

*Chapitre II*  
*Arduino et capteurs*

## Introduction

Aujourd'hui, l'électronique est de plus en plus remplacée par de l'électronique programmée. On parle de système embarqué ou d'informatique embarquée. Son but est de simplifier les schémas électroniques et par conséquent de réduire l'utilisation de composants électroniques, réduisant ainsi le coût de fabrication d'un produit. Il en résulte des systèmes plus complexes et performants pour un espace réduit.

Depuis que l'électronique existe, sa croissance est fulgurante et continue encore aujourd'hui. L'électronique est devenue accessible à toutes personnes ayant l'envie d'y accéder. Ce que nous allons apprendre dans ce travail est un mélange d'électronique et de programmation. Nous allons en effet parler d'électronique embarquée qui est un sous domaine de l'électronique et qui a l'habileté d'unir la puissance de la programmation à la puissance de l'électronique.

### I.1. Définition du module Arduino

Le module Arduino est un circuit imprimé en matériel libre, issu du travail d'une équipe d'enseignants et d'étudiants de l'école de Design d'interaction d'ivrea, les plans de la carte elle-même sont publiés en licence libre dont certains composants de la carte, comme le microcontrôleur et les composants complémentaires qui ne sont pas en licence libre. Un microcontrôleur programmé peut analyser et produire des signaux électriques de manière à effectuer des tâches très diverses. Arduino est utilisé dans beaucoup d'applications comme l'électrotechnique industrielle et embarquée; le modélisme, la domotique mais aussi dans des domaines différents comme l'art contemporain et le pilotage d'un robot, commande des moteurs et faire des jeux de lumières, communiquer avec l'ordinateur, commander des appareils mobiles (modélisme). Chaque module d'Arduino possède un régulateur de tension +5 V et un oscillateur à quartz 16 MHz (ou un résonateur céramique dans certains modèles). Pour programmer cette carte, on utilise l'logiciel IDE Arduino.

Les projets Arduino peuvent être autonomes, comme ils peuvent communiquer avec d'autres logiciels installés sur l'ordinateur tel que Flash, processing ou MaxMPS, Matlab.

Ces cartes sont faites à base d'une interface entrée/sortie simple et d'un environnement de développement proche du langage [7][8].

## I.2. Les gammes de la carte Arduino

Actuellement, il existe plus de 20 versions de module Arduino, nous citons quelques-unes afin d'éclaircir l'évaluation de ce produit scientifique et académique:

- Le NG d'Arduino, avec une interface d'USB pour programmer et usage d'un ATmega8.
- L'extrémité d'Arduino, avec une interface d'USB pour programmer et usage d'un Microcontrôleur ATmega8.
- L'Arduino Mini, une version miniature de l'Arduino en utilisant un microcontrôleur ATmega168.
- L'Arduino Nano, une petite carte programme à l'aide porte USB cette version utilisant un microcontrôleur ATmega168 (ATmega328 pour une plus nouvelle version).
- Le LilyPad Arduino, une conception de minimaliste pour l'application wearable en utilisant un microcontrôleur ATmega168.
- Le NG d'Arduino plus, avec une interface d'USB pour programmer et usage d'un ATmega168.
- L'Arduino Bluetooth, avec une interface de Bluetooth pour programmer en utilisant un microcontrôleur ATmega168.
- L'Arduino Diecimila, avec une interface d'USB et utilise un microcontrôleur ATmega168.
- L'Arduino Duemilanove ("2009"), en utilisant un microcontrôleur l'ATmega168 (ATmega328 pour une plus nouvelle version) et actionné par l'intermédiaire de la puissance d'USB/DC.
- L'Arduino Mega, en utilisant un microcontrôleur ATmega1280 pour I/O additionnel et mémoire.
- L'Arduino UNO, utilisations microcontrôleur ATmega328.
- L'Arduino Mega2560, utilisations un microcontrôleur ATmega2560, et possède toute la mémoire à 256 KBS. Elle incorpore également le nouvel ATmega8U2 (ATmega16U2 dans le jeu de puces d'USB de révision 3).
- L'Arduino Leonardo, avec un morceau ATmega3U4 qui élimine le besoin de raccordement d'USB et peut être employé comme clavier [9].

Parmi ces types, nous avons choisi une carte Arduino Mega. L'intérêt principal de cette carte est de faciliter la mise en œuvre d'une telle commande qui sera détaillée par la suite.

### I.3.La carte Arduino Mega 2560

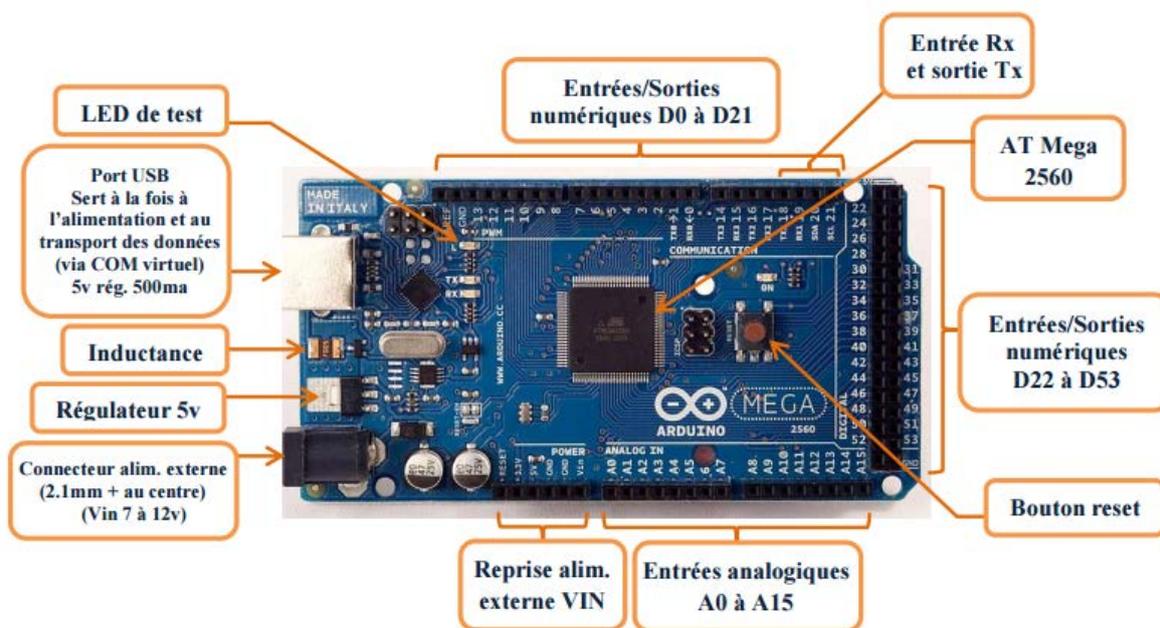


Figure 2: Description de la carte Arduino MEGA 2560

La carte Arduino Mega 2560 est une carte à microcontrôleur basée sur un ATmega2560. Cette carte dispose :

- de 54 broches numériques d'entrées/sorties (dont 14 peuvent être utilisées en sorties PWM (MLI : Modulation de largeur d'impulsion)).
- de 16 entrées analogiques (qui peuvent être utilisées en broches entrées/sorties numériques).
- de 4 UART (port série matériel).
- d'un quartz de 16Mhz.
- d'une connexion USB.
- d'un connecteur d'alimentation jack.
- d'un connecteur ICSP (programmation "in-circuit").
- d'un bouton de réinitialisation (reset).

Elle contient tout ce qui est nécessaire pour le fonctionnement du microcontrôleur ; Pour pouvoir l'utiliser, il suffit simplement de la connecter à un ordinateur à l'aide d'un câble USB (ou de l'alimenter avec un adaptateur secteur ou une pile, mais ceci n'est pas indispensable,

l'alimentation étant fournie par le port USB). La carte Arduino Mega 2560 est compatible avec les circuits imprimés prévus pour les cartes Arduino Uno, Duemilanove ou Diecimila.

## I.4.Caractéristique technique de la carte Arduino Mega 2560

Un module Arduino est généralement construit autour d'un microcontrôleur ATMEL AVR, et de composants complémentaires qui facilitent la programmation et l'interfaçage avec d'autres circuits. Chaque module possède au moins un régulateur linéaire 5V et un oscillateur à quartz 16 MHz (ou un résonateur céramique dans certains modèles). Le microcontrôleur est préprogrammé avec un bootloader de façon à ce qu'un programmeur dédié ne soit pas nécessaire.

### I.4.1 Partie matérielle

Généralement tout module électronique qui possède une interface de programmation est basé toujours dans sa construction sur un circuit programmable ou plus.

Microcontrôleur	ATMEGA2560
Tension de fonctionnement	5V
Tension d'alimentation	7 à 12V
Broches E/S numérique	54 (dont 14 disposent de sortie PWM)
Broches d'entrées analogiques	16
Vitesse d'horloge	16 MHz
Mémoire programme Flash	25 6KB dont 8 KB utilisés en bootloader
Mémoire SRAM	8 KB
Mémoire EEPROM	4 KB

Tableau 4: Constitution de la carte Arduino Mega 2560

#### I.4.1.1.Le Microcontrôleur ATmega2560

Un microcontrôleur ATmega2560 est un circuit intégré qui rassemble sur une puce plusieurs éléments complexes dans un espace réduit, c'est le processeur de la carte, s'occupe

de tout ce qui est calculs, exécution des instructions du programme et gestion des ports d'entrée/sortie [10].



Figure 3: Microcontrôleur ATmega2560

### I.4.1.2. Les mémoires

L'ATmega 2560 à 256Ko de mémoire FLASH pour stocker le programme (dont 8Ko également utilisés par le bootloader), également 8 ko de mémoire SRAM (volatile) et 4Ko d'EEPROM (non volatile - mémoire qui peut être lue à l'aide de la librairie EEPROM).

