

الجمهورية الجزائرية الديمقراطية الشعبية

وزارة التعليم العالي والبحث العلمي

UNIVERSITÉ BADJI MOKHTAR - ANNABA
BADJI MOKHTAR – ANNABA UNIVERSITY



جامعة باجي مختار – عنابة

Faculté: TECHNOLOGIE

Département : Electronique

Domaine: SCIENCES ET TECHNOLOGIES

Filière: Automatique

Spécialité: Automatique et système

Mémoire

Présenté en vue de l'obtention du Diplôme de Master

Thème:

Réalisation et conception d'un scanner 3D

Présenté par : *Chelli Imad Eddine*

Boukhentef Abdelkrim

Encadrant : *Gherbi Sofiane*

Grade : *Professeur*

U.B.M.A

Jury de Soutenance :

Benmoussa Samir	MCA	U.B.M.A	Président
Gherbi Sofiane	PROF	U.B.M.A	Encadrant
Chaker Karima	MCB	U.B.M.A	Examineur

Année Universitaire : 2022/2023

Remerciement

Avant tout, nous remercions Dieu, le Miséricordieux, pour toutes les grâces qu'Il nous a accordées, pour Sa bienveillance constante tout au long de notre vie et en particulier au cours de ces cinq dernières années. Nous Lui exprimons notre profonde gratitude pour nous avoir guidés et protégés à chaque étape de ce parcours.

Nous souhaitons adresser nos sincères remerciements à nos parents, qui méritent toute notre gratitude et notre reconnaissance. Leur amour, leur soutien inconditionnel et leur dévouement ont été des piliers essentiels dans notre parcours académique et personnel.

Leur encouragement constant, leurs sacrifices et leur confiance en nous ont été une source de motivation et de force indéniables.

*De plus, nous tenons à exprimer notre profonde gratitude et notre immense honneur à Monsieur **Gherbi Sofiane** professeur à l'université Badji Mokhtar d'Annaba pour son encadrement exceptionnel et sa guidance précieuse tout au long de cette mémoire. Nous considérons sa présence et son soutien comme une bénédiction, qui ont joué un rôle essentiel dans la réussite de ce travail.*

Je tien aussi, à remercier les membres des jurys de me faire l'honneur de lire et d'évaluer ce travail

Nous souhaitons exprimer notre sincère appréciation envers tous les enseignants du département d'électronique pour leur dévouement sur le travail. Leur expertise et leur engagement ont été d'une valeur inestimable dans notre parcours académique.

Dédicace

Nous dédions ce travail

À nos chers parents, qui ont joué un rôle essentiel dans nos vies. Votre soutien inconditionnel, votre guidance et votre dévouement ont été une source d'inspiration inestimable tout au long de notre parcours.

À nos frères et sœurs, dont les encouragements constants nous ont donné la force et la détermination pour persévérer dans nos efforts.

À nos amis et camarades, pour les bons moments partagés et leur soutien précieux tout au long de ce projet.

Nous n'oublions pas de mentionner M. Gherbi Sofiane pour tous ses conseils avisés et sa disponibilité.

Table des matières

Remerciement	I
Dédicace	II
Table des matières	III
Liste des figures	VI
Liste des abréviations	VIII
Résumé	IX
Abstract	X
ملخص	XI
Introduction générale	XII

Chapitre 1 : Généralités sur les scanners d'objets en 3D

1.1	Introduction.....	2
1.2	Diffèrent type de scanner	2
1.2.1	Scanneur 3D du bureau:.....	2
1.2.2	Scanneur 3D portable.....	3
1.2.3	Scanners 3D professionnels/industriels et métrologie	3
1.2.4	Applications mobiles scanner 3D avec un smartphone	3
1.3	Les types de capteurs utilisés dans les Scanner 3D	3
1.4	Les domaines d'utilisations.....	4
1.5	Les technologies de numérisation 3D.....	5
1.5.1	Le scan laser par triangulation optique	6
1.5.2	Le scan a la lumière structurée.....	7
1.5.3	La photogrammétrie.....	7
1.6	Type de données	7
1.7	Conclusion	9

