

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

*Université de BLIDA 1*

*Institut d'Aéronautique et des Etudes spatiales*

Département de navigation aérienne



04/10/18  
Ex 1



*Mémoire*

*Pour l'Obtention du Diplôme*

*MASTER Aéronautique*

*Option: CNS/ATM*

Réalisation d'une station météorologie et un radar de  
Localisation à base d'ARDUINO

*Dirigé par :*

*Mr. BENSELAMA Zoubir*

*Présenté par :*

*Mr. BACHIRI Billal*

*Mr. SEDDIK Abderrezak*

**Promotion: 2017/2018**

## **Dédicace**

**Je dédie ce travail à Ma mère qui m'a entouré d'amour,  
d'affection qui a sacrifier sa vie pour moi et qui fait tout  
pour ma réussite, que dieu la garde**

**Ma sœur Samira qui m'a aidé durant tout mon parcours**

**A ma grande famille**

**Enfin, je remercie tous mes Ami(e)s que j'aime tant Pour  
leur sincère amitié et confiance**

**Bilal**

## Résumé

Ce projet a pour but de faire une étude générale des tours de contrôle de navigation aérienne en réalisant un prototype de système radar capable de détecter des obstacles et une station météo pour mesurer les paramètres météorologiques grâce à des capteurs et à base d'une carte ARDUINO pour traiter les données récupérées ensuite les envoyer au PC pour l'affichage de balayage radar et à l'afficheur LCD pour afficher les informations météo.

**Mot clé :** radar, ARDUINO, météo, détecté, capteur, LCD

يهدف هذا المشروع إلى إجراء دراسة عامة لأبراج مراقبة الحركة الجوية من خلال إنتاج نظام رادار ذي نموذج أصلي للكشف عن العوائق ومحطة الطقس لقياس معالم الأرصاد الجوية بفضل أجهزة الاستشعار وعلى أساس لوحة آردوينو لمعالجة البيانات التي تجمع وإرسالها إلى جهاز الكمبيوتر لمسح شاشة الرادار و LCD لعرض معلومات الطقس

كلمات مفتاحية: رادار، آردوينو، الطقس، لكشف، الاستشعار

The purpose of this project is to make a general study of air traffic control towers by producing a prototype radar system capable of detecting obstacles and a weather station for measuring meteorological parameters thanks to sensors and based on an ARDUINO chart to process the recovered data then sent them to the PC for radar scan display and LCD display for displaying weather information.

Keyword : Radar, ARDUINO, weather, Detecting, sensors

### Leste des abréviations

**Radar** radio detection and ranging

**IEEE** institute of electrical and electronics

**UIT** union internationale des télécommunications

**CW** Continuous Wave.

**MTI** moving target indicator

**PRF** pulse repetition frequency

**IPP** interpulse period

**PRI** pulse repetition interval

**Fr** fréquence reçue

**Ft** fréquence transmise

**METAR** météorologie aéroport rapport

**TAF** terminal aéroport prévision

**i/o** input/output

**ROM** read only mémoire

**USB** universal serial bus

**RAM** random access memory

**CNA** convertisseur numérique analogique

**IDE** integrated development environment



## Sommaire

Introduction générale .....	1
<b>Chapitre 1 : GENERALITE SUR LES TOURS DE CONTRÔLE</b>	
1.1 Introduction .....	3
1.2 Principe générale de radar .....	3
1.2.1 Définition de radar.....	3
1.2.2 Principe de fonctionnement.....	4
1.2.3 Fréquence utilisé dans un système radar .....	5
1.2.4 Classification des radars .....	6
1.2.4.1 radars primaires .....	6
1.2.4.1.1 Radar à ondes continues.....	6
1.2.4.2 Radar à impulsions.....	7
1.2.5.2 Radar secondaire.....	8
1.2.6 Antenne pour les radars.....	10
1.2.6.1 Antennes matelas.....	10
1.2.6.2 Antenne parabolique.....	10
1.2.7 Game .....	11
1.2.7.1. Résolution de plage .....	12
1.2.7.2 Fréquence doppler.....	13
1.2.8 Équation radar.....	15
1.2.8.1 Etablir de l'équation radar.....	15
1.2.8.2 La portée maximale en fonction du bruit.....	16
1.2.9 Type de radar.....	17
1.2.9.1 Les catégories des radars utilisés dans une tour de contrôle.....	18
1.3 Généralité sur la météorologie.....	18
1.3.1 Atmosphère .....	18
2.3.1.1 Définition de l'atmosphère.....	18
2.3.1.2 Structure de l'atmosphère.....	19
1.3.2 Pression atmosphérique.....	20

1.3.2.1	Caractéristiques de la pression atmosphère .....	20
1.3.3	Température de l'aire .....	21
1.3.3.1	Définition de la température .....	21
1.3.3.2	Unité .....	21
1.3.4	Humidité.....	22
1.3.5	Météorologie aéronautique.....	23
1.3.5.1	Type de prévision météorologique aéronautique.....	23
1.3.5.1.1	Les Stations de Météorologie Aéronautique.....	23
1.3.5.1.2	Observations radar.....	24
1.3.5.1.3	Observations par satellites météorologiques.....	24
1.4	Conclusion .....	25

## **Chapitre 2 : ETUDE DE LA CARTE ARDUINO**

2.1	Introduction.....	27
2.2	Définition de l'Arduino .....	27
2.2.1	LE MICROCONTROLEUR ARDUIN.....	28
2.2.2	Les composants d'un microcontrôleur .....	28
2.3	Matériel Arduino .....	29
2.3.1	Alimentation.....	30
2.3.2	Visualisation.....	31
2.3.3	La connectique.....	31
2.4	Type de carte arduino.....	32
2.4.1	Carte Arduino UNO.....	32
2.4.2	Carte Arduino Leonardo.....	33
2.4.3	Carte Arduino Méga.....	34
2.4.4	Carte Arduino Mega ADK.....	34
2.4.5	Carte Arduino Due.....	35
2.4.6	Carte Arduino Nano.....	35
2.4.7	Carte Arduino Mini Pro.....	36
2.4.8	Carte Arduino Yun.....	36
3.5	Étude de la partie logicielle.....	38
3.5.1	Le langage de programmation.....	38
3.5.2	L'interface du logiciel.....	38
3.5.3	Les boutons.....	39

3.5.4 La structure d'un programme.....	40
3.6 Le logiciel PROCESSING.....	40
3.6.1 L'interface de PROCESSING.....	41
3.6.2 Communication entre la carte ARDUINO et le logiciel PROCESSING.....	42
3.7 Conclusion.....	43

## **Chapitre 3 : REALISATION DE PROJET**

3.1 Introduction.....	45
3.2 Description de projet .....	45
3.3 Schéma synoptique .....	45
3.4 Description du matériel .....	46
3.4.1 Carte arduino UNO .....	46
3.4.2 Servomoteur .....	46
3.4.3 Capteur sonar à Ultrasons HC-SR04.....	47
3.4.3.1 Caractéristiques du capteur ultrasons HC-SR04.....	48
3.4.3.2 Mise en œuvre du module HC-SR04.....	48
3.4.4 Capteur de température et d'humidité DHT11 .....	49
3.4.5 Capteur de pression.....	50
3.4.5.1 Capteur de pression mpx4250dp.....	50
3.4.6 Afficheur LCD.....	50
3.4.6.1 Description des pins d'afficheur LCD.....	50
3.4.7 Potentiomètre.....	52
3.5 Réalisation de projet .....	52
3.5.1 Présentation de l'interfaçage Arduino-PC par câble USB.....	52
3.5.2 Branchement les capteurs a la carte arduino .....	53
3.5.3 Alimentation de la carte et des capteurs.....	58
3.5.4 Précission de capteur ultrason.....	58
3.5.5 Performances des capteurs à Caractéristiques métrologiques.....	59
4..5.6 Diagramme de programme.....	60
3.5.7 Installation de montage à l'extérieur de Laboratoire.....	61
3.5.8 Interface de balayage radar .....	62
3.6 Conclusion .....	63
Conclusion générale.....	64
Programme .....	67

## Liste des figures

Figure 1.1: Principe de radar .....	4
Figure 1.2 : Classification des radars .....	6
Figure 1.3 : Signal émis et reçus .....	7
Figure 1.4 : Paramètre d'un radar a impulsion .....	7
Figure 1.5 : Principe de la suppression des lobes secondaire a l'interrogation .....	9
Figure 1.6 : Intensité relative des impulsions .....	9
Figure 1.7 : Antenne matelas .....	10
Figure 1.8 : Antenne parabolique .....	11
Figure 1.9 : Diagramme synoptique radar pulse simplifier .....	11
Figure 1.10: Train d'impulsion transmit et reçus .....	12
Figure 1.11 : Détection de deux cibles .....	13
Figure 1.12 : Effet du mouvement de la cible sur les formes d'onde de l'equiphase réfléchi.....	13
Figure 1.13 : Équation radar.....	16
Figure1.14 : Différent types de bruit traité par un récepteur.....	17
Figure 1.15 : Structure de l'atmosphère .....	19
Figure 1.16 : La variation de la pression en altitude .....	20
Figure 1.17 : Station météorologique a transmission radio .....	24
Figure 1.18 : Tour aéronautique de station météorologique .....	24
Figure 1.19 : Satellite météorologique (tiros1).....	25
Figure 2.1 : Défèrent composent de la carte arduino.....	27
Figure 2.2 : Symbole d'un convertisseur numérique analogique.....	29
Figure 2.3 : Présentation de la carte ardino .....	30
Figure 2.4 : Organisation phtisque de la carte arduino.....	32
Figure 2.5 : Carte arduino uno.....	33
Figure 2.6 : Carte arduino lionardo .....	33
Figure 2.7 : Carte arduino mega.....	34
Figure 2.8 : Carte arduino ADK.....	34
Figure 2.9 : Carte arduino DUE.....	35

Figure 2.10 : Carte arduino nano.....	35
Figure 2.11 : Carte arduino mini pro.....	36
Figure 2.12 : Carte arduino yun .....	36
Figure 2.13 : Interface de logiciel arduino.....	38
Figure 2.14 : Les boutons de logiciel arduino .....	39
Figure 2.15 : Structure de programme arduino.....	40
Figure 2.16 : Interface de logiciel processing .....	41
Figure 2.17 : Communication entre la carte arduino et le logiciel processing .....	42
Figure 3.1 : Schéma synoptique de la réalisation .....	45
Figure 3.2 : Carte arduino uno .....	46
Figure 3.3 : Servomoteur .....	46
Figure 3.4 : Fils de connexion d'un servomoteur .....	47
Figure 3.5 : Capteur sonore a ultrason HC-SR04.....	47
Figure 3.6 : Signal entré et sorti du capteur HC-SR04.....	49
Figure 3.7 : Capteur de température et d'humidité DHT11.....	49
Figure 3.8 : Capteur de pression mpx 4250 dp .....	50
Figure 3.7 : afficheur LCD.....	51
Figure 3.10 : Potentiomètre .....	52
Figure 3.11 : Image réel de montage connecté par USB .....	52
Figure 3.12 : Branchement de servomoteur a la carte arduino.....	53
Figure 3.13 : Branchement de capteur ultrason a la carte arduino .....	54
Figure 3.14 : Branchement de capteur DTH 11 a la carte arduino .....	55
Figure 3.15 : Branchement de mpx 4250 dp à la carte arduino .....	55
Figure 3.16 : schéma global de la réalisation .....	56
Figure 3.17 : Alimentation de la carte arduino .....	58
Figure 3.18 : précision du capteur ultason.....	59
Figure 3.19 : Graphique exprime la variation de température .....	61
Figure 3.20 : Graphique exprime la variation d'humidité .....	61
Figure 3.21 : Graphique exprime la variation de la pression .....	62
Figure 3.22 : interface du projet radar.....	62

## **Liste les tableaux**

Tableau 1.1 : les bandes de fréquence radar .....	5
Tableau 1.2 : la pression de vapeur d'eau en fonction de la température et la degré de saturation de vapeur .....	22
Tableau 2.1 : tableau comparatif entre différents carte arduino.....	37
Tableau 3.1 : spécification de capteur sonar a ultrason HC-SR04.....	48
Tableau 3.2 : présentation de branche de l'afficheur LCD .....	51
Tableau 3.3 : connexion des pins de circuit .....	57
Tableau 3.4 : précession de capteur ultrason.....	58

## **Introduction générale**

L'aéronautique est un domaine important dans la vie quotidienne elle permet de faire plusieurs activités aériennes ou de déplacer d'un point à un autre en toute sûreté et sécurité, elle est aussi un domaine très vaste de recherche scientifique et de nouvelle technologie en construction aérienne ou en navigation aérienne.

La tour de contrôle est l'un des points fondamentaux dans ce domaine, elle est divisée en plusieurs étages comme l'information aéronautique et le contrôle aérien qui utilise plusieurs instruments de navigation de communication et de surveillance.

Dans notre projet nous allons réaliser un radar de détection à l'aide d'un capteur ultrason et une station météo pour mesurer les phénomènes physiques avec des capteurs météorologiques capteur température d'humidité et de pression.

Les deux instruments que nous allons réaliser seront connectés par une carte Arduino programmable par un ordinateur et un afficheur LCD.

L'objectif de notre projet est la réalisation d'un montage qui va détecter la distance des cibles leur angle d'orientation et capter les paramètres météorologiques.

# Chapitre 1 : généralité sur les tours de contrôle



## CHAPITRE 1 : Généralité sur les tours de contrôle

### 1.1 Introduction

La tour de contrôle est le bâtiment le plus caractéristique sur un aéroport. Elle est facilement reconnaissable, même intégrée à un bloc technique. En premier lieu toutes les correspondances entre avions et contrôleurs se font par radio. De même la coordination entre les différents éléments se fait essentiellement par communications vocales directes (téléphone) ou par systèmes automatisés ou liaison entre ordinateurs. Toutes les informations qui se transmettent s'articulent sur deux éléments nécessaires au contrôle aérien, le radar qui permet d'analyser l'environnement de l'espace aérien et une station météo pour détecter les paramètres météorologiques, la température ; humidité la pression et surtout la pluviométrie et la visibilité et ceci pour une prise de décision adéquate pour la navigation aérienne. Nous allons, dans ce chapitre, décrire et détailler les éléments essentiels dont on vient de parler à savoir les radars et la station météo.

### 1.2 Principe générale de radar

Le radar est une invention qui a révolutionné le domaine de l'aviation et de la stratégie militaire au XX<sup>e</sup> siècle qui est utilisé en météorologie, en astronautique ou encore pour le contrôle du trafic aérien et du trafic routier.

#### 1.2.1 Définition de radar

##### Radar

Un système de mesure radar complet comprend un émetteur avec antenne, un trajet de transmission, la cible réfléchissante, un autre trajet de transmission (généralement identique au premier) et un récepteur avec antenne. Deux antennes séparées peuvent être utilisées, mais souvent seulement l'un est utilisé pour l'émission et la réception du signal radar [1].

##### 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 Principe de fonctionnement

Le principe utilisé par les radars est voisin de celui de la réflexion des ondes sonores. Lorsque vous criez dans la direction d'un objet qui peut réfléchir le son de votre voix (dans un canyon ou dans une grotte par exemple), vous en entendez l'écho. Si vous connaissez la vitesse du son dans l'air, vous pouvez alors estimer la distance et la direction générale de l'objet. Le temps nécessaire à l'aller et au retour du son vers vous peut être converti en distance si vous connaissez sa vitesse.

Le radar utilise des impulsions d'énergie électromagnétique à peu près de la même manière, Le signal hyperfréquence est émis en direction de la cible. Une petite partie de l'énergie transmise est réfléchié par la cible dans la direction du radar. Cette énergie renvoyée par la cible jusqu'au radar est appelée ECHO, exactement comme lorsque l'on considère les ondes sonores. Un radar utilise l'écho afin de déterminer la direction et la distance de l'objet qui a réfléchi son signal [2].

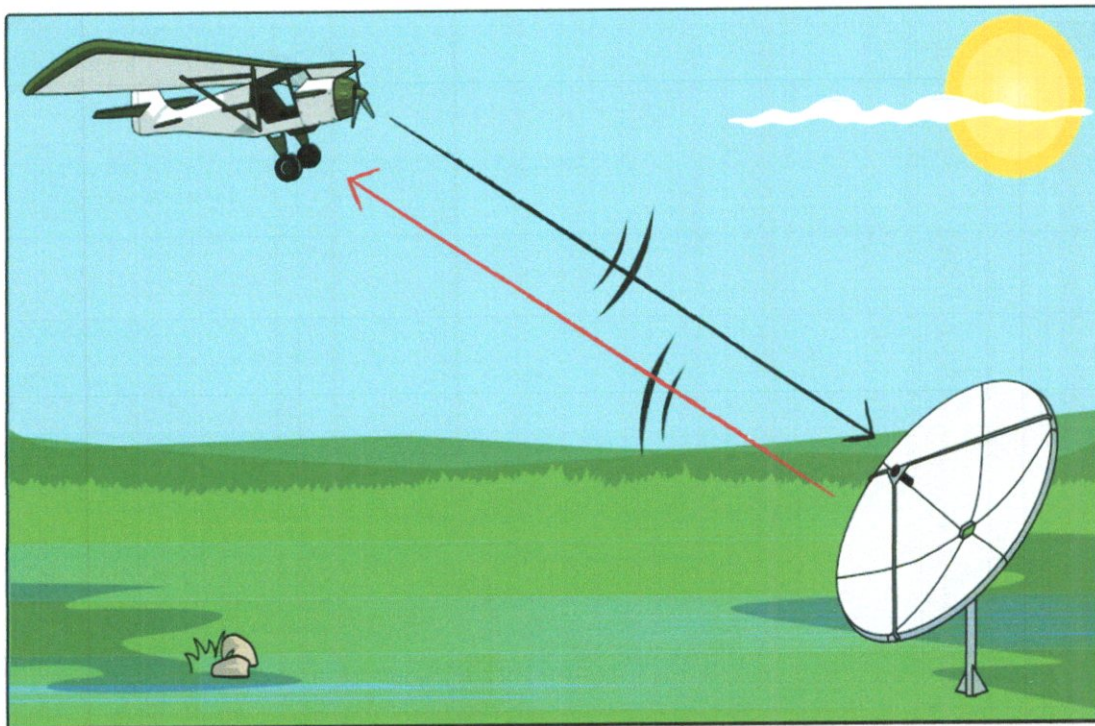


Figure 1.1: principe de radar



**1.2.3 Fréquence utilisé dans un 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 UIT (l'Union Internationale des Télécommunications) pour le Radar [3]

Longueur d'onde	nomenclature OTAN		bandes radar		
100 m	A	0 à 250 MHz			
10 m	B	250 à 500 MHz	HF	3 à 30 MHz	
1 m	C	0,5 à 1 GHz	VHF	30 à 300 MHz	
			UHF	0,3 à 1GHz	
30 cm	D	1 à 2 GHz	L	1 à 2 GHz	
	E	2 à 3 GHz	S	2 à	
10 cm	F	3 à 4 GHz		4 GHz	
5 cm	G	4 à 8 GHz	C	4 à	
	H	6 à 8 GHz		8 GHz	
3 cm	I	8 à 10 GHz	X	8 à 12 GHz	
2 cm	J	10 à	Ku	12 à 18 GHz	
1 cm		20 GHz		K	18 à 27 GHz
	K	20 à 40 GHz		Ka	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 : Les bandes de fréquence de radar [4]

### 1.2.4 Classification des radars

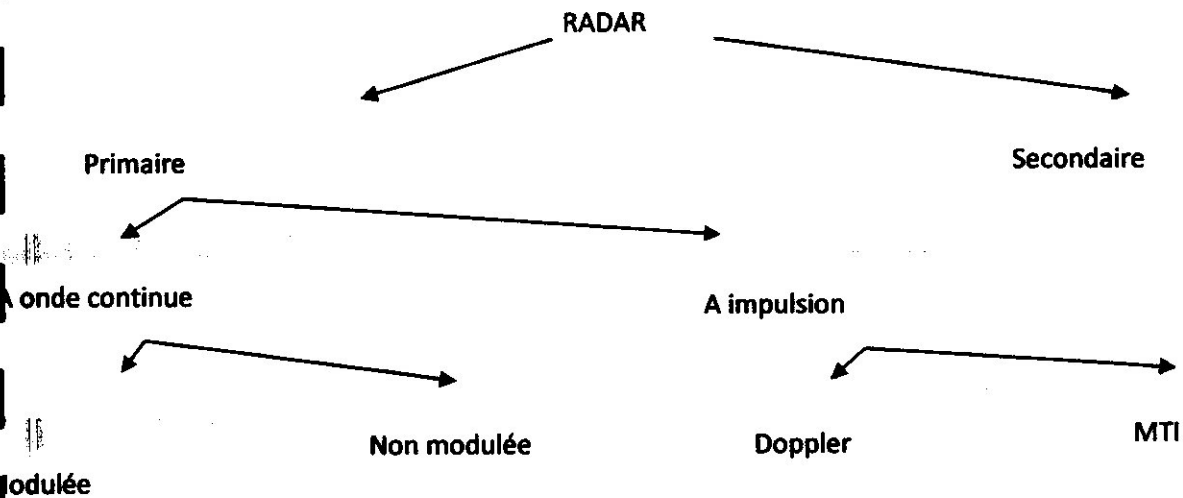


Figure 1.2 : classification de radar

#### 1.2.4.1 radars primaires

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]

##### 1.2.4.1.1 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). [5]

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 différencier 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  $f_1$  et  $f_2$ . En examinant les fréquences réfléchies reçues (et avec la connaissance de la fréquence transmise), le calcul de la distance peut être effectué par [5]

$$R = \frac{c \Delta t}{2} \quad (1.1)$$

Avec

R : distance

$\Delta t$  : différence de temps entre le signal émis et l'écho

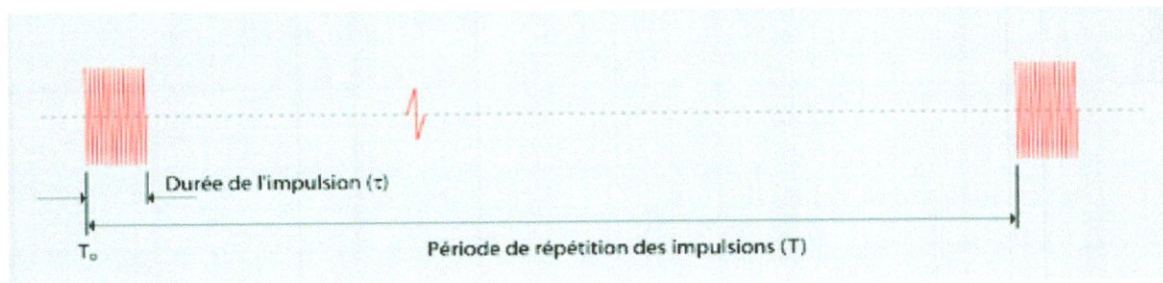


Figure 1.3 : signal émis et reçu

#### 1.2.4.2 Radar à impulsions

Ce type de Radar transmet des impulsions à hautes fréquences avec une puissance très élevée vers la cible. Ensuite, il attend l'écho du signal transmis pendant un certain temps avant qu'il ne transmette une nouvelle impulsion [5]

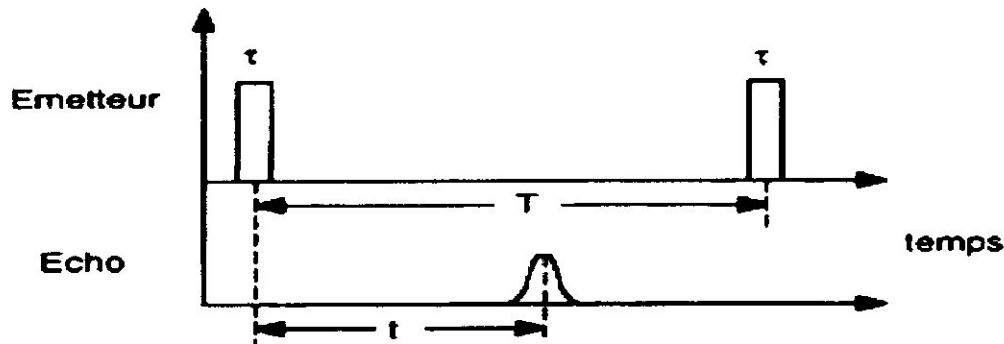


Figure 1.4 : paramètre d'un radar a impulsion

Deux types de radar utilisent le décalage doppler :

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

Ce Radar utilise une faible fréquence de répétition d'impulsions (PRF) pour éviter les ambiguïtés de distance, mais ces Radars peuvent avoir des ambiguïtés Doppler [5].

