جامعة سيدي محمد بن عبدالله بفاس
ⵜⴰⵙⴷⴰⵡⵉⵜ ⵙⵉⴷⵉ ⵎⵓⵃⴰⵎⵎⴷ ⴱⵏ ⵄⴱⴷⵍⵍⴰⵀ ⴷⴼⴰⵙ
UNIVERSITÉ SIDI MOHAMED BEN ABDELLAH DE FES

المدرسة العليا للتكنولوجيا
ⵜⵉⵎⵙⵙⵓⵔⵉⵏ ⵜⵉⵏⵏⴰⵡⵉⵜ ⵏ ⵜⵉⴽⵏⵓⵍⵓⵊⵉⵜ
ECOLE SUPÉRIEURE DE TECHNOLOGIE

RAPPORT DE PROJET DE FIN D'ETUDES

Pour l'obtention du Diplôme Universitaire de Technologie

Département Génie Électrique & Systèmes Intelligents

Filière : Systèmes Embarqués

Sous le thème :

RELISATION D'UN SUIVEUR SOLAIRE COMMANDE PAR IOT

Réalisé par :

- *EDDOUCHE Yasser*
- *HARRAOUI Sohaib*

Encadré par :

- Mme EL HAMMOUMI
- M. CHAIBI

Année Universitaire : 2022 – 2023

Dédicace

Nous tenons à dédier ce rapport à nos familles respectives, qui ont été une source de soutien inconditionnel tout au long de notre parcours universitaire. Votre amour, votre confiance et votre patience nous ont permis de persévérer et de poursuivre nos rêves malgré les obstacles.

À nos encadrants pédagogiques pour leur expertise, leur soutien et leur accompagnement tout au long de ce projet en système embarqué. Vos précieux conseils et votre expertise ont été déterminants pour la réussite de ce projet.

À nos amis et nos proches pour leur soutien constant, leur amitié et leur encouragement. Votre présence à nos côtés a été une source de motivation et d'inspiration pour nous tout au long de ce projet.

À toutes les personnes qui ont contribué de près ou de loin à la réalisation de ce rapport et de notre PFE. Leur collaboration et leur soutien ont été essentiels pour que nous puissions atteindre nos objectifs et réaliser ce projet avec succès.

Remerciements

Nous tenons à remercier Madame EL HAMMOUMI et M. Rédouane CHAIBI pour leur formation bien encadrée et leurs conseils fructueux dans la réalisation de notre projet de fin d'études. Nous les remercions également pour les expériences riches et intéressantes qu'ils nous ont fait vivre durant leur apprentissage de deux ans.

Grand merci pour M. Hamza AZIZI pour son aide, vaste patience ainsi que pour son bienveillance. Nous avons eu la chance d'avoir notre premier projet dans des conditions très favorables. C'est grâce à nos professeurs compétents et qualifiés et à nos installations bien équipées.

Nous profitons de l'occasion pour exprimer notre profonde gratitude aux professeurs de l'Ecole Supérieur de Technologie pour nous avoir fait partager leurs connaissances et prodiguer un enseignement de qualité. Nous remercions M. Zakaria MOUTAKKI pour sa générosité, sa disponibilité et ses encouragements. Un merci spécial à ma famille pour ses conseils et son soutien inconditionnel.

Nous tenons à exprimer notre gratitude envers nos amis et nos proches pour leur soutien constant, leur amitié et leur encouragement. Votre présence à nos côtés a été une source de motivation et d'inspiration pour nous tout au long de ce projet.

Nous tenons également à remercier le jury d'avoir accepté d'évaluer ce projet.

Table des matières

Liste des figures

Introduction

Le soleil est une source d'énergie renouvelable primordiale pour la production d'électricité à l'aide de la technologie photovoltaïque ou de concentrateurs solaires.

Cependant, pour maximiser l'efficacité des panneaux solaires, il est essentiel de les orienter correctement vers le soleil tout au long de la journée. C'est là que les systèmes de suivi solaire entrent en jeu, permettant d'optimiser la production d'énergie solaire en temps réel.

Dans ce contexte, ce projet de fin d'études (PFE) vise à concevoir et à réaliser un suiveur solaire commandé par l'Internet des objets (IoT) qui est constitué de 2 axes et qui suivre automatiquement le soleil à l'aide des capteurs LDR, ou manuellement par l'utilisateur via le tableau de bord d'une application IOT.

Notre mémoire de projet est divisé en trois chapitres distincts :

Le premier chapitre présente des informations générales sur l'énergie et les suiveurs solaires.

Le deuxième chapitre se concentre sur une étude approfondie du suiveur solaire en expliquant ses composants et son principe de fonctionnement.

Enfin, le troisième chapitre détaille la procédure utilisée pour réaliser un suiveur solaire en utilisant une carte Arduino, intégrant l'IoT pour la commande manuelle et la logique floue pour une autre méthode de commande automatique.

Une conclusion générale clôture notre mémoire de projet.

Chapitre 1

Généralités sur l'énergie et les suiveurs solaires

1. L'énergie solaire :

L'énergie solaire peut être convertie en électricité grâce à des panneaux solaires ou des centrales thermiques qui capturent les rayons du soleil et les convertissent en électricité. Le principe de base de la conversion est de transformer l'énergie portée par les photons de la lumière en électricité. Cette conversion est réalisée par une cellule photovoltaïque qui absorbe l'énergie des photons lumineux pour générer un courant électrique continu. Ce courant continu sera converti en courant alternatif à l'aide d'un onduleur.

L'électricité produite par l'énergie solaire peut être utilisée pour alimentation des maisons, des équipements agricoles, équipements de loisirs, équipements de communication.

2. Les panneaux solaires photovoltaïques :

Les panneaux solaires photovoltaïques, sont des dispositifs qui convertissent la lumière du soleil en électricité à l'aide de cellules photovoltaïques, généralement avec 36 cellules par module. Chaque cellule peut produire entre 1 et 2 watts de puissance électrique sous une irradiation solaire standard (1000 watts par mètre carré) et il est constitué généralement constituer de couches de silicium dopées (positivement et négativement) qui génèrent un courant électrique lorsqu'elles sont exposées à la lumière du soleil.

Les panneaux solaires PV sont utilisés pour produire de l'électricité à partir de sources d'énergie renouvelable, ce qui en fait une alternative propre et durable aux sources d'énergie fossiles telles que le pétrole, le gaz naturel et le charbon. Les panneaux solaires PV sont souvent

utilisés dans les systèmes solaires pour les maisons, les entreprises, les fermes et les projets gouvernementaux, et peuvent être installés sur les toits, les murs et même sur le sol..

3. Les suiveurs solaires :

3.1. Les principes de base des suiveurs solaires :

Un suiveur solaire est un dispositif utilisé pour orienter un panneau photovoltaïque vers le soleil, ce qui améliore considérablement son efficacité, surtout pendant les heures du matin et de l'après-midi lorsque les panneaux fixes sont inclinés dans la mauvaise direction. Le coût initial d'un système de suivi solaire dépend du type et de la taille du suiveur utilisé. Cependant, un système bien conçu qui utilise un suiveur nécessitera moins de panneaux pour atteindre un rendement optimal, entraînant ainsi une réduction des coûts initiaux d'installation.

3.2. Les avantages des panneaux mobiles par rapport aux panneaux fixes :

Pendant la journée, le soleil se déplace continuellement dans le ciel, tandis qu'un panneau photovoltaïque reste immobile et perd une quantité importante d'énergie qui pourrait être utilisée. Dans une installation fixe, la production d'énergie des panneaux PV est maximale seulement à midi lorsque les panneaux sont perpendiculaires aux rayons du soleil. Toutefois, si les panneaux sont orientés en permanence vers le soleil, cela équivaut à une position perpendiculaire constante, ce qui permet de maximiser la production d'énergie tout au long de la journée. Les panneaux photovoltaïques équipés de suiveurs solaires ont un rendement énergétique nettement supérieur à celui des installations fixes. Par exemple, sur une journée ensoleillée, un système fixe de 1 kW bien orienté produit 5,5 kWh d'énergie, alors que le même

système équipé d'un suiveur solaire produit 11 kWh d'énergie dans les mêmes conditions d'ensoleillement.

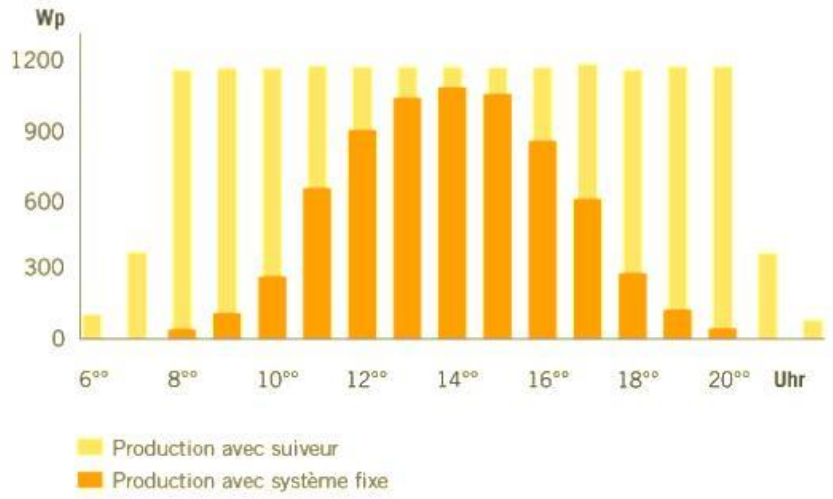


Figure 1 : Diagramme de comparaison entre la production avec suiveur et la production avec système fixe.

3.3. Les types de suiveurs solaires :

On distingue principalement deux grandes familles de suiveurs solaires : les passifs et les actifs qui comportent les suiveurs mono-axe et double axe.

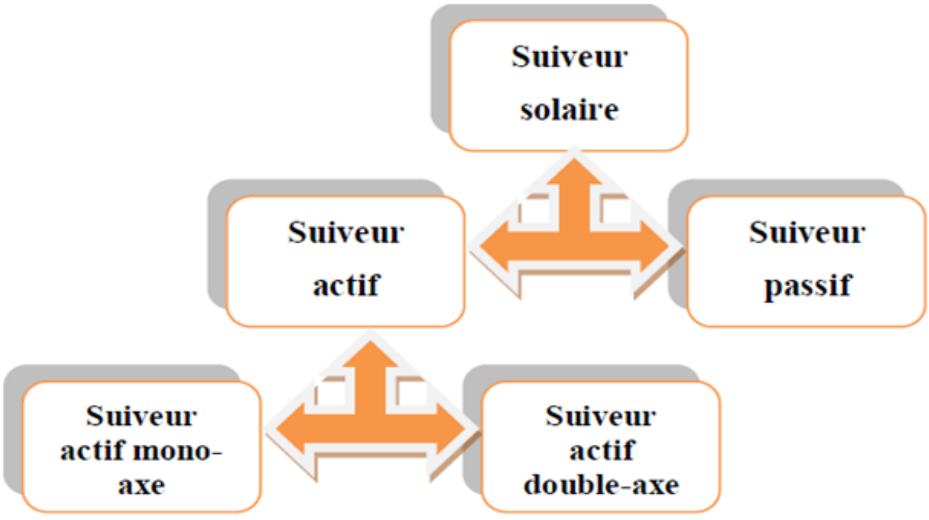


Figure 2 : Schéma des types de suiveurs solaires

3.3.1. Suiveurs solaire passif :

Un suiveur solaire passif se compose de deux tubes en cuivre fixés aux côtés est et ouest du panneau solaire. Les tubes contiennent des fluides chimiques qui ont la capacité de se vaporiser à basse température. Lorsque le rayonnement solaire augmente, la température d'un côté du panneau solaire augmente, ce qui provoque la vaporisation du composé dans le tube en cuivre. La partie gazeuse du composé occupe alors un volume interne plus important, tandis que sa partie liquide se déplace vers le côté ombragé. Ce transfert de masse rééquilibre le panneau solaire en le faisant tourner vers la source des rayons solaires. Ce type de suiveur solaire ne nécessite pas d'énergie supplémentaire pour repositionner le panneau solaire.

