

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

UNIVERSITE MOULOUD MAMMERI DE TIZI-OUZOU  
FACULTE DE GENIE ELECTRIQUE ET INFORMATIQUE  
DEPARTEMENT DE L'INFORMATIQUE



## Mémoire de fin d'études en Master académique



Domaine : Mathématique et informatique

Filière : Informatique.

Spécialité : Systèmes informatiques.

**Réalisé par:**

*Melle : Hammadache Fadhila*

*Melle: Khames Imane*

### **Thème**

*Conception et réalisation d'un système de  
protection contre les incendies et fuites de gaz  
domestique.*

Mémoire soutenu publiquement le 13/07/2017 devant le jury composé de :

**Présidente : M<sup>me</sup> R.AOUDJIT.**

**Promoteur : M<sup>r</sup> S.TALBI.**

**Examineur : M<sup>r</sup> I.FILLALI.**

**Examineur : M<sup>r</sup> MS.HABET.**

2016/2017

# *Remerciements*

*Au terme de ce travail, nous tenons à  
exprimer nos vifs remerciements à  
notre promoteur Mr S.TALBI  
de nous avoir accompagnées durant notre  
travail.*

*Nous remercions également Mr  
SAADANI, Mr DAOUI, Mr HAMDANI,  
et Mme AOUDJIT pour leurs accueils  
chaque fois que nous avons sollicité leurs  
aides, ainsi que pour leurs multiples  
conseils.*

*Nous profitons aussi de ce mémoire pour  
exprimer nos plus vifs remerciements  
envers tous les professeurs qui nous ont  
apportés du soutien durant nos études et  
envers tous nos amis qui ont été toujours  
près de nous avec leurs encouragements,  
critiques et conseils.*

*En fin, aux membres de jury qui nous font  
l'honneur d'examiner notre modeste  
travail.*

# Dédicace

*Je dédie ce modeste travail à toutes les personnes qui me sont chères, en particulier :*

- A mes très chers parents qui m'ont toujours soutenu, et font de leurs mieux pour que je sois un être meilleur.*
- A mes frères : Idir, Mohand.*
- A mes sœurs : Radia, Nacera, Karima, Kahina, Lynda.*
- A mes deux plus belle copines Fatma et Malha, ainsi leurs familles.*
- A tous mes amis (es) et en particulier N.Rafik ainsi sa famille.*
- A ma binôme Imane ainsi sa famille, Et tous ceux qui me sont chers au monde.*

*H. Fadhlila*

# *Dédicace*

*Je dédie ce modeste travail :*

*A mes chers parents qui ont tant  
donné.*

*Pour leur immense soutien, leur grand  
amour, leurs sacrifices et leurs prières.*

*A mes frères et sœurs et leurs petites  
familles.*

*A mon binôme Fadhlila, ainsi que sa  
famille.*

*A mon très cher mari qui a toujours  
était à mes cotés.*

*A ma belle famille*

*Et tous ceux qui me sont chers au  
monde.*

*K.Imane*

# Sommaire

**Liste des figures.**

**Liste des tables.**

**Introduction générale.**

**Table des matières**

## **Chapitre 1 : Systèmes embarqués et Domotique**

1.1. Introduction .....	6
1.2. Les systèmes embarqués.....	6
1.2.1. Définition.....	6
1.2.2. Historique .....	6
1.2.3. Architecture d'un système embarqué .....	6
1.2.4. Classification des systèmes embarqués .....	8
1. Système transformationnel.....	8
2. Système réactifs.....	8
3. Système interactif.....	8
4. Système temps réel.....	8
1.2.5. Contraintes.....	9
1. Fonctionnement en temps réel.....	9
2. Faible encombrement, poids et consommation .....	9
3. Cout, sureté et sécurité .....	9
1.2.6. Domaines d'application .....	9
1.2.7. Caractéristiques .....	10
1.2.8. Structures de base des systèmes embarqués .....	10
1.3. La domotique.....	11
1.3.1. Définition .....	11
1.3.2. Historique .....	11
1.3.3. Objectifs de la domotique.....	12
1.3.4. Quelques applications dédiées à la domotique.....	12
1.3.5. Les composants principaux . .....	14
1. Le module émetteur.....	14
2. Le module récepteur.....	15
3. Le module détecteur .....	15

1.3.6. Les fonctions de la domotique .....	16
1. La fonction de sécurité .....	16
2. La fonction de surveillance .....	16
3. La fonction de gestion d'énergie .....	17
4. La fonction de scénarisation .....	17
5. La fonction de communication .....	17
6. La fonction de confort .....	17
1.3.7. Les modes de communication .....	18
1. Transmission filaire .....	18
2. Le mode de communication CPL .....	19
3. Transmission sans fil .....	20
1.3.8. Fonctionnement de la domotique .....	21
1. Les commandes .....	21
2. Les actionneurs .....	21
3. Les capteurs .....	21
I.4. Conclusion .....	23

## **Chapitre 2 : Généralités sur les systèmes Arduino**

2.1. Introduction .....	25
2.2. Présentation générale des cartes Arduino .....	25
2.3. Quelques cartes Arduino .....	25
2.3.1. La description de la carte Arduino UNO .....	27
1. Microcontrôleur ATmega328 .....	28
2. Les sources d'alimentation de la carte .....	30
3. Les entrées/sorties .....	31
4. Les ports de communications .....	32
2.4. L'environnement de développement : .....	33
2.4.1. IDE Arduino .....	33
2.4.2. Structure générale d'un programme Arduino .....	33
2.4.3. Téléverser un programme dans la carte .....	35
2.5. Les shields de la carte Arduino .....	37
2.5.1. Les shields de communications .....	37
1. Le shield Bluetooth .....	37
2. Le shield wifi .....	38
3. Les shields XBee .....	38
2.5.2. Les capteurs .....	39
2.5.3. Les drivers .....	39

1. Afficheur LCD .....	39
2. Le shield SD CARD .....	39
3. Le relais .....	40
2.6. Conclusion .....	41

## **Chapitre 3 : Conception du système**

3.1. Introduction .....	43
3.2. Objectifs de système .....	43
3.3. Architecture générale du système .....	43
3.4. Comportement prévu du système .....	44
3.5. Conception matériel .....	45
3.5.1. Les capteurs .....	45
1. Les capteurs analogiques numériques .....	45
2. Capteur de qualité de l'air MQ135 .....	46
3. Le capteur de température et d'humidité SHT75 .....	48
3.5.2. Le contrôleur .....	49
3.5.3. Les actionneurs .....	50
1. Buzzer .....	50
2. Les leds .....	51
3. Le module de communication GSM SIM900 .....	52
3.5.4. Le relais .....	55
3.6. Conception logicielle .....	57
3.6.1. Fonctionnalités de base .....	57
1. Fonctions destinées aux broches .....	57
2. Fonction de gestion du temps .....	58
3.6.2. Fritzing .....	58
3.6.3. Organigramme de fonctionnement .....	59
3.7. Conclusion .....	62

## **Chapitre 4 : Réalisation du système**

4.1. Introduction .....	65
4.2. Outils de développements .....	65
4.2.1. Matériels .....	65
4.2.2. Logiciels .....	67
1. Partie Arduino .....	67
2. Partie Processing .....	70
3. Processing et Arduino .....	72

4.3. Test du système .....	74
4.3.2. Test de capture de fuite de gaz .....	75
4.3.3. Test de capture d'incendie .....	76
4.4. Conclusion.....	77

## **Conclusion générale et perspectives.**

## **Bibliographie.**



## Liste des figures

Figure1.1: Architecture d'un système embarqué. ....	7
Figure1.2: Classification des systèmes embarqués. ....	8
Figure1.3: L'interaction d'un système embarqué avec son environnement. ....	10
Figure1.4: Exemples des services offerts par la domotique .....	11
Figure1.5: Exemple d'alarme. ....	13
Figure1.6: Exemple de store. ....	13
Figure1.7: Exemple de porte de garage. ....	13
Figure1.8: Exemple de chauffage. ....	14
Figure1.9: Box domotique. ....	14
Figure1.10: Exemple d'actionneur. ....	15
Figure1.11: Principaux capteurs utilisés dans la domotique. ....	15
Figure1.12: Services offerts par la domotique. ....	16
Figure1.13: Technologie bus filaire. ....	19
Figure1.14: Exemple de communication CPL. ....	20
Figure1.15: Fonctionnement de la domotique (1). ....	21
Figure1.16: Fonctionnement de la domotique(2). ....	22
Figure2.1: Carte Arduino Mega 2860. ....	26
Figure2.2: Carte Arduino Due. ....	26
Figure2.3: Carte Arduino NANO. ....	27
Figure2.4: Carte Arduino Uno. ....	28
Figure2.5: Microcontrôleur ATmega 328. ....	29
Figure2.6: Alimentation de la carte Arduino .....	31
Figure2.7: Différents parties de la fenêtre principale de logiciel Arduino. ....	33
Figure2.8 : Fonction setup et loop. ....	34
Figure2.9: Fonctionnement du la structure de base du programme. ....	35
Figure2.10: Choix de la carte. ....	35
Figure2.11: Choix du port. ....	36
Figure2.12: Compilation et téléversement du programme Arduino. ....	37
Figure2.13: Trois types du module bluetooth. ....	38
Figure2.14: Shield wifi. ....	38
Figure2.15: Shield XBee. ....	38
Figure2.16: Capteur ultrason et capteur de température .....	39

Figure2.17: Afficheur LCD. ....	39
Figure2.18: Carte SD. ....	40
Figure2.19: Relais. ....	40
Figure3.1: Architecture générale du système. ....	44
Figure3.2: Concept de communication lors d'une fuite de gaz. ....	44
Figure3.3: Concept de communication lors d'incendie. ....	45
Figure3.4: Capteur MQ135. ....	46
Figure3.5: Montage du capteur MQ135. ....	47
Figure3.6: Branchement de MQ135 avec la carte Arduino. ....	47
Figure3.7: Capteur SHT75. ....	48
Figure3.8: Bbranchement du capteur SHT75 avec la carte Arduino. ....	49
Figure3.9: Buzzer. ....	50
Figure3.10: Branchement du buzzer avec la carte Arduino. ....	51
Figure3.11: Leds. ....	51
Figure3.12: Branchement des leds avec la carte Arduino. ....	52
Figure3.13:Module de communication GSM SIM900. ....	53
Figure3.14: Vue de dessous du GSM SIM900. ....	53
Figure3.15: Vue de dessus du GSM SIM900. ....	54
Figure3.16: Branchement du GSM SIM900 avec la carte Arduino. ....	55
Figure 3.17: Exemple de relais. ....	56
Figure 3.18: Symbole du relais. ....	56
Figure 3.19: Exemple explicatif. ....	58
Figure 3.20: L'interface Fritzing. ....	59
Figure3.21: Organigramme de fonctionnement. ....	60
Figure4.1: Schéma globale du système ....	67
Figure4.2: Fonction temperature(). ....	68
Figure4.3: Fonction saveTMP(). ....	69
Figure4.4: Fonction CalcDIF_TMP(). ....	69
Figure4.5: Fonction alarme(). ....	70
Figure4.6: Fonction envoyer_sms(). ....	70
Figure4.7: L'interface principale et l'interface de visualisation Processing. ....	71
Figure4.8: Système actif. ....	75
Figure4.9: Teste de capture de fuite de gaz. ....	75
Figure4.10: Teste de capture de feu. ....	76

## Liste des tables

Tableau2.1: Caractéristiques des cartes Arduino Mega 2560,Due et Nano .....	27
Tableau2.2: Caractéristiques de la carte Arduino UNO .....	28
Tableau4.1: Composants utilisés. ....	65

# **Introduction générale**

Tout le monde rêve de pouvoir automatiser plusieurs fonctions de sa maison (sécurité, confort, communication,...). Il s'est ainsi développé un domaine particulier de l'automatique dédié à la l'usage de la maison appelé domotique.

L'homme veut maîtriser l'environnement et la nature par la technique. Il veut plier la nature à ses désirs, plutôt que de s'y adapter. Et de ce point de vue, la technologie représente le pouvoir. Ce désir de domination s'étend à nos habitations : nous voulons des maisons de plus en plus à notre service, des maisons que nous pouvons contrôler, et c'est dans ce contexte que la domotique s'inscrit.

Le concept de domotique fait l'objet de nombreuses définitions. De la maison intelligente à la maison communicante, toutes les gradations sont possibles. De fait, la domotique concerne l'application à l'habitat de toutes les technologies dites nouvelles pouvant s'intégrer à ce contexte.

La domotique est l'ensemble des techniques et technologies des systèmes embarqués qui facilitent la vie quotidienne et permettent une gestion automatisée des fonctions électriques de l'habitation. Ce domaine concerne principalement les appareils électriques que l'on utilise dans le confort, la sécurité, la gestion de l'énergie.

Elle permet de mettre une maison ou un local sous une surveillance permanente, et une communication de l'information à distance et en temps réel.

## 1. Problématique

L'incendie domestique constitue un véritable problème de santé publique avec un bilan inquiétant dont les conséquences finales sont fréquemment tragiques : S'il ne tue pas, l'incendie provoque chez les victimes de très sérieuses séquelles physiques, traumatiques, respiratoires et psychologiques.

- Tous les ans, près de 10 000 individus sont blessés lors d'un incendie domestique.
- Tous les ans, plus de 750 individus meurent à cause d'un incendie domestique.
- 70% des incendies domestiques se déclarent en pleine journée cependant 70% des incendies mortels se produisent durant la nuit.
- Lors d'un incendie, 80% des victimes décèdent intoxiquées à cause des fumées.
- En moyenne, un incendie domestique sur quatre est causé par une installation électrique déficiente.
- Le feu est le premier motif de mortalité chez les enfants âgés de moins de 5 ans.
- Tous les ans, plus de 250 000 incendies domestiques menaçant sont signalé ; ce qui correspond une intervention des sapeurs-pompiers toutes les 7 minutes environ.

Multiplés sont les foyers qui utilisent encore quotidiennement le gaz naturel comme source d'énergie afin d'alimenter leur chaudière et obtenir du chauffage et de l'eau chaude sanitaire en dépit des risques qu'il représente. Cette source d'énergie fossile constitue un danger bien établi et avec des conséquences finales lourdes et diverses tel que l'asphyxie, incendie, brûlures, explosion, etc.

# Introduction générale

---

Les dégâts sont de genre d'intoxication au monoxyde de carbone causé par une fuite de gaz. En sachant que 30% des cas d'incendies, se déclarent pendant la nuit et 70% d'entre eux sont mortels, étant donné que les victimes n'ont pas le temps de réagir. La fuite est indétectable, le monoxyde de carbone (CO) étant inodore et invisible ; il opère à la manière d'un gaz asphyxiant extrêmement toxique, ce qui fait de lui la principale cause de mort toxique accidentelle en milieu domestique. Ce danger qui fait l'inquiétude des habitants, soient à leurs présence à l'intérieur ou lorsque le quitter en partant travailler.

## 2. Solution

C'est dans ce cadre que se situe notre projet de fin d'études intitulé « Conception et réalisation d'un système de protection contre les incendies et fuites de gaz domestiques ».

Ce projet a pour objectif de développer un système embarqué ou une application en utilisant la technologie Arduino. Cette application permet de surveiller en temps réel une maison contre les incendies, et fuites de gaz.

Ainsi, pour mener à bien notre projet et venir à bout des objectifs fixés, nous avons structuré notre mémoire en quatre chapitres comme suit :

- Le premier chapitre « Systèmes embarqués et Domotique » permettra en premier lieu de présenter les systèmes embarqués pour passer en suite à la présentation de la domotique.
- Le deuxième chapitre « Généralités sur les systèmes Arduino » permettra de présenter la carte Arduino, en se focalisant sur le modèle UNO ainsi que son IDE.
- Le troisième chapitre « Conception du système » est consacré à la description des différents outils, logiciels et matériels utilisés dans notre réalisation, ainsi l'architecture générale de notre système.
- Le quatrième chapitre « Réalisation du système » qui comme son nom l'indique, présentera le fruit de notre travail. En premier lieu, nous allons présenter les outils de développement matériels et logiciel, puis le comportement de notre système en présentant des captures d'écran témoignant ses différents comportements.
- En fin, en termine par une conclusion générale et des perspectives.

The title is enclosed in large, light green square brackets that have rounded corners. The text is centered within these brackets.

# **Chapitre 1**

## **Systemes embarqués et Domotique**

## **1.1. Introduction**

Nous présenterons dans ce chapitre une description du système embarqué et une autre pour la domotique. D'une part nous avons donné une définition pour les systèmes embarqués, son historique, architecture, ses caractéristiques et contraintes, puis on donnera un aperçu sur le domaine d'application de ce système, et ces structures de base. D'autre part, nous présenterons la domotique, donner sa définition, son historique, ses objectifs, composants principaux, et les fonctions de la domotique. Et nous terminerons par cité les modes de communication, et le fonctionnement de la domotique.

## **1.2. Les systèmes embarqués**

### **1.2.1. Définition**

Un système embarqué est un système complexe qui intègre du matériel et du logiciel, conçu 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 stocké dans des mémoires. Le système matériel et application (logiciel) sont intimement liés et immergés dans le matériel et ne sont pas aussi facilement discernables comme dans un environnement de travail classique de type ordinateur de bureau.

Un système embarqué est autonome, et ne possède pas des entrées/sorties standards tels qu'un clavier ou un écran ordinateur.

Système embarqué = Système électronique/informatique conçu pour réaliser une ou plusieurs tâches précises (contrôle, surveillance, communication,...) <sup>[1]</sup>.

### **1.2.2. Historique**

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, contrôleur d'entrées/sorties.

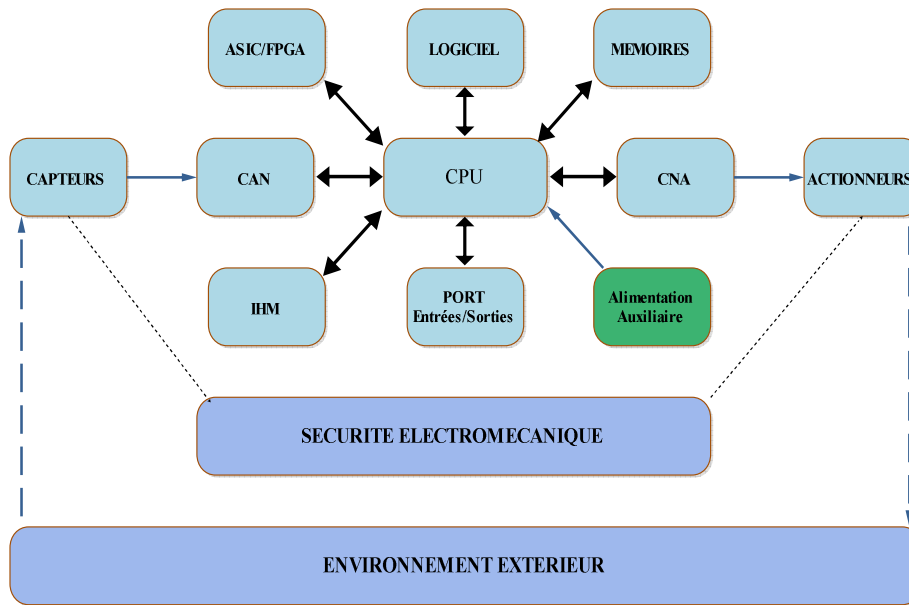
Alors qu'il fallait auparavant plusieurs circuits intégrés différents, chacun dédié à une tâche particulière, un seul microprocesseur pouvait assurer autant de travaux différents que possible. Très rapidement, des objets quotidiens tels que fours à micro-ondes, télévisions et automobiles à moteur à injection électronique ne tardèrent pas à être équipés de microprocesseurs. Ce sont alors les débuts de l'informatique embarquée <sup>[2]</sup>.

### **1.2.3. Architecture d'un système embarqué**

Un système électronique embarqué ou enfoui est un élément constitutif d'un système plus complexe pour lequel il rend des services bien précis (contrôle, surveillance, communication...). Il est constitué de parties matérielles et logicielles qui sont conçues spécifiquement pour réaliser une fonction dédiée <sup>[3]</sup>.



L'architecture d'un système embarqué est résumée par le schéma suivant:



**Figure1.1: Architecture d'un système embarqué [3].**

**CAN :** Convertisseur Analogique Numérique.

**CNA :** Convertisseur Numérique Analogique.

**FPGA:** Field Programmable Gate Array (un circuit logique programmable).

**UC :** unité centrale formé de :

- CPU : microprocesseur(s) ou des microcontrôleurs,
- RAM : mémoire centrale.

