



LICENCE SCIENCES ET TECHNIQUES Génie Electrique

RAPPORT DE FIN D'ETUDES

Intitulé :

Commande des ouvertures d'aération d'une serre agricole

Réalisé Par :

- Jean Remy Lamah
- Al Hussein Bagayogo

Encadré par :

Pr A. Mechaqrane (FST FES)

Soutenu le 05-07-2021 devant le jury

Pr A. Mechaqrane (FST FES)

Pr T. Lamhamdi (FST FES)

Table des matières

Remerciements	3
Résumé	3
Liste des figures	4
Liste des tableaux	4
Introduction Générale.....	5
<u>CHAPITRE I : Présentation du lieu de stage</u>	6
I-Lieu du stage..	6
Equipes de recherche	6
Organigramme du laboratoire	6
Effectif & Etablissement d'accueil.....	7
Thématiques de recherche	8
<u>CHAPITRE II : Description générale d'une serre agricole.....</u>	9
II.1.Cahier des charges	9
II.1.1.Problématique	9
II.1.2.Solutions.....	9
II.1.3.Automatisation via les automates	10
II.1.4.Automatisation via arduino mega	10
II.1.5.Automatisation via raspberry	11
II.2.Serres agricoles.....	12
II.2.1 Définition de la serre agricole	12
II.2.2 Intérêt de la serre	12
II.2.3 Type de construction.....	13
II.2.4 Types de la serre	13
<u>CHAPITRE III: Environnement et outils de travail</u>	14
III.1. Support matériel.....	15
III.1.1. Actionneur (Moteur à Vérin)	15
III.1.2. Présentation de la carte Arduino Méga.....	17
III.1.3. Le Driver L298N.....	17
III.1.4. DHT22 : Capteur de température et d'humidité	17

III.1.5. L'afficheur LCD.....	18
III.2. Logiciels et Environnements de programmation.....	18
<u>CHAPITRE IV : Réalisation et développement.....</u>	21
IV-1- Implémentation du menu déroulant	21
IV-2-Commande manuelle.....	23
IV-3-Commande automatique et variation de consigne.....	24
IV-4-Transmission à distance.....	27
IV-4-1-Différentes techniques de	27
IV-4-1-1Transmission par Bluetooth.....	27
IV-4-1.2-Transmission par GSM (GSM shield).....	28
IV-4-1.3-Transmission dans un réseau local (wifi).....	28
IV-4-2-Test pratique par réseau local wifi.....	28
Conclusion et Perspectives.....	30
• Références.....	30
• Annexe.....	31

Remerciements

En préambule à ce mémoire, nous adressons nos remerciements les plus sincères aux personnes qui nous ont apportés leur aide et qui ont contribué à l'élaboration de ce travail ainsi qu'à la réussite de cette formidable année académique.

On tient à remercier sincèrement le professeur MECHAQRANE Abdellah, encadrant de ce mémoire, pour ses précieux conseils, son aide, sa disponibilité tout au long de ce stage.

Nos remerciements s'adressent également à nos enseignants pour leur disponibilité et leurs précieux enseignements qui nous ont été d'une grande utilité, mais aussi à Monsieur BYOU Abdelillah pour ses conseils et son suivi inconditionnel.

Les mots nous manquent pour exprimer notre profonde reconnaissance et notre gratitude à l'égard de nos familles dont l'amour, la patience et le sacrifice s'inscrivent à chaque page de ce document.

Enfin, on adresse nos remerciements les plus chaleureux à tous nos proches et amis, qui nous ont soutenus et encouragés durant la réalisation de ce mémoire.

Résumé

*Nous proposons dans ce mémoire la réalisation d'un système de commande des ouvertures d'aération d'une serres agricoles intelligentes. Ce système modulaire permet d'asservir suivant le choix de l'opérateur (menu déroulant : Commande **manuelle**, **automatique** ou par **consigne**) les ouvertures d'aération d'une serre agricole. L'interface utilisateur permet aussi de visualiser les paramètres climatiques de la serre. Ce système peut contribuer à optimiser les ressources et les efforts des agriculteurs et d'améliorer la productivité en termes de quantité et de qualité.*

Par l'intermédiaire d'un site web, l'opérateur peut se renseigner sur les conditions climatiques intérieures et extérieures de la serre.

Liste des figures

<u>Figure 1</u> : Organigramme du laboratoire de recherche SIGER (FST Fès).....	6
<u>Figure 2</u> : Automate programmable industriel	10
<u>Figure 3</u> : Carte Arduino Mega	10
<u>Figure 4</u> : Carte Raspberry	11
<u>Figure 5</u> : Exemple de serre agricole.....	12
<u>Figure 6</u> : Serre tunnel	13
<u>Figure 7</u> : Serre multi-chapelle	13
<u>Figure 8</u> : Serre d'essai du laboratoire SIGER.....	15
<u>Figure 9</u> : Moteur à vérin	15
<u>Figure 10</u> : Driver L298N	17
<u>Figure 11</u> : Capteur de température DHT22.....	17
<u>Figure 12</u> :Ecran LCD	18
<u>Figure 13</u> : Structure Arduino	20
<u>Figure 14</u> : Schéma de câblage du menu déroulant.....	21
<u>Figure 15</u> : Organigramme du menu déroulant	22
<u>Figure 16</u> : Câblage commande manuelle.....	23
<u>Figure 17</u> : Programme commande manuelle	24
<u>Figure 18</u> : Programme commande automatique.....	25
<u>Figure 19</u> : GSM shield.....	26
<u>Figure 20</u> :Wifi esp8266.....	27
<u>Figure 21</u> : Schéma structurel de la transmission.....	28
<u>Figure 22</u> : Température et Humidité sur thingspeak	29

Liste des tableaux

<u>Tableau 1</u> : Etude des types de serre.....	14
<u>Tableau 2</u> : Description du LCD	19

Introduction Générale

Ces dernières années, on assiste à une évolution spectaculaire des nouvelles technologies dans différents domaines d'application. En effet, avec l'avancée technologique, l'homme a pour ambition de se faciliter la vie quotidienne en s'offrant un certain confort tout en observant des règles de sécurité conformes aux normes technologiques actuelles.

Selon l'Organisation des Nations Unies pour l'alimentation et l'agriculture [1], la production alimentaire doit augmenter de 60% d'ici à 2050 pour pouvoir nourrir la population croissante, qui devrait atteindre 9 milliards. Une productivité accrue des cultures est nécessaire de toute urgence et constitue la pierre angulaire de toute solution pour faire face aux problèmes de pénurie alimentaire et de rentabilité des exploitations agricoles.

Dans le domaine agricole, la recherche permanente d'un meilleur rendement et de procédés les plus efficaces, font qu'aujourd'hui, on remarque un engouement spectaculaire pour l'utilisation des moyens technologiques des plus avancés.

De nos jours, surtout dans les pays développés, les rendements agricoles sont optimisés. Ceci est le résultat de l'approche techno-agricole.

Cette approche permet de surveiller avec les moyens technologiques les plus modernes de manière rigoureuse toutes les étapes de la production.

La fonction principale d'une serre agricole est de créer un microclimat pour assurer le bon déroulement de l'évolution des plantes. La température, l'humidité et le rayonnement, représentent les paramètres les plus importants pour la gestion du climat d'une serre. Ainsi, pour élaborer une stratégie de contrôle, on doit mettre en place un système muni de différents capteurs qui permettent à la fois l'acquisition des données et le contrôle automatique des paramètres de l'environnement intérieur pour favoriser un meilleur développement des plantes.

Dans le cadre de notre projet, on s'est intéressé à la problématique d'ouverture/fermeture des fenêtres d'aération dans une serre agricole. Ceci est important pour économiser les dépenses en chauffage et refroidissement. On a pensé qu'il serait utile de concevoir un système de commande modulaire allant de la simple commande manuelle jusqu'à la commande par un système de gestion centrale ou la commande en fixant des valeurs de consigne de température et d'humidité relative intérieures. On a abordé deux des aspects importants: la sélectivité de la commande visant à rendre le système plus autonome et la transmission des données provenant du capteur.

