

**REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE**  
*Ministère de L'Enseignement Supérieure et de la Recherche Scientifique*  
*Université Mouloud MAMMERI de Tizi-Ouzou*  
*Faculté de Génie Electrique et Informatique*  
*Département Informatique*



***MEMOIRE de FIN D'ETUDES***  
***DE MASTER ACADÉMIQUE***

*Domaine : Mathématique et informatique*

*Filière : Informatique.*

*Spécialité : Systèmes informatiques.*

*Présenté par :*

*BERSI Ouiza*

*LEBBAL Nassima.*

**Thème**

**Conception et réalisation d'un système de surveillance  
(température et mouvement) d'une chambre de  
malade/nourrisson.**

Mémoire soutenu publiquement le 10/07/2017 devant le jury composé de :

Président : M<sup>r</sup> M.HABET.

Promoteur : M<sup>r</sup> C.HEMDANI.

Examinatrice : M<sup>elle</sup> L.BOUGCHICHE.

Examinatrice : M<sup>elle</sup> Y.YESLI.

Promotion 2016/2017

# Remerciement

*En préambule à ce mémoire*

*Nous remercions ALLAH qui nous a aidés et nous a donnés la patience et le courage*

*Durant*

*Ces longues années d'études.*

*Nous souhaitons adresser nos remerciements les plus sincères aux personnes qui*

*Nous*

*ont apporté leur aide et qui ont contribué à l'élaboration de ce*

*Mémoire*

*Nous tenant à remercier sincèrement Mr C.HEMMDANI en tant que Encadreur,*

*qui a*

*Toujours montré à l'écoute et très disponible tout au long de la réalisation de ce*

*mémoire,*

*Nous tenons à remercier les membres de jury qui ont bien accepté de juger notre*

*travail.*

*Enfin, nous adressons nos plus sincères remerciements à tous nos proches et*

*amis,*

*qui nous ont toujours soutenus et encouragés*

*au cours de la réalisation de ce mémoire.*

## *Dédicace*

*Je rends grâce à Dieu qui nous a fourni le courage pour mener à bien notre travail malgré les obstacles rencontrés.*

*À la mémoire de ma grand mère « Aldjia » ;*

*À mes grands parents*

*Que dieu leurs préserve bonne santé et longue vie.*

*À mes très chers parents :*

*Votre confiance et vos encouragements ont été pour moi la première source de persévérance. Que vous trouvez dans ces modestes mots le témoignage de ma gratitude et ma sincère appréciation.*

*A mon unique et cher frère, Yacine, pour son inquiétude et ces précieux conseils.*

*À tous mes oncles ;*

*À mes chères tantes ;*

*À mes chers cousins et cousines ;*

*À mes chères amies « Celia, lila, sonia et lydia »*

*A ma chère amie et camarade Ouiza ainsi qu'à toute sa famille.*

*Et à tout mes amies sans exception, Et je remercie tous ceux qui ont contribué à la réussite de ce travail de près ou de loin.*

*Nassima*

# *Dédicace*

*Je rends grâce à Dieu qui nous a fourni le courage pour mener à bien notre travail malgré les obstacles rencontrés.*

*Je dédie ce modeste travail aux êtres qui me sont les plus chers :*

*D'abord à mes parents pour leurs confiance, leurs sacrifices, leurs soutien inconditionnel depuis ma tendre enfance, ceux qui ont toujours veillé à ma réussite, ceux qui m'ont jamais privé d'amour et d'encouragement.*

*Aucun hommage ne pourrait être à la hauteur du bonheur dont ils ne cessent de me combler.*

*A mon unique frère, mon héro Mouloud pour son inquiétude et ces précieux conseils.*

*Aux plus adorables soeurs au monde Horia et sa belle famille , lynda et Chahrazed et mon cher neveu Moumouh ainsi qu'à toute ma famille sans exception.*

*A mon cher époux Bousaad celui qui m'a soutenu tout au long de cette rude période.*

*A mes beaux parents.*

*A mes belles sœurs Madila , Samia , Hanane et Lyna .et mon beaux frère Bélaid  
A ma meilleure amie au monde Flafla et à toute sa famille*

*A ma chère amie et camarade Sara ainsi qu'à toute sa famille.*

*Ouiza*

# Sommaire

Sommaire .....	I
Liste des figures.....	IV
Liste des tables.....	VII
Introduction générale.....	VIII
<b>I. Domotique et systèmes embarqués</b>	
I.1. Introduction.....	1
I.2 Domotique.....	1
I.2.1 Définition : .....	1
I.2.2 Les principes de la domotique.....	1
I.2.3 Les fonctions de la domotique .....	2
I.2.4 Composants principaux d'un système domotique.....	4
I.2.5 Fonctionnement de la domotique .....	6
I.2.6 Les modes de communication .....	7
I.3 Systèmes embarqués .....	10
I.3.1. Système embarqué .....	10
I.3.2.Historique .....	11
I.3.3.Comparaison aux systèmes informatiques standards :.....	11
I.3.5.Les domaines d'application des systèmes embarqués.....	12
I.3.6.Caractéristiques des systèmes embarqués.....	13
I.3.7.Classification des systèmes embarqués :.....	13
I.3.8.Architecture d'un système embarqué :.....	14
I.3.8.Conclusion .....	15
<b>I. Présentation de l'Arduino</b>	
II.1 Introduction : .....	16
II.5 La carte Arduino UNO .....	18
II.5.1 La partie matérielle : .....	19
II.5.1.1 Un microcontrôleur : .....	19
II.5.1.3 Visualisation.....	20
II.5.1.4 La connectique .....	21

II.5.2.1 l'environnement de la programmation .....	22
II.5.2.2 Structure générale de l'interface de programmation .....	22
II.5.2.3 Structure générale d'un programme Arduino.....	23
II.5.2.4 Téléverser un programme dans la carte.....	24
II.6 Les Shields de la carte Arduino .....	26
II.6.1.1 Le module Arduino Bluetooth.....	26
II.6.1.2 Le module Shield Arduino Wifi.....	27
II.6.1.3 le module XBee: .....	27
II.6.2 les capteurs.....	28
II.6.3. divers.....	28
□ Les moteurs électriques .....	28
□ Les afficheurs LCD .....	29
II.7.La différence entre les modules de communication (wifi, Bluetooth et Xbee).....	30

### **III.Conception**

III.1 Introduction .....	31
III.2 Structure générale du système.....	31
III.3 Conception matérielle .....	32
III.3.1 Fritzing. ....	32
III.3.2 Capteurs .....	33
III.3.3.Le bouton.....	36
Figure III.7. Câblage d'un bouton poussoir.....	37
III.3.4. Les actionneurs .....	37
III.3.4.1 Les modules de communications RF (radio fréquence) .....	37
III.3.4.2 L'afficheur LCD .....	39
III.3.4.3 Les LEDs .....	41
III.3.4.4 Le Buzzer .....	42
III.3.5.le contrôleur :.....	42
Schéma générale du système.....	42
III.4.Conception logicielle.....	45
III.4.1 Le principe de fonctionnement .....	45
III.4.1.1 L'organigramme de sous-système (a) :.....	46
III.4.1.2 L'organigramme de sous-système (b) :.....	48

III.5 Conclusion.....	48
<b>IV. Réalisation</b>	
IV.1 Introduction.....	49
IV.2 Les composants matériels utilisés .....	49
Tableau IV.1 l'ensemble de matériel utilisés. ....	49
IV.3Branchement des différents composants avec la carte Arduino : .....	50
IV.3.1.les module RF.....	50
IV.3.2.les capteurs .....	51
IV.3.3.Les LEDs.....	52
IV.3.4.Le Buzzer. ....	52
IV.3.5 Écran LCD.....	53
IV.3.6.le Bouton poussoir.....	53
IV.4 Implémentation de l'application : .....	53
IV.4.1.la description de quelques fonctions et librairies utilisées dans le sous-système (a)54	
IV.4.1.1 Au niveau de la partie déclarative.....	54
IV.4.1.2 Au niveau de la fonction d'initialisation setup () : .....	54
IV.4.1.3 Au niveau de la boucle principale loop() :.....	54
IV.4.2.la description de quelques fonctions et librairies utilisées dans le sous-système (b)55	
IV.4.2.1 Au niveau de la partie déclarative.....	55
IV.4.2.2 Au niveau de la partie setup() .....	55
IV.4.2.3 Au niveau de la partie loop() .....	55
IV.5 Conclusion .....	56
Conclusion générale et pespectives .....	57
Annexe .....	58
Bibliographie.....	63

# Liste des figures

Figure I.1 : maison intelligente.....	1
Figure I.2 : Les fonctions de la domotique.....	2
Figure I.3 : Détecteur et capteur.....	4
Figure I.4 : Actionneur (moteurs).....	5
Figure I.5 : Centrale domotique.....	5
Figure I.6 : Exemple d'interface de commande (écran tactile).....	6
Figure I.7: communication courant porteur en ligne.....	8
Figure I.8 : Communication pour bus filaire.....	9
Figure I.9 communication sans fil.....	10
Figure I. 1: Système embarqué dans son environnement.....	11
Figure I. 2: Architecture d'un système embarqué.....	14
Figure II. 1. Carte Arduino UNO.....	16
Figure II. 2. Microcontrôleur Atmega 328p.....	19
Figure II. 3. Constitution de la carte Arduino UNO.....	22
Figure II. 4. Interface IDE Arduino.....	23
Figure II. 5. Comment choisir le type de la carte.....	24
Figure II. 6. Comment choisir le port série de la carte.....	25
Figure II. 7. Compilation et cahrgement du code Arduino.....	26
Figure II. 8. Les modules Bluetooth.....	27
Figure II. 9. Le Shield Wifi.....	27
Figure II. 10. Le module XBee.....	28
Figure II. 11. Exemples de capteurs.....	28
Figure II. 12. Les moteurs électriques.....	29

<b>Figure II. 13.Afficheur LCD.....</b>	<b>29</b>
<b>Figure II. 14.Le relais.....</b>	<b>30</b>
<b>Figure III.1.structure de notre système. ....</b>	<b>32</b>
<b>Figure III.2.le logiciel Fritizing.....</b>	<b>33</b>
<b>Figure III. 3. Capteur de température. ....</b>	<b>34</b>
<b>Figure III. 4. Câblage du capteur de température à la carte Arduino.....</b>	<b>35</b>
<b>Figure III. 5. Capteur de mouvement.....</b>	<b>35</b>
<b>Figure III.6.Câblage du capteur de mouvement.....</b>	<b>36</b>
<b>Figure III.7. Câblage d'un bouton poussoir. ....</b>	<b>37</b>
<b>Figure III.8.les modules RF. ....</b>	<b>37</b>
<b>Figure III.9.Câblage de module RF (partie émetteur).....</b>	<b>38</b>
<b>Figure III.10. Câblage de module RF (partie récepteur). ....</b>	<b>39</b>
<b>Figure III.11.L'afficheur LCD.....</b>	<b>40</b>
<b>Figure III.12.Câblage d'un afficheur LCD.....</b>	<b>40</b>
<b>Figure III. 13. LED.....</b>	<b>41</b>
<b>Figure III.14. Câblage d'une LED.....</b>	<b>41</b>
<b>Figure III.15.Câblage d'une alarme (buzzer).....</b>	<b>42</b>
<b>Figure III.16.Schéma du sous-système (a).....</b>	<b>43</b>
<b>Figure III.17 Schéma du sous-système(b).....</b>	<b>44</b>
<b>Figure III. 18 .L'organigramme du fonctionnement du système (a).....</b>	<b>47</b>
<b>Figure III. 19 .L'organigramme du fonctionnement du système (b).....</b>	<b>48</b>
<b>La figure IV.1 Branchement du module RF (émetteur).....</b>	<b>50</b>
<b>Figure IV.2. Branchement du module RF (récepteur).....</b>	<b>50</b>
<b>Figure IV.3 Branchement du capteur de température LM35.....</b>	<b>51</b>
<b>Figure IV.4 Branchement du capteur de mouvement PIR.....</b>	<b>51</b>

<b>Figure IV.5 Branchement d'une LED .....</b>	<b>52</b>
<b>Figure IV.6 présente le branchement d'un Buzzer.....</b>	<b>52</b>
<b>Figure IV.7 Branchement du LCD.....</b>	<b>53</b>
<b>Figure IV.8 : Branchement du bouton poussoir.....</b>	<b>53</b>
<b>Figure A.1 le programme du sous système (a).....</b>	<b>59</b>
<b>Figure A.2 le programme du sous système (b).....</b>	<b>60</b>
<b>Figure A.3.branchement du sous-système (a).....</b>	<b>61</b>
<b>Figure A.3.le branchement du sous-système (b).....</b>	<b>62</b>

# Liste des Tables.

I.1. Comparaison entre l'informatique standard et l'informatique embarqué.....	12
II.1.Caractéristiques de quelque carte Arduino.....	18
IV.1 l'ensemble de matériel utilisés .....	49

## I.1. Introduction

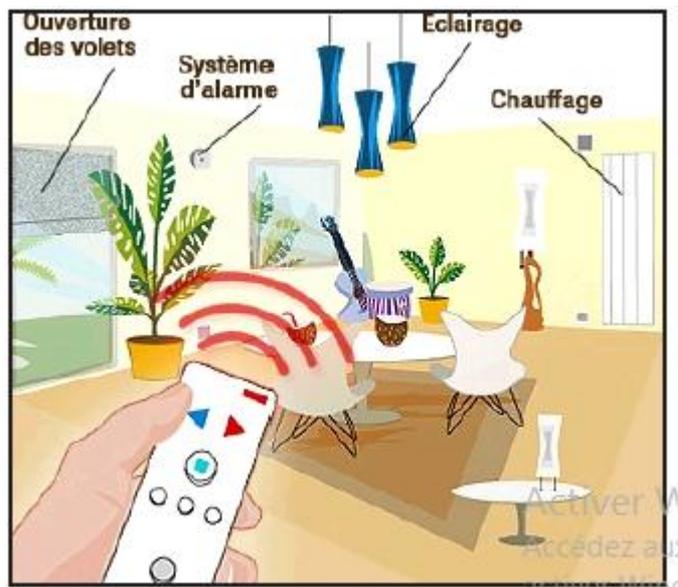
Dans ce chapitre nous présenterons d'une manière sommaire la technologie domotique, ces principaux domaines d'utilisation, son fonctionnement, et les différents modes d'utilisation.

Nous présenterons également les systèmes embarqués, leurs caractéristiques et enfin leur architecture.

## I.2 Domotique

### I.2.1 Définition :

Ce mot issu du latin "Domus" qui signifie maison, "tique " qui signifie automatique, est l'ensemble des technologies de l'électronique, de l'information et des télécommunications permettant de contrôler et d'automatiser une maison. [1]



**Figure I.1 : maison intelligente.**

Elle repose sur la mise en réseau d'un grand nombre d'appareils électriques qui peuvent être gérés de manière centralisée : Elle couvre les portes automatiques, les systèmes de sécurité et de télésurveillance, le chauffage, la gestion de l'énergie, de l'équipement électroménager et audiovisuel...etc.la figure I.1 présente un exemple d'une maison intelligente.

### I.2.2 Les principes de la domotique

La domotique, c'est la communication des objets de la maison pour une gestion intelligente et centralisée. Pour la mettre en œuvre, il faut établir un ou plusieurs réseaux de communication entre les différents appareils que nous souhaitons contrôler. Il existe

# CHAPITRE I.LA DOMOTIQUE ET LES SYSTEMES EMBARQUES

différentes solutions pour les faire communiquer. Ces solutions peuvent se catégoriser selon le mode de transmission qui conviendra à notre maison : l'infrarouge, le courant porteur (CPL), les ondes radios (RF), le réseau Ethernet (LAN ou Wifi), les réseaux 1-wire et d'autres réseaux dont le protocole est spécifique et protégé par une marque.[2]

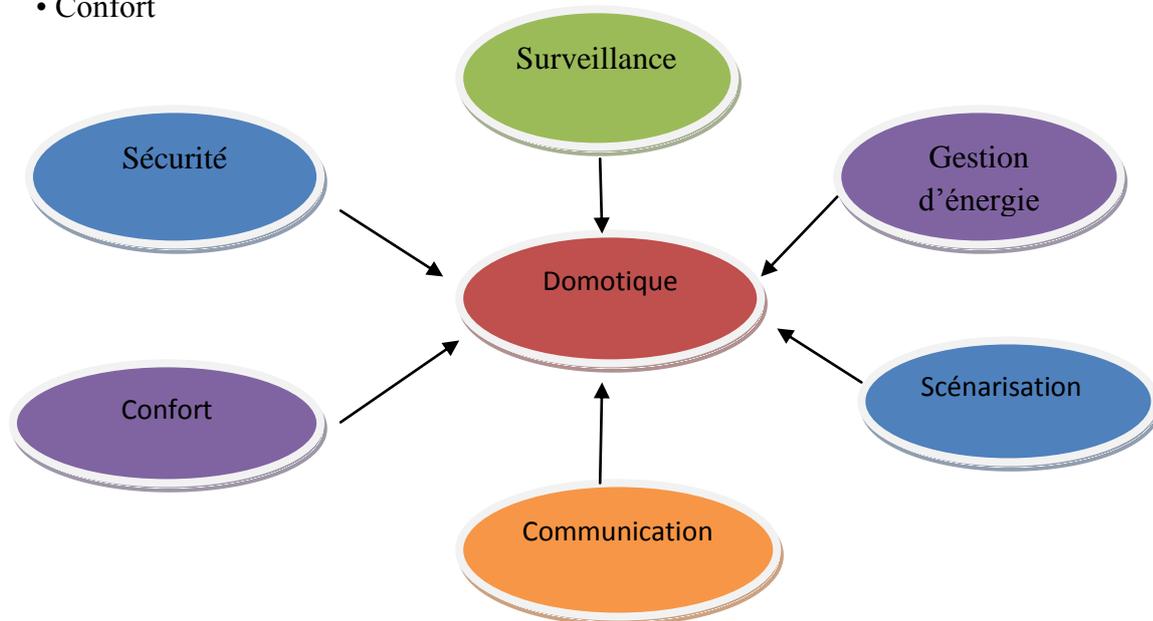
Les réseaux les plus connus sont listés ci-dessous:

- Infrarouge : IRDA, RC5 Philips, etc.
- Courant porteur : X10, X2D Deltadore, In One Legrand, etc
- Ondes radios : RF433 mhz, Bluetooth, Chacon, Zwave, XBee, protocoles spécifiques, etc
- Réseaux Ethernet : IP, Wifi

## I.2.3 Les fonctions de la domotique

Dans cette partie nous allons présenter les domaines d'applications que l'on peut réaliser grâce aux technologies intégrées dans la domotique :

- Sécurité
- Surveillance
- Gestion de l'énergie
- Scénarisation des actions
- Communication
- Confort



**Figure I.2 : Les fonctions de la domotique.**

## **Sécurité**

Contrairement à un système d'alarme traditionnel, une centrale domotique agit sur toute l'installation électrique de l'habitation. Elle dissuade les intrus en simulant une présence par l'allumage aléatoire des éclairages, de la radio ou l'ouverture des volets durant la journée, Les accès à un logement ou à un commerce sont contrôlés et enregistrés et, si une intrusion est détectée, la centrale prend les mesures qui s'imposent tels que :

- Allumage de tous les éclairages de la maison.
- Appel d'un centre de surveillance, d'un voisin ou d'un téléphone mobile À partir d'un téléphone, il est possible d'écouter et de s'adresser directement aux intrus grâce au hautparleur de la centrale.

Certains system proposent même d'écouter les bruits au sein de son logement via un téléphone mobile pour être sûr qu'il ya eu réellement une intrusion.

## **Surveillance**

Différents capteurs détectent les anomalies :

- inondation,
- incendie,
- fuite de gaz,
- arrêt du congélateur,
- coupure de courant...etc

La centrale intervient instantanément pour couper les alimentations, remonter les stores, couvrir la piscine, appeler les numéros d'urgence ou faire retentir la sirène si l'occupant est présent.

## **Gestion de l'énergie**

Toutes les possibilités offertes par la domotique (veille, gestion de l'éclairage et du chauffage) permettent de réduire les dépenses d'énergie de 6 à 10 %. Pour ce faire, les appareils de la maison sont reliés à un ordinateur central grâce au Wifi, un réseau de câbles ou un réseau électrique. Il suffit de rentrer en mémoire des applications comme : la fermeture des volets, l'extinction de l'éclairage, la programmation du chauffage, la mise en route de l'alarme, etc. L'ordinateur transmettra les ordres programmés à un instant T, comme : arroser le jardin ou encore allumer le four à 11 h 30 .Une seule pression, même à distance, sur une télécommande ou un écran tactile, permettra d'activer un ou plusieurs de ces scénarios.[3]

## **Scénarisation des actions**

Au moment de quitter un habitat ou un commerce, la mise en fonction de l'alarme déclenche une série de contrôles et d'actions, (centralisation des commandes) :

- fermeture de toutes les lumières,
- coupure de l'arrivée de gaz,

# CHAPITRE I.LA DOMOTIQUE ET LES SYSTEMES EMBARQUES

- vérification de la fermeture de toutes les fenêtres,
- allumage de la lumière extérieure durant quelques minutes s'il fait nuit, ...

