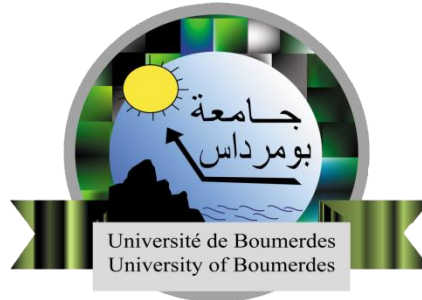


République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

UNIVERSITE M'HAMED BOUGARA-BOUMERDES



Faculté des Sciences de l'Ingénieur
Département d'Ingénierie des Systèmes Electriques

Mémoire de Master

Présenté par :

BOUGHEDADA Aymen

SAOULA Rafik

En vue de l'obtention du diplôme de Master en
Electronique des systèmes embarqués

Thème :

**Exploitation des systèmes embarqués pour la
réalisation d'une canne intelligente**

Mr BOUREKECHE

UMBB

Président

Mr SMANI

UMBB

Examineur

Mme MAHDI

UMBB

Examinatrice

M^{me} Y.GUERBAI MCB

UMBB

Directrice de mémoire

-Promotion juin 2018 -

Remerciements

Ce travail n'aurait jamais vu le jour sans la volonté de DIEU, Le Tout Puissant.

On ne voit pas comment exprimer notre plus profonde gratitude et notre éternelle reconnaissance envers DIEU, Le Plus Généreux, qui nous a offert santé, force, courage et volonté jusqu'au dernier moment. Nous remercions infiniment Allah en secret et en public pour ça, pour le fait d'être en vie, et pour tous Tes bienfaits cachés et apparents qui nous sont indénombrables.

Nos remerciements, notre gratitude, notre reconnaissance et considération, les plus sincères, vont à notre encadreur Mme Y.Guerbai. Merci de nous avoir encadrés. J'aimerais aussi vous remercier pour votre aide, votre patience, votre orientation, votre suivi, votre soutien, vos corrections, et vos précieux conseils qui nous ont permis de mener à bien ce travail.

On tient à remercier vivement le président de jury pour avoir bien voulu présider le jury de ce travail. On tient à remercier également les membres de jury pour avoir bien voulu évaluer ce travail.

Afin de n'oublier personne, Nos vifs remerciements s'adressent à tous ceux qui nous ont aidés de près ou de loin à la réalisation de ce modeste travail.

Table des matières

Table des matières.....	I
Liste des Figures.....	V
Liste des tableaux.....	VII
Liste des acronymes.....	VIII
Introduction.....	1
Problématique.....	1
Objectif.....	2

Chapitre 01 : généralité sur les systèmes embarqués

I.1.1.Historiques des systèmes embarqués	3
I.1.2.Qu'est-ce qu'un système embarqué	3
I.1.3.Caractéristiques des systèmes embarqués.....	3
I.1.4.Les contraintes des systèmes embarqués	4
I.1.5.Classification des systèmes embarqués.....	4
I.1.6.Architecture des systèmes embarqués.....	5
I.1.6.1. L'architecture Harvard et Von Neumann.....	6
I.1.6.2. Architecture Von Neumann	6
I.1.6.3. Application et caractéristiques de l'architecture Von Neumann	7
I.1.6.4. Architecture Harvard	8
I.1.6.5. Application et caractéristiques de l'architecture de Harvard.....	9
I.1.6.6. Pourquoi l'architecture de Harvard est-elle efficace ?	10
I.1.6.7. Différence entre Von Neumann et Harvard Architecture	10
I.2. Programmation :	12
I.3. Développement.....	13
I.4. Debug	15
I.5. Système d'exploitation.....	15
I.6. Les langages de programmation des systèmes embarqués.....	15

Table des matières

I.7. Domaines d'application des systèmes embarqués :	17
I.8. Les systèmes embarqués temps réel	18
I.8.1. Définition	18
I.8.2. Principale caractéristique des systèmes temps réel	19
I.8.3. Caractéristique temporelles des systèmes temps réel	19
I.9. La détection d'obstacle	20
I.9.1. Définition	20
I.10. Capteur et méthode pour la détection d'obstacles	20
I.10.1. Les télémètres	20
I.10.2. Le télémètre à ultrasons	21
I.10.3. Le radar	21
I.10.4. Le lidar	22
I.10.5. La vision monoculaire	22
I.11. Les primitives à rechercher	23
I.12. Des algorithmes de reconnaissance de forme	23
I.13. L'exploitation d'images successives	23
I.14. La combinaison de plusieurs approches	23
I.15. La stéréovision	24
I.16. Les systèmes coopératifs :	24
I.17. Conclusion	24

Chapitre 02 : Principaux composant formant le système

II.1. Arduino UNO	26
II.1.1. Définition	26
II.1.2. Schéma des ports	27
II.1.3. Caractéristique technique	27
II.1.4. Détails technique	28
II.1.4.1. Mémoire	29

Table des matières

II.1.4.2. Entrées et sorties	29
II.1.4.3. Communication	29
II.2.Breadboard.....	30
II.2.1. Définition	30
II.2.2. Les bases de Breadboard	30
II.2.3. Breadboard Connexion	32
II.3.le Capteur a ultrason	32
II.3.1. Définition.....	32
II.3.2. Caractéristiques	33
II.4.Principe de fonctionnement	33
II.4.1. Quelles sont les céramiques piézoélectriques ?.....	34
II.5.Types et formes d'objets de détection.....	34
II.6.Buzzer électronique	35
II.6.1. Définition.....	35
II.6.2. Caractéristiques techniques	35
II.7.Photorésistance	36
II.8.La Résistance	37
II.9.Module SD.....	37
II.9.1. Définition	37
II.9.2. Câblage de la carte.....	38
II.10.Le détecteur d'eau	38
II.10.1. Définition	38
II.10.2. Caractéristiques	38
II.10.3. Applications.....	39
II.10.4. Câblage du détecteur d'eau.....	39
II.11.Conclusion	39

Table des matières

Chapitre 03 : Programmation et mise en marche de notre système

III.1.Programmation	40
III.2.Aperçu du logiciel ARDUINO	40
III.2.1. Principe général d'utilisation.....	41
III.2.2. Transfert des programmes vers la carte Arduino	42
III.3.Organigramme.....	43
III.3.1. Capteur a ultrason.....	43
III.3.1.1. Mode de fonctionnement	44
III.3.2. La photorésistance.....	44
III.3.2.1. Mode de fonctionnement.....	45
III.3.3. Le capteur a ultrason avec le speaker	45
III.3.3.1. Mode de fonctionnement.....	46
III.3.4. Le Détecteur d'eau	46
III.3.4.1. Mode de Fonctionnement	47
III.4.L'organigramme général	47
III.4.1. Mode de fonctionnement.....	49
III.5.Simulation	49
III.6.Réalisation et test	50
III.7.Conclusion.....	51
Conclusion générales.....	52

Table des figures

Chapitre 01

Figure. I. 1:schéma explicatif d'un système embarqué.....	4
Figure. I. 2:architecture des systèmes embarqué.....	5
Figure. I. 3:Architecture de Von Neumann	7
Figure. I. 4:Caractéristiques de l'architecture Von Neumann	9
Figure. I 5:Caractéristiques de l'architecture de Harvard	9
Figure. I. 6:Caractéristiques de l'architecture de Harvard (2)	11
Figure. I. 7:Architecture Von Neumann et Harvard Architectures pour ma mémoire.....	12
Figure. I. 8:Principale possibilité de programmation	13
Figure. I. 9:Le processus de développement	14
Figure. I. 10 : Organigramme de développement.....	15
Figure. I. 11:Représentation schématique d'un système de contrôle commande.....	19
Figure. I. 12:Télemètre a ultrason	22
Figure. I. 13:Le radar.....	22
Figure. I.14:Le lidar.....	23
Figure. I. 15:Exemple de la technique de stéréovisions	25

Chapitre 02

Figure. II. 1:Schéma de port carte arduino UNO	27
Figure. II. 2:Breadboard (1).....	30
Figure. II. 3:Breadboard (2).....	31
Figure. II. 4:Connexion de breadboard avec différent composant(1)	32
Figure. II. 5:Connexion breadboard et différent composant(2).....	33
Figure. II. 6:Capteur a ultrason.....	33
Figure. II. 7:Explicatif des céramiques piézoélectriques.....	35
Figure. II. 8:Types et formes d'objets de détection	35
Figure. II. 9:Buzzer électronique.....	36
Figure. II. 10:Photorésistance.....	37
Figure. II. 11:Résistance.....	38
Figure. II. 12:Module SD	38
Figure. II. 13:Câblage de la carte SD	38
Figure.II.14:Le détecteur d'eau	39
Figure 15 : Câblage du détecteur d'eau.....	40

Table des figures

Chapitre 03

Figure. III.1: L'écran principal du logiciel Arduino (interface IDE).....	42
Figure. III.2: Détail de la barre de bouton.....	42
Figure. III.3: La vérification terminé.....	43
Figure. III.4 : Organigramme de calcul de distance par le capteur a ultrason ..	44
Figure. III.5: organigramme de la photorésistance.....	45
Figure. III.6 : organigramme du capteur a ultrason avec le speaker ..	47
Figure.III.7 : organigramme du détecteur d'eau ...	48
Figure. III.8 : organigramme général du programme.....	49
Figure. III.9 : Simulation du circuit électrique de notre système.....	50
Figure. III.10: photo de notre système vu d'en haut ..	51

Liste des tableaux

Tableau 01 : Les langages les plus utilisés pour le développement de systèmes embarqués.	16
Tableau 02 : caractéristique de la carte arduino UNO.....	27
Tableau 03 : Norme de lumière en lux	46

Liste des acronymes

ABS: Anti Blockier System.

ALU: Arithmetic Logic Unit.

ASIC: Application-Specific Integrated Circuit.

AREF: pin on the arduino.

CPU: Central Processing Unit.

DSP: Digital Signal Processor.

EEPROM : Electrically-Erasable Programmable Read-Only Memory ou mémoire morte effaçable électriquement et programmable.

GAB : Guichet automatique bancaire .

GND : Ground.

IDE : Integrated Development Environment.

IHM : Interface Homme machine.

JTAG : Joint Test Action Group .

LDR: light-dependent resistor.

LED : light-emitting diode.

PC : Personal compture .

RAM: Random Access Memory.

RISC: Reduced Instruction Set Computer.

ROM: Read-Only Memory.

SD : Secure Digital.

TTL : Le Time to Live.

UNO : un en italien.

USB : Universal Serial Bus.

Introduction générale

Introduction

La nature a donné à l'homme et aux animaux des organes complexes qui leur permettent d'interpréter les différentes informations de leur environnement. Ces organes peuvent capter des grandeurs physiques qui sont envoyées au cerveau pour pouvoir interpréter les changements dans le monde qui nous entoure ; ce sont les sens. L'un des phénomènes physiques capté par nos organes est "les ondes", tantôt mécaniques avec l'ouïe, tantôt électromagnétiques avec la vue, qui constituent les deux principaux sens de l'homme. Or ces sens ne captent qu'une infime partie de tout le spectre existant des ondes.

Malheureusement ces sens peuvent être endommagés, ce qui devient une contrainte pour la personne affectée. D'après l'OMS (Organisation mondiale de la santé) il y a environ 285 millions de personnes handicapées visuellement dont 39 millions sont aveugles. Heureusement, il existe différentes techniques, outils et des technologies disponibles pour permettre aux handicapés de réaliser leurs activités quotidiennes.

Un des outils les plus utilisés est la canne blanche qui permet à l'utilisateur de détecter des obstacles qui se trouvent à un mètre de lui environ et, également, de déceler l'état du sol sur lequel ils marchent. Cependant, il existe encore des déficiences pour détecter des obstacles plus hauts ainsi que pour toucher les obstacles et les reconnaître. La technologie des capteurs nous permet d'identifier des grandeurs physiques et de les transformer en informations grâce à la connaissance des phénomènes physiques qui y interviennent. La compréhension des ondes a permis de développer des capteurs qui peuvent détecter des objets à distance grâce à l'analyse des échos reçus.

Pour notre projet de fin d'étude, nous avons eu l'idée d'améliorer et de développer une canne pour les aveugles. Le difficile accès aux différents transports publics ou les mauvaises conditions des rues sont des exemples d'obstacles auxquels ils doivent faire face tous les jours.

L'aide au aveugles a été notre principale motivation et la solution que nous avons trouvée été la conception d'une canne électronique qui puisse les avertir de la présence de certains obstacles à une distance plus grande et de certains risques situés à une hauteur plus grande que celle du sol.

Problématique

Les personnes atteintes par une déficience visuelle rencontrent des difficultés spécifiques dans leur vie quotidienne. Ces répercussions demeurent largement méconnues ; elles touchent pourtant un nombre ne croissant de personnes en raison du vieillissement de la population ou d'un accident. Lorsqu'une personne est subitement privée de la vue il refuse de se faire aider car il n'accepte pas son handicap c'est une réaction classique d'un handicapé incapable de faire quoi que ce soit tout seul, il lui faut du temps pour réagir car tout devient difficile

Il faut donc qu'ils réapprennent à accomplir un grand nombre d'actes simples de la vie quotidienne, ils sont généralement aidés par leurs familles, entourage.

Puis leur revient l'envie de faire des choses et surtout, de les faire seul. Il arrive un moment où ils ne supportent plus de dépendre des autres c'est le chemin vers une vie autonome ou la canne blanche intelligente est indispensables.

Objectifs

L'objectif de la canne intelligente est de détecter les obstacles d'au moins 50 cm avec et l'obscurité ainsi la détection d'eau en utilisant des composants adaptés et utilisables dans toutes les circonstances, et de proposer différents types de moyens d'avertissements talque le buzzer l'hautparleur ou le vibreur adaptables aux préférences de l'utilisateur ainsi faciliter le déplacement des personnes non-voyantes et leur rendre les taches de la vie quotidiennes plus faciles.

CHAPITRE 01

Généralités sur les systèmes embarqués

I. Généralités sur les systèmes embarqués

I.1.1. Historique des systèmes embarqués

Les premiers systèmes embarqués sont apparus en 1971 avec l'apparition du Intel 4004. L'Intel 4004 développée en 1971, le premier microprocesseur, était le premier circuit intégré incorporant tous les éléments d'un ordinateur dans un seul boîtier : unité de calcul, mémoire, coté des entrées/sorties.

Alors qu'il fallait auparavant plusieurs circuits intégrés différents, chacun dédié à une tâche particulière, un seul microprocesseur pouvait assurer autant de travaux différents que possible.

Très rapidement des objets de la vie quotidienne tels que le four à micro-ondes, télévision, téléphone portable, moyens de transport (automobiles, trains, avions,) ne tardent pas à être équipés de microprocesseur.

Ce sont alors les débuts de l'informatique embarquée.

I.1.2. Qu'est-ce qu'un système embarqué ?

Un système embarqué peut être défini comme un système électronique et informatique autonome, qui est dédié à une tâche bien précise, possédant des ressources d'ordre spatial (taille limitée) et énergétique (consommation restreinte) limitées. Le terme de « Système Embarqué » désigne aussi bien le matériel que le logiciel utilisé. [1]

I.1.3. Caractéristiques des systèmes embarqués

Les systèmes embarqués ont pour but de permettre aux objets de réagir à l'environnement. Ils peuvent aussi apporter une interface avec l'utilisateur. La structure de base de ces systèmes est illustrée sur la figure 1. L'environnement est mesuré par divers capteurs. L'information des capteurs est acheminée pour être traitée par le cœur du système embarqué.

Puis le résultat du traitement est converti en signaux analogique qui génèrent les actions sur l'environnement (afficheur d'informations pour l'utilisateur, actionneur, transmission d'informations, etc.)

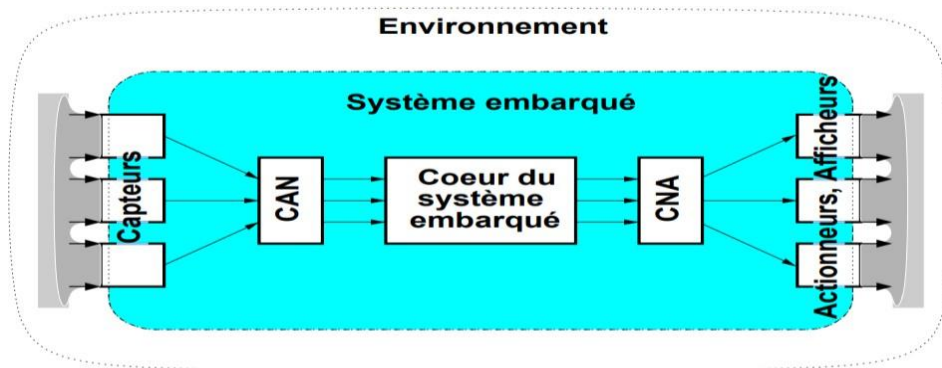


Figure.I.1: Schéma explicatif d'un système embarqué

I.1.4. Les contraintes des systèmes embarqués

Les systèmes embarqués possèdent des contraintes bien particulières par rapport aux autres systèmes électroniques :

- l'encombrement : ils doivent souvent être transportés et doivent être de taille réduite.
- la consommation : étant transportés ils n'ont souvent accès qu'à des ressources limitées en énergie (pile, batteries), il est donc important qu'ils consomment le moins possible.
- le temps : ils sont souvent employés dans des applications où le temps est important.
- la sécurité : ils sont souvent employés dans des domaines tels que l'aéronautique. [2]

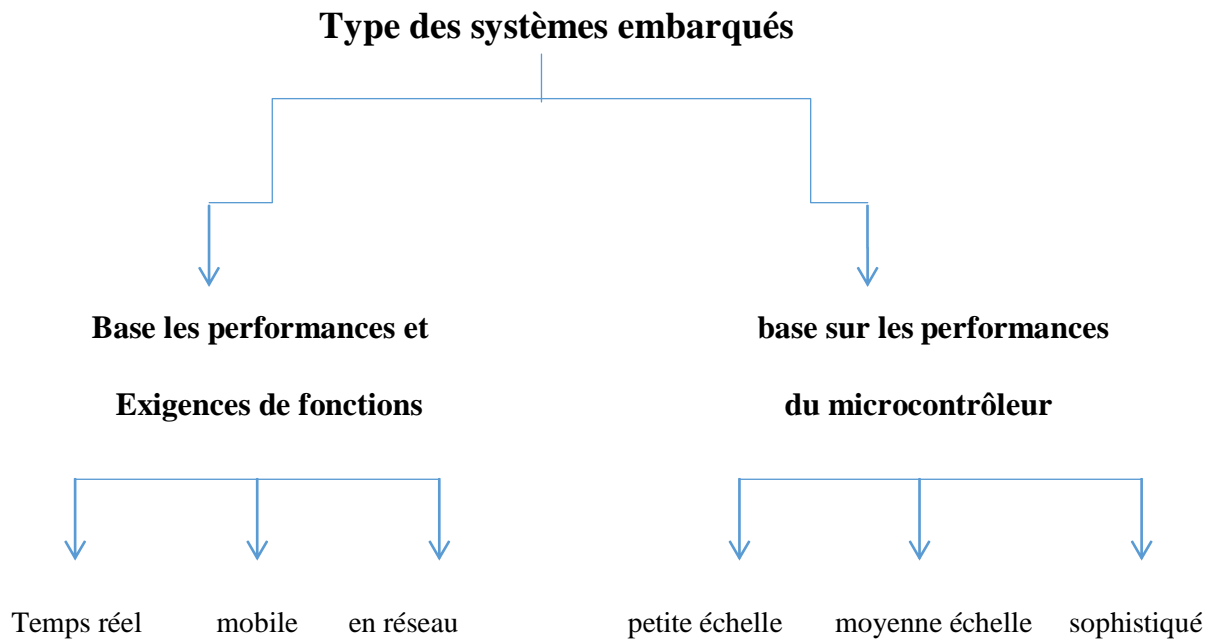
I.1.5. Classification des systèmes embarqués

Les systèmes embarqués sont principalement classés en différents types en fonction de la complexité du matériel et du logiciel et du microcontrôleur (8 ou 16 ou 32 bits). Ainsi, en fonction des performances du microcontrôleur, les systèmes embarqués sont classés en trois types tels que :

- Systèmes embarqués à petite échelle.
- Systèmes embarqués à moyenne échelle.
- Systèmes embarqués sophistiqués.

