FILIÈRE

MP1ERTA

DANS LE CADRE DE

BUREAU D'ÉTUDES EN INFORMATIQUE INDUSTRIELLE

GD : DÉTECTEUR DE GAZ

RÉALISER PAR

BENAMOR BRAHIM

ANNÉE UNIVERSITAIRE

2023 - 2024

Contents

List of Figures

Introduction

La sécurité contre les fuites de gaz est un enjeu majeur dans de nombreux environnements , qu'ils soient industriels ou domestiques . Ils peuvent présenter des dangers sérieux pour la santé et la sécurité des individus , allant des risques d'incendie et d'explosion aux effets néfastes sur la santé en cas d'inhalation .

Dans ce contexte , le détecteur de gaz joue un rôle vital dans la sécurité des personnes et des biens en permettant la détection précoce de fuites de gaz .

En identifiant rapidement la présence de gaz nocifs , il contribue à la prévention d'accidents graves et permet la mise en place de mesures correctives pour assurer la sécurité des occupants. Ainsi, son bon fonctionnement et son efficacité revêtent une importance capitale pour prévenir les incidents liés aux gaz dangereux .

Chapter 1

Hardware

On s'intéresse dans cette chapitre sur `le matériel` qu'on est utilisé : on donne **la définition** de chaque composant , leurs **caractéristiques** ainsi **démontrer notre choix** .
Pour concevoir un modèle de base de détecteur de gaz , plusieurs composants sont nécessaires pour assurer différentes fonctions , tels qu'un composant de contrôle , un capteur de détection de gaz et des éléments d'affichage des résultats .

Soit le matériel utilisé dans notre projet ainsi leurs caractéristiques

1.1 Carte Arduino

Les cartes Arduino offrent une plateforme flexible et peu coûteuse pour les projets électroniques , allant des projets simples de domotique et de capteurs à des projets plus complexes de robotique et d'automatisation industrielle .
Généralement , on distingue divers types des cartes tels que la carte Arduino Arduino Uno ; la carte Arduino Mega ; la carte Arduino Due ; la carte Arduino Leonardo ; la carte Arduino MKR WiFi 1010 . Dans la suite , on aura utilisée la carte Arduino Uno .

1.1.1 Définition

La carte Arduino Uno Dip REV3 est une carte de développement électronique open-source basée sur le microcontrôleur **ATmega328P** .
Elle est conçue pour faciliter la création de projets électroniques interactifs en offrant un environnement de développement convivial et facile à utiliser .

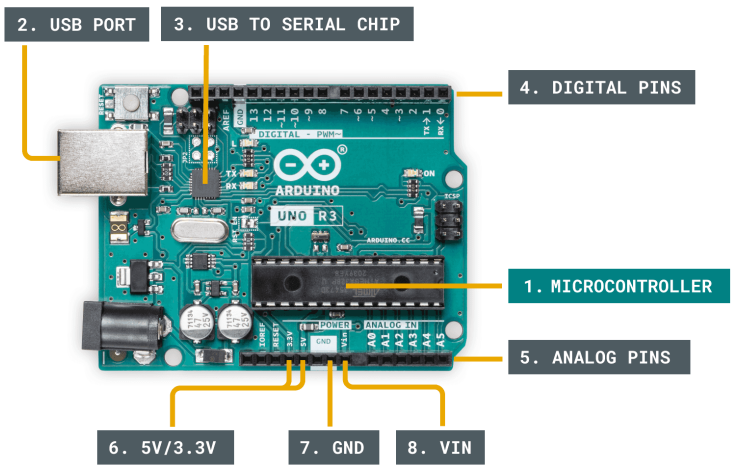

Figure 1.1: Carte Arduino Uno Dip REV3

1.1.2 Caractéristiques

Soit les principaux caractéristiques du la carte Arduino Uno Dip REV3 :

- ♠Microcontrôleur : ATmega328P
- ♠Tension de fonctionnement : 5V
- ♠Tension d'entrée (recommandée) : 7-12V
- ♠Tension d'entrée (limite) : 6-20V
- ♠Broches d'entrée/sortie numériques : 14 (dont 6 peuvent être utilisées en PWM)
- ♠Broches d'entrée analogique : 6
- ♠Courant continu par broche de sortie : 40 mA
- ♠Courant continu pour 3.3V : 50 mA
- ♠Mémoire flash : 32 KB (ATmega328P) dont 0.5 KB utilisés par le bootloader
- ♠RAM : 2 KB (ATmega328P)
- ♠EEPROM : 1 KB (ATmega328P)
- ♠Fréquence d'horloge : 16 MHz

1.1.3 Justification du choix

Dans le cadre de réaliser un modèle basic du détecteur de gaz , on utilise la carte Arduino Uno Dip REV3 en raison de ses caractéristiques , notamment :

- ♠Microcontrôleur ATmega328P assez puissant pour la programmation et le contrôle .
- ♠La tension de fonctionnement de 5V est adaptée à des nombreux capteurs de gaz .
- ♠Possibilité d'utiliser 6 broches d'entrée/sortie numériques en PWM pour le contrôle précis des composants .