## Communication

La communication est un concept grâce auquel nous pouvons, à partir d'un dispositif disposé dans n'importe quelle pièce, piloter l'ensemble de la maison (éclairage, chauffage, fermeture des volets ...).

## Confort

Ouvrir le portail sans descendre de voiture, arroser automatiquement le jardin, ouvrir ou fermer les volets ou les stores, disposer de la télévision dans plusieurs pièces, adapter la température aux conditions extérieures, ...etc. Autant de fonctionnalités qui peuvent devenir indispensables.

Un émetteur radio, une télécommande infrarouge, un téléphone, une horloge ou un détecteur : différents dispositifs de commande peuvent agir sur le même appareil et un même détecteur peut engendrer des actions différentes (ex : commander un éclairage à partir d'une télécommande tout en conservant les fonctions de l'interrupteur mural). Un détecteur de présence peut donner l'alarme en cas d'absence de l'occupant mais allumera la lumière dans l'autre cas. [4]

## I.2.4 Composants principaux d'un système domotique

### Capteur et détecteur

Ce sont des dispositifs transformant l'état d'une grandeur physique observée en une grandeur utilisable. Ils renvoient des informations afin de contrôler les paramètres du bâtiment et du système. Cela peut être la présence ou non de personne, de fuite de gaz, les températures à différents endroits. La figure I.3 illustre deux exemples, à gauche est un détecteur de fumé et à droite un capteur de lumière



Figure I.3 : Détecteur et capteur.

### Actionneurs

## CHAPITRE I.LA DOMOTIQUE ET LES SYSTEMES EMBARQUES

Les actionneurs sont placés dans la Partie Opérative d'un système automatisé, ils permettent de transformer l'énergie reçue en un phénomène physique (déplacement, dégagement de chaleur, émission de lumière ...).

La figure I.4 présente différents types de moteurs (actionneurs).

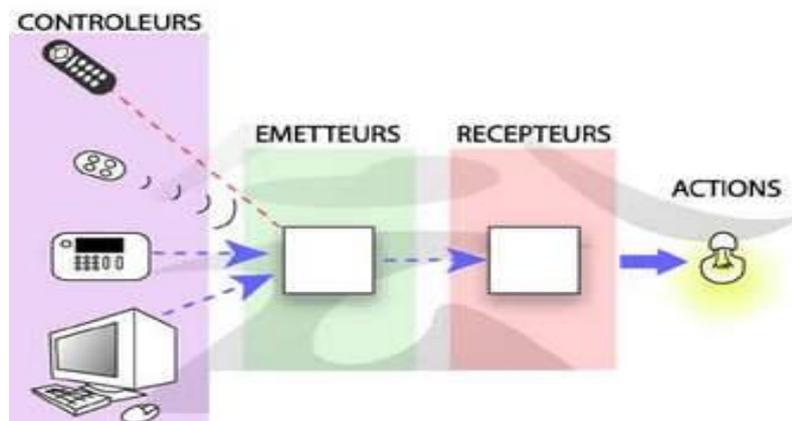


**Figure I.4 : Actionneur (moteurs).**

### Centrale domotique

La centrale domotique ou aussi box domotique est le cerveau de l'installation domotique. La box domotique est un accessoire relié à un réseau internet qui permet d'interagir à distance avec tous les éléments de la maison, comme les éléments d'automatisation, les équipements de confort, les éléments multimédia, et les accessoires de sécurité contrairement à une centrale, elle n'est pas obligatoirement reliée à l'internet.

La figure I.5 présente un exemple d'une centrale domotique.



**Figure I.5 : Centrale domotique.**

## Interface de commande

L'interface de commande permet de piloter et de paramétrer en temps réel le fonctionnement des appareils domotiques, plusieurs interfaces de gestion sont possibles :

- Un écran de contrôle tactile.
- Une télécommande domotique.
- Un ordinateur ou une tablette.
- Un Smartphone.

Chaque interface a des avantages en termes d'installation et de simplicité d'utilisation. Une interface communique avec les appareils domotiques en passant par des box domotique, qui regroupe toutes les connexions des appareils électriques.

La figure I.6 présente un exemple d'interface de commande (écran tactile).



**Figure I.6 : Exemple d'interface de commande (écran tactile).**

## I.2.5 Fonctionnement de la domotique

Pour faire fonctionner un system domotique on fait appel à une centrale domotique ou un centre de control, la première fonction est assurée par des capteurs qui renvoient des informations afin de contrôler les paramètres du bâtiment et des systèmes. Cela peut être la présence ou non de personnes, ensuite les commandes vont permettre d'interagir avec l'installation, cela peut être des boutons poussoirs, interrupteurs,...etc. Ces commandes vont contrôler un ou plusieurs actionneurs qui sont les organes actifs qui vont agir sur les systèmes pour effectuer une tache précise sur les appareils concernés.

Pour un fonctionnement optimal, l'équipement du réseau peut être activé sous deux modes, automatique ou imprévu :

## CHAPITRE I.LA DOMOTIQUE ET LES SYSTEMES EMBARQUES

---

- **Le mode automatique** : repose sur un scénario domotique, les fonctions de sécurité, de gestion d'énergie et de confort seront automatiquement activées. La programmation répond effectivement aux besoins d'un logement de basse consommation, faire des économies d'énergie et pallier aux oublis quotidiens.
- **Le mode imprévu** : il se déclenche par la centrale domotique et permet de déclencher une action et envoyer un message sur l'interface de commande.

### I.2.6 Les modes de communication

Les supports de transmission les plus utilisés sont :

- les courants porteurs (CPL)
- les liaisons sans fil (radio ou infrarouge)
- les liaisons filaires à savoir :
  - le câble coaxial
  - la paire torsadée
  - la fibre optique

Dans ce qui suit nous allons définir les trois modes de communication composants une installation domotique.

#### Les courants porteurs en ligne(CPL)

Le principe de base du réseau CPL (Courant Porteur en Ligne) est d'utiliser les circuits de distribution électrique de la maison comme support physique pour véhiculer des données et des commandes. Ce réseau est actuellement le plus développé dans l'habitat.

La figure I.7 illustre une architecture d'une communication pour courant porteur en ligne.

On distingue généralement trois types de courant porteur :

**Courant porteur domestique** : cette technologie est conçue pour piloter l'éclairage, le chauffage, les automatismes, les prises de courant et la sécurité.

**Courant porteur informatique** : ce type de courant porteur en ligne autorise le transport des données informatiques, permettant ainsi de constituer un véritable réseau local reliant ordinateur, imprimantes, accès Internet, serveur multimédia, écran tactile, point d'accès WiFi.

**Courant porteur audiovisuel** : cette technologie permet de distribuer l'image et le son dans la maison (standard HomePlug AV).

**Avantage :**

- Les CPL ne nécessitent aucun câblage supplémentaire puisque la plus part des infrastructures résidentielles ont un vaste réseau électrique.
- Offre une grande souplesse d'utilisation puisque le system électronique qui y est connecté implique dans la plupart des cas une alimentation en énergie par le secteur.

## Désavantage :

- risque de manque de fiabilité en raison de la faible immunité aux parasites sur les lignes
- bus dédié davantage à la commande (manque de retour d'information)

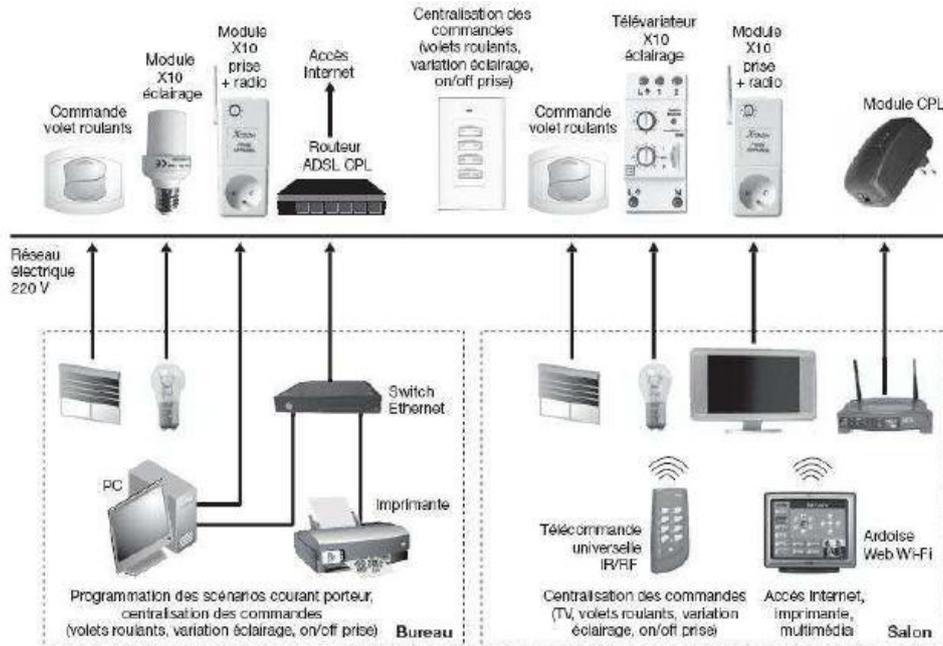


Figure I.7: communication courant porteur en ligne.

## Les réseaux filaires

Le bus filaire est constitué d'une ou plusieurs paires de conducteurs, permettant la circulation de données à haut débit. Il s'agit de la solution la plus fiable et la plus performante. La communication sans câbles ne se perturbe pas. Les deux types de bus de commande standards de cette catégorie sont les suivants :

**EIB (European Installation Bus)/Konnex :** Ce protocole a été retenu par plus de 150 constructeurs et s'implante peu à peu dans l'habitat résidentiel. Le bus EIB utilise une paire torsadée comme support de communication. La plus petite entité du système est appelée la ligne, elle comprend une alimentation qui fournit le 29V continu nécessaire au bon fonctionnement de l'ensemble. Chaque appareil dispose de son propre microprocesseur qui le rend entièrement paramétrable. Le système EIB est par conséquent très flexible et peut à tout moment être adapté aux nouvelles exigences.

**Lonworks :** LonWorks (réseau d'exploitation local) est une plate-forme de réseau spécialement conçue pour répondre aux besoins des applications de contrôle. La plate-forme est basée sur un protocole créé par Echelon Corporation pour les périphériques réseau sur des supports tels que la paire torsadée, les lignes électriques, les fibres optiques. Il est utilisé pour l'automatisation de diverses fonctions dans les projets d'immeubles de bureaux ou d'hôpitaux, tels que l'éclairage. LonWorks permet à tous les produits reliés au réseau de communiquer directement entre eux.

La figure I.8 présente une architecture d'une communication par bus filaire

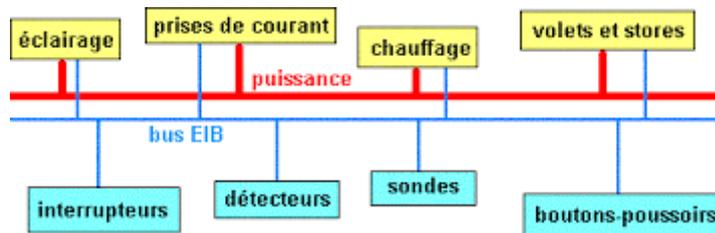


Figure I.8 : Communication pour bus filaire.

### Les réseaux sans fil

La communication sans fil permet la transmission des informations entre les composants d'une installation domotique sans liaison filaire.

Les technologies domotiques à câble sont souvent complétées par celles sans fil. Cela permet de mettre de câble sur l'installation qu'on va refaire plus tard, puis utiliser de la communication sans fil sur la partie de la maison non changeable.

L'avantage de la communication sans fil est qu'elle permet de faire évoluer une installation sans grand travaux.

Dans ce qui suit nous allons présenter deux exemples de la communication sans fil tel que :

**Zigbee:** est un protocole de haut niveau permettant la communication radio, à consommation réduite, conçu par **zigbee alliance** basé sur la norme IEEE 802.15.4 pour les réseaux à dimension personnelle (Wireless Personal Area Networks/WPAN).

Cette technologie a pour but de fournir les mêmes services que la technologie Bluetooth, tout en étant moins chère et plus simple, avec une portée maximale de 100 mètre environ, elle permet d'obtenir des débits pouvant atteindre 250Kb/s et une moindre consommation d'énergie.[10]

**Z-wave :** Z-Wave est un protocole de communication sans fil utilisé principalement pour la domotique. Il est orienté vers le marché du contrôle et de l'automatisation résidentielle et est destiné à fournir une méthode simple et fiable pour contrôler sans fil l'éclairage, les systèmes de sécurité, le home cinéma, les fenêtres automatisées, etc... Comme d'autres protocoles et systèmes destinés au marché de la maison et du bureau, un système d'automatisation Z-Wave peut être contrôlé via Internet, avec une passerelle Z-Wave ou un dispositif de contrôle central. Chaque module peut agir comme un fil qui étend la portée du système et assure que les commandes destinées à un autre dispositif dans le réseau sont reçues. Z-wave est optimisé pour une faible latence, avec des débits de données allant jusqu'à 100KB /s. La performance réelle dans une maison dépend du type de construction, la quantité de métal entre les dispositifs, et le nombre de dispositifs qui répètent les signaux sans fil.

La figure I.9 présente une infrastructure d'une communication sans fil.

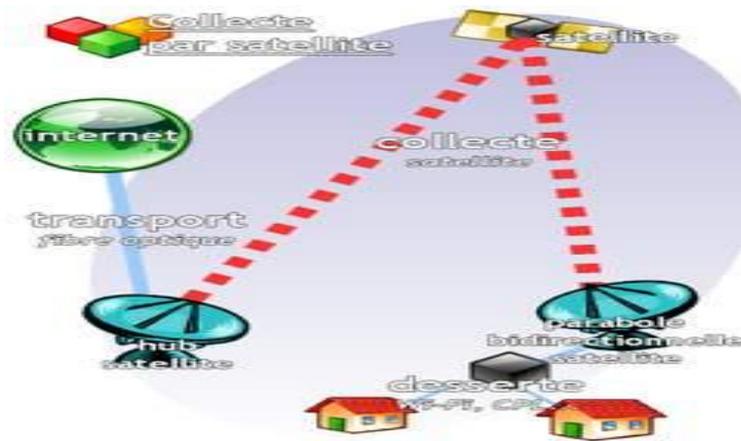


Figure I.9 communication sans fil.

## I.3 Systèmes embarqués

De nos jours presque tous les aspects de notre quotidien de vie emploient des composants électroniques. Dans tous les cas, ces composants sont des ordinateurs basés systèmes qui ne sont pas, cependant, utilisés ou perçus comme ordinateurs. Par exemple, ils n'ont pas souvent un clavier ou un écran pour interagir avec l'utilisateur et ils ne font pas tourner de manière classique les systèmes d'exploitation et les applications. Quelquefois, ces systèmes sont embarqués dans un autre système pour qu'ils fournissent de meilleures fonctionnalités et de performance (par exemple : un appareil de contrôle d'un moteur de véhicule). Ce sont des systèmes embarqués.

Dans le domaine de la domotique, on citera des systèmes embarqués qui sont intégrés à des équipements afin d'apporter une surveillance intelligente d'une maison, économiser l'énergie, contrôler le stock des aliments, etc.

### I.3.1. Système embarqué [9]

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.

Quelle que soit la nature et la complexité du système, on décompose un système embarqués en

- Le système contrôlé,
- Le système de contrôle.

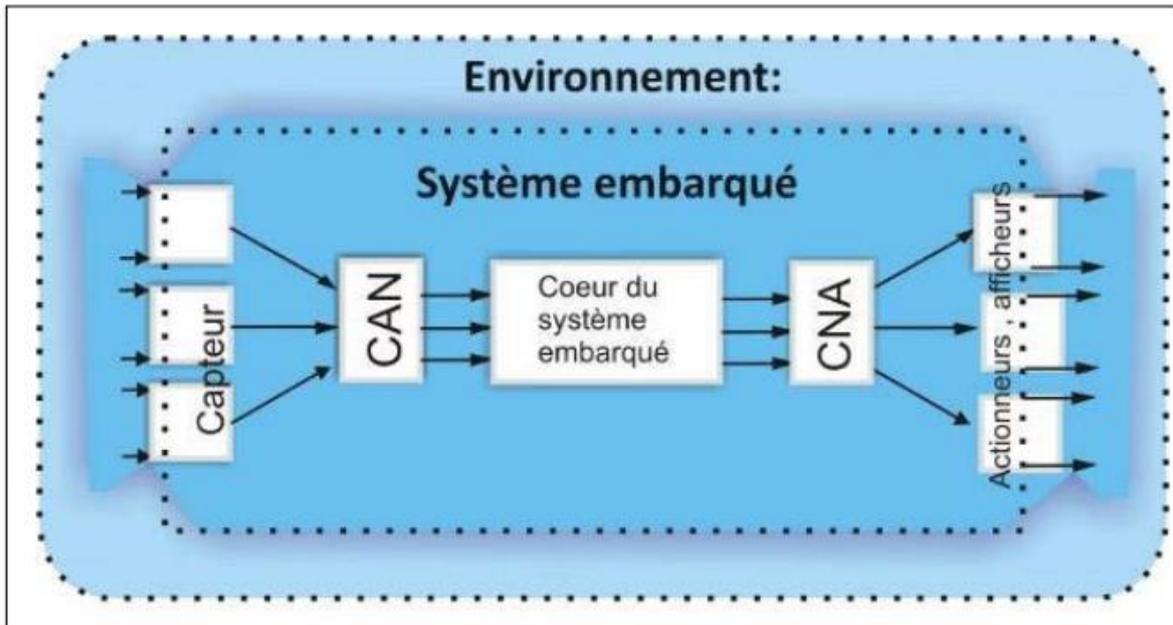
#### Le système contrôlé

Le système contrôlé est un environnement (procédé) équipé d'une instrumentation qui réalise l'interface avec le système de contrôle

#### Le système de contrôle

Le système de contrôle est éléments matériels (microprocesseurs...) et logiciels dont la mission est d'agir sur le procédé via les actionneurs en fonction de l'état de ce procédé indiqué par les capteurs de manière à maintenir ou conduire le procédé dans un état donné.

L'environnement du système embarque est résumé par la figure I.10.



**Figure I. 1: Système embarqué dans son environnement.**

Le cœur du système embarqué effectue des traitements sur des informations captées par les capteurs et converties auparavant par CAN (convertisseur analogique numérique), puis il envoie à son environnement via un actionneur une tâche relative à l'information captée qui doit être transformée par le CNA (convertisseur numérique analogique) avant d'être envoyée.

### **I.3.2. Historique [9]**

Les premiers systèmes embarqués sont apparus en 1971 avec l'apparition de l'Intel 4004. L'Intel 4004 développé 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, contrôle 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 quotidiens tels que fours à micro-ondes, télévisions et automobiles à moteur à injection électronique ne tardèrent pas à être équipés de microprocesseurs.

Ce sont alors les débuts de l'informatique embarquée.

### **I.3.3. Comparaison aux systèmes informatiques standards :**

Le tableau suivant donne une brève comparaison entre l'informatique standard et l'informatique embarqué.

# CHAPITRE I.LA DOMOTIQUE ET LES SYSTEMES EMBARQUES

<b>Informatique standard :</b>	<b>Informatique Embarqué :</b>
<ul style="list-style-type: none"><li>▪ Processeur standard<ul style="list-style-type: none"><li>• Multiples unités fonctionnelles</li><li>• Vitesse élevée (&gt; GHz)</li><li>• Consommation électrique élevée</li><li>• Chaleur</li><li>• Taille</li></ul></li><li>▪ MMU (mémoire virtuelle)</li><li>▪ OS</li><li>▪ Cache</li><li>▪ Grand nombre de périphériques</li></ul>	<ul style="list-style-type: none"><li>▪ Processeur dédié (contrôleur)<ul style="list-style-type: none"><li>• Architecture adaptée</li><li>• Vitesse faible (~200 MHz)</li><li>• 8-32bits : mémoire limitée</li><li>• Basse consommation</li></ul></li><li>▪ Quelques Mo de mémoire</li></ul>

**Tableau I.1. Comparaison entre l'informatique standard et l'informatique embarqué.**

Les cahiers des charges des systèmes embarqués comportent plusieurs contraintes ; on peut citer :

### **Fonctionnement en Temps réel :**

- Réactivité : des opérations de calcul doivent être faites en réponse à un événement extérieur (interruption matérielle).
- La validité d'un résultat dépend du moment où il est délivré (deadlines).
- Rater une échéance peut causer une erreur de fonctionnement.
- La plus part des systèmes sont « multi rate » car ils traitent des informations à différents rythmes.

