



République Algérienne Démocratique et Populaire
Ministère de l'enseignement Supérieur
et de la Recherche scientifique



Université Echahid Hamma Lakhdar d'El-Oued
Faculté de Technologie

Mémoire de Fin d'Étude
En vue de l'obtention du diplôme de
MASTER ACADEMIQUE

Domaine : Sciences et Technologie
Filière : Télécommunications
Spécialité : Systèmes de Télécommunication

Thème

**Réalisation d'un système d'alarme intelligent à base
d'un smart phone contre l'intoxication due au gaz
CO dans la maison**

Réalisé par :

- CHAMI Ahmed Chaouki
- MOUSSAOUI Mohammed Ali
- NID Mohammed Bachir

Soutenu en juin 2019 devant le jury composé de :

Dr. HETTIRI Messaoud	MCA	Président
Dr. AJGOU Riadh	MCA	Examineur
Dr. TIR Zoheir	MCA	Examineur
Dr. GHENDIR Said	MCA	Directeur du mémoire

Année Universitaire : 2018/2019

Remerciements

Avant tout développement sur cette expérience, il apparaît opportun de commencer ce projet par des remerciements pour Allah d'abord et à ceux qui nous ont beaucoup appris au cours de ce travail.

C'est avec plaisir que nous réservons ces quelques lignes en signe de gratitude et de profonde reconnaissance à l'égard de tous ceux qui nous ont aidés à la réalisation de notre projet de fin d'étude.

Nous adressons nos meilleurs remerciements à notre encadreur Dr Ghendir Said qui nous a tendu la main pour accomplir notre projet et nous lui souhaitons beaucoup de succès et bonheur dans sa vie.

Nous remercions chaleureusement Mr Moussaoui Abderahmane pour leur aide de réaliser notre application portable.

N'oublions pas l'administration de direction de commerce surtout le chef service de protection de consommateur Said Hayat qui fut assez visionnaire pour nous offrir tous les moyens de réussite.

Nous remercions chaleureusement les membres du jury pour l'honneur qu'ils nous ont fait en acceptant d'évaluer notre travail.

Nos salutations à tous les enseignants de la Faculté De Technologie Département De Génie électrique et spécialement les enseignants de la spécialité télécommunication, sans oublier les étudiants de master télécommunication.

Nous tenons à exprimer mes reconnaissances à tous mes amis et collègues pour le soutien moral et matériel.

Et enfin, nous remercions également tous ceux qui ont participé de près ou de loin à l'élaboration de ce projet.

إهداء

إلهي لا يطيب الليل إلا بشكرك ولا يطيب النهار إلا بطاعتك، لك الحمد حتى ترضى ولك الحمد إذا رضيت ولك الحمد بعد الرضى

إلى من تجرّع الكأس فارغا ليسقيني قطرة حب ، إلى من كلّت أنامله ليُقدم لنا لحظة سعادة ، إلى من حصد الأشواك عن دربي ليُمهد لي طريق العلم ، إلى حكمتي وعلمي ، إلى أدبي وحلمي ، إلى طريقي المستقيم .. **أبي الغالي**

إلى من أَرْضَعْتَنِي الحب والحنان ، إلى رمز الحب وبلسم الشفاء ، إلى القلب الناصع بالبياض ، إلى من كافحت وجاهدت لسعادتي وهنائي ، إلى من كان دعاؤها سر نجاحي .. **أمي الغالية حواء الله**

أنتما قدوتي في الحياة ، فلا شيء أعظم من امرأة عظيمة قامت بتربيتي وأب أفخر دائما عندما يختتم إسمي بإسمه ، أرجوا من الله أن يمد عمركما لثريا ثمارا قد حان قطافها بعد طول انتظار
إلى أُمي الثانية ، إلى التي لم تبخل عليّ بشيء من أجل دفعي الى طريق النجاح ، إلى التي علمتني كيف أرتقي سلم الحياة ، إلى قدوتي و سندي .. **إلى جدتي دة هادي خديجة**
إلى القلوب الطاهرة الرقيقة والنفوس البريئة، إلى رياحين حياتي..

لأخوتي هشام حمزة ليلي خولة ومحمد الرحيم

إلى من أظهروا لي أجمل ما في الحياة، إلى من كانوا ملاذي وملجئي في أصعب اللحظات، إلى من عشت معهم أجمل الذكريات، إلى من ساقهم الله لي، إلى أخوتي الذين لم تلدهم أُمي..

إلى أصدقائي محمد، البشير، الجيلاني، يوسف، محمد الكريم ومحمد الخالق.

إلى من جاهد معنا لإتمام مذكرتنا في أحسن حال، إلى من تكرم بوقته الثمين وصبر علينا، إلى من نورنا بعلمه وتوجيهاته .. **إلى مؤطري الدكتور محمد بن سعيد**

إلى من علمونا حروفا من ذهب وكلمات من درر وعبارات من أسمى العبارات في العلم، إلى من صاغوا لنا علمهم حروفا وفكرهم منارة تنير لنا مسيرة العلم والنجاح ... **إلى أساتذتنا الكرام**

موساوي محمد علي

الملخص:

كثرت في الآونة الأخيرة الحوادث المنزلية ومن بين هاته الحوادث المميتة التسممات بغاز اوكسيد الكربون والمنبعث جراء الاحتراق الغير الكامل من المسخّنات التي تعمل بالغاز، في الجزائر حوالي 200 قتيل كل سنة جراء الاختناق بهذا الغاز لذا قمنا بدراسة وانجاز نظام للكشف عن وجود غاز اول اكسيد الكربون لمنع وقوع مثل هاته التسممات بالموازات مع قياس الحرارة والرطوبة، وذلك استنادا بالمتحكم الدقيق اردوينو واجهزة استشعار عن بعد بواسطة شبكة الهاتف النقال.

عندما يكتشف النظام ان تركيز غاز اول اوكسيد الكربون تجاوز عتبة معينة يتم اعلان المسؤول والمساعدين الذين كان قد اُضاف أرقامهم وهم 3 من اختياره عن طريق رسالة الى الهاتف المحمول, اضافة الى أنه يمكن للنظام إتخاذ اجراءات وقائية تلقائيا في حالة الخطر من خلال تشغيل مروحة لتقليل تركيز الغاز او اقفال الصمام الكهربائي لمدخل الغاز. للتحكم في هذا النظام صممنا تطبيق سننشره في متجر قوقل لتسهيل الوصول اليه من طرف المستخدمين.

الكلمات المفتاحية : غاز اول اكسيد الكربون, اردوينو, اندرويد, تسمم غاز, منزل ذكي

Résumé :

L'intoxication par le monoxyde de carbone, provoquée par la combustion incomplète de systèmes de chauffage à gaz, a fait l'objet de nombreux accidents domestiques. Ces dernières années en Algérie, environ 200 personnes se sont mortes chaque année par l'intoxication de ce gaz.

Nous avons donc étudié et mis au point un système de détection de monoxyde de carbone afin de prévenir de tomber dans ce type d'accidents, où on a basé sur le microcontrôleur Arduino et les détecteurs tout en exploitant le réseau mobile GSM.

Lorsque le système découvre que la concentration de monoxyde de carbone a dépassé un certain seuil, le responsable sera averti par message au téléphone mobile d'administrateur ainsi que trois autres aides qui ont été enregistrés suivant de son choix. Le système prend automatiquement des mesures préventives en cas de danger en faisant actionner un ventilateur pour réduire la concentration de gaz, ou déclencher l'électrovanne de l'accès de gaz.

Pour contrôler ce système, nous avons développé une application qui sera disponible dans le play-store pour la laisser à la portée de tous les utilisateurs.

Mots clés : Monoxyde de carbone, Arduino, Android, intoxication gaz, domotique

Abstract:

Carbon monoxide poisoning, caused by the incomplete combustion of gas heating systems has been the subject of many domestic accidents. Recently, in Algeria, about 200 people are dying each year by the poisoning of this gas.

So, we studied and developed a carbon monoxide detection system to prevent such type of accidents, where we focused on the Arduino microcontroller and the detectors while exploiting the GSM mobile network.

When the system detects that the carbon monoxide concentration has exceeded a certain threshold, the administrator mobile phone will be notified by message, as well as three other registered users of his choice. The system takes precautionary measures automatically in case of danger by operating a fan to reduce the gas concentration or triggering the solenoid valve of the gas access.

To control this system, we have developed an application that will be available in the play-store to be within the reach of all users.

Key words : Carbon Monoxide , Arduino, Android, gas poisoning, smart home

Sommaire

Remerciement

Dédicaces

Résumé

Sommaire

Listes des figures

Listes des tableaux

Liste des abréviations

Introduction générale

Chapitre I : OUTILS ET DEVELOPPEMENT

I.1 Introduction	1
I.2 Cadre du projet	1
I .3 Cahier de charge propose	1
I .3.1 Le microcontrôleur Arduino	2
I .3.1.1 Entrées/Sorties	2
I .3.1.2 La communication serie avec Arduino	3
I.3.2 Les capteurs	4
I.3.2.1 Captur de gaz CO	4
Caractéristiques MQ7	5
Fonctionnement de capteur MQ7	6
Module de capteur MQ7	6
Schéma equivalent	7
Calibrer le module MQ7	7
Brochage du module MQ7	8
I.3.2.2 Module DHT11	8
Brochage du module DHT 11	9
Mode de fonctionnement DHT 11	10
Signal de départ	10
Signal de réponse	11
Données 40 BITS	11
Lire la température et l'humidité	12

I.3.3. Module GSM	12
Le SIM 900 de SIMCOM	13
Caractéristiques générales	13
Caractéristiques des données	14
Le module GSM SIM900A Shields Arduino	14
Les commandes AT	15
La norme GSM07.07	16
I.3.4 Afficheur LCD 16*2 I2C	18
I.3.4. Le Buzzer	18
I.3.5. Relay	19
I .4. Logiciels utilisés	19
I .4.1 l'environnement de la programmation (IDE Arduino)	19
Structure générale du programme	20
I .4.2 Android studio	20
Comparaison entre l'Eclipse et l'Android Studio	22
Les outils utilisent	22
Conclusion	23

Chapitre II : ARCHITECTURE ET PROGRAMME

II.1 Introduction	24
II.2 Description générale	24
II.3 Le mode de fonctionnement de système	25
II.4 Les types de programmation	25
II.5 Organigrammes	26
II.5.1 Organigrammes Arduino	26
II.5.2 Organigramme principale	26
II.5.3 Organigramme sous-programme config	28
II.5.5 Organigramme sous-programme state	28
II.5.5 Organigramme sous-programme action	29
II.5.6 Organigramme sous-programme alarme	29
II.5.7 Organigramme sous-programme Clear	30
II.5.8 Organigramme application Android	30
II.6 Lecture SMS reçu à partir Arduino	31

II.7 Envoi SMS à partir Arduino vers portable	32
Conclusion	33

Chapitre III : CONCEPTION ET REALISATION

III.1 Introduction	34
III.2 Partie hardware	34
III.2.1 La réalisation électrique	34
III.2.1.1 Alimentation	35
III.2.1.2 Les capteurs	36
Le PPM (partie par million)	36
III.2.1.3 Afficheur LCD	37
III.2.1.4 Commande de relais	37
III.2.1.5 Commande de siren	38
III.2.1.6 Télécommande GSM	38
Schéma électrique globale	39
III.2.2 Travaux expérimentaux	40
III.2.2.1 Moniteur serie de Arduino ide	40
III.3 Partie software	41
Conclusion	45

Conclusion générale

Annexe

Références

Listes des Figures

Figure 1 : structure générale.	1
Figure 2 : microcontrôleur Arduino.	3
Figure 3 : capteur MQ7.	5
Figure 4 : module de capteur de gaz MQ7.	7
Figure 5 : schéma équivalent de MQ7.	7
Figure 6 : module MQ7.	8
Figure 7 : les pins de MQ7.	8
Figure 8 : capteur DHT11.	9
Figure 9 : les pins de DHT11.	9
Figure 10 : chaine de donnée entre MCU et DHT11.	10
Figure 11 : signal de départ.	10
Figure 12 : signal de réponse.	11
Figure 13 : données 40 bits.	11
Figure 14 : lire la température et l'humidité.	12
Figure 15 : Les information sur SIM900.	13
Figure 16 : le schéma de branchement.	15
Figure 17 : AT commande.	16
Figure 18 : Afficheur LCD 16*2 I2C.	18
Figure 19 : buzzer.	18
Figure 20 : Relay.	19
Figure 21 : Structure générale d'IDE Arduino.	20
Figure 22 : Android studio.	21
Figure 23 : Le schéma bloque.	24
Figure 24 : organigramme principale.	27
Figure 25 : organigramme sous-programme config.	28
Figure 26 : organigramme sous-programme state.	28
Figure 27 : organigramme sous-programme action.	29
Figure 28 : organigramme sous-programme Alarme.	29
Figure 29 : organigramme sous-programme Clear.	30
Figure 30 : organigramme de l'application Android.	30
Figure 31 : lecture sms reçu à partir Arduino.	31
Figure 32 : envoi sms à partir Arduino.	32

Figure 33 : le schéma de bloc de notre carte électronique.	34
Figure 34 : logiciel Fritzing.	35
Figure 35 : câblage des capteurs.	36
Figure 36 : câblage de LCD I2C 16*2.	37
Figure 37 : câblage de Relay.	38
Figure 38 : câblage de Buzzer.	38
Figure 39 : câblage de module GSM.	39
Figure 40 : schéma électrique globale.	39
Figure 41 : Serial Monitor.	41
Figure 42 : interface principale.	42
Figure 43 : interface de paramètre.	42
Figure 44 : Interface d'action.	43
Figure 45 : Confirmation.	43
Figure 46 : exemple d'affichage de l'état.	44

Listes des tableaux

Tab 1 : Condition de travail standard.	5
Tab 2 : Conditions environnementales.	6
Tab 3 : Illustration des informations de module.	13
Tab 4 : Caractéristiques générales de SIM900.	13
Tab 5 : les AT commandes.	17
Tab 6 : comparaisant entre l'Eclipse et l'Android Studio.	22

Liste des abréviations

APK : Android Package Kit

CO : Monoxyde de Carbone

FCC : Federal Communications Commission

GSM : Global System for Mobile Communications

GND : Ground

I2C : Inter Integrated Circuit Bus

IMEI : International Mobile Equipment Identity

IDE : Integrated Développement Environnement

IBM : International Business Machines

LCD : Liquid Crystal Display

MCU : Marvel Cinematic Universe

MSB : Most Significant Bit

M2M : Machine to Machine

MMCX : Micro-Miniature Coaxiaux

ME : Mobile Equipement

NDK : Native Development Kit

NTC : Nanotubes de Carbone

Ppm : Partie Par Million

RF : Radio-Fréquence

SPI : Serial Peripheral Interface

SCL : Serial Clock Line

SDA : Serial Data line

SMS : Short Message Service

SIM : Subscriber Identity Module

TE : Terminal Equipement

USB : Universal Serial Bus

UART : Universal Asynchronous Receiver Transmitter

Introduction générale

L'intoxication au monoxyde de carbone (CO) reste la principale cause de décès par intoxication dans le monde. Comme exemple en Algérie, 200 décès et plus de 2928 hospitalisations liées aux intoxications de gaz CO ont été recensés au cours de la période allant de janvier 2017 à janvier 2018. Actuellement, les technologies de l'information et des télécommunications sont devenues une condition suffisante pour s'assurer une communication illimitée avec tous les habitants de la planète. Nous pouvons alors exploiter la technologie existante et disponible (les Smart phones) pour sauver des vies et passer une ère de sécurité sanitaire. Pour cela, on va réaliser un détecteur de CO qui fonctionne comme un actionneur et avertisseur lié par un smart phone.

La communication homme-machine ou machine-machine peut être considérée comme étant un nouveau type de dialogue possible. En effet, depuis une dizaine, d'années les appareils deviennent intelligents, agissent selon le profil des utilisateurs et sont capables de prendre des décisions de manière autonome.

En pensant à la technologie de télécommunication GSM, il nous vient immédiatement à l'esprit la communication vocale, l'envoi et la réception des SMS et des MMS, l'internet mobile, et tout ce qui fonctionne en utilisant cette technologie moderne introduite dans notre vie quotidienne.

Actuellement le réseau GSM algérien compte plus de 30 millions d'utilisateurs. "Le portable " est devenu en quelques années un produit de consommation courante. Ce petit trésor de technologie ouvre la porte à de nombreuses applications électroniques sans fil à celui qui sait l'interfacer avec un PC ou un microcontrôleur.

La forte augmentation des ventes de smart phones et de tablettes électronique se fait en même temps qu'une adoption rapide par le grand public des technologies de la domotique ainsi que l'autopilotage. Au fond, le smart phone, avec son système d'exploitation ouvert pour développer les applications androïde, devient une télécommande universelle pour tout système électrique.

