

HELICE HOLOGRAPHIQUE

Projet 3IL



20 JANVIER 2023

ALMIA Zied
MIDOUX Oscar
PUECH Mathys
TOURNU Paul

Sommaire

Sommaire	1
Introduction	3
1 Présentation du sujet	3
1.1 Affichage holographique	3
1.2 L'hologramme scénique à taille humaine.....	4
1.3 L'hélice holographique	5
1.4 Objectifs du projet	6
2 Conception et réalisation du prototype	7
2.1 Problématiques et solutions proposées.....	7
2.2 Source lumineuse.....	7
2.3 Motorisation	8
2.4 Transmission du contact en rotation.....	10
2.5 Utilisation d'une courroie	11
2.6 Quelques soudures	11
2.7 Mesure de la vitesse de rotation.....	11
3 Conception électronique	12
3.1 Alimentation	12
3.2 Capteur de présence	14
3.3 Schéma électrique	15
3.4 Prototype	16
4 Programmation du prototype	18
4.1 Description globale	18
4.2 Extraction points de couleur	19
4.3 Affichage des points de couleur sur l'hélice.....	19
4.3.1 Librairie FastLed pour arduino	19
4.3.2 Problématique mémoire RAM de l'Arduino : utilisation de PROGMEM.....	20
4.3.3 Allumage des LEDs.....	20
4.4 Interface web	21
5 Propositions d'amélioration	23
6 Conclusion	25

Sommaire des figures

Figure 1 : Fonctionnement de l'affichage holographique	3
Figure 2 : L'hologramme scénique à taille humaine	4
Figure 3 : Fonctionnement de l'œil humain	5
Figure 4 : Illustration de la persistance rétinienne	6
Figure 5 : Bandeau de LED choisi.....	7
Figure 6 : Premier moteur	9
Figure 7 : Second choix de moteur	9
Figure 8 : bague collectrice pleine.....	10
Figure 9 : Image de la courroie et de la bague collectrice	11
Figure 10: Quelques soudures.....	11
Figure 11: générateur de tensions.....	12
Figure 12 : Alimentation Raspberry/Arduino UNO	13
Figure 13 : Alimentation du capteur de présence à travers l'Arduino UNO	13
Figure 14 : Capteur de présence KY-032.....	14
Figure 15: Schéma électronique de notre prototype	15
Figure 16: Photographies du prototype	16
Figure 17: Pôle métallique cassée	16
Figure 18: Prototype en mouvement	17
Figure 19: Visualisation des points de couleurs	18
Figure 20: Code extraction des points.....	19
Figure 21: Code inversement de swap	21
Figure 22: Interface Web	21
Figure 23: Déclaration du nombre de LEDs	22
Figure 24: Prévisualisation du rendu	22

Introduction

Le présent rapport a pour objet de décrire le projet de fin d'étude. Il s'agissait de concevoir, fabriquer et programmer un dispositif d'affichage à hélice holographique. Ce type de technologie, qui est actuellement en vogue, permet d'afficher des contenus (images fixes, vidéos ou animations 3D) de manière à donner l'impression qu'ils flottent dans l'air, sans que l'observateur ne puisse voir le support d'affichage. Il s'agit là de l'objectif premier d'un affichage holographique. Il existe différentes technologies d'affichage holographique, telles que l'hologramme scénique, l'hologramme vitrine, l'hologramme pyramide ou encore l'hélice holographique. Ce dernier dispositif utilise une propriété de la vision humaine, la persistance rétinienne, pour créer une expérience utilisateur plus immersive.

1 Présentation du sujet

1.1 Affichage holographique

Définition :

Le terme "hologramme" est dérivé des mots grecs "holos" qui signifie "entier" et "gramma" qui signifie "écriture". Ainsi, il désigne une technique permettant de reproduire une image en volume, en quelque sorte une photographie en 3D. Cette idée d'une image en volume a été évoquée en premier lieu dans la littérature, notamment dans l'œuvre de Jules Verne "le Château des Carpathes", où l'auteur décrit un procédé permettant de faire réapparaître une cantatrice décédée sous la forme d'un spectre à travers un système de miroirs inclinés devant son portrait peint.

Un hologramme est une image fixe reproduite en relief sur un support, cette définition est conforme à l'étymologie du terme. Cette technique est devenue possible grâce à l'invention du laser dans les années 60. Pour créer un hologramme, un faisceau laser est divisé en deux parties à l'aide d'un verre de séparation. L'une des parties est dirigée vers l'objet à reproduire, l'autre vers la plaque photographique sur laquelle l'image sera inscrit. L'interférence entre le faisceau laser et les ondes réfléchies par l'objet permet d'obtenir l'amplitude lumineuse de l'objet et d'inscrire une image en relief sur la plaque photographique. Cette technique nécessite une maîtrise des propriétés physiques de la cohérence lumineuse et de l'interférence d'ondes de lumière. Il est important de noter que cette technique, bien qu'extrêmement complexe, ne permet de créer que des images en relief.

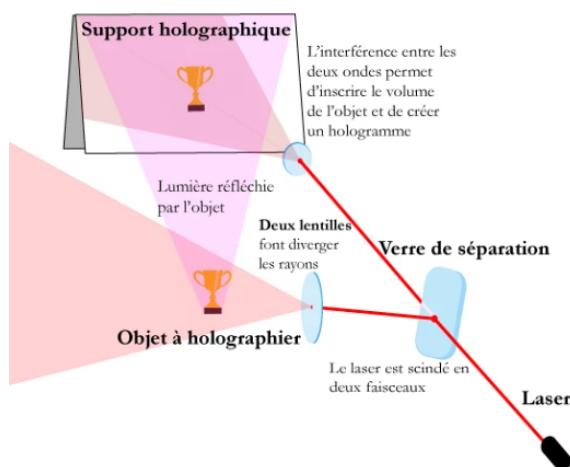


Figure 1 : Fonctionnement de l'affichage holographique

1.2 L'hologramme scénique à taille humaine

L'utilisation d'une vitre comme support de projection pour des images holographiques est courante en scène. Cependant, il est important d'avoir une bonne intensité lumineuse pour obtenir une image de bonne qualité et qui ne ressemble pas à un simple reflet. Il est également possible d'utiliser un tulle, qui est un tissu très fin qui n'est pas visible pour le spectateur, mais suffisamment dense pour capter la lumière projetée. L'utilisation de tulle présente l'avantage d'être plus maniable, étant plus léger et facile à enruler et à transporter.

L'essence de l'illusion créée par les hologrammes repose sur la profondeur qui existe entre le fond de la scène et l'hologramme. Cela permet de donner l'impression que des artistes réels se produisent en même temps que l'hologramme, alors qu'ils se trouvent en réalité derrière le tulle ou la vitre utilisée comme support de projection. Cette technique a été utilisée dans des performances live, comme lorsque le rappeur Snoop Dogg a pu rapper aux côtés de l'hologramme de Tupac.