### **Faible encombrement, poids et consommation :**

- Consommation électrique minimisée,
- Difficulté de packaging (analogique, numérique et RF),
- Batterie de 8 heures et plus (PC portable : 2heurs),
- Environnement hostile (température, vibration interférences RF, eau feu)

### **Coût, sureté et sécurité :**

- Le système doit toujours fonctionner correctement (faible coût).
- Sûreté de fonctionnement du logiciel .

### **I.3.5.Les domaines d'application des systèmes embarqués**

Les systèmes embarqués prennent de plus en plus une place primordiale dans divers domaines tels que :

## + La domotique

- Allumage du chauffage, fermeture et éteignement de volet.

## + Domaine grand public

- Smart phone, console de jeux, appareil photos, lecteur audio.

## + Moyens de transport

- Gestion moteur/entraînement, ordinateur de bord, GPS, système de navigation, système d'aide (EPS, ..).
- Automobiles, avions, trains, bateau, véhicule électrique.

## + Equipement médicaux (diagnostic, thérapeutique, vital)

- Imagerie (rayon X, ultra-sons, IRM), endoscopie, caméra, monitoring, perfusion, lasers, chirurgie, stimulateur cardiaque.

## + Equipements de télécommunication

- Station mobile, routeur, gateway, satellite.

## + Equipement industriels

- Commande, contrôle réparti, capteurs intelligents.

### I.3.6.Caractéristiques des systèmes embarqués.

Plutôt que des systèmes universels qui effectuant plusieurs tâches, les systèmes embarqués sont conçus pour effectuer des tâches précises. Certains doivent répondre à des contraintes de temps réel pour des raisons de fiabilité et de rentabilité. D'autres ayant peu de contraintes au niveau performances permettent de simplifier le système et de réduire les couts de fabrication.

Les systèmes embarqués ne sont pas toujours des modules indépendants. Le plus souvent ils sont intégrés dans le dispositif qu'ils contrôlent.

Le logiciel créé pour les systèmes embarqués est appelé firmware. Il est stocké dans la mémoire en lecture seule ou de la mémoire flash plutôt que dans un disque dur. Il fonctionne le plus souvent avec des ressources matérielles limitées : écran et clavier de tailles réduites, voire absents, peu de mémoire, capacités de calcul relativement faibles.

### I.3.7.Classification des systèmes embarqués :

Les systèmes embarqués peuvent être repartis en plusieurs classes de systèmes tels que :

#### **Système Transformationnel :**

Le système transformationnel se distingue pas son activité de calcul : il lit ses données et ses entrées lors de son démarrage, fournit ses sorties, puis meurt.

## Système Interactif :

Le système interactif est un système en interaction quasi permanente avec son environnement, y compris après l'initialisation du système; la réaction du système est déterminée par les événements reçus et par l'état courant (fonction des événements et des réactions passés); le rythme de l'interaction est déterminé par le système et non par l'environnement.

## Système Réactif ou Temps Réel :

Le système réactif est un système en interaction permanente avec son environnement, y compris après l'initialisation du système; la réaction du système est déterminée par les événements reçus et par l'état courant (fonction des événements et des réactions passés); mais le rythme de l'interaction est déterminé par l'environnement et non par le système.

### I.3.8.Architecture d'un système embarqué :

Les systèmes embarqués sont caractérisés par une architecture qui varie selon les systèmes, une de ces architectures est présentée par la figure I.11

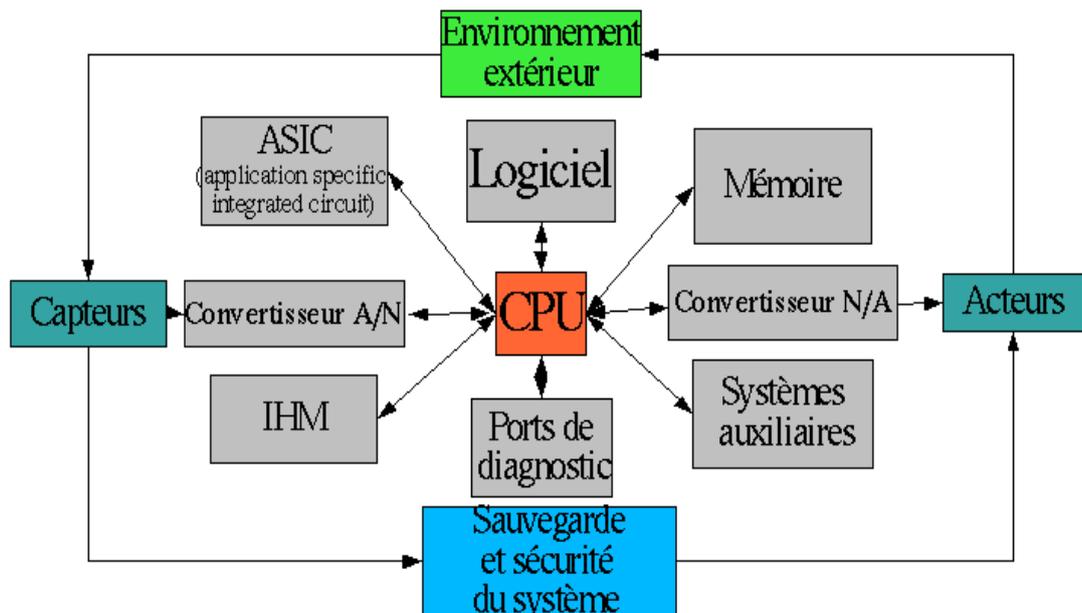


Figure I. 2:Architecture d'un système embarqué.

Nous allons décrire quelques composants apparaissant sur la figure ; d'autres tels que les capteurs et les actionneurs seront étudiés dans le chapitre suivant

**IHM** : Interface Homme Machine, définissent les moyens et outils mis en œuvre afin qu'un humain puisse contrôler et communiquer avec une machine.

**Convertisseur A/N** : convertisseur analogique numérique, Il transforme la tension de l'échantillon (analogique) en un code binaire (numérique).

**Convertisseur N/A** : convertisseur numérique analogique, Il effectue l'opération inverse du CAN, il assure le passage du numérique vers l'analogique en restituant une tension proportionnelle au code numérique.

## CHAPITRE I.LA DOMOTIQUE ET LES SYSTEMES EMBARQUES

---

**CPU** : De l'acronyme Central processing unit, désigne l'unité de traitement ou microprocesseur principal d'un ordinateur. Il est soumis à divers paramètres tels que la fréquence, la mémoire cache ; il est Chargé de l'exécution des instructions des programmes.

**ASIC** : Application Specific Integrated Circuit (un circuit intégré propre à une application) ; un **ASIC** est un type de circuit électronique particulier. Il s'agit d'un circuit qui intègre, sur une même puce, l'ensemble des éléments actifs indispensables pour qu'une fonction ou qu'un ensemble électronique puisse se réaliser. Autrement dit, l'**ASIC** est un circuit intégré exclusivement dédié à une application et à un utilisateur.

Le fonctionnement d'un système embarqué est défini sous forme d'opérations chainées qui est :

- ✓ Il reçoit des informations de l'environnement extérieur qu'il converti en signal numérique ;
- ✓ L'unité de traitement traite en suite les informations reçues ;
- ✓ Le traitement génère éventuellement une sortie qui est envoyée vers la sortie, les systèmes auxiliaires, les ports de monitoring ou l'IHM.

### **I.3.8.Conclusion**

Au cours de ce chapitre nous avons présenté des brèves définitions des concepts de la domotique et des systèmes embarqués. En effet on a subdivisé ce chapitre en deux parties. La première partie est dédiée à la domotique, ses domaines d'application, ses composants ainsi son mode de fonctionnement.

Pour la deuxième partie nous avons présenté les systèmes embarqués, leurs domaines d'applications et leur architecture.

Dans le chapitre qui suit nous allons découvrir le dispositif programmable Arduino, qui permet de mettre en œuvre des systèmes embarqués dans différents domaines notamment dans le domaine de la domotique.

### II.1 Introduction

L'électronique qui depuis son existence ne cesse de croître, elle continue de plus en plus en lui intégrant des circuits programmables.

En effet, ces derniers sont des éléments incontournables qui ont l'habileté d'unir la puissance de la programmation à la puissance de l'électronique.

Et vu que les cartes Arduino sont un puissant outil de prototypage pour les cartes électroniques, et elles permettent un accès facile et intuitif à l'informatique embarqué ; on se focalisera dans ce chapitre sur la technologie Arduino où nous allons faire une étude générale sur la carte Arduino en détaillant la carte Arduino UNO, ainsi que son IDE et les étapes de téléversement d'un programme sur cette carte.

### II.2 La Carte Arduino

La carte Arduino est une carte sur laquelle se trouve un microcontrôleur qui peut être programmé pour analyser et produire des signaux électrique, de manière à effectuer des tâches très diverses comme la domotique, la télésurveillance ...etc.

La figure II.1.présente la carte Arduino UNO. [1]



Figure II. 1.Carte Arduino UNO

### II.3 Domaines d'utilisation

Les cartes Arduino UNO peuvent être utilisées dans différents domaines parmi lesquelles on peut citer :

- **Physical computing** : Au sens large construire des systèmes physiques interactifs qui utilisent des logiciels et du matériel pouvant s'interfacer avec des capteurs et de actionneurs ;
- **capter et analyser de données scientifiques** (environnement, énergie, etc.) à des fins éducatives, de recherche ou d'appropriation citoyenne ;
- **spectacle vivant**, grâce aux nombreuses fonctions d'interaction offertes par Arduino, il est possible de créer des performances de VJing, utiliser le mouvement des danseurs pour générer en temps réel des effets sonores et visuels dans un spectacle ;

- **projets pédagogiques** à destination d'étudiants, du grand public selon les porteurs de ces initiatives : écoles supérieures, centres de formation spécialisée ou des Media Labs.

### II.4 les différentes cartes Arduino [4]

Les cartes ARDUINO doivent satisfaire des exigences diverses et variées, certains utilisateurs souhaiteront effectuer du prototypage et de tester de nouveaux montages ou idées, pour cela différentes cartes sont mises en œuvre. Dans le tableau II.1 on se focalisera sur quelques types et leurs caractéristiques.

À partir du tableau II.1, on remarque que toutes les caractéristiques sont différentes d'une carte à une autre. Pour cela de nombreux paramètres sont à prendre en compte pour bien choisir la carte à utiliser pour son projet, à savoir :

- **dimension** : la dimension est l'un des paramètres qui doit être pris en considération, car plus c'est petit, plus c'est facile à faire rentrer.
- **Le nombre d'entrées sorties** : le nombre d'entrées sorties est important en effet si on a besoins de 50 entrées sorties il vaut mieux prendre celle qui en a suffisamment.
- **La taille mémoire** : Si on a un programme très lourd mieux vaut qu'il y ait suffisamment de place dans la carte.
- **La fréquence d'horloge** : est aussi un critère très important à prendre en considération.
- **Le prix** : éventuellement est une contrainte qui reste aussi à voire et à prendre en compte.

Nom	Carte Arduino Mini	Carte Arduino Due	Carte Arduino Mega 2560	Carte Arduino UNO
<b>Microcontrôleur</b>	ATmega 328	AT91SA3X8E	ATmega2560	ATmega328
<b>SRAM</b>	2KO	96KO	8KO	2KO
<b>EEPROM</b>	1KO	/	4KO	1KO
<b>Mémoire flash</b>	32KO	512KO	256KO	32KO
<b>Entrées sorties numériques</b>	14 E/S	54 E/S	54 E/S	14 E/S
<b>Entrées Analogiques</b>	8	12	16	6
<b>Dimension</b>	30 mm*18mm	101mm*53mm	101mm*53mm	68mm*53mm
<b>Tension d'alimentation</b>	7-9 Voltes	7-12 Voltes	7-12 Voltes	7-12 voltes
<b>Fréquence d'horloge</b>	16MHz	84MHz	16MHz	16MHz
<b>Figure</b>				

Table II.1.Caractéristiques de quelque carte Arduino.

### II.5 La carte Arduino UNO [5]

Un des modèles les plus répandu de la carte Arduino est l'Arduino UNO. C'est la première version stable de carte Arduino. Elle possède toutes les fonctionnalités d'un microcontrôleur classique en plus de sa simplicité d'utilisation ; elle utilise une puce ATmega328P cadencée à **16Mhz**. Elle possède **32ko** de **mémoire flash** destinée à recevoir le programme, **2ko** de **SRAM** (mémoire vive) et **1 ko d'EEPROM** (mémoire morte destinée aux données). Elle offre **14** broches d'entrée/sortie numériques dont **6** pouvant générer des signaux **PWM** (Pulse Width Modulation). Elle permet aussi de mesurer des grandeurs analogiques grâce à ces **6 entrées analogiques**. Chaque broche est capable de délivrer un courant de **40mA** pour une tension de **5V**.

Cette carte Arduino peut aussi s'alimenter et communiquer avec un ordinateur grâce à son port USB. On peut aussi l'alimenter avec une alimentation comprise en 7V et 12V grâce à son connecteur Power Jack.

Dans notre projet nous avons opté à choisir et à utiliser la carte Arduino UNO par le fait qu'elle est peu coûteuse comparativement aux autres.

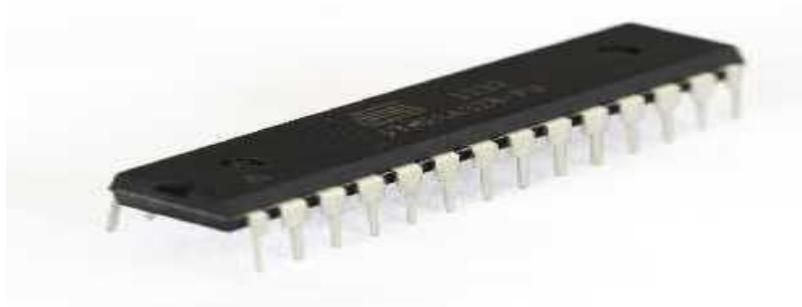
### II.5.1 La partie matérielle :

La carte Arduino UNO est constituée de :

#### II.5.1.1 Un microcontrôleur :

C'est le cerveau de notre carte, C'est lui qui va recevoir le programme créé et qui va le stocker dans sa mémoire puis l'exécuter. Grâce à ce programme, il va savoir faire des choses, qui peuvent être : faire clignoter une LED, afficher des caractères sur un écran, envoyer des données à un ordinateur.

La figure II.2 présente un microcontrôleur Atmega328p.



**Figure II. 2. Microcontrôleur Atmega 328p**

Le microcontrôleur est constitué par un ensemble d'éléments qui ont chacun une fonction bien déterminée. Il est en fait constitué des mêmes éléments que sur la carte mère d'un ordinateur. Globalement, l'architecture interne de ce circuit programmable se compose essentiellement de :

**Un processeur** : il permet l'exécution des instructions d'un programme.

**La mémoire Flash**: C'est celle qui contiendra le programme à exécuter. Cette mémoire est effaçable; sa capacité est de 32Ko.

**RAM** : c'est la mémoire dite "vive", elle va contenir les variables du programme. Elle est dite "volatile" car elle s'efface si on coupe l'alimentation du microcontrôleur. Sa capacité est 2 ko.

**Une mémoire morte (EEPROM)** : C'est le disque dur du microcontrôleur. On y enregistre des informations qui ont besoin de survivre dans le temps, même si la carte doit être arrêtée. Cette mémoire ne s'efface pas lorsque l'on éteint le microcontrôleur ou lorsqu'on le reprogramme. sa capacité est de 1ko.

**Une interface d'entrées/sorties :** elle permet la communication du microcontrôleur avec d'autres composants externes.

**Bus d'adresse :** permettant l'accès aux différentes cases mémoires.

**Bus de contrôle :** permettant de se positionner en lecture ou en écriture sur ces différentes cases mémoires.

**Bus de données :** il est conçu pour la transition des données de la mémoire vers le processeur.

### II.5.1.2 Une alimentation [5]

Pour fonctionner, la carte a besoin d'une alimentation. Le microcontrôleur fonctionnant sous 5V, la carte peut être alimentée en 5V par le port USB, ou bien par une alimentation externe qui est comprise entre 7V et 12V. Cette tension doit être continue et peut par exemple être fournie par une pile de 9V. Un régulateur se charge ensuite de réduire la tension à 5V pour le bon fonctionnement de la carte.

Les broches d'alimentation sont les suivantes :

- La broche VIN. La tension d'entrée positive lorsque la carte Arduino est utilisée avec une source de tension externe (à distinguer du 5V de la connexion USB ou autre source 5V régulée). On peut alimenter la carte à l'aide de cette broche, ou, si l'alimentation est fournie par le jack d'alimentation, accéder à la tension d'alimentation sur cette broche.
- La broche de 5V. La tension régulée utilisée pour faire fonctionner le microcontrôleur et les autres composants de la carte. Le 5V régulé fourni par cette broche peut donc provenir soit de la tension d'alimentation VIN via le régulateur de la carte, ou bien de la connexion USB (qui fournit du 5V régulé) ou de tout autre source d'alimentation régulée.
- La broche de 3V3. Une alimentation de 3.3V fournie par le circuit intégré FTDI (circuit intégré faisant l'adaptation du signal entre le port USB de votre ordinateur et le port série de l'ATmega) de la carte est disponible : ceci est intéressant pour certains circuits externes nécessitant cette tension au lieu du 5V. L'intensité maximale disponible sur cette broche est de 50mA
- La broche de la GND. Broche de masse (ou 0V).

### II.5.1.3 Visualisation

La carte Arduino contient 3 LEDs dont la taille est de l'ordre du millimètre. Ces LEDs servent à deux choses :

- Une est connectée à une broche du microcontrôleur et va servir pour tester le matériel, notamment quand on branche la carte au PC, elle clignote quelques secondes.
- Les deux autres LEDs: servent à visualiser l'activité sur la voie série (une pour l'émission et l'autre pour la réception). Le téléchargement du programme dans le microcontrôleur se fait par cette voie, on peut les voir clignoter lors du chargement.

### II.5.1.4 La connectique

La carte Arduino ne possédant pas de composants qui peuvent être utilisés pour un programme, mis à part la LED connectée à la broche 13 du microcontrôleur, il est nécessaire de les rajouter. Mais pour ce faire, il faut les connecter à la carte. C'est là qu'intervient la connectique de la carte.

Cette carte possède 14 broches numériques (numérotée de 0 à 13) dont chacune peut être utilisée soit comme une entrée numérique, soit comme une sortie numérique, en utilisant les instructions

`pinMode()`, `digitalWrite()` et `digitalRead()` du langage Arduino. Ces broches fonctionnent en 5V. Chaque broche peut fournir ou recevoir un maximum de 40mA d'intensité et dispose d'une résistance interne (déconnectée par défaut) de 20-50 KOhms. Cette résistance interne s'active sur une broche en entrée à l'aide de l'instruction `digitalWrite(broche, HIGH)`.

De plus certaines broches ont des fonctions spécialisées telle que :

- **Interruptions Externes:** Broches 2 et 3 ; Ces broches peuvent être configurées pour déclencher une interruption sur une valeur basse, sur un front montant ou descendant, ou sur un changement de valeur.
- **PMW:** Broches 3, 5, 6, 9, 10, et 11. Fournissent une impulsion PWM 8-bits à l'aide de l'instruction `analogWrite()`.
- **SPI:** (Interface Série Périphérique): Broches 10, 11, 12, 13. Ces broches supportent la communication SPI disponible avec la bibliothèque pour communication SPI. C'est un bus de données série synchrone, qui opère au mode Full-duplex.
- **I2C:** Broches 4 et 5 supportent les communications de protocole I2C disponible en utilisant la bibliothèque `Wire/I2C` (ou `TWI - Two-Wire interface - interface "2 fils"`), un bus informatique qui permet de relier facilement un microprocesseur et différents circuits
- **UART(Universal Asynchronous Receiver/Transmitter)** :est un composant clé qui fait la liaison entre l'ordinateur et le port série il véhicule les octets de données et transmet les bites individuel de manière séquentielle .

- **les broches analogiques**

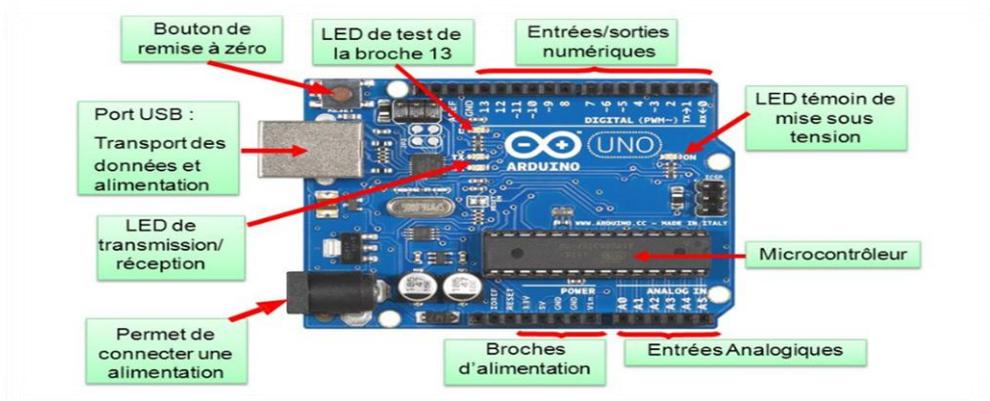
La carte Arduino Uno contient 6 entrées analogiques (numérotées de 0 à 5), chacune pouvant fournir une mesure d'une résolution de 10 bits (càd sur 1024 niveaux soit de 0 à 1023) à l'aide de la très utile fonction `analogRead()` du langage Arduino. Par défaut, ces broches mesurent entre le 0V (valeur 0) et le 5V (valeur 1023), mais il est possible de modifier la référence supérieure de la plage de mesure en utilisant la broche AREF et l'instruction `analogReference()` du langage Arduino.