**- Radars Doppler à impulsions**

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

**1.2.5.2 Radar secondaire**

Un radar secondaire a généralement deux antennes l'antenne principale directionnelle et une seconde omnidirectionnelle.

La première émet deux impulsions en mode interrogation (P1 et P3) dont l'espacement temporel

Cette antenne a un faisceau principal mince (ou un faisceau somme mince dans le cas d'une antenne mono pulse) de l'ordre de  $2^\circ$  à  $3^\circ$  en azimut où elle concentre son énergie.

L'antenne omnidirectionnelle émet une impulsion de « contrôle » P2 de façon isotrope, juste après l'impulsion P1 Il peut s'agir d'un dipôle ou de la partie en opposition de phase dans une antenne mono pulse (ligne rouge de la figure 1.5) Cette impulsion doit être au moins 9 dB plus faible que les impulsions P1 et P3 dans le faisceau principal.

Les impulsions P1 et P3 sont aussi émises dans les lobes secondaires mais avec une intensité



beaucoup moindre. Il en découle que le rapport entre les intensités de P1 et P2 sera donc plus faible que 9 dB si un avion les reçoit à partir de la direction d'un lobe secondaire. Ce rapport sera de plus en plus faible à mesure qu'on s'éloigne de l'axe principal et P2 pourra même devenir plus intense que P1 après un certain angle [6].

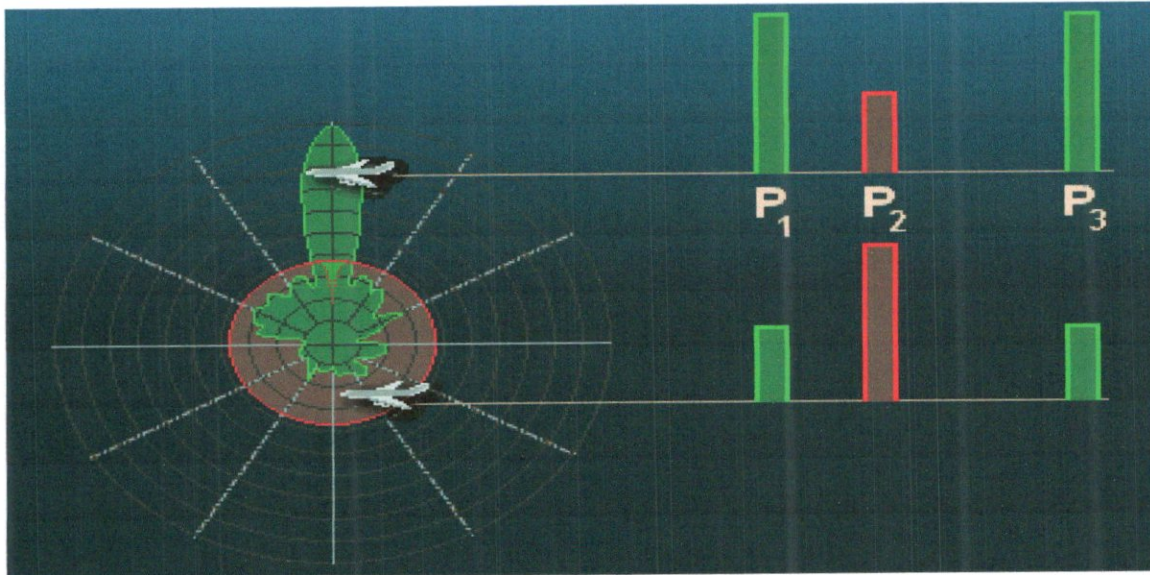


Figure 1.5 : Principe de la suppression des lobes secondaires à l'interrogation.

Il en résulte que si l'amplitude maximale de P2 est :

- de plus de 9 dB plus faible que celle de P1, le transpondeur reçoit bien la question du faisceau principal et y répond.
- est plus grande que P1 ou que la différence est de moins 9 dB, le transpondeur reçoit une question d'un lobe secondaire et ne doit pas répondre [6].

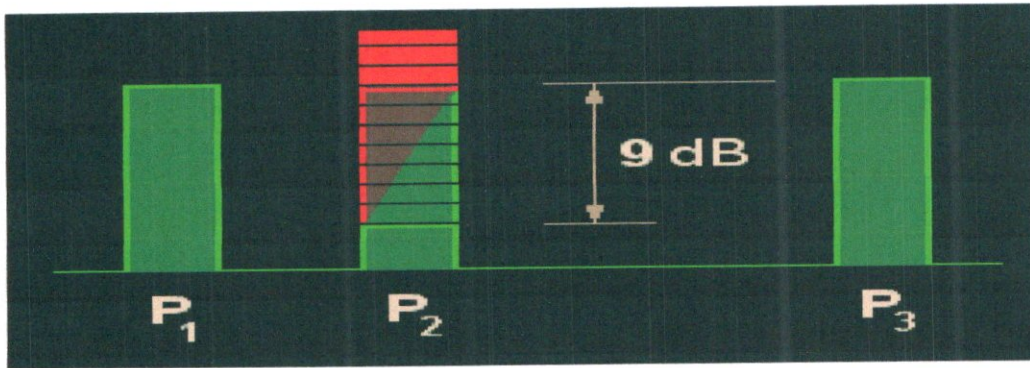


Figure 1.6 : Intensité relative des impulsions

### 1.2.6 Antenne pour les radars

Les radars utilisent plusieurs type d'antenne dans les plus important sont les suivant

#### 1.2.6.1 Antennes matelas

À la fréquence basses employées par les radars on utilise un rideau d'antennes il s'agit d'un ensemble de dipôles actifs  $\lambda/2$  située dans le même plan et alimentés en phase ils sont organisés en rangées étai également distantes de  $\lambda/2$  [7].

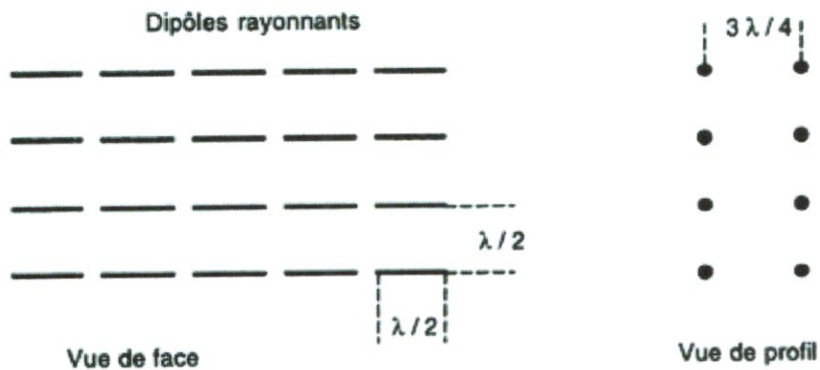


Figure 1.7 : antenne matelas

#### 1.2.6.2 Antenne parabolique

À la fréquence dépassant le gigahertz les antennes les plus utilisées sont les antennes paraboliques elles sont constituées de la source d'énergie et d'un réflecteur.

Le réflecteur est parfois réalisé à l'aide d'un grillage ou d'une tôle perforée si le diamètre des mailles est inférieur à la longueur d'onde le réflecteur peut prendre plusieurs formes [7].



## CHAPITRE 1 : Généralité sur les tours de contrôle

- Paraboloïde cylindrique
- Paraboloïde de révolution
- Paraboloïde de pelure d'orange

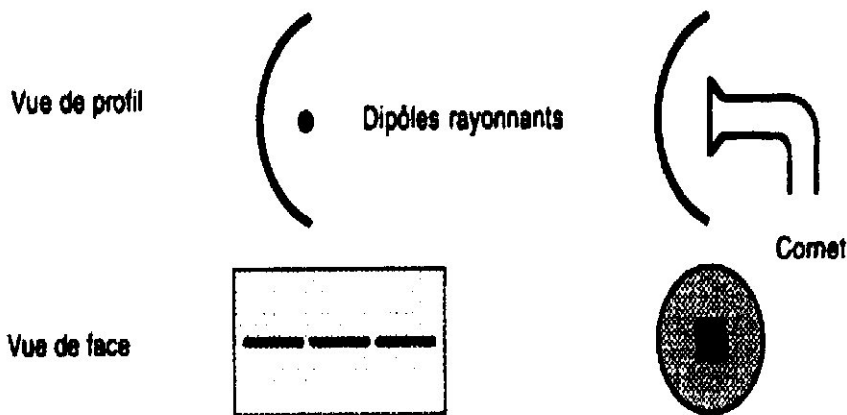


Figure 1.8 : antenne parabolique

### 1.2.7 GAMME

Un signal modulé est généré et envoyé à l'antenne par le bloc modulateur / émetteur. La commutation de l'antenne entre les modes d'émission et de réception est contrôlée par le duplexeur. Le duplexeur permet d'utiliser une antenne à la fois pour émettre et recevoir. Pendant la transmission, il dirige l'énergie électromagnétique du radar vers l'antenne. Alternativement, en réception, il dirige les échos radar reçus vers le récepteur.

Le récepteur amplifie les retours du radar et les prépare au traitement du signal. L'extraction des informations de la cible est effectuée par le bloc du processeur de signal. La plage de la cible est calculé en mesurant la temporisation; il faut une impulsion pour parcourir le chemin à double sens entre le radar et la cible. Puisque les ondes électromagnétiques se déplacent à la vitesse de la lumière [8].

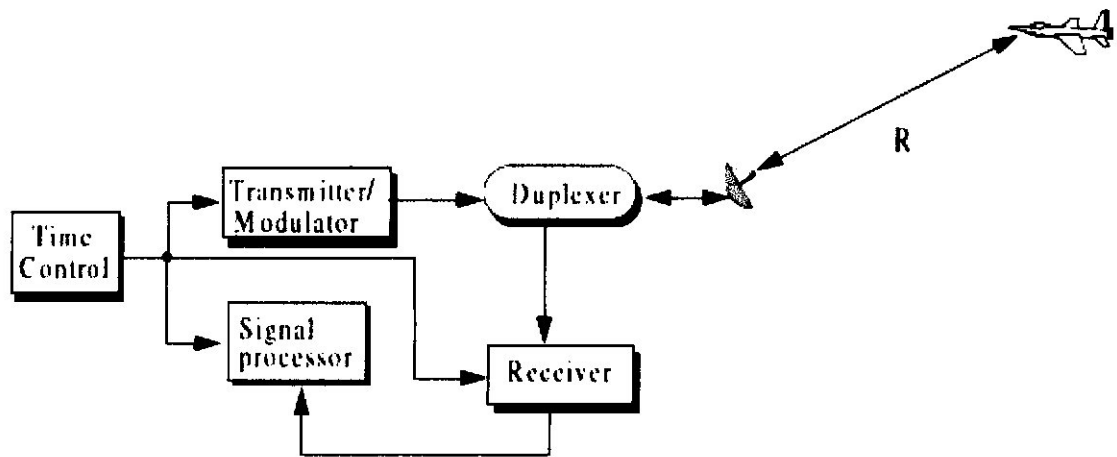


Figure 1.9 : Diagramme synoptique radar pulsé simplifié

En général un radar pulsé transmet et reçoit un train d'impulsions, comme illustré par la Figure (1.10) La période Inter Pulse (IPP) est  $\tau$  et la largeur d'impulsion est  $t$  L'IPP est souvent appelé Intervalle de répétition d'impulsion (PRI, Pulse Répétition Intervalle) L'inverse du PRI est le PRF qui est désigné par [8].

$$f_r = \frac{1}{\text{PRI}} = \frac{1}{\tau} \quad (1.2)$$

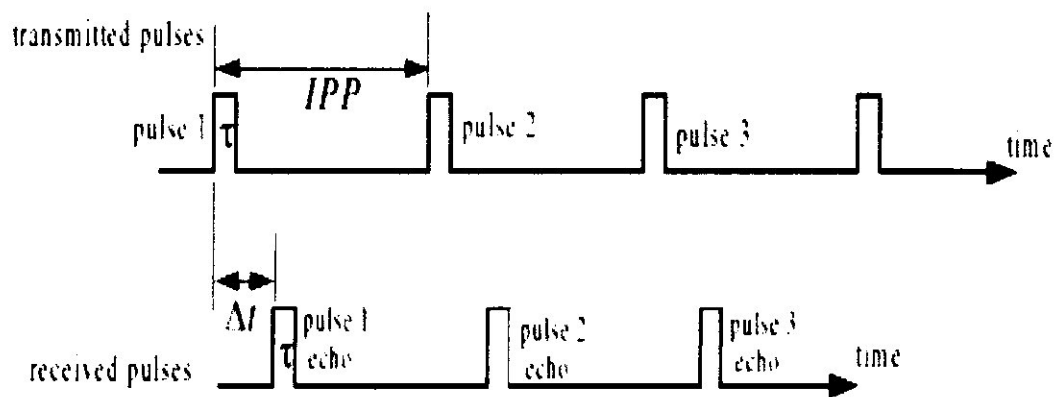


Figure 1.10 : Train d'impulsions transmises et reçues

### 1.2.7.1 Résolution de plage

La résolution de portée désignée par  $\Delta R$  est une mesure radar qui décrit sa capacité à détecter des cibles à proximité les unes des autres en tant qu'objets distincts. Les systèmes radar sont normalement conçus pour fonctionner entre une portée minimale  $R_{min}$  et portée maximale  $R_{max}$

Les cibles séparées d'au moins seront complètement résolues à portée, comme illustré à la figure (1.11) Les cibles situées dans le même intervalle peuvent être résolues en azimut en utilisant des techniques de traitement du signal [8].

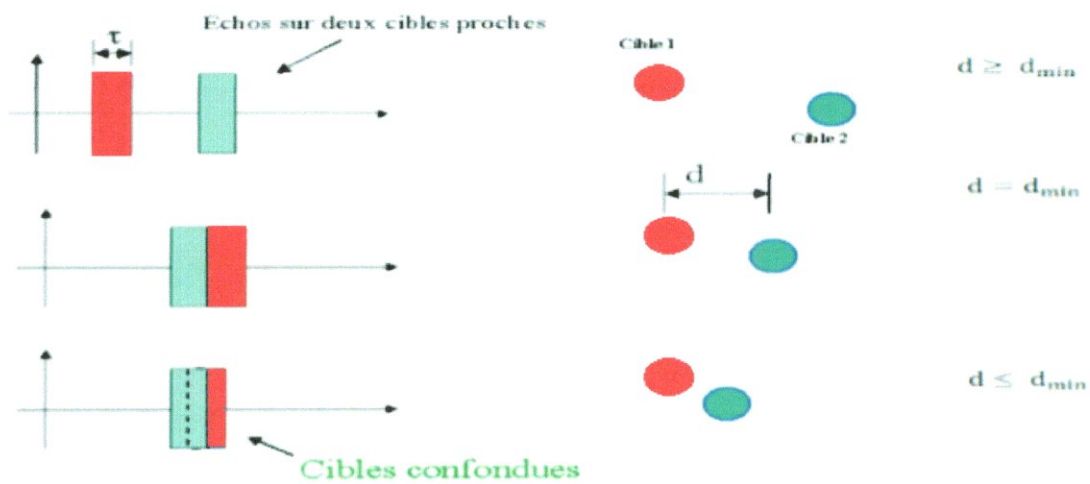


Figure 1.11 : Détection de deux cibles

### 1.2.7.2 Fréquence doppler

Les radars utilisent la fréquence Doppler pour extraire la vitesse radiale cible (taux de distance) ainsi que pour faire la distinction entre des cibles mobiles ou stationnaires ou des objets tels que le fouillis. Le phénomène Doppler décrit le décalage de la fréquence centrale d'une forme d'onde incidente due au mouvement de la cible par rapport à la source de rayonnement. Selon la direction du mouvement de la cible, ce décalage de fréquence peut être positif ou négatif. Un incident de forme d'onde sur une cible a des fronts d'onde Equiphase séparés par  $\lambda$  la longueur d'onde [8].

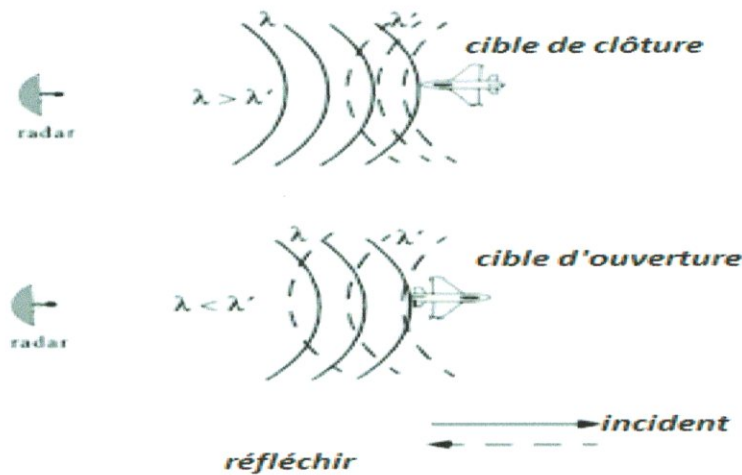


Figure 1.12 : Effet du mouvement de la cible sur les formes d'ondes de l'équiphase réfléchie

Un radar Doppler de base utilise l'effet Doppler-Fizeau de l'écho réfléchi par une cible pour mesurer sa vitesse radiale. Le signal micro-onde émis par l'antenne directionnelle du radar à fréquence  $F_t$  est réfléchi par la cible et comparé en fréquence avec le signal original allé et retour ( $F_r$  est la fréquence reçue) Il permet ainsi une mesure directe et extrêmement précise de la composante vitesse de la cible dans l'axe du faisceau. Une correction doit donc être appliquée pour obtenir sa vitesse réelle si l'opérateur connaît l'angle de déplacement de la cible par rapport à la radiale au radar [9].