En outre, en fonction des performances et des exigences fonctionnelles du système embarqué le système est classé en quatre types qui sont :

- Systèmes embarqués en temps réel.
- Systèmes embarqués autonomes.
- Systèmes embarqués en réseau.
- Systèmes embarqués mobiles. [3]



I.1.6. Architecture des systèmes embarqués

L'architecture d'un système embarqué se définit par le schéma suivant :

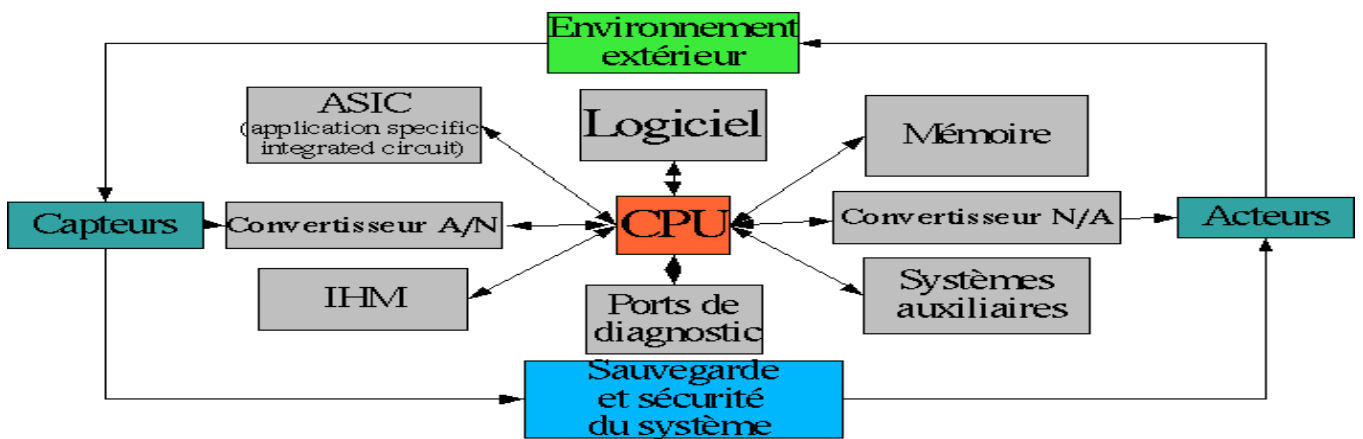


Figure.I.2: Architecture des systèmes embarqués

Cette architecture peut varier selon les systèmes : on peut par exemple, ne pas trouver de systèmes auxiliaires dans de nombreux systèmes embarqués autonomes et indépendants. En revanche, l'architecture de base est la plupart du temps composée d'une unité centrale de

traitement (CPU), d'un système d'exploitation qui réside parfois uniquement en un logiciel spécifique (ex : routeur), ou une boucle d'exécution (ex : ABS). De même l'interface IHM n'est pas souvent existante, mais est souvent utile pour reconfigurer le système ou vérifier son comportement.

Le fonctionnement du système se résume ainsi :

- Il reçoit des informations de l'environnement extérieur qu'il converti en signal numérique.
- L'unité de traitement composée du CPU, de la mémoire, du logiciel, de l'ASIC et éventuellement de système externes traite l'information.
- Le traitement génère éventuellement une sortie qui est envoyée vers la sortie, les systèmes auxiliaires, les ports de monitoring ou l'IHM. [4]

I.1.6.1. L'architecture Harvard et Von Neumann

De nos jours, l'audio et la vidéo sont totalement numérisés. Lorsque le signal audio est hors de toute source audio ou microphone, alors l'analogique sera échantillonné, codé, modulé, compressé et traité dans la forme numérique comme zéro et un, de sorte qu'il est nommé comme DSP (Digital Signal Processing). La refonte du signal numérique en signal audio est effectuée par le traitement du signal numérique. L'architecture du X86 traditionnel s'appelle "Von Neumann", et ne convient pas à la gestion de plusieurs algorithmes pour acheminer ce type de données numériques. L'architecture Harvard la plus populaire est utilisée pour gérer des algorithmes DSP complexes, et cet algorithme est utilisé dans les processeurs RISC les plus courants et les plus avancés. Ici, dans ce mémoire, nous avons discuté de l'architecture de Von Neumann et de l'architecture de Harvard. [5]

I.1.6.2. Architecture Von Neumann

Une architecture Von Neumann n'est rien mais c'est un art qui permet de stocker un ordinateur électronique. Ce n'est pas du tout un nouveau concept il existe depuis longtemps et nous suivons le principe de cette architecture de Von Neumann. Si nous revenons dans l'histoire, il est tout à fait évident que l'architecture de Von Neumann a été publiée pour la première fois dans le rapport de John von Neumann le 30 juin 1945.

L'architecture dite architecture de Von Neumann, est un modèle pour microcontrôleur qui utilise une structure de stockage unique pour conserver à la fois les instructions et les données requises ou générées par le calcul. La séparation entre le stockage et le processeur est

implicite dans ce modèle. Une même instruction permet l'accès aux programmes ou aux données, cependant pas simultanément. Cette architecture est principalement utilisée pour la conception des processeurs d'ordinateurs (PC, MAC).

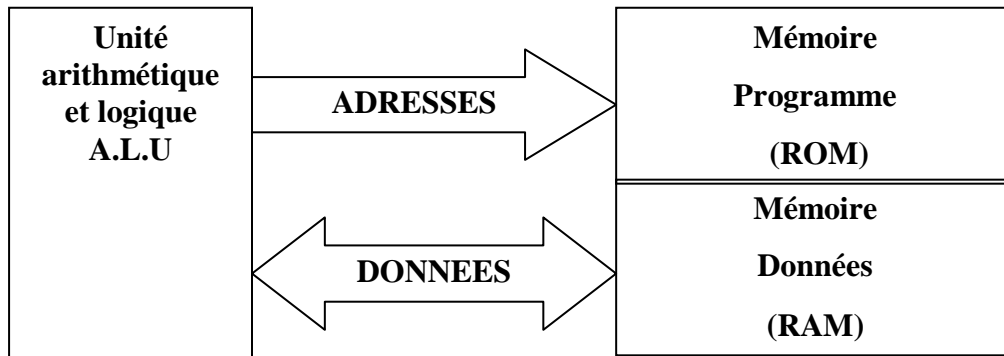


Figure.I.3: Architecture de Von Neumann

I.1.6.3. Applications et caractéristiques de l'architecture Von Neumann

L'architecture de Von Neuman a de nombreuses applications dans la vie quotidienne. En gardant à l'esprit la vaste application, l'architecture Von Neuman a été introduite comme un sujet dans le degré d'engendrement. L'architecture de Von Neuman comporte quelques caractéristiques importantes et nous aurons ici une élaboration détaillée des caractéristiques.

➤ **Mémoire:**

Nous savons tous que le Von Neuman n'est rien d'autre qu'un ordinateur ayant la fonction de stockage de données. Dans l'architecture du Von Neuman, la mémoire joue un rôle essentiel et est considérée comme l'une des caractéristiques importantes. Principalement, cela est responsable de la tenue et du démarrage des données et des données de programmation. Dans les temps modernes, cela a été remplacé par la RAM et maintenant nous utilisons la RAM à cette fin.

➤ **Unité de contrôle:**

Cette unité est principalement responsable de l'aspect de contrôle. Toutes les données stockées dans la mémoire et pendant le traitement des données, l'unité de contrôle joue le rôle et gère le flux de données. En fait, pour être plus typiquement c'est "One At A Time". L'unité de contrôle suit le principe du One at A Time et traite en conséquence toutes les données.

➤ **Entrée-sortie:**

Comme tous les appareils électroniques, l'architecture Von Neuman dispose également d'une architecture d'entrée / sortie. C'est la fonction de base est la même et rien n'a été spécialement conçu pour l'architecture d'entrée et de sortie. Avec le périphérique d'entrée et de sortie sur une personne peut communiquer avec l'appareil.

□ **ALU:**

ALU ou Arithmetic Logic Unit a une grande importance dans l'architecture de Von Neuman. Toute sorte d'addition, soustraction, multiplication et division des données sera effectuée par cette ALU. En plus de cela, tout autre type de fonction algorithmique et d'activités sera effectué par l'ALU. Ce sont l'aspect fondamental de l'architecture de Von Neuman dont vous devez être conscient

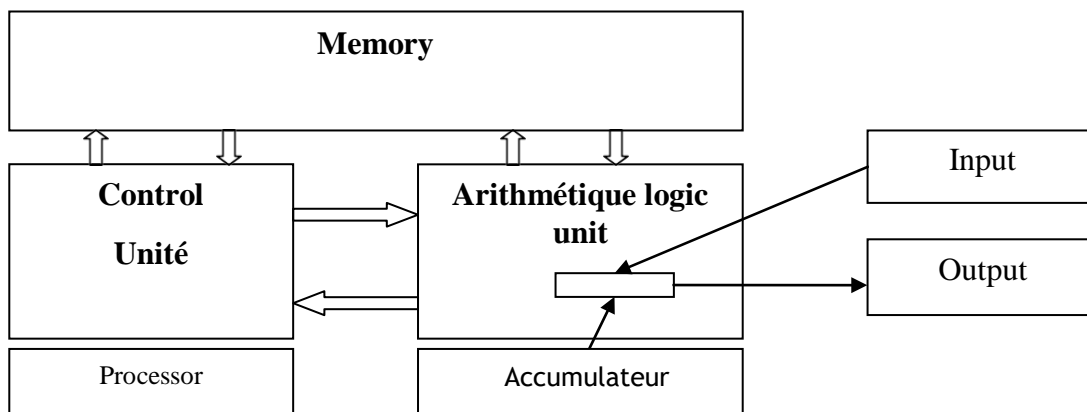


Figure .I.4: Caractéristiques de l'architecture Von Neumann

I.1.6.4. Architecture Harvard

L'architecture de Harvard n'est rien d'autre qu'une sorte de stockage de données. En ce qui concerne le stockage physique des données, l'architecture de Harvard a toujours été la première. Bien que le concept ne soit pas nouveau, l'architecture de Harvard a toujours été

très appréciée. L'ordinateur basé sur le relais Harvard Mark I est le terme où le concept de l'architecture de Harvard se pose d'abord et à un développement significatif avec cette architecture. La fonction principale de cette architecture est de stocker séparément et physiquement les données et de donner les voies de signal pour l'instruction et les données.

Cette structure permet un accès simultané aux données et aux instructions l'exécution des programmes est plus rapide. En revanche elle a des instructions différentes pour accéder à la mémoire et à la mémoire de données.

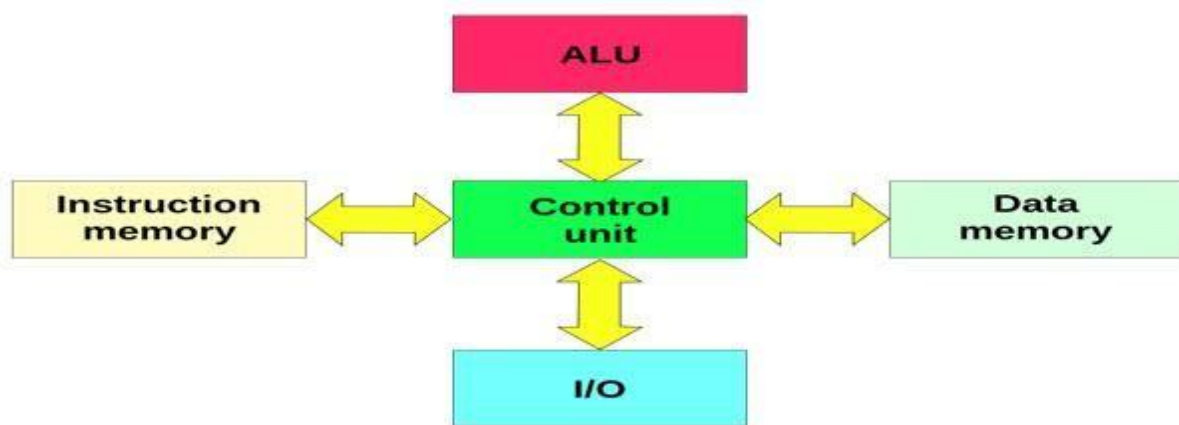


Figure .I.5: Caractéristiques de l'architecture Von Neumann

I.1.6.5. Applications et caractéristiques de l'architecture de Harvard

Il a une application étendue dans les produits de traitement audio et vidéo et avec chaque instrument de traitement audio et vidéo, vous remarquerez la présence de l'architecture Harvard. Les processeurs de Blackfin par Analog Devices, Inc. est l'appareil particulier où il a eu un premier usage. Dans d'autres produits à base de puces électroniques, l'architecture Harvard est également largement utilisée

➤ État de la mémoire :

Dans l'architecture précédente, nous avons assisté à la présence de deux mémoires. Cependant, dans le cas de l'architecture de Harvard, vous ne trouverez pas deux mémoires et il y aura une mémoire. La mémoire existante sera capable d'effectuer toutes les fonctions. Oui, il n'y aura que de la mémoire en lecture seule et cette mémoire sera utilisée pour la lecture, le codage, le décodage et le stockage des données.

➤ L'aspect de la vitesse :

On a beaucoup parlé de l'architecture de Harvard, mais sans la vitesse, aucune architecture ne peut être acceptée. Mais dans le cas de l'architecture Harvard, les constructeurs ont conçu l'architecture de manière à pouvoir traiter les données avec une vitesse beaucoup plus élevée. Oui, tout a été fait pour que l'architecture puisse traiter les données à haute vitesse.

En mettant en œuvre la même formule, les CPU modernes sont fabriquées de sorte que le nouveau CPU peut fonctionner avec une accélération beaucoup plus rapide et peut également traiter les données de manière efficace. Le concept du cache CPU est également implémenté lors de la conception de l'architecture Harvard.

I.1.6.6. Pourquoi l'architecture de Harvard est-elle efficace?

La réponse est assez claire et simple parce que l'architecture est capable de lire une instruction et qu'elle peut également effectuer un accès à la mémoire de données simultanément à une vitesse rapide. D'où l'architecture de Harvard est largement acceptée.

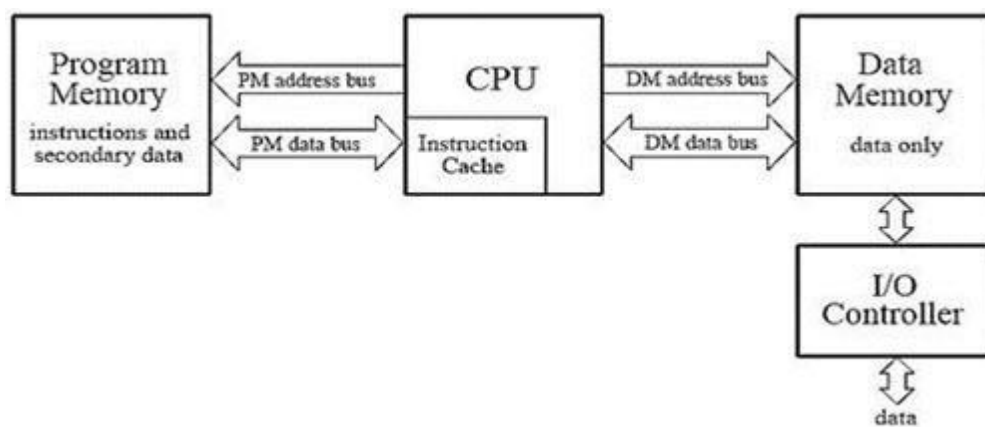


Figure .I.6: Caractéristiques de l'architecture de Harvard

I.1.6.7. Différence entre L'architecture Von Neumann et Harvard

Les caractéristiques et les spécifications des deux architectures sont discutées et la fonction de base des deux architectures est de concevoir l'architecture de l'ordinateur. Cependant, dans

certaines aspects, les deux architectures diffèrent les unes des autres et ici, nous allons élaborer la différence fondamentale entre l'architecture Von Neumann et Harvard.

Chaque fois que vous regardez une vidéo, elle est synchronisée entre l'audio et la vidéo. Si le microphone ou la puce audio est hors de contrôle, vous ne pouvez pas profiter de la vidéo et vice versa, et le même concept s'applique ici aussi.

Von Neumann est une sorte d'architecture qui est plus visible dans le cas du processeur conventionnel et aujourd'hui, le même principe est mis en œuvre. Les ordinateurs et les serveurs, ainsi que les systèmes embarqués font partie des processeurs conventionnels et traditionnels qui fonctionnent uniquement avec la fonction de contrôle et vous remarquerez ici l'architecture de Van Neumann.

Au contraire, l'architecture Harvard est vue dans le cas des processeurs modernes et récents comme les DSP et autres processeurs. En plus des systèmes de communication mobiles ci-dessus, les systèmes de traitement de l'audio, de la parole et de l'image sont l'endroit où vous trouverez l'application de l'architecture Harvard.

Dans l'architecture Von Neumann, les données et les programmes sont ignorés dans la même mémoire, mais dans le dernier cas, la mémoire séparée est utilisée dans le but ci-dessus. Par conséquent, dans le stockage, l'aspect des données diffère l'un de l'autre.