♠6 broches d'entrée analogique pour la lecture des autres capteurs de gaz .
♠Mémoire flash de 32 KB pour stocker le programme du détecteur .
♠Fréquence d'horloge de 16 MHz pour des opérations rapides et précises .

1.2 Capteur de Gaz

Dans le cadre des capteurs de gaz , on distingue divers modèles tels que capteur de gaz MQ2 ; capteur de gaz MQ-3 ; capteur de gaz MQ-4 ; capteur de gaz MQ-5 ... ect , notamment que le capteur MQ2 est le capteur le plus basique de la chaîne MQ .
Puisque on a dans le cas de réaliser un modèle basic du détecteur de gaz , le capteur de gaz MQ2 est répondu au nos besoins .

1.2.1 Définition

Le capteur de gaz MQ2 est un composant électronique sensible aux gaz, capable de détecter plusieurs types de gaz , notamment le méthane , le propane , le monoxyde de carbone et l'alcool . Leur principe de fonctionnement est basé sur l'utilise une résistance électrique interne qui varie en fonction de la concentration de gaz détectée dans l'air , fournissant ainsi une indication de la présence et de la quantité de gaz dans l'environnement .

Figure 1.2: Capteur de gas MQ2

1.2.2 Caractéristiques

Soit les principaux caractéristiques du le capteur de gaz MQ2 :
♠Matériel : Mélange
♠Taille : 32 mm x 22 mm x 27 mm
♠Puce principale : LM393, ZYMQ - 2
♠Tension de fonctionnement : 5 VCC

1.2.3 Justification du choix

Dans le cadre de réaliser un modèle basic du détecteur de gaz , on choisit le capteur de gaz MQ2 en raison de ses caractéristiques , notamment :
♠Sensibilité aux gaz multiples .
♠Taille compacte qui le rend facile à intégrer dans des dispositifs de détection de gaz .
♠ La puce principale contribuent à la fonctionnalité en précision du capteur de gaz .
♠La tension de fonctionnement de 5V est compatible au tension délivré par la carte Arduino Uno Dip REV3 .

1.3 Afficheur LCD

Les afficheurs LCD sont largement utilisés dans de nombreux domaines tels que les calculateurs ; les instruments de mesure ect .
On distingue une variété de modèles est disponible pour répondre à différents besoins notamment l'afficheur 7SEG-DIGITAL ; l'afficheur LCD 16*2 ; l'afficheur LCD I2C 2*16 qui l'on aura utilisée .

1.3.1 Définition

L'afficheur LCD I2C 2*16 est un écran à cristaux liquides (LCD) de 2 lignes par 16 caractères qui utilise le protocole de communication I2C pour être contrôlé . Ce protocole de communication série utilisé pour permettre à plusieurs composants électroniques de communiquer entre eux sur un même bus de données avec seulement deux fils : **une ligne de données** (SDA) et une **ligne d'horloge** (SCL) . Il permet le transfert de données bidirectionnel et la synchronisation entre les composants connectés . Cela permet de connecter facilement l'écran à d'autres périphériques via un nombre réduit de fils .

