République Algérienne Démocratique Populaire

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université d'Ibn Khaldoun – Tiaret

Faculté des Mathématiques et de l'Informatique

Département Informatique

Mémoire de Master

Thème

Acquisition et traitement d'un signal audio

Pour l'obtention du diplôme de Master

Spécialité : Génie informatique

Option : Système d'information et technologie web

Rédigé par :

- **BOUKHORS Fethi**

Dirigé par :

- **Mr. TIFFOUR Abdelkader** *Encadreur*
- **Mr. Prof HAMOUDI Abdelhamid** *Co Encadreur*

Devant le jury d'examen composé de :

- **Mr.DAOUD Bachir** *Président*
- **Mme.BENOTHMANE Lilia** *Examinateur*

Année universitaire : 2016/2017

Dédicace

Je dédie ce mémoire à :

· Mes parents :

Ma mère, qui a œuvré pour ma réussite, de par son amour, son soutien, tous les sacrifices consentis et ses précieux conseils, pour toute son assistance et sa présence dans ma vie, reçois à travers ce travail aussi modeste soit-il, l'expression de mes sentiments et de mon éternelle gratitude.

Mon père, qui peut être fier et trouver ici le résultat de longues années de sacrifices et de privations pour m'aider à avancer dans la vie malgré son état de santé. Puisse Dieu faire en sorte que ce travail porte son fruit ; Merci pour les valeurs nobles, l'éducation et le soutient permanent venu de toi.

Mes frères et ma sœur qui n'ont cessé d'être pour moi des exemples de persévérance, de courage et de générosité.

Mes amis qui m'ont soutenu et encourager.

Mes professeurs de L'Université qui doivent voir dans ce travail la fierté d'un savoir bien acquis.

BOUKHORS Fethi

Remercîment

Nulle œuvre n'est exaltante que celle réalisée avec le soutien moral et financier des personnes qui nous sont proches.

Je tiens à exprimer ma plus profonde reconnaissance à :

A DIEU, pour m'avoir donné la force dans les moments difficiles d'éditer ce mémoire.

Mon père BOUKHORS Abdelkader et à ma mère Siad Fatima Zohra qui m'ont toujours entouré et motivé à sans cesse devenir meilleur ;

Mes frères et ma sœurs : Abdelkrim, Ouafaà, Mohamed, et Mounir qui m'ont assisté dans ces moments difficiles et m'ont servi d'exemple ;

Mes tantes, oncles, cousins et cousines, neveux et nièces paternels que je pourrais tous citer ;

Mes amis et amies de par le monde qui n'ont cessé de m'encourager ;

Tous mes professeurs d'Université pour leurs disponibilité et conseils ;

Mes professeurs encadreurs Mr TIFFOUR Abdelkader et Mr HAMOUDI Abdelhamid pour son aide et sa précieuse attention.

Mr DAOUD Bachir et Mr OUARED Djilali qui m'ont aidé dans mon mémoire ;

Tous mes compagnons de promotion ;

Trouvez ici l'expression de ma profonde gratitude et reconnaissance.

BOUKHORS Fethi

Résumé

L'automatisation des tâches dans la vie quotidienne est devenue quelque chose de primordiale dans dives domaine médicale ; industriel, économique, …etc. Le domaine du traitement du signal avec ces théories sont appliquées et déployées dans des divers machines et appareilles.

Le traitement du signal audio consiste à extraire des informations et les exploitées pour prendre une décision. Notre travail s'inscrit dans le cadre du traitement du signal audio des poulets de chair. La problématique c'est la détection de changement de son chez les poulets pour prendre une décision de soin.

Notre objectif dans ce mémoire est de proposer un système d'acquisition et de comparaison entre deux sons audio afin de détecter cette maladie.

Pour réaliser notre système nous avons modélisé le fonctionnent du processus d'acquisition et intégrée notre solution dans une carte Arduino. Afin de valider notre proposions nous avons procède a une étude d'expérimentation. Le résultat obtenu montre la faisabilité de notre système.

Mot clés : automatisation, traitement du signal, signal audio, acquisition, détection du changement.

Abstract

The automation of tasks in everyday life has become something of primordial in the medical field; Industrial, economic, etc. The field of signal processing with these theories are applied and deployed in various machines and devices

Audio signal processing involves extracting informations and exploiting it to make a decision. Our work is part of the audio signal processing of broiler chickens. The problem is the detection of sound change in chickens to make a careful decision.

Our objective in this thesis is to propose a system of acquisition and comparison between two audio sounds in order to detect this disease.

To realize our system we have modelled the function of the acquisition process and integrated our solution into an Arduino card. In order to validate our proposals we proceeded to an experimental study. The result shows the feasibility of our system.

Keywords: automation, signal processing, audio signal, acquisition, detection of change.

Introduction générale

Cadre d'étude

- L'automatisation des tâches dans
- Le domaine du traitement du signal avec ces théories est appliqué et déployées dans de divers domaines.

Problématique

Les traitements de signal audio

Le traitement de signal audio consiste à extraire des informations et les exploitées pour la, prise décision. Notre travail s'inscrit dans le cadre de de traitement du signal audio des poulets de chair. La problématique c'est la détection de changement de son chez les poulets pour prendre une décision de soin.

Objectif

Notre objectif dans ce mémoire de proposer un système d'acquisition de comparaison entre deux sons audio afin de détecter cette maladie.

Pour réaliser notre système nous avons modélisé le fonctionnent de processus acquisition et intégrée notre solution dans une carte Arduino.

La modélisation de notre système est basée sur le langage de modélisant UML. Nous avons utilisé quatre type de diagrammes : (i) diagramme de contexte, (ii) diagramme de cas d'utilisation, (iii) Diagramme de séquence.

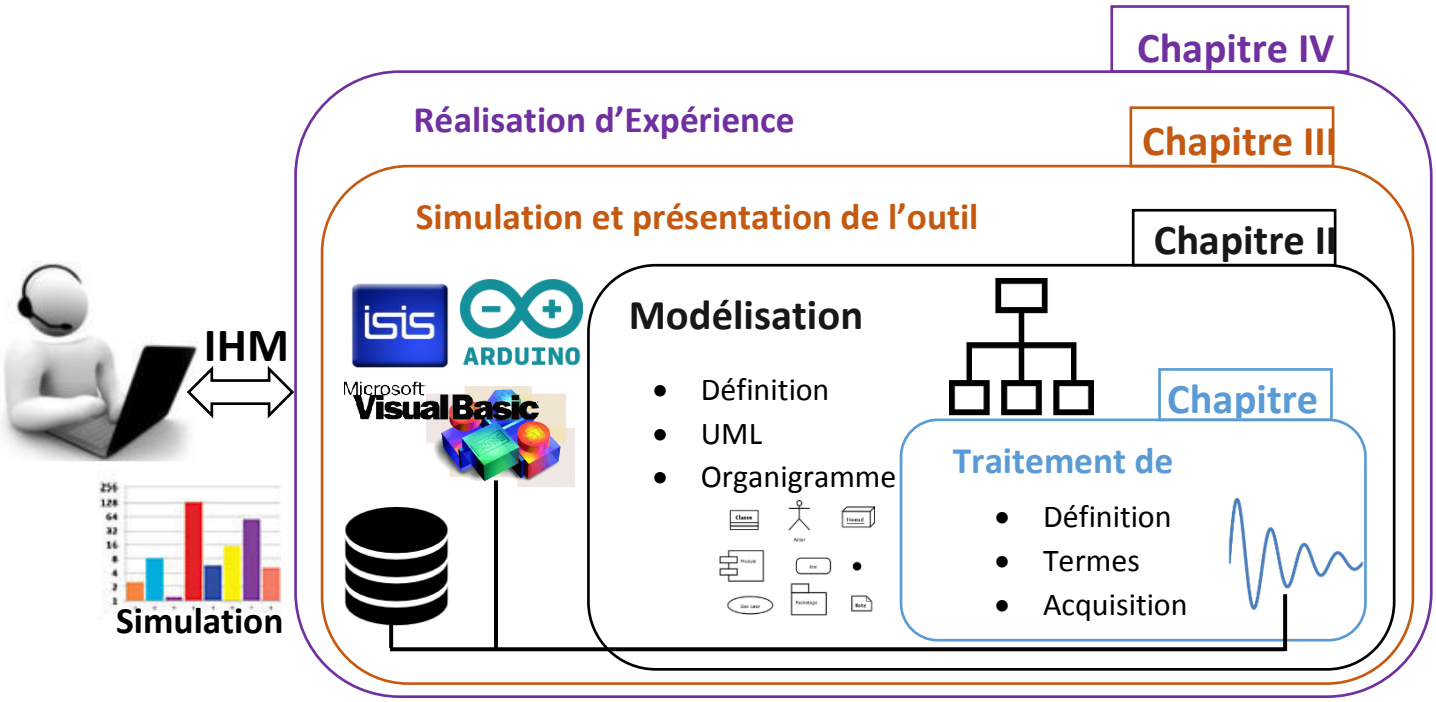
Afin de valider notre proposions nous avons procède a une étude d'expérimentation. Les résultats obtenus montrent la faisabilité de notre système.

Organisation de mémoire

Notre mémoire est ornais en deux chapitres :

- ✓ **Chapitre 01 :** Nous présentons un état de l'art dans lequel on expose des généralités sur le traitement de signal, ainsi que la définition des principales termes de ce domaine.
- ✓ **Chapitre 02 :** Nous présentons la modélisation de notre système informatique pour faciliter sa compréhension, son étude et enfin la solution de problème.
- ✓ **Chapitre 03 :** Nous avons présenté la simulation de notre système à l'aide des outils de simulations.
- ✓ **Chapitre 04 :** Dans ce chapitre on a fait une expérience pour tester la fiabilité de noter système.

Enfin conclusion et perspectives de notre études



I. Chapitre I introduction au traitement de signal

I.1. Introduction

L'information portée par le signal audio peut être analysée de bien des façons. On distingue, généralement, plusieurs niveaux de description non exclusifs : Acoustique, phonétique et bien d'autres. [1].

Dans ce chapitre nous allons, dans un premier temps, décrire les définitions et les termes de traitement de signal puis nous donnerons un aperçu sur les notions de domaine. Nous terminerons par la conversion de la parole en signal électrique et nous rappellerons quelque outil de base utilisée en traitement de signal audio et acoustique.

I.2. Le traitement de signal

Un signal est la représentation physique de l'information qu'il transporte de sa source à son destinataire. Il sert de vecteur à une information. Il constitue la manifestation physique d'une grandeur mesurable (courant, tension, force, température, pression, etc.). Les signaux considérés, sont des grandeurs électriques variant en fonction du temps $s(t)$ obtenues à l'aide de capteurs. Mais le traitement du signal s'applique à tous les signaux physiques (onde acoustique, signal optique, signal magnétique, signal radioélectrique, etc.). Le traitement d'images peut être considéré comme une extension du traitement du signal aux signaux bidimensionnels (images).
L'interprétation des signaux sont regroupés dans la discipline appelée traitement du signal. [2]

I.2.1. Les termes de traitement de signal

- ✓ **Le bruit :** est défini comme tout phénomène perturbateur gênant la perception ou l'interprétation d'un signal, par analogie avec les nuisances acoustiques (interférence, bruit de fond, etc.). La différentiation entre le signal et le bruit est artificielle et dépend de l'intérêt de l'utilisateur : les ondes électromagnétiques d'origine galactique sont du bruit pour un ingénieur des télécommunications par satellites et un signal pour les radioastronomes.
- ✓ **La théorie du signal :** a pour objectif fondamental la « description mathématique » des signaux. Cette représentation commode du signal permet de mettre en évidence ses principales caractéristiques (distribution fréquentielle, énergie, etc.) et d'analyser les modifications subies lors de la transmission ou du traitement de ces signaux.
- ✓ **Le traitement du signal :** est la discipline technique qui, s'appuyant sur les ressources de l'électronique, de l'informatique et de la physique appliquée, a pour objet l'élaboration ou l'interprétation des signaux. Son champ d'application se situe donc dans tous les domaines

concernés par la perception, la transmission ou l'exploitation des informations véhiculées par ces signaux.

- ✓ **Le traitement de l'information :** fournit un ensemble de concepts permettant d'évaluer les performances des systèmes de transfert d'informations, en particulier lorsque le signal porteur de message est bruité. Cela inclut les méthodes de « codage de l'information » dans le but de la réduction de redondance, de la correction des erreurs, de la confidentialité (cryptage). L'ensemble des concepts et méthodes développés dans le traitement de l'information et du signal forme la théorie de la communication. [2]

I.3. Principales fonctions du traitement du signal

Les fonctions du traitement du signal peuvent se diviser en deux catégories : l'élaboration des signaux (incorporation des informations) et l'interprétation des signaux (extraction des informations). Les principales fonctions intégrées dans ces deux parties sont les suivantes :

I.3.1. Élaboration des signaux :

- **Synthèse** : création de signaux de forme appropriée en procédant par exemple à une combinaison de signaux élémentaires
- **Modulation, changement de fréquence** : moyen permettant d'adapter un signal aux caractéristiques fréquentielles d'une voie de transmission
- **Codage** : traduction en code binaire (quantification), etc.

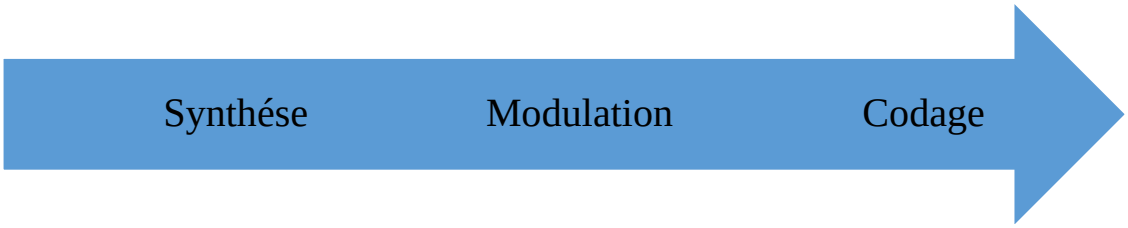


Figure I-1 : *Elaboration des signaux*

I.3.2. Interprétation des signaux :

- ♦ **Filtrage** : élimination de certaines composantes indésirables.
- ♦ **Détection** : extraction du signal d'un bruit de fond (corrélation) ;
- ♦ **Identification** : classement d'un signal dans des catégories préalablement définies.
- ♦ **Analyse** : isolement des composantes essentielles ou utiles d'un signal de forme complexe (transformée de Fourier) ;

- **Mesure** : estimation d'une grandeur caractéristique d'un signal avec un certain degré de confiance (valeur moyenne, etc.). [2]

Figure I-2 : *Interpretation des signaux*

I.3.3. Différents secteurs et différentes branches

- Les signaux sont présents dans différents secteurs (électronique, optique, audiovisuels, informatiques…).
- Quelques branches particulières qui nous intéressent.
  - traitement d'image (déjà vu).
  - traitement de son (Parole, Voix,).