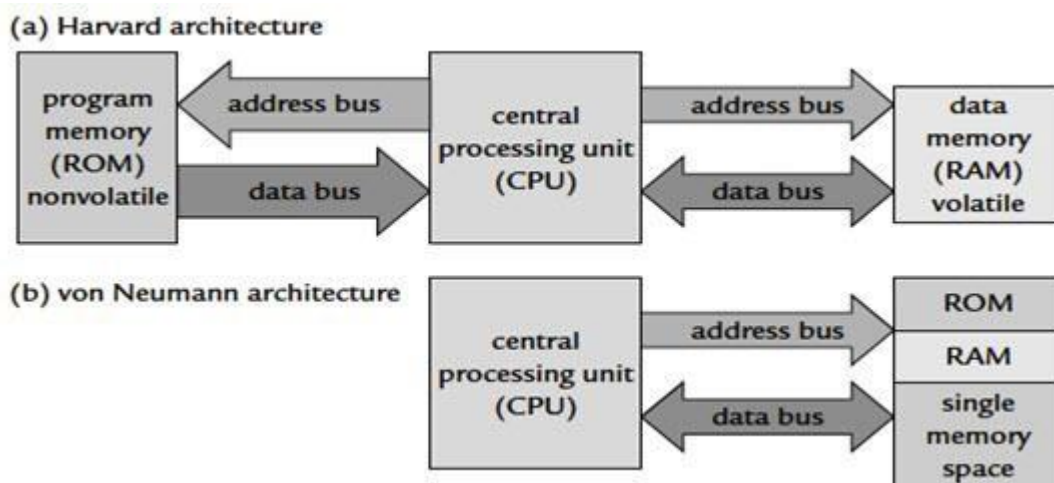


Figure.I.7: Architecture Von Neumann et Harvard Architectures

Le codage diffère également dans l'architecture. Le Von Neumann est une sorte d'architecture où vous trouverez que les codes sont exécutés en série et prend plus de cycles d'horloge, alors que dans le cas de la dernière, il n'en est pas de même. L'architecture de Harvard suit un niveau de codage parallèle.

Vous ne trouverez pas de nombre excessif de multiplicateurs dans le cas de l'architecture de Von Neumann, mais dans le cas de l'architecture de Harvard, vous trouverez un grand nombre de multiplicateurs. En plus de cela, les deux architectures diffèrent les unes des autres dans divers aspects comme la présence de shifter de baril, l'optimisation de la programmation, la vitesse de traitement et beaucoup plus.

I.2. Programmation

Un système embarqué doté d'un processeur a besoin de code pour faire quelque chose. Ce dernier est placé dans une mémoire accessible par le processeur afin de pouvoir être exécuté, instruction par instruction. Typiquement, le software d'un système embarqué sera présenté sous la forme d'une image ROM, définitive et non modifiable.

Étant donné un système embarqué, on connaît la taille de la ROM qu'il possède et donc les adresses de tous les emplacements mémoire disponibles. Pour chacun de ces emplacements, il faut en définir le contenu binaire représentant des instructions à exécuter. Ces instructions sont écrites en respectant le jeu d'instructions du processeur utilisé, ce qui peut évidemment prendre beaucoup de temps à comprendre avant de commencer à programmer.

Pour éviter de mémoriser les codes machine de toutes les instructions d'un processeur donné, on peut programmer à l'aide d'un langage d'assemblage, représentation textuelle de plus haut niveau du code machine. Un programme écrit dans un langage d'assemblage est traduit en code machine par l'assembleur. Programmer à un tel niveau reste une tâche fortement consommatrice de temps, mais peut être plus efficace si l'on s'aide de bibliothèques de code préexistant.

Enfin, pour gagner du temps de développement, il est possible d'utiliser un langage de haut niveau qui offre des constructions et abstractions plus pratiques et proches de la façon de penser du développeur et du problème qu'il veut résoudre. Un exemple d'un tel langage, adapté aux systèmes embarqués, et qui reste d'assez bas niveau et proche du hardware est le langage C. Un programme écrit dans un langage de haut niveau est compilé en code machine par le compilateur.

La figure résume les trois principales possibilités de programmation d'un système embarqué : directement en langage machine, via un langage d'assemblage ou à l'aide d'un langage de haut niveau. La phase de traduction d'un langage d'assemblage est quasi immédiate et unique tan-

dis que la phase de compilation d'un langage de haut niveau peut être faite de plusieurs manières et donc optimisée. [6]

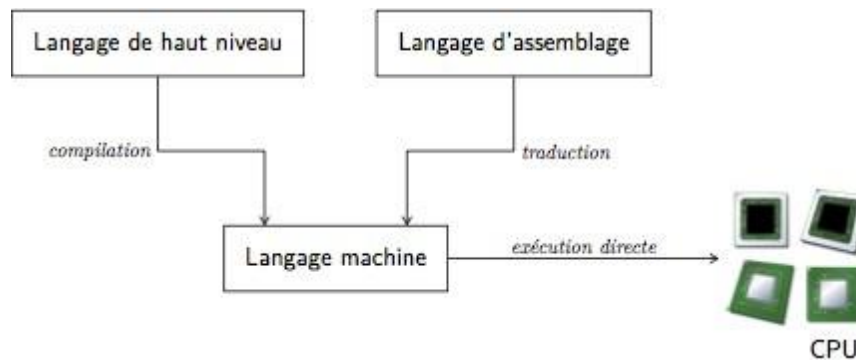


Figure .I.8: Principale possibilité de programmation

I.3. Développement

Développer un logiciel pour un système embarqué ne se fait pas complètement de la même manière que pour un logiciel destiné à un ordinateur « traditionnel ». En effet, il ne sera pas toujours possible de directement développer sur le système embarqué, mais il faudra le faire sur un système hôte. La figure résume le processus de développement type. Une fois que le développeur a compilé un code machine sur le système hôte, il l'envoie sur le système cible ou il l'exécute sur un émulateur ou simulateur sur le système hôte. Dans le cas où le code est envoyé sur le système cible, il est possible de le débiter en run time, par exemple avec le standard JTAG présenté plus loin dans ce chapitre.

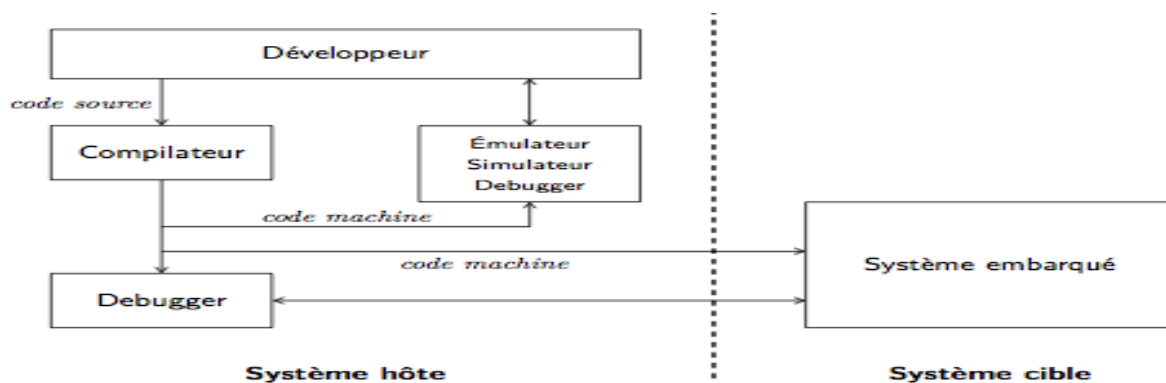


Figure .I.09 : processus de développement

Le développement est typiquement réalisé sur une plateforme hôte avant envoi du code et exécution sur la plateforme cible. Le debugging peut se faire à différents niveaux, en émulation sur l'hôte ou directement sur la cible. Concernant la procédure de développement, il n'y en a évidemment pas une seule unique à suivre. Il est cependant important de coordonner le développement des aspects hardware avec celui des aspects software. La figure 09 reprend les grandes étapes de développement.

La première phase consiste évidemment à établir les spécifications précises du système à développer. Il faut notamment caractériser l'environnement d'exécution et identifier les contraintes qu'il impose. De plus, il faudra faire des choix par rapport aux niveaux de fiabilité et d'efficacité désirés.

À partir de ces spécifications, il faudra déterminer ce qui sera fait en hardware et en software, et décider des composants à utiliser. Plusieurs itérations peuvent être nécessaires à cette seconde étape, qui est par ailleurs très importante car c'est un point de non retour.

Vient ensuite le développement, en parallèle, des parties software et hardware qui seront ensuite intégrées, puis testées et validées. Une fois le système développé et déployé, on entre dans une phase importante de maintenance. Le système doit être monitoré, des bugs éventuels doivent être corrigés et des mises à jour peuvent être appliquées.

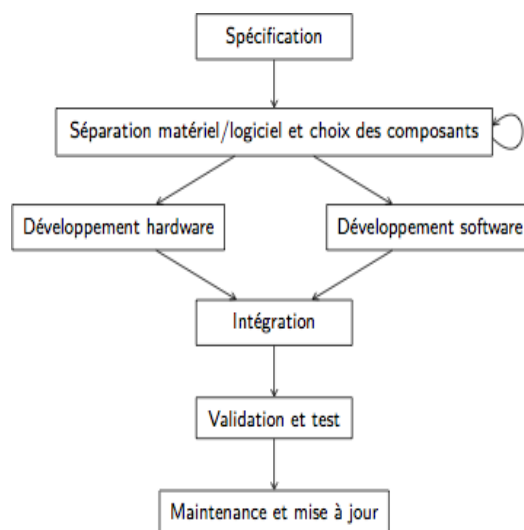


Figure .I.10 : organigramme de développement

I.4. Debug

Débugger le software d'un système embarqué est une tâche plus compliquée que pour un logiciel développé sur un ordinateur « traditionnel ». Une possibilité consiste à utiliser les standards JTAG (le Joint Test Action Group(JTAG) est une association d'industries d'électroniques qui développe des méthodes pour vérifier et tester des circuits imprimés) qui offrent notamment un protocole basé sur une communication série pour accéder au processeur et bus système.

I.5. Système d'exploitation

Terminons ce tour des aspects software en introduisant la notion de *système d'exploitation*. Il s'agit d'un programme qui est exécuté sur le hardware et qui joue le rôle d'intermédiaire entre le hardware et les applications utilisateur. Sur un ordinateur « traditionnel », les systèmes d'exploitation les plus connus sont Windows, MacOS et Linux.

Un système d'exploitation offre des abstractions des composants hardware pour faciliter la programmation. Par exemple, on peut manipuler des fichiers sur le disque dur, des fenêtres sur l'écran, des processus sur le processeur, etc. Le développeur ne doit ainsi plus se soucier de savoir contrôler le hardware à très bas niveau, il exploite les services offerts par le système d'exploitation.

I.6. Les langages de programmation des systèmes embarqués

Pour la programmation des systèmes embarquée, il faut noter que plusieurs langages de programmation se veulent dédiés à l'embarqué. Parmi ces langages se trouvent « Ada » et le langage Assembleur, ce dernier restant encore un choix approprié pour les systèmes soumis à des contraintes sévères de temps réel.

Des langages proches de la machine comme le langage C et dans une moindre mesure le C++ sont aussi utilisés

Classement	Langage	Catégorie
1	C	Mobile Entreprise Système embarqué
2	C++	Mobile Entreprise Système embarqué
3	Arduino	Système embarqué
4	Assembly	Système embarqué
5	Haskell	Entreprise Système embarqué
6	D	Web Système embarqué
7	LabView	Entreprise Système embarqué
8	VHDL	Système embarqué
9	Ladder Logic	Système embarqué
10	Erlang	Entreprise Système embarqué
11	Verilog	Système embarqué
12	Ada	Entreprise Système embarqué
13	TCL	Entreprise Système embarqué
14	Forth	Système embarqué

Tableau 1 : Les langages les plus utilisés pour le développement de systèmes embarqués

Les langages C et C++ confortent également leur place respective dans le dernier classement de l'IEEE des meilleurs langages pour les systèmes embarqués. On voit aussi d'autres langages tels qu'Arduino, Haskell, D, LabVIEW et VHDL qui sont bien classés.

En dehors de certains langages populaires (à usage général) qui reviennent dans les deux classements, il peut être important de donner quelques précisions sur les différents langages :

- **Arduino** : il s'agit du langage natif pour le microcontrôleur Arduino, qui est devenu la base d'un grand nombre de dispositifs de fabrication et de prototypage.

- **LabView** : créé par National Instruments, LabView est conçu pour l'acquisition de données et le contrôle industriel.
- **VHDL** : VHSIC Hardware Description Language (VHDL) est un langage de description matériel utilisé dans la création et l'analyse de circuits électroniques ;
- **Ladder Logic** : il s'agit d'un langage de programmation destiné au développement de contrôleurs logiques programmables industriels.
- **Erlang** : créé par Ericsson pour les applications de téléphonie embarquées, la publication d'Erlang en tant que langage open source en 1998 a renforcé sa popularité parmi les programmeurs qui développent des applications qui doivent gérer de nombreuses tâches simultanées.
- **Verilog** : comme VHDL, Verilog (ou Verilog HDL) est un langage de description matériel utilisé dans la création et l'analyse de circuits électroniques.
- **Ada** : à l'origine conçu pour le département de défense des États-Unis, Ada est utilisé pour des applications où la fiabilité est critique, comme les systèmes de contrôle aérospatial.
- **TCL** : il s'agit d'un langage de script destiné au prototypage rapide et supportant l'interface utilisateur graphique Tk utilisée principalement avec les systèmes Unix.
- **Forth** : conçu à l'origine pour contrôler les radiotélescopes, Forth est toujours utilisé pour des applications telles que les boots loaders et d'autres firmwares.
- **Scade** : il s'agit d'un langage pour l'embarqué critique. C'est le langage de modélisation de SCADE Suite, un environnement de développement intégré pour la conception de systèmes critiques.

I.7. Domaines d'application des systèmes embarqués:

Les domaines dans lesquels on trouve des systèmes embarqués sont de plus en plus nombreux :

- Astronautique : fusée, satellite artificiel, sonde spatiale etc.
- Automate programmable industriel, contrôle-commande.
- Electroménager : télévision, four à micro-ondes.
- Environnement.

- Équipement médical.
- Guichet automatique bancaire (GAB).
- Impression : imprimante multifonction, photocopieur, etc.
- Informatique : disque dur, lecteur de disquette, etc.
- Métrologie.
- Militaire : missile.
- Multimédia : console de jeu, assistant personnel.
- Télécommunication : set-top box, téléphonie, routeur, pare-feu, téléphone portable.
- Transport : automobile, aéronautique, ferroviaire. [7]

I.8. Les systèmes embarqués temps réel

I.8.1. Définition

La plupart des systèmes embarqués temps réel sont avant tout des systèmes de contrôle-commande. Un système de contrôle-commande est un système informatique de contrôle de procédé. Le terme procédé est un terme générique désignant un système physique contrôlé. Afin de contrôler le procédé, le système informatique est en relation avec l'environnement physique externe par l'intermédiaire de capteurs et/ou d'actionneurs. Les grandeurs physiques acquises grâce aux capteurs permettent au système de contrôle-commande de s'informer de l'état du procédé ou de son environnement. Le système de contrôle-commande, à partir de consignes décrivant l'état voulu du procédé, calcule des commandes qui sont alors appliquées sur le procédé par l'intermédiaire d'actionneurs. [8]

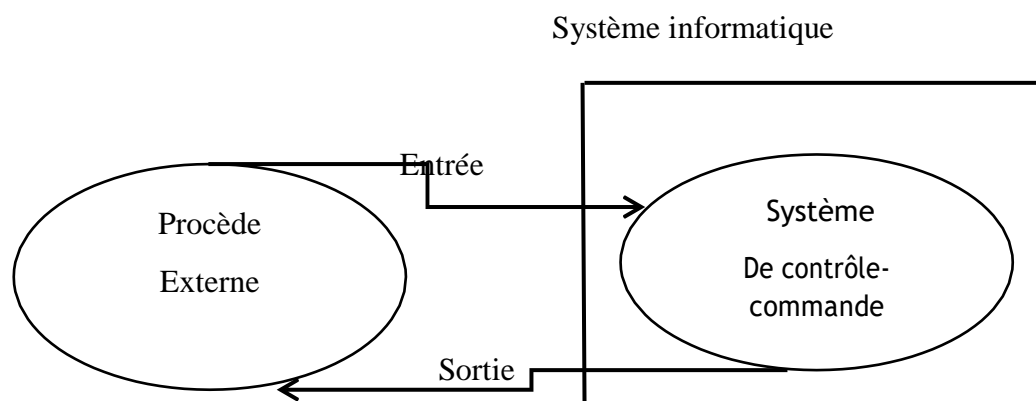


Figure .I.11: Représentation schématique d'un système de contrôle commande

I.8.2. Principales caractéristiques des systèmes temps réel

Les principales caractéristiques des systèmes de contrôle-commande sont

- **Grande diversité des dispositifs d'entrées/sorties** : les données à acquérir qui sont fournies par les capteurs et les données à fournir aux actionneurs sont de types très variés (continu, discret, tout ou rien ou analogique). Il est aussi nécessaire de piloter un bus de terrain pour les communications.
- **Prise en compte des comportements concurrents** : l'ensemble de ces données physiques qui arrivent de l'extérieur et le réseau qui permet de recevoir des messages ne sont pas synchronisés au niveau de leurs évolutions, par conséquent, le système informatique doit être capable d'accepter ces variations simultanées des paramètres.
- **Respect des contraintes temporelles** : la caractéristique précédente impose de la part du système informatique d'avoir une réactivité suffisante pour prendre en compte tous ces comportements concurrents et en réponse à ceux-ci, de faire une commande en respectant un délai compatible avec la dynamique du système.
- **Sûreté de fonctionnement** : les systèmes de type contrôle-commande mettent souvent en jeu des applications qui demandent un niveau important de sécurité pour raisons de coût ou de vies humaines. Pour répondre à cette demande, il est nécessaire de mettre en œuvre toutes les réponses de la sûreté de fonctionnement (développements sûrs, tests, méthodes formelles, prévisibilité, déterminisme, continuité de service, tolérance aux fautes, redondance, ségrégation des moyens de calcul et de communication, etc.).

I.8.3. Caractéristiques temporelles des systèmes temps réel

Les contraintes temporelles qui sont classiquement présentées sont des contraintes de bout en bout, appelées aussi contraintes de latence. Ces contraintes représentent le délai maximal entre lecture de l'état du système (lecture des capteurs) et commande de celui-ci (commande des actionneurs).

