



Republique Tunisienne
Ministère de l'Enseignement Supérieur et la recherche scientifique
Institut Supérieur des Études Technologiques – Rades

Département Genie électrique
Projet de Fin d'Année (mini projet)

Serre agricole autonome



Specialite : automatisme et informatique industrielle – AII

Présenté par:

- **Brary Nour Elhouda**
- **Ghozzi Med Aziz**

Encadré par :

- **Mr Abidi Hatem**
- **Mr Ben Abdallah Mohamed**

Année universitaire : 2023/2024

Table des matières

Introduction.....	5
Chapite 1 :Description du besoin.....	6
1-Étude du besoin	8
2-Les serres agricoles	10
2-1 Importance de la serre :.....	10
2-2 Types des serres:.....	12
Chapitre 2 :Internet des objets	13
1-Introduction.....	13
2- Généralité sur internet des objets.....	14
2-1 Définition d'internet des objets :.....	14
2-2 Intérêt de l'internet des objets :.....	15
2-3 Inconvénients de l' internet des objets:	15
2-4 Les composantes d'un système d'internet des objets.....	16
2-5 Fonctionnement de l'Internet des Objets :.....	16
3- Protocoles de communication	17
3-1 Bluetooth	17
3-2 Zigbee	18
3-3 Wi-Fi.....	19
4- Les applications internet of things.....	20
5- Conclusion	22
Chapitre 3 : Conception et Réalisation	23
1- Partie Materielle:	23
1-1 Choix du microcontrôleur.....	23
1-2 Brochage de ESP32 :	24
1-3 Affichage.....	32
2- Partie logicielle :.....	34
2-1 Plateforme : ARDUINO IDE :	34
2-2 Plateforme : Proteus ISIS :	41
2-3 Plateforme : Adafruit io	43
3- Partie construction :	51

3-1 Integration capteurs et actionneurs :	51
3-2 Construction Carcasse :.....	52
3-3 Construction Final :	53
Conclusion	54

Table des Figures

Figure 1: representation de l'analyse fonctionnel du besoin.....	8
Figure 2: vue globale d'une serre réelle	10
Figure 3: serre arc gothique.....	12
Figure 4: internet of things.....	13
Figure 5: représentation simplifiée de la nouvelle dimension de l'IoT	14
Figure 6:Schéma explicatif de fonctionnement de l'IoT	17
Figure 7:application de Bluetooth	18
Figure 8:application de zigbee	19
Figure 9:applications de reseau Wi-Fi.....	19
Figure 10:agriculture moderne.....	20
Figure 11: IOT dans l'agriculture	20
Figure 12:Les différentes applications d'IOT	21
Figure 13:Architecture ESP32.....	24
Figure 14:Schéma du broches ESP32 Wroom Devkit.....	25
Figure 15:Description de DHT11	
Figure 16 :DHT11	26
Figure 17:capteur d'humidité du Sol	
Figure 18: Description de capteur d'humidité du Sol	28
Figure 19:un capteur de Gaz MQ2.....	29
Figure 20:Description de capteur de niveau d'eau	
Figure 21:un capteur de niveau d'eau	30
Figure 22:Un Relais	31
Figure 23: une ventilateur	32
Figure 24:Afficheur LCD avec interface I2C	33
Figure 25: Logo Arduino.....	34
Figure 26: Interface Arduino IDE	35
Figure 27: installation bibliotheque arduino IDE.....	35
Figure 28: partie declaration	36
Figure 29: Partie void setup()	37
Figure 30: programme principale de la serre.....	39
Figure 31: ajout de COM3	40

Figure 32: affichage du port COM3.....	41
Figure 33: Interface ISIS.....	42
Figure 34: schema ISIS de la serre.....	43
Figure 35: étapes necessaire adafruit io.....	45
Figure 36: Dashboard de la temperature.....	45
Figure 37: Dashboard Humidity.....	46
Figure 38:Dashboard sol sensor	46
Figure 39: Dashboard indicators	47
Figure 40: bibliotheques necessaires pour Adafruit io	47
Figure 41: code Adafruit io	50
Figure 42: config.h de l'Adafruit io	51
Figure 43: integration matérielles	51
Figure 44: carcasse	
Figure 45: reservoir d'eau	52
Figure 46: Etat final de notre serre	53

Table des tables

Tableau 1:Comparaison entre DHT 11 et DHT22.....	27
Tableau 2:Comparaison entre Capteur SKU : SEN0193 & yl-38.....	28

Introduction

L'agriculture d'aujourd'hui doit faire face à plusieurs enjeux : un monde de plus en plus chaud dû, au dérèglements climatiques, l'explosion démographique, des cultures de plus en plus intensives, l'épuisement des ressources fossiles, les enjeux environnementaux (protection de la biodiversité, lutte contre les pollutions,), et sanitaires, qui imposent un besoin immédiat d'adaptation , et un changement de modèle pour tendre vers plus de durabilité. La culture sous serre, permet de lutter contre, ces différents obstacles, car, les serres permettent : une diversité des cultures, la prolongation des saisons, la protection des plantes contre les maladies, et un meilleur rendement.

Cependant, elles doivent faire face, à un marché de plus en plus concurrentiel et conditionné par des normes de qualité sévères. Les systèmes de production des serre agricoles deviennent considérablement sophistiqués et donc incroyablement coûteux. C'est pourquoi, il est primordial d'optimiser les investissements des serristes, par une plus grande maîtrise des conditions de production.

Dans ce cadre, ce projet, a pour finalité, la réalisation d'un contrôle à distance, d'une serre agricole.

Le mémoire se divise en trois parties. Dans un premier temps, nous verrons les concepts généraux, de l'IOT (Internet of Things), puis nous verrons un peu plus en détail, l'agriculture intelligente et les différents paramètres, à prendre en compte, dans une serre agricole, et enfin, dans un dernier temps, nous passerons à la conception et réalisation de ce système de contrôle.

Chapite 1 :Description du besoin

Pourquoi une serre autonome?

La culture des végétaux se fait généralement en plein air dans des jardins potagers ou des jardins d'agréments sur des balcons. C'est une activité professionnelle pour certains ou de loisirs et de plaisirs pour d'autres.

De nombreuses contraintes existent avant de permettre à une plante de se développer.

1. La lumière: est un élément permettant la photosynthèse, c'est à dire la capacité par laquelle les plantes vertes synthétisent des matières organiques grâce à l'énergie lumineuse, en absorbant le gaz carbonique de l'air et en rejetant l'oxygène. Plus simplement, une plante a un besoin constant de lumière, aussi il est nécessaire et de lui apporter la quantité de lumière suffisante quelque soit l'éclairage extérieur (naturel ou artificiel).

2. L'eau: Comme tout être vivant, une plante a besoin d'eau. Celle-ci déployant ses racines dans la terre, un certain taux d'humidité doit être respecté. Un niveau d'humidité dans l'air est également un facteur favorisant la croissance du végétal.

3. La température: est également un élément important pour le développement d'une plante. Une température trop élevée conduit généralement à la destruction du végétal.

4.Nutrition organique: L'apport de nutriment organique peut être fait grâce à des bâtonnets que nous insérons dans la terre. De nombreux produits existent déjà et permettent une autonomie de plusieurs semaines pouvant aller à plusieurs mois.

Le fait de pouvoir déplacer la plante et de l'installer successivement à plusieurs endroits doit être pris en compte et devra pouvoir se faire sans difficulté notamment en terme de manutention. La plante restera constamment visible par l'utilisateur.

Pour certains passionnés et autres horticulteurs le développement de la plante est un souci constant. Or, lors d'absences prolongées, il deviant impossible pour ces personnes de suivre la bonne croissance des végétaux. Des solutions permettant de surveiller les paramètres vitaux précédents (lumière, humidité de la terre et de l'air, température) doivent donc pouvoir être mis en oeuvre afin de rassurer le jardinier.

De la même façon, mais cette fois lors de la présence de l'utilisateur les mêmes paramètres physiques doivent être visibles facilement par celui-ci. A partir de ces besoins, plusieurs questions apparaissent.

Comment avoir un système permettant d'accueillir et de déplacer un ou plusieurs végétaux de taille moyenne ?

Comment avoir un système permettant de gérer de façon autonome tous les paramètres physiques nécessaires au développement d'une plante?

Comment informer directement l'utilisateur de l'état des paramètres physiques ?

Comment informer à distance l'utilisateur de l'état de ces mêmes paramètres?



1-Étude du besoin

Avant de se lancer dans la conception, il convient de bien identifier et formaliser les objectifs du projet. Cette phase est essentielle, car elle fixe la direction du travail qui va être entrepris. Pour faciliter cette tâche, il existe un outil pratique afin d'explicitier les besoins plus aisément : la bête à cornes.

Ce diagramme simple d'utilisation sert de guide pour mener une analyse fonctionnelle du besoin.

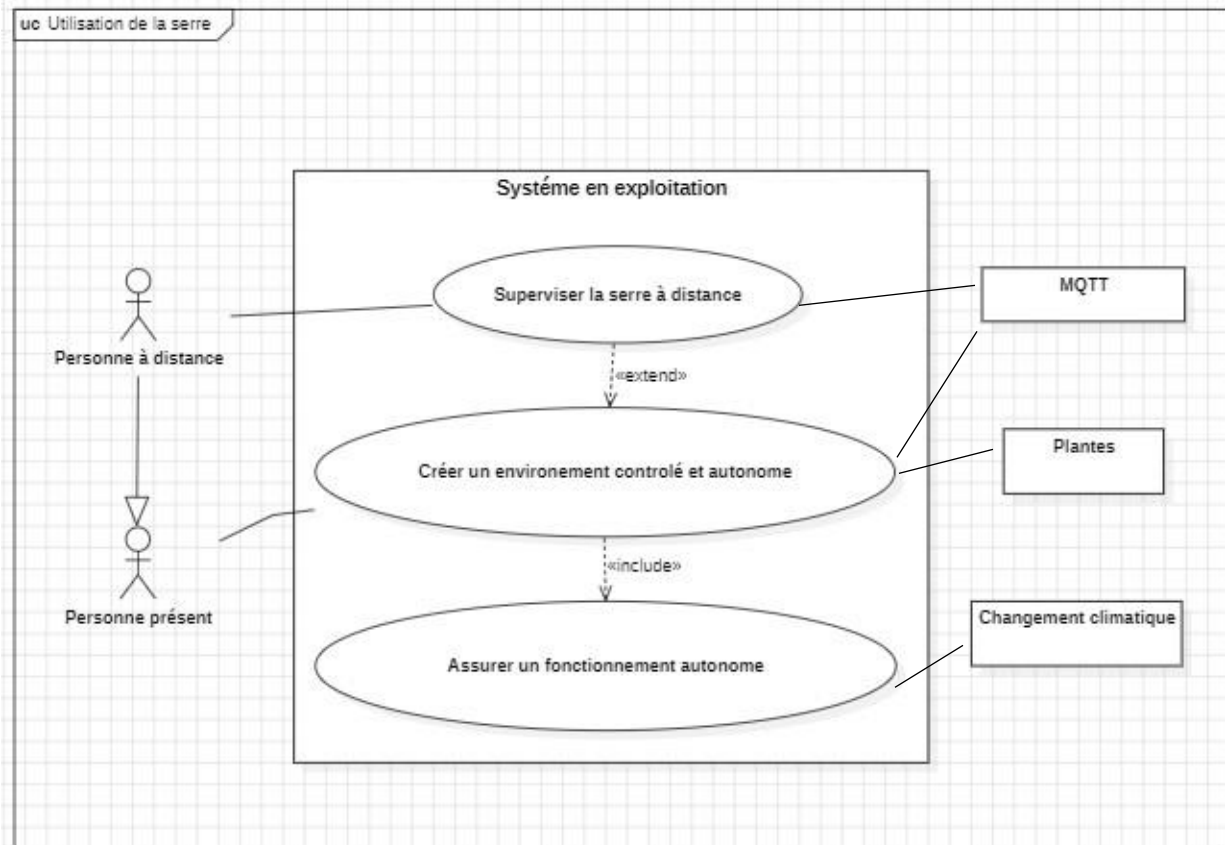


Figure 1: représentation de l'analyse fonctionnel du besoin

Reprenons les 3 questions précédentes et apportons des réponses plus développées. **A qui l'objet technique rend-il service?**

- Une personne désirant avoir des plantes dans son habitat, mais étant débutant dans le domaine de la agriculture et la culture des végétaux ou n'ayant que peu de temps à y consacrer.
- Une personne qui s'absente régulièrement et souhaitant pouvoir suivre à distance le développement des ses plantes.

- Un passionné de la culture des plantes souhaitant pouvoir suivre de façon précise le développement de ses végétaux.
- Un enseignant désirant montrer l'importance des paramètres et les conditions physiques nécessaires au bon développement d'une plante.

Sur qui ou sur quoi agit-il ?

-L'objet technique va agir sur :

- les paramètres physiques (luminosité, humidité de la terre et de l'air, la température) de la plante de plus que la serre.
- la plante en contrôlant tous les paramètres physiques nécessaires à sa croissance.
- l'utilisateur en l'informant des niveaux des paramètres

Il s'agit à présent de faire un contrôle de validité afin de s'assurer de la pertinence de l'analyse. Il convient de poser des questions complémentaires pour s'assurer que le besoin est valide et pérenne.

Qu'est-ce qui pourrait faire disparaître le besoin de l'objet technique ?

- Le fait que personne n'ait plus envie de cultiver des végétaux.
- Une plante ne nécessitant aucun entretien et par conséquent plus besoin d'eau, plus besoin de respecter une certaine température.
- Un coût trop élevé de cet objet technique.

Quel est le risque de disparition ?

- «Le fait que personne n'ait plus envie de cultiver des végétaux» Ce risque semble très limité dans la mesure où actuellement de plus en plus de personnes se tournent vers davantage de «nature» ou d'une ambiance plus naturelle.
- «Une plante ne nécessitant aucun entretien et par conséquent plus besoin d'eau, plus besoin de respecter une certaine température».

Ce risque semble très limité. Effectivement, il existe des végétaux ne nécessitant que très peu d'eau ou pouvant vivre avec peu de lumière mais dans un souci de diversité, les amateurs souhaitent avoir des végétaux de différentes sortes et tous ne peuvent pas se développer avec des contraintes extérieures extrêmes.

- «Un coût trop élevé de cet objet technique»

L'aspect financier peut effectivement être un frein à l'achat de cet objet technique. Toutefois, il existe déjà dans le commerce des produits de ce type et le coût maximal relevé est proche de 500€. Un prix de revient de cet objet technique avoisinant la moitié de ce prix maximum semble être un bon compromis. De plus, il n'est pas rare de voir des prix de végétaux pouvant atteindre

plusieurs dizaines d'euros. Une solution technique permettant de s'affranchir de nombreuses contraintes et avoisinant le prix de 400€ semble être un bon compromis.

