

الجمهورية الجزائرية الديمقراطية الشعبية  
وزارة التعليم العالي والبحث العلمي

UNIVERSITE BADJI MOKHTAR - ANNABA  
BADJI MOKHTAR – ANNABA UNIVERSITY



جامعة باجي مختار – عنابة

Faculté : Sciences de L'ingénieur  
Département : Electronique  
Domaine : Sciences et Techniques  
Filière : Automatique  
Spécialité : Automatique et informatique industrielle

Mémoire

Présenté en vue de l'obtention du Diplôme de Master

Thème:

**Gestion d'une serre agricole dans les modes local et à distance**

Présenté par : *Bendjeddou Amar*

Encadrant : *Lakal Rabah*

Professeur

*Annaba*

Jury de Soutenance :

Debbache Nasr Eddine	Professeur	Annaba	Président
Lakel Rabah	Professeur	Annaba	Encadrant
Khaldouna Zahia	MCB	Annaba	Examineur

Année Universitaire : 2020/2021

# Remerciements

*Je remercie, avant tout, Dieu, le Tout-Puissant, de m'avoir accordé ses innombrables Grâces, santé et courage pour accomplir ce travail.*

*Je tiens à exprimer notre reconnaissance à notre directeur de mémoire. Pr Lakel Rabah pour sa patience, sa disponibilité et surtout ses judicieux conseils, qui sont contribués à alimenter notre réflexion.*

*Je remercie mon chère ami et frère Raid Daroui .*

*Je remercie chaleureusement mes chères parents, mes frères.*

*Je remercie tous mes ami(e)s et mes partenaires dans ce travail pour leur sincère amitié et confiance, et pour leur aide et information à qui je dois ma reconnaissance et mon attachement. À tous ces intervenants, je présente mes remerciements, mon respect et ma gratitude.*

*Enfin je souhait témoigner également mes profonds remerciements à tous les membres et les étudiants de notre promo en Automatique qui, de près ou de loin, ont contribué à l'avancement de ce travail.*

# *Dédicaces*

*À nos très chers parents, source de vie, d'amour et d'affection*

*À nos chers frères, source de joie et de bonheur*

*À notre famille, source d'espoir et de motivation*

*À tous nos amis, et tous les étudiants de notre promotion*

*À vous cher lecteur*

## **Résumé :**

L'objectif principal de la production végétale sous serres est d'optimiser les revenus financiers des agriculteurs; Ceci est réalisé en trouvant un juste équilibre entre L'amélioration de la qualité et de la quantité des produits horticoles et la maîtrise des coûts financiers indispensables pour obtenir le microclimat nécessaire à la croissance de la plante. La croissance de la plante est influencée principalement par les variables climatiques, la quantité d'eau.

Alors que la serre est un milieu fermé dans lequel les variables climatiques et l'irrigation fertilisante peuvent être contrôlés. Ce mémoire est structuré en trois chapitres, le premier chapitre est réservé à la présentation d'un modèle de connaissance de la croissance de la plante : cas de la tomate. Le chapitre deux est réservé à la présentation des matériels et les logiciels utilise. Le chapitre trois est réservé à la présentation de notre projet.

## ملخص :

إن الهدف الرئيسي من الزراعة في البيوت البلاستيكية هو الوصول إلى أكبر مردود مالي ممكن للفلاحين, و هذا بتحقيق توازن بين تحسين نوعية الإنتاج و كميته و المصاريف اللازمة لإيجاد الجو المناسب لنمو النبتة. يتأثر نمو النبات بشكل أساسي بالمتغيرات المناخية وكمية المياه.

بينما البيت البلاستيكي هو مكان مغلق يسهل فيه التحكم في الجو و الماء. هذا البحث يتكون من ثلاثة أجزاء, الجزء الاول مخصص لعرض نموذج لمعرفة نمو النبات: حالة الطماطم. الجزء الثاني مخصص لعرض استخدامات الأجهزة والبرامج. الجزء الثالث مخصصا لعرض مشروعنا.

## **Abstract:**

The main objective of greenhouse crop production is to optimize the financial income of farmers; This is achieved by finding the right balance between improving the quality and quantity of horticultural products and controlling the financial costs essential to obtain the microclimate necessary for the growth of the plant. The growth of the plant is influenced mainly by climatic variables, the amount of water.

While the greenhouse is a closed environment in which climatic variables and fertilizing irrigation can be controlled. This dissertation is structured in three chapters, the chapter one is reserved for the presentation of a model of knowledge of plant growth: the case of the tomato. Chapter two is reserved for the presentation of hardware and software uses. Chapter three is reserved for the presentation of our project.

# SOMMAIRE

Introduction Générale.....	1
<b>Chapitre I : ----- Modélisation de la croissance de la plante -----</b>	
<b>Application à la tomate</b>	
I.1. Introduction.....	3
I.2. Mesures utilisées par le système de régulation.....	3
I.3. Intérêts de la détermination du microclimat.....	4
I.3.1. La température.....	4
I.3.1. L'humidité.....	5
I.4. Exemples des serres agricoles.....	5
I.4.1. Les exigences climatique de la tomate .....	6
I.4.1.1. Température de la tomate.....	6
I.4.1.2. humidité de sol.....	6
I.4.1.2. humidité d'aire.....	7
I.5. Gestion du climat.....	7
I.5.1. Gestion de température et d'humidité.....	7
I.5.1.1. Réseau de capteur de température et humidité d'air.....	7
I.5.1.2. Réseau de capteur de l'humidité du sol.....	8
I.5.1.3. Moyens de communication.....	8
I.6. Conclusion.....	8

## **Chapitre II : ----- Présentation des matériels et les logiciels-----**

II.1. Introduction.....	10
II.2. Partie matériel.....	10
II.2.1. La carte Arduino.....	10
II.2.1.1. Types des cartes Arduino.....	10
II.2.1.2. Carte Arduino uno.....	10
II.2.1.3. Microcontrôleur.....	11
II.2.1.4. Entrées/Sorties numérique d'une carte Arduino.....	13
II.2.1.4. Entrées/Sorties analogique d'une carte Arduino.....	13
II.2.2. Composants électroniques.....	14
II.2.2.1. GSM module sim800l.....	14
II.2.2.1.1. Description .....	14
II.2.2.1.2. Caractéristiques .....	15
II.2.2.1.3. Le fonctionnement.....	15
II.2.2.2. Capteur d'humidité YL-69.....	15
II.2.2.2.1. Description.....	15
II.2.2.2.2. Branchement sur la carte Arduino.....	16
II.2.2.2.3. Caractéristiques du capteur d'humidité.....	16
II.2.2.3. Capteur de température et d'humidité DHT22.....	17
II.2.2.3.1. Description.....	17
II.2.2.3.2. Les principales caractéristiques de DHT22.....	18
II.2.2.4. Pompe d'arrosage.....	18
II.2.2.5. Le relais.....	19
II.2.2.5.1. Brochage du relai.....	20



II.2.2.6. Ventilateur.....	20
II.2.2.7. le servomoteur.....	21
II.2.2.8. Le LED.....	22
II.3. Partie logiciels.....	22
II.3.1. Arduino.....	22
II.3.1.1. L’environnement de la programmation.....	22
II.3.1.2. Langage de programmation.....	23
II.3.1.3. Structure générale d’un programme en langage Arduino.....	23
II.3.1.4. Etapes de l’implémentation sur le microcontrôleur.....	24
II.3.2. Matlab.....	25
II.3.2.1 Introduction .....	25
II.3.2.2. Interface graphique.....	25
II.4. Schéma de nos matériels par fritzing.....	27
II.5. Conclusion.....	27

## **Chapitre III : ----- Réalisation pratique -----**

III.1. Introduction.....	29
III.2. Réalisation partique.....	29
III.2.1. Réalisation de la serre.....	29
III.2.2. Montage des capteurs et les actionnaires.....	30
III.2.3. Programme Arduino IDE.....	31
III.3. Partie local.....	33
III.4. Partie à distance.....	37
III.4.1. Communication avec site.....	37
III.4.2. Contrôle et commande de la serre.....	40

III.4. Conclusion..... 46

# LISTE DES FIGURES

## Chapitre I : ----- Modélisation de la croissance de la plante ----- Application à la tomate

Figure I.1 : une serre intelligence.....	3
Figure I.2 : culture de tomate.....	5

## Chapitre II : ----- Présentation des matériels et les logiciels-----

Figure II.1 : carte Arduino UNO.....	11
Figure II.2 : microcontrôleur ATmega328.....	12
Figure II.3 : schéma ATmega328.....	13
Figure II.4 : module Gsm sim800l.....	14
Figure II.5 : capteur yl-69.....	16
Figure II.6 : capteur de température et d'humidité.....	18
Figure II.7 : pompe d'une machine à laver.....	19
Figure II.8 : le relais.....	19
Figure II.9 : ventilateur DC 5v.....	21
Figure II.10 : servomoteur.....	21
Figure II.11 : structure d'un programme sous IDE.....	24
Figure II.12 : Matlab interface.....	26
Figure II.13 : schéma par fritzing.....	27

## Chapitre III : ----- Réalisation pratique -----

<b>Figure III.1</b> : plante de tomate.....	29
<b>Figure III.2</b> : la serre de tomate.....	30
<b>Figure III.3</b> : montages des capteurs et des actionneurs.....	30
<b>Figure II.4</b> : programme Arduino IDE (partie des capteurs et des actionneurs.....	31
<b>Figure II.5</b> : Arduino IDE (communication avec Matlab) .....	32
<b>Figure II.6</b> : Arduino IDE (communication avec site web).....	33
<b>Figure II.7</b> : interface.m.....	34
<b>Figure II.8</b> : interface.fig.....	35
<b>Figure II.9</b> : interface d'une serre actuelle.....	36
<b>Figure II.10</b> : interface avec régulation.....	37
<b>Figure II.11</b> : moniteur série.....	38
<b>Figure II.12</b> : site web lorsqu'on lance.....	39
<b>Figure II.13</b> : site web avec régulation.....	40
<b>Figure II.14</b> : récupération des données.....	41
<b>Figure II.15</b> : moniteur série.....	41
<b>Figure II.16</b> : contrôle des actionneurs.....	42
<b>Figure II.17</b> : moniteur série.....	43
<b>Figure II.18</b> : modifie les données avec des messages.....	44
<b>Figure II.19</b> : moniteur série.....	45

# LISTE DES TABLEAUX

<b>Tableau II.1</b> : caractéristiques de capteur d'humidité YL-69.....	17
---	----

## **Introduction Générale :**

Contrairement aux animaux, les plantes sont immobiles. En conditions de plein champ, elles sont soumises à des conditions climatiques qui peuvent varier beaucoup. Les plantes ne peuvent se soustraire aux vents violents aux températures extrêmes aux pluies, aux développements de maladies et de ravageurs.....etc. Les agriculteurs ont certes quelques moyens d'action sur l'environnement : ils peuvent planter des brise-vents installé des réseaux de drainage et d'irrigation apporter des engrais, traiter contre les maladies et les ravageurs, qui menacent leurs cultures. Ils peuvent aussi utiliser des cultivars résistants aux stress, aux maladies et aux ravageurs. Mais leurs moyens d'action restent limités, ne serait-ce que les techniques performantes sont difficiles à rentabiliser pour la culture de plein champ. Le produit agricole n'atteint habituellement pas de prix suffisants pour justifier les dépenses et les investissements très élevés, ce qui autorise et encourage l'investissement dans des outils de maîtrise d'environnement autrement dit : La production sous serre.

Cependant la serre, à l'origine conçue comme simple enceinte limitée par une paroi transparente, amplifie certaines caractéristiques du climat extérieur, créant alors des conditions favorables aux cultures. Mais le métier de producteur sous serre est l'un des plus exigeants qui soit. Les producteurs sous serre doivent savoir tout à la fois. L'acte de produire apparaît lui-même incroyablement complexe. Les producteurs sont confrontés, dans la pratique journalière de leur activité, à des problèmes qui mettent en jeu les interactions du climat, de l'irrigation fertilisante, des cultures, des agents pathogènes....etc. Le développement d'outil d'aide à la décision, repose sur des modèles, devrait faciliter la tâche de producteur à l'avenir.

Cette mémoire est structuré en trois chapitres, le premier chapitre est la modélisation de la croissance de la plante application à la tomate, dans ce chapitre nous parlons d'une serre de tomate et ses besoins (température et humidité). Le deuxième chapitre est réservé pour la présentation des matérielles et les logiciels, dans ce chapitre nous parlons des matériaux dont nous avons besoin et le rôle de chaque composant et les logiciels que nous avons utilisé. Le troisième chapitre est une réalisation pratique, dans ce chapitre on va parler de notre prototype qu'on a réalisé, ce prototype à deux parties, une partie locale et une partie à distance.

# Chapitre I

*Modélisation de la croissance de la plante*

*Application à la tomate*

## **I.1. Introduction :**

Durant l'année la production agricole est très demandée. Elle nécessite un milieu ayant des conditions climatiquement favorable à la production. Pour avoir une production assez riche malgré les conditions climatiques locales ou régionales non favorables, il est nécessaire de s'orienter vers l'exploitation de la production sous serre. Ceci demande un système de régulation et de contrôle des paramètres climatiques à l'intérieur de la serre. La supervision climatique à l'intérieur d'une serre agricole a une grande importance pour la vie de la plante.

En utilisant le système d'irrigation automatisé basé sur des capteurs constitue une solution prometteuse pour les agriculteurs lorsque la présence de l'agriculteur dans les champs n'est pas obligatoire. Dans cette catégorie l'agriculteur a la possibilité de piloter à distance le système d'irrigation et le taux d'humidité ainsi que la température de son exploitation agricole quel que soit sa position géographique et le temps au moyen de son téléphone mobile suivant une distribution d'une chaîne d'équipement intelligente. La serre agricole permet également de collecter et diffuser en temps réel et à distance les données météorologiques notamment la température, le taux d'humidité du sol et l'air. Ce procédé permet à l'exploitant un gain de temps et d'énergie [1].

