الجمهورية الجزائرية الديمقراطية الشعبية

**REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE**

وزارة التعليم العالي والبحث العلمي

**Ministère de l'Enseignement Supérieur et de la Recherche Scientifique**

جامعة أبي بكر بلقايد– تلمسان –

Université Aboubakr Belkaïd– Tlemcen –

Faculté de TECHNOLOGIE

# MEMOIRE

Présenté pour l'obtention du **diplôme** de **MASTER**

**En** : Télécommunications

**Spécialité** : Réseaux et Systèmes de Télécommunications

**Par** : **DJEDOUI Nacéra & BOUKERN Dounya**

## Sujet

# Étude et réalisation d'un RADAR de détection

Soutenu publiquement, le **19/06/ 2017**, devant le jury composé de :

| | | |
|---|---|---|
| Mr S.M MERIAH | Professeur à l'université de Tlemcen | Président |
| Mr M. BOUSAHLA | MCB à l'université de Tlemcen | Examinateur |
| Mr F.T BENDIMERAD | Professeur à l'université de Tlemcen | Encadreur |
| Mr M.Y BENDIMERAD | MCB à l'université de Bechar | Co-encadreur |

# *Remerciement*

*Avant tout nous tenons tout d'abord à remercier notre dieu tout puissant de nous avoir donné, la force et le courage, la santé, les moyens afin de pouvoir accomplir ce modeste travail.*

*Les premières personnes que nous tenons à remercier particulièrement sont nôtres encadreurs Mr F.T BENDIMERAD et Mr M.Y BENDIMERAD pour l'orientation, la confiance, la patience, qui ont constitué un apport considérable sans lesquels ce travail n'aurait pas pu être mené au bon port.*

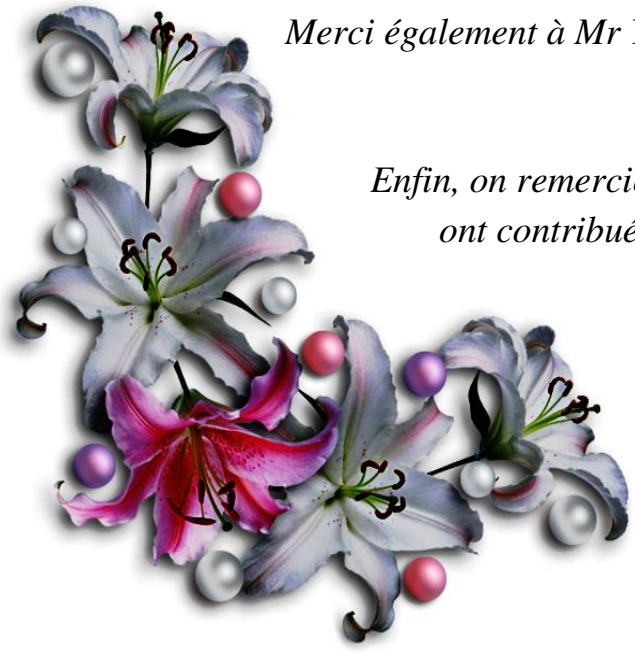
*Nos vifs remerciements vont également aux membres du jury Mr S.M MERIAH et Mr M. BOUSAHLA pour l'intérêt qu'ils ont porté à notre recherche en acceptant d'examiner notre travail et de l'enrichir par leurs propositions.*

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

*Nous souhaitons également exprimer notre gratitude au responsable du laboratoire électronique pour Toutes tentatives de fournir des composants nécessaires de projet*

*Merci également à Mr Y. BOULOUIZ qui nous a aidé et nous a encouragé.*

*Enfin, on remercie tous ceux qui, de près ou de loin, ont contribué à la réalisation de ce travail.*



# *Dédicaces*

*C'est avec profonde gratitude et sincères mots,*
*que je dédie ce modeste travail de fin d'étude à*
*mes chers parents; qui ont sacrifié leur vie pour*
*ma réussite et m'ont éclairé le chemin par*
*leurs conseils judic*
*ieux.*
*j'espère qu'un jour,*
*Je pourrai leurs rendre un peu de ce qu'ils*
*ont fait pour moi, que dieu leur prête bonheur et longue vie.*
*Je dédie aussi ce travail à mes frères Ahmed, Amr et ma sœur*
*Bouchra, ma joie ne sera pas complète que lorsque*
*elle obtiendra son baccalauréat Inchallah.*
*J'espère que ce travail soit un exemple pour*
*vous de persévérance, de courage et de générosité.*
*Je tiens de dédier ce travail à toute ma famille ainsi*
*mes ami(e)s, Youssef, Bouchra et Emen*
*et chacun(e) par son nom.*
*Je dédie ce travail a ma chère binôme Mlle N.DJEDOUI.*

*Dounya*

***« Le succès n'est pas la clé du bonheur. Le bonheur est la clé de succès. Si vous aimez ce que vous faites vous réussirez. »***

***Albert Schweitzer***

# *Dédicaces*

*Du plus profond de mon cœur, je dédie ce travail*

*à tous ceux qui me sont chers,*

***A MES CHERS PARENTS***

*Aucune dédicace ne saurait exprimer mon respect, mon amour eternel et ma considération pour les sacrifices que vous avez consenti pour mon instruction et mon bien être.*

*Je vous remercie pour tous les soutiens et l'amour que vous me portez et j'espère que votre bénédiction m'accompagne toujours.*

*Que ce modeste travail soit l'exaucement de vos vœux tant formolés, le fruit de vos innombrables sacrifices. Puisse Dieu, le très haut, vous accorder santé, bonheur et longue vie.*

***A MES CHERS FRERES***

*Hadj Mohamed, Mustapha, Hadj Redouane, Abdelghani, Ahmed, Youcef.*

*pour les encouragements, et leur soutien.*

***A MES CHERES SŒURS,***

*Malika, Zobida, Khatima.*

*Je tiens de dédier ce travail à tous les membres de ma famille ainsi mes amies, Naima, Nabila.*

*Je dédie ce travail à ma chère binôme*

*Mlle N.BOUKERN qui a trouvé la patience de me supporter durant ce mémoire, et qui m'a soutenu et m'encouragé*

*Nacéra*

# Résumé

**Résumé**

Ce projet a pour but de réaliser un prototype de système RADAR à base de carte Arduino, capable de détecter des objets fixes et mobiles, le travail consiste à faire 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 matériel C et d'affichage Processing, la fin de travail un dispositif RADAR a été conçut, et testé sur des cibles réelle.

**Mots clés** : Prototypage, Radar, Arduino, détecteur, ultrason, servomoteurs, LCD,

**Abstract**

The aim of this project is to produce a prototype of an Arduino-based RADAR system capable of detecting fixed and mobile targets. The aim is to make On the one hand a general study of the Radar theory, the electronic card (Arduino ) and on the other hand the hardware programming language C , and the display processing language, at the end, a RADAR device is designed, conserved and tested on real targets.

**Keywords:** Prototyping, Radar, Arduino, detector, ultrasonic, servo motors, LCD,

**ملخـص**

يهدف هذا المشروع إلى تحقيق نموذج رادار يمكنه الكشف عن الأجسام الثابتة والمتحركة، باستعمال بطاقة أردوينو الذي يتجسد هذا العمل في إجراء دراسة نظرية عامة عن الرادارات و الأردوينو من جهة ومن جهة أخرى حول لغة البرمجة 'س' و لغة الكشف بروسيسنغ وفي الأخير تم تصميم جهاز رادار واختباره حقيقة في وجود الأجسام.

**كلمات مفتاحيه:** نماذج, رادار, أردوينو, كاشف, الموجات فوق الصوتية, مضاعفات الحركة

# Liste des abréviations

| | |
|---|---|
| **Radar** | Radio Detection And Ranging. |
| **IEEE** | Institute of Electrical and Electronics Engineers. |
| **MTI** | Moving Target Indicator. |
| **CW** | Continuous Wave. |
| **FMCW** | Frequency-Modulated Continuous-Wave. |
| **PRF** | Pulse Repetition Frequency. |
| **LO** | Local Oscillator. |
| **RF** | Radio Frequency. |
| **IF** | Intermediate Frequency. |
| **CRT** | Cathode-ray Tube. |
| **DPA** | Détection et Poursuite Automatique. |
| **FFT** | Fast Fourrier Transform. |
| **SER** | Surface Equivalente Radar. |
| **SNR** | Signal Noise Ratio. |
| **Pdf** | Probabilité de détection de fausse alarme. |
| **DSP** | Densité spectrale de puissance. |
| **LED** | Light-Emitting Diode |
| **IDE** | Integrated development environment |
| **AVR** | Automatic Voltage Regulator |
| **USB** | Universal Asynchronous Receiver Transmitter |
| **RS232** | Recommended Standard 232 |
| **PWM** | Pulse-width modulation |
| **RAM** | Random-access memory |
| **EEPROM** | Electrically-erasable programmable read-only memory |
| **SRAM** | Static random-access memory |
| **Gnd** | Ground |
| **RX** | Receive/Receiver/Reception |
| **TTX** | Transmission |
| **TTL** | Time To Live |
| **MOSI** | Master Out Slave In |
| **MISO** | Master In Slave Out |
| **SCK** | Serial Clock |

| | |
|---|---|
| **SDA** | Serial Data Access |
| **SCL** | Serial Clock |
| **TWI** | Two-Wire Interface |
| **SPI** | Serial Peripheral Interface |
| **AREF** | Analog Reference |
| **UNO** | One (in Italy) |
| **LCD** | Liquid Crystal Display |
| **RCO** | Rapport Cyclique d'Ouverture |
| **CAO** | Conception Assisté par Ordinateur |
| **HTML** | HyperText Markup Language |
| **PC** | Personal Computer |

# SOMMAIRE



## CHAPITRE 1: THÉORIE DU RADAR

## CHAPITRE 2 : ÉTUDE DE LA PARTIE MATÉRIELLE ET LOGICIELLE DU PROJET

**CHAPITRE 3 : RÉALISATION DU PROJET**

# Liste des figures

# Liste des tableaux

# INTRODUCTION GÉNÉRALE

# Introduction générale

Ces dernières années, il y a eu une augmentation constante de la demande pour une performance accrue des systèmes radar. Des nouvelles fonctionnalités en plus de la détection habituelle, ont été nécessaires pour des radars civils et militaires.

Le radar est un système de télédétection, qui est largement utilisé pour la surveillance, le suivi et les applications d'imagerie. Dès l'instant où le système radar est mis en marche, il devient électro-magnétiquement lié à son environnement qui présente une influence forte et continue sur les échos radars.

Radar est un système de détection d'objet, qui utilise des 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.

Le but principal de ce projet et l'étude et la réalisation d'un radar de détection, 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.

# CHAPITRE 1
# THÉORIE DU RADAR

## Introduction

Le principe du Radar était déjà connu et vérifié expérimentalement à la fin du 19$^{ieme}$ siècle, Son histoire de détection a débuté par les travaux du physicien britannique James Clerk Maxwell, en 1864, qui a prédit mathématiquement que les radiations, qui seront connues ensuite sous le nom d'ondes électromagnétiques, ont quelques propriétés communes avec les ondes lumineuses. En particulier, la vitesse de propagation et la réflexion par les objets métalliques et diélectriques. Cela conduit à se demander quelles sont les connaissances basiques, la pertinence de cette problématique s'est d'ailleurs confirmé au cours des travaux préparatoires de la présente étude pour l'intérêt de cette technologie.

Ce chapitre tend à expliquer graduellement la théorie du Radar dans sa forme générale.

## 1.2 Principes généraux

### 1.2.1 Principe du Radar

- **Définition du Radar**

Le Radar, 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 [2].

- **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.

### 1.2.2 Fréquences utilisées en système Radar

Au début du développement du Radar, un code alphabétique tel que S, X, L, etc…, a été employé pour désigner des bandes de fréquences Radar. Son but initial est de garder le secret militaire, les désignations ont été maintenues, probablement par habitude ainsi que la nécessité d'une nomenclature courte et pratique. Cet usage devient maintenant une pratique

acceptée par des ingénieurs de Radar. Le tableau suivant énumère la nomenclature de la bande de fréquences Radar adoptée par l'IEEE. Ceux-ci sont liés aux bandes spécifiques attribuées par l'Union Internationale des Télécommunications pour le Radar [3].

| Longueur d'onde | nomenclature OTAN | | bandes radar | |
|---|---|---|---|---|
| 100 m<br>10 m<br>1 m | A<br>B<br>C | 0 à 250 MHz<br>250 à 500 MHz<br>0,5 à 1 GHz | <br>HF<br>VHF<br>UHF | <br>3 à 30 MHz<br>30 à 300 MHz<br>0,3 à 1GHz |
| 30 cm | D | 1 à 2 GHz | L | 1 à 2 GHz |
| 10 cm | E<br>F | 2 à 3 GHz<br>3 à 4 GHz | S | 2 à<br>4 GHz |
| 5 cm | G<br>H | 4 à 6 GHz<br>6 à 8 GHz | C | 4 à<br>8 GHz |
| 3 cm | I | 8 à 10 GHz | X | 8 à 12 GHz |
| 2 cm<br>1 cm | J<br><br>K | 10 à<br>20 GHz<br>20 à 40 GHz | Ku<br>K<br>Ka | 12 à 18 GHz<br>18 à 27 GHz<br>27 à 40 GHz |
| inférieures à 1 cm | L | 40 à 60 GHz | V | 40 à 70 GHz |
| | M | 60 à 100 GHz | W | 70 à 100 GHz |

Tableau 1.1 **:** Bandes de fréquences Radar [4]

### 1.2.3 Classification des Radars

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

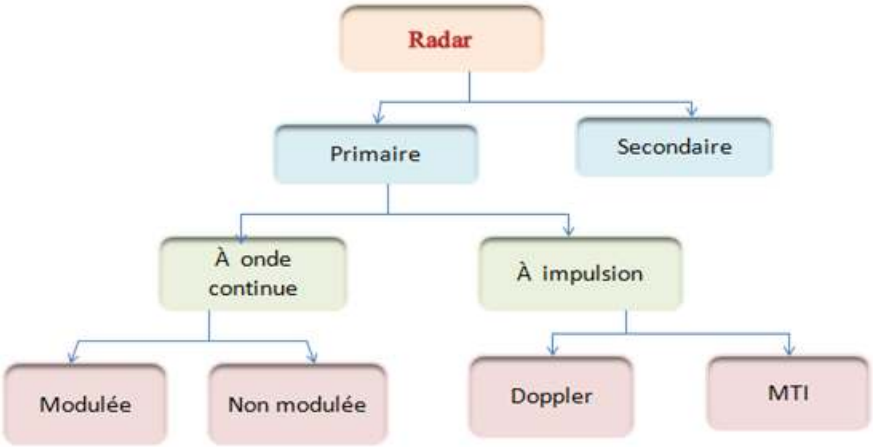



Figure 1.1 : Classification des Radars.

#### 1.2.3.1 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 [5].

##### 1.2.3.1.1 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 [5].
Deux grandes catégories de Radars à impulsions utilisant des décalages Doppler sont :

- **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 ambigüités de distance, mais ces Radars peuvent avoir des ambigüités Doppler [5].

- **Radars Doppler à impulsions**

Contrairement au Radar MTI, le Radar Doppler à impulsions utilise un PRF élevé pour éviter les ambigüités Doppler, mais il peut avoir de nombreuses ambigüités de distance [5].

##### 1.2.3.1.2 Radar à ondes continues

Les Radars CW transmettent continuellement un signal à haute fréquence et l'énergie réfléchie 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 :

- ✓ **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 différer entre deux cibles réfléchissantes [5].

- ✓ **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 f1 et f2. En examinant les fréquences réfléchies reçues (et avec la connaissance de la fréquence transmise), le calcul de la distance peut être effectué par [5]:

$$R = \frac{c \,.\, \Delta t}{2} \qquad (1.1)$$

Avec R: distance

$\Delta t$ : 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.

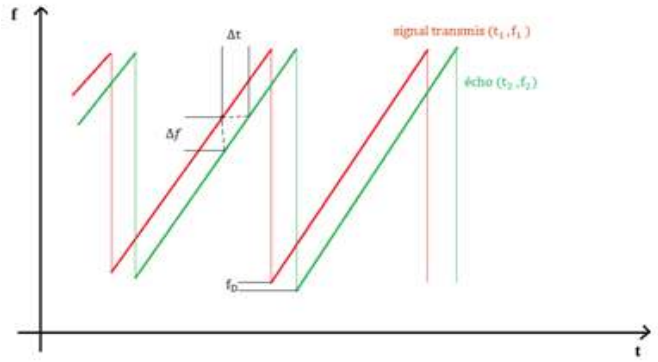



Figure 1.2 **:** Mesure de distance avec un Radar FMCW

### 1.2.3.2 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 ou 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) [5].

## 1.2.5 Radar à impulsions

### 1.2.5.1 Organisation (Description)

Ce type de Radar utilise la propriété des ondes électromagnétique de se réfléchir 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 $\tau$ de l'ordre de quelque microsecondes se propage dans l'atmosphère à la vitesse de la lumière (C =3.10$^{e}$8 m/s). Une partie de ce signal

est réfléchie 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 $\tau$ des impulsions et la fréquence de répétition PRF.

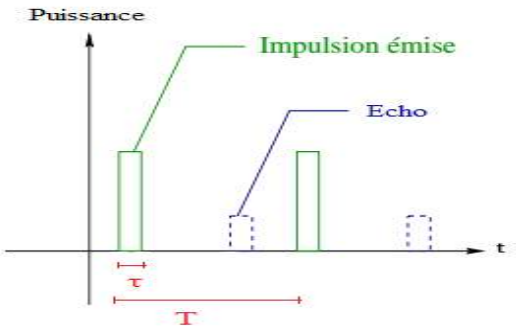



Figure 1.3 : Génération d'échos

### 1.2.5.2 Antenne, émetteur et récepteur

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

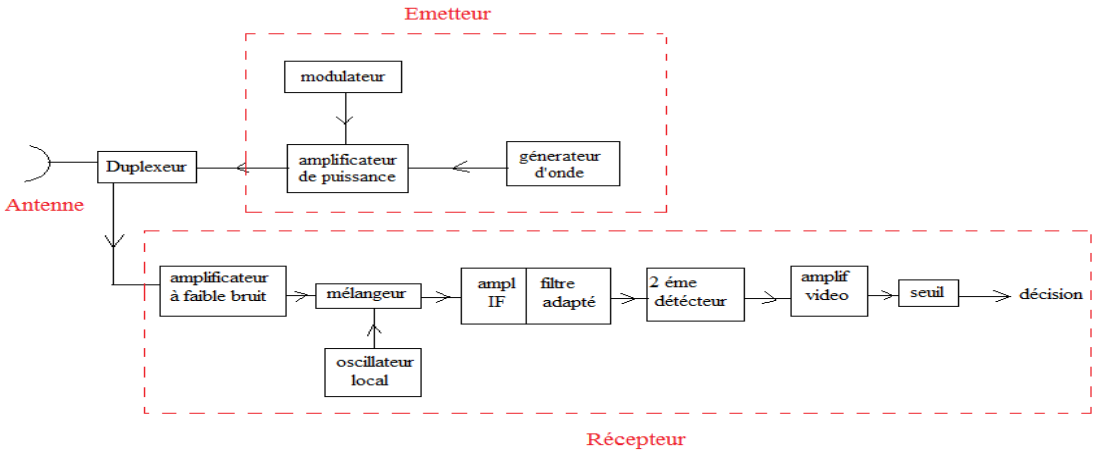



Figure 1.4 **:** Diagramme de bloc d'un Radar à impulsion

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. Le magnétron a été utilisé des différents générateurs de micro-ondes pour le Radar. 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 est un commutateur électronique, 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 lorsque on utilise un Radar monostatique, Il permet par conséquent d'utiliser la même antenne pour les deux fonctions.

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.

Si le Radar est utilisé pour la poursuite, certains moyens sont nécessaires pour détecter l'emplacement angulaire d'une cible en mouvement et permettre à l'antenne de se verrouiller automatiquement et de suivre la cible. Des dispositifs de surveillance sont habituellement inclus pour s'assurer que l'émetteur fournit l'impulsion de forme appropriée au niveau de puissance approprié et que la sensibilité du récepteur n'a pas dégradé. Des dispositions peuvent également être incorporées dans le Radar pour repérer les défaillances de l'équipement afin que les circuits défectueux puissent être facilement trouvés et remplacés.

Au lieu d'afficher la sortie d'un Radar directement sur le tube cathodique, elle peut d'abord être traitée par un dispositif de détection et de poursuite automatique (DPA) qui quantifie la couverture Radar en cellules de résolution de portée azimutale, ajoute (ou intégrer) Toutes les

impulsions d'écho reçues, établit un seuil qui permet aux échos à passer en rejetant le bruit, établit et conserve les trajectoires de chaque cible, et affiche les informations traitées à l'opérateur. Ces opérations d'un DPA sont généralement mises en œuvre avec la technologie informatique numérique [3].

### 1.2.5.3 Mesure de distance

Une manière de mesurer la distance à un objet est d'émettre une courte impulsion de signal radio, et de mesurer le temps que prendre l'onde pour revenir après avoir été réfléchie. La distance est la moitié du temps de retour de l'onde (Car le signal doit aller à la cible puis revenir) multipliée par la vitesse du signal (Qui est proche de la vitesse de la lumière dans le vide si le milieu traversé est l'atmosphère).

Quand l'antenne est à la fois émettrice et réceptrice (Ce qui est le cas le plus courant). Pendant que le signal est émis, on ne peut pas savoir si le signal mesuré est l'original ou l'écho. Cela implique qu'un Radar a une distance minimale, pour détecter des cibles plus proches, il faut utiliser une durée d'impulsion plus courte. La fréquence de répétition des impulsions (PRF) est également un facteur qui se traduit par la définition d'une distance maximale non-ambigüe au-delà de laquelle les échos arrivent après l'émission des impulsions suivantes, d'où une estimation erronée de la distance de cible. La distance non-ambigüe maximale, $R_{un}$, est égale à [1]:

$$R_{un} = \frac{c.T_p}{2} = \frac{c}{2.f_p} \qquad (1.2)$$

Où $T_p$ : la période de répétition de l'impulsion

$f_p$ : la fréquence de répétition des impulsion et c : la vitesse de la lumière.

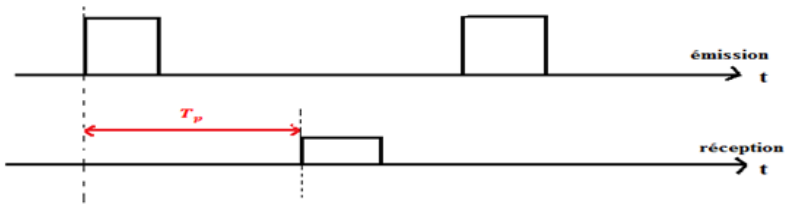



Figure 1.5 : Mesure de la distance

### 1.2.5.4 Mesure de la direction

La mesure de la direction est déterminer 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].