- **Autres broches**

Il y a deux autres broches disponibles sur la carte :

- **AREF** : Tension de référence pour les entrées analogiques (si différent du 5V). Utilisée avec l'instruction `analogReference()`.
- **Reset** : Mettre cette broche au niveau BAS entraîne la réinitialisation du microcontrôleur. Typiquement, cette broche est utilisée pour ajouter un bouton de réinitialisation sur le circuit qui bloque celui présent sur la carte.

La figure II.4 présente les éléments de la carte Arduino UNO.



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

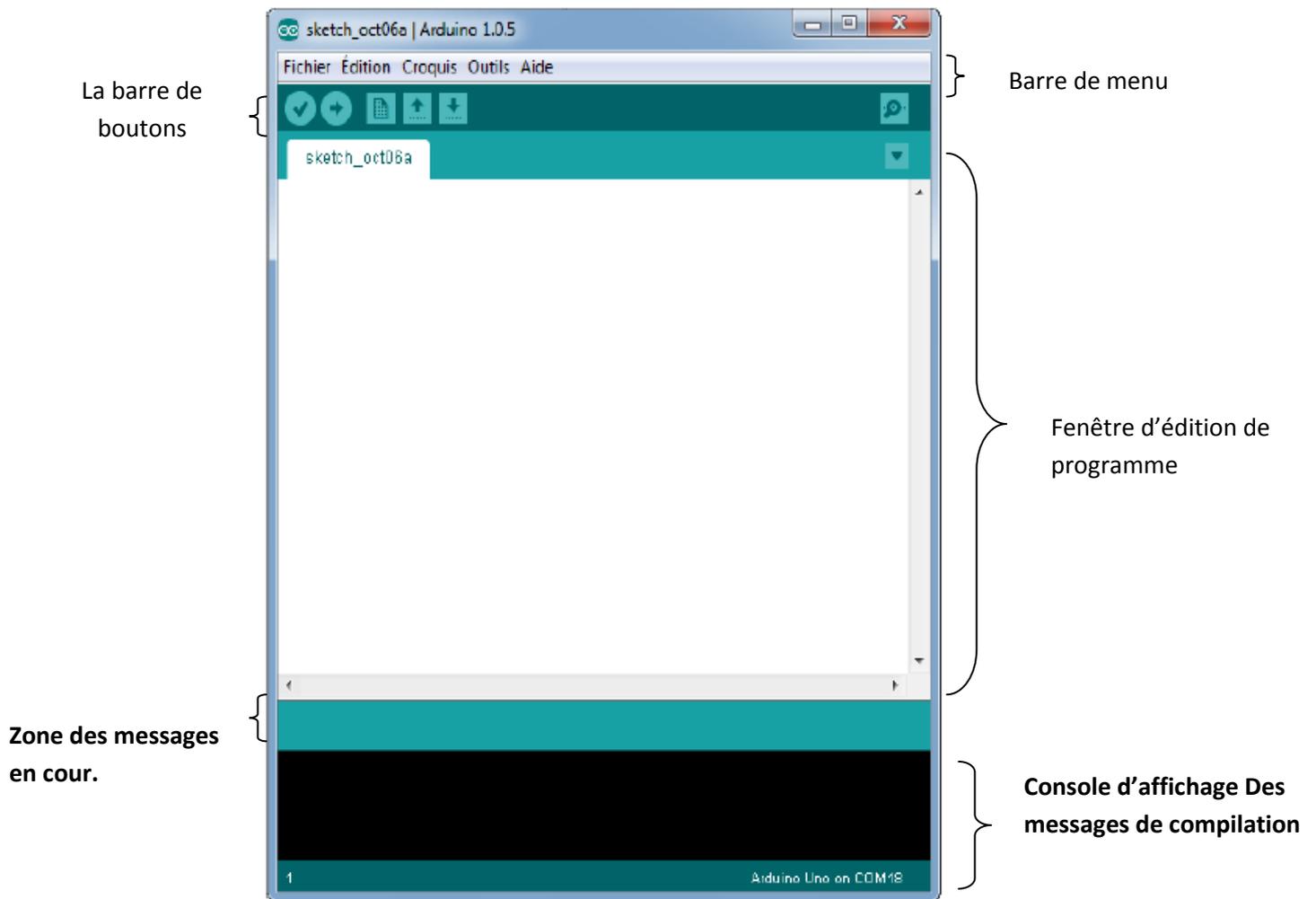
### II.5.2 la partie logicielle :

Le logiciel de programmation de la carte Arduino est un environnement de développement intégré fonctionnant sur divers systèmes d'exploitation (*Windows, Mac OS, Gnu/Linux*) qui permet d'éditer le programme sur un ordinateur et de le transférer via le port USB.

#### II.5.2.1 l'environnement de la programmation

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

#### II.5.2.2 Structure générale de l'interface de programmation



**Figure II. 4.Interface IDE Arduino**

### II.5.2.3 Structure générale d'un programme Arduino

Un programme Arduino contient trois parties :

- ✚ La partie déclaration des variables (optionnelle)

```
Int brocheCapteur=A0 ; //sélection de la broche sur laquelle est connecte le capteur  
Int brocheLED=4 ; //sélection de la broche sur laquelle est connectée la LED
```

- ✚ La partie initialisation et configuration des entrées/sorties : la fonction **setup ()**

**La fonction setup ()** est la première à être exécutée dans le programme ainsi elle sera exécutée une seule fois.

```
1. void setup ()  
2. {  
3. pinMode (ledpin, output) ; //broche la LED configurée en sortie  
4. }
```

✚ La partie principale qui s'exécute en boucle : la fonction **loop ()**

**La fonction loop ()** :c'est la où on va écrire le contenu du programme. Elle est appelée en permanence, à chaque fin d'exécution on le ré-exécute encore ; on parle de boucle infinie.

```
1. void loop ()  
2. {  
3. digitalWrite(brocheLED,HIGH) ; //met la broche en ON  
4. delay(1000) ; //une pause pendant 1 seconde  
5. digitalWrite(brocheLED,LOW) ; //met la broche en OFF  
6. delay(3000) ; //une pause pendant 3 seconde  
7. }
```

### II.5.2.4 Téléverser un programme dans la carte

En premier lieu il faut d'abord broncher notre carte Arduino avec câble USB ; puis une fois que la carte est reconnu par l'ordinateur, il faut sélectionner le port et le type adéquat, comme le montre la figure II.6.et II.7

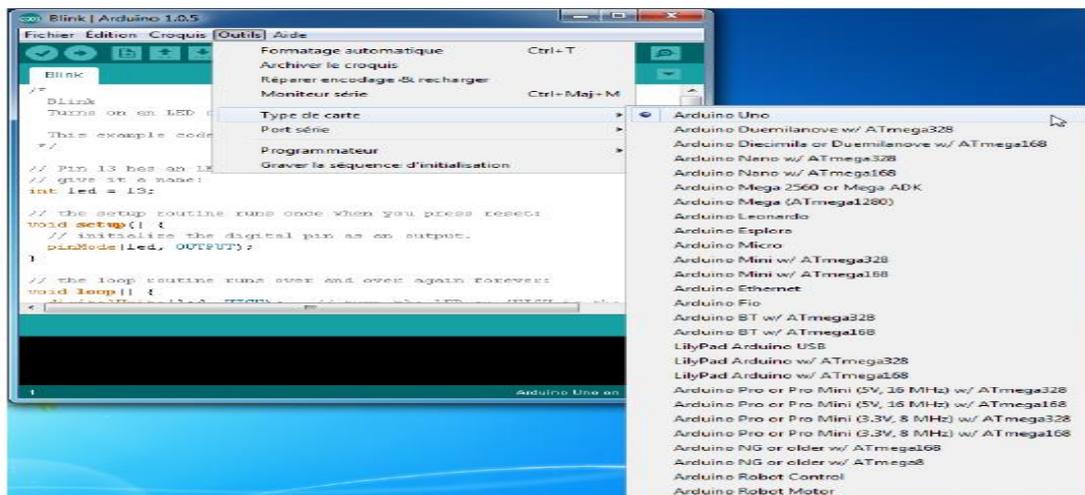
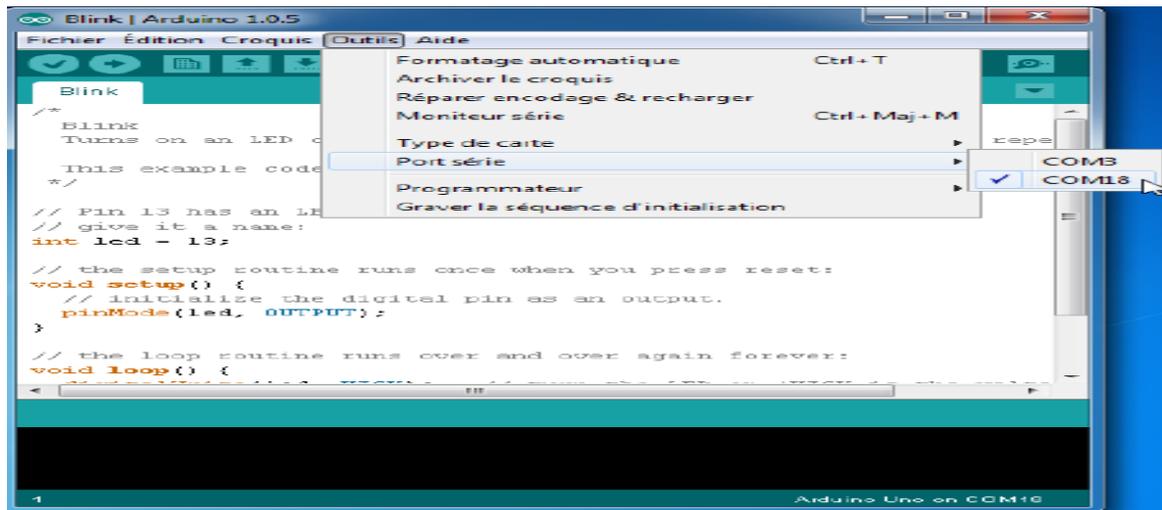


Figure II. 6. Comment choisir et le type de la carte



**Figure II. 7. Comment choisir le port série**

En fin une courte opération enchainée doit être suivie afin d'injecter un code qui fonctionne d'une manière cohérente sur la carte Arduino.

1. D'abord, On conçoit ou on ouvre un programme existant avec le logiciel IDE Arduino ;
2. On vérifie ce programme avec le logiciel Arduino (compilation) ;
3. Si des erreurs sont signalées, on modifie le programme ;
4. On charge le programme sur la carte ;
5. L'exécution du programme est automatique après quelques secondes ;
6. On vérifie que notre montage fonctionne.

En effet, au départ le programme est sous forme de texte saisi au niveau de l'IDE Arduino, puis il est transformé en un langage machine composé de 0 et 1 ; le programme passe sous cette forme dans le câble USB qui relie l'ordinateur à notre carte

En suite, à la réception du programme ce dernier va subir une petite transformation qui permet d'adapter le signal électrique correspondant au programme vers un signal plus approprié pour le microcontrôleur. On passe ainsi d'un signal codé pour la norme USB à un signal codé pour une simple voie série Puis ce "nouveau" signal est alors intercepté par le microcontrôleur.

La figure II.8 présente les étapes de compilation et le téléversement dans la carte

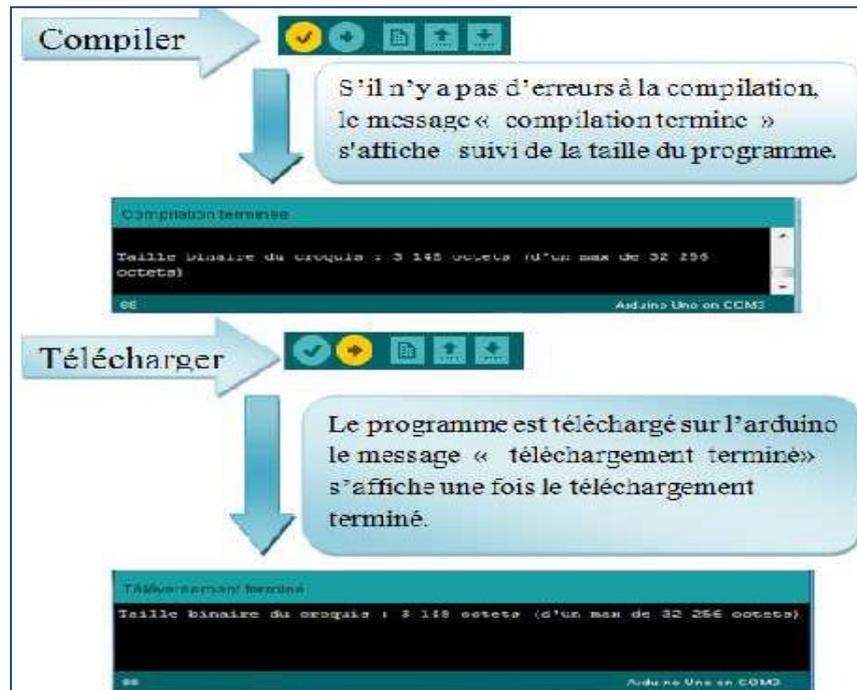


Figure II.8.Compilation et chargement du code Arduino.

## II.6 Les Shields de la carte Arduino

Pour la plupart des projets, il est souvent nécessaire d'ajouter des **fonctionnalités** aux cartes Arduino. En ajoutant des composants extérieurs tels que les **shields**.

Un Shield est une carte que l'on connecte directement sur la carte Arduino qui a pour but d'ajouter des composants sur la carte. Ces Shields viennent généralement avec une librairie permettant de les contrôler. On retrouve par exemple, des Shield Ethernet, de contrôle de moteur, lecteur de carte SD ; Le principal avantage de ces Shields est leurs simplicités d'utilisation. Il suffit des les emboîter sur la carte Arduino pour les connecter. [6]

Dans ce qui suit nous allons voir quelques exemples de ces Shields.

### II.6.1.1 Le module Arduino Bluetooth

Le module Arduino Bluetooth permet la communication avec la carte Arduino.il sera alors possible de communiquer avec Arduino en utilisant un pc ou tous autres équipements disposant de la norme Bluetooth.

La figure II.9 présente deux modules Bluetooth.



**Figure II. 9. Les modules Bluetooth**

### **II.6.1.2 Le module Shield Arduino Wifi**

Le module Shield Arduino Wifi permet de connecter une carte Arduino à un réseau internet sans fil Wifi.

La figure II.10 présente un exemple de Shield Wifi.



**Figure II. 10.Le Shield Wifi**

### **II.6.1.3 le module XBee:**

Le XBee est un microcontrôleur sans fil qui utilise un émetteur-récepteur sans fil 2,4 GHz pour communiquer avec un autre module XBee. Ces modules sont capables de communiquer avec plus d'un module XBee.

La figure II.11 présente le module XBee.





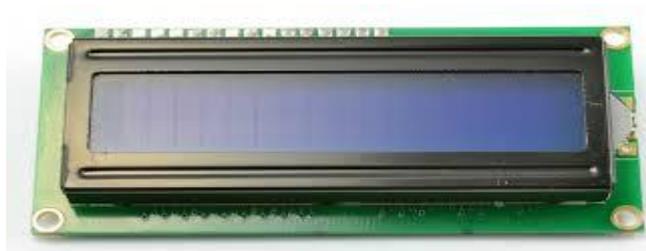
**Figure II.13. Les moteurs électriques**

### ➤ Les afficheurs LCD

Les afficheurs LCD sont devenus incontournables dans les systèmes techniques qui nécessitent l'affichage des paramètres de fonctionnement.

En effet ils permettent d'afficher des lettres, des chiffres et quelques caractères spéciaux.

La figure II.14. Illustre un exemple d'afficheur LCD.



**Figure II.14. Afficheur LCD**

Malgré que Plusieurs afficheurs sont disponibles sur le marché qui diffèrent les uns des autres, par leurs dimensions, leurs caractéristiques technique et leur tension de service mais les afficheurs LCD restent les plus employés.

### **Le relais**

Un relais est un organe électrique permettant la commutation de liaisons électriques. Il est chargé de transmettre un ordre de la partie commande à la partie puissance d'un appareil électrique. Ils sont utilisés très régulièrement, par exemple pour démarrer un moteur, commander une lumière...etc.

La figure II.15 présente un exemple de relais



**Figure II.15.Le relais**

### **II.7.La différence entre les modules de communication (wifi, Bluetooth et Xbee)**

Nous voudrions terminer ce chapitre par une brève étude comparative des trois principales technologies de communication à savoir (wifi, Bluetooth, Xbee)

#### **WIFI**

Taux de transmission de données élevé (54 Mbit / s), mais aussi une forte consommation d'énergie. Utilisé lorsque vous devez vous connecter directement à Internet, tel qu'un périphérique, et avoir une source d'alimentation externe.

#### **XBee:**

Taux de transmission de données faible (250kbit / s) et faible consommation d'énergie et faible prix. Utilisé pour créer des réseaux de capteurs de type maillage; Chaque périphérique XBee peut communiquer les uns avec les autres et l'un avec l'autre via le maillage à des périphériques hors limites. Utilisé pour les réseaux de périphériques, la domotique particulière et les capteurs.

#### **Bluetooth:**

Taux moyen de données et consommation moyenne. La norme IEEE comporte différents profils de périphériques pour permettre l'interopérabilité entre les périphériques. Il n'est pas utile pour les réseaux de capteurs, mais bon pour contrôler les appareils à l'aide d'un ordinateur portable ou d'un téléphone qui a habituellement Bluetooth à bord. Communique avec Arduino via les broches série RX et TX

### **II.8-Conclusion**

Ce chapitre étant le point de départ pour l'élaboration de notre projet, dans la mesure où il décrivait son contexte général, en présentant la carte Arduino pour laquelle nous avons expliqué ses parties logicielle et matérielle. Dans le prochain chapitre nous allons voir les étapes et les détails de la conception de notre application.

### III.1 Introduction

On assiste ces derniers temps à une inquiétude majeure des parents sur la sécurité de leurs nourrissons ; en effet l'angoisse de la mort subite des nourrissons est tous à fait compréhensible.

On remarque aussi cette inquiétude qui s'accroît sur des malades qui ne peuvent pas se prendre en charge eux-mêmes et qui nécessitent une assistance permanente.

Vu que les chambres des malades et des nourrissons peuvent être négligées, il s'avère indispensable de fournir un système de détection de la température ainsi que le mouvement dans la chambre .et d'informer le garde malade ou la nourrisse pour qu'ils puissent intervenir dans l'immédiat.

Dans ce chapitre nous allons présenter d'une manière détaillée la solution adoptée pour réaliser un système de surveillance la chambre du nourrisson ou du malade et cela en abordant la conception de chaque partie du système.

