

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE
UNIVERSITE MOULOU D MAMMERI DE TIZI-OUZOU



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

Mémoire
MASTER PROFESSIONNEL
Spécialité : *Electronique Industriel*

Présenté par :

SLIMANI Juba

SAID OUAMER Aghiles

Thème :

**Etude et réalisation d'un radar détections à base de la carte
Arduino**

Devant les membres du jury :

Président :ATTAF .Y

Examineur :HAMEG.S

Encadreur/rapporteur: Mme. REBAH née SLIMANI Kahina

Soutenu le : 27 / 09 / 2018

Remerciements

Nous remercions tout d'abord par excellence sa grandeur « LE BON DIEU » qui nous a donné le courage et la patience tout au long de notre vie.

Nous remercions notre promotrice Mme.REBAH nee Slimani kahina pour ses précieuses orientations, son aide et ses conseils tout au long de ce projet.

Nous remercions également le président de jurys ainsi que les membres de jurys d'avoir accepté dévalué et juger notre travail.

Nos remerciements les plus chaleureux vont à nos chères FAMILLES pour leurs encouragements, leur patience et leur grand soutien durant toutes ces années d'études.

Enfin, un remerciement tout particulier à toutes les personnes qui ont contribué de près ou de loin à la réalisation de ce projet.

Dédicaces

Je dédie ce travail :

*A mes chers parents qui m'ont toujours soutenue tout au
long de mes études*

A mes très chers frères et sœurs

A mes très chers amis ;

A toutes ma famille ;

*A mon binôme et meilleur amis « juba » ainsi à toute sa
familles ;*

A tous ceux que j'aime ;

A tous ceux qui m'aiment ;

A tous mes ami(e)s et mes collègues

Aghiles Said ouamer

Dédicaces

Je dédie ce travail :

*A mes chers parents qui m'ont toujours soutenue tout au
long de mes études*

*A mes très chers frères « Messoud ,Hamou et
Djugurtha ;*

A ma très chère sœur« Kahina» ;

A mes très chère petite nièce « Anais et Amillia » ;

A mon chers neveux «Amir » ;

A mes très chers amis «Anis ,Hosni,et Mohand » ;

A toutes ma famille ;

*A mon binôme et meilleur amis « Aghiles » ainsi à toute
sa familles ;*

A tous ceux que j'aime ;

A tous ceux qui m'aiment ;

A tous mes ami(e)s et mes collègues

Juba Slimani

Juba Slimani

Table des figures

Figure 1.1: Principe d'un système Radar	3
Figure 1.2 : Mesure de la direction.....	4
Figure 1.3 : Classification des Radars.....	6
Figure 1.4 : Génération d'échos.....	7
Figure 1.5 : Composition d'un Radar à impulsion.....	8
Figure 1.6 : Relation entre les puissances émises et réfléchies.....	11
Figure 1.7 : Différents types de bruit traités par un récepteur	13
Figure 1.8 : Spectre du signal impulsionnel et du bruit « Blanc ».....	13
Figure 1.9 : Variation du SNR en fonction de largeur de bande.....	14
Figure 2.1: Carte Arduino.....	17
Figure 2.2 : Constitution de la carte Arduino UNO.....	19
Figure 2.3 : Microcontrôleur ATmega328.....	20
Figure 2.4: Sources de l'Alimentation de la carte Arduino UNO.....	21
Figure 2.5 : Capteur Ultrasons HC-SR04.....	23
Figure 2.6 : Le diagramme temporel.....	24
Figure 2.7 : Montage HC-SR04 avec Arduino.....	25
Figure 2.8 : Distance supérieur à 3 mètres.....	25
Figure 2.9 Surface peut réfléchissante de l'objet.....	26
Figure 2.10 : Objet trop petit.....	26
Figure 2.11 : Afficheurs LCD : (a) (16x2), (b) (20x4).....	26
Figure 2.12 : Connecteur de l'afficheur LCD.....	27
Figure 2.13: Servomoteur à rotation angulaire.....	29

Figure 2.14 : Caractéristiques du servomoteur à rotation angulaire (Micro-Servo).....	30
Figure 2.15: Câble de commande pour servomoteur.....	31
Figure 2.16 : montage servomoteur avec Arduino.....	31
Figure 2.17 : Plateforme Arduino.....	32
Figure 2.19 : Structure générale du programme (IDE Arduino).....	33
Figure 2.20. : Diagramme temporel de HC-SR04.....	36
Figure 2.21. Communication entre les composants du projet.....	36
Figure 3.1: Capteur HC-sr04 sous Proteus.....	33
Figure 3.2 : Rajout du fichier UltrasonicTEP.HEX sous Proteus.....	39
Figure 3.3 : Le circuit global de notre simulation.....	40
Figure 3.4: Injection du fichier .HEX sous proteus.....	41
Figure 3.5: Résultats de simulation sur le terminal virtuel.....	42
Figure 3.6 : Schéma synoptique de notre Radar.....	43
Figure 3.7 : Algorithme principale de détection d'un objet.....	44
Figure 3.8: l'arc et les lignes du Radar.....	45
Figure 3.9: l'objet détecté par la fonction drawObject().....	45
Figure 3.10 : écriture d'un texte à l'aide de la fonction drawText().....	46
Figure 3.11: Simulation complète sur Processing.....	46
Figure 3.12. Circuit final de notre projet de réalisation du Radar de détection.....	48
Figure 3.13: Visualiser la distance et l'angle dans moniteur série.....	49
Figure 3.14 : Image global de la mise en marche du Radar.....	50
Figure 3.15 : Détection de l'objet 09.....	51
Figure 3.16 : Interface du Processing avec détection (Présence de l'objet 09)	51

Liste des tableaux

Tableau 2.1 : Caractéristiques de la Carte Arduino UNO.....	18
Tableau 2.2 : Spécifications des capteurs ultrason HC-SR04.....	24
Tableau 2.3 : Nomenclature du connecteur de l’afficheur LCD.....	27
Tableau 3.1 : Broche et connexion de notre circuit électronique.....	47
Tableau 3.2 : Les résultats des différents objets détectés par notre radar.....	52

Sommaire

Introduction générale.....1

Chapitre 1: Aperçu sur les radars

1.1. Introduction.....3

1.2. Le Radar.....3

 1.2.1. Définition.....3

 1.2.2. Principe du Radar.....3

1.3. La cible.....5

1.4. Classification des Radars.....5

 1.4.1. Le Radar primaire.....6

 1.4.2. Le Radar à impulsions.....6

 1.4.3. Radar à ondes continues.....6

 1.4.4. Le Radar secondaire.....7

1.5. Radar à impulsions.....7

 1.5.1 Description.....7

 1.5.2. Composition d'un Radar à impulsion.....8

 1.5.2.1. L'émetteur.....8

 1.5.2.2. Le duplexeur.....8

 1.5.2.3. Le récepteur9

1.6. Equation du Radar9

 1.6.1. Equation Radar.....9

 1.6.2. Influence des pertes sur la portée.....12

 1.6.3. Portée maximale du Radar en présence du bruit thermique.....12

 1.6.3.1. Bruit à la réception.....12

 1.6.3.2. Filtrage non optimal.....13

 1.6.3.3. Filtrage optimal (Filtre adapté)14

1.7. Radar à diversité de fréquences.....14

1.8. Applications du Radar.....15

1.9. Conclusion.....15

Chapitre 2 : Etude et mise en œuvre du radar

2.1. Introduction.....	16
2.2. Description Matériel	16
2.2.1. Définition du module Arduino.....	16
2.2.2. la carte Arduino Uno	17
2.2.2.1. Les composants de la Carte Arduino	18
2.2.2.2. Avantage de carte Arduino Uno.....	22
2.2.3. Capteur Ultrasons HC-SR04.....	22
2.2.3.1. Fonctionnement du capteur Ultrasons HC-SR04.....	23
2.2.3.2. Caractéristiques et spécification du capteur.....	24
2.2.4. Module Afficheur LCD.....	26
2.2.5. Servomoteur.....	29
2.3. Logiciels de programmation.....	31
2.3.1. Le logiciel de programmation du module Arduino.....	31
2.3.2. Logiciel IDE de PROCESSING.....	34
2.3.3. Logiciel PROTEUS ISIS.....	34
2.3.4. Communication entre la carte Arduino, logiciel Proteus et logiciel Processing....	34
2.4. Fonctionnement du radar de détection.....	35
2.5. L'organigramme du projet.....	36
2.6. Conclusion.....	37

Chapitre 3 : Simulation et réalisation du radar de détection

3.1. Introduction.....	38
3.2. Les différentes étapes du projet.....	38
3.2.1. Partie simulation du Radar de détection.....	38
3.2.1.1. Circuit global de la simulation.....	40
3.2.1.2. Injection du fichier .HEX sous proteus.....	41
3.2.1.3. Schéma synoptique de notre Radar.....	43
3.2.1.4. Algorithme principale de détection d'un objet.....	44
3.2.1.5. Dessiner un radar su r processing.....	45
3.2.2. Partie réalisation de notre Radar détection.....	47
3.2.2.1. Les différentes connexions et numéro des pins du circuit.....	47

3.2.2.2. Principe de fonctionnement du circuit.....	48
3.2.2.3. Visualisation de la distance et de l'angle sur le moniteur série.....	49
3.2.2.4. Résultats des différents objets détectés.....	50
3.2.2.5. Interprétations des résultats trouvés.....	52
3.3. Conclusion.....	53
Conclusion générale.....	54

Introduction générale

Le Radar est l'un des plus sophistiqués technologies de l'observation de la terre et de l'exploration planétaire. Les radars météorologiques ont vu le jour après la seconde guerre mondiale. Aujourd'hui, les systèmes radar sont aussi très largement répandus dans des applications civiles telles que la surveillance du trafic aérien, la météorologie ou encore la sécurité automobile. Le radar est donc aujourd'hui un système indispensable et très largement utilisé qu'il sert à surveiller un territoire ou pour éviter des collisions

Le terme radar est issu de l'acronyme anglais de (Radio Detection And Ranging) désigne un système qui diffuse une onde électromagnétique dans une portion de l'espace, et reçoit les ondes réfléchies par les objets qui s'y trouvent, permettant de détecter leurs existence et déterminer certaines caractéristiques de ces objets tel que la position, l'altitude, la vitesse et parfois la forme de ces objets.

Le but principal de ce projet et l'étude et la réalisation d'un radar de détection en utilisant la carte Arduino, d'où trois objectifs ont été visés : faire une étude générale sur la théorie des radars, regrouper suffisamment d'informations sur une grande catégorie des cartes de prototypage (Arduino), réaliser un système Radar pour la détection des objets à l'aide des composants disponibles.

Le système obtenu est composé de deux parties principales ; la première se compose de la partie matérielle, par laquelle l'opération de détection est effectuée et la deuxième partie celle du logiciel où les données entrantes en provenance du matériel sont analysées et affichées.

La partie matérielle est composée : d'un capteur ultrasonique qui a pour objectif la détection des objets et de la distance, d'un servomoteur pour tourner le capteur et l'envoi de l'angle de rotation pour le logiciel et d'un afficheur LCD, tous ces composants sont contrôlés par un micro-contrôleur à base de l'environnement Arduino. La partie logicielle est conçue à l'aide de la plate-forme de traitement par lequel une application graphique est conçue pour afficher un éventuel objet détecté, la position et la distance.

Notre travail que nous présentons dans cette thèse est structuré en trois chapitres.

-Dans le premier chapitre nous donnerons un aperçu sur les Radars.

-Dans le second chapitre, nous présentons une étude et une mise en œuvre d'un Radar de détection.

-Dans le troisième chapitre, nous présenterons une simulation et une réalisation du Radar de détection.

-Une conclusion générale et des perspectives font l'objet de la dernière partie.

1.1. Introduction

Les premières applications de la radioélectricité furent les télécommunications puis la radionavigation mais dès le début du siècle, des précurseurs envisagèrent la possibilité de détecter la présence d'objets métalliques par l'utilisation d'ondes électromagnétiques. Cette nécessité de détecter les objets sans la participation de l'objet lui-même répondait au besoin de sécurité de la navigation et de l'anticollision, notamment dans le domaine des transports maritimes. C'est cependant le besoin militaire de la défense aérienne et maritimes qui fut le principal moteur de cette technique nouvelle à partir des années trente. Ce premier chapitre sera consacré d'une part, aux généralités sur les Radars à savoir, leur définition, leurs variétés et d'autre part à la présentation de différent type de Radars.

1.2. Le Radar

1.2.1. Définition

Le terme radar est issu de acronyme anglais de (Radio Detection And Ranging) désigne un système qui diffuse une onde électromagnétique dans une portion de l'espace, et reçoit les ondes réfléchies par les objets qui s'y trouvent, permettant de détecter leurs existence et déterminer certaines caractéristiques de ces objets tel que la position, l'altitude, la vitesse et parfois la forme de ces objets. Ces données permettent au Radar de renseigner l'utilisateur, mais aussi d'éliminer un grand nombre des objets indésirables pour ne conserver que les « cibles » intéressantes [1].

1.2.2. Principe du Radar

Le principe de base des systèmes radars peut être résumé en quatre étapes : émission et propagation d'une onde électromagnétique, rétrodiffusion de cette onde par une cible sous forme d'écho de l'onde, réception et analyse du signal reçu par le radar.

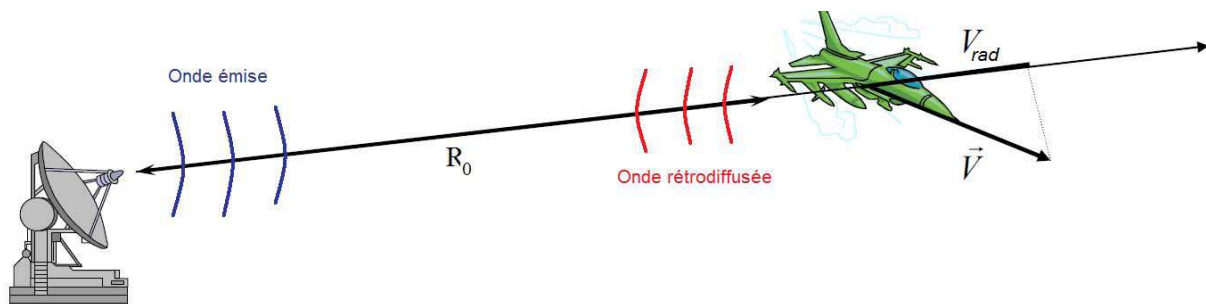


Figure 1.1 : Principe d'un système RADAR

- **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 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 [2][3].

$$\tau = \frac{2R_0}{c} \quad (1)$$

La mesure de ce retard τ permet alors de déterminer la distance R_0 de la cible.

- **Mesure de la direction**

La mesure de la direction est déterminée grâce à l'angle entre la direction du nord et celle de la cible (Azimut), cet angle est obtenu par la directivité de l'antenne. En mesurant cette direction dans laquelle l'antenne est pointée à l'instant où elle reçoit l'écho, on peut déterminer l'azimut et aussi le site de la cible (l'altitude). La précision de la mesure de ces angles dépend de la directivité de l'antenne. Pour une fréquence émise donnée (Ou une longueur d'onde définie), la directivité d'une antenne est en fonction de ses dimensions propres [6].

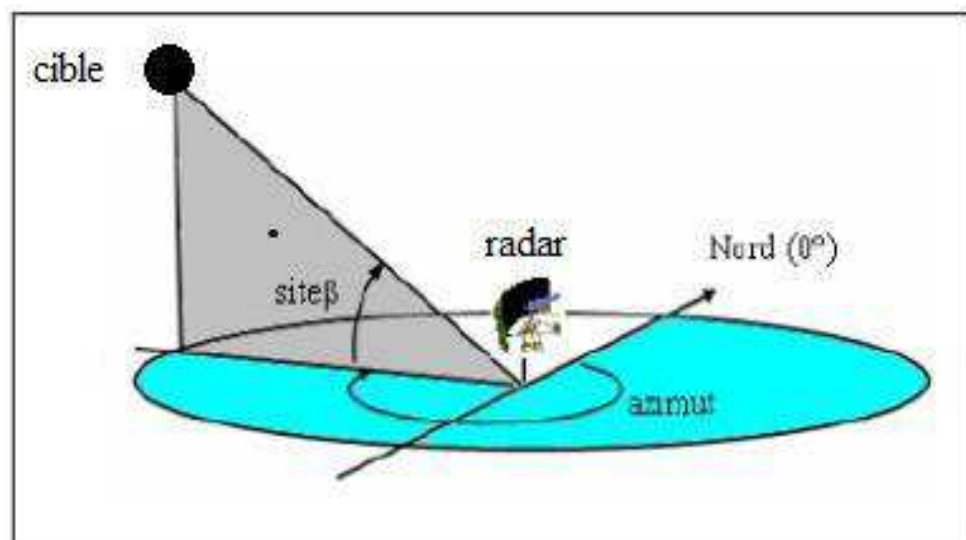


Figure 1.2 : Mesure de la direction

- **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.