Ce mémoire est structuré en quatre chapitres dont l'agencement reflète la démarche utilisée et les étapes de conception.

Le premier chapitre est consacré à la présentation du lieu du stage.

En accord avec les objectifs du mémoire, les deuxièmes et troisièmes chapitres sont consacrés à la description générale de la serre et des différents matériels et logiciels utilisés dans notre étude.

Le quatrième chapitre est réservé à la description de la réalisation pratique.

On terminera par une conclusion générale et quelques perspectives de notre travail.

CHAPITRE I :

Présentation du lieu de stage

Dans ce chapitre, nous présenterons le lieu du stage, le cahier des charges de notre projet de fin d'études, une revue globale des différents types de serres agricoles et les différentes techniques de transmission à distance.

Lieu de stage

Notre stage a été effectué à la Faculté des Sciences et Techniques de Fès (FST Fès) au sein du laboratoire Systèmes Intelligent Géo-ressources et Énergies Renouvelables (SIGER).

Equipes de recherche

Le laboratoire SIGER de la FST Fès est constitué de quatre équipes de recherche:

- ① Mathématique, Informatique et Intelligence Artificielle (**M2IA**)
- ② Géologie et Géophysique appliquées aux Géoréources et Géorisques (**2GA2G**)
- ③ Systèmes Embarqués, Électronique et Télécommunications (**S2ET**)
- ④ Energies Renouvelables et Développement Durable (**ER2D**)

Nous avons effectué notre stage au sein de l'équipe Energies Renouvelables et Développement Durable (ER2D).

Organigramme du laboratoire

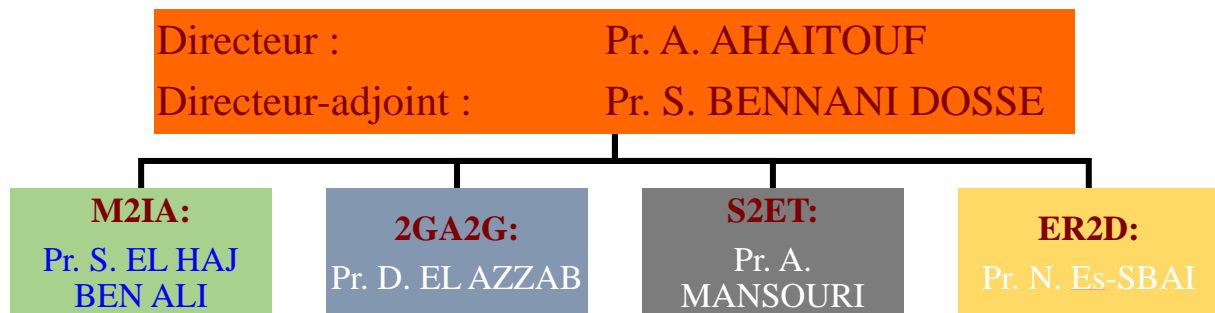


Figure 1 : Organigramme du laboratoire de recherche SIGER (FST Fès)

Effectif & Etablissement d'accueil

- Le laboratoire a un effectif de :
 - 30 Enseignants chercheurs permanents

- 77 Doctorants
- 29 chercheurs associés
- Etablissement d'accueil: FST de Fès
- Etablissement membre: ENSA de Fès

Thématiques de recherche

Les différentes équipes du laboratoire SIGER travaillent dans différents domaines de recherche complémentaires et innovants. Les principales thématiques de recherche pour chaque équipe sont:

① Mathématique, Informatique et Intelligence Artificielle

- Problème de la médiane ordonnée d'un nombre fini de fonctions rationnelles, la Théorie de Localisation et ses applications pratiques, comme en gestion et valorisation des ressources naturelles et gestion des risques.
- Calcul Scientifique, Optimisation Polynomiale, Optimisation Multi-objectifs et la Programmation Semi-définie positive, utiles aux modélisations visant la connaissance, la maîtrise et la gestion des systèmes naturels et environnementaux;
- Services Web, Ingénierie dirigée par les Modèles, Adaptations et applications aux problèmes de gestion, voire d'anticipation de l'évolution des systèmes; • Risques et sécurité des systèmes d'information;
- L'innovation industrielle au service de l'environnement et de la technologie

② Géologie et Géophysique appliquées aux Géo ressources et Géo risques

- Problème de la médiane ordonnée d'un nombre fini de fonctions rationnelles, la Théorie de Localisation et ses applications pratiques, comme en gestion et valorisation des ressources naturelles et gestion des risques.
- Calcul Scientifique, Optimisation Polynomiale, Optimisation Multiobjectifs et la Programmation Semi-définie positive, utiles aux modélisations visant la connaissance, la maîtrise et la gestion des systèmes naturels et environnementaux;
- Services Web, Ingénierie dirigée par les Modèles, Adaptations et applications aux problèmes de gestion, voire d'anticipation de l'évolution des systèmes; • Risques et sécurité des systèmes d'information;
- L'innovation industrielle au service de l'environnement et de la technologie

③ Systèmes Embarqués, Électronique et Télécommunications

- Physique des composants et des capteurs de l'échelle micro à l'échelle nanométrique, capteurs intelligents et leurs applications;

Traitement de signaux, traitement d'image et traitement de vidéo;

- Systèmes embarqués : Description matérielle, architectures parallèles, systèmes temps réel; Adéquation algorithme-architecture
- Conception des circuits intégrés analogiques et numériques multi-usagers.
- Télécommunications, Antennes microonde, propagation et RFID et exploration des opportunités de leurs usages prospectifs, y compris dans le domaine de l'environnement;

④ **Énergies Renouvelables et Développement Durable**

- Énergie solaire, énergie Éolienne, Gisement solaires, Gisement éolien
- Intégration dans les réseaux, Réseaux intelligents (smart grids)
- Véhicule électriques et hybrides, Batteries intelligents
- Solaire photovoltaïque et Photovoltaïque à concentration
- Micro-réseaux et sites isolés
- Efficacité énergétique dans le bâtiment, bâtiment vert, séchage solaire

Conclusion:

Dans le présent chapitre, nous avons présenté le lieu de stage en faisant le point sur l'organigramme du laboratoire (SIGER) notamment les différentes équipes de recherche et les thématiques abordées.

Chapitre II:

Description Générale des Serres agricoles

Dans ce chapitre, nous allons faire la description générale de la serre en évoquant la problématique de notre sujet, les éventuelles solutions, les types de serres et son intérêt.

II.1. -Cahier des charges

II.1.1. Problématique :

La fonction principale d'une serre agricole étant de créer un microclimat pour assurer le bon déroulement de l'évolution des plantes. La température, l'humidité et le rayonnement (effet de serre), représentent les paramètres les plus importants pour la gestion du microclimat d'une serre.

Pour établir ce microclimat et favoriser le développement des plantes nous sommes amenés à utiliser les systèmes de climatisation, d'irrigation et commander les ouvertures/fermetures les volets d'aération de la serre. Cette dernière action peut générer des économies colossales dans les dépenses en climatisation (chauffage-refroidissement).

II.1.2. Solutions :

Parmi les méthodes d'aération, nous nous intéressons aux systèmes utilisant des moteurs à vérins qui peuvent être commandés:

- Manuellement : à l'aide de boutons poussoir pour l'ouverture et la fermeture des volets d'aération de la serre,
- Automatiquement : en fonctions de valeurs de consignes de température et d'humidité relative intérieures de la serre,
- A distance : à partir d'un système de gestion central de la serre.

Pour réaliser ces commandes plusieurs solutions s'offrent à nous :

L'automatisation par le microcontrôleur ARDUINO, l'automatisation via la carte RASPBERRY et l'automatisation via les automates programmables industriels (API).

II.1.3. Automatisation via les automates:

L'automate est une machine électro-informatique qui permet d'automatiser les systèmes industriels notamment les serres multi-tunnels. Il peut être programmé en ladder, en logigramme (langage graphique), en grafcet....



Figure 2 : Automate programmable industriel.

