**MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEURE ET DE LA RECHERCHE SCIENTIFIQUE**

**UNIVERSITÉ Dr YAHIA FARÈS DE MÉDÉA**

**FACULTÉ DE TECHNOLOGIE**

**DEPARTEMENT DE GENIE MÉCANIQUE**

**Mémoire de fin d'étude**

**En vue de l'obtention du diplôme de Master**

**Filière Électromécanique**

**Spécialité**

# Mécatronique

**Présenté par :**

**Mr DJEDDI Ahmed**

**THÈME**

# Programmation et simulation d'un système embarqué pour la détection des gaz dangereux

**Soutenu en Juillet 2021**

Encadreur : Dr REBHI Redha

Année universitaire 2020-2021

# Remerciements

Tout d’abord, je remercie Dieu de m’avoir donné le courage et la volonté de terminer ce modeste travail.

J’exprime mes vifs remerciements pour Monsieur **REBHI Redha** et Monsieur **ABUDURA Salam** pour leurs accueils chaleureux, leurs disponibilités, leurs soutiens et leurs précieux conseils tout au long de cet humble travail.

J'adresse mes plus vifs remerciements à Mme **Boudjedir Imen** pour son mémoire de fin d'études, qui s'intitule « **un système embarqué pour la détection des gaz dangereux** », sur laquelle je me suis appuyé dans mon travail.

Enfin, un grand merci à tous les enseignants et le personnel administratif de la faculté de technologie, et particulièrement à ceux que j’ai eu le plaisir d’avoir en parcours de Master qui n’ont ménagé aucun effort pour le bon déroulement de notre formation.

Que toute personne ayant contribué de près ou de loin à la réalisation de ce travail trouve ici toute ma gratitude.

DJEDDI Ahmed

# Dédicace

Avant tout, je dédie ce travail avec toute ma gratitude et mon affection à vous chers parents, qui m'avez tant donné et encouragé, je vous dédie ce mémoire avec tout mon amour et mon respect, en espérant vous satisfaire avec ma réussite.

A mes frères et sœurs et leurs enfants,
A toute la famille grande et petite,
A mes amis, mes voisins, mes collègues,

A nos maîtres d'écoles, nos professeurs de collège, de lycée et nos enseignants de la faculté de technologie de l'université Dr YAHIA FARÈS de Médéa et nos enseignants de l'université de Tlemcen, A tous ceux qui se sont sacrifiés pour notre instruction et la construction de notre savoir.

Également, à tous mes amis de parcours de master dont j'apprécie beaucoup l'amitié.

A tous qui ont contribués de près ou de loin à l'achèvement de ce modeste travail.

DJEDDI Ahmed.

# Table des matières

**Remerciements** ..... **i**

**Dédicace** ..... **ii**

**Table des matières** ..... **iii**

**Liste des figures** ..... **vi**

**Liste des tableaux** ..... **ix**

**Liste des abréviations** ..... **x**

**Introduction générale** ..... **1**

**Chapitre I. Motivation pour un système embarqué de détection des gaz dangereux** ..... **3**

I.1 Asphyxies au monoxyde de carbone en Algérie ..... 3

I.1.1 Introduction ..... 3

I.1.2 L'effet de concentration du gaz CO sur la personne ..... 4

I.2 Les systèmes embarqués ..... 6

I.2.1 Introduction ..... 6

I.2.2 Définition d'un système embarqué ..... 6

I.2.3 Caractéristiques d'un système embarqué ..... 6

I.3 Conclusion ..... 10

**Chapitre II. Circuits de commande et de mesure** ..... **11**

II.1 Circuit de commande Arduino ..... 11

II.1.1 Introduction ..... 11

II.1.2 Présentation générale de module Arduino ..... 12

II.1.3 Description technique ..... 12

II.1.3.1 Alimentation ..... 12

II.1.3.2 Horloge ..... 13

II.1.3.3 Reset ..... 14

II.1.3.4 Entrées/sorties ..... 14

II.1.3.5 Mémoire ..... 15

II.1.4 Les gammes de la carte Arduino ..... 17

II.1.5 La carte Arduino Uno ..... 18

II.1.5.1 Présentation ..... 18
II.1.5.2 Les Caractéristiques techniques ..... 21
II.1.5.3 Avantages ..... 21
II.2 Circuit de mesure : les capteurs de gaz ..... 22
II.2.1 Présentation générale ..... 22
II.2.2 Principaux paramètres d'un capteur de gaz ..... 23
II.2.2.1 La sensibilité ..... 23
II.2.2.2 La réponse relative ..... 23
II.2.2.3 La sélectivité ..... 24
II.2.2.4 Temps de réponse ..... 24
II.2.2.5 Temps de recouvrement ..... 25
II.2.3 Les différents types des capteurs MQ ..... 25
II.2.3.1 Capteur MQ2 ..... 25
II.2.3.2 Capteur MQ5 ..... 27
II.2.3.3 Capteur de gaz MQ7 ..... 28
II.3 Conclusion ..... 31
**Chapitre III. Réalisation de l'application de détection des gaz ..... 32**
III.1 Introduction ..... 32
III.2 Partie Hardware ..... 32
III.2.1 Afficheur LCD (Liquide Crystal Display) ..... 32
III.2.2 LED ..... 35
III.2.3 Buzzer ..... 36
III.2.4 Moteur pas à pas ..... 37
III.2.5 Driver en pont H L298N ..... 38
III.2.5.1 Détails techniques ..... 39
III.2.5.2 Activation moteur ..... 40
III.2.5.3 Rotation Moteur DC A/B ..... 40
III.2.5.4 Rotation Moteur pas-à-pas ..... 40
III.2.6 Ventilateur ..... 41
III.2.7 Module GSM SIM 800 ..... 42
III.2.7.1 Principales caractéristiques ..... 42

III.2.8 Application globale ..... 43
III.2.8.1 Circuit de simulation (Isis-Proteus) ..... 44
III.2.8.2 Ensemble de l'application pratique ..... 45
III.3 Partie software ..... 46
III.3.1 Programme Principal ..... 46
III.3.2 Fonction Lecture ..... 48
III.3.3 Fonction Calcul PPM ..... 48
III.3.4 Fonction Alarme ..... 49
III.3.5 Code programme Arduino IDE ..... 50
III.3.6 Interface et simulation ..... 54
III.4 Conclusion ..... 56
**Conclusion générale ..... 57**
**Bibliographie ..... 58**
**Index ..... 60**

# Liste des figures

Figure I:1 Système embarqué typique [3] 10

Figure II:1 Schéma explicatif des différentes broches du microcontrôleur ATmega328 17

Figure II:2 Différentes Cartes Arduino [26] 17

Figure II:3 Description détaillée du brochage de la carte Arduino UNO 19

Figure II:4 Schéma représentatif des temps de réponse et de recouvrement d'un capteur de gaz à base de semi-conducteur 24

Figure II:5 Caractéristiques de sensibilité typiques du MQ-2 pour plusieurs gaz. [16] 27

Figure II:6 Caractéristiques de sensibilité typiques du MQ-5 pour plusieurs gaz. [16] 28

Figure II:7 Caractéristiques de sensibilité typiques du MQ-7 pour plusieurs gaz. [19] 30

Figure II:8 Capteur de gaz 30

Figure II:9 Schéma électrique des capteurs MQx utilisés 30

Figure III:1 Brochage de l'afficheur LCD 16x2 33

Figure III:2 Schéma de câblage d'un LCD 16x2 avec Arduino 34

Figure III:3 Schéma de câblage d'un I²C avec Arduino 35

Figure III:4 Schéma de câblage d'un LED avec Arduino 36

Figure III:5 Schéma de câblage buzzer avec Arduino 36

Figure III:6 Moteur pas à pas 37

Figure III:7 Modèle conceptuel du moteur pas à pas 38

Figure III:8 DriverL298N 38

Figure III:9 Brochage de moteur à courant continue et driver avec Arduino 40

Figure III:10 Brochage de moteur pas à pas et driver avec Arduino 41

Figure III:11 Ventilateur 41

Figure III:12 Schéma de branchement du ventilateur avec Arduino UNO 42

Figure III:13 GSM-GPRS sim800 ____ 43

Figure III:14 Schéma synoptique du montage principal ____ 44

Figure III:15 Schéma du circuit de simulation (Isis-Proteus) ____ 45

Figure III:16 L'application virtuelle du circuit de détection de gaz dans (Isis-Proteus) ____ 46

Figure III:17 Organigramme principale de détection de gaz ____ 47

Figure III:18 Organigramme de lecture ____ 48

Figure III:19 Organigramme « calcul PPM » ____ 49

Figure III:20 Organigramme de la fonction d'alarme ____ 49

Figure III:21 1 partie du programme ____ 50

Figure III:22 2 partie du programme ____ 50

Figure III:23 3 partie du programme ____ 51

Figure III:24 4 partie du programme ____ 51

Figure III:25 5 partie du programme ____ 52

Figure III:26 6 partie du programme ____ 52

Figure III:27 7 partie du programme ____ 53

Figure III:28 8 partie du programme ____ 53

Figure III:29 9 partie du programme ____ 54

Figure III:30 Interface du circuit ____ 54

Figure III:31 Interface du circuit au cours de simulation ____ 55

Figure III:32 Interface du circuit au cours calibration ____ 55

Figure III:33 Détection de l'air pure ____ 55

Figure III:34 Détection de l'air malsain ____ 56

Figure III:35 Détection de l'air pollué ____ 56

# Liste des tableaux

Tableau I:1 L'effet de concentration du gaz CO.....5

Tableau II:1 Différentes gammes de la carte Arduino.[10].....17

Tableau II:2 Caractéristiques de l'Arduino Uno. [13].....21

Tableau II:3 Condition de travail standard. [16].....25

Tableau II:4 Condition environnementales.[16].....25

Tableau II:5 Caractéristique de sensibilité.[16].....26

Tableau II:6 Structure et configuration, circuit de mesure de base. [16].....26

Tableau II:7 Caractéristiques MQ5. [16].....27

Tableau II:8 Condition de travail standard. [19].....29

Tableau II:9 Conditions environnementales. [19].....29

Tableau II:10 Caractéristique de sensibilité. [19].....29

Tableau III:1 Branchement de l'afficheur LCD.....33

Tableau III:2 Fiche technique du driver L298N. [24].....39

# Liste des abréviations

| | |
|---|---|
| PPM | Particule par million |
| OMS | Organisation mondiale de la santé |
| TLV | Threshold Limit Value |
| TWA | Time Weighted Average |
| IDHL | Immediately dangerous for life and health |
| PWM | Pulse Width Modulation (Modulation de Largeur d'Impulsion) |
| SRAM | Static Random Access Memory (mémoire vive statique) |
| EEPROM | Electrically-Erasable Programmable Read-Only Memory (mémoire morte effaçable électriquement et reprogrammable) |
| CAO | Conception assisté par ordinateur |

# Introduction générale

La sécurité humaine joue un rôle important dans le monde d'aujourd'hui et il est nécessaire d'installer des systèmes de sécurité efficaces dans les maisons et les environnements industriels ainsi que dans les établissements d'enseignement.

Ces dernières années, l'Algérie a enregistré un nombre impressionnant de décès par suffocation au gaz dangereux. Des centaines de citoyens sont morts d'empoisonnement au monoxyde de carbone (CO), surtout chaque hiver. Les causes de cette catastrophe sont multiples, dont une peut être citée : une mauvaise ventilation, une contrefaçon, une mauvaise installation et le non-respect des normes d'entretien. En plus de ce risque, il existe également un risque lié à la fuite de gaz naturel inflammable CH4, qui présente un risque d'explosion et met en danger les personnes et les bâtiments.

Afin de contrer ces risques, le développement de la microélectronique et de la micro-informatique dans les circuits modernes, en particulier la mémoire utilisera un microcontrôleur pour concevoir des systèmes embarqués à bas coût basés sur les cartes Arduino et divers types de capteurs de gaz lorsque ces substances toxiques fuient ou sont présentes, comme le monoxyde de carbone CO et le méthane CH4, en minimisant la source du problème et en alertant l'utilisateur en amont afin qu'il puisse réagir à temps et avant que les dégâts se produisent.

Ce document se compose de trois chapitres :

Le premier chapitre fournit une description générale du système embarqué et quelques statistiques sur les intoxications au monoxyde de carbone en Algérie.

Le deuxième chapitre se concentre sur une étude détaillée des principales organes utilisées pour mettre en œuvre l'application considérée. Quant aux contrôle-commande, nous allons nous pencher sur le modèle de base de l'Arduino UNO, qui est la carte du microcontrôleur Arduino qui détaille le schéma de structure, l'environnement de programmation et le principe de fonctionnement et simplifie son utilisation. En complément, une étude de la famille de capteurs de gaz MQx pour la partie acquisition, avec la présentation des caractéristiques techniques liées à la plage de mesure.

Le chapitre trois décrit les divers composants matériels et logiciels permettant de développer et d'activer des applications de mesure de concentration de gaz pour avertir lorsqu'un seuil acceptable est dépassé. Chaque étape comprend la programmation et la validation partielle, cette dernière incluant la simulation et la réalisation virtuelle.

Enfin, nous concluons par une conclusion générale qui résume les bénéfices de notre étude : compte tenu des différentes observations et résultats obtenus, des difficultés rencontrées en programmation et simulation et des perspectives d'amélioration de l'application.

# Chapitre I. **Motivation pour un système embarqué de détection des gaz dangereux**

## I.1 Asphyxies au monoxyde de carbone en Algérie

### I.1.1 Introduction

La suffocation mortelle par le CO a atteint des niveaux alarmants en Algérie. À l'approche de l'hiver, des centaines de citoyens meurent d'une intoxication au monoxyde de carbone (CO).

Par exemple, depuis début 2010, 123 personnes ont suffoqué. En 2009, 253 décès ont été enregistrés par les services de protection civile. Il ne se passe pas une année sans que ce danger n'emporte avec lui des familles entières à travers les différentes régions du pays. Ce gaz invisible, inodore, suffocant et indétectable surprend la victime, surtout lorsqu'elle dort la nuit. Il est incorporé dans la circulation sanguine au lieu de l'oxygène et meurt en une heure. Dans la plupart des cas, même une personne consciente ne peut pas réagir.

Selon les statistiques, la présence de 0,1% CO dans l'air provoque la mort en 1 heure, 1% en 15 minutes, et la mort instantanée en 10%. C'est le niveau de danger qui survient lorsque ce gaz est libéré dans l'atmosphère. Il y a plusieurs raisons principales à l'étouffement du dioxyde de carbone. Un radiateur ancien et défectueux. Le manque de ventilation et de systèmes de ventilation dans la maison cause également des dommages personnels et matériels aux locataires. Lorsque le monoxyde de carbone n'est pas immédiatement expulsé de la maison, il remplace l'oxygène et est donc irréparable ! Cela devrait être ajouté à la falsification de l'équipement de chauffage utilisé. Ceux-ci ne répondent pas aux normes de sécurité requises. Il est vendu à bas prix sur le marché, mais la qualité n'est pas garantie.

Face à cette situation, plusieurs campagnes de sensibilisation ont été lancées dans différentes Wilayah. Le but est de prodiguer des conseils préventifs aux personnes afin qu'elles puissent se protéger des risques d'intoxication au monoxyde de carbone qu'elles présentent si elles ne font pas attention et ne respectent pas les consignes de sécurité. [5] [6]

Le monoxyde de carbone est le résultat d'une combustion inadéquate, quelle que soit la source d'énergie utilisée : butane, essence, charbon, gaz naturel, pétrole ou propane. "C'est la densité est proche de la densité de l'air, donc il se propage très rapidement dans l'environnement." Il existe deux types de dépendance. La forme faible est dite « chronique » et se manifeste par des

maux de tête, des nausées, de la confusion et des malaises. Elle est en retard et peut ne pas montrer de symptômes immédiatement. L'autre est aiguë et rapide, provoquant des vertiges, une perte de conscience, une faiblesse musculaire, des troubles du comportement, voire le coma et la mort. [4]

On peut aussi ajouter au risque du monoxyde de carbone CO, le risque lié aux fuites du gaz naturel, le méthane CH4 qui alimente nos installations domestiques. La fuite de ce gaz et sa nature inflammable, présente un danger majeur sur les personnes et les immeubles, où plusieurs explosions enregistrées suite à une étincelle issue à l'allumage de l'éclairage en présence d'une fuite importante du méthane avec un manque d'aération.

Il peut également augmenter le risque de monoxyde de carbone CO dû à la fuite de méthane CH, un gaz naturel fourni aux installations domestiques. Cette fuite de gaz et son inflammabilité représentent un grave danger pour les personnes et les bâtiments. Plusieurs explosions ont été enregistrées à la suite d'étincelles provoquées par l'allumage d'appareils d'éclairage avec de sévères pertes de méthane par manque de ventilation.

De tout ce qui précède, nous avons obtenu les avantages d'un système embarqué à faible coût pour détecter, mesurer et avertir de ces gaz dangereux avant qu'une catastrophe ne se produise.

### I.1.2 L'effet de concentration du gaz CO sur la personne

Le taux habituel de CO dans l'air ambiant est d'environ 0,2 ppm (particule par million). L'Organisation mondiale de la Santé à défini, pour l'ensemble de la population, y compris les femmes enceintes et les personnes âgées atteintes d'affections cardiaques ou respiratoires (connues ou non), des valeurs de référence considérées comme inoffensives en fonction de la durée de l'exposition

Les niveaux typiques de CO dans l'air ambiant sont d'environ 0,2 ppm (particules par million). L'Organisation mondiale de la santé a identifié des références à considérer comme inoffensives pendant la période d'exposition pour la population générale, y compris les femmes enceintes et les personnes âgées atteintes de maladies cardiaques ou respiratoires (connues ou non), des valeurs de référence considérées comme inoffensives en fonction du temps d'exposition :

- 10 mg/m3 (10 ppm) pendant 8 heures.

- 30 mg/m3 (25 ppm) pendant 1 heure.
- 60 mg/m3 (50 ppm) pendant 30 min.
- 100 mg/m3 (90 ppm) pendant 15 min.

La gravité de l'intoxication dépend de la concentration de CO dans l'air et de la durée d'exposition :

Tableau I:1 L'effet de concentration du gaz CO

| CO (ppm) | % CO dans l'air | Symptômes |
|---|---|---|
| 100 | 0.01 | |
| 200 | 0.02 | Maux de tête, vertiges, nausées, fatigue. |
| 400 | 0.04 | Maux de tête intenses, danger de mort après 3 heures. |
| 800 | 0.08 | Maux de tête, vertiges, nausées. Perte de connaissance en 45 min, décès après 2-3 heures. |
| 1600 | 0.16 | Symptômes sévères après 20 min, décès endéans l'heure. |
| 3200 | 0.32 | Maux de tête, vertiges, nausées après 5 min, perte de connaissance après 30 min. |
| 6400 | 0.64 | Céphalées et vertiges après 1 à 2 min, perte de connaissance après 10-15 min. |
| 12800 | 1.28 | Perte de connaissance immédiate, décès en 1 à 3 minutes. |

Les réglementations du travail fixent souvent la concentration maximale à laquelle les travailleurs peuvent être exposés en toute sécurité. Ils ne sont valables qu'au travail. Il est à noter qu'une quantité acceptable de CO pour des travailleurs en bonne santé peut être dangereuse pour les personnes âgées et les enfants.

La TLV-TWA (Threshold Limit Value - Time Weighted Average) représente la concentration moyenne considérée comme inoffensive pour un travailleur qui y est exposé 40 heures par semaine pendant toute sa carrière. Elle est de 25 ppm.

La TLV-C (Threshold Limit Value - Ceiling) représente la valeur plafond qui ne peut jamais être dépassée. Dans le cas du CO, la TLV-C est de 200 ppm.

La TLV-IDHL (Threshold Limit Value –Immediately dangerous for life and health) représente la concentration à partir de laquelle l'évacuation immédiate s'impose. Bien qu'une évacuation prenne en général moins de 30 minutes, pour des raisons de sécurité, les valeurs TLV-IDHL

sont basées sur les effets d'une exposition durant 30 minutes. Elle est de 1.500 ppm dans le cas du CO. [7].

