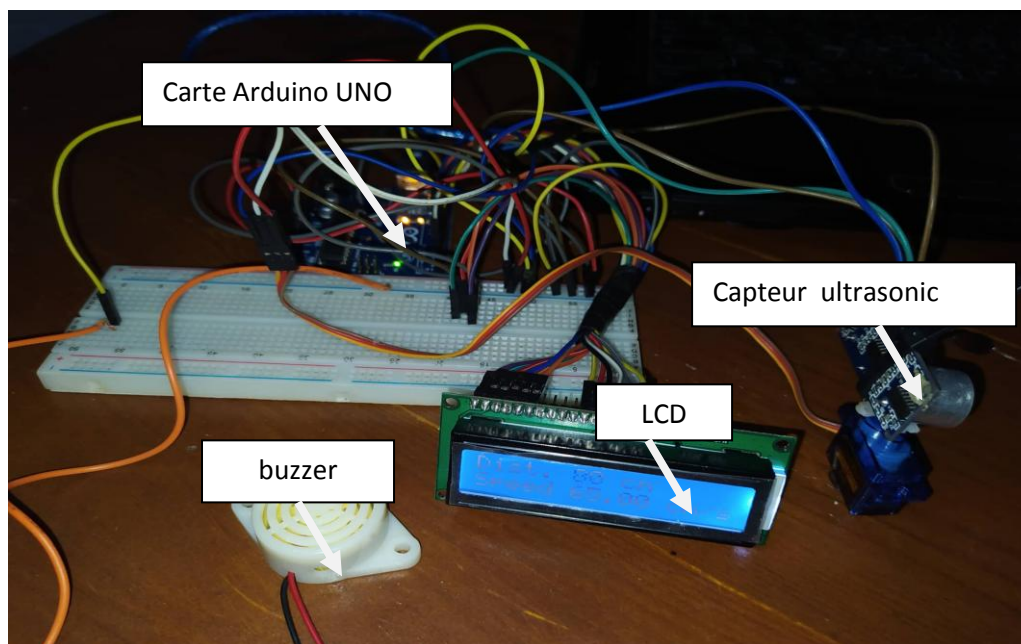


Figure 3.9 : Le circuit global de notre projet de réalisation Radar

Chapitre 3: Simulation Et Réalisation

3.7.2- Les différentes connexions et numéro des pins du circuit

Le tableau (3.3) représente les différentes connexions et numéro de pins du circuit que nous avons réalisé:

Tableau 3.3 : Broche et connexion du circuit de la pratique.

Composant	Pin composant	Pin Arduino Uno
Capteur à ultrason	VCC	+5V
	Trigger	7
	Echo	6
	GND	GND
Afficheur LCD	RS pin	12
	Enable pin	11
	D4	5
	D5	4
	D6	3
	D7	2
	R/W	GND
	VSS pin	GND
	VCC	+5V
Servomoteur	Marron	GND
	Rouge	VCC
	Orange	9
Buzzer	Rouge	13
	Noir	GND

Chapitre 3: Simulation Et Réalisation

3.7.3- Les étape de la réalisation

3.7.3.1- Alimentation du circuit

La carte Arduino, le capteur à ultrason, l'afficheur LCD, le buzzers nécessitent une alimentation stable de (+5V), pour notre travail de réalisation, nous avons alimenté notre montage à travers le port USB de l'ordinateur.

3.7.3.2- Test du fonctionnement de la carte Arduino

Dans un premier temps, nous avons testé le fonctionnement de la carte Arduino en connectant cette dernière avec le port USB de PC. Si le voyant d'alimentation est allumé, la carte est OK. Ensuite, nous cliquons sur le bouton de réinitialisation de la carte pour supprimer tous les anciens programmes et le réinitialiser, Maintenant, nous connectons les broches de chaque composant selon le tableau 3.3, et on compilons le programme sous l'IDE Arduino, puis nous téléchargeons le programme sur la carte de développement pour l'exécution via le câble USB.

La carte Arduino envoie un signal (+5V) à toutes ces broches connectées. Le déclenchement du servomoteur est très important, il fournit au capteur une action de rotation de cette façon, il peut détecter des objets en mouvement et être à moins de 180 degrés.

La carte envoie une impulsion de niveau haut de largeur (10 μ s) sur la broche TRIGGER du capteur qui régénère une série de (08) ondes ultrasonores à une fréquence de 40 KHz dans l'air de l'émetteur (inaudible pour l'homme). Les ultrasons voyagent dans l'air jusqu'à ce qu'il rencontre un obstacle et revienne au capteur dans l'autre sens via la broche ECHO, Le capteur détecte la largeur d'impulsion pour calculer la distance et la vitesse .

