



الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne Démocratique et Populaire
وزارة التعليم العالي والبحث العلمي
Ministère de l'Enseignement Supérieure et de la Recherche Scientifique

جامعة وهران 2 محمد بن أحمد
Université d'Oran 2 Mohamed Ben Ahmed
معهد الصيانة والأمن الصناعي
Institut de Maintenance et de Sécurité Industrielle

Département de Maintenance en Instrumentation

MÉMOIRE

Pour l'obtention du diplôme de Master

Filière : Génie Industriel

Spécialité : Génie Industriel

Thème

Réalisation d'un Télémètre a Ultrason

Présenté et soutenu publiquement par :

Nom : Hamadouche

Prénom : Mohamed Ilyes

Nom : Hankour

Prénom : Mohammed El Mehdi

Devant le jury composé de :

Nom et Prénom	Grade	Etablissement	Qualité
Lalaoui Mohamed EL Amine	MAA	IMSI	Président
Nekrouf Djilali	MAA	IMSI	Encadreur
Rouan Serik Mehdi	MCB	IMSI	Examinateur

Juillet 2021

Remerciement

Avant tout, nous remercions Allah tout puissant qui nous a guidés tout au long de notre modeste vie, qui nous a donnés courage et patience pour passer tous les moments difficiles, qui nous a permis d'achever ce travail et de pouvoir le mettre entre vos mains aujourd'hui.

Ce mémoire est aujourd'hui l'occasion de remercier notre chère petite famille, nos parents qui sont la cause de toutes les bonnes choses de nos vies, et nos amis pour leur support.

A nos professeurs dans le département de maintenance en instrumentations, surtout notre encadreur : M^r Nekrouf, pour son assistance, encouragements et orientations bien guidées.

Nous remercions toute personne ayant contribué, de près ou de loin, dans ce travail, merci infiniment.

DEDICACE

Je dédie cet ouvrage

À mes très chers parents, pour m'avoir mis au monde et pour m'avoir accompagné tout le long de ma vie et m'orienter vers l'aboutissement de mes études ;

À mes frères et sœurs et amis qui ont su m'encourager ;

À Yasmine ma cousine qui nous a guidés toute au long de ce projet ;

Et je tiens spécialement à remercier chaleureusement mon encadreur

M^r NEKROUF qui a su me conseiller et mon encadrer judicieusement.

Mohamed Ilyas

DEDICACE

Je dédie cet ouvrage

À mes très chers parents, pour m'avoir mis au monde et pour m'avoir accompagné tout le long de ma vie et m'orienter vers l'aboutissement de mes études ;

À mes frères et sœurs et amis qui ont su m'encourager ;

À la bande organisée car personne ne peut nous canaliser ;

Et je tiens spécialement à remercier chaleureusement mon encadreur

M^r NEKROUF qui a su me conseiller et mon encadrer judicieusement.

Mohammed EL Mehdi

Sommaire

Chapitre I :Généralités sur la Télémétrie à Ultrasons

I.1. Introduction	1
I.2. Définition des ultrasons	1
I.3. Production et détection des ultrasons:.....	2
<i>I.3.1.1. La Piézoélectricité :</i>	3
<i>I.3.1.2. Emetteur à magnétostriction</i>	5
<i>I.3.1.3. Transducteurs à ondes acoustiques de surface</i>	6
I.3.2. Détection:.....	7
<i>I.3.2.1. Temps de parcours de l'onde ultrasonore</i>	7
I.4Propriétés des ultrasons :.....	10
I.4.1 Amortissement :	10
I.4.2 Directivité :	10
I.4.3 Vitesse :	10
I.4.4 Réflexion :.....	11
I.4.5 Réfraction :.....	12
I.4.6 Longueur d'onde :	12
I.4.7 Propagation :	13
1.5. Forme de l'objet:	13
1.6. Principe de fonctionnement du télémètre:.....	15
Chapitre II : Description de la Partie Matérielle et Logicielle	
II-1 Introduction.....	16
• II-2 Description de la partie matérielle	16
• II-2-1 La carte Arduino UNO	16
• II-2-2 capteur ultrason (HC-SR04)	21
II-2-3 Bluetooth HC-05 :	23
• II-3 Présentation de «IDE ».....	25
• II-3-1 Programmer avec Arduino	26
II-3-2 Syntaxe des commandes Arduino :	28
II-3-3 Fritzing :.....	31
II-4 MIT App Inventor :.....	32
II-4-1 Historique de logiciel App Inventor	33
II-4-2 ligne de code :.....	34

II-4-3 Visualisation directement sur SmartPhone :	38
II-4-4 Téléchargement :	38
II-4-5 Des blocs de connectivité	39
II-5 Conclusion	40
Chapitre III : Conception et Réalisation Du Télémètre à Ultrasons	
Introduction :	41
• III-1 Réalisation du Projet	41
III-1-1 Composants utilisés :	41
III-2 Communication :	42
III-2-1 La connexion entre l'Arduino et le détecteur a ultrason HC-SR 04 :	42
III-2-2 La connexion entre l'Arduino et Bluetooth HC-05:	42
III-2-3 La connexion entre l'Arduino et Bluetooth HC-05 et le détecteur a ultrason HC-SR 04 :	43
III-3 Le principe de notre projet :	44
• III-4 Programmations De L'arduino	44
III-4-1 le programme :	44
III-4-2 programme sous IDE:	45
• III-5 l'application	46
III.6.Simulations	49
Tab	49
III-6 conclusion	51

Liste des figures :

Figure I.1 : Transmission et réflexion de l'onde ultrasonore à travers une interface.

Figure I.2 : Cristal de Quartz naturel.

Figure I.3 : Effets piézoélectriques directs et indirects.

Figure I.4 : Triplet de Longevin.

Figure I.5 : Ligne à retard avec émetteur et récepteur d'ultrasons magnétostrictifs.

Figure I.6 : Dispositif à ondes acoustiques de surface.

Figure I.7 : Réflexion de l'onde ultrasonore sur un obstacle.

Figure I.8 : Cône de détection.

Figure I.9 : Propagation dans un milieu vers un autre.

Figure I.10 : Exemple de forme d'objet.

Figure I.11 : Principe du télémètre à ultrasons.

Figure I.12 : Schéma synoptique du télémètre à ultrasons.

Figure II.1: La carte Arduino UNO.

Figure II.2 : Les composants de la carte Arduino UNO.

Figure II.3 : Constitution de la carte Arduino UNO.

Figure II.4 : Capteur Ultrason HC-SR04.

Figure II.5 : Principe de fonctionnement d'un capteur Ultrason HC-SR04.

Figure II.6 : Vue du Module Bluetooth HC-05.

Figure II.7 : Interface IDE.

Figure II.8 : structure du programme IDE.

Figure II.9 : structure Mit app inventor.

Figure II.10 : interface de programmation par bloc.

Figure II.11: Première interface de la création APP INVENTOR.

Figure II.12 : Résumer des quatre étapes.

Figure II.13 : Synchronisation du design avec un SmartPhone Android.

Figure II.14 : l'application sur SmartPhone.

Figure II.15 : Blocs de connectivité Bluetooth dans MIT App Inventor.

Figure II.16: L'organisation générale de l'interface de développement.

Figure III.1 : Schéma synoptique de notre projet.

Figure III.2: La connexion entre l'arduino et le détecteur a ultrason HC-SR 04.

Figure III.3: La connexion entre l'Arduino et Bluetooth HC-05.

Figure III.4: La connexion entre l'Arduino et Bluetooth HC-05 et le détecteur a ultrason HC-SR 04.

Figure III.5 : Programme télémètre sous ide.

Figure III.6 : Organigramme de mesure de distance.

Figure III.7 : applications sous app inventor.

Figure III .8 : block pour crée l'application.

Figure III .9 : interface de l'application.

Figure III.10: Graphique de linéarité du capteur.

Figure III.11 : Graphique d'erreurs du capteur.

Figure III.12 : montage final d'un télémètre a ultrason avec Bluetooth.

- **Liste des tableaux :**

Tab. I.1 Vitesse de propagation des ultrasons dans quelques milieux.

Tab III.1 tableau des mesures prise avec marge d'erreur.

Introduction Générale

• **Introduction Générale :**

L'objectif du projet est de développer un télémètre à ultrasons qui permettra de mesurer la distance sans contact. La mesure de distance sans contact est un domaine d'application vaste et passionnant.

Il existe aujourd'hui de nombreuses technologies qui peuvent attirer différents domaines électroniques. Le principe d'un télémètre à ultrasons s'inspire des chauves-souris et consiste à envoyer une rafale d'ultrasons vers des objets qui les reflètent.

Nous obtenons la distance entre le télémètre et l'objet en mesurant le temps mis par l'écho retourné au télémètre. Connaissant la vitesse du son, on en déduit la distance requise.

Cet appareil de mesure présente de nombreux avantages. La vitesse de mesure est très rapide, même dans quelques mètres, il est sûr de mesurer la distance qui est difficile à atteindre.

Le fonctionnement du télémètre se base sur quatre modules : un module émetteur, un module de réception, module de contrôle et module d'affichage de distance mesurée.

Les modules de contrôle qui interagissent avec les trois autres sont nécessaires pour vérifier le fonctionnement de l'ensemble du télémètre. Le principe est d'émettre des ondes ultrasonores vers des objets massifs ou des murs, La détection de l'écho renvoyé par l'obstacle permet de mesurer la distance entre le système émetteur-récepteur et l'obstacle. Pour ce faire, le projet est divisé en trois chapitres :

- **Le chapitre 1** expose les éléments qui composent le télémètre à ultrasons ;
- **Le chapitre 2** présente le principe de son fonctionnement ;
- **Le chapitre 3** met en relief spécifiquement la mise en œuvre pratique de notre projet et Mode d'emploi du télémètre.
- Puis on terminera par une conclusion générale.

Chapitre I :
Généralités sur la Télémétrie
à Ultrasons

- **I.1. Introduction:**

L'utilisation d'ultrasons pour mesurer la distance d'une cible est un large éventail d'applications. C'est très excitant que dans le premier chapitre, nous allons présenter Télémétrie à ultrasons.

- **I.2. Définition des ultrasons :**

Lazzaro Spallanzani soupçonna pour la première fois l'existence des ultrasons en 1794. Ils ont été découverts en 1883 par le physiologiste britannique Francis Galton.

Contrairement à la lumière, qui est une onde électromagnétique, elle peut se propageant dans le vide, le son est une onde mécanique qui nécessite un support matériel. Les ultrasons sont la vibration mécanique de la matière à une fréquence inaudible par l'oreille humaine ($> 20\,000$ Hz) [1].

La vitesse de propagation des ultrasons dépend de la nature du milieu, Cela n'a rien à voir avec la fréquence de l'onde. Par exemple, la vitesse du son dans l'air est Plus de 300 m/s, alors que dans l'eau c'est 1500 m/s. Dans le corps humain, les ultrasons se déplaceront à une vitesse proche de 1500 m/s selon la nature de l'organe qu'ils passent [1]

Le tissu montre une certaine résistance au passage des ultrasons. Cette résistance, ce qu'on appelle l'impédance sera fonction du module d'élasticité et de la densité du milieu considéré. L'impédance d'un organe est différente de celle d'un autre, et la frontière entre les deux organes constitue une interface. Tout comme la lumière dans l'optique, chaque fois que le son rencontre une interface, L'énergie incidente est transmise (elle passe par l'interface), tandis que l'autre partie est réfléchi (Figure I.1). La direction de transmission et de réflexion dépendra de l'angle L'incidence des ondes sonores. Si l'incidence est directe, c'est-à-dire perpendiculaire à l'interface, la transmission se fait à La direction et celle des ondes sonores sont les mêmes, et la réflexion se produit dans La même direction et la direction opposée. Si l'incident n'est pas perpendiculaire à l'interface, l'onde transmise déviara, c'est la réfraction.

L'onde réfléchi est également déviée d'un angle égal à l'onde incidente Orthogonal à l'interface. Le rapport de l'énergie transmise et réfléchi et l'angle de réfraction sont des fonctions La différence d'impédance entre les deux tissus.

Plus la différence d'impédance est grande, plus la partie réfléchi est grande. Lorsque la taille de l'interface est inférieure à Longueur d'onde des ultrasons. La diffusion est le retour de toutes les ondes incidentes dans l'espace. Enfin, il existe un phénomène dans lequel le tissu croisé absorbe de l'énergie ; la conversion de l'énergie sonore en énergie thermique [1,2].

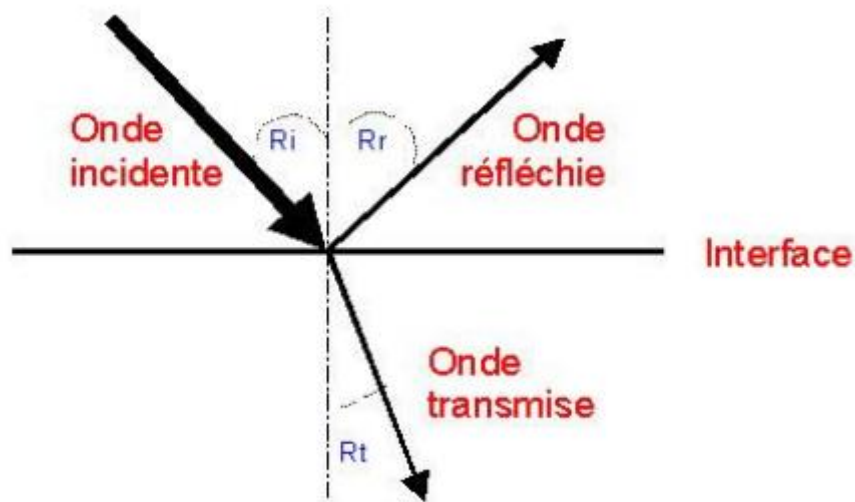


Figure. I.1. Transmission et réflexion de l'onde ultrasonore à travers une interface

I.3. Production et détection des ultrasons:

Le sifflet d'air comprimé est le premier mode de génération d'ultrasons ; mais même s'il y a des améliorations, les ultrasons obtenus par ce procédé ne sont pas satisfaisants. Après avoir épuisé la possibilité d'utiliser le phénomène pour générer des ultrasons en mécanique simple, les ingénieurs ont étudié l'usage au début du XXe siècle du phénomène électrique. Les méthodes de génération et de détection les plus couramment utilisées aujourd'hui l'échographie qui utilise le phénomène de « piézoélectricité ». Il existe d'autres moyens de générer des ultrasons. L'un d'eux utilise des effets électromagnétiques, il s'agit d'un émetteur magnétique [1,3].

1.3.1.1. La Piézoélectricité :

La plaque de quartz se polarise sous pression ou traction, et Au contraire, la plaque de quartz va se dilater ou se contracter sous l'action du champ électrique selon la direction du champ (Figure I.2).