I.3.4. Classification des signaux

I.3.4.1. Signaux analogiques

Signaux produits de manière naturelle, continus (capteurs, amplificateurs, CNA). Traitement réalisé par circuits électroniques, (ou manuellement)

I.3.4.2. Signaux numériques

Signaux utilisés dans le traitement informatique, discrets, facilité et rapidité de traitement. Ils sont artificiels. Traitement réalisé par micro-ordinateurs, DSP (microprocesseurs spécialisés).

I.4. Traitement de signal audio

I.4.1. Définition

Un signal audio est une représentation du son, typiquement comme une tension électrique. Les signaux audios ont des fréquences dans la plage de fréquences audio d'environ 20 à 20 000 Hz (les limites de l'audition humaine). Les signaux audios peuvent être synthétisés directement ou peuvent provenir d'un transducteur tel qu'un microphone, un capteur d'instrument de musique, une cassette de phonographe ou une tête de bande. Les haut-parleurs ou les écouteurs convertissent un signal audio électrique en son. Les représentations numériques des signaux audio existent dans une variété de formats. [3].

Il existe de nombreuses façons de classer les signaux audio. Si l'on considère la source des signaux audio, on peut les classer en trois catégories principales :

- **Les sons des animaux** : les aboiements du chien, le miaulement du chat, le coassement de la grenouille. (En particulier, Bioacoustiques est une science interdisciplinaire, qui étudie la production sonore et la réception chez les animaux.).
- **Sons des non-animaux** : Sons des moteurs de voiture, tonnerre, claquement de porte, instruments de musique, etc.
- **La parole humaine** : les différentes voix et tension et amplitude des vois humaines nécessite un traitement spécial de ka paroles humaine [3].

I.4.2. Principaux termes de signal audio

I.4.2.1. L'enveloppe

L'enveloppe du son est son évolution dans le temps. On découpe cette évolution en quatre parties, Les quatre parties sont : *Attaque - Decay - Sustain - Release* :

- **Attaque**= c'est l'ampleur rejoint très rapidement sa valeur maximum.
- **Decay**= après, une partie de l'énergie initiale se perd et l'ampleur diminue.
- **Sustain**= l'ampleur maintient un niveau à peu près constant pendant un certain temps.
- **Release**= après, l'ampleur va recommencer à diminuer jusqu'à s'annuler.

Il est important de connaître ces quatre étapes pour comprendre la vie d'une note, d'un son. Elles se retrouvent souvent pour les réglages de périphériques, comme par exemple les compresseurs.

Ci-dessous on peut voir un exemple d'enveloppe *ADSR* où la forme d'onde d'un son est circonscrite par une courbe qui décrit le cheminement de l'ampleur ; ce qui, en mathématiques prend le nom **d'enveloppe.** Vue la symétrie de la forme d'onde, et aux fins d'une évaluation de l'enveloppe, n'est tenue en considération que la partie positive.

Dans cette figure, on peut observer que la partie initiale du son détient un nombre majeur de hautes fréquences qui deviennent ainsi les premières à s'éteindre. Généralement, dans la phase de suivante le contenu des hautes fréquences s'est atténué, alors que continuent à être présentes les fréquences basses.

Ci-après est décrit le son comme il est produit par la 5éme corde d'une guitare acoustique (La) et sa visualisation dans le domaine du temps.

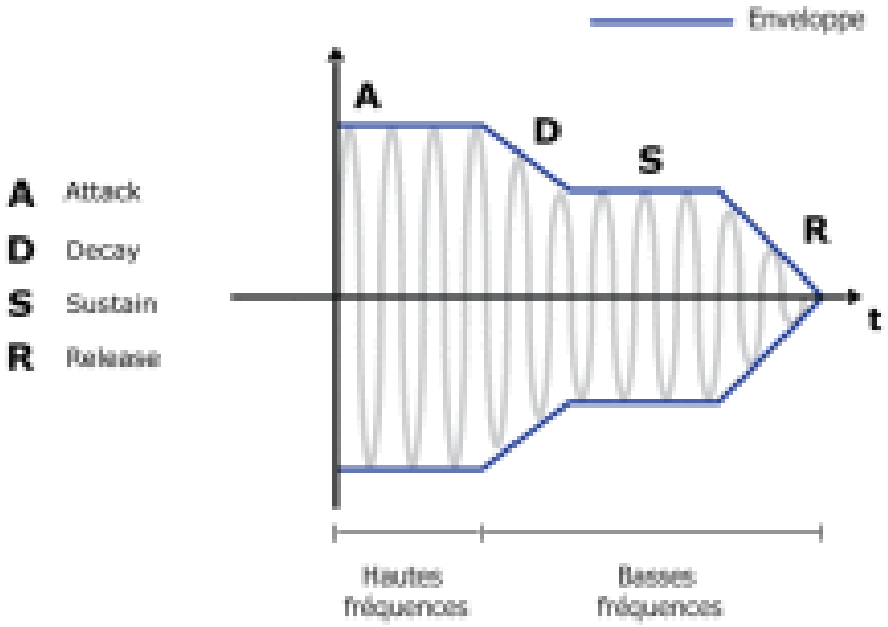


Figure I-3 : *L'enveloppe ADSR*

I.4.2.2. Bande Passante

La bande passante est une bande de fréquence comprise entre deux valeurs extrêmes. L'exemple le plus connu est la bande passante de l'oreille humaine qui est 20 Hz - 20000Hz. Ce sont les deux valeurs extrêmes des fréquences que l'on peut entendre. Si on parle de bande passante d'enceintes cela correspondra aux valeurs extrêmes des fréquences que cette dernière pourra émettre.

C'est important de bien comprendre ce qu'est une bande passante car beaucoup de matériel audio est défini par cette dernière. Pour un micro, la bande passante correspondra aux fréquences audios qu'il pourra reproduire électriquement. On parle aussi de bande passante pour les supports d'enregistrement. La bande passante n'est bien sûr pas utilisée qu'en son, on la retrouve dans beaucoup d'autres domaines.

I.4.2.3. Décibel

Le décibel est l'unité dont on se sert pour parler de puissance acoustique. Il sert aussi pour parler d'autre puissance mais nous verrons cela en temps voulu. Tout d'abord, il faut savoir que sa progression logarithmique, cela est dû à la perception de notre oreille. Le décibel est une unité de comparaison, on comparera toujours à une valeur connue. Dans le cadre de la puissance acoustique, traduisez volume du son, on parle de dBSPL (Sound Presure Level) l'unité de référence est le 0 Db SPL qui correspond à 2.10-5 Pa.

Cette valeur correspond en théorie au plus petit son que l'on peut entendre. Il faut savoir qu'il y a toujours autour de nous un bruit de fond d'au moins une vingtaine de décibel dans un endroit très calme. Cela est dû au son de notre propre corps, au bruit de la terre et à tous les autres parasites.

I.4.2.4. Dynamique

La dynamique s'exprime souvent en dB SPL, c'est la différence entre le niveau le plus fort et le bruit de fond. Par exemple : l'oreille humaine a une dynamique de 120 dB SPL. Le son le plus petit que l'on peut entendre est 0 dB SPL et au-delà de 130 dB SPL le son est douloureux et dangereux. La dynamique sert aussi à caractériser le matériel audio. [3]

I.5. Acquisition d'un signal

Acquérir un signal, c'est récupéré une information numérique ou analogique par un système : scanner, capteur…etc.

I.5.1. Chaine d'acquisition d'un signal audio

Une chaîne d'acquisition numérique peut se représenter selon la figure suivante :

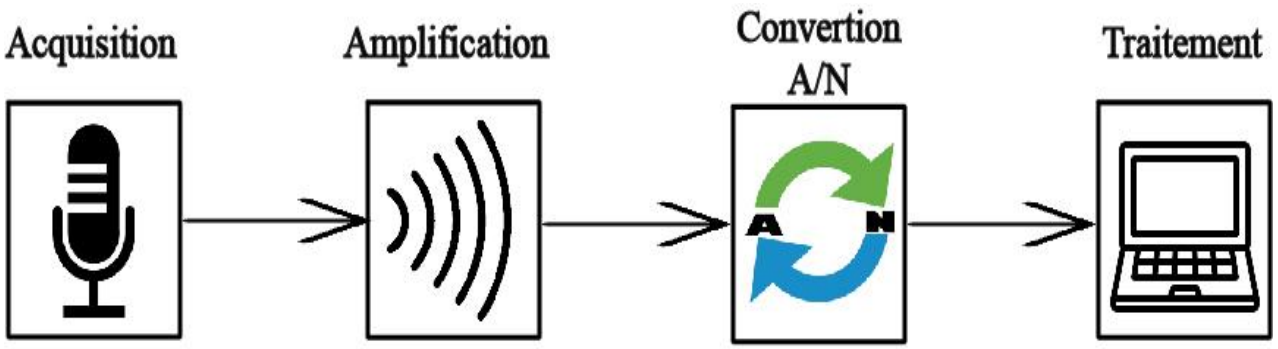


Figure I-4 : *Chaine d'acquisition d'un signal audio*

I.5.1.1. Capteur

Premier élément de la chaîne d'acquisition, le capteur a pour fonction de délivrer un signal électrique de sortie fonction du mesurande *m*, Autrement dit c'est l'interface entre le monde physique et le monde électrique. Il va délivrer un signal électrique image du phénomène physique que l'on souhaite numériser. Il est toujours associé à un circuit de mise en forme. Il est à noter que certains capteurs sont passifs (fonctionnent sans alimentation électrique) et d'autres actifs (nécessitent une alimentation électrique). [4]

I.5.1.1.1. Types des capteurs

Les capteurs ont des types diffèrent par rapport à leur classe et on va parler de quelques classes et types des capteurs

❖ Classification par apport génétique :

- ♦ **Capteur actif :** On parle de capteur actif lorsque le phénomène physique qui est utilisé pour la détermination du mesurande effectue directement la transformation en grandeur électrique. C'est la loi physique elle-même qui relie mesurande et grandeur électrique de sortie.
- ♦ **Capteur passif :** Ils ont besoin dans la plupart des cas d'apport d'énergie extérieure pour fonctionner. Ce sont des capteurs modélisables par une impédance.

❖ Classification par types de sortie :

- ♦ **Capteur analogique :** La sortie est une grandeur électrique dont la valeur est une fonction de la grandeur physique mesurée par le capteur.
- ♦ **Capteur numérique :** Ou capteurs TOR. La sortie est un état logique que l'on note 1 ou 0. La sortie peut prendre ces deux valeurs.

➢ Par grandeur physique mesurée

- ✓ Angle
- ✓ Contrainte
- ✓ Courant
- ✓ Champ magnétique
- ✓ Débit
- ✓ Déplacement
- ✓ Distance
- ✓ Force
- ✓ Inertiels
- ✓ Lumière
- ✓ Niveau
- ✓ Position
- ✓ Pression
- ✓ Son
- ✓ Température

I.5.1.2. Amplification

Lorsque les signaux électriques issus des capteurs sont de faible amplitude, il peut être nécessaire de les amplifier pour les adapter à la chaîne de transmission. Il faut savoir que l'amplification (en tension ou en puissance) du signal électrique issu du capteur est un phénomène bruyant : elle s'accompagne d'une dégradation du rapport signal sur bruit. Cela signifie que si l'amplitude du signal utile issue du capteur se trouve augmentée, les parasites (bruit) le sont également mais dans des proportions plus grandes encore [4] .

I.5.1.2.1. Le principe de fonctionnement

Un amplificateur électronique utilise un ou plusieurs composants actifs (transistor ou tube électronique) afin d'augmenter la puissance électrique du signal présent en entrée. Les composants actifs utilisés dans les amplificateurs électroniques permettent de contrôler leur courant de sortie en fonction d'une grandeur électrique (courant ou tension), image du signal à amplifier. Le courant de sortie des composants actifs est directement tiré de l'alimentation de l'amplificateur. Suivant la façon dont ils sont implantés dans l'amplificateur, les composants actifs permettent ainsi d'augmenter la tension et/ou le courant du signal électrique d'entrée.

Les amplificateurs peuvent être conçus pour augmenter la tension (amplificateur de tension), le courant (amplificateur suiveur) ou les deux (amplificateur de puissance) d'un signal. Les amplificateurs électroniques peuvent être alimentés par une tension simple (une alimentation positive ou négative, et le zéro) ou une tension symétrique (une alimentation positive, une négative et le zéro). L'alimentation peut aussi porter le nom de « bus » ou « rail ». On parle alors de bus positif ou négatif et de rail de tension positive ou négative.

Les amplificateurs sont souvent composés de plusieurs étages disposés en série afin d'augmenter le gain global. Chaque étage d'amplification est généralement différent des autres afin qu'il corresponde aux besoins spécifiques de l'étage considéré. On peut ainsi tirer avantage des points forts de chaque montage tout en minimisant leurs faiblesses.

Le principe de fonctionnement d'un amplificateur est présenté dans le schéma simplifié ci-dessous.
Les amplificateurs peuvent être conçus pour augmenter la tension (amplificateur de tension), le courant (amplificateur tampon ou suiveur) ou les deux (amplificateur de puissance) d'un signal. Les amplificateurs électroniques peuvent être alimentés par une tension simple (une alimentation positive ou négative, et la masse) ou une tension symétrique (une alimentation positive, une négative et la masse).

L'alimentation peut aussi porter le nom de « bus » ou « rail ». On parle alors de bus positif ou négatif et de rail de tension positive ou négative.

Les amplificateurs sont souvent composés de plusieurs étages disposés en série afin d'augmenter le gain global. Chaque étage d'amplification est généralement différent des autres afin qu'il corresponde aux besoins spécifiques de l'étage considéré. On peut ainsi tirer avantage des points forts de chaque montage tout en minimisant leurs faiblesses.