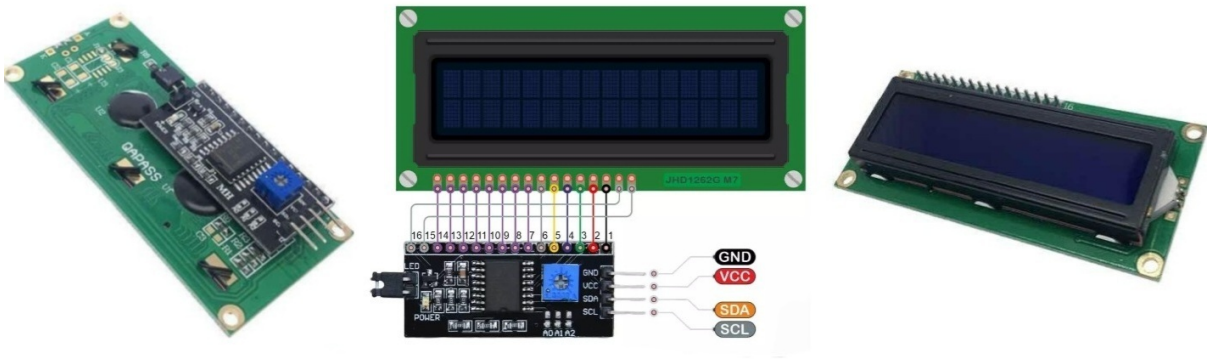

Figure 1.3: Afficheur LCD I2C 16x2

1.3.2 Caractéristiques

Soit les principaux caractéristiques du l'afficheur LCD I2C 16x2 :

- ♠Tension d'alimentation : +5 V
- ♠Prend en charge le protocole I2C
- ♠Nombre de broches : 4 broches
- ♠Taille : 80 * 36 * 18mm

1.3.3 Justification du choix

Dans le cadre de réaliser un modèle basic du détecteur de gaz , on choisit l'afficheur LCD I2C 16x2 en raison de ses caractéristiques , notamment :

- ♠La tension de fonctionnement de 5V est compatible au tension délivré par la carte Arduino Uno Dip REV3 .
- ♠En utilisant 4 broches en entrée, cela permet d'économiser davantage de broches sur la carte Arduino.
- ♠Taille compacte qui le rend facile à intégrer dans des dispositifs de détection de gaz .

1.4 Buzzer actif

1.4.1 Définition

Un buzzer actif est un type de dispositif de **signalisation acoustique** qui produit un son lorsqu'il est alimenté en courant . Contrairement à un buzzer passif qui nécessite une oscillation externe pour produire un son , un buzzer actif contient un oscillateur interne qui génère un signal audio lorsqu'il est alimenté en courant .

Figure 1.4: Buzzer actif

1.4.2 Caractéristiques

Soit les caractéristiques principales du buzzer sont les suivantes :

♠Tension : 5 v
♠Courant: $< 250mA$

1.5 Module feu tricolore à LED

1.5.1 Définition

Un module feu tricolore à LED est un dispositif électronique utilisé pour créer un signal de feu tricolore (rouge , jaune et vert) .

Figure 1.5: Module feu tricolore à LED

1.5.2 Caractéristiques

Soit les caractéristiques principales du module feu tricolore à LED sont les suivantes :

♠Faible consommation d'énergie
♠Contrôle programmable

1.6 Principe du fonctionnement

Le fonctionnement du détecteur de gaz MQ2 repose sur la détection de la présence de gaz dans l'air en mesurant la résistance électrique de son élément sensible . Ce capteur , conçu pour détecter les gaz inflammables tels que **le propane** ; **le butane** ; **le méthane** et **l'hydrogène** , se compose de deux parties principales : une partie sensible au gaz et une partie de détection de la température .

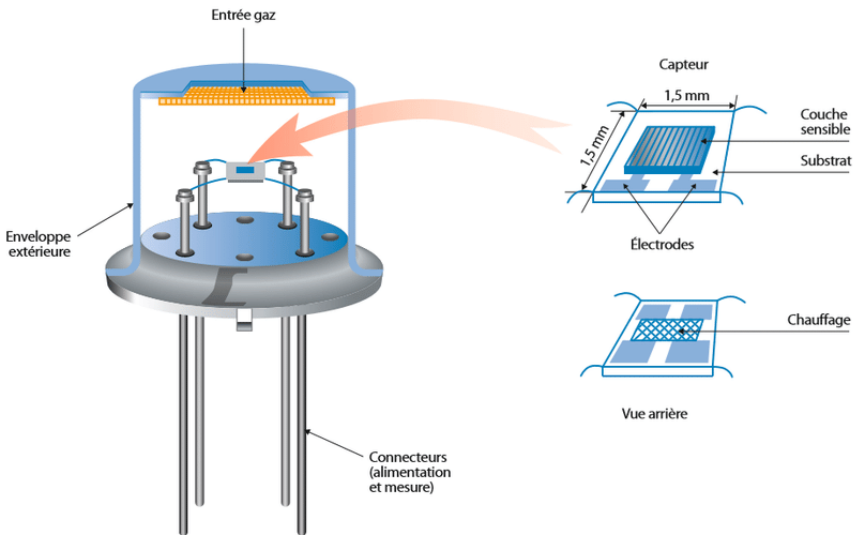

Figure 1.6: Structure intérieure du capteur de gaz MQ2

La partie sensible au gaz intègre un élément chauffant et un élément de détection , en l'occurrence une couche de film d'oxyde métallique chauffée par un courant électrique . Lorsqu'un gaz inflammable entre en contact avec cette couche , la conductivité de l'élément sensible au gaz augmente , entraînant une diminution de la résistance électrique du capteur .