La formule de la fréquence reçue de la cible dépend de la vitesse de la lumière dans l'air

$$F_r = F_t \left( \frac{1 + \frac{v}{c'}}{1 - \frac{v}{c'}} \right) \quad (1.3)$$

Avec

$F_r$  : Fréquence reçue

$F_t$  : Fréquence transmise

$c'$  : vitesse de la lumière

$v$  : vitesse de la cible

La variation Doppler est donc

$$F_d = F_r - F_t = 2 v \frac{F_t}{(c' - v)} \quad (1.4)$$

Et comme la vitesse de la lumière est beaucoup plus grande que la vitesse de la cible alors

$$F_d = 2 v \frac{F_t}{(c')} \quad (1.5)$$

Avec

$F_d$  : Fréquence doppler

## 1.2.8 Équation radar

### 1.2.8.1 Etablir de l'équation radar

Est un bilan des puissances sur le trajet aller-retour d'une onde émise par un radar. Celle-ci dépend des caractéristiques du radar (antenne, circuits électroniques, guide d'onde, pertes de signal, etc.), de celles de la cible et du milieu traversé le long du trajet

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

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

$$\frac{Pt}{4\pi R^2} \quad (1.6)$$

Pour une antenne directive

$$\frac{Pt}{4\pi R^2} G_t \quad (1.7)$$

Puissance réémise par la cible

$$\frac{PtG_t}{4\pi R^2} \sigma \quad (1.8)$$

Densité de puissance au niveau du Radar

$$\frac{PtG_t}{4\pi R^2} \frac{\sigma}{4\pi R^2} \quad (1.9)$$

Puissance reçue par le Radar

$$\frac{PtG_t}{4\pi R^2} \frac{\sigma}{4\pi R^2} A_e \quad (1.10)$$

D'où

$$\frac{PtG_t}{4\pi R^2} \frac{\sigma}{4\pi R^2} \frac{G_r \lambda^2}{4\pi} \quad (1.11)$$

Puisque le Radar est mono statique donc  $G_t=G_r= G$  et la puissance reçue devient :

$$P_r = \frac{P_t G^2 \sigma \lambda^2}{(4\pi)^3 R^4} \quad (1.12)$$

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

$$R_{max} = \left( \frac{P_t G^2 \sigma \lambda^2}{(4\pi)^3 S_{mis}} \right)^{1/4} \quad (1.13)$$

Avec:

$P_r$ : puissance du signal reçu (aux bornes de l'antenne)

$P_t$ : puissance du signal transmis (aux bornes de l'antenne)

$G_t$ : gain de puissance de l'antenne de transmission

$G_r$ : gain de puissance de l'antenne de réception

$\delta$  : section cible de radar

$\lambda$  : longueur d'onde

$R$  : distance radar-cible

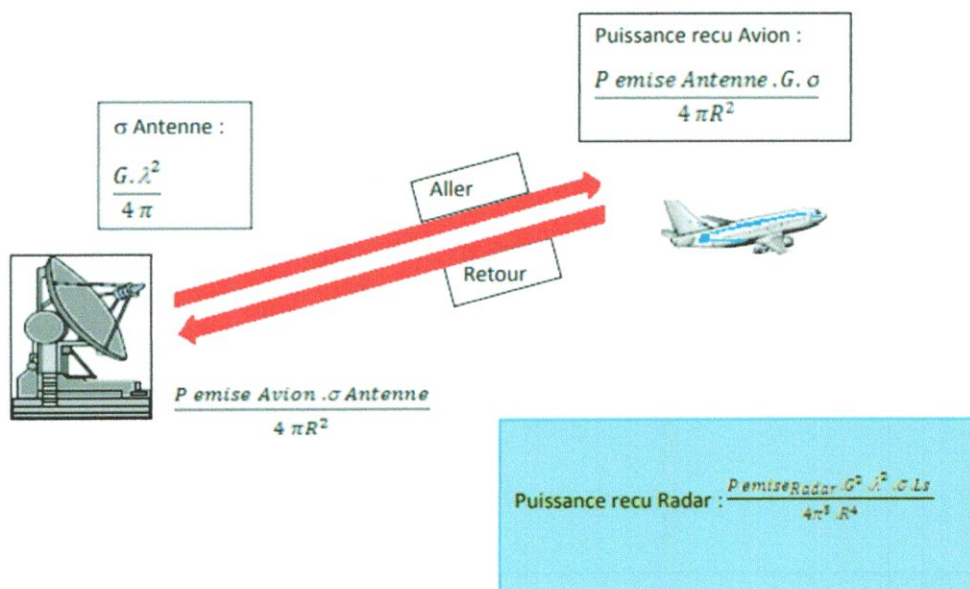


Figure 1.13 équation radar

### 1.2.8.2 La portée maximale en fonction du bruit

En considérons  $T_e$  la température effective de bruit de l'antenne, la puissance de bruit à l'entrée du récepteur est :

$$N_i = K T_e b_n \quad (1.14)$$

Comme :  $F_n = \frac{\text{bruit à la sortie d'un récepteur réel}}{\text{bruit à la sortie d'un récepteur idéal}} = \frac{S_i/N_i}{S_0/N_0}$

Avec Facteur de bruit :  $F_n$

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

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

$S_0$  : Bruit à la sortie d'un récepteur réel

$N_0$  : Bruit à la sortie d'un récepteur idéal

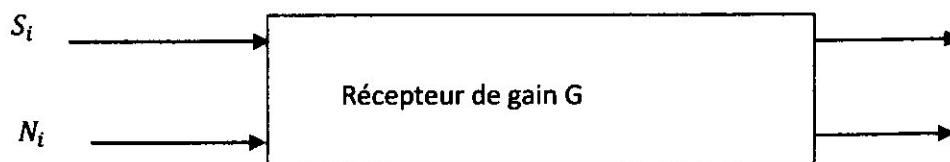


Figure 1.14 : Différents types de bruit traités par un récepteur

$$F_n = \frac{S_i/N_i}{S_0/N_0} \quad \text{d'où} \quad S_i = F_n N_i \left(\frac{S}{N}\right)_0$$

Alors on a :

$$S_{min} = F_n T_e K b_n \left(\frac{S}{N}\right)_{min} \quad (1.16)$$

Remplaçant  $S_{min}$  dans l'équation Radar (1.14) on aura :

$$R_{max} = \left( \frac{P_t G^2 \sigma \lambda^2}{(4\pi)^3 F_n T_e K b_n \left(\frac{S}{N}\right)_{min}} \right)^{1/4} \quad (1.16)$$

### 1.2.9 type de radar

- 1- Radars de surveillance aérienne
- 2- Radar de mesure de vitesse à onde continue
- 3- Radar météorologique

## CHAPITRE 1 : Généralité sur les tours de contrôle

4- Radioaltimètres

5- Radar Doppler aéroporté de navigation

6- Radars de veille pour navires

7- Applications spatiales du radar

8- Radars sol de poursuite de missiles et satellites

9- Radars de suivi de terrain

10- Radar à vision latérale aéroporté

### 1.2.9.1 Les catégories des radars utilisés dans une tour de contrôle

- **Radar sol**

Portée : 4 à 9 km

Résolution : 12 m

Bande X (9 GHz) ou Ku (16 GHz)

- **Radar d'approche**

Portée : (50 à 80 nm)

Bande de fréquence : Bande S ou L

- **Radars de contrôle en route**

Portée : 150 à 220 nm

Altitude : 7600 km

Bande de fréquence : Bande L

## 1.3 Généralité sur la météorologie

### 1.3.1 Atmosphère

#### 1.3.1.1 Définition de l'atmosphère

L'atmosphère terrestre est la fine pellicule gazeuse constituée d'air qui entoure notre planète. Elle est retenue par l'attraction terrestre et, du fait de sa compressibilité, elle va en se raréfiant constamment à mesure qu'on s'élève, La limite supérieure de l'atmosphère n'est pas nette : 50 % de sa masse se situe entre 0 et 5 km d'altitude, 90 % entre 0 et 16 km et 99% au-dessous de 32 km [10].



### 1.3.1.2 Structure de l'atmosphère

En partant du sol, la température commence par décroître jusqu'à une certaine altitude. Cette couche s'appelle la troposphère. On y trouve les nuages et les phénomènes météorologiques

Au-dessus, se trouve la stratosphère, où la température est soit constante, soit croît à nouveau. La limite entre ces deux couches s'appelle la tropopause,

La stratosphère s'étend jusqu'à la stratopause, qui se situe vers 50 km d'altitude et dont la température est comprise entre -20 et +20 degrés Celsius

Au-dessus de la stratopause, la température recommence à décroître, formant une couche appelée mésosphère limitée par la mésopause située entre 70 et 85 km d'altitude là où la température atteint un minimum de -120°C à -50°C

Au-dessus de la mésopause, la température recommence à croître, dans une couche appelée thermosphère, vers 600 km d'altitude, la température varie de 750°C à 1 100°C et présente un rythme diurne très marqué [10].

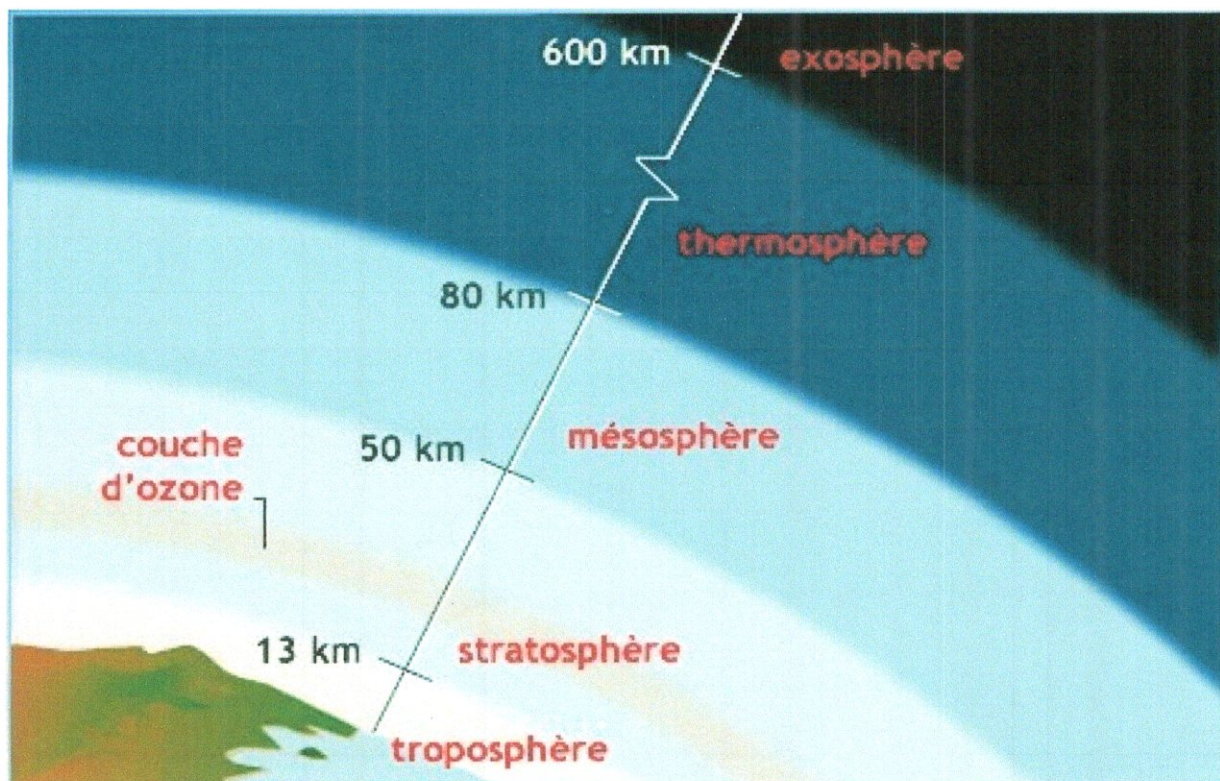


Figure 1.15 : structure de l'atmosphère

**Remarque :** En aviation civile les avions dépassent pas la troposphère si ta dire 11km d'après l'OACI

### 1.3.2 Pression atmosphérique

La pression atmosphérique peut être associée à la pression qu'exerce l'air sur un point donné. A un endroit précis, la pression atmosphérique est donc le poids exercé par une colonne d'air partant du sol et s'étirant jusqu'au sommet de l'atmosphère.

La pression se mesure à l'aide d'un baromètre et s'exprime en hectopascal (hPa). Un hectopascal équivaut à 100 Pascals (Pa) ou encore à 1 millibar [12].

#### 1.3.2.1 Caractéristiques de la pression atmosphérique

La pression atmosphérique doit être mesurée à une certaine altitude la pression diminue au fur et à mesure que l'altitude augmente, puisque le poids de l'air qui nous surplombe est moins important.

A noter également le fait que 90 % de la pression atmosphérique se concentre presque dans les 15 premiers kilomètres de l'atmosphère. Autrement dit, la pression va avoir tendance à diminuer plus intensément dans les basses couches qu'en altitude [11].

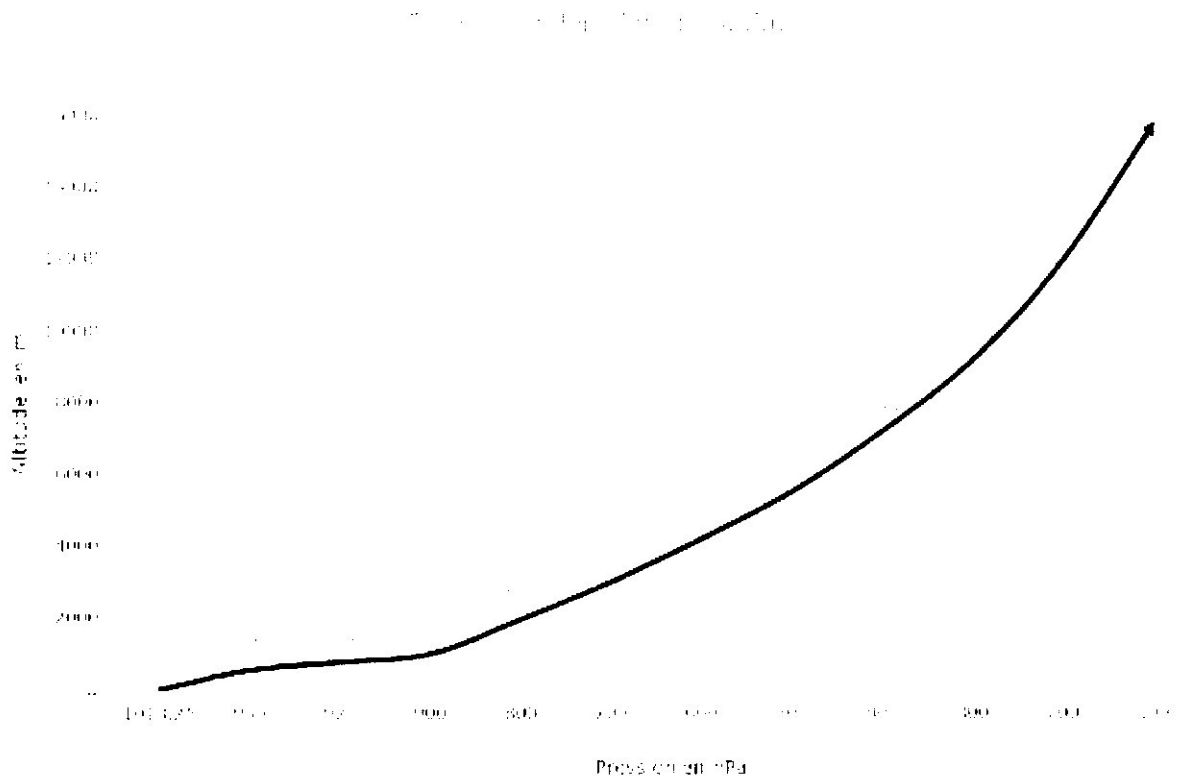


Figure 1.16 : la variation de la pression en fonction de l'altitude

**Remarque :** dans un aéroport L'instrument permettant de mesurer la pression atmosphérique est le baromètre

### 1.3.3 Température de l'air

#### 1.3.3.1 Définition

Dans un fluide (gaz ou liquide), les molécules sont en agitation permanente et se déplacent de façon désordonnée. C'est le mouvement brownien : les molécules se déplacent à grande vitesse et se heurtent sans cesse, La température est une grandeur qui caractérise l'état d'agitation des molécules.

#### 1.3.3.2 Unité

Une échelle de comparaison est donc nécessaire pour chiffrer, le degré de chaud ou de froid.

##### Échelle Celsius

L'unité est le degré Celsius (C).

A la pression atmosphérique normale (101 325 Pa) :

- la fusion de la glace correspond à 0 C
- l'ébullition de l'eau correspond à 100 C

Le degré Celsius (anciennement dénommé degré centigrade) est la  $\frac{1}{100}$  partie de cet espace 0 C -100 C et les températures inférieures à 0 C sont comptées négativement.

##### Échelle Kelvin

L'unité est le kelvin (**K**) et représente l'unité de température dans le système international

Il ne peut donc pas y avoir de température négative dans ce système.

A pression atmosphérique normale (101 325 Pa) :

- la fusion de la glace (0 °C) correspond à 273,15 K
- l'ébullition de l'eau (100 °C) correspond à 373,15 K.

Pour convertir de degré de celsius en kelvin

$$T(k)=t (^{\circ}C) +275 \quad (1.18)$$

### Échelle Fahrenheit

L'unité est le degré Fahrenheit (°F)

Seuls les Etats-Unis utilisent encore cette échelle,

Elle diffère des deux autres à la fois par son unité et par son origine.

0°C = 32 °F et 100 °C = 212 °F

Pour convertir de degré celsius en fahrenheit

$$T (^{\circ}\text{F}) = \frac{9}{5} t(^{\circ}\text{C}) + 32 \quad (1.19)$$

### 1.3.4 Humidité

L'humidité est le contenu réel en eau de l'air qui peut être exprimé soit en densité de vapeur (g m) ou en pression de vapeur (kPa). Cependant en pratique, il est plus commode d'exprimer le contenu en eau en humidité relative (HR). L'humidité relative est le rapport du contenu réel en eau de l'air à la quantité maximale d'eau que l'air peut contenir à une température donnée autrement dit l'humidité relative est le rapport de la pression réelle de vapeur à la pression de vapeur saturé [12]

Température (c°)	Humidité Relative				
	100 %	80 %	50 %	20 %	10 %
30	4.24	3.40	2.12	0.85	0.42
20	2.34	1.87	1.17	0.47	0.23
10	1.23	0.98	0.61	0.24	0.12

Tableau 2.1 : la pression de vapeur d'eau en fonction de la température et le degré de saturation de la vapeur [12]



### **1.3.5 Météorologie aéronautique**

Des observations régulières sont faites aux aérodromes et stations supplémentaires. Elles sont transmises toutes les heures sous forme de messages codés appelés METAR qui donnent le vent, la visibilité, la température, l'hygrométrie, la hauteur des nuages, la pression atmosphérique et les phénomènes particuliers. Des METAR sont également émis lorsque les conditions changent au-delà d'une certaine limite, par exemple lorsque le plafond ou la visibilité font passer les conditions de *vol à vue* aux conditions de *vol aux instruments*.

Des prévisions appelées TAF sont émises par les services nationaux de météorologie pour les aéroports. Elles sont émises à toutes les 3 ou 6 heures Ces messages indiquent les mêmes éléments que le METAR mais donnent en plus des indications sur la turbulence et le givrage prévus et sur l'évolution du temps [14].

#### **1.3.5.1 Type de prévision météorologique aéronautique**

En aéronautique on peut utiliser plusieurs instruments pour suivre et étudier le changement climatique :

##### **1.3.5.1.1 Les Stations de Météorologie Aéronautique**

Ces stations sont qualifiées pour :

- effectuer des mesures et des observations météorologiques pour l'aviation
- établir et diffuser des messages d'observation météorologiques destinés aux usagers aéronautiques.
- des renseignements sur le vent, la visibilité, les nuages, la température et la pression
- une prévision d'atterrissage



Figure 1.17 : station météo a transmission radio

#### 1.3.5.1.2 Observations radar

Le radar peut aussi servir à localiser les zones précipitant et à suivre leur déplacement puisque les gouttes d'eau réfléchissent les ondes radar de longueur 10 cm peut provoquer un écho à la réception de radar ce type de radar est mono statique a l'impulsion, ça veut dire il émit une seule impulsion a pour recevoir un l'écho à la réception.



figuer 1.18 : Tour aéronautique de station météorologique

### 1.3.5.1.3 Observations par satellites météorologiques

Ce type de satellite est conçu pour faire des prévisions météorologique à l'aide des ondes émis par le satellite et permettent de nombreuses mesures

- 1- température au sol ou en altitude
- 2- température à la surface des océans ou au sommet des nuages
- 3- vitesse du vent
- 4- degré d'humidité

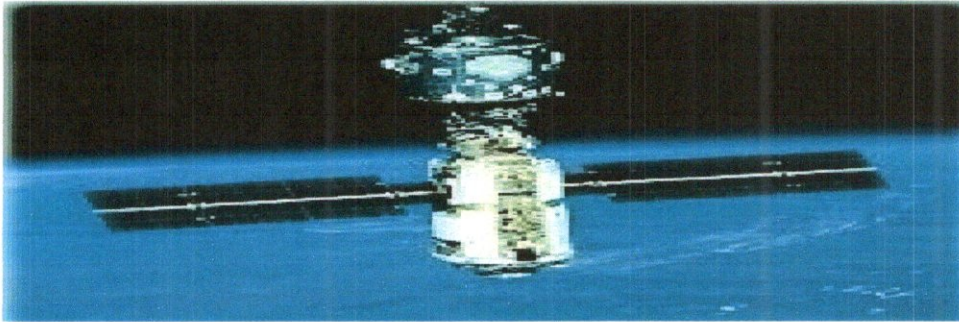


Figure 1.19 : 1<sup>er</sup> satellite météorologique (TIROS 1)

### 1.4 Conclusion

Dans ce chapitre nous avons détaillé les principaux éléments constituant une tour de contrôle à savoir le radar pour l'analyse de l'environnement aérien et la station météo pour la collecte des données météorologique. Dans le chapitre qui suit nous allons détailler l'élément de base qui nous permettra de réaliser notre objectif qui est la réalisation d'un simulateur d'une tour de contrôle ce dernier s'articulera sur une carte programmable qui n'est autre que la carte ARDUINO.

# CHAPITRE 2 : ETUDE DE LA CARTE ARDUINO



## 2.1 Introduction

Les cartes ARDUINO sont conçues pour réaliser des prototypes et des maquettes de cartes électroniques pour l'informatique embarquée, ces cartes permettent un accès simple et peu coûteux à l'informatique embarquée. De plus, elles sont entièrement libres de droit, autant sur l'aspect du code source (Open Source) que sur l'aspect matériel (Open Hardware). Ainsi, il est possible de refaire sa propre carte ARDUINO dans le but de l'améliorer ou d'enlever des fonctionnalités inutiles au projet. Ce dernier n'est fonctionnel que si on l'associe au langage ARDUINO qui se distingue des langages utilisés dans l'industrie de l'informatique embarquée de par sa simplicité. En effet, beaucoup de bibliothèques et de fonctionnalités de base occulte certains aspects de la programmation de logiciel embarquée afin de gagner en simplicité. Cela en fait un langage parfait pour réaliser des prototypes.

Les possibilités des cartes ARDUINO sont énormes, un grand nombre d'applications ont déjà été réalisées et testées par bon nombre de chercheurs. On retrouve par exemple diverse forme de robot contrôleur de moteur pas à pas par joystick avec ARDUINO etc..... [15].

## 2.2 Définition de l'ARDUINO :

Elle est considérée comme un grand équipement électronique pour les amateurs ainsi que les professionnels, l'ARDUINO est un puissant ordinateur à carte unique avec une utilisation facile à adapter. C'est une open source, ce qui signifie que vous devez payer son matériel et le logiciel requis est gratuit. Son logiciel est basé sur la puce ATMEGA [16].

