

الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne démocratique et populaire

وزارة التعليم العالي والبحث العلمي
Ministère de l'enseignement supérieur et de la recherche scientifique

جامعة سعد دحلب البلدة
Université SAAD DAHLAB de BLIDA

كلية التكنولوجيا
Faculté de Technologie

قسم الإلكترونيك
Département d'Électronique



Mémoire de Master

Filière Électronique
Spécialité Instrumentation

présenté par

MATARI RANIA

&

RHIMI YOUSRA

Kit didactique à base de la carte Arduino Uno

Proposé par : Mr Yekhlef Farid

Année Universitaire 2019-2020

Nous ne saurons présenter ce PFE sans avoir à remercier DIEU tout puissant de nous avoir prodiguées l'énergie pour l'élaboration de ce modeste travail.

Nous remercions tout particulièrement Mr F. Yekhlef de nous avoir offert l'occasion de nous former dans ce domaine, nous exprimons notre profonde gratitude pour l'attention portée à l'élaboration de ce mémoire, son aide précieuse, ses conseils, son soutien permanent et pour tes efforts.

Nous remercions les plus vifs s'adressent aussi aux messieurs le président et les membres de jury qui ont bien voulu nous faire l'honneur d'examiner et d'évaluer notre travail.

Que tous ceux qui ont contribué de près ou de loin à réalisation du présent travail.

Ce travail modeste est dédié :

À nos chères parents, Que nous les souhaites la bonne santé et longue vie À tous nos proches de la famille Matari et Rhimi,

Et plus particulièrement, nos sœurs et frères et neveux tout à son nom.

À tous nos chers amis, spécialement à Randa et nos collègues de l'Université de Saad Dahleb.

Et à tous ce qui nous ont enseigné au long de notre vie Scolaire.

Et à tous ceux qui ont contribué de près ou de loin pour que ce projet soit possible, nous vous disons merci.

ملخص:

هذه المذكرة عبارة عن دراسة لعدة مشاريع صغيرة باستخدام لوحة اردوينو والعديد من المستشعرات، المشروع الأول هو العصا البيضاء الذكية، والثاني هو كاشف الغاز، والثالث محطة الطقس، والأخير هو سلة المهملات الذكية.

لكل مشروع أهميته الخاصة، لكن الهدف الرئيسي من هذه المشاريع هو تسهيل الحياة اليومية وتلبية احتياجات الناس وحمايتهم أيضاً (عوائق، تسربات، ضربة شمس، جراثيم...).

كلمات المفاتيح: اردوينو، مستشعر، مشروع

Résumé :

Ce mémoire est une étude de plusieurs mini projets à l'aide d'une carte Arduino et plusieurs capteurs, Le premier projet est la canne blanche intelligente, le deuxième est le détecteur des gaz, le troisième est la station météo, le dernier est la poubelle intelligente.

Chaque projet à sa propre importance, mais le but principale de ces projets est pour faciliter la vie quotidienne et pour répondre aux besoins des gens aussi les protéger (obstacles, fuites, coup de chaleur, microbes...).

Mots clés : arduino, capteur, projet

Abstract :

This thesis is a study of several mini projects using an Arduino board and several sensors, The first project is the smart white cane, the second is the gas detector, the third is the weather station, the last is the smart trash can.

Each project has its own importance, but the main goal of these projects is to make everyday life easier and to meet people's needs and also protect them (obstacles, leaks, heatstroke, germs...).

Keywords : arduino,sensor,project

Listes des acronymes et abréviations

E/S : entré/sortie

PWM : Pulse Width Modulation (La modulation de largeur d'impulsion)

SRAM : static random access memory (mémoire vive statique)

EEPROM : electrically erasable programmable read-only memory (mémoire morte programmable effaçable électriquement)

SPI : Serial Peripheral Interface (interface périphérique série)

AREF : analog reference (référence analogique)

LCD : liquid crystal display

GND : Ground

Max : Maximum

Min : Minimum

I2C : Inter-Integrated Circuit

PCB : printed circuit board

NTC : Negative Temperature Coefficient

TCXO : Temperature Compensated Crystal Oscillator

Table des matières :

Introduction générale	1
Chapitre I : Généralités sur l'arduino et les composants utilisés	
Introduction.....	4
I/Généralités sur l'arduino.....	4
I.1/QUE VEUT DIRE OPEN SOURCE ?.....	4
I.2/Qui sont les utilisateurs ?.....	5
I.3/Comment ça marche ?.....	5
I.4/L'interface du logiciel Arduino se présente de la façon suivante.....	6
I.5/Quelle est l'utilité de l'arduino ?.....	6
I.6/Les types de cartes arduino	6
I.7/Arduino Uno.....	9
I.7.1/Présentation de la carte	9
I.7.2/Les caractéristiques	10
I.7.3/Les éléments de la carte	10
I.7.4/Avantage de la carte arduino.....	14
➤ Les composants	14
II/Les capteurs	14
✓ Définition principale du capteur	14
II.1/Les capteurs MQ	14
a/Définition	14
b/schéma de principe d'un capteur de gaz dans un module de capteur de gaz.....	15
c/Caractéristiques des capteurs MQ	15
d/Schéma électrique de MQ	16
e/Applications des MQ	16
II.1.1/Capteur Mq2	16
II.1.1.2/Qu'est-ce que le capteur de gaz MQ2 ?.....	16
II.1.1.3/Structure interne du capteur de gaz MQ2	17
II.1.1.4/Principe de fonctionnement de mq2	17
II.1.1.5/Applications de mq2.....	18
II.1.1.6/Configuration des broches	18
II.1.1.7/Courbe caractéristique de sensibilité	19
II.1.1.8/Câblage de mq2 avec arduino	19
II.1.2/Capteur mq7.....	20
II.1.2.1/Qu'est-ce que le capteur de gaz MQ7 ?.....	20
II.1.2.2/Caractéristiques.....	20
II.1.2.3/Données techniques.....	20
II.1.2.4/Configuration des broches	21
II.1.2.5/Principe de fonctionnement de mq7.....	21
II.1.2.6/Présentation du capteur de monoxyde de carbone MQ-7.....	22
II.1.2.7/Applications	22

II.1.2.8/Câblage de mq7 avec arduino	22
II.1.3/Capteur dht11.....	23
II.1.3.1/Définition	23
II.1.3.2/Différence entre le capteur DHT11 et le module.....	24
II.1.3.3/Principe de fonctionnement de dht11.....	24
II.1.3.4/Identification et configuration des broches.....	24
II.1.3.5/Applications de dht11	25
II.1.3.6/Câblage de dht11 avec 4 broches.....	25
II.1.3.7/Câblage de dht11A avec 3 broches.....	25
II.1.4/Capteur BMP280.....	26
II.1.4.1/Définition	26
II.1.4.2/Caractéristiques.....	26
II.1.4.3/Puissance requise	27
II.1.4.4/Brochage du capteur BME280	27
II.1.4.5/Câblage du module BME280 à Arduino.....	28
II.1.5/Le DS3231.....	28
II.1.5.1/Définition.....	28
II.1.5.2/Principe de fonctionnement	29
II.1.5.3/CARACTÉRISTIQUES.....	29
II.1.5.4/Les broches du ds3231.....	30
II.1.5.5/Applications	30
II.1.5.6/Câblage du module RTC DS3231 à Arduino UNO.....	31
II.1.6/Capteur à ultrason.....	31
II.1.6.1/Définition.....	31
II.1.6.2/Principe de fonctionnement	32
II.1.6.3/Calcul.....	32
II.1.6.4/Spécification.....	32
II.1.6.5/Applications.....	33
II.1.6.6/Câblage avec la carte arduino	33
III/VENTILATEUR DC 12V.....	33
III.1/Définition	33
III.2/Câblage de ventilateur avec arduino	34
IV/Diode électroluminescente (LED)	34
IV.1/Définition.....	35
IV.2/Les avantages des LED	35
IV.3/Les applications des leds	35
IV.4/Câblage de led avec arduino	36
V/Lcd 16*2.....	36
V.1/Description de Lcd 16*2	36
V.2/Diagramme de brochage LCD 16X2.....	37
V.3/Caractéristiques du module LCD 16 x 2.....	38
V.4/Les avantages de lcd	38
V.5/Câblage de lcd avec arduino.....	38
VI/ Les résistances	38
VI.1/ Définition.....	38
VI.2/Les types de résistances.....	39
VI.3/code des couleurs de résistance	39
VI.4/APPLICATIONS	40
VI.5Câblage des résistances avec arduino.....	41

VII/Le Buzzer.....	42
VII/Définition.....	42
VII.2/Caractéristiques et spécifications du buzzer.....	42
VII.3Les types du buzzers	43
VII.4/Les différences entre piézo et électromécanique	43
VII.5/Configuration des broches du buzzer	44
VII.6/Applications des buzzers	44
VII.7/Câblage du buzzer avec arduino	45
VIII/LE Servomoteur.....	45
VIII.1/Définition	45
VIII.2/le fonctionnement d'un servomoteur	46
VIII.3/Contrôle des servomoteurs.....	46
VIII.4/Types de servomoteurs	47
VIII.5/Applications du servomoteur.....	47
VIII.6/Câblage de servomoteur avec arduino.....	47
IX/Conclusion du chapitre	48

Chapitre II : fonctionnements et importances des projets

Introduction	50
I/ canne blanche intelligente	50
I.1/Introduction.....	50
I.2/Principe de fonctionnement	51
I.3/L'importance du projet	51
I.4/Conclusion	52
II/ détecteur de gaz	52
II.1/ Introduction.....	52
II.2/L'effet du gaz butane au corps humain	52
II.3/L'effet du gaz CO au corps humain	53
II.4/Principe de fonctionnement	54
II.5/L'importance de détecteur de fuites des gazes	54
II.6/L'installation le détecteur de gaz.....	55
II.7/Conclusion.....	55
III/La station météo.....	56
III.1/Introduction.....	56
III.2/La température	56
III.3/L'humidité.....	56
III.4/La pression.....	57
III.5/Principe de fonctionnement d'une station météo.....	57
III.6/L'importance d'une station météo	57
III.7/Conclusion	58
IV/La poubelle intelligente.....	58
IV.1/Introduction.....	58
IV.2/Principe de fonctionnement d'une poubelle intelligente	58
IV.3/L'importance d'une poubelle intelligente	59
IV.4/Conclusion	59
V/Conclusion du chapitre	59

Chapitre III : Partie réalisations

Introduction	61
I/Canne blanche intelligente	61
I.1/Réalisation du projet	61
I.2/Simulation avec proteus.....	61

I.3/Programme arduino	61
II/Détecteur de gaz	63
II.1/Réalisation du projet	63
II.2/Pourquoi nous avons utilisé un transistor et une diode avec le ventilateur ?.....	63
II.3/Simulation avec proteus	63
II.4/Programme arduino	64
III/Station météo.....	66
III.1/Réalisation du projet.....	66
III.2/Câblage fritzing	66
III.3/Programme arduino	66
IV/Poubelle intelligente	77
IV.1/Réalisation du projet.....	77
IV.2/Simulation proteus	77
IV.3/Programme arduino	78
Conclusion du chapitre	79
Conclusion generale.....	80
Bibliographie.....	81

Liste de figures :

Figure 1:interface d'arduino	5
Figure 2:les boutons d'arduino.	6
Figure 3:schéma block de la carte arduino uno.	10
Figure 4:description de la carte arduino uno.	11
Figure 5:microcontrôleur atmega328.	11
Figure 6:broche digitale	13
Figure 7: schéma de principe d'un capteur de gaz dans un module de capteur de gaz	15
Figure 8:schéma électrique de MQ.	16
Figure 9:MQ2 capteur de gaz.	17
Figure 10:les caractéristiques de sensibilité typique du MQ2 pour plusieurs gaz.	19
Figure 11:cablage de MQ2 avec arduino.	20
Figure 12:configuration de broche.	21
Figure 13:les caractéristiques de sensibilité typique du MQ7 pour plusieurs gaz.	22
Figure 14:cablage de mq7 avec arduino.	23
Figure 15:capteur d'humidité et de température.	23
Figure 16:configuration de broche.	24
Figure 17:cablage de dht11 à 4 broches avec arduino.	25
Figure 18:cablage de dht11 à 3 broches avec arduino.	26
Figure 19:capteur BMP280.	27
Figure 20:broche de BMP280.	28
Figure 21:cablage du module bmp280 avec arduino.	28
Figure 22:module DS3231.	29
Figure 23 : Les broches de module DS3231.	30
Figure 24:cablage de module DS3231 avec l'arduino.	31
Figure 25:capteur à ultrason.	31
Figure 26:mise en marche de capteur à ultrason.	32
Figure 27:cablage de capteur à ultrason avec l' arduino.	33
Figure 28:ventilateur DC.	34
Figure 29 : câblage de ventilateur avec l'arduino.	34
Figure 30:diode électroluminescente.	35
Figure 31:câblage de la diode avec l'arduino.	36
Figure 32:afficheur LCD.	36
Figure 33:les broches de l'afficheur LCD.	37
Figure 34:cablage de l'afficheur avec arduino.	38
Figure 35:resistance.	39
Figure 36:résistance fixe.	Error! Bookmark not defined.
Figure 37:resistance variable.	40
Figure 38:code de couleurs	40
Figure 39:cablage des résistances avec arduino.	41
Figure 40:buzzer.	42

Figure 41: cablage de buzzer avec l'arduino. -----	44
Figure 42: servomoteur -----	45
Figure 43: cablage du servomoteur avec arduino. -----	47
Figure 44: mise en marche de capteur à ultrason. -----	51

Introduction générale

En 2005, à Ivrea, en Italie, un projet a été lancé pour fabriquer un dispositif de contrôle de projets de conception interactive construits par les étudiants, moins coûteux que les autres systèmes de prototypage disponibles à l'époque. Ce projet est nommé arduino créé par Hernando Barragán, Massimo Banzi et David Cuartielles, un appareil programmable facile à utiliser pour des projets de conception d'art interactif, basé sur le câblage. Avant longtemps, Gianluca Martino et Tom Igoe ont rejoint le projet, et les cinq sont connus comme les fondateurs originaux d'Arduino. Ils voulaient un appareil simple, facile à connecter à diverses choses (comme les relais, les moteurs et les capteurs) et facile à programmer. Les étudiants et les artistes ne sont pas connus pour avoir beaucoup d'argent de rechange pour cela ils ont sélectionné la famille AVR de microcontrôleurs 8 bits conçu une carte de circuit imprimé autonome avec des connexions faciles à utiliser et ont tout regroupé dans un développement intégré simple environnement (IDE) qui utilisé des programmes appelés «croquis». Le résultat a été l'Arduino

Aujourd'hui, nous sommes à l'ère de la technologie et nous avons besoin d'un projet qui nous facilite la vie ,pour cela ,dans cet mémoire on a étudié et réalisées plusieurs mini projet à base d'arduino qui ont plusieurs avantages ainsi utilisées dans les propriétés publiques ou privés tel que les entreprises et les habitations aussi dans les rues et la vie quotidienne ..

Ce mémoire décrit l'ensemble de nos travaux et il est constitué d'une introduction générale, et trois chapitres avec des introductions pour chaque chapitre et des conclusions.

Le premier chapitre est consacré à la description des composants de nos mini projets avec leur fonctionnement, caractéristiques et leurs câblages avec arduino.

Le deuxième chapitre est consacré pour décrire les projets en détails avec leur fonctionnement et l'importance de chaque un.

Le dernier chapitre est consacré à la programmation et la simulation proteus et fritzing de nos projets.

Enfin, une conclusion générale qui résume l'ensemble de travaux de ce mémoire.

Objectifs de ce mémoire :

- Réaliser plusieurs projets à base d'arduino.
- Utiliser des composants adaptés et utilisables sous toutes circonstances.
- aider les étudiants à programmer et simuler des projets facilement.

Chapitre1 :

Généralités sur l'arduino et les composants utilisés

❖ Introduction :

Dans ce chapitre nous allons donner une description détaillée de l'Arduino UNO et les capteurs utilisés avec leur mode de fonctionnement, caractéristiques et applications pour chaque composant utilisé dans les projets et comment peut-on les connecter avec la carte Arduino UNO.

I/Généralités sur l'arduino :

Comment faire des montages électroniques simplement en utilisant un langage de programmation ?

La réponse, c'est le a l'aide de la carte Arduino

Dans la technologie arduino, toutes les pièces électroniques sont intégrées au microcontrôleur. Un panneau qui permet à l'utilisateur de concevoir facilement plusieurs projets sans changer les pièces électroniques de base.

Arduino est le nom d'une série de cartes de prototypage comportant un microcontrôleur qu'on peut facilement programmer au moyen d'un ordinateur. Grâce à ce microcontrôleur hautement performant et facilement programmable, la carte Arduino a révolutionné le mouvement Do It Yourself.

Un microcontrôleur se compose d'une puce sur une carte de circuit imprimé avec capacités de lecture-écriture, mémoire, entrées et sorties.

.Arduino est une plateforme open source utilisée pour la construction de projets électroniques.

L'Arduino est une carte logicielle électronique composée de matériel et de logiciels ou IDE (Integrated Développement Environnement), qui utilise un langage c qui s'exécute sur l'ordinateur, utilisé pour écrire et télécharger du code informatique sur la carte physique.

La carte Arduino se connecte à l'ordinateur via un fil connecté à un support de données USB ou un distributeur, et est alimentée en la connectant à une source d'alimentation externe ou à une batterie électrique séparée.

Le rôle de la carte Arduino est de stocker un programme et de le faire fonctionner.

I.1/QUE VEUT DIRE OPEN SOURCE ?

La base de l'idée de l'arduino était que tout le monde devrait y avoir accès et le développer.

La carte arduino est connue par sa conception et il est possible que tout le monde peut télécharger le schéma de conception et l'étudier et modifier, le reproduire et le vendre.

I.2/ Qui sont les utilisateurs ?

Des « bidouilleurs » dont beaucoup ont des artistes qui ont besoin d'animer leurs œuvres ou de créer des interactions avec elles des connaissances très limitées en électronique. Des étudiants et des élèves (y compris de collèges).

I.3/ Comment ça marche ?

L'arduino est programmé par un langage de programmation similaire à C++. Le programme arduino sur l'ordinateur contient également un ensemble de bibliothèques pour les commandes de pièces jointes électroniques. la structure de programme est comme ceci :

- Une section « setup » 1 seule exécution après Raz.
- Une section « loop » exécutée indéfiniment en boucle.

De très nombreuses bibliothèques logicielles disponibles.

I.4/ L'interface du logiciel Arduino se présente de la façon suivante :

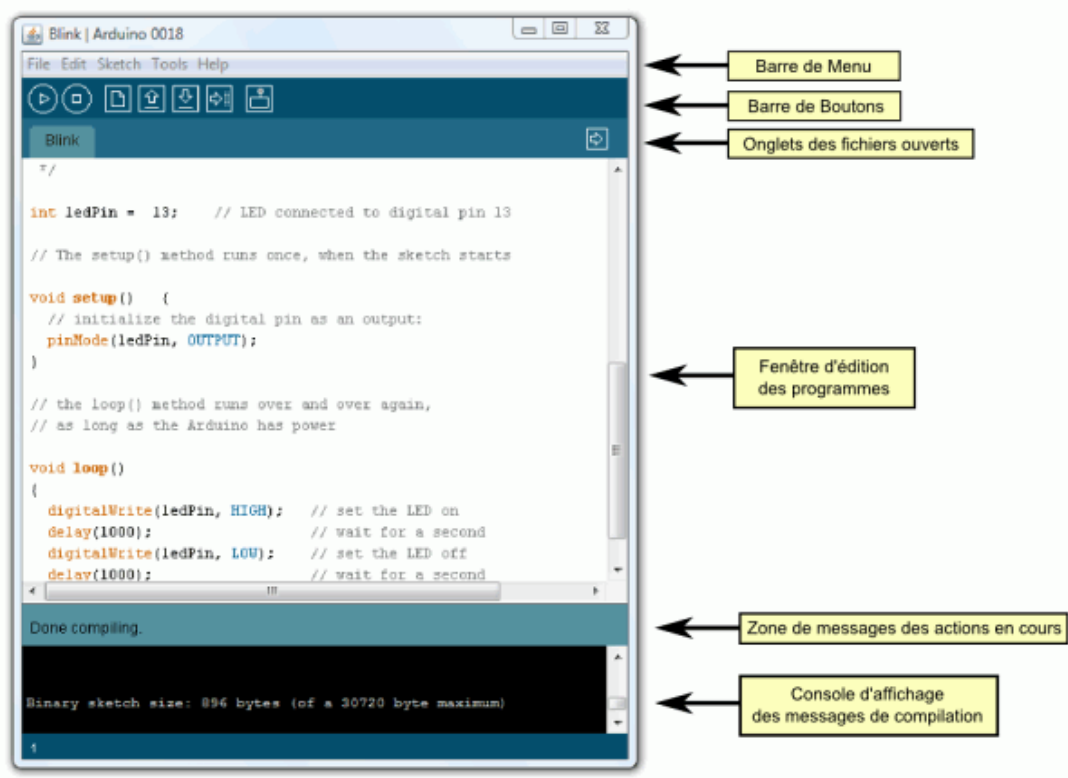


Figure 1: interface d'arduino

- Les boutons arduino :

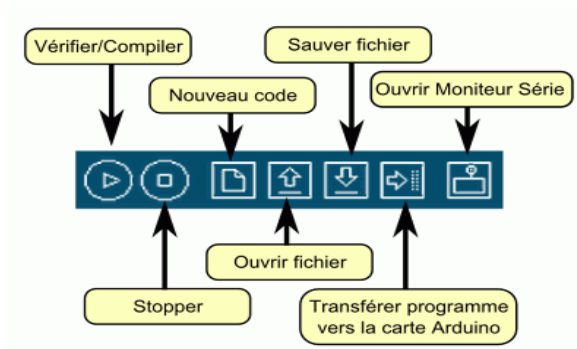


Figure 2:les boutons d'arduino.

I.5/Quelle est l'utilité de l'arduino ?

La carte arduino est devenue aujourd'hui un élément indispensable dans de nombreux dispositifs électroniques. Sa simplicité d'utilisation, l'étendue de ses applications en outre elle est peu couteuse

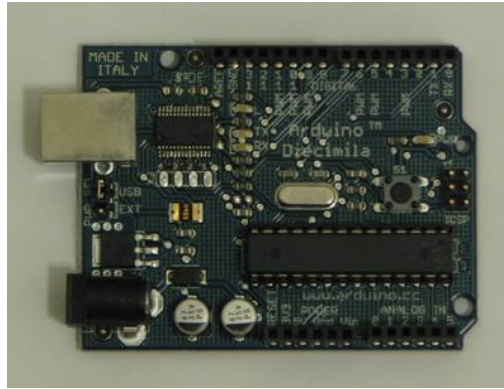
Le système Arduino, nous donne la possibilité d'allier les performances de la programmation à celles de l'électronique. Plus précisément, Le gros avantage de l'électronique programmée c'est qu'elle simplifie grandement les schémas électroniques et par conséquent, le coût de la réalisation, mais aussi la charge de travail à la conception d'une carte électronique. La valeur de la maîtrise de l'Arduino réside dans le fait qu'il donne à son utilisateur la possibilité de contrôler l'environnement qui l'entoure grâce à des capteurs optiques et des logiciels installés sur les panneaux, c'est-à-dire qu'il permet simplement à ses clients de contrôler n'importe quel objet.