L'automate est constitué de ports d'entrée sur lesquels seront branchés les différents capteurs notamment la sonde de température (pour la commande automatique) et les boutons poussoirs pour la commande manuelle et des ports de sorties reliés à un relais qui, à son tour, est connecté au moteur.

Cette technique d'automatisation est la mieux indiquée pour les grands systèmes car elle nécessite plus de moyens matériels et financiers. Elle résiste plus à certaines conditions climatiques (fortes chaleurs et fortes humidités).

II.1.4. Automatisation via Arduino:

L'Arduino Méga est un microcontrôleur basé sur l'ATmega2560. Son utilisation est simple et accessible. Il offre plusieurs avantages par rapport à d'autres cartes utilisées dans d'autres systèmes notamment : Peu coûteuse, Multiplateformes (fonctionne sur plusieurs systèmes d'exploitation), Logiciels et Matériel extensibles.

Il est basé sur un environnement de programmation qui est une variante du C et C++, allégé et restreint à l'utilisation de la carte.



Figure 3 : Carte Arduino Méga

II.1.5. Automatisation via RASPBERRY :

Raspberry est une carte mère d'un mini-ordinateur qui peut être branchée à n'importe quel périphérique (souris, clavier...). Cette carte est fabriquée pour aider à étudier les ordinateurs et pour représenter un moyen d'apprentissage de la programmation informatique en plusieurs langages (C, C++, python, scratch...) et d'être utilisée dans les systèmes embarqués. Elle est aussi capable de lire les vidéos à haute définition et même à installer des jeux vidéo.

On peut la câbler avec tous les composants utilisables sur Arduino et a la particularité d'être équipée des modules wifi, Bluetooth et GSM pour la transmission des données.



Figure 4 : Carte Raspberry

Suite à une étude comparative entre les différentes techniques d'automatisation, il en est ressorti que :

- La technique de l'automate et de la carte RASPBERRY nécessitent de grands fonds
- L'usage de l'automate pour la réalisation de ce projet est bien faisable mais n'est pas approprié car il s'agit d'un projet de petite envergure.

Pour cela nous avons été amenés à utiliser la technique via Arduino qui est moins chère et mieux adaptée pour notre projet.

II.2. Serres agricoles

II.2.1 Définition de la serre agricole

Une serre est une structure destinée en général à la production agricole protégée. Elle vise à créer un microclimat intérieur protégé pour une meilleure gestion des besoins des plantes et pour en accélérer la croissance ou les produire indépendamment des saisons.

II.2.2 Intérêt de la serre

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 la température et de l'humidité relatives intérieures. Elle joue un rôle économique en présentant des produits sur le marché hors saison.

La serre permet ainsi d'obtenir une production végétale dans des conditions meilleures que celles existantes naturellement et 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. Ceci suppose la connaissance des interactions entre les divers paramètres.



Figure 5 : Exemple de serre agricole

II.2.3 Type de construction

Les serres peuvent être entièrement vitrées, ou bien avoir un ou plusieurs côtés en planches ou en briques. Il y a des avantages et des inconvénients à l'une et à l'autre. Des demi-parois en brique, en bois ou en amiante fournissent une bonne isolation et abaissent les frais de chauffage de la serre.

II.2.4. Types de serres

La classification des serres est complexe et difficile. Elle est souvent faite selon les formes données par les cadres porteurs qui constituent l'assemblage. On distingue deux principaux types appartenant à deux grandes familles de serres : Les serres tunnels et les serres chapelles.

a Serres tunnel

Une serre tunnel (Figure 5) 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.



Figure 6 : Serre tunnel

b Serres multi-chapelle

Les différentes pièces qui composent la structure de la gamme à serres Multi chapelle (Figure 7) 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 7: Serre multi-Chapelle

Le tableau ci-dessous récapitule les différents types de serres ainsi que leurs avantages et inconvénients [2].

TYPES DE SERRES	AVANTAGES	INCONVÉNIENTS
Serres tunnels	→ Les plus utilisées par les agriculteurs. Basiques et économiques	→ Pas hautes et pas adaptées pour les tomates hors-sol
Serres plastiques Multi chapelles simple ou double paroi	→ Un grand volume d'air grâce à leur hauteur optimale ainsi qu'une largeur importante pour une production intense. → Très isolantes. Leur système d'aération avec ouverture permanente ou fermeture possible permet une ambiance saine et une aération optimale.	→ Manque de lumière
	→ Économiques et robustes	
Serres verres	→ Très lumineuses → Hauteur optimale → Très robustes	→ Plus coûteuses → Plus fragiles (sensibles à la grêle)
Serres fermées en verre	→ Isolation totale pour la serre, lui permettant de n'être affectées par aucun insecte ni maladie → Gestion de l'humidité → Adaptées à une culture très intensive de la tomate (pied jusqu'à 8 mètres de haut)	→ Très onéreuses et destinées à des gros industriels

Tableau 1 : Différents types de serres agricoles [2].

Conclusion :

Dans ce présent chapitre, nous avons fait une description générale de la serre en évoquant les différentes problématiques, son intérêt et les différents types de constructions possibles.

Chapitre III

Environnement et outils de travail

Dans ce chapitre, nous allons présenter l'ensemble des supports matériels et logiciels que nous avons utilisé dans notre projet.

III.1. Support matériel:

Le laboratoire SIGER dispose d'une petite serre d'essais (figure 8). Cette serre est dotée de deux châssis (ouvertures) d'aération.



Figure 8: Serre d'essai du laboratoire SIGER.

Pour commander ce type d'ouverture, on a opté pour l'utilisation d'un moteur à vérin.

III.1.1. Actionneur (Moteur à Vérin)

Il permet de faire l'ouverture et la fermeture de la fenêtre d'aération de la serre.

Ses caractéristiques sont les suivantes :

Longueur de levage : 61 cm.

Vitesse : 5, 6 mm/sec.

Résolution (Reed) : 48 pulses/pouces.

Tension d'entrée : 36 V DC.

Charge statique : maximum 450 kg.

Charge dynamique : maximum 250 kg.

Ce moteur sera commandé à travers une carte Arduino.



Figure 9 : Moteur a vérin

III.1.2. Présentation de la carte Arduino Méga

La carte Arduino Méga 2560 est un microcontrôleur basé sur un ATmega2560 cadencé à 16 MHz. Elle dispose de 54 E/S dont 14 PWM, 16 analogiques et 4 UARTs. Elle est idéale pour des applications exigeant des caractéristiques plus complètes que la carte Arduino Uno.

Des connecteurs situés sur les bords extérieurs du circuit imprimé permettent d'enficher une série de modules complémentaires.

Le contrôleur ATmega2560 contient un boot loader qui permet de modifier le programme sans passer par un programmeur.

Cette carte est livrée avec un support en plastique mais sans cordon USB (voir articles conseillés).

- **Caractéristiques:**

- Alimentation:
 - via port USB ou
 - 7 à 12 V sur connecteur alim
- Microprocesseur: ATmega2560
- Mémoire flash: 256 kB
- Mémoire SRAM: 8 kB
- Mémoire EEPROM: 4 kB
- 54 broches d'E/S dont 14 PWM
- 16 entrées analogiques 10 bits
- Intensité par E/S: 40 mA
- Cadencement: 16 MHz
- 3 ports série
- Bus I2C et SPI
- Gestion des interruptions
- Fiche USB B
- Version: Rev 3
- Dimensions: 107 x 53 x 15 mm

En plus de la carte Arduino, le moteur utilisé sera piloté à l'aide du driver L298N.

[III.1.3. Le Driver L298N](#)

Le driver moteur L298N est une carte d'extension compatible à la carte Arduino et qui permet aussi bien de piloter deux moteurs à courant continu que de les alimenter électriquement. La carte d'extension driver moteur L298N permet un pilotage sous une tension constante comprise entre 5 V et 35 V.