## **I.2. Mesures utilisées par le système de régulation :**

Une serre est une clôture, dans laquelle on peut contrôler son environnement et le modifier pour améliorer ses préférences et les rendre plus convenables pour augmenter le rendement, élargir la période de production et pour stabiliser des paramètres climatiques tels que la température, l'humidité. Beaucoup de travaux ont été réalisés sur le développement des modèles de la serre. En général, ces modèles sont basés sur des modèles physiques à partir de mesures réelles enregistrées par les capteurs dans la serre agricole. L'objectif de ce travail est d'instrumenter la serre par utilisation de moyens technologiques plus modernes pour contrôler la température et l'humidité à l'intérieur de la serre [2].





**Figure I.1** : une serre intelligente.

### **I.3. Intérêts de la détermination du microclimat :**

Pour répondre à ces besoins, de nouveaux outils et techniques de production ont été développés. L'outil serre et ses équipements de chauffage permettent de mieux gérer la croissance des plantes car la gestion du climat est maîtrisée, l'un des importantes températures et humidité.

#### **I.3.1. La température :**

Le contrôle adéquat de la température est le facteur le plus important pour garantir une bonne croissance des plantes. En effet, la température est l'un des paramètres le plus important dans la gestion du climat, et aussi le plus difficiles à gérer. La température optimale diffère d'une culture à une autre et selon le stade de la culture elle influe sur plusieurs phénomènes bioénergétiques (respiration, photosynthèse,...) Toute grande déviation de la température par rapport aux températures typiques d'une plante provoque un ralentissement de la croissance et la détérioration dans certains cas, de ce fait on note l'importance de ce paramètre dans les systèmes de gestions des serres agricoles [3].

### I.3. 2. L'humidité :

L'humidité est la présence d'eau ou de vapeur d'eau dans l'air ambiant, elle influe sur la croissance de la plante et sur le développement des maladies. En pratique quand on parle de la mesure de l'humidité on fait allusion au taux d'humidité exprimé en % ce qui représente l'humidité relative. L'humidité de l'air et la température présentent une grande relation. En effet la capacité de l'air à retenir la vapeur d'eau double chaque fois que la température augmente d'une dizaine de degrés. Maintenir un niveau d'humidité et de température optimal dans une serre permet d'avoir un bon rendement de croissance. Le contrôle de tous ces paramètres se fait via de nombreux dispositifs des plus simples au plus complexes tels que : les ventilateurs extracteurs, les chauffages, vapeur d'eau [4].

### I.4. Exemples des serres agricoles :

Le but majeur de notre projet est la culture de tomate, la tomate d'origine tropicale a des exigences particulières : est sensible au froid, craint beaucoup le gel, craint les vents chauds et très exigeants en température.



Figure I.2 : Culture de tomate.

### **I.4.1. Les exigences climatiques de la tomate :**

#### **I.4.1. 1. Température de la tomate :**

La température est le facteur le plus déterminant dans la production de la tomate .Celle-ci réagit énormément aux variations thermiques.

La gestion des températures est une tâche difficile, car elles n'influent pas seulement sur la photosynthèse, elles influent également sur la respiration et la translocation. Il est important de noter que la température joue de rôle totalement différent : la température favorise la respiration (perte de matière sèches : activité à minimiser), et la photosynthèse (production de sucre : activité à maximiser) [5].

- **Les basses températures (<10 °C) :**

Ralentissent la croissance et le développement des plantes entraînant :

- La formation d'un feuillage abondant au détriment de la production.
- Une température basse peut entraîner aussi des ramifications des bouquets.
- Difficultés de nouaison et formation des fleurs fasciées.

- **Au-dessous de (17°C) :**

-Le pollen germe mal

- **Au-dessous de (30°C) :**

Le lycopène, pigment responsable de la couleur rouge de fruit ne se forme plus. C'est le pigment bêta-carotène qui se forme donnant ainsi une coloration jaune-orange au fruit.

Et d'après (Madripem, Derd.1999) La température optimale est :

-Températures diurnes (jour) : 20-25°C.

#### **I.4.1.2. Humidité du sol :**

Une humidité relative du sol de 75% est jugée optimale. Elle permet d'avoir des fruits de

bons calibres, et sans défaut de coloration. Une HR trop élevée, couplée à une température élevée, entraîne une végétation luxuriante. Elle favorise aussi le développement des maladies [6].

Donc l'humidité optimale est :

-L'humidité du sol : 75%.

#### **I.4.1.3. Humidité d'air :**

Une humidité relative de l'air de 75% convient à tous les stades de développement. Elle doit être surtout respectée au moment de la floraison où l'on peut craindre, par temps sec, une mauvaise réceptivité des stigmates et par hygrométrie excessive, une dissémination insuffisante du pollen.

#### **I.5. Gestion du climat :**

La maîtrise du climat est la raison des serres, on peut créer un environnement idéal pour la croissance des plantes.

#### **I.5.1. Gestion de température et d'humidité :**

L'humidité de l'air est directement liée à la température régnant dans la serre. Plus elle est élevée, plus l'air peut être humide. Dans ce cas le réseau de capteur qu'on utilise jouera le rôle d'un régulateur de température et d'humidité: système qui régule la température de certaines zones en allumant et éteignant les appareils de chauffage et de refroidissement et en fonction de la température et de l'humidité. Il peut être connecté au ventilateur, chauffage... Ces appareils seront allumés et éteints selon les besoins. Automatique abordée dans cet article est basée sur le contrôleur Arduino.

#### **I.5.1. 1. Réseau de capteur de température et humidité d'air :**

En ce qui concerne l'air dans la serre il suffit de gérer sa température, et le réseau de capteur utilisé est structuré autour de :

- Capteur de température et humidité de l'air.

- LED (représentant le chauffage).
- Ventilateur (unité de refroidissement).
- servomoteur (ouvrir fenêtre).
- Microcontrôleur Arduino.

### **I.5.1.2. Réseau de capteur de l'humidité du sol :**

Concernant le sol, il suffit de gérer le taux d'humidité privilégiant dans le sol de la serre, pour ceci le réseau utilisé est structuré autour de :

- Capteur d'humidité de sol.
- Eau (électrovanne).
- Microcontrôleur Arduino.

### **I.5.1.3. Moyens de communication :**

- Site web
- GSM
- Interface (sous Matlab)

## **I.6. Conclusion :**

Il s'agit d'une technologie moderne de production contre saison à l'aide d'un contrôle climatique et de l'utilisation de matières améliorant la productivité. L'usage des serre multi-chapelle permet également d'assurer des conditions de croissance optimales, de mieux organiser la profession, de combattre les parasites et les maladies Dans ce chapitre introductif nous avons présenté sommes basés sur la culture sous serre de tomate en déterminant les paramètres et les exigences climatiques agissants sur leurs croissances.

# Chapitre II

*Présentation des matériels et les logiciels*

## **II.1. Introduction :**

La réalisation pratique de notre projet comporte les deux aspects : le côté matériel et logiciel. Dans ce chapitre on présente chaque composant utilisé en précisant son rôle et son fonctionnement.

## **II.2. Partie matériel :**

### **II.2.1. La carte Arduino :**

Ce sont des cartes électroniques programmables (donc dotées d'un processeur et de mémoire) sur lesquelles nous pouvons brancher des capteurs de température, d'humidité,.... Une carte Arduino est un cerveau qui permet de rendre intelligent des systèmes électroniques et d'animer des dispositifs mécaniques.

#### **II.2.1.1. Types des cartes Arduino :**

Il existe plusieurs types des cartes Arduino, on trouve les originaux et les dérivés (compatibles avec Arduino). Parmi les plus utilisées dans le monde des systèmes embarqués on a : la UNO, la LEONARDO, la DUE, la MEGA et sa petite amélioration la MEGA 2560, et la Yun.....

Notre choix s'est porté sur la carte UNO, car ses performances sont largement suffisantes pour gérer le fonctionnement de notre système.

#### **II.2.1.2. Carte Arduino Uno :**

La carte à microcontrôleur Arduino UNO est la première version stable de carte Arduino. Elle utilise un microcontrôleur AVR d'architecture ATmega328p cadencé à 16Mhz. Elle possède 32ko de mémoire flash destinée à recevoir le programme, 2ko de SRAM (mémoire vive) et 1 ko d'EEPROM (mémoire morte destinée aux données). Elle contient 14 broches d'entrée/sortie numériques dont 6 peuvent être utilisées comme sorties PWM (Modulation

à largeur d'impulsions) et elle permet aussi de mesurer des grandeurs analogiques grâce à ces 6 entrées analogiques [7].

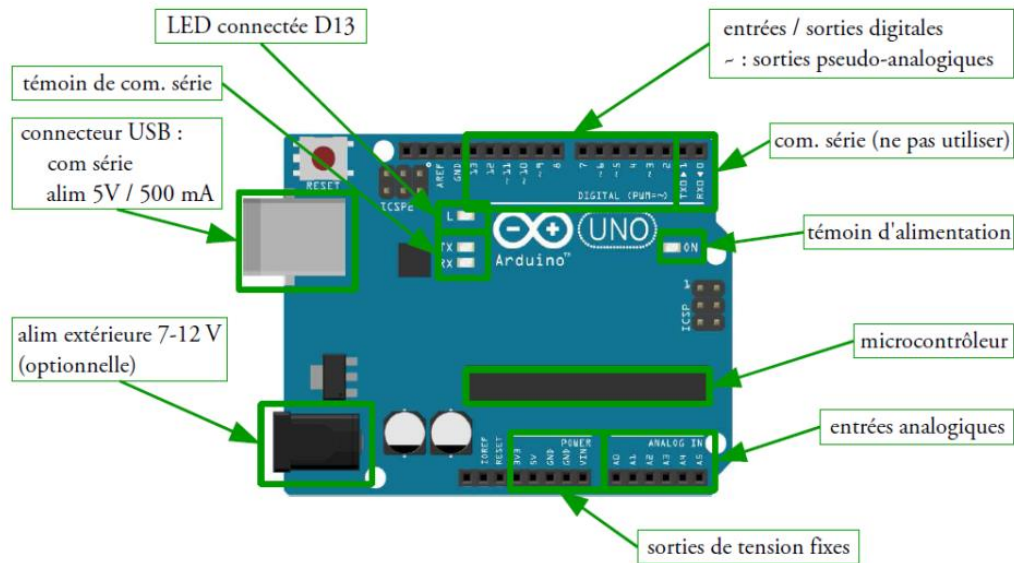
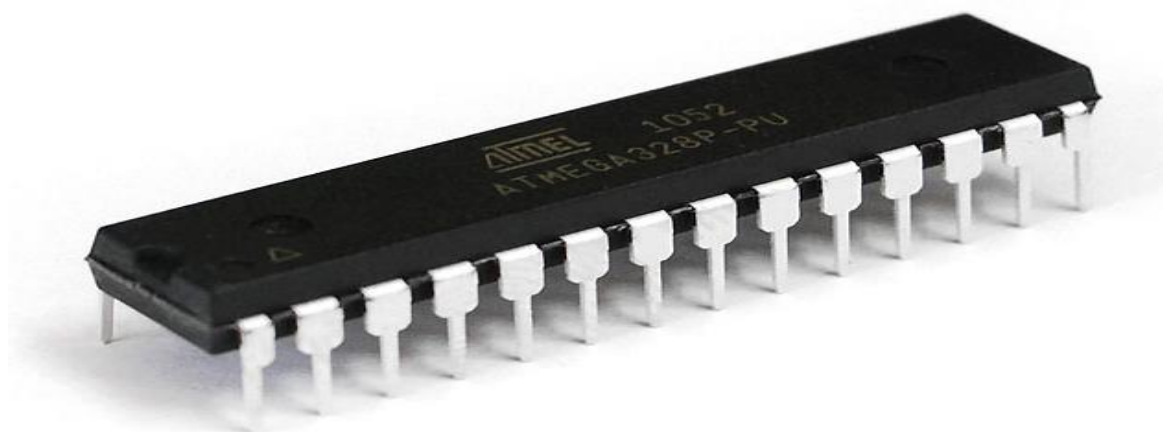


Figure II.1 : Carte Arduino UNO.

### II.2.1.3. Microcontrôleur :

Le modèle UNO de la société Arduino est une carte électronique dont le cœur est un microcontrôleur ATMEL de référence ATmega328. L'ATmega328 est un microcontrôleur 8bits de la famille AVR dont la programmation peut être réalisée en langage C/C++. Microcontrôleur un circuit intégré qui rassemble sur une puce plusieurs éléments dans un espace réduit, qui est un système à microprocesseur contenant des périphériques intégrés tel mémoire données, des programmes pouvant être utilisés comme un système embarqué [8].





**Figure II.2 :** Microcontrôleur ATmega328.

Principales caractéristiques sont :

**-FLASH** = mémoire programme de 32Ko

**-SRAM** = données (volatiles) 2Ko

**-EEPROM** = données (non volatiles) 1Ko

**-Digital I/O (entrées-sorties Tout Ou Rien)** = 3 ports PortB, PortC, PortD (soit 23 broches en tout I/O)

**-Timers/Counters** : Timer0 et Timer2 (comptage 8 bits), Timer1 (comptage 16bits)  
Chaque timer peut être utilisé pour générer deux signaux PWM. (6 broches OCxA/OCxB)

**-Plusieurs broches multifonctions** : certaines broches peuvent avoir plusieurs fonctions différentes, choisies par programmation. Elles ont alors plusieurs noms sur le brochage  
Par exemple, les broches PB1, PB2, PB3, PD3, PD5, PD6 peuvent servir de sortie PWM (Pulse Width Modulation), c'est-à-dire des sorties qui joueront le rôle de sorties analogiques.