Nous pouvons donc imaginer, en se servant de ces services fournis à travers la technologie GSM, que le contrôle et la commande à distance des machines, des systèmes d'alarme et de surveillance sont possibles.

C'est dans ce cadre que se situe notre projet de fin d'études intitulé " Etude et Réalisation d'un système d'alarme intelligent à base d'un smart phone contre l'intoxication due au gaz CO dans la maison " Pour se faire, nous avons divisé le travail en trois parties.

Le premier chapitre sera consacré à une étude approfondie sur tous les outils de la carte d'interface tel que l'Arduino, puis on a focalisé la lumière sur le module GSM, sa construction, son environnement de programmation et son principe de fonctionnement avec Arduino.

Le deuxième chapitre aborde l'architecture et la structure du travail sous forme de blocs fonctionnels ; où on expliquera le fonctionnement de notre programme, déclenchement d'alarme et les actions à distance.

Le troisième chapitre touche la conception et la réalisation concrète du notre système, une étude expérimentale, une description pour chaque bloc du circuit, développement d'une application Android comme interface pour l'utilisateur, où elle sera capable de gérer les commandes nécessaires sur smarte phone. Finalement, on terminera avec une conclusion générale qui résumera l'intérêt de notre étude.

Chapitre I

Outils et Développement

I.1 Introduction

L'objectif de ce chapitre introductif est de mettre notre travail dans son contexte général. Tout d'abord, nous commençons par faire une présentation succincte du sujet en détaillant son cadre et ses fonctionnalités. Ensuite, nous détaillons le cahier des charges proposé pour la réalisation de notre projet.

I.2 Cadre du projet

Notre projet a été réalisé dans le but de répondre à un ensemble de besoins qui spécifient précisément les services demandés et attendus par l'utilisateur ; ces services concernent principalement la surveillance (surveillance contre l'intoxication et commande à distance et appareil domestique), la sécurité (confidentialité d'utilisation par administrateur, détection de la température et humidité), la surveillance à distance en récupérant à tout instant l'état de système via l'envoi d'un SMS.

Dans le cadre du projet de fin d'étude, nous proposons de réaliser une carte électronique de commande connectée via un port série avec un module GSM(SIM900) permettant la surveillance et la commande à distance.

L'utilisation de cette carte est facile dont l'utilisateur pourra, à l'aide de son Smartphone, commander, demander les états et les mesure ou surveiller l'intoxication de gaz à travers un appel téléphonique ou par l'envoi d'un simple SMS code qui contient une information ou un ordre, ou bien la réception d'un message de compte rendu qui lui informe sur l'état du système

I.3 Cahier de charge proposé

La figure 1 présente la structure générale de notre système à réaliser :

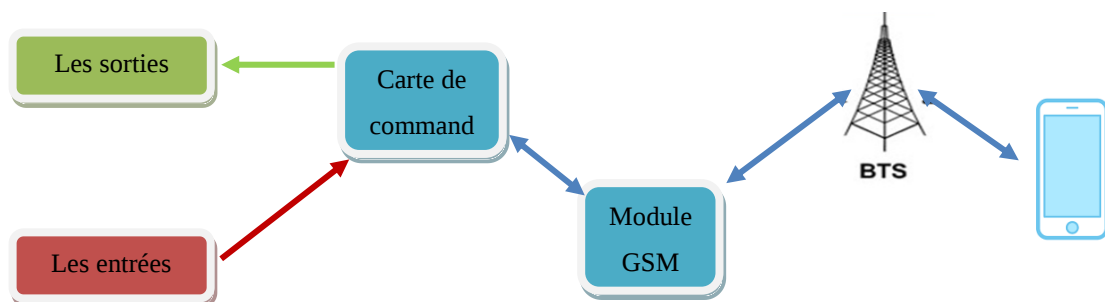


Figure 1 : structure générale

La réalisation de ce projet nécessite une étude approfondie sur certaines notions et langage de programmation qui touchent non seulement le cadre général du projet, mais aussi son implémentation. Pour bien assimiler notre connaissance et nouvelle information, nous détaillons, dans un premier lieu, tous les outils utilisés et les environnements de développement à utiliser pour la réalisation notre projet.

Pour la réalisation de notre projet qui consiste à la « réalisation d'un système d'alarme intelligent à base d'un smart phone contre l'intoxication due au gaz CO dans la maison ». Ce projet est construit par une carte mère qui interface avec un smart phone. Cette carte mère est basée sur un microcontrôleur de type ARDUINO, module GSM, capteurs de gaz CO et température et humidité, afficheur LCD et autres composants électroniques pour indication et alarme et une application portable créée à partir d'Androïde studio. Cette application doit être téléchargeable et disponible dans Google Play (magasin d'applications).

I .3.1 Microcontrôleur Arduino

Le cerveau de notre carte de commande est un microcontrôleur Arduino qui va gérer toutes les commandes nécessaires pour le fonctionnement de notre système.

Arduino est un ensemble d'outils matériel et logiciel pour le prototypage électronique et l'apprentissage de la programmation des microcontrôleurs. C'est une plate-forme de prototypage d'objets interactifs à usage créatif constituée d'une carte électronique et d'un environnement de programmation. C'est un pont tendu entre le monde réel et le monde numérique, Arduino permet d'étendre les capacités de relations homme/machine ou environnement/machine. [01]

I .3.1.1 Entrées/Sorties

Les entrées et sorties 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 2,54 mm, 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.

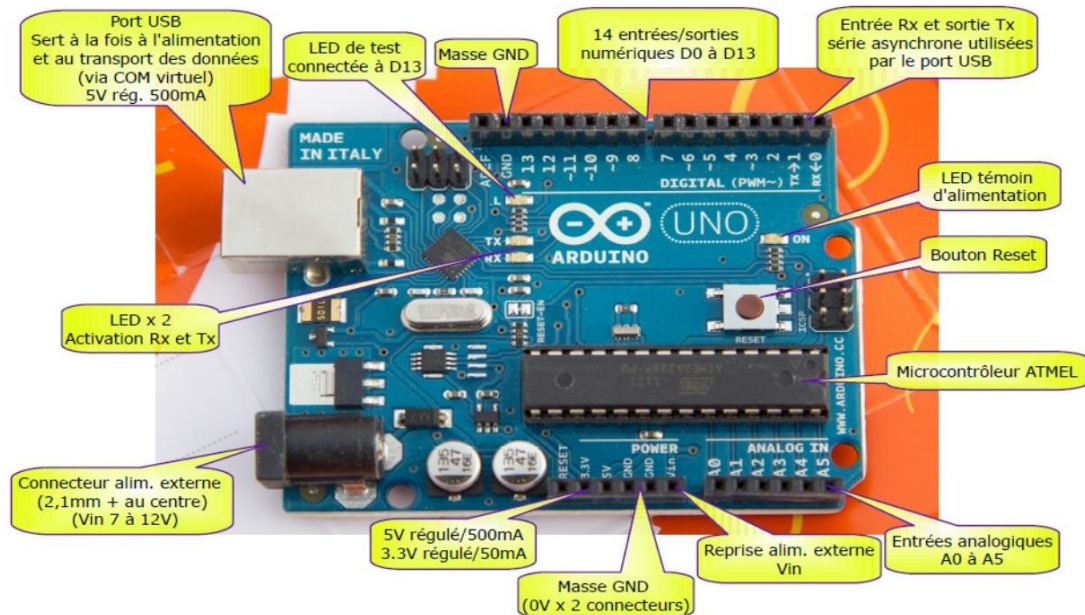


Figure 2 : microcontrôleur Arduino

I .3.1.2 La communication série avec Arduino

La communication série est indispensable pour dialoguer avec Arduino puisque c'est le mode de transmission utilisé pour communiquer avec la carte. En communication série, on découpe l'information à transmettre en petits blocs de taille fixe avant de la transmettre. La taille des blocs correspond au nombre des lignes disponibles pour la transmission des données. Ce type de communication s'oppose à la communication parallèle. En communication parallèle, il y a une ligne par bits à transmettre. Tous les bits sont donc transmis en même temps. Pour une même fréquence de communication, la communication parallèle est donc plus rapide. [02]

L'avantage de la communication série sur la communication parallèle est qu'elle nécessite moins de lignes, donc moins de broches, donc moins de composants. Son coût est donc plus faible. Les protocoles de communication série utilisés avec Arduino sont :

- Le protocole USB
- Le protocole I2C

I.3.2 Les capteurs :

Les capteurs sont des composants essentiels de la chaîne d'acquisition dans un système fonctionnel. Ce sont les premiers éléments rencontrés dans une chaîne de mesure, ils prélèvent une information sur le comportement de la partie opérative qui sera exploitée par la partie commande ; c'est à dire qu'ils transforment les grandeurs que l'on veut mesurer.

I.3.2.1 Capteur de gaz CO

Le monoxyde de carbone (CO) est un des composants oxygénés du carbone les plus connus avec le dioxyde de carbone (CO₂), plus communément appelé gaz carbonique. La confusion entre ces deux gaz est fréquente bien qu'ils diffèrent totalement, surtout par leur toxicité. Le CO est un gaz incolore et inodore à température et pression normale, avec une densité voisine de celle de l'air. Il est le résultat d'une combustion incomplète, quel que soit le combustible utilisé (bois, butane, charbon, essence, gaz naturel...). Mais l'une des principales caractéristiques de ce gaz est sa toxicité dans un environnement confiné. Il agit comme un gaz asphyxiant très toxique qui, absorbé en quelques minutes par l'organisme, se fixe sur l'hémoglobine. [03]

- 0,1 % de CO dans l'air tue en 1 heure
- 1 % de CO dans l'air tue en 15 minutes
- 10 % de CO dans l'air tue immédiatement

Après une recherche approfondie dans le marché est consultation des sites internet pour choisir un capteur de gaz CO (monoxyde de carbone) efficace et disponible pour notre projet, on a trouvé le capteur MQ-7.

Le MQ7, capteur de gaz monoxyde de carbone, est un semi-conducteur capteur de gaz qui détecte la présence du gaz monoxyde de carbone à des concentrations de 10 ppm à 10000 ppm dans l'air et fait sortir le résultat comme une tension analogique. Le capteur est composé d'un micro-tube en céramique AL₂O₃, d'une couche sensible au dioxyde d'étain (SnO₂), Le capteur est enfermé dans deux couches de maillage fin en acier inoxydable appelé réseau anti-explosion. Cela garantit que l'élément chauffant à l'intérieur du capteur ne provoquera pas d'explosion, car nous détectons des gaz inflammables. Le capteur peut fonctionner à des températures comprises entre -10 et 50 ° C et consomme moins de 150 mA à 5 V. [04]



Figure 3 : capteur MQ7

Caractéristiques MQ7 :

Selon le data sheet de capteur MQ7, on trouve les caractéristiques suivantes :

a. Conditions de travail standard

Symbole	Nom de paramètre	Condition technique	Remarque
VC	Tension de circuit	$5V \pm 0.1$	AC ou DC
VH (H)	Tension de chauffage (élevée)	$5V \pm 0.1$	AC ou DC
VH (L)	Tension de chauffage (basse)	$1.4V \pm 0.1$	AC ou DC
RL	Résistance de charge	Peut ajuster	
RH	Résistance au chauffage	$33\Omega \pm 5\%$	Température ambiante
TH (H)	Temps de chauffage (élevé)	60 ± 1 secondes	
TH (L)	Temps de chauffage (basse)	90 ± 1 secondes	
PH	Consommation de chauffage	Environ 350mW	

Tab 1 : Conditions de travail standard. [05]

b. Conditions environnementales

Symbole	Nom du paramètre	Condition technique	Remarques
Tao	Utilisation de Tem	-20°C-50°C	
Tas	Stockage Tem	-20°C-70°C	Conseils en utilisant la portée
RH	Humidité relative	Moins de 95% de Rh	
O2	Concentration d'Oxygène	21% (condition standard) Oxygène Concentration peut affecter la sensibilité.	Valeur minimale est Plus de 2%

Tab 2 : Conditions environnementales. [05]

Fonctionnement de capteur MQ7 :

Lorsque le dioxyde d'étain (particules semi-conductrices) est chauffé dans l'air à haute température, l'oxygène est adsorbé à la surface. En air pur, les électrons donneurs du dioxyde d'étain sont attirés par l'oxygène adsorbé à la surface du matériau à détecter. Cela empêche la circulation du courant électrique. En présence de gaz réducteurs, la densité de surface de l'oxygène adsorbé diminue à mesure qu'il réagit avec les gaz réducteurs. Les électrons sont ensuite libérés dans le dioxyde d'étain, permettant ainsi au courant de circuler librement dans le capteur. [06]

Module de capteur MQ7

Puisque le capteur de gaz MQ7 n'est pas compatible avec notre montage, nous utilisons le module MQ7. Il est très facile à utiliser et vient avec deux sorties différentes. Il fournit non seulement une indication binaire de la présence de gaz combustibles, mais également une représentation analogique de leur concentration dans l'air.

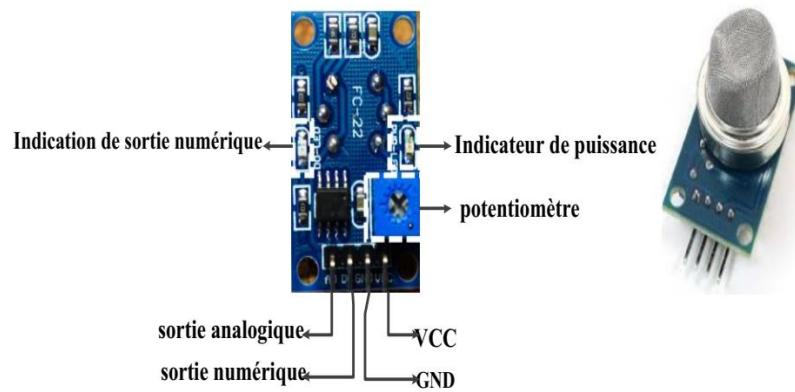


Figure 4 : module de capteur de gaz MQ7.

La tension de sortie analogique fournie par le capteur change proportionnellement à la concentration de gaz CO. Plus la concentration de gaz est grande, plus la tension de sortie est élevée, tandis que la moindre concentration de gaz entraîne une faible tension de sortie.

Schéma équivalent

Le signal analogique du capteur de gaz MQ7 est ensuite envoyé au comparateur de haute précision LM393 (soudé au bas du module), bien sûr pour numériser le signal. Avec le comparateur, un petit potentiomètre que vous pouvez tourner pour ajuster la sensibilité du capteur. Vous pouvez l'utiliser pour ajuster la concentration de gaz à laquelle le capteur le détecte.

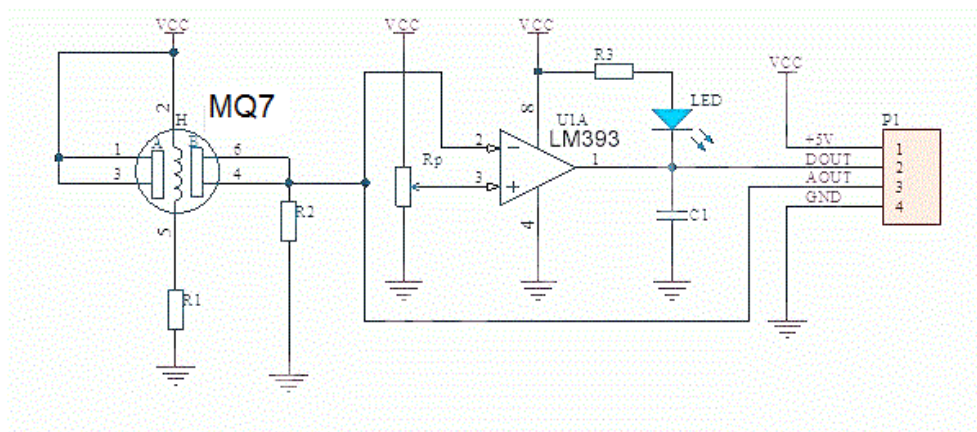


Figure 5 : schéma équivalent de MQ7

Calibrer le module de capteur de gaz MQ7

Pour calibrer le capteur de gaz, vous pouvez le maintenir près du gaz CO et continuez à tourner le potentiomètre jusqu'à ce que le voyant rouge du module commence à briller. Tournez la vis dans le sens des aiguilles d'une montre pour augmenter la sensibilité ou dans le sens contraire pour la diminuer. Le comparateur du module vérifie en permanence si

la broche analogique (A0) a atteint la valeur de seuil définie par le potentiomètre. Lorsqu'il franchit le seuil, la broche numérique (D0) passe à l'état HIGH et le voyant de signal s'allume.

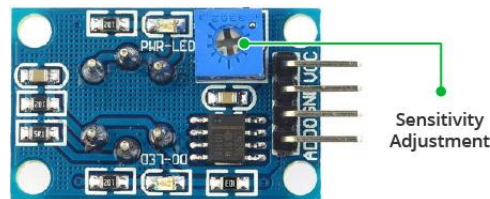


Figure 6 : module MQ7.