Caractéristiques du driver moteur L298N :

- Pilote : L298N de type double canal en pont en H
- Tension logique : 5 VCC
- Tension d'entraînement : 5 à 35 V
- Courant d'entraînement : 2A
- Puissance : 25 W
- Dimensions : 43*43*27 mm
- Câblage : Bornes à vis

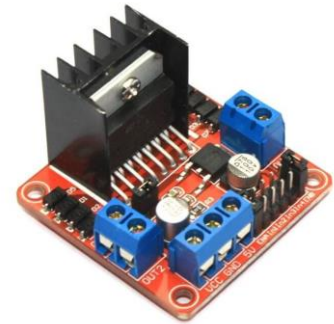


Figure 10 : Driver L298N

Pour la commande en fonction de la température de consigne et de l'humidité relative de consigne, on a utilisé le capteur DHT22.

III.1.4. DHT22 : Capteur de température et d'humidité

Le premier capteur installé et testé est le DHT 22. C'est un capteur numérique d'humidité relative et de température qui, en utilisant un capteur d'humidité capacitif et une thermistance mesurant l'air environnant, produit un signal numérique sur la broche de données.

Le capteur doit être alimenté entre 3,3 V et 5 V (et fonctionnera de -40 °C à + 80 °C avec une précision de +/- 0,5 °C pour la température et +/- 2% pour l'humidité relative). Sa période de détection est en moyenne de 2 secondes (temps minimum entre les lectures).

Le DHT22 a 4 broches :

1. VCC (3 à 5V)
2. DONNÉES (sortie de données)
3. NC (non connecté)
4. GND (Terre)

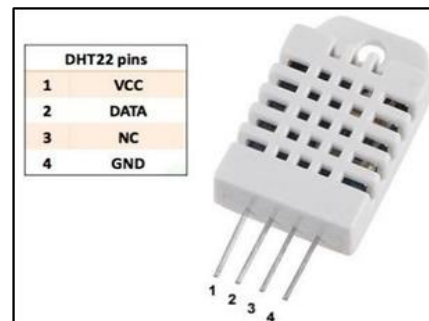


Figure 11 : Capteur DHT22

Pour défiler le menu et choisir le mode de commande désiré et afficher la température et l'humidité relative à l'intérieur de la serre, on a utilisé un afficheur LCD.

III.1.5. L'afficheur LCD

Le composant LCD est un module qui permet d'afficher des textes et de lire des données provenant des capteurs.

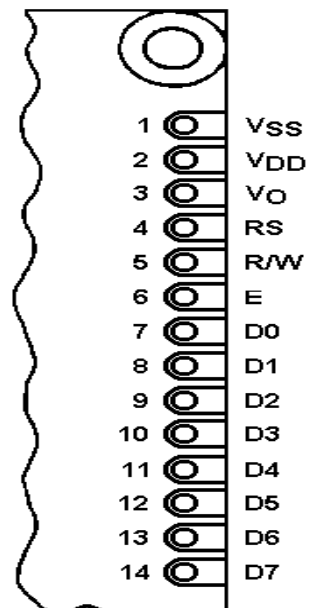


Figure 12 : Ecran LCD

L'application entre les deux faces d'une tension alternative basse fréquence de quelques volts (3 à 5 V) le rend absorbant. Les caractères apparaissent sombres sur fond clair. N'émettant pas de lumière, un afficheur à cristaux liquides réfléchissant ne peut être utilisé qu'avec un bon éclairage ambiant. Sa lisibilité augmente avec l'éclairage. Pour rendre un tel afficheur lisible, il est nécessaire de l'éclairer par l'arrière, comme c'est le cas pour les modèles rétroéclairés.

Un circuit intégré spécialisé est chargé de la gestion du module. Il remplit une double fonction: d'une part il commande l'affichage et de l'autre se charge de la communication avec l'extérieur.

Broche	Nom	Niveau	Fonction
1	Vss	-	Masse
2	Vdd	-	Alimentation positive +5V
3	Vo	0-5V	Cette tension permet, en la faisant varier entre 0 et +5V, le réglage du contraste de l'afficheur.
4	RS	TTL	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	TTL	Lecture ou écriture (Read/Write) L : Écriture H : Lecture
6	E	TTL	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	TTL	Bus de données bidirectionnel 3 états (haute impédance lorsque E=0)
8	D1	TTL	
9	D2	TTL	
10	D3	TTL	
11	D4	TTL	
12	D5	TTL	
13	D6	TTL	
14	D7	TTL	
15	A	-	Anode rétroéclairage (+5V)
16	K	-	Cathode rétroéclairage (masse)

Tableau 2 : Description du LCD [référence]

III.2. Logiciels et Environnements de programmation

Lors de notre réalisation, nous nous sommes basés sur deux logiciels, Arduino IDE et [Circuito.io](https://circuitio.io). Le premier nous a servi pour programmer et configurer notre carte Arduino pour le bon fonctionnement de notre application. Le deuxième software nous a servi dans une plateforme d'essais et de tests avant d'entamer la programmation sur la plateforme réelle.

- **Structure générale de l'environnement :**

Le logiciel de programmation de la carte Arduino sert d'éditeur de code (langage proche du C). Une fois, le programme écrit 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 logiciel est appelé IDE Arduino [3].

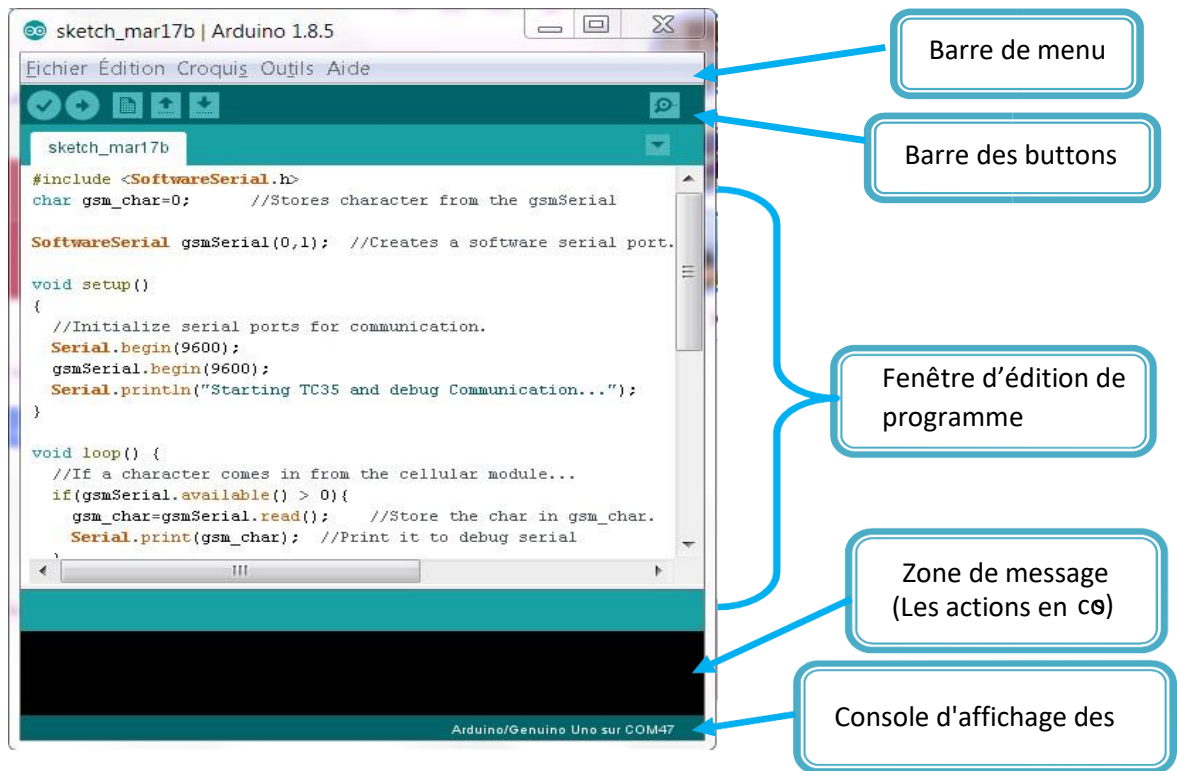


Figure 13 : structure arduino

Conclusion:

Dans le présent chapitre, nous avons présenté l'ensemble des supports matériels et logiciel que nous avons utilisé dans notre projet.