**Entrées :**

- Les capteurs/convertisseurs (pression, audio, température...)
- Le clavier, boutons poussoirs ou télécommandes (infrarouge, Bluetooth, radio...).
- Les lecteurs de tags ou de codes-barres.

**Sorties :**

- Les écrans et afficheurs LCD.
- Le Système d'alarme ou synthèse vocale.
- L'imprimante en tous genres comme papier, étiquettes, photos...

**IHM:** C'est un dispositif qui sert à communiquer entre l'humain et la machine.

Un exemple réaliste d'IHM est l'écran avec les dispositifs «TouchScreen».

Cette architecture peut varier selon les systèmes: on peut par exemple, ne pas trouver de systèmes auxiliaires dans de nombreux systèmes embarqués autonomes et indépendants. En revanche, l'architecture de base est la plupart du temps composée d'une unité centrale de traitement (CPU), d'un système d'exploitation qui réside parfois uniquement en un logiciel

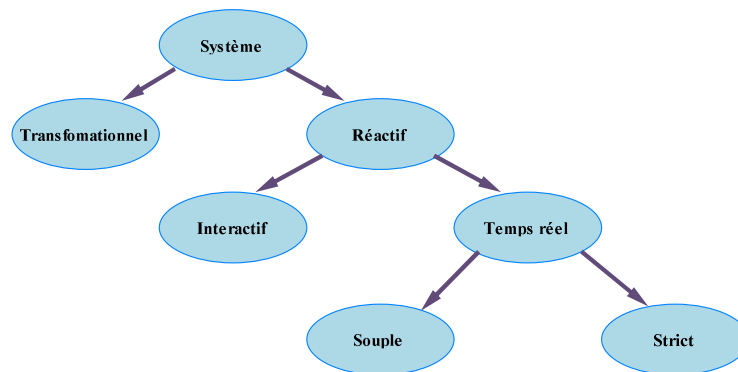
spécifique (ex: routeur), ou une boucle d'exécution. De même l'interface IHM n'est pas souvent existante, mais est souvent utile pour reconfigurer le système ou vérifier son comportement.

Le fonctionnement d'un système embarqué est définie sous formes d'opérations chainés qui sont :

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

## 1.2.4. Classification des systèmes embarqués

La classification des systèmes embarqués est résumée par le schéma suivant <sup>[4]</sup> :



**Figure1.2: Classification des systèmes embarqués <sup>[4]</sup>.**

**1. Système transformationnel** : produit des sorties en fonction d'entrées selon un processus de calcul indépendant de l'environnement.

**2. Système réactifs** : est un système ouvert répondant constamment aux sollicitations de son environnement en produisant des actions sur celui-ci :

- Par opposition : systèmes transformationnels (indexation d'une base de données).
- Abstraction : système qui tourne à l'infini.
- Vie du système divisée en modes d'exécution (sol, décollage, croisière, ...).

Tout système réactif est au minimum un système temps réel souple.

**3. Système interactif** : réagit aux stimuli en entrée, et à sa vitesse propre.

- Les résultats dépendent de données produites par l'environnement.
- Les instants de production respectent des valeurs statiques.

**4. Système temps réel** : système dont les entrées et les sorties sont soumises à des contraintes temporelles, en plus des contraintes de correction fonctionnelles classiques, autrement dit, un

un système temps réel est un système réactif devant fournir des sorties logiquement correctes tout en respectant strictement des contraintes temporelles explicites. La plupart des systèmes dits "temps réel" comportent à la fois des fonctions temps réel durs et des fonctions temps réel souples.

### 1.2.5. Contraintes

Le système embarqué possède des contraintes particulières par rapport aux autres systèmes électroniques [5].

#### 1. 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 plupart des systèmes sont « multirate » (traitement d'informations à différents rythmes).

#### 2. Faible encombrement, poids et consommation

- Consommation électrique minimisée ;
- Difficulté de packaging (analogique, numérique, et RF) ;
- Batterie de 8 heures et plus (température, vibrations, variations d'alimentation, interférences RF, corrosion, eau, feu, radiations, etc.).

#### 3. Coût, sûreté et sécurité

- Le système doit toujours fonctionner correctement (faible coût et une redondance minimale) ;
- Sûreté de fonctionnement du logiciel (système opérationnel même quand un composant électrique lâche).

### 1.2.6. Domaines d'application

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

- Domotique : la sécurité, le confort, la communication, gestion de chauffage,
- Transport : automobile, aéronautique (avionique), etc.
- Astronautique : fusée, satellite artificiel, sonde spatiale, etc.
- Militaires : missile.
- Télécommunication : Set-top box, téléphonie, routeur, pare-feu, serveur de temps, téléphone portable, etc.
- Electroménager : télévision, four à micro-ondes.
- Impression : imprimante multifonctions, photocopieur, etc.
- Informatiques : disque dur, lecteur de disquette, etc.
- Multimédia : console de jeux vidéo, assistant personnel.

### 1.2.7. Caractéristiques

Le système embarqué possède d'un ensemble de caractéristiques qui sont les suivantes <sup>[1]</sup> :

- Encombrement mémoire (mémoire limitée, pas de disque en général);
- Consommation d'énergie (batterie : point faible des SE) ;
- Poids et volume ;
- Autonomie ;
- Mobilité ;
- Communication (attention : la communication affecte la batterie) ;
- Contraintes de temps réel ;
- Contraintes de sécurité ;
- Coût de produits en relation avec le secteur cible.

### 1.2.8. Structures de base des systèmes embarqués

Les systèmes embarqués interagissent avec leurs environnement pour le quelle ils rendent des services bien précis (contrôle, surveillance, communication,...). Une information est captée par un capteur, une transformation est réalisée sur cette information par le convertisseur analogique numérique (CAN), avant d'être lisible par le cœur du système embarqué (constitué d'une partie matérielle et une partie logiciel), qui effectue un traitement spécifique à cette information pour rendre à son tour une réponse à son environnement, cette réponse doit être transformée par convertisseur numérique analogique (CNA) avant d'être envoyée à l'environnement via un actionneur <sup>[4]</sup>.

La figure ci-dessous résume l'interaction d'un système embarqué avec son environnement :

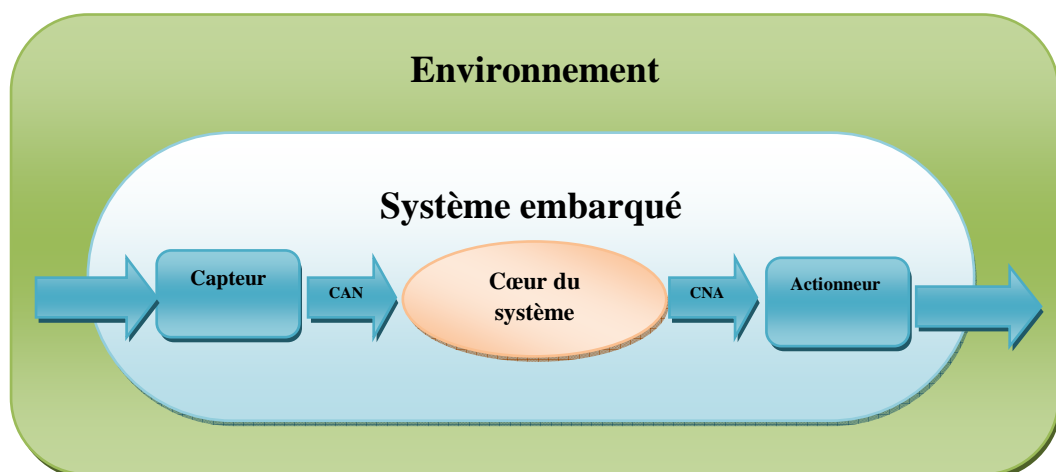


Figure1.3: L'interaction d'un système embarqué avec son environnement <sup>[4]</sup>.

## 1.3. La domotique

### 1.3.1. Définition

Le terme domotique est une contraction des mots "domus" (maison en latin) et "informatique". La domotique regroupe l'ensemble des technologies permettant l'automatisation des équipements de l'habitat. En combinant et en automatisant des fonctions d'éclairage, de chauffage, de volets roulants, d'alarme... la domotique contribue à simplifier la vie de ses habitants et apporte un certain niveau de confort et de sécurité.

Cette technologie s'articule aujourd'hui autour d'un ensemble de solutions simples, pratiques et modulables, permettant d'automatiser les gestes quotidiens en fonction des besoins et des attentes de l'utilisateur final [6].



**Figure1.4: Exemples des services offerts par la domotique [6].**

### 1.3. 2. Historique

#### – Les années 1980, la domotique, signe extérieur de la richesse

Soutenue par une exigence accrue des ménages en termes de confort, la domotique fait véritablement son apparition dans les années 80. Les avancées dans les domaines de l'électronique (miniaturisation des composants), de l'informatique (arrivée des premiers ordinateurs dans les foyers), et de la communication (numérisation des réseaux, minitel) sont à l'origine de cette nouvelle technologie.

#### – Les années 1990, la domotique, en progrès constant

La maison en 1990 est automatisée, la télécommunication commence à se développer. Les réseaux informatiques deviennent sans fil grâce au Wifi. Certaines actions sont planifiées

(ouverture des volets à une heure donnée, allumage des lumières quand on passe à proximité ou quand le soleil se couche...), les télécommande remplacent les interrupteurs, et les alarmes s'associent à la télésurveillance.

### – Les années 2000 à ce moment, la domotique est devenue grande !

Le secteur de la domotique ne cesse de croître depuis 2000. La maison est désormais pilotable à distance. Avec le développement d'Internet, les automatismes sont programmés et commandés via un appareil mobile (Smartphone, tablette, ordinateur). La maison est obéissante et les solutions 'sur-mesure' permettent de proposer un écosystème de produits parfaitement adaptés aux besoins du client. En d'autres termes, la domotique se démocratise, aidés par les professionnels du bâtiment (architectes, artisans, thermiciens, etc.) qui sont de plus en plus nombreux à se former pour offrir conseils et compétences à leurs clientèles.

### 1.3.3. Objectifs de la domotique

- Vivre plus confortablement grâce aux nouvelles technologies, en ayant accès à des fonctionnalités simples et pratiques pour piloter vos appareils à distance.
- Réaliser des économies d'énergies en ne consommant que le strict nécessaire en matière d'énergie.
- Piloter vos appareils sans avoir besoin de vous déplacer (un avantage considérable pour les personnes en situation de dépendance ou de handicap).
- Corriger les oublis grâce à la possibilité d'éteindre une lumière ou de verrouiller la maison à distance.
- Anticiper en planifiant certaines tâches à distance (mise en route du lave-linge, contrôler la température du chauffage, etc.).
- Sécuriser votre habitation avec la caméra de surveillance ou le système d'alarme notamment.

### 1.3.4. Quelques applications dédiées à la domotique

#### – Une alarme

L'alarme est un dispositif de surveillance qui permet de signaler un événement comme par exemple une intrusion dans une maison, un incendie, la chute d'une personne dans une piscine, Pour «voir» l'évènement, elle utilise différents capteurs comme un capteur de mouvement, un capteur volumétrique, un capteur infrarouge, un capteur de température, un capteur de pression.



**Figure1.5: Exemple d'alarme** <sup>[28]</sup>.

### – Les stores

À l'aide d'un capteur de luminosité, votre store sort ou rentre automatiquement en fonction du soleil. Pour éviter tout risque, à l'aide d'un capteur de vent, votre store rentre tout seul s'il y a trop de vent.



**Figure1.6. Exemple de store** <sup>[28]</sup>.

### – Porte de garage

Pas envie de sortir de la voiture pour ouvrir la porte du garage ? Perte de temps, vous gênez avec votre voiture au milieu de la rue ? Télécommandez l'ouverture et fermeture de votre porte de garage à l'aide d'une télécommande.



**Figure1.7. Exemple de porte de garage** <sup>[28]</sup>.

### – Gestion du chauffage

Inutile de chauffer si vous n'êtes pas là ! Programmez vos plages de chauffage et ainsi économisez en énergie. Reliez votre chauffage aux bons capteurs, économisez encore plus.



Figure1.8: Exemple de chauffage <sup>[28]</sup>.

### 1.3. 5. Les composants principaux

La domotique fonctionne comme un circuit sur lequel des composants indispensables (module émetteur, récepteur, et détecteur) prennent place. Un ordinateur central est relié à des modules récepteurs, émetteurs et détecteurs. Cet ordinateur central permet de tout contrôler et de tout paramétrer, non seulement les tâches automatiques (lever un volet roulant ou démarrer la cafetière à une heure précise) mais aussi les actions imprévues. Par exemple, il peut fermer un volet en cas de pluie <sup>[7]</sup>.

Les composants cités précédemment ont également des rôles précis :

**1. Le module émetteur** reçoit et transmet les ordres de l'ordinateur central à module récepteur. Par exemple :

**Une box domotique :** C'est elle qui va centraliser toutes les informations de votre maison et déclenche des actions. On l'appelle toujours le cerveau, car elle est réellement un équipement qui va donner de l'intelligence à votre maison.

Grace aux informations recueillies par les différents capteurs disséminés à travers la maison, va pouvoir déclencher des actions en conséquence. Par exemple, si aucun détecteur de présence dans la maison ne détecte personne, le « cerveau » pourra demander aux ordinateurs de passer en mode éco et aux lampes de s'éteindre. Comme l'illustre la figure suivante :



Figure1.9: Une box domotique <sup>[7]</sup>.



**2. Le module récepteur** se branche sur une prise électrique, reçoit les prises des appareils de la maison et exécute les tâches. Par exemple :

**Un actionneur :** Savoir ce qui se passe, c'est bien, mais pouvoir réaliser des actions en conséquence c'est mieux. Les actionneurs sont donc des périphériques qui vont permettre de piloter des appareils (radiateur, machine à laver, etc.), des lampes, ou encore des automatismes volets, porte de garage, etc.).

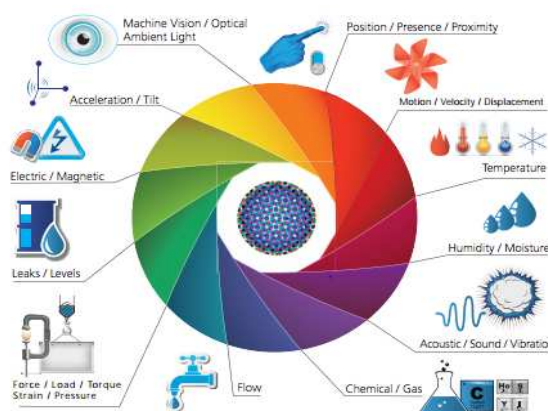


**Figure1.10: Exemple d'actionneur [7].**

**3. Le module détecteur** participe à la sécurité de la maison en détectant les événements inhabituels (intrusion, casse, etc.) et transmet l'information à l'ordinateur central. Par exemple :

**Un capteur :** Un capteur est un dispositif transformant l'état d'une grandeur physique observée en une grandeur utilisable, telle qu'une tension électrique, une hauteur de mercure, une intensité ou la déviation d'une aiguille.

Une box domotique sans vue, sans odorat ne sert pas à grand-chose. Les capteurs sont des périphériques qui vont donner des sens à votre maison. Relever la température, le niveau de CO2, détecté de la fumée, une fuite de gaz, etc. Grâce à eux la maison saura tout ce qui se passe.

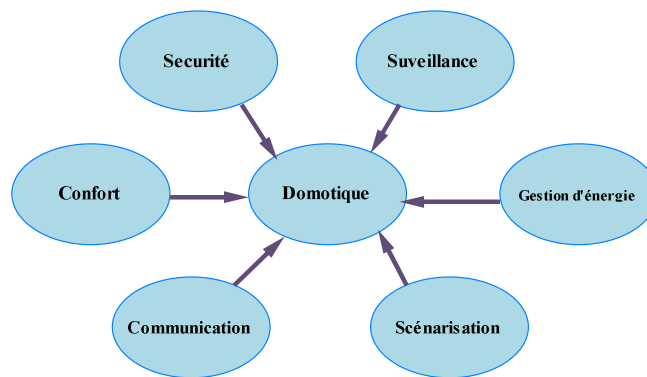


**Figure1.11: Principaux capteurs utilisés dans la domotique [7].**

L'ensemble des composants de la domotique participent à la programmation et à l'automatisation des tâches et fonctionne grâce à un réseau.

### 1.3.6. Fonctions de la domotique

Les fonctions suivantes peuvent être réalisées grâce aux technologies intégrées dans la domotique <sup>[8]</sup> :



**Figure1.12: Les services offerts par la domotique <sup>[8]</sup>.**

#### 1. La fonction de 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 :

- Sirène ;
- 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.

#### 2. La fonction de surveillance

Différents capteurs détectent les anomalies :

- Inondation ;
- Incendie ;
- Fuite de gaz ;
- Arrêt du congélateur ;
- Coupure de courant ;
- Vent ou pluie.

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.

### 3. La fonction de gestion d'énergie

La programmation des seuils de température est l'une des principales sources d'économie. La programmation peut être journalière ou hebdomadaire. Des sondes de température renseignent la centrale sur les valeurs de température dans chaque pièce. Elle tient compte de la présence d'un occupant et s'adapte automatiquement à son emploi du temps :

- Extinction des éclairages inutiles ;
- Réglage de l'intensité lumineuse en fonction de l'activité, ...

Un simple appel téléphonique suffit pour augmenter le chauffage en prévision de l'arrivée des occupants.

Le délestage permet la souscription d'un abonnement EDF d'une puissance inférieure et la gestion des différentes formules jour/nuit, (Tempo, EJP, ...)

### 4. La fonction de scénarisation

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 ;
- Vérification de la fermeture de toutes les fenêtres ;
- Allumage de la lumière extérieure durant quelques minutes s'il fait nuit, ...

À partir d'un bouton unique, tous les éclairages du living seront ajustés pour le dîner, une soirée télévision ou la création d'une ambiance lumineuse adaptée l'activité de l'occupant.

### 5. La fonction de communication

Aujourd'hui, une centrale domotique sait communiquer

- Par téléphone;
- Par ordinateur (Internet).

Ceci permet à une personne de recevoir l'état de son installation et d'émettre des alertes et piloter sa maison de n'importe quel endroit du monde, de son bureau ou de sa voiture.

### 6. La fonction de confort

Ouvrir le portail sans descendre de voiture, allumer automatiquement la lumière en ayant les bras chargés de paquets, arroser automatiquement le jardin, ouvrir ou fermer les volets ou les stores, programmer une ambiance sonore dans la maison, disposer de la télévision dans plusieurs pièces, déclencher la préparation du café avant le réveil, adapter la température aux conditions extérieures, ...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.

### 1.3.7. Les modes de communication

Le réseau utilisé pour la domotique constitue un composant indispensable bien qu'il ne soit pas visible concrètement. Plusieurs réseaux peuvent être utilisés pour la communication des appareils <sup>[9]</sup> :

#### 1. Transmission filaire

La méthode filaire consiste à transmettre les données par protocole par les fils aussi appelés bus de commande. Ce genre de transmission est robuste mais nécessite de tirer des fils. Cela implique donc plus de main d'œuvre (téléphone, électricité, Internet, etc.).

Ce genre de solution est à privilégier dans une installation neuve, au moment de l'étude de projet.

L'utilisation d'un mode de communication filaire en domotique aura une implication directe sur la conception de l'installation électrique.

Les solutions filaires sont principalement représentées par la technologie KNX (KONNEX).

Une installation domotique KNX est composée de capteurs et d'actionneurs reliés à un bus de donnée leur permettant de communiquer entre eux. Les capteurs permettent de commander l'installation, ce sont les donneurs d'ordre, comme par exemple, les interrupteurs, les détecteurs ou les mesures. Ces capteurs sont uniquement reliés au bus KNX, et peuvent être alimentés via le bus. Les actionneurs sont les éléments qui reçoivent les ordres et sont commandés par l'installation domotique, c'est par exemple, l'éclairage, le système de chauffage ou les volets. Les actionneurs sont connectés d'une part au bus KNX afin de recevoir les ordres de commande et d'autre part à une alimentation 230V pour alimenter le circuit de puissance.