Brochage du module MQ7

La figure ci-dessus présente les pins du MQ7



Figure 7 : les pins du MQ7.

VCC - Alimente le module. Vous pouvez le connecter à la sortie 5V de votre Arduino.

GND - Est la broche de terre et doit être connecté à la broche GND de l'Arduino.

D0- Fournit une représentation numérique de la présence de gaz combustibles.

A0 - Fournit une tension de sortie analogique proportionnelle à la concentration de fumée / gaz.

I.3.2.2 Module DHT11

Le module DHT11 est un capteur de température et d'humidité qui se compose de deux parties une pour détection d'humidité ainsi qu'un capteur de température NTC (ou une thermistance).

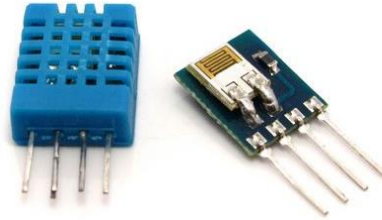


Figure 8 : capteur DHT11.

Le composant de détection d'humidité comporte deux électrodes avec un substrat retenant l'humidité (généralement un sel ou un polymère plastique conducteur) pris en sandwich entre elles. Les ions sont libérés par le substrat lors de l'absorption de vapeur d'eau, ce qui augmente la conductivité entre les électrodes. La variation de résistance entre les deux électrodes est proportionnelle à l'humidité relative. Une humidité relative plus élevée diminue la résistance entre les électrodes, tandis qu'une humidité relative plus basse augmente la résistance entre les électrodes. [07]

Le composant de détection température est une thermistance de type NTC. Une thermistance est une résistance thermique qui change de résistance en fonction de la température. Le terme "NTC" qui signifie que la résistance diminue avec l'augmentation de la température. [07]

Brochage du module DHT 11

La figure ci-dessus présente le module DHT11



Figure 9 : pins de DHT11.

VCC : Alimente le module. Vous pouvez le connecter à la sortie 5V de votre Arduino.

GND : Est la broche de terre et doit être connecté à la broche GND de l'Arduino.

DATA : Fournit une chaîne de données.

Mode de fonctionnement DHT 11

Une fois que le capteur est mis sous tension, il faut jusqu'à 2 secondes pour devenir stable. Pendant cette période, le capteur teste la température et l'humidité de l'environnement et enregistre les données relatives. Une fois terminé, il entre automatiquement en mode veille.

La communication commence lorsque le maître (MCU) envoie un signal de départ. Avec cela, le capteur sort du mode veille, passe en mode haute vitesse et envoie un signal de réponse. Suite à cela, il génère une chaîne de données de 40 bits comprenant les valeurs relatives de température et d'humidité. Une fois l'opération terminée, le capteur repasse automatiquement en mode veille en attendant la prochaine communication. Le chronogramme correspondant est présenté ci-dessous.

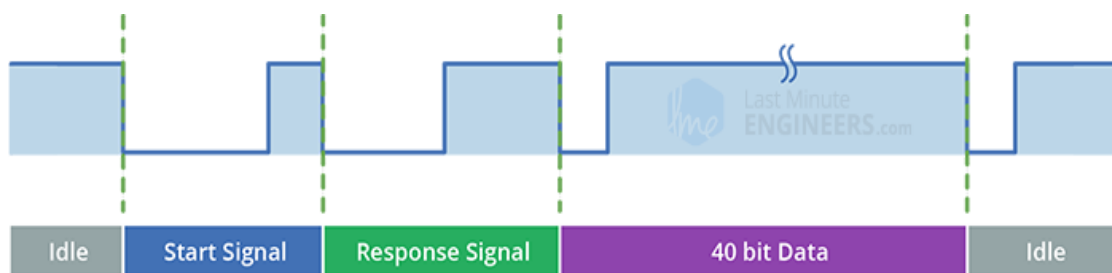


Figure 10 : chaîne de donnée entre MCU et DHT11.

Signal de départ :

Pour demander une lecture, le microcontrôleur tire la ligne de données LOW pendant plus de 800 μs (le temps de maintien typique est de 1 ms). Après ce temps assez long, la ligne est à nouveau élevée, mais pour un temps beaucoup plus court de 20 μs . Cette séquence agit comme un signal de démarrage et réveille le capteur de son état de repos.

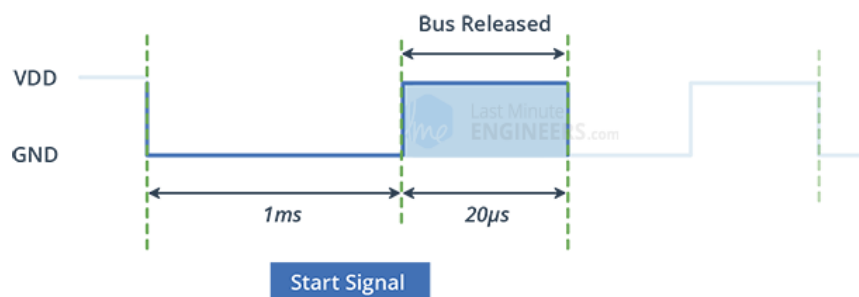


Figure 11 : signal de départ.

Signal de réponse :

Une fois le signal de départ terminé, le capteur envoie un signal de réponse. Le capteur tire la ligne de données LOW pendant $80\ \mu\text{s}$ et de nouveau sur HIGH pendant $80\ \mu\text{s}$. Durant cette période, le capteur teste à nouveau la température et l'humidité de l'environnement, enregistre les données relatives et se prépare pour la transmission des données.

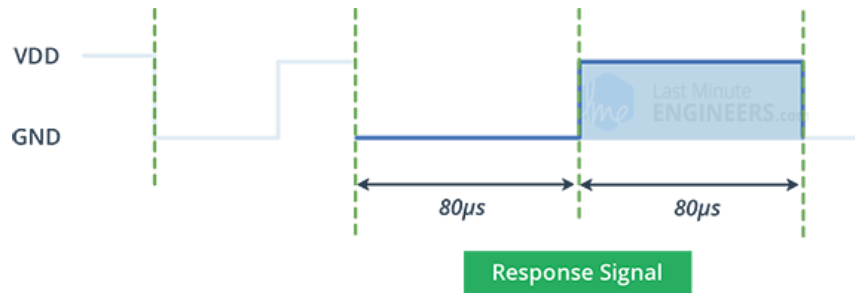


Figure 12 : signal de réponse.

Données 40 bits

Lorsque le signal de réponse est terminé, le capteur commence à émettre des données série de 40 bits en continu. Avant d'envoyer chaque bit, le capteur tire la ligne de données LOW pendant environ $50\ \mu\text{s}$. Ensuite, la durée de l'impulsion HIGH détermine l'état du bit, à savoir 0 ou 1. Si la ligne est HIGH pendant $26\text{--}28\ \mu\text{s}$, elle indique le bit '0', tandis que l'impulsion longue de $70\ \mu\text{s}$ indique le bit '1'.

Une fois que tous les 40 bits ont été transmis, le capteur tire la ligne de données LOW pendant $50\ \text{ms}$ pour indiquer la fin et passe automatiquement en mode veille.

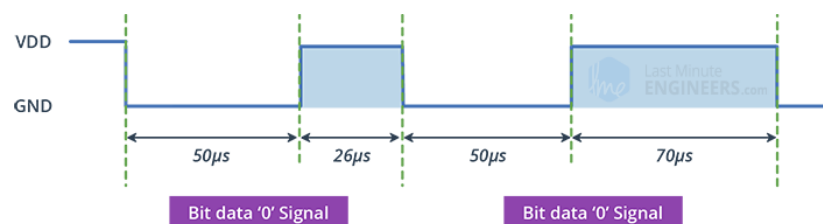


Figure 13 : données 40 bits.

Lecture température et humidité

Maintenant, l'un des aspects les plus mystérieux du capteur comprendre la signification des données 40 bits. Bien que l'humidité relative soit toujours un nombre non négatif, la température peut être négative. Et n'oubliez pas la fraction décimale du nombre. Voyons si nous pouvons résoudre ce mystère.

Les données de 40 bits contiennent 16 bits d'humidité, 16 bits de température et les 8 derniers bits représentent des bits de parité (forme la plus simple du code de détection d'erreur) avec le bit le plus significatif (MSB) en premier. Le chronogramme correspondant est présenté ci-dessous.

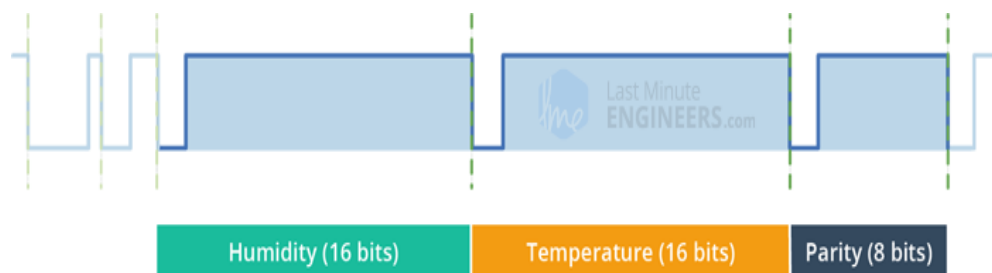


Figure 14 : lecture température et humidité.

I.3.3. Module GSM

On trouve désormais sur le marché des téléphones GSM intégrés débarrassées de leurs interfaces homme-machine, ne subsiste que la partie interface machine-machine (M2M), qui correspond physiquement à un connecteur multibroche quelconque ou encore à un connecteur DB9 facilitant la connexion à un PC. Ces modules sont universels puisqu'ils supportent les normes GSM07.07 et GSM07.05 permettent de ce fait l'échange de données, de SMS, d'emails et même de télécopies (fax) via le réseau de téléphonie mobile. Leur simplicité de mise en œuvre ouvre des perspectives très intéressantes concernant la réalisation de montages électroniques sans fil. Il est spécialement conçu pour la réalisation d'applications de contrôle de commande à distance.

En Algérie, les module GSM son des équipement sensibles et l'exploitation des équipements sensibles sont soumis à un agrément délivré par le Ministre de l'Intérieur et des Collectivités Locales ; ce qui nous rend difficile de trouver ce type de module GSM dans le marché algérien pour résoudre ce problème ou na commandé une à l'étranger. [08]

Le SIM 900 de SIMCOM

Parmi les différents modèles proposés par SIMCOM, notre choix s'est porté sur le SIM900 fabriqué par la société SIMCOM. Comportant une interface industriellement compatible, le SIM900 fournit la représentation de GSM/GPRS 850/900/1800/1900MHz. Il est capable de fonctionner dans les modes voix, données, fax et surtout, le plus intéressant pour nous, dans le mode SMS.

Conçu avec un processeur d'un seul morceau très puissant intégrant le noyau d'AMR926EJS, il nous permet de tirer bénéfice de petites dimensions et de solutions rentables. Le module dispose d'un support destiné à recevoir l'indispensable carte SIM et un connecteur MMCX permettant de relier une petite antenne RF également fournie par SIMCOM. [08]

La figure suivante montre les informations du module SIM900 :



Figure 15 : Les informations sur SIM900

Article	Description
A	Logo de SIM COM
B	Nom de module
C	Numéro de la pièce du module
D	Numéro de série du module et code `à barres
E	Module IMEI et code `à barres
F	Identification authentifiée par FCC
G	Logo authentifié par CE

Tab 3 : Illustration des informations de module. [08]

Caractéristiques générales

Transmission	Voix, données et SMS
Alimentation	3.4 V `à 4.5 V, typiquement 3.8 V
Bandes de fréquences	GSM 850 MHz, EGSM 900 MHz, DCS 1800 MHz, PCS 1900 MHz
Courant absorbe	GSM900 : 147 mA (900 mA max) GSM1800 : 127 mA (700 mA max) GSM1900 : 113 mA (650 mA max)

Puissance d'émission	Class 4 (2 W) pour bandes GSM/EGSM Class 1 (1 W) pour bandes DCS/PCS
Lecteur de carte SIM	Intégré au module, supporte les cartes SIM 3v et 1,8v
Température d'utilisation	- 40°C à + 85 °C
Antenne	Externe par connecteur MMCX
Masse	3.4 g
Taille	24 mm x 24 mm x 3mm
Normes respectées	GSM07.07 et GSM07.05
GPRS Data Services	GPRS multi-slot class (MSC) 10 (4+1, 3+2), GPRS PBCCH/PCCCH support, GPRS Class B and CC
Messages SMS (Short Message Service)	Il est ainsi possible d'envoyer de petites quantités de données via le réseau mobile (ASCII ou données hexadécimales)
Embarque un stack TCP/UDP	Permettant d'envoyer des données sur un serveur web
Contrôle à l'aide de commandes AT	Commandes standard : GSM 07.07 et 07.05 Commandes étendues : SIMCOM AT Command

Tab 4 : Caractéristiques générales de SIM900. [08]

Caractéristiques des données

- Classe 10 de GPRS : maximum 85,6 Kbps (liaison descendante).
- Transmission de données CSD, GPRS.
- SIM900 intégré protocole TCP / IP et étendu TCP / IP commande AT qui sont très utiles pour les applications de transfert de données.
- Mode non transparent.

Le module GSM Sim900A Shields Arduino

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 de la société SIMCOM. Il est contrôlé via les commandes AT depuis une carte Arduino. Un connecteur au dos de la platine est prévu pour recevoir une carte SIM ainsi qu'une pile Lithium CR1220 pour l'horloge temps réel. La communication entre le module et une carte Arduino est réalisée par la liaison série asynchrone UART ou une liaison série logicielle.

Les Shields Arduino sont des cartes qui se branchent sans souder aux cartes Arduino ou à d'autres Shields Arduino pour augmenter leurs capacités. Ces cartes ont des dimensions sensiblement voisines de la carte Arduino et peuvent s'enficher directement sur les connecteurs de celle-ci. Les Shields Arduino conservent l'esprit original d'Arduino, facile à produire et à utiliser.

Les cartes Arduino, comme les Shields Arduino sont particulièrement adaptées pour le monde des Makers et du DIY (Do It Yourself).

La Figure 16 montre le schéma de branchement de module Arduino avec carte GSM

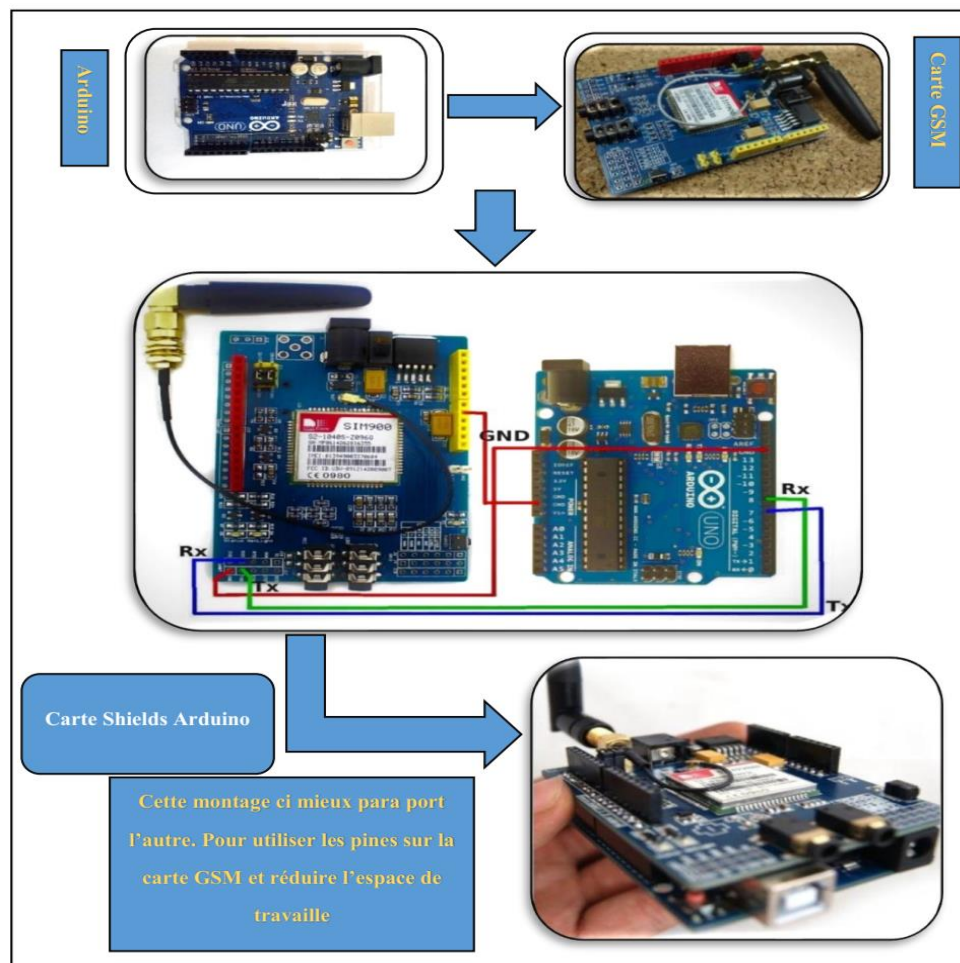


Figure 16 : schéma de branchement.