Dans le chapitre suivant, nous allons aborder le côté développement et réalisation de notre système de commande des volets d'une serre agricole.

CHAPITRE IV : Développement et Réalisation

Dans ce chapitre, nous présenterons l'implémentation des différentes actions nécessaires pour notre projet. Nous décrirons, en particulier, la sélection du type de commande à utiliser (manuelle, par consigne, automatique) à partir d'un menu déroulant qui servira d'interface utilisateur et la transmission des données provenant du capteur de température et d'humidité relative (DHT22).

IV.1. Implémentation du menu déroulant

Cette partie est un support nécessaire pour notre projet, dans la mesure où elle vise à concevoir un menu permettant à un utilisateur de choisir entre les différentes commandes notamment : manuelle, automatique, ou par consigne.

Pour cela on a utilisé cinq (5) boutons poussoirs.

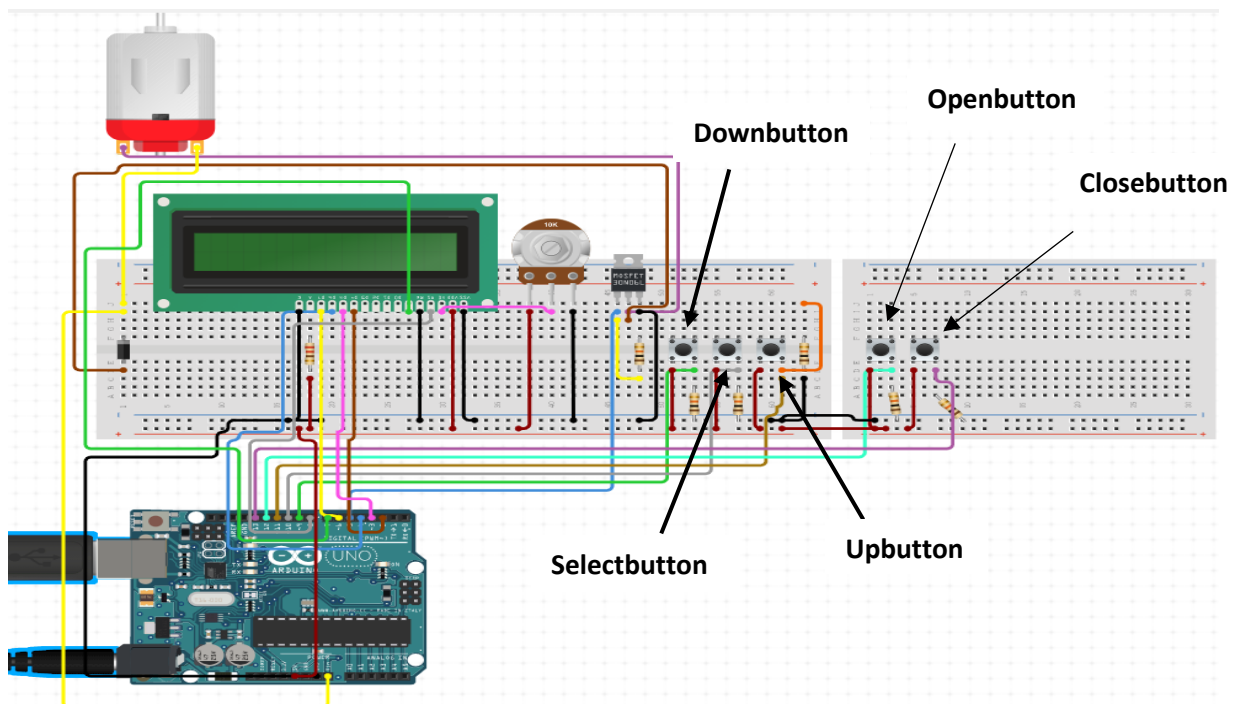


Figure 14 : Schéma de câblage - menu déroulant

Les trois premiers boutons représentent le Downbutton pour descendre, le Upbutton pour monter et le Selectbutton (bouton central) pour sélectionner.

Les deux derniers boutons permettent, suivant le choix de l'opérateur, d'incrémenter ou de décrémenter la température de référence s'il s'agit de varier la consigne ou de déclencher la marche avant (Openbutton) ou arrière (Closebutton) du moteur s'il s'agit de la commande manuelle.

Le menu est structuré comme suit :

→ Commande manuelle

- Commande automatique
- Consigne
- Affichage (Pour afficher la température et l'Humidité)

Ordinogramme :

Le programme général du menu est structuré comme suit :

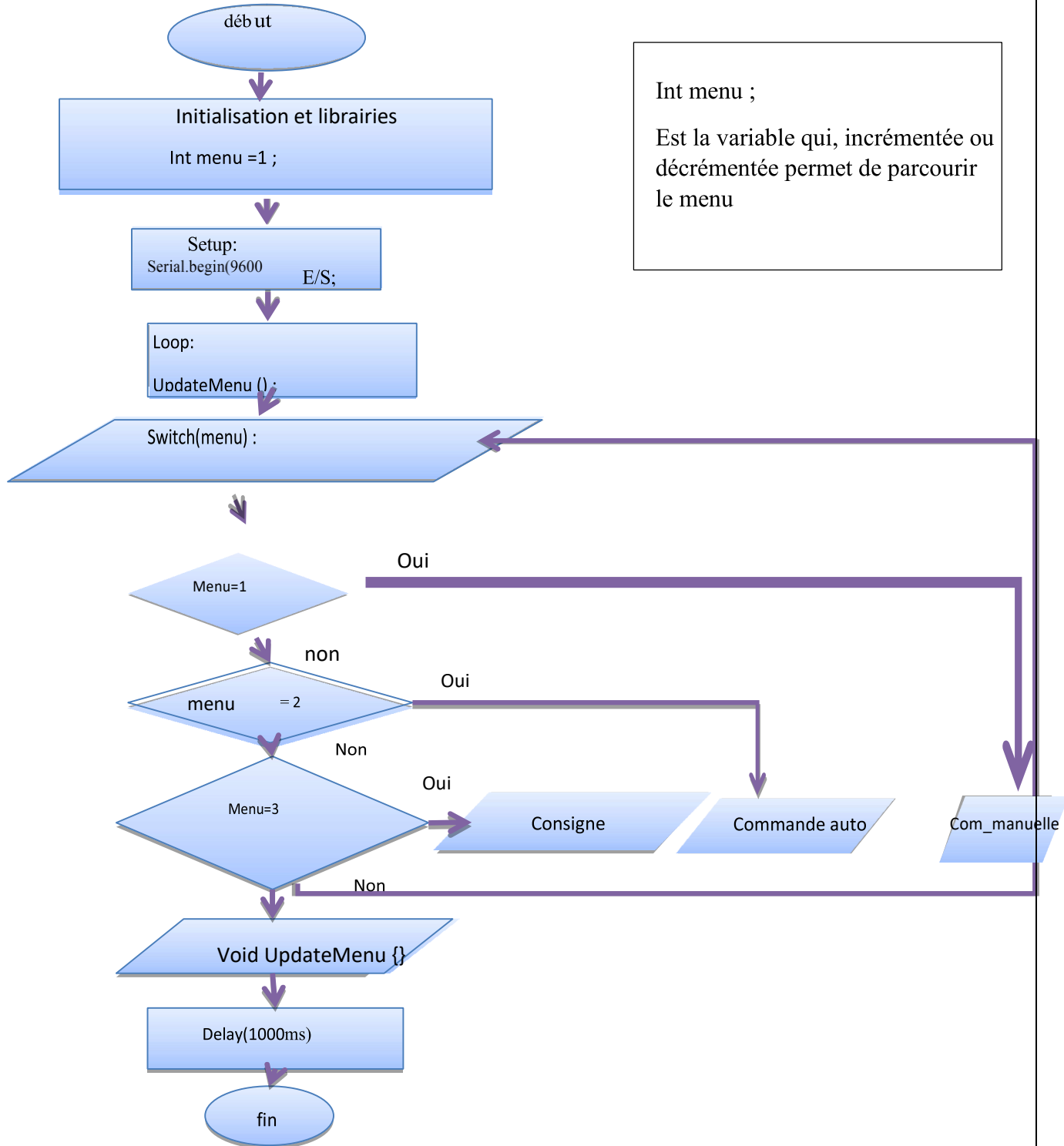
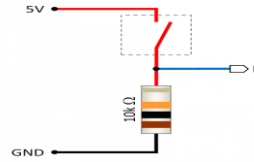


Figure 15: Organigramme du menu déroulant

Nb : Le programme général est mis en annexe.