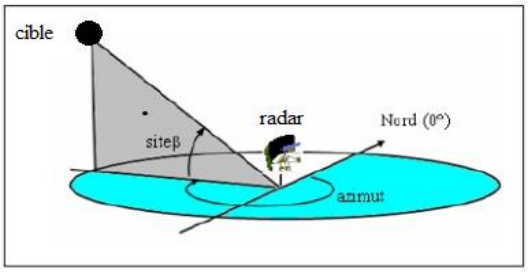



Figure 1.6 : Mesure de la direction

### 1.2.6 Radar à onde continue

#### 1.2.6.1 Organisation (Description)

Les Radars à ondes continues, comme leur nom l'indique, émettent des ondes électromagnétiques en permanence. Il y plusieurs types de Radar à ondes continues, le plus simple étant le Radar à ondes continues (CW) non-modulées. Sachant que l'émetteur transmet en même temps que les réflexions arrivent, ce Radar n'est capable que de visualiser les cibles mobiles (MTI).

Les versions modulées de ces Radars comprennent :

- Radar à ondes continues modulées en fréquence (FMCW),
- Radar à ondes continues et interrompues modulées en fréquence (IFMCW)
- Radar à ondes continues modulées en phase

✓ **Effet Doppler**

Physicien autrichien (1803-1853), Doppler étudie la modification apparente de la fréquence fixe émise par une source sonore, mobile par rapport à un observateur immobile, ou encore les modifications apparentes de la fréquence d'une source immobile lorsque l'observateur se déplace par rapport à cette source.

Bien que s'appliquant à une onde électromagnétique, l'effet DOPPLER trouve une explication plus aisée avec un signal acoustique. Une onde sonore de fréquence f, d'une source ponctuelle et immobile, se propage dans le plan sous forme de cercles concentriques

séparés chacun d'une longueur d'onde. La longueur d'onde est identique quelle que soit la position de l'observateur. Celui-ci entendra donc un son de fréquence f constante.

Si la source sonore est en mouvement, la compression des couches d'air introduit une variation de la longueur d'onde.

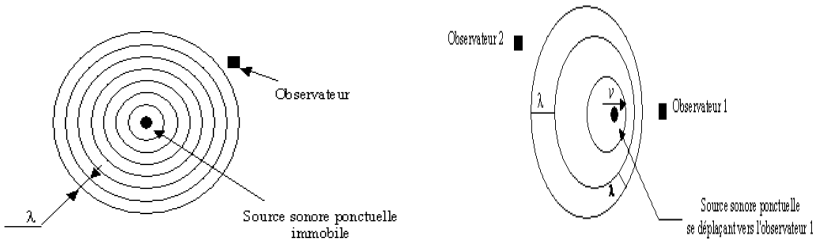



Figure 1.7 : Variation de fréquence d'une onde sonore pour une source mobile et immobile

La source sonore est à vitesse v constante, comme $\lambda = v/f$ et donc $f = v/\lambda$ une diminution de $\lambda$ se traduit par une augmentation de f et inversement. D'où l'observateur (1) entend donc un son plus aigu que l'observateur (2).

Les Radars à effet Doppler sont souvent employés par la police pour déterminer la vitesse des voitures [7].

### 1.2.6.2 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 La plupart du temps car sa réalisation est plus simple.

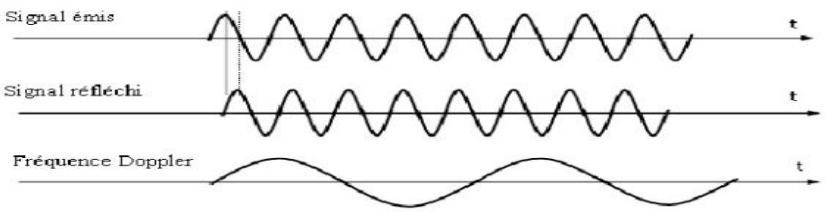



Figure 1.8 : Signaux Radar continu

Le mouvement de la cible cause la déviation de la fréquence de l'écho par :

$$f_d = \frac{2.V_r}{\lambda_o} \qquad (1.3)$$

Où $f_d$ est la fréquence Doppler, $V_r$ est la vitesse radiale entre la cible et le Radar et $\lambda_0$ est la longueur d'onde dans l'espace libre.

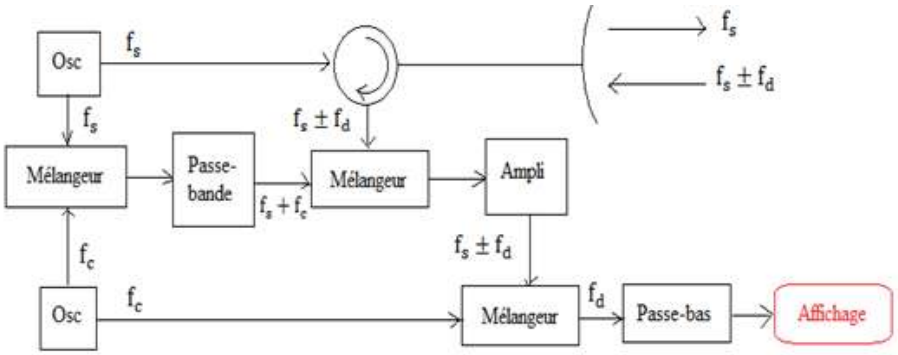



Figure 1.9 : Diagramme de bloc d'un Radar à onde continue

### 1.2.7 Performances des Radars

Pour fonctionner correctement, un Radar doit envoyer une très grande quantité d'énergie concentrée dans une direction pendant un temps très court (1 ms) et être assez sensible pour détecter la partie renvoyée sous forme d'écho.

Pour calculer les performances d'un Radar, on remplace les cibles réelles par des sphères métalliques de rayon R qui génèrent le même écho Radar [7].

#### 1.2.7.1 Ambiguïté en distance et vitesse

✓ **Ambiguïté sur la mesure de la distance**

L'écho (1) en réception correspond t-il à l'émission du train d'onde (1), du train d'ondes (2) ou d'un précédent, alors il y a une ambiguïté.

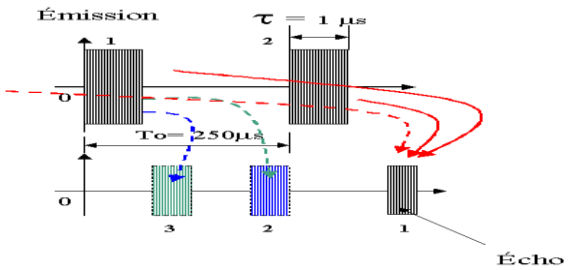



Figure 1.10 : Ambiguïté en distance

Pour l'émission du train d'ondes (1), seuls les échos reçus pendant le temps $T_0 - \tau \simeq T_0$ peuvent être pris en compte échos (2 et 3).

La distance maximum sans ambigüité est donc : $d_{max} = c.T_0/2 = c/(2.PRF)$

Les Radars Doppler utilisent la même antenne pour l'émission et la réception, de ce fait, pendant la durée de l'émission, la réception est impossible.

Cette distance minimale définit la zone aveugle du Radar $d_{min} = c.\tau/2$ .

Il est également impossible de distinguer deux échos successifs, s'ils ne sont pas espacés de la durée $\tau$. Cela se traduit par une segmentation des distances.

Le plus petit intervalle de distance mesurable est donc égal à $\Delta d = c.\tau/2$ , cette résolution des distances $\Delta d$ est appelée résolution radiale [7].

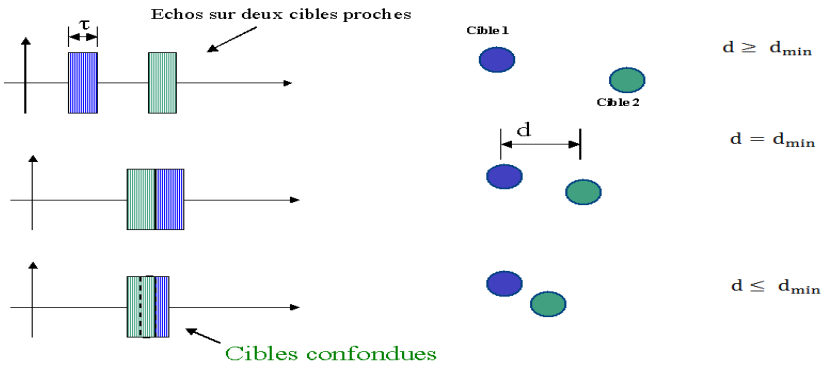



Figure 1.11 : Détection de deux cibles

✓ **Ambiguïté sur la mesure de la vitesse**

L'ambiguïté de la fréquence Doppler, exprime la possibilité d'attribuer différentes valeurs de la vitesse radial $v_r$ à une fréquence donnée Doppler $f_d$, c'est le résultat du caractère périodique du spectre Doppler dans des Radars à impulsions [7]. Le maximum de la fréquence non ambiguë Doppler est :

$$f_d = \pm \frac{f_r}{2} \qquad (1.4)$$

La vitesse radiale maximale non ambigüe sous la forme suivante avec $f_r$ est une fréquence de récurrence :

$$v_r = \pm \frac{\lambda\, f_r}{4} \qquad (1.5)$$

### 1.2.7.2 Résolutions des Radars

La puissance réfléchie ou rétrodiffusée par une cible atmosphérique est proportionnelle à la puissance crête émise pour chaque impulsion et au cycle de travail $\tau/T_0$. En conséquence, plus la puissance émise est importante, plus la distance à laquelle cette cible pourra être détectée sera grande. D'autre part, si le cycle de travail diminue, la portée Radar s'en trouve

affectée. Pour obtenir une portée Radar importante, il est nécessaire d'envoyer des impulsions longues (Mais dans ce cas, la résolution radiale et la zone aveugle du Radar augmentent), ou de diminuer la période de répétition des impulsions To (Mais dans ce cas la distance maximale ambiguë diminue) [7].

La résolution radiale d'exprime par :

$$\Delta d = c.\tau/2$$

Supposant un signal de fréquence F pulsé avec une PRF, $F_0 = 1/T_0$ , la vitesse de Nyquist est :

$$\left|V_{r_{max}}\right| \leq (\lambda/4).F_0 \qquad (1.6)$$

Ce qui traduit donc par une fréquence Doppler :

$$\Delta F = 2.v.F/c = F_0/2. \qquad (1.7)$$

Si l'on souhaite avoir une précision sur la mesure de la vitesse $dv = 0.1\,m/s$ , cela induit une variation Doppler :

$$d(\Delta F) = dF = 2.dv.F/c \qquad (1.8)$$

Si l'on utilise un échantillonnage suivi d'une FFT pour connaitre le spectre du signal démodulé en réception, la résolution fréquentielle $dF = 1/\Delta T$ avec $\Delta T$, la durée de la mesure.

$$dF = 1/\Delta T = 1/(N.T_0) = F_0/N \qquad (1.9)$$

Avec : N le nombre d'échantillons de la FFT.

La résolution en vitesse s'exprime par : $dv = 2.\left|V_{r_{max}}\right|/N$

$$dv = \frac{(\lambda . F_0)}{2}.N = (\lambda/2).dF \qquad (1.10)$$

### 1.2.7.3 Précisions des mesures

La mesure de l'instant d'arrivée des échos est d'autant plus précise lorsque le rapport signal sur bruit est élevé.

La précision des mesures sont données par les formules de Woodward est inversement proportionnelle au rapport signal sur bruit [6].

Les différentes précisions sont exprimées comme suite :

- Distance $\sigma_R = \frac{c}{2\,.\,\Delta f\sqrt{S/N}}$ [m] (1.11)
- Vitesse $\sigma_V = \frac{\lambda}{2\,.\,T_m\sqrt{S/N}}$ [m/s] (1.12)
- Angle $\sigma_\theta = \frac{\lambda}{2\,.\,D\sqrt{S/N}}$ [rad] (1.13)

Avec :

$\sigma_R, \sigma_V$ et $\sigma_\theta$ : Incertitudes de distance, vitesse et direction.

$\Delta f$ : Largeur de bande de l'impulsion émise.

$T_m$ : Durée de mesure.

D : Dimensions de l'antenne.

En pratique, les valeurs d'incertitudes des mesures sont :

- En distance (100 m).
- En direction (0.35°) en azimut et (0.15°) en élévation [6].

## 1.3 Equation du Radar

### 1.3.1 Etablissement de l'équation du Radar en espace libre

#### 1.3.1.1 Propagation des ondes Radar

La présence du sol et de l'atmosphère modifie les performances du Radar. Ou les ondes réfléchies peuvent arriver en phase avec l'onde directe et dans ce cas, elles viennent renforcer celle-ci, mais elles peuvent aussi arriver avec une phase telle que l'interférence soit destructive. La portée pratique en relation avec la courbure de la terre, ou elle est limitée à la zone optique.

La distance maximale $d_{max}$ vaut :

$$d_{max} = \sqrt{2.k.R.h} + \sqrt{2.k.R.H} \quad (1.14)$$

Ou H : Hauteur de l'objet vu par le Radar

h : Hauteur du Radar

k : Décrit l'effet de réfraction atmosphérique (K= 4/3)

R : Rayon de la terre égale à 6366 Km

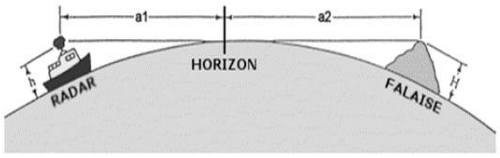



Figure 1.12 : Zone optique d'un Radar

Les principaux absorbants des ondes Radar dans l'atmosphère sont l'oxygène et la vapeur d'eau. La présence de gouttelettes d'eau ou de cristaux de glace va fortement augmenter l'absorption atmosphérique particulièrement aux plus hautes fréquences.

La meilleure solution pour ne pas être trop gêné est de travailler à fréquence suffisamment basse pour que la longueur d'onde soit nettement supérieure au diamètre des particules, dans ces conditions, l'onde est peu atténuée et les particules sont en quelques sortes invisibles [8].

### 1.3.1.2 Surface équivalente Radar

La SER ou RCS, acronyme anglais (Radar Cross Section) : Propriété physique inhérente des objets indiquant l'importance relative de la surface de réflexion d'un faisceau électromagnétique qu'ils provoquent. Elle est en fonction de la forme de l'objet, de la nature de ses matériaux constitutifs ainsi que de la longueur d'onde, des angles d'incidence et de réflexion du rayonnement.

Un objet de SER 'σ', se comporte comme, si il captait 'σ' fois la densité d'énergie dans la quelle il est baignée, et la réémettait de manière omnidirectionnelle pour que le Radar peut la détecter. Elle est défini par :

$$\boldsymbol{\sigma = \frac{4 \,.\, \pi \,.\, r^2 S_r}{S_t}} \qquad [\mathbf{m^2}] \tag{1.15}$$

Avec :

$\boldsymbol{\sigma}$: Capacité d'objet de rétrodiffuser vers le Radar

$\mathbf{S_t}$ : Énergie rétrodiffusée par l'objet en $[\mathbf{W/m^2}]$

$\mathbf{S_r}$ : Énergie reçue par l'objet en $[\mathbf{W/m^2}]$

La SER est donc le rapport entre l'énergie reflétée dans la direction du Radar par un objet et celle d'une sphère lisse de ($1m^2$) émettant de manière omnidirectionnelle.

La figure suivante montre l'équation de 'σ' pour les différentes formes pour représenter une cible [8][10].

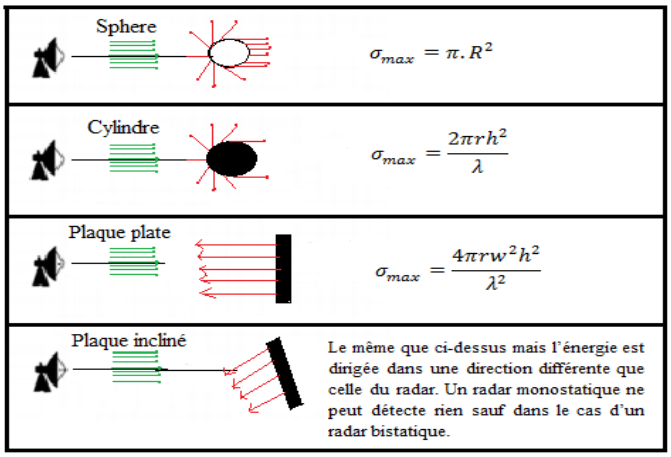



Figure 1.13 : Relation entre la forme de la cible et surface équivalente Radar

La SER dépend fortement de la fréquence f de fonctionnement de Radar, lorsque f est grand la SER est importante et la détection des petites cibles se faire par utilisation des fréquences élevées.

### 1.3.1.3 Equation Radar

L'équation du Radar est un bilan des puissances sur le trajet aller-retour d'une onde émise, intervenant les caractéristiques d'émetteur, récepteur, des antennes d'émission et de réception, de cible et de l'environnement.

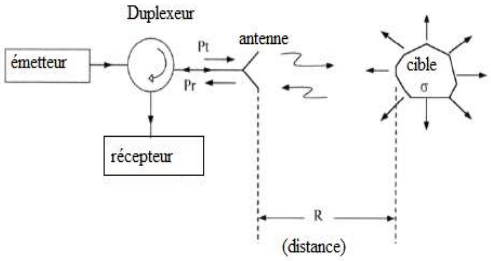



Figure 1.14 : Schéma synoptique d'un Radar

On supposera que les ondes électromagnétiques se propagent dans des conditions idéales, sans subir une quelconque perturbation. Et on considère un Radar monostatique doté d'une antenne directive utilisée en émission et en réception de gain max G et de surface équivalente $A_e$ [9].

$$\begin{cases} G = \frac{4 . \pi . A_e}{\lambda^2} \\ A_e = \eta . A \end{cases}$$

Avec $\lambda$ : Longueur d'onde
$\eta$ : Efficacité de l'antenne
A: Surface de l'ouverture de l'antenne

- La densité de puissance dirigée vers la cible par le Radar avec antenne isotrope est :

$$\frac{P_t}{4.\pi.\ R^2} \quad (1.16)$$

- Pour une antenne directive $\frac{P_t}{4\pi R^2} G_t$ (1.17)

- Puissance réémise par la cible $\frac{P_t G_t}{4\pi\ R^2} \sigma$ (1.18)

- Densité de puissance au niveau du Radar : $\frac{P_t G_t}{4\pi R^2} \frac{\sigma}{4\pi R^2}$ (1.19)

- Puissance reçue par le Radar : $\frac{P_t\ G_t}{4\pi\ R^2} \frac{\sigma}{4\pi R^2} A_e$ d'où $\frac{P_t G_t}{4\pi R^2} \frac{\sigma}{4\pi R^2} \frac{G_r \lambda^2}{4\pi}$ (1.20)

- Puisque le Radar est monostatique, donc $G_t = G_r = G$ et la puissance reçue devient

$$P_r = \frac{P_t\ G^2\ \sigma\ \lambda^2}{(4\pi)^3\ R^4} \quad (1.21)$$

Si on suppose la puissance reçue minimum détectable par le récepteur notée $S_{min}$, $R_{max}$ est la distance maximale de détection, et l'Equation Radar devient comme suite :

$$R_{max} = \left(\frac{P_t\ G^2\ \sigma\ \lambda^2}{(4\pi)^3\ S_{min}}\right)^{1/4} \quad (1.22)$$

Avec $P_t$ est la puissance transmise, si on utilise un Radar à impulsion, cette puissance définit par une puissance crête sinon une puissance moyenne d'où,

$$P_{av} = \frac{P_t\ \tau}{T_0} = P_t.\tau.PRF \quad (1.23)$$

### 1.3.2 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).

Pour cela on introduit un facteur de pertes $L_{sys}$ dans l'Equation Radar pour tenir compte ces pertes [6].

$$R_{max} = \left( \frac{P_t \,.\, G^2 \,.\, \sigma \,.\, \lambda^2}{(4 \,.\, \pi)^3 \,.\, L_{sys} \,.\, S_{min}} \right)^{1/4} \tag{1.24}$$

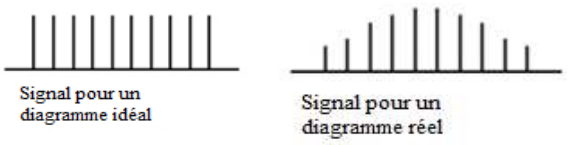



Figure 1.15 : Effet de pertes sur le signal reçu

### 1.3.3 Portée maximale du Radar en présence du bruit thermique

#### 1.3.3.1 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 générait 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 $\mathbf{b_n}$ (en hertz) à une température **T** (en Kelvin) est égale à [12]:

$$P_{th} = k \,.\, T \,.\, b_n \tag{1.25}$$

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

$$b_n = \frac{\int_{-\infty}^{+\infty} |H(f)|^2 \ df}{|H(f_0)|^2} \quad (1.26)$$

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

$$F_n = \frac{bruit\ à\ la\ sortie\ d'un\ récepteur\ réel}{bruit\ à\ la\ sortie\ d'un\ récepteur\ idéal} = \frac{S_i/N_i}{S_0/N_0} \quad (1.27)$$

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

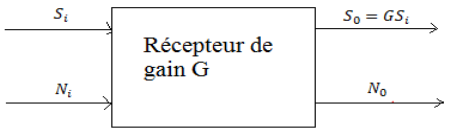



Figure 1.16 : Différents types de bruit traités par un récepteur

Avec $N_i = k.T.b_n$

Le facteur de bruit devient : $F_n = \frac{N_0}{G.k.T.b_n}$ (1.28)

### 1.3.3.2 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 ou la réponse est réduite à (0.707 donc 3 dB). La largeur de bande à 3 dB est largement utilisée grâce à facilité de mesure car elle nécessite une connaissance complète de la caractéristique de réponse H(f) [6].

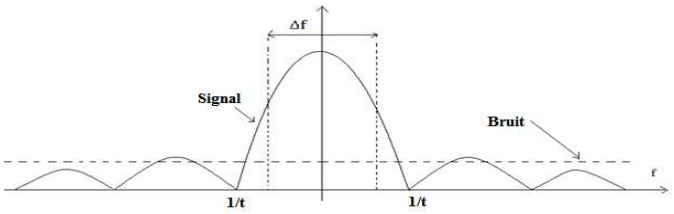



Figure 1.17 : Spectre du signal impulsionnel et du bruit « «Blanc »

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 de bruit, et vice versa.

D'après les calcules, 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].

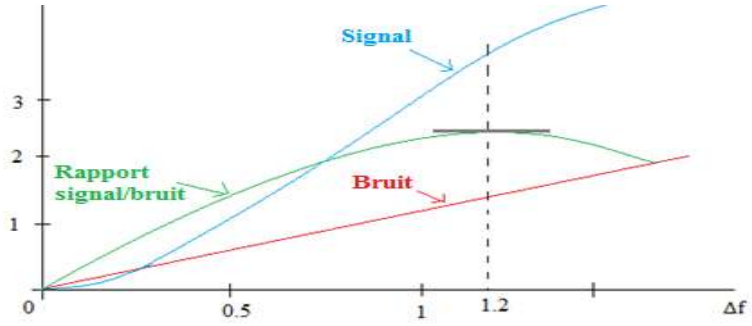



Figure 1.18 : Variation du SNR en fonction de largeur de bande

### 1.3.3.3 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 $S_n(f)$ sont 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 [11].