Peut être utilisé pour concevoir facilement plusieurs projets intelligents :

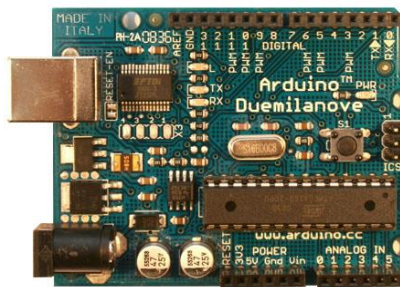
- contrôler les appareils domestiques.
- fabriquer robot.
- faire un jeu de lumières.
- communiquer avec l'ordinateur.
- télécommander un appareil mobile (modélisme).

I.6/Les types de cartes arduino :

Arduino diecimila :



Arduino duemilanove :



Arduino uno :



Arduino Leonardo :



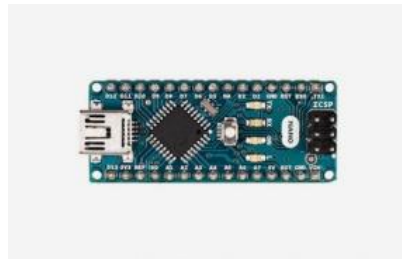
Arduino mega :



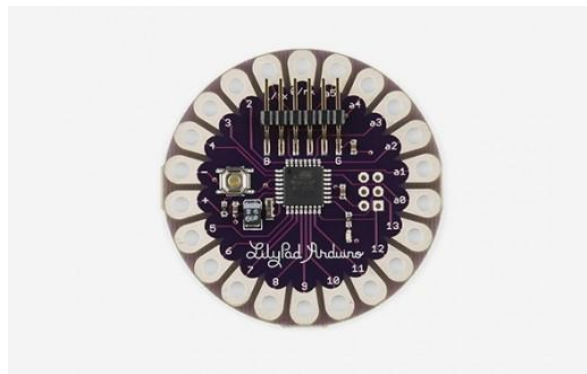
Arduino mega 2560 :



Arduino nano :



Arduino lilypad :



Arduino esplora :



Arduino yun :



arduino due :



- ❖ Arduino est l'une des cartes électroniques les plus répandues et les plus populaires consacrées à l'éducation en raison de l'extrême facilité qu'elle offre soit en termes de programmation, soit en termes d'équipement, et même en termes de disponibilité de plusieurs circuits électriques, programmes et bibliothèques dans divers sujets. Pour cela on a choisi arduino uno pour qu'on réalise notre mémoire.

I.7/Arduino Uno :

I.7.1/Présentation de la carte :

La carte Arduino UNO est une carte basé sur un microcontrôleur ATmega328 programmable permettant de faire fonctionner des composants (moteur, LED...).

La carte Arduino contient une mémoire morte de 1 kilo. Elle est dotée de 14 entrées/sorties digitales (dont 6 peuvent être utilisées en tant que sortie PWM), 6 entrées analogiques et un oscillateur à quartz à 16 MHz, une connexion USB, un bouton de remise à zéro et une prise jack d'alimentation.

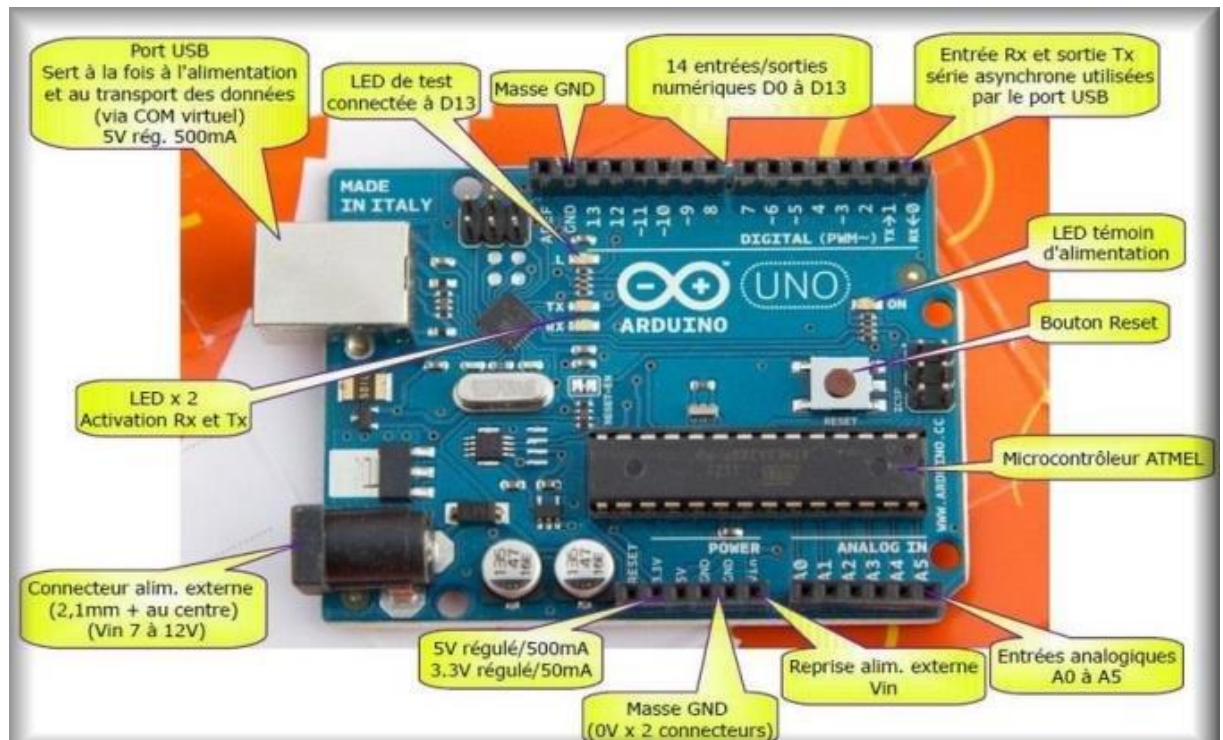


Figure 3:schéma block de la carte arduino uno.

I.7.2/Les caractéristiques :

- Tension de fonctionnement : 5V.
- Tension d'entrée (recommandé) 7-12V.
- Tension d'entrée (limite) 6-20V.
- E / S numériques Pins 14 (dont 6 fournissent la sortie PWM*).
- PWM numérique E / S Pins 6.
- Pins d'entrée analogique 6.
- DC Courant par I O Pin / 20 Ma.
- Courant DC pour 3.3V Pin 50 mA.
- Mémoire flash 32 KB.
- SRAM : 2 KB.
- EEPROM : 1 KB
- Vitesse de l'horloge : 16 MHz.
- Longueur : 68,6 mm.
- Largeur : 53,4 mm.
- Poids : 25g.

I.7.3/Les éléments de la carte :

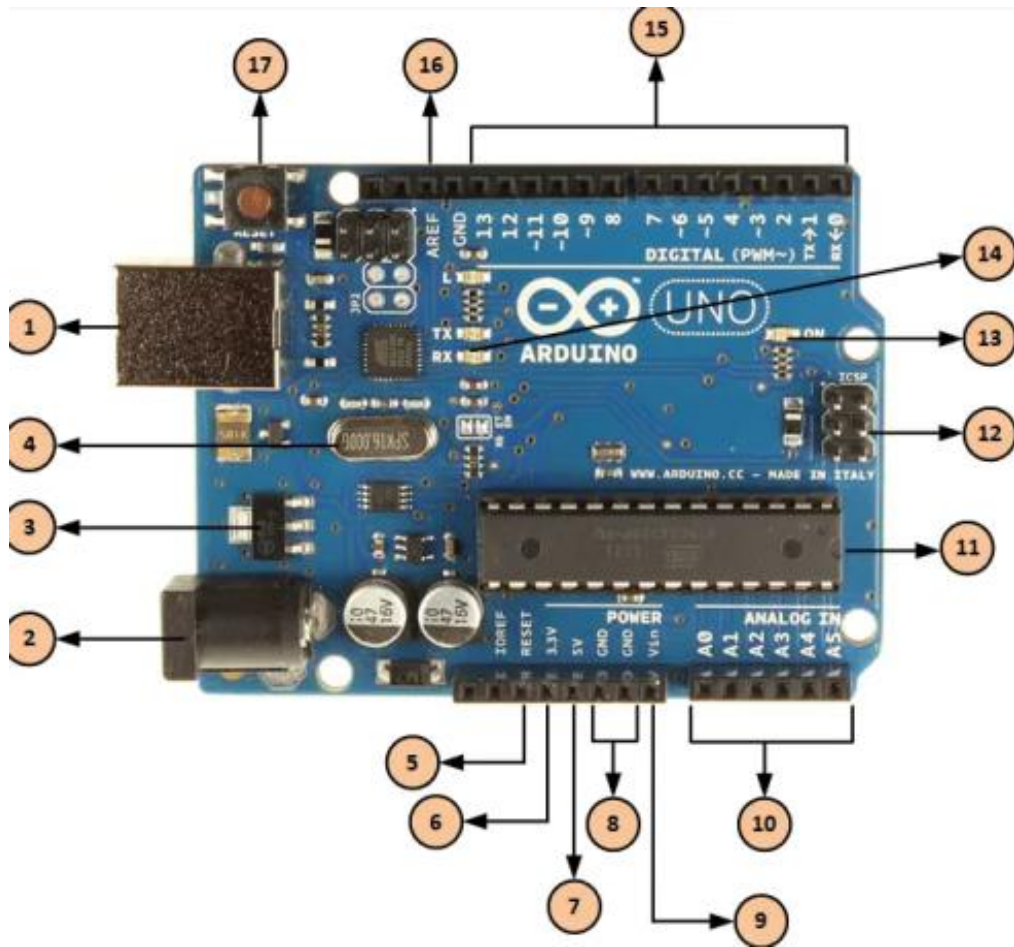


Figure 4: description de la carte arduino uno.

- **Un microcontrôleur(11)** : est un système qui ressemble à un ordinateur : il a une mémoire, un processeur, des interfaces avec le monde extérieur. Les microcontrôleurs sont de faible taille et consomment peu d'énergie, les rendant indispensables dans toute solution d'électronique embarquée (voiture, porte de garage, robots, ...). C'est lui qui va recevoir le programme que nous aurons créé et qui va le stocker dans sa mémoire puis l'exécuter.



Figure 5: microcontrôleur atmega328.

Le microcontrôleur ATmega328 est constitué par un ensemble d'éléments qui sont les mêmes dans la carte mère de l'ordinateur.

❖ L'architecture interne de ce circuit se compose sur :

La mémoire possède 4 types :

- ✓ La mémoire flash : C'est celle qui contiendra le programme à exécuter, elle est effaçable et réinscriptible de 32K.
- ✓ RAM : c'est la mémoire dite "vive", elle va contenir les variables du programme. Elle est dite «volatile» car elle s'efface si on coupe l'alimentation du microcontrôleur. Sa capacité est 22 ko.
- ✓ EEPROM : C'est le disque dur du microcontrôleur. Vous pourrez y enregistrer des infos qui ont besoin de survivre dans le temps, même si la carte doit être arrêtée. Cette mémoire ne s'efface pas lorsque l'on éteint le microcontrôleur ou lorsqu'on le reprogramme.
- ✓ Les registres : c'est un type de mémoire utilisé par le processeur.
- L'alimentation de la carte : L'alimentation de la carte Arduino Uno se fait normalement entre 7V et 12V de courant continu. Cependant, il est possible de faire fonctionner la carte Arduino Uno au maximum entre 6V et 20V. En deca de 6V, la carte n'est plus stable. Au-delà de 20V, le régulateur de tension surchauffe, et peut endommager la carte. L'alimentation de la carte peut se faire à travers le **port USB(1)** lorsqu'il est branché sur l'ordinateur, ou via la connectique d'alimentation avec le **port jack (2)** ou l'entrée d'alimentation.
- Régulateur de tension (3) : La fonction du régulateur de tension (3) 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 quartz (4) : 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 de 16, 000,000 Hertz, soit 16 MHz.
- Les pins d'alimentation sont les suivants :
 - L'entrée d'alimentation (VIN) (9) : permet d'utiliser une broche afin d'alimenter votre carte Arduino en électricité. Il est conseillé d'utiliser une alimentation entre 7V et 12V de courant continu pour ne pas endommager la carte.
 - 5V (7) : Ce pin délivre un 5V régulé par la carte.
 - 3,3V (6) : Une alimentation de 3,3 volts générée par le régulateur. La consommation de courant maximale est de 50 mA.
 - Les prises de terre ou GND(8) : permettent de fermer le circuit.

- Les entrées et sorties :
- Les entrées et sorties numériques (15) :

La carte possède 14 broches numériques (de D0 à D13) sur la carte Uno peut être utilisée comme une entrée ou une sortie, en utilisant les fonctions `pinMode ()` , `digitalWrite ()` , et `digitalRead ()`. Ils fonctionnent à 5 volts. Chaque broche peut fournir ou recevoir 20 mA en état de fonctionnement recommandée.

Un maximum de 40mA est la valeur qui ne doit pas être dépassée sur toutes les broches d'Entrée/Sorties pour éviter des dommages permanents aux microcontrôleurs. Les signaux véhiculés par ces connecteurs sont des signaux logiques compatibles TTL, c'est-à-dire qu'ils ne peuvent prendre que deux états HAUT (5 Volts) ou BAS (0 Volt).



Figure 6:broche digitale

- Les entrées analogiques (10):
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 d'humidité ou de température. La carte Arduino utilise un convertisseur analogique/numérique (convertisseur CAN) pour permettre la lecture du signal par le microcontrôleur. Un signal sera converti sur 10 bits. La valeur pourra être lue sur une échelle 1024 points.
- Certaines broches ont fonctions spécialisé :
- Broches couplées de liaison série (14) : broches 0 (RX) et 1(TX). Ces broches sont particulières puisque leur rôle peut être étendu à une entrée série (RX) et une sorties séries (TX)
- Broches PWM : broches 3, 5, 6, 9, 10 et 11. Ces broches sont utilisées comme sorties PWM en utilisant la fonction `analogWrite ()`.
- Broche SPI : 10(SS), 11(MOSI), 12(MISO) et 13(SCK). Ces broches peuvent être utilisées pour gérer une communication SPI.
- LED : (13) : permet de vérifier qu'une carte arduino fonctionne bien, que vous êtes capable de la programmer.
- Indicateur led d'alimentation (13): Ce voyant doit s'allumer lorsque vous branchez votre Arduino sur une source d'alimentation pour indiquer que votre carte est correctement alimentée.
- LEDs TX et RX(14) : Sur votre carte, il y a 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 (13). Le voyant TX clignote à une vitesse variable lors de l'envoi des données série.

- **Broche aref (16) :**

AREF est l'acronyme anglais de "référence analogique". Cette broche est parfois utilisée pour définir une tension de référence externe (entre 0 et 5 Volts) comme limite supérieure pour les broches d'entrée analogiques.

- **Bouton Reset(17) :** Cela aura pour effet de redémarrer le programme depuis le début. on peut redémarrer l'Arduino UNO de deux manières : soit en utilisant le bouton "Reset", soit en connectant un bouton externe sur la broche de la carte Arduino mentionnée "RESET" (5).

I.7.4/AVANTAGE DE LA CARTE ARDUINO :

- Pas cher.
- Multiplateforme : Le logiciel Arduino, écrit en Java ou C, tourne sous les systèmes d'exploitation Windows, Macintosh et Linux.
- Environnement de programmation clair et simple.
- Logiciel et matériel open source et extensible.
- Nombreuses bibliothèques disponibles avec diverses fonctions implémentées.
- **Les composants :**

II/Les capteurs :

- ✓ **Définition principale du capteur :**

Les capteurs sont les appareils électroniques utilisés pour l'interaction avec l'environnement extérieur. Il existe différents types de capteurs disponibles pour détecter la lumière, la température, le bruit, la fumée, la proximité, etc....

II.1/Les capteurs MQ :

a/Définition :

Lorsqu'il s'agit de mesurer ou de détecter un gaz particulier, les **capteurs de gaz de la série MQ** sont les plus couramment utilisés. Ils sont parmi les plus appréciés, car ils permettent - dans les différentes applications - la détection de différents types de gaz). De plus, ils sont très bon marché et peuvent être facilement acquis.

Les capteurs MQ sont constitués d'un élément chauffant, appelé réchauffeur, et d'un capteur électrochimique ; le réchauffeur est nécessaire pour amener le capteur aux conditions de fonctionnement appropriées, car seulement à certaines températures la surface sensible du capteur (généralement, un oxyde métallique) réagira et laissera les gaz et les particules (ceux que nous souhaitons détecter) le pénétrer.

Le composant réagissant à la variation du gaz a été traité chimiquement, afin de modifier sa résistance électrique en fonction de la présence de gaz spécifiques. En pratique, il s'agit d'une résistance variable, dont les fluctuations dépendent de la quantité de gaz se trouvant dans l'air dans lequel se trouve le capteur.

b/schéma de principe d'un capteur de gaz dans un module de capteur de gaz:

A et B sont les bornes d'entrée et de sortie et H est la borne de la bobine de chauffage. Le but de la résistance variable est d'ajuster la tension de sortie et de maintenir une sensibilité élevée.

Si aucune tension d'entrée n'est appliquée à la bobine de chauffage, le courant de sortie sera très inférieure, Lorsqu'une tension suffisante est appliquée à la borne d'entrée et à la bobine de chauffage, la couche de détection se réveille et est prête à détecter tout gaz combustible à proximité

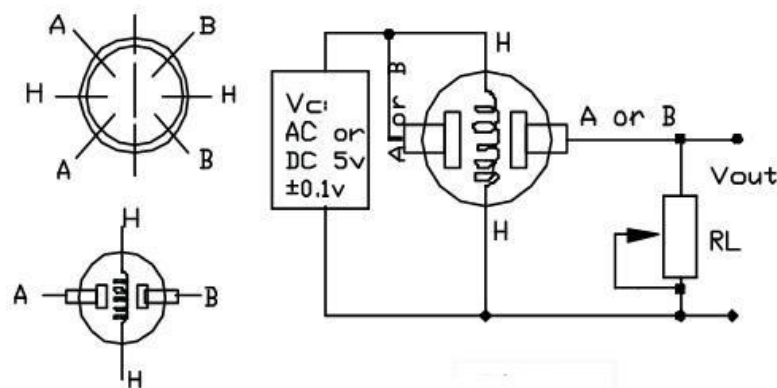


Figure 7: schéma de principe d'un capteur de gaz dans un module de capteur de gaz

c/Caractéristiques des capteurs MQ :

La plupart des capteurs de la série MQ fonctionnent avec des tensions de travail qui sont généralement à 5 V

Les capteurs de la série MQ sont principalement utilisés dans des environnements fermés qui ne devraient de toute façon pas être particulièrement froids ou chauds, Chaque capteur de la série réagit à plus d'un gaz, Vous trouverez une liste des principaux capteurs, ainsi que les gaz qu'ils détectent le plus, comme suit :

Capteur de gaz MQ2 : Sensible au méthane, au butane, au GPL, à la fumée. Ce capteur est sensible aux gaz inflammables et combustibles.

Capteur de gaz MQ3 : Sensible à l'alcool, à l'éthanol, à la fumée

Capteur de gaz MQ4 : Sensible au méthane et au GNC / gaz naturel.

Capteur de gaz MQ5 : Sensible au gaz naturel et au GPL.

Capteur de gaz MQ6 : Sensible au GPL, au gaz naturel, au charbon et au butane

Capteur de gaz MQ7 : Sensible au monoxyde de carbone

Capteur de gaz MQ8 : Sensible à l'hydrogène gazeux

Capteur de gaz MQ9 : Sensible au monoxyde de carbone, aux gaz inflammables.

Capteur de gaz MQ131 : Sensible à l'ozone

Capteur de gaz MQ135 : Pour la qualité de l'air. Sensible au benzène, à l'ammoniac, à l'alcool et à la fumée.

Capteur de gaz MQ136 : Sensible au sulfure d'hydrogène gazeux.

Capteur de gaz MQ137 : Sensible à l'ammoniac.

Capteur de gaz MQ138 : Sensible au benzène, au toluène, à l'alcool, à l'acétone, au propane, au formaldéhyde et à l'hydrogène.

d/Schéma électrique de MQ :

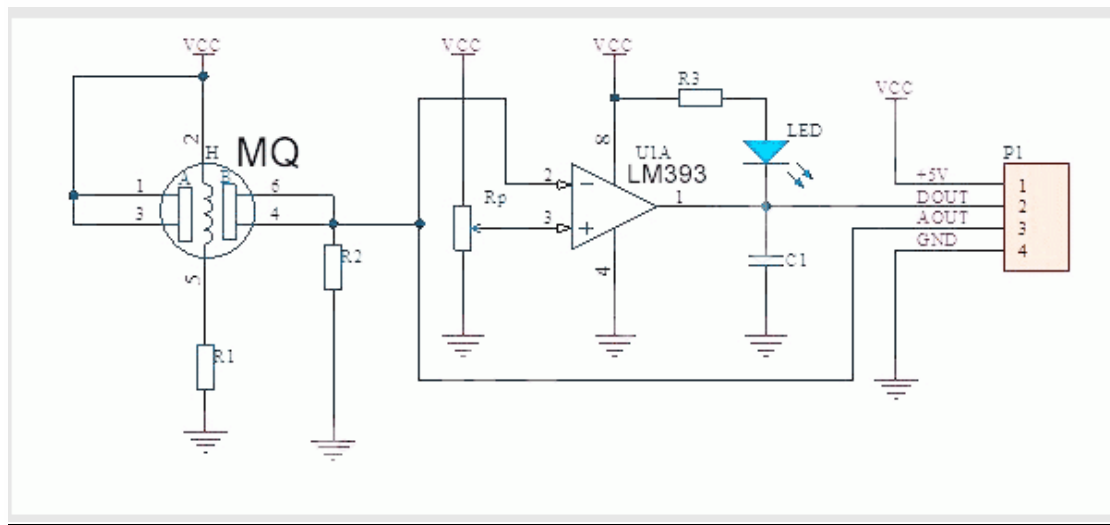


Figure 8:schema électrique de MQ.

e/Applications des MQ :

- Industries de contrôle des processus
- Surveillance de l'environnement
- Contrôle de chaudière
- Détection d'incendie
- Tests d'alcoolémie
- Détection de gaz nocifs dans les mines
- Sécurité à la maison
- Classement des agro-produits comme le café et les épices

f/Traits :

- Haute sensibilité
- Une réponse rapide
- Large plage de détection
- Performances stables et longue durée de vie

Dans notre projet nous avons utilisées les capteurs mq2 et mq7

II.1.1/Capteur Mq2 :

II.1.1.2/Qu'est-ce que le capteur de gaz MQ2 ?