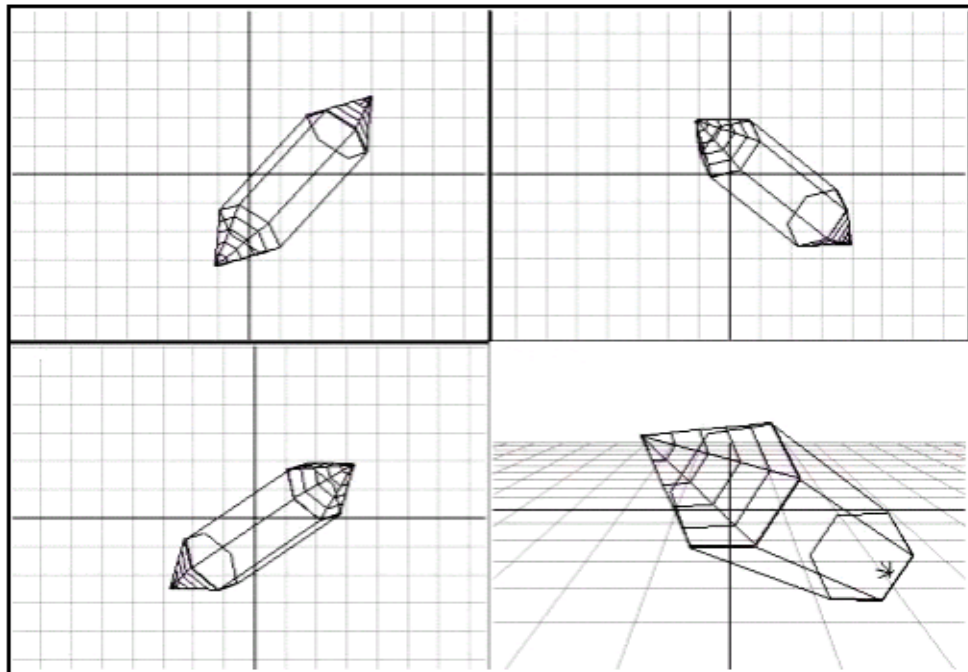


Figure. I.2. Cristal de Quartz naturel

La piézoélectricité caractérise l'aptitude de certains matériaux cristallins de convertir une contrainte mécanique en une énergie électrique et réciproquement. Il s'agit d'une polarisation électrique induite par une déformation mécanique dans des cristaux appartenant à certaines classes cristallines. Cette polarisation est proportionnelle à la déformation et change de signe avec elle.

Cet effet appelé «effet piézoélectrique direct», est réversible : l'application d'un champ électrique externe induit une déformation mécanique ; c'est l'effet piézoélectrique inverse.

A l'échelle de la maille cristalline, la Piézoélectricité se manifeste par une polarisation de la maille, résultant de la séparation du centre de gravité des charges positives et négatives. L'origine de l'effet piézoélectrique est liée à la structure et la symétrie cristallines des matériaux. En effet, pour qu'un cristal soit piézoélectrique, il doit être non centrosymétrique c'est-à-dire que les barycentres des charges positives et négatives ne coïncident pas dans la maille Figure I.3.

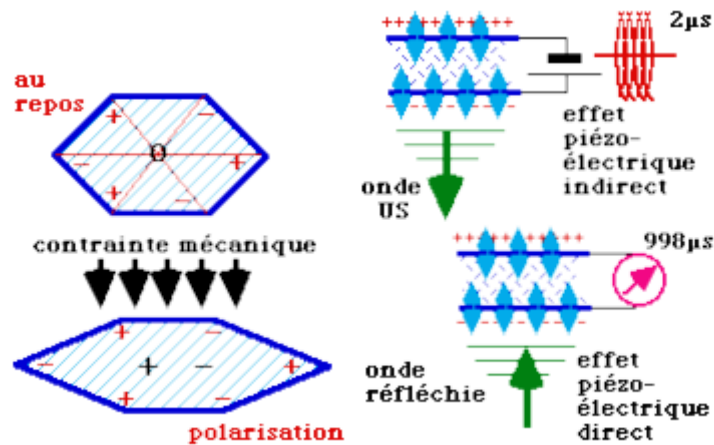


Figure. I.3. Effets piézoélectriques directs et indirects

Nous sommes ainsi capables de générer et de détecter des fluctuations de dimension d'une aube. (Convenablement choisi) extrait d'un cristal de quartz. Naturellement, l'exploitation du phénomène de la piézoélectricité a connu plusieurs évolutions lui permettant d'atteindre des fréquences plus élevées en ce qui concerne la production d'ondes élastiques voulu: -L'utilisation du phénomène de résonance permet ainsi d'augmenter les rendements (une tension inférieure est alors nécessaire pour atteindre la même fréquence)

Le triplet quartz-acier de Langevin (Figure. I.4), constitué d'une fine bande de quartz collé entre deux lames en acier, permet d'économiser du quartz et une tension supplémentaire requise diminué [1, 2,3].

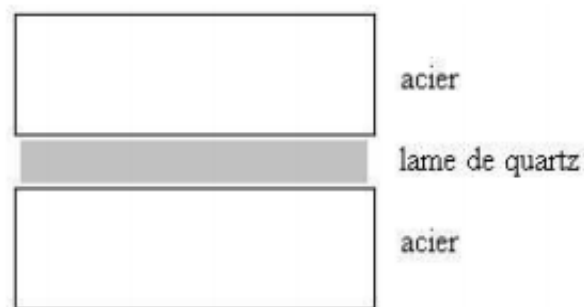


Figure. I.4. Triplet de Langevin

1.3.1.2. Emetteur à magnétostriction:

Décrivons d'abord le phénomène utilisé pour cet autre mode de production d'ultrasons : Enroulez un enroulement conducteur autour d'une barre cylindrique de nickel; quand le courant passe, il y a une diminution de la longueur du cylindre. Quant à l'émetteur piézoélectrique, les dimensions de l'émetteur et de la fréquence pour bénéficier du phénomène de résonance. Utiliser céramiques composées d'oxydes mixtes de fer, de nickel, de zinc et de plomb, appelées de ferrites, a permis d'atteindre des fréquences n'approchant que 100 kHz. Après les nombreuses améliorations qui lui ont été apportées, l'émetteur utilisant l'effet piézoélectrique est capable, à titre de comparaison, d'atteindre des fréquences de plusieurs GHz, il n'est donc pas surprenant de constater que ce mode de production est le plus courant actuellement [3]

Dans les deux cas, il s'agit de la production d'ondes ultrasonores par transformation des oscillations électriques aux oscillations élastiques, et dans les deux cas nous chercherons ainsi à profiter du phénomène de résonance pour augmenter les rendements, c'est-à-dire atteindre des fréquences plus élevées avec moins d'énergie [1,3]. L'effet magnétostrictif permet de réaliser des transducteurs et des capteurs, notamment émetteurs et récepteurs d'ondes acoustiques. Par exemple, nous pouvons faire une ligne à retard par le système simple suivant (Figure. I.5) :

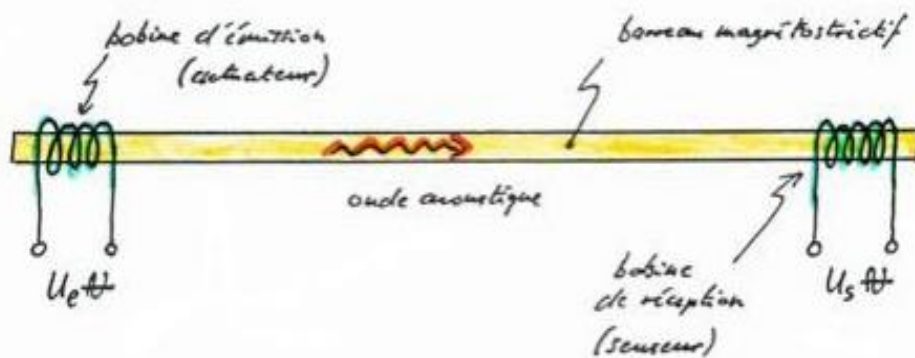


Figure. I.5. Ligne à retard avec émetteur et récepteur d'ultrasons magnétostrictifs

1.3.1.3. Transducteurs à ondes acoustiques de surface:

Un appareil à ondes acoustiques de surface (appelé SAW en anglais pour Surface Acoustic Wave) sont couramment utilisés dans les systèmes de compression d'impulsions. Ils utilisent les propriétés d'un substrat piézoélectrique pour transporter les ondes acoustiques vers la surface. Ce dispositif est constitué de fils métalliques minces déposés par gravure acide, tels qu'à la figure I.6, sur le substrat. Ces fils servent de transducteur numérique et convertissent le signal électrique analogique en ondes acoustiques vers le substrat. La faible vitesse de propagation de ces ondes permet d'introduire un retard important même sur une faible distance, ce qui en fait une ligne à retard. La réponse en fréquence de cette ligne dépend de l'espacement des transducteurs [1, 3,4].

Dans la figure (I.6), l'impulsion entre à gauche et ressort à droite comme une onde compression. Les fréquences les plus hautes subissent le plus grand retard de propagation et se superposent ainsi aux plus basses fréquences. Le résultat est une impulsion comprenant un ensemble de fréquences dans une cellule de résolution du radar. La présence d'harmoniques dans le signal compressé va ralentir le traitement de l'impulsion par les filtres de décompression ce qui donnera des lobes secondaires temporels, aussi appelés lobes secondaires en portée [3,4].

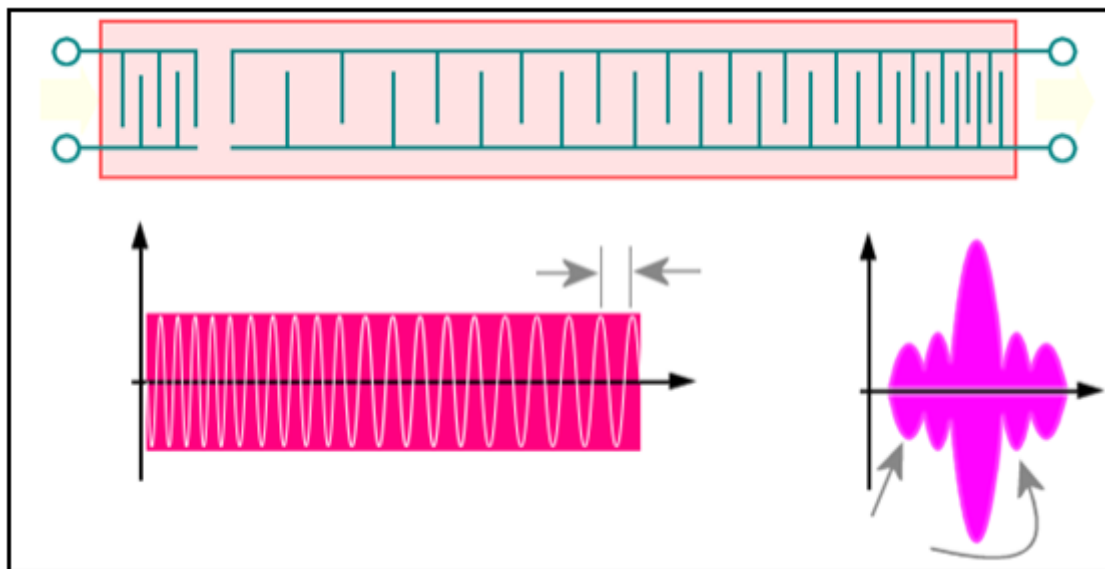


Figure. I.6. Dispositif à ondes acoustiques de surface

I.3.2. Détection:

I.3.2.1. Temps de parcours de l'onde ultrasonore:

Pour mesurer une distance avec des ultrasons, il faut d'abord connaître leur vitesse de Déplacement. C'est ce que nous venons de voir. Elle est de l'ordre de 340 m par seconde, ce qui correspond à plus de 1200 Km par heure [1,4]. On peut déterminer la distance qu'a parcourue l'onde Ultra sonore si nous connaissons le temps qu'elle a mis pour parcourir cette même distance [1].

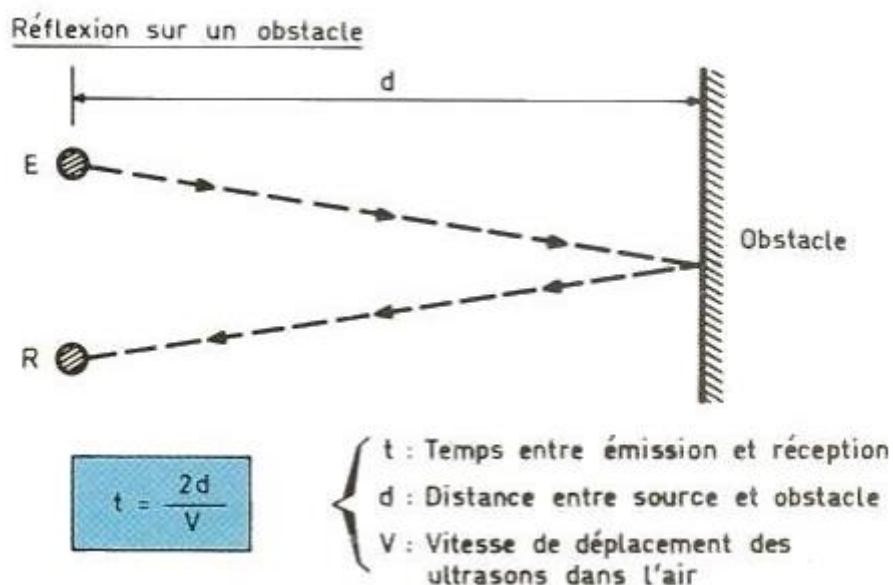


Figure. I.7. Réflexion de l'onde ultrasonore sur un obstacle

Il montre comment calculer la distance entre un capteur à ultrasons et un objet. On peut voir deux petits cercles nommés E et R (Figure I.7). Le cercle E représente l'émetteur à ultrasons, ce genre d'enceinte émettra Ultrason. Le cercle R représente le récepteur ultrasonore, c'est-à-dire le type de microphone qui reçoit les ondes ultrasonores de l'émetteur. Les ondes ultrasonores sont d'abord émises par l'émetteur du capteur à ultrasons puis traversent une certaine distance d avant de heurter un obstacle. Alors l'onde est réfléchié par l'obstacle, c'est le reflet de l'onde. Enfin, avant d'être reçu, il doit parcourir à nouveau la même distance en sens inverse d récepteur de capteur à ultrasons.

Il faut noter que l'onde doit être le double de la distance entre le capteur à ultrasons et le capteur d'obstacle. C'est pourquoi vous trouvez $2d$ dans la formule contenue dans l'image. On doit savoir que dans le capteur à ultrasons, il y a une sorte de chronomètre qui permet de mesurer plus précisément, il s'agit du temps écoulé entre le moment où l'émetteur émet une onde ultrasonore et le moment où l'onde ultrasonore est émise. Le moment où le destinataire le reçoit. Le temps t n'est donc plus calculé mais mesuré. Cela signifie que lorsque le

capteur à ultrasons effectue une mesure, il émet une onde ultrasonore. Sonnez et mesurez le temps qu'il a reçu.

La seule inconnue est donc la distance. Il suffit d'une petite transformation de la formule ci-dessus, car nous voulons calculer maintenant la distance d :

$$d = Vt/2 \quad (I.2)$$

Par exemple, si le capteur d'ultrasons mesure un temps de 4 millisecondes (0,004s) entre l'émission et la réception de l'onde ultra sonore :

$$d = Vt/2 = 340 \times 0.004 / 2 = 1.36 / 2 = 0.68 \text{m} \quad (I.3)$$

Cela veut dire que l'onde a parcourue 1,36 m au total et que donc l'obstacle se trouve à une distance de 0,68m.

L'obstacle doit être exactement devant le capteur pour qu'il puisse mesurer son distance. Les capteurs à ultrasons peuvent détecter les obstacles qui ne sont pas positionnés avec précision Devant lui, mais sur les côtés. C'est ce qu'on appelle le cône de détection [1, 2,4].

- ***1.3.2.2. Cône de détection:***

Le télémètre à ultrasons peut détecter relativement les obstacles dans le cône. Sa largeur est d'environ 30 degrés, comme le montre la Figure I.8.