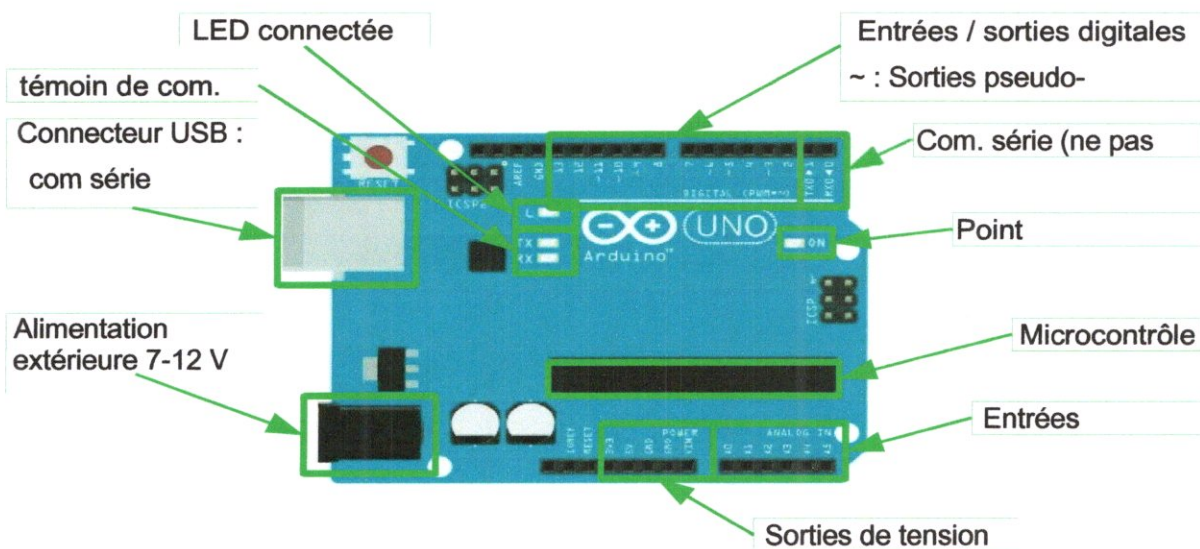


Figure 2.1 : différents composants de la carte ARDUINO

### 2.2.1 LE MICROCONTROLEUR ARDUINO

Un microcontrôleur est un système qui ressemble à un ordinateur : il a une mémoire, un processeur, des interfaces avec le monde extérieur. Les microcontrôleurs ont des performances réduites, mais sont de faible taille et consomment peu d'énergie, les rendant indispensables dans toute solution d'électronique embarquée (voiture, porte de garage, robots, ...). La carte ARDUINO n'est pas le microcontrôleur le plus puissant, mais son architecture a été publiée en open-source, et toute sa philosophie s'appuie sur le monde du libre, au sens large.

La carte ARDUINO se relie à un ordinateur par un câble USB. Ce câble permet à la fois l'alimentation de la carte et la communication série avec elle [17].

### 2.2.2 Les composants d'un microcontrôleur

Un microcontrôleur le plus simple qu'il soit, possède au minimum les éléments suivants :

- 1- Une unité centrale qui est le cœur de système, également appelé CPU (Central PROCESSING Unit) dans cette unité centrale nous retrouverons plusieurs éléments telle que l'unité arithmétique et logique (UAL)
- 2- Une mémoire contenant le programme à exécuter par le microcontrôleur, généralement appelée mémoire morte ou ROM, mémoire à lecture seule. Cette mémoire a la particularité de sauvegarder en permanence les informations qu'elle contient, même en absence de tension (ce qui est primordiale, sino il faudrait reprogrammer le microcontrôleur à chaque remise sous tension !).
- 3- Une mémoire vive (RAM), cette mémoire permet de sauvegarder temporairement des informations. Il est à noter que le contenu d'une RAM n'est sauvegardé que pendant la phase d'alimentation du circuit. Le microcontrôleur pourra utiliser cette mémoire pour stocker des variables temporaires ou faire des calculs intermédiaires.
- 4- Un port d'entrées/sorties permettant au microcontrôleur de dialoguer avec l'extérieur pour par exemple prendre l'état d'un capteur, d'un interrupteur ou bien pour allumer une LED ou piloter un relais (via un transistor bien sûr).
- 5- Une base de temps : elle peut être assurée par un cristal de quartz externe, un oscillateur programmable, un résonateur céramique, ou d'une base de temps interne [19]

Des bus internes permettent la communication entre les différents éléments intégrés au microcontrôleur.

Et pour certains microcontrôleurs les plus sophistiqués peuvent contenir des éléments supplémentaires, tels que :

Convertisseur analogique-numérique (CAN): un convertisseur analogique-numérique a une tension d'entrée analogique et produit une sortie binaire représentative de sa valeur. Dans un CAN linéaire, la plage de valeurs d'entrée analogiques est divisé en intervalles spatiaux de façon égale; dont chacune est affectée a un mot de code binaire. [19]

Un convertisseur numérique-analogique (CNA): un CNA accepte un mot d'entrée de n bits  $b_1, b_2, b_3, \dots, b_n$  en binaire et produire un signal proportionnel analogique à elles. La figure ci-dessous montre son symbole [20].

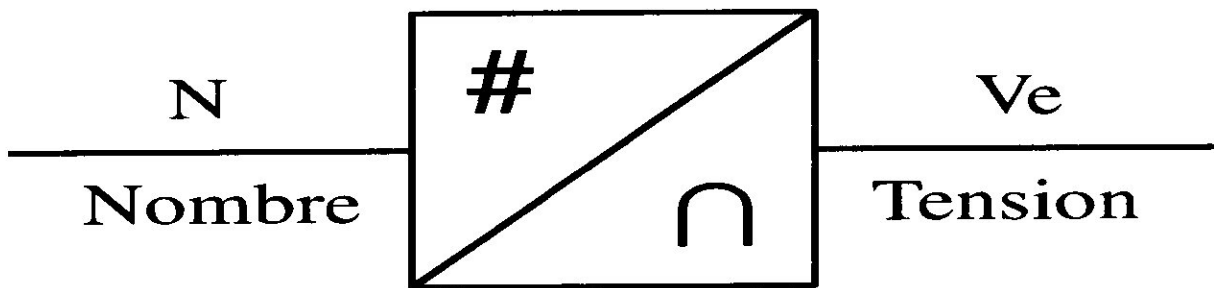


Figure 2.2 : Symbole d'un convertisseur numérique/analogique

### 2.3 Matériel ARDUINO

La puissance de l'ARDUINO n'est pas sa capacité à croquer le code, mais plutôt sa capacité à interagir avec le monde extérieur grâce à ses broches d'entrée-sortie (E / S). L'ARDUINO a 14 broches d'E / S numériques étiquetées de 0 à 13 qui peuvent être utilisées pour allumer et éteindre les moteurs et les lumières et lire l'état des commutateurs.

Chaque broche numérique peut couler ou produire environ 40 mA de courant. Cela est plus que suffisant pour l'interfaçage avec la plupart des appareils, mais cela signifie que des circuits d'interface sont nécessaires pour commander les appareils autres que les simples DEL. En d'autres termes, vous ne pouvez pas exécuter un moteur directement en utilisant le courant disponible à partir d'une broche ARDUINO, mais vous devez plutôt avoir un lecteur de broche, un circuit d'interface qui à son tour entraîne le moteur. Une section ultérieure de ce document montre comment interfacier avec un petit mo



## CHAPITRE 2 : Etude de la carte ARDUINO

Pour interagir avec le monde extérieur, le programme règle les broches numériques à une valeur haute ou basse en utilisant les instructions du code C, ce qui correspond à +5 V ou 0 V sur la broche. La broche est connectée à l'électronique d'interface externe puis à l'appareil en cours de mise en marche et à l'arrêt. La séquence de l'évènement [21].

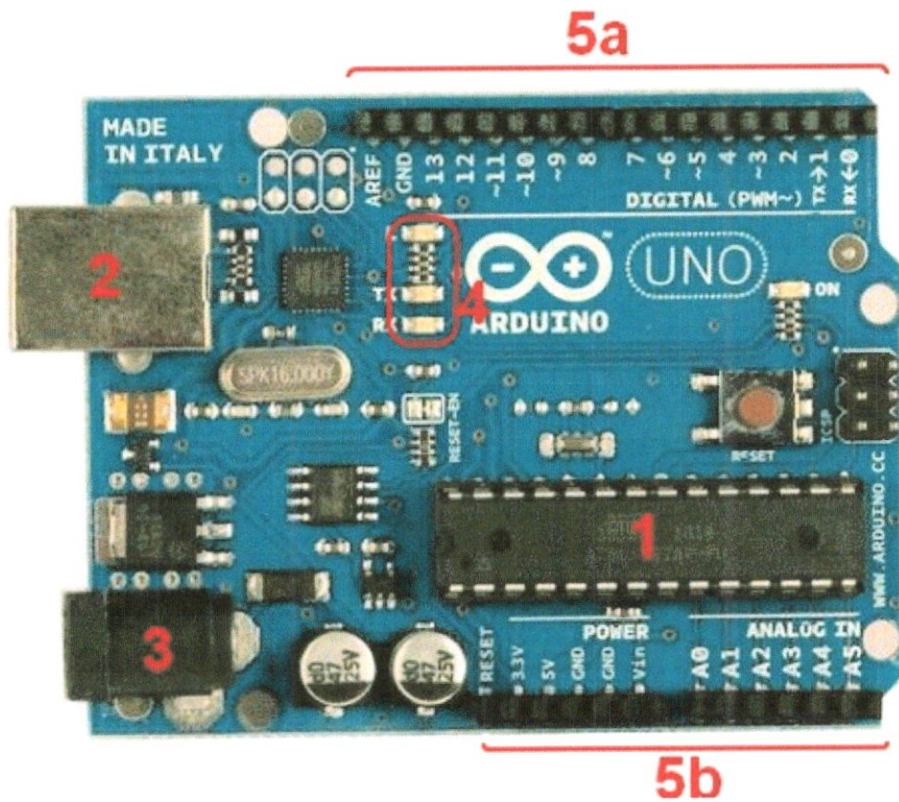


Figure 2.3 : Présentation de la carte ARDUINO

### 2.3.1 Alimentation

Pour fonctionner, la carte a besoin d'une alimentation. Le microcontrôleur fonctionnant sous 5V, la carte peut être alimentée en 5V par le port USB (en 2) ou bien par une alimentation externe (en 3) qui est comprise entre 7V et 12V. Cette tension doit être continue et peut par exemple être fournie par une pile 9V. Un régulateur se charge ensuite de réduire la tension à 5V pour le bon fonctionnement de la carte. Pas de danger de tout griller donc! Veuillez seulement à respecter l'intervalle de 7V à 15V (même si le régulateur peut supporter plus, pas la peine de le retrancher dans ses limites) [22].

### 2.3.2 Visualisation

Les trois "points blancs" entourés en rouge (4) sont en fait des LED dont la taille est de l'ordre du millimètre. Ces LED servent à deux choses :

- Celle tout en haut du cadre : elle est connectée à une broche du microcontrôleur et va servir pour tester le matériel.

Notation : Quand on branche la carte au PC, elle clignote quelques secondes.

- Les deux LED du bas du cadre : servent à visualiser l'activité sur la voie série (une pour l'émission et l'autre pour la réception). Le téléchargement du programme dans le microcontrôleur se faisant par cette voie, on peut les voir clignoter lors du chargement [23].

### 2.3.3 La connectique

La carte ARDUINO ne possédant pas de composants qui peuvent être utilisés pour un programme, mis à part la LED connectée à la broche 13 du microcontrôleur, il est nécessaire de les rajouter. Mais pour ce faire, il faut les connecter à la carte. C'est là qu'intervient la connectique de la carte (en 5a et 5b) [23].



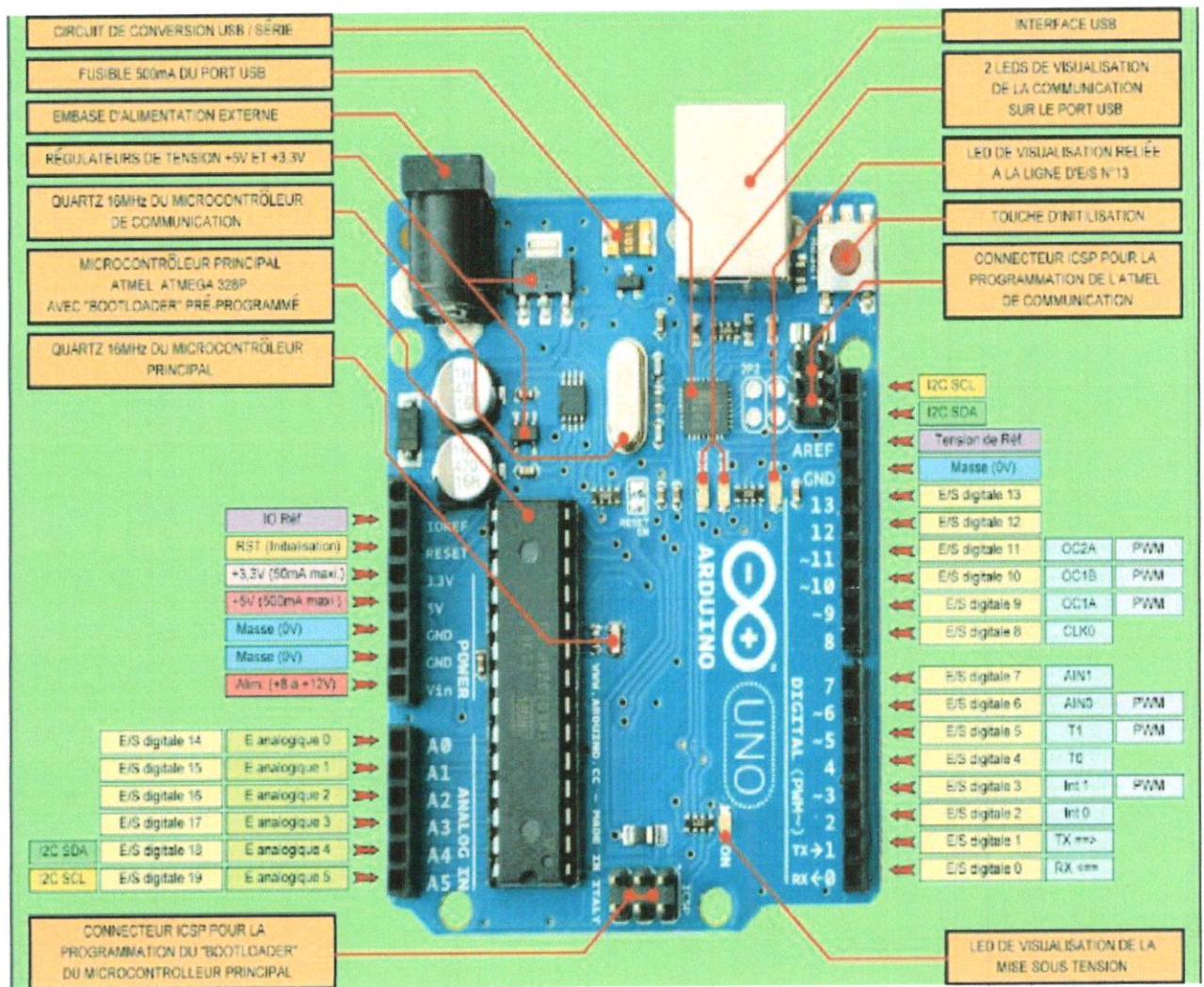


Figure 2.4 : organisation physique de la carte l'ARDUINO-UNO

La figure 2.5 montre les différents composants et l'organisation du module ARDUINO-UNO avec la fonction de chaque broche. Ce module comporte trois types de mémoire (flash, ram et EEPROM) et une vitesse d'horloge de 16 MHz, nous disposons de 20 lignes d'entrée/sortie dont 6 dédiées au convertisseur analogique/numérique sur 10 bits.

## 2.4 Type de carte ARDUINO

On distingue plusieurs types de cartes ARDUINO à cause de la qualité de la carte, sa tension d'entrée ou sa mémoire.



### 2.4.1 La carte ARDUINO UNO

C'est la carte idéale pour découvrir l'environnement ARDUINO. Elle permet à tout débutant de se lancer dans tous ses premiers petits projets. Comme c'est la carte la plus utilisée.

Sa simplicité devient par contre un handicap lorsqu'il s'agit de multiplier les périphériques, de manipuler des algorithmes lourds ou d'interagir avec les OS Android pour les quels d'autres cartes ARDUINO sont plus adaptées.

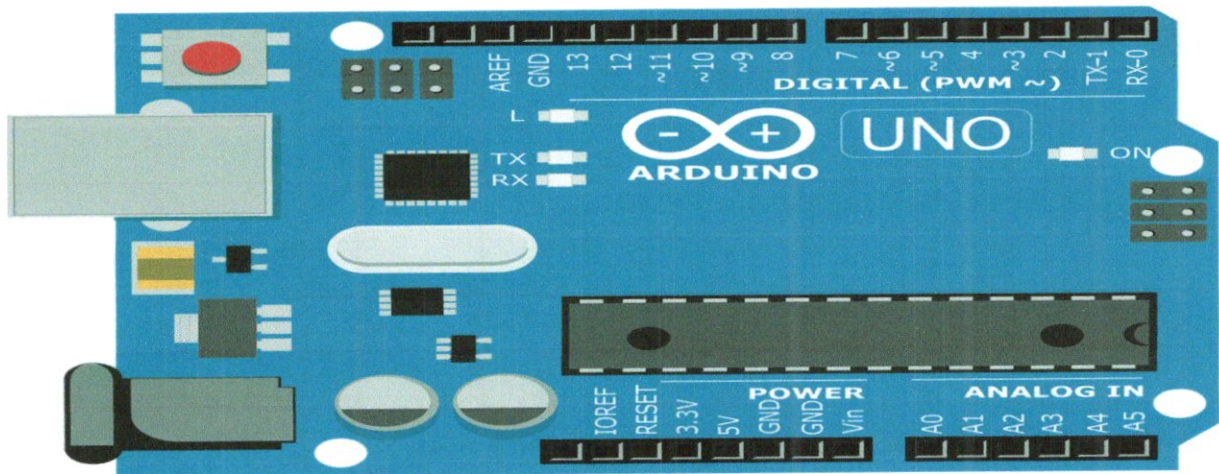


Figure 2.5 : Carte ARDUINO UNO

### 2.4.2 La carte ARDUINO LIOARDO

C'est la carte qui est prévue pour succéder à la carte ARDUINO UNO en présentant des caractéristiques équivalentes mais une ergonomie revue et une stabilité plus éprouvée. Sa diffusion moins importante limite le support utilisateur.

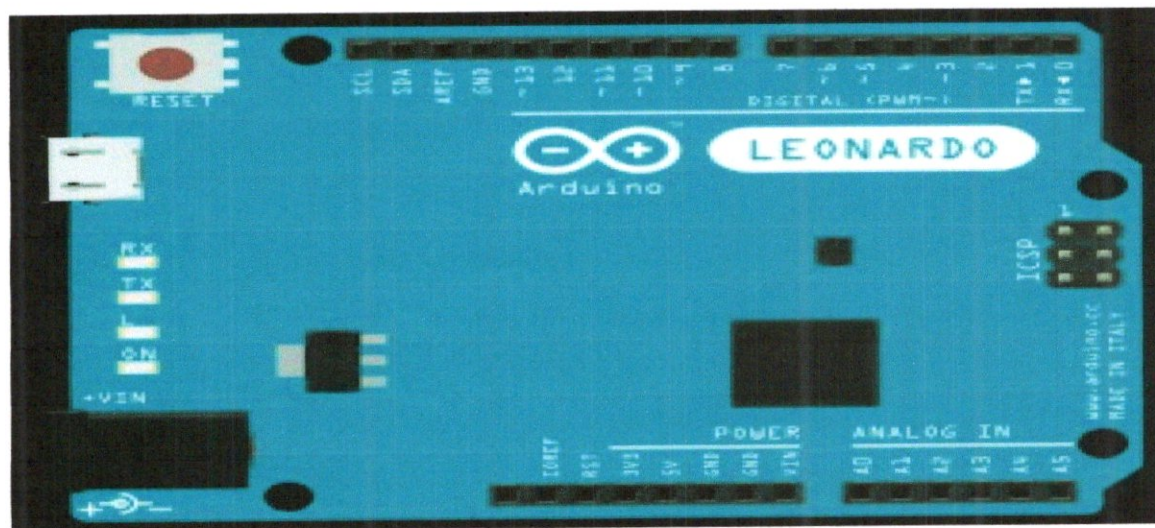


Figure 2.6 : carte ARDUINO LIONARDO



### 2.4.3 La carte ARDUINO Méga

La carte ARDUINO Méga est la carte la plus diffusée après la carte ARDUINO UNO Elle offre un nombre d'entrées/sorties beaucoup plus important (54 contre 14), un processeur plus puissant doté d'une mémoire plus vaste qui permet d'exploiter des algorithmes plus complexes

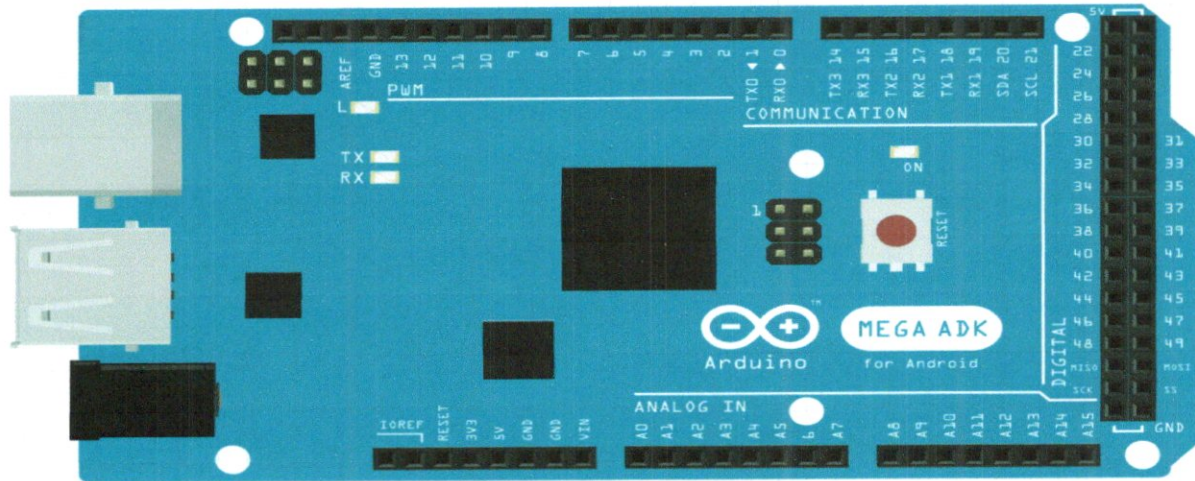


Figure 2.7 : carte ARDUINO Méga

### 2.4.4 La carte ARDUINO Méga ADK

La carte Arduino méga ADK offre les mêmes caractéristiques techniques que la carte Arduino méga mais son port USB permet de la connecter avec un environnement Android ouvrant de nouvelles perspectives d'interaction avec le monde des smartphones et des capteurs dont ils sont dotés. Sa mise en œuvre nécessite par contre de solides connaissances en Java et la capacité à développer ses propres applications.