Le signal sur la broche ECHO du capteur reste au niveau haut pendant l'émission et la réception, permettant la mesure du temps d'aller-retour de l'ultrason et déterminant ainsi la distance à partir de laquelle la vitesse doit être calculée. La figure 3.10 montre la représentation finale du circuit d'un projet de détection radar.

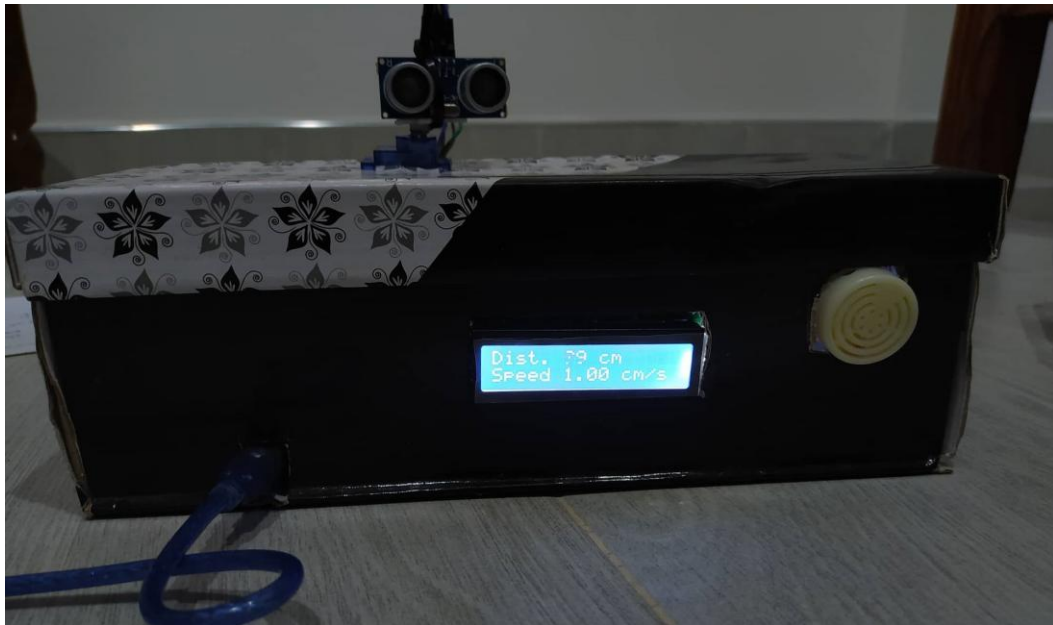


Figure 3.10 : présentation final de notre projet de réalisation du Radar de détection.

3.7.3.3 - Dessiner un radar sur processing

Pour dessiner un radar sur processing, on le fait en trois étapes :

➤ 1^{er} étape : Dessiner l'arc et les lignes du radar

Pour dessiner l'arc nous utilisons la fonction **drawRadar ()** et pour dessiner les lignes nous utilisons la fonction **drawLine()**.

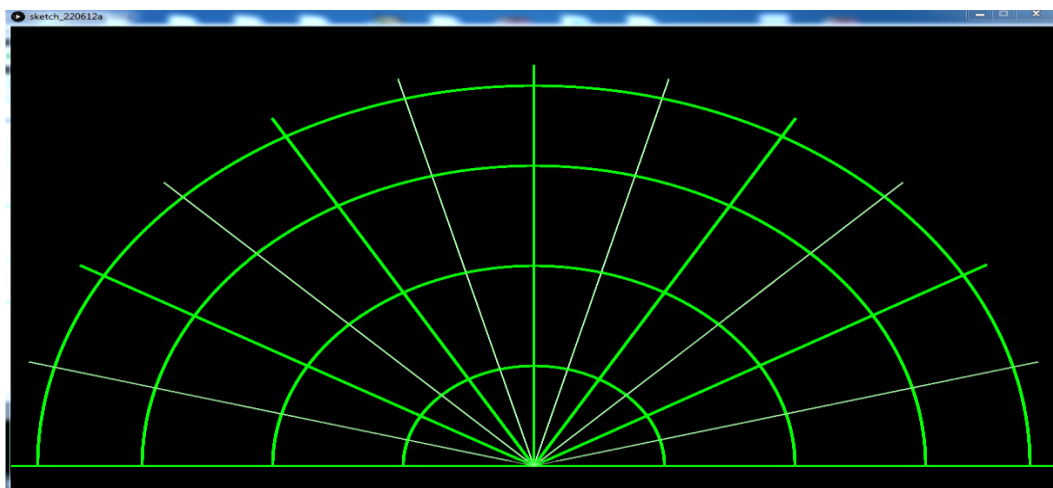


Figure 3.11 : Dessin de l'arc et les lignes du radar.

