

**Ministère de l'enseignement supérieur,
De la recherche scientifique et de la technologie
Université de Sousse**

**Institut Supérieur des sciences Appliquées et de Technologie
De Sousse (ISSATS)**



**Mini –projet : « protection humaine contre le gaz et
la flamme »**

**Réalisé par :-Djelassi Chaima
- Tekarri Zayneb**

Filière : LFEEA-A3-G1

Encadré par : Monsieur Chebbi Thameur

2020/2021

Table des matières

Remercîments	4
Introduction générale	5
Chapitre 1 : les différents composants du notre système embarque	6
Introduction :	7
Cahier de charges :	7
La Présentation de carte Arduino Uno :	7
Entrées/Sorties numériques :	8
Broches analogiques	8
Alimentation / Programmation par USB	8
Microcontrôleur principal	9
Régulateur de tension :	9
Oscillateur à quartz :	9
Interface SPI :	9
Transfert I2C :	9
Connecteur ICSP :	9
LEDs TX et RX :	10
Capteur MQ2 :	10
Description :	10
Principaux paramètres d'un capteur de gaz :	11
Sensibilité et leurs caractéristiques :	11
Capteur de flamme :	14
Description :	14
Principe de fonctionnement :	14
La composition du capteur :	14
Caractéristiques :	14
LED :	15
Le buzzer :	15
Ventilateur :	16
Servomoteur :	17
Présentation de module GSM :	17

Caractéristiques principales :	18
Description du module :	18
Conclusion :	19
Chapitre 2 : la partie software et hardware de notre système embarque	20
Introduction :.....	21
Partie software :	21
Initiation à logiciel arduino :	21
Partie hardware :	23
Initiation à Isis PROTEUS :	23
Conception de la carte électronique sur le logiciel KiCad :	30
Conclusion générale :.....	36
Références :.....	37

Remercîments

A mon cher encadreur :

M.Thameur Chebbi

Nous avons eu l'honneur et le privilège d'être parmi vos élèves et de bénéficier de vos riches enseignants.

Vos qualités pédagogiques et Humaines sont pour nous un modelé, votre gentillesse et votre disponibilité ont toujours suscitées notre admiration.

Introduction générale

La sécurité humaine joue un rôle important dans le monde d'aujourd'hui et il est nécessaire que des systèmes de sécurité efficaces soient installés dans des environnements domestiques et industriels et dans des établissements d'éducation. Durant ces dernières années, des chiffres alarmants ont été enregistré en Tunisie sur le phénomène des décès par asphyxie par des gaz dangereux et par brûlure par des flammes. Des centaines de citoyens trouvent la mort par intoxication au monoxyde de carbone (CO), ceci surtout à l'approche de chaque saison hivernale. Plusieurs raisons sont à l'origine de cette catastrophe, dont on peut citer : la mauvaise aération, la contrefaçon, les mauvaises installations et le non-respect des normes d'entretien. Dans le but de lutter contre ces risques, le développement de la microélectronique et de la micro-informatique dans les circuits modernes ; notamment les microcontrôleurs sera exploité dans ce projet, pour concevoir un système embarqué basé sur : la carte Arduino et de différents types de capteurs, la prévention et l'alerte en cas de fuite ou de présence d'un de ces gaz toxiques, tel que : le monoxyde de carbone CO et le méthane CH₄, en minimisant les sources du problème et en alertant l'utilisateur d'une façon anticipée pour que sa réaction soit à temps et avant que les dégâts se produisent

Chapitre 1 : les différents composants du notre système embarque

Introduction :

Dans notre vie quotidienne, on trouve toujours des accidents mortels comme l'étranglement du gaz ou le risque de mort à cause de brûlure.

Pour cela dans ce chapitre nous présentons les composants nécessaires pour la réalisation de notre mini-projet qui consiste en «un système embarqué pour la détection de gaz dangereux et la détection de flamme ».

Aussi nous présentons les organes nécessaires pour la prévention et la génération d'alarme .

Cahier de charges :

Point 1 : programmation du microcontrôleur Atmega328 a l'aide du logiciel Arduino .

Point 2 : Simulation du système sur le logiciel proteus 8 professionnel.

Point 3 : Conception de notre carte sur le logiciel KiCad.

La Présentation de carte Arduino Uno :

La carte Arduino Uno est le produit populaire parmi les cartes Arduino. Parfaite pour débiter la programmation Arduino, elle est constituée de tous les éléments de base pour construire des objets d'une 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. La description de toutes les connectiques de la carte Arduino UNO est présentée dans l'image ci-dessous :

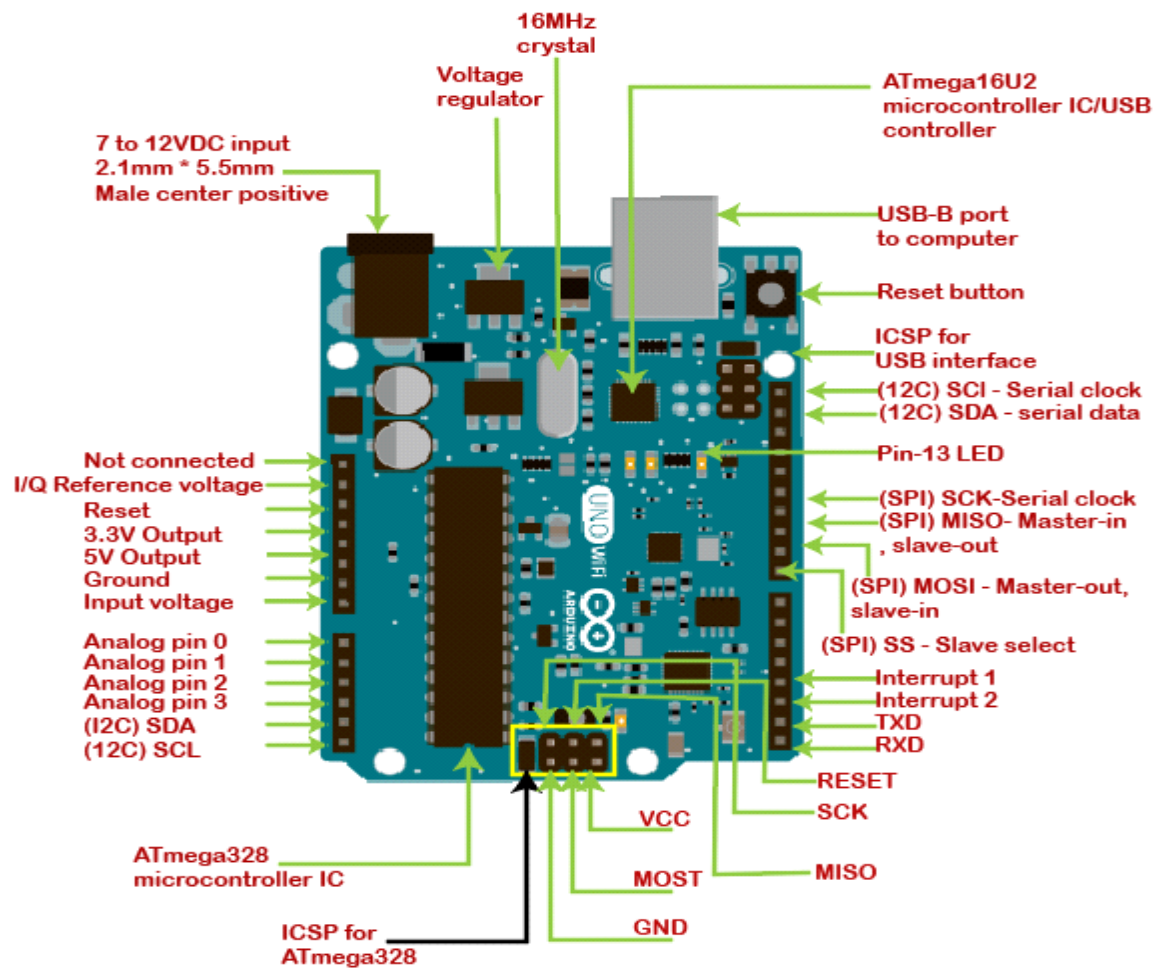


Figure 1 : schéma de l'arduino

Entrées/Sorties numériques :

La carte Arduino UNO possède **14 broches d'Entrées / Sorties numériques**, dont 6 peuvent fournir une sortie PWM (Pulse Width Modulation or pulse-duration modulation). Ces broches peuvent être configurées pour fonctionner comme des broches numériques d'entrée pour lire des valeurs logiques (0 ou 1) ou numériques. Elles peuvent également être utilisées comme des broches de sortie pour piloter différents modules comme des LEDs, des relais, etc.

Broches analogiques

L'Arduino UNO possède **5 broches d'entrées analogiques numérotée de A0 jusqu'à A5**. Ces broches permettent de lire un signal analogique d'un capteur comme un capteur de gaz ou de flamme.

Alimentation / Programmation par USB

La carte Arduino peut être alimentée avec un câble USB relié à votre ordinateur. Ou elle peut alimenter via le connecteur jack.

Microcontrôleur principal

Chaque carte Arduino possède son propre microcontrôleur . Nous pouvons le considérer comme le cerveau de la carte Arduino.

Régulateur de tension :

La fonction du régulateur de tension est de contrôler la tension d'alimentation de l'Arduino pour la stabiliser à la bonne tension du microcontrôleur et de chaque éléments de la carte. La tension de stabilisation est de 5 Volts sur les cartes UNO.

Oscillateur à quartz :

Un oscillateur à quartz est un élément électronique qui a la particularité de posséder un quartz à l'intérieur qui vibre sous l'effet piézoélectrique. Les propriétés électromécaniques du quartz sont telles qu'on arrive à faire vibrer le quartz à une fréquence très précise. Cet élément aide l'Arduino UNO à calculer les données de temps. Sur le dessus du composant, on peut lire 16.000H9H. Cela signifie que la fréquence est 16 MHz.

Interface SPI :

Une liaison SPI (pour *Serial Peripheral Interface*) est un bus de données séries synchrone baptisé ainsi par Motorola, au milieu des années 1980 qui opère en mode full-duplex. elle permet de dialoguer avec d'autres composants SPI (écrans, capteurs, etc...).



Figure2 : liaison SPI (un maitre et un esclave)

Transfert I2C :

Un bus I²C (pour *Inter Integrated Circuit*) est unbus série et synchrone composé de trois fils conduisant :

- le signal de données (SDA) ;
- le signal d'horloge (SCL) ;
- la référence (masse).