MQ2 est l'un des capteurs de gaz couramment utilisés dans la série de capteurs MQ. Il s'agit d'un capteur de gaz de type semi-conducteur à oxyde métallique (MOS) également connu sous le nom de chimirésistances car la détection est basée sur le changement de résistance du matériau de détection lorsque le gaz entre en contact avec le matériau. À l'aide d'un

simple réseau de diviseurs de tension, les concentrations de gaz peuvent être détectées, le résultat peut être obtenu en mesurant la tension sur la sortie analogique.

Le capteur de gaz MQ2 fonctionne sur 5 V et consomme environ 800 mW avec une consommation de courant d'environ 150 mA. Il peut détecter des concentrations de GPL, de fumée, d'alcool, de propane, d'hydrogène, de méthane et de monoxyde de carbone de 200 à 10000 ppm.

Il s'agit d'une sortie numérique et analogique, ce qui le rend compatible avec la plupart des modules de mise en service, y compris le Raspberry Pi et l'Arduino

Voici les spécifications complètes :

- Tension de fonctionnement : 5V.
- Résistance à la charge : 20K Ω .
- Résistance de chauffage : 33 $\Omega \pm 5\%$.
- Consommation de chauffage : < 800 mW.
- Résistance de détection : 10 K Ω - 60 K Ω .
- Portée de concentration : 200-1000ppm.
- Temps de préchauffage : plus de 24h.

II.1.1.3/Structure interne du capteur de gaz MQ2 :

Le capteur est en fait enfermé dans deux couches de maillage fin en acier inoxydable appelée réseau anti-explosion. Il garantit que l'élément chauffant à l'intérieur du capteur ne provoquera pas d'explosion, car nous détectons des gaz inflammables.

Il protège également le capteur et filtre les particules en suspension afin que seuls les éléments gazeux puissent passer à l'intérieur de la chambre. Le maillage est lié au reste du corps via une bague de serrage plaquée cuivre.



Figure 9:MQ2 capteur de gaz.

II.1.1.4/Principe de fonctionnement de mq2 :

Ce capteur contient un élément sensible, principalement en céramique à base d'oxyde d'aluminium, recouvert de dioxyde d'étain, L'élément sensible a six pattes de connexion qui lui sont attachées. Deux fils sont responsables du

chauffage de l'élément sensible, les quatre autres sont utilisés pour les signaux de sortie.

L'oxygène est adsorbé à la surface du matériau de détection lorsqu'il est chauffé dans l'air à haute température. Ensuite, les électrons donneurs présents dans l'oxyde d'étain sont attirés vers cet oxygène, empêchant ainsi la circulation du courant.

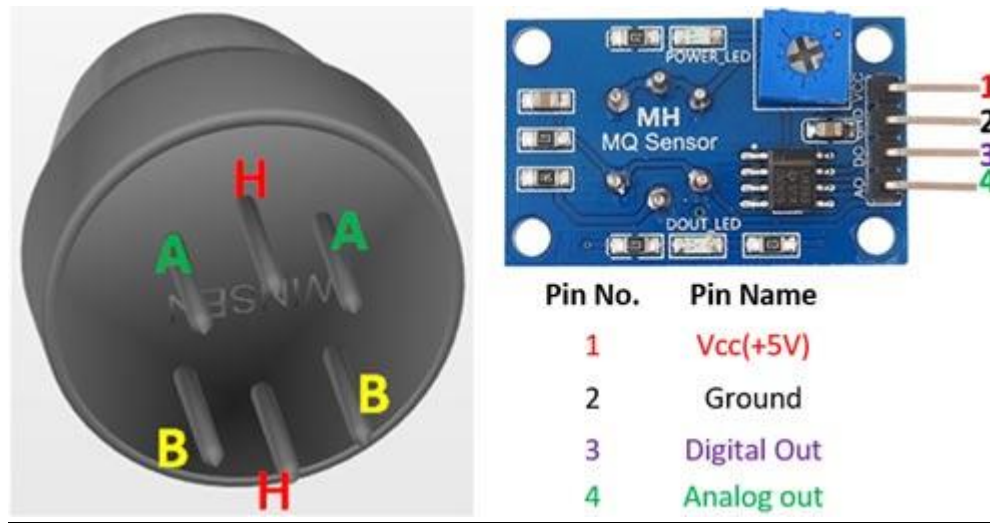
En présence de gaz réducteurs, la densité de surface de l'oxygène adsorbé diminue à mesure qu'il réagit avec les gaz réducteurs. Les électrons sont ensuite libérés dans le dioxyde d'étain, permettant au courant de circuler librement à travers le capteur.

II.1.1.5/Applications de mq2 :

Le mq2 est Utilisé pour détecter la présence de gaz dans l'air tels que le méthane, le butane, le GPL et la fumée

Ce capteur est également utilisé pour la surveillance de la qualité de l'air, l'alarme de fuite de gaz et pour le maintien des normes environnementales dans les hôpitaux. Dans les industries, ils sont utilisés pour détecter les fuites de gaz nocifs.

II.1.1.6/Configuration des broches :



a/Le module

b/ le capteur

a/Pour le module :

Numéro de broche :	Nom de broche :	La description
1	VCC	Cette broche alimente le module, généralement la tension de fonctionnement est de + 5V
2	GND	Utilisé pour connecter le module à la masse du système

3	Sortie numérique	Vous pouvez également utiliser ce capteur pour obtenir une sortie numérique à partir de cette broche, en définissant une valeur de seuil à l'aide du potentiomètre
4	Sortie analogique	Cette broche délivre une tension analogique de 0 à 5 V en fonction de l'intensité du gaz

b/Pour le capteur :

<u>1</u>	H -Pins	Sur les deux broches H, une broche est connectée à l'alimentation et l'autre à la terre
<u>2</u>	A-Pins	Les broches A et B sont interchangeables. Ces broches seront liées à la tension d'alimentation.
<u>3</u>	Broches B	Les broches A et B sont interchangeables. Une broche agira comme sortie tandis que l'autre sera tirée à la masse.

II.1.1.7/Courbe caractéristique de sensibilité :

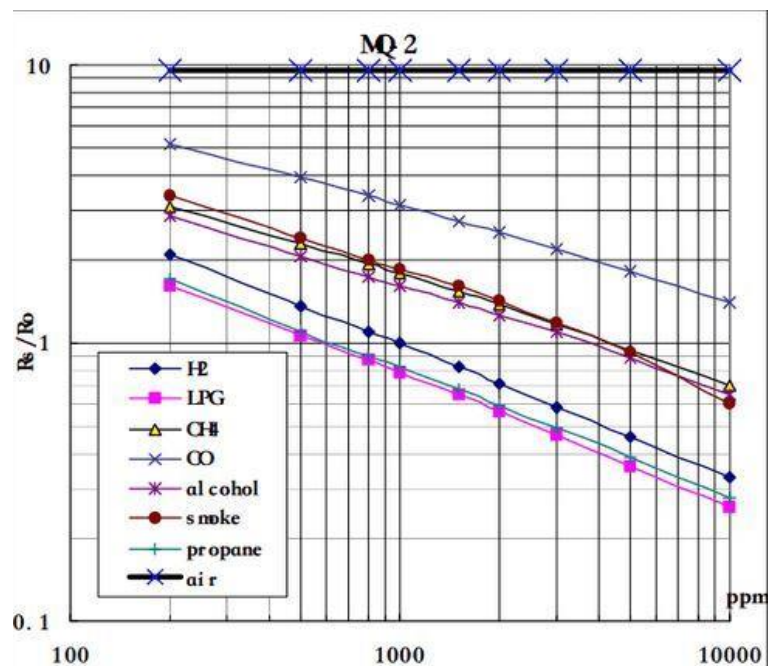


Figure 10:les caractéristiques de sensibilité typique du MQ2 pour plusieurs gaz.

II.1.1.8/Câblage de mq2 avec arduino :

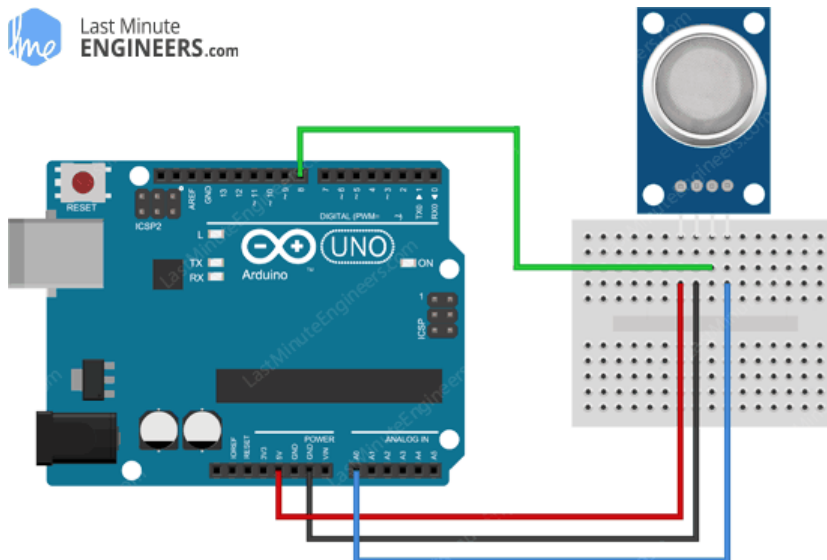


Figure 11: cablage de MQ2 avec arduino.

.II.1.2/Capteur mq7 :

II.1.2.1/Qu'est-ce que le capteur de gaz MQ7 ?

Il s'agit d'un capteur de monoxyde de carbone (CO) simple à utiliser, adapté à la détection des concentrations de CO dans l'air et affiche sa lecture comme une tension analogique. Le capteur peut mesurer des concentrations de 10 à 10000 ppm. Le capteur peut fonctionner à des températures de -10 à 50 ° C et consomme moins de 150 mA à 5 V.

Le matériau sensible du capteur de gaz MQ-7 est le SnO₂, qui a une conductivité plus faible dans l'air pur. Il fait la détection par la méthode du cycle haute et basse température, et détecte le CO à basse température (chauffé par 1,5V). La conductivité des capteurs est plus élevée avec l'augmentation de la concentration de gaz. Lorsqu'il est à haute température (chauffé de 5,0 V), il nettoie les autres gaz adsorbés à basse température.

II.1.2.2/Caractéristiques :

- Haute sensibilité aux gaz combustibles dans une large gamme
- Haute sensibilité au gaz naturel
- Une réponse rapide
- Large plage de détection
- Performances stables, longue durée de vie, faible coût
- Circuit d'entraînement simple

II.1.2.3/Données techniques :

Model No.		MQ-7	
Sensor Type		Semiconductor	
Standard Encapsulation		Plastic	
Detection Gas		Carbon Monoxide	
Concentration		10-10000ppm CO	
Circuit	Loop Voltage	V_c	$\leq 10V$ DC
	Heater Voltage	V_H	5.0V \pm 0.2V AC or DCHigh 1.5V \pm 0.1V AC or DCLow
	Heater Time	T_L	60 \pm 1S High 90 \pm 1S Low
	Load Resistance	R_L	Adjustable
Character	Heater Resistance	R_H	31 Ω \pm 3 Ω Room Tem.
	Heater consumption	P_H	$\leq 350mW$
	Sensing Resistance	R_s	2K Ω -20K Ω (in 100ppm CO)
	Sensitivity	S	$R_s(\text{in air})/R_s(100\text{ppm CO}) \geq 5$
	Slope	α	$\leq 0.6 (R_{300\text{ppm}}/R_{100\text{ppm CO}})$
Condition	Tem. Humidity	20 \pm 265% \pm 5%RH	
	Standard test circuit	$V_c: 5.0V \pm 0.1V$ $V_{H\text{High}}: 5.0V \pm 0.1V$ $V_{H\text{Low}}: 1.5V \pm 0.1V$	
	Preheat time	Over 48 hours	

II.1.2.4/Configuration des broches :

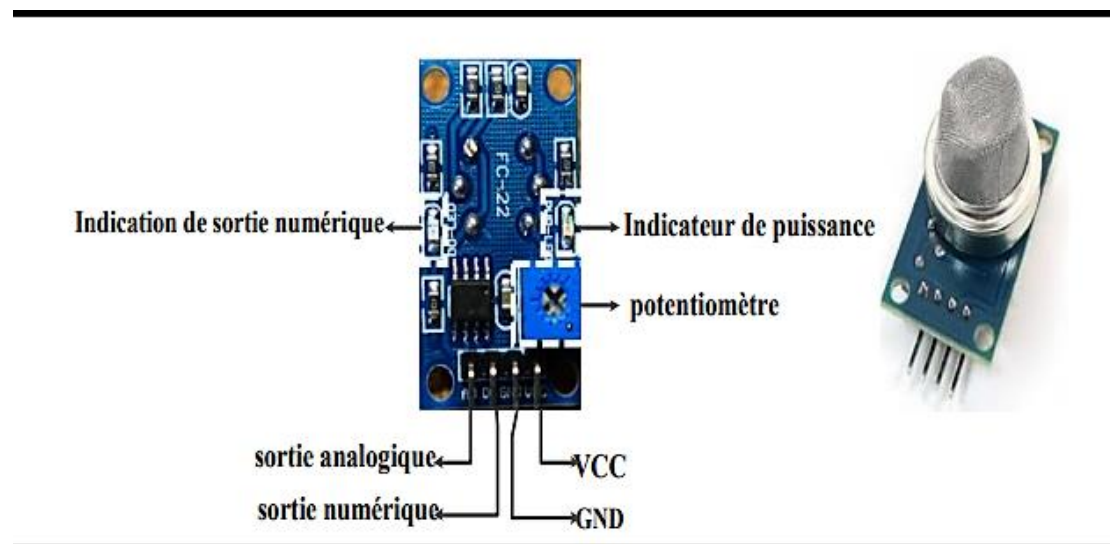


Figure 12: configuration de broche.

Il y a 4 pins : + 5V, AOUT, DOUT et GND.

Les pins VCC et GND établissent l'alimentation du capteur, Les 2 autres fils sont AOUT (sortie analogique) et DOUT (sortie numérique)

II.1.2.5/Principe de fonctionnement de mq7 :

Le fonctionnement du capteur est que la borne AOUT fournit une sortie de tension analogique proportionnelle à la quantité de monoxyde de carbone détectée par le capteur. Plus il détecte de CO, plus la tension analogique qu'il émet est élevée. Inversement, moins il détecte de CO, moins il émet de tension analogique. Si la tension analogique atteint un certain seuil, elle enverra la broche numérique DOUT est active. Une fois que cette broche DOUT est élevée, l'arduino le détectera et signalera que le seuil de CO a été atteint et est maintenant au-dessus de la limite. Vous pouvez modifier ce niveau de seuil en ajustant le potentiomètre pour augmenter ou diminuer le niveau.

II.1.2.6/Présentation du capteur de monoxyde de carbone MQ-7 :

D'après la fiche technique le capteur de monoxyde de carbone MQ-7 détecte de 50 à 4000 ppm de CO dans l'air. Voici sa courbe caractéristique de sensibilité :

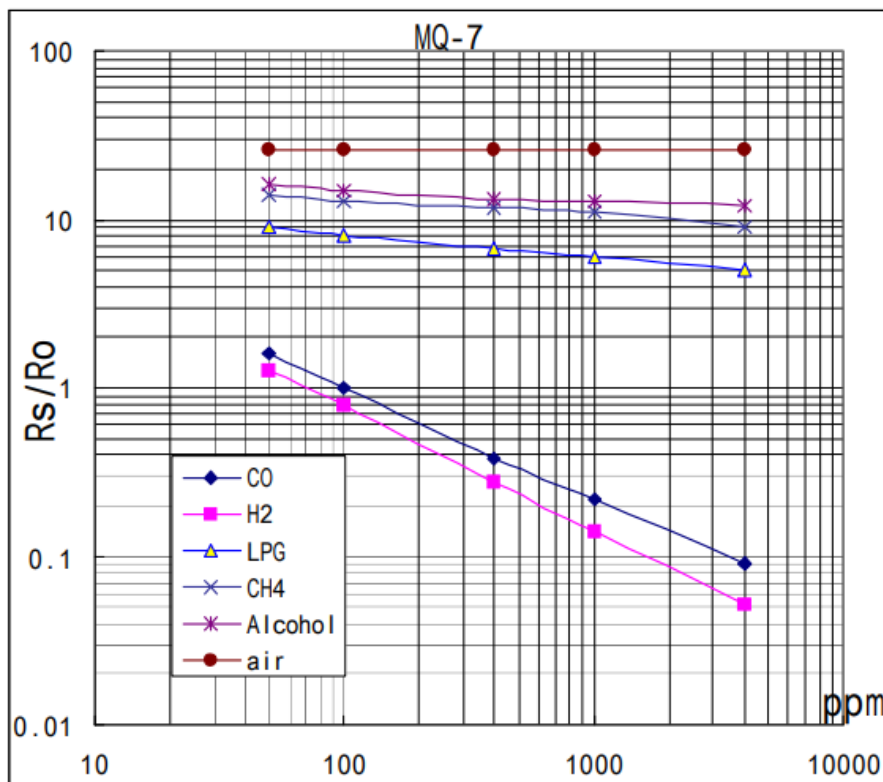


Figure 13:les caractéristiques de sensibilité typique du MQ7 pour plusieurs gaz.

II.1.2.7/Applications :

- Détecteur de fuite de gaz domestique
- Détecteur de CO industriel
- Détecteur de gaz portable

II.1.2.8/Câblage de mq7 avec arduino :

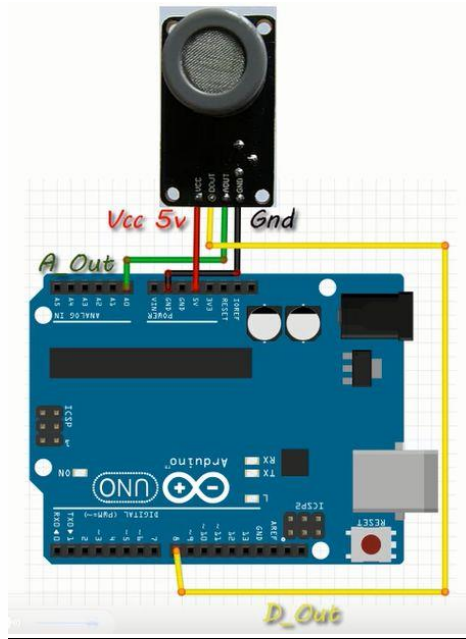


Figure 14: cablage de mq7 avec arduino.

II.1.3/Capteur dht11 :

II.1.3.1/Définition :

Le DHT11 est un capteur numérique ne coûte pas cher pour détecter la température et l'humidité. Il détecte l'humidité relative des environnements immédiats dans lesquels ils sont placés. Il mesure à la fois l'humidité et la température de l'air et expriment l'humidité relative en pourcentage du rapport entre l'humidité de l'air et la quantité maximale pouvant être contenue dans l'air à la température actuelle. Le capteur d'humidité et de température DHT11 est disponible en tant que capteur et module. Il mesure la température avec un capteur de température NTC monté en surface (thermistance) intégré à l'unité.

Ce capteur peut être facilement interfacé avec n'importe quel microcontrôleur tel qu'Arduino, Raspberry Pi etc... pour mesurer instantanément l'humidité et la température.

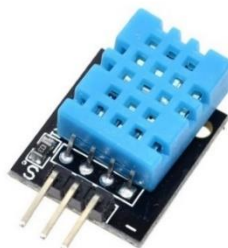


Figure 15: capteur d'humidité et de température.

II.1.3.2/Différence entre le capteur DHT11 et le module :

Le capteur DHT11 peut être acheté en tant que capteur ou module. Dans les deux cas, les performances du capteur sont identiques. Le capteur sera fourni sous la forme d'un boîtier à 4 broches dont seules trois broches seront utilisées tandis que le module sera fourni avec trois broches

La seule différence entre le capteur et le module est que le module aura un condensateur de filtrage et une résistance de rappel intégrés, et pour le capteur, vous devez les utiliser en externe si nécessaire.

II.1.3.3/Principe de fonctionnement de dht11 :

Le DHT11 détecte la vapeur d'eau en mesurant la résistance électrique entre deux électrodes. Le composant de détection d'humidité est un substrat retenant l'humidité avec des électrodes appliquées sur la surface. Lorsque la vapeur d'eau est absorbée par le substrat, des ions sont libérés par le substrat, ce qui augmente la conductivité entre les électrodes. Le changement de résistance entre les deux électrodes est proportionnel à l'humidité relative. Une humidité relative plus élevée diminue la résistance entre les électrodes, tandis qu'une humidité relative plus faible augmente la résistance entre les électrodes.

D'autre part, pour mesurer la température, ces capteurs utilisent un capteur de température NTC ou une thermistance. Une thermistance est en fait une résistance variable qui change sa résistance avec le changement de la température. Le terme «NTC» signifie «coefficient de température négatif», ce qui signifie que la résistance diminue avec l'augmentation de la température.

La plage de température du DHT11 est de 0 à 50 degrés Celsius avec une précision de 2 degrés. La plage d'humidité de ce capteur est de 20 à 80% avec une précision de 5%. Le DHT11 est de petite taille avec une tension de fonctionnement de 3 à 5 volts. Le courant maximum utilisé lors de la mesure est de 2,5 mA.

II.1.3.4/Identification et configuration des broches:

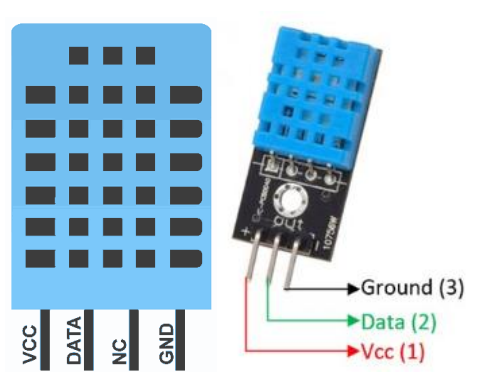


Figure 16:configuration de broche.

