

الجمهورية الجزائرية الديمقراطية الشعبية
وزارة التعليم العالي والبحث العلمي

UNIVERSITE BADJI MOKHTAR - ANNABA
BADJI MOKHTAR – ANNABA UNIVERSITY



جامعة باجي مختار – عنابة

Faculté : TECHNOLOGIE

Département : ELECTRONIQUE

Domaine : SCIENCES ET TECHNIQUES

Filière : ELECTRONIQUE

Spécialité : INSTRUMENTATION

Mémoire

Présenté en vue de l'obtention du Diplôme de Master

Thème:

Smart Green House

Présenté par : *Talai Haithem*

Hemaissia Faiçal

Encadrant : *Zadam Mohammed*

MCB

UBMA

Jury de Soutenance :

Brik Fatima	MCA	UBMA	Président
Zadam Mohammed	MCB	UBMA	Encadrant
Hafs Toufik	MCA	UBMA	Examineur

Année Universitaire : 2021/2022

Remerciements

Nous tenons à remercier, dans un premier temps, ALLAH, qui nous aide et nous donne la patience et le courage durant ces longues années d'étude.

Nous voulons remercier notre dirigeant, Dr Zadam M. pour son aide et ses conseils tout au long de l'achèvement du projet.

Nous tenons aussi à remercier Dr Talai Z. enseignant au département d'informatique pour son aide et son assistance durant la rédaction de notre mémoire

Nous remercions nos parents et toute notre famille, de nous avoir soutenu et encouragé pendant toutes nos études.

Enfin, nous exprimons nos remerciements les plus sincères à tous nos enseignants du département d'électronique de l'université Badji Mokhtar Annaba et plus particulièrement notre jury de soutenance.

أوصى أبي درداء رضي الله عنه فقال :

{ اطلبوا العلم فإن عجزتم فأحبوا أهله ،

فإن لم تحبهم ،

فلا تبغضوهم }

ملخص

الذي سنقوم بعرضه هو بيت الأخضر يعمل على توفير البيئة المناسبة التي تلائم النبات الذي نريد زراعته. نعلم ان مجال الزراعة من المجالات المساهمة في اقتصاد وعلى هذا الأساس قمنا بعرض هذا المشروع الذي يعمل على قياس درجة حرارة ورطوبة البيئة وكمية الضوء .
انشاء آلية تعمل على تنشيط واغلاق المساهمين (مروحية كمية)
(كمية)
الهدف من هذه التقنيات هي الحد والزيادة في ونوعية وكمية المنتج قصير، وهي مجدية وميسورة التكلفة ويستخدم بيت الأخضر الذكي بطاقة الاردوينو " مبرمج لجميع إشارات ا يعطي نتا جه الى جهاز المرحل الذي يقوم بتشغيل المساهمين . قمنا ايضا باستخدام بطاقة درع الانترنت لكي يتم متابعة هذا النظام بواسطة موقع على الواب و الهاتف الذكي المحمول باستخدام تطبيق يعرض جميع البيانات على .
الكلمات الرئيسية: بيت الأخضر الذكي، انترنت الاشياء، مستشعر رطوبة التربة، مستشعر كمية الضوء، dht 11, Arduino méga , ethernet Shield

Résumé

Le projet que nous allons présenter, est la maison verte intelligente, son rôle sert à fournir le meilleur environnement suivant la plante que nous voulons cultiver. Comme nous le savons, l'agriculture est l'un des domaines qui contribuent à l'économie du pays. Sur cette base, nous avons présenté ce projet, qui vise à mesurer la température et l'humidité de l'environnement, l'humidité du sol et la quantité de lumière à l'aide de capteurs, et de créer un mécanisme automatisé qui fait fonctionner les actionnaires. (Pompe, ventilateur, servomoteur, lampe) lors de la détection de tout phénomène physique (humidité, température, humidité du sol, quantité de lumière) à l'aide des capteurs, sans interférence humaine. Le but de ces technologies est de réduire l'interférence humaine, de permettre l'obtention en un temps réduit un produit en plus grande quantité et de meilleure qualité. Ce projet « smart Green House » utilise une carte Arduino méga programmée pour tous les signaux d'entrée basés sur des capteurs et la sortie est liée à un relais qui active les actionnaires. Nous avons également utilisé le Shield Ethernet pour nous permettre de suivre l'évolution environnementale de notre green house via un site web Thinkspk et un smartphone en utilisant l'application virtuino qui nous affiche toutes les données.

Mots-clés : Smart Green House, internet des objets, Capteur d'humidité du sol, Capteur de quantité de lumière, dht 11, Arduino méga, Shield Ethernet.

Abstract

The project we will show is the smart green house, its role serves to provide the right environment for the plant we want to grow, As we know, agriculture is one of the areas that contribute to the country's economy. On this basis, we presented this project, which aims to measure the temperature and humidity of the environment, soil moisture and the amount of light using sensors, and create an automated mechanism that activates and closes shareholders. (Pump, fan, servo motor, lamp) when detecting any content (humidity, temperature, soil humidity, amount of light) using the sensors, without human interference.

The goal of these technologies is to reduce human interference and increase the quality and quantity of the product and in a short time, which is feasible and affordable, and this smart Green House project uses a huge Arduino mega board programmed for all input signals based on sensors and gives its output to the relay that activates the shareholders.

We also used the Ethernet Shield to track this access via a website and mobile smartphone using an app that displays all the data on the system.

Keywords: Smart Green House, Internet of Things, Soil moisture sensor, Light quantity sensor, dht 11, Arduino mega, Shield Ethernet.

Table des matières

Introduction générale.....	1
I. Chapitre : Les piliers d'un Smart Green House.....	2
I.1 Introduction	3
I.2 Les capteurs.....	3
I.2.1 Le capteur de température et d'humidité (DHT11)	3
I.2.2 Le capteur d'humidité de sol	4
I.2.3 Le capteur de lumière (LDR)	6
I.3 Les actionnaires.....	7
I.3.1 La pompe à eau (Micro Submersible Water Pump DC 3V-5V).....	7
I.3.2 Servomoteur DF9GMS.....	8
I.3.3 Adda Ventilateur 12V.....	9
I.3.4 Module Relais 4 canaux 12V DC - 250V AC	10
I.4 Conclusion.....	10
II. Chapitre : Le cerveau d'un Smart Green House.....	11
II.1 Introduction	12
II.2 Carte de traitement des données	12
II.2.1 Carte Arduino Méga 2560.....	12
II.3 Carte de transmission des données	13
II.3.1 La connexion de l'Ethernet Shield avec le router.....	15
II.4 Logiciel utilisé Arduino IDE (programme).....	17
II.5 Conclusion.....	17
III. Chapitre : La conception et la réalisation d'un Smart Green House	18
III.1 Introduction	19
III.2 Schéma en bloc du système à réaliser	19
III.3 L'organigramme	20
III.4 Schéma électrique du montage sur Fritzing	23
III.5 Affichage Arduino-PC par câble USB et écran LCD.....	24
III.6 Choix d'affichage.....	25
III.6.1 Avantages de l'Internet des objets	26
III.6.2 Choix du site d'affichage	26
III.7 Suivez la Maison Verte à distance.....	27
III.8 Utilisation de l'application Virtuino 6.....	28
III.9 La construction de la maison verte intelligente	31
III.10 Conclusion.....	31
Conclusion générale	32

Liste des figures

Figure 1 : L'image d'un capteur DHT11

Figure 2 : câblage du capteur DHT11 avec l'Arduino Méga

Figure 3 : L'image d'un capteur FC-28

Figure 4 : câblage du capteur FC-28 avec l'Arduino Méga

Figure 5 : L'image d'un capteur de lumière (LDR)

Figure 6 : câblage du capteur de lumière (LDR)

Figure 7 : l'image d'une pompe à eau DC 3-5V

Figure 8 : l'image d'une Servomoteur DF9GMS

Figure 9 : L'image d'un ventilateur

Figure 10 : L'image d'un module relais .

Figure 11 : l'image de Méga Arduino 2560

Figure 12 : l'image d'Ethernet shield

Figure 13 : l'image d'Ethernet shield enchainé avec l'Arduino Méga

Figure 14 : les paramètres de router

Figure 15 : Interface IDE Arduino

Figure 16 : Schéma synoptique de système à réaliser.

Figure 17 : L'interfaçage de logiciel de simulation Fritzing .

Figure 18 : L'organigramme 1

Figure 19 : L'organigramme 2

Figure 20 : L'organigramme 3

Figure 21 : Montage en plaque d'essai

Figure 22 : Image du montage réalisé connecté par USB.

Figure 23: Fenêtre d'affichage des résultats sur le moniteur sériel.

Figure 24 : l'utilisation de l'internet des objets

Figure 25 : l'interface de site web (ThingSpeak)

Figure 26 : Courbe graphique exprime le changement de la Température.

Figure 27 : Courbe graphique exprime le changement de l'Humidité

Figure 28 : Courbe graphique exprime le changement de l'Humidité de sol

Figure 29 : Courbe graphique exprime le changement de la quantité de lumière

Figure 30 : logo d'App Virtuino 6

Figure 31 : l'inscription à Virtuino 6

Figure 32 : crée l'affichage sur Virtuino 6

Figure 33 : L'affichage numérique et analogique

Figure 34 : image réel de maison intelligente

Introduction générale

L'évolution rapide de la Technologies d'Information et de L'internet des objets depuis les années 1990 touche le monde du travail de plein fouet. Nous analyserons les effets actuels et futurs de l'évolution de L'internet des objets dans domaine l'agriculture.

L'agriculture est un domaine très difficile qui exige des efforts matériels et financiers. C'est pourquoi les agriculteurs se heurtent à un certain nombre d'obstacles, y compris le manque de main-d'œuvre et les matériaux qui contribuent à la qualité et à la quantité du produit. De ce point de vue, nous avons développé ce projet afin d'aider et de progresser dans ce domaine.

le projet que nous allons vous présenter est la SMART GREEN HOUSE, structure qui mesure la température, l'humidité de l'environnement la quantité de lumière, qui s'adapte au type de plante que nous cultivons et que nous pouvons surveiller grâce a des données que nous envoyons à un site WEB ou une application, tout cela grâce à la carte Arduino méga et la carte Ethernet Shield, des capteurs et des actionnaires. L'accès aux données se fait via le Site web ThingSpeak avec l'application Vituino 6 où toutes les informations sont disponibles à tout moment.

Aujourd'hui, le nombre de travailleurs agricoles se diminue, car leur tâche demande un effort physique énorme. Pour résoudre ce problème, nous proposons une idée simple et efficace pour alléger le fardeau des travailleurs agricoles. Notre travail consiste à construire une structure qui fonctionne, automatiquement grâce à des capteurs de phénomène naturels: température, humidité, humidité de sol, quantité d lumière. Ces actions seront Contrôlées par la carte Anduino Mega

Cette structure SMART GREEN HOUSE reliée à la carte Ethernet Shield que nous Connecterons au site Web pour le transfert des données. Notre projet pilote est : créer une maison verte, Presque indépendante de l'homme grâce à la liaison a avec Anduino Mega, Internet (Site Web) et Virturmo 6 pour le traitement des données.

I. Chapitre : Les piliers d'un Smart Green House

I.1 Introduction

Dans cette partie, nous allons donner une brève description des différents composants (capteurs et actionneurs) qui vont nous permettre de contrôler les différents paramètres influents en agriculture et intervenir si cela est nécessaire en cas de panne d'un des éléments de notre green house.

I.2 Les capteurs

I.2.1 Le capteur de température et d'humidité (DHT11)

L'élément DHT11 fonctionne avec un capteur de température et d'humidité complexe consistant en une sortie de signe virtuel calibré.

L'utilisation d'une sorte Virtual-sign-acquisition, technologie de détection de température et d'humidité, il garantit une haute fiabilité et une excellente stabilité à long terme.

Ce capteur se compose d'une dimension d'humidité de type résistif aspect et une dimension de la température NTC connecté à un Microcontrôleur 8 bits de haute performance, offrant réponse rapide d'excellente qualité sans interférence d'une grande capacité et rentabilité.

Tout détail DHT11 sont calibré avec précision quant à l'étalonnage de l'humidité.

Les coefficients d'étalonnage sont enregistrés en tant que programmes dans la mémoire OTP, qui pourrait être utilisé par le processus de détection du signal interne du capteur. L'interface série du cordon non marié rend l'intégration de la machine courte et facile. Sa petite taille, faible consommation d'énergie et jusqu'à 20 transmission de signalisation de compteur. [1]

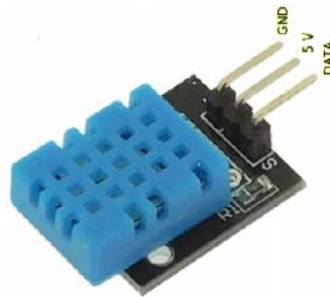


Figure 1 : L'image d'un capteur DHT11

Il suffit de connecter la première broche à gauche (Vcc) à l'alimentation de 3 à 5 V, la deuxième broche doit brancher à la carte Arduino, au pin A9 numérique La troisième broche est montée à la masse GND.