- **L'effet Doppler :** Lorsque la cible est mobile par rapport au radar, son retard évolue au cours du temps

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

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 1.1 arrive en 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 [3]. Ce phénomène porte le nom d'effet Doppler. Le décalage en fréquence sur la porteuse, appelée fréquence Doppler, s'écrit :

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

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

1.3. La cible

Une cible est tout objet qui interfère avec l'onde émise, et réfléchit une partie de l'énergie vers le Radar, elle se comporte comme une antenne de forme complexe. L'énergie émise dans la direction du Radar est fortement fluctuante et dépend énormément de l'orientation de la cible par rapport au Radar.

1.4. Classification des Radars

La classification basée sur la fonction principale du Radar est illustrée dans la figure suivante:

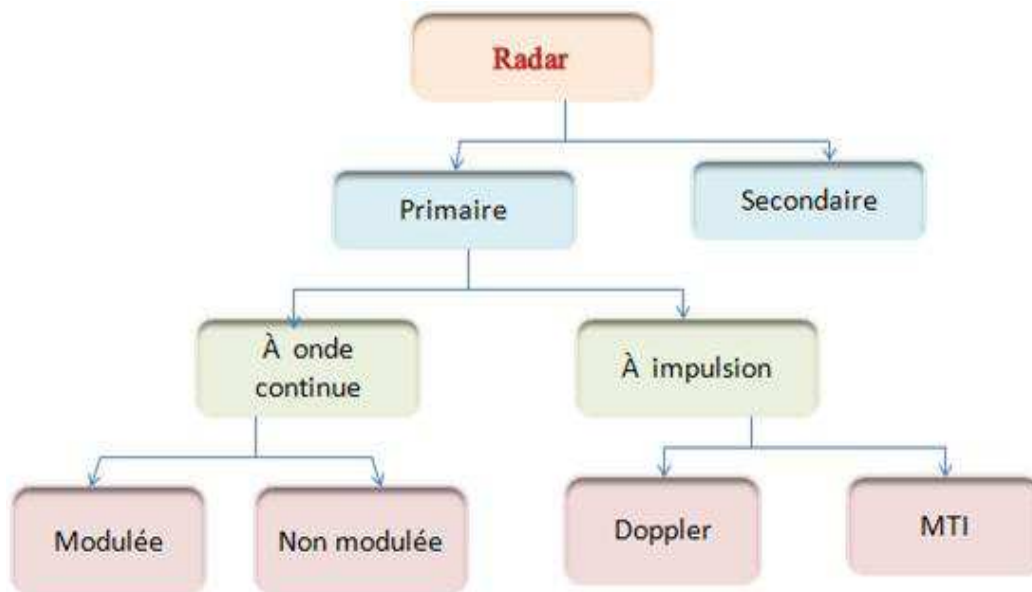


Figure 1.3 : Classification des Radars

1.4.1. Le Radar primaire

Un radar primaire émet des signaux hyperfréquences qui sont réfléchis par les cibles. L'énergie réfléchie ou les échos sont en outre traités pour extraire des informations utiles [4].

1.4.2. Le 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 le 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 [5].

1.4.3. Radar à ondes continues

Les Radars CW (Continuous Wave), le signal est émis de façon continue. Les Radars utilisant ce type d'émission sont destinés à mesurer la vitesse en exploitant l'effet doppler.

- **Radars à ondes continues modulées :** ces Radars émettent sans interruption un signal hyperfréquence, l'écho est donc reçu et traité continuellement.
- **Radars à ondes continues non modulées :** Les Radars CW non modulés émet une

Seul fréquence à amplitude constante. Le Radar CW qui transmet une puissance non modulée, ne peut qu'en utilisant l'effet Doppler.[5]

1.4.4. 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) [6].

1.5. Radar à impulsions

1.5.1 Description

Ce type de Radar utilise des impulsions à hautes fréquence, qui se réfléchies sur toute cible, créant ainsi une onde de retour susceptible d'être décelée par un récepteur adapté à ce signal. Chaque impulsion de durée très brève de l'ordre de quelque microsecondes se propage dans l'atmosphère à la vitesse de la lumière ($C = 3.10^8$ m/s). Une partie de ce signal est réfléchié 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 des impulsions et la fréquence de répétition PRF.

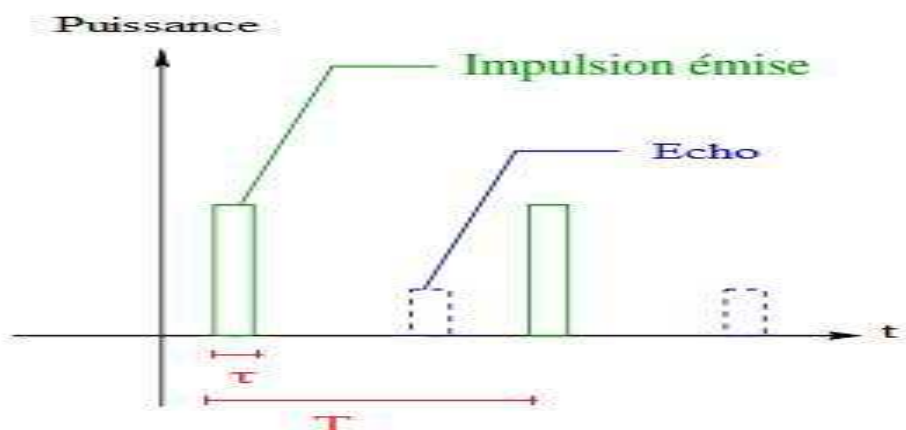


Figure 1.4 : Génération d'échos.

1.5.2. Composition 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.

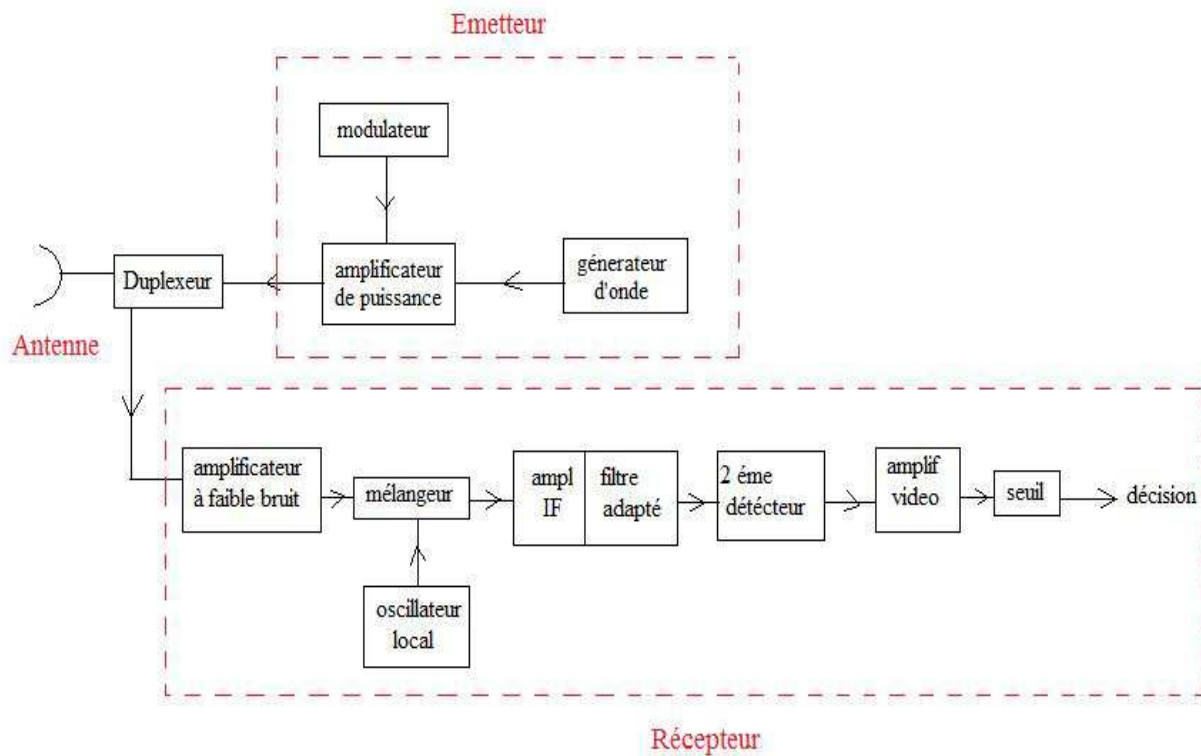


Figure 1.5 : Composition d'un Radar à impulsion

1.5.2.1. 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.

1.5.2.2. 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.

1.5.2.3. 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

1.6.1 Equation 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.

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 uniforme dans toutes les directions. Les zones d'égales 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 surface sphérique. Cela revient à dire que la densité de puissance, dans une direction donnée, diminue lorsque la distance de l'émetteur augmente [5].

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

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

S_u : Densité de puissance «omnidirectionnelle» [w/m^2]

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 du à la « concentration » de la puissance émise dans une seule direction.

La formule permettant de calculer la densité de puissance dans la direction du gain maximum est la suivante :

$$S_g = S_u \cdot G \quad (5)$$

S_g : Densité de puissance « directive » [w/m^2]

S_u : Densité de puissance « omnidirectionnel » [w/m^2]

G : gain d'antenne

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

$$S_u = \frac{P_s}{4\pi R_1^2} \cdot G \cdot \sigma \quad (6)$$

P_r =puissance réfléchi [w]

P_s = puissance émise [w]

σ = surface équivalente radar [m^2]

R_1 = distance antenne - cible [m]

D'une façon simplifiée nous pouvons considérer la cible comme un émetteur (du signal réfléchi). La puissance réfléchi 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 (5) pour déterminer la densité de puissance S_e atteignant l'emplacement de l'antenne du radar.

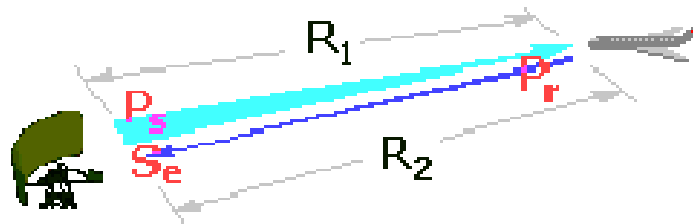


Figure 1.6: Relation entre les puissances émises et réfléchies

$$S_e = \frac{P_r}{4\pi R_2^2} \quad (7)$$

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 :

$$P_e = \frac{P_r}{4\pi R_2^2} \cdot A \cdot K_a \quad (8)$$

$$P_e = \frac{P_s \cdot G}{(4\pi)^2 \cdot R_1^2 \cdot R_2^2} \cdot A \cdot K_a \quad (9)$$

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 :

$$P_e = \frac{P_s \cdot G^2 \cdot \lambda^2 \sigma}{(4\pi)^3 \cdot D^4} \quad \text{En watt} \quad (10)$$

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

$$D = \left[\frac{P_s \cdot G^2 \cdot \lambda^2 \sigma}{(4\pi)^3 \cdot P_e} \right]^{\frac{1}{4}} \quad \text{En mètre} \quad (11)$$

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

1.6.3.1 Bruit à la réception

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.

➤ Figure de bruit

On peut considéré 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. La figure de bruit d'un récepteur est définie par l'équation [6]:

Avec :

S_i : Bruit à l'entrée d'un récepteur réel S_0 : Bruit à la sortie d'un récepteur réel
 N_i : Bruit à l'entrée d'un récepteur idéal N_0 : Bruit à la sortie d'un récepteur idéal

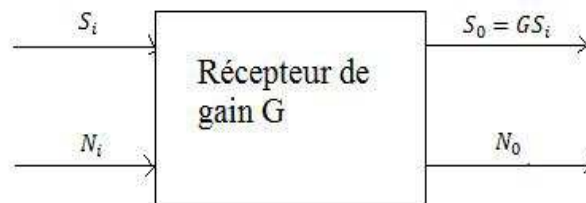


Figure 1.7 : Différents types de bruit traités par un récepteur

1.6.3.2 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 sa facilité de mesure car elle nécessite une connaissance complète de la caractéristique de réponse $H(f)$ [6].

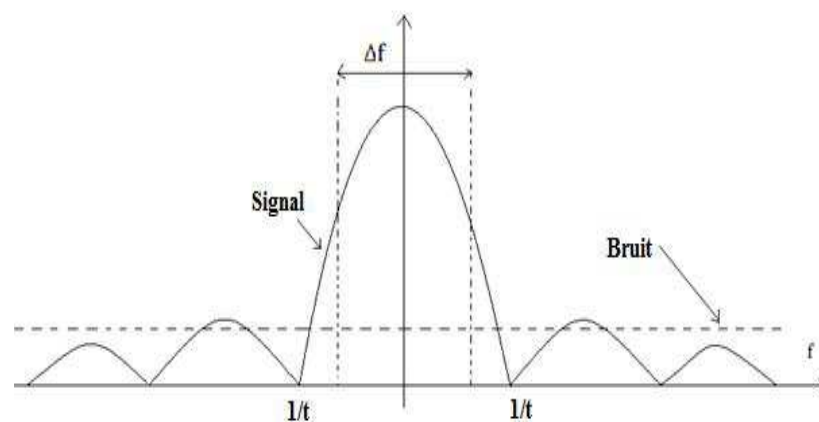


Figure 1.8 : Spectre du signal impulsionnel et du bruit « Blanc »

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].

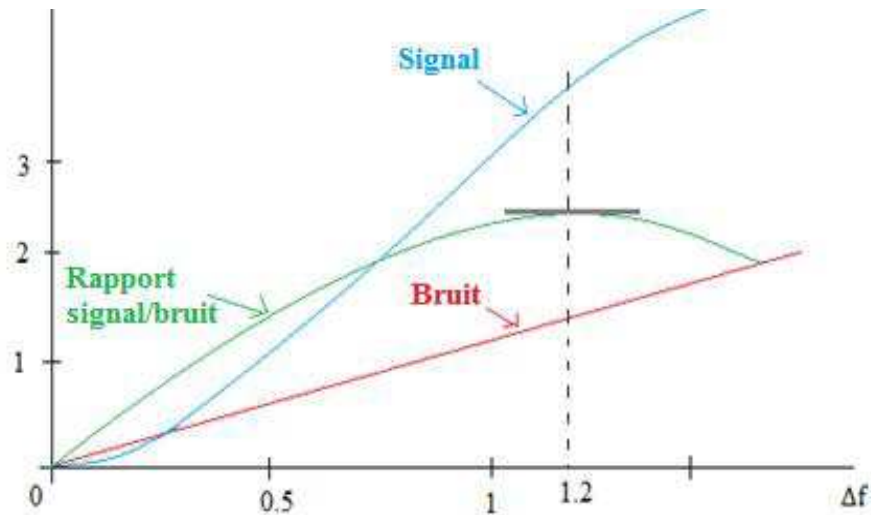


Figure 1.9 : Variation du SNR en fonction de largeur de bande

1.6.3.3 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 [5].

On applique à l'entrée :

$$x(t)=s(t)+n(t), t \in [0, T] \quad (12)$$

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]:

$$SNR(t_0)^{max} = \frac{2 \cdot E}{N_0} \quad (13)$$

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 ou 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 soit indépendante [6].

1.8. Applications du Radar

Les applications du Radar sont nombreuses et variées les plus courantes sont :

- La météorologie : détection des nuages et des pluies.
- La défense : détection d'avion et de missiles.
- La navigation aérienne : contrôle du trafic aérien, détection et poursuite d'aéronefs et surveillance de pistes.
- La navigation maritime et fluviale : contrôle du trafic dans les ports.
- Contrôle de la circulation routière : mesure de la vitesse des véhicules.

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.