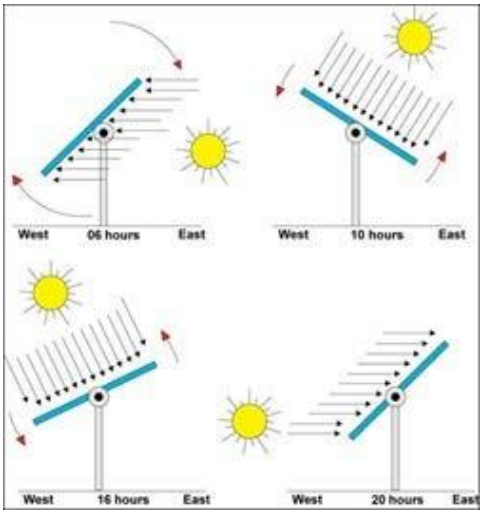


Figure 3 : Suiveur solaire passif

3.3.2. Suiveurs solaire actif :

Les suiveurs solaires actifs sont conçus pour détecter la lumière et suivre la trajectoire du soleil afin d'optimiser l'angle d'incidence du rayonnement solaire sur leur surface.

Il y a deux types de suiveurs actifs : les suiveurs mono-axe et les suiveurs double axe. Comparés aux suiveurs passifs, les suiveurs actifs ont l'avantage d'être plus précis dans leur suivi et ne nécessitent pas d'intervention manuelle pour les ajuster

3.3.2.1. Suiveur solaire mono-axe :

Le suiveur solaire mono-axial doit son nom à sa rotation autour d'un seul axe, souvent positionné en azimut, soit d'Est en Ouest, tout au long de la journée. Le panneau photovoltaïque est fixé selon un angle optimal pour recevoir le maximum de rayons solaires tout au long de l'année, qui est déterminé par la latitude du lieu où est situé le capteur. Bien que l'angle d'inclinaison reste constant, l'azimut varie en fonction du mouvement du soleil. Ce système de suivi est le plus couramment utilisé en raison de sa simplicité.

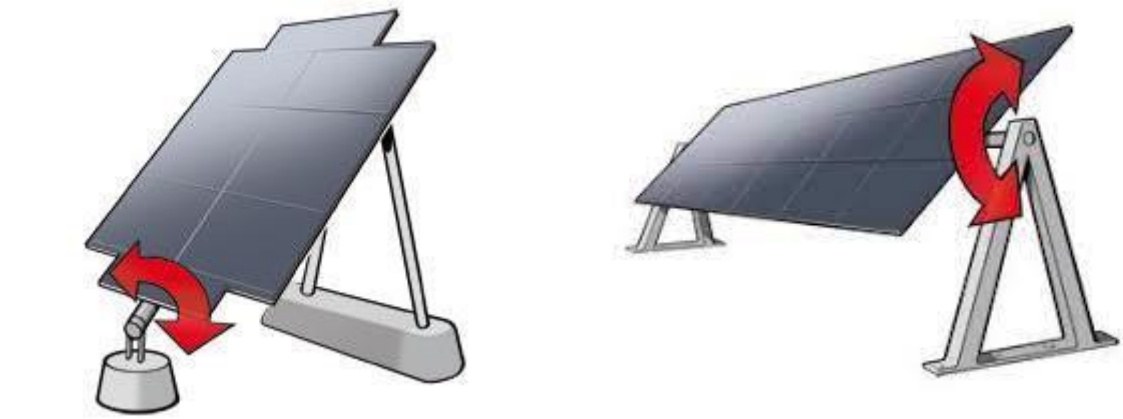


Figure 4 : Panneaux solaires avec système de poursuite mono-axiale.

3.3.2.2. Suiveur solaire double axe :

Ce suiveur solaire double-axe est équipé de deux axes qui lui permettent de suivre les mouvements en azimut et en inclinaison du soleil tout au long de la journée, maintenant ainsi le capteur constamment orienté vers la source de lumière. Contrairement au suiveur mono-axe, ce type de suiveur est plus imposant mais offre de meilleures performances.

Figure 5 : Panneau solaire avec système de poursuite biaxiale.

3.4. Les avantages et les inconvénients associés à l'utilisation de suiveurs solaires :

3.4.1. Les avantages :

- Une efficacité accrue : un suiveur solaire peut suivre la trajectoire du soleil tout au long de la journée, augmentant ainsi la quantité d'énergie solaire captée et produisant jusqu'à 40% d'énergie supplémentaire par rapport à un panneau solaire fixe.

- Une utilisation optimale de l'espace : un suiveur solaire peut être utilisé avec un nombre réduit de panneaux solaires pour produire la même quantité d'énergie que plusieurs panneaux fixes, ce qui permet une utilisation optimale de l'espace.

- Une maintenance réduite : un suiveur solaire peut améliorer la durée de vie des panneaux solaires en réduisant l'accumulation de poussière, qui peut diminuer leur efficacité.

- Une réduction de l'empreinte carbone : en augmentant la production d'énergie solaire, un suiveur solaire peut réduire l'utilisation d'énergie provenant de sources non renouvelables, réduisant ainsi l'empreinte carbone de l'utilisateur.

- Une flexibilité de placement : en raison de leur capacité à suivre le soleil, les suiveurs solaires peuvent être placés dans des zones ombragées ou difficiles d'accès, où les panneaux solaires fixes ne seraient pas efficaces.

- Une valeur ajoutée : les suiveurs solaires peuvent ajouter de la valeur à une propriété en améliorant l'efficacité énergétique et en réduisant les coûts énergétiques à long terme.

3.4.2. Les inconvénients :

- Maintenance plus complexe : le suiveur solaire nécessite des opérations de maintenance régulières pour assurer son bon fonctionnement, ce qui peut entraîner des coûts supplémentaires et une perte de temps.

- Plus grande vulnérabilité aux intempéries : en raison de la complexité du système, un suiveur solaire est plus vulnérable aux intempéries telles que les tempêtes de vent, la neige, la glace et la grêle, ce qui peut entraîner des dommages et des coûts de réparation supplémentaires.

- Consommation d'énergie : les suiveurs solaires actifs nécessitent de l'énergie pour leur fonctionnement, ce qui peut entraîner une augmentation de la consommation d'énergie globale de l'installation.

Chapitre 2

Etude du suiveur solaire

1. Description du systeme :

Le système de suiveur solaire proposé est une solution innovante qui permet de maximiser la production d'énergie solaire. Ce suiveur solaire est basé sur l'IoT (Internet des objets) et peut être utilisé de manière automatique ou manuelle pour ajuster la position du panneau photovoltaïque en fonction de la position du soleil.

Le système utilise des capteurs LDR pour détecter l'intensité lumineuse et ainsi déterminer la position du soleil. Les données collectées sont ensuite envoyées à un contrôleur Arduino Méga qui les traite pour commander les servomoteurs qui permettent au panneau de se tourner vers le soleil. Les capteurs associés aux panneaux PV collectent également les données relatives à la tension et au courant générés par le panneau, qui sont envoyés à l'Arduino via le Shield Ethernet. Les données collectées par le système sont envoyées à un serveur Web pour être stockées dans une base de données Firebase. Les informations relatives à la performance du panneau solaire, y compris les capteurs LDR, l'erreur horizontale et verticale et la position des servomoteurs, sont affichées en temps réel sur le tableau de bord du serveur Web. En mode manuel, les servomoteurs peuvent être ajustés manuellement à l'aide des curseurs associés dans l'application IoT, ce qui permet à l'utilisateur de rechercher le meilleur angle possible pour maximiser la production d'énergie solaire.

Le système compose de 2 méthodes de fonctionnement en mode automatique, la première est basée sur la logique normale et la deuxième sur la logique floue qui est considérée comme un outil de l’intelligence artificielle, car elle permet de traiter des données complexes et incertaines de manière plus efficaces que les approches classiques basées sur la logique binaire.

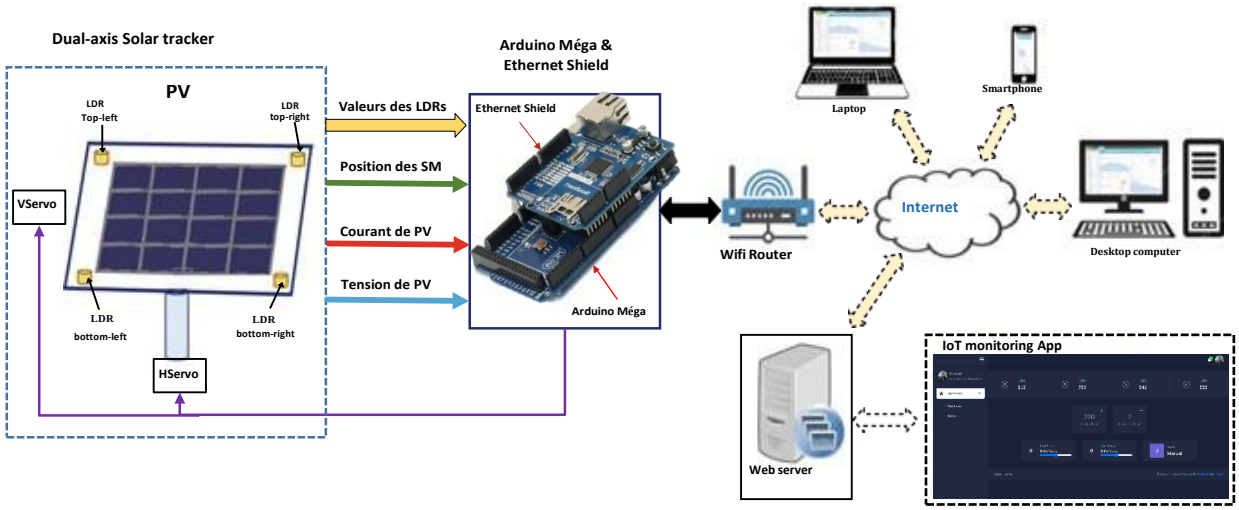


Figure 6 : Schéma fonctionnel de système

2. Composants principaux du suiveur solaire :

La carte Arduino Méga disposant d'un nombre important des E S (54 pins numériques et 16 analogiques) et des performances compatibles avec les exigences de notre projet sans augmenter le budget de réalisation.

Pour réaliser ce projet, Nous avons utilisé les composants suivants :

- Arduino Méga
- 4 Capteurs LDR
- 2 Servomoteurs
- Capteur de tension
- Capteur d'intensité de courants
- W5100 Shield Ethernet

2.1. Capteur LDR :

Ou photorésistance est un composant électronique dont la résistance varie en fonction de la lumière perçue. La valeur de la luminosité est convertie en résistance qu'il est facile d'acquérir avec l'aide d'une carte Arduino au travers d'une entrée analogique.

Les valeurs de résistance LDR sont de plusieurs méga ohms dans l'obscurité, puis tombent à quelques centaines d'ohms en lumière vive.