Il est nécessaire de préciser et de formaliser les caractéristiques temporelles d'un système. Cette caractérisation peut prendre de nombreuses formulations. Ainsi, nous pouvons définir de manière non exhaustive :

- Durée d'exécution d'une activité.

- Cadence de répétition ou périodicité d'une activité.
- Date au plus tôt ou date de réveil.
- Date de démarrage .
- Date de fin.
- Date au plus tard ou échéance.
- Temps de réponse.
- Gigue temporelle.

Aujourd'hui encore, et malgré toutes les avancées en matière de détection d'obstacles et de cartographie en 3D, les personnes non-voyantes s'en remettent majoritairement à la canne blanche ou au chien-guide pour pouvoir se déplacer. Il existe bien des dispositifs électroniques qui se servent de rayons infrarouges ou d'ultrasons pour détecter et évaluer la distance des obstacles.

I.9. La détection d'obstacle

I.9.1. Définition

Afin de définir la détection d'obstacle il faut tout d'abord comprendre ce que c'est un obstacle. Le dictionnaire de l'académie française donne la définition suivante :

<< Ce qui arrête ou ralentit le passage, ce qui empêche d'avancer >>

Au sens large, la détection d'obstacles est une tâche perceptive, dont le but est la détection et la caractérisation des objets présents dans une scène, pour déterminer les trajectoires possibles. [9]

I.10. Capteur et méthode pour la détection d'obstacles

I.10.1. Les télémètres

La première catégorie de capteurs utilisés pour la détection d'obstacles regroupe les télémètres. Ceux-ci sont des capteurs actifs émettant une onde qui se réfléchit à la surface des objets situés dans la direction de propagation. La comparaison entre l'onde incidente avec son écho permet d'estimer un temps de vol, un déphasage ou un éventuel glissement de fréquence, qui fournit ensuite la distance séparant le capteur de la surface formant l'écho. [9]

I.10.2. Le télémètre à ultrasons

Il est aisé et très peu coûteux de construire des télémètres exploitant des ondes acoustiques. La fréquence de celles-ci ne pouvant pas être située dans le spectre audible, ce sont des ultrasons qui sont utilisés. Ceux-ci présentent également les avantages de pouvoir être émis par une source de petite taille et d'être réfléchis par des objets à taille humaine.

La portée de ces capteurs est très faible (2 à 4 m) et leur champ d'application est donc restreint.



Figure .I.12:Télémètre à ultrason

I.10.3. Le radar

Le principe est l'émission d'une onde électromagnétique et la réception de l'écho formé par cette onde sur une surface conductrice pour estimer la distance à cette surface.

Une portée et une précision importantes, couplées à un fonctionnement assuré dans la plupart des conditions météo (brouillard, pluie et neige) sont les avantages indéniables du radar.

Cependant, le radar présente un certain nombre de limitations. Tout d'abord, l'angle de vue du capteur est faible (environ 10 degrés pour un radar longue portée) et ne permet donc pas de percevoir des objets situés autour de ce cône d'ouverture.



Figure .I.13: Le radar

I.10.4. Le lidar

Fondé sur l'utilisation d'un laser, le lidar (ou télémètre laser) exploite la réflexion d'un rayon lumineux pour mesurer une distance. Il exploite généralement le temps de vol de l'onde pour estimer une distance. Il est possible d'utiliser par ailleurs une mesure de réflectance pour caractériser les surfaces de réflexion et ainsi distinguer des objets dont la surface est différente.

Le lidar fournit une mesure très précise de distance sur une grande plage de portée. Il permet de plus de détecter tout type d'objet (voiture, piéton, carton . . .) pourvu que ceux-ci renvoient une partie de la lumière incidente.



Figure .I.14: Le lidar

I.10.5. La vision monoculaire

Parallèlement à l'utilisation des capteurs actifs que nous venons de présenter, grand nombre d'études portent sur la vision artificielle. L'utilisation d'une caméra permet, pour un coût très faible, d'accéder à une grande quantité d'informations sur un large champ de vue. Cependant, comme une caméra ne fournit pas immédiatement de mesure de distance, contrairement aux télémètres, elle doit faire appel à des algorithmes de traitement d'images plus ou moins complexes pour réaliser la détection.

Nous allons présenter rapidement les algorithmes les plus couramment utilisés pour la détection d'obstacles.

I.11. Les primitives à rechercher

Peuvent être de types très variés et nous allons en présenter un aperçu non exhaustif.

- Les contours image des objets sont souvent utilisés, car faciles à détecter et très présents sur le type d'objets recherchés.
- La texture et les ombres sont également des primitives exploitées pour la détection.
- Lorsque la caméra utilisée permet d'acquérir des images en couleur, d'autres primitives sont envisageables. Par exemple, la recherche de zones de forte intensité
- Les longueurs d'ondes au-delà du visible peuvent également se révéler pertinentes.
- L'utilisation d'une caméra sensible au rayonnement infrarouge lointain est une solution bien adaptée à la détection des obstacles.

I.12. Des algorithmes de reconnaissance de forme

Sont ensuite appliqués dans les régions où ont été détectées ces primitives, pour reconnaître si elles appartiennent effectivement à un type d'obstacle connu.

I.13. L'exploitation d'images successives

Fournies par la caméra embarquée est également une possibilité, car elle permet de calculer un flot optique. Tous les éléments immobiles de la scène sont alors, du point de vue du système de détection, animés d'un même mouvement. L'estimation de ce mouvement dominant permet donc de trouver les objets immobiles qui, une fois supprimés, laissent apparaître des pixels appartenant théoriquement aux objets mobiles. Ceux-ci peuvent alors être suivis pour assurer leur détection

I.14. La combinaison de plusieurs approches

Permet enfin d'assurer une plus grande fiabilité à l'algorithme de détection en permettant de bénéficier d'informations complémentaires. Par exemple le système FADE, qui exploite la fusion des résultats d'algorithmes fondés sur la détection d'ombres, la détection des feux arrière, la détection de contours et la détection de symétrie.

I.15. La stéréovision

La stéréovision est une technique de vision fondée sur l'utilisation de plusieurs caméras (généralement deux, trois pour le système de Carnegie Mellon). En combinant leur différent point de vue, cela permet d'accéder à des informations de profondeur difficilement accessibles lorsqu'une unique vue est utilisée. L'exploitation de cette information permet de bâtir des techniques de détection d'obstacles génériques et robustes.



Figure.I.15: Technique de stéréovisions

I.16. Les systèmes coopératifs

Une voie de recherche actuellement jugée prometteuse pour la détection d'obstacles et les aides à la conduite en général concerne les systèmes coopératifs. Cette appellation générique désigne des systèmes permettant aux différents acteurs d'une scène de "s'entraider" afin de fournir un gain en performance global. Ceux-ci peuvent donc être des dispositifs destinés à ce que l'obstacle à détecter aide à sa propre détection et éventuellement l'enrichisse d'informations supplémentaires. Des stratégies très variées existent en ce sens.

I.17. Conclusion

Nous venons de présenter les capteurs généralement utilisés pour la détection d'obstacles, ainsi que les méthodes de traitement qui y sont associées.

Il semble donc nécessaire de bâtir une architecture de détection qui couvre une plage de fonctionnement maximale, avec un nombre minimum de capteurs, tout en assurant une très grande robustesse de fonctionnement.

Chapitre 02

ELEMENT DE CONCEPTION

D'UNE CANNE

INTELLIGENTE

II. Introduction

Un système embarqué est un système complexe qui intègre du logiciel et du matériel conçus ensemble afin de fournir des fonctionnalités données. Il contient généralement un ou plusieurs microprocesseurs destinés à exécuter un ensemble de programmes définis lors de la conception et stockés dans des mémoires. Le système matériel et l'application (logiciel) sont intimement liés et immergés dans le matériel et ne sont pas aussi facilement discernables comme dans un environnement de travail classique de type ordinateur de bureau PC.

Dans ce chapitre nous allons voir et décrire les éléments de conception de notre système

II.1. Arduino UNO

II.1.1. Définition

Arduino UNO est une carte microcontrôleur basée sur l'ATmega328P (datasheet). Il possède 14 broches d'entrée / sortie numériques (dont 6 peuvent être utilisées comme sorties PWM) elle est dotée de :

- 14 entrées/sorties (dont 6 fournissent la sortie PWM)
- 6 entrées analogiques
- Un quartz à 16 MHz
- Une connexion USB
- Une prise d'alimentation
- Un connecteur ICSP
- Un bouton de réinitialisation.

Elle contient tout le nécessaire pour supporter le microcontrôleur; connectez-le simplement à un ordinateur avec un câble USB ou alimentez-le avec un adaptateur AC-DC ou une batterie pour commencer.

"Uno" signifie un en italien et a été choisi pour marquer la sortie d'Arduino Software (IDE) 1.0. Le conseil d'uno et la version 1.0 d'Arduino Software (IDE) étaient les versions de référence d'Arduino, maintenant évolué vers des versions plus récentes. La carte UNO est la

première d'une série de cartes Arduino USB et le modèle de référence de la plate-forme Arduino. [10]

II.1.2. Schéma des ports

Cette figure définit le schéma des ports de la carte arduino

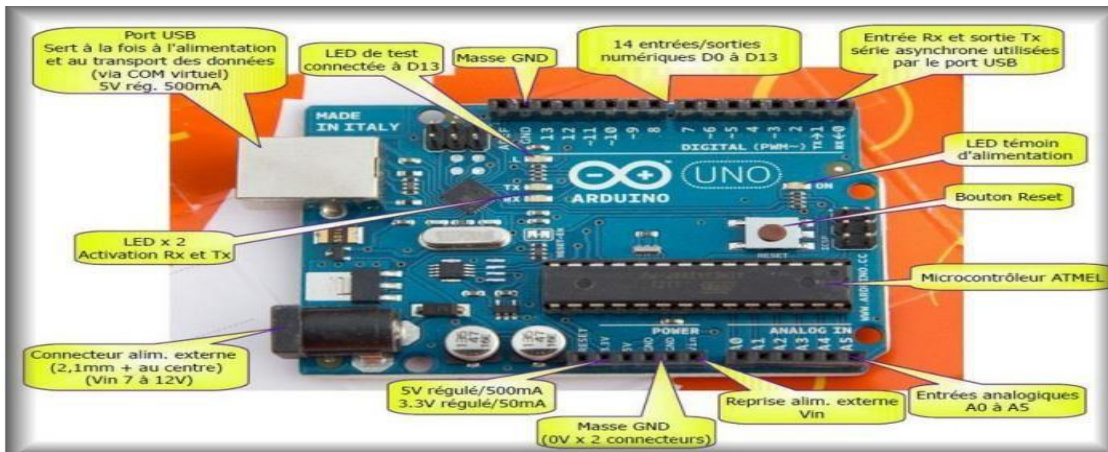


Figure .II.01 : schéma de port de la carte arduino

II.1.3. Caractéristique technique

Microcontrôleur	ATmega328P
Tension de fonctionnement	5V
Tension d'entrée (recommandé)	7-12V
Tension d'entrée (recommandé)	6-20V
Tension d'entrée (recommandé)	Tension d'entrée (recommandé)
PWM numérique E / S Pins	6
Pins d'entrée analogique	6
DC Courant par I O Pin /	20mA
Courant DC pour 3.3V Pin	50mA
Mémoire flash	32 KB (ATmega328P) dont 0,5 KB utilisé par bootloader

SRAM	2 KB (ATmega328P)
EEPROM ***	1 KB (ATmega328P)
Vitesse de l'horloge	16 MHz
Longueur	68.6 mm
Largeur	53.4 mm
Poids	25g

Tableau 2 : caractéristique de la carte arduino UNO

II.1.4. Détails techniques

La carte Arduino Uno peut être alimentée via la connexion USB ou avec une alimentation externe. La source d'alimentation est automatiquement sélectionnée. Une alimentation externe peut provenir soit d'un adaptateur AC-DC ou d'une batterie. L'adaptateur peut être connecté en branchant une prise 2.1mm dans la prise d'alimentation de la carte ou à partir d'une batterie connectée dans le pin (ou broche) GND et V-in (alimentation externe). Le processeur peut fonctionner sur une alimentation externe de 6 à 20 volts. Cependant, si la tension est inférieure à 7V, le pin 5V peut fournir moins de cinq volts et le processeur peut devenir instable. Si la tension est supérieure à 12V, le régulateur de tension peut surchauffer et endommager la carte. La plage recommandée est de 7 à 12 volts. Les pins (ou broches) d'alimentation sont les suivantes:

- V-in. Tension d'entrée à la carte Arduino à l'aide d'une source d'alimentation externe (par opposition à 5 volts de la connexion USB ou une autre source d'alimentation régulée). Si l'alimentation en tension est faite par l'intermédiaire de la prise d'alimentation, on pourra y accéder via ce pin.
- 5V. Cette pin délivre un 5V régulé par la carte. Le processeur peut être alimenté soit à partir de la prise d'alimentation DC (7-12V), le connecteur USB (5V), ou le pin V-in de la carte (7-12). La fourniture d'une tension via les 5V ou 3,3V contourne le régulateur, et peut endommager votre processeur. A déconseiller !
- 3V 3. Une alimentation de 3,3 volts générée par le régulateur. La consommation de courant maximale est de 50 mA.
- GND. masse

- IOREF. Ce pin sur la carte Arduino fournit la référence de tension avec laquelle le microcontrôleur fonctionne.

II.1.4.1. Mémoire

L'ATmega328 a 32 Ko (avec 0,5 KB occupées par le bootloader). Il a également 2 Ko de SRAM et 1 Ko de mémoire EEPROM (qui peut être lu et écrit avec la bibliothèque de l'EEPROM).

II.1.4.2. Entrées et sorties :

Chacune des 14 broches numériques sur la carte Uno peut être utilisée comme une entrée ou une sortie, en utilisant les fonctions `pinMode`, `digitalWrite` et `digitalRead`. Ils fonctionnent à 5 volts. Chaque broche peut fournir ou recevoir 20 mA en état de fonctionnement recommandée et a une résistance de pull-up interne (déconnecté par défaut) de 20-50k ohm. Un maximum de 40mA est la valeur qui ne doit pas être dépassée sur toutes les broches d'Entrée/Sorties pour éviter des dommages permanents au microcontrôleur. Certaines broches ont des fonctions spécialisées:

- Série: 0 (RX) et 1 (TX). Permet de recevoir (RX) et transmettre (TX) TTL données série. Ces pins sont connectés aux pins correspondants de l'USB-TTL puce Serial ATmega8U2.
- LED: 13. Il est équipé d'un conduit par la broche numérique 13. LED Lorsque la broche est à la valeur HIGH, la LED est allumée, lorsque la broche est faible, il est hors tension.

L'Uno dispose de 6 entrées analogiques, A0 à A5, dont chacune fournit 10 bits de résolution (ou 1024 valeurs différentes). Par défaut, la tension est de 5 volts. Il est cependant possible de changer la limite supérieure de la gamme en utilisant la broche AREF et la fonction `analogReference`. Autres broches de la carte:

- AREF. Tension de référence pour les entrées analogiques. Pin utilisé avec `analogReference`
- Réinitialiser

II.1.4.3. Communication

Arduino a un certain nombre de moyens pour communiquer avec un ordinateur, une autre carte Arduino, ou autres microcontrôleurs. L'ATmega328 fournit UART TTL (5V) en communication série, disponible sur les broches numériques 0 (RX) et 1 (TX).

II.2. Breadboard

II.2.1. Définition

Breadboard est une carte de circuit utilisée pour faire des circuits temporaires. C'est un dispositif ayant des conceptions d'électronique et de circuits de test. Les éléments électroniques à l'intérieur des circuits électroniques peuvent être interchangés en insérant les bornes et les conducteurs dans des trous et en les connectant plus tard à l'aide de fils appropriés.

L'appareil a des bandes de métal sous la carte qui relie les trous placés sur le dessus de la carte. Les connexions de Breadboard sont pour la plupart temporaires et les éléments peuvent être réassemblés et réutilisés sans aucun dommage. Breadboard sont généralement utilisées en génie électrique. Les ingénieurs utilisent des Breadboard pour tester différents produits fabriqués par eux. L'utilisation de la planche à pain est le moyen le plus efficace de tester et aussi ils sont rentables. Ils peuvent être réutilisés encore et encore dans le but de tester. Aujourd'hui, à partir de minuscules circuits analogiques et numériques jusqu'aux gros processeurs compliqués, tout peut être testé grâce à cela. [11]

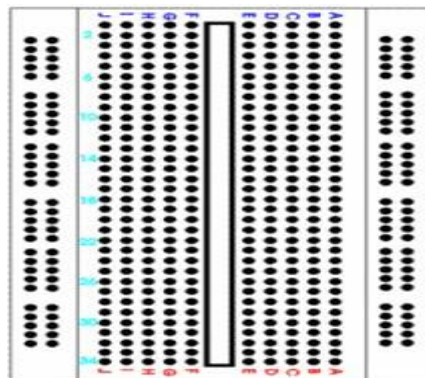


Figure.II. 02:Breadboard (1)

II.2.2. Les bases de Breadboard

Breadboard est un circuit de nature temporaire et utilisé pour tester et prototyper des circuits. Il est facile de prototyper des circuits à l'aide de Breadboard car c'est rapide et facile. Les tableaux de bord sont généralement utilisés pour tester les circuits.

Comme cet appareil a des trous dedans. Afin de former un circuit, les fils sont insérés simplement à l'intérieur des trous. Un avantage de l'utilisation de breadboard est que les

positions des fils peuvent être modifiées si elles sont placées dans un mauvais ordre. Dans le diagramme ci-dessous on peut voir les alphabets sont utilisés afin d'identifier les colonnes verticales et les numéros sont utilisés afin d'identifier les colonnes verticales

Dans le diagramme ci-dessous, vous pouvez voir à la fois les colonnes verticales et horizontales à connecter en interne. Dès la mise sous tension, le courant passe par ces connexions internes.

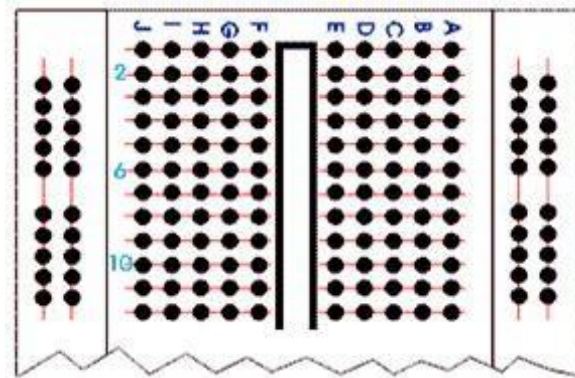


Figure.II.03:Breadboard(2)

Dans le diagramme ci-dessous, vous pouvez voir comment une résistance de 380 ohms et une LED sont installées sur la plaque d'essais. Une batterie de 9 volts est finalement fixée à la lumière LED. Remplacer la résistance de courant avec une résistance ayant 680 ohms, vous pouvez voir la résistance à être plus grande et la lumière LED à être plus faible.

