

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA
RECHERCHE SCIENTIFIQUE
UNIVERSITE M'HAMED BOUGARA DE BOUMERDES



Faculté de Technologie
Département Ingénierie des Systèmes Electriques
Mémoire de Master

Présenté par

M.BENNADJI Abdelhadi & M. ALLOUANE Mohamed Khalil

Filière : Electronique

Spécialité : Electronique des Systèmes Embarqués

**Gestion d'une serre via une application
ANDROID**

Soutenu le 23 / 06 / 2022 devant le jury composé de :

Mr. MESSAOUDI	Noureddine	MCA	UMBB	Président
Mme. HAROUN	RADIA	MAB	UMBB	Examineur
Mme. DICHOU	Karima	MCB	UMBB	Reporteur

Année Universitaire : 2021/2022

Remerciement

Tout d'abord, nous remercions Allah, le tout puissant qui nous a donné la volonté, la force et la patience pour élaborer notre travail.

Nos vifs remerciements à notre encadreur Mme K. DICHOU pour son aide précieuse, ses conseils constructifs et ses orientations bénéfiques et objectives pour la réalisation de notre projet de fin d'études.

Nos remerciements vont également au membre du jury Mme R. HAROUN qui nous honore de leur présence.

Nos remerciements vont aussi à notre Chef de département Mr. N. MESSAOUDI, ainsi que tous les enseignants du département Ingénierie des Systèmes Electriques.

Dédicace

Je dédie ce modeste travail A ma chère mère, mon grand regret est que vous soyez parti très tôt avant d'avoir bénéficié des fruits de vos efforts investis en moi.

A mon cher père, mes estimes pour vous sont immenses, je vous remercie pour tout ce que vous avez fait pour moi. Que dieu vous préserve une longue vie heureuse.

A ma chère femme et mes adorables enfants.

A ma belle-famille, A qui je souhaite une vie pleine de bonheur et de prospérité.

A mes frères et sœurs et leurs petites familles.

A Toute ma famille, Tous ceux que j'aime et ceux qui m'aiment.

A tous ceux qui, un jour, ont pensé à moi, les plus beaux mots ne sauraient exprimer ma redevance.

A Mon collègue de travail Karim ZEROUALI à qui je souhaite beaucoup de réussite et prospérité.

ABDELHADI

Dédicace

*Je dédie ce modeste travail à mon très cher père, ma
très chère mère qui m'ont soutenu dans mes études
que dieu les protège et les garde pour nous*

*A ma très chère femme qui m'a beaucoup soutenu
pendant mon cycle d'étude*

Mon très cher frère

Mes très chères sœurs

A mes très chers neveux

A toute la famille ALOUANE

A tous mes amis

A toutes la promotion Master 2 ESE (UMBB) 2022

A toutes la promotion MCIL (UBBA)

A toutes mes collègues de travail

A toutes les promotions lesquelles j'ai rencontré

A tous ceux qui m'aiment et tous ceux que j'aime.

MOHAMED KHALIL

Résumé :

La serriculture se développe depuis plusieurs années dans tous les pays soucieux d'augmenter la qualité et la quantité des produits agricoles destinés à la consommation alimentaire des populations et de leurs milieux de vie. Les systèmes de production en serre sont devenus très avancés grâce à la mécanisation des outils.

Dans les cas où les serres situées dans des endroits difficiles d'accès ou s'il y a plusieurs serres, il est nécessaire d'automatiser les serres et avoir un système de connexion et commande à distance.

Nous avons réalisé un système afin de gérer une serre automatique via application Androïde.

Les résultats ont été concluants et peuvent être exploités pour des réalisations pratiques.

Mots clés : Serre, Humidification, Microcontrôleur, Capteur de Température, Capteur de lumière, Afficheur LCD, Wokwi.

Abstract:

Greenhouse cultivation has been developing for several years in all countries concerned with increasing the quality and quantity of agricultural products intended for the food consumption of populations and their living environments. Greenhouse production systems have become very advanced due to the mechanization of tools.

In cases where the greenhouses are located in places that are difficult to access or if there are several greenhouses, it is necessary to automate the greenhouses and have a connection and remote-control system.

We have created a system to manage an automatic greenhouse via Android application.

The results were conclusive and can be exploited for practical achievements.

Keywords: Greenhouse, Humidification, Microcontroller, Temperature Sensor, Light Sensor, LCD Display, Wokwi.

ملخص :

تطورت الزراعة في الصوبات لعدة سنوات في جميع البلدان المعنية بزيادة جودة وكمية المنتجات الزراعية المخصصة للاستهلاك الغذائي للسكان وبيئاتهم المعيشية. أصبحت أنظمة إنتاج البيوت المحمية متقدمة جدًا بسبب إمكانية الأدوات.

في الحالات التي توجد فيها الدفيئات الزراعية في أماكن يصعب الوصول إليها أو في حالة وجود العديد من البيوت الزجاجية ، من الضروري أتمتة البيوت البلاستيكية ولديها نظام اتصال وتحكم عن بعد.

لقد أنشأنا نظامًا لإدارة دقيقة تلقائية عبر تطبيق Android.

كانت النتائج حاسمة ويمكن استغلالها لتحقيق إنجازات عملية.

الكلمات الرئيسية: البيت الزجاجي ، الترطيب ، متحكم دقيق ، مستشعر درجة الحرارة ، مستشعر الضوء ، شاشة LCD ، Wokwi.

Sommaire

INTRODUCTION GENERALE.....	1
CHAPITRE 01: GENERALITES SUR LES SERRES	
1.1. INTRODUCTION.....	2
1.2. DEFINITION	2
1.3. IMPORTANCE DE LA SERRE	2
1.4. LES DIFFERENTS TYPES DES SERRES.....	2
1.4.1 Serres Tunnel	2
1.4.2 Serres multi-chapelle.....	3
1.4.3 Serre de Jardin	4
1.5. PRINCIPAUX PARAMETRES CLIMATIQUES	5
1.5.1 La Température	5
1.5.1.1 Chauffage.....	5
1.5.1.2 Ventilation	7
1.5.1.3 Système d’ouverture de serre	7
1.5.2 Humidité.....	8
1.5.2.1 Humidification	8
1.5.2.2 Déshumidification	9
1.5.3 Eclairage.....	11
1.5.3.1 Eclairage naturel.....	11
1.5.3.2 Eclairage artificiel	11
1.5.4 L’irrigation.....	11
1.5.5 Ombrage.....	12
1.6. PROBLEMATIQUE.....	13
1.7. SOLUTION PROPOSEE	13
1.8. CONCLUSION	13
CHAPITRE 02 : ENVIRONNEMENT ET OUTILS DE TRAVAIL	
2.1. INTRODUCTION	14
2.2. LA CARTE ARDUINO	14
2.2.1 Présentation de l’Arduino UNO.....	14

2.2.2 Fiche Technique de l'Arduino UNO.....	15
2.2.3 L'Alimentation	15
2.2.4 Communication	16
2.2.5 Les Broches	16
2.2.6 La Mémoire.....	16
2.3. LES CAPTEURS	16
2.3.1 Capteur de Température et d'Humidité DHT22	16
2.3.2 Capteur d'Humidité du Sol	17
2.3.3 Capteur de Lumière LDR.....	17
2.3.4 Capteur d'Air MQ 135.....	18
2.4. AFFICHEUR LCD 2x16 CARACTERES	19
2.4.1 Caractéristiques de l'Afficheur LCD	20
2.4.2 Caractéristiques Mécaniques de l'Afficheur	20
2.4.3 Broches de l'Afficheur	20
2.5. LE MODULE WIFI ESP8266	22
2.5.1 Définition	22
2.5.2 Les commandes AT	23
2.6. LES RELAIS ELECTROMECHANQUES	23
2.7. LES RELAIS ELECTROMECHANQUES	23
2.8. LOGICIEL ARDUINO IDE.....	26
2.9. PLATEFORMES DE SIMULATION	28
2.9.1 Tinkercad	28
2.9.2 Wokwi.....	32
2.10. CONCLUSION	33
CHAPITRE 03 : PREPARATION DE LA REALISATION	
3.1. INTRODUCTION	34
3.2. ESSAIS DU CAPTEUR DHT22.....	34
3.2.1 Code source du capteur DHT22	34
3.2.2 Simulation du capteur DHT22.....	35
3.2.3 Câblage du capteur DHT22 avec la carte Arduino.....	35
3.2.4 Résultats obtenues lors de l'essai du capteur DHT22	36
3.3. ESSAIS DU CAPTEUR DE GAZ MQ 135	36
3.3.1 Code source du capteur de GAZ MQ 135	36
3.3.2 Simulation du capteur de GAZ MQ 135	37
3.3.3 Câblage du capteur de GAZ MQ 135 avec la carte Arduino.....	37
3.3.4 Résultats obtenues lors de l'essai du capteur de GAZ MQ 135.....	38
3.4. ESSAIS DU CAPTEUR D'HUMIDITE DE SOL.....	38

3.4.1 Code source du capteur d'Humidité de Sol.....	38
3.4.2 Simulation du capteur d'Humidité de Sol.....	39
3.4.3 Câblage du capteur d'Humidité de Sol avec la carte Arduino	39
3.4.4 Résultats obtenues lors de l'essai du capteur d'Humidité de Sol	40
3.5. ESSAIS DU CAPTEUR DE LUMIERE LDR	40
3.5.1 Code source du capteur de lumière LDR	40
3.5.2 Simulation du capteur de lumière LDR	41
3.5.3 Câblage du capteur de lumière LDR avec la carte Arduino	41
3.5.4 Résultats obtenues lors de l'essai du capteur de lumière LDR.....	42
3.6. ESSAIS DE L’AFFICHEUR LCD	42
3.6.1 Code source de l’afficheur LCD	42
3.6.2 Simulation de l’afficheur LCD	43
3.6.3 Câblage de l’afficheur LCD avec la carte Arduino.....	43
3.6.4 Résultats obtenues lors de l’essais de l’afficheur LCD	44
3.7. ESSAIS DU MODULE WIFI ESP 8266	44
3.7.1 Code source du module WIFI ESP 8266	44
3.7.2 Câblage du module WIFI ESP 8266 et les deux LED avec la carte Arduino.	45
3.7.3 Résultats obtenues lors des l’essais du module WIFI ESP 8266	46
3.8. CONCLUSION	46
CHAPITRE 04 : LA REALISATION	
4.1. INTRODUCTION	47
4.2. ALGORITHME	47
4.3. LE CODE SOURCE DU PROJET	48
4.4. PHOTOS PRISE LORS DES ESSAIS DU PROJET	51
4.5. CONCLUSION	55
CONCLUSION GENERALE	56

Liste des figures

CHAPITRE 01 : GENERALITES SUR LES SERRES

Figure 1-1 : Une serre tunnel.....	3
Figure 1-2 : Une serre multi-chapelle.....	3
Figure 1-3 : Une serre jardin.....	4
Figure 1-4 : Générateur d'air chaud.....	6
Figure 1-5 : Exemple d'émission d'eau chaude.....	6
Figure 1-6 : Ventilateur de serre.....	7
Figure 1-7 : Ouverture et fermeture d'une fenêtre de serre automatique	8
Figure 1-8 : Opération de brumisation dans une serre.....	9
Figure 1-9 : Exemple de Plastique Anti-goutte.....	10
Figure 1-10 : Exemple d'ombrage d'une serre.....	12

CHAPITRE 02 : ENVIRONNEMENT ET OUTILS DE TRAVAIL

Figure 2-1 : Composants techniques de l'Arduino.....	14
Figure 2-2 : Capteur DHT22.....	16
Figure 2-3 : Capteur d'Humidité de Sol.....	17
Figure 2-4 : Capteur de Lumière LDR.....	18
Figure 2-5 : Capteur d'Air MQ-135.....	18
Figure 2-6 : Afficher LCD 2x16 Caractères.....	19
Figure 2-7 : Les Broches de l'Afficheur LCD 2x16.....	21
Figure 2-8 : Le Module ESP8266.....	22
Figure 2-9 : Pins du module ESP8266.....	22
Figure 2-10 : Relais électromécanique.....	23
Figure 2-11 : L'éditeur de l'application Remote XY.....	24
Figure 2-12 : L'onglet « configuration ».....	24
Figure 2-13 : Paramètres de l'interface du module.....	25
Figure 2-14 : Configuration réseau de l'application Remote XY sur Smartphone.....	26
Figure 2-15 : Les Zones de l'Interface Graphique.....	27
Figure 2-16 : Rejoindre Tinkercad.....	29
Figure 2-17 : Création du circuit.....	29
Figure 2-18 : Choix des composants.....	30
Figure 2-19 : Conception et Simulation du Circuit par Tinkercad.....	31

Figure 2-20 : Rejoindre Wokwi.....	32
Figure 2-21 : Conception et Simulation du Circuit par Wokwi.....	32

CHAPITRE 03 : PREPARATION DE LA REALISATION

Figure 3-1 : Simulation du capteur DHT22	35
Figure 3-2 : Câblage du capteur DHT22 avec la carte Arduino.....	35
Figure 3-3 : Résultats obtenues lors de l'essai du capteur DHT22.....	36
Figure 3-4 : Simulation du capteur de GAZ MQ 135.....	37
Figure 3-5 : Câblage du capteur de GAZ MQ 135 avec la carte Arduino	37
Figure 3-6 : Résultats obtenues lors de l'essai du capteur de GAZ MQ 135	38
Figure 3-7 : Simulation du capteur d'Humidité de Sol	39
Figure 3-8 : Câblage du capteur d'Humidité de Sol avec la carte Arduino	39
Figure 3-9 : Résultats obtenues lors de l'essai du capteur d'Humidité de Sol	40
Figure 3-10 : Simulation du capteur de lumière LDR	41
Figure 3-11 : Câblage du capteur de lumière LDR avec la carte Arduino	41
Figure 3-12 : Résultats obtenues lors de l'essai du capteur de lumière LDR	42
Figure 3-13 : Simulation de l'afficheur LCD	43
Figure 3-14 : Câblage de l'afficheur LCD avec la carte Arduino	43
Figure 3-15 : Résultats obtenues lors de l'essais de l'afficheur LCD	44
Figure 3-16 : Câblage du module WIFI ESP8266 et les deux LED avec la carte Arduino	45
Figure 3-17 : Résultats obtenues lors de l'essais du module WIFI ESP 8266	46

CHAPITRE 04 : LA REALISATION

Figure 4-1 : Affichage de la Température et Humidité sur l'afficheur LCD	51
Figure 4-2 : Affichage des grandeurs mesurés sur le Smart Phone.....	51
Figure 4-3 : Pompe en arrêt au cas de $vsl < ref_sol$	52
Figure 4-4 : Mise en marche de la Pompe par le Smart phone.....	52
Figure 4-5 : Lampe et Pompe en arrêt au cas de $vallum < ref_lum$	53
Figure 4-6 : Lampe et Pompe en arrêt au cas de $vallum < ref_lum$ et $vsl < ref_sol$	53
Figure 4-7 : Ventilos en Marche au cas de $T_in > ref_Tin$ ou $H_in > ref_H_in$	54
Figure 4-8 : Mise en marche du Ventilos par le Smart phone	54
Figure 4-9 : Mise en marche de la Lampe par le Smart phone	55

Liste des tableaux

CHAPITRE 02 : ENVIRONNEMENT ET OUTILS DE TRAVAIL

Tableau 2-1 : Fiche technique de l'Arduino.....	15
Tableau 2-2 : Les niveaux de pollutions d'air.....	19
Tableau 2-3 : Spécifications Mécaniques de LCD 2x20 Caractères.....	20
Tableau 2-4 : Brochage de LCD 2x16 Caractères.....	21

Liste des abréviations

API : Automate Programmable Industriel.

CGRAM : Character Graphics Random Access Memory.

CGROM : Character Graphics Read Only Memory.

CO₂ : Dioxyde de carbone.

DDRAM : Data Display Random Access Memory.

DHT : Capteur de Température et Humidité Digital.

E : Enable.

EEPROM : Electrically Erasable Programmable Read Only Memory.

GND : Ground (masse).

IDE : Integrated Development Environment.

LCD : Liquid Crystal Display.

LDR : Light Dependent Resistor.

LED : Light Emitting Diode.

NC : Normaly Close.

NO : Normaly Open.

PWM : Pulse Width Modulation.

R/W : Read / Write.

RS : Register Select.

RX : Réception.

SIG : Signal.

SnO₂ : Dioxyde d'étain.

SRAM : Static Random Acces Memory.

TX : Transmission.

UART : Universal Asynchronous Receiver Transmitter.

USB : Universal Serial Bus.

VAC/AC : Volt Alternatif Courant.

VDC/VCC/DC : Courant Continu.

VDD : Drain Supply Voltage.

Vin : tension d'entrée.

Vout : Output Voltage.

VSS : Source Supply Voltage.

WIFI : Wireless Fidelity.

Introduction Générale

INTRODUCTION GENERALE

La culture sous serre connaît depuis plusieurs années, un développement important dans tous les pays concernés par la volonté d'augmenter la qualité et la quantité des produits agricoles, destinés à la consommation alimentaire des populations et à leurs cadres de vie. Les systèmes de production sous serre deviennent considérablement sophistiqués. C'est grâce à la mécanisation des outils.

Il est important de choisir un contrôle des paramètres climatiques qui correspond aux besoins visés. Pour améliorer la rentabilité, on doit faire croître les cultures dans des environnements optimaux.

L'agriculteur trouve des difficultés d'intervenir à ses produits agricoles surtout dans des cas où les serres se trouvent dans des endroits inaccessibles ou le cas de plusieurs serres. pour résoudre ce problème il est indispensable d'automatiser les serres en contrôlant la température, l'humidité de l'air, l'humidité du sol et le taux de CO₂, aussi l'éclairage peut être régulé au moyen d'un éclairage artificiel pour accélérer le développement des plantes dans les cas où les serres se trouve dans des lieux sous l'ombre.

Mais l'automatisation n'est pas suffisante, l'agriculteur veut aussi avoir des informations sur ses serres, là où il faut que le système automatique soit connecté à distance avec un terminal qui résoudre le dite problème.

Dans le cadre de ce travail, nous souhaitons développer une plateforme composée d'une carte de commande ARDUINO à base de microcontrôleur afin d'automatiser une serre et de connecter le système via un smart phone.

CHAPITRE 01

GENERALITES SUR LES SERRES

1.1. INTRODUCTION

La serre est un abri exploitant le rayonnement solaire, destiné à la culture et à la protection des plantes.

L'objectif des serres étant de créer un environnement propice à leur développement en tirant parti de l'influence du climat. En créant un micro climat, la serre permet d'influencer le cycle végétatif des plantes.

Dans ce chapitre, nous présenterons une vue globale sur les différents types de serres agricoles, ainsi que les paramètres climatiques d'environnement dans une serre.

1.2. DEFINITION [1]

Une serre est une structure qui peut être parfaitement fermé, Elle est destinée en général à la production agricole. L'objectif étant de créer un environnement propice à leur développement en tirant parti de l'influence du climat, en créant un micro climat, pour une meilleure gestion des besoins des plantes, pour en accélérer la croissance ou pour les produire indépendamment des saisons.

1.3. IMPORTANCE DE LA SERRE [1]

La serre de culture offre la possibilité de s'affranchir des contraintes climatiques extérieures (pluie, vent, froid). Elle est conçue pour recréer un environnement donné (microclimat). Elle permet le chauffage de l'air et des racines, le contrôle de l'irrigation et de la fertilisation, l'enrichissement en CO₂ et le contrôle de l'humidité. Elle joue un rôle économique en présentant des produits sur le marché en hors saison. La serre permet ainsi d'obtenir une production végétale dans des conditions meilleures que celles existantes naturellement et ce par une meilleure qualité du produit. Pour aboutir à ces résultats, il faut répondre minutieusement aux exigences de la culture pour les différents facteurs intervenant dans sa croissance et son développement.

1.4. LES DIFFERENTS TYPES DES SERRES [2]

Même si les serres agricoles ont le même rôle, mais nous pouvons voir quelques différences concernant leurs formes, leurs porteurs ou les matières de couverture.

1.4.1 Serres tunnel

Une serre tunnel (Figure 1-1) est formée de plusieurs grands arceaux métalliques, recouverts d'un film souple en plastique transparent, qui lui donne la forme d'un tunnel. Les dimensions de la serre dépendent des besoins du jardinier.



Figure 1-1 Une serre tunnel

- **Caractéristique des serres tunnel**
 - La hauteur varie généralement entre 2 et 2,50 mètres.
 - La largeur se situe entre 3 et 5 mètres.
 - Une surface au sol minimale de 6 m².
 - La deuxième porte est nécessaire dès que la serre tunnel dépasse les 6 mètres de long.

1.4.2 Serres multi-chapelle

Les différentes pièces qui composent la structure de la gamme à serres Multi chapelle (Figure 1-2) sont conçues pour être parfaitement assemblées grâce à des joints vissés qui simplifient le montage tout en absorbant idéalement les différentes forces qui ont un impact sur la structure. Elles sont très demandées pour leur robustesse et leur grande capacité d'adaptation aux dimensions et aux caractéristiques du terrain.



Figure 1-2 Une serre multi-chapelle

- **Caractéristique des serres multi-chapelle**
 - Matériaux de revêtement peut être :
 - Film plastique.
 - Matériaux semi-rigides
 - Plaque rigide
 - Les éléments en aluminium et les profils d'ancrage.
 - Un volume intérieur important.
 - L'inclinaison des arceaux permettent de tirer le meilleur parti des rayons solaires.

1.4.3 Serre de jardin

C'est un petit modèle, parfait pour faire pousser des plantes, idéale pour protéger ses plantes en hiver, récolter des fruits et légumes toute l'année ou réaliser des semis sous abri. Une serre peut être construite avec divers matériaux et selon différentes formes, même avec des matériaux de récupération (figure 1-3).



Figure 1-3 Une serre jardin

- **Caractéristique d'une serre de jardin**

- Les matériaux fréquemment utilisés pour l'armature de la serre :
 - Les structures en aluminium.
 - Les structures en acier.
 - Les structures en bois.
- Les matériaux fréquemment utilisés pour les panneaux de la serre :
 - Le verre horticole.
 - Le verre trempé.
 - Le polycarbonate alvéolaire.

1.5. PRINCIPAUX PARAMETRES CLIMATIQUES [3]

La maîtrise du climat est la raison d'être des serres ; on peut créer un environnement idéal pour la croissance des plantes. Sa gestion est souvent confiée à automate programmable industriel (API).

La température, l'humidité, irrigation et la lumière restent jusqu'à maintenant parmi les paramètres les plus influents dans la production des plantes. Pour améliorer leurs rentabilités, il est indispensable d'optimiser le climat environnant aux cultures. Il est donc important de contrôler les paramètres climatiques cités ci-dessus. Dans ce cadre, il est nécessaire d'équiper la serre par un ensemble de capteurs climatiques et d'actionneurs afin d'agir sur les paramètres climatiques.

1.5.1 La Température

La température sous serre est le paramètre le plus important pour la production des cultures sous serre. Il se décompose en trois types :

- Température issue du sol,
- Température de la plante,
- Température ambiante à la culture.

La température a une grande influence sur la croissance végétative. En effet, elle intervient dans beaucoup de phénomènes biologiques tels que la photosynthèse et la respiration. La vitesse de réaction augmente facilement avec la température ambiante. Citons par exemple la photosynthèse qui se multiplie presque en deux lorsque la température augmente de 10 °C.

Sachant que la température monte et descend dans la serre, on doit utiliser le chauffage pour la chauffer et la ventilation pour le refroidissement (système d'ouverture du toit pour une ventilation naturelle).