➤ 2^{ème} étape : Dessiner l'objet

Pour dessiner l'objet nous utilisons la fonction **drawObject ()** , il reçoit la distance du l'ultrason et la transforme en pixel objet .

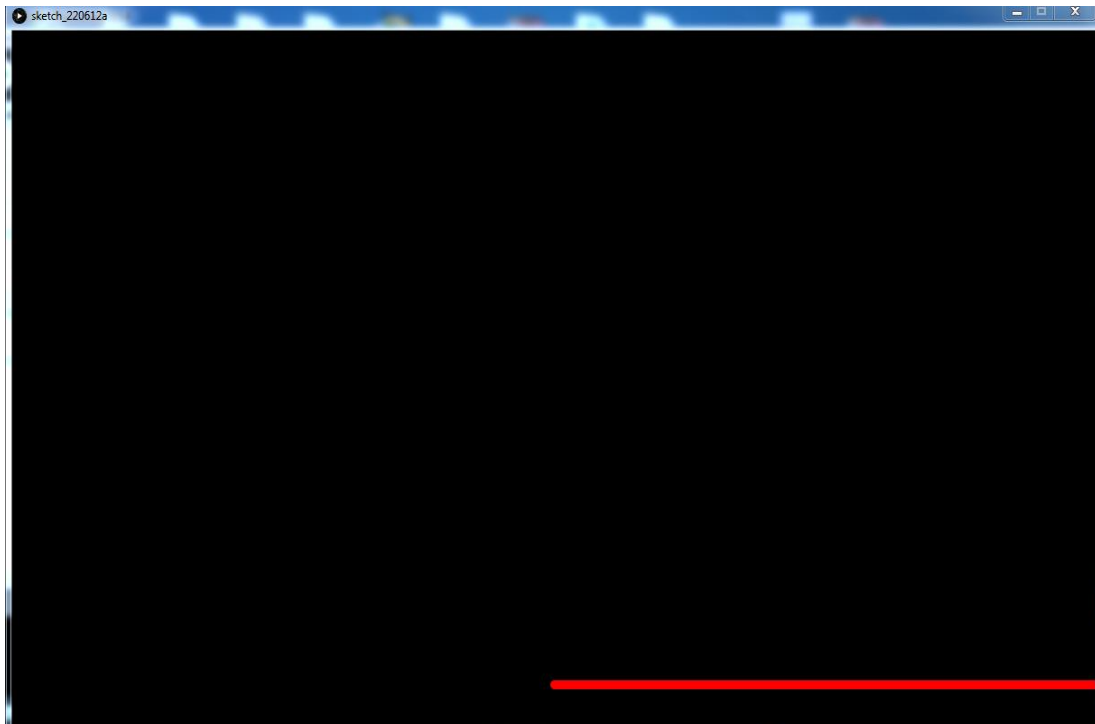


Figure 3.12 : l'objet détecté par la fonction `drawObject ()`.

➤ 3^{ème} étape : L'écriture du texte sur Processing

Pour écrire un texte sur l'écran nous utilisons la fonction `drawText()` sur des emplacements particulier .