Les Ccommandes AT

Il existe un standard de télécommunication européen (ETS) qui spécifie une liste de commandes AT qui permettent l'accès aux fonctions d'un téléphone portable par l'intermédiaire d'un terminal. Ces commandes s'inspirent fortement du standard Hayes, du nom de la société américaine qui dans les années 1970 a défini une liste de commandes universelles permettant de piloter un modem.

Chaque instruction débute par les caractères ASCII « AT » tirés de l'abréviation « ATtention » et se termine par un retour chariot, CR : Carriage Return, d'où le nom souvent donné à cette série de commandes : instructions « AT ». On peut effectivement comparer un

téléphone portable à un modem sans fil, il est donc logique qu'il utilise des instructions semblables au modem fixe qui équipe nos PC.

Les constructeurs se doivent de fabriquer des téléphones portables qui respectent ces normes. La première baptisée GSM07.07 permet l'accès aux fonctions générales du téléphone, la deuxième GSM07.05 concerne la gestion des SMS. Dans les textes officiels qui traitent du GSM on retrouve les termes ME pour Mobile Equipement qui correspond par exemple à un téléphone portable, TE pour Terminal Equipement qui physiquement peut être un ordinateur ou un microcontrôleur et TA pour Terminal Adaptator qui assure la liaison entre le ME et le TE, à ne pas confondre avec le câble série. [08]

Dans la pratique, il y a trois possibilités concernant la disposition des différents éléments (figure [17]) :

- TA, ME et TE sont trois entités distinctes ;
- TA et ME forment une seule entité, ce qui est le cas le plus fréquent. Par exemple, un téléphone portable standard ou un terminal GSM contient dans son boîtier à la fois le TA et le ME. Le TE forme une entité à part, par exemple il peut s'agir d'un ordinateur de type PC qui dispose d'un port série ou d'un circuit électronique basé sur un μ C qui implémente un port série. [08]
- TA, ME et TE forment une seule entité.

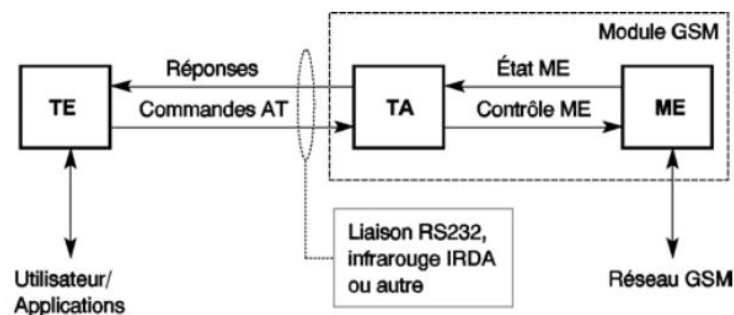


Figure 17 : AT commande.

La norme GSM07.07

La norme GSM07.07 regroupe environ 80 commandes permettant d'accéder à toutes les fonctions du ME. Nous n'allons pas détailler la totalité de ces commandes mais seulement celles qui seront susceptibles de nous intéresser dans notre application (tableau [5]) [08].

Commandes	Fonction
AT+CGMI	Identification fabricant
AT+CGMM	Identification modèle
AT+CGMR	Identification version
AT+CGSN	Identification numéro de série (IMEI)
AT+CIMI	Information d'identité internationale du mobile (IMSI)
AT+CLIP	Présentation du numéro
AT+CSCS	Alphabet utilisé par le TE
AT+CPAS	État d'activité du téléphone
AT+CPIN	Entre le code PIN
AT+CBC	État de charge batterie
AT+CREG	Enregistrement sur le réseau
AT+CSQ	Qualité du signal
AT+CIND	Indicateurs de contrôle
AT+CPBS	Sélectionne un répertoire téléphonique
AT+CPBR	Lecture du répertoire téléphonique
AT+CPBF	Recherche une entité dans le répertoire téléphonique
AT+CPBW	Écriture dans le répertoire téléphonique
AT+CCLK	Horloge
AT+CALA	Alarme
AT+CMEE	Signalisation d'une erreur
AT+CSMS	Sélection du service de messagerie
AT+CPMS	Sélection de la zone mémoire pour le stockage des SMS
AT+CMGF	Sélection du format du SMS (PDU ou TEXT)
AT+CSCA	Définition de l'adresse du centre de messagerie
AT+CSDH	Affiche en mode TEXT le paramétrage des SMS
AT+CSAS	Sauvegarde du paramétrage
AT+CRES	Restauration du paramétrage par défaut
AT+CNMI	Indication concernant un nouveau SMS
AT+CMGL	Liste les SMS stockés en mémoire
AT+CMGR	Lecture d'un SMS
AT+CMGS	Envoie un SMS
AT+CMSS	Envoie d'un SMS stocké en mémoire
AT+CMGW	Écriture d'un SMS
AT+CMGD	Efface un SMS

Tab 5 : les AT commandes.

I.3.4 Afficheur LCD 16*2 I2C

Pour afficher les informations nécessaires de notre évènement et l'indication des mesures, on a utilisé un afficheur LCD 16*2 de type I2C qui utilise la moindre connexion juste avec 2 fils.

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 afficheurs et des microcontrôleurs. 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.



Figure 18 : Afficheur LCD 16*2 I2C.

I.3.4. Le Buzzer

Pour simuler l'effet d'une sirène, un buzzer est utilisé pour générer une alarme sonore. Un buzzer ou un bipper est un dispositif de signalisation audio, qui peut être mécanique, électromécanique ou piézoélectrique.



Figure 19 : buzzer.

I.3.5. Relais

Un relais est un commutateur électrique qui peut être activé ou désactivé, laissant le courant passer ou non, et pouvant être contrôlé avec des tensions faibles, comme le 5V fourni par les broches Arduino.



Figure 20 : Relay.

I .4. Logiciels utilisés

Lors de notre réalisation, nous nous sommes basés sur deux logiciels, Arduino IDE et Android Studio, et un autre logiciel pour le dessin du schéma électrique. Le premier nous a servi à programmer et configurer notre carte Arduino pour le bon fonctionnement de notre application. Le deuxième software nous a servi à une application mobile pour faciliter les tâches à distance de notre système.

I .4.1 L'environnement de la programmation (IDE ARDUINO)

L'environnement de développement intégré (IDE) Arduino est une application multiplateforme (pour Windows, mac OS, Linux) écrite en langage de programmation Java. Il est utilisé pour écrire et télécharger des programmes sur des cartes compatibles Arduino, mais aussi, avec des cartes de développement d'autres fournisseurs. L'IDE Arduino supporte les langages C et C ++ en utilisant des règles spéciales de structuration du code. L'IDE Arduino fournit une bibliothèque de logiciels du projet, qui fournit de nombreuses procédures d'entrée et de sortie communes. Le code écrit nécessite seulement deux fonctions de base, pour le démarrage de l'esquisse et de la boucle du programme principal, qui sont compilées et liées à un talon de programme `main ()` dans un programme exécutable cyclique avec la chaîne d'outils `Loop ()`, également incluse dans la distribution IDE. L'IDE Arduino utilise le programme AVRDUDE pour convertir le code exécutable en un fichier texte au codage hexadécimal chargé dans la carte Arduino par un programme de chargement dans le microprogramme de la carte.

Structure générale du programme

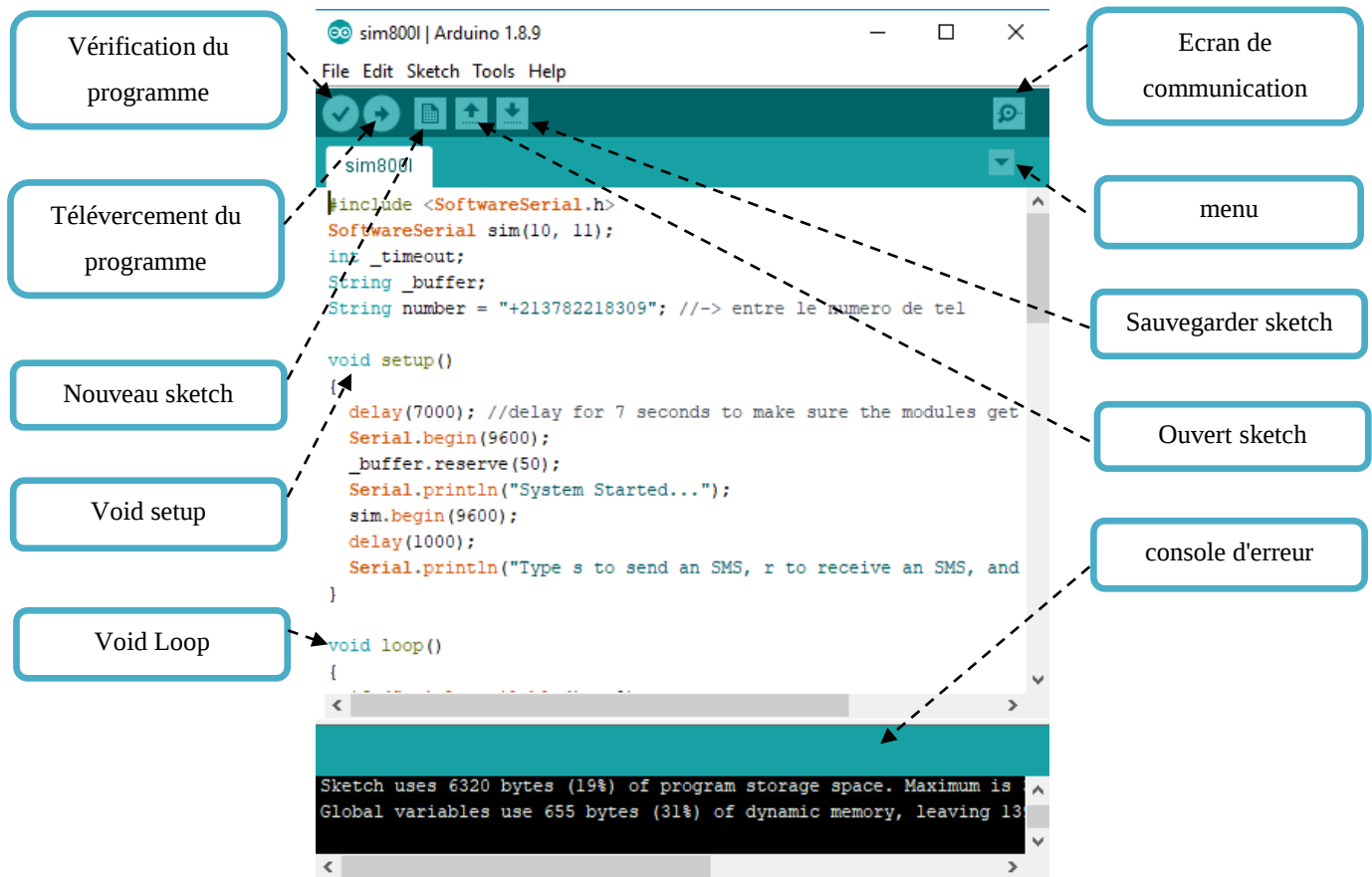


Figure 21 : Structure générale d'IDE Arduino.

I .4.2 Android Studio

Aujourd'hui, les applications mobiles occupent une place importante dans notre vie quotidienne grâce aux services offerts par ces outils qui couvrent une grande variété de domaines.

Pour développer une application native sur Android, il y a plusieurs environnements appelé IDE, Integrated Développement Environnement (EDI environnement de développement intégré en français) [09], tels qu'Eclipse, NetBeans et Android Studio [09].

Google propose un IDE appelé Android Studio pour la création d'applications Android. Le premier IDE indispensable pour les développeurs Android, Android Studio est un environnement de développement basé sur IntelliJ IDEA, qui avait pour objectif de supplanter Eclipse [10].

- **Eclipse**

Eclipse est un logiciel qui simplifie la programmation en proposant un certain nombre de raccourcis et d'aide à la programmation. Il est développé par IBM, est gratuit et disponible pour la plupart des systèmes d'exploitation. [11]

- **Android studio**

Android studio est un environnement de développement et de programmation entièrement intégré, cet environnement a été lancé par Google pour les systèmes à base d'Android. Il a été conçu pour fournir un environnement de développement et une alternative à Eclipse qui est l'IDE le plus utilisé.

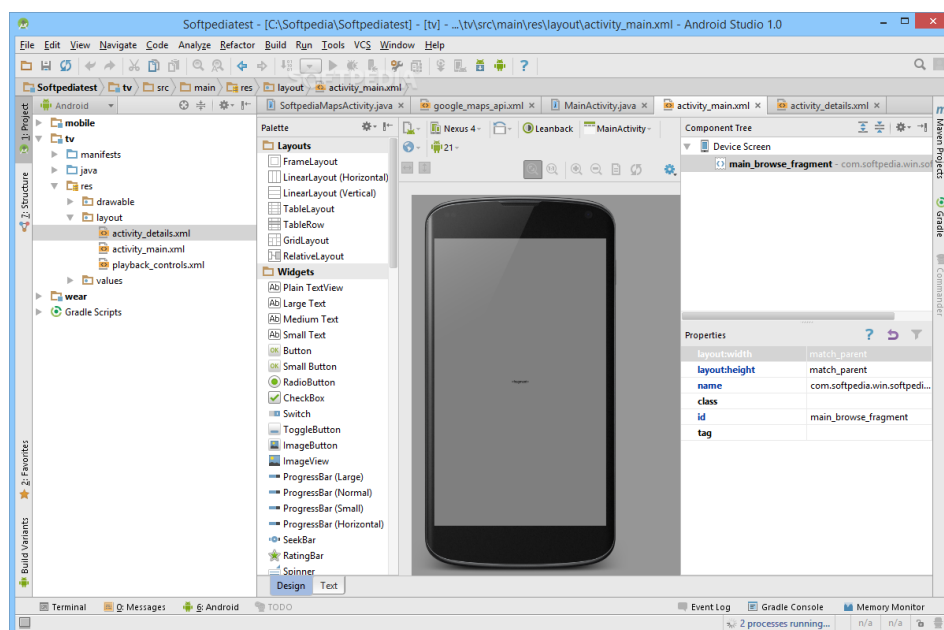


Figure 22 : Android studio.

Comparaison entre l'Eclipse et l'Android Studio

	Eclipse	Android Studio
Facilité d'installation	Moyen	Simple
Langue	Nombreuses	Anglais
Performance	Peut être lourd	Rapide
Système de built	Ant	Gradle
Génération de variante et de multiple APK	Non	Oui
Android Code completion	Base	Avancé
Editeur d'interface graphique	Oui	Oui
Signature d'APK et gestion de Keystore	Oui	Oui
Support NDK	Oui	A venir

Tab 6 : comparaissant entre l'Eclipse et l'Android Studio [12].

Les outils utilisés

Comme on a vu sur la Tab [6], nous avons choisi l'environnement Android studio parmi les autres outils de développement.

JAVA

C'est le langage utilisé sur la plateforme Androïde studio. Java est un langage de programmation. Ce langage a la particularité principale que les logiciels écrits avec ce dernier sont très facilement portables sur plusieurs systèmes d'exploitation tels qu'Unix, Microsoft Windows, Mac OS ou Linux avec peu ou pas de modifications. C'est la plate-forme qui garantit la portabilité des applications développées en Java. [13]

Conclusion

Tout au long de ce chapitre, nous avons pu situer le cadre général de notre projet de fin d'études, à savoir la présentation de ses objectifs, le cahier des charges proposé. Nous avons décrit les bases théoriques développées dans notre projet, et pris une idée générale sur tous les composants essentiels de notre carte électronique avec les logiciels utilisés.

Dans le chapitre suivant, on va décrire la structure générale et le mode de fonctionnement de notre système.

Chapitre II

Architecture et Programme

II.1 Introduction

Après avoir étudié les outils essentiels que nous avons utilisés dans notre projet, nous allons, dans ce chapitre, décrire l'architecture et tous les organigrammes de programmation de notre carte Arduino et notre application Android et avec le schéma bloc du système.

II.2 Description générale

Dans le but de réaliser un système d'alarme à distance performante, un cahier de charge a été élaboré en collaboration avec notre encadreur, dont les grandes lignes du projet ont été élaborées comme suit.

- ✚ Le système doit être connecté au réseau GSM.
- ✚ Le système sera conçu sous forme une carte électronique.
- ✚ Le système peut être aussi utilisé facilement avec l'utilisateur.
- ✚ Le système doit informer les assistants d'utilisateur automatiquement.
- ✚ Le système doit interfacer avec une application androïde pour faciliter les tâches.
- ✚ Le système doit activer automatiquement les actions nécessaires en cas d'urgence.