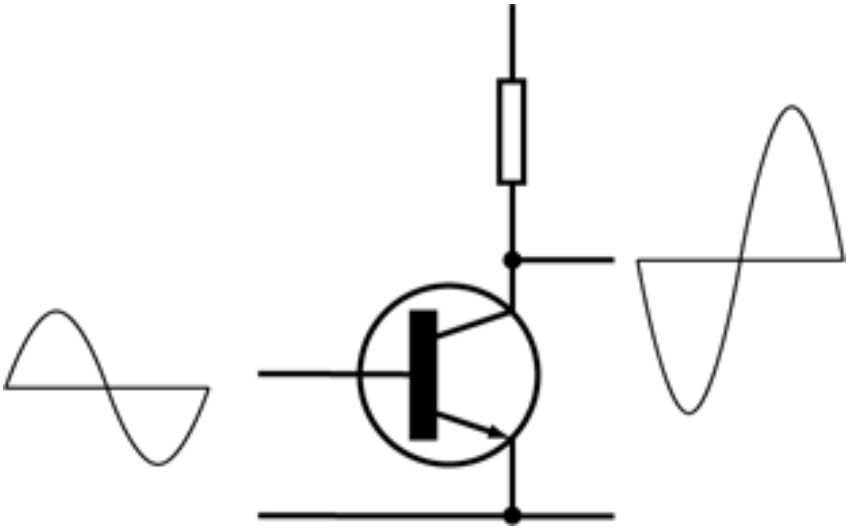


Figure I-5 : *Schéma de principe de fonctionnement d'un amplificateur*

I.5.1.2.2. Types d'amplificateurs

On distingue plusieurs types d'amplificateurs, on va parler que de deux, à savoir : l'amplificateur audio et l'amplificateur opérationnel.

- Amplificateur audio.

Un amplificateur audio est un amplificateur électronique conçu pour amplifier un signal électrique audio afin d'obtenir une puissance suffisante pour faire fonctionner un haut-parleur situé dans une enceinte acoustique ou un casque audio.

Figure I-6 : *Amplificateur à tube*

❖ **Amplificateur opérationnel**

Les amplificateurs opérationnels (aussi dénommé ampli-op ou ampli op, AO, AOP, ALI, AIL ou encore CIL) ont été initialement conçus pour effectuer des opérations mathématiques en utilisant la tension comme image d'une autre grandeur. C'est le concept de base des calculateurs analogiques dans lesquels les amplificateurs opérationnels sont utilisés pour modéliser les opérations mathématiques de base (addition, soustraction, intégration, dérivation…).

Cependant, un amplificateur opérationnel idéal est extrêmement souple d'utilisation et peut effectuer bien d'autres applications que les opérations mathématiques de base. En pratique, les amplificateurs opérationnels sont constitués de transistors, tubes électroniques ou de n'importe quels autres composants amplificateurs et ils sont implémentés dans des circuits discrets ou intégrés.

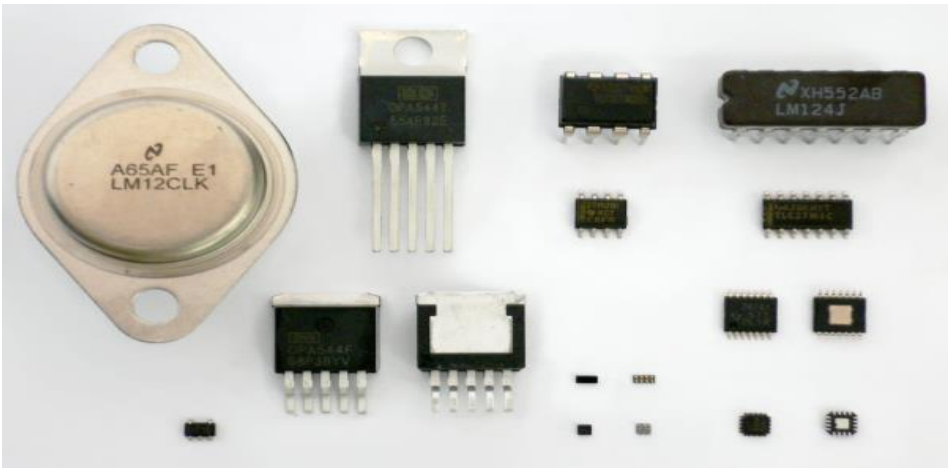


Figure I-7 : *Les amplificateurs opérationnels*

I.5.1.3. Conversion analogique numérique

La conversion analogique/numérique consiste à transformer la tension analogique (issue du capteur) en un code binaire (numérique) adapté à son exploitation dans un processus de régulation, de contrôle, de calculs ou encore de stockage. La conversion analogique/numérique n'est pas systématique, un stockage ou une régulation pouvant également être réalisés à partir de données analogiques.

Le Convertisseur Analogique Numérique (CAN) transforme le signal analogique, signal continûment variable pouvant prendre une infinité de valeurs, en un signal numérique, signal discontinu pouvant être représenté aux moyens de données binaires (0 et 1) La conversion analogique/numérique comporte deux étapes, L'échantillonnage et la conversion (Figure 4) [4].

I.5.1.4. Echantillonnage

L'échantillonnage consiste à représenter un signal analogique continu $s(t)$ par un ensemble de valeurs s (nT_e) avec n entier situées à des instants discrets espacés de T_e constante, appelée la **période d'échantillonnage**. Cette opération est réalisée par un circuit appelé « préleveur ou échantillonneur » symbolisé souvent par un interrupteur. Dans une première phase, nous pouvons faire l'hypothèse que cette durée de prélèvement du signal est très courte et négligeable.

En supposant que le système numérique ne réalise aucun traitement sur le signal enregistré, l'enchaînement des différents signaux dans une chaîne d'acquisition et de restitution de données par un système numérique est celui présenté sur la figure 6.1. Le signal analogique d'entrée V_e est échantillonné pour donner un signal discrétisé temporellement $V_e(nT_e)$.

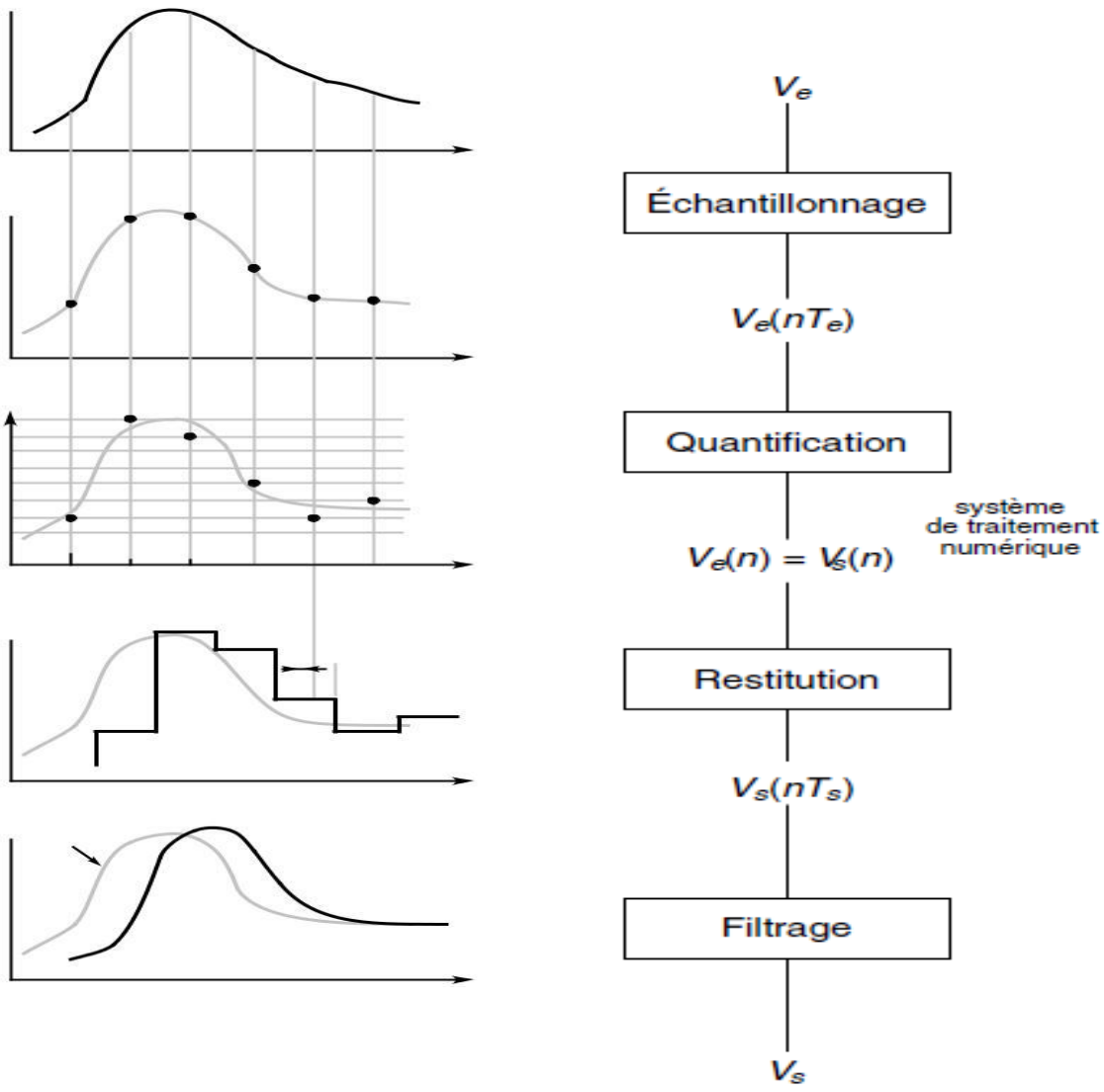


Figure I-8 : *Évolution d'un signal à travers une chaîne d'acquisition et de restitution de données sans modification des valeurs. Te est la période d'échantillonnage et Ts la période restitution supposée égale à Te*

Mais après cette phase de prélèvement d'échantillons, il est nécessaire de coder la donnée réelle obtenue dans un ensemble fini de valeurs : opération de quantification. Après cette quantification du signal, les différentes valeurs sont mémorisées dans le système numérique selon l'ordre de leurs arrivées, formant ainsi une suite de valeurs numériques $V_e(n)$. Si, comme supposé, le système numérique n'effectue aucun traitement sur ces valeurs $V_e(n) = V_s(n)$; cette suite de nombres $V_s(n)$ est envoyée vers le procédé externe en deux étapes : restitution de la valeur analogique $V_s(nT_s)$ et ensuite filtrage de ce signal pour obtenir un signal de sortie V_s sans fronts raides. [2]

I.5.1.5. Quantification

L'opération de quantification consiste à attribuer un nombre binaire à toute valeur prélevée au signal lors de l'échantillonnage. C'est le CAN (convertisseur analogique numérique) qui réalise cette opération. Chaque niveau de tension est codé sur p bits, chaque bit pouvant prendre deux valeurs (0 ou 1). Donc un convertisseur à p bits possède 2p niveaux de quantification. Considérons un CAN 4 bits, il n'y a donc que $2^4 = 16$ valeurs possibles attribuables à toutes les valeurs prélevées lors de l'échantillonnage.

L'opération se fait donc avec une perte d'information d'autant plus grande que p est petit. Le schéma ci-dessous représente une partie de la caractéristique de transfert d'un convertisseur 4 bits ; à tous les niveaux de tension d'un même palier, le convertisseur fait donc correspondre un seul et même nombre binaire : [5] .

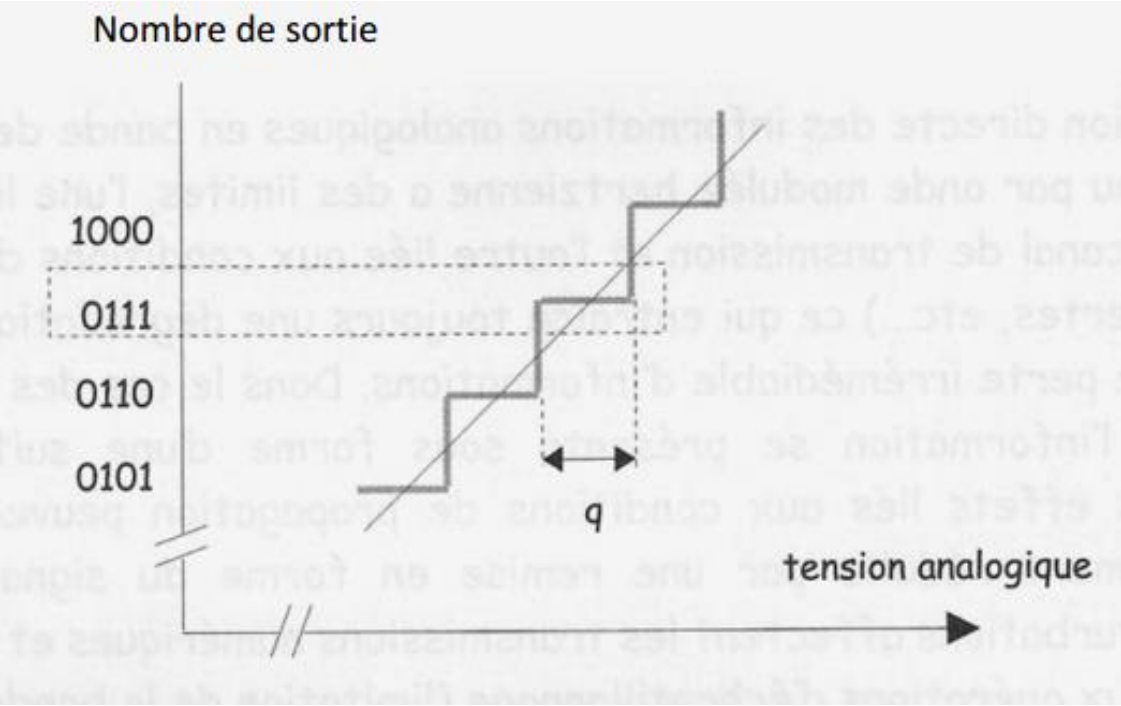


Figure I-9 : *Caractéristique de transfert d'un CAN – Quantification à 4 bits*

q c'est le pas de quantification : il correspond à la plus petite variation de tension que le convertisseur peut coder. On voit bien que plus q est faible, meilleure sera la précision de codage.

I.5.1.6. Outil de conversion A/N (Arduino)

Arduino est une carte électronique open source qui peut etre utilisé par n'importe quel utilisateur à cause de son matériel et logiciel qui sont faciles à utiliser. Avec Arduino il est possible de fabriquer des robots, de gérer des caméras, de commander des moteurs, d'alimenter automatiquement une plante au bout d'un laps de temps *x* etc.