Connecteur ICSP :

Avant tout, le connecteur ICSP (*In-Circuit Serial Programming*) est une connectique AVR comprenant les broches MOSI, MISO, SCK, RESET, VCC et GND. Il s'agit d'un connecteur de programmation. Ce connecteur permet entre autre de programmer directement le microcontrôleur sur les couches les plus basses (bootloader...). Le bootloader est un composant logiciel ayant deux fonctions principales :

- chargement d'une application dans la mémoire FLASH du microcontrôleur à partir de la ligne série

- lancement de l'application

LEDs TX et RX :

Dans notre carte, nous trouverons deux indicateurs : **TX (émission)** et **RX (réception)**. Ils apparaissent à deux endroits sur la carte Arduino UNO. Tout d'abord, sur les broches numériques 0 et 1, pour indiquer les broches responsables de la communication série. Deuxièmement, les LEDs TX et RX. Le voyant TX clignote à une vitesse variable lors de l'envoi des données série. La vitesse de clignotement dépend de la vitesse de transmission utilisée par la carte. RX clignote pendant le processus de réception. La vitesse de transmission s'exprime en bauds, soit l'équivalent du bits/seconde si le signal est binaire.

Les Caractéristiques techniques :

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)
Pins 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) 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

Tab 1 :les Caractéristiques d'arduino UNO

Capteur MQ2 :

Description :

Les capteurs MQ sont des capteurs spéciaux conçus pour être sensibles à la détection de certains gaz comme le GPL, le CO, le propane, etc. En chinois, «sensible» signifie «Mǐngǎn» et «Gaz à» signifie «Qǐ lai».

Le capteur de gaz MQ2 est un semi-conducteur capteur de gaz/fumée qui détecte la présence du gaz ou fumée à des concentrations de 300 ppm à 10000 ppm. Il peut détecter différents gaz comme l'hydrogène H₂, le GPL (liquefied petroleum gas), le méthane CH₄, le butane, 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.

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.

Sensibilité et leurs caractéristiques :

La sensibilité est un paramètre qui exprime la variation de la réponse du capteur en fonction de la variation du mesurande (concentration de gaz). Un capteur de gaz est dit sensible si une petite variation de concentration entraîne une importante variation du signal de sortie. La définition générale de la sensibilité est donc :

$$S_i = \frac{\Delta R}{\Delta [C]_i}$$

Avec S_i , la sensibilité au gaz i

R , la réponse du capteur (la sortie du capteur, résistance ou conductance, ...)

$[C]_i$, la concentration du gaz i .

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).

Symbole	Nom du paramètre	Paramètre technique	Remarques
Rs	Résistance à la détection	3K Ω -30K Ω (1000 ppm d'isobutane)	Détection de la concentration
α (3000ppm/1000ppm CH ₄)	Concentration Taux de pente	≤ 0.6	portée: 200ppm-5000ppm GPL et propane
Condition de détection standard	Temp: 20 °C \pm 2 °C Vc: 5V \pm 0,1 Humidité: 65% \pm 5% Vh: 5V \pm 0,1		300ppm-5000ppm butane
Temps de préchauffage	Plus de 24 heures		5000ppm-20000ppm méthane 300 ppm-5000 ppm H ₂ 100 ppm-2000 ppm De l'alcool

Tab 2 : Caractéristique de sensibilité

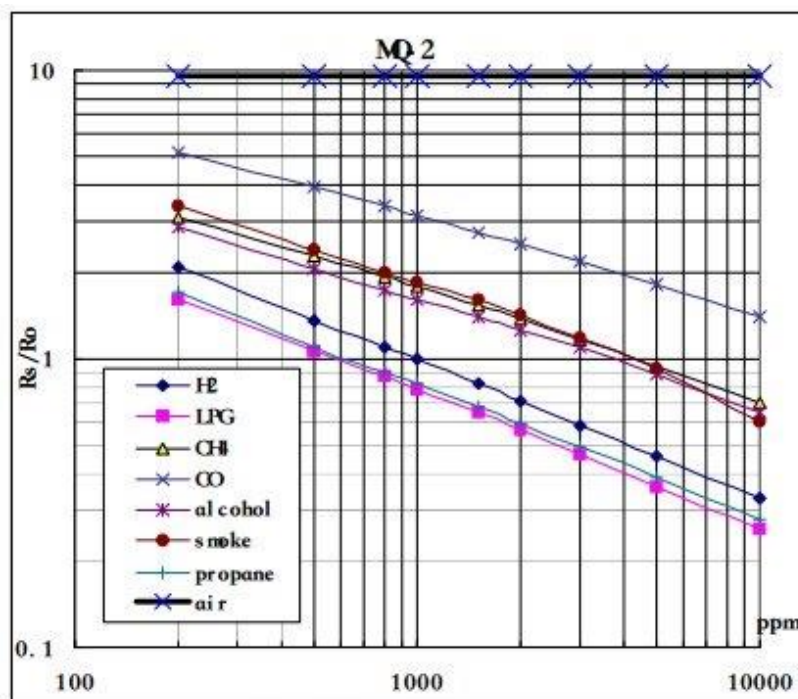


Fig.2 sensitivity characteristics of the MQ-2

Fig.3 is shows the typical sensitivity characteristics of the MQ-2 for several gases. in their: Temp: 20°C、Humidity: 65%、O₂ concentration 21% RL=5k Ω
Ro: sensor resistance at 1000ppm of H₂ in the clean air.
Rs:sensor resistance at various concentrations of gases.

Figure3:les caractéristiques de sensibilité typiques du MQ-2 pour plusieurs gaz.

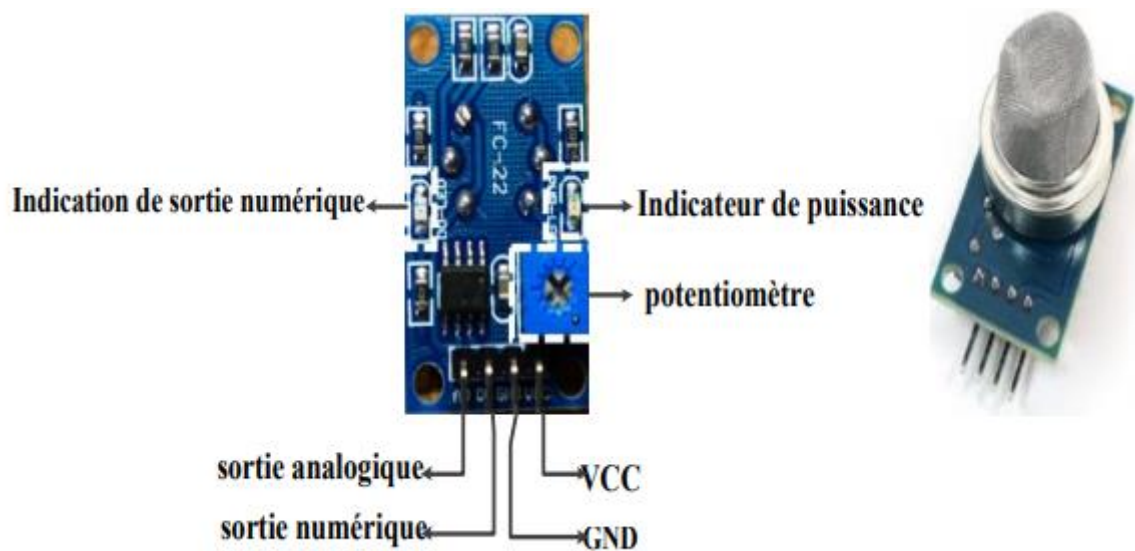


Figure4 : Circuit typique d'un capteur de gaz

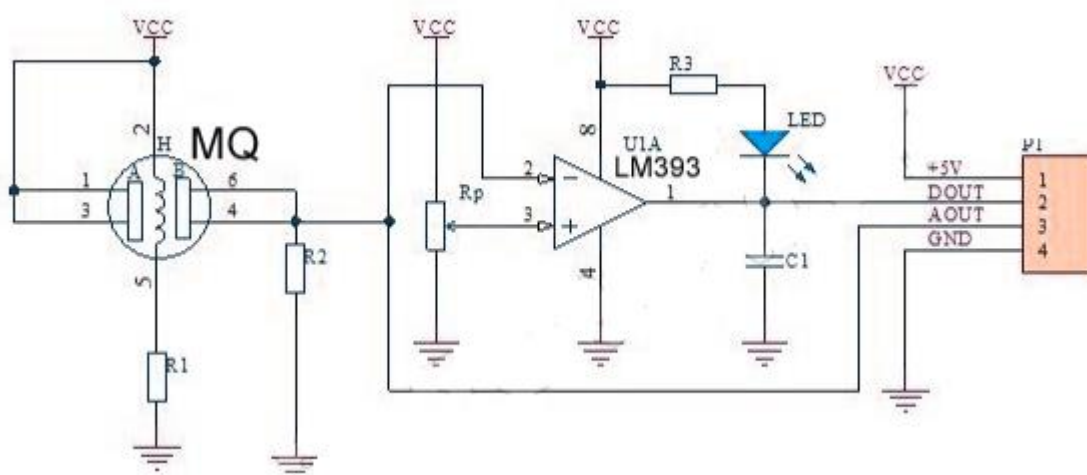


Figure5 : Schéma électrique des capteurs MQx utilisés

Capteur de flamme :

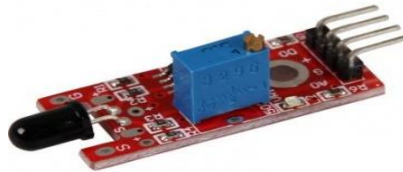


Figure6 :capteur de flamme

Description :

Le détecteur de flamme est conçu pour la détection de feux ouverts. Il est sensible à la lumière émise par les flammes durant la combustion. Il est capable de faire la distinction entre les flammes et les autres sources lumineuses, en étant sensible uniquement aux longueurs d'ondes optiques et fréquences de pulsation spécifiques de flamme. Ceci permet au détecteur d'éviter les fausses alarmes engendrées par des facteurs tels que les fluctuations de la lumière solaire.

Principe de fonctionnement :

Les détecteurs de flammes utilisent des technologies optiques. Il est établi que les flammes émettent des rayonnements électromagnétiques dans les longueurs d'ondes infrarouges (IR), la lumière visible et les longueurs d'ondes ultraviolettes (UV), selon la source du combustible. C'est pourquoi des technologies optiques ont été mises au point pour détecter les flammes en utilisant le spectre lumineux allant des infrarouges aux ultraviolets. Les détecteurs s'appuient sur la portée optique du rayonnement émis dans les bandes spectrales pour déterminer s'il s'agit véritablement d'une flamme.