Comme la montre la figure ci-dessous :

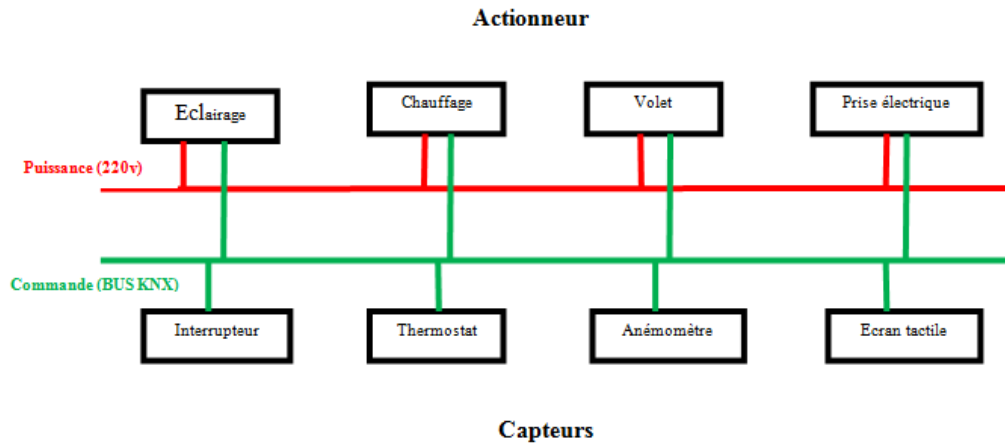


Figure1.13: Technologie bus filaire <sup>[9]</sup>.

Il existe d'autres technologies filaires moins représentatives en domotique résidentielle « contemporaine » et grand public. Citons par exemples :

- 1-Wire ;
- Télé-information (Spécificité Française);
- Fil pilote ;
- Sur IP (en Ethernet) : BacNet, Modbus, xPLsur IP (Protocole) ;
- VDI (Voix Données Images : Réseau résidentiel) qui n'est qu'une architecture convergente sur le support IP des équipements résidentiels

## 2. Le mode de communication CPL

Le CPL pour Courant Porteur en Ligne est un mode de communication alternatif intéressant. Il utilise les fils de l'installation électrique existante comme moyens de communication entre appareils. Chaque appareil connecté sur le réseau électrique compatible CPL peut donc communiquer avec ce protocole.

Le CPL est connu pour ces boitiers permettant de partager le réseau informatique facilement en rénovation.

On distingue généralement trois types de courant porteur en ligne :

**1. Courant porteur domestique** : cette technologie est conçue pour piloter l'éclairage, le chauffage, les automatismes, les prises de courant, et la sécurité (Ex : norme X10, In One de Legrand, X2D).

**2. Courant porteur informatique** : ce type autorise le transport des données informatiques, permettant ainsi de constituer un véritable réseau local reliant ordinateur, imprimante, serveur multimédia, tablettes, etc.

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

La figure ci-dessous illustre un exemple de mode de communication CPL :

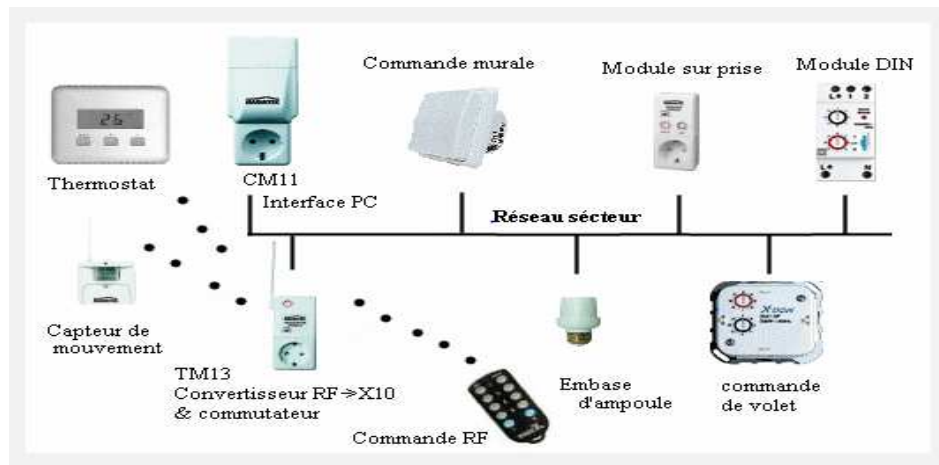


Figure1.14: Exemple de communication CPL <sup>[9]</sup>.

### 3. Transmission sans fil

C'est un mode de communication invisible, l'émetteur et le récepteur communiquent ensemble (les ondes radio, infrarouges, Wi-Fi ou Bluetooth). Ce moyen de communication est intéressant en rénovation car il permet de tirer des fils.

L'utilisation de la domotique sans fils est également très modulable. Voici, les technologies les plus attendues dans la transmission sans fil :

**Zwave** : est probablement la technologie la plus répandue sur le marché grand public et accessible à l'auto-installation de son système. A voir évoluer le marché Zwave et la quantité impressionnante de modules (périphériques), on peut dire que ce standard est à ce jour le leader des systèmes sans-fil. Il faut aussi dire que c'est la technologie la plus plébiscitée.

**Zigbee** : semblait mieux positionné sur le marché initial du grand public, mais à ce jour peu de système résidentiel ont vu le jour. Cette solution offre sur le papier le même niveau de fiabilité et de performances que son frère le Zwave, mais sa visibilité sur le marché est bien inférieure. Zigbee s'appuie sur le standard IEEE 802.15.4 qui offre de belles perspectives à venir.

**RF433** : La radiofréquence 433 MHz était une technologie très répandue par le passé et encore aujourd'hui. Une quantité importante de solutions propriétaires dérivées est présente sur le marché avec comme principale handicap la compatibilité entre constructeurs. On peut considérer que c'est une solution de transition technologique permettant de faire vivre encore quelques temps un écosystème d'équipements existants, comme pour beaucoup les sondes météorologiques.

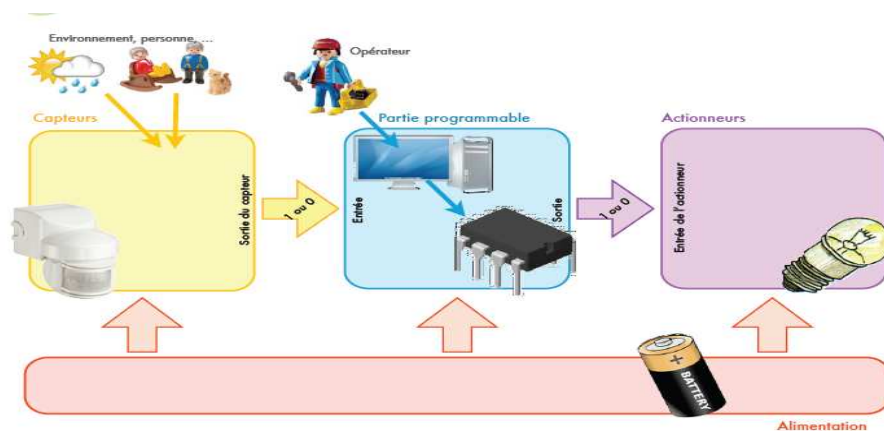
**Communication infrarouge :** Cette technologie est ancienne certes mais encore très usité par nos systèmes multimédia de salon. A moins de réinvestir dans vos équipements Home Cinéma nous utiliserons encore quelques années cette solution.

Tous ces composants sont nécessaires au bon fonctionnement de la domotique.

## 1.3. 8. Fonctionnement de la domotique

Le principe de la domotique est de programmer et contrôler à distance ou localement le comportement d'appareils que l'on aura intégrés dans un réseau de capteurs et d'actionneurs <sup>[10]</sup>.

Le fonctionnement repose principalement sur cette représentation simple :



**Figure1.15: Fonctionnement de la domotique (1)** <sup>[10]</sup>.

Toutes les fonctions communiquent entre elles. Et le centre nerveux devient la visualisation (et donc le contrôle) continue de tout l'ensemble.

**1. Les commandes** vont permettre d'interagir avec l'installation, cela peut être des boutons poussoirs, interrupteurs, des télécommandes, des poignées de portes, des badges RFID, des écrans tactiles, commandes vocales,... Ces commandes vont contrôler un, plusieurs, ou l'ensemble des actionneurs.

**2. Les actionneurs** sont les organes actifs qui vont agir sur les systèmes, comme la coupure ou la variation de l'électricité, le changement de la couleur de l'éclairage, la coupure de l'eau ou du gaz, etc.

**3. Les capteurs** renvoient des informations afin de contrôler les paramètres du bâtiment et des systèmes. Cela peut être la présence ou non de personnes, de fuite d'eau ou de gaz, les températures à différents endroits, la luminosité, taux de moisissure,... Ils peuvent prendre la main directement sur les actionneurs, ou ne renvoyer que l'information à l'utilisateur.

On peut distinguer deux grandes catégories de capteurs :

- **Les capteurs passifs** : ils ont besoin d'une énergie extérieure pour fonctionner. Il est nécessaire de leur appliquer une tension pour obtenir un signal de sortie.
- **Les capteurs actifs** : fonctionnent comme un générateur. Ainsi, l'énergie physique prélevée est directement transformée en grandeur électrique.

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

- **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.

La figure ci-dessous représente une autre méthode de fonctionnement de la domotique :

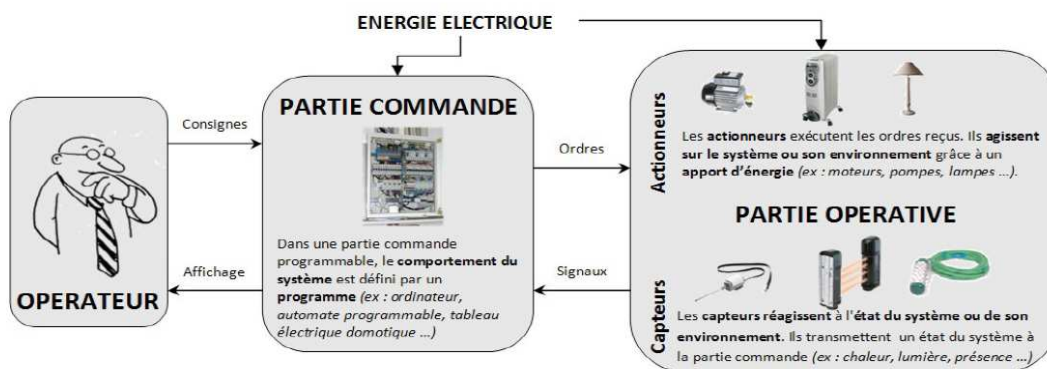


Figure1.16: Fonctionnement de la domotique(2) <sup>[10]</sup>.



**I.4. Conclusion**

Dans ce chapitre, nous avons pu faire une présentation pour le projet et préserver son contexte général en deux parties. Dans la première partie de ce chapitre, nous avons présenté les systèmes embarqués, son architecture, ses domaines d'applications, ainsi ses contraintes, et pour la deuxième partie nous avons fait une étude sur la domotique, donner sa définition, son historique, ses buts, composants principaux, et les fonctions de la domotique. On a terminé par citer les modes de communication, et le fonctionnement de la domotique.

Dans le prochain chapitre nous allons découvrir et présenter la technologie Arduino.

The title is enclosed in large, light green square brackets that have rounded corners. The text is centered within these brackets.

# **Chapitre 2**

## **Généralités sur les systèmes Arduino**

## **2.1. Introduction**

L'Arduino connaît aujourd'hui un grand succès planétaire, pourtant il ne s'agissait au départ que d'une simple carte électronique de petit format supportant un microcontrôleur entouré de composants nécessaires à son fonctionnement de base jusque-là rien qui puisse expliquer son succès.

En effet, cette carte supporte également, sur sa périphérie une rangée de connecteurs dans lesquels peuvent venir s'enficher directement des cartes d'interface, appelées les « shields ». Ces cartes d'interface couvrent aujourd'hui tout le besoin des applications courantes.

Mais ce n'est pas tout. L'Arduino est également équipé d'un connecteur USB permettant de le raccorder à un micro-ordinateur, PC, utilisé pour écrire le programme destiné à le piloter, mais également pour piloter le microcontrôleur.

L'Arduino permet de réaliser à moindre coût des applications à base de microcontrôleur quasiment sans avoir des connaissances en électronique et sans devoir investir dans des outillages spécialisés. C'est en grande partie à cette conjonction de facteurs que l'Arduino doit son succès.

Dans ce chapitre nous allons présenter la technologie Arduino, en commençant par une présentation générale des cartes Arduino en se focalisant sur le modèle UNO ainsi que son IDE, en finissant par la présentation de quelques cartes d'interfaces.

## **2.2. Présentation générale des cartes Arduino**

L'Arduino est une carte de circuit imprimé supportant un microcontrôleur et la circuiterie minimum nécessaire pour lui permettre de fonctionner, associée à une interface USB. Permettant à partir d'événements détectés par des capteurs, de commander des actionneurs. De manière à effectuer des tâches diverses dans différents domaines comme la domotique (le control des appareils domestiques, l'éclairage, chauffage...), et le pilotage d'un robot, etc. <sup>[11]</sup>.

## **2.3. Quelques cartes Arduino**

La forme, la taille ou les possibilités de connexion jouent un rôle décisif dans le choix de la carte adaptée. C'est pourquoi les développeurs d'Arduino ont mis au point un vaste choix de carte à microcontrôleur afin que chacun trouve le modèle qui répond à ses besoins <sup>[12]</sup>.

Avant de parler sur la carte Arduino Uno utilisé dans notre projet nous citons quelques exemples de cartes Arduino, en présentant les caractéristiques de chaque une :

### – La carte Arduino Mega 2560

La carte Arduino Mega est inévitablement plus grande vu le nombre d'entrées et sorties qu'elle comporte (54), plus de mémoire également (256KO de mémoire flash) ce qui peut être utile pour les programmes complexes.

La carte Mega 2560 est en effet elle aussi équipée d'un microcontrôleur Atmel AVR mais il s'agit cette fois-ci d'un ATmega 2560, d'où le nom donné à la carte. La figure suivante montre un aperçu de la carte Arduino Méga 2560 :

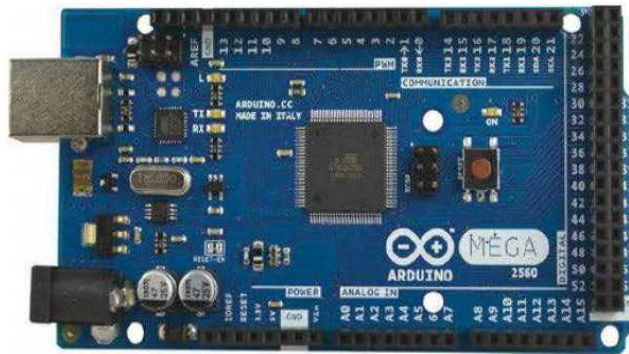


Figure2.1: Carte Arduino Mega 2560 <sup>[14]</sup>.

### – Carte Arduino Due

La carte Arduino Due est une variante de la carte Arduino Mega, c'est la première carte Arduino équipée d'un processeur 32 bits. La fréquence d'horloge de 84 MHz lui permet de réaliser des calculs complexes en un temps record. La figure suivante présente la carte Arduino Due :

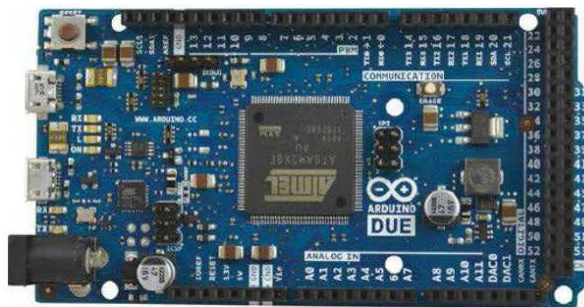
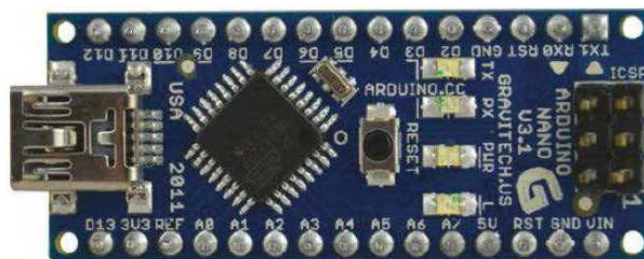


Figure2.2: Carte Arduino Due <sup>[14]</sup>.

### – Carte Arduino Nano

Contrairement à la carte Arduino Uno, cette carte possède des connecteurs au dos qui permettent de l'enficher facilement sur une plaque d'essais, elle propose plus ou moins les mêmes fonctionnalités que l'Uno. En effet cette carte est une Arduino miniaturisée, utile pour les minis hélicoptères et les espaces réduits. La figure ci-dessous montre la carte Arduino Nano :

Figure2.3: Carte Arduino Nano <sup>[14]</sup>.

Le tableau suivant résume les caractéristiques des cartes citées ci-dessus :

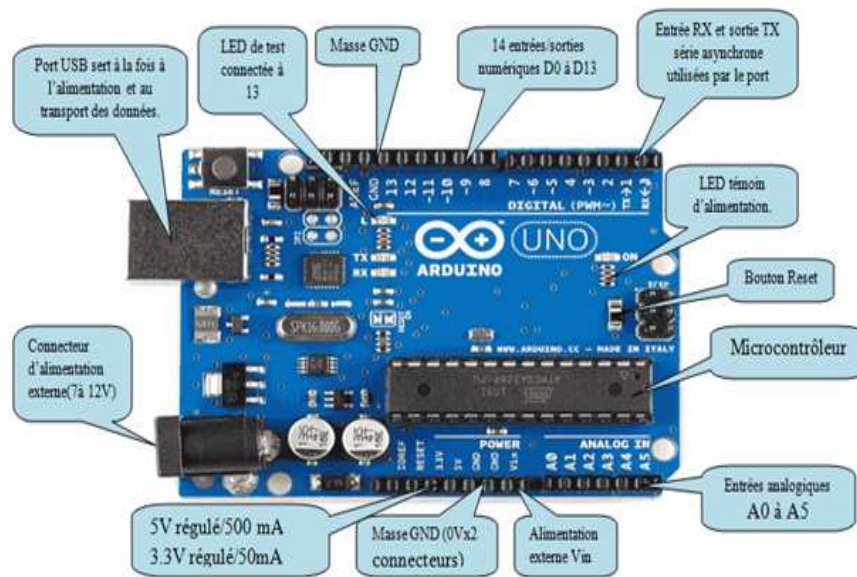
Nom	Carte Arduino Mega 2560	Carte Arduino Due	Carte Arduino Nano	
Microcontrôleur	ATmega2560	AT91SAM3X8E	ATmega168	ATmega 328
SRAM	8KO	96KO	1KO	2KO
EEPROM	4KO		512 octets	1KO
Mémoire flash	256 KO	512 KO	16 KO	32 KO
Entrées/sorties numériques(sorties PWM)	54 E/S (15 PWM)	54 E/S (12 PWM)	14 E/S (6 PWM)	
Entrées analogiques	16	12	8	
Taille	101 mm X 53 mm	101 mm X 53 mm	44 mm X 18 MM	
Tension d'alimentation	7-12 volts	7-12 volts	7-12 volts	
Fréquence d'horloge	16 MHz	84 MHz	16 MHz	

Tableau2.1: Caractéristiques des cartes Arduino Mega 2560, Due et Nano.

### 2.3.1. La description de la carte Arduino UNO

La carte Arduino-UNO est constituée d'un microcontrôleur Atmel AVR, et de composants complémentaires qui facilitent la programmation et l'interfaçage avec d'autres circuits. Il est possible de l'alimenter directement par le port USB, ou par une source externe comprise entre 7 et 12 volts <sup>[13]</sup>.

La figure suivante donne un aperçu sur l'organisation de la carte Arduino UNO :

Figure2.4: Carte Arduino UNO <sup>[14]</sup>.

Le tableau suivant présente les caractéristiques de la carte Arduino :

Carte	Arduino UNO
Microcontrôleur	ATMega
SRAM	2 KO
EEPROM	1KO
Mémoire flash	32 KO
Entrées/Sorties numériques (sorties PWM)	14 (6PWM)
Entrées analogiques	6
Taille	68mm x 53 mm
Tension d'alimentation	7-12 volts
Fréquence d'horloge	16 MHz
Tension de fonctionnement	5 volts

Tableau2.2: Caractéristiques de la carte Arduino UNO <sup>[14]</sup>.