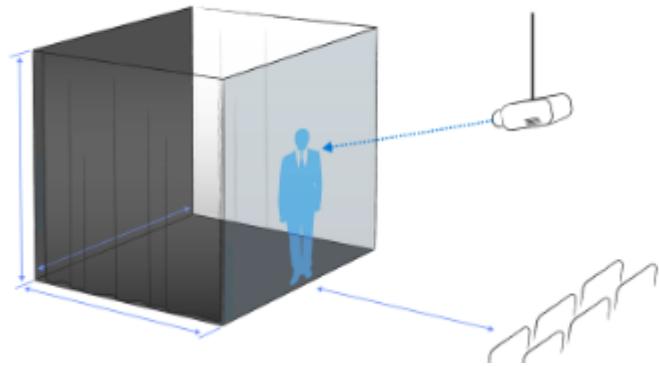


Figure 2 : L'hologramme scénique à taille humaine

1.3 L'hélice holographique

Parmi les différentes technologies d'affichage holographique actuellement disponibles, l'hélice tournante offre des résultats efficaces malgré la simplicité apparente du dispositif. Cette technique repose sur l'utilisation du principe de persistance rétinienne pour créer l'illusion d'images en relief.

La persistance rétinienne est un phénomène neurophysiologique lié à la vision qui décrit la façon dont l'image d'un objet persiste sur la rétine après que l'objet ait disparu. Il est dû à la capacité de certains neurones de la rétine à stocker de l'énergie lumineuse pendant un court instant, après quoi ils la libèrent lentement.

Ce phénomène est lié aux caractéristiques de la rétine humaine, qui est formée de photorécepteurs appelés cônes et bâtonnets. Les cônes et les bâtonnets travaillent ensemble pour capturer la lumière et la transformer en impulsions électriques qui sont ensuite transmises au cerveau. Bâtonnets étant plus sensibles aux mouvements rapides que les cônes, ils ont un temps de réponse plus court que les cônes.

Ce processus n'est pas instantané, et l'image convertie est stockée en mémoire pendant quelques instants (environ 50 millisecondes) afin d'assurer la continuité de notre vision avant d'être remplacée par une nouvelle image. Cela permet au cerveau de maintenir la continuité de notre vision. Il est important de noter que la durée de cette persistance varie selon la luminosité de l'image, plus l'image est lumineuse, plus elle est persistante.

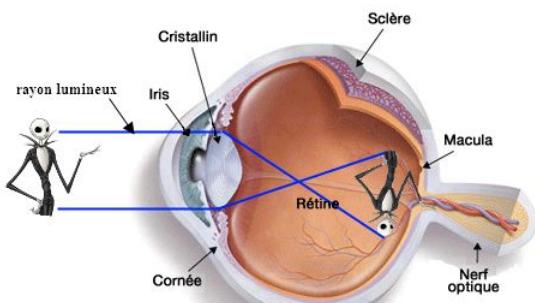


Figure 3 : Fonctionnement de l'œil humain

Pour observer le phénomène de persistance rétinienne, une expérience simple est « Le Thaumatrope ». C'est un jouet optique qui exploite le phénomène de persistance rétinienne. L'objet est un disque, maintenu par deux ficelles, sur lequel on peut observer une cage sur une face et un oiseau sur l'autre. Ainsi, lorsqu'on faisait tourner le disque suffisamment vite, on avait l'illusion que l'oiseau était dans la cage.

Le principe de persistance rétinienne est mis en jeu dans ce jouet, car lorsque les images sur le disque changent rapidement, l'œil humain ne peut pas suivre le rythme de ces changements, et l'image précédente persiste dans la vision pendant un court instant. Cela crée l'illusion d'une image animée, même si le disque ne contient qu'une image fixe à chaque côté.

Il est important de noter que le principe de persistance rétinienne est lié à la capacité de la rétine à stocker de l'énergie lumineuse pendant un court instant, avant de la libérer lentement. Il est également lié à la faible résolution temporelle de la vision humaine, qui est plus élevée pour les mouvements rapides que pour les mouvements lents.

En utilisant ce principe, le thaumatrope peut créer une illusion d'animation en utilisant des images fixes, montrant ainsi comment la persistance rétinienne peut être utilisée pour créer des illusions visuelles.

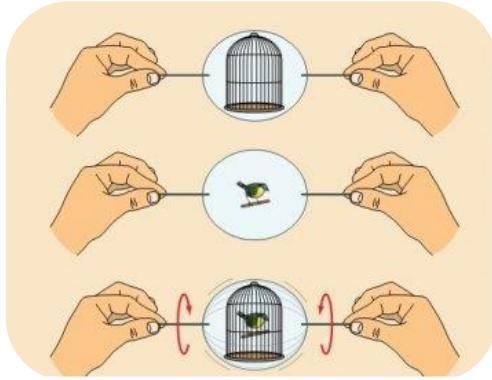


Figure 4 : Illustration de la persistance rétinienne

En utilisant ce phénomène, les hélices holographiques peuvent afficher des images en relief en projetant des images très rapides sur un support transparent tournant. L'œil humain perçoit alors l'image comme étant projetée dans l'espace, plutôt que sur un support.

1.4 Objectifs du projet

Le but de ce projet est de réaliser une étude approfondie concernant les différents paramètres qui permettent d'influer sur la qualité de l'image (type de LEDs, nombre de LEDs, vitesse de rotation, nombre de pâles, etc.), développer un dispositif d'affichage holographique à hélice rotative. Il s'agit de concevoir et de construire un prototype qui comprend les aspects mécaniques, électroniques et de programmation.

Il est également attendu que le prototype développé soit suffisamment fonctionnel et facile à utiliser pour être utilisé dans des applications réelles.

2 Conception et réalisation du prototype

2.1 Problématiques et solutions proposées

La conception d'un tel prototype n'est pas simple. Ainsi, nous avons fait de nombreuses recherches avant de se lancer sur cette conception. Dans les parties suivantes nous allons donc aborder cette conception en expliquant les choix de matériels et les problèmes que nous avons pu rencontrer pendant la création de notre hélice holographique.

2.2 Source lumineuse

Une source lumineuse est nécessaire pour créer une image holographique sur une hélice holographique. Cette source peut être de différents types, comme les LED (diodes électroluminescentes) et les micro-LED. Les LED sont des sources lumineuses très efficaces car elles consomment très peu d'énergie et ont une longue durée de vie.

Nous avons plusieurs contraintes liées à ce choix :

- Plusieurs éléments
- Éléments le plus rapproché possible
- Élément pouvant s'allumer de différentes couleurs
- Élément pouvant être contrôlés séparément les uns des autres
- L'élément doit pouvoir se fixer aux pâles de notre hélice.