1.5.1.1 Chauffage

Le chauffage permet d'adapter les apports thermiques aux besoins de la culture et d'éliminer l'humidité de l'air. Il fonctionne également sur la mesure de température. L'intensité du chauffage peut dépendre de l'éclairage avec peu de lumière solaire et vice versa.

Il existe deux types de radiateurs :

- Émissions d'air chaud : Selon le produit d'une unité autonome générant de l'air chaud, ou d'un appareil de chauffage utilisant de l'eau chaude produite à partir d'une chaudière.



Figure 1-4 Générateur d'air chaud

- Les émissions d'eau chaude : sont envoyées dans des tubes métalliques répartis dans la serre. Les deux principaux types d'émissions sont caractérisés par des températures basses et élevées soutenues par la culture (émergence classique, condensation, brûleur, etc.).



Figure 1-5 Exemple d'émission d'eau chaude

1.5.1.2 Ventilation

Le traitement de l'air a une fonction tout aussi importante que la lumière lorsque la culture est dans un espace clos. Un ventilateur permet de renouveler l'air et d'apporter aussi le CO₂ nécessaire à la photosynthèse tout en garantissant une régulation de la température et de l'hygrométrie. Le mouvement horizontal de l'air offre plusieurs avantages. Plus particulièrement, il réduit sensiblement les gradients de la température dans la serre et enlève l'humidité dans le reste de la serre, ce qui favorise une grande homogénéité du climat sous serre. Cette ventilation est considérée comme un refroidisseur mécanique (forcé).



Figure 1-6 Ventilateur de serre

1.5.1.3 Système d'ouverture de serre

Ce système considéré comme ventilation naturelle travaille sur l'échange d'air de l'extérieur par un capteur de température, qui est lu par un circuit électronique, et décide jusqu'à quelle température il doit être maintenue ouvert (grâce à un petit moteur et un mouvement de rotation).

Lorsque le système fonctionne, la serre est complètement indépendante. Lorsque la température augmente, celle-ci s'ouvre pour laisser entrer l'air pour le refroidissement et la ventilation.

Une fois que le capteur de température atteint le maximum (imposé par l'utilisateur), elle conduit à ouvrir les engrenages de moteur à commande de toit pour ventilation des gaz à effet de serre, à une certaine température (réglée par l'utilisateur) supprime le signal pour fermer le toit.



Figure 1-7 Ouverture et fermeture d'une fenêtre de serre automatique

1.5.2 Humidité

Le confinement et l'étanchéité de la serre favorisent l'augmentation de l'humidité absolue tandis que l'élévation de la température de l'air tend à accroître le déficit de la saturation. Les conséquences de l'humidité sont :

- Pendant le jour : L'élévation de la température de l'air peut entraîner un abaissement exagéré de son humidité relative et provoquer un véritable "stress hydrique" au niveau de la végétation d'où la nécessité de prévoir un système de ventilation de la serre.
- Pendant la nuit : Les serres étant généralement fermées, l'humidité relative est élevée. Au cours de la nuit, la température baisse. Il se produit fréquemment des condensations sur les parois et les gouttes condensées peuvent tomber sur la végétation (conditions favorables au développement de certaines maladies, etc.).

1.5.2.1 Humidification

Plusieurs techniques d'humidifications permettent de faire augmenter l'humidité :

- Brumisateur : Fait augmenter l'humidité dans la serre. Il consiste à atomiser l'eau en

finesparticules (d'environ 10 μm). Ensuite un système à haute pression d'air les disperse dans le climat sous serre. Il s'agit d'un système qui produit de la brume sans laisser de gouttelettes sur les plants afin de ne pas favoriser le développement des maladies fongiques. L'augmentation de l'humidité dans la serre provoquée par le système de brumisation, peut entraîner une diminution de la température puisque l'eau nécessite de l'énergie pour qu'il s'évapore.



Figure 1-8 Opération de brumisation dans une serre

- Écran de refroidissement : Bien qu'utilisé pour réduire la température de la serre, l'écran de refroidissement peut également servir pour accroître l'humidité. L'eau s'écoule à travers l'écran et permet de refroidir et humidifier l'air qui y circule de l'extérieur vers l'intérieur de la serre.
- Arrosage des allées : Cette méthode, peu coûteuse, permet d'augmenter l'humidité de l'air mais de manière ponctuelle. Les effets bénéfiques de l'arrosage sur l'humidité se font sentir sur une très courte période.
- Ventilation : La réduction de la ventilation naturelle, par la fermeture partielle ou totale des ouvrants contribue à accroître l'humidité de la serre sous certaines conditions. Ainsi, pour une culture dont la surface foliaire est importante, la fermeture des ouvrants peut entraîner un accroissement de la quantité de vapeur d'eau dans l'air. Cette vapeur d'eau provient de la transpiration des plantes. Cependant, lorsque l'humidité atteint une certaine valeur, la transpiration cesse et l'humidité ne pourra s'accroître davantage. Les deux possibilités qui sont le refroidissement de l'air et l'enrichissement en vapeur d'eau sont généralement combinées.

1.5.2.2 Déshumidification

L'humidité de l'air est généralement exprimée en humidité relative ou déficit hydrique. On constate que la technique de déshumidification traditionnellement employée par les producteurs combine l'aération et le chauffage. Le chauffage permet d'une part de diminuer l'humidité relative de l'air en augmentant la pression de vapeur saturante, d'autre part de faciliter l'évacuation de l'air chaud chargé en humidité grâce à l'aération. Cette technique entraîne une perte d'énergie puisqu'une partie de l'énergie dégagée par le chauffage est évacuée par l'aération. La pratique de la déshumidification représente environ 20% à 30%

des dépenses énergétiques. Afin de maintenir la transpiration active, il est essentiel d'éliminer l'excès d'humidité de la serre. L'évapotranspiration est fonction du déficit de pression de vapeur, de l'énergie disponible pour la vaporisation de l'eau et du mouvement d'air à l'intérieur de la serre.

- **Ventilation et chauffage** : Cette méthode de déshumidification est la plus utilisée. En admettant de l'air extérieur plus froid et en le chauffant, on augmente sa capacité à contenir la vapeur d'eau et on réduit par conséquent l'humidité dans la serre. Plus le gradient de température entre l'extérieur et l'intérieur de la serre est élevé, plus l'échange d'air sera favorisé. Cette méthode de déshumidification montre une bonne efficacité mais son utilisation est limitée par les conditions climatiques extérieures. En hiver, lorsque la température est très froide, le gel ou l'accumulation de neige empêche parfois l'ouverture des ouvrants. De plus, l'arrivée massive d'air froid dans la serre peut créer un stress considérable à la culture. Les coûts énergétiques associés à la déshumidification par le chauffage et la ventilation sont importants.
- **Condensation** : La condensation de la vapeur d'eau sur les parois de la serre permet également de réduire le contenu en vapeur d'eau. Le type de recouvrement de serre influence ce phénomène. Les serres de double polyéthylène présentent des parois plus chaudes que celles des serres de verre puisque l'air contenu entre les deux plastiques agit comme isolant. Pour des serres de polyéthylène, il est fortement recommandé d'utiliser un plastique anti-goutte à l'intérieur, afin d'éviter le dégouttement de l'eau de condensation sur les plantes. En hiver, la température est généralement basse et la paroi de verre se refroidit de façon importante.

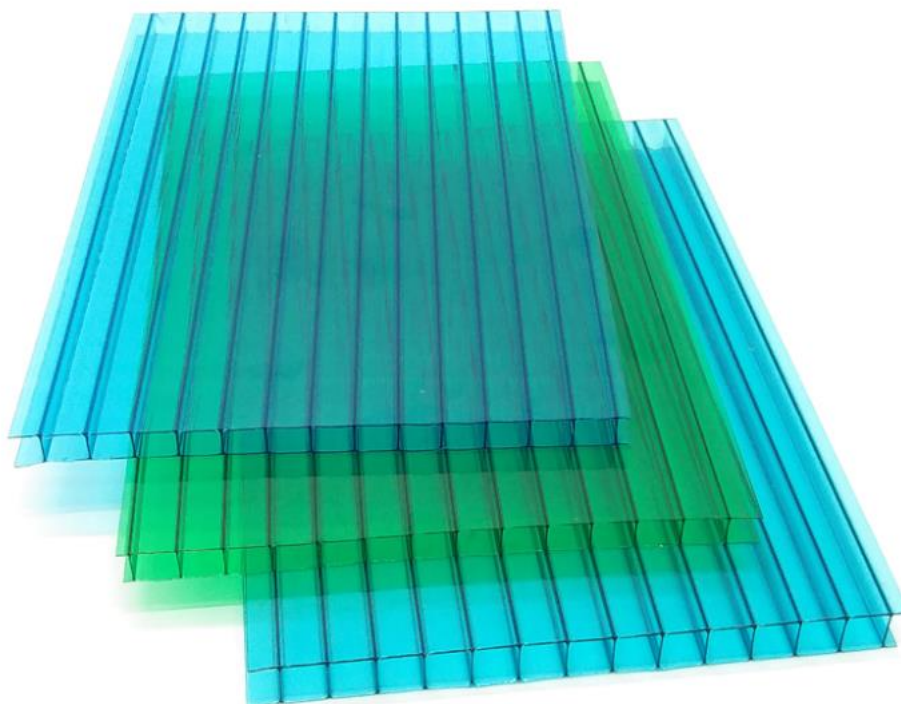


Figure 1-9 Exemple de Plastique Anti-goutte

1.5.3 Eclairage

1.5.3.1 Eclairage naturel

La lumière et la chaleur du soleil se propagent jusqu'à la terre sous forme de radiations de courte longueur d'onde qui traversent aisément une feuille de plastique ou une plaque de verre, ces radiations échauffent tout ce qu'elles touchent, sol, tablettes, terre, pots et même les plantes elles-mêmes, réfléchissent une partie de cette chaleur sous forme de rayons de grande longueur d'onde, c'est parce que le verre ne permet pas le passage des rayons de grande longueur d'onde car il se produit une accumulation de chaleur à l'intérieur de la serre, dès que la serre est plongée dans l'obscurité ou que le soleil se couche, la chaleur se dissipe par écoulement d'air à travers

Des craquelures et sous forme de radiation de grande longueur d'onde à travers les murs en dur et le bâti.

En pénétrant dans une serre en feuilles de polyéthylène, les radiations se diffusent et les rayons de grande longueur d'onde qui se forment ne sont pas piégés.

L'échauffement de la serre donne naissance à des courants de convection et l'air chaud entre en un mouvement cyclique qui varie légèrement avec la forme et la taille de la serre ainsi qu'avec l'ampleur de l'aération, en théorie les courants de convection chauffent tout l'espace intérieur mais, en fait, il subsiste généralement de petites poches d'air froid et d'air chaud.

1.5.3.2 Éclairage artificiel

La lumière joue un grand rôle dans la croissance de la plante car elle intervient dans beaucoup de phénomènes physiologiques et conditionne surtout la photosynthèse. En effet, l'énergie lumineuse fixe dans la plante le gaz carbonique et l'eau de l'air pour produire le sucre et l'amidon. Par conséquent, la croissance et le niveau de production des plantes dépendent fortement de la quantité du soleil que la culture reçoit tout au long de sa croissance. Toutefois, on peut prolonger ou raccourcir l'éclairage en utilisant respectivement les lumières artificielles ou les stores. Ceux-ci permettront aux cultures de passer au stade de développement désiré.

1.5.4 L'irrigation

L'irrigation est une opération qui consiste à apporter artificiellement de l'eau à des végétaux cultivés pour augmenter la production, et permettre leur développement normal en cas de déficit d'eau induit par un déficit pluviométrique, un drainage excessif ou une baisse de nappe, en particulier dans les zones arides.

Tout système d'irrigation se compose de quatre parties essentielles :

- Le pompage de l'eau
- Le traitement de l'eau
- La distribution de l'eau et/ou des solutions fertilisantes
- L'entreposage / Récupération de l'eau et/ou des solutions

Une pompe électrique est utilisée pour pomper l'eau du réservoir et la stocker dans le réservoir, ce dernier ayant un pressostat qui contrôle la pompe électrique, ce qui signifie que la pompe démarre automatiquement lorsque la pression est inférieure à celle indiquée par l'utilisateur.

A la sortie du ballon, l'électrovanne (un pouce) 24 VAC ou tout ce qui est connecté à l'unité de commande programmable a été fixée. Pour commander l'électrovanne, il faut choisir un capteur d'humidité.

On peut distinguer plusieurs techniques d'irrigation :

- Manuelle (arrosoir, seau...), réservée aux très petites surfaces.
- Par écoulement de surface, sous le simple effet de la gravité, au moyen de canaux et rigoles : irrigation gravitaire appelée aussi irrigation de surface, irrigation par sillons ou « à la raie ».
- Par aspersion, technique qui consiste à reproduire la pluie.
- Par micro aspersion, semblable à la précédente mais plus localisée donc plus économe en eau.
- Par micro irrigation ou goutte à goutte, technique économe en eau et qui permet d'éviter l'arrosage, mais présente le grave inconvénient de charger à la longue les sols en sels qui en modifient les caractéristiques.
- Par infiltration, au moyen de tuyaux poreux enterrés, variante de la technique de la goutte à goutte.

1.5.5 Ombrage

Pour ombrager une serre, il y a deux moyens fondamentaux : peindre les vitres ou y pulvériser un liquide, ou bien utiliser des stores. L'inconvénient majeur de l'emploi des liquides pour ombrager est que, pendant les inévitables périodes de temps gris et frais de l'été, Les plantes souffrent d'un manque de lumière et de chaleur, juste au moment où elles en auraient le plus besoin, c'est pourquoi les stores sont préférables car plus efficaces, les stores à rouleau se fixent indifféremment à l'extérieur sont les plus efficaces car ils évitent l'accumulation de la chaleur, les stores montés à l'intérieur des vitres empêchent la lumière d'atteindre les plantes mais la chaleur traverse le vitre et chauffe la serre comme elle le ferait sans store, il vaut donc mieux, en général, monter les stores à l'extérieur, mais en tenant compte des intempéries, surtout des vents violents, on peut éventuellement les enrouler en hiver pour obtenir une certaine protection contre les gelées.



Figure 1-10 Exemple d'ombrage d'une serre

1.6. PROBLEMATIQUE

Dans plusieurs cas, les serres agricoles se trouvent dans des endroits loin ou difficile d'intervenir journalier afin de contrôler et de réaliser un bon climat et environnement pour notre produit.

1.7. SOLUTION PROPOSEE

Notre projet permettra réaliser une serre agricole automatique commandable à distance par un système intelligent qui permet d'intervenir à base aux informations nécessaires aux produits agricoles.

1.8. CONCLUSION

Après avoir des informations sur les différents types des serres, et les paramètres qu'il faut mesurer pour commander les actionneurs qui font résoudre les besoins des produits alimentaires, dans le chapitre suivant nous allons parler aux outils de travail qu'il faut utiliser pour réaliser la solution proposée.

CHAPITRE 02
ENVIRONNEMENT
ET OUTILS DE
TRAVAIL

2.1. INTRODUCTION

Le travail que nous décrivons dans ce mémoire est divisé en deux parties. La première partie consiste à l'automatisation du système d'irrigation dans une serre qui se sert d'un ensemble de capteurs et de microcontrôleurs afin de s'autogérer. La deuxième partie consiste à se servir de contrôler notre serre à distance à l'aide d'un Smart phone via WIFI. L'objectif de ce chapitre est donc de présenter l'ensemble des outils matériels et logiciels utilisés pour la réalisation de notre travail.

Dans notre projet le choix des capteurs et actionneurs dépend de leurs applications, et des différents paramètres à contrôler. Ce chapitre comporte une présentation du prototype proposé, une présentation des capteurs utilisés et les logiciels utilisés afin d'aboutir à une solution performante.

2.2. LA CARTE ARDUINO [4]

2.2.1 Présentation de l'Arduino UNO

Les cartes Arduino font partie de la famille des microcontrôleurs. Un microcontrôleur est une petite unité de calcul accompagnée de mémoire, de ports d'entrée/sortie et de périphériques permettant d'interagir avec son environnement.

Une carte Arduino est une interface programmable capable de piloter des capteurs et des actionneurs afin de simuler ou créer des systèmes automatisés. Elle peut stocker un programme et le faire fonctionner. La carte reçoit des informations analogiques ou numériques sur ces entrées. Le microcontrôleur traitera ces informations et les transmettra vers les sorties numériques.

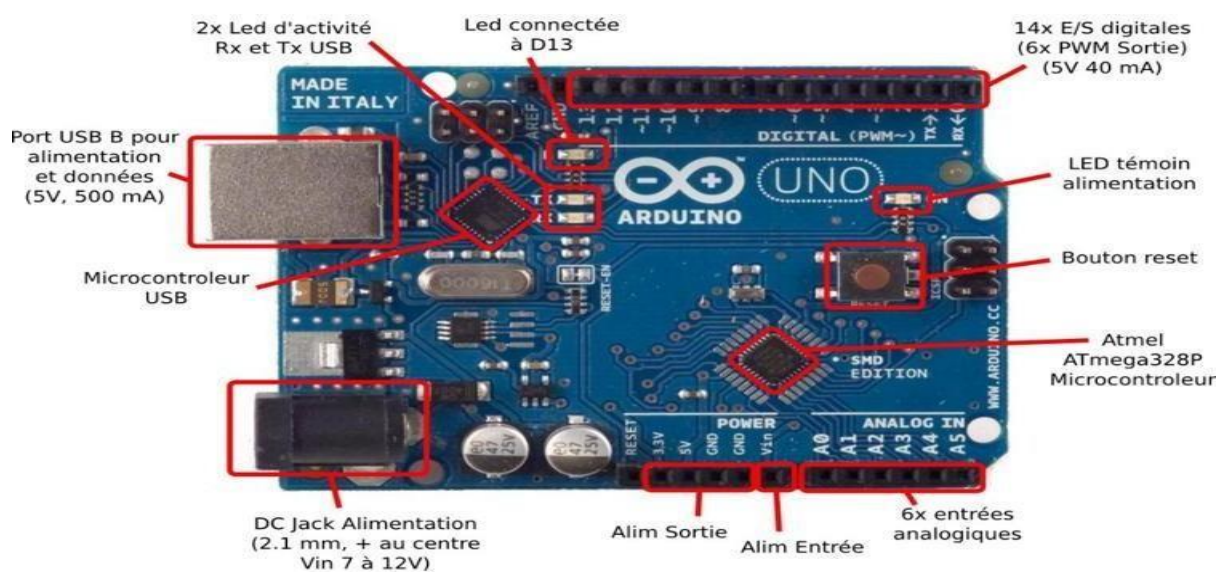


Figure 2-1 Composants techniques de l'Arduino

2.2.2 Fiche Technique de l'Arduino UNO

Microcontrôleur	ATmega328P
Tension de fonctionnement	5V
Tension d'entrée	7-12V
Tension d'entrée	6-20V
E / S numériques Pins	14
PWM numérique E / S Pins	6
Pins d'entrée analogique	6
Courant DC pour 3.3V Pin	50 mA
Mémoire flash	32 KB
SRAM	2 KB
Vitesse de l'horloge	16 MHz
Longueur	68,6 mm
Largeur	53,4 mm
Poids	25 g

Tableau 2-1 Fiche techniques de l'Arduino

2.2.3 L'Alimentation

La carte Arduino UNO peut être alimentée via la connexion USB ou avec une alimentation externe. La source d'alimentation est automatiquement sélectionnée. Une alimentation externe peut provenir soit d'un adaptateur AC-DC ou d'une batterie. L'adaptateur peut être connecté en branchant une prise 2.1mm dans la prise d'alimentation de la carte ou à partir d'une batterie connectée dans le pin (ou broche) GND et V-in (alimentation externe). Le processeur peut fonctionner sur une alimentation externe de 6 à 20 volts. Cependant, si la tension est inférieure à 7V, le pin 5V peut fournir moins de cinq volts et le processeur peut devenir instable. Si la tension est supérieure à 12V, le régulateur de tension peut surchauffer et endommager la carte.

La plage recommandée est de 7 à 12 volts. Les pins (ou broches) d'alimentation sont les suivantes :

- Vin : Tension d'entrée à la carte Arduino à l'aide d'une source d'alimentation externe (par opposition à 5 volts de la connexion USB ou une autre source d'alimentation régulée). Si l'alimentation en tension est faite par l'intermédiaire de la prise d'alimentation, on pourra y accéder via ce pin.
- 5V : Ce pin délivre un 5V régulé par la carte. Le processeur peut être alimenté soit à partir de la prise d'alimentation DC (7-12V), le connecteur USB (5V), ou le pin Vin de la carte (7-12). La fourniture d'une tension via les 5V ou 3,3V contourne le régulateur, et peut endommager votre processeur.
- 3V3 : Une alimentation de 3,3 volts générée par le régulateur. La consommation de courant maximale est de 50mA.
- GND: Masse

2.2.4 Communication

Arduino a un certain nombre de moyens pour communiquer avec un ordinateur, une autre carte Arduino, ou autres microcontrôleurs. L'ATmega328 fournit UART TTL (5V) en communicationsérie, disponible sur les broches numériques 0 (RX) et 1 (TX).

2.2.5 Les Broches

- Les broches Analogiques

L'UNO dispose de 6 entrées analogiques, A0 à A5, dont chacune fournit 10 bits de résolution (ou 1024 valeurs différentes). Par défaut, la tension est de 5volts.

- Les broches Numériques

Chacune des 14 broches numériques sur la carte UNO peut être utilisée comme une entrée ou une sortie. Ils fonctionnent à 5 volts. Chaque broche peut fournir ou recevoir 20 mA en état de fonctionnement recommandée et a une résistance de pull-up interne (déconnecté par défaut) de 20-50k ohm. 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 au microcontrôleur.

2.2.6 La Mémoire

L'ATmega328 a 32 Ko (avec 0,5 KB occupées par le bootloader). Il a également 2 Ko de SRAM et 1 Ko de mémoire EEPROM.

2.3. LES CAPTEURS

Un capteur est un dispositif permettant de détecter, en vue de le quantifier et de le représenter, un phénomène physique sous la forme d'un signal, généralement électrique. Le capteur se différencie du détecteur et du senseur par sa possibilité de délivrer une grandeur physique directement utilisable pour une mesure ou une commande.

2.3.1 Capteur de Température et d'Humidité DHT22 [5]



Figure 2-2 Capteur DHT22

Le DHT22 est un capteur qui nous donne la température et l'humidité avec une très grande précision. Il est fiable et peut nous donner des valeurs négatives par rapport à la mesure de ces grandeurs ; Ce dernier est composé de 04 pins (1-Vcc, 2-Data, 3-Non utilisé et 4-GND).

2.3.2 Capteur d'Humidité du Sol

Le capteur d'humidité du sol mesure l'humidité du sol à partir des changements de conductivité électrique de la terre (la résistance du sol augmente avec la sécheresse), il est composé d'une platine qui constitue le conditionnement et une fourche résinée qui protège contre l'oxydation et se plante verticalement dans la terre.

Le fonctionnement électrique de ce capteur est basé sur l'immersion des deux tiges, qui en contact avec le sol permettent la circulation d'un courant, ce qui permet la lecture du niveau d'humidité par rapport à la résistance. Plus y'a d'eau dans la terre, plus la conduction de courant entre les tiges est meilleure, en raison de la faible résistance, ce qui permet à la tension au niveau du capteur d'approcher 5V. Alors que la conductivité d'un sol sec est faible en raison d'une résistance élevée, résultant en un signal proche de 0V.