Chapitre 2 : Conception et réalisation d'un scanner 3D

2.1	Introduction.....	11
2.2	Matérielle utilisée.....	11
2.2.1	Carte Arduino Uno.....	11
2.2.2	Le module de pilote A4988.....	12
2.2.3	Le moteur NEMA 17	13
2.2.4	Le capteur de distance Sharp GP2Y0A41SK0F	14
2.2.5	Le micro SD shield	15
2.2.6	Module Bluetooth HC-06	16
2.2.7	Interrupteur (switch) et bouton poussoir.....	17
2.2.8	Schéma du câblage final du projet.....	17
2.2.9	Circuit imprimé.....	18
2.2.10	Partie mécanique.....	19
2.3	Logiciel utilisée.....	20
2.3.1	Arduino IDE.....	20
2.3.2	Fritzing.....	20
2.3.3	MeshLab	21
2.3.4	SolidWorks	21
2.4	Schéma fonctionnel du scanner 3D.....	21
2.5	Conception du scanner 3D.....	22
2.5.1	L'assemblage toutes les pièces.....	24
2.6	Test et résultat.....	25
2.7	Conclusion	27

Chapitre 3 : Développement d'une application mobile pour la commande du scanner 3D

3.1	Introduction.....	29
3.2	La plateforme MIT App inventor	29
3.3	Utilisation dans notre projet.....	31

3.3.1	L'espace designer.....	31
3.3.2	L'espace Blocs	35
3.4	Test et résultat	38
3.4.1	Test réel de l'application.....	40
3.5	Conclusion	42
	Conclusion générale et perspective.....	43
	Bibliographie	44
	ANNEXE.....	48

Liste des figures

Chapitre 1

Figure 1. 1. Numérisation 3D d'éléments mécaniques et de pièces industrielles	5
Figure 1. 2. Numérisation 3D d'éléments orthopédiques	5
Figure 1. 3. Scanner de numérisation 3D avec contact par bras robotisé	6
Figure 1. 4. les techniques du numérisation 3D.....	6
Figure 1. 5. principe du scanner par triangulation optique	7

Chapitre 2

Figure 2.1. Tous les composant d'arduino Uno	11
Figure 2. 2. Le module A4988.....	12
Figure 2. 3. Schéma de principe de montage A4988	13
Figure 2. 4. Moteur NEMA17.....	13
Figure 2. 5. L e capteur Sharp GP2Y0A41SK0F.....	14
Figure 2.6. Graphe analogie distance/voltage du capteur Sharp GP2Y0A41SK0F	15
Figure 2. 7. La carte de micro SD shield	15
Figure 2. 8. Câblage plaque d'essai/Arduino/ micro SD shield.....	16
Figure 2. 9. Le module HC-06.....	16
Figure 2. 10. Bouton poussoir.....	17
Figure 2. 11. Capteur fin de course.....	17
Figure 2. 12. représente le câblage final de matérielle électrique.....	18
Figure 2. 13. le circuit électronique du projet.....	18
Figure 2. 14. Photo réel de la tige 8mm avec l'écrou anti jeu	19
Figure 2. 15. Axe linéaire de 8mm et 200mm	19
Figure 2. 16. Roulement de 8mm.....	20
Figure 2. 17. Schéma fonctionnel du scanner 3d.....	22
Figure 2. 18. La création du modèle 3D du support des moteurs	23
Figure 2. 19. Le modèle 3D de fixateur de capteur	23
Figure 2. 20. Modèle 3D de coupleur d'arbre	23
Figure 2. 21. Le modèle 3D de plateau tournant.....	23
Figure 2. 22. Modèle 3D de la plaque de fixation	24
Figure 2. 23. La conception finale du scanner 3d.....	24
Figure 2. 24. Photo réelle du scanner d'objet 3D réalisé.....	25
Figure 2. 25. Objet de test réel	25
Figure 2. 26. Données obtenues après le scan représentées par le logiciel MeshLab	26
Figure 2. 27. Résultat obtenu après traitement des données par MeshLab.....	26

Chapitre 3

Figure 3. 1. L'espace designer.....	30
Figure 3. 2. L'espace bloc	30
Figure 3. 3. Emulateur de MIT App	31
Figure 3. 4. HorizontalArrangements et leurs propriétés.....	32
Figure 3. 5. ListPicker.....	32
Figure 3. 6. HorizontalArrangements et l'étiquette "Not connected"	33
Figure 3. 7. L'étiquette MOTOR 'OFF'	33
Figure 3. 8. Les boutons turn On et Turn Off	34
Figure 3. 9. Les composant non visible	34
Figure 3. 10. Groupe de blocs pour les appareils Bluetooth.....	35
Figure 3. 11. Groupe de blocs pour sélectionner notre Bluetooth	35
Figure 3. 12. Groupe de blocs de 'timer'+ statut du Bluetooth	36
Figure 3. 13. Groupe de blocs de bouton TURN : ON	36
Figure 3. 14. Groupe de blocs de bouton TURN: OFF.....	36
Figure 3. 15. Organigramme de l'application mobile.....	37
Figure 3. 16. Interface d'application scanner 3D.....	38
Figure 3. 17. Les blocs de l'application	38
Figure 3. 18. Le code QR de l'application	39
Figure 3. 19. L'icône de l'application scanner_3D sur le smartphone	39
Figure 3. 20. Interface de l'application	40
Figure 3. 21. Sélection du réseau Bluetooth	40
Figure 3. 22. Boutons de sélection marche/arrêt des moteurs	41
Figure 3. 23. Statuts des moteurs et du scan	41