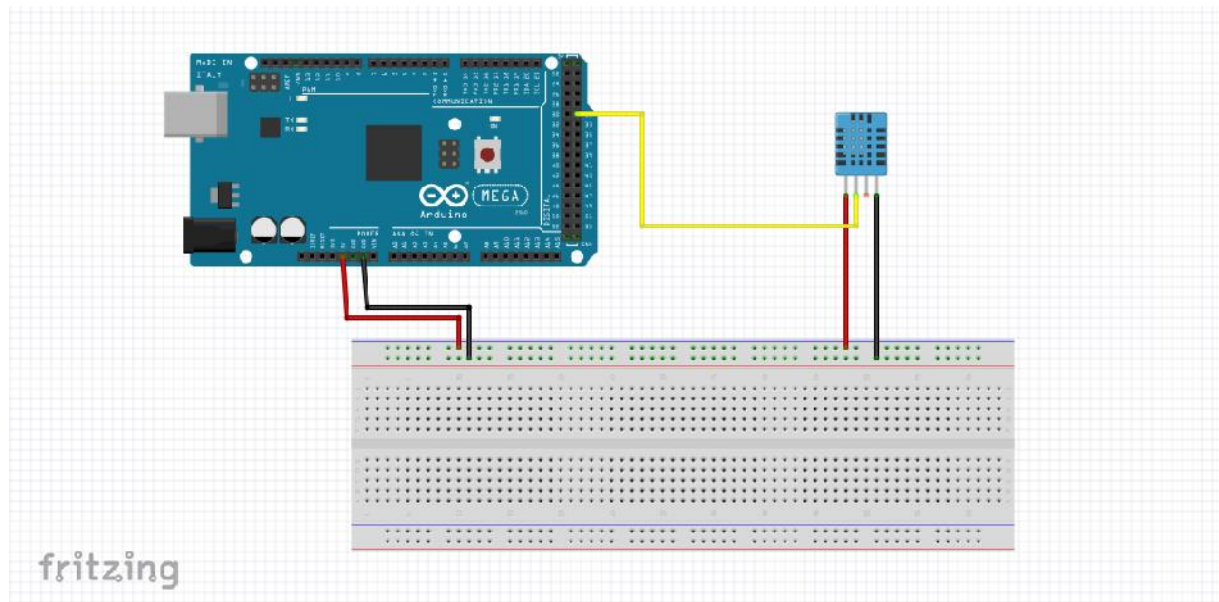


Figure 2 : câblage du capteur DHT11 avec l'Arduino Méga

Les caractéristiques de DHT11

- Puissance de E / S est 3.5 à 5.5 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 12 mm × 15.5mm × 5.5 mm .
- Poids : 2,4 g. [1]

I.2.2 Le capteur d'humidité de sol

FC-28 est un capteur d'humidité du sol ou un module hygrométrique. Il se compose de deux sondes qui sont utilisées pour mesurer la teneur volumétrique de l'eau. Il fonctionne en mesurant la résistance entre deux sondes métalliques qui sont insérées dans le sol à surveiller. Le paquet se compose d'un potentiomètre, un comparateur LM393 & sondes de test. Le capteur peut être utilisé dans deux modes de fonctionnement distincts analogique et numérique. Pour le mode analogique, nous devons utiliser la sortie analogique du capteur. Ici le capteur nous donne une valeur de 0 à 1023. En mode numérique, la sortie numérique du capteur est connectée à la broche numérique de l'Arduino. Le module de capteur contient un potentiomètre, qui est utilisé pour définir la valeur de seuil.[2]

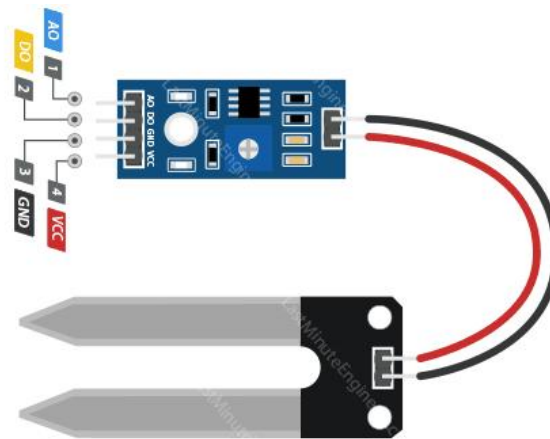


Figure 3 : L'image d'un capteur FC-28

Connectez les fils de détection du FC-28 avec les sondes Après cela, connectez la broche Vcc du capteur FC-28 avec la broche 5V de l'Arduino.
joindre la broche de terre du capteur FC-28 avec la broche de terre de l'Arduino.
Aussi, connecter la broche analogique du capteur avec la broche A1 de l'Arduino.

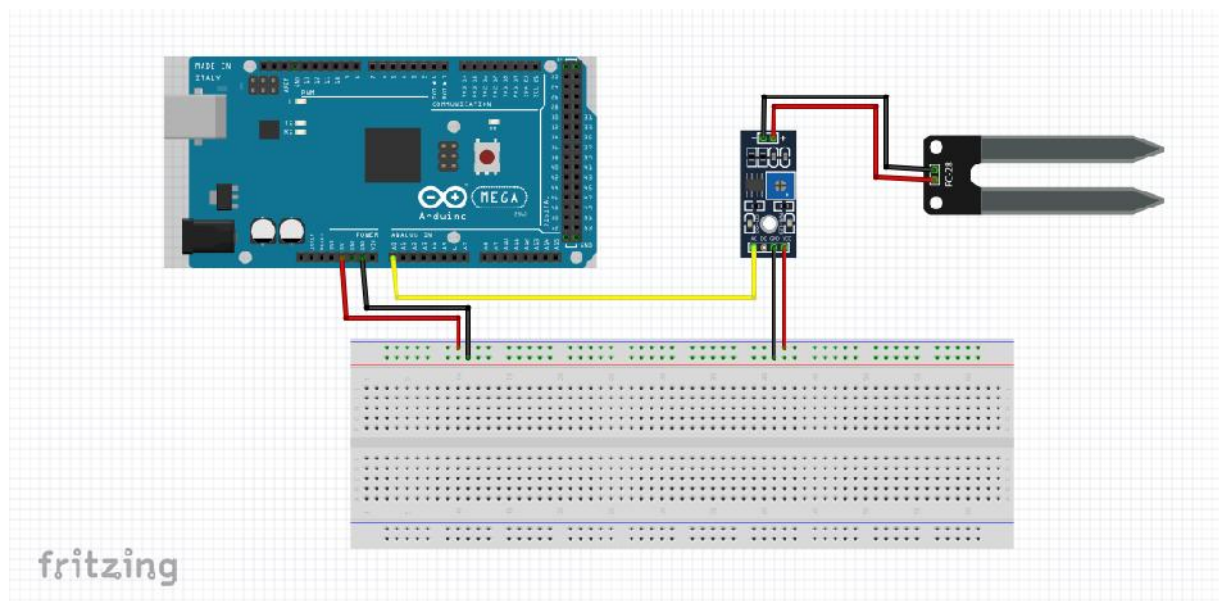


Figure 4 : câblage du capteur FC-28 avec l'Arduino Méga

Les caractéristiques de FC-28

- Tension de fonctionnement : 3,3 V à 5 Vcc.
- Courant de fonctionnement : 15 mA
- Sortie numérique – 0V à 5V, niveau de déclenchement réglable à partir du pré-réglage
- Sortie analogique – 0V à 5V en fonction du rayonnement infrarouge provenant de la flamme qui tombe sur le capteur
- LED indiquant la sortie et la puissance

- Sortie analogique de la teneur en humidité
- Sortie numérique de la teneur en humidité avec point de consigne réglable
- Taille des BPC : 3,2 cm x 1,4 cm
- Puce de comparaison : LM393
- Sonde de sol en forme de fourche
- Câble d'interface F/F 5 fils, 8 longs
- Facile à utiliser avec les microcontrôleurs ou même avec un circuit intégré numérique/analogique normal [2]

I.2.3 Le capteur de lumière (LDR)

La photorésistance est un composant dont la résistance spécifique dépend de la luminosité ambiante. En termes simples, il s'agit d'une résistance dont la valeur change avec la lumière. Il existe différents types de photorésistances, chacune avec une résistance différente en fonction de la lumière ambiante. Le type de photorésistance le plus courant est 1M ohm (sombre) / 12K ohm (pleine lumière). Ce type de photorésistance sera utilisé plus loin dans ce tutoriel. Indépendamment du diamètre, de la teinte ou de la valeur lumineuse totale de la photorésistance, sa résistance diminue lorsque la photorésistance est éclairée. Par conséquent, vous pouvez utiliser la photorésistance pour mesurer la luminosité ambiante.

Sans faire une liste exhaustive, voici quelques exemples d'utilisations très classiques pour une photorésistance :

- Détection jour / nuit.
- Mesure de luminosité ambiante (pour ajuster un éclairage par exemple).
- Suiveur de lumière (pour panneaux solaires, robots) pour nous Smart Green House. [3]

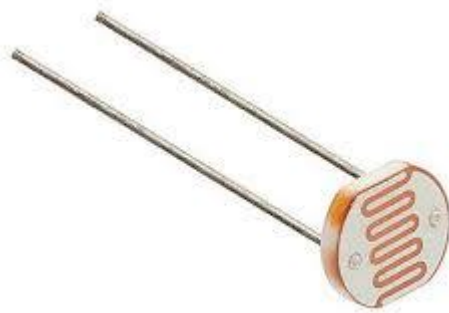


Figure 5 : L'image d'un capteur de lumière (LDR)

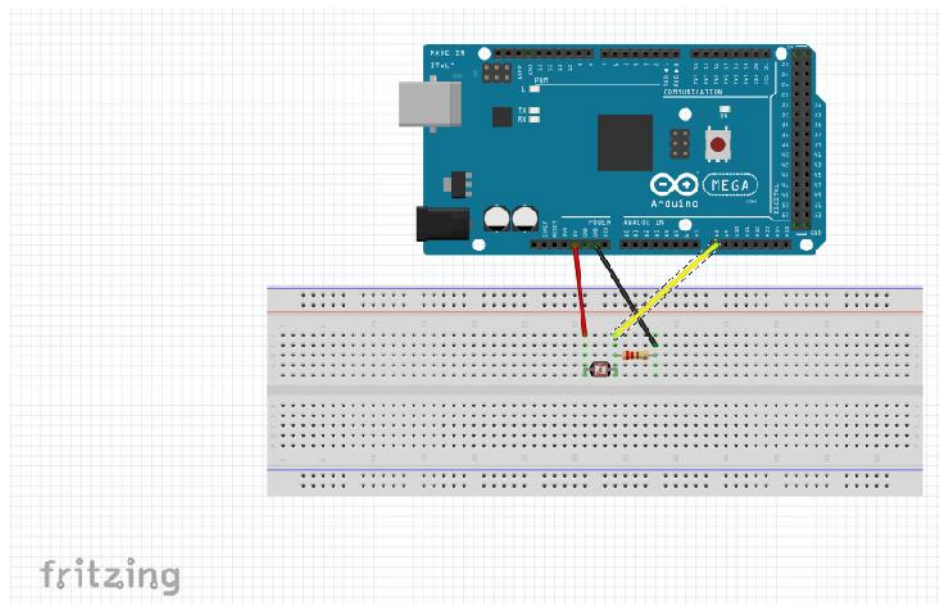


Figure 6 : câblage du capteur de lumière (LDR)

I.3 Les actionnaires

I.3.1 La pompe à eau (Micro Submersible Water Pump DC 3V-5V)

La Micro Submersible Water Pump DC 3V-5V, peut être facilement intégrée à notre projet de système d'eau. Elle fonctionne en méthode d'aspiration par l'entrée et jet par la sortie. Nous pouvons utiliser la pompe à eau comme un système d'irrigation intelligent pour notre maison. [4]



Figure 7 : l'image d'une pompe à eau DC 3-5V

Il suffit de connecter le fil rouge (+) avec la branche NO et le fil noir (-) à GND. Pour utiliser la pompe il faut l'immerger dans l'eau et lorsque l'alimentation est activée, l'eau s'écoule dans l'entrée de la pompe et s'écoule par la sortie.

Les caractéristiques de La pompe à eau

- Tension d'entrée : CC 3V-5V
- Débit : 1,2 à 1,6 L/min
- Température de fonctionnement : 80 °C
- Courant de fonctionnement : 0,1-0,2 A
- Distance d'aspiration : 0,8 mètre (max.)
- Diamètre extérieur de la sortie d'eau : 7.5mm
- Diamètre intérieur de la sortie d'eau : 5,0 mm
- Diamètre de l'eau Entrée : 5,0 mm
- Longueur du fil : 200 mm
- Dimensions : 45 x 30 x 25 mm
- Poids : 30 g [4]

I.3.2 Servomoteur DF9GMS

Le DFRobot DF9GMS est un microservo à 360 degrés, léger avec engrenage en plastique compact. Peut être utilisé pour ouvrir les fenêtres automatiquement vous pouvez contrôler le sens de rotation et la vitesse.