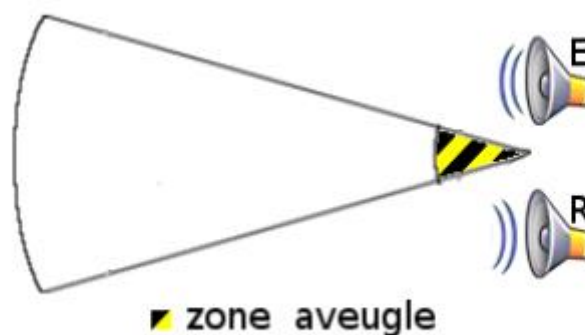


Figure. I.8. Cône de détection

Cette caractéristique est à la fois un avantage et un inconvénient. Ceci est un inconvénient car les obstacles détectés ne sont pas localisés avec précision à l'intérieur du cône de détection. Pour une distance donnée, les obstacles peuvent être trouvés n'importe où sur toute la largeur du cône de détection. Par

conséquent, la distance mesurée est plus fiable. Mais la localisation des obstacles n'est pas précise [1,4].

Par contre, c'est un avantage car la zone de détection du capteur à ultrasons est plus grande que d'utiliser un capteur plus précis. Ainsi, il peut détecter des obstacles relativement fins, tels que Par exemple, les pieds de table ou les pieds de chaise, lorsque cela peut ne pas être détecté par le télémètre avec un cône de détection plus faible [1, 2,4].

- ***1.3.2.3. Zone aveugle:***

Les télémètres à ultrasons ont une soi-disant "zone aveugle" de quelques centimètres de large en dessous de cette longueur où ils ne peuvent pas détecter les obstacles. Ceci est dû au fait le récepteur du capteur à ultrasons est désactivé peu de temps après la transmission Ultrason. En fait, si ce n'est pas le cas, car le récepteur est situé à proximité de l'émetteur du capteur à ultrasons, il recevra immédiatement un écho ultrasonore, ce qui faussera le résultat de la mesure distance.

Par conséquent, afin de surmonter ce problème, le récepteur dispose d'un petit délai pendant son off est exécuté immédiatement après l'émission de l'onde ultrasonore. Une fois ce lorsque le délai expire, le récepteur est à nouveau activé et est prêt à recevoir l'écho des obstacles lointains.

Deuxièmement, cela dépend également de la sensibilité du récepteur d'ultrasons. Autrement dit Il peut détecter des ondes ultrasonores très faibles. De même que nos oreilles ont une certaine sensibilité, le son que nous entendons est plus ou moins, Le récepteur du capteur à ultrasons ne peut pas détecter les ondes ultrasonores trop faibles renvoyées-t-ils. Cela limite également la portée du capteur à ultrasons. C'est pourquoi la portée théorique du capteur à ultrasons est de 2,5 m, soit de l'ordre de 250 cm.

Cette gamme, à l'exception de la diminution selon la batterie ou la puissance de la batterie, Cela dépendra également de la surface et de la forme de l'objet détecté [1,2].

I. 4 Propriétés des ultrasons :

Parmi les propriétés des ultrasons nous pouvons citer :

I.4.1 Amortissement :

Le coefficient d'amortissement est dû à la dissipation de l'énergie de l'onde ultrasonore lorsqu'elle s'éloigne de la source. Cet amortissement se traduit par une décroissance de l'intensité ultrasonique en fonction de la distance parcourue suivant la loi :

$$I = I_0 e^{-2al}$$

A : Coefficient d'amortissement

I₀ : Intensité de la source

L : Distance parcourue dans le milieu

I.4.2 Directivité :

La puissance acoustique rayonnée par une source n'est généralement pas répartie de manière uniforme dans toutes les directions de l'espace. La directivité dépend de la longueur d'onde et du diamètre de la source ultrasonore.

$$\theta = 2\lambda/D$$

$$\begin{array}{l} \theta : A \quad d'o \\ \lambda : L \quad d'o \end{array}$$

D : Diamètre de la source.

I.4.3 Vitesse :

La vitesse de propagation dépend de plusieurs paramètres :

- La densité du milieu.
- L'élasticité du milieu.
- Nature des matériaux.
- Mode de propagation.

Quelques valeurs typiques de la vitesse des ondes ultrasoniques correspondant aux types de matériaux sont données sur la table.1.1 :

MATIERE	VITESSE (m /s)
Air (Dans des conditions normales)	330
Mercure	1450
L'eau	1480
Cristallin	1620
Boite crânienne	4080
Aluminium	6400

Tab. I.1 Vitesse de propagation des ultrasons dans quelques milieux.

Cette vitesse est donnée par la relation suivante :

$$V = (\gamma / \rho)^{1/2}$$

γ : Coefficient de Poisson d'un matériau à 1,4.

ρ : Densité du matériau en kg/m³.

T : Tension superficielle en N/m.

I.4.4 Réflexion :

Ce phénomène de réflexion présente un cas particulier en physique. Il est très utilisé dans la pratique et exprimé par le facteur de réflexion.

$$r = \frac{Z_2 - Z_1}{Z_1 + Z_2}$$

Z_1 : impédance du milieu de l'onde incidente

Z_2 : impédance du milieu de l'onde réfractée

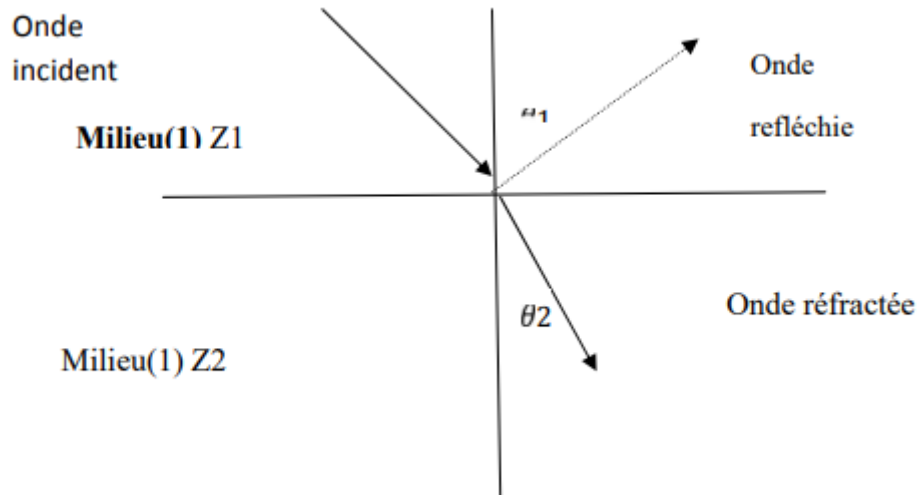


Figure I.9: Propagation dans un milieu vers un autre

Deux cas peuvent se présenter :

$r=0$ ($Z2=Z1$) ce qui implique une transmission parfaite

$r=1$ ($Z2 \gg Z1$) réflexion presque totale

Ce résultat est également intéressant dans la pratique chaque fois que la rupture de continuité d'un milieu est détectée par la réflexion d'une onde ultrasonore.

I.4.5 Réfraction :

La réfraction est le phénomène d'incurvation des rayons sonores lorsqu'ils se propagent dans un milieu dont la célérité varie spatialement.

I.4.6 Longueur d'onde :

Si un point milieu élastique est le siège de vibration périodique, la longueur d'onde λ explicitant la périodicité dans l'espace, est définie par analogie à T. Elle est égale au chemin parcouru par un front d'onde pendant une période. Elle est donnée par relation suivante :

$$\lambda = C$$

I.4.7 Propagation :

Dans un milieu déterminé de dimensions infinies, les ultrasons se propagent comme des ondes acoustiques. L'émetteur d'ultrasons provoque la mise en vibration des particules du milieu qui l'entoure, ensuite les oscillations se transmettent de proche en proche, nous distinguons deux modes de propagation :

- Le mode longitudinal : Les particules du milieu vibrent suivant la direction de propagation.
- Le mode transversale : Les particules du milieu vibrent suivant la direction perpendiculaire à la direction de propagation

1.5. Forme de l'objet:

La taille de la surface de l'objet détecté est importante, mais sa forme semble plus importante. En effet, un objet très plat est perpendiculaire au capteur à ultrasons renvoie toutes les ondes de retour dans la même direction ; ce qui facilitera sa détection.

Ce n'est pas le cas pour les objets de forme ronde, comme les boules, renverra les ondes dans plusieurs directions, donc seule une très petite partie sera vue et atteint le récepteur du capteur à ultrasons (Figure. I.9).

Ceci est illustré par des flèches de différentes couleurs dans la figure ci-dessous. La Flèche Le bleu est l'onde émise par l'émetteur du capteur à ultrasons, la flèche verte est la petite partie, cette même onde retourne au récepteur du capteur à ultrasons.

D'autres flèches Représente la partie de l'onde émise dans plusieurs autres directions, et ne sera pas reçu par le récepteur.



Figure. I.10. Exemple de forme d'objet

Par conséquent, nous pouvons remarquer que l'objet plan perpendiculaire au capteur à ultrasons a la même surface que cette balle renverra plus d'ondes que la balle, elle sera donc plus détectée en tenant à l'écart du capteur à ultrasons. On parle ici de la surface visible du capteur à ultrasons, celui qui revient à la réception est la moitié de la surface réelle de la balle.

Par conséquent, en termes de portée, la forme d'un objet est aussi importante que sa surface. Le capteur à ultrasons détecte l'objet. La texture de l'objet affectera sa distance de détection. Avoir un par exemple, des balles en caoutchouc, en mousse ou en plastique rendront les vagues plus grosses ou moins absorbées par les matériaux texturés [1,2].

1.6. Principe de fonctionnement du télémètre:

Le fonctionnement de notre télémètre repose sur quatre modules :
(Figure. I.10) :

- Module d'émission. (NE 555 • 4538)
- Module de réception. (Amplification • Filtrage)
- Module qui affiche la distance mesurée. (Horloge • Cycle de mesure • Masse virtuelle)
- Module de commande et traitement du signal qui interagira avec les trois premiers modules pour un bon contrôle du fonctionnement de l'ensemble du télémètre. (Horloge • Bascule)

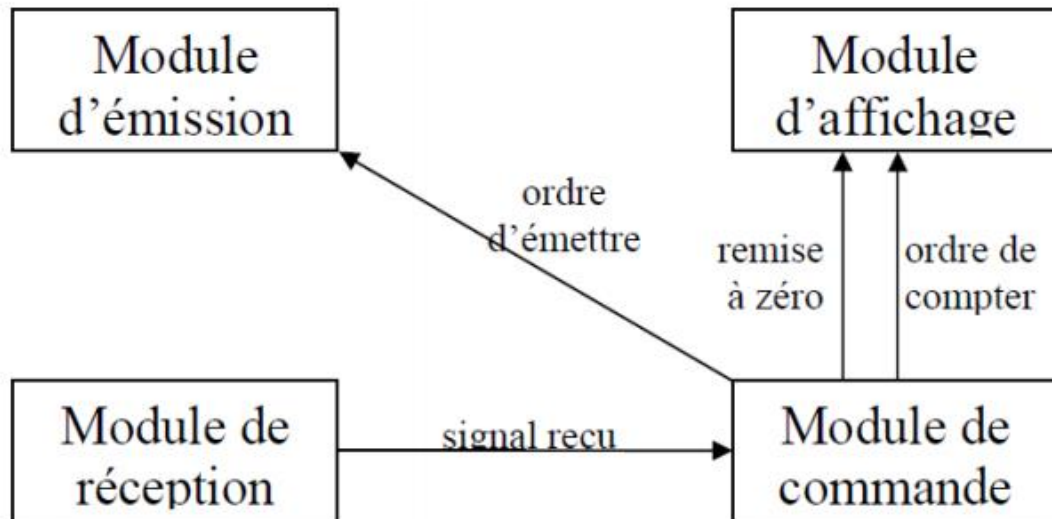


Figure. I.11. Schéma synoptique du télémètre à ultrasons

Chapitre II : Description de la Partie Matérielle et Logicielle

II-1 Introduction

L'objectif de ce chapitre est de mettre l'accent sur la description des différents matériels et logiciels qu'on a utilisés pour la réalisation, programmation et création d'une application Androïde. A savoir :

- 1-La carte Arduino
- 2- Les Capteurs
- 3- L'Environnement Arduino, Fritzing, IDE,APP Inventor

- **II-2 Description de la partie matérielle :**
- **II-2-1 La carte Arduino UNO :**
- *II-2-1-1 Présentation de la carte :*

Arduino est une plate-forme de prototypage d'objets interactifs à usage créatif constituée d'une carte électronique et d'un environnement de programmation. Sans tout connaître ni tout comprendre de l'électronique, cet environnement matériel et logiciel permet à l'utilisateur de formuler ses projets par l'expérimentation directe avec l'aide de nombreuses ressources disponibles en ligne.

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) [5] [7].

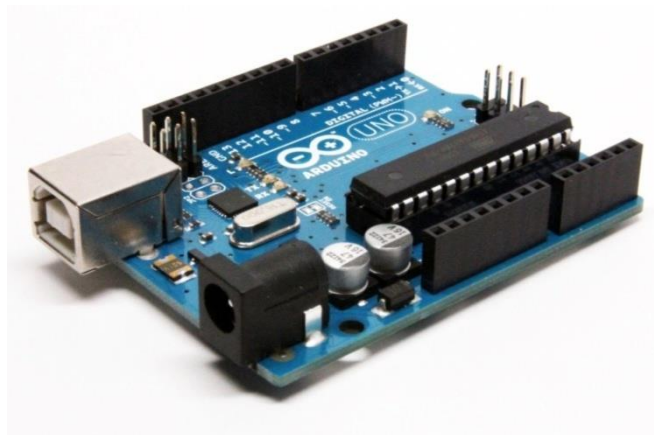


Figure II.1. La carte Arduino UNO

II-2-1-2 Caractéristiques de la carte Arduino UNO :

Micro contrôleur	ATmega328
Tension d'alimentation interne	5V
Tension d'alimentation (recommandée)	7 à 12V, limites = 6 à 20 V
Entrées/sorties numériques	14 dont 6 sorties PWM
Entrées analogiques	6
Courant max par broches E/S	40 mA
Courant max sur sortie 3,3V	50mA
Mémoire Flash	32 KB dont 0.5 KB utilisée par le
	bootloader
Mémoire SRAM	2 KB
Mémoire EEPROM	1 KB
Fréquence horloge	16 MHz

• II-2-1-3 Les différents composants de la carte Arduino UNO :

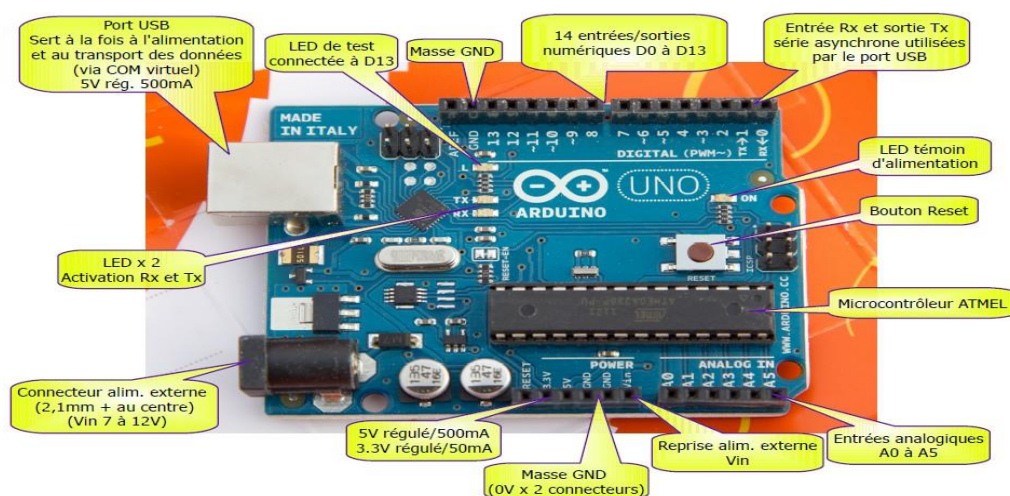


Figure II.2. Les composants de la carte Arduino UNO

II-2-1-4 Les sources d'alimentation de la carte Arduino UNO

On peut distinguer deux genres de sources d'alimentation (Entrée Sortie) et cela comme suit :

VIN La tension d'entrée positive lorsque la carte **Arduino** est utilisée avec une source de tension externe (à distinguer du 5V de la connexion USB ou autre source 5V régulée). On peut alimenter la carte à l'aide de cette broche, ou, si l'alimentation est fournie par le jack d'alimentation, accéder à la tension d'alimentation sur cette broche.