Liste des abréviations

3D	Trois dimension
MMT	Machine à mesurer Tridimensionnelles
ICSP	In-circuit serial programming
SRAM	Static Random-Access Memory
EEPROM	Electrically Erasable Programmable Read-Only Memory
CNC	Computer Numerical Control
SD	Secure Digital
SPI	Serial Peripheral Interface
IDE	Integrated Development Environment
PCB	Printed Circuit Board
CAO	Conception Assistée par Ordinateur
MIT	Massachusetts Institute of Technology
JVM	machine virtuelle java
API	application programming interfaces
ASCII	American standard code information interchange

Résumé

Ce mémoire de fin d'études présente une étude approfondie sur les scanners d'objets en 3D, la conception et la réalisation d'un scanner 3D fonctionnel, ainsi que le développement d'une application mobile de commande associée. Il explore les différents types de scanners 3D, les technologies de numérisation utilisées et les domaines d'application. Le scanner 3D développé utilise des composants matériels tels qu'une carte Arduino Uno, un module de pilote A4988, un moteur NEMA 17 et un capteur de distance Sharp GP2Y0A41SK0F, et est contrôlé par une application mobile créée avec la plateforme MIT App Inventor. Les tests réalisés ont validé le bon fonctionnement du scanner 3D et de l'application mobile. En conclusion, ce mémoire fournit une vision complète des scanners 3D et présente une solution concrète pour la numérisation tridimensionnelle d'objets, tout en mettant en évidence les possibilités d'amélioration future dans ce domaine en constante évolution.

Abstract

This dissertation presents an in-depth study of 3D object scanners, the design and implementation of a functional 3D scanner, and the development of an associated mobile application for control. It explores different types of 3D scanners, the utilized scanning technologies, and their application domains. The developed 3D scanner incorporates hardware components such as an Arduino Uno board, an A4988 driver module, a NEMA 17 motor, and a Sharp GP2Y0A41SK0F distance sensor, while being controlled by a mobile application created using the MIT App Inventor platform. The conducted tests have confirmed the proper functioning of the 3D scanner and the mobile application. In conclusion, this thesis provides a comprehensive overview of 3D scanners and presents a concrete solution for three-dimensional object scanning, while highlighting potential avenues for future improvement in this ever-evolving field.

تقدم هذه الأطروحة دراسة مفصلة لمساحات الكائنات ثلاثية الأبعاد، وتصميم وتنفيذ ماسح ثلاثي الأبعاد عملي، بالإضافة إلى تطوير تطبيق جوال مرتبط للتحكم. يستكشف هذا البحث أنواعًا مختلفة من مساحات ثلاثية الأبعاد، وتقنيات المسح المستخدمة، ومجالات تطبيقها. يتضمن الماسح ثلاثي الأبعاد المطور مكونات هارديوير مثل لوحة Arduino Uno، ووحدة تشغيل A4988، ومحرك NEMA 17، ومستشعر المسافة Sharp GP2Y0A41SK0F، بينما يتم التحكم فيه عن طريق تطبيق جوال تم إنشاؤه باستخدام منصة MIT App Inventor. أكدت الاختبارات التي أجريت سلامة عمل ماسح ثلاثي الأبعاد وتطبيق الجوال. في الختام، يقدم هذا البحث نظرة شاملة عن مساحات ثلاثية الأبعاد ويقدم حلاً ملموساً لمسح الكائنات ثلاثية الأبعاد، مع التركيز على الإمكانيات المحتملة للتحسين في هذا المجال المتطور باستمرار.

Introduction générale

La numérisation tridimensionnelle des objets occupe une place prépondérante dans de nombreux domaines tels que l'ingénierie, l'architecture, la conception industrielle, la médecine, l'archéologie, et bien d'autres encore. Les scanners 3D, en permettant la capture précise et détaillée des formes et des textures des objets réels, offrent des possibilités infinies en termes de visualisation, d'analyse et de reproduction.