Il n'y a pas d'arrêt matériel pour cela L'arbre peut tourner librement, comparé aux moteurs à courant continu ordinaires, ce n'est pas le cas des servomoteurs à 360 degrés. Nécessite un pilote de moteur supplémentaire, plug and Play, compact et pratique, contrôle Arduino La méthode est la même.

Si le servo est entraîné par une alimentation 4.8-6V, le couple peut atteindre 1.2-1.6kgxcm [5]



Figure 8 : l'image d'une Servomoteur DF9GMS

Le câblage est relativement facile, Un fil rouge du servomoteur à la broche 5V sur la carte Arduino, un fil noir à la broche GND et un fil blanc à la broche D9 sur la carte Arduino (ou jaune selon le fabricant).

Les caractéristiques d'une Servomoteur DF9GMS

- Tension de fonctionnement : 3,5 6,0 V
- Tension d'alimentation : 4,8 V ~ 6,0 V
- Largeur de la zone morte : 5usec
- Vitesse de travail : 0,12 s/60 (4,8 V sans charge)
- Couple de décrochage : 1,2 kg/cm (4,8 V), 1,6 kg/cm (6,0 V)
- Emplacement neutre : 1500us
- Description de l'interface :
- Brun : GND
- Rouge : CAV
- Orange : S
- Longueur du câble : 250 mm
- Température de fonctionnement : 30 °C à 60 °C
- Dimensions : 22,6 x 12,2 x 30 mm /0,89x0,48x1,18 po
- Poids : 9g [5]

I.3.3 Adda Ventilateur 12V

Nous utilisons ce ventilateur pour réduire la température dans notre maison. On utilise



juste 2 pin, à partir d'un relais nous pouvons le commander.

Figure 9 : L'image d'un ventilateur

- Marque Adda
- Type de connecteur 3 broches
- Ventilateur / Cooling Case Fan.
- Marque : ADDA / HP / Compaq
- Modèle : AD0912HS-A76GL
- Référence : 326704-001
- Caractéristiques : DC 12V / 0.25A
- Dimensions : 92 x 92 x 25 mm
- Connexion : 3-Pin
- Longueur de câble : 24 cm

I.3.4 Module Relais 4 canaux 12V DC - 250V AC

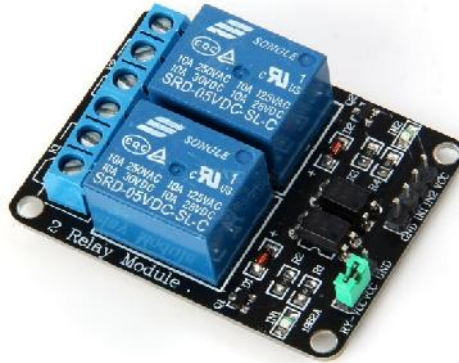


Figure 10 : L'image d'un module relais .

Module relais, Il peut être contrôlé directement avec un microcontrôleur (PIC, Arduino, Atmel, etc.). La diode protège le circuit des pointes de tension générées par la bobine. Comme la LED s'allume, il est facile de démarrer. Comprend des contacts normalement fermés et normalement ouverts. Chaque relais a une LED pour indiquer s'il est actif ou non. L'alimentation est de 12V DC. Les contacts supportent 250V avec un courant de 10A et sont équipés de bornes à vis. [6]

I.4 Conclusion

A la fin de cette partie, nous avons classé les éléments de base. Ces éléments permettent de prendre des informations associées à certains facteurs influents et d'autres qui aident à les changer (domaine agricole), surtout pour notre maison.

Nous les représentons la manière dont ils sont connectés au microcontrôleur. Dans le chapitre suivant Nous montrerons les deux composants responsables du traitement et la transmission.

II. Chapitre : Le cerveau d'un Smart Green House

II.1 Introduction

A travers de ce chapitre, nous allons expliquer le rôle d'Arduino, le responsable qui traite les informations des capteurs et de contrôlé des actionnaires, et le l'objectif d'Ethernet Shield le responsable de la liaison entre le microcontrôleur au site web (internet).Et nous allons donner un aperçu sur la manipulation (programmation) de ces deux.

II.2 Carte de traitement des données

Notre projet smart green house se deux partie :

La premier partie charge et traité la mesure des paramètres environnementaux premièrement et deuxièmement la réaction à partir de la mesure prenons, nous l'appellerons le smart green house qui contenant la carte Arduino et les différents objets.

Les principales de fonctionnalités devaient être :

- La mesure de l'humidité de l'air.
- La mesure de la température de l'air.
- La mesure de l'humidité de sol.
- La mesure de la lumière.
- La commande ventilation qui sert a réduit la température.
- La commande pompe pour l'irrigation.
- La commande fenêtre (Micro. Servo) qui sert a réduit l'humidité de l'air.
- La commande pour l'éclairage.

II.2.1 Carte Arduino Méga 2560

Arduino Méga 2560 est une carte microcontrôleur basée sur ATmega2560. 54 broches d'E / S numériques (dont 15 peuvent être utilisées comme sorties PWM), 16 entrées analogiques, 4 UART (ports série matériels), oscillateurs à cristal 16 MHz, ports USB, prises de courant, en-têtes ICSP et bouton de réinitialisation.

Il contient tout ce dont vous avez besoin pour prendre en charge un microcontrôleur. Commencez par vous connecter à votre ordinateur avec un câble USB ou allumez-le avec un adaptateur CACC ou une batterie.

La carte Méga 2560 est compatible avec la plupart des boucliers conçus pour Uno et les anciennes cartes Duemilanove ou Diecimila. [7]



Figure 11 : l'image de Méga Arduino 2560

Caractéristiques:

- Microcontrôleur atmega2560
- Tension de fonctionnement 5v
- Tension d'entrée (recommandée) 7-12v
- Tension d'entrée (limite) 6-20 v
- Broches d'e/s numériques 54 (dont 15 fournissent une sortie PWM)
- Broches d'entrée analogiques 16
- Courant cc par broche e/s 20 ma
- Courant cc pour broche 3,3 v 50 ma
- Flash memory 256 kb dont 8 kb utilisés par boot loader
- SRAM 8 kb
- EEPROM 4 kb
- Vitesse d'horloge 16 mhz
- led_builtin 13
- Longueur 101,52 mm
- Largeur 53,3 mm
- Poids 37 g [7]

Avantages et inconvénients :

- Nombreuses bibliothèques disponibles avec diverses fonctions implémentées.
- Logiciel et matériel open source et extensible.
- Nombreux conseils, tutoriaux et exemples en ligne (forums, site perso, etc.).
- Existence de « shield » (boucliers en français).
- Nombreuses entrées et sorties pour raccorder des capteurs ou des actionnaires
- Capacité de mémoire suffisante pour les gros projets.
- Plus de broches UART (4 ports de communication série). Plus de broches MLI (15 sorties numériques peuvent être utilisées comme MLI).
- Compatible avec la plupart des shields conçues pour Arduino
- Facteur de forme plus élevée que pour Arduino Uno
- Deux fois plus chère que l'Arduino Uno.
- Simplifie grandement les schémas électroniques.
- Diminué le coût de la réalisation.
- La charge de travail à la conception d'une carte électronique.
- Environnement de programmation clair et simple.
- Multiplateforme : tourne sous Windows, Macintosh et Linux.

II.3 Carte de transmission des données

La deuxième partie nomme base smart contenant l'Ethernet Shield avec l'unité de stockage et le router ceux-ci ont pour tâche de stocker les données et expédier vers un site web avec mise à jour régulière



Figure 12 : l'image d'Ethernet shield

Le Shield Ethernet Arduino permet à une carte Arduino de se connecter à Internet. Il est basé sur la puce Wiznet W5100 ethernet (fiche technique). Le Wiznet W5100 fournit une pile réseau (IP) capable de TCP et UDP. Il prend en charge jusqu'à quatre connexions simultanées. Nous pouvons utiliser la bibliothèque Ethernet pour écrire des croquis qui se connectent à Internet en utilisant le Shield. Le blindage Ethernet se connecte à la carte Arduino via un long en-tête de broche d'enroulement qui pénètre dans le blindage. Cela laisse la disposition des broches intacte et nous permettons d'empiler un autre bouclier sur le dessus. [8]

Caractéristiques:

- Σ Tension d'alimentation de 5V (fournit par la carte Arduino)
- Σ Contrôleur Ethernet : W5100
- Σ Vitesse de connexion : 10/100Mb
- Σ Connection avec l'Arduino : SPI
- Socle Ethernet RJ45
- Dimensions: 73 x 53 x 30 mm



Figure 13 : l'image d'Ethernet Shield enchainé avec l'Arduino Méga

Les principales fonctionnalités devaient être :

- Le stockage des données sur carte mémoire SD.
- L'envoi des données vers le site web.

II.3.1 La connexion de l'Ethernet Shield avec le router

On doit réserver une adresse IP statique pour l'Ethernet Shield, et activer DHCP du router pour ne pas avoir un conflit d'adresse IP.

300Mbps Wireless N ADSL2+ Modem Router

Interface
Quick Start
Interface Setup
Advanced Setup
Access Management
Maintenance
Status
Help

Internet
LAN
Wireless
6RD
Guest Network

Router Local IP

IP Address : 192.168.1.1
IP Subnet Mask : 255.255.255.0
Dynamic Route : RIP2-B Direction : Both
Multicast : IGMP v2
IGMP Snoop : ☐ Disabled ☒ Enabled
MldSnoop : ☒ Disabled ☐ Enabled

DHCP

DHCP Server

DHCP
☐ Disabled ☒ Enabled ☐ Relay

DHCP Server

Starting IP Address : 192.168.1.100 Current Pool Summary
IP Pool Count : 120
Lease Time : 120 seconds (0 sets to default value of 259200)
Physical Ports : ☒ 1 ☒ 2 ☒ 3 ☒ 4

DHCP Table

Hostname	IP Address	MAC Address	Status	Expire Time
	192.168.1.102	Manual Config	Static	
dhoppo0	192.168.1.100	D0:8F:4A:8C:E5:C3	Auto	0days, 0:1:15
Mypc	192.168.1.101	64:5A:04:69:1E:75	Auto	0days, 0:1:42
Noui	192.168.1.103	18:CF:5E:13:A1:17	Auto	0days, 0:1:10
RedmiNote8-RedmiNote	192.168.1.105	4C:63:71:8A:EA:A9	Auto	0days, 0:1:29
OPPO-A93	192.168.1.106	FA:DC:35:7B:09:82	Auto	0days, 0:1:36

DNS

DNS Relay : Use Auto Discovered DNS Server Only
Primary DNS Server : N/A
Secondary DNS Server : N/A

Radvd

Figure 14 : les paramètres de router

- L'adresse MAC de Ethernet Shield doit unique dans notre LAN.
- L'adresse IP de Ethernet Shield doit unique dans notre LAN.

II.4 Logiciel utilisé Arduino IDE (programme)

Après le câblage entre nos éléments notre réalisation ne peut pas fonctionner sans la partie software et l'utilisation de logiciel Arduino IDE nous permettons de manipuler les différents composants avec un programme écrit par un langage simple qui facilite le langage C++ Ce figure ci-dessus montre l'interface de logiciel :

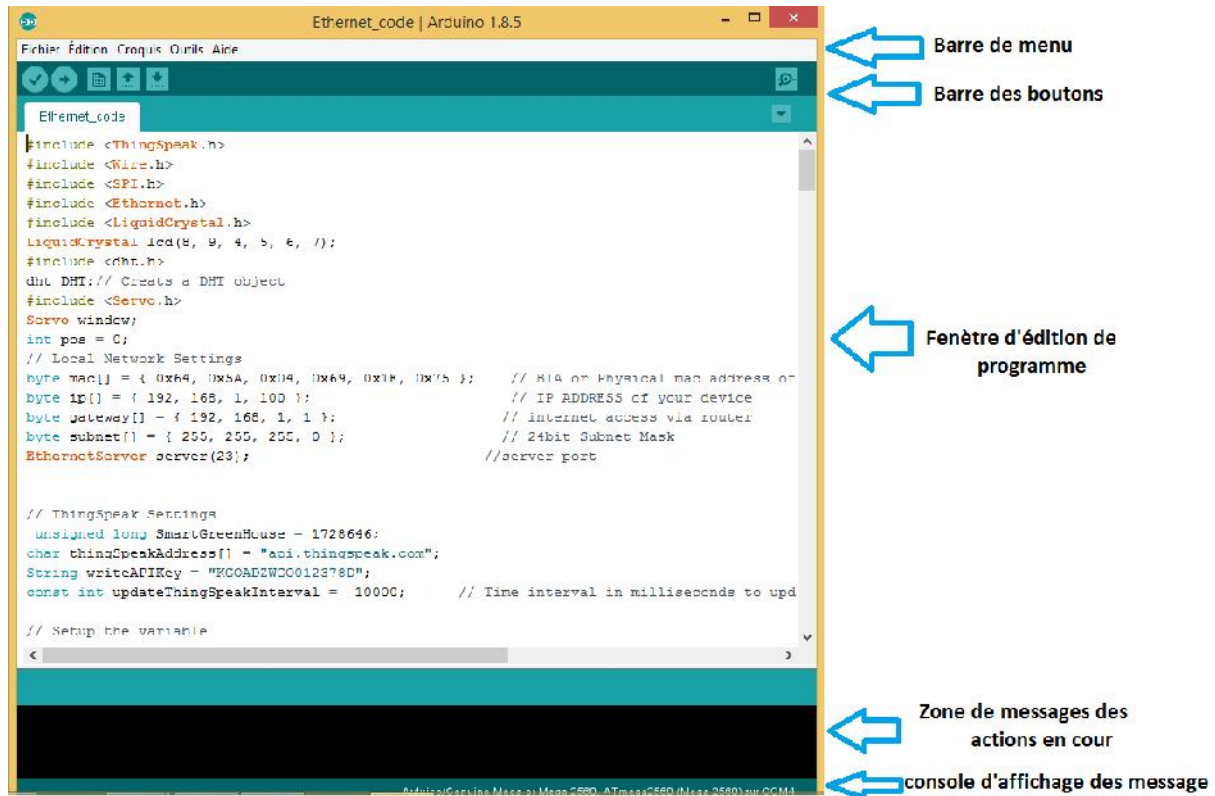


Figure 15 :Interface IDE Arduino

II.5 Conclusion

Dans cette partie, nous avons abordé sur les deux éléments importants de l'industrie d'une maison intelligente : L'Arduino méga Responsable de toutes les opérations, traitement de données et la contre agir de system. L'Ethernet shield le Responsable du tr de connexion à Internet et de transmission d'informations. Ces deux composant représentent le cerveau d'un smart green house.