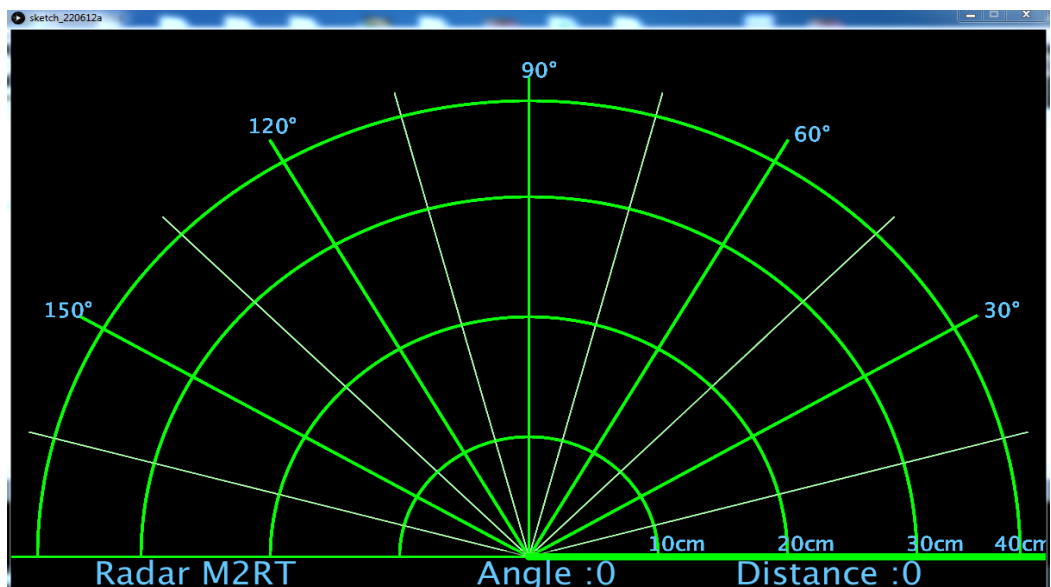


Figure 3.13 : écriture d'un texte à l'aide de la fonction `drawText()`.

Chapitre 3: Simulation Et Réalisation

➤ Simulation complète sur Processing

Sur Processing la détection d'objet est affichée en rouge, la distance mesurée ne dépasse pas 100 cm .

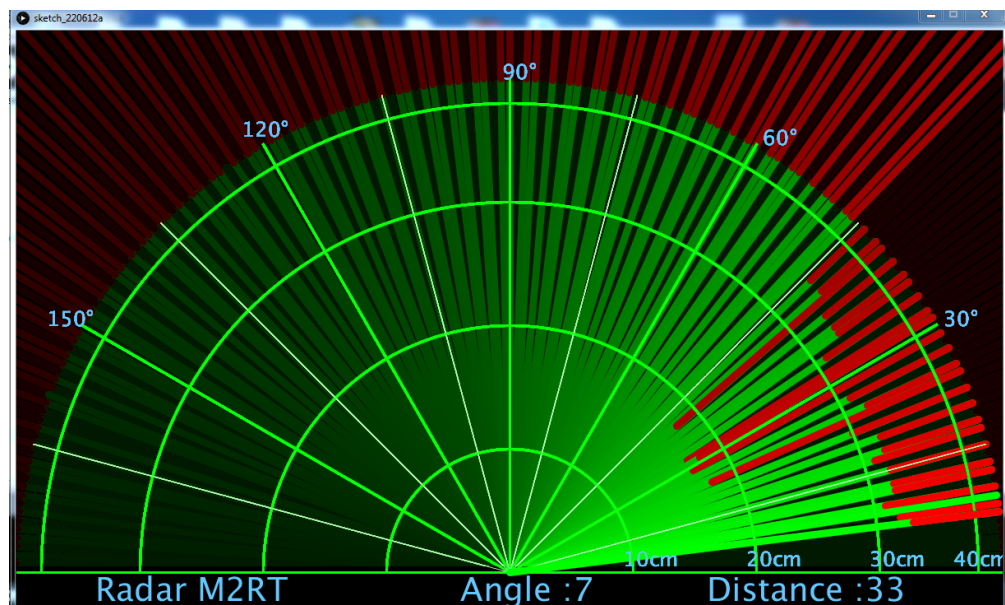


Figure 3.14 : l'affichage de la simulation complète sur processing.

3.8 - Résultats de la réalisation

Le capteur fait le balayage de 0 à 180°, il détecte des objets fixes et mobiles.

3.8.1- Visualisation de la distance,l'angle et la vitesse(speed)sur le moniteur série

La carte Arduino envoie les résultats de calcul du capteur ultrasonique au pc et en cliquant sur le moniteur série du logiciel Arduino, il affiche les résultats de détection (figure 3.15)


```
COM5
Distance = 8.
Angle = 81.
Distance = 7.
Angle = 82.
Distance = 7.
Angle = 83.
Distance = 7.
Angle = 84.
Distance = 7.
Angle = 85.
Distance = 7.
Angle = 86.
Distance = 7.
Angle = 87.
Distance = 7.
Angle = 88.
Distance = 7.
Angle = 89.
Distance = 7.
Angle = 90.
Distance = 8.
Angle = 91.
Distance = 8.
Angle = 92.
Distance = 8.
Angle = 93.
Distance = 8.
Angle = 94.
Distance = 8.
Angle = 95.
Distance = 8.
Angle = 96.
Distance = 8.
Angle = 97.
Distance = 8.
Angle = 98.
Distance =
```

Figure 3.15 : visualisation des résultats sur le moniteur série.