### 1. Microcontrôleur ATMega328

Un microcontrôleur ATMega328 est un circuit intégré qui rassemble sur une puce plusieurs éléments complexes dans un espace réduit. Au temps des pionniers de l'électronique, un grand nombre de composants encombrants ont été soudés sur des cartes plus ou moins grandes.

Aujourd'hui, tout peut être logé dans un petit boîtier en plastique noir muni d'un certain nombre de broches lui permettant d'effectuer la communication <sup>[12]</sup>. La figure ci-dessous montre un microcontrôleur ATMega 328 :



**Figure2.5:Microcontrôleur ATmega328** <sup>[27]</sup>.

Le microcontrôleur ATmega328 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 :

1. **Le processeur** : pour l'exécution des instructions d'un programme.
2. **La mémoire flash** : C'est celle qui contiendra le programme à exécuter. Cette mémoire est effaçable et réinscriptible mémoire programme de 32Ko
3. **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 de 2 ko.
4. **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 1 Ko.
5. **Une interface d'entrées/sorties** : elle permet la communication du Microcontrôleur avec d'autres composants externes.
6. **Le bus d'adresse** : permettant l'accès aux différentes cases mémoires.
7. **Bus de contrôle** : permettant de se positionner en lecture ou en écriture sur ces différentes cases mémoires.
8. **Bus de données** : pour le transit des données de la mémoire vers le processeur et vice-versa.

**2. Les sources d'alimentation de la carte**

La carte Arduino Uno peut être alimentée via la connexion USB ou avec une alimentation externe. La source d'alimentation est automatiquement sélectionnée. Une alimentation externe peut provenir soit d'un adaptateur AC-DC ou d'une batterie. L'adaptateur peut être connecté en branchant une prise 2.1mm dans la prise d'alimentation de la carte ou à partir d'une batterie connectée dans le pin (ou broche) GND et V-in (alimentation externe).

Le processeur peut fonctionner sur une alimentation externe de 6 à 20 volts. Cependant, si la tension est inférieure à 7V, le pin 5V peut fournir moins de cinq volts et le processeur peut devenir instable. Si la tension est supérieure à 12V, le régulateur de tension peut surchauffer et endommager la carte. La plage recommandée est de 7 à 12 volts <sup>[15]</sup>.

Les pins d'alimentation sont les suivantes:

**V-in :** Tension d'entrée à la carte Arduino à l'aide d'une source d'alimentation externe (par opposition à 5 volts de la connexion USB ou une autre source d'alimentation régulée). Si l'alimentation en tension est faite par l'intermédiaire de la prise d'alimentation, on pourra y accéder via ce pin.

**5V :** Ce pin délivre un 5V régulé par la carte. Le processeur peut être alimenté soit à partir de la prise d'alimentation DC (7-12V), le connecteur USB (5V), ou le pin V-in de la carte (7-12). La fourniture d'une tension via les 5V ou 3,3V contourne le régulateur, et peut endommager votre processeur.

**3.3V :** C'est une alimentation de 3,3V fournie par le circuit intégré faisant l'adaptation du signal entre le port USB de notre ordinateur et le port série de la carte. Ceci est intéressant pour certains circuits externes nécessitant cette tension au lieu de 5V. L'intensité maximale disponible sur cette broche est de 50mA.

**GND :** C'est la broche de masse (0V)

**RESET :** Cette broche permet de réinitialiser l'Arduino et donc de redémarrer le programme au début. Pour réinitialiser l'Arduino, il suffit de mettre cette broche à 0V puis de la repasser à 5V, ou 3,3V selon l'Arduino.

**IOREF :** Cette broche est destinée à indiquer aux shields la tension de fonctionnement de l'Arduino. Sur un Arduino 5V, elle aura une tension de 5V et sur un 3,3V une tension de 3,3V



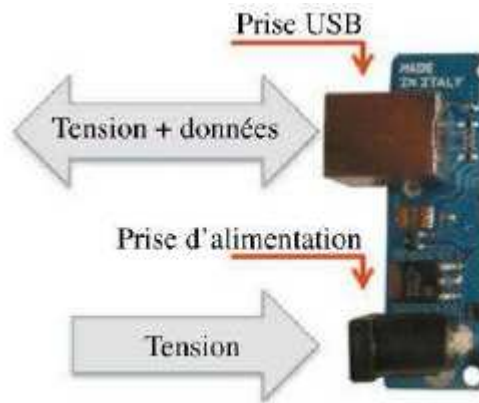


Figure2.1: Alimentation de la carte Arduino <sup>[14]</sup>.

### 3. Les entrées/sorties

Les ports E/S représentent l'interface du microcontrôleur. Il existe plusieurs chemins ou canaux pour échanger les données. Il se produit grâce et par le biais de ces canaux de communications, une interaction avec l'environnement.

Les cartes Arduino utilisent des données provenant de capteurs (de température, de lumière ou d'humidité) pour réagir en conséquence et entreprendre des actions appropriées. Elle peut aussi activer des dispositifs lumineux et sonores, ou agir sur des actionneurs (moteurs et capteurs).

#### - Les entrées/ sorties numériques

Quatorze lignes d'entrées/sorties numériques parallèles, numérotés de 0 à 13, sont disponibles ; chacune d'entre elles pouvant fonctionner en entrée ou en sortie sous le contrôle du programme en utilisant les instructions `pinMode()`, `digitalWrite()` et `digitalRead()` du langage Arduino et ce sens de fonctionnement pouvant même changer de manière dynamique pendant son exécution. Ces broches fonctionnent en 5V. Chaque broche peut fournir ou recevoir un maximum de 40mA d'intensité et dispose d'une résistance interne de 20-50 KOhms. Cette résistance interne s'active sur une broche en entrée à l'aide de l'instruction `digital Write (broche, HIGH)` <sup>[16]</sup>.

En plus, certaines broches ont des fonctions spécialisées :

**Interruptions Externes:** Les Broches 2 et 3 peuvent être configurées pour déclencher une interruption sur une valeur basse, sur un front montant ou descendant, ou sur un changement de valeur.

**Impulsion PWM :** les broches 3, 5, 6, 10 et 11 peuvent être configurées en sorties analogiques, elles sont repérées par un signe ~ sur la carte.

**SPI :** (Interface Série Périphérique) Broches 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). Ces broches supportent la communication SPI disponible avec sa librairie.

Les broches SPI sont également connectées sur le connecteur ICSP qui est mécaniquement compatible avec les cartes Mega.

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

**LED:** Broche 13. Il y a une LED incluse dans la carte connectée à la broche 13. Lorsque la broche est au niveau HAUT, la LED est allumée, et lorsque qu'elle est au niveau BAS, la LED est éteinte.

#### **- Les entrées analogiques**

Il dispose pour cela de six entrées numérotées de A0 à A5, contrairement aux entrées/sorties numériques qui peuvent prendre que deux états HIGH et LOW. Ces entrées peuvent admettre toute tension analogique comprise entre 0 et 5 volts, mais pouvant bien sûr prendre n'importe quelle valeur dans cette plage.

Pour pouvoir être traitées par le microcontrôleur, ces entrées analogiques sont prises en charge par un CAN (Convertisseur Analogique Numérique) dont le rôle est de convertir l'échantillon de tension en une grandeur numérique entre 0 et 1023, et peuvent être configurées comme entrées/sorties numérique <sup>[17]</sup>.

#### **4. Les ports de communications**

La carte Arduino UNO a de nombreuses possibilités de communications avec l'extérieur. L'ATmega328 possède une communication série UART pour communication série de niveau TTL (5V), grâce aux broches numériques 0 (RX) et 1 (TX).

Un circuit intégré ATmega8U2 sur la carte assure la connexion entre cette communication série vers le port USB de l'ordinateur et apparait comme un port COM virtuel pour les logiciels de l'ordinateur. Le code utilisé pour programmer l'ATmega8U2 utilise le driver standard USB COM, et aucun autre driver externe n'est nécessaire.

L'ATmega328 supporte également la communication par les protocoles I2C dit aussi TWI (Two Wire Interface) qui est un bus informatique permettant de relier facilement un microprocesseur de différents circuits et SPI (Serial Peripheral Interface).

Le logiciel Arduino inclut les librairies Wire et SPI qui simplifient l'utilisation du bus I2C et SPI <sup>[18]</sup>.

**UART :** c'est le composant utilisé pour faire la liaison entre l'ordinateur et le port série.

**I2C** ou **TWI :** est un bus informatique, qui permet de relier facilement un microprocesseur et différents circuits.

**SPI :** est un bus de données série synchrone, qui opère en mode Full-duplex.

## 2.4. Environnement de développement

Une telle carte d'acquisition qui se base sur sa construction sur un microcontrôleur doit être dotée d'une interface de programmation comme est le cas de notre carte. L'environnement de programmation open-source pour Arduino peut être téléchargé gratuitement (pour Mac OS X, Windows, et Linux).

### 2.4.1. IDE Arduino

L'IDE Arduino est un logiciel multiplateforme, il offre une interface minimale et épurée pour développer un programme sur les cartes avec un langage propre à lui dont la structure est proche aux langages C et C++. Il est doté d'un éditeur de code avec coloration syntaxique et d'une barre d'outils rapide. Ce sont les deux éléments les plus importants de l'interface, On retrouve aussi une barre de menus plus classique qui est utilisé pour accéder aux fonctions avancées de l'IDE. Enfin, une console affichant les résultats de la compilation du code source, des opérations sur la carte <sup>[19]</sup>. La figure suivante montre à quoi ressemble l'IDE Arduino :

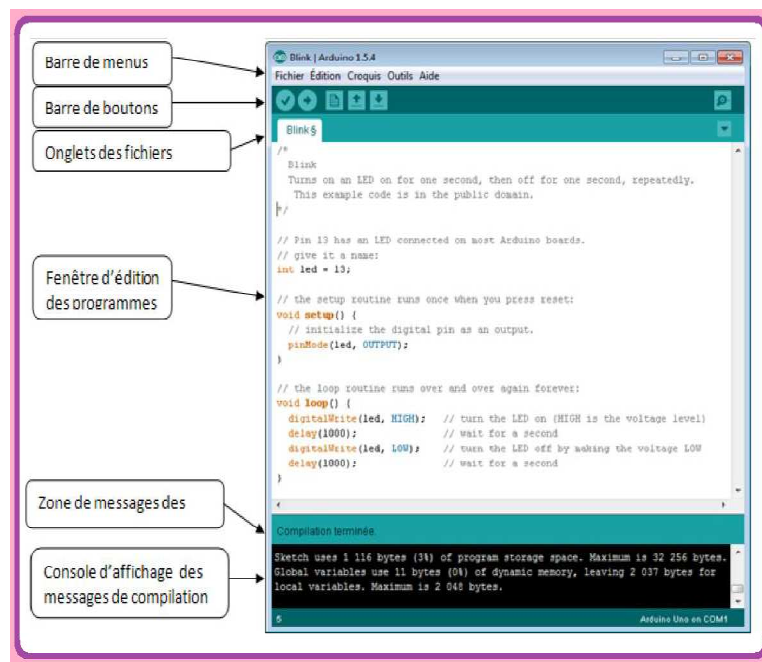


Figure2.7: Les différentes parties de la fenêtre principale de logiciel Arduino.

### 2.4.2. Structure générale d'un programme Arduino

Un programme Arduino est une suite d'instructions élémentaires sous forme textuelle (Ligne par ligne). La carte lit puis effectue les instructions les unes après les autres dans l'ordre défini par les lignes de codes <sup>[15]</sup>.

- **Commentaires**

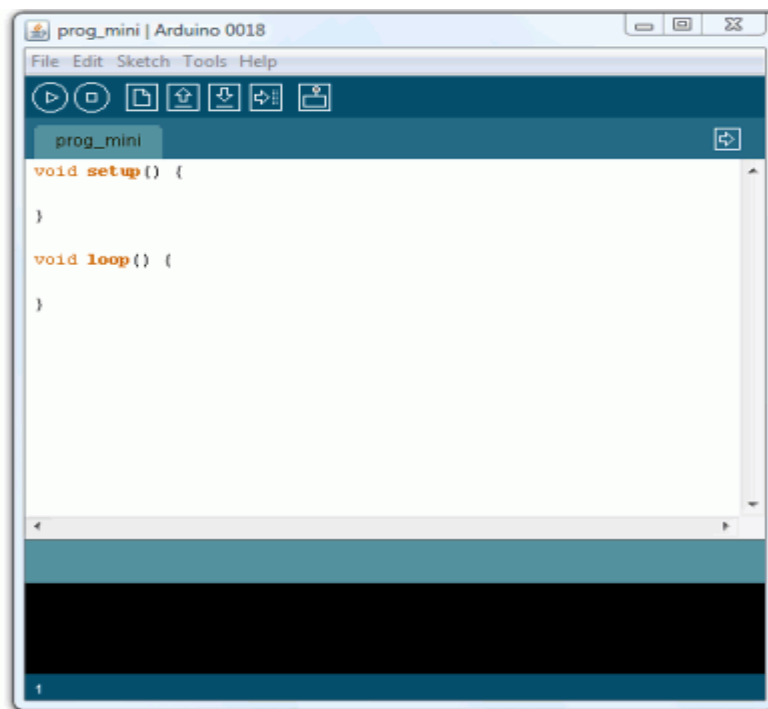
Les commentaires sont des portions du code source entre `/*et*/` ignorées par le compilateur ou l'interpréteur, car ils ne sont pas censés influencer l'exécution du programme.

- **Définition des variables**

Une variable est un emplacement mémoire utilisé pour stocker une donnée. Une variable a un nom, un type et une valeur.

- **Les fonctions obligatoires `setup()` et `loop()`**

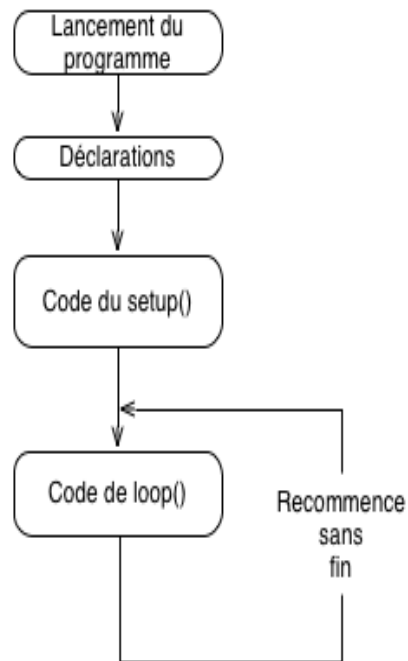
Il y a deux fonctions spéciales, obligatoires, qui sont partie intégrante de tout programme en langage Arduino :



**Figure2.8: La fonction `setup` et `loop`.**

**La fonction `setup()`** : doit être appelée en premier, lorsque le programme commence. Cette fonction n'est exécutée qu'une seule fois au démarrage du programme. C'est le lieu privilégié pour réaliser les tâches d'initialisation des broches numériques ou des librairies.

**La fonction `loop()`** : suit immédiatement la fonction `setup` et comprend le code à exécuter en boucle, lisant les capteurs en entrée et déclenchant les actionneurs en sortie, etc. Cette fonction est le noyau de tout programme Arduino et réalise l'essentiel du travail. La figure2.9 présente le fonctionnement de la structure de base d'un programme :



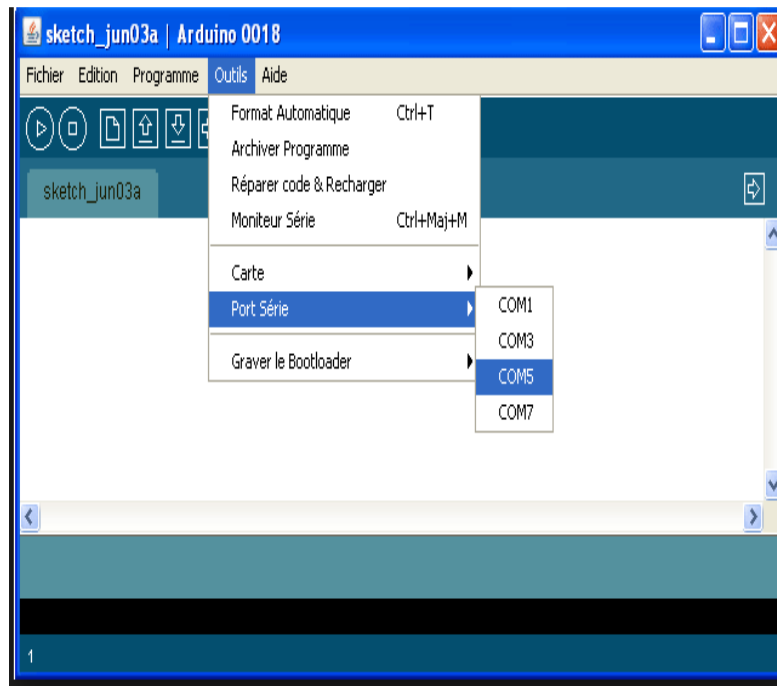
**Figure2.9:Fonctionnement de la structure de base du programme.**

### 2.4.3. Téléverser un programme dans la carte

Avant le transfert du programme dans la carte, il est nécessaire de sélectionner le type de la carte et le numéro de port USB comme le montre les figures suivante :



**Figure2.10: Choix de la carte.**



**Figure2.11: Choix du port.**

La figure suivante résume la compilation et le téléversement du programme dans la carte :

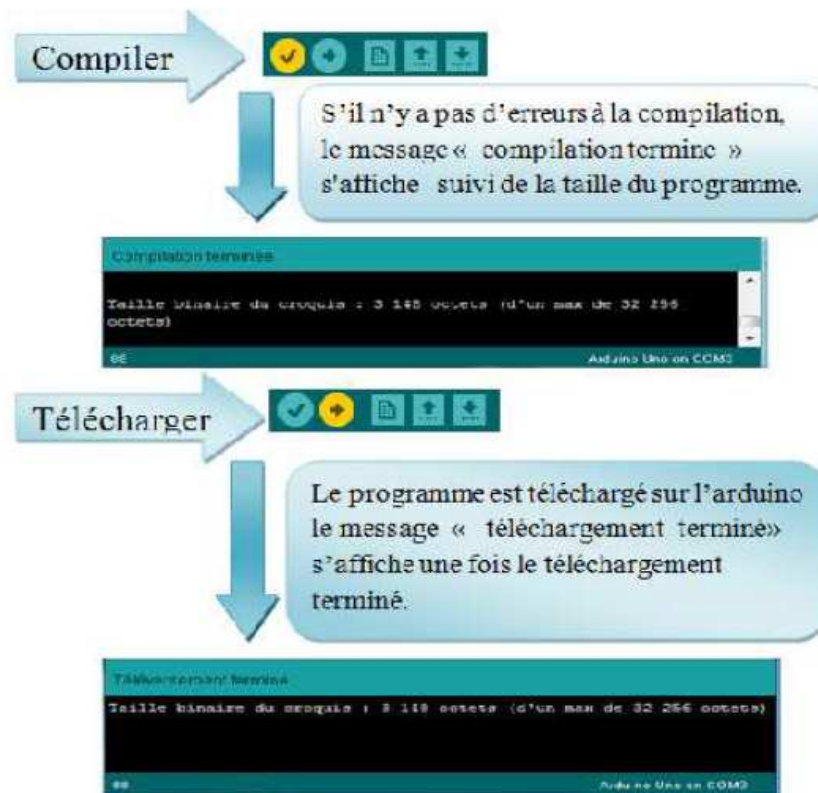


Figure2.12: Compilation et téléversement d'un programme Arduino.

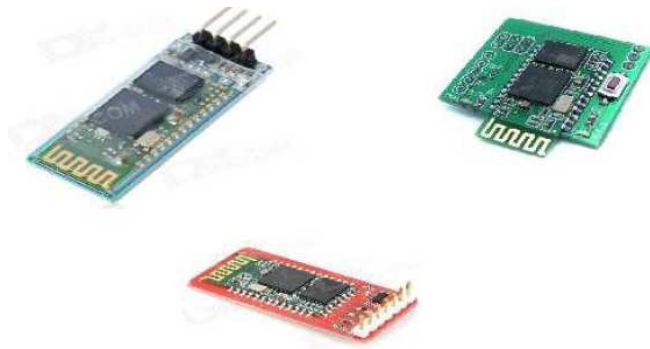
## 2.5. Les shields de la carte Arduino

La carte Arduino généralement est associée aux accessoires qui simplifient les réalisations, dites shields capables de couvrir la plupart des besoins d'une application embarquée <sup>[27]</sup>.