N ° de broche	Nom de broche	Description de la broche
1	VCC	Source de courant 3,3 à 5,5 volts CC
2	DATA(LES DONNÉES)	Produit à la fois la température et l'humidité via des données série
3	NC	Hors service
4	GND	Sol

II.1.3.5/Applications de dht11 :

- la mesure des valeurs d'humidité et de température dans les systèmes de chauffage, de ventilation et de climatisation
- Station météo locale
- Contrôle automatique du climat
- Surveillance de l'environnement

II.1.3.6/Câblage de dht11 avec 4 broches:

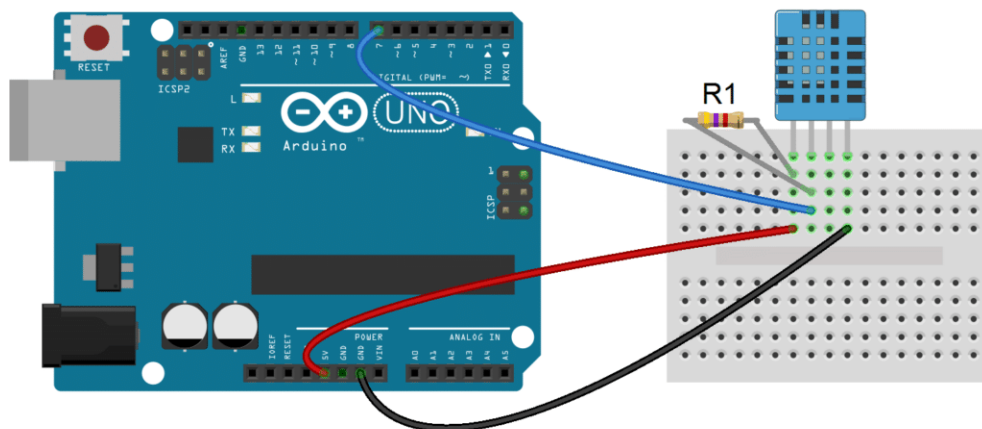


Figure 17:cablage de dht11 à 4 broches avec arduino.

II.1.3.7/Câblage de dht11A avec 3 broches:

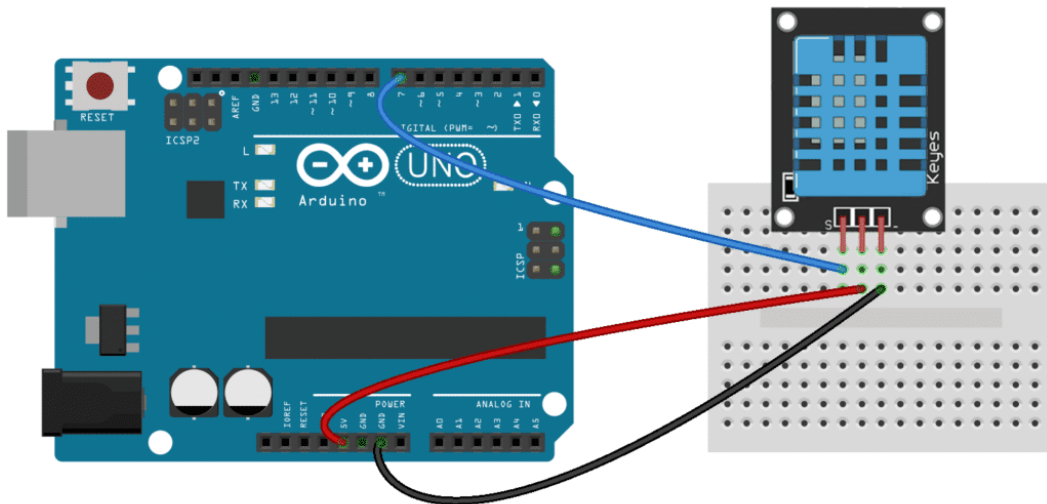


Figure 18: cablage de dht11 à 3 broches avec arduino.

II.1.4/Capteur BMP280 :

II.1.4.1/Définition :

Les capteurs de pression sont l'un des capteurs les plus utilisés et peuvent être trouvés dans les sondes pour les mesures en laboratoire, mais plus souvent dans des milliards d'appareils, y compris les smartphones, les vêtements portables, les automobiles, les drones, les centres météorologiques et les instruments médicaux.

Le module capteur BME280 lit la pression la température et l'humidité. La pression change avec l'altitude, vous pouvez également estimer l'altitude. Il existe plusieurs versions de ce module capteur. Le capteur BME280 utilise le protocole de communication I2C pour échanger des données avec un microcontrôleur nous pouvons l'utiliser comme altimètre avec une précision de ± 1 mètre (précision de la pression = ± 1 hPa).

II.1.4.2/Caractéristiques :

Tension d'alimentation: 1.8 – 5V DC

Une interface I2C (jusqu'à 3.4MHz), SPI (jusqu'à 10 MHz),

Gamme opérationnelle:

Température: -40 à + 85 ° C

Humidité: 0-100%

Pression: 300-1100 hPa

Résolution:

Température: 0,01 °C

Humidité: 0,008%

Pression: 0.18Pa

Exactitude:

Température: + -1 ° C

Humidité : + -3%

Pression: + -1Pa

adresse : I2C

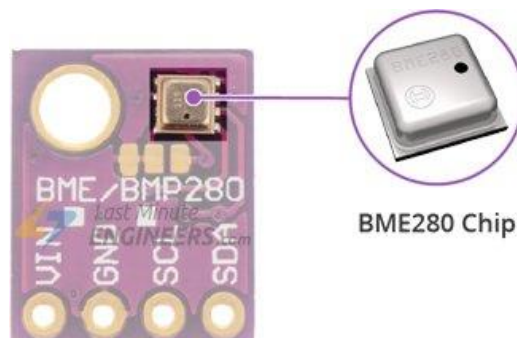
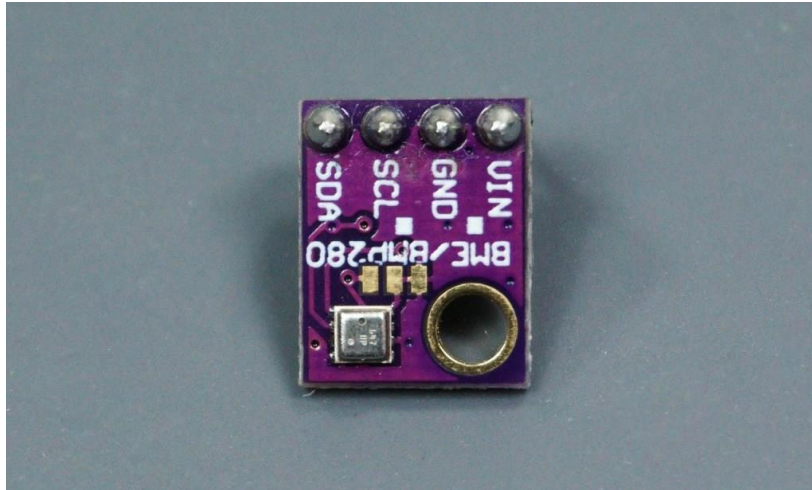
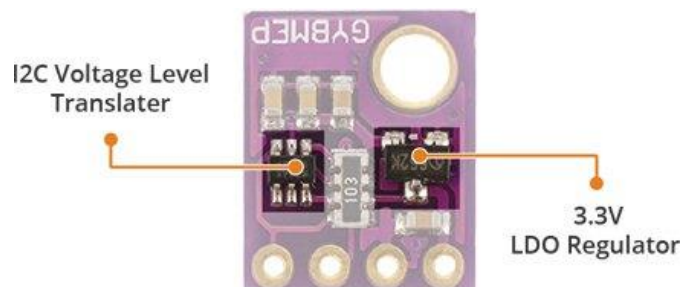


Figure 19:capteur BMP280.

II.1.4.3/Puissance requise :

Le module est livré avec un régulateur LM6206 3,3 V intégré et un convertisseur de niveau de tension I2C, vous pouvez donc l'utiliser sans souci avec un microcontrôleur logique 3,3 V ou 5 V comme Arduino.



Le BME280 consomme moins de 1 mA pendant les mesures et seulement 5 μ A pendant le ralenti. Cette faible consommation d'énergie permet la mise en œuvre dans des appareils alimentés par batterie tels que des combinés, des modules GPS ou des montres.

II.1.4.4/Brochage du capteur BME280 :

Le module BME280 ne dispose que de 4 broches qui le connectent au monde extérieur. Les connexions sont les suivantes:

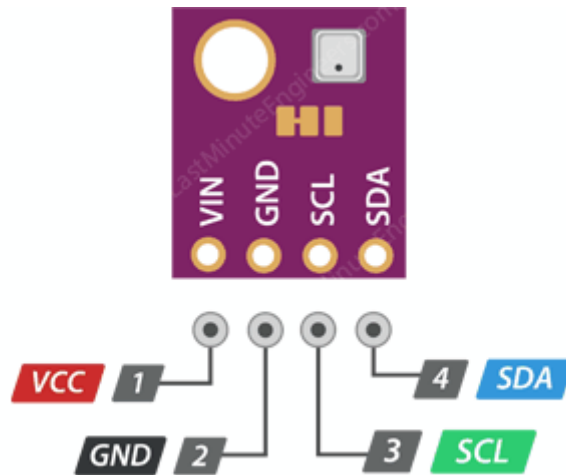


Figure 20:broche de BMP280.

VCC est l'alimentation du module, qui peut être comprise entre 3,3 V et 5 V.

GND doit être connecté à la terre d'Arduino

SCL est une broche d'horloge série pour l'interface I2C.

SDA est une broche de données série pour l'interface I2C

II.1.4.5/Câblage du module BME280 à Arduino :

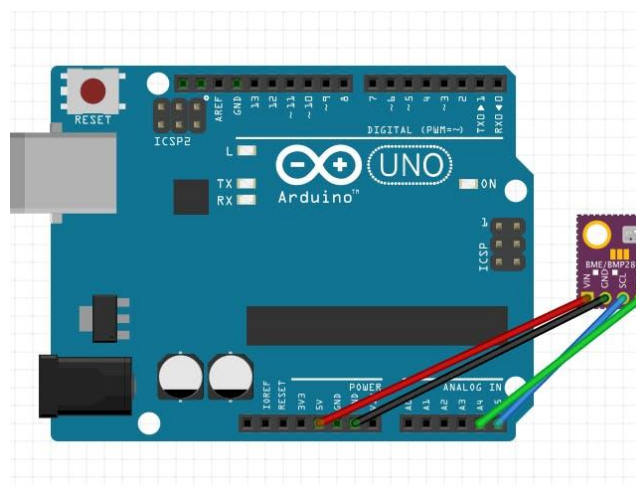


Figure 21:cablage du module bmp280 avec arduino.

II.1.5/Le DS3231 :

II.1.5.1/Définition :

Le DS3231 est une horloge temps réel I2C (RTC) très précise et peu coûteuse qui peut conserver les heures, les minutes et les secondes, ainsi que le jour, le mois et l'année avec un oscillateur à cristal compensé en

température (TCXO) .En outre, il a une compensation automatique pour les années bissextiles et pour les mois de moins de 31 jours.

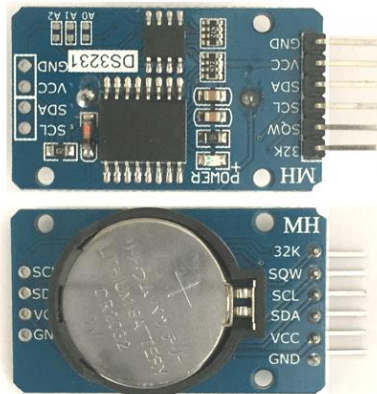


Figure 22:module DS3231.

Le module peut fonctionner sur 3,3 ou 5 V, ce qui le rend approprié pour de nombreuses plates-formes de développement ou microcontrôleurs. L'entrée de la batterie est de 3 V et une batterie CR2032 3 V typique peut alimenter le module et conserver les informations pendant plus d'un an.

II.1.5.2/Principe de fonctionnement :

Le module d'horloge en temps réel DS3231 garde une trace de l'heure même lorsque le module n'est pas alimenté. Il a une batterie 3V intégrée qui met à jour l'heure. Nous obtiendrons l'heure et la date du module RTC en utilisant les fonctions de la bibliothèque

La plupart des RTC utilisent un cristal de synchronisation externe de 32 kHz qui est utilisé pour garder le temps avec une faible consommation de courant, ce RTC est dans un emballage costaud parce que le cristal est à l'intérieur du puce Et juste à côté du cristal intégré se trouve un capteur de température. Ce capteur compense les changements de fréquence en ajoutant ou en supprimant des tics d'horloge afin que l'indication de l'heure reste dans les délais. L'adresse et les données sont transférées en série via un bus bidirectionnel I2C.

II.1.5.3/CARACTÉRISTIQUES :

- RTC très précis gère entièrement toutes les fonctions de chronométrage
- L'horloge en temps réel compte les secondes, les minutes, les heures, la date du mois, le mois, le jour de la semaine et l'année, avec une compensation de l'année bissextile valable jusqu'à 2100
- Précision ± 2 ppm de 0°C à $+40^{\circ}\text{C}$
- Précision $\pm 3,5$ ppm de -40°C à $+85^{\circ}\text{C}$
- Sortie du capteur de température numérique: précision de $\pm 3^{\circ}\text{C}$
- Deux alarmes horaires

- Signal de sortie à onde carrée programmable
- Une interface série simple se connecte à la plupart des microcontrôleurs
- Interface I2C rapide (400 kHz)
- Entrée de batterie de secours pour un chronométrage continu
- Le fonctionnement à faible puissance prolonge le temps d'exécution de la batterie de secours
- Fonctionnement 3.3V

II.1.5.4/Les broches du ds3231 :

Le module DS3231 RTC a un total de 6 broches qui le connectent au monde extérieur. Les connexions sont les suivantes:

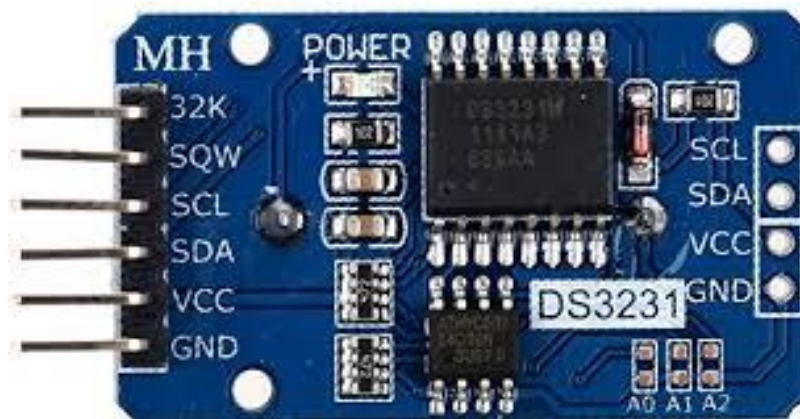


Figure 23 : Les broches de module DS3231.

32K la broche émet une horloge de référence stable (compensée en température) et précise.

SQW la broche émet une belle onde carrée à 1 Hz, 4 kHz, 8 kHz ou 32 kHz et peut être gérée par programme. Cela peut en outre être utilisé comme interruption en raison de la condition d'alarme dans de nombreuses applications temporelles.

SCL est une broche d'horloge série pour l'interface I2C.

SDA est une broche de données série pour l'interface I2C.

VCC la broche alimente le module. Il peut être compris entre 3,3 V et 5,5 V.

GND est une broche de terre.

II.1.5.5/Applications :

- Robotique
- Gaming
- Les serveurs
- Périphériques d'ordinateur
- GPS
- Compteurs de puissance

II.1.5.6/Câblage du module RTC DS3231 à Arduino UNO :

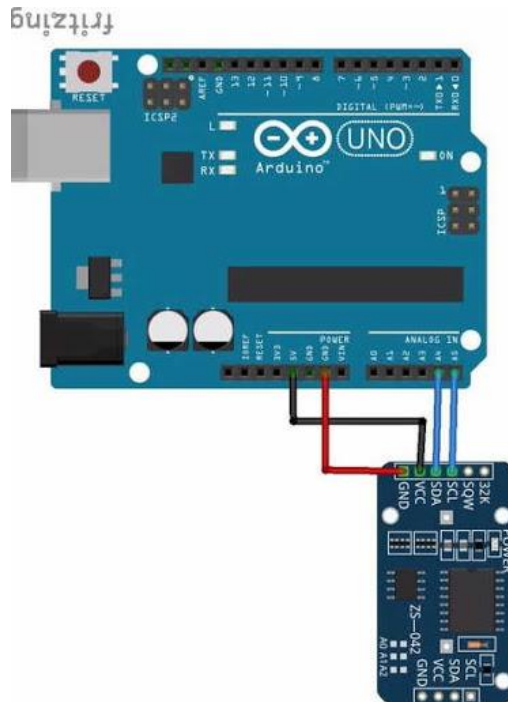


Figure 24:cablage de module DS3231 avec l'arduino.

II.1.6/Capteur à ultrason :

II.1.6.1/Définition :

Le module mesure environ 45x20x15 mm et dispose d'une connexion à 4 broches ; Le module de mesure à ultrasons HC - SR04 fournit une mesure sans contact de 2 à 400 cm, avec une précision de mesure allant jusqu'à 3 mm. Le module comprend des émetteurs ultrasoniques, un récepteur et des circuits de commande. Ce télémètre à ultrasons HC-SR04 est un capteur très populaire que l'on trouve dans de nombreuses applications où il nécessite de mesurer la distance et de détecter les objets. Le module a deux yeux comme des projets à l'avant qui forment l'émetteur et le récepteur à ultrasons. Le capteur à ultrasons HC-SR04 utilise un sonar pour déterminer la distance à un objet comme le font les chauves-souris ou les dauphins.



Figure 25:capteur à ultrason.

Un capteur à ultrasons émet à intervalles réguliers de courtes impulsions sonores à haute fréquence (40 khz). Ces impulsions se propagent dans l'air à la vitesse du son. L'oreille humaine ne peut pas entendre ces ondes sonores et quand ces ondes rentrent en collision avec objet, elles sont réfléchies et reçues via le port Echo.

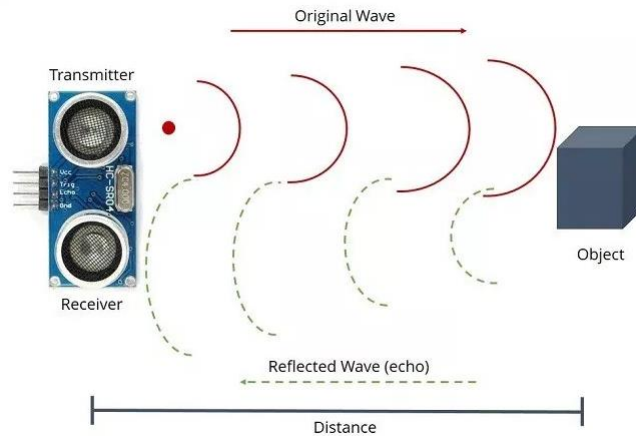


Figure 26: mise en marche de capteur à ultrason.

II.1.6.2/Principe de fonctionnement :

Le principe de fonctionnement du capteur est entièrement basé sur la vitesse du son.

On envoie une impulsion HIGH de $10\mu\text{s}$ sur la broche TRIGGER du capteur. Le capteur envoie alors une série de 8 impulsions ultrasoniques à 40KHz (inaudible pour l'être humain, c'est quand plus agréable qu'un biiiiiiip).

Les ultrasons se propagent dans l'air jusqu'à toucher un obstacle et retournent dans l'autre sens vers le capteur.

Le capteur détecte l'écho et clôture la prise de mesure.

Le signal sur la broche echo du capteur reste à high durant les étapes 3 et 4, ce qui permet de mesurer la durée de l'aller-retour des ultrasons et donc de déterminer la distance.

La distance de l'objet détecté est égale au temps de l'impulsion de sortie divisée par deux (le signal fait un aller-retour) et multipliée par la vitesse du son dans l'air (340m/s).

II.1.6.3/Calcul :

Calcul Distance parcourue par le son : vitesse du son (340 m/s) * temps aller-retour du son / 2

$$\text{Distance} = 340\text{m/s} * t \text{ en sec} / 2 = 34000 \text{ cm} / 1000000\mu\text{s} * t \text{ en } \mu\text{s} / 2 = 17 / 1000 * t$$

$$t = \text{Distance} * 1000 / 17 = D * 58,82 \text{ en } \mu\text{s}$$

$$\text{Distance en cm} = \text{temps propagation en } \mu\text{s} / 58$$

II.1.6.4/Spécification :

Tension de fonctionnement : + 5V.

Distance de mesure théorique : 2 cm à 450 cm.

Précision : 3 mm.
Angle de mesure couvert : <15 °.
Courant de fonctionnement : <15mA.
Fréquence de fonctionnement : 40 Hz.

II.1.6.5/Applications :

Ce capteur peut être utilisé de plein de façons, entre autres :

- visualiser sur un TCO la présence d'un convoi ou non sur une portion de voie.
- connaître la position de quelque chose
- arrêter un train en gare à une position précis

II.1.6.6/Câblage avec la carte arduino :

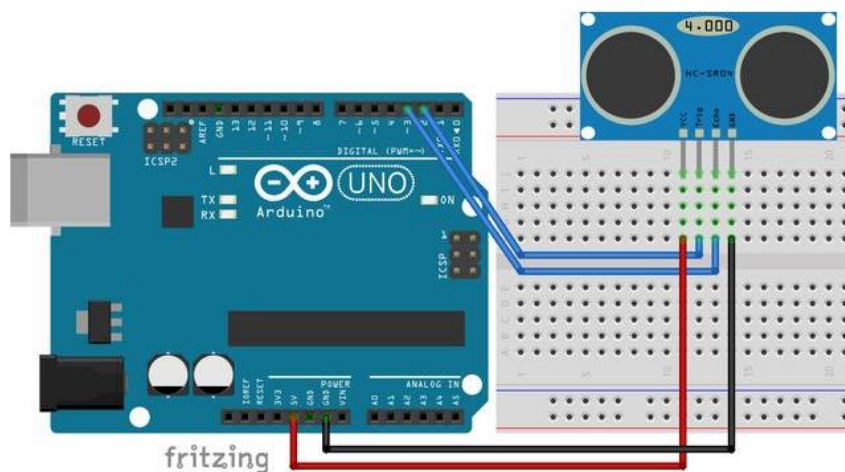


Figure 27:câblage de capteur à ultrason avec l'arduino.

L'alimentation 5v de la carte Arduino va sur la broche vcc du capteur.
La broche GND de la carte Arduino va sur la broche GND du capteur.
La broche D2 de la carte Arduino va sur la broche TRIGGER du capteur.
La broche D3 de la carte Arduino va sur la broche ECHO du capteur.

III/VENTILATEUR DC 12V :

III.1/Définition :

Les ventilateurs sont devenus indispensables dans les installations industrielles modernes. Que ce soit pour la ventilation des endroits ou tout simplement pour l'air frais

Nous avons utilisé un ventilateur 12v de type DC



Figure 28:ventilateur DC.

Les moteurs des ventilateurs sont très simples de construction. Ils sont constitués d'un moteur électrique synchrone ou asynchrone, avec une roue de ventilateur de l'arbre associé, et des connexions pour les lignes. Cependant, des moteurs à courant continu sont utilisés (par exemple dans les ordinateurs ou dans une voiture). Tous les moteurs se composent d'un stator et contiennent un rotor rotatif à l'intérieur, qui est entouré de pièces polaires.

Les ventilateurs DC sont largement considérés comme le type de ventilateur le plus efficace. Ils consomment beaucoup moins d'énergie que les ventilateurs AC. En fait, les ventilateurs DC consomment jusqu'à 70% d'énergie en moins pour produire la même sortie que les types de ventilateurs AC traditionnels.