### I.4.1.3. Les sources de l'alimentation de la carte

La carte Arduino Mega 2560 peut être alimentée soit via la connexion USB (qui fournit 5V jusqu'à 500mA) ou à l'aide d'une alimentation externe. La source d'alimentation est sélectionnée automatiquement par la carte.

La carte peut fonctionner avec une alimentation externe de 6 à 20 volts. Cependant, si la carte est alimentée avec moins de 7V, la broche 5V pourrait fournir moins de 5V, et la carte pourrait être instable. Si on utilise plus de 12V, le régulateur de tension de la carte pourrait chauffer ce dernier. Les broches d'alimentation sont les suivantes :

- **VIN** : La tension d'entrée positive lorsque la carte Arduino est utilisée avec une source de tension externe (à distinguer du 5V de la connexion USB ou autre source 5V régulée).
- **5V**. La tension régulée utilisée pour faire fonctionner le microcontrôleur et les autres composants de la carte. Le 5V régulé fourni par cette broche peut donc provenir soit de

la tension d'alimentation VIN via le régulateur de la carte, ou bien de la connexion USB ou de tout autre source d'alimentation régulée.

- **3.3V** : Une alimentation de 3.3V fournie par le circuit intégré FTDI (circuit intégré faisant l'adaptation du signal entre le port USB de votre ordinateur et le port série de l'ATmega) de la carte. L'intensité maximale disponible sur cette broche est de 50mA
- **GND** : Broche de masse (0V).

#### I.4.1.4. Entrées et sorties numériques

Chacune des 54 broches numériques de la carte Mega peut être utilisée soit comme une entrée numérique, soit comme une sortie numérique. Ces broches fonctionnent en 5V. Chaque broche peut fournir ou recevoir un maximum de 40mA d'intensité et dispose d'une résistance interne de "rappel au plus" (pull-up) (déconnectée par défaut) de 20-50 KOhms. Cette résistance interne s'active sur une broche en entrée à l'aide de l'instruction `digitalWrite(broche, HIGH)`.

De plus, certaines broches ont des fonctions spécialisées :

- **Communication Série**: Port Série Serial : 0 (RX) et 1 (TX); Port Série Serial 1: 19 (RX) and 18 (TX); Port Série Serial 2: 17 (RX) and 16 (TX); Port Série Serial 3: 15 (RX) and 14 (TX).
- **Interruptions Externes**: Broches 2 (interrupt 0), 3 (interrupt 1), 18 (interrupt 5), 19 (interrupt 4), 20 (interrupt 3), et 21 (interrupt 2). Ces broches peuvent être configurées pour déclencher une interruption sur une valeur basse, sur un front montant ou descendant, ou sur un changement de valeur.
- **Impulsion PWM** (largeur d'impulsion modulée): Broches 0 à 13. Fournissent une impulsion PWM 8-bits à l'aide de l'instruction `analogWrite()`.
- **SPI** (Interface Série Périphérique): Broches 50 (MISO), 51 (MOSI), 52 (SCK), 53 (SS). Ces broches supportent la communication SPI (Interface Série Périphérique).
- **I2C**: Broches 20 (SDA) et 21 (SCL). Supportent les communications de protocole I2C.
- **LED**: Broche 13. Il y a une LED incluse dans la carte connectée à la broche 13.

### I.4.1.5. Broches analogiques

La carte Mega2560 dispose de 16 entrées analogiques, chacune pouvant fournir une mesure d'une résolution de 10 bits (1024 niveaux soit de 0 à 1023) à l'aide de la très utile fonction `analogRead()` du langage Arduino. Par défaut, ces broches mesurent entre le 0V (valeur 0) et le 5V (valeur 1023). Les broches analogiques peuvent être utilisées en tant que broches numériques.

### I.4.1.6. Autres broches

Il y a deux autres broches disponibles sur la carte :

- **AREF** : Tension de référence pour les entrées analogiques (si différent du 5V). Utilisée avec l'instruction `analogReference()`.
- **Reset** : Mettre cette broche au niveau BAS entraîne la réinitialisation du microcontrôleur. Comme un port de communication virtuel pour le logiciel sur l'ordinateur, La connexion série de l'Arduino est très pratique pour communiquer avec un PC, mais son inconvénient est le câble USB, pour éviter cela, il existe différentes méthodes pour utiliser ce dernier sans fil.

## I.5. Les Accessoires de la carte Arduino

La carte Arduino généralement est associée aux accessoires qui simplifient les réalisations.

### I.5.1. Communication

Le constructeur a suggéré qu'une telle carte doit être dotée de plusieurs ports de communications ; on peut éclaircir actuellement quelques types.

#### I.5.1.1. Le module Arduino Bluetooth

Le Module Microcontrôleur Arduino Bluetooth est la plateforme populaire Arduino avec une connexion série Bluetooth à la place d'une connexion USB, très faible consommation d'énergie mais aussi très faible portée (sur un rayon de l'ordre d'une dizaine de mètres), faible débit, très bon marché et peu encombrant.



Figure 4: Module bluetooth pour Arduino

### I.5.1.2. Le module shield Arduino Wifi

Le module Shield Arduino Wifi permet de connecter une carte Arduino à un réseau internet sans fil Wifi.



Figure5: Module wifi pour Arduino

### I.5.1.3. Le Module XBee

Ce module permet de faire de la transmission sans fil, faible distance /consommation /débit/ prix.



Figure 6:Module XBee pour Arduino

## I.5.2. Les drivers

Il existe plusieurs drivers comme des cartes auxiliaires qui peuvent être attachées avec l'Arduino afin de faciliter la commande ; on peut citer quelques types.

### I.5.2.1. Les afficheurs LCD

Les afficheurs LCD sont devenus indispensables dans les systèmes techniques qui nécessitent l'affichage des paramètres de fonctionnement. Ces Afficheurs permettent d'afficher des lettres, des chiffres et quelques caractères spéciaux. Les caractères sont prédéfinis.

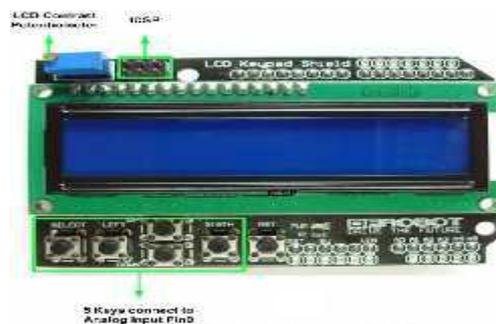


Figure 7:Afficheurs LCD

### I.5.2.2. Le relais

C'est un composant qui possède une bobine (électro-aimant) qui est parcourue par un courant électrique agissant sur un ou plusieurs contacts. Le relais est une solution à la commande en puissance. Il assure en outre une isolation galvanique en mettant en œuvre un mouvement mécanique [12].



Figure 8: relais

## II. Les capteurs

Un capteur est une interface entre un processus physique et une information manipulable. Il ne mesure rien, mais fournit une information en fonction de la sollicitation à laquelle il est soumis. Il fournit cette information grâce à une électronique à laquelle il est associé, il existe trois familles de capteurs : [11] [13]

- **Capteurs TOR** : Permettent de détecter un événement ou un objet, le signal électrique en sortie est de type logique (2niveaux logiques 0 ou 1).
- **Capteurs Numériques** : La sortie est une séquence d'états logiques, elle peut prendre une infinité de valeurs discrètes.
- **Capteurs Analogiques** : La sortie est une grandeur électrique dont la valeur est une fonction de la grandeur physique mesurée par le capteur, elle peut prendre une infinité de valeurs continues.

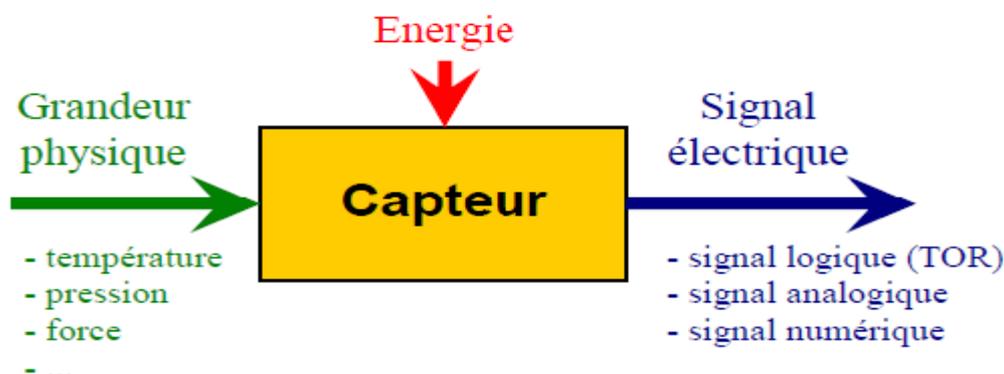


Figure 9: Fonctionnement d'un capteur

### II.1. Capteur de température et d'humidité DHT11

Ce capteur combiné d'humidité relative et de température DHT11 est un appareil de mesure facile à utiliser avec un prix abordable, des températures de 0 à +50°C avec une précision de +/- 2°C et des taux d'humidité relative de 20 à 80% avec une précision de +/- 5% avec une mesure toute les deux secondes. Le DHT11 et ses quatre broches à dimensions standard le rendent à la fois soudable et enfichable.