### 2.5.1. Les shields de communications

#### 1. Le shield Bluetooth

Le Module Microcontrôleur Arduino Bluetooth est la plateforme populaire d'Arduino avec une connexion série Bluetooth à la place d'une connexion USB, permettant la communication entre la carte et le PC ou autre équipement disposant de la norme Bluetooth.



**Figure2.13: Les trois type de modules Bluetooth <sup>[27]</sup>.**

## 2. Le shield wifi

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



**Figure2.14: Shield wifi <sup>[27]</sup>.**

## 3. Les shields XBee

Les shields XBee sont des circuits de communication sans-fil utilisant les protocoles 802.15.4 et Zigbee, permettant d'établir une liaison sans fil entre deux shields.



**Figure2.15: Shield XBee <sup>[27]</sup>.**



### 2.5.2. Les capteurs

Les capteurs sont des dispositifs transformant l'état d'une grandeur physique observé en un signal analogique ou numérique. la figure 2.16 présente deux capteurs :

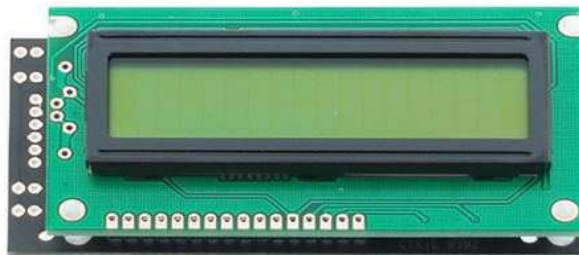


**Figure2.16: Capteur ultrason et capteur de température <sup>[27]</sup>.**

### 2.5.3. Les drivers

#### 1. Afficheur LCD

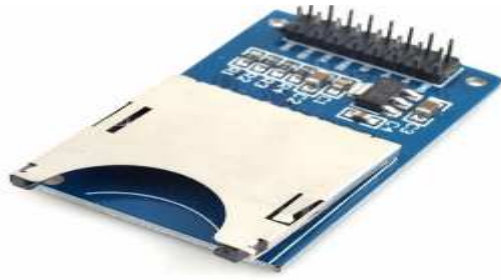
Les afficheurs LCD sont devenus indispensables dans les systèmes techniques qui nécessitent l'affichage des paramètres de fonctionnement. Ces Afficheurs permettent d'afficher des lettres, des chiffres et quelques caractères spéciaux. La figure suivante présente un afficheur LCD :



**Figure2.17: Afficheur LCD <sup>[27]</sup>.**

#### 2. Le shield SD CARD

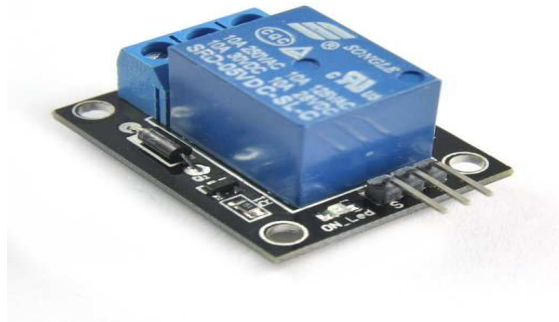
L'Arduino offre la possibilité d'écrire et de lire des données sur une carte SD, en utilisant une librairie intégrée dans l'IDE. la figure suivant montre la carte SD :



**Figure2.18: Carte SD <sup>[27]</sup>.**

### **3. Le relais**

C'est une sorte d'interrupteur télécommandé, piloté avec une carte Arduino. Contrairement au microcontrôleur l'ATMega328, le relais permet de commander des objets plus puissants comme les moteurs. La figure suivante présente un relais :



**Figure2.19: Le relais <sup>[27]</sup>.**

**2.6. Conclusion**

Dans ce chapitre, nous avons présenté la technologie Arduino, ainsi que l'environnement de développement associé, et les différents types d'accessoires utilisés par la carte. Les concepts traités dans ce chapitre nous aideront à comprendre au mieux notre environnement de développement et les notions fondamentales pour mener à bien notre travail.

Le chapitre suivant sera consacré à l'analyse et la conception de notre projet.

# **Chapitre 3**

## **Conception du système**

### **3.1. Introduction**

Après avoir clarifié les différentes technologies sur lesquelles se base notre système, au cours des deux premiers chapitres, nous allons maintenant présenter la conception de notre projet.

Dans ce chapitre, nous parlerons sur l'architecture adoptée pour notre application, nous allons ensuite décrire la conception matériel, qui a pour but de préciser et détailler encore plus les composants de cette architecture à l'aide de quelques schémas électrique, en fin nous présenterons la conception logicielle que nous expliciterons par un organigramme de fonctionnement du système.

### **3.2. Objectifs du système**

Dans ce travail nous allons réaliser un système qui a pour objectif de surveiller l'habitat pour prévenir les accidents domestiques (fuite de gaz, incendie,..).

### **3.3. Architecture générale du système**

L'ordinateur est relié au microcontrôleur par un port USB qui sert à transmettre les informations ainsi qu'à alimenter en électricité la carte Arduino. Les données extérieures des capteurs sont envoyées au microcontrôleur. S'il faut effectuer une action, l'ordinateur envoie une instruction au microcontrôleur qui, via un actionneur, agit sur l'appareil électrique. De plus, les données relevées par les capteurs peuvent être visibles sur un écran.

Pour pouvoir faire tout cela, il faut créer un programme permettant de contrôler les appareils électriques en fonction des données reçues.

Dans notre cas, notre système est constitué d'une carte à microcontrôleur de type Arduino, qui est connecté à un ensemble de périphériques constitués de capteurs et d'actionneurs. qui sont les suivants :

- Capteur de qualité d'air MQ135 ;
- Capteur de température et d'humidité SHT75 ;
- Le module de communication GSM SIM900 ;
- Le buzzer ;
- Les leds ;
- Le relais.

La figure suivante présente l'architecture générale de système a réalisé :

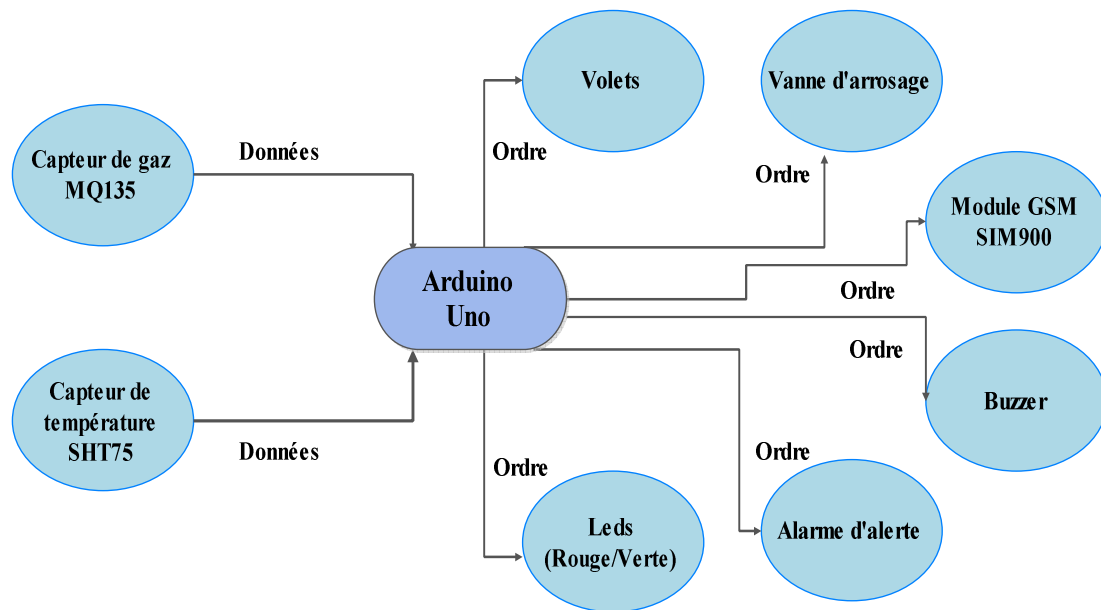


Figure3.1: Architecture générale du système.

### 3.4. Comportement prévu du système

Le système représenté dans la figure3.1 assurera deux fonctions principales (détection de gaz et incendie). Dans le schéma suivant, nous allons présenter les composants intervenants dans chaque fonction, et le concept de communication entre eux :

#### 1. Détection de gaz :

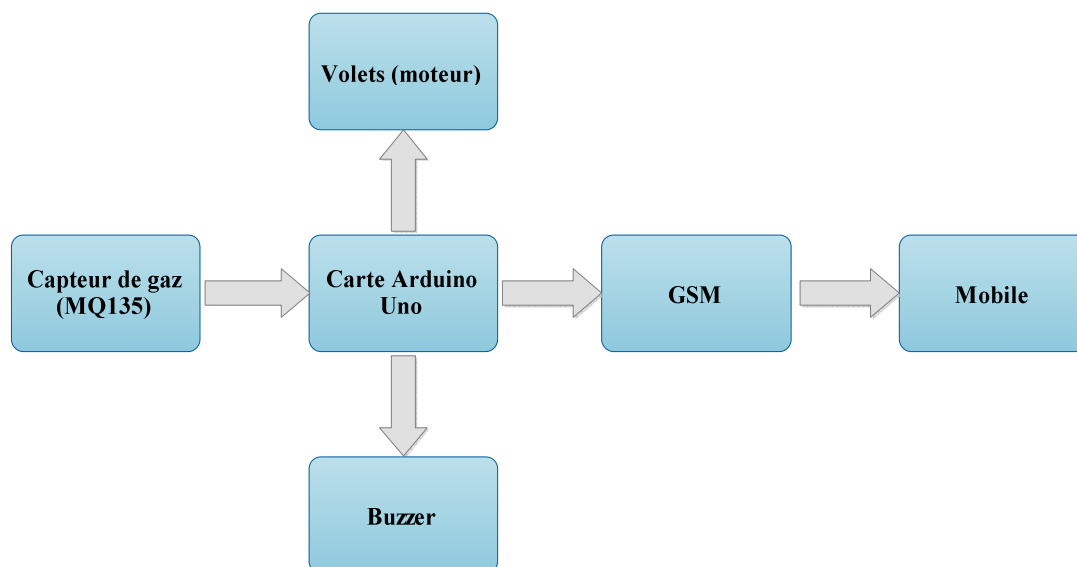


Figure3.2: Concept de communication lors d'une détection de gaz.

## 2. Détection d'incendie :

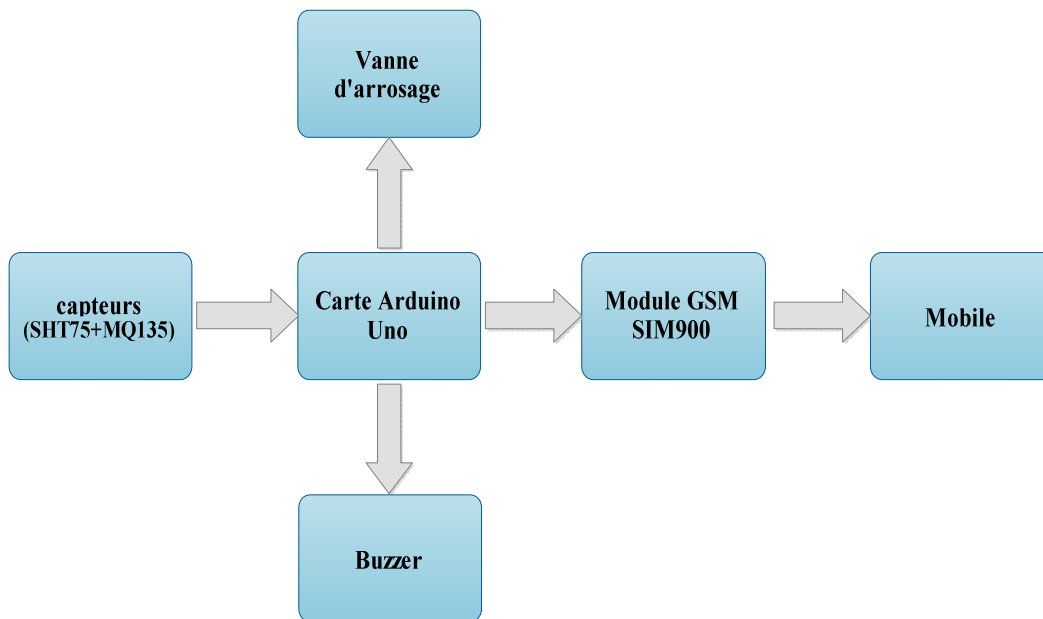


Figure3.3: Concept de communication lors d'une détection d'incendie.

## 3.5. Conception matériel

Afin de concevoir le système que nous avons décrit précédemment, nous aurons besoin d'un ensemble de composants matériels, que nous détaillerons dans la suite de ce chapitre. En fin nous présenterons un schéma global de notre système.

### 3.5.1. Les capteurs

Pour assurer la sécurité de notre maison contre les fuites de gaz et les incendies, on fait appel à un ensemble de capteurs qui permettent de les surveiller en permanence. Parmi ces capteurs il y a ceux qui ont des sorties analogique ou bien numérique.

#### 1. Les capteurs analogiques et numériques

Les capteurs à sortie analogique sont des organes chargés de prélever une grandeur physique à mesurer et de la transformer en une grandeur exploitable (signal électrique (courant, tension, fréquence, valeur moyenne)). Ce signal sera converti par le convertisseur analogique numérique en une grandeur numérique manipulable, sur 10bits (0 et 1023).

Les capteurs à sortie numérique est une séquence d'états logiques qui, en se suivant, forment un nombre. La sortie peut prendre une infinité de valeurs discrètes. Le signal des capteurs numériques peut être du type : train d'impulsions, avec un nombre précis d'impulsions ou avec une fréquence précise ; code numérique binaire.

Pour réaliser notre projet nous avons utilisé un capteur à sortie analogique MQ135, un capteur à sortie numérique SHT75.

## 2. Capteur de qualité de l'air MQ135

### – Description

Le MQ135 est un capteur qui permet de mesurer la qualité de l'air. Il est sensible aux principaux polluants présents dans l'atmosphère de la maison. Comme le CO<sub>2</sub>, l'alcool, le Benzène, l'oxyde d'azote (NOX) et l'ammoniac (NH<sub>3</sub>), et le monoxyde de carbone.

Ce capteur utilise le dioxyde d'étain (SnO<sub>2</sub>) dont la conductivité électrique varie en fonction de la présence de polluant(s). Comme tous les capteurs de la série MQ il faudra procéder à une calibration dans une atmosphère de référence pour mesurer la présence d'un polluant en particulier. <sup>[20]</sup>

La figure suivante montre un aperçu du capteur MQ135 :



Figure3.4: Capteur MQ135.

Broches	Désignation
VCC	Alimentation du capteur (5v).
DOUT	Sortie digital (aucune).
AOUT	Sortie analogique (A0).
GND	Masse du capteur (0v).

### – Caractéristique

Alimentation	2.5 ----- 5.5 V.
Puissance	0.8 W.
Courant	0.16 A.
Tension de sortie	5V.
Nombre de broche	4 broches.



Dimension	18.0*17.0*6.0 mm.
Type d'interface	Analogique.
Stabilité	Système stable à longue durée de vie.
Circuit de contrôle	Simple.
Temps de réponse	Rapide et haute sensibilité.

Le montage le plus utilisé pour ce composant est donnée par la figure ci-dessous :

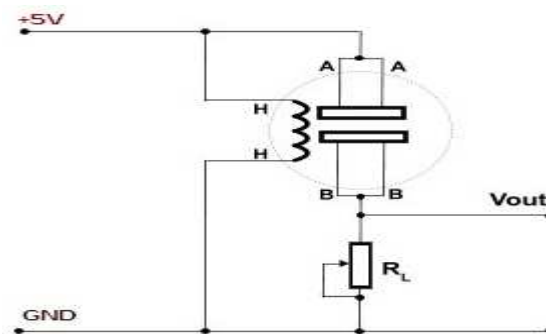


Figure3.5: Montage du capteur MQ135 <sup>[20]</sup>.

La figure suivante représente le branchement du capteur MQ135 avec notre carte Arduino :

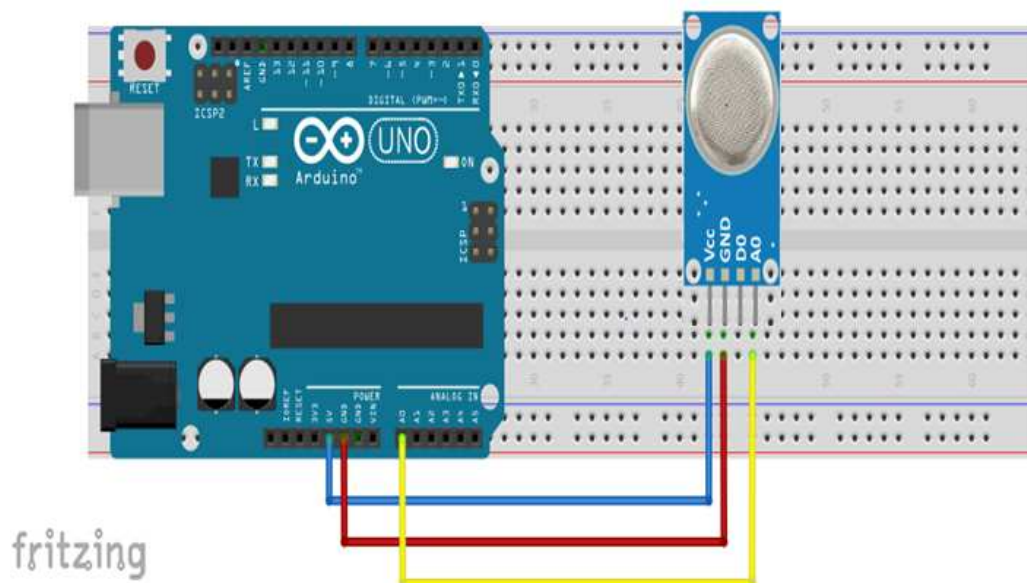


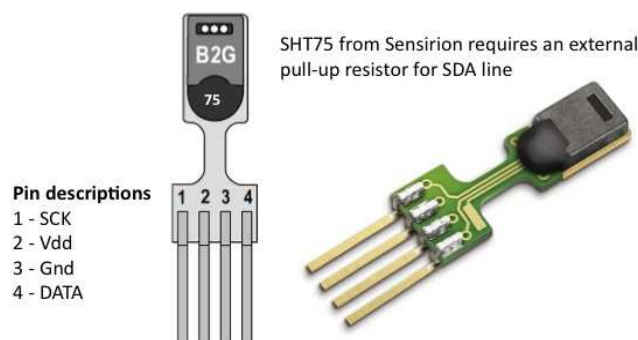
Figure3.6: Branchement de MQ135 avec la carte Arduino.

### 3. Le capteur de température et d'humidité SHT75

#### – Description

Le SHT75 est un capteur numérique de température et d'humidité relative en boîtier SIP 4 broches. Ce capteur intègre des éléments de capteur et de traitement de signaux dans un format compact. Offre une sortie numérique entièrement calibrée. Un élément de capteur capacitif unique est utilisé pour mesurer l'humidité relative tandis que la température est mesurée par un capteur à bande interdite. La technologie CMO Sens appliquée garantit une excellente fiabilité et une stabilité à long terme. Le SHT75 est couplé à un convertisseur A/N 14 bits et à un circuit d'interface série. Cela permet une qualité supérieure des signaux, un temps de réponse rapide et une insensibilité aux perturbations externes (EMC). L'interface série 2 fils et la régulation de tension interne permet une intégration au système particulièrement aisée. Du fait de sa taille compacte et de sa faible consommation, le SHT75 constitue le choix idéal pour les applications les plus exigeantes <sup>[21]</sup>.