III. Chapitre : La conception et la réalisation d'un Smart Green House

III.1 Introduction

Dans ce chapitre, conception et réalisation de systèmes intelligents pilotés par smartphones. Cette partie montre les étapes à suivre pour exécuter l'intégralité des parties logicielles et matérielles du système étudié. Ensuite, la présentation des résultats expérimentaux obtenus clôt cette phase.

Le fonctionnement de ce système consiste à contrôler et à modifier automatiquement l'environnement de la serre. Tout d'abord, il s'agit d'un système entièrement automatisé et ne nécessite pas de ressources humaines pour contrôler le système. Des blindages Ethernet sont utilisés pour contrôler l'ensemble du processus

Tente de fournir une petite application IoT pour utiliser le site ThingSpeak. Cela vous permet de connecter votre système via le blindage Ethernet. A la fin de ce chapitre, nous montrerons les résultats obtenus après la réalisation de cette maison.

III.2 Schéma en bloc du système à réaliser

Notre système est divisé en quatre parties, la première partie, des capteurs qui reçoivent des phénomènes physiques (température, humidité et l'humidité du sol) et la deuxième partie des dispositifs responsables de modifier ces phénomènes et la troisième partie est l'unité de traitement des données (Arduino Mēga) et la dernière partie est l'unité de transfert de données (Ethernet Shield)

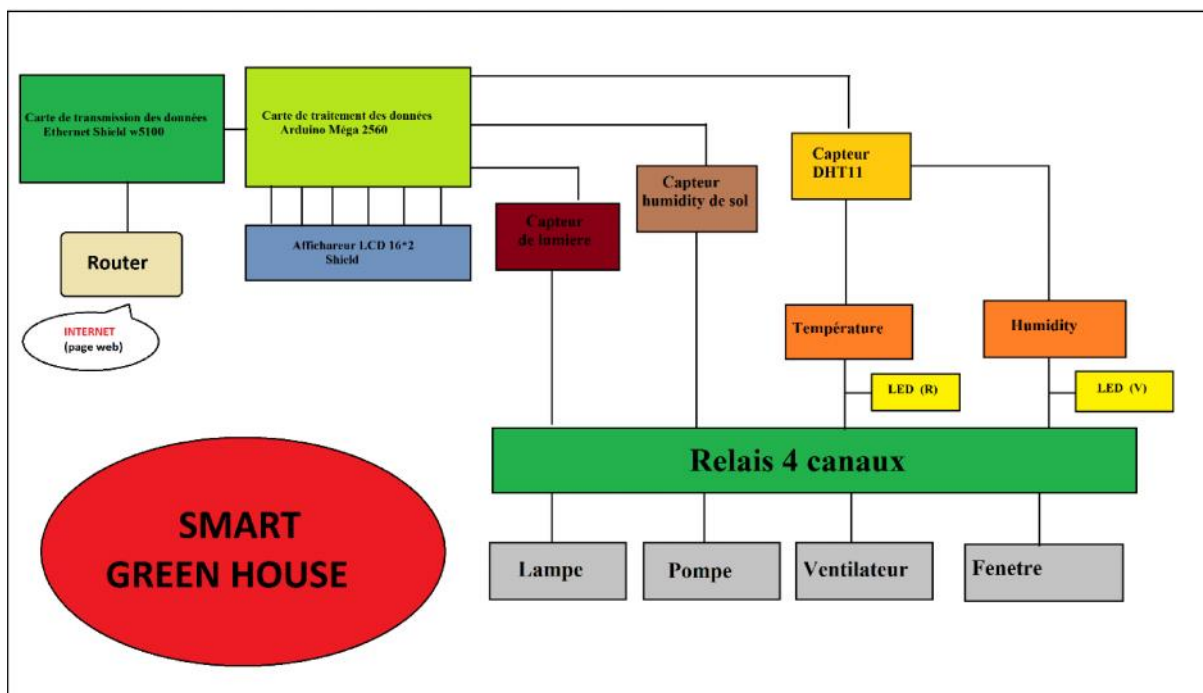


Figure 16 :Schéma synoptique de système à réaliser.

III.3 L'organigramme

Lorsque le system est alimenté, l'Arduino traite les différentes valeurs du capteur et affiche sur LCD chaque départ. Après 04 opérations seront faites en même temps pour expliquer ces 04 opérations on a divisé l'organigramme sur 3 partie :

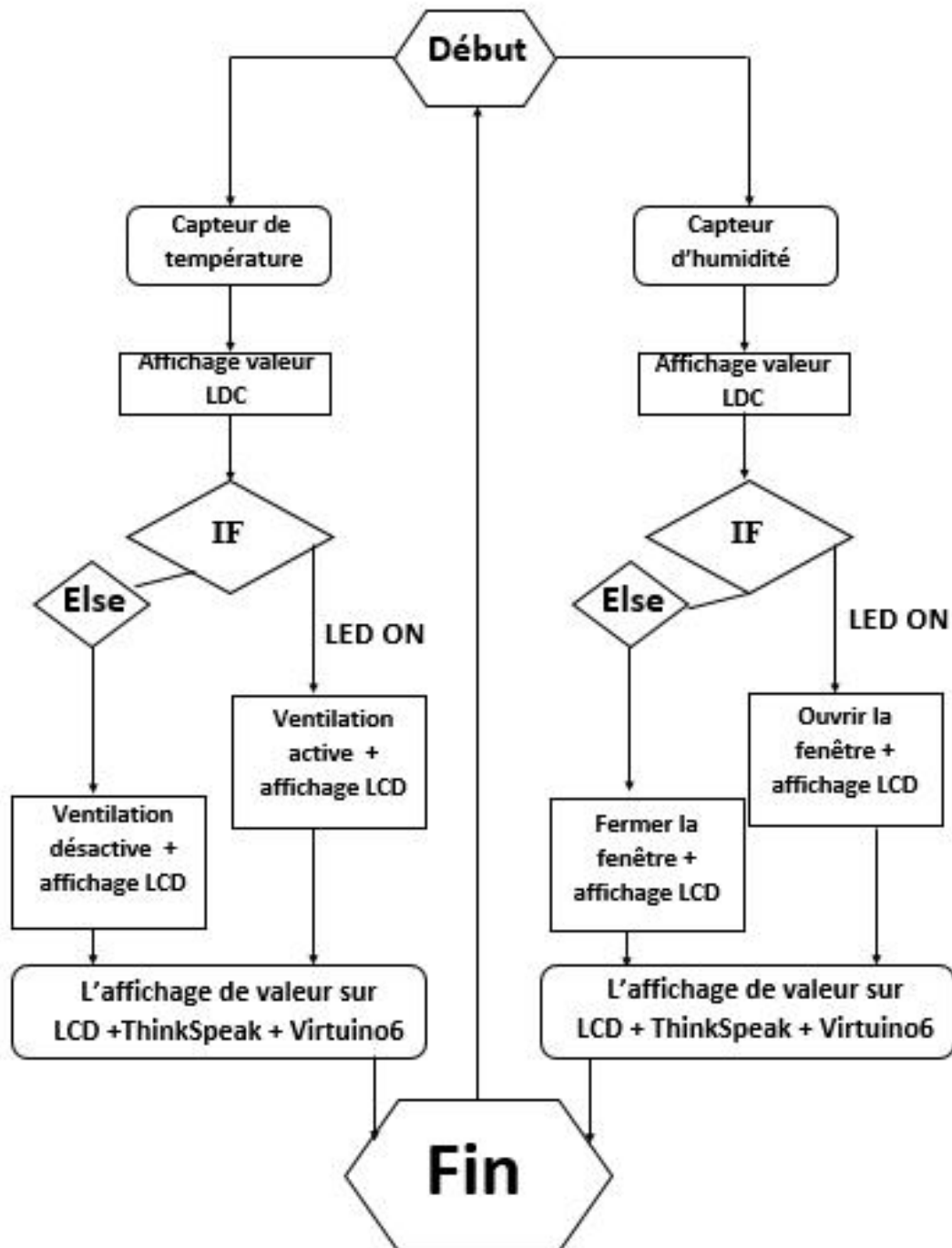


Figure17 : L'organigramme 1

Le capteur DHT11 prend les valeurs de la température et l'humidité cela permet à l'Arduino a l'affichage de ces deux, selon la valeur max ou l'intervalle les deux actionnaires vont activer ou désactiver (l'ouverture de fenêtre et l'activation de la ventilation), finalement la dernière valeur va afficher sur l'écran LCD, le site web ThingSpeak et l'application Android Virtuino6.

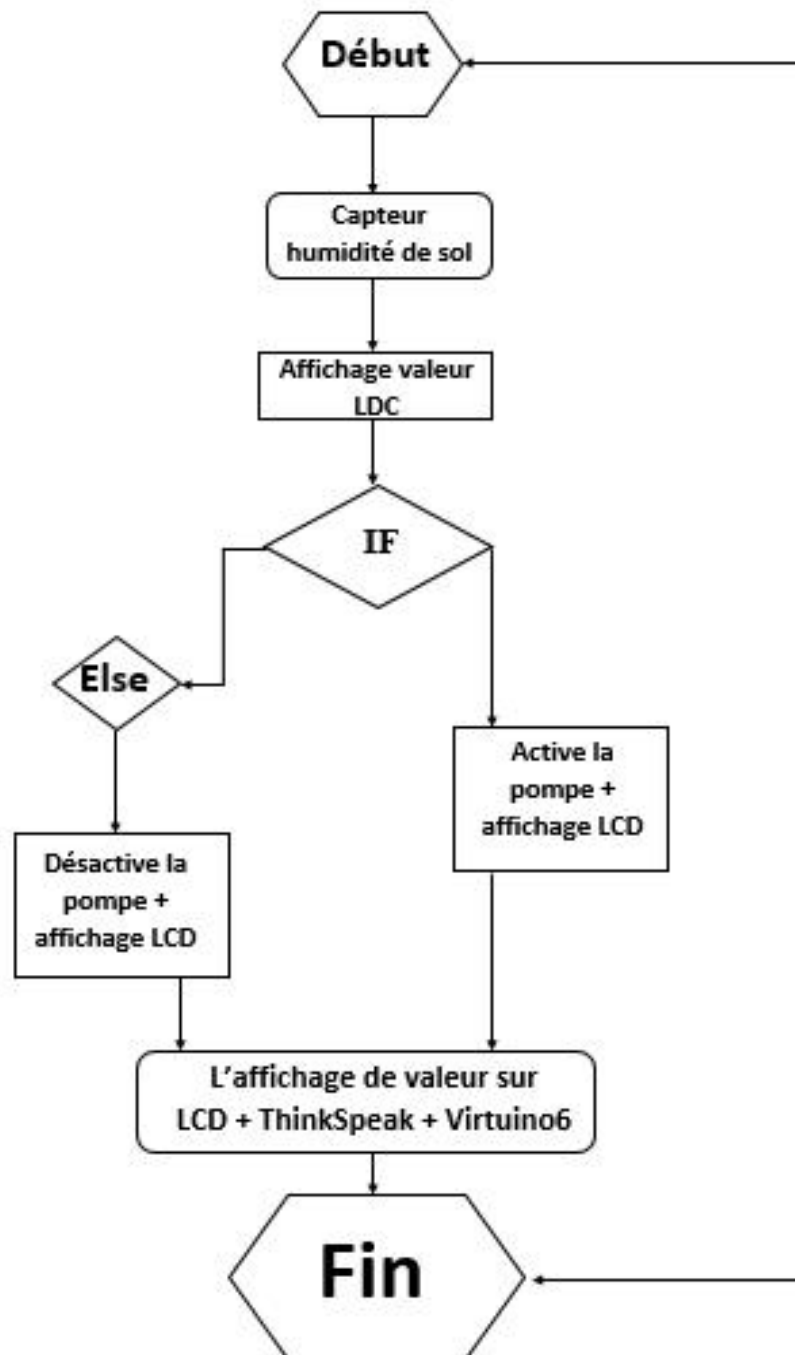


Figure 18: L'organigramme 2

Quand le capteur d'humidité de sol saisi la valeur, l'Arduino agit selon cette valeur (sol sec) il commande la pompe d'eau pour arroser ou pas. Finalement les dernières valeurs vont s'afficher sur l'écran LCD, le site web ThingSpeak et l'application Android Virtuino6.