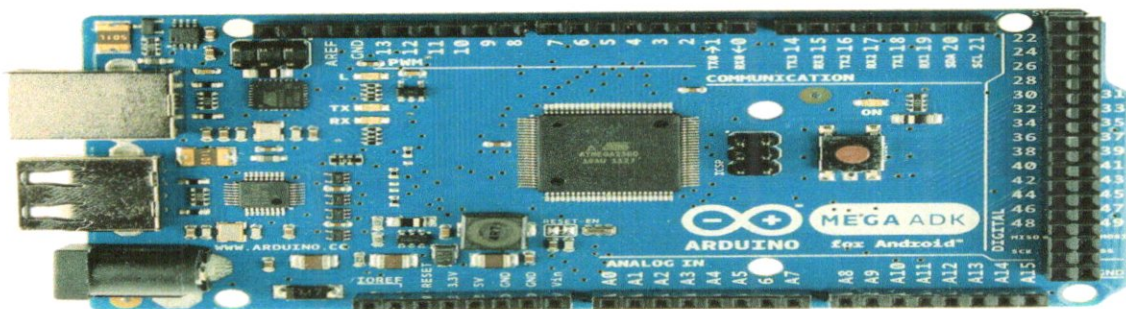


Figure 2.8 : carte ARDUINO Méga ADK



## CHAPITRE 2 : Etude de la carte ARDUINO

### 2.4.5 La carte ARDUINO Due

La carte ARDUINO Due est une évolution de la carte ARDUINO Méga et offre des performances réputées 3 fois supérieures. Elle permet de manipuler rapidement des algorithmes lourds particulièrement utiles dans le monde de la robotique par exemple.

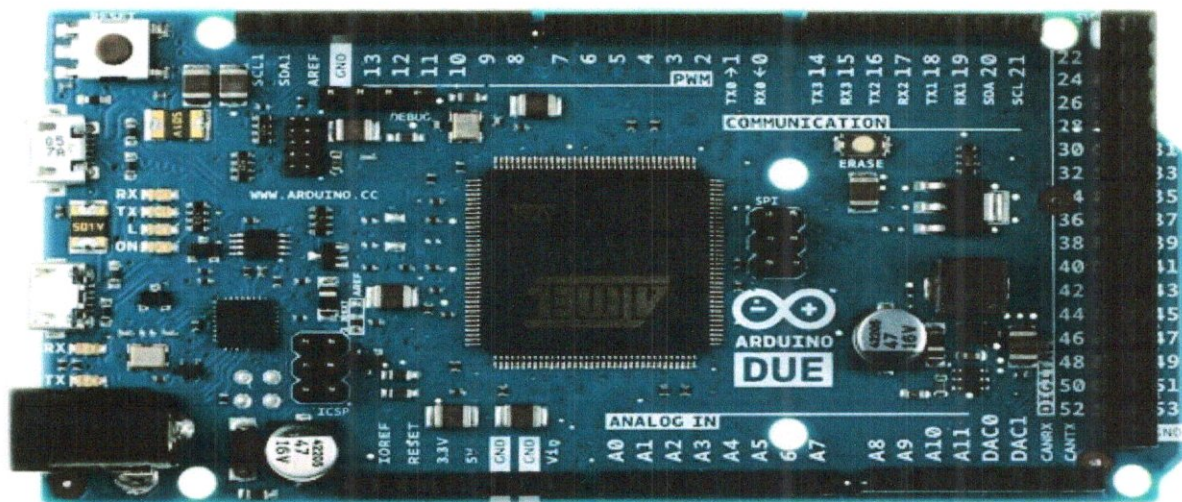


Figure 2.9 : carte ARDUINO DUE

### 2.4.6 La carte ARDUINO Nano

La carte ARDUINO nano n'est ni plus ni moins qu'une carte ARDUINO UNO miniaturisée. Sa taille et son poids réduits la destinent à une utilisation dans des espaces réduits (en textile par exemple) ou dans des applications de robotique ou de modélisme pour lesquels le poids et la taille sont des facteurs déterminant.

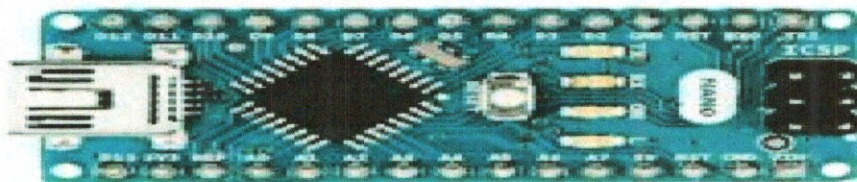


Figure 2.10 : carte ARDUINO nano

### 2.4.7 La carte ARDUINO Mini Pro

La carte ARDUINO Mini Pro est une carte ARDUINO simplifiée à l'extrême permettant néanmoins de piloter de petits projets ou certains éléments d'un projet. Attention, cette carte n'intègre pas de port USB ce qui rends sa connectivité délicate.

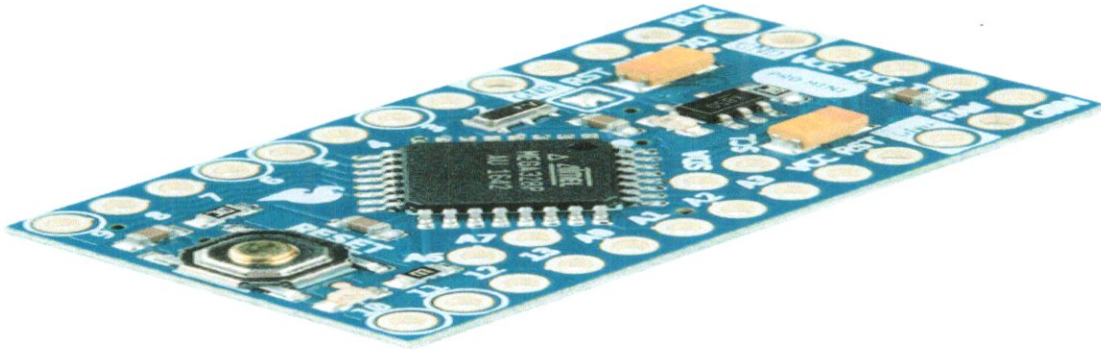


Figure 2.11 : carte ARDUINO mini pro

### 2.4.8 La carte ARDUINO\_Yun

La carte ARDUINO Yun, récemment proposée par ARDUINO, est conçue pour contrer les avantages de la carte RASPBERRY. Elle est un dérivé de la carte Leonardo et a pour objectif de combiner la puissance de Linux avec la facilité d'utilisation d'une carte ARDUINO.

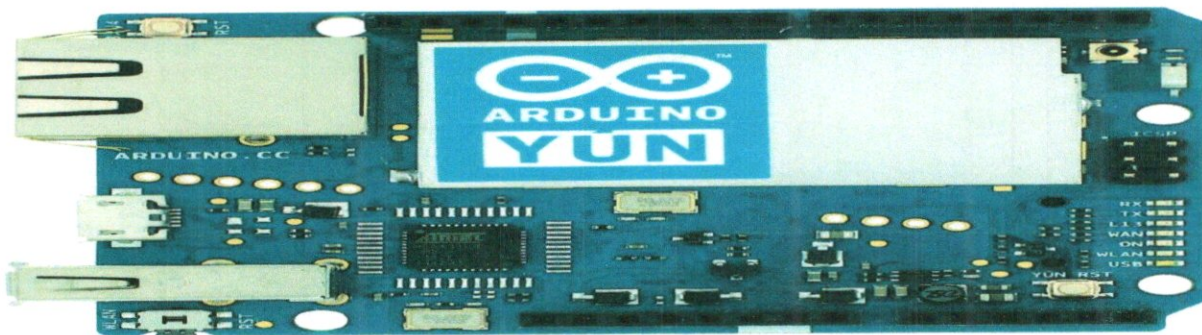


Figure 2.12 : carte ARDUINO Yun



CHAPITRE 2 : Etude de la carte ARDUINO

Cartes Arduino Caractéristiques	UNO R3 (classique & CMS)	UNO R3 Ethernet (classique & POE)	Leonardo	Mega 2560	Mega ADK	DUE	Esplora	Mini	Nano	Yun (classique & POE)	Zero PRO
Microcontrôleur	ATmega328P	ATmega328P	ATmega32u4	ATmega2560	ATmega2560	AT91SAM3X8E	ATmega32u4	ATmega328P	ATmega328P	ATmega32u4	ATSAMD21G18
Cadencement Horloge	16 MHz	16 MHz	16 MHz	16 MHz	16 MHz	84 MHz	16 MHz	16 MHz	16 MHz	16 MHz	48 MHz
Tension d'entrée	7 - 12V	7 - 12V	7 - 12V	7 - 12V	7 - 12V	7 - 12V	7 - 12V	7 - 9V	7 - 9V	5V	5V
Tension de fonctionnement	5V	5V	5V	5V	5V	3,3V	5V	5V	5V	5V	3,3V
Entrée/Sortie Numérique	14/6	14/4	20/7	54/15	54/15	54/12		14/6	14/6	20/7	14/12
Entrée-Sortie (PWM) Analogique	6/0	6/0	12/0	16/0	16/0	12/2 (DAC)		8/0	8/0	12/0	6/1 (DAC)
Mémoire vive (Flash)	32 Ko	32 Ko	32 Ko	256 Ko	256 Ko	512 Ko	32 Ko	32 Ko	32 Ko	32 Ko	256 Ko
Mémoire vive (SRAM)	2 Ko	2 Ko	2,5 Ko	8 Ko	8 Ko	96 Ko	2,5 Ko	2 Ko	2 Ko	2,5 Ko	32 Ko
Mémoire morte (EEPROM)	1 Ko	1 Ko	1 Ko	4 Ko	4 Ko		1 Ko	1 Ko	1 Ko	1 Ko	16 Ko
Interface USB	USB-B mâle	USB-B mâle	Micro-USB	USB-B mâle	USB-B mâle & USB-A pour Android	2 ports micro- USB (Native et programming)	Micro-USB		Mini-USB	Micro-USB	2 ports micro- USB (Native et programming)
Port UART	1	1	1	4	4	4			1	1	2
Carte SD											
Ethernet											
Wi-Fi											
Dimensions	68x53mm	68x53mm	68x53mm	101x53mm	101x53mm	101x53mm	165x60mm	30x18mm	45x18mm	68x53mm	68x53mm

Tableau 2.1 : Tableau comparatif entre des différentes cartes ARDUINO.

Tout d'abord, si on cherche à commencer à programmer à moindre coût, alors l'UNO est idéale. Si on préfère brancher une multitude de capteurs, utiliser plusieurs périphériques en série, alors la Méga est parfaite pour ça, si on veut les mêmes propriétés que la Méga mais avec une puissance de calcul supérieure alors c'est plutôt la DUE qu'il nous faut ! Si on cherche à miniaturiser notre système, on doit choisir la carte Mini ! Enfin, si on cherche un véritable ordinateur embarqué la Yun est le choix le plus évident.

## 2.5 Étude de la partie logicielle

### 2.5.1 Le langage de programmation

Un langage de programmation est un langage permettant à un être humain d'écrire un ensemble d'instructions (code source) qui seront directement converties en langage machine grâce à un compilateur (c'est la compilation). L'exécution d'un programme ARDUINO s'effectue de manière séquentielle, c'est-à-dire que les instructions sont exécutées les unes à la suite des autres.

### 2.5.2 L'interface du logiciel

L'interface du logiciel ARDUINO se présente de la façon suivante :

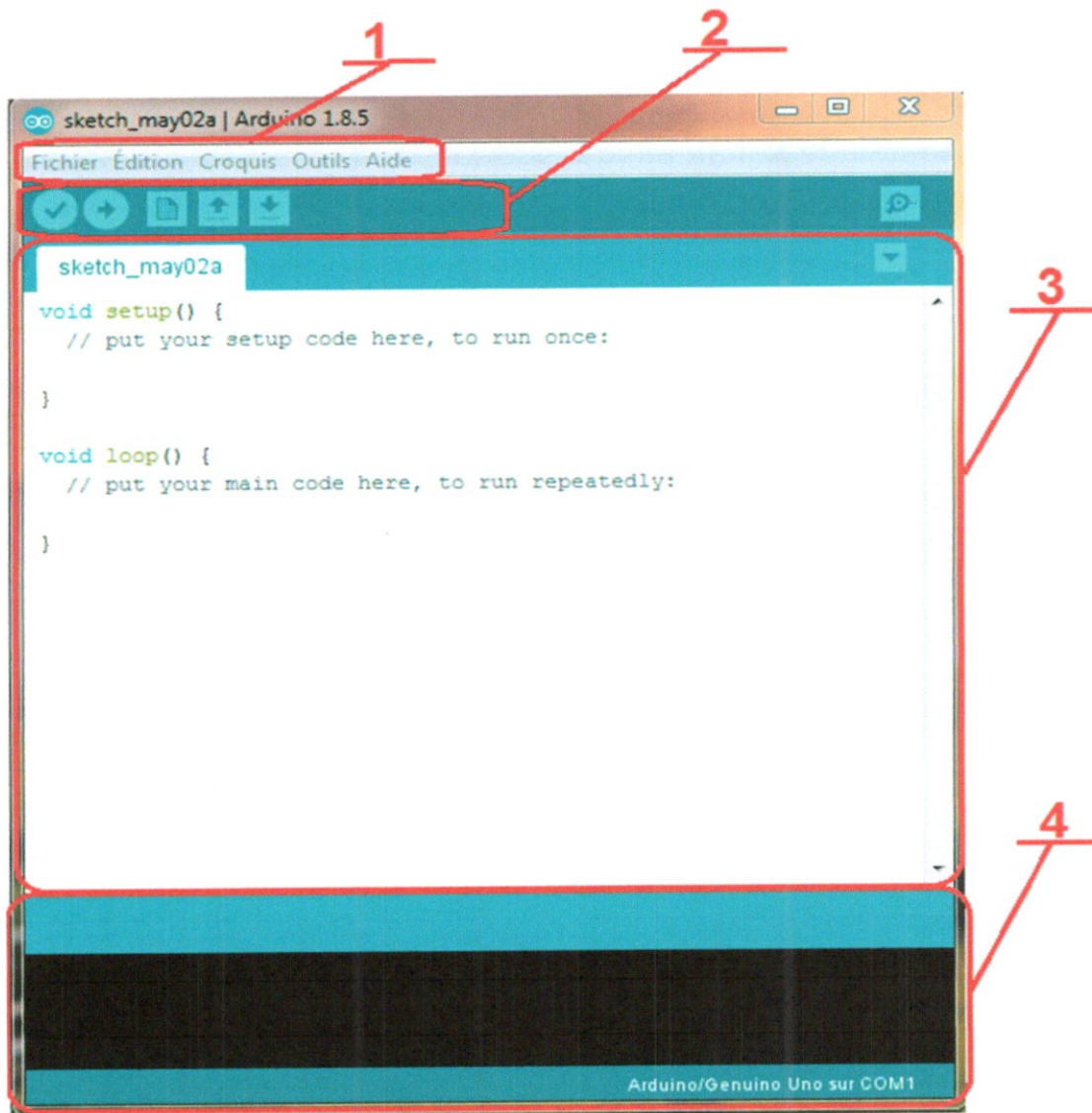


Figure 2.13 : interface de logiciel ARDUINO



1. options de configuration du logical
2. boutons pour la programmation des cartes
3. programme à créer
4. débogueur (affichage des erreurs de programmation)

### 2.5.3 Les boutons

L'interface ARDUINO contient plusieurs boutons :

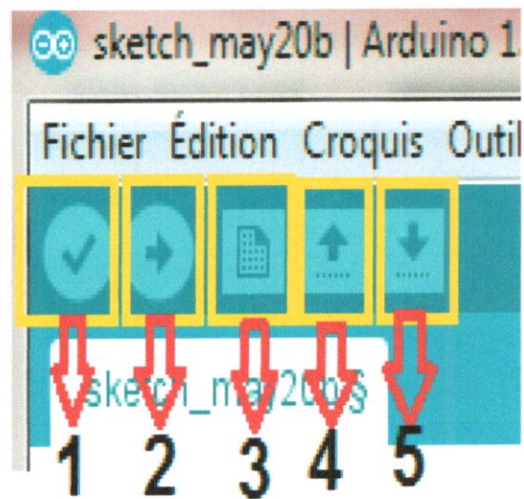


Figure 3.12 : les boutons dans un logiciel ARDUINO

1. Permet de vérifier le programme, il actionne un module qui cherche les erreurs dans le programme
2. Compiler et envoyer le programme vers la carte
3. Créer un nouveau fichier
4. Charger un programme existant
5. Sauvegarder le programme en cours

### 2.5.4 La structure d'un programme

Un programme ARDUINO comporte trois parties

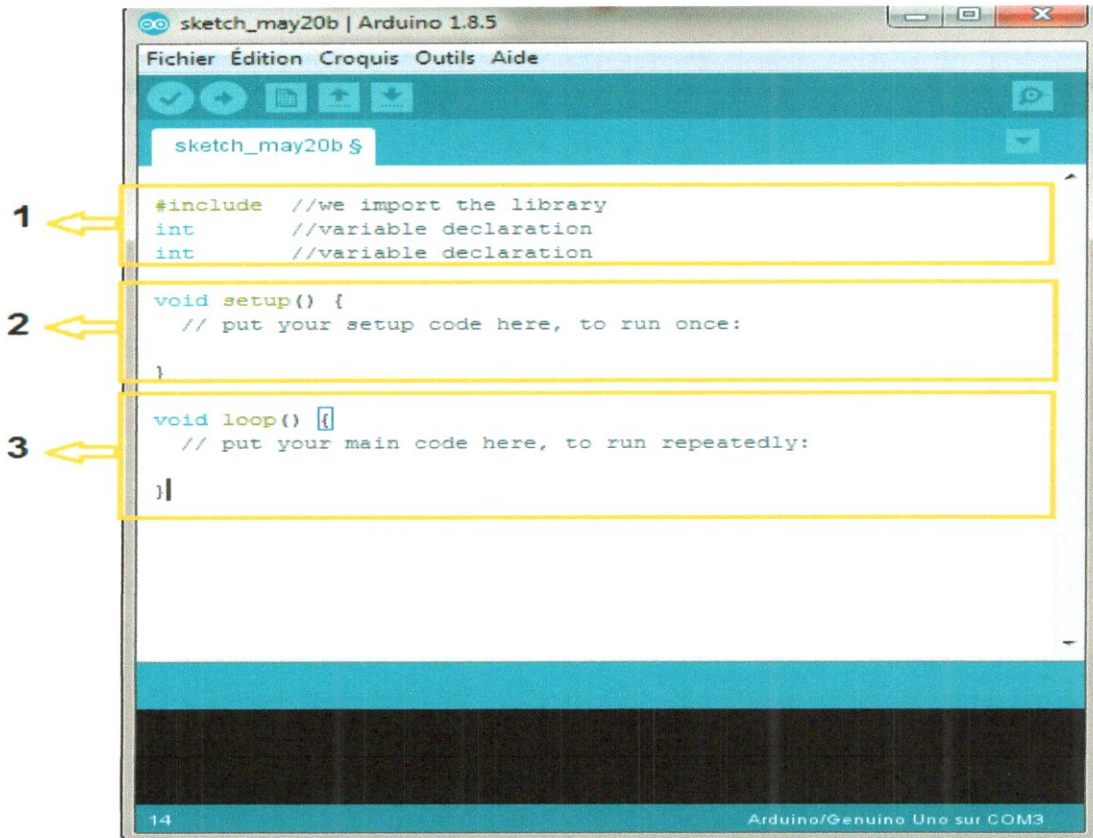


Figure 2.15 : structure de programme ARDUINO

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

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

### 2.6 Le logiciel PROCESSING :

PROCESSING est un langage de programmation et un environnement open source pour les personnes désirant créer des images, des animations et des interfaces. Originellement conçu comme software de prototypage et pour enseigner les bases de la programmation dans un contexte visuel, PROCESSING s'est également étendu en un outil pour générer des travaux finis.

Aujourd'hui, de nombreux étudiants, artistes, designers, chercheurs, et passionnés utilisent PROCESSING pour apprendre, prototyper, et produire.