## I.2 Les systèmes embarqués

### I.2.1 Introduction

De nos jours, les systèmes embarqués nous entourent et nous sommes littéralement envahis par eux. Il suffit de regarder autour de soi au quotidien pour s'en rendre compte. On retire de l'argent dans un distributeur, c'est un système embarqué. On prend sa voiture, le régulateur de vitesse, la direction assistée électrique, le système de contrôle de trajectoire c'est des systèmes embarqués. Le pilote automatique dans un avion est un système embarqué, et les exemples ne manquent pas.

Dans divers domaines, au cours de ces dernières années, le progrès technologique a fait que beaucoup de fonctions réalisées manuellement, mécaniquement et hydrauliquement se dématérialisent au profit de l'électronique et de l'informatique embarquée. Un confort considérable auquel il est de plus en plus difficile de renoncer. Les systèmes électroniques sont de plus en plus présents dans la vie courante. Les ordinateurs et micro-ordinateurs sont des systèmes électroniques bien connus. Mais l'électronique se trouve maintenant embarquée dans de très nombreux objets usuels : les téléphones, les agendas électroniques, les voitures.

Ce sont ces systèmes électroniques enfouis dans les objets usuels qui sont appelés systèmes embarqués.

Les systèmes embarqués font intervenir des technologies différentes, en plus de comporter une partie logicielle et une partie matérielle. [1]

### I.2.2 Définition d'un système embarqué

Un système embarqué ou « Embedded system » en anglais, tout système conçu d'une architecture d'ordinateur pour résoudre un problème ou une tache spécifique mais n'est pas un ordinateur d'usage général.

### I.2.3 Caractéristiques d'un système embarqué

- Utilise généralement un microprocesseur combiné avec d'autres matériels et logiciel pour résoudre un problème de calcul spécifique.

- Système électronique et informatique autonome ne possédant pas des entrées-sorties standards.

- Le système matériel et l'application sont intimement liés et noyés dans le matériel et ne sont discernables comme dans un environnement de travail classique de type PC.

- N'est pas visible en tant que tel, mais est intégré dans un équipement doté d'une autre fonction ; ou dit aussi que le système est enfoui, ce qui traduit plus fidèlement le terme anglais « Embedded ».

- Une faible barrière existe entre les systèmes embarqués et les systèmes temps réel (un logiciel embarqué n'a pas forcément de contraintes temps réel).

- La conception de ces systèmes est fiable (avions, système de freinage ABS) à cause de leurs utilisations dans des domaines à fortes contraintes mais également parce que l'accès au logiciel est souvent difficile une fois le système fabriqué.

- Les microprocesseurs utilisés s'étendent depuis de simples microcontrôleurs 8bits aux 64bit les plus performants et les plus sophistiqués.

- Le logiciel système inclus s'étend d'un petit programme à un grand logiciel d'exploitation en temps réel (RTOS) avec une interface utilisateur graphique (GUI). Typiquement, le logiciel système inclus doit répondre aux événements d'une manière déterministe et devrait toujours être opérationnel. [2]

▪ Fonctionnement en Temps Réel

Le Temps Réel est un concept un peu vague. On pourrait le définir comme : "Un système est dit Temps Réel lorsque l'information après acquisition et traitement reste encore pertinente". Cela veut dire que dans le cas d'une information arrivant de façon périodique (sous forme d'une interruption périodique du système), les temps d'acquisition et de traitement doivent rester inférieurs à la période de rafraîchissement de cette information. Pour cela, il faut que le noyau ou le système Temps Réel soit déterministe et préemptif pour toujours donner la main durant le prochain tic à la tâche de plus forte priorité prête. [3]

▪ Généralement, un système embarqué doit respecter :

- Des contraintes temporelles fortes (Hard Real Time).
- On y trouve enfoui un système d'exploitation ou un noyau Temps Réel (Real Time Operating System, RTOS).
- Réactivité : des opérations de calcul doivent être faites en réponse à un événement extérieur (interruption matérielle).
- La validité d'un résultat (et sa pertinence) dépend du moment où il est délivré.
- Rater une échéance va causer une erreur de fonctionnement :
  - ✓ Temps Réel dur : plantage.
  - ✓ Temps Réel mou : dégradation non dramatique des performances du système.
- Beaucoup de systèmes sont « multi rate » : traitement d'informations à différents rythmes.

▪ Faible encombrement, faible poids

- Électronique « IoT : Internet of Things », applications portables où l'on doit minimiser la consommation électrique (bio instrumentation...).
- Difficulté pour réaliser le packaging afin de faire cohabiter sur une faible surface : électronique analogique, électronique numérique, RF sans interférences.

▪ Faible consommation

- Batterie qui dure plusieurs heures, voire des jours.

▪ Environnement

- Température, vibrations, chocs, variations d'alimentation, interférences RF, corrosion, eau, feu, radiations.
- Prise en compte des évolutions des caractéristiques des composants en fonction de la température, des radiations...
- Fonctionnement critique pour la sécurité des personnes (Sûreté)

- Le système doit toujours fonctionner correctement.

- Sûreté à faible coût avec une redondance minimale.

- Sûreté de fonctionnement du logiciel.

- Système opérationnel même quand un composant électronique lâche.

▪ Contrainte de prix (Faible coût)

- Beaucoup de systèmes embarqués sont fabriqués en grande série et doivent avoir des prix de revient extrêmement faibles, ce qui induit :

  ✓ Une faible capacité mémoire.

  ✓ Un petit processeur (8 bits). Petit mais en grand nombre.

▪ La consommation est un point critique pour les systèmes avec autonomie.

- Une consommation excessive augmente le prix de revient du système embarqué car il faut alors des batteries de forte capacité.

- Optimisation du prix de revient.

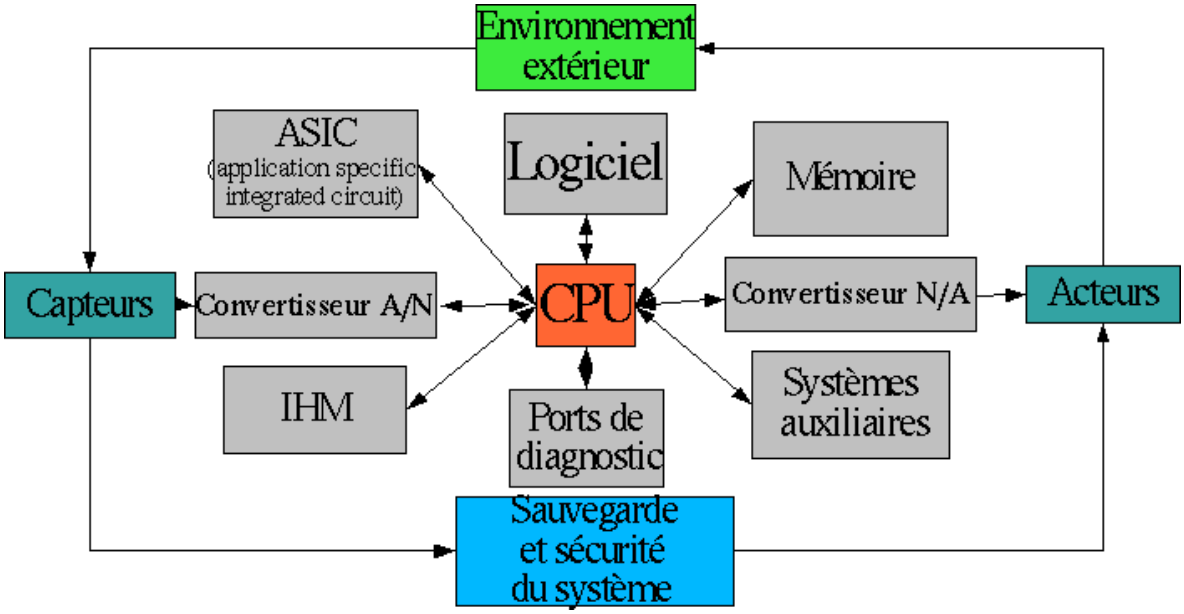



Figure I:1 Système embarqué typique [3]

## I.3 Conclusion

Dans ce chapitre, nous avons fournie une présentation générale de l'asphyxie au monoxyde de carbone en Algérie, et l'effet de concentration du gaz CO sur la personne. Aussi, une description générale sur les systèmes embarqués et leur caractéristique.

# Chapitre II. **Circuits de commande et de mesure**

## II.1 Circuit de commande Arduino

### II.1.1 Introduction

Les cartes Arduino sont conçues pour le prototypage et la simulation de cartes électroniques pour l'informatique embarqué. Ces cartes offrent un accès facile et peu coûteux à l'ordinateur intégré. De plus, il est totalement néfaste du point de vue de la source (open source) et du matériel (open hardware). Par conséquent, vous pouvez recommencer avec votre carte Arduino pour l'améliorer ou supprimer des fonctionnalités inutiles de votre projet.

Le langage Arduino se distingue des langages utilisés dans l'industrie de l'informatique embarquée de par sa simplicité. En effet, beaucoup de librairies et de fonctionnalités de base occulte certains aspects de la programmation de logiciel embarquée afin de gagner en simplicité. Cela en fait un langage parfait pour réaliser des prototypes ou des petites applications dans le cadre de hobby. [8]

L'Arduino fournit un environnement de développement s'appuyant sur des outils open source comme interface de programmation. L'injection du programme déjà converti par l'environnement sous forme d'un code « HEX » dans la mémoire du microcontrôleur se fait d'une façon très simple par la liaison USB. En outre, des bibliothèques de fonctions "clé en main" sont également fournies pour l'exploitation d'entrées-sorties. Cette carte est basée sur un microcontrôleur ATmega 328 et des composants complémentaires. La carte Arduino contient une mémoire morte de 1 kilo. Elle est dotée de 14 entrées/sorties digitales (dont 6 peuvent être utilisées en tant que sortie PWM), 6 entrées analogiques et un cristal à 16 MHz, une connexion USB et Possède un bouton de remise à zéro et une prise jack d'alimentation.[17]

Le système Arduino permet de :

- Contrôler les appareils domestiques
- Fabriquer votre propre robot
- Faire un jeu de lumières
- Communiquer avec l'ordinateur

- Télécommander un appareil mobile (modélisme)
- Etc… [9]

### II.1.2 Présentation générale de module Arduino

Arduino est une série d'électronique open source, principalement basée sur des microcontrôleurs du fabricant Atmel. Ces circuits intègrent les composants nécessaires pour rendre le microcontrôleur rapide et facile à utiliser. Cette simplification se veut accessible à tous, y compris pour la création et la programmation d'objets ou de dispositifs interactifs. Ces objets peuvent contenir toutes sortes de capteurs d'activation, voyants lumineux ou interrupteurs.

Entre autres, les cartes Arduino sont équipées de connecteurs standardisés pour brancher des modules compatibles appelés 'shields'. Ces derniers sont des circuits d'une taille plus ou moins semblable à celle de l'Arduino et qui viennent s'empiler sur ces connecteurs. Ils proposent des extensions matérielles qui permettent d'ajouter des fonctionnalités originales à son projet. En plus de ces connecteurs, les cartes possèdent toutes une connectique USB permettant de programmer facilement le microcontrôleur qu'elles embarquent.

Arduino propose en plus gratuitement un environnement de développement intuitif avec lequel la création de programmes est simple, même pour les débutants. De plus, cet environnement est compatible avec les principaux systèmes d'exploitation : Windows, Mac OS et Linux.

Il existe en plus une grande variété de cartes Arduino qui possède chacune leurs avantages et inconvénients. Il faudra alors adapter son choix en fonction du projet que l'on souhaite réaliser.[12]

### II.1.3 Description technique

Comme énoncé précédemment, il existe beaucoup de cartes Arduino différentes, mais elles possèdent toutes des éléments en commun.

#### II.1.3.1 Alimentation

Le microcontrôleur présent en général sur les cartes Arduino est alimenté par une tension de 5V. En fonction du modèle de la carte, cette tension peut être fournie soit par une des prises

d'alimentation présentes sur la carte, soit par la prise USB utilisée pour la connecter à un ordinateur. La valeur de la tension à fournir sue une des prises d'alimentation doit être comprise entre 7 et 12V mais cette tension n'a pas besoin d'être stabilisée en raison de la présence d'un régulateur de tension sur la carte.

Il peut exister deux prises d'alimentation sur une carte Arduino. La première prise est souvent la plus simple à mettre en place. Il s'agit d'une simple prise jack standard, présente sur les cartes Arduino Uno et Arduino Mega, sur laquelle il suffit de raccorder un bloc d'alimentation fournissant la tension requise pour démarrer sa carte. Cette prise n'est pas présente sur toutes les cartes Arduino, notamment pour les cartes les plus petites pour lesquelles ces prises n'étaient pas vitales.

La seconde prise en revanche est présente sur tous les modèles. Il s'agit d'une broche nommée Vin présente sur un des côtés de la carte. Il suffit alors de raccorder la branche positive de son alimentation à cette broche, comme on le ferait pour une pile, et la branche négative sur une broche GND.

La carte possède également deux broches de tensions stabilisées de 3,3V et de 5V. Ces deux tensions sont généralement utilisées pour alimenter les shields qui viendront s'ajouter sur l'Arduino mais elles peuvent être utilisées pour d'autres circuits.

### II.1.3.2 Horloge

L'horloge détermine la fréquence ou la rapidité à laquelle les instructions seront exécutées par le microcontrôleur. Cette vitesse peut varier d'un microcontrôleur à l'autre et est exprimée en hertz (Hz). Si les PC et Mac peuvent atteindre des fréquences de plusieurs gigahertz, la fréquence est en revanche beaucoup plus faible pour les microcontrôleurs, avec des fréquences de quelques mégahertz seulement. Cette fréquence est déterminée par un oscillateur à quartz ou résonateur céramique. Ces composants électroniques sont les seuls capables de délivrer un signal électrique carré stable (signal alternant rapidement entre deux valeurs hautes et basses) nécessaire au bon fonctionnement du microcontrôleur. Sur les cartes Arduino, l'horloge est généralement cadencée à 16 MHz.

### II.1.3.3 Reset

La fonction reset est une fonction physique permettant au microcontrôleur, comme son nom l'indique, de réinitialiser son état. Un microcontrôleur exécute en effet les instructions contenues dans sa mémoire de manière cyclique et infinie. Il n'est ainsi pas rare, notamment lors de la conception de son circuit et infinie. Il n'est ainsi pas rare, notamment lors de la conception de son circuit, que le programme soit soumis à une erreur bloquant le bon déroulement du programme.

La fonction reset permet alors d'interrompre le programme et de le remettre à zéro. Il est possible d'effectuer cette même opération en coupant et en remettant le courant sur la carte Arduino. Lors de sa mise sous tension, elle commencera toujours par reprendre le programme du début, ce qui équivaut donc à une réinitialisation. Il est toutefois plus aisé d'utiliser cette fonction reset matérialisée sur la carte par un bouton poussoir.

### II.1.3.4 Entrées/sorties

Les entrées et sortie sont les moyens que possède le microcontrôleur pour communiquer avec le monde extérieur. Typiquement, on appelle entrées / sorties les pattes métalliques qui donnent cette forme familière aux puces électroniques. Ces pattes reçoivent ou émettent des signaux logiques qui peuvent alors être interprétés par le microcontrôleur ou d'autres circuits.

Les cartes Arduino facilitent l'accès à ces entrées / sorties en les reportant sur le pourtour du circuit. En utilisant des connecteurs femelles standards, d'un pas de 2,54mm, il devient très simple d'ajouter ou de retirer des composants sans devoir forcément passer par une étape de soudure. C'est sur ces connecteurs que se branchent les shields et les différents circuits additionnels employés.

Le nombre de broches peut varier d'un microcontrôleur à l'autre, et même d'une carte Arduino à l'autre et il peut être intéressant d'étudier les différentes cartes avant de choisir celle adaptée à son projet. Généralement, ces broches fonctionnent sous une tension maximale de 5V et ne peuvent recevoir et fournir que 40 mA de courant.

On peut ranger ces broches par catégorie suivant leur fonctionnalité :

- Les broches digitales : ces broches fournissent des données digitales sous forme de signaux logiques. Elles ne peuvent donc contenir que deux valeurs, un 0 logique correspondant à une absence de tension et une 1 logique correspondant à une tension de 5V. certaines de ces broches digitales possèdent une seconde fonctionnalité. C'est le cas notamment des broches PWM (Pulse Width Modulation ou Modulation de Largeur d'Impulsion) ou de certaines broches de communication. Sur les cartes Arduino, les broches digitales sont généralement nommées par un simple numéro, sauf dans le cas de l'Arduino Nano sur laquelle ce numéro est précédé de 1 lettre D.

- Les broches analogiques : ces broches ne fonctionnent qu'en entrée, c'est-à-dire qu'elles ne servent qu'à la lecture de données. Elles acceptant des tensions comprises entre 0 et 5V. ces tensions sont ensuite utilisées par un convertisseur analogique/numérique qui s'occupe de traduire ces valeurs physiques en données numériques sur 10 bits, soit des données comprises entre 0 et 1023. Les broches analogiques sont nommées sur les cartes Arduino par un chiffre précédé de la lettre A.

- Les broches d'alimentation : ces broches servent à alimenter des shields ou circuits externes. Elles permettent également d'alimenter la carte Arduino sans passer par la prise jack présente sur certains modèles. Ces broches sont réunies sous l'étiquette POWER.

- Les broches de communication : il existe trois ports de communication sur les cartes Arduino : le port série, le port I²C et le port SPI. Ces ports ne disposent pas de broches dédiées à leur fonctionnement. Ils partagent en revanche leur fonctionnalité avec d'autres broches. Les broches partagées sont différentes d'une carte à l'autre et c'est pourquoi il est recommandé de consulter la documentation de chaque carte afin de connaitre les numéros des broches correspondant à ces ports.

#### II.1.3.5 Mémoire

Les microcontrôleurs ATmega dont sont équipées la plupart des cartes Arduino disposent de trois types de mémoires : la mémoire flash, la SRAM (Static Random Access Memory ou mémoire vive statique) et le EEPROM (Electrically-Erasable Programmable Read-Only Memory ou mémoire morte effaçable électriquement et reprogrammable).

La mémoire flash et l'EEPROM sont deux mémoires non volatiles dites mortes, c'est-à-dire qu'elles conservent leurs données même en l'absence de courant électrique. C'est d'ailleurs dans la mémoire flash qu'est contenu le programme qui sera exécuté par le microcontrôleur lors de chaque mise sous tension. Cette mémoire flash contient également un bootloader, ou chargeur d'amorçage, sur quelques octets.

Le bootloader est un programme s'exécutant au démarrage de chaque machine, habituellement pour choisir quel système d'exploitation utiliser.

Dans le cas de l'Arduino, le bootloader présent initialement sur chaque carte permet de simplifier le transfert de programmes sur le microcontrôleur. De manière simple, lors de la mise sous tension de la carte, le bootloader vérifie s'il y a une tentative d'écriture d'un nouveau programme. Si c'est le cas, le programme présent dans la mémoire flash sera remplacé, sinon, le programme présent sur la mémoire flash est démarré. Il est possible de se passer de ce bootloader pour gagner un peu d'espace dans la mémoire mais il sera nécessaire dans ce cas de passer par un programmateur spécifique.

L'EEPROM sert quant à elle à stocker des variables que l'on souhaite conserver d'une mise hors tension à l'autre. Elle peut être utile si l'on souhaite stocker des données globales qui sont susceptibles d'être réutilisées tout au long de la durée de vie de son projet.

Ces deux types de mémoires mortes présentent l'avantage de la conservation de données, mais la technologie employée pour ce genre de mémoire réduit significativement leur durée de vie. On estime ainsi que ces deux mémoires ne peuvent supporter que 100000 écritures.