La tension régulée utilisée pour faire fonctionner le microcontrôleur et les autres composants de la carte (pour info : les circuits électroniques numériques nécessitent une tension d'alimentation parfaitement stable dite "tension régulée" obtenue à l'aide d'un composant appelé un régulateur et qui est intégré à la carte **Arduino**).

Le 5V régulé fourni par cette broche peut donc provenir soit de la tension d'alimentation VIN via le régulateur de la carte, ou bien de la connexion USB (qui fournit du 5V régulé) ou de tout autre source d'alimentation régulée.

Une alimentation de 3.3V fournie par le circuit intégré FTDI (circuit intégré fait l'adaptation du signal entre le port USB de votre ordinateur et le port série de l'ATmega) de la carte est disponible: ceci est intéressant pour certains circuits externes nécessitant cette tension au lieu du 5V.

L'intensité maximale disponible sur cette broche est de 50mA [6].

- **1 Les entrées /sorties :**

Cette carte possède 14 broches numériques (numérotée de 0 à 13) peut être utilisée soit comme une entrée numérique, soit comme une sortie numérique, en utilisant les instructions `pinMode`, `digitalWrite` et `digitalRead` du langage Arduino. Ces broches fonctionnent en 5V. Chaque broche peut fournir ou recevoir un maximum de 40mA d'intensité et dispose d'une résistance interne de "rappel au plus" (pull-up) (déconnectée par défaut) de 20-50 KOhms. Cette résistance interne s'active sur une broche en entrée à l'aide de l'instruction `digitalWrite` (broche, HIGH). En plus, certaines broches ont des fonctions spécialisées :

- **2 Interruptions Externes :**

- Les broches 2 et 3 peuvent être configurées pour déclencher une interruption sur une valeur basse, sur un front montant ou descendant, ou sur un changement de valeur.
- Impulsion PWM (largeur d'impulsion modulée) : Broches 3, 5, 6, 9, 10, et 11 fournissent une impulsion PWM 8-bits à l'aide de l'instruction `analogWrite`.
- **SPI (Interface Série Périphérique):**

Broches 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). Ces broches supportent la communication SPI (Interface Série Périphérique) disponible avec la librairie pour communication SPI. Les broches SPI sont également connectées sur le connecteur ICSP qui est mécaniquement compatible avec les cartes Mega.

- **I2C :**

Broches 4 (SDA) et 5 (SCL) supportent les communications de protocole I2C (ou interface TWI (Two Wire Interface - Interface "2 fils"), disponible en utilisant la librairie `Wire/I2C` (ou `TWI - Two-Wire interface - interface "2 fils"`).

- **LED :**

A la broche 13, Il y a une LED incluse dans la carte connectée. Lorsque la broche est au niveau HAUT, la LED est allumée, lorsque la broche est au niveau BAS, la LED est éteinte. La carte UNO dispose 6 entrées analogiques (numérotées de 0 à 5), chacune pouvant fournir une mesure d'une résolution de 10 bits (c.à.d. sur 1024 niveaux soit de 0 à 1023) à l'aide de la très utile fonction `analogRead` du langage Arduino. Par défaut, ces broches mesurent entre le 0V (valeur 0) et le 5V (valeur 1023), mais il est possible de modifier la référence supérieure de la plage de mesure en utilisant la broche AREF et l'instruction `analogReference` du langage Arduino. La carte Arduino UNO intègre un fusible qui protège le port USB de l'ordinateur contre les surcharges en intensité (le port USB est généralement limité à 500mA en intensité). Bien que la plupart des ordinateurs aient leur propre protection interne, le fusible de la carte fournit une couche supplémentaire de protection. Si plus de 500mA sont appliqués au port USB, le fusible de la carte coupera automatiquement la connexion jusqu'à ce que le court-circuit ou la surcharge soit stoppé [5].

• 3 Les ports de communications :

La carte Arduino UNO a de nombreuses possibilités de communications avec l'extérieur. L'Atmega328 possède une communication série UART TTL (5V), grâce aux broches numériques 0 (RX) et 1 (TX). On utilise (RX) pour recevoir et (TX) transmettre (les données séries de niveau TTL). Ces broches sont connectées aux broches correspondantes du circuit intégré ATmega328 programmé en convertisseur USB – vers – série de la carte, composant qui assure l'interface entre les niveaux TTL et le port USB de l'ordinateur. Comme un port de communication virtuel pour le logiciel sur l'ordinateur, La connexion série de l'Arduino est très pratique pour communiquer avec un PC, mais son inconvénient est le câble USB, pour éviter cela, il existe différentes méthodes pour utiliser ce dernier sans fil :

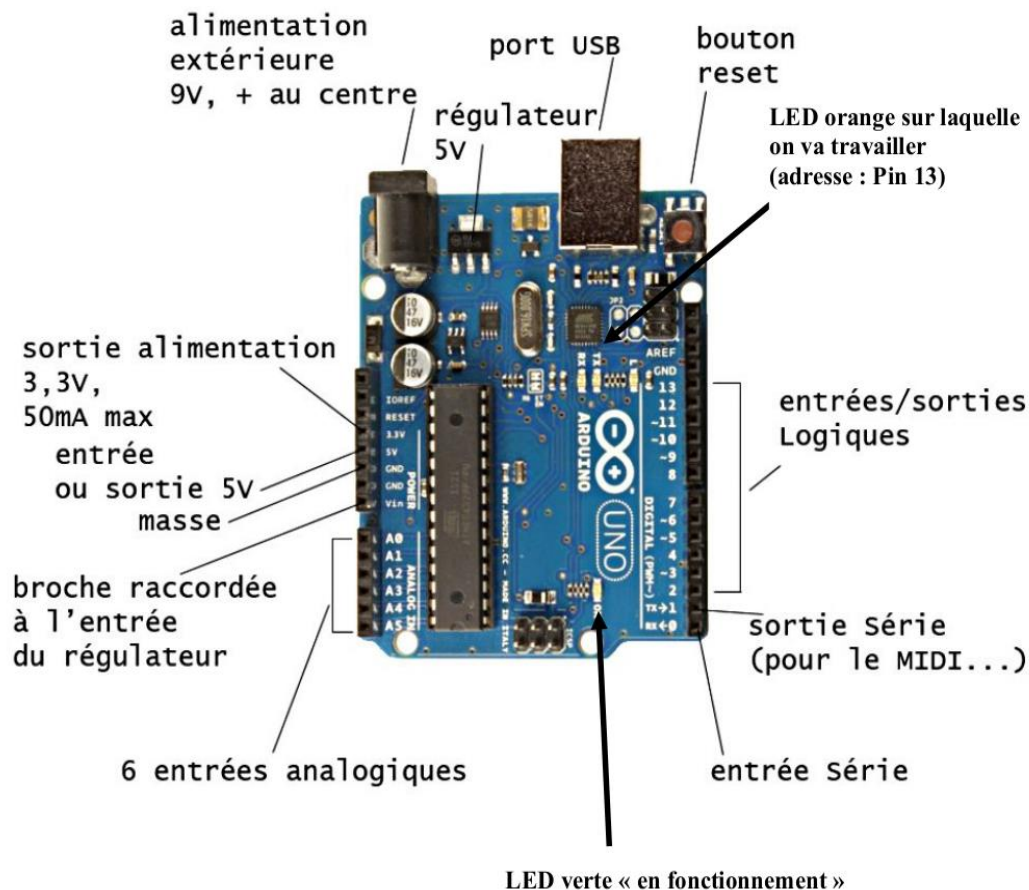


Figure II.3. Constitution de la carte Arduino UNO

• II-2-2 capteur ultrason (HC-SR04) :

Le HC-SR04 est un capteur qui utilise les ultrasons pour déterminer la distance d'un objet de 2cm à 400cm. La mesure s'effectue "sans contact" et dispose d'une précision de 3mm dans les conditions optimales. Le module inclus un émetteur et un récepteur ultrason ainsi qu'un système de contrôle [14].

Nombre de pattes	4
Dimension	45mm x 20mm x 15mm
Alimentation max	5.5 V DC
Alimentation min	4.5 V DC
Plage de mesure	2cm à 400cm
Courant de fonctionnement max	10 mA
Courant de fonctionnement min	20 mA
Résolution de la mesure	0.3 cm
Angle de mesure	15°
Largeur d'impulsion sur l'entrée de déclenchement	10 µs
La fréquence des ultrasons	40 KHZ
TRIG	Entrée de déclenchement de la mesure (trigger IN PUT)
ECHO	Sortie de mesure donnée en écho (ECHO OUT PUT)
GND	Masse de l'alimentation



Figure II.4. Capteur Ultrason HC-SR04

II-2-2-1 Principe de fonctionnement :

- 1- Nous envoyons une impulsion sur l'entrée Trig.
- 2- Le module émet une onde sonar composée d'une série de 8 impulsions à 40 kHz.
- 3- Quand le signal revient, la sortie ECHO passe au niveau haut durant toute la période où l'onde voyage vers l'objet et revient après avoir été réfléchi par se dernier.

Le calcul de la distance parcourue par un son s'effectue par l'équation suivante :

$$\text{Distance} = \text{vitesse} \times \text{temps} \quad (D = V \times T)$$

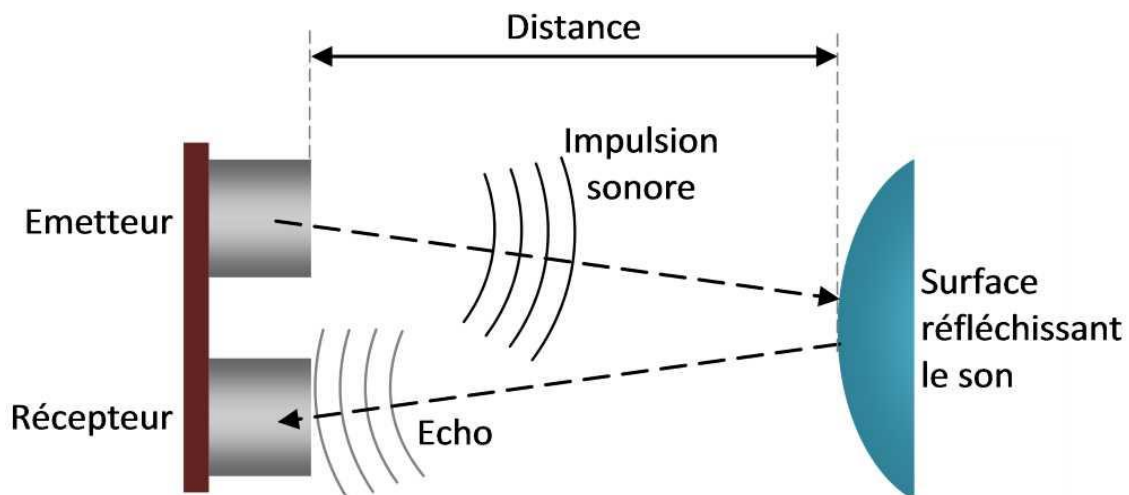
V : c'est la vitesse de la lumière (34'000 cm/1'000'000 μ s)

T : c'est le temps écoulé par les ondes ultrason aller /retour, le HC-SR04 donne une durée d'impulsion en dizaines de μ s. donc Il faut multiplier la valeur obtenue par 10 μ s pour obtenir le temps T (T= valeur x 10 μ s)

Puisque le son fait un aller-retour. La distance vaut donc la moitié

$$D = 34'000 \text{ cm/1'000'000 } \mu \text{ s} * \text{valeur} \times 10 \mu \text{ s} / 2$$

En simplifiant : $D = 17/100 \text{ cm} * \text{valeur}$ [23].



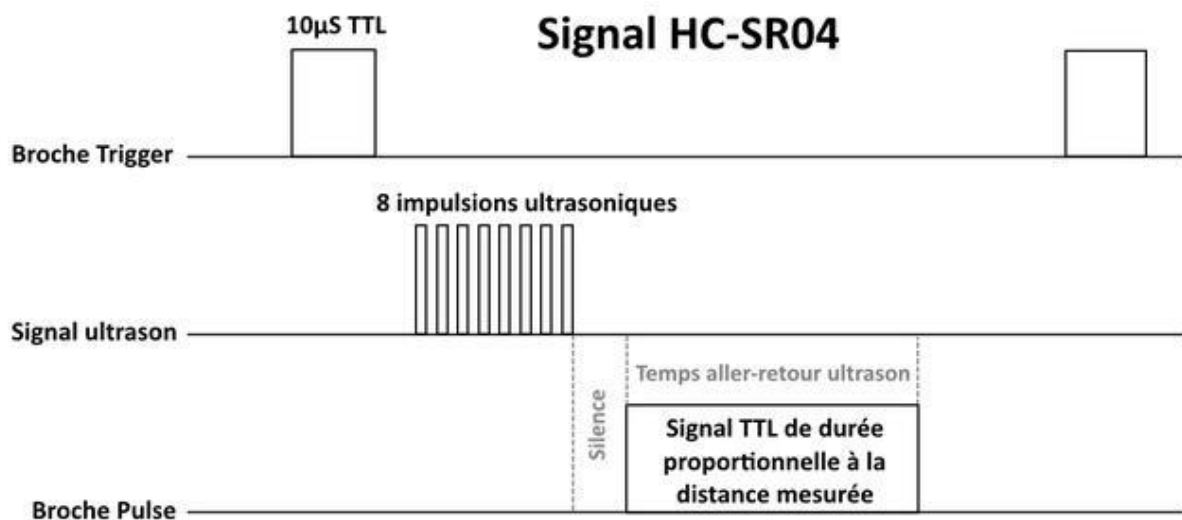


Figure II.5. Principe de fonctionnement d'un capteur Ultrason HC-SH04

II-2-3 Bluetooth HC-05 :

- *II-2-3-1 Description :*

Le Bluetooth est un protocole de communication sans fil, permettant l'échange bidirectionnel de données à une distance bien précise (15 à 20 mètres), en utilisant des ondes radio UHF sur une bande de fréquence de 2.4 GHz. Son objectif est de simplifier les connexions entre les appareils électroniques en supprimant les liaisons filaires.

La norme « Bluetooth » fut mise au point en 1994 par le fabricant Suédois Ericsson à Lund. Le terme signifie littéralement en français : « dent bleue » [13]. Le module HC-05 n'est pas plus gros qu'un pouce. Il est en fait un montage d'un module Bluetooth sur un petit P.C.B, cela permet de s'affranchir de certaines contraintes comme la soudure du module (qui est très délicate), la conversion 5V et 3.3V, la régulation de l'alimentation (3.3V de nouveau) ou encore l'ajout de LED de signal, tout cela est déjà intégré.



Figure II.6. Vue du Module Bluetooth HC-05

Le Bluetooth est utilisé comme un standard de communication permettant l'échange bidirectionnel de donnée entre la carte Arduino UNO et le Smartphone. La figure ci-dessous montre le schéma synoptique de cette communication.



