

**REPUBLIQUE ALGERIENNE DEMOQUERATIQUE ET POPULAIRE  
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE  
SCIENTIFIQUE  
UNIVERSITE SAAD DAHLEB BLIDA 1**



**Faculté de Technologie**

**Département D'Automatique et Electrotechnique**

**Spécialité d'Automatique et Informatique Industrielle**

**Mémoire de projet de fin d'Etude de Master**

**Intitulé du projet**

# **Rayonnage Intelligent**

**Réalisé par :**

**Bouzouidja Hadjer**

**Benzira Hadjer**

**Encadré par :**

**Mr. Bennila Nouredine**

**Promotion : 2022-2023**

## Dédicace

Ce travail est dédié à ceux qui ont été ma source d'inspiration, de soutien et de motivation.

À toi, ma chère maman, qui m'a accompagné tout au long de ce parcours académique, je te dédie ce mémoire avec une profonde gratitude. Tes encouragements constants, ta confiance indéfectible en moi et ton amour inconditionnel ont été les fondations solides sur lesquelles j'ai pu construire ma réussite.

À mon cher père, qui m'a comblé d'amour et m'a donné la force d'être indépendante et de croire en moi-même. Grâce à toi, je suis capable de compter sur moi-même et de relever les défis qui se présentent sur ma route.

À ma famille chérie, mes frères et sœurs qui partagent avec moi des liens indéfectibles. Vous êtes mes compagnons de vie, mes alliés inconditionnels et mes meilleurs amis. Vos encouragements, vos conseils et votre amour constant m'ont accompagné tout au long de ce parcours. Je suis reconnaissante d'avoir grandi à vos côtés, de pouvoir compter sur vous et de partager ensemble tant de moments précieux. Ce mémoire est aussi le vôtre, car il témoigne de notre unité et de notre soutien mutuel.

À mes amis chers, vous êtes mes rayons de soleil dans ma vie. Votre présence joyeuse, votre écoute bienveillante et votre amitié sincère ont illuminé mes journées. Vous m'avez soutenue et encouragée lorsque les défis semblaient insurmontables. Vos rires partagés, nos moments de complicité et nos conversations inspirantes ont rendu cette aventure encore plus spéciale. Ce mémoire est également un hommage à notre amitié solide et durable.

Je vous aime tous infiniment, Merci d'être mes piliers inébranlables.

Cette dédicace est la vôtre, avec tout mon amour.

**Bouzouidja Hadjer**

## **Dédicace**

Avec l'expression de ma reconnaissance, je dédie ce modeste travail à ceux qui, quels que soient les termes embrassés, je n'arriverais jamais à leur exprimer mon amour sincère.

A l'homme, mon précieux offre du dieu, qui doit ma vie, ma réussite et tout mon respect : mon cher père Mohamed.

A la femme, qui a souffert sans me laisser souffrir, qui n'a jamais dit non à mes exigences et qui n'a épargné aucun effort Pour me rendre heureuse : mon adorable mère Khadoudja.

A mes chères sœurs : Fella et Rahil

A mon mari et sa famille

Qui n'ont pas cessé de me conseiller, encourager et soutenir tout au long de mes études, que dieu les protège et leurs offre la chance et le bonheur.

Sans oublier ma binôme Bouzouidja Hadjer pour son soutien moral, sa patience et sa compréhension tout au long de ce projet.

**Benzira Hadjer**

## **Remerciements**

Nous remercions tout d'abord le bon Dieu, le tout puissant de nous avoir armé de force et de courage pour mener à terme ce projet.

Tout d'abord, nous souhaitons adresser nos chaleureux remerciements à notre promoteur, Monsieur Bennila Noureddine. Votre encadrement attentif, vos conseils éclairés et votre expertise ont été d'une valeur inestimable pour nous. Vous avez su guider nos efforts et nous inspirer tout au long de cette étude. Votre soutien constant a été une véritable source de motivation et nous avons énormément appris de votre enseignement.

Un remerciement spécial est également adressé au président du club de robotique MegaTronic, Monsieur Ait Ahmed Mohamed. Votre accueil chaleureux, votre bienveillance et votre ouverture d'esprit ont permis la réalisation de ce projet au sein du club. Nous sommes profondément reconnaissants de sa générosité en mettant à notre disposition les équipements nécessaires, ce qui nous a permis de donner vie à nos idées et de concrétiser notre vision avec succès. Sa contribution précieuse et son soutien logistique ont été des éléments clés dans la réalisation de ce projet. Nous tenons à exprimer notre gratitude sincère envers Monsieur Ait Ahmed pour son engagement et sa confiance en notre équipe.

Enfin, nos remerciements s'adressent à tous les enseignants du Master 2 Automatique et Informatique Industrielle. Votre enseignement rigoureux, vos connaissances approfondies et votre passion pour votre domaine ont été une source d'inspiration tout au long de nos études. Votre engagement envers notre formation a joué un rôle déterminant dans notre croissance académique et professionnelle.

## **Résumé**

Nous avons développé un prototype d'un système de rayonnage intelligent conçu pour optimiser la gestion du stockage pour les entreprises et les organisations. Le système est composé de plusieurs rayons fermés qui maximisent l'utilisation de l'espace. Pour localiser des objets spécifiques, nous avons développé un écran de supervision à l'aide du logiciel Unity, connecté à une base de données SQLite. En entrant simplement le nom de l'objet, le rayon correspondant s'ouvre automatiquement et l'écran de supervision affiche sa position parmi les six emplacements dans chaque rayon. Le système utilise Unity pour une visualisation en temps réel des mouvements des rayons et utilise Arduino avec des moteurs pas à pas Nema 17 et des pilotes A4988 pour le contrôle. Notre projet surmonte des défis tels que la sélection d'une base de données appropriée. Le système de rayonnage intelligent améliore considérablement l'efficacité en ciblant directement et en localisant les objets, ce qui permet une utilisation optimisée de l'espace et réduit le temps de recherche et de gestion des stocks.

## **Abstract**

We have developed a prototype of an intelligent shelving system designed to optimize storage management for businesses and organizations. The system consists of multiple closed shelves that maximize space utilization. To locate specific items, we created a supervisor screen using Unity software, connected to an SQLite database. By simply entering the item name, the corresponding shelf automatically opens, and the supervisor screen displays its position among the six positions in each shelf. The system utilizes Unity for real-time visualization of shelf movements and employs Arduino with Nema 17 steppers and A4988 drivers for control. Our project overcomes challenges such as selecting an appropriate database. The intelligent shelving system significantly improves efficiency by directly targeting and locating items, resulting in optimized space utilization and reduced search and inventory management time.

# Table des matières

<b>Introduction générale.....</b>	<b>1</b>
<b>Chapitre I : Généralité sur le rayonnage mobile .....</b>	<b>1</b>
I.1 Introduction.....	3
I.2 Fonctionnement du rayonnage mobile.....	3
I.3 Avantages du rayonnage mobile .....	4
I.4 Applications du rayonnage mobile.....	5
I.5 Limitations du rayonnage mobile.....	7
I.6 Sécurité.....	7
I.7 Conclusion .....	7
<b>Chapitre II : Composants et logiciels clés du projet .....</b>	<b>1</b>
II.1 Introduction.....	8
II.2 Carte Arduino méga 2560 .....	8
II.2.1 Définition .....	8
II.2.2 Alimentation de la carte Arduino Mega2560.....	9
II.2.3 Programmation .....	9
II.3 Le moteur pas à pas Nema 17 .....	10
II.3.1 Définition .....	10
II.3.2 Fonctionnement .....	10
II.3.3 Caractéristiques.....	10
II.3.4 Justification de choix : .....	11
II.4 Driver A4988.....	12
II.4.1 Définition .....	12
II.4.2 Fonctionnement .....	12
II.5 Bouclier v3.....	13
II.5.1 Description V3 .....	13
II.6 Fin de course .....	14
II.6.1 Définition .....	14
II.7 Les logiciels utilisés pour la réalisation de ce projet .....	15
II.7.1 SolidWorks.....	15
II.7.1.1 Définition.....	15
II.7.1.2 Caractéristiques et fonctionnalités .....	15
II.7.1.3 Application.....	15
II.7.2 Unity .....	16
II.7.2.1 Définition.....	16

II.7.3 Blender .....	17
II.7.3.1 Définition .....	17
II.8 Du prototype à l'implémentation industrielle : Sélection des composants pour une solution de rayonnage intelligent à grande échelle .....	18
II.9 Conclusion .....	19
<b>Chapitre III : "Du virtuel à la réalité : La conception et la mise en œuvre du rayonnage intelligent" .....</b>	<b>1</b>
III.1 Introduction.....	21
III.2 La simulation .....	22
III.2.1 La conception 3D par Blender .....	22
III.2.2 Simulation Unity .....	24
III.2.3 Programmation de la simulation.....	26
III.3 La conception du projet .....	29
III.4 Programmation .....	33
III.4.1 Configuration des composant .....	35
III.4.2 Le fonctionnement .....	38
III.4.3 Algorigramme.....	41
III.5 Supervision .....	42
III.5.1 Conception du superviseur.....	42
III.5.2 La base de données .....	45
III.5.3 La programmation .....	46
III.5.4 Observation des signaux numériques transmis par le superviseur à l'Arduino à l'aide d'un oscilloscope .....	47
III.5.5 Algorigramme.....	51
III.6 Conclusion .....	51
<b>Conclusion générale.....</b>	<b>1</b>

# Liste des figures

Figure I-1 Rayonnage mobile.....	3
Figure I-2 Rayonnage mobile à commande électronique.....	4
Figure I-3 Rayonnage industriel.....	5
Figure I-4 Stockage d'atelier et réserves sur rayonnage mobile .....	5
Figure I-5 Rayonnage mobile de bibliothèque .....	6
Figure I-6 Stockage d'archives sur rayonnage mobile .....	6
Figure II-1 Carte Arduino Mega 2560 .....	8
Figure II-2- Arduino software .....	9
Figure II-3 moteur pas à pas (Nema17).....	10
Figure II-4 driver A4988.....	12
Figure II-5 Bouclier V3.....	14
Figure II-6 fin de course.....	14
Figure II-7 conception d'un rayon par solidworks .....	16
Figure II-8 exemple par unity.....	17
Figure II-9 Interface du Blender.....	18
Figure III-1 le Volant du rayon.....	23
Figure III-2 Le rayon.....	23
Figure III-3 les nœuds utilisés pour avoir une couleur métallique .....	23
Figure III-4 Rayon1 .....	24
Figure III-5 les 3 rayons dans Unity.....	24
Figure III-6 l'ouverture du rayon 1 .....	25
Figure III-7 l'ouverture des rayons 1 et 2.....	25
Figure III-8 l'ouverture des 3 rayons .....	26
Figure III-9 la fermeture des 3 rayons.....	26
Figure III-10 une classe du script.....	27
Figure III-11 Interface Unity.....	28
Figure III-12 la courbe qu'on a utilisé.....	28
Figure III-13 les différentes courbes que Unity offre.....	29
Figure III-14 le dos et l'étagère du rayon.....	30
Figure III-15 liste de boulonnerie.....	30
Figure III-16 la conception finale.....	31
Figure III-17 Aluco découpé par la machine CNC laser .....	31
Figure III-18 Découpage laser.....	32
Figure III-19 support de courroie imprimé en plastique .....	32
Figure III-20 support de courroie brisé .....	32
Figure III-21 Assemblage final .....	33
Figure III-22 circuit du projet.....	35
Figure III-23 shield v3 pinout .....	35
Figure III-24 pins des fins de courses.....	36
Figure III-25 broches du pilote A4988.....	36
Figure III-26 configuration des broches de pas et direction.....	37
Figure III-27 configuration du mode de Microstepping.....	37
Figure III-28 les bibliothèques utilisées .....	38
Figure III-29 ouvrir la communication série .....	39
Figure III-30 condition de la réinitialisation .....	39
Figure III-31 la fonction de réinitialisation .....	40

Figure III-32 Poulie pour moteur pas à pas .....	40
Figure III-33: Interface du Superviseur .....	42
Figure III-34: interface du communication série .....	43
Figure III-35: interface d'ajout d'un objet.....	43
Figure III-36:L'affichage du placement de l'objet.....	44
Figure III-37:Interface de recherche.....	44
Figure III-38: Indication visuelle de la position de l'objet recherché .....	44
Figure III-39: commande de la base de données .....	46
Figure III-40: programmation du bouton de rafraichissement .....	46
Figure III-41: ouverture la communication série.....	47
Figure III-42: commande pour trouver l'objet recherché dans la base de données .....	47
Figure III-43 L'utilisation de l'oscilloscope.....	48
Figure III-44 Code Ascii .....	49
Figure III-45 La séquence binaire correspond au caractère "1" .....	49
Figure III-46 La séquence binaire correspond au caractère "2" .....	50
Figure III-47 La séquence binaire correspond au caractère "3" .....	50
Figure III-48 La séquence binaire correspond au caractère "0" .....	50

## Liste des Algorigrammes

Algorigramme III-1 Algorigramme du fonctionnement de la simulation .....	27
Algorigramme III-2: algorigramme du code Arduino.....	41
Algorigramme III-3: Résumé de la logique de fonctionnement des scripts de Unity .....	51

## Liste des Tableaux

Tableau III-1 les modes du microstepping .....	38
--	----

## **Introduction générale**

## **Introduction :**

Dans un monde en constante évolution, l'efficacité et la productivité constituent des enjeux majeurs pour les entreprises. Cependant, la gestion du temps et des ressources peut souvent représenter un défi de taille, notamment lorsqu'il s'agit de la manipulation et de l'organisation des stocks. La perte de temps et d'efforts liée à des processus de rangement et de recherche inefficaces peut avoir des conséquences importantes sur la rentabilité et la compétitivité des entreprises.

C'est dans ce contexte que s'inscrit le travail que nous allons développer, visant à optimiser la gestion des stocks et à minimiser les pertes de temps. Notre objectif est de concevoir un système de rayonnage automatisé capable de localiser et de récupérer rapidement les articles demandés, tout en garantissant une utilisation efficace de l'espace de stockage.

Pour atteindre cet objectif, nous avons choisi d'utiliser une combinaison de programmation Arduino et de supervision par logiciel Unity. L'utilisation d'Arduino, une plateforme électronique open-source facilement adaptable à de nombreux projets, nous permettra de contrôler les différents composants matériels du système de rayonnage, tandis que Unity nous offrira un environnement graphique convivial pour la visualisation et la gestion du système, ce qui est essentiel pour assurer une utilisation fluide et efficace.

Dans ce mémoire, nous présenterons tout d'abord les problématiques rencontrées dans la gestion traditionnelle des stocks, en mettant particulièrement l'accent sur les pertes de temps et d'efforts. Ensuite, nous explorerons une étape de simulation préliminaire réalisée par Unity, qui nous permettra de modéliser le système de rayonnage, d'évaluer différentes stratégies de gestion et d'obtenir une meilleure compréhension du système. Puis, nous examinerons les avantages de l'utilisation de la programmation Arduino pour contrôler les composants matériels du système de rayonnage. Enfin, nous soulignerons l'importance du logiciel Unity en tant que superviseur centralisé, permettant la visualisation en temps réel et la gestion efficace de l'ensemble du système.

Nous espérons que ce mémoire contribuera à une meilleure compréhension des enjeux liés à la gestion des stocks et à la mise en place de solutions innovantes pour optimiser l'efficacité opérationnelle des entreprises.

Ce mémoire s'organise de la manière suivante :

### **Chapitre I : généralité sur le rayonnage mobile**

Contrairement aux rayonnages fixes traditionnels, les rayonnages mobiles sont montés sur des rails ou des chariots qui leur permettent de se déplacer latéralement. Cela permet d'optimiser l'utilisation de l'espace, d'améliorer l'accessibilité, d'accroître la sécurité et de répondre aux besoins spécifiques de chaque entreprise.

### **Chapitre II : Composants et logiciels clés du projet**

Ce chapitre offre un aperçu concis des composants matériels et des logiciels clés utilisés dans le projet, soulignant leur importance et leur contribution à sa réalisation globale.