La composition du capteur :

Ce module est composé de trois éléments fonctionnels. Le capteur situé à l'avant du module effectue la mesure, le signal analogique est ensuite envoyé sur l'amplificateur. Celui-ci amplifie le signal en fonction du gain déterminé par le potentiomètre et envoie le signal à la sortie analogique du module.

Il convient de noter que le signal est inversé: plus la valeur mesurée par le capteur est haute, plus la tension de sortie est faible.

La troisième partie est composée d'un comparateur qui commute la sortie numérique et la diode lorsque le signal tombe en dessous d'une certaine valeur.

Caractéristiques :

- Extrêmement sensible aux longueurs d'ondes entre 760-1100nm
- Seuil de détection de flamme modifiable par un potentiomètre

- Plage d'angle de détection: environ 60 degrés
- Alimentation: 3.3-5.5 VDC

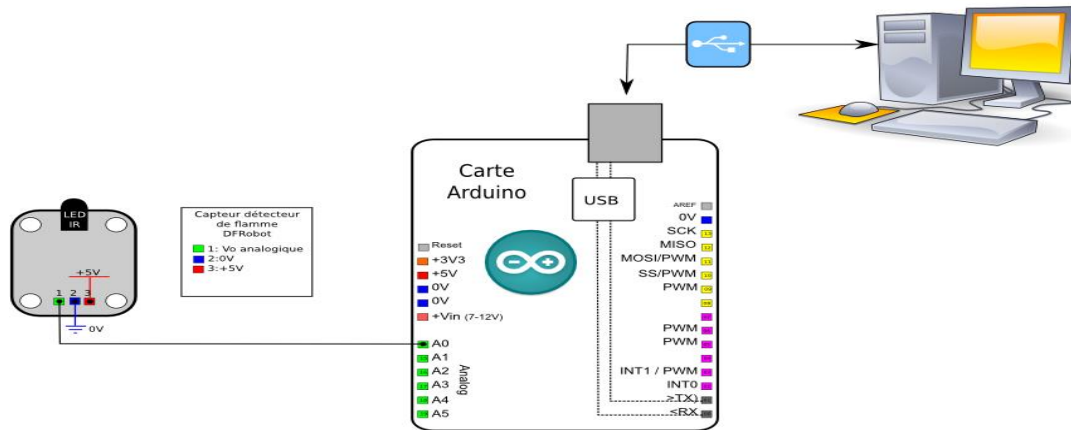


Figure7 : Schéma de câblage d'un capteur de flamme avec Arduino

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 une LED jaune s'allume et le LED vert s'éteint pour signaler qu'il y a un gaz. Dans le cas où il y a un feu le LED vert s'éteint et les LEDs rouge et jaune s'allument.

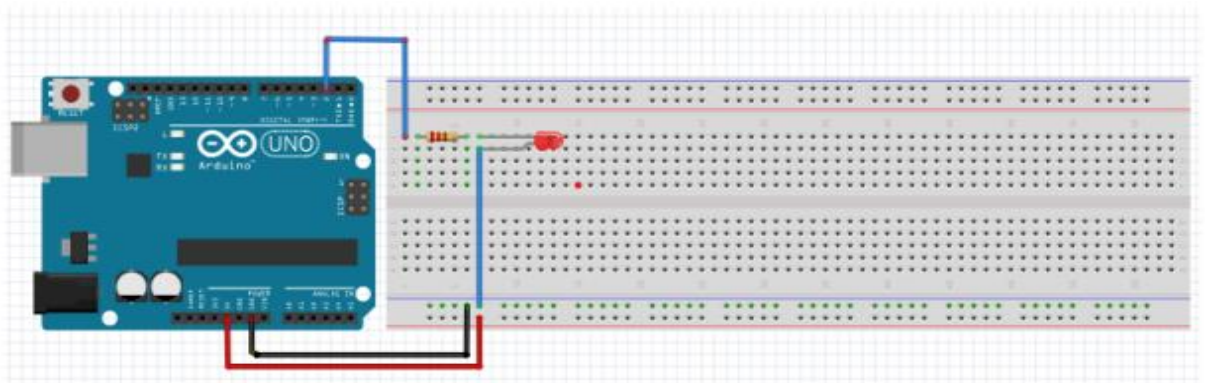


Figure8 :Schéma de câblage d'un LED avec Arduino

Le buzzer :

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.

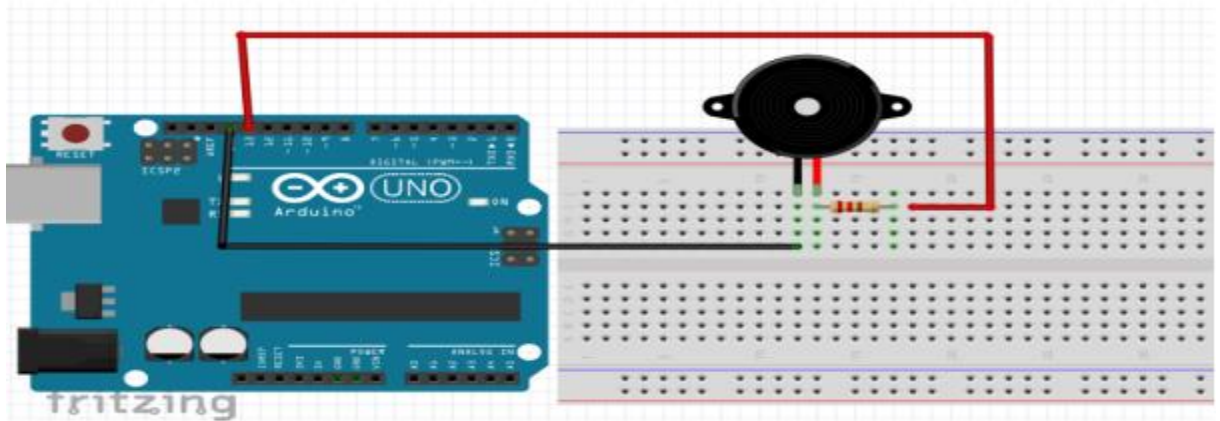


Figure9 : Schéma de câblage buzzer avec Arduino

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.



Figure10 : ventilateur

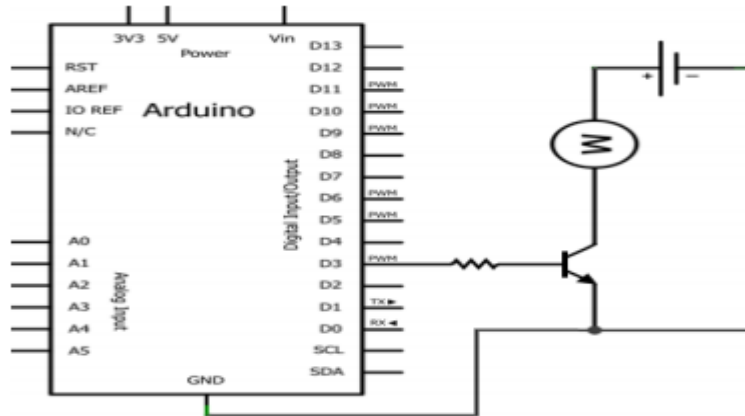


Figure11 : schéma de branchement du ventilateur avec Arduino UNO

Servomoteur :

Les servomoteurs sont des moteurs un peu particuliers, qui peuvent tourner avec une liberté d'environ 180° et garder de manière relativement précise l'angle de rotation que l'on souhaite obtenir.

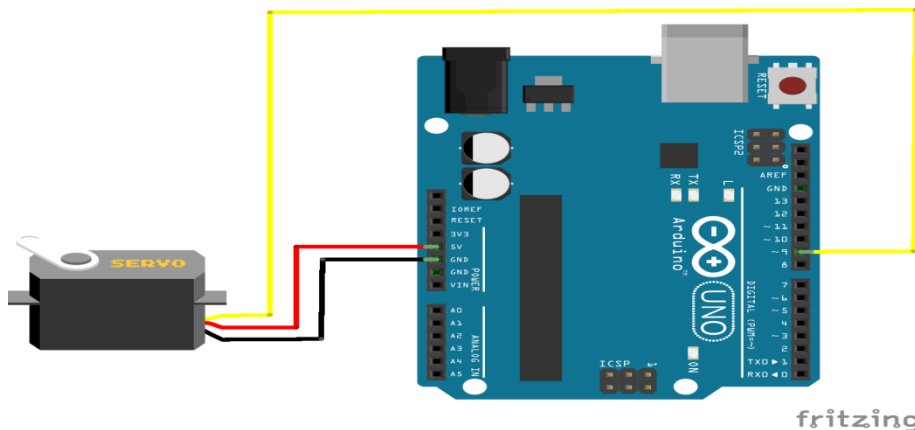


Figure13 : Schéma de câblage servomoteur avec Arduino

Présentation de module GSM :

Le module GSM/GPRS est une carte d'interface compatible Arduino. Elle permet d'envoyer et recevoir des SMS, des données ou des communications vocales depuis le réseau mobile. Le module est basé sur le circuit SIM900. Il est contrôlé via les commandes AT depuis une carte Arduino. Le module est livré avec une antenne patch déportée. Un connecteur au dos de la platine est prévu pour recevoir une carte SIM ainsi qu'une pile Lihtium CR1220. La communication entre le module et une carte Arduino est réalisée par la liaison série asynchrone : UART ou une liaison série logicielle.