Des enquêtes plus précises permettraient d'aboutir à une conclusion très proche de nos suppositions.

2-Les serres agricoles

Une serre est une structure close ou semi-ouverte translucide, en verre ou en plastique, soutenue par une structure métallique ou en bois, destinée en general à la production agricole. Elle vise à protéger les cultures vivrières ou de loisir des éléments climatiques, afin d'améliorer la production des plantes, d'en accélérer la croissance et de les produire indépendamment des saisons grâce à un gain de température par effet de serre sous la structure.



Figure 2: vue globale d'une serre réelle

2-1 Importance de la serre :

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

Avantages d'une production sous serre :

La production sous serre permet de pallier les problèmes rencontrés de la culture en plein air :

- Les besoins des cultures sont limités dans le climat local.
- Les conditions climatiques externes entravent la production sous serre.

- La serre permet de gérer aisément les facteurs climatiques qui ne sont pas le cas pour une culture en plein air.

Parmi les avantages majeurs d'une production sous serre, nous citons :

- 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 de la production.
- Réduction de la consommation de fongicides et insecticides. La serre agricole contribue largement à la modernisation du secteur agricole par l'implémentation de nouvelles technologies.

Paramètres d'une culture sous serre :

- ❖ La culture sous serre, requiert, le respect de différents critères et conditions:

1)La gestion du climat:

Le climat, est un critère indispensable, pour la création d'un environnement idéal, pour la croissance des différents plants.

2) La gestion de la lumière:

La lumière peut être artificielle. Elle sert notamment à favoriser l'induction florale(phénomène botanique qui fait qu'un bourgeon à feuille évolue en bouton à fleur.) de certaines espèces de plantes de jour long en rallongeant la durée du jour. L'horticulteur, peut choisir, la durée, de l'éclairage, qu'il juge nécessaire au bien être des plantes .

3)La gestion de la température:

La gestion de la température des serres est contrôlée par la ventilation en cas d'excès. Si les températures baissent à un niveau inférieur à celui accepté par les cultures, on utilise de puissantes chaudières au gaz naturel ou à fioul ou autre moyen de chauffage pour élever la température.

4) La gestion de l'air:

Air et température, sont deux facteurs qui sont liés: plus l'air circule, plus la température de la serre va être proche de celle de l'extérieur. Les plantes convertissent le dioxyde de carbone en dioxygène durant la période diurne de la journée grâce à la photosynthèse. Pour augmenter la productivité de certaines cultures, il arrive que l'on fasse brûler du gaz propane dans les serres pour y augmenter le taux de dioxyde de carbone (CO2).

5) La gestion de l'eau:

L'eau dans la serre est à l'image de la sève dans la plante: indispensable et source de vie. Une gestion rigoureuse de la qualité et de la quantité est le garant d'une production contrôlée et maîtrisée.

2-2 Types des serres:

Il existe plusieurs types de serres, la classification se fait en général, selon les formes données par les cadres porteurs qui constituent l'assemblage, on distingue ainsi, deux principaux types appartenant à deux grandes familles de serres : les serres tunnels et les serres chapelles.

- **Les serres tunnels:**

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



Figure 3: serre arc gothique

Chapitre 2 : Internet des objets

1-Introduction

Depuis la fin des années 1980, Internet a connu une évolution spectaculaire, se transformant en un vecteur essentiel de diffusion de l'information. Il s'est imposé comme une infrastructure cruciale dans de nombreux domaines, tant pour les individus que pour les entreprises et les institutions. Cependant, ses capacités d'expansion, au-delà des ordinateurs et terminaux mobiles, restent considérables. En effet, Internet devrait faciliter l'interaction entre un nombre croissant d'objets connectés, que ce soit entre eux ou avec nous-mêmes.

Chaque année, de nouveaux termes émergent, la plupart étant liés au monde moderne et à la technologie. Récemment, le terme "Internet des objets" a fait son apparition, symbolisant la nouvelle génération d'Internet ou de réseau. Ce paradigme offre la possibilité aux appareils interconnectés de se comprendre mutuellement via un protocole Internet. Ces dispositifs incluent des instruments, des capteurs, des actionneurs, ainsi que divers outils d'intelligence artificielle.

L'IoT a connu un développement rapide et massif : depuis 2014, le nombre d'objets connectés a dépassé celui des humains connectés, et on prévoit que 50 milliards d'objets seront connectés d'ici 2020.

Ce chapitre se consacre à une étude approfondie des systèmes IoT afin de comprendre leurs caractéristiques et leurs applications. L'objectif est de développer un système Internet des Objets pour la gestion intelligente des serres agricoles.

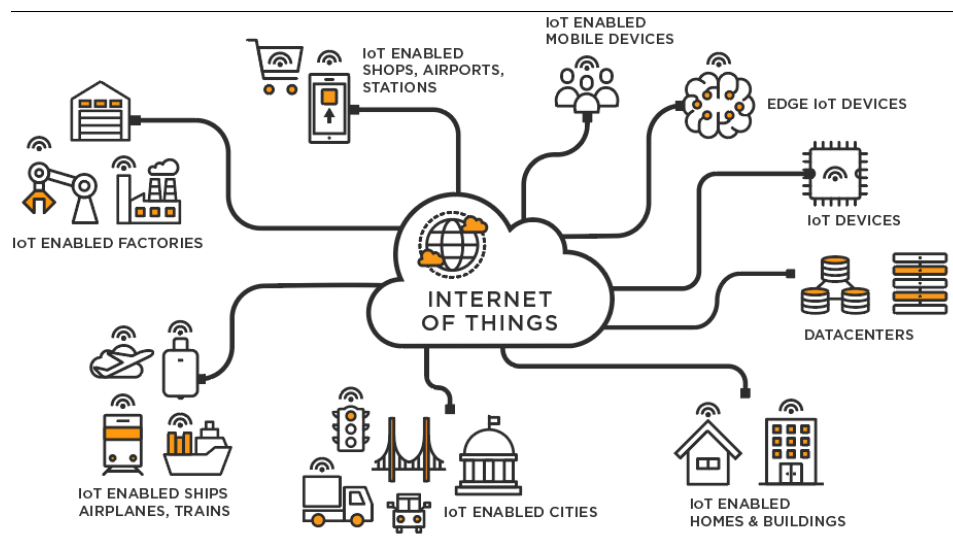


Figure 4: internet of things

Link : ([What is the Internet of Things \(IoT\)? | TIBCO Software](#))

2- Généralité sur internet des objets

2-1 Définition d'internet des objets :

Le terme "Internet des objets" n'a pas encore une définition standard, car le concept est en cours de construction. Il existe plusieurs définitions de l'IoT, mais celle qui est la plus pertinente pour notre travail de recherche est proposée par Weill et Souissi. Selon eux, l'IoT est une extension de l'Internet actuel vers tout objet capable de communiquer directement ou indirectement avec des équipements électroniques connectés à l'Internet.

Cette expansion de l'Internet présente des enjeux technologiques, économiques et sociaux importants, notamment en termes d'économies potentielles grâce à l'ajout de technologies favorisant la standardisation, surtout en matière de communication. Il est essentiel de garantir la protection des droits et des libertés individuelles dans ce nouveau domaine.

L'IoT peut être considéré comme l'infrastructure dynamique d'un réseau global. Ce réseau global possède des capacités d'auto-configuration basées sur des standards et des protocoles de communication interopérables. Dans ce réseau, les objets physiques et virtuels ont des identités, des attributs physiques, des personnalités virtuelles et des interfaces intelligentes, et ils sont intégrés de manière transparente.

D'un point de vue technique, l'IoT implique l'identification numérique directe et normalisée d'un objet physique grâce à un système de communication sans fil tel qu'une puce RFID, Bluetooth ou Wi-Fi. Cette approche permet une représentation simplifiée de la nouvelle dimension de l'IoT.

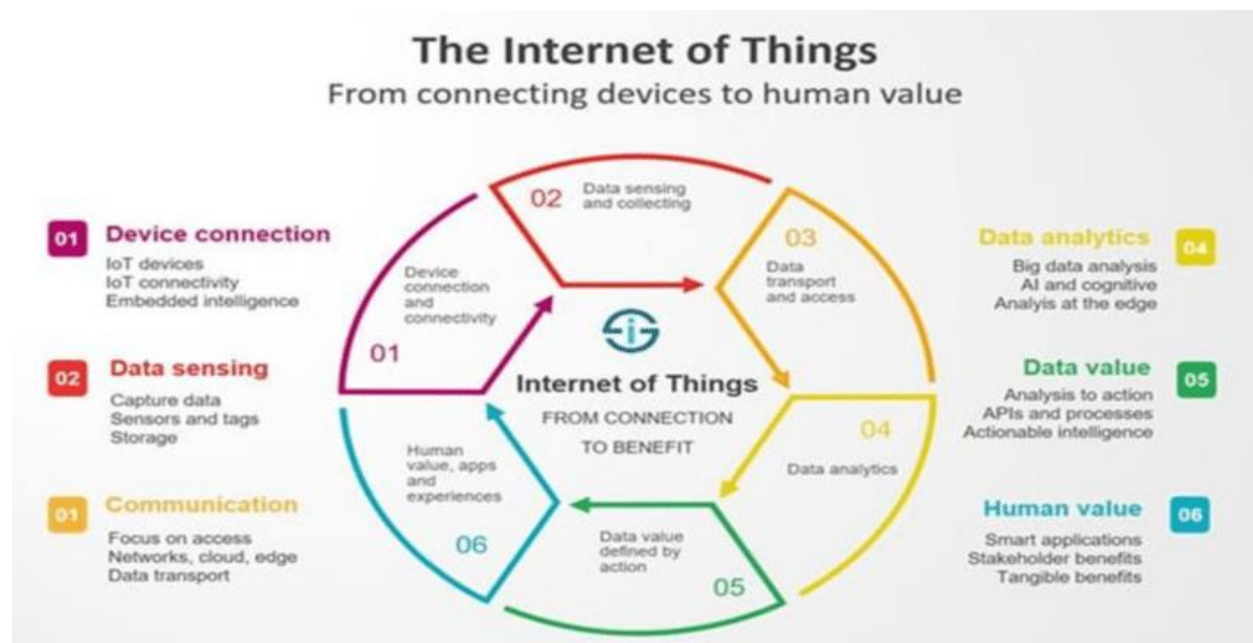


Figure 5: représentation simplifiée de la nouvelle dimension de l'IoT

Link: ([Une nouvelle dimension pour l'IoT | Download Scientific Diagram \(researchgate.net\)](https://www.researchgate.net/publication/351111111))

2-2 Intérêt de l'internet des objets :

L'Internet des objets (IoT) présente plusieurs avantages et intérêts dans divers domaines. Voici quelques-uns des points clés :

- Automatisation et Optimisation : L'IoT permet l'automatisation des tâches et des processus, ce qui contribue à une utilisation plus efficace des ressources. Il offre la possibilité d'optimiser les opérations et de réduire les coûts.
- Collecte de Données en Temps Réel : Les objets connectés génèrent une grande quantité de données en temps réel. Cette collecte de données permet une prise de décision plus rapide et éclairée, améliorant ainsi l'efficacité des systèmes.
- Optimisation Agricole : Dans l'agriculture, l'IoT est utilisé pour surveiller et gérer les cultures, améliorant ainsi le rendement agricole grâce à une utilisation plus précise des ressources telles que l'eau et les engrais.
- Sécurité et Surveillance : L'IoT peut être appliqué pour renforcer la sécurité, que ce soit dans les systèmes industriels, les villes intelligentes ou même à domicile. Les capteurs et dispositifs connectés peuvent détecter des anomalies et assurer une surveillance constante.

2-3 Inconvénients de l' internet des objets:

- Sécurité et Confidentialité : Les préoccupations liées à la sécurité et à la confidentialité des données sont importantes, car la multiplication des dispositifs connectés accroît les points d'accès potentiels pour les cyberattaques.
- Complexité et Interopérabilité : L'intégration de nombreux dispositifs et plateformes IoT peut être complexe, nécessitant des normes et des protocoles d'interopérabilité pour garantir une communication fluide.
- Dépendance aux Technologies : Une défaillance du réseau ou des dispositifs connectés peut entraîner des interruptions et des perturbations, soulignant la dépendance croissante à la technologie.
- Coûts Initiaux et Frais de Maintenance : Les coûts initiaux d'installation des infrastructures IoT peuvent être élevés, et les frais de maintenance doivent également être pris en compte.

2-4 Les composantes d'un système d'internet des objets

L'Internet des Objets (IoT) ne se réduit pas à une simple technologie, mais plutôt à un système complexe composé de plusieurs éléments interconnectés. Cette intégration engendre une complexité que l'interopérabilité, bien que bénéfique, ne peut totalement éliminer. Les principaux systèmes technologiques nécessaires au fonctionnement de l'IoT comprennent :

1. Identification :

- Reconnaissance unique de chaque objet avec des technologies telles que la radio-identification complexe, les ondes acoustiques de surface et l'ADN.

2. Détection :

- Collecte d'informations de l'environnement à l'aide de capteurs d'humidité, de capteurs de température, de dispositifs de détection portables, etc.

3. Protocoles de Communication :

- Connexion des systèmes entre eux par le biais de divers protocoles tels que Wi-Fi, Bluetooth, IEEE 802.15.4, Z-wave, LTE-Advanced, Near Field Communication (NFC), bande passante ultra large (UWB), etc.

4. Calculateurs :

- Utilisation d'unités de traitement matériel telles que microcontrôleurs, microprocesseurs, système sur des puces (SoC), matrices de portes programmables (FPGA) et applications logicielles.

5. Platforms :

- Matérielles (Arduino, UDOO, FriendlyARM, Intel Galileo, Raspberry PI, Gadgeteer, etc.).
- Logicielles (TinyOS, LiteOS, RIoTOS, etc.).

6. Services :

- Classes de services incluant l'identité, l'agrégation d'informations, la collaboration et l'omniprésence. Les services liés à l'identité posent les bases pour d'autres types de services.

7. Sémantique :

- Capacité d'extraire intelligemment des connaissances pour fournir les services requis, incluant la découverte des ressources, l'utilisation des ressources, la modélisation des informations, et la reconnaissance des données.

2-5 Fonctionnement de l'Internet des Objets :

Pour établir un "Internet des Objets", plusieurs éléments sont nécessaires, tels que des capteurs, des dispositifs, une connectivité, un traitement des données et une interface utilisateur. Les

capteurs collectent des données de leur environnement, puis ces données sont envoyées au cloud via divers moyens de connectivité (cellulaire, satellite, Wi-Fi, Bluetooth, etc.).