### **Chapitre III : Du virtuel à la réalité : La conception et la mise en œuvre du rayonnage intelligent**

Dans ce chapitre, nous explorons en détail les différentes étapes qui ont été suivies : la simulation, la conception, la programmation et la supervision. Nous mettons en évidence l'importance de chaque étape dans la réalisation d'un système fonctionnel et performant.

## **Chapitre I : Généralité sur le rayonnement mobile**

## **I.1 Introduction**

De nombreuses organisations sont confrontées à une série de défis; chercher des objets parmi des rayons encombrés, ce qui entraîne une perte de temps et d'effort considérable. C'est une problématique courante qui entrave la productivité et l'efficacité.

Le rayonnage mobile est une technologie qui utilise des capteurs et des algorithmes pour optimiser le stockage et la récupération des marchandises dans un entrepôt. La technologie optimise l'espace de stockage, réduit le temps de préparation des commandes et augmente la productivité globale de l'entrepôt. (Figure 1.1)



*Figure I-1 Rayonnage mobile*

## **I.2 Fonctionnement du rayonnage mobile**

Le rayonnage mobile fonctionne en utilisant des capteurs pour surveiller l'inventaire et les besoins en produits. Les données collectées sont ensuite analysées par des algorithmes pour déterminer le meilleur emplacement pour chaque produit.

Les rayonnages peuvent être automatisés afin de déplacer les produits vers les emplacements préconisés, ce qui diminue les erreurs humaines et raccourcit le temps nécessaire pour trouver les produits. (Figure 1.2)



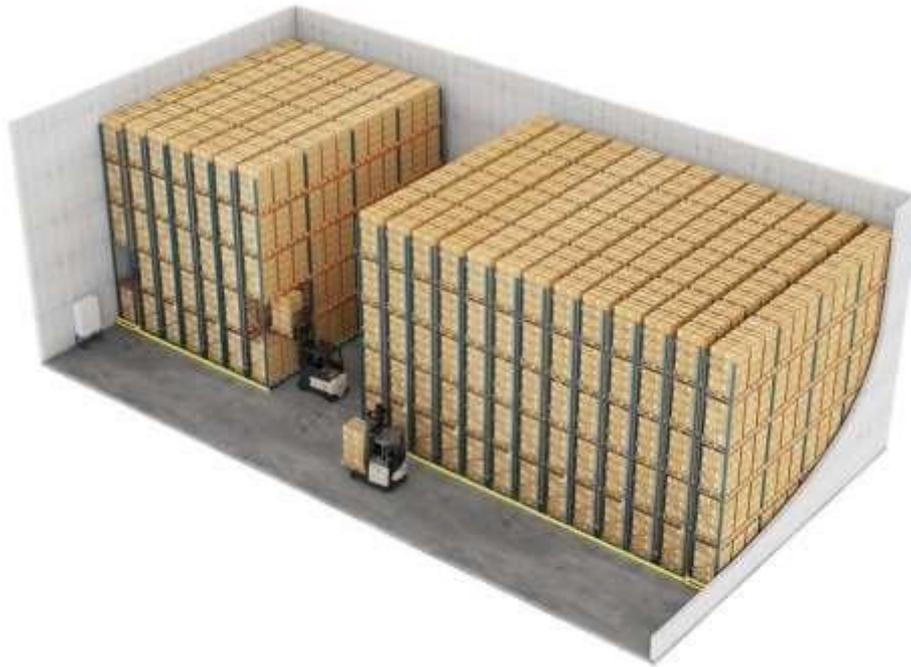
*Figure I-2 Rayonnage mobile à commande électronique*

### **I.3 Avantages du rayonnage mobile**

Le rayonnage mobile offre de nombreux avantages pour les entreprises, notamment une utilisation plus efficace de l'espace de stockage, une réduction des temps de préparation des commandes et une amélioration de la précision des stocks.

En outre, cette technologie peut aider à réduire les coûts de main-d'œuvre et à améliorer la sécurité des travailleurs en éliminant la nécessité de monter sur des échelles pour atteindre des produits en hauteur.

De plus, le rayonnage mobile permet un accès facile aux produits stockés, ce qui peut améliorer l'efficacité du stockage et réduire les temps d'attente pour les clients. (Figure 1.3)



*Figure I-3 Rayonnage industriel*

## **I.4 Applications du rayonnage mobile**

Le rayonnage mobile est utilisé dans une variété d'industries, notamment la logistique, la distribution, la fabrication et le commerce de détail. (Figure 1.4)



*Figure I-4 Stockage d'atelier et réserves sur rayonnage mobile*

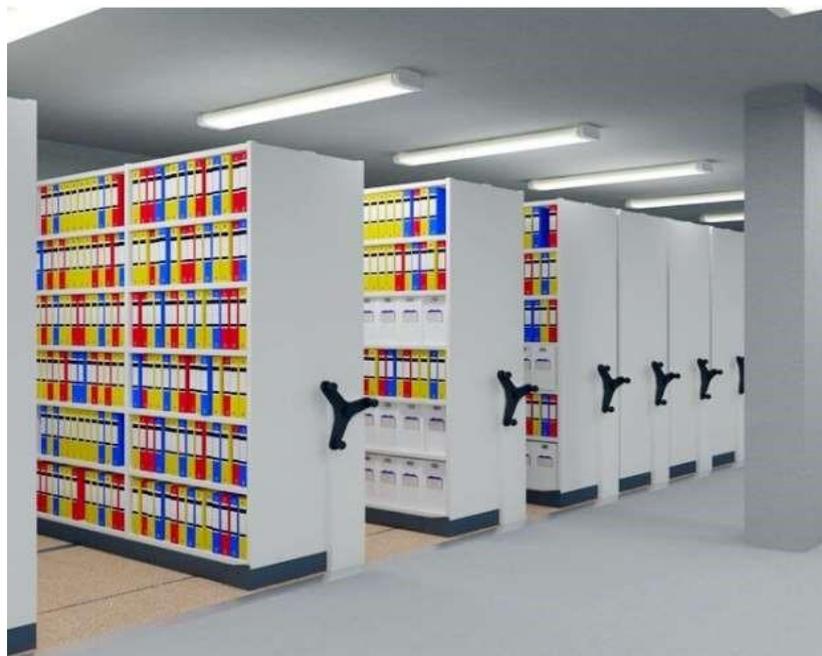
Aussi, il est utilisé dans une variété d'applications, notamment dans les entrepôts, les bibliothèques, les archives et les musées. (Les figures suivantes).



*Figure I-5 Rayonnage mobile de bibliothèque*

Ce système de rangement pour bibliothèques et médiathèques est à la fois mobile et statiques, possède de nombreux avantages :

- ✓ Réduire les coûts administratifs grâce à une utilisation plus efficace de l'espace.
- ✓ Regrouper l'équipement et les dossiers et en améliorer l'organisation, le contrôle et la sécurité.
- ✓ Réduire les frais de maintenance.



*Figure I-6 Stockage d'archives sur rayonnage mobile*

Cette technologie est particulièrement utile pour les entreprises qui ont besoin de stocker de grandes quantités de produits différents dans un espace limité, ou qui doivent traiter des commandes rapidement et avec précision.

## **I.5 Limitations du rayonnage mobile**

Bien que le rayonnage mobile offre de nombreux avantages, il présente également certaines limites. Par exemple, cette technologie peut être coûteuse à mettre en place et à entretenir, ce qui peut rendre son adoption difficile pour les petites entreprises.

De plus, le rayonnage mobile peut ne pas être adapté à toutes les situations de stockage, en particulier pour les produits volumineux ou fragiles qui nécessitent une manipulation manuelle.

## **I.6 Sécurité**

La sécurité est une considération importante lors de l'utilisation du rayonnage mobile. Les chariots doivent être équipés de capteurs pour détecter les obstacles et les personnes.

De plus, les travailleurs doivent être formés à l'utilisation sûre du système et les étagères doivent être correctement chargées pour éviter les accidents.

## **I.7 Conclusion**

En somme, le rayonnage mobile offre une solution prometteuse pour optimiser le stockage, la récupération et l'efficacité des entrepôts. Cependant, il convient de peser soigneusement les avantages, les limitations et les considérations de sécurité avant de décider de mettre en œuvre cette technologie. En restant conscient des défis potentiels et en adoptant une approche proactive, les organisations peuvent exploiter tout le potentiel du rayonnage mobile pour améliorer leurs opérations et leur compétitivité.

## **Chapitre II : Composants et logiciels clés du projet**

## II.1 Introduction

Ce chapitre met en lumière les équipements essentiels qu'on a sélectionnés avec soin afin de garantir le succès du projet. Parmi ceux-ci, on a choisi d'incorporer des capteurs de fin de course pour assurer la sécurité et déterminer la position initiale. On a opté pour les moteurs pas à pas Nema 17 en raison de leur fiabilité et de leur précision inégalée. De plus, on a fait appel au contrôleur Arduino Mega, reconnu pour ses performances exceptionnelles, avec son microcontrôleur ATmega 2560.

Pour la concrétisation du projet de rayonnage intelligent, on a également exploré divers logiciels tels que SolidWorks, Unity et Blender. Ces outils puissants nous ont permis de concevoir, simuler et développer un superviseur, offrant ainsi une expérience immersive. Ces avancées technologiques nous ont ouvert de nouvelles perspectives pour réaliser le projet.

## II.2 Carte Arduino méga 2560

### II.2.1 Définition

L'Arduino Mega 2560 est une carte microcontrôleur basée sur l'ATmega 2560 (Figure 2.1).

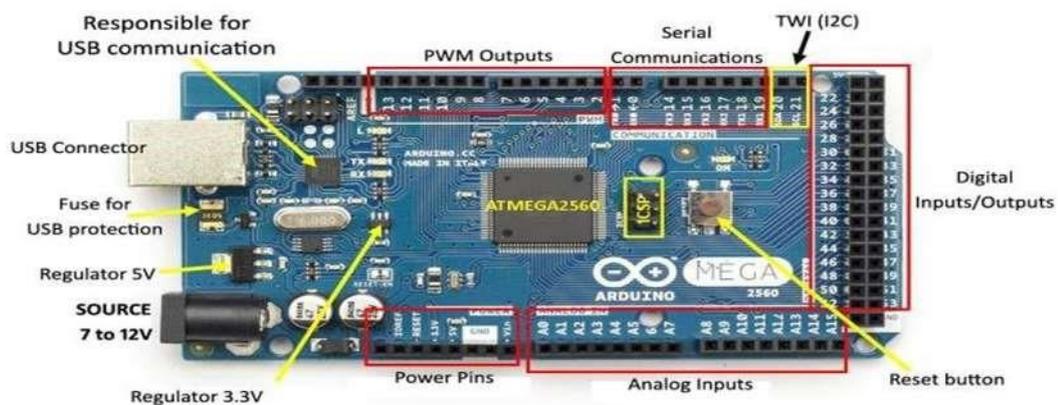


Figure II-1 Carte Arduino Mega 2560

Il dispose :

- 54 broches d'entrée/sortie numériques (dont 15 peuvent être utilisées comme sorties PWM).
- 4 UART (port série matériel).
- 16 entrées analogiques.
- Une connexion USB
- Une prise d'alimentation

- Un oscillateur à cristal 16 MHz
- Un en-tête ICSP
- Un bouton de réinitialisation (Reset)

Il contient tout le nécessaire pour prendre en charge le microcontrôleur ; connectez-le simplement à un ordinateur avec un câble USB ou alimentez-le avec un adaptateur AC-DC ou une batterie pour commencer. La carte Mega 2560 est compatible avec la plupart des shields conçus pour la Uno et les anciennes cartes Duemilanove ou Diecimila. [1]

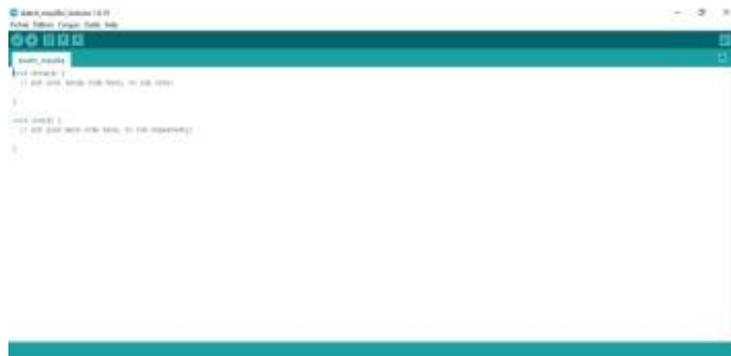
### II.2.2 Alimentation de la carte Arduino Mega2560

La carte Arduino Mega 2560 peut être alimentée soit via la connexion USB, soit à l'aide d'une alimentation externe branchée sur le connecteur jack de la carte. La source d'alimentation est sélectionnée automatiquement par la carte. La carte fournit ainsi les tensions de 5V et 3.3V pour alimenter d'autres modules par exemple.

### II.2.3 Programmation

La carte Arduino Mega peut être programmée à l'aide de l'IDE Arduino, un environnement de développement intégré basé sur Processing. L'IDE Arduino utilise un langage de programmation simplifié basé sur C++.

Les programmes sont téléchargés sur la carte via le port USB ou le connecteur ICSP. La carte peut être programmée pour effectuer une grande variété de tâches, telles que la commande de moteurs, la collecte de données de capteurs, la communication sans fil et bien plus encore. (Figure2.2)



Arduino software2-II Figure

## II.3 Le moteur pas à pas Nema 17

### II.3.1 Définition

Un moteur pas à pas est un moteur brushless, qui se commande comme son nom l'indique pas par pas, avec précision. Comme tout moteur électrique, le moteur pas à pas sert de champ magnétique pour transformer l'énergie électrique en énergie mécanique. Il possède 4 demi-bobines, câblées en parallèles 2 à 2 pour former 2 bobines. [2]

Le nom Nema17 fait référence aux dimensions du moteur - il a une largeur de 1,7 pouces (43,2 mm). Il existe également d'autres tailles de moteurs Nema, telles que le Nema14, le Nema23 et le Nema34. (Figure 2.3)



Figure II-3 moteur pas à pas (Nema17)

### II.3.2 Fonctionnement

Le moteur pas à pas Nema17 fonctionne en envoyant des impulsions électriques à ses bobines, qui créent un champ magnétique qui attire le rotor du moteur vers la position désirée. Les impulsions sont envoyées par un contrôleur, qui peut être programmé pour déterminer la vitesse et la direction du mouvement.

### II.3.3 Caractéristiques

Le moteur Nema 17 présente plusieurs caractéristiques qui en font un choix populaire pour de nombreuses applications. Voici certaines des caractéristiques clés du moteur Nema 17 :

- Taille : Le moteur Nema 17 appartient à la famille des moteurs pas à pas et tire son nom de sa dimension standard de 1,7 pouce (43,18 mm) pour la largeur du corps. Sa taille compacte facilite son intégration dans des espaces restreints et offre une grande flexibilité de conception.

- **Couple :** Le moteur Nema 17 est réputé pour son couple élevé par rapport à sa taille. Il peut générer un couple significatif, permettant de gérer des charges importantes tout en conservant une précision élevée dans le positionnement.
- **Pas :** Le moteur Nema 17 a généralement un pas de 1,8 degré, ce qui signifie qu'il effectue une rotation de 1,8 degré à chaque impulsion électrique. Ce pas fin permet un contrôle précis de la position et facilite la réalisation de mouvements précis.
- **Tension et Courant :** Les moteurs Nema 17 sont disponibles dans différentes variantes avec des tensions nominales allant de 2,8 V à 4,2 V. Le courant nominal varie également en fonction du modèle, allant généralement de 0,5 A à 1,5 A.
- **Inductance :** Les moteurs Nema 17 ont une inductance relativement basse, ce qui leur confère une réponse dynamique rapide et facilite leur utilisation à des vitesses élevées.
- **Fiabilité :** Les moteurs Nema 17 sont réputés pour leur fiabilité et leur durabilité. Ils sont conçus pour fonctionner pendant de longues périodes sans surchauffe, ce qui en fait un choix fiable pour des applications industrielles.
- **Facilité de contrôle :** Les moteurs Nema 17 sont compatibles avec des contrôleurs de moteur pas à pas standard tels que le A4988 ou le DRV8825. Ils peuvent être facilement contrôlés en envoyant des signaux d'impulsion et de direction, ce qui facilite leur intégration dans des systèmes électroniques.