Ces dernières années, les bandeaux de LED se sont beaucoup développés, ils sont généralement utilisés comme décoration d'intérieur mais la plupart d'entre eux n'offrent pas la possibilité de contrôler chaque élément séparément. C'est-à-dire que le bandeau de LED sera que d'une couleur, alors que notre objectif est de pouvoir avoir chaque LED du bandeau d'une couleur différente. On a donc dû s'orienter vers un bandeau de LED dit « adressables ». Après s'être renseigné sur les caractéristiques et la manière de les contrôler nous avons compris que la densité est un élément primordial pour notre choix. Cette densité correspond aux nombres de LED par unité de longueur. En effet, cette densité va dépendre directement de la résolution radiale de notre affichage. La densité maximale disponible sur le marché est de 144LED par mètre.

Nous avons donc commandé directement sur Amazon ce type de bandeau de LED :



Figure 5 : Bandeau de LED choisi

2.3 Motorisation

Vitesse de rotation :

D'après plusieurs études la persistance rétinienne moyenne de l'œil humain dure environ 50ms. Ainsi, pour un fonctionnement optimal notre hélice doit faire un tour complet en moins de 50ms, ce qui implique que la vitesse de rotation optimal de l'hélice doit être au minimum de :

$$v_{\text{Hélice}} = \frac{1}{50 \cdot 10^{-3}} = 1200 \text{ RPM.}$$

Or, il est possible de réduire la vitesse de rotation de l'hélice holographique pour obtenir une image satisfaisante. La vitesse de rotation de 50ms mentionnée précédemment est basée sur la persistance rétinienne positive, qui se produit sans source lumineuse. Cependant, la durée de la persistance rétinienne dépend directement de la luminosité de l'élément observé. Alors, en utilisant une source lumineuse plus brillante comme un bandeau de LED, nous pourrions obtenir des résultats satisfaisants avec une vitesse de rotation réduite, comme 500 tours par minute.

De plus, en utilisant 2 pâles nous pouvons réduire cette vitesse par deux c'est comme si nous opérons seulement un demi-tour au lieu d'un tour complet :



D'ailleurs, le constructeur « HYPERVSN » d'hélice holographique utilise une vitesse de rotation de 670 RPM et permet d'obtenir une résolution d'image tout à fait satisfaisante.

Couple du moteur :

Le couple du moteur est aussi une caractéristique à prendre en compte dans le choix du moteur. Le couple d'un moteur est une mesure de la force de rotation qu'il peut générer. Il est souvent exprimé en unités de Newton-mètre (Nm). En général, il est associé à la puissance d'un moteur, car il est nécessaire pour entraîner une charge tournante.

Pour calculer le couple d'un moteur, on peut utiliser la formule suivante : Couple = Force * Distance.

La force est générée par le moteur et s'exprime en newton (N), la distance est la distance entre le point où la force est appliquée et l'axe de rotation du moteur, on l'exprime en mètre (m).

D'après nos recherches, le couple d'un moteur est généralement fourni par le fabricant dans les données techniques. En général, plus le couple est élevé, plus le moteur est capable de déplacer des charges lourdes. Cependant, il est important de s'assurer que le moteur choisi est capable de fournir suffisamment de couple pour entraîner la charge à une vitesse désirée.

Premier choix de moteur :

Notre premier choix de moteur c'est porté sur un ventilateur que nous avions à la maison. Nous l'avons alors démonté pour en sortir le moteur. Après quelques calculs nous avons pu en déduire que la vitesse de rotation de ce moteur était de 1200 RPM. C'était un moteur à courant alternatif. Mais celui-ci n'était pas assez puissant pour faire tourner les pâles sur la tige du moteur. Nous nous sommes donc orientés vers un autre choix.



Figure 6 : Premier moteur

Second choix de moteur :

Notre second choix de moteur c'est fait au près d'un fabricant. C'est un moteur électrique 12V à 3000tr/min. C'est un moteur à courant continu et à aimant permanent.

Il est généralement utilisé en bricolage, pour des machines de polissages et autres équipements mécaniques. Ainsi, ce moteur a un couple élevé et n'aura pas de mal à faire tourner les pâles de notre hélice holographique. Ce moteur nous permettra donc de faire tourner l'hélices tout en ayant une vitesse de rotation suffisante pour répondre à nos besoins.



Figure 7 : Second choix de moteur

2.4 Transmission du contact en rotation

Maintenant que nous avons notre bandeau de LED et notre moteur nous allons avoir un problème lié à l'approvisionnement en électricité pour l'hélice. Les LEDs, qui tourneront à grande vitesse vont avoir besoin d'être alimenté pour pouvoir s'allumer et pour transmettre la bonne couleur. Alors, nous avons besoins de plusieurs fils électriques :

- 2 pour le +5v : Un par bandeau de LED
- 2 pour la masse : Un par bande de LED
- 2 pour les données : Un par bande de LED

Le problème est qu'avec la vitesse de rotation nous ne pouvons pas faire passer ces câbles sinon nous allons avoir un souci de nœuds. Nous avons donc pensé à deux solutions :

- **1^{ère} solution :**

Utiliser une batterie au niveau des pâles (sur la partie en rotation) et alimenter directement les bandeaux à l'aide de cette batterie. Le problème est qu'il faudra mettre en place aussi un système de communication sans fil pour la récupération des données et aussi que la batterie pèse un poids important ce qui implique un couple de moteur encore plus élevé. De plus, une batterie a besoin de se recharger, ainsi, il sera impossible de faire tourner l'hélice en continu sur du long terme.

- **2^{ème} solution :**

Cette solution est plus simple à mettre en place. Elle consiste à utiliser une pièce appelé « bague collectrice » afin de transmettre le courants électriques et les données entre la partie fixe et la partie en rotation. Pour notre prototype nous avons opté pour choisir une bague collectrice pleine. Elle fonctionne comme une bague collectrice standard mais elle est conçue pour être utilisée dans des applications où l'arbre tourne à des vitesses plus élevées. Cette bague collectrice pleine est entièrement recouverte de balais, ce qui permet une meilleure récupération de l'énergie électrique. Cela signifie qu'il y a plus de contact entre les balais et l'arbre, ce qui augmente la quantité d'énergie produite. En outre, la bague collectrice pleine peut être fabriquée avec des matériaux plus résistants et durables pour résister aux forces de frottement plus élevées générées par les vitesses plus élevées.

Pour notre projet nous avons donc trouvé plus judicieux de partir sur la seconde solution, et donc d'utiliser une bague collectrice pleine :



Figure 8 : bague collectrice pleine

2.5 Utilisation d'une courroie

En effet, nous avons dû utiliser une courroie pour utiliser la bague collectrice étant donné que la tige de notre moteur n'était pas à arbre creux.

Nous avons donc répliqué la rotation de la tige du moteur sur une tige à arbre creux comme vous pouvez voir sur la photo de droite.