Dans le cloud, le logiciel effectue un traitement sur les données, de manière simple (vérification de la température) à complexe (analyse vidéo par ordinateur). L'utilisateur peut intervenir en cas de conditions spécifiques (température élevée, intrusion) via des alertes (email, texte) ou une interface utilisateur.

Le fonctionnement peut également être automatisé selon des règles prédéfinies. Par exemple, le système peut ajuster la température automatiquement ou alerter automatiquement les autorités en cas d'intrusion. La figure ci dessous illustre de manière schématique le fonctionnement de l'Internet des Objets.

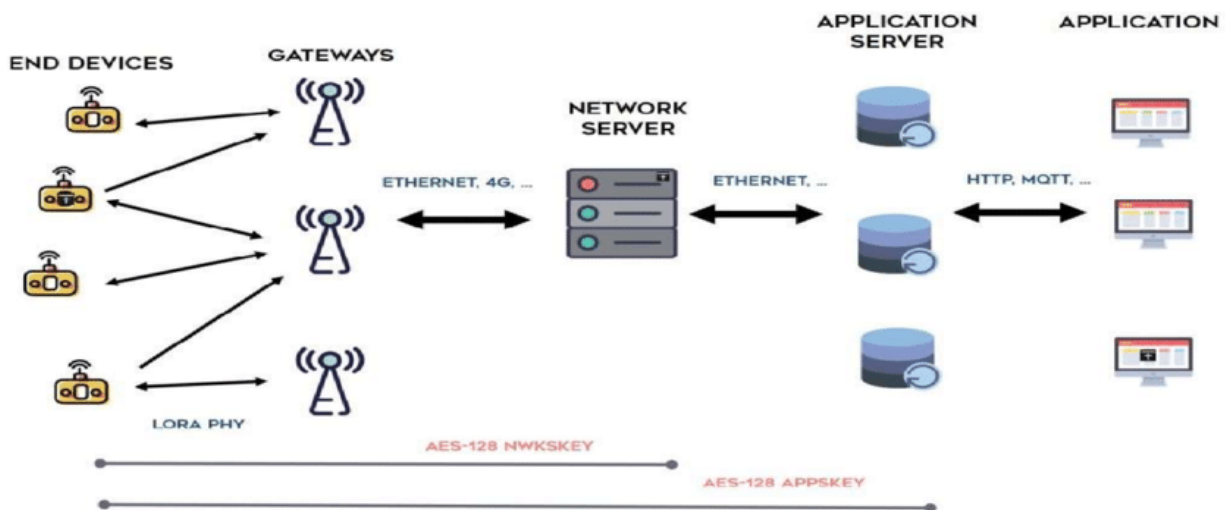


Figure 6: Schéma explicatif de fonctionnement de l'IoT

Link : ([Schéma du réseau de l'IoT \[19\] | Download Scientific Diagram \(researchgate.net\)](#))

3- Protocoles de communication

3-1 Bluetooth

En 1994, la société suédoise Ericsson a inventé le protocole Bluetooth, un standard de transfert de données sans fil. Ce protocole utilise une faible bande passante, ce qui le rend idéal pour transférer des données sur de courtes distances tout en consommant très peu d'énergie. Présent dans la plupart des téléphones mobiles, Bluetooth facilite la communication entre deux téléphones ou entre un téléphone et un objet connecté.

Les applications de Bluetooth sont nombreuses, allant des oreillettes sans fil aux montres intelligentes, en passant par les moniteurs de fréquence cardiaque, les enceintes portatives, les stations météo, les thermostats, et bien d'autres. Il est également utilisé sur des capteurs statiques tels que les "beamers" pour mesurer des flux, comme le nombre de clients dans un magasin.

L'objectif principal de Bluetooth est de permettre la transmission de données ou de voix entre des appareils équipés d'un circuit radio peu coûteux, sur des distances allant d'une dizaine de mètres à moins d'une centaine de mètres, tout en minimisant la consommation électrique. La figure suivante résume l'application de Bluetooth.

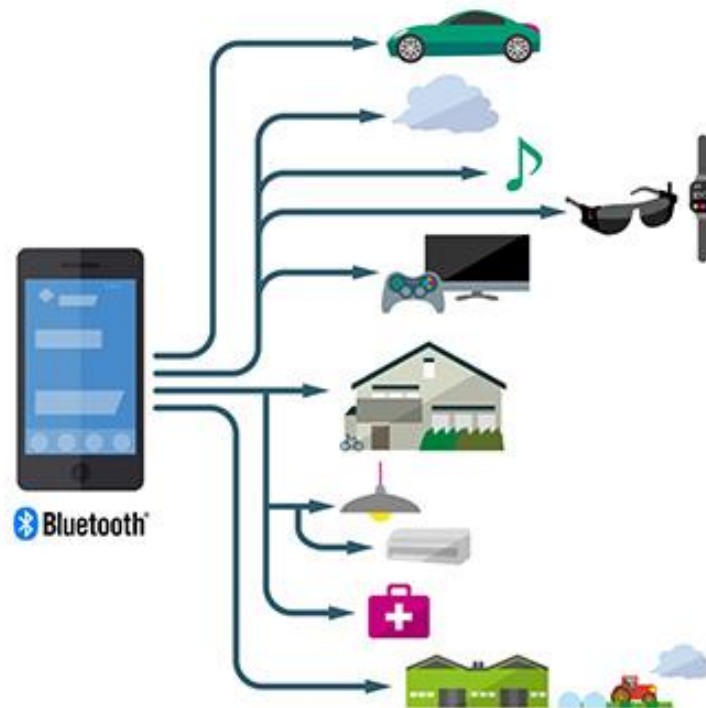


Figure 7:application de Bluetooth

3-2 Zigbee

Zigbee est un protocole de communication radio spécialement créé pour les applications de domotique, ayant une portée moyenne d'environ 10 mètres. Il est parfait pour le transfert de petites quantités de données, économe en énergie, et adapté aux appareils fonctionnant sur pile ou batterie, notamment les capteurs. Zigbee est conçu pour des échanges de données à bas débit.

Ce protocole fonctionne en réseau maillé, où chaque nœud peut recevoir, envoyer et relayer des données. Il est souvent utilisé dans des dispositifs tels que les détecteurs de fumée. La figure ci-dessous donne un aperçu des différentes applications de Zigbee dans le domaine de la domotique.



Figure 8: application de zigbee

3-3 Wi-Fi

Le Wi-Fi englobe un ensemble de protocoles de communication sans fil, offrant des connexions à haut débit sur des distances allant de 20 à 100 mètres. Il constitue un réseau local sans fil efficace, mais énergivore, adapté aux appareils connectés à une source d'alimentation électrique régulière. Cette technologie permet un transfert rapide de grandes quantités de données.

Le Wi-Fi est particulièrement approprié pour les dispositifs alimentés par secteur. Il existe différentes normes Wi-Fi, chacune offrant une portée et un débit variables. Cette polyvalence ouvre la voie à de nombreuses applications pratiques. Le Wi-Fi est compatible avec IPv4 et IPv6, favorisant le développement de nouveaux algorithmes distribués.



Figure 9: applications de reseau Wi-Fi

Link: ([Working with WiFi - SparkFun Electronics](#))

4- Les applications internet of things

Agriculture intelligente

Dans ce domaine, des réseaux de capteurs interconnectés à l'IOT peuvent être utilisés pour la supervision de l'environnement des cultures. Ceci permettra une meilleure aide à la décision en agriculture, notamment pour optimiser l'eau d'irrigation, l'usage des intrants, et la planification de travaux agricoles. Ces réseaux peuvent être aussi utilisés pour récolter les informations utiles sur l'état du sol, taux d'humidité, taux des sels minéraux, etc. et envoyer ces informations au fermier pour prendre les mesures nécessaires garantissant la bonne production ainsi lutter contre la pollution de l'air, du sol et des eaux et améliorer la qualité de l'environnement en general. Les figures suivante montre l'agriculteur qui utilise l'iot



Figure 10: agriculture moderne



Figure 11: IOT dans l'agriculture

Maison intelligente

Le confort de la vie à domicile est enrichi en le rendant plus pratique et facile à gérer à distance, permettant le contrôle des appareils et des systèmes domestiques tels que le micro-ondes, le four, la climatisation, les systèmes de chauffage, et bien d'autres.

Soins de santé intelligents

Les performances des applications de soins de santé sont améliorées, en intégrant des capteurs et des actionneurs aux patients et à leurs médicaments pour surveiller et suivre les patients. Par exemple, en collectant et en analysant les données corporelles des patients avec des capteurs et en fournissant des données analysées à un centre de traitement, les soins cliniques pourraient surveiller les états physiologiques des patients en temps réel et prendre des mesures appropriées si nécessaire.

Automatisation industrielle

Avec une intervention humaine minimale, les dispositifs robotiques sont automatisés pour accomplir des tâches de fabrication. Les opérations, fonctionnalités, et taux de productivité des machines sont contrôlés et surveillés de manière automatique. La Figure ci dessous illustre différentes applications de l'Internet des Objets (IoT).



Figure 12: Les différentes applications d'IOT

5- Conclusion

Dans ce chapitre, nous avons introduit l'Internet des Objets (IoT), une avancée majeure appelée à façonner l'avenir de multiples domaines. Nous avons exploré la notion d'objets connectés, considérés comme les unités fondamentales, ainsi que les composants essentiels nécessaires à la mise en place d'un système IoT.

Au cours de notre exploration, nous avons examiné diverses technologies, architectures, et protocoles associés à cette nouvelle approche révolutionnaire. L'Internet des Objets, avec sa multitude d'applications, trouve notamment un terrain propice dans le domaine de l'agriculture intelligente. L'adoption de cette technologie a permis de minimiser l'intervention humaine, rendant l'ensemble du processus non seulement rentable, mais également plus précis. Cette avancée marque ainsi le début d'une ère prometteuse où la connectivité intelligente redéfinit notre manière d'interagir avec le monde qui nous entoure.

Chapitre 3 : Conception et Réalisation

1- Partie Matérielle:

1-1 Choix du microcontrôleur

L'une des étapes les plus importantes à la réalisation de notre projet est le choix du microcontrôleur, il est nécessaire de choisir le plus adapté à nos objectifs, Il existe beaucoup de critères de sélection dont nous devons tenir compte comme :

1. Le nombre d'entrées/sorties analogiques.
2. Le nombre d'entrées/sorties numériques.
3. Puissance de calcul suffisamment élevée pour gérer des algorithmes en temps réel.
4. Taille de la mémoire programmée (pour contenir l'ensemble du programme).
5. La taille de la mémoire RAM (pour les calculs que le microcontrôleur doit effectuer).
6. La mémoire EPROM (si on a besoin que certaines données soient sauvegardées si l'alimentation se coupe).
7. Le prix et la disponibilité du microcontrôleur sur le marché. Idéalement : la possibilité de se connecter à internet pour envoyer et recevoir des données depuis un serveur distant [32].

Le microcontrôleur ESP32

L'ESP32 est un microcontrôleur à faible consommation d'énergie et à faible coût, fabriqué par Espressif Systems. Il s'agit d'une puce système à microcontrôleur (SoC) qui combine un processeur à architecture Xtensa dual-core, une connectivité Wi-Fi et Bluetooth intégrée, ainsi que diverses interfaces périphériques.

L'ESP32 est un microcontrôleur très polyvalent largement utilisé dans le domaine de l'Internet des objets (IoT). Il se distingue par son processeur dual-core Xtensa 32 bits, offrant des fréquences d'horloge allant jusqu'à 160 MHz, surpassant ainsi son prédécesseur en termes de puissance de traitement

1. Processeur RISC 32 bits cadencé à 80 MHz (par défaut) ou à 160 MHz.
2. 64 Ko de RAM pour les instructions et 96 Ko pour les données.
3. Mémoire flash externe QSPI entre 512 Ko et 4 Mo selon les modèles.
4. Puce Wi-Fi 2,4 GHz (802.11 b/g/n) avec antenne intégrée.
5. Authentification WEP ou WPA/WPA2, ou réseaux ouverts.
6. 16 entrées/sorties numériques GPIO.

7. PWM / ADC 10 bits (variante 12E).

8. UART / I2C / I2S / SPI.

9. Alimentation en 3,3 V.

10. Consommation : entre 60 mA et 215 mA en fonctionnement normal, quelques dizaines de μA en veille.

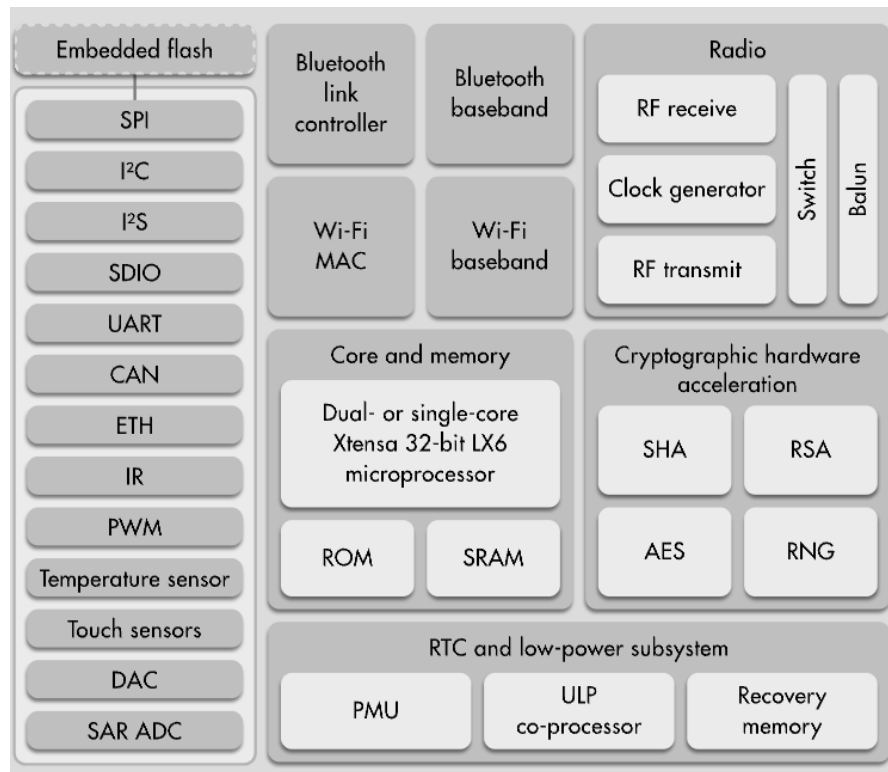


Figure 13: Architecture ESP32

1-2 Brochage de ESP32 :

Le module ESP32 NodeMCU WROOM 32 version 1.1 dispose de plusieurs broches qui peuvent être utilisées pour différentes fonctions. Voici les broches principales de l'ESP32 NodeMCU WROOM 32 version 1.1 :

1. Broches d'alimentation :

3V3 : Tension de sortie de 3.3V (alimentation).