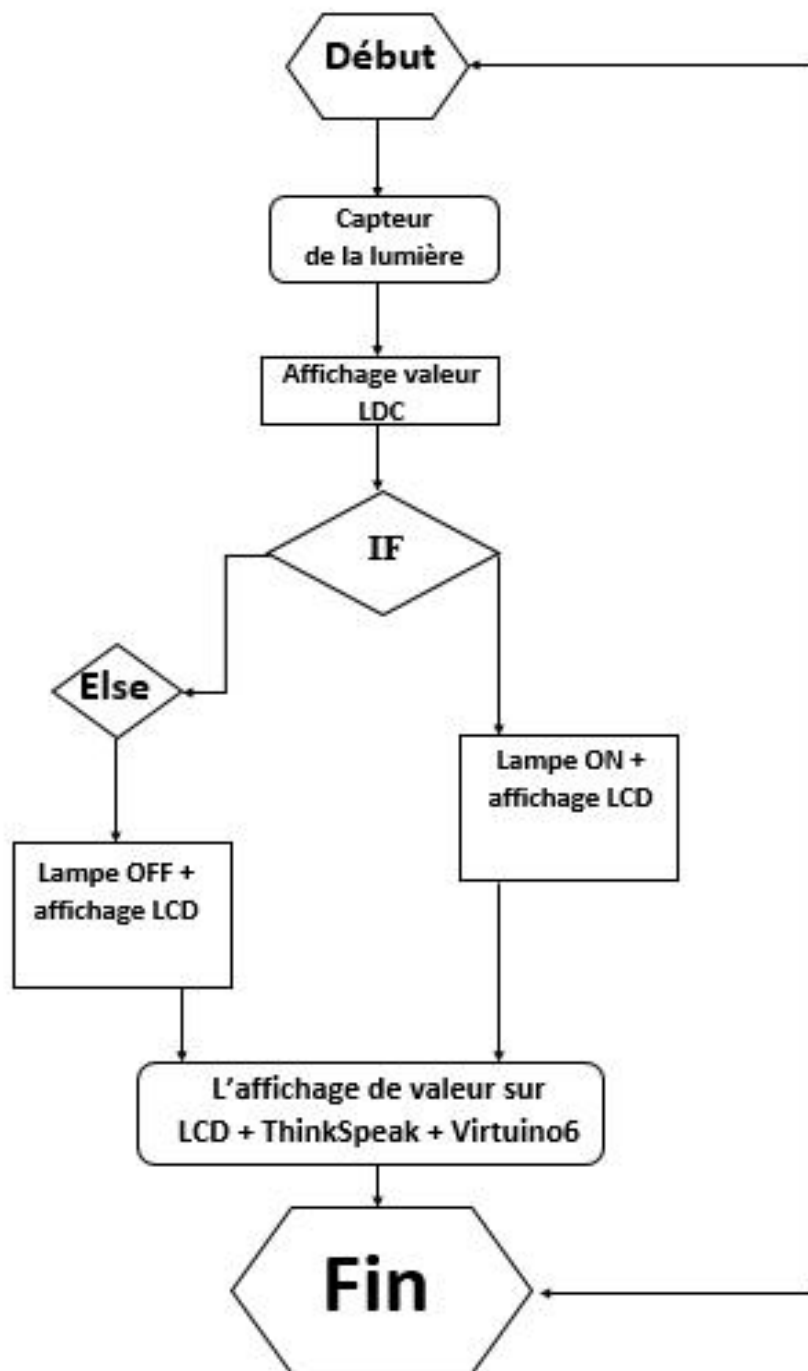


Figure 19: L'organigramme 3

Le capteur de lumière prend la valeur qui sert à signaler l'absence de lumière, l'Arduino affiche la valeur sur LCD et commande la lampe, finalement la dernière valeur ça va être affichée sur l'écran LCD, le site web ThingSpeak et l'application Android Virtuino6.

Remarque :

Nous pouvons changer les valeurs limites et le temps de prendre les valeurs et l'activation des actionnaires selon la plante que nous voulons.

III.4 Schéma électrique du montage sur Fritzing

Dans un premier temps, on dessine le schéma électrique du système avec le logiciel Fritzing. Le logiciel Fritzing est un outil de création de projets et de circuits électroniques, elle permet aussi le traitement des circuits imprimés, elle est disponible gratuitement sur internet. Son objectif principal est de faciliter le remplacement des circuits électroniques libres et de soutenir l'apprentissage de la conception de circuits.

Le logiciel comporte trois vues principales:

- La « Platine d'essai », où l'on voit les composants tels qu'ils sont dans la réalité et où l'on construit le montage.
- La « Vue schématique », représentant le schéma fonctionnel du circuit.
- Le « Circuit imprimé », représentant la vue du circuit imprimé tel qu'il sera sorti en PDF pour être imprimé [9].

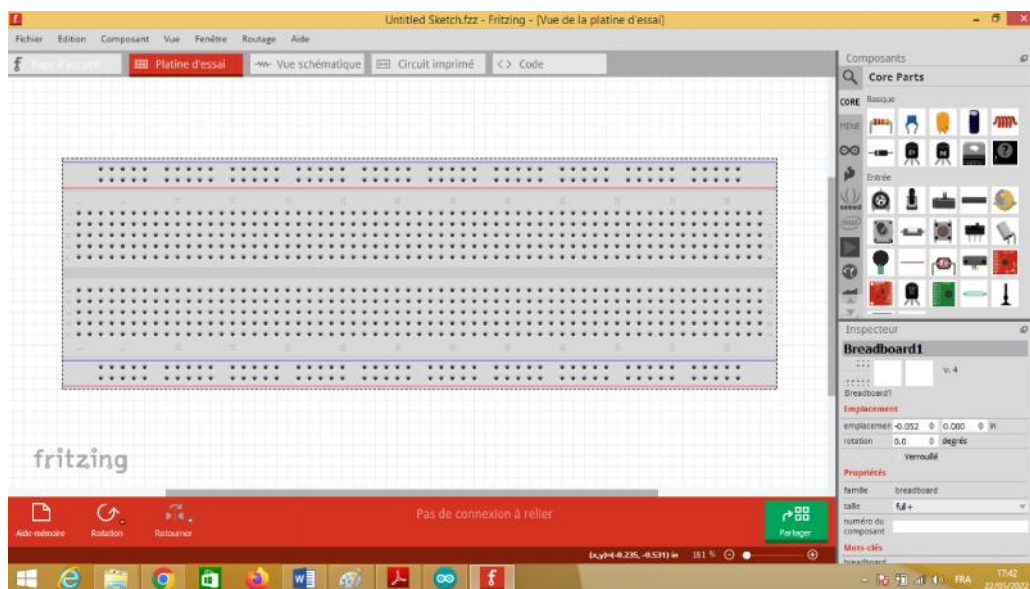


Figure 20 : Interfaçage de logiciel de simulation fritzing .

Dans la figure 21 on présente le montage de notre système sur plaque
D'essai obtenu aussi par Fritzing

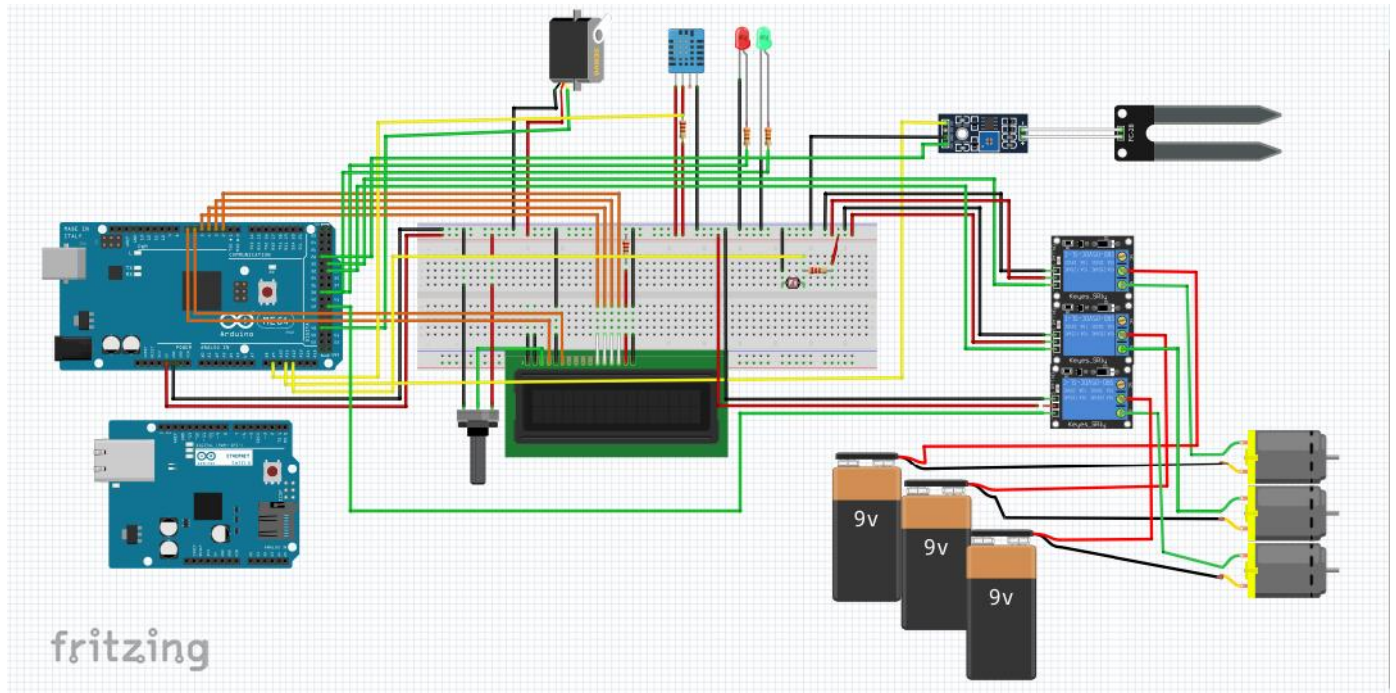


Figure 21 : Montage en plaque d'essai

III.5 Affichage Arduino-PC par câble USB et écran LCD

On trouve que la fonction du capteur utilisés : (DHT11, humidité du sol, LDR) et Les actionnaires Utilisés : (pompe à eau, servomoteur, ventilateur, lampe). Entre ceux-ci, on a câbler le méga Arduino. La serre réalisée est simplement connectée avec un câble USB. La figure montre une image réelle d'une serre intelligente créée sur une carte de test et connectée via un câble USB est présentée à la figure 22.