ARDUINO = 1 carte à microcontrôleur (**Figure 5**) + 1 outil de développement (**Figure 6**) + 1 communauté active (Figure 7) [6] [7].

❖ **La carte Arduino**

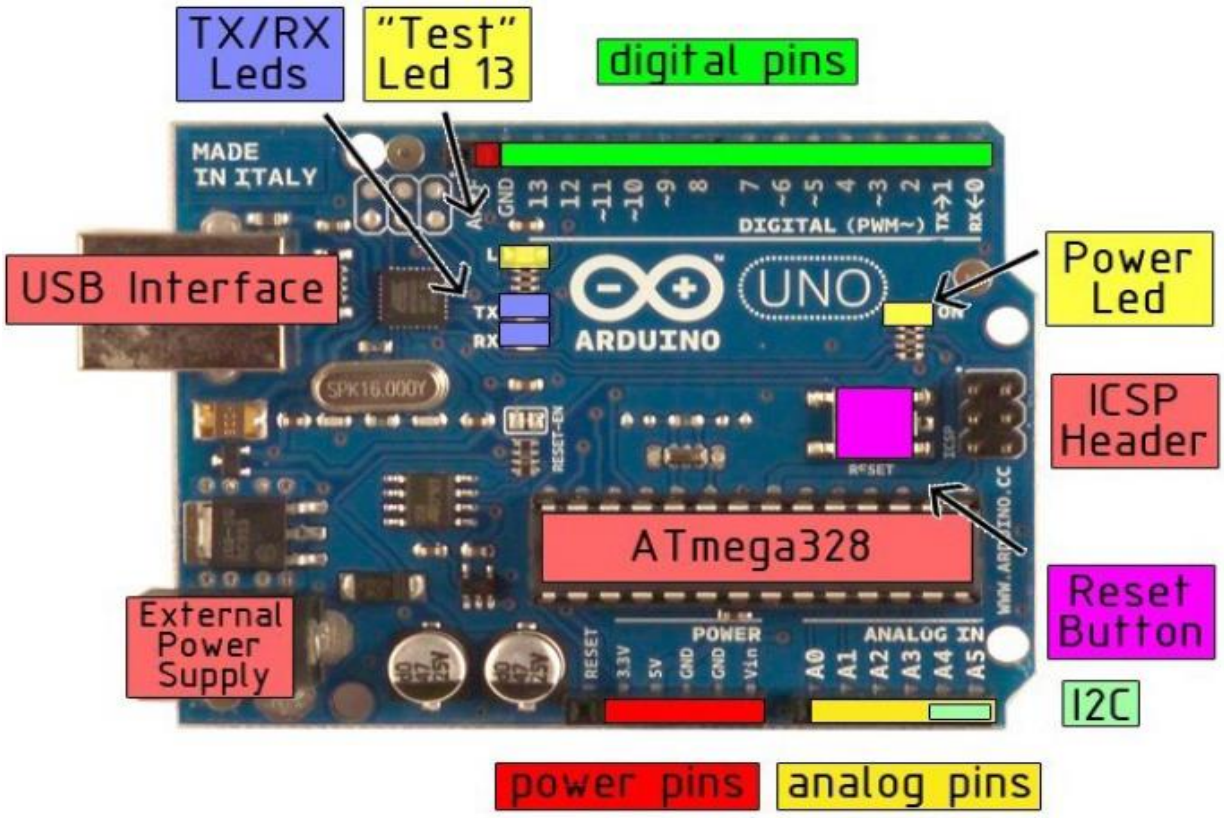


Figure I-10 : La carte Arduino et ses objets

❖ **Développement d'un projet**

Le développement sur Arduino est très simple :

- On code l'application : Le langage Arduino est basé sur les langages C/C++, avec des fonctions et des librairies spécifiques à Arduino (gestions des e/s).
- On relie la carte Arduino au PC et on transfert le programme sur la carte.
- on peut utiliser le circuit.

Le logiciel de programmation des modules Arduino est une application Java multiplateformes (fonctionnant sur tout système d'exploitation), servant d'éditeur de code et de compilateur, et qui peut transférer le firmware (et le programme) au travers de la liaison série (RS232, Bluetooth ou USB selon le module).

Le logiciel est très simple à prendre en main, il existe de très bons tutoriaux très bien faits avec même des explications en français. De très nombreux exemples sont fournis.

Les fichiers exemples sont disponibles et permettent de coder des programmes très compliquées sans trop d'efforts. Les bibliothèques fournies permettent d'utiliser des composants complexes très simplement en quelques lignes très claires (afficheur ou liaison SPI etc.). [7]

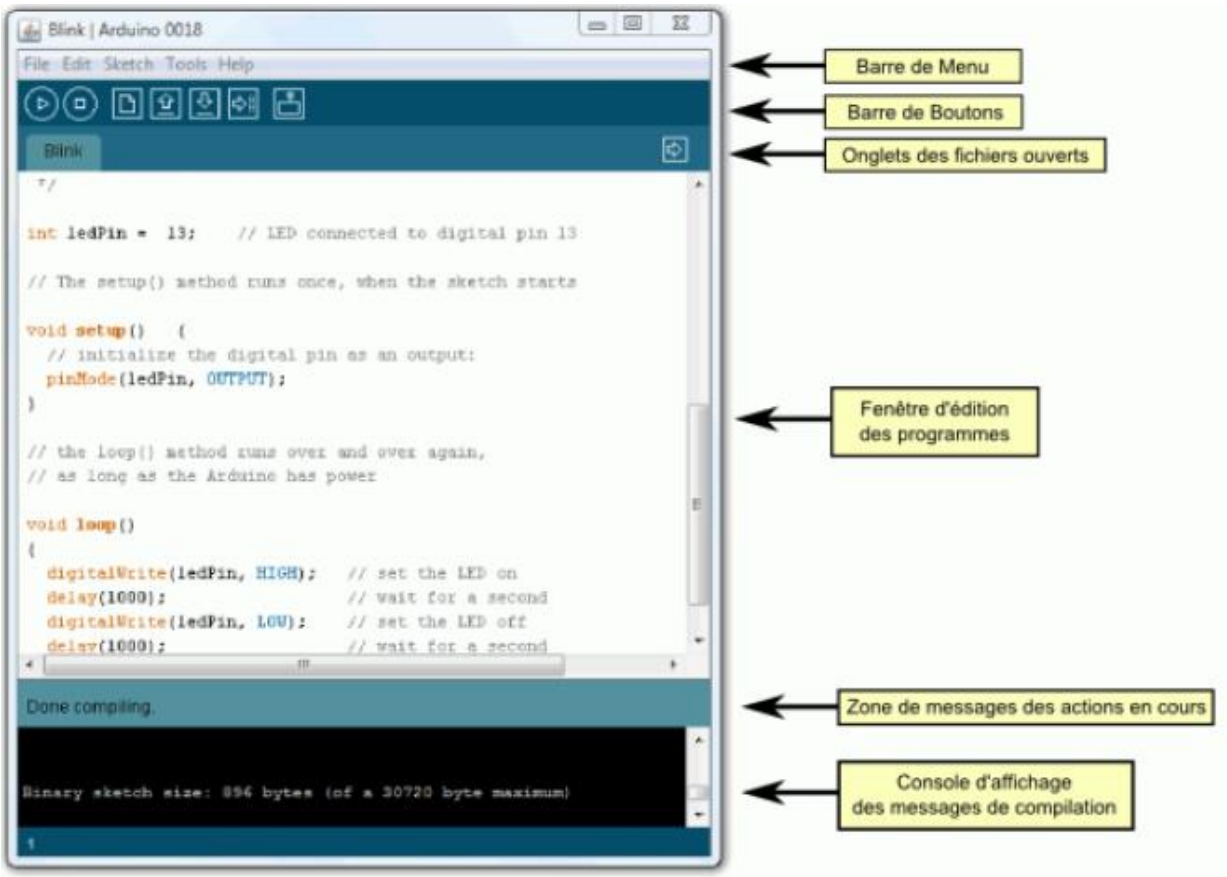


Figure I-11 : Outil de développement pour Arduino (l'écran principal du logiciel Arduino)

I.5.1.7. Traitement

Le traitement est basé sur la comparaison entre le signal de référence et les autres signaux qui entrent dans le système par le biais d'un capteur

I.5.1.7.1. Le signal de référence

Un signal de référence c'est un signal de base (repère) où on compare les signaux entrés dans le système avec ce signal qui est la référence et la base de comparaison et la relation entre les signaux.

I.6. Conclusion

Duran ce chapitre nous avons présenté le domaine de traitement de signal qui est devenu quelque chose primordiale dans le domaine industriel pour faciliter l'intervention humaine dans divers domaines

médicale, chimique et beaucoup d'autres domaine. Et aussi nous avons présenté des définitions sur les termes de traitement de signal ; Les types des signaux et surtout le signal audio ; La chaine d'acquisition d'un signal ; et des définitions de quelque outil utilisé dans la chaine d'acquisition. Cette synthèse bibliographique nous a permis de comprendre le fonctionnement et le principe de traitement de signal pour développer notre réflexion. Le chapitre suivant présente l'analyse de notre système d'étude et les étapes de modélisation pour automatiser le processus actuel.

Bibliographie

Chapitre I

[1] R.Boite, «Traitement de al parole,» *press polytechnique et universitaire romande,* Novembre 1999.

[2] F. Cottet, AIDE-MÉMOIRE TRAITEMENT DU SIGNAL, Paris, 2005.

[3] sahliano, «Le Signal Audio,» 29 AVRIL 2009. [En ligne]. Available: https://www.scribd.com/doc/14758771/Le-Signal-Audio.

[4] H. H. M. -. E. V. –. J.-P. BARBOT, «Capteurs et chaîne d'acquisition,» 08 06 2015. [En ligne]. Available: http://eduscol.education.fr/sti/si-ens-cachan/.

[5] T. GERVAIS, «INTRODUCTION A L'ELECTRONIQUE NUMERIQUE,» [En ligne]. Available: https://www.lycee-champollion.fr/IMG/pdf/quantification_-_echantillonnage_-_cours.pdf.

[6] O. Adruino, «Introduction,» [En ligne]. Available: https://www.arduino.cc/en/guide/introduction.

[7] L. REYNIER, «C'est quoi Arduino,» [En ligne]. Available: http://www.louisreynier.com.

[8] s. w. o. d'Arduino, «Introduction,» [En ligne]. Available: https://www.arduino.cc/en/guide/introduction.

[9] h. nouredine, Écrivain, *modelisation et uml.* [Performance].

[10] N. Gicquiaud, *Les organigrammes.*

[11] L. Duzart, «Organigramme d'entreprise,» [En ligne]. Available: http://www.liliandauzat.com/gestion-entreprise/organigramme-entreprise/. [Accès le 09 06 2017].

[12] F. Balena, «Microsoft Visual Basic 2005,» chez Microsoft Visual Basic 2005, Paris, Microsoft press, 2006, p. 640.

[13] M. Halvorson, Microsoft Visual Basic 2010, Paris: Miscrosoft press, 2010.

[14] «Product Family Life Cycle Guidelines for Visual Basic 6.0».

II. Chapitre II Analyse et Modélisation de système

II.1. Introduction

Le « cycle de vie d'un logiciel » (en anglais *software lifecycle*), désigne toutes les étapes de développement d'un logiciel, dès la conception jusqu'à la disparition. L'objectif d'un découpage est de permettre de définir des jalons intermédiaires permettant la validation du développement logiciel, c'est-à-dire la conformité du logiciel avec les besoins exprimés et la vérification du processus de développement, c'est-à-dire l'adéquation des méthodes mises en œuvre.

- Expression des besoins
- Spécification
- Analyse
- Conception
- Implémentation
- Test et vérification
- Validation
- Maintenance et évolution

Et on va parler de tout ça et faire une modélisation pour notre système pour le rendre fiable

II.2. Modélisation

La modélisation est la conception d'un modèle. Selon son objectif et les moyens utilisés, la modélisation est dite mathématique, géométrique, 3D, mécaniste (ex : modélisation de réseau trophique dans un écosystème), cinématique... Elle nécessite généralement d'être calée par des vérifications *in situ*, lesquelles passent par le paramétrage et le calibrage des « *modèles* » utilisés.

II.2.1. Principe

Un modèle est une abstraction permettant de mieux comprendre un objet complexe (bâtiment, économie, logiciel, …).

II.2.2. Autre principe

Un petit dessin vaut mieux qu'un long discours.

Les modèles sont donc souvent graphiques, même si l'objet à créer n'est pas matériel.

II.2.3. La modélisation informatique

La construction d'un système d'information, d'un réseau, d'un logiciel complexe, de taille importante et par de nombreuses personnes oblige à modéliser.

Le modèle d'un système informatique sert :

- De document d'échange entre clients et développeurs
- D'outil de conception
- De référence pour le développement
- De référence pour la maintenance et l'évolution

II.2.4. Intérêt informatique de la modélisation

- Les modèles aident à visualiser un système existant ou futur (tel que l'on souhaite qu'il devienne)
- Les modèles permettent de spécifier la structure et le comportement d'un système
- Les modèles documentent les choix effectués

II.2.5. Langages de modélisation

Un langage de modélisation doit définir :

La sémantique des concepts ;

Une notation pour la représentation de concepts ;

Des règles de construction et d'utilisation des concepts.

Des langages à différents niveaux de formalisation

Langages formels (Z,B,VDM) : le plus souvent mathématiques, au grand pouvoir d'expression et permettant des preuves formelles sur les spécifications

Langages semi-formels (MERISE, UML...) : le plus souvent graphiques, au pouvoir d'expression moindre mais plus faciles d'emploi.[9]

II.2.6. Construction d'applications

La construction d'une application informatique se résume à réaliser du code pour répondre au besoin d'un utilisateur, aussi appelé client.