**-PWM** = 6 broches OC0A(PD6), OC0B(PD5), OC1A(PB1), OC1B(PB2), OC2A(PB3), OC2B(PD3)

**-Analog to Digital Converter (résolution 10bits)** = 6 entrées multiplexées ADC0(PC0) à ADC5(PC5)  
**Gestion bus I2C** = le bus est exploité via les broches SDA(PC5)/SCL(PC4).

**-Port série (USART)** = émission/réception série via les broches TXD(PD1)/RXD(PD0)

**-Comparateur Analogique** = broches AIN0(PD6) et AIN1 (PD7) peut déclencher interruption

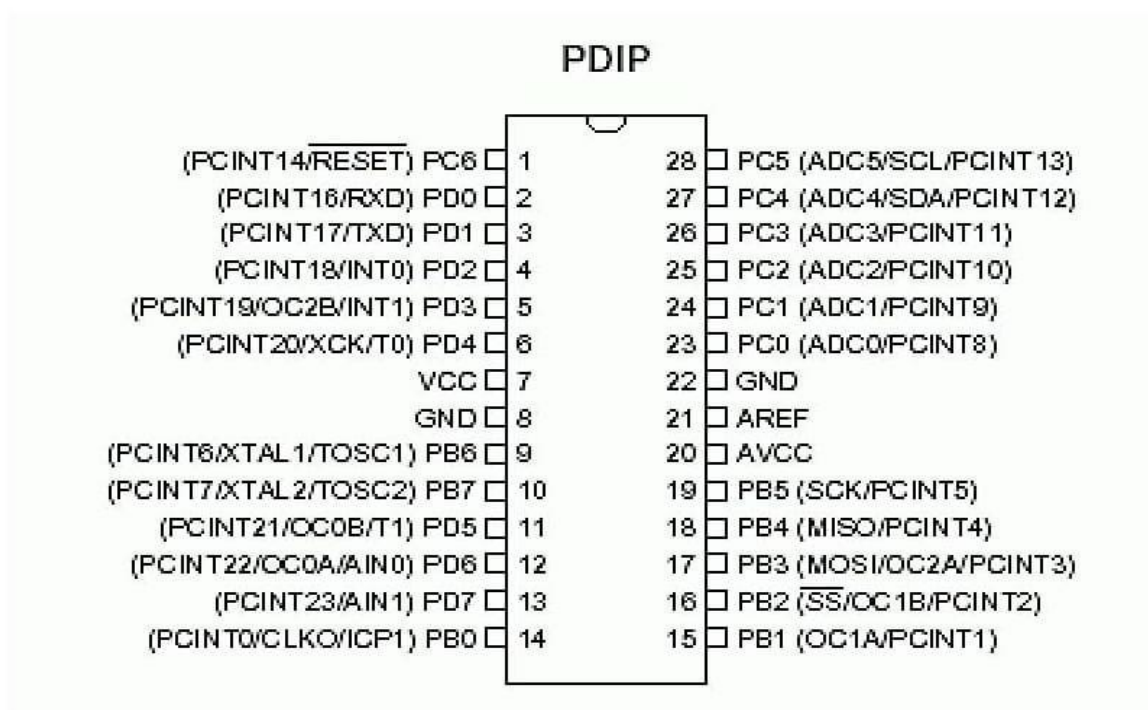


Figure II.3 : Schéma ATmega328.

#### II.2.1.4. Entrées/Sorties numériques d'une carte arduino:

Cette carte possède 14 broches numériques (numérotée de 0 à 13) qui sont programmables par le biais d'instructions `pinMode()`, `digitalWrite()` et `digitalRead()`. Ces broches fonctionnent à 5V, chacune peut fournir ou recevoir un courant maximal de 40mA disposant d'une résistance interne de (rappel au plus) (pull-up) (déconnectée par défaut) de 20-50KOhms. Cette résistance interne s'active sur une broche en entrée à l'aide de l'instruction `digitalWrite(broche, HIGH)`.

#### II.2.1.5. Entrées analogiques d'une carte arduino:

La carte UNO dispose de 6 entrées analogiques (A0 à A5), où chacune peut fournir une mesure avec fonction `analogRead()` du langage Arduino sur une résolution de 10 bits (de 0 à 1023). Par défaut, ces broches mesurent une tension comprise entre le 0V correspondant au niveau 0 et le 5V correspondant au niveau 1023. Notons qu'il est

possible de modifier le niveau supérieure de la plage de mesure en modifiant la tension sur la broche AREF ou en utilisant l'instruction analog Reference ( ) du langage Arduino.

## II.2.2. Composants électroniques :

### II.2.2.1. GSM module sim800l :

#### II.2.2.1.1. Description :

Le module SIM800L est un petit module GSM / GPRS et idéal pour les petits projets idéaux. Le module prend en charge le réseau GSM / GPRS quadri-bande, disponible pour la transmission à distance de données de messages SMS et GPRS. Prend en charge les commandes telles que 3GPP TS 27.007, 27.005 et les commandes AT améliorées SIMCOM. Le SIM800L peut fonctionner jusqu'à 2 ampères de courant en pointe. Il dispose également d'une fonction de faible consommation d'énergie qui consomme 1 mA de courant en mode veille. Vous devez alimenter le module de 3,7 V à 4,2 V. Plus que cela endommagerait le module. Nous avons donc utilisé un convertisseur de tension pour diminuer la tension vers 4v.

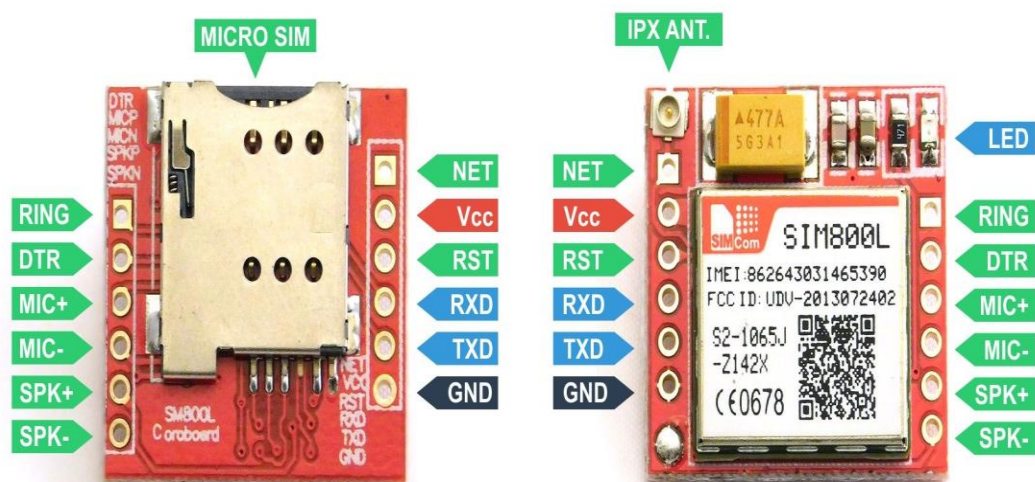


Figure II.4 : Module Gsm sim800l.

### II.2.2.1.2. Caractéristiques :

- 2G quadri-bande 850/900/1800 / 1900MHz
- Recevez et passez des appels à l'aide des sorties haut-parleur et microphone
- Recevoir et envoyer des SMS
- Écoutez les émissions de radio FM
- Connectivité GPRS multi-slot classe12: max. 85,6 kbps (téléchargement / téléchargement)
- Station mobile GPRS classe B
- Contrôlé par AT Command (3GPP TS 27.007, 27.005 et SIMCOM Enhanced AT Commands)
- Prend en charge l'horloge en temps réel
- Plage de tension de fonctionnement 3,7 V ~ 4,2 V
- Prend en charge A-GPS
- Faible consommation d'énergie, 1mA en mode veille
- Carte micro SIM

### II.2.2.1.3. Le fonctionnement:

La fonction de module dans ce projet est la communication avec le site web pour récupérer et recevoir les données (Température et humidité), elle nous permet aussi de gérer et de prendre control de notre serre.

### II.2.2.2. Capteur d'humidité YL-69 :

#### II.2.2.2.1. Description :

Le capteur d'humidité du sol (YL-69) est généralement utilisé pour détecter l'humidité du sol. Ainsi, il sert à construire un système d'arrosage automatique pour surveiller l'humidité du sol des plantes. Le capteur est configuré en deux parties: la carte électronique et la sonde à deux électrodes qui détecte la teneur .Ce dernier dispose d'un potentiomètre intégré pour le réglage de la sensibilité de la sortie numérique (D0), d'une LED d'alimentation et d'une LED de sortie numérique.

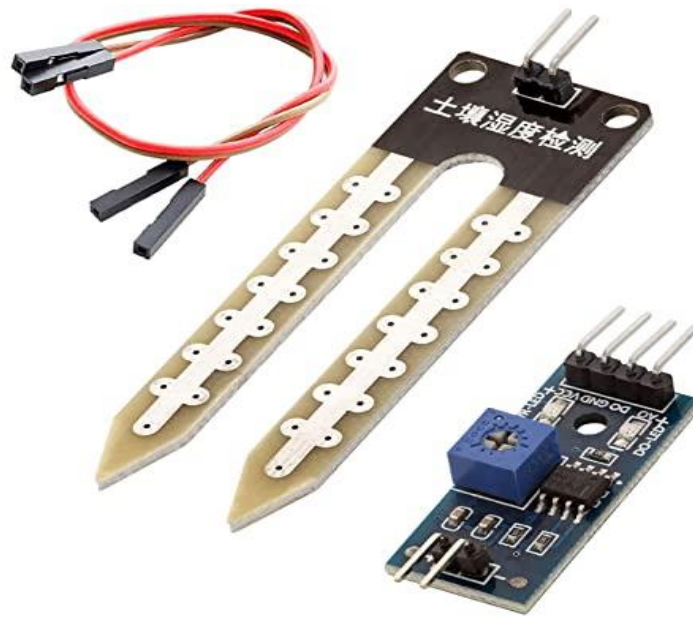


Figure II.5 : Capteur y1-69.

#### II.2.2.2.2 Branchement sur la carte Arduino :

Pour exploiter le capteur y1-69 il suffit :

- De relier la sonde a la carte électronique avec deux files.
- D'alimenter les pattes VCC et GND de la carte.
- De brancher les pattes au milieu de la carte a des entrée analogique (A0...A5) et des E/S digitales (0..13) de la carte Arduino.

#### II.2.2.2.3. Caractéristiques du capteur d'humidité :

Le tableau II.1 résume les caractéristiques du capteur d'humidité YL-69.

VCC	3,3V ou 5V
GND	GND
AO	0 - 200 dans l'eau 200 -400 milieu d'arrosage parfait 400 – 700 milieu d'arrosage moyen 700_ 950 sol sec 1023 dans l'air
DO	0 OU 1
taille	60×20×5MM
Courant	35 MA

**Tableau II.1 :** Caractéristiques de capteur d'humidité YL-69

### **II.2.2.3. Capteur de température et d'humidité DHT22 :**

#### **II.2.2.3.1. Description :**

Le DHT22 est un capteur de température et d'humidité numérique. Il utilise un capteur d'humidité capacitif et une thermistance pour mesurer l'air environnant et émet un signal numérique sur la broche de données. Il est assez simple à utiliser. La lecture de ce capteur peut durer jusqu'à 2 secondes.



**Figure II.6 :** Capteur de température et d'humidité DHT22.

Il suffit de connecter la première broche à gauche à l'alimentation de 3 à 5 V, la deuxième broche doit être branchée à la carte Arduino. La troisième broche est montée à la masse GND.

#### **II.2.2.3.2. Les principales caractéristiques de DHT22 :**

- Puissance et E / S de 3 à 5 V.
- 2.5 mA max courant d'utilisation pendant la conversion (lors de la demande de données).
- Bon pour 0-100% de lectures d'humidité avec 2-5% de précision.
- Bon pour les lectures de température de -40 à 80 ° ± Précision ± 0,5 ° C.
- Pas plus de 0,5 Hz de fréquence d'échantillonnage (une fois toutes les 2 secondes).
- Taille du corps 27 mm × 59 mm × 13.5 mm (1.05 "× 2.32" × 0.53 ").
- Poids : 2,4 g.

#### **II.2.2.4. Pompe d'arrosage :**

Une pompe est un dispositif permettant d'aspirer et de refouler l'eau. Pour notre prototype, nous avons utilisé une pompe d'une machine à laver alimentée par une source de tension 220v. Le rôle de pompe est l'arrosage de la plante lorsque l'humidité de sol augmente.



**Figure II.7 :** Pompe d'une machine à laver

### II.2.2.5. Le relais :

Le relais joue le rôle d'un interrupteur commandé. Un relais électromécanique est doté d'un bobinage en guise d'organe de commande. La tension appliquée à ce bobinage va créer un courant, ce courant produit un champ électromagnétique à l'extrémité de la bobine (électro-aimant). Ce champ magnétique va être capable de faire déplacer un élément mécanique métallique monté sur un axe mobile, qui déplacera alors des contacts électriques. La fonction de relais est quand il reçoit un signal qu'il fonction comme un interrupteur. Nous avons besoin de deux relais pour notre projet (pompe et ventilateur).



**Figure II.8 :** Le relais



Les principales caractéristiques techniques du relais :

- Signal de commande 5V.
- Maximum AC courant et voltage : 10A 250VAC.
- Maximum DC courant et voltage : 10A 30VDC.
- Il existe un contact normalement ouvert et un contact normalement fermé.
- Pour que la bobine du relais soit excitée, vous devez avoir une entrée de 0 dans la broche de signal.

#### **II.2.2.5.1. Brochage du relais :**

- + : 5V source.
- - : Masse.
- S : Signale de commande.
- NC : normalement fermé.
- NO : normalement ouvert.