Figure 7 : LDR

2.2. Capteur de tension :

Ce capteur de tension basé sur diviseur par résistances, tension d'entrée est réduite 5 fois, l'entrée analogique d'Arduino maximale est 5 V, donc la tension d'entrée de la détection de tension ne peut pas être supérieure à 5 V * 5 = 25 V (si un système 3.3 V est utilisé, la tension d'entrée ne peut pas être plus grande que 3.3Vx5 = 16.5 v).

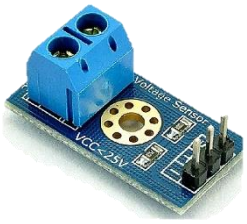


Figure 9 : Capteur de tension

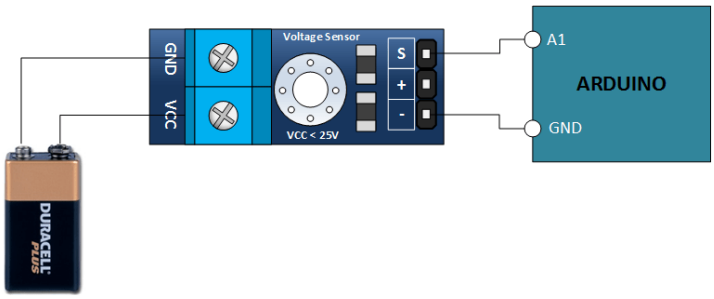


Figure 8 : Branchement de capteur de tension

2.3. Capteur d’intensité de courant :

Basé sur la puce ACS712 d'Allegro, ce capteur se branche en série avec la charge sur un circuit alternatif (AC) ou continu (DC) et permet de mesurer le courant qui traverse le capteur. Il utilise le champ magnétique généré par le courant (et donc l'effet hall) pour mesurer le courant qui le traverse. Le module propose en sortie une tension continue proportionnelle au courant à raison de 0.066V/A (66mV par ampère) et 2.5v en sortie pour 0A sous 5v.

Figure 11 : Capteur de courant

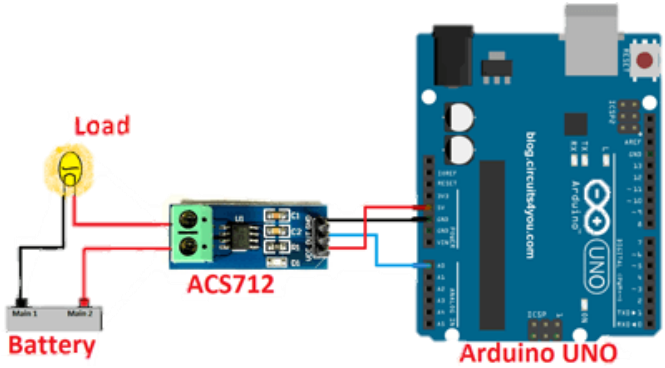


Figure 10 : Branchement de capteur de courant

2.4. Servo moteurs :

Un servomoteur est un moteur capable de maintenir une opposition à un effort statique et dont la position est vérifiée en continu et corrigée en fonction de la mesure. C'est donc un système asservi. Le servomoteur intègre dans un même boitier, la mécanique (moteur et engrenage), et l’électronique, pour la commande et l'asservissement du moteur.

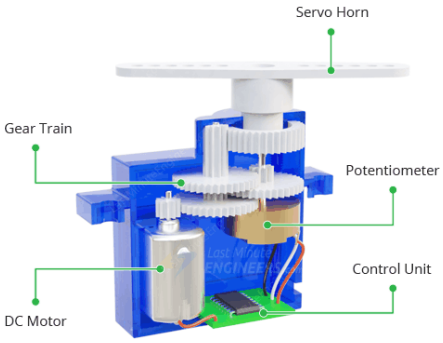


Figure 12 : Servo Moteur

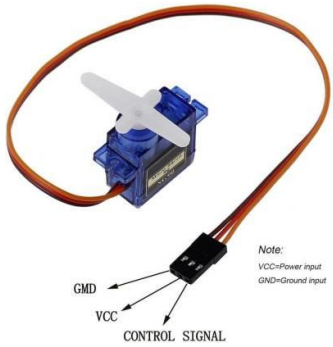


Figure 13 : bornes de Servo moteur

Les servomoteurs sont commandés par l'intermédiaire d'un câble électrique à trois fils qui permet d'alimenter le moteur et de transmettre des consignes de position sous forme d'un signal codé en largeur d'impulsion plus communément appelé PWM. Cela signifie que c'est la durée des impulsions qui détermine l'angle absolu de l'axe de sortie et donc la position du bras de commande du servomoteur. Le signal se répète périodiquement, toutes les 20 millisecondes, ce qui permet à l'électronique de contrôler et de corriger continuellement la position angulaire de l'axe de sortie, cette dernière étant mesurée par le potentiomètre.

2.5. W5100 Shield Ethernet

Le Shield Arduino Ethernet, basé sur le chip Wiznet W5100, permet à une carte Arduino de se connecter à un réseau local Ethernet ainsi qu'à l'Internet, il inclut également un emplacement de carte micro-SD permettant de stocker des données venant du réseau.

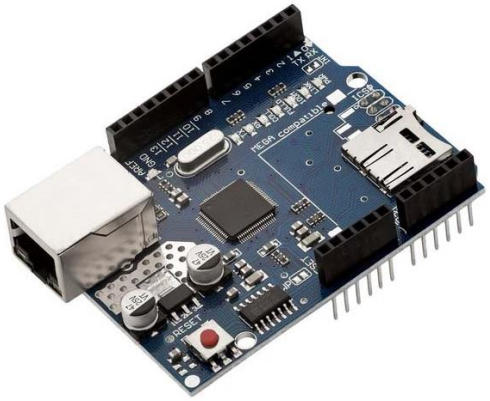


Figure 14 : W5100 Shield Ethernet

3. **Conception** :

On fait la conception sur Proteus pour démontrer la structure matérielle de notre projet :

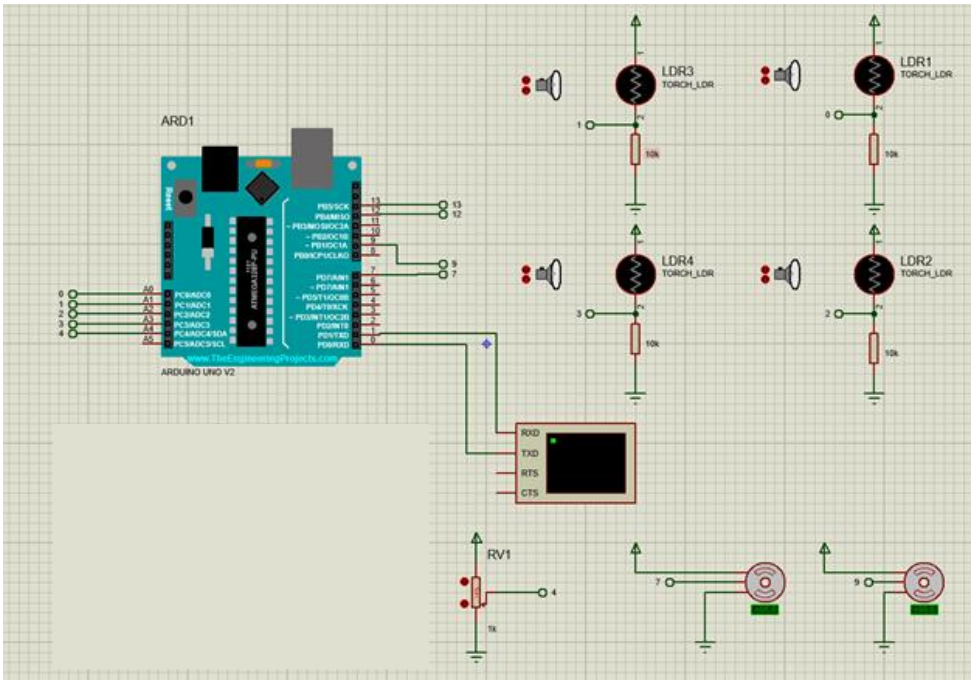


Figure 15 : Conception matériel

4. La commande et mode de fonctionnement :

La commande de suiveur de soleil se fait totalement à travers tableau de bord sous forme d'un site web avec lequel nous interagissons via un navigateur web.

Le suiveur de solaire dispose de deux modes de fonctionnement : manuel et automatique. Un bouton créé dans le tableau de bord a pour rôle de basculer entre les deux modes Lorsqu'on clique sur le bouton une fonction FETCH REQUEST envoie par un fichier javascript avec l'état actuel de bouton au serveur web. Ce dernier va traiter cette requête et stocker l'état de bouton à l'objet de contrôle dans la base des données Firebase puis il l'envoi vers le serveur pour changer le mode de fonctionnement.

Pour que le contrôleur connaisse le mode de fonctionnement sélectionné, il suffit d'envoyer un http requête au serveur sur la chemin /get, ce dernier vat retourner l'objet de control de Firebase vers Arduino qui va extraire le mode de fonctionnement et stocker la position de servo moteur à l'intérieur des variable pour l'utiliser dans le cas du mode manuel,

et pour faciliter le traitement par l'Arduino, la même requête obtient le mode de fonctionnement et les position des servo moteurs et les stocke dans la base de données.

3.1. Mode manuel :

Si le mode manuel est sélectionné, l'utilisateur peut directement contrôler les positions des servomoteurs pour orienter le panneau photovoltaïque vers EST/OUEST par le servomoteur **H Servo** et vers NORD/SUD par le servomoteur **V Servo**. Le contrôle se fait à partir des curseurs associés aux servomoteurs dans le tableau de bord de l'application IoT.

Dans ce mode, Arduino utilise les variables déjà mise à jour par requête de changement de mode de fonctionnement.

3.2. Mode automatique :

Le mode automatique est activé pour contrôler les axes horizontaux et verticaux séparément en utilisant l'algorithme présenté dans la figure16.

L'algorithme commence par lire les valeurs analogiques envoyées par les capteurs LDR et les traite pour commander les servomoteurs qui déplacent le panneau photovoltaïque en direction du soleil. Pour le mouvement du suiveur solaire basé sur l'axe vertical, les valeurs moyennes des deux capteurs LDR de gauche et de droite sont comparées. Si les capteurs de gauche reçoivent plus de lumière, le panneau PV se déplace dans cette direction grâce au servomoteur horizontal dans le sens des aiguilles d'une montre, arrêtant lorsque le résultat de la différence est positif et compris entre la plage de sensibilité. Cette plage permet de stabiliser le contrôleur et de réduire la consommation d'énergie des servomoteurs. Si les capteurs de droite reçoivent plus de lumière, le panneau PV se déplace dans la direction opposée des aiguilles d'une montre, arrêtant lorsque le résultat de la différence est dans la plage de sensibilité. La même approche est utilisée pour le mouvement du suiveur solaire basé sur l'axe horizontal.