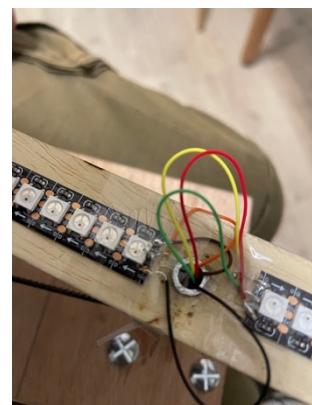
La tige du moteur (tige du haut) relie alors la tige creuse (tige du bas) à l'aide d'une courroie (en noir). Et, nous venons fixer notre double pâle sur la tige creuse.



Nous venons par la suite fixer la bague collectrice à l'arrière de la tige creuse de manière à faire passer les câbles pour alimenter et envoyer les données aux bandeaux de LEDs.

2.6 Quelques soudures

Maintenant vient la partie pour lier l'alimentation est les données à chaque bandeau de LEDs. Pour cela nous avons fait quelques soudures :



2.7 Mesure de la vitesse de rotation

Figure 10: Quelques soudures

Une fois l'assemblage du bloc moteur terminé, il était nécessaire de vérifier que la vitesse de rotation serait satisfaisante. Pour cela, nous avons utilisé un ralenti très puissant pour mesurer la vitesse de rotation sur une seconde. Cela nous a permis d'obtenir des données précises sur la vitesse à laquelle l'arbre tournait à un moment donné. Nous avons obtenu une vitesse de 6.5 tr/min.

Nous avons ensuite retransmis cette vitesse sur une minute en multipliant la vitesse obtenue sur une seconde par soixante. $6.5 * 60 = 390\text{tr/min}$. On obtient alors une vitesse de 390tr/min.

3 Conception électronique

3.1 Alimentation

Nous avons choisi les alimentations en fonction des besoins que nous aurions en termes de tensions électriques. Nous avons 3 éléments à alimenter en courant :

- Le moteur à courant continu, celui fonctionne sous 5-12-24V, en fonction de cette tension la vitesse de rotation sera plus ou moins élevée.
- Le Raspberry Pi fonctionne lui sous 3.3V
- L'Arduino UNO qui sera alimenté directement à l'aide du Raspberry
- Les deux bandeaux de LEDs qui doivent être alimentés en 5V
- Capteur de présence 5V

- Moteur :

Pour l'alimentation du moteur nous avons choisi d'utiliser une alimentation numérique 1 voie.

Une alimentation numérique est un type d'alimentation électrique pour convertir l'énergie électrique de sortie d'un réseau de distribution en une tension et un courant régulé pour alimenter des équipements électroniques. Les alimentations numériques sont souvent utilisées pour des applications qui nécessite une régulation précise de la tension et du courant, ce qui dans notre cas est très utiles pour ajuster la vitesse du moteur.

Voici une photographie de notre générateur de courants et de tensions. Nous avons trouvé une vitesse idéale pour notre affichage à 2A et 4.8V.



Figure 11: générateur de tensions

- Raspberry et Arduino UNO :

Nous utilisons directement l'alimentation fournie avec le Raspberry qui est une alimentation en 5VDC et 2A. Ensuite, à l'aide d'un câble USB to USB-typeB nous pouvons alimenter directement l'Arduino UNO.



Figure 12 : Alimentation
Raspberry/Arduino UNO

- Deux bandeaux de LED :

Pour les bandeaux de LED nous utilisons une alimentation 5V directement relié au 220V. Puis nous séparons en deux le 5V et la masse pour pouvoir obtenir deux alimentations de 5V. Ensuite nous relions les deux tensions 5V à 2 fils de la bague collectrice afin de pouvoir les lier ensuite au 5V et au GND(Ground) du bandeau de LED.

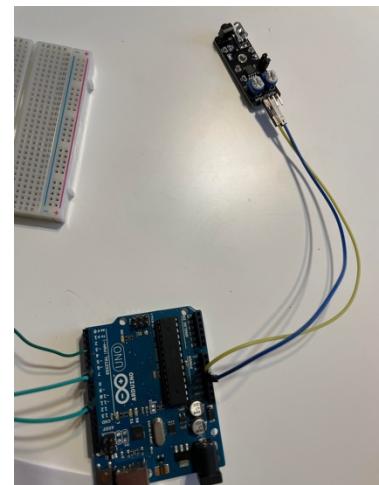


Figure 13 : Alimentation du capteur
de présence à travers l'Arduino UNO

3.2 Capteur de présence

La prise d'origine est un problème qui nous a beaucoup occupés dans le cadre de ce projet. La prise d'origine est essentielle pour que l'afficheur soit asservi : elle permet d'indiquer à l'Arduino, à chaque tour de l'hélice, la position (et donc l'instant) où doit commencer la séquence d'affichage pour que l'image soit affichée correctement.

Pour cela, nous avons utilisé le capteur de présence **KY-032** :



Figure 14 : Capteur de présence KY-032

Le capteur d'évitement d'obstacles Arduino KY-032 est un capteur de proximité infrarouge réglable en distance conçu pour les robots à roues. Aussi connu sous AD-032.

La distance de détection du capteur varie de 2 cm à 40 cm, elle peut être ajustée en tournant le bouton du potentiomètre. La tension de fonctionnement est de 3,3 V-5 V, elle convient donc à une variété de microcontrôleurs tels que Arduino, ESP32, Raspberry Pi et autres.

Il a une forte adaptabilité à la lumière ambiante et il est assez précis pour détecter les changements dans l'environnement environnant.

Le module possède une paire de LED infrarouges, un émetteur et un récepteur. La LED émettrice envoie des impulsions lumineuses infrarouges à une certaine fréquence. Lorsque la lumière frappe, un obstacle est renvoyé vers la LED du récepteur.

Ainsi, dès que la pâle passera devant le capteur infrarouge nous pourrons indiquer à l'Arduino que nous devons commencer une nouvelle séquence d'affichage.

Pour finir, ce capteur nous permettra aussi d'afficher l'image dans le bon sens car nous serons exactement à quel moment il faut afficher l'image.

3.3 Schéma électrique

Voici le schéma électrique de notre prototype :