2.1. Introduction

Le Radar de détection est un instrument de détection des cibles utiles, définies par les spécifications de besoin des utilisateurs, donc avec rejet des signaux parasites après identification. Des paramètres discriminants sont déterminés pour séparer signaux utiles et parasites.

L'objectif de ce chapitre est de présenter les composants matériels et logiciels utilisés pour la conception et la mise en œuvre du radar de détection, l'explication de son principe de fonctionnement et la représentations de son organigramme.

2.2. Description Matériel

Pour réaliser le Radar de détection nous avons besoin :

- D'une carte Arduino Uno.
- D'un capteur ultrasons HC-SR04.
- D'un servomoteur.
- D'un afficheur LCD.

2.2.1 Définition du module Arduino

Arduino, et son récent synonyme Genuino, est une marque qui couvre des cartes 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 sont publiés en licence libre. Cependant, certains composants, comme le microcontrôleur par exemple, ne sont pas sous licence libre [7].

Une carte Arduino est une interface programmable capable de piloter des capteurs et des actionneurs afin de simuler ou créer des systèmes automatisés. Elle peut stocker un programme et le faire fonctionner. La carte reçoit des informations analogiques ou numériques sur ces entrées.

Le modèle UNO de la société ARDUINO est une carte électronique dont le cœur est un microcontrôleur ATMEL de référence ATmega328. Le microcontrôleur ATmega328 est un microcontrôleur 8bits de la famille AVR dont la programmation peut être réalisée en langage C. [8]

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.

2.2.2. La Carte Arduino Uno

Les unités de traitement existent sous différentes formes. Les microcontrôleurs en font partie. Leurs couts et les fonctionnalités qu'ils proposent dépendent du cahier de charge du système à concevoir. Nous utiliserons la carte Arduino Uno [8]

La carte Arduino Uno fondée sur un microcontrôleur ATmega328 de 8bits cadencé à 16 MHz qui peut être programmée pour analyser et produire des signaux électriques. Aux niveaux de programmation on peut contrôlée l'Arduino en langage C ou C++. Et contient un boot loader qui permet de modifier le programme sans passer par un programmeur.



Figure 2.1: Carte Arduino

2.2.2.1. Composantes de la carte Arduino Uno

La carte Arduino Uno est basée sur un Microcontrôleur ATmega328 cadencé à 16 MHz. C'est la plus récente et la plus économique carte à microcontrôleur [9]. Les caractéristiques techniques de la carte Arduino UNO sont présentées dans le tableau 2.1

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)

Tableau 2.1 : Caractéristiques de la Carte Arduino UNO

Les différents composants constituant la carte Arduino UNO sont représentés sur la figure 2.2.

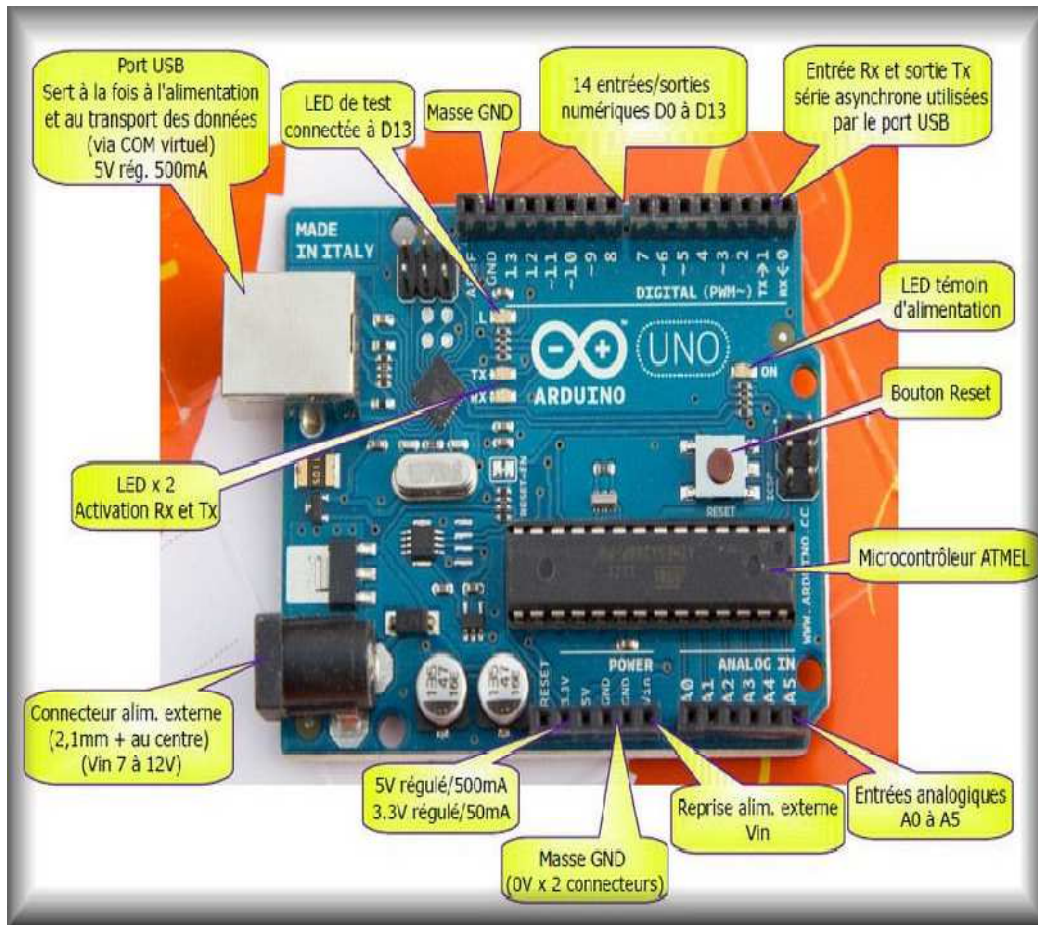


Figure 2.2 : Constitution de la carte Arduino UNO.

➤ 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 [9][16].

La figure (2.3) montre deux types de microcontrôleur ATmega328, qu'on trouve sur les cartes Arduino.

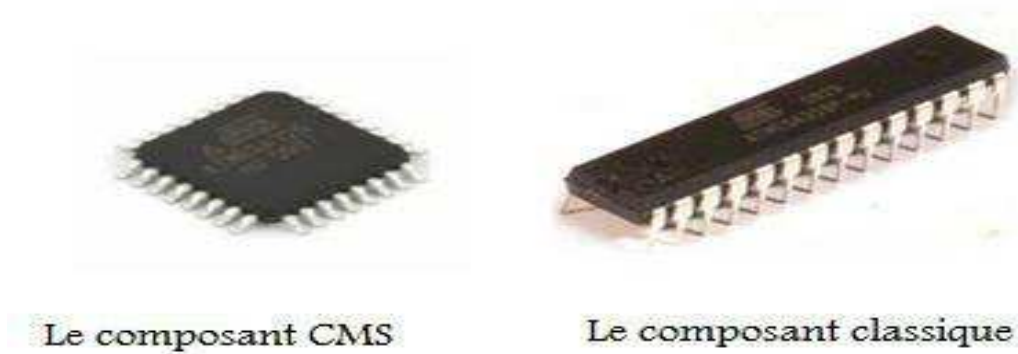


Figure 2.3 : Microcontrôleur ATmega328

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 [9].
- **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 [9]. L'alimentation externe peut être soit un adaptateur secteur (Pouvant fournir typiquement de 3V à 12V sous 500mA) ou des piles.
- **Les broches d'entrée/sortie** : L'Arduino UNO possède 14 broches d'entrée/sortie digitale (Numérotées des 0 à 13), ces broches peuvent 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.

La carte UNO dispose de (06) six 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 fonction `analogRead()` du langage Arduino. Les broches analogiques peuvent être utilisées en tant que broches numériques : Elles sont numérotées en tant que broches numériques de 14 à 19 [8].

- **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[9].
- **Oscillateur à quartz**
- **Interrupteur de réinitialisation**

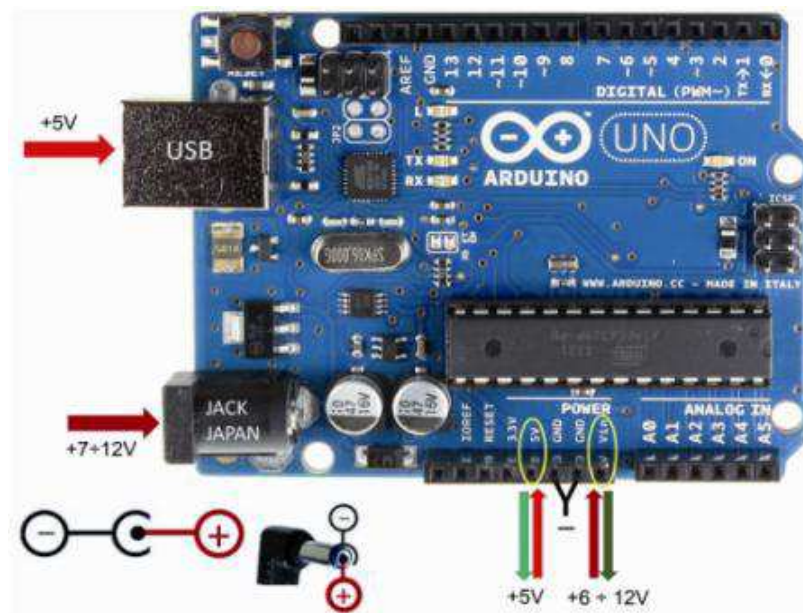


Figure 2.4: Sources de l'Alimentation de la carte Arduino UNO.

➤ 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 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 [8] [15].

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 [9].

2.2.2.2. Avantage de carte 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.

2.2.3. Capteur Ultrasons HC-SR04

Un capteur est une interface entre un processus physique et une information manipulable. Il ne mesure rien, mais fournit une information en fonction de la sollicitation à laquelle il est soumis [11][17].

Le **HC-SR04** est un capteur de distance utilise les ultrasons, Il offre une excellente plage de détection avec des mesures stables. Son fonctionnement n'est pas influencé par la lumière du soleil ou des matériaux sombres, bien que des matériaux comme les vêtements puissent être difficiles à détecter. Ce capteur fonctionne avec une tension d'alimentation de 5 volts, dispose d'un angle de mesure de 15° environ et permet de faire des mesures de distance entre 2 centimètres et 3 mètres avec une précision de 3mm.

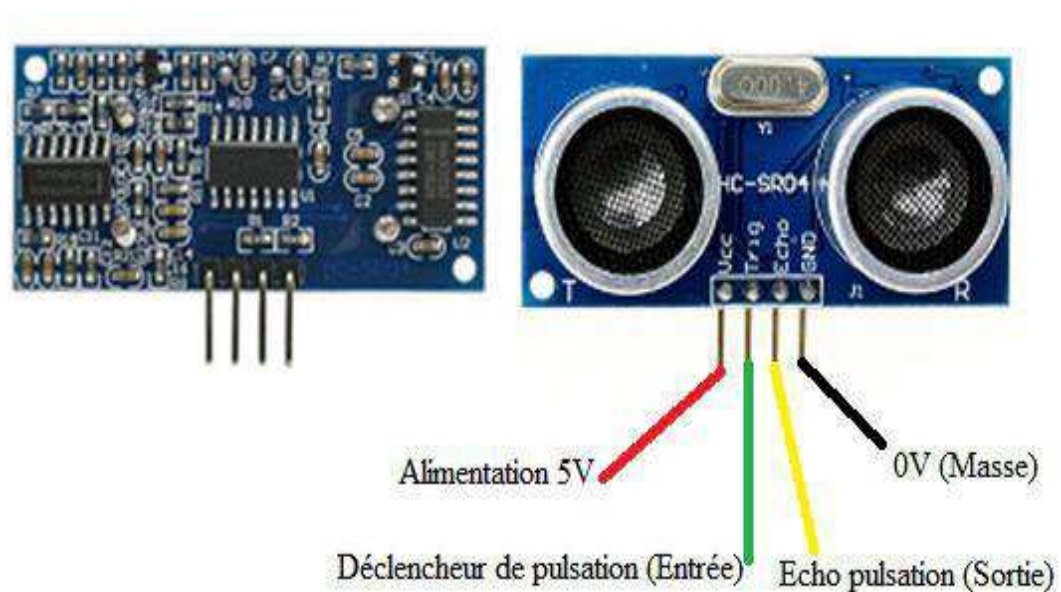


Figure 2.5 : Capteur Ultrasons HC-SR04

2.2.3.1 Fonctionnement du capteur Ultrasons HC-SR04

➤ **Le principe de fonctionnement consiste à :**

- 1) Envoyer un signal numérique à l'état haut sur l'émetteur pendant 10 μ s.
- 2) Le capteur envoie automatiquement 8 impulsions d'ultrasons à 40 kHz et détecte les signaux qui reviennent.
- 3) Si le signal revient, la durée de l'état haut du signal reçue correspond au temps entre l'émission des ultrasons et leur réception.

Calcul de la distance : $\text{Distance} = (\text{temps à l'état haut signal reçu} * \text{vitesse du son}) / 2$
 (vitesse du son dans l'air : 340 m/s).

➤ **Le diagramme temporel**

Le diagramme temporel est affiché ci-dessous. Il faut fournir un signal à l'état haut pendant 10 μ 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.

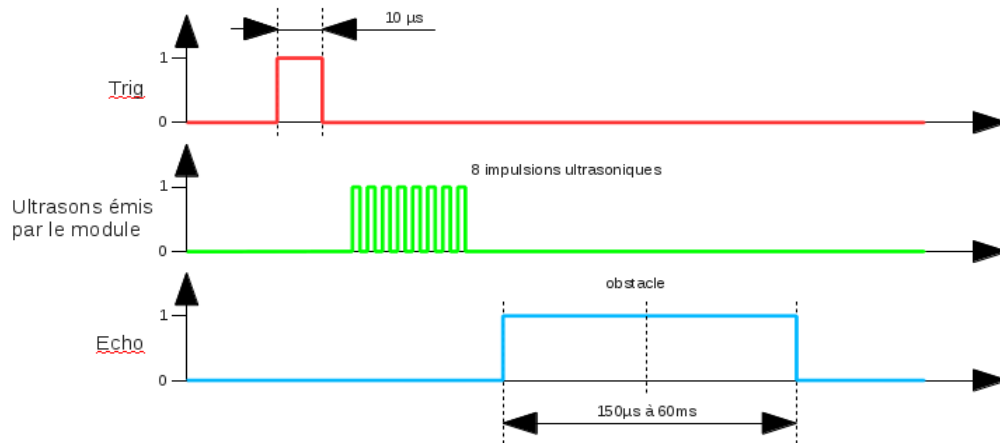


Figure 2.6 : Le diagramme temporel.

2.2.3.2. Caractéristiques et spécification du capteur

Les caractéristiques en détail du Capteur Ultrasons HC-SR04 sont présentées dans le tableau ci-dessous [11].

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 µs TTL impulsion
Dimensions L x W x H	45 mm x 20 mm x 15mm
Poids	8.5g

Tableau 2.2 : Spécifications des capteurs ultrason HC-SR04.

- **Montage capteur HC-SR04 avec Arduino :**

Le module HC-SR04 possède 4 broches :

- 1) **Vcc** : Alimentation +5 V DC sert à alimenter le circuit du HC-SR04.
- 2) **Trig** : est l'entrée qui nous permet de déclencher l'émission d'un ultrason.
- 3) **Echo** : est la sortie qui se met à 5 V pendant une durée égale à l'aller-retour de l'ultrason, (Echo output).
- 4) **GND** : Masse de l'alimentation.

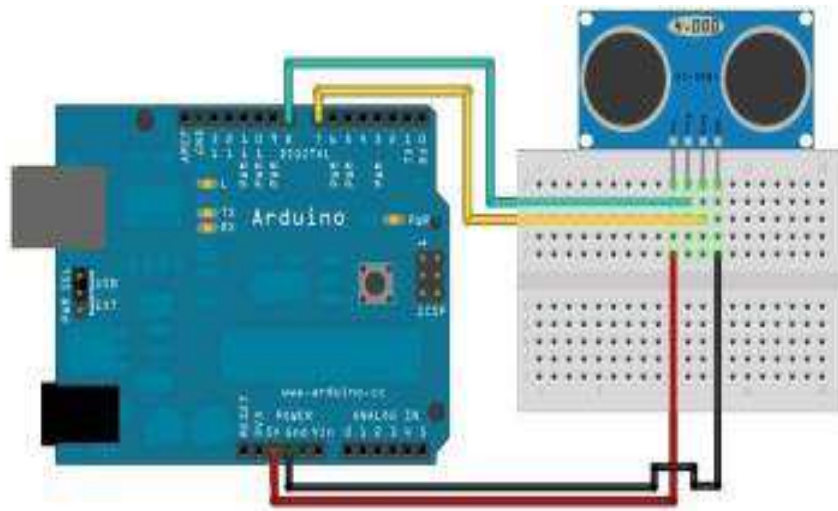


Figure 2.7 : Montage HC-SR04 avec Arduino.