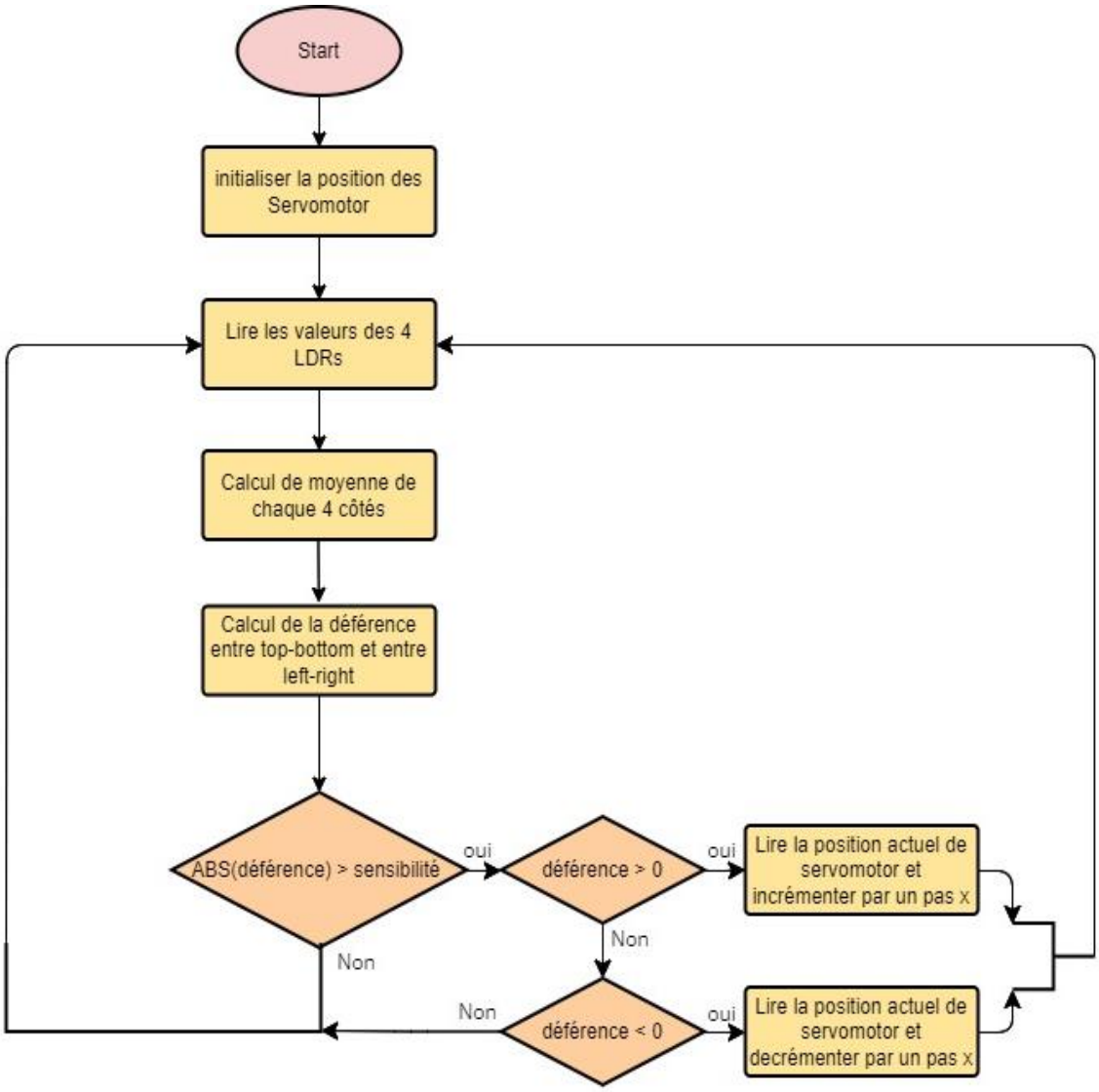


Figure 16 : Organigramme de suivi la lumière pour un axe

De même dans mode automatique, Arduino Méga envoie les positions de servo moteur vers le chemin de serveur /push. Afin de les traiter et mettre à jour les valeurs des positions dans l'objet de control dans la base de données.

Chapitre 3

Réalisation du suiveur solaire

1. Réalisation de la carte de commande et support :

Une carte est construite en utilisant quatre capteurs LDR, disposés en forme de croix, pour identifier les quatre directions (droite, gauche, haut et bas). Si le panneau n'est pas positionné perpendiculairement au soleil, au moins un des capteurs LDR sera recouvert d'ombre par la croix, ce qui entraînera une variation de l'intensité lumineuse. Les valeurs de luminosité des quatre directions seront transmises par les ports analogiques de la carte Arduino via les capteurs LDR. Les calculs nécessaires pour déterminer la position optimale du panneau solaire seront effectués à l'aide de ces données afin de commander les moteurs.

1.1. Carte de commande :

On réalise le montage grâce à un circuit imprimé universel, 4 LDR, et 4 résistances de 13kΩ.

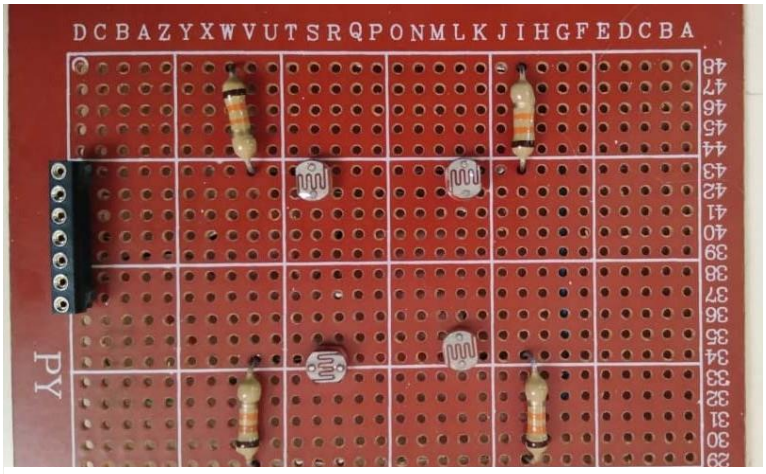


Figure 17 : carte de commande

1.2. Carte de distribution d'alimentation et d'acquisition :

a) Étalonnage de capteur de tension :

L'étalonnage est une opération qui concerne les appareils de mesure ou de restitution de données. Deux appareils différents de conception différente, mais aussi deux appareils de la même gamme (même marque, même modèle) ne réagissent pas exactement de la même manière. Il faut donc une procédure permettant d'obtenir le même résultat à partir de la même situation initiale.

On utilise comme une référence pour l'étalonnage du capteur de tension un GBF, après avoir remplir un tableau Excel avec défirent valeur de tension de 0.3 V jusqu'à 19.4 V de tension de référence et noter la mesure de capteur pour chaque tension de référence. Ensuite on génère un repère (CHART) dans son axe horizontal la mesure de capteur et en son axe vertical les tensions de référence pour enfin d'extraire une équation linéaire d'étalonnage (voir la figure 19).

Figure 18: GBF

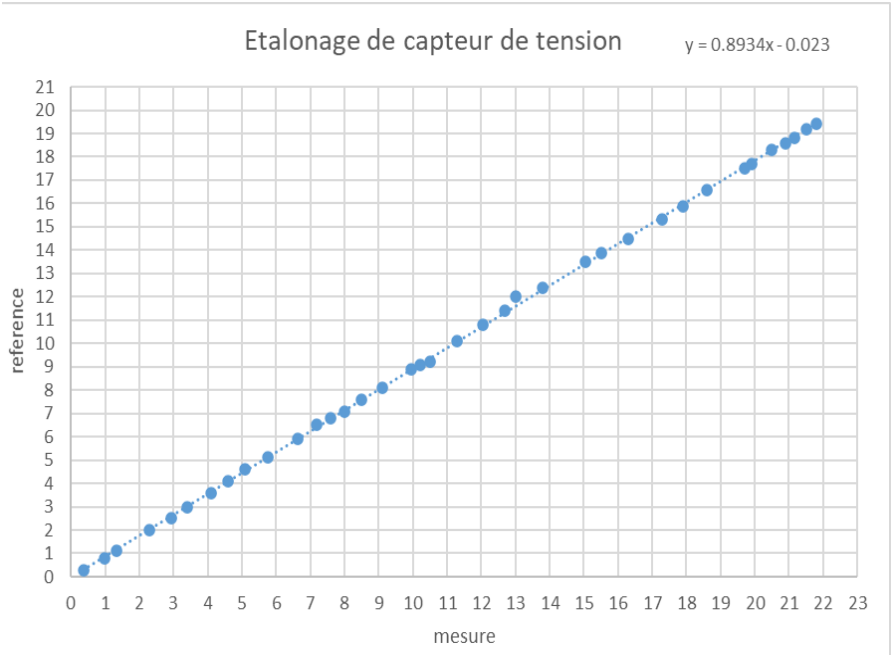


Figure 19 : Graphe de courbe de linéarisation de capteur

Et le résultat d'étalonnage est une équation linaire qui est : $y = 0.8934x - 0.023$ avec y représente la mesure corrigée par l'équation d'étalonnage et x représente la mesure par le capteur de tension.

b) Conception de la carte de distribution et d'acquisition :

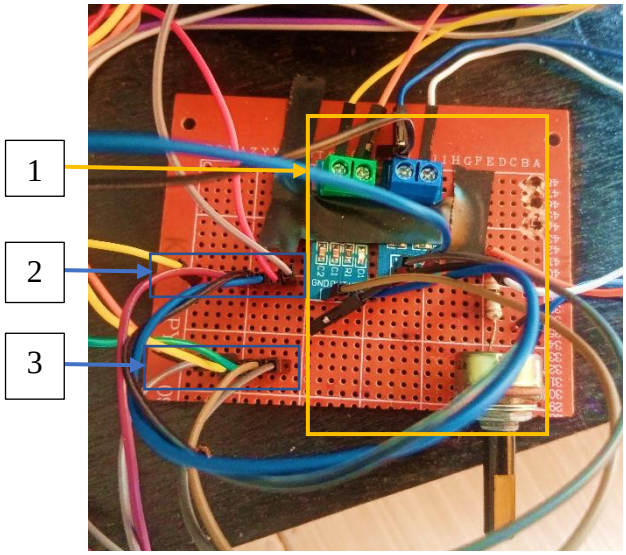


Figure 20 : Carte de distribution et d'acquisition

Cette carte est de but de distribuer l'alimentation 5V d'Arduino pour alimenter les 4 LDR, les 2 servo moteur et les capteurs de tension et d'intensité de courant.

1-Montage des capteurs courant et tension

2-GND

3-5V

Cette carte a également un circuit d'acquisition à base de capteurs de tension/courant. Le montage qu'on utilise dans ce circuit et celui-là :

Montage des capteurs de tension et de courant :

L'ajout d'une charge résistive à un circuit d'acquisition. Cette charge résistive est contrôlée par un potentiomètre de 15kΩ. Pour éviter tout risque de court-circuit lorsque la valeur de résistance du potentiomètre est presque nulle, une résistance supplémentaire est ajoutée en série avec le potentiomètre pour protéger le circuit.

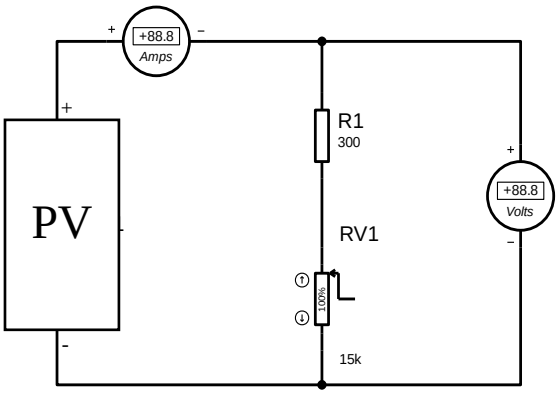


Figure 21 : circuit d'acquisition

1.3. Support et résultat final :

Ce support est réalisé par les étudiants de l'année dernière :

- Meryem EL OUAZZANI-TAYIBI
- Ismail YADEN

Le support du suiveur solaire, qui constitue la partie mécanique de l'appareil, est principalement composé de deux éléments distincts :

- Le socle, qui accueille le moteur responsable de la rotation horizontale.
- La partie supérieure du suiveur, où est fixé le second moteur pour permettre la rotation verticale du panneau solaire.