Les zones de la mémoire qui auront dépassé leur espérance de vie deviendront simplement inutilisables. Malheureusement, dans les rares cas de pannes, aucune erreur ne sera relevée et le comportement du programme en cours d'exécution risque d'être aléatoire. La lecture de ces mémoires n'affecte pas leur durée de vie et est donc illimitée. [12]

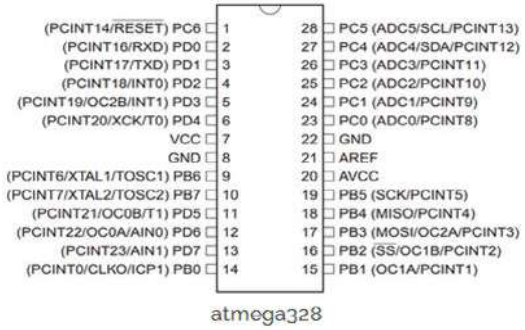



Figure II:1 Schéma explicatif des différentes broches du microcontrôleur ATmega328

### II.1.4 Les gammes de la carte Arduino

Tableau II:1 Différentes gammes de la carte Arduino.[10]

| | | | |
|---|---|---|---|
| Arduino Uno | Arduino Leonardo | Arduino Due | Arduino Yun |
| Arduino Tre | Arduino Micro | Arduino Robot | Arduino Esplora |
| Arduino Mega ADK | Arduino Ethernet | Arduino Mega 2560 | Arduino Mini |
| LilyPad Arduino USB | LilyPad Arduino Simple | LilyPad Arduino SimpleSnap | LilyPad Arduino |
| Arduino Nano | Arduino Pro Mini | | |

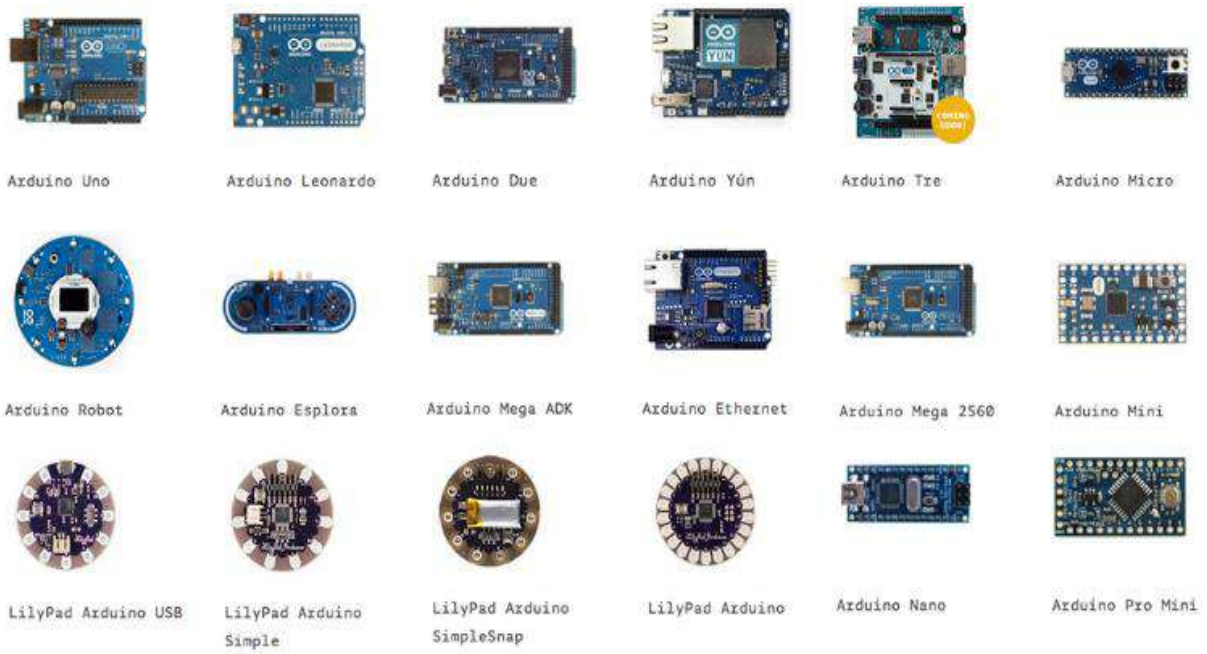



Figure II:2 Différentes Cartes Arduino [26]

### II.1.5 La carte Arduino Uno

#### II.1.5.1 Présentation

La carte Arduino Uno est un produit populaire parmi les cartes Arduino. C'est un excellent endroit pour commencer avec la programmation Arduino et couvre toutes les bases de la construction d'objets de complexité relativement faible.

La carte Arduino Uno, comme son nom l'indique, a été la première à utiliser la version de programmation Arduino 1.0, et elle est devenue le symbole de l'univers Arduino.

La carte Arduino Uno est constituée de 14 broches d'entrées/sorties digitales, dont six sont utilisables en PWM, de 6 broches d'entrées analogiques, d'une connectique USB, d'une connectique d'alimentation, d'un port ICSP et d'un bouton RESET.[11]

La description de toutes les connectiques de la carte Arduino UNO est présentée dans l'image ci-dessous :

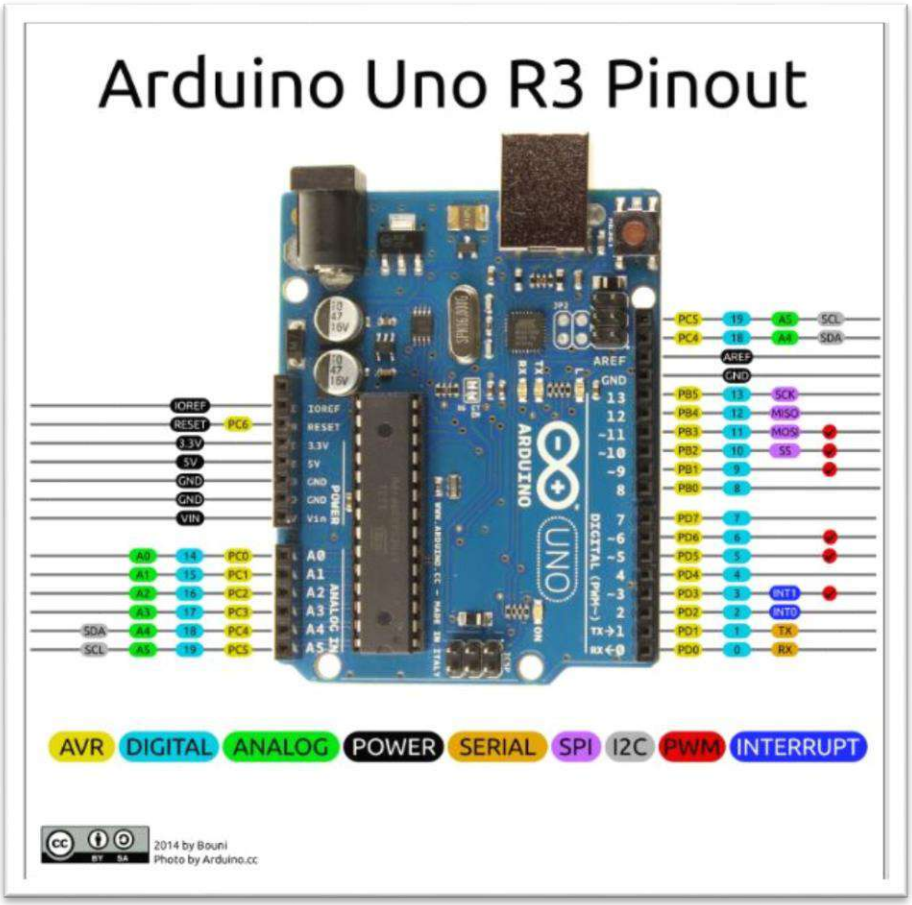



Figure II:3 Description détaillée du brochage de la carte Arduino UNO

L'alimentation de la carte Arduino Uno se fait normalement entre 7V et 12V de courant continu. Cependant, il est possible de faire fonctionner la carte Arduino Uno au maximum entre 6V et 20V. En deca de 6V, la carte n'est plus stable. Au-delà de 20V, le régulateur de tension surchauffe, et peut endommager la carte.

L'alimentation de la carte peut se faire à travers le port USB lorsqu'il est branché sur l'ordinateur, ou via la connectique d'alimentation avec le port jack ou l'entrée d'alimentation. L'alimentation est sélectionnée de manière automatique par la carte Arduino. La source d'alimentation fournissant le meilleur voltage sera sélectionnée comme source d'alimentation par la carte.

En utilisant la connectique d'alimentation, une batterie ou un chargeur spécifique délivrant un courant continu de 9V convient parfaitement pour alimenter d'autres éléments.

L'entrée d'alimentation (VIN) permet d'utiliser une broche afin d'alimenter votre carte Arduino en électricité. Il est conseillé d'utiliser une alimentation entre 7V et 12V de courant continu pour ne pas endommager la carte.

La broche 5V est une connexion de sortie permettant de récupérer un courant généré par le régulateur de la carte. La broche 3.3V permet aussi de récupérer un courant mais de 3.3V et de 50 mA au maximum.

Les prises de terre ou GND permettent de fermer le circuit.

Les broches d'entrées et sorties pour la carte Arduino Uno peuvent être décrites en deux parties.

Les 14 broches d'entrées et sorties digitales sont utilisables comme leur nom l'indique en entrée ou en sortie en utilisant comme leur nom l'indique en entrée ou en sortie en utilisant les fonctions pinMode, Digital Read/ ou Digital Write. Chaque broche opère à 5V et peut fournir ou recevoir au maximum 40mA. De plus, chaque broche dispose d'une résistance interne de 20 à 50 KΩ, non connectée par défaut.

Parmi ces broches, plusieurs possèdent des rôles plus étendus :

- Broches couplées de liaison série : broches 0 (RX) et 1(TX). Ces broches sont particulières puisque leur rôle peut être étendu à une entrée série (RX) et une sorties séries (TX).

- Broches interruptrices : broches 2 et 3. Ces broches peuvent jouer le rôle d'interrupteurs sur différents composants en utilisant la fonction attach Interrupt().

- Broches PWM : broches 3, 5, 6, 9, 10 et 11. Ces broches sont utilisées comme sorties PWM en utilisant la fonction AnalogWrite ().

- Broche SPI : 10(SS), 11(MOSI), 12(MISO) et 13(SCK). Ces broches peuvent être utilisées pour gérer une communication SPI.

- Broche Led-13 : sur la carte Arduino UNO, il existe une LED qui est contrôlable directement à travers la broche 13 en utilisant les valeurs HIGH pour allumer et LOW pour éteindre la LED.

La carte Arduino UNO possède également six broches d'entrées analogiques étiquetées de A0 à A5. Elles mesurent l'entrée de courant sur 5V sur une résolution de 10 bits, soit sur une échelle de 0 à 1023.

Sur les broches A4, appelée aussi broche SDA, et A5, appelée broche SCL, il est possible de gérer la communication I²C.

D'un point de vue plus technique, cette carte se base sur le processeur ATmega328, un puissant microcontrôleur disposant d'une mémoire flash qui donne des performances très élevées tout en ayant une basse consommation.

Il dispose d'une mémoire de 32 Ko, et de 2 Ko de SRAM. Il possède également une mémoire de 2 Ko d'EEPROM, programmable en utilisant la librairie EEPROM. [12]

### II.1.5.2 Les Caractéristiques techniques

Tableau II:2 Caractéristiques de l'Arduino Uno. [13]

| | |
|---|---|
| Microcontrôleur | ATmega328 |
| Tension de fonctionnement | 5v |
| Tension d'Input (recommandée) | 7-12v |
| Tension d'Input (limites) | 6-20v |
| Pins I/O digitales | 14 (dont 6 fournissent une sortie PWM) |
| Pinces d'E/S numériques PWM | 6 |
| Pins Input Analogiques | 6 |
| Courant DC par pin I/O | 20 mA |
| Courant DC pour la broche 3.3V | 50 mA |
| Mémoire flash | 32 Ko (ATmega328P)<br>Dont 0,5 Ko utilisé par bootloader |
| SRAM | 2KB(ATmega328) |
| EEPROM | 1 KB (ATmega328P) |
| Vitesse de l'horloge | 16 MHz |
| LED_BUILTIN | 13 |
| Longueur | 68.6 mm |
| Largeur | 53.4 mm |
| Poids | 25 g |

### II.1.5.3 Avantages

- Pas cher.
- Environnement de programmation clair et simple.

- Multiplateforme : tourne sous Windows, Macintosh et Linux.
- Nombreuses librairies disponibles avec diverses fonctions implémentées.
- Logiciel et matériel open source et extensible.
- Nombreux conseils, tutoriaux et exemples en ligne (forums, site perso etc.…).
- Existence de « Shield » (boucliers en français) : ce sont des cartes supplémentaires qui se connectent sur le module Arduino pour augmenter les possibilités comme par exemple : afficheur graphique couleur, interface Ethernet, GPS, etc...

Par sa simplicité d'utilisation, Arduino est utilisé dans beaucoup d'applications comme l'électronique industrielle et embarquée, le modélisme, la domotique mais aussi dans des domaines différents comme l'art contemporain ou le spectacle.

On constate sur la toile, que par le biais d'Arduino, de nombreux curieux redécouvrent l'électronique. [11]

## II.2 Circuit de mesure : les capteurs de gaz

### II.2.1 Présentation générale

Un détecteur de gaz est un dispositif conçu pour alerter l'utilisateur lorsque l'atmosphère devient explosive (avec gaz inflammable) ou toxique (avec gaz dangereux ou manque d'oxygène). Les détecteurs de gaz combustibles sont également appelés « explosimètre » en français.

Pour définir un seuil de danger avec précision, l'appareil doit réaliser une mesure : il mesure la concentration du gaz présent, ou la plupart du temps sa pression partielle. Ce type d'appareil est donc utilisé en surveillance de l'atmosphère, et non en contrôle de processus, ce qui le différencie des analyseurs qui sont plus précis et donc plus coûteux. Son usage est le plus souvent de nature industrielle ; des tentatives existent cependant pour le faire entrer dans les usages domestiques (détecteur de fuites de méthane).

L'utilisation des microprocesseurs permet d'élargir son domaine à l'hygiène industrielle et même au contrôle de pollution ; il devient possible de mémoriser les mesures, de calculer des moyennes dans le temps... Cependant ces mesures, bien souvent réglementaires, exigent des précisions ou des règles spécifiques de mesurage qui ne peuvent pas toujours être obtenues ou

utilisées avec ce type d'appareil. Méthodes et spécifications sont en cours d'élaboration au niveau des Communautés européenne et internationale (travaux du CEN – Comité européen de normalisation – et de l'ISO – Organisation internationale de normalisation).

Il convient donc d'être prudent dans le choix de tels appareils pour des mesures autres que de simples indications de présence ou d'absence de gaz. [14]

### II.2.2 Principaux paramètres d'un capteur de gaz

Parmi les grandeurs caractéristiques qui décrivent les performances d'un capteur, la sensibilité, la sélectivité et le temps de réponse sont des paramètres essentiels.

#### II.2.2.1 La sensibilité

Correspond à la variation de la grandeur mesurée, engendrée par une variation de la concentration en gaz. Elle se définit par la relation suivante :

$$sesibilité(X)_p = \left|\frac{\Delta X}{\Delta Y}\right|_P$$

Avec X la grandeur mesurée (par exemple, R pour une résistance, U pour une tension, I pour une intensité électrique ou ϕ pour une intensité lumineuse).

C'est la concentration du gaz, pour un jeu de paramètres p (température, pression, humidité).

#### II.2.2.2 La réponse relative

C'est la grandeur la plus souvent employée dans la littérature, qui correspond au rapport entre les grandeurs mesurées dans l'air ($R_{air}$, $U_{air}$) et en présence du gaz ($R_{gaz}$, $U_{gaz}$). Elle s'exprime, dans le cas d'un gaz oxydant, par :

$$Réponse_{relative} = \frac{R_{gaz}}{R_{air}} \; ou \; Réponse_{Prelative} = \frac{R_{gaz} - R_{air}}{R_{gaz}}$$

Et dans le cas d'un gaz réducteur, par :

$$Réponse_{relative} = \frac{R_{air}}{R_{gaz}} \; ou \; Réponse_{Prelative} = \frac{R_{air} - R_{gaz}}{R_{air}}$$

Un capteur est d'autant plus sensible qu'une faible variation de concentration du gaz entraînera une variation importante de la résistance mesurée. La sensibilité (ou la réponse relative) est un des points forts des capteurs à base d'oxydes métalliques dont la variation de la résistance peut être mesurée pour des concentrations de l'ordre de ppm (partie par million) voir de la centaine ppb (partie par milliard).

### II.2.2.3 La sélectivité

D'un capteur est également un paramètre déterminant. Cette sélectivité va permettre de discriminer un gaz cible au sein d'un mélange de gaz. Elle se définit comme le rapport des sensibilités (ou des réponses relatives) entre deux gaz pour une même concentration en gaz.

$$Sélectivité_{gaz\,1/gaz\,2} = \frac{Réponse_{relative(gaz\,1)}}{Réponse_{relative(gaz\,2)}}$$

Contrairement à la sensibilité, la sélectivité des capteurs conçus à base d'oxydes métalliques apparaît comme leur point faible. En effet, ceux-ci apparaissent peu sélectifs vis-à-vis d'un gaz en particulier à moins de recourir à des traitements de dopage du semi-conducteur.

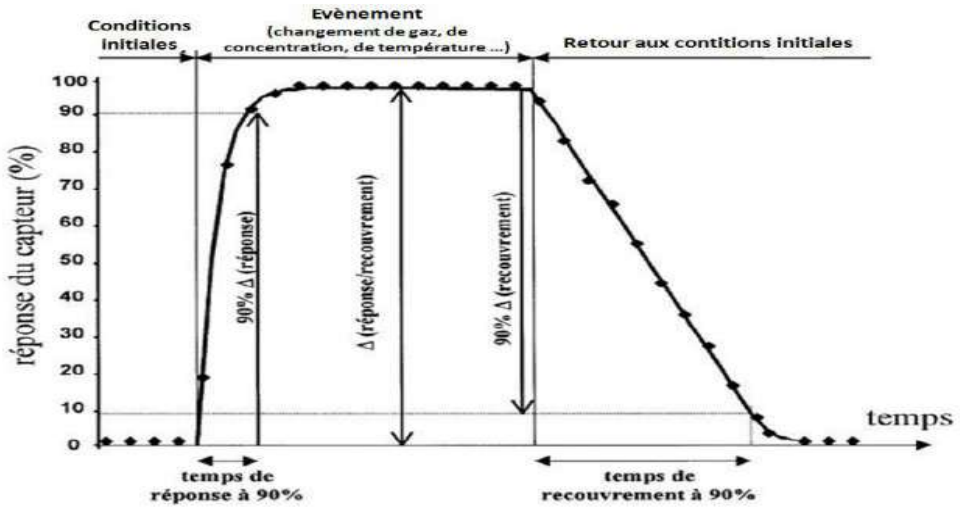



Figure II:4 Schéma représentatif des temps de réponse et de recouvrement d'un capteur de gaz à base de semi-conducteur

### II.2.2.4 Temps de réponse

Qui correspond au temps mis par le capteur pour réagir suite à la mise en présence d'un gaz.

#### II.2.2.5 Temps de recouvrement

En pratique ces temps caractéristiques sont pondérés d'un facteur 0,9 puisqu'au-delà de 90%, le temps de réaction n'évolue plus linéairement, ceci est valable aussi bien pour la mise en présence que pour le retrait du gaz. [15]

### II.2.3 Les différents types des capteurs MQ

### II.2.3.1 Capteur MQ2

#### II.2.3.1.1Description

Capteur de gaz MQ2 est un utile pour la détection des fuites de gaz (dans la maison et l'industrie). Il peut détecter le H2, le GPL, le CH4, le CO, l'alcool, la fumée, le propane. Basé sur son temps de réponse rapide.