On applique à l'entrée : $x(t) = s(t) + n(t), \quad t \in [0, T]$

En utilisant un filtrage linéaire de réponse impulsionnelle h(t) :

$$y(t) = y_s(t) + y_n(t)$$

$$y(t) = s(t) * h(t) + n(t) * h(t)$$

Le rapport signal/bruit en sortie du filtre à $t_0$ est défini par :

$$\mathrm{SNR}(t_0) = \frac{y_s^2(t_0)}{E\left[y_s^2(t_0)\right]} \tag{1.29}$$

Si H(f) désigne la réponse impulsionnelle de ce filtre et signal d'entrée S(f) = TF[s(t)], des calcules élémentaires permettent d'obtenir :

$$\mathrm{SNR}(t_0) = \frac{y_s^2(t_0)}{E[y_s^2(t_0)]} = \frac{\left|\int_{-\infty}^{+\infty} H(f)\,.\,S(f)\,.\,e^{j2\pi f t_0}\,df\right|^2}{\int_{-\infty}^{+\infty}|H(f)|^2\,.\,S_n(f)\,df} \tag{1.30}$$

Avec numérateur : $y_s(t) = TF^{-1}[S(f)\,.\,H(f)] = \int_{-\infty}^{+\infty} H(f)\,.\,S(f)\,.\,e^{j2\pi f t}\,df$

Dénominateur : $S_{y_n} = S_n(f)\,.|H(f)|^2$

Donc $P_{y_n} = E[y_s^2(t_0)] = \int_{-\infty}^{+\infty} S_n(f)\,.|H(f)|^2 df$

Après certain vérification, $H(f) = K.\frac{S^*(f)}{S_n(f)}\,.e^{-j2\pi f t_0}$

Pour un bruit blanc de DSP $S_n(f) = \frac{N_0}{2}$, on a donc $H(f) = KS^*(f)e^{-j2\pi f t_0}$

D'où $h(t) = Ks^*(t_0 - t)$ Avec $K = 2\,.\,k/N_0$

t : Temps de retard correspond à la distance de cible.

La réponse impulsionnelle du filtre est donc le signal renversé translaté. De plus, le rapport signal à bruit maximal vérifie :

$$SNR(t_0)^{max} = \int_{-\infty}^{+\infty} \frac{2}{N_0}\,.|S(f)|^2\;\; df = \frac{2E}{N_0}$$

Ou 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 [11]:

$$\mathrm{SNR}(t_0)^{\max} = \frac{2\,.\,E}{N_0} \tag{1.31}$$

#### 1.3.3.4 Expression de la portée maximale en fonction du bruit

En appelant$T_e$, la température effective de bruit de l'antenne, la puissance de bruit à l'entrée

du récepteur est :

$$N_i = K \,.\, \mathrm{T_e} \,.\, \mathrm{b_n}$$

Comme $\mathrm{F_n} = \frac{\mathrm{S_i/N_i}}{\mathrm{S_0/N_0}}$ D'où $\mathrm{S_i} = \mathrm{F_n}.\mathrm{N_i}\left(\frac{\mathrm{S}}{\mathrm{N}}\right)_0$

Alors $\mathrm{S_{min}} = \mathrm{F_n}.\mathrm{K}\,.\,\mathrm{T_e}\,.\,\mathrm{b_n}\left(\frac{\mathrm{S}}{\mathrm{N}}\right)_{\mathrm{min}}$

Remplaçant $S_{min}$ dans l'équation Radar, on obtient [6],

$$R_{max} = \left(\frac{P_t \,.G^2\,.\,\sigma\,.\,\lambda^2.\,\tau}{(4.\pi)^3\,.\,\mathrm{L_{sys}}\,.\,\mathrm{F_n}\,.K\,.\,\mathrm{T}\,.\,(\mathrm{e}1.2)\,.\,\left(\frac{S}{N}\right)_{min}}\right)^{1/4} \qquad (1.32)$$

### 1.3.4 Discussion de l'équation du Radar

L'équation représentée au dessus entraîne plusieurs remarques :

- La portée du Radar est proportionnelle à l'énergie du signal émise**E .**
- La portée est proportionnelle à la surface équivalente $'\sigma'$ de la cible.
- La portée dépend de la longueur d'onde de différentes manières :
  - ✓ Comme$\mathrm{R_{max}} = \mathrm{K}\sqrt{\mathrm{A_e}/\lambda}$, la surface d'antenne constante donc la distance varie comme $1/\sqrt{\lambda}$.
  - ✓ Comme$\mathrm{R_{max}} = \mathrm{K}\sqrt{\mathrm{G}.\lambda}$ , le gain d'antenne constant, la distance croit comme$\sqrt{\lambda}$.

D'autres paramètres comme les pertes atmosphériques, les puissances réalisables, la surface équivalente des cibles, dépendent aussi de la longueur d'onde.

Le choix de la fréquence affecte les paramètres suivants :

- ❖ Dimension : Plus la fréquence est élevée, plus la dimension du Radar est petite
- ❖ Puissance émise pour les basses fréquences sera plus facile à manipuler que pour les hautes fréquences
- ❖ Gain antenne / ouverture : Grand gain implique des fréquences élevées
- ❖ Atténuation atmosphérique : Moins de pertes pour les fréquences basses
- ❖ Bruit : Faible dans la bande 1-10 GHz
- ❖ La fréquence Doppler : augmente avec la fréquence.

Le choix de la forme d'onde :

- o Largeur de bande
- o PRF (distance ambigüe)
- o Puissance moyenne [6]

## 1.4 Détection Radar

### 1.4.1 Principe de la détection Radar -Test de Newman-Pearson

Le test de Neyman-Pearson dans les statistiques pour déterminer la validité d'une hypothèse statistique spécifiée. Par conséquent, en termes statistiques, il est prétendu être un test uniformément le plus puissant et est optimal, peu importe les probabilités a priori de signal et de bruit [3].

Ce principe de détection subit à deux hypothèses mutuellement exclusives :

$$\begin{cases} H_0 = n(t) & \text{Cible absente} \\ H_1 = s(t) + n(t) & \text{Cible présente} \end{cases}$$

Ces conditions sont inconnues au moment de la décision. A cause de bruit, cela peut conduire à quatre résultats possibles, chacun d'eux caractérisé par une certaine probabilité :

| Décision | Hypothèses vraies H | |
|---|---|---|
| | H0 (absence) | H1 (Présence) |
| H0 (absence) | (H0 H0) Décision correcte : Proba (1-Pfa) | (H0 H1) Décision incorrecte :Proba (1-Pd) |
| H1 (Présence) | (H1 H0) Décision incorrecte : Proba (Pfa) | (H1 H1) Décision correcte : Proba (Pd) |

Tableau 1.2 : Hypothèse de décision.

Avec :

$P_{fa}$ : Probabilité de fausse alarme défini par :

$$P_{fa} = \text{proba}\{\text{bruit} > seuil\ v_0\} = \int_{v_0}^{\infty} P_{H_0}(v)\, dv$$

$P_d$ : Probabilité de détection défini par :

$$P_d = \text{proba}\{\text{bruit} + \text{signal} > seuil\ v_0\} = \int_{v_0}^{\infty} P_{H_1}(v)\, dv$$

Le critère de Neyman-Pearson est bien adapté pour l'application Radar et est habituellement utilisé dans la pratique, d'où la décision est faite en comparant le signal reçu avec un seuil qui comporte le minimum possible de fausse alarme [6].

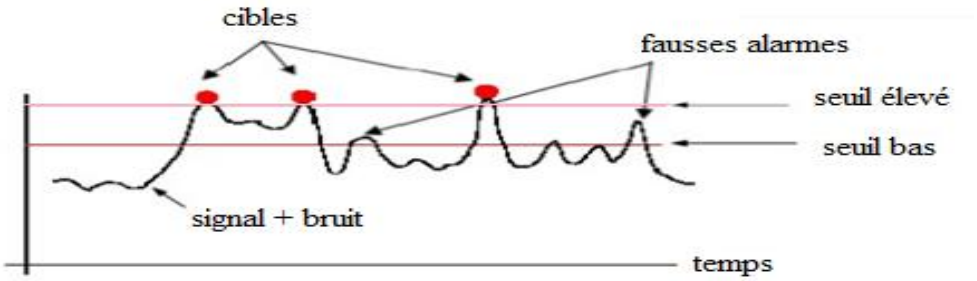



Figure 1.19 : Principe de détection

### 1.4.2 Détection d'une cible non fluctuante

Une cible non fluctuante est une cible qui conserve une SER constante.

Afin de déterminer les probabilités de détection et de fausse alarme dans ce cas, le récepteur comporte un détecteur d'enveloppe qui restitue l'enveloppe du signal entaché de bruit, puis comparé ensuite à un seuil pour décider de la présence ou non de la cible.

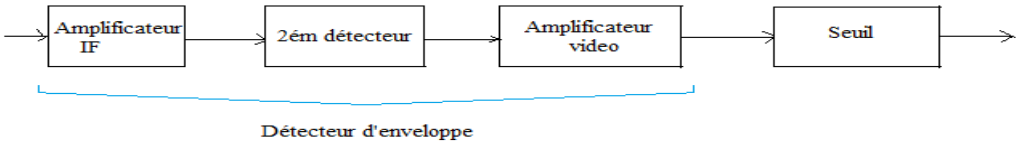



Figure 1.20 : détecteur d'enveloppe

La figure au dessus montre un détecteur d'enveloppe de récepteur Radar superhétérodyne, il se compose d'un amplificateur IF qui a une bande$B_{IF}$, amplificateur vidéo à une bande $B_V$ et un $2^{ém}$ détecteur (dispositif non linéaire). Un détecteur d'enveloppe exige deux conditions sont : $B_v \geq \frac{B_{IF}}{2}$ et $f_{IF} \gg B_{IF}$

La sortie de ce détecteur d'enveloppe est un enveloppe (modulation du signal IF) ou on va extraire l'information et rejeter la porteuse.

On suppose à l'entrée du récepteur un bruit (thermique) de Pdf gaussienne

de moyenne nulle: $$p(n) = \frac{1}{\sqrt{2.\pi.\Psi_n}}\, e^{\frac{-n^2}{2.\Psi_n}} \qquad (1.33)$$

Avec $\Psi_n = \sigma_n^2$ , la variance de la tension de bruit n(t).

Rice a montré que lorsqu'un bruit gaussien travers un filtre IF, l'enveloppe de la sortie exprimé par une Pdf de Rayleigh [3]. $$P_n(r) = \frac{r}{\Psi_n}\, e^{\frac{r^2}{2.\Psi_n}} \qquad (1.34)$$

#### 1.4.2.1 Probabilité de fausse alarme

On obtient une fausse alarme chaque fois que le bruit seul dépasse le seuil $r_0$ ,

$$P_{fa} = \int_{r_0}^{+\infty} \frac{r}{\Psi_n} e^{\frac{r^2}{2.\Psi_n}} dr \quad \text{soit : } P_{fa} = e^{\frac{-r^2}{2.\Psi_n}} \tag{1.35}$$

L'intervalle moyen $t_{fa}$ entre deux fausses alarmes est une valeur moyenne :

$$t_{fa} = \lim_{N\to\infty} \frac{1}{N} \sum_{K=1}^{N} T_K$$

$$P_{fa} \approx \frac{\tau}{t_{fa}} \approx \frac{1/\Delta f}{t_{fa}} \tag{1.36}$$

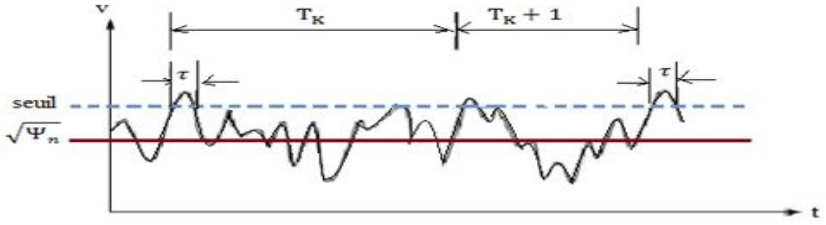



Figure 1.21 : Impulsion de bruit

En pratique, les $t_{fa}$ est très grands, d'où la probabilité $P_{fa}$ est très petite ($P_{fa} < 10^{-6}$) [3].

### 1.4.2.2 Probabilité de détection

Dans le cas d'une détection d'un objet, à l'entée de récepteur il y'a un signal d'information entaché de bruit gaussien. La densité de probabilité de l'enveloppe à la sortie de l'amplificateur vidéo est donnée par :

$$P_{s+n}(r) = \frac{r}{\Psi_n} . e^{\frac{-r^2+A^2}{2.\Psi_n}} . I_0 . \left(\frac{rA}{\Psi_n}\right) \tag{1.37}$$

Avec $I_0(z) = \frac{\exp(z)}{\sqrt{2\,\pi\, z}}(1 + \frac{1}{8z} + \cdots)$ La fonction de Bessel modifiée d'ordre 0.

A : l'amplitude de signal déterministe (écho) renvoie par la cible non fluctuante.

La probabilité de détection est la probabilité pour que à densité de probabilité de

l'enveloppe dépasse le seuil : $P_d = \int_{r_0}^{+\infty} P_{s+n}(r)dr$

$$P_d = \int_{r_0}^{+\infty} \frac{r}{\Psi_n} . e^{\frac{-r^2+A^2}{2.\Psi_n}} . I_0 . \left(\frac{rA}{\Psi_n}\right) dr \qquad (1.38)$$

La probabilité de détection $P_d$ n'est pas facile à calculer, pour cela il est préférable d'utiliser le SNR au lieu de $A^2/2\Psi_n$ dans l'équation de probabilité de détection [6].

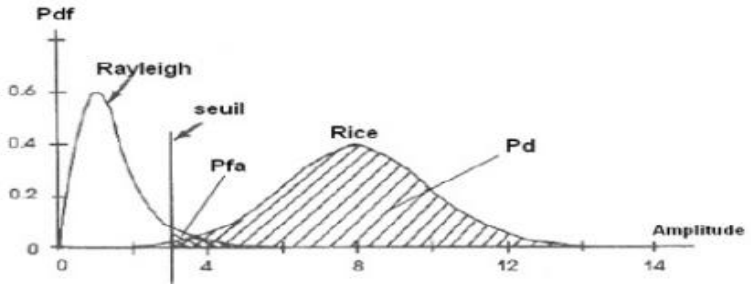



Figure 1.22 : Pdf du bruit et du bruit plus signal

La probabilité de détection peut être présentée en fonction de SNR et du rapport $V_T^2/2\Psi_n$

$$\begin{cases} P_d = f\left(SNR, \frac{V_T^2}{2\Psi_n}\right) \\ P_{fa} = f\left(\frac{V_T^2}{2\Psi_n}\right) \end{cases} \quad donc\ P_d = f(P_{fa})$$

Albersheim a développé une équation qui va relier$P_d$, $P_{fa}$ et le SNR comme suite [3] :

$$SNR = A + 0.12\,AB + 1.7\,B \qquad \text{Avec} \qquad A = \ln\left[0.62/P_{fa}\right] \qquad (1.39)$$

$$B = \ln[P_d(1 - P_d)] \qquad (1.40)$$

### 1.4.2.3 Intégration des échos

Lorsqu'une cible est éclairée par le faisceau du Radar, plusieurs impulsions (échos) ont capté par le récepteur pendant le balayage d'antenne, ce qui peut améliorer la probabilité de détection par une sommation des impulsions retournées.

Le nombre d'échos n retournés par la cible peut être s'exprimer par:

$$n = \frac{\varphi . PRF}{\theta_b} \qquad (1.41)$$

Où $\varphi$ : Largeur / l'ouverture d'antenne (degré)

PRF : fréquence de répétition (Hz)

$\theta_b$ : Taux de balayage d'antenne (degré/s)

Ce processus de sommation des échos s'appelle intégration des impulsions, qui peut être réalisé de différentes manières [6]:

- ✓ **Intégration cohérente**

Ce type d'intégration s'effectue avant le détecteur d'enveloppe, préserve La phase de chaque écho revenu. Si n impulsions sont parfaitement intégrées par le détecteur cohérent, le rapport signal/bruit est exactement multiplié par n [9].

- ✓ **Intégration non cohérente ou post-détection**

Cette intégration est située après le détecteur d'enveloppe, elle n'utilise que l'amplitude qui varie d'écho à l'autre sans corrélation. L'information de phase est perdue, l'efficacité de ce type d'intégration est moindre car il y a des pertes de détection qui diminue le SNR. Elle est calculée par Marcum, lorsque toutes les impulsions sont d'amplitude égale. Elle peut être définie comme suit:

$$E_i(n) = \frac{(S/N)_1}{n \,.\, (S/N)_n} \qquad (1.42)$$

Avec n : nombres des impulsions à intégrer

$(S/N)_1$ : Valeur du rapport signal / bruit d'une seule impulsion nécessaire pour produire une probabilité détection donnée (pour n = 1).

$(S/N)_n$ : Valeur du rapport signal / bruit par impulsion nécessaire pour produire la même probabilité de détection lorsque n impulsions sont intégrées [10][11].

- ✓ **Intégration binaire**

C'est une technique qui utilise une détection à double seuil, ou le signal est comparé à un premier seuil analogique puis codé par (1 ou 0). Le résultat des comparaisons de **N** échos successifs pour la même distance est mis en mémoire et leur somme est comparée à un deuxième seuil numérique **M** : Si la somme est supérieure ou égal à **M**, on décide la présence d'une cible.

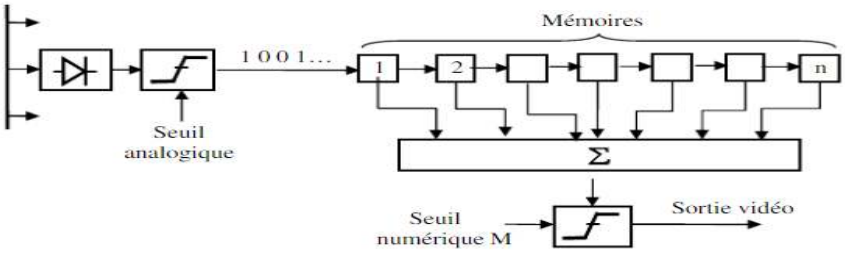



Figure 1.23 : Intégration binaire

### 1.4.3 Détection de signaux fluctuants

#### 1.4.3.1 Modèles de cibles

La cible est modélisée par une surface équivalente (SER) pour représenter la capacité de la cible de recevoir et émettre la puissance dans la direction de l'antenne de réception.

Il est possible de calculer la SER dans les cas simples, mais dans le cas complexe, il est difficile de mesurer la SER à cause d'incertitudes sur le mouvement de la cible, pour cela Swerling a proposé cinq modèles de fluctuation, ces modèles caractérisent la SER comme un processus aléatoire (la SER varie aléatoirement avec du temps) [6].

| Cas | Application | Type de fluctuation | Pdf |
|---|---|---|---|
| SW0 | | Non fluctuante | |
| SW1 | Plusieurs réflecteurs comparables (avion) | Lent : F'amplitude des n impulsions reçues est constante pendant le balayage (scan) mais varie de façon indépendante de balayage à balayage | $p(\sigma) = \frac{1}{\sigma_0} e^{-\frac{\sigma}{\sigma_0}}$ |
| SW2 | Plusieurs réflecteurs comparables (avion) | Rapide : Fluctuations indépendantes d'une impulsion à l'autre (Radar à agilité de fréquence) | $p(\sigma) = \frac{1}{\sigma_0} e^{-\frac{\sigma}{\sigma_0}}$ |
| SW3 | Cibles avec réflecteurs principal (missile) | Lent : L'amplitude des n impulsions reçues est constante pendant le balayage (scan) mais varie de façon indépendante de balayage à balayage | $p(\sigma) = \frac{4}{\sigma_0^2} e^{\frac{-2.\sigma}{\sigma_0}}$ |
| SW4 | Cibles avec réflecteurs principal (missile) | Rapide : Fluctuations indépendantes d'une impulsion à l'autre Radars à agilité de fréquence | $p(\sigma) = \frac{4}{\sigma_0^2} e^{\frac{-2.\sigma}{\sigma_0}}$ |

Tableau 1.3 : Modèles de cibles de Swirling

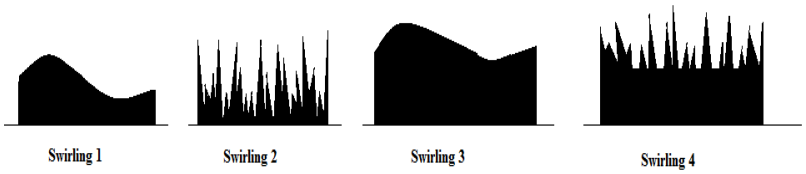



Figure 1.24 **:** Modèles de Swirling

### 1.4.3.2 Détection d'une cible fluctuante sur une impulsion

Les cibles fluctuantes nécessitent un rapport signal/bruit plus grand que les cibles non fluctuantes pour obtenir la même probabilité de détection. Par exemple, pour obtenir une **Pd** de (0.99) dans le cas de SW2, il faut augmenter de (17 dB) le rapport signal/bruit par rapport à une cible non fluctuante [6].

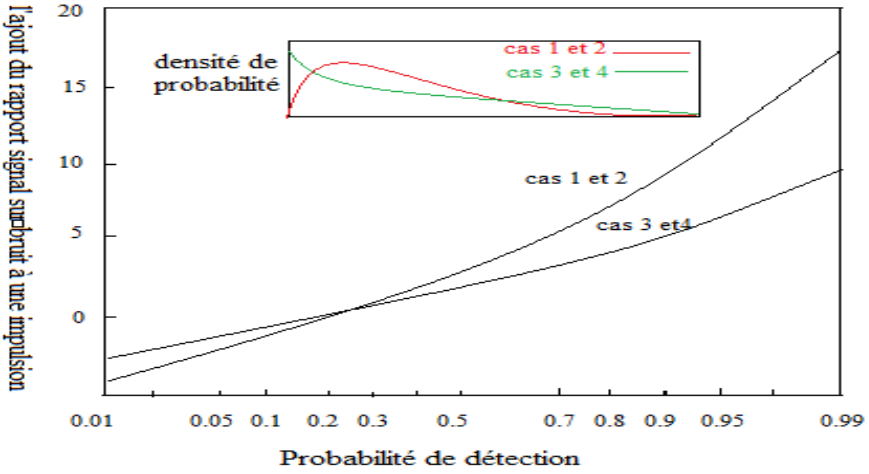



Figure 1.25 : Rapport signal sur bruit additionnel sur une cible non fluctuante

### 1.4.3.3 Calcul de la Pd d'une cible lentement fluctuante sur une Impulsion

Un grand nombre de réflecteurs indépendants est une caractéristique d'une cible de type Swerling 1, dont le signal réfléchi peut être représenté par une loi gaussienne. Dans la détection linéaire ou quadratique, l'enveloppe du signal (information + bruit) va suivre une loi de Rayleigh caractérisée par une pdf :

$$p(r) = \frac{r}{\Psi_s+\Psi_n} e^{\frac{-r^2}{2\,(\Psi_s+\Psi_n)}} \qquad (1.43)$$

Avec : $\Psi_s$ la variance du signal et $\Psi_n$ la variance du bruit [6].