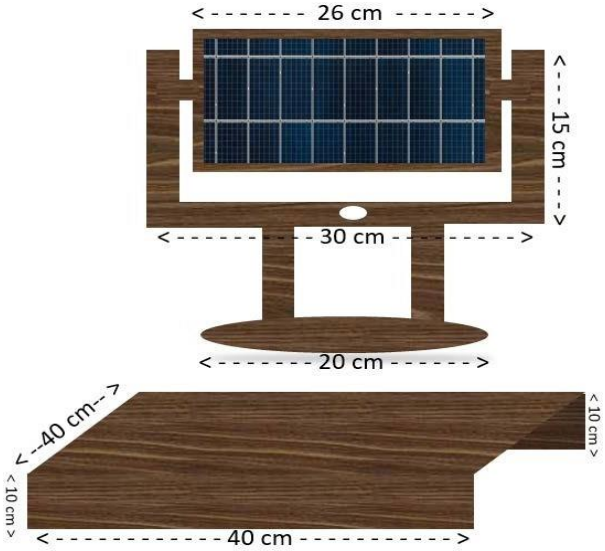


Figure 22 : Représentation schématique du support du suiveur solaire

Après avoir monté et fixé les différentes cartes du suiveur, on obtient le produit suivant :

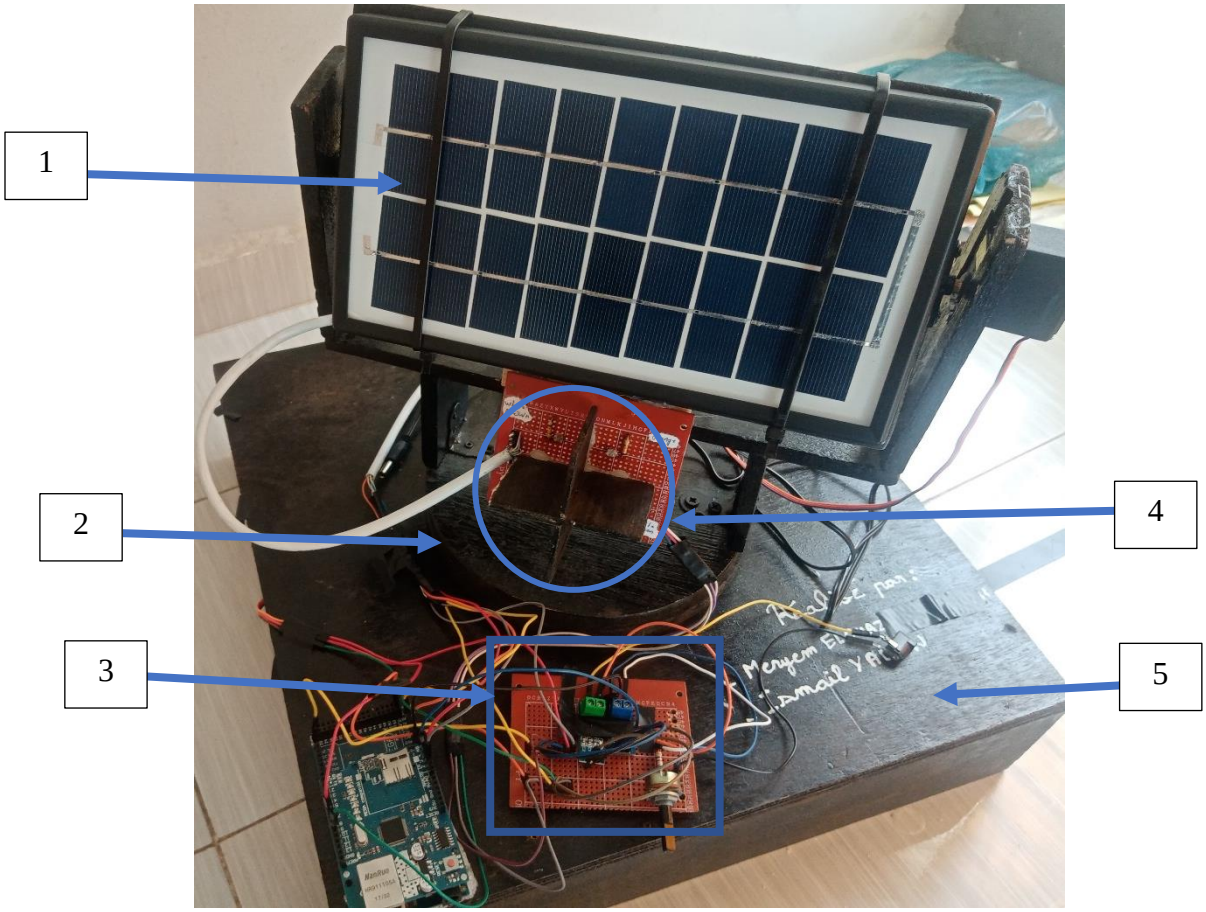


Figure 23 : Prototype de suiveur de soleil

1- Panneau voltaïque qui est fixé à la partie supérieure du suiveur, c'est le cadre permettant la rotation suivant l'axe vertical.

2- Support du de l'ensemble cadre plaque, contenant le servomoteur à gauche qui génère la rotation verticale.

3- Carte de distribution d'alimentation et d'acquisition de tension/courant de panneau voltaïque.

4- Carte de commande.

5- Le socle avec le servomoteur fixé au centre, permettant la rotation horizontale de l'ensemble des éléments cités auparavant.

2. Développement d'application de surveillance IOT :

2.1. Architecture d'application IOT :

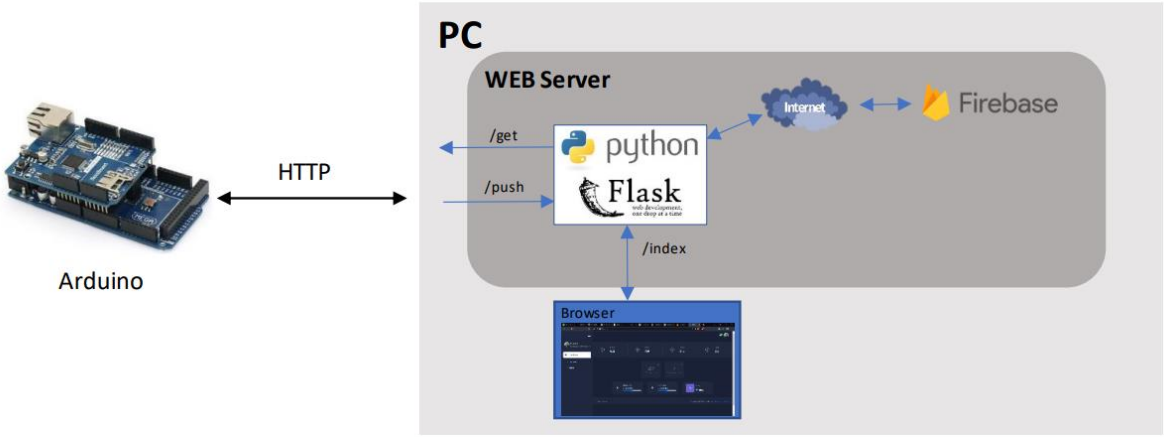


Figure 24 : Architecture d'application IOT

Fonctionnement d'architecture :

- **Étape 1 :** Arduino envoie une requête HTTP au serveur Web, qui est une demande pour accéder à une ressource spécifique sur le serveur. Cette requête contient des informations sur ce que Arduino souhaite obtenir du serveur.
- **Étape 2 :** Le serveur Web reçoit la requête HTTP d'Arduino et exécute la fonction Python correspondante au chemin de la requête. Cette fonction est un script Python qui a été écrit pour gérer les demandes spécifiques que le serveur peut recevoir.
- **Étape 3 :** La fonction Python récupère les données de la requête HTTP et les traite pour les rendre exploitables. Ensuite, la fonction Python interagit avec la base de données Firebase pour récupérer ou enregistrer les données nécessaires.
- **Étape 4 :** La fonction Python traite les résultats de la requête et renvoie une réponse HTTP à Arduino, qui contient les informations demandées ou un message indiquant si la demande a réussi ou non. La réponse HTTP est un protocole de communication qui permet à Arduino de comprendre les données transmises.

Cette architecture est utilisée pour développer une application IoT pour notre projet. Chaque fois que la fonction `loop()` d'Arduino est exécutée, elle envoie les lectures des capteurs LDR ainsi que les lectures de tension et de courant du panneau solaire à l'aide de la fonction `pushLDR_Power()`. Cette fonction construit une requête HTTP avec les données nécessaires de LDR, de tension et de courant et les envoie au serveur pour traitement. Une autre requête HTTP est envoyée séparément pour indiquer le mode de fonctionnement (automatique ou manuel).

Lorsque le mode automatique est activé, une autre requête est envoyée par Arduino vers le serveur web. Cette requête est basée sur l'algorithme de suivi de lumière (voir la figure 16) et permet de changer la position des servomoteurs. À la fin de la fonction `automaticMode()`, une autre fonction appelée `pushPosition()` est utilisée pour construire une requête avec la nouvelle position des servomoteurs. Cette requête est ensuite envoyée au serveur pour mettre à jour l'objet de contrôle dans la base de données Firebase.

2.2. Technologie utilisée :

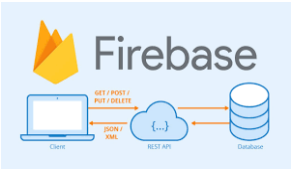
a. Python Flask:

Flask est un micro Framework open-source de développement web en Python. Il est classé comme micro-Framework car il est très léger. Flask a pour objectif de garder un noyau simple mais extensible. Il n'intègre pas de système d'authentification, pas de couche d'abstraction de base de données, ni d'outil de validation de formulaires. Cependant, de nombreuses extensions permettent d'ajouter facilement des fonctionnalités.

Le noyau d'application IOT de suiveur de solaire qui est le serveur web est développé on utilise python le Framework de développement web Flask.

b. Firebase :

Firebase est un outil d'aide au développement d'applications web, Android, iOS, et Unity. Il propose un hébergement en nuage et utilise NoSQL pour héberger des bases de données.



Il propose des logiciels utilisés pour le développement d'applications mobiles pour enregistrer des données, envoyer des notifications et des publicités, remonter les erreurs et les clics effectués sur l'application. Il fonctionne dans une application et est invisible de l'utilisateur de l'application. Firebase aussi est populaire dans les applications IOT.

Conception de base de données :

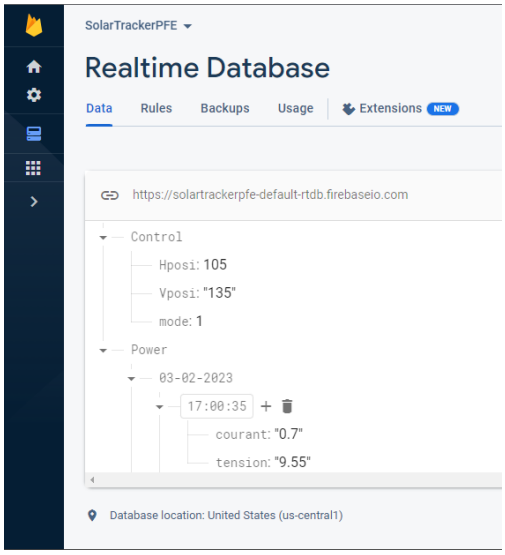


Figure 25: contenu de base de données

Premier élément présenter dans notre base de données est l'objet « Control » qui est utilisé pour stocker les positions des servo moteurs par une valeur de 0 à 180 dans ces deux variables :

- **Hposi** pour la position de servo moteur de rotation horizontal.
- **Vposi** pour la position de servo moteur de rotation vertical.
- **mode** pour stocker l'état de mode de fonctionnement de suiveur de soleil (1 = mode automatique , 0 = mode manuel).