Les mesures peuvent être prises dès que possible. De plus, la sensibilité peut être réglée par le potentiomètre. [20]

#### II.2.3.1.2Caractéristiques

a. **Condition de travail standard**

Tableau II:3 Condition de travail standard. [16]

| Symbole | Nom du paramètre | Condition technique | Remarques |
|---|---|---|---|
| $V_C$ | Tension de circuit | 5V± 0.1 | AC OU DC |
| $V_H$ | Tension de chauffage | 5V± 0.1 | AC OU DC |
| $R_L$ | Résistance de charge | Peut ajuster | |
| $R_H$ | Résistance de chauffage | 33Ω ± 5% | Température ambiante |
| $P_H$ | Consommation de chauffage | Moins de 800mW | |

b. **Conditions environnementales**

Tableau II:4 Condition environnementales.[16]

| Symbole | Nom du paramètre | Condition technique | Remarques |
|---|---|---|---|
| Tao | Utilisation de Tem | -20℃-50℃ | |
| Tas | Stockage Tem | -20℃-70℃ | |

| | | | |
|---|---|---|---|
| $R_H$ | Humidité relative | Moins de 95% de Rh | |
| $O_2$ | Concentration d'oxygène | 21% (condition standard) Oxygène Concentration peut affecter la sensibilité | Valeur minimale est Plus de 2% |

c. **Caractéristique de sensibilité**

Tableau II:5 Caractéristique de sensibilité.[16]

| Symbole | Nom du paramètre | Condition technique | Remarques |
|---|---|---|---|
| Rs | Résistance à la détection | | Détection de la concentration portée : 200ppm-5000ppm GPL et propane 300ppm-5000ppm butane 5000ppm-20000ppm méthane 300 ppm-5000 ppm H2 100 ppm-2000 ppm de l'alcool |
| α(3000ppm/1000ppm CH4) | Concentration Taux de pente | | |
| Condition de détection standard | Temp : 20 ℃± 2 ℃ Vc : 5V ± 0,1 Humidité : 65% ± 5% Vh: 5V ± 0,1 | | |
| Temps de préchauffage | Plus de 24 heures | | |

d. **Structure et configuration, circuit de mesure de base**

Tableau II:6 Structure et configuration, circuit de mesure de base. [16]

| | Pièces | Matériaux |
|---|---|---|
| 1 | Capteur de gaz | $SnO_2$ |
| 2 | Électrode | Au |
| 3 | Ligne d'électrode | Pt |
| 4 | Bobine de chauffage | Alliage Ni-Cr |
| 5 | Céramique tubulaire | $Al_2O_3$ |
| 6 | Réseau anti-explosion | Gaz d'acier inoxydable (SUS316 100 mesh) |
| 7 | Bague de serrage | Cuivre plaqué Ni |
| 8 | Base de résine | Bakélite |
| 9 | Tube Pin Cuivre plaqué | Cuivre plaqué Ni |

e. **Courbe caractéristique de sensibilité.**

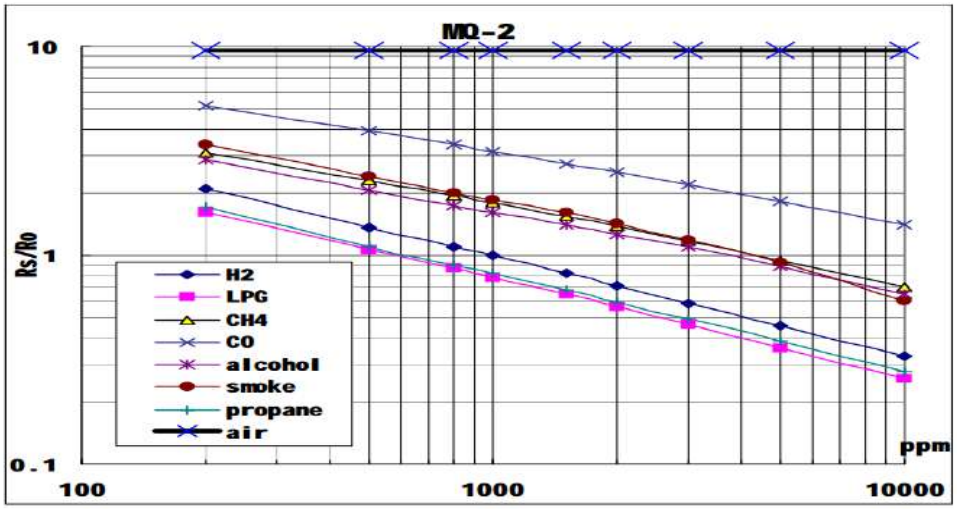



Figure II:5 Caractéristiques de sensibilité typiques du MQ-2 pour plusieurs gaz. [16]

### II.2.3.2 Capteur MQ5

#### II.2.3.2.1Description

Le capteur de gaz MQ-5 a une grande sensibilité au méthane, au propane et au butane, et pourrait être utilisé pour détecter le méthane et le propane. Le capteur pourrait être utilisé pour détecter différents gaz combustibles, en particulier le méthane, il est à faible coût et convient à différentes applications.[16]

#### II.2.3.2.2Caractéristiques

a. **Condition de travail**

Tableau II:7 Caractéristiques MQ5. [16]

| | | | |
|---|---|---|---|
| Numéro de modèle | | | MQ-5 |
| Type de capteur | | | Semi-conducteur |
| Encapsulation standard | | | Bakélite (Bakélite noire) |
| Détection de gaz | | | LPG, méthane, gaz de houille |
| Concentration | | | 300-10000 ppm (méthane, propane, butane, $H_2$) |
| Circuit | Tension de boucle | $V_c$ | ≤24V DC |
| | Tension de chauffage | $V_H$ | 5.0V±0.2V AC ou DC |
| | Résistance de charge | $R_L$ | Ajustable |
| Caractère | Résistance de chauffage | $R_H$ | 31Ω±3Ω (Room Tem.) |
| | Consommation de chauffage | $P_H$ | ≤ 900mW |
| | Résistance à la détection | $R_S$ | 2KΩ-20KΩ (en 2000 ppm $C_3H_8$) |
| | Sensibilité | S | $R_S$ (dans l'air) / $R_S$ (1000 ppm C3H8) ≥5 |

| | | | |
|---|---|---|---|
| | Pente | A | ≤0,6(R1000ppm/R500ppm $H_2$) |
| Condition | Tem. Humidité | | 20C°±2C° : 65%±5%$R_H$ |
| | Circuit de test standard | | $V_C$ : 5.0V±0.1V- $V_H$ : 5.0V±0.1V |
| | Temps de préchauffage | | Plus de 48 heures |

b. **Courbe caractéristique de sensibilité**

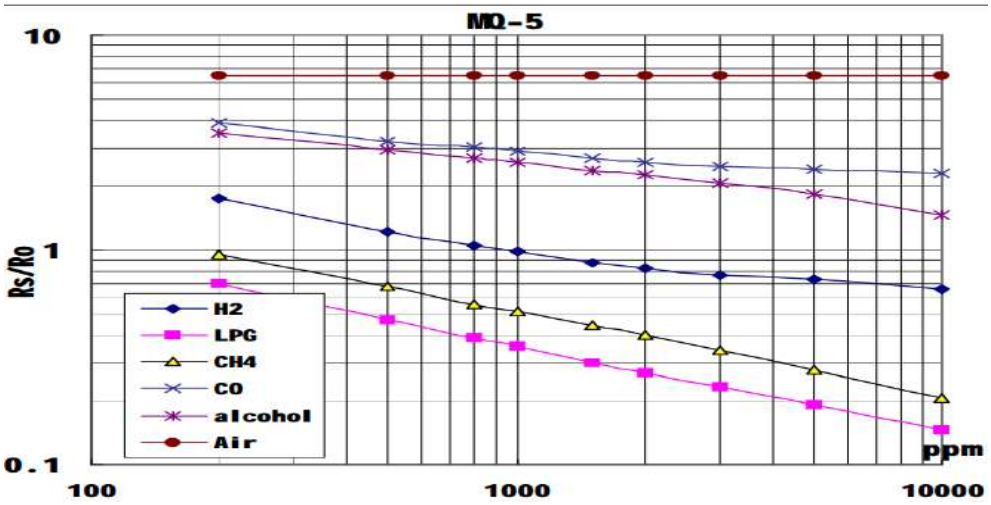



Figure II:6 Caractéristiques de sensibilité typiques du MQ-5 pour plusieurs gaz. [16]

### II.2.3.3 Capteur de gaz MQ7

#### II.2.3.3.1Description

Le capteur de gaz monoxyde de carbone MQ-7 est un capteur de gaz semi-conducteur qui détecte la présence de gaz monoxyde de carbone à des concentrations de 10 ppm à 10 000 ppm. L'interface de tension analogique simple du capteur nécessite une seule broche d'entrée analogique à partir de votre microcontrôleur.

Le capteur de gaz monoxyde de carbone MQ-7 détecte la concentration de gaz monoxyde de carbone dans l'air et évite sa lecture en tant que tension analogique. Le capteur peut fonctionner à des températures comprises entre -10 et 50 ° C et consomme moins de 150 mA à 5 V. [18]

#### II.2.3.3.2Caractéristiques

a. **Condition de travail standard**

Tableau II:8 Condition de travail standard. [19]

| Symbole | Nom du paramètre | Condition technique | Remarques |
|---|---|---|---|
| $V_C$ | Tension de circuit | 5V±0.1 | AC ou DC |
| $V_H$ (H) | Tension de chauffage (élevée) | 5V±0.1 | AC ou DC |
| $V_H$ (L) | Tension de chauffage (basse) | 1.4V±0.1 | AC ou DC |
| $R_L$ | Résistance de charge | Peut ajuster | |
| $R_H$ | Résistance au chauffage | 33 Ω ±5% | Température ambiante |
| $T_H$ (H) | Temps de chauffage (élevé) | 60 ± 1 secondes | |
| $T_H$ (L) | Temps de chauffage (basse) | 90 ± 1 secondes | |
| $P_H$ | Consommation de chauffage | Environ 350mW | |

b. **Conditions environnementales**

Tableau II:9 Conditions environnementales. [19]

| Symbole | Nom du paramètre | Condition technique | Remarques |
|---|---|---|---|
| Tao | Utilisation de Tem | -20℃-50℃ | |
| Tas | Stockage Tem | -20℃-70℃ | Conseils en utilisant la portée |
| RH | Humidité relative | Moins de 95% de Rh | |
| $O_2$ | Concentration d'oxygène | 21% (condition standard) Oxygène Concentration peut affecter la sensibilité | Valeur minimale est plus de 2% |

c. **Caractéristique de sensibilité**

Tableau II:10 Caractéristique de sensibilité. [19]

| Symbole | Nom du paramètre | Condition technique | Remarques |
|---|---|---|---|
| Rs | Résistance superficielle du corps sensible | 2KΩ-20KΩ | En 100 ppm de monoxyde de carbone |
| α(3000ppm/1000ppm) | Concentration Taux de pente | Moins de 0.5 | $R_S$(300ppm)/Rs(100ppm) |
| Condition de détection standard | Température -20 ℃ ± 2 ℃ humidité relative 65% ± 5% RL : 10K Ω ± 5% | | |
| | $V_C$ :5V±0.1V $V_H$ :5V±0.1V $V_H$ :1.4V±0.1V | | |
| Temps de préchauffage | Pas moins de 48 heures | | Plage de détection : 20 ppm 2000 ppm de monoxyde de carbone |

### d. **Courbe caractéristique de sensibilité**

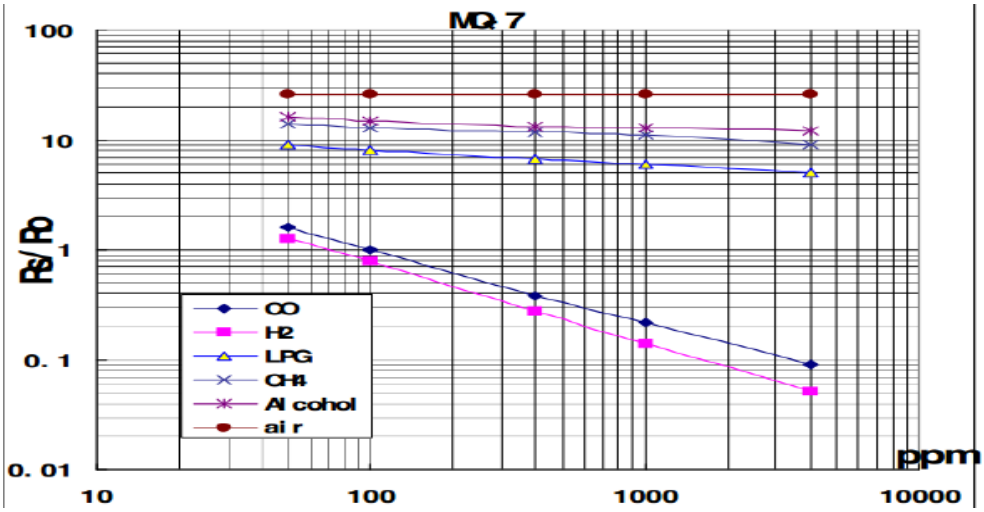



Figure II:7 Caractéristiques de sensibilité typiques du MQ-7 pour plusieurs gaz. [19]



Figure II:8 Capteur de gaz

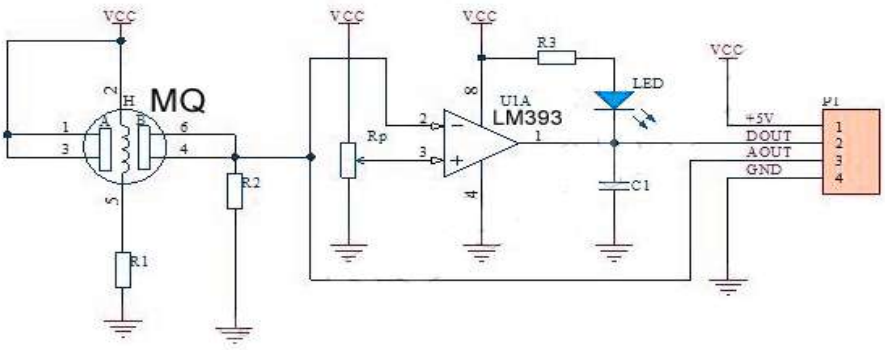



Figure II:9 Schéma électrique des capteurs MQx utilisés

## II.3 Conclusion

Ce chapitre a montré les principes de fonctionnement et les caractéristiques de la carte Arduino et a décrit deux parties importantes de l'Arduino : le matériel et la partie programmation. Le circuit de mesure a également montré les caractéristiques de divers capteurs qui permettent une plus large gamme de détection de gaz.

# Chapitre III. **Réalisation de l'application de détection des gaz**

## III.1 Introduction

Ce chapitre présente les étapes et les composants nécessaires à la réalisation du projet, tels que « Système embarqué pour la détection des gaz dangereux ». Dans le chapitre précédent, nous avons décrit les détails techniques du circuit de contrôle, qui est une carte microcontrôleur de type Arduino UNO, et le circuit de mesure basé sur différents capteurs de type MQx.

La section suivante présente les autorisations requises pour afficher, empêcher, notifier et générer des alertes.

## III.2 Partie Hardware

### III.2.1 Afficheur LCD (Liquide Crystal Display)

L'afficheur a été utilisé pour affichage de la concentration du gaz, cette information est surtout nécessaire pour une personne qualifiée, qui peut vérifier les mesures issues des différents capteurs sans avoir besoin de brancher un PC à la carte Arduino et utiliser un logiciel spécialisé.

Les écrans LCD ou écrans à cristaux liquides sont de plus en plus courants dans notre environnement, que ce soit pour afficher des informations utiles ou pour servir de sélecteur de commande. Il existe des écrans de tout type allant du simple afficheur de caractères aux écrans géants couleur.

Les écrans les plus couramment utilisés dans des applications Arduino sont les écrans à caractères alphanumérique. [12]

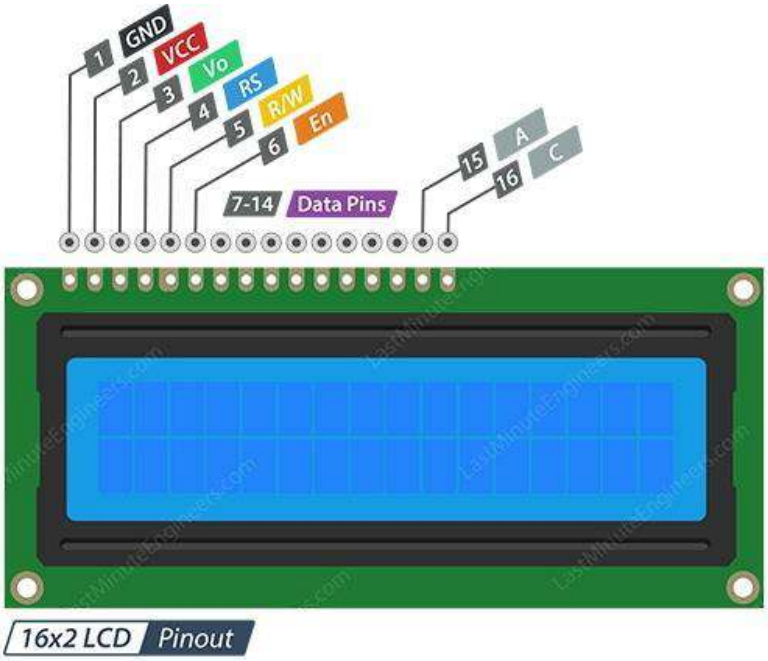



Figure III:1 Brochage de l'afficheur LCD 16x2

Tableau III:1 Branchement de l'afficheur LCD

| N° | Nom | Rôle |
|---|---|---|
| 1 | VSS | Masse |
| 2 | VDD | +5V |
| 3 | VO | Réglage du contraste |
| 4 | RS | Sélection du registre (commande ou donnée) |
| 5 | R/W | Lecture ou écriture |
| 6 | E | Entrée de validation |
| 7 à 14 | D0 à D7 | Bits de données |
| 15 | A | Anode rétroéclairage (+5V) |
| 16 | K | Cathode rétroéclairage (masse) |

Généralement, pour tous les écrans LCD (non graphiques) ce brochage est le même.

Les broches utiles qu'il faudra relier à l'Arduino sont les broches 4, 5 (facultatives), 6 et les données (7 à 14 pouvant être réduite à 8 à 14) en oubliant pas l'alimentation et la broche de réglage du contraste. Pour envoyer des données sur l'écran, il suffit de :

1. Placer la broche RS à 1 ou 0 selon que l'on veut envoyer une commande.

2. Place sur les 8 broches de données (D0 à D7) la valeur de la donnée à afficher.

3. Faire une impulsion sur E d'au moins 450 ns pour indiquer à l'écran que les données sont prêtes.

La communication parallèle prend beaucoup de broches, il existe un mode "semi-parallèle". Ce dernier se contente de travailler avec seulement les broches de données D4 à D7 (en plus de RS et E) et il faudra mettre les quatre autres (D0 à D3) à la masse. Il libère donc quatre broches. Dans ce mode, on fera donc deux fois le cycle "envoi des données puis impulsion sur E" pour envoyer un octet complet. On utilisera une libraire nommée Liquide Crystal qui se chargera de gérer les timings et l'ensemble du protocole.