GND : Terre (masse).

2. Broches de communication :

RX0, TX0 : Ports série pour la communication UART.

SDA, SCL : Ports pour la communication I2C.

MOSI, MISO, SCK : Ports pour la communication SPI.

3. Broches GPIO (General Purpose Input/Output) :

GPIO0 à GPIO39 : Broches utilisées pour les entrées/sorties numériques.

4. Broches analogiques :

A0 : Entrée analogique.

5. Broches de contrôle :

EN : Activation du module.

RST : Réinitialisation.

BOOT : Contrôle du mode de démarrage.

6. Broches spéciales :

IO0 : Utilisée lors de la programmation pour entrer en mode de programmation (flash).

IO13 : Témoin d'activité intégré (peut être utilisé pour la programmation et le débogage).

ESP32 Wroom DevKit Full Pinout

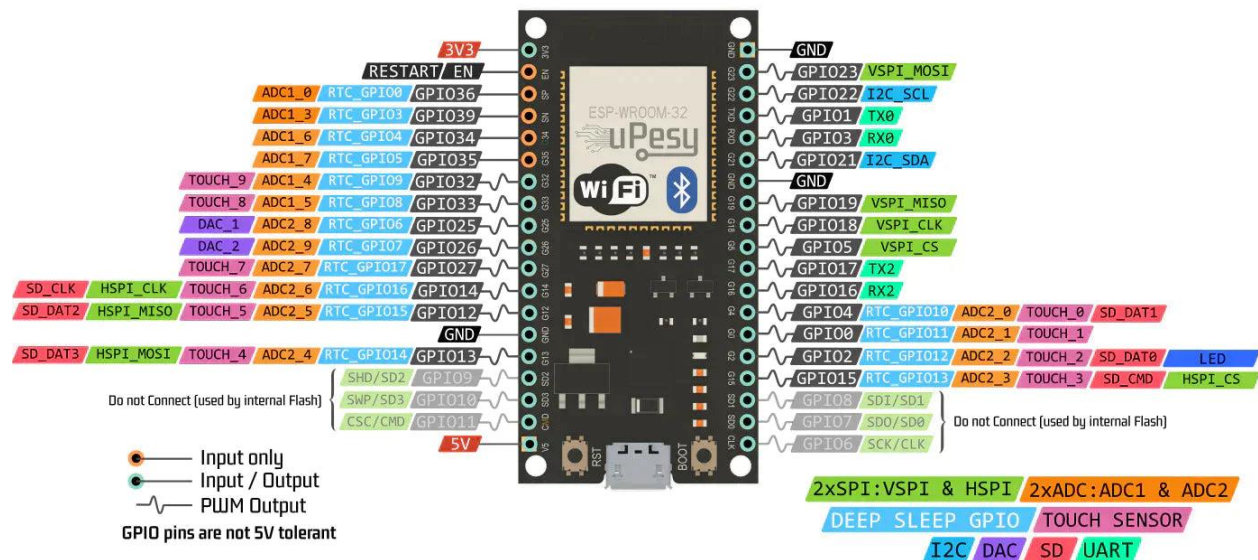


Figure 14: Schéma du broches ESP32 Wroom Devkit

Il est important de noter que certaines broches peuvent avoir des fonctions spéciales pendant le démarrage ou la programmation, donc il est recommandé de consulter la documentation officielle de l'ESP32 NodeMCU WROOM 32 version 1.1 pour plus de détails sur l'utilisation de chaque broche et les précautions à prendre lors de la programmation.

❖ Capteurs utilisés :**1.Capteur DHT11:**

Le choix du capteur DHT11 est souvent privilégié en raison de sa simplicité de mise en œuvre et de son coût abordable. Ce capteur combine un capteur de température basé sur une NTC (Coefficient de Température Négatif) avec un capteur d'humidité résistif. Un microcontrôleur intégré se charge de mesurer, convertir et transmettre les données. De plus, le DHT11 est préalablement calibré en usine, et ses paramètres de calibration sont stockés dans sa mémoire interne.

Ce capteur offre une programmation aisée grâce aux bibliothèques disponibles pour les plates-formes Arduino et Raspberry Pi. Sa facilité d'utilisation en fait un choix populaire pour les projets nécessitant la surveillance de la température et de l'humidité.

La Figure ci dessous fournit une illustration détaillée du capteur DHT11, présentant ses composants internes et mettant en évidence son architecture. La représentation graphique facilite la compréhension de la structure du capteur, ce qui est bénéfique pour les utilisateurs souhaitant intégrer ce composant dans leurs projets.

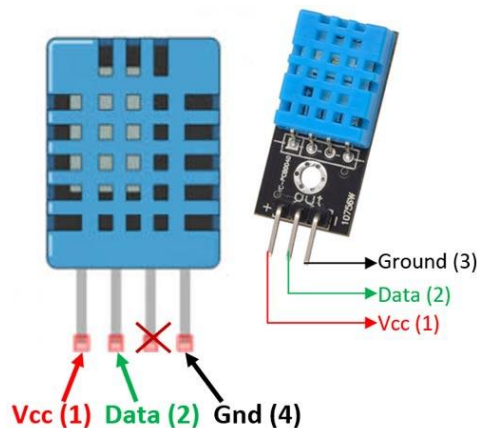


Figure 15:Description de DHT11

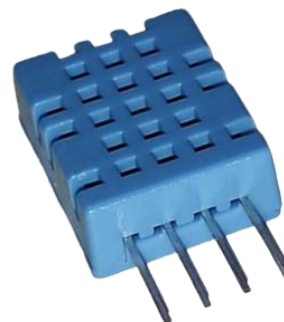


Figure 16 :DHT11

Le capteur de température et d'humidité DHT11 communique avec un microcontrôleur via un port série. Livré avec résistance de "pull-up" pour la sortie data.

Le tableau suivant représente une comparaison entre les 2 capteur de dht11 et dht22 :

DHT11	DHT22
<ul style="list-style-type: none"> o ultra low cost o Alimentation 3 à 5 V et E / S o Utilisation de courant maximale de 2,5 mA pendant la conversion (lors de la demande de données) o Bon pour des lectures d'humidité de 20 à 80% avec une précision de 5% o Bon pour les lectures de température de 0 à 50 ° C \pm 2 ° C de précision o Pas plus de 1 Hz de fréquence d'échantillonnage (une fois par seconde) o Taille du corps 15,5 mm x 12 mm x 5,5 mm o 4 broches avec espacement de 0,1 " 	<ul style="list-style-type: none"> o À bas prix o Alimentation 3 à 5 V et E / S o Utilisation de courant maximale de 2,5 mA pendant la conversion (lors de la demande de données) o Bon pour des lectures d'humidité de 0 à 100% avec une précision de 2 à 5% o Bon pour des lectures de température de -40 à 80 ° C \pm 0,5 ° C de précision o Pas plus de 0,5 Hz de fréquence d'échantillonnage (une fois toutes les 2 secondes) o Taille du corps 15,1 mm x 25 mm x 7,7 mm o 4 broches avec espacement de 0,1 "

Tableau 1: Comparaison entre DHT 11 et DHT22

1- Capteur d'humidité du Sol :

La surveillance précise de l'humidité du sol représente une pièce maîtresse essentielle dans l'optimisation de l'approvisionnement en eau des cultures agricoles. Assurer une alimentation en eau adéquate, au moment opportun, est crucial pour favoriser une croissance saine et maximiser le rendement des cultures. Cela s'avère d'autant plus crucial dans le contexte où une irrigation bien ajustée permet une utilisation efficace de l'eau et des éléments nutritifs, minimisant ainsi le gaspillage et contribuant à une agriculture plus durable.

L'efficacité d'une irrigation atteint son apogée lorsque la quantité d'eau fournie à la culture est parfaitement adaptée à ses besoins spécifiques, et que le sol est capable de retenir cette humidité de manière optimale. Cela implique une gestion fine, où la mesure précise de l'humidité du sol joue un rôle fondamental. En adoptant cette approche, on garantit que les décisions d'irrigation sont basées sur des données concrètes, évitant ainsi le risque de sur-arrosage ou de sous-arrosage.

La durée des opérations d'irrigation revêt une importance particulière. En prenant le temps de mesurer l'humidité du sol de manière régulière, les agriculteurs peuvent ajuster les schémas d'irrigation de manière intelligente. Cette démarche proactive permet d'adapter les pratiques

d'irrigation en fonction des besoins réels des cultures, favorisant ainsi une utilisation plus efficace des ressources disponibles.

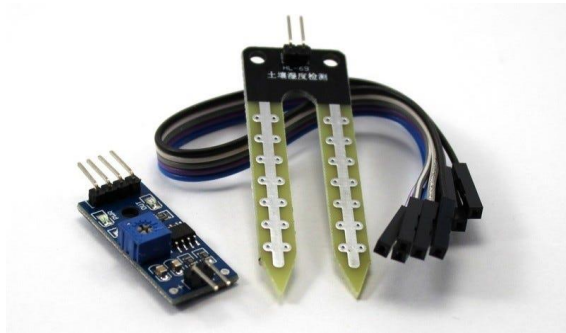


Figure 17: capteur d'humidité du Sol

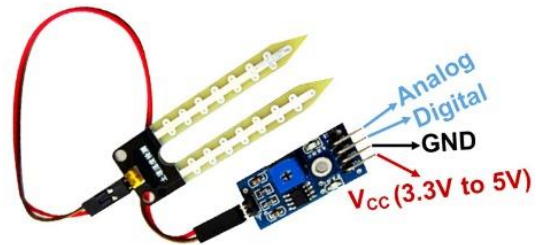


Figure 18: Description de capteur d'humidité du Sol

Soil moisture sensor yl-38	Capacitive Soil Moisture Sensor SKU:SEN0193
<ul style="list-style-type: none"> o Tension de fonctionnement: DC 3,3 V - 5 V o Signal de tension de sortie: 0 ~ 4,2 V o Courant: 35mA o LED: indicateur d'alimentation (rouge) et indicateur de sortie de commutation numérique (vert) o Taille: 60 x 20 x 5 mm 	<ul style="list-style-type: none"> o Tension de fonctionnement: 3,3 ~ 5,5 VDC o Tension de sortie: 0 ~ 3.0VDC o Courant de fonctionnement: 5mA o Interface: PH2.0-3P o Dimensions: 3,86 x 0,905 pouces (L x l) o Poids: 15g

Tableau 2: Comparaison entre Capteur SKU : SEN0193 & yl-38

En résumé, la surveillance continue de l'humidité du sol constitue une stratégie incontournable pour prendre des décisions d'irrigation informées et améliorer la gestion globale de l'eau dans l'agriculture. En intégrant des technologies de mesure avancées, les agriculteurs peuvent non seulement maximiser la productivité de leurs cultures, mais aussi contribuer de manière significative à la préservation des ressources naturelles.

2- Capteur du gaz MQ2:

Le module de capteur MQ2 est un capteur de gaz électronique qui peut détecter la concentration de gaz dans l'air tels que le propane, l'alcool, la fumée, le CO, l'hydrogène, le méthane et le monoxyde de carbone . Il contient un matériau de détection dont la résistance change lorsqu'il entre en contact avec le gaz . Le capteur MQ2 est également connu sous le nom de chemisorbeur . Il fonctionne sur une tension de 5V DC et consomme environ 800 mW . Le module peut détecter des concentrations de propane, de fumée, d'alcool, d'hydrogène, de méthane et de monoxyde de carbone allant de 200 à 10000 ppm . La concentration de 1 ppm signifie que si vous pouviez compter un million de molécules de gaz, 1 serait du gaz détecté et les 999 999 autres seraient d'autres gaz . Notez que le capteur MQ2 détecte plusieurs gaz, mais ne peut pas les identifier . Par conséquent, il est préférable de mesurer les variations de densité de gaz connus plutôt que de détecter celui qui change . Le capteur MQ2 est un capteur à semi-conducteur MOS (Metal Oxide Semiconductor) . Les capteurs à oxyde métallique sont également connus sous le nom de chimiorésistances car la détection est basée sur le changement de résistance du matériau de détection lorsqu'il est exposé aux gaz . Le capteur MQ2 est chauffé par un élément chauffant et est recouvert de deux couches de maille en acier inoxydable fin connues sous le nom de "réseau anti-explosion" . Il assure que l'élément chauffant à l'intérieur du capteur ne provoque pas d'explosion car nous détectons des gaz inflammables. Il protège également le capteur et filtre les particules en suspension, ne permettant que les éléments gazeux de passer à travers la chambre .

Dans notre cas on a choisi d'utiliser ce capteur comme un capteur de dioxyde de carbone CO2 et ci dessous est une figure qui représente le capteur



Figure 19: un capteur de Gaz MQ2

3- Capteur de niveau d'eau

Il mesure 4 cm, capteur à plonger dans le liquide que l'on veut mesurer. Il donne le niveau d'eau, correspondant. Par exemple, si le capteur est plongé, dans un support, où l'eau ne doit pas dépasser 20 cm, si le capteur donne comme valeur, 1 cm, le volume d'eau sera donc de 19 cm. Pour interpréter les données du capteurs, il est nécessaire de relever les valeurs analogiques

correspondantes à chaque niveau d'eau du capteur. Pour ce faire, il faut plonger le capteur dans un verre d'eau, puis le jauger, et relever les différentes valeurs. Le capteur fait 4 cm, il est nécessaire de mesurer les valeurs, données, lorsque le capteur est n'est pas immergé, dans l'eau, puis lorsqu'il l'ait à 5 mm, 10 mm, etc.... jusqu'à arriver à 40 mm.

Les mesures réalisées sont affichées en ppm. L'acronyme « ppm » signifie « partie par million ». Il s'agit d'une unité de mesure communément utilisée par les scientifiques, notamment pour calculer le taux de pollution dans l'air et plus globalement dans l'environnement. Comme son nom l'indique, le ppm permet de savoir combien de molécules de polluant on trouve sur un million de molécules d'air. Il permet donc de rendre compte de manière assez simple de la quantité de pollution dans une masse d'air donnée et de l'impact nocif de ces polluants sur l'atmosphère.