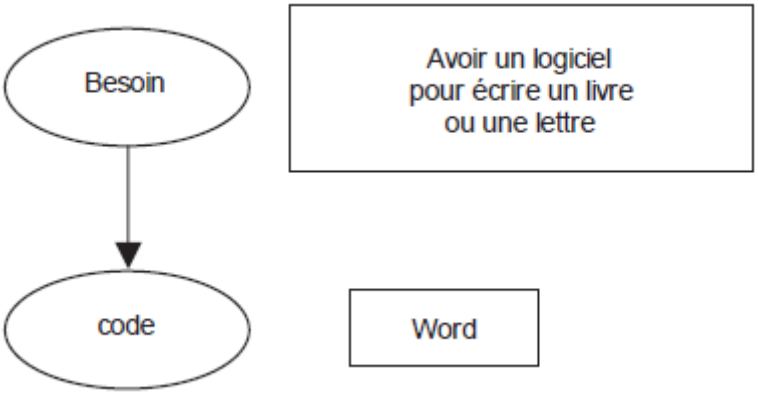


Figure II-1 : *Simplification extrême de la réalisation d'une application informatique*

La finalité de l'activité de construction d'applications informatiques, est de réaliser le code, mais est-ce que c'est suffisant.[9]

II.2.7. Etapes de réalisation d'un système informatique

Figure II-2 : *Les étapes de réalisation d'un système informatique*

Expression des besoins

- ✓ Définition d'un cahier des charges (Interview de l'expert métier)

Spécification

- ✓ Ce que le système doit être et comment il peut être utilisé

Analyse

- ✓ Éléments intervenant dans le SI, leurs structures et relations
- ✓ Elle doit décrire chaque objet selon 3 axes :

1. Savoir-faire de l'objet : axe fonctionnel (précise les fonctions réalisées par les objets)
2. Structure de l'objet : axe statique
3. Cycle de vie de l'objet : axe dynamique (ces états, les évènements déclenchant. Ces Chang.)

Conception

- ✓ Apport de solutions techniques : Architecture, Performance et optimisation.
- ✓ Définition des structures et des algorithmes.

Implémentation

- ✓ Réalisation et programmation.

Tests et vérification

- ✓ Contrôles de qualité.
- ✓ Instaurés tout au long du cycle de développement.

Validation

- ✓ Vérification de la correspondance avec le cahier des charges + discussion avec users.

Maintenance et Evolution

- ✓ Maintenance corrective : traiter les erreurs (bugs).
- ✓ Maintenance évolutive : intégration de nouveaux changements.[9]

II.3. Future système

II.3.1. Objectif

L'objectif de notre futur système est d'acquérir un signal audio à partir d'un microphone et le faire entrer sur un ordinateur et le comparer avec la référence qui est déjà enregistré dans l'ordinateur sur une base de donnée et les comparer et prendre des décisions si les sons sont pareils ou non avec un certain pourcentage.

II.3.2. Vue générale de notre système

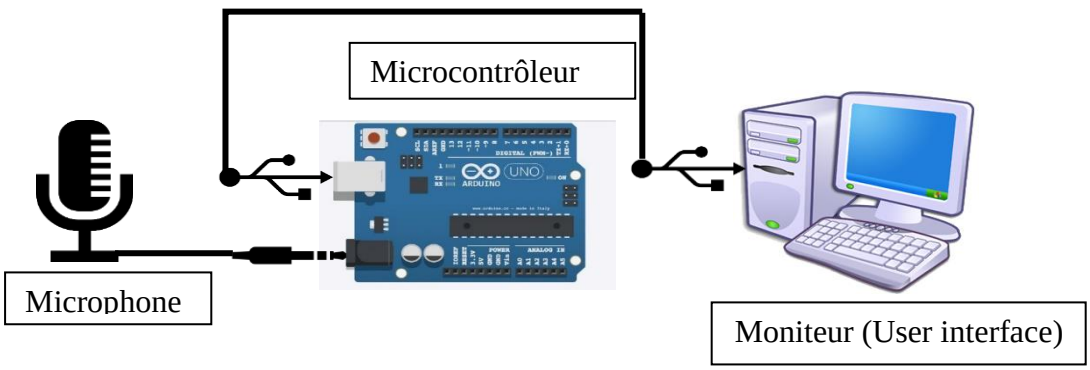


Figure II-3 : *Vue générale du système*

Le système est basé sur trois composants principaux : ce sont le microphone et le microcontrôleur et l'ordinateur.

Le signal est acquit par le microphone et il est transféré à carte Arduino avec un câble jack auxiliaire et converti dans la carte puis il est transmis vers l'ordinateur avec un câble USB pour faire le traitement dans l'ordinateur à l'aide des outils informatique qui sont installer sur ce dernier pour que le développeur peut manipuler avec une interface.

II.3.3. L'architecture d'utilisation du système

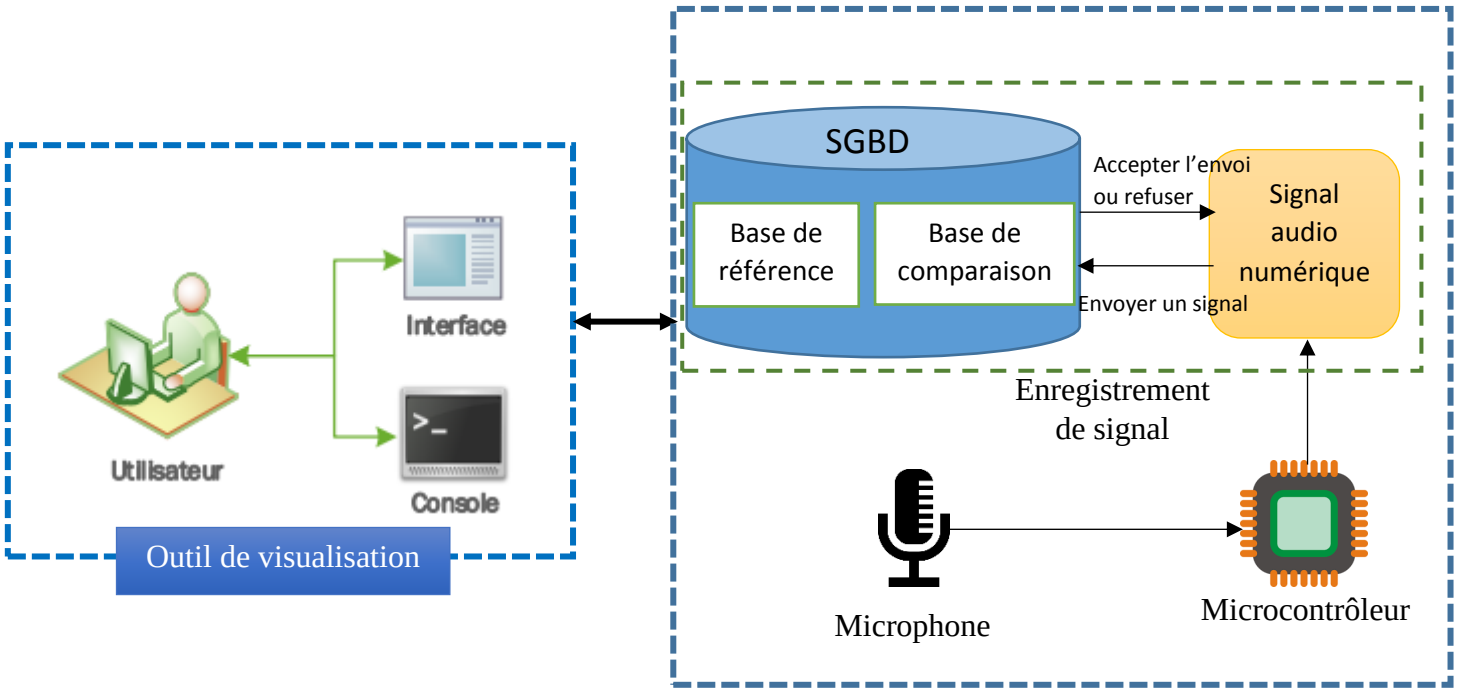


Figure II-4 : *Architecture d'utilisation du système*

L'utilisation du système est basée sur une interface graphique qui va faciliter le travail pour l'utilisateur de système.

L'utilisation commence par entrer un son avec le microphone mais avant ca il faut lancer au système qu'un son est en route pour que le système peut sauvegarder et traiter, et il faut choisir quel type de signal on veut l'entrer dans le système de base de donnée et ce signal va passer par un microcontrôleur qui va convertir le son et fait un traitement complet avant de transférer ce signal à l'ordinateur et toutes ces étapes doivent se faire en temps réel.

Le signal acquit vas enregistrer dans une des bases de données soit la base de référence ou base de comparaison aux choix de l'utilisateur qui va choisir dans quelle base de donnée il va enregistrer le signal.

II.3.4. Modélisation de notre système par UML

II.3.4.1. Représentation de modélisation avec UML

UML (Unified Modeling Language ou « langage de modélisation unifié »)
Un modèle est une représentation simplifiée d'un problème. UML permet d'exprimer les modèles objets à travers un ensemble de diagrammes. Ces derniers sont des moyens de description des objets ainsi que des liens qui les relient. Il est apparu dans le monde du génie logiciel, dans le cadre de la « conception orientée objet ». Couramment utilisé dans les projets logiciels, qu'il peut être appliqué à toutes sortes de systèmes et ne se limitant pas au domaine informatique.

II.3.4.2. Spécification du système

Parmi les diagrammes UML largement connus on cite :

- Diagramme de cas d'utilisation : il permet de recueillir, d'analyser et d'organiser les besoins. Avec lui débute l'étape d'analyse de notre système.
- Diagramme de séquence : il permet de représenter des interactions entre objets et acteurs, selon un point de vue temporel avec une chronologie des envois de messages, c'est un type de diagramme d'interaction.

II.3.4.3. Identification des acteurs

Dans notre système on distingue un seul acteur qui est l'administrateur. La figure suivante illustre le diagramme contexte dynamique

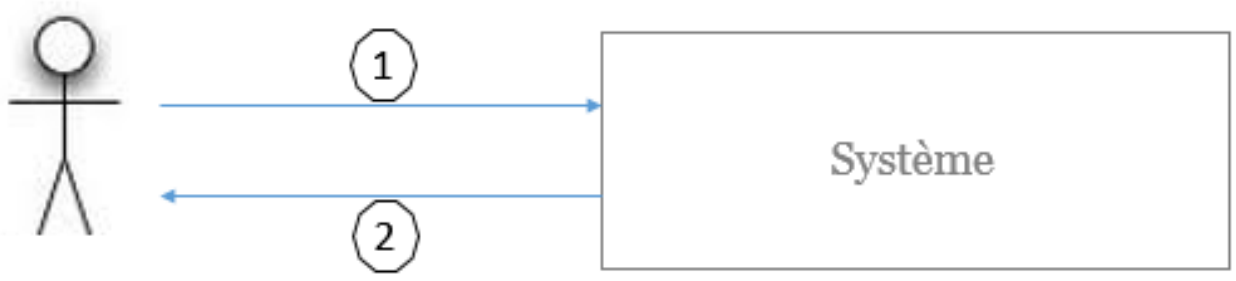


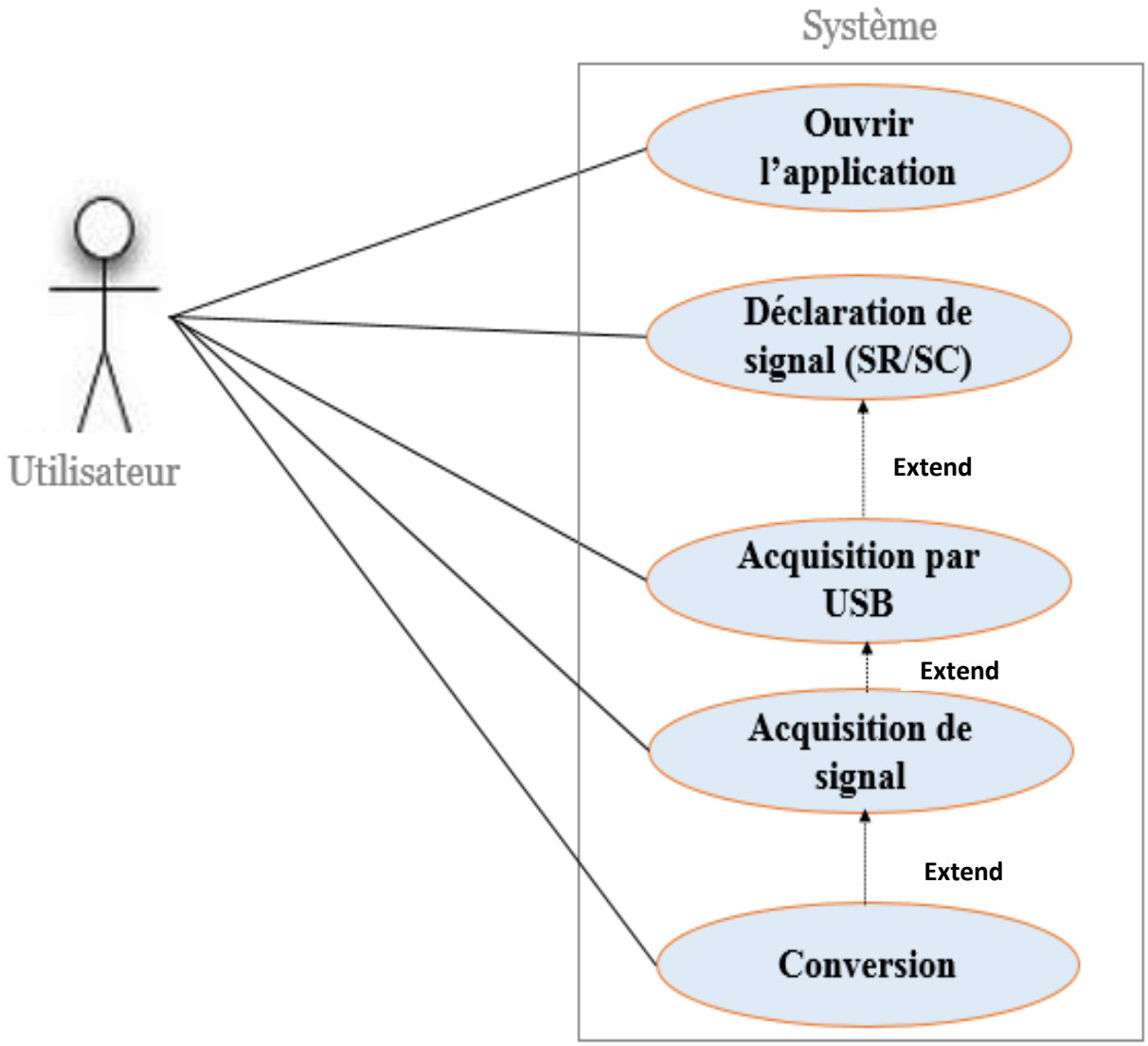
***Figure II-5** : Diagramme de contexte*

II.3.4.4. Les cas d'utilisation

Les cas d'utilisation constituent une technique qui permet de déterminer les besoins des utilisateurs et de capturer les exigences fonctionnelles d'un système. En d'autres termes, ils décrivent le comportement d'un système du point de vue de ses utilisateurs. Ils décrivent les interactions entre les utilisateurs d'un système et le système lui-même.