Afin de répondre à ce cahier de charge, la technologie Arduino sera employée. Le schéma bloc choisi pour réaliser notre système est donnée par la figure ci-dessous.

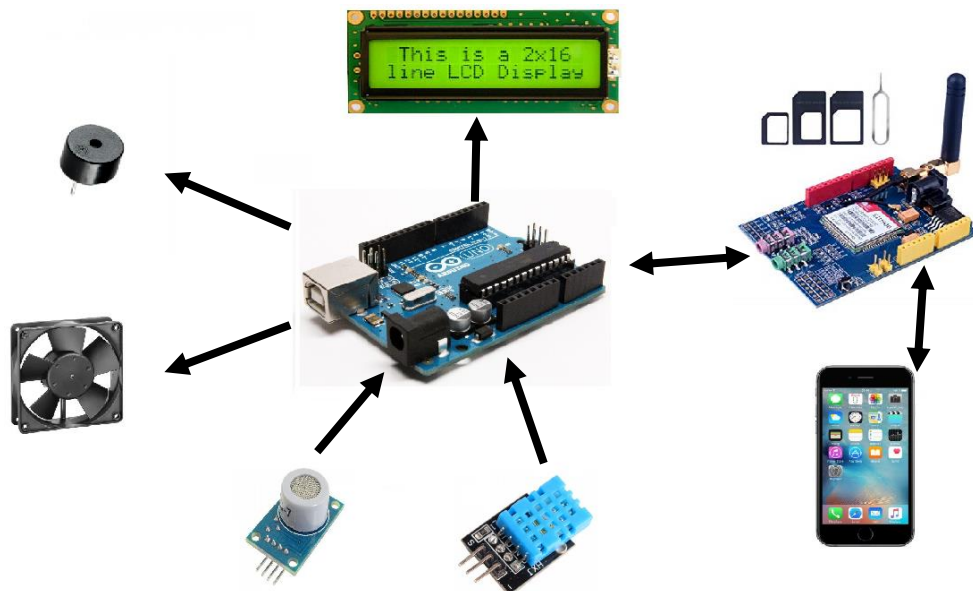


Figure 23 : Le schéma bloc.

II.3 Le mode de fonctionnement de système

Le système conçu sera élaboré comme suit :

Insérer une carte SIM valide avec code pin désactivé puis connecter l'adaptateur électrique. Ensuite, brancher l'adaptateur au secteur. Attendre jusqu'à ce que le voyant power du module GSM s'allume. Dans le cas contraire, un message « E1 » s'affiche à l'écran ce qui indique l'absence de connexion entre Arduino et le module GSM.

Après l'allumage LED power, attendre que le LED net du module GSM clignote une fois toutes les 3 secondes ; ce qui indique l'enregistrement aux réseaux GSM. Dans le cas contraire, un message « E2 » s'affiche à l'écran ; ce qui n'indique pas d'enregistrement aux réseaux GSM.

La carte électronique fonctionnant normalement, on va lire la température, l'humidité et la concentration de gaz CO. Dans le cas de mauvaise connexion avec le capteur de température, le message « E3 » s'affiche à l'écran.

Le système va lire s'il y a un nouveau SMS reçu, Si oui, on distingue quatre cas différents.

Si SMS= « CONFIG », on va extraire le numéro de téléphone administrateur et ses assistants et les sauvegarder dans la mémoire EEPROM de microcontrôleur.

Si SMS= « ACTION », il va vers une action ouvrir l'aspirateur ou un autre appareil domestique connectée au système.

Si SMS= « STATE », le système envoie un SMS qui contient les valeurs de température, humidité et la concentration de gaz CO à l'administrateur.

Si SMS = « CLEAR », le système va effacer tous les numéros enregistrés dans la mémoire EEPROM et le système attend une nouvelle configuration.

Si SMS = autre sms, le système va l'effacer et retour au programme principal.

En cas d'alarme après le système, il va alimenter l'aspirateur ou fermer l'électrovanne d'entrée gaz et envoyer des SMS à l'administrateur et tous les autres assistants pour l'informer de la présence de danger.

Le système est toujours en attente de recevoir un nouveau SMS envoyé à partir de l'administrateur et vérifier le contenu du SMS et va l'effacer après le traitement pour libérer l'espace mémoire de SMS dans la carte SIM.

II.3 Les types de programmation

Du premier chapitre, on peut comprendre que notre réalisation software a besoin de deux étapes : la première consiste à un programme qui va s'injecter aux microcontrôleurs de la carte Arduino après avoir été convertie par l'IDE en code HEX et la deuxième a une application qui va se manipuler sous Android Studio et s'installer sous smart phone avec une extension « *.apk ».

II.4 Organigrammes

Organigramme est une représentation graphique normalisée de l'enchaînement des opérations et des décisions effectuées par un programme d'ordinateur.

II.4.1 Organigrammes ARDUINO

Avant de passer à la programmation, nous devons réaliser un organigramme qui explique le déroulement des différentes séquences, tant intérieures qu'extérieures : il comportera plusieurs boucles et des sous programmes.

II.4.2 Organigramme principale

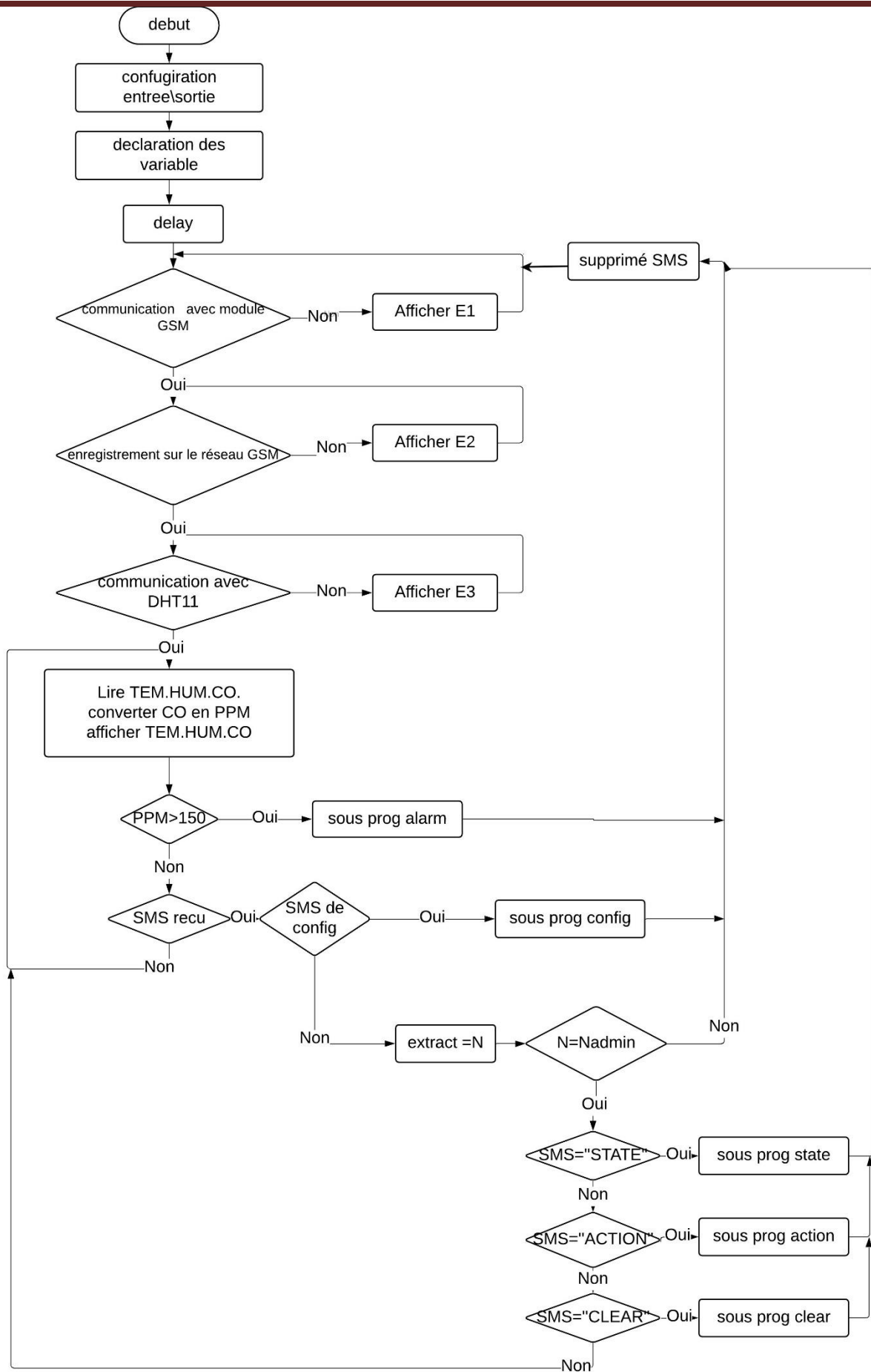


Figure 24 : organigramme principale.

II.4.3 Organigramme sous-programme CONFIG

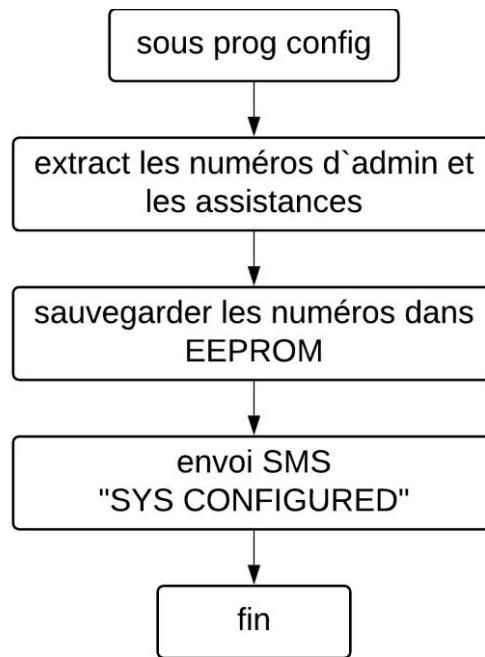


Figure 25 : organigramme sous-programme config

II.4.5 Organigramme sous-programme STATE

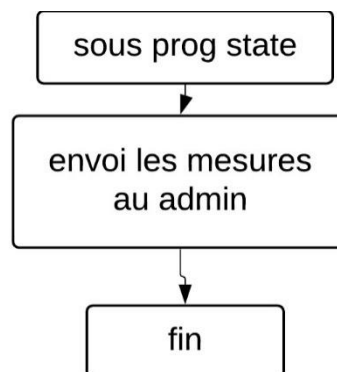


Figure 26 : organigramme sous-programme state

II.4.5 Organigramme sous-programme ACTION

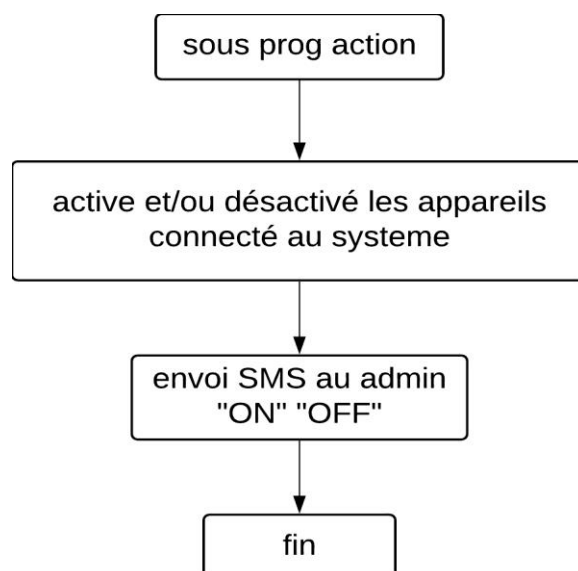


Figure 27 : organigramme sous-programme action

II.4.6 Organigramme sous-programme ALARME

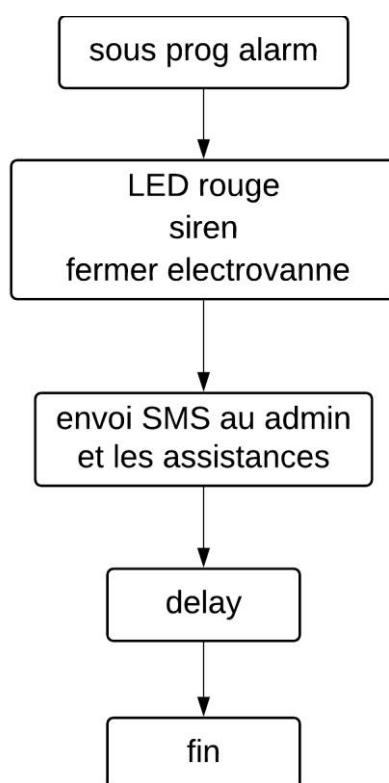


Figure 28 : organigramme sous-programme Alarme

II.4.7 Organigramme sous-programme CLEAR

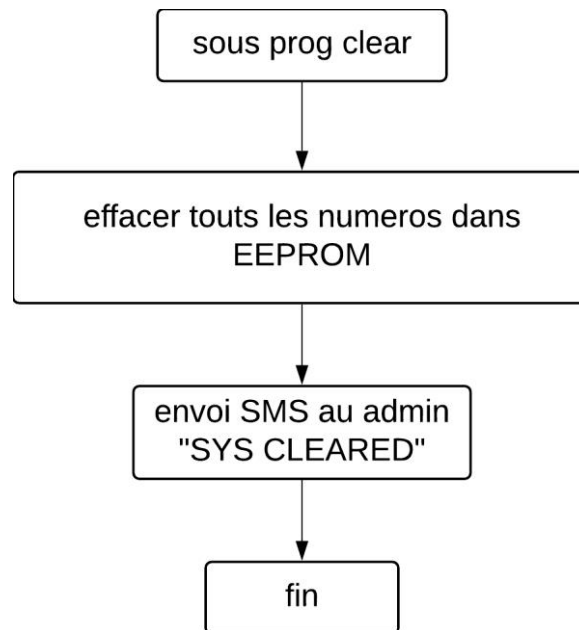


Figure 29 : organigramme sous-programme Clear

II.4.8 Organigramme application ANDROID

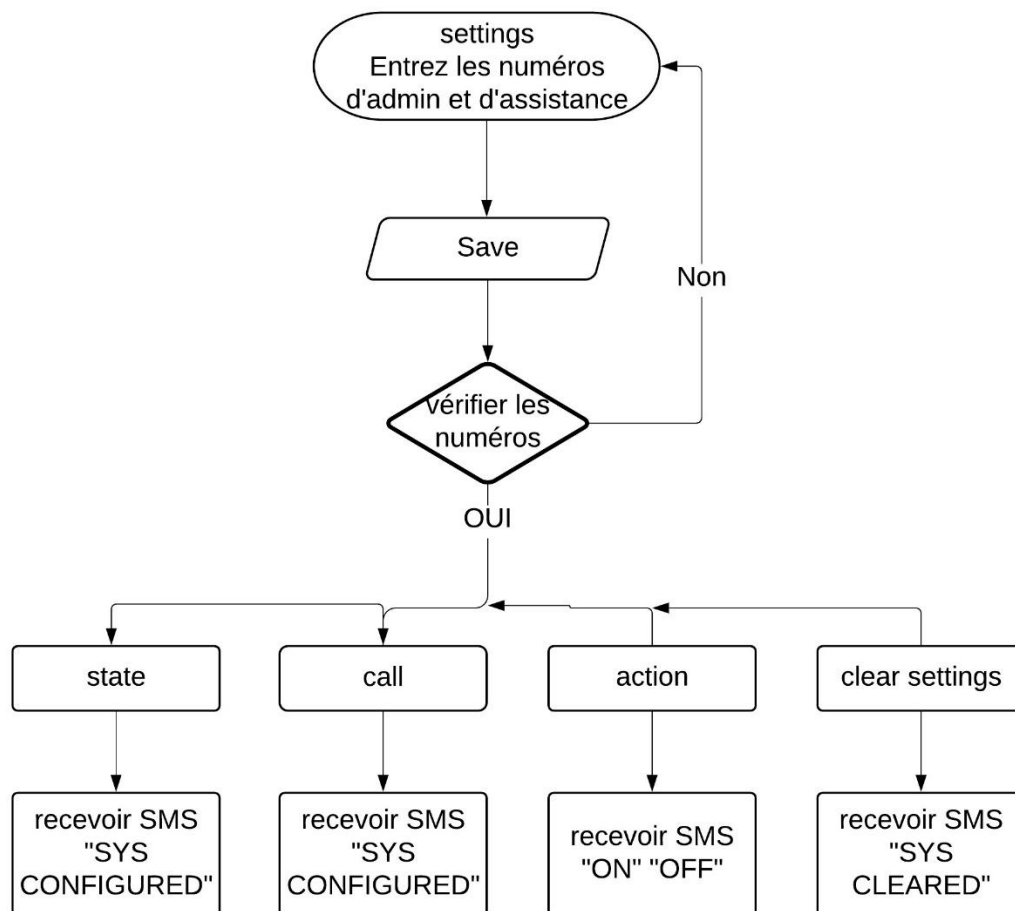


Figure 30 : organigramme de l`application Android

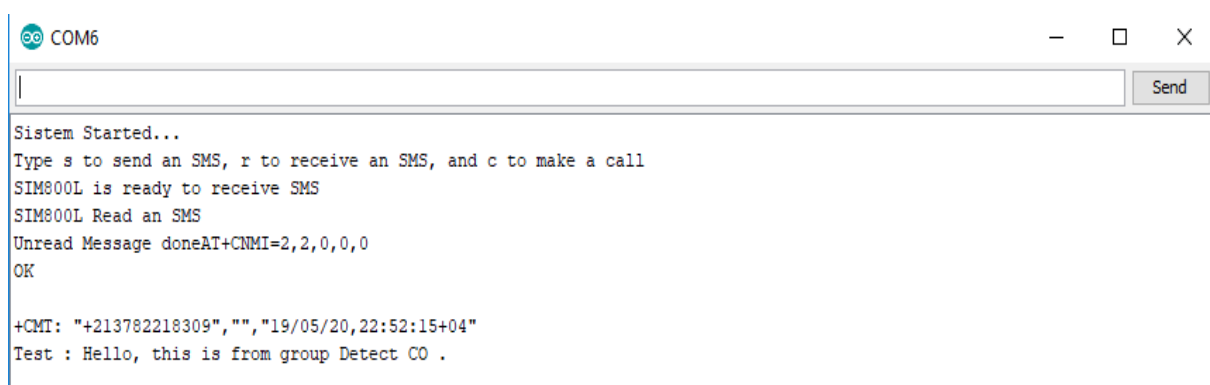
II.5 Lecture sms reçu à partir d'ARDUINO

Notre système est basé généralement sur le traitement des SMS reçus. On va démontrer la procédure et la forme de ces SMS.