IV.2. Commande manuelle

Dans cette partie il s'agit de commander les ouvertures d'aération (moteur à verin) à l'aide des boutons poussoirs.



L'appui sur un bouton produit une impulsion et cette impulsion est transmise au programme arduino .

a- Problématique :

La problématique de ce volet repose sur :

- Comment adapter un moteur à verin (parabole) à un programme arduino ?
 - le fonctionnement des boutons poussoirs (mode de branchement aval ou amont)
 - Comment asservir le moteur par teleruption c'est-à-dire suite à un appui sur un bouton le moteur marche dans le sens direct jusqu'à ce qu'on appui sur un autre bouton ?

b- Solutions apportées :

En ce qui concerne le moteur à verin, c'est un moteur DC (courant continu) de 36v. Pour l'asservir dans un premier temps, nous avons utilisé un SENSOR pour alimenter son circuit magnétique (adaptation) et pour la programmation avec arduino nous avons utilisé un relais de 5-36 v (driver L298N) car arduino ne delivre que 5V à la sortie.

Schéma de câblage :

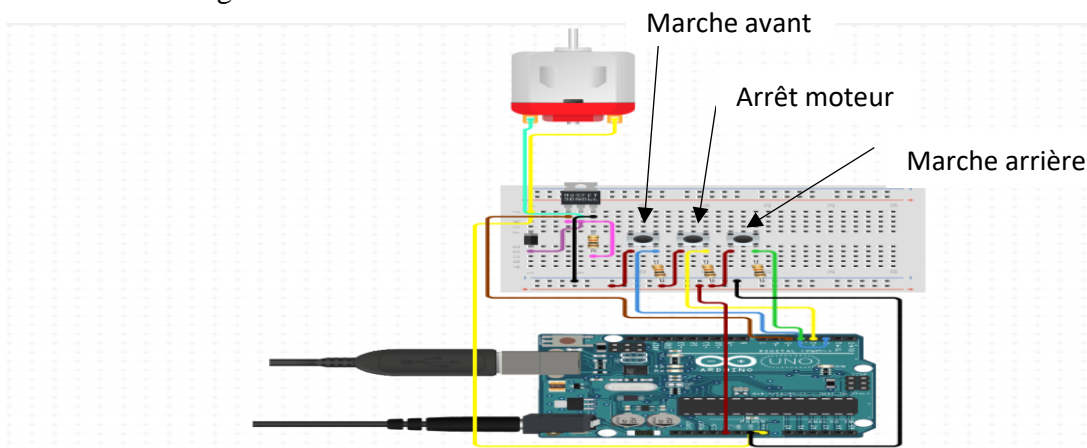


Figure 16 : Câblage commande manuelle

A ce niveau, trois boutons poussoirs sont mis en jeu: un premier pour la marche avant du moteur, un deuxième pour la marche arrière et le troisième pour l'arrêt .

Sur ce chemin nous avons utilisé un petit moteur DC avec le même principe que le moteur à verin.

Programme :

```
int g=2;
int h=2;
if (digitalRead(cons_up)== LOW){
    while(g>0){
        digitalWrite
(in1,LOW);
        digitalWrite
(in2,HIGH);
        analogWrite(
enA,200);
        g--;
    }
    while(digitalRead(cons_up
)== LOW);
    s--;
}
if (digitalRead(cons_down)== LOW){
    while(h>0){
        digitalWrite(in1,HIGH);
        digitalWrite(in2,LOW);
        analogWrite(enA,200);
        h--;}
}

while(digitalRead(cons_down)== LOW);
s--;
}
if (digitalRead(upButton)== LOW){
    digitalWrite(in1,LOW);
    digitalWrite(in2,LOW);
    analogWrite(enA,200);
    h=0;
g=0;
s=0;
while(digitalRead(upButton)== LOW);
}
}
```

Figure 17 : Programme commande manuelle

Les tests effectués ont montré que l'implémentation est correcte et qu'on peut utiliser, efficacement, cette solution pour commander manuellement l'ouverture et la fermeture du châssis de notre serre d'essais.

IV.3. Commande automatique et variation de la consigne

Il s'agit de commander les ouvertures d'aération de la serre en fonction de la température et de l'humidité relative de consigne c'est-à-dire qu'une température de référence (27.50°C) a été fixée à partir de laquelle les ouvertures et fermetures seront commandées. Si initialement la température dans la serre est supérieure à 27.5°C, le verin actionne l'ouverture sur une portée de 30cm puis s'arrête et si la température descend en dessous de 27.5°C, le verin actionne la fermeture puis s'arrête et le cycle reprend .

Nous avons eu à utiliser un moteur pas a pas de 5V pour exécuter cette tâche et par la suite on a adapté ce programme à notre moteur à verin.

a-**Problématique qui se posait:**

-Comment établir un programme permettant de remplir ce cahier de charge ?

b-Solutions apportées :

-Pour remplir ce cahier de charges plusieurs solutions se sont offertes à nous :

Au début nous avons déclaré dans le programme un tableau dynamique permettant de stocker les différentes valeurs de la température et ensuite comparer les $t(i)$ et $t(i-1)$ mais cette alternative n'était pas optimale car elle nécessitait d'allouer de l'espace dans la mémoire de l'arduino

Finalement on a déclaré une variable globale ($c=1$) qui a été la 2^e condition pour la commande.

Programme :

```
float temperature;

float humidite;

    lcd.clear();
    lcd.print(" Moteur en marche");
temperature = dht.readTemperature();
humidite = dht.readHumidity();

if(isnan(temperature) || isnan(humidite)){

else
{
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("Temp: ");
    lcd.print(temperature);
    lcd.print("C");
    lcd.setCursor(0,1);
    lcd.print("Humi: ");
    lcd.print(humidite);
    lcd.print("%");
}
}
```

```
Serial.print("c= ");
Serial.println(c);

if(temperature<temp_ref && c==1)
{
    i=0;
    ouvrir();
}

if(temperature>=temp_ref && c==0)
{
    j=0;
    fermer();
}

// delay(500);

if(temperature<temp_ref)
    c=0;
else
    c=1;
```

Figure18 :Programme commande automatique

c-Variation de la consigne pour la Commande automatique :

Dans cette partie, il s'agit d'implémenter un programme permettant de varier la température de référence de la commande automatique en se servant des boutons poussoirs ;

Tout comme la commande manuelle, on utilise trois boutons : le 1^{er} pour incrémenter la valeur initiale de la température de référence, le 2^e pour décrémenter et le 3^e pour sélectionner notre choix.

L'intérêt de ce volet réside dans le fait qu'on peut modifier manuellement la température de référence de la commande automatique pour l'adapter en fonction de la plante à l'intérieur de la

serre.

Programme :

```
lcd.clear();
lcd.print(">Executing #4");
delay(50);
lcd.clear();
lcd.print(temp_ref);

r=5;
while (r>1){
if (digitalRead(cons_up)==LOW)
{
temp_ref++;
lcd.clear();
lcd.setCursor(0,0);
lcd.print(temp_ref);
delay(500);
while((digitalRead(cons_up)==LOW));
}
if (digitalRead(cons_down)==LOW)
{
temp_ref--;
lcd.clear();
lcd.setCursor(0,0);
lcd.print(temp_ref);
delay(100);
while((digitalRead(cons_down)==LOW));
}
if(digitalRead(upButton)==LOW)
{
r=0;
while(digitalRead(upButton)==LOW);
}

}
lcd.clear();
lcd.print(temp_ref);
delay(100);
```

Les tests effectués ont montré que l'implémentation est correcte et qu'on peut utiliser, efficacement, cette solution pour commander par des consignes l'ouverture et la fermeture du châssis de notre serre d'essais.