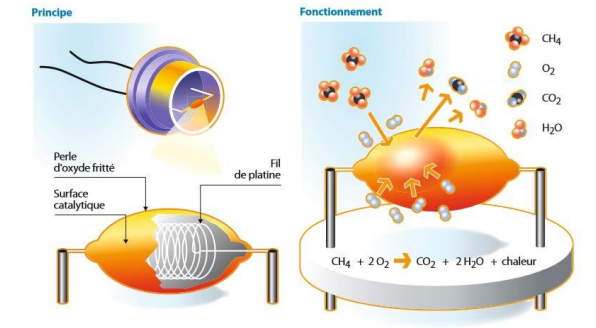

Figure 1.7: Déroulement à l'intérieur du capteur de gaz MQ2 lors de détection du gaz

- Puisque l'objectif est d'assurer la sécurité en surveillant les niveaux de gaz dans un environnement spécifique , et en utilisant un capteur de gaz MQ2 connecté une broche analogique de la carte Arduino UnoDip REV3 , le système peut détecter la présence de gaz nocifs dans l'air . Les LEDs de différentes couleurs (vert ; orangé ; rouge) du module feu tricolore ainsi le buzzer servent à fournir des indications visuelles et sonores aux utilisateurs en cas de niveaux de gaz dangereux .

- La LED verte indique une zone sécurisée où les niveaux de gaz sont sûrs . Lorsque les niveaux de gaz dépassent un seuil prédéfini mais ne sont pas encore considérés comme dangereux , la LED orangé s'allume pour avertir les utilisateurs de la présence de gaz potentiellement nocifs . En même temps , le buzzer émet un son d'alerte modéré .

- Si les niveaux de gaz deviennent critiques et dépassent un seuil de danger défini , la LED rouge s'allume et le buzzer émet un signal sonore fort et continu pour alerter immédiatement les personnes présentes de la nécessité d'évacuer ou de prendre des mesures correctives .

- En résumé , ce système offre une surveillance en temps réel des niveaux de gaz , ce qui permet de détecter rapidement les situations dangereuses et de protéger la sécurité des personnes et des biens dans l'environnement surveillé .

Soit le montage approche du modèle détecteur de gaz :

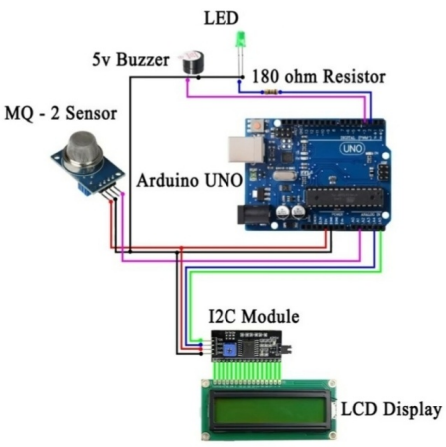

Figure 1.8: Montage approche du modèle détecteur de gaz

Chapter 2

Software

2.1 Le code

Soit Le code qui permet de répondre à notre principe de fonctionnement :

```
#include <Wire.h>
#include <SoftwareSerial.h>
#include <LiquidCrystal_I2C.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27,16,2);
// pins
int gasPin = A0; //Broche de capteur de gas
int greenled = D5; //Broche de led vert
int orangeled = D6; //Broche de led orangé
int redled = D7; //Broche de led rouge 2
int buzzer = D8; //Broche du buzzer
int gasSensorThres = 300;
int gasSensor;
int x;
void setup() {
pinMode(greenled,OUTPUT); //Led vert est confiuguré comme sortie
pinMode(orangeled,OUTPUT); //Led orangé est confiuguré comme sortie
pinMode(redled,OUTPUT); //Led rouge est confiuguré comme sortie
pinMode(buzzer,OUTPUT); // buzzer est confiuguré comme sortie
pinMode(gasPin,INPUT); // capteur de gas est confiuguré comme entré
digitalWrite(buzzer,LOW);
digitalWrite (greenled  ,HIGH);
}
 void loop() {
  gasSensor = analogRead(gasPin);
  Serial.println(gasSensor);
  Serial.print("gasPin value");
  Serial.println(gasSensor);
  delay(100);
  if ((gasSensor > gasSensorThres) && (gasSensor<570))
  {
   lcd.clear();
   digitalWrite(orangeled,HIGH);
   digitalWrite(redled,LOW);
   tone(buzzer,2000);
   digitalWrite(greenled,LOW);
   lcd.setCursor(1,0);
   lcd.print("Attention:fuite Apparaitre");
   x=(gasPin*100)/1023;
    lcd.setCursor(0,1);
    lcd.print(x,'%');
  }
  else if (gasSensor > gasSensorThres)
   {
    lcd.clear();
    digitalWrite(orangeled,LOW);
    digitalWrite(redled,HIGH);
    tone(buzzer,7000);
    digitalWrite(greenled,LOW);
    lcd.setCursor(1,0);
    lcd.print("Attention:Taux de Gaz détectée");
    x=(gasPin*100)/1023;
    lcd.setCursor(0,1);
    lcd.print(x,'%');
    delay(1000);
   }
   else if (gasSensor < gasSensorThres)
  {
   lcd.clear();
   digitalWrite(orangeled,LOW);
   digitalWrite(redled,LOW);
   noTone(buzzer);
   digitalWrite(greenled,HIGH);
   lcd.setCursor(0,0);
   lcd.print("zone de sécurité");
  }
```