Dans la figure suivante nous avons présenté le diagramme de cas d'utilisation pour la compréhension du fonctionnement du système.

***Figure II-6** : Diagramme de cas d'utilisation*



II.3.4.5. Diagramme de séquence

Les diagrammes de séquences permettent de représenter les interactions entre objets selon un point de vue temporel. L'accent est mis sur la chronologie des envois de messages.

Dans cette partie, nous allons présenter différents diagrammes de séquence comme suit :

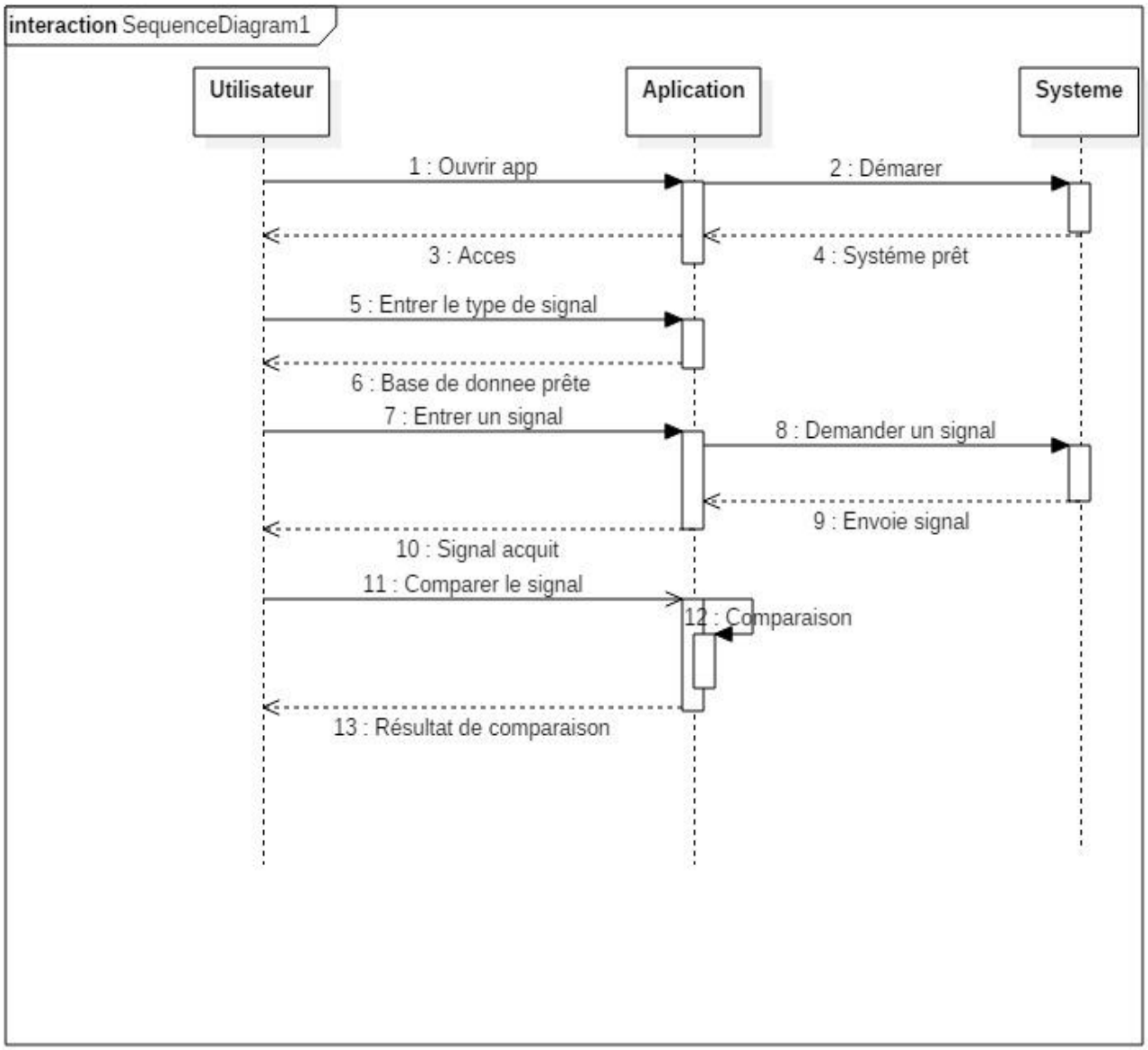


***Figure II-7** : Diagramme de séquence*

II.3.5. Modélisation avec organigramme

II.3.5.1. Définition

L'organigramme est une représentation schématique des liens fonctionnels, organisationnels et hiérarchiques d'une entreprise. Il sert ainsi à donner une vue d'ensemble de la répartition des postes et fonctions au sein d'une structure. Cette cartographie simplifiée permet de visualiser les différentes relations de commandement ainsi que les rapports de subordination d'où une vision simple et claire des structures complexes.

II.3.5.2. Rôle de l'organigramme

Une entreprise commerciale ou industrielle, les ministères publics, les hôpitaux, les agences de communication et diverses autres structures recourent à l'organigramme pour organiser efficacement la multiplicité des tâches requises à leurs fonctions.

En fonction du secteur d'activité, l'organigramme peut être établi sous diverses formes (trèfle, château, pyramide) pour un résumé graphique d'une organisation. On observe souvent les différents postes d'une entreprise représentés par des rectangles liés entre eux par des traits. Les traits verticaux indiquent les rapports de commandement et de subordination tandis que les traits horizontaux représentent des rapports d'égalité et de complémentarité. Il arrive que les postes à responsabilité soient représentés par des ovales. En somme, le but reste de donner en un clin d'œil une vue d'ensemble de l'organisation, laissant apparaître les détenteurs de pouvoirs et les chefs responsables.

II.3.5.3. Utilité de l'organigramme pour un projet

Pour l'élaboration d'un projet ou d'un programme, l'organigramme permettra de définir de manière brève toutes les fonctions, ce qui aidera dans la répartition efficace des tâches et le suivi des missions. Ainsi, la multiplicité des tâches au sein d'une équipe est régie de manière rigoureuse et l'organigramme devient un outil de travail. Les relations de commandement étant bien précisées, il n'y a pas de risque d'ambiguïté possible, ce qui facilite la compréhension, la communication et la progression de tous les collaborateurs.

II.3.5.4. Types d'organigramme

- **L'organigramme en pyramide (ou en bannière ou linéaire ou de travail)**

Il est très répandu **;** les éléments ayant le même niveau sont sur une même ligne horizontale**,** le plus élevé est en haut.

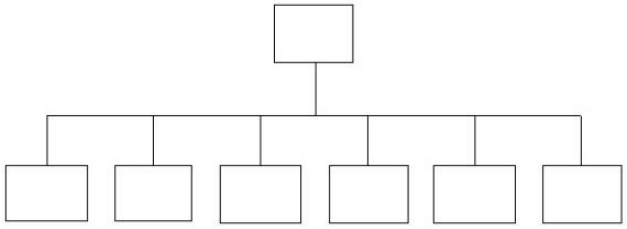


Figure II-8 *: Organigramme en pyramide*

- **L'organigramme en arbre**

Les éléments de même niveau hiérarchique sont sur la même ligne verticale **;** le plus élevé est à gauche.

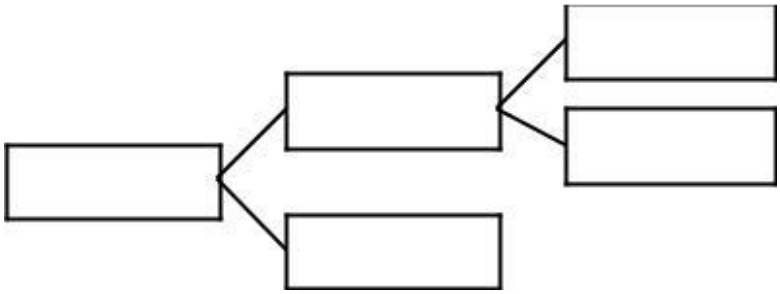


Figure II-9 : *Organigramme en arbre*

L'organigramme replié

Il associe la présentation de l'organigramme en pyramide et de l'organigramme en arbre ; les cartouches sont décalés du haut vers le bas et de gauche à droite.[10]

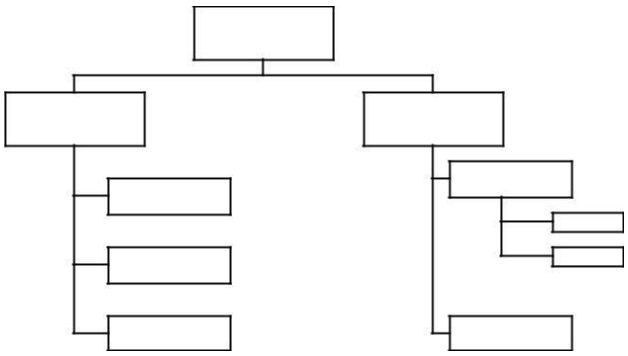


Figure II-10 : *Organigramme replié*

II.3.5.5. Comment créer un organigramme ?

Il existe des logiciels spécialisés élaborés pour faciliter la tâche aux concepteurs d'organigramme. Certaines sociétés informatiques mettent à la disposition du grand public et des entreprises des produits spécifiques pour la création simplifiée d'organigrammes. Il est cependant important de ne pas faire d'amalgame entre les différents organigrammes (programmation, gestion, produit…).

En somme, l'organigramme est sans conteste un outil de travail garantissant l'efficacité du personnel et permettant la révélation de la culture et de la stratégie de l'entreprise. Un organigramme issu d'une vision commune de l'organisation assure ainsi la multiplicité des compétences[11]

II.4. Synthèse sur le fonctionnement globale de notre système

Notre système a une synthèse de fonctionnement et passe par des étapes pour finir le traitement de notre signal et pour mieux comprendre le système on a tout représenter dans l'illustration suivante :

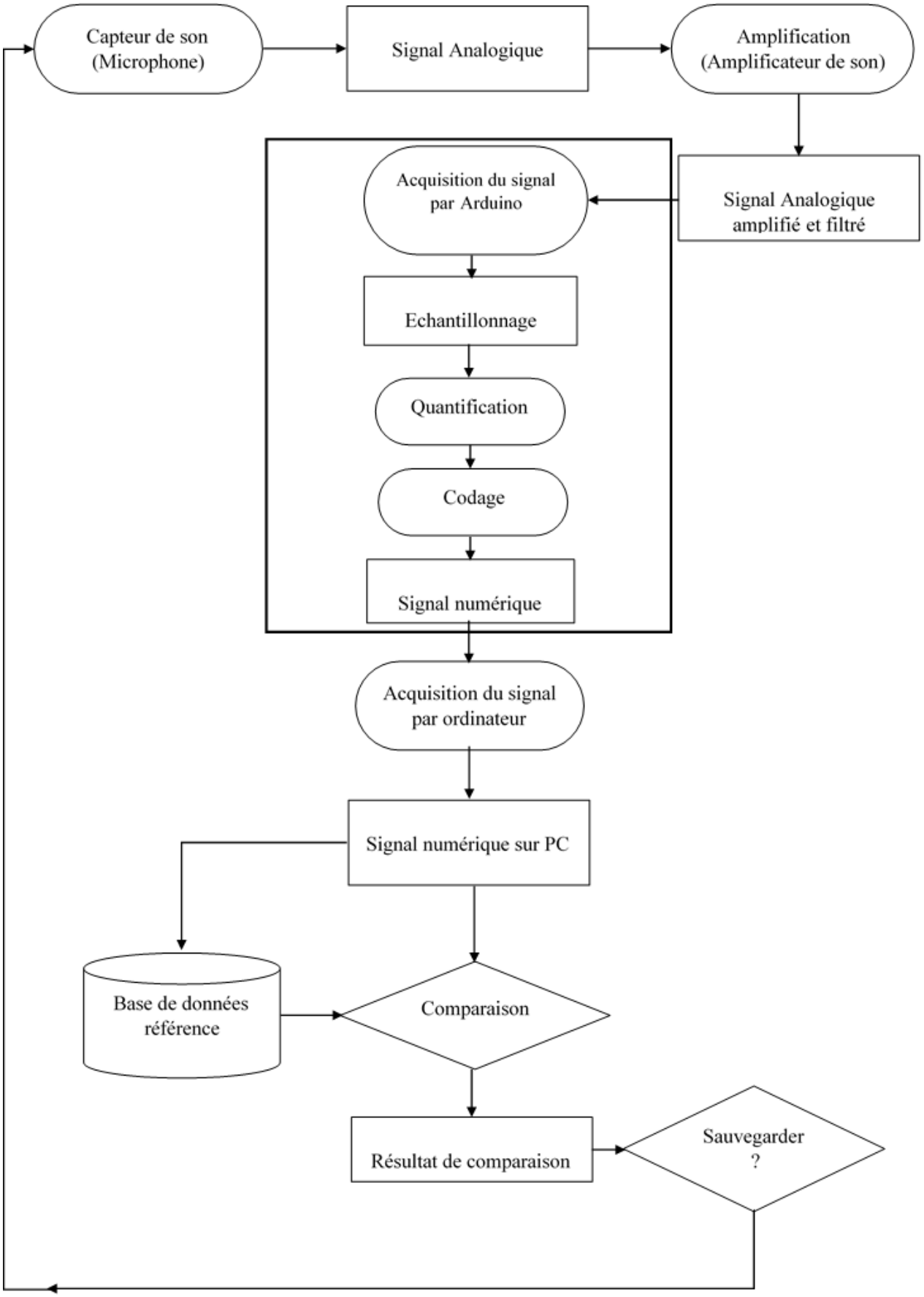


Figure II-11 : *Organigramme de l'architecture du système*

II.4.1. Description de la synthèse de fonctionnement globale

Le son est détecté par un capteur de son, dans ce cas c'est un microphone qui est l'interface entre le monde physique et le monde électrique.

Le microphone va délivrer un signal électrique que l'on va numériser.
Il est toujours associé à un circuit de mise en forme.

A la fin de cette étape on va avoir un signal analogique brut ; ce signal est un signal faible, on va l'amplifier pour pouvoir le traiter. Dans cette étape, on a besoin d'un amplificateur de son pour pouvoir adapter le niveau du signal issu du capteur à la chaîne globale d'acquisition.