#### **II.2.2.6. Ventilateur :**

Un ventilateur est un appareil destiné à créer un vent artificiel, comme son nom l'indique, autrement dit un ventilateur est un appareil mettant de l'air en mouvement. La fonction du ventilateur est réduire la température d'aire.



**Figure II.9 :** Ventilateur DC 5V

### II.2.2.7. Le servomoteur :

Un servomoteur est un moteur électrique spécifique pour un déplacement angulaire. Nous avons permis d'abord le pilotage des servomoteurs avec Arduino. Un servomoteur permet de se positionner précisément sur un angle. La plage de l'angle est généralement de 0 à 180 degrés. La fonction de servomoteur est d'ouvrir ou de fermer une fenêtre lorsque l'humidité de l'air augmente ou diminue.



**Figure II.10 :** Servomoteur.

- **Elément visible :**

-Les fils, qui sont au nombre de trois

- rouge : pour l'alimentation positive (4.5V à 6V en général).
- marron : pour la masse (0V).
- orange, G : entrée du signal de commande.

-L'axe de rotation sur lequel est monte un accessoire en plastique ou en métal.

-Le boîtier qui le protège.

- **Eléments non visibles :**

- Un moteur à courant continu.
- Des engrenages pour former un réducteur (en plastique ou en métal).
- Un capteur de position de l'angle d'orientation de l'axe (un potentiomètre bien souvent).
- Une carte électronique pour le contrôle de la position de l'axe et le pilotage du moteur à courant continu.

### **II.2.2.8. Le LED :**

Utilisé comme un chauffage lorsque la température est diminuée.

## **II.3. Partie logiciels :**

### **II.3.1. Arduino :**

#### **II.3.1.1. L'environnement de la programmation :**

Le logiciel de programmation de la carte Arduino sert d'éditeur de code (langage proche du C). Une fois, le programme tapé ou modifié au clavier, il sera transféré et mémorisé dans la carte à travers de la liaison USB. Le câble USB alimente à la fois en énergie la carte et transporte aussi l'information ce programme appelé IDE Arduino [07].

### II.3.1.2. Langage de programmation :

Un langage de programmation est un langage permettant d'écrire un ensemble d'instructions qui seront converties en langage machine grâce à un compilateur. L'avantage du langage Arduino est qu'il est basé sur les langages C/C++ qui supporte toutes les syntaxes standards du langage C et quelques-unes des outils du C++. En plus, la disponibilité des bibliothèques permettent de faciliter la communication avec le matériel connecté à la carte (carte réseaux GSM, capteurs, servomoteurs... etc.). Pour écrire un programme avec le langage Arduino, il faut respecter certaines règles. Notons que l'exécution d'un programme Arduino s'effectue d'une façon séquentielle, c'est-à-dire que les instructions sont exécutées les unes à la suite des autres. Avant tout le compilateur doit vérifier l'existence de deux structures obligatoires à tout programme en langage Arduino qui sont :

- la partie initialisation et configuration des entrées/sorties (la fonction `setup ()`).
- la partie principale qui s'exécute en boucle (la fonction `loop ()`).

### II.3.1.3. Structure générale d'un programme en langage Arduino :

Pour utiliser l'IDE standard Arduino (`arduino.exe`), il suffit de saisir le code dans la fenêtre dédiée, de compiler et de téléverser le programme sur la carte Arduino. La carte doit être reliée à l'ordinateur via un câble USB. Le modèle de la carte Arduino ainsi que le port série sur lequel elle est branchée doivent être déclarés dans le menu de l'IDE Outils/type de carte et Outils/port série. Une fois le programme compilé téléversé dans le microcontrôleur, son exécution sera lancée.

La fonction `setup ()` s'exécute une seule fois après la mise sous tension et après chaque redémarrage. Par contre la fonction `loop ()` s'exécute en boucle.

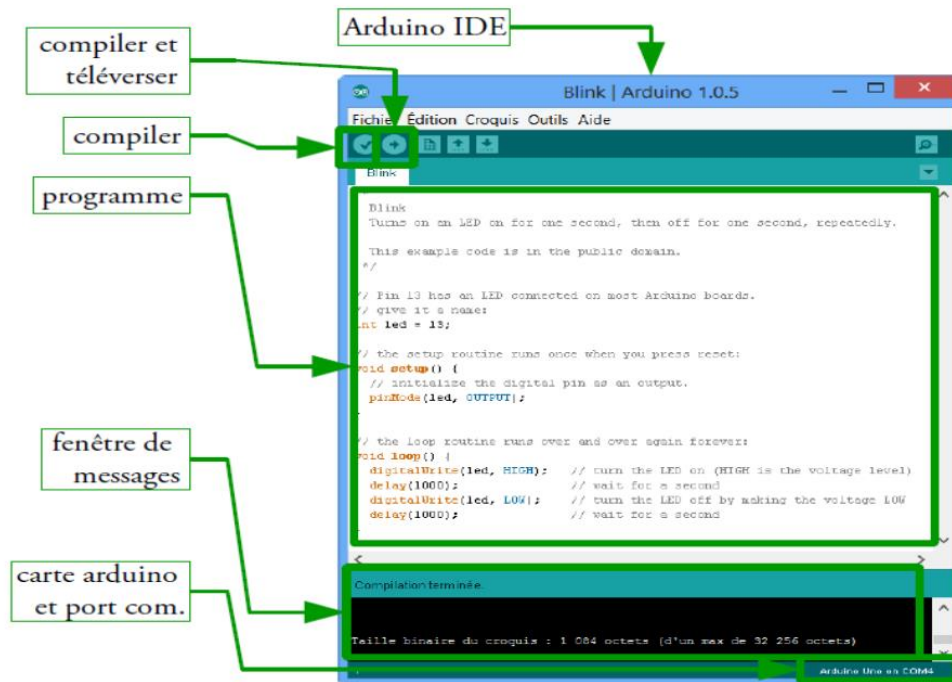


Figure II.11 : Structure d'un programme sous IDE.

Un programme Arduino comporte trois parties principales :

1. la partie déclaration des variables (optionnelle).
2. la partie initialisation et configuration des entrées/sorties : la fonction `setup()`.
3. la partie principale qui s'exécute en boucle : la fonction `loop()`.

Dans chaque partie du programme sont utilisées différentes instructions issues de la syntaxe du langage Arduino [08].

#### II.3.1.4. Etapes de l'implémentation sur le microcontrôleur :

- La création d'un projet.
- L'écriture du programme ensuite enregistrement.
- La vérification de la syntaxe et correction d'éventuelles erreurs.
- Le téléversement vers le microcontrôleur. [9]

## II.3.2. Matlab :

### II.3.2.1. Introduction :

MATLAB (« matrix laboratory ») est un langage de programmation de quatrième génération émulé par un environnement de développement du même nom ; il est utilisé à des fins de calcul numérique. Développé par la société The MathWorks, MATLAB permet de manipuler des matrices, d'afficher des courbes et des données, de mettre en œuvre des algorithmes, de créer des interfaces utilisateurs, et peut s'interfacer avec d'autres langages comme le C, C++, Java, et Fortran. Les utilisateurs de MATLAB (environ un million en 2004) sont de milieux très différents comme l'ingénierie, les sciences et l'économie dans un contexte aussi bien industriel que pour la recherche. Matlab peut s'utiliser seul ou bien avec des toolbox (« boîte à outils »).

### II.3.2.2. Interface graphique :

Les interfaces graphiques (ou interfaces homme-machine) sont appelées GUI (pour Graphical User Interface) sous MATLAB. Elles permettent à l'utilisateur d'interagir avec un programme informatique, grâce à différents objets graphiques (boutons, menus, cases à cocher...). Ces objets sont généralement actionnés à l'aide de la souris ou du clavier. MATLAB possède un outil dédié à la création des interfaces graphiques appelé GUIDE (pour Graphical User Interface Development Environment). Le GUIDE est un constructeur d'interface graphique qui regroupe tous les outils dont le programmeur a besoin pour créer une interface graphique de façon intuitive. Le GUIDE possède également des outils pour gérer l'alignement des objets et pour créer des barres d'outils ou des menus. Une fois l'interface graphique terminée, son enregistrement donne deux fichiers portant le même nom mais dont les deux extensions sont .fig et .m.

- Le fichier .fig contient la définition des objets graphiques (positions et propriétés). Ce fichier peut être ouvert ultérieurement avec le GUIDE pour modifier les objets graphiques.
- Le fichier .m contient les lignes de code qui assurent le fonctionnement de l'interface graphique (actions des objets). Ce fichier peut être édité dans le MATLAB Editor pour y

ajouter des actions à la main. C'est ce fichier qui doit être lancé pour utiliser l'interface graphique.

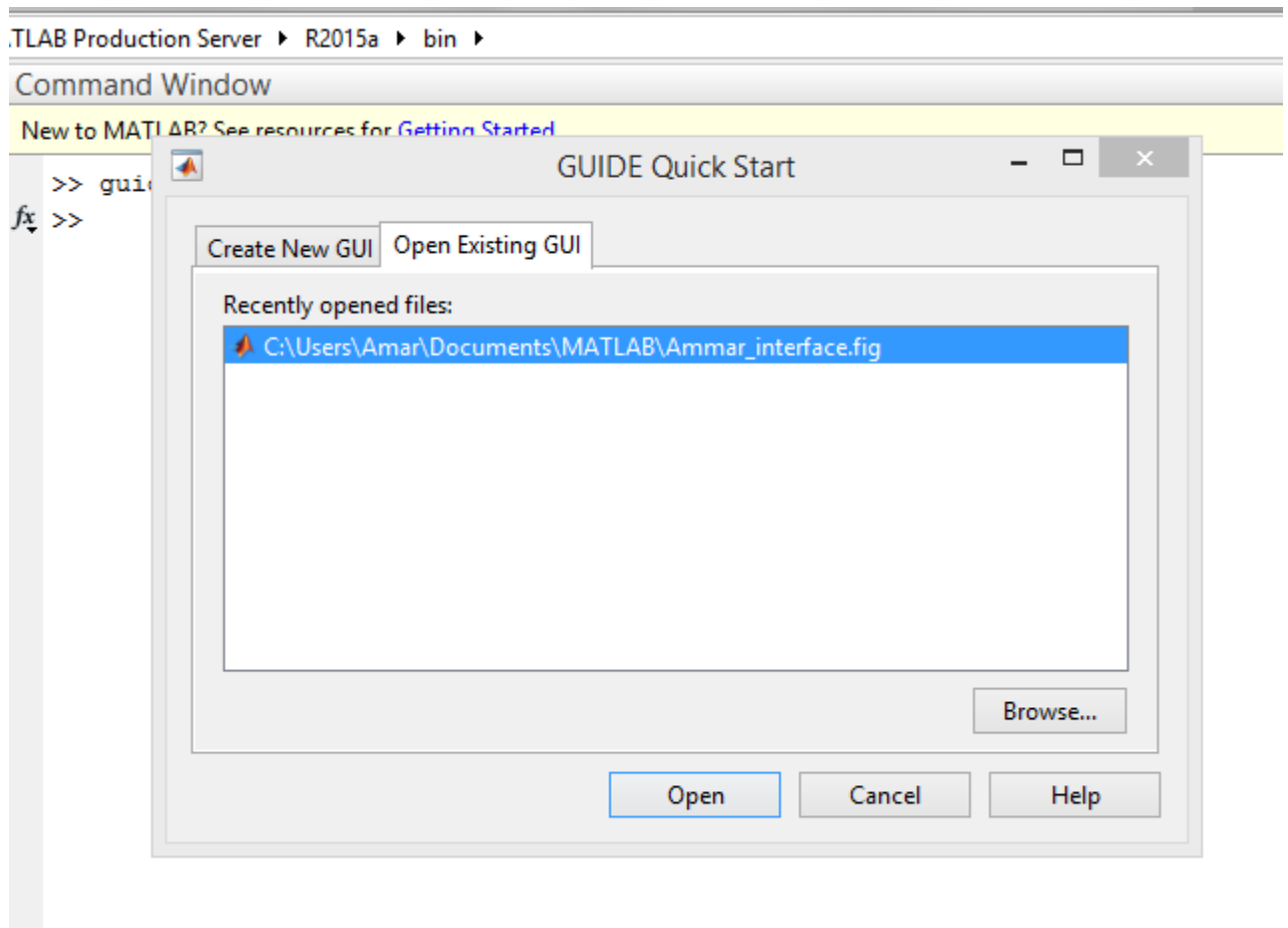


Figure II.12 : Matlab interface.