Figure 2.1: Le code qui permet de détecter le gaz à l'aide des différentes composants

2.2 Analyse du code

• **#include** <**Wire.h**> : permet d'inclure la bibliothèque Wire pour la communication I2C .
• **#include** <**SoftwareSerial.h**> : permet d'inclure la bibliothèque SoftwareSerial pour la communication série logicielle
• **#include** <**LiquidCrystal_I2C.h**> : permet d'inclure la bibliothèque LiquidCrystal_I2C pour l'utilisation d'un écran LCD I2C .
• **#include LiquidCrystal_I2C lcd(0x27,16,2)** : permet de créer une instance de l'objet lcd de la classe LiquidCrystal_I2C avec l'adresse 0x27 , 16 colonnes et 2 lignes .
• **int gasPin = A0** : permet de définir la broche pour le capteur de gaz .
• **int greenled = D5** : permet de définir la broche pour la LED verte .
• **int orangeled = D6** : permet de définir la broche pour la LED orangé .
• **int redled = D7** : permet de définir la broche pour la LED rouge .
• **int buzzer = D8** : permet de définir la broche pour le buzzer .
• **int gasSensorThres = 300** : permet de définir le seuil du capteur de gaz .
• **int gasSensor** : permet de déclarer une variable pour stocker la lecture du capteur de gaz .
• **int x** : permet de Déclarer une variable pour calculer un pourcentage .

• **void setup()** • **pinMode(greenled,OUTPUT)** : permet de configurer la broche de la LED verte en sortie .
• **pinMode(orangeled,OUTPUT)** : permet de configurer la broche de la LED orangé en sortie .
• **pinMode(redled,OUTPUT)** : permet de configurer la broche de la LED rouge en sortie .
• **pinMode(buzzer,OUTPUT)** : permet de configurer la broche du buzzer en sortie .
• **pinMode(gasPin,INPUT)** : permet de configurer la broche du capteur de gaz en entrée .
• **digitalWrite(buzzer,LOW)** : permet de désactiver le buzzer .
• **digitalWrite (greenled ,HIGH)** : permet d'allumer la LED verte .
• **gasSensor = analogRead(gasPin)** : permet de lire la valeur analogique du capteur de gaz .
• **Serial.println(gasSensor)** : permet d'afficher la valeur mesurée par le capteur de gaz sur le moniteur série .
• **Serial.print("gasPin value")** : permet d'afficher une étiquette pour la valeur du capteur de gaz .
• **Serial.println(gasSensor)** : permet d'afficher la valeur du capteur de gaz sur le moniteur série .
• **delay(100)** : permet d'attendre 100 millisecondes .

• **if ((gasSensor** > **gasSensorThres)** && **(gasSensor**<**570))** : c'est une condition que si la valeur du capteur de gaz dépasse le seuil et est inférieure à 570 .
• **lcd.clear()** : permet d'effacer l'écran LCD .
• **digitalWrite(orangeled,HIGH)** : permet d'allumer la LED orangé .
• **digitalWrite(redled,LOW)** : assure que la LED rouge est éteindre .
• **tone(buzzer,2000)** : permet de générer une tonalité sur le buzzer à 2000 Hz .
• **digitalWrite(greenled,LOW)** : assure que la LED verte est éteindre .
• **lcd.setCursor(1,0)** : permet de positionner le curseur de l'écran LCD à la première colonne et à la première ligne .
• **lcd.print("Attention:fuite Apparaitre")** : permet d'afficher le message "Attention:fuite Apparaitre" sur l'écran LCD .
• **x=(gasPin*100)/1023** : permet de calculer le pourcentage de gaz dans l'air depuis la valeur mesurée par le capteur de gaz .
• **lcd.setCursor(0,1)** : permet de positionner le curseur de l'écran LCD à la première colonne et à la deuxième ligne .
• **lcd.print(x,'%')** : permet d'afficher le pourcentage sur l'écran LCD .