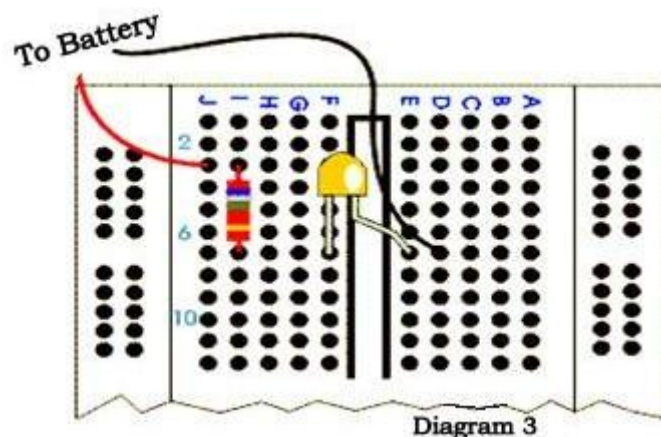


Figure.II 04: Connexion de breadboard avec différent composant(1)

II.2.3. Breadboard Connexion

Breadboard comme mentionné précédemment est utilisée pour faire un circuit temporaire pour les tests et à d'autres fins. L'avantage d'utiliser une carte de test pour tester est que la connexion peut être modifiée si elle est incorrecte. De plus, les parties du circuit ne sont pas endommagées et peuvent facilement être réutilisées. Breadboard consiste généralement en de nombreux trous afin que les fils puissent facilement être enfoncés. Les essais pour presque tous les projets électroniques commencent à partir de Breadboard.

Breadboard a beaucoup de minuscules ornements comme des trous disposés dans une grille de 0,1. Les fils que la plupart des éléments ont peuvent facilement être poussés à l'intérieur de ces trous. Les circuits intégrés sont poussés à l'intérieur de l'espace avec leur point sur la gauche. Les fils standard ne peuvent pas être utilisés pour les plaques d'essai, car ils sont facilement endommagés et, par conséquent, ils nécessitent des fils recouverts d'un seul noyau de plastique d'un diamètre de 0,6 mm. Les fils standard, s'ils sont utilisés, peuvent également endommager la carte.

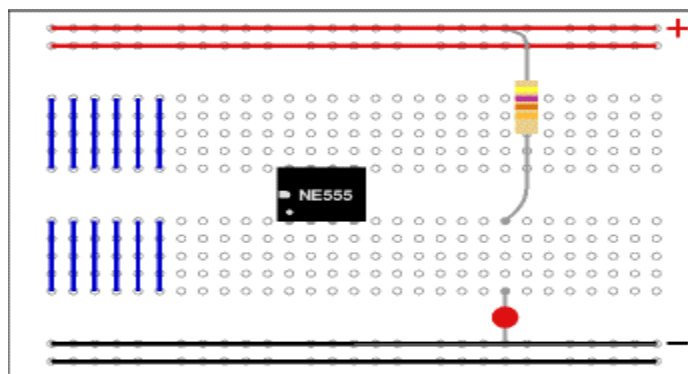


Figure.II.05:Connexion breadboard et différent composant(2)

II.3. Capture Ultrasonic sensor

II.3.1. Définition

Le capteur à ultrasons réfléchissant (capteur réfléchissant convergent à distance ou à réglage de zone) envoie des ondes ultrasoniques d'un émetteur à un objet de détection, reçoit ensuite les ondes réfléchies avec un détecteur. Le capteur utilise les informations sur les avantages pour déterminer la présence d'un objet ou pour mesurer la distance à l'objet. Ce type de capteur détermine la distance du capteur à un objet en fonction du temps nécessaire à partir de l'envoi des ondes ultrasonores jusqu'à la réception de la vitesse du son.

Il existe également des capteurs à faisceau traversant qui détectent la présence d'un objet en détectant l'atténuation ou l'état interrompu des ondes ultrasonores causées par un objet passant entre l'émetteur et le détecteur. [12]



Figure .II.06:Ultrasonic sensor

II.3.2. Caractéristiques

Les couleurs n'influencent pas la détection :

Contrairement aux capteurs photoélectriques, les capteurs à ultrasons peuvent détecter un objet sans être influencés par ses couleurs. Par exemple, si deux objets ont la même forme, même si l'un est transparent, tel que du verre, et l'autre est en plastique noir, ils peuvent tous deux être détectés avec les mêmes paramètres.

➤ Détecter des objets sur une zone étendue

Les capteurs à ultrasons détectent la réflexion à partir d'une zone plus large que les capteurs photoélectriques, ce qui leur permet de contrôler une zone étendue en même temps.

➤ Détection sans contact

Parce que les capteurs ultrasoniques détectent les objets sensibles sans les toucher, ils ne rayent pas les objets sensibles.

II.4. Principe de fonctionnement

Les céramiques piézoélectriques sont utilisées pour la transmission et la réception par ultrasons.

II.4.1. Quelles sont les céramiques piézoélectriques?

Les céramiques piézoélectriques génèrent une force électromotrice entre les électrodes proportionnellement à la quantité de force mécanique appliquée à l'élément. L'inverse est également vrai. Si une tension est appliquée entre les électrodes, le déplacement mécanique est généré proportionnellement à cette tension. A partir de l'amplitude de la force électromotrice, la présence d'un objet est détectée et la distance entre le capteur et l'objet est mesuré

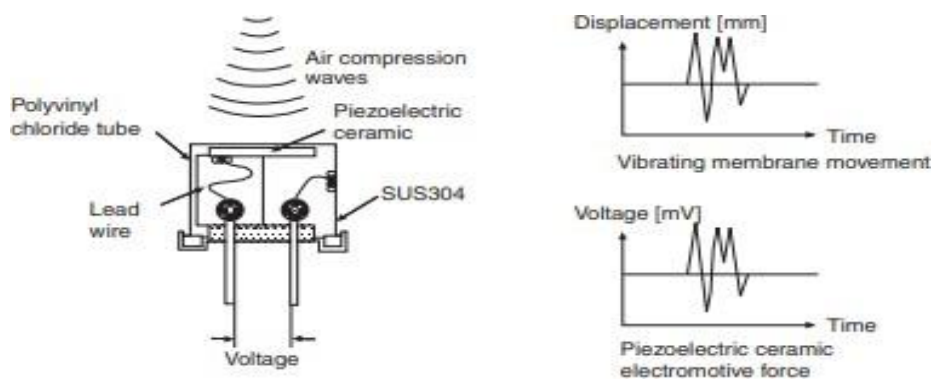


Figure.II. 07: Explicatif des céramiques piézoélectriques

II.5. Types et formes d'objets de détection

• Les objets détectés peuvent être classés comme suit:

A) Objets à surface plane tels que fluides, boîtes, plastique feuilles, papier et verre.

(B) des objets cylindriques tels que des canettes, des bouteilles et des humains corps.

(C) des poudres et des objets en morceaux comme des minéraux, roches, charbon, coke et plastique.

• L'efficacité réfléchissante varie en fonction de la forme de ces objets. Dans le cas de (A), la plus grande quantité d'ondes réfléchies revient, cependant, ceci est fortement affecté par l'inclinaison de l'objet. Dans le cas de (B) et (C), des réflexions parasites se produisent et la lumière réfléchie n'est pas uniforme, cependant, l'effet de l'inclinaison est faible

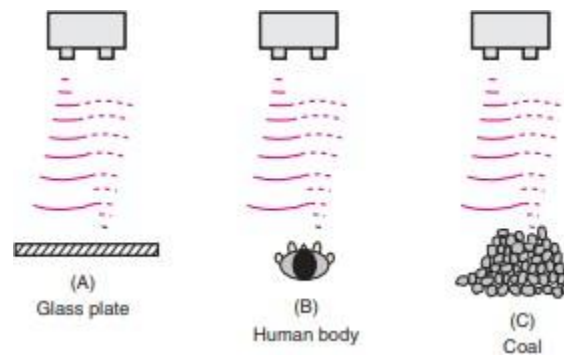


Figure .II.08: Types et formes d'objets de détection

II.6. Buzzer électronique

II.6.1. Définition

Un buzzer est un élément électromécanique ou électronique qui produit un son quand il lui applique une tension. Certains réservent une tension continue, d'autres réservent une tension alternative (transducteurs piézo-électriques). [13]



Figure.II. 09: Buzzer électronique

II.6.2. Caractéristiques techniques

- Matériau : - Boîtier : mélange polycarbonate et ABS.
- Couvercle : en polycarbonate.
- Dimensions : Ø 72 mm x H 40 mm (hauteur de la partie en saillie).
- Résistance à la température : de -20°C à 50°C.

- Puissance : maximum 100 dB (puissance réglable au dos de l'appareil, même une fois installé).
- Fixation : insertion du buzzer dans un perçage de \varnothing 22.5 mm (M22) avec système anti-torsion.
- Câblage : Borniez débrochable à vis max. 1.5 mm².
- Durée de vie : supérieur à 5 000 heures.
- Indice de protection élevé : IP 65. Cet indice indique le degré de protection d'un matériel contre l'accès aux parties dangereuses, la pénétration des corps solides étrangers (1er chiffre) et la pénétration de l'eau avec effets nuisibles (2ème chiffre).

II.7. Photorésistance

Une photorésistance est un composant électronique dont la résistivité fluctue (fréquemment une diminution) selon l'augmentation de lumière qui l'atteint. On peut aussi le nommer résistance photo-dépendante (light-dependent resistor (LDR)) ou photoconducteur.

Une photorésistance se compose d'un semi-conducteur à haute résistivité. Si la lumière incidente est de fréquence suffisamment élevée, les photons absorbés par le semi-conducteur donneront aux électrons liés assez d'énergie pour sauter dans la bande de conduction, les électrons libres (avec leurs trous d'électron) ainsi produits abaissant la résistance de la totalité.

[14]



Figure.II. 10:Photorésistance

II.8. La Résistance

Une résistance est un composant électronique ou électrique dont la caractéristique principale est d'opposer une plus grande résistance à la circulation du courant électrique. La résistance électronique est l'un des composants primordiaux dans le domaine de l'électricité. [15]



Figure .II.11: Résistance

II.9. Module SD

II.9.1. Définition :

L'Arduino SD Card Shield est une solution simple pour transférer des données depuis et vers une carte SD standard. Le brochage est directement compatible avec Arduino, mais peut également être utilisé avec d'autres microcontrôleurs. Il vous permet d'ajouter du stockage de masse et de l'enregistrement de données à votre projet. [16]

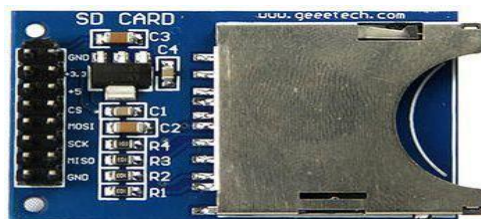


Figure .II.12: Module SD

II.9.2. Câblage de la carte

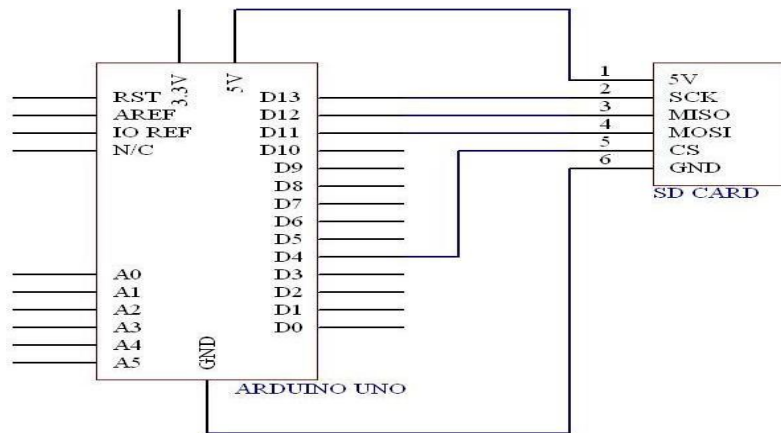


Figure.II.13:Câblage de la carte SD

II.10. Water sensor

II.10.1. Définition :

Le module Capteur d'eau fait partie du système Grove. Il indique si le capteur est sec, humide ou complètement immergé dans l'eau en mesurant la conductivité. Les traces du capteur ont une faible résistance de pullup de 1 M Ω . La résistance va tirer la valeur de trace du capteur jusqu'à ce qu'une goutte d'eau shorts la trace du capteur à la trace à la terre. Croyez-le ou non, ce circuit fonctionnera avec les broches d'E / S numériques de votre Arduino ou vous pouvez l'utiliser avec les broches analogiques pour détecter la quantité de contact induite par l'eau entre les traces mises à la terre et celles du capteur. [17]

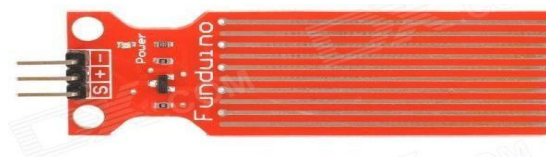


Figure.II.14:Water sensor

II.10.2. Caractéristiques

- Interface compatible Grove
- Basse consommation énergétique

- Module Grove 2.0cm x 2.0cm
- Haute sensibilité

II.10.3. Applications

- Détection des précipitations
- Fuite de liquide
- Détecteur de débordement de réservoir

II.10.4. Câblage du water sensor

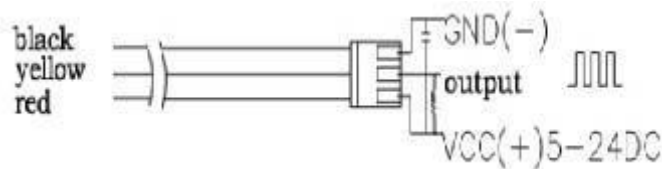


Figure.II.15: Câblage du water sensor

II.11. Conclusion

Dans ce chapitre nous avons présenté les différents composants dont on a besoin pour la réalisation de notre projet (la canne intelligente). On a cité les critères de chaque composant avec ces spécifications techniques ainsi que leurs fonctionnements.

Chapitre 03

Programmation et mise ne marche de notre système

III. Introduction

Après avoir bien détaillé dans le chapitre précédent les différents composants pour la réalisation de notre canne intelligente, nous nous concentrons dans ce chapitre à la programmation, et à la réalisation de notre projet. Donc, nous allons dans un premier temps présenter l'organigramme du programme nécessaire pour le fonctionnement de notre système, ensuite nous allons mettre en pratique et en marche notre réalisation et bien expliquer son mode de fonctionnement.

III.1. Programmation

Afin de mettre en marche notre canne intelligente nous devons en premier lieu injecter le programme conçu avec le langage de programmation Arduino au microcontrôleur de la carte Arduino.

Le programme est représenté par des organigrammes qui décrivent et regroupent les principales étapes à suivre pour assurer le meilleur fonctionnement de notre canne intelligente.

III.2. Aperçu du logiciel ARDUINO

Le logiciel de programmation des modules Arduino est une application Java, libre et multiplateforme compatible avec (Windows, Linux, Mac), servant d'éditeur de code et de compilateur, et qui peut transférer le firmware et le programme au travers de la liaison série (RS-232, Bluetooth ou USB selon le module).

Le langage de programmation utilisé est le C++, compilé avec avr-g++, et lié à la bibliothèque de développement Arduino, permettant l'utilisation de la carte et de ses entrées/sorties. La mise en place de ce langage standard rend aisé le développement de programmes sur les plateformes Arduino, à toute personne maîtrisant le C ou le C++.

L'IDE Arduino est bien connue pour que son code source a seulement deux fonctions principales : un `setup()` et une boucle `loop()`. La fonction `setup()` ne s'exécute qu'une seule fois au début de l'exécution. Comme le nom l'indique, la fonction est faite pour configurer les pré-attributs de la fonction `loop()`. La boucle `loop()` - fonction prend soin de toutes les opérations.

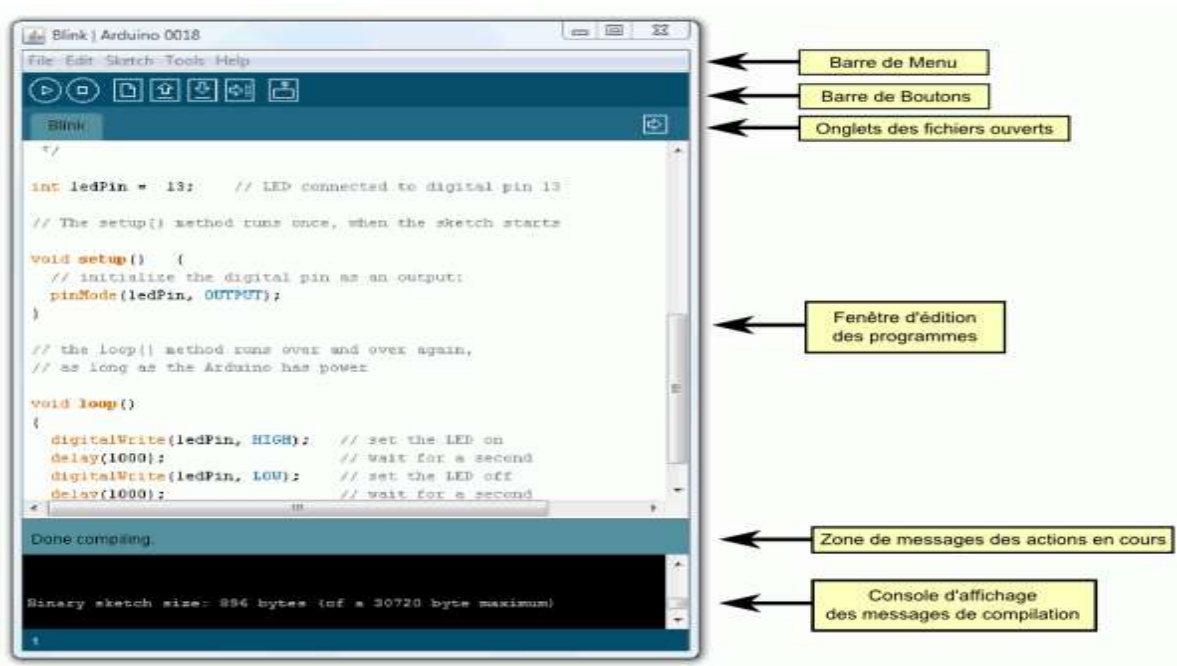


Figure .III. 1 : L'écran principal du logiciel Arduino (interface IDE)

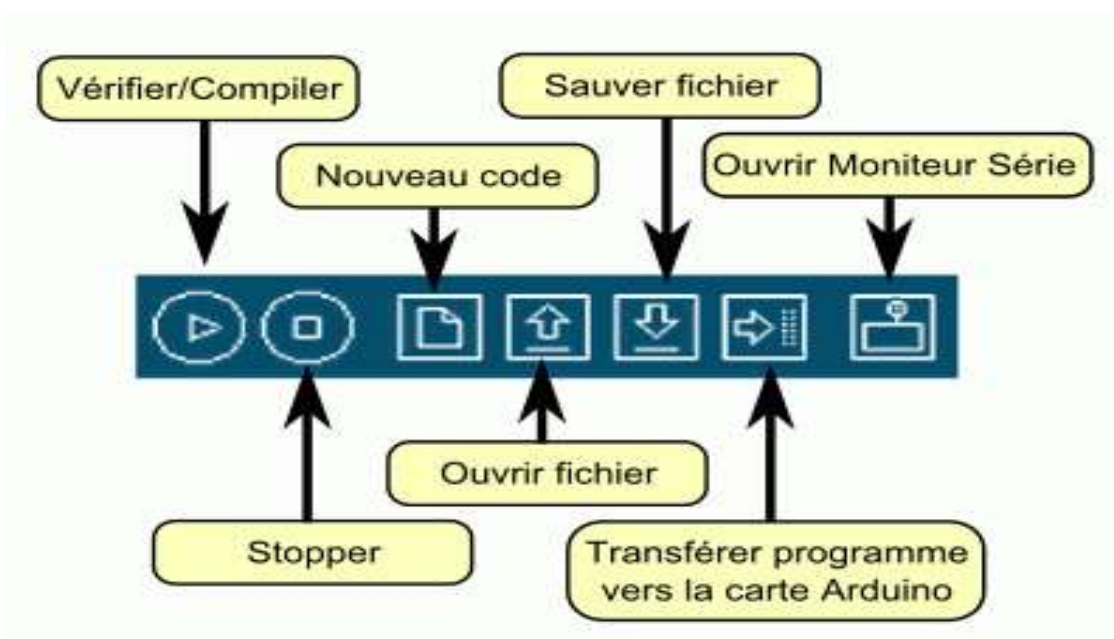


Figure. III. 2 : Détail de la barre de bouton

III.2.1. Principe général d'utilisation

Le code écrit avec le logiciel Arduino est appelé un programme (ou une séquence - sketch en anglais) :

- Ces programmes sont écrits dans **l'éditeur de texte**. Celui-ci a les fonctionnalités usuelles de copier/coller et de rechercher/remplacer le texte.