3.8.2 - Résultats des différents objets détecté

➤ La Mise en marche du Radar

La figure 3.16 montre que notre radar de détection démarre face à un objet cible.

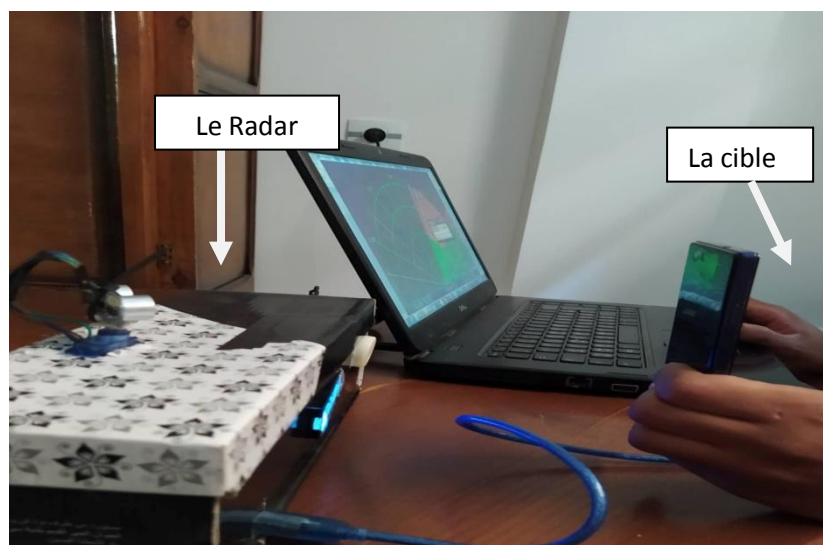


Figure 3.16 : La mise en marche de notre Radar .

Chapitre 3: Simulation Et Réalisation

➤ Détection de différents objets

Le tableau 3.4 présente des résultats de différents objets détectés dans plusieurs positions .

Tableau 3.4: Les résultats des différents objets détectés .

Numéro de détection	Distance (Cm)	L'angle (°)	Vitesse/Speed(Cm/S)
1	21	19°	0
2	25	26°	18
3	28	55°	0
4	13	81°	4
5	18	110°	2
6	23	38°	5
7	43	108°	3
8	26	83°	2
9	17	35°	4
10	36	180°	6

➤ Exemple pratique de détection d'un objet à une distance de 14 cm et angle 52° et vitesse 0 cm/s

La figure (3.17) illustre un exemple d'objet que nous avons détecté à une distance de 14cm ,un angle de 52° et une vitesse de 0 cm/s (la vitesse et de 0 cm/s par ce que l'objet est fixe) .

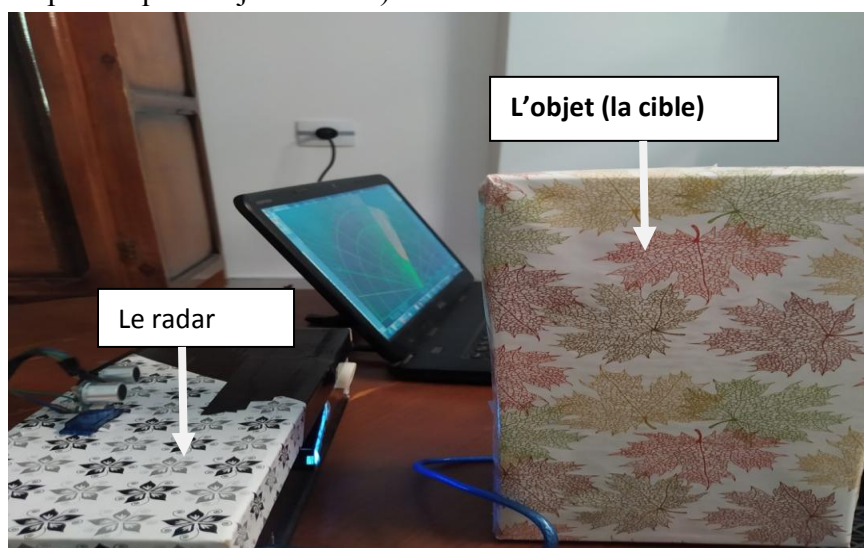


Figure 3.17 : exemple pratique de détection d'un objet fixe.