• **else if (gasSensor** > **gasSensorThres)** : c'est la deuxième condition que si la valeur du capteur de gaz dépasse le seuil .
• **digitalWrite(orangeled,LOW)** : assure que la LED orangé est éteindre .
• **digitalWrite(redled,HIGH)** : permet d'allumer la LED rouge .
• **digitalWrite(greenled,LOW)** : assure que la LED verte est éteindre .
• **lcd.setCursor(1,0)** : permet de positionner le curseur de l'écran LCD à la première colonne et à la première ligne .
• **lcd.print("Attention:Taux de Gaz détectée")** : permet d'afficher le message "Attention:Taux de Gaz détectée" sur l'écran LCD .
• **lcd.setCursor(0,1)** : permet de positionner le curseur de l'écran LCD à la première colonne et à la deuxième ligne .
• **lcd.print(x,'%')** : permet d'afficher le pourcentage sur l'écran LCD .
• **delay(1000)** : Attente de 1000 millisecondes .

• **else if (gasSensor** < **gasSensorThres)** : c'est la troisième condition que si la valeur du capteur de gaz est inférieure au seuil .
• **digitalWrite(orangeled,LOW)** : assure que la LED orangé est éteindre .

• **digitalWrite(redled,LOW)** : assure que la LED rouge est éteindre .
• **noTone(buzzer)** : permet d'arrêter la tonalité du buzzer .
• **digitalWrite(greenled,HIGH)** : permet d'allumer la LED verte .
• **lcd.setCursor(0,0)** : permet de positionner le curseur de l'écran LCD à la première colonne et à la première ligne .
• **lcd.print("zone de sécurité")** : permet d'afficher le message "zone de sécurité" sur l'écran LCD .

Chapter 3

Simulation

3.1 Montage réalisé avec Proteus

Soit le montage réalisé par le logiciel Proteus :