La figure suivante monte un aperçu du capteur SHT75 :



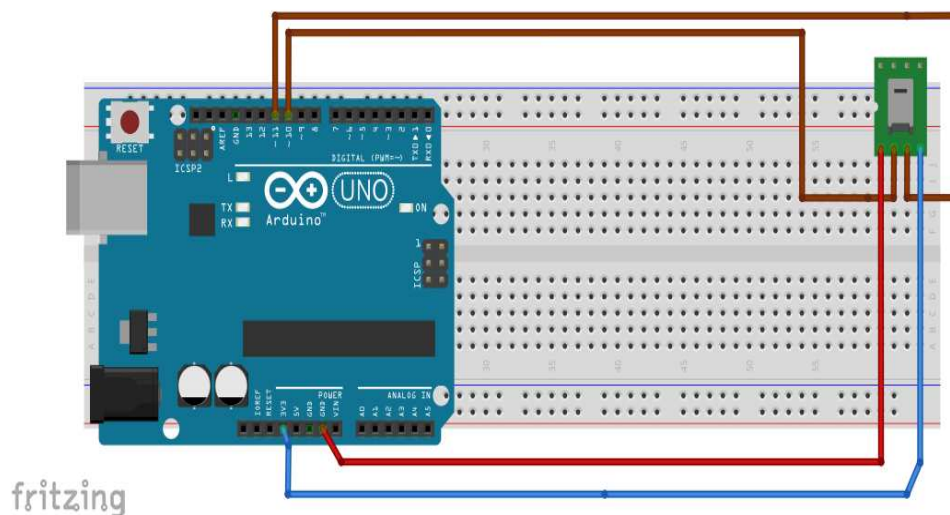
**Figure3.7: Capteur SHT75 <sup>[21]</sup>.**

Broches	Désignation
SCK	Horloge série, entrée seulement(D11).
Vdd	Alimentation du capteur (3.3v).
GND	Masse du capteur.
DATA	Sortie numérique (D10).

## – Caractéristique

Fonction de capteur	Température et humidité
Type d'interface	Série-2 fils
Précision	1.8%
Type de montage	Traversant
Nombre de broche	4
Plage de température de fonctionnement	-40 ----- +123.8°C
Température de fonctionnement minimum	-40 °C
Température d'utilisation maximum	+123.8 °C
Tension d'alimentation de fonctionnement min	2.4 V
Tension d'alimentation fonctionnement max	5.5 V
Type de boîtier	SIP
Longueur	5.08 mm
Hauteur	13.5 mm
Largeur	3.1 mm
Dimension	5.08*3.1*15.5 mm
Résolution	14 bit

La figure suivante représente le branchement du capteur SHT75 avec notre carte Arduino :



**Figure3.8: Branchement du capteur SHT75 avec la carte Arduino.**

### 3.5.2. Le contrôleur

Cet élément est indispensable pour la réalisation de notre système. Sa fonction est le contrôle de tous les autres composants. Dans notre cas, on utilise une carte Arduino UNO décrite précédemment.

La carte Arduino permet de vérifier et traiter les données recueillies par les capteurs. Ensuite elle envoie les commandes à exécuter vers les actionneurs, ainsi qu'elle permet de communiquer les informations via le

### 3.5.3. Les actionneurs

La fonction des actionneurs est l'exécution d'une tâche sur ordre du microcontrôleur. Si par exemple, la concentration de monoxyde de carbone est élevée, l'action pourrait être le déclenchement d'une alarme d'alerte et l'ouverture des fenêtres. Et si un incendie est détecté, l'action pourrait être le déclenchement d'une vanne d'arrosage, et une alarme d'alerte.

Dans notre cas, nous avons choisi d'utiliser les actionneurs suivants :

- Le buzzer,
- Les leds,
- Le module de communication GSM SIM900.
- Le relais.

#### 1. Buzzer

Un buzzer est un élément électromécanique ou électronique qui produit un son quand on lui applique une tension. Certains nécessitent une tension continue (buzzers électromécaniques), d'autres nécessitent une tension alternative (transducteurs piézo-électrique) <sup>[22]</sup>.

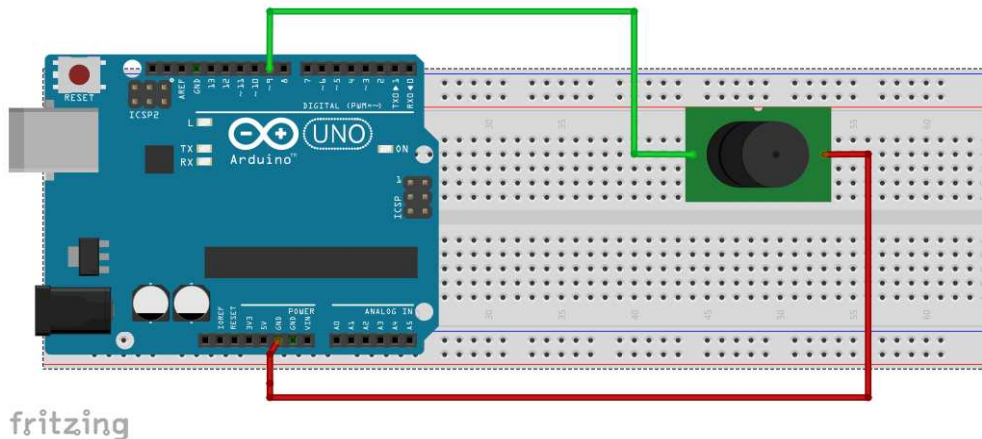
La figure suivante montre un aperçu du buzzer :



**Figure3.9: Buzzer.**

Broches	Désignation
OUT	Sortie proportionnelle du buzzer (D9).
GND	Masse du capteur (0v).

La figure suivante représente le branchement d'un buzzer avec notre carte Arduino :



**Figure3.10: Branchement du buzzer avec la carte Arduino.**

## 2. Les leds

Après une analyse, nous avons vu qu'il fallait mettre en place un autre actionneur pour le système, qui permet d'informer les personnes présentes dans la maison sur l'état du système.

Pour cela nous avons choisi d'utiliser des lampes à LED (Light-Emitting Diode), qui permettent de s'allumer quand un événement est détecté, par exemple :

- Si le système est activé, et aucun problème dans la maison, alors la LED verte s'allume.
- Si fumée est capter, alors la LED rouge s'allume.
- Si fuite de gaz est capter, alors la LED rouge s'allume.

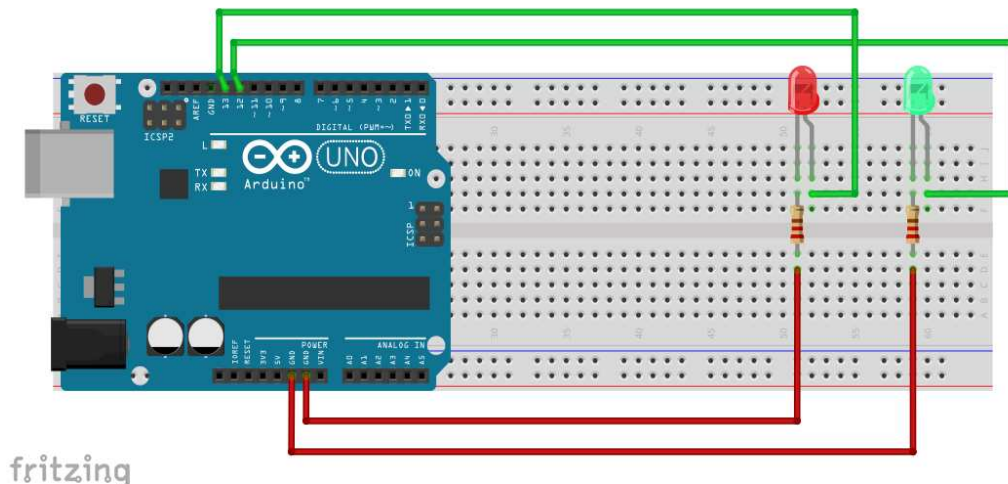
La figure suivante monte un aperçu du leds :



**Figure3.11: Leds.**

Broches	Désignation
OUT	Sortie proportionnelle du la led (D12, D13).
GND	Masse du capteur (0v).

La figure suivante représente le branchement d'une LED avec notre carte Arduino :



**Figure3.12: Branchement des leds avec la carte Arduino.**

### 3. Le module de communication GSM SIM900

Le SIM900 est un module sans fil fiable et ultra-compact. C'est un module quadri-bande GSM/GPRS complet de type SMT et conçu avec un processeur à puce unique très puissant intégrant une cour ARM926EJ-S, ce qui vous permet de bénéficier de solutions de petites dimensions et économiques. Doté d'une interface standard de l'industrie, le SIM900 offre des performances GSM/GPRS à 850/ 900/ 1 800/ 1 900 MHz pour la voix, les SMS, le fax et les données dans un petit facteur de forme et avec une faible consommation d'énergie <sup>[23]</sup>.

La figure suivante monte un aperçu du module de communication GSM SIM900 :



Figure3.13: Module de communication GSM SIM900 <sup>[23]</sup>.

Broches	Désignation
VCC	Alimentation du module GSM(9v).
DOUT	Sortie digital (D7 et D8).
GND	Masse du capteur (0v).

La vue de dessous :

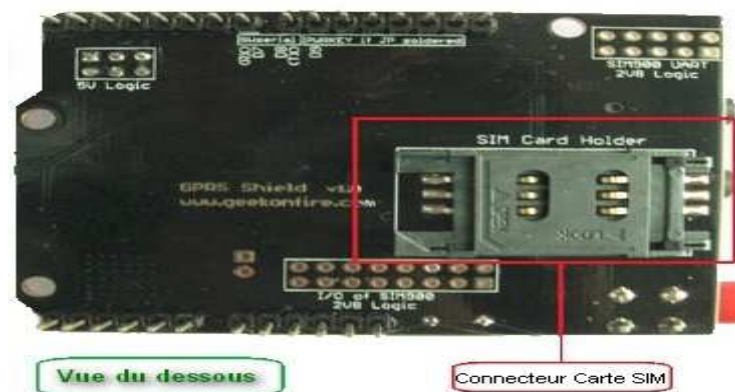


Figure3.14: Vue de dessous du GSM SIM900 <sup>[23]</sup>.



La vue de dessus :

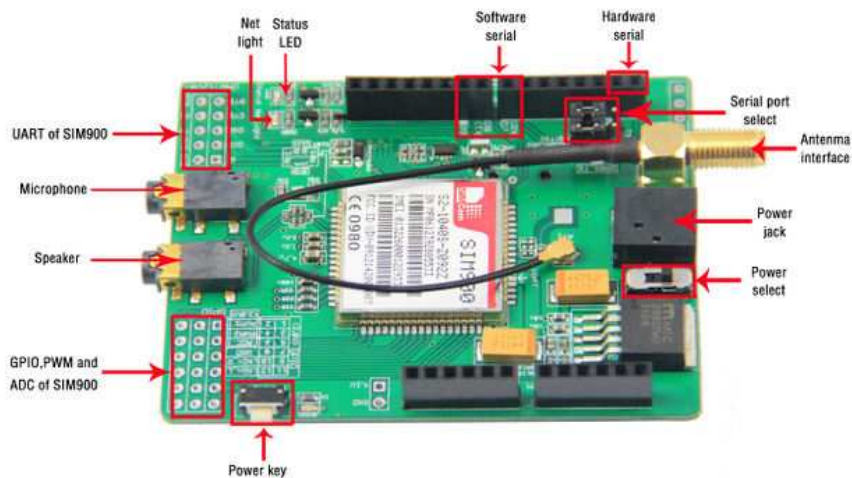


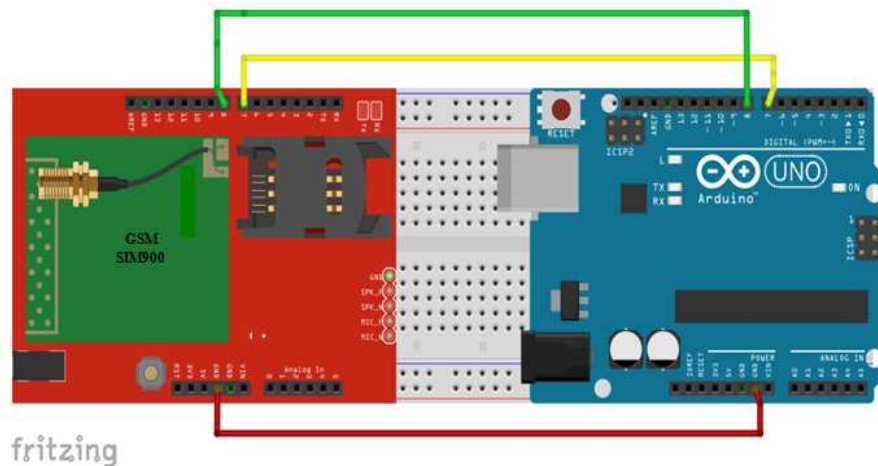
Figure3.15: Vue de dessus du GSM SIM900 [23].

Caractéristique

Tension d'alimentation	4.5 à 5.5 VCC
Tension en entrée (VH)	0.7 à 5.5 VCC
Tension en entrée (VL)	-0.3 à 0.3 VCC
Consommation électrique (impulsion)	2000 mA
Consommation électrique (en continu)	500 mA
Vitesse de transmission	115200 bps
Dimensions de la carte	71.4*66*1.6 mm
Poids	40 g

La figure suivante représente le branchement du module de communication GSM SIM900 avec notre carte Arduino :





**Figure3.16: Branchement du GSM SIM900 avec la carte Arduino.**

### Les Commandes AT

Le microcontrôleur commande le module GSM en utilisant des commandes AT. Les commandes AT (ou commandes de Hayes) constituent un langage de commande développé afin de commander les modems, ce langage est développé à l'origine pour le modem Hayes Smart modem 300. Ce jeu de commandes s'est ensuite retrouvé dans tous les modems produits. Toutes les commandes doivent être précédées de AT. Par exemple, ATA signifie "Répondre". Certaines commandes, comme ATA, sont communes à tous les modems, d'autres sont variables selon le modèle spécifique, comme l'affichage de la configuration peuvent varier selon l'équipement. Il faut donc se reporter à la documentation de son modem. Suivant le cahier des charges, on aura besoin des commandes AT suivantes :

- AT+CMGF : Sélection du format des messages.
- AT+CMGS : Envoi de messages.

### 3.5.4. Le relais

#### – Description

Un relais électromécanique est un organe électrique permettant de dissocier la partie puissance de la partie commande, il permet l'ouverture / fermeture d'un circuit électrique par un second circuit complètement isolé (isolation galvanique) et pouvant avoir des propriétés différentes [24].

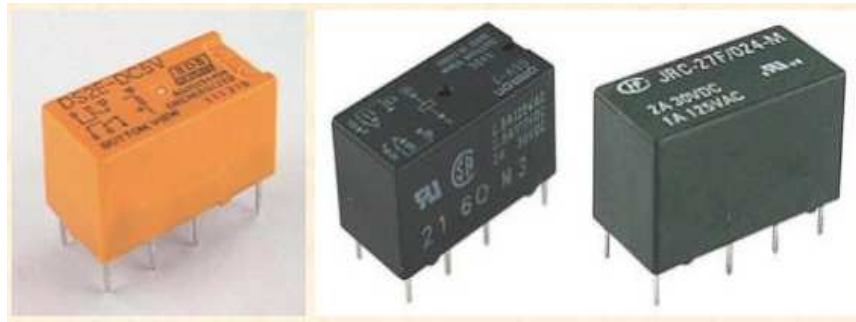


Figure 3.17: Exemple de relais <sup>[24]</sup>.

Un relais est composé principalement d'un électroaimant, qui lorsqu'il est alimenté, transmet une force à un système de commutation électrique : les contacts.

L'électroaimant peut être, suivant les spécifications et besoins, alimenté en TBT (Très Basse Tension) (moins de 12 V, 24 V, 48 V) continu ou alternatif ou en BT (Basse Tension) (230 V, 400 V).

Le système de commutation peut être composé d'un ou plusieurs interrupteurs simples effets appelés contacts normalement ouverts (NO) ou normalement fermés (NF), d'un ou plusieurs inverseurs (contacts repos-travail RT). Ces commutateurs sont adaptés aux courants et à la gamme de tensions à transmettre à la partie puissance.

Dans les systèmes mettant en œuvre une certaine puissance, on appelle les relais des contacteurs. Divers systèmes mécaniques ou pneumatiques peuvent créer un retard à l'enclenchement ou au relâchement.

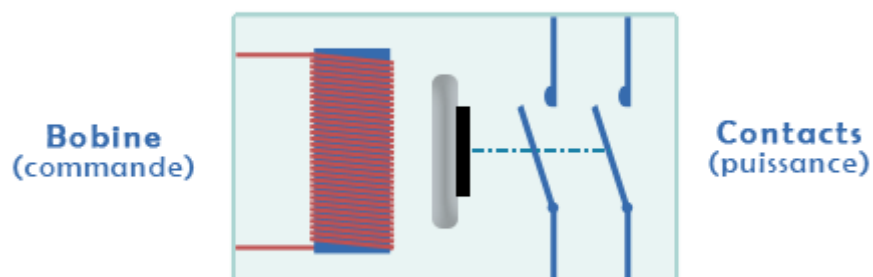


Figure 3.18: Symbole du relais <sup>[24]</sup>.

### – Fonctionnement des relais

Un relais peut être monostable ou bistable :

**Fonctionnement monostable** : les contacts commutent quand la bobine est alimentée et le retour à l'état initial se fait quand la bobine n'est plus alimentée.

**Fonctionnement bistable à une bobine** : on alimente la bobine pour que les contacts commutent : l'état ne change pas quand la bobine n'est plus alimentée, un système mécanique bloque le retour. Pour revenir à l'état initial, on alimente à nouveau la bobine pour débloquer le mécanisme, dans certain cas en inversant la polarité de l'alimentation.

**Fonctionnement bistable à deux bobines** : on alimente la première bobine pour que les contacts commutent : l'état ne change pas quand la bobine n'est plus excitée. Pour revenir à l'état initial, on alimente la deuxième bobine.

## 3.6. Conception logicielle

### 3.6.1. Fonctionnalités de base

Le langage Arduino vient avec un nombre important de fonction de base permettant d'interagir avec son environnement. Les fonctions les plus utilisées sont les fonctions d'entrée/sorties et les fonctions de gestion du temps :

#### 1. Fonctions destinées aux broches

Ce sont elles qui permettent d'envoyer ou de mesurer une tension sur une des broches de la carte. Dans un premier temps, avant d'effectuer une mesure ou d'envoyer une commande. Il est nécessaire de définir la direction des broches utilisées. Pour cela on fait appel à la fonction `pinMode` en lui donnant d'une part, la broche concernée, et d'autre part, la direction :

#### – `pinMode(pin, OUTPUT/INPUT)`

Toutes les fonctionnalités sur les broches d'entrées sorties sont utilisables par le biais de quatre fonctions :

**`digitalRead(pin)`** : mesure une donnée numérique sur une des broches, la broche en question doit être réglée en entrée.

**`digitalWrite(pin, value)`** : écrit une donnée numérique sur une des broches, la broche concernée doit être réglée en sortie. Le paramètre `value` doit être égal à HIGH (état 1 soit 5V) ou LOW (état 0 soit 0V).

**`analogRead(pin)`** : mesure une donnée analogique sur une des broches (compatible seulement), la broche doit être réglée sur entrée.

**analogWrite(pin, value)** : écrit une donnée sous forme de PWM sur une des broches (compatible uniquement), la broche doit être réglée en sortie. Le paramètre value doit être compris dans l'intervalle [0;255].

## 2. Fonction de gestion du temps

Pour la plupart des applications de domotique, il est nécessaire de faire intervenir des intervalles de temps. Par exemple, pour gérer le temps d'appui sur un bouton ou pour faire une sonnerie qui se répète un certain nombre de fois. Le langage Arduino fournit quelques fonctions permettant de gérer le temps.

**delay(ms)** et **delayMicroseconds(us)** : insère une pause suivant le paramètre passé (en milliseconde pour l'un, en microseconde pour l'autre). Cependant ces fonctions bloquent le microcontrôleur, on ne peut alors plus effectuer aucune action.

**Millis()** et **micros()** : ces fonctions incrémentent une variable (interne) pour retourner le nombre de milliseconde (respectivement microseconde) depuis le lancement de la carte.

