الجمهورية الجزائرية الديمقراطية الشعب
وزارة التعليم العالي والبحث العلمي

UNIVERSITE BADJI MOKHTAR - ANNABA -
BADJI MOKHTAR – ANNABA UNIVERSITY

جامعة باجي مختار – عنابـــــــــة

Faculté : TECHNOLOGIE

Département : ELECTRONIQUE

Domaine : SCIENCES ET TECHNIQUES

Filière : Automatique

Spécialité : Automatique et systèmes

Mémoire

Présenté en vue de l'obtention du Diplôme de Master

Thème :

Conception et contrôle d'une serre agricole à base d'arduino

Présenté par : Heni Mouhamed Yacine
Attoui Seifédin

Encadrant : Chaker Karima M.C.B Badji Mokhtar -Annaba-

Jury de Soutenance :

Kherfane Hamid	Prof	Badji Mokhtar -Annaba-	Président
Chaker Karima	M.C.B	Badji Mokhtar -Annaba-	Encadrant
Ait Izem Tarek	M.C.B	Badji Mokhtar -Annaba-	Examinateur

Année Universitaire : 2021/2022

Sommaire

CHAPITRE IV : Simulation de la serre avec Arduino

Liste des abréviations

ADC : **A**nalog to **D**igital **C**onverter

CPU : **C**entral **P**rocessing**U**nit

GND: The Ground ou la mass

HR : **h**umidité **r**elative

I2C : **I**nter **I**ntegrated**C**ircuit

IDE : **I**ntegrated**D**evelopment**E**nvironment Impulsion PWM : largeur d'impulsion modulée

LCD : **L**iquid**C**rystal **D**isplay ou afficheur à cristaux liquide

LDR : **L**ight-**D**ependent**R**esistor ou cellule photoconductrice

LED :Light-**E**mitting**D**iode

RAM : **R**andom**A**ccess **M**emory ou mémoire à accès direct

RCSF : **R**éseau de **C**apteur **S**ans **F**il

RTC : **R**eal **T**ime **C**lock ou horloge temps reel

SRAM : **S**tatic**R**ead **A**ccess **M**emory

SPI : Interface **S**érie **P**ériphérique:**S**erial **P**eripheral**I**nterface Bus

TWI : **T**wo**W**ire**I**nterface

VCC : **V**oltage **C**ommon **C**ollector

VDD : **V**oltage **D**rain **D**rain

VSS : **V**oltage **S**ource **S**ource

Liste des figures

Introduction Générale

De nos jours, nous trouvons des légumes, des fruits, des fleurs et des plantes hors saison ; hors de leurs cycles physiologiques. Ils ne peuvent se manifester normalement que si des conditions favorables et particulières sont réunies et bien gérées.

La culture sous serre est un monde de production intensive qui exige que les facteurs de production soient maximisés afin d'assurer une rentabilité. Dans toutes les serres, ils existent toujours des périodes pendant lesquelles, la température, l'humidité, la lumière à l'intérieur de la serre deviennent extrêmement dangereux pour la plante. Ceci arrive à cause de l'incapacité de l'homme à avoir des jugements précis et rapides.

Dans ce sens, nous proposons un dispositif piloté par la carte Arduino qui joue un rôle régulateur des paramètres microclimatiques. Des que certains paramètres sont en dessous ou en dessus d'une limite fixée préalablement (consignes) une action adéquate est générée. La description du travail réalisé est sanctionnée par un mémoire structuré en quatre chapitres :

- Le premier chapitre comporte une présentation générale des serres agricoles.

- Le deuxième chapitre présente le système utilisé pour le contrôle et la gestion climatique en se basant sur des réseaux de capteurs.

- Le troisième chapitre est consacré à une étude microcontrôleur Arduino, en détaillant ses buts, son architecture et ses caractéristiques.

- La partie centrale de notre travail est présentée dans le quatrième chapitre. Elle a pour objectif la réalisation du système voulu pour le contrôle et la gestion microclimatique de la serre. Nous allons d'abord décrire le logiciel que nous avons utilisé dans le cadre de notre application. Il s'agit du logiciel «Proteus Professionnelle 8 » qui est le logiciel de simulation des circuits électronique. Ensuite, nous allons présenter notre carte suivie de la présentation des résultats de la simulation.

Enfin, nous allons couronner notre travail par une conclusion.

Chapitre I

Généralités sur les Serres

I.1 Introduction

La plupart des plantes qui poussent en plein air dans un jardin sont rustiques, c'est-à-dire qu'elles sont adaptées au cycle des conditions météorologiques auxquelles on peut s'attendre dans la région. Quant à la serre, elle est destinée à fournir un milieu plus propice pour la culture des plantes moins rustiques qui auraient du mal à se développer dans les conditions ambiantes locales normales. Les plantes cultivées sous serre ou sous tout autre abri du même genre ne sont généralement pas rustiques et leur réussite tient uniquement, de la couverture de verre ou de plastique et de la source de chaleur artificielle, faute de quoi elles mourraient. La différence essentielle entre la culture en plein air et la culture sous abri repose donc sur une maitrise totale de l'environnement.

Dans ce chapitre, nous allons définir ce qu'est une serre, présenter ses avantages et les différents types et matériaux utilisés pour sa réalisation.

I.2 Définition

Une serre est une structure qui peut être parfaitement close destinée en général à la production agricole. Elle vise à soustraire aux éléments climatiques, les cultures produites pour l'alimentation ou le plaisir de l'homme pour une meilleure gestion des besoins des plantes et pour en accélérer la croissance ou les produire en toute saison. La culture sous serre s'appelle la serri-culture. [01]

Figure I.1 Exemple de serre pédagogique.

I.3 Avantages de la serre

Elle est principalement destinée à protéger du froid les plantes non rustiques et à favoriser la croissance des cultures (légumes, fleurs) en créant des conditions climatiques plus favorables que le climat local, elle permet de palier les problèmes rencontrés lors d'une culture en plein air, nous pouvons citer les avantages majeurs :

- Assure des récoltes précoces ou retardées
- Production plus élevée grâce à la possibilité de contrôler les conditions climatiques de la culture et de favoriser la production à toutes les saisons,
- Augmentation du rendement et de la qualité de la récolte,
- Précocité et retard de la production,
- Réduction de la consommation de fongicides et insecticides.

Ainsi, la serre agricole contribue largement à la modernisation du secteur agricole par l'implémentation de nouvelles technologies. [01]

I.4 Les différents types de serre

Les serres sont de tailles, de forme et de type très divers. Pour répondre à tous les besoins, il existe une multitude de modèle en fonction de la surface disponible au sol et du nombre de plantes à mettre dans la serre. Dans cette partie nous allons présenter quelques modèles.

I.4.1 Serre tunnel

Les serres tunnels, ou encore appelées souterraines, sont destinées avant tout à la production de plants, légumes, fruits, etc. Elles sont proposées à des prix attractifs par rapport aux serres en verre. Une alternative beaucoup plus abordable et efficace aux serres en verre est le wali Pini, un mot indien, qui signifie lieu chaleureux, également connue comme une serre souterraine ou à ciel. D'abord développée dans les années 1980, pour les régions montagneuses froides d'Amérique du Sud, cette méthode permet aux producteurs de maintenir un potager productif toute l'année, même dans les climats les plus froids. La serre est creusée dans le sol, bénéficiant ainsi d'une bonne isolation thermique avec l'air extérieur. Le toit de la serre est incliné perpendiculairement à la hauteur du soleil au solstice d'hiver afin de maximiser l'apport de lumière en saison froide. [01]

Figure I.2 Serre tunnel

I.4.2 Serre en verre polyvalente

Les serres en verre possèdent une structure métallique (aluminium généralement) et un vitrage verre ou polycarbonate. Elles peuvent être peintes (laquées), possèdent souvent une porte coulissante. [01]

Figure I.3 Serre en verre

Elles sont adaptées à la culture de **plantes en pots**, à l'hivernage, etc. Leur agrément est évident, elles peuvent donc servir de véranda, d'endroit de détente dans le jardin, être adossées à la maison, etc. Les serres en verre demandent un budget plus important, du fait de leur matière et de certaines options (portes coulissantes, ouvrants automatiques…).[01]

I.4.3 Mini serres balcon ou châssis

Les mini-serres ont été inventé pour répondre aux besoins des logements modernes. Ne disposant pas d'un jardin, on peut profiter des avantages d'une serre grâce à ces produits. Très compactes, ces serres permettent tout de même en général de faire pousser différents légumes (type tomates), pour le plaisir du jardinage, même en appartement, elles sont souvent rehaussées, et permettent de travailler à hauteur. [01]

I.4.3.1 Serres châssis

De petite taille, la serre châssis sert à la levée des semis, à la culture de plantes aromatiques ou encore pour la protection des fleurs. Elle est efficace contre les intempéries. Elle fait partie des « serres froides », c'est-à-dire qu'on ne chauffe pas. Les serres châssis sont généralement en bois ou en aluminium, avec des panneaux en verre ou en polycarbonate. Leur couvercle se soulève pour accéder facilement au contenu de la serre, mais également pour un renouvellement efficace de l'air. Ces serres sont non chauffées, mais vous pouvez améliorer l'isolation et la protection en période très froide avec des paillasses, des voiles d'hivernage ou encore des plaques de polystyrène Généralement dédiées à la culture en pleine terre (le châssis recouvrant la zone de culture), il existe également des serres châssis équipées d'un fond pour une utilisation sur un balcon ou une terrasse, ou encore en intérieur en tant que décoratif. [01]

Figure I.4 Serre châssis

I.4.3.2 Serres de balcon

Les serres de balcon ou serres de terrasses sont la plupart du temps de serres adossées. Elles sont surtout destinées à la protection des plantes les plus frileuses en périodes froides. Elles peuvent être souples, c'est-à-dire avec un film plastique, généralement transparent, recouvrant une structure en acier à étages, ou plus solides avec une armature en aluminium ou en bois, avec des panneaux de verre ou de polycarbonate. En cas de froid intense, il est possible de chauffer l'intérieur de ces serres en utilisant des petits chauffages électriques. [01]

Figure I.5 Serre de balcon.

I.4.4 Serre mobiles

Les exploitants commerciaux utilisent des serres mobiles qui peuvent être déplacées au-dessus des cultures sur un système de rails, cela facilite leur programme de rotation. [01]

Figure I.6 Serre mobile.

I.5 Structure de serre [02]

En fonction des espèces cultivées, on peut définir trois types de serre :

- **La serre froide :** dont la température peut descendre jusqu'à 4°, à réserver aux plantes non gélives.
- **La serre tempérée :** où l'on peut cultiver des espèces subtropicales non frileuses.
- **La serre chaude :** ou serre tropicale dont la température se situera entre 18 et 26° qui permet de cultiver nombres d'espèces tropicales et autres plantes rarissimes.

Une fois le choix déterminé, il faut se pencher sur les matériaux. Il en existe quatre types pour la structure : le bois, le PVC, l'aluminium, et l'acier.

- **Le bois** est le matériau le plus esthétique et le meilleur isolant thermique, la qualité du bois est un critère de longévité, une serre en bois devra être isolée du sol par un support en briques, évitant ainsi tous les problèmes de dégradation de la base.
- **Le PVC** est le moins cher de tous les matériaux, c'est en outre un bon isolant, qui limite la condensation et permet des économies d'énergie, de plus, son entretien est aisé. Cependant il faut savoir que le PVC ternit avec le temps, sa longévité est moindre par rapport au bois ou à l'aluminium et il ne supporte pas un poids très élevé, ce qui l'écarte pour la construction de grandes structures.
- **L'aluminium** est la structure la plus courante, elle apporte une grande résistance notamment aux vents violents. Léger, il nécessite peu d'entretien et ne rouille pas. Les serres en aluminium haut de gamme peuvent avoir une durée de vie d'une centaine d'années. Côtés inconvénients, l'aluminium n'est pas un très bon isolant.

- **L'acier** est idéal quant à lui, pour la construction de très grandes serres car il est souple et résistant, il est rarement destiné à l'usage des particuliers et doit être galvanisé pour éviter la rouille.

En ce qui concerne le choix entre le verre et le polycarbonate (vitrage) on a.

- **Le verre** plus lourd, est néanmoins plus translucide et meilleur vecteur de luminosité.
- **Le polycarbonate alvéolaire** plus léger et plus isolant, il est aussi plus résistant aux chocs en cas de grêle. Mais il craint les vents violents et il faudra le changer au bout d'une dizaine d'années car il a tendance à devenir opaque.

I.6 Conclusion

Après avoir présenté les différents types de serres ainsi que leurs rôles et leurs avantages nous allons mettre en évidence dans le chapitre qui suit les informations nécessaires au fonctionnement d'un système automatisé, ce qui nous permettra de commander une serre automatisée.

CHAPITRE II

Contrôle et gestion

Climatique de la serre

II.1 Introduction

Afin de garantir un meilleur produit alimentaire, l'intervention technologique dans le domaine de l'agriculture est devenue importante et quasi nécessaire. La serre de culture, était une technique agricole qui permet d'avoir une meilleure qualité du produit même hors saison et qui dépend de multiples paramètres tels que la température et l'humidité ; Cependant elle ne présentait pas jusqu'ici de résultats parfaits du point de vue qualité.

Ceci est dû au fait que ces paramètres ne sont pas calibrés à des valeurs optimales permettant une meilleure production soit dans la qualité gustative du produit soit dans sa pigmentation.
L'intervention technologique qui est le sujet de ce projet de fin d'études se trouve dans le fait de calibrer et optimiser les variables d'entrées de la serre de culture.

II.2 Gestion du climat

La maitrise du climat est la raison d'être des serres ; on peut créer un environnement idéal pour la croissance des plantes.

II.2.1 Gestion d'humidité de sol [03]

- **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 POT-HG ;
- Ampoule (représentant le bouton d'arré) ;
- Moteur (représentant la pompe d'arozage) ;
- Microcontrôleur Arduino.

- **Principe** : Le capteur d'humidité POT-HGdétectent la température et l'humidité ambiantes (d'air et du solrespectivement), et fournissent une lecture au contrôleur. En fonction de latempérature et de l'humidité captée, le contrôleur décide quels appareilsdoivent être allumés.

- **Capteur d'humidité POT-hg :** le potentiomètre est le dispositif à trois bornes utilisé pour mesurer les différences de potentiel en faisant varier manuellement les résistances.

La tension connue est attirée par la cellule ou toute autre source d'alimentation. Le potentiomètre utilise la méthode comparative qui est plus précise que la méthode de déviation. Donc, il est principalement utilisé dans les endroits où une plus grande précision est requise ou où aucun courant ne coule de la source sous test. Le potentiomètre est utilisé dans le circuit électronique, notamment pour contrôler le volume. [03]

Caractéristiques du capteur POT-hg : les caractéristiques majeures du potentiomètre sont :

- Le potentiomètre est très précis car il travaille sur la méthode de comparaison plutôt que sur la méthode du pointeur de déflexion pour déterminer les tensions inconnues.

- Il mesure le point zéro ou le point d'équilibre qui ne nécessite pas de puissance pour la mesure.