### **II.3.4 Justification de choix :**

Nous avons choisi le moteur Nema 17 pour notre projet en raison de ses caractéristiques exceptionnelles qui répondent parfaitement à nos besoins spécifiques. Dans notre conception, nous utilisons des rayons en verre acrylique d'une épaisseur de 4 mm, ce matériau ayant un poids d'environ 6 kg/m<sup>2</sup>. Chaque rayon vide a un poids d'environ 8,34 kg. De plus, nous avons la possibilité d'ajouter jusqu'à 6 livres sur chaque rayon. Le poids des livres peut varier entre 100 g et 300 g, selon leur épaisseur. Par conséquent, le poids total d'un rayon peut atteindre environ 10,14 kg.

Nous avons pris en compte la plage typique de couple de maintien des moteurs Nema 17, qui se situe entre 0,4 Nm et 0,6 Nm (40 kg-cm et 60 kg-cm). Sur la base de ces spécifications, le modèle Nema 17 17HS16-2004S devrait fournir un couple suffisant pour supporter un poids de 8 kg.

## II.4 Driver A4988

### II.4.1 Définition

Le driver moteur pas à pas A4988 est un circuit intégré utilisé pour contrôler les moteurs pas à pas. Il est souvent utilisé dans des projets d'automatisation, de robotique et d'impression 3D.

Le driver A4988 permet un contrôle précis du mouvement du moteur pas à pas en ajustant le courant qui passe à travers les bobines. Il peut également détecter les erreurs de mouvement et protéger le moteur contre les surcharges. (Figure 2.4)



Figure II-4 driver A4988

### II.4.2 Fonctionnement

Le A4988 est un pilote de moteur pas à pas couramment utilisé dans de nombreuses applications. Il est capable de contrôler des moteurs pas à pas bipolaires, offrant des fonctionnalités de micro-pas et de régulation du courant. Voici un aperçu de son fonctionnement :

- Alimentation électrique : Le A4988 nécessite une alimentation électrique distincte pour fournir la tension et le courant nécessaires au moteur pas à pas. Cette alimentation doit être adaptée aux spécifications du moteur.
- Alimentation électrique de la logique : Le A4988 nécessite également une alimentation électrique de la logique, généralement d'environ 5V, pour alimenter sa circuiterie interne.
- Contrôle du pas et de la direction : Le A4988 accepte des signaux de pas et de direction pour contrôler le mouvement du moteur pas à pas. Le signal de pas détermine les incréments par lesquels le moteur se déplace, tandis que le signal de direction spécifie la direction de rotation. Ces signaux sont généralement fournis par un microcontrôleur ou tout autre dispositif capable de générer des signaux numériques.

- Impulsion de pas : Pour déplacer le moteur pas à pas, le A4988 nécessite une série d'impulsions de pas. Chaque impulsion correspond à un micro-pas, qui est un incrément de mouvement plus petit qu'un pas complet. La fréquence de ces impulsions de pas détermine la vitesse du moteur.
- Temporisation des impulsions : La durée de chaque impulsion de pas est essentielle pour un bon fonctionnement. Le A4988 dispose d'une horloge interne qui définit la temporisation des impulsions. Vous pouvez configurer le micro-pas en définissant différentes combinaisons de broches (MS1, MS2 et MS3) sur le pilote. Cela vous permet de contrôler le nombre de micro-pas par pas complet, allant du mode pas complet (1 micro-pas par pas complet) à des résolutions de micro-pas plus élevées (par exemple, 1/2, 1/4, 1/8, 1/16, etc.).
- Régulation du courant : Le A4988 permet également de réguler le courant en fixant le courant maximal qui circule dans les enroulements du moteur pas à pas. Cela se fait en ajustant la tension de référence à l'aide d'un potentiomètre sur le circuit du pilote.

En configurant correctement les signaux de pas et de direction, la temporisation des impulsions, la résolution du micro-pas, la régulation du courant et le mode de désactivation, on peut contrôler la vitesse, la position et le couple du moteur pas à pas à l'aide du pilote A4988.

## **II.5 Bouclier v3**

### **II.5.1 Description V3**

Le bouclier V3, (figure 2.5) également connu sous le nom de "Arduino Motor Shield Rev3", est un bouclier d'extension conçu spécifiquement pour les cartes Arduino. Il est utilisé pour faciliter le contrôle des moteurs à courant continu (DC) et des moteurs pas à pas, offrant une solution pratique pour les projets de robotique et d'automatisation. [3]

Voici une description des principales caractéristiques et fonctionnalités du bouclier V3:

- Support des moteurs pas à pas : Le bouclier V3 prend également en charge le contrôle des moteurs pas à pas. Il dispose de quatre prises pour connecter facilement des moteurs pas à pas bipolaires.

- Protection de courant : Le bouclier V3 est doté d'un circuit de protection de courant intégré qui limite le courant fourni aux moteurs, ce qui protège le bouclier et les moteurs contre les surcharges et les courts-circuits.
- Connecteurs pratiques : Le bouclier V3 est équipé de borniers à vis qui permettent de connecter facilement les fils des moteurs. Il dispose également de broches de connexion pour les entrées/sorties supplémentaires, ce qui offre une grande flexibilité dans la configuration du système.
- Compatibilité Arduino : Le bouclier V3 est conçu spécifiquement pour être compatible avec les cartes Arduino, telles que l'Arduino Uno, Méga et d'autres modèles compatibles. Il se connecte simplement sur le dessus de la carte Arduino, facilitant ainsi l'ajout de fonctionnalités de contrôle des moteurs.



Figure II-5 Bouclier V3

## II.6 Fin de course

### II.6.1 Définition

Les fins de course (figure 2.6) est un concept important dans la mécanique et l'ingénierie. Il s'agit d'un dispositif qui permet de limiter le mouvement d'un objet à une position donnée.



Figure II-6 fin de course

Les fins de course sont utilisés pour but de sécurité et pour réinitialiser la position des rayons (tous les rayons fermés ; position 0) sinon on peut programmer

le mouvement des rayons (l'ouverture / fermeture) par commander le moteur pas à pas et programmer les positions par des pas.

## **II.7 Les logiciels utilisés pour la réalisation de ce projet**

### **II.7.1 SolidWorks**

#### **II.7.1.1 Définition**

Solidworks est un logiciel de conception assistée par ordinateur, appartenant à la société Dassault Systèmes. Il utilise le principe de conception paramétrique et génère trois types de fichiers qui sont liés : la pièce, l'assemblage, et la mise en plan. Il est largement utilisé par les ingénieurs, les concepteurs industriels et d'autres professionnels pour créer des modèles 3D, effectuer des simulations et générer des dessins techniques. [4]

#### **II.7.1.2 Caractéristiques et fonctionnalités**

Les fonctionnalités clés de SolidWorks comprennent :

- **Modélisation 3D** : SolidWorks permet de créer des modèles 3D détaillés à l'aide d'outils de modélisation puissants. Les utilisateurs peuvent créer des pièces individuelles, des assemblages de pièces et des ensembles de composants complexes en utilisant des fonctionnalités de base telles que l'esquisse, les opérations de base (extrusion, révolution, etc.) et des fonctionnalités avancées (bosses, découpes, nervures, etc.).
- **Assemblage** : SolidWorks facilite la création d'assemblages en permettant aux utilisateurs de positionner et de connecter les composants entre eux de manière réaliste. Les contraintes et les relations peuvent être appliquées pour simuler les mouvements et les interactions entre les pièces.
- **Simulation** : SolidWorks propose des outils de simulation intégrés qui permettent aux utilisateurs d'effectuer des analyses virtuelles des pièces et des assemblages. Cela inclut des fonctionnalités pour la simulation de contraintes, la simulation thermique, la dynamique des fluides, l'analyse des mouvements, etc.

#### **II.7.1.3 Application**

SolidWorks est utilisé dans de nombreux secteurs industriels tels que l'ingénierie mécanique, l'aérospatiale, l'automobile, l'électronique, le génie civil, etc. Il aide les concepteurs et les ingénieurs à transformer leurs idées en modèles précis, à optimiser les performances des produits et à accélérer le processus de développement.

Nous avons opté pour SolidWorks, pour créer un rayonnage sur mesure (Figure 2.9) qui facilite grandement le processus de découpe CNC, car il prend en charge plusieurs formats de fichiers compatibles avec les machines CNC, ce qui nous a permis de transférer facilement les données de conception sans perte de précision. Avec SolidWorks, on a pu donner vie à notre vision en concevant un modèle 3D précis et détaillé du rayonnage.

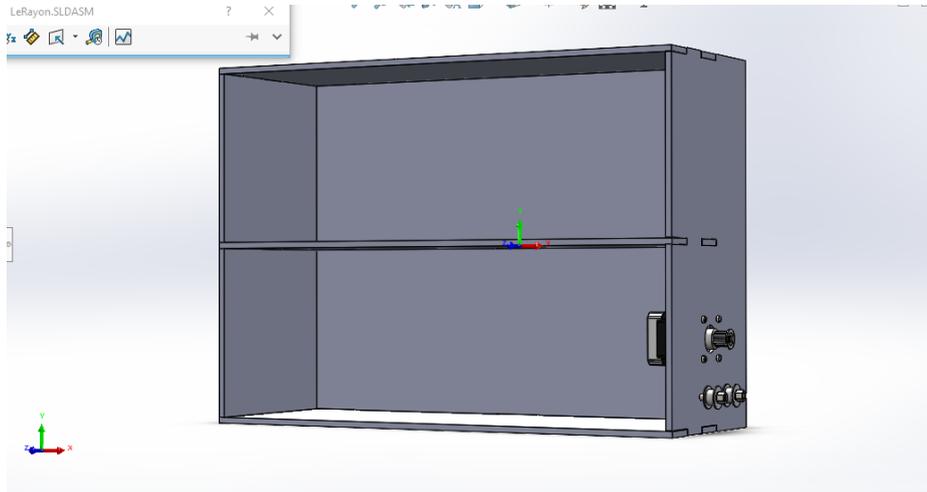


Figure II-7 conception d'un rayon par solidworks

## II.7.2 Unity

### II.7.2.1 Définition

Unity est la première plate-forme de développement 3D en temps réel au monde, offrant tout ce dont vous avez besoin pour créer et développer des jeux et des expériences incroyables sur la plus large gamme de plates-formes du secteur. [5]

Les outils de simulation 2D ou 3D de unity offrent aux développeurs les moyens des simulations interactives en exploitant des fonctionnalités physiques avancées.

Unity est utilisé dans divers domaines professionnels, ainsi que l'architecture et le design, la formation et l'éducation ; comme le logiciel Factory I/O, industrie automobile pour crée des simulations de conduite, des expériences de réalité augmentée pour les ventes et la configuration des voitures, il est aussi utilisé dans le secteur de santé et le marketing. (Figure 2.10)

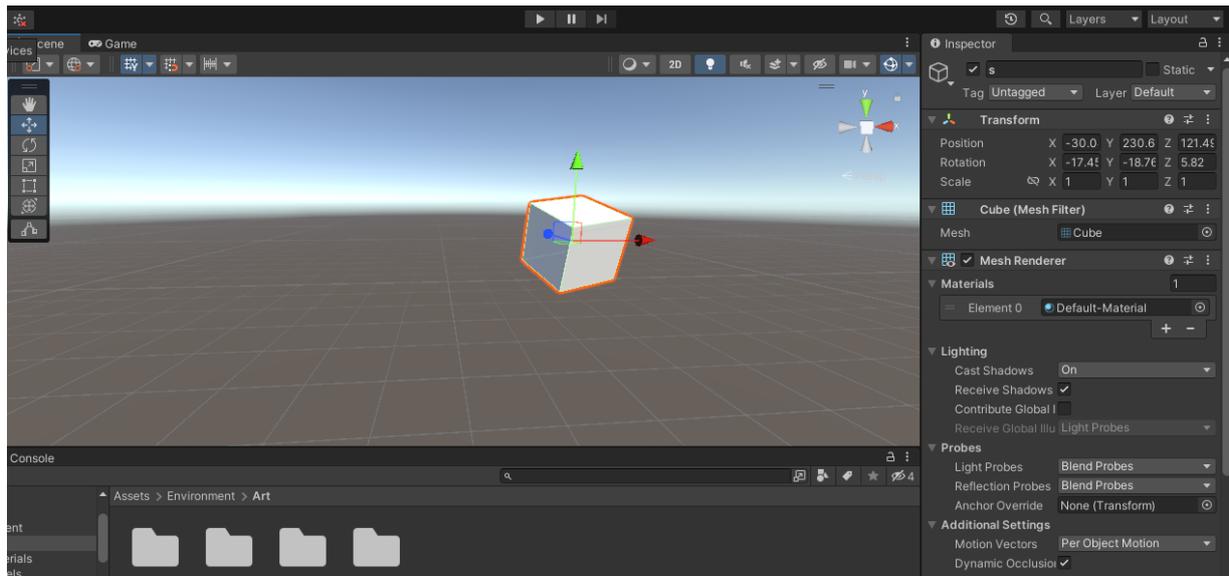


Figure II-8 exemple par unity

On a opté pour l'utilisation de ce logiciel pour réaliser la simulation et développer le superviseur. Grâce aux fonctionnalités puissantes de ce logiciel, on a pu créer des simulations réalistes qui nous ont permis de visualiser et de valider le fonctionnement du système.

## II.7.3 Blender

### II.7.3.1 Définition

Blender est un logiciel de modélisation, d'animation et de rendu 3D créé en 2 janvier 1994.

Il prend en charge l'intégralité du pipeline 3D : modélisation, rigging, animation, simulation, rendu, composition et suivi de mouvement, même montage vidéo et création de jeux. Les utilisateurs avancés utilisent l'API de Blender pour les scripts Python pour personnaliser l'application et écrire des outils spécialisés. [6] (Figure 2.11).

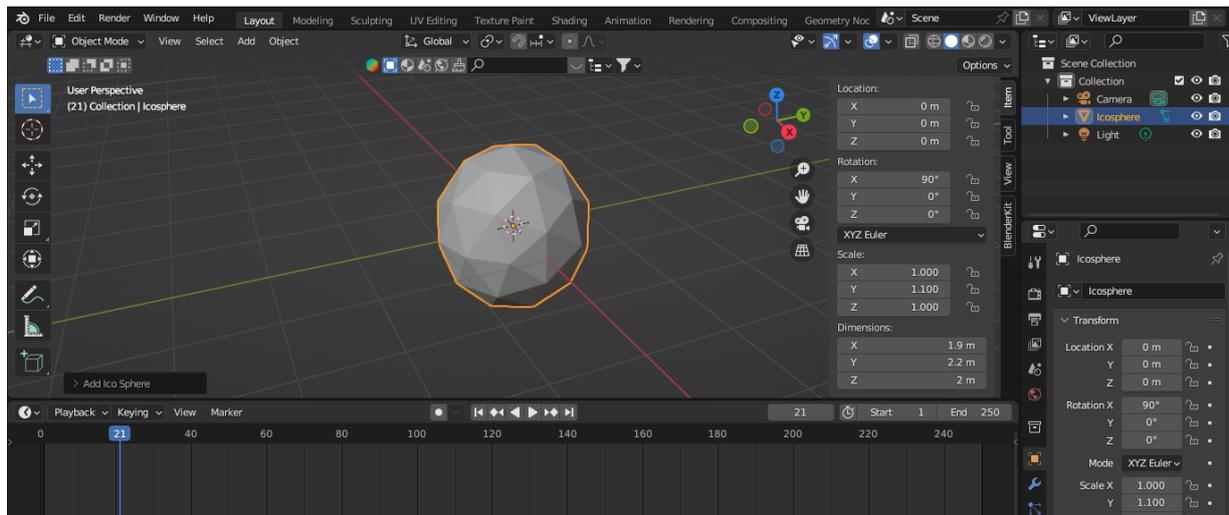


Figure II-9 Interface du Blender

On a choisi Blender pour réaliser une conception en 3D des rayons afin de les intégrer de manière saisissante dans le superviseur.