On peut rencontrer deux types différents du DHT11, un type à quatre broches et l'autre type a trois broches monté sur une petite PCB. La version montée sur PCB est plus pratique car elle comprend une résistance « pull-up » de 10K Ohm montée en surface pour la ligne de signal. Voici les sorties pour les deux versions:

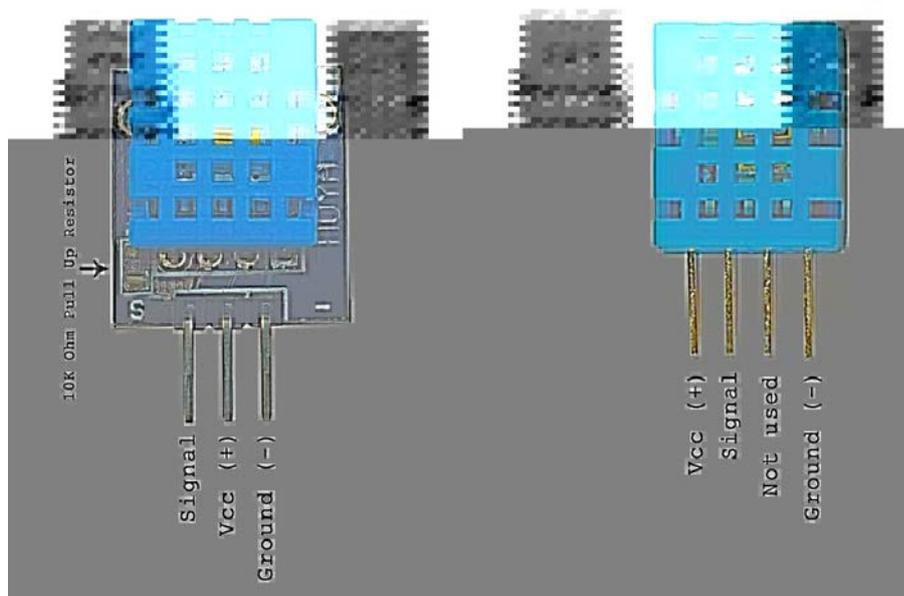


Figure 10: les deux version du capteur DHT11

## II. 2. Capteur de mouvement PIR

Les capteurs PIR (capteur infrarouge passif) permettent de détecter la présence d'humains mobiles dans le champ du capteur. Ils sont utilisés dans divers systèmes de sécurité et détecteurs de mouvements. On en trouve à bas coût, et ils sont très simples à utiliser.

Ces capteurs n'émettent rien, mais au contraire captent les rayonnements infrarouges émis, les êtres vivants à sang chaud passant dans le champ de détection du capteur. Pour élargir la zone de détection, ceux-ci sont souvent recouverts d'une lentille de Fresnel.

Des composants électroniques divers traitent la sortie brute du capteur en lui-même et permettent d'obtenir une sortie numérique en fonction du mouvement détecté.

Pour ce qui est du brochage, ce capteur est composé de trois broches :

- 5V.
- GND.
- DATA.

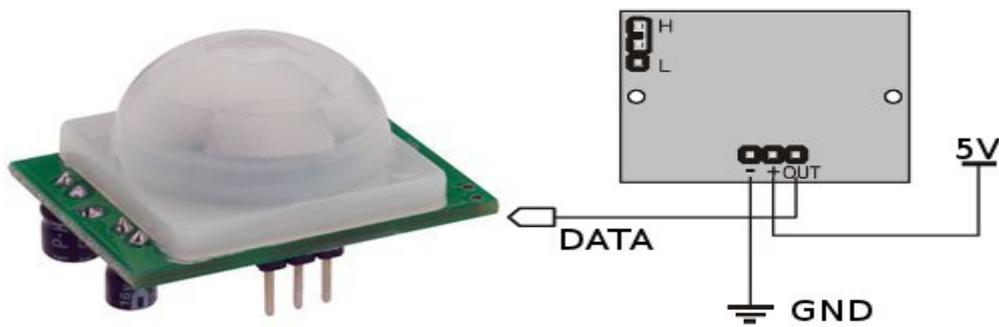


Figure 11: Capteur de mouvement PIR

### II.3. Capteur de GAZ MQ2

Le MQ2 est un capteur qui permet de détecter du gaz ou de la fumée à des concentrations de 300 ppm à 10000 ppm. Après calibration, le MQ-2 peut détecter différents gaz comme le GPL (LPG), l'i-butane, le propane, le monoxyde de carbone, le méthane, l'alcool, l'hydrogène ainsi que les fumées. Il est conçu pour un usage intérieur à température ambiante.

Le MQ2 doit être alimenté en 5V pour le capteur physico-chimique puisse atteindre sa température de fonctionnement. Il dispose d'une sortie analogique et d'un réglage de la sensibilité par potentiomètre.



Figure 12: Capteur de Gaz MQ2

## II.4. Capteur de lumière

Une photorésistance est un composant dont la résistivité dépend de la luminosité. Pour faire simple, c'est une résistance dont la valeur change en fonction de la lumière qu'elle reçoit.

Il existe différents types de photorésistances, chacune ayant des valeurs de résistance différentes en fonction de la luminosité ambiante. Le type le plus classique de photorésistances est de 1M ohms (obscurité) jusqu'à 12K ohms (pleine lumière).

Quand une photorésistance est illuminée, sa résistance diminue. On peut donc utiliser une photorésistance pour mesurer la luminosité ambiante.

Nous pouvons utiliser les photorésistances dans divers domaines et applications, voici quelques exemples :

- Détection du jour et de la nuit.
- Mesure de luminosité ambiante (pour ajuster un éclairage par exemple).
- Suiveur de lumière (pour panneaux solaires, robots, etc..).

Et voici ci-dessous une photorésistance et son symbole. (Figure II 13)

Ainsi que des courbes de la résistivité de chaque type de photorésistance en fonction de la luminescence, où on remarque que plus la luminescence est plus grande la résistivité diminue.

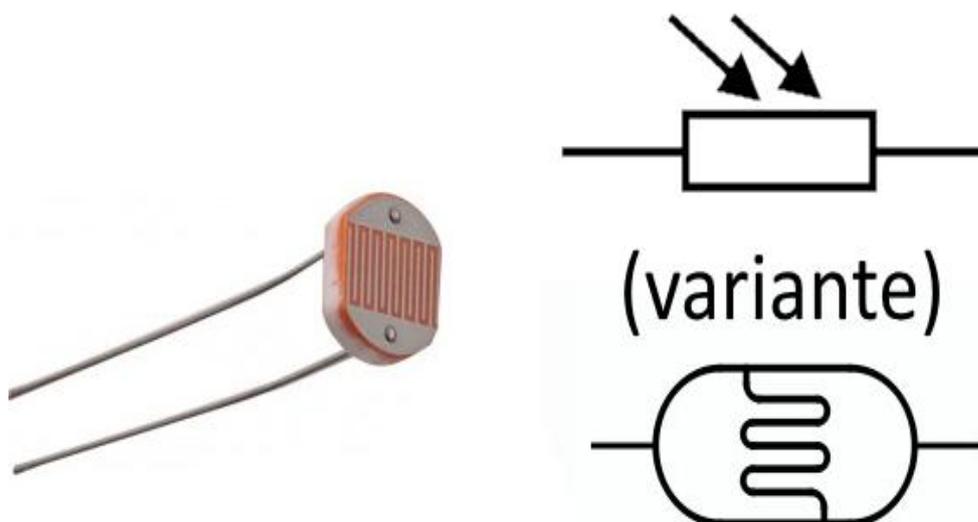


Figure 13: Potorésistance et symbole.

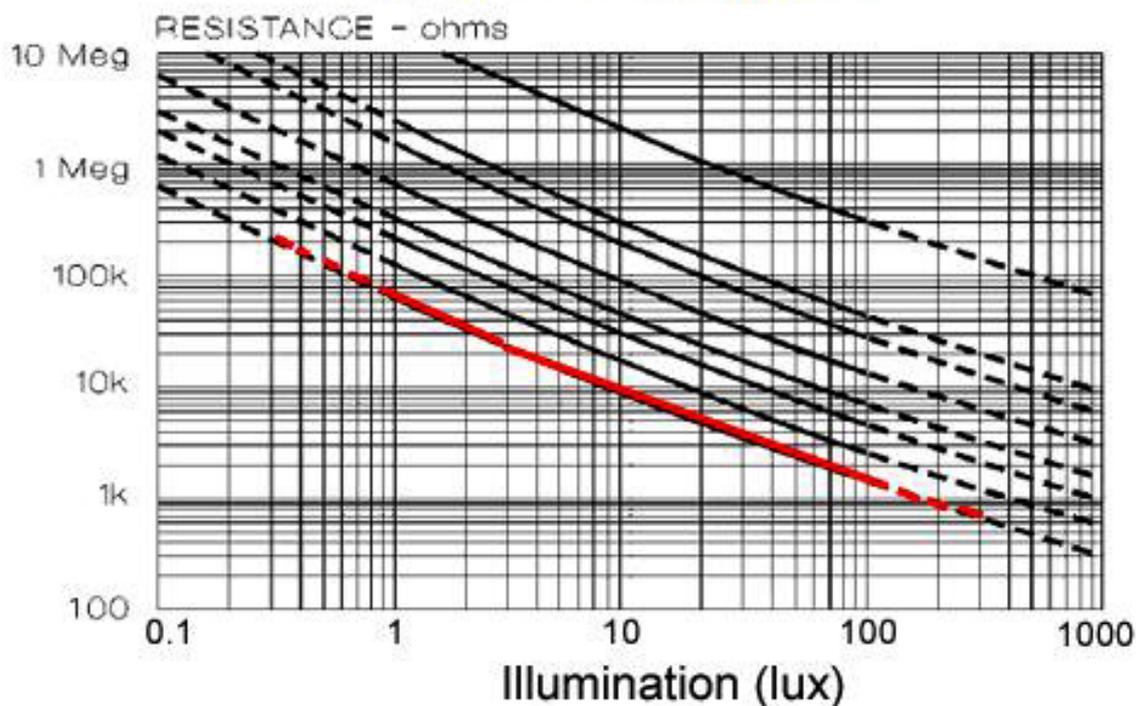


Figure 14: Courbe lumière VS résistance

### III. Partie programme

Une telle carte d'acquisition qui se base sur sa construction sur un microcontrôleur doit être dotée d'une interface de programmation comme est le cas de notre carte. L'environnement de programmation open-source pour Arduino peut être téléchargé gratuitement (pour Mac OS X, Windows, et Linux).