- Le fonctionnement du potentiomètre est libre de la résistance de la source car aucun courant ne circule à travers le potentiomètre lorsqu'il est équilibré.

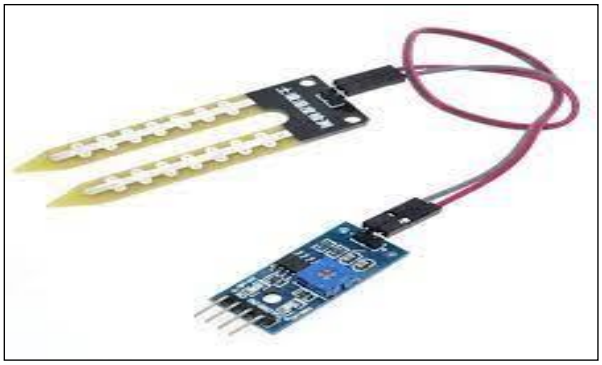


Figure II.1 Capteur d'humidité POT-hg

- **Organigrammes à paramètres commandés pour commande de l'humidité :**

Figure II.2 Grafcet d'humidité

II.2.2 Gestion de la température

- **Réseau de capteur de température :** en ce qui concerne l'air dans laserre il suffit de gérer sa température durant le jour et la nuit, et le réseau decapteur utilisé est structuré autour de :

- Capteur de température de l'air LM35 ;
- Moteur (représentant le ventilateur ou toute unité de réchauffement) ;
- Afficheur LCD ;
- 3 LED (rouge,vert,bleu)
- Microcontrôleur Arduino (voir chapitre III).

- **Principe** : Le capteur de température LM35 détectent la température ambiantes, et fournissent une lecture au contrôleur. En fonction de la température captée, le contrôleur décide quels appareils doivent être allumés. En allumant les dispositifs de chauffage / refroidissement appropriés

- **Capteur de température LM35**

Le capteur de température LM35 est un capteur analogique de température fabriqué par Texas Instruments. Il est extrêmement populaire en électronique, car précis, peu couteux, très simple d'utilisation et d'une fiabilité à toute épreuve.

Le capteur de température LM35 est capable de mesurer des températures allant de -55°C à +150°C dans sa version la plus précise et avec le montage adéquat.

La sortie analogique du capteur est proportionnelle à la température. Il suffit de mesurer la tension en sortie du capteur pour en déduire la température. Chaque degré

Celsius correspond à une tension de +10mV. [04]

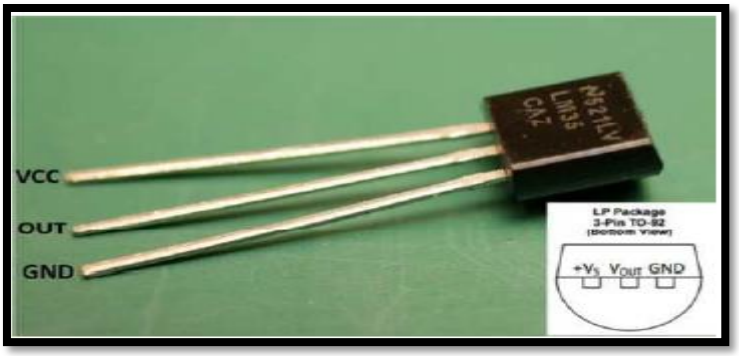


Figure II.3 Le capteur de température LM3

Caractéristiques du capteur :

✓ Alimentation +5 V (3.5 - 5.5 V).

✓ Température : de 0 à 50 °C, précision : ± 2 °C.

✓ Humidité : de 20 à 96 % RH, précision ± 5% RH.

- **Organigrammes à paramètres commandés pour commande de la température :**

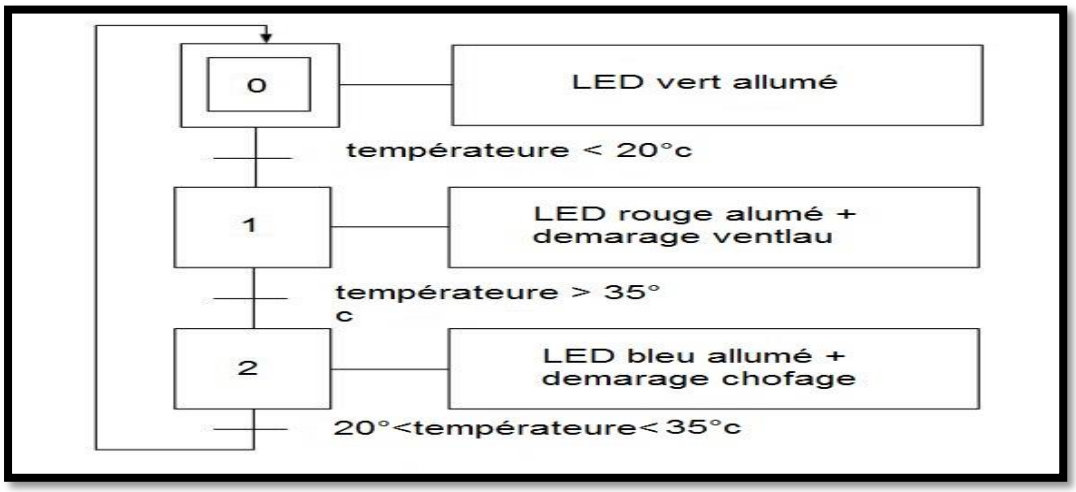


Figure II.4 Grafcet de température

II.2.3 Gestion de la lumière

La lumière a une forte influence sur la croissance des plantes. Comme on a cité dans le premier chapitre, l'exigence de lumière pour une bonne croissance. Pour mettre en évidence les besoins en lumière de la serre, on a pensé à intégrer un réseau de capteur.

- **Réseau de capteur utilisé :**il est structuré autour de :

- Capteur LDR ;
- Une torche (présente la lumière dans notre réalisation);
- Microcontrôleur Arduino ;
- LED (l'éclairage artificielle) ; [04]

- **Principe** : Dans ce cas le réseau de capteur qu'on utilise jouera le rôle d'un régulateur de lumière, en se basant sur une photorésistance LDR, une torche qui joue le rôle de la lumière et une LED. [04]

La commande de la LED est faite à l'aide du capteur LDR.

- **Capteur LDR**

LDR est une abréviation de la résistance dépendante de la lumière. Il est également connu comme photorésistance ou photocellule. Son symbole est montré dans la figure ci-dessous:

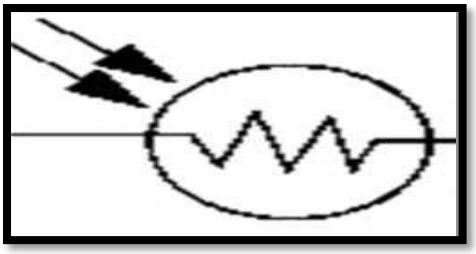


Figure II.5 Capteur LDR

Le capteur LDR est utilisé pour la détection de la lumière. En interne, il possède une résistance sensible à la lumière, comme indiqué dans le symbole.
Chaque fois que la lumière tombe sur le capteur LDR, sa résistance commence à diminuer et quand il fait sombre, sa résistance commence à augmenter. En utilisant la valeur de la résistance, on peut facilement détecter s'il y a de la lumière ou non. [04]

- **Organigrammes à paramètres commandés pour commande de la luminosité**

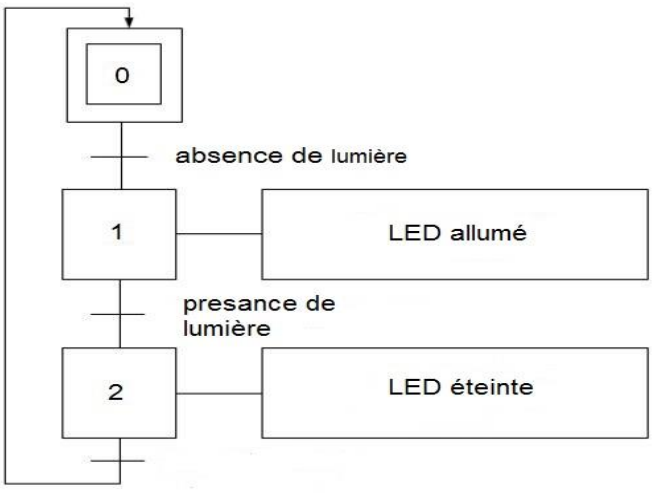


Figure II.6 Grafcet de lumière

II.3 Conclusion

La régulation des paramètres climatiques qui règnent à l'intérieur de la serre demande, tout d'abord, l'acquisition de ses différents paramètres grâce à des dispositifs d'instrumentation industrielles « capteurs » qui convertissent les grandeurs physiques en tensions électriques avant d'entamer par la suite la régulation. Cette régulation est effectuée par l'intermédiaire d'un dispositif de commande qui tend à optimiser les valeurs captées et qui est régi par le type de programmation (ARDUINO). Le fonctionnement global du système part du principe d'un enchainement des différents rôles effectués par les différents blocs qui sont liés entre eux par des connexions caractéristiques (connecteurs, fils électrique et bus).

En effet, ses différents blocs sont présentés tout d'abord par une carte de capteurs qui collectent les informations issues du milieu serriste, ensuite l'information qui a subi la variation sera transmise par la suite au bloc de gestion de tache qui gère cette variation et applique à sa sortie les instructions nécessaires qui vont assurer la régulation de ce paramètre par une injection de cette commande à la terminale effectrice trouvant dans les circuits de commande.

CHAPITRE III

Description de microcontrôleur utilisé

III.1 Introduction

Cette partie est consacrée pour la présentation du microcontrôleur Arduino qui est le type utilisé dans la réalisation de notre projet. On présentera un petit aperçu sur les microcontrôleurs, en se basant sur L'arduino uno, son architecture interne, son brochage et sa communication avec d'autres supports.

III.2 Définition d'un microcontrôleur

En électronique, c'est le "cerveau" de l'appareil qui en possède un. En le programmant, il est capable de prendre des décisions, faire des calculs, compter le temps, gérer l'USB etc. C'est un circuit imprimé qui exécute des instructions et coordonne tous les composants du système. Cependant, il ne faut pas confondre microcontrôleur et microprocesseur. Le microcontrôleur, est un processeur dont les composants ont été suffisamment miniaturisés pour être regroupés dans un unique circuit intégré. [05]

Il possède de multiples avantages, il est tout d'abord moins cher et consomme moins qu'un microprocesseur. De plus le microprocesseur a besoin de multiples éléments externes pour fonctionner comme de la mémoire, des périphériques, une horloge etc. Les microcontrôleurs améliorent l'intégration et le coût d'un système à base de microprocesseurs en rassemblant ces éléments essentiels dans un seul circuit intégré. [05]

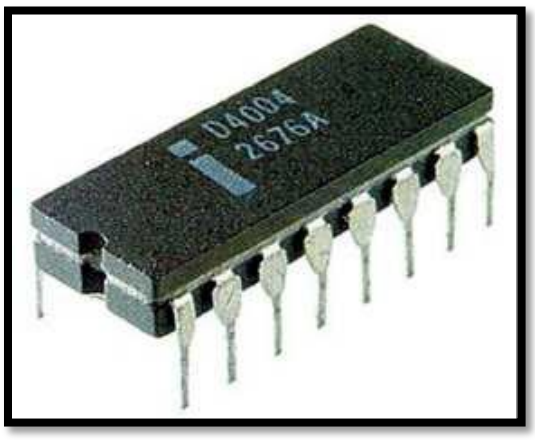


Figure III.1 Exemple du microcontrôleur.

III.3 Microcontrôleur Arduino

L'Arduino est un circuit imprimé en matériel libre sur lequel se trouve un microcontrôleur qui peut être programmé. Les avantages de cette carte sont qu'elle est peu coûteuse, elle facilite les montages électriques, elle possède un environnement de programmation clair et simple et on peut effectuer des tâches très diverses comme la domotique (le contrôle des appareils domestiques - éclairage, chauffage…), qui est le but de notre pratique, le pilotage d'un robot, etc. Nous appelons cela : "une carte magique".[05]

Le système Arduino, nous donne la possibilité d'allier les performances de la programmation à celles de l'électronique. La carte Arduino est programmable dans un langage similaire au C.
La carte Arduino, que ce soit une Uno ou une Nano est composée de plusieurs éléments, dont le principal est le microcontrôleur ATMega328. Cette dernière comporte trois éléments principaux : - L'unité centrale (CPU) - Les mémoires (ROM et RAM) - Des ports d'entrée et sortie. Elle dispose :

- de 14 broches numériques d'entrées/sorties (dont 6 peuvent être utilisées en sorties PWM

- de 6 entrées analogiques (qui peuvent également être utilisées en broches entrées/sorties numériques) ;

- d'un quartz 16Mhz ;

- d'une connexion USB ;

- d'un connecteur d'alimentation jack ;

- d'un connecteur ICSP (programmation "in-circuit") ;

- et d'un bouton de réinitialisation (reset).

Elle contient tout ce qui est nécessaire pour le fonctionnement du microcontrôleur ; Pour pouvoir l'utiliser et se lancer, il suffit simplement de la connecter à un ordinateur à l'aide d'un

câble USB (ou de l'alimenter avec un adaptateur secteur ou une pile, mais ceci n'est pas indispensable, l'alimentation étant fournie par le port USB). [05]

III.3.1 Carte ArduinoUno

"Uno" signifie un en Italien .L'Uno est la carte de base Arduino. Elle sera la plus adaptée pour effectuer des montages simples comme nous voulons faire. C'est une plateforme basée sur une interface entrée/sortie. C'est-à-dire que la carte va pouvoir recevoir des informations (d'un capteur comme dans notre exemple), les traiter à l'aide du microcontrôleur, et renvoyer des instructions à divers composants (comme un moteur, ventilateur…). [05]

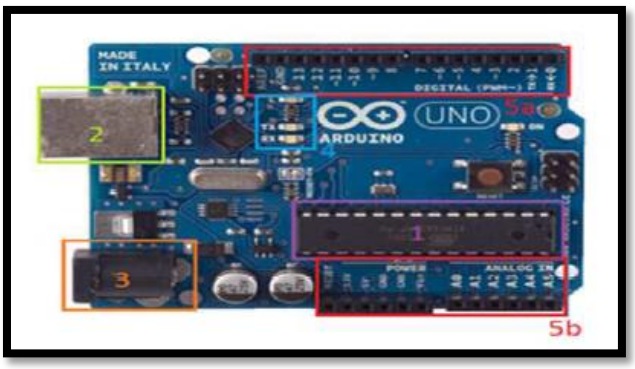


Figure III.2 Carte ArduinoUno.