Ce mémoire de fin d'études se concentre sur l'étude, la conception et la réalisation d'un scanner 3D, ainsi que sur le développement d'une application mobile de commande associée. Il s'agit d'une exploration approfondie du domaine des scanners d'objets en 3D, visant à comprendre les différents types de scanners, les technologies de numérisation utilisées, les domaines d'application et les défis techniques.

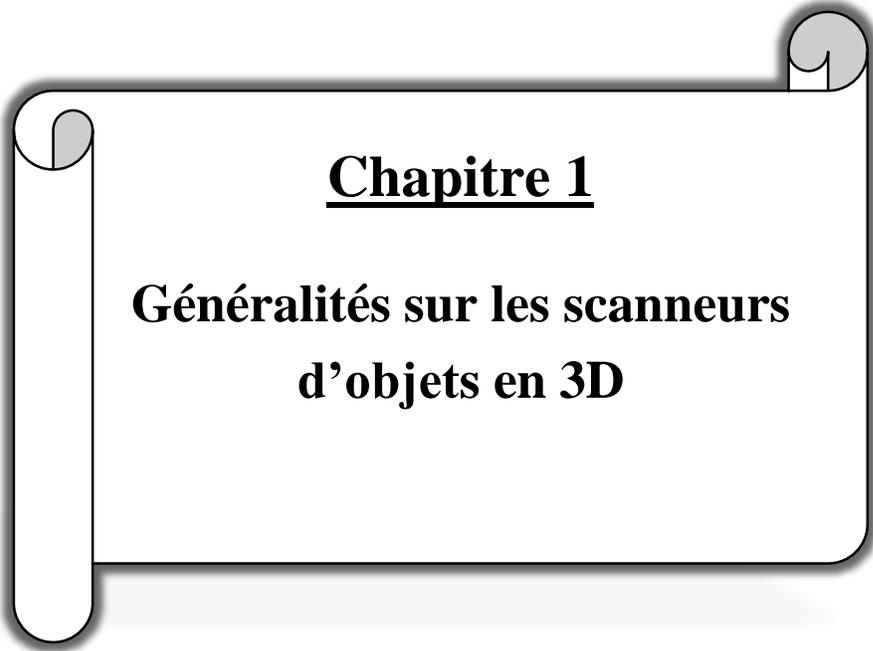
Le premier chapitre présente une vue d'ensemble des scanners d'objets en 3D. Nous examinons les différents types de scanners, tels que les scanners 3D de bureau, les scanners 3D portables, les scanners 3D professionnels/industriels et les applications mobiles de scan 3D avec un smartphone. Nous explorons également les types de capteurs utilisés dans les scanners 3D et les technologies de numérisation 3D, notamment le scan laser par triangulation optique, le scan à la lumière structurée et la photogrammétrie. De plus, nous analysons les domaines d'utilisation des scanners 3D et leur importance dans divers secteurs.

Le deuxième chapitre se concentre sur la conception et la réalisation d'un scanner 3D fonctionnel. Nous décrivons en détail les composants matériels utilisés, tels que la carte Arduino Uno, le module de pilote A4988, le moteur NEMA 17, le capteur de distance Sharp GP2Y0A41SK0F, et d'autres éléments essentiels. Nous présentons également les logiciels utilisés pour la programmation et la modélisation 3D, tels que Arduino IDE, Fritzing, MeshLab et SolidWorks. Nous expliquons le schéma fonctionnel du scanner 3D et détaillons la conception et le montage des pièces. Enfin, nous présentons les tests réalisés et les résultats obtenus pour évaluer les performances du scanner 3D.

Le troisième chapitre aborde le développement d'une application mobile dédiée à la commande du scanner 3D. Nous nous appuyons sur la plateforme MIT App Inventor pour concevoir et développer cette application. Nous détaillons les différentes étapes de conception dans l'espace Designer et l'espace Blocs, mettant l'accent sur les

fonctionnalités de contrôle du scanner 3D. Des tests réels sont ensuite effectués pour évaluer l'efficacité et la convivialité de l'application.

Enfin, nous espérons que cette étude contribuera à vulgarisation des technologies des scanners 3D et inspirera de nouvelles études dans ce domaine en constante évolution.



Chapitre 1

Généralités sur les scanners d'objets en 3D

1.1 Introduction

Les dispositifs de scanners 3D d'objets sont utilisés pour enregistrer les informations tridimensionnelles d'un objet physique et créer une représentation numérique en trois dimensions de celui-ci. Cette technologie permet de convertir des objets réels en modèles 3D exploitables sur un ordinateur. L'objectif principal de la numérisation 3D est de produire un modèle numérique précis et détaillé de l'objet, qui peut ensuite être utilisé dans divers domaines tels que la conception assistée par ordinateur, l'ingénierie inverse, la fabrication assistée par ordinateur, la réalité augmentée, la conservation du patrimoine et l'archéologie. Les scanners 3D d'objets sont devenus de plus en plus accessibles et polyvalents grâce aux progrès technologiques. Ils utilisent différentes méthodes de capture, comme la lumière structurée, la stéréophotogrammétrie ou le lidar, pour obtenir des données précises sur la forme et la texture d'un objet [2].