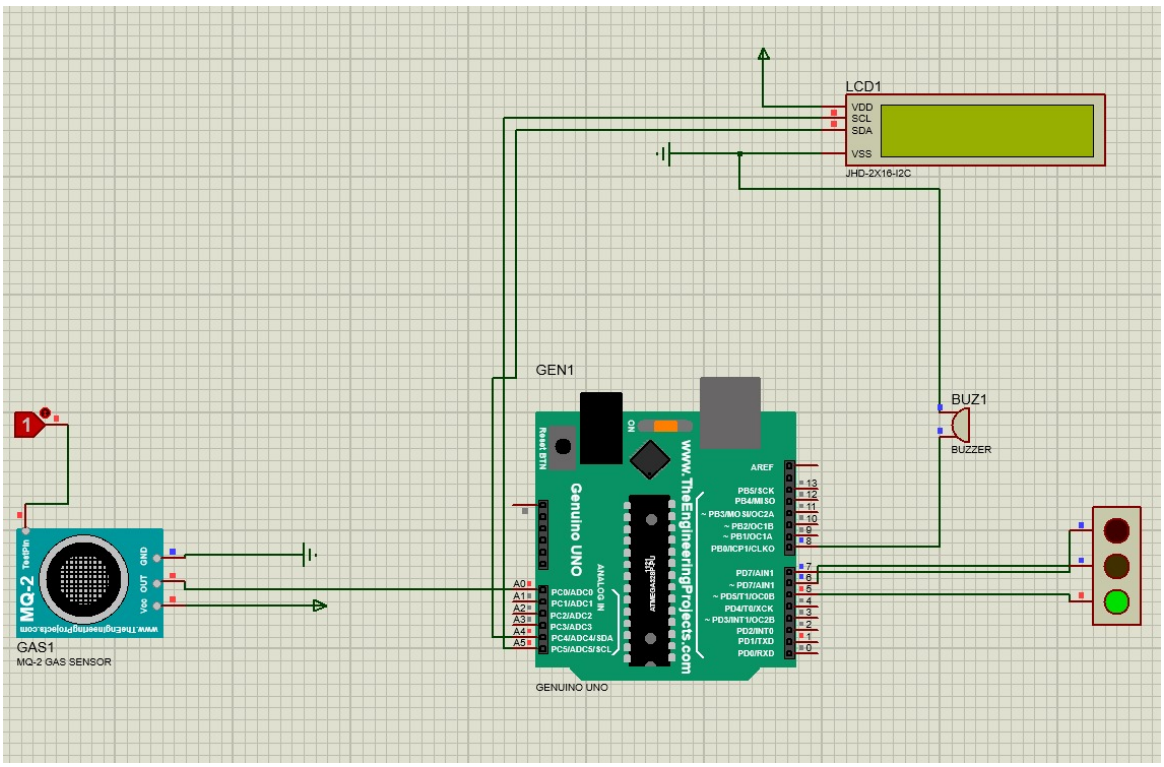

Figure 3.1: Montage du détecteur de gaz réalisé avec Proteus

3.2 Explications

On peut noté à propos le montage quelques explications notamment :

• **SCL (Serial Clock Line = Ligne d'Horloge Série)** : c'est la broche d'horloge utilisée pour synchroniser la transmission des données entre les composants connectés via le bus I2C . Elle est contrôlée par le maître et est utilisée pour indiquer quand les données peuvent être lues ou écrites .

• **SDA (Serial Data Line = Ligne de Données Série)** : c'est la broche de données utilisée pour transférer les informations entre les composants connectés . Les données sont envoyées en série sur cette ligne , avec un bit d'adresse suivi des données elles-mêmes .

• Les broches **SDA** et **SCL** sont respectivement reliées aux broches analogiques A4 et A5 de la carte Arduino .

• **VDD** : c'est la tension d'alimentation positive .

• **VSS** : c'est la tension d'alimentation négative ou la masse (ground) .

• Il faut faire attention , pour que le capteur de gaz MQ2 fonctionne correctement , il nécessite une bibliothèque appelée **GasSensorTEP.HEX** , qui doit être installée .