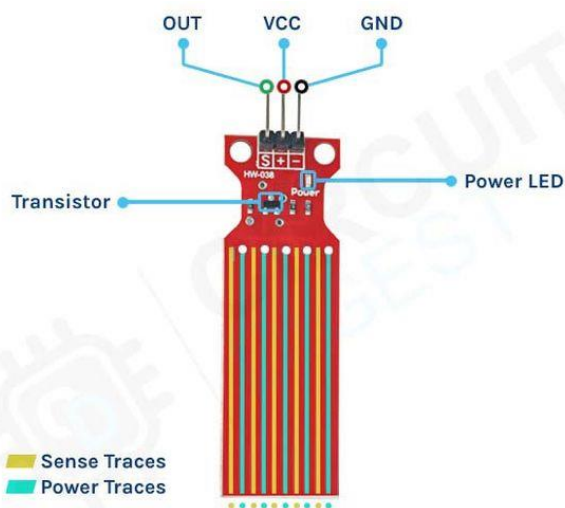


Figure 20:Description de capteur de niveau d'eau

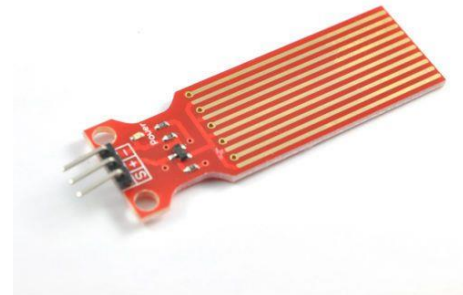


Figure 21:un capteur de niveau d'eau

4- Les actionneurs

Pompe électrique :

C'est une pompe submersible qui fonctionne sur dc 3-5v avec une rentabilité et portable. Elle est capable de prendre environ 120 litres par heure avec une utilisation en courant extrêmement faible. Le niveau d'eau doit être plus élevé car si le moteur est utilisé sans eau, il peut endommager les pièces de cet appareil en raison d'une surchauffe. Il existe de nombreuses applications telles que le débit d'eau contrôlé de la fontaine, les systèmes hydroponiques, le système d'arrosage contrôlé du jardin .

Relais:

C'est un appareil de commutation comme montre la figure ci dessous. Pour contrôler mécaniquement un commutateur, de nombreux relais utilisent un électro-aimant, mais certains autres principes fondamentaux peuvent également être utilisés comme des relais à semi-conducteurs. Lorsqu'il est important de faire fonctionner un circuit au moyen d'un signal indépendant de faible puissance ou si différents circuits sont gérés au moyen d'un seul signal, des relais sont utilisés. Le relais agit donc comme un interrupteur automatisé qui fonctionne sur un circuit à courant élevé utilisant un signal à courant faible.



Figure 22:Un Relais

Caractéristique du relais:

- Module de relais 5 V.
- Compatible avec Arduino.
- Tension nominale : 5 V en CC.
- Courant nominal du relais : 10 A à 250 V en CA, 10 A à 30 V en CC (non inductif)

Ventilateur

- Ventilateur 5V cc avec connecteur 2 broches
- Structure en PBT thermoplastique noir
- Tension de fonctionnement : 5Vdc
- Vitesse (RPM) : 6500tr/min
- Débit d'air : 9,8 CFM, bruit : 15 dBA



Figure 23: une ventillateur

1-3 Affichage

Afficheur lcd_i2c

Un afficheur LCD I2C offre une solution pratique et conviviale pour afficher des informations sur deux lignes de 16 caractères chacune. Il simplifie la connexion avec un microcontrôleur tel qu'Arduino ou Raspberry Pi grâce à son interface I2C, nécessitant seulement 4 broches à l'arrière de l'écran. Le rétro-éclairage vert met en valeur les caractères blancs, assurant une lisibilité optimale.

Pour une expérience utilisateur améliorée, ce module propose un contraste ajustable via un potentiomètre intégré, permettant une personnalisation selon les besoins spécifiques du projet. De plus, l'adresse I2C du module est facilement configurable grâce aux jumpers A1, A2 et A3, offrant une flexibilité accrue lors de l'intégration dans des systèmes complexes.

Ce module est particulièrement adapté pour les projets interactifs impliquant un écran LCD, offrant une solution élégante qui n'utilise qu'un nombre minimal de broches du microcontrôleur. Sa conception simplifiée et ses fonctionnalités avancées en font un choix idéal pour les développeurs et les amateurs souhaitant intégrer un affichage LCD de manière efficace et esthétique dans leurs projets.

Caracteristiques techniques

- 2 lignes de 16 caractères.
- Rétro-éclairage bleu, caractères blancs
- I2C, adresse de base 0x20
- 8 adresses disponibles en fonction des jumpers A1, A2 et A3
- Alimentation : 5V
- Dimensions : 36 x 79 x 20 mm
- Poids : 40 grammes



Figure 24:Afficheur LCD avec interface I2C

2- Partie logicielle :

2-1 Plateforme : ARDUINO IDE :

Arduino IDE est un logiciel open source qui est principalement utilisé pour écrire et compiler le code dans le module Arduino. Il s'agit d'un logiciel officiel Arduino, ce qui rend la compilation de code trop facile pour que même une personne ordinaire sans connaissances techniques préalables puisse se mouiller avec le processus d'apprentissage. Il est facilement disponible pour les systèmes d'exploitation comme MAC, Windows, Linux et s'exécute sur la plate-forme Java qui comprend des fonctions et des commandes intégrées qui jouent un rôle essentiel pour le débogage, l'édition et la compilation du code dans l'environnement. Une gamme de modules Arduino disponibles, notamment Arduino Uno, Arduino Mega, Arduino Leonardo, Arduino Micro et bien d'autres. Chacun d'eux contient un microcontrôleur sur la carte qui est réellement programmé et accepte les informations sous forme de code. Le code principal, également appelé croquis, créé sur la plate-forme IDE générera finalement un fichier hexadécimal qui sera ensuite transféré et téléchargé dans le contrôleur de la carte. L'environnement IDE contient principalement deux parties de base: l'éditeur et le compilateur où l'ancien est utilisé pour écrire le code requis et plus tard est utilisé pour compiler et télécharger le code dans le module Arduino donné. Cet environnement prend en charge les langages C et C ++



Figure 25: Logo Arduino

Arduino IDE libraries utilisée :

- DHT.h
- Wire.h
- LiquidCrystal_I2C.h

```
1  #include "DHT.h"
2  #include <Wire.h>
3  #include <LiquidCrystal_I2C.h>
4
```

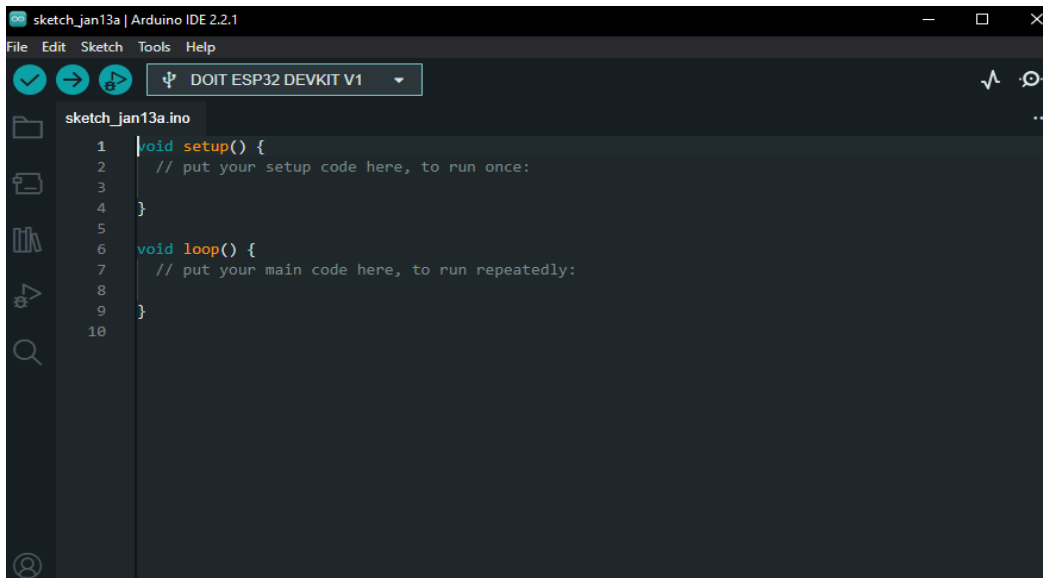


Figure 26: Interface Arduino IDE

2-1.1 Installation des bibliothèques

Pour installer une nouvelle bibliothèque dans l'IDE Arduino:

- ☐ Ouvrir le menu Croquis> Inclure la bibliothèque> Gérer les bibliothèques

Une fois que vous avez ouvert le gestionnaire de bibliothèque, vous pouvez utiliser le menu en haut

pour filtrer les résultats.

- ☐ Cliquez sur la bibliothèque de votre choix, sélectionnez une version dans le menu déroulant, puis cliquez sur installer.

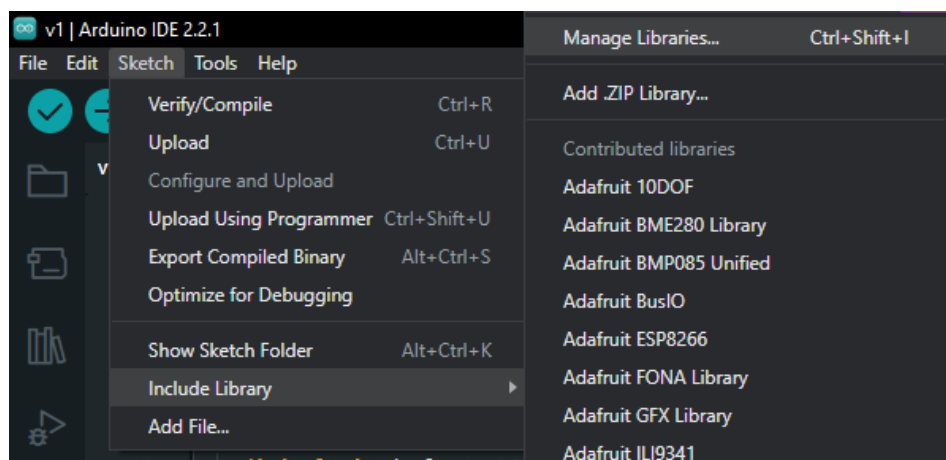


Figure 27: installation bibliotheque arduino IDE

2-1.2 Declaration et configuration des composants :

```

1  #include "DHT.h"
2  #include <Wire.h>
3  #include <LiquidCrystal_I2C.h>
4
5
6  LiquidCrystal_I2C lcd(0x27, 20, 4);
7  #define DHTPIN 23
8  #define DHTTYPE DHT11
9  #define relay1 27
10 #define relay2 13
11 const int moisturePin = 36;
12 int moistureReading;
13 #define POWER_PIN 17
14 #define SIGNAL_PIN 34
15

```

```

16 int value = 0;
17 DHT dht(DHTPIN, DHTTYPE);
18
19 const int temperatureThreshold = 18;
20 const int gazThreshold = 50;
21 const int humsoldesired= 10;
22
23 int Gas_analog = 25;
24 int Gas_digital = 26;
25

```

Figure 28: partie declaration

Ce code initialise plusieurs composants électroniques. Tout d'abord, il utilise la bibliothèque LiquidCrystal_I2C pour configurer un écran LCD en mode I2C. Ensuite, il définit la broche pour le capteur DHT11 qui mesure la température et l'humidité. Deux relais sont également configurés avec des broches spécifiques (relay1 et relay2) pour contrôler des dispositifs externes. Un capteur d'humidité du sol est connecté à la broche moisturePin, et sa lecture est stockée dans la variable moistureReading. Enfin, deux broches, POWER_PIN et SIGNAL_PIN, sont définies pour le contrôle de l'alimentation et du signal, respectivement. L'utilisation de ces composants suggère un projet impliquant la surveillance de l'environnement, le contrôle d'appareils externes et la gestion de l'alimentation.

2-1.3 Partie void setup() :

Explications :

- **Serial.begin(9600);**: Initialise la communication série avec une vitesse de 9600 bauds, permettant la communication avec le moniteur série pour le débogage.
- **dht.begin();**: Initialise le capteur DHT11 pour la mesure de la température et de l'humidité.
- **lcd.init();**: Initialise l'écran LCD en mode I2C.
- **lcd.backlight();**: Allume le rétroéclairage de l'écran LCD.
- **pinMode(relay1, OUTPUT);** et **pinMode(relay2, OUTPUT);**: Configure les broches des relais en tant que sorties pour les contrôler.

- **pinMode(Gas_digital, INPUT);** Configure la broche **Gas_digital** en tant qu'entrée pour lire un signal lié à un capteur de gaz.
- **pinMode(POWER_PIN, OUTPUT);** et **digitalWrite(POWER_PIN, LOW);** Configure la broche **POWER_PIN** en tant que sortie et initialise son état à LOW, indiquant potentiellement une désactivation initiale de l'alimentation sur cette broche.

```
27 void setup() {
28     Serial.begin(9600);
29     dht.begin();
30     lcd.init();
31     lcd.backlight();
32
33     pinMode(relay1, OUTPUT);
34     pinMode(relay2, OUTPUT);
35
36     Serial.begin(9600);
37     pinMode(Gas_digital, INPUT);
38
39     Serial.begin(9600);
40     pinMode(POWER_PIN, OUTPUT);
41     digitalWrite(POWER_PIN, LOW);
42
43 }
```

Figure 29: Partie void setup()

2-1.4 Partie void loop () :

La fonction **loop()** dans le langage Arduino est une boucle infinie qui s'exécute continuellement après l'exécution de la fonction **setup()**. Elle est utilisée pour mettre en œuvre la logique principale du programme.

Cette fonction effectue des lectures à partir de différents capteurs (température, humidité du sol, gaz, niveau d'eau) et affiche ces lectures sur le moniteur série et un écran LCD. En fonction des valeurs lues, elle active ou désactive les relais en accord avec des seuils prédéfinis. Ces seuils (temperatureThreshold, gazThreshold, humsolvedesired) peuvent être ajustés pour déclencher des actions spécifiques en fonction des conditions environnementales mesurées :

1. Lecture des capteurs :

- La fonction commence par lire les valeurs des capteurs connectés, y compris le capteur d'humidité du sol, le capteur DHT11 pour la température et l'humidité, le capteur de gaz, et le capteur de niveau d'eau.

2. Calcul du pourcentage d'humidité du sol :

- Utilise la lecture du capteur d'humidité du sol pour calculer un pourcentage d'humidité du sol.

3. Affichage sur le moniteur série :

- Les lectures des capteurs, y compris la température, l'humidité, le pourcentage d'humidité du sol, le pourcentage de gaz, et la lecture du capteur de niveau d'eau, sont affichées sur le moniteur série pour le débogage.

4. Activation de l'alimentation pour la lecture du capteur de niveau d'eau :

- La fonction active temporairement l'alimentation sur une broche spécifiée (**POWER_PIN**) pour une durée de 10 ms, ce qui pourrait être lié à la lecture d'un capteur de niveau d'eau.

5. Affichage et lecture du capteur de niveau d'eau :

- La fonction lit la valeur du capteur de niveau d'eau après avoir activé l'alimentation. La valeur est ensuite affichée sur le moniteur série.