## II.8 Du prototype à l'implémentation industrielle : Sélection des composants pour une solution de rayonnement intelligent à grande échelle

Pour réaliser ce projet, on a utilisé des équipements tels que le plexiglas pour les rayons, l'Arduino comme contrôleur, le moteur Nema 17 avec son driver, ainsi que des fins de course pour la détection. Ces composants nous ont permis de créer un prototype parfaitement fonctionnel, démontrant ainsi la faisabilité du concept.

Cependant, pour réaliser ce projet à grande échelle et répondre aux normes mondiales, il est nécessaire d'utiliser d'autres équipements plus adaptés. En effet, lors de la conception formelle du système pour des entreprises, il est important de choisir des alternatives qui répondent aux exigences de robustesse, de fiabilité et de compatibilité avec les standards industriels.

Pour la construction des rayons, des matériaux tels que l'acier ou l'aluminium peuvent être préférés, offrant une plus grande résistance et durabilité. De même, les moteurs utilisés peuvent être des modèles plus puissants et adaptés aux charges de stockage prévues dans un environnement commercial.

Le choix du moteur dépend principalement du poids qu'il peut supporter. Dans le domaine industriel, il existe différentes alternatives aux moteurs pas à pas Nema 17, conçues pour répondre à des exigences spécifiques en termes de capacité de charge. Par exemple, le

moteur pas à pas Nema 34 offre une puissance supérieure au Nema 17, ce qui le rend plus adapté aux charges plus lourdes ou aux applications nécessitant un couple plus élevé. De plus, il existe des moteurs pas à pas linéaires qui intègrent un mouvement linéaire directement dans le moteur, éliminant ainsi le besoin d'un mécanisme de conversion de mouvement.

De plus, lors de la conception pour une utilisation en entreprise, il est important de considérer les exigences de sécurité, d'évolutivité et de maintenance. Des systèmes de détection plus avancés, tels que des capteurs de proximité ou des technologies de vision, peuvent être intégrés pour une précision accrue dans la localisation des objets. Les logiciels de gestion de stock et de supervision peuvent également être adaptés pour répondre aux besoins spécifiques de l'entreprise.

Les contrôleurs peuvent être des systèmes industriels spécialisés, offrant des fonctionnalités avancées de contrôle et de connectivité, comme les contrôleurs programmables (PLC); Les contrôleurs logiques programmables sont largement utilisés dans l'automatisation industrielle. Ils offrent une grande flexibilité et sont capables de gérer des tâches complexes de contrôle et de surveillance.

Pour conclure, ce prototype fonctionnel a permis de valider l'idée du rayonnage intelligent, mais pour une implémentation à grande échelle dans le cadre d'une entreprise, il est nécessaire d'adopter des équipements répondant aux normes industrielles mondiales. Ce choix garantit une performance optimale, une compatibilité élargie et une intégration réussie du système dans les environnements professionnels.

## **II.9 Conclusion**

En conclusion, les équipements sélectionnés ont constitué une base solide pour la réalisation de notre projet, mais il est important de rester ouvert aux possibilités d'amélioration et d'adaptation pour répondre aux évolutions du domaine. Avec une approche proactive et une volonté constante d'innover, on peut continuer à repousser les limites et à créer des solutions toujours plus performantes et efficaces.

Cependant, il est important de noter que ces équipements peuvent être remplacés par d'autres plus adaptés si on souhaite réaliser le projet à plus grande échelle. Selon les besoins et les contraintes spécifiques, il peut être judicieux de choisir des capteurs, des moteurs et des contrôleurs plus avancés pour répondre aux exigences du projet.

## **Chapitre III : "Du virtuel à la réalité : La conception et la mise en œuvre du rayonnage intelligent"**

### **III.1 Introduction**

Ce chapitre présente la réalisation du projet de rayonnage intelligent, qui a été réalisé en suivant plusieurs étapes clés. L'objectif principal était de concevoir un système automatisé de rayonnage. Pour atteindre cet objectif, on a utilisé des outils et des technologies modernes, allant de la simulation par logiciel Unity à la création d'une conception 3D par SolidWorks, en passant par la mise en place d'une partie commande avec Arduino, et enfin le développement d'un superviseur pour le contrôle global du système.

Le premier volet de notre travail a consisté à simuler le fonctionnement du rayonnage intelligent à l'aide du logiciel Unity. Cette simulation nous a permis de tester différentes configurations et d'optimiser les performances du système avant de passer à la phase de conception réelle. Grâce à cette approche, on a pu évaluer et affiner notre modèle, en tenant compte de diverses contraintes et exigences spécifiques au projet.

Ensuite, on a utilisé SolidWorks pour créer une conception 3D détaillée du rayonnage intelligent. Cette étape nous a permis d'explorer différentes options de conception, d'intégrer les composants mécaniques et électroniques nécessaires, et de vérifier l'assemblage du système dans son ensemble. La modélisation 3D nous a offert une vision claire et précise du rayonnage intelligent, facilitant ainsi la fabrication et l'assemblage ultérieurs.

Une fois la conception validée, on a procédé à la réalisation de la partie commande du système à l'aide de la plateforme Arduino. Cette phase a impliqué la programmation des différents capteurs, actionneurs et modules de communication nécessaires pour assurer le bon fonctionnement du rayonnage intelligent. Grâce à Arduino, on a pu mettre en œuvre les fonctionnalités clés du système, telles que la détection des objets, le déplacement des rayons et la communication avec le superviseur.

Enfin, on a développé un superviseur pour le contrôle global du système de rayonnage intelligent. Ce superviseur permet de surveiller en temps réel l'état du rayonnage, de gérer les commandes et de générer des rapports statistiques sur les stocks. Il offre également une interface conviviale pour les opérateurs, leur permettant d'interagir facilement avec le système et d'effectuer des tâches de maintenance ou de dépannage.

Tout au long de ce chapitre, on mettra en valeur le travail accompli à chaque étape du processus de réalisation du projet de rayonnage intelligent. Nous soulignerons les défis rencontrés, les solutions mises en œuvre et les résultats obtenus, en mettant l'accent sur les avantages et la valeur ajoutée de ce système automatisé.

## **III.2 La simulation**

Dans cette section, nous explorons en détail la première étape cruciale de la réalisation du projet : la simulation à l'aide du logiciel Unity. Cette approche novatrice nous a permis d'optimiser et de préciser notre concept avant de passer à la phase de conception réelle.

La simulation avec Unity offre de nombreux avantages significatifs pour la conception de systèmes complexes tels que le rayonnage intelligent. En effet, elle permet de créer un environnement virtuel réaliste où nous pouvons tester différentes configurations et évaluer les performances du système. Grâce à la simulation, on a pu explorer diverses options et ajuster les paramètres pour atteindre une efficacité et une précision optimales.

De plus, l'un des principaux bénéfices de la simulation réside dans sa capacité à réduire les coûts et les erreurs. En simulant le fonctionnement du rayonnage intelligent, on a pu identifier et résoudre les problèmes potentiels à un stade précoce, sans avoir à investir dans des prototypes physiques coûteux.

En fin, la simulation avec Unity a facilité le processus de conception ultérieur. Les informations et les observations recueillies lors de la simulation ont permis de créer une conception 3D plus précise et détaillée à l'aide de SolidWorks. Cette transition en douceur entre la simulation virtuelle et la conception réelle a permis une meilleure compréhension des exigences techniques et une intégration plus fluide des composants mécaniques et électroniques.

### **III.2.1 La conception 3D par Blender**

Tout d'abord, la première étape a été de créer des modèles 3D détaillés et personnalisés des rayons, en utilisant le logiciel Blender, on a pu ajouter des détails spécifiques, comme les étagères, le volant, les numéros (Figures 3.1 , 3.2), ainsi que choisir des couleurs pour une représentation visuelle plus réaliste grâce à l'utilisation des nœuds dans Blender qui permettent de créer des matériaux complexes et d'appliquer des textures, des reflets et d'autres attributs visuels aux surfaces des objets. (Figure 3.3)

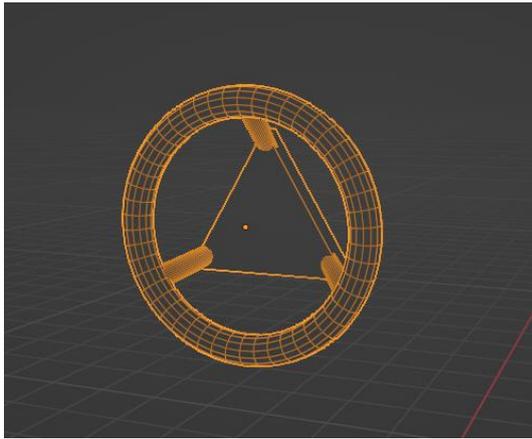


Figure III-1 le Volant du rayon

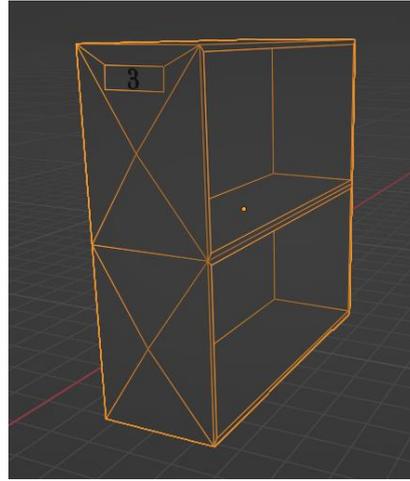


Figure III-2 Le rayon

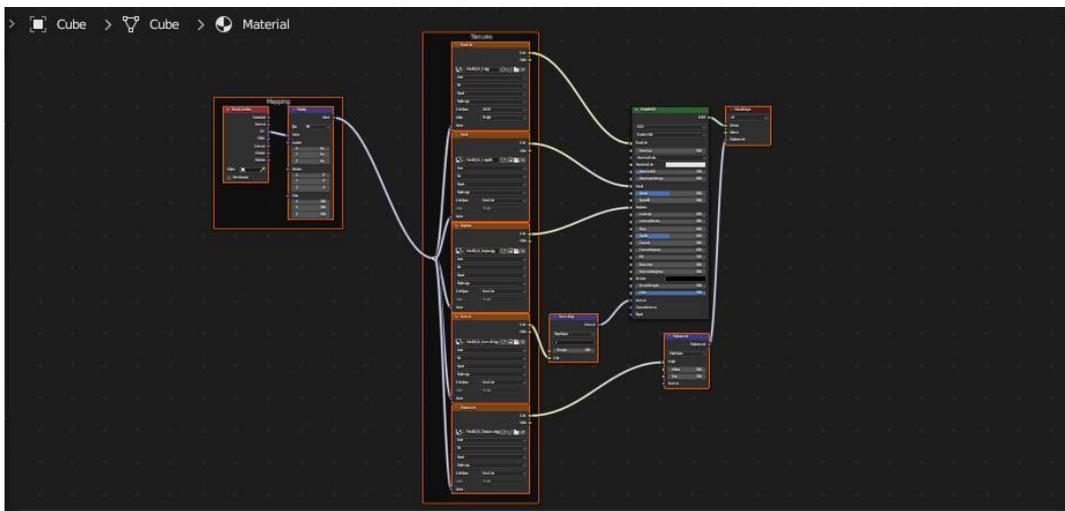


Figure III-3 les nœuds utilisés pour avoir une couleur métallique

En conséquence, on a obtenu des modèles 3D détaillés et proche de la réalité des rayons. (Figure 3.4)



Figure III-4 Rayon1

### III.2.2 Simulation Unity

Une fois qu'on a finalisé les modèles 3D dans Blender, on a procédé à leur exportation vers Unity, où on a pu les intégrer à notre scène de simulation.

On a choisi d'utiliser trois rayons (figure 3.5), car trois représente le nombre minimum requis pour englober toutes les conditions de commande. En simulant trois rayons, nous avons pu prendre en compte différentes configurations et scénarios, couvrant ainsi un large éventail de situations réelles.

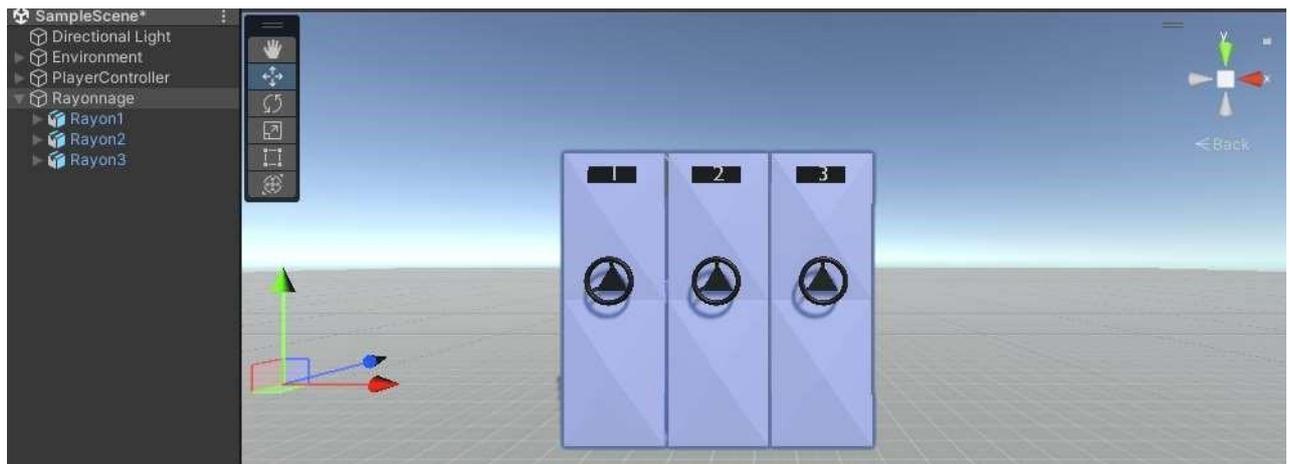
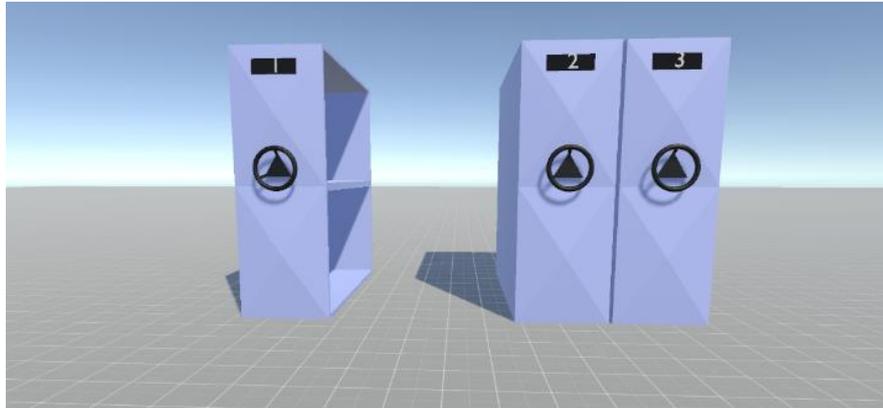


Figure III-5 les 3 rayons dans Unity

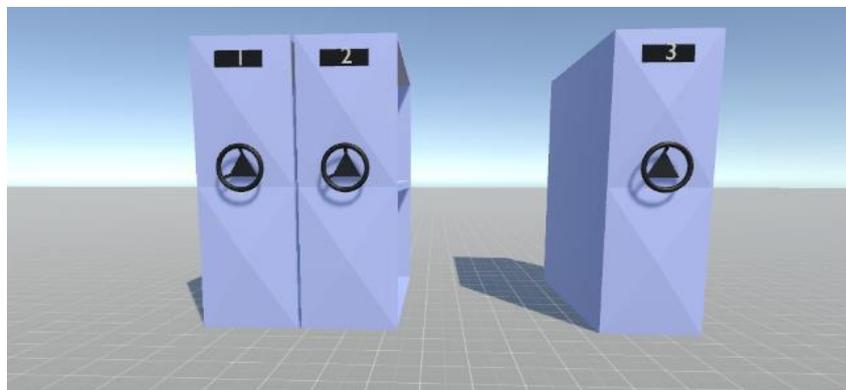
Le fonctionnement de la simulation est de commander les rayons par le clavier. Chaque rayon est associé à une touche spécifique, et en cliquant sur la touche correspondante, nous pouvons contrôler l'ouverture et la fermeture des rayons.