- La **zone de messages** donne l'état de l'opération en cours lors des sauvegardes, des exportation et affiche également les erreurs.
- La **console texte** affiche les messages produit par le logiciel Arduino incluant des messages d'erreur détaillés et autres informations utiles.
- La **barre de boutons** vous permet de vérifier la syntaxe et de transférer les programmes, créer, ouvrir et sauver votre code, et ouvrir le moniteur série.
- La barre des menus vous permet d'accéder à toutes les fonctionnalités du logiciel Aduino.

III.2.2. Transfert des programmes vers la carte Arduino

Le Transfert des programmes vers la cartes arduino se fait comme suit :

- **Saisir votre programme et vérifier le code**

On suppose ici qu'un programme correctement écrit se trouve dans la fenêtre éditeur. Pour votre première programmation de la carte, aller dans le menu **File>Examples>Digital>Blink** : un programme s'ouvre avec du code dans la fenêtre éditeur.

Appuyez alors sur le bouton **Verify** de la barre d'outils pour lancer la vérification du code :



Si tout va bien, aucun message d'erreur ne doit apparaître dans la console et la zone de message doit afficher **Done Compiling** attestant que la vérification s'est bien déroulée.

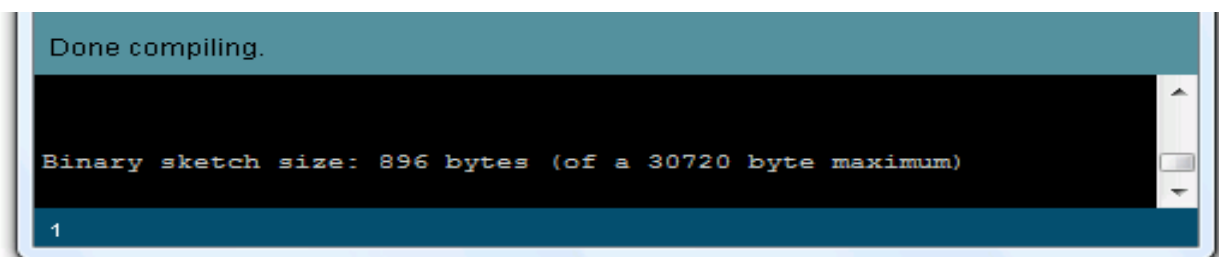


Figure. III. 3 : la vérification terminée

- **Sélectionner le bon port série et la bonne carte puis télécharger le programme :**

Avant de transférer votre programme vers la carte Arduino, vous devriez selectionner la bonne carte Arduino et le bon port depuis **Tools > Serial Port** (Outils > Port Série).

Une fois que vous avez sélectionné le bon port série et la bonne carte Arduino, cliquez sur le bouton **UPLOAD** (Transfert vers la carte) dans la barre d'outils, ou bien sélectionner le menu **File>Upload to I/O board** (Fichier > Transférer vers la carte).

- **Le programme transférer avec succès se lance**

Une fois le transfert terminé, le logiciel Arduino doit afficher un message indiquant que le transfert est bien réalisé, ou montrer des messages d'erreurs... (Reprendre dans ce cas la procédure, et si le problème persiste).

Quand vous transférez un programme, en utilisant le bootloader Arduino, un petit programme (code binaire) a été chargé dans le microcontrôleur sur votre carte Arduino. Cette technique vous permet comme vous avez pu le voir de transférer votre programme sans aucun matériel externe

Une fois que la carte est réinitialisée à la fin du transfert, le dernier programme programmé dans la carte s'exécute.

III.3. Organigramme

III.3.1 le capteur a ultrason

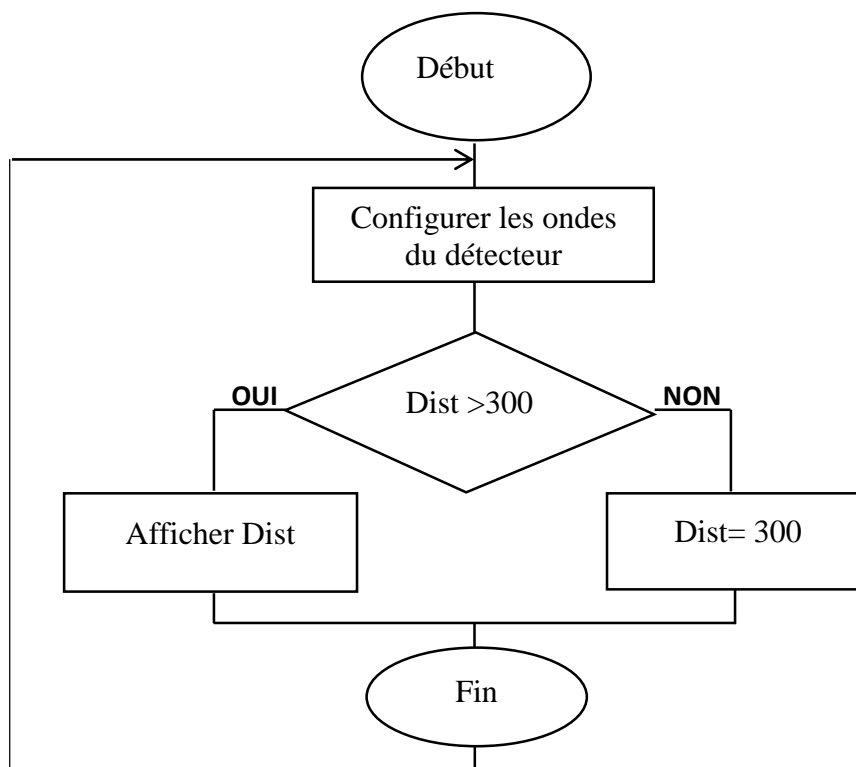


Figure. III. 4 : Organigramme de calcul de distance par le capteur a ultrason

III.3.1.1. Mode de fonctionnement :

Au début nous allons chercher comment l'ultrason travaille et ceci en définissant le temps pris par une seule onde en faisant une émission et réception par l'écho et le trigger en s'appuyant de la loi de la distance suivante

$$\text{Dist} = \text{time_taken} * 0.034 / 2$$

En tenant compte de la limite du périmètre de détection de l'ultrason qui est d'environ 3 à 4 mètre pour ensuite afficher dans le monitor de l'arduino la distance calculer et si elle dans notre cas elle dépasse les 3 mètre on va mettre $\text{Dist} = 300$.

III.3.2. La photorésistance

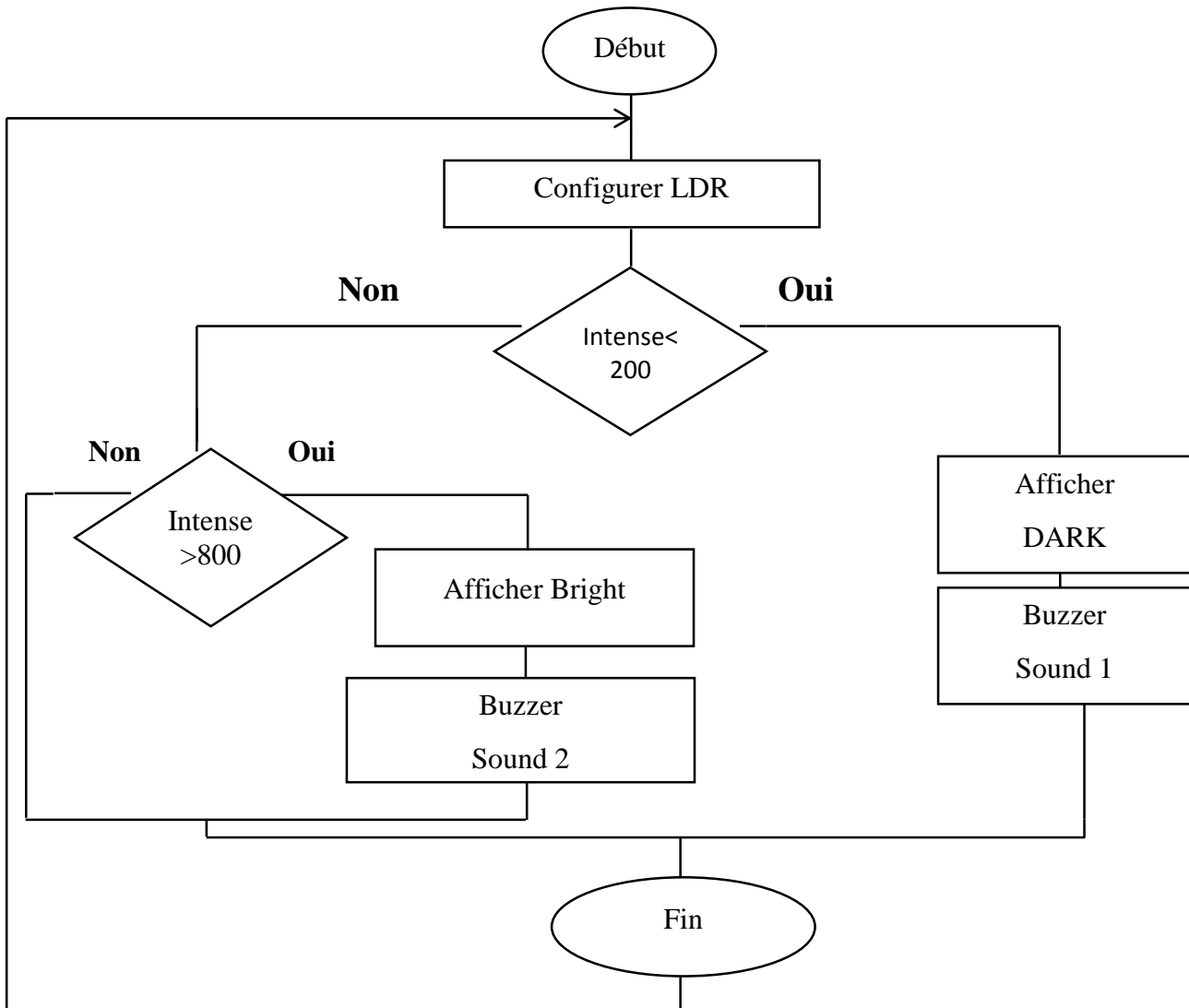


Figure. III. 5 : organigramme de la photorésistance

III.3.2.1. Mode de fonctionnement

Pour la détection de lumière nous allons tout d'abord chercher de trouver les normes d'une zone éclairer en lux et d'une zone sombre on a trouvé que dans une zone dite éclairage normale c'est entre 200 lux et 800 lux c'est pour cela qu'on configure notre photorésistance sur ces normes-là afin de signaler une zone sombre ou une zone très claire comme le montre le tableau suivant.

Si intense < 200

On a un signal spécifique à une zone sombre du buzzer surnomme SOUND 1 avec un affichage sur le monitor << Dark >>.

Si intense > 800

On a un signal spécifique à une zone jugée trop éclairer du buzzer surnomme SOUND 2 avec un affichage sur le monitor << Bright >>.

Activité ou lieu concerné	Éclairage moyen
Sensibilité d'une caméra	0,001 lux
Nuit de pleine lune	0,5 lux
Rue de nuit bien éclairée	20 à 70 lux
Local de vie	100 à 200 lux
Appartement bien éclairé	200 à 400 lux
Local de travail	200 à 3 000 lux
Extérieur par ciel couvert	500 à 25 000 lux

Tableau 3 : norme de lumière en lux

III.3.3. Le capteur a ultrason avec le speaker :

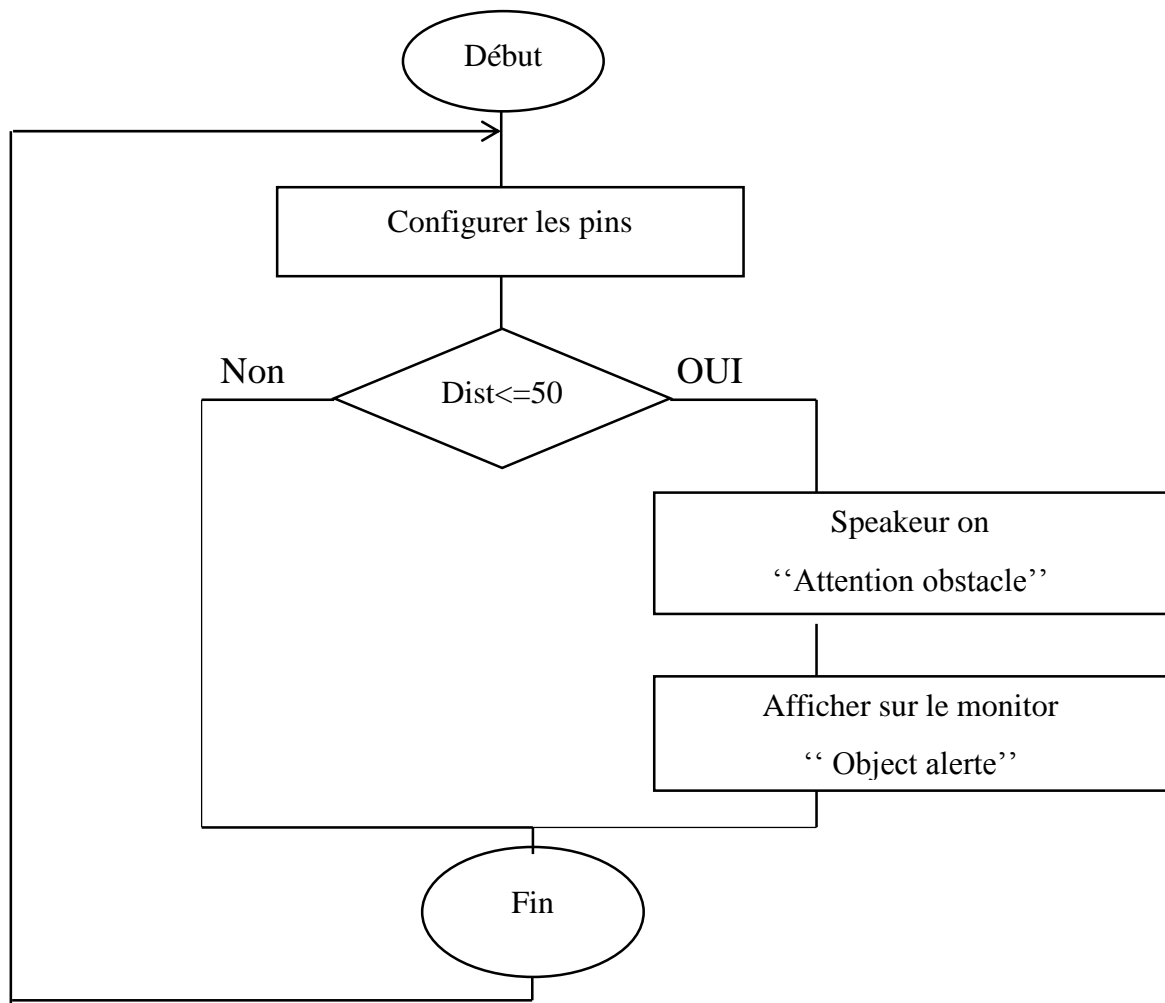


Figure. III. 6 : organigramme du capteur a ultrason avec le speaker

III.3.3.1 Mode de fonctionnement

Nous avons mis en place un détecteur d'obstacle et nous l'avons configuré afin de détecter un obstacle à une distance inférieure ou égale à 50 cm juger une distance assez suffisante pour la personne utilisant la canne intelligente afin de prendre précaution du danger qui va affronter

Et pour alerter cette personne nous avons relié dans notre programme le détecteur d'obstacle à un speaker afin d'alerter la personne d'un éventuel danger.

Si la distance est inférieure à 50 cm le speaker se met en marche en indiquant << attention un obstacle >>.