### III.2 Structure générale du système

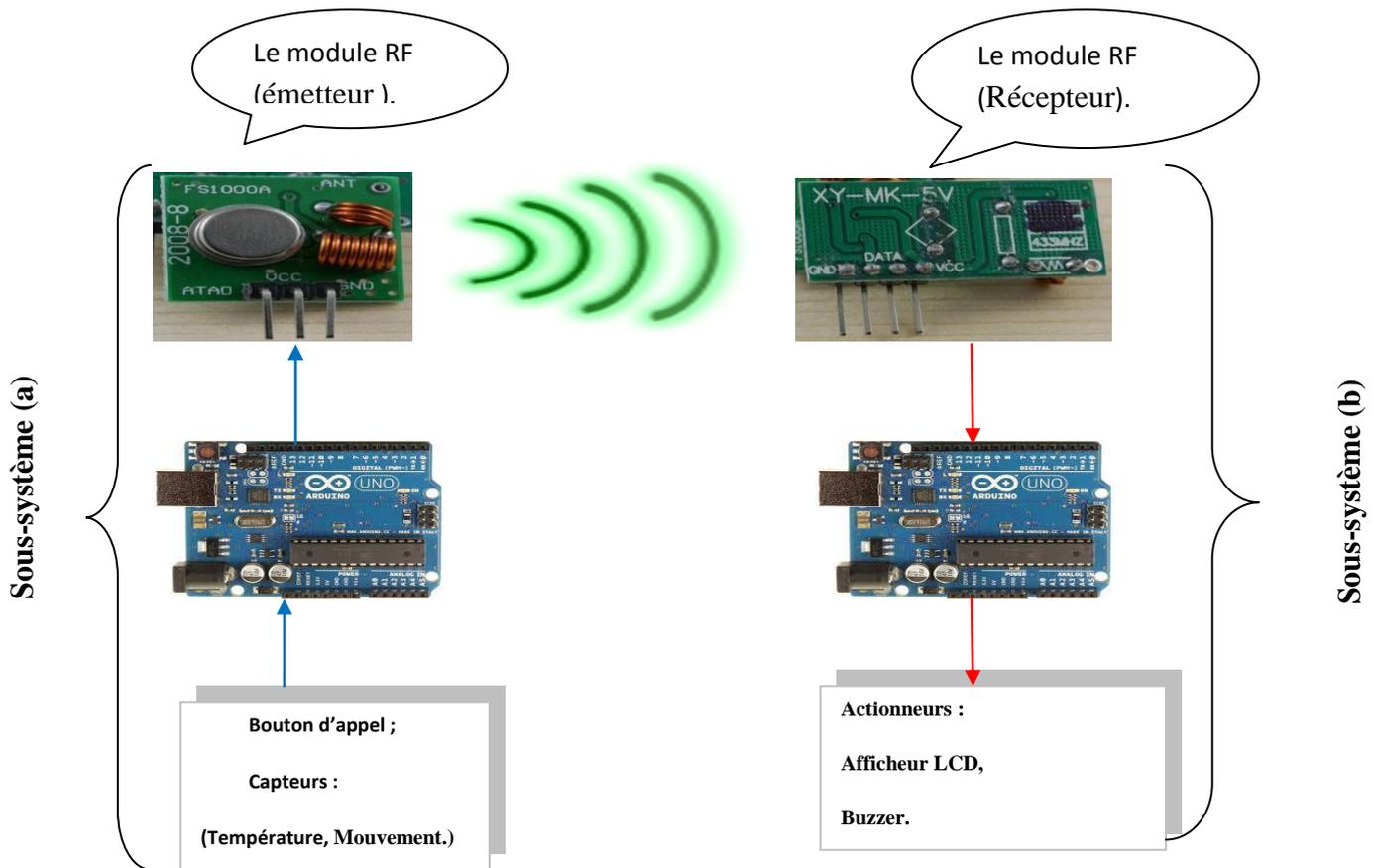
Notre système est composé de deux sous-systèmes « sous-système(a) »et« sous-système(b)» qui se communiquent avec des modules RF(radiofréquence) ;deux cartes Arduino UNO ainsi que d'autres composants tels que les capteurs et les actionneurs.

Le sous-système (a) a pour but de capter les données physiques présentées dans les chambres de malades ou de nourrissons et cela par des capteurs (capteur de mouvement et capteur de température) et l'appuie d'un bouton par un malade.

Une fois ces données sont captées ils seront traités par le microcontrôleur présent dans la carte Arduino UNO.

Si les seuils maximaux des données captées sont dépassés, le microcontrôleur exécutera un ensemble d'actions qui seront envoyées par le module RF au sous système(b)

La figure III.1 présente la structure générale de notre système.



**Figure III.1.structure de notre système.**

### III.3 Conception matérielle

Notre système nécessite un ensemble de dispositifs tel que les deux modules RF et deux cartes Arduino UNO que nous avons détaillé précédemment dans le chapitre II, deux types de capteurs (capteur de mouvement et le capteur de température), des modules de communication RF, une alarme et un écran LCD.

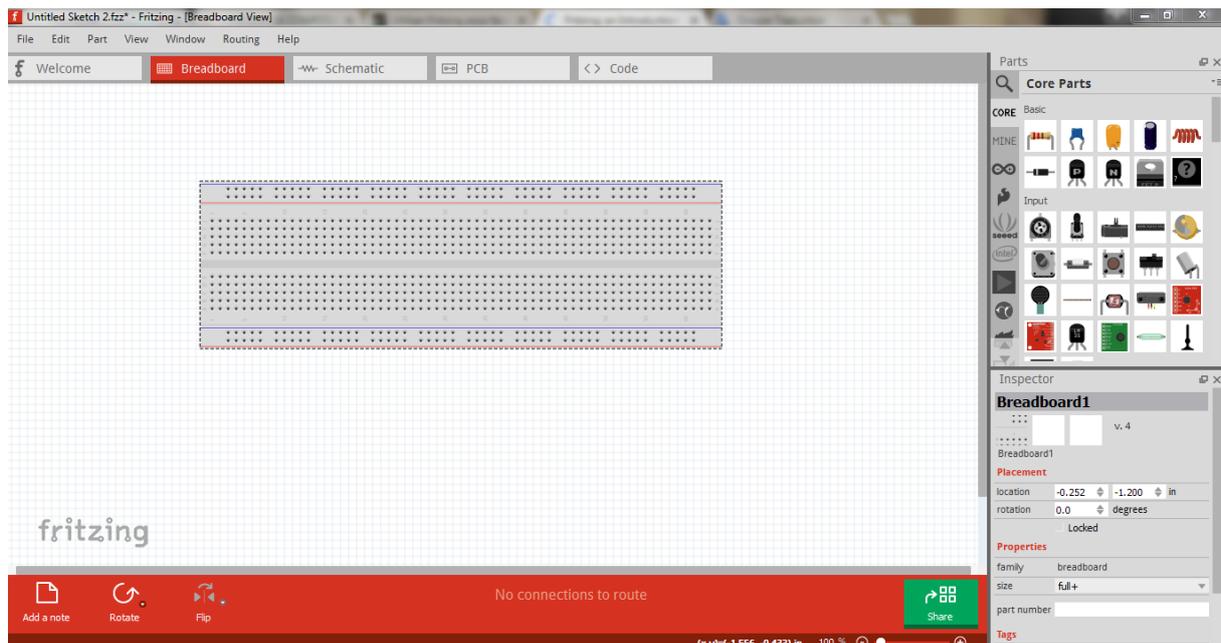
Dans cette partie nous allons donner un bref aperçu de ces dispositifs dont on aura besoin ainsi que l'outil utilisé pour le câblage de ces dispositifs.

#### III.3.1 Fritzing.

Lors de la création d'un projet ou d'un circuit on doit généralement extraire un schéma relatif au câblage du circuit et cela avant tout commencement.

Et à fin de pouvoir élaborer ces schéma on a opté pour l'utilisation du programme connu sous le nom Fritzing

Fritzing est un logiciel open source, qui a été développé à l'université des sciences appliquées de Potsdam, en Allemagne, permet de concevoir des circuits, de créer un schéma électrique et même de faire fabriquer un circuit imprimé à partir des informations donnés. [2]



**Figure III.2. le logiciel Fritzing.**

### III.3.2 Capteurs [12]

Un capteur est un petit dispositif électronique capable de mesurer une valeur physique environnementale (température, mouvement, lumière, pression, humidité, vibration, ..., etc.), et de la transformer en une autre grandeur physique manipulable par exemple un microphone est un capteur qui permet de transformer une onde sonore en un signal électrique.

Parmi ces capteurs on cite les capteurs qui sont utilisés par notre système qui sont les suivants :

#### ✓ Le capteur de température LM35

##### Description :

Le capteur de température LM35 est un capteur analogique de température fabriqué par Texas Instruments. Précis, peu coûteux, très simple d'utilisation et d'une fiabilité à toute épreuve.

Il est capable de mesurer des températures allant de  $-55^{\circ}\text{C}$  à  $+150^{\circ}\text{C}$  dans sa version la plus précise et avec le montage adéquat, de quoi mesurer n'importe quelle température.

La sortie analogique du capteur est proportionnelle à la température. Il suffit de mesurer la tension en sortie du capteur pour en déduire la température. Chaque degré Celsius correspond à une tension de  $+10\text{mV}$ .

Les figures III.2 et III.3 présentent respectivement le capteur de température LM35 et son câblage.



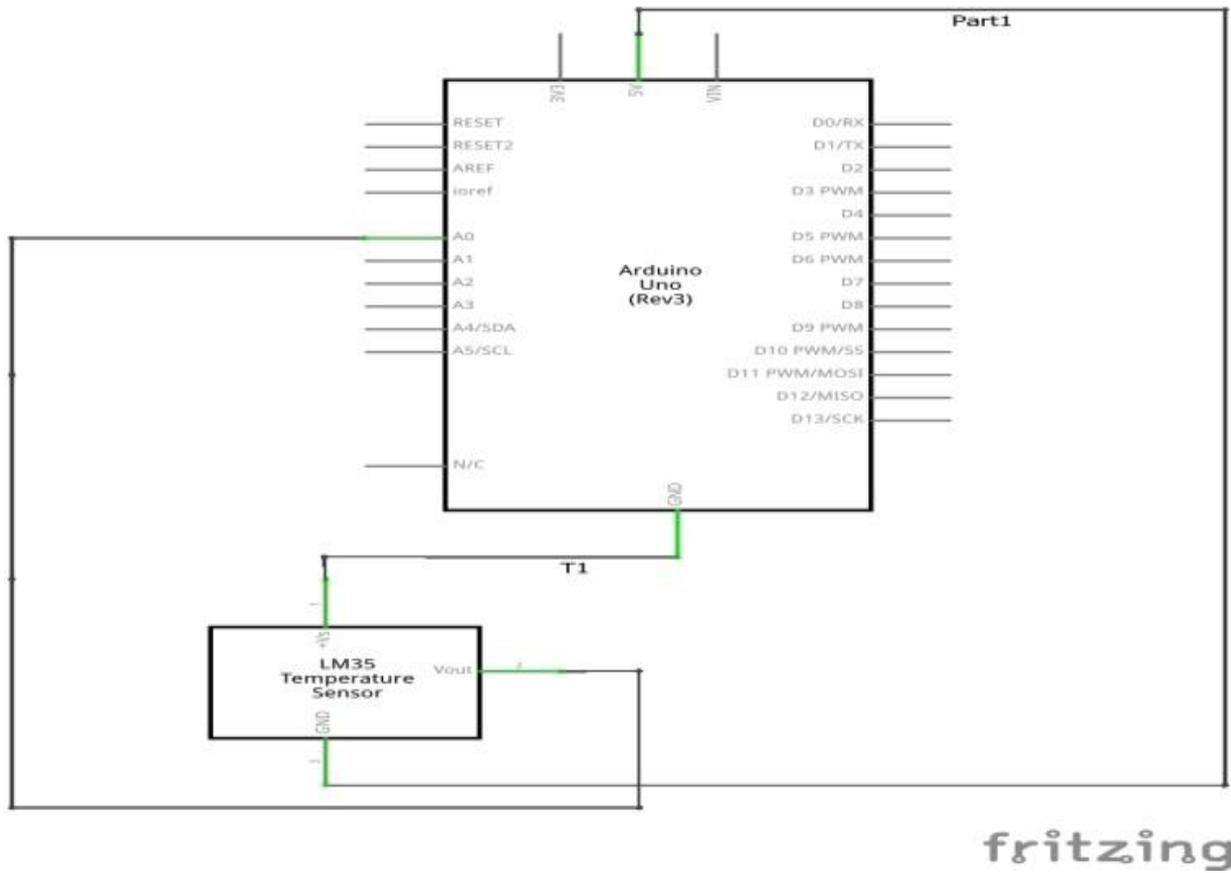


Figure III.4. Câblage du capteur de température à la carte Arduino



Figure III.5. Capteur de mouvement

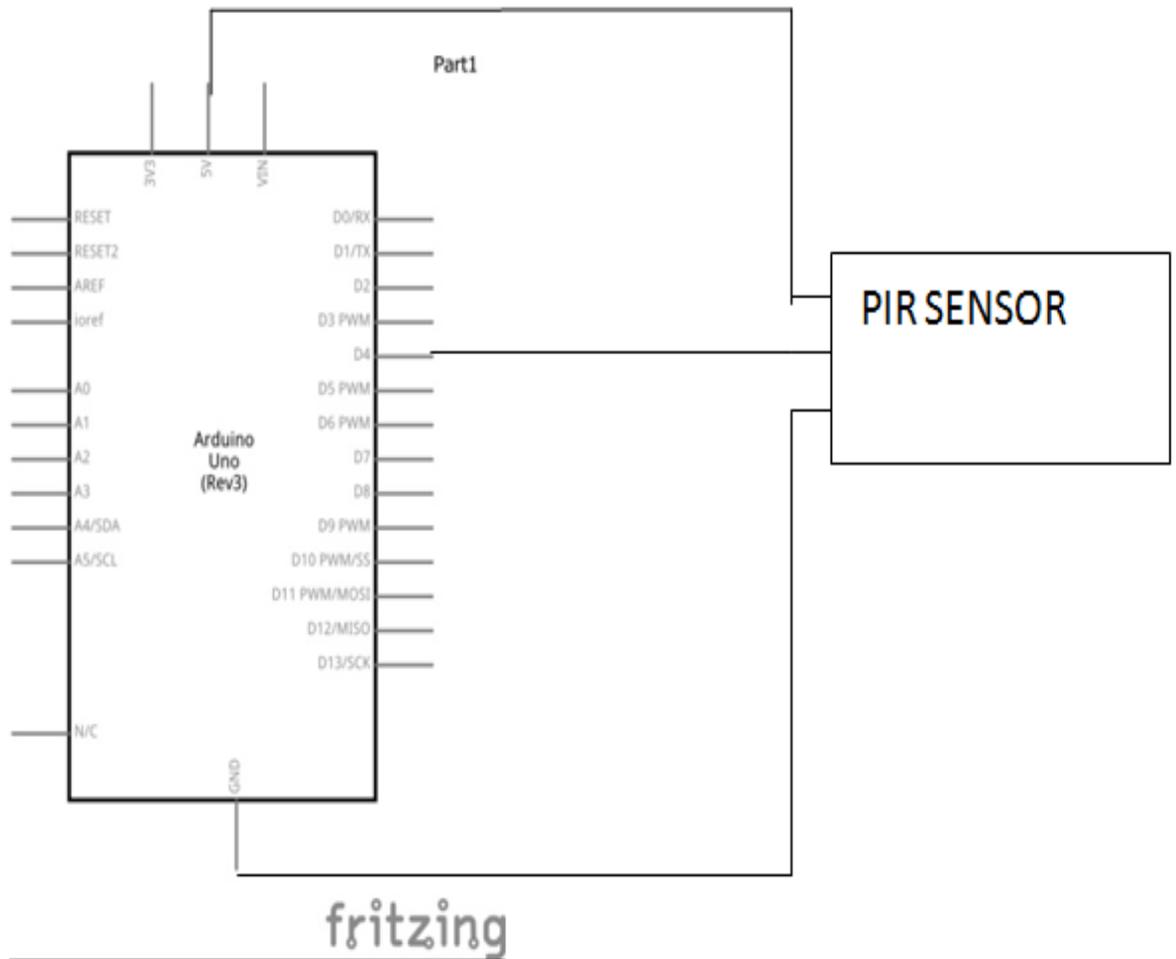
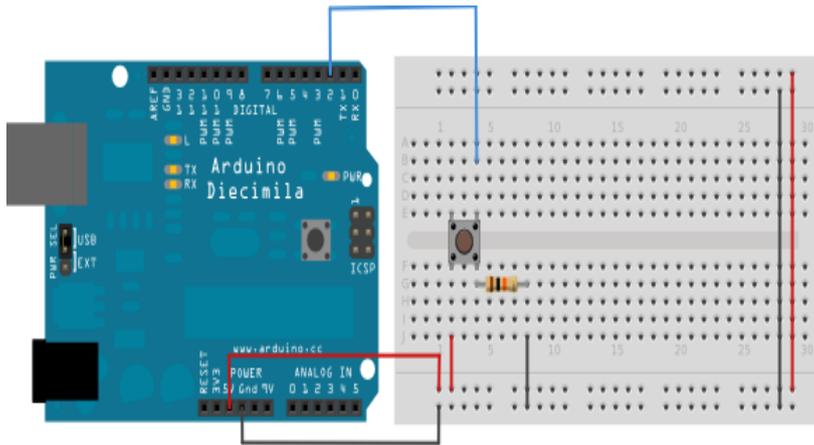


Figure III.6. Câblage du capteur de mouvement

### III.3.3. Le bouton

Un bouton poussoir sert à faire passer le courant lorsqu'on appuie dessus ou au contraire garder le circuit « éteint » lorsqu'il est relâché.

La figure III.7 Présente le câblage d'un bouton poussoir.



**Figure III.7. Câblage d'un bouton poussoir.**

### III.3.4. Les actionneurs

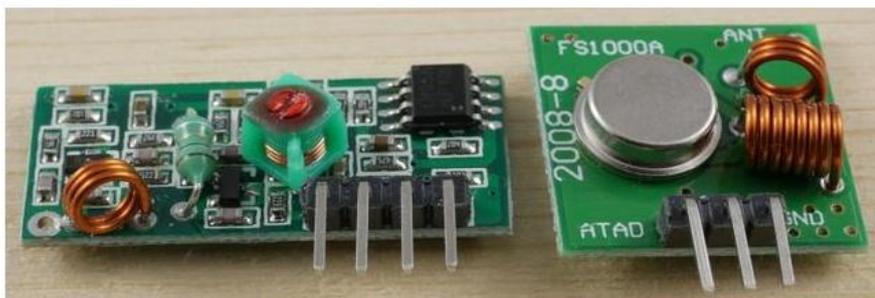
Un actionneur est un appareil qui transforme l'énergie qui lui est fournie en un phénomène physique qui fournit un travail, modifie le comportement ou l'état d'un système. Dans notre cas, on se servira des actionneurs suivants : allumage des leds, un afficheur LCD et un Buzzer.

#### III.3.4.1 Les modules de communications RF (radio fréquence) [8]

A fin de pouvoir faire communiquer deux Arduinos sans utiliser de liaison filaire, on se retrouve souvent confronté au choix plus ou moins difficile du type de communication à mettre en place. Pour cela on pourrait penser au différents types de communication sans fil tel que le bluetooth, wifi ou encore la radiofréquence. Dans notre système on a utilisé la radiofréquence.

Un module RF (module radiofréquence) est un petit dispositif électronique utilisé pour transmettre et / ou recevoir des signaux radio entre deux appareils. Dans un système embarqué, il est souvent souhaitable de communiquer avec un appareil sans fil. Cette communication sans fil peut être effectuée par une communication optique, soit par la radio fréquence (RF).

La figure III.8 présente un exemple de modules RF (émetteur et récepteur).



**Figure III.8.les modules RF.**

Les modules radio ont trois broches parfois quatre : VCC, DATA et GND, ainsi qu'une broche pour l'antenne qui est constituée d'un fil conducteur d'environ 17 cm qui permet d'augmenter considérablement la portée de notre paire émetteur-récepteur. La fréquence de la porteuse correspond à la fréquence sur laquelle le module radio émet ou écoute. L'émetteur et le récepteur doivent fonctionner sur la même fréquence pour pouvoir communiquer correctement. Ces modules sont généralement conçus pour fonctionner avec une alimentation de 5 volts sur la broche VCC.

Les signaux en entrée de la broche DATA de l'émetteur et en sortie de la broche DATA du récepteur sont de type logique TTL (tout ou rien, 0 volt ou 5 volts). Il n'y a donc pas besoin d'ajouter de conversion de niveaux logiques entre ces modules radio et une carte Arduino "classique" fonctionnant en 5 volts.

Dans le cas d'une carte Arduino fonctionnant en 3.3 volts, un convertisseur de niveaux logiques est requis au niveau du récepteur pour ne pas endommager l'entrée logique de la carte Arduino.

Caractéristiques : [7]

- Fréquence: 433Mhz.
- Portée radio (alimenté en 5V): 40m à l'intérieur, et 100m en extérieur.