6. Affichage sur l'écran LCD :

- Les lectures des capteurs, y compris la température, l'humidité, le pourcentage d'humidité du sol, le pourcentage de gaz, et la lecture du capteur de niveau d'eau, sont affichées sur un écran LCD.

7. Contrôle des relais en fonction des seuils :

- La fonction utilise des seuils prédéfinis (**temperatureThreshold**, **gazThreshold**, **humsoildesired**) pour contrôler les relais en fonction des conditions mesurées. Par exemple, elle peut activer ou désactiver un relais en fonction de la température ou du pourcentage de gaz détecté.

```
45 void loop() {
46     moistureReading = analogRead(moisturePin);
47     float hum = dht.readHumidity();
48     float temp = dht.readTemperature();
49     int moisturePercentage = map(moistureReading, 0, 4095, 100, 0);
50     int gassensorAnalog = analogRead(Gas_analog);
51     int gazpourcentage = map(gassensorAnalog, 0, 4095, 0, 100);
52     int level = map(value, 0, 1024, 0, 100);
53
54     Serial.print("Temperature:");
55     Serial.print(temp);
56     Serial.print((char)223);
57     Serial.print("C");
58
59     Serial.print(" Humidity:");
60     Serial.print(hum);
61     Serial.print("%");
```

```

63 Serial.print(" Soil Moisture is = ");
64 Serial.print(moisturePercentage);
65 Serial.println("%");
66
67 Serial.print(" Gas Sensor: ");
68 Serial.print(gazpourcentage);
69 Serial.print("%");
70
71 digitalWrite(POWER_PIN, HIGH);
72 delay(10);
73 value = analogRead(SIGNAL_PIN);
74 digitalWrite(POWER_PIN, LOW);
75 Serial.print("water sensor : ");
76 Serial.println(level);

```

```

78 lcd.clear();
79 lcd.print("T:" + String(temp) + (char)223 + "C" + "|" + "H:" + String(hum) + "%");
80 lcd.setCursor(0, 1);
81 lcd.print("S:" + String(moisturePercentage) + "%" + "|" + "C:" + String(gazpourcentage) + "%" + "|" + "W:" + String(level));
82
83 delay(2000);
84

```

```

85 if ((temp > temperatureThreshold) || (gazpourcentage > gazThreshold)) {
86     digitalWrite(relay1, LOW);
87 } else {
88     digitalWrite(relay1, HIGH);
89 }
90
91
92 if (moisturePercentage < humsoldesired) {
93     digitalWrite(relay2, LOW);
94 }
95 else {
96     digitalWrite(relay2, HIGH);
97 }
98 }

```

Figure 30: programme principale de la serre

Dans l'Arduino IDE, la configuration du port COM (Communication) est nécessaire pour identifier le port sur lequel votre carte Arduino est connectée.

- Sélectionner le Type de Carte :

- Allez dans le menu "Outils" (Tools).

- Sélectionner le Type de Carte :

- Sous le menu "Outils" (Tools), sélectionnez le type de carte que vous utilisez (par exemple, Arduino Uno, ESP32, etc.).

- Sélectionner le Port COM :

- Sous le même menu "Outils" (Tools), recherchez l'option "Port" (Port) ou "Port Série" (Serial Port).
- Vous verrez une liste des ports disponibles, et votre port série (COM3 dans votre cas) devrait y être répertorié.

- Sélectionner le Port COM Correct :

- Choisissez le port COM auquel votre carte Arduino est connectée. Dans votre cas, choisissez le port COM3.

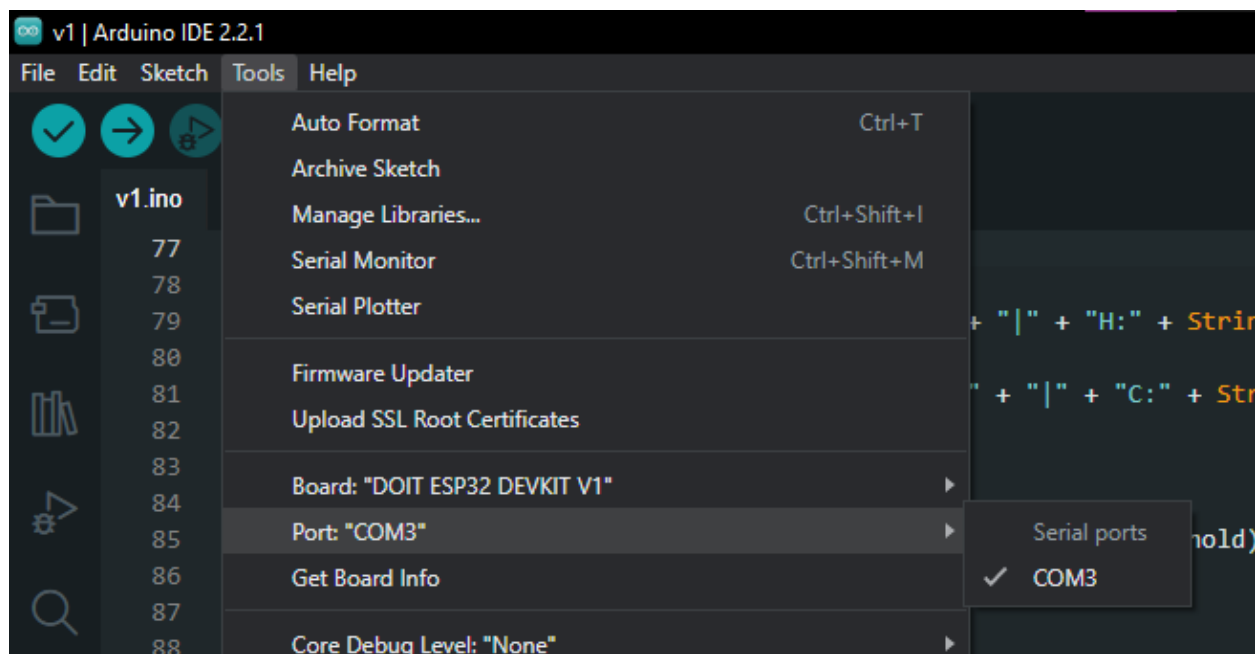
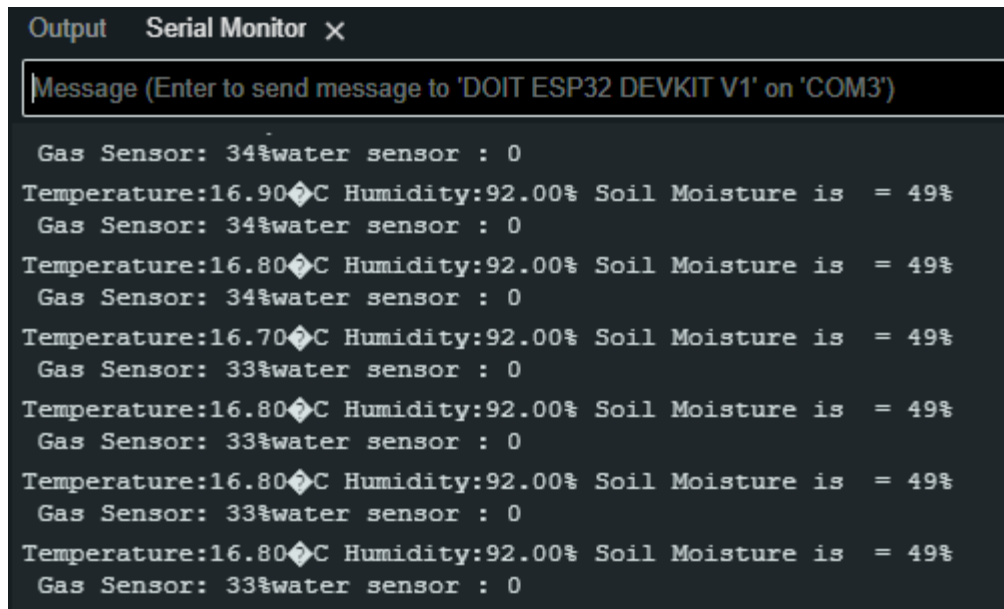


Figure 31: ajout de COM3



The screenshot shows a Serial Monitor window with the title 'Output Serial Monitor X'. Below the title bar is a text input field containing 'Message (Enter to send message to 'DOIT ESP32 DEVKIT V1' on 'COM3')'. The main area of the window displays a series of sensor readings in a monospaced font. The readings are as follows:

```

Gas Sensor: 34%water sensor : 0
Temperature:16.90°C Humidity:92.00% Soil Moisture is = 49%
Gas Sensor: 34%water sensor : 0
Temperature:16.80°C Humidity:92.00% Soil Moisture is = 49%
Gas Sensor: 34%water sensor : 0
Temperature:16.70°C Humidity:92.00% Soil Moisture is = 49%
Gas Sensor: 33%water sensor : 0
Temperature:16.80°C Humidity:92.00% Soil Moisture is = 49%
Gas Sensor: 33%water sensor : 0
Temperature:16.80°C Humidity:92.00% Soil Moisture is = 49%
Gas Sensor: 33%water sensor : 0
Temperature:16.80°C Humidity:92.00% Soil Moisture is = 49%
Gas Sensor: 33%water sensor : 0

```

Figure 32: affichage du port COM3

2-2 Plateforme : Proteus ISIS :

La Proteus Design Suite est une suite de logiciels utilisée dans le domaine de la conception électronique. Elle est développée par Labcenter Electronics. La suite comprend plusieurs outils, dont ISIS (Intelligent Schematic Input System) et ARES (Advanced Routing and Editing Software).

- ISIS (Intelligent Schematic Input System):

- ISIS est l'outil de simulation de schémas électroniques de Proteus. Il permet aux ingénieurs et aux concepteurs de créer des schémas électroniques en utilisant une interface graphique conviviale.
- Le logiciel propose une vaste bibliothèque de composants électroniques, ce qui facilite l'assemblage virtuel de circuits complexes.
- Une fois le schéma électronique créé, ISIS permet de simuler le comportement du circuit, ce qui permet de détecter d'éventuels problèmes avant la fabrication physique.

- ARES (Advanced Routing and Editing Software):

- ARES est l'outil de conception de circuits imprimés (PCB) de Proteus. Une fois que le schéma électronique est simulé et validé avec ISIS, il peut être transféré vers ARES pour la conception du PCB.
- ARES propose des fonctionnalités avancées de routage et d'édition pour créer un agencement optimal des composants sur la carte de circuit imprimé.

- Simulations:

- Proteus permet la simulation de circuits analogiques et numériques, ainsi que la simulation mixte.
- Il prend en charge la simulation de microcontrôleurs, ce qui est particulièrement utile pour les projets embarquant des systèmes microcontrôlés.

- Bibliothèque étendue:

- Proteus inclut une vaste bibliothèque de composants électroniques, de microcontrôleurs et de périphériques, facilitant ainsi la conception de circuits variés.

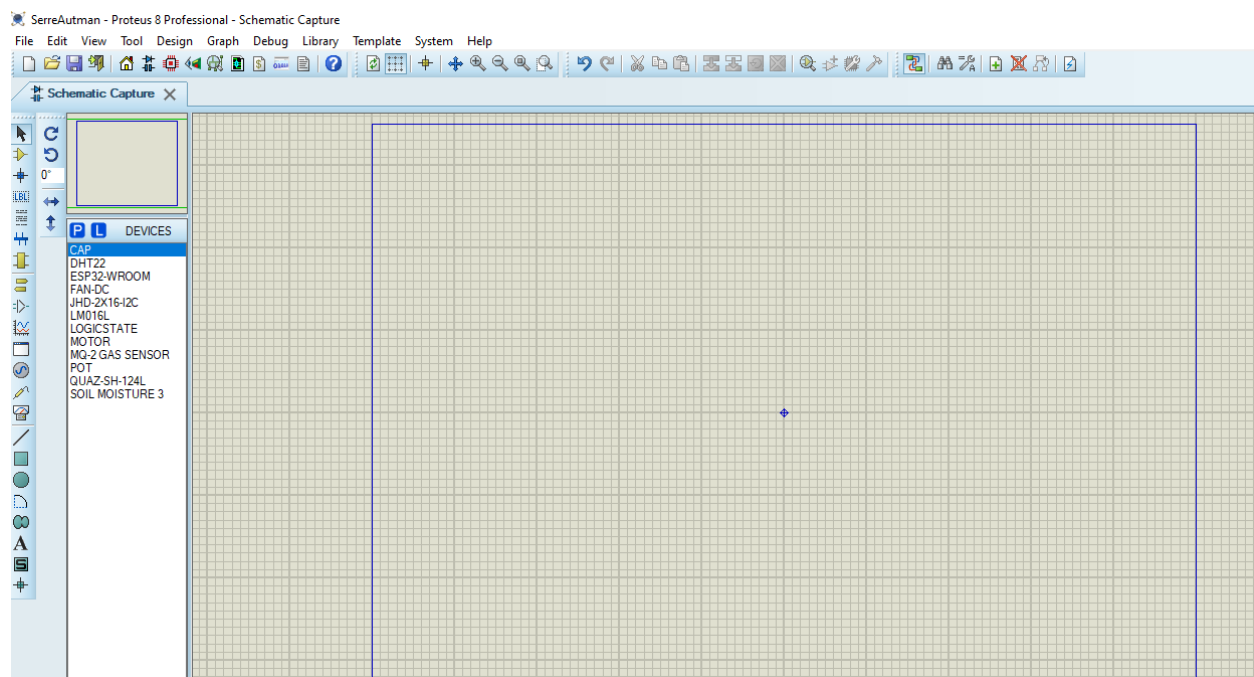
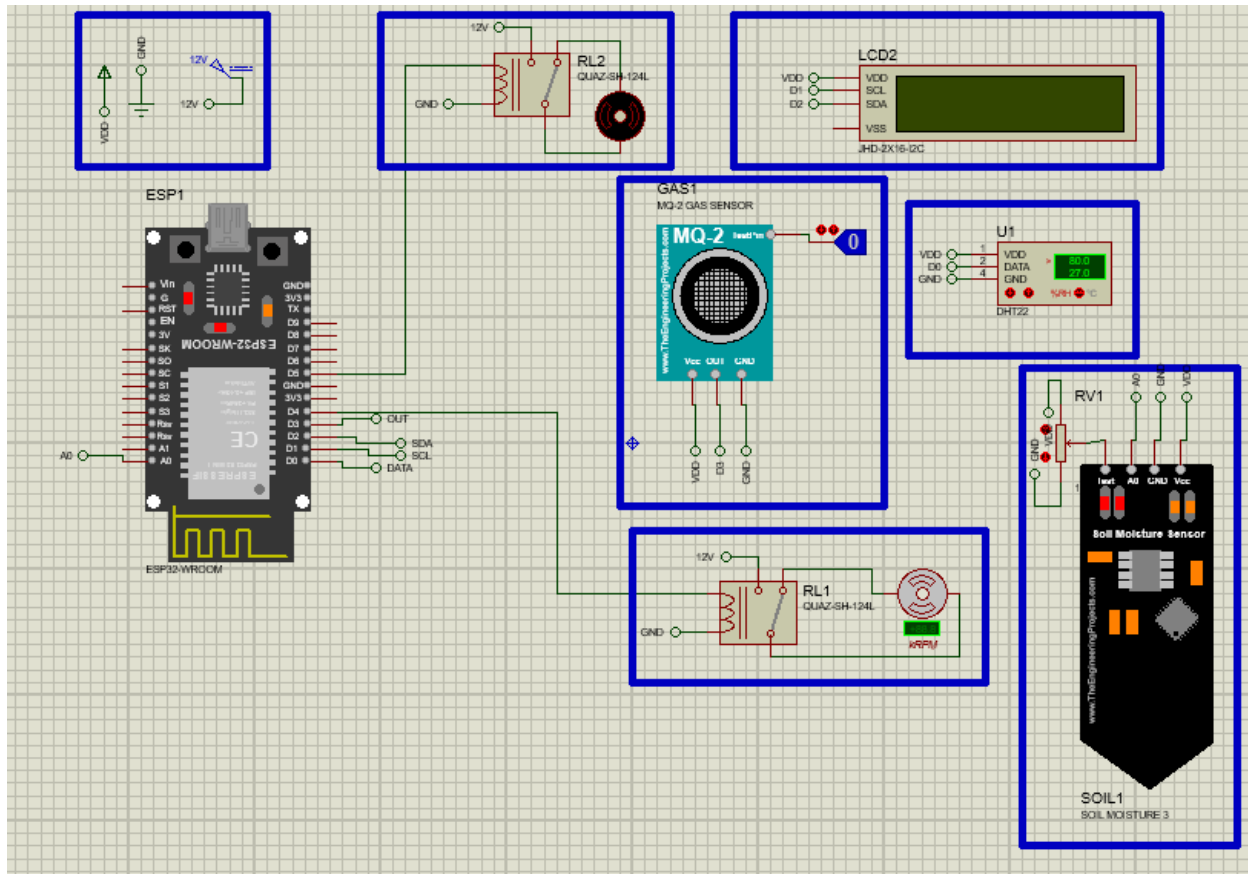