Lorsqu'on appuie sur la touche "1" du clavier, le rayon 1 s'ouvre pour permettre l'accès aux produits qui y sont stockés. (Figure 3.6)



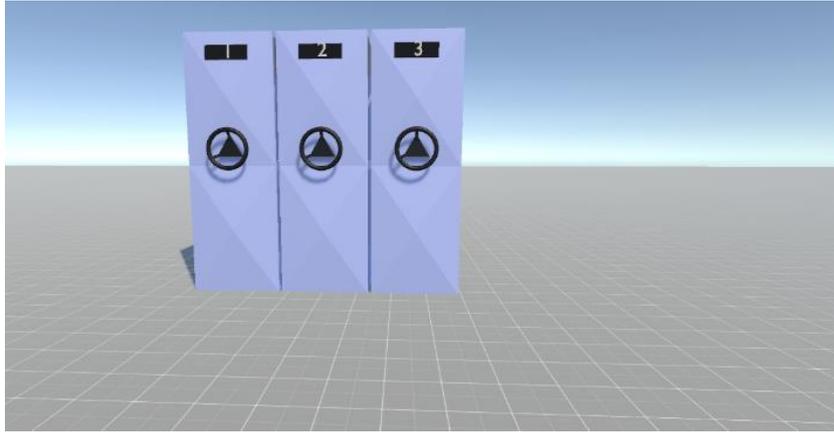
*Figure III-6 l'ouverture du rayon 1*

De même, en appuyant sur la touche "2" du clavier, le rayon 2 s'ouvre. Néanmoins, il faut que le rayon 1 s'ouvre aussi, car il est impossible d'ouvrir le rayon 2 sans l'ouverture du rayon 1. (Figure 3.7)



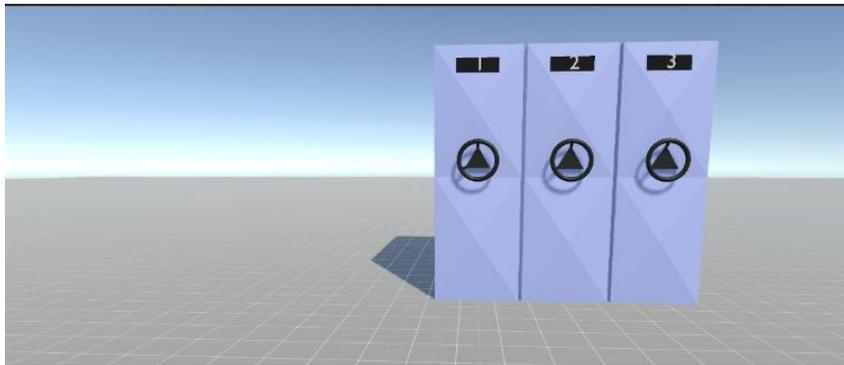
*Figure III-7 l'ouverture des rayons 1 et 2*

Enfin, lorsque on appuie sur la touche "3" du clavier, les trois rayons s'ouvrent simultanément. Les étagères du rayon 3 sont accessibles comme on a l'accès à tous les produits stockés dans ce rayon. (Figure 3.8)



*Figure III-8 l'ouverture des 3 rayons*

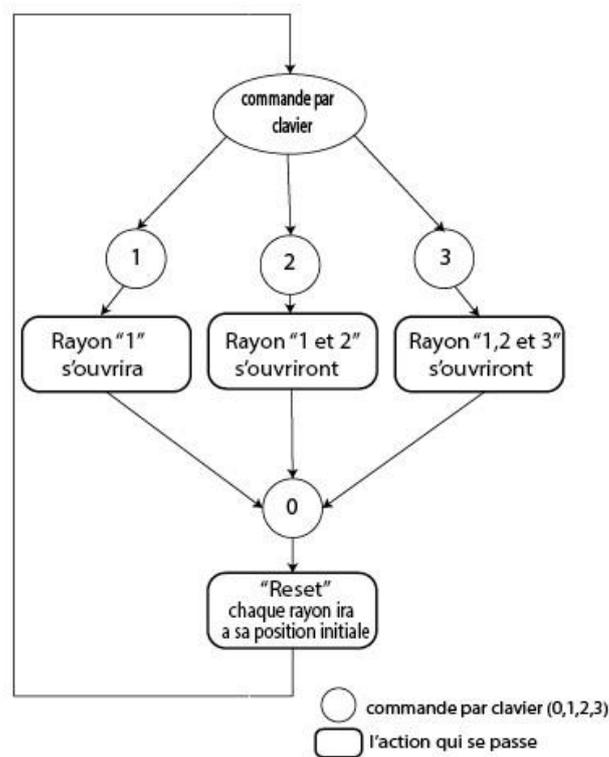
Pour fermer les rayons on clique sur la touche "0" et toutes les positions des rayons seront réinitialiser. (Figure 3.9)



*Figure III-9 la fermeture des 3 rayons*

### **III.2.3 Programmation de la simulation**

On peut résumer le tout dans un seul schéma fonctionnel ou un algorithme pour fournir une compréhension claire et structurée : (Algorithme 3.1)



Algorithme III-1 Algorithme du fonctionnement de la simulation

Pour pouvoir réaliser cette simulation on a créé et développé des scripts en C# dans Visual Studio pour mettre en œuvre la fonctionnalité de commande des rayons par clavier, on a utilisé des classes publiques (Figure 3.10) pour optimiser le travail et développer une simulation plus efficace et évolutive, car en utilisant de classe on crée une interface dans Unity (Figure 3.11) qui nous permet de faire des changements ou des améliorations à tout moment sans retourner au script.

```

5 public class Positionner : MonoBehaviour
6 {
7     public Transform rayon1;
8     public Transform rayon2;
9     public Transform rayon3;
10    [SerializeField] private AnimationCurve curve;
11    [SerializeField] private Vector3 goalPosition1;
12    [SerializeField] private Vector3 goalPosition2;
13    [SerializeField] private Vector3 goalPosition3;
14    [SerializeField] private Vector3 currentPosition1;
15    [SerializeField] private Vector3 currentPosition2;
16    [SerializeField] private Vector3 currentPosition3;
17    [SerializeField] private float speed = 0.5f;
18    private float current1, current2, current3, target1, target2, target3;
19 }
  
```

Figure III-10 une classe du script



Figure III-11 Interface Unity

Cette interface nous offre une modularité et une flexibilité pour tester les différentes positions des rayons pour améliorer la qualité de la simulation. Il est également possible de choisir la courbe de vitesse du mouvement des rayons à utiliser. (Figure 3.12)

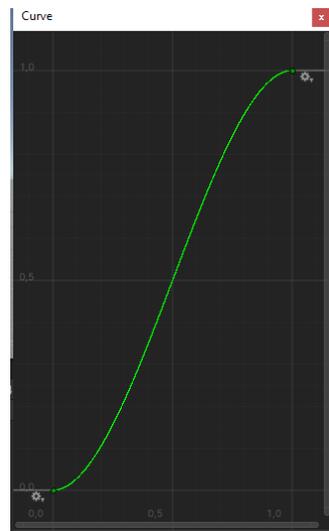


Figure III-12 la courbe qu'on a utilisé

On a choisi une courbe qui démarre avec une accélération non fulgurante, suivi par une petite décélération pour assurer un mouvement fluide qui corresponde à la réalité. Unity offre d'autres courbes aussi, qu'on peut utiliser si on veut adapter une différente conception. (Figure 3.13)

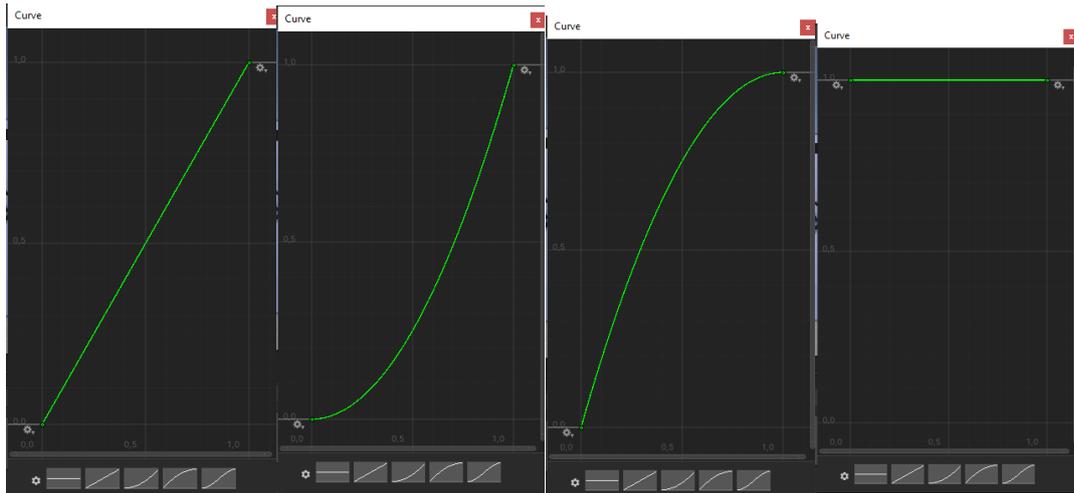


Figure III-13 les différentes courbes que Unity offre

### III.3 La conception du projet

Lorsqu'il s'agit de concevoir des objets en trois dimensions (3D), SolidWorks offre des outils avancés qui permettent aux concepteurs et aux ingénieurs de créer des modèles 3D précis et propose des fonctionnalités spécifiques et des outils CAM (computer-aided manufacturing ou en français FAO ; la fabrication assistée par ordinateur) intégrés qui vous permettent de générer des programmes de découpage CNC (Computer Numerical Control) à partir de modèles 3D.

Pour cette raison on a utilisé ce logiciel puissant, pour pouvoir concevoir des modèles précis et les découper plus tard avec la machine CNC laser, car ce processus de fabrication automatisé nous présente la possibilité d'avoir des pièces personnalisées et précises.

On a conçu les modèles 3D des rayons de manière intelligente afin que chaque pièce soit connectée à l'autre et puisse être assemblée facilement comme des Lego. (Figure 3.14)

SolidWorks nous a permis aussi d'ajouter les vis et boulons nécessaires pour l'assemblage des pièces du projet pour avoir un travail bien organisé avec une liste de tout ce que on est besoin. (Figure 3.15)

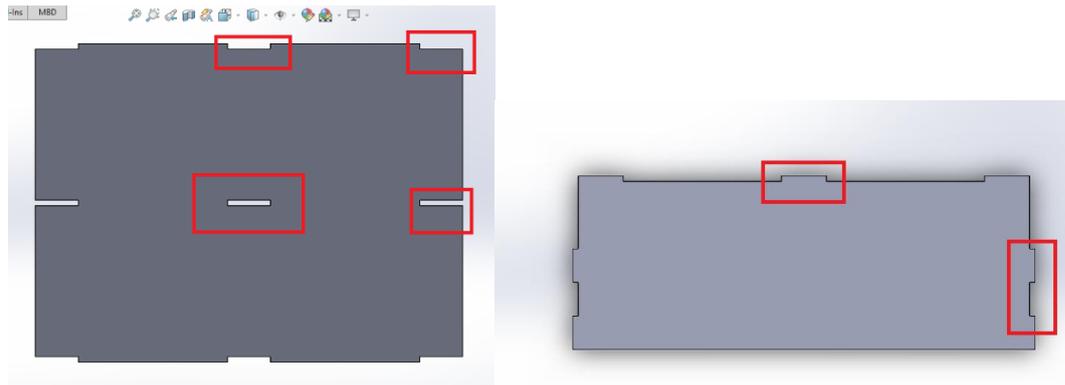


Figure III-14 le dos et l'étagère du rayon

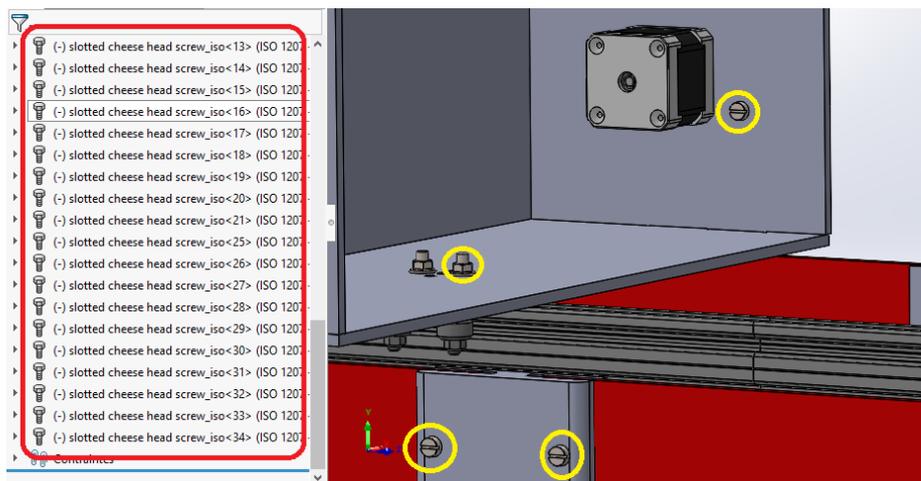


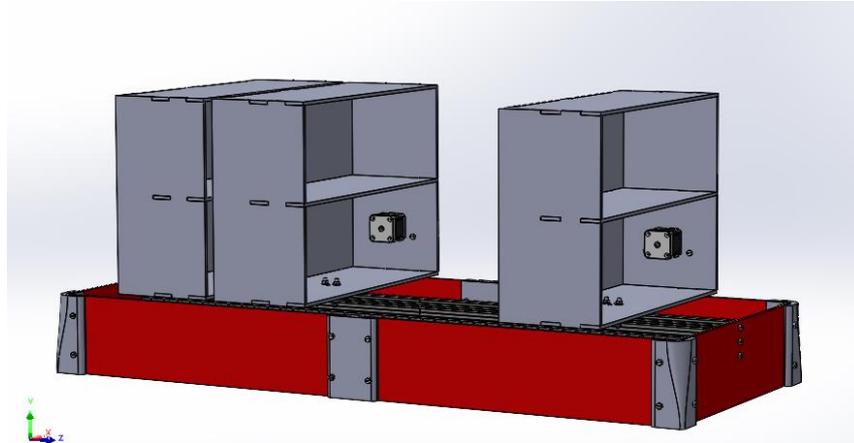
Figure III-15 liste de boulonnerie

Pour le système d'opération, on a opté pour l'utilisation des moteurs pas à pas (Nema 17) qui sont couramment utilisés dans les systèmes de contrôle de mouvement pour leurs performances, fiabilité et précision, avec un système de courroie qui offre également un certain nombre d'avantages, notamment la transmission efficace du mouvement sans perte de puissance, la capacité à absorber les vibrations et les chocs, et une maintenance facile, car il peut être remplacé ou modifié pour adapter les mouvements à l'évolution des exigences du système.

On a comme capteurs, un fin de course placé à l'extrémité de chaque rayon. Ce dernier joue un rôle essentiel pour plusieurs aspects clés du projet, notamment le repérage du point de départ et la sécurité.