Dans la partie qui suit nous allons nous intéresser à la transmission des données du capteur de température et de l'humidité relative pour permettre à un utilisateur de conditions climatiques de la serre à distance.

IV. 4. Transmission à distance

Pour pouvoir choisir un système de transmission adéquat à notre problématique, on a commencé par faire un inventaire des solutions existantes.

Evidemment, l'intérêt de transmission des données relève un intérêt primordial dans plusieurs domaines (météorologie, contrôle à distance, agriculture de précision, ...). De ce fait, différentes solutions ont été proposées. On s'intéressera dans ce chapitre aux solutions à base de la carte Arduino Méga que nous utilisons dans notre application.

IV.4.1. Différentes techniques de transmission :

IV.4.1.1. Transmission par Bluetooth

Pour envoyer des données de la carte Arduino Méga par Bluetooth, on peut utiliser le module HC-06 qui permet de connecter cette carte à un autre système (smartphone, ordinateur ...).

Mais ce module a une très faible portée (sur un rayon de l'ordre d'une dizaine de mètres) et un faible débit.

IV.4.1.2. Transmission par GSM (GSM shield)

Pour faire une liaison entre notre carte microcontrôleur Arduino Méga et un téléphone portable, on peut utiliser un module GSM. Ce module permet d'envoyer et recevoir des SMS, des données ou des communications vocales depuis le réseau mobile.



Figure 19 : GSM shield

IV.4.1.3. Transmission par réseau local (wifi)

Ce support facilite la réalisation des projets connectés, robotique ou objets intelligents.

Description (ESP8266)

Parmi les modules WIFI les plus utilisés, on trouve le module WIFI ESP8266. Ce module est un circuit intégré à microcontrôleur avec connexion Wifi développé par le fabricant chinois Expressif.



Figure 20 : Wifi esp8266

IV.4.2. Test pratique par réseau local (wifi)

Après l'inventaire des différentes techniques de transmission, notre choix s'est porté sur la transmission par Wifi du fait de sa grande portée.

Pour Cela, on a utilisé une Plateforme dédiée à l'IOT (internet of things) permettant la commande à distance des objets connectés dénommée Thingspeak.com

Pour Son utilisation il faut suivre les étapes suivantes :

- Création d'un compte
- Création d'une Chaîne
- Création de deux Fields(domaine) pour afficher la température et l'humidité
- Un module Wifi (ESP8266) câblé avec l'arduino et le DHT22
- Un programme

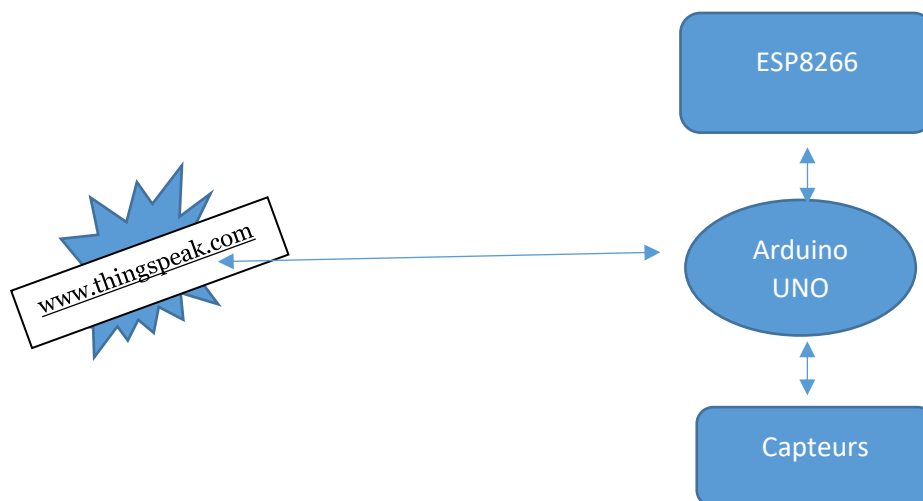


Figure 21 : Schéma structurel de la transmission

Par l'intermédiaire de la carte arduino UNO, on a établi un code permettant de transmettre en temps réel les valeurs de la température et de l'humidité provenant du capteur (DHT) sur thingspeak.com comme le montre la figure suivante :

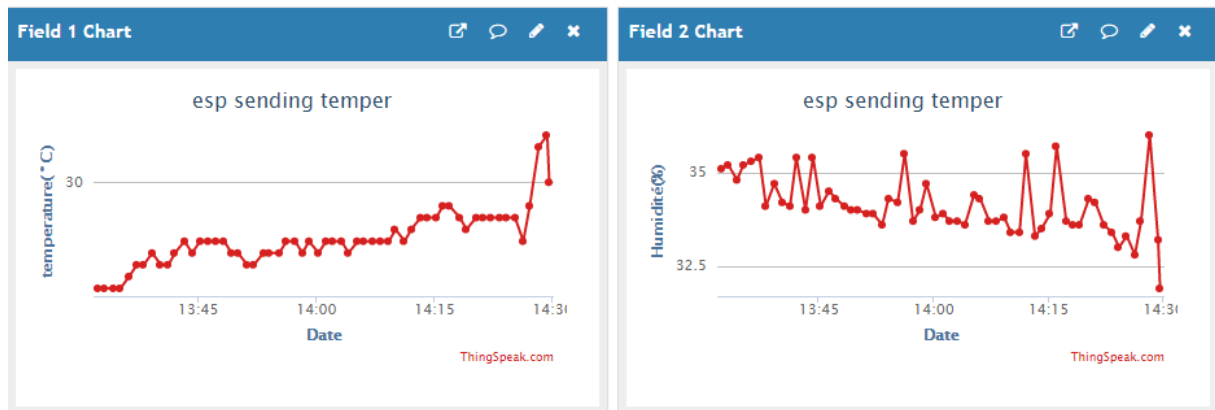


Figure 22 : Température et Humidité sur thingspeak

En mettant le curseur sur la courbe, nous obtenons la date et la valeur de la donnée (température ou humidité) en temps réel.

Conclusion et perspectives

Dans le cadre de ce projet, nous nous sommes intéressés à la conception et au développement d'un système électronique. Au cours de la période d'élaboration du projet, nous avons atteint l'objectif attendu : *d'une part la commande des ouvertures d'aération de la serre* agricole soit manuellement, soit automatiquement en fonction de la température soit par variation de consigne et le tout structuré à partir d'un menu déroulant pour rendre la commande sélective et *d'autre part la transmission des données délivrées par le capteur de température et d'humidité relative* permettant de recueillir ces données à distance.

Nettement ce projet, nous a permis d'approfondir nos connaissances théoriques et d'acquérir une certaine expérience au niveau de la réalisation pratique. Nous avons eu l'occasion d'étudier, de concevoir et d'utiliser une diversité de matériels et logiciels et appliquer notre savoir et savoir-faire acquis lors de notre formation.

Malgré les efforts que nous avons déployés pour réaliser ce modeste travail, malgré les concepts que nous avons acquis, important est de signaler que nous ne sommes qu'au début d'un long chemin. L'intérêt accordé à notre système réside dans le fait qu'il peut être amélioré. Un programme Wifi permet d'apporter une connexion Wifi par ligne série au module WIFI (ESP8266). Cela va permettre de commander la serre à distance, ce système peut être équipé par un moteur alternatif pour déclencher une ventilation dans la serre dans le cas où la température atteint la valeur maximale. On peut aussi utiliser une alarme sonore qui nous indique si la température et le taux d'humidité ne sont pas dans la plage de fonctionnement.