### 2.6.1 L'interface de PROCESSING

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

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

1. Barre de menu
2. Barre d'actions
3. Barre d'onglets
4. Zone d'édition (pour y saisir votre programme)
5. Fenêtre de visualisation (espace de dessin)
6. Console (destinée aux tests et messages d'erreur)

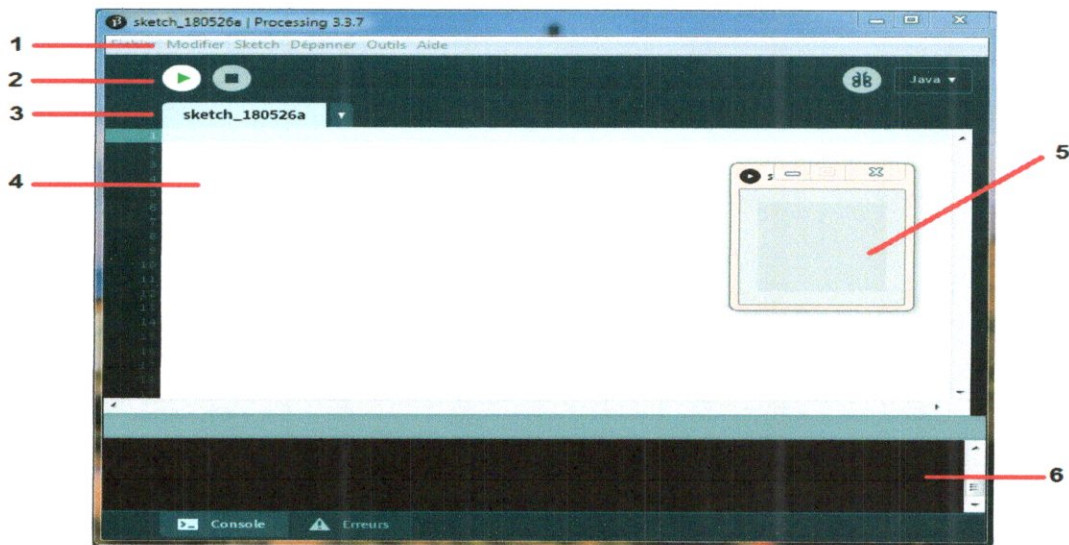


Figure 2.16: interface de logiciel PROCESSING



### 2.6.2 Communication entre la carte ARDUINO et le logiciel PROCESSING

L'IDE de traitement de PROCESSING communique avec l'IDE a ARDUINO avec une communication en série, l'IDE de PROCESSING est similaire a ARDUINO il en les mêmes fonctions de configuration et fonction de dessin de cette manière on peut échanger d'information entre les l'IDE de PORCESSING et l'IDE de l'ARDUINO

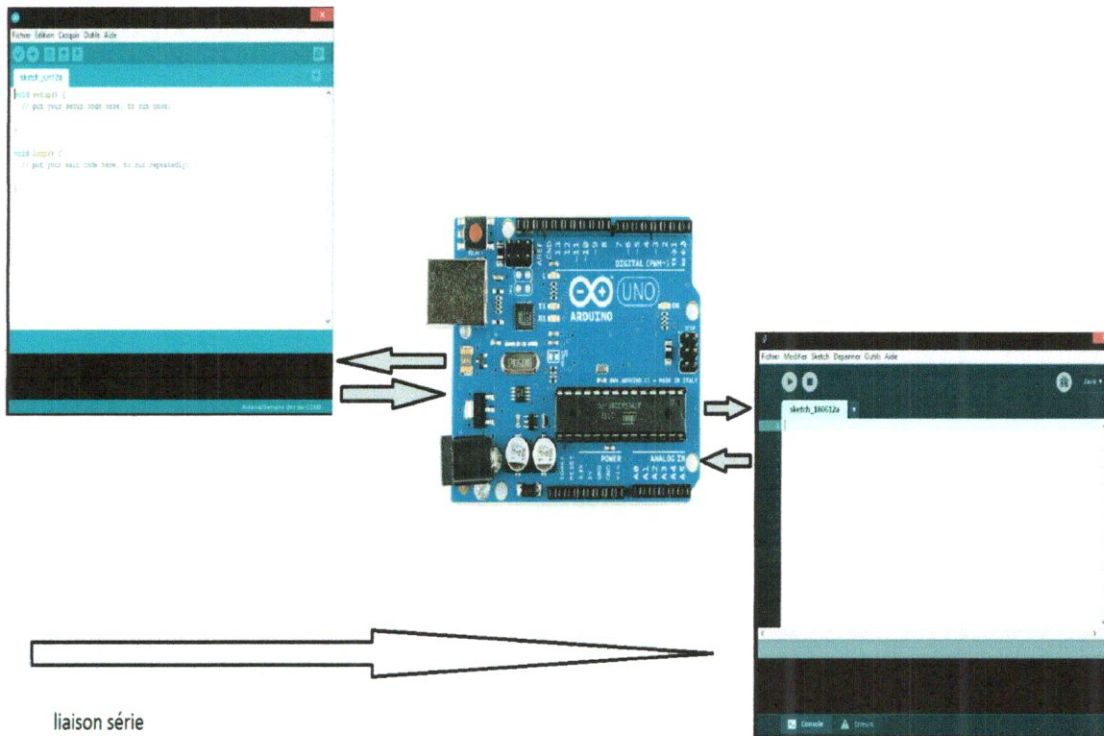


Figure 2.17: Communication entre la carte ARDUINO et PROCESSING



### **2.7 Conclusion**

Dans ce chapitre nous avons parlé des principaux éléments constituant la carte de prototypage ARDUINO à savoir les différents composant de la carte, la conversion analogique numérique, la transmission de données, les types de carte existantes , ainsi que la partie logiciel qui permettra de charger l'application sur la carte associe au logiciel du balayage afin de simuler l'environnement. Les descriptions faite au chapitre I et chapitre II nous permettra de concevoir notre simulateur de Tour de contrôle qui sera l'objet du chapitre qui suit

# Chapitre 3 : réalisation de projet

### 3.1 Introduction

Après avoir décrit et détaillé les éléments constituant une tour de contrôle, nous passerons dans ce chapitre à la réalisation de notre simulateur en présentant en premier lieu le synoptique nécessaire et en second lieu en détaillant les principaux éléments constituant notre synoptique.

### 3.2 Description de projet

Ce projet a pour but de réaliser un radar de localisation des cibles fixe ou en mouvement et une station météorologique, en principe le radar utilise des ondes radio pour la détection d'une cible la surveillance aérienne l'information météorologique la circulation aérienne ou et d'approche, dans notre projet nous allons utiliser les ondes ultrason pour la localisation des cibles et des capteurs météo.

### 3.3 Schéma synoptique

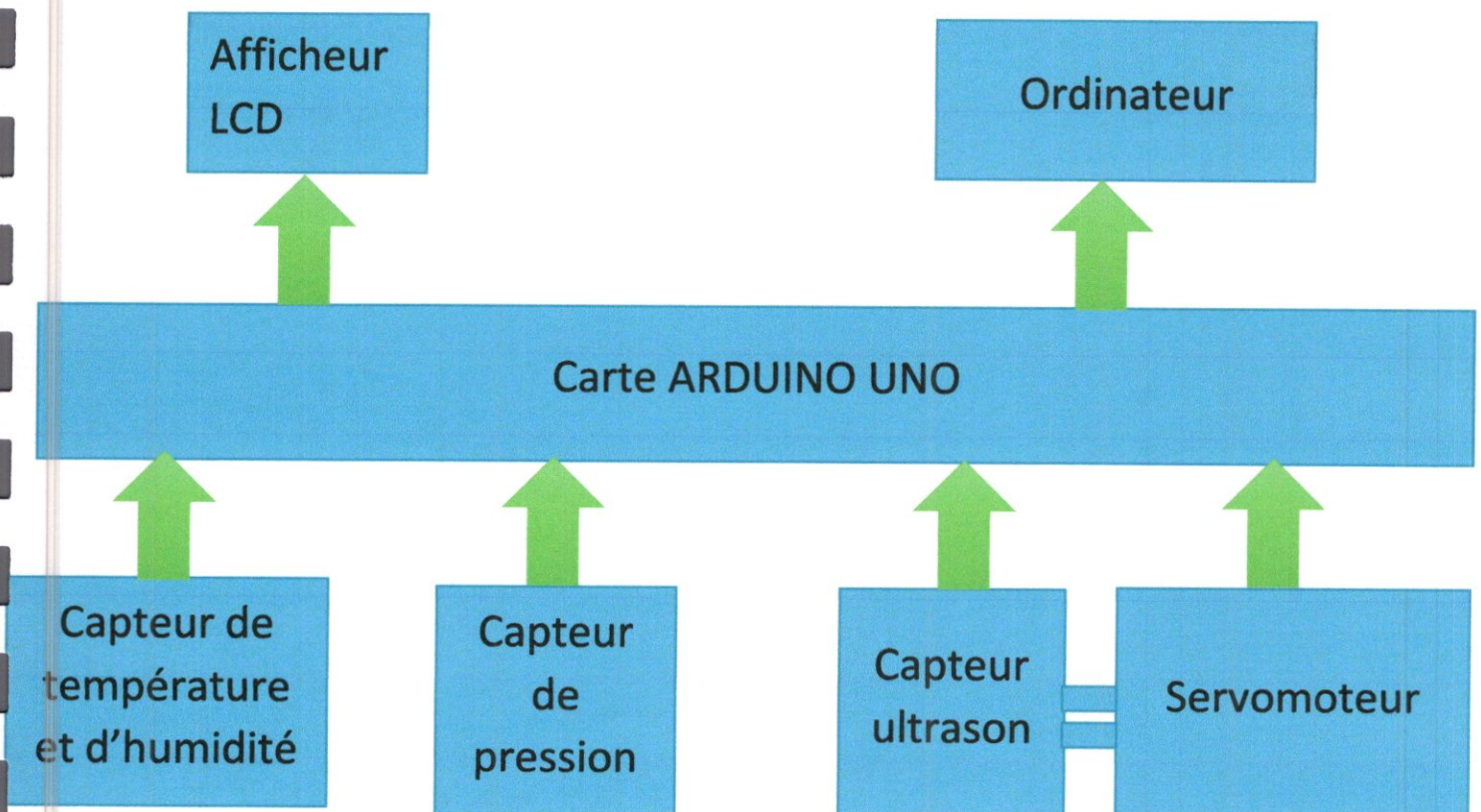


Figure 3.1 : schéma synoptique de la réalisation

### 3.4 Description du matériel

Pour compléter notre réalisation on a besoin de

#### 3.4.1 carte ARDUINO UNO

cette dernière représente le cœur actif de notre projet elle servira comme carte d'acquisition de commande et de transmission

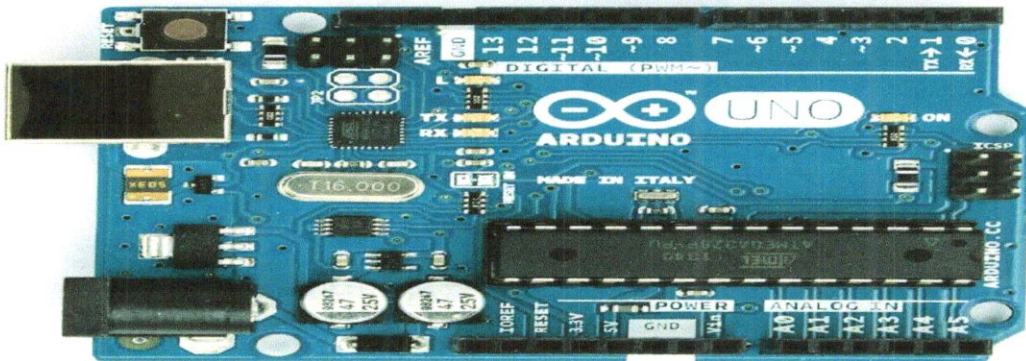


Figure 3.2 : carte ARDUINO UNO

#### 3.4.2 Servomoteur

Les servomoteurs sont des actionneurs. Très utilisés en modélisme et dans l'industrie, ils ont comme caractéristique principale leur « couple », c'est-à-dire la force de rotation qu'ils peuvent exercer. Plus un servomoteur aura de couple et plus il pourra actionner des « membres » lourds comme déplacer un bras qui porte une charge.

Pour la robotique de loisirs, les servomoteurs ont en général peu de couple et sont de taille réduite, bien adaptée à un encombrement minimal et à une énergie disponible limitée.

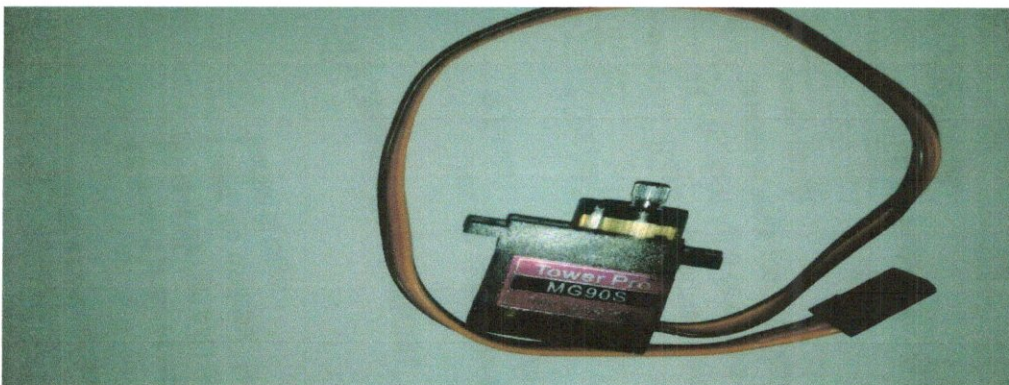


Figure 3.3: servomoteur

Les servomoteurs sont pilotés par un fil de commande et alimentés par deux autres fils. Habituellement, ces 3 fils sont rassemblés dans une prise au format standard



### CHAPITRE 3 : Réalisation de projet

Un fil rouge est relié à l'alimentation positive (+5 ou +6 V selon le servo), le fil noir est relié à la masse (GND) et le fil jaune est utilisé pour la commande.

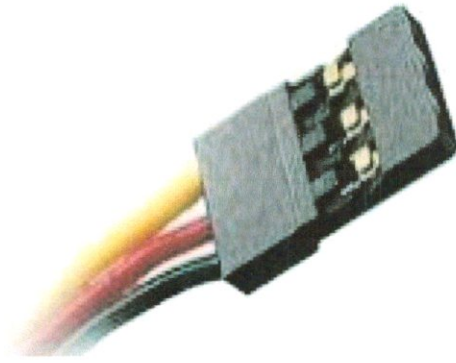


Figure 3.4 : fils de connexion d'servomoteur

#### 3.4.3 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.

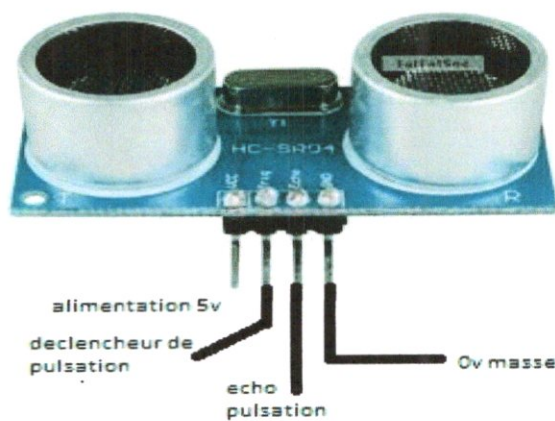


Figure 3.5 : Capteur Sonar à Ultrasons HC-SR04

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 [23].

Ce capteur a une distance de mesure 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 possède un émetteur ultrason et un récepteur ultrason.

### 3.4.3.1 Caractéristiques du capteur ultrasons HC-SR04

Les caractéristiques en détail du Capteur sonar à Ultrasons HC-SR04 sont présentées dans le tableau ci-dessous [29]

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

**Tableau 3.1** : Spécifications des capteurs sonar à ultrason HC-SR04.

### 3.4.3.2 Mise en œuvre du module HC-SR04

Pour utiliser le module HC-SR04, notre programme devra suivre le séquentiel suivant

- Créer une impulsion de 20  $\mu$ s (>10  $\mu$ s) sur la patte Trig
- Mesurer la durée  $\Delta T$  de l'impulsion Echo en  $\mu$ s
- Calculer la distance d en cm correspondante
- Afficher sur le moniteur série la durée  $\Delta T$  et la distance d
- Attendre 200 ms avant de réaliser une nouvelle mesure

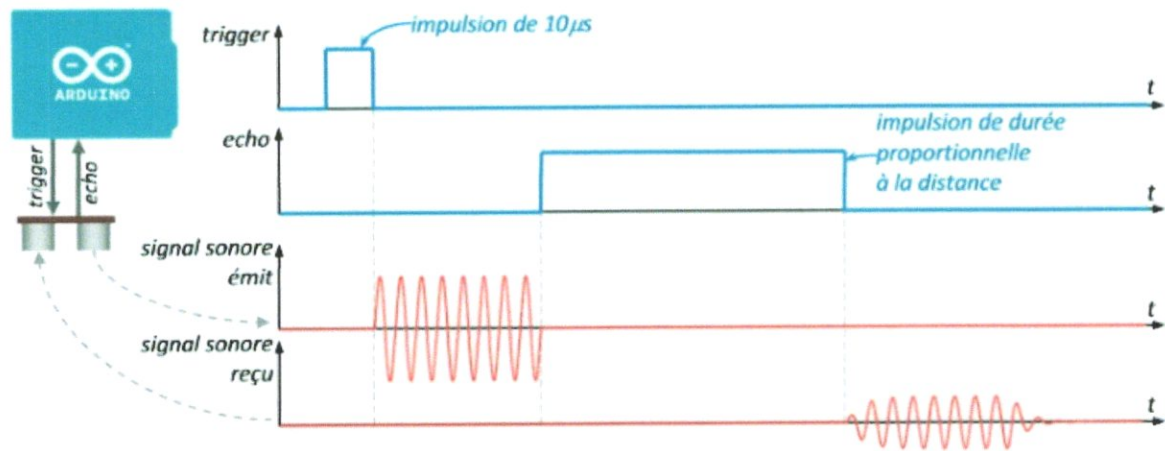


Figure 3.6 : Signal d'entrée et sortie du capteur HC-SR04

### 3.4.4 Capteur de température et d'humidité DHT11

Le capteur DHT11 permet de relever la valeur analogique de la température et de l'humidité ambiante la précision est correcte sans être excellente, pour une meilleur précision, on préférera la sonde DHT22 (précise au demi degré près et a 2 % d'humidité).

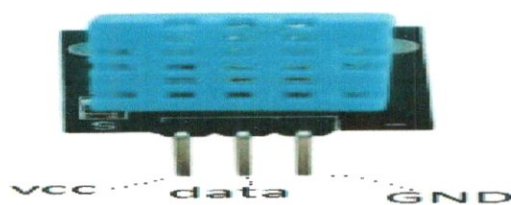


Figure 3.7 : capteur de température et d'humidité DHT11



### 3.4.5 Capteur de pression

La pression constitue une variable essentielle pour l'étude métrologique d'un milieu environnant qui peut être soit un gaz soit un fluide.

La mesure de cette variable est réalisée à l'aide d'un capteur de pression, dispositifs capable d'associer à la grandeur mesurée un signal électrique reconnaissable appelé "réponse". Le capteur de pression comme étant un système constitué de deux parties de traitement de l'information par l'intermédiaire d'un circuit électronique que l'on peut appeler "Circuit électronique de traitement" ou encore "Circuit convertisseur". La partie détection est constituée d'un "corps d'épreuve" et d'un "transducteur" qui transforme la déformation de ce corps d'épreuve en une grandeur physique, ensuite à une grandeur électrique.

#### 3.5.5.1 Capteur de pression mpx4250dp

Est le type de capteur différentiel sa pression minimal est 0 kPa sa pression maximal 250 kPa avec une alimentation de 4.85 v à 5.35 v

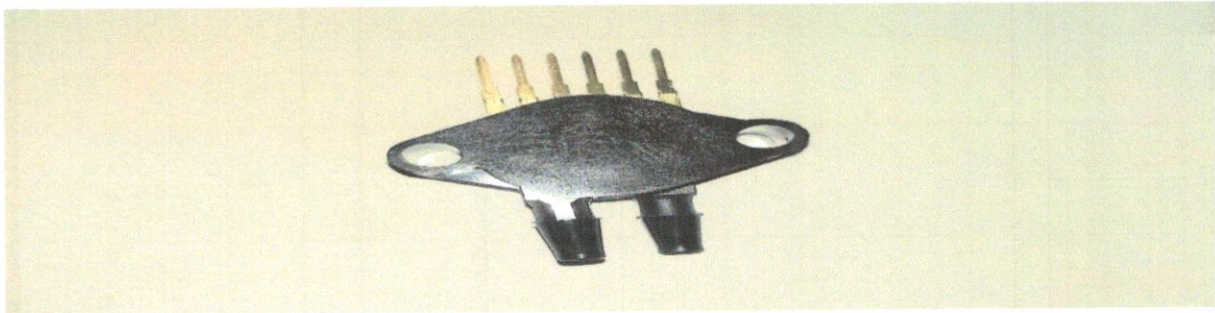


Figure 3.8 : Capteur de pression mpx4250dp

### 3.4.6 Afficheur LCD

La fonction de ce module est d'afficher le résultat de la mesure, Les afficheurs LCD est un élément très important pour afficher les paramètres technique.

#### 3.4.6.1 Description des pins de l'afficheur LCD

1/ Les pins 1, 2 et 3 de l'afficheur sont des broches d'alimentation, on peut les utiliser pour le réglage du contraste en y associant un potentiomètre.

2/ le pin 4 pour sélectionner le registre.

3/ le 5 pour lire/écrire sur l'afficheur.

6/ Le 6 pour la validation

7/ Les pins 7 à 14 sont des entrées de données

8/ Les deux dernières sont pour illuminer l'afficheur

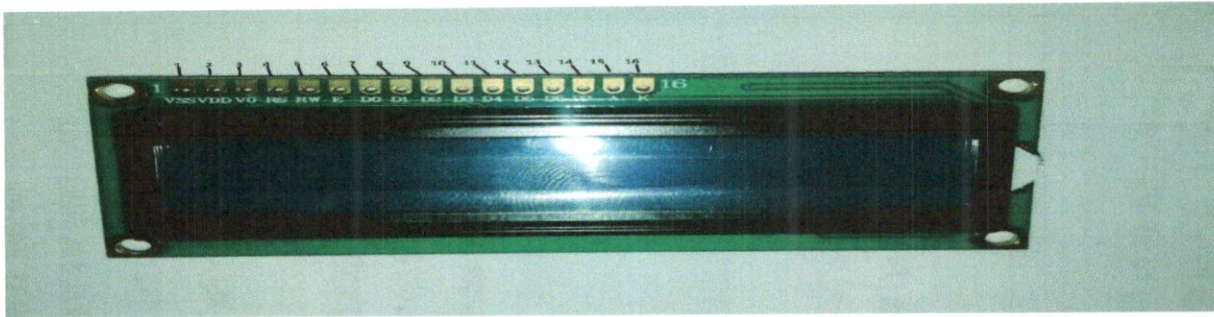


Figure 3.7 : afficheur LCD

**Le tableau suivant présente le rôle de chaque branche de l'afficheur :**

Numéro de la branche	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	Action pour un transfert (enable)
7	DB0	Bite 0 de la donnée/ commande
8	DB1	Bite 1 de la donnée / commande
9	DB2	Bite 2 de la donnée / commande
10	DB3	Bite 3 de la donnée / commande
11	DB4	Bite 4 de la donnée / commande
12	DB5	Bite 5 de la donnée / commande
13	DB6	Bite 6 de la donnée / commande
14	DB7	Bite 7 de la donnée / commande
15	LED +	Anode (+) du rétro-éclairage
16	LED -	Anode (-) du rétro-éclairage

Tableau 3.2 : présentation du branchement de l'afficheur LCD



### 3.4.7 Potentiomètre

Ce sont des résistances que l'on peut faire varier manuellement (les potentiomètres de tableau) ou avec un tournevis (les potentiomètres ajustables). Ils peuvent être aussi appelés rhéostat ou résistance variable.