Le branchement du capteur se fait comme suit : On relie la broche V_{CC} au 5V, GND à la masse et la sortie à une entrée analogique du microcontrôleur, c'est cette dernière qui nous permettra de recueillir l'information (variation de la tension). Il existe une sortie numérique active quand un seuil réglable est dépassé.

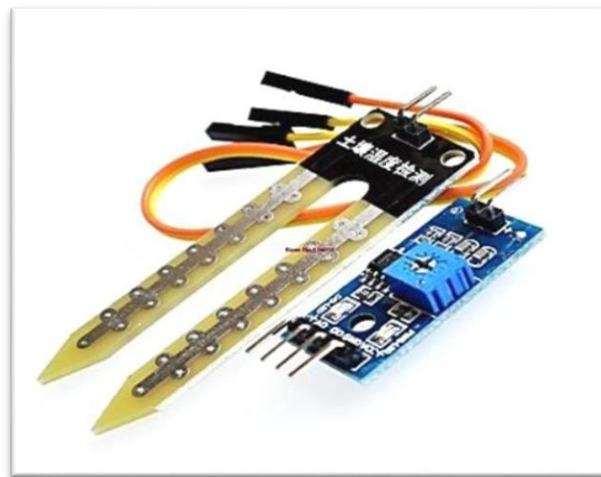


Figure 2-3 Capteur d'Humidité de Sol

2.3.3 Capteur de Lumière LDR [3]

Une photorésistance est un composant électronique dont la résistivité varie en fonction de la quantité de lumière incidente. On peut également la nommer résistance photo-dépendante (Light Dependent Resistor LDR) ou cellule photoconductrice.

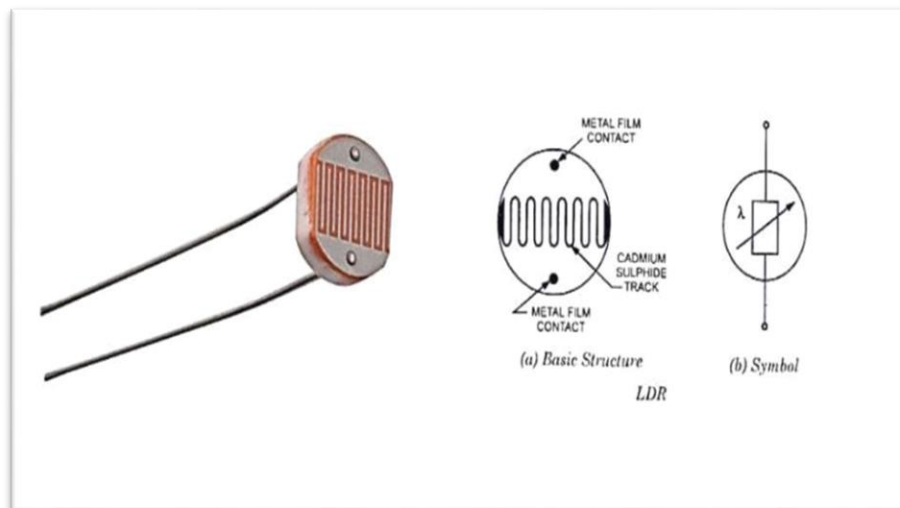


Figure 2-4 Capteur de Lumière LDR

2.3.4 Capteur d'Air MQ 135 [5]

Pour la qualité de l'air, un petit capteur très pratique nous donne un indicateur global de la qualité de l'air en tenant compte de plusieurs facteurs dont le niveau de CO₂, alcool et le monoxyde de carbone.

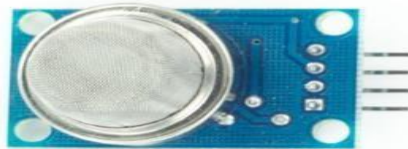


Figure 2-5 Capteur d'Air MQ 135

Le MQ135 est un capteur qui permet de mesurer la qualité de l'air. Ce capteur est sensible aux principaux polluants présents dans l'atmosphère. Il est sensible au CO₂, à l'alcool, au Benzène, à NO_x et NH₃.

Le MQ135 est très économique et utilise le SnO₂ qui bénéficie d'une conductivité inférieure en présence d'air "pur". La conductivité augmentera en fonction de la pollution de l'air. La conversion d'un signal de sortie correspondant à la concentration du gaz. Il est sensible aux gaz de type ammoniac, Oxyde d'azote, Benzène, fumée, CO₂ et autres gaz nocifs.

Le tableau ci-dessous représente les différents niveaux de pollution dans l'air :

IQA	niveau de Pollution d'air	Impact sur la santé
0-50	Bon	La qualité de l'air est jugée satisfaisante, et la pollution de l'air pose peu ou pas de risque.
50-100	Modéré	La qualité de l'air est acceptable
101-150	Mauvais pour les groupes sensibles	La qualité de l'air est acceptable, pour certains polluants, il peut y avoir un problème de santé modérée pour un très petit nombre de personnes qui sont particulièrement sensibles à la pollution de l'air.
151-200	Mauvais	Tout le monde peut commencer à ressentir des effets sur la santé ; les membres des groupes sensibles peuvent ressentir des effets de santé plus graves.
201-300	Très mauvais	Avertissements de santé de conditions d'urgence. Toute la population est plus susceptible d'être affectée.
300+	Dangereux	Alerte de santé : tout le monde peut ressentir des effets de santé plus graves.

Tableau 2-2 Les niveaux de pollutions d'air

2.4. AFFICHEUR LCD 2x16 CARACTERES

Les afficheurs à cristaux liquides, autrement appelés afficheur LCD (Liquid Crystal Display), sont des modules compacts intelligents et nécessitent peu de composants externes pour un bon fonctionnement.

Plusieurs afficheurs sont disponibles sur le marché et différents les uns des autres, non seulement par leurs dimensions, (de 1 à 4 lignes de 6 à 80 caractères) mais aussi par leurs tensions de service.

Dans notre distributeur de gel hydro alcoolique, on a utilisé l'afficheur LCD 2x16 caractères, comme le montre la Figure 2-6, pour afficher la température de la personne testée.



Figure 2-6 Afficheur LCD 2x16 Caractères

2.4.1 Caractéristiques de l’Afficheur LCD

- Format : 2 lignes de 16 caractères.
- Connecteur : 1 rangée en haut à gauche.
- Rétro-éclairage : LED (bleu).
- Mode d’affichage disponible : STN.
- Mode d’interfaçage avec un microcontrôleur externe : Mode 4 ou 8 bits.
- Fonctions accessibles par programmation : Effacement de l’écran, curseur en haut à gauche, affichage/extinction de l’écran, affichage/extinction du curseur, clignotement caractère, décalage de l’écran ou du curseur.
- Circuit de RESET automatique à la mise sous tension.
- Circuit oscillateur interne.
- Mémoires intégrées : CG ROM (caractères prédéfinis sur matrice 5 x 7 et 5 x 10 points), CG RAM (générateur de caractères sur matrices 5 x 7 et 5 x 10 points), DD RAM (mémoire usuelle 80 octets).
- Contrôleur / driver intégrés : S6A0069 ou équivalent.
- Consommation rétro-éclairage : 110 mA. [11]

2.4.2 Caractéristiques Mécaniques de l’Afficheur

On a organisé les spécifications mécaniques de LCD 2x20 caractères dans le tableau ci-dessous

Dimensions du module	80 (L) x 36 (H) x 13,2 (e) mm.
Dimensions de la fenêtre	65 (L) x 15 (H) mm.
Dimensions d'un caractère	2,95 x 4,35 mm (matrice 5 x 7 points).
Dimensions des points constituant la matrice	0,55 x 0,50 mm.
Diamètre trous de montage	2,5 mm
Poids	35 gramme

Tableau 2-3 Spécifications Mécaniques de LCD 2x20 Caractères

2.4.3 Broches de l’Afficheur

Il existe 16 broches (figure 2-6), qui permettent le branchement de LCD 2x16 caractères.

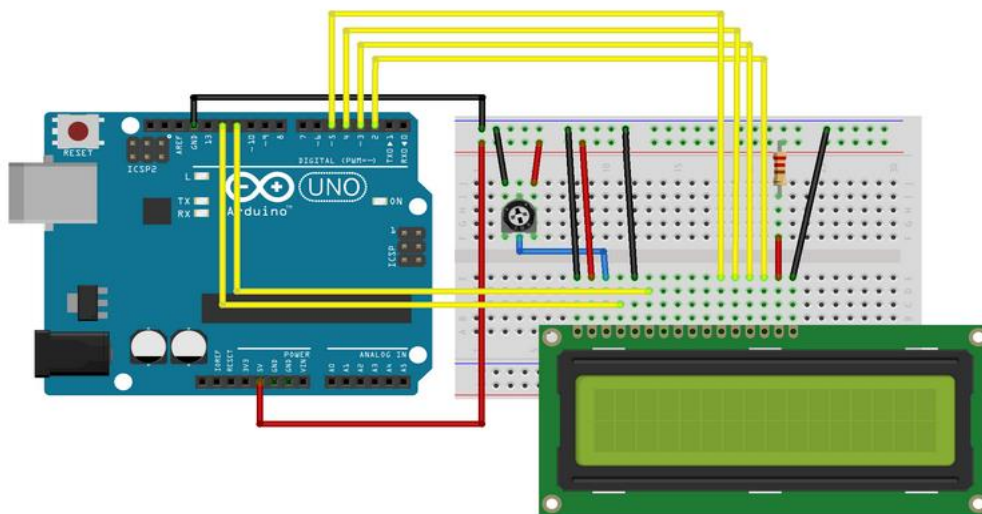


Figure 2-7 Les Broches de l’Afficheur LCD 2x16.

Le tableau ci-dessous exprime le nom et la fonction de chaque broche.

Broche	Nom	Fonction
1	Vss	Masse
2	Vdd	Alimentation positive +5V
3	Vo	Cette tension permet, en la faisant varier entre 0 et +5V, le réglage du contraste de l'afficheur.
4	RS	Sélection du registre (Register Select) Grâce à cette broche, l'afficheur est capable de faire la différence entre une commande et une donnée. Un niveau bas indique une commande et un niveau haut indique une donnée.
5	R/W	Lecture ou écriture (Read/Write) L : Écriture H : Lecture
6	E	Entrée de validation (Enable) active sur front descendant. Le niveau haut doit être maintenue pendant au moins 450 ns à l'état haut.
7	D0	Bus de données bidirectionnel 3 états (haute impédance lorsque E=0)
8	D1	
9	D2	
10	D3	
11	D4	
12	D5	
13	D6	
14	D7	
15	A	Anode rétroéclairage (+5V)
16	K	Cathode rétroéclairage (masse)

Tableau 2-4 Brochage de LCD 2x16 Caractères

N.B: Les broches 15 et 16 ne sont présentes que sur les afficheurs LCD avec rétro-éclairage

2.5. LE MODULE WIFI ESP8266 [6]

2.5.1 Définition

L'ESP8266 (figure 2-8) est un module qui sert à faire la liaison avec un réseau WIFI disponible. C'est une puce qui contient un microcontrôleur interne qui donne l'accès sans fil aux réseaux, avec d'autres options plus développées. Plus précisément, l'ESP8266 est un système-sur-puce qui fonctionne dans la bande des fréquences de 2,4 GHz, ce module utilise le protocole de communication WIFI 802.11 b/g/n, qui permet au microcontrôleur interne de se connecter à un réseau WIFI et de faire des connexions TCP / IP simples, ou d'être détectable par d'autres périphériques qui utilisent le même protocole WIFI.

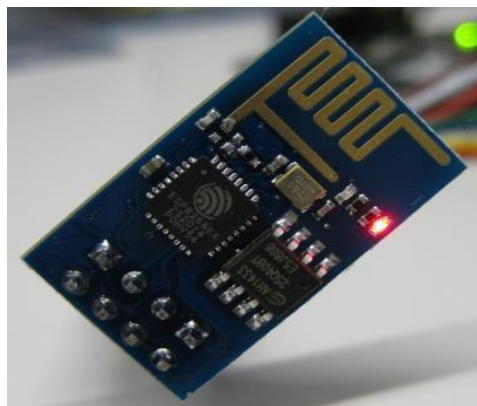


Figure 2-8 Le module ESP8266

L'ESP8266 utilise huit Pins (Figure 2-9) parmi les trente-deux pins de l'ESP8266. Ces huit pins suffisent pour que le module travaille en mode WIFI.

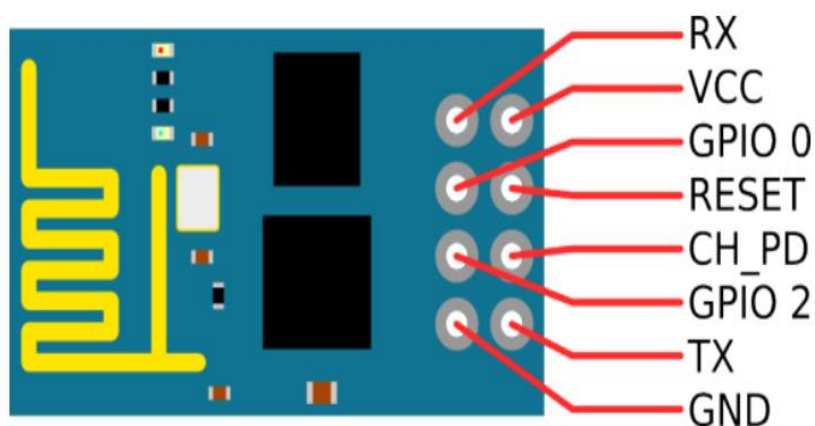


Figure 2-9 Pins du module ESP8266

2.5.2 Les commandes AT

Une fois le module ESP8266 est relié avec un microcontrôleur externe, il faut que notre routeur ou notre ordinateur soit capable de communiquer avec le module. Cette communication est assurée par les commandes AT, qui constituent un langage de communication pour le module ESP8266 pour exécuter les opérations d'une façon automatique (accéder à un réseau WIFI, ajouter un mot de passe, quitter le réseau...). On abordera ce point plus en détail dans le chapitre suivant qui va décrire la conception et la réalisation.

2.6. LES RELAIS ELECTROMECHANIQUES

Un relais électromécanique (Figure 2-10) est un organe électrique permettant de distribuer la puissance à partir d'un ordre émis par la partie commande -Signale de sortie Arduino-. Ainsi, un relais permet l'ouverture et la fermeture d'un circuit électrique de puissance -source pour alimentation des actionneurs- à partir d'une information logique. Les deux circuits, puissance et information, sont complètement isolés et peuvent avoir des caractéristiques d'alimentation électrique différentes.

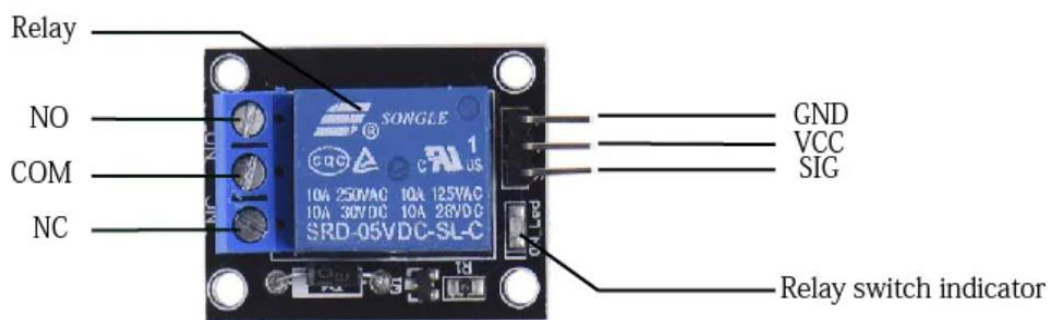


Figure 2-10 Relais électromécanique

2.7. APPLICATION REMOTE XY [7]

Remote XY est un moyen facile de créer et d'utiliser une interface utilisateur graphique mobile pour les cartes de contrôleur à contrôler via un smartphone ou une tablette. Le système comprend :

- Editeur d'interfaces graphiques mobiles pour cartes contrôleur, situées sur le site remotexy.com
- Application mobile Remote XY qui permet de se connecter au contrôleur et de le contrôler via une interface graphique.

L'application mobile androïde a été conçue à l'aide de Remote XY suivant les étapes suivantes :

- **Étape 1** : Créez un projet d'interface utilisateur graphique (Figure 2-11)



Figure 2-11 L'éditeur de l'application Remote XY

- **Étape 2** Configuration du projet

Dans le volet droit, sélectionnez les paramètres suivants sous l'onglet "Configuration" (Figure 2-12).



Figure 2-12 L'onglet « configuration »

Les paramètres indiquent que l'ESP8266 doit être connecté à l'Arduino via la série matérielle (broches 0 et 1) à une vitesse de 115200 (Figure 2-13).

Les paramètres spécifient également le nom du futur point d'accès et le mot de passe du point d'accès.

Properties

► Configuration

▼ Module interface

Connection interface:
Hardware Serial ▼

Serial port:
Serial, pins 0(RX) and 1(TX) ▼

Speed (baud rate):
115200 ▼

Wi-Fi access point:

Name (SSID):
RemoteXY

☐ Open point

Password (8 or more chars):
12345678

Port:
6377

Figure 2-13 Paramètres de l'interface du module

○ **Étape 3 : Obtenez le croquis de l'Arduino**

Appuyez sur le bouton "Obtenir le code source" pour télécharger le code et l'ouvre dans l'IDE Arduino.

Téléchargez également la bibliothèque Remote XY (le lien "Télécharger la bibliothèque") à partir de cette page.

La bibliothèque étant installée correctement, le code source de l'esquisse doit être compilé sans erreur.

○ **Étape 4 : Connectez-vous à partir de l'application mobile**

Après l'installation de l'application mobile Remote XY sur le Smartphone, appuyez sur le nouveau bouton de connexion "+" (Figure 2-14) dans le panneau supérieur de l'application. Dans la fenêtre qui s'ouvre, sélectionnez la connexion "Point d'accès WIFI".

Dans l'application Androïde, activez le WIFI et appuyez sur le bouton de mise à jour pour obtenir la liste des points d'accès disponibles. Sélectionnez le point d'accès Remote XY. Dans la fenêtre qui s'ouvre, saisissez le mot de passe du point d'accès "12345678", puis appuyez sur le bouton "Connecter".

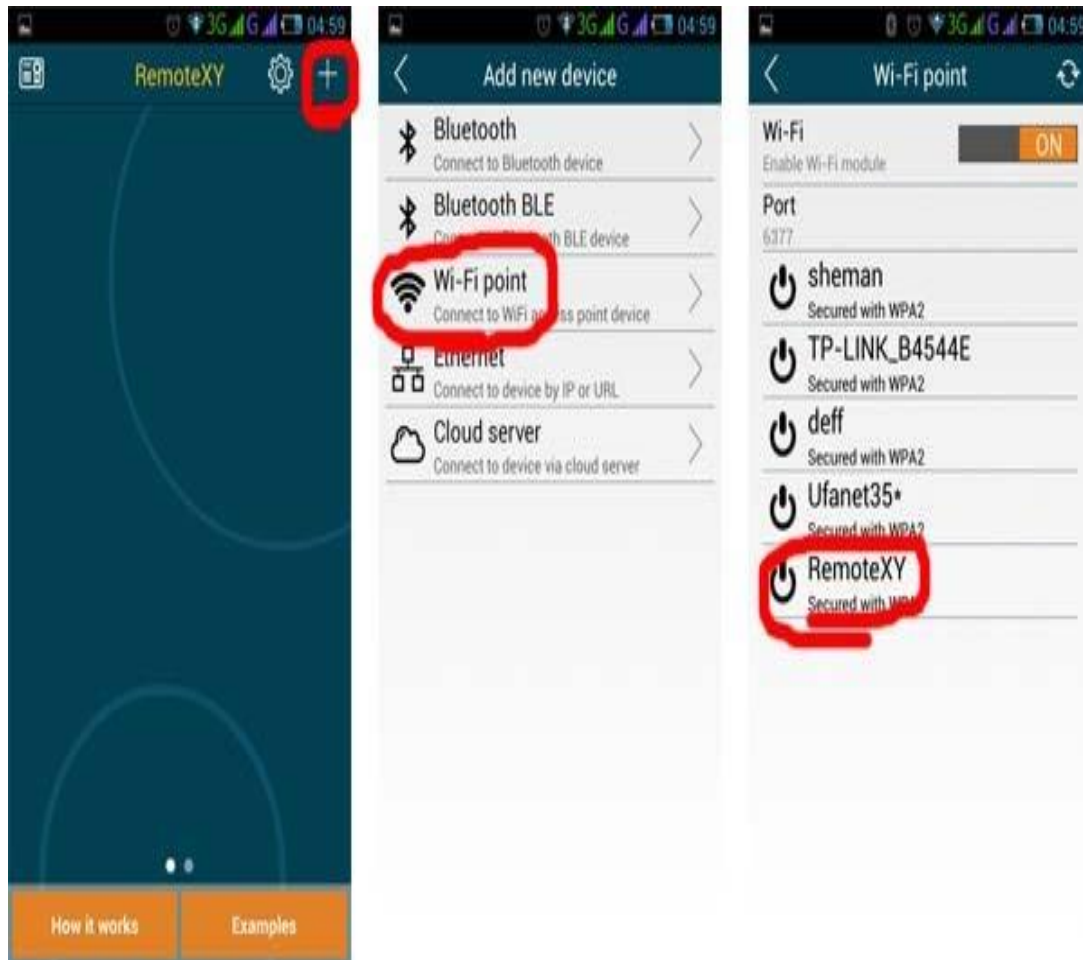


Figure 2-14 Configuration réseau de l'application Remote XY sur Smartphone

2.8. LOGICIEL ARDUINO IDE

Le logiciel Arduino est une version simplifiée de programme C++. C'est un ensemble d'instructions qui informe le matériel de ce que faire et comment le faire. C'est un environnement de développement intégré, qui se présente sous la forme d'une interface graphique. Cette interface permet de lire et écrire des lignes de texte et d'interagir d'une manière visuelle avec l'ordinateur. Les programmes écrits pour Arduino s'appellent des croquis.