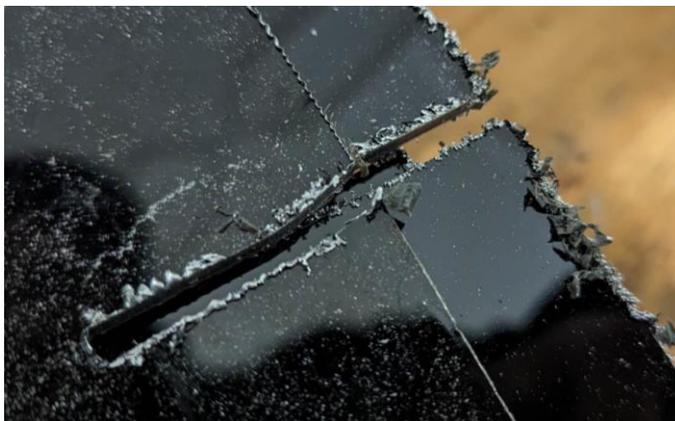
Ce système permet d'obtenir une synchronisation fluide, une précision de stockage et récupération des produits et une extensibilité, comme on peut ajouter plus de rayons facilement.

En conséquence, une conception de projet détaillée et parfaitement optimisée a été réalisée. (Figure 3.16)



*Figure III-16 la conception finale*

Pour le choix de la matière du projet on a essayé d'utiliser la matière d'Aluco ; une combinaison du plastique et de couche d'aluminium qui est solide et léger, on a pu l'utiliser pour la base des rayons (la partie rouge de la conception) mais pour les rayons eux-mêmes, il était impossible d'utiliser l'aluco, car la couche d'aluminium reflète les faisceaux laser de la machine CNC et rend la découpe imprécise. (Figure 3.17)



*Figure III-17 Aluco découpé par la machine CNC laser*

Afin de résoudre cette complication, on a remplacé l'Aluco par du plexiglass (PMMA), pour obtenir un découpage laser bien précis avec une marge d'erreur minimale. (Figure 3.18)



*Figure III-18 Découpage laser*

Par la suite, on utilise l'imprimante 3D dans le but de créer les supports de la base et de la courroie personnalisés et sur mesure. (Figure 3.19)



*Figure III-19 support de courroie imprimé en plastique*

Pour maintenir la courroie en place sans risque de briser le support (figure 3.20), il était nécessaire d'utiliser 50% ou plus de plastique pour l'impression 3D des supports.



*Figure III-20 support de courroie brisé*

Lorsque tout est prêt, on peut faire l'assemblage total (Figure 3.21) et le câblage nécessaire.



*Figure III-21 Assemblage final*

En conclusion, la phase de conception du projet a été menée avec succès. Les modèles 3D ont été soigneusement conçus pour représenter fidèlement le rayonnement, tandis que le choix des moteurs pas à pas et du système de courroie offre une solution efficace, flexible et fiable pour contrôler les mouvements des rayons.

Cette phase de conception a permis de visualiser et de planifier les aspects physiques du système de rayonnement intelligent, en mettant l'accent sur la personnalisation des pièces, la précision des mouvements et la facilité de maintenance.

### **III.4 Programmation**

La transition de la conception à la programmation marquera une étape importante dans la réalisation de ce projet, en combinant l'aspect physique du rayonnement avec la

puissance de la programmation pour créer un code qui permettra de contrôler les mouvements des rayons, de gérer la logistique du rayonnage et d'optimiser les opérations. Grâce à la solide base de conception établie, nous sommes confiants dans notre capacité à créer un système de rayonnage intelligent fonctionnel et performant.

La programmation est une étape cruciale pour concrétiser notre système automatisé. Elle permet de contrôler les capteurs et d'interagir avec les actionneurs ; dans notre cas les moteurs pas à pas et les fins de courses.

Le choix d'utiliser Arduino comme outil de commande dans le système de rayonnage présente de nombreux avantages, tels que l'accessibilité ; une plateforme de développement open-source abordable et largement disponible, une communauté active, comme il a une communauté mondiale composée de concepteurs, de programmeurs et de passionnés qui partagent leurs connaissances et leurs projets, flexibilité et l'évolutivité, car il offre la possibilité d'étendre les fonctionnalités du système de commande au fur et à mesure de l'évolution du projet.

Dans un premier temps, il faut effectuer les connexions électriques nécessaires. (Figure 3.22) Pour connecter les 3 fins de course et les 3 moteurs pas à pas à l'Arduino via le bouclier v3 et les drivers, on a suivi ces étapes : empiler correctement le bouclier v3 sur l'Arduino méga 2560, connecter chaque driver moteur (A4988) aux broches correspondantes sur le bouclier v3 ; C1, C2 et C3 et relier chaque moteur à son driver à l'aide des broches du bouclier v3 ; X, Y et Z (Figure 3.23), connecter les fins de course aux broches du bouclier v3 qui sont déjà connectées aux broches numériques de l'Arduino ; connect to D9/10/11 (Figure 3.23) et de fournir l'alimentation appropriée au bouclier v3 et aux drivers moteur conformément à leurs spécifications.



```
pinMode(11, INPUT_PULLUP);
pinMode(10, INPUT_PULLUP);
pinMode(9, INPUT_PULLUP);
```

Figure III-24 pins des fins de courses

Autrement, pour configurer les moteurs pas à pas, on passe par la configuration des broches connectées au pilote A4988, les broches principales du pilote sont : (Figure 3.25)

- **VMOT** : Cette broche est utilisée pour fournir l'alimentation du moteur pas à pas.
- **VDD** : Il s'agit de la broche d'alimentation de la logique interne du A4988.
- **STEP** : Cette broche reçoit les signaux d'impulsions (step) pour contrôler le mouvement du moteur pas à pas. Un front montant ou descendant sur cette broche déclenche une impulsion de pas.
- **DIR** : Cette broche reçoit le signal de direction pour déterminer la rotation du moteur pas à pas. Un niveau logique élevé ou bas sur cette broche indique la direction du mouvement.
- **MS1, MS2, MS3** : Ces broches permettent de configurer le mode de micro-pas du A4988. En connectant ces broches à la masse (GND) ou à la tension d'alimentation (VDD) selon un certain schéma, vous pouvez sélectionner différents modes de micro-pas (pas complet, demi-pas, quart de pas, etc.).

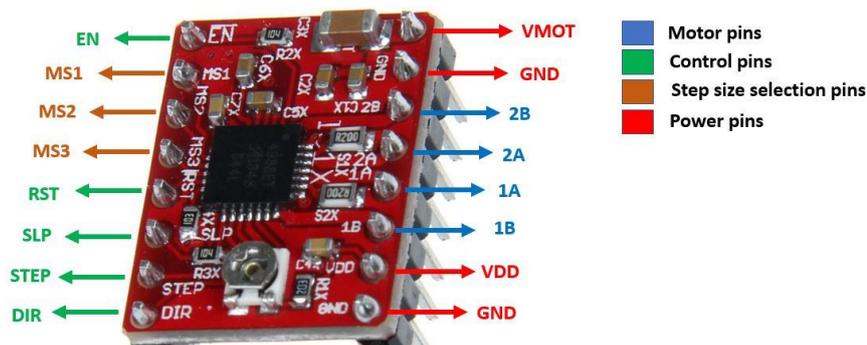


Figure III-25 broches du pilote A4988

Le A4988 accepte des signaux de pas et de direction pour contrôler le mouvement du moteur pas à pas. Ces signaux numériques sont fournis le microcontrôleur d'Arduino, Alors pour générer ces signaux et contrôler les 3 moteurs on configure les broches de pas et direction de chaque moteur avec les broches Arduino. (Figure 3.26)

```

#define X_Step 2
#define X_Dir 5

#define Y_Step 3
#define Y_Dir 6

#define Z_Step 4
#define Z_Dir 7

```

Figure III-26 configuration des broches de pas et direction

La durée de chaque impulsion de pas est essentielle pour un bon fonctionnement. Le A4988 dispose d'une horloge interne qui définit la temporisation des impulsions, Donc Pour avoir un mouvement fluide, on a utilisé le Microstepping ou le micro-pas en choisissant le mode 1/16 pas pour le mouvement du moteur pas à pas. Cela permet une résolution plus fine du mouvement, réduisant les sauts brusques entre les positions et offrant une meilleure précision et fluidité. Le 1/16 pas réduit également les vibrations et le bruit générés par le moteur, permet un meilleur contrôle de la vitesse et du couple, et contribue à réduire les problèmes de résonance.

Le réglage du mode de pas se fait en modifiant les cavaliers (jumpers) présents sur le bouclier v3 (Figure 3.27), Selon le tableau de configuration (tableau 3.1), on a placé les cavaliers correspondants aux broches "MS1", "MS2" et "MS3" dans la position appropriée pour sélectionner le mode 1/16 pas.

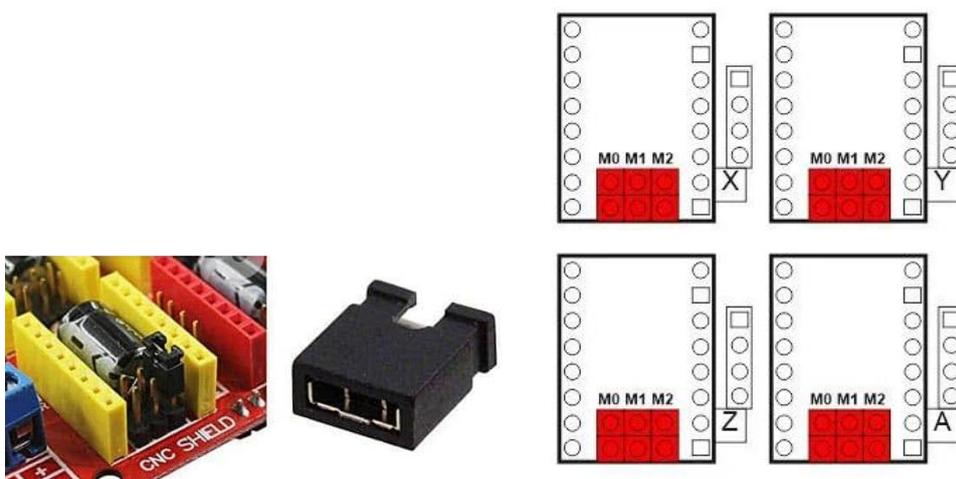


Figure III-27 configuration du mode de Microstepping

Tableau III-1 les modes du microstepping

M0	M1	M2	Microstep resolution
Low	Low	Low	Full step
High	Low	Low	1/2 step
Low	High	Low	1/4 step
High	High	Low	1/8 step
High	High	High	1/16 step

### III.4.2 Le fonctionnement

Tout d'abord, pour contrôler les moteurs pas à pas, on utilise la bibliothèque **AccelStepper**. (Figure 3.28) Cette bibliothèque offre des fonctionnalités avancées pour le contrôle précis et fluide des moteurs pas à pas, notamment en permettant l'accélération et la décélération progressives. Nous pouvons ainsi définir des déplacements précis pour chaque moteur du système de rayonnage.

De plus, la bibliothèque **MultiStepper** qui permet de coordonner plusieurs moteurs pas à pas simultanément. Cela est particulièrement utile lorsque on doit effectuer des mouvements synchronisés entre plusieurs rayons du système de rayonnage. En utilisant cette bibliothèque, on peut programmer des séquences de mouvement complexes en spécifiant les positions cibles pour chaque moteur, ce qui assure un contrôle précis et synchronisé des mouvements.

```
#include <AccelStepper.h>
#include <MultiStepper.h>
```

Figure III-28 les bibliothèques utilisées

Ensuite, on utilise les fins de course pour définir la position initiale du système de rayonnage. En plaçant des fins de course aux positions extrêmes du système, on peut déterminer avec précision la position 0 du système. Cela permet d'établir une référence de positionnement fiable pour les mouvements des moteurs.

Afin de relier le superviseur qu'on a développé en utilisant Unity à l'Arduino on utilise la communication série, qui est un moyen de transférer des données séquentiellement entre deux dispositifs, tels que l'Arduino et le superviseur, en utilisant

un seul fil pour la transmission des données dont il s'agit du câble du port USB de l'Arduino.

Alors on ouvre la communication série en utilisant la fonction **Serial.begin()**. Cette fonction permet de définir la vitesse de communication (bauds) pour la connexion série ; on a choisi une vitesse de 9600 bauds (Figure 3.29).

```
Serial.begin(9600);
```

*Figure III-29 ouvrir la communication série*

Une fois la communication série ouverte, on peut lire les données reçues par le superviseur en utilisant la fonction **Serial.read()**. Il est important de noter que la communication série nécessite une synchronisation entre le superviseur Unity et l'Arduino qu'on va établir ultérieurement dans la partie superviseur pour pouvoir envoyer des données du superviseur à l'Arduino.

Afin de mettre en œuvre un système optimale de la gestion des stocks, on a développé un code qui permet de commander de manière efficace chaque rayon par un numéro spécifique.

Plus précisément, chaque rayon est commandé en envoyant un numéro spécifique (donnée) du superviseur à l'Arduino via la communication série. Les numéros 1, 2 et 3 permettent de commander les rayons. Ainsi, Le numéro 0 permet de réinitialiser le système (position initiale ; 0) et de fermer tous les rayons. (Figure 3.30)

```
else if (data == '0') {  
Position0(); }
```

*Figure III-30 condition de la réinitialisation*

Ce système novateur qu'on a créé offre une commande numérique efficace et optimale, car le superviseur analyse le nom de l'objet à stocker ou récupérer et vérifie sa position dans la base de données. Il envoie ensuite le numéro de commande approprié à l'Arduino, quant à lui, reçoit uniquement les numéros 0, 1, 2 ou 3, transmis par le superviseur. En fonction du numéro reçu, l'Arduino active le mouvement approprié pour le rayon correspondant.

Restant dans le concept du système optimale et intelligent, pour identifier chaque position des rayons, on a choisi d'utiliser des fonctions (Figure 3.31) pour obtenir une

structure organisée du code, une modularité, réutilisabilité, car chaque fonction peut-être développer et tester individuellement et facilite la maintenance, comme on peut facilement ajouter plus de rayons (des moteurs).

```
void Position0() {
    pos[0] = 0;
    pos[1] = 0;
    pos[2] = 0;
    Rayonnage.moveTo(pos);
}
```

Figure III-31 la fonction de réinitialisation

Afin de réaliser le positionnement des rayons, il est nécessaire de convertir la distance en centimètres en un nombre de pas approprié pour le moteur pas à pas. Pour un moteur pas à pas NEMA 17, l'angle de pas le plus courant est de 1,8 degrés par pas. Donc pour un tour complet de 360 degrés on a **360/1.8=200 pas**.