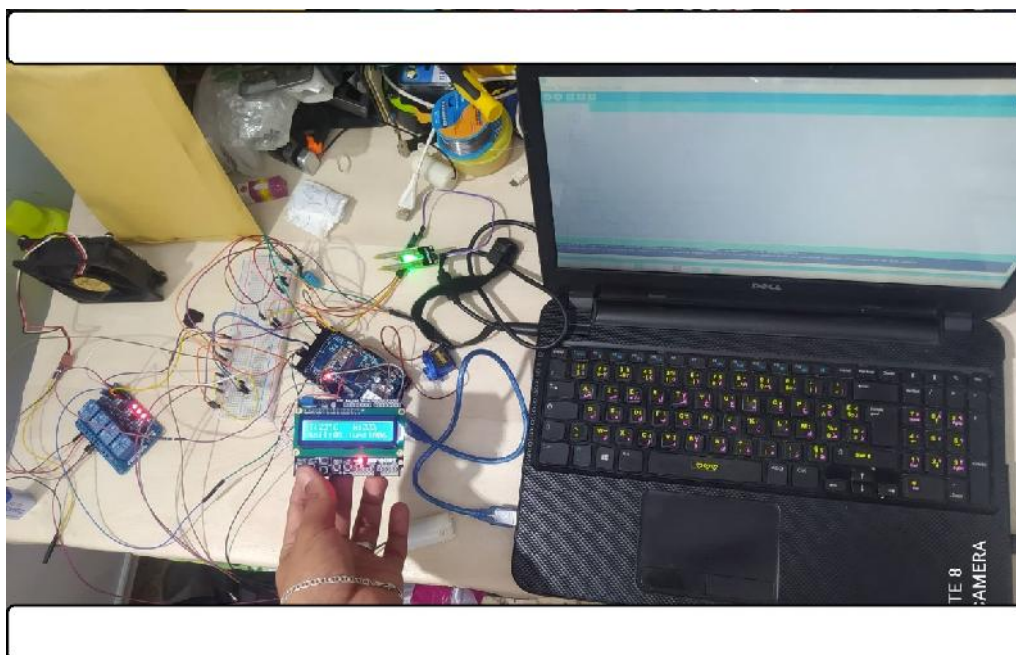
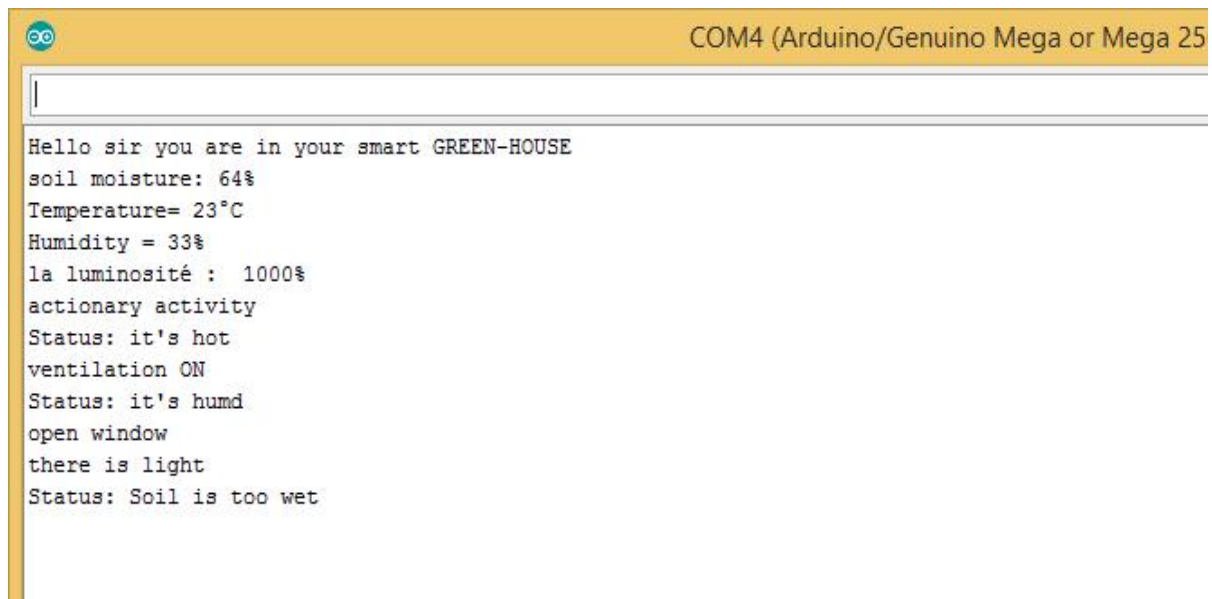


Figure 22 : Image du montage réalisé connecté par USB.

Le programme de gestion des données permet d'afficher toutes les valeurs en temps réel et chaque deux secondes sur une fenêtre du moniteur sérié (Voir figure 23).



```
COM4 (Arduino/Genuino Mega or Mega 25)
Hello sir you are in your smart GREEN-HOUSE
soil moisture: 64%
Temperature= 23°C
Humidity = 33%
la luminosité : 1000%
actionary activity
Status: it's hot
ventilation ON
Status: it's humd
open window
there is light
Status: Soil is too wet
```

Figure 23 : Fenêtre d'affichage des résultats sur le moniteur sérié.

III.6 Choix d'affichage

L'Internet des objets, parfois écrit IdO ou IOT (Internet of things), désigne l'ensemble des infrastructures et technologies mises en place pour faire fonctionner des objets divers par le biais d'une connexion Internet. On parle alors d'objets connectés. Ces objets sont pilotables à distance, le plus souvent à l'aide d'un ordinateur, d'un smartphone ou d'une tablette. Ainsi, le terme Internet des objets regroupe tous les objets et appareils physiques qui possèdent une identité numérique. Il peut s'agir d'objets du quotidien omniprésents dans les logements (télévision, réfrigérateur, machine à laver, système de chauffage, porte de garage électrique), d'appareils ou de systèmes plus complexes comme des véhicules (avions, voitures autonomes) et l'éclairage d'une ville. L'infrastructure crée permet de construire une passerelle entre le monde virtuel et les objets physiques, grâce aux technologies de l'information et de la communication. L'interopérabilité, qui consiste à adapter le comportement d'un objet au comportement d'un autre, est l'une des principales caractéristiques de l'Internet des objets.[10]



Figure 24 : l'utilisation de l'internet des objets

III.6.1 Avantages de l'Internet des objets

- Gagné beaucoup temps.
- Perfection.
- Accessibilité et mobilité : à tout moment, en tout lieu, tout appareil ».
- Les objets connectés feront partis intégrante de notre quotidien dans tous les domaines (Santé, voiture, Life style, météo). [10]

III.6.2 Choix du site d'affichage

Après avoir sélectionné l'Internet des objets pour afficher les résultats L'étape consiste à sélectionner une page d'affichage pour afficher les résultats dans le temps En fait, j'ai choisi le site "Thing Speak". Le site Web ThingSpeak est une entreprise qui offre une variété de services. Dédié à la création d'applications IoT.

- Ce site Web fournit un support IoT en activant les éléments suivants :
- Collecter des données en temps réel (fréquence de 15 secondes ou plus)
- Visualiser les données collectées sous forme de graphique
- Créez des plugins et des applications pour travailler avec les services Web, les réseaux sociaux et d'autres API.

Vous devez créer un projet pour envoyer des informations de la station météorologique à ThingSpeak. Procédez comme suit :

- Enregistrer

- Créer une chaîne Créer 5 "boîtes" sur le canal
- Obtenir la clé de mise à jour (clé API ; clé d'écriture)

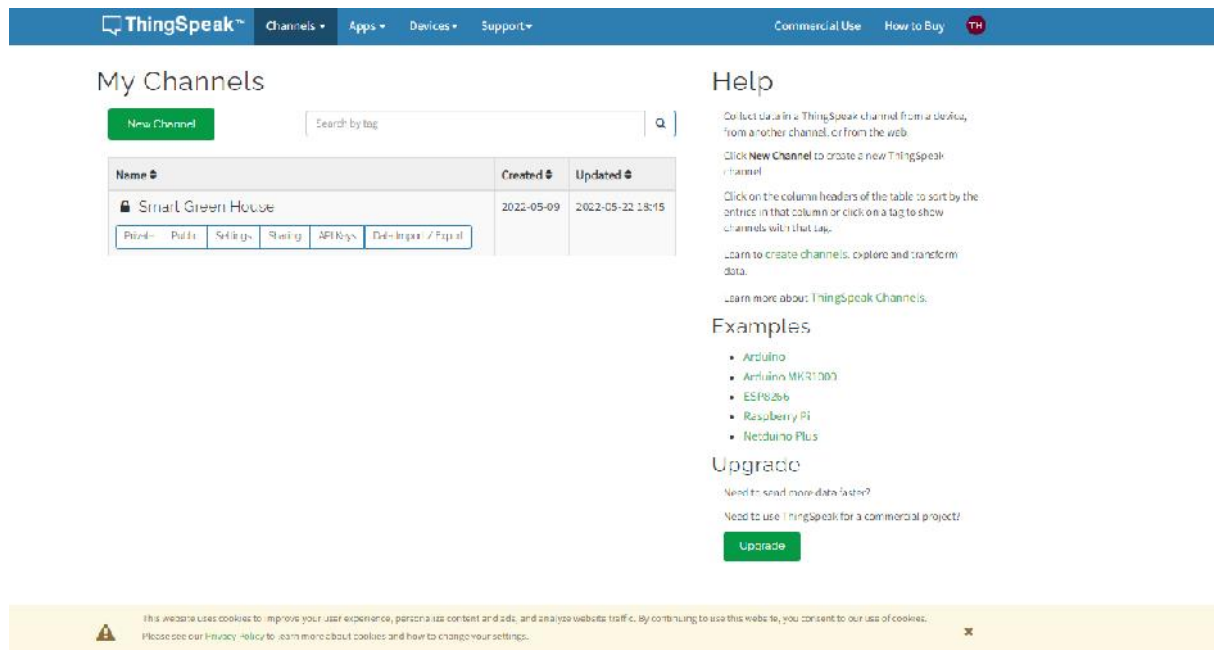


Figure 25 : l'interface de site web (ThingSpeak)

III.7 Suivez la Maison Verte à distance

Après la fin de l'industrie du Smart Green House, nous pouvons la suivre sur ThingSpeak en temps réel et prendre toutes les données de la maison à travers ces données nous pouvons voir toute réaction par les actionnaires sachant que nécessite de la connexion de maison au LAN (router) dans le site choisi.

Nous avons acquis les mesures suivantes :

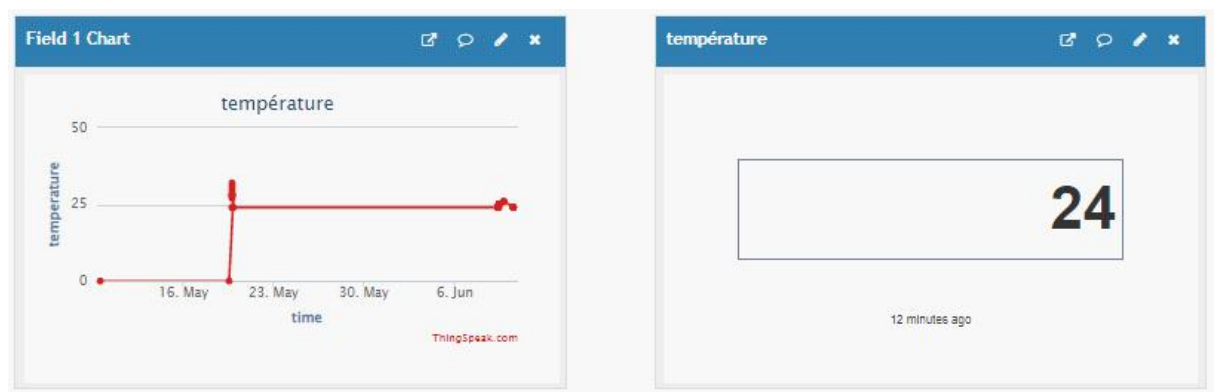


Figure 26 : Courbe graphique exprime le changement de la Température.



Figure 27 : Courbe graphique exprime le changement de l'humidité

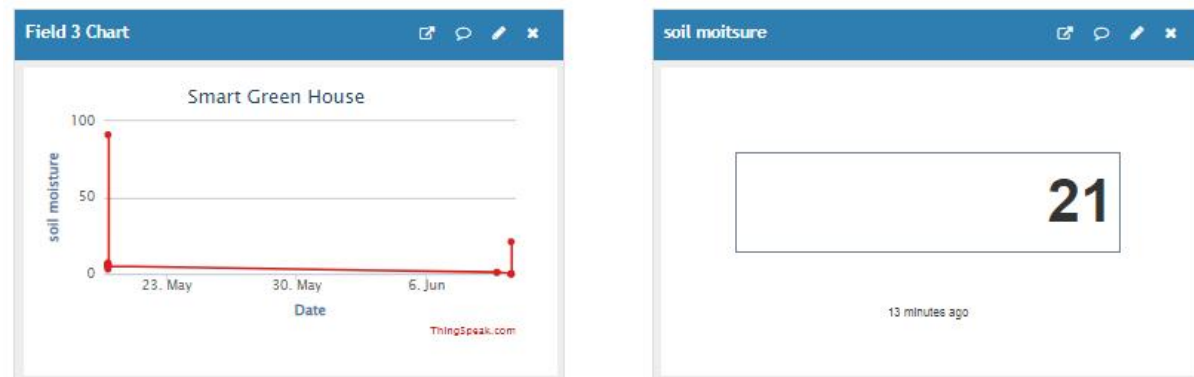


Figure 28 : Courbe graphique exprime le changement de l'humidité de sol.

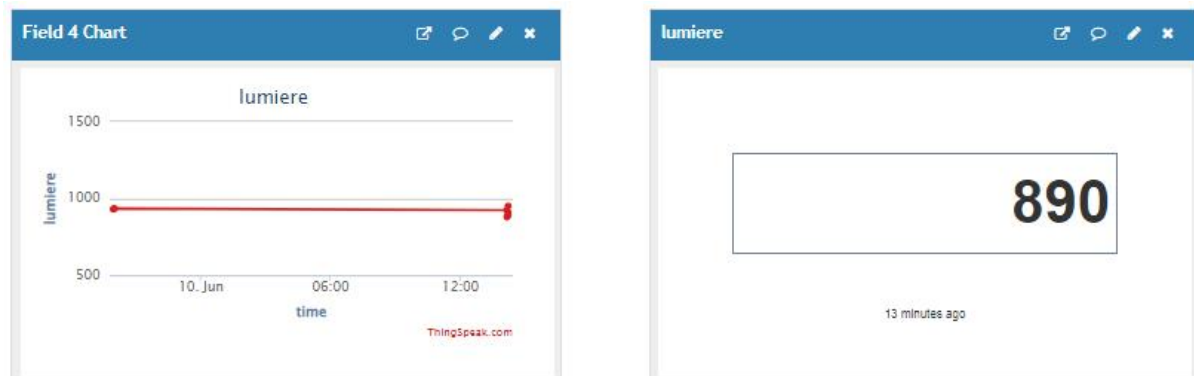


Figure 29 : Courbe graphique exprime le changement de la lumière.