La fenêtre qui constitue l'interface graphique d'Arduino est divisée en quatre zones principales (repérées sur la Figure 2-15) :

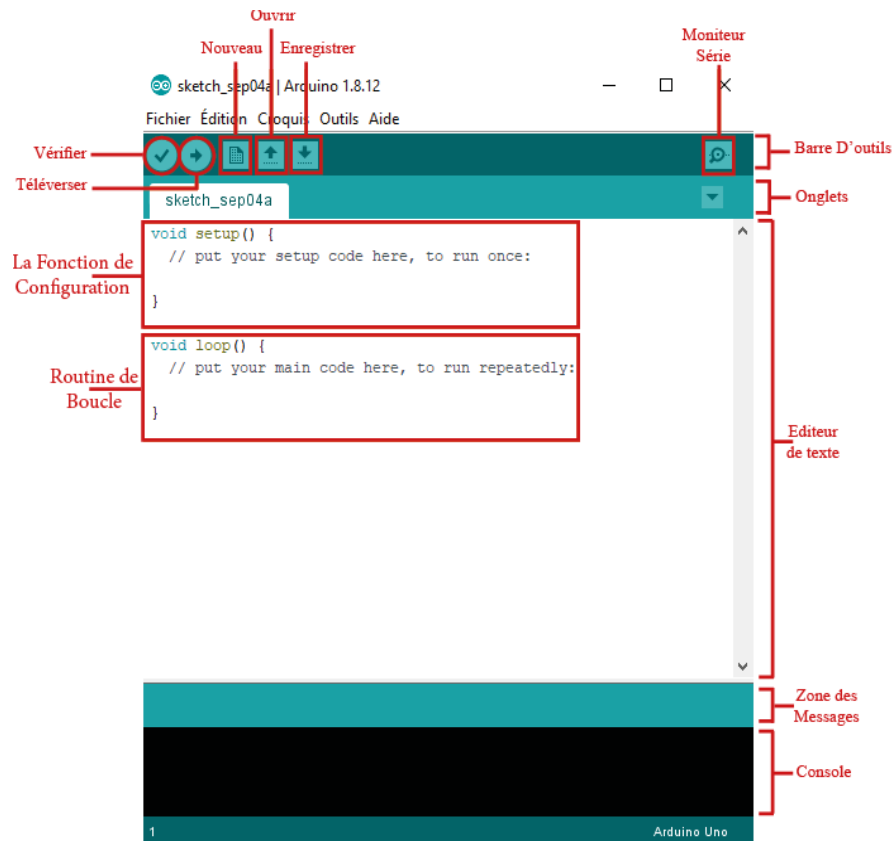


Figure 2-15 Les Zones de l'Interface Graphique

- La barre de menus : Elle contient des menus déroulants pour tous les outils, paramètres et informations du programme.
- La barre d'outils : La barre d'outils contient plusieurs boutons qui sont communément utilisés lorsque vous écrivez des croquis pour Arduino. Ces boutons, qui sont aussi accessibles dans la barre de menus, permettent d'accomplir les actions suivantes :
 - Vérifier : Assurer que le code est correct pour le programme Arduino. Aussi désigné par le terme de compilation, ce processus est une sorte de correction de l'orthographe et de la grammaire. Sachez cependant que si le compilateur vérifie le code pour des erreurs apparentes, il n'y a aucune garantie que le dessin fonctionnera correctement.
 - Téléverser : Envoie le croquis à la carte Arduino connectée à l'ordinateur. Le croquis est compilé avant d'être envoyé.
 - Nouveau : Crée un nouveau croquis.
 - Ouvrir : Ouvre un croquis existant.
 - Enregistrer : Sauvegarde le croquis courant.
 - Moniteur série : Permet de visualiser les données qui sont envoyées ou reçues par la carte Arduino.
- Éditeur de texte : Le croquis est affiché sous forme d'un texte dans cette zone. C'est approximativement un éditeur de texte, avec des fonctionnalités supplémentaires. Il contient :

- La fonction de configuration : C'est la fonction « **Void Setup** », elle permet d'initialiser et sélectionner l'état initial des variables. Le code initial va être exécuté une seule fois.
- Routine de boucle : C'est la fonction « **Void Loop** », elle présente une boucle qui fonctionne ou exécute le code principal d'une manière fréquente.
- Zone des messages : C'est l'une des premières sources d'explications. Elle montre le message de l'IDE dans la zone noire.

2.9. PLATEFORMES DE SIMULATION

Avant de mettre directement la main à la phase de réalisation, il est en effet possible de tester notre composants et projet sur un simulateur Arduino. Il en existe plusieurs, mais un bon simulateur devra nous permettre de reproduire de manière virtuelle plusieurs aspects du processus, notamment les points suivants :

- Concevoir nos propres composants et circuits (ou en importer depuis un catalogue).
- Simuler des interactions entre l'Arduino, les interfaces d'entrées et de sorties et le programme.
- Concevoir des cartes et des schémas (facultatif, mais utile).
- Exporter des cartes et des schémas pour produire des circuits imprimés (facultatif).

L'inconvénient qui se pose, est l'inexistence de tous les composants et le sur une seule plateforme de simulation, pour cela nous avons utilisé le TINKERCAD et le WOKWI afin de simuler tous les composants utilisés dans notre projet.

2.9.1 Tinkercad

Pour réaliser des projets sur Tinkercad il faut suivre les étapes suivantes :

- Rejoindre Tinkercad par création d'un nouveau compte ou se connecter par un compte Gmail (Figure 2-16).

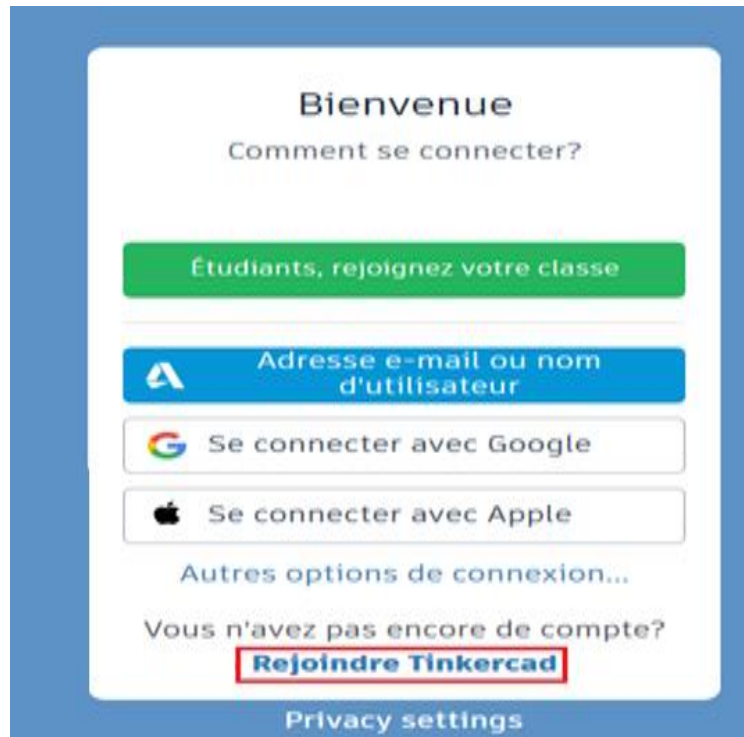


Figure 2-16 Rejoindre Tinkercad

- Création du circuit en cliquant sur les étapes 1 et 2 sur la Figure 2-17.

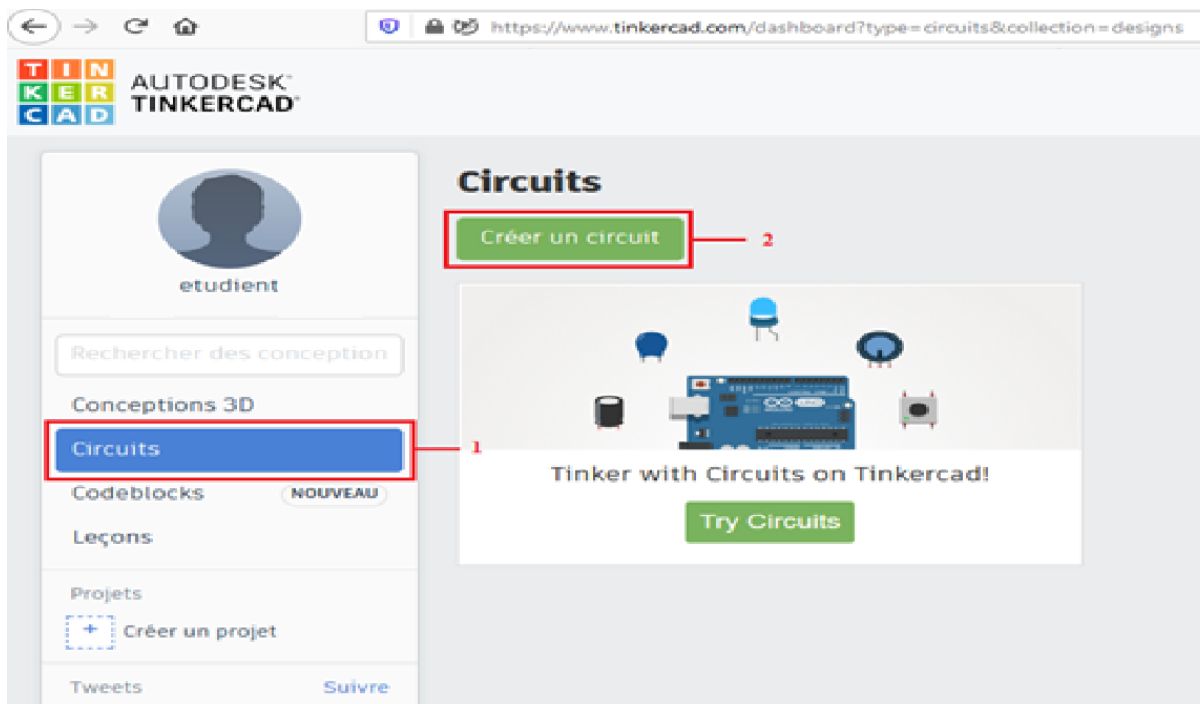


Figure 2-17 Création du circuit

- Choix des composants du circuit comme le montre la figure 2-18.

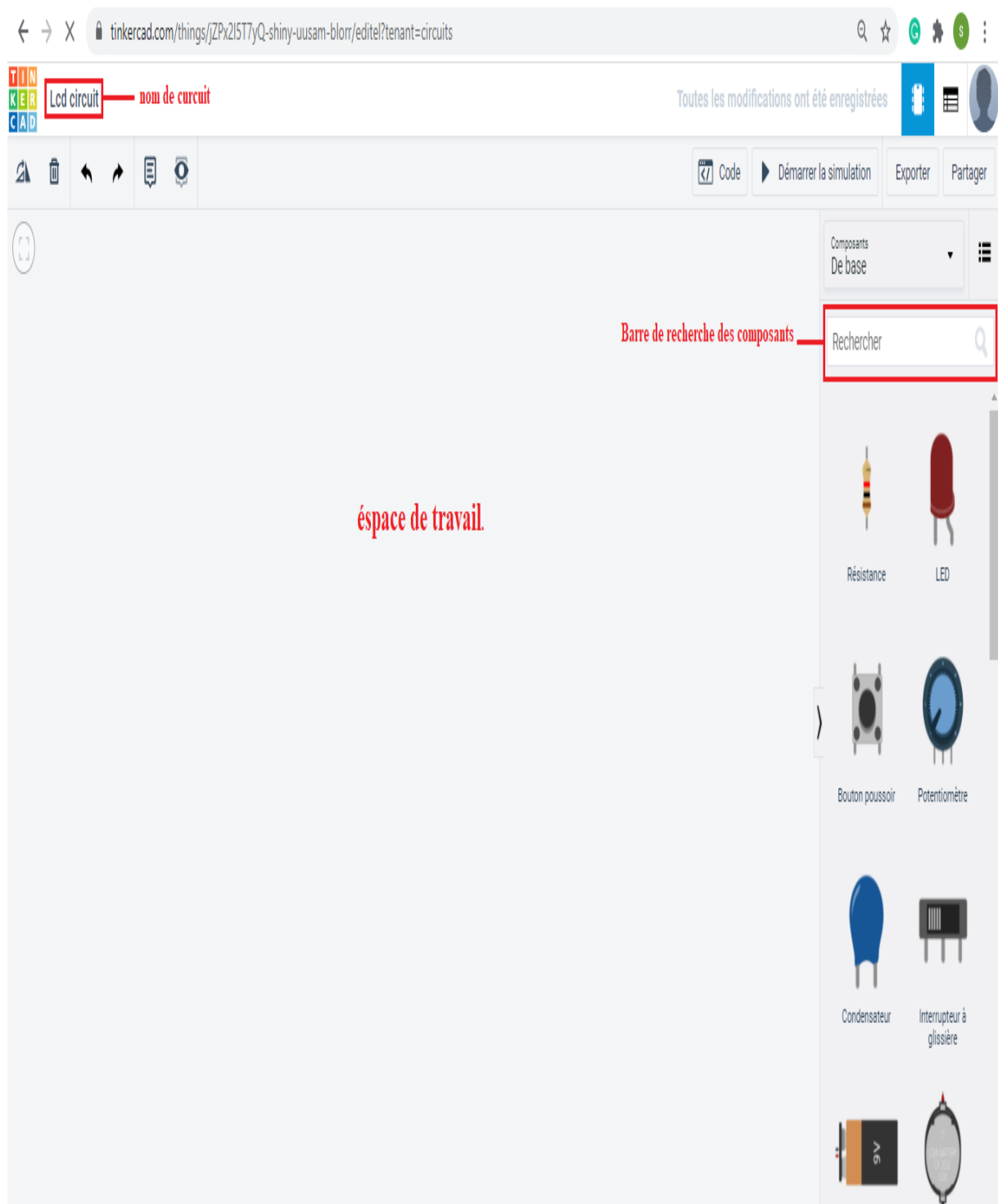


Figure 2-18 Choix des composants

- L'écriture du code source sur le microcontrôleur de la carte Arduino et la simulation du fonctionnement comme le montre la figure 2-19.

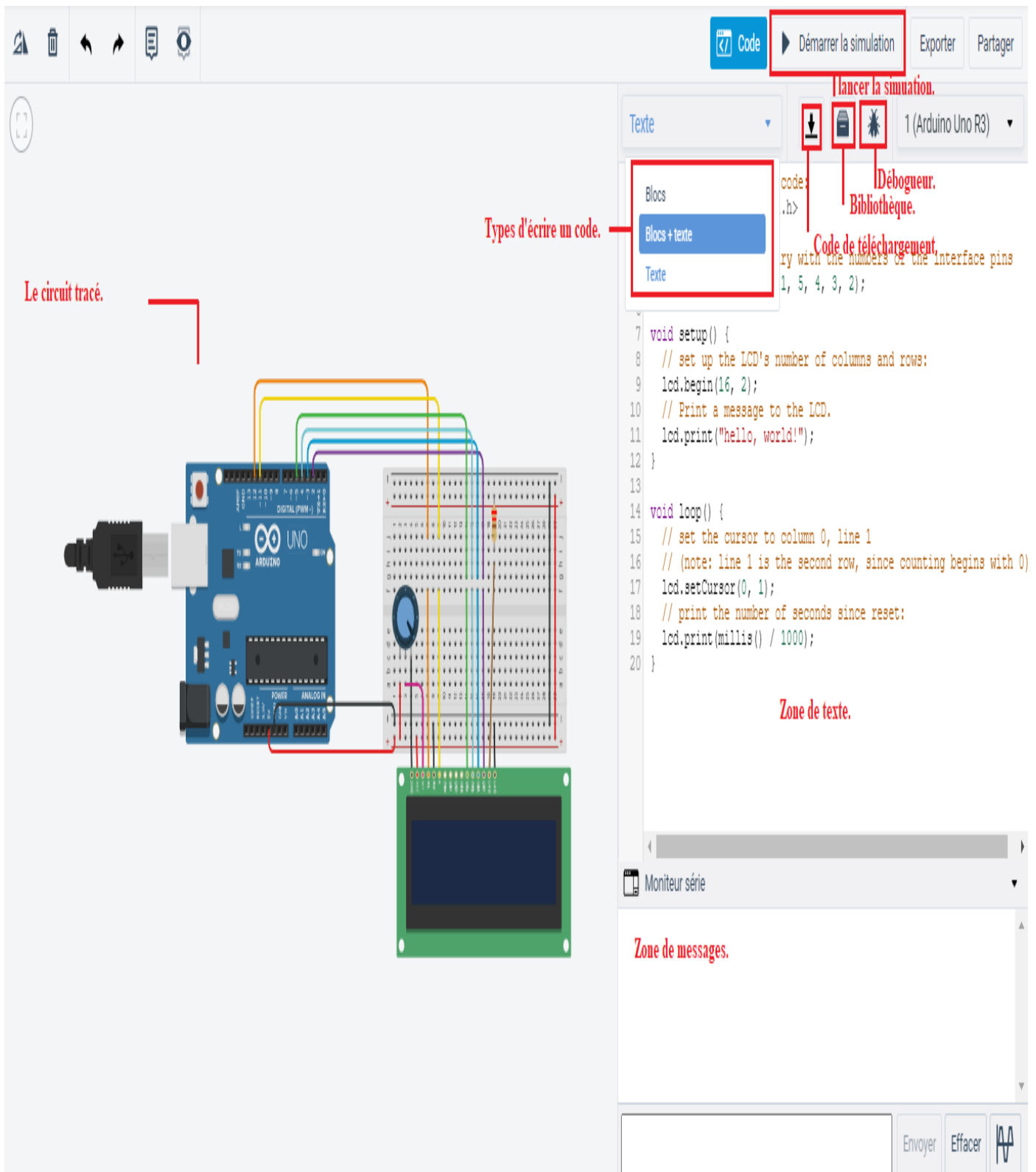


Figure 2-19 Conception et Simulation du Circuit par Tinkercad

2.9.2 Wokwi

La réalisation des projets et la simulation sur la plateforme Wokwi en suivant les étapes suivantes :

- Rejoindre la plateforme peut être sans inscription en choisissons la carte Arduino utilisée (Figure 2-20)

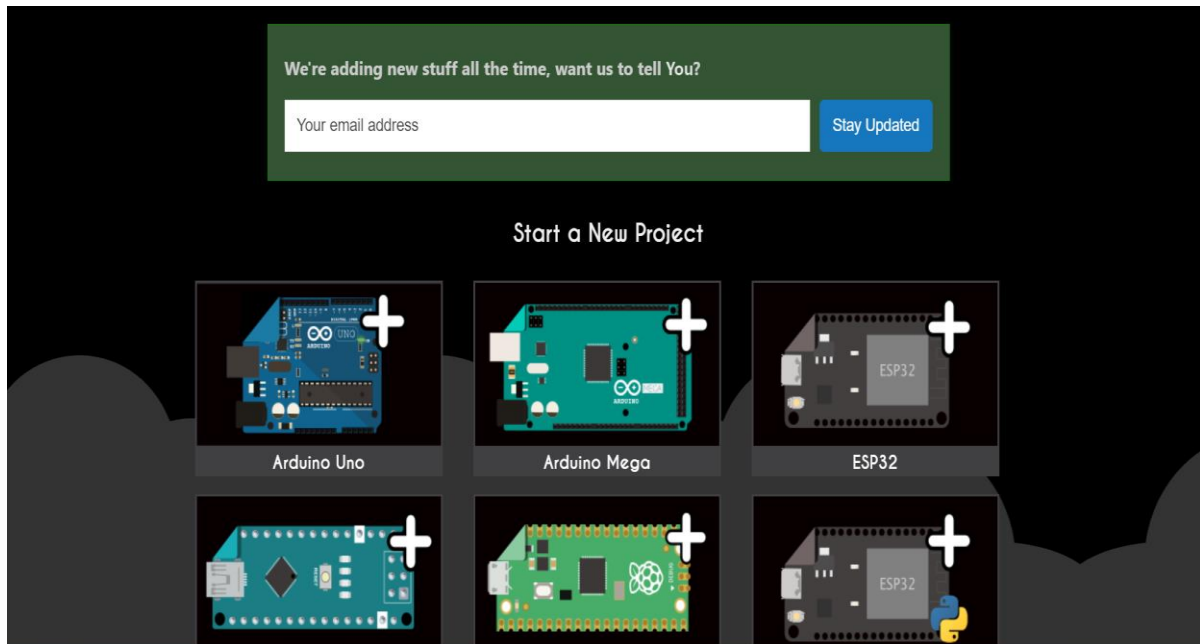


Figure 2-20 Rejoindre Wokwi

- Le choix des composants se fait en cliquant sur l'étape 1 (Figure 2-21)
- Nous écrivons le code source sur la zone comme indiqué sur l'étape 2 (Figure 2-21)
- La 3eme étape représente le symbole où nous exécutons la simulation (Figure 2-21)

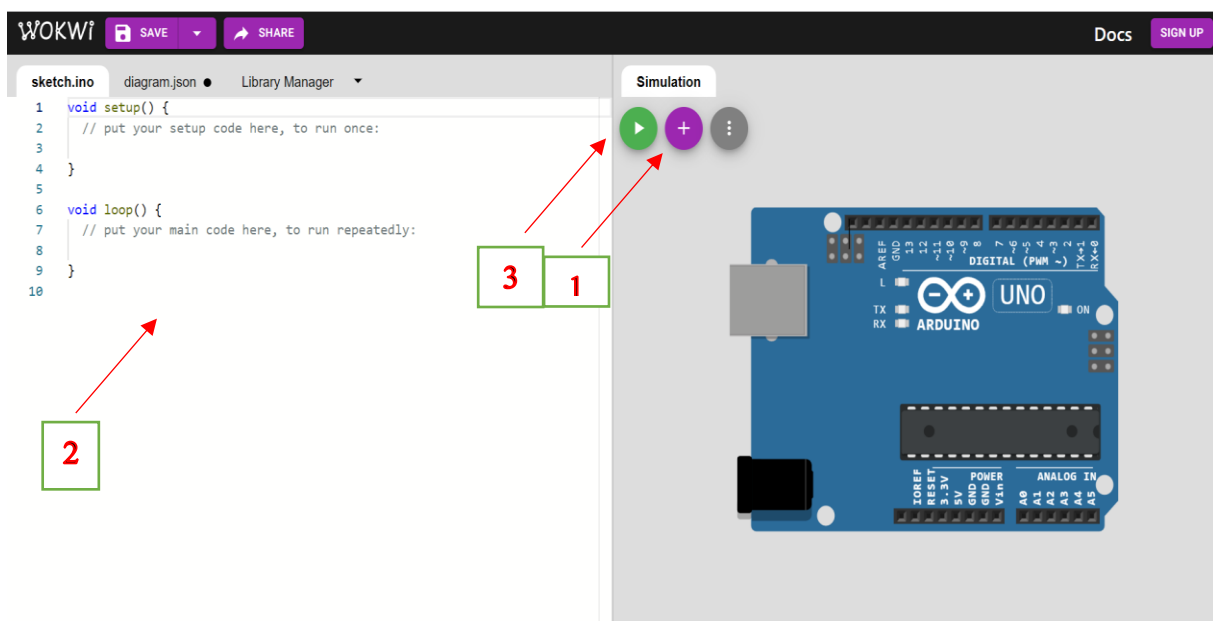


Figure 2-21 Conception et Simulation du Circuit par Wokwi

2.10. CONCLUSION

Dans ce chapitre, nous avons présenté les principaux composants que nous utiliserons pour la conception de notre nouveau système, à savoir : une carte Arduino UNO, le capteur de température DHT22, le capteur d'humidité de sol, capteur de Lumière LDR, le capteur d'Air MQ 135 et l'afficheur LCD 2x16 caractères ainsi la présentation de l'application Remote XY et les plateformes de simulation Tinkercad et Wokwi.

Ce qui sera présenté dans les chapitres suivants porte sur les étapes de développement, de programmation et de mise en œuvre du programme d'automatisation de notre serre.

CHAPITRE 03 :

PREPARATION DE

LA REALISATION

3.1. INTRODUCTION

La réalisation de projet est l'ensemble des activités visant à organiser le bon déroulement du projet et à en atteindre les objectifs en temps selon les objectifs visés. Elle consiste à appliquer les méthodes techniques, pour ça, il est indispensable de préparer le travail.

Dans ce chapitre nous allons préparer notre projet en essayant notre outillage individuellement afin de connaître la situation des outils.