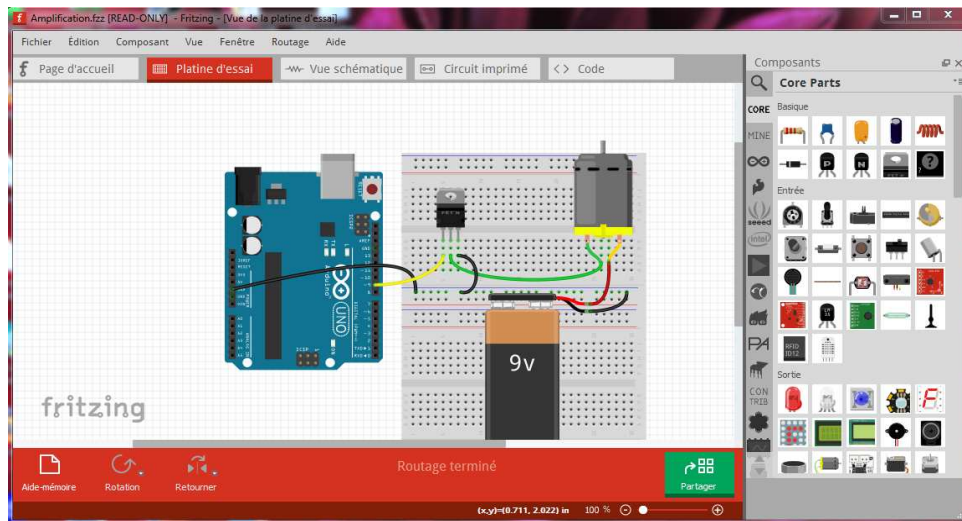
Voici un exemple explicatif :

```
const int pinLed = 13;
void setup()
{
  pinMode(pinLed, OUTPUT); // Broche 13 en sortie
}
void loop()
{
  delay(500); // Attente d'une demi seconde
  digitalWrite(pinLed, HIGH); // Allumage de la LED
  delay(500);
  digitalWrite(pinLed, LOW); // Eteignage de la LED
}
```

**Figure3.19: Exemple explicatif.**

### 3.6.2. Fritzing

Fritzing est logiciel open-source multiplateforme permettant de construire des schémas des circuits que nous utilisons avec Arduino. Plusieurs vues sont disponibles : platine d'essai, schémas électriques et circuits imprimé. Il permet l'export en image pour figurer sur internet <sup>[25]</sup>

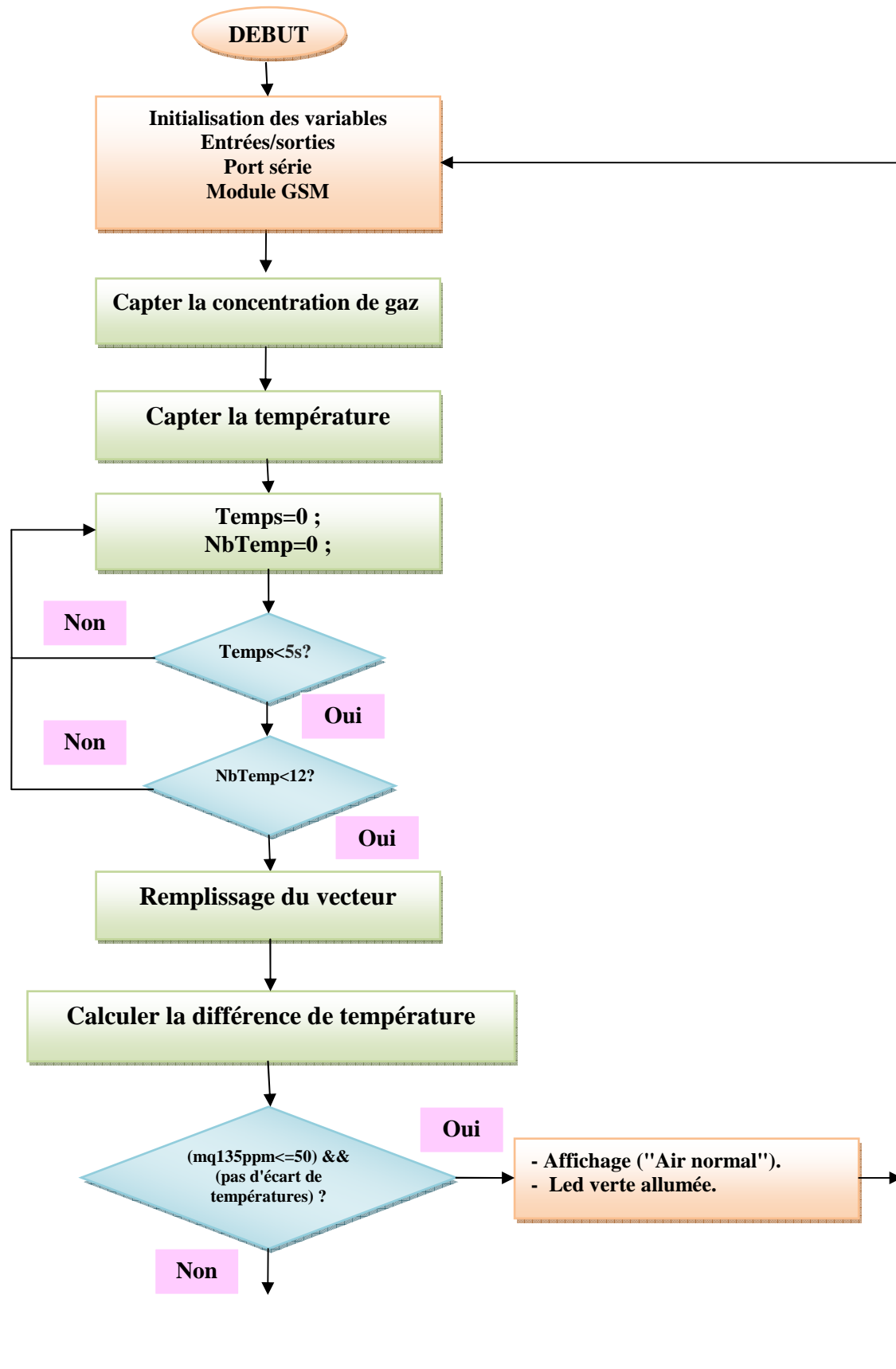


**Figure3.20: L'interface Fritzing [25].**

### 3.6.3. Organigramme de fonctionnement

Le microcontrôleur traite les valeurs qu'il reçoit par les capteurs utilisés dans notre système (MQ135, SHT75). Si les seuils limites définis dans le programme sont dépassés le microcontrôleur effectuera un ensemble d'actions. Par exemple dans le cas d'une fuite de gaz ou bien la présence élevé de certains gaz toxiques, le microcontrôleur allumera la led rouge, déclenchera l'alarme, donnera l'ordre aux fenêtres de s'ouvrir et envoie un message d'alerte dont le contenu est « ATTENTION, DANGER GAZ » afin de prévenir le propriétaire de la maison.

L'organigramme suivant résume le fonctionnement de notre système.



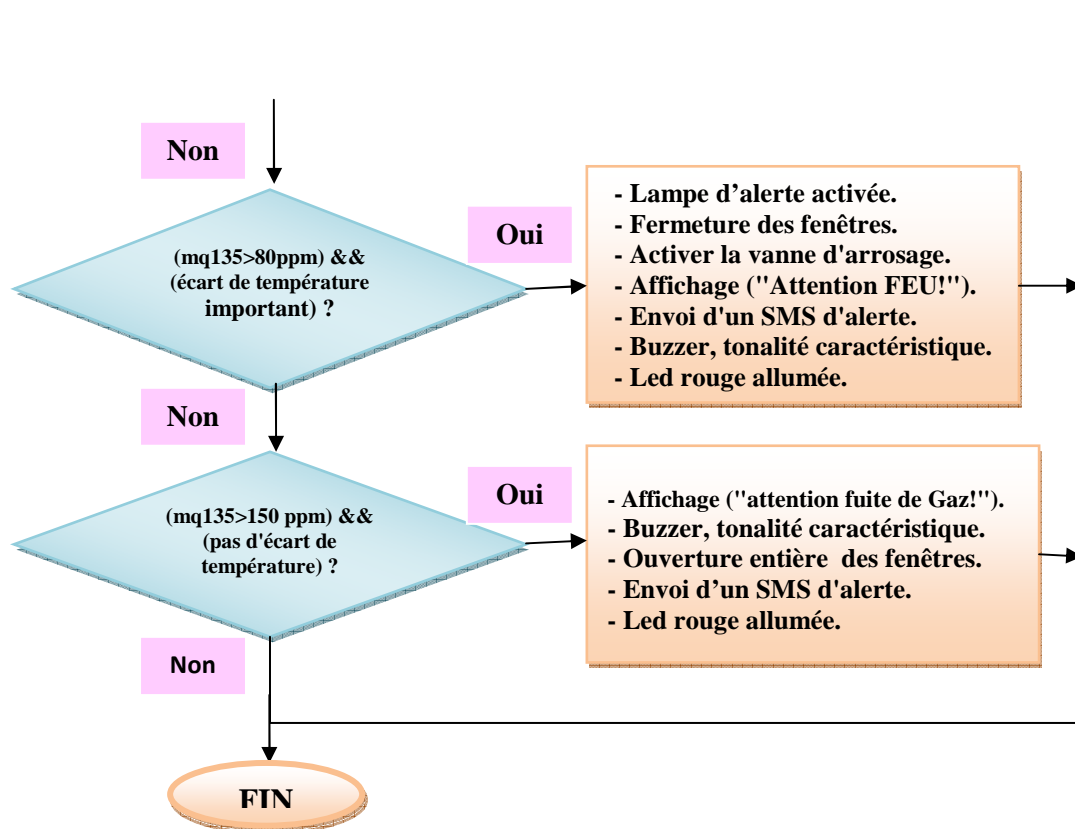


Figure3.21: Organigramme de fonctionnement.

**3.7. Conclusion**

Dans ce chapitre nous avons exposé la structure générale du système en premier lieu, puis nous avons entamé la description de la conception matériels dans laquelle nous avons cité les différents composants qui ont servi à la réalisation de notre système ainsi que leur branchement électrique. En fin nous avons terminé par la présentation de la conception logiciel, là où nous avons décrit les fonctions de base du langage Arduino et l'organigramme de fonctionnement.

Dans le dernier chapitre, nous allons passer à la réalisation de notre système.



# **Chapitre 4**

## **Réalisation du système**

### 4.1. Introduction

L'étape de réalisation est la phase de développement pur, et l'étape qui met en œuvre la solution conceptuelle décrite dans le chapitre précédent.

Dans ce chapitre nous allons présenter les outils et langages qui ont servi à la réalisation de notre projet. Pour finir, nous effectuons une série de test afin d'observer le comportement de notre système.

### 4.2. Outils de développements

Pour la réalisation de notre projet nous avons utilisé des outils logiciels et matériels ainsi qu'un ensemble de langages.

#### 4.2.1. Matériels

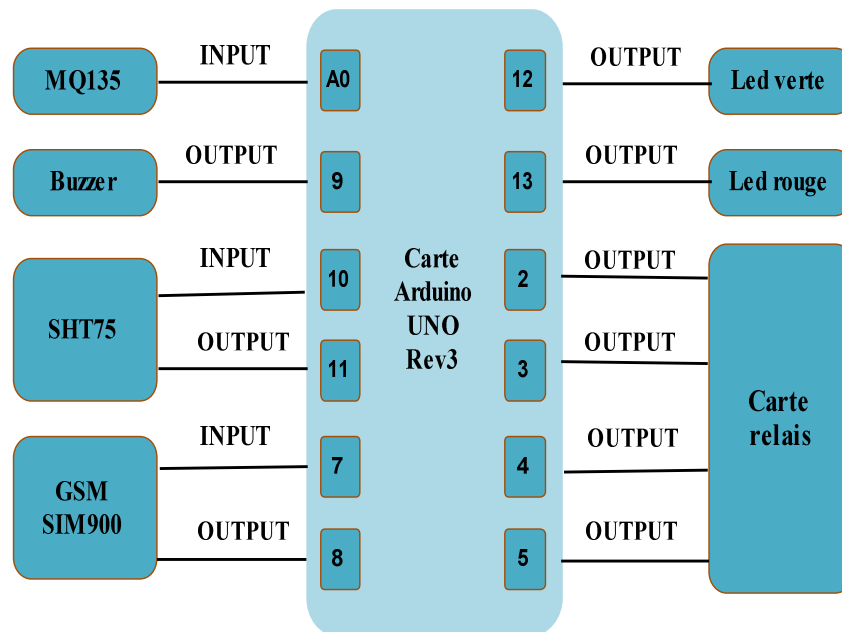
Pour mener notre projet nous avons utilisé un ensemble de composants matériels, dont chacun a son rôle. Nous allons citer ci-dessous les différents composants utilisés dans notre réalisation en précisant le rôle et le prix de chaque composant, comme le montre le tableau ci-dessous :

Composants	Nombres	Prix unitaires (DA)	Rôle
<b>Carte Arduino UNO</b>	1	3000	C'est le composant le plus important dans notre système, il permet de gérer tous les autres composants.
<b>Module GSM sim900</b>	1	15000	Il permet d'établir la communication entre le système et le propriétaire de la maison.
<b>Capteur de gaz</b>	1	1100	Il permet de capter le gaz, et la fumée dans l'aire.
<b>Capteur de température et humidité</b>	1	4800	Il permet de mesurer la température et le taux d'humidité.
<b>Buzzer</b>	1	100	Il permet de produire un sens quand on lui applique une tension.

<b>Leds</b>	2	15	Elle permet d'indiquer un état du système.
<b>Fils</b>	20	15	Ils permettent de connecter les différents composants.
<b>Bread board</b>	1	600	Il permet d'établir rapidement un circuit électrique.
<b>Câble USB</b>	1	180	Il permet de faire communiquer la carte Arduino avec l'ordinateur, et au même temps l'alimenter.
<b>Résistances</b>	2	5	Elles permettent de réduire l'intensité de courant et/ou la tension.
<b>Alimentation SIM900 (Chargeur)</b>	1	800	Il permet d'alimenter le module GSM SIM900 avec 9V.
<b>Le relais</b>	1	500	C'est l'intermédiaire entre la basse tension et la haute tension.

**Tableau 4.1: Composants utilisés.**

La réalisation matériel est faite en premier lieu, chaque partie est tester et réaliser séparément. Après les avoir expérimentés et adopter séparément, nous les avons regroupé et réaliser sur la carte Arduino UNO, comme le montre la figure suivante :



**Figure 4.1: Schéma globale du système.**

La carte est à base de microcontrôleur ATmega328P cadencé par un quartz externe de fréquence 16 Mhz.

- Les broches D10 et D11 sont réservées pour le capteur de température et d'humidité SHT75.
- La broche A0 est réservée pour le capteur MQ135.
- La broche D9 est réservée pour le buzzer.
- Les broches D12 et D13 sont réservées pour les leds verte et rouge.
- Les broches D7 et D8 sont réservées pour le module de communication GSM SIM900.
- Les broches D2, D3, D4, et D5 sont réservées pour la care relais.

#### 4.2.2. Logiciels

Pour l'implémentation de notre application nous avons utilisé Processing qui sert à réaliser la simulation de notre application et IDE Arduino qui nous permet de programmer la carte Arduino que nous avons présenté dans le deuxième chapitre. Dans ce qui suit nous allons présenter les différentes fonctions des deux programmes.

##### 1. Partie Arduino

Notre programme Arduino est constitué d'un ensemble de fonctions qui sont appelées par le programme principal.

**La fonction température () :**

Tandis que le système est alimenté et la fonction est appelée, elle assure le calcul de la température du milieu où se trouve notre système. La figure 4.2 présente la fonction température.

Cette méthode offre des instructions qui permettent de :

- Lancer la capture et récupérer les valeurs retournées par le capteur SHT75.
- Retourner la valeur de la température calculée afin de l'utiliser dans d'autres fonctions pour vérifier si elle franchit le seuil maximal.

```
int Temperature() {  
    // initialisation des variables.  
    // mesurer la température avec le capteur sht75  
    // puis affecter la valeur à une variable temp.  
  
    return temp;  
}
```

**Figure 4.2: Fonction température.**

**La fonction saveTMP() :**

Cette fonction permet d'enregistrer toutes les 5 secondes les valeurs retournées par la fonction température dans un vecteur de 12 entiers.

La fonction est constituée d'instructions qui permettent de :

- Affecter la valeur retournée par la fonction millis() qui permet de calculer le temps en milliseconde depuis la mise en marche du système à la variable saveMillis. Cette instruction sera répétée toutes les 5 secondes.
- Remplir un vecteur en utilisant la valeur retournée par la fonction température si le compteur ne dépasse pas le nombre maximal des éléments du tableau (12 éléments). Si la variable CMP dépasse 12, la fonction exécutera la boucle lui permettant de décaler les éléments du vecteur afin de ne pas écraser les valeurs existantes.

La figure suivante montre la fonction saveTMP() :

```

void SaveTMP () {
    if (millis() - SaveMillis >= 5000) {
        SaveMillis = millis();
        if (CMP <= 12) {
            CMP++;
            // remplissage du vecteur
        }
        if (CMP >= 12) {
            for (int i = 1; i < 12; i++) {
                // décalage du contenu du vecteur d'une case
            }
            // remplissage de la dernière case du vecteur
        }
    }
}

```

**Figure4.3: Fonction saveTMP.**

#### La fonction calcDIF\_TMP() :

Cette fonction indique si il n'y a pas eu une augmentation brusque de température causé par un incendie dans le milieu où se trouve le système.

La fonction calcDIF\_TMP() permet de :

- Initialiser la variable booléen « Etat » à false a chaque appel de la fonction.
- Parcourir le vecteur rempli par la fonction save\_TMP et mettre la variable état à true si la température a augmentée.
- Retourner la variable état.

La figure suivante montre la fonction calcDIF\_TMP() :

```

bool CalcDIF_TMP() {
    // initialisation de la variable ETAT à false.
    for (int i = 0; i < 11; i++) {
        // Tester s'il y'a une augmentation brusque
        // de la température , mettre la variable ETAT a true
    }
    return Etat;
}

```

**Figure4.4: Fonction CalcDIF\_TMP().**

#### La fonction Alarme () :

La fonction présentée dans la figure4.5 est appelé dans le programme pour prévenir les personnes présentes dans la maison en cas de danger.

Elle permet :

- Déclencher et d'arrêter une alarme.

```
void Alarme(int FRQ, int TMP) {  
    // Déclencher le buzzer avec une  
    // fréquence FRQ passé en argument.  
    // attendre un délai TMP.  
    // arrêter le buzzer.  
    // attendre un délai de TMP  
}
```

**Figure4.5:Fonction alarme ().**

#### **La fonction envoyer\_sms () :**

La fonction envoyer\_sms() présentée dans la figure 4.6 permet d'envoyer un message d'alerte au propriétaire. Cette fonction utilise la librairie <SoftwareSerial.h> afin de créer (émuler) une autre voie série autre que 0 et 1.

Elle permet de :

- Configurer le mode de fonctionnement en mode sms texte.
- Envoie du message au numéro du propriétaire en cas de feu ou de fuite de gaz.

```
void envoyer_sms(const char * msg) {  
    // configuration du module.  
    // L'envoi du numéro de téléphone avec l'indicatif(+213).  
    // l'envoi du corps du message .  
    // attente de 5 seconde le temps de l'envoi du message  
}
```

**Figure4.6: Fonction envoyer\_sms().**

## **2. Partie Processing**

Le Processing est un logiciel de création multimédia, il possède la particularité d'utiliser des instructions informatiques pour dessiner, réaliser des animations en 2 ou 3 dimensions, créer des œuvres sonores et visuelles, concevoir des objets communicants qui interagissent avec leur environnement. Ce mode d'expression artistique par le code utilise les caractéristiques propres à l'informatique (rapidité d'exécution, automatisation des actions et des répétitions, interaction, etc.) pour produire des créations originales qui n'auraient pas vu le jour autrement ou dont la réalisation,

à l'aide de procédés plus classiques ou de logiciels plus complexes, aurait demandé davantage de temps.