Après ça, on filtre le son et on l'échantillonne aussi pour le convertir en signal numérique. Le rôle de filtrage est de limiter le contenu spectral du signal aux fréquences qui nous intéressent.

Ainsi on élimine les parasites. On va utiliser un filtre passe bas que l'on caractérise par sa fréquence de coupure et son ordre. Apres ça on fait la conversion analogique numérique.

Cette étape consiste à le découper doublement ; d'abord et selon l'axe des temps, qu'on nomme échantillonnage et la période d'échantillonnage l'épaisseur des tranches de temps. Pour le signal audio, c'est la mesure périodique de son intensité, au rythme de la période d'échantillonnage.

On nomme échantillons les valeurs mesurées aux instants d'échantillonnage.

Après l'échantillonnage on fait l'étape de quantification, cette étape consiste à approximer les valeurs réelles des échantillons selon une échelle de n niveaux appelée échelle de quantification. Alors la quantification consiste pour chaque échantillon, à lui associer une valeur d'amplitude. Dans l'étape suivante on doit exprimer cette valeur d'amplitude en bit. L'action de transformer la valeur numérique de l'amplitude en valeur binaire s'appelle le codage.

Maintenant on a fini la conversion analogique numérique et on a un signal numérique représenté en binaire.

L'étape suivante qui est le traitement et la comparaison de deux signaux audio. Ça consiste à prendre un signal brut et un signal de référence, qu'on a acquis au préalable, pour comparer les deux.

Pour cette étape on a besoin d'un système de comparaison en temps réel et deux bases de données où on stocke les résultats de la comparaison.

Pour le système de comparaison, on va utiliser une plateforme de programmation où on va écrire et exécuter notre programme.

Pour la base de données, on a besoin d'un outil de traitement de base de données, pour sauvegarder et extraire les signaux et les informations.

La première base de données, c'est la base de référence où on met les signaux de référence qui viennent de notre système d'acquisition pour pouvoir extraire et comparer avec les autres signaux qui viennent du même système et pour la base de comparaison on va mettre les signaux qu'on veut comparer avec le signal de référence.

Après ça, notre programme de comparaison va décider si les signaux sont similaires ou différents.
Enfin, le système nous laisse le choix de décider si on veut sauvegarder le résultat de comparaison ou le rejeter et entamer une nouvelle opération d'acquisition.

II.5. Conclusion

Dans ce chapitre nous avons spécifié les besoins de notre futur système, et nous avons modélisé notre système avec le langage UML. Dans notre modélisation nous avons utilisé quelque diagramme ; diagramme de cas d'utilisation et diagramme de séquence.

Le fonctionnement global de notre système est synthétisé par un schéma global qui décrit les opérations, les actions, le contrôle et les composants.

Dans le chapitre suivant nous allons présenter l'implémentation et la mise en œuvre de notre système et la simulation de notre système.

III. Bibliographie

[1] h. nouredine, Écrivain, *modelisation et uml.* [Performance].

[2] N. Gicquiaud, *Les organigrammes.*

[3] L. Duzart, «Organigramme d'entreprise,» [En ligne]. Available: http://www.liliandauzat.com/gestion-entreprise/organigramme-entreprise/. [Accès le 09 06 2017].

III. Chapitre III Simulation et réalisation du Système

III.1. Introduction

Après avoir terminé la modélisation de notre système, nous passons maintenant à la quatrième activité du processus de notre travail, il s'agit de l'étape de simulation que nous résumons dans ce chapitre.

Le chapitre est composé de trois parties principales. Nous présentons dans la première les outils matériels et logicielle et technologies utilisés pour implémenter notre système. Nous parlons dans la deuxième partie des architectures de notre outil. Finalement, nous présentons dans la dernière partie les principales fonctionnalités de notre application.

III.2. Simulation

La simulation est un outil utilisé par le chercheur, l'ingénieur, le militaire, etc. pour étudier les résultats d'une action sur un élément sans réaliser l'expérience sur l'élément réel.

Lorsque l'outil de simulation utilise un ordinateur on parle de simulation numérique. Il a également existé des simulateurs analogiques et il a été envisagé dans les années 1970 d'en construire des stochastiques.

Les chercheurs, les ingénieurs, les militaires et bien d'autres professionnels se posent souvent la question : quel est le résultat que j'obtiens si j'exerce telle action sur un élément ?

Le moyen le plus simple serait de tenter l'expérience, c'est-à-dire d'exercer l'action souhaitée sur l'élément en cause pour pouvoir observer ou mesurer le résultat. Dans de nombreux cas l'expérience est irréalisable, trop chère ou contraire à l'éthique. On a alors recours à la simulation : rechercher un élément qui réagit d'une manière semblable à celui que l'on veut étudier et qui permettra de déduire les résultats.

III.3. Environnement de développement

Pour développer notre système nous avons utilisé un pc avec le système d'exploitation Windows 7 SP1.

Le PC c'est un laptop ASUS ; Processeur I5 ; RAM 6 GO ; Disque dur 500 GO.

III.4. Technologie utiliser

L'implémentation du nouveau système a nécessité l'utilisation de différents outils de développement, et à cause de ca nous avons choisi des outils qui sont souple et adaptable à notre travail. Nous présentons dans ce qui suit les principaux outils de développement que nous avons utilisés.

III.4.1 Visual Basic

est un langage de programmation événementielle de troisième génération ainsi qu'un environnement de développement intégré, créé par Microsoft pour son modèle de programmation COM[11]. Visual Basic est directement dérivé du BASIC et permet le développement rapide d'applications, la création

d'interfaces utilisateur graphiques, l'accès aux bases de données en utilisant les technologies DAO, ADO et RDO, ainsi que la création de contrôles ou objets ActiveX. Les langages de script tels que Visual Basic for Applications et VBScript sont syntaxiquement proches de Visual Basic, mais s'utilisent et se comportent de façon sensiblement différente[12].

Visual Basic a été conçu pour être facile à apprendre et à utiliser. Le langage permet de créer des applications graphiques de façon simple, mais également de créer des applications véritablement complexes. Programmer en VB est un mélange de plusieurs tâches, comme disposer visuellement les composants et contrôles sur les formulaires, définir les propriétés et les actions associées à ces composants, et enfin ajouter du code pour ajouter des fonctionnalités. Comme les attributs et les actions reçoivent des valeurs par défaut, il est possible de créer un programme simple sans que le programmeur ait à écrire de nombreuses lignes de code. Les premières versions ont souffert de problèmes de performance, mais avec l'apparition d'ordinateurs plus rapides et grâce à la compilation en code natif, ce problème de performance s'est estompé.

III.4.1.1. **Historique**

VB 1.0 a vu le jour en 1991. Le principe de connexion d'un langage de programmation avec une interface utilisateur graphique est dérivé d'un prototype appelé Tripod, développé par Alan Cooper. Microsoft avait alors contacté Cooper et ses associés pour développer un Shell programmable pour Windows 3.0, sous le nom de code Ruby (aucun lien avec le langage de programmation Ruby).

III.4.1.2. **Les Visual Basic de VB1 à VB6**

- Le projet *'Thunder'* est lancé.
- Visual Basic 1.0 (mai 1991) pour Windows est présenté au Comdex/Windows à Atlanta, Géorgie.
- Visual Basic 1.0 pour DOS est présenté en septembre 1992. Le langage n'était pas totalement compatible avec Visual Basic pour Windows, car il était en fait la nouvelle version des compilateurs Microsoft BASIC pour DOS, Quick Basic et *BASIC Professional Development System*. L'interface était en mode texte et utilisait le jeu de caractères ASCII étendu pour simuler une interface graphique.
- Visual Basic 2.0 est présenté en novembre 1992. L'environnement de développement était plus facile à utiliser et la vitesse avait été améliorée. Un fait marquant était que les formulaires étaient devenus des objets instanciables, posant ainsi le concept de base de modules de classe, qui devinrent plus tard disponibles dans VB4.
- Visual Basic 3.0 fut présenté pendant l'été 1993, disponibles en édition Standard ou Professionnelle. VB3 intégrait la version 1.1 du moteur de base de données Microsoft Jet, qui pouvait lire et écrire les bases de données Jet (ou Access) 1.x.

- Visual Basic 4 (août 1995) a été la première version qui pouvait générer des programmes Windows 16 et/ou 32 bits. Cette version introduisait également la possibilité d'écrire des classes sans interface utilisateur. Des incompatibilités entre les différentes versions de VB4 causèrent des problèmes d'installation et de fonctionnement.
- La version 5.0 de Visual Basic, sortie en 1997 ne fonctionnait plus qu'avec les versions 32 bits de Windows. Les programmeurs préférant développer des applications 16-bits pouvaient importer les programmes de VB4 vers VB5, et inversement. Visual Basic 5.0 a aussi introduit la possibilité de créer des contrôles utilisateurs personnalisés ainsi que la possibilité de générer des exécutables Windows natifs, ce qui améliorait la vitesse d'exécution des programmes effectuant beaucoup de calculs.
- Visual Basic 6.0 (mi-1998) a apporté de nombreuses améliorations dans différents domaines, notamment la possibilité de créer des applications Web-based. VB6 n'est plus supporté par Microsoft depuis mars 2008.
- La dernière version de Visual basic c'est le a version Visual Basic 6.0 est c'est lancer le 31 mars 2005. La version finale de VB6 était en mars 2008. En réaction, la communauté des utilisateurs Visual Basic a fait part de ses préoccupations à ce sujet et a incité les utilisateurs à signer une pétition afin de permettre de maintenir le produit en vie. À ce jour, Microsoft a refusé de modifier sa position à ce sujet. Ironiquement, il fut révélé au même moment que le nouvel Anti-Spyware de Microsoft, Microsoft AntiSpyware (racheté à GIANT Software Company) était écrit en Visual Basic 6.0. Windows Defender Beta 2 a été réécrit en C++/CLI[14].

Figure III-1 *: Logo de visual basic*

III.4.2 Microsoft Access

Microsoft Access (officiellement Microsoft Office Access) est une base de données relationnelle éditée par Microsoft. Ce logiciel fait partie de la suite Microsoft Office.

MS Access est composé de plusieurs programmes : le moteur de base de données Microsoft Jet, un éditeur graphique, une interface de type Query by Example pour interroger les bases de données, et le langage de programmation Visual Basic for Applications.

Depuis les premières versions, l'interface de Microsoft Access permet de gérer graphiquement des collections de données dans des tables, d'établir des relations entre ces tables selon les règles habituelles des bases de données relationnelles, de créer des requêtes avec le QBE (Query by Example, ou directement en langage SQL), de créer des interfaces homme/machine et des états d'impression. Comme pour les autres logiciels Office, le VBA,Visual Basic for Applications, permet de créer des applications complètes et en réseau local, y compris en utilisant, créant ou modifiant les fichiers (documents Word, classeurs Excel, instances Outlook, etc.) des autres logiciels de la suite sans quitter Access.

La dernière version en date est la version 2016 ; elle fait partie de la suite Microsoft Office 2016 et est incluse dans certaines options de l'abonnement à Office 365. La version par abonnement, Microsoft Office Access 365, est actualisée automatiquement comme celle de Windows 10.

***Figure III-2** : Logo de Microsoft Access*

III.4.3 Porteus ISIS et ARES

Proteus est une suite logicielle destinée à l'électronique. Développé par la société Labcenter Electronics, les logiciels incluent dans Proteus permettent la CAO dans le domaine électronique. Deux logiciels principaux composent cette suite logicielle : ISIS, ARES, PROSPICE et VSM.

III.4.3.1. Présentation générale

Cette suite logicielle est très connue dans le domaine de l'électronique. De nombreuses entreprises et organismes de formation (incluant lycée et université) utilisent cette suite logicielle. Outre la popularité de l'outil, Proteus possède d'autres avantages

- Pack contenant des logiciels facile et rapide à comprendre et utiliser
- Le support technique est performant
- L'outil de création de prototype virtuel permet de réduire les coûts matériel et logiciel lors de la conception d'un projet

III.4.3.1.1. ISIS

Le logiciel ISIS de Proteus est principalement connu pour éditer des schémas électriques. Par ailleurs, le logiciel permet également de simuler ces schémas ce qui permet de déceler certaines erreurs dès l'étape

de conception. Indirectement, les circuits électriques conçus grâce à ce logiciel peuvent être utilisé dans des documentations car le logiciel permet de contrôler la majorité de l'aspect graphique des circuits.

Par défaut ISIS inclut plusieurs bibliothèques des composants électroniques tel que les microcontrôleurs, Afficheurs, circuits analogique ou numérique, des actionneurs … etc, mais l'Arduino n'en fait pas partie. Nous allons donc ajouter la bibliothèque Arduino à ISIS, afin de pouvoir simuler notre système.

***Figure III-3** : Logo de simulateur ISIS*

III.4.4 Arduino IDE

Le logiciel de programmation de la carte Arduino sert d'éditeur de code (langage proche du C). Une fois, le programme tapé ou modifié au clavier, il sera transféré et mémorisé dans la carte à travers de la liaison USB. Le câble USB alimente à la fois en énergie la carte et transporte aussi l'information ce programme appelé IDE Arduino[14]

III.4.4.1. **Structure générale du programme (IDE Arduino)**

Comme n'importe quel langage de programmation, une interface souple et simple est exécutable sur n'importe quel système d'exploitation Arduino basé sur la programmation en C.

III.5. Dispositif matériel

III.5.1 La carte Arduino

Le module Arduino est un circuit imprimé en matériel libre (plateforme de contrôle) dont les plans de la carte elle-même sont publiés en licence libre dont certains composants de la carte : comme le microcontrôleur et les composants complémentaires qui ne sont pas en licence libre. Un microcontrôleur programmé peut analyser et produire des signaux électriques de manière à effectuer des tâches très diverses. Arduino est utilisé dans beaucoup d'applications comme l'électrotechnique industrielle et embarquée ; le modélisme, la domotique mais aussi dans des domaines différents comme l'art contemporain et le pilotage d'un robot, commande des moteurs et faire des jeux de lumières, communiquer avec l'ordinateur, commander des appareils mobiles (modélisme). Chaque module d'Arduino possède un régulateur de tension +5 V et un oscillateur à quartez 16 MHz (ou un résonateur céramique dans certains modèles). Pour programmer cette carte, on utilise l'logiciel IDE Arduino.[14]

Figure III-4 *: Logo d'Arduino*

III.5.2 Pourquoi 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. De la même façon, le système Arduino simplifie la façon de travailler avec les microcontrôleurs tout en offrant à personnes intéressées plusieurs avantages cités comme suit :

- **Le prix (réduits)** : les cartes Arduino sont relativement peu coûteuses comparativement aux autres plates-formes. La moins chère des versions du module Arduino peut être assemblée à la main, (les cartes Arduino préassemblées coûtent moins de 2500 Dinars).