3.2. ESSAIS DU CAPTEUR DHT22

3.2.1 Code source du capteur DHT22

```
#include <DHT.h>
#define DHTPIN 3
#define DHTTYPE DHT22 // DHT 22 (AM2302)
DHT dht(DHTPIN, DHTTYPE);
void setup() {
  Serial.begin(9600);
  Serial.println(F("DHTxx test!"));
  dht.begin();
}
void loop() {
  delay(2000);
  float h = dht.readHumidity();
  float t = dht.readTemperature();
  float f = dht.readTemperature(true);
  if (isnan(h) || isnan(t) || isnan(f)) {
    Serial.println(F("Failed to read from DHT sensor!"));
    return;
  }
  float hif = dht.computeHeatIndex(f, h);
  float hic = dht.computeHeatIndex(t, h, false);
  Serial.print(F("Humidity: "));
  Serial.print(h);
  Serial.print(F("% Temperature: "));
  Serial.print(t);
  Serial.print(F("°C "));
  Serial.print(f);
  Serial.print(F("°F Heat index: "));
  Serial.print(hic);
  Serial.print(F("°C "));
  Serial.print(hif);
  Serial.println(F("°F"));
}
```

3.2.2 Simulation du capteur DHT22

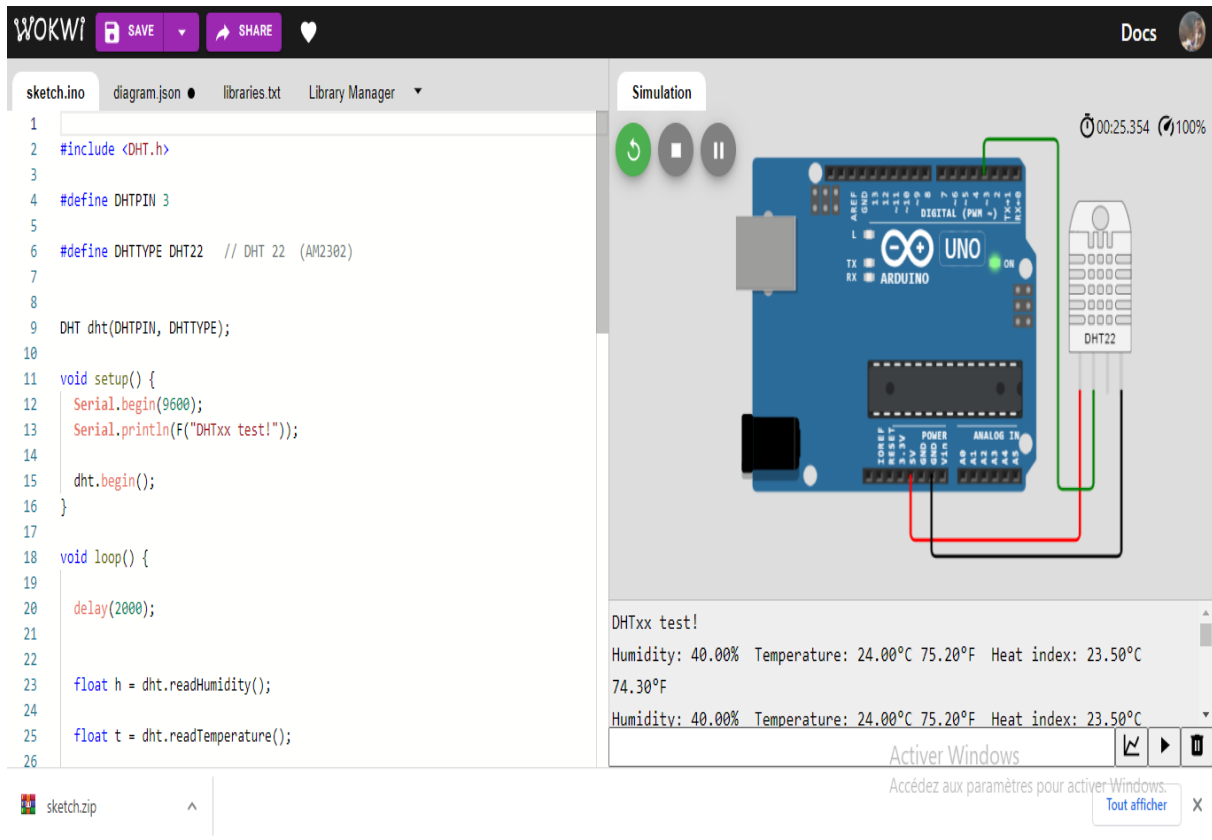


Figure 3-1 Simulation du capteur DHT22

3.2.3 Câblage du capteur DHT22 avec la carte Arduino

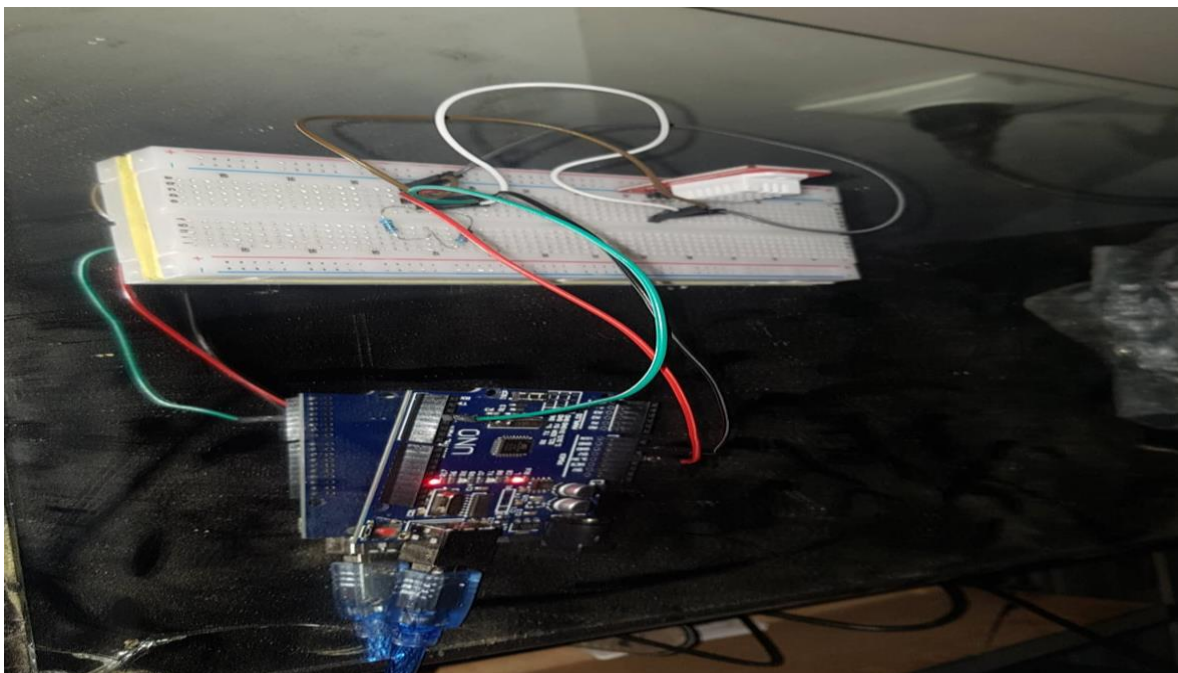


Figure 3-2 Câblage du capteur DHT22 avec la carte Arduino

3.2.4 Résultats obtenues lors de l'essai du capteur DHT22

The screenshot shows the Arduino IDE interface. The left pane displays the code for a DHT22 sensor connected to an Arduino Uno. The code includes the DHT library, defines the pin and sensor type, and sets up a serial connection at 9600 baud. The main loop reads humidity, temperature, and heat index data, printing them to the serial monitor with a 2000ms delay.

The right pane shows the serial monitor output, displaying a series of readings from the DHT22 sensor. Each line shows the time, humidity percentage, temperature in Celsius and Fahrenheit, and the heat index in Celsius and Fahrenheit.

```

23:33:25.650 -> Humidity: 83.20% Temperature: 28.40°C 83.12°F Heat index: 33.71°C 92.67°F
23:33:27.664 -> Humidity: 83.70% Temperature: 27.90°C 82.22°F Heat index: 32.41°C 90.34°F
23:33:29.714 -> Humidity: 85.50% Temperature: 28.40°C 83.12°F Heat index: 34.17°C 93.50°F
23:33:31.700 -> Humidity: 86.30% Temperature: 28.10°C 82.58°F Heat index: 33.47°C 92.24°F
23:33:33.744 -> Humidity: 77.90% Temperature: 28.30°C 82.94°F Heat index: 32.50°C 90.51°F
23:33:35.751 -> Humidity: 80.00% Temperature: 28.40°C 83.12°F Heat index: 33.12°C 91.62°F
23:33:37.808 -> Humidity: 80.70% Temperature: 28.40°C 83.12°F Heat index: 33.25°C 91.85°F
23:33:39.812 -> Humidity: 82.80% Temperature: 28.40°C 83.12°F Heat index: 33.63°C 92.54°F
23:33:41.821 -> Humidity: 81.80% Temperature: 28.30°C 82.94°F Heat index: 33.18°C 91.72°F
23:33:43.858 -> Humidity: 82.70% Temperature: 28.10°C 82.58°F Heat index: 32.79°C 91.02°F
23:33:45.862 -> Humidity: 81.40% Temperature: 28.10°C 82.58°F Heat index: 32.58°C 90.64°F
23:33:47.925 -> Humidity: 81.20% Temperature: 28.30°C 82.94°F Heat index: 33.07°C 91.53°F
23:33:49.936 -> Humidity: 81.00% Temperature: 28.50°C 83.30°F Heat index: 33.57°C 92.43°F
23:33:51.963 -> Humidity: 76.20% Temperature: 28.10°C 82.58°F Heat index: 31.76°C 89.16°F
23:33:53.968 -> Humidity: 76.10% Temperature: 28.90°C 84.02°F Heat index: 33.66°C 92.60°F
23:33:55.987 -> Humidity: 75.00% Temperature: 28.30°C 82.94°F Heat index: 32.03°C 89.65°F
23:33:58.025 -> Humidity: 75.70% Temperature: 28.30°C 82.94°F Heat index: 32.14°C 89.86°F
23:34:00.043 -> Humidity: 76.20% Temperature: 28.50°C 83.30°F Heat index: 32.70°C 90.86°F
23:34:02.054 -> Humidity: 75.90% Temperature: 27.90°C 82.22°F Heat index: 31.26°C 88.27°F
23:34:04.107 -> Humidity: 75.80% Temperature: 28.30°C 82.94°F Heat index: 32.16°C 89.89°F
23:34:06.123 -> Humidity: 75.90% Temperature: 28.00°C 82.40°F Heat index: 31.49°C 88.67°F
23:34:08.161 -> Humidity: 76.00% Temperature: 28.30°C 82.94°F Heat index: 32.19°C 89.94°F
23:34:10.177 -> Humidity: 75.80% Temperature: 28.30°C 82.94°F Heat index: 32.16°C 89.89°F
  
```

At the bottom of the IDE, a status bar indicates memory usage: "Le croquis utilise 6728 octets (20%) de l'espace de stockage de programmes. Le maximum est de 32256 octets. Les variables globales utilisent 219 octets (10%) de mémoire dynamique, ce qui laisse 1829 octets pour les variables locales. Le maximum est de 2048 octets."

Figure 3-3 Résultats obtenues lors de l'essai du capteur DHT22

3.3. ESSAIS DU CAPTEUR DE GAZ MQ 135

3.3.1 Code source du capteur de GAZ MQ 135

```

void setup() {
  pinMode (A5,INPUT);
  Serial.begin(9600);
}

void loop() {
  int vc = analogRead (A5);
  Serial.print ("la constriation de CO2 est : ");
  Serial.print ( vc, DEC);
  Serial.println(" PPM");
  delay (1000);
}
  
```

3.3.2 Simulation du capteur de GAZ MQ 135

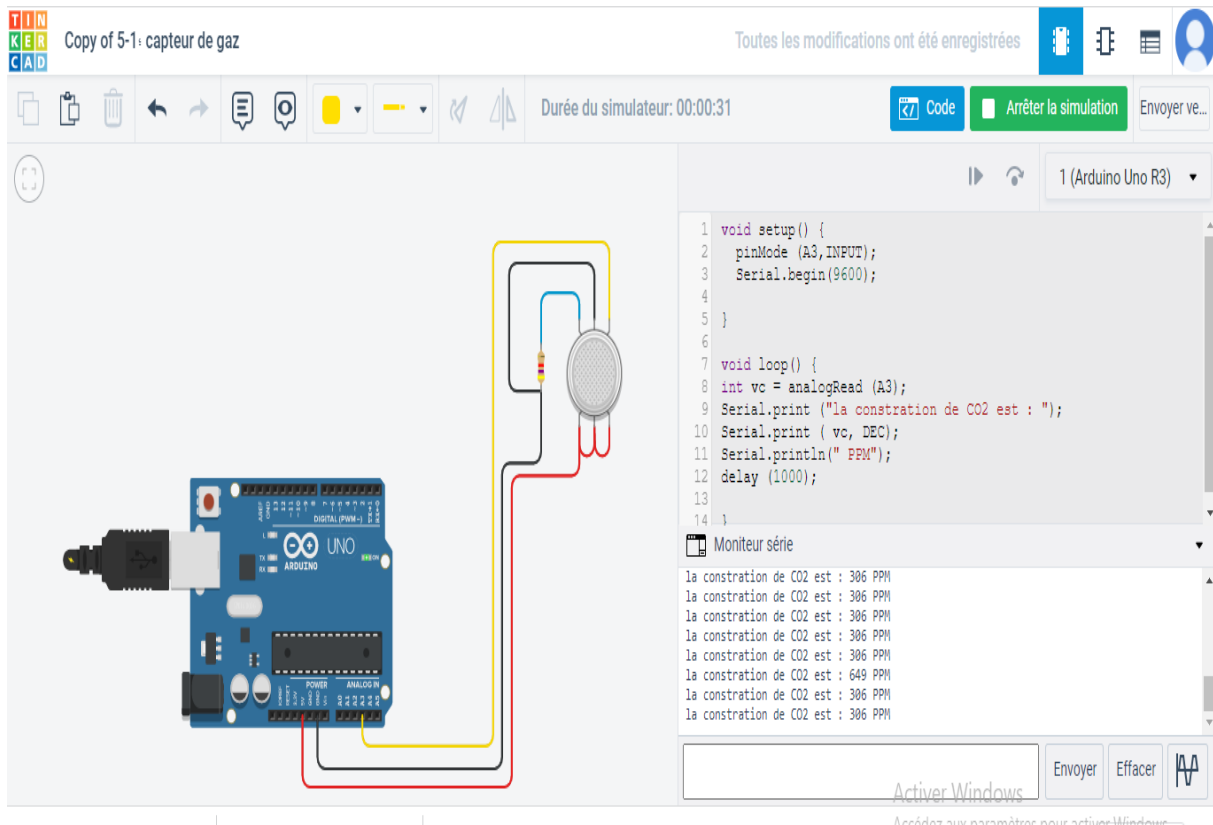


Figure 3-4 Simulation du capteur de GAZ MQ 135

3.3.3 Câblage du capteur de GAZ MQ 135 avec la carte Arduino

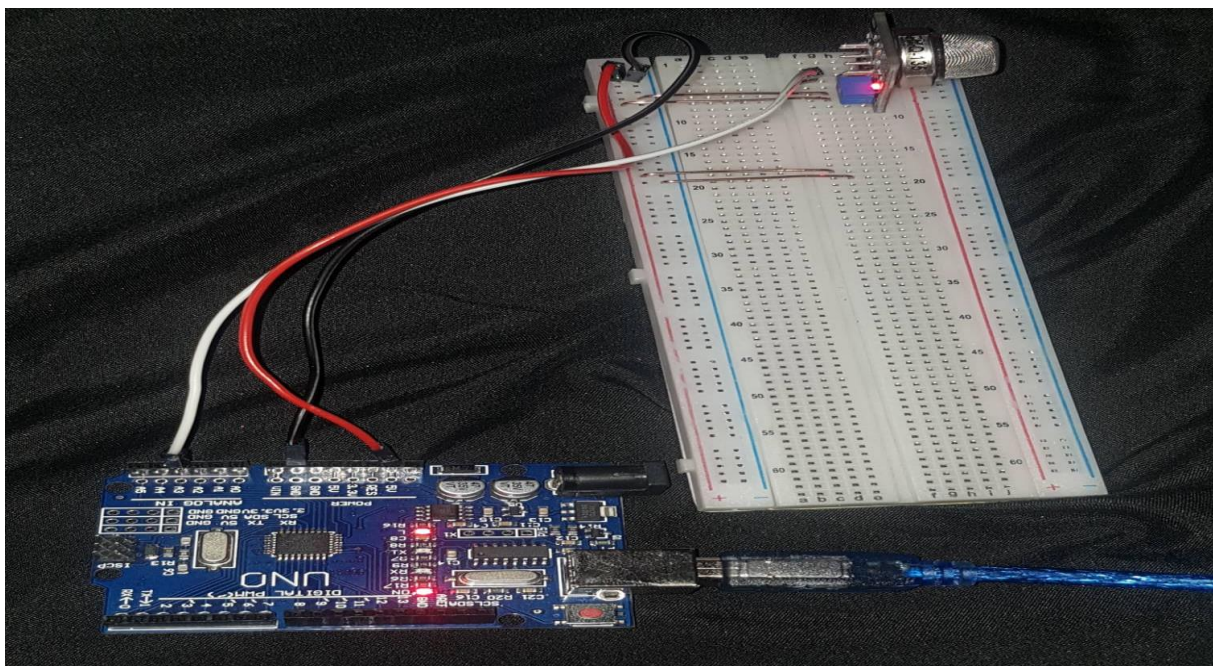


Figure 3-5 Câblage du capteur de GAZ MQ 135 avec la carte Arduino

3.3.4 Résultats obtenues lors de l'essai du capteur de GAZ MQ 135

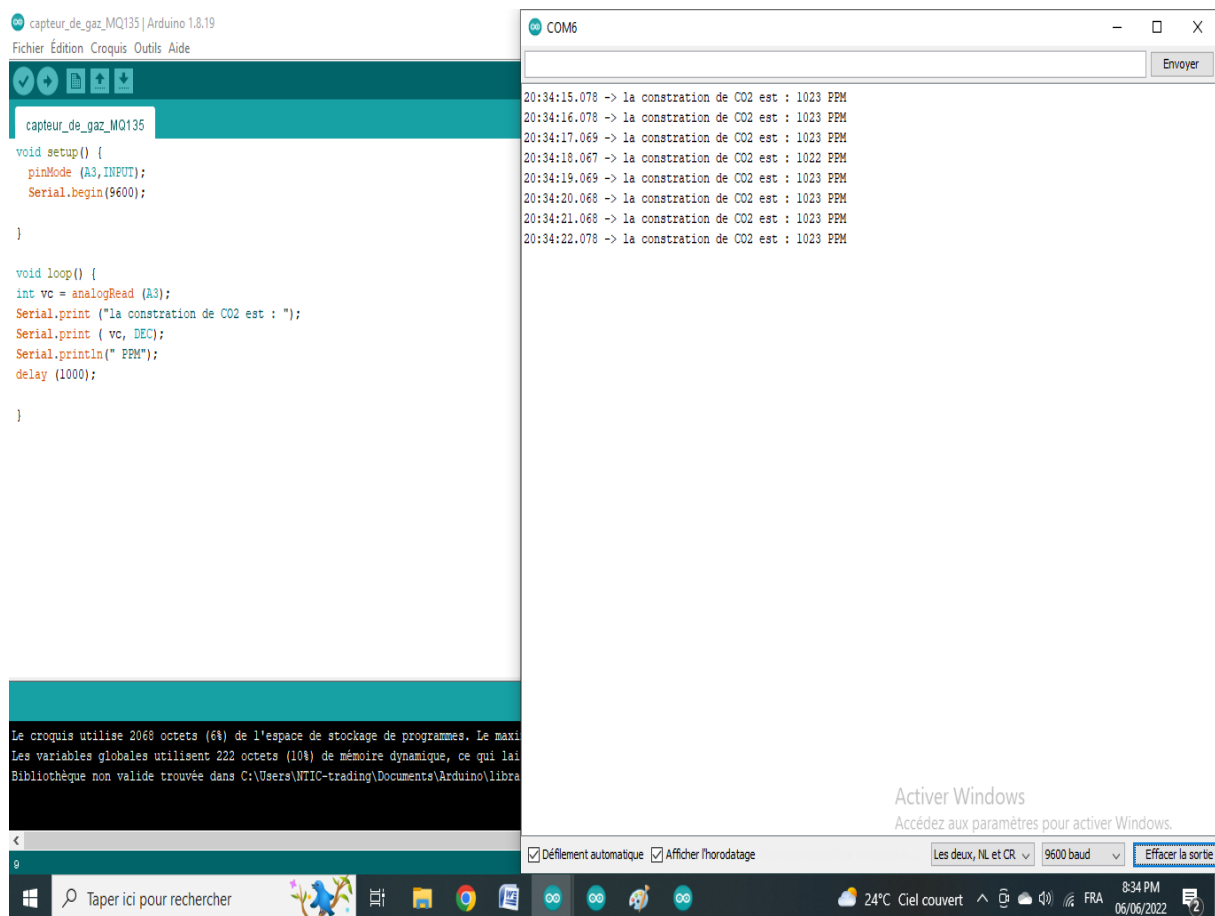


Figure 3-6 Résultats obtenues lors de l'essai du capteur de GAZ MQ 135

3.4. ESSAIS DU CAPTEUR D'HUMIDITE DE SOL

3.4.1 Code source du capteur d'Humidité de Sol

```

void setup() {
  // put your setup code here, to run once:
  pinMode (6,OUTPUT);
  pinMode (A2, INPUT);
  Serial.begin (9600);
}

void loop() {
  // put your main code here, to run repeatedly:
  int val=analogRead (A2);
  int vsl=map(val,0,50,0,255);
  analogWrite (6,vsl);
  Serial.println (vsl);
  delay (100);
}