## II.4. Schéma de nos matériels par fritzing :

le schéma final :

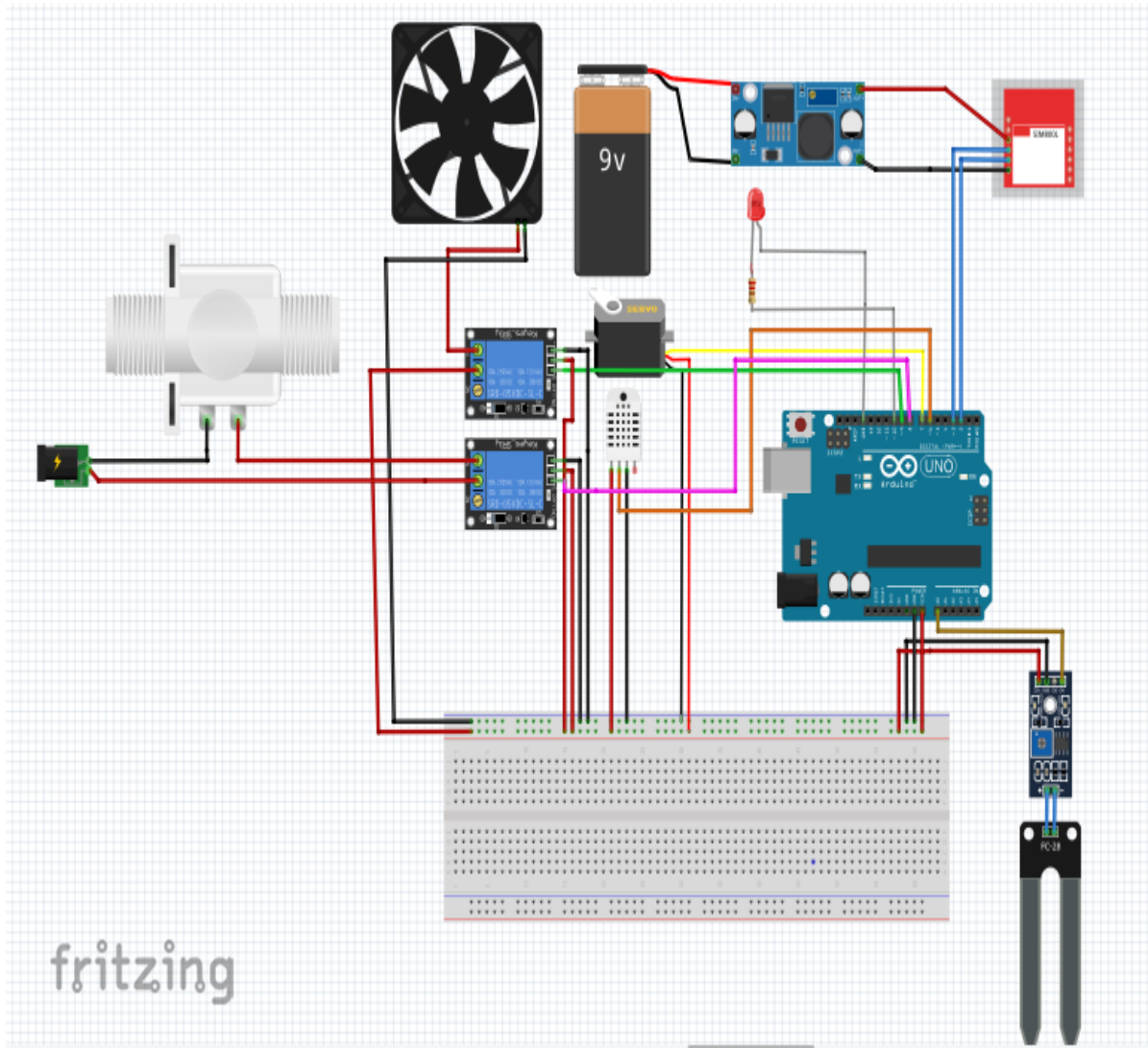


Figure II.13 : Schéma par fritzing

## II.5. Conclusion :

A travers ce chapitre nous avons présenté les composants utilisés et les logiciels. Ce système est composé d'une carte à microcontrôleurs Arduino (Uno), les capteurs : de température et d'humidité. Des actionneurs : pompe à eau, ventilateur et un servomoteur, et logiciel dont nous avons besoin.



# Chapitre III

## *Réalisation pratique*

### III.1. Introduction :

Dans ce chapitre, nous présentons en premier lieu, la réalisation de la serre de la structure jusqu'au placement des capteurs et les actionneurs et aussi programme Arduino. En seconde lieu, la partie local dans lequel la serre est contrôlé avec Matlab et nous affiche la température et l'humidité d'air et du sol en valeurs numérique et ces courbes, et aussi vous pouvez changer et contrôler les valeurs de température et d'humidité et voir l'état des sorties (ventilateur, pompe, fenêtre, chauffage).

En dernier lieu, la partie à distance est faite avec un site web qui affiche la température et l'humidité et aussi contrôle la serre par des messages.

### III.2. Réalisation pratique :

#### III.2. 1. Réalisation de la serre :

Nous avons plante un plant de tomate la serre puis installer les capteurs et les actionneurs.



Figure III.1: Plante de tomate.



Figure III.2: La serre de tomate.

### III.2.2. Montage des capteurs et les actionneurs :

La figure représente le montage en place

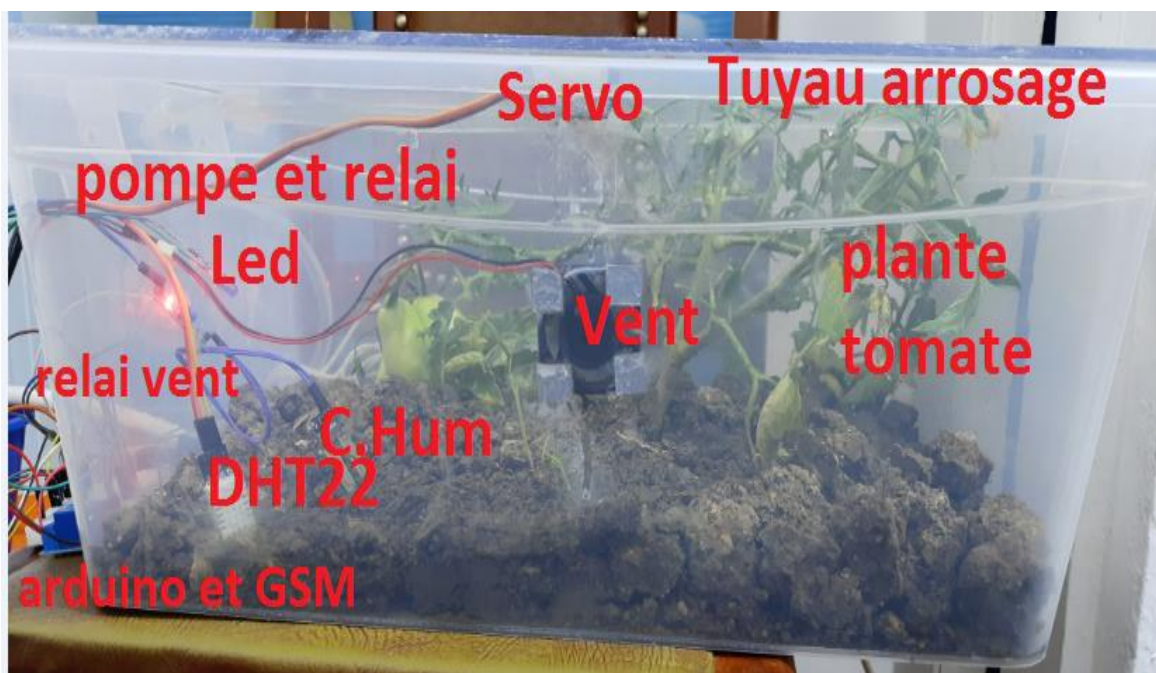
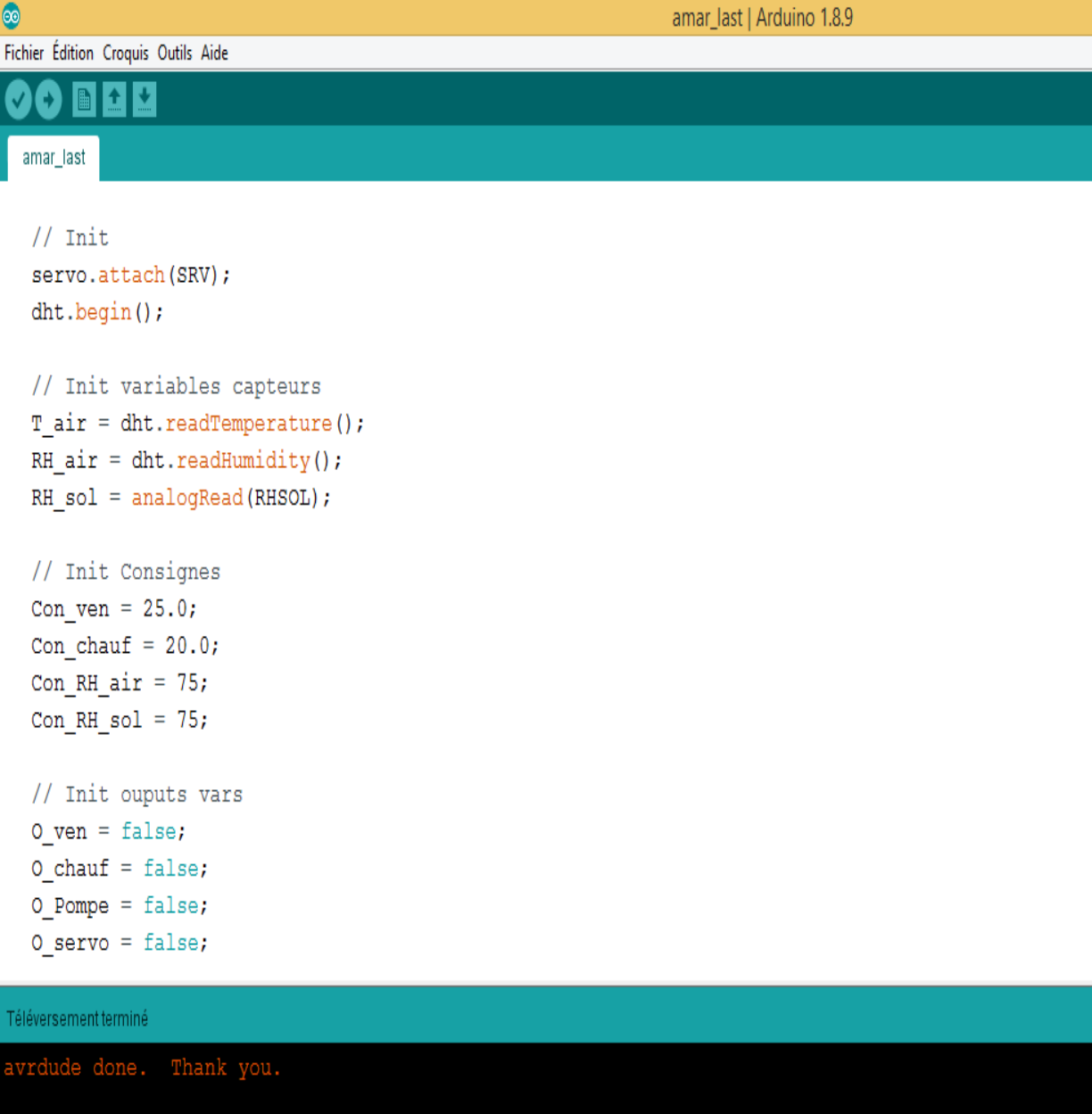


Figure III.3 : Montages des capteurs et des actionneurs.

### III.2.3. Programme Arduino IDE :

On a écrit un programme dans IDE divisé en 3 parties :

- Dans cette partie nous allons énoncer les capteurs et les actionneurs, et mettre les conditions de température et d'humidité nécessaire.



```
amar_last | Arduino 1.8.9
Fichier Édition Croquis Outils Aide
amar_last

// Init
servo.attach(SRV);
dht.begin();

// Init variables capteurs
T_air = dht.readTemperature();
RH_air = dht.readHumidity();
RH_sol = analogRead(RHSOL);


// Init Consignes
Con_ven = 25.0;
Con_chauf = 20.0;
Con_RH_air = 75;
Con_RH_sol = 75;

// Init ouputs vars
O_ven = false;
O_chauf = false;
O_Pompe = false;
O_servo = false;

Téléversement terminé
avrdude done. Thank you.
```

**Figure III.4 :** Programme Arduino IDE (partie des capteurs et des actionneurs).

- Dans cette partie on va programmer la communication avec Matlab.



```
amar_last |
Fichier Édition Croquis Outils Aide
amar_last
11 (Serial.available()) {
    Serial.readBytes(buff, 7);
    if (buff[0]==ID && buff[6]==CF) {
        switch (buff[1]) {
            case CMD1: res[0] = byte(T_air);
                       res[1] = byte(RH_air);
                       res[2] = byte(RH_sol);
                       res[3] = byte(O_ven);
                       res[4] = byte(O_chauf);
                       res[5] = byte(O_Pompe);
                       res[6] = byte(O_servo);
                       res[7] = byte(CF);
                       Serial.write(res, 8);
                       break;
            case CMD2: Con_ven = float(buff[2]);
                       Con_chauf = float(buff[3]);
                       Con_RH_air = int(buff[4]);
                       Con_RH_sol = int(buff[5]);
                       res[0] = byte(CF);
                       Serial.write(res[0]);
                       break;
            default: break;
        }
    }
}
```

Figure III.5 : Arduino IDE (communication avec Matlab).