- **Considérations pratiques pour l'utilisation de ce capteur**

Le Capteur ne peut pas mesurer avec précision la distance d'un objet :

- 1) Qui est supérieure à 3 mètres

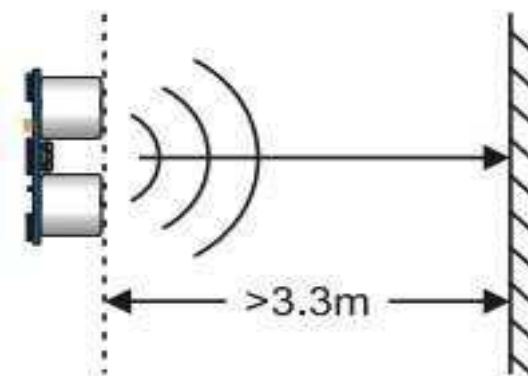


Figure 2.8 : distance supérieur à 3 mètres.

- 2) Qui à une surface réfléchissante à un angle faible de sorte que le son ne sera pas renvoyé vers le capteur

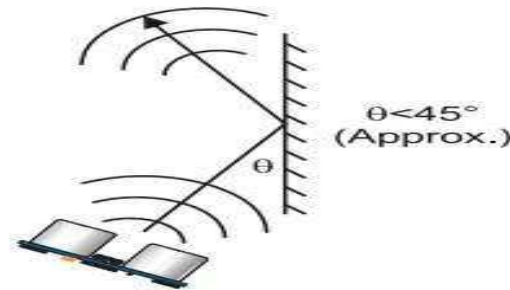


Figure 2.9 Surface peut réfléchissante de l'objet.

- 3) Qu'est trop petit pour tenir compte du retour de son écho vers le capteur

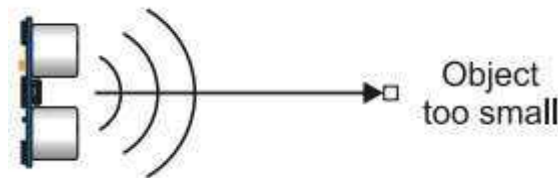


Figure 2.10 : Objet trop petit.

2.2.4. Module Afficheur LCD

Les afficheurs LCD sont devenus indispensables dans les systèmes techniques qui nécessitent l'affichage des paramètres de fonctionnement. Ces Afficheurs permettent d'afficher des lettres, des chiffres et quelques caractères spéciaux. Vu de l'extérieur, les écrans LCD alphanumériques sont essentiellement caractérisés par leur taille [12].

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 2.11.



Figure 2.11 : Afficheurs LCD : (a) (16x2), (b) (20x4)

- **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 ci-dessous donne la nomenclature des broches de ce connecteur :

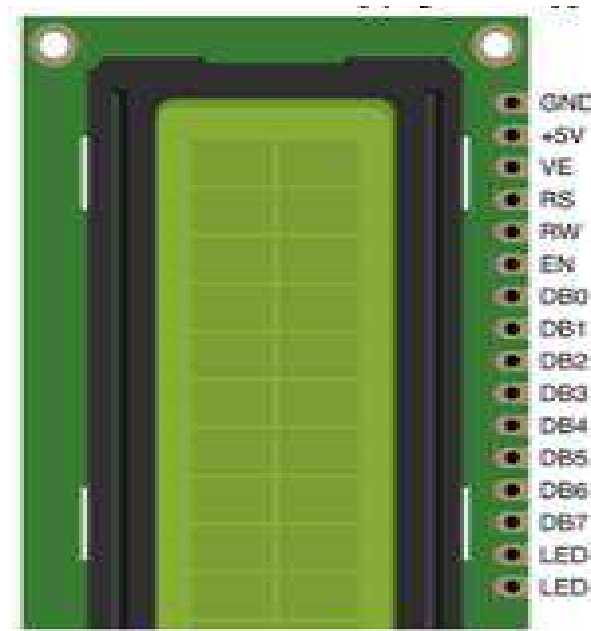


Figure 2.12 : Connecteur de l'afficheur LCD

Le tableau suivant présente les spécifications des broches et leur rôle [8] :

Numéro de la broche	Désignation	Description
1	GND	Masse 0V
2	VCC	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-	Anode (-) du rétro-éclairage

Tableau 2.3 : Nomenclature du connecteur de l'afficheur LCD.

- **Communication avec l'afficheur LCD**

L'afficheur LCD peut fonctionner en mode 8 bits ou en mode 4 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 [12].

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 datasheet 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.

2.2.5. Servomoteur

Un servomoteur (souvent abrégé en « servo », provenant du latin *servus* qui signifie « esclave ») est un moteur capable de maintenir une opposition à un effort statique et dont la position est vérifiée en continu et corrigée en fonction de la mesure. C'est donc un système asservi. Le servomoteur intègre dans un même boîtier, la mécanique (moteur et engrenage), et l'électronique, pour la commande et l'asservissement du moteur. La position est définie avec une limite de débattement d'angle de 180 degrés, mais également disponible en rotation continue.

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 2.13: Servomoteur à rotation angulaire.

➤ Commander le servomoteur

Le principe du moteur est assez simple, il suffit d'envoyer une impulsion, le temps que durera cette impulsion déterminera l'angle du servomoteur. Ce temps d'impulsion est de quelques millisecondes et doit être répété à un intervalle régulier.

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.

➤ **caractéristiques du servomoteur sont présentées dans la figure ci-dessous.**

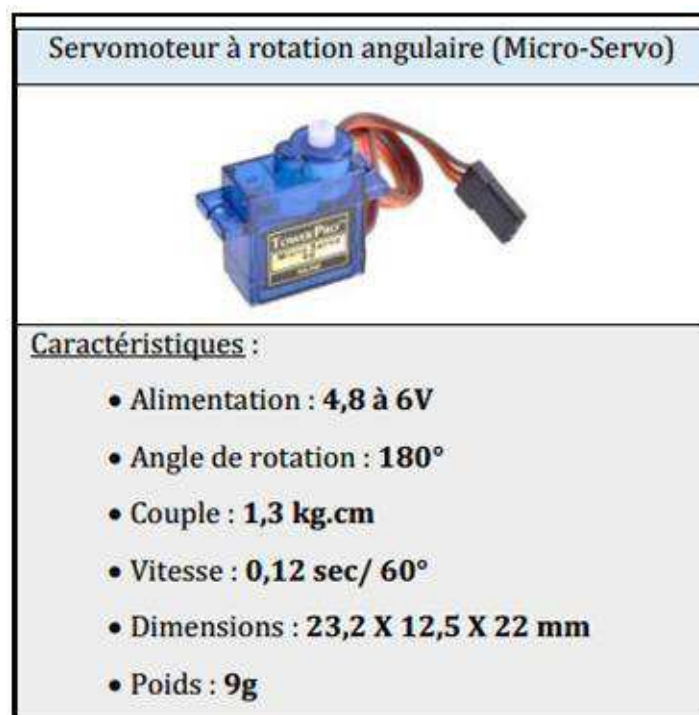


Figure 2.14 : Caractéristiques du servomoteur à rotation angulaire (Micro-Servo).

➤ **Connecteur du servomoteur**

Un servomoteur se pilote par l'intermédiaire d'un câble à (03) trois fils. Ce câble permet à la fois de l'alimenter et de lui transmettre des consignes de position par le fil de signal :

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

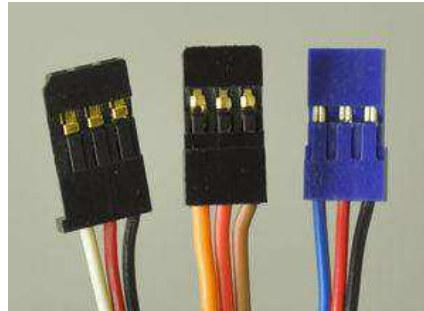


Figure 2.15 : Câble de commande pour servomoteur

➤ Montage du servomoteur avec Arduino

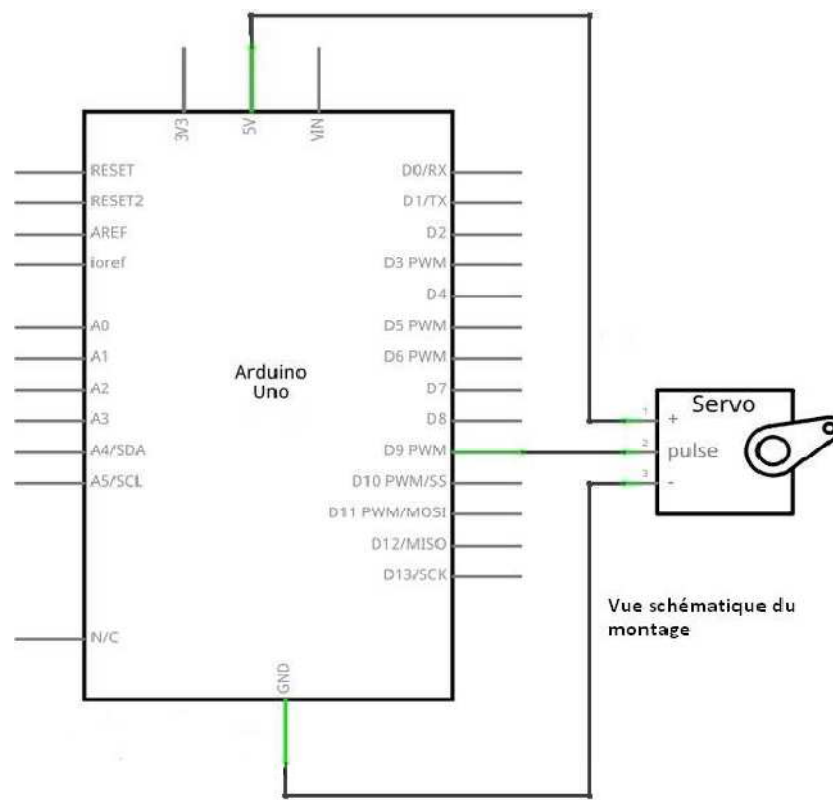


Figure 2.16 : montage servomoteur avec Arduino.

2.3. Logiciels de programmation

2.3.1. Le logiciel de programmation du module Arduino

Le logiciel de programmation du module Arduino est une application Java, libre et multiplateformes, utilisant le langage C++ et lié à la bibliothèque de développement Arduino qui s'appelle « Arduino c ». La mise en place de ce langage standard rend aisé le développement de programmes sur les plates-formes Arduino, à toute personne maîtrisant le C++.

Un programme Arduino doit impérativement être divisé en 2 parties : une fonction « **setup** » et une fonction « **loop** ». Ces deux fonctions sont de type **void**, c'est à dire qu'elles ne peuvent pas prendre de valeurs. La fonction setup est la fonction qui se lance au début du programme. Elle permet d'initialiser les variables et de définir les broches de la carte Arduino qui seront utilisées. La fonction loop se lance après la fonction setup et, comme son nom l'indique, fait une boucle jusqu'à ce que la carte soit débranchée.

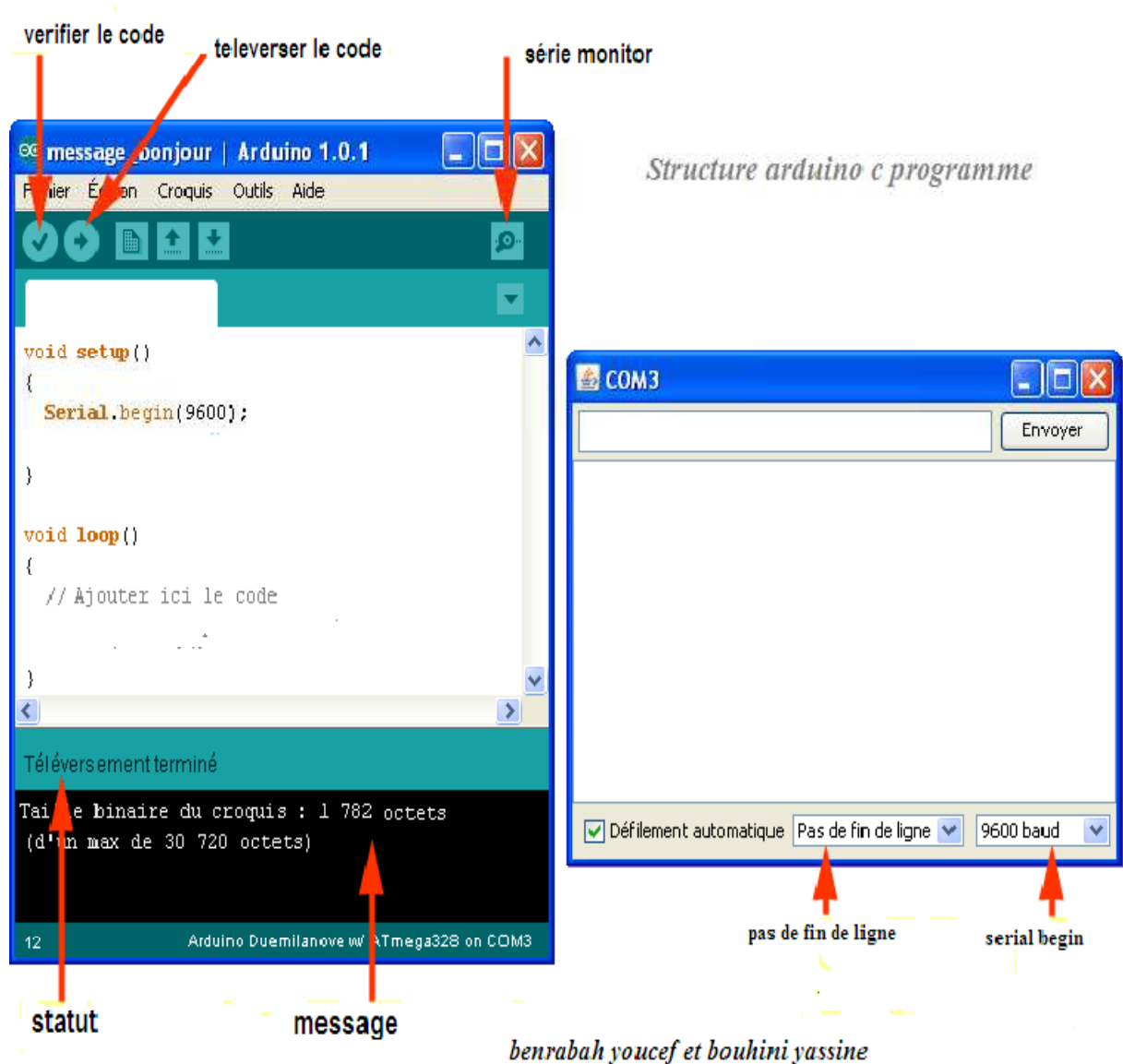


Figure 2.17 : Plateforme Arduino.