```

3.4.2 Simulation du capteur d'Humidité de Sol

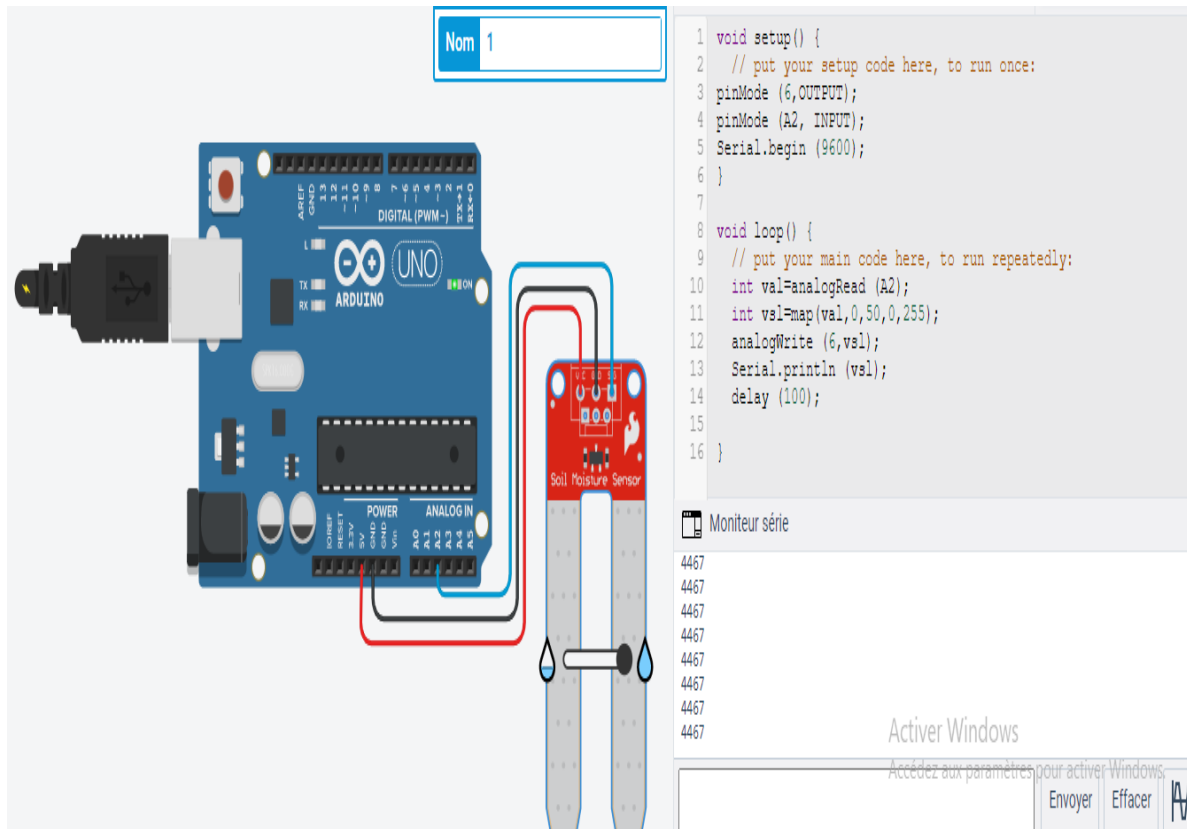


Figure 3-7 Simulation du capteur d'Humidité de Sol

3.4.3 Câblage du capteur d'Humidité de Sol avec la carte Arduino

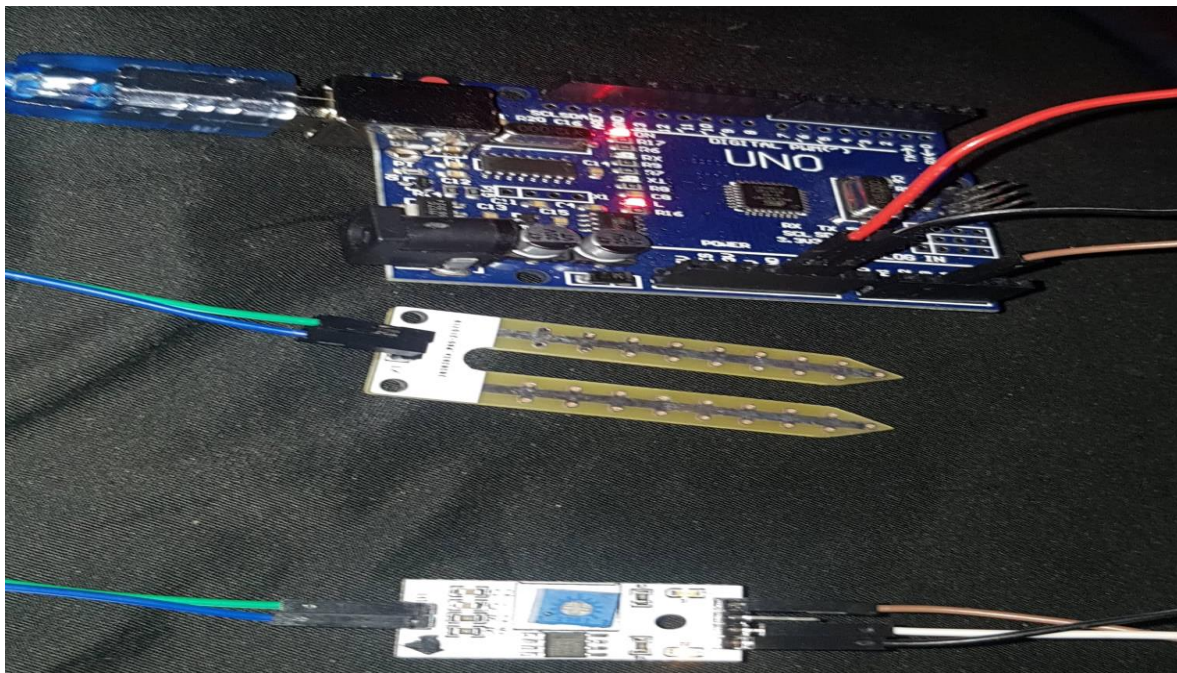


Figure 3-8 Câblage du capteur d'Humidité de Sol avec la carte Arduino

3.4.4 Résultats obtenues lors de l'essai du capteur d'Humidité de Sol

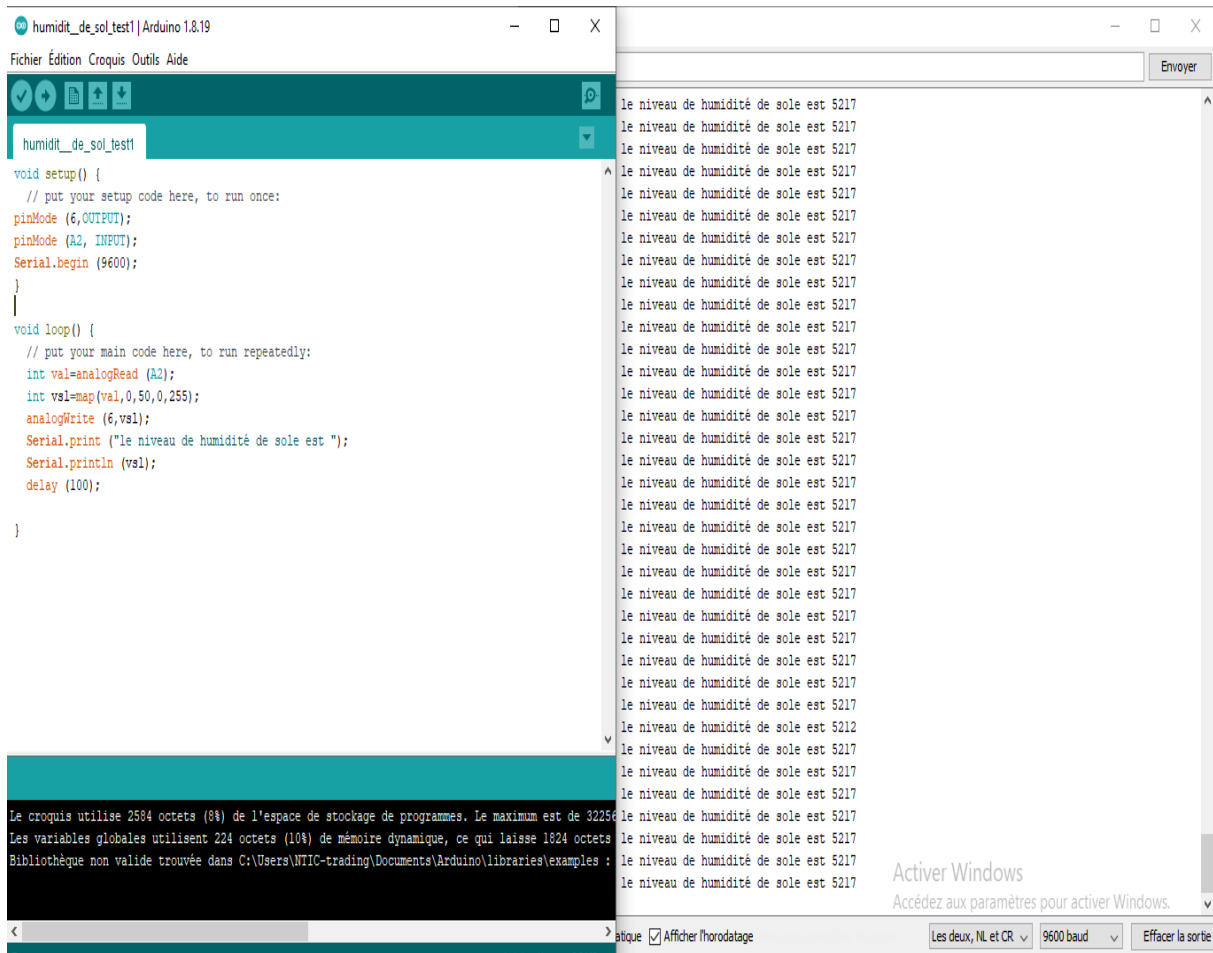


Figure 3-9 Résultats obtenues lors de l'essai du capteur d'Humidité de Sol

3.5. ESSAIS DU CAPTEUR DE LUMIERE LDR

3.5.1 Code source du capteur de lumière LDR

```

#define LDR A1
#define pompe 9
//define comande 4
int vallum;
void setup() {
  Serial.begin(9600);
  pinMode(pompe, OUTPUT);
  pinMode(LDR, INPUT);
  pinMode (4,INPUT);
}
void loop() {
  vallum = analogRead(LDR);
  Serial.println(vallum);
}

```

```

if ((vallum > 600) && (digitalRead(4) == HIGH))
{
  digitalWrite(pompe, HIGH);
  Serial.println("arrosage démarre");
  Serial.println("la pompe marche");
}

```

3.5.2 Simulation du capteur de lumière LDR

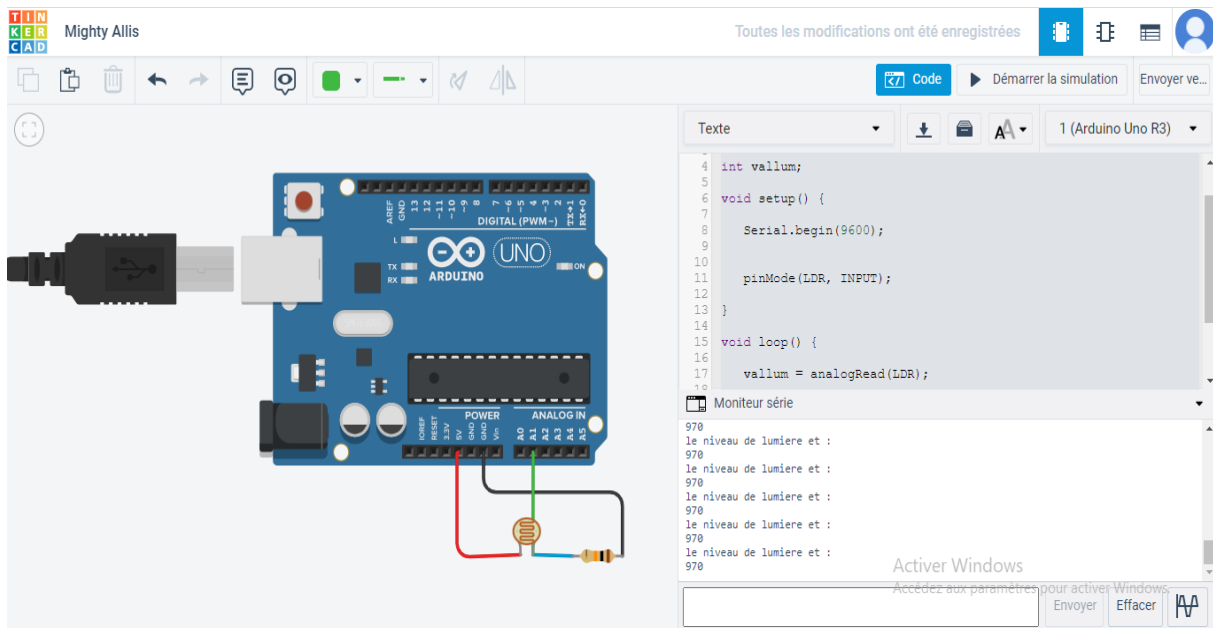


Figure 3-10 Simulation du capteur de lumière LDR

3.5.3 Câblage du capteur de lumière LDR avec la carte Arduino

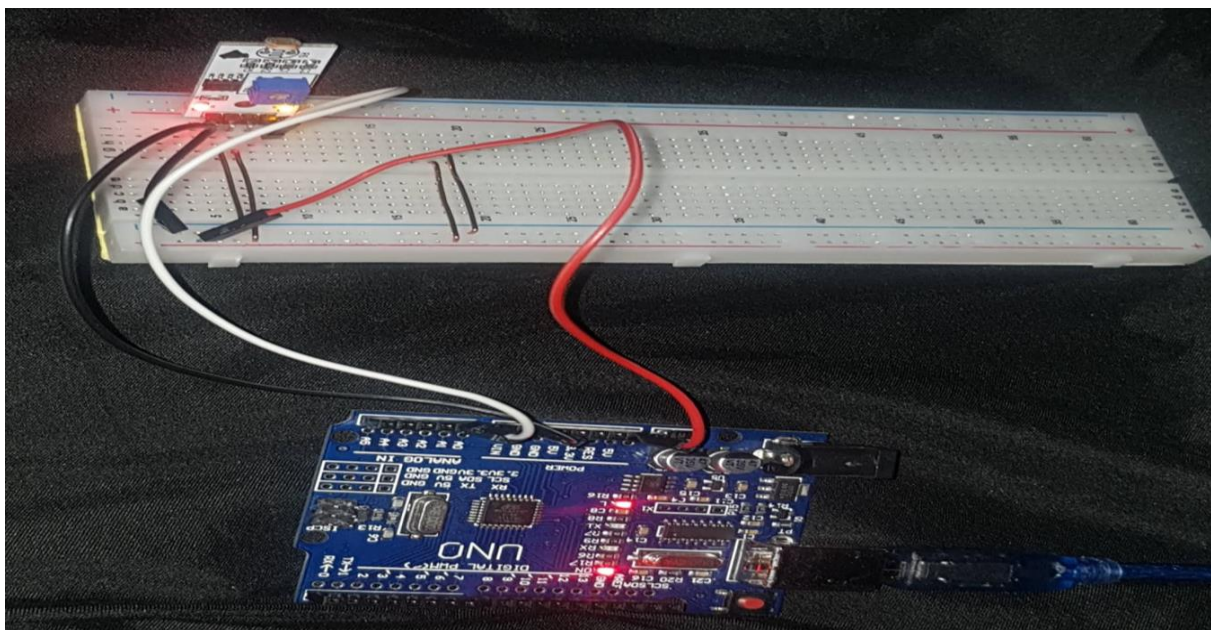


Figure 3-11 Câblage du capteur de lumière LDR avec la carte Arduino

3.5.4 Résultats obtenues lors de l'essai du capteur de lumière LDR

```

ldr_simple
#define LDR A1
int vallum;

void setup() {
  Serial.begin(9600);
  pinMode(LDR, INPUT);
}

void loop() {
  vallum = analogRead(LDR);

  Serial.println("le niveau de lumiere et :");
  Serial.println(vallum);

  delay(2000);
}

```

```

20:06:25.091 -> le niveau de lumiere et :
20:06:25.091 -> 773
20:06:27.086 -> le niveau de lumiere et :
20:06:27.086 -> 772
20:06:29.094 -> le niveau de lumiere et :
20:06:29.094 -> 773
20:06:31.097 -> le niveau de lumiere et :
20:06:31.097 -> 773
20:06:33.055 -> le niveau de lumiere et :
20:06:33.055 -> 772
20:06:35.059 -> le niveau de lumiere et :
20:06:35.059 -> 772
20:06:37.076 -> le niveau de lumiere et :
20:06:37.076 -> 773
20:06:39.080 -> le niveau de lumiere et :
20:06:39.080 -> 771
20:06:41.085 -> le niveau de lumiere et :
20:06:41.085 -> 772
20:06:43.095 -> le niveau de lumiere et :
20:06:43.095 -> 773
20:06:45.096 -> le niveau de lumiere et :
20:06:45.096 -> 772
20:06:47.066 -> le niveau de lumiere et :
20:06:47.066 -> 772
20:06:49.072 -> le niveau de lumiere et :
20:06:49.072 -> 772
20:06:51.088 -> le niveau de lumiere et :
20:06:51.088 -> 782
20:06:53.083 -> le niveau de lumiere et :
20:06:53.083 -> 885
20:06:55.089 -> le niveau de lumiere et :
20:06:55.089 -> 793
20:06:57.100 -> le niveau de lumiere et :
20:06:57.100 -> 790
20:06:59.103 -> le niveau de lumiere et :
20:06:59.103 -> 798

```

Figure 3-12 Résultats obtenues lors de l'essai du capteur de lumière LDR

3.6. ESSAIS DE L’AFFICHEUR LCD

3.6.1 Code source de l’afficheur LCD

```

#include <Wire.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27,16,2); // set the LCD address to 0x27 for a 16 chars and 2 line display
void setup()
{
}
void loop()
{
  lcd.init();           // initialize the lcd

  // Print a message to the LCD.
  lcd.backlight();
  lcd.setCursor(0,0);
  lcd.print("Hello Every one");
  lcd.setCursor(1,1);
  lcd.print("i'm Alouane Med");
  delay(5000);
}

```

3.6.2 Simulation de l'afficheur LCD

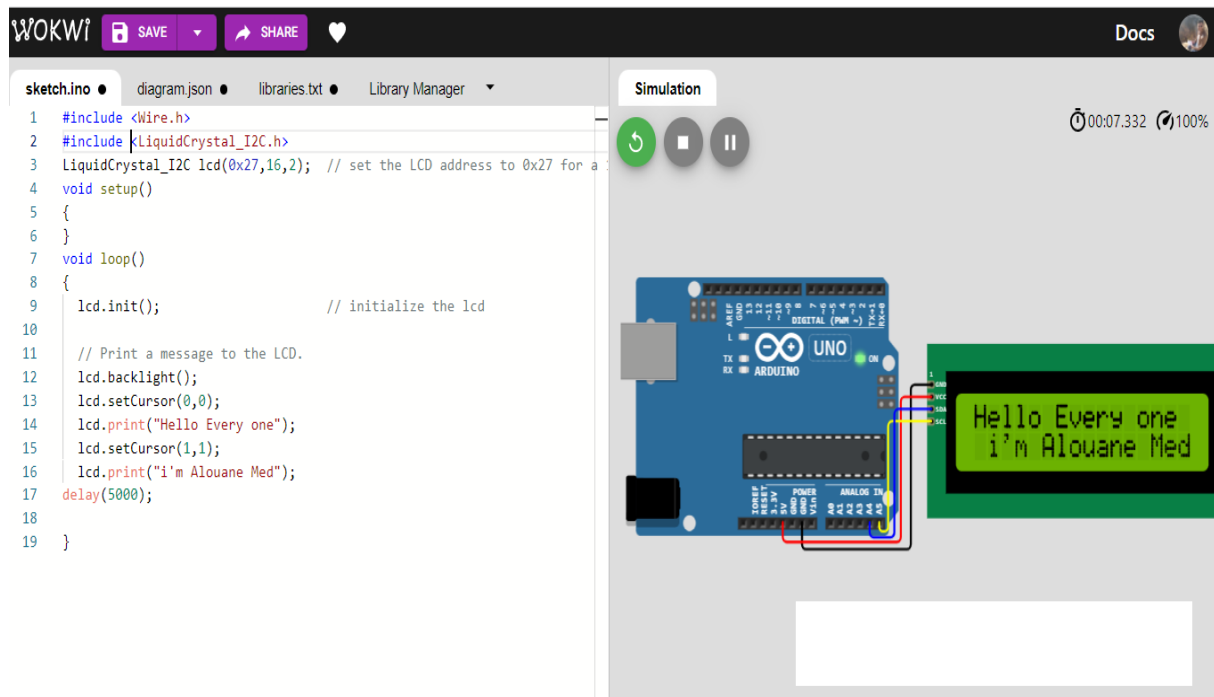


Figure 3-13 Simulation de l'afficheur LCD

3.6.3 Câblage de l'afficheur LCD avec la carte Arduino

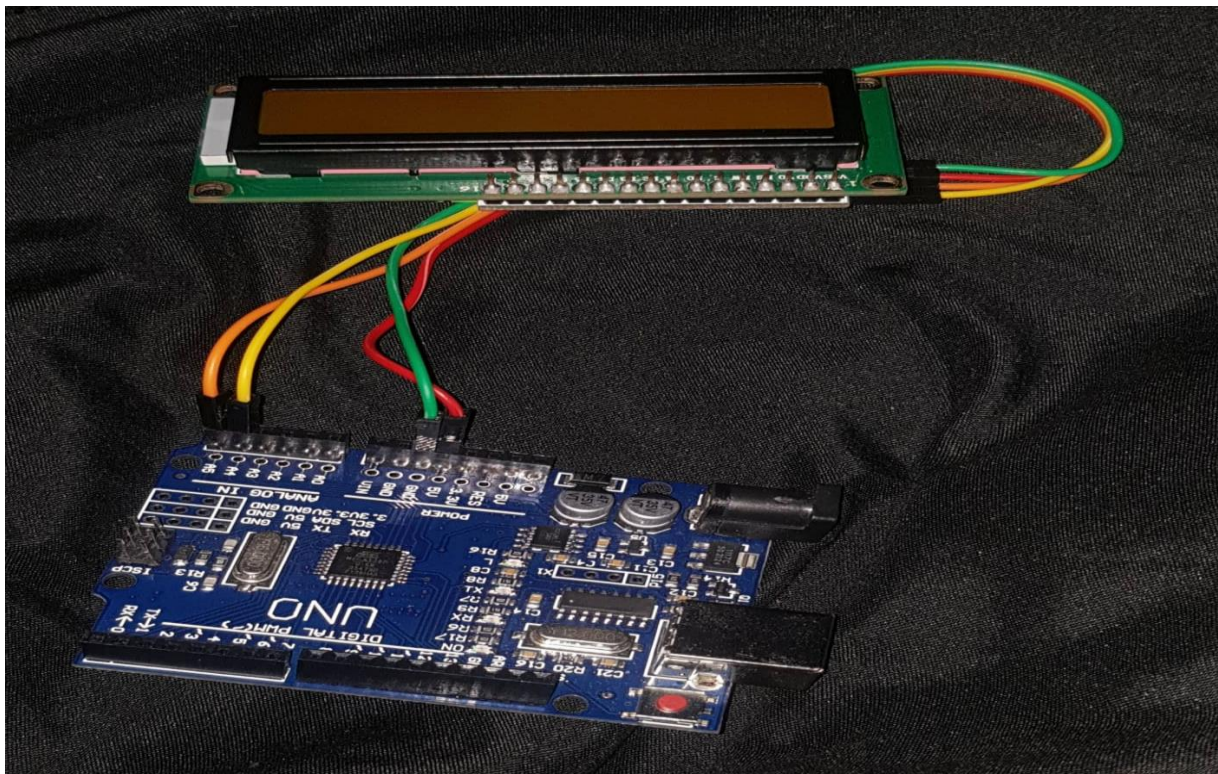


Figure 3-14 Câblage de l'afficheur LCD avec la carte Arduino

3.6.4 Résultats obtenues lors de l'essais de l'afficheur LCD

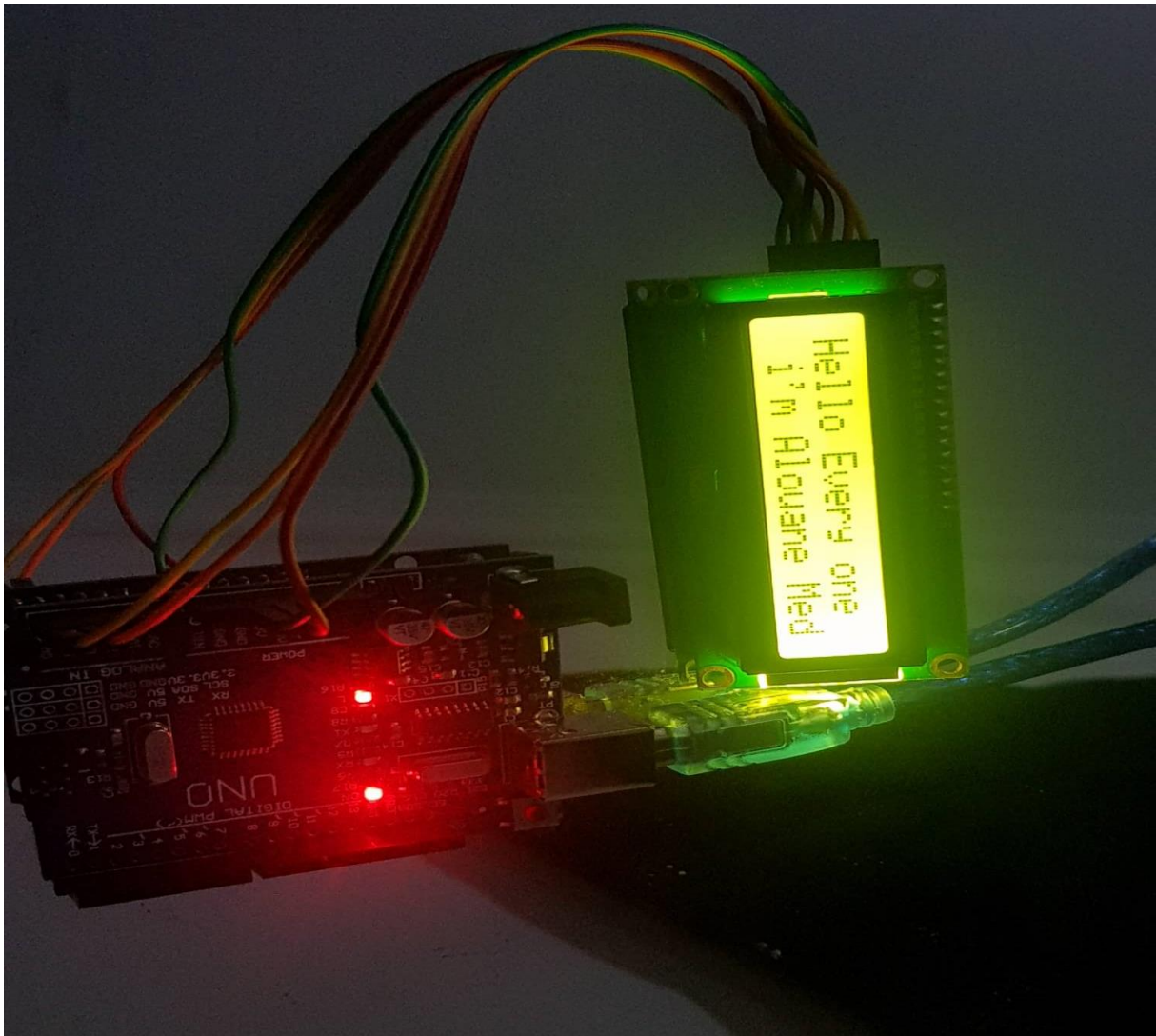


Figure 3-15 Résultats obtenues lors de l'essais de l'afficheur LCD

3.7. ESSAIS DU MODULE WIFI ESP 8266

Pour essayer le module WIFI ESP 8266, nous avons utilisé deux LED afin de les commander à travers deux boutons crée sur un smart phone à l'aide de l'application RemoteXY.