- Dans cette partie on va faire la communication pour transférer les informations (température, humidité) au travers d'un site web.

```

ShowSerialData();

gprsSerial.println("AT+CIPSPRT=0");
delay(3000);

ShowSerialData();

gprsSerial.println("AT+CIPSTART=\"TCP\", \"api.thingspeak.com\", \"80\"); //start up the connection
delay(6000);

ShowSerialData();

gprsSerial.println("AT+CIPSEND"); //begin send data to remote server
delay(4000);
ShowSerialData();

String str="GET https://api.thingspeak.com/update?api_key=9RTDWD74M9Y5W7I&field1=" + String(T_air) + "&field2="+S
Serial.println(str);
gprsSerial.println(str); //begin send data to remote server

```

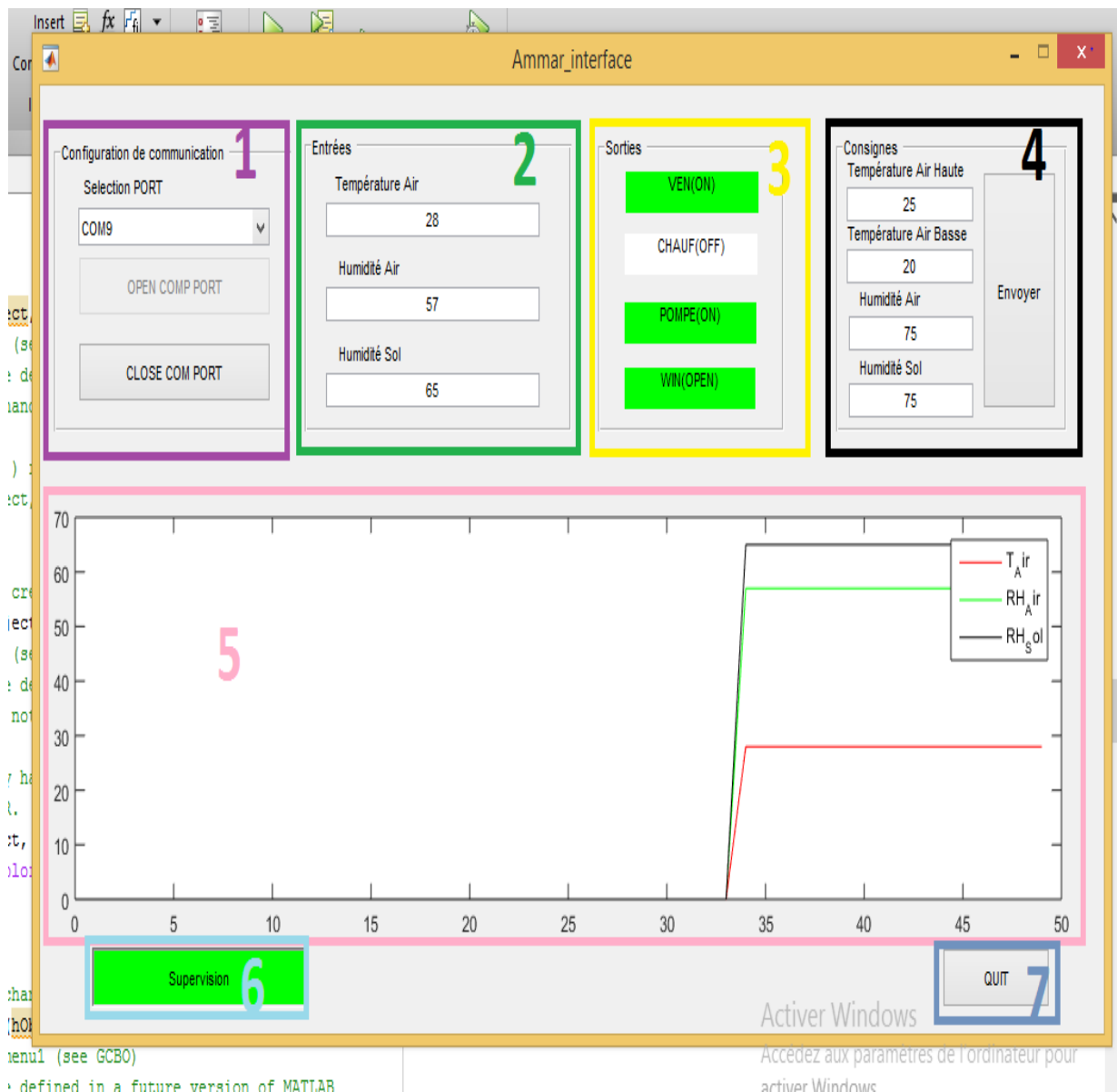
**Figure III.6 :** Arduino IDE (communication avec site web).

### III.3. Partie local :

Dans cette partie on utilise Matlab comme interface dans lequel on va faire la communication avec Arduino, en utilisant GUIDE

```
Ammar_interface.m x +
104 - pause(0.1);
105 - buffer = fread(COM, COM.BytesAvailable, 'uint8');
106 - % traitement de la réponse
107 - T_Air = double(buffer(1));
108 - RH_Air = double(buffer(2));
109 - RH_Sol = double(buffer(3));
110 - Etat_VON = double(buffer(4));
111 - Etat_CHAUF = double(buffer(5));
112 - Etat_POMPE = double(buffer(6));
113 - Etat_WIN = double(buffer(7));
114 - % update interface
115 - set(handles.edit1, 'String', num2str(T_Air));
116 - set(handles.edit2, 'String', num2str(RH_Air));
117 - set(handles.edit3, 'String', num2str(RH_Sol));
118 - if Etat_VON
119 -     set(handles.text6, 'String', 'VEN(ON)', 'BackgroundColor', 'g');
120 - else
121 -     set(handles.text6, 'String', 'VEN(OFF)', 'BackgroundColor', 'w');
122 - end
123 - if Etat_CHAUF
124 -     set(handles.text7, 'String', 'CHAUF(ON)', 'BackgroundColor', 'g');
125 - else
126 -     set(handles.text7, 'String', 'CHAUF(OFF)', 'BackgroundColor', 'w');
127 - end
128 - if Etat_POMPE
129 -     set(handles.text8, 'String', 'POMPE(ON)', 'BackgroundColor', 'g');
130 - else
131 -     set(handles.text8, 'String', 'POMPE(OFF)', 'BackgroundColor', 'w');
132 - end
133 - if Etat_WIN
```

Figure III.7 : Interface.m



**Figure III.8 :** Interface.fig

- 1- Configuration de communication : dans cette partie nous choisissons le port arduino (com), et il y a deux boutons open com et close com.
- 2- Entrée : cette partie affiche la température et d'humidité (d'aire et sol) actuelle d'une serre.
- 3- Sortie : affiche l'état des sorties (ventilateur, pompe, chauffage, fenêtre), s'il est ouvert ou fermé.
- 4- Consignes : ou on peut changer la température et l'humidité de la serre.
- 5- Les courbes : on peut visualiser les courbes de température et d'humidité.



- 6- Supervision : pour démarrer l'interface.
- 7- Quitter : un bouton pour quitter l'interface.

Après le téléversement de programme Arduino, on lance l'interface matlab.

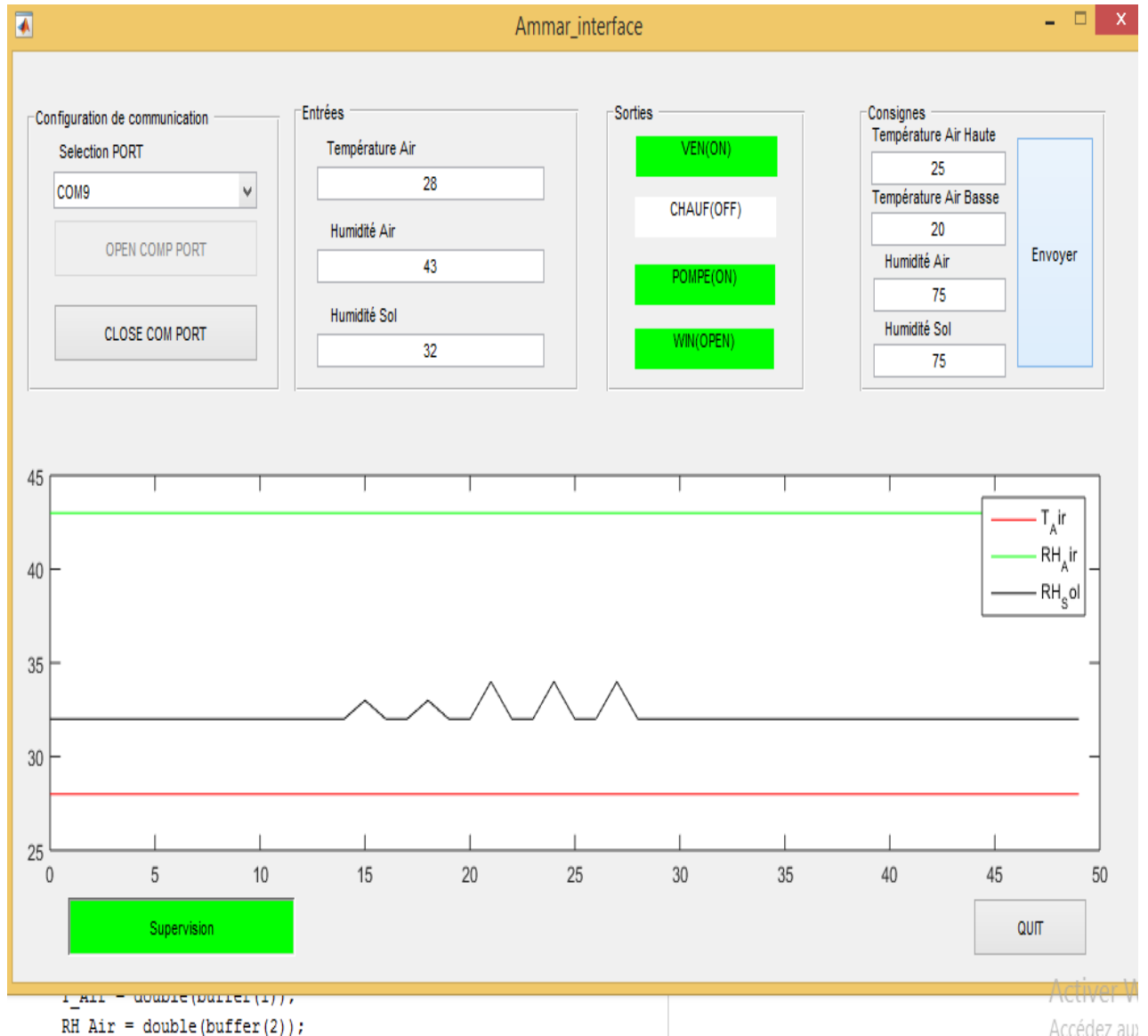


Figure III.9 : Interface d'une serre actuelle.

Après certain temps on voit la température et l'humidité stabilisé dans des conditions initiales.