### L'interface

L'interface du logiciel se présente de la façon suivante :

1. options de configuration du logiciel.
2. boutons pour la programmation des cartes.
3. Zone pour programmer.
4. débogueur (affichage des erreurs de programmation).
5. partie déclaration de variables (globales).

6. Initialisation « Void Setup () {} » : Au démarrage de l'Arduino toutes les instructions comprise entre les deux accolades seront exécuter qu'une seul fois.

7. Boucle principale « Voidloop(){} » : Les instruction sont répéter indéfiniment tant que l'Arduino fonctionne.

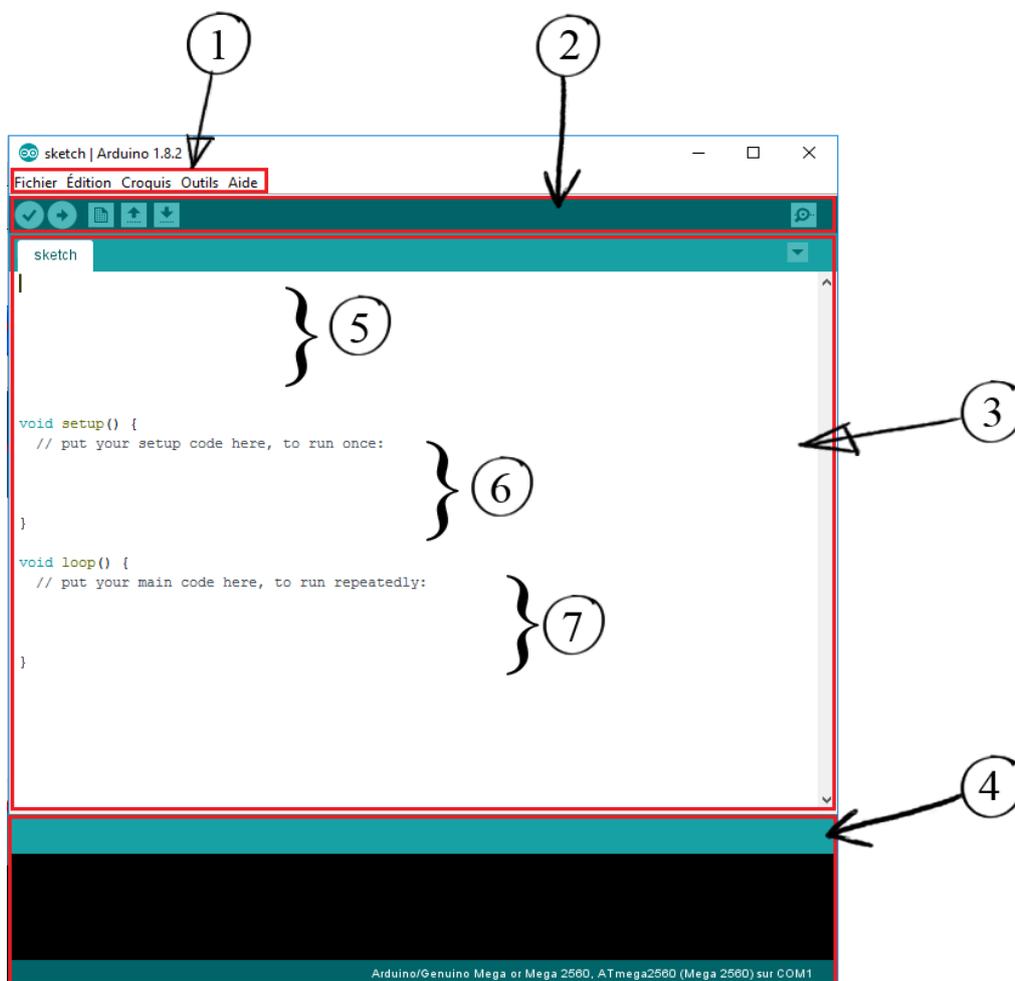


Figure 15 :L'interface du logiciel Arduino

**Les boutons du logiciel**



Figure 16:les boutons du logiciel de programmation

1. Vérifier : permet de vérifier le programme, il actionne un module qui cherche les erreurs dans le programme.
2. Téléverser : compiler et envoyer le programme vers la carte.
3. Nouveau : créer un nouveau fichier.
4. charger un programme existant.
5. Sauvegarder le programme en cours.
6. Moniteur série : de base sur la carte Arduino on ne peut pas afficher de texte, il faut ajouter un module d'affichage ou bien se servir du moniteur série pour utiliser l'écran de notre ordinateur pour savoir où on en est dans l'exécution du programme

### **III.1. L'environnement de la programmation**

Le logiciel de programmation de la carte Arduino sert d'éditeur de code (langage proche du C). Une fois, le programme tapé ou modifié au clavier, il sera transféré et mémorisé dans la carte à travers de la liaison USB. Le câble USB alimente à la fois en énergie la carte et transporte aussi l'information ce programme appelé IDE Arduino.

### **III.2. Structure générale du programme (IDE Arduino)**

Comme n'importe quel langage de programmation, une interface souple et simple est exécutable sur n'importe quel système d'exploitation Arduino basé sur la programmation en C.

### **III.3. Injection du programme**

Avant d'envoyer un programme dans la carte, il est nécessaire de sélectionner le type de la carte (Arduino Mega2560) et le numéro de port USB (COM).

### **III.4. Description du programme**

Un programme Arduino est une suite d'instructions élémentaires sous forme textuelle (ligne par ligne). La carte lit puis effectue les instructions les unes après les autres dans l'ordre défini par les lignes de codes.

Les commentaires sont, en programmation informatique, des portions du code source ignorées par le compilateur ou l'interpréteur, car ils ne sont pas censés influencer l'exécution du programme.

**Définition des variables :** Pour notre montage, on va utiliser une sortie numérique de la carte qui est par exemple la troisième sortie numérique ; cette variable doit être définie et nommée; la syntaxe est pour désigner un nombre entier est **int**.

#### **Configuration des entres et des sorties void setup ()**

Les broches numériques de l'Arduino peuvent aussi bien être configurées en entrées numériques ou en sorties numériques; ici on configure pin en sortie ; pin mode (nom, état).l'etat est soit OUTPUT pour les sorties ou INPUT lorsqu'il s'agit d'entrées.

#### **Programmation des interactions voidloop :**

Dans cette boucle, on définit les opérations à effectuer dans l'ordre digitalwrite(nom, état) est une autre des quatre fonctions relatives aux entrées – sorties numériques.

- delay(temps en mili-seconde) est la commande d'attente entredeux instructions.
- chaque ligne d'instruction est terminée par un point-virgule.
- ne pas oublier les accolades qui encadrent la boucle.

### **III.5. Les étapes de téléchargement du programme**

Une simple manipulation enchaînée doit être suivie afin d'injecter un code vers la carte Arduino via le port USB.

1. On conçoit ou on ouvre un programme existant avec le logiciel IDE Arduino.
2. On vérifie ce programme avec le logiciel Arduino (compilation).
3. Si des erreurs sont signalées, on modifie le programme.
4. On charge le programme sur la carte.
5. On câble le montage électronique.
6. L'exécution du programme est automatique après quelques secondes.
7. On alimente la carte soit par le port USB, soit par une source d'alimentation autonome (pile 9 volts par exemple).

On vérifie que notre montage fonctionne

### **Conclusion**

Dans ce chapitre, nous avons projeté la lumière sur une carte contrôle qui est l'Arduino donnant ainsi les raisons pour lesquelles on l'a choisie, puis nous avons cité des différents types de cette dernière. Ensuite, nous avons expliqué les deux parties essentielles de l'Arduino; la partie matérielle et la partie programmation. Nous avons également expliqué les différentes options.

Nous avons aussi expliqué le fonctionnement des capteurs et la façon de les intégrer ainsi que leurs accessoires à notre système.

***Chapitre III***  
***Conception et réalisation***

## Introduction

Dans ce chapitre, nous nous intéresserons à la programmation de la carte Arduino et la simulation du programme, à l'aide du logiciel de simulation ISIS/PROTEUS, avec les différentes bibliothèques, ainsi que la réalisation de la carte.

Le travail qui porte sur la réalisation d'un système de distribution d'énergie pour une maison intelligente nécessite plusieurs capteurs et mécanismes. La programmation de la carte se fait à l'aide du programme ARDUINO.

Une fois la programmation terminée, nous passerons à la simulation grâce au programme ISIS PROTEUS munis de différentes bibliothèques préalablement téléchargées et installées pour la carte ARDUINO et les capteurs.

## Les différentes étapes de la réalisation

Notre travail est basé sur quatre grands axes :

- L'étude du système.
- La programmation du logiciel ARDUINO.
- La simulation avec ISIS PROTEUS.
- La réalisation de la carte.

### I. Etude du système

L'étude de notre système est divisée en 4 parties:

- Détection des gaz dangereux.
- Température.
- Humidité.
- Lumière.
- Détection de présence.

#### I.1. Détecteur de gaz dangereux

Le détecteur de gaz est une partie essentielle dans notre travail. Aujourd'hui, ce type de dispositif est essentiel dans une maison. En effet, il peut réduire le risque de mourir par asphyxie au monoxyde de carbone. Ce capteur est relié à la carte Arduino qui déclenche l'ouverture des fenêtres et la fermeture de la vanne principale de gaz dès l'approche du seuil de gaz maximal

autorisé. C'est aussi grâce à ce programme que les occupants peuvent être prévenus du danger encouru par le déclenchement d'une alarme.