Caractéristiques principales :

Module Quad-band	850/900/1800/1900 MHz
Protocoles supportés	TCP/UDP
Tension d'alimentation	5 V par la broche 5V 6,5 V à 12 V par la broche Vin
Consommation	1,5 mA en veille 400 mA max
Puissance	Classe 4 (bandes 850/ 900 MHz) : 2 W Classe 1 (bandes 1800/ 1900 MHz) : 1 W
Température de fonctionnement	- 40 °C à + 85 °C
Dimensions	68.58 x 53.34mm

Tableau 3 :caracteristique du module GSM

Description du module :

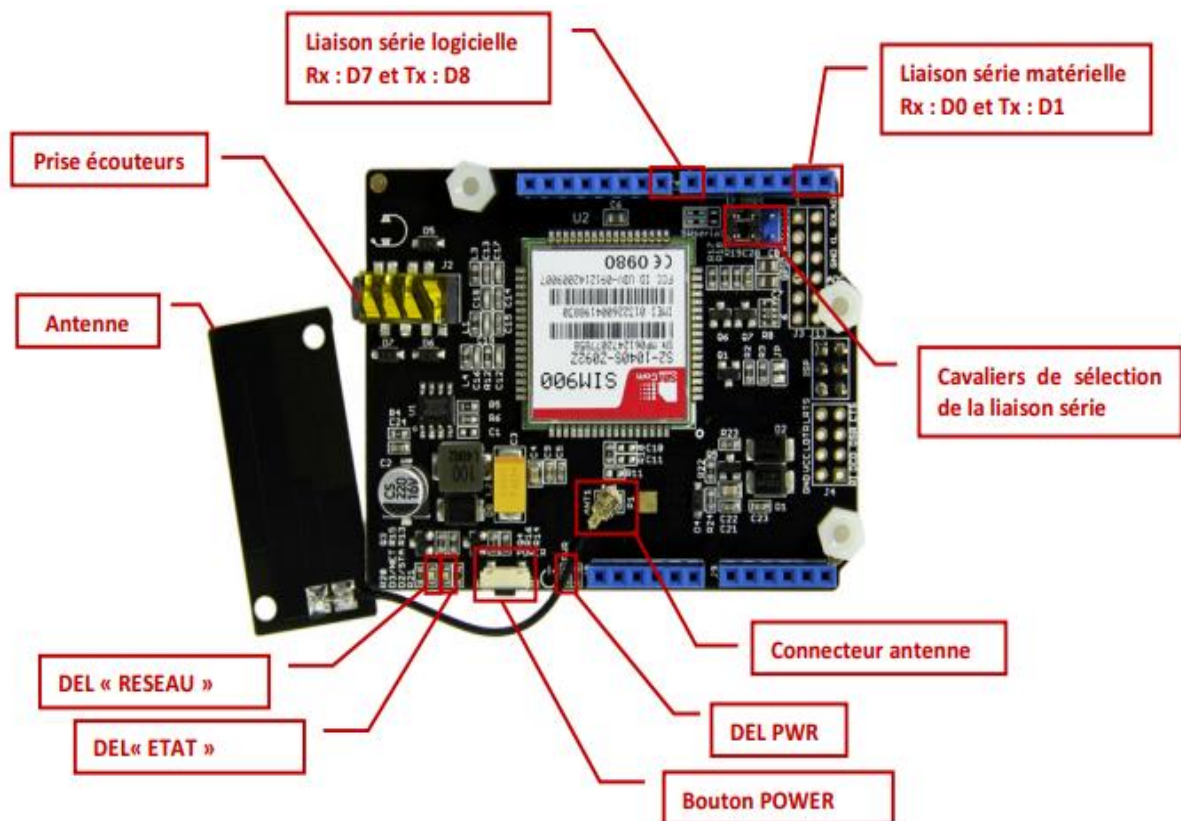


Figure12 :Circuit typique d'un module GSM (SIM900)

Le statut de la carte est affiché à l'aide de 3 LEDs de couleurs différentes. Ses indications sont :

DEL « PWR » (verte)	Eteinte	Le shield est hors tension
	Allumée	Le shield est sous tension
DEL « ETAT » (Rouge)	Eteinte	Le module SIM900 est hors tension
	Allumée	Le module SIM900 est sous tension
DEL « RESEAU » (Verte)	Eteinte	Le module SIM900 en veille
	Allumée : 64 ms Eteinte : 800 ms	Le module ne trouve pas de réseau
	Allumée : 64 ms Eteinte : 3000 ms	Le module a trouvé un réseau
	Allumée : 64 ms Eteinte : 300 ms	Communication GPRS

Tableau4 :le fonctionnement des LEDS

Conclusion :

Dans ce chapitre, nous avons présenté le principe de fonctionnement et les caractéristiques de la carte Arduino. Aussi nous avons présenté le capteur de gaz MQ2 et le capteur de flamme et les différents composants utilisés.

Chapitre 2 : la partie software et hardware de notre système embarque

Introduction :

Dans ce chapitre, nous présentons l'application globale de notre mini projet ainsi la simulation du l'application sur le logiciel proteus 8 professionnel (ISIS) et la conception de notre carte électronique sur le logiciel KiCad.

Le système proposé se compose de trois parties principales, nommément un circuit de contrôle basé sur Arduino UNO, un circuit de détection de gaz, un circuit de détection de flamme et un circuit des sorties englobant : un module de prévention et d'activation alarme

Partie software :

Initiation à logiciel arduino :

Le **logiciel Arduino** est un environnement de développement (IDE) open source et gratuit, téléchargeable sur le site officiel **Arduino**. L'IDE **Arduino** permet : d'éditer un programme : **des croquis** (sketch en Anglais), de communiquer avec la carte **Arduino** grâce au terminal.



Figure 14 : le logiciel arduino

Le « Programme Principal » : où on configure le microcontrôleur pour les différentes entrées analogiques et numériques, ainsi que les sorties digitales à utiliser.

```

#include <SoftwareSerial.h>
#include <Servo.h> //cette bibliotheque servo est nécessaire pour contrôler le servomoteur.
Servo monServomoteur; // Créer un objet servo pour nous permettre de contrôler le servomoteur.
SoftwareSerial gsm (9,10); // rx ,tx
const unsigned long eventInterval = 6000;
unsigned long previousTime = 0; //Type de données: unsigned long . c'est le nombre de millisecondes écoulées depuis le démarrage du programme.
int flag = 0 ;
void setup() {
  monServomoteur.attach(7);
  pinMode(4,OUTPUT); // buzzer
  pinMode(8,OUTPUT); // LED rouge
  pinMode(11,OUTPUT); //LED vert
  pinMode(12,OUTPUT); //LED jaune
  pinMode(5,OUTPUT); //ventilateur
  pinMode(A0,INPUT); // capteur de gaz
  pinMode(A1,INPUT); // capteur de flamme
  Serial.begin (9600); // définir le taux de date pour la publication de logiciels
  gsm.begin(9600);
}
void loop() {
  if (analogRead(A0)<400 && analogRead(A1)<400){
    flag =0 ;
    Serial.print("no risk\r"); //affichage des messages
    digitalWrite(11,HIGH); // la led verte est allumée
    digitalWrite(8,LOW); // la led rouge est éteinte
    digitalWrite(12,LOW); // led jaune est éteinte
    digitalWrite(4,LOW); // buzzer désactivé
    digitalWrite(5,LOW); //ventilateur désactivé
    for (int position = 90; position >= -90; position--) {
      monServomoteur.write(position); } // Fait bouger le bras de 180° à 0°
    delay(1000);
  }
  while (analogRead(A0)>400 && analogRead(A1)<400 && flag == 0){
    flag = 1 ;
    digitalWrite(11,LOW); //la led verte est éteinte
    digitalWrite(8,LOW); //la led rouge est éteinte
    digitalWrite(12,HIGH); // la led jaune est allumée
    digitalWrite(8,LOW); //la led rouge est éteinte
    digitalWrite(4,HIGH); // buzzer actif
    digitalWrite(5,HIGH); //ventilateur actif
    for (int position = -180; position <= 180; position++) {
      monServomoteur.write(position); } // Fait bouger le bras de 0° à 180°
    delay(100);
    Serial.println("AT"); // affichage de commande AT
    gsm.println ("AT");
    delay(100);
    gsm.print("AT+CMGF = 1\r"); // Mode Texte
    delay (100);
    gsm.println("AT+CMGS = \"+21623171934\""); // Numéro du consommateur
    delay (100);
    gsm.print("fuite de gaz detecte"); // SMS Texte
    delay(100);
    Serial.print((char) 26);
    gsm.write((char) 26); // CTRL-Z
    delay (100);
  }
}

```

```

gsm.println();
Serial.println();
break ; }

unsigned long currentTime = millis(); //La fonction millis renvoie le nombre de millisecondes pendant lesquelles notre carte Arduino a été mise sous tension.
if ( currentTime - previousTime >= eventInterval && analogRead(A0)>400 && analogRead(A1)<400 ) {
    previousTime = currentTime; // Mettre à jour le timing pour la prochaine fois
    flag = 0 ;}
else if ( currentTime - previousTime >= eventInterval && analogRead(A0)>400 && analogRead(A1)>400 )
    (previousTime = currentTime;// Mettre à jour le timing pour la prochaine fois
    flag =0; }
while (analogRead(A1)>400 && analogRead(A0)>400 && flag == 0 ){
    flag = 1;
    digitalWrite(8,HIGH);//la led rouge est allumée
    digitalWrite(12,HIGH);//la led jaune est allumée
    digitalWrite(11,LOW);//la led verte est éteinte
    digitalWrite(4,HIGH);// buzzer actif
    digitalWrite(5,LOW);//ventilateur actif
    for (int position = -180; position <= 180; position++) {
        monServomoteur.write(position);} // Fait bouger le bras de 0° à 180°
    Serial.println("AT");
    gsm.println ("AT");
    delay(100);
    gsm.println("AT+CMGF = 1"); // Mode Texte
    delay (100);
    gsm.println("AT+CMGS = \"+21650863290\""); // le numero du protection civile
    delay (100);
    gsm.println (" il y a un feu danger dans la rue de kalaa soghra"); // text SMS

    Serial.print((char) 26);
    gsm.print((char) 26); // CTRL-Z
    delay (100);
    gsm.println();
    Serial.println();
    delay(100);
    break ;}}

```

Figure15 :le code du système

Partie hardware :

Initiation à Isis PROTEUS :

Isis Proteus est un logiciel de développement et de simulation d'application via un environnement graphique simple et interactif.

Le lancement de PROTEUS donne un environnement classique de type Windows, constitué d'une fenêtre principale, et d'un ensemble de barres d'outils. Outre le menu classique permettant la gestion des fichiers, de l'affichage, et des options des projets, la fenêtre principale comprend une Zone de travail destinée au développement des circuits à simuler et à tester. Une Bibliothèque d'objets affiche la liste des objets (circuits électriques, électroniques, ...) utilisés dans l'application en cours. Les différentes Touches magnétoscope constituées des raccourcis permettant le lancement de la simulation, ainsi que la mise en pause, l'exécution pas à pas, et l'arrêt de la simulation.