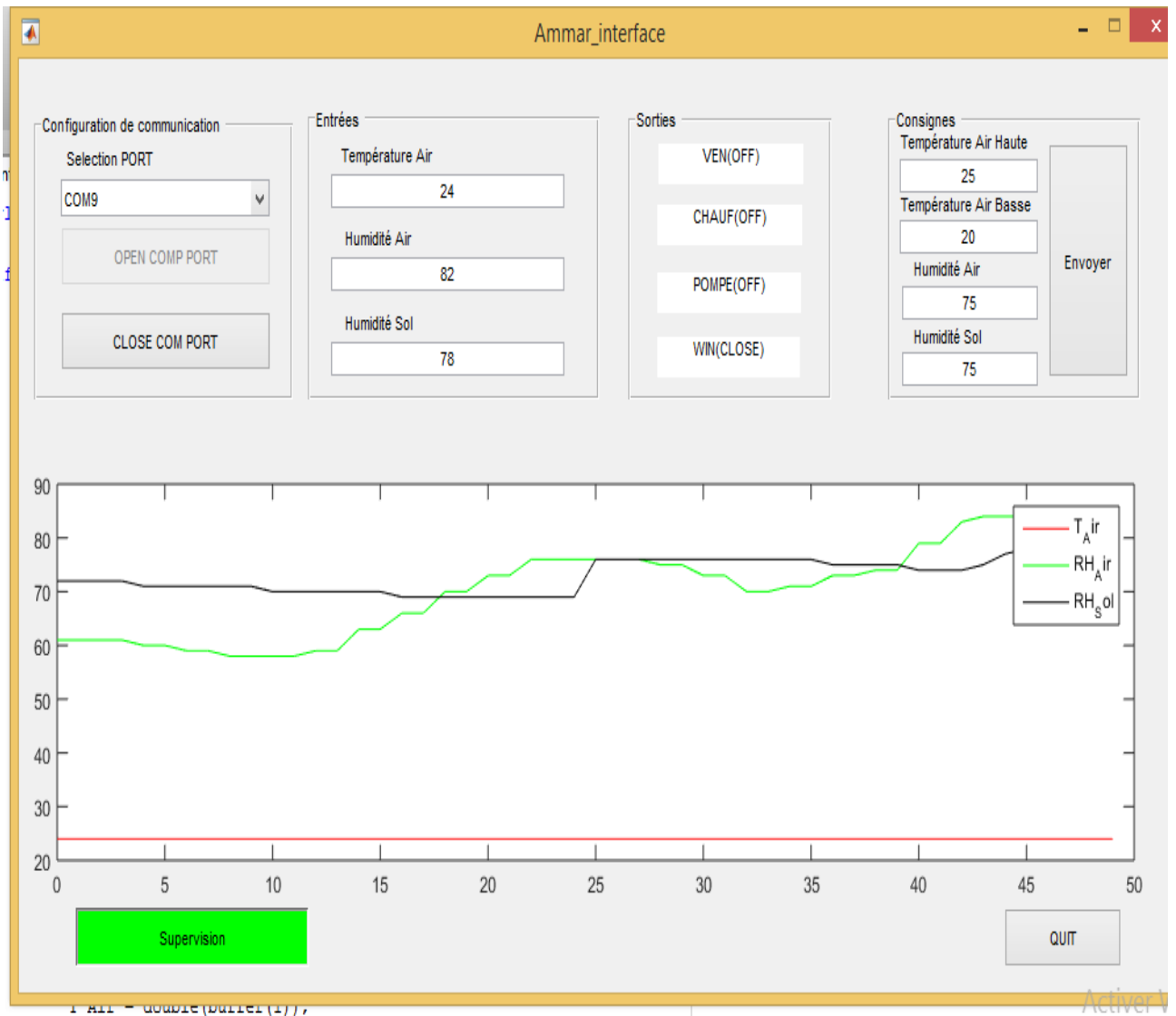


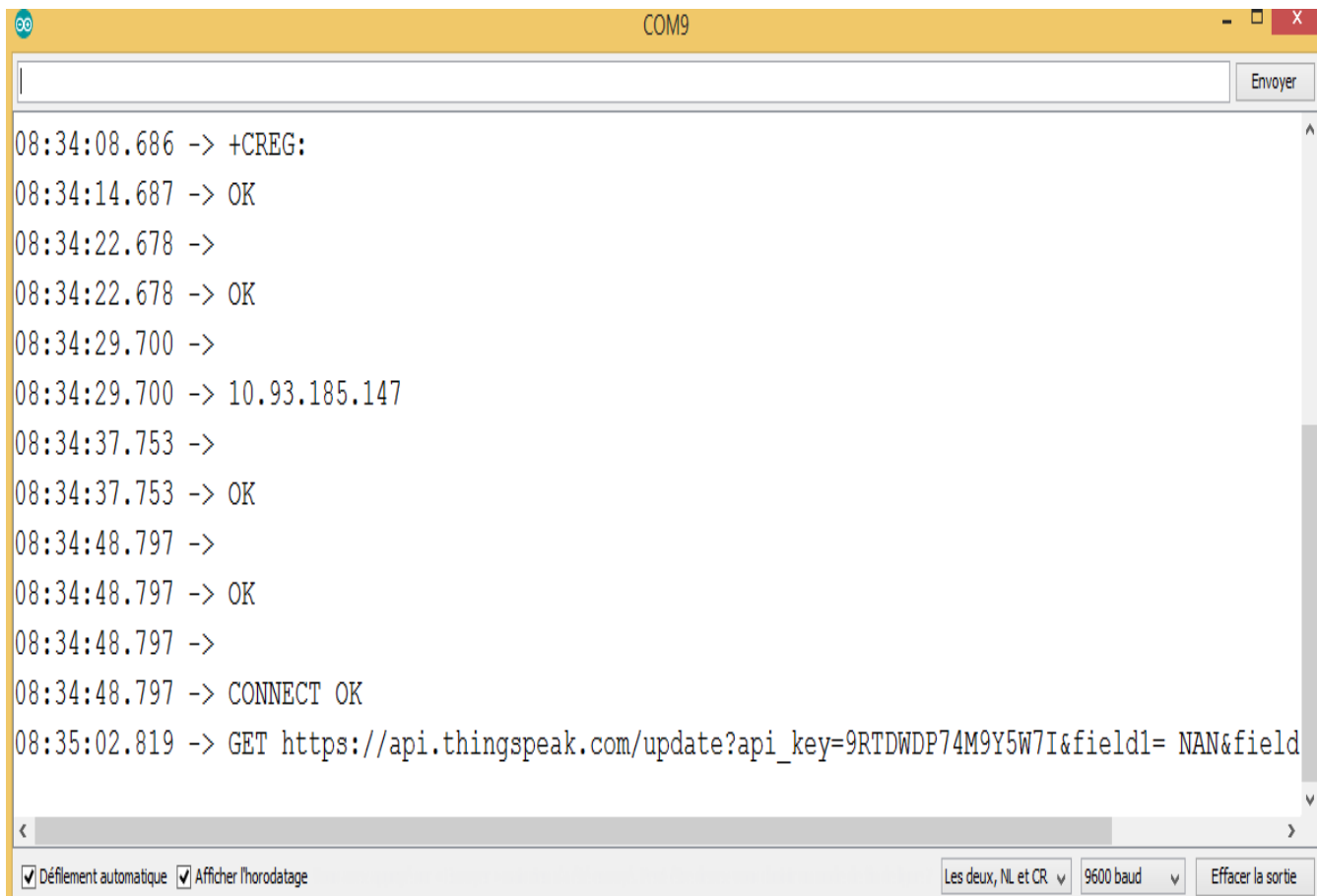
Figure III.10 : Interface avec régulation.

### III.4. Partie à distance :

#### III.4.1. Communication avec site :

Dans cette partie en utilise la communication à distance, on va utiliser une carte GSM pour transmettre les données (Température, humidité)

-Après la téléversment on vérifie le moniteur série.



**Figure III.11 :** Moniteur série.

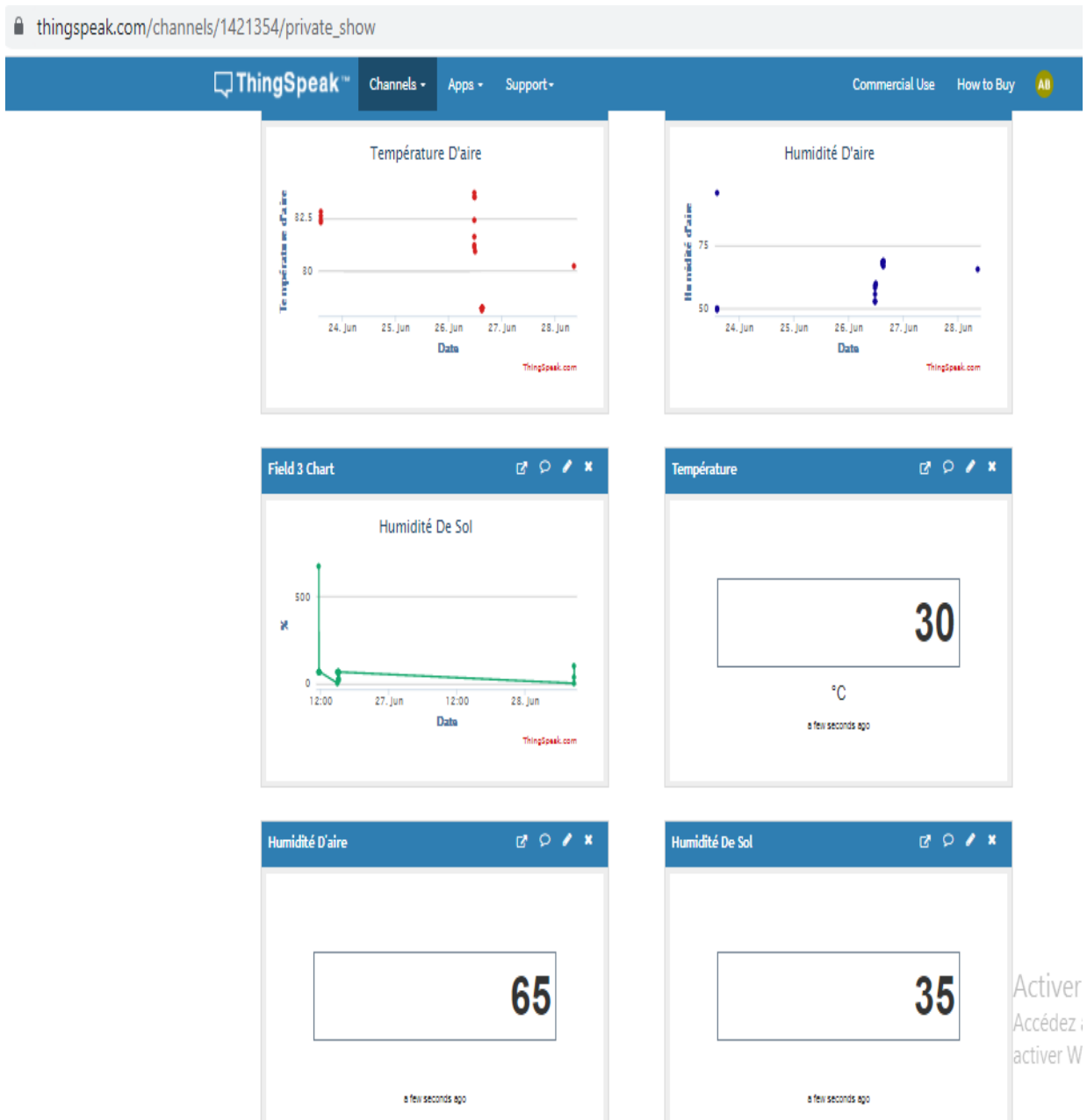


Figure III.12 : Site web lorsqu'on lance.

Après certain temps on voit la température et l'humidité stabilisé dans ces conditions initiales.

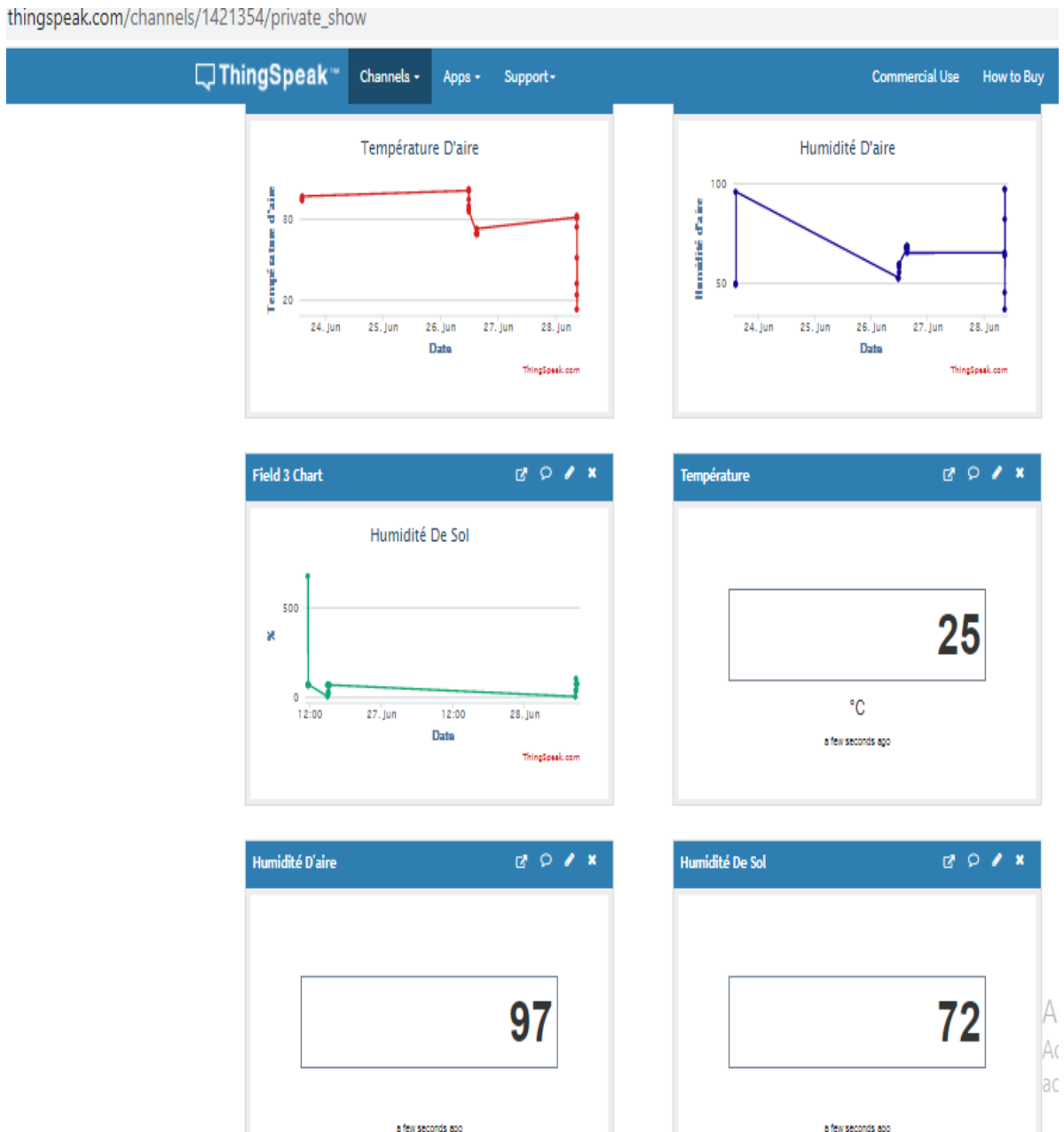


Figure III.13 : Site web avec régulation.

### III.4.2. Contrôle et commande de la serre :

Dans cette partie, nous avons communiqué avec la serre à travers des messages :

1. Quand nous envoyons un message contenant 'DHT', GSM elle nous répond par un message indiquant l'état actuel de la température et de l'humidité.



**Figure III.14 :** Récupération des données.

```

23:52:16.837 ->
23:52:16.837 -> OK
23:52:26.355 ->
23:52:26.355 -> +CMT: "+21367535 ", "Amar Mob 1", "21/07/03,23:52:20+04"
23:52:26.448 -> DHT
23:52:26.448 -> found DHT
23:52:26.448 -> T_air = 25.80°C      DHT22 RH_air = 82.30 %
23:52:26.495 -> RH_sol = 67 %
23:52:26.542 -> Sending Data...
23:52:28.183 -> Data Sent.
23:52:29.167 ->
23:52:29.167 -> OK
23:52:29.167 ->
23:52:29.167 -> >
23:52:58.188 ->
23:52:58.188 ->
23:52:58.228 -> OK