Figure II.7. Schéma synoptique de la communication Bluetooth-Android

- **Le logiciel Arduino :**

Le logiciel de programmation de la carte Arduino de code (langage proche de 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.

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

- **II-3 Présentation de «IDE » :**

Le logiciel Arduino est un environnement de développement (IDE) open source et gratuit, téléchargeable sur le site officiel d'Arduino.

L'IDE Arduino permet :

- D'éditer un programme : des croquis (sketch en Anglais),
- De compiler ce programme dans le langage « machine » de l'Arduino,
- De téléverser le programme dans la mémoire de l'Arduino,
- De communiquer avec la carte Arduino grâce au terminal, Des boutons de commande en haut, une page blanche vierge, une bande noire en bas.

A l'ouverture, l'interface visuelle du logiciel ressemble à ceci :



Figure II.6. Interface IDE

• II-3-1 Programmer avec Arduino :

Un programme utilisateur Arduino est une suite d'instructions élémentaires sous forme textuelle, ligne par ligne.

La carte lit puis effectue les instructions les unes après les autres, dans l'ordre défini par les lignes de code.

II-3-1-1 Structure d'un programme :

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

Structure d'un programme

Il y a trois phases consécutives:

Commentaires multilignes pour se souvenir du pin ==>

Commentaires

1/La définition des constantes et des variables

```
int ledPin = 13; // LED connectée à la broche 13
```

2/La configuration des entrées et sorties

```
void setup()
{
  pinMode(ledPin, OUTPUT); // configure ledPin comme une sortie
}
```

3/La programmation des interactions et comportements

```
void loop()
{
  digitalWrite(ledPin, HIGH); // met la sortie à l'état haut (led allumée)
  delay(3000); // attente de 3 secondes
  digitalWrite(ledPin, LOW); // met la sortie à l'état bas (led éteinte)
  delay(1000); // attente de 1 seconde
}
```

Une fois la dernière ligne exécutée, la carte revient au début de la troisième phase et recommence sa lecture et son exécution des instructions successives. Et ainsi de suite.

Cette **boucle** se déroule des milliers de fois par seconde et anime la carte.

Done compiling.

22

Figure II.7. Structure du programme IDE

II-3-2 Syntaxe des commandes Arduino :

(Syntaxe en marron, paramètres utilisateur en vert)

```
/* Ce programme fait clignoter une LED branchée sur la broche 13
 * et fait également clignoter la diode de test de la carte
 */

int BrocheLED = 13; // Définition de la valeur 13 et du nom de la broche à
utiliser

void setup()
{
  pinMode(BrocheLED, OUTPUT); // configure BrocheLED comme une
sortie
}

void loop()
{
  digitalWrite(BrocheLED, HIGH); // met la sortie num. à l'état haut (led
allumée)
  delay(3000); // attente de 3 secondes
  digitalWrite(BrocheLED, LOW); // met la sortie num. à l'état bas (led
éteinte)
  delay(1000); // attente de 1 seconde
}
```


II-3-2-1 Définition des variables :

Pour notre montage, on va utiliser une sortie numérique de la carte, qui est par exemple la 13^{ème} sortie numérique. Cette variable doit être définie et nommée ici: on lui donne un nom arbitraire BrocheLED .

Le mot de la syntaxe est pour désigner un nombre entier est int Configuration des entrées-sorties void setup()

Les broches numériques de l'Arduino peuvent aussi bien être configurées en entrées numériques ou en sorties numériques. Ici on va configurer BrocheLED en sortie.

pinMode (nom, état) est une des quatre fonctions relatives aux entrées-sorties numériques.

Programmation des interactions void loop()

Dans cette boucle, on définit les opérations à effectuer, dans l'ordre:

- digitalWrite (nom, état) est une autre des quatre fonctions relatives aux entrées-sorties numériques.
- delay(temp en millisecondes) est la commande d'attente entre deux autres instruction
- Chaque ligne d'instruction est terminée par un point virgule
- Ne pas oublier les accolades, qui encadrent la boucle.

La figure ci-dessous vous montre un exemple de programme bien structurée :

(Syntaxe en marron, paramètres utilisateur en vert)

```
/* Ce programme fait clignoter une LED branchée sur la broche 13
 * et fait également clignoter la diode de test de la carte
 */

int BrocheLED = 13; // Définition de la valeur 13 et du nom de la broche à
utiliser

void setup()
{
  pinMode(BrocheLED, OUTPUT); // configure BrocheLED comme une
  sortie
}

void loop()
{
  digitalWrite(BrocheLED, HIGH); // met la sortie num. à l'état haut (led
  allumée)
  delay(3000); // attente de 3 secondes
  digitalWrite(BrocheLED, LOW); // met la sortie num. à l'état bas (led
  éteinte)
  delay(1000); // attente de 1 seconde
}
```

II-3-3 Fritzing :

Fritzing est un logiciel open-source multiplateforme permettant de construire des schémas et des circuits électroniques que nous utilisons avec Arduino. Plusieurs vues sont disponibles : platine d'essai, schémas électriques et circuit imprimé. Le projet dispose d'un site web, et se veut un outil qui permet aux utilisateurs de documenter leurs prototypes et les partager avec d'autres

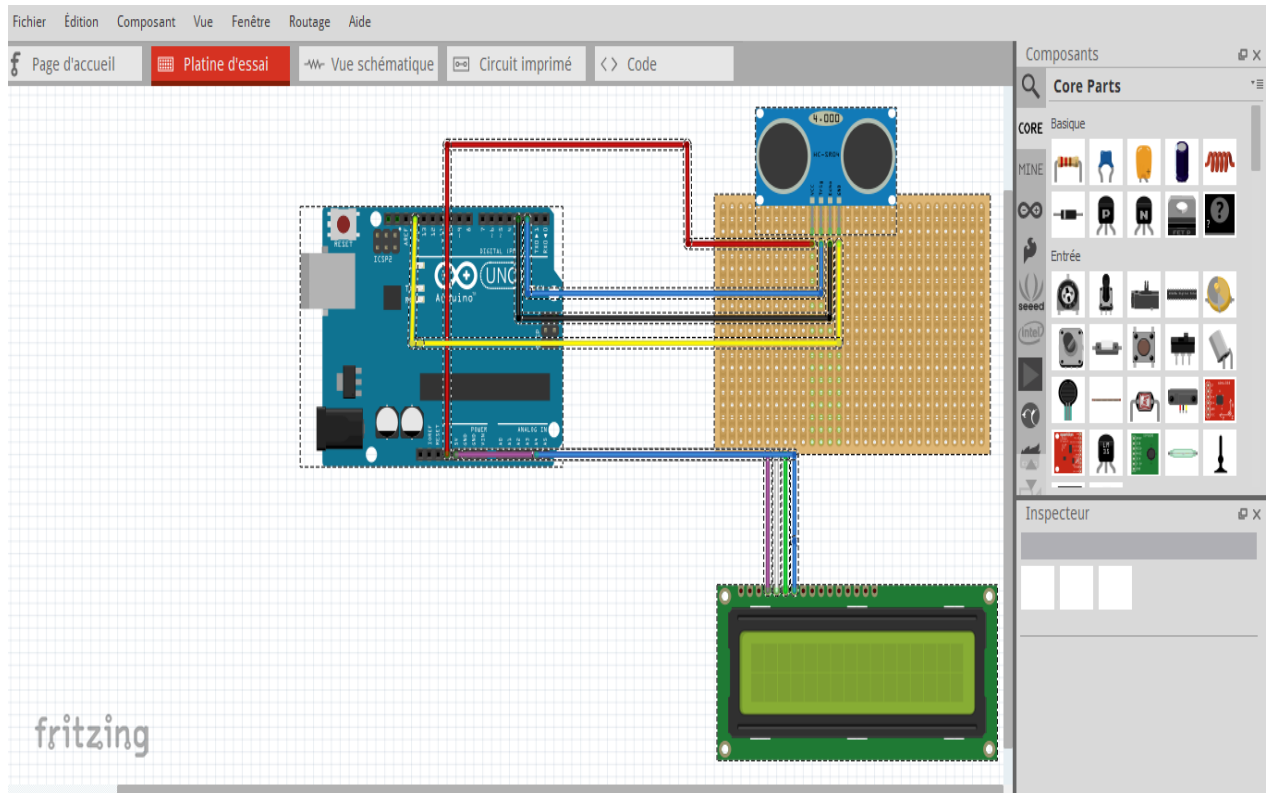


Figure II.8. L'interface de Fritzing

II-4 MIT App Inventor :

App Inventor pour Android est une application développée par Google. Elle est actuellement entretenue par le Massachusetts Institute of Technology (MIT).

Elle simplifie le développement des applications sous Android et le rend accessible même pour les novices et ceux qui ne sont pas familiers avec les langages de programmation. Elle est basée sur une interface graphique similaire à Scratch et à celle de StarLogo TNG. Grâce à son interface entièrement graphique et à l'absence totale de ligne de code, elle est particulièrement adaptée à l'initiation des enfants à la programmation, et ce dès l'école primaire.

Google publie l'application le 15 décembre 2010 et met fin à son activité le 31 décembre 2011. Dès l'été 2011, Google travaille sur un projet similaire Blockly¹, développé cette fois en JavaScript. Depuis le retrait de Google, c'est le centre d'études mobiles au MIT qui gère le support technique de cette application sous le nouveau nom "MIT App Inventor".

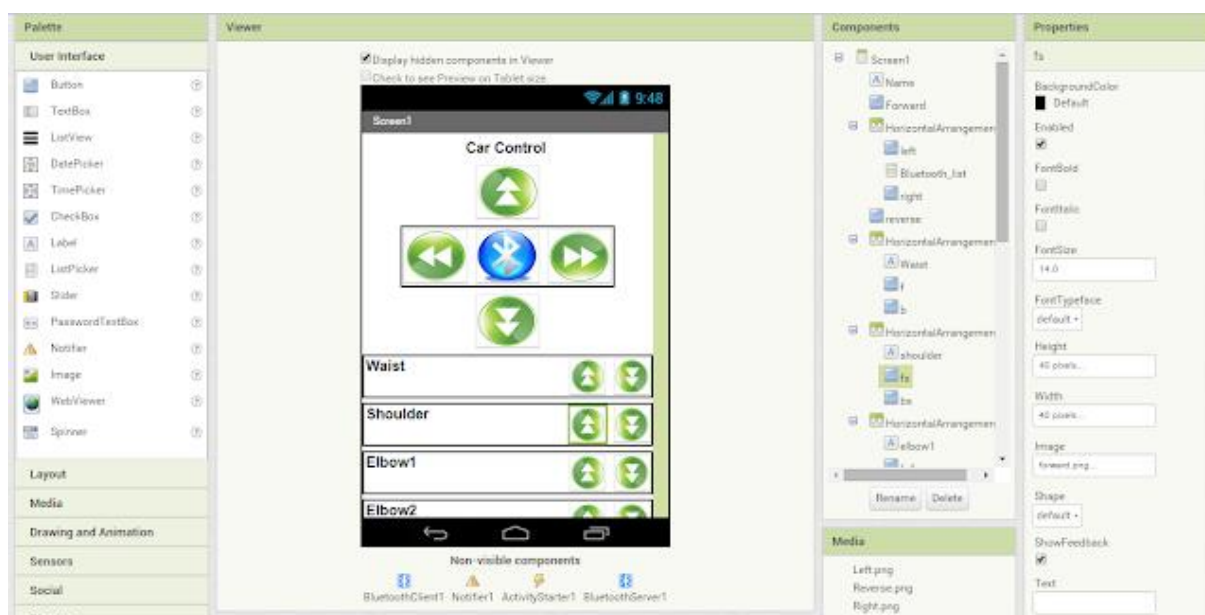


Figure II.9. Structure Mit app inventor



Figure II.10. Interface de programmation par bloc

II-4-1 Historique de logiciel App Inventor

2009 : Début du développement du logiciel App Inventor par Google à partir de recherches dans l'enseignement de l'informatique faites par le MIT, Boston près de New York. L'objectif de l'enseignement permet à des étudiants qui débutent en informatique d'apprendre à programmer sans se noyer sous le code Java.

2011: Google rend APP INVENTOR open source. Le MIT poursuit le développement.

2012 : Version beta APP INVENTOR diffusé par le MIT encore en version beta aujourd'hui.

II-4-2 ligne de code :

Comme l'outil n'utilise pas une seule ligne de code, il ne requière aucune connaissance particulière de la programmation.

C'est donc un outil idéal pour les néophytes et milieux scolaires.

En plus de réaliser des applications, il permet également de s'initier à la programmation.

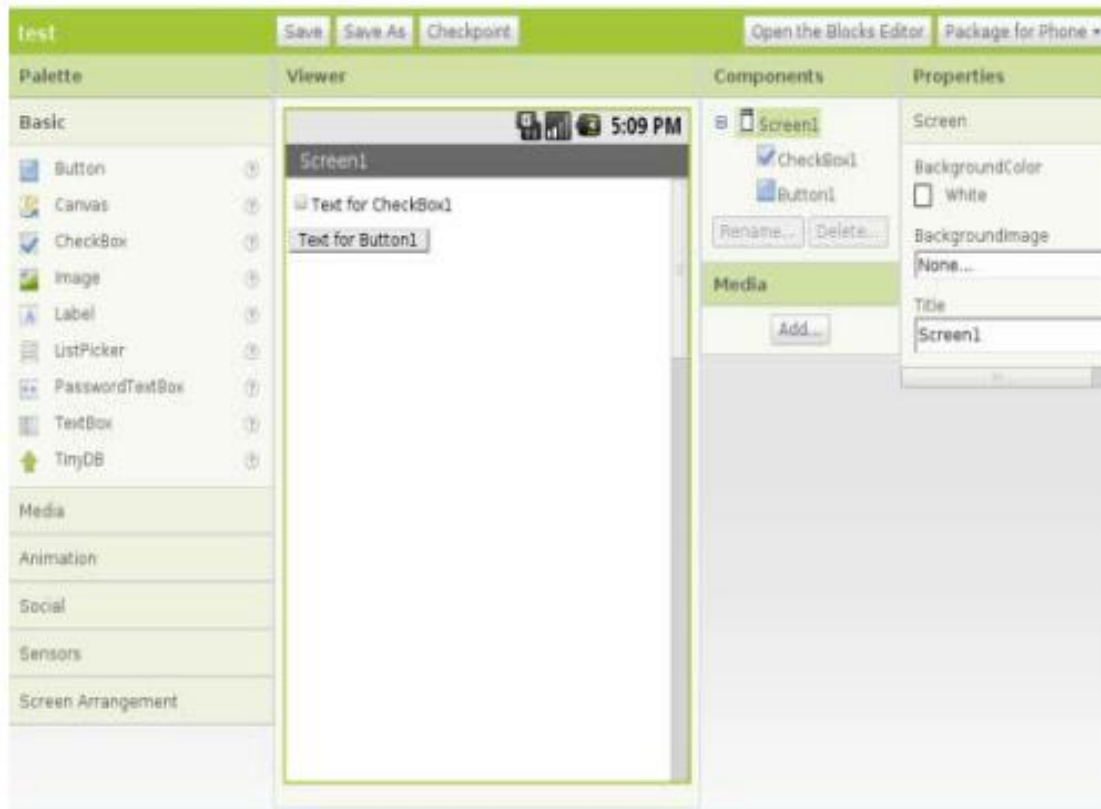
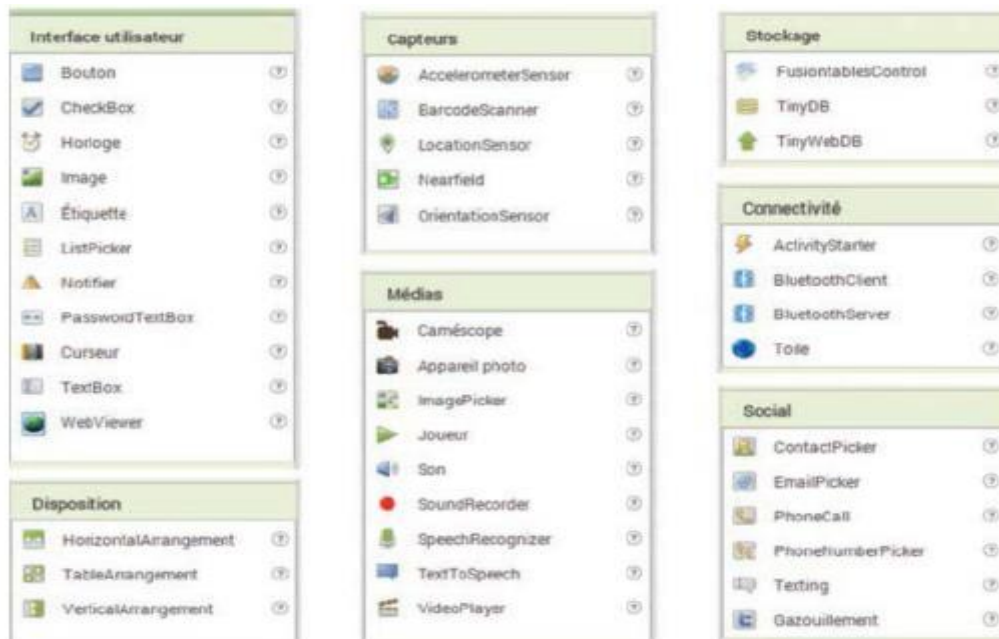