Avant de lire le SMS réception par le module GSM, il faut activer une option sur le module pour la détection des nouveaux SMS reçus et non lus à partir de l'envoi de la commande « AT+CMNI=2,2,0,0,0 » alors, lorsqu'un SMS est reçu au module, il envoie au microcontrôleur une attention qu'un SMS reçu avec l'envoi des caractères suivants « +CMT ». Le microcontrôleur prend l'attention pour lire et sauvegarder le contenu de SMS pour la détection du numéro téléphone de l'expéditeur et le contenu donne un SMS envoyé à partir du module GSM et le microcontrôleur est comme suit :

+CMT : "+213782218309 ","", "19/05/20,22:50:28+04" Test Hello, this is from group Detect CO .

- ✓ Le contenu du premier champ "+213782218309 " est le numéro téléphone de l'expéditeur.
- ✓ La valeur du deuxième champ doit être une chaîne contenant le texte associé au numéro de téléphone de l'expéditeur "+213782218309 " dans le répertoire. Dans cet exemple, ce champ est vide.
- ✓ La valeur du quatrième champ, "19/05/20, 22: 50: 10 + 04", indique que le SMSC a reçu le message texte à 22:50:10 (GMT + 1) le 20 mai 2019. (Don n'oubliez pas que l'unité de fuseau horaire est fois 4, "+04" signifie GMT +1 heures.)
- ✓ La valeur du cinquième champ, " Test Hello, this is from group Detect CO ", Est le corps du message SMS.



```

Sistem Started...
Type s to send an SMS, r to receive an SMS, and c to make a call
SIM800L is ready to receive SMS
SIM800L Read an SMS
Unread Message doneAT+CMNI=2,2,0,0,0
OK
+CMT: "+213782218309","", "19/05/20,22:52:15+04"
Test : Hello, this is from group Detect CO .

```

Figure 31 : lecture sms reçu à partir Arduino

II.6 Envoi sms à partir ARDUINO vers portable



```
envoi_sms

#include <SoftwareSerial.h>
#define SIM900_TX_PIN 8;//SIM900 TX is connected to Arduino D8
#define SIM900_RX_PIN 7;//SIM900 RX is connected to Arduino D7
SoftwareSerial serialSIM900(SIM900_TX_PIN,SIM900_RX_PIN);//Create software serial object to communicate with SIM900
void setup() {
  Serial.begin(9600); //Begin serial communication with Arduino and Arduino IDE (Serial Monitor)
  serialSIM900.begin(9600); //Begin serial communication with Arduino and SIM900
  delay(1000);
  Serial.println("Setup Complete!");
}
void loop() {
  SendTextMessage();
}
void SendTextMessage()
{
  Serial.println("Sending Text...");
  serialSIM900.print("AT+CMGF=1\r"); // Set module gsm on SMS mode
  delay(100);
  serialSIM900.print("AT+CMGS=\"0669374934\"\r"); //0669374934 le numeros de destinateur
  delay(200);
  serialSIM900.print("it is from group capteur a gas "); //le content de message
  serialSIM900.print("\r");
  delay(500);
  serialSIM900.print((char)26);//fin de sms
  delay(100);
  serialSIM900.println();
  Serial.println("Text Sent.");
  delay(500);
}

Enregistrement terminé.
```

Figure 32 : envoi sms à partir d'Arduino

Conclusion

Sur le plan pratique, une manipulation adéquate avec le logiciel « Arduino » nous permet alors d'utiliser un compilateur ; il s'agit de « IDE ». Ce dernier possède une capacité de créer un code HEX, qui peut être injecté sur son microcontrôleur d'une part ; d'une autre part, une simple élaboration de l'environnement androïde studio qui nous pousse à réaliser une application « *.apk » sous smart phone capable de lier une carte Arduino et un smart phone afin d'exécuter des ordres bien définis. On peut conclure que les résultats obtenus sont satisfaisants compte tenu des limitations du matériel et des moyens dont nous disposons.

Chapitre III

Conception et Réalisation

III.1 Introduction

Après avoir étudié les organigrammes essentiels que nous avons utilisé dans notre projet, nous allons, dans ce chapitre, appliquer toutes les étapes nécessaires pour la réalisation de notre projet qui se divise en deux grandes parties : la réalisation de la carte qui est basé sur un microcontrôleur Arduino Uno pour contrôler toutes les fonctions du système et interfacier tous les composants de ce système et la deuxième partie est une application Android créée à partir le plateforme Android Studio qui va être publiée au Play store pour la télécharger et l'installer dans le smart phone de l'utilisateur. Nous pouvons maintenant passer à la réalisation physique du système.

III.2 Partie hardware

Dans cette partie, on va démontrer la réalisation électrique de chaque composant électronique et les travaux expérimentaux.

III.2.1 La réalisation électrique

Tout d'abord, nous allons rappeler le schéma fonctionnel et le schéma de bloc de notre carte électronique avant de passer à présenter sa réalisation physique. En effet, dans la figure 12, notre carte électronique contient plusieurs modules de fonctions différentes que nous avons déjà détaillées dans le chapitre "I". Ces modules sont :

- Module de traitement de l'information
- Module de commande.
- Module de contrôle.
- L'interface de mesure.
- La Carte de communication GSM.

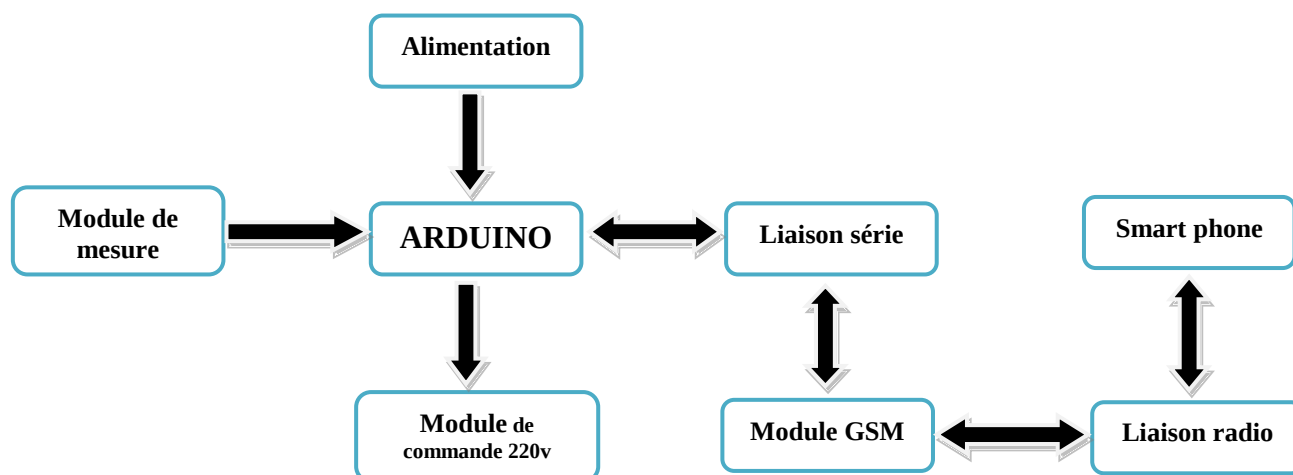


Figure 33 : le schéma de bloc de notre carte électronique

Tous d'abord, on a tracé le schéma électrique du système sur le logiciel Fritzing. Le logiciel Fritzing est un outil de création des projets et des circuits électroniques ; il permet aussi l'édition de circuit imprimé, il est disponible gratuitement sur internet. Il a notamment pour vocation de favoriser l'échange de circuits électroniques libres et d'accompagner l'apprentissage de la conception de circuits. Il est un bon outil didactique pour apprendre à bidouiller en électronique par la pratique :

Le logiciel comporte trois vues principales :

- La « Platine d'essai », où l'on voit les composants tels qu'ils sont dans la réalité et où l'on construit le montage.
- La « Vue schématique », représentant le schéma fonctionnel du circuit.
- Le « Circuit imprimé », représentant la vue du circuit imprimé tel qu'il sera sorti en PDF pour être imprimé.

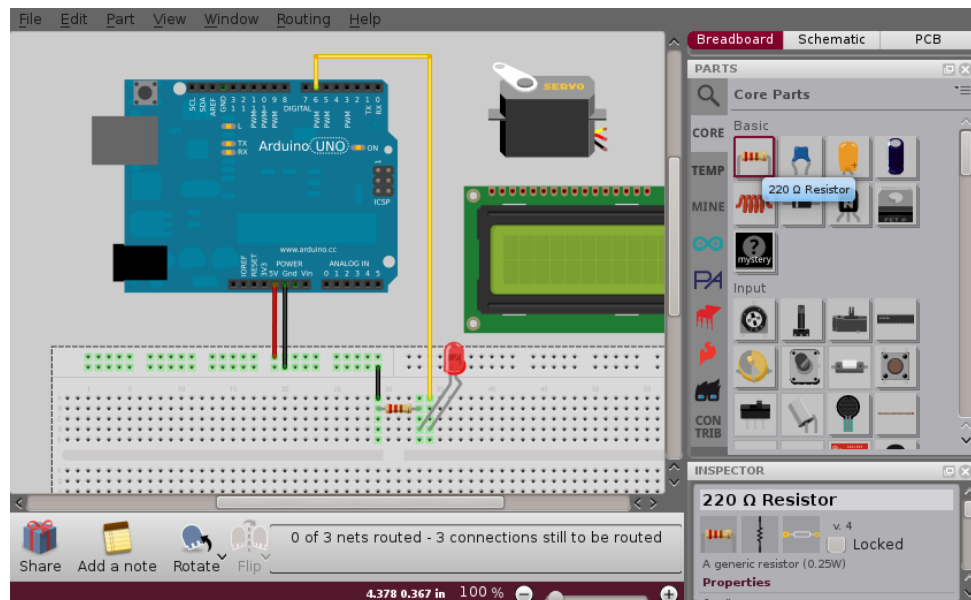


Figure 34 : logiciel Fritzing.

III.2.1.1 Alimentation

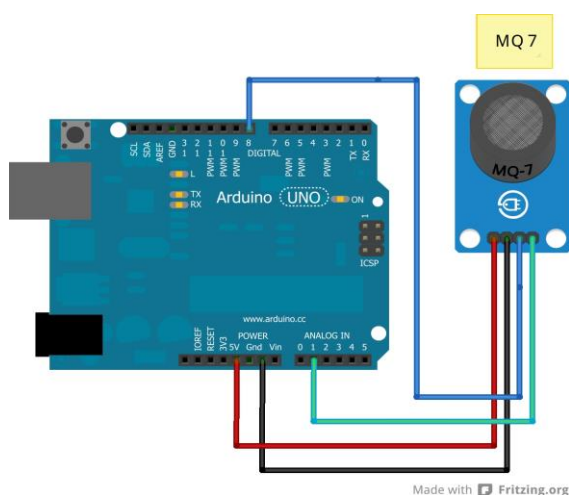
Après avoir étudié les caractéristiques électriques de tous les composants électroniques nécessaires pour notre carte, on a utilisé une alimentation stabilisée, composée d'un transformateur 220V/5V, qui transforme la tension de secteur 220 v alternative en tension contenue de 5 volts avec 3000 mA d'intensité de courant.

La majorité des applications à base de microcontrôleur doit communiquer avec le milieu extérieur par le biais de ce que l'on désigne sous le nom générique d'entrées / sorties. Ces derniers varient évidemment très fortement d'une application à une autre mais elles font appel à des techniques de base classique qu'on essaye de développer ci-dessous.

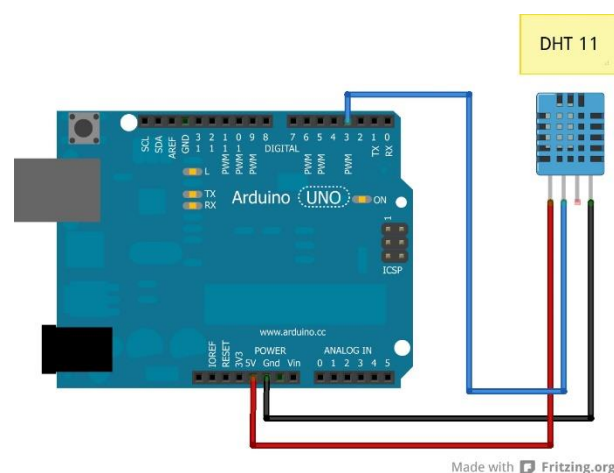
III.2.1.2 Les capteurs

Le microcontrôleur possède un convertisseur analogique/numérique chargé de convertir un signal analogique en une donnée numérique. Les signaux sont généralement de nature électrique délivrés par le capteur MQ7 qui va nous donner la concentration de gaz CO avec une unité de PPM (parties par million). La broche A0 de MQ7 est connecté à l'entrée analogique de Arduino A1, la broche numérique de MQ7 est connectée à l'entrée numérique Arduino D8 comme présente la figure 02.

Le capteur DHT11 nous donne à partir de sa boîte DATA un signal numérique qui contient les informations de température et humidité est relie directement à l'entrée numérique de microcontrôleur D3.



Câblage de MQ7



câblage de DHT11

Figure 35 : câblage des capteurs

Le PPM

Lors de la mesure de gaz tels que le dioxyde de carbone, le terme de concentration est utilisé pour décrire la quantité de gaz en volume dans l'air. Les deux unités de mesure les plus courantes sont les parties par million et la concentration en pourcentage.

Le nombre de parties par million (ppm en abrégé) est le rapport d'un gaz à un autre. Par exemple, 1 000 ppm de CO signifie que si vous pouvez compter un million de molécules de gaz, 1 000 d'entre elles seraient constituées de monoxyde de carbone et 999 000 molécules seraient d'autres gaz.

III.2.1.3 Afficheur LCD :

Pour l’affichage des notifications des SMS, appel, mesure et alarme, on utilise l’afficheur LCD 16*2 c'est-à-dire deux lignes de 16 caractères. Avec un module de protocole I2C intégré qui nous facilite le travail car notre microcontrôleur comporte d’instructions spécialement dédiées pour le pilotage de cette affichage et aussi moins le câblage de fils.

I2C est un bus série synchrone bidirectionnel half-duplex. Cette liaison est utilisable avec seulement 2 broches (une broche de donnée et une broche d’horloge) et nécessite l’utilisation de deux broches analogiques de l’Arduino (broche A4 et A5).