III.3.4. Le détecteur d'eau

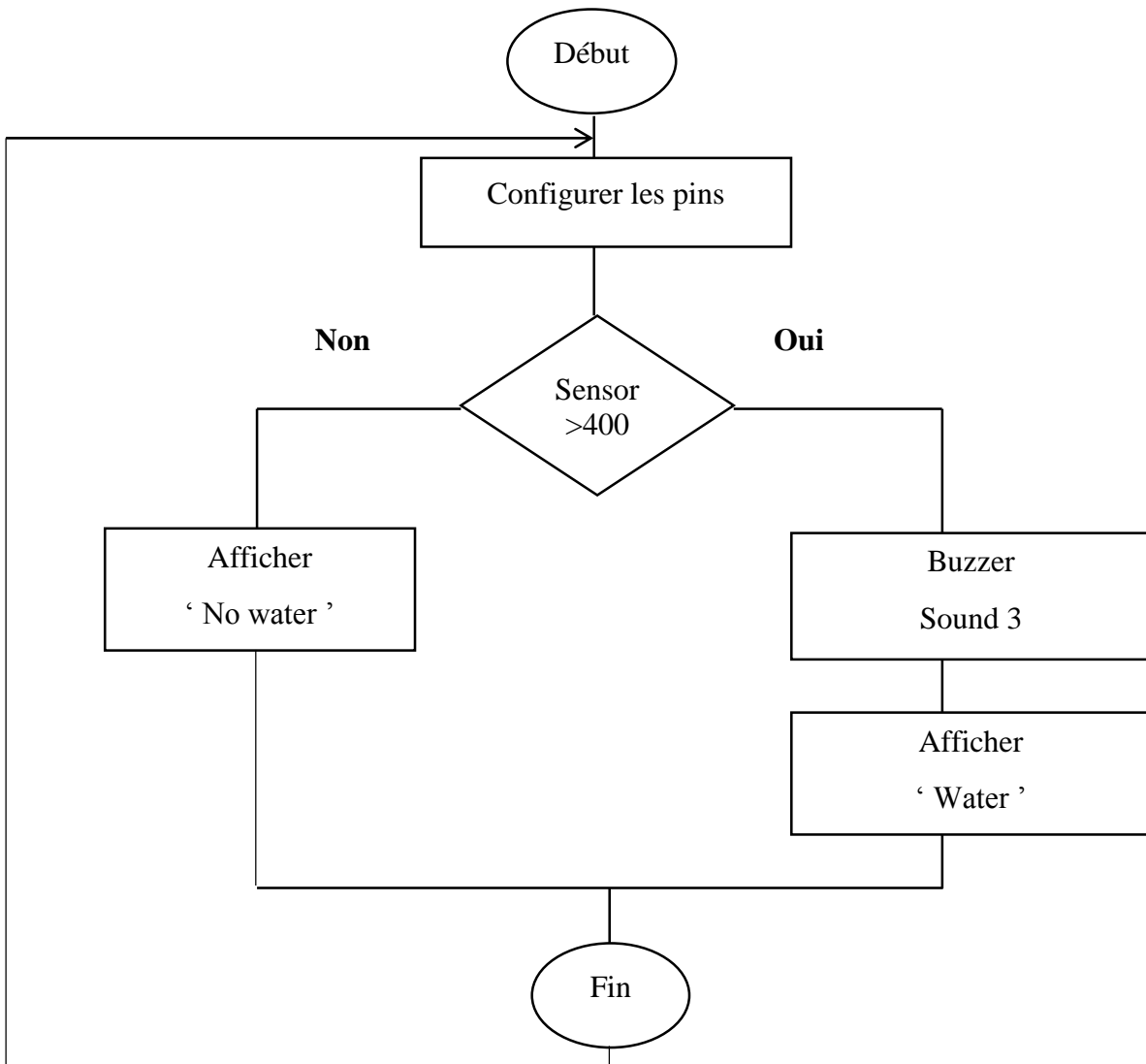


Figure. III. 7 : organigramme du détecteur d'eau

III.3.4.1 Mode de Fonctionnement :

Nous avons mis en place un détecteur d'eau de détecter flaque d'eau pour la personne utilisant note canne intelligente et comme le détecteur d'eau est très sensible il est même sensible à l'humidité comme dans notre cas nous lui avons mis un seuil afin de commencer la détection d'eau qui de 4 cm de niveau d'eau.

Si sensor > 400 le buzzer se déclenche et le monitor affiche << water >>

Sinon buzzer est à l'arrêt et le monitor affiche << No water >>.

III.4. L'organigramme général :

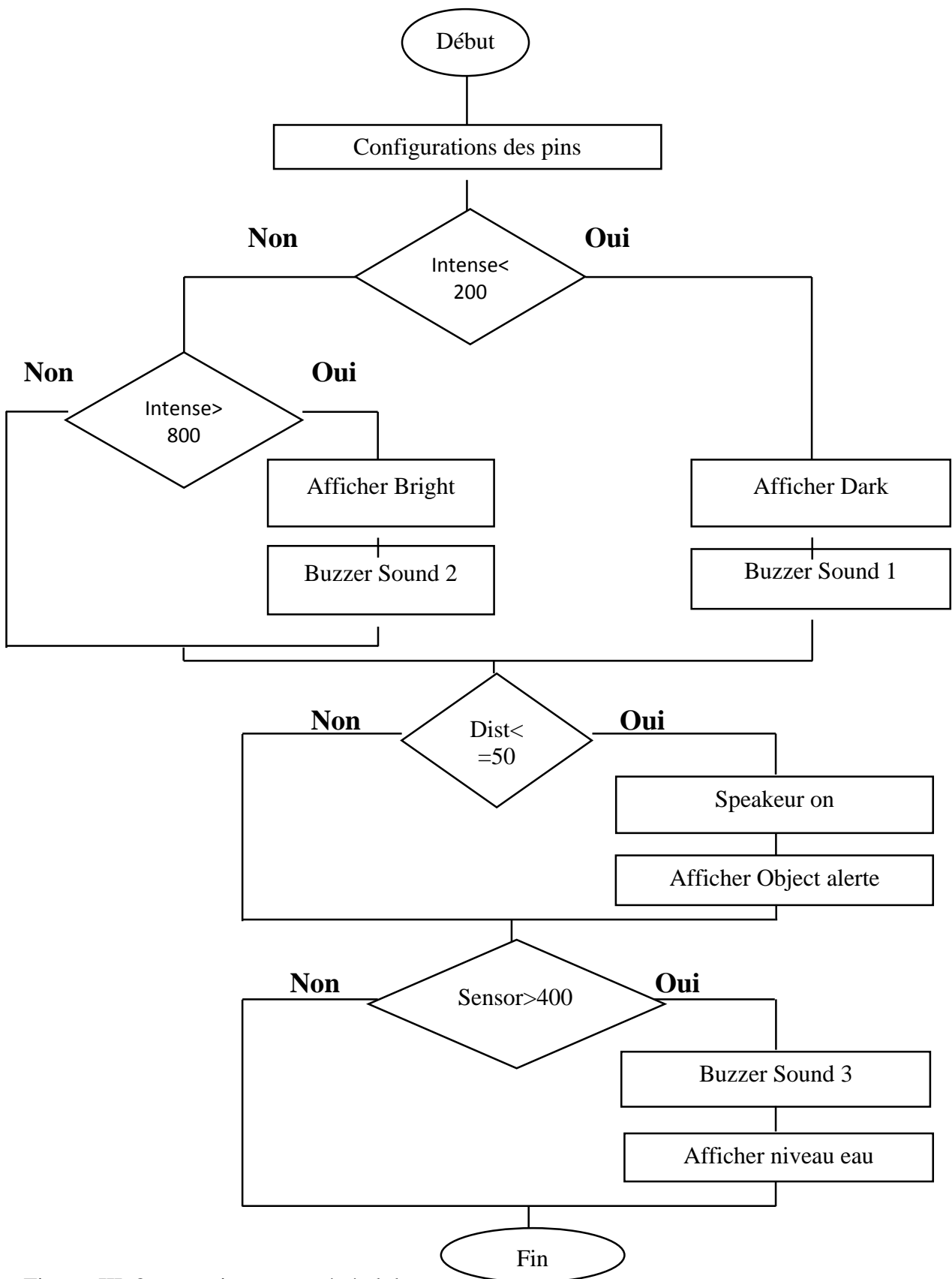


Figure. III. 8 : organigramme général du programme

III.4.1. Mode de fonctionnement

Notre système fonctionne avec des boucles à condition et ces boucles la sont gérer par une hiérarchie de priorité cette dernière diffèrent selon le besoin de l'utilisateur de la canne.

Dans notre cas nous avons mis en place la LDR photorésistance qui détecte la lumière < 200 lux pour l'obscurité accompagné d'un son du buzzer et une lumière > 800 lux pour la luminosité accompagné d'un son du buzzer , ensuite nous l'Avons suivi par un capteur a ultrason reliaer à un speakeur qu'une fois l'obstacle est dans le périmètre du capteur , le speakeur alerte la personne via une voix indiquant (attention un obstacle), pour finir nous avons mis un détecteur d'eau , si un niveau d'eau bien spécifique est détecté le buzzer se met en marche .

III.5. Simulation

Après avoir conçu la partie électronique (alimentation, carte de contrôle/commande), nous allons éditer son schéma électrique à l'aide du logiciel ISIS Proteus comme le montre la figure suivante.

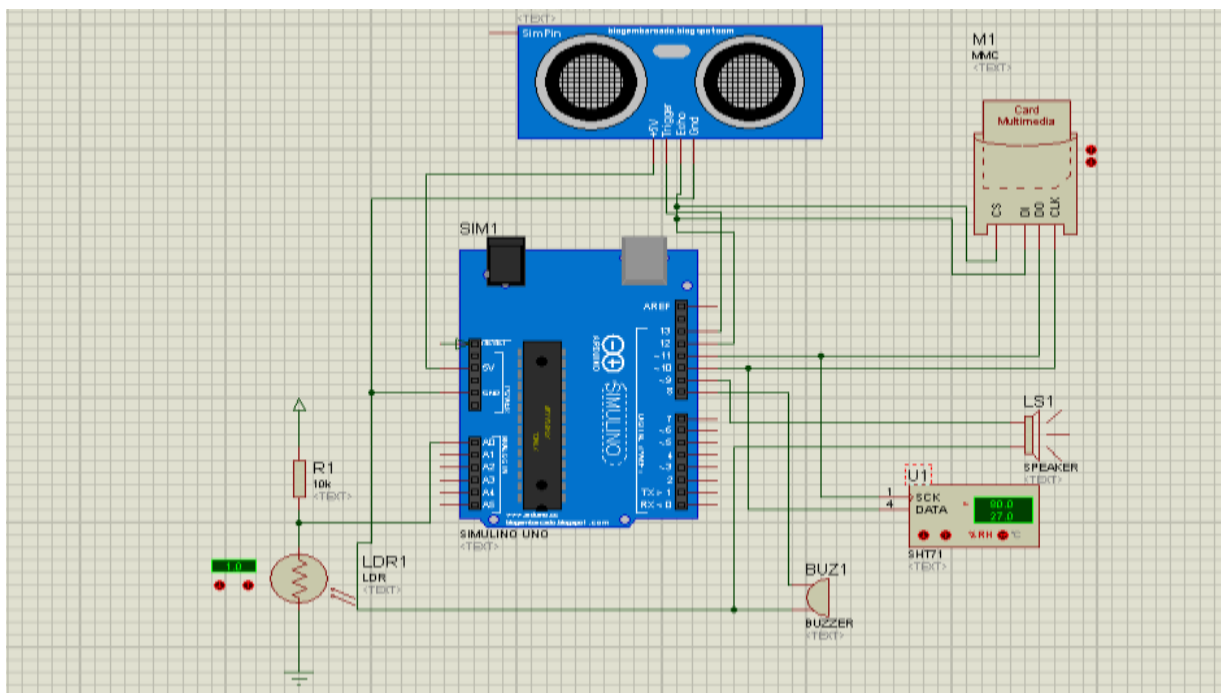


Figure. III. 9 : Simulation du circuit électrique de notre système

III.6. Réalisation et test :

Après les étapes de la conceptions, programmation et simulations du circuit électrique de notre projet, qui nous a permet de déceler et de corriger quelque erreur, nous allons passer à la phase de la réalisation et la démonstration, faite par nos propres moyens.

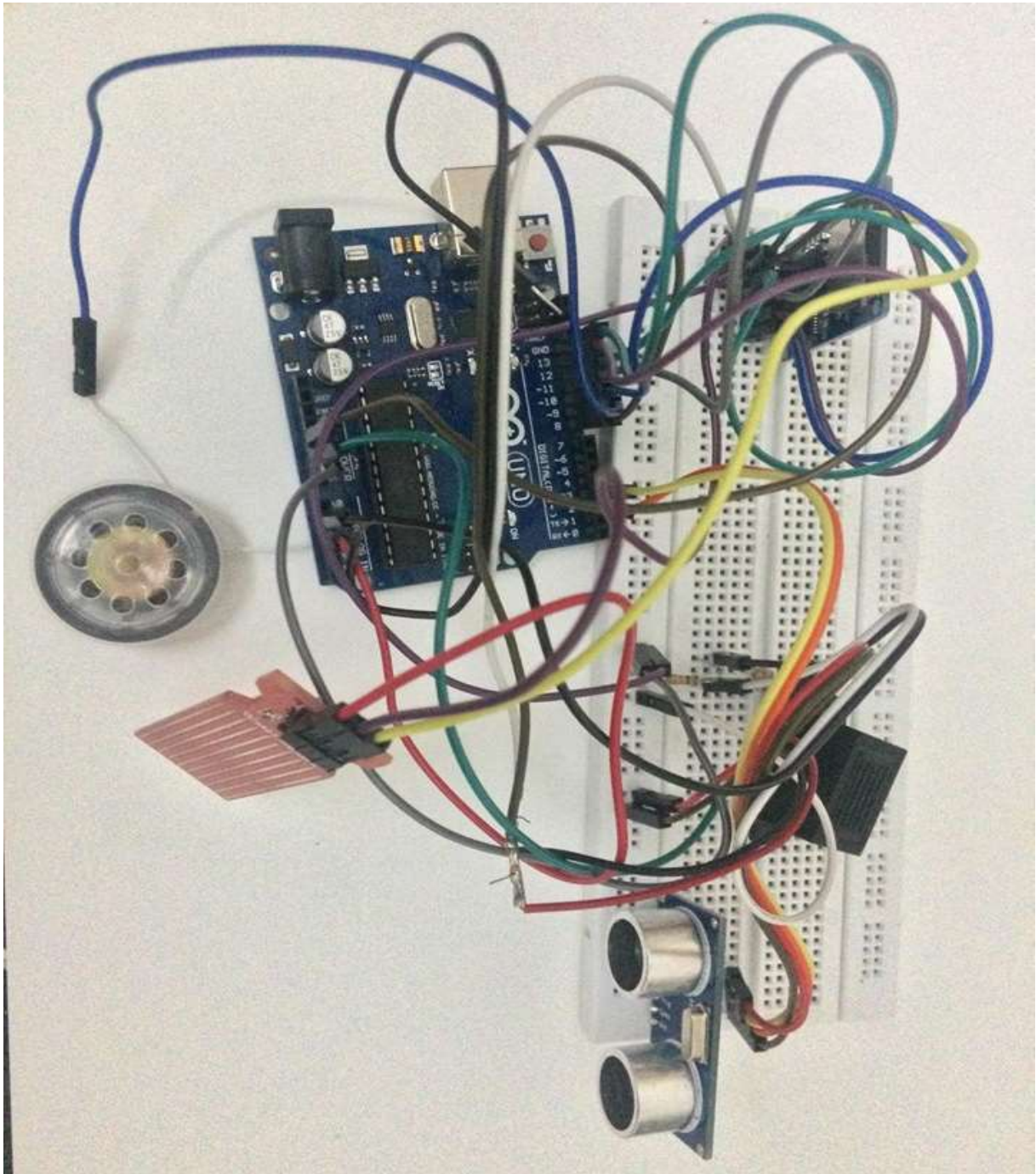


Figure. III. 10 : photo de notre système vu d'en haut

III.7. Conclusion

Dans ce chapitre nous avons d'abord présenter l'organigramme de chaque composant et nous avons bien expliquer comment la programmation a été fait et élaboré sous ARDUINO permettant l'acquisition et le traitement de données et l'affichage des résultats, en effet l'organigramme réaliser décrit les principales étapes à suivre afin de réaliser notre projet et d'assurer un bon fonctionnement du système.

Au final nous avons réalisé et tester notre système et les résultats affichés confirment le bon fonctionnement de notre système permettant aux personnes non-voyantes d'éviter les risques d'obstacles du quotidien et d'assurer leur vie.

Conclusion générale

Conclusion générale :

L'accompagnement des personnes mal voyantes et âgées est un des grands défis des années à venir. L'enjeu est économique et sociétal. L'allongement de la durée de vie, la volonté des personnes âgées de changer leur quotidien longtemps possible, les transformations de nos modes de vie, la nécessité de pouvoir accueillir les personnes âgées dans les structures sûres, rechercher des solutions innovantes pour mieux répondre à ces changements sociétaux.

Le travail présenté dans ce mémoire, s'inscrit dans le cadre de notre projet de fin d'études, intitulé «Exploitation des systèmes embarqués pour la réalisation d'une canne intelligente». Il a été le fruit d'une longue réflexion, pendant laquelle il a fallu répondre à la problématique, qui consistait à faciliter le déplacement des personnes non-voyantes et leur rendre les tâches de la vie quotidiennes plus aisées et faciles. Pour ceci, et dans le but de bien cerner la problématique, il a fallu, dans un premier temps s'intéresser aux systèmes embarqués et systèmes temps réel.

Ensuite, et dans un deuxième temps nous avons présenté les éléments constituant le système, en donnant la définition, l'architecture, et le principe de fonctionnement de chaque élément ; nous avons cité les outils nécessaires pour la conception.

Et on a finalisé notre projet avec la programmation et la simulation de notre canne intelligente.

De même nous avons élaboré un programme approprié en langage Arduino pour permettre le bon fonctionnement du système. Une simulation complète du circuit électrique de commande a été effectuée, vérifiant ainsi le bon fonctionnement du système.

Nous pouvons affirmer que ce projet a constitué pour nous une bonne expérience, dans la mesure où il nous a permis, non seulement de mobiliser toutes les connaissances acquises auparavant, mais aussi d'apprendre plus de techniques, et d'explorer des logiciels de simulation et programmation. Également, ce projet nous a introduits au domaine de l'électronique médicale.

En effet, le travail élaboré n'est qu'une étape primaire d'un travail plus avancé. Il est, donc susceptible d'être amélioré. À cet effet, nous proposons en perspective la réalisation d'une canne intelligente munie d'un système de reconnaissance et de classification des objets relié à

Conclusion générale

un GPS pour mémoriser les itinéraires et les lieux fréquentés quotidiennement et guider cette personne ou qu'elle soit à ces lieux.