Le symbole est aussi parfois utilisé pour exprimer que dans un circuit, la valeur de la résistance est commandée.

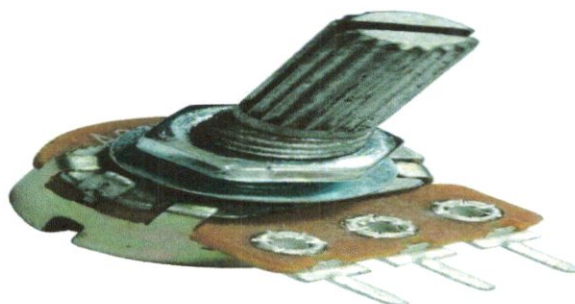


Figure 3.10 : potentiomètre

### 3.5 Réalisation de montage

Pour finir la réalisation de notre montage on a besoin de passer par plusieurs étapes testé le matériel faire le branchement a la carte arduino et l'affiché le balayage radar sur processing et les paramètres météorologique sur l'afficheur LCD.

#### 3.5.1 Présentation de l'interfaçage Arduino-PC par câble USB

Premièrement pour tester le fonctionnement des capteurs utilisés: (ultrason, dth11 et mpx4250 et le moteur), On a essayé de faire un câblage entre ces capteurs et la carte Arduino UNO. La carte réalisée est reliée de façon simple par un câble USB. Une image réelle de la carte météo réalisée sur une plaque d'essai et connectée par un câble USB est présentée à la figure



Figure 3.11 : image réel de montage connecté par USB

### 3.5.2 Branchement les capteurs a la carte ARDUINO

Pour commencer la réalisation on prend le matériel qu'on a préparé ensuite en la branche à l'aide des câbles la carte ARDUINO

- La première étape de notre réalisation est de branché le servomoteur on choisissant les pins d entré et de sorti de l'ARDUINO

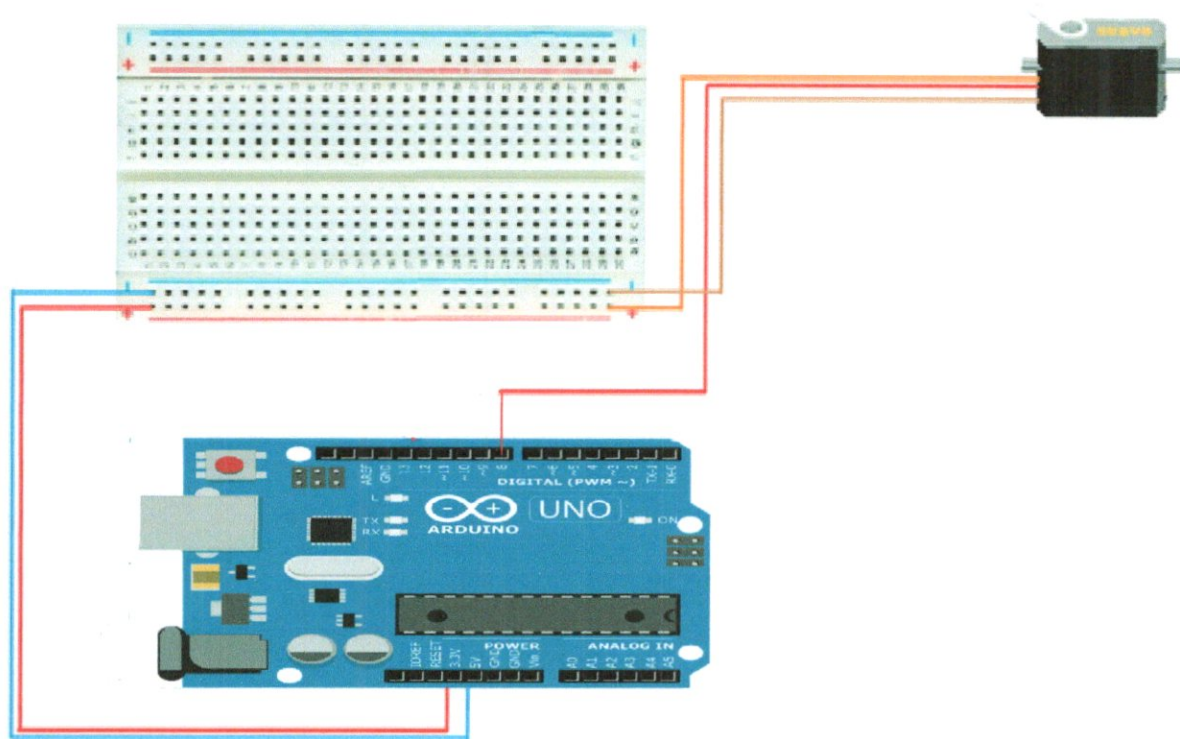


Figure 3.12 : branchement de servomoteur a la carte arduino

- Le câble rouge au port 5 v
- Le câble noir a la masse
- Le câble jaune commende de la position au 9

Deuxième étape est le branchement de capteur ultrason à la carte ARDUINO

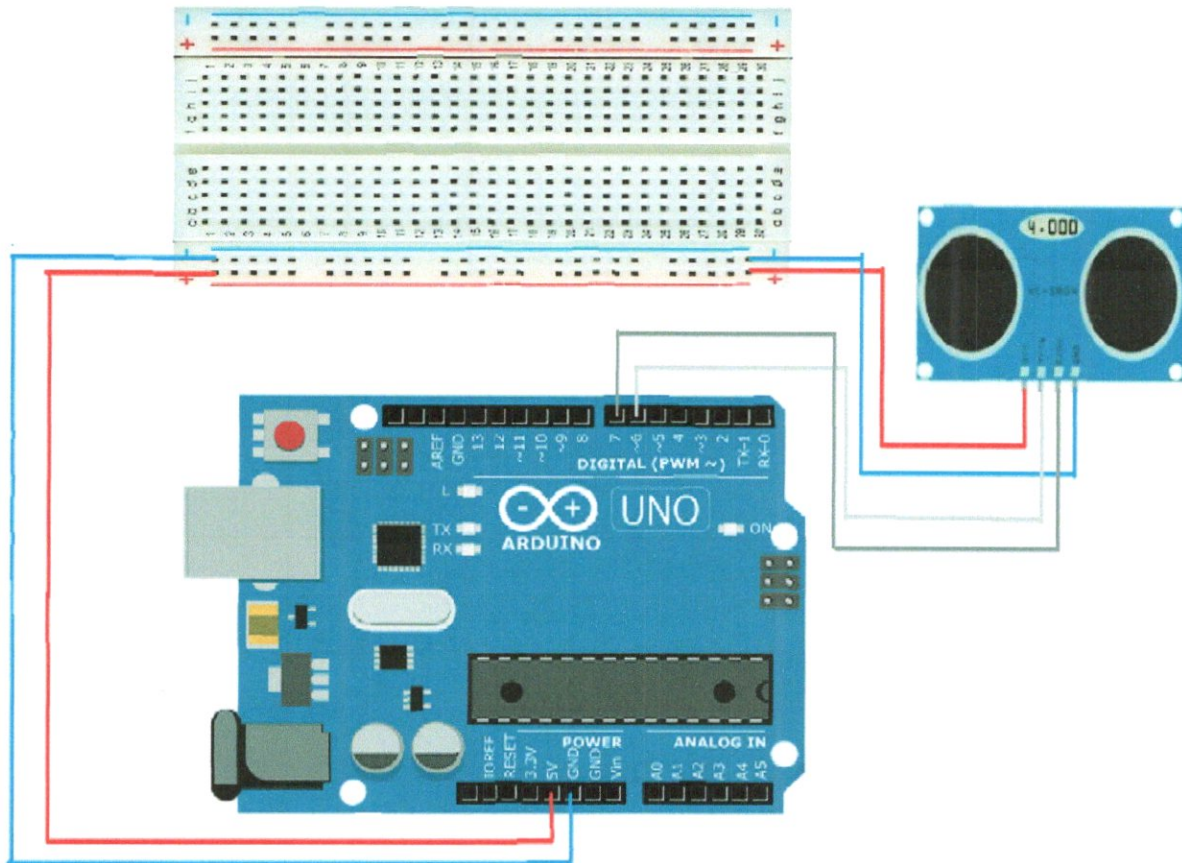


Figure 3.13 : branchement de de capteur ultrason

- VCC au port 5 v
- Trig au port 4
- Echo au port 5
- GND au port GND



- En quatrième étape de la réalisation est le branchement de capteur de température et d'humidité DHT11

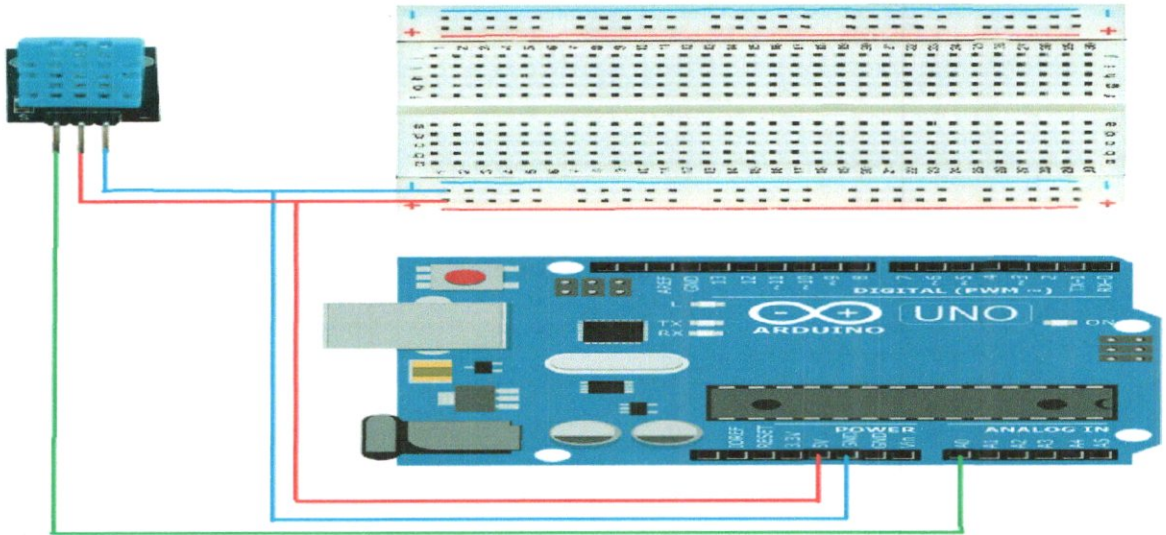


Figure 3.14 : branchement de capteur DHT11 a la carte ARDUINO

- Le capteur de pression mpx4250dp à la carte ARDUINO

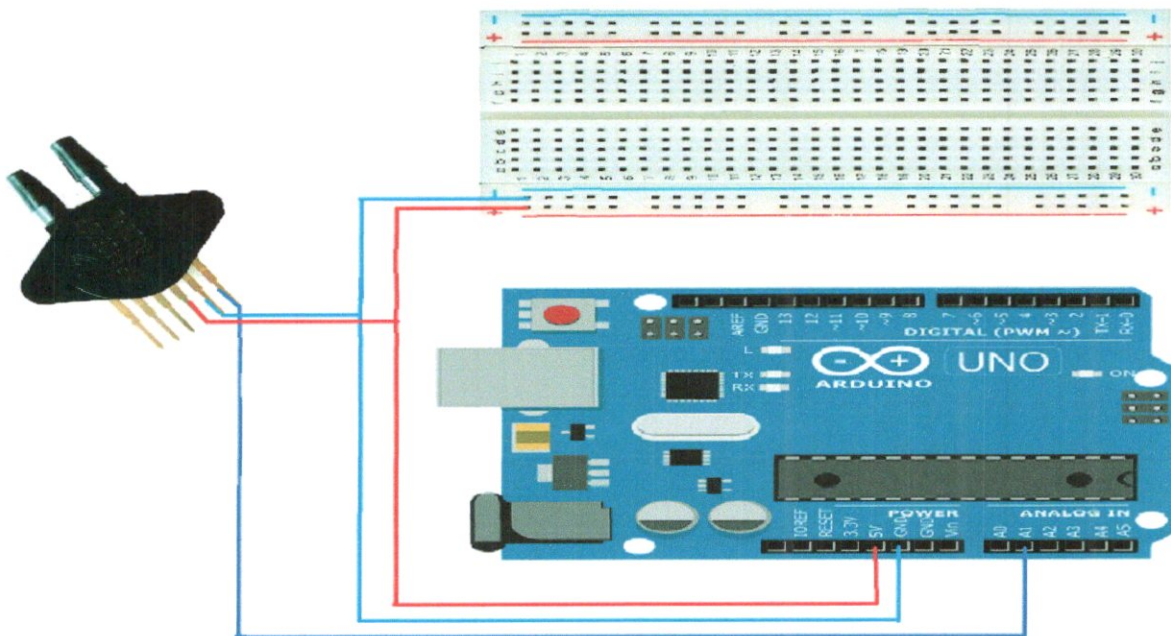


Figure 3.15 : branchement de capteur de pression à la carte arduino

Afin d'affiché les paramètres que les capteurs en capté on doit brancher un afficheur LCD



Tableau des connexions des capteurs aux pins de la carte arduino

Composent	Pins des	Pin d'arduino
Servomoteur	Le rouge	5v
	Le maron	GND
	Orange	8
Capteur ultrason	Vcc	5v
	Trig	7
	Echo	6
	GND	GND
Afficheur LCD	1	GND
	2	5V
	3 VE	
	4	12
	5	
	6 EN	11
	11 DB4	5
	12 DB5	4
	13DB6	3
	14 DB7	2
	15 LED+	
16LED-		
Capteur de température et d'humidité	Pin +	5v
	Pin -	GND
	Pin3	A0
Capteur de pression	Pin+	5v
	Pin -	GND
	Pin3	A1

Tableau 3.3 : connexion des pins de circuit

### 3.5.3 Alimentation de la carte et des capteurs

La consommation de la carte arduino des capteurs (capteur ultrason, capteur de température et d'humidité, capteur de pression, capteur de vitesse de vent) et d'afficheur LCD est très faible il y a pas de courant important la carte exige une alimentation de 5v et pour cela on a utilisé le port USB de l'ordinateur.

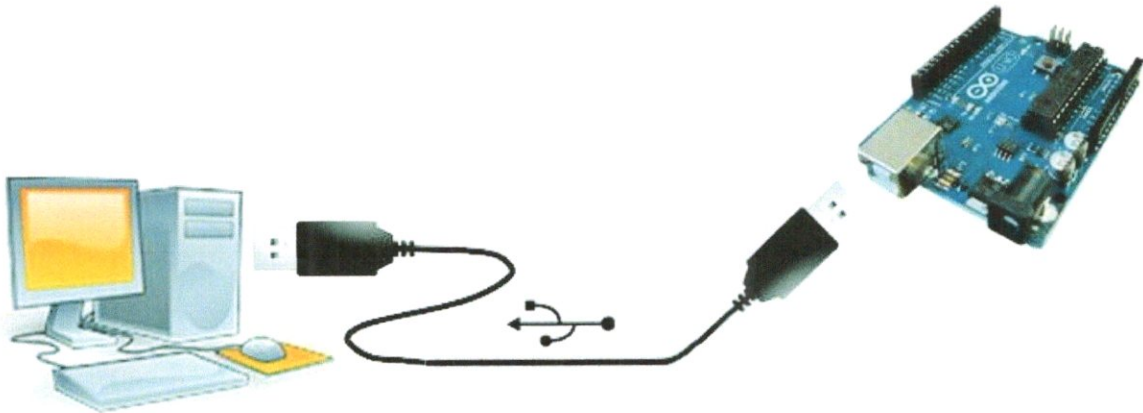


Figure 3.17 : Alimentation de la carte arduino

### 3.5.4 Précision de capteur ultrason

Pour tester la précision de capteur ultrason on doit mettre un obstacle en face de capteur et prendre des mesures réelles et les mesure donné par le capteur

Distance en Cm	Mesure de capteur	Erreur	Erreur %
1	1.02	0.2	2 %
5	4.2	0.8	16 %
10	9.68	0.32	3.2 %
15	14.45	0.55	3.66 %
20	18.78	1.22	6.1 %
25	23.7	1.3	5.2 %
30	29.15	0.85	2.83 %
35	34.20	0.8	2.28 %
40	39.08	0.92	2.3 %

Tableau3.4 : précision de capteur ultrason





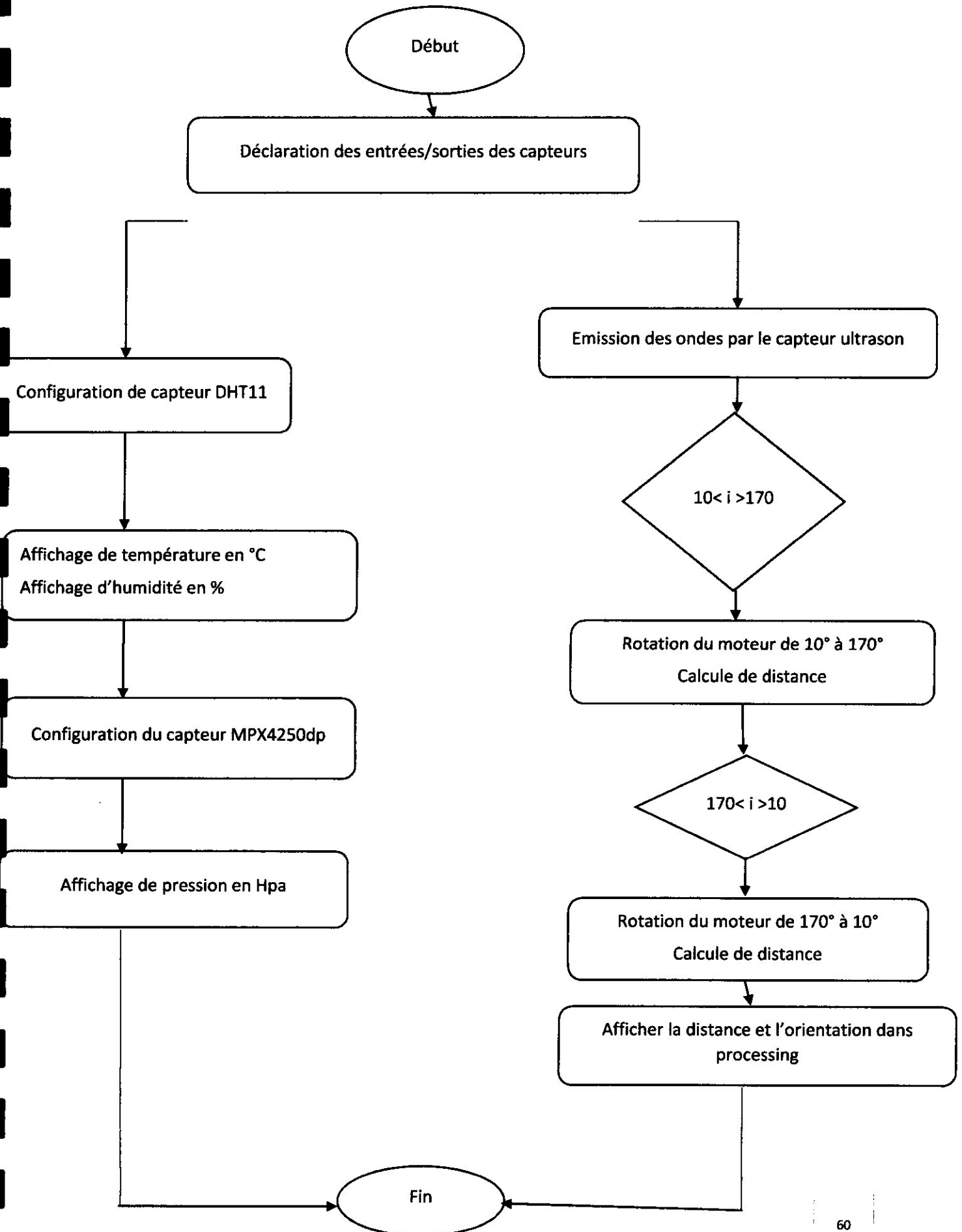
Figure 3.18 : précession de capteur ultrason

### 3.5.5 Performances des capteurs à Caractéristiques métrologiques

De manière à classer les capteurs en fonction de leurs performances, on est amené à définir des paramètres qui permettent de les sélectionner en fonction de l'application. Chaque capteur présente certaines caractéristiques métrologiques qui définissent ses limites d'utilisation et de précision. Ces limites dépendent non seulement des mesures, mais aussi des grandeurs d'influence qui viennent perturber l'élément des mesures