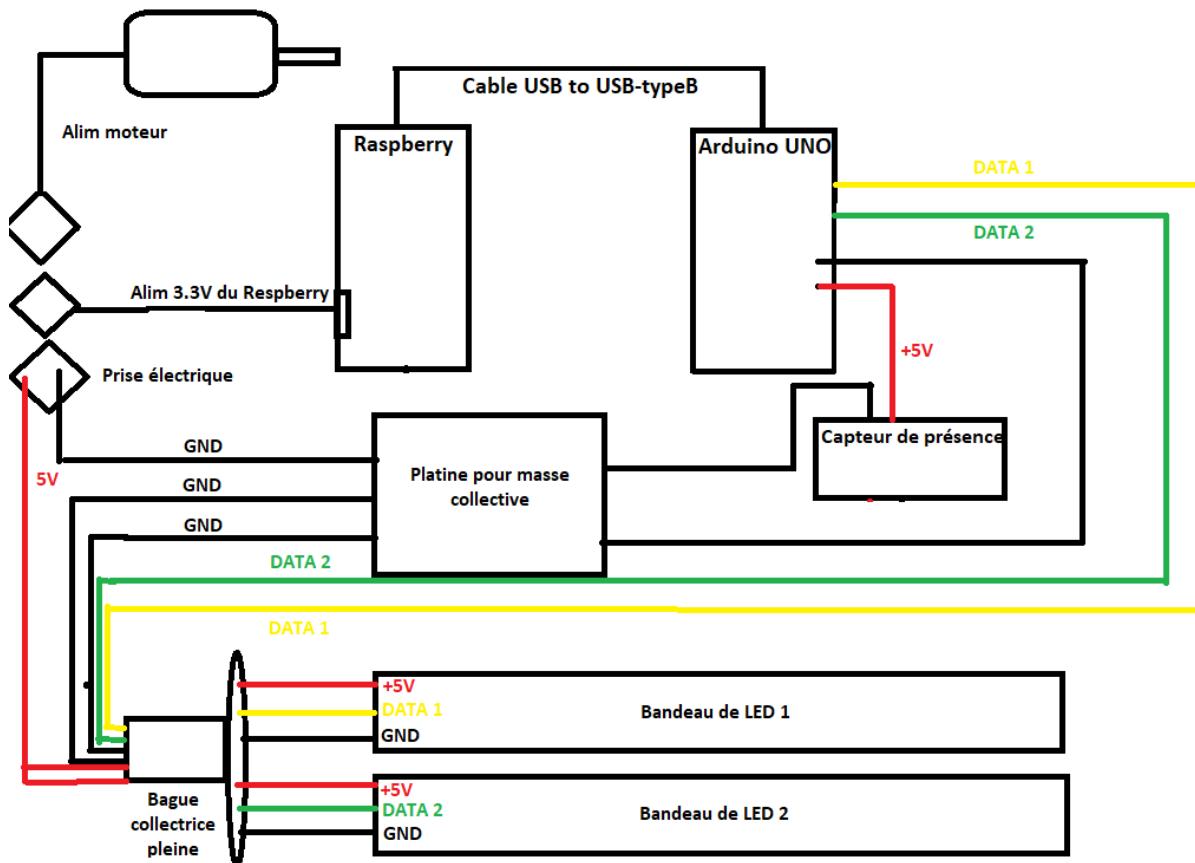


Figure 15: Schéma électrique de notre prototype

Légende du schéma :

— : +5V — : Ground (masse) — : Donnée du bandeau de LEDs 1

— : Donnée du bandeau de LEDs 2



Par la suite, nous avons trouvé plus judicieux de remplacer la platine par un connecteur de fils pour éviter l'encombrement des câbles.

Ainsi, on peut voir que le capteur de présence et les bandeaux de LEDs ont la même masse.

3.4 Prototype

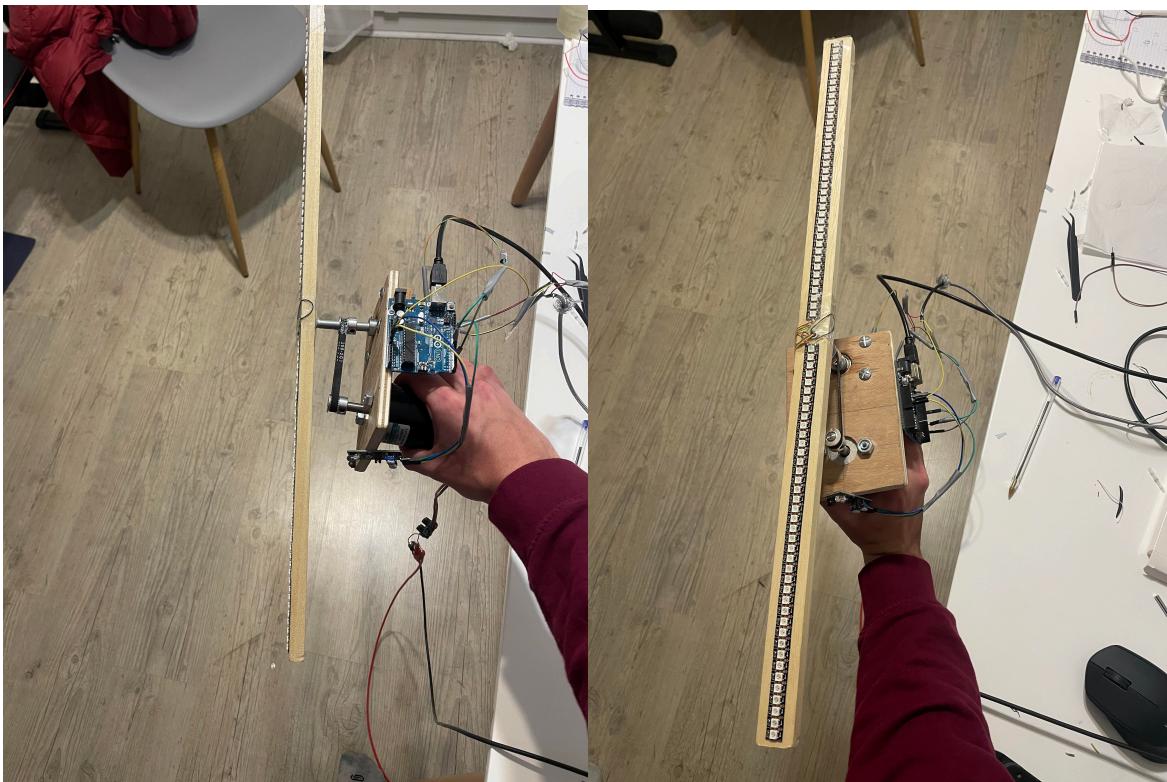


Figure 16: Photographies du prototype

Voici notre prototype final, nous avons utilisé une planche en bois en guise de pôle afin de pouvoir avoir une surface plate pour coller les bandeaux de LEDs dessus. Au début, notre pôle était métallique mais avec la vitesse de rotation du moteur nous avons cassés notre arbre creux au quelle elle était fixée, voici une photographie de notre pôle cassé :



Figure 17: Pôle métallique cassée

Voici maintenant quelques exemples de notre prototype en mouvement avec l'éclairage des LEDs :