➤ **Structure générale du programme (IDE Arduino)**

Un programme utilisateur Arduino est une suite d'instructions élémentaires sous forme textuelle, ligne par ligne. La carte lit puis effectue les instructions les unes après les autres, dans l'ordre défini par les lignes de code. La structure d'écriture d'un programme sous Arduino est de la forme suivante :

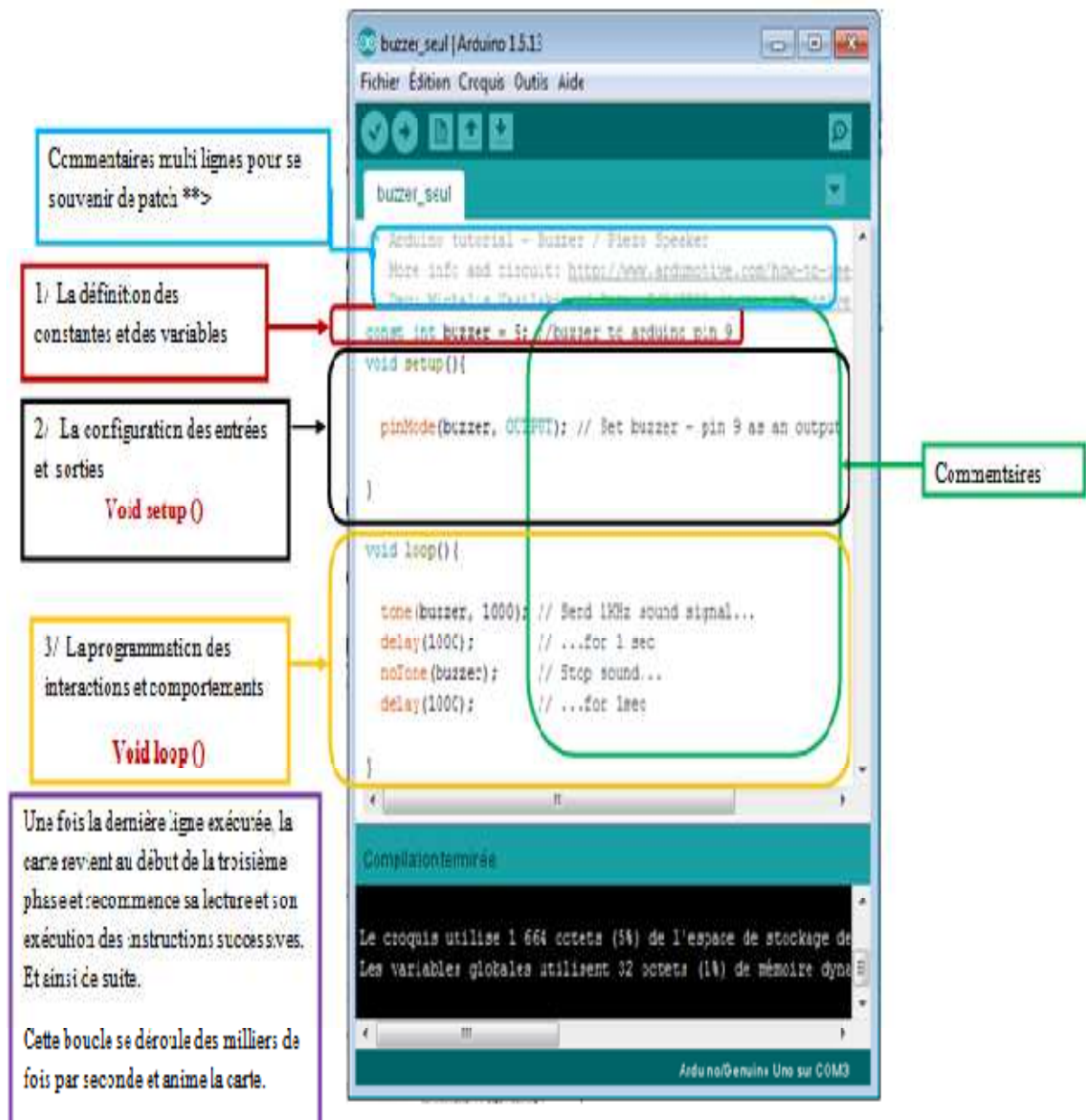


Figure 2.19 : Structure générale du programme (IDE Arduino).

2.3.2. Logiciel IDE de PROCESSIN

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 [13].

2.3.3. Logiciel PROTEUS ISIS

Le logiciel PROTEUS ISIS à la capacité de simuler des montages à base de microcontrôleur, il permet davantage de donner une idée sur la réalisation matérielle et la conception des circuits imprimés. Grâce à des modules additionnels, ISIS est également capable de simuler le comportement d'une carte Arduino et son interaction avec les composants qui l'entourent (capteur ultrason, écran LCD....ex).

2.3.3. Communication entre la carte Arduino, logiciel Proteus et logiciel Processing

L'IDE de Processing est similaire à Arduino en termes de structure. 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 résultants de traitement seront transférées vers le logiciel Processing sous la même liaison.

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.

2.4. Fonctionnement du radar de détection

L'Arduino va démarrer le **temps1** et envoie une impulsion **HIGH** de 10µs sur la broche **tri** du capteur, ce dernier reçoit un ordre pour l'envoi d'une onde ultrasonore de 8 impulsions à 40KHz dans l'air et met la sortie écho à l'état **haut**.

L'Arduino détecte l'état **haut** de sa broche numérique connectée à l'écho et enregistre la valeur du **temps1**, une fois l'onde réfléchit par la cible elle revienne au récepteur ultrasons du capteur. Ce dernier va changer l'état de l'écho vers l'état **bas**. L'Arduino va détecter le changement de l'état sur sa broche numérique et arrête l'incréméntation du **timer1** et enregistre sa valeur dans une variable **temps2** puis il calcule la différence entre le **temps 2** et le **temps 1** et il calcule la distance et l'envoie par la liaison parallèle à l'ordinateur. Le servomoteur commandé par la carte ARDUINO, effectue un balayage de gauche à droite couvrant le champ exploré de 180 degré, puis le balayage inverse, et ainsi de suite.

➤ L'opération qui permet de calculer la distance [14] :

$$d = (v \cdot t) / 2 \quad (2.1)$$

Avec :

d : la distance en cm

v : la vitesse du son dans l'air (v=340 m/s).

t : temps écoulé entre l'émission de l'onde et la réception de l'écho.

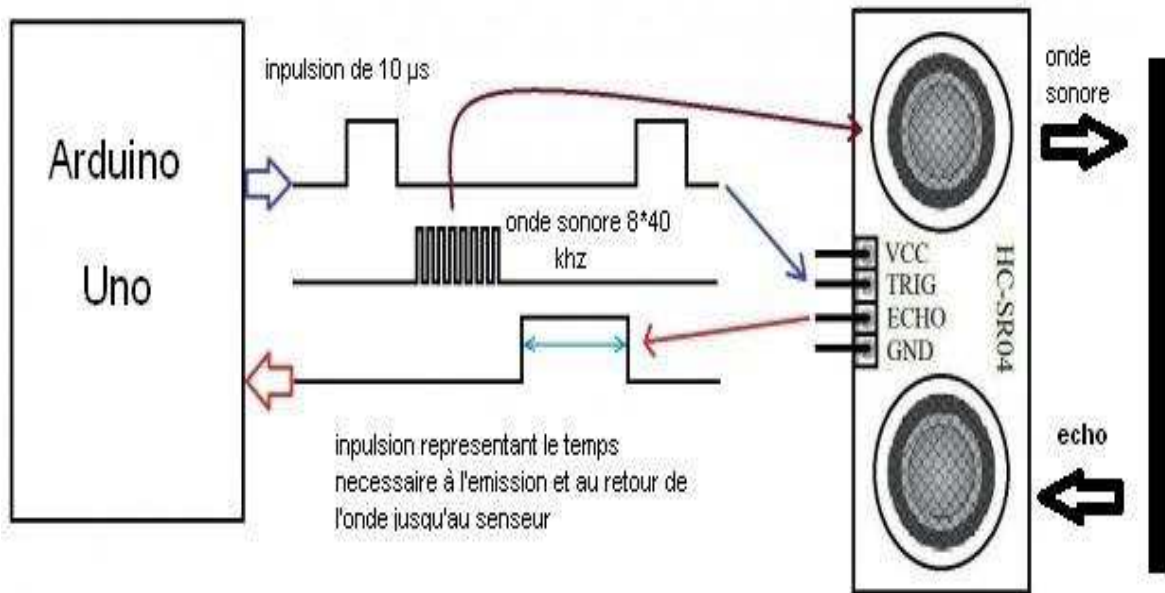


Figure 2.20. : Diagramme temporel de HC-SR04.

2.5. L'organigramme du projet

Le logiciel Arduino effectue la lecture du port et enregistre la valeur dans une variable puis lit l'angle du servomoteur et calcule la coordonnée cartésienne de la cible pour afficher le résultat sur l'écran radar.

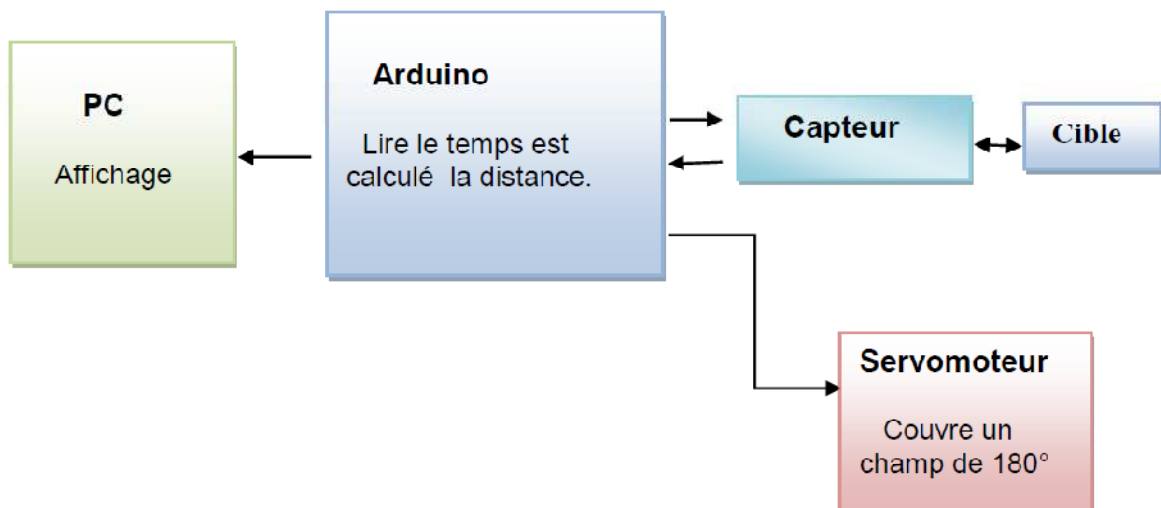


Figure 2.21. Communication entre les composants du projet

2.6. Conclusion

Dans ce chapitre, nous avons fait une description des différents éléments constituant le Radar de détection. Une description Matériel tels que la carte Arduino Uno et de ces composantes, le capteur Ultrasons HC-SR04, l'afficheur LCD ainsi que le servomoteur. Une description des logiciels utilisés tels que le logiciel de programmation du module Arduino, logiciel IDE de Processing et logiciel Proteus ISIS.

Nous avons expliqué le fonctionnement du radar et nous avons donné son organigramme. Le chapitre suivant sera consacré à la réalisation d'un dispositif Radar de détection à capteur ultrason.

3.1. Introduction

Dans ce chapitre, nous présenterons le dispositif expérimental « La réalisation d'un Radar de détection ». Après avoir donné dans le chapitre précédent une description théorique sur la carte Arduino, le servomoteur, le capteur à ultrason, l'afficheur LCD. Nous allons procéder à l'application expérimentale qui se développera en deux étapes, la première étape est la simulation et la deuxième étape est la réalisation.

3.2. Les différentes étapes du projet

La réalisation de notre Radar a été faite en deux parties:

- 1) La première partie est la conception assistée par ordinateur CAO et la simulation avec Proteus.
- 2) La deuxième partie est la réalisation, le montage pratique.

3.2.1. Partie simulation du Radar de détection

Les composants utilisés dans cette simulation sont :

- Un capteur Ultrason HC-SR04
- Arduino UNO R-3
- Une Résistance variable.
- Un servomoteur
- Un écran LCD 16*2.
- Un buzzer.

Avant de simuler le logiciel Proteus nous avons ajouté la bibliothèque du capteur ultrasonique (la bibliothèque ultrasonique contient une librairie de la carte arduino) nommé Arduino.IDX,Arduino.LIB. Malgré l'ajout de cette bibliothèque dans Proteus cela ne fonctionnera pas car nous n'avons pas encore ajouté le fichier HEX.

Pour ajouter ce fichier, nous faisons un double-clic sur le capteur à ultrasons (il devient rouge comme il est représenté sur la figure 3.1). Ce qui nous permet d'avoir ces propriétés et de sélectionner la section fichier du programme et nous choisissons le fichier UltrasonicTEP.HEX comme indiqué sur la figure 3.2.

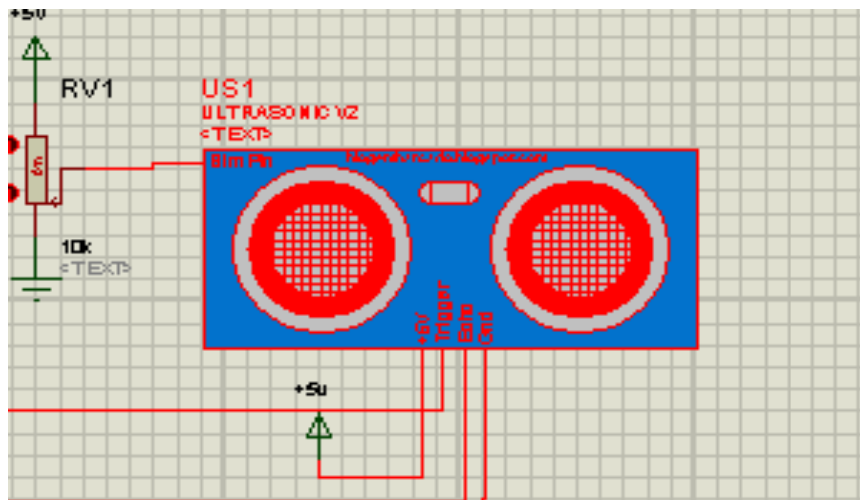


Figure 3.1 : Capteur HC-sr04 sous Proteus.

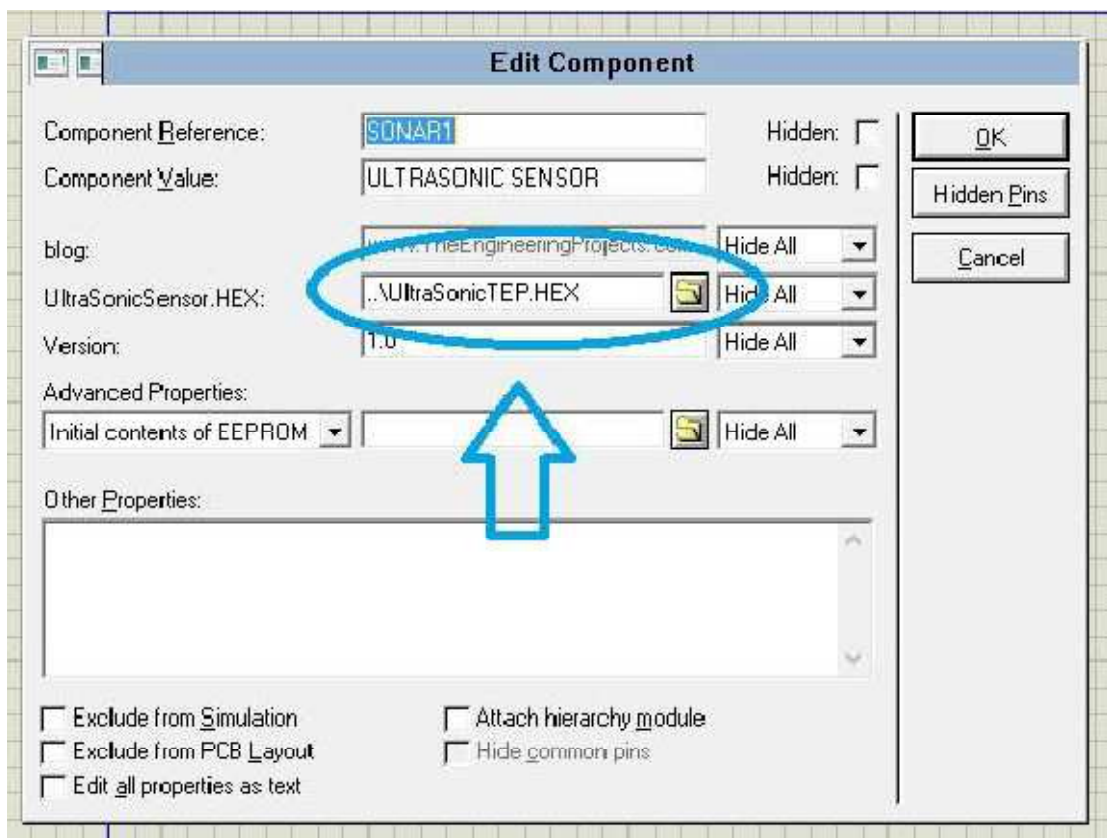


Figure 3.2 : Rajout du fichier UltrasonicTEP.HEX sous Proteus.