III.2/Câblage de ventilateur avec arduino :

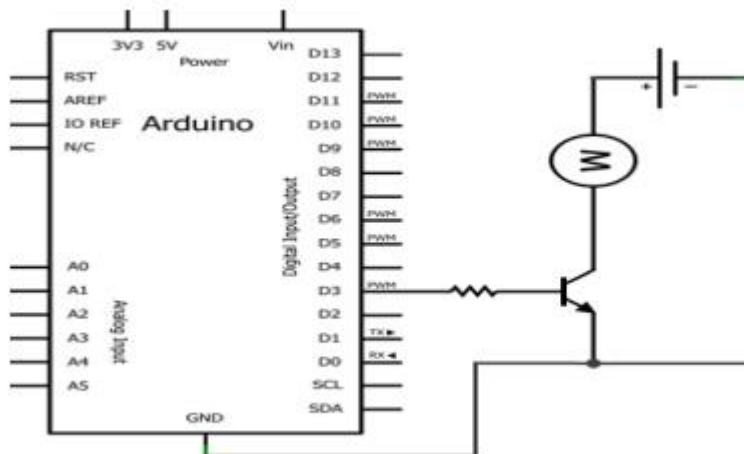


Figure 29 : câblage de ventilateur avec l'arduino.

IV/Diode électroluminescente (LED) :

IV.1/Définition :

nous avons utilisé des leds pour l'alarme du danger.

Une diode électroluminescente (LED) est un dispositif semi - conducteur qui émet de la lumière visible lorsqu'un courant électrique la traverse.

Les LED sont des sources de lumière à semi-conducteur qui combinent un semi-conducteur de type P (concentration de trous plus importante) avec un semi-conducteur de type N (concentration d'électrons plus importante). L'application d'une tension directe suffisante entraînera la recombinaison des électrons et des trous à la jonction PN, libérant de l'énergie sous forme de lumière.

La sortie d'une LED peut aller du rouge (à une longueur d'onde d'environ 700 nanomètres) au bleu-violet (environ 400 nanomètres). Certaines LED émettent de l'énergie infrarouge (IR) (830 nanomètres ou plus).

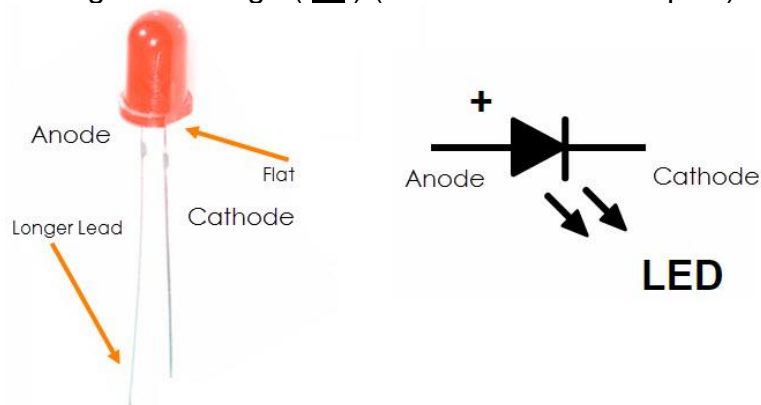


Figure 30:diode électroluminescente.

IV.2/Les avantages des LED :

Les avantages des LED par rapport aux dispositifs d'éclairage incandescents et fluorescents, comprennent :

- **Faible consommation d'énergie** : la plupart des types peuvent fonctionner avec des alimentations par batterie.
- **Haute efficacité** : la majeure partie de l'énergie fournie à une LED est convertie en rayonnement sous la forme souhaitée, avec une production de chaleur minimale.
- **Longue durée de vie** : Lorsqu'elle est correctement installée, une LED peut fonctionner pendant des décennies.

IV.3/Les applications des leds :

- **Voyants lumineux** : ceux- ci peuvent être à deux états (c.-à-d., Marche / arrêt), à barres ou à affichage alphabétique-numérique.
- **Rétro-éclairage du panneau LCD** : des LED blanches spécialisées sont utilisées dans les écrans d'ordinateur à écran plat.
- **Transmission de données par fibre optique** : la facilité de modulation permet une large bande passante de communication avec un minimum de bruit, résultant en une vitesse et une précision élevées.

- **Télécommande** : la plupart des "télécommandes" de divertissement à domicile utilisent des IRED pour transmettre des données à l'unité principale.
- **Optoisolateur** : les étapes d'un système électronique peuvent être connectées ensemble sans interaction indésirable.

IV.4/Câblage de led avec arduino :

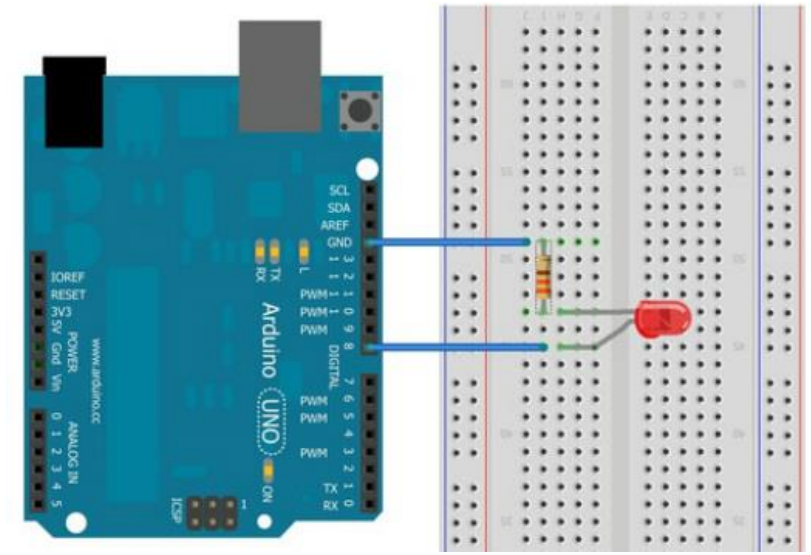


Figure 31:câblage de la diode avec l'arduino.

V/Lcd 16*2 :

V.1/Description de Lcd 16*2 :

Nous rencontrons des écrans LCD partout autour de nous. Les ordinateurs, les calculatrices, les téléviseurs, les téléphones portables, les montres numériques utilisent une sorte d'affichage pour afficher l'heure. Un LCD est un module d'affichage électronique qui utilise des cristaux liquides pour produire une image visible. L'écran LCD 16 x 2 est un module très basique couramment utilisé dans les bricolages et les circuits. Le 16 x 2 traduit un affichage de 16 caractères par ligne sur 2 de ces lignes. Sur cet écran LCD, chaque caractère est affiché dans une matrice 5 x 7 pixels.

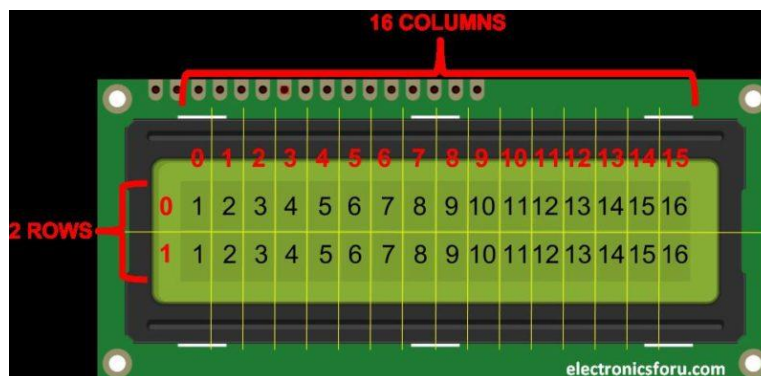


Figure 32:afficheur LCD.

V.2/Diagramme de brochage LCD 16X2

<u>N DE PIN</u>	<u>Nom de pin</u>	<u>Fonction</u>
1	GND	Masse (0V)
2	Vcc	Tension d'alimentation; 5 V (4,7 V - 5,3 V)
3	Vo / VEE	Réglage du contraste; la meilleure façon est d'utiliser une résistance variable telle qu'un potentiomètre. La sortie du potentiomètre est connectée à cette broche. Tournez le bouton du potentiomètre vers l'avant et vers l'arrière pour régler le contraste de l'écran LCD.
4	RS(Register Select)	Bascule entre le registre des commandes / données
5	Read/write	Utilisé pour lire ou écrire des données. Normalement mis à la terre pour écrire des données sur l'écran LCD
6	Enable	Connecté à la broche du microcontrôleur et basculé entre 1 et 0 pour la reconnaissance des données
7	DB0	Broches de données 8 bits
8	DB1	
9	DB2	
10	DB3	
11	DB4	
12	DB5	
13	DB6	
14	DB7	
15	Rétro-éclairage VCC (5V)	Led +
16	GND De rétroéclairage (0 V)	Led -

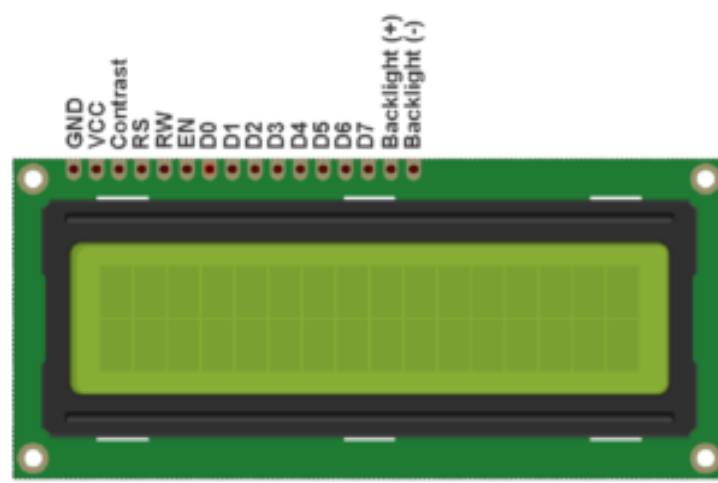


Figure 33:les broches de l'afficheur LCD.

V.3/Caractéristiques du module LCD 16 x 2

- La tension de fonctionnement est de 4,7 V à 5,3 V
- La consommation actuelle est de 1mA sans rétro-éclairage
- Module d'affichage LCD alphanumérique, ce qui signifie qu'il peut afficher des alphabets et des chiffres
- Se compose de deux lignes et chaque ligne peut imprimer 16 caractères.
- Chaque personnage est construit par une boîte de 5 x 8 pixels
- Peut fonctionner en mode 8 bits et 4 bits
- Il peut également afficher tous les caractères générés personnalisés
- Disponible en rétro-éclairage vert et bleu

V.4/Les avantages de lcd :

- L'image est parfaitement nette à la résolution native du panneau.
- Une intensité de crête élevée produit des images très lumineuses. Idéal pour les environnements fortement éclairés
- Les lcds sont parfaitement plats.
- mince, avec un faible encombrement. Consomme peu d'électricité et produit peu de chaleur.

V.5/Câblage de lcd avec arduino :

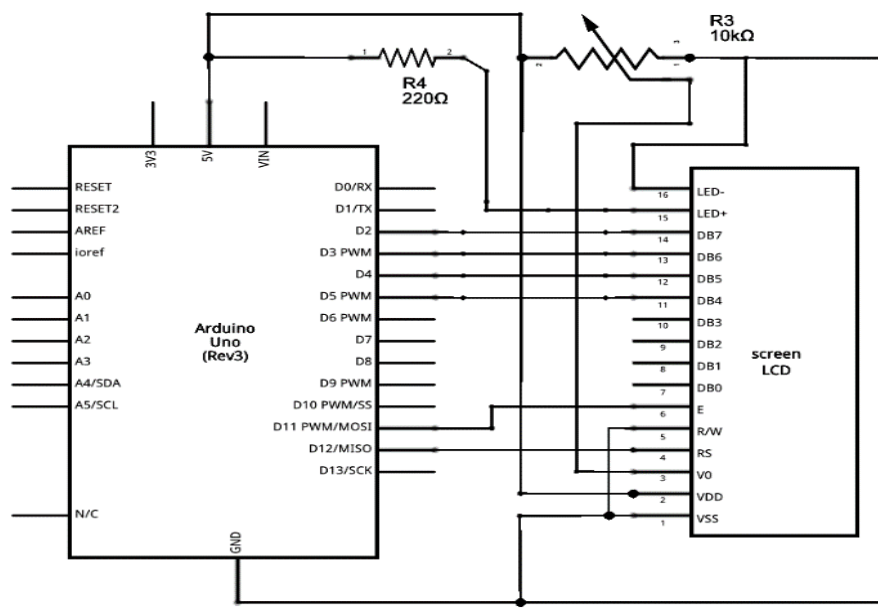


Figure 34:cablage de l'afficheur avec arduino.

VI/ Les résistances :

VI.1/ Définition :

La résistance est un composant électrique passif pour créer une résistance dans le flux de courant électrique. Dans presque tous les réseaux électriques et circuits électroniques, ils peuvent être trouvés. La résistance est mesurée en ohms. Un ohm est la résistance qui se produit lorsqu'un courant d'un

ampère passe à travers une résistance avec une chute d'un volt à ses bornes. Le courant est proportionnel à la tension aux bornes. Ce rapport est représenté par la loi d'Ohm : $R=V/I$.

Il existe un grand nombre de types de résistance différents qui peuvent être achetés et utilisés. Les propriétés de ces différentes résistances varient et cela aide à obtenir le bon type de résistance pour une conception donnée afin de garantir les meilleures performances.

Il existe deux symboles de circuit principaux utilisés pour les résistances. Le plus ancien est encore largement utilisé en Amérique du Nord et se compose d'une ligne dentelée représentant le fil utilisé dans une résistance. L'autre symbole de circuit de résistance est un petit rectangle, et cela est souvent appelé le symbole de résistance international et il est plus largement utilisé en Europe et en Asie.

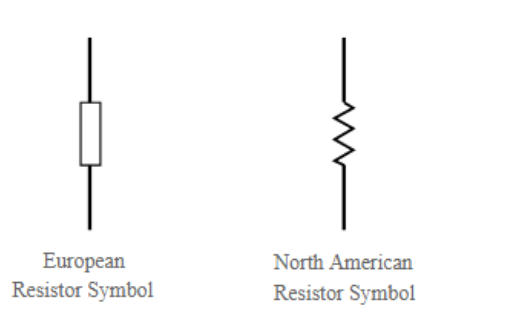


Figure 35:resistance.

VI.2/Les types de résistances :

Il existe 2 types de résistances :

- **Résistances fixes :** les résistances fixes sont de loin le type de résistance le plus utilisé. Ils sont utilisés dans les circuits électroniques pour définir les bonnes conditions dans un circuit. Leurs valeurs sont déterminées pendant la phase de conception du circuit, et elles ne devraient jamais avoir besoin d'être modifiées pour "ajuster" le circuit. Il existe de nombreux types de résistance différents qui peuvent être utilisés dans différentes circonstances et ces différents types de résistance sont décrits plus en détail ci-dessous.



Figure 36:résistance fixe.

Résistances variables : Ces résistances se composent d'un élément de résistance fixe et d'un curseur qui tapote sur l'élément de résistance principal. Cela donne trois connexions au composant : deux connectées à l'élément fixe et la troisième est le curseur. De cette façon, le composant agit comme un diviseur de potentiel variable si les trois connexions sont utilisées. Il est possible de se connecter au curseur et à une extrémité pour fournir une résistance à résistance variable.



Figure 37:resistance variable.

VI.3/code des couleurs de résistance :

TOLERANCE

GOLD	±5%
SILVER	±10%

BLACK	0	0	x10	
BROWN	1	1	x100	
RED	2	2	x1000	$K\Omega = x1,000\Omega$
ORANGE	3	3	x1,0000	
YELLOW	4	4	x10,0000	$M\Omega = x1,000,000\Omega$
GREEN	5	5	x100,0000	
BLUE	6	6	x1,000,0000	
VIOLET	7	7		
GRAY	8	8		
WHITE	9	9		

MULTIPLIER

03 APR 2013
ver 3 rev 0

Figure 38:code de couleurs

VI.4/APPLICATIONS :

Les résistances sont utilisées à de nombreuses Champs. Quelques exemples incluent la délimitation du courant électrique, la division de la tension, la génération de chaleur, les circuits d'adaptation et de chargement, le gain de contrôle et les constantes de temps fixes. Ils sont disponibles dans le

commerce avec des valeurs de résistance sur une plage de plus de neuf ordres de grandeur. Ils peuvent être utilisés comme freins électriques pour dissiper l'énergie cinétique des trains, ou être plus petits qu'un millimètre carré pour l'électronique.

VI.5 Câblage des résistances avec arduino :

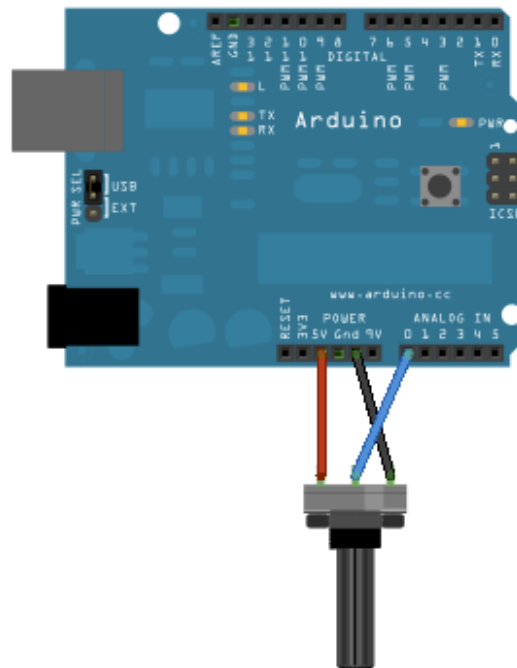
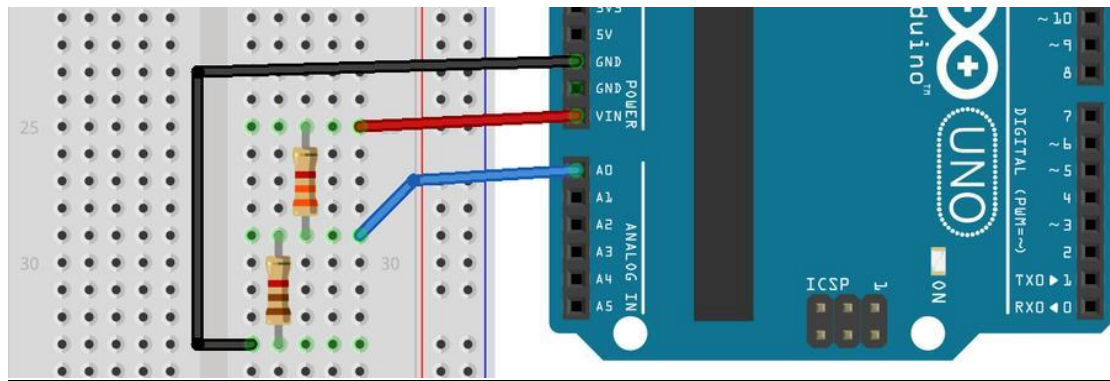


Figure 39:cablage des résistances avec arduino.

VII/Le Buzzer :

VII/Définition :

Un buzzer est un composant petit mais efficace pour ajouter des fonctionnalités sonores à notre projet / système. Il est très petit et compact à 2 broches et peut donc être facilement utilisé sur une planche à pain , une carte perforée et même sur des PCB, ce qui en fait un composant largement utilisé dans la plupart des applications électroniques.

Les buzzers sont également connus sous le nom d'alarmes audio, indicateurs audio, transducteurs audio, buzzers piézo, bips et / ou alarmes sonores.

buzzer est une structure intégrée de transducteurs électroniques, alimentation DC, largement utilisé dans les ordinateurs, imprimantes, copieurs, alarmes, jouets électroniques, automobile équipement électronique, téléphones, minuteries et autres produits électroniques pour le son dispositifs.



Figure 40:buzzer.

VII.2/Caractéristiques et spécifications du buzzer

- Tension nominale : 6V DC
- Tension de fonctionnement : 4-8V DC
- Courant nominal : <30mA
- Type de son : bip continu
- Fréquence de résonance : ~ 2300 Hz
- Emballage scellé petit et soigné
- Compatible avec la planche à pain et la carte perforée

VII.3Les types du buzzers :

Ils peuvent être divisés en dispositifs piézoélectriques et électromécaniques.

- **Piézoélectriques** : les buzzers piézoélectriques contiennent une plaque de vibration piézoélectrique (également appelée élément piézoélectrique) dans un boîtier moulé. Le son est émis lorsqu'une tension est appliquée et l'élément piézo à l'intérieur du boîtier vibre.
- **Électromécanique** : les buzzers électromécaniques utilisent un électro-aimant et un disque en métal. Lorsqu'une tension est appliquée, un champ magnétique est généré, provoquant le déplacement de l'aimant et la vibration du diaphragme, générant un son audible.

VII.4/Les différences entre piézo et électromécanique sont :

	Pièzo	Électromécanique
Plage de tension de fonctionnement	3V-250V	1.5V-30V
Consommation de courant	Habituellement moins de 30mA	Jusqu'à 100mA
Fréquence nominale	Plus haute	Inférieur

Sortie sonore	Généralement plus élevé	Généralement plus faible
---------------	-------------------------	--------------------------

Les buzzers piézo utilisent généralement moins de courant, ils ont une sortie sonore plus élevée et une tension de fonctionnement plus large. Cependant, les buzzers électromagnétiques sont capables de produire du son à basse fréquence dans de petits boîtiers compacts.

Les buzzers peuvent également être divisés en indicateurs audio et transducteurs audio:

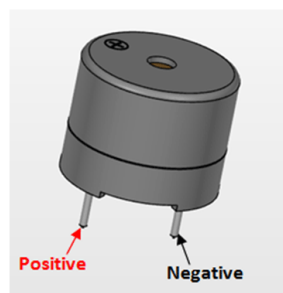
- **Indicateurs audio** : ces appareils génèrent du son simplement en appliquant une tension continue. Ils sont auto-générateurs ils ont des circuits d'entraînement internes avec des tonalités préprogrammées.