Figure 33: Interface ISIS



Surveillance en Temps Réel :

Les utilisateurs peuvent surveiller en temps réel les données générées par leurs capteurs et dispositifs connectés. Des widgets dynamiques, tels que des graphiques et des jauges, facilitent la visualisation des changements.

Historique des Données Accessible :

Adafruit IO conserve un historique des données, permettant aux utilisateurs d'analyser les tendances au fil du temps. Cela offre une perspective précieuse sur les performances passées de leurs projets.

Gestion des Notifications et des Alertes :

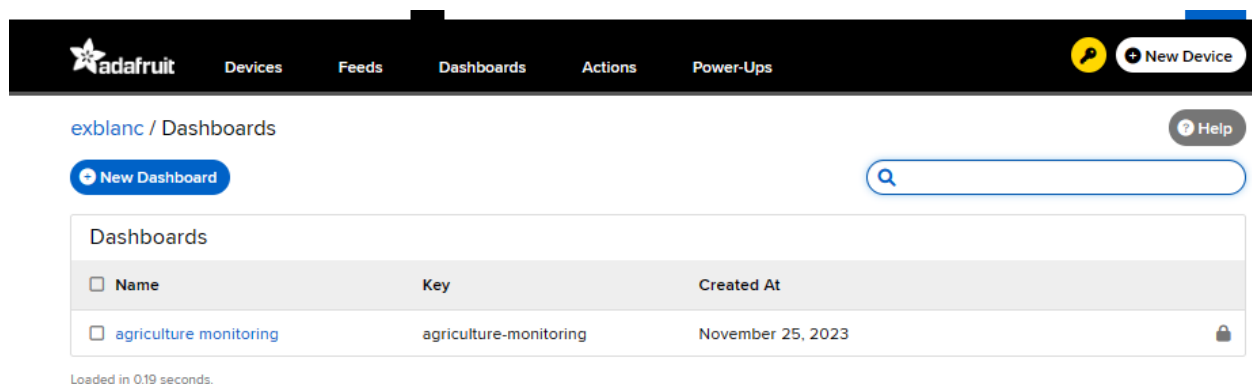
La plateforme permet la configuration d'alertes et de notifications basées sur des seuils prédéfinis. Cela offre aux utilisateurs la possibilité d'être informés rapidement de changements importants dans leurs données.

Intégration Simple avec d'Autres Services :

Adafruit IO propose des API qui facilitent l'intégration avec d'autres services et applications. Cela permet aux utilisateurs de créer des solutions complètes en combinant les fonctionnalités d'Adafruit IO avec d'autres outils.

Soutien Actif de la Communauté :

La plateforme encourage la collaboration au sein d'une communauté active. Les utilisateurs peuvent partager leurs projets, poser des questions et trouver des solutions aux défis qu'ils rencontrent.



adafruit Devices Feeds Dashboards Actions Power-Ups New Device

exblanc / Dashboards Help

New Dashboard

Name	Key	Created At
agriculture monitoring	agriculture-monitoring	November 25, 2023

Loaded in 0.19 seconds.

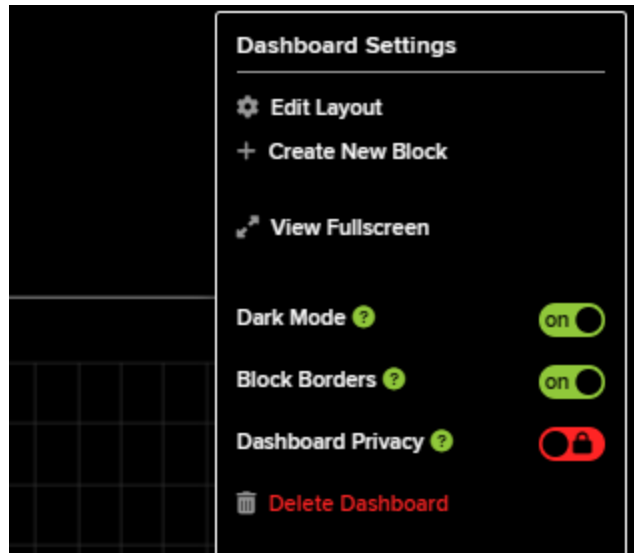
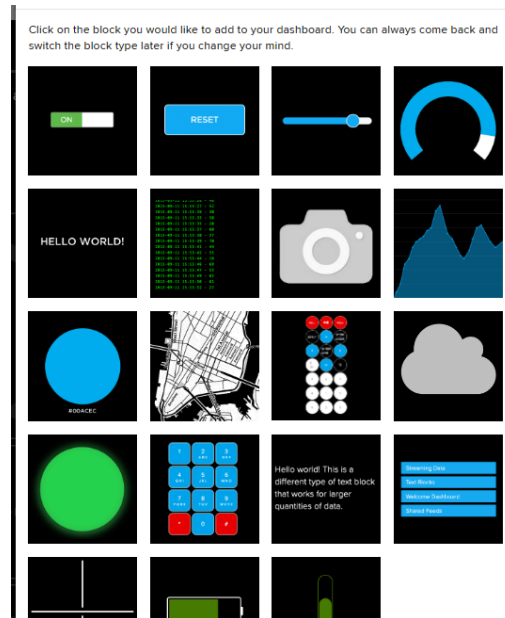


Figure 35: étapes nécessaire adafruit io



Finalement :