La probabilité de détection est fournie par :

$$\mathrm{Pd} = \mathrm{Proba}(\mathrm{r} > \mathrm{r}_0) = \int_{\mathrm{r}_0}^{\infty} \frac{\mathrm{r}}{\Psi_\mathrm{s} + \Psi_\mathrm{n}} \mathrm{e}^{\frac{-\mathrm{r}^2}{2\,(\Psi_\mathrm{s}+\Psi_\mathrm{n})}} \mathrm{dr}$$

$$\mathrm{Pd} = \mathrm{e}^{-\frac{\mathrm{r}_0^2}{2\,(\Psi_\mathrm{s}+\Psi_\mathrm{n})}} \qquad (1.44)$$

On posant $(S/N) = \frac{\Psi_s}{\Psi_n}$ le rapport signal/bruit moyen pour une impulsion, on trouve :

$$Pd = e^{-\frac{\mathrm{r}_0^2}{2\ \Psi_n(S/N+1)}} \qquad (1.45)$$

Comme $Pfa = e^{-\frac{\mathrm{r}_0^2}{2\Psi_n}}$, alors : $\frac{\log(P_d)}{\log(P_{fa})} = \frac{1}{(S/N)+1}$

Où $$(S/N) = \frac{\log(P_{fa})}{\log(P_d)} - 1 \qquad (1.46)$$

Pour terminer un simple schéma illustre la détection en générale :

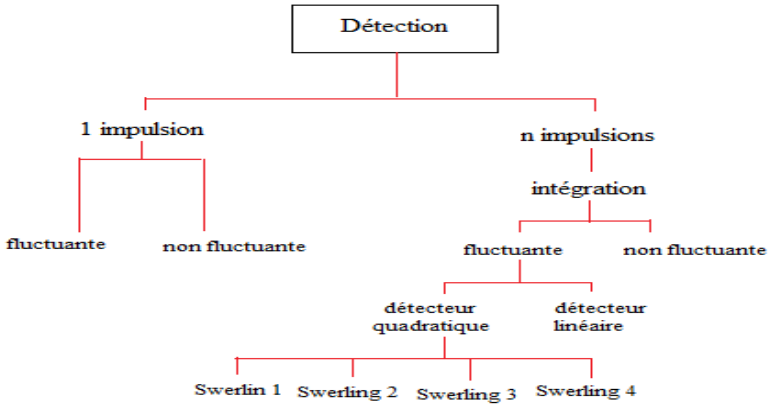



Figure 1.26 : Types de détection

### 1.4.4 Radar 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 ($> 60MHz$) pour que la réception être indépendante [6].

## 1.5 Applications du Radar

Les applications du Radar sont nombreuses et variées. Elles sont soit militaires citant les Radars de détection et de surveillance aérienne au sol ou embarqués les brouilleurs Radars, les satellites Radars d'observation de la terre, soit civile comme l'aéronautique ou on peut trouver les Radars de navigation, Radars de contrôle du trafic aérien, et Radars pour guidage d'approche d'Aéroport, aussi dans la maritime, la météorologie et en circulation et sécurité routière comme Radars de contrôle de la vitesse des automobiles [8].

## 1.6 Conclusion

Au terme de cette étude théorique, nous avons vu les principes du Radar ainsi nous avons pu aborder l'essentiel de notre travail concernant l'étude d'un Radar de détection, le chapitre suivant sera consacré sur les outils nécessaires pour la réalisation de notre projet.

# CHAPITRE 2

# ÉTUDE DE LA PARTIE MATÉRIELLE ET LOGICIELLE DU PROJET

## 2.1 Introduction

Avec le développement de matériel et de logiciel embarqué, diverses technologies sont de plus en plus intégrées. Son but est de simplifier les schémas électroniques et par conséquent réduire l'utilisation de composants électroniques, en réduisant ainsi le coût de fabrication d'un produit. et en résulte des systèmes plus complexes et performants pour un espace réduit. Au cours des dernières années, le mouvement open source du matériel est populaire dans le monde [13]. Arduino est un chef de file dans ce mouvement d'où les groupes d'utilisateurs répartis des ingénieurs aux étudiants, puis aux élèves de collège ou même les enfants de l'école primaire. L'émergence d'une variété de plates-formes matérielles open source réduit considérablement la courbe d'apprentissage, stimule l'innovation et accélère la conversion de l'idée à la réalisation.

## 2.2 Etude de la partie matérielle

### 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 [12].

Arduino est une plate - forme électronique open-source basé sur le matériel et le logiciel facile à utiliser. Les cartes Arduino sont capables de lire les entrées - La lumière sur un capteur, un doigt sur un bouton ou un message Twitter - et la transformer en une sortie - activation d'un moteur, d'allumer une LED, afficher une écriture. On peut commander la carte Arduino en envoyant un ensemble d'instructions au microcontrôleur disposé sur la carte. Pour ce fait, on utilise le langage de programmation Arduino (Basé sur le câblage) et le logiciel Arduino (IDE) , basé sur le traitement .

Au fil des années, Arduino a été le cerveau de milliers de projets, des objets de tous les jours à des instruments scientifiques complexes. Une communauté mondiale des décideurs, étudiants, amateurs, artistes, programmeurs et professionnels - sont rassemblés autour de cette plate - forme open source, leurs contributions ont ajouté une quantité incroyable de connaissances accessibles qui peuvent être d'une grande aide [13].

Toutes les cartes Arduino sont complètement open source, permettant aux utilisateurs de les construire de façon autonome et les adapter à leurs besoins particuliers.
Le logiciel IDE Arduino est aussi en open-source, et il se développe à travers les contributions des utilisateurs dans le monde entier. Le site officiel Arduino est : https://www.arduino.cc

### 2.2.2 Types des cartes Arduino

On peut classer les cartes Arduino en deux grandes familles.

- Cartes Arduino Officielles (Classique)
- Cartes Arduino Compatibles (Dérivées)

#### 2.2.2.1 Cartes Arduino Officielles (Classique)

Ces cartes sont fabriquées en Italie par le fabricant officiel : *Smart Projects,* sont site officiel est Arduino.cc ou Arduino.org , Pour tout ce qui est des cartes Arduino dites « Officielles» elles sont basées généralement sur le même micro-contrôleur AVR à savoir un ATmega328p du fabricant ATMEL [14].


Figure 2.1 : Carte Arduino officielle

#### 2.2.2.2 Cartes Arduino Compatibles (Dérivées)

Ces cartes ne sont pas fabriquées par *Smart Projects*, mais qui sont totalement compatibles avec les shields Arduino classiques (Mais pas avec l'IDE Arduino de base), elles sont fabriquées par diverse entreprises et commercialisées sous un nom différent (Exemples Cartes Arduino Compatible: Chinoises, Olimex, Selectronic, Freeduino, Seeduino, Femtoduino [14].


Figure 2.2 : Carte Arduino Compatible

### 2.2.3 Avantage de la carte Arduino UNO

Il y a de nombreuses cartes électroniques qui possèdent des plateformes basées sur des microcontrôleurs disponibles pour l'électronique programmée. Tous ces outils prennent en charge les détails compliqués de la programmation et les intègrent dans une présentation facile à utiliser [13]. De la même façon, le système Arduino simplifie la façon de travailler avec les microcontrôleurs tout en offrant aux personnes intéressées plusieurs avantages cités comme suit:

- ✓ Le prix (Réduit) : Les cartes Arduino sont relativement peu coûteuses comparativement aux autres plates-formes.

- ✓ Multi plateforme : Le logiciel Arduino, écrit en JAVA, tourne sous les systèmes d'exploitation Windows, Macintosh et Linux. La plupart des systèmes à microcontrôleurs sont limités à Windows.

- ✓ Un environnement de programmation clair et simple : L'environnement de programmation Arduino (le logiciel Arduino IDE) est facile à utiliser pour les débutants, tout en étant assez flexible pour que les utilisateurs avancés puissent en tirer profit également.

- ✓ Logiciel Open Source et extensible : Le logiciel Arduino et le langage Arduino sont publiés sous licence open source, disponible pour être complété par des programmateurs expérimentés. Le logiciel de programmation des modules Arduino est une application JAVA multi plateformes (Fonctionne sur tout système d'exploitation), servant d'éditeur de code et de compilateur, et qui peut transférer le programme à travers de la liaison série (RS232, Bluetooth ou USB selon le module).

- ✓ Matériel Open source et extensible : Les cartes Arduino sont basées sur les Microcontrôleurs Atmel ATMEGA8, ATMEGA168, ATMEGA328, les concepteurs des circuits expérimentés peuvent réaliser leur propre version des cartes Arduino, en les complétant et en les améliorant. Même les utilisateurs relativement inexpérimentés peuvent fabriquer la version sur plaque d'essai de la carte Arduino, dont le but est de comprendre comment elle fonctionne pour économiser le coût [16].

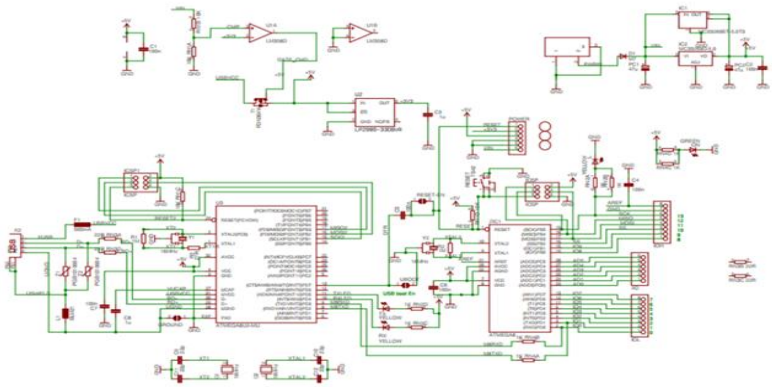


Figure 2.3 : Schéma de référence Arduino UNO [24]

### 2.2.4 Technologie 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 [17]. 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) |

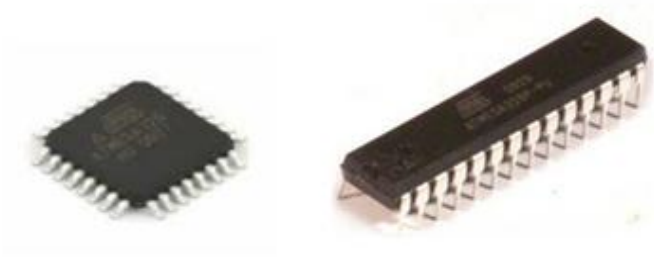
Tableau 2.1 : Caractéristiques de la Carte Arduino UNO

#### 2.2.4.1 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. Aujourd'hui, 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 [18].
La figure (2.4) montre deux types de microcontrôleur ATmega328, qu'on trouve sur les cartes Arduino.



Le composant CMS Le composant classique

Figure 2.4 **:** 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 [19].

#### 2.2.4.2 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 [20].

L'alimentation externe peut être soit un adaptateur secteur (Pouvant fournir typiquement de 3V à 12V sous 500mA) ou des piles. L'adaptateur secteur peut être connecté en branchant une

prise (2.1mm) positif au centre dans le connecteur jack de la carte. Les fils en provenance d'un bloc de piles ou d'accus peuvent être insérés dans les connecteurs des broches de la carte appelées Gnd (masse ou 0V) et Vin (Tension positive en entrée) du connecteur d'alimentation.

La carte peut fonctionner avec une alimentation externe de (6 à 20 volts). Cependant, si la carte est alimentée avec moins de (7V), la broche (5V) pourrait fournir moins de (5V) et la carte pourrait être instable. Si on utilise plus de (12V), le régulateur de tension de la carte pourrait chauffer et endommager la carte. Aussi, la plage idéale recommandée pour alimenter la carte Arduino 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). On peut alimenter la carte à l'aide de cette broche, ou, si 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 tout 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).

- **GND :** Broche de masse (Ou 0V) [20].

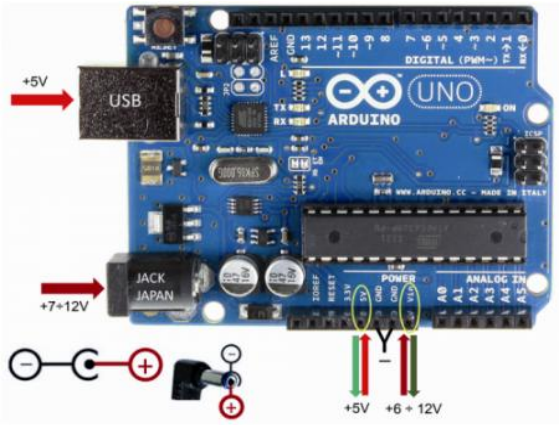



Figure 2.5: Sources de l'Alimentation de la carte Arduino UNO.

### 2.2.4.3 Entrées & sorties

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 et dispose d'une résistance interne de "Rappel au plus" (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) [24].

De plus, certaines broches ont des fonctions spécialisées (Voir annexe 1) :

- **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.
- **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().Les

applications de modulation de largeur d'impulsion (PWM) peuvent être trouvées dans le nombre d'applications, par exemple les télécommunications, le contrôle des servomoteurs, la régulation de la tension, la remise en puissance, etc. mesurer la largeur de PWM à l'aide d'un microcontrôleur, De plus, comment les capteurs à ultrasons (Qui peuvent être utilisés pour la mesure à distance) peuvent fonctionner conjointement avec PWM.

- **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.
- **I2C**: Broches 4 (SDA) et 5 (SCL). Supportent les communications de protocole I2C (ou interface TWI (TwoWire 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.

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 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. Voir figure 2.6.

**Note** : Les broches analogiques peuvent être utilisées en tant que broches numériques : Elles sont numérotées en tant que broches numériques de 14 à 19 [20].

La carte Arduino UNO intègre un fusible qui protège le port USB de l'ordinateur contre les surcharges en intensité (le port USB est généralement limité à (500mA) en intensité). Bien que la plupart des ordinateurs aient leur propre protection interne, le fusible de la carte fournit une couche supplémentaire de protection. Si plus de (500mA) sont appliqués au port USB, le fusible de la carte coupera automatiquement la connexion jusqu'à ce que le court- circuit ou la surcharge soit stoppé [13].

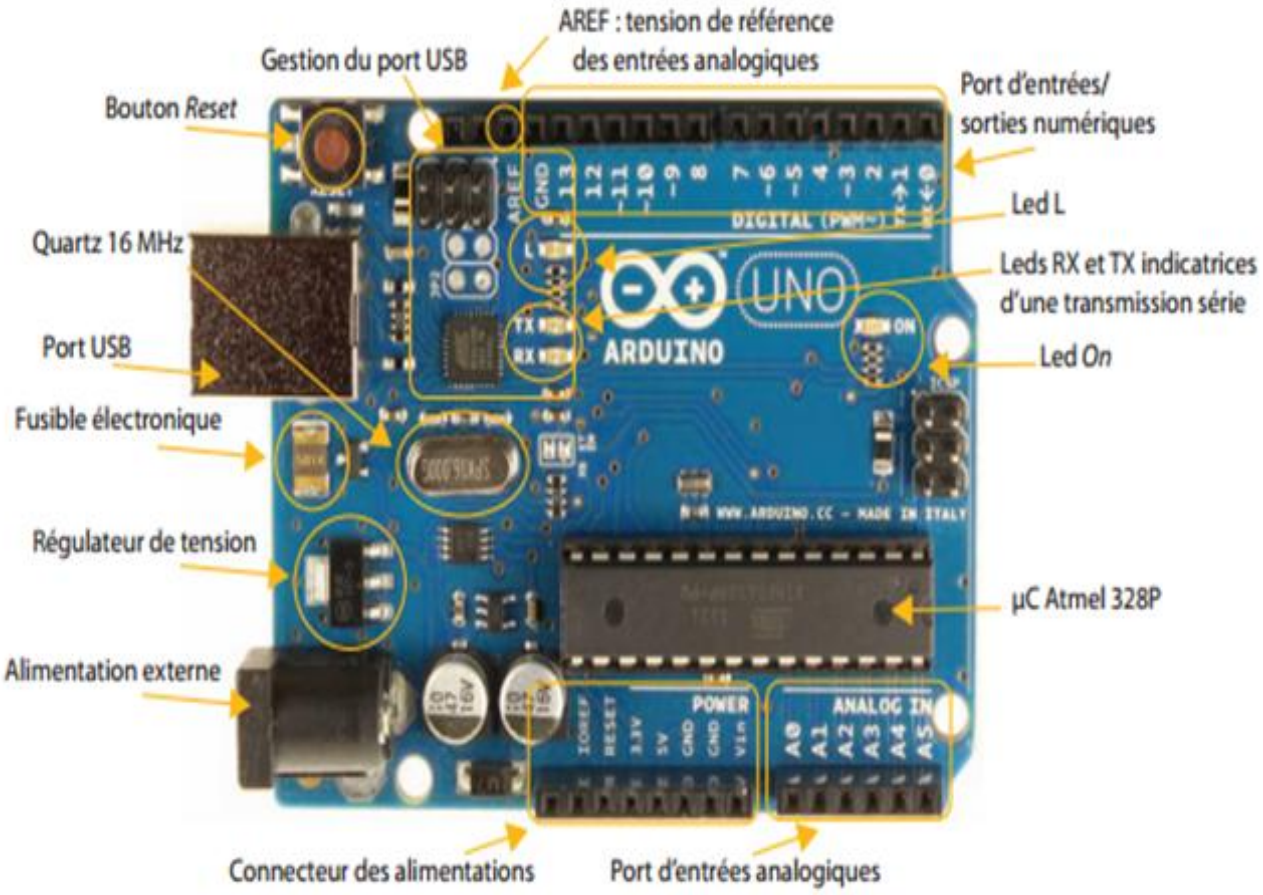



Figure 2.6 : Constitution de la carte Arduino UNO

### 2.2.4.4 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 [20].

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 (mais pas pour les communications série sur les broches 0 et 1).

Une librairie Série Logicielle permet également la communication série (limitée cependant) 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), la librairie pour communication SPI est disponible [20].

### 2.2.5 Capteur sonar à 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 [21]. Il fournit cette information grâce à une électronique à laquelle il est associé voir figure 2.7 .

Le capteur HC-SR04 utilise les ultrasons pour déterminer la distance d'un objet. Il offre une excellente plage de détection sans contact, avec des mesures de haute précision et stables. Son fonctionnement n'est pas influencé par la lumière du soleil ou des matériaux sombres, bien que des matériaux comme les vêtements puissent être difficiles à détecter [22].

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°). Le sonar HC-SR04 comprend un émetteur ultrasons, un récepteur ultrasons ainsi qu'un circuit de contrôle.

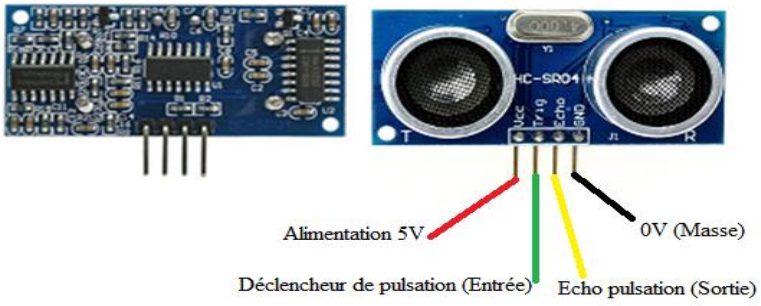



Figure 2.7 : Capteur Sonar à Ultrasons HC-SR04

### 2.2.5.1 Caractéristiques et spécification du capteur

Les caractéristiques en détail du Capteur sonar à Ultrasons HC-SR04 sont présentées dans le tableau ci-dessous [23].

| | |
|---|---|
| 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'entré 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 sonar à ultrason HC-SR04.

**Remarque** : La borne GND doit être connectée en premier, avant l'alimentation sur Vcc.

### 2.2.5.2 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 [19].

### 2.2.5.3 Fonctionnement

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

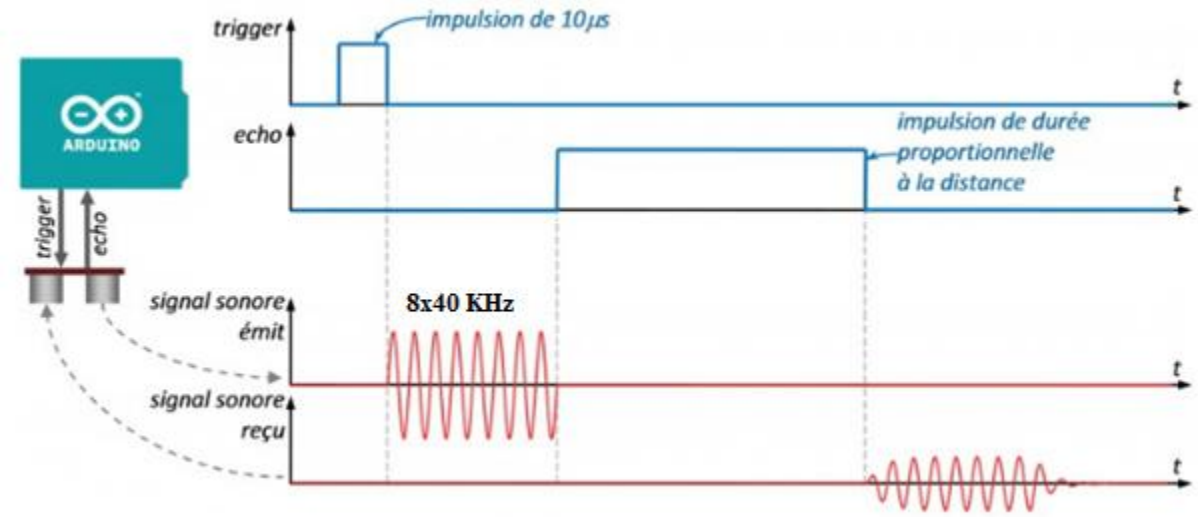



Figure 2.8: Signal d'entrée et sortie du capteur HC-SR04

### 2.2.5.4 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 µs) par le temps de propagation, soit :

$$d = v.t \qquad (2.1)$$

**d :** distance en mètre**; v :** vitesse de son mètre par seconde **; t :** temps en seconde.

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

$$d = \left(\frac{34000}{1000000}.10.\text{Valeur}\right).\frac{1}{2} \qquad [\text{cm}] \qquad (2.2)$$

En simplifiant :

$$d = \frac{17000}{1000\,000}.\text{Valeur} \qquad [\text{cm}] \qquad (2.3)$$

Finalement,

$$d = \frac{17}{100}.\text{Valeur} \qquad [\text{cm}] \qquad (2.4)$$

$$d = \frac{\text{durée}}{58} \qquad [\text{cm}] \qquad (2.5)$$

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 m² [19].