- **Transducteurs audio** - ils sont pilotés à l'aide d'une tension alternative (un signal d'onde) et nécessitent des circuits supplémentaires pour produire un son. Ces appareils offrent une plus grande flexibilité en termes de gamme de sortie sonore



VII.5/Configuration des broches du buzzer :



Code PIN	Nom de broche	La description
1	Positif	Identifié par le symbole (+) ou un câble terminal plus long. Peut être alimenté par 6V DC

2	Négatif	Identifié par un câble terminal court. Généralement connecté à la masse du circuit
---	---------	--

VII.6/Applications des buzzers :

Il existe de nombreux types d'applications qui utilisent des buzzers, généralement à des fins d'alarme ou d'identification. Ceux-ci inclus:

- Produits de sécurité et de sûreté
- Des appareils portables
- Équipement médical
- Systèmes de télésurveillance
- Minuteries
- Appareils ménagers
- Instruments de mesure et de débit
- Machines de pesage
- Reconnaissance d'entrée utilisateur

VII.7/Câblage du buzzer avec arduino :

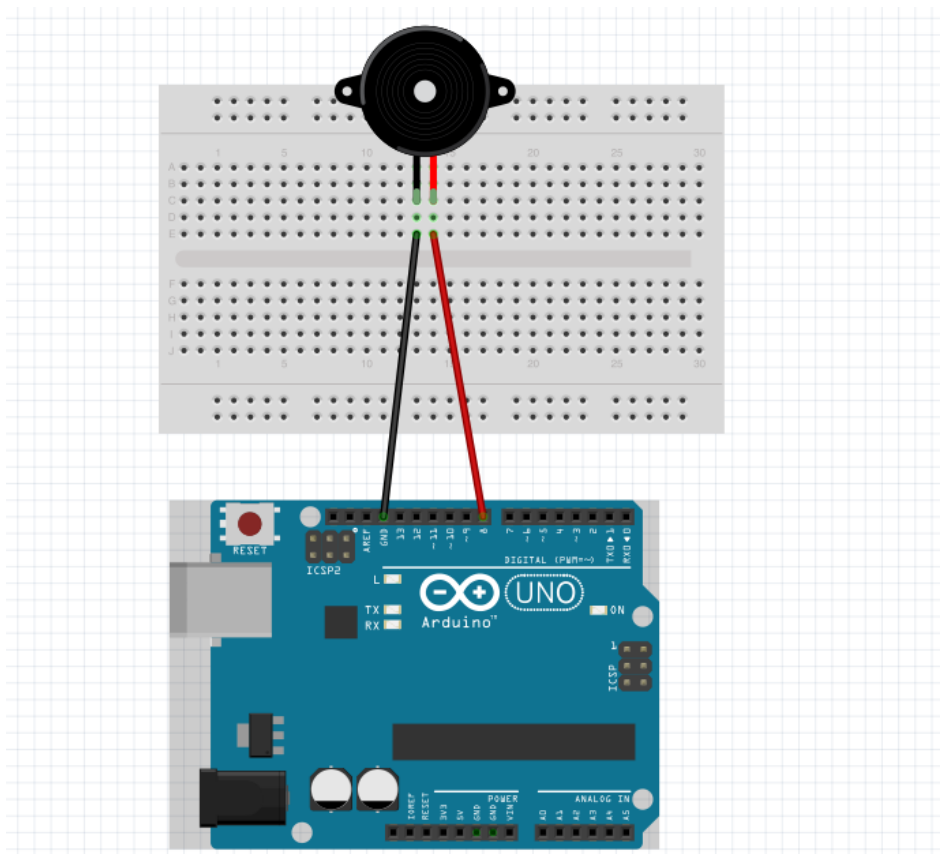


Figure 41: cablage de buzzer avec l'arduino.

VIII/LE Servomoteur :

VIII.1/Définition :

Il existe certains types spéciaux d'applications d'un moteur électrique où la rotation du moteur n'est requise que pour un certain angle. Pour ces applications, nous avons besoin de certains types de moteurs spéciaux avec une disposition spéciale qui fait tourner le moteur d'un certain angle pour une entrée électrique (signal) donnée. À cet effet, le servomoteur entre en scène.

Un servomoteur est un actionneur linéaire ou un actionneur rotatif qui permet un contrôle précis de la position linéaire ou angulaire, de l'accélération et de la vitesse. Il se compose d'un moteur couplé à un capteur pour le retour de position. Il nécessite également un contrôleur relativement sophistiqué, souvent un module dédié spécialement conçu pour être utilisé avec des servomoteurs.

La principale raison derrière l'utilisation d'un servo est qu'il fournit une précision angulaire, c'est-à-dire qu'il ne tournera que autant que nous le voulons, puis s'arrêtera et attendra que le signal suivant prenne d'autres mesures. Le servomoteur est différent d'un moteur électrique standard qui commence à tourner comme lorsque nous lui mettons sous tension, et la rotation continue jusqu'à ce que nous coupions l'alimentation. Nous ne pouvons pas contrôler la progression de rotation du moteur électrique, mais nous ne pouvons contrôler que la vitesse de rotation et pouvons l'activer et le désactiver. Les petits servomoteurs sont inclus dans de nombreux kits de démarrage Arduino pour débutants, car ils sont faciles à utiliser dans le cadre de petits projets électroniques.

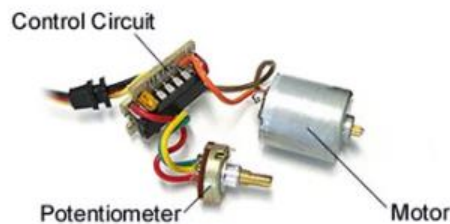
Les servomoteurs sont disponibles en différentes formes et tailles. Nous utiliserons le **servomoteur SG90** dans ce projet. SG90 est un servomoteur à 180 degrés. Ainsi, avec ce servo, nous pouvons positionner l'axe de 0 à 180 degrés.



Figure 42:servomoteur

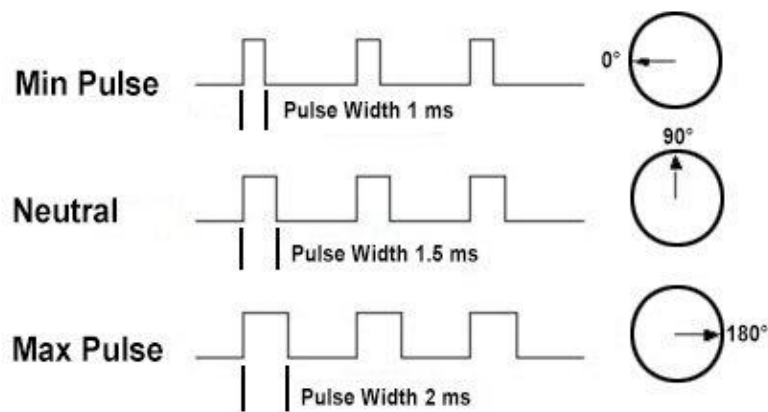
VIII.2/le fonctionnement d'un servomoteur :

Pour bien comprendre le fonctionnement du servo, vous devez jeter un œil sous le capot. À l'intérieur, il y a une configuration assez simple: un petit moteur à courant continu qui peut être piloté par des variations de tension ou par allumage/extinction, un potentiomètre peut détecter la position mécanique de l'arbre, et un circuit de commande. Le moteur est fixé par des engrenages à la molette de commande. Au fur et à mesure que le moteur tourne, la résistance du potentiomètre change, de sorte que le circuit de commande peut réguler avec précision le mouvement et la direction. Lorsque l'arbre du moteur est dans la position souhaitée, l'alimentation fournie au moteur est arrêtée. Sinon, le moteur tourne dans le bon sens. La position souhaitée est envoyée via des impulsions électriques à travers le fil de signal. La vitesse du moteur est proportionnelle à la différence entre sa position réelle et la position souhaitée. Donc, si le moteur est proche de la position souhaitée, il tournera lentement, sinon il tournera rapidement. C'est ce qu'on appelle le contrôle proportionnel. Cela signifie que le moteur ne fonctionnera aussi dur que nécessaire pour accomplir la tâche à accomplir, un petit gars très efficace.



VIII.3/Contrôle des servomoteurs :

Les servos sont contrôlés en envoyant une impulsion électrique de largeur variable, ou modulation de largeur d'impulsion (PWM), à travers le fil de commande. Il y a une impulsion minimale, une impulsion maximale et un taux de répétition. Un servomoteur ne peut généralement tourner que de 90 ° dans les deux sens pour un mouvement total de 180 °. Le PWM envoyé au moteur détermine la position de l'arbre, et en fonction de la durée de l'impulsion envoyée via le fil de commande; le rotor tournera à la position souhaitée. Le servomoteur s'attend à voir une impulsion toutes les 20 millisecondes (ms) et la durée de l'impulsion déterminera jusqu'où le moteur tourne. Par exemple, une impulsion de 1,5 ms fera tourner le moteur en position 90 °. et plus longtemps que 1,5 ms fera tourner le servo dans le sens horaire vers la position 180 °.



VIII.4/Types de servomoteurs :

Les servomoteurs peuvent être de différents types en fonction de leurs applications. Les plus importants d'entre eux sont: servomoteur à courant alternatif, servomoteur à courant continu, servomoteur à courant continu sans balai, servomoteur à rotation positionnelle, servomoteur à rotation continue et servomoteur linéaire.

Un servomoteur typique comprend trois fils, à savoir l'alimentation, la commande et la terre. La forme et la taille de ces moteurs dépendent de leurs applications.

VIII.5/Applications du servomoteur :

Les servos sont utilisés dans les avions radiocommandés pour positionner les surfaces de contrôle comme les ascenseurs, la marche d'un robot ou l'utilisation de pinces. Les servomoteurs sont petits, ont des circuits de commande intégrés et ont une bonne puissance pour leur taille.

Aussi à chaque articulation du robot, nous connectons un servomoteur. Donnant ainsi au bras du robot son angle précis.

Un servomoteur très précis intégré à l'appareil photo corrige l'objectif de l'appareil photo pour affiner les images floues.

Les servomoteurs ajustent l'angle des panneaux solaires tout au long de la journée et, par conséquent, chaque panneau continu de faire face au soleil, ce qui permet de tirer le maximum d'énergie du lever au coucher du soleil.

VIII.6/Câblage de servomoteur avec arduino :

Figure 43:cablage du servomoteur avec arduino.

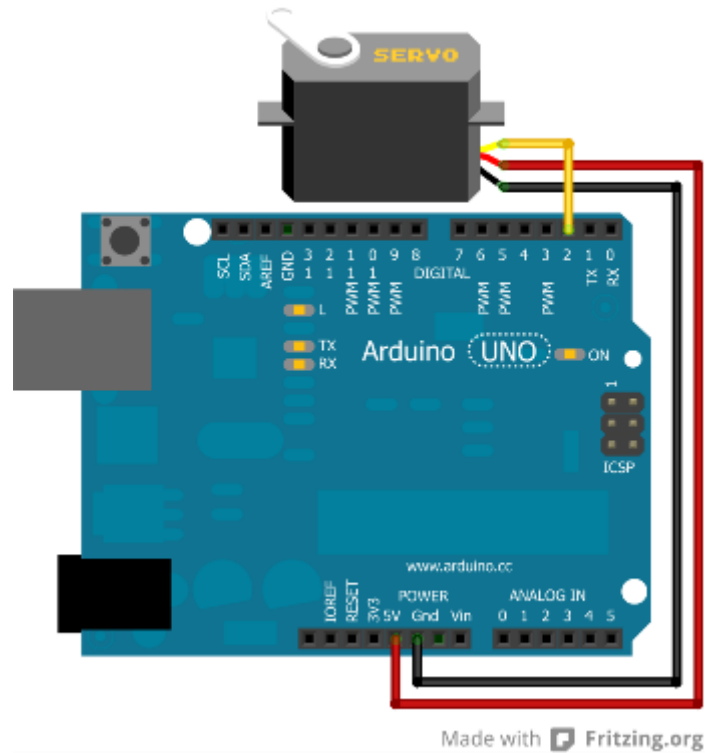


Figure 44: cablage du servomoteur avec arduino.

XI/Conclusion du chapitre :

A fin de ce chapitre, nous avons décrit brièvement l'arduino uno et tous les composants utilisés pour notre projets.

Chapitre 2 :

Fonctionnements et importances des projets

❖ Introduction :

Nous allons expliquer dans ce chapitre le principe de fonctionnement de ces 4 projets (cane blanche intelligente, détecteur de gazes, station météo, poubelle intelligente) utilisant les composants que nous l'avons décrit précédemment dans le chapitre 1, aussi l'importance de ces projets dans la vie quotidienne.

I/ canne blanche intelligente :

I.1/Introduction :

L'œil est l'un des organes importants que Dieu a doté pour l'homme, car il est l'organe responsable du sens de la vision, et grâce à lui une personne peut voir les choses et les couleurs qui les entourent. Sans l'œil, l'homme n'a pas pu développer différentes sciences de l'observation puisque les quatre cinquièmes de ce que nous mémorisons dépendent de ce que nous voyons. L'œil a pour fonction d'enregistrer les sensations lumineuses pour les transmettre au cerveau qui les interprète en images et en couleurs.

Malheureusement ce sens peut être endommagé car l'œil est un organe fragile susceptible d'imperfections dans sa constitution. Selon les statistiques de l'organisation mondiale de la santé il y a environ 285 millions des personnes malvoyantes ou aveugles dans le monde.

En 1969, l'association internationale de prophylaxie de la cécité New Delhi a défini un aveugle par : Personne privée de la vue et, plus précisément, personne empêchée par une baisse de l'acuité visuelle (égale ou inférieure à 1/20) de poursuivre son travail habituel.

Les aveugles et les malvoyants souffrent de difficultés lorsqu'ils se déplacent d'un endroit à un autre et cela conduit à un sentiment d'indépendance, car il fait face à de nombreux obstacles qui entravent leur chemin, tels que collision de choses, trébucher et tomber, ou traverser la route au milieu des voitures et d'autres obstacles qui les obligent à demander l'aide des autres.

Ils ont peu de contact avec l'environnement ils ne peuvent pas distinguer où ils se trouvent.

Un des outils les plus utilisés par les aveugles est la canne blanche qui permet à l'utilisateur de détecter des obstacles qui se trouvent à un mètre de lui environ.

Notre but principal est d'aider un aveugle à vivre une vie meilleure, nous avons été motivé de trouver une meilleure solution pour faciliter aux aveugles le déplacement, donc on a proposé un bâton électronique intelligent à faible coût afin qu'ils puissent se déplacer d'un endroit à un autre de manière simple et plus confortablement car bâton normal n'est pas efficace pour les personnes malvoyantes.

Le système est très utile pour les personnes malvoyantes et qui ont souvent besoin de l'aide des autres.

I.2/Principe de fonctionnement :

Dans notre projet on va faire une étude et réalisation d'un système de canne blanche intelligente qui sert à détecter des obstacles représentant un risque pour l'utilisateur.

Et pour mieux comprendre le principe de fonctionnement de ce système. Nous avons réalisé notre projet à base de carte Arduino (microcontrôleur) qui constituait le programme de guidage avec le capteur ultrason. ce capteur transmet une impulsion sonore à haute fréquence, puis calcule le temps de réception du signal de l'écho sonore pour le refléter. Le capteur a 2 cercles. L'un d'eux agit comme émetteur et transmet les ondes ultrasonores. L'autre agit comme un récepteur (principalement un petit microphone) et reçoit le signal sonore en écho.

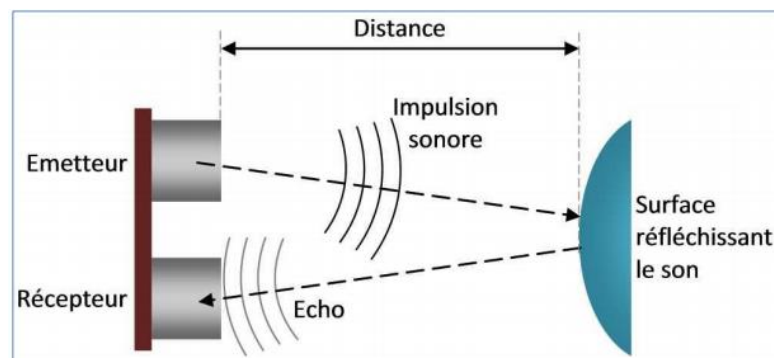


Figure 45:mise en marche de capteur à ultrason.

Le capteur est calibré en fonction de la vitesse du son dans l'air. Avec cette entrée calibrée, la différence de temps entre la transmission et la réception de l'impulsion sonore est déterminée pour calculer la distance de l'objet.

Lorsque l'obstacle est à moins de 5cm l'information sera devenue une tension électrique qui mettra en marche le buzzer et la led pour avertir l'aveugle de la présence de ce type d'obstacles.

I.3/L'importance du projet :

Nous avons conçu et construit un « bâton de marche aveugle à ultrasons » a base d'arduino qui aidera les personnes aveugles à marcher facilement et de façon autonome dans leur vie quotidienne lorsqu'ils se déplacent d'un endroit à l'autre en raison des difficultés qu'ils rencontrent . En version plus simple, nous n'avons utilisé qu'un seul capteur à ultrasons dans ce projet. Pour une meilleure précision et assistance .Le bâton peut détecter automatiquement un obstacle et donner à l'utilisateur une réponse de rétroaction en faisant vibrer le bâton de marche et en émettant un son d'avertissement, ce bâton est à faible cout et une Durée de vie convenable et suffisante, ainsi qu'une source d'énergie facilement remplaçable.

I.4/Conclusion :

Dans ce mémoire, nous nous avons intéressé à une catégorie spéciale de la société qui est les personnes ayant une déficience visuelle, à l'aide de leur témoignages nous avons pu réaliser un cahier de charge pour construire une canne intelligente, Son objectif est de résoudre les problèmes rencontrés par les aveugles dans leur vie quotidienne pour assurer leur sécurité.

On a finalisé ce projet par l'obtention d'un système de guidage simple et bon marché fiable, portable, faible consommation d'énergie et solution robuste pour la navigation avec un temps de réponse court évident.

II/ détecteur de gaz :

II.1/ Introduction :

Il est devenu nécessaire de rendre nos endroits intelligents, car c'est devenu en tendance introduire la technologie à la maison et en bureau pour assurer nos sécurité en installant des systèmes à des endroits adéquats.

Le gaz naturel est composé d'hydrocarbure : méthane, éthane, propane, butane, pentane.il est inflammable et ses produits peuvent être toxiques, s'il y a une combustion dans un milieu fermé le monoxyde de carbone se produira qui est immédiatement absorbée par le système sanguin qui provoque la mort.

En cas de fuite, les gaz se caractérisent par deux risques d'intoxication et un risque d'inflammation et d'explosion du fait de leur capacité à se mélanger facilement à l'air. Les gaz sont très inflammables lorsqu'ils sont mélangés avec de l'air sont très dangereux, ils sont plus lourds que l'air et se déposent au niveau le plus bas et donc ils traversent les sols et restent longtemps et elle a un léger effet anesthésiant, Cela peut provoquer une suffocation si elle

a été inspiré avec une concentration élevée. Et si la concentration du gaz dans un espace clos augmente à plus de 2,5 et qu'il est exposé à une flamme, cela conduit à une explosion, exposant tout le lieu à l'effondrement.

Où la protection civile a enregistré la mort de dizaines d'Algériens par suffocation au gaz et au monoxyde de carbone, notamment en hiver, ce qui est une tragédie que les spécialistes attribuent à l'absence d'une culture de maintenance et de suivi périodique des raccordements de gaz

Plus de 200 décès ont été enregistrés dans les accidents d'asphyxie et ça sur une année entière, au cours de la période de janvier 2017 à janvier 2018.

Plusieurs raisons sont à l'origine de cette catastrophe comme la mauvaise aération, L'ancienneté et le manque d'entretien des appareils électroménagers de gaz tels que le four qui va provoquer asphyxie ,intoxication, brûlure, incendie, explosion .

Pour éviter ces risques, l'atmosphère doit être surveillé régulièrement, car les fuites incontrôlées auront des effets nocifs sur l'homme, par exemple les oxydes de carbone entraînent la suffocation et le gaz de pétrole naturel ou liquéfié entraîne des incendies ou des explosions.

Il faut donc faire appel à la technologie pour prévenir des accidents de fuite de gaz, dans le but de contrôler contre ces risques.

Pour cela notre projet est intitulé « réalisation d'un système d'alarme de fuite de gaz à base d'arduino » pour diminuer les menaces citées en ci-dessus, et on s'est intéressé de faire l'étude sur les deux gaz butane et monoxyde de carbone.

II.2/L'effet du gaz butane au corps humain :

Le butane est un gaz hautement inflammable, incolore et inodore. C'est un hydrocarbure présent dans les produits ménagers et industriels etc.

Le butane est un dépresseur du système nerveux central qui ralentit l'activité du cerveau, affectant les réponses physiques et mentales. Lorsque le butane est inhalé il sera rapidement absorbé par les poumons vers la circulation sanguine. Les produits chimiques sont solubles dans la graisse corporelle et passent rapidement au cerveau et aux organes,

Le butane est hautement inflammable, il existe donc un risque élevé de brûlures ou d'explosions, en particulier pour les fumeurs.

Décès par suffocation ou une forme d'insuffisance cardiaque connue sous le nom de «syndrome de mort subite par reniflement»

Il peut aussi résulter : des Difficultés pour parler, Perte de mémoire courte pouvant être très importante, discours ralenti, arythmie....

II.3/L'effet du gaz CO au corps humain :

Le monoxyde de carbone (CO) est un gaz très dangereux qui est inodore, incolore, il ne peut donc pas être fondue, vu ou goûté Le monoxyde de carbone est produit par l'oxydation partielle des composés contenant du carbone ; il se forme lorsqu'il n'y a pas assez d'oxygène pour produire du dioxyde de carbone (CO)

Les symptômes les plus courants de l'intoxication au CO sont les maux de tête, les nausées, les vomissements, les étourdissements, la fatigue et une sensation de faiblesse. Les signes neurologiques comprennent la confusion, la désorientation, les troubles visuels, la syncope et les convulsions

II.4/Principe de fonctionnement :

Le détecteur de gaz est un appareil fait pour donner une alarme lorsque l'atmosphère devient explosible (présence d'un gaz combustible) ou toxique (présence d'un gaz dangereux pour la santé).

Dans ce projet, nous avons conçu un détecteur de gaz butane et monoxyde de carbone avec Arduino utilisant les capteurs MQ2 et MQ7, ces capteurs détectera les fuites de gaz une fois le système est lancé. Ce système déclenche le buzzer (alarme sonore) et allume la led rouge (alarme visuelle) et actionne le ventilateur d'aération au même temps lorsque le niveau de gazes devient supérieur à la limite souhaitée dans le programme, et affichera à lcd la présence des gaz.

La quantité de gaz dans l'atmosphère est lue en lisant la quantité de tension qui sort à la sortie A0 des capteurs et convertissez-la en un nombre fixe de 0 à 1024 pour l'afficher à l'écran LCD.

Et quand le système ne détecte pas des gaz ou le niveau des gaz et inférieur à la limite souhaitée le buzzer et le ventilateur s'éteignent et la led verte s'allume et LCD affiche l'absence du gaz.

II.5/L'importance de détecteur de fuites des gaz :

L'utilisation du gaz comme source d'énergie est essentielle dans notre vie, mais il y a des risques pour cette utilisation sur la vie humaine en cas de fuites .Les détecteurs de fumée et de fuite de gaz sont très utiles pour détecter la fumée ou le feu dans les bâtiments, tout comme les paramètres de sécurité pour prévenir les catastrophes.

Ce système surveille l'atmosphère de locaux de travail pour limiter tout risque d'incendie ou d'explosion lors de la présence de personnel, Par exemples les Travailleurs d'usine les utilisateurs de laboratoire pourraient empêcher une explosion car ils utilisent des gaz / fumées toxiques.