Le branchement de ces modules se présente sur la figure III.9 pour l'émetteur et la figure III.10 pour le récepteur

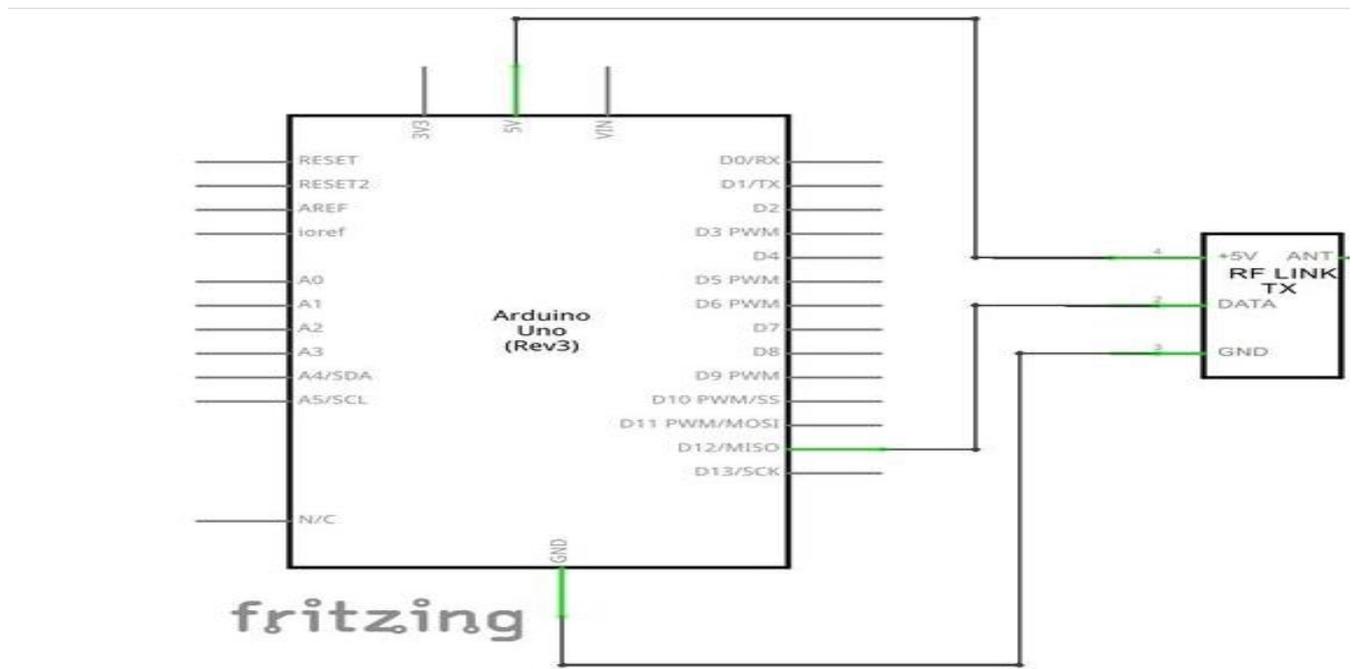
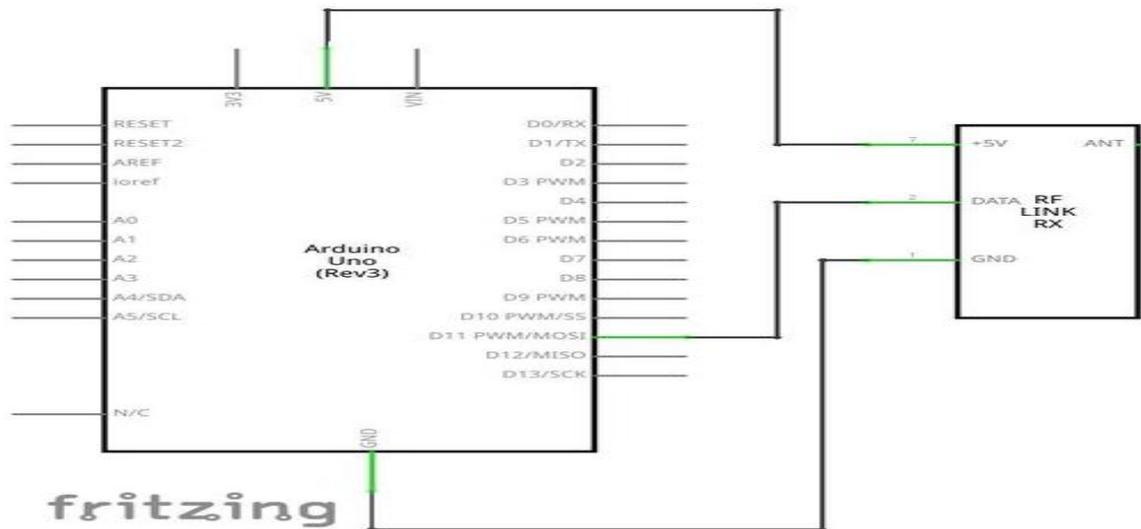


Figure III.9. Câblage de module RF (partie émetteur)



**Figure III.10. Câblage de module RF (partie récepteur).**

### III.3.4.2 L'afficheur LCD

Lors de la communication d'informations dans notre système nous aurons besoin d'un afficheur LCD sur lequel nous allons afficher les différentes informations sur la situation de la chambre du nourrisson ou du malade.

Les afficheurs LCD sont des modules compacts intelligents et nécessitent peu de composants externes pour un bon fonctionnement. Ils consomment relativement peu (de 1 à 5 mA).

Plusieurs afficheurs sont disponibles sur le marché et diffèrent les uns des autres, non seulement par leurs dimensions, (de 1 à 4 lignes de 6 à 80 caractères), mais aussi par leurs caractéristiques techniques et leur tension de service. Certains sont dotés d'un rétro éclairage de l'affichage. Cette fonction fait appel à des LED montées derrière l'écran du module, cependant, cet éclairage est gourmand en intensité (de 80 à 250 mA).

Dans notre système nous avons utilisé un afficheur LCD 2\*16(2 lignes et 16 colonnes).

Ces afficheurs contiennent des broches que chaque une a un rôle tel que :

**VCC, Masse** : alimentation de l'afficheur LCD. Un afficheur LCD s'alimente entre 0V et 5V.

**Contraste** : entrée permettant de régler le contraste de l'afficheur LCD. Il faut appliquer une tension continue réglable (entre 0V et 5V) à l'aide d'un potentiomètre.

**Vled** : différence de potentiel permettant de commander le rétro éclairage.

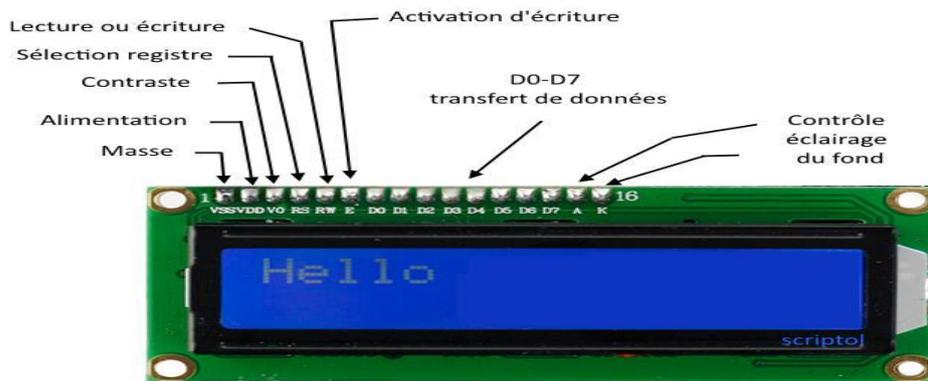
**E** : entrée de validation (ENABLE), elle permet de valider les données sur un front descendant. Lorsque E=0 alors le bus de données est à l'état haute impédance.

**RS** : Register Select cette entrée permet d'indiquer à l'afficheur si l'on souhaite réaliser une commande (RS=0) par des instructions spécifiques ou écrire une donnée (envoi du code du caractère à afficher) sur le bus (RS=1).

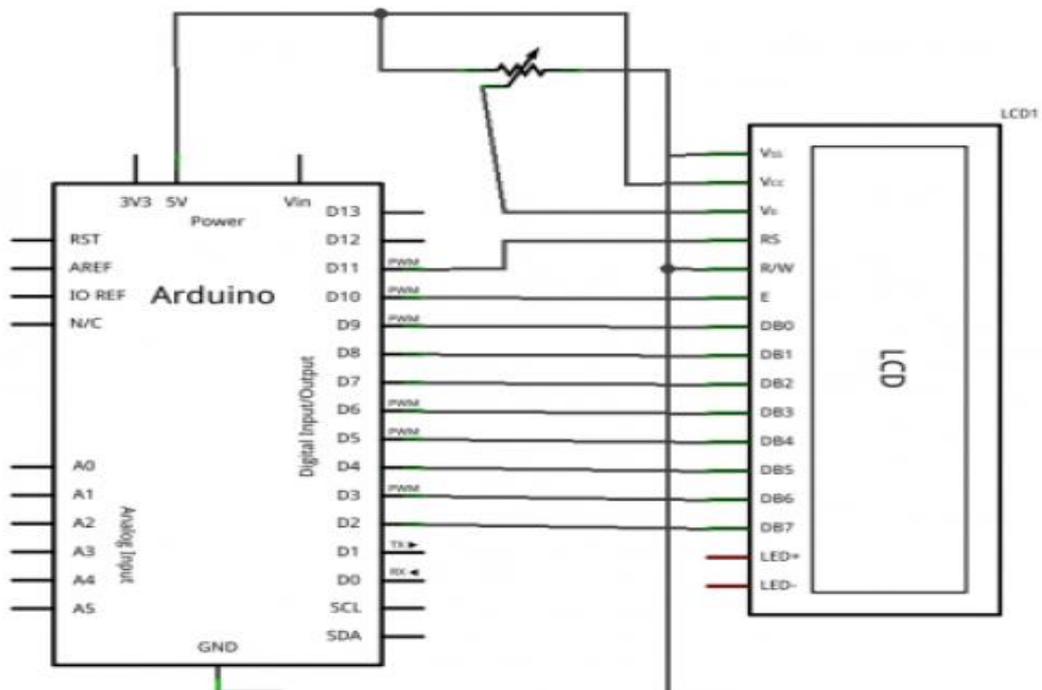
**R/W** : entrée de lecture (R/W=1) et d'écriture (R/W=0). Lorsqu'on commande l'afficheur LCD il faut se placer en écriture.

**D7...D0** : bus de données bidirectionnel, il permet de transférer les instructions ou les données à l'afficheur LCD.

Les figures III.11 et III.12 représentent respectivement l'afficheur LCD, le câblage d'un afficheur LCD.



**Figure III.11.L'afficheur LCD.**



**Figure III.12.Câblage d'un afficheur LCD.**

## III.3.4.3 Les LEDs

Pour signaler des évènements tels que la température élevée et un mouvement survenu dans la chambre que nous surveillons ; des leds sont mise en œuvre.

Une LED est une abréviation anglaise de *Light Emitting Diode* qui désigne un composant électronique. Ce composant présente la particularité d'émettre de la lumière lorsqu'il est alimenté par un courant électrique.

Les figures III.13 et III.14 représentent respectivement une led, le câblage de cette Led.

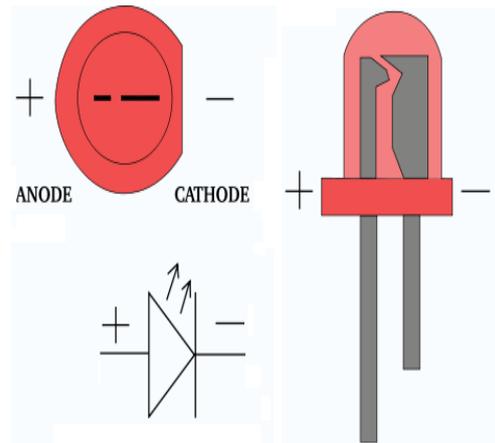


Figure III. 13. LED.

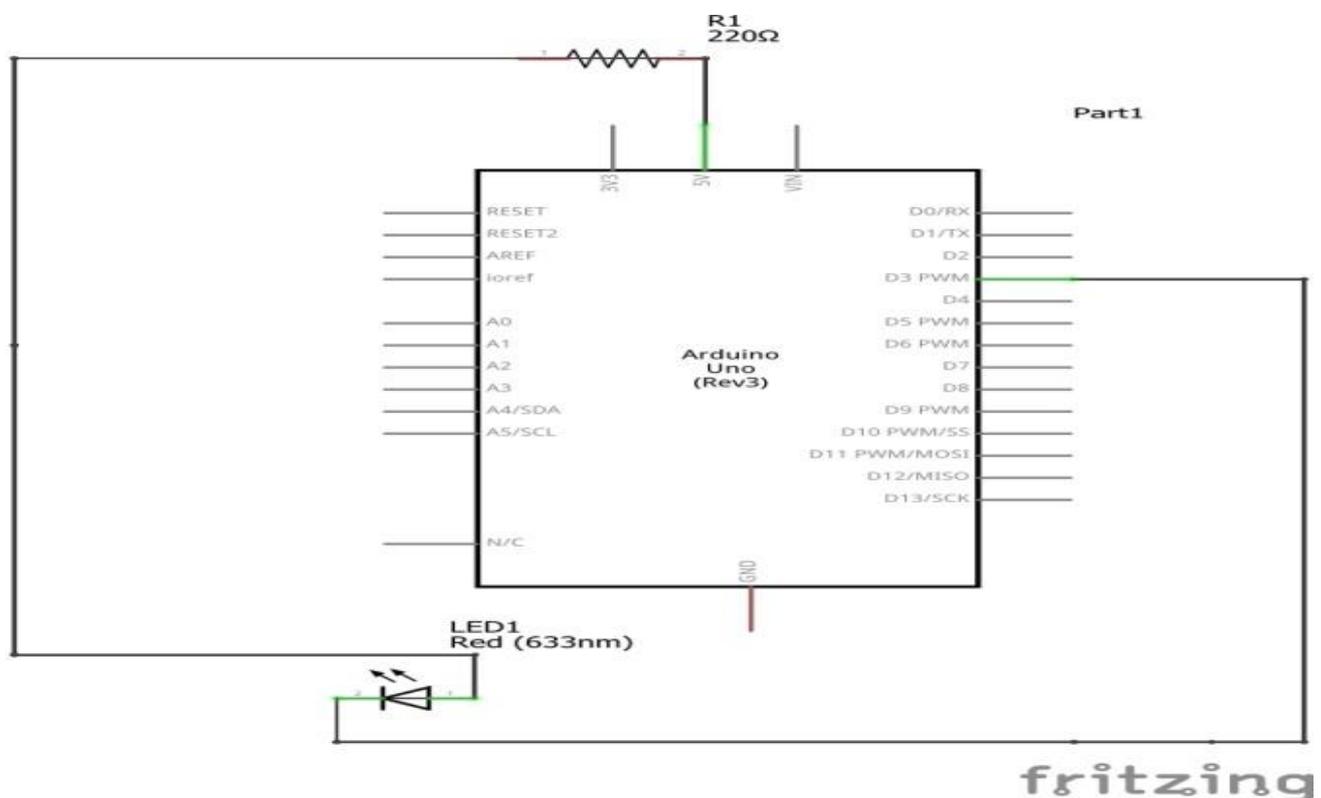


Figure III.14. Câblage d'une LED.

## III.3.4.4 Le Buzzer

Dans notre système, une signalisation sonore est un élément appréciable en cas d'un événement survenu, pour cela on utilise un buzzer.

La figure III.15 présente le branchement d'une alarme

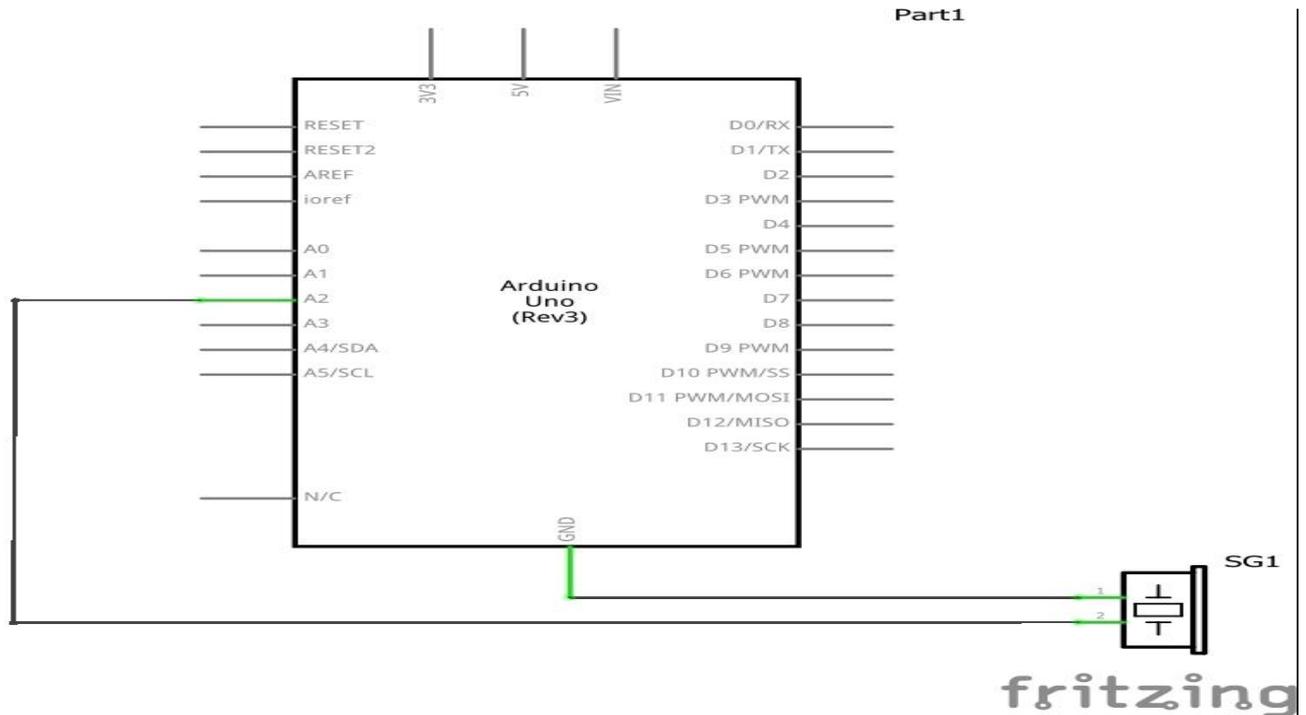


Figure III.15.Câblage d'une alarme (buzzer).

## III.3.5.le contrôleur :

La carte Arduino UNO est l'élément essentiel pour le contrôle de notre système, en effet toutes les données reçues par les capteurs seront traitées et exploitées par cette carte.

Elle envoie aussi des commandes qui seront exécutés suite à des données captés vers les actionneurs adéquats.

## Schéma générale du système

Les figures III.16 et III.17 présentent respectivement le schéma du sous système(a) et le sous système (b)





### III.4. Conception logicielle

Une partie logicielle étant une partie incontournable pour le fonctionnement de notre système, elle permet en outre de programmer et vérifier les composants matériels utilisés dans notre système et qui sont connectés au carte Arduino.

Dans cette partie nous allons détailler quelques composants de notre système et nous finirons cette partie par l'organigramme de notre système.

#### III.4.1 Le principe de fonctionnement

Dans le Sous-système(a) on dispose de :

- Une carte Arduino UNO.
- Un Breadbord.
- Des fils pour l'alimentation.
- Et un câble USB pour la communication entre la carte arduino et l'ordinateur.
- Capteur de température LM35 Son rôle est de capter à chaque instant la valeur de la température dans la chambre de nourrisson ou de malade.
- Capteur de mouvement PIR qui indique la présence ou non de mouvement dans la chambre.
- Un bouton poussoir qui sera appuyé par le malade en cas de besoin.
- Un module RF émetteur qui permet de transmettre toutes les valeurs captées.
- Des LEDs.

Dans le système(b) on dispose de :

- une carte Arduino UNO.
- Un Breadbord.
- Des fils pour la connectivité et l'alimentation.
- Et un câble USB pour la communication entre la carte arduino et l'ordinateur.
- Une alarme (buzzer)
- Un afficheur LCD
- Un module RF récepteur qui permet d'acquérir les valeurs envoyées par l'émetteur

Au niveau du Sous-système (a) la valeur captée par LM35 est traitée comme suit :

- $18^{\circ} < T < 25^{\circ}$  on affiche sur l'écran LCD « Température idéale » au niveau du système(b)
- $T < 18^{\circ}$  on allume une led rouge au niveau du système émetteur et dans le système de réception on affiche sur l'écran LCD « Température basse » et on déclenche l'alarme.
- $T > 25^{\circ}$  on allume une led rouge au niveau du système émetteur et dans le système de réception on affiche sur l'écran LCD « Température élevée » et on déclenche l'alarme.

Le capteur de mouvement a deux valeurs :

- Si la valeur du mouvement = 1 donc un mouvement est détecté dans la chambre pour cela une led verte sera allumée au niveau du sous-système (a) et au niveau du sous-système (b) un message d'alerte « Attention mouvement » sera affiché sur l'écran LCD accompagné d'un avertissement sonore
- Sinon si la valeur du mouvement=0 il y aura pas de réaction.

Si le malade a appuyé sur le bouton d'appel, dans ce cas sa valeur deviendra 1, donc on allume une led bleu et l'avertissement sonore sera déclenché au niveau du sous-système (b).sinon sa valeur restera toujours 0

Vu qu'à tout moment on peut avoir un problème lors de la transmission des messages d'information ou d'alerte, donc pour pouvoir intervenir à temps on sera alerté par un message « problème de communication » accompagné d'un signal sonore.