L'étape suivante c'est l'ajout des composants ci-dessous dans l'espace de travail :

- Capteur Ultrason HC-SR04
- Carte Arduino UNO R-3
- Résistance variable.
- Un écran LCD 16*2
- Un buzzer

3.2.1.1. Circuit global de la simulation

La figure suivante présente le circuit global de notre simulation sous Proteus.

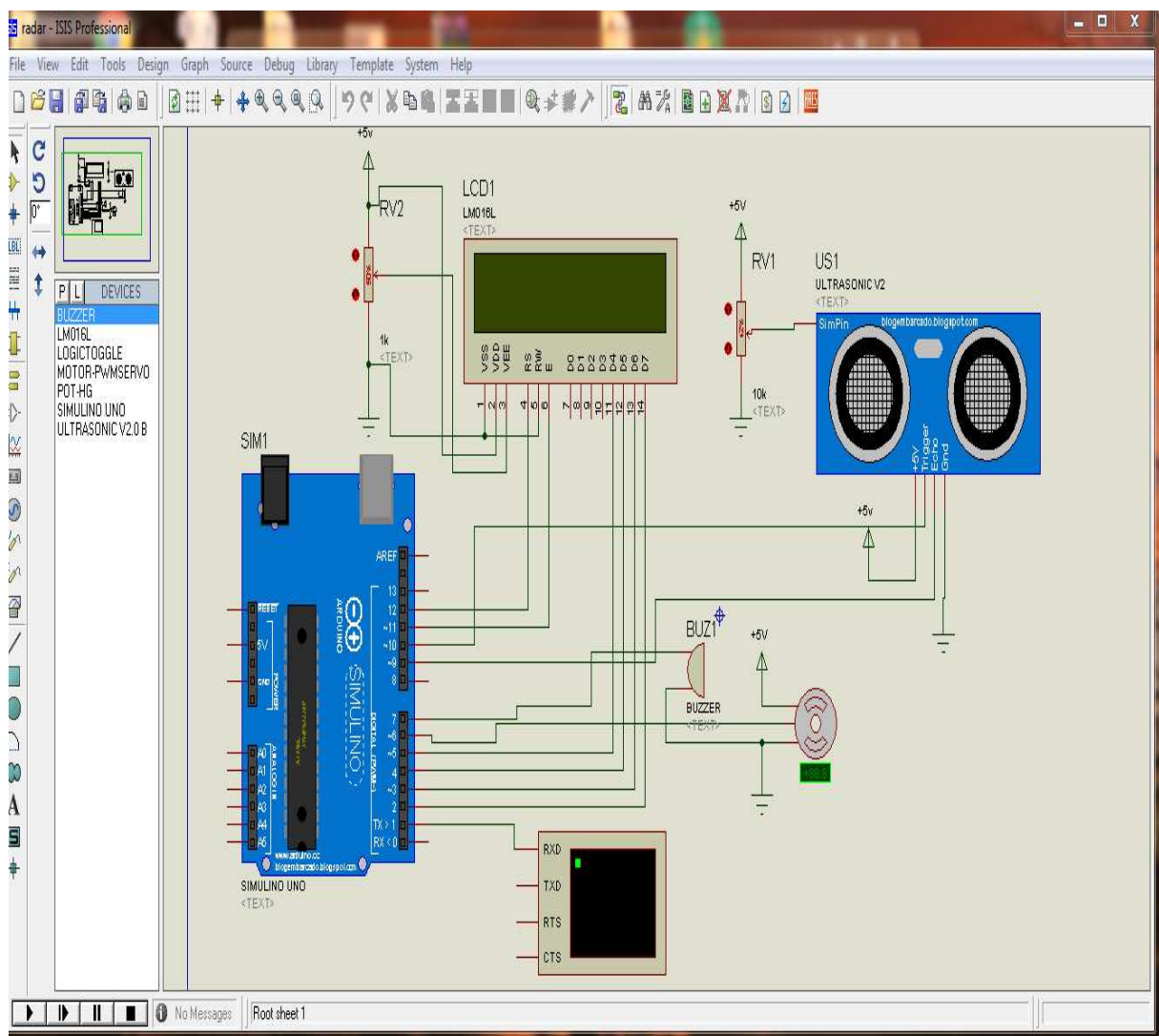


Figure 3.3 : Le circuit global de notre simulation

3.2.1.2. Injection du fichier .HEX sous proteus

Après avoir copier le fichier hexadécimal sur le logiciel ARDUINO, nous allons faire un double cliquer sur la carte Arduino, ce qui nous permet d'ouvrir ses propriétés et de sélectionner le fichier du programme afin de parcourir le fichier .HEX et de le télécharger comme indiqué ci-dessous (figure 3.4).

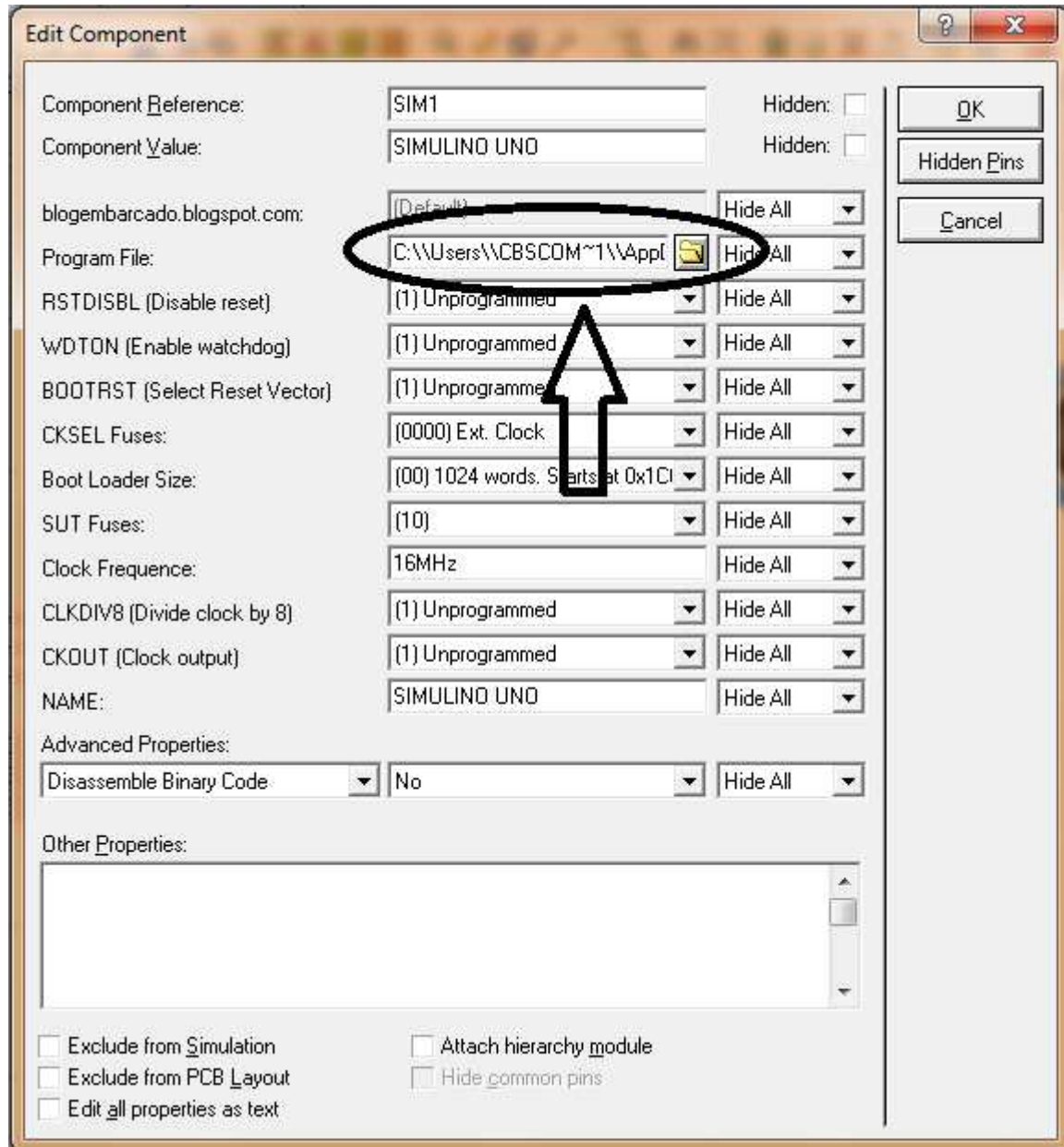


Figure 3.4 : Injection du fichier .HEX sous proteus.

Pour avoir les résultats de simulation nous appuyons sur le bouton Démarrer et les résultats seront affichés comme sons représentés sur la figure ci-dessous.

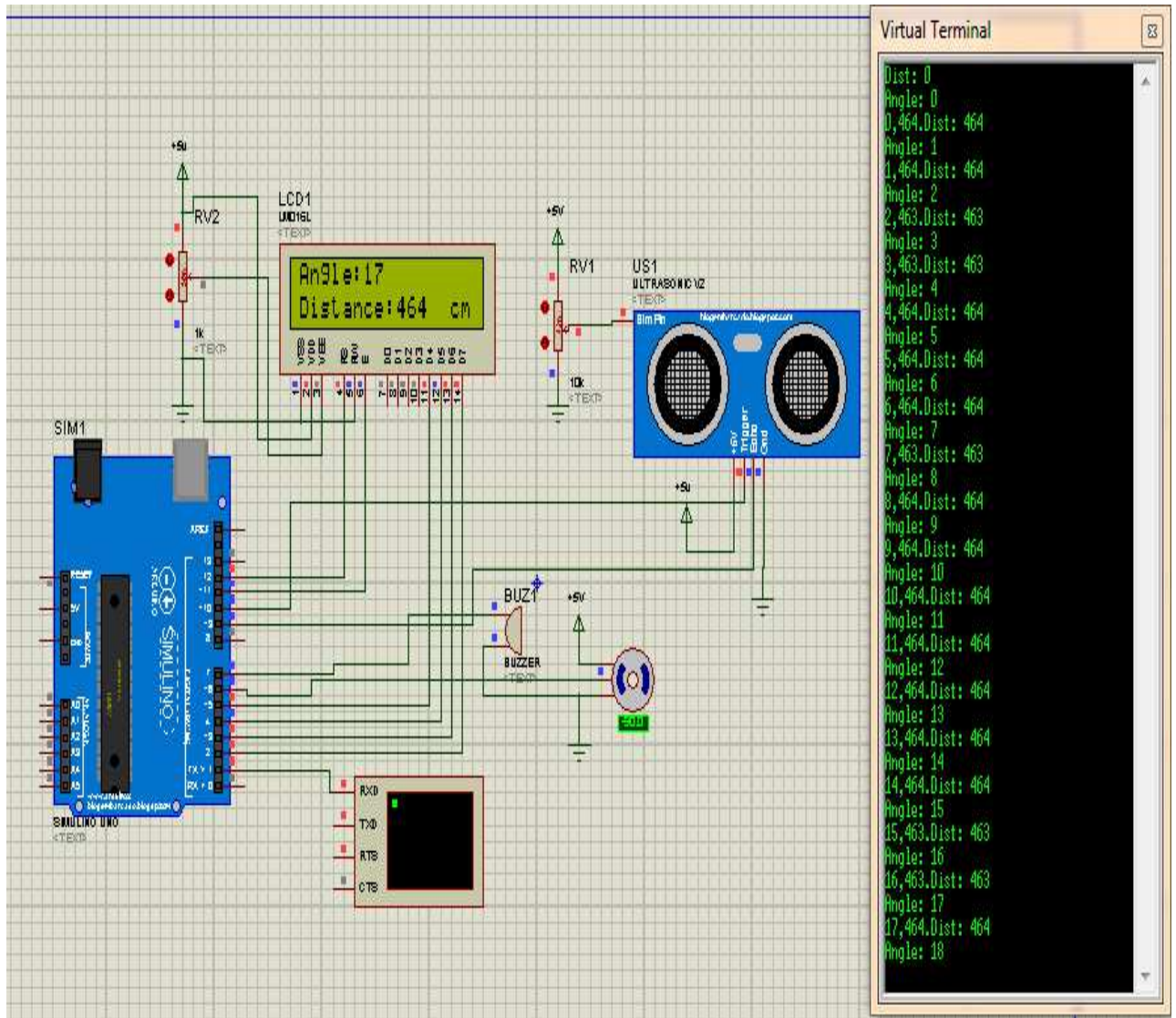


Figure.3.5 : Résultats de simulation sur le terminal virtuel.

Le terminal virtuel affiche la valeur de distance et la valeur de l'angle, cette valeur dépend de la résistance variable associée au capteur à ultrasons, lorsque nous augmentons ou nous réduisons la tension de la résistance variable la valeur de la distance et de l'angle change aussi.

3.2.1.3. Schéma synoptique de notre Radar

La figure ci-dessous illustre le schéma synoptique de notre radar de détection :

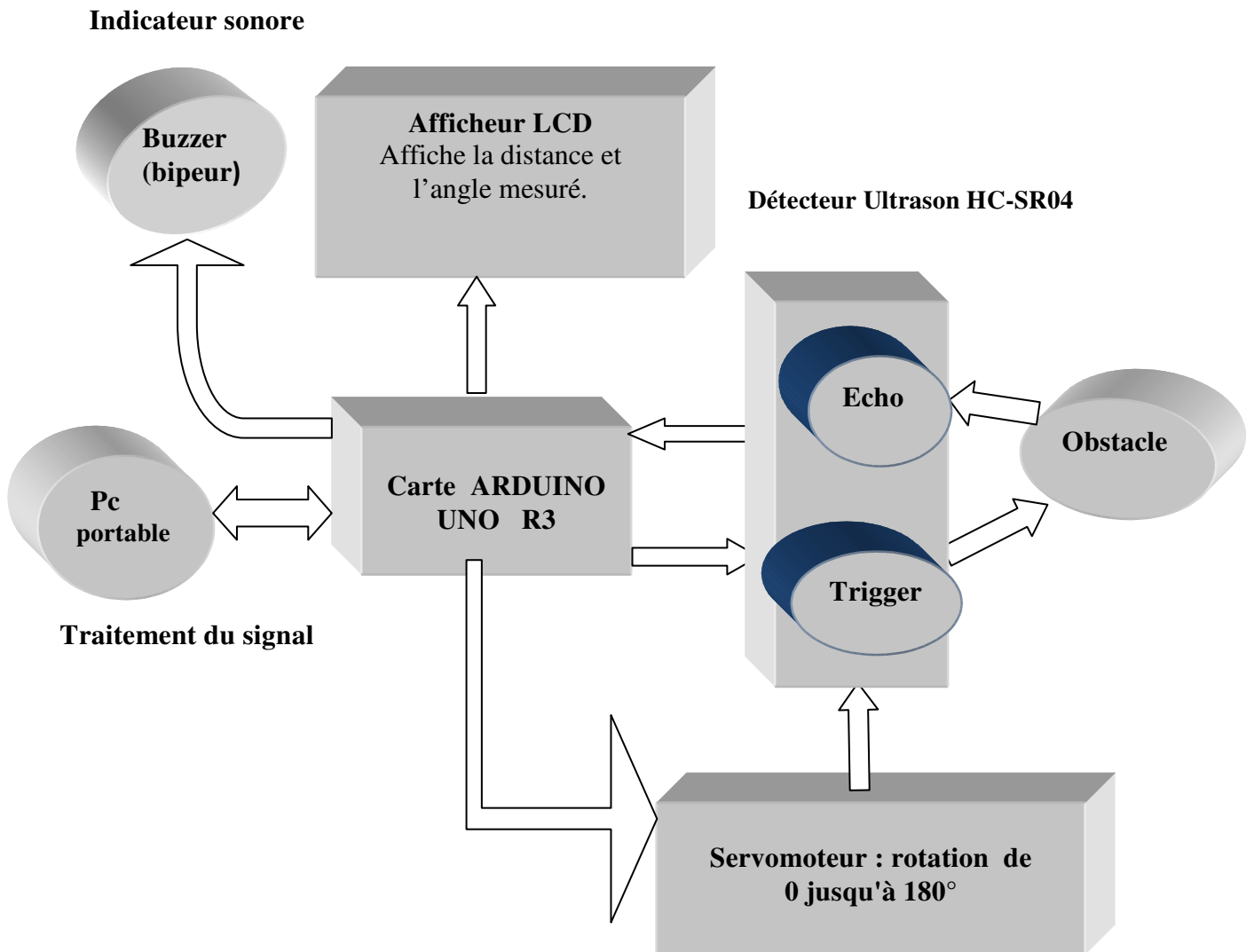


Figure 3.6 : Schéma synoptique de notre Radar.