L'afficheur LCD utilise 6 à 10 broches de données ((D0 à D7) ou (D4 à D7) + RS + E) et deux d'alimentations (+5V et masse). La plupart des écrans possèdent aussi une entrée analogique pour régler le contraste des caractères. On branchera dessus un potentiomètre de 10 kΩ. Les 10 broches de données peuvent être placées sur n'importe quelles entrées/sorties numériques de l'Arduino. [21]

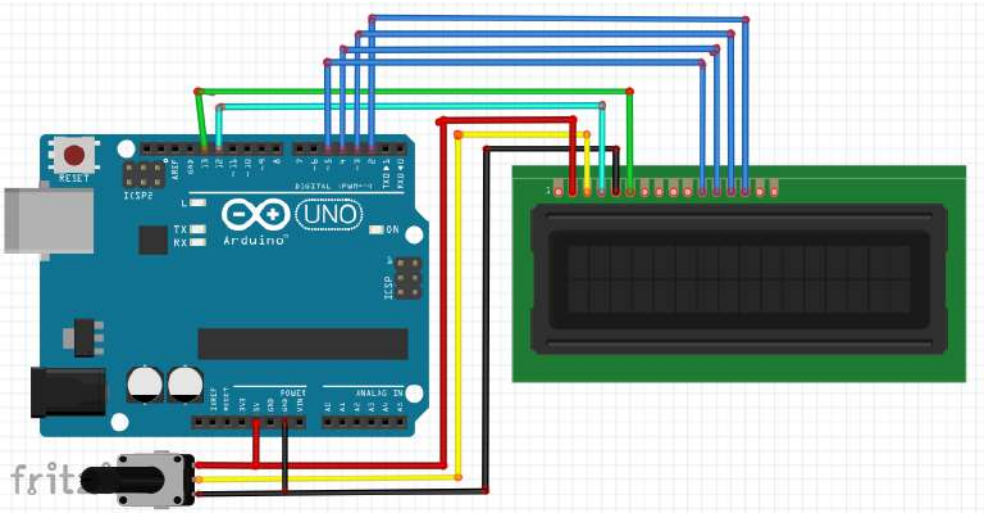



Figure III:2 Schéma de câblage d'un LCD 16x2 avec Arduino

❖ Module I2C pour LCD

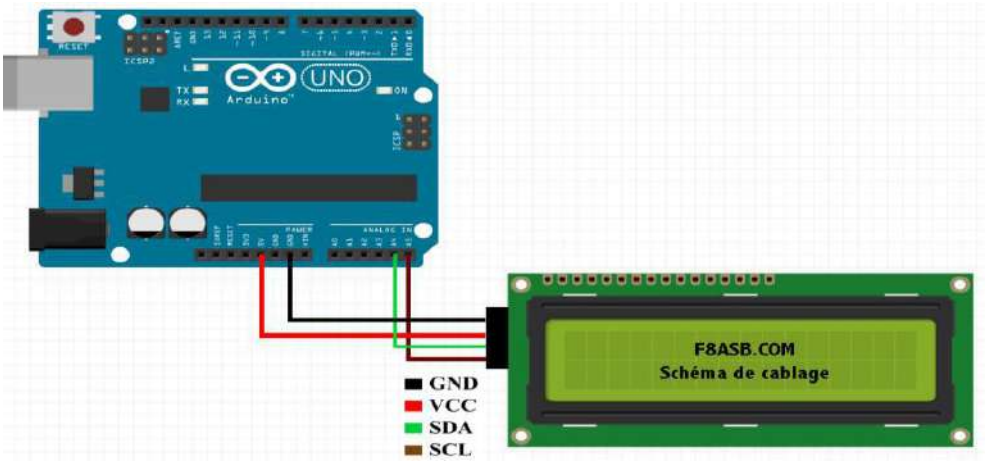



Figure III:3 Schéma de câblage d'un I²C avec Arduino

Les normes I2C (Inter-Integrated Circuit) et SPI (Serial Peripheral Interface) ont été créées pour fournir un moyen simple de transférer des informations numériques entre des capteurs et des microcontrôleurs.

Les bibliothèques Arduino pour I2C et SPI facilitent l'utilisation de ces deux protocoles. Le choix entre I2C et SPI est en général déterminé par les périphériques que l'on souhaite connecter. Certains appareils offrent les deux standards, mais habituellement un périphérique ou une puce ne supporte qu'une seule des deux normes.

Les deux connexions du bus I2C se nomment SCL (Serial Clock Line) et SDA (Serial Data line). Elles sont disponibles sur une carte standard Arduino en employant la broche analogique 5 pour SCL qui fournit un signal d'horloge, et la broche analogique 4 pour SDA, qui s'occupe du transfert des données. [25]

### III.2.2 LED

Les LEDs ont été utilisées dans le seul but d'une alerte visuelle.

Une LED verte s'allume lorsque les gaz dangereux ne sont pas détectés dans l'environnement. Dans le cas où ces gaz sont détectés, la LED verte s'éteint et une LED rouge s'allume pour signaler la situation dangereuse.

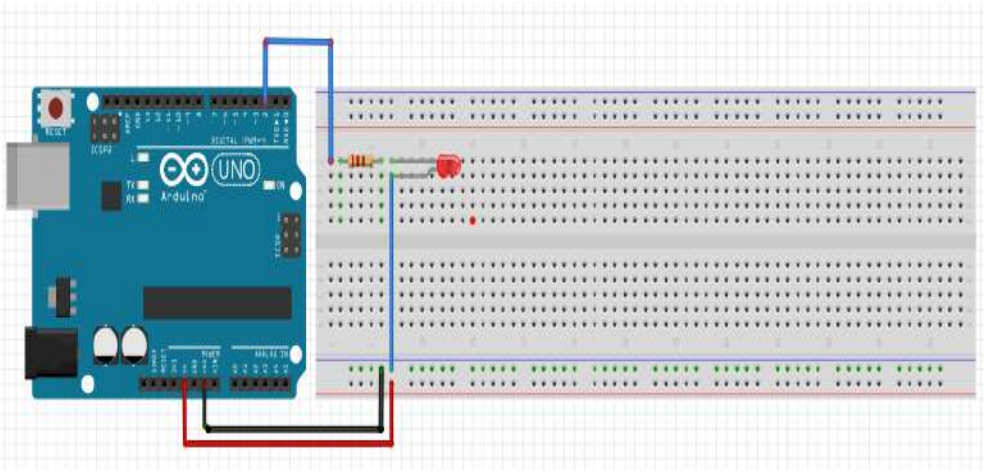


Figure III:4 Schéma de câblage d'un LED avec Arduino

### III.2.3 Buzzer

Ici pour simuler l'effet d'une sirène, un buzzer est utilisé à sa place pour générer une alarme sonore.

Un buzzer ou un bipper est un dispositif de signalisation audio, qui peut être mécanique, électromécanique ou piézoélectrique. Les utilisations typiques des buzzers et des bippers comprennent les dispositifs d'alarme, les horloges et la confirmation de l'entrée de l'utilisateur, comme un clic de souris ou une frappe de touche.

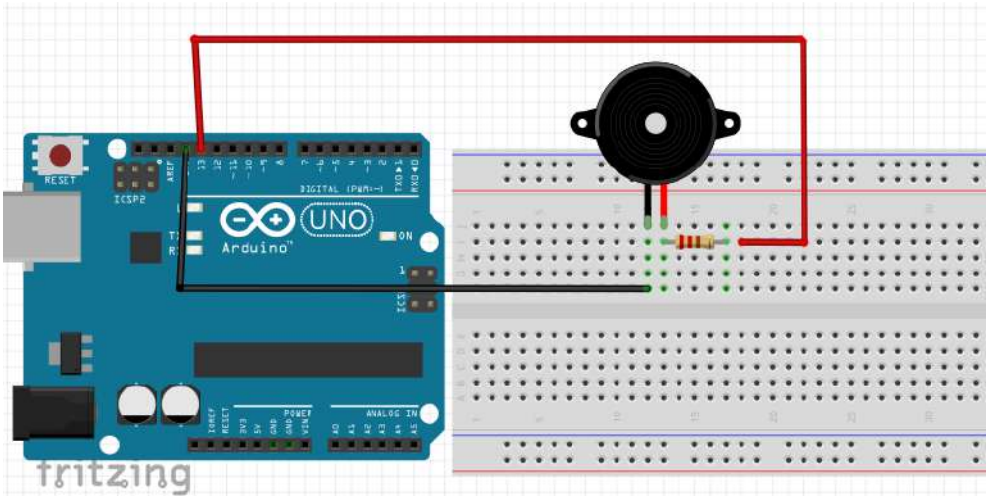



Figure III:5 Schéma de câblage buzzer avec Arduino

Le buzzer est une structure intégrée de transducteurs électroniques, d'alimentation en courant continu, largement utilisée dans les ordinateurs, les imprimantes, les copieurs, les alarmes, les

jouets électroniques, les équipements électroniques automobiles, les téléphones, les minuteries et autres produits électroniques pour appareils sonores. Buzzer actif 5V

L'alimentation nominale peut être directement connectée à un son continu, ce module dédié à l'extension de capteur et à la carte en combinaison peut compléter une conception de circuit simple [22].

### III.2.4 Moteur pas à pas

Dans le but de réduire la fuite du gaz dangereux et de prévenir l'augmentation de sa concentration ; que ce soit issue d'une conduite d'une installation domestique de méthane ou d'un gaz brulé générant le CO ; un moteur pas à pas représentant une électrovanne proportionnelle principale d'alimentation du gaz est utilisé.

Les moteurs pas à pas sont un type de moteur assez particulier. Comme leur nom l'indique, ils ne peuvent tourner qu'en effectuant un petit pas autour de leur axe de rotation. C'est pourquoi ils portent ce nom de moteur pas à pas.

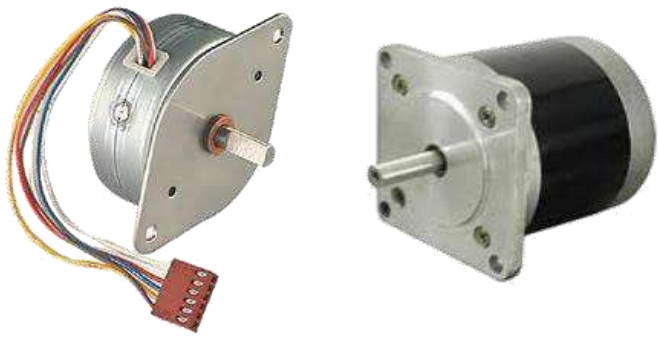


Figure III:6 Moteur pas à pas

Pour faire simple, l'axe de rotation du moteur pas à pas contient un aimant. Il est entouré d'une série de bobines dont les champs magnétiques peuvent être modifiés selon l'utilisation. Cette modification des champs entraine alors l'aimant qui à son tour entraine l'axe de rotation. Cet avancement peut être contrôlé avec une grande précision, permettant de l'axe dans l'espace comme il peut aussi permettre de maintenir la position du moteur.

On peut retrouver ces moteurs dans un des objets quotidiens très utilisés de nos jours, le scanner informatique. La position du moteur et l'acquisition de l'image permettent alors aux drivers de scanner de reconstituer des images fidèles.

Comme pour le servo moteur, les moteurs pas à pas disposent d'une bibliothèque prête à l'emploi, la bibliothèque Stepper, dans l'environnement de développement Arduino. [12]

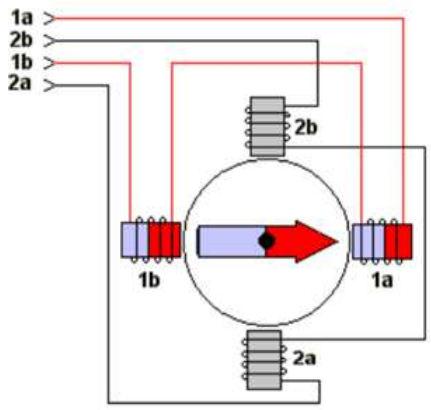



Figure III:7 Modèle conceptuel du moteur pas à pas

### III.2.5 Driver en pont H L298N

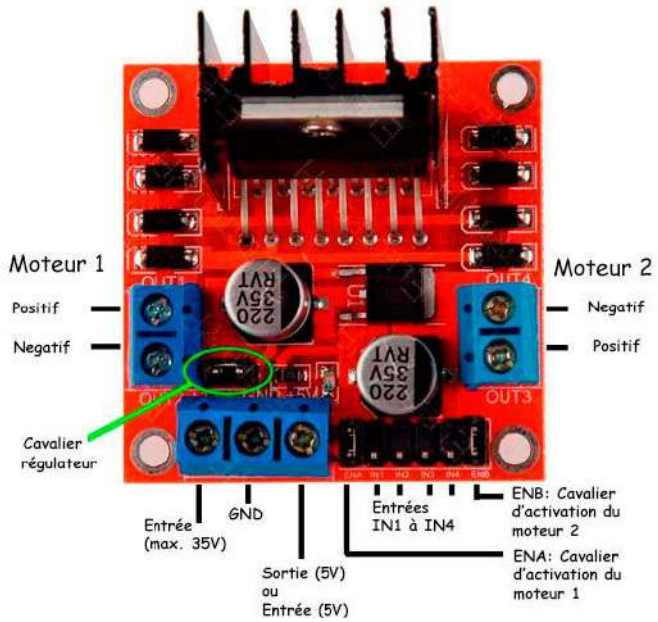



Figure III:8 DriverL298N

Ce circuit est un Double Pont-H destiné au contrôle de moteur continu. Il est basé sur le composant L298N qui est un double Pont-H conçu spécifiquement pour ce cas d'utilisation. C'est un module extrêmement utile pour le contrôler de robots et ensembles mécanisés. Il peut contrôler deux moteur courant continu ou un moteur pas-à-pas 4 fils 2 phases. Il est conçu pour

supporter des tensions plus élevées, des courants importants tout en proposant une commande logique TTL (basse tension, courant faibles, idéal donc pour un microcontrôleur).

Il peut piloter des charges inductives comme des relais, solénoïdes, moteurs continus et moteurs pas à-pas. Les deux types de moteurs peuvent être contrôlés aussi bien en vitesse (PWM) qu'en direction. Toutes les sorties en puissance sont déjà protégées par des diodes anti-retours. Il s'agit d'un module prêt à l'emploi. [24]

### III.2.5.1 Détails techniques

Tableau III:2 Fiche technique du driver L298N. [24]

| Nom de connecteur | Direction | Description | Utilisation |
|---|---|---|---|
| VMS GND | | Connecté à l'alimentation externe (6V~35V) | ENA(ENB) raccordés à un niveau haut (HIGH) activera MOTORA(MOTORB).<br>IN1(IN3) raccordés à 5V et IN2(IN4) a GND<br>MOTORA(MOTORB) tournera dans le sens horlogique.<br>IN1(IN3) raccordés à GND et IN2(IN4) à 5V MOTORA(MOTORB) tournera dans le sens Anti-horlogique.<br>Si vous voulez contrôler la vitesse, vous pouvez connecter ENA(ENB) sur une sortie PWM. |
| ENA | Entrée | Compatible TTL, peut donc être utilise directement avec Arduino.<br>Entrée "Enable" : un niveau bas LOW désactive le Pont A | |
| IN1 | Entrée | Compatible TTL. Entrée de commande de sens du pont A. | |
| IN2 | Entrée | Compatible TTL. Entrée de commande de sens du pont A. | |
| ENB | Entrée | Compatible TTL. Entrée "Enable" : un niveau bas LOW désactive le Pont B | |
| IN3 | Entrée | Compatible TTL. Entrée de commande de sens du pont B | |
| IN4 | Entrée | Compatible TTL. Entrée de commande de sens du pont B | |
| MOTORA | Sortie | Sortie du pont A. Commande en puissance du Moteur A | |
| MOTORB | Sortie | Sortie du pont A. Commande en puissance du Moteur A | |

### III.2.5.2 Activation moteur

- ENA raccordés à un niveau haut (HIGH) activera MOTORA.
- ENB raccordés à un niveau haut (HIGH) activera MOTORB. Si vous voulez contrôler la vitesse, vous pouvez connecter ENA(ENB) sur une sortie PWM.
- Dans le cas du moteur pas à pas ENA et ENB sont raccordés à un niveau haut (HIGH).

### III.2.5.3 Rotation Moteur DC A/B

- IN1/IN3 raccordés à 5V et IN2/IN4a GND MOTORA/B tournera dans le sens horaire.
- IN1/IN3 raccordés à GND et IN2/IN4 à 5V MOTORA/B tournera dans le sens Anti-horaire.

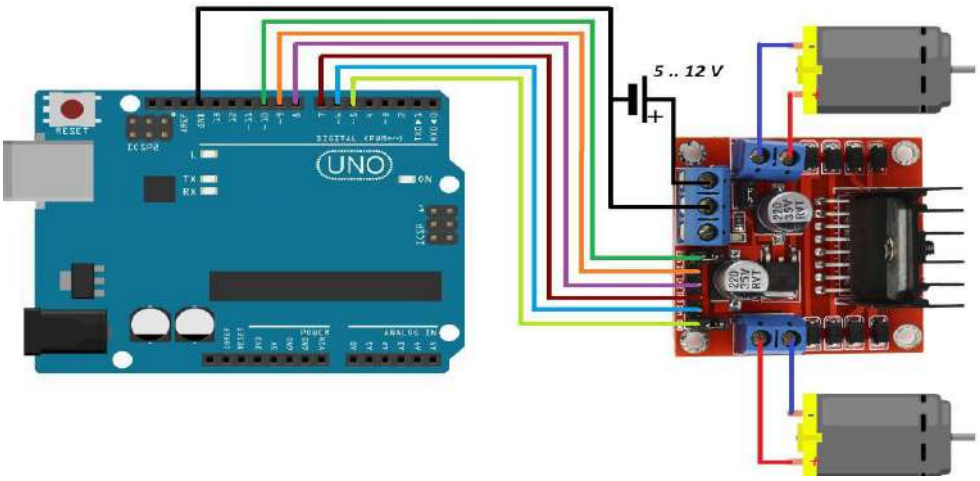



Figure III:9 Brochage de moteur à courant continue et driver avec Arduino

### III.2.5.4 Rotation Moteur pas-à-pas

- On envoie la séquence logique (0110)-(0101)-(1001)-(1010) à IN1, IN2, IN3, IN4 ; avec des temporisations entre chaque séquence pour que le moteur tournera dans le sens horaire.
- Pour inverser le sens, on inverse l'ordre de la séquence envoyée.

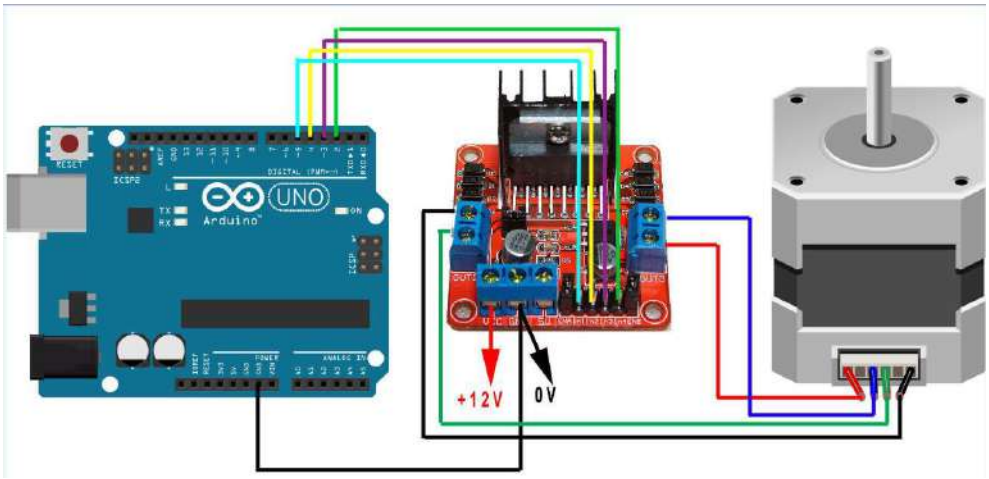



Figure III:10 Brochage de moteur pas à pas et driver avec Arduino

### III.2.6 Ventilateur

Pour diminuer l'effet de la concentration d'un gaz dangereux suite à une fuite ; un ventilateur de type DC est utilisé pour aérer la zone contaminée en évacuant le gaz à l'extérieur.



Figure III:11 Ventilateur

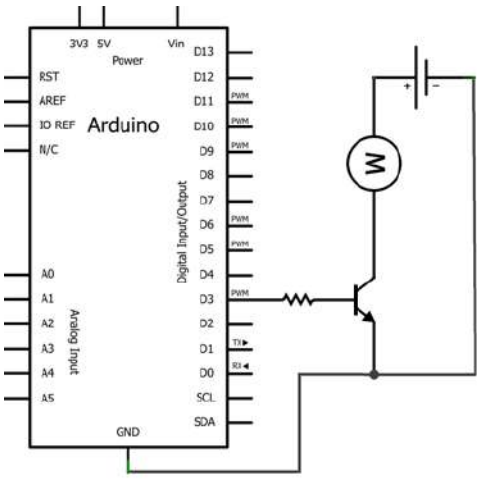



Figure III:12 Schéma de branchement du ventilateur avec Arduino UNO

### III.2.7 Module GSM SIM 800

Pour la notification des personnes distantes (typiquement 05 personnes), qui peuvent être en dehors de la scène où la fuite du gaz dangereux s'est produite ; un kit GSM de type SIM800 est utilisé pour informer ces personnes par des messages SMS d'alerte.