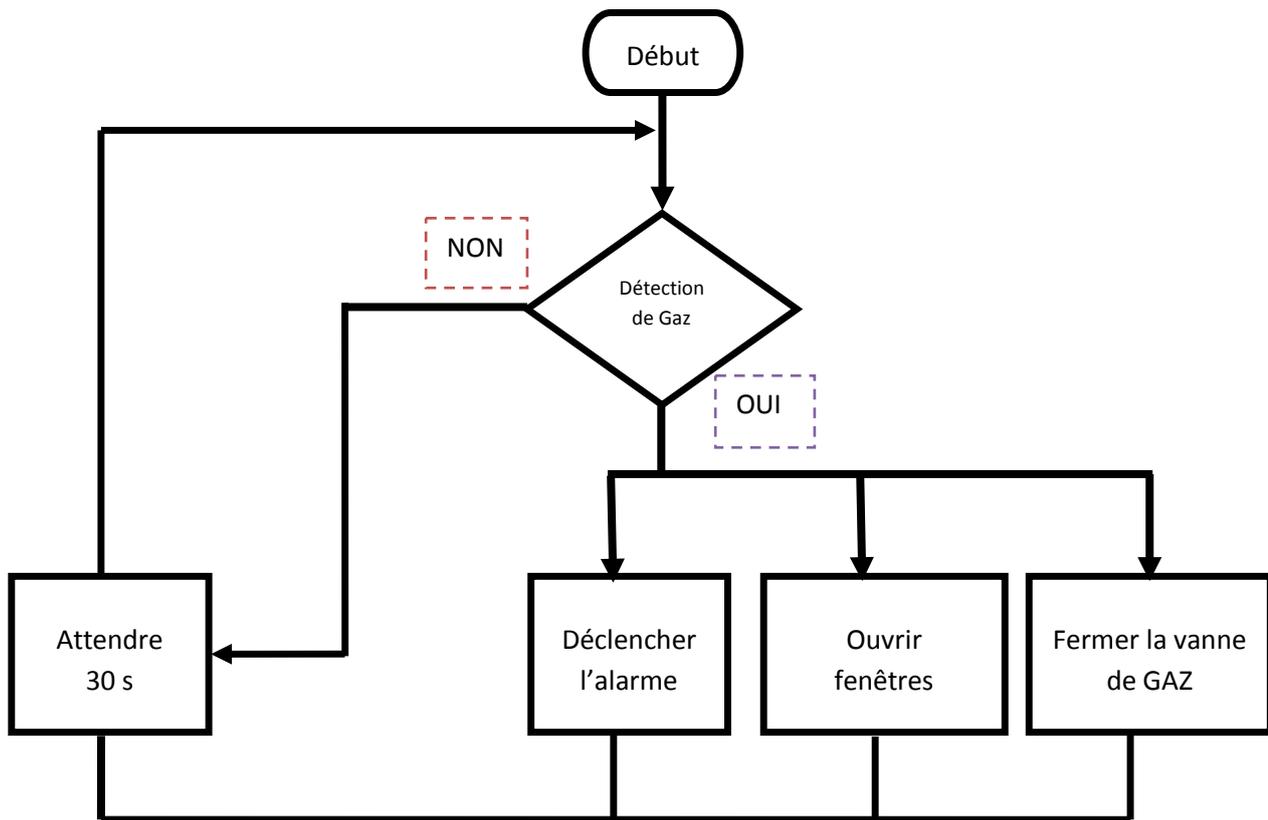


Figure 17: Logigramme de fonctionnement du détecteur de gaz nocif

Le logigramme résume le principe de fonctionnement de notre programme, dès la détection de gaz nocif trois tâches s'exécuteront, la fermeture de la vanne de gaz principale, l'ouverture des fenêtres et le déclenchement d'une alarme pour prévenir les occupants et les riverains.

## I.2. Température

Le mécanisme de régulation de la température dans notre maison est divisé en deux parties. La première en l'absence de personne dans la maison où on cherche à garder une température minimale pour ne pas dégrader ainsi qu'une économie d'énergie, dans la deuxième partie une commande manuelle de la température par l'occupant de la maison, où il peut choisir une plage de température comprise entre 14° et 26° en fonction de ses besoins et de son confort en fonction des saisons, avec une ouverture automatique des fenêtres, en comparant la température extérieure et intérieure, sinon le chauffage sera allumé automatiquement dès le

dépassement de la température minimale dans la maison ,ainsi que le déclenchement de la climatisation en cas de température au-dessus de la température maximale autorisée tout en favorisant les ouvertures pour une meilleure aération de la maison et une économie énergétique, les variables 'A' et 'B' sont les températures maximales et minimales réglables par l'utilisateur grâce à un petit bouton.

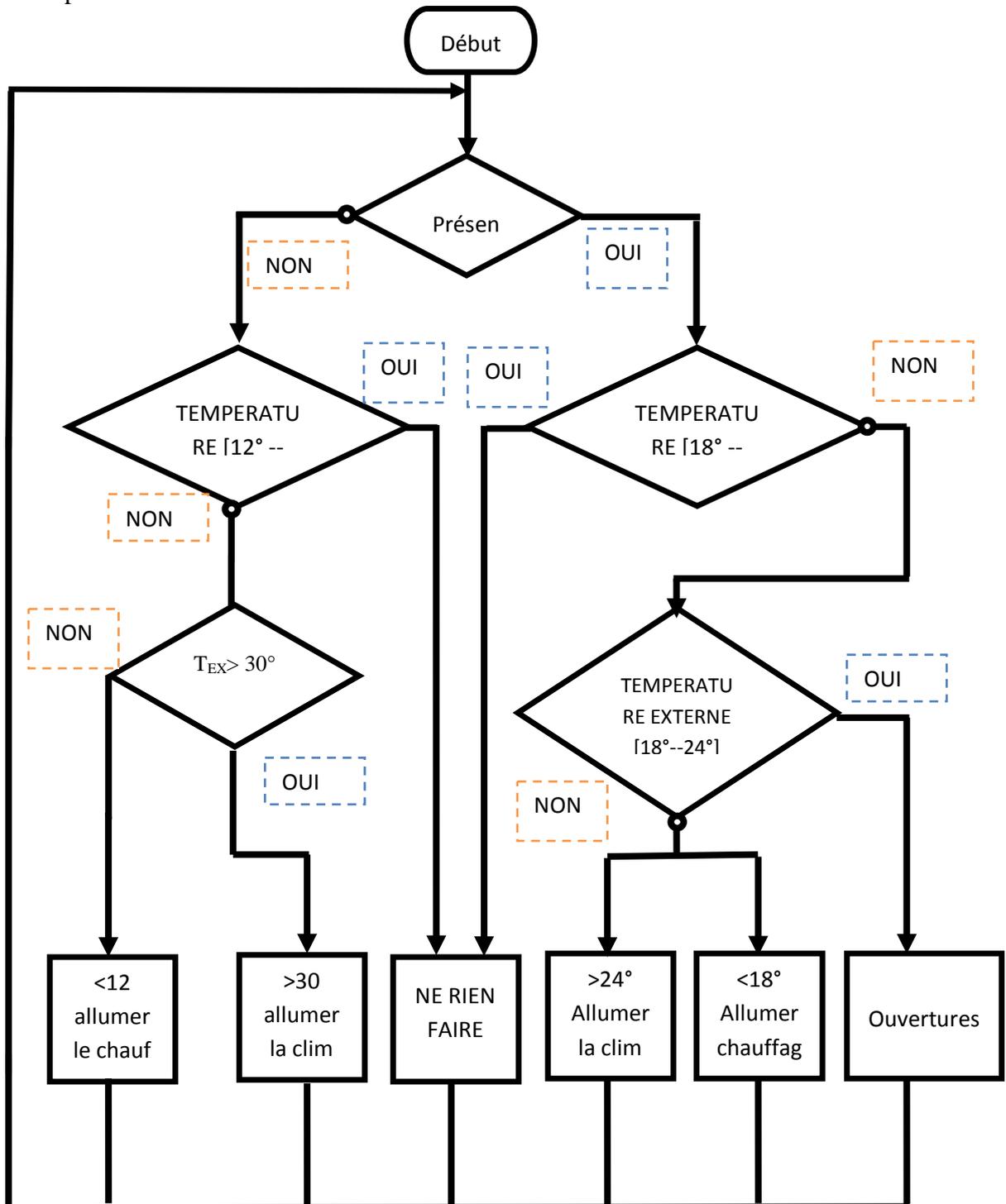


Figure 18: Logigramme de fonctionnement de la température.

### I.3. Humidité

La partie humidité de notre système, est une partie automatique qui protège les occupants d'un milieu trop humide ou bien trop sec, elle est constituée entre autre la carte et des relais, de trois parties qui sont : le capteur d'humidité d'un humidificateur (qui humidifie en cas d'assèchement) et d'un climatiseur qui absorbe l'humidité.