3.2.1.4. Algorithme principale de détection d'un objet

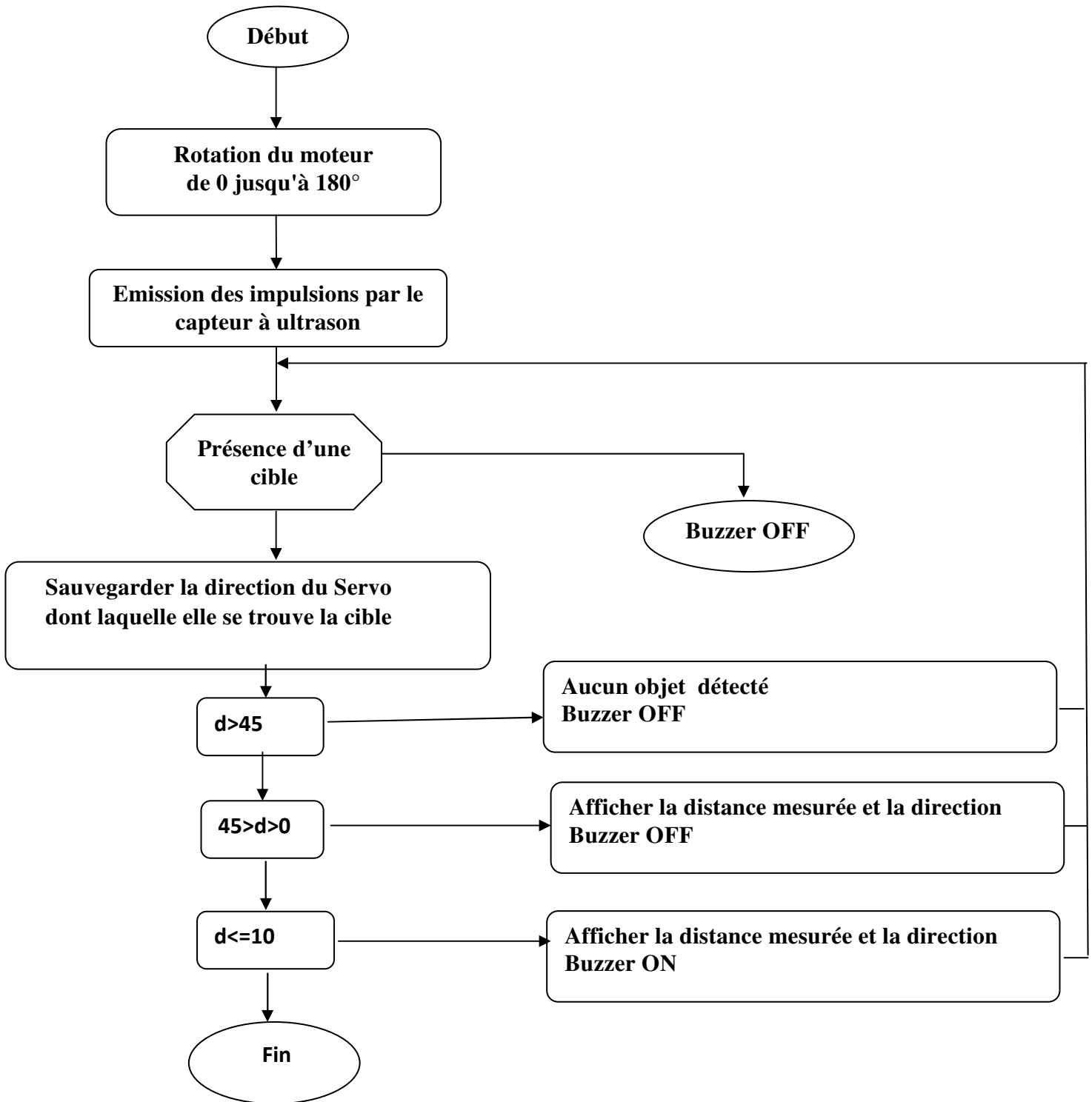


Figure 3.7 : Algorithme principale de détection d'un objet

3.2.1.5. Dessiner un radar sur processing

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

1) Dessiner l'arc et les lignes du Radar

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

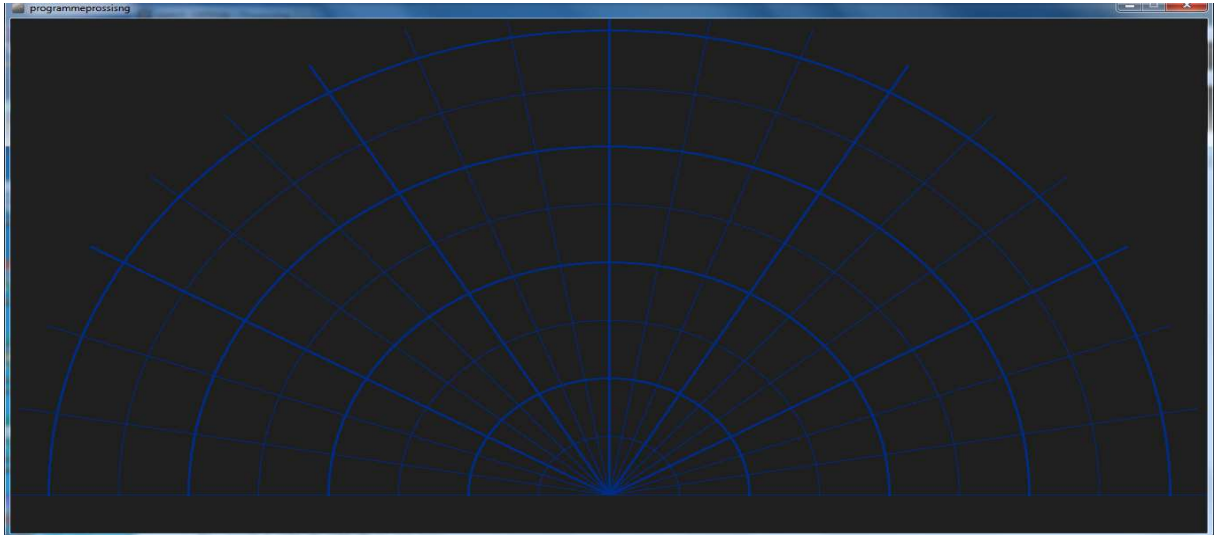


Figure 3.8 : l'arc et les lignes du Radar.

2) Dessiner l'objet

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

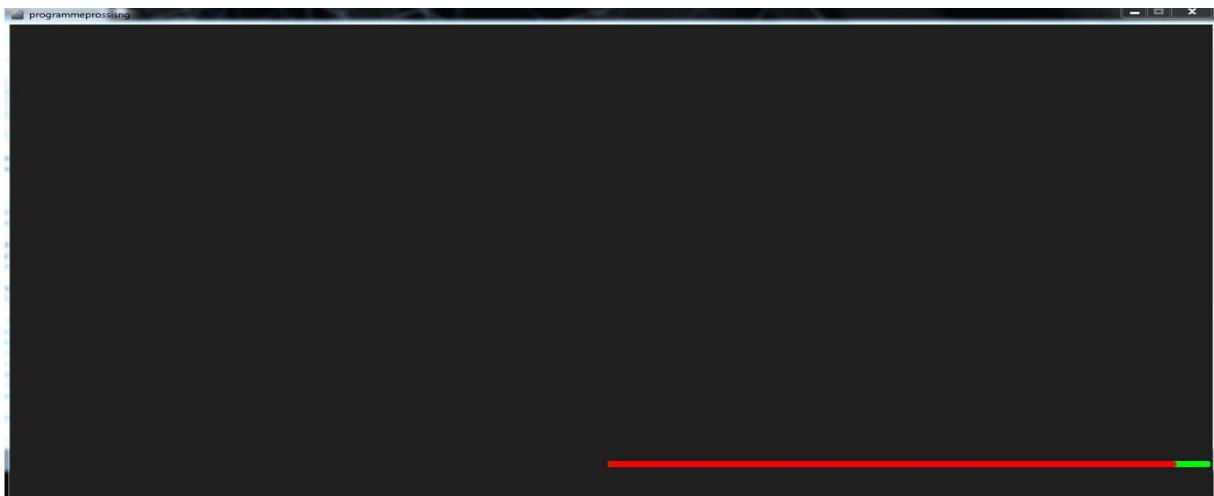


Figure 3.9 : l'objet détecté par la fonction **drawObject()**.

3) Ecriture du texte sur processing

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



Figure 3.10 : écriture d'un texte à l'aide de la fonction `drawText()`

4) Simulation complète sur Processing

Sur Processing la détection d'objet est affichée en rouge, la distance mesurée ne dépasse pas 45cm.

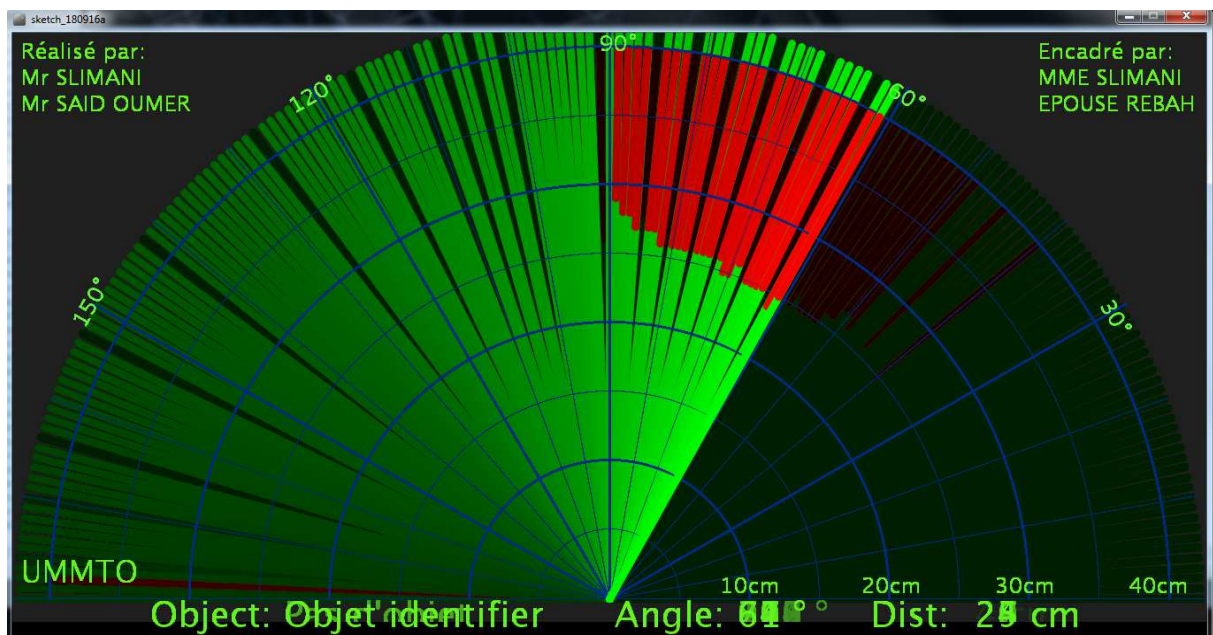


Figure 3.11 : Simulation complète sur Processing.

3.2.2. Partie réalisation de notre Radar détection

3.2.2.1. Les différentes connexions et numéro des pins du circuit

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

Composant	Pin composant	Pin Arduino Uno
Capteur A Ultrason	VCC	+5V
	Trigger	10
	Echo	9
	GND	GND
Afficheur LCD	RS pin	12
	Enable pin	11
	D4	5
	D5	4
	D6	3
	D7	2
	R/W	GND
	VSS pin	GND
Servomoteur	VCC	+5V
	Marron	GND
	Rouge	+ 5v
Buzzer	Orange	6
	Pin+	07
Port Com liaison Virtuelle	Pin-	GND
	3	TX
	2	RX

Tableau 3.1 : Broche et connexion de notre circuit électronique

3.2.2.2. Principe de fonctionnement du circuit

La carte Arduino brancher sur l'USB du pc envoie un signal de (+5V) vers tout ces pins connectés d'où le déclenchement du servomoteur qui est très important, il offre l'action rotationnelle au capteur pour qu'il puisse détecter les objets mobiles et situant dans les 180 degrés.

La carte envoie une impulsion HIGH de largeur (10 μ s) sur la broche TRIGGER du capteur pour régénérer une séries de (08) ondes ultrasonores de fréquence de 40 KHz dans l'air provenant de l'émetteur (Inaudible pour l'être humain). Les ultrasons se propagent dans l'air jusqu'à toucher un obstacle et retourne dans l'autre sens vers le capteur par la broche ECHO, Le capteur détecte la largeur de l'impulsion pour calculer la distance.

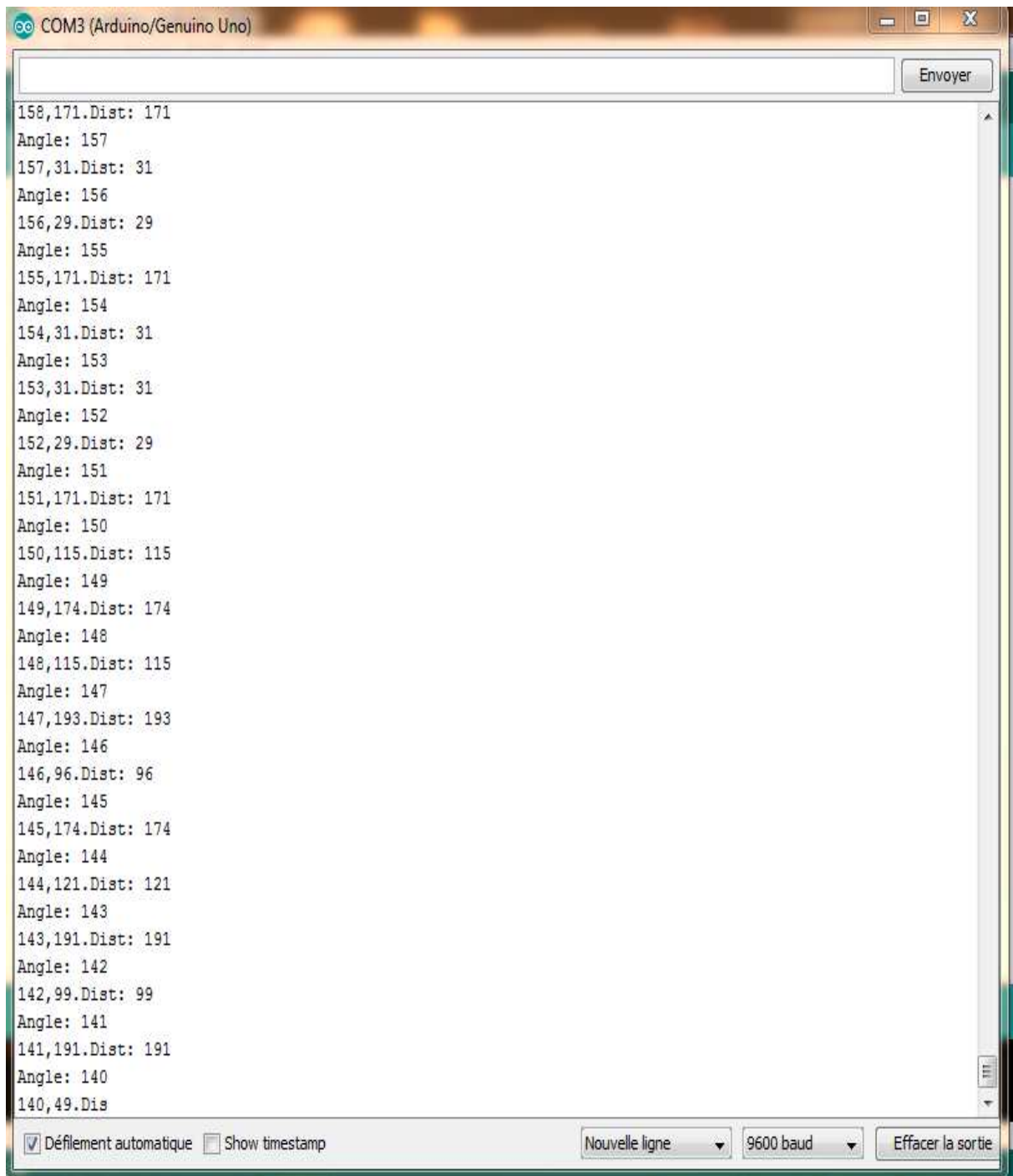
Le signal sur la broche ECHO du capteur reste à HIGH pendant l'envoi et la réception, ce qui permet de mesurer la durée de l'aller-retour des ultrasons et donc de déterminer la distance. La représentation Circuit final de notre projet de réalisation du Radar de détection est faite sur la figure 3.12.