```

Serial Monitor Settings:  Défilement automatique  Afficher l'horodatage | Les deux, NL et CR | 9600 baud | Effacer la sortie

```

RH_SOL = 100 - RH_SOL;
Serial.print("T_air = "); Serial.print(T_aire); Serial.print("°C      DHT22 RH_air = "); Serial.print(RH_aire); Ser

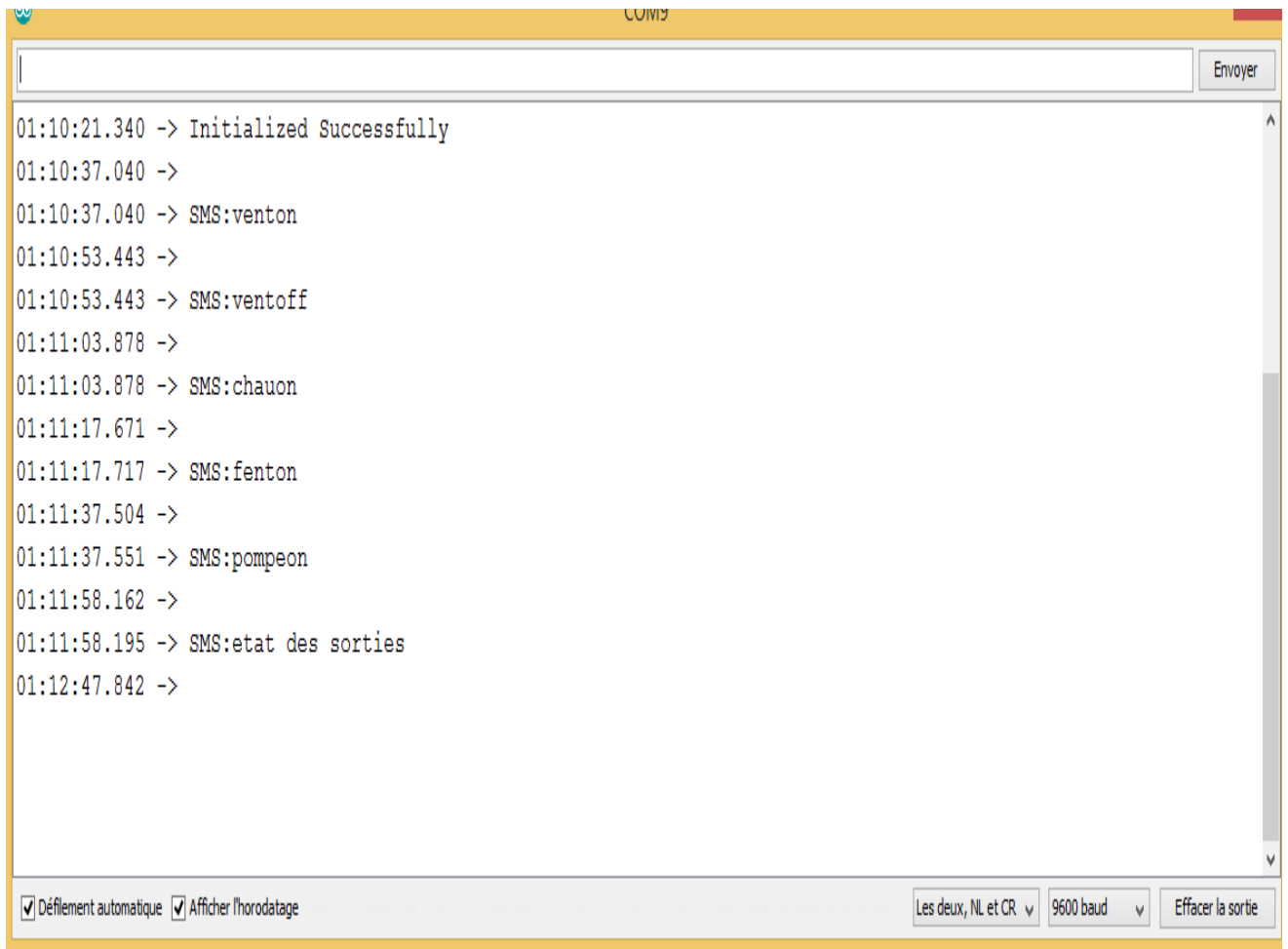
```

**Figure III.15 :** Moniteur série.

2. La commande des actionneurs avec des messages ON et OFF, et connaître l'état des actionneurs.



Figure III.16 : Contrôlé des actionneurs.



```
if (GSM_Msg=="fentoff"){pompe=0; sendSMS(GSM_Nr,"Ok fenêtre est Off");}
```

**Figure III.17 :** Moniteur série.



3. En peut modifier les conditions initiales par des messages

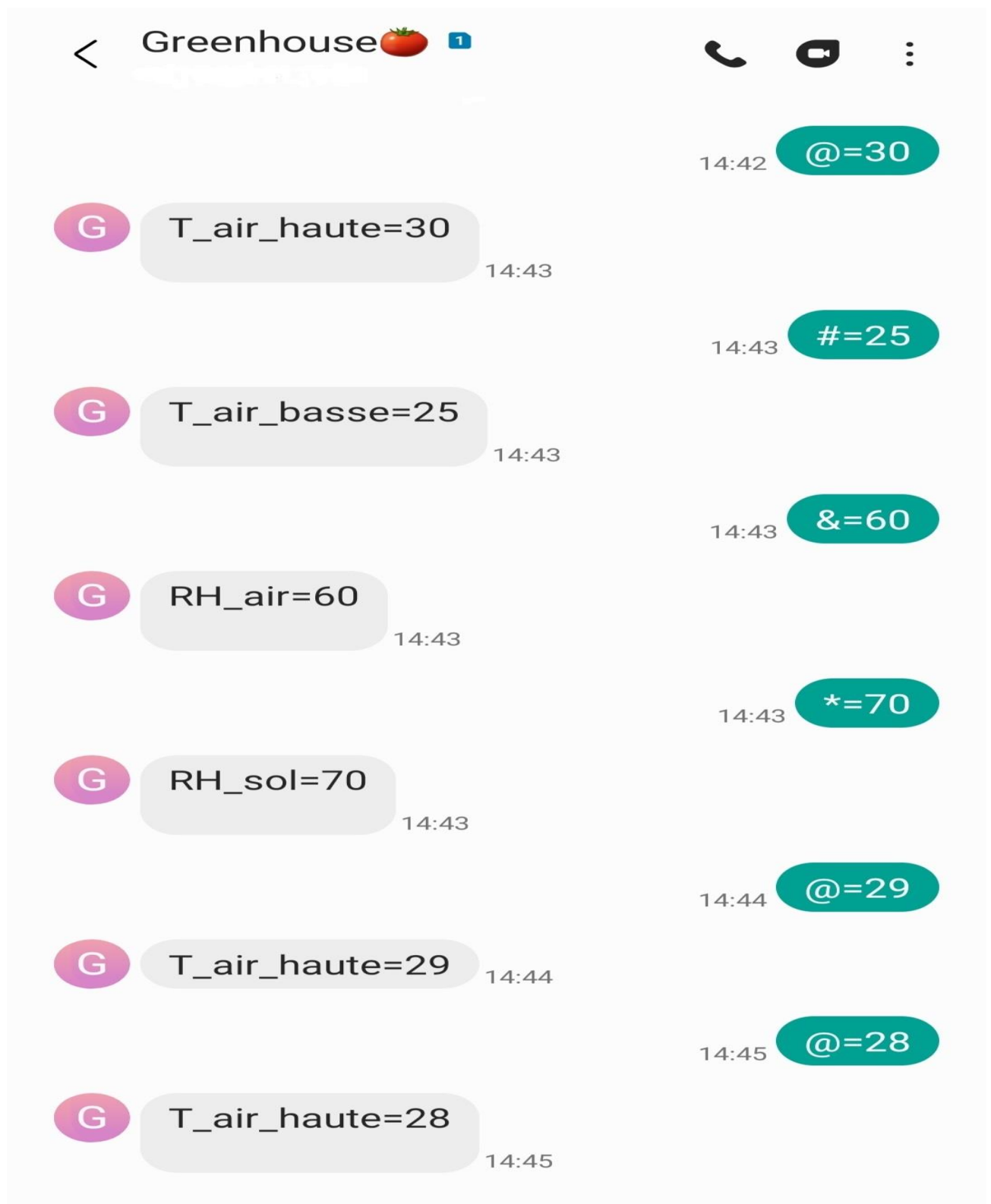
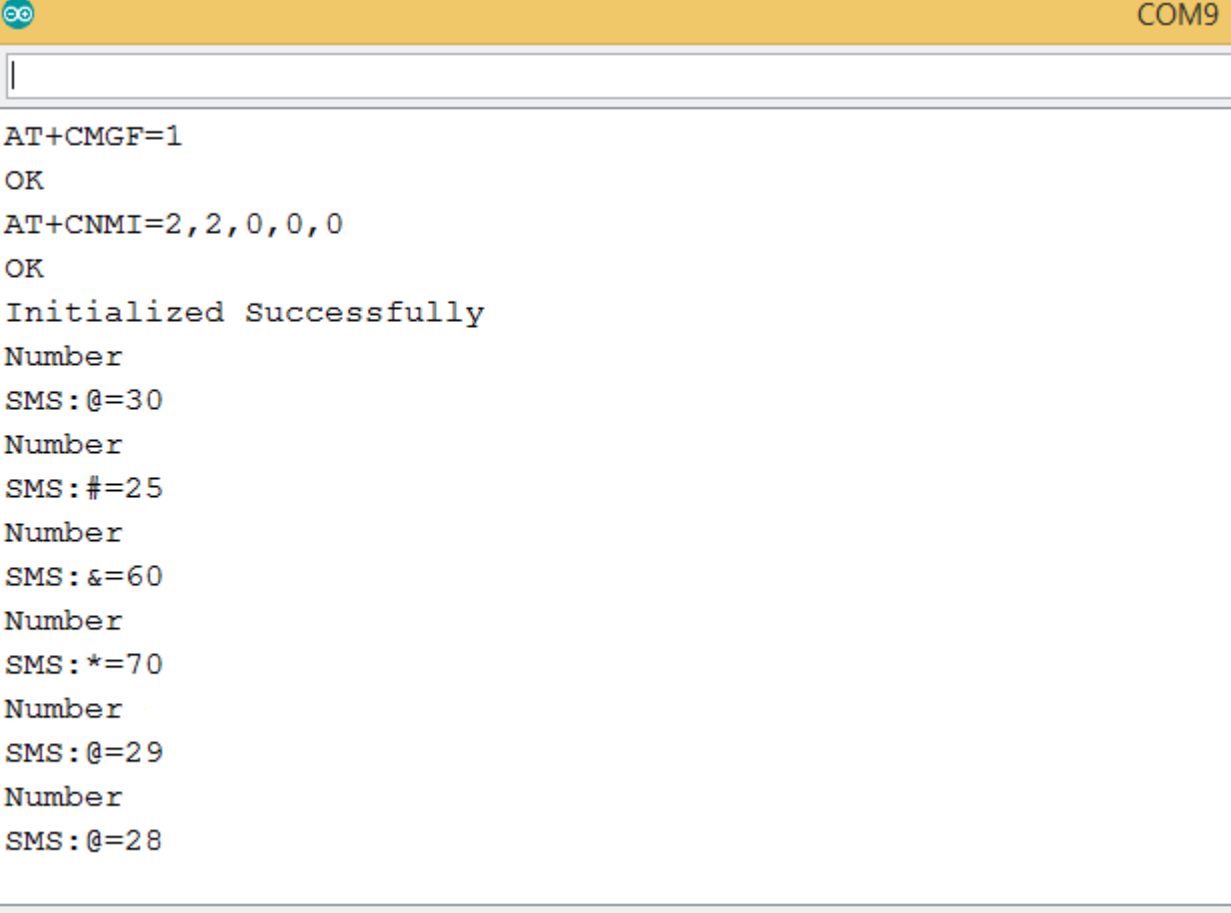


Figure III.18 : Modifie les données avec des messages.

Envoi voir moniteur série



The screenshot shows a serial monitor window with a yellow header bar containing a refresh icon on the left and the text 'COM9' on the right. Below the header is a text area displaying the following output:

```
AT+CMGF=1
OK
AT+CNMI=2,2,0,0,0
OK
Initialized Successfully
Number
SMS:@=30
Number
SMS:#=25
Number
SMS:&=60
Number
SMS:*=70
Number
SMS:@=29
Number
SMS:@=28
```

Figure III.19 : Moniteur série.

#### III.4. Conclusion :

Après la phase de montage de la structure et l'installation des équipements à l'intérieur et à l'extérieur de la serre, nous avons pu constituer une base de données climatique. A partir des résultats obtenus des tests sur les conditions réelles, nous avons constaté que la commande appliquée sur le système donne des résultats satisfaisants.

## **Conclusion générale :**

Afin d'augmenter la qualité et la quantité des produits agricoles issue de la culture sous serres et de faire face à un marché très concurrentiel, il est nécessaire d'avoir un contrôle climatique optimal.

Pour cela, nous avons la conception et la réalisation d'une serre agricole autonome contrôlé en local et à distance. Elle est composée d'une partie électronique constituée d'une carte de commande Arduino et une carte de réseau GSM, et des capteurs et des actionnaires. Et d'une partie logicielles sous forme d'une interface en locale et d'une application Web pour le contrôle du système à travers des messages.

Nous avons organisé ce mémoire en trois chapitres :

Dans le premier chapitre, nous avons déterminé les exigences dont les tomates ont besoin pour pousser dans un environnement à température et humidité appropriées.

Dans le deuxième chapitre, nous avons présenté les matériels et les logiciels en a besoin.

Dans le troisième chapitre, nous avons présenté notre serre de tomate et comment on peut la contrôler en locale et à distance.

Ce travail nous permet de mettre à l'épreuve d'une part et d'améliorer d'autre part nos connaissances théoriques et pratique en électronique et en automatique, et construire une serre sur les normes et produire des tomates qui poussent dans un environnement approprié pour elle.

**Références bibliographiques**

[1] **B. BOUCHIKHI, A. ED-DAHAK, A. LACHHAB, AND L. EZZINE, Les 26-27 Avril (2006)**, Automatic control of the climate and the drip fertirrigation under greenhouse, 1ere Salon International de l'Agriculture au Maroc-Meknès.

[2] **J. M. Aaslyng, N. Ehler and L. Jakobsen, 2015**, Climate control software integration with greenhouse, Environmental Control computer environmental Modelling & Software 521-527.

[3] **R. Salazar, A. Rojano, I. Lopez, novembre 2010**, "A Model for the Combine Description of the Temperature and Relative Humidity Regime in the Greenhouse," Ninth Mexican International Conference on Artificial Intelligence.

[4] **Catalogue de la Société ULMA-Agricola, 2008**, "Serres-multi chapelle".

[5] **Gent. M, 1988**, Effect of diurnal temperature variation on early yield and fruit size of greenhouse tomato. Applied agronomical Research, 3,257-263.

[6] **Madrep\_Derd, 1999**, transfert de technologie en agriculture, fiche technique, tomate sous serre N°57, 1-4pp.

[7] **Erik BARTMANN, 2015**, « Le grand livre d'Arduino».

[8] **Hocine TAKHI, 2014**, «Conception et réalisation d'un robot mobile à base d'arduino».

[09] **Jean- Noël,2006** , « livret Arduino en français », centre de ressources art sensitif.