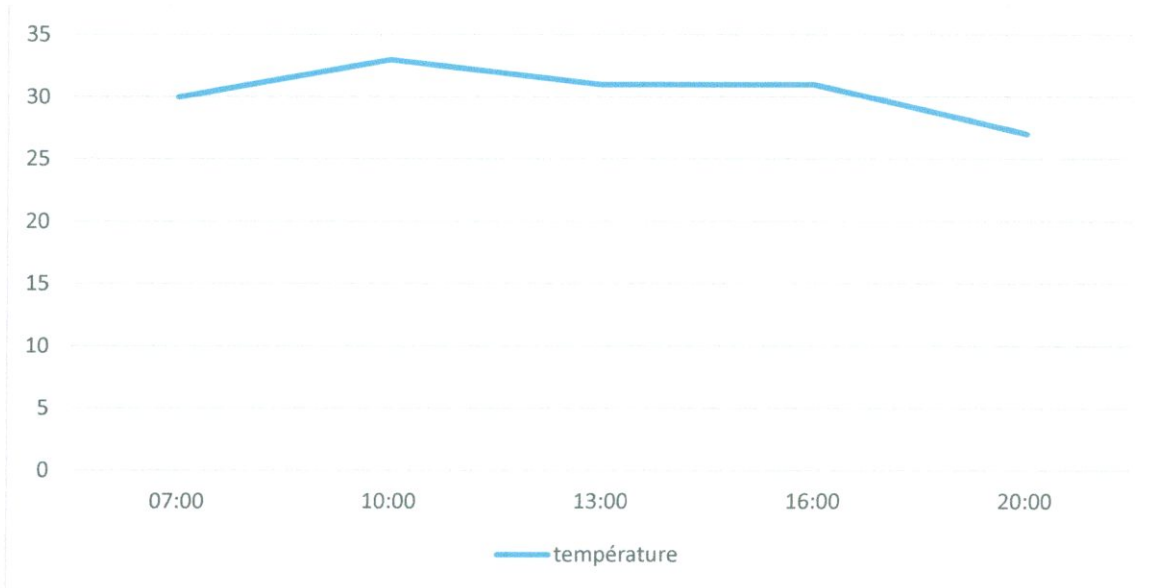
3.5.6 Diagramme de programme



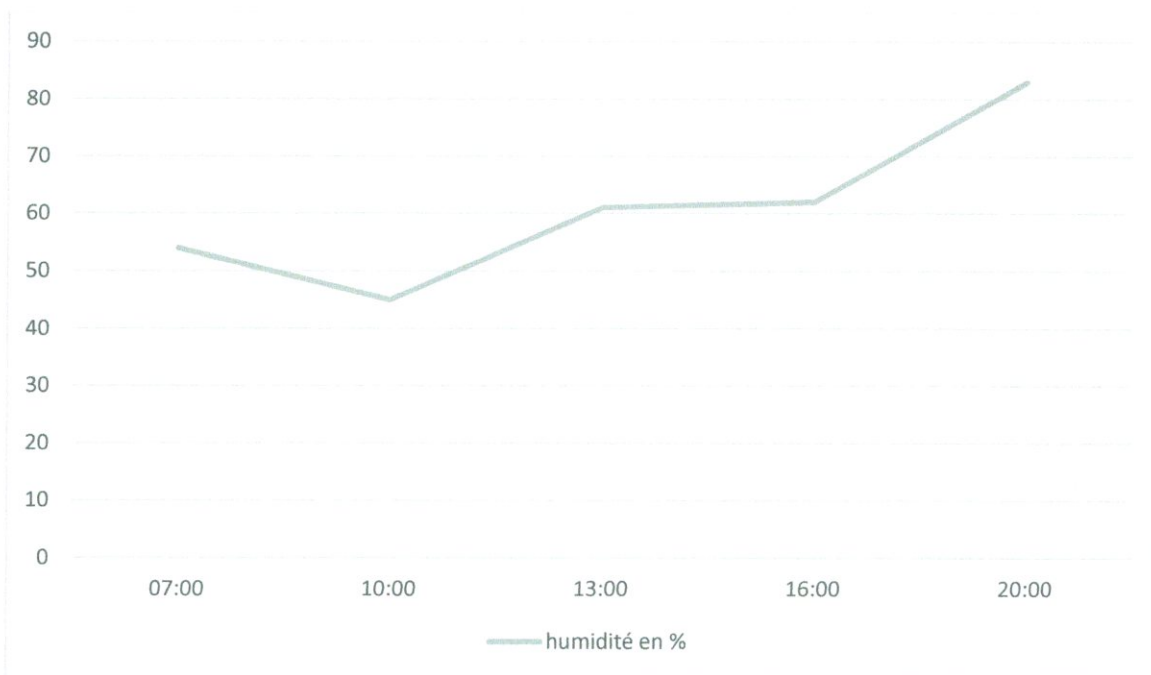
### 3.5.7 Installation de montage à l'extérieur de Laboratoire

Nous avons installé la réalisation à l'extérieur toute au long au long de la journée afin d'obtenir des résultats des changements météorologique (température, humidité, pression) a défèrent moment de la journée

**Remarque** les défèrent résultat concernant Bejaia le 06/08/2018



**Figure 3.19** graphe exprime la variation de la température



**Figure 3.20** : graphe exprime la variation humidité

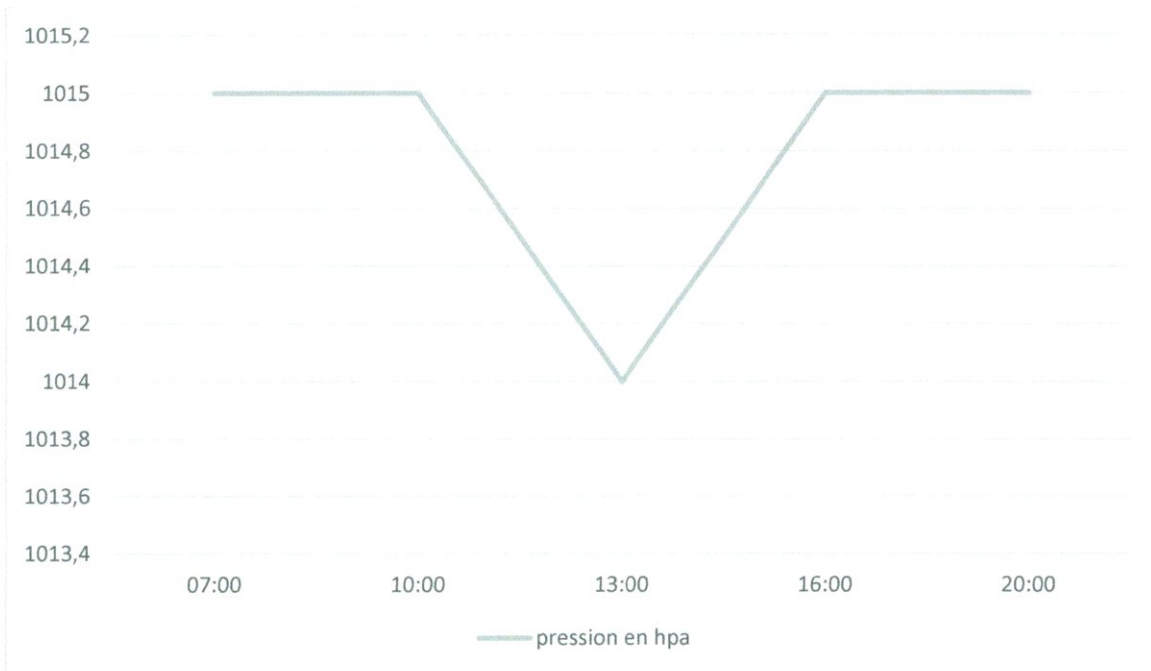


Figure 3.21 : graphe exprime la variation de la pression

### 3.5.8 Interface de balayage radar

Il y a plusieurs méthodes qu'on peut utiliser pour afficher les résultats et dans notre projet nous avons choisi de faire affiché le balayage radar à l'aide de logiciel processing

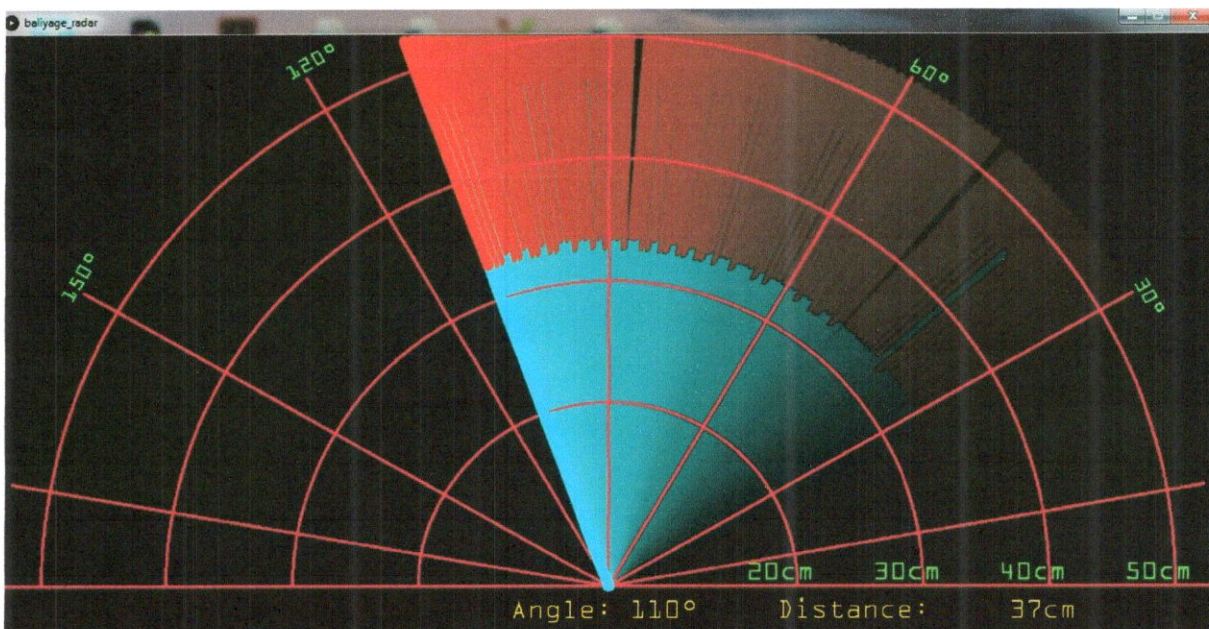


Figure 3.22 : interface de projet radar

### 3.6 Conclusion

Dans ce chapitre nous avons entamé la phase finale de notre projet, simulation des deux importantes installations technique d'une tour de contrôle à l'aide d'une carte ARDUINO UNO et différent capteurs

La simulation du radar de détection a été faite avec un capteur ultrason, à fin de mesuré la distance et la direction des cibles de 10 à 170 degré avec l'aide du servomoteur.

La distance et la direction mesurée sont légèrement différentes à la valeur réelle.

La simulation de la station météorologique avec les capteurs, DHT11 pour la température et l'humidité, MPX4250DP pour la pression.

Les résultat obtenus sont notés et discuté dans la figure (3.19 ,3.20 ,3.21)

Finalement et pour terminer notre travaille on a affiché les résultats obtenus sur une interface PROCESSING à fin de simplifié la lecture des résultats et de donner ressemblance d'affichage comme sur les écrans d'un contrôleur aérien



## Conclusion générale

Notre projet comporte un travail théorique accompagné d'une réalisation, son objectif consiste sur l'étude et la simulation d'une tour de contrôle. Pour faire preuve de notre réalisation nous avons utilisé une carte ARDUINO UNO, un capteur de distance à ultrason, un servomoteur, un capteur de température et d'humidité DHT11, un capteur de pression MPX4250dp, et un afficheur LCD 2\*16.

Dans ce travail on a donné les informations essentielles sur les radars, les stations météorologique et les composant électronique utiliser à fin de bien comprendre le fonctionnement de chaque capteurs et son domaine d'utilisation

Notre étude est diviser en deux partie, la partie radar et la partie station météorologique.

Le radar à détection fonction en rotation de 10 à 170 degrés à l'aide du servomoteur, il détecte les cibles, calcule la distance et l'orientation suivant un programme avancé et selon plusieurs étapes et pour conclure tout ça on à utiliser un détecteur ultrason HC-SR04 qu'on à trouver plus convenable pour cette étape de travail (détection, orientation, distance).

La station météorologique est composé de deux détecteurs (DHT 11 pour la température et l'humidité, MPX4250dp pour la pression) à l'aide de ces capteurs et un afficheur LCD on a peu détecté et afficher les différents paramètres météorologique nécessaire pour le pilote à fin de s'informer sur les conditions météo dont il a besoin pour son vol.

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 dont ses caractéristiques particulières nous à faciliter les taches surtout en ce qui concerne sa programmation

Et pour terminer on à utiliser le programme PROCESSING pour réaliser une interface radar de détection pour afficher l'emplacement et la distance des cibles que soit fixe ou mobile et on a minimisé la distance à 50cm pour réduire l'erreur de précision du capteur ultrason

## Bibliographié

- [1] Fundamentals of Radar Technology for Level Gauging © KROHNE 07/2003  
7.02337.22.00GR  
th edition (revised and expanded) by Dr.-Ing. Detlef Brumbi KROHNE Messtechnik GmbH & Co. KG
- [2] <http://www.radartutorial.eu/01.basics/Principe%20du%20Radar.fr.html>  
(consulté le 22/04/2018)
- [3] Thierry Rabaud Sagem Défense Sécurité Directeur du Département 2006
- [4] GIRARD M.C. & GIRARD C.M. 1999. Traitement des données de télédétection.  
DUNOD Ed. Paris, pages 482 à 495
- [5] Preeti Jain. Different Types of RADARs ,Engineers Garage .... Inspiring creations
- [6] Pierre Vaillant et Christian Wolff, « Suppression des lobes secondaires à l'interrogation (ISLS) » [archive], sur Radartutorial (consulté le 20 mars 2018)
- [7] robert du bois structure et application des émetteur et des récepteur radiotélévision radar communication par faisceau hertzien ou par satellite
- [8] Bassem R. Mahafza Radar Systems Analysis and Design Using MATLAB
- [9] Skolnik, Merrill, 1990: Radar Handbook 2nd Edition
- [10] JEANE MERMOS institut d'aéronautique météorologie Edition numéro 1
- [11] <https://www.meteocontact.fr/pour-aller-plus-loin/la-pression-atmospherique>  
(consulté.15/05/2018)
- [12] william G. hopkins physiologie végétale
- [13] andré hufty introduction à la climatologie
- [14] Lechalupé. J, "cours d'initiation à Arduino", Université Paul Sabatier, Mai 2014.
- [15] Arduino: A Complete Guide (Par Tech Guru) ; Auteur : Tech Guru
- [16] Frédéric Bouquet et Julien Bobroff MICROCONTROLEUR ARDUINO Projets de physique statistique Magistère de physique fondamentale Université Paris Sud
- [17] Microcontrollers Fundamentals for Engineers and Scientists (Par Steven Frank Barrett, Daniel J. Pack); Auteur: Steven Frank Barrett, Daniel J. Pack, First Edition
- [18] Embedded Systems Interfacing for Engineers Using the Freescale HCS08 ... (Par Douglas Summerville); Auteur: Douglas Summerville, ISBN9781608450053

[19] Microprocessor & Microcontroller (Par A.P.Godse,D.A.Godse) ; Auteur :  
A.P.Godse,D.A.Godse,ISBN9788184317695

[20] W. Durfee Université de Minnesota Arduino Microcontrôleur Guide

[21] Astalaseven , Eskimon et olyte Arduino pour bien commencer en électronique et en programmation

[22] Astalaseven , Eskimon et olyte Arduino pour bien commencer en électronique et en programmation

## Programme ARDUINO

```
#include <Servo.h> // bibliothèque servomoteur
#include <LiquidCrystal.h> // bibliothèque afficheur lcd
#include <dht.h> //bibliothèque dht11

//declaration des entrées/sorties
//declaration des variables
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
const int trigPin=6;
const int echoPin = 7;
long duration ;
int distance ;
Servo monServo;
dht DHT;
#define DHT11_PIN A0

void setup(){
  pinMode(trigPin,OUTPUT );
  pinMode (echoPin, INPUT);
  Serial.begin (9600);
  lcd.begin(16,2);
  monServo.attach(8);

}
void loop() {
  // rotation de servo moteur de 10 a 170 degrees
  for ( int i=10;i<=170;i++){
    monServo.write (i) ;
    delay(30);
```



```

    distance = calculDistance();
    Serial.print(i);
    Serial.print(",");
    Serial.print(distance);
    Serial.print(".");
}

//rotation de servo moteur de 170 a 10 degrees
for(int i=170;i>10;i--){
    monServo.write(i);
    delay(30);
    distance = calculDistance();
    Serial.print(i);
    Serial.print(",");
    Serial.print(distance);
    Serial.print(".");
}
// lire la valeur de capteur MPX
{
    float pression = readPression(A1);

    // lire les valeur de capteur dht11
}
{ int chk = DHT.read11 (DHT11_PIN);
  switch (chk) {
    case 0: Serial.print ("");
  }}

//affichage d'humidity et de température
lcd.clear();
lcd.setCursor(0,0);

```

```

lcd.print("Humidity = ");
lcd.print ( DHT.humidity);
lcd.setCursor(0,1);
lcd.print("Temp = ");
lcd.println (DHT.temperature);

}

```

```

//affichage de pression

```

```

float readPression(int pin){
    int pressionVal = analogRead(pin);
    float pression=(((pressionVal/1024.0)+0.095)/0.000009)/10;
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("p = ");
    lcd.println(pression);
    lcd.println("hpa");
    delay(2000);

}

```

```

// calcule de distance

```

```

int calculDistance (){
    digitalWrite (trigPin, LOW);
    delayMicroseconds(100);
    digitalWrite(trigPin,HIGH);
    delayMicroseconds(100);
    digitalWrite(trigPin,LOW);
    duration = pulseIn(echoPin, HIGH);
    distance= duration*0.034/2;
    return distance;}

```

## Programme processing

```
import processing.serial.*; // importé la libriré de comunication
import java.awt.event.KeyEvent; //bibliothèque d'importations pour lire les données du port série
import java.io.IOException;

Serial myPort;
//definir les variables
String angle="";
String distance="";
String data="";
String noObject;
float pixsDistance;
int iAngle, iDistance;
int index1=0;
int index2=0;
PFont orcFont;
void setup() {

    size (1280, 650);
    smooth();
    // démarre la communication série
    myPort = new Serial(this,"COM5", 9600);
    myPort.bufferUntil('.');
    orcFont = loadFont("OCRAExtended-30.vlw");
}
void draw() {

    fill(98,245,31);
    textFont(orcFont);
    // simuler le flou de mouvement et le fondu lent de la ligne mobile

    noStroke();
```

```

fill(0,4);
rect(0, 0, width, height-height*0.065);

fill(98,1,52);
// dessiner le radar
Radar();
Line();
Object();
Text();
}
//lire les données du port série
void serialEvent (Serial myPort) {

data = myPort.readStringUntil('.');
data = data.substring(0,data.length()-1);

index1 = data.indexOf(",");
angle= data.substring(0, index1);
distance= data.substring(index1+1, data.length());

iAngle = int(angle);
iDistance = int(distance);
}

void Radar() {
pushMatrix();
translate(width/2,height-height*0.074);
noFill();
strokeWeight(2);
stroke(300,20,52);

```



```

// dessine les lignes d'arc

arc(0,0,(width-width*0.0625),(width-width*0.0625),PI,TWO_PI);
arc(0,0,(width-width*0.27),(width-width*0.27),PI,TWO_PI);
arc(0,0,(width-width*0.479),(width-width*0.479),PI,TWO_PI);
arc(0,0,(width-width*0.687),(width-width*0.687),PI,TWO_PI);

// dessine les lignes d'angle
line(-width/2,0,width/2,0);
line(0,0,(-width/2)*cos(radians(10)),(-width/2)*sin(radians(10)));
line(0,0,(-width/2)*cos(radians(30)),(-width/2)*sin(radians(30)));
line(0,0,(-width/2)*cos(radians(60)),(-width/2)*sin(radians(60)));
line(0,0,(-width/2)*cos(radians(90)),(-width/2)*sin(radians(90)));
line(0,0,(-width/2)*cos(radians(120)),(-width/2)*sin(radians(120)));
line(0,0,(-width/2)*cos(radians(150)),(-width/2)*sin(radians(150)));
line(0,0,(-width/2)*cos(radians(170)),(-width/2)*sin(radians(170)));

line((-width/2)*cos(radians(30)),0,width/2,0);
popMatrix();
}

void Object() {
  pushMatrix();
  translate(width/2,height-height*0.074);
  strokeWeight(12);
  stroke(255,50,18);
  pixsDistance = iDistance*((height-height*0.3800)*0.025);
  // limiter la portée à 50 cm
  if(iDistance<50){
    // dessiner l'objet en fonction de l'angle et de la distance

```

```

    line(pixsDistance*cos(radians(iAngle)),-pixsDistance*sin(radians(iAngle)),(width-
width*0.505)*cos(radians(iAngle)),-(width-width*0.505)*sin(radians(iAngle)));
}
popMatrix();
}

```

```

void Line() {
    pushMatrix();
    strokeWeight(12);
    stroke(30,250,260);
    translate(width/2,height-height*0.074);
    line(0,0,(height-height*0.15)*cos(radians(iAngle)),-(height-height*0.15)*sin(radians(iAngle))); //
dessine la ligne en fonction de l'angle
    popMatrix();
}

```

//dessiner les textes sur l'écran

```

void Text() {

    pushMatrix();

    fill(0,0,0);
    noStroke();
    rect(0, height-height*0.0648, width, height);
    fill(98,245,31);
    textSize(30);

    text("20cm",width-width*0.3854,height-height*0.0833);
    text("30cm",width-width*0.281,height-height*0.0833);
    text("40cm",width-width*0.177,height-height*0.0833);
    text("50cm",width-width*0.0729,height-height*0.0833);
}

```

```

textSize(30);
fill(113,10,250);
text("",width-width*0.975, height-height*0.0207);
fill(250,238,10);
text("Angle: " + iAngle + "°", width-width*0.58, height-height*0.0207);
text("Distance: ", width-width*0.36, height-height*0.0207);
if(iDistance<80) {
text("  " + iDistance + "cm", width-width*0.225, height-height*0.0207);
}
textSize(25);
fill(98,245,60);
translate((width-width*0.4994)+width/2*cos(radians(30)),(height-height*0.0907)-
width/2*sin(radians(30)));
rotate(-radians(-60));
text("30°",0,0);
resetMatrix();
translate((width-width*0.503)+width/2*cos(radians(60)),(height-height*0.0888)-
width/2*sin(radians(60)));
rotate(-radians(-30));
text("60°",0,0);
resetMatrix();
translate((width-width*0.507)+width/2*cos(radians(90)),(height-height*0.0833)-
width/2*sin(radians(90)));
rotate(radians(0));
text("90°",0,0);
resetMatrix();
translate(width-width*0.513+width/2*cos(radians(120)),(height-height*0.07129)-
width/2*sin(radians(120)));
rotate(radians(-30));
text("120°",0,0);
resetMatrix();
translate((width-width*0.5104)+width/2*cos(radians(150)),(height-height*0.0574)-
width/2*sin(radians(150)));

```

```
rotate(radians(-60));
```

```
text("150°",0,0);
```

```
popMatrix();
```

```
}
```