SIM800 est une solution Quad-band GSM / GPRS complète dans un type SMT qui peut être intégré dans les applications client.

SIM800 supportant Quad-band 850/900/1800 / 1900MHz, il peut transmettre des informations de voix, de SMS et de données avec une faible consommation d'énergie. Avec une taille mince de 24*24*3 mm, il peut s'adapter à des exigences minces et compactes de la conception du client. Avec Bluetooth et Embedded AT, il permet des économies de coûts totales et un délai de mise sur le marché rapide pour les applications clients.[23]

#### III.2.7.1 Principales caractéristiques

- Interface de commande AT
- Quad-band
- Effectuer et recevoir des appels vocaux
- Envoyer et recevoir des messages SMS
- Envoyer et recevoir des données GPRS (TCP / IP, HTTP, etc.)
- Bluetooth : compatible avec 3.0 + EDR
- Connecteur USB pour la mise à jour du microprogramme

- Vitesse de transmission configurable (9600-115200, valeur par défaut d'usine : 9600)
- Connecteurs pour haut-parleur externe et microphone
- Interface sélectionnable entre le port série matériel et le port série du logiciel
- Pile de protocoles TCP / IP puissante intégrée pour le transfert de données Internet sur GPRS
- Circuits de décalage de niveau pour le rendre Arduino-Safe
- Connecteur SMA avec antenne externe
- Voyants lumineux pour alimentation et connectivité
- Socket SMA Standard
- Protection ESD sur TVS Zenerarray
- Séparer les interrupteurs de réinitialisation pour Arduino et l'écran
- Glissez le commutateur pour échanger le blindage entre Arduino et PC
- Fourniture pour sélectionner entre les ports série matériels et logiciels. [23]



Figure III:13 GSM-GPRS sim800

### III.2.8 Application globale

Le système proposé se compose de trois parties principales, nommément un circuit de contrôle basé sur Arduino UNO, un circuit de mesure et de détection de gaz et un circuit des sorties englobant : un module de prévention et d'activation alarme et un module d'affichage. Le schéma fonctionnel du système proposé est illustré à la figure (III.14).

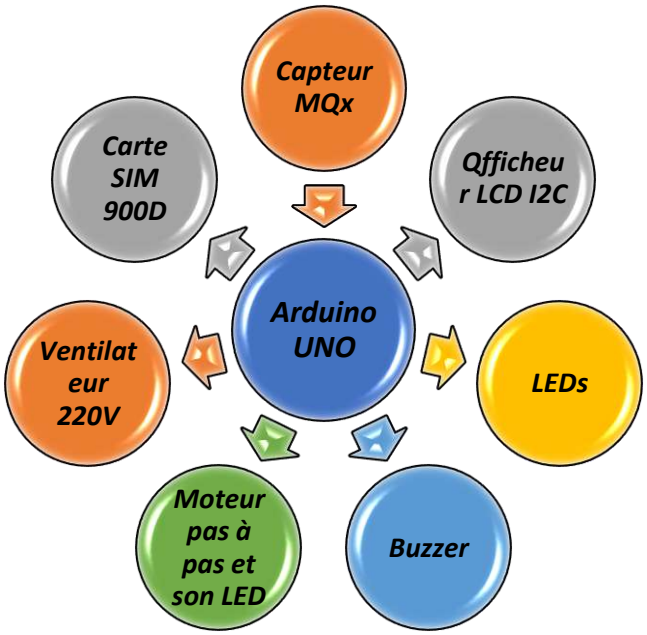



Figure III:14 Schéma synoptique du montage principal

Parmi les phases les plus importantes dans le développement de l'application, la phase de calibrage du capteur, le meilleur temps de préchauffage du capteur est supérieur à 24 heures. Le capteur de gaz qui est conçu pour interagir avec un microcontrôleur et surveiller en permanence la fuite de gaz dans l'air et l'envoie de ces informations au microcontrôleur. Lorsque la concentration de gaz dépasse le seuil toléré, le microcontrôleur alerte automatiquement le consommateur sur la présence des fuites en envoyant des SMS d'alerte via GSM aux numéros stockés, active la LED (alarme visuelle) et le Buzzer (alarme sonore), ferme la vanne principale d'alimentation en gaz et actionne le ventilateur d'aération.

### III.2.8.1 Circuit de simulation (Isis-Proteus)

La figure III.15 représente notre circuit de simulation (Isis-Proteus).

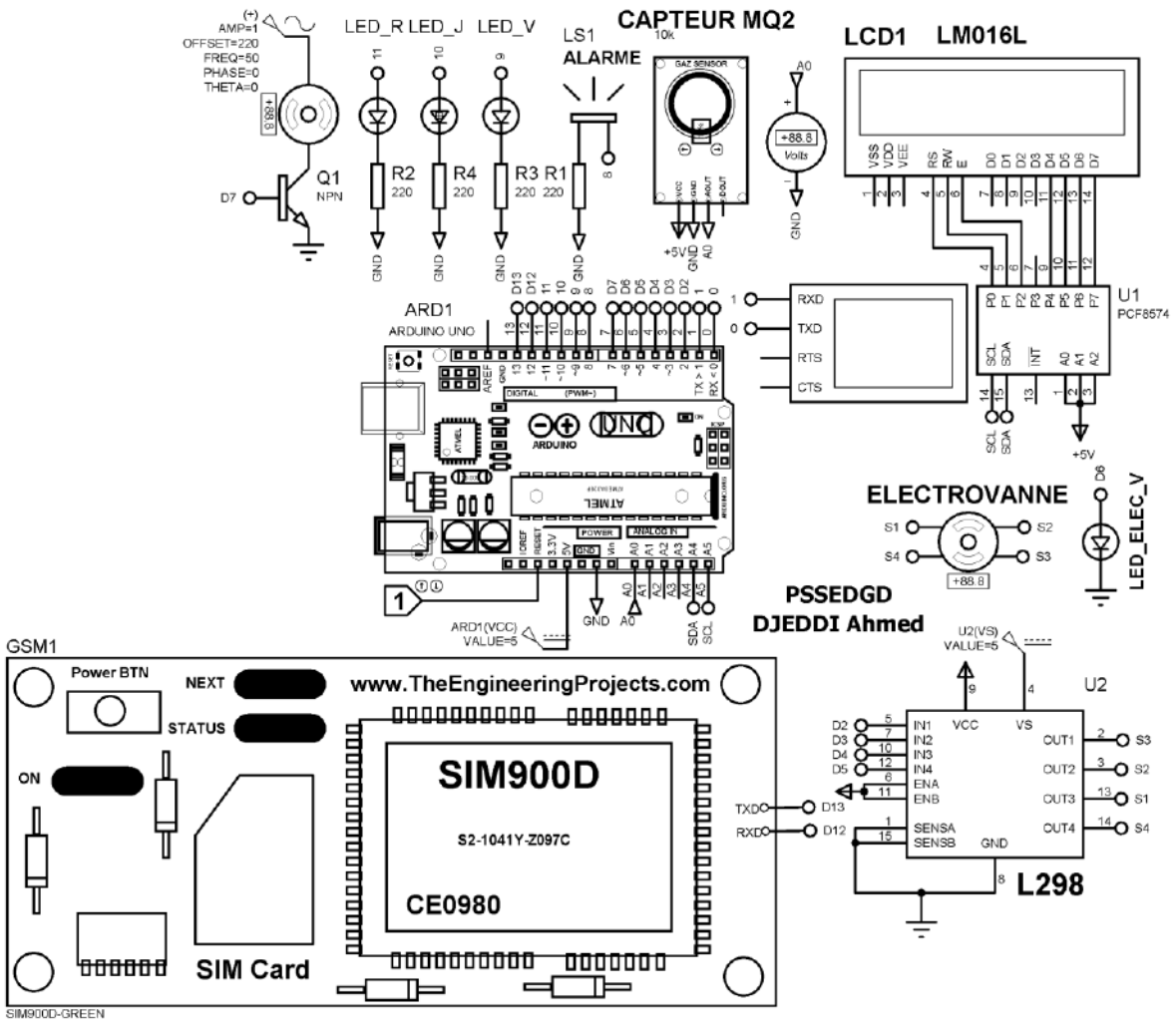



Figure III:15 Schéma du circuit de simulation (Isis-Proteus)

### III.2.8.2 Ensemble de l'application pratique

La figure III.16 montre l'image de l'ensemble de l'application pratique : Arduino, capteur MQx, L298, moteur, GSM, Buzzer, ventilateur, afficheur et LEDs.

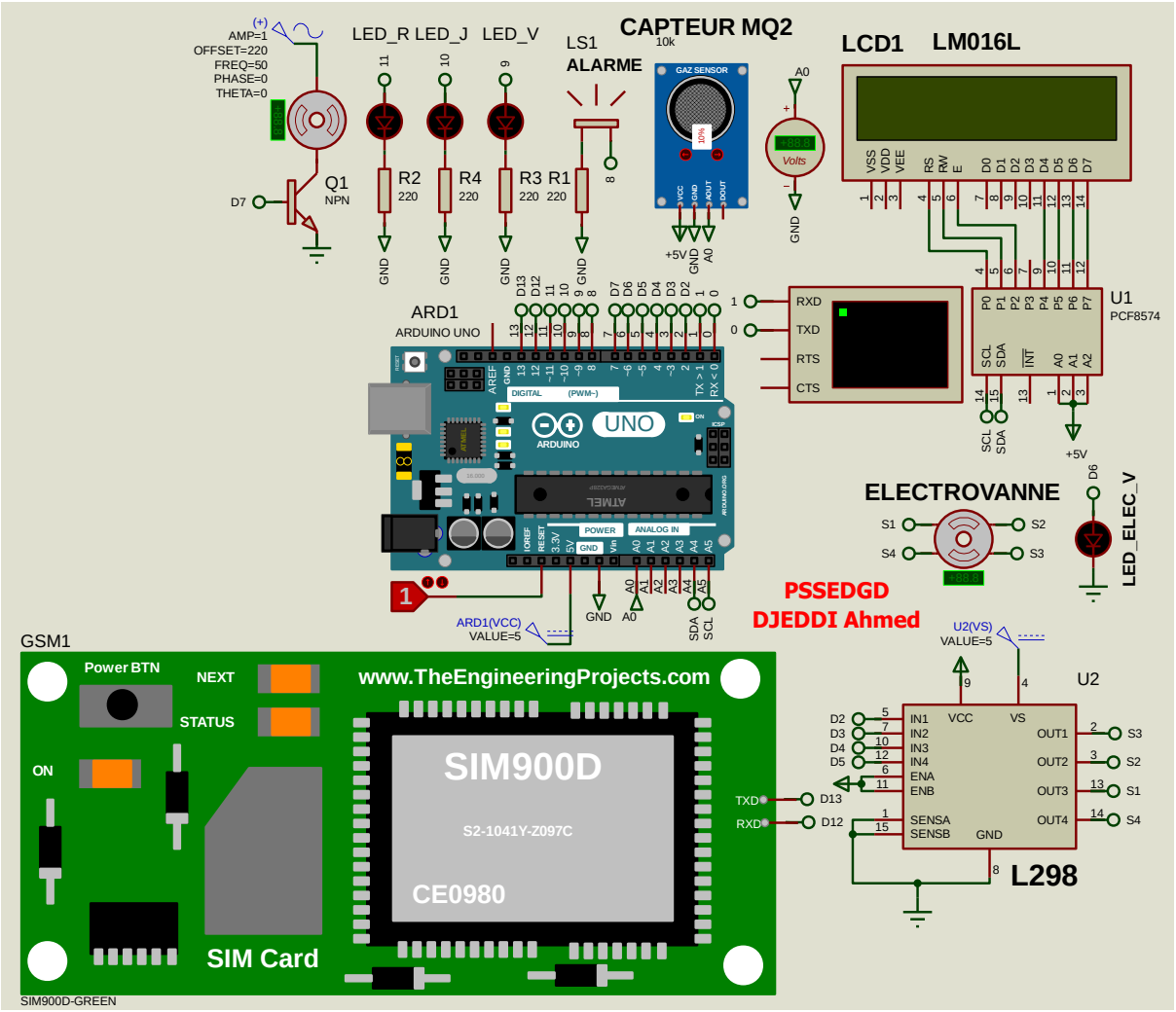



Figure III:16 L'application virtuelle du circuit de détection de gaz dans (Isis-Proteus)

## III.3 Partie software

Pour programmer les étapes de fonctionnement de notre système, l'environnement Arduino-IDE est utilisé. Notre programme comporte différentes routines ; chacune est destinée à remplir une tache spécifique :

### III.3.1 Programme Principal

Le « Programme Principal » : où on configure le microcontrôleur pour les différentes entrées analogiques et numériques, ainsi que les sorties digitales à utiliser. Aussi, on déclare les différentes constantes relatives aux nombres d'échantillons et l'intervalle d'échantillonnage à prendre dans les phases de calibrage et de mesure, la résistance de charge utilisée et les points

de mesure qui donnent la résistance relative par rapport à la concentration du gaz mesuré en échelle logarithmique, cette dernière dépond du type de gaz ainsi que du capteur utilisé.

La figure III.17 représente l'organigramme principal de détection de gaz.

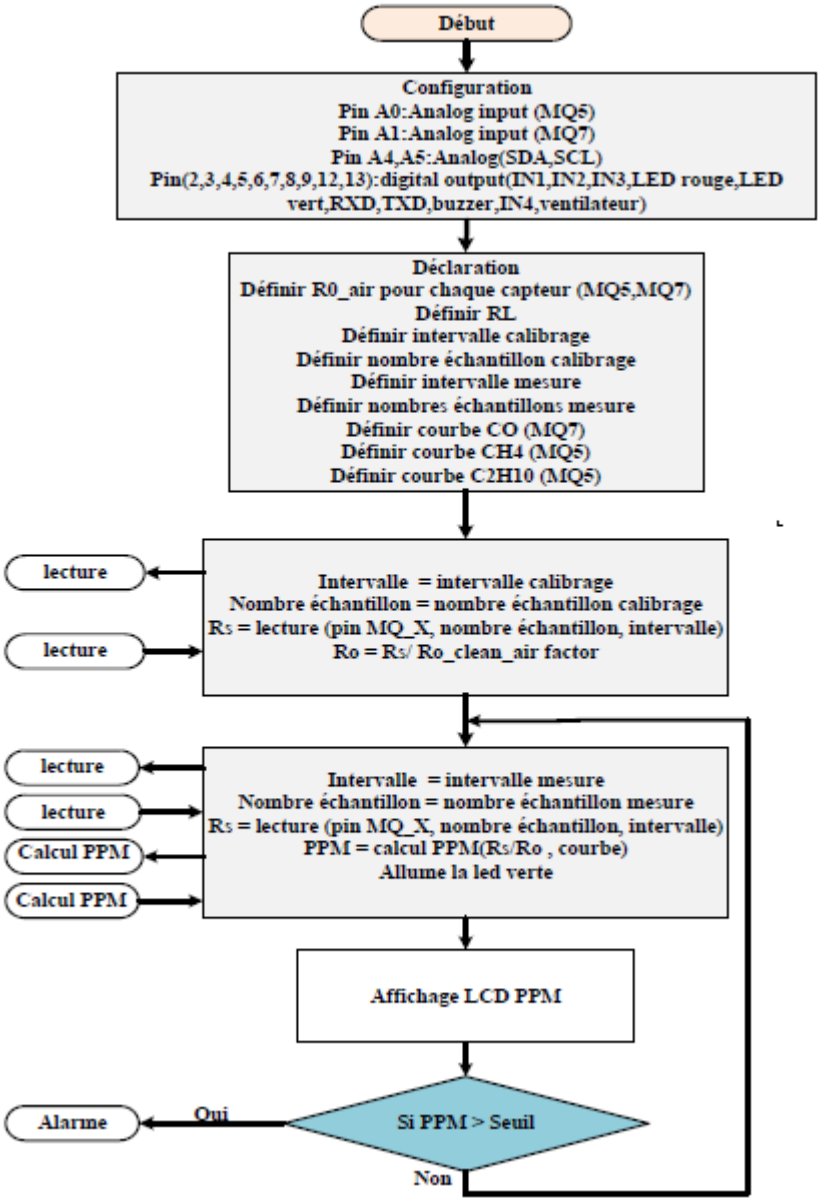



Figure III:17 Organigramme principale de détection de gaz

### III.3.2 Fonction Lecture

La fonction « Lecture » : cette routine est destinée à donner une valeur moyenne des mesures issues du capteur après conversion analogique numérique, et ceci pour la mesure et le calibrage.

L'organigramme (Figure III.18) suivant représente la fonction de « lecture ».

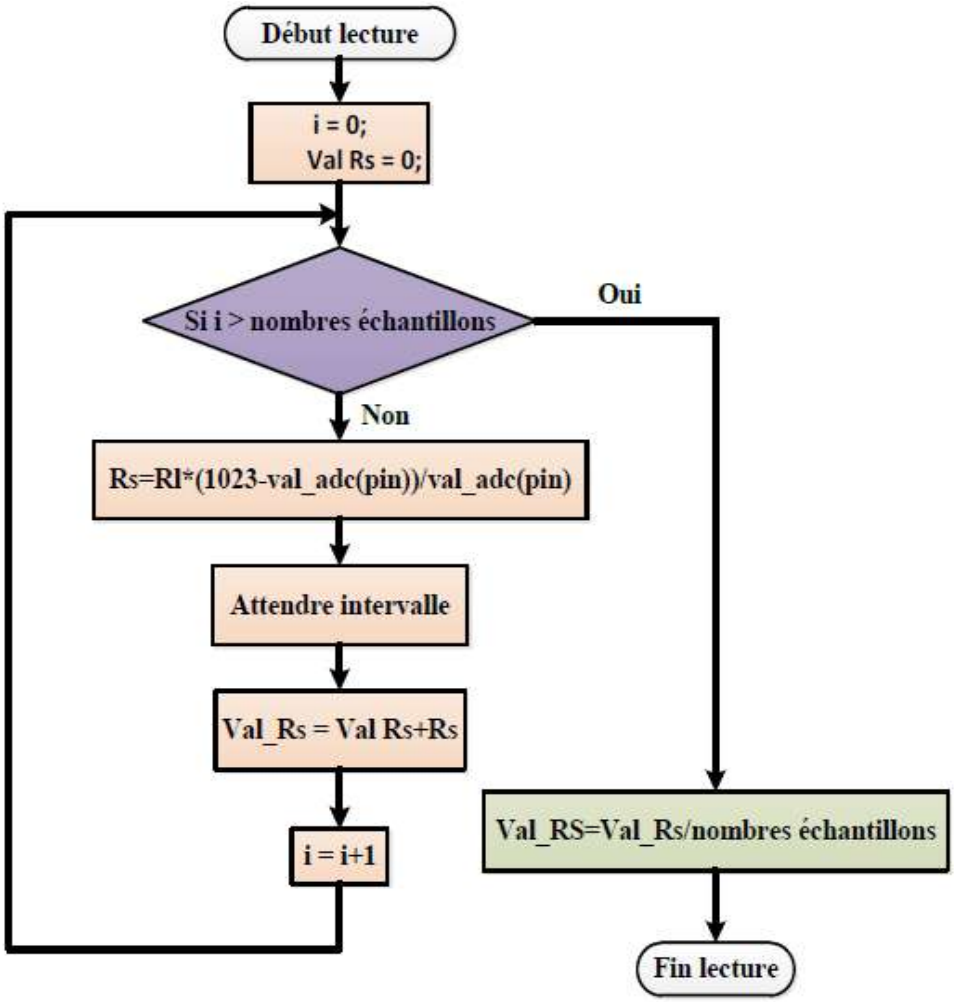



Figure III:18 Organigramme de lecture

### III.3.3 Fonction Calcul PPM

La fonction « Calcul PPM » : qui permet l'interpolation depuis la courbe de la datasheet qui représente la réponse du capteur au gaz (Figure II.12 et Figure II.13), pour trouver la concentration correspondante du gaz mesuré.

L'organigramme suivant (Figure III.19) représente la fonction de calcul de concentration : « Calcul PPM ».