Le **1**, c'est le cerveau de notre carte. Le microcontrôleur va recevoir le programme, le stocker dans sa mémoire, puis l'exécuter. Grâce à ce programme, il va être capable de faire diverses actions comme : faire clignoter une LED, afficher des caractères sur un écran, envoyer des données à un ordinateur... [05]

Le **2** et le **3** concernent l'alimentation de l'arduino. Il faut savoir qu'en électronique, on utilise la plupart du temps du courant continu : le courant continu est un courant qui circule toujours dans le même sens (du + vers le -). De plus son intensité est constante au cours du temps. Au contraire le courant alternatif (celui qui sort des prises) est un courant qui change de sens. De même son intensité varie au cours du temps. [05]

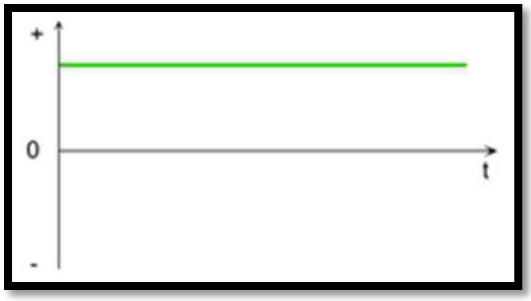


Figure III.3 Courant continu.

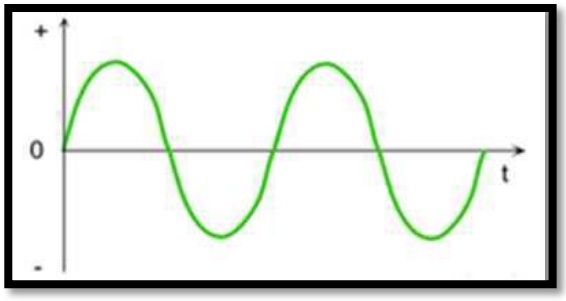


Figure III.4 Courant alternatif.

Le courant continu sera donc plus adapté pour nos montages électroniques qui demandent une quantité d'énergie électrique précise et fixe au cours du temps.

L'arduino doit être alimenté sous 5v, et plus particulièrement le microcontrôleur.
Elle peut être alimentée en 5v par un port USB ou bien par une alimentation externe comprise entre 7v et 12v comme par exemple une pile 9v. La tension est ensuite réduite par un régulateur afin de fournir seulement 5v au microcontrôleur. Ceci est très important car il arrive parfois de "bruler" tous ses composants à cause d'un problème d'alimentation.

Les 3 petits points blancs sont les seuls composants programmables sur la carte. Ce sont en réalité des LED. L'une d'entres elle sert à tester le matériel. À l'aide d'un code simple on peut la faire clignoter. Les deux autres s'allument quand on charge un programme, ou que celui-ci transmet des données.

Les broches sur les côtés de la carte sont les éléments les plus intéressants (**5a** et **5b**). Comme la carte ne possède pas de composants qui peuvent être utiles pour un programme (à part la LED de test) il est nécessaire de les rajouter. Et c'est la qu'intervient toute la puissance de l'Arduino, la carte est complètement modulable.

C'est à dire que l'on peut par exemple brancher à la carte une LED, un chaffage, un ventilateur, un écran, etc.

La différence entre 5a et 5b vient du fait que les broches 5a seront utilisées comme sorties. En revanche les broches 5b sont utilisées comme entrées. C'est à partir d'elles (notamment d'A0 à A5) que le microcontrôleur va recevoir des informations, comme par exemple un capteur qui signale la présence d'une température ambiante. Le microcontrôleur pourra donc réagir en

fonction de cette information et envoyer au chauffage l'ordre de s'arrêter. Le reste des pins d'entrées sont utiles à l'alimentation externe de l'arduino. [05] [06]

III.3.2 Les ports d'entrée et de sortie

Notre microcontrôleur peut maintenant effectuer des opérations, et stocker des résultats, mais il devra aussi communiquer avec le monde extérieur. Par exemple : lorsqu'il reçoit une information d'un bouton poussoir ou qu'il envoie une information à une LED. Ce sont les ports d'entrée et sortie situés sur les côtés de la carte qui servent à cette fin. [06]

- **Brochage de la carte Uno**

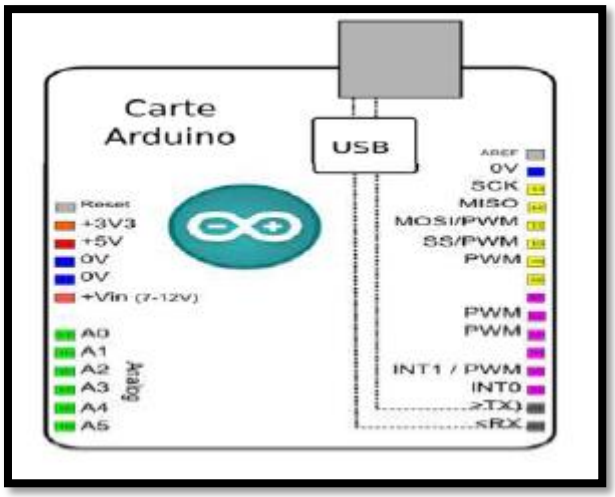


Figure III.5 Brochage de la carte Arduino

- **Entrées et sorties numériques :** Chacune des 14 broches numériques de la carte

UNO qu'on a démontré dans ce qui précède (numérotées des 0 à 13) peut être utilisée soit comme une entrée numérique, soit comme une sortie numérique, en utilisant les instructions **pin Mode** (), **digital write**() et **digital Read ()** du langage Arduino. Ces broches fonctionnent en 5V. Chaque broche peut fournir ou recevoir un maximum de 40mA d'intensité et dispose d'une résistance interne de "rappel au plus" (pull-up) (déconnectée par défaut) de 20-50 KOhms. Cette résistance interne s'active sur une broche en entrée à l'aide de l'instruction **digital write**(broche, **HIGH**). [06]

De plus, certaines broches ont des fonctions spécialisées :

- **Communication Série** : Broches 0 (RX) et 1 (TX). Utilisées pour recevoir (RX) et transmettre (TX) les données séries de niveau TTL. Ces broches sont connectées aux broches correspondantes du circuit intégré ATmega8U2 programmé en convertisseur USB-vers-série de la carte, composant qui assure l'interface entre les niveaux TTL et le port USB de l'ordinateur. [06]

- **Interruptions Externes** : Broches 2 et 3. Ces broches peuvent être configurées pour déclencher une interruption sur une valeur basse, sur un front montant ou descendant, ou sur un changement de valeur. Voir l'instruction **attachInterrupt**() pour plus de détails. [06]

- **Impulsion PWM (largeur d'impulsion modulée)**: Broches 3, 5, 6, 9, 10, et 11.Fournissent une impulsion PWM 8-bits à l'aide de l'instruction analogWrite ().[06]

- **SPI (Interface Série Périphérique)** : Broches 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). Ces broches supportent la communication SPI (Interface Série Périphérique) disponible avec la librairie pour communication SPI. Les broches SPI sont également connectées sur le connecteur ICSP qui est mécaniquement compatible avec les cartes Mga. [06]

- **I2C** : Broches 4 (SDA) et 5 (SCL). Supportent les communications de protocole I2C (ou interface TWI (TwoWire Interface - Interface "2 fils"), disponible en utilisant la librairie Wire/I2C (ou TWI - Two-Wireinterface - interface "2 fils").[06]

- **LED** : Broche 13. Il y a une LED incluse dans la carte connectée à la broche 13. Lorsque la broche est au niveau HAUT, la LED est allumée, lorsque la broche est au niveau BAS, la LED est éteinte. [06]

- **Broches analogiques :** La carte Uno dispose de 6 entrées analogiques (numérotéesde 0 à 5), chacune pouvant fournir une mesure d'une résolution de 10 bits (càd sur1024 niveaux soit de 0 à 1023) à l'aide de la très utile fonction analogRead() dulangage Arduino. Par défaut, ces broches mesurent entre le 0V (valeur 0) et le 5V (valeur

1023), mais il est possible de modifier la référence supérieure de la plage demesure en utilisant la broche AREF et l'instruction analogReference() du langageArduino. [06]

Note : les broches analogiques peuvent être utilisées en tant que broches numériques : elles sont numérotées en tant que broches numériques de 14 à 19. [06]
Autres broches

Il y a deux autres broches disponibles sur la carte :

- **AREF** : Tension de référence pour les entrées analogiques (si différent du 5V).

Utilisée avec l'instruction analogReference().

- **Reset** : Mettre cette broche au niveau BAS entraîne la réinitialisation (= leredémarrage) du microcontrôleur. Typiquement, cette broche est utilisée pour ajouterun bouton de réinitialisation sur le circuit qui bloque celui présent sur la carte.

III.3.3 Les mémoires de la carte ARDUINO UNO [06]

La mémoire est l'un des principaux composants d'un ordinateur ou d'un microcontrôleur comme l'Atmega, coeur de l'arduino.

Dans les microcontrôleurs Arduino, comme dans la plupart des équipements informatiques, on trouve deux types de mémoire :

- la mémoire vive (mémoire de programme) ;
- la mémoire morte (mémoire de données).

La mémoire vive nécessite d'être reliée au courant pour garder les informations alors que la mémoire morte peut conserver des données sans alimentation. On appelle aussi ces mémoires, mémoire de programme et mémoire de données. On peut stocker des informations dans la carte sur la mémoire morte en écrivant des programmes, ou en accédant directement à elle grâce à la fonction EEPROM.
L'ATmega 328 à 32Ko de mémoire FLASH pour stocker le programme (dont

0.5Ko également utilisés par le bootloader). L'ATmega 328 a également 2ko de mémoire SRAM (volatile) et 1Ko d'EEPROM (non volatile - mémoire qui peut être lue à l'aide de la librairie EEPROM).

Nous allons examiner en ce qui suit les 3 types de mémoires disponibles au sein d'un Arduinouno, leurs capacités et leurs rôles afin d'en comprendre les nuances et usages.

Les principales caractéristiques d'une mémoire sont :

- sa capacité de stockage en kilooctet (ko – 2_{10} octets) ou en gigaoctet (Go – 2_{30} octets) pour un pc de bureau ;

- sa vitesse en lecture et écriture (mégaoctet par seconde : Mb/s) ;

- sa capacité à conserver les données dans le temps avec ou sans alimentation électrique ;

- son nombre de cycle d'écriture avant obsolescence.

III.3.4 L'unité centrale [06]

L'unité centrale, aussi appelée Central Processing Unit (CPU) est l'élément primordial du microcontrôleur. C'est le CPU qui décode les instructions, les exécutes et gère les relations entre les mémoires et les ports d'entrée et de sortie.

On peut donc structuré le fonctionnement de l'Arduino comme suit :

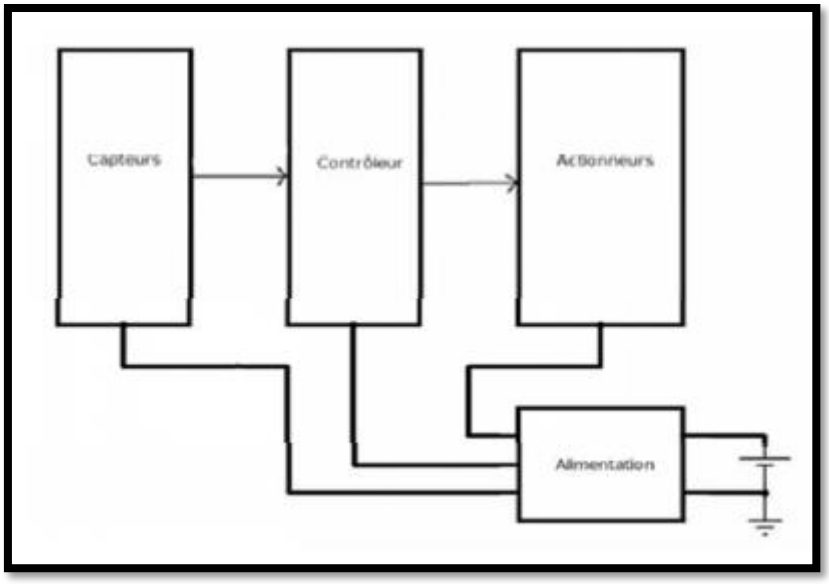


Figure III.6 Principe de fonctionnement d'une carte Arduino.

III.3.5 Alimentation [06]

Les broches d'alimentation sont les suivantes :

- **VIN** : La tension d'entrée positive lorsque la carte Arduino est utilisée avec une source de tension externe (à distinguer du 5V de la connexion USB ou autre source 5V régulée). Vous pouvez alimenter la carte à l'aide de cette broche, ou, sil'alimentation est fournie par le jack d'alimentation, accéder à la tension d'alimentation sur cette broche.

- **5V :** La tension régulée utilisée pour faire fonctionner le microcontrôleur et lesautres composants de la carte (pour info : les circuits électroniques numériques nécessitent une tension d'alimentation parfaitement stable dite "tension régulée" obtenue à l'aide d'un composant appelé un régulateur et qui est intégré à la carte Arduino). Le 5V régulé fourni par cette broche peut donc provenir soit de la tension d'alimentation VIN via le régulateur de la carte, ou bien de la connexion USB (qui fournit du 5V régulé) ou de tout autre source d'alimentation régulée.

- **3V3** : Une alimentation de 3.3V fournie par le circuit intégré FTDI (circuit intégré faisant l'adaptation du signal entre le port USB de votre ordinateur et le port série de l'ATmega) de la

carte est disponible : ceci est intéressant pour certains circuits externes nécessitant cette tension au lieu du 5V). L'intensité maximale disponible sur cette broche est de 50mA.
GND : Broche de masse (ou 0V).

III.3.6 Communication [06]

La carte Arduino Uno dispose de toute une série de facilités pour communiquer avec un ordinateur, une autre carte Arduino, ou avec d'autres microcontrôleurs.

L'ATmega328 dispose d'une UART (Universal Asynchronous Receiver Transmitter ou émetteur-récepteur asynchrone universel en français) pour communication série de niveau TTL (5V) et qui est disponible sur les broches 0 (RX) et 1 (TX). Un circuit intégré ATmega8U2 sur la carte assure la connexion entre cette communication série vers le port USB de l'ordinateur et apparaît comme un port COM virtuel pour les logiciels de l'ordinateur.

Le logiciel Arduino inclut une fenêtre terminal série (ou moniteur série) sur l'ordinateur et qui permet d'envoyer des textes simples depuis et vers la carte Arduino. Les LEDs (RX et TX sur la carte) clignote lorsque les données sont transmises via le circuit intégré USB-vers-série et la connexion USB vers l'ordinateur (mais pas pour les communications série sur les broches 0 et 1).

Une librairie Série Logicielle permet également la communication série (limitée cependant) sur n'importe quelle broche numérique de la carte UNO.

L'ATmega328 supporte également la communication par protocole I2C (ou interface TWI (Two Wire Interface - Interface "2 fils") et SPI :

Le logiciel Arduino inclut la librairie Wire qui simplifie l'utilisation du bus I2C. Pour utiliser la communication SPI (Interface Série Périphérique), il faut utiliser la librairie pour communication SPI.

III.2.7 Programmation [06]

La carte Arduino Uno peut être programmée avec le logiciel Arduino. Il suffit de sélectionner "ArduinoUno" dans le menu Tools >Board (en fonction du microcontrôleur présent sur la carte).

Le microcontrôleur ATmega328 présent sur la carte Arduino Uno est livré avec un bootloader (petit programme de démarrage) préprogrammé qui permet de transférer le nouveau programme dans le microcontrôleur sans avoir à utiliser un matériel de programmation externe. Ce bootloader communique avec le microcontrôleur en utilisant le Protocol original STK500.