Pour déplacer un rayon à une distance de 20 cm(200mm), il est essentiel de prendre en compte la circonférence d'un tour complet. Dans notre cas, on a utilisé une poulie avec un diamètre de 10mm. (Figure 3.32)



Figure III-32 Poulie pour moteur pas à pas

Donc le nombre de tour nécessaire pour 20cm est :  **$T=200/(10*\pi) \approx 6.369$  tours.**

En fin, le nombre de pas est le nombre de tours multiplier par le nombre de pas par tour, alors :  **$P=200*6.366\approx 1273.8$  pas**

Mais comme on a utilisé un micro-pas de 1/16 il faut multiplier ce nombre par 16, donc :  **$P=1273.8*16\approx 20381$  pas.**

Ou bien on peut juste utiliser la règle de trois ; on sait qu'il faut **3200pas** pour accomplir un tour (200\*16 du micro-pas) et la circonférence de la poulie et **31.41mm** donc pour parcourir

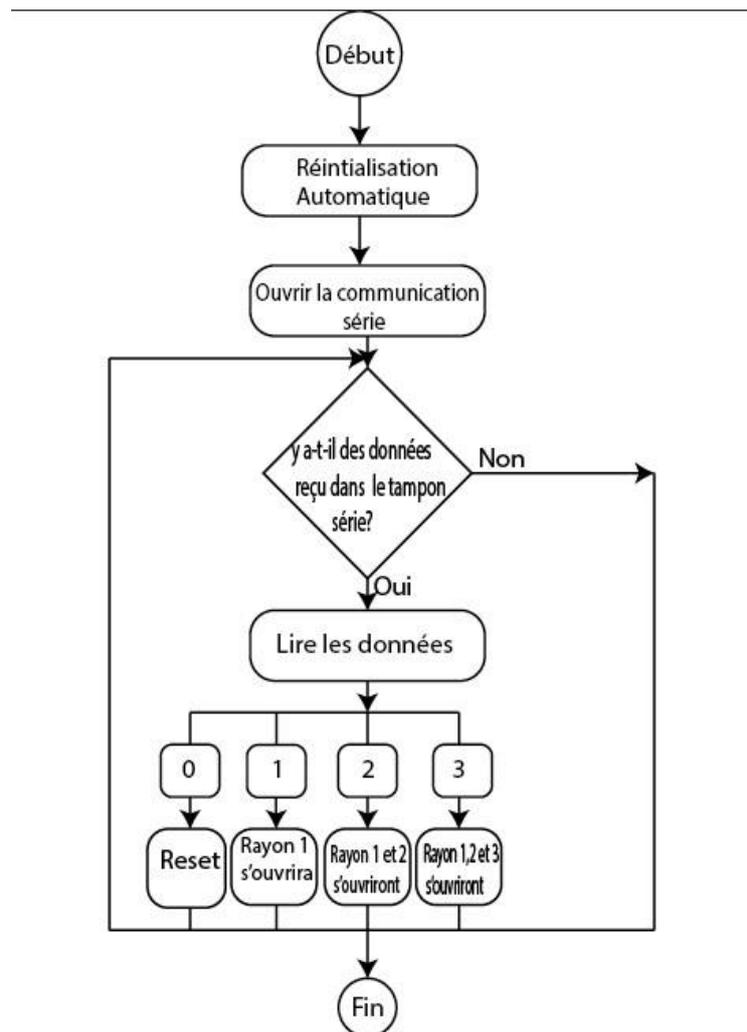
la distance de **200mm** on a :  **$P = \frac{3200*200}{31.41} = 20371.8$  pas**

Cette conversion est ensuite utilisée dans le code pour effectuer le positionnement précis.

### III.4.3 Algorithme

En résumé, l'utilisation des bibliothèques **AccelStepper** et **MultiStepper** était nécessaire pour un contrôle précis et fluide des moteurs pas à pas. Les fins de course nous permettent de définir la position 0 du système de rayonnage. En ouvrant la communication série, l'Arduino pourrait recevoir les commandes émises par le superviseur et y répondre en conséquence.

Ces étapes clés constituent les fondements du fonctionnement du code Arduino dans le contexte de la gestion du système de rayonnage, On peut exprimer ça avec un algorithme approprié qui permet d'avoir une vision plus claire et pour réduire la complexité du code Arduino en le décomposant en étapes majeures. (Algorithme 3.2)



Algorithme III-2: algorithme du code Arduino

Pour conclure, il y a toujours des possibilités d'amélioration et d'expansion du projet, C'est là que réside l'importance du développement d'un code structurel, modulé et fonctionnel, permet une gestion efficace et optimisée.

### III.5 Supervision

Dans un projet, la supervision joue un rôle crucial en permettant une gestion centralisée et efficace de l'ensemble du système. Elle offre une vision d'ensemble, facilite le contrôle et fournit des informations précieuses pour la prise de décisions éclairées.

Dans notre cas, on a choisi d'utiliser Unity pour développer un superviseur, car cette plateforme offre des fonctionnalités puissantes et polyvalentes. Grâce à Unity, on a pu créer une interface utilisateur conviviale et immersive, visualiser en temps réel l'état des rayons, recevoir et analyser les données, ainsi que commander et surveiller les actions à travers une interaction intuitive. (Figure 3.33)



Figure III-33: Interface du Superviseur

Pour parvenir au résultat final du superviseur développé, on a suivi plusieurs étapes clés.

#### III.5.1 Conception du superviseur

Tout d'abord, on a créé un canvas dans Unity, qui sert de fenêtre d'affichage pour l'interface utilisateur. On a utilisé les fonctionnalités de Unity pour ajouter des boutons, menu déroulant (DROP DOWN) et des champs de saisie (INPUTFIELD) nécessaires

pour permettre aux utilisateurs d'interagir facilement avec le superviseur en saisissant les informations pertinentes.

Le superviseur qu'on a développé offre une interface utilisateur attrayante et intuitive. Il comprend un menu déroulant permettant de choisir le port de connexion à l'Arduino, un bouton de connexion garantissant un accès sécurisé au système, tandis qu'un voyant change de couleur, passant du rouge au vert, pour indiquer la connexion réussie avec le port sélectionné. Cette fonctionnalité visuelle ajoute une touche de dynamisme et permet de vérifier rapidement l'état de la communication. (Figure 3.34)

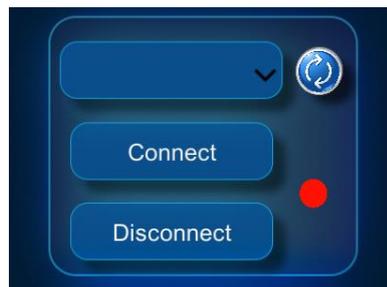


Figure III-34: interface du communication série

Le superviseur offre deux possibilités d'utilisation pour une flexibilité maximale de gérer le stockage et la récupération des objets. Pour le stockage, on dispose d'un champ de saisie permettant d'entrer le nom de l'objet à stocker. De plus, deux menus déroulants sont disponibles : l'un pour choisir le rayon dans lequel l'objet sera stocké, et l'autre pour sélectionner la position spécifique. En cliquant sur le bouton "ADD Item", le rayon choisi s'ouvre automatiquement, permettant d'ajouter l'objet dans la position souhaitée. Pour fermer le rayon, il suffit de cliquer sur le bouton "Close". Cette approche fluide et interactive facilite grandement la gestion du stockage des objets. (Figure 3.35)



Figure III-35: interface d'ajout d'un objet

La deuxième possibilité est la recherche et la récupération des objets. Un champ de saisie est prévu pour entrer le nom de l'objet recherché. En cliquant sur l'icône de recherche, un texte s'affiche (Figure 3.36) pour nous informer de la position et du rayon où l'objet recherché se trouve. Deux boutons sont également disponibles : l'un pour ouvrir le rayon correspondant à l'objet recherché "Open", et l'autre pour le fermer une fois l'objet récupéré "Close". Cette fonctionnalité facilite considérablement la localisation et la récupération des objets, tout en offrant une expérience utilisateur fluide et interactive. (Figure 3.37)

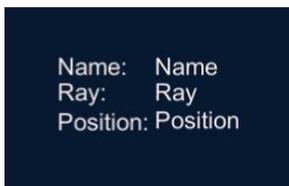


Figure III-36: L'affichage du placement de l'objet



Figure III-37: Interface de recherche

On a aussi ajouté une touche d'interactivité et de convivialité au superviseur pour faciliter la recherche des objets. Une série de numéros, allant de 1 à 6, représentant les différentes positions possibles où un objet peut être situé dans le rayon. Un voyant lumineux s'allume à côté du numéro correspondant à la position de l'objet lors de la recherche. Cette représentation visuelle permet une localisation rapide et efficace de l'objet recherché, en offrant une vue d'ensemble claire et concise des différentes positions possibles. (Figure 3.38)

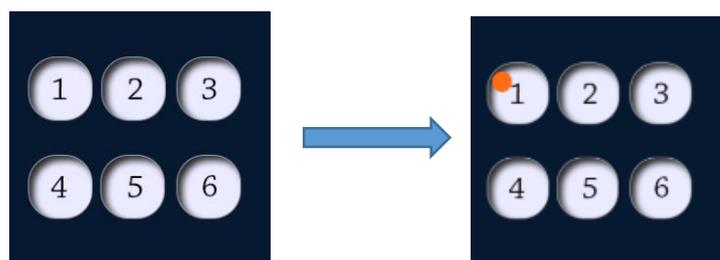


Figure III-38: Indication visuelle de la position de l'objet recherché

Dans l'ensemble, le superviseur qu'on a développé à l'aide d'Unity combine un design esthétique et fonctionnel pour offrir une expérience utilisateur attirante et efficace. Les menus déroulants, les boutons d'interaction, les champs de saisie et les indicateurs visuels contribuent à une utilisation intuitive du système. Que ce soit pour le stockage ou la recherche d'objets, ce superviseur permet une gestion précise, rapide et simplifiant.

### III.5.2 La base de données

Ensuite, on a établi une liaison étroite entre le superviseur et la base de données SQLite, qui est une bibliothèque logicielle qui fournit un système de gestion de base de données relationnelles pour les systèmes embarqués et les applications autonomes. On a utilisé SQLite, car elle est conçue d'être léger, rapide et locale sans avoir besoin d'un serveur. Cela nous permet d'accéder rapidement aux données stockées dans la base de données, telles que les informations sur les objets, leurs positions. Grâce à cette connexion, on peut récupérer ou stocker et afficher les données pertinentes dans le superviseur, offrant ainsi une vision en temps réel de l'état du système de stockage.

Une des grandes forces de SQLite réside dans sa simplicité. Elle ne nécessite aucune configuration de serveur ou d'administration complexe. Il suffit de lier la bibliothèque SQLite à l'application et de commencer à l'utiliser. La base de données est simplement un fichier sur le disque.

De plus, SQLite prend en charge le langage SQL (Structured Query Language), ce qui permet d'effectuer des opérations courantes sur la base de données. On peut créer, modifier et supprimer des tables, insérer, mettre à jour et supprimer des données, ainsi que réaliser des requêtes complexes pour extraire les informations dont on a besoin.

Pour accéder et modifier la base de données dans Unity, on utilise un script qui facilite cette tâche. Tout d'abord, on importe la fonctionnalité « **using Mono.Data.Sqlite** » en C#, qui nous donne accès à diverses fonctionnalités pour exécuter des commandes SQL. Ensuite, on crée un fichier dédié pour stocker les données en utilisant la déclaration « **private string dbName = "URI=file : Inventory.db"** ». Cette déclaration spécifie le nom et le répertoire du fichier. Pour établir une connexion entre Unity et le fichier, on utilise « **var connection = new SQLiteConnection(dbName)** ». Enfin, pour envoyer des commandes de stockage, on utilise « **var command = connection.CreateCommand()** ». (Figure 3.39)

```

using (var connection = new SQLiteConnection(dbName))
{
    connection.Open();
    using (var command = connection.CreateCommand())
    {
        command.CommandText = "INSERT INTO books (name, ray, position)";
        command.ExecuteNonQuery();
        Debug.Log("Data Sent");
        data.SendData(ray.options[ray.value].text);
        actualPosRay.text = ray.options[ray.value].text;
        bookName.text = "";
    }
    connection.Close();
}

```

Figure III-39: commande de la base de données

Grâce à l'utilisation de SQLite, on dispose d'une solution pratique et puissante pour gérer les données dans le cadre du projet de rayonnage intelligent. Cela contribue à une expérience utilisateur fluide et efficace, tout en offrant une gestion optimale des informations relatives au système de stockage.

### III.5.3 La programmation

Pour garantir exécution efficace du superviseur, on a développé six scripts qui jouent des rôles essentiels. Tout d'abord, on a créé un script pour le positionnement des rayons en fonction des commandes reçues pour la récupération ou le stockage des objets. Ce script assure une gestion automatisée des mouvements des rayons, permettant ainsi de placer les objets dans les positions appropriées.

Ensuite, on a conçu un script dédié au choix du port, qui facilite la configuration de la connexion entre le superviseur et l'Arduino. De plus, on a ajouté un bouton de rafraîchissement, (Figure 3.40) ce qui permet de mettre à jour la liste des ports disponibles, offrant une flexibilité supplémentaire lors de la configuration de la connexion. Ce script permet aux utilisateurs de choisir facilement le port de communication adéquat, assurant ainsi une connexion fiable.

```

public void RefreshPorts()
{
    dropdown.ClearOptions();

    string[] ports = SerialPort.GetPortNames();

    foreach (string port in ports)
    {
        dropdown.options.Add(new Dropdown.OptionData(port));
    }

    dropdown.value = 0;
    dropdown.RefreshShownValue();
}

```

Figure III-40: programmation du bouton de rafraichissement

On a également mis en place un script pour gérer la connexion et la déconnexion du port, le script active la communication série (Figure 3.41), permettant ainsi l'échange de données entre les deux systèmes. De même, lorsque l'utilisateur souhaite interrompre la connexion, le script ferme la communication série de manière sécurisée. Cette fonctionnalité garantit une interaction stable entre le superviseur et l'Arduino.

```
serialPort = new SerialPort(portName, 9600);  
serialPort.Open();  
Debug.Log("Serial Opened");
```

Figure III-41: ouverture la communication série

Pour la fonction de recherche d'un objet, on a développé un script spécifique qui permet de localiser rapidement l'objet recherché dans la base de données, ce script offre une fonctionnalité de recherche avancée, permettant à l'utilisateur de saisir le nom de l'objet recherché. Ensuite, le script traite cette commande et effectue une requête (Figure 3.42) dans la base de données pour récupérer les informations de position et de rayon de l'objet correspondant. Ces informations sont ensuite affichées à l'utilisateur, lui permettant de localiser facilement l'objet recherché dans le système de stockage

```
command.CommandText = "SELECT * FROM books WHERE name
```

Figure III-42: commande pour trouver l'objet recherché dans la base de données

Un autre script a été développé pour gérer l'ajout d'un nouvel objet dans la base de données. Une fois les informations saisies, le script s'assure de les enregistrer correctement, garantissant ainsi une gestion précise et organisée des nouveaux objets.

Enfin, un script dédié à la localisation de la position a été mis en place en utilisant un voyant lumineux. Lorsque l'utilisateur effectue une recherche d'objet, ce script active le voyant lumineux correspondant à la position de l'objet recherché.

Grâce à ces six scripts, on assure un fonctionnement fluide, efficace et organisé du superviseur. Chaque script remplit un rôle spécifique, contribuant ainsi à une expérience utilisateur idéale et à une gestion optimisée des objets dans le système de stockage.

### **III.5.4 Observation des signaux numériques transmis par le superviseur à l'Arduino à l'aide d'un oscilloscope**

Après avoir utilisé la console série de l'IDE Arduino pour afficher les données reçues ou envoyées. L'oscilloscope peut être un outil précieux pour une visualisation et une analyse plus

approfondie des signaux dans le but de mieux comprendre la communication série entre le superviseur et l'Arduino.

Lorsqu'on utilise le protocole de communication série UART (Universal Asynchronous Receiver/Transmitter), le signal capturé sur l'oscilloscope pendant l'envoi des caractères se présentera sous la forme d'une séquence binaire composée de 8 bits de données, accompagnés d'un bit de démarrage (start) de valeur 0, suivi bit d'arrêt (stop) de valeur 1. (Figure 3.43)

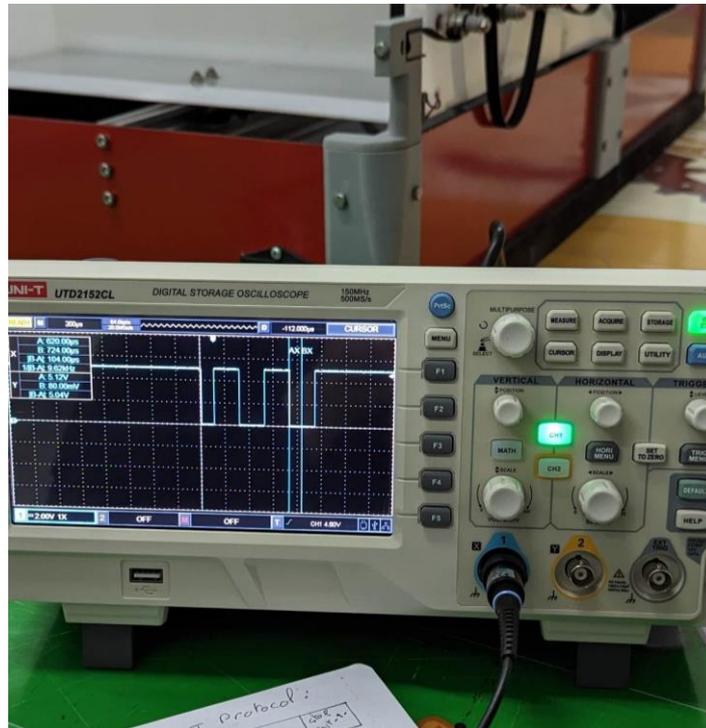


Figure III-43 L'utilisation de l'oscilloscope

Chaque bit sera représenté par un niveau de tension haut (logique 1) ou bas (logique 0) pendant une durée spécifique, en fonction de la vitesse de transmission (baud rate). Dans notre cas, où on a utilisé une vitesse de transmission de 9600 bauds ou bit/s puisque la donnée est transmise bit par bit, cela signifie que chaque bit est transmis en 104 microsecondes.