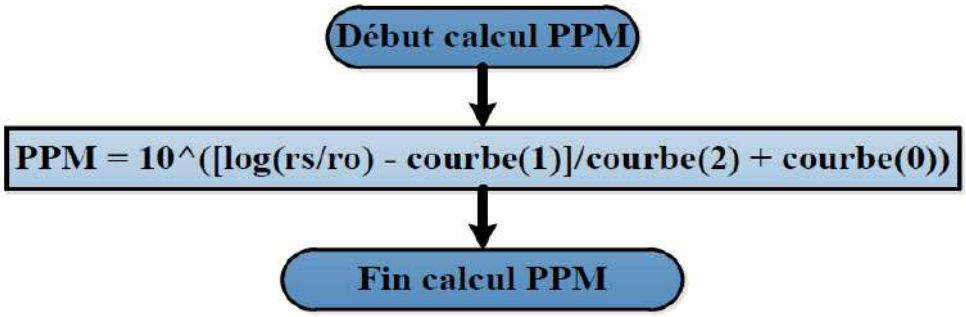



Figure III:19 Organigramme « calcul PPM »

### III.3.4 Fonction Alarme

La fonction « Alarme » : qui permet d'appeler ou d'envoyer un sms à un ou plusieurs numéros de téléphones stockés, activer l'alarme visuelle (LED rouge), activer l'alarme sonore (Buzzer), fermer la vanne principale (Moteur pas-à-pas) et actionner le ventilateur d'aération.

La figure III.20 représente l'organigramme de la fonction d'alarme.

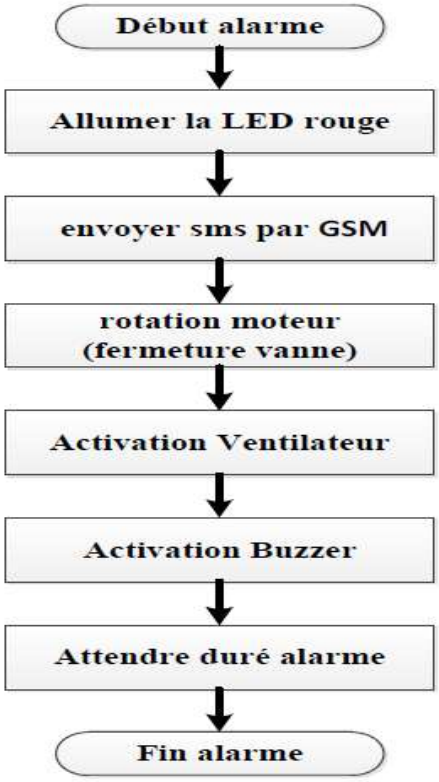



Figure III:20 Organigramme de la fonction d'alarme

### III.3.5 Code programme Arduino IDE

Les figures de III.21 à III.29 représentent le code complet de notre programme de détection des gaz.

```
/*************************************** 1 partie du programme ***************************************************/
/*****************************************************************************************************************/
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27,16,2);
#include <SoftwareSerial.h>
SoftwareSerial SIM900(12, 13);
int D2=2, D3=3, D4=4, D5=5, D6=6, D7=7, buzzer=8, LED_V=9, LED_J=10, LED_R=11;
int D12=12, D13=13;
int d=400;
#define         Broche_Cpateur                  (0)  //Définition de l'entrée analogique A0 comme broche de capteur
#define         RL_VALUE                        (5)  //Définition de la résistance de charge sur la carte, en kilo ohms
#define         Facteur_R0                    (9.83) //R0: Résistance du capteur dans l'air propre
                                                     //qui est dérivé du graphique dans la feuille de données
#define         NEPPE                           (3)  //Définition du nombre d'échantillons prélevé dans la phase d'étalonnage
#define         ITEE                          (500)  //Définition d'intervalle de temps (en milliseconde) entre chaque échantillon
                                                     //dans la phase d'étalonnage
#define         NEPFN                           (3)  //Définition de nombre d'échantillons préleveé en fonctionnement normal
#define         ITEEN                           (5)  //Définition d'intervalle de temps (en millisecondes) entre chaque échantillon
#define         GAZ_GPL                         (0)  //en fonctionnement normal
#define         GAZ_CO                          (1)
#define         GAZ_FUMEE                       (2)
float           Courbe_GPL[3]  =  {2.3, 0.21, -0.47};      //Deux points sont tirés de la courbe.
                                                           //avec ces deux points, une ligne est formée qui est "approximativement équivalente"
                                                           //à la courbe d'origine.
                                                           //format de données:{ x, y, pente}; point1: (lg200, 0,21), point2: (lg10000, -0,59)
float           Courbe_CO[3]  =  {2.3, 0.72, -0.34};      //format de données:{ x, y, pente}; point1: (lg200, 0,72), point2: (lg10000, 0.15)
float           Courbe_Fumee[3] = {2.3, 0.53, -0.44};      //format de données:{ x, y, pente}; point1: (lg200, 0,53), point2: (lg10000, -0,22)
float           R0          =  10;                        //R0 est initialisé à 10 kilo ohms
```

Figure III:21 1 partie du programme

```
/*************************************** 2 partie du programme ***************************************************/
/*****************************************************************************************************************/
void setup()
                                                                                {
 pinMode(buzzer,OUTPUT); pinMode(LED_V,OUTPUT);          //
 pinMode(LED_J,OUTPUT); pinMode(LED_R,OUTPUT);           //
 noTone(buzzer); digitalWrite(LED_V, LOW);               //Brouches LEDs et Buzzer avec un état initial
 digitalWrite(LED_J, LOW); digitalWrite(LED_R, LOW);     //désactivé

 pinMode(D2,OUTPUT);pinMode(D3,OUTPUT);
 pinMode(D4,OUTPUT);pinMode(D5,OUTPUT);                  //
 pinMode(D6,OUTPUT);                                     //
 digitalWrite(D2, LOW); digitalWrite(D3, LOW);           //
 digitalWrite(D4, LOW); digitalWrite(D5, LOW);           //Brouches driver L298
 digitalWrite(D6, LOW);                                  //et son LED
 pinMode(D7,OUTPUT); pinMode(D12,OUTPUT);
 pinMode(D13,OUTPUT);
 digitalWrite(D7, LOW); digitalWrite(D12, LOW);
 digitalWrite(D13, LOW);

  SIM900.begin(2400);                                    //Définition du débit de données en bits par seconde (baud) pour la transmission
                                                         // des données GSM en série
  Serial.begin(9600);                                    //Communication entre la carte Arduino à travers un port série ( UART ou USART).
                                                         //Il communique sur les broches numériques 0 (RX) et 1 (TX).
  lcd.init();                                            //Initialiser l'écran lcd
  lcd.backlight();
  lcd.print(" IN HTE NAME OF");
  lcd.setCursor(4,1);
  lcd.print(" ALLAH ");
```

Figure III:22 2 partie du programme

```
/*****************************************  3  partie du programme  ********************************************/
/*************************************************************************************************************/
  Serial.println(" ************* IN HTE NAME OF ALLAH ************* ");
  delay(500);
  lcd.clear();
  lcd.print("  PFE  PSSEDGD  ");
  Serial.println("PFE PSSEDGD ");
  delay(500);
  lcd.setCursor(2,1);
  lcd.print("DJEDDI Ahmed ");
  Serial.println("DJEDDI Ahmed ");
  delay(1000);lcd.clear();
  lcd.print("Calibration...");
  Serial.println("Calibration...\n");
  R0 = Calibration_Cap(Broche_Cpateur);                      //Calibrage du capteur...
  lcd.clear();
  lcd.print("Calibration est");
  lcd.setCursor(3, 1);
  lcd.print("terminee...");
  Serial.println("La calibration est terminee...\n");
  delay(500);
  lcd.clear();
  lcd.setCursor(2, 1);
  lcd.print("R0=");
  lcd.print(R0);
  lcd.print("KOhm");
  Serial.print("R0=");
  Serial.print(R0); Serial.println("kohm");
  delay(500); lcd.clear();                                                    }
```

Figure III:23 3 partie du programme

```
/*****************************************  4  partie du programme  ********************************************/
/*************************************************************************************************************/
void loop()
                                                                              {
Affichage();                                                 //Fonction d'affichage des données: GPL, CO et FUMEE
                                                             //en ppm sur l'écran lcd et serial monitor
if (      MQGAZ_Pourcentage(Lire_Cap(Broche_Cpateur) / R0, GAZ_GPL) < 800 &&
          MQGAZ_Pourcentage(Lire_Cap(Broche_Cpateur) / R0, GAZ_CO) < 200  &&
          MQGAZ_Pourcentage(Lire_Cap(Broche_Cpateur) / R0, GAZ_FUMEE)< 300  )    {
  noTone(buzzer);
  digitalWrite(LED_V, HIGH);
  digitalWrite(LED_J, LOW); digitalWrite(LED_R, LOW);
  lcd.clear(); lcd.print("  AIR CORRECT"); delay(500);                        }
else                                                                          {
  if (    MQGAZ_Pourcentage(Lire_Cap(Broche_Cpateur) / R0, GAZ_GPL)   < 4000 &&
          MQGAZ_Pourcentage(Lire_Cap(Broche_Cpateur) / R0, GAZ_CO)    < 400   &&
          MQGAZ_Pourcentage(Lire_Cap(Broche_Cpateur) / R0, GAZ_FUMEE)< 400   )   {
  noTone(buzzer);
  digitalWrite(LED_V, LOW);
  digitalWrite(LED_J, HIGH); digitalWrite(LED_R, LOW);
  lcd.clear(); lcd.print("  AIR MALSAIN"); delay(400);                        }
else                                                                          {
  digitalWrite(D6,HIGH);digitalWrite(D7,HIGH);
  digitalWrite(D2,LOW); digitalWrite(D3,LOW); digitalWrite(D4,HIGH); digitalWrite(D5,LOW);
  delay(d);
  digitalWrite(D2,LOW); digitalWrite(D3,HIGH); digitalWrite(D4,LOW); digitalWrite(D5,LOW);
  delay(d);
  digitalWrite(D2,LOW); digitalWrite(D3,LOW); digitalWrite(D4,LOW); digitalWrite(D5,HIGH);
  delay(d);
```

Figure III:24 4 partie du programme

```
/*****************************************  5  partie du programme  ********************************************/
/*************************************************************************************************************/
digitalWrite(D2,HIGH); digitalWrite(D3,LOW); digitalWrite(D4,LOW); digitalWrite(D5,LOW);
delay(d);
SIM900.println("ATD + +213777777777;"); // Entrez le numéro de mobile ici pour appeler
Serial.println(" ");                   //La commande ATD compose le numéro de téléphone.
Serial.println("Appel en cours...");
Serial.println("ATD + +213777777777;");
delay(100);
SIM900.println();
delay(2000);
SIM900.println("ATH");
Serial.println("ATH");
  do                                                                                {
  tone(buzzer,200,1000);
  digitalWrite(LED_V, LOW);
  digitalWrite(LED_J, LOW); digitalWrite(LED_R, HIGH);
  lcd.clear(); lcd.print("ALERTE POLLUTION");delay(500);
  Affichage();                                                                      }
  while ( 4000 <  MQGAZ_Pourcentage(Lire_Cap(Broche_Cpateur) / R0, GAZ_GPL)     ||
          400 <   MQGAZ_Pourcentage(Lire_Cap(Broche_Cpateur) / R0, GAZ_CO)      ||
          400 <   MQGAZ_Pourcentage(Lire_Cap(Broche_Cpateur) / R0, GAZ_FUMEE)   );
  digitalWrite(D6,LOW);digitalWrite(D7,LOW);
  digitalWrite(D2,LOW); digitalWrite(D3,LOW); digitalWrite(D4,LOW); digitalWrite(D5,HIGH);
  delay(d);
  digitalWrite(D2,LOW); digitalWrite(D3,HIGH); digitalWrite(D4,LOW); digitalWrite(D5,LOW);
  delay(d);
  digitalWrite(D2,LOW); digitalWrite(D3,LOW); digitalWrite(D4,HIGH); digitalWrite(D5,LOW);
  delay(d);                                                                         } } }
```

Figure III:25 5 partie du programme

```
/*****************************************  6  partie du programme  ********************************************/
/*************************************************************************************************************/
/***************** Calc_Resistance_Cap ***********************************
  Entrée:       Tension_brute - valeur brute lue à partir de adc (analog-to-digital converter), qui représente la tension
  Sortie :      la résistance calculée du capteur
  Remarques :   Le capteur et la résistance de charge forment un diviseur de tension. Compte tenu de la tension
                à travers la résistance de charge et sa résistance, la résistance du capteur
                pourrait être dérivé.
*************************************************************************************/
float Calc_Resistance_Cap(int Tension_brute)
  { return ( ((float)RL_VALUE * (1023 - Tension_brute) / Tension_brute));       }
/************************** Calibration_Cap ***********************************
  Entrée:       broche_capteur - canal analogique
  Sortie:       R0 du capteur
  Remarques :   Cette fonction suppose que le capteur est dans l'air pur. Il utilise
                Calc_Resistance_Cap pour calculer la résistance du capteur dans l'air pur
                puis le divise avec Facteur_R0. Facteur_R0 est à peu près
                10, qui diffère légèrement entre les différents capteurs.
*************************************************************************************/
float Calibration_Cap(int broche_capteur)                                       {
  int i;
  float val = 0;
  for (i = 0; i < NEPPE; i++)       {   //take multiple samples
    val += Calc_Resistance_Cap(analogRead(broche_capteur));
    delay(ITEE);                }
  val = val / NEPPE;                //calculate the average value
  val = val / Facteur_R0;                   //divided by Facteur_R0 yields the R0
                                            //according to the chart in the datasheet
  return val;                                                                   }
```

Figure III:26 6 partie du programme

```
/*****************************************  7  partie du programme  ***********************************************/
/*****************************************************************************************************************/
/***************************  Lire_Cap ***************************************
  Entrée:      broche_capteur - canal analogique
  Sortie:      R_Cap : résistance du capteur
  Remarques :  Cette fonction utilise Calc_Resistance_Cap pour calculer la resistenc de capteur (R_Cap).
               Le R_Cap change au fur et à mesure que le capteur est dans le consentement différent de la cible
               gaz. Les temps d'échantillonnage et l'intervalle de temps entre les échantillons peuvent être configurés
               en modifiant la définition des macros.
*****************************************************************************/
float Lire_Cap(int broche_capteur)                                            {
  int i;
  float R_Cap = 0;
  for (i = 0; i < ITEEN; i++)                                                 {
    R_Cap += Calc_Resistance_Cap(analogRead(broche_capteur));
    delay(NEPFN);                                                             }
  R_Cap = R_Cap / ITEEN;
  return R_Cap;                                                               }
/***************************  MQGAZ_Pourcentage ********************************
Obtenir le pourcentage de gaz

  Entrée:     R_Cap_R0_ratio - R_Cap divisé par R0
              gas_id - type de gaz cible
  Sortie:     ppm du gaz cible
  Remarques:  Cette fonction transmet différentes courbes à la fonction MQ_Pourcentage qui
              calcule les ppm (parties par million) du gaz cible.
*****************************************************************************/
```

Figure III:27 7 partie du programme

```
/*****************************************  8  partie du programme  ***********************************************/
/*****************************************************************************************************************/

int MQGAZ_Pourcentage(float R_Cap_R0_ratio, int gas_id)                       {
  if ( gas_id == GAZ_GPL )                                                    {
    return MQ_Pourcentage(R_Cap_R0_ratio, Courbe_GPL);                        }
  else if ( gas_id == GAZ_CO )                                                {
    return MQ_Pourcentage(R_Cap_R0_ratio, Courbe_CO);                         }
  else if ( gas_id == GAZ_FUMEE )                                             {
    return MQ_Pourcentage(R_Cap_R0_ratio, Courbe_Fumee);                      }
  return 0;                                                                   }
/***************************  MQ_Pourcentage ***********************************
Obtenir le pourcentage

  Entrée:     R_Cap_R0_ratio - R_Cap divisé par R0
              Point_Courbe - pointeur vers la courbe du gaz cible
  Sortie:     ppm du gaz cible
  Remarques : En utilisant la pente et un point de la ligne. Le x(valeur logarithmique de ppm)
              de la ligne pourrait être dérivée si y(R_Cap_R0_ratio) est fourni. Comme il s'agit d'un
              coordonnée logarithmique, la puissance de 10 est utilisée pour convertir le résultat en non logarithmique
              valeur.

*****************************************************************************/
int  MQ_Pourcentage(float R_Cap_R0_ratio, float *Point_Courbe)                {
  return (pow(10, ( ((log(R_Cap_R0_ratio) - Point_Courbe[1]) / Point_Courbe[2]) + Point_Courbe[0])));
                                                                              }
```

Figure III:28 8 partie du programme

```
/*************************************** 9 partie du programme ***************************************/
/*****************************************************************************************************/
void Affichage()                                                    {
  Serial.print("GPL:");
  Serial.print(MQGAZ_Pourcentage(Lire_Cap(Broche_Cpateur) / R0, GAZ_GPL) );
  Serial.print( " ppm" );
  Serial.print("    ");
  Serial.print("CO:");
  Serial.print(MQGAZ_Pourcentage(Lire_Cap(Broche_Cpateur) / R0, GAZ_CO) );
  Serial.print( " ppm" );
  Serial.print("    ");
  Serial.print("FUMEE:");
  Serial.print(MQGAZ_Pourcentage(Lire_Cap(Broche_Cpateur) / R0, GAZ_FUMEE) );
  Serial.println( " ppm" );
  lcd.setCursor(0, 0);
  lcd.print("GPL:");
  lcd.print(MQGAZ_Pourcentage(Lire_Cap(Broche_Cpateur) / R0, GAZ_GPL) );
  lcd.print("     ");
  lcd.setCursor(9, 0);
  lcd.print("CO:");
  lcd.print(MQGAZ_Pourcentage(Lire_Cap(Broche_Cpateur) / R0, GAZ_CO) );
  lcd.print("       ");
  lcd.setCursor(0, 1);
  lcd.print("FUMEE:");
  lcd.print(MQGAZ_Pourcentage(Lire_Cap(Broche_Cpateur) / R0, GAZ_FUMEE) );
  lcd.setCursor(11, 1);
  lcd.print("(PPM)" );
  delay(500);                                                       }
```

Figure III:29 9 partie du programme

### III.3.6 Interface et simulation

Les figures de III.21 à III.29 montrent l'interface complète et la simulation de notre circuit de détection des gaz.