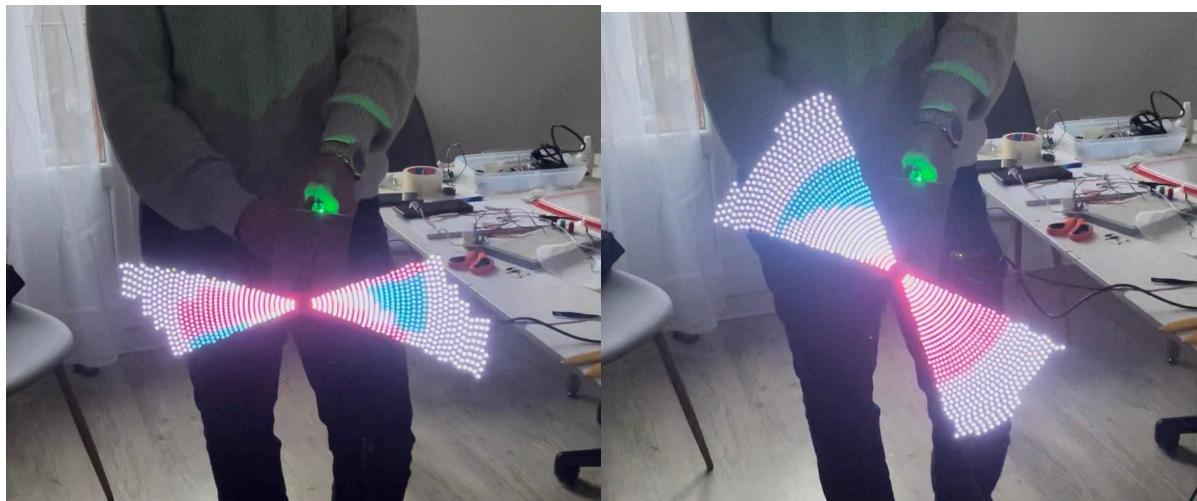


Figure 18: Prototype en mouvement

Malheureusement, les téléphones modernes disposent d'un appareil photo de haute résolution, mais ils ont encore des limites en ce qui concerne la qualité d'image. Bien que les images capturées par les téléphones soient généralement de très bonne qualité, elles ne sont pas toujours suffisamment détaillées pour capturer toutes les LEDs en rotation dans leur intégralité à cause de la vitesse de rotation et à la luminosité des LEDs.

Pour capturer des images de LEDs en rotation de haute qualité, il est recommandé d'utiliser un appareil photo professionnel avec des fonctionnalités de prise de vue rapide et une résolution élevée.

Malgré cette qualité d'image, on arrive quand même à visualiser une partie du logo pepsi (utilisé pour l'exemple).

4 Programmation du prototype

4.1 Description globale

Notre solution holographique consiste en l'utilisation d'un Raspberry Pi pour contrôler un Arduino qui allume les LED de l'hélice.

Le Raspberry Pi est utilisé pour gérer l'interface utilisateur et la communication avec l'Arduino, tandis que l'Arduino est utilisé pour contrôler les LED de l'hélice.

Le système fonctionne de la manière suivante :

- Le Raspberry Pi est connecté à internet, ce qui permet à l'utilisateur de contrôler l'hélice à distance en utilisant une interface web.
- Le Raspberry Pi exécute un mini serveur web qui gère l'importation de l'image de l'utilisateur via l'interface web et extrait les points de couleurs puis affiche la prévisualisation du rendu.

liser le renau

rendu :

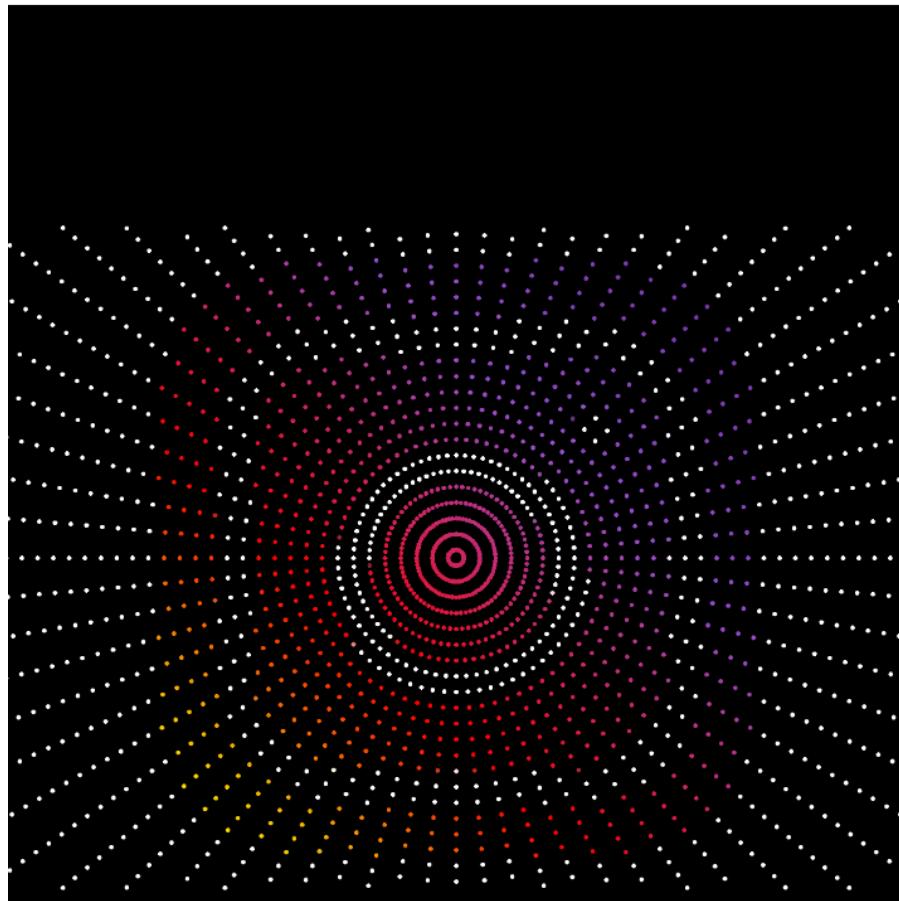


Figure 19: Visualisation des points de couleurs

- Le Raspberry Pi communique avec l'Arduino en utilisant la connexion USB pour envoyer les points de couleur à l'Arduino et exécuter le script d'allumage des leds.
- L'Arduino reçoit les commandes du Raspberry Pi et utilise celles-ci pour contrôler les LED de l'hélice.

4.2 Extraction points de couleur

Le programme d'extraction des points de couleur est écrit en python en utilisant la librairie PIL.

La méthode consiste à calculer d'abord les coordonnées polaires (θ , r) des points d'une grille circulaire de même taille que l'image, enregistrés dans un tableau.

Pour chaque point de cette grille, on convertit les coordonnées polaires en coordonnées cartésiennes (x , y), puis on récupère la couleur du pixel de l'image correspondant à ces coordonnées (x , y) sous forme d'un triplet (r , g , b) qui est enregistré dans le même tableau.

Ensuite, on trie ce tableau en fonction des secteurs puis on enregistre le résultat final dans un fichier.h.