### **III.4.1.1 L'organigramme de sous-système (a) :**

Le fonctionnement du sous- système (a) est résumé par l'organigramme de la figure III.18

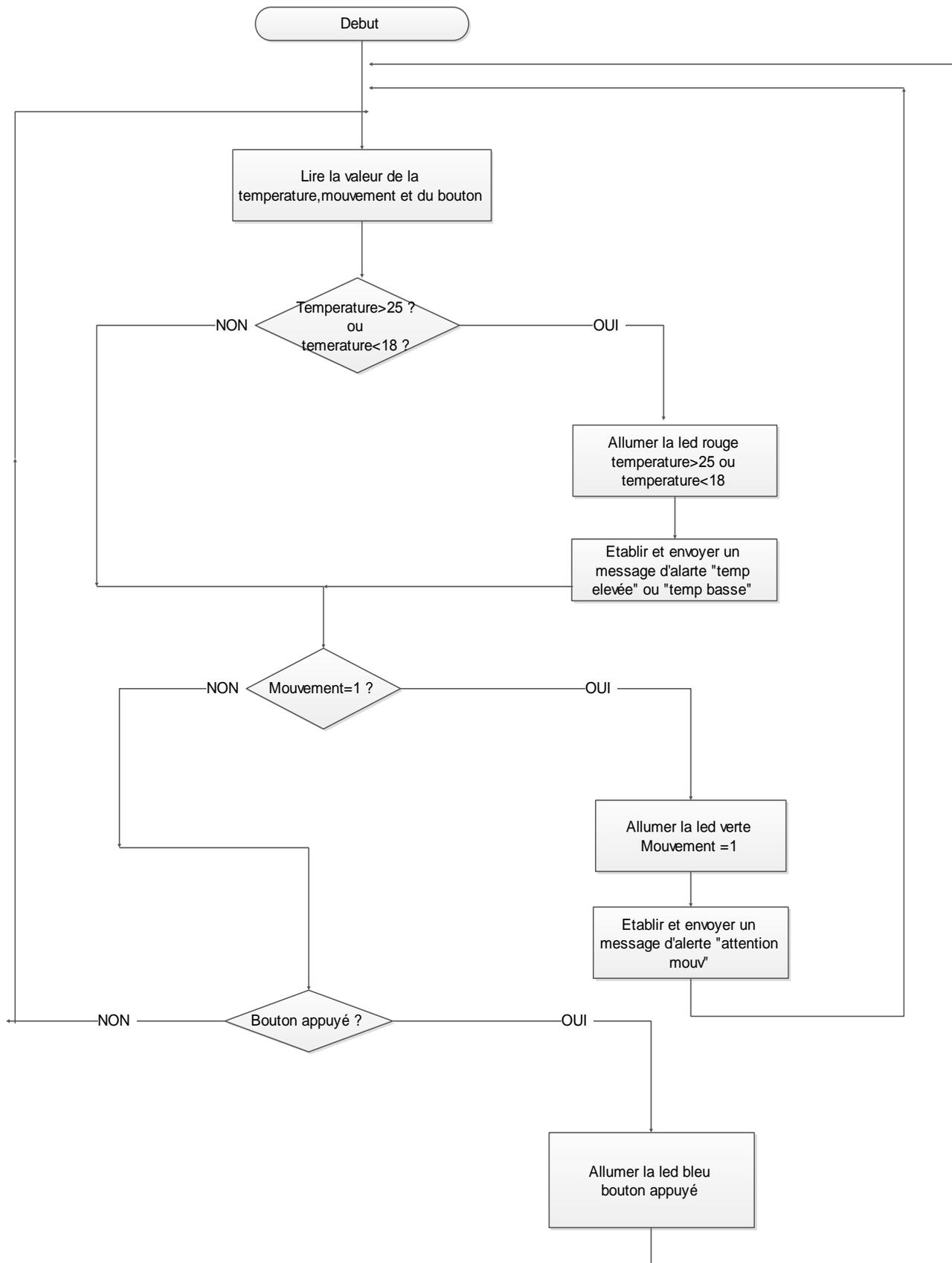
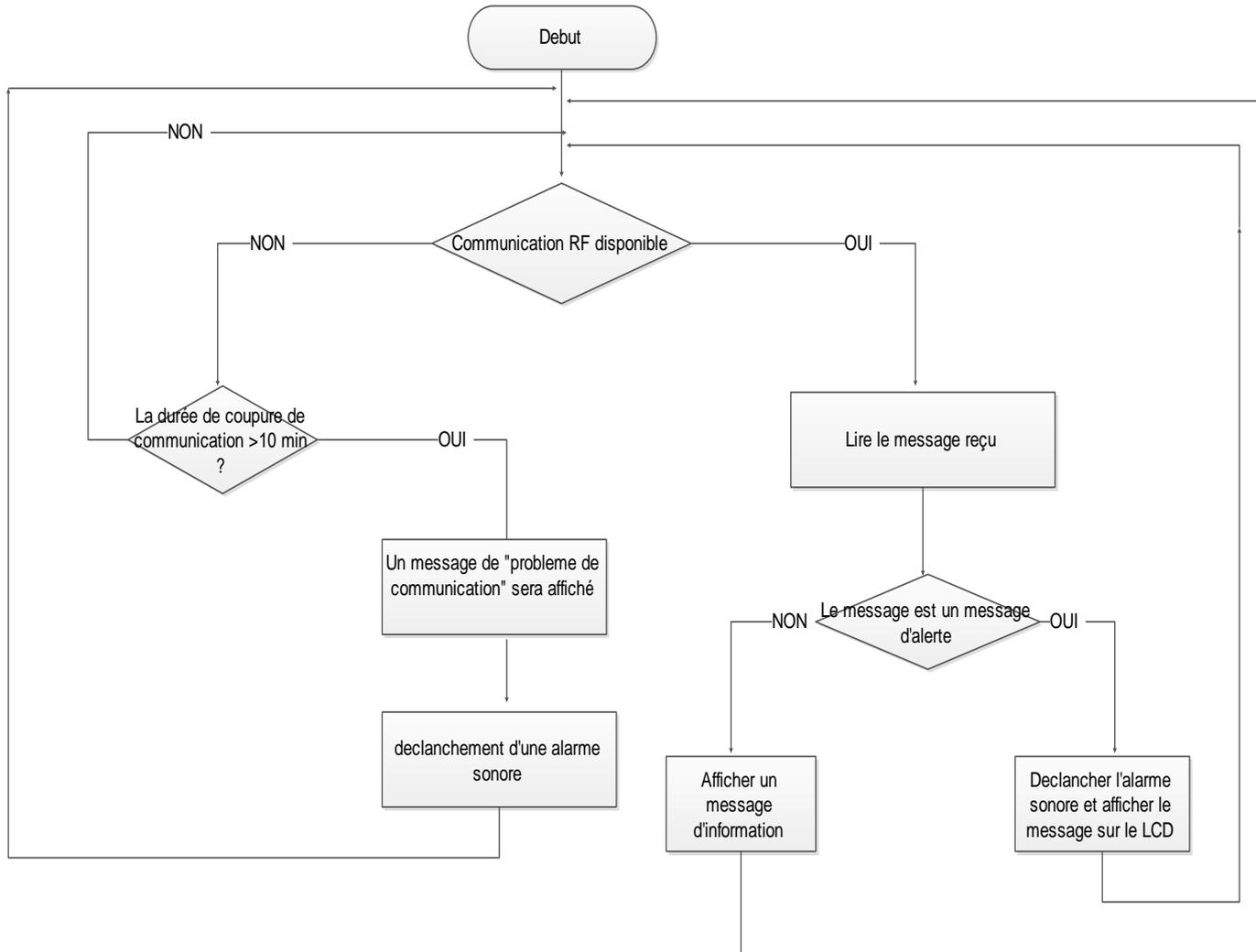


Figure III. 18 .L'organigramme du fonctionnement du système (a).

## III.4.1.2 L'organigramme de sous-système (b) :

Le fonctionnement du sous- système (b) est résumé par l'organigramme de la figure III.19



**Figure III. 19. L'organigramme du fonctionnement du système (b).**

## III.5 Conclusion

Le but de ce chapitre était de décrire la structure général de notre système, présenter la conception matérielle dans laquelle nous avons pu détailler l'ensemble des dispositifs dont on aura besoin pour la réalisation de notre système puis la conception logicielle dans laquelle à partir d'un exemple illustré nous avons décrit quelques fonctions du langage Arduino.

Dans le chapitre suivant, on passe à la réalisation et la mise en œuvre de notre système.

### IV.1 Introduction

Après avoir achevé la partie conception, nous entamons dans ce chapitre la phase réalisation de notre application. Nous allons donc présenter en première partie les environnements matériels, ensuite nous présenterions l'implémentation de notre application.

### IV.2 Les composants matériels utilisés

Pour la réalisation de notre projet nous avons besoin d'un ensemble de composants matériels

Qui seront présenté dans le tableau suivant en décrivant le rôle de chaque outil.

Composant	Rôle
Carte Arduino UNO	C'est le composant principal de notre projet car est le gestionnaire autre composant
Module RF	Il est fait pour établir la communication entre deux sous-systèmes
Capteur de température	Il permet de mesurer la température
Capteur de mouvement	Il permet de capter le mouvement
Buzzer	Est un actionneur permet de produire un son quand on lui applique une tension
Bouton poussoir	Permet d'activer ou de désactiver l'alarme
Led	Elle permet d'indiquer un état d'un system
Afficheur LCD	Il permet d'afficher le message d'alerte et d'information
Potentiomètre	Son rôle est d'ajuster le contraste de l'afficheur LCD
Fils	Ils permettent de connecter les différents composants
Bread board	Il permet d'établir un circuit électronique
Câble USB	Il permet de faire communiquer la carte arduino avec l'ordinateur en même temps de l'alimenter
Résistance	Elle permet de réduire l'intensité de courant et ou la tension

**Tableau IV.1 l'ensemble de matériel utilisés.**

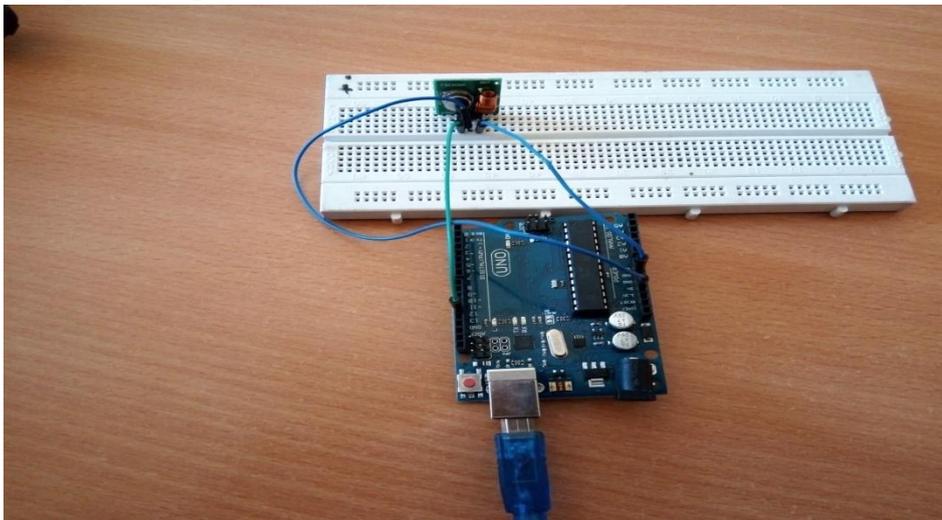
### IV.3 Branchement des différents composants avec la carte Arduino :

#### IV.3.1. les module RF

##### ➤ Le module RF (émetteur)

Pour câbler le module RF (émetteur) avec notre carte Arduino UNO, on doit connecter respectivement les broches VCC, GND et ATAD du module RF aux broches 5V, GND et D12 de la carte Arduino UNO du sous-système (a)

La figure IV.1 présente le branchement du module RF (émetteur).



La figure IV.1 Branchement du module RF (émetteur).

##### ➤ Le module RF (récepteur)

Pour câbler le module RF (récepteur) avec notre carte Arduino UNO, on doit connecter respectivement les broches VCC, GND et DATA du module RF aux broches 5V, GND et D11 de la carte Arduino UNO du sous-système (b).

La figure IV.2. présente le branchement du module RF (récepteur).

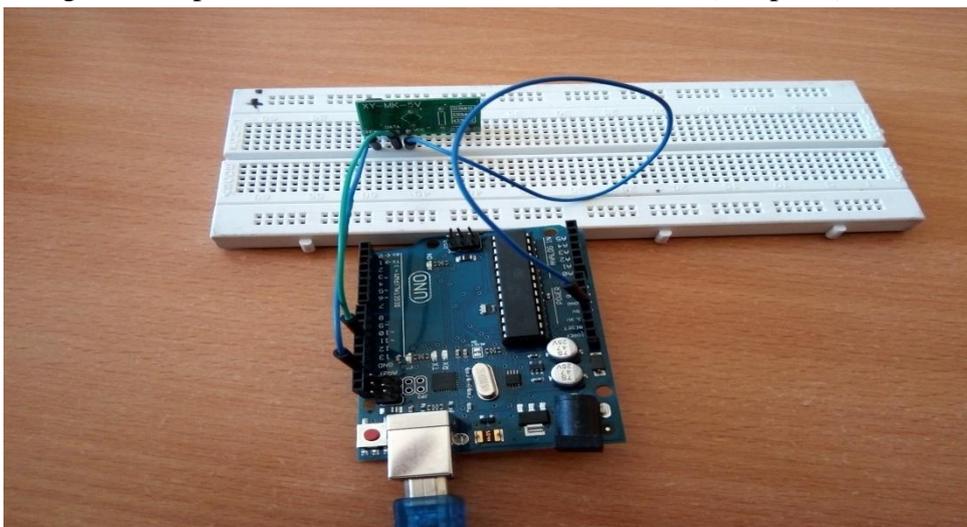


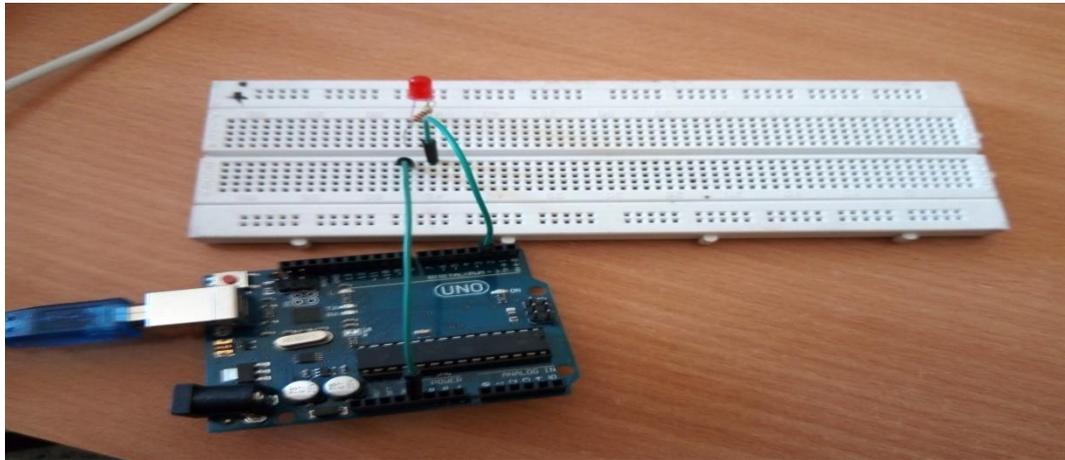
Figure IV.2. Branchement du module RF (récepteur).



### IV.3.3.Les LEDs.

Pour câbler une LED avec notre carte Arduino UNO, on doit connecter la plus grande patte de la LED à une résistance puis à la broche 5V de la carte Arduino UNO du sous système (a) et la petite patte sera connectée à la broche GND de la carte Arduino UNO.

La figure IV.5 présente le branchement d'une LED

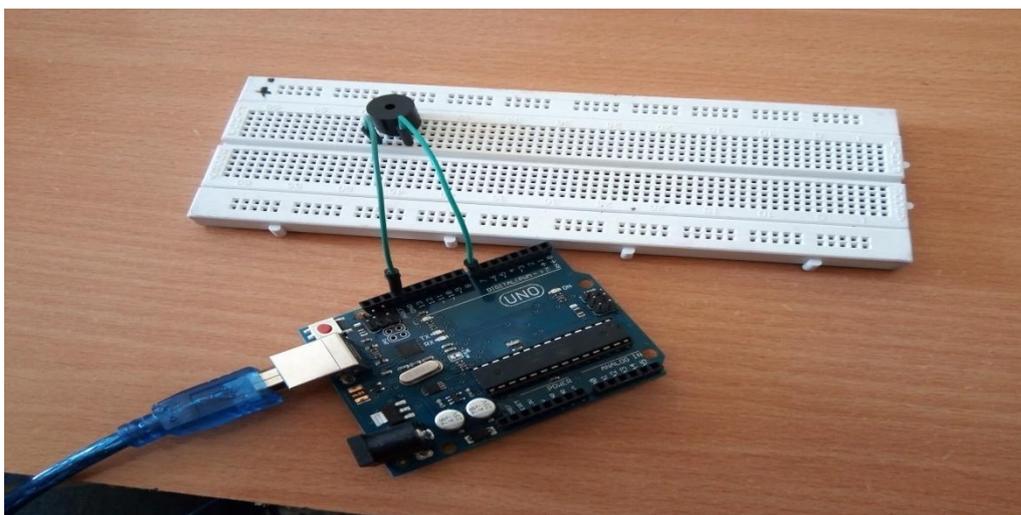


**Figure IV.5 Branchement d'une LED**

### IV.3.4.Le Buzzer.

Pour câbler un Buzzer avec notre carte Arduino UNO du sous-système(b), on doit connecter sa plus grande patte à la broche D7 de la carte Arduino UNO et la petite patte sera connectée à la broche GND de la carte Arduino UNO.

La figure IV.6 présente le branchement d'un Buzzer.

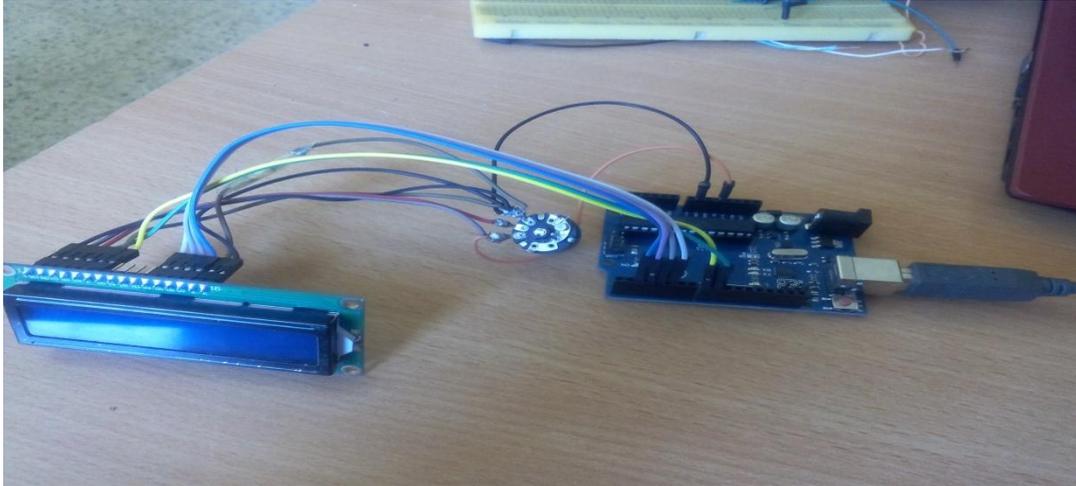


**Figure IV.6 présente le branchement d'un Buzzer.**

### IV.3.5 Écran LCD.

Pour câbler l'écran LCD avec la carte Arduino UNO du sous-système(b), nous avons utilisé les broches (D2,D3 ,D4 ,D5,D8,D9)ainsi que les broches :GND et VCC de la carte Arduino UNO.

La figure IV.7 présente le branchement du LCD

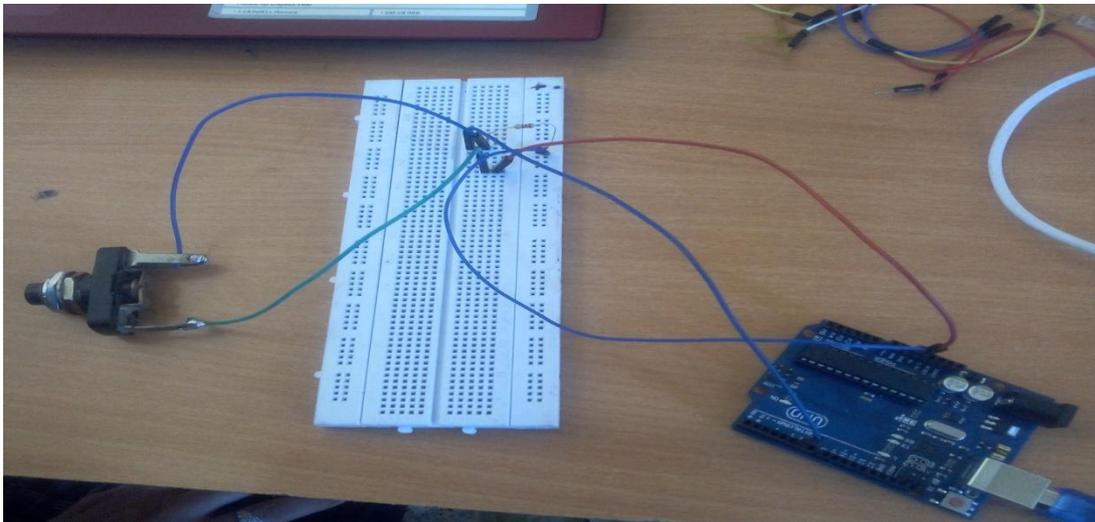


**Figure IV.7 Branchement du LCD.**

### IV.3.6.le Bouton poussoir

Pour câbler le bouton poussoir avec notre carte Arduino UNO du sous-système(a), on doit connecter la broche D6 à l'un des fils de bouton et l'autre à la broche 5V de la carte Arduino UNO.