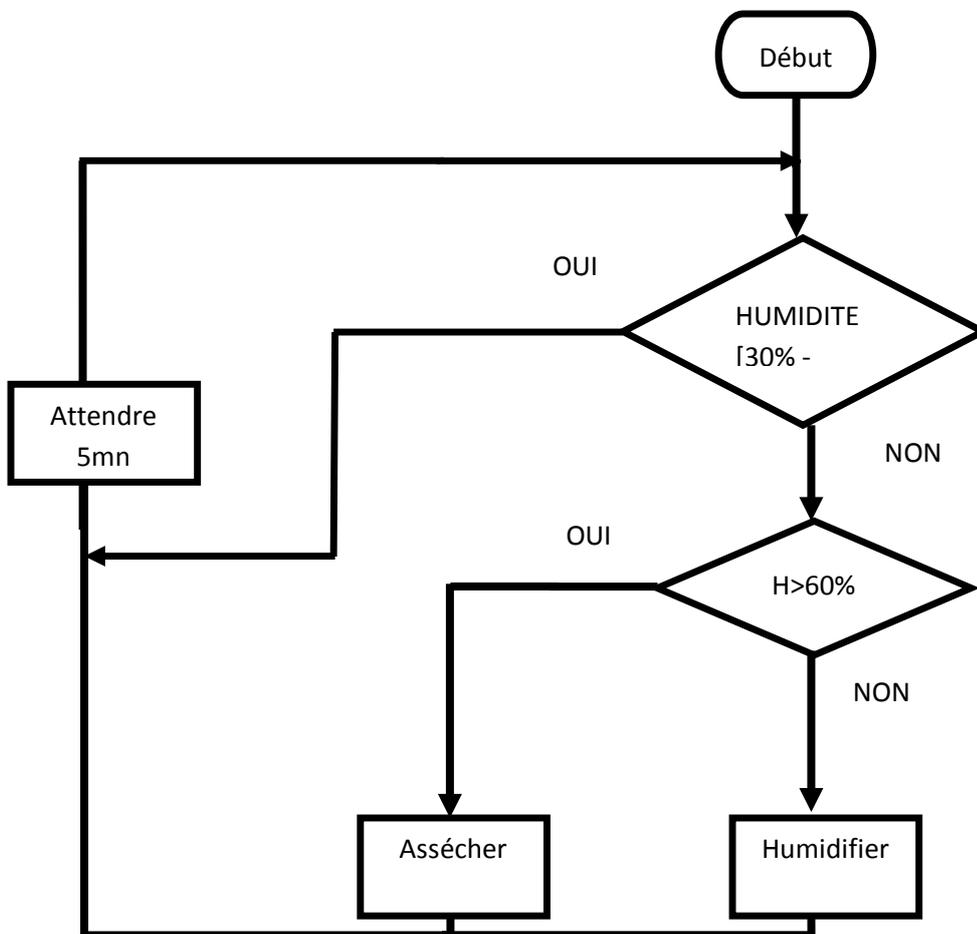


Figure 19: logigramme d'humidité.

### I.4. Détection de présence

La détection de présence est assurée par un capteur PIR basé sur la technologie infrarouge, relié à la carte Arduino, dès la détection de présence il envoie un état logique vers la pine configuré comme entrée.

I.5. Lumière

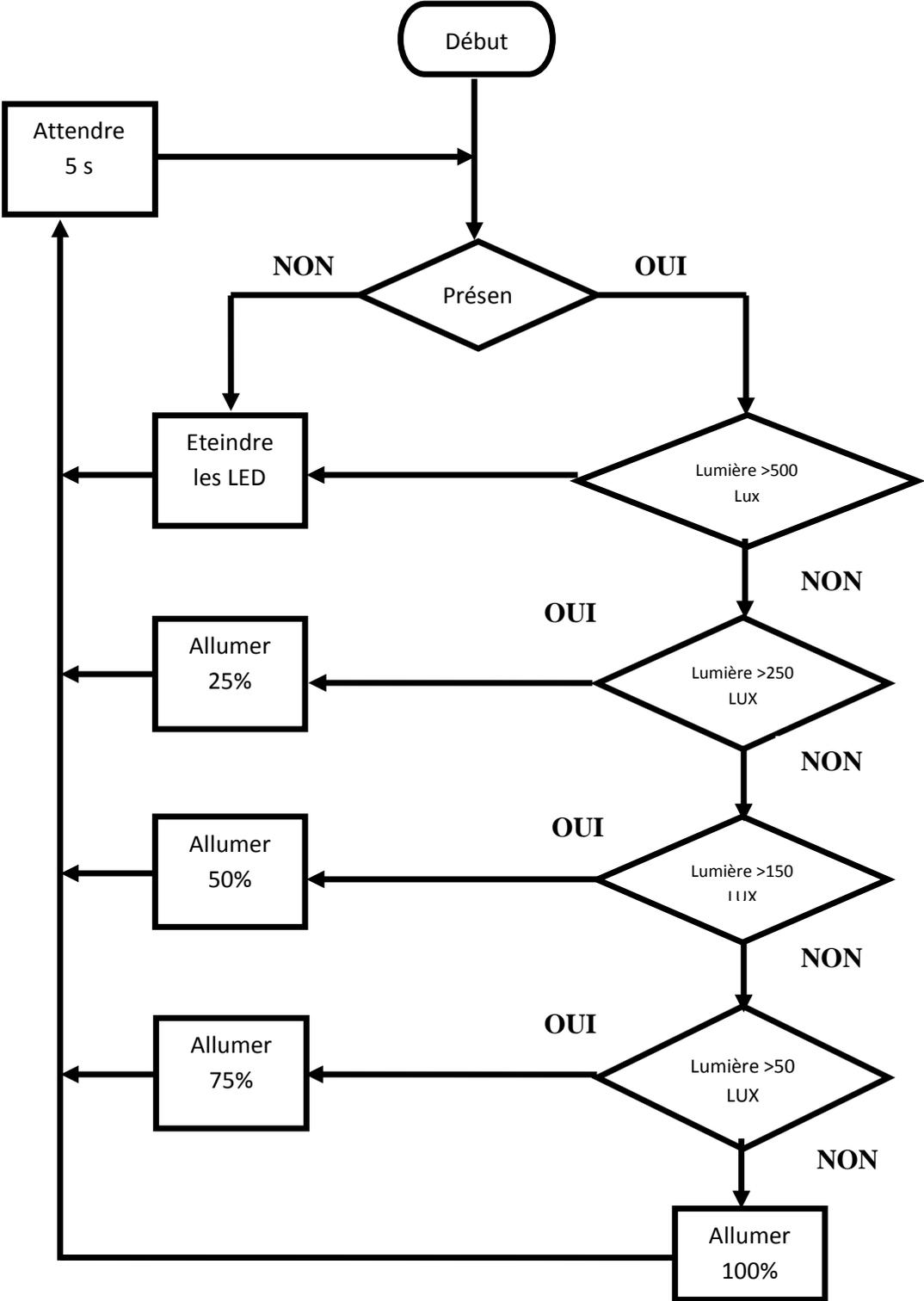


Figure 20: Logigramme de fonctionnement de la lumière.

Comme illustré dans l'organigramme précédent, le principe mis en œuvre pour la gestion d'éclairage repose sur le fait que, pour assurer le confort lumineux nécessaire dans une pièce donnée de la maison, les LED servant à éclairer cette pièce ne s'allument qu'une fois la pièce est occupée et la lumière journalière entrante soit insuffisante pour un bon éclairage, et nous avons eu à utiliser des lampes LED à différents niveaux d'éclairage mais toutes ajustable à quatre états allant du 0 à 100% de la luminescence totale de la lampe.

Et pour réaliser ceci nous avons utilisé deux détecteurs de lumière pour mesurer le niveau d'éclairage à l'extérieur (ouvrir les volets si y a lumière journalière ou fermer s'il n'y a pas) ainsi qu'un autre à l'intérieur pour mesurer la luminescence et régler l'ambiance.

## **II. Programme ARDUINO**

Pour commencer la programmation avec le logiciel ARDUINO préalablement téléchargé et installé, on ouvre un nouveau fichier vide puis on commence la programmation.

Nous commençons par inclure la bibliothèque requise pour nos capteurs, puis on déclare les constantes, les variables et surtout définir les pins de la carte ARDUINO à utiliser avant de passer à la deuxième partie du programme « void setup ». Dans cette partie nous configurons les entrées et les sorties de la carte, les capteurs en entrées, le chauffage, la climatisation ainsi que l'humidification en sortie on passe ensuite à la troisième étape du programme ARDUINO, qui est le programme à exécuter en boucle.

Une fois la programmation terminée, on vérifie le programme avec la commande 'vérifier', et avant de passer à l'étape de téléversement du programme vers la carte on choisit le port vers lequel nous allons la brancher, et nous choisissons le type de carte en notre possession (ARDUINO méga), puis on téléverse le programme grâce à la commande « téléverser ».

## **III. Simulation avec PROTEUS**

Le logiciel PROTEUS est composé de deux modules essentiels :

- L'éditeur de schéma ISIS
- L'outil de conception de circuit imprimé ARES

Dans notre cas on va s'intéresser qu'au module d'édition de schéma ISIS qui va nous permettre de réaliser notre simulation.

### III.1. Présentation de l'interface d'ISIS Proteus 8 Professional

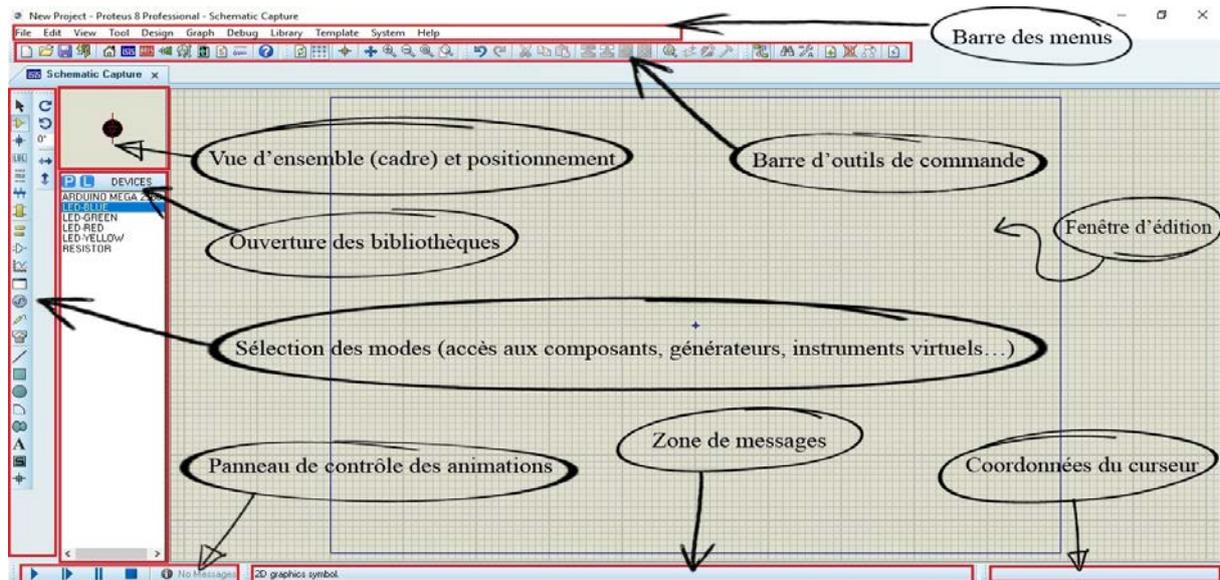


Figure 21: Interface utilisateur ISIS

#### Barre de menus

Elle permet de gérer les travaux (ouverture, sauvegarde...) sur nos fichiers.