Chapitre 3: Simulation Et Réalisation

➤ L'interface processing qui illustre la détection d'objet

La figure (3.18) présente l'affichage de la distance, l'angle de l'objet à détecter.

L'affichage de la vitesse (speed) est présenté sur l'écran LCD (figure 3.19).

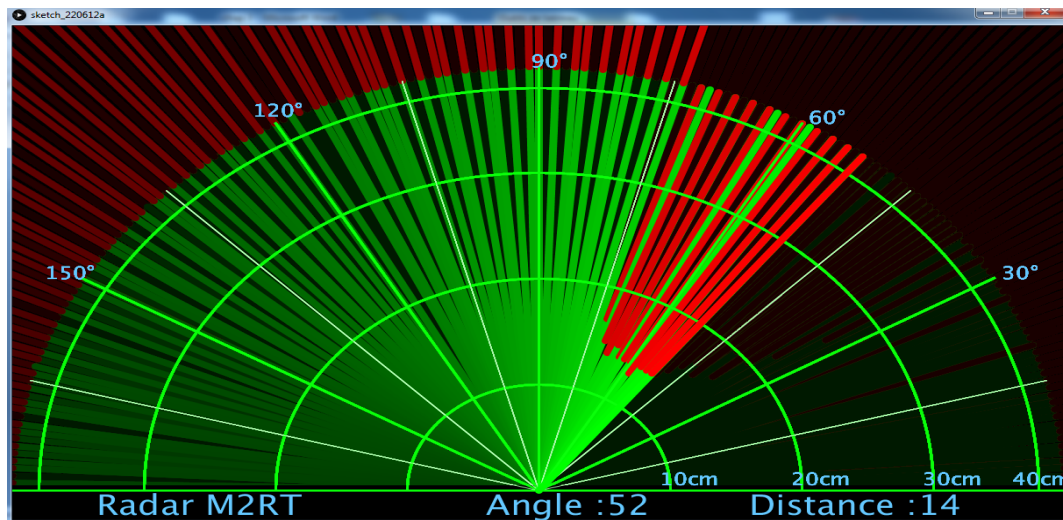


Figure 3.18 : L'affichage des résultats de détection d'objet sur l'interface Processing.



Figure 3.19 : L'affichage de la distance et la vitesse (speed) de l'objet détecté sur LCD.

Chapitre 3: Simulation Et Réalisation

3.8.3 - Interprétation des résultats

Le tableau 3.4 est un extrait des résultats pratiques que nous avons trouvés, il contient des informations (distance, direction et vitesse) pour de différents objets fixes et mobiles détectés.

Le déclenchement du buzzer indique que la distance d'objet détecté est inférieure ou égal à une distance de 100 cm. Lorsque des cibles fixes ou mobiles sont présentes, des variations importantes de puissance et de fréquence sont réfléchies vers le récepteur.

Le signal d'écho, présente un signal basse fréquence, lorsque nous plaçons une cible en face de capteur, on remarque ensuite un changement de fréquence, donc une détection d'objet fixe.

3.9 – conclusion

Dans ce chapitre, nous avons vu la détection radar à l'aide de cartes électronique Arduino Uno qui mesure la distance, la direction et la vitesse des objets via un capteurs à ultrasons et dans des directions de 0 à 180° à l'aide de servomoteurs.

Nous avons trouvé lors de la simulation d'ISIS Proteus, La distance dépend de la résistance variable associée au capteur à ultrasons, lorsque on augmente ou on diminue la valeur de la tension, on changeant la résistance variable, un changement de distance est obtenue (augmentation ou diminution).

Dans la partie implémentation, nous avons fait des mesures de distance avec un capteur en mouvement rotationnel de 0 à 180°, dans le cas d'une cible fixe ou mobile. Les résultats obtenus (Distance, direction et vitesse) sont affichés sous l'interface Processing et l'afficheur LCD.

Conclusion
Générale

Conclusion Générale

Conclusion générale

Dans ce mémoire nous avons cherché à concevoir et réaliser un radar de détection ultrason. Ce radar est composé d'un capteur Ultrason HC_SR04 pour la détection des obstacles, d'une carte à microcontrôleur Arduino, pour le traitement des données et le calcul de la distance, et la vitesse pour les objet mobile et fixe, d'un servomoteur permettant la rotation du capteur ultrason en vue de couvrir une zone d'obstacles de 0° à 180°, d'un afficheur LCD pour l'affichage de la distance et la vitesse (speed) de l'obstacle et on a aussi utilisé le logiciel processing pour afficher la distance et l'Angel sur le pc.