III.2.8 Caractéristiques mécaniques [06]

Les longueurs et largeurs maximales de la Uno sont respectivement 6.86 cm et 5.33 cm, avec le connecteur USB et le connecteur d'alimentation Jack s'étendant au-delà des dimensions de la carte. Quatre trous de vis permettent à la carte d'être fixée sur une surface ou dans un boîtier. Noter que la distance entre les broches 7 et 8 est de 0.16 pouces, et non un multiple des 0.1 pouces séparant les autres broches.

III.4 Conclusion

En effet, le microcontrôleur ARDUINO est "une carte magique", elle facilite les montages électriques, en possédant un environnement de programmation clair et simple et effectue des tâches très diverses, et dans notre pratique elle présente l'élément majeur du montage ce qui nous permet de gérer le climat de la serre étudiée.

CHAPITRE IV

Simulation de la Serre

Avec Arduino Uno

IV.1 Introduction

Pour une réalisation d'un circuit électronique, l'étape de simulation est primordiale avant la mise en œuvre du circuit. Pour se faire, ils existent plusieurs logiciels de simulation telle que : EasyEDA, PartSim et ISIS proteus.

Dans ce projet de fin d'étude nous allons utiliser le logiciel « proteus professionnelle version : 8.11» qui est l'outil de simulation le plus récent et utilisé par les électroniciens. Il nous permettra de créer, planifier, dimensionner, et réaliser des liaisons électriques entre les différents composants du circuit.

Notre travail consiste à :

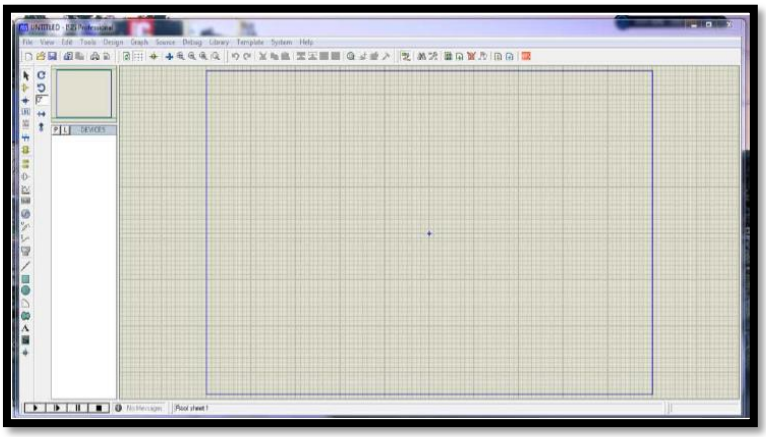
- Créer 3 réseaux de capteurs pour contrôler et gérer les trois facteurs climatiques de la serre en introduisant pour chaque réseau ses propres paramètres;

- Décrire les étapes du programme sous Arduino, puis le compiler.

- Simulation du chaque circuit réalisé.

IV.2 description de l'interface de logiciel utilisé pour la simulation

IV.2 1.Espace de travail



FigureIV.1 L'espace de travail de Proteus 8.

- **Barre de menus :** Cette barre permet de gérer le travail (ouverture, sauvegarde, impression, mode d'affichage, etc.).

FigureIV.2 La barre de menus deProteus 8.

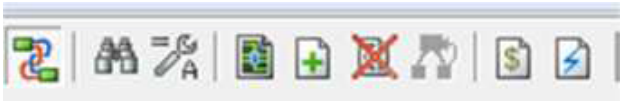
- **Barres d'outils de commande** : Ces barres fournissent un accès équivalent aux commandes des menus. Elles peuvent être masquées par la commande

"Barre d'outils" du menu "Affichage".

Commandes Fichier/Projet.

Commande d'Affichage.

Commande Edition / Bibliothèque.



Commandes Outils.

- **Barre d'outils de sélection de mode** *:* Cette barre permet de sélectionner un outil parmi les 3 modes d'édition disponibles.

Mode Principal.

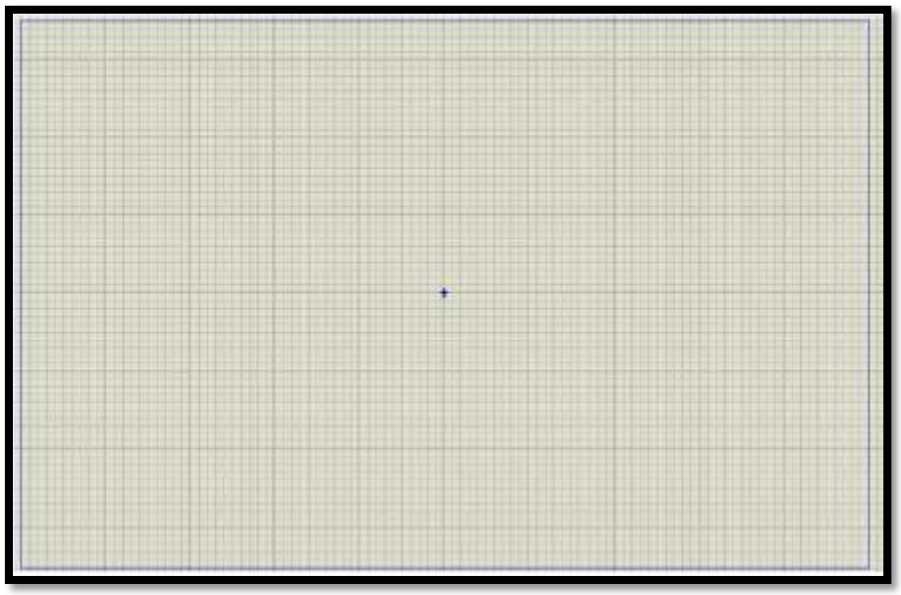
Mode gadgets.

Mode graphique.

- **Barre d'outils d'orientation :** Cette barre permet d'afficher et de contrôler la rotation et la réflexion d'un objet placé ou à placer.

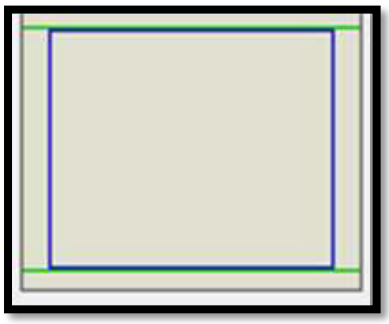
- **Zone d'édition des schémas :** Cet espace rectangulaire correspond à la zone de travail effectif. Tous vos schémas apparaîtront dans cette zone et seront

visualisés avec le coefficient d'agrandissement ou de réduction choisi.



FigureIV.3 Zone d'édition des schémas.

- **Vue d'ensemble du schéma** (cadre extérieur) : et de positionnement (cadreintérieur) Elle montre une représentation simplifiée de la totalité du dessin. Le cadre bleu marque le contour de la feuille, alors que le cadre vert montre la zone du schéma actuellement visible dans la fenêtre d'édition. Dans cette fenêtre, apparaît également l'aperçu d'un objet sélectionné pour un placement.



FigureIV.4 Espace pour vue d'ensemble du schéma.

- **Sélecteur d'objets :** Le sélecteur d'objets liste les différents éléments, selon le mode de travail choisi. Les types d'objets qui peuvent y apparaître sont les composants, les terminaux, les pattes, les symboles graphiques, les marqueurs, les graphes.

FigureIV.5 Icône permettant de sélectionner l'objet.

IV.2.2 Gestion d'un PROJET

- Création d'un projet **:** Menu Fichier Nouveau Projet ou appuyer sur l'icône .

- Ouverture d'un projet : Menu Fichier Ouvrir projet ou appuyer sur l'icône .

- Enregistrement d'un projet **:** Menu Fichier Enregistrer projet sous.

- Sauvegarde d'un projet Menu Fichier Enregistrer projet ou appuyer sur l'icône  .

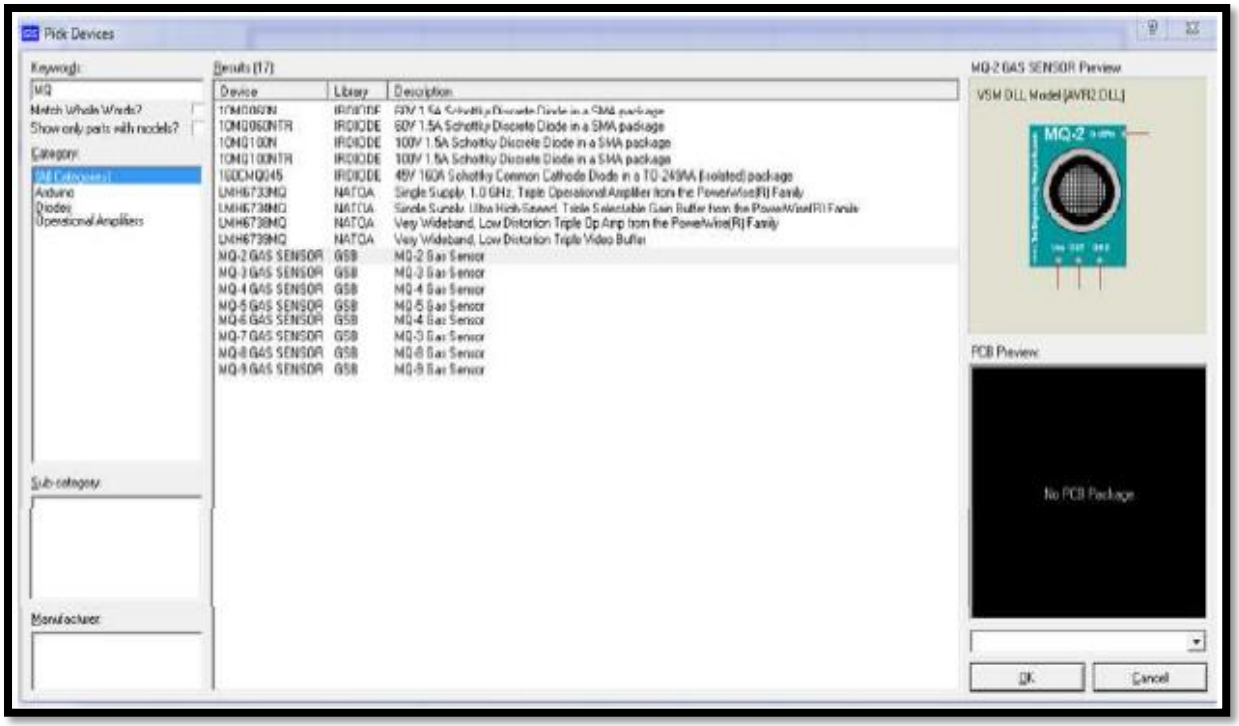
IV.2.3 Edition d'un schéma

- **Chargement de composants :** Avant de commencer un projet, vous devez d'abord pré-charger et mettre en attente les différents composants que vous comptez utiliser. Il est néanmoins possible de les charger plus tard. Les composants disponibles sont regroupés dans des bibliothèques classées par thèmes. Pour charger un composant :

- Cliquer sur l'icône  "Composant de la barre d'outils de sélection de modes.

- Appuyer sur la touche **P** du sélecteur d'objet  .

La fenêtre "Pickdevices" apparaît : Dans le cas où vous ne connaissez pas la catégorie où se trouve le composant désiré, il suffit de taper son nom ou quelques lettres dans la zone "Mots clés" afin que l'application sélectionne certaines catégories susceptibles d'accueillir le composant recherché. Une fois ce choix fait ou si vous connaissez lacatégorie, il suffit de: Sélectionner la catégorie, Cliquer sur l'objet, son aperçu apparaît dans la fenêtre de droite, Double-cliquer sur l'objet désiré pour le charger.
Celui-ci sera ajouté dans la liste "Device" du sélecteur d'objets. Une fois tous les objets chargés, fermer la fenêtre "PickDevices".



FigureIV.6 Espace permettant de charger les composants.

- **Placement d'un composant ou d'un objet** :

- Cliquer sur l'icône désirée de la barre d'outils de sélection de mode.

- Choisir, dans le sélecteur d'objet, le composant ou l'objet à placer, celui-ci apparaît dans la fenêtre vue d'ensemble et peut être orienté suivant les besoins grâce aux boutons de la barre outils d'orientation.

- Positionner le curseur dans la zone d'édition et cliquer sur le lieu de placement du composant ou de l'objet.

- **Connexion :**

Placement des connexions : Proteuse 8 supporte 2 types de connexion : manuelle ou automatique. Pour faciliter la connexion, vérifiez que l'option Auto-routeur de connexions du menu Outils est cochée.

Connexion manuelle	Connexion automatique
-Placer le curseur sur l'extrémité de la patte à connecter. -Le pointeur se transforme en un crayon. -Cliquer pour valider le point de départ. -Déplacer le curseur et valider chaque changement de direction par un clic jusqu'à atteindre le point d'arrivée. -Pour arrêter la pose d'une connexion, il suffit d'appuyer sur la touche Echap.	-Placer le curseur sur l'extrémité de la patte à connecter. Le pointeur se transforme en un crayon. -Cliquer pour valider le point de départ. -Placer directement le curseur sur le point d'arrivée, puis cliquer. -La liaison est faite automatiquement. Ce mode est plus rapide.

FigureIV.7 Placement des connexions.

a) **Suppression des connexions :**

- Cliquer droit sur la connexion à effacer ;
- Cliquer sur "Deletewire" ;

b) **Modification de tracé d'une connexion** :

- Sélectionner le fil par un clic droit ;

- Cliquer en gardant appuyé le bouton gauche de la souris sur le lieu à modifier, puis glisser vers la nouvelle position ;

- Répéter autant de fois que nécessaire l'étape précédente.

c) **Label de connexion :** Protuese 8 offre la possibilité de nommer les connexions ; Pour cela :

- Cliquer sur l'icône "Label de fil" de la barre d'outils de sélection de mode.

- Cliquer sur le fil, la fenêtre "Edit WireLabel" apparaît

- Saisir le nom de la connexion dans la zone "Chaîne"

- Choisir l'orientation du texte à l'aide des cases des zones "Rotation" et "Justifier" Lacase "Auto-Sync" permet d'affecter le même nom à tous les fils connectés au fil sélectionné.

- Cliquer sur OK.