#### Barre des outils de commande

Elle reprend ce qui est accessible par les menus (Commandes sur les fichiers, Commandes d'affichage, Commandes Edition, Commandes Outils...)

#### Fenêtre de vue d'ensemble

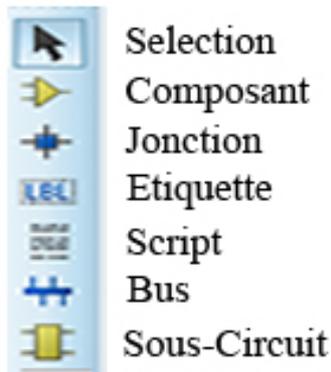
Le cadre en bleu délimite l'espace de travail tel qu'il a été défini par la commande « *set sheet sizes* » du menu « *system* » dans la barre d'outils. Le cadre en vert délimite la zone de travail, c'est à dire la partie du schéma visible dans la fenêtre principale.

On peut déplacer cette zone de travail en pointant la souris sur la zone désirée de la fenêtre de vue d'ensemble et en effectuant un clic gauche.

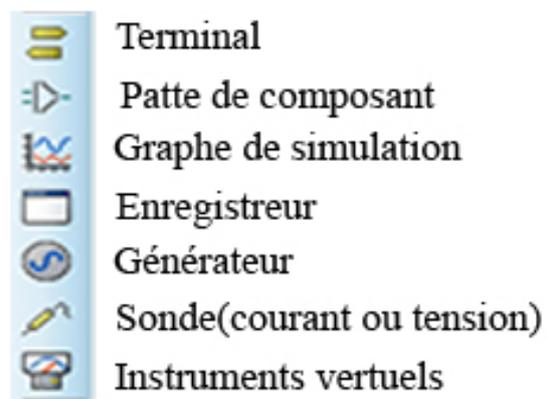
#### Barre d'outils de sélection des Modes

- Mode Principal: permet la sélection des objets, des composants, des nœuds.
- Mode Gadgets: il contient toutes sortes de générateurs, terminal et d'instruments virtuels.
- Mode Graphique: pour écrire des textes et dessiner des traits, des arcs et différentes formes.

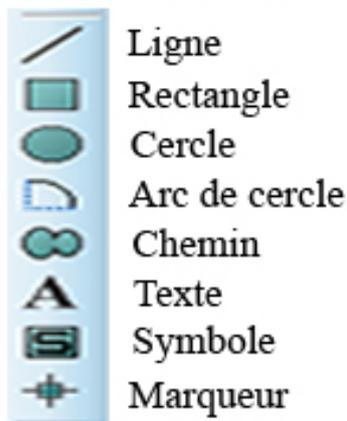
## Mode Principal



## Mode Gadgets



## Mode Graphique



## Orientation d'objets

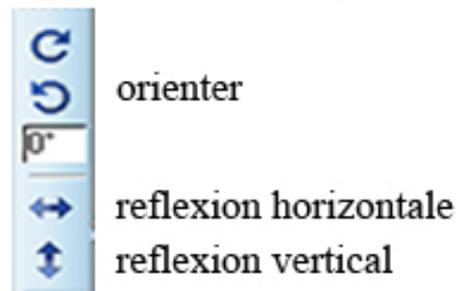


Figure 22: Organisation de Barre d'outils de sélection des Modes ISIS

### Zone d'édition

C'est la zone rectangulaire délimité en bleu, où nous déposons les composants pour dessiner le schéma du modèle a simulé, pour sélectionner un composant nous allons dans la barre d'édition sur « *Component Mode* » puis on sélectionne la lettre « *P* » dans la barre d'outil de sélection, la recherche des composants à utiliser se fait facilement, dans notre cas on aura besoin :

- D'une carte Arduino mega2560.
- Capteur de température et d'humidité DHT11.
- Capteur de présence PIR.
- Photorésistance.
- Capteur de gaz MQ2.
- Moniteur série.
- Résistances.
- LED.
- Potentiomètre.

**Coordonnées du curseur**

Les coordonnées déterminent la position du curseur par rapport à l'origine qui par défaut se trouve au centre de la fenêtre d'édition.

**Panneau de contrôle des animations**

Permet de lancer mettre en pause ou arrêter la simulation.

**Zone de messages**

En positionnant le curseur sur un composant, il affiche le nom et divers informations sur le composant, c'est aussi la zone où est affichée la durée de la simulation.

**III.2. Simulation**

Nous avons d'abord commencé par ramener chaque composant de la bibliothèque, vers la fenêtre d'édition, ensuite nous avons branché les pins, vers les différents composants, comme c'est programmé avec Arduino, en respectant la numérotation et les broches analogiques et numériques de la carte Arduino, et nous avons divisé l'espace de travail en plusieurs parties :

**Lumière**

Dans la partie lumière nous avons utilisé deux photorésistances, l'une pour la lumière extérieure et l'autre pour la lumière intérieure. Et nous varions la luminosité venant de la torche qui se projette sur le devant de la photorésistance, ensuite la carte Arduino reçoit les données sur une broche analogique, et en fonction de cette dernière quatre LED sont commandées, en fonction de la valeur de la photorésistance reçue, l'allumage des lampes est permis que s'il y a une présence dans la pièce.

**Température**

C'est l'une des parties les plus importantes de notre simulation, parce que c'est la partie qui consomme le plus d'énergie si elle est mal gérée.

-Présence égale à « 1 » :

Notre programme va vérifier si la température intérieure est dans l'intervalle  $[18^{\circ}-24^{\circ}]$  si c'est le cas il ne va rien faire. Il va attendre 2s avant de vérifier à nouveau.

Si la température intérieure n'est pas dans l'intervalle  $[18-24]$ , notre programme va d'abord vérifier si la température externe est dans cet intervalle, si c'est le cas il va ouvrir les ouvertures afin de mettre la maison à la même température que l'extérieur.

Sinon deux cas se présentent, un cas où la température est inférieure à  $18^{\circ}$  là on allume le chauffage, et l'autre supérieure à  $24^{\circ}$  où on allume la climatisation.

Présence égale à « 0 » :

Le programme va fermer les ouvertures pour des raisons de sécurité, si la température de la maison est dans l'intervalle  $[12^{\circ}-30^{\circ}]$  la climatisation et le chauffage sont éteint pour économiser de l'énergie, Sinon deux cas se présentent à lui, un cas où la température est inférieure à  $12^{\circ}$  là on allume le chauffage, et l'autre supérieur à  $30^{\circ}$  où on allume la climatisation.

### **Humidité**

Pour cette partie on a fixé un intervalle de confort qui va de 30% à 60%, dans cette plage notre système ne fait rien, mais une fois qu'il y a un dépassement de cet intervalle deux cas s'offre à lui.

Le premier c'est l'humidité supérieure à 60%, là on déclenche le mécanisme d'assèchement.

Le deuxième cas c'est l'humidité inférieure 30%, là on déclenche l'humidificateur.

### **Gaz**

Ce capteur fait partie de la sécurité de la maison, et trois commandes sont directement liées à ce dernier, une électrovanne, une ouverture de fenêtre ainsi qu'une alarme sont déclenchés pour aérer l'espace et prévenir les occupants.