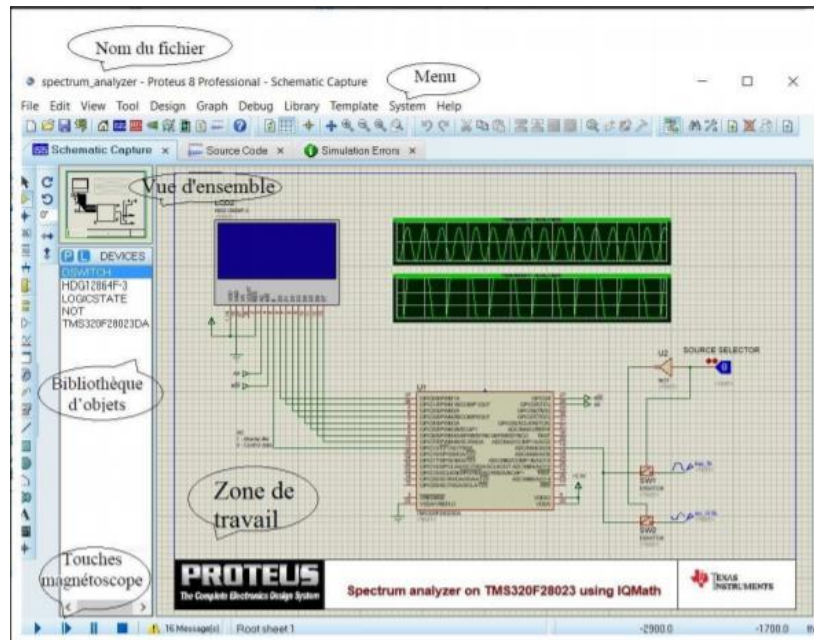


Figure16 : le logiciel proteus 8 professionnel

La boîte verticale de boutons, comprend les principaux raccourcis nécessaires au développement rapide d'applications.

- Component mode (Mode Composant), Un clic sur ce bouton puis une bibliothèque d'objets permet l'ajout de différents composants.
- Generator Mode (Mode Générateur) permettant l'accès aux différents types de générateurs.
- Instruments un raccourci permettant l'ajout des appareils de mesure tels que le voltmètre, l'ampèremètre, et l'oscilloscope.
- Terminal permettant d'ajouter des points particuliers dans un schéma tels que les entrées/sorties ou Gnd.

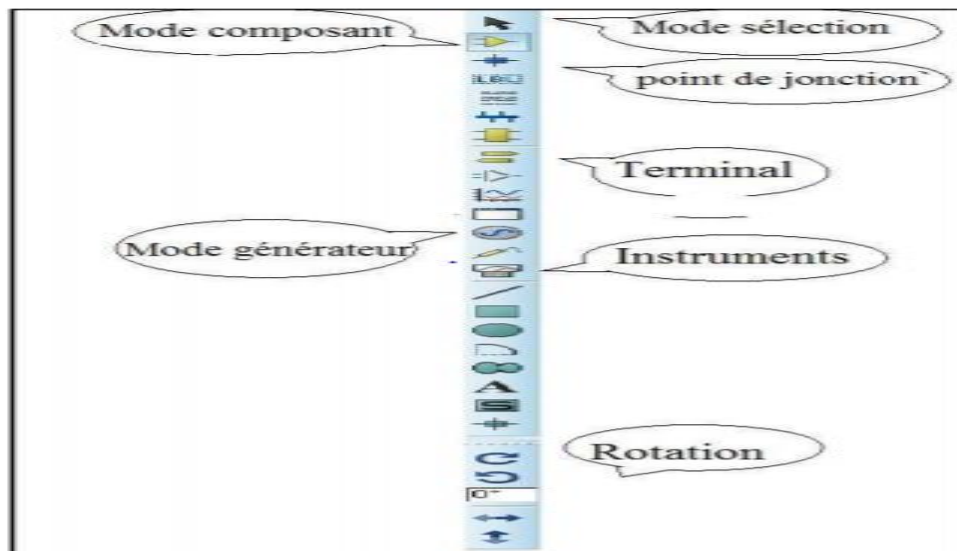


Figure17 : des icones du logiciel proteus 8 professionnel

La réalisation de l'application sur Proteus 8 (isis) :

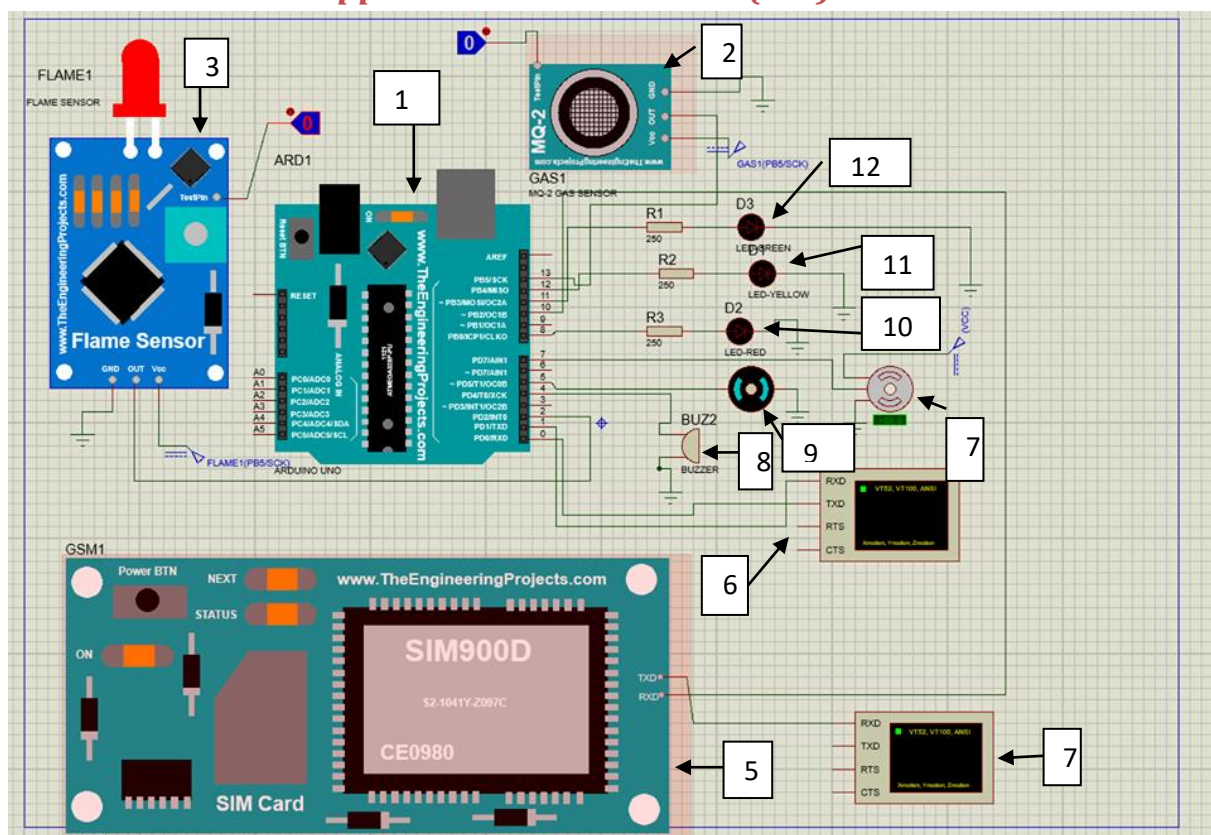


Figure 18 : Schéma du circuit de simulation (Isis-Proteus)

1: carte arduino Uno c'est la carte électronique utilisé pour notre programmation.

2 : 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'envoi de ces informations au microcontrôleur.

3 : Le capteur de flamme qui est conçu pour interagir avec un microcontrôleur et surveiller en permanence le feu dans l'espace et l'envoi de ces informations au microcontrôleur.

4 : Le module GSM est une carte d'interface qui permet d'envoyer et recevoir des SMS, des données ou des communications vocales depuis le réseau mobile.

5 : virtual terminal qui affiche les messages envoyés par le module GSM.

6 : virtual terminal qui affiche la commande envoyée par l'Arduino.

7 : servomoteur pour réduire la fuite du gaz dangereux et de prévenir l'augmentation de sa concentration.

8 : un buzzer est utilisé pour générer une alarme sonore.

9 : moteur qui on a mis au lieu d'un ventilateur. La rotation du moteur signifie que le ventilateur tourne et ça pour diminuer l'effet de la concentration d'un gaz dangereux suite à une fuite.

10 : un diode LED rouge s'il s'allume signifie qu'il y a une flamme.

11: un diode LED jaune s'il s'allume signifie qu'il y a une fuite de gaz.

12: un diode LED vert s'allume lorsque les gaz dangereux ne sont pas détectés dans l'environnement.

Les états de notre application :

L'application de notre mini-projet compose de 3 parties principales :

1^{er} cas : la détection d'une fuite de gaz :

Lorsque la concentration de gaz « **propane et butane** » (le potentiomètre1) dépasse le seuil toléré (400), le microcontrôleur alerte automatiquement le consommateur sur la présence des fuites en envoyant un SMS d'alerte « fuite de gaz détecté » via GSM aux numéros stockés, active la LED jaune et le Buzzer (alarme sonore), les LEDs vert et rouge s'éteignent ,ferme la vanne principale d'alimentation en gaz , actionne le ventilateur d'aération ,le Virtual terminal 1 va afficher la commande « AT » ,et le Virtual terminal 2 va afficher le message envoyé par le GSM.

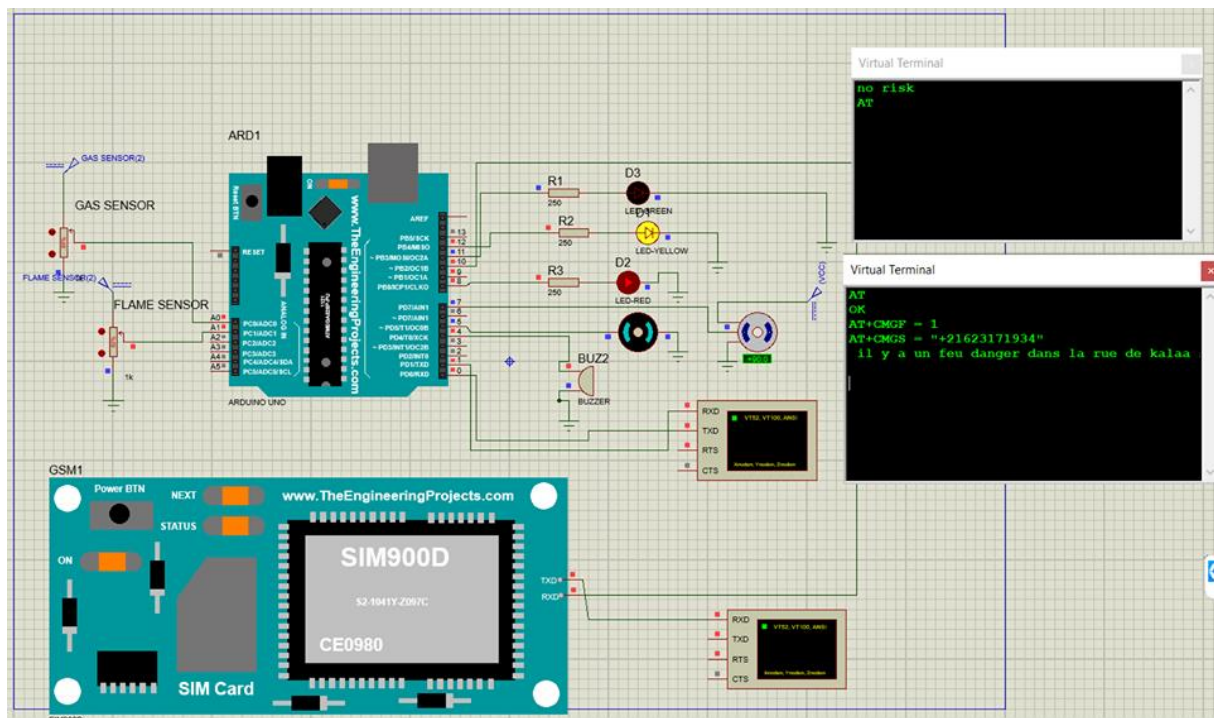


Figure19: Schéma du circuit de simulation dans le cas qu'il y a fuite de gaz détecté »

Grace à la fonction `millis ()` on peut répéter le message jusqu'à la concentration du gaz devendra inferieur le seuil désire (400).