Extrait du code d'extraction des points :

```

sourceImg = Image.open(imageFileName)
sourceImg.convert('RGBA')
imgMaxSize = max(sourceImg.size)
baseImgSize = imgMaxSize*100//zoomFactor
centeredImage = Image.new('RGB', (baseImgSize,)*2, 'black')

centeredImage.paste(sourceImg, (
    baseImgSize-sourceImg.size[0])//2, (baseImgSize-sourceImg.size[1])//2))
centeredImage.save('TEST.png')

pickingPoints = getPickingPoints(
    baseImgSize, nbEllipsesPerDiametralLine, angleStep)
pickedColors = pickColors(pickingPoints, centeredImage)

saveArrayPickedColorsInFile(
    nbSectors, nbEllipsesPerDiametralLine/2, pickedColors)

renderPickedPointsPreview(outputFileName, pickedColors, 10)

```

Figure 20: Code extraction des points

4.3 Affichage des points de couleur sur l'hélice

4.3.1 Librairie FastLed pour arduino

FastLED est une librairie pour Arduino qui permet de contrôler facilement des LED RGB ou des LED RGBW (rouge, vert, bleu et blanc) en utilisant un grand nombre de protocoles de communication tels que SPI, UART, OneWire, et plus encore. Cette librairie offre une grande flexibilité et une grande performance.

FastLED fournit également des fonctions pour gérer les couleurs en utilisant différents espaces de couleur (HSV, RGB, etc.). Il est compatible avec un grand nombre de contrôleurs de LED populaires tels que WS2811, utilisés dans notre projet.

En utilisant FastLED, on a pu créer le script qui va contrôler les LEDS de l'hélice holographique et afficher les points de couleurs stockées dans un fichier.h.

4.3.2 Problématique mémoire RAM de l'Arduino : utilisation de PROGMEM

L'Arduino possède une mémoire RAM limitée, et si on génère un nombre de points de couleurs correspondant à 28 Leds par pale et 16 secteurs cette mémoire se voit saturée et il peut ne plus y avoir suffisamment de mémoire pour d'autres tâches importantes.

Pour résoudre ce problème, on a utilisé la fonctionnalité PROGMEM d'Arduino. PROGMEM est une fonctionnalité qui permet de stocker des données en mémoire flash plutôt qu'en RAM. La mémoire flash a une capacité plus grande que la RAM, et elle conserve les données même après la mise hors tension de l'Arduino.

Pour utiliser PROGMEM, on a dû stocker vos données dans une variable déclarée comme étant en mémoire flash, en utilisant le mot clé "const uint8_t variableNom PROGMEM".

```
fillStripWithImage.io      test_color_array.h
#define NUM_LEDS 35
#define NB_SECTORS_USED 36
const uint8_t colorsForStrip[36][35][3] PROGMEM ={{{{208, 36, 92}, . . .
const uint8_t colorsForStrip2[36][35][3] PROGMEM ={{{{204, 39, 99}, . . .
```

Ensuite on a également utilisé une fonction spéciale pour accéder aux données stockées en mémoire flash, comme pgm_read_byte().

```
.d FillStrip1WithDifferentColors( CRGBArray<NUM_LEDS> leds, uint8_t :
    for(int i = 0; i < NUM_LEDS; i++) {
        leds[i].setRGB(pgm_read_byte_near(&colorsForStrip[sector][i][0]), . . .
```

4.3.3 Allumage des LEDs

Le script d'allumage des LEDs est relativement simple.

Il se décompose en les étapes suivantes exécutés en boucle :

- 1- Vérifier si une interruption de broche 10 est détectée (Hélice au point 0).
- 2- Si oui, obtenir le temps actuel en microsecondes et calculer la vitesse de rotation de l'hélice en utilisant la formule : vitesse = 1 000 000 / (temps actuel - temps précédent) * 60.
- 3- Afficher la vitesse sur le Moniteur série (console Arduino).
- 4- Mettre à jour le temps précédent pour la prochaine interruption.
- 5- Utiliser une boucle for pour parcourir les secteurs utilisés de l'hélice.
- 6- Dans la boucle for, remplir chaque secteur avec des couleurs différentes en utilisant les fonctions FillStrip1WithDifferentColors et FillStrip2WithDifferentColors en fonction de la variable swap.
- 7- Afficher les LED avec la fonction FastLED.show().

8- Inverser la valeur de swap pour inverser les hélices à la prochaine itération de la boucle.

```

void loop(){
    if(!digitalRead(10)) // Début de séquence d'affichage
    {
        unsigned long currentSensorTime = micros(); // Temps actuel
        speed = 1000000.0 /(currentSensorTime - lastSensorTime) * 60;
        Serial.print("Vitesse : ");
        Serial.println(speed);

        lastSensorTime = currentSensorTime; // Mettre à jour le temps
        for (uint8_t sector = 0; sector < NB_SECTORS_USED; sector++)
        {
            if(swap) {
                FillStrip1WithDifferentColors(leds2, sector);
                FillStrip2WithDifferentColors(leds, sector);
            } else {
                FillStrip1WithDifferentColors(leds, sector);
                Fillstrip2WithDifferentColors(leds2, sector);
            }
            FastLED.show();
        }
        swap = !swap ;
    }
}

```

Figure 21: Code inversement de swap

4.4 Interface web

L'interface web est développé grâce au Framework web Flask du langage Python. Elle se présente comme une page web unique. Un premier bouton permet d'uploader une photo au format png ou jpg.

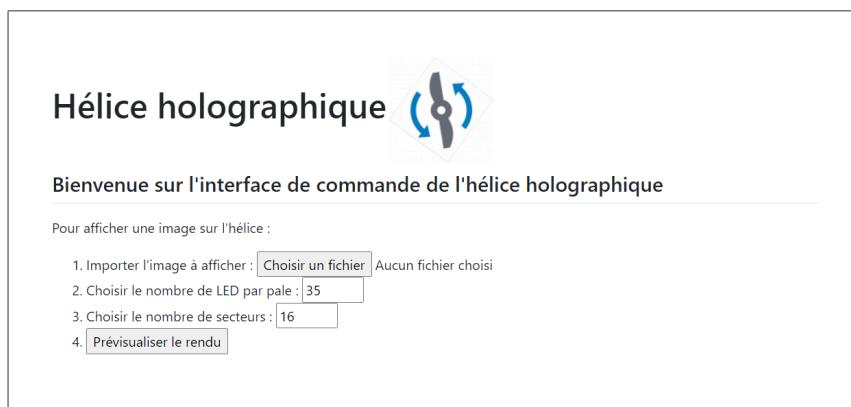


Figure 22: Interface Web

Ensuite, on sélectionne le nombre de LEDs par pale et le nombre de secteurs souhaités.

Hélice holographique 

Bienvenue sur l'interface de commande de l'hélice holographique

Pour afficher une image sur l'hélice :

1. Importer l'image à afficher : Instagram-logo.jpg
2. Choisir le nombre de LED par pale :
3. Choisir le nombre de secteurs :
4. Prévisualiser le rendu.

Figure 23: Déclaration du nombre de LEDs

Au clic sur le bouton “prévisualiser le rendu”, une image de prévisualisation des points de couleurs extraits est affichée ainsi que les boutons Démarrer l'hélice qui va enregistrer les points de couleurs sur l'Arduino puis exécuter le script d'affichage.