### **Présence**

Le détecteur de présence est le composant dont toutes les parties dépendent, toutes les parties ont deux fonctionnements en absence et en présence d'occupant, en cas d'absence la consommation d'énergie est minimale

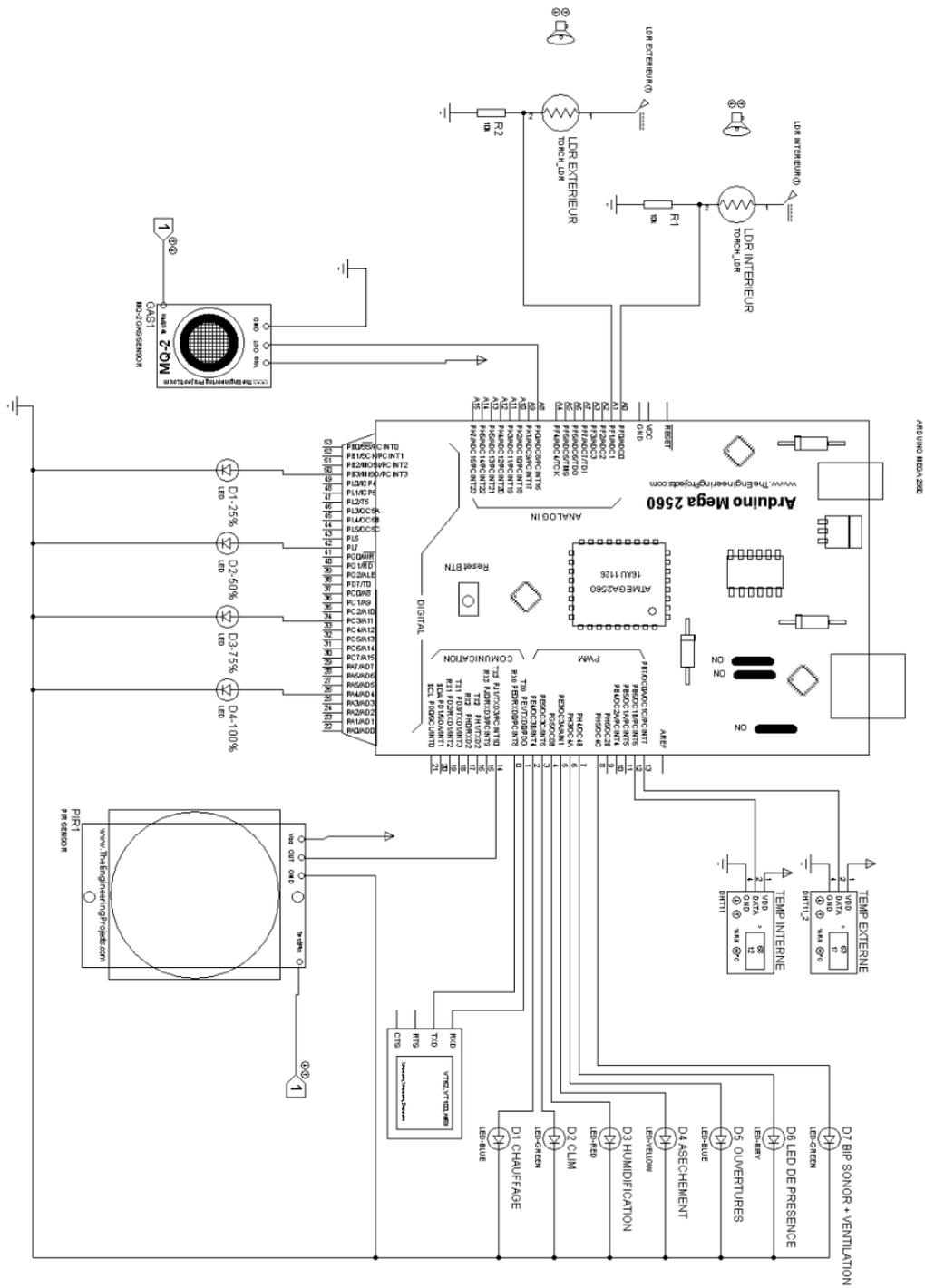


Figure 23: Montage sur Isis Proteus

#### IV. Réalisation et conception

A défaut de réaliser notre projet et l'installer dans une maison nous l'avons réalisé sur une maquette représentative. Pour commencer la partie réalisation, on a décidé de tester chaque montage à part avant de tout regrouper dans une seule maquette, à la fin avec une carte Arduino Méga 2560.

Pour les tests, on a utilisé une carte ARDUINO nano, très pratique pour les petits circuits, le tout premier test est celui de la lumière, nous avons branché deux photorésistances à la carte Arduino une doit être placée à l'intérieur et l'autre à l'extérieur, en fonction des données récoltées on agit sur les ouvertures des volets et les lampes LED, l'ouverture des volets est autorisée uniquement en cas de présence d'occupant, dans la journée on privilégie la lumière naturel, dans le cas où la lumière est insuffisante on allume un nombre bien déterminé de LED.

Le second test concerne la température et l'humidité, avec deux capteurs de température et d'humidité DHT11 très pratique, ils sont peu coûteux et offrent une précision très acceptable. Le premier placé à l'intérieur le second à l'extérieur, la carte reçoit des données des 2 capteurs et en fonction des comparaisons de température reçue la carte donne des instructions aux différents mécanismes remplacés par des LED de différentes couleurs sur notre maquette.

D'abord le système d'humidité est constitué d'un appareil qui humidifie l'espace de vie en cas d'un assèchement, ou assèche l'espace en cas d'une humidité qui dépasse les normes définies dans les paramètres de la carte. Pour la température, elle est en relation avec les volets et les différentes ouvertures qui s'ouvre dès que la température externe est plus adéquate que la température interne, ou le déclenchement d'un relais qui actionne soit un climatiseur pour faire baisser la température ou le chauffage pour la faire augmenter, les deux vers la plage définie toujours dans les paramètres par les occupants de notre maison, où le déclenchement du mode automatique en cas d'absence d'occupant pour préserver la maison des températures trop élevée ou trop basse qui risque d'abîmer la maison, dans nos tests on s'est assuré du fonctionnement des différents cas de figure en rapprochant nos capteurs des milieux chaud ou froid pour constater le bon fonctionnement de notre système.

Le troisième axe de notre réalisation est le détecteur de gaz, dans notre réalisation on a remplacé le capteur de gaz MQ2 par un switch, pour simuler une détection de gaz on met

notre switch a l'état « 1 » un buzzer et un système de ventilation qui sont remplacés par une LED se déclenche, dans le cas où il y a pas de gaz détectés le switch est à l'état « 0 » donc le système est éteint.

Le dernier paramètre est celui de la détection de présence par infrarouge lorsqu'elle est à « 1 », la carte étudie différents paramètres qui sont la température la lumière et le gaz qui déclenche une série d'actions, quand il y a pas de présence détectée « 0 » le système de lumière est éteint de climatisation est éteinte dans la plage [12-30] pour économiser de l'énergie.

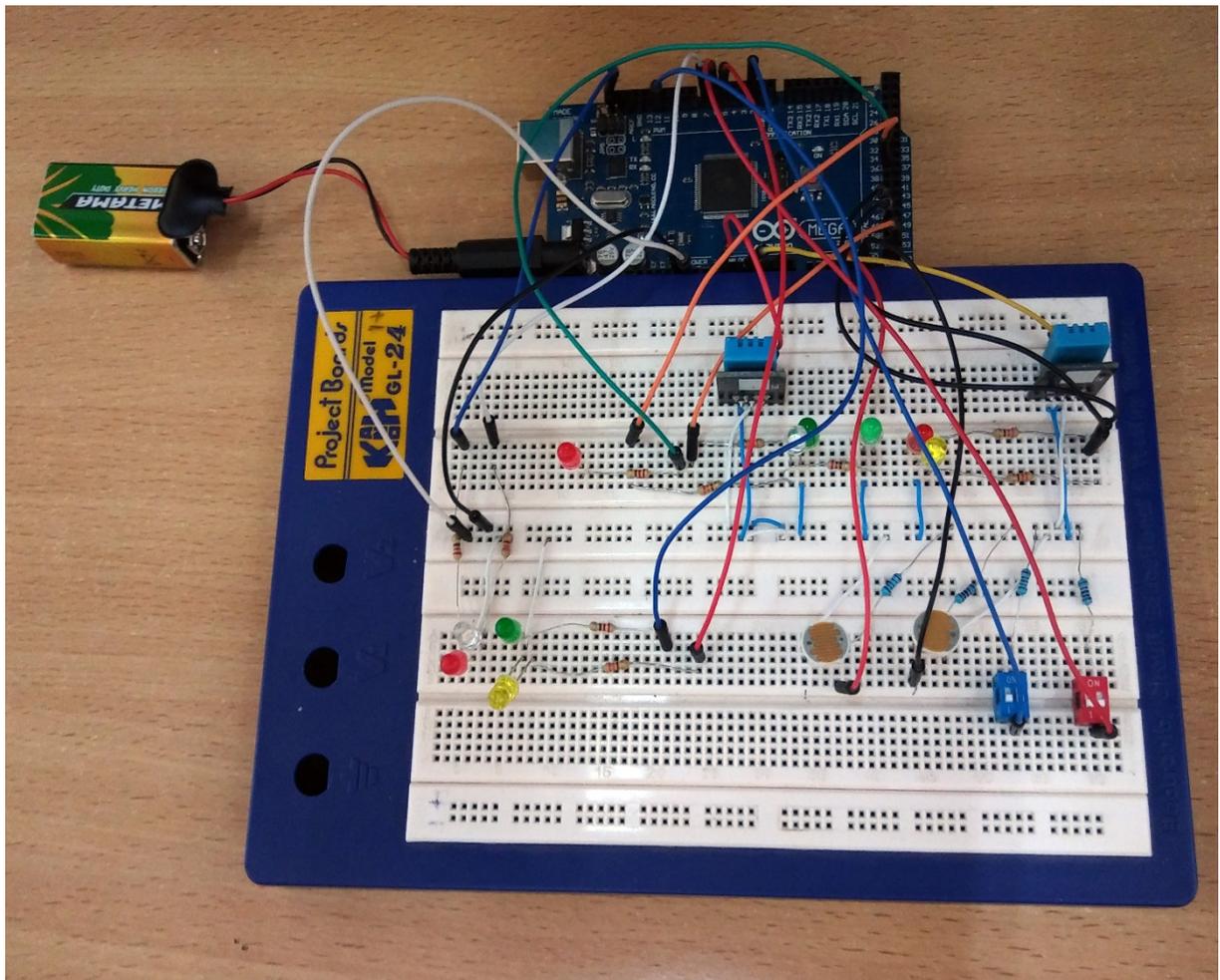


Figure 24: Image de notre réalisation

**Conclusion**

Dans ce chapitre nous avons fait une étude théorique qui nous a mené à faire une application pratique en passant par les deux logiciels Arduino et Proteus, nous avons programmé la carte Arduino ensuite simulé notre programme dans le logiciel Proteus, en se basant sur des organigrammes élaborés qui nous ont facilité la tâche dans la partie réalisation.