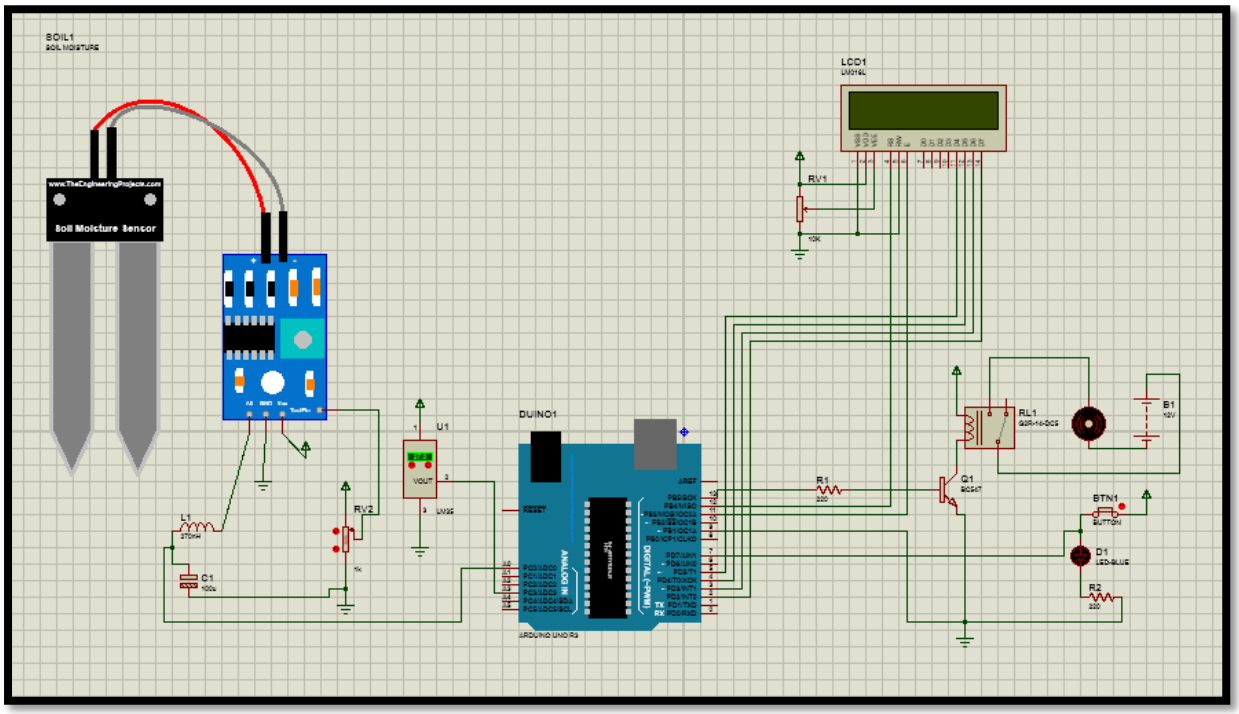
IV.2.4 Table des icones

MODE PRINCIPAL		MODE GADGETS		MODE GRAPHIQUE	
	Composants		Terminal		Ligne
	Point de jonction		Patte de composant		Rectangle
	Label de fil		Graphe		Cercle
	Script de texte		Cassette		Arc
	Bus		Générateurs		Chemin
	Sous circuit		Sonde de tension		Texte
	Édition		Sonde de courant		Symbole
			Appareils		Marqueur origine

FigureIV.8. Significations des différentes icones.

IV.3 Création et simulation des réseaux de capteurs

IV.3.1 Réseau de capteur pour la gestion de l'irrigation



FigureIV.9 Réseau de la gestion de l'irrigation

- **Programme Arduino utilisé pour la gestion de l'irrigation**

```
#include<LiquidCrystal.h> //LCD Library

#define NOTE_C4 262
#define NOTE_D4 294
#define NOTE_E4 330
#define NOTE_F4 349
#define NOTE_G4 392
#define NOTE_A4 440
#define NOTE_B4 494
#define NOTE_C5 523
```

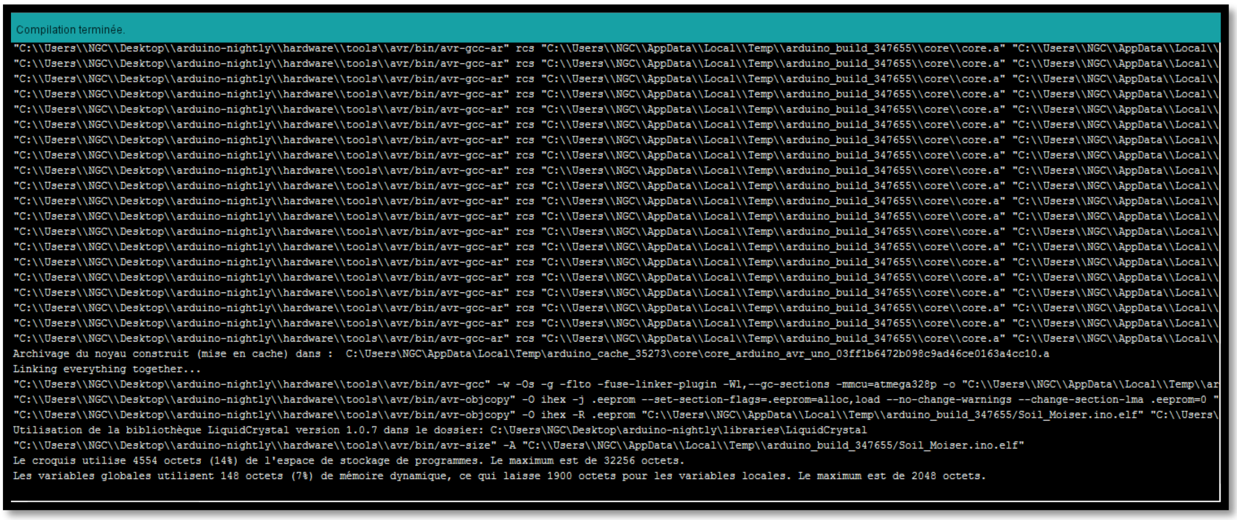
```
inttemp;
intT_Sensor = A3;
intM_Sensor = A0;
intW_led = 7;
intP_led = 13;
int Speaker = 9;
intval;
intcel;

LiquidCrystallcd(12, 11, 5, 4, 3, 2);

voidsetup()

 { //initialisation de la fonction setup
lcd.begin(16, 2);
lcd.clear();
pinMode(13,OUTPUT);
pinMode(7,INPUT);
pinMode(9,OUTPUT);

   val = analogRead(T_Sensor); //Lecture des valeure de capteur de tempirature
int mv = ( val/1024.0)*5000;
cel = mv/10;



lcd.setCursor(0,0);
```

```
lcd.print("Irrigarion");
lcd.setCursor(0,1);
lcd.print("Automatique ");
delay(1000);

}//fin de la fonctione setup

voidloop()
{ // initialisation de la fonction loop

lcd.clear();
intMoisture = analogRead(M_Sensor); //lecture des valeures du capteur

lcd.setCursor(0,0);
lcd.print("TEMP:");
lcd.setCursor(5,0);
lcd.print(cel);
lcd.setCursor(7,0);
lcd.print("*C");

if (Moisture> 700)   // pour la terre séche
  {
lcd.setCursor(11,0);
lcd.print("Terre");
lcd.setCursor(11,1);
lcd.print("seche");
      if (digitalRead(W_led)==1) //test de l'availabilité de l'au dans la pompe
      {
```

```
digitalWrite(13, HIGH);
lcd.setCursor(0,1);
lcd.print("PUMP:ON");
      }
else
      {
digitalWrite(13, LOW);
lcd.setCursor(0,1);
lcd.print("PUMP:OFF");

tone(Speaker, NOTE_C4, 500);
delay(500);
tone(Speaker, NOTE_D4, 500);
delay(500);
tone(Speaker, NOTE_E4, 500);
delay(500);
tone(Speaker, NOTE_F4, 500);
delay(500);
tone(Speaker, NOTE_G4, 500);
delay(500);
      }
   }

if (Moisture>= 300 &&Moisture<=700) //pour la terre moyenne
   {
lcd.setCursor(11,0);
lcd.print("terre");
lcd.setCursor(11,1);
lcd.print("Normal");
```

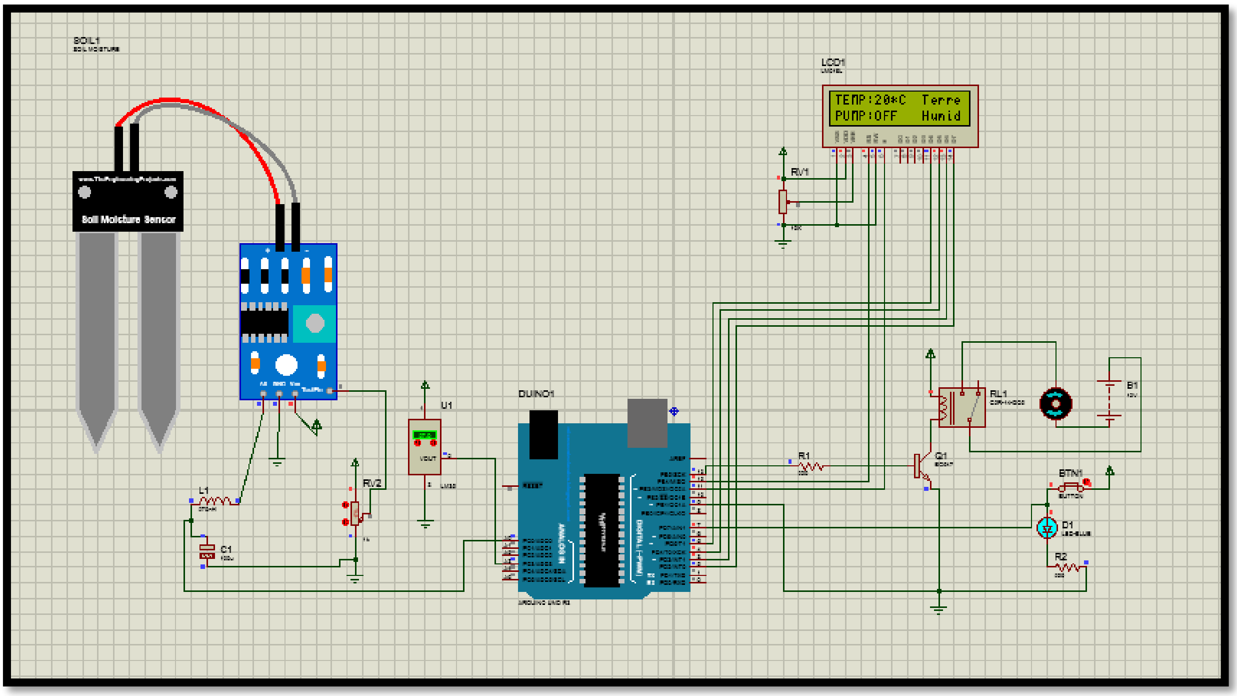
```
digitalWrite(13,LOW);
lcd.setCursor(0,1);
lcd.print("PUMP:OFF");
  }

  if (Moisture< 300)  // pour la terre huméde
  {
lcd.setCursor(11,0);
lcd.print("Terre");
lcd.setCursor(11,1);
lcd.print("Humide");
digitalWrite(13,LOW);
lcd.setCursor(0,1);
lcd.print("PUMP:OFF");
  }
delay(1000);

} //fin de la fonctioneloop
```

- **Validation de programme Arduino**



- **Simulation de fonctionnement de l'irrigation dans proteus 8**

1) Pour une terre humide

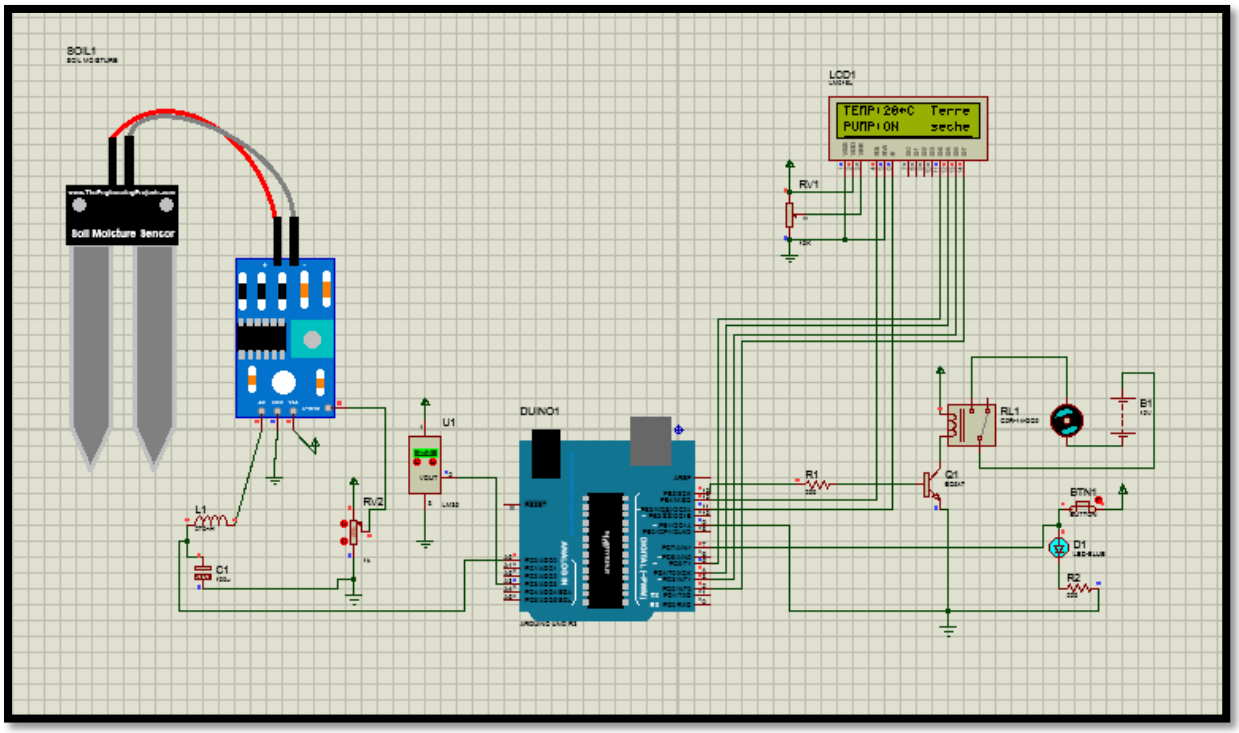


FigureIV.10 Simulation pour une terre humide.

❖ **Interprétation**

- L'humidité captée dans la terre est supérieurà 300 donc le microcontrôleurconsidère comme une terre humide.
- L'état de moteur de la pompe est off.
- La LED est allumé, vu dire la présence de l'eau dans la pompe.
- L'alarme du bouton d'arrêt est entente.