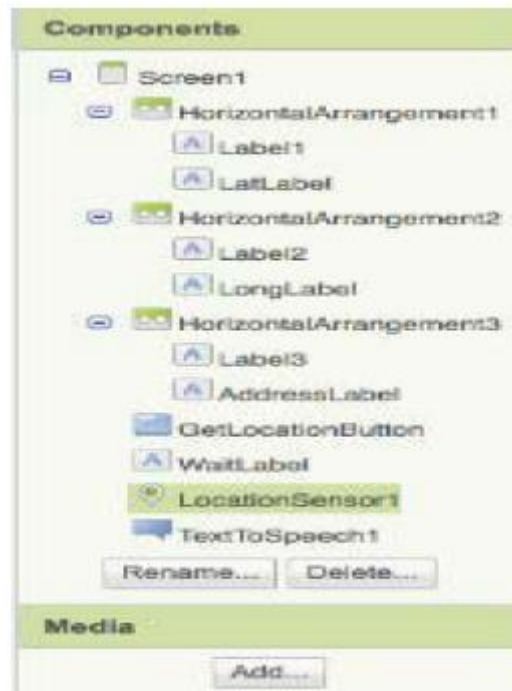
Figure II-11: Première interface de la création APP INVENTOR

Partie 1 : Les Composants Graphiques :

Partie 2 : C'est la surface du téléphone ajusté automatiquement par APP INVENTOR ou manuellement par nous même en utilisant le composant « Screene arrangement ».



Partie 3 : La liste des éléments et des medias utilisés sur l'écran.



Partie 4 : Les propriétés des différents éléments utilisés par exemple la couleur et la taille du bouton ou texture.



Pour ajouter un bouton sur l'écran d'émulateur virtuel, on clique sans lâcher sur le mot « Bouton » dans la palette en déplaçant la souris sur Viewer et relâcher le bouton et là un nouveau bouton va apparaître sur le Viewer.

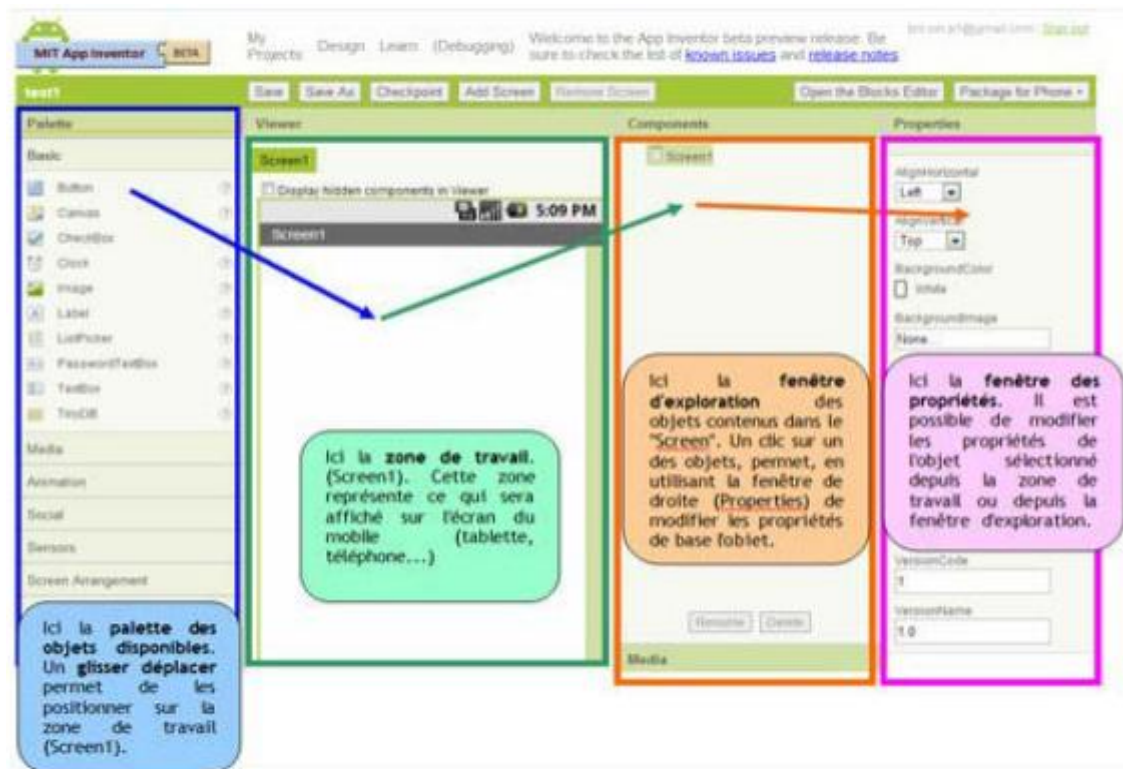


Figure II-12 : Résumé des quatre étapes

II-4-3 Visualisation directement sur SmartPhone :

En téléchargeant l'application "MIT App companion", il est possible de synchroniser l'éditeur visuel de MIT App Inventor et son smartphone. Cela permet de constater immédiatement le résultat du design sur votre smartphone.

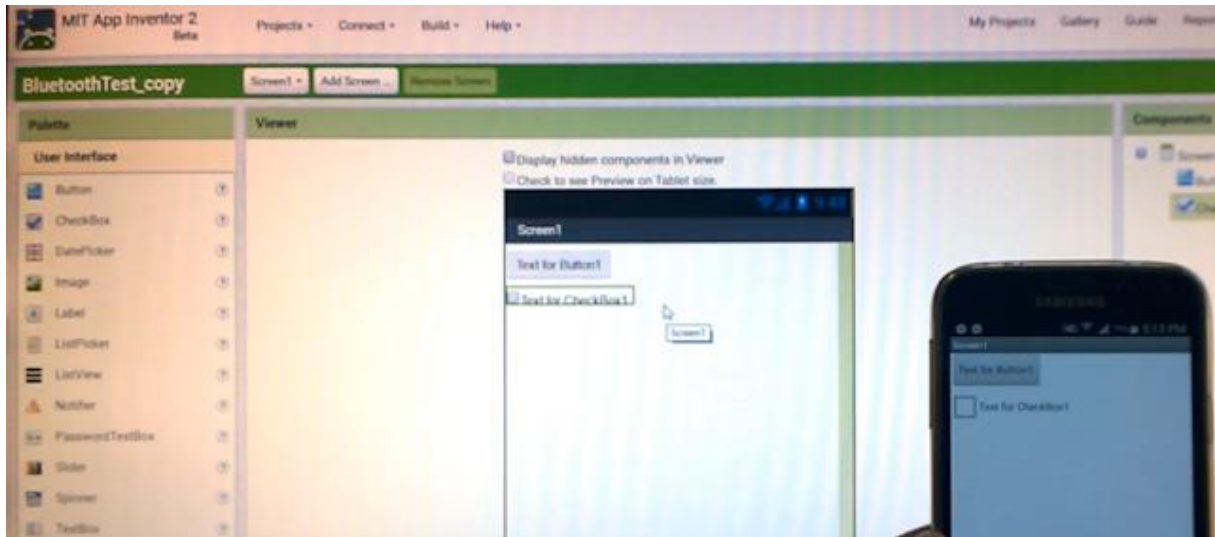


Figure II.13: Synchronisation du design avec un SmartPhone Android

II-4-4 Téléchargement :

App Inventor est un outil en ligne... il permet de compiler:

- un fichier APK à télécharger pour l'installer sur votre Smartphone.
- une application Android téléchargeable directement si votre SmartPhone dispose d'une connexion Internet (via un QR Code).

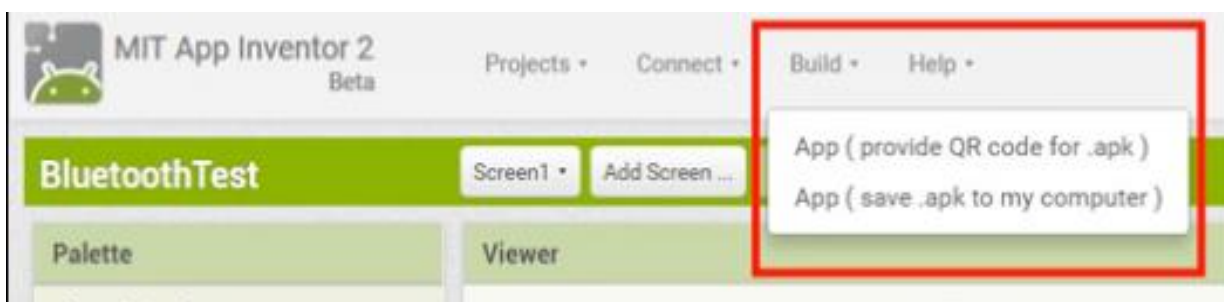


Figure II.12. L'application sur SmartPhone

- **II-4-5 Des blocs de connectivité :**

La connectivité n'est certainement pas le laissé pour compte de l'application MIT App Inventor.

La connectivité offre des services WEB, Bluetooth... pratique pour agir avec des projets locaux via Bluetooth ou projet distant via connexion Internet.

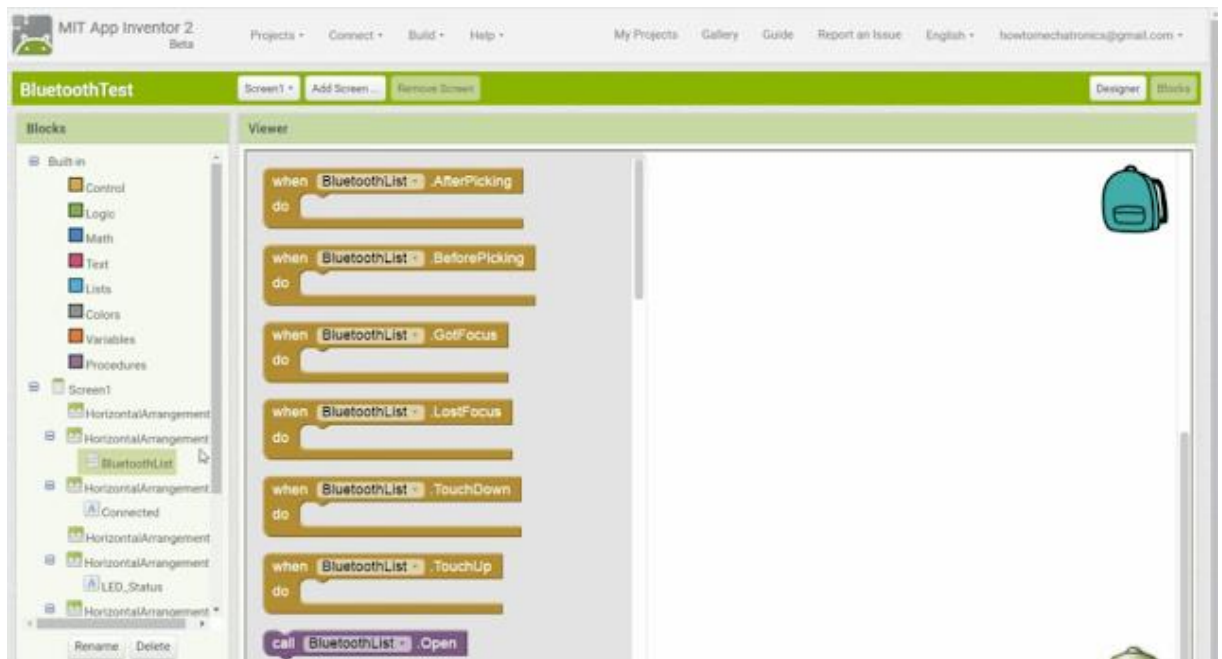


Figure II.13 : Blocs de connectivité Bluetooth dans MIT App Inventor

La connectivité Bluetooth permet de commander facilement des projets de type Arduino par l'intermédiaire d'un module Bluetooth Série HC05

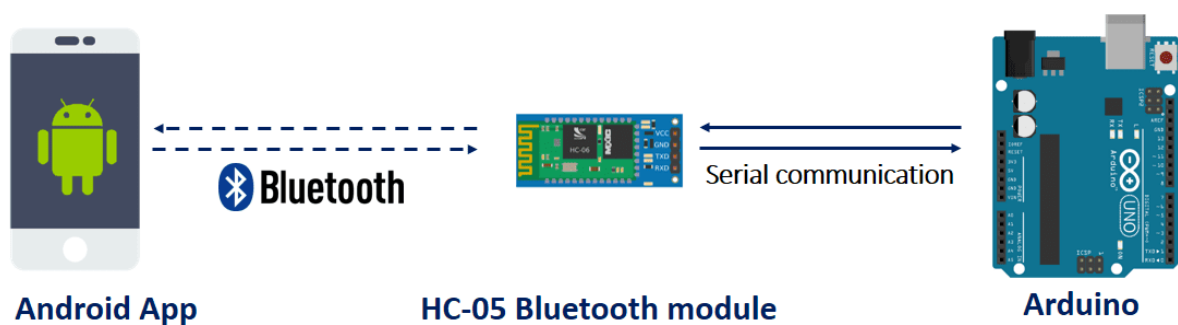


Figure II.14 : Utiliser App Inventor pour commander des projets Arduino



Figure II-16: L'organisation générale de l'interface de développement

II-5 Conclusion

Dans ce chapitre nous avons présenté les différents matériels et logiciels utilisés dans notre projet pour réaliser : la simulation, ainsi les différents programmes.

Le chapitre suivant sera consacré pour la réalisation pratique de notre projet et la simulation de système finale.

Chapitre III : Conception et Réalisation Du Télémètre à Ultrasons

Introduction :

Dans ce chapitre, on présente le dispositif expérimental « La réalisation d'un Télémètre à Ultrasons ».

Après avoir donné dans le chapitre précédent une description théorique sur la carte Arduino et le capteur à ultrason, on va procéder à l'application proprement dit. Pour cette raison, plusieurs blocs ont été nécessaires afin de réaliser une telle combinaison avec le module Arduino et son environnement de développement IDE.