La figure IV.8 présente le branchement du bouton poussoir.



**Figure IV.8 : Branchement du bouton poussoir.**

### IV.4 Implémentation de l'application :

Pour le fonctionnement de notre système, une partie logicielle qui est une partie incontournable doit être mise en œuvre, en effet elle permet de programmer et contrôler chaque composant matériel de notre système.

Dans cette partie nous allons expliquer et décrire quelques fonctions et les bibliothèques utilisées afin de programmer les composants matériels de notre système.

### IV.4.1 Description de quelques fonctions et bibliothèques utilisées dans le sous-système (a)

#### IV.4.1.1 Au niveau de la partie déclarative

##### a) Inclusion de quelques bibliothèques :

- **Inclusion de la bibliothèque VirtualWire**

Afin de pouvoir établir une communication fiable entre 2 cartes Arduino, nous allons inclure la bibliothèque « **VirtualWire** » et cela comme suit :

```
#include <VirtualWire.h>
```

##### b) Déclaration de quelques constantes :

- **Déclaration des constantes pour les broches utilisées dans le programme :**

```
int bouton=6; // le bouton.
```

```
int sensor = 0; // capteur de température.
```

```
int mouv=7; //capteur de mouvement
```

#### IV.4.1.2 Au niveau de la fonction d'initialisation setup () :

Initialisation des fonctionnalités utilisées :

- **Vitesse de transmission de l'Arduino :**

```
Serial.begin(9600); // vitesse de transmission de l'Arduino ou bien Initialisation du port série pour avoir un retour sur le serial monitor.
```

**Remarque :** on a utilisé cette fonction d'initialisation « Serial.begin » uniquement pour vérifier les résultats sur le moniteur série.

- **Initialisation de la bibliothèque VirtualWire**

```
vw_setup(2000); //initialisation de la bibliothèque VirtualWire à 2000 bauds
```

- **Définition de la broche d'envoi.**

```
vw_set_tx_pin(12); // définit la broche D12 comme broche d'émission.
```

Configuration des broches utilisées :

- **La configuration des broches de LED**

```
pinMode(ledrouge, OUTPUT); // initialise la LED comme une sortie.
```

#### IV.4.1.3 Au niveau de la boucle principale loop() :

On commence d'abord par la lecture des valeurs des capteurs (mouvement et température) :

```
temp = analogRead(sensor); // la lecture de la valeur du capteur de température.
```

Ensuite si la valeur de la température est supérieure à 25 ou inférieure à 18 une LED rouge sera allumée

```
digitalWrite(2, HIGH); // permet d'allumer la led rouge.
```

Enfin, la valeur de la température sera envoyée au sous-système (b) :

```
vw_send((byte*)&Lmsg, sizeof(msg)); // transmission des données
```

```
vw_wait_tx(); // Il attend la bonne émission du message.
```

### IV.4.2. la description de quelques fonctions et librairies utilisées dans le sous-système (b)

#### IV.4.2.1 Au niveau de la partie déclarative

##### a) Inclusion de quelques librairies :

De même que le sous-système(a) la librairie « **VirtualWire** » sera utilisée

- **Inclusion de la librairie LiquidCrystal.h**

Afin de pouvoir afficher des messages sur l'écran LCD, on doit inclure une librairie « **LiquidCrystal.h** » dans le programme du sous-système(b) et cela comme suit :

```
#include <LiquidCrystal.h>
```

- **Déclaration d'un objet LCD alphanumérique**

```
LiquidCrystal lcd(9,8,5,4,3,2); // broches utilisées pour connecter l'afficheur LCD.
```

#### IV.4.2.2 Au niveau de la partie setup()

- **Initialisation de l'afficheur LCD :**

```
Lcd.begin(16,2); //initialise le LCD avec 16 colonnes et 2 lignes.
```

- **Démarrage de la partie réception de la librairie VirtualWire**

```
vw_rx_start(); // fonction qui permet le démarrage de la partie de la réception.
```

- **Définition de la broche de réception.**

```
vw_set_rx_pin(11); //définit la brocheD11 comme broche de réception.
```

- **La configuration des broches du Buzzer**

```
pinMode(7,OUTPUT) //initialise le buzzer comme sortie
```

#### IV.4.2.3 Au niveau de la fonction loop()

Dans cette partie on doit d'abord vérifier le temps de l'arrivée d'un nouveau message, si elle dépasse un certain temps, une erreur de « problème de communication » sera affichée sur l'écran LCD et cela est réalisé grâce à la fonction `vw_wait_rx_max()` sinon cette partie donne la main à la suite du programme qui est résumée par la procédure suivante :

Copie le message reçu grâce à la fonction (`vw_get_message((byte *) &Lmsg, &taillemsg)`), puis un message sera établi et affiché sur l'écran LCD ,de plus si le message affiché sur l'écran est un message d'alerte alors il sera accompagné par un déclenchement d'une alarme .

Le déclenchement d'une alarme est fait par la fonction `tone(7,200,200);`

### **IV.5 Conclusion**

Ce chapitre étant la clôture de notre projet dont lequel nous avons décrit l'étape de réalisation. D'abord nous avons commencé par la présentation des composants utilisés pour développer notre application qui permet de surveiller en permanence la chambre du nourrisson/malade, ensuite nous sommes passés à l'étape qui consiste à présenter la réalisation de notre application.

## Conclusion générale et perspectives

Nous avons, au cours de ce mémoire, contribué à la Conception et réalisation d'un système de surveillance (température et mouvement) d'une chambre de malade/nourrisson.

Et pour réaliser ce système on a subdivisé notre mémoire en deux parties :

- Dans la première partie, nous avons fait une étude théorique des concepts de base nécessaires à la mise en œuvre de notre système plus précisément des notions sur les systèmes embarqués, la domotique et la carte Arduino UNO.
- Dans la deuxième partie, nous avons présenté la conception et la réalisation et la mise en œuvre de notre système, en effet dans la conception on a pu présenter le câblage des deux sous-systèmes de notre système ainsi que leurs organigrammes ; dans l'implémentation on a pu faire des tests sur notre système à fin de bien vérifier le bon fonctionnement du système.

Ce présent travail nous a permis :

- D'approfondir nos connaissances théoriques et pratiques en rapport avec les systèmes embarqués, la domotique et le dispositif programmable Arduino.
- D'acquérir de nouvelles connaissances sur la réalisation des systèmes embarqués en effet c'est notre premier système embarqué réalisé au cours de notre cursus.

Nous estimons avoir atteint les objectifs fixés au préalable, néanmoins des perspectives peuvent être envisagées afin d'améliorer les fonctionnalités offertes par notre système :

- Pouvoir déclencher ou arrêter un climatiseur l'été et un chauffage l'hiver on intégrant au système un relais et ce afin de rendre le système encore plus autonome.
- Mettre en place un clavier numérique qui permet le paramétrage du système et les seuils des capteurs.

Notre système est composé de deux sous-système pour cela deux programmes sont mise en œuvre et qui sont les suivants :

### 1. Le programme du sous-système(a)

La figure A.1 présente le programme du sous système (a).

```
#include <VirtualWire.h>

float temp;
int sensor = 0;
float msg1;
int ledrouge=2;
int ledverte=3;
int ledbleu=4;
int bouton=6;
int valeurmouv=0;

void setup() {
  Serial.begin(9600);
  vw_set_tx_pin(12);          //définit la brocheD12 comme broche de l'émission.
  vw_setup(2000);
  pinMode(6, INPUT);
  pinMode(ledrouge, OUTPUT);
  pinMode(ledverte, OUTPUT);
}

void loop() {
  int msg[6];
  temp = analogRead(sensor);
  temp = temp * 0.48828125;
  if (temp>30){
    digitalWrite(2, HIGH);
  }else
    digitalWrite(2, LOW);

  Serial.print("la temperature est ");
  Serial.println(temp);

  msg[0]=digitalRead(7);
  if (msg[0]==1){
    digitalWrite(4, HIGH);
  }else
    digitalWrite(4, LOW);
}
```

```
msg[1]=int(temp);
valeurmouv=digitalRead(6);
if(valeurmouv==1){

    digitalWrite(3,HIGH);
}else
    digitalWrite(3,LOW);
msg[2]=valeurmouv;
Serial.print("la valeur du mouvement est");
    Serial.println(msg[0]);
vw_send((byte*)&msg, sizeof(msg)); // envoyer la donnée
vw_wait_tx(); // attendre la fin d'emission de message
delay(200);
    Serial.println(msg[2]);

}
```

Figure A.1 le programme du sous système (a)

## 2. Le programme du sous-système(b)

La figure A.2 présente le programme du sous-système (b).

```
#include <LiquidCrystal.h>

#include <VirtualWire.h>
#include <VirtualWire_Config.h>
LiquidCrystal lcd(9,8,5,4,3,2);

    long temp;
void setup() {
    lcd.begin(16,2);
    Serial.begin(9600);
    pinMode(3,OUTPUT);

    vw_setup(2000);
    vw_rx_start(); // On peut maintenant recevoir des messages
    vw_set_rx_pin(11); //définit la brocheD12 comme broche de l'émission.
}
void loop() {
    int msg[6];
    byte taillemsg=sizeof(msg);
    if( vw_wait_rx_max( 5000)){
if (vw_get_message((byte *) &msg, &taillemsg)) {
    Serial.print("la valeur de mouvement est ");
    Serial.println(msg[0]);
    if (msg[0]==1){
        lcd.setCursor(0,1);
        lcd.print("att mouv");
    }
}
```

```
    delay(1000);
    Serial.print("la temperature est ");
    Serial.println(msg[1]);
    if( msg[1]>30){
    Serial.println("elevé");
    lcd.setCursor(0,0);
    lcd.print("temp eleve");
    tone(7,200,200);
    }else if( msg[1]<=29){
    lcd.clear();
    Serial.println("basse");
    lcd.setCursor(0,0);
    lcd.print("temp basse");
    tone(7,200,200);
    }
    else {
    Serial.println("temperature normal");
    lcd.setCursor(0,0);
    lcd.print("temp ideale");
    noTone(7);
    }

    delay(1000);
    Serial.print("la valeur du botton est ");
    Serial.println(msg[2]);
    if(msg[2]==1){
    tone(7,200,200);
    }else noTone(7);
    }
    }else{
    lcd.setCursor(0,0);
    lcd.print("Probleme de ");
    lcd.setCursor(0,1);
    lcd.print("communication ");
    Serial.println("erreur d'envoi");}}
```

**Figure A.2** le programme du sous système (b).

3. le Branchement du sous-système (a).  
la figure A.3 présente le branchement du sous-système (a).

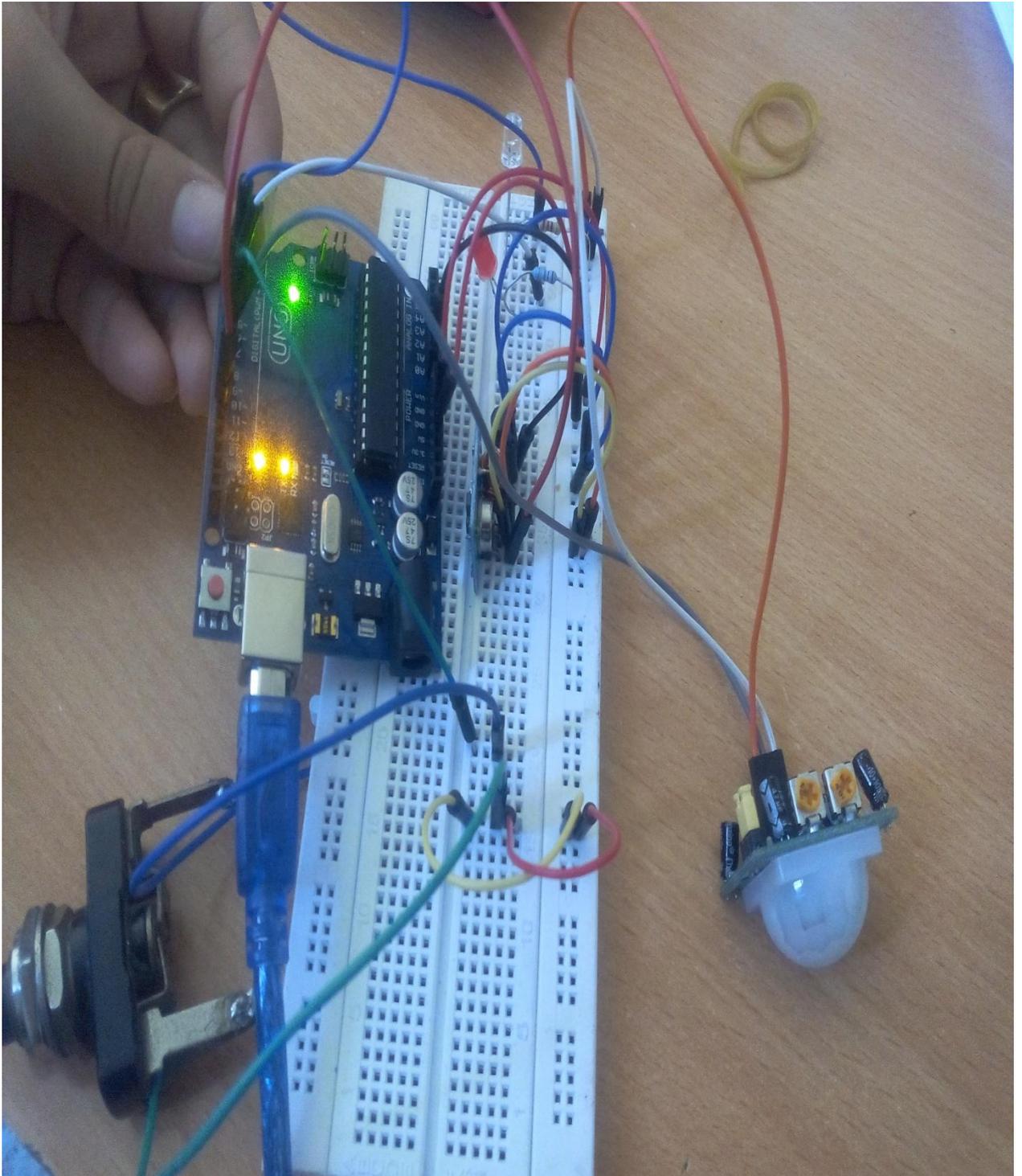
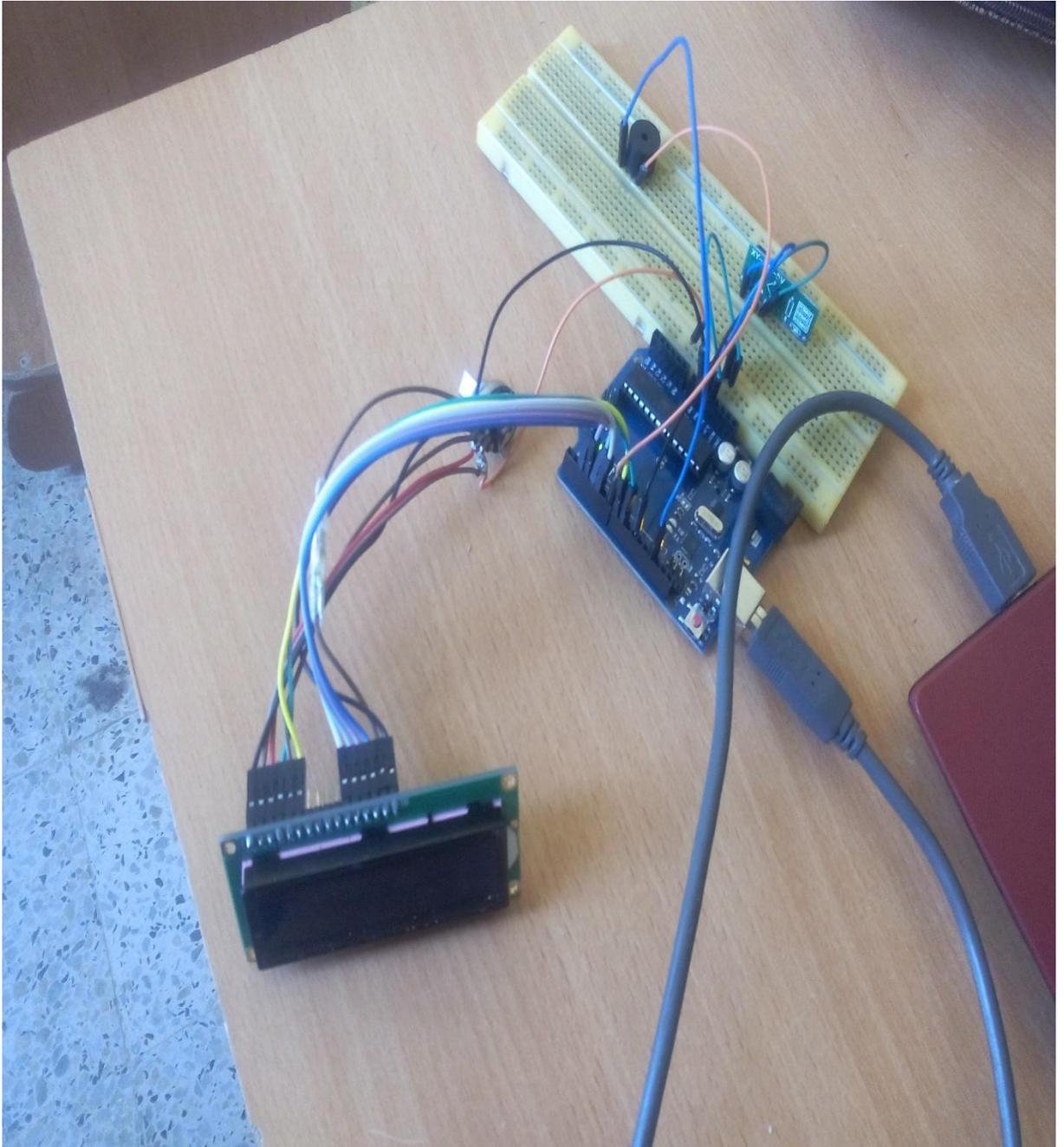


Figure A.3.branchement du sous-système (a)

**4. le Branchement du sous-système (b)**

la figure A.4 présente le branchement du sous-système(b).



**Figure A.3.le branchement du sous-système (b).**

# BIBLIOGRAPHIE

- [1] le site arduino <https://www.arduino.cc>
- [2] <http://www.sirlan.com/livreblanc.pdf>
- [3] <http://www.forumconstruire.com/guides/guide-domotique/principes-domotique>
- [4] <https://economie-d-energie.ooreka.fr/comprendre/domotique-economie-energie>
- [5] <http://cirta.fr/wp-content/uploads/2013/07/46-Domotique.pdf>
- [7] <http://www.technologuepro.com/cours-systemes-embarques/cours-systemes-embarques-introduction.htm>
- [8] <https://fr.wikipedia.org/wiki/Capteur>
- [9] <https://www.carnetdumaker.net/articles/communiquer-sans-fil-en-433mhz>
- [10] Hassina ait issad et liliaa terki. déploiement d'un réseau de capteur sans fil pour la gestion de l'éclairage public, université Mouloud Mammeri de Tizi-Ouzou , 2016
- [11] Hamadi Lyes et Harbane Fazia .conception et réalisation d'un dispositif de surveillance-protection contre l'inondation et l'incendie, université Mouloud Mammeri de Tizi-Ouzou, 2016
- [12] Erik Bartmann.le grand livre arduino,Eyrolles.2015
- [13] Simon Landrault Hippolyte Weisslinger.Premier pas en informatique embarqué, 2014
- [14] Christian Tavernier.Arduino. Maitrisez sa programmation et ses cartes d'interface.DUNOD ,2014.
- [15] Simon Monk. 30 Arduino Projects for the Evil Genius.Mc Graw hill,2014.
- [16] Bertrand cotenceau. Carte arduino microcontrôleur AIMega328.2014.