### 2.2.6 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 [13]. Deux modèles se rencontrent très fréquemment et sont les meilleurs marché, celui ayant 2 lignes et16 colonnes d'affichage et celui ayant 4 lignes et 20 colonnes d'affichage voir figure 2.9.


(a) (b)

Figure 2.9 : Afficheurs LCD (a) (16x2) (b) (20x4)

#### 2.2.6.1 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 [13].

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

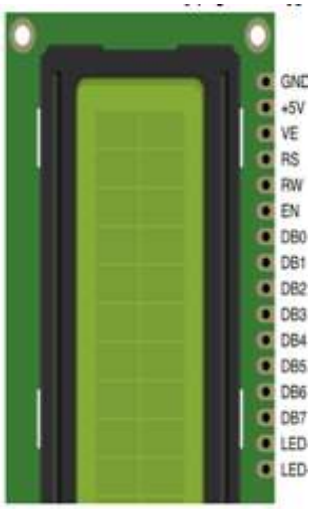



Figure 2.10 : Connecteur de l'afficheur LCD

Le tableau suivant présente les spécifications des broches et leur rôle [19] :

| 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

### 2.2.6.2 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 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 [19].

### 2.2.7 Servomoteur

Un servomoteur est un type de moteur électrique. C'est un dispositif typiquement utilisé en modélisation pour, par exemple, contrôler la direction d'une voiture télécommandée. Sur un servomoteur, l'angle de l'axe reste fixé dans une position et peut varier entre (0 et 180°) en fonction du signal envoyé [24].

Un servomoteur comprend :

- Un moteur électrique (continu), généralement assez petit.
- Des engrenages réducteur en sortie du ce moteur (pour avoir moins de vitesse et plus de couple ou de force).
- Un capteur type "potentiomètre" raccordé sur la sortie.
  - ✓ Il s'agit donc d'une résistance qui varie en fonction de l'angle, ce qui permet de mesurer l'angle de rotation sur l'axe de sortie.
  - ✓ Un asservissement électronique pour contrôler la position/rotation, de cet axe de sortie pour le maintenir à la bonne position.

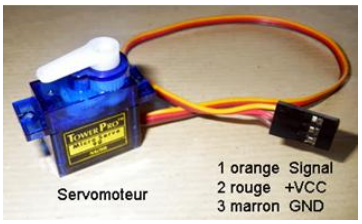



Figure 2.11: Servomoteur à rotation angulaire [24].

#### 2.2.7.1 Fonctionnement

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).

Les différentes caractéristiques du servomoteur sont présentées dans la figure ci-dessous [26].

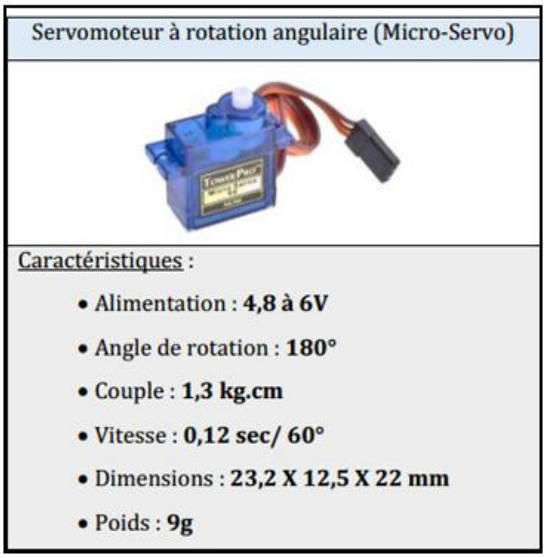



Figure 2.12 : Caractéristiques du servomoteur à rotation angulaire (Micro-Servo) [26].

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].

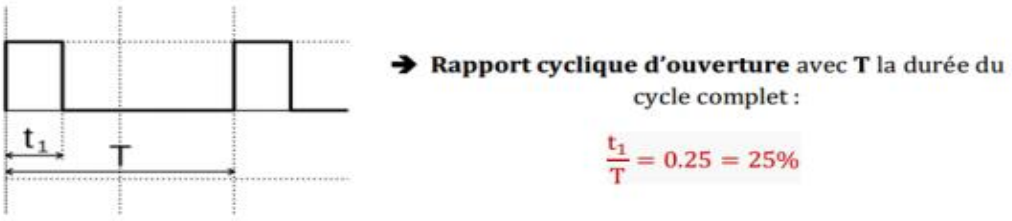



Figure 2.13 : Principe d'un signal de commande par PWM

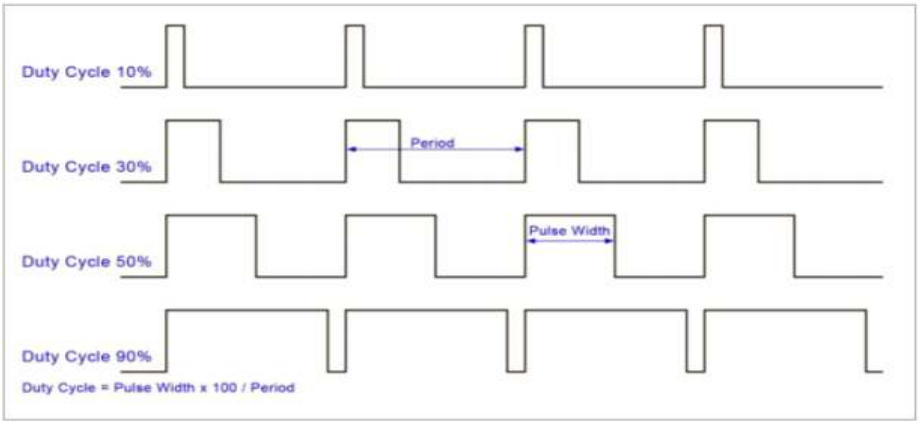



Figure 2.14 : Exemples des signaux PWM

#### 2.2.7.2 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

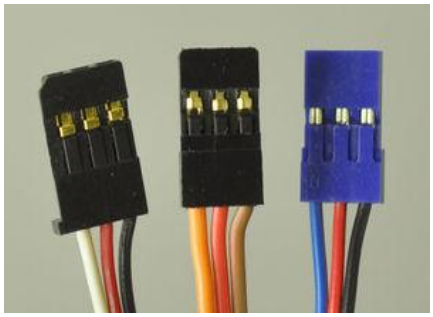


Figure 2.15 : Câble de commande pour servomoteur

## 2.3 Étude de la partie logicielle

Une carte d'acquisition que sa construction se base sur un microcontrôleur doit être dotée d'une interface de programmation comme est le cas de notre carte. L'environnement de programmation open-source pour Arduino peut être téléchargé gratuitement (pour Mac OS X, Windows, et Linux).

### 2.3.1 Plateforme de programmation Arduino

#### 2.3.1.1 Présentation

L'interface de l'IDE Arduino est plutôt simple voir figure 2.16, il offre une interface minimale et épurée pour développer un programme sur les cartes Arduino. Il est doté d'un éditeur de code avec coloration syntaxique et d'une barre d'outils rapide. Ce sont les deux éléments les plus importants de l'interface, c'est ceux que l'on utilise le plus souvent. On retrouve aussi une barre de menus, plus classique qui est utilisée pour accéder aux fonctions avancées de l'IDE. Enfin, une console pour afficher les résultats de la compilation du code source, des opérations sur la carte, etc.

Le langage Arduino est inspiré de plusieurs langages. On retrouve notamment des similarités avec le **C**, le **C++**, le **Java** et le **Processing**. Le langage impose une structure particulière typique de l'informatique embarquée [26].

- La fonction « **Setup** » contiendra toutes les opérations nécessaires à la configuration de la carte (directions des entrées sorties, débits de communications série, etc.).

- La fonction « **Loop** », elle est exécutée en boucle après l'exécution de la fonction setup. Elle continuera de boucler tant que la carte n'est pas mise hors tension, redémarrée (par le bouton reset). Cette boucle est absolument nécessaire sur les microcontrôleurs étant donné qu'ils n'ont pas de système d'exploitation. En effet, si l'on omettrait cette boucle, à la fin du code produit, il sera impossible de reprendre la main sur la carte Arduino qui exécuterait alors du code aléatoire.

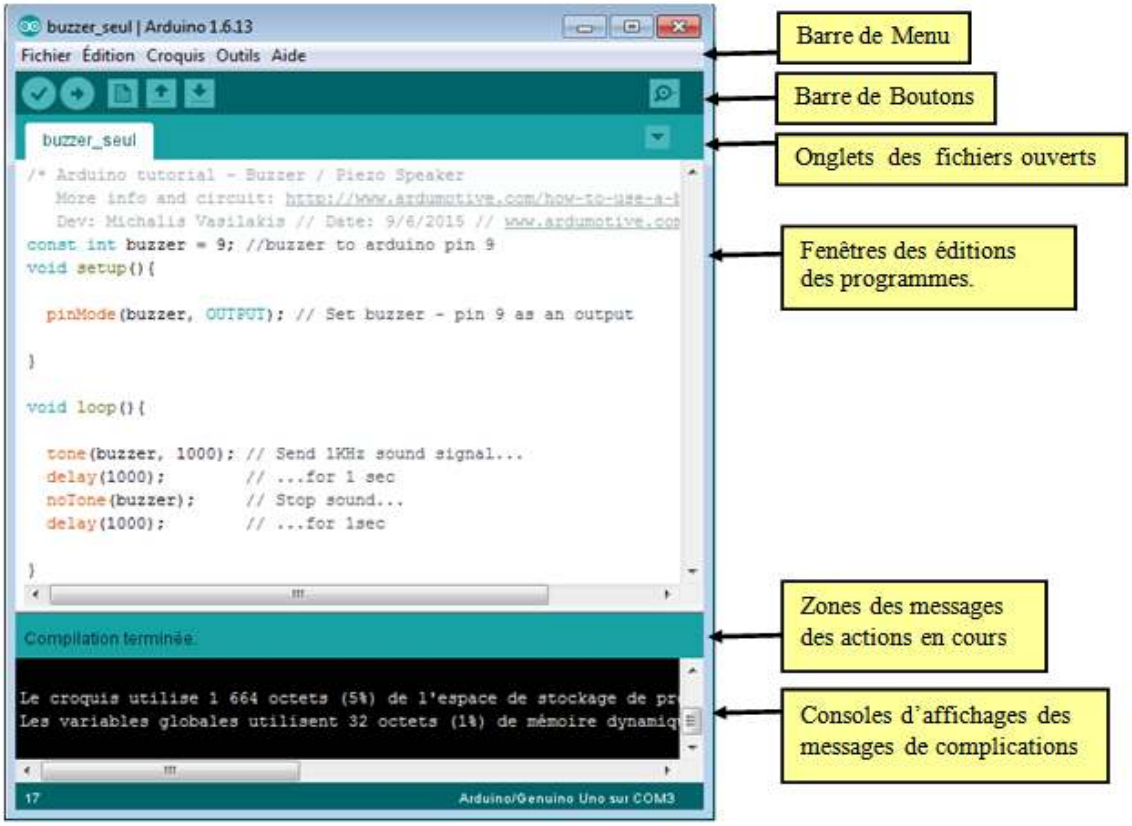



Figure 2.16 : Interface de la plateforme Arduino

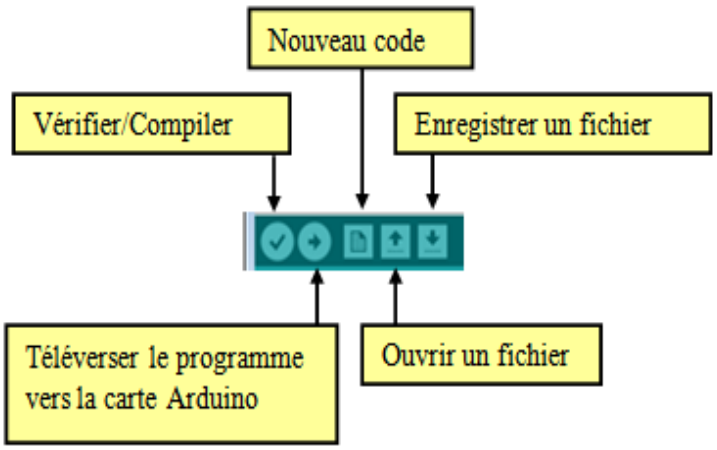



Figure 2.17 : Barre de boutons Arduino

Le logiciel comprends 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 2.18 : HyperTerminal de l'Arduino (Moniteur Série)

### 2.3.1.2 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 :

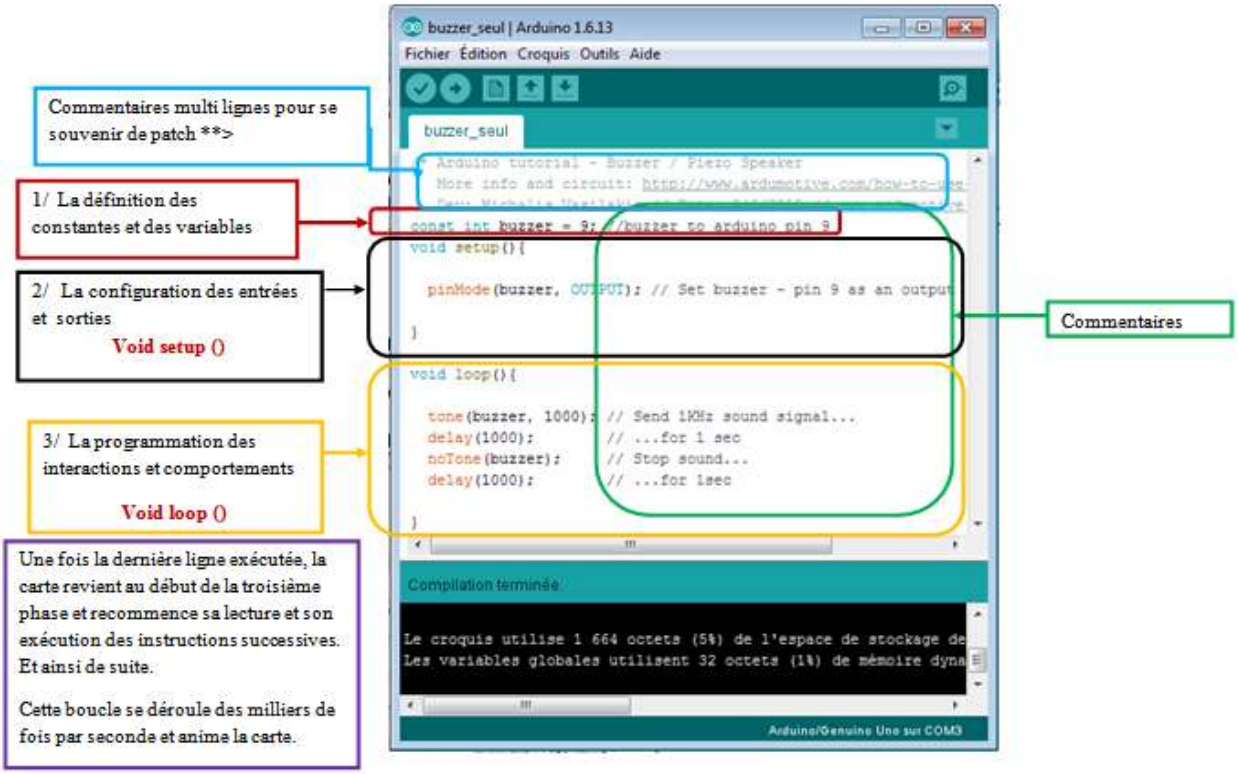



Figure 2.19 : Structure générale du programme (IDE Arduino).

### 2.3.2 Logiciel PROTEUS ISIS

Il existe plusieurs logiciels de simulation des circuits électroniques tels que Workbench, PROTEUS, Tina…etc.

Les premiers tests de simulation du circuit Radar sont faits sur Proteus ISIS, le logiciel fameux des simulations des montages électroniques. En plus de sa 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 Logiciel IDE de PROCESSING

L'IDE de Processing est une excellente source de création de graphiques. Le langage de Processing est un langage de programmation de texte spécifiquement conçu pour générer et modifier des images. Le Processing s'efforce de parvenir à un équilibre entre la clarté et les fonctionnalités avancées. De nombreuses techniques d'infographie et d'interaction peuvent être réalisées, y compris le dessin vectoriel, le traitement d'image, les modèles de couleurs, la

communication réseau et la programmation orientée objet. Les bibliothèques étendent facilement la capacité d'envoyer / recevoir des données dans divers formats et importer et exporter des formats de fichiers 2D et 3D.

Le Processing 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 [28].

### 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 résultants de traitement seront transférées vers le logiciel Processing sous la même liaison, voir figure 3.4.

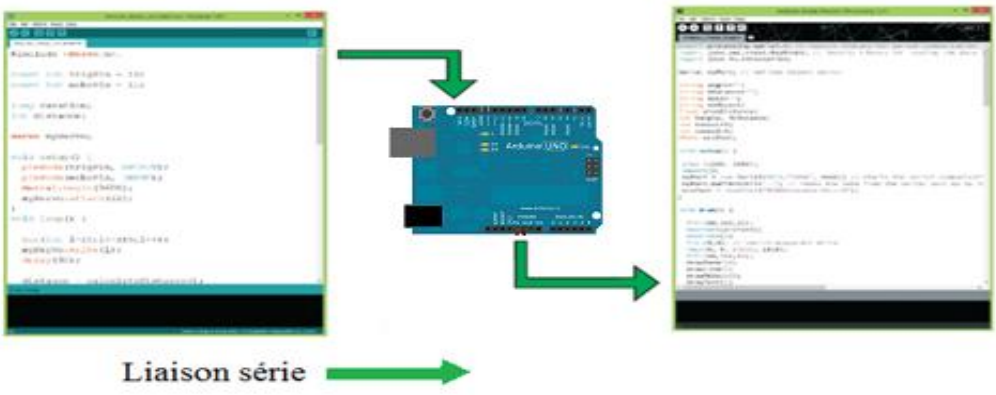



Figure 2.20 **:** Communication entre la carte Arduino et Processing

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 coté de processing et l'autre pour la carte Arduino d'une part et d'autre part il faut rajouter un adaptateur pin-port a coté 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 suivante :

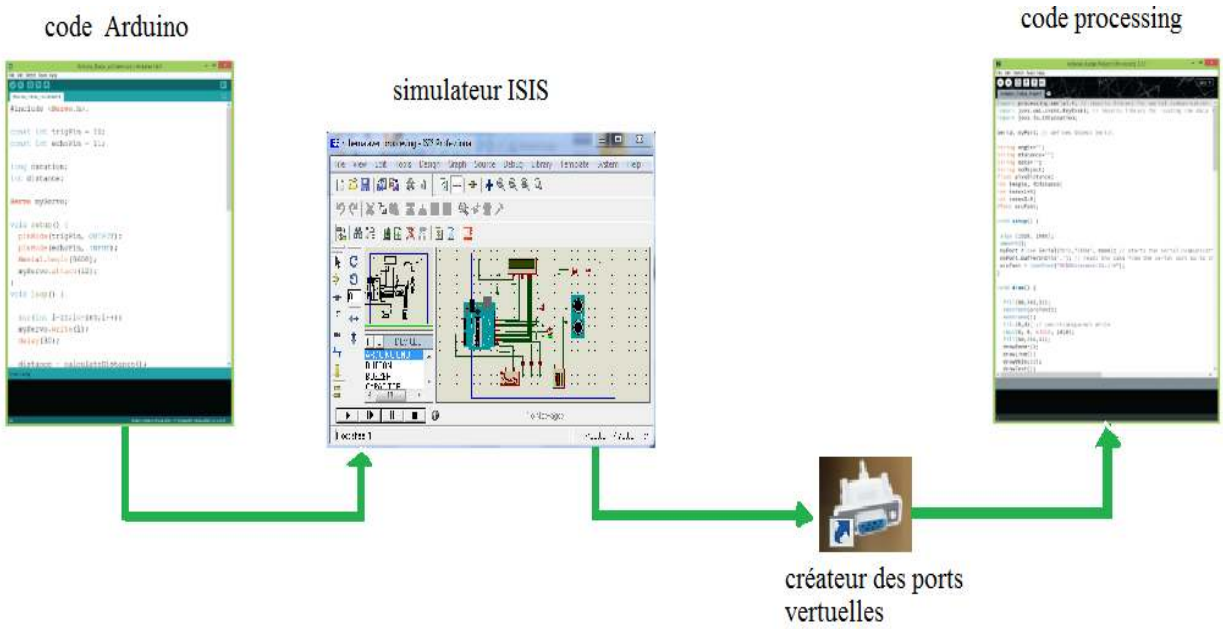



Figure 2.21: Communication entre PROTEUS et PROCESSING

## 2.4 Conclusion

Dans ce chapitre, nous avons vu une carte d'acquisition qui est l'Arduino donnant ainsi les raisons pour lesquelles on l'a choisie, puis nous avons cité des différents types de cette dernière, ainsi quelques descriptions théoriques sur le servomoteur, le capteur ultrason, l'afficheur LCD. Ensuite, nous avons expliqué les deux parties essentielles de l'Arduino; plus précisément (la partie matérielle et la partie de programmation). Nous avons également expliqué le principe de fonctionnement de la carte Arduino sans oublier ses caractéristiques. Le chapitre suivant sera consacré à la réalisation d'un dispositif Radar de détection à capteur ultrason.

# CHAPITRE 3
# RÉALISATION DU PROJET

## 3.1 Introduction

Dans ce chapitre, on présentera 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, on va procéder à l'application expérimentale, pour cette raison, plusieurs blocs ont été nécessaires afin de réaliser une telle combinaison avec le module Arduino et son environnement de développement IDE.

## 3.2 Principe de fonctionnement

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 Déroulement du projet

Notre projet de réalisation a été fait en deux parties:

- La première partie est la conception assistée par ordinateur CAO, la simulation avec PROTEUS.
- La deuxième partie est la réalisation, le montage pratique.

Pour les deux parties simulation et 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 général

Le schéma synoptique général de notre projet est indiqué par la figure 3.1.