Deuxième élément l'objet « Power », qui est utiliser pour stocker les enregistrements des tension/courant par « Heure:Minute:Seconde » et aussi en même temps par « Jour:Mois:Ans ».

2.3. Conception logiciel d'application IOT :

a. Front-end :

La page de surveillance IOT ou (Interface graphique utilisateur) ou table de bord est développé on utilise les langues de web : HTML, CSS, jQuery.

- JQuery est un Framework JavaScript conçu pour simplifier la traversée et la manipulation de l'arbre HTML DOM, ainsi que le traitement des événements, l'animation CSS et l'Ajax. Il s'agit d'un logiciel libre et open source sous la

licence MIT. En août 2022, jQuery est utilisé par 77% des 10 million de sites Web les plus populaires.

Le table de bord porte un système d'authentification il faut aux premières fois à créer un compte dans la page d'inscription et ensuite se connecter au page de connexion pour accéder au page de surveillance de suiveur de soleil.

Page de connexion :

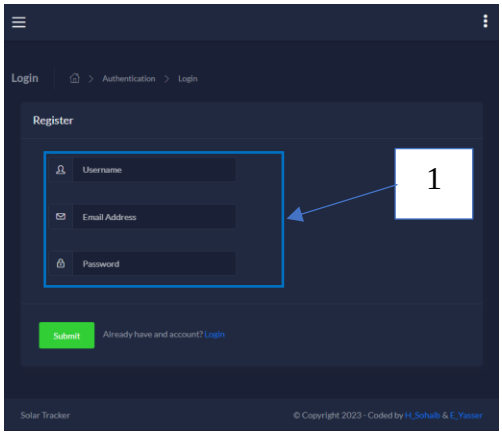


Figure 26 : Page d'inscription

1- Lors de la première connexion, l'utilisateur sera invité à entrer ses informations (nom d'utilisateur, l'adresse email et un mode passe). Une fois l'utilisateur appuie sur le bouton « Submit » les informations dans les champs d'entrer sera envoyer vers le serveur web via un POST http requête.
   Le serveur extraire les informations et réserve une ligne à la table dans une autre base de données de type SQLite utilisé pour le système d'authentification pour pouvoir se connecter la prochaine fois.

   - SQLite est un moteur de base de données écrit dans le langage de programmation C. Ce n'est pas une application autonome, c'est plutôt une bibliothèque que les développeurs de logiciels intègrent dans leurs applications. Il appartient donc à la famille des bases de données intégrées.

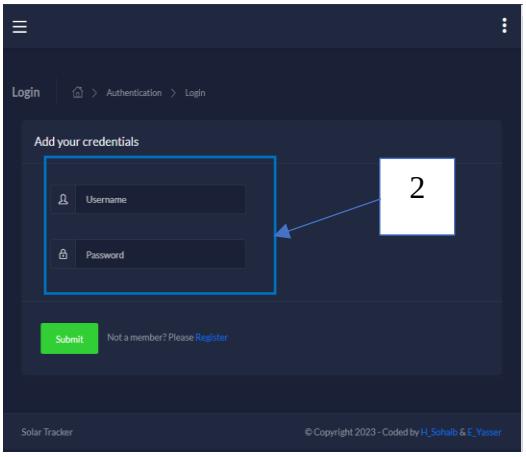


Figure 27 : Page de connexion

2- Après l'enregistrement et création du compte, l'utilisateur est demande de saisir le nom d'utilisateur et le mode passe pour ces informations d'arriver au serveur web via un autre POST requête, et de sa part vérifie si existe-t-il un compte enregistré aux bases de données du système d'authentification pour le nom d'utilisateur et le mode passe entrer pour accéder enfin à la page de surveillance.

Tableau de bord :

Figure 28 : page de surveillance

3. Partie que l'utilisateur peut choisir parmi les 2 pages (Tableau de bord ou tableaux Des enregistrements d'intensité de courant et tension de panneau voltaïque).

4. Affichage des valeurs des LDR au temps réel.

5. Affichage de différences horizontale et verticale ou l'erreur de position verticale et horizontale en se base sur les valeurs LDR si ces grandeur et proche de 0 c'est indiqué que le panneau voltaïque est bien orienté vers le soleil.
6. Le contrôle du servo horizontal et vertical dans le cas du mode manuel et en mode automatique les 2 vecteurs sont verrouillés et sont affichés les nouvelles positions des servo moteurs qui est changé par algorithme de suivi de lumière (voir la fig. 16) au temps réel.
7. bouton de sélection de mode automatique et manuel.

Page des tableaux de courant et tension :

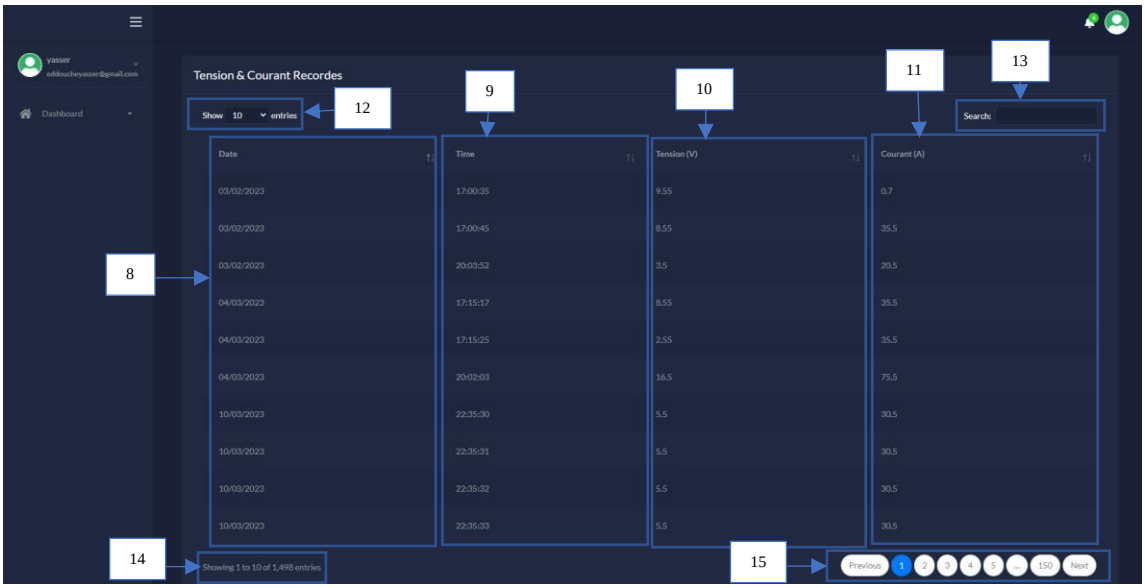


Figure 29 : page des enregistrement tension/courant

8. Affichage des dates de chaque enregistrement.
9. Affichage de temps de chaque enregistrement.
10. Affichage de tension enregistrer.
11. Affichage d'intensité de courant enregistrer.
12. Choisir le nombre des enregistrements affiché dans une seul page, peut être « 10, 25, 50 ou 100 ».
13. Barre de recherche des enregistrements par la date, le temps et les valeurs.
14. Visualisation des nombres des enregistrements afficher.

15. Visualisation des pages des enregistrements.

b. Interface de programmation d'application (Les APIs) :

API (Application Programming Interface) est un moyen de communication entre différents programmes informatiques. Elle permet à ces programmes de se connecter et d'échanger des informations ou des services. C'est une sorte de pont qui permet à différents programmes de travailler ensemble de manière transparente.

Une spécification API est un document qui décrit comment les programmes doivent communiquer entre eux en utilisant l'API. Elle peut décrire les formats de données acceptables, les méthodes d'accès, les protocoles de sécurité, et d'autres détails techniques nécessaires pour utiliser l'API de manière efficace. Ces spécifications sont essentielles pour assurer l'interopérabilité entre différents programmes et pour permettre le développement de nouvelles applications utilisant l'API existante.

On utilise les APIs développées en JavaScript pour récupérer en temps réel les données du suiveur de soleil à partir du serveur et les afficher sur la page de surveillance.

1- API prend les valeurs des curseurs et les envoyer vers le serveur, puis il interagit avec Firebase ce qu'il mise à jour les positions dans l'objet de contrôle de base des données. Enfin, Arduino envoie une requête à l'objet de contrôle, qui permet de contrôler les servomoteurs du tableau de bord.

2- Une API permet de récupérer les valeurs des capteurs LDR et des positions des servomoteurs d'Arduino. Ces données sont stockées dans des vecteurs de servomoteurs en mode automatique.

3- API qui prend l'état du bouton à partir de tableau de bord, à chaque appui il l'envoie vers le serveur qui la traite et mettre à jour l'état de mode dans l'objet de contrôle du Firebase.

c. Back-end:

Au niveau du serveur web, chaque chemin exécute une fonction qui prend en charge toute la logique en arrière-plan et gère les réponses des requêtes provenant soit d'Arduino, soit des API de la table de bord.

- /get :

```python
@app.route("/get")
def get():
    # get control child from firebase
    try:
        control = firebaseDB.child("Control").get().val()
        # print(control)
    except requests.exceptions.ConnectionError:
        return "Check Ur connection !!"
    except requests.exceptions.ConnectTimeout:
        return "Connection Time Out"
    except Exception as e:
        print(e)
        return "unexpected error occurred !!"
    # send response
    return f'start {control["mode"]},{control["Vposi"]},{control["Hposi"]} end.'
```

Figure 30 : fonction qui exécute à chaque requête au chemin /get

Arduino envoie une requête pour récupérer l'objet « Control » de la base de données Firebase. Ce chemin ne renvoie qu'une seule réponse. À chaque requête GET d'Arduino, la fonction associée à ce chemin interagit avec la base de données Firebase pour récupérer l'objet « Control » et extraire son contenu pour reformer une chaîne de caractères, car seules les chaînes de caractères peuvent être renvoyées en tant que réponse du serveur. Cette fonction est appelée `getControlObject()`. Elle collecte la réponse du serveur sous forme de chaîne de caractères et la manipule pour extraire l'état de mode et la position des servo-moteurs.

- /push :

Ce chemin est utilisé pour envoyer les données acquises par Arduino, telles que les valeurs des LDR et les valeurs acquises par le capteur de tension et de courant, en utilisant une

requête GET vers ce chemin avec des arguments (chaîne de requête) contenant les données mentionnées précédemment.