Figure III.1 : Schéma synoptique de notre projet.

• III-1 Réalisation du Projet :

III-1-1 Composants utilisés :

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

- Carte Arduino Uno.
- Capteur sonar à Ultrasons HC-SR04.
- Bluetooth HC-05

III-2 Communication :

III-2-1 La connexion entre l'Arduino et le détecteur a ultrason HC-SR 04 :

- La pin 5v du HC-RS04la pin 5v
- La pin GND du HC-RS04.....la pin GND
- La pin Trig 1la pin 2
- La pin echo1.....la pin 3

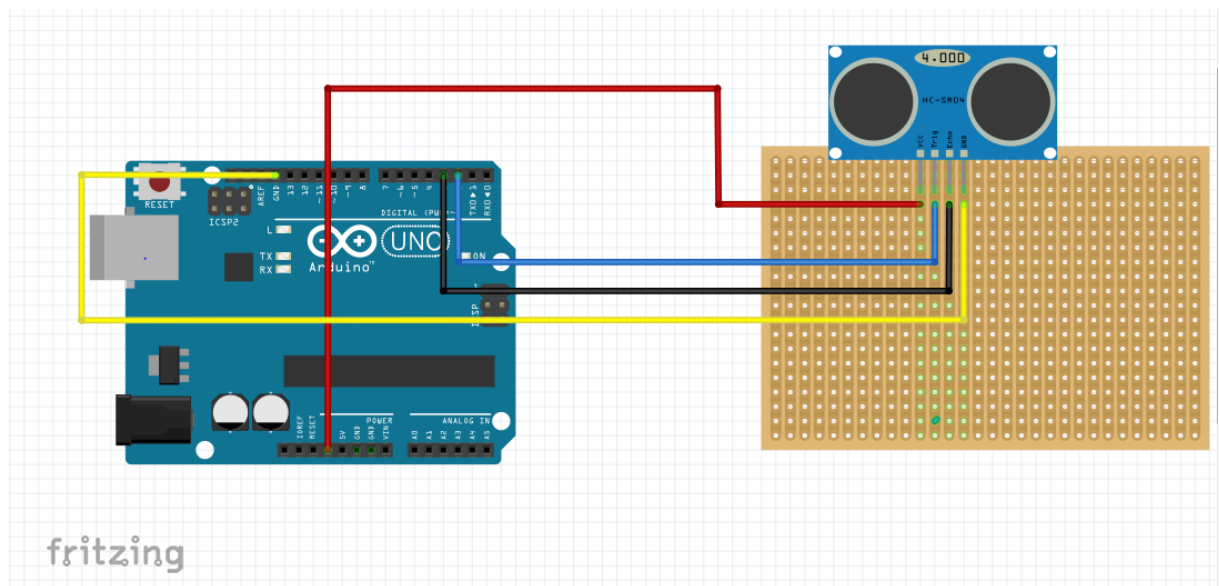


Figure III.2: La connexion entre l'arduino et le détecteur a ultrason HC-SR 04

III-2-2 La connexion entre l'Arduino et Bluetooth HC-05:

- La pin GND du HC-05.....la pin GND
- La pin SEla pin 5V
- La pin RXD.....la pin TXD1

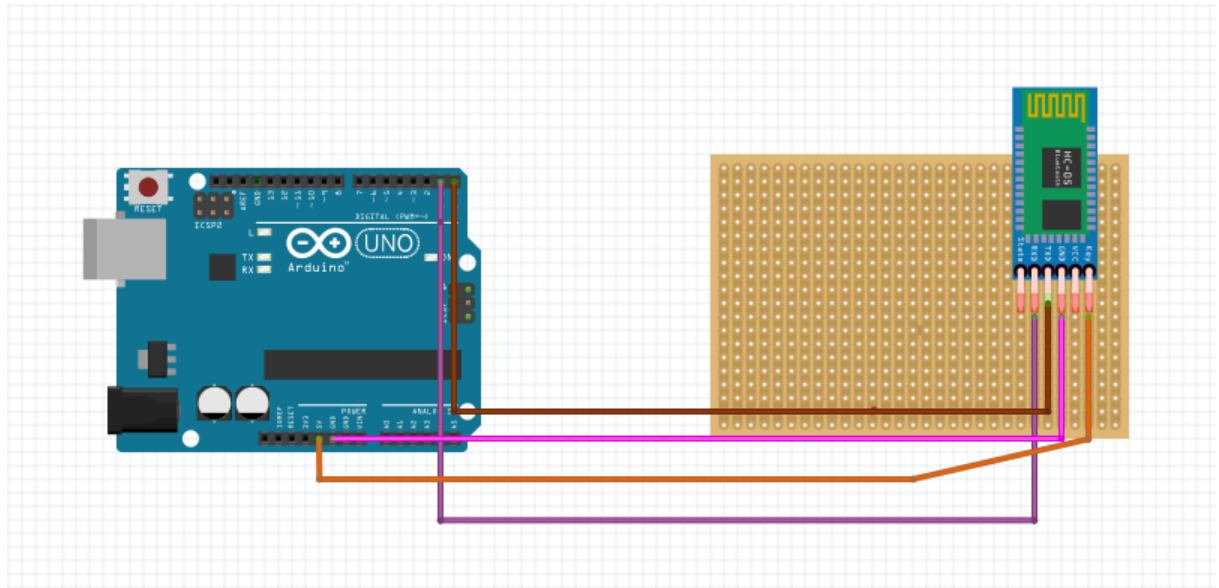


Figure III.3: La connexion entre l'Arduino et Bluetooth HC-05

III-2-3 La connexion entre l'Arduino et Bluetooth HC-05 et le détecteur a ultrason HC-SR 04 :

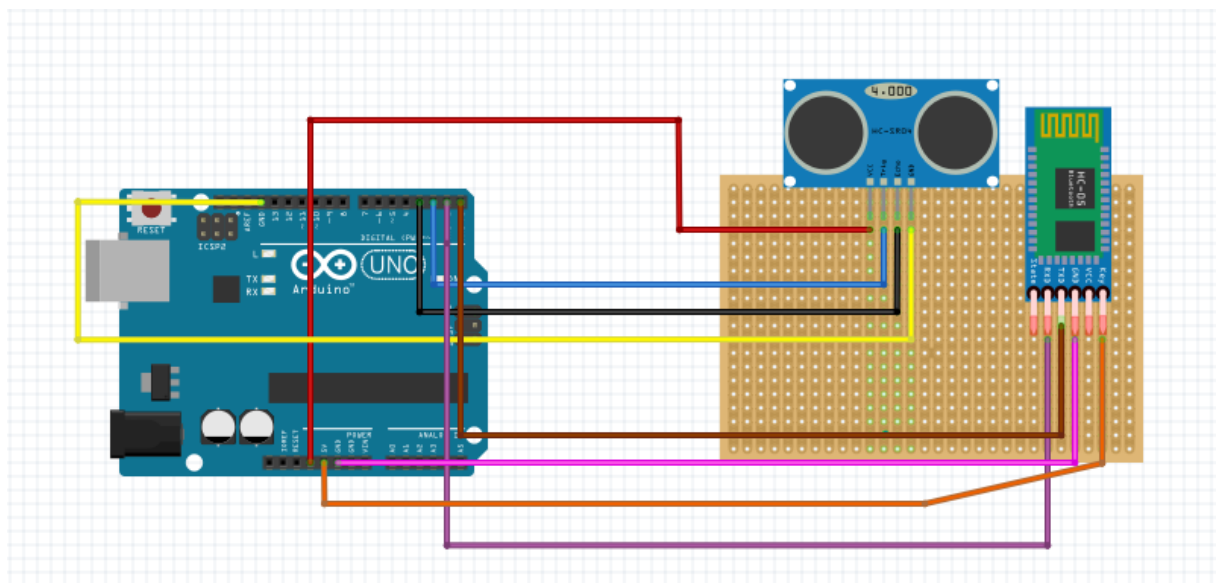


Figure III.4: La connexion entre l'Arduino et Bluetooth HC-05 et le détecteur a ultrason HC-SR 04

III-3 Le principe de notre projet :

Les télémètres à ultrasons transmettent des signaux à des fréquences ultrasonores et attendent le retour de l'écho provoqué par la réflexion de l'onde transmise. La distance est mesurée en fonction de la vitesse de propagation de l'onde et du temps entre la génération du signal et la réception de l'écho. La vitesse du son dans l'air est de 342 m/s à 20°C (331 m/s à 0°C).

- **III-4 Programmations De L'arduino :**

III-4-1 le programme :

```
const int trigPin = 9;

const int echoPin = 10;

long duration;

int distanceCm, distanceInch;

int oldValue = 0 , newValue = 0;

void setup() {

  pinMode(trigPin, OUTPUT);

  pinMode(echoPin, INPUT);

  Serial.begin(9600);

}

void loop() {

  digitalWrite(trigPin, LOW);

  delayMicroseconds(2);

  digitalWrite(trigPin, HIGH);

  delayMicroseconds(10);

  digitalWrite(trigPin, LOW);

  duration = pulseIn(echoPin, HIGH);

  distanceCm = duration * 0.0340 / 2;
```

```
newValue = distanceCm;
if(newValue != oldValue)
{
    Serial.print("Distance: ");
    Serial.print(distanceCm);
    Serial.println(" cm");
    oldValue = newValue;
}
delay(2000);
}
```

III-4-2 programme sous IDE:



```
Arduino_distance $
const int trigPin = 9;
const int echoPin = 10;
long duration;
int distanceCm, distanceInch;
int oldValue = 0, newValue = 0;

void setup() {
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  Serial.begin(9600);
}
void loop() {
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  duration = pulseIn(echoPin, HIGH);
  distanceCm = duration * 0.0340 / 2;
  newValue = distanceCm;
  if(newValue != oldValue)
  {
    Serial.print("Distance: ");
    Serial.print(distanceCm);
    Serial.println(" cm");
    oldValue = newValue;
  }
  delay(2000);
}

Compilation terminée.
Le croquis utilise 3176 octets (94) de l'espace de stockage de programmes. Le maximum est de 32256 octets.
Les variables globales utilisent 208 octets (104) de mémoire dynamique, ce qui laisse 1840 octets pour les variables locales. Le maximum est de 2048 octets.
27 Arduino Uno
```

Figure III.5 Programme télémètre sous ide

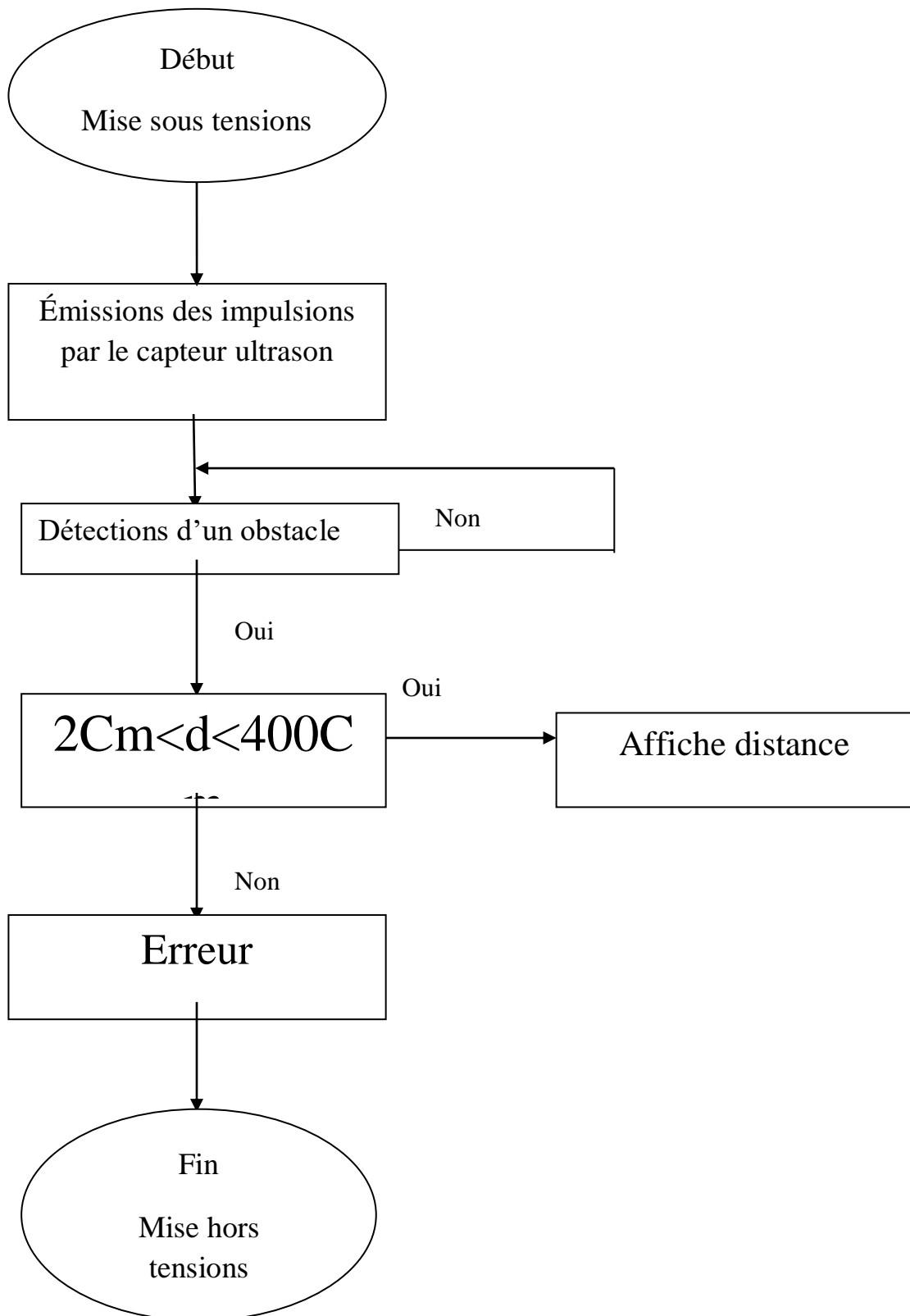


Figure III.6 : Organigramme de mesure de distance

• III-5 l'application :

L'application qui est connectée au Bluetooth va afficher la distance sur un Smartphone au lieu d'un afficheur LCD ou écran.