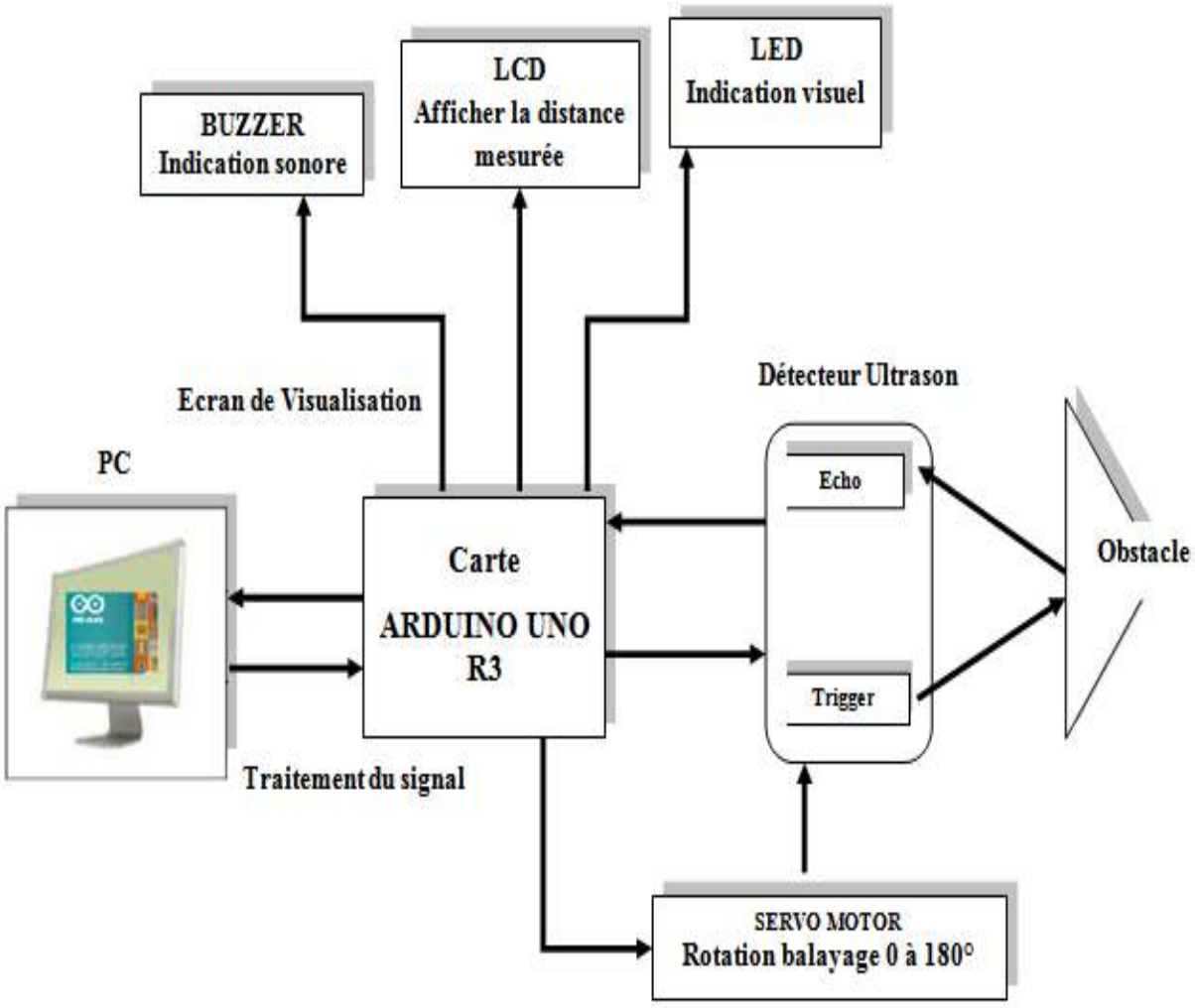
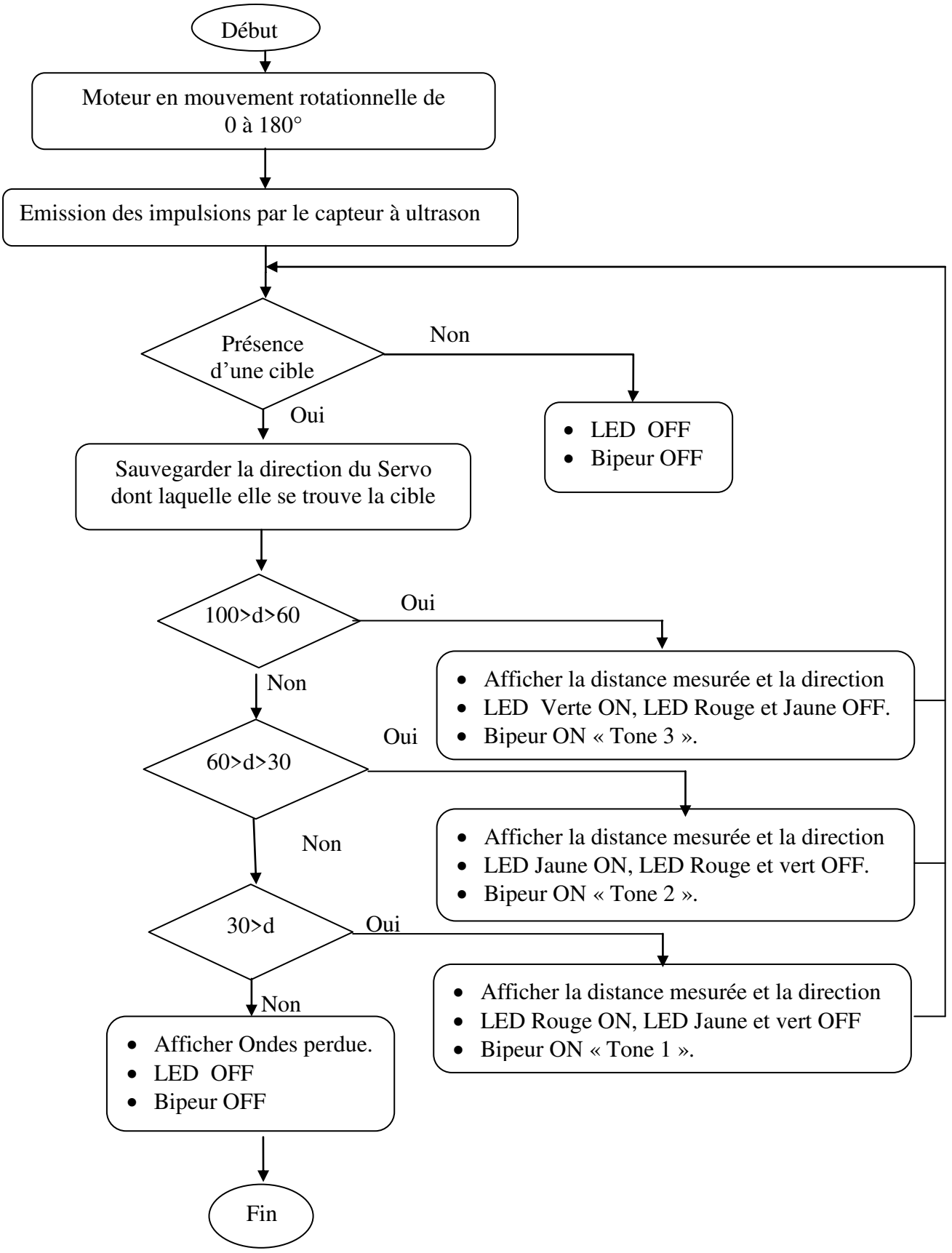



Figure 3.1 : Schéma synoptique de notre projet Radar.

## 3.5 Algorithme principale de détection d'un objet



## 3.6 Simulation du projet sous PROTEUS

### 3.6.1 Présentation

Avant de passé à la réalisation, nous avons procédé à une simulation, d'où nous avons travaillé avec plusieurs logiciels, principalement logiciel Arduino IDE pour fonctionner le circuit, logiciel Proteus pour simuler les circuits électronique et le logiciel IDE de Processing pour afficher les résultats.

### 3.6.2 Démarche de la simulation

#### 3.6.2.1 Bibliothèque Arduino pour Proteus

Pour simuler la carte Arduino sur Proteus, tout d'abord, il faut télécharger la librairie Arduino disponible sur internet, décompresser et copier les deux fichiers nommés « ArduinoTEP.LIB et ArduinoTEP.IDX » et les placer dans le dossier des bibliothèques de notre logiciel Proteus.

Maintenant, on redémarre le logiciel Proteus et dans la recherche de sections de composants pour ArduinoTEP on choisit Arduino Uno comme indiqué ci-dessous:

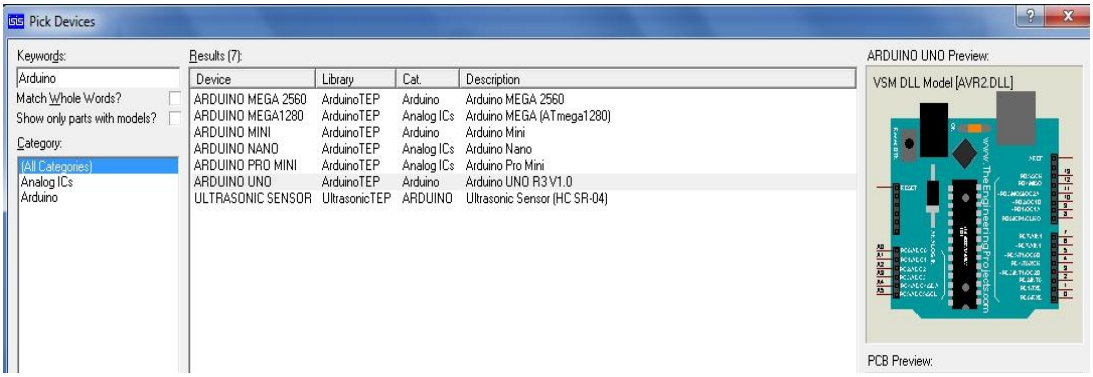



Figure 3.2 : Intégrer carte Arduino sous Proteus

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

#### 3.6.2.2 Bibliothèque de capteurs à ultrasons pour PROTEUS

Pour interagir le capteur à ultrasons avec Arduino on télécharge une bibliothèque de capteurs ultrasoniques pour Proteus, disponible sur internet et qui présente trois fichiers nommés « UltrasonicTEP.IDX, UltrasonicTEP.LIB, UltrasonicTEP.HEX » Maintenant, on les place dans la bibliothèque de logiciel Proteus.

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

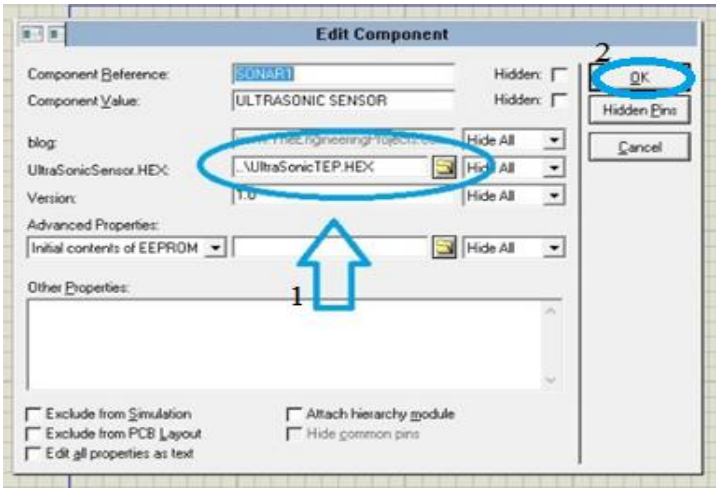



Figure 3.3 : Fichier UltrasonicTEP.HEX sous Proteus.

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

### 3.6.2.3 Circuit global de la simulation

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

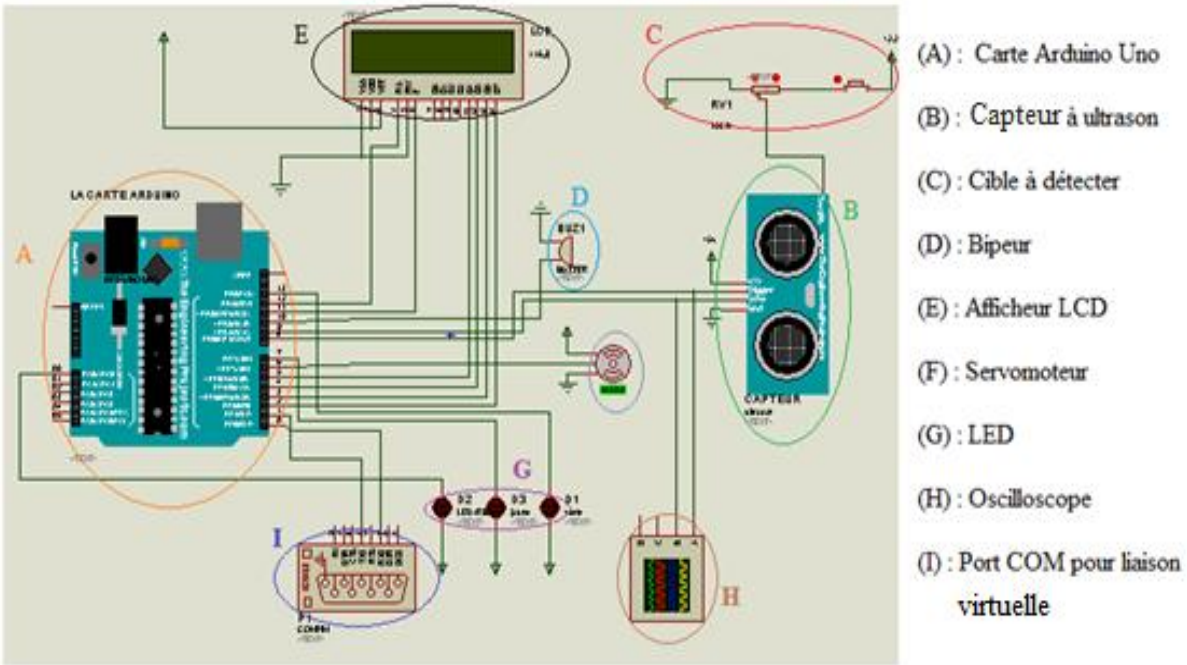



Figure 3.4 : Circuit électronique globale du projet

Les différentes connexions et numéro des pins de circuit sont présentés par le tableau suivant :

| Composant | Pin composant | Pin Arduino Uno |
|---|---|---|
| Capteur à ultrason | VCC | +5V |
| | Trigger | 8 |
| | 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 |
| | VCC | +5V |
| Servomoteur | Marron | GND |
| | Rouge | + 5v |
| | Orange | 6 |
| LED-Verte | PIN + | 13 |
| | PIN - | GND |
| LED-Jaune | PIN + | 7 |
| | PIN - | GND |
| LED-Rouge | PIN + | 14 |
| | PIN - | GND |
| Bipeur | Pin+ | 10 |
| | Pin- | GND |
| Port Com liaison virtuelle | 3 | TX |
| | 2 | RX |

Tableau 3.1 : Broche et connexion du notre circuit électronique.

Pour commencer la simulation, on compile le code de programme Arduino afin de récupérer le fichier **.Hex** et on l'injecte dans la carte Arduino, puisque c'est une simulation, donc on a représenté la cible par une source de tension, un bouton et un potentiomètre connectés avec le capteur au pin de test, d'où le bouton poussoir sert à indiquer la présence ou l'absence de la cible. On clique sur démarrer pour lancer la simulation.

Un oscilloscope a été connecté au capteur pour visualiser les deux signaux Trigger et Echo

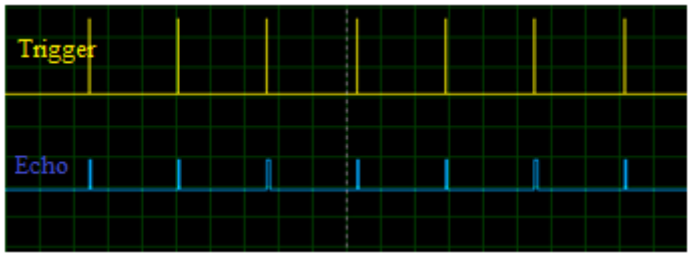



Figure 3.5 : Signal écho et signal trigger de l'ultrason.

Le port COM est utilisé pour procéder à une liaison virtuelle entre la carte Arduino et le logiciel IDE de Processing.

### 3.6.3 Résultats de la simulation

Pour obtenir des différentes détections, on a changé la valeur de potentiomètre afin d'avoir une variation de puissance au pin de test de capteur à ultrason, la distance calculée est proportionnelle à cette puissance.

La distance peut varie de (04cm) à (05m), les résultats de simulation sont présentés dans un fichier « html », Le tableau suivant extrait du fichier « Tableau.html » résume les différentes détections obtenues :

| Numéro de détection | Distance (cm) | Direction (°) |
|---|---|---|
| 1 | 4 | 1 |
| 2 | 9 | 5 |
| 3 | 15 | 8 |
| 4 | 24 | 14 |
| 5 | 37 | 16 |
| 6 | 39 | 35 |
| 7 | 48 | 39 |
| 8 | 57 | 40 |
| 9 | 85 | 43 |
| 10 | 94 | 45 |

Tableau 3.2 : Résultats de plusieurs détections de simulation

Les résultats obtenus (Distance et direction) vont être transférés vers Processing pour les afficher où l'objet est représenté par un point rouge situant dans un point d'intersection entre la direction et la distance.

On remarque bien dans la figure de Processinge : Objet détecté, distance = 90cm et angle =56° d'où la détection d'une cible est identifiée par deux paramètres importants sont, la distance et la direction.

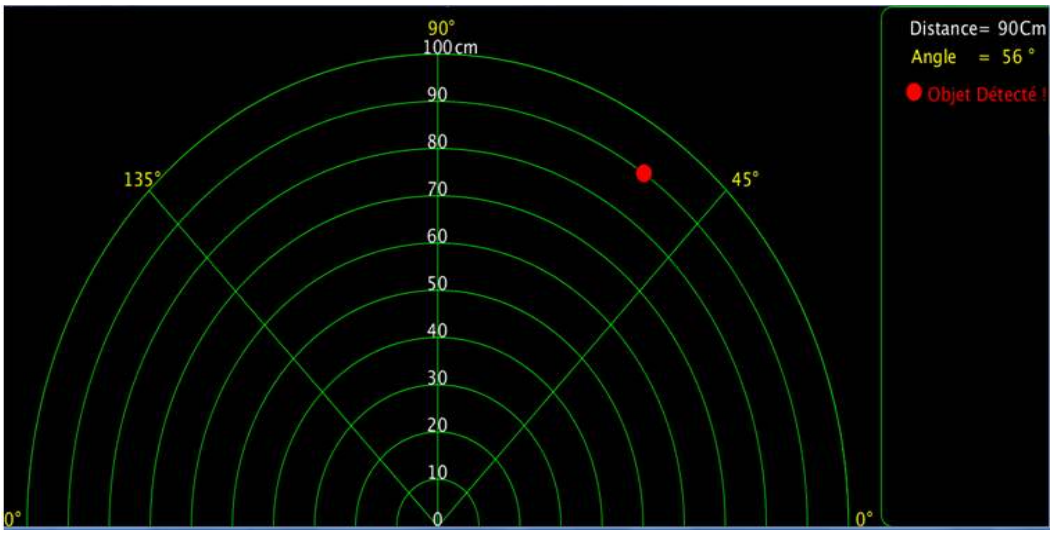



Figure 3.6 : Résultat de détection lors de la simulation.

## 3.7 Réalisation de projet

### 3.7.1 Composants utilisés

Pour notre réalisation, nous avons assemblé les différents composants suivants :

- (01) Carte Arduino Uno R3;
- (01) LCD 2*16 ;
- (03) LEDs rouge, verte, et jaune ;
- (01) SERVO Moteur ;
- (04) Résistances (330 ohms) ;
- (01) Potentiomètre variable;
- (01) Capteur sonar à Ultrasons HC-SR04 ;
- (01) Bipeur .

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

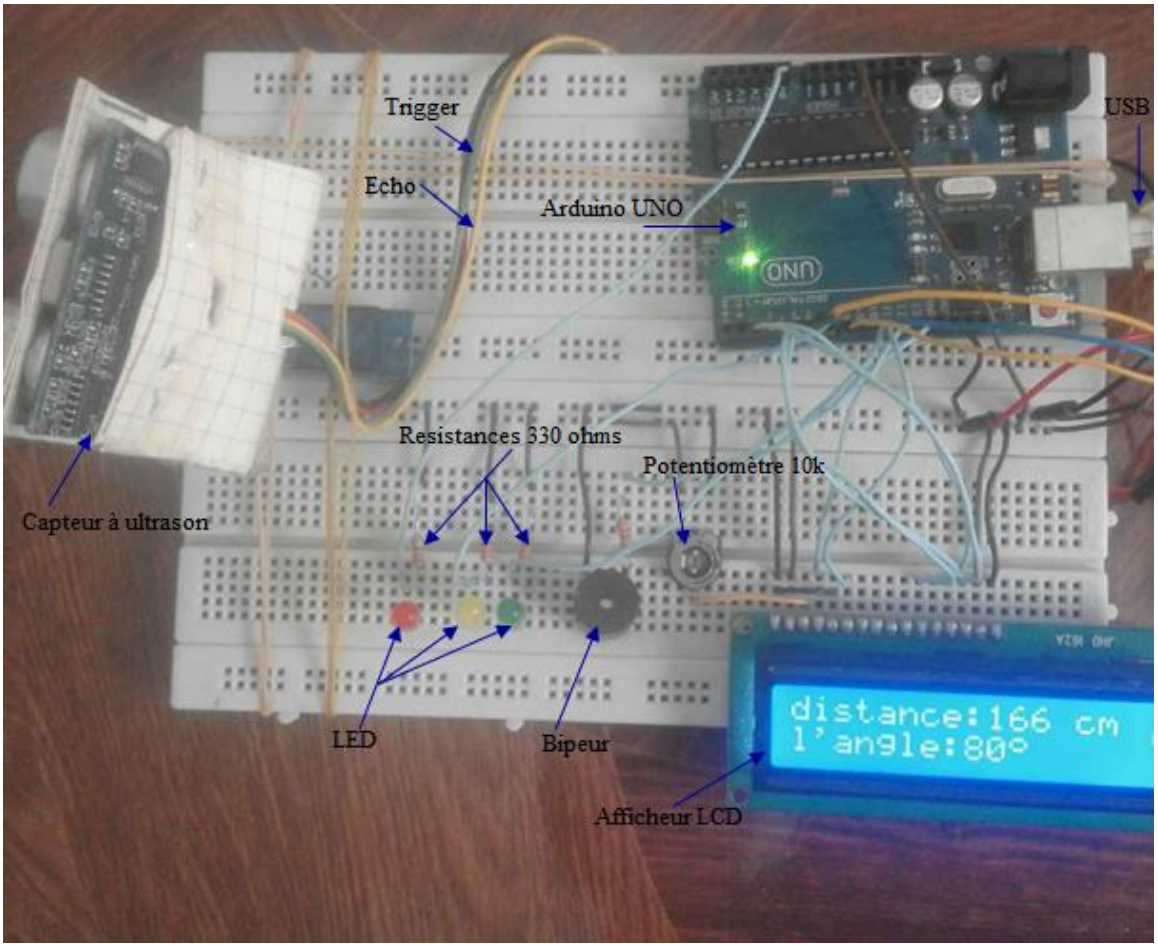



Figure 3.7 : Circuit final de notre projet de réalisation Radar de détection

### 3.7.2 Description et étape de la réalisation

#### 3.7.2.1 Alimentation du circuit

L'ensemble des dispositifs Arduino, capteur ultrason, afficheur LCD, bipeur exigent une alimentation stabilisée de (+5V), pour notre travail de réalisation, nous avons alimenté notre montage à travers le port USB de l'ordinateur.