Ce détecteur a un haut niveau de précision. La précision de détection du butane et du monoxyde de carbone est très élevée, il peut être conçu pour continuer à surveiller l'environnement pendant que les travailleurs travaillent en raison de la faible quantité d'énergie nécessaire car il peut fonctionner avec une alimentation de 5 V tandis que d'autres systèmes ont besoin d'un minimum de 24 V, il a un temps de réponse très vite qui est compris entre 20 et 30 s.

II.6/L'installation le détecteur de gaz :

Le détecteur de gaz utilise de la même façon que les détecteurs de fumée ainsi que l'alarme incendie. Quant à son emplacement, il varie en fonction du type de gaz à identifier.

- Les alarmes anti-méthane sont posées au plafond ;
- Les détecteurs de propane et de butane sont installés au sol.

Dans une habitation, l'alarme à gaz s'installe dans :

- **Les chambres** : bien que les risques d'accidents domestiques soient minimes, le gaz anesthésiant est souvent utilisé par les cambrioleurs pour dévaliser les maisons.
- **Le salon** : il s'agit de l'une des pièces les plus fréquentées de la maison. De ce fait, privilégiez un capteur qui détecte le monoxyde de carbone pour éviter les risques d'asphyxie ;
- **La salle à manger** : les risques d'explosion sont toujours présents dans les endroits situés près de la cuisine.

II.7/Conclusion :

Une des causes les plus mortelles dans notre vie quotidienne c'est inhalation des gaz, ils présentent des risques dangereux sur la vie d'être humain en cas de fuite.

Dans ce cadre, on s'intéresse de présenter une méthodologie pour la détection de fuite de gaz dans des environnements fermés ou intérieurs, pour surveiller la concentration de butane et monoxyde de carbone.

Dans ce contexte, on a réalisé un système intelligent pour la détection des gaz toxiques, cela peut nous aider à prévenir des accidents dans le but de sauver la vie humaine.

III/La station météo :

III.1/Introduction :

A notre époque, il est important d'obtenir les mesures des paramètres environnementaux car on a besoin de savoir si le temps fait chaud ou froid, sec ou humide. L'observation météorologique était très primitive, Cela dépend principalement de l'apparence du ciel et de la sensation de l'air,

On doit être informé par les changements météorologiques dans les domaines suivants : agriculture, l'aviation, la photographie aérienne.

La météorologie est connue comme un ensemble de disciplines scientifiques qui s'intéressent à l'étude des prévisions climatiques et météorologiques, et elle se concentre sur l'étude de plusieurs éléments dans l'atmosphère : la température, la pression atmosphérique et la vapeur d'eau.

La météorologie joue un rôle majeur dans diverses activités économiques et domaines de la vie quotidienne. L'agriculture a été le premier domaine à bénéficier des services météorologiques, évolution des prévisions météorologiques a permis d'éviter de nombreux risques.

Les conditions météorologiques jouent un rôle important dans la détermination de l'emplacement de nombreuses installations économiques, telles que les usines, les aéroports, les ports maritimes, les lignes de transport d'électricité, les transports et la mise en place d'installations hydrauliques.

Dans ce cadre, nous allons réaliser une station météorologique pour mesurer les paramètres environnementaux nécessaire qui sont température humidité pression avec des capteurs spécifiques avec arduino en affichant aussi le temps.

III.2/La température :

La température est définie comme une indication de la quantité d'énergie thermique stockée par n'importe quel corps, car elle est considérée comme un indicateur du mouvement des atomes et des molécules dans le corps, et donc la capacité de déterminer son état interne,. L'unité internationale de température est le kelvin (K). Le degré Celsius (°C) est une autre unité très répandue en Europe.

III.3/L'humidité :

L'humidité c'est la vapeur d'eau dans l'atmosphère, elle est un facteur magique du changement climatique .La vapeur d'eau pénètre dans l'air par l'évaporation de l'eau sur terre et sur mer, et le taux de vapeur d'eau varie d'un endroit à l'autre en fonction du degré de La chaleur, le pourcentage

maximum de vapeur d'eau à une certaine température dans l'atmosphère est appelée air saturé.

L'humidité de l'air absolue donne la quantité d'eau dans une unité de volume d'un mélange gazeux. Son unité est gramme de vapeur par mètre cube d'air.

L'humidité relative est le produit de la division de la quantité de vapeur d'eau à une certaine température dans le gaz, divisée par la quantité de vapeur saturée à cette température, multipliée par 100, son unité est :%.

III.4/La pression :

La force affectant une zone ou une surface spécifique, par un corps fluide, et les fluides sont des liquides et des gaz. La pression est donnée mathématiquement par la relation suivante : $p = f/s$, et son unité est le pascal.

III.5/Principe de fonctionnement d'une station météo :

La station météo est un système a pour mesurer des grandeurs physique comme la température la pression le temps aussi l'humidité à base d'arduino pour nous informer sur la situation météorologique d'une manière régulière.

Dans ce projet nous avons utilisés le capteur dht11 pour la mesure de l'humidité, le capteur bmp280 pour la température et la pression, et le ds3231 pour le temps et les jours.

L'Arduino collecte les données des capteurs, on lit tout d'abord l'heure et la date, puis la température, ensuite l'humidité et à la fin la pression, les données sont affichées sur l'écran LCD, la température est affiché en degré Celsius, l'humidité en pourcentage et la pression en hectopascal.

L'affichage de la date et de l'heure est contrôlé par quatre boutons poussoirs, le premier bouton pour passer à la grandeur destinée ou pour passer aux chiffres suivants de l'heur pour les régler, le 2eme bouton pour éditer, le troisième bouton pour ajouter un nombre à l'heure, le quatrième pour diminuer un nombre et revenir au nombre précédent.

III.6/L'importance d'une station météo :

L'importance de ce système est pour nous informer tous les jours sur les conditions météorologiques pour savoir quel temps il fera dans les prochains

jours, par exemple si le climat est beau ou il pleuve pour prendre nos précautions.

C'est un système très rapide qui change les degrés de température, quantités de pression, humidité automatiquement, rapidement et au même temps.

Il est à faible cout avec des capteurs qui ne coute pas cher et une pile qui conserve les données de jours et de temps plus qu'un an avec la possibilité de l'ajuster à tout moment.

III.7/Conclusion :

Ce projet nous nous sommes intéressés à la réalisation d'une simple station météo avec des capteurs spécifiques : température humidité pression et horloge de temps .nous avons réalisé ce système qui mesure ces paramètre à base d'une carte arduino qui traite les données délivrés par les capteurs.

IV/La poubelle intelligente :

IV.1/Introduction :

La technologie fait partie intégrante de la vie humaine, car la société en dépend aujourd'hui de façon périodique et permanente, car les gens l'utilisent dans tous les aspects de leur vie, elle a provoqué un changement fondamental dans la vie.

Les gens améliorent de plus en plus les différentes choses du ménage. Les robots sont des aspirateurs, des réfrigérateurs intelligents, etc. Mais aujourd'hui, nous allons créer une corbeille qui peut s'ouvrir automatiquement lorsque nous voulons y jeter quelque chose. Car les déchets sont depuis longtemps un grave problème dans la vie de tous les jours.

Chaque jour la pollution de notre environnement augmente, la santé humaine est de plus en plus mise en danger.

Dans ce cadre, nous devons créer une nouvelle innovation afin que nous puissions attirer le public à disposer des déchets à sa place, pour que l'environnement devienne propre et confortable. Pour faire face aux dangers des ordures qui contient beaucoup de germes et de bactéries. En utilisant une carte arduino pour que le couvercle de la poubelle s'ouvre automatiquement.

IV.2/Principe de fonctionnement d'une poubelle intelligente :

La poubelle intelligente est une poubelle qui s'ouvrira automatiquement lorsque vous vous approcherez avec des déchets à base d'arduino.

Ce système est réalisé avec 2 capteur un ultrason HC-SR04 et d'un petit servomoteur 90g.

Nous avons programmé l'arduino pour que s'il voit des déchets (littéralement n'importe quoi) dans une plage de 50 cm il prend des mesures et contrôle le servomoteur, ensuite le servo passe à 50 degrés et frappe le couvercle supérieur du bac avec le soutien du bras étendu, de sorte que le couvercle supérieur s'ouvre et attend ensuite trois secondes, puis tourne automatiquement à 160 degrés et ainsi le couvercle supérieur se ferme.

IV.3/L'importance d'une poubelle intelligente :

La poubelle intelligente, possède des caractéristiques qui permettent de rendre plus facile la vie des personnes qui sont responsables du maintien

Grâce à ce projet, le couvercle de la poubelle reste fermé, afin que les déchets ne soient pas exposés (pour éviter les mouches et les moustiques), et lorsque vous souhaitez éliminer les déchets, il ouvrira automatiquement le couvercle.

Elle nous permet de d'ouvrir la poubelle sans la toucher et saler les mains et sans faire aucun effort.

IV.4/Conclusion :

Le système de poubelle intelligent basé sur Arduino est proposé pour garder notre ville propre, car il est rentable et respectueux de l'environnement car cela réduira les risques et la propagation des maladies infectieuses.

Conclusion du chapitre :

A la fin de ce chapitre on conclut que les projets qu'on a étudié ils sont très pratiques, importants et que nous l'avons besoin dans notre vie.

Chapitre 3 :

Partie réalisations

❖ Introduction :

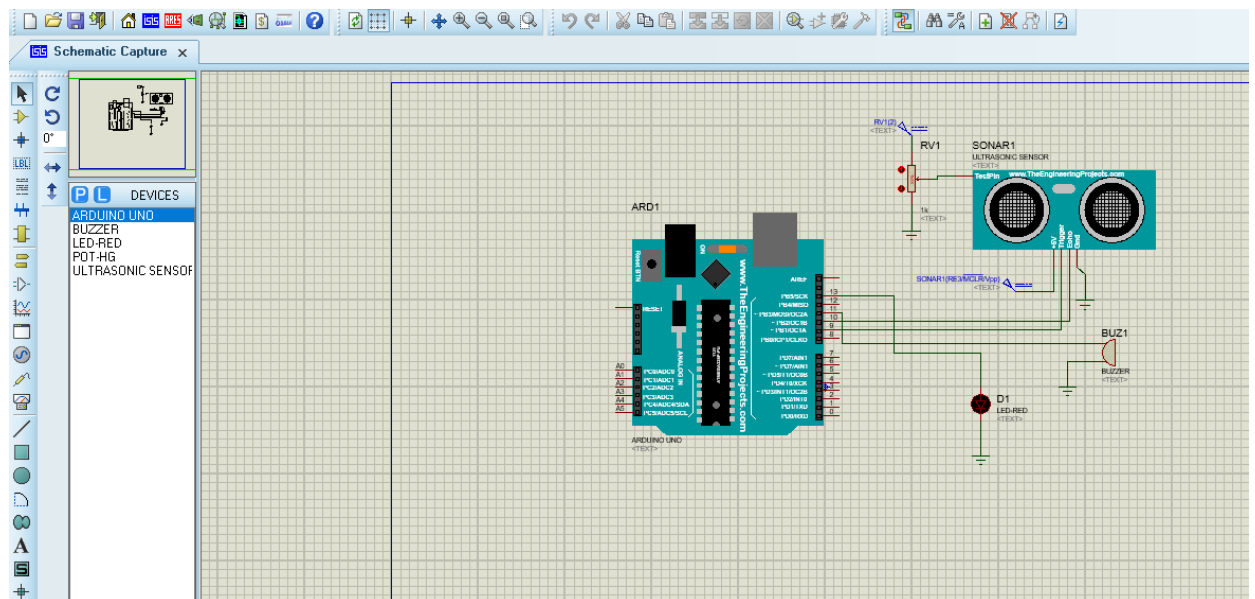
Dans ce chapitre on va faire la partie réalisation de notre projets, tous d'abord on va décrit le brochage entre les composants et les pins d'arduino, ensuite on va montrer les simulations proteus, fritzing de ces projets, à la fin le programme de chaque un.

I/Canne blanche intelligente :

I.1/Réalisation du projet :

En réalisant ce projet, on lie arduino et ultrasonique sur la canne par un fil. on fait lier entre le capteur et arduino comme suivant : vcc avec 5v, gnd avec gnd, écho avec pin 10, trig avec pin 9. On ajoute aussi un buzzer le négative avec GND et le positive avec pin 11, on connectant la led avec pin 13. Ce projet est alimenté avec une batterie de 9v.

I.2/Simulation avec proteus :



I.3/Programme arduino :

```
const int trigPin = 9;  
const int echoPin = 10;  
const int buzzer = 11;  
const int ledPin = 13;
```

```
// defines variables  
long duration;  
int distance;
```



```

int safetyDistance;

void setup() {
  pinMode(trigPin, OUTPUT); // Sets the trigPin as an Output
  pinMode(echoPin, INPUT); // Sets the echoPin as an Input
  pinMode(buzzer, OUTPUT);
  pinMode(ledPin, OUTPUT);
  Serial.begin(9600); // Starts the serial communication
}

void loop() {
  // Clears the trigPin
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);

  // Sets the trigPin on HIGH state for 10 micro seconds
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);

  // Reads the echoPin, returns the sound wave travel time in microseconds
  duration = pulseIn(echoPin, HIGH);

  // Calculating the distance
  distance= duration*0.034/2;

  safetyDistance = distance;
  if (safetyDistance <= 5){
    digitalWrite(buzzer, HIGH);
    digitalWrite(ledPin, HIGH);
  }
  else{
    digitalWrite(buzzer, LOW);
    digitalWrite(ledPin, LOW);
  }

  // Prints the distance on the Serial Monitor
  Serial.print("Distance: ");
  Serial.println(distance);
}

```

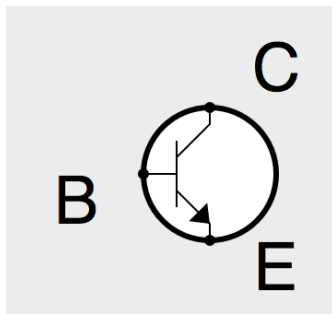
II/Détecteur de gaz :

II.1/Réalisation du projet :

Pour la réalisation de ce projet, nous avons branché les 2 capteurs mq2 et mq7 avec les pins A0 et A1 de l'arduino, le vcc et le gnd de ces capteurs avec le 5v et le gnd de l'arduino, le buzzer avec le pin 9, la led verte avec le pin 12, et la led rouge avec le pin 13, un lcd avec les pins (2,3,4,5,6,7) (RS, E, d4, d5, d6, d7) , un ventilateur avec le pin 11 qui est connecté avec un transistor bd139 et une diode 1N4007. Ces derniers sont connectés avec une résistance de protection.

II.2/Pourquoi nous avons utilisé un transistor et une diode avec le ventilateur ?

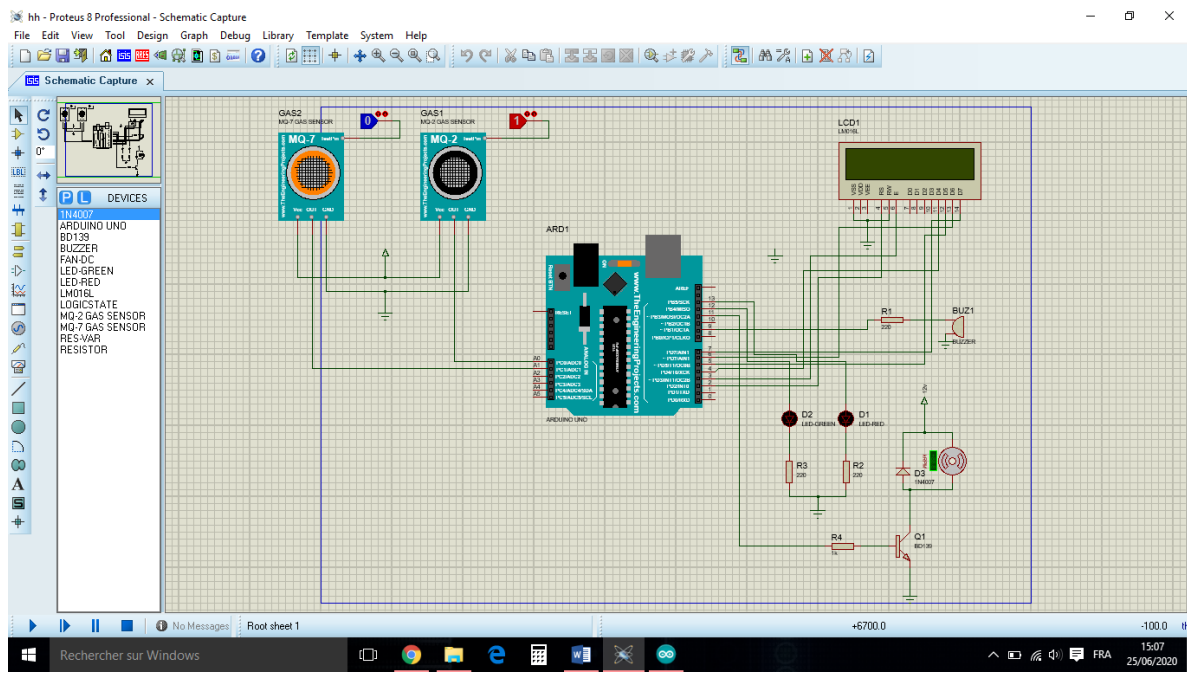
Il nous faut un transistor comme "interface" de puissance. C'est un interrupteur pour laisser passer ou non le courant.



Le B représente la base : c'est ici que l'on va commander le transistor
Le C représente le collecteur : C'est cette borne que l'on va relier au moteur
Le E représente l'émetteur : Cette borne va être reliée à la terre.
Le transistor est commandé par une sortie de la carte Arduino via la résistance sur la base Lorsque l'état de la sortie est au niveau 0, le transistor est bloqué et le courant ne le traverse pas. Le moteur ne tourne pas.

Lorsque la sortie vaut 1, le transistor est commandé et devient saturé c'est-à-dire qu'il laisse passer le courant et le moteur se met à tourner, ce moteur parvient à sa vitesse de rotation qui va générer une tension et un courant à ses bornes alors le transistor sera saturé en lui appliquant une tension sur sa base et ce moteur envoie des parasites et des tensions dangereuses pour le transistor qui risque de le griller pour cela on doit ajouter une diode qui va empêcher un retour de courant lorsque le moteur continue de tourner, car la diode ne laisse passer le courant que dans un sens. Du coup le transistor (qui sera en mode bloqué) enverra le courant vers la diode (qui ne l'autorisera que dans un sens). Ce courant viendra en opposition du courant induit (qui est dû à l'inertie du moteur qui s'arrête) et du coup viendra freiner le moteur.

II.3/Simulation avec proteus :



II.4/Programme arduino :

```

#include <LiquidCrystal.h>
LiquidCrystal lcd(2, 3, 4, 5, 6, 7);
#define buzzerPin 9
#define MotorPin 11
#define greenLed 12
#define redLed 13
#define gasPinMQ2 A0
#define gasPinMQ7 A1
void setup() {
//Serial.begin(9600);
lcd.begin(16,2);
pinMode (buzzerPin, OUTPUT);
pinMode(MotorPin, OUTPUT);
pinMode(redLed, OUTPUT);
pinMode (greenLed, OUTPUT) ;
pinMode (gasPinMQ2, INPUT);
pinMode(gasPinMQ7, INPUT);
lcd.setCursor(0,0);
lcd.print ("CALIBRAGE MQx");
for(int i = 0; i <10; i++){
if (i==4)
{
lcd.setCursor(0,1);
lcd.print(".");
}
else lcd.print(".");
delay(500);
}
lcd.setCursor(5, 1);

```

```

lcd.print("PRET");
delay(1000);
lcd.clear();
lcd.setCursor(0,0);
lcd.print("MQ2 ACTIF");
lcd.setCursor(0, 1);
lcd.print("MQ7 ACTIF");
delay(1500);
lcd.clear ();

}
void loop() {
int gasSensorMQ2 = analogRead(gasPinMQ2);
int gasSensorMQ7 = analogRead(gasPinMQ7);
if (gasSensorMQ2 > 350 || gasSensorMQ7 > 350)
{
digitalWrite(buzzerPin,HIGH);
digitalWrite(MotorPin, HIGH);
digitalWrite(greenLed,LOW);
digitalWrite(redLed,HIGH);
lcd.setCursor(0,0);
lcd.print("MQ2=");
lcd.print(gasSensorMQ2);
Serial.print(gasSensorMQ2);
lcd.setCursor(0,1);
lcd.print("MQ7=");
lcd.print(gasSensorMQ7);
Serial.print(gasSensorMQ7);
//Serial.print("\t");
delay(2000);
lcd.clear();
lcd.setCursor(0,1);
//Serial.println("Gas is Detected");
lcd.print("DETECTION GAZ");
delay(1000);
lcd.clear();
}
else if (gasSensorMQ2 < 350 && gasSensorMQ7 < 350)
{
digitalWrite(buzzerPin,LOW);
digitalWrite(MotorPin,LOW);
digitalWrite(greenLed,HIGH);
digitalWrite(redLed,LOW) ;
lcd.setCursor(0,0);
lcd.print("MQ2=");
lcd.print(gasSensorMQ2);
//Serial.print(gasSensorMQ2);
lcd.setCursor(0, 1);
lcd.print("MQ7=");
lcd.print(gasSensorMQ7);
}
}

```

```

//Serial.print(gasSensorMQ7);
//Serial.print("\t");
delay(2000);
lcd.clear();
lcd.setCursor(10,0);
//Serial.println("No Gas");
lcd.print("NO GAZ");
lcd.setCursor(10,1);
//Serial.println("No Gas");
lcd.print("NO GAZ");
delay(300);
}
}

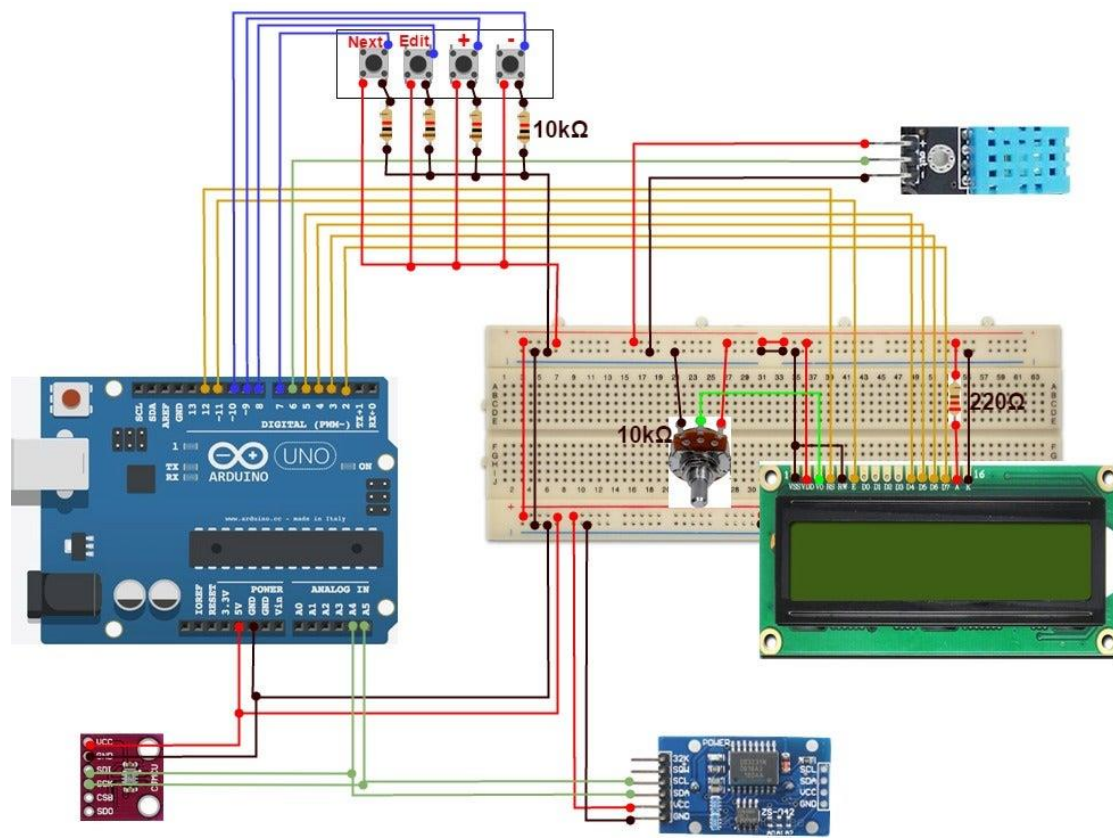
```

III/Station météo :

III.1/Réalisation du projet :

Tous d'abord nous avons connectés les pins SDA de bmp280 et de DS3231 avec le pin A4 de l'arduino et les pins SCL de ces 2 capteurs avec l'A5 de l'arduino, le pin data du capteur dht11 avec le pin 6 de l'arduino, les boutons poussoirs avec les pins 7, 8, 9,10 ,et lcd avec les pins 12, 11, 5, 4, 3, 2 ces boutons sont connectés avec des résistances pour la protection, les vcc et les gnd de ces composants sont connectés avec le 5v et le gnd de l'arduino.