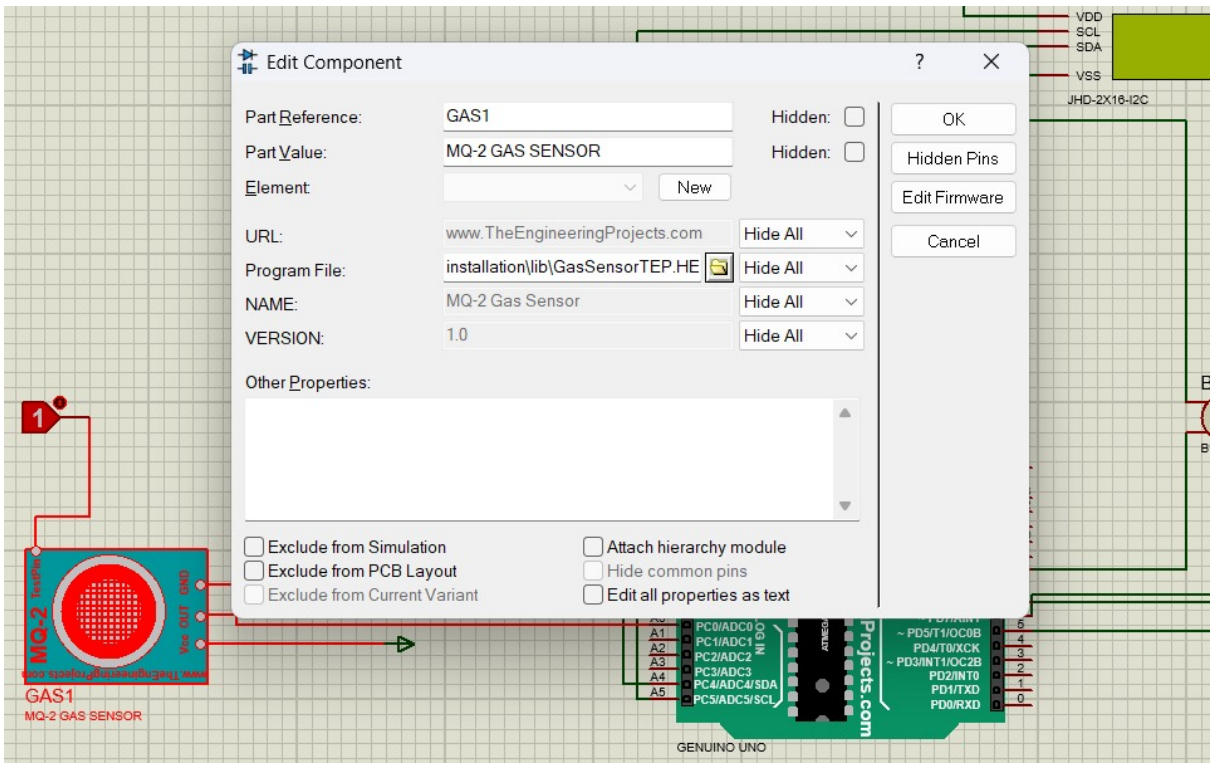

Figure 3.2: Configuration du capteur de gaz MQ2

• **Logic Toggler** associé au capteur de gaz MQ2 dans Proteus est utilisé pour simuler le changement d'état du capteur de gaz dans le logiciel . En effet , il est utilisé pour simuler la présence de gaz en activant ou désactivant la sortie du capteur dans la simulation .

Chapter 4

Perspectives

Les perspectives qui intéressent sont l'amélioration du modèle pour le rendre plus efficace et performant . On note :

• **Remplacement de la carte Arduino par une carte ESP32** : offre une connectivité Wi-Fi intégrée , ce qui permet une surveillance à distance facile via une application mobile ou un tableau de bord en ligne . Cela permet aux utilisateurs de surveiller les niveaux de gaz et de recevoir des alertes où qu'ils soient .

• **Intégration de capteurs de gaz spécialisés** : pour détecter des gaz spécifiques , tels que le monoxyde de carbone (CO), le dioxyde de carbone (CO2), le méthane (CH4) ou le gaz propane (C3H8). Cela permettrait une détection plus précise et une réponse rapide à différents types de dangers .

• **Ajout de capteurs de qualité de l'air** : tels que des capteurs de particules fines (PM2.5), de température et d'humidité , on pourrait fournir aux utilisateurs une image complète de leur environnement intérieur .

• **Intégration d'un écran tactile** : pourrait être ajouté pour offrir une interface utilisateur conviviale permettant aux utilisateurs de visualiser facilement les données de détection de gaz et de contrôler les paramètres du système .

• **Alertes par SMS ou e-mail** : en cas de détection de niveaux de gaz dangereux , le système pourrait envoyer des alertes automatiques par SMS ou e-mail aux utilisateurs enregistrés ainsi informer la caserne d'où une interactions des pompiers , leur permettant de prendre des mesures immédiates pour assurer leur sécurité .

• **Plateforme de cloud** : dont la quelle les données collectées par le système pourraient être stockées et analysées sur une plateforme de cloud , offrant ainsi une surveillance à long terme , la possibilité de visualiser les tendances et les schémas de détection de gaz , et la génération de rapports détaillés .

On résume que les améliorations proposées visent à transformer le système de détection de gaz en une solution plus avancée et polyvalente , alignée sur les principes de l'**IOT** et de la "**Smart House**" . En intégrant **des capteurs spécialisés , une connectivité Wi-Fi , une interface utilisateur conviviale et des fonctionnalités d'alerte avancées** , le système devient un élément essentiel pour assurer la sécurité et le confort dans les environnements domestiques modernes .

On interpréte que cette évolution représente une étape importante vers la création de maisons plus sûres et plus intelligentes .

Chapter 5

Conclusion

La simulation du détecteur de gaz avec le logiciel Proteus présente plusieurs avantages .

• Tout d'abord , elle permet de tester le fonctionnement du détecteur dans un environnement virtuel , ce qui réduit les coûts et les risques associés à la manipulation de gaz réels .

• De plus , le logiciel Proteus offre la possibilité de simuler différents scénarios et conditions , permettant ainsi d'explorer diverses situations potentielles sans nécessiter de matériel physique supplémentaire .

• Finalement , la simulation à l'aide du logiciel Proteus permet une analyse approfondie des performances du détecteur , facilitant ainsi le processus de débogage et d'optimisation du système avant sa mise en œuvre réelle .

Cependant, il est possible d'améliorer ce projet en ajoutant un système d'alerte à distance pour une surveillance à distance , un enregistrement des données pour une analyse ultérieure ou en utilisant un capteur de gaz plus précis pour une détection plus fine des gaz présents dans l'air . Ces améliorations pourraient permettre une utilisation plus poussée de ce système dans des contextes professionnels ou industriels . Donc , ce projet a permis de réaliser un système de détection du gaz efficace et peu coûteuse grâce à l'utilisation de la carte Arduino et de capteurs peu coûteuse et facilement disponibles sur le marché .