#### 3.7.2.2 Test du fonctionnement de la carte Arduino

Initialement, on teste le fonctionnement de la carte Arduino, en connectant cette dernière avec le port USB de PC. Si la LED power s'allume la carte est bonne.

En suite, on clique sur le bouton Reset de la carte, pour supprimer tout ancien programme et de la réinitialiser,

Maintenant, on connecte les broches des composants suivant le tableau 3.1, on compile le programme sous Arduino IDE, et on téléverse le programme vers la carte pour l'exécution via le câble USB.

### 3.7.3 Principe de fonctionnent du circuit

La carte Arduino 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 (10µs) 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.

**Remarque :** Il y a toujours un silence de durée fixe après l'émission des ultrasons pour éviter de recevoir prématurément un écho en provenance directement du capteur.

L'afficheur LCD associe à notre dispositif afin d'afficher la distance calculée ainsi l'angle de Servomoteur.

Le bipeur est un composant supplémentaire, il déclenche lorsqu'il y a une détection, Et les Trois LEDs déterminent la zone dont laquelle l'objet est situé (Zone proche, moyenne, ou lointaine).

Schéma de Circuit global des connexions des composants pour notre projet de réalisation du Radar de détection est présenté par la figure suivante.

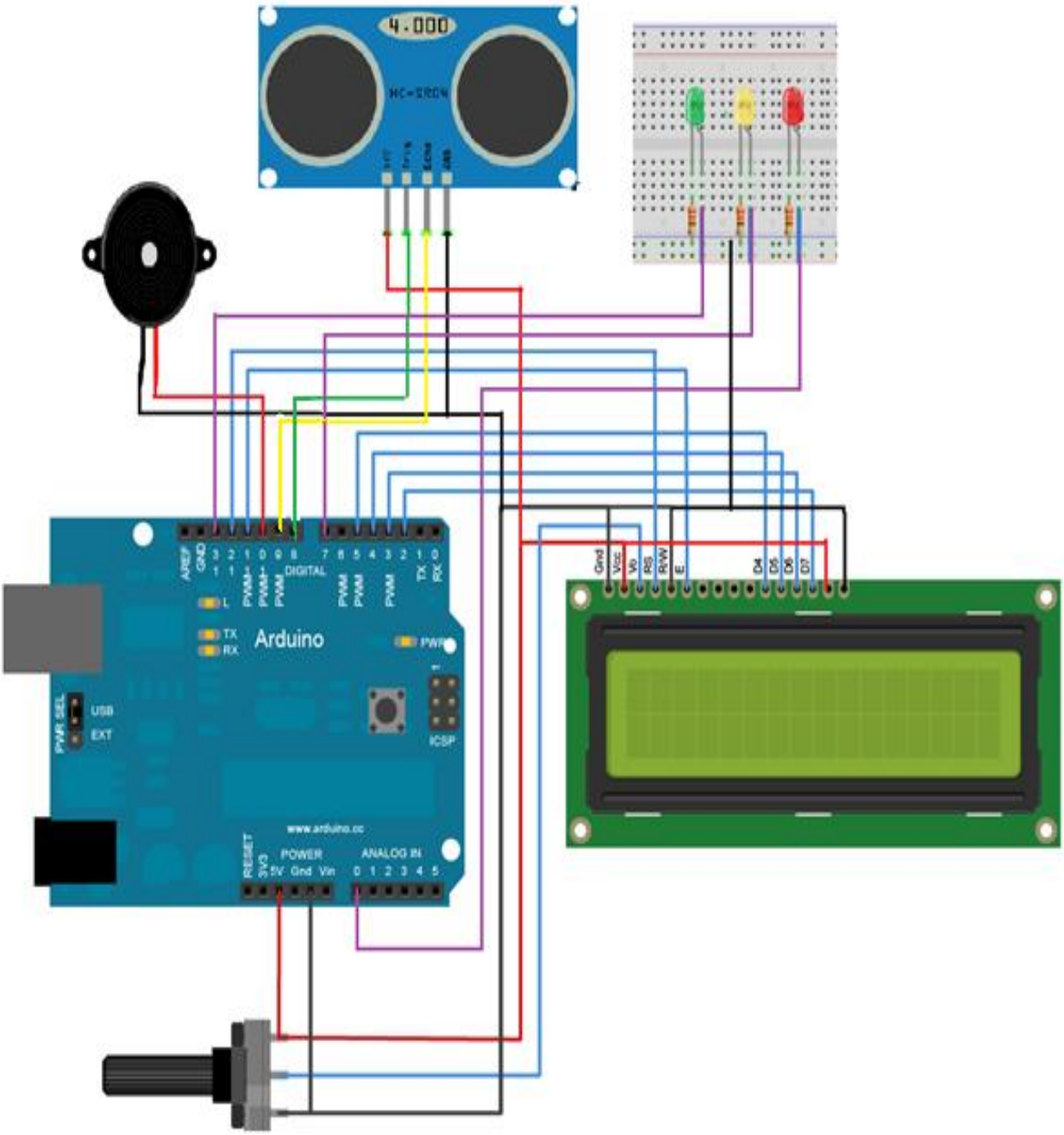



Figure 3.8 : Schéma de Circuit global des connexions pour notre réalisation du projet Radar

### 3.7.4 Résultats de la réalisation

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

### 3.7.4.1 Détection de la cible fixe

Les résultats des mesures pour une cible fixe sont présentés dans le tableau ci-dessous.

| Objet_N | Numéro de détection | Distance (cm) | Direction (°) |
|---|---|---|---|
| Objet_1 | 1 | 8 | 29 |
| | 2 | 11 | 31 |
| | 3 | 12 | 32 |
| Objet_2 | 4 | 19 | 75 |
| | 5 | 17 | 76 |
| Objet_3 | 6 | 15 | 89 |
| | 7 | 15 | 90 |
| | 8 | 15 | 91 |
| Objet_4 | 9 | 13 | 120 |
| | 10 | 13 | 121 |
| | 11 | 12 | 123 |
| | 12 | 13 | 124 |
| | 13 | 12 | 125 |
| Objet_5 | 14 | 6 | 107 |
| Objet_6 | 15 | 24 | 81 |
| | 16 | 24 | 80 |
| Objet_7 | 17 | 34 | 31 |
| | 18 | 33 | 29 |
| | 19 | 33 | 28 |
| Objet_8 | 20 | 24 | 20 |

Tableau 3.3 : Distance et la direction obtenue pour détection de la cible fixe.

Le tableau au dessus est un extrait du fichier « Tableau.html », il contient les informations (Distance et direction) de plusieurs objets détectés fixes.

On remarque dans le tableau que les distances et les directions sont proches l'une des l'autres qui indique la présence d'un seul objet.

Les angles séquentiels avec une distance invariante indiquent indique que l'objet est fixe avec une largeur supérieure à arcsin (15°), (L'ouverture de champs de détection de capteur à ultrason).

Le déclanchement de bipeur et le clignotement des LEDs (Rouge, jaune et vert) lors de la direction avec différentes tonalités et couleurs indiquent le changement de la distance des cibles.

Lorsqu'il y a une présence d'une cible fixe, il y'a une puissance importante réfléchie vers le récepteur et une variation de fréquence.

Le signal écho, initialement pour une absence d'objet, présente un signal de faible fréquence, lorsqu'on place une cible devant le capteur, On remarque le changement de fréquences par la suite, d'où la détection de l'objet fixe.

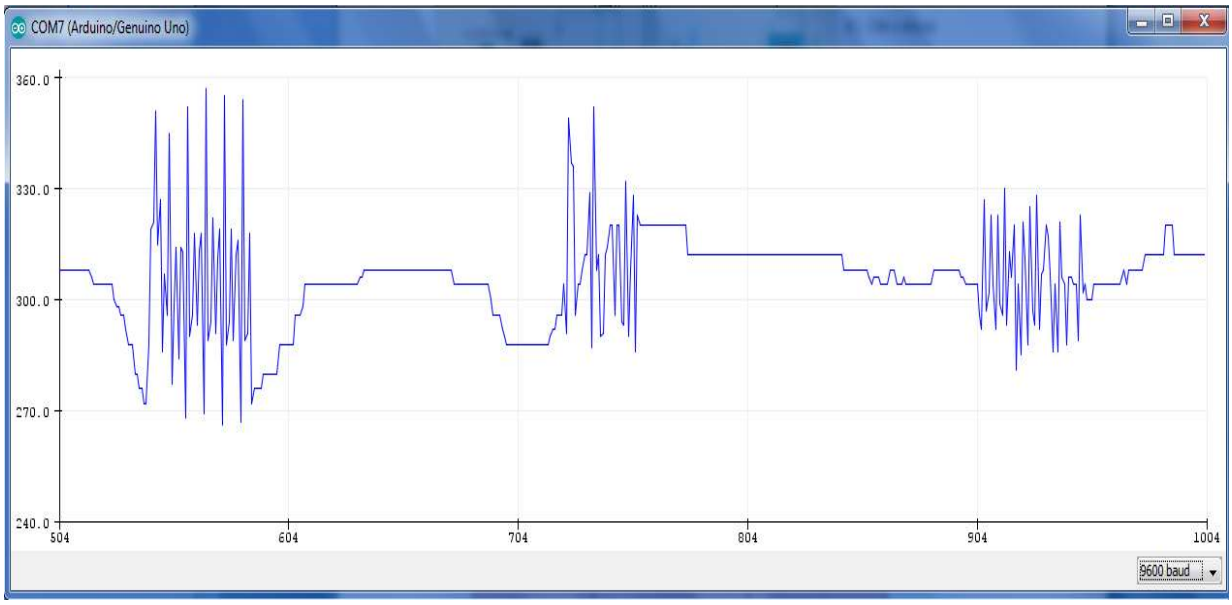



Figure 3.9 **:** Signal écho de détection d'objet fixe à l'entrée de carte Arduino

### 3.7.4.2 Détection de la cible mobile

Pour la détection de la cible mobile, on a fait tourner l'objet suivant la direction du mouvement de servomoteur,

Les résultats des mesures pour une cible mobile sont présentés dans le tableau ci-dessous.

| Numéro de détection | Distance (cm) | Angle (°) |
|---|---|---|
| 1 | 14 | 13 |
| 2 | 14 | 14 |
| 3 | 13 | 15 |
| 4 | 16 | 16 |
| 5 | 15 | 17 |
| 6 | 15 | 17 |
| 7 | 15 | 18 |
| 8 | 15 | 19 |
| 9 | 14 | 20 |
| 10 | 15 | 21 |
| 11 | 14 | 22 |
| 12 | 15 | 23 |
| 13 | 15 | 23 |
| 14 | 15 | 24 |
| 15 | 17 | 26 |
| 16 | 16 | 27 |
| 17 | 15 | 28 |
| 18 | 15 | 29 |
| 19 | 15 | 29 |
| 20 | 15 | 30 |
| 21 | 15 | 31 |
| 22 | 15 | 32 |
| 23 | 16 | 33 |

Tableau 3.4 : Distance et la direction obtenue pour détection cible mobile.

On remarque que les valeurs de la distance sont proches, presque constantes pendant le balayage des directions séquentielles du servomoteur, on constate que la cible est en mouvement synchronisée avec les rotations du Servomoteur.

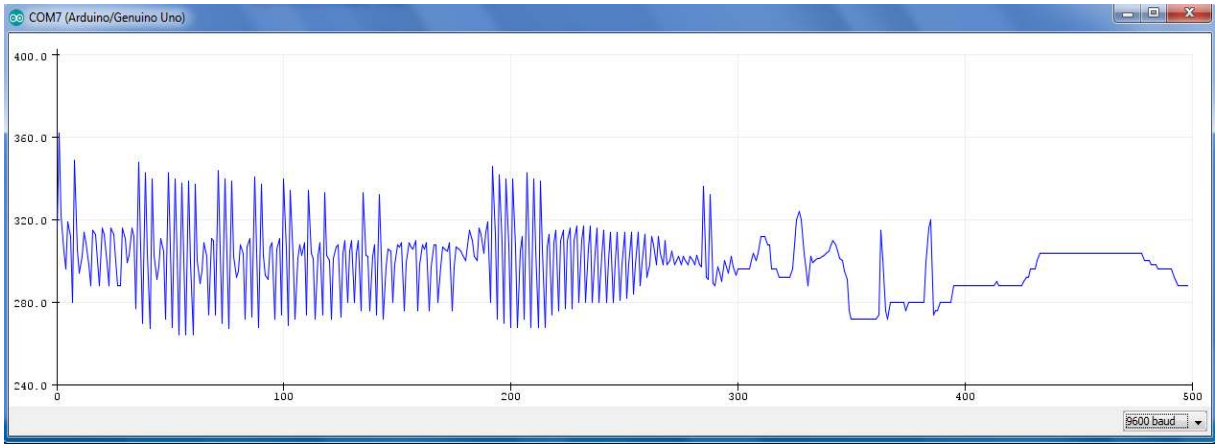



Figure 3.10 **:** Signal écho détection d'objet mobile à l'entrée de carte Arduino

La figure au dessus représente le signal écho en temps réel dans le cas d'objet en rotation proportionnel avec le capteur à ultrason.

On remarque que le signal écho, présente deux partie différentes, une pour la détection objet mobile et l'autre pour l'absence de la détection.

### 3.7.5 Interface de Processing pour détection d'objets fixe et mobiles

Les figures suivantes représentent l'interface Processing lors d'absence et de présence de cible :

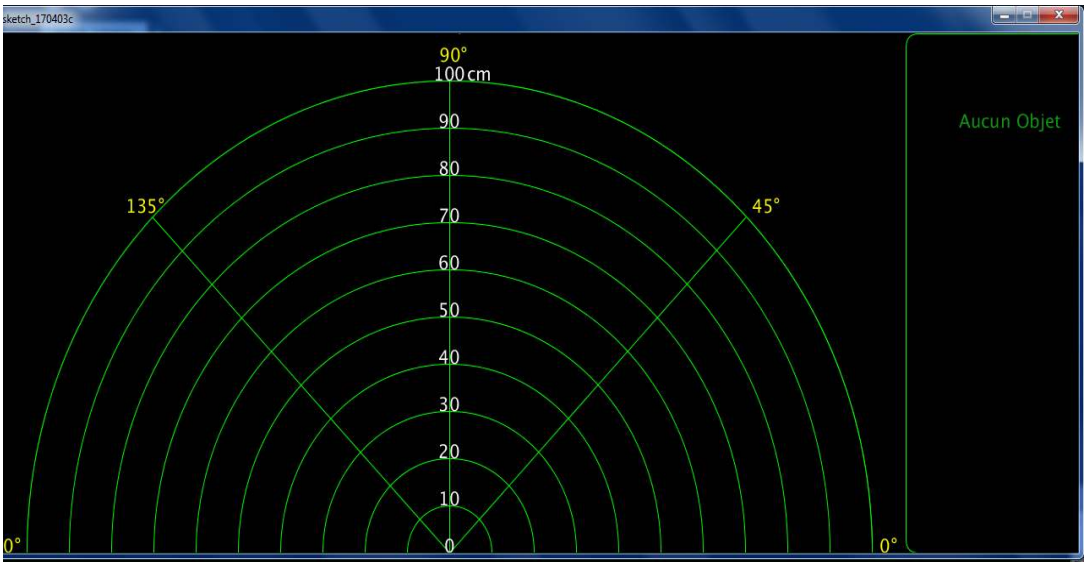



Figure 3.11 : Interface du Processing avec non détection (Absence d'objet)

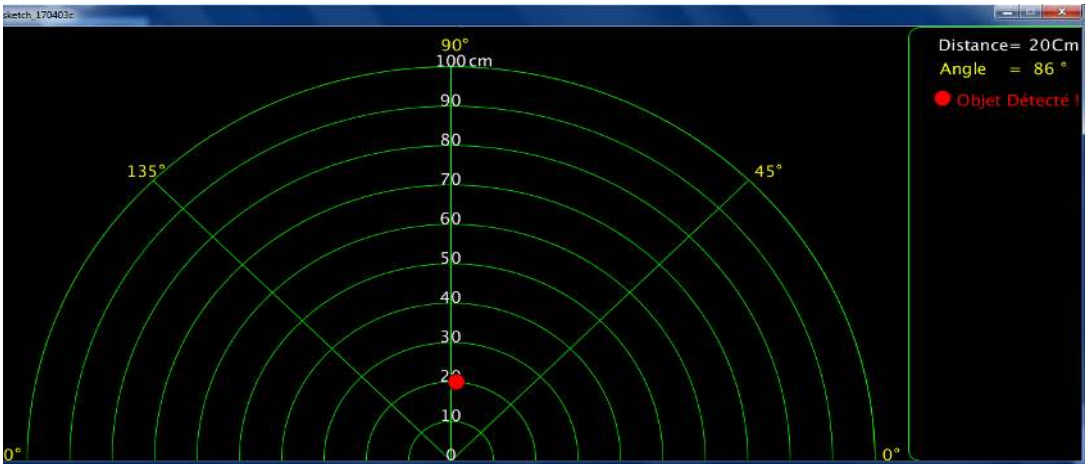



Figure 3.12 : Interface du Processing avec détection (Présence d'objet)

Le tableau.html est un fichier qui enregistre les détections obtenues dans un tableau afin d'avoir une poursuite des objets mobiles et fixes. Il est présenté dans l'ANNEXE B, et C.

## 3.8 Précision du capteur

Pour terminer notre travail, on va essayer de mesurer la précision du capteur à ultrason. Tout d'abord nous allons bloquer le Servomoteur pour fixer le capteur dans la direction de l'objet.

Pour un exemple d'obstacle de petite taille (Un stylo) sur une table, l'obstacle est bien trop petit pour donner des échos "Utile" et de multiples obstacles se trouvent sur les côtés de la zone de mesure.

Ce capteur à ultrason a besoin d'une zone vide, avec une surface non absorbante (du son), dure et lisse et large pour donner des résultats corrects.

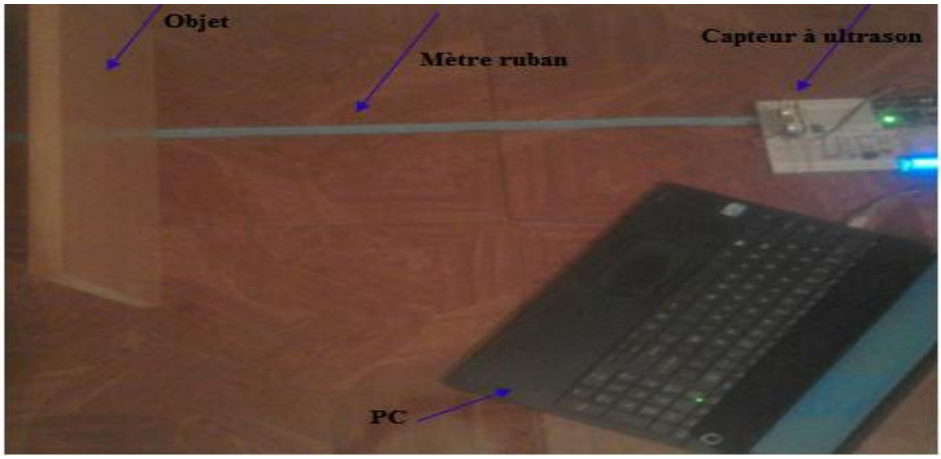



Figure 3.13 **:** Test de réalisation avec un objet d'une surface large

Pour commencer nos tests de précision, nous allons fixer les distances à mesurer dans une plage de (2cm à 1m), puis nous allons refaire les mesures (10) dix fois pour chaque distance, et à l'aide de logiciel Excel, nous allons calculer la moyenne et l'erreur de mesure.

Les résultats de test de précision du capteur sont présentés dans le tableau suivant :

| Distance (mm) | Moyenne des distances mesurées (mm) | Erreur de mesure (mm) |
|---|---|---|
| 20 | 25,9 | 5.9 |
| 30 | 34,8 | 4.8 |
| 40 | 39,4 | 0.6 |
| 50 | 48,7 | 1.3 |
| 60 | 57,2 | 2.8 |
| 70 | 70,3 | 0.3 |
| 80 | 78,4 | 1.6 |
| 90 | 88,7 | 1.3 |
| 100 | 102,1 | 2.1 |
| 150 | 150,4 | 0.4 |
| 200 | 204,1 | 4.1 |
| 250 | 253,2 | 3.2 |
| 300 | 299 | 1 |
| 350 | 349,8 | 0.2 |
| 400 | 398 | 2 |
| 450 | 444 | 6 |
| 500 | 491,63 | 8.37 |
| 550 | 536,5 | 13.5 |
| 700 | 687,2 | 12.8 |
| 800 | 787,7 | 12.3 |
| 900 | 883,1 | 16.9 |
| 1000 | 979,6 | 20.4 |

Tableau 3.5 : Mesures de test de précision.

**Remarque**

Les distances mesurées en pratique sont différente de celles de la théorie, L'erreur de mesure dans notre test est de (0.2 mm à 20,4 mm), avec une moyenne de (05 mm) qui est supérieur à l'erreur données par le constructeur qui égal à (03 mm).

## 3.9 Conclusion

Dans ce chapitre on a vu la réalisation d'un Radar de détection à l'aide de la carte électronique Arduino Uno, afin de mesurer la distance et la direction des objets à traves un capteur à ultrason et dans les directions de 0 à 180° à l'aide du servomoteur.

Lors de la simulation, la distance calculée est proportionnelle à la puissance appliquée à l'entrée de pin de test du capteur veut dire, lorsqu'on augmente la valeur de la résistance du potentiomètre, la distance augmente.

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

Finalement, et pour terminer notre travail, on a discuté sur la précision du capteur d'où le capteur à ultrason est fixé dans la direction de l'objet. La distance mesurée pratiquement est légèrement différente au celle fixée.

# CONCLUSION GÉNÉRALE

# Conclusion générale

Notre projet comporte un travail théorique accompagné d'une réalisation, son objectif consiste sur étude et la réalisation d'un radar de détection. Pour faire preuve notre réalisation nous avons utilisé une carte Arduino, le capteur de distance à ultrason, un servomoteur et un afficheur LCD 2*16.

Notre radar fonctionne en rotation, il détecte et calcule les distances suivant un programme avancé. Pour réaliser ce travail, on a passé par différentes étapes : On a utilisé un detecteur à ultrason HC-SR04 pour la détection des obstacles et le calcul de la distance entre le radar et l'objet.

L'ensemble de système de détection et de déplacement est commandé par la carte Arduino programmable qui doit en utilisant les informations actuelles, décider l'action à prendre. Pour notre cas; on a utilisé la carte Arduino Uno R3 dont ses caractéristiques particulières nous à faciliter les taches surtout en ce qui concerne sa programmation.