III.8 Utilisation de l'application Virtuino 6

L'application Virtuino 6 nous aide à pré-visualiser les changements dans notre Smart Green House et en permanence tant qu'il y a une connexion Internet en cours. Cette application est une mise à jour de l'ancienne et plus complexe. L'utilisation de cette application est très facile et utilisé dans n'importe quel type de communication.

La méthode de communication avec cette application est la suivante :

- Enregistrer
- Obtenir la clé de mise à jour (Channel ID ; Clé de lecture)
- Lier les courbes créer par les Fields



Figure 30 : logo d'App Virtuino 6

Et les étapes à faire pour ouvrir un compte et l'utiliser :

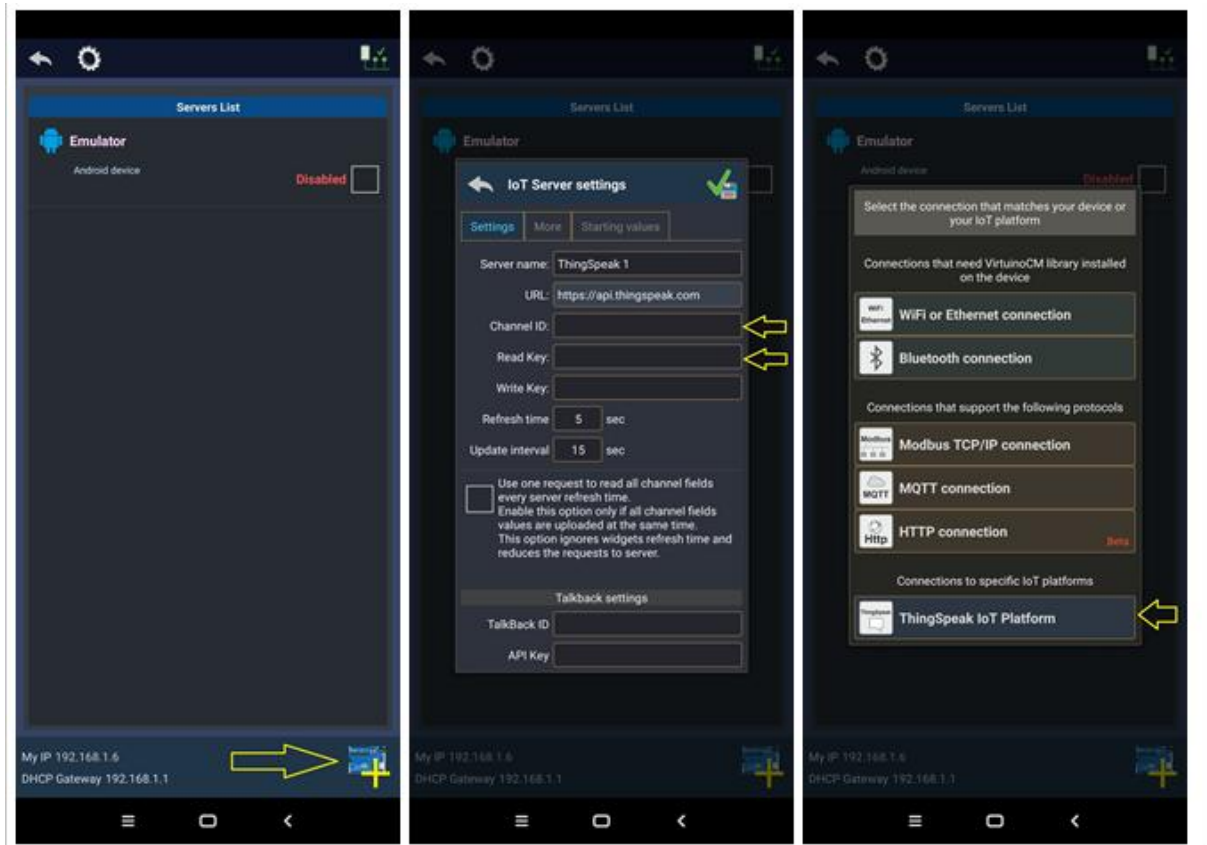


Figure 31 : l'inscription a Virtuino 6

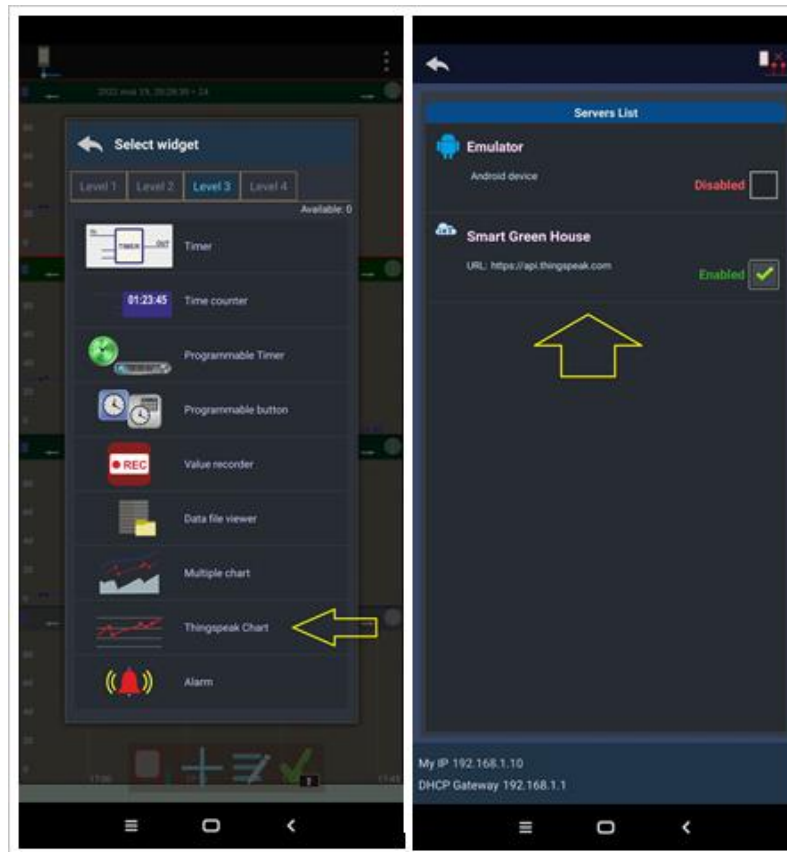


Figure 32 : crée un affichage sur Virtuino 6

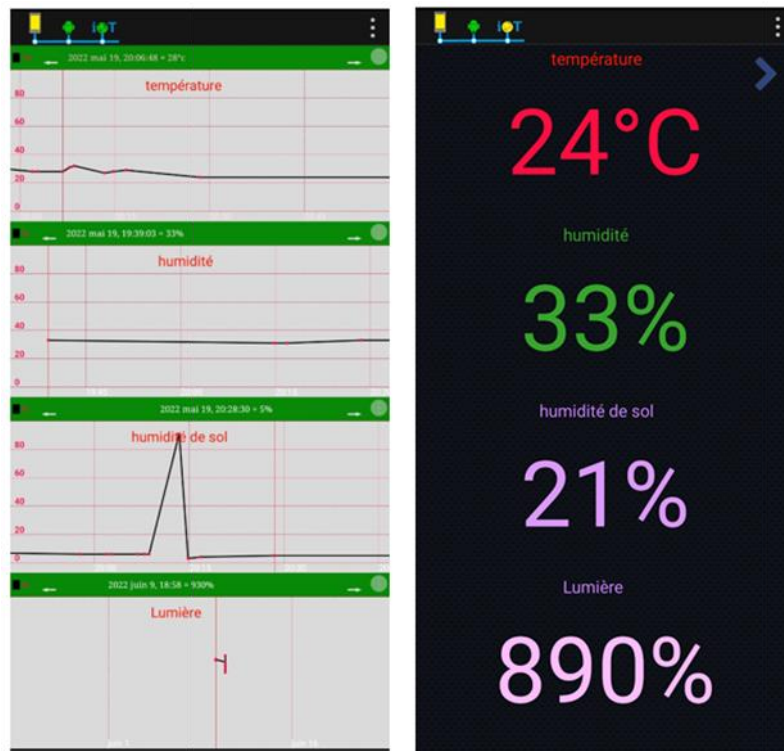


Figure 33 : L'affichage numérique et analogique

III.9 La construction de la maison verte intelligente

Dans ce titre, nous présentons le résultat de la construction de la maison qui est un prototype miniaturisé d'une maison verte intelligente équipée de toutes les technologies



Figure 34 : image réel de maison intelligente

III.10 Conclusion

Dans ce chapitre, nous expliquons comment concevoir Smart Green House et comment cela fonctionne automatiquement avec l'envoi de toutes les données à un site Web et comment utiliser l'application Virtuino 6 aussi.

Nous avons fait des mesures expérimentales de divers phénomènes (température et humidité d'environnement, humidité de sol et la quantité de lumière) et c'est traité par la maison.

Toutes les mesures sont affichées sous forme de graphiques en tant que fonction temps réel.

Conclusion générale

Tout au long de cette étude nous avons tenté de démontrer le rôle d'internet des objets et la réalisation d'un Smart Green House.

Cette recherche nous a permis de nous développer dans le domaine de l'électronique et de son utilisation en agriculture, dans l'ensemble, nous évaluons positivement les effets de " Smart Greene house" dans notre vie quotidienne et améliorer l'économie du pays par une série de réparation au niveau de l'agriculture qui permet d'éviter la dépendance, réaliser la sécurité alimentaire.

Premièrement nous avons fait une réalisation d'un Smart Green house à base d'une carte Arduino Méga , D'après les résultats obtenus de la tests pratiques, l'utilisation de la carte Arduino Méga comme interface entre Les actionnaires , les captures et le PC permet de faciliter la commande de notre système. Puis nous avons essayé de développer notre réalisation avec la carte Shield Ethernet et utiliser la connexion avec l'application Virtuino 6 Et le site web ThingSpeak à distance.

À travers ce projet nous avons acquis une bonne maitrise pour la création d'une application Arduino avec l'environnement Virtuino App après avoir démontré les éléments de contrôle de notre réalisation à distance ou par internet par l'utilisation d'un Shield Ethernet.

Pour conclure, c'est une maison en plastique intelligente qui fonctionne pour trouver ou créer une atmosphère appropriée pour n'importe quelle plante. Comme nous le savons, chaque plante a besoin d'une atmosphère chaude, mauvaise ou modérée pour compléter sa croissance. Cette maison peut cultiver n'importe quelle plante. Elle offre une protection contre les parasites pour devenir de haute qualité et la récolte précoce et parce que la maison en plastique n'a pas besoin d'intervention humaine.

Comme point de vue, on utilise l'API à la place de la carte Arduino Méga avec des câbles réseaux, ce travail peut aussi aider les chercheurs de l'irrigation, qui donne des résultats précis et rapides, pour cette opération on peut ajouter des capteurs de : PH, sel de sol...ect.

Bibliographie

- [1] Gay, Warren. "DHT11 sensor." *Advanced Raspberry Pi*. Apress, Berkeley, CA, 2018. 399-418.
- [2] Bhadani, Prahlad, and Vasudha Vashisht. "Soil moisture, temperature and humidity measurement using Arduino." *2019 9th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*. IEEE, 2019.
- [3] Kumar, Prashant. "Design and implementation of Smart Home control using LabVIEW." *2017 Third International Conference on Advances in Electrical, Electronics, Information, Communication and Bio-Informatics (AEEICB)*. IEEE, 2017.
- [4] Yue, Soon Jian, et al. "IoT based automatic water level and electrical conductivity monitoring system." *2020 IEEE 8th Conference on Systems, Process and Control (ICSPC)*. IEEE, 2020.
- [5] Mohan, M. Chandra, and M. Purushothaman. "Design and fabrication of prosthetic human hand using eeg and force sensor with arduino micro controller." *2017 Third International Conference on Science Technology Engineering & Management (ICONSTEM)*. IEEE, 2017.
- [6] Yasin, Alim, et al. "A Roadmap to Integrate Digital Twins for Small and Medium-Sized Enterprises." *Applied Sciences* 11.20 (2021): 9479.
- [7] Rose, Karen, Scott Eldridge, and Lyman Chapin. "The internet of things: An overview." *The internet society (ISOC)* 80 (2015): 1-50.
- [8] Barrett, Steven F. "Arduino microcontroller processing for everyone!." *Synthesis Lectures on Digital Circuits and Systems* 8.4 (2013): 1-513.
- [9] Nivedan, V., and R. Kannusamy. "Weather monitoring system using IoT with Arduino Ethernet Shield." *International Journal for Research in Applied Science & Engineering Technology(IJRASET)* 7.1 (2019): 2321-9653.
- [10] Monk, Simon. *Fritzing for inventors: take your electronics project from prototype to product*. McGraw-Hill Education, 2016.