Figure 3.12. Circuit final de notre projet de réalisation du Radar de détection.

3.2.2.3. Visualisation de la distance et de l'angle sur le moniteur série

La carte Arduino envoie les résultats de calcul du capteur ultrasonique au pc et on cliquant sur le moniteur série du logiciel Arduino, il affiche les résultats de détection (figure 3.13).



The screenshot shows the Arduino Serial Monitor window for COM3 (Arduino/Genuino Uno). The window displays a list of data points received from the Arduino, each consisting of a distance value and an angle value. The data points are as follows:

Angle	Distance
158	171
157	31
156	29
155	171
154	31
153	31
152	29
151	171
150	115
149	174
148	115
147	193
146	96
145	174
144	121
143	191
142	99
141	191
140	49

The window also features a 'Envoyer' button at the top right and a control bar at the bottom with options for 'Défilement automatique' (checked), 'Show timestamp' (unchecked), 'Nouvelle ligne' (dropdown), '9600 baud' (dropdown), and 'Effacer la sortie'.

Figure 3.13: Visualiser la distance et l'angle dans moniteur série

3.2.2.4. Résultats des différents objets détectés

Les figures ci-dessous extraites des résultats pratiques

➤ Mise en marche du Radar

La figure suivante (3.14) représente la mise en marche de notre Radar de détection face un objet à repérer.

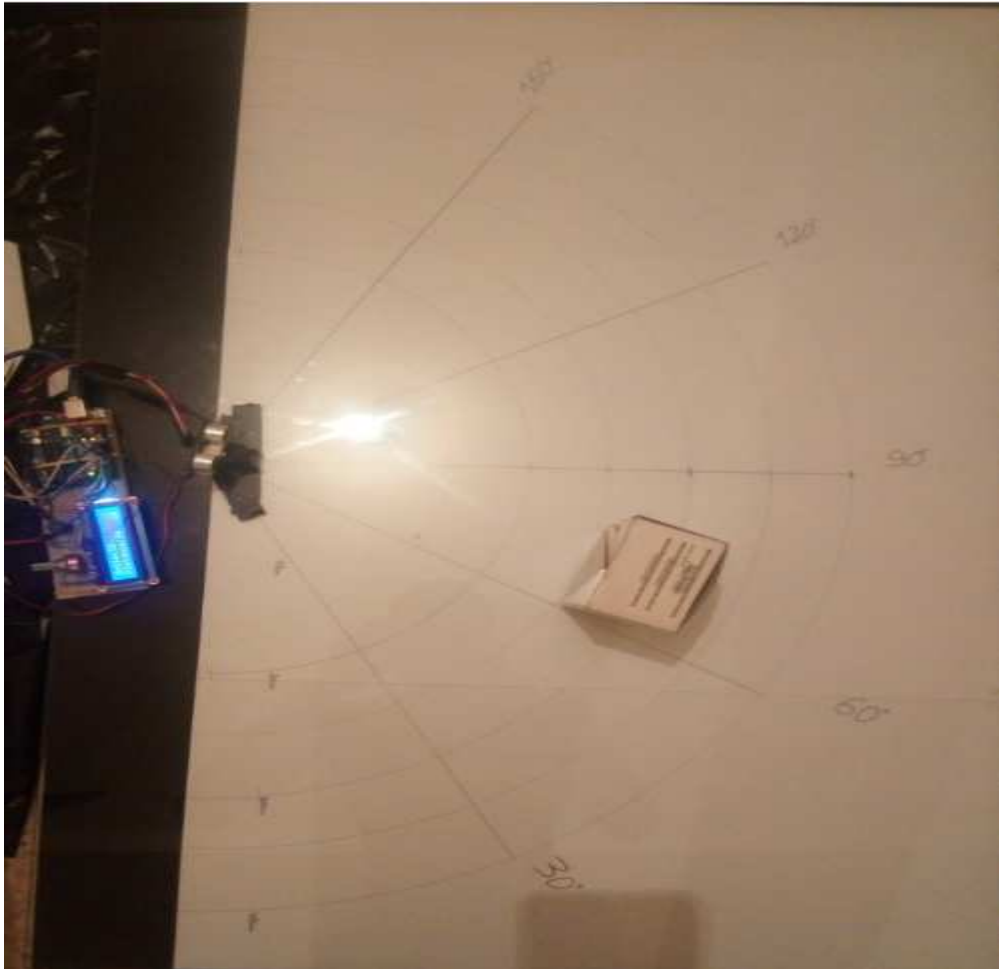


Figure 3.14 : Image global de la mise en marche du Radar.

➤ **Détection d'un objet à une distance 5cm et à un angle de 37°**

Cette figure illustre un exemple d'objet que nous avons détecté à une distance 5cm et à un angle de 37° (objet 09 du tableau 3.2)

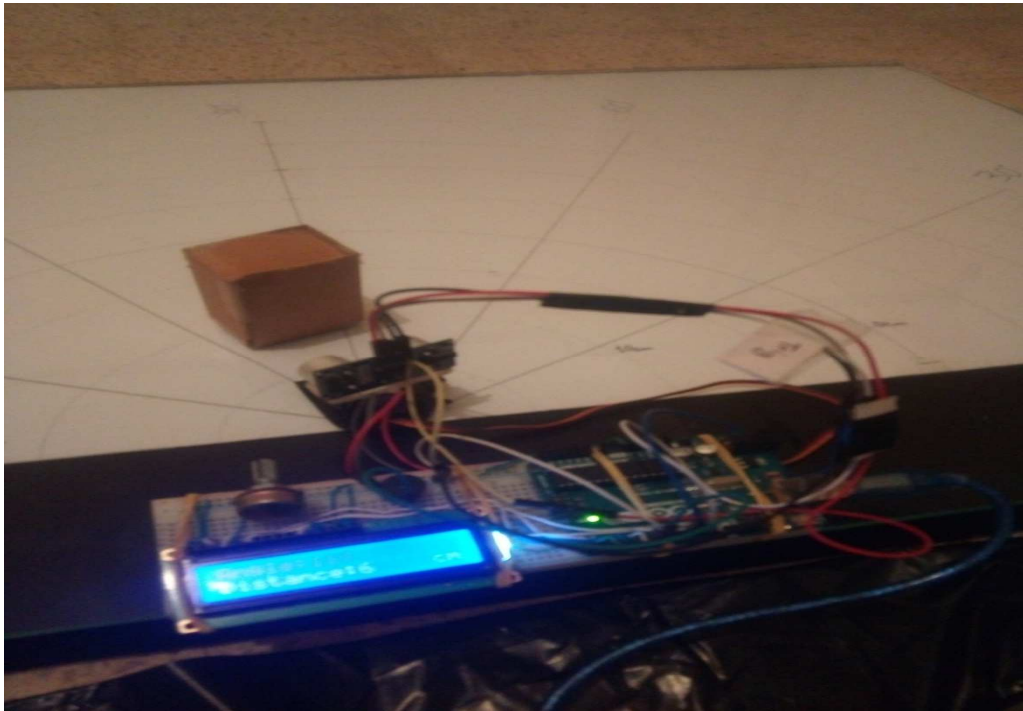


Figure 3.15 : Détection de l'objet 09.

➤ **Interface de Processing qui illustre la détection de l'objet 09**

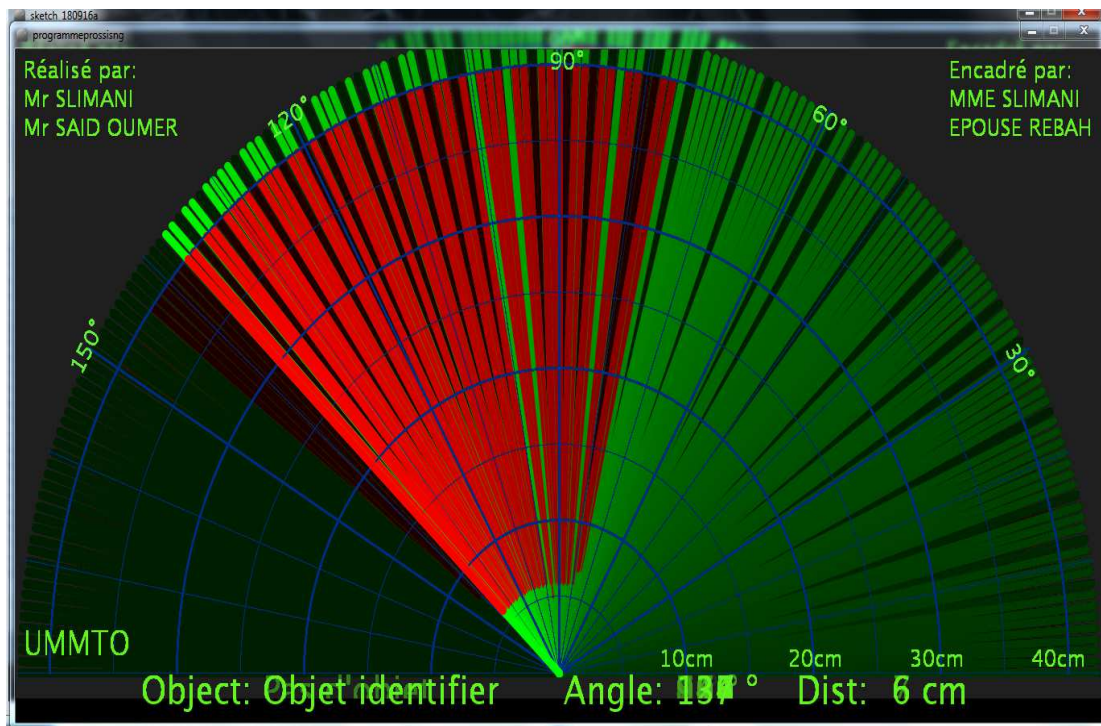


Figure 3.16 : Interface du Processing avec détection (Présence de l'objet 09).

Les deux figures précédentes obtenues dans notre réalisation du projet radar de détections (figure 3.15, figure 3.16) montrent que les résultats obtenus avec la simulation ressemblent aux résultats pratiques.

➤ **Détection de plusieurs objets à des différentes distances et à différents angle**

Le tableau ci-dessous montre les différents objets détectés dans plusieurs places.

N° d'objets détectés	Distance mesurés	L'angle mesuré
Objet 01	34	20
Objet 02	40	29.35
Objet 03	10	89
Objet 04	9	98
Objet 05	8	136
Objet 06	38	180
Objet 07	19	97
Objet 08	20	73
Objet 09	5	37
Objet 10	33	41
Objet 11	23	73
Objet 12	37	165

Tableau 3.2 : Les résultats des différents objets détectés par notre radar

3.2.2.5. Interprétations des résultats trouvés

Le tableau au dessus est un extrait des résultats pratiques que nous avons trouvé, il contient les informations (Distance et direction) de plusieurs objets fixes détectés.

Le déclenchement de bipeur indique que l'objet est détecté à d'une distance inférieur ou égale à 10 cm. Lorsqu'il y a une présence d'une cible fixe, il y'a une puissance importante réfléchié vers le récepteur et une variation de fréquence.

Le signal écho, présente un signal de faible fréquence, lorsque nous mettons une cible devant le capteur, nous remarquons le changement de fréquences par la suite, d'où la détection de l'objet fixe.

3.3. Conclusion

Dans ce chapitre nous avons fait une simulation et une réalisation d'un Radar de détection à l'aide de la carte Arduino Uno R3, afin de mesurer la distance et la direction des objets à travers un capteur à ultrason et dans les directions de 0 à 180° à l'aide du servomoteur.

Nous avons constaté lors de la simulation sur ISIS Proteus que la valeur de la distance dépend de la résistance variable associée au capteur à ultrasons, lorsque nous augmentons ou diminuons la tension de la résistance variable la valeur de la distance change (augmente ou diminue).

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

Conclusion générale

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

Notre mémoire avait pour objectif l'étude et la réalisation d'un Radar de détection en utilisant la carte Arduino, afin de le réaliser nous l'avons fait en deux parties.

La première partie est la conception assistée par ordinateur CAO, la simulation consiste en l'ajout d'un Potentiomètre connecté au pin de test du capteur à ultrason et la variation de la valeur du potentiomètre nous donne une variation de puissance. Les résultats obtenus nous indiquent que la valeur de la distance mesurée dépend de la résistance variable associée au capteur à ultrasons.

La deuxième partie est la réalisation du projet du radar de détection, pour l'effectuer nous avons passés par plusieurs étapes : la première consiste à programmer la carte Arduino Uno R3 (programmation des composants : capteur ultrason, buzzer, servomoteur et l'écran LCD) et de leurs affectés l'action à exécuter, puis la programmation du logiciel Processing pour afficher l'interface du radar. Dans la deuxième étape nous avons ajoutés un servomoteur qui peut tourner de 0 à 180° et un capteur ultrason HC-SR04 qui permet la détection des obstacles et le calcul de la distance entre le radar et l'objet. La dernière étape est consacré à l'affichage de la distance et de l'angle mesuré à l'aide d'un afficheur LCD2*16. Le contrôle de la luminosité de l'écran LCD est fait par le Potentiomètre et l'ajout d'un buzzer sonore actif pour signaler qu'un objet est détecté sur une distance égal ou inferieur à 10 cm.

Sur l'ensemble d'objets que nous avons détectés nous avons constaté que la distance dépend de la résistance variable associée au capteur à ultrasons, lorsque nous augmentons ou diminuons la tension de la résistance variable la valeur de la distance change.

Les résultats trouvés sont satisfaisant mais l'utilisation d'un capteur laser à la place du capteur ultrasonique peut améliorer ces résultats.

- [1] **Michel-Henri CARPENTIER**, «RADAR», Encyclopaedia Universalis, URL : <http://www.iniversalis.fr/encyclopedie/radar/>
- [2] **M. Skolnik** , « Introduction to radar systems». Mc-Graw-Hill, New-York, 1981
- [3] **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.
- [4] **ACHACHI ABDELALI**, « Modélisation d'un Radar secondaire pour un trafic aérien » Thèse de Magister en Electronique, Université de Batna, Algérie. 02/03/2010.
- [5] **KRATTOU Mohamed Riad**, « Étude De La Détection Radar Dans Un Milieu Homogène », mémoire de master 2 en télécommunications, l'université de Tlemcen, Algérie. Promotion 2012/2013.
- [6] **DJEDOUI 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] **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.
- [8] **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.
- [9] **B.Cottenceau**, « Carte ARDUINO Uno Microcontrôleur ATmega328 ». Microcontrôleurs EI3 Option AGI.
- [10] **Capteur à Ultrasons HC-SR04** .www.robot-maker.com.
- [11] **LAKRID DJAHID**, « Emulation de Système SCADA pour le Contrôle et Distribution d'eau », mémoire de master 2 en automatique industriel, Université Badji Mokhtar Annaba. Algérie. Promotion 2017.

[12] **KRAMA Abdelbasset, GOUGUI Abdelmoumen.** « Etude et réalisation d'une carte de contrôle par Arduino via le système Androïde ». Mémoire de Master Académique 2 en Electrotechnique Industrielle. Université Kasdi Merbah Ourgla. Algérie. Le 08/06/2015.

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

[14] **hc-sr04** « pj2-hc-sr04-utilisation-avec-picaxe-1343.pdf »

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

[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] **Kadam D.B, Patil YuvrJ B, Chougale Krishnat V, Perdeshi Swagat ,** « Arduino Based Moving Radar System », www.ijisset.org, ISSN 2455-4863 (Online), Volume: 3 Issue: 4 | April 2017.