Notre projet de réalisation a été fait en deux parties: La première partie est la conception assistée par ordinateur CAO, et la deuxième partie est la réalisation, pour les deux parties simulation et réalisation, on a passé par la programmation orientée objet ainsi la visualisation des résultats trouvés.

Dans la partie simulation, pour obtenir des différentes détections, on a changé la valeur de potentiomètre connecté au pin de test de capteur à ultrason, afin d'avoir une variation de puissance, résultat de la simulation : La distance calculée est proportionnelle à cette puissance. Pour la réalisation, on a fait des mesures de distance avec un détecteur en mouvement pour un balayage de (0 à 180°), d'où on a affiché les résultats sur le Processing.

Pour terminer notre travail, on a mesuré la précision du capteur, l'erreur des distances mesurées est supérieure à l'erreur théorique d'où il est préférable d'utiliser une distance minimale égale à (10 cm) pour des résultats plus précise.

# RÉFÉRENCES

# Références


[1] Stevens Jamal Romaric. *Le RADAR*. Linkedin Corporation 2017.

[2] Michel-Henri CARPENTIER, *« RADAR »,* Encyclopaedia Universalis [en ligne], consulté le 11 juin 2017.URL : http://www.iniversalis.fr/encyclopedie/radar/

[3] Merrill I. Skolnik. *INTRODUCTION TO RADAR SYSTEMS*. International Edition . 2$^{éme}$ édition 1981.Gayle J.Cerra, Singapore : Angelson, Frank, 1980-1962. 1. Radar. I.TItle. II. Series. TK6575.S477 1980 621.3848 79-15354, ISBN 0-07-057909-1

[4] GIRARD M.C. & GIRARD C.M. *Traitement des donn1ées de télédétection*. 1999. DUNOD Ed. Paris, pages 482 à 495

[5] Preeti Jain. *Different Types of RADARs* ,Engineers Garage …. Inspiring creations.

[6] Ms.S.MEGHELATI. *COURS RADAR*, Université de Blida, Faculté des sciences de l'ingénieur, Département Aéronautique.

[7] Dominique Nardi. *Effet Doppler: Applications en télédétection (Radar - Sonar – Echographie).*

[8] Annick PLAGELLAT-PENARIER. *Introduction aux RADARS.2015-2016*. Master2, Département EEA, Institut d'électronique, Université de MONTPELLIER.

[9] The University of Sydney. *Sensors and Signals,Sydney*, Australia: Director, Australian Centre for Field Robotics2002-2017,09/08/2016.

[10] CHRISTIAN WOLFF. *Radar tutorial*.Creative Commons Attribution-Share Alike 3.0 Unportes licence, additional terms may applay.(Online since November 1998).

[11] Bassem R.Mahafza,Ph.D.*Radar Systems Analysus and Design Using MATLAB*. United States of America : CHAPMAN& HLL/CRC,2000. 1. Radar. 2. System analysis—Data processin gM. A3T.LAB. I. Title.. ISBN 1-58488-182-8.

[12] Mathieu DUMONT; Loick ATTAGNANT; Killian JACQUET. *Arduino* [en ligne]. Publié le 13/01/2017.

[13] KRAMA Abdelbasset, GOUGUI Abdelmoumen. 08/06/2015. *Etude et realization d'une carte de controle par Arduino via le systéme Androide*. Mémoire de Master Académique en sciences et technologies,Filiére Génie électrique, Spécialité Electrotechnique Industrielle. Université Kasdi Merbah Ourgla.

[14] Simon Landrault (Eskimon), Hippolyte Weisslinger (olyte).*Arduino: Premiers pas en informatique embarquée*[en ligne]. Edition du 19 juin 2014. Le blog d'Eskimon .

[15] S.V.D REYVANTH, G.SHIRISH, D.SIVA VARMA , M. RAMU . 2009-2013. *Pid controller using arduino* . Projet report for the Award of the Degree of Bacjelor of Téchnology in Electrical And Electronics Engineering Gokaraju Rangaraju Institute of Engineering & Technology Bachupally, Hyderabad-72

[16] X. HINAULT - 2010 – 2012 . adapté par David Gilbert, powered by Pm Wiki.

[17] B. Cottenceau . *Carte ARDUINO UNO Microcontrôleur ATMega328*[en ligne]. Microcontrôleurs EI3 Option AGI.

[18] Christian Tavernier. ***Arduino*** *Applications avancées*[en ligne]. Dunod, Paris, 2012. ISBN 978-2-10-058205-1

[19] Jean-Luc. *Les écrans LCD alphanumériques* [en ligne].Le 3 mars 2015.

[20] Xavier HINAULT (2010) Référence Arduino français.

[21] Astalaseven , Eskimon et olyte. *Arduino pour bien commencer en électronique et en programmation* [en ligne]. Licence Creative Commons BY-NC-SA 2.0.

[22] Lucien Bachelard. *HC-SR04 - Module de détection aux ultrasons - Utilisation avec Picaxe*[en ligne]. 28 novembre 2015.

[23] Lucien Bachelard, *HC-SR04 - Module de détection aux ultrasons - Utilisation avec Picaxe* Le 28 novembre 2015.

[24] A poros de MCHobby-Wiki. Servo-Moteur [en ligne].

[25] Arduino UNO Reference Design.

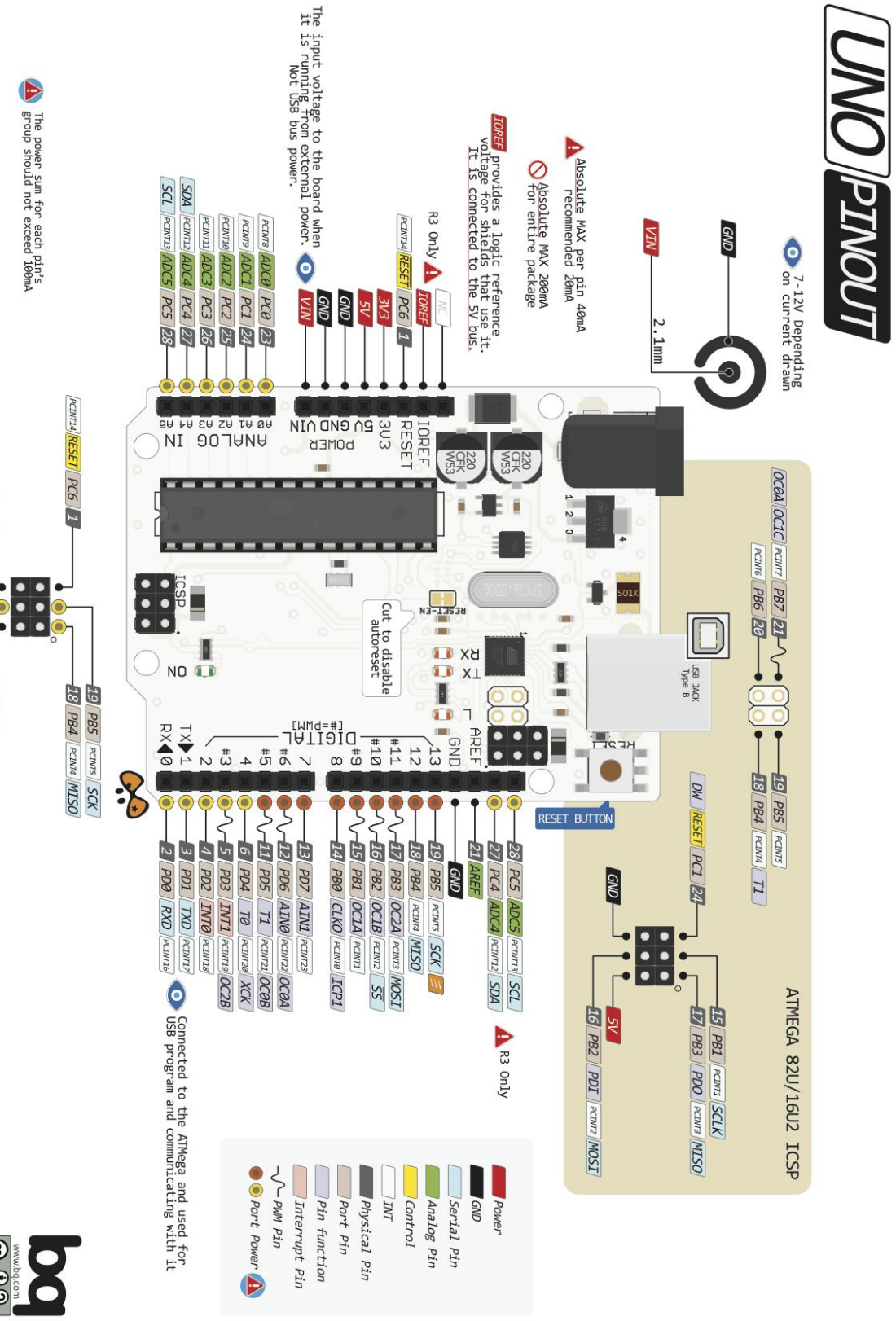
[26] Hamid HAMOUCHI. *Conception et réalisation d'une centrale embarquée de la domotique « Smart Home »*.06/07/2015, mémoire de master en génie électrique, Université Mohammed V École Normale Supérieure d'Enseignement Technique – Rabat.

[27] *The Application of PWM Capture (Data Acquisition) and Ultrasonic Sensors*[en ligne].

[28] Reas and Ben Fry. *Processing: A Programming Handbook for Visual Designers and Artists (Second Edition)*[en ligne]. publié December 2014, The MIT Press.

# ANNEXES

# ANNEXE A


UNO PINOUT
7-12V Depending on current drawn
GND
VIN
2.1mm
Absolute MAX per pin 40mA recommended 20mA
Absolute MAX 200mA for entire package
IOREF provides a logic reference voltage for shields that use it. It is connected to the 5V bus.
The input voltage to the board when it is running from external power. Not USB bus power.
The power sum for each pin's group should not exceed 100mA
R3 Only
RESET BUTTON
USB JACK Type B
ATMEGA 82U/16U2 ICSP
ANALOG IN
POWER
DIGITAL [#=PWM]
ICSP
Cut to disable autoreset
Connected to the ATMega and used for USB program and communicating with it
Power
GND
Serial Pin
Analog Pin
Control
INT
Physical Pin
Port Pin
Pin function
Interrupt Pin
PWM Pin
Port Power
www.bq.com
17 JUL 2014
ver 3 rev 0

# ANNEXE B

## Poursuit d'un objet fixe

| [Numéro de détection] | [Distance] | [Angle] |
|---|---|---|
| 1 | 309cm | 3° |
| 2 | 40cm | 22° |
| 3 | 39cm | 23° |
| 10 | 7cm | 29° |
| 11 | 36cm | 30° |
| 20 | 36cm | 37° |
| 21 | 36cm | 38° |
| 22 | 36cm | 39° |
| 23 | 15cm | 40° |
| 24 | 36cm | 41° |
| 35 | 36cm | 50° |
| 36 | 29cm | 51° |
| 45 | 5cm | 59° |
| 46 | 37cm | 60° |
| 47 | 37cm | 60° |
| 48 | 36cm | 61° |
| 49 | 36cm | 62° |
| 50 | 37cm | 63° |
| 72 | 60cm | 81° |
| 73 | 10cm | 82° |
| 74 | 64cm | 83° |
| 75 | 42cm | 107° |
| 76 | 40cm | 108° |
| 77 | 37cm | 109° |
| 78 | 39cm | 110° |
| 79 | 39cm | 110° |
| 80 | 39cm | 111° |
| 81 | 37cm | 112° |
| 82 | 39cm | 113° |
| 83 | 39cm | 114° |
| 84 | 38cm | 115° |
| 85 | 39cm | 116° |
| 86 | 38cm | 117° |
| 87 | 38cm | 117° |
| 88 | 38cm | 118° |
| 89 | 39cm | 119° |
| 90 | 39cm | 120° |
| 91 | 38cm | 121° |
| 92 | 38cm | 122° |
| 93 | 38cm | 122° |
| 94 | 38cm | 123° |
| 95 | 38cm | 124° |
| 96 | 38cm | 125° |
| 97 | 38cm | 126° |
| 98 | 38cm | 127° |
| 99 | 38cm | 127° |
| 100 | 38cm | 128° |
| 101 | 38cm | 129° |

# ANNEXE C

## Poursuit d'un objet mobile

| [Numéro de détection] | [Distance] | [Angle] |
|---|---|---|
| 1 | 8cm | 4° |
| 2 | 7cm | 6° |
| 3 | 6cm | 7° |
| 4 | 7cm | 8° |
| 5 | 7cm | 9° |
| 6 | 10cm | 11° |
| 7 | 10cm | 11° |
| 8 | 13cm | 12° |
| 9 | 14cm | 13° |
| 10 | 14cm | 14° |
| 11 | 13cm | 15° |
| 12 | 16cm | 16° |
| 13 | 15cm | 17° |
| 14 | 15cm | 17° |
| 15 | 15cm | 18° |
| 16 | 15cm | 19° |
| 17 | 14cm | 20° |
| 18 | 15cm | 21° |
| 19 | 14cm | 22° |
| 20 | 15cm | 23° |
| 21 | 15cm | 23° |
| 22 | 15cm | 24° |
| 23 | 17cm | 26° |
| 24 | 16cm | 27° |
| 25 | 15cm | 28° |
| 26 | 15cm | 29° |
| 27 | 15cm | 29° |
| 28 | 15cm | 30° |
| 29 | 15cm | 31° |
| 30 | 15cm | 32° |
| 31 | 16cm | 33° |
| 32 | 16cm | 34° |
| 33 | 14cm | 35° |
| 34 | 14cm | 35° |
| 35 | 14cm | 36° |
| 36 | 13cm | 37° |
| 37 | 13cm | 38° |
| 38 | 14cm | 39° |
| 39 | 16cm | 40° |
| 40 | 14cm | 41° |
| 41 | 12cm | 42° |
| 42 | 12cm | 42° |
| 43 | 15cm | 43° |
| 44 | 18cm | 44° |
| 45 | 26cm | 45° |
| 46 | 13cm | 46° |
| 47 | 13cm | 47° |
| 48 | 13cm | 47° |
| 49 | 13cm | 48° |

# ANNEXE D

## Code Arduino

```
#include "Servo.h"
#include "LiquidCrystal.h"
Servo motor;
LiquidCrystal lcd(12,11,5,4,3,2);
const int buzzer = 10 ;
const int pingPin = 8;
const int echoPin = 9;
const int rouge = A0;
const int vert = 13;
const int jaune = 7;
int echo;
char ping;
int ping0;
byte deg[]{14,17,17,14,0,0,0,0} ;     // tableu de charactèr °
   void setup()
     {
      Serial.begin (9600);              //initialisation conexion serie
      lcd.begin(16, 2);                 // Initialisation de l'écran (2 lignes 16 )
      pinMode(pingPin, OUTPUT);
      pinMode(echoPin, INPUT);
      pinMode(buzzer, OUTPUT);
      pinMode(rouge, OUTPUT);
      pinMode(vert, OUTPUT);
      pinMode(jaune, OUTPUT);
      motor.attach(6);
      lcd.createChar(1,deg);         // une fonction pour créer le caractère
     }
void loop()   {
      int duration, d;
      int i ;
      for (i=0;i<=180;i++) {
```

```
motor.write(i);
digitalWrite(pingPin, LOW);
 delayMicroseconds(2);
 digitalWrite(pingPin, HIGH);
 delayMicroseconds(10);
  digitalWrite(pingPin, LOW);  //trigger low = 0v
   duration = pulseIn(echoPin,HIGH);
   d=duration/58;
   echo=analogRead(echoPin);
    //   Serial.println(echo);
     if((d<100) && (d>60) ) {
           tone(buzzer, 3000);
           digitalWrite(vert, HIGH);
           digitalWrite(rouge, LOW);   }
       if((d<60) && (d>30) ) {
            tone(buzzer, 1500);
            digitalWrite(jaune, HIGH);
            digitalWrite(rouge, LOW);
            digitalWrite(vert, LOW);
              }
       if(d<30) {
            tone(buzzer, 300) ;
           digitalWrite(rouge, HIGH);
           digitalWrite(jaune, LOW);
           digitalWrite(vert, LOW);
                     }
    if (d>100){
           noTone(buzzer);
           digitalWrite(rouge, LOW);
           digitalWrite(jaune, LOW);
           digitalWrite(vert, LOW);
                     }
```

```
        Serial.print(i);
        Serial.print(",");
        Serial.print(d);
        Serial.print(".");
        lcd.setCursor(0,0);
        lcd.print("distance:");lcd.print(d);lcd.print(" cm");lcd.print(" ");
        lcd.setCursor(0,1);
        lcd.print("l'angle:");lcd.print(i);lcd.write(byte(1));lcd.print("    ");
        delay(400);
                }
for (i=180;i>=0;i--)
                {
        motor.write(i);
        digitalWrite(pingPin, LOW);
        delayMicroseconds(2);
        digitalWrite(pingPin, HIGH);
        delayMicroseconds(10);// il reste 5v pandant 10 micro sec
        digitalWrite(pingPin, LOW);  //trigger low = 0v
        duration = pulseIn(echoPin,HIGH);
         d=duration/58;
        echo=analogRead(echoPin);
   //   Serial.println(echo);
  if((d<100) && (d>60) ) {
          tone(buzzer, 3000);
          digitalWrite(vert, HIGH);
          digitalWrite(rouge, LOW);
          digitalWrite(jaune, LOW);   }
    if((d<60) && (d>30) ) {
          tone(buzzer, 1500);
          digitalWrite(jaune, HIGH);
          digitalWrite(rouge, LOW);
          digitalWrite(vert, LOW);
                        }
```

```
if(d<30) {
        tone(buzzer, 300);
        digitalWrite(rouge, HIGH);
        digitalWrite(jaune, LOW);
        digitalWrite(vert, LOW);
                              }
  if (d>100){
        noTone(buzzer);
        digitalWrite(rouge, LOW);
        digitalWrite(jaune, LOW);
        digitalWrite(vert, LOW);
                  }
                Serial.print(i);
                Serial.print(",");
                Serial.print(d);
                Serial.print(".");
                lcd.setCursor(0,0);
                lcd.print("distance:");lcd.print(d);lcd.print(" cm");lcd.print(" ");
                lcd.setCursor(0,1);
                lcd.print("l'angle:");lcd.print(i);lcd.write(byte(1));lcd.print("    ");
                delay(400);
            }}
```

# ANNEXE E

## Code Processing

```
import processing.serial.*;
Serial myPort;
String data;
Table table;
int k = 0 ;
int ii,flag,numero;
 int deg,dis,index1;
String val ;
String angle ;
String radius ;
int click=0;
PImage bg;
String time,diss,degg;

          void setup() {
  size(1500, 550); // la taille de fenetre
 // myPort = new Serial ( this, "COM2", 9600);//pour la simullation
 myPort = new Serial ( this, "COM7", 9600);//pour la réalisation
 myPort.bufferUntil('.');
 bg= loadImage("1.jpg");
 table = new Table();
 table.addColumn("[Numero de détection]");
  table.addColumn("[Distance]");
  table.addColumn("[Angle]");
  table.addColumn("[Heur de test]");    }

            void serialEvent(Serial myPort) {
 data = myPort.readStringUntil('.') ;
  data = data.substring(0,data.length()-1);
 index1 = data.indexOf(",");
  angle= data.substring(0, index1);
  radius= data.substring(index1+1, data.length());
```

```
    deg=int(angle);
    dis=int(radius);
    dis= dis*5;  }
    void mouseClicked() {
     click = 1;
     }

              void draw()  {
background(bg);
if(click==1){
 background(000);
 if((dis<500) && (dis>4)) {
  clearAll ();
 pushMatrix();
 translate(550,550);
 rotate(radians(-90));  // direction of x axe to the right and y axes is up
  fill(255, 0, 0); // couleur rouge
  stroke(#000000); // couleur des bordures
  ellipse( dis * sin(radians(deg)),  dis * cos(radians(deg)), 20,20);
   if((deg == 180)||(deg == 0))   {
   delay(1000);
   ellipse(dis * sin(radians(deg)), dis * cos(radians(deg)) , 20,20);   }
    popMatrix();
             ii++;
   if (ii==19) {
    flag=1;
  time=String.valueOf(hour())+":"+String.valueOf(minute())+":"+String.valueOf(secon
  d());  }
   if(flag==1) {
  diss=String.valueOf(dis/5)+"cm";
  degg=String.valueOf(deg)+"°";
  TableRow newRow = table.addRow();
   newRow.setInt("[Numero de détection]", table.getRowCount());
   newRow.setString("[Distance]", diss);
```

```
newRow.setString("[Angle]", degg);
newRow.setString("[heur de test]", time);
saveTable(table, "tableau.html");
ii=0;
flag=0;   }}
if(dis>500){
stroke(#00ff00);
arc(550, 550, 1000, 1000, PI, TWO_PI);
fill(#000000);
   for (int i = 0; i < 11; i++)   {
 arc(550,550,1000-(i*100),1000-(i*100),PI,TWO_PI); }
 textSize(20);
 fill(255, 255, 0);
  text("0°",1070,550);
  text("45°",925,190);
  text("90°",555,30);
  text("135°",190,190);
  text("180°",19,550);
  line(550,550,550,50);
  line(550, 550,199,195);
  line(550, 550,900,195);
  textSize(20);
  textAlign(CENTER);
  fill(#ffffff);
        for (int i = 0; i < 12; i++)  {
    text(i*10, 550, 550-(50*i))   ;  }
    text("cm",585, 50)   ;
    fill(#000000);
    rect(1090,0,300,550,15);
   fill(20, 148, 20);
   text("Aucun Objet  ",1220,100);
       }  }
```

```
      void clearAll ()   {
  stroke(#00ff00);
  arc(550, 550, 1000, 1000, PI, TWO_PI);
  fill(#000000);
        for (int i = 0; i < 11; i++)   {
  arc(550,550,1000-(i*100),1000-(i*100),PI,TWO_PI);   }
textSize(20);
fill(255, 255, 0);
 text("0°",1070,550);
  text("45°",925,190);
  text("90°",555,30);
   text("135°",190,190);
  text("180°",19,550);
   line(550,550,550,50);
   line(550, 550,199,195);
   line(550, 550,900,195);
   textSize(20);
   textAlign(CENTER);
   fill(#ffffff);
             for (int i = 0; i < 12; i++) {
  text(i*10, 550, 550-(50*i))   ;   }
   text("cm",585, 50)   ;
  fill(#000000);
  rect(1090,0,300,550,15);
  textSize(20);
  fill(255,255,255);
   text("Distance=",1175,30);
   text(radius,1245,30);
  text("Cm",1275,30);
   fill(255, 255, 0);
   text("Angle    =",1175,60);
  text(angle,1250,60);
  text("°",1273,60);
  if(deg%2==0) {
```

```
stroke(#000000);
fill(#ff0000);
ellipse(1130,90, 20,20);
text("Objet Détecté !",1220,100);  }
else{
  stroke(#000000);
  fill(#000000);
ellipse(1130,90, 20,20);
text("Objet Détecté !",1220,100);  }  }
```