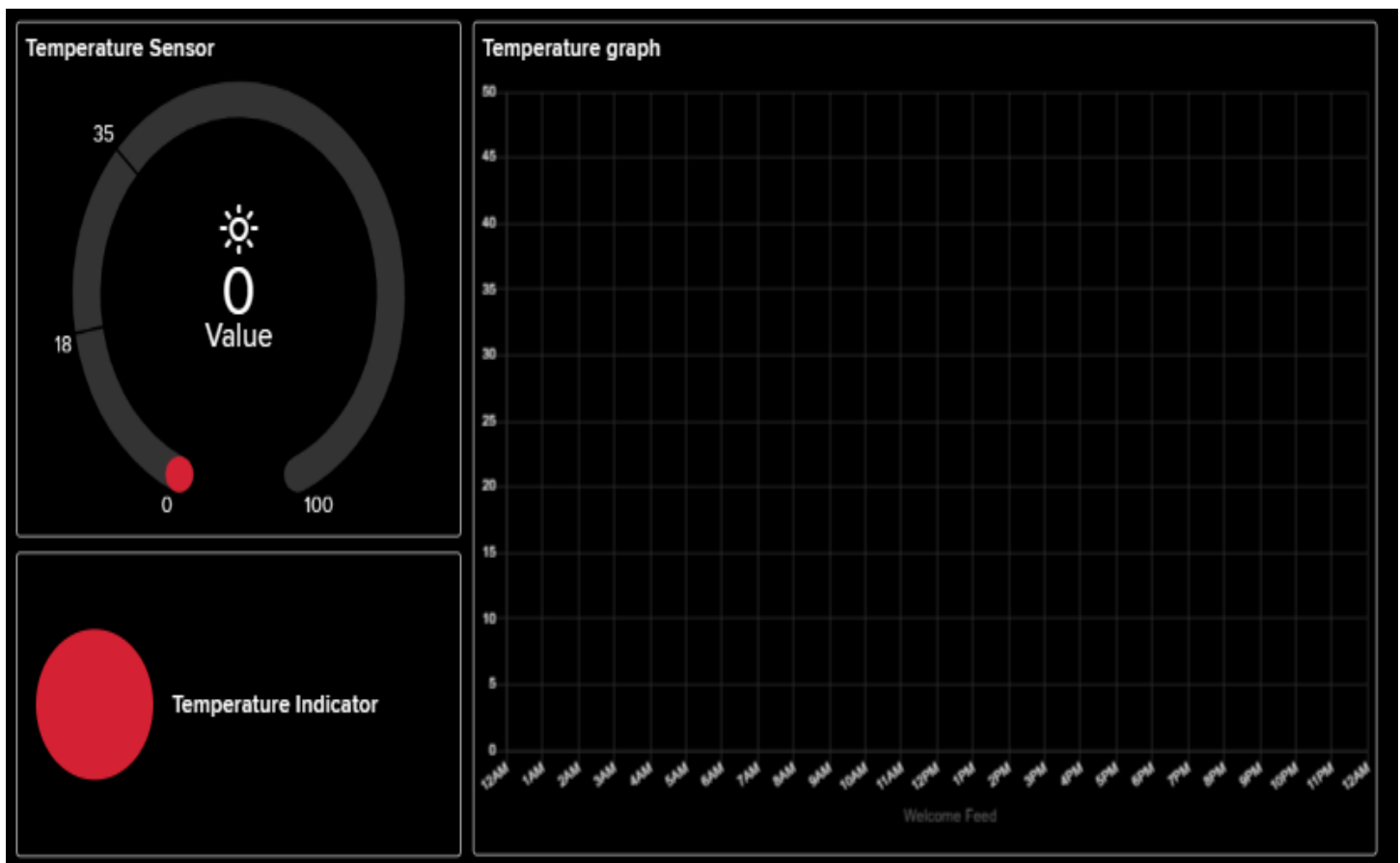


Figure 36: Dashboard de la temperature

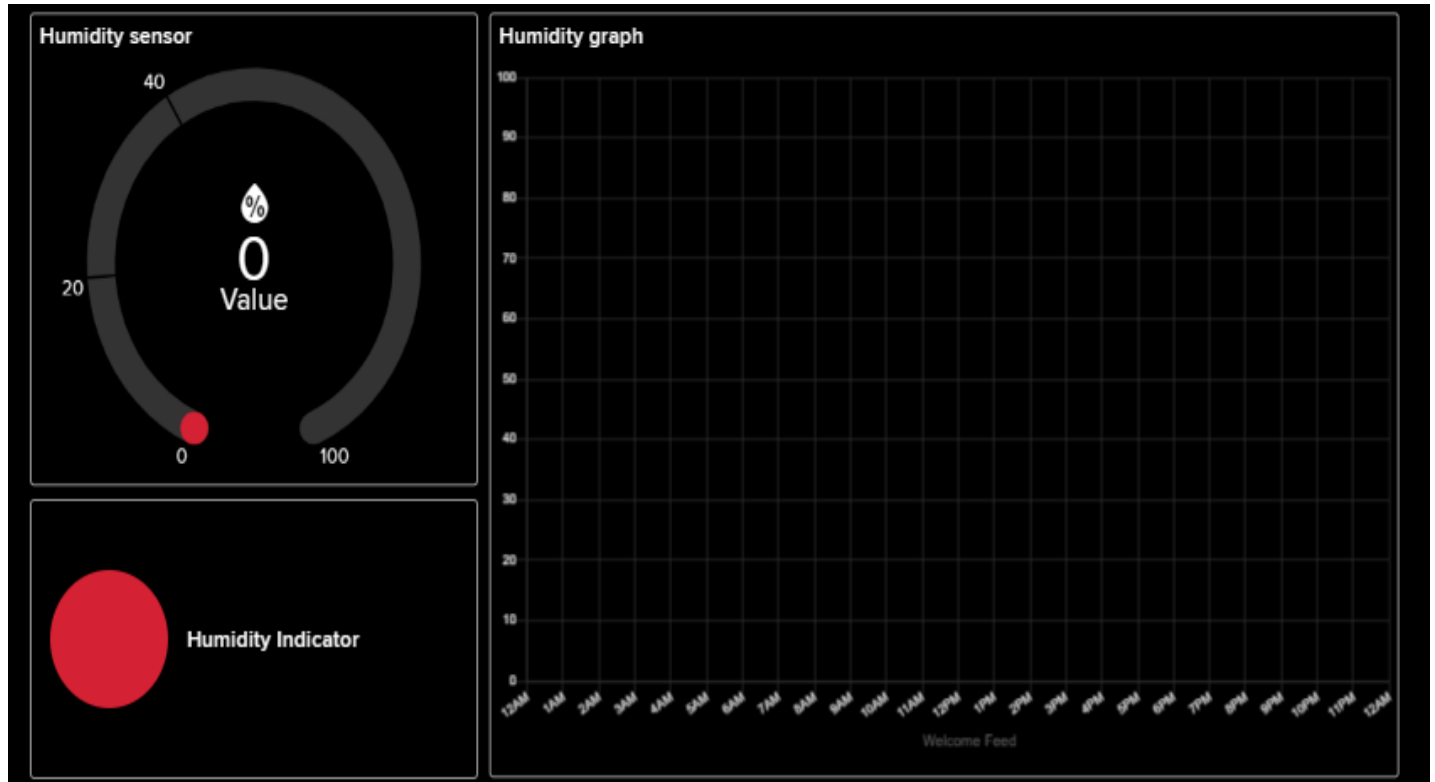


Figure 37: Dashboard Humidity

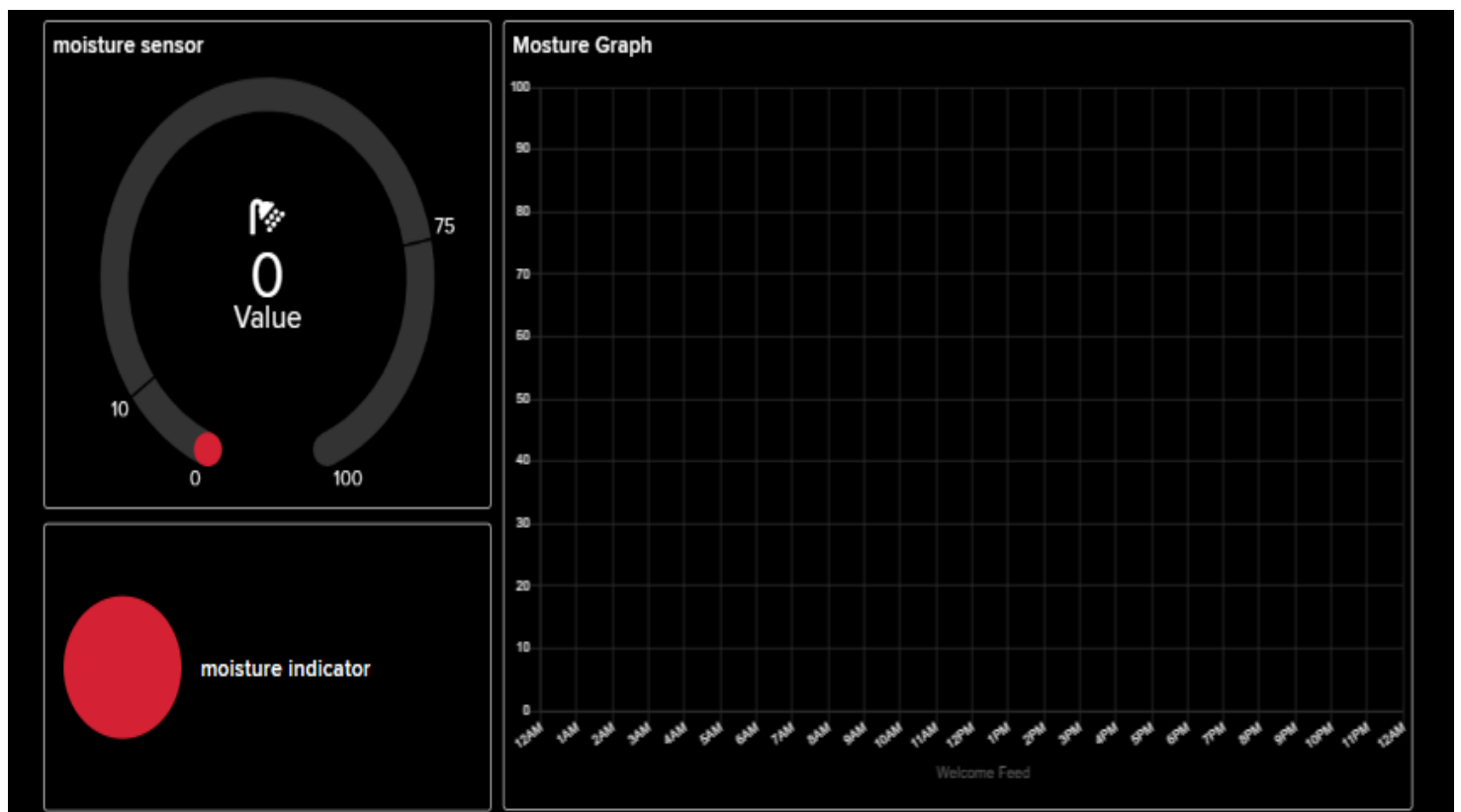


Figure 38:Dashboard sol sensor



Figure 39: Dashboard indicators

2-3.2 Partie programmation d'interface Adafruit io :

Tout d'abord il faut installer les bibliothèques nécessaires pour faire un code parfait alors on a besoin de ces bibliothèques :

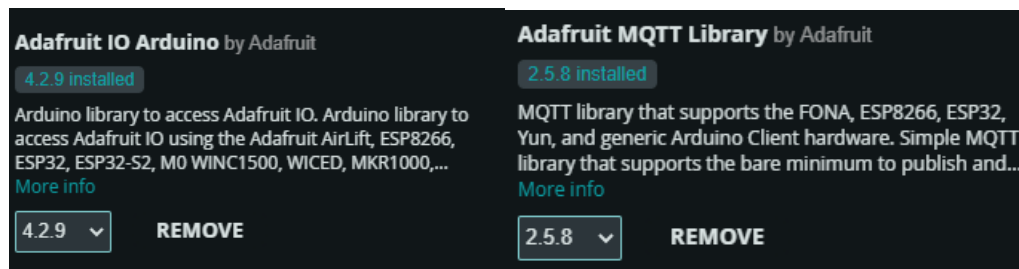
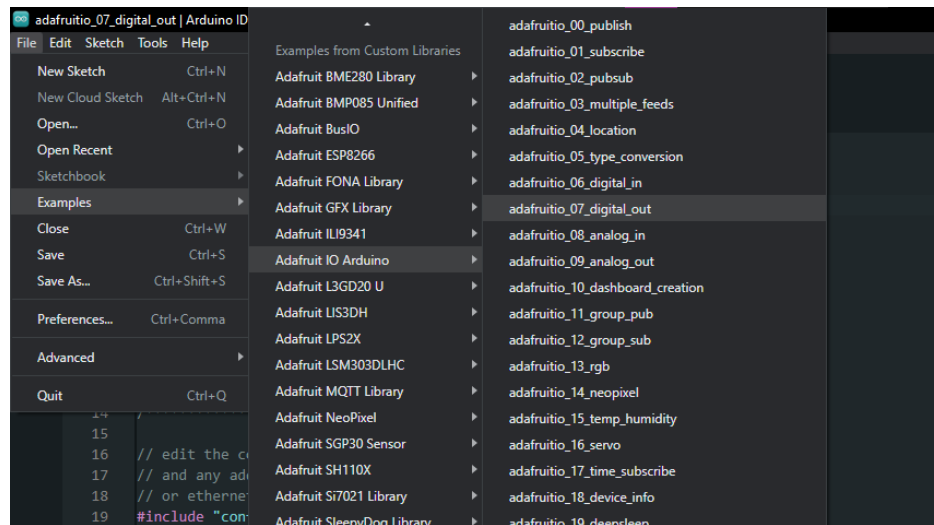


Figure 40: bibliotheques necessaires pour Adafruit io

Après nous avons obtenu le code à reformuler à partir du schema suivant :



Examples >> Adafruit IO >> Adafruitio_07_digital_out

```
// Adafruit IO Digital Output Example
// Tutorial Link: https://learn.adafruit.com/adafruit-io-basics-digital-output
//
// Adafruit invests time and resources providing this open source code.
// Please support Adafruit and open source hardware by purchasing
// products from Adafruit!
//
// Written by Todd Treece for Adafruit Industries
// Copyright (c) 2016 Adafruit Industries
// Licensed under the MIT license.
//
// All text above must be included in any redistribution.

/***** Configuration *****/

// edit the config.h tab and enter your Adafruit IO credentials
// and any additional configuration needed for WiFi, cellular,
// or ethernet clients.
#include "config.h"

/***** Example Starts Here *****/

// digital pin 5
#define LED_PIN 5

// set up the 'digital' feed
AdafruitIO_Feed *digital = io.feed("digital");

void setup() {
```

```
pinMode(LED_PIN, OUTPUT);

// start the serial connection
Serial.begin(115200);

// wait for serial monitor to open
while(! Serial);

// connect to io.adafruit.com
Serial.print("Connecting to Adafruit IO");
io.connect();

// set up a message handler for the 'digital' feed.
// the handleMessage function (defined below)
// will be called whenever a message is
// received from adafruit io.
digital->onMessage(handleMessage);

// wait for a connection
while(io.status() < AIO_CONNECTED) {
    Serial.print(".");
    delay(500);
}

// we are connected
Serial.println();
Serial.println(io.statusText());
digital->get();
}

void loop() {

    // io.run(); is required for all sketches.
    // it should always be present at the top of your loop
    // function. it keeps the client connected to
    // io.adafruit.com, and processes any incoming data.
    io.run();
}

// this function is called whenever an 'digital' feed message
// is received from Adafruit IO. it was attached to
// the 'digital' feed in the setup() function above.
```

```

void handleMessage(AdafruitIO_Data *data) {

    Serial.print("received <- ");

    if(data->toPinLevel() == HIGH)
        Serial.println("HIGH");
    else
        Serial.println("LOW");

    digitalWrite(LED_PIN, data->toPinLevel());
}

```

Figure 41: code Adafruit io

Pour ce code on a besoin d'un code **config.h** qui est relié directement avec le code lorsque on a crée précédemment :

```

>> #define IO_USERNAME "yexblanc"

#define IO_KEY "aio_eVU180urRJBpQd17nvD57WyR8uPn"

/***** WIFI *****/

// the AdafruitIO_WiFi client will work with the following boards:
//   - HUZZAH ESP8266 Breakout -> https://www.adafruit.com/products/2471
//   - Feather HUZZAH ESP8266 -> https://www.adafruit.com/products/2821
//   - Feather HUZZAH ESP32 -> https://www.adafruit.com/product/3405
//   - Feather M0 WiFi -> https://www.adafruit.com/products/3010
//   - Feather WICED -> https://www.adafruit.com/products/3056
//   - Adafruit PyPortal -> https://www.adafruit.com/product/4116
//   - Adafruit Metro M4 Express AirLift Lite ->
//     https://www.adafruit.com/product/4000
//   - Adafruit AirLifts Breakout -> https://www.adafruit.com/product/4201
//   - Adafruit AirLift Shield -> https://www.adafruit.com/product/4285
//   - Adafruit AirLift FeatherWing -> https://www.adafruit.com/product/4264

#define WIFI_SSID "*****"
#define WIFI_PASS "*****"

// uncomment the following line if you are using airlift
// #define USE_AIRLIFT

// uncomment the following line if you are using winc1500
// #define USE_WINC1500

```

```
// uncomment the following line if you are using mrk1010 or nano 33 iot
// #define ARDUINO_SAMD_MKR1010

// comment out the following lines if you are using fona or ethernet
#include "AdafruitIO_WiFi.h"

#if defined(USE_AIRLIFT) || defined(ADAFRUIT_METRO_M4_AIRLIFT_LITE) || \
    defined(ADAFRUIT_PYPORTAL)
// Configure the pins used for the ESP32 connection
#if !defined(SPIWIFI_SS) // if the wifi definition isnt in the board variant
// Don't change the names of these #define's! they match the variant ones
#define SPIWIFI SPI
#define SPIWIFI_SS 10 // Chip select pin
#define NINA_ACK 9 // a.k.a BUSY or READY pin
#define NINA_RESETN 6 // Reset pin
#define NINA_GPIO0 -1 // Not connected
#endif
AdafruitIO_WiFi io(IO_USERNAME, IO_KEY, WIFI_SSID, WIFI_PASS, SPIWIFI_SS,
                  NINA_ACK, NINA_RESETN, NINA_GPIO0, &SPIWIFI);
#else
AdafruitIO_WiFi io(IO_USERNAME, IO_KEY, WIFI_SSID, WIFI_PASS);
#endif
```

Figure 42: config.h de l'Adafruit io

3- Partie construction :

3-1 Integration capteurs et actionneurs :

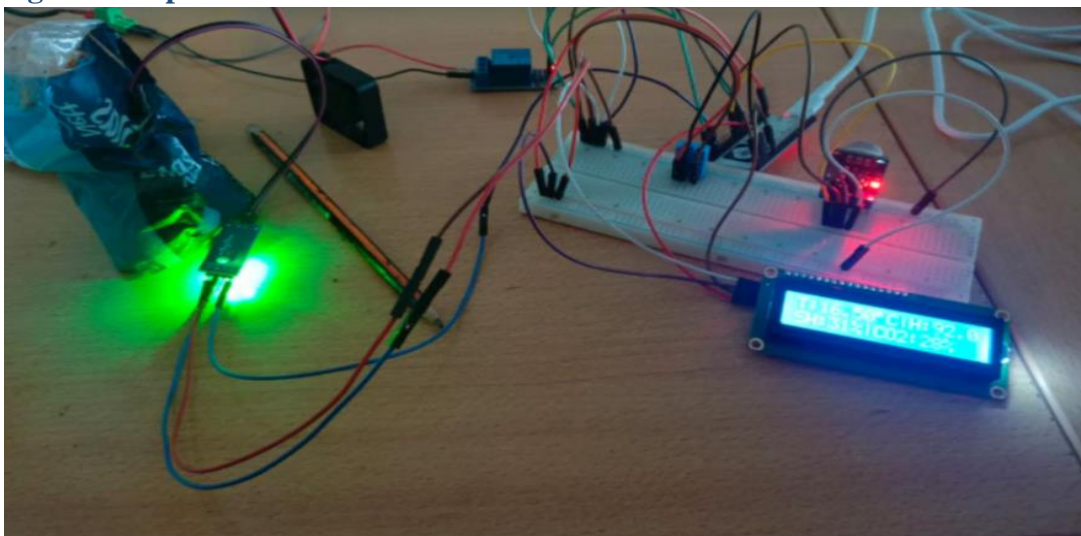


Figure 43: integration matérielles

3-2 Construction Carcasse :



Figure 44: carcasse



Figure 45: reservoir d'eau

3-3 Construction Final :



Figure 46: Etat final de notre serre

Conclusion

Notre projet vise à la conception d'une mini serre intelligente pour la production agricole. Avec sa capacité à s'autogérer, notre serre va se charger de régler les différents paramètres climatiques tels que la température, l'humidité et le gaz CO₂ à l'intérieur de la serre, et aussi une gestion plus optimale de l'arrosage. Pour ce faire, nous avons accompli notre travail à différents niveaux : Dans la première partie on a présenté l'état de l'art La deuxième partie de ce travail est consacrée à a présentation des différents matériels et logiciels nécessaires pour a réalisation de notre projet comme : Présentation et application de l'ESP32 , la simulation, la supervision, ainsi les différents programmes. La troisième partie de cette étude est réservée à la réalisation du notre serre intelligente. Dans cette partie, nous avons présenté les différentes étapes de la réalisation de notre système, les branchements des différents capteurs, les actionneurs, le déroulement des différentes étapes du programme .