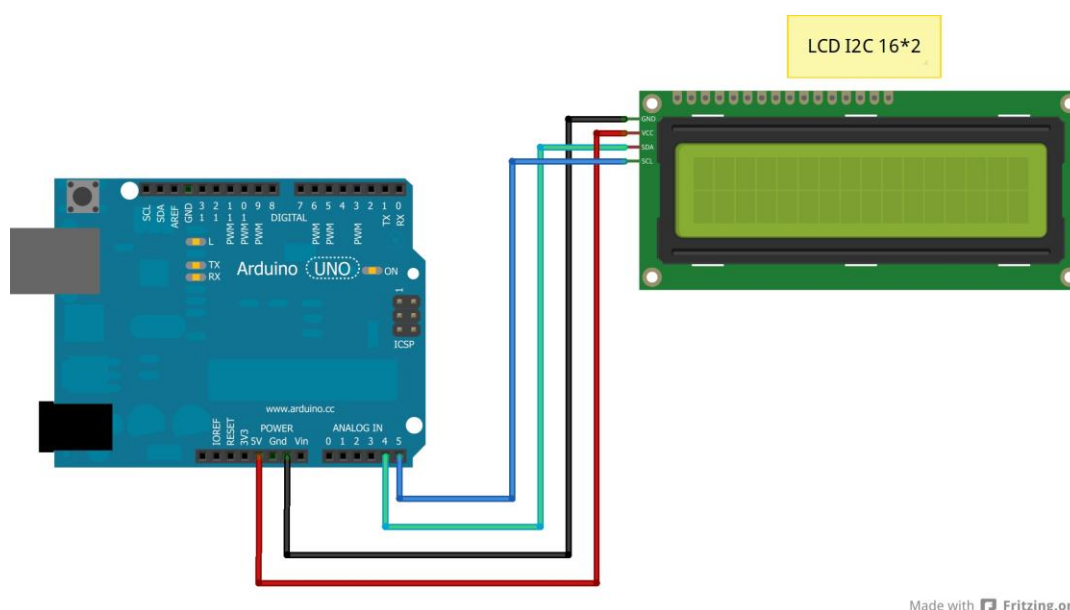


Figure 36 : câblage de LCD I2C 16*2.

III.2.1.4 Commande de relais

Le relais permet de commander tous types d’appareillage à partir d’un Microcontrôleur. Quel que soit le voltage ou le courant, il existe un relais approprié pour chaque cas. Ce sont en effet des interrupteurs presque parfaits, facile à commander et qui assurent un isolement galvanique entre le montage et la charge.

Le module Relay Arduino est connecté directement à la sortie D04 (voir figure ci-dessous).

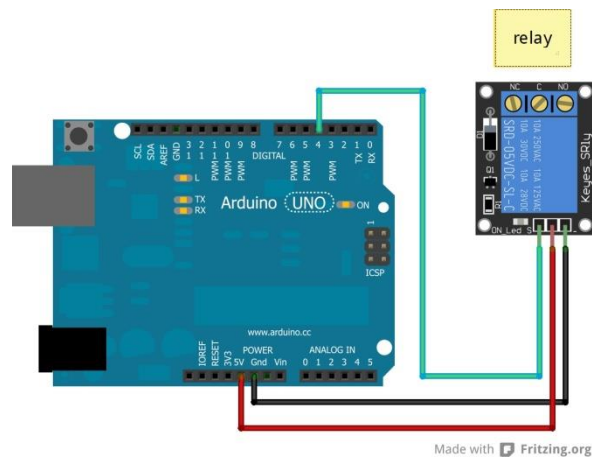


Figure 37 : câblage de Relay.

III.2.1.5 Commande de sirène

La gestion d'une sirène ou une alarme se fait avec un son de sirène de police. Pour ce faire, on joue sur la fréquence d'entrée de buzzer qui est relié directement au proche D10 de sorite de Arduino comme représente la figure 38.

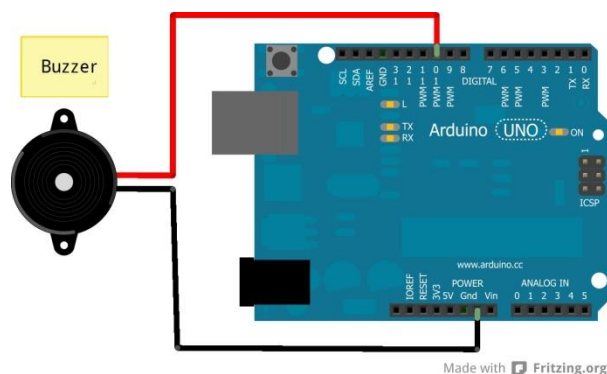


Figure 38 : câblage de Buzzer.

III.2.1.6 Télécommande GSM

La liaison entre le module GSM et le microcontrôleur Arduino est une liaison série asynchrone UART.

Le câblage entre TX et RX de module GSM et le Arduino se fait aux broches de D7 et D8.

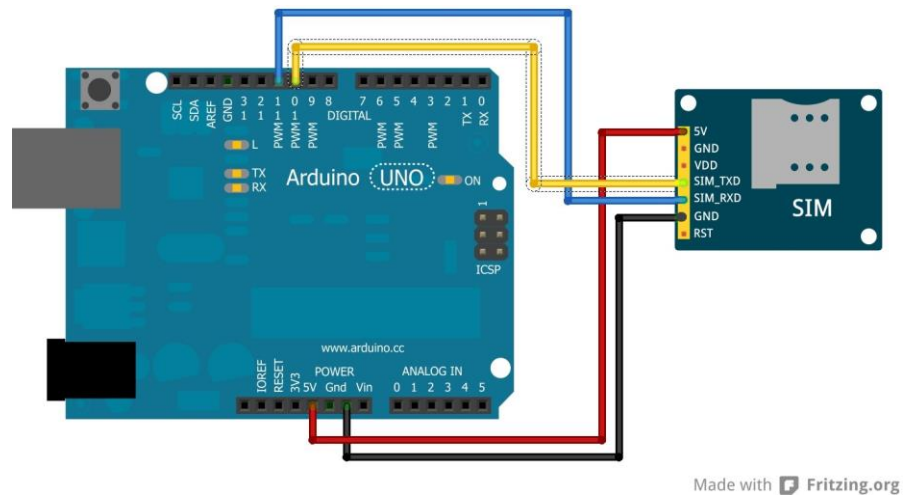


Figure 39 : câblage de module GSM.

Schéma électrique globale

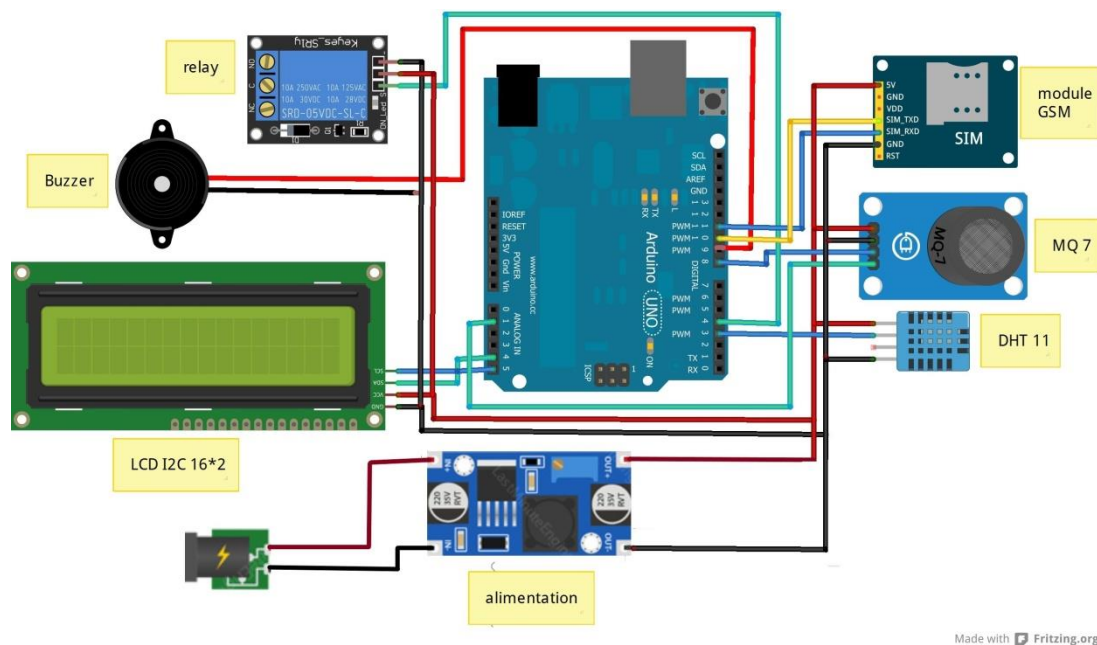


Figure 40 : schéma électrique globale

III.2.2 Travaux expérimentaux

Notre travail expérimental se base en premier temps sur le service monitor de Arduino IDE et puis sur les tests au labo.

III.2.2.1 Moniteur série d'ARDUINO IDE

Arduino a une caractéristique principale, la communication avec notre ordinateur via le port série. Ceci est connu sous le nom de communication série. Parce que l'utilisation de ce port a été quelque peu désaffectée au profit de la technologie USB, Arduino dispose d'un convertisseur Série-USB qui permet à notre carte d'être reconnue par notre ordinateur comme un périphérique connecté à un port COM, même lorsque la connexion physique avec USB.

Arduino IDE nous fournit un outil qui nous permet d'envoyer et d'afficher les données traitées via le port série. Cet outil s'appelle Serial Monitor et se trouve dans le menu Outils, dans l'option "Serial Monitor". C'est le moyen le plus simple d'établir une communication série avec Arduino.

À travers cette fenêtre, vous pouvez envoyer ou recevoir des informations via le port série. Notez que pour ouvrir cette fenêtre, il est nécessaire que notre carte Arduino soit connectée à notre PC via USB.

Pour démarrer la communication série avec Arduino à l'aide du moniteur série, vous devez établir certaines commandes dans l'IDE Arduino, puis les télécharger sur le microcontrôleur.

Dans la fonction de configuration, nous initialisons la communication série avec l'instruction `Serial.begin(9600)`. Le 9600 indique le débit en bauds ou le nombre de bauds que le port série gérera. Le Baud est défini comme une unité de mesure, utilisée dans les télécommunications, qui représente le nombre de symboles par seconde dans un support de transmission, analogique ou numérique.

Pour nos besoins, nous utiliserons couramment une vitesse de symbole de 9600. Chaque fois que nous allons communiquer avec Arduino via un port série, nous devons appeler l'instruction `Serial.begin(9600)`.

Si nous téléchargeons le code de notre programme sur Arduino, nous pouvons observer le comportement de notre algorithme via le moniteur série avec l'appel de la fonction `Serial.println`.

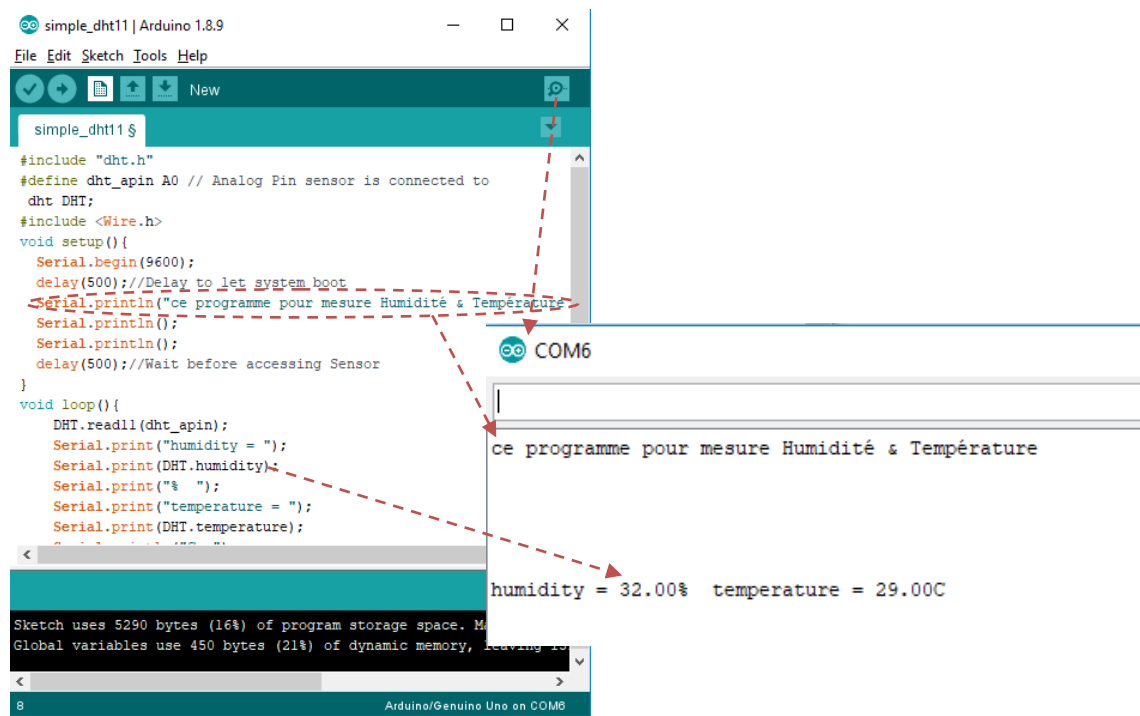


Figure 41 : Serial Monitor.

III.3 Partie software

Dans cette partie, on va expliquer l'utilisation de l'application et comment faire saisir les informations nécessaires, cela pour s'assurer que l'application fonctionne correctement.

Interface principale

Cette interface présente quelques informations et les fonctions disponibles dans l'application, après un clic sur le bouton « SETTINGS », l'interface de réglage apparaîtra sur l'écran pour saisir les champs nécessaires



Figure 42 : interface principale.

Interface de réglages <SETTINGS>

A partir de cette Interface, l'utilisateur peut ajouter les numéros de téléphone mobile d'appareille CO detector, Admin, Helpers et Firemen. En appuyant sur le bouton enregistrer « Save » pour sauvegarder tous les numéros dans l'application et un sms envoyer vers le système d'alarme qui contient tous les numéros téléphones pour la configuration de système.

L'application retour au menu principal.



Figure 43 : interface de paramètre.

Après toutes les procédures de configuration, si l'administrateur a besoin de contrôler les appareils domestique qui se connectent à notre système en appuyant sur le bouton action pour afficher l'interface d'action

Interface d'action

L'interface d'action présentée dans la Figure 44 vous permet de contrôler les appareils connecté.



Figure 44 : Interface d'action.

Confirmation

Après avoir cliqué sur le bouton Effacer les paramètres<CLEAR SETTINGS>, un message apparaît. Pour effacer les paramètres, on va cliquer sur < YES >.

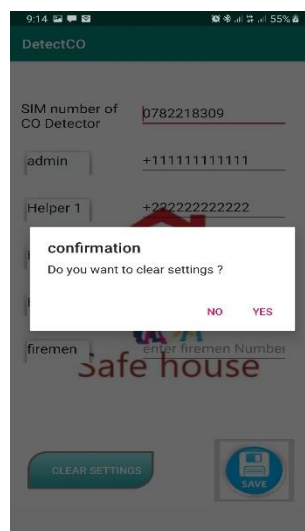


Figure 45 : Confirmation.

Message reçu

Si vous voulez voir les valeurs de température, humidité et la concentration de gaz CO, il vous suffit de cliquer sur l'icône d'appel téléphonique ou l'icône de message.

Ceci est un exemple pour un état recevoir.



Figure 46 : exemple d'affichage de l'état.

Conclusion

Nous avons présenté dans ce chapitre, les phases de conception et réalisation de notre carte électronique de son côté Hardware : le choix du matériel et les considérations pratiques, et du côté software : le mode d'emploi d'application Android. L'expérience menée durant ce chapitre nous a permis d'acquérir plusieurs concepts, et a permis la réalisation d'un système efficace pour la détection, la mesure de concentration de gaz CO, de prévention et d'alarme à base d'Arduino.

Conclusion générale

L'utilisation du gaz dans la vie quotidienne est primordiale. Néanmoins, il présente un risque dangereux sur la vie de l'être humain en cas de fuite, en particulier le monoxyde de carbone reste la première cause d'intoxication domestique mortelle en Algérie.

Dans ce contexte, nous avons réalisé un système d'alarme intelligent pour la détection et la mesure de concentration de gaz toxiques (monoxyde de carbone et méthane), ainsi que pour la prévention et l'alerte à distance. Ce système peut être aussi utilisé dans les environnements domestiques et industriels.

Dans ce cadre, nous avons essayé de développer un système, qui permet aux utilisateurs d'éviter de tomber dans des cas d'accidentés d'intoxication de gaz monoxyde de carbone CO. Ce système vise à surveiller une maison à distance tout en réalisant un détecteur CO conjointement avec les mesures de température et d'humidité. Ce détecteur fonctionne parallèlement avec un module GSM pour l'envoi et la réception des SMS et des appels à partir d'une application installée d'un smart phone.

Dans le cadre de ce projet de fin d'études, nous avons eu l'occasion d'avoir beaucoup de savoir-faire et d'étudier les méthodes d'envoi et de réception des messages SMS, et l'entité intervenante dans la transmission de données qui est le module GSM avec toutes ses particularités. En plus, on avait l'occasion de maîtriser le soft Android studio pour développer l'application. Cela, sans oublier la bonne expérience avec le cerveau de notre carte électronique qui est le microcontrôleur Arduino avec lequel différents composants électroniques sont impliqués.