Au moment où on cherche un objet et il est détecté dans le rayon 1, le superviseur envoie un caractère "1" à l'Arduino pour indiquer l'ouverture du rayon 1. Les données sont transmises en utilisant l'ordre du bit le moins significatif d'abord, ce qui signifie que la trame correspondant au caractère "1" serait : 0011 0001 (sans tenir compte des bits de start et de stop). (Figure 3.44)

## ASCII Code: Character to Binary

0	0011 0000	O	0100 1111	m	0110 1101
1	0011 0001	P	0101 0000	n	0110 1110
2	0011 0010	Q	0101 0001	o	0110 1111
3	0011 0011	R	0101 0010	p	0111 0000
4	0011 0100	S	0101 0011	q	0111 0001
5	0011 0101	T	0101 0100	r	0111 0010
6	0011 0110	U	0101 0101	s	0111 0011
7	0011 0111	V	0101 0110	t	0111 0100
8	0011 1000	W	0101 0111	u	0111 0101
9	0011 1001	X	0101 1000	v	0111 0110
A	0100 0001	Y	0101 1001	w	0111 0111
B	0100 0010	Z	0101 1010	x	0111 1000
C	0100 0011	a	0110 0001	y	0111 1001
D	0100 0100	b	0110 0010	z	0111 1010
E	0100 0101	c	0110 0011	.	0010 1110
F	0100 0110	d	0110 0100	,	0010 0111
G	0100 0111	e	0110 0101	:	0011 1010
H	0100 1000	f	0110 0110	;	0011 1011
I	0100 1001	g	0110 0111	?	0011 1111
J	0100 1010	h	0110 1000	!	0010 0001
K	0100 1011	I	0110 1001	'	0010 1100
L	0100 1100	j	0110 1010	"	0010 0010
M	0100 1101	k	0110 1011	(	0010 1000
N	0100 1110	l	0110 1100	)	0010 1001
				space	0010 0000

Figure III-44 Code Ascii

Par conséquent sur l'oscilloscope on observe cette même séquence binaire en commençant par le dernier bit qui est le bit le plus significatif (de la droite à gauche). (Figure 3.45)

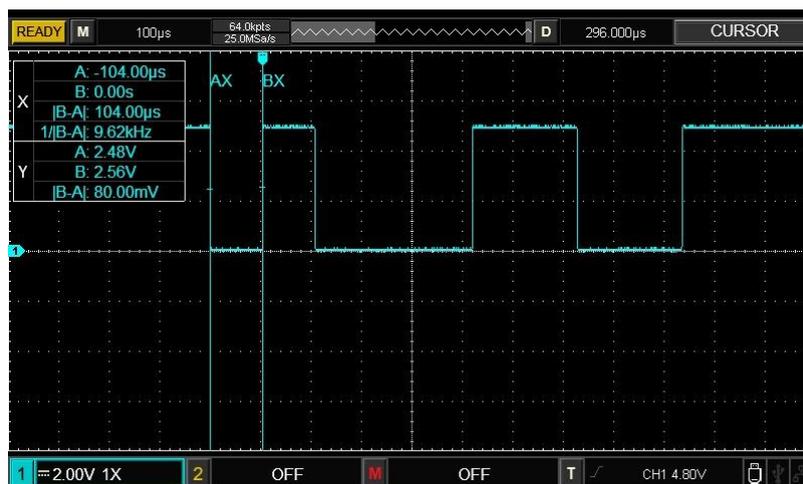


Figure III-45 La séquence binaire correspond au caractère "1"

De manière similaire, pour les autres cas, le superviseur envoie le caractère "2" pour ouvrir le rayon 2, le caractère "3" pour ouvrir le rayon 3 et le caractère "0" pour la fermeture des rayons ou la réinitialisation.

Sur l'oscilloscope le caractère "2" sera représenté par une séquence binaire spécifique : 0011 0010 (Figure 3.46), le caractère "3" aura une autre séquence binaire : 0011 0011 (Figure 3.47), et le caractère "0" aura une séquence binaire différente : 0011 0000 (Figure 3.48).

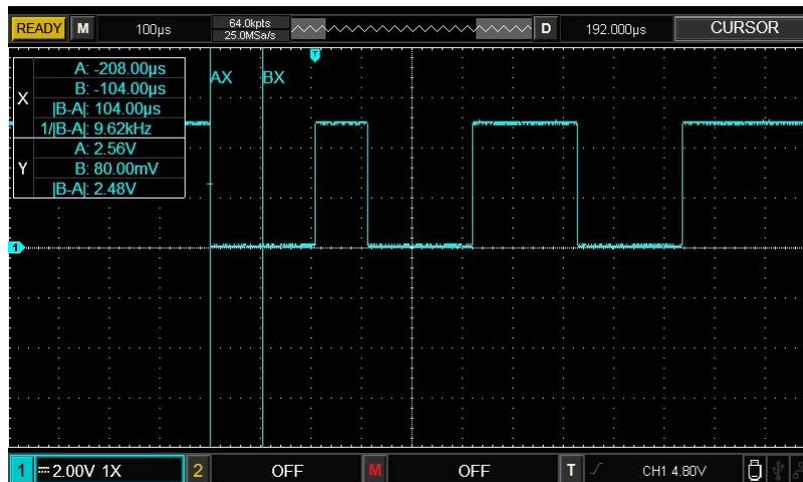


Figure III-46 La séquence binaire correspond au caractère "2"

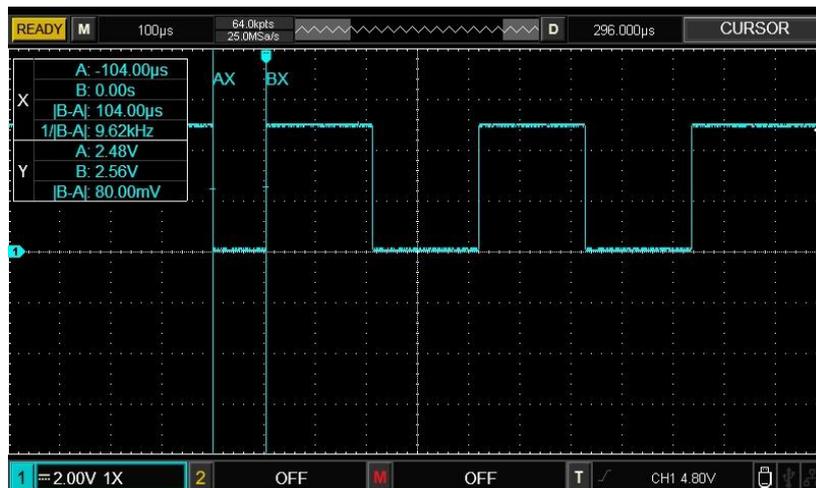


Figure III-47 La séquence binaire correspond au caractère "3"

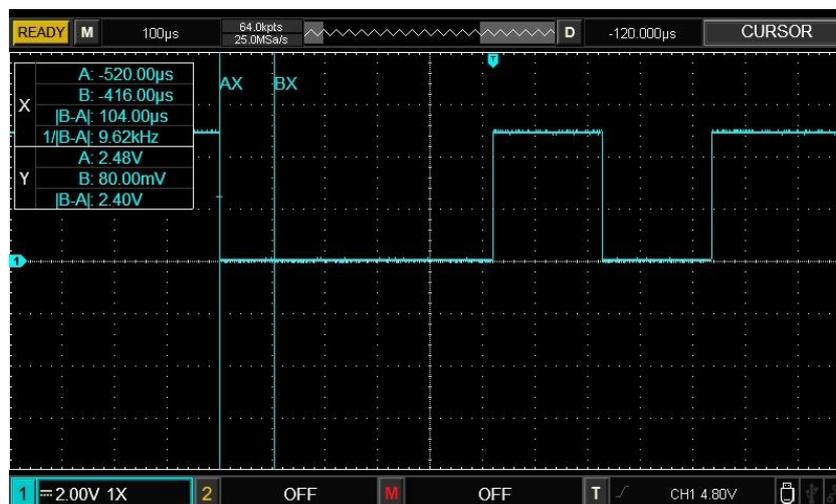
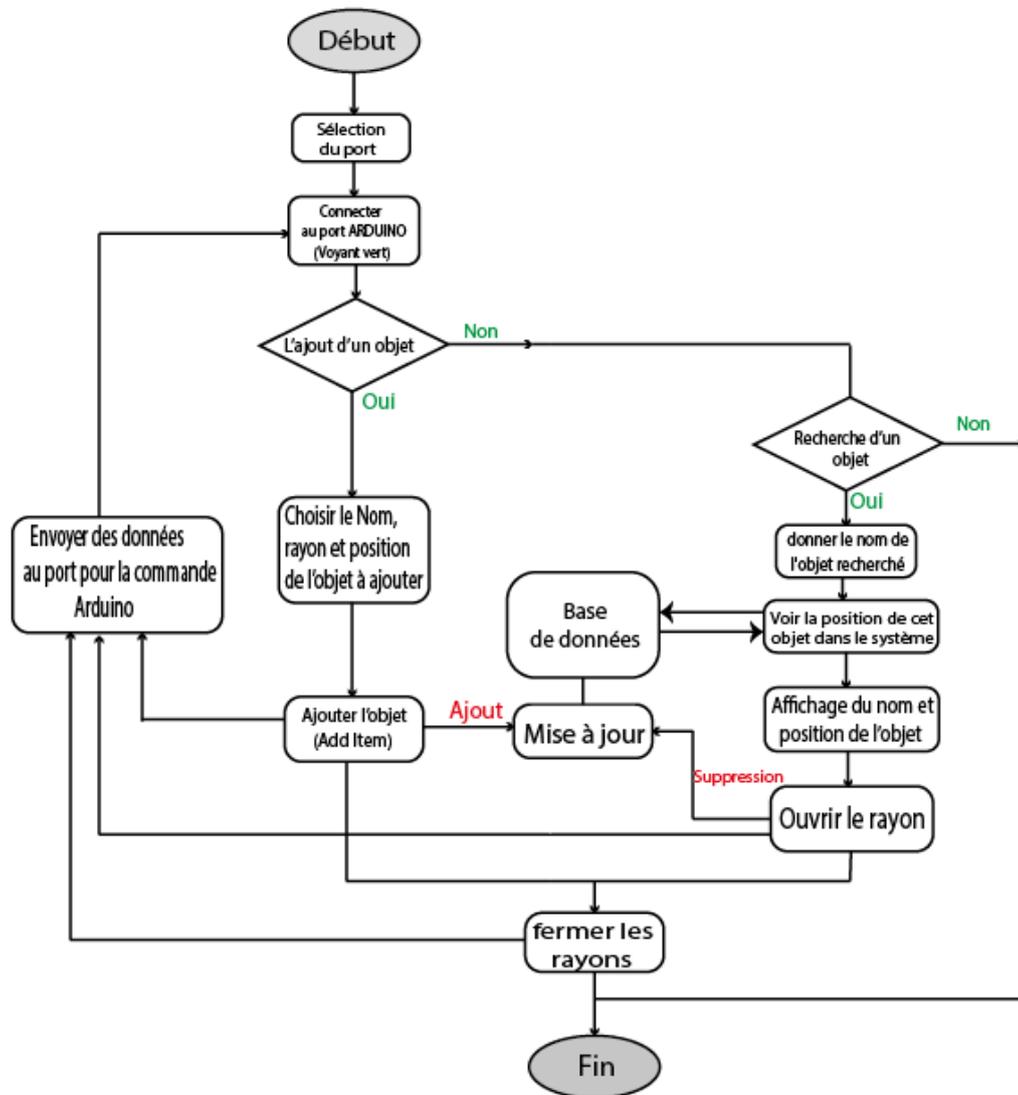


Figure III-48 La séquence binaire correspond au caractère "0"

### III.5.5 Algorithme

On peut résumer le tout par l'algorithme suivant : (Algorithme 3.3)



Algorithme III-3: Résumé de la logique de fonctionnement des scripts de Unity

### III.6 Conclusion

Le projet de rayonnage intelligent que nous avons développé a été couronné de succès. Grâce à une approche méthodique et à l'utilisation de technologies avancées, nous avons réussi à relever les défis du projet de manière efficace.

Ce système automatisé a amélioré la productivité et la rentabilité de la gestion des stocks, tout en réduisant les erreurs humaines. Nous avons pu optimiser les performances, utiliser l'espace de stockage de manière optimale et améliorer la précision des commandes, ce qui a apporté des avantages tangibles à notre solution.

## **Conclusion générale**

## Conclusion

En conclusion, ce projet de rayonnage intelligent s'est avéré être une solution efficace pour améliorer la gestion des stocks et réduire les pertes de temps et d'efforts dans un entrepôt.

En combinant la programmation via Arduino et la supervision par le logiciel Unity, on a créé un système automatisé efficace et convivial facilement adaptable et ergonomique.

Grâce à cette solution, les tâches de recherche et de récupération des produits dans l'entrepôt sont simplifiées et optimisées, réduisant ainsi les erreurs humaines et les pertes de temps.

L'interface graphique intuitive de supervision par Unity permet une utilisation aisée du système, même pour les utilisateurs novices.

Ce projet a ouvert la voie à de nouvelles possibilités dans le domaine de la logistique en intégrant des technologies avancées dans la gestion des stocks. Les avantages de l'utilisation de l'Arduino et d'Unity sont nombreux, offrant une flexibilité et une adaptabilité accrues, tout en réduisant les coûts de développement.

Cependant, des améliorations peuvent encore être apportées pour optimiser davantage notre système de rayonnage intelligent. Par exemple, l'intégration de capteurs de détection plus précis comme le capteur inductif et de mécanismes de localisation avancés pourrait améliorer la précision et l'efficacité du système.

En fin de compte, le travail qu'on a mis au point est un exemple concret de la manière dont la technologie peut améliorer la gestion des stocks et aider les entreprises à gagner en efficacité et en compétitivité. Il ouvre également des perspectives passionnantes pour l'avenir, notamment l'application de l'intelligence artificielle et de l'apprentissage automatique pour améliorer encore plus les performances du système.

En somme, ce mémoire sur la réalisation d'un projet de rayonnage intelligent met en évidence l'importance de l'automatisation dans la logistique, offrant des gains de temps, d'effort et de précision. Grâce au choix judicieux de programmation avec Arduino et de supervision avec Unity, on a pu concevoir un système performant, prometteur pour les défis futurs de la gestion des stocks dans les entrepôts.

## Références bibliographiques

[1] : Arduino, <https://docs.arduino.cc/hardware/mega-2560?queryID=undefined>, Consulté le 11/06/23.

[2] : Robot maker, <https://www.robot-maker.com/shop/moteurs-et-actionneurs/14-moteur-pas-a-pas-nema-17-14.html>, Consulté le 11/06/23.

[3] : Maîtrisez sa programmation et ses cartes d'interface. Christian Tavernier, Collection : EEA, Dunod, 2014 - 2ème édition, Consulté le 11/06/23.

[4] : Sculpteo, <https://www.sculpteo.com/fr/glossaire/solidworks-definition-fr/?fbclid=IwAR2q84zMopM-YwZRpIyTZ8bGM-0aevFNKoYoAmAeg905ZBQEPRZJ1h6nZfI>, Consulté le 11/06/23.

[5]: Unity, <https://unity.com/fr>, Consulté le 11/06/23.

[6]: Blender, <https://www.blender.org/>, Consulté le 11/06/23 .