Les notions de base du radar sont exploitées dans notre applications qui va porter sur l'utilisation des capteurs ultrasoniques pour l'émission et la réception, donc les ondes utilisées ici sont de nature sonore et non pas électromagnétiques.

La conception d'un système radar passe par deux étapes principales : L'interconnexion des différents composants utilisés et la programmation de la carte Arduino. Une fois terminé, le système radar tourne dans une plage de 180 degrés, et il calcule la distance, l'angle et la vitesse de l'objet à moins ou égale à 100 cm. Le radar conçu est contrôlé par un programme implémenté sur une carte Arduino, qui comporte la facilité des tâches de conception. Plusieurs tests ont été effectués avec le radar mise en place, Les résultats se sont avérés décisifs.

En outre l'utilisation du logiciel graphique Processing permettra la visualisation d'obstacles détectés sur ordinateur en temps réel.

Comme perspectives de ce travail, des améliorations sur le système radar peuvent être réalisées tel que l'utilisation d'un capteur de distance plus performant pour la détection d'objet sur une plus grande portée. L'utilisation d'un matériel plus sophistiqué donnera des résultats plus fiables et plus précis.

Enfin nous espérons que ce mémoire sera utile et enrichi par les futures promotions.

Références

Références

Références:

- [1] **Laurent Déjean** , «Détection de petites cibles marines en milieu côtier par radar naéroporté », Thèse doctorat de Telecom Bretagne, 10 novembre 2009.
- [2] **Roger MEVEL** , «Généralités sur le RADAR» , utl-morlaix, URL : https://www.ummt0.dz/dspace/bitstream/handle/ummt0/6345/SlimaniJuba_SaidOuamerAghiles.pdf?sequence=1&isAllowed=y
- [3] **M. Skolnik** , « Introduction to radar systems», Mc-Graw-Hill, New-York, 1981
- [4] **Philippe Goy**, «Détection d'obstacles et de cibles de collision par un radar FMCW aéroporté». Thèse de doctorat de l'université de Toulouse, France. mardi 18 décembre 2012.
- [5] **Christian Wolff** ,«<Les Principes du Radar>>, radartutorial.eu, URL: <https://www.radartutorial.eu/01.basics/Mesure%20de%20l'angle%20de%20site.fr.html>
- [6] **DJEDOU Nacéra & BOUKERN Dounya**, « Étude et réalisation d'un RADAR de détection », mémoire de master 2 en télécommunications, Université Aboubakr Belkaïd, Tlemcen. Algérie. 19/06/ 2017.
- [7] **SLIMANI Juba & SAID OUAMER Aghiles** ,« Etude et réalisation d'un radar détections à base de la carte Arduino », Mémoire de Master Professionnel Electronique Industriel, Université Mouloud Mammeri De Tizi-Ouzou 2018
- [8] **KRATTOU Mohamed Riad**, « Étude De La Détection Radar Dans Un Milieu Homogène », mémoire de master 2 en télécommunications, l'université de Tlemcen, Algérie. Promtion2012/2013.
- [9] **Mathieu DUMONT, Loick ATTAGNANT, Killian JACQUET**. Arduino [en ligne]. Publié le 13/01/2017.
- [10] **DJAFRI Menad & CHELOUCHE Djalal**, « Étude et réalisation d'une Carte Arduino», mémoire de master 2 en Electronique, Université A.MIRA de Bejaia. Algérie. 20/06/2016.
- [11] **Decouverte_arduino**, techmania.fr, URL: http://www.techmania.fr/arduino/Decouverte_arduino.pdf