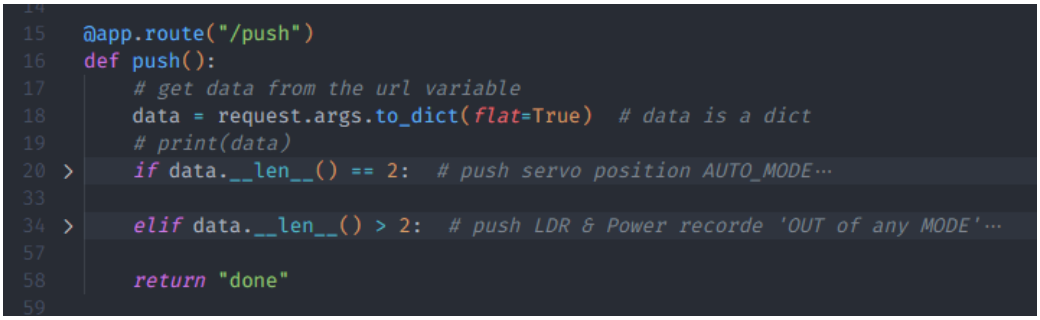


Figure 31 : fonction qui exécute à chaque requête au chemin /push

La fonction push() associée au chemin reçoit la requête contenant les informations envoyées par Arduino. Elle prend ces données et les convertit en un type de données en Python appelé "dictionnaire" afin que le serveur puisse les manipuler et déterminer s'il s'agit des valeurs des LDR et de la tension/courant ou des positions des servo-moteurs. En effet, chaque type de données sera traité différemment afin d'être stocké à sa place dans la base de données Firebase.

- /index :

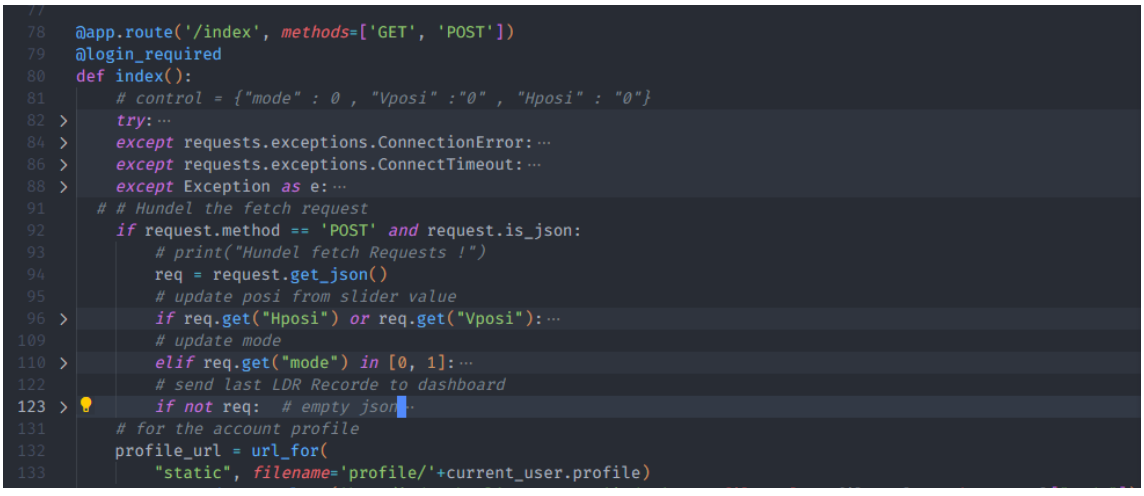


Figure 32 : fonction qui exécute à chaque requête au chemin /index

Ce chemin est responsable de fournir le tableau de bord et de fournir toutes les informations nécessaires aux APIs de tableau de bord. La fonction « index() » sera exécutée à chaque demande de l'utilisateur ou des APIs de tableau de bord. Cette fonction recevra la demande, l'analysera pour déterminer si elle provient d'une API qui demande la valeur LDR ou de l'API qui met à jour le mode de fonctionnement.

- /index/tables :

```
136
137  @app.route('/index/tables')
138  @login_required
139  def tables():
140      try:
141          data = dict(firebaseDB.child("Power").get().val())
142      except requests.exceptions.ConnectionError:
143          return "Check Ur connection"
144      except requests.exceptions.ConnectTimeout:
145          return "Connection Time Out"
146      except Exception as e:
147          print(e)
148          return "unexpected error occurred !!"
149
150      profile_url = url_for("static", filename='profile/'+current_user.profile)
151      return render_template('home/tables-data.html', profile_url=profile_url, data=data)
152
```

Figure 33 : fonction qui exécute à chaque requête au chemin /index/tables

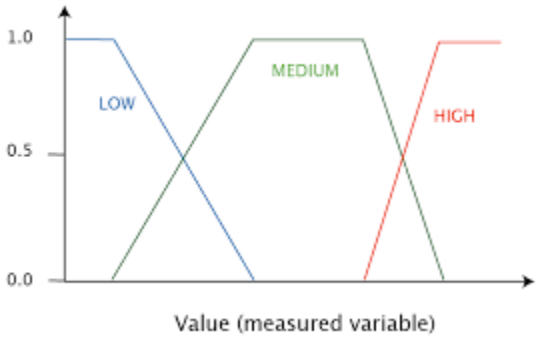
Ce chemin a pour but de servir la page qui contient les tableaux d'affichage des enregistrements des valeurs de tension/courant. La fonction associée à ce chemin interagit avec la base de données Firebase pour récupérer tous les enregistrements et les stocker dans une variable de type "dictionnaire". Ensuite, cette variable est transmise à la partie front-end pour être présentée sous la forme d'un tableau.

3. Logique floue (Fuzzy logic) :

L'objectif de l'utilisation de la logique floue est de fournir une méthode de contrôle automatique alternative qui permet d'intégrer l'intelligence artificielle dans notre projet, et qui remplace l'algorithme de suivi de lumière de base présenté dans la figure 16.

3.1. Définition :

La logique floue est une méthode utilisée en intelligence artificielle basée sur des "valeurs ou degrés de vérité" sous forme de nombres réels compris entre 0 et 1. Cela diffère de la logique booléenne traditionnelle, qui est basée sur deux valeurs "vrai ou faux". (1 ou 0). En d'autres termes, cette logique dite universelle admet la possibilité d'une vérité partielle aux extrêmes entre 0 et 1. La logique floue a été proposée par le mathématicien Lotfi Zadeh dans les années 1960 dans le cadre de ses recherches sur la compréhension du langage.



Difficile à traduire en valeur absolue et en termes binaires tels que 0 ou 1, le langage naturel ne peut être décrit par une logique booléenne stricte. Pour résoudre ce problème, il faut développer une logique plus proche du fonctionnement du cerveau humain, qui tend à combiner des faits partiels pour prendre des décisions acceptables. La logique floue le fait en représentant plus fidèlement les capacités cognitives humaines. À cet égard, la conception et le développement de capacités d'IA pour des tâches complexes et/ou inhabituelles sont essentiels

3.2. Architecture de Logique floue :

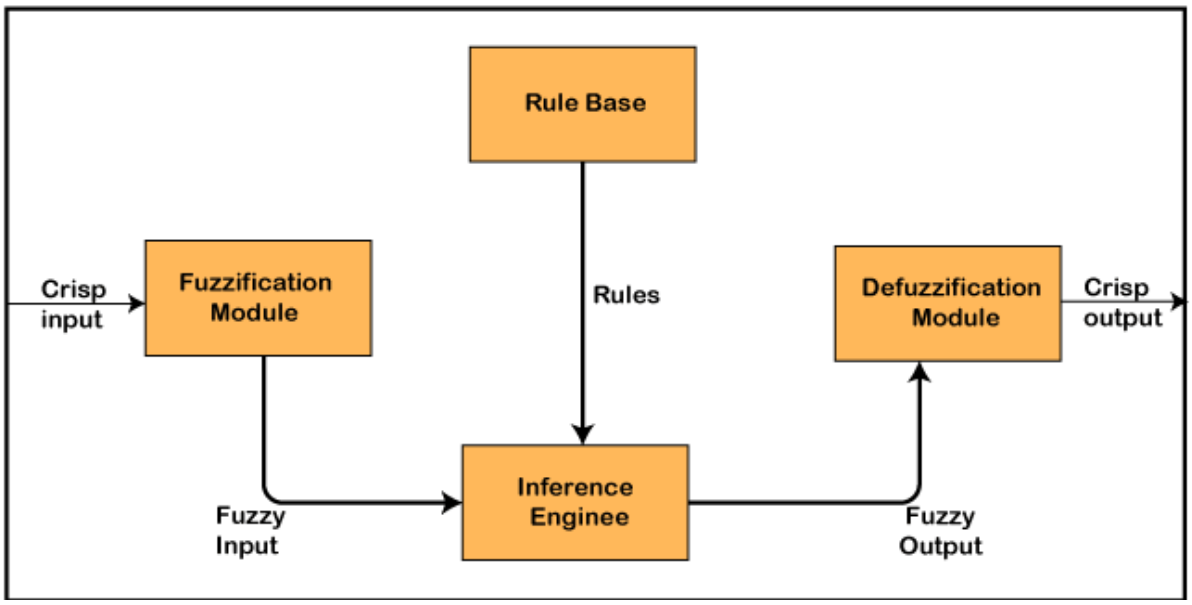


Figure 34 : Architecture du Fuzzy Logic.

Fuzzification module: Il transforme les données d'entrée du système, qui sont des nombres bruts, en ensembles flous. Dans notre cas on a 2 étapes :

- -Light : petite luminosité.
- +Light : grande luminosité.

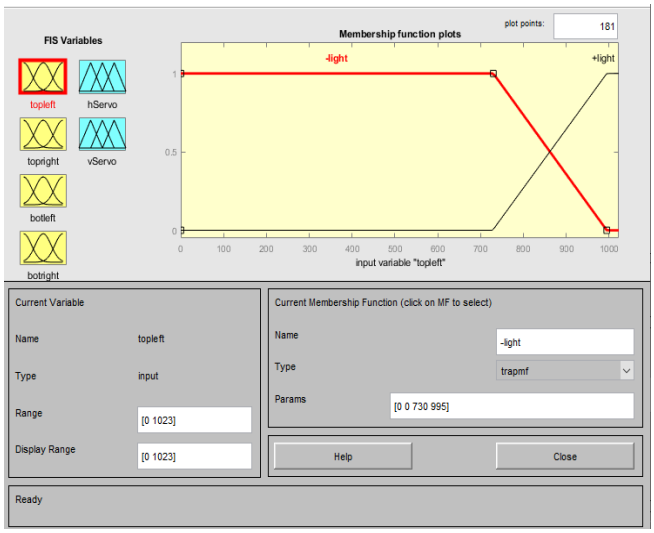


Figure 35 : Fuzzification module

Rule Base: Il stocke les règles IF-THEN.

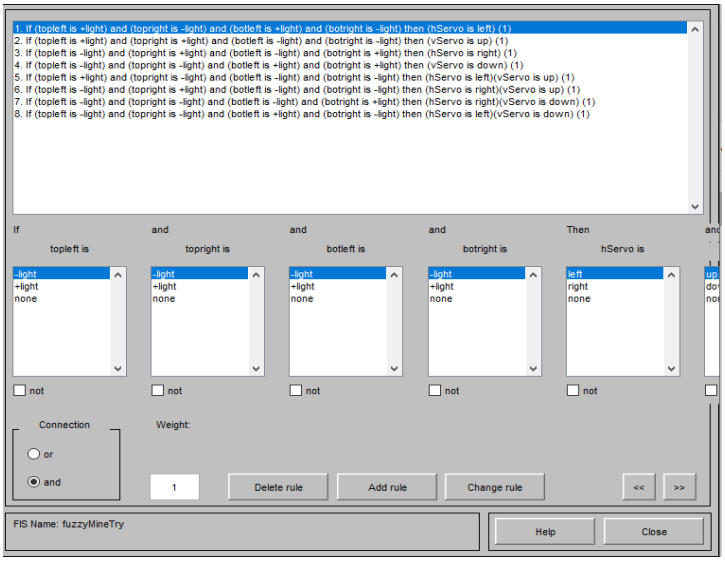


Figure 36 : Rule base

Inference Engine: Il simule le processus de raisonnement humain en effectuant des inférences floues sur les entrées et les règles IF-THEN.