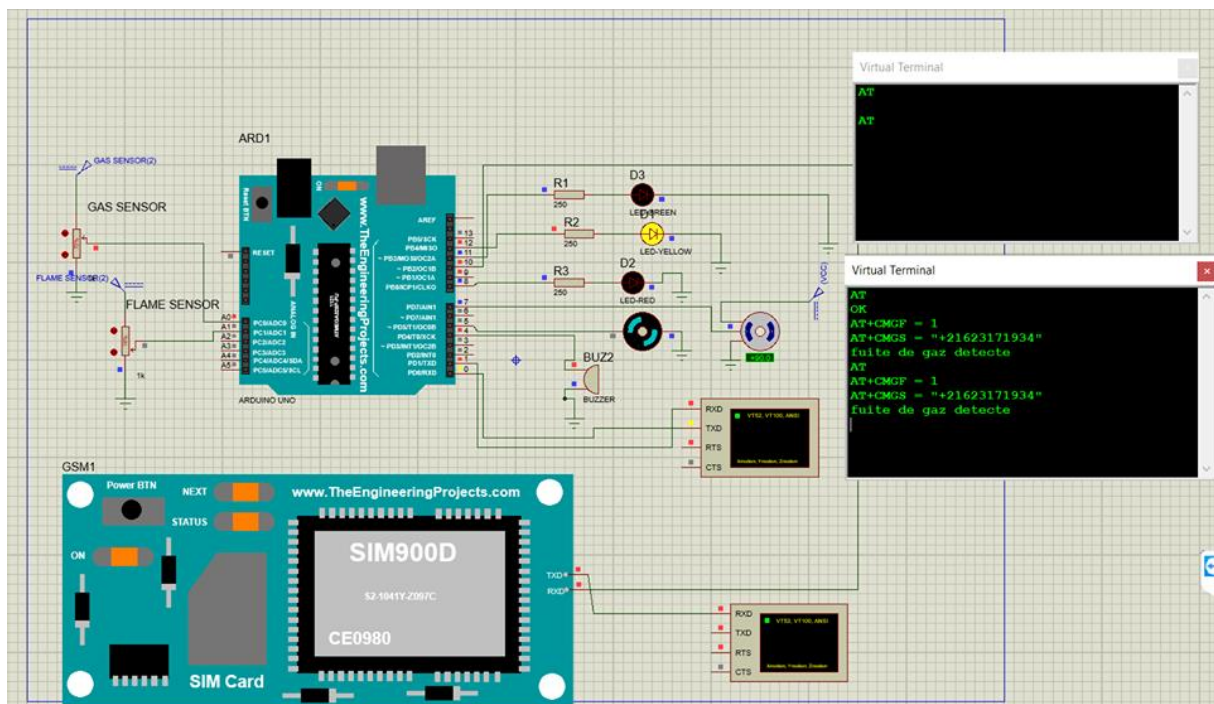


Figure 20: Schéma du circuit de simulation dans le cas ou la fuite de gaz est encore existe

2eme cas : le détection d'une flamme :

Lorsque la concentration de gaz « **smoke** » (le potentiomètre1) dépasse le seuil toléré (300) et le capteur de flamme (le potentiomètre2) détecte la présence de feu donc le microcontrôleur alerte automatiquement la protection civile sur la présence des flammes en envoyant un SMS d'alerte « Il y a un feu danger dans la rue de kalaa soghra » via GSM aux numéros stockés, active la LED jaune , la LED rouge et le Buzzer (alarme sonore), la LED vert S'éteint ,ferme la vanne principale d'alimentation en gaz ,le Virtual terminal 1 va afficher la commande « AT »,et le Virtual terminal 2 va afficher le message envoyé par le GSM.

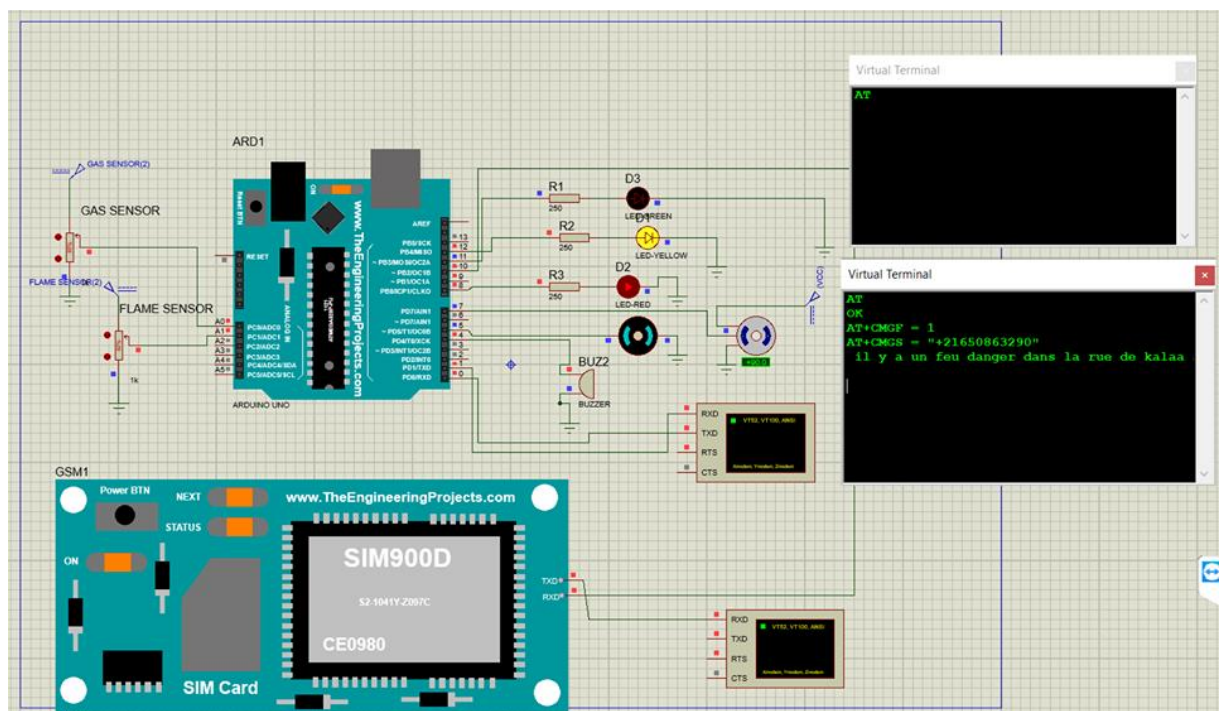


Figure 21 : Schéma du circuit de simulation dans le cas qu'il y a un feu danger

Grace à la fonction **millis ()** on peut répéter le message jusqu'à la concentration du gaz deviendra inferieur le seuil désire (300) et ne détecte pas la flamme.

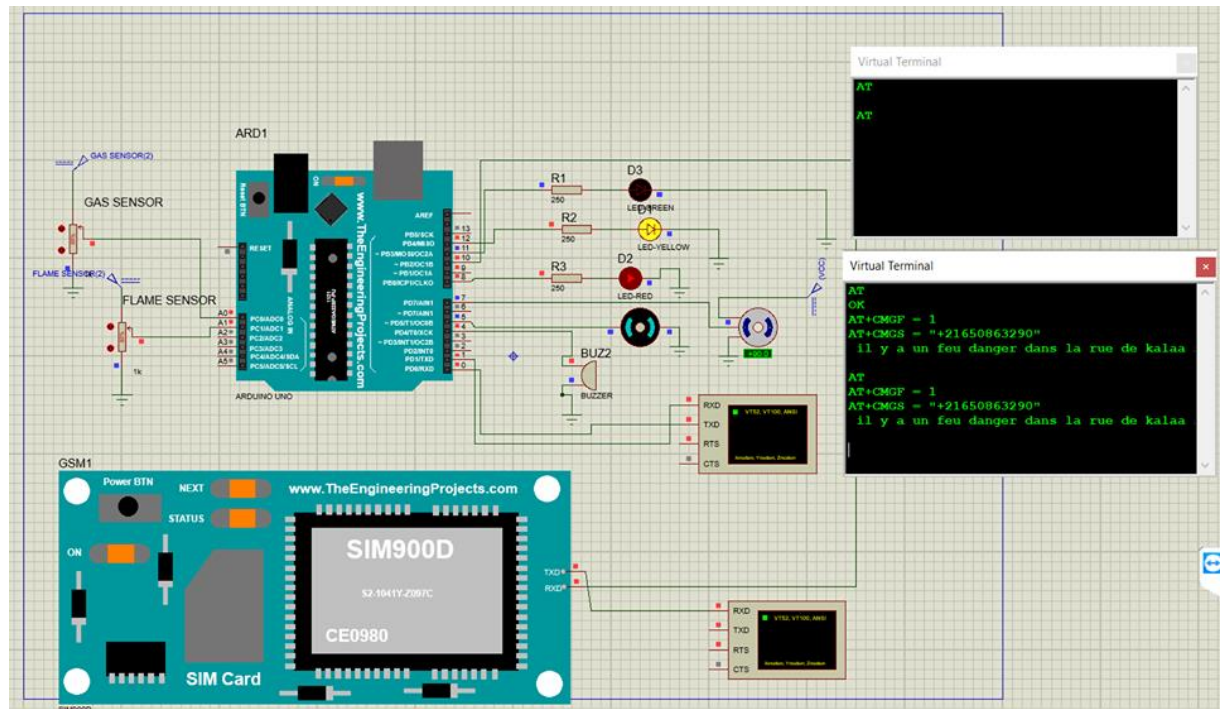


Figure 22 : Schéma du circuit de simulation dans le cas ou le feu danger est encore existe

3eme cas : pas d'incident :

Dans ce cas , la concentration de gaz (le potentiomètre1) ne dépasse pas le seuil toléré (300) et le capteur de flamme (le potentiomètre2) ne détecte pas la présence de feu,active la LED vert, les LEDs rouge et jaune s'éteignent ,arrêté le Buzzer (alarme sonore),la vanne principale d'alimentation en gaz est ouvert ,le Virtual terminal 1 va afficher le message « no risk».