2) Pour une terre sèche

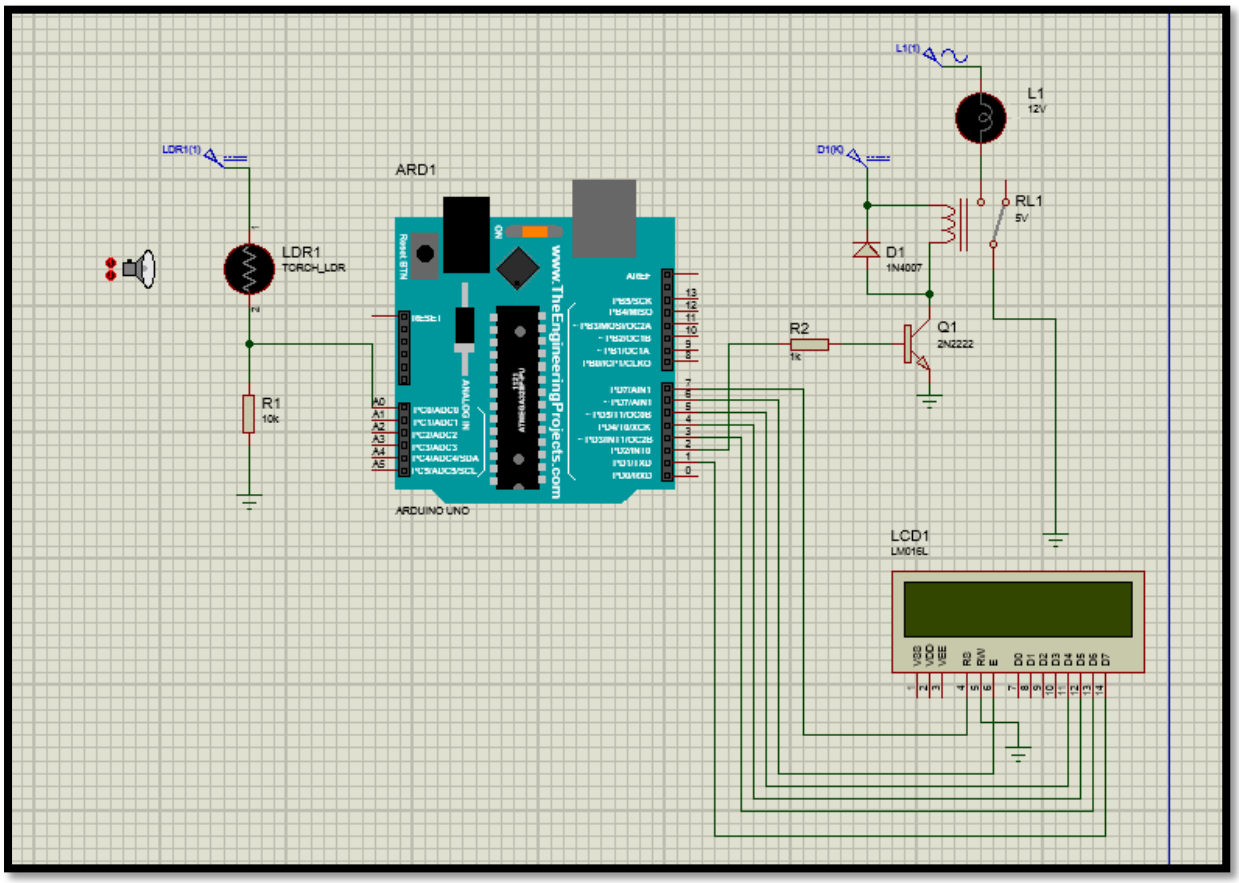


FigureIV.11 Simulation pour une terre sèche.

❖ **Interprétation**

- L'humidité captée dans la terre est inférieure à 300 donc le microcontrôleur considère comme une terre humide.
- L'état de moteur de la pompe est ON.
- La LED est allumé, vu dire la présence de l'eau dans la pompe.
- L'alarme du bouton d'arrêt est entente.

IV.3.2 Réseau de capteur pour la gestion de lumière



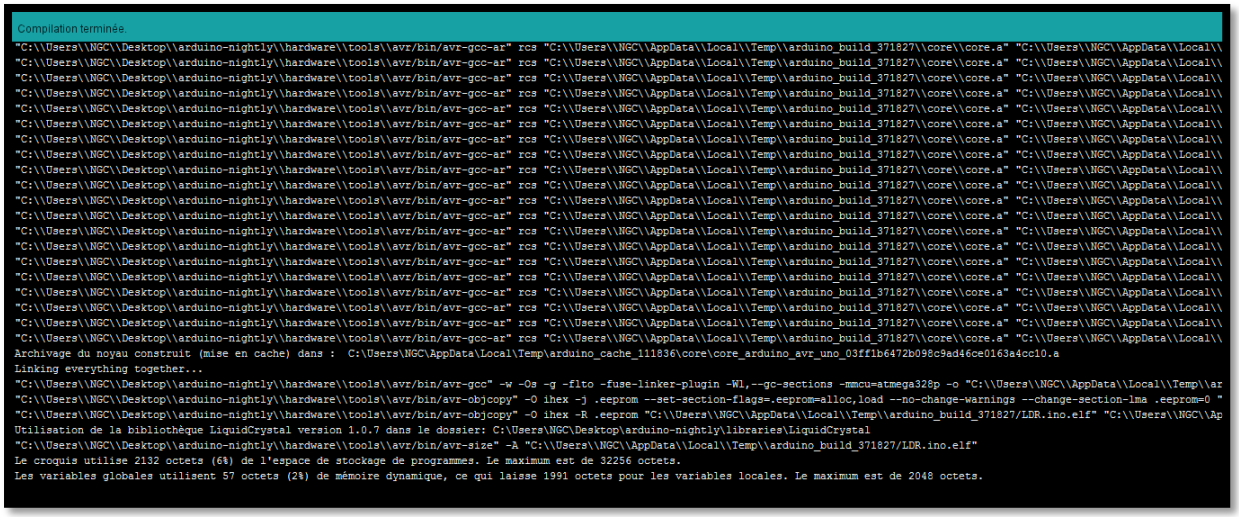
FigureIV.12 Réseau de capteur pour la gestion de lumière.

- **Programme Ardino utilise pour la gestion de lumière**

```
#include<LiquidCrystal.h>
LiquidCrystallcd(7, 6, 5, 4, 3, 1);//(rs,e,d4,d5,d6,d7)

intldrpin= A0;   // selection de l'input pin pour le capteur LDR
intldrvalue = 0;  // variable qui arrive vers le capteur

voidsetup() {
 // debut de la fonction setup

lcd.begin(16, 1);
```

```
// Print a message to the LCD.
lcd.print("Ldr out= ");

pinMode(2, OUTPUT); //pin connecté au relis
} // fin de la fonction setup

voidloop()
{ //deut de la fonction loop


  // lecture des valeures de capteur LDR
ldrvalue = analogRead(ldrpin);

lcd.setCursor(10,0);
lcd.print(ldrvalue);

if(ldrvalue> 300)
digitalWrite(2,LOW); //fermé le relis

else

digitalWrite(2,HIGH); //overire le relis

delay(100);

}// fin de la fonction loop
```

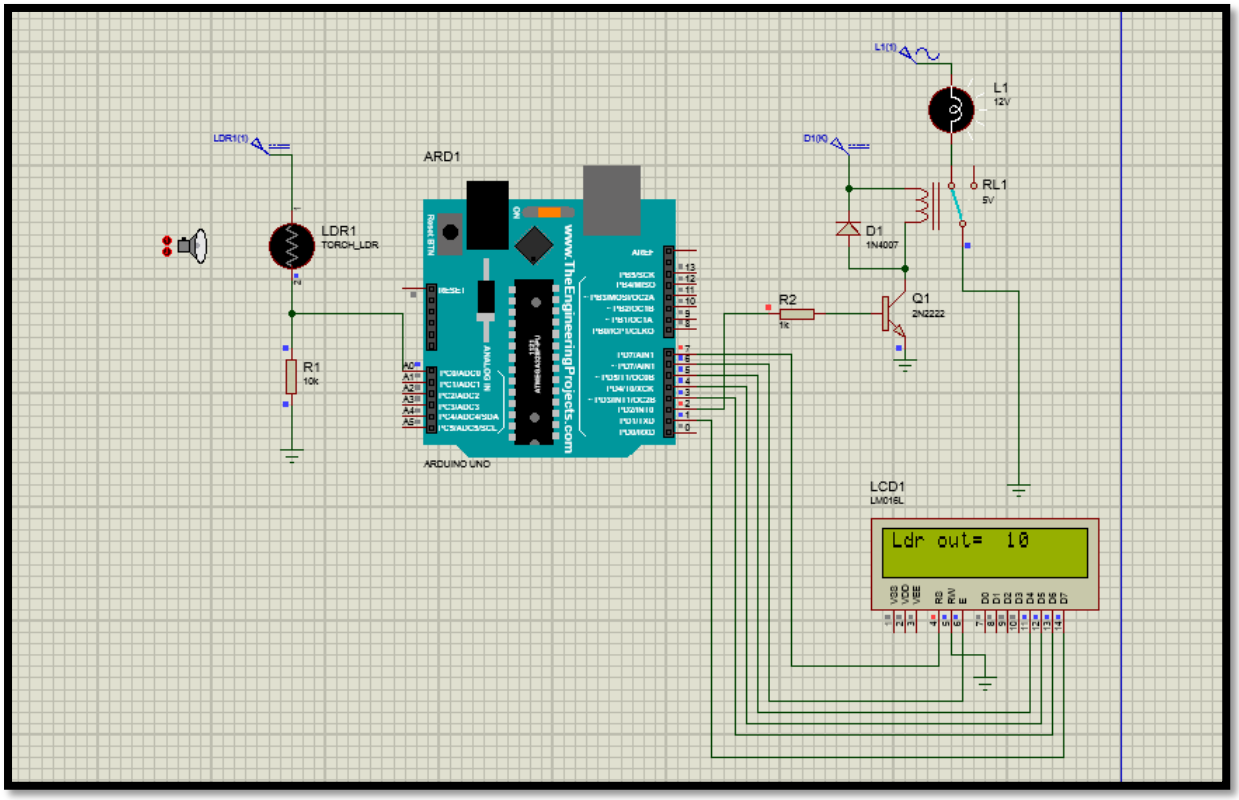
- **Validation de programme Arduino**



FigureIV.13. Validation du programme de la gestion de lumière.

- **Simulation de fonctionnement de la lumière dans Protuse 8**

1) **En absence de source de lumière**



FigureIV.14 Simulation en absence de source de lumière.

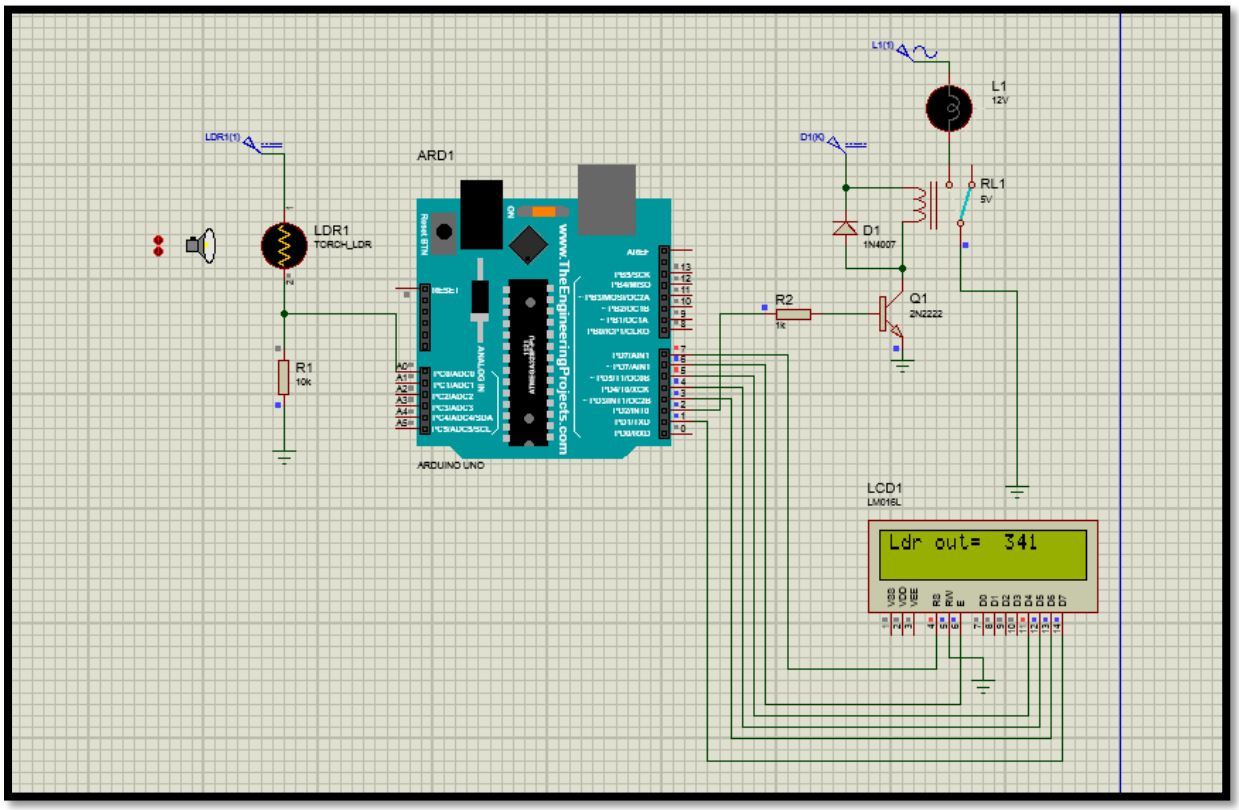
❖ **Interprétation**

- La distance de la source de lumière est inférieure a 300, le microcontrôleurconsidère que il n y a pas de lumière (<300 considère comme une lumièrenégligeable).

- L'afficheur LCD affiche la distance entre la source de lumière et le capteur LDR.

- La LED est allumé pour générer la lumière nécessaire à la plante. (éclairage artificielle ON)

2) En présence de source de lumière



FigureIV.15 Simulation en présence de source de lumière.

❖ **Interprétation**

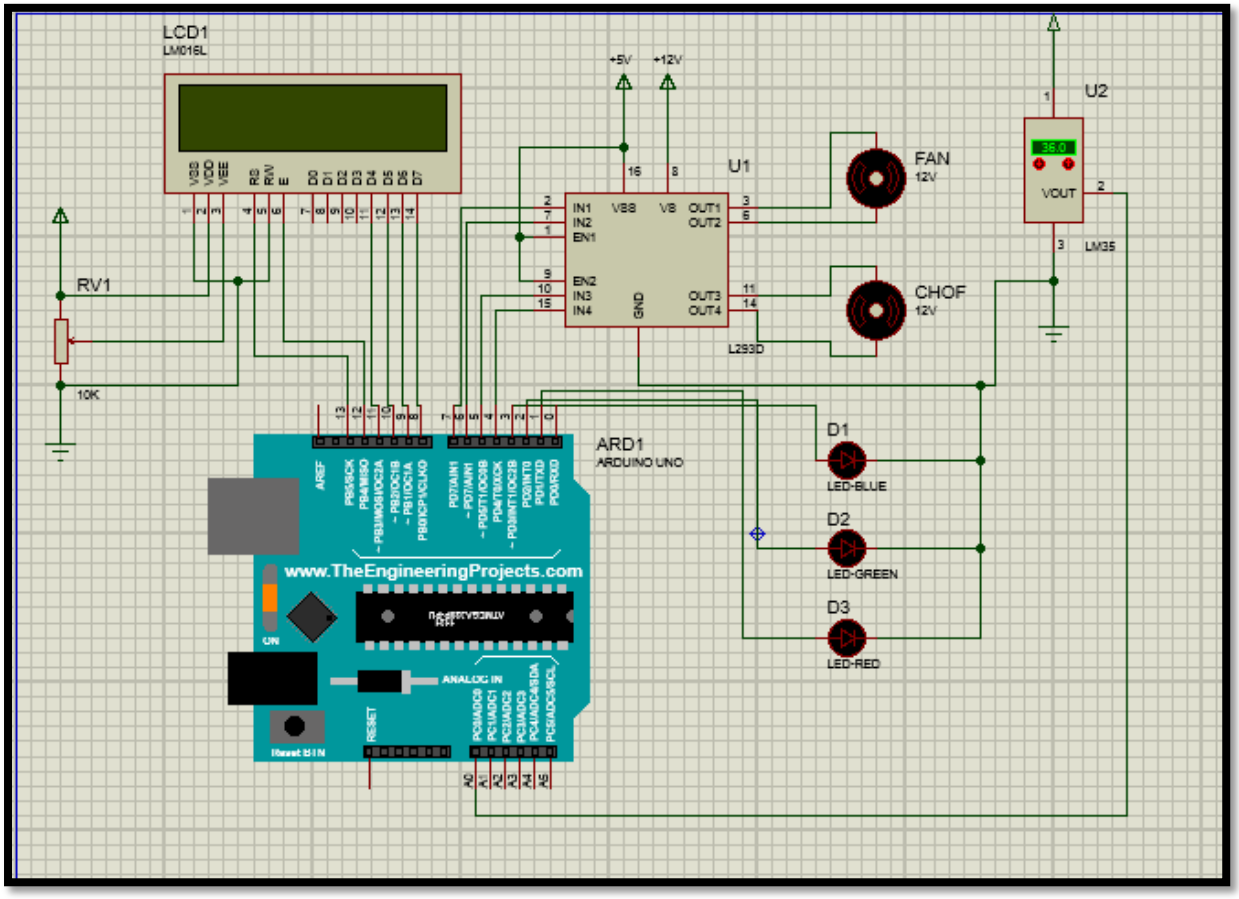
- La distance de la source de lumière est supérieure à 300, le microcontrôleur considère la présence d'une lumière.