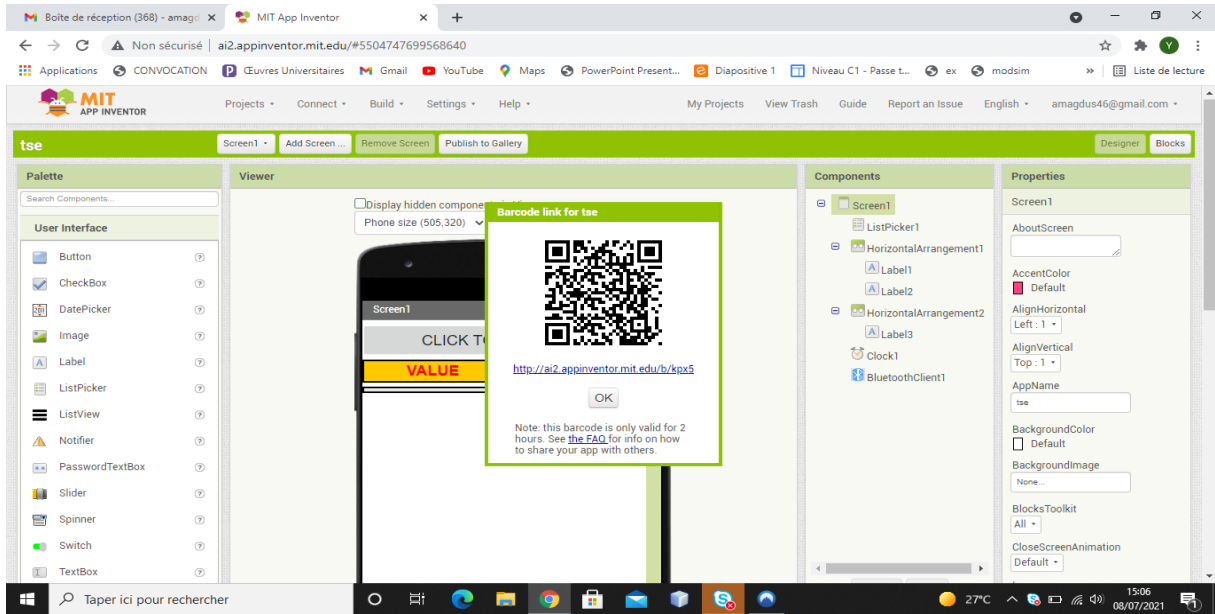


Figure III.7 Application sous App inventor

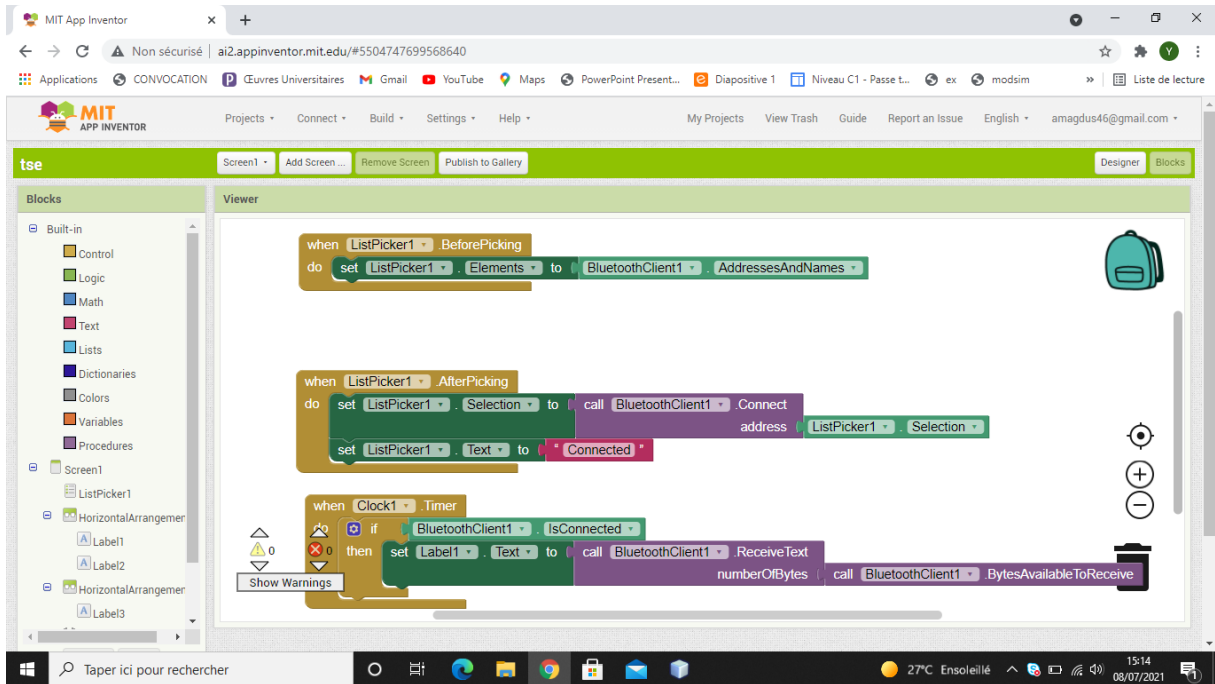


Figure III .8 block pour crée l'application

Il suffit de scanner le code QR sur le site de **MIT APP INVENTOR** pour accéder à l'application qui affichera la distance donnée par notre télémètre

L'application est connectée grâce au **Bluetooth** qui est lui-même connectée à notre **arduino**

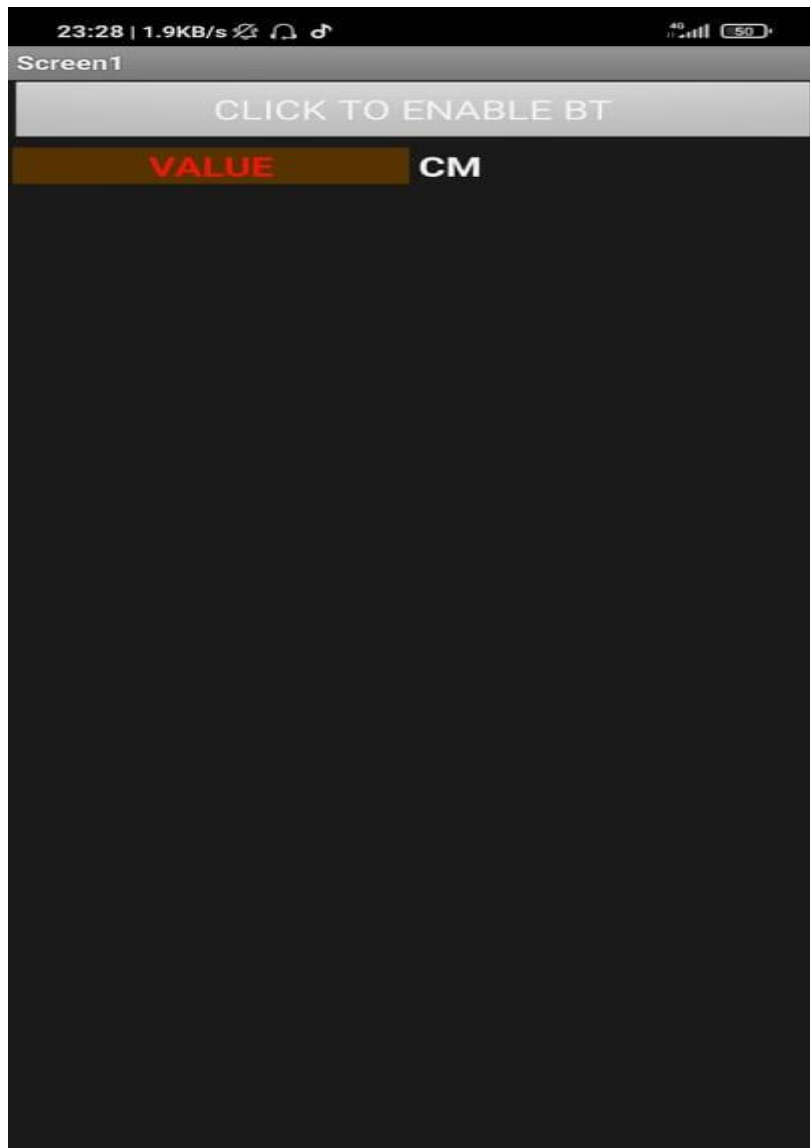


Figure III .9 Interface de l'application

III.6. Simulations

Voici les résultats de mes mesures sur une plage limitée de 10 centimètres à 100 centimètres :

Distance cm	Mesure	Erreur cm
10	10,2	0,2
20	19,7	0,3
30	29,3	0,7
40	40,1	0,1
50	49,2	0,8
60	59,7	0,3
70	69,9	0,1
80	80,1	0,1
90	89,8	0,2
100	99,5	0,5

Tab III.1 Tableau des mesures prise avec marge d'erreur



Figure III.10. Graphique de linéarité du capteur

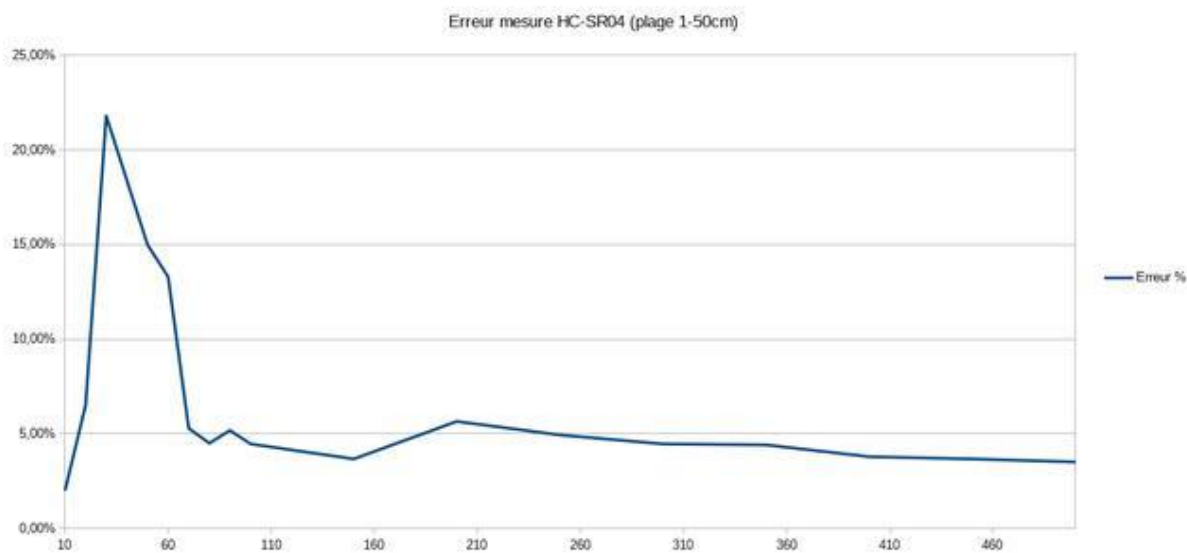


Figure III.12 Graphique d'erreurs du capteur

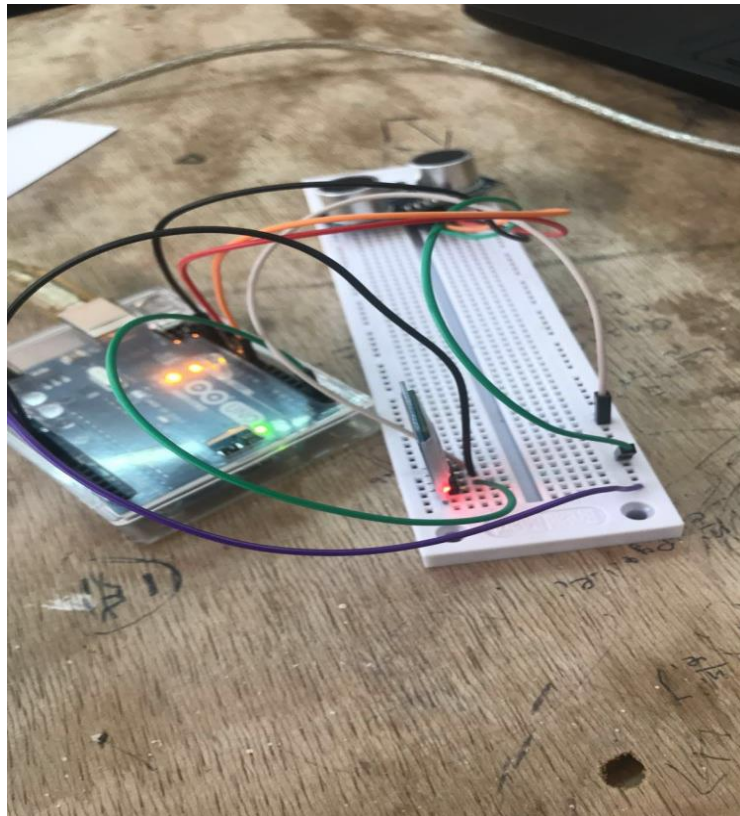


Figure III.13: Montage final d'un télémètre a ultrason avec Bluetooth

III-6 conclusion :

Ce chapitre est consacré à la réalisation du télémètre à ultrasons. Pour le faire fonctionner, tous les appareils matériels et logiciels doivent fonctionner ensemble. Concernant les matériaux utilisés, nous avons expliqué l'assemblage de chaque élément qui constitue notre système, nous avons donc présenté le programme Arduino, sa structure et comment charger le programme pour que le télémètre puisse fonctionner

- **Conclusion général :**

Ce projet nous permet d'établir un lien entre la recherche théorique des composants électroniques et sa mise en œuvre concrète, et sachez qu'il y a une différence entre les deux. En partie parce que parfois nous omettons certains paramètres dans le calcul mais en mise en œuvre nous les prenons en compte. Cette petite différence nous encourage à trouver des solutions et faire d'autres choix pour atteindre nos objectifs, ce qui nous permet d'avoir plus d'expérience dans ce domaine.

Notre projet comporte un travail théorique accompagné d'une réalisation, son objectif consiste sur l'étude d'un télémètre à ultrasons. Pour faire preuve notre réalisation, nous avons utilisé une carte Arduino UNO, un capteur à ultrason HC-SH04, Bluetooth HC-05.

Le télémètre à ultrasons est très utile dans le domaine industriel car il est très utile pour mesurer les distances de façon précise.

Comme travaux de futurs, nous voulons poursuivre ce projet en considérant ce télémètre à ultrasons: nous proposons d'ajouter un bipper et des LEDs pour réaliser un radar de recul ; plus utile dans le domaine automobile.

- **Mots clés :**

Télémètre, les ultrasons, émetteur ultrasons, récepteur ultrasons, piézoélectricité, et l'onde ultrasonore.

- **Résumé:**

Les travaux présentés dans ce projet sont principalement consacrés à la réalisation d'un télémètre à ultrasons. Le travail à effectuer est divisé en plusieurs parties. La première étape est la conception de l'installation, avec un câblage proposé pour tester et valider les solutions. La deuxième étape de la fabrication du télémètre sous forme de circuit imprimé. Enfin, il s'agit d'effectuer les réglages et la calibration du télémètre, pour vérifier son bon fonctionnement et éventuellement déterminer les limites.

Bibliographie :

- [1] <https://e-cours.univ-paris1.fr/modules/uved/envcal/html/rayonnement/1-rayonnement-electromagnetique/1-2-les-ondes-electromagnetiques.html>
 - [2] <https://ultrasonographie-vasculaire.edu.umontpellier.fr/files/2012/08/MD-Bases-Echographie-DIUE-2014B2.pdf>
 - [3] <https://eduscol.education.fr/sti/sites/eduscol.education.fr.sti/files/ressources/pedagogiques/7619/7619-5elev-apporttheoriqueultrasons.docx>
 - [4] <https://tel.archives-ouvertes.fr/tel-02983267/document>
 - [5] <http://www.generationrobots.com/fr/152-arduino>. consulter le : mars 2015.
 - [6] S.V.D.Reyvanth, G.Shirish, « PID controller using Arduino »
 - [7] <http://fr.flossmanuals.net/arduino/index>
 - [8] T.Karvinen, K.Karvinen, V.Valtokari « Les capteurs pour arduino et Raspberry Pi » DUNOD, 2014
 - [9] Lucien Bachelard, User'sManual du HC-SR04/www.manuel-esteban.com, Le 28 novembre 2015
 - [10] Hippolyte Weisslinger (olyte), Landrault (Eskimon) Arduino : Premiers pas en informatique embarquée, le Blog d'Eskimon Edition du 19 juin 2014
 - [11] Simon Monk, Fritzing for inventors : Take Your Electronics Project from Prototype to Product by Simon Monk, McGraw-Hill Education TAB ; 1 edition, 2015.
 - [12] www.fritzing.org Consulter le juin/2017
 - [13] <https://fr.wikipedia.org/wiki/Bluetooth>
 - [14] <https://www.orbit-dz.com/product/capteur-ultrason-hc-sr04/>
-