Une telle réalisation n'est pas dénuée de difficultés. Il est à noter que nous avons confrontés plusieurs problèmes surtout dans la partie de la connexion de notre module GSM avec le réseau mobile, sans citer le problème de l'indisponibilité de ce module en Algérie. Un vrai défi a été confronté dans ce travail qui est la difficulté de création du gaz CO pour les tests expérimentaux. Cependant, on peut dire que malgré ces difficultés et défis, les résultats obtenus à travers cette étude qu'ils soient pratiques ou théoriques, permettent d'ouvrir la porte à d'autres études. Nous espérons que ce mémoire sera une bonne référence aux personnes désirant développer et réaliser des projets et systèmes à distance basé sur une carte Arduino.

Finalement, sous la lumière de tout cela, l'objectif que nous avons fixé au départ est atteint. Néanmoins, nous regrettons de ne pas avoir eu assez de temps pour perfectionner notre projet, c'est à dire rajouter de nouvelles fonctions ou bien de le mettre sur un support finalisé.

Annexe

TECHNICAL DATA

MQ-7 GAS SENSOR

FEATURES

- * High sensitivity to carbon monoxide
- * Stable and long life

APPLICATION

They are used in gas detecting equipment for carbon monoxide(CO) in family and industry or car.

SPECIFICATIONS

A. Standard work condition

Symbol	Parameter name	Technical condition	Remark
Vc	circuit voltage	$5V \pm 0.1$	Ac or Dc
VH (H)	Heating voltage (high)	$5V \pm 0.1$	Ac or Dc
VH (L)	Heating voltage (low)	$1.4V \pm 0.1$	Ac or Dc
RL	Load resistance	Can adjust	
RH	Heating resistance	$33 \Omega \pm 5\%$	Room temperature
TH (H)	Heating time (high)	60 ± 1 seconds	
TH (L)	Heating time (low)	90 ± 1 seconds	
PH	Heating consumption	About 350mW	

b. Environment conditions

Symbol	Parameters	Technical conditions	Remark
Tao	Using temperature	-20℃-50℃	
Tas	Storage temperature	-20℃-50℃	Advice using scope
RH	Relative humidity	Less than 95%RH	
O ₂	Oxygen concentration	21%(stand condition) the oxygen concentration can affect the sensitivity characteristic	Minimum value is over 2%

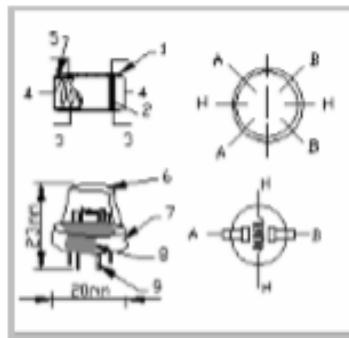
c. Sensitivity characteristic

symbol	Parameters	Technical parameters	Remark
Rs	Surface resistance Of sensitive body	2-20k	In 100ppm Carbon Monoxide
a (300/100ppm)	Concentration slope rate	Less than 0.5	Rs (300ppm)/Rs(100ppm)
Standard working condition	Temperature -20℃ $\pm$ 2℃ relative humidity 65% $\pm$ 5% RL:10K Ω $\pm$ 5% Vc:5V $\pm$ 0.1V VH:5V $\pm$ 0.1V VH:1.4V $\pm$ 0.1V		
Preheat time	No less than 48 hours	Detecting range: 20ppm-2000ppm carbon monoxide	

D. Structure and configuration, basic measuring circuit

Structure and configuration of MQ-7 gas sensor is shown as Fig. 1 (Configuration A or B), sensor composed by micro AL₂O₃ ceramic tube, Tin Dioxide (SnO₂) sensitive layer, measuring electrode and heater are fixed into a crust made by plastic and stainless steel net. The heater provides necessary work conditions for work of sensitive components. The enveloped MQ-7 have

6 pin ,4 of them are used to fetch signals, and other 2 are used for providing heating current.



Parts	Materials
1 Gas sensing layer	SnO ₂
2 Electrode	Au
3 Electrode line	Pt
4 Heater coil	Ni-Cr alloy
5 Tubular ceramic	Al ₂ O ₃
6 Anti-explosion network	Stainless steel gauze (SUS316 100-mesh)
7 Clamp ring	Copper plating Ni
8 Resin base	Bakelite
9 Tube Pin	Copper plating Ni

Fig.1

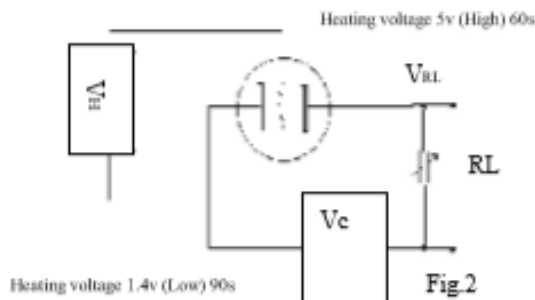
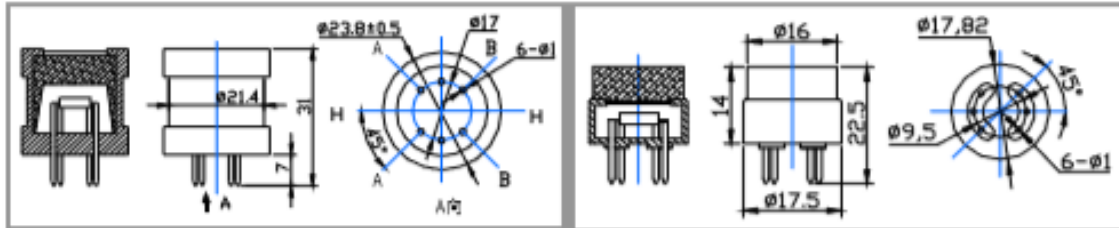


Fig.2

Standard circuit:

As shown in Fig 2, standard measuring circuit of MQ-7 sensitive components consists of 2 parts. one is heating circuit having time control function (the high voltage and the low voltage work circularly). The second is the signal output circuit, it can accurately respond changes of surface resistance of the sensor.

Electric parameter measurement circuit is shown as Fig.2

E. Sensitivity characteristic curve

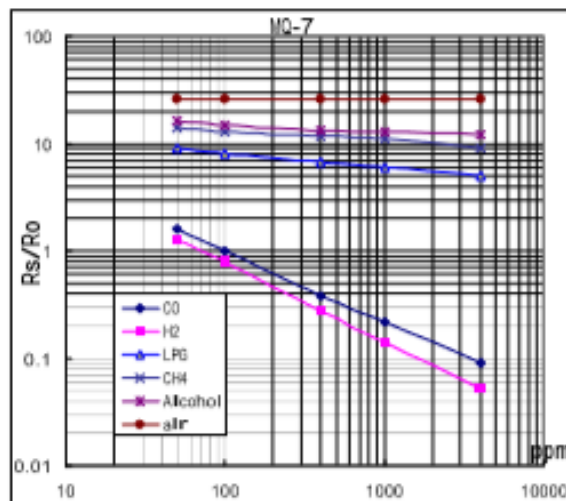


Fig.3 sensitivity characteristics of the MQ-7

Fig.3 is shows the typical sensitivity characteristics of the MQ-7 for several gases.

in their: Temp: 20℃、

Humidity: 65%、

O₂ concentration 21%

RL=10kΩ

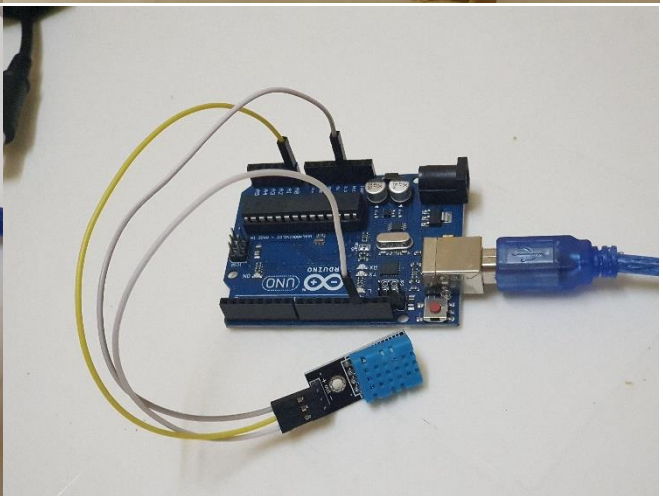
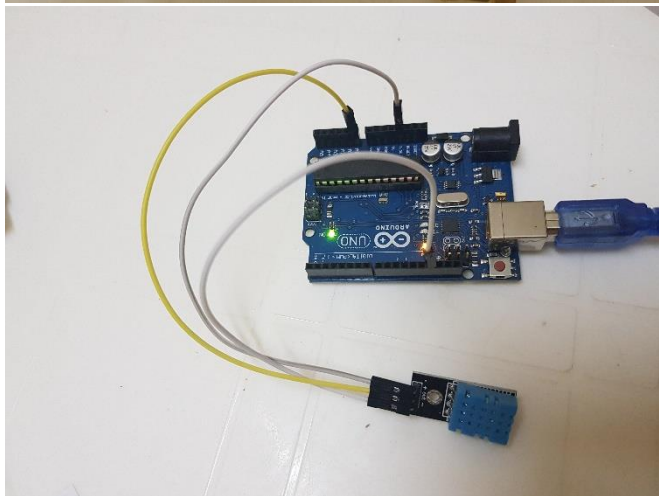
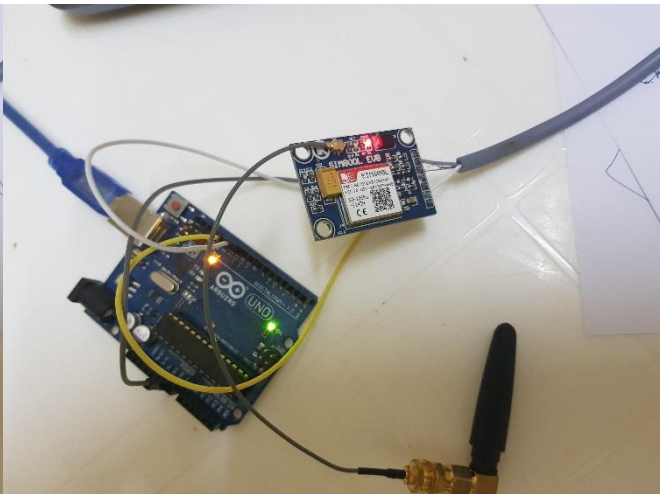
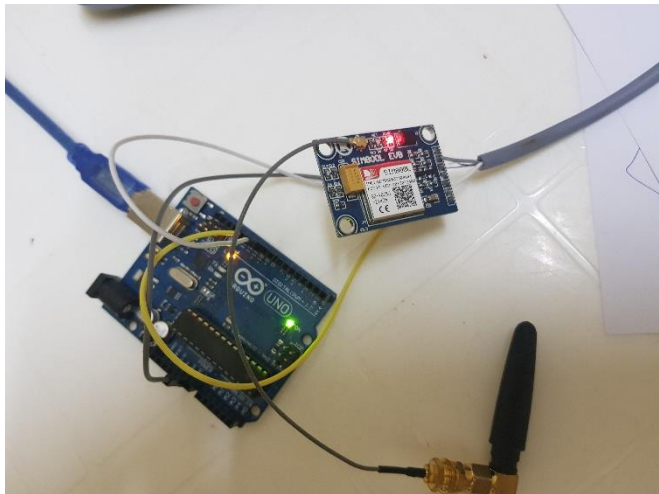
Ro: sensor resistance at 100ppm

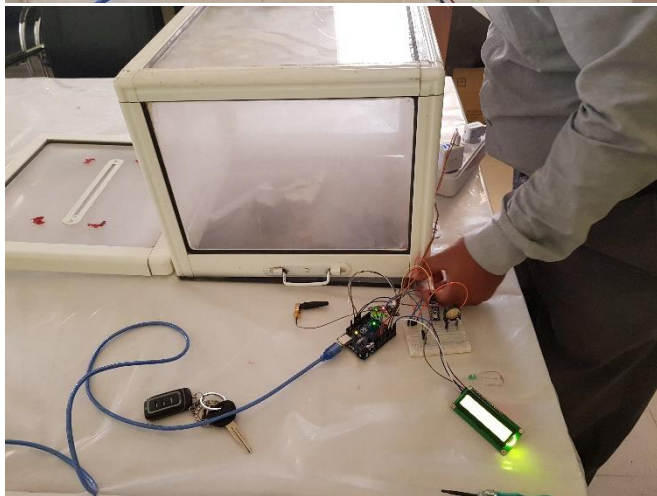
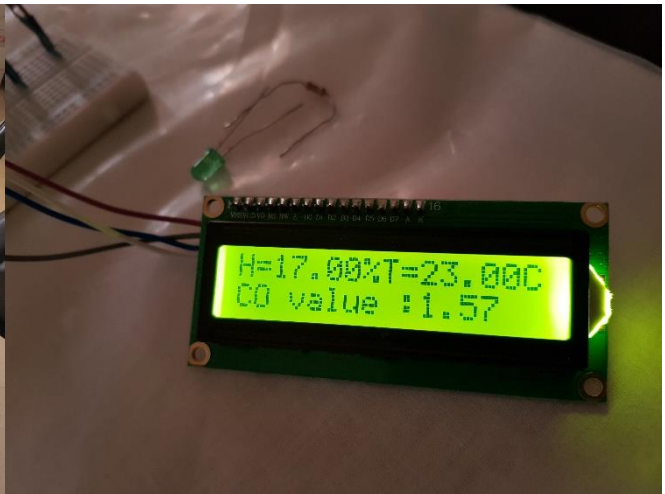
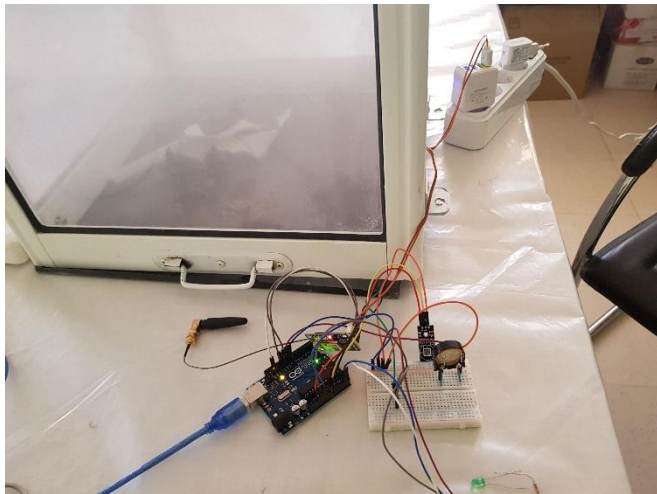
CO in the clean air.

Rs: sensor resistance at various concentrations of gases.



Symptômes et effets sur la sante du CO : intoxications aigue et chronique
(d`apres l`association puor la prevention de la pollution atmospherique www.appa.asso.fr)





List de prix

L'ARTICLE	PRIX
ARDUINO	3500.00
MODULE GSM 900L	6500.00
MODULE DHT11	350.00
MODULE MQ7	700.00
MODULE RELAY	200.00
AFFICHEUR LCD I2C	2000.00

Références

- [01] Erik Bartmann, le grand livre d'arduino 3^e édition. EYROLLES, 2017.
- [02] https://zestedesavoir.com/tutoriels/686/arduino-premiers-pas-en-informatique-embarquee/744_la-communication-avec-arduino/3426_generalites-sur-la-voie-serie/
- [03] <https://www.interieur.gouv.fr/A-votre-service/Ma-securite/Conseils-pratiques/A-votre-domicile/Monoxyde-de-carbone-attention-danger>
- [04] <http://fr.hobbytronics.co.uk/mq7-carbon-monoxide-sensor>
- [05] <https://www.sparkfun.com/datasheets/Sensors/Biometric/MQ-7.pdf>
- [06] <https://lastminuteengineers.com/dht11-dht22-arduino-tutorial/>
- [07] <http://www.interieur.gov.dz/index.php/fr/mes-d%C3%A9marches-administratives/activit%C3%A9s-r%C3%A9glement%C3%A9es/equipements-sensibles>
- [08] P. GUEULLE. *Téléphones portables et PC 3e édition*. Dunod, 2006.
- [09] https://fr.wikipedia.org/wiki/Environnement_de_d%C3%A9veloppement, consulté le 03/05/2019.
- [10] <https://android-studio.fr.uptodown.com/windows>, consulté le 03/05/2019.
- [11] <https://www.techno-science.net/definition/517.html> , consulté le 03/05/2019.
- [12] Medileh S, Cours BDD, Université El-Oued, 2013-2014.
- [13] www.ipeti.forumpro.fr/t21-definition-de-langage-java-java-script, consulté le 03/05/2019.