Defuzzification Module: Il transforme l'ensemble flou obtenu par le moteur d'inférence en une valeur nette.

- **Left :** le Servo tourne à gauche.
- **Right :** le Servo tourne à droit.

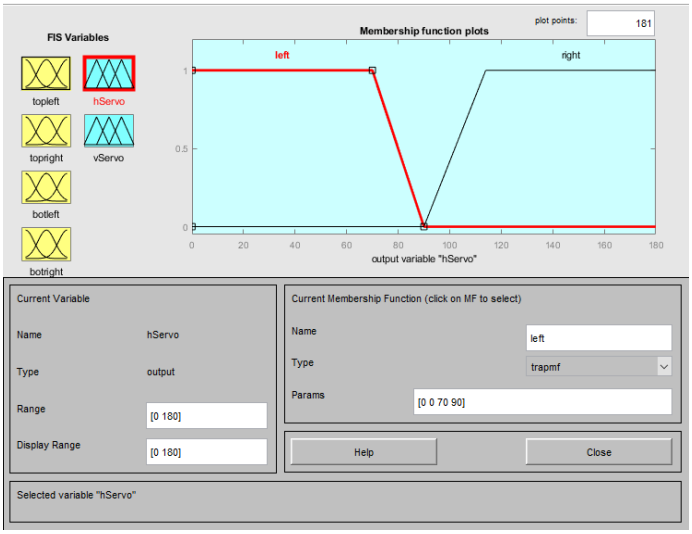


Figure 37 : Defuzzification Module

Membership fuctions (Les fonctions d'appartenance) : fonctionnent sur des ensembles flous de variables. Les fonctions d'appartenance permettent de quantifier un terme linguistique et de représenter graphiquement un ensemble flou.

Chaque composant a ses fonctions d'appartenance :

- LDR : chaque LDR a les mêmes fonctions d'appartenance :

  topleft = -Light [0 0 775 825] & +Light [849 966 1023 1023]
  topright = -Light [0 0 775 825] & +Light [849 966 1023 1023]
  botleft = -Light [0 0 775 825] & +Light [849 966 1023 1023]
  botright = -Light [0 0 775 825] & +Light [849 966 1023 1023]

- Servo : Servo horizontal et verticale aussi ont les mêmes :

  Hori Servo = Left [0 0 70 90] & Right [90 114 180 180]
  Verti Servo = Left [0 0 70 90] & Right [90 114 180 180]

3.3. Conversion de conception de la logique floue en code Arduino:

Le développement d'un système de logique floue se fait en Arduino à l'aide d'une bibliothèque « eFLL » qui contient des classes programmées en C++ au but de réaliser les 3 phase de conception d'une logique floue.

a) eFLL ou Fuzzy :

eFLL (Embedded Fuzzy Logic Library) est une bibliothèque standard pour les systèmes embarqués permettant de mettre en œuvre des systèmes flous simples et efficaces.

Écrit en C++/C, utilise uniquement la bibliothèque de langage C standard « stdlib.h », ainsi eFLL est une bibliothèque conçue non seulement pour Arduino, mais tout système intégré ou non la façon dont les commandes sont écrites en C.

Il ne comporte aucune limite explicite quant à la quantité de règles floues, d'entrées ou de sorties, ces limites de puissance de traitement et de stockage de chaque microcontrôleur.

b) Fuzzification module :

- La création des fonctions d'appartenance en utilisant la class **FuzzySet.**

```
// FuzzyInput      :   intervals   Vdifferent
FuzzySet *tlMlight = new FuzzySet(0, 0, 775, 825);
FuzzySet *tlPlight = new FuzzySet(849, 966, 1175, 1175);
// FuzzyInput      :   intervals   Vdifferent
FuzzySet *trMlight = new FuzzySet(0, 0, 775, 825);
FuzzySet *trPlight = new FuzzySet(849, 966, 1175, 1175);
// FuzzyInput      :   intervals   Vdifferent
FuzzySet *brMlight = new FuzzySet(0, 0, 775, 825);
FuzzySet *brPlight = new FuzzySet(849, 966, 1175, 1175);
// FuzzyInput      :   intervals   Vdifferent
FuzzySet *blMlight = new FuzzySet(0, 0, 775, 825);
FuzzySet *blPlight = new FuzzySet(849, 966, 1175, 1175);

// FuzzyOutput      SH
FuzzySet *left = new FuzzySet(0, 0, 45, 95);
FuzzySet *right = new FuzzySet(85, 135, 180, 180);
// FuzzyOutput      SV
FuzzySet *down = new FuzzySet(0, 0, 45, 95);
FuzzySet *up = new FuzzySet(85, 135, 180, 180);
```

Figure 38: Fuzzifiaction module pour Arduino

- La création des entrées et sorties des systèmes en spécifient pour chaque entrée ses fonctions d'appartenance :

```
// FuzzyInput  tr
FuzzyInput *tr = new FuzzyInput(1);

tr->addFuzzySet(trMlight);
tr->addFuzzySet(trPlight);
fuzzy->addFuzzyInput(tr);
// FuzzyInput  tf
FuzzyInput *tl = new FuzzyInput(2);

tl->addFuzzySet(tlMlight);
tl->addFuzzySet(tlPlight);
fuzzy->addFuzzyInput(tl);
// FuzzyInput  br
FuzzyInput *br = new FuzzyInput(3);

br->addFuzzySet(brMlight);
br->addFuzzySet(brPlight);
fuzzy->addFuzzyInput(br);
// FuzzyInput  bl
FuzzyInput *bl = new FuzzyInput(4);

bl->addFuzzySet(blMlight);
bl->addFuzzySet(blPlight);
fuzzy->addFuzzyInput(bl);
```

Figure 39: Création des entrées de système

- À les sorties du système, on utilise **`addFuzzyOutput`** à la place de **`addFuzzyInput.`**

```
// FuzzyOutput   SH
FuzzyOutput *sh = new FuzzyOutput(1);

sh->addFuzzySet(left);
sh->addFuzzySet(right);
fuzzy->addFuzzyOutput(sh);
// FuzzyOutput   SV
FuzzyOutput *sv = new FuzzyOutput(2);

sv->addFuzzySet(down);
sv->addFuzzySet(up);
fuzzy->addFuzzyOutput(sv);
```

Figure 40 : Création des sorties de système

c) Rules Base :

'if (topleft is +light) and (topright is -light) and (botleft is +light) and (botright is -light) then (hServo is left)'

En code Arduino cette règle est codé de cette manière :

```
// Building FuzzyRule ********** 1
FuzzyRuleAntecedent *tlmlightANDtrmlight = new FuzzyRuleAntecedent();
tlmlightANDtrmlight->joinWithAND(tlMlight, trMlight);
FuzzyRuleAntecedent *blplightANDbrmlight = new FuzzyRuleAntecedent();
blplightANDbrmlight->joinWithAND(blPlight, brMlight);
// if
FuzzyRuleAntecedent *rul1 = new FuzzyRuleAntecedent();
rul1->joinWithAND(tlmlightANDtrmlight, blplightANDbrmlight);
// then
FuzzyRuleConsequent *thenShLeft = new FuzzyRuleConsequent();
thenShLeft->addOutput(not(left));
```

Figure 41: Création d'une règle

d) Mise en marche le système en Arduino :

Après avoir créé les entrées, les sorties, les fonctions d'appartenance et les règles pour la logique floue, il faut démarrer le système pour effectuer les étapes de **Fuzzification**, **Inference Engine** et **Defuzzification** afin d'obtenir les valeurs des sorties **:**

```
// les entrees de system
  fuzzy->setInput(1, topr);
  fuzzy->setInput(2, topl);
  fuzzy->setInput(3, botr);
  fuzzy->setInput(4, botl);
// demarrage de phase de Fuzzification module
  fuzzy->fuzzify();
// demarrage de phase Inference Engine et Defuzzification
  int shposi = fuzzy->defuzzify(1);
  int svposi = fuzzy->defuzzify(2);
```

Figure 42: Mise en marche

Conclusion

La réalisation de suiveur solaire contrôlé par l'IoT est une contribution importante à l'efficacité énergétique et à la durabilité environnementale. Les trackers solaires aident à maximiser la production solaire en suivant la trajectoire du soleil tout au long de la journée, augmentant ainsi la quantité d'énergie produite.

Nous avons conçu et réalisé un suiveur solaire commandé par l'Internet des objets (IoT) qui est constitué de 2 axes et qui suivre automatiquement le soleil à l’aide des capteurs LDR, ou manuellement par l’utilisateur via le tableau de bord d’une application IOT.

En fin de compte, le projet a été un succès, prouvant la faisabilité de la conception et de la mise en œuvre d'un suiveur solaire contrôlé par l'IoT. Il peut être utilisé dans diverses applications telles que les maisons, les bâtiments commerciaux, les centrales solaires et les voitures solaires.

Annexe A

Fiche technique

Servo moteur :

Dimensions	40 x 20 x 36,5 mm
Type	standard "TowerPro MG995"
Poids	environ 60g
Vitesse	0.16 sec/60° sous 4.8V - 0.13 sec/60° sous 6.0V
Couple	11Kg/cm sous 4.8V - 13Kg/cm sous 6.0V
Tension	4.8V - 6V
Prise servomoteur standard	Orange Rouge Marron
Débattement angulaire	180°
Consommation	120 mA sans charge, 1 450 mA pour 11Kg/cm environ
Temps de réponse de largeur d'impulsion	5 microsecondes ou moins
Erreur de déviation angulaire	retour à 0 degrés, 45 ° autour de chaque 3 ° ou moinsd'erreur.
Orange	Signal PWM TTL standard de modélisme, de période20ms avec un temps haut compris entre 0,7 et 2,3ms environ
Rouge	Alimentation positive entre 4,8V et 6V
Marron	Masse de l'alimentation et du signal

Capteur Tension B25 0-25V :

Plage de mesure de tension	0 à 25 volts
Précision de mesure	Inférieure à 1 % de la pleine échelle
Impédance d'entrée	Supérieure à 1 MΩ
Temps de réponse	Inférieur à 1 seconde
Tension de sortie	Comprise entre 0 et 5 volts
Courant de sortie	Inférieur à 20 mA
Fréquence de réponse	Supérieure à 20 Hz
Température de fonctionnement	Comprise entre -20 et 70 °C
Humidité relative de fonctionnement	Inférieure à 85 %
Plage de mesure	DC 0.02445V ~ 25V
Résolution de mesure	0.00489V

Capteur Courant ACS712 – 5A :

Dimensions	31x13x15mm
Puce	ACS712ELEC-05A
Gamme de courant mesuré	-5A à +5A
Vréf à 0A	Vcc/2 soit 2.5V
Sensibilité	185mV/A
Isolation	2.1KV
Consommation	10mA
Erreur	1.5% à 25°C
Alimentation	5VDC (4.5-5.5VDC)
Poids	2g

Bibliographie

[1] https://github.com/H-sohaib/SolarTracker-with-WebServer_CodeArduino.git : code Arduino utilisé

[2] https://github.com/H-sohaib/PFESolarTrackerDashboard.git : code source de tableau de bord de suiveur de soleil

[3] https://fr.wikipedia.org/

[4]https://www.tutorialspoint.com/artificial_intelligence/artificial_intelligence_fuzzy_logic_systems.htm

[5] https://blog.zerokol.com/2012/09/arduinofuzzy-fuzzy-library-for-arduino.html : documentation de logique floue pour Arduino

[6] https://arduinogetstarted.com/tutorials/arduino-mysql