3.7.1 Code source du module WIFI ESP 8266

```
#define REMOTEXY_MODE__ESP8266_HARDSERIAL_POINT
#include <RemoteXY.h>
#define REMOTEXY_SERIAL Serial
#define REMOTEXY_SERIAL_SPEED 115200
#define REMOTEXY_WIFI_SSID "RemoteXY"
#define REMOTEXY_WIFI_PASSWORD "12345678"
#define REMOTEXY_SERVER_PORT 99
#pragma pack(push, 1)
uint8_t RemoteXY_CONF[] =
{ 255,2,0,0,0,37,0,16,31,1,2,0,16,14,22,11,1,26,31,31,
```

```

79,78,0,79,70,70,0,2,0,16,35,22,11,135,26,31,31,79,78,0,
79,70,70,0 };
struct {
  uint8_t s1; // =1 if switch ON and =0 if OFF
  uint8_t s2; // =1 if switch ON and =0 if OFF
  uint8_t connect_flag;
} RemoteXY;
#pragma pack(pop)
#define PIN_S1 5
#define PIN_S2 7
void setup()
{
  RemoteXY_Init ();
  pinMode (PIN_S1, OUTPUT);
  pinMode (PIN_S2, OUTPUT);
}
void loop()
{
  RemoteXY_Handler ();
  digitalWrite(PIN_S1, (RemoteXY.s1==0)?LOW:HIG);
  digitalWrite(PIN_S2, (RemoteXY.s2==0)?LOW:HIG);
}

```

3.7.2 Câblage du module WIFI ESP 8266 et les deux LED avec la carte Arduino

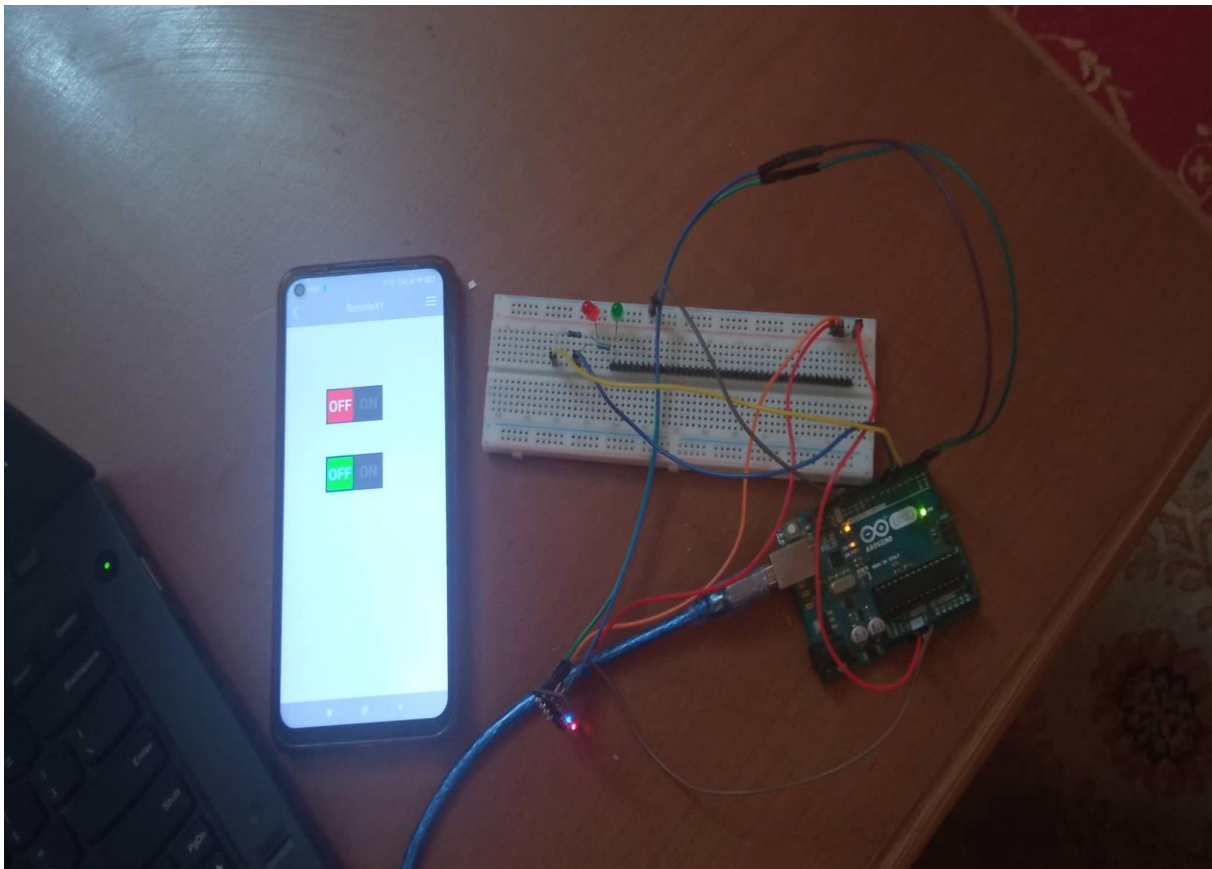
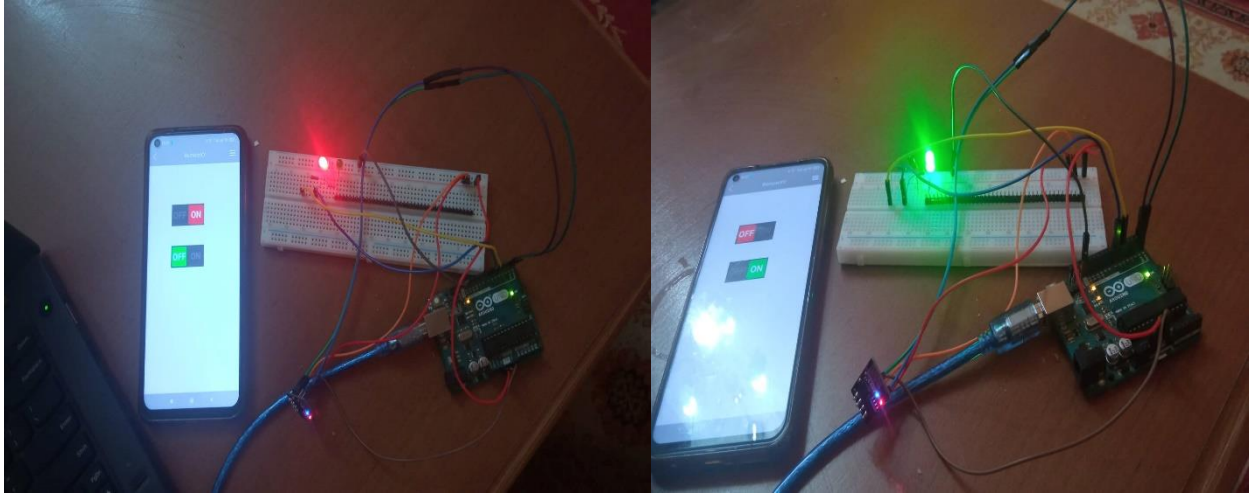
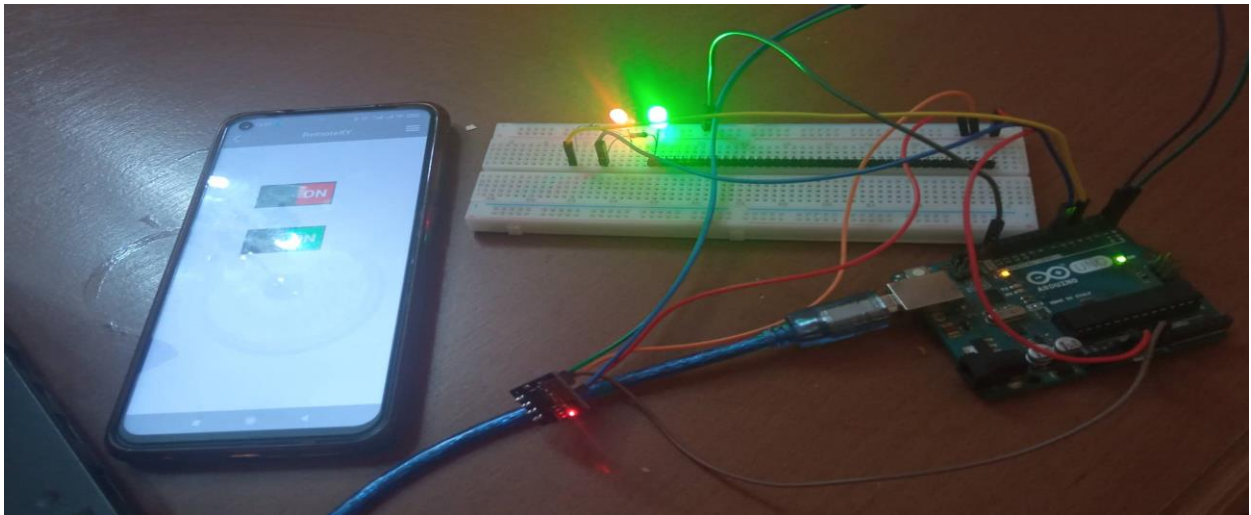


Figure 3-16 Câblage du module WIFI ESP8266 et les deux LED avec la carte Arduino

3.7.3 Résultats obtenues lors des l'essais du module WIFI ESP 8266



a) Allumage de la LED rouge par le bouton rouge b) Allumage de la LED verte par le bouton vert



c) Allumage des deux LED rouge et verte par les deux boutons

Figure 3-17 Résultats obtenues lors de l'essais du module WIFI ESP 8266

3.8. CONCLUSION

Nous avons essayé nos outils de travail et nous avons su leur état de marche, grâce à ça nous pouvons réaliser notre projet.

Dans le chapitre prochain nous parlons au détails et résultats de la réalisation de notre projet.

CHAPITRE 04 :

LA REALISATION

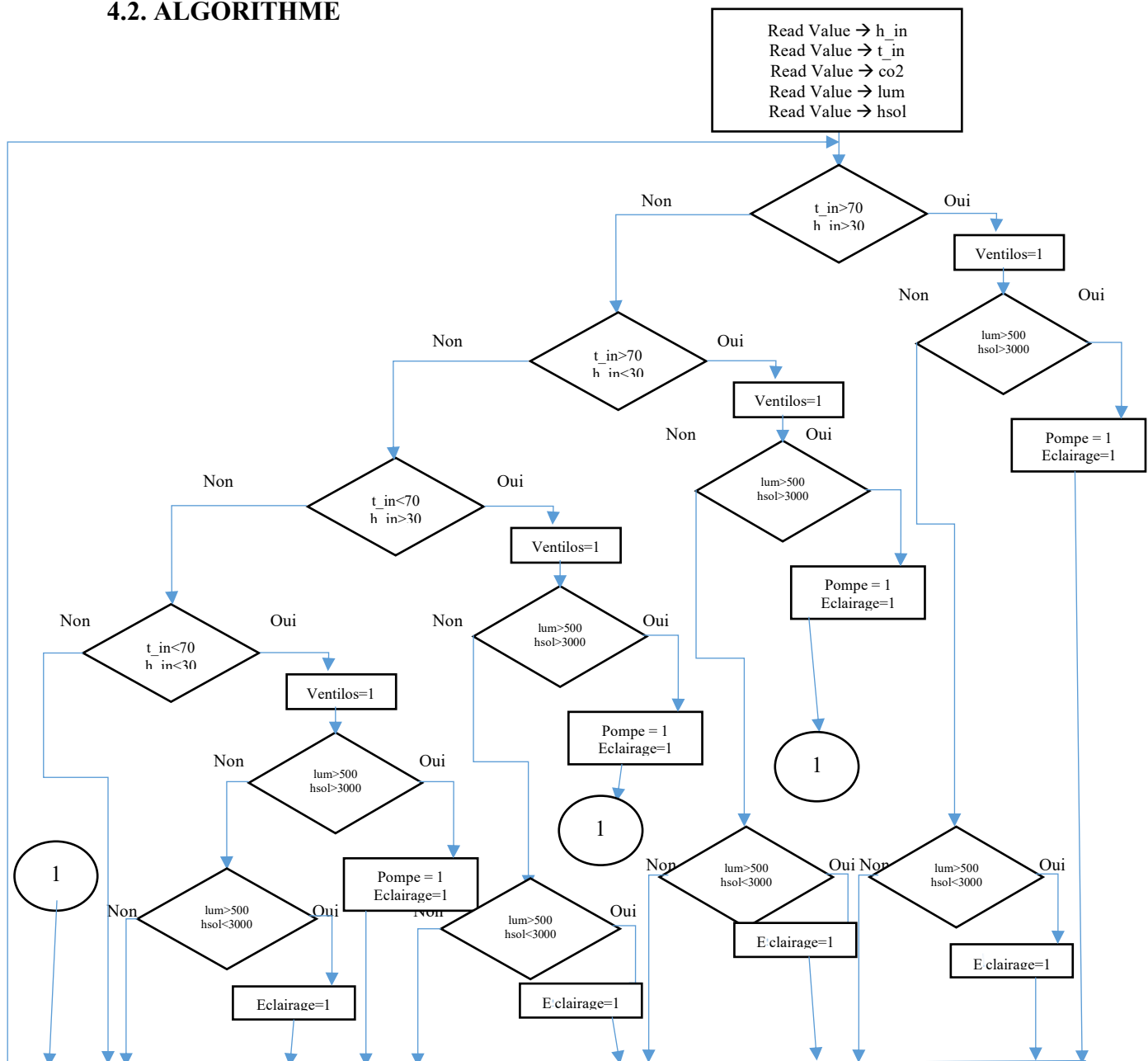
4.1. INTRODUCTION

Après de la confirmation de la bonne marche des outils de travail, ça commence la phase de la réalisation pratique.

Dans ce chapitre nous allons suivre des étapes afin de réaliser notre projet, ces étapes sont :

- Traduction du cahier des charges en Algorithme.
- Rédaction du code source pour programmer notre Microcontrôleur.
- Câblage, téléversement du code source et essais de fonctionnement du projet.

4.2. ALGORITHME



Avec **t_in** : température interne ; **h_in** : humidité interne
lum : lumière ; **hsol** : humidité de sol ; **co2** : gaz mesuré par LDR

4.3. LE CODE SOURCE DU PROJET

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27,16,2);
#include <DHT.h> //la bibliothèque
#include <DHT_U.h>
#define DHTPIN 3
#define DHTTYPE DHT22
DHT dht(DHTPIN, DHTTYPE);
#define ventilo 4
#define CO_2 A3
#define LDR A1
#define HSol A2
#define pompe 5
#define ecli 6
int ref_lum=500 ;
int ref_sol=3000 ;
int ref_Tin=30 ;
int ref_Hin=70 ;

////////////////////////////////////
// RemoteXY include library //
////////////////////////////////////

#define REMOTEXY_MODE __ESP8266_HARDSERIAL_POINT
#include <RemoteXY.h>
#define REMOTEXY_SERIAL Serial
#define REMOTEXY_SERIAL_SPEED 115200
#define REMOTEXY_WIFI_SSID "final"
#define REMOTEXY_WIFI_PASSWORD "12345678"
#define REMOTEXY_SERVER_PORT 6377
#pragma pack(push, 1)
uint8_t RemoteXY_CONF[] = // 244 bytes
{ 255,3,0,58,0,237,0,16,27,0,2,1,44,8,15,8,1,135,31,31,
  79,78,0,79,70,70,0,2,1,76,7,15,8,1,136,31,31,79,78,0,
  79,70,70,0,2,1,13,8,15,8,1,136,31,31,79,78,0,79,70,70,
  0,129,0,41,1,20,6,8,80,111,109,112,101,0,129,0,72,1,23,6,
  8,86,101,110,116,105,108,111,115,0,129,0,13,1,16,6,8,76,97,109,
  112,0,129,0,18,42,21,5,1,84,32,105,110,116,32,40,194,176,67,41,
  0,129,0,59,43,20,5,6,72,32,105,110,116,32,40,37,41,0,70,16,
  16,17,9,9,16,1,0,70,16,79,17,9,9,16,135,0,70,16,47,17,
  9,9,16,3,0,67,5,18,50,20,10,2,26,11,67,5,59,50,20,10,
  2,26,11,67,5,12,28,17,7,2,26,11,129,0,1,30,10,5,17,67,
  79,50,0,129,0,31,30,10,5,17,72,83,111,108,0,129,0,63,30,10,
  5,17,76,117,109,0,67,5,43,28,18,7,2,26,11,67,5,75,28,18,
  7,2,26,11 };

struct {
    // input variables
    uint8_t s1; // =1 if switch ON and =0 if OFF
    uint8_t s2; // =1 if switch ON and =0 if OFF
    uint8_t s3; // =1 if switch ON and =0 if OFF
    // output variables
    uint8_t led_3; // led state 0 .. 1
    uint8_t led_2; // led state 0 .. 1
    uint8_t led_1; // led state 0 .. 1
    char tint[11]; // string UTF8 end zero
    char hint[11]; // string UTF8 end zero
    char CO2[11]; // string UTF8 end zero
    char sol[11]; // string UTF8 end zero
    char ecl[11]; // string UTF8 end zero
}
```

```

    // other variable
    uint8_t connect_flag; // =1 if wire connected, else =0

} RemoteXY;
#pragma pack(pop)

////////////////////
//      END RemoteXY include      //
////////////////////
#define PIN_S1 5
#define PIN_S2 4
#define PIN_S3 6
void setup()
{
    RemoteXY_Init ();
    pinMode (PIN_S1, OUTPUT);
    pinMode (PIN_S2, OUTPUT);
    pinMode (PIN_S3, OUTPUT);
    // TODO you setup code
    pinMode (ventilo,OUTPUT);
    pinMode (pompe, OUTPUT);
    pinMode (ecli,OUTPUT);
    pinMode (CO_2,INPUT);
    pinMode (HSol, INPUT);
    pinMode (LDR, INPUT);
    lcd.init();
    lcd.backlight();
    lcd.setCursor(0,0);
    lcd.print("Bienvenue");
    dht.begin();
}
void loop()
{ lcd.init();//initialisation de LCD
  lcd.clear();
  RemoteXY_Handler ();
  digitalWrite(PIN_S1, (RemoteXY.s1==0)?LOW:HIGH);
  digitalWrite(PIN_S2, (RemoteXY.s2==0)?LOW:HIGH);
  digitalWrite(PIN_S3, (RemoteXY.s3==0)?LOW:HIGH);
  int vsl=analogRead (HSol);
  vsl=map(vsl,0,50,0,255);
  int vallum = analogRead (LDR);
  int vco = analogRead (CO_2);
  float h_in = dht.readHumidity();//humidité
  float t_in = dht.readTemperature();//température

  lcd.setCursor(0,0);
  lcd.print ("Temp = ");
  lcd.setCursor(8,0);
  lcd.print (t_in);
  lcd.setCursor(13,0);
  lcd.print (" C ");

  lcd.setCursor(0,1);
  lcd.print ("Hum = ");
  lcd.setCursor(8,1);
  lcd.print (h_in,DEC);
  lcd.setCursor(13,1);
  lcd.print (" % ");