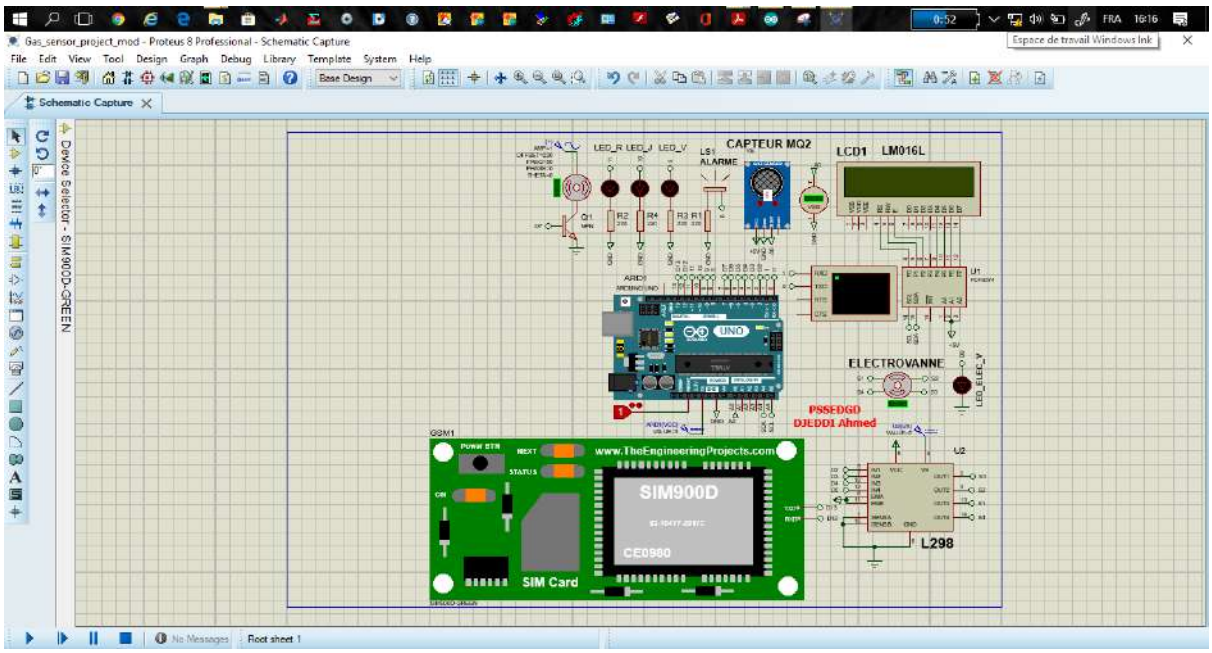



Figure III:30 Interface du circuit

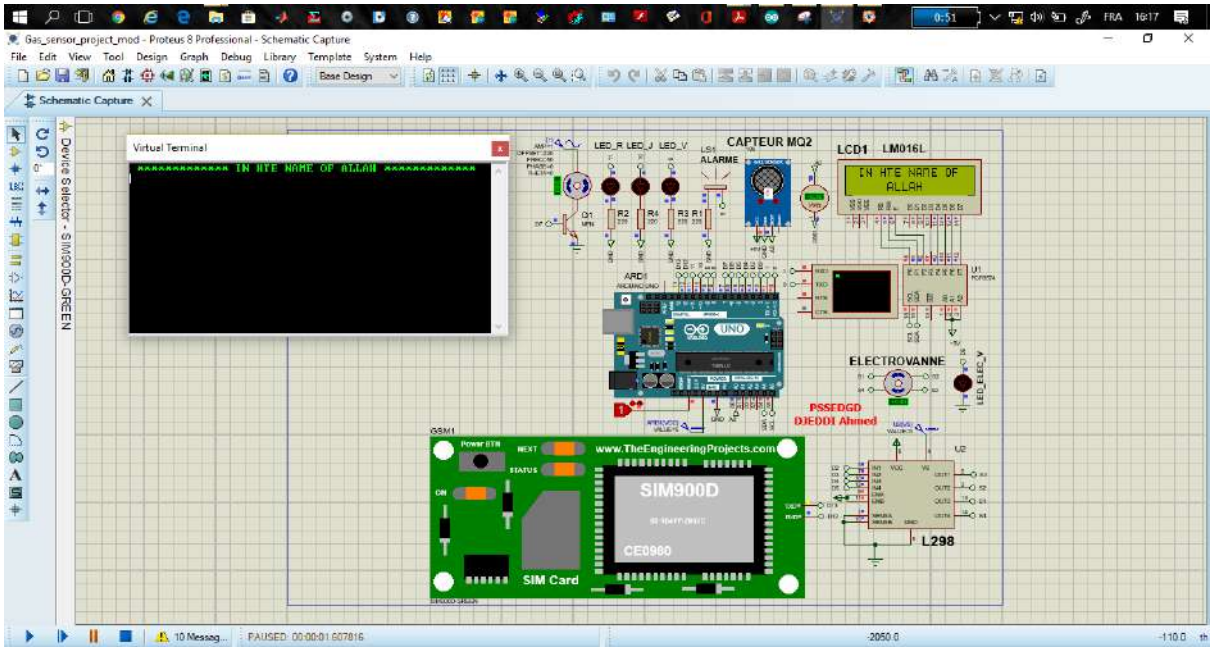



Figure III:31 Interface du circuit au cours de simulation

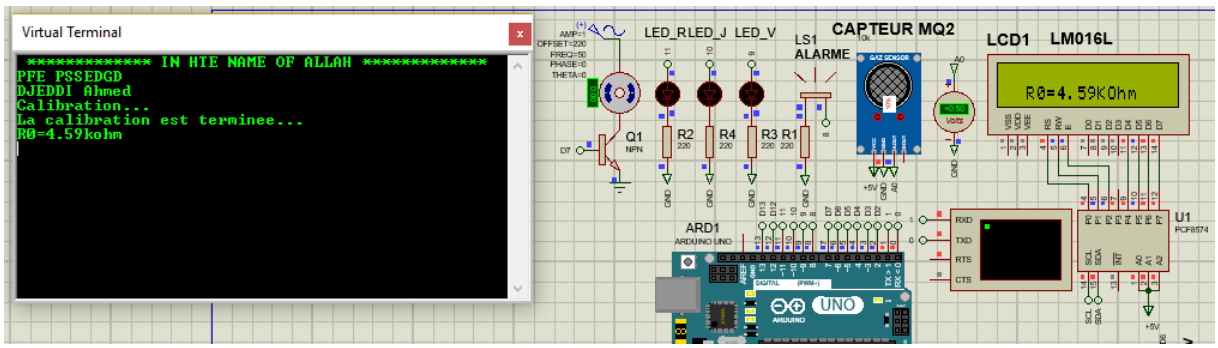



Figure III:32 Interface du circuit au cours calibration

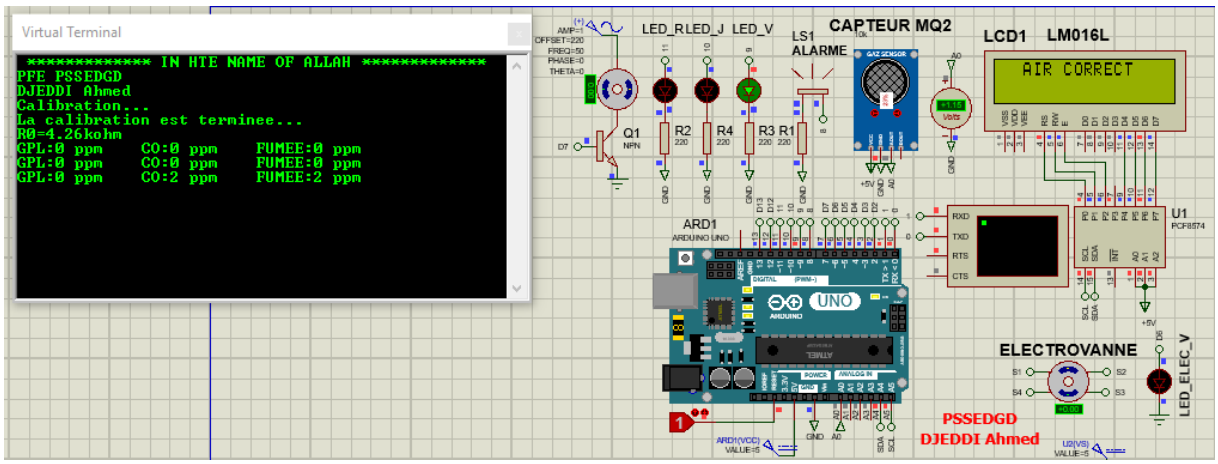



Figure III:33 Détection de l'air pure

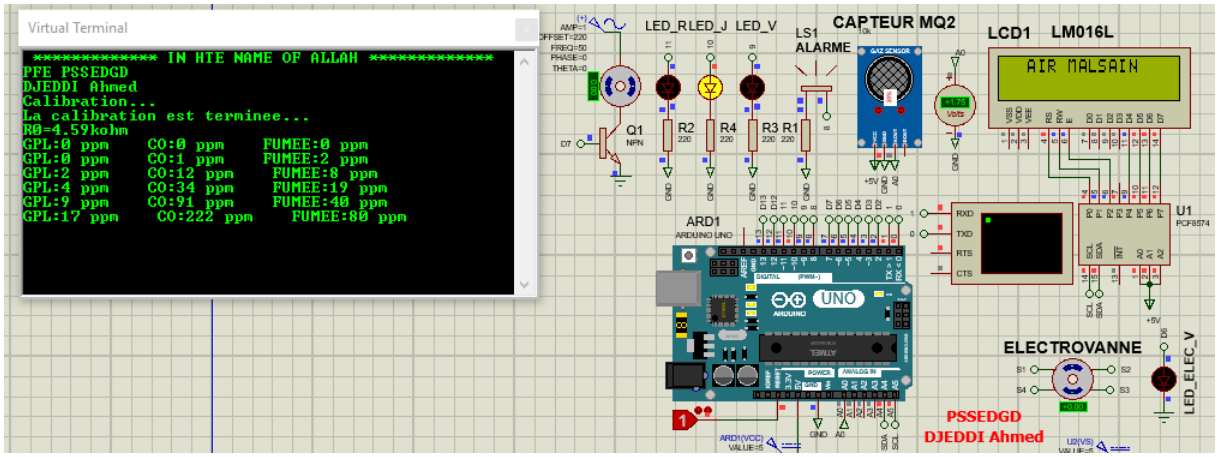



Figure III:34 Détection de l'air malsain

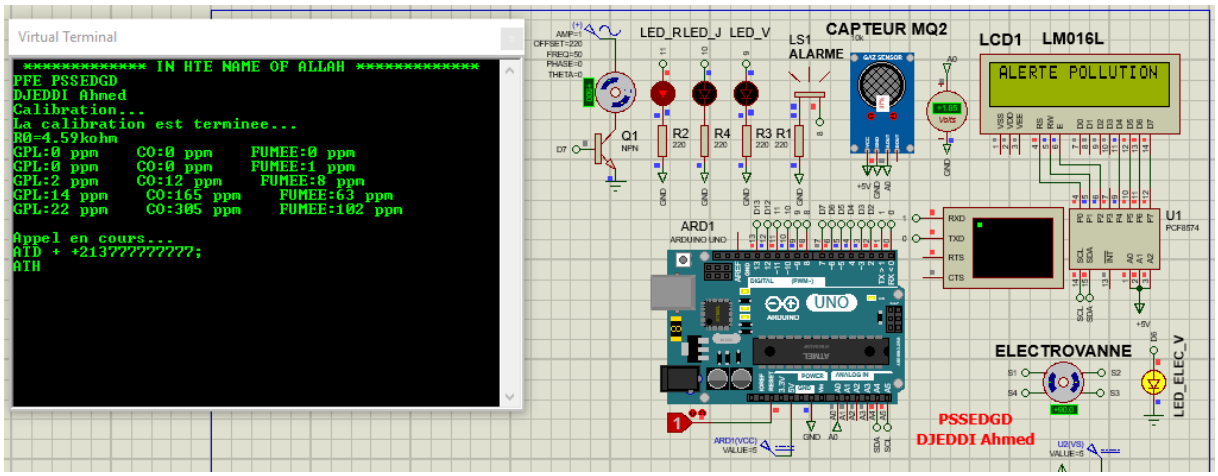



Figure III:35 Détection de l'air pollué

## III.4 Conclusion

Nous avons présenté dans ce chapitre, les phases de développement de l'application dans son coté Hardware : le choix du matériel et les considérations pratiques, et le coté software : la programmation Arduino IDE pour microcontrôleur, les différents calculs et calibrages relatifs diverses fiches techniques fournies par les fabricants de capteurs pour diverses mesures précises des différents gaz.

L'expérience de ce chapitre nous a aidés à apprendre certains concepts et à établir des systèmes efficaces pour la détection basée sur Arduino, la mesure de la concentration de gaz dangereux, la prévention et l'avertissement.

# Conclusion générale

L'utilisation du gaz comme source d'énergie est essentielle à notre vie, et cette utilisation présente un risque pour la vie humaine en cas de fuite. Le monoxyde de carbone en particulier reste la première cause d'accidents mortels en Algérie.

A cet égard, nous avons proposé un système de détection et de mesure des concentrations de gaz toxiques (monoxyde de carbone et méthane) à des fins de prévention et d'alerte. Ce système est également essentiel dans l'environnement domestique et industriel et permet de sauver des vies et de protéger les biens.

Ce projet a permis de lier les études théoriques des clusters électrifiés avec des aspects interdisciplinaires, dont l'électronique embarquée et l'informatique industrielle. Pour développer cette application, plusieurs étapes étape par étape ont été démarrées et complétées par l'ensemble de l'application, ou chaque étape a créé deux parties de la conception. Logiciel : inclut la programmation en langage Arduino dans l'environnement Arduino IDE. Matériel : Inclut la simulation de circuit basée sur la simulation (ISIS PROTEUS).

Les résultats obtenus démontrent l'efficacité du système dans la détection et l'avertissement en temps réel de la présence de l'un des gaz nocifs et son évolutivité à d'autres gaz.

À l'avenir, nous envisagerons d'introduire des calibrateurs plus sophistiqués pour améliorer les performances des applications en réduisant les coûts de la batterie tout en minimisant la puissance de fonctionnement de la batterie. Ce produit est vendu sur le marché à faible coût.

# Bibliographie

[1] https://tel.archives-ouvertes.fr/tel-00192045/document.

[2] http://www.memoireonline.com/05/12/5830/Les-systemes-embarques.html.

[3] raisin.u-bordeaux.fr/IMG/pdf/Introduction_SE_kadionik_raisin.pdf.

[4] www.liberte-algerie.com/actualite/alerte-aux-accidents-de-gaz-85551.

[5] www.cna.dz/.../1/.../RP_Spéciale_Accidents_Domestiques_2015.pdf.

[6] https://www.presse-dz.com/revue-de-presse/monoxyde-de-carbone-300-algeriens-sont morts-en-une-annee.

[7] http://www.centreantipoisons.be/monoxyde-de-carbone/le-monoxyde-decarbone-co-en-d-tail/quelles-sont-les-concentrations-toxiques-de

[8] https://fablab.univ-tlse3.fr/wiki/images/9/.../Cours_arduino_v0.2.pdf

[9] projet.eu.org/pedago/sin/tutos/arduino.odt

[10] http://www.ShieldList.org/

[11] http://www.louisreynier.com « c'est quoi Arduino ».

[12] Nicolas GOILAV Geoffrey LOI « Arduino » (page12, 13, 14, 15, 16, 17, 18, 267, 268, 302, 303)

[13] https://www.arduino.cc/en/Main/ArduinoBoardUno

[14] http://www.techniques-ingenieur.fr/base-documentaire/mesuresanalyses-th1/metrologie-relative-aux-gaz-42539210/explosimetresdetecteurs-

de-gaz-r2380/

[15] https://tel.archivesouvertes.fr/tel00975072/file/TH2013PEST1173_complete

.pdf

[16] http://www.hdevbot.com/datasheet/datasheet_MQ-2.pdf

[17] C. Tavernier, « Arduino applications avancées ». Version Dunod.

[18] http://fr.hobbytronics.co.uk/mq7-carbon-monoxide-sensor

[19] https://www.sparkfun.com/datasheets/Sensors/Biometric/MQ-7.pdf

[20] https://docs.particle.io/assets/datasheets/electronsensorkit/MQ-2.pdf

[21] projet.eu.org/pedago/sin/tutos/arduino.odt

[22] tinkbox.ph/sites/tinkbox.ph/files/.../5V_BUZZER_MODULE.pdf

[23] sensorembedded.com/product_extra_files/GSM%20SIM800.pdf

[24] www.micronet91.com/wa.../Introduction_20sur_20le_20L298N.pdf

[25] electroniqueveynes.free.fr/IMG/pdf/Bus_I2C.pdf

[26] https://www.didel.com/product-category/carte/cardui

# Index


## *A*

afficheur · 22, 32, 33, 34, 45
Alarme · 49
analogique · 8, 15, 28, 34, 35, 48
application · 1, 2, 7, 32, 44, 45, 46, 56, 57
Arduino · 1, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 31, 32, 33, 34, 35, 36, 38, 39, 40, 41, 42, 43, 45, 46, 50, 56, 57, 58, 59, 64

## *B*

buzzer · 36, 64
Buzzer · 36, 37, 44, 45, 49

## *C*

CH4 · 1, 4, 25, 26
circuit · 14, 20, 25, 26, 29, 31, 32, 37, 38, 43, 44, 45, 46, 54, 55, 57
code · 11, 50
concentration · 2, 4, 5, 10, 22, 23, 24, 26, 28, 32, 37, 41, 44, 47, 48, 56, 64
conversion · 48

## *D*

datasheet · 48, 59
DC · 21, 25, 27, 29, 40, 41
détection · i, 3, 25, 26, 27, 29, 31, 32, 43, 46, 47, 50, 54, 56, 57, 64
Driver · 38

## *F*

fuite · 1, 4, 37, 41, 42, 44, 57

## *G*

gaz · 1, i, 1, 2, 3, 4, 5, 10, 22, 23, 24, 25, 26, 27, 28, 30, 31, 32, 35, 37, 41, 42, 43, 44, 46, 47, 48, 50, 54, 56, 57, 58, 59, 64
gaz dangereux · i, 1, 3, 4, 22, 32, 35, 37, 41, 42, 56, 64
GSM · 42, 43, 44, 45, 59, 64

## *I*

interface · 7, 11, 22, 28, 54
interpolation · 48
intoxication · 3, 5
Isis · 44, 45, 46

## *L*

L298N · 38, 39
LCD · 32, 33, 34
Lecture · 33, 48
LED · 20, 21, 35, 36, 44, 49, 64
logarithmique · 47
logiciel · 6, 7, 9, 11, 32, 43

## *M*

mesure · 1, 2, 11, 22, 26, 31, 32, 43, 46, 48, 56, 57
microcontrôleur · 1, 11, 12, 13, 14, 16, 17, 21, 28, 32, 38, 44, 46, 56
Module · 34, 42

monoxyde de carbone · 1, 3, 4, 10, 28, 29, 57
moteur · 37, 38, 39, 40, 41, 45
MQ · 25, 27, 28, 30, 59

## *N*

numérique · 8, 15, 48

## *O*

organigramme · 47, 48, 49

## *P*

particules par million · 4
programmation · 1, 2, 11, 12, 18, 21, 31, 56, 57
Proteus · 44, 45, 46, 64

## *S*

SIM · 42
SIM800 · 42
simulation · 2, 11, 44, 45, 54, 55, 57
software · 46, 56
suffocation · 1, 3
système embarqué · i, 1, 3, 4, 6, 7, 9
systèmes de sécurité · 1

## *V*

ventilateur · 41, 42, 44, 45, 49
ventilation · 1, 3, 4

## ملخص

يهدف هذا العمل إلى إضافة مبحث حول برمجة و محاكاة دوائر كهربائية كاشفة للغازات السامَّة، و التي تعتمد على لوحة الأردوينو أونو لغرض اجتناب حوادث الاختناق المُميتة.

عندما يكتشف النظام عن طريق مُستشعرالغاز أن تركيز الغازات السامَّة قد تجاوز الحدود المسموحة، يقوم بإرسال رسالة نصيِّة قصيرة و يتصل برقم المستخدم عن طريق بطاقة GSM لإنذاره، و إنذار الأشخاص الموجودون في مكان التسرب عن طريق جهاز الإنذار المحتوي على أضواء و جرس التنبيه، و كذلك يقوم بقطع تغذية الغاز و تشغيل أجهزة التهوية.

## Résumé

Dans ce travail, nous avons introduit des recherches pour programmer et simuler des circuits de détection de gaz dangereux basés sur la carte Arduino UNO pour éviter les étouffements accidentels mortelles.

Lorsque le système utilise un capteur de gaz pour détecter que la concentration de gaz a dépassé un certain seuil, le système utilise une carte GSM pour appeler le numéro de téléphone et envoyer un SMS à l'utilisateur. Il vous alerte instantanément et alerte les personnes à domicile en activant l’alarme. Cela comprend des lumières LED, un buzzer simultané et prend des précautions en activant l'aspirateur pour réduire la concentration de gaz dans l'air et arrêter l'alimentation en gaz naturel.

## Abstract

In this work, we introduced research to program and simulate hazardous gas detection circuits based on Arduino UNO board to prevent accidental fatal choking.

When the system uses a gas sensor to detect that the gas concentration has exceeded a certain threshold, the system uses a GSM card to call the phone number and send an SMS to the user. It alerts you instantly and alerts people in the home by activating the alarm. This includes LED lights, a simultaneous buzzer and takes precautions by activating the vacuum cleaner to reduce the concentration of gas in the air and stop the natural gas supply.



## Mots clés

Système d’alerte, Carte Arduino, Carte GSM, capteur de gaz, driver L298, Arduino UNO, Proteus.