- L'afficheur LCD affiche la distance entre la source de lumière et le capteur LDR.

- La LED d'éclairage.(éclairage artificielle OFF)

- L'éclairage naturel est le générateur de lumière de la plante.

IV.3.3 Réseau de capteur pour la gestion de la température



FigureIV.16 Réseau de capteur pour la gestion de la température.

- **Programme Ardino utilise pour la gestion de température :**

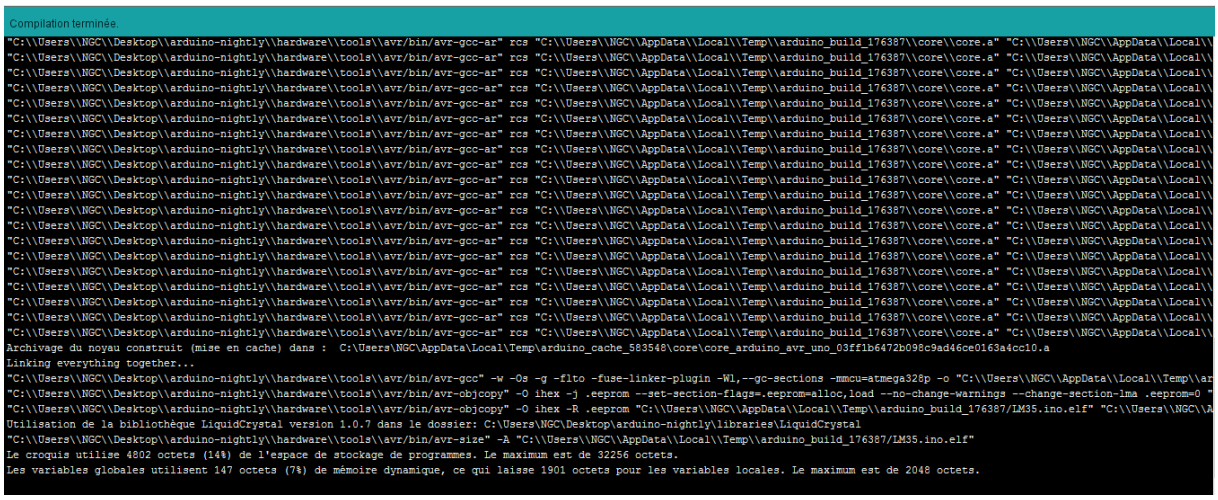
```
#include<LiquidCrystal.h>
constintrs = 13 , en = 12 , d4 = 11 , d5 = 10 , d6 = 9 , d7 = 8;
LiquidCrystallcd(rs, en, d4, d5, d6, d7);
inttempPin = 0; //declartion des variales
intval;
short celcius;
char temp[16];

voidsetup() { //debut de fonction setup

pinMode (7, OUTPUT);
pinMode (6, OUTPUT);
pinMode (5, OUTPUT);
pinMode (4, OUTPUT);
pinMode (3, OUTPUT);
pinMode (2, OUTPUT);
pinMode (1, OUTPUT);
lcd.begin(16, 2);
lcd.setCursor(0, 0);
lcd.print("Temp.Monitoring");
lcd.setCursor(1, 1);
lcd.print("Temp.Read:");
} //fin de fonction setup

voidloop() { //debut de fonction loop

Readtemp();

  //pour la tempirature normale

if (celcius<= 35 &&celcius>= 20) {
digitalWrite(2, HIGH);
```

```
digitalWrite(3, LOW);
digitalWrite(1, LOW);
digitalWrite(7, LOW);
digitalWrite(6, LOW);
digitalWrite(5, LOW);
digitalWrite(4, LOW);
  }

  //poure une haute température

else if (celcius> 35) {
FanON();
digitalWrite(1, HIGH);
digitalWrite(2, LOW);
digitalWrite(3, LOW);
digitalWrite(5, LOW);
digitalWrite(4, LOW);
delay(800);
lcd.setCursor(0, 1);
lcd.print("                ");
lcd.setCursor(1, 1);
lcd.print("The Fan is ON");
  }

  //pour une base température

else if (celcius< 20) {
HeaterON();
digitalWrite(3, HIGH);
digitalWrite(2, LOW);
digitalWrite(1, LOW);
digitalWrite(7, LOW);
digitalWrite(6, LOW);
delay(800);
```

```
lcd.setCursor(0, 1);
lcd.print("                ");
lcd.setCursor(0, 1);
lcd.print("The Heateris ON");
  }
}

voidReadtemp() {
delay(500);
lcd.setCursor(0, 1);
lcd.print("                ");
  val = analogRead(tempPin);
celcius = val*4.88/10;
sprintf(temp, "%0.2d", celcius);
lcd.setCursor(1, 1);
lcd.print("Temp.Read:");
lcd.print(temp);
lcd.write(B11011111);
lcd.print("C");
}

voidFanON() { //démarre de ventilateur

digitalWrite(7, HIGH);
digitalWrite(6, LOW);
delay(100);
}

voidHeaterON() { //démarrage de chauffage

digitalWrite(5, HIGH);
digitalWrite(4, LOW);
delay(100);
```

}//fin de fonction loop

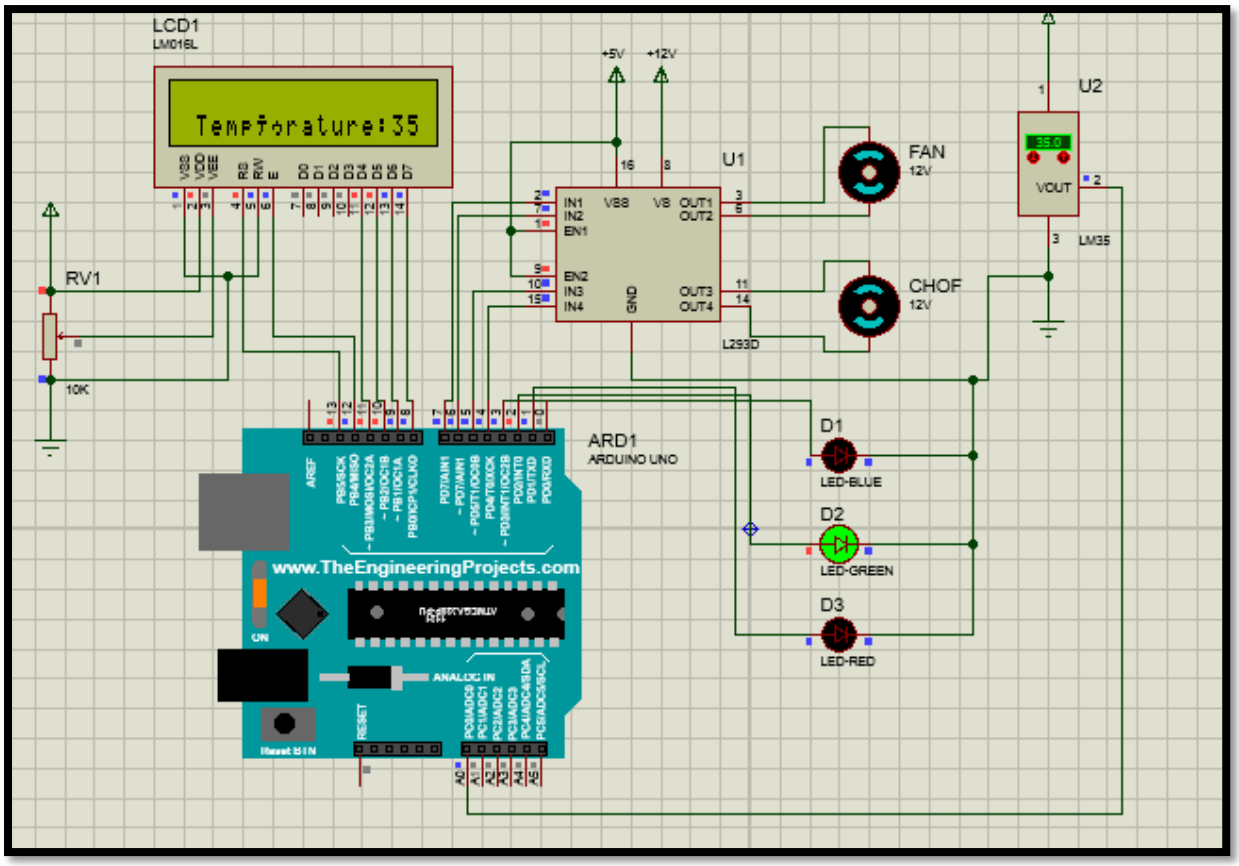
- **Validation de programme Arduino**



FigureIV.17 Validation de programme Arduino pour la gestion de température

- **Simulation de fonctionnement de la température dans Protuse 8**

1) Pour une température moyenne

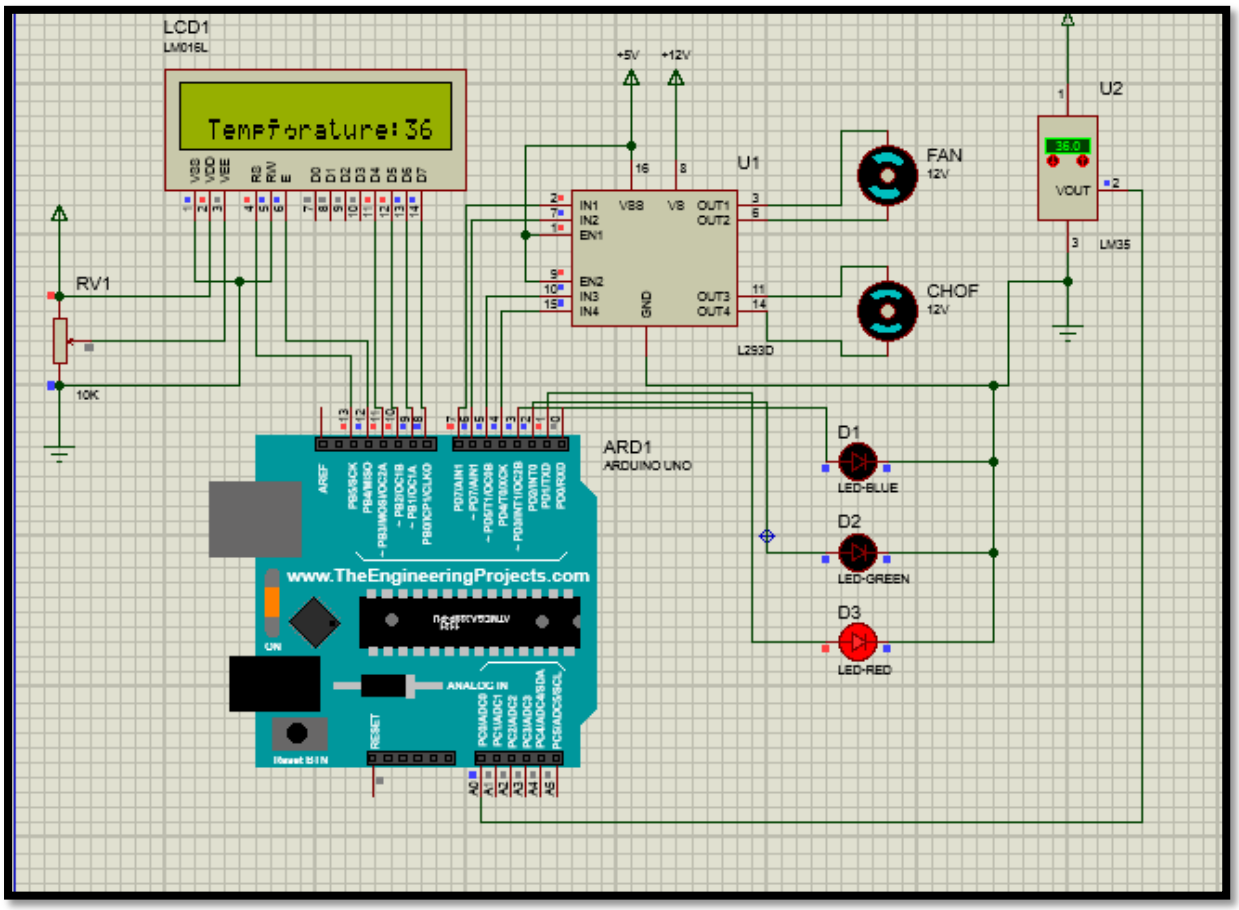


FigureIV.18 Simulation de fonctionnement pour une température moyenne.

❖ Interprétation

- Température capté par LM35 est 35°c.
- Le microcontrôleur considère la température capté comme une température moyenne.
- La LED vert est allumé.
- L'afficheur LCD affiche la température capté.
- L'état de ventilateur OFF.
- L'état de chauffage OFF.