Annexe

Programmation de l'Arduino :

```
#include <Servo.h>

#include <LiquidCrystal.h>

LiquidCrystal lcd(8, 9, 4, 5, 6, 7);

#include <dht.h>

dht DHT;// Creates a DHT object

Servo window;

int pos;

#define sensorled 12

#define sensorvalue A11

#define DHT11 A9


#define ventilation 33

#define pompe 31

#define LDR A8

#define lampe 35


#define soilWet 50

#define soilDry 10

#define TMAX 22

#define HMAX 25

#define HMIN 20

#define LIGHT 1000


void setup() {

    Serial.begin(9600);

    lcd.begin(16,2);

    window.attach(48);


    pinMode(sensorled, OUTPUT);
```

```

pinMode(39, OUTPUT);
pinMode(38, OUTPUT);
pinMode(ventilation, OUTPUT);
pinMode(pompe, OUTPUT);
pinMode(lampe, OUTPUT);

pinMode(LDR, INPUT);
digitalWrite(sensorled, HIGH);
Serial.println("Hello sir you are in your smart GREEN-HOUSE ");
lcd.setCursor(0,0);
lcd.print("  WELCOM I AM");
lcd.setCursor(0,1);
lcd.println("YOUR GREEN-HOUSE");
delay(2000);
lcd.clear();
}

void loop() {
// lcd.backlight();
int HumTemp = DHT.read11(DHT11);
int t = DHT.temperature;
int h = DHT.humidity;
int sm = analogRead(sensorvalue);
int moisture = map(sm,1023,210,0,100);
int lum = analogRead(LDR);
delay(2000);
Serial.print("soil moisture: ");
Serial.print(moisture);
Serial.println("%");
lcd.setCursor(0,1);
lcd.print("soil:");
lcd.print(moisture);

```



```

lcd.print("%");
//////////
lcd.setCursor(0,0);
lcd.print("T:");
lcd.print(t);
lcd.print((char)223);
lcd.print("C");
Serial.print("Temperature= ");
Serial.print(t);
Serial.println("°C");
//////////
lcd.setCursor(9,0);
lcd.print("H:");
lcd.print(h);
lcd.print("%");
Serial.print("Humidity = ");
Serial.print(h);
Serial.println("%");
//////////
Serial.print("la luminosité : ");
Serial.print(lum);
Serial.println("%");
lcd.setCursor(8,1);
lcd.print("lum:");
lcd.print(lum);
lcd.print("%");
//////////
delay(5000);
lcd.clear();
if (t < TMAX) {
    digitalWrite(39, LOW);
    digitalWrite(ventilation, LOW);
}

```

```

Serial.println("tempirateur: it's perfect");
// quand la temperature <<TMAX
}
else {
    Serial.println("actionary activity");
    digitalWrite(39, HIGH);
    Serial.println("Status: it's hot");
    Serial.println("ventilation ON");
    digitalWrite(ventilation, HIGH);
    lcd.setCursor(0,0);
    lcd.print("it's hot");
    lcd.setCursor(0,1);
    lcd.print("ventilation ON");
    delay(2000);
    lcd.clear();
}
if (h >= HMIN && h < HMAX) {
    digitalWrite(38, LOW); // turn the LED OFF
    Serial.println("humidity: it's perfect");
    for (pos = 0; pos <= 90; pos += 1) {
        window.write(pos);
    }
    delay(1000);
    window.detach();
}
else {
    digitalWrite(38, HIGH);
    Serial.println("Status: it's humd");
    Serial.println("open window");
    lcd.setCursor(0,0);
    lcd.print("it's hum");
    lcd.setCursor(0,1);

```

```

lcd.print("open window");

delay(2000);

for (pos = 90; pos >= 0; pos -= 1) {
  window.write(pos);
}

delay(1000);

window.detach();

lcd.clear();
}

if (lum > LIGHT) {
  digitalWrite(lampe, LOW);
  Serial.println("there is light");
  // quand la temperature <<TMAX
}

else {
  Serial.println("Status: it's dark");
  Serial.println("lump ON");
  digitalWrite(lampe, HIGH);
  lcd.setCursor(0,0);
  lcd.print("it's dark");
  lcd.setCursor(0,1);
  lcd.print("lump on");
  lcd.clear();
}

if (moisture > soilWet) {
  digitalWrite(pompe, LOW);
  Serial.println("Status: Soil is too wet");
}

else {
  for(int arrosage = 1 ; arrosage <= 3 ; arrosage = arrosage+1)

```

```

{int sm = analogRead(sensorvalue);
  if (moisture < soilDry)
  {
digitalWrite(pompe, HIGH);
delay(2000);
digitalWrite(pompe, LOW);
delay(1000);
Serial.println("Status: Soil is too dry - time to water!");
lcd.print("it's dry pomp on");
  delay(2000);
  lcd.clear();
  }
else
  {
    Serial.println(" watering finish ");
    lcd.setCursor(0,1);
    lcd.print("water finish");
    delay(500);
    lcd.clear();
  }
}
}
}
}

```

Programmation de Ethernet :

```

#include <ThingSpeak.h>
#include <Wire.h>
#include <SPI.h>
#include <Ethernet.h>
#include <LiquidCrystal.h>
LiquidCrystal lcd(8, 9, 4, 5, 6, 7);
#include <dht.h>
dht DHT;// Creates a DHT object

```

```

#include <Servo.h>

Servo window;

int pos = 0;

// Local Network Settings

byte mac[] = { 0x64, 0x5A, 0x04, 0x69, 0x1E, 0x75 }; // BIA or Physical mac address of
W5100 Shield

byte ip[] = { 192, 168, 1, 100 }; // IP ADDRESS of your device

byte gateway[] = { 192, 168, 1, 1 }; // internet access via router

byte subnet[] = { 255, 255, 255, 0 }; // 24bit Subnet Mask

EthernetServer server(23); //server port


// ThingSpeak Settings

unsigned long SmartGreenHouse = 1728646;

char thingSpeakAddress[] = "api.thingspeak.com";

String writeAPIKey = "KGOADZWGO012378D";

const int updateThingSpeakInterval = 10000;


// Setup the variable

long lastConnectionTime = 0;

boolean lastConnected = false;

int failedCounter = 0;


// Initialize Ethernet Client

EthernetClient client;


#define sensorled 12

#define sensorvalue A11

#define DHT11 A9

#define ventilation 33

#define pompe 31

#define LDR A8

```

```

#define lampe 35

#define soilWet 50
#define soilDry 10
#define TMAX 22
#define HMAX 25
#define HMIN 20
#define LIGHT 1000

void setup()
{
    Serial.begin(9600);
    lcd.begin(16,2);
    window.attach(48);
    pinMode(sensorled, OUTPUT);
    pinMode(39, OUTPUT);
    pinMode(38, OUTPUT);
    pinMode(ventilation, OUTPUT);
    pinMode(pompe, OUTPUT);
    pinMode(LDR, INPUT);
    pinMode(lampe, OUTPUT);

    digitalWrite(sensorled, HIGH);
    Serial.println("Hello sir you are in your smart GREEN-HOUSE ");
    lcd.setCursor(0,0);
    lcd.print("  WELCOM I AM");
    lcd.setCursor(0,1);
    lcd.println("YOUR GREEN-HOUSE");
    delay(2000);
    lcd.clear();

    // start the Ethernet connection and the server:

```

```

Ethernet.begin(mac, ip, gateway, subnet);
server.begin();
Serial.print("server is at ");
Serial.println(Ethernet.localIP());
ThingSpeak.begin(client);

// Start Ethernet on Arduino

startEthernet();
}

void loop()
{

// lcd.backlight();
int HumTemp = DHT.read11(DHT11);
int t = DHT.temperature;
String temp = String (t);
int h = DHT.humidity;
String humid = String (h);
int sm = analogRead(sensorvalue);
int moisture = map(sm,1023,210,0,100);
String soilmoisture = String (moisture);
int lumiere = analogRead(LDR);
String caplumiere = String(lumiere);
//delay(2000);
Serial.print("soil moisture: ");
Serial.print(moisture);
Serial.println("%");
lcd.setCursor(0,1);
lcd.print("soil:");
lcd.print(moisture);

```

```

lcd.print("%");
//////////
lcd.setCursor(0,0);
lcd.print("T:");
lcd.print(t);
lcd.print((char)223);
lcd.print("C");
Serial.print("Temperature= ");
Serial.print(t);
Serial.println("°C");
//////////
lcd.setCursor(9,0);
lcd.print("H:");
lcd.print(h);
lcd.print("%");
Serial.print("Humidity = ");
Serial.print(h);
Serial.println("%");
//////////
Serial.print("la luminosité : ");
Serial.print(lumiere);
Serial.println("%");
lcd.setCursor(8,1);
lcd.print("lum:");
lcd.print(lumiere);
lcd.print("%");
//////////
delay(5000);
lcd.clear();
if (t < TMAX) {
    digitalWrite(39, LOW);
    digitalWrite(ventilation, LOW);
}

```



```

Serial.println("tempirateur: it's perfect");
// quand la temperature <<TMAX

}
else {
    Serial.println("actionary activity");
    digitalWrite(39, HIGH);
    Serial.println("Status: it's hot");
    Serial.println("ventilation ON");
    digitalWrite(ventilation, HIGH);
    lcd.setCursor(0,0);
    lcd.print("it's hot");
    lcd.setCursor(0,1);
    lcd.print("ventilation ON");
    delay(2000);
    lcd.clear();
}
if (h >= HMIN && h < HMAX) {
    digitalWrite(38, LOW);
    Serial.println("humidity: it's perfect");
    for (pos = 0; pos <= 90; pos += 1) {
        window.write(pos);
    }
    delay(1000);
    window.detach();
}
else {
    digitalWrite(38, HIGH); // turn the LED ON
    Serial.println("Status: it's humd");
    Serial.println("window open");
    lcd.setCursor(0,0);
    lcd.print("it's hum");
}

```

```

    lcd.setCursor(0,1);
    lcd.print("open window");
    delay(2000);
    for (pos = 90; pos >= 0; pos -= 1) {
        window.write(pos);
    }
    delay(1000);
    window.detach();
    lcd.clear();
}

if (lumiere > LIGHT) {
    digitalWrite(lampe, LOW);
    Serial.println("there is light");
    // quand la temperature <<TMAX
}
else {
    Serial.println("Status: it's dark");
    Serial.println("lump ON");
    digitalWrite(lampe, HIGH);
    lcd.setCursor(0,0);
    lcd.print("it's dark");
    lcd.setCursor(0,1);
    lcd.print("lump on");
    lcd.clear();
}

if (moisture > 50) {
    digitalWrite(pompe, LOW);
    Serial.println("Status: Soil is too wet");
}

else {
    for(int arrosage = 1 ; arrosage <= 3 ; arrosage = arrosage+1)

```

```

{int sm = analogRead(sensorvalue);
  if (moisture < soilDry)
  {
digitalWrite(pompe, HIGH);
delay(2000);
digitalWrite(pompe, LOW);
delay(1000);
Serial.println("Status: Soil is too dry - time to water!");
lcd.print("it's dry pump on");
  delay(2000);
  lcd.clear();
  }
else
  {
    Serial.println(" watering finish ");
    lcd.setCursor(0,1);
    lcd.print("water finish");
    delay(500);
    lcd.clear();
  }
}

if (client.available())
{
  char c = client.read();
  Serial.print(c);
}

// Disconnect from ThingSpeak
if (!client.connected() && lastConnected)
{

```

```

Serial.println("...disconnected");
Serial.println();

client.stop();
}

// Update ThingSpeak
if(!client.connected() && (millis() - lastConnectionTime > updateThingSpeakInterval))
{
//  updateThingSpeak("field1="+analogValue0);

updateThingSpeak("field1="+temp+"&field2="+humid+"&field3="+soilmoisture+"&field4=
"+caplumiere);

}

// Check if Arduino Ethernet needs to be restarted
if (failedCounter > 3 ) {startEthernet();}

lastConnected = client.connected();
}

void updateThingSpeak(String tsData)
{
if (client.connect(thingSpeakAddress, 80))
{
client.print("POST /update HTTP/1.1\n");
client.print("Host: api.thingspeak.com\n");
client.print("Connection: close\n");
client.print("X-THINGSPEAKAPIKEY: "+ writeAPIKey +" \n");
client.print("Content-Type: application/x-www-form-urlencoded\n");
client.print("Content-Length: ");

```

```
client.print(tsData.length());  
client.print("\n\n");  
  
client.print(tsData);  
  
lastConnectionTime = millis();  
  
if (client.connected())  
{  
    Serial.println("Connecting to ThingSpeak Server");  
    Serial.println();  
  
    failedCounter = 0;  
}  
else  
{  
    failedCounter++;  
  
    Serial.println("Connection to ThingSpeak failed (" + String(failedCounter, DEC) + ")");  
    Serial.println();  
}  
  
}  
else  
{  
    failedCounter++;  
  
    Serial.println("Connection to ThingSpeak Failed (" + String(failedCounter, DEC) + ")");  
    Serial.println();  
  
    lastConnectionTime = millis();  
}
```

```

}

void startEthernet()
{

  client.stop();

  Serial.println("Connecting to current network please wait...");
  Serial.println();

  delay(1000);
  // Connect to network and obtain an IP address using DHCP
  if (Ethernet.begin(mac) == 0)
  {
    Serial.println("DHCP Failed, reset the device and try again");
    Serial.println();
  }
  else
  {
    Serial.println("Connected to network w/d DHCP");
    Serial.println();
  }
  delay(1000);
}

```