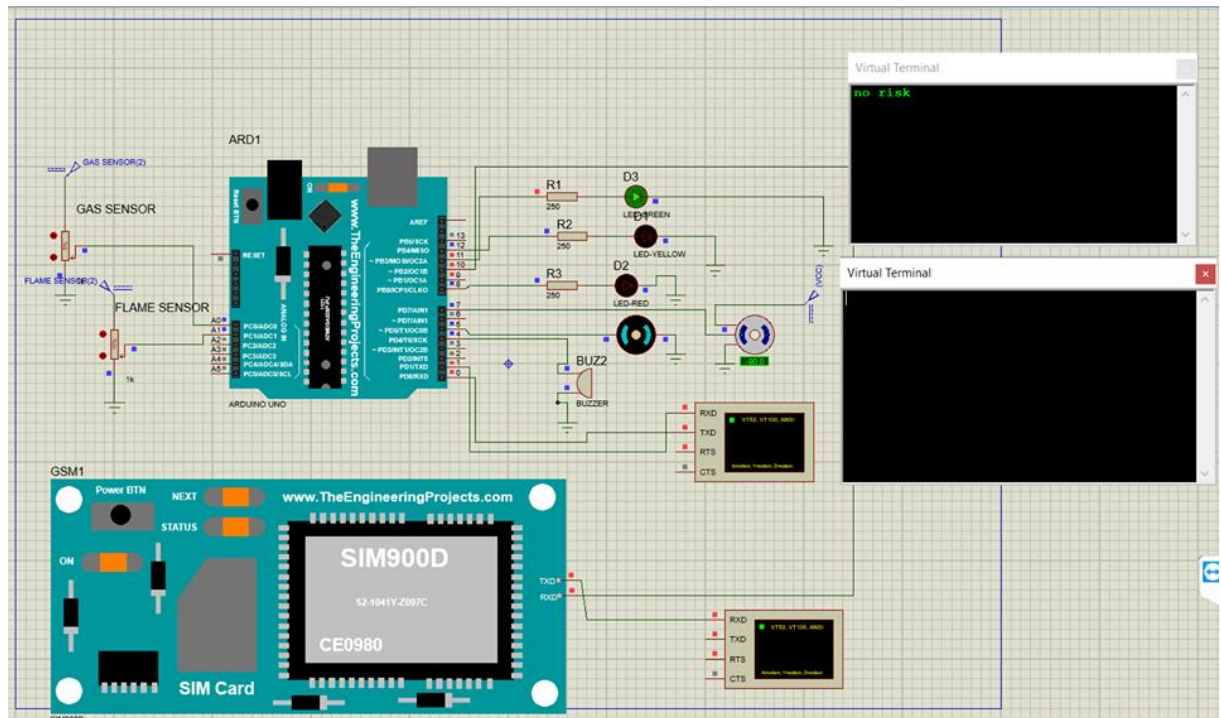


Figure 23 : Schéma du circuit de simulation dans le cas pas d'incident

Conception de la carte électronique sur le logiciel KiCad :

La conception d'un circuit imprimé est généralement de mise lorsqu'on veut relier des composants électroniques entre eux. De cette façon, on est en mesure de réaliser un circuit électronique complexe. Le circuit imprimé est présent dans pratiquement tous les appareils électroniques. Et il est assez facile de le concevoir à condition d'avoir les connaissances requises.

Initiation à KiCad:

KiCad est une suite logicielle libre de conception pour l'électronique pour le dessin de schémas électroniques et la conception de circuits imprimés.

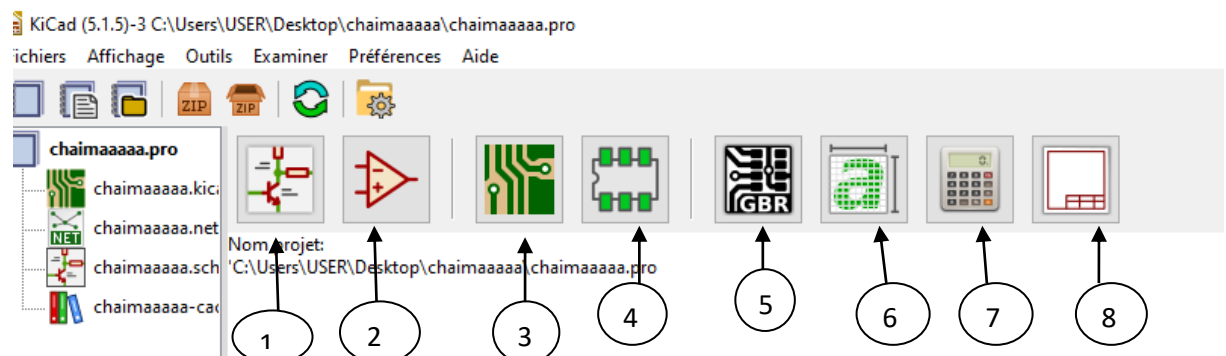


Figure24 : le logiciel KiCad

KiCad est composé des logiciels suivants :

1:Eeschema : l'éditeur de schémas ;

2 : Editeur de symbole

3: Pcbnew : l'éditeur de circuits imprimés

4 :Cvpcb : l'utilitaire de sélection des empreintes physiques des composants utilisés dans le schéma ;

5:Gerbview : le visualiser de fichier Gerber qu' il contient la description des diverses couches de connexions électriques (les pistes, les pastilles, les vias...).

6 : Bitmap2component : convertisseur d'images en symbole schématiques ou empreintes.

7 :Pcbcalculator : outil d'aide à la conception qui permet de calculer des valeurs résistances, des largeurs de pistes..

8 : Editeur de description de page. Editer les graphiques et texte du cartouche

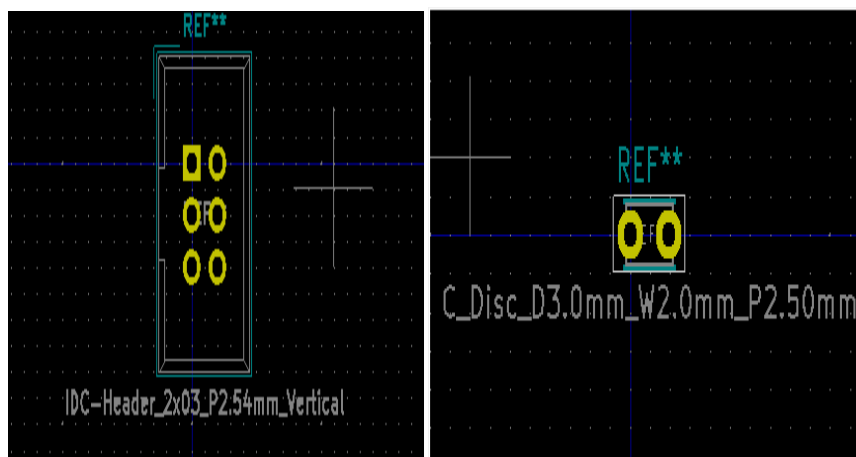
Dessin du schématique :

Dans notre schéma, on utilise le microcontrôleur atmega328 , USB type B , une processeur Avr-isp-6 , des connecteur femelle et male de quelle façon chaque composant a son connecteur désire. Des connecteur femelle pour le capteur de gaz , capteur de flamme et le module GSM .Des connecteurs male pour le ventilateur et le servomoteur.

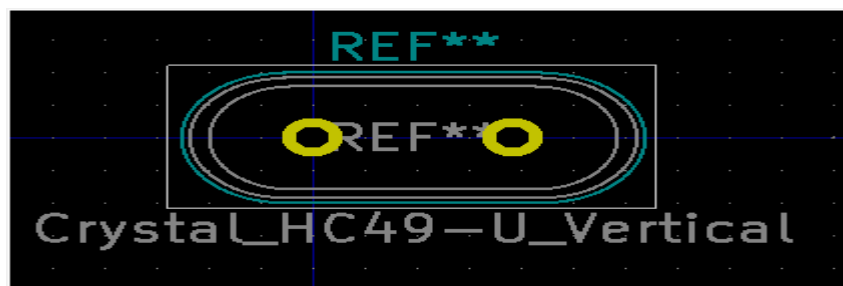
1	C3 -	C : Capacitor_THT:C_Disc_D3.0mm_W2.0mm_P2.50mm
2	C4 -	C : Capacitor_THT:C_Disc_D3.0mm_W2.0mm_P2.50mm
3	J1 - Conn_01x02_Male :	Connector_JST:JST_EH_B2B-EH-A_1x02_P2.50mm_Vertical
4	J2 - Conn_01x04_Female :	Connector_JST:JST_EH_B4B-EH-A_1x04_P2.50mm_Vertical
5	J3 - Conn_01x02_Male :	Connector_JST:JST_EH_B2B-EH-A_1x02_P2.50mm_Vertical
6	J4 - Conn_01x03_Male :	Connector_JST:JST_EH_B3B-EH-A_1x03_P2.50mm_Vertical
7	J5 - Conn_01x02_Male :	Connector_JST:JST_EH_B2B-EH-A_1x02_P2.50mm_Vertical
8	J6 - Conn_01x04_Female :	Connector_JST:JST_EH_B4B-EH-A_1x04_P2.50mm_Vertical
9	J7 -	USB_B : Connector_USB:USB_B_OST_USB-B1HSxx_Horizontal
10	J8 -	AVR-ISP-6 : Connector_IDC:IDC-Header_2x03_P2.54mm_Vertical
11	J9 - Conn_01x04_Male :	Connector_TE-Connectivity:TE_MATE-N-LOK_350211-1_1x04_P5.08mm_Vertical
12	R1 -	R : Resistor_THT:R_Axial_DIN0204_L3.6mm_D1.6mm_P1.90mm_Vertical
13	U1 -	ATmega328-AU : Package_QFP:TQFP-32_7x7mm_P0.8mm
14	Y1 -	Crystal : Crystal:Crystal_HC49-U_Vertical