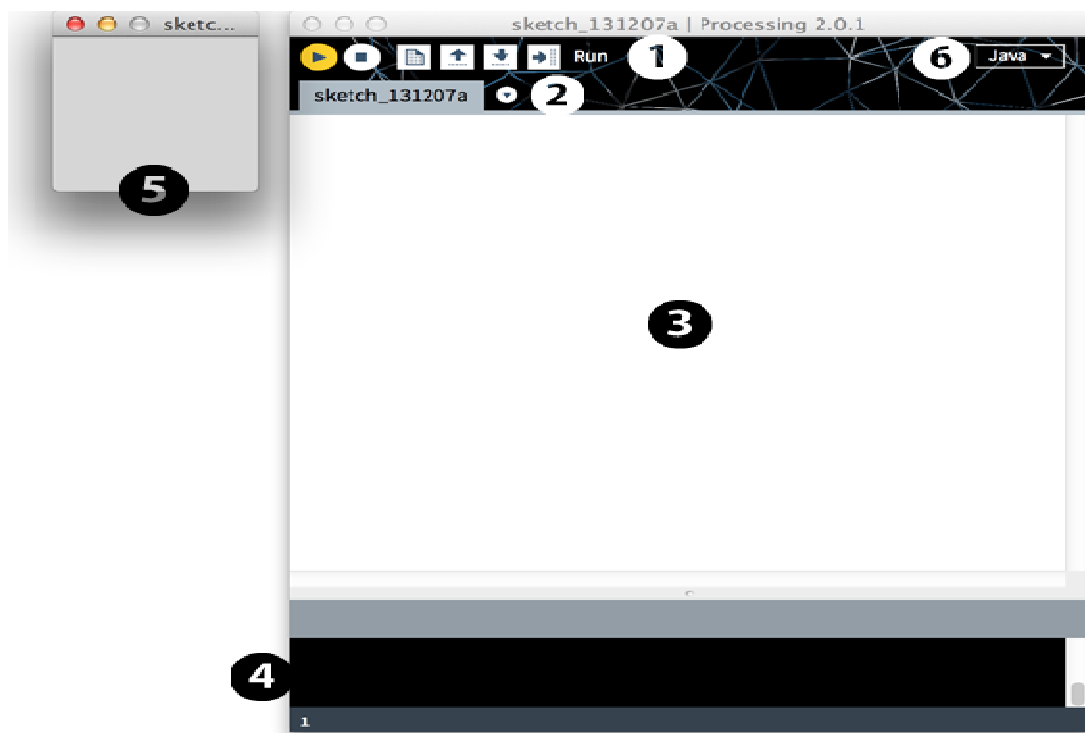
Processing permet également de programmer des circuits électroniques qui interagissent avec le milieu qui les entoure. Connectés à des capteurs sonores, thermiques, de mouvement, ces circuits électroniques peu coûteux, dénommés microcontrôleurs, peuvent

en retour générer des images, actionner un bras articulé, envoyer des messages sur internet [26].

### L'interface d'utilisation Processing

L'interface d'utilisation de Processing est composée de deux fenêtres distinctes : la fenêtre principale pour l'écriture du code et la fenêtre de visualisation dans laquelle les créations (dessins, animations, vidéos) apparaissent.

La figure suivante montre l'interface principale sur la droite et l'interface de visualisation sur la gauche :



**Figure 4.7:l'interface principale et l'interface de visualisation Processing.**

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

Processing permet de travailler dans plusieurs modes, lui permettant de programmer dans un environnement spécifique à chaque plateforme visée (ex : application, animation pour une page web, application pour tablette Android). L'utilisateur peut changer ce mode à tout moment depuis l'interface, en ayant au préalable sauvegardé le sketch.



Les gestionnaires `setup()` et `draw()` sont deux fonctions prédéfinies très importantes dans Processing :

### La fonction `Setup()` :

La fonction `setup()` est exécutée une fois, lorsque le programme démarre. Il sert à définir les propriétés d'environnement initiales telles que la taille de l'écran et à charger des supports tels que des images et des polices au démarrage du programme. Il ne peut y avoir qu'une fonction de configuration pour chaque programme et ne doit pas être appelée à nouveau après sa première exécution, elle retourne rien. Les variables déclarées dans `setup()` ne sont pas accessibles dans d'autres fonctions, y compris `draw()`.

#### Syntaxe :

```
Void setup() // s'exécute une fois
{
  Instruction 1;
  Instruction 2;
  ...
}
```

### La fonction `Draw()` :

La fonction `draw()` est appelée directement après la configuration, elle exécute en continu les lignes de code contenues dans son bloc jusqu'à ce que le programme soit arrêté ou `noLoop()` soit appelé. `Draw()` s'appelle automatiquement et ne doit jamais être appelé explicitement.

La fonction `draw()` est exécutée à l'intervalle régulier, sa vitesse de répétition est par défaut de 60 fois par seconde.

#### Syntaxe :

```
Void draw() // boucles indéfiniment.
{
  Instruction1;
  Instruction2;
  ...
}
```

## 3. Processing et Arduino

Le circuit électronique Arduino comporte des entrées et sorties sur lesquelles on peut brancher différents appareils. Les capteurs servent à mesurer des informations sur le monde extérieur, alors que les actionneurs permettent d'agir sur celui-ci. On peut utiliser

des capteurs a fin d'influencer ce qui se passe dans un sketch Processing. À l'inverse, on peut utiliser des actuators pour que les instructions qui s'exécutent dans le sketch influencent le monde extérieur.

Notre programme Arduino permet de lire les informations obtenues du capteur qui mesure la qualité de l'air (MQ135), et du capteur de température (SHT75). Ensuite, à partir de ces informations et du traitement fait sur notre sketch, on pourrait contrôler les fenêtres et déclencher la vanne d'arrosage anti incendie pour éteindre le feu.

Pour faciliter la programmation, nous avons créé des fonctions qui permettent de décomposer notre programme en série de sous programmes. Les fonctions utilisées sont les suivantes :

### **La fonction affichage () :**

La fonction affichage () permet d'afficher les image (.GIF, .JPG, .tga, .png) importées par loadImage qui a comme rôle d'importer une image à partir du sous dossier data du sketch. Les images peuvent être affichées dans les espaces 2D et 3D.

### **La fonction create\_fire () :**

C'est la fonction qui met en œuvre des formules mathématiques liés aux particules, des classes servant de particules sont adaptés à la création des feux,des jets d'eau...etc.

### **La fonction update\_fire () :**

Chaque particule a une durée de vie bien définie dans la fonction dédiée,ainsi a la fin de vie de cette particule,cette dernière doit être mise à jour afin que le feu puisse se maintenir.

### **La fonction draw\_fire ():**

Cette fonction sert a afficher les particules composants le feu en reliant toutes les méthodes qui chaque une contrôle un élément de la formule de chaque particule constituant l'incendie, à savoir : les dimensions, la durée de vie, l'accélération, la trajectoire..Etc.

### **La fonction Arrosage\_Anti\_Incendie () :**

C'est la fonction qui permet de simuler l'action du système d'arrosage qui intervient après la détection d'incendie .cette fonction s'exécute automatiquement dès que l'événement incendie est confirmé.

### **La fonction serialEvent(Serial myport) :**

C'est la fonction qui permet de faire la communication entre le Processing et l'Arduino. Elle est appelée lorsque des données sont disponibles. Elle utilise la méthode read () pour capturer ces données. Le serialEvent () peut être configuré avec buffer ()

pour déclencher uniquement après qu'un certain nombre d'éléments de données sont lus et peuvent être configurés avec `bufferUntil ()` pour déclencher uniquement après la lecture d'un caractère spécifique. Le paramètre `this` contient le nom du port où de nouvelles données sont disponibles, mais il est seulement utile quand il y a plus d'une connexion série ouverte et il est nécessaire de faire la distinction entre les deux.

**Syntaxe :**

```
import processing.serial.*;    //Import de la librairie de communication série Processing

Serial myPort;    // Déclaration du port série pour la lecture des données envoyées par
                  l'Arduino.

//Traitements d'initialisation

void setup() {

    MyPort = new Serial ( this,"COM3",19200);    // Ouvrir le port à utiliser au débit de
                                                  données souhaité

    MyPort.bufferUntil ("\n"); // permet de définir un octet spécifique pour le tampon avant
                              d'appeler serialEvent ().

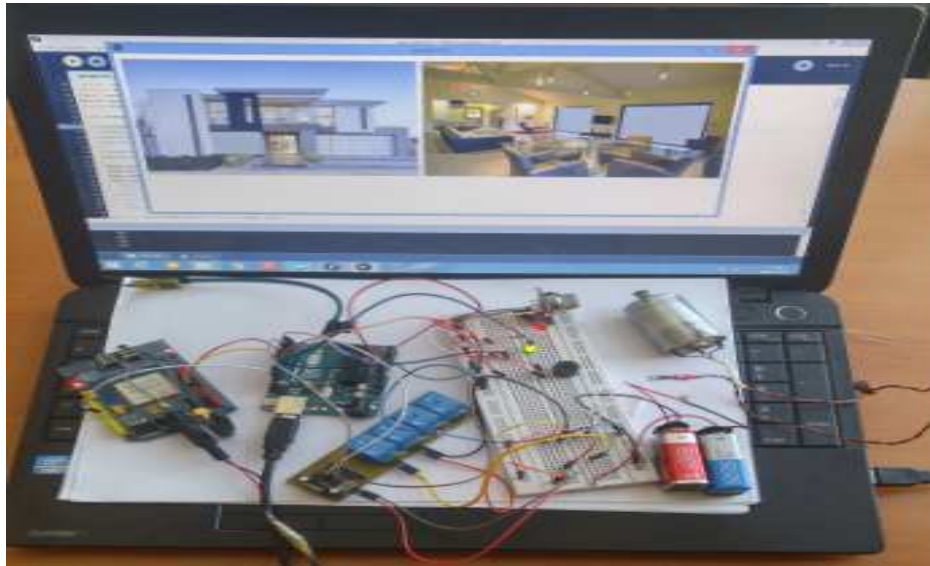
}
```

**4.3. Teste du système**

Dans cette partie nous allons vous présenter le comportement de notre système lors des tests effectués sur ce dernier et pour cela nous avons utilisé une feuille brûlée, du gaz et une flamme.

A l'état inactif toutes les LEDs sont éteintes dans ce cas de figure la protection est désactivé donc même si il y a une fuite de gaz ou un feu qui se déclenche y aura pas de prévention sur le danger.

Une fois le système est alimenté la LED verte clignote et un signal sonore se déclenche pour prévenir que le système est allumé. Ensuite la led verte s'allume et reste figée.

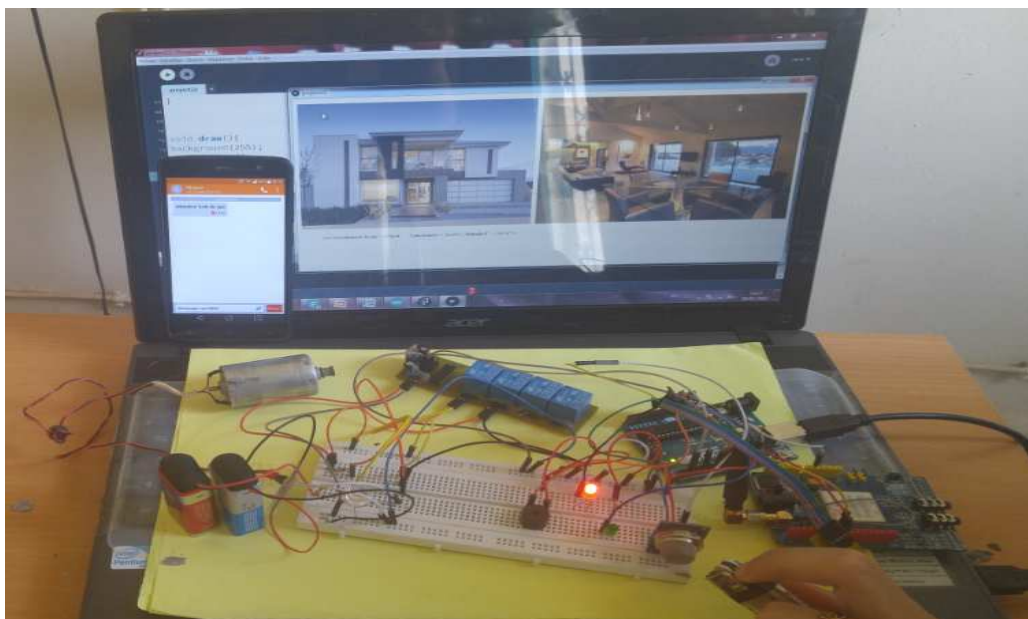


**Figure 4.8: Système actif.**

#### 4.3.1. Teste de capture de fuite de gaz

Lors d'une fuite de gaz le système éteint la LED verte, le buzzer s'active, la LED rouge s'allume et reste figée jusqu'à ce que la qualité de l'air redevienne normale et le moteur tourne afin d'ouvrir les fenêtres entièrement sur la fenêtre de visualisation, ensuite un message d'alerte est envoyé aux propriétaires.

Nous illustrons le comportement du système à la présence de gaz dans figure suivante :

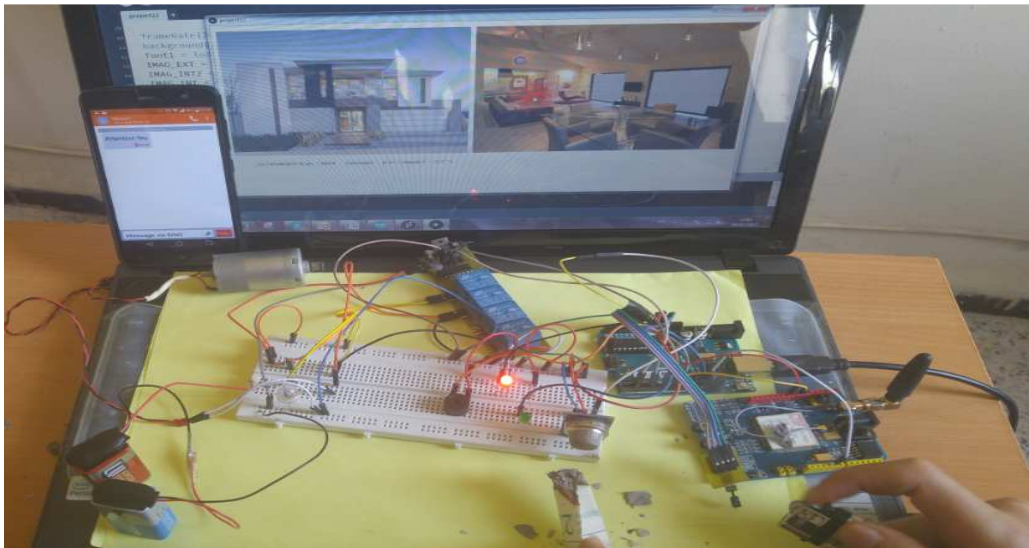


**Figure 4.9: teste de capture de fuite de gaz.**

**4.3.2. Teste de capture de feu**

Le comportement du système lorsqu'un feu se déclenche est illustré dans la figure ci-dessous. Le système éteint la LED verte, allume la LED rouge qui reste figée, ensuite déclenche le système d'extinction d'incendie présent dans la maison et envoie un message d'alerte à l'aide du module GSM qui est branché au système.

La figure suivante montre la réception du message d'alerte, la simulation du feu et le déclenchement de la vanne d'arrosage sur la fenêtre de visualisation de Processing en cas d'un feu :



**Figure 4.10: teste de capture de feu.**

**4.4. Conclusion**

Dans ce chapitre, nous avons présenté en détail la réalisation de notre projet. Nous avons commencé par la description des outils de développement matériels et logiciels. Ensuite, nous avons réalisé quelque test de notre système en présentant des captures d'écran témoignant ses différents comportements.

# **Conclusion générale**

## Conclusion générale et perspectives

---

La « maison intelligente » est au service de nos besoins. La domotique améliore les conditions de vie et le quotidien des individus. Elle a besoin d'une haute habilité technologique pour la maîtrise dans un objectif bien défini et qui est celui d'accompagner les usagers.

La réalisation de la domotique dans ce présent travail nous a permis avant tout de découvrir la technologie Arduino, son fonctionnement pratique et son utilité primordiale.

Le système que nous avons développé représente un prototype fonctionnel et prometteur d'un système de protection et de sécurité d'une maison qui permet de surveiller, d'avertir et d'agir dans le cas d'un danger.

Dans le travail que nous avons entretenu, nous avons pu atteindre notre objectif qui est de créer un système autonome de prévention et de lutte contre les incendies et les fuites de gaz. Il permet de mieux sécuriser en avertissant le propriétaire et les habitants du danger rapidement et d'une manière permanente, diminuer l'étendu des dégâts et minimiser les risques.

Les perspectives seraient donc d'élargir ce système en ajoutant quelques options et fonctionnalités qui peuvent servir dans le cadre de la sécurité :

- Ajouter la possibilité d'appel vocal afin d'appeler la sécurité civil au moment où l'incendie se déclenche.
- Donner la possibilité à utilisateur de gérer ses appareils avec un simple clic à partir d'une interface graphique.
- Gérer le système à partir de la communication sans fil à l'aide du module GSM.



# **Bibliographie**

- [1]: Cours-Modules ASTRE, Z.MAMMERI IRIT-UPS-Toulouse.
- [2]: Ramzi BOULKROUNE, « Les systèmes embarqués », Université de Annaba-Ingénieur d'état en électronique option télécommunication 2009.
- [3]: Les systèmes embarqués introduction, Richard Crisel – Professeur des Université – Université de Rouen.
- [4]: <http://www.technologuepro.com/cours-systemes-embarques/cours-systemes-embarques-introduction.htm>.
- [5]: [Richard.grisel.free.fr/Master\\_OSM/2\\_Introduction\\_Embedded\\_systems.pdf](http://richard.grisel.free.fr/Master_OSM/2_Introduction_Embedded_systems.pdf)
- [6]: [http://www.sigma-tec.fr/domotique/texte\\_definition.html](http://www.sigma-tec.fr/domotique/texte_definition.html)
- [7]: [http://www.m-habitat.fr/electricite/domotique/les-composants-de-la-domotique-439\\_A](http://www.m-habitat.fr/electricite/domotique/les-composants-de-la-domotique-439_A)
- [8]: [crt.a.fr/wp-content/uploads/2013/07/46-Domotique.pdf](http://crt.a.fr/wp-content/uploads/2013/07/46-Domotique.pdf) CRTA.
- [9]: [biotec.ac-dijon.fr/logement/domotique.pdf](http://biotec.ac-dijon.fr/logement/domotique.pdf)
- [10]: De l'automatisme à la domotique, Dossier ressource « Automatisme » par M.Pujades.
- [11]: <http://eskimon.fr/73-arduino-101-presentation>
- [12]: KRAMA ABDELBASSET, GOUGUI ABDELMOUMEN. « Etude et réalisation d'une carte de contrôle par Arduino via le système Androide ».2014/2015.
- [13]: BERTRAND COTTENCEAU. « carte Arduino UNO Microcontrôleur ATmega328 ».
- [14]: ERIK Bartmann . « Le grand livre d'Arduino ». 2eme Edition, page12, 02/01/2014.
- [15]:[http://www.monclubelec.fr/pmwiki\\_reference\\_arduino/pmwiki.php?n=Main.MaterielUno](http://www.monclubelec.fr/pmwiki_reference_arduino/pmwiki.php?n=Main.MaterielUno)
- [16]: <http://www.locoduino/spip.php?article57>
- [17]: CHRISTIAN TARVERNIER. Maitrisez sa programmation et ses cartes d'interfaces (shields). 2ème édition, 2014.

- [18]: HAMADI LYES, HARBANE FAZIA. Conception et réalisation d'un dispositif de surveillance-protection contre l'inondation et l'incendie, option 2015/2016.
- [19]: MASSIMO BANZI et MICHAEL SHILOHDEMARRER, Démarrer avec Arduino- 3<sup>e</sup> édition.
- [20]: <https://projetsdiy.fr/mq135-mesure-qualite-air-polluant-arduino/#.WR17ZTfjLIU>
- [21]: <http://fr.rs-online.com/web/p/capteurs-de-temperature-et-humidite/6675271/>
- [22]: <https://zestedesavoir.com/forums/sujet/4941/des-buzzers-electronique-arduino-qui-ne-buzzent-plus/>
- [23]: <http://www.hobbyist.co.nz/?q=arduino-gsm-module>.
- [24]: [https://www.sonelec-musique.com/electronique\\_theorie\\_relais.html](https://www.sonelec-musique.com/electronique_theorie_relais.html)
- [25]: [www.fritzing.org](http://www.fritzing.org).
- [26]: <http://fr.flossmanuals.net/processing/>.
- [27]: SIMON BNDRAULT, HIPPOLYTE WEISSLINGER, premiers pas en informatique embarquée, 19 juin 2014.
- [28]: <http://www.domotique-info.fr/2013/11/debuter-en-domotique/>