Références

<https://www.aurel32.net/elec/lcd.php> [III.1.5. L'afficheur LCD/page 18] 03/05/2021

<https://prototype3d.fr/fr/664-driver-moteur-l298n.html> [driver L298N/page 16] 17/05/2021

<https://www.arduino.cc/> [Site de téléchargement de l'arduino] 03/05/2021

<https://www.circuito.io/app?5components=512,11021> [Figure13: câblage - menu déroulant/page 21] 19/05/2021

<https://www.orbitadigital.com/fr/satellite-terrestre/antennes/paraboliques/moteurs/226-superjack-harl3624-moteur-d-actionneur-pour-positionneur-24-pouces-pour-antennes-180-240cm.html> [III.1.1. Actionneur (Moteur à Vérin) /page 15] 14/05/2021

<https://fr.wikipedia.org/wiki/ESP8266> [Description ESP8266/page 27] 18/06/2021

https://thingspeak.com/channels/1401821/private_show [figure 19/page 28] 24/06/2021

Annexe :

Programme général du projet

```
#include <Wire.h>

#include <LiquidCrystal.h>

#include <DHT.h>

#define DHTPIN 39

#define DHTTYPE DHT22

DHT dht (DHTPIN,DHTTYPE);

LiquidCrystal lcd(34,32,30,28,26,24);

float temp_ref=27.50;

int upButton =22;

int downButton = 7;

int selectButton = 6;

int cons_up=46;

int cons_down=48;

int menu = 1;

int act2;

int v=0;

int enA = 43;

int in1 = 45;

int in2 = 49;

int j=0;

int i=0;

int c=1;

int r;

int s;

int t=1;

float temperature;

float humidite;

void ouvrir() {

    while( i<100)
```

```
    {

        digitalWrite(in1,HIGH);

        digitalWrite(in2,LOW);

        analogWrite(enA,200);

        delay(500);

        i++;

    }

    if(i==100)

    {

        digitalWrite(in1,LOW);

        digitalWrite(in2,LOW);

        delay(500);

    }

}

void fermer() {

    while( j<100)

    {

        digitalWrite(in1,LOW);

        digitalWrite(in2,HIGH);

        analogWrite(enA,200);

        delay(500);

        j++;

    }

    if(j==100)

    {

        digitalWrite(in1,LOW);

        digitalWrite(in2,LOW);

        delay(500);

    }

}

void setup() {

    Serial.begin(115200);
```

```

pinMode(enA, OUTPUT);
pinMode(in1, OUTPUT);
pinMode(in2, OUTPUT);

pinMode(cons_reset, INPUT_PULLUP);
pinMode(cons_set, INPUT_PULLUP);
pinMode(cons_up, INPUT_PULLUP);
pinMode(cons_down, INPUT_PULLUP);

dht.begin();
lcd.begin(16,2);

pinMode(upButton, INPUT_PULLUP);
pinMode(downButton, INPUT_PULLUP);
pinMode(selectButton, INPUT_PULLUP);

updateMenu();
}

void loop() {

while(t==1)
{
if (digitalRead(downButton)==LOW){

menu++;

updateMenu();

delay(100);

while (digitalRead(downButton)==LOW);

}

if (digitalRead(upButton)==LOW){

menu--;

updateMenu();

delay(100);

while(digitalRead(upButton)==LOW);

}

if (digitalRead(selectButton)==LOW){

updateMenu();

executeAction();

delay(100);

```

```

while (digitalRead(selectButton)==LOW);

}

}

if(t==2)
{

lcd.clear();

lcd.print(" Moteur en marche");

temperature = dht.readTemperature();

humidite = dht.readHumidity();

if(isnan(temperature) || isnan(humidite)){

else

{

lcd.clear();

lcd.setCursor(0,0);

lcd.print("Temp: ");

lcd.print(temperature);

lcd.print("C");

lcd.setCursor(0,1);

lcd.print("Humi: ");

lcd.print(humidite);

lcd.print("%");

}

Serial.print("c= ");

Serial.println(c);

if(temperature<temp_ref && c==1)

{

i=0;

```

```

        }
        //delay(100);
    if(temperature>=temp_ref && c==0)
    {
        j=0;
        fermer();
    }
    // delay(500);

    if(temperature<temp_ref)
        c=0;
    else
        c=1;
    /* if(digitalRead(cons_up)==LOW)
    {
        digitalWrite(in1,LOW);
        digitalWrite(in2,LOW);
        t=1;
        while(digitalRead(cons_up)==LOW);
    }*/
}

while(t==3)
{
    lcd.clear();
    //lcd.print(">Executing #5");

    temperature = dht.readTemperature();
    humidite = dht.readHumidity();
    //delay(500);
    if(isnan(temperature) || isnan(humidite)){
    }

    else{
        lcd.setCursor(0,0);
        lcd.print("Temp: ");

```

```

        lcd.print(temperature);
        lcd.print("C");
        lcd.setCursor(0,1);
        lcd.print("Humi: ");
        lcd.print(humidite);
        lcd.print("%");
    }
    delay(500);
    if (digitalRead(upButton)==LOW)
    {
        t=1;
        while(digitalRead(upButton)==LOW);
    }
}

void updateMenu(){
    switch (menu) {
        case 0:
            menu = 1;
            break;
        case 1:
            lcd.clear();
            lcd.print(">com_manuelle");
            lcd.setCursor(0, 1);
            lcd.print(" com_auto");
            break;
        case 2:
            lcd.clear();
            lcd.print(" com_manuelle");
            lcd.setCursor(0, 1);
            lcd.print(">com_auto");
            break;
        case 3:

```

```

    lcd.print(">com_distance");
    lcd.setCursor(0, 1);
    lcd.print(" consigne");
    break;
case 4:
    lcd.clear();
    lcd.print(" com_distance");
    lcd.setCursor(0, 1);
    lcd.print(">consigne");
    break;
case 5:
    lcd.clear();
    lcd.setCursor(0, 1);
    lcd.print(">affichage");
    break;
case 6:
    menu = 5;
    break;
}
}

```

```

void executeAction() {
    switch (menu) {
        case 1:
            action1();
            break;
        case 2:
            action2();
            break;
        case 3:
            action3();
            break;
        case 4:
            action4();
            break;
        case 5:

```

```

            action5();
            break;
        }
    }

void action1() {
    lcd.clear();
    lcd.print(">Executing #1");
    s=1;
    while(s>0){
        int g=2;
        int h=2;
        if (digitalRead(cons_up)== LOW){
            while(g>0){
                digitalWrite(in1,LOW);
                digitalWrite(in2,HIGH);
                analogWrite(enA,200);
                g--;
            }
            while(digitalRead(cons_up)== LOW);
            s--;
        }
    }
    if (digitalRead(cons_down)== LOW){
        while(h>0){
            digitalWrite(in1,HIGH);
            digitalWrite(in2,LOW);
            analogWrite(enA,200);
            h--;
        }
        while(digitalRead(cons_down)== LOW);
        s--;
    }
}

```

```

digitalWrite(in1,LOW);
digitalWrite(in2,LOW);
analogWrite(enA,200);
h=0;
g=0;
s=0;
while(digitalRead(upButton)== LOW)
}

}

}

void action2() {
lcd.clear();
lcd.print(">Executing #2");
t=2;

}

void action3() {
lcd.clear();
lcd.print(">Executing #3");
delay(1500);
}

void action4() {
lcd.clear();
lcd.print(">Executing #4");
delay(50);
lcd.clear();
lcd.print(temp_ref);

r=5;

while (r>1){
if (digitalRead(cons_up)==LOW)
{

```

```

temp_ref++;
lcd.clear();
lcd.setCursor(0,0);
lcd.print(temp_ref);
delay(500);
while((digitalRead(cons_up)==LOW));
}
if (digitalRead(cons_down)==LOW)
{
temp_ref--;
lcd.clear();
lcd.setCursor(0,0);
lcd.print(temp_ref);
delay(100);
while((digitalRead(cons_down)==LOW));
}
if(digitalRead(upButton)==LOW)
{
r=0;
while(digitalRead(upButton)==LOW);
}

}

}

lcd.clear();
lcd.print(temp_ref);
delay(100);
}

void action5() {
t=3;
}

```