2) Pour une température élevée

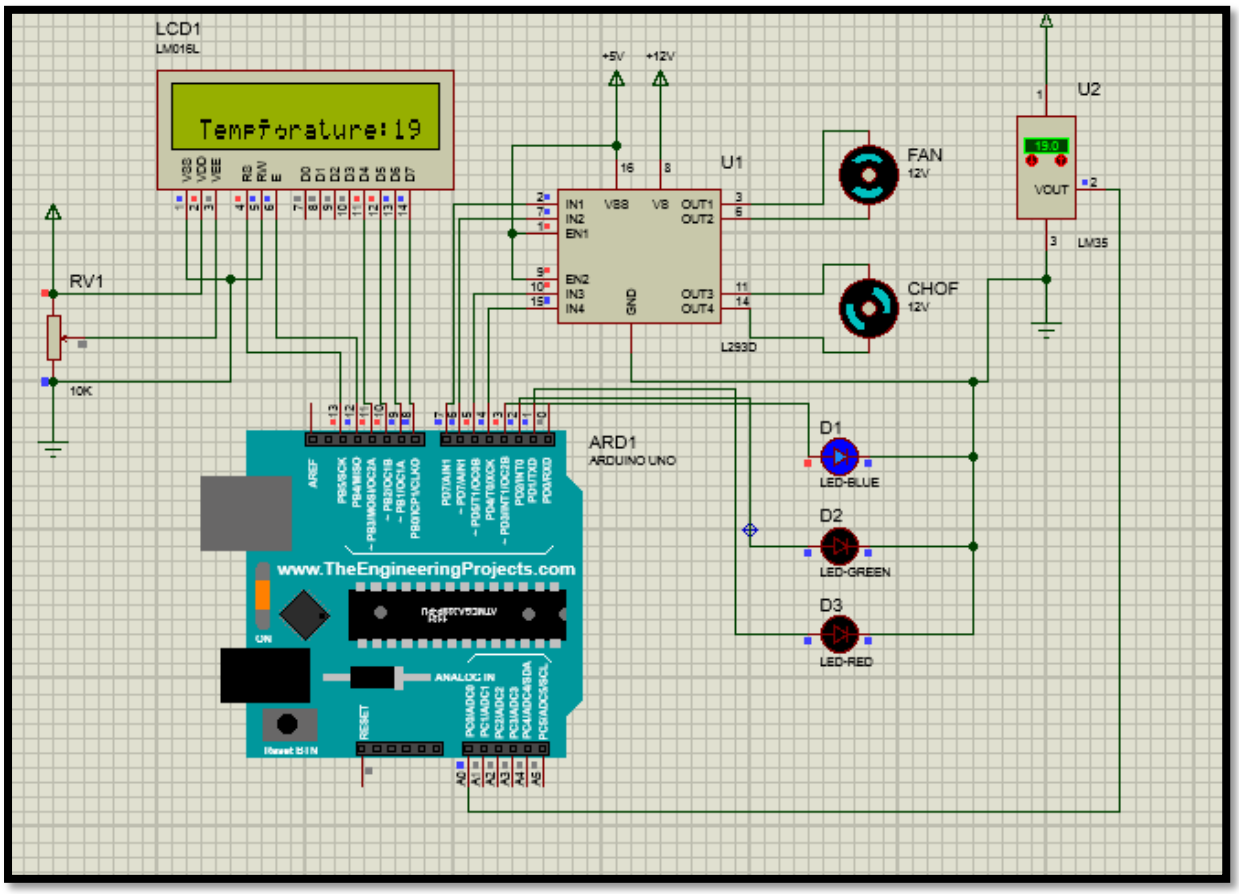


FigureIV.19 Simulation de fonctionnement pour une température élevée.

❖ Interprétation

- Température capté par LM35 est 36°c.
- Le microcontrôleur considère la température capté comme une température élevée.
- La LED rouge est allumé.
- L'afficheur LCD affiche la température capté.
- L'état de ventilateur ON.
- L'état de chauffage OFF.

3) Pour une base température



FigureIV.20 Simulation de fonctionnement pour une base température.

❖ Interprétation

- Température capté par LM35 est 19°c.
- Le microcontrôleur considère la température capté comme une base température.
- La LED bleu est allumé.
- L'afficheur LCD affiche la température capté.
- L'état de ventilateur OFF.
- L'état de chauffage ON.

IV.3.4 Réseau de fonctionnement général de notre serre

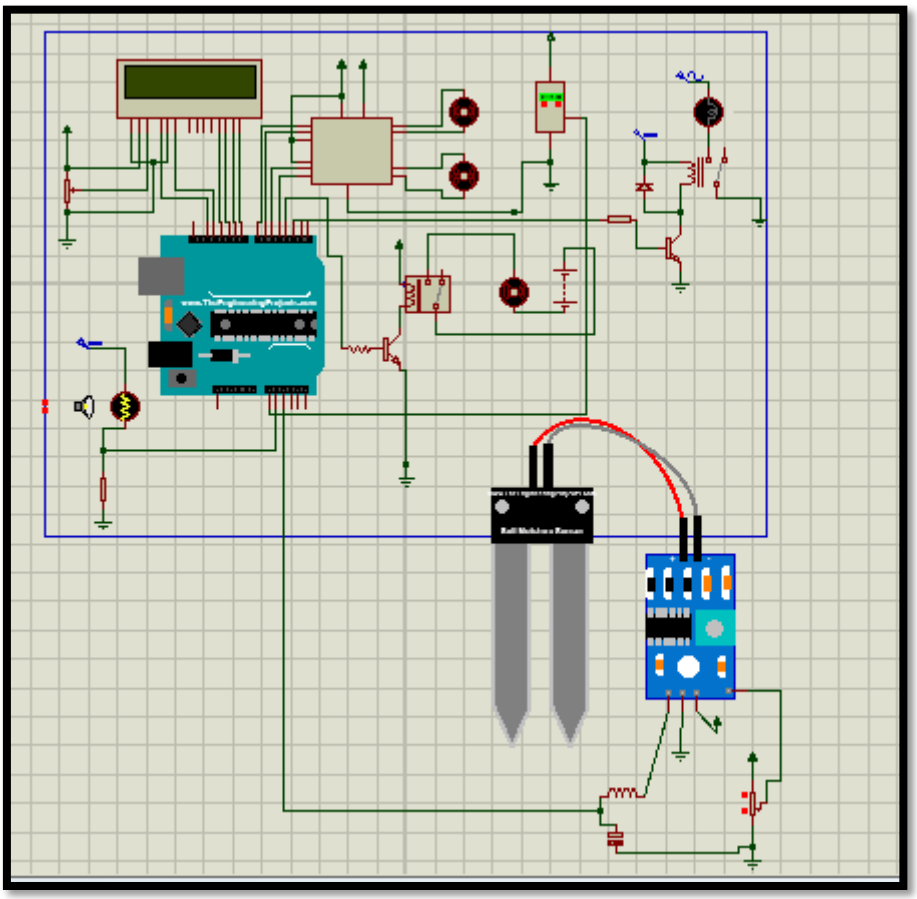


Figure IV.21 Réseau de fonctionnement général de notre serre

- **Programme Ardino utilise pour le fonctionnement général**

  C'est le résultat des trois programmes (irrigation ; température ; Lumière) compilé tous ensemble avec quelque modification et déclaration des variables globaux pour éviter les erreurs de la fonction setup et la fonction loop principales dans chaque programme utilisé.

```
#include <LiquidCrystal.h>
int ldrpin= A1;   // selection de l'input pin pour le capteur LDR
int ldrvalue = 0;  // variable to store the value coming from the sensor
const int rs = 13 , en = 12 , d4 = 11 , d5 = 10 , d6 = 9 , d7 = 8;
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);
int tempPin = 0; //declartion des variales
int val;
short celcius;
char temp[16];
int M_Sensor = A2;

void setup() {
 // debut de la fonction setup
  pinMode(3, OUTPUT); //connect le capteur d'humldité de sol
```

```
  delay(1000);
  pinMode(2, OUTPUT); //pin connecté au relis
  pinMode (7, OUTPUT);
  pinMode (6, OUTPUT);
  pinMode (5, OUTPUT);
  pinMode (4, OUTPUT);
  lcd.begin(16, 2);
  lcd.setCursor(0, 0);
  lcd.print("");
  lcd.setCursor(1, 1);
  lcd.print("Température:");

}
void loop()
{ //deut de la fonction loop
  // lecture des valeures de capteur LDR
  ldrvalue = analogRead(ldrpin);
    if(ldrvalue > 300)
  digitalWrite(2,LOW); //fermé le relis
    else
    digitalWrite(2,HIGH); //overire le relis
    delay(100);
    Readtemp();
  //NORMAL TEMP
  if (celcius <= 35 && celcius >= 20) {
    digitalWrite(7, LOW);
    digitalWrite(6, LOW);
    digitalWrite(5, LOW);
    digitalWrite(4, LOW);
  }
  //HIGH TEMP
  else if (celcius > 35) {
    FanON();
    digitalWrite(5, LOW);
    digitalWrite(4, LOW);
    delay(800);
    lcd.setCursor(0, 1);
    lcd.print("                ");
    lcd.setCursor(1, 1);
    lcd.print("ventelau ON");

  }
  //LOW TEMP
  else if (celcius < 20) {
    HeaterON();
     digitalWrite(7, LOW);
    digitalWrite(6, LOW);
    delay(800);
    lcd.setCursor(0, 1);
    lcd.print("                ");
```

```
    lcd.setCursor(0, 1);
    lcd.print("chauffage ON");
      }
      int Moisture = analogRead(M_Sensor); //lecture des valeures du capteur
  if (Moisture > 700)  // pour la terre séche
  {
      digitalWrite(3, HIGH);
    }
    else{
        digitalWrite(3, LOW);
      }

  if (Moisture >= 300 && Moisture <= 700) //pour la terre moyenne
  {
      digitalWrite(3, LOW);
    }
  if (Moisture < 300)  // pour la terre huméde
  {
     digitalWrite(3, LOW);
   }
  delay(1000);
}
void Readtemp() {
  delay(500);
  lcd.setCursor(0, 1);
  lcd.print("                ");
  val = analogRead(tempPin);
  celcius = val*4.88/10;
  sprintf(temp, "%0.2d", celcius);
    lcd.setCursor(1, 1);
    lcd.print("Température:");
    lcd.print(temp);
    lcd.write(B11011111);
    lcd.print("C");
}
void FanON() {
  digitalWrite(7, HIGH);
  digitalWrite(6, LOW);
  delay(100);
}
void HeaterON() {
  digitalWrite(5, HIGH);
  digitalWrite(4, LOW);
  delay(100);
}
```

- **Validation de programme Arduino**

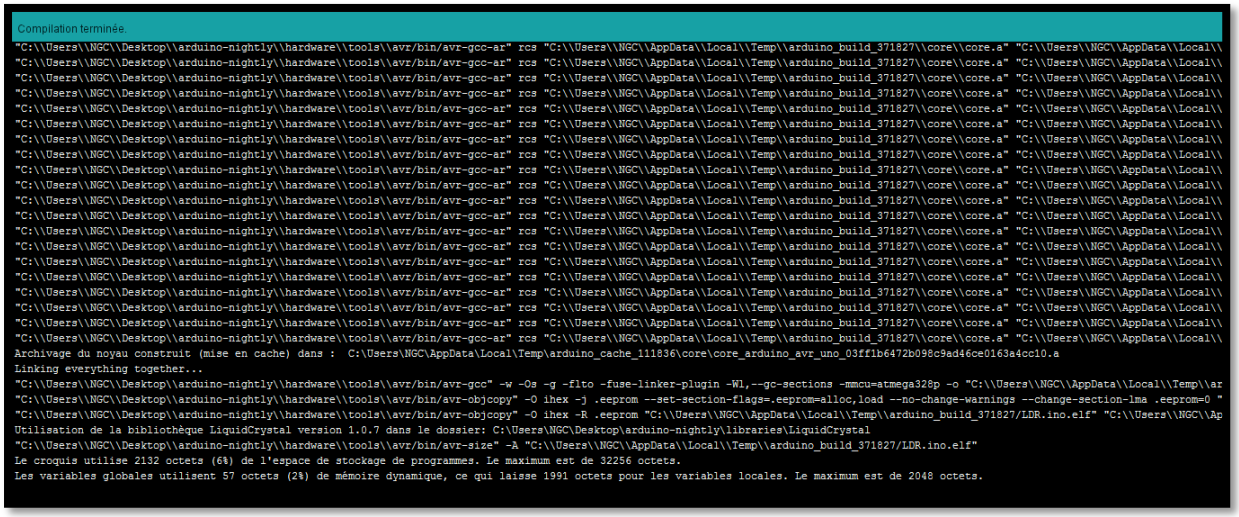


Figure IV.22 Validation du programme de fonctionnement général.

IV.4 Conclusion

Nous avons étudié dans ce chapitre les étapes à suivre pour simuler une serre agricole, en se basant sur trois réseaux de capteur qui sont liés à un microcontrôleur qui est la carte Arduino. Les circuits réalisés dans ce projet nous permettent de contrôler et gérer le climat de la serre automatiquement sans l'interruption de l'être humain.

Conclusion Générale

L’objet de ce projet était de réaliser un système de régulation des paramètres d’une serre, à travers les résultats que nous avons obtenus après simulation et réalisation du circuit électronique, nous pouvons dire que l’objectif qui nous a été fixé au départ est atteint. néanmoins ce travail peut être modifié (en utilisant par exemple le capteur DHT11 ou DHT22 qui est un capteur de température et d’humidité en même temps, ou par intégration du WIFI pour simulation à distance et également la gestion de plusieurs serres en même temps en intégrant des boutons par exemple…) comme il peut être amélioré afin d’atteindre d’autres objectifs et de satisfaire d’autres exigences et répondre ainsi aux besoins des agriculteurs dont la concurrence devient de plus en plus agressive, et à laquelle ces derniers doivent faire face en assurant une bonne gestion du climat dans la serre.

Référence

[01] - MEDJBER Ahmed, « Automatisation d'une serre agricole : commande et régulation », Editions Universitaires Européennes, Allemagne, 2012.

[02] - B. BOUCHIKHI, A. ED-DAHHAK, A. LACHHAB, et L.EZZINE. « Control of the climate and the drip Fert irrigation undergreenhouse », Automatic 1er Salon International de l'agriculture au Maroc Meknès. Les 20-27 Avril (2006).

[03] - Y. Bouteraa, "Automatisation d'une serre agricole, " Magister en Sciences Agronomiques,
Ecole Nationale Supérieure D'agronomie-El Harrach, 2012.

[04] - AstalasevenEskimon et olyte, " Arduino pour bien commencer en électronique et en programmation ", Licence Creative Commons BY-NC-SA 2.0, Dernière mise à jour le 4/08/2012.

[05] -Becky Stewart, "À l'aventure avec ARDUINO, Découvre Arduino et l'électronique grâce à 9 aventures trépidantes !".

[06] - "Carte Arduino", [en ligne].Disponible : http:// www.editions-eyrolles.com/go/arduino.

Bibliographe

[W1] - https://www.lesoleil.com/maison/horticulture/lhistoire-des-serres-des-romains-a-aujourdhui-81ffb6ae10e8f8ffbd6a5325b807e6d9

[W2] - http://www.95collegiens.sitew.com/Le_Projet.B.htm#Le_Projet.B

[W3] - https://blog.jardincouvert.com/serres-aluminium/differents-types-serres/

[W4] - https://www.ma-serre-de-jardin.com/content/33-une-serre-a-quoi-ca-sert-quels-sont-les-differents-types-de-serres

[W5] - https://reporterre.net/Qu-est-ce-qu-une-serre-aquaponique

[W6] - http://www.serre-jardin.com/img/cata_euro_serre_2016.pdf.

[W7] -http://www.ulmaagricola.com/fr/serres/equipements.