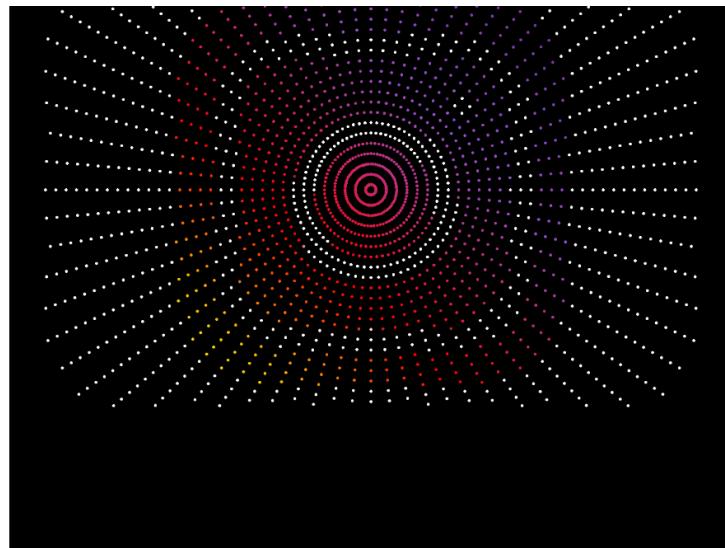


Figure 24: Prévisualisation du rendu

5 Propositions d'amélioration

Durant toute la durée du projet nous avons dû faire des choix de conception, certains nous ont été imposés soit parce que nous n'avions pas imaginé un futur problème, soit pour des raisons financières. Dans cette partie, nous allons expliquer les différents moyens d'amélioration de notre prototype.

Moteur à arbre creux :

Comme nous l'avons vu, l'un des choix de conception a été de passer par une courroie de façon à pouvoir faire passer les câbles reliant l'Arduino aux LED. Dans cette optique, nous pouvons imaginer une amélioration. Il est, en effet, possible de faire passer les câbles directement par le moteur, cela impliquerait d'avoir un moteur à arbre creux, dans ce cas, la conception de l'hélice aurait été plus facile à mettre en œuvre.

LED centrale :

Lors de la mise en marche du prototype, nous pouvons remarquer que le point du centre ne possède pas de LED ce qui a pour effet de laisser un point noir au centre. La méthode de correction que nous avons imaginée consiste à ne s'appuyer que sur un bandeau de LED recouvrant l'entièreté des deux pâles. Pour cela, il faudrait faire passer un câble derrière la pâle à la sortie de la bague collectrice de façon à relier les LED par une extrémité.

Il existe cependant un problème à cette solution :

Cela augmenterait le temps de transmission des données à la dernière LED. Dans ce cas-là, un décalage pourrait se créer dans l'image. Cela reste néanmoins difficilement remarquable au vu des images que nous affichons aujourd'hui. Il est aussi important de préciser que les micro-LED n'auront plus ce problème, car elles sont branchées en parallèle.

Miniaturisation :

L'un des problèmes les plus évidents, lorsqu'on observe notre prototype, est sa taille. En effet, la principale amélioration que nous imaginons est sa réduction pour cela, nous pouvons utiliser des Micro-LED à la place des LED.

- Micro-LED au lieu des LED :

Les micro-LED sont plus petites que les LED traditionnelles, ce qui les rend plus faciles à utiliser pour les applications nécessitant de nombreux points lumineux très proches les uns des autres, comme les hélices holographiques. Les micro-LED ont également un meilleur rendement énergétique et une durée de vie plus longue, ce qui les rend plus efficaces et plus fiables pour une utilisation dans des applications à long terme. En outre, ils peuvent être plus facilement intégrés dans des dispositifs plus petits et plus fins, ce qui est souhaitable pour les applications portables et de réalité augmentée.

- Amélioration de la dépense énergétique :

En plus des micro-LED, la réduction de la taille peut permettre une économie énergétique au niveau du moteur en effet réduire la taille veut aussi dire réduire les pales et donc une réduction du couple du moteur pour atteindre la vitesse voulue.

Utilisation du vide :

D'une manière plus avancée, nous pourrions imaginer mettre le système sous vide cela aurait plusieurs avantages.

Tout d'abord, le vide peut éliminer les perturbations causées par la turbulence de l'air, qui peuvent affecter la qualité de l'image holographique. En utilisant un vide, on peut également réduire les réflexions de la lumière sur les surfaces de l'hélice, ce qui peut améliorer la qualité de l'image. En outre, le vide peut également améliorer la durée de vie des composants de l'hélice, car ils sont exposés à moins d'oxydation et de corrosion. Enfin, l'utilisation d'un vide peut également améliorer la performance de l'hélice en réduisant la résistance mécanique.

6 Conclusion

Pour conclure, nous avons grâce un Arduino UNO, un Raspberry PI 3, un bandeau LED, un moteur de 3000tr/min, une bague collectrice pleine et un capteur de présence IR créé un prototype d'hélice holographique fonctionnel.

Durant ce projet, nous nous sommes confrontés à de nombreuses problématiques que nous avons, pour certaines, pu résoudre.

Il y a plusieurs voies d'améliorations possibles, notamment les micro-LEDs qui permettront d'avoir des taux de contrastes plus élevés, de réduire la consommation d'énergie mais aussi d'avoir une plus grande densité de pixel et donc une meilleure résolution.

Le paramètre le plus important pour une hélice holographique est la vitesse optimale de rotation de l'hélice holographique. Celle-ci dépend de plusieurs facteurs, tels que la taille de l'hélice, la résolution des images affichées et les conditions d'environnements. En général, il est souhaitable que la vitesse soit suffisamment élevée pour que l'image soit visible et stable, mais pas si élevée pour éviter que l'image devienne floue ou difficile à percevoir. Dans notre cas, nous arrivons à avoir une image acceptable à partir de 500tr/min même si cela pourrait être amélioré.

Il est aussi important de se pencher sur la conception de la pâle afin de limiter le poids et les frottements tout en gardant assez de rigidité pour résister à la vitesse.

Afin de terminer ce rapport nous voudrions parler de ce que nous a apporté ce projet. Ce projet étant principalement un défi mécanique et électrique cela nous a conduit à prendre des décisions basées sur des facteurs qui, dans l'informatique, sont moins mis en avant comme les finances ou les problèmes de casse. En effet, là où dans notre domaine d'expertise une casse n'implique rien de grave mise à part le temps de corriger le bug. Une casse en mécanique est bien plus contraignante puisqu'il nous faut trouver une autre pièce ou une autre méthode. En conclusion ce projet nous a apporté une vision plus planificatrice mais aussi plus inventive qu'un projet 100% informatique.

Si aujourd'hui nous pouvions refaire ce projet plein de choix serait différent mais nous sommes fiers du résultat produit.