III.2/Câblage fritzing :



III.3/Programme arduino :

```
//DHT11 temperature & humidity sensor
#include "dht11.h"
//LCD display
#include <LiquidCrystal.h>
//DS3231 real time clock module
#include "DS3231.h"
//I2C interface used for BMS280 and DS3231
#include "Wire.h"
//BMP280 Barometric Sensor
#include "Adafruit_BMP280.h"
//Humidity and Temperature sensor object DHT11
dht11 DHT11;
//Barometric sensor BMP280
Adafruit_BMP280 bmp;
//Real Time clock DS3231
DS3231 Clock;
//DHT11 module is using D6
#define DHT11PIN 6
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
//Next Button Pin
const int nextButton = 7;
const int editButton = 8;
const int increaseButton = 9;
const int decreaseButton = 10;
//DHT11 sensor variables and functions
int checkDHT11(void);
//BMP280 variables
float BMP_pressure;
float BMP_temperature;
int BMP_altimeter;
void getBMP280Values(void);
//DS3231 variables and functions
int second,minute,hour,date,month,year,DStemperature;
void ReadDS3231(void);
bool Century=false;
bool h12;
bool PM;
byte ADay, AHour, AMinute, ASecond, ABits;
bool ADy, A12h, Apm;
byte Byear, Bmonth, Bdate, BDoW, Bhour, Bminute, Bsecond;
//Intro variables
int startPoint;
int endPoint;
int i, j;
int speed = 50;
// text to display
String txtMsg = "Entropy";
String txtMsg1 = "Station";
```

```

#define DisplayDelay 1000
#define EditDelay 1000
void DisplayIntro(void);
char *getDayofweek(int d, int m, int y);
int isLeapYear(int y);
//General purpose variables
int displayValue=0;
bool edit=false;
//print data from all sensors
void printAll(void);
//print time from DS3231
void printTime(void);
void printHourMinute(void);
//print date from DS3231
void printDate(void);
//print temperature from DHT11
void printTemperature(void);
//print humidity from DHT11
void printHumidity(void);
//print barometer from BMP280
void printBarometer(void);
void setup() {

//set the buttons
  pinMode(nextButton, INPUT);
  pinMode(editButton, INPUT);
  pinMode(increaseButton, INPUT);
  pinMode(descreaseButton, INPUT);
// set up the LCD's number of columns and rows:
  lcd.begin(16, 2);
//initiate serial
  Serial.begin(115200);
//initiate I2C interface
  Wire.begin();
//Begin the BMP280 sensor
  bmp.begin();
//Display the "entropy station" intro
  DisplayIntro();
//clear the LCD display
  lcd.clear();
}
void loop() {
if ( digitalRead(editButton) == HIGH) { //check Edit mode
  edit=!edit;
  editValue=0;
  displayValue=0;
  delay(EditDelay);
}
if(edit==false) //edit mode is off, display the values from sensors

```

```

{
  //no Blinking when displaying the data
  lcd.noCursor();
  if( digitalRead(nextButton) ==HIGH){
    displayValue++;
  }
  switch(displayValue){
    case 0:    //Print data from all sensors
      lcd.clear();
      printAll();
      break;
    case 1:    //print Time
      ReadDS3231();
      lcd.clear();
      printTime();
      break;
    case 2:    //print Date
      ReadDS3231();
      lcd.clear();
      printDate();
      break;
    case 3:    //Print Temperature from DHT11
      checkDHT11();
      lcd.clear();
      printTemperature();
      break;
    case 4:    //Print Humidity from DHT11
      checkDHT11();
      lcd.clear();
      printHumidity();
      break;
    case 5:    //Print Barometric Pressure from BMP280
      getBMP280Values();
      lcd.clear();
      printBarometer();
      break;
    default:
      lcd.clear();
      displayValue=0;
      printAll();
      break;
  }
  delay(DisplayDelay);
}
else //the edit mode is on, setting the date&time
{
  delay(EditDelay);
  lcd.clear();
  if( digitalRead(nextButton) ==HIGH){ //selecting which value to modify
    editValue++;
  }
}

```



```

}
switch(editValue){
  case 0:
    lcd.noCursor();
    ReadDS3231(); //read data from DS3231 (time and date)
    printHourMinute();
    //lcd.setCursor(5, 0);
    if( digitalRead(increaseButton) ==HIGH){
      hour++;
      if(hour>23) hour=0;
    }
    if( digitalRead(descreaseButton) ==HIGH){
      hour--;
      if(hour<0) hour=23;
    }
    Clock.setHour(hour);
    printHourMinute();
    lcd.setCursor(5, 0);
    lcd.cursor();
  break;
  case 1:
    lcd.noCursor();
    ReadDS3231();
    printHourMinute();
    //lcd.setCursor(8, 0);
    if( digitalRead(increaseButton) ==HIGH){
      minute++;
      if(minute>59) minute=0;
    }
    if( digitalRead(descreaseButton) ==HIGH){
      minute--;
      if(minute<0) minute=59;
    }
    Clock.setMinute(minute);
    printHourMinute();
    lcd.setCursor(8, 0);
    lcd.cursor();
  break;
  case 2:
    lcd.noCursor();
    ReadDS3231();
    printDate();
    //lcd.setCursor(9, 0);
    if( digitalRead(increaseButton) ==HIGH){
      date++;
      if(month == 1 || month == 3 || month == 5 || month == 7 || month == 8 ||
month == 10 || month == 12)
      {
        if (date>31) date=1;
      }
    }

```

```

else if(month == 4 || month == 6 || month == 9 || month == 11 )
{
  if (date>30) date=1;
}
else if(month==2)
{
  if(isLeapYear(year)==1)
  {
    if (date>29) date=1;
  }
  else if(isLeapYear(year)==0)
  {
    if (date>28) date=1;
  }
}
}
}
if( digitalRead(decreaseButton) ==HIGH){
  date--;
  if(month == 1 || month == 3 || month == 5 || month == 7 || month == 8 ||
month == 10 || month == 12)
  {
    if (date<1) date=31;
  }
  else if(month == 4 || month == 6 || month == 9 || month == 11 ) {
    if (date<1) date=30;
  }
  else if(month==2)
  {
    if(isLeapYear(year)==1)
    {
      if (date<1) date=29;
    }
    else if(isLeapYear(year)==0)
    {
      if (date<1) date=28;
    }
  }
}
}
Clock.setDate(date);
printDate();
lcd.setCursor(5, 0);
lcd.cursor();
break;
case 3:          //Edit Month
  lcd.noCursor();
  ReadDS3231(); //read data from DS3231 (time and date)
  printDate();

  if( digitalRead(increaseButton) ==HIGH){
    month++;

```

```

        if (month>12) month=1;
    }
    if( digitalRead(decreaseButton) ==HIGH){
        month--;
        if (month<1) month=12;
    }
    Clock.setMonth(month);
    printDate();
    lcd.setCursor(8, 0);
    lcd.cursor();
break;
case 4:
    lcd.noCursor();
    ReadDS3231(); //read data from DS3231 (time and date)
    printDate();
    //lcd.setCursor(9, 0);
    if( digitalRead(increaseButton) ==HIGH){
        year++;
        if (year>99) year=1;
    }
    if( digitalRead(decreaseButton) ==HIGH){
        year--;
        if (year<1) year=99;
    }

    Clock.setYear(year);
    printDate();
    lcd.setCursor(11, 0);
    lcd.cursor();
break;
default:
    editValue=0;
    lcd.clear();
    lcd.setCursor(3,0);
    lcd.print("Time & Date");
break;
}
}
}
//print barometer from BMP280
void printBarometer()
{
//get value from sensor
lcd.setCursor(2, 0);
    lcd.print("Barometer:");
    lcd.setCursor(5, 1);
    lcd.print((int)BMP_pressure);
    lcd.print("hPa");
}
}

```

```

//print date from DS3231
void printDate()
{
//read data from DS3231 (time and date)
//date
  lcd.setCursor(4, 0);
  if(date<10) lcd.print('0');
  lcd.print(date);
  lcd.print('-');
  if(month<10) lcd.print('0');
  lcd.print(month);
  lcd.print('-');
  if(year<10) lcd.print('0');
  lcd.print(year);
  lcd.setCursor(4, 1);
  lcd.print(getDayofweek(date, month, year+2000));
}
//print time from DS3231
void printTime()
{
//read data from DS3231 (time and date)
//TIME
  lcd.setCursor(4, 0);
  if(hour<10) lcd.print('0');
  lcd.print(hour);
  lcd.print(':');
  if(minute<10) lcd.print('0');
  lcd.print(minute);
  lcd.print(':');
  if(second<10) lcd.print('0');
  lcd.print(second);
}
void printHourMinute()
{
//read data from DS3231 (time and date)
//TIME
  lcd.setCursor(4, 0);
  if(hour<10) lcd.print('0');
  lcd.print(hour);
  lcd.print(':');
  if(minute<10) lcd.print('0');
  lcd.print(minute);
  lcd.print(':');
  if(second<10) lcd.print('0');
  lcd.print(second);
}
//print data from all sensors
void printAll(void)
{
  //read data from DHT11 (temperature and Humidity)

```

```

    if (checkDHT11()== 1)
    {
        lcd.print("DHT11 error");
    }
// read data from DS3231 (time & date )
ReadDS3231();
// read data from BMP280 (barometer)
getBMP280Values();
//TIME
lcd.setCursor(1, 0);
if(hour<10) lcd.print('0');
lcd.print(hour);
lcd.print(':');
if(minute<10) lcd.print('0');
lcd.print(minute);
//DATE
lcd.setCursor(7, 0);
if(date<10) lcd.print('0');
lcd.print(date);
lcd.print('-');
if(month<10) lcd.print('0');
lcd.print(month);
lcd.print('-');
if(year<10) lcd.print('0');
lcd.print(year);
//Temperature
lcd.setCursor(0, 1);
lcd.print(DHT11.temperature);
// lcd.print((char)178);
lcd.print('C');
//Humidity
lcd.setCursor(4, 1);
//lcd.print(DHT11.humidity);
lcd.print(DHT11.humidity);
lcd.print('%');
//Barometer
lcd.setCursor(9, 1);
lcd.print((int)BMP_pressure);
lcd.print("hPa");
}
void DisplayIntro(void)
{
    startPoint = 0; //set starting point
    endPoint = 12; //set ending point
    lcd.clear();
//for each letter of the string starting from the last one.
for (i = txtMsg.length() - 1; i >= 0; i--)
{
    startPoint = 0;
    for (j = 0; j < endPoint; j++)

```

```

    {
    lcd.setCursor(startPoint, 0);
    lcd.print(txtMsg[i]);
    delay(speed);
    if (startPoint != endPoint - 1) {
        lcd.setCursor(startPoint, 0);
        lcd.print(' ');
    }
    startPoint++;
    }
    startPoint = 0;
    for (j = 0; j < endPoint; j++)
    {
    lcd.setCursor(startPoint, 1);
    lcd.print(txtMsg1[i]);
    delay(speed);

    if (startPoint != endPoint - 1) {
        lcd.setCursor(startPoint, 1);
        lcd.print(' ');
    }
    startPoint++;
    }

    endPoint--;
    delay(speed);
    }
// hold the string on the display for 2 sec.
delay(2000);
}
* read values from DS3231
void ReadDS3231(void)
{
    second=Clock.getSecond();
    minute=Clock.getMinute();
    hour=Clock.getHour(h12, PM);
    date=Clock.getDate();
    month=Clock.getMonth(Century);
    year=Clock.getYear();
    DStemperature=Clock.getTemperature();
}
void getBMP280Values(void)
{
    //Read values from the sensor:
    BMP_pressure = bmp.readPressure()/100;
    BMP_temperature = bmp.readTemperature();
    BMP_altimeter = bmp.readAltitude (1003);
}
* print Temperature on the LCD display from DHT11
*/

```

```

void printTemperature()
{
  lcd.setCursor(2, 0);
  lcd.print("Temperature:");

  lcd.setCursor(5, 1);
  lcd.print(DHT11.temperature);
  //lcd.print((char)178);
  lcd.print('C');
}
* print Humidity on the LCD display from DHT11
void printHumidity()
{
  lcd.setCursor(2, 0);
  lcd.print("Humidity:");

  lcd.setCursor(5, 1);
  lcd.print(DHT11.humidity);
  lcd.print('%');

}
//Reading the DHT11 sensor condition
int checkDHT11()
{
  int chk = DHT11.read(DHT11PIN);

  Serial.print("Reading DHT11 sensor: ");
  switch (chk)
  {
    case DHTLIB_OK:
      Serial.println("DHT11:OK\n");
      return 0;
      break;
    case DHTLIB_ERROR_CHECKSUM:
      Serial.println("DHT11:Checksum error\n");
      return 1;
      break;
    case DHTLIB_ERROR_TIMEOUT:
      Serial.println("DHT11:Time out error\n");
      return 1;
      break;
    default:
      Serial.println("DHT11:Unknown error\n");
      return 1;
      break;
  }
}

char *getDayofweek(int d, int m, int y)

```

```

{
  static int t[] = { 0, 3, 2, 5, 0, 3, 5, 1, 4, 6, 2, 4 };
  int day;
  y -= m < 3;
  day = ( y + y/4 - y/100 + y/400 + t[m-1] + d ) % 7;
  switch(day){
    case 0 :return("Sunday");
    case 1 :return("Monday");
    case 2 :return("Tuesday");
    case 3 :return("Wednesday");
    case 4 :return("Thursday");
    case 5 :return("Friday");
    case 6 :return("Saturday");
    default:return("Error: Invalid Argument Passed");
  }
}
int isLeapYear(int y)
{
  if ( y%400 == 0)
    return 1;
  else if ( y%100 == 0)
    return 0;
  else if ( year%4 == 0 )
    return 1;
  else
    return 0;

  return 0;
}

```

IV/Poubelle intelligente :

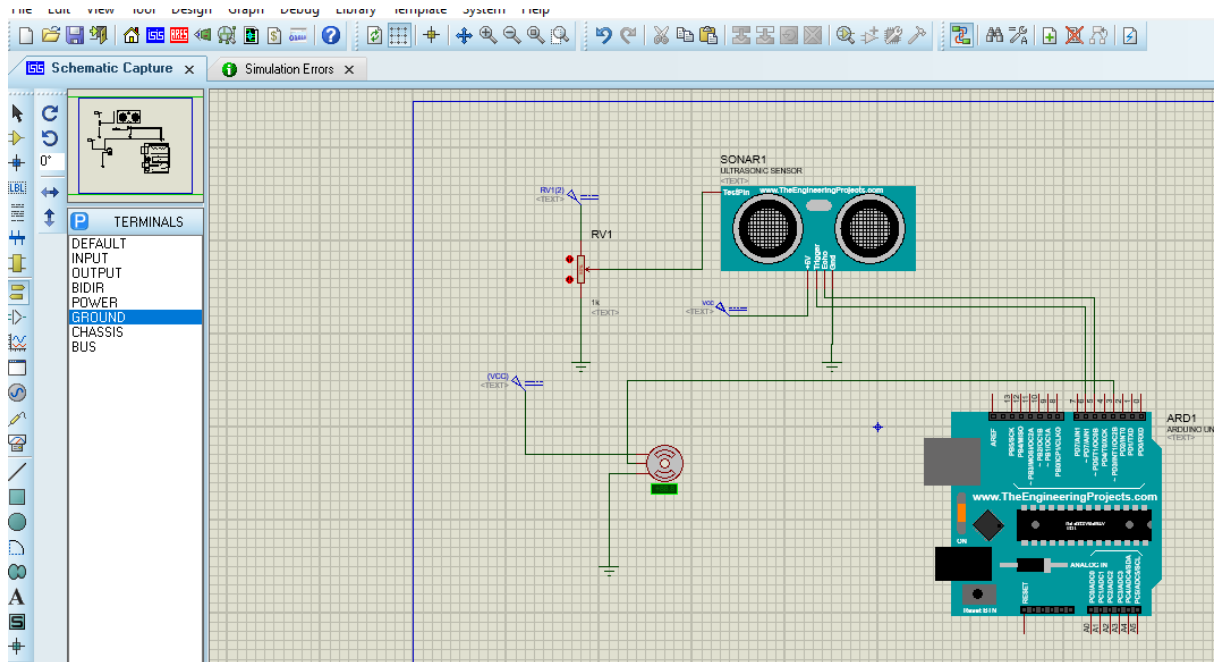
IV.1/Réalisation du projet :

Ce projet est simple à réaliser à partir d'une boîte en carton avec carte Arduino UNO, et un capteur de distance à ultrason HC-SR04 et d'un petit servo moteur.

On va faire lier le GND du capteur arduino avec le capteur à ultrason, le pin 6 avec le trig, le pin 5 avec l'écho, et le 5v d'arduino avec 5 v du capteur.

On relie le servo moteur avec le 5v d'arduino aussi et un fil vers le GND, le troisième fil avec pin 3 d'arduino.

IV.2/Simulation proteus :



IV.3/Programme arduino :

```
#include<Servo.h>
Servo servo;
int const trigPin = 6;
int const echoPin = 5;
```

```
void setup()
```

```
{
pinMode(trigPin, OUTPUT);
pinMode(echoPin, INPUT);
servo.attach(3);
```

```
}
```

```
void loop()
```

```
{
int duration, distance;
digitalWrite(trigPin, HIGH);
delay(1);
digitalWrite(trigPin, LOW);
// Measure the pulse input in echo pin
duration = pulseIn(echoPin, HIGH);
// Distance is half the duration divided by 29.1 (from datasheet)
distance = (duration/2) / 29.1;
// if distance less than 0.5 meter and more than 0 (0 or less means over
range)
if (distance <= 50 && distance >= 0) {
servo.write(50);
delay(3000);
}
```

```
else {  
  
servo.write(160);  
}  
// Waiting 60 ms  
delay(60);  
}
```

Conclusion du chapitre :

L'expérience menée durant ce chapitre nous a permis de réaliser plusieurs projets facilement avec la simulation proteus, fritzing et la programmation arduino.

Conclusion générale

Au bout de notre cursus en master "Instrumentation" nous avons été chargés de réaliser un projet de fin d'études. Notre travail consisté en la réalisation de plusieurs mini projets à base d'arduino.

L'élaboration de ce travail dans le cadre du projet fin d'étude, nous a permis d'approfondir nos connaissances théoriques en électronique et d'acquérir une bonne expérience au niveau pratique. Nous avons essayé de réaliser un détecteur de gaz, une canne intelligente, une station météo, et une poubelle intelligente.

Ce travail constitue un des multiples aspects de richesse des applications de l'outil informatique couplé avec l'électronique. En effet, il permet de satisfaire déjà le besoin principal pour chaque projet.

Ce projet nous à donner une meilleur idée sur la complémentarité entre le volet théorique et le volet pratique.

Nous souhaitons vivement que ce projet puisse servir comme élément de base pour d'autres étudiants pour réalisés ces projet ou prendre une idée sur l'arduino et comment le programmer ou l'utilisé avec d'autre projets.

Bibliographie

- [1]Boudjedir Imen ,these de master, Ain el Beida en génie électrique .2017.
- [2]Journal of physics :conference series .volume 1120.the 8th internationnal conference on Theoretical and applied physics 20-21 september 2018.medan.indonesia.
- [3]International Journal of Innovative Technology and Exploring Engineering (IJITEE) ISSN: 2278-3075.Volume-8. April 2019.
- [4] International Journal of Engineering and Advanced Technology (IJEAT) ISSN: 2249 – 8958, Volume 8. December 2018.
- [5]<http://www.iosrjournals.org>.
- [6]anwar,ashraf, a smart stick for assisting for Blind people . IOSR Journal of Computer Engineering.(pp.68-90).2017.Saoudi arabia.
- [7]G.SRINIVAS, G.M.RAJU, D.RAMESH, S.SIVARAM, SMART BLIND STICK CONNECTED SYSTEM USING ARDUINO.volume 6.2019.
- [8]Prof. Dr. K. Srinivasa Rao .Internet of Things (IOT) Based Weather Monitoring system.volume 5.india.2016.
- [9] REGUIG BERRA Imadeddine, ALLAM Yassine, Etude et réalisation d'une station météo.mémoire de master. UNIVERSITE MOHAMED BOUDIAF - M'SILA ,2017.
- [10] Karthik Krishnamurthi. Suraj Thapa, Lokesh Kothari.Arun Prakash.Arduino Based Weather Monitoring System.volume 3.india,2015.
- [11]MAAFA ABDELKADER, Réalisation d'une station météorologique. mémoire de master.université badji mokhtar annaba.2018.

[12] K. Kranthi Kumar, Dr.E. Ramaraj, Dr.P. Geetha. IOT BASED TRASH COLLECTION BIN USING ARDUINO.journal of critical reviews.volume 7.india.2020.

[13] Rhonnel S. Paculanan, Israel Carino. "LPG Leakage Detector using Arduino with SMS Alert and Sound Alarm. Blue Eyes Intelligence Engineering Retrieval Number.volume 8.2019.

[14] Balamurugan S, Abhishek Ajith, Snehal Ratnakaran, S. Balaji, R. Marimuthu. Design of Smart Waste Management System .School of Electrical Engineering.india.