  if ((h_in > ref_Hin) && (t_in > ref_Tin))

```

```

{  digitalWrite (ventilo,HIGH);
  if ((vallum>ref_lum) && (vsl>ref_sol))
  {
    digitalWrite(pompe, HIGH);
    digitalWrite(ecli, HIGH);
  }
  else if ((vallum>ref_lum) && (vsl<ref_sol))
  {digitalWrite(ecli, HIGH);
  }
  else if ((h_in >ref_Hin) && (t_in <ref_Tin))
  {digitalWrite (ventilo,HIGH);
  if ((vallum>ref_lum) && (vsl>ref_sol))
  { digitalWrite(pompe, HIGH);
    digitalWrite(ecli, HIGH);
  }
  else if ((vallum>ref_lum) && (vsl<ref_sol))
  {digitalWrite(ecli, HIGH);
  }
  else if ((h_in<ref_Hin) && (t_in >ref_Tin))
  { digitalWrite (ventilo,HIGH);
    if ((vallum>ref_lum) && (vsl>ref_sol))
    { digitalWrite(pompe, HIGH);
      digitalWrite(ecli, HIGH);
    }
    else if ((vallum>ref_lum) && (vsl<ref_sol))
    {digitalWrite(ecli, HIGH);
    }
  }
  else if ((h_in<ref_Hin) && (t_in <ref_Tin))
  {
    if ((vallum>ref_lum) && (vsl>ref_sol))
    { digitalWrite(pompe, HIGH);
      digitalWrite(ecli, HIGH);
    }
    else if ((vallum>ref_lum) && (vsl<ref_sol))
    {digitalWrite(ecli, HIGH);
    }
  }
  // use the RemoteXY structure for data transfer
  dtostrf(h_in,0,1,RemoteXY.hint);
  dtostrf(t_in,0,1,RemoteXY.tint);
  dtostrf(vco,0,1,RemoteXY.CO2);
  dtostrf(vallum,0,1,RemoteXY.ecl);
  dtostrf(vsl,0,1,RemoteXY.sol);
  if (digitalRead(5) == HIGH) // if pin 5 enjoyed a high level voltage
    RemoteXY.led_1 = 1; // then turn on red light
  else // else
    RemoteXY.led_1 = 0; // turn off red
  if (digitalRead(4) == HIGH) // if pin 4 enjoyed a high level voltage
    RemoteXY.led_2 = 1; // then turn on red light
  else // else
    RemoteXY.led_2 = 0; // turn off red
  if (digitalRead(6) == HIGH) // if pin 6 enjoyed a high level voltage
    RemoteXY.led_3 = 1; // then turn on red light
  else // else
    RemoteXY.led_3 = 0; // turn off red
  delay (1200);
}

```

4.4. PHOTOS PRISE LORS DES ESSAIS DU PROJET

Les figures suivantes représentent quelques essais des mesures et de la mise en marche et arrêt des LED (les LED remplacent les actionneurs d'où le rouge remplace la lampe, le jaune remplace la pompe et le vert remplace le ventilos).

- Affichage des mesures de la température et l'Humidité interne sur l'afficheur LCD (Figure 4-1).

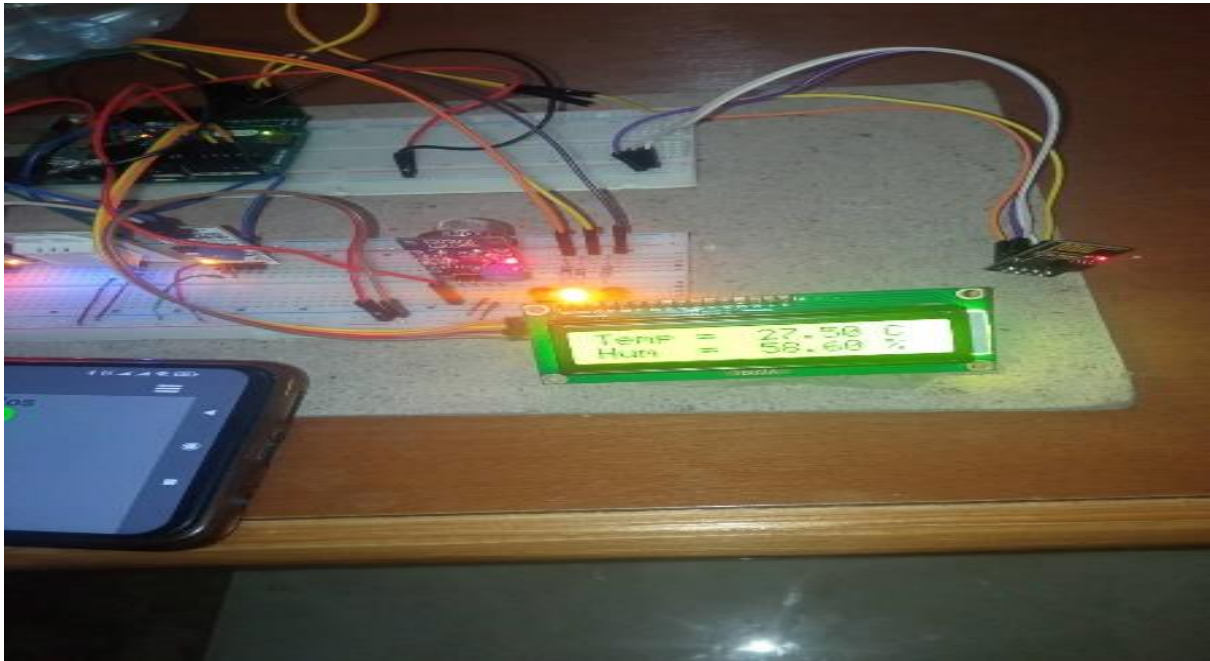


Figure 4-1 Affichage de la Température et Humidité sur l'afficheur LCD

- Affichage de la Température, Humidité, Humidité de Sol, CO₂ et le Taux de la Lumière sur le Smart Phone (Figure 4-2).



Figure 4-2 Affichage des grandeurs mesurés sur le Smart Phone

- Résultat du cas d'une Sol Humide (Figure 4-3).

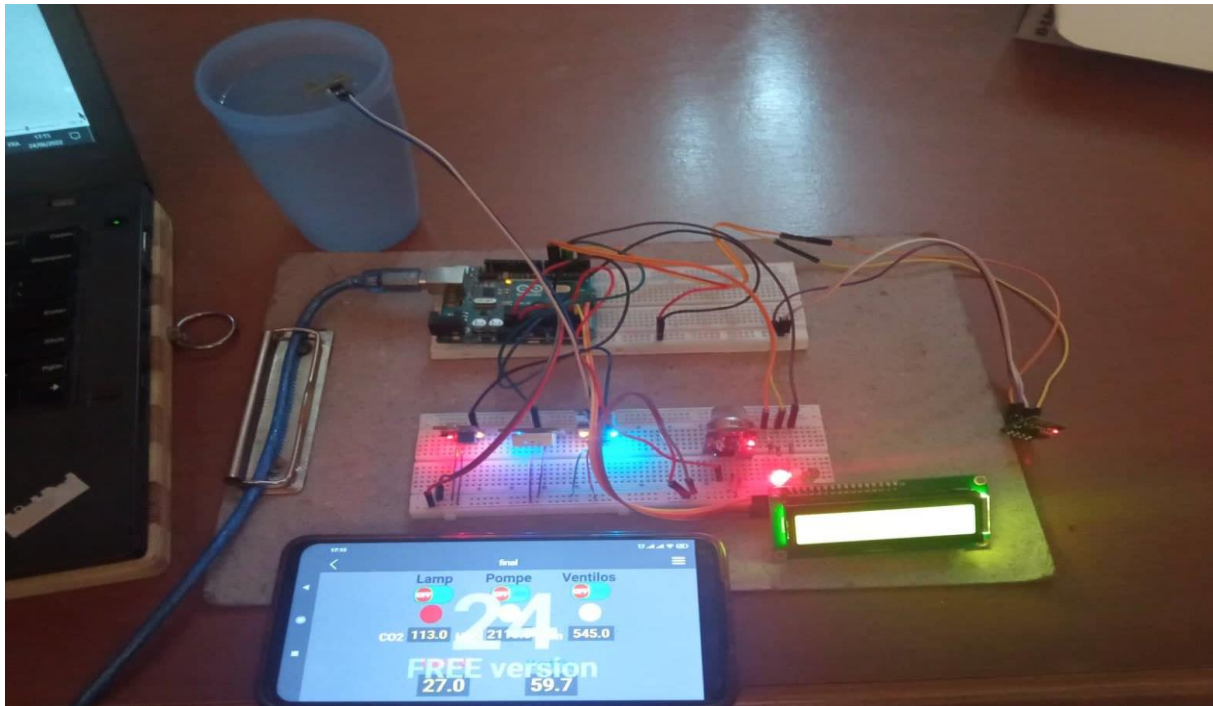


Figure 4-3 Pompe en arrêt au cas de $vsl < ref_sol$

- Résultat au cas d'un Sol humide avec ordre de marche de la pompe par le Smart Phone (Figure 4-4).

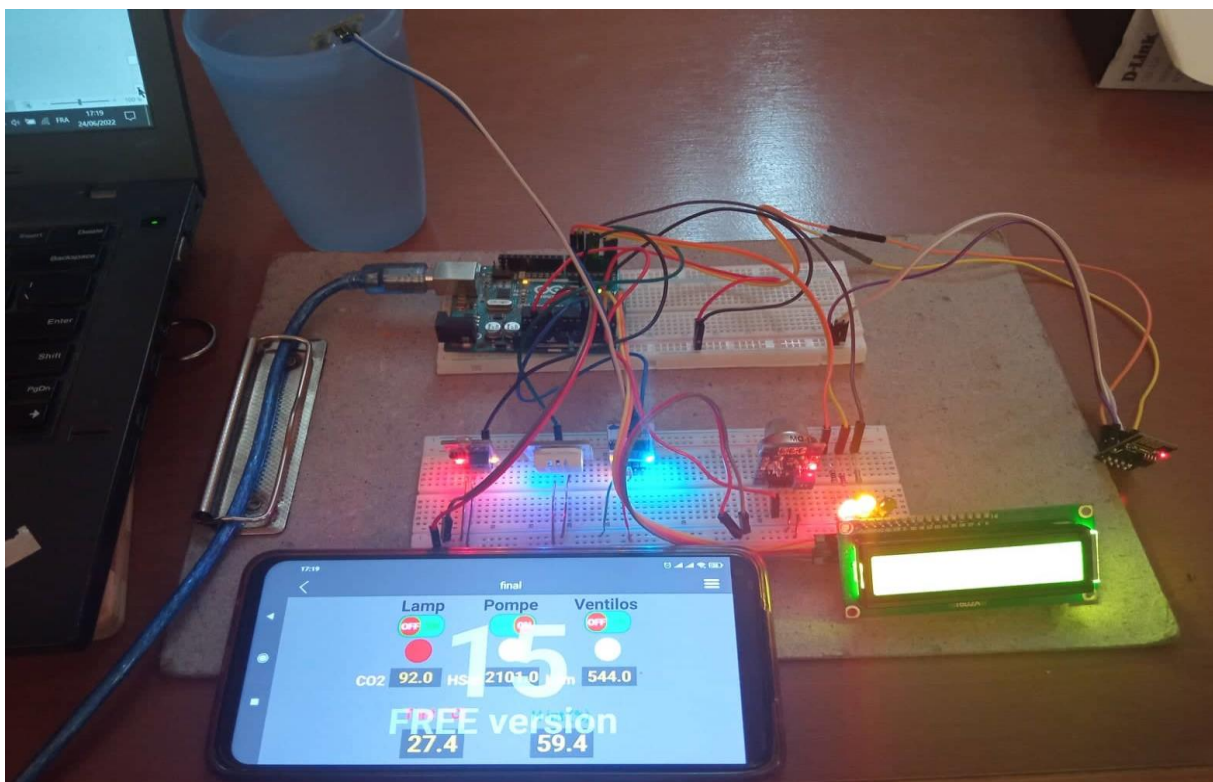


Figure 4-4 Mise en marche de la Pompe par le Smart phone

- Résultat dans le cas où le taux de la lumière inférieur à la valeur de référence (Figure 4-5).

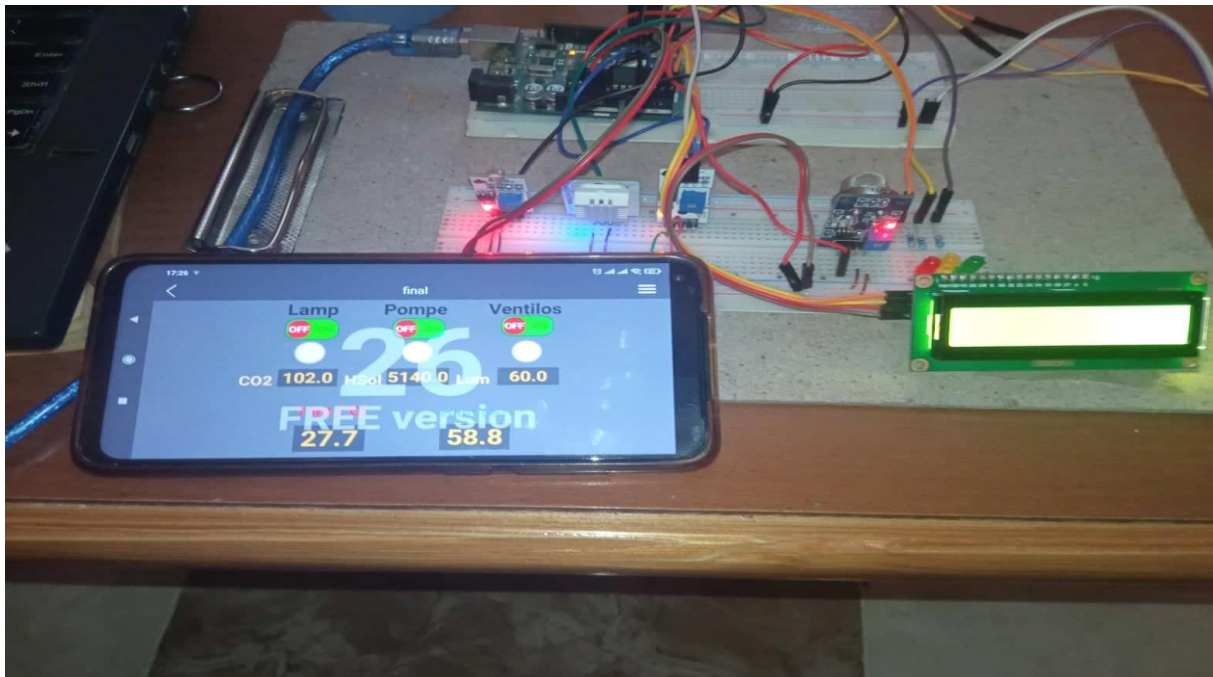


Figure 4-5 Lampe et Pompe en arrêt au cas de $v_{lum} < ref_lum$

- Résultat dans le cas où le taux de la lumière et inférieur à la valeur de référence et Sol humide (Figure 4-6).



Figure 4-6 Lampe et Pompe en arrêt au cas de $v_{lum} < ref_lum$ et $v_{sl} < ref_sol$

- Résultat dans le cas où la Température ou l'Humidité interne élevée à la valeur de référence (Figure 4-7).

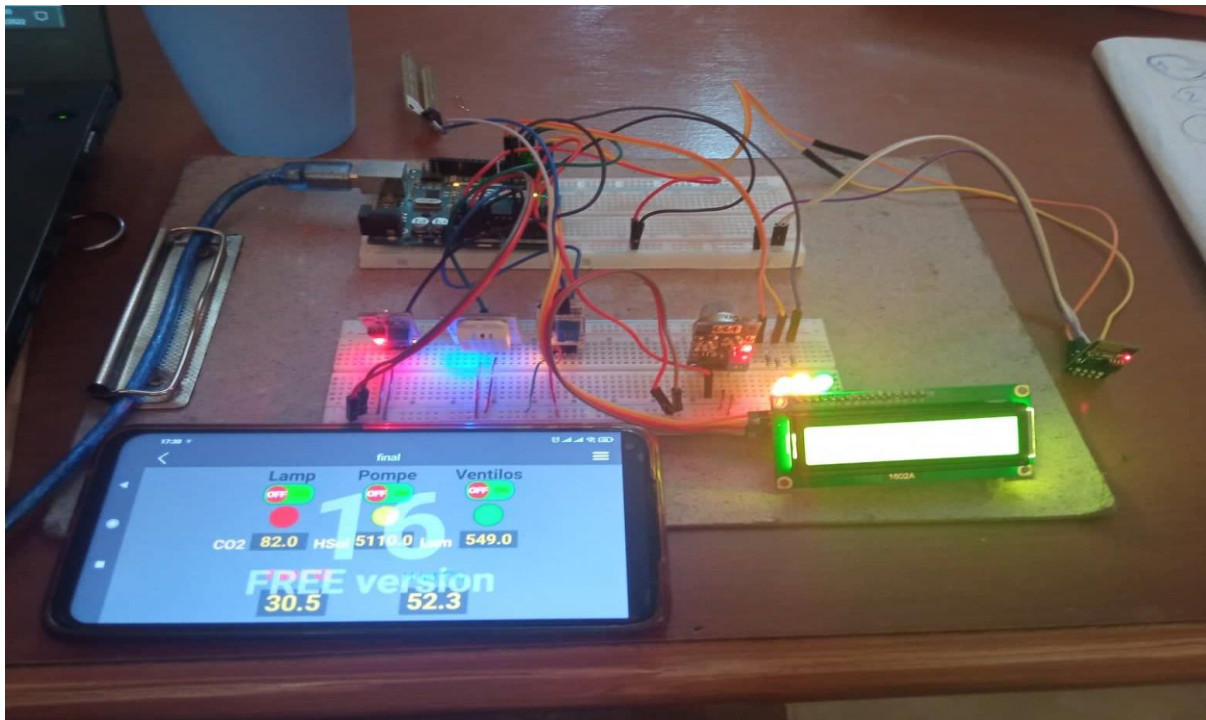


Figure 4-7 Ventilos en Marche au cas de $T_{in} > ref_T_{in}$ ou $H_{in} > ref_H_{in}$

- Résultat de la mise en marche du ventilateur à partir du Smart Phone quand la Température et l'Humidité interne sont inférieure à valeur de référence (Figure 4-8).



Figure 4-8 Mise en marche du Ventilos par le Smart phone

- Résultat de la mise en marche de la Lampe dans le cas où le taux de la lumière est supérieur à la valeur de référence (Figure 4-9).

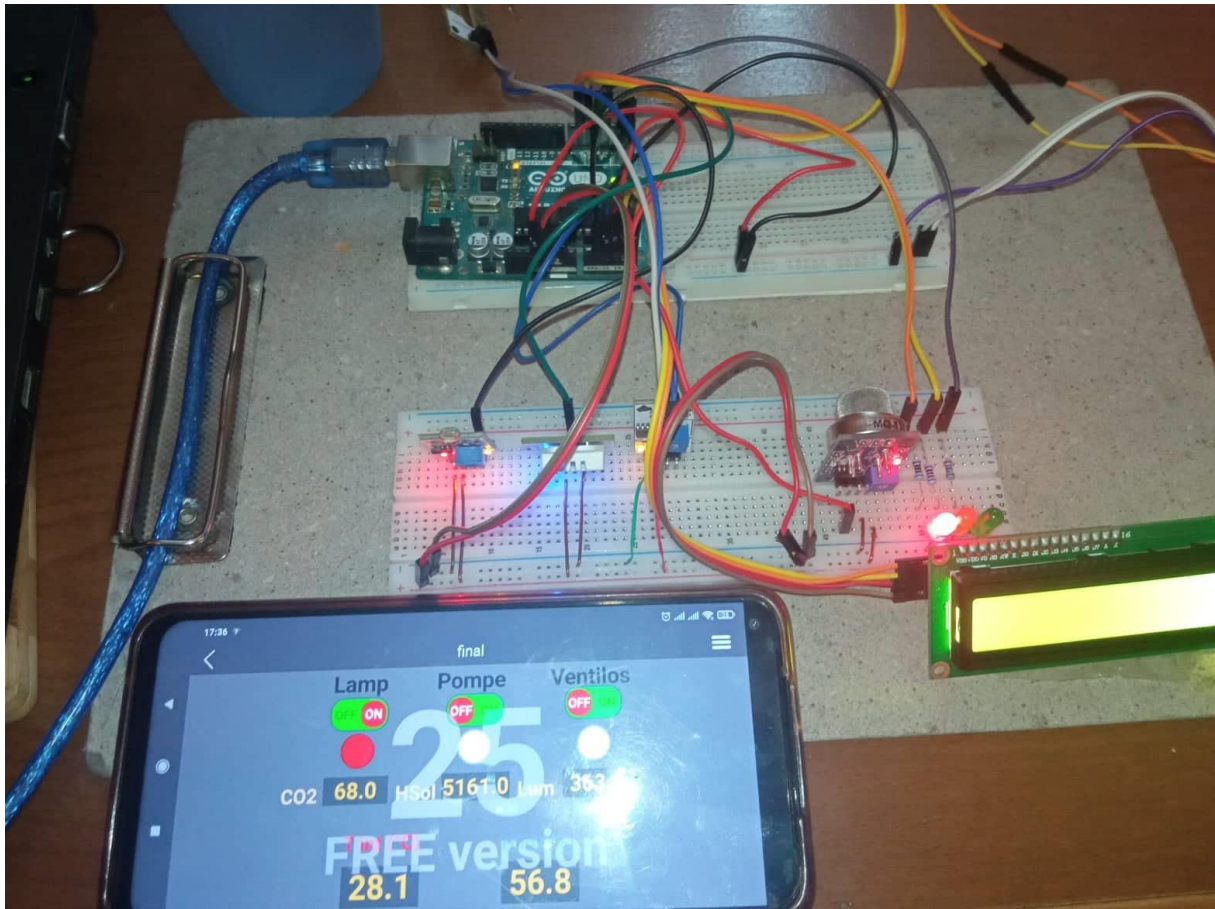


Figure 4-9 Mise en marche de la Lampe par le Smart phone

4.5. CONCLUSION

Dans ce chapitre, nous avons traduit notre cahier des charges, rédigé le code source, simulé et réalisé notre projet d'où les résultats étaient concluantes.

CONCLUSION

GENERALE

CONCLUSION GENERALE

Nous avons réalisé dans notre projet l'automation d'une serre agricole commandable à distance par smart phone via Wifi, dont nous avons suivi quatre étapes nécessaires présentées en quatre chapitres dans cette mémoire.

Dans le premier chapitre nous avons collecté des informations sur les différents types de serres et les paramètres qu'il faut mesurer pour découvrir les besoins des produits agricoles.

Et dans le deuxième chapitre nous avons étudié les capteurs, actionneurs et logiciels qu'il faut utiliser pour réaliser notre projet.

Ensuite dans le chapitre trois nous avons simulé et essayé notre outillage avant de l'utiliser, le but de cette étape est de déduire les causes en cas de problème au moment de la mise en marche de notre projet.

Dans le dernier chapitre nous avons traduit notre cahier des charges - les besoins de notre produit agricole par rapport aux informations mesurées - en algorithmes afin de faciliter la programmation de notre microcontrôleur qui doit gérer notre système automatique, ainsi que le câblage des capteurs, la carte Arduino, le module Wifi et les actionneurs.

Après la réalisation de la partie pratique, la mise en marche de notre projet était concluante par l'utilisation des LED remplaçant les actionneurs de puissance.

Au cours de la réalisation de notre projet, nous avons déduit un problème en cas où l'agriculteur change le produit agricole situé dans la serre, ce dernier peut exiger un autre besoin différent de celui que nous avons réalisé pour notre système. Pour cela, nous proposons d'améliorer notre projet en utilisant le smart phone pour changer les valeurs d'exécution dépendant des informations de n'importe quel produit alimentaire posé dans la serre.

Références

- [1] Y. BOUTERAA, Automatisation d'une serre agricole, Mémoire de Magister en Sciences Agronomiques, Ecole Nationale Supérieure D'agronomie, El Harrach ALGER, 2012.
- [2] Z. BEN SAIDJ, Gestion d'une serre agricole à base d'ARDUINO, Mémoire de Master en Electronique, Université MOULOUD MAMMERI, TIZI OUZOU, 2018.
- [3] S. BENDIDANI et A. MILOUD ABID, Etude d'un Système Automatisé «Cas d'une Serre», Mémoire de Master en Electromécanique, Université BELHADJ Bouchaib, AIN-TEMOUCHENT, 2018.
- [4] I. BENAMROUCHE et S. SAOUDI, Conception et Réalisation d'un Distributeur de Gel Hydro Alcoolique Automatique doté d'un Capteur de Température, Mémoire de Master en Génie Biomédical, Université M'Hamed BOUGARA, BOUMERDES, 2020.
- [5] S. LADJIMI et L. MAGUEMOUN, Conception et Réalisation d'un système de contrôle de température, humidité et la qualité de l'air dans un milieu hospitalier, Mémoire de Master en Electronique Biomédicale, Université MOULOUD MAMMERI, TIZI OUZOU, 2016.
- [6] M. OUALI ALAMI, La Conception d'une prise connectée basée sur la technologie d'IoT (Internet of Things), Mémoire de Master en Electromécanique, Université Sidi Mohamed Ben Abdellah, Fès MAROC, 2018.
- [7] S. CHIBANI SAMIR et F. BOUKHADRA, Commande intelligente des différents objets d'une maison, Mémoire de Master en Electronique des systèmes embarqués, Université Mohamed Larbi Ben M'hidi, Oum El bouaghi, 2021.