Figure26 : le Cvppcb de notre carte électronique

Des exemples des composants dans lepcbnew :



AVR-isp-6condensateur disc



Oscillateur à quartz

Conception de la carte finale :

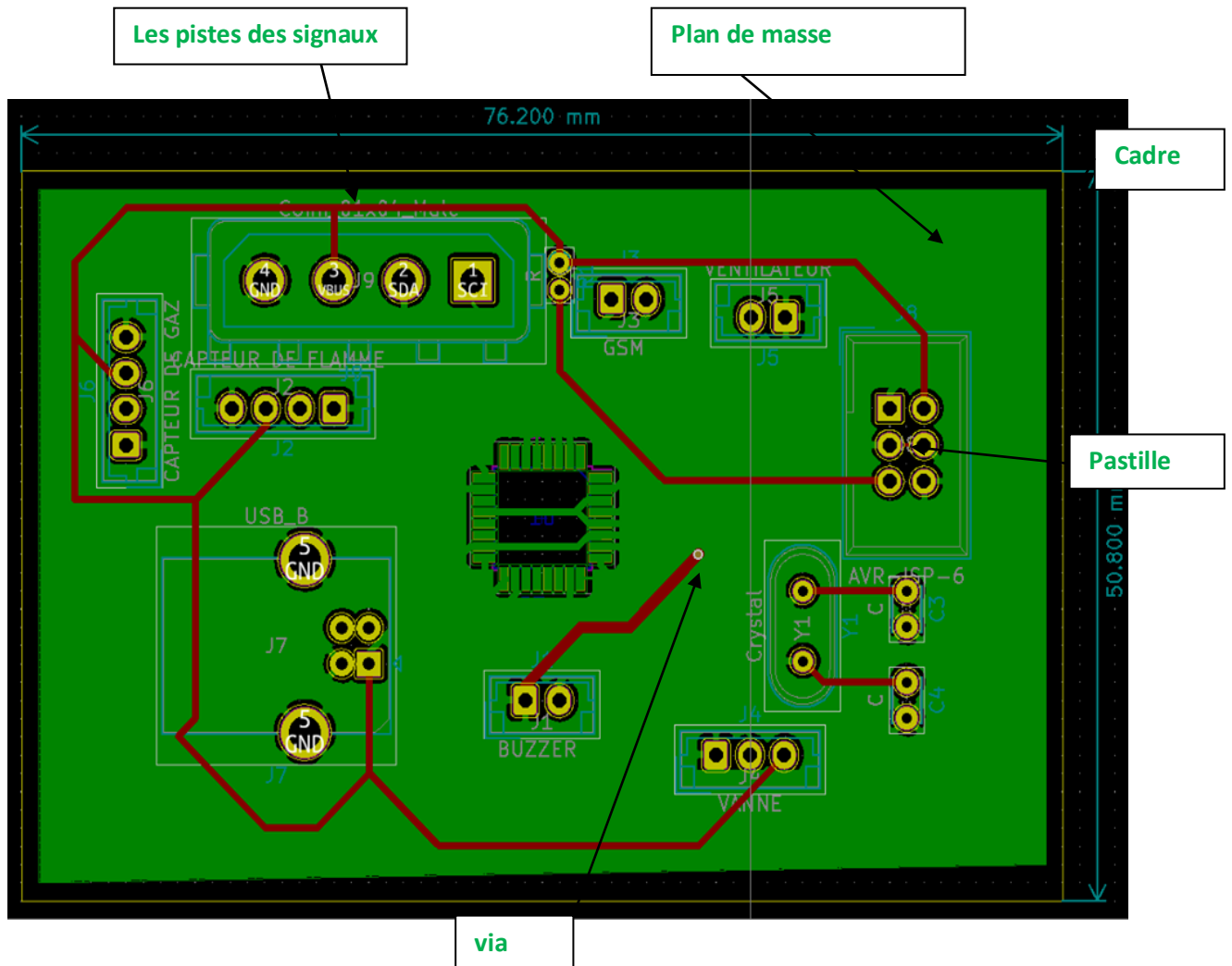


Figure27 :le Pcbnew de notre carte électronique

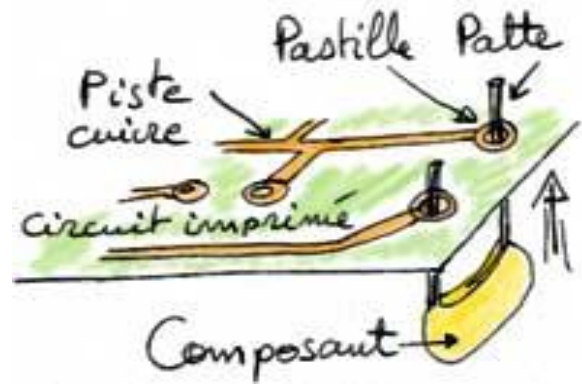
Les pistes relient électriquement différentes zones de circuit imprimée

Les pastilles une fois perforées, établissent une liaison électrique, soit entrés les composants saudes à travers le circuit imprimé, soit entre les différentes couches de cuivre. Dans certain cas des pastilles non perforées servent à souder des composants montés en surface.

Un via est un trou métallisé qui permet d'établissement une liaison électronique entre 2 couches.

Plan de masse: fait relier toutes les pistes d'un circuit électronique entre elles pour ne former qu'une seule nappe de cuivre.

Exemples d'un composant:



Portrait de la carte en 3D :

il est possible de visualiser notre carte électronique en 3 dimensions

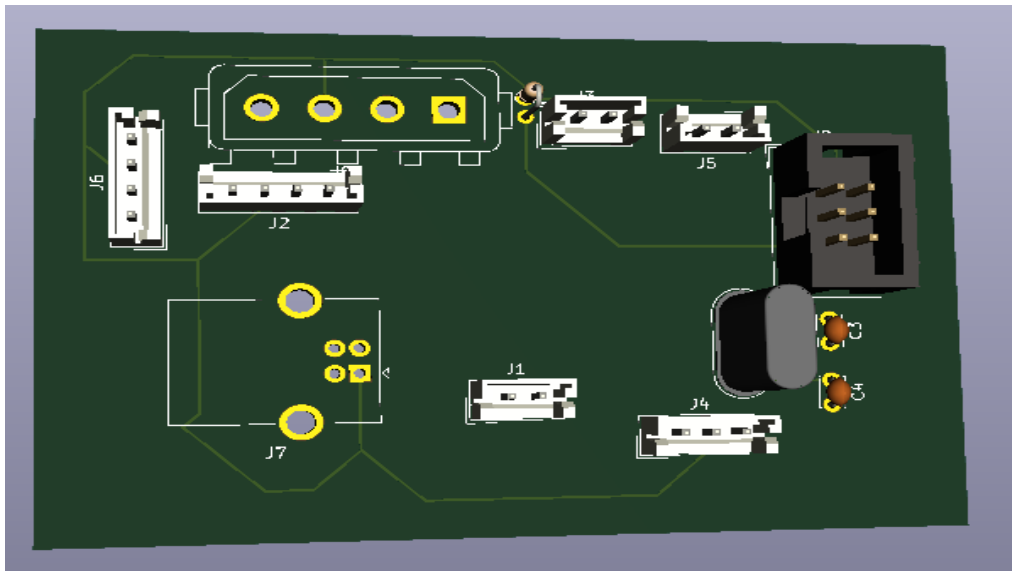


Figure28 : visualisation du carte électronique en 3D

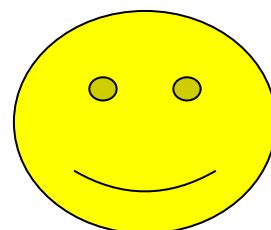
La carte électronique est un circuit formé de composants et de conducteurs qu'on intègre à une structure mécanique. Elle est présente dans presque tous les appareils électroniques couramment utilisés. Et la conception de circuit imprimé se fait en suivant des étapes bien déterminées

Conclusion générale :

L'utilisation du gaz comme source d'énergie est essentielle dans notre vie, il y a des risques pour cette utilisation sur la vie humaine en cas de fuites ou en cas de devienra un feu danger . le monoxyde de carbone reste la première cause d'accidents domestiques mortel en Tunisie .

Dans ce contexte, nous avons proposé un système pour la détection de gaz toxiques (monoxyde de carbone) et la detection de flamme , ainsi que pour la prévention et l'alerte.Ce système peut être aussi très important dans les environnements domestiques et industriels, et cela peut contribuer à sauver la vie humaine et conserver les biens.

Ce projet nous a permis de faire le lien entre l'étude théorique d'un montage électronique et sa réalisation pratique avec un aspect pluridisciplinaire qui englobe l'électronique embarqué et l'informatique industrielle.



Références :

<https://bentek.fr/2-arduino-uno/>

https://www.google.com/search?q=sensibilite+de+capteur+mq2&sxsrf=ALeKk035YVlI0cSJq71tfF6IJ9kBh0qbtw:1591884673391&source=lnms&tbm=isch&sa=X&ved=2ahUKEwiGg-zD-PnpAhVIXRoKHVLQAd0Q_AUoAXoECAwQAw&biw=1366&bih=576#imgsrc=YuJgQ8UyS-GNcM

<http://bib.univ-oeb.dz:8080/jspui/bitstream/123456789/8108/1/memoire-pdf.pdf>

<https://www.emerson.com/fr-fr/automation/measurement-instrumentation/flame-gas-detection/about-flame-detection>

<http://www.ardwtech.com/shop/development-accessoire/69-capteur-de-flamme-arduino.html>

http://gce.ovh/wiki/index.php?title=Le_d%C3%A9tecteur_de_flamme

<https://www.matooma.com/fr/definitions/gsm>

<https://fr.hobbytronics.co.uk/mq2-gas-smoke-sensor>

https://en.wikipedia.org/wiki/Liquefied_petroleum_gas

<https://projetsdiy.fr/utiliser-detecteur-de-gaz-fumees-mq2/>

https://www.univ-usto.dz/images/coursenligne/Introduction_Isis_Proteus.pdf

https://fr.wikipedia.org/wiki/User_Datagram_Protocol

https://fr.wikipedia.org/wiki/Transmission_Control_Protocol

https://lewebpedagogique.com/isneiffel/files/2017/06/Module-GSM_GPRS_Seed-Studio.pdf

<http://www.haoyuelectronics.com/Attachment/MQ-2/MQ-2.pdf>

<https://www.commentcamarche.net/contents/1131-transmission-de-donnees-les-modes-de-transmission>