Références

- [12] **Carte Arduino** , generationrobots.com , URL : <https://www.generationrobots.com/blog/fr/bien-choisir-sa-carte-arduino-ce-nest-pas-si-difficile/>
- [13] **A.Ahmed Azi**, <<contribution au développement au diagnostic de performance des MPPT dans la chaîne de conversion photovoltaïque>>. Mémoire magister. Université de M'sila, Algérie, 2014.
- [14] **schema de la carte arduino uno** ,URL : <http://cours.education/dufacilitateur/2017/10/17/petite-carte-et-grand-champ-des-possibles/>
- [15]**B.Cottenceau**, « Carte ARDUINO Uno Microcontrôleur ATmega328 ». Microcontrôleurs EI3 Option AGI.
- [16]**Mohanad Mahdi Abdulkareem**, Qusay Adil Mohammed Muhanned Mahmood Shakir, « A Short Range Radar System», Electric and Electronic Engineering Department, University of Turkish Aeronautical Association Ankara, Turkey ,2016.
- [17] **ATMEGA328P**, riton-duino.blogspot.com, URL: <http://riton-duino.blogspot.com/2019/07/montage-et-chargement-dun-atmega328p.html> , jeudi 25 juillet 2019.
- [18] **Jean-Luc**. Les écrans LCD alphanumériques [en ligne].Le 3 mars 2015.
- [19] **Xavier HINAULT** (2010) Référence Arduino français.
- [20]**B.Cottenceau**, « Carte ARDUINO Uno Microcontrôleur ATmega328 ». Microcontrôleurs EI3 Option AGI
- [21] **La carte Arduino Uno** , mon-club-elec.fr,URL : http://www.mon-club-elec.fr/pmwiki_reference_arduino/pmwiki.php?n=Main.MaterielUno
- [22] **BENRABAH Youcef Islam & BOUHINI Mouhammed Yacine**, « Étude et réalisation d'un Radar électronique « Radar à ultrason», mémoire de master 2 en Télécommunications ,Université Djilali Bounaama Khemis Miliana. Algérie. Promotion 2016/2017.

Références

[23] **Frank Ervin**, « Arduino-based Object Detection System », Project Final Report, April 27, 2010.

[24] **Détection par ultrasons** , << Détection par ultrasons- principes de base et application>> , fr.jf-parede.pt ,URL : <https://fr.jf-parede.pt/ultrasonic-detection-basics-application>

[25] **HADDOUCH Yamina & BESTANI Djedjiga** , << Conception et réalisation d'un radar ultrason>> , Mémoire de fin d'étude de Master Académique Spécialité Instrumentation , Université Mouloud Mammeri de Tizi-Ouzou Faculté de Génie Electrique et d'Informatique ,2017 /2018.

[26] **Servo-Moteur**, wiki.mchobby.be ,<< Servo-Moteur>> URL : <https://wiki.mchobby.be/index.php?title=Servo-Moteur>.

[27] **HC-SR04** , gotronic.fr ,<< HC-SR04 - Module de détection aux ultrasons>> , URL : <https://www.gotronic.fr/pj2-hc-sr04-utilisation-avec-picaxe-1343.pdf?fbclid=IwAR3SNYo8b6W0G5ywxU536TWUYJSg3l5a0vLm2pHZwAaS83SHHkjihxteLU>

[28] **Cfaury**, arduino.blaisepascal.fr,<< Capteur de distance à Ultrasons>>, janvier 2016,URL : <https://arduino.blaisepascal.fr/capteur-de-distance-a-ultrasons/>

[29] **itechnofrance** , itechnofrance.wordpress.com , <<Utilisation du servomoteur SG90 avec l' Arduino>> , 05/05/2013, URL : <https://itechnofrance.wordpress.com/2013/05/05/utilisation-du-servomoteur-sg90-avec-larduino/>

[30] **julien terrier**, www.developpez.net,<< Train d'impulsion servo-moteur>>,06/10/2016,URL : <https://www.developpez.net/forums/d1605571/general-developpement/programmation-systeme/embarque/arduino/train-d-impulsion-servo-moteur-sm-s4303r/>

[31] **MOUSSAOUI Zineb & SAHBI Meriem** , <<Simulation et réalisation d'un radar de détection d'objets à base d'arduino>>, Mémoire de Fin d'Études En vue de l'obtention du diplôme de MASTER ACADEMIQUE Spécialité : Systèmes des Télécommunications, Université Echahid Hamma Lakhdar El-Oued Algérie , 2019/2020

Références

[32] **Reas and Ben Fry**, «Processing: A Programming Handbook for Visual Designers and Artists (Second Edition) », The MIT Press. Publié :December 2014.

[33] **Courrs processing.pdf**, www.flossmanualsfr.net, <<processing>>, Publié: 2017-11-29, License : GPLv2+ ,URL :

<https://www.flossmanualsfr.net/media/files/processing/processing.web.pdf>

[34] **Angona Biswas & Sabrina Abedin & Md. Ahasan Kabir**,<< Moving Object Detection Using Ultrasonic Radar with Proper Distance, Direction, and Object Shape Analysis>>,Article Journal of Information Systems Engineering and Business Intelligence, Vol.6, No.2, October 2020,URL:

<http://ejournal.unair.ac.id/index.php/JISEBI>.