Bibliographie

- [1] C.alonso, B.estibals ,“conception et commande de système électrique embarques,” université paul Sébastien Toulouse III, 17 jan 2013.[online] available <http://galilee.ups-tlse.fr/inter/syllabus/Sylla%20L3%20ProCCSEE-Toulouse.pdf> .
- [2] Web sémantique sur les système embarques, “ Système embarques (se) ,”université de nantes,15-mai-2012.[online]available<http://pagesperso.lina.univ-nantes.fr/~prie-y/archives/VEILLE-2009-2012/2012/wsembarq/3.html> .
- [3] T.Agarwal ,“what is embedded systems and its real time applications,” electronics projects focus.
- [4] A. benaissa, “les systèmes embarques, ”mémoire master, institut d’électronique et d’informatique gaspard-Monge, paris, 2003.
- [5] T.Argwal, “What is the difference between Von Neumann and Harvard Architecture, ”EDGEFX, 16-oct-2013.
- [6] S.Combefis, “Programmations des systemes embarques, ” UKO universite de louvin, 9-fev.2015.[Online].Available :<https://www.ukonline.be/cours/embeddedsystems/programming/chapitre1-3>.
- [7] “système embarque, ”Wikipédia.6 avr 2018.
- [8] F. Cottet, E. grolleau, S. Gerard, J. Huges, Y. Ouhammou, S.Tucci-piergiovanni :Systemes temps reel embarques specification,conception,implementation et validation temporelle, : DUNOD.2005.2014.
- [9] M. Mathias perrolaz, “détection d’obstacles multi-capteurs supervisée par stéréovision, ” thèse de doctorat, université pierre et marie curie, paris, 13 juin 2008.
- [10] Xavier HINAULT, “Arduino”, [online].available: <https://store.arduino.cc/usa/arduino-uno-rev3>. [Accessed: 2018]
- [11]ADMINISTRATOR, “Breadboard Basics and Connections”, Electronics hub, 10-aout-2015.[Online].available: <https://www.electronicshub.org/breadboard-basics-and-connections/>. [Accessed: 15-04-2012]
- [12] CSM ultrasonic ,“Technical Explanation for Ultrasonic Sensors”, Omron, 2014.[online].Available:

Bibliographie

https://www.ia.omron.com/data_pdf/guide/50/ultrasonic_tg_e_1_1.pdf .

[Accessed: 13-05-2018].

[13] Slam Dunk “buzzer”, Seton, 2013. [Online].available:

https://www.seton.be/media/cms/files/stbefr/DMEU_TDS_2391_std.lang.all.pdf . [Accessed:

25-03-2018]

[14] “Photorésistance”, Wikipédia.25-03-2018.

[15] “Résistance”, Wikipédia.08-05-2018

[16] X .HINAULT, “Arduino SD card module”, Geeetech, 23-07-2016.[online].Available:

http://www.geeetech.com/wiki/index.php/Arduino_SD_card_Module . [Accessed: 04-05-

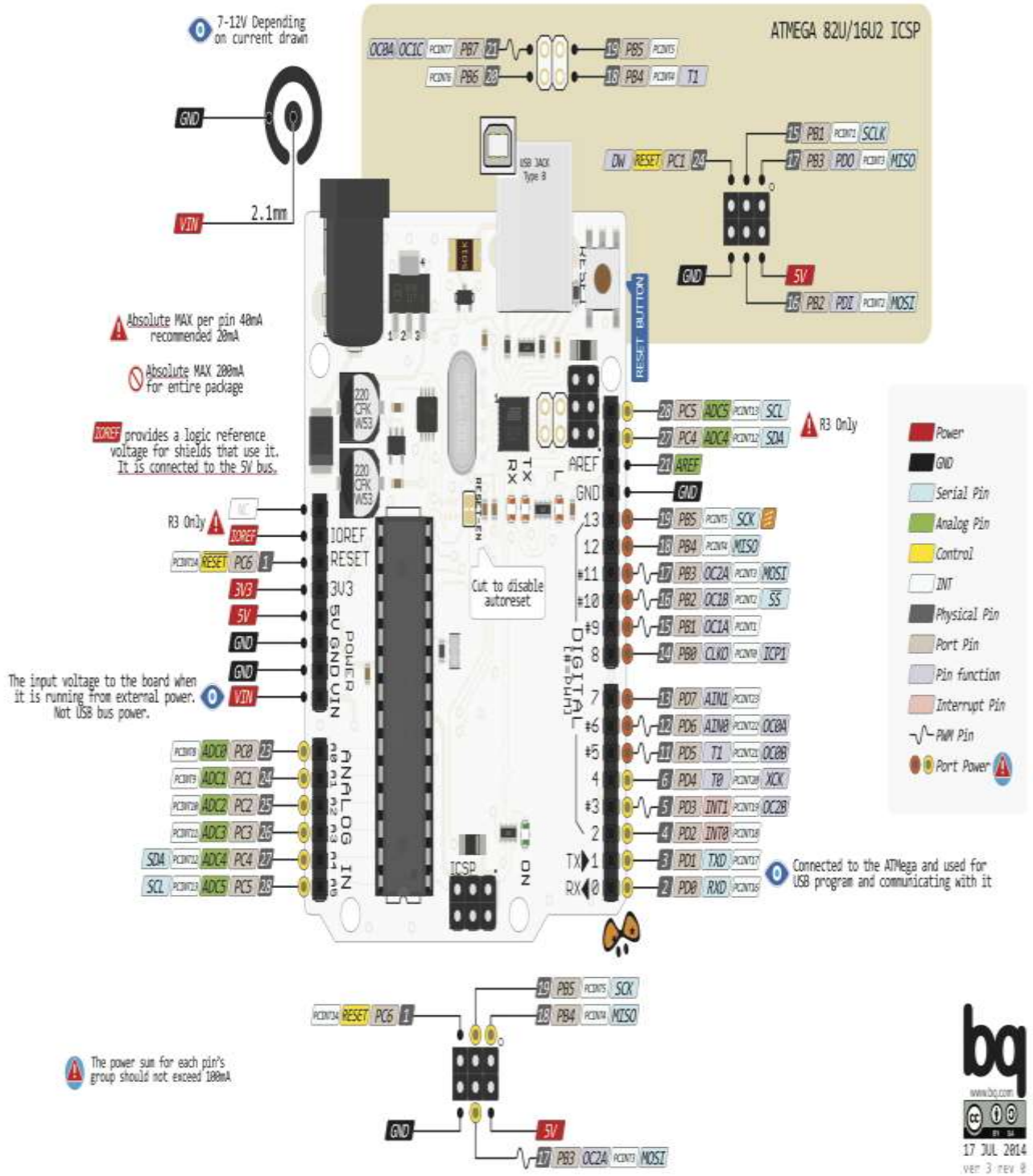
2018]

[17] seeed technologie “water sensor”, mouser , 25-03-2014. [online].available :

https://www.mouser.com/ds/2/744/Seeed_101020018-1217524.pdf . [Accessed: 01-04-2017]

La carte Arduino Uno

UNO PINOUT

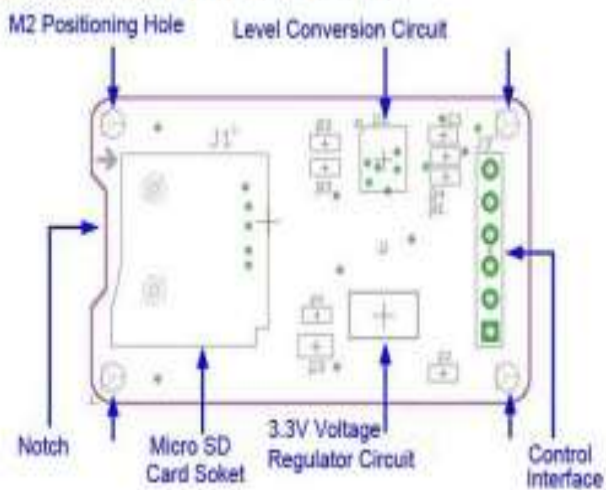


Le Module SD

Interface Parameters:

Items	Min	Typical	Max	Unit
Power Voltage VCC	4.5	5	5.5	V
Current	0.2	80	200	mA
Interface Electrical Potential	3.3 or 5			V
Support Card Type	Micro SD Card(<=2G), <u>Mirco</u> SDHC Card(<=32G)			—
Size	42X24X12			mm
Weight	5			g

Mirco SD Card Interface Module:



La photorésistance LDR



Email: info@sunrom.com or sunrom@gmail.com

Visit us at <http://www.sunrom.com>

Document: Datasheet

Date: 28-Jul-08

Model #: 3190

Product's Page: www.sunrom.com/p-510.html

Light Dependent Resistor - LDR

Two cadmium sulphide(cds) photoconductive cells with spectral responses similar to that of the human eye. The cell resistance falls with increasing light intensity. Applications include smoke detection, automatic lighting control, batch counting and burglar alarm systems.



Applications

Photoconductive cells are used in many different types of circuits and applications.

Analog Applications

- Camera Exposure Control
- Auto Slide Focus - dual cell
- Photocopy Machines - density of toner
- Colorimetric Test Equipment
- Densitometer
- Electronic Scales - dual cell
- Automatic Gain Control – modulated light source
- Automated Rear View Mirror

Digital Applications

- Automatic Headlight Dimmer
- Night Light Control
- Oil Burner Flame Out
- Street Light Control
- Absence / Presence (beam breaker)
- Position Sensor

Electrical Characteristics

Parameter	Conditions	Min	Typ	Max	Unit
Cell resistance	1000 LUX	-	400	-	Ohm
	10 LUX	-	9	-	K Ohm
Dark Resistance	-	-	1	-	M Ohm
Dark Capacitance	-	-	3.5	-	pF
Rise Time	1000 LUX	-	2.8	-	ms
	10 LUX	-	18	-	ms
Fall Time	1000 LUX	-	48	-	ms
	10 LUX	-	120	-	ms
Voltage AC/DC Peak		-	-	320	V max
Current		-	-	75	mA max
Power Dissipation				100	mW max
Operating Temperature		-60	-	+75	Deg. C

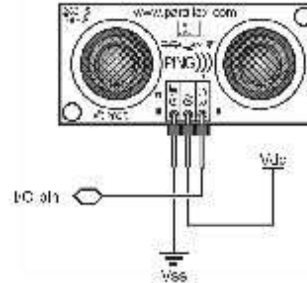


Le capteur a ultrason

Pin Definitions

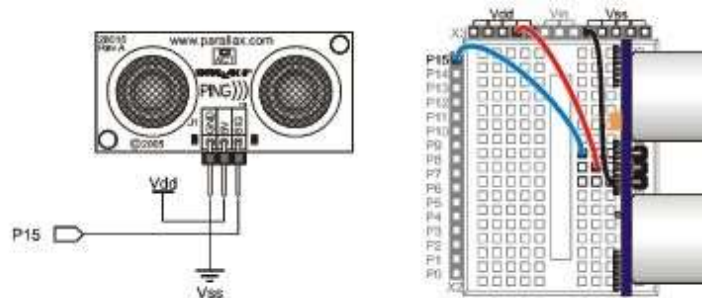
GND	Ground (Vss)
5 V	5 VDC (Vdd)
SIG	Signal (I/O pin)

The PING))) sensor has a male 3-pin header used to supply power (5 VDC), ground, and signal. The header allows the sensor to be plugged into a solderless breadboard, or to be located remotely through the use of a standard servo extender cable (Parallax part #805-00002). Standard connections are show in the diagram to the right.



Quick-Start Circuit

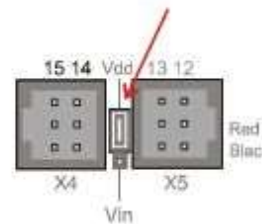
This circuit allows you to quickly connect your PING))) sensor to a BASIC Stamp[®] 2 via the Board of Education[®] breadboard area. The PING))) module's GND pin connects to Vss, the 5 V pin connects to Vdd, and the SIG pin connects to I/O pin P15. This circuit will work with the example program Ping_Demo.BS2 listed on page 7.



Servo Cable and Port Cautions

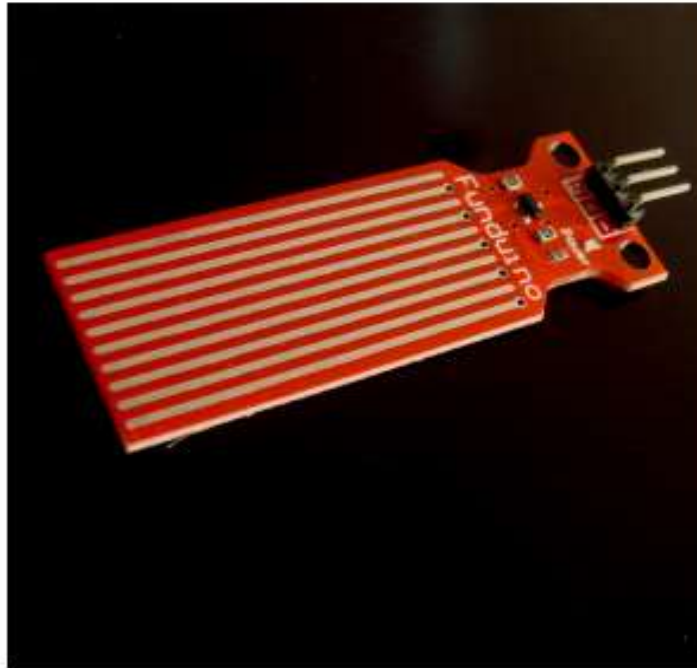
If you want to connect your PING))) sensor to a Board of Education using a servo extension cable, follow these steps:

1. When plugging the cable onto the PING))) sensor, connect Black to GND, Red to 5 V, and White to SIG.
2. Check to see if your Board of Education servo ports have a jumper, as shown at right.
3. If your Board of Education servo ports have a jumper, set it to Vdd as shown.
4. If your Board of Education servo ports do not have a jumper, do not use them with the PING))) sensor. These ports only provide Vin, not Vdd, and this may damage your PING))) sensor. Go to the next step.
5. Connect the servo cable directly to the breadboard with a 3-pin header. Then, use jumper wires to connect Black to Vss, Red to Vdd, and White to I/O pin P15.



Board of Education Servo Port Jumper, Set to Vdd

Water Sensor Module User's Manual



1. Notice

(1) did not carefully read the instructions before you do not give the driver board is powered !
Avoid faulty wiring caused permanent damage to the drive plate .

(2) Please carefully check pin function , attention condensed identifier , correct wiring ! Do not
reverse the power cord , resulting in
Electronic devices burned.

2. the product introduction

2013 latest Water Sensor is a Easy to use, compact and lightweight , high cost of water , droplets
identification and detection sensors. This sensor is working The principle is to measure the size
of the trace amount of water droplets through the line with a series of parallel wires exposed .
And domestic and foreign Products compared not only small , powerful, and cleverly designed
with the following features : First, the amount of water to simulate Conversion ; Second, plasticity
, based on the sensor output analog values ; Third, low power consumption , high sensitivity ;
Fourth, can Directly connected to a microprocessor or other logic circuitry , and the controller
board for a variety of , for example : ArduinoController , STC microcontroller , AVR microcontroller
and so on .

3, the specification parameters

- 1 Product Name: water level sensor
- 2 Item :. K-0135
- 3 Operating voltage :. DC5V
- 4 Working current : less than 20mA
- 5 Sensor Type : Analog
- 6 detection area :. 40mm x16mm
- 7 Production process :. FR4 double-sided HASL
- 8 mounting hole size : 3.0mm
- 9 user-friendly design : half-moon -slip handle depression
- 10 Working temperature :. 10 °C -30 °C
- 11 Operating Humidity : 10% ~ 90 % non -condensing
- 12 Weight :. 3g
- 13 Product Dimensions : 65mm x 20mm x 8mm

4., the test Water Sensor Module

We use the Arduino controller to be tested , need to use hardware devices as follows :

- 1, Arduino controller × 1
- 2, Arduino sensor expansion board × 1
- 3, Water Sensor Module × 1
- 4, 3P sensor cable × 2
- 5, IR & LED Modue (red) × 1
- 6, USB data communication cable × 1

Water Sensor DuPont line will be connected to the Arduino sensor expansion board interface A1. The use of sensors

The red line will be connected to the Arduino piranha light sensor expansion board D8. After completing the hardware connection , the code is compiled

After downloading the Arduino inside .

Le Buzzer



Email: info@sunrom.com
Visit us at <http://www.sunrom.com>

Document: Datasheet

Date: 28-Sep-10

Model #: 3113

Product's Page: www.sunrom.com/p-320.html

Buzzer - Electromagnetic

This buzzer is an electromagnetic type audio signaling device, which has a coil inside which oscillates a metal plate against another, which when given voltage difference produces sound of a predefined frequency. You must be aware of such sounds of buzzer like BEEP sound in many appliances.



Features

- These high reliability electromagnetic buzzers are applicable to general electronics equipment.
- Compact, pin terminal type electromagnetic buzzer with 2048 Hz output.
- Pin type terminal construction enables direct mounting onto printed circuit boards.

Applications

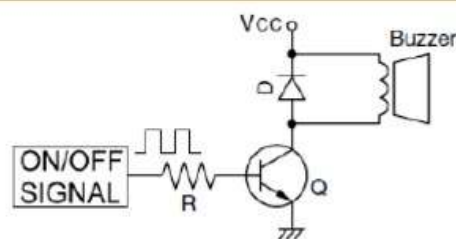
Security Alerts, Clocks, travel watches, keyboards, toys, various alarms of equipments.

Specification

Parameter	Value
Operating Voltage	3-15V DC
Current Consumption	60mA
Frequency	2048 Hz
Sound Pressure at 10cm	80 db at 12V DC
Coil DC Resistance	60 Ohms
Operating Temperature	-40 to +80 deg C

Recommended Operating Circuit

As the buzzer uses a coil, it has an inductive load. Protect the drive circuit by putting the diodes in parallel into the buzzer.



ملخص

كرس هذا العمل لدراسة و تصميم عصا ذكية للأشخاص اللذين يعانون من الضعف البصري و هذا من اجل مساعدتهم علي تخطي كل انواع العقبات اللاتي من المحتمل مواجهتها في الحياة اليومية و كذلك من اجل الكشف عن الاماكن المظلمة و المضيئة و عن المياه. وهذا ضمانا لسلامتهم و تبسيطا للمهام اليومية و كل هذا بتكلفة منخفضة.

Résumé

L'objectif de notre travail est la réalisation d'une canne intelligente pour les malvoyants cette dernière a pour mission de détecter différent type d'obstacle afin d'éviter tout incident probable.

Notre canne permet de veiller sur leur sécurité et simplifié les taches de leur quotidien avec un cout réduit par l'intégration d'une multitude de capteurs.

Abstract

The objective of our work is the realization of an intelligent cane for the visually impaired, the latter's mission is to detect different types of obstacles in order to avoid any probable incident.

Our cane makes it possible to watch over their safety and simplify the tasks of their daily life with a reduced cost by the integration of a multitude of sensors.