- **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 (fonctionnant sur tout système d'exploitation), servant d'éditeur de code et de compilateur, et qui peut transférer le programme au 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, ATMEGA 328, les schémas des modules sont publiés sous une licence créative Commons, et 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[14]

III.6. Architecture de l'outil

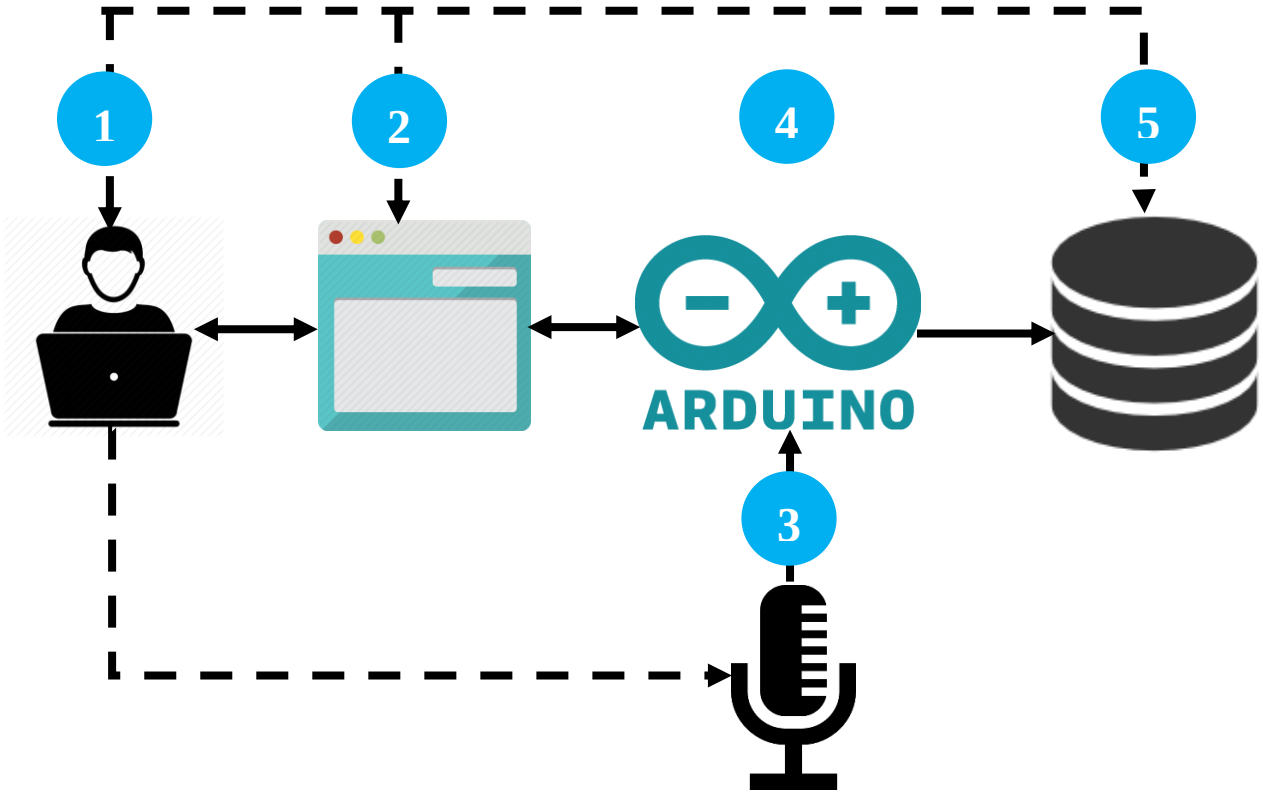


Figure III-5 : Architecture de l'outil

Avec l'intervention de l'utilisateur qui va lancer l'interface de programme et intervenue pour le microphone aussi après ça la carte va convertir le son et l'utilisateur choisi soit de stoker le signal dans la base de donnée soit comparer avec un autre signal et dans les deux cas l'interface va afficher soit que le son est enregistrer dans la base de donnée soit le résultat de traitement

III.7. Démonstration de notre outil

III.7.1 Interface de début de la simulation d'Arduino avec POTEUS

Cette figure montre le début de la simulation de la carte Arduino.

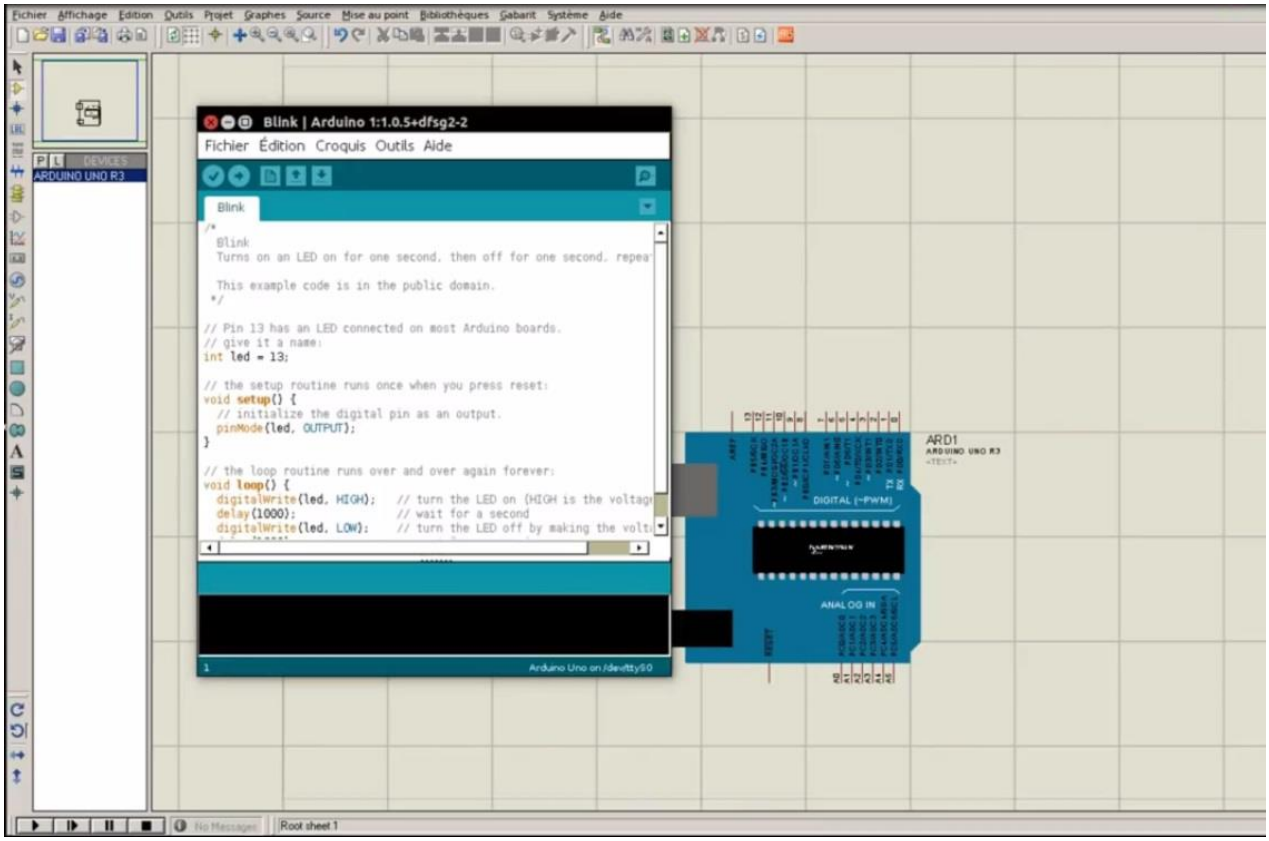


Figure III-6: *Début de simulation avec Arduino*

III.7.2 Interface de la simulation final d'Arduino avec POTEUS

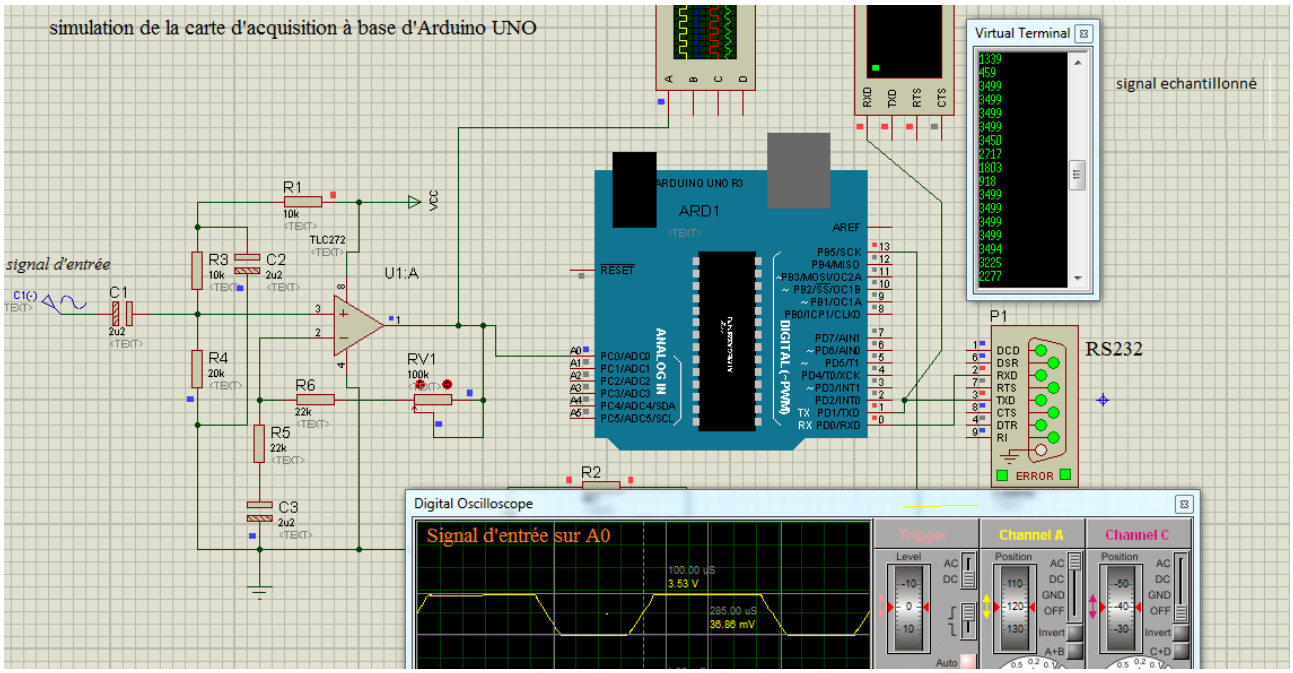


Figure III-7 : *Fenêtre de la simulation complète d'Arduino avec POTEUS*

III.7.3 Ecran de la simulation avec VB6

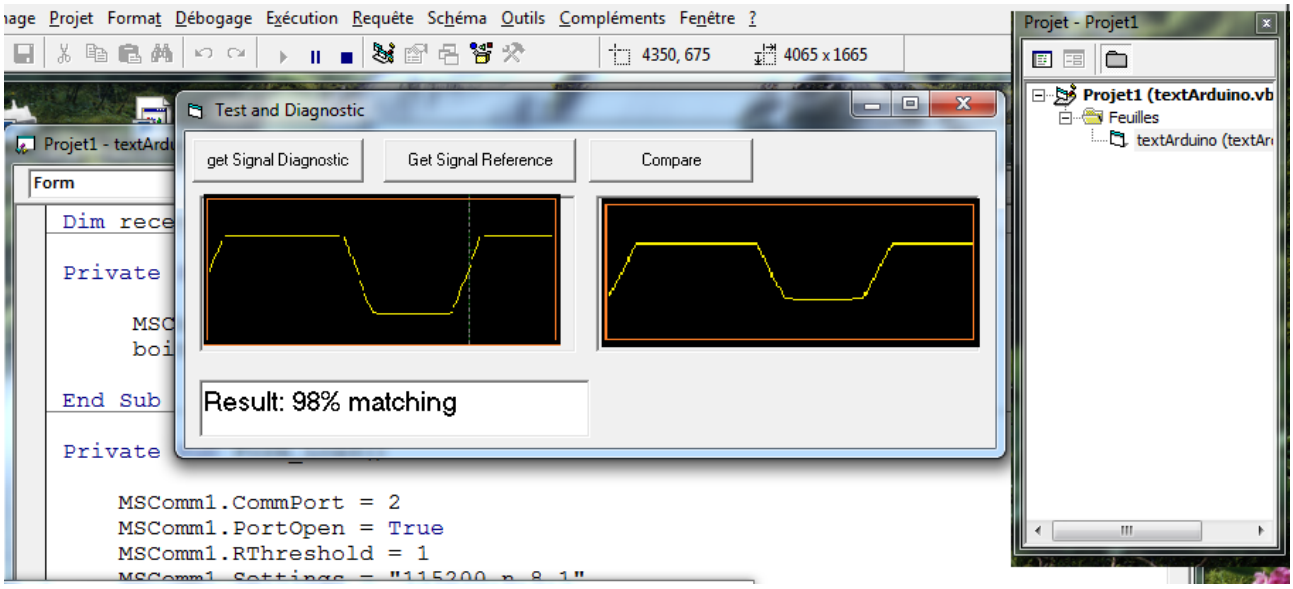


Figure III-8 : *Ecran de la simulation avec VB6*

Ecran obtenu à l'aide de VB6 sur PC qui montre l'opération de comparaison des deux signaux et affichage du résultat.

III.8. Conclusion

Dans ce chapitre nous avons fait une simulation de notre système et une réalisation après ca nous avons montré des capture d'écran de la simulation ainsi que la réalisation, ce dernier permettant de comparer entre deux signaux audio et les tracer dans sur un écran.

Dans le prochain chapitre on va faire une expérience qui va tester la fiabilité de notre système.

IV. Bibliographie

[12] F. Balena, «Microsoft Visual Basic 2005,» chez *Microsoft Visual Basic 2005*, Paris, Microsoft press, 2006, p. 640.

[13] M. Halvorson, Microsoft Visual Basic 2010, Paris: Miscrosoft press, 2010.

[14] «Product Family Life Cycle Guidelines for Visual Basic 6.0».