1.2 Différent type de scanner

Il existe plusieurs catégories de scanners 3D, selon leur usage, typologie, technologie de scan 3D et leur design. Des scanners 3D de bureau aux scanners 3D portables et sans-fil en passant par les scanners 3D professionnels et industriels et autres systèmes de mesure de haute précision On peut citer [3] :

1.2.1 Scanneur 3D du bureau:

Les scanners 3D de bureau, qu'ils soient de type fermé ou ouvert, sont généralement équipés d'un plateau rotatif ou d'une table tournante automatisée pour positionner l'objet à numériser. Pendant la numérisation en 3D, le plateau effectue une rotation afin de capturer l'objet sous tous les angles. Le scanner 3D lui-même, c'est-à-dire le capteur, reste fixe pendant ce processus. Le plateau tournant peut-être connecté au scanner 3D et contrôlé automatiquement grâce au logiciel associé. Il en existe deux types [3]:

1.2.1.1 Scanneurs 3D de bureau fermés

Ces dispositifs sont fréquemment employés dans des industries où il est nécessaire d'obtenir une résolution élevée lors de la numérisation d'objets de petite taille, comme la joaillerie ou la dentisterie. Les scanners fermés sont conçus sous la forme d'une boîte où l'utilisateur positionne l'objet à numériser, puis lance la numérisation 3D à partir d'un ordinateur équipé du logiciel approprié.

1.2.1.2 Scanneurs 3D de bureau ouverts

Les scanners sont spécialement conçus pour numériser des objets de plus grandes dimensions. Ils sont généralement équipés d'un bras articulé qui maintient la

caméra ou le capteur laser, ce qui permet une plus grande liberté de mouvement lors de la numérisation

3D. Ces scanners sont largement utilisés dans divers secteurs tels que l'architecture, l'ingénierie, le design et la production de mobilier.

1.2.2 Scanneur 3D portable

Les scanners 3D portables sans fil sont conçus pour être tenus à la main, l'utilisateur dirige le scanneur 3D vers l'objet ou le sujet à numériser en 3D et effectue une rotation autour de celui-ci pour capturer tous les détails de l'objet. Les scanners 3D portables sont à la fois pratiques et polyvalents, ce qui les rend idéaux dans de nombreuses situations. Ils sont faciles à manipuler et conviennent parfaitement pour effectuer des numérisations 3D dans des endroits difficiles d'accès.

1.2.3 Scanners 3D professionnels/industriels et métrologie

Les scanners 3D professionnels et industriels sont spécifiquement développés pour répondre à des exigences élevées de précision et de fiabilité. Ils sont largement utilisés dans des domaines tels que la rétro-ingénierie et l'inspection de pièces. Certains de ces scanners, également appelés Machines à Mesurer Tridimensionnelles (MMT), sont spécialement conçus pour des mesures de précision. Ils se basent principalement sur la capture précise des formes, en générant des modèles 3D détaillés sous forme de nuages de points. Ils ne capturent généralement pas les couleurs ni les textures.

1.2.4 Applications mobiles scanner 3D avec un smartphone

Les applications mobiles de scan 3D permettent d'utiliser l'appareil photo du smartphone ou de la tablette pour capturer des images sous différents angles et créer des modèles 3D. Elles sont disponibles pour iOS et Android, certaines étant gratuites. Cependant, leur précision et leur qualité sont limitées par rapport aux scanners 3D professionnels. La résolution de l'appareil photo et les conditions d'éclairage affectent également la qualité des modèles obtenus.

1.3 Les types de capteurs utilisés dans les Scanner 3D

Les scanners 3D peuvent être classés en fonction du type de capteurs utilisés pour capturer les données sur l'objet à numériser. Voici quelques types de capteurs couramment utilisés [4] :

➤ **Capteurs laser**

Ces scanners projettent un faisceau laser sur l'objet et mesurent la lumière réfléchie pour obtenir des informations sur sa géométrie. Les scanners à triangulation laser et les scanners à temps de vol sont des exemples de scanners utilisant des capteurs laser.

➤ **Capteur optique**

Ces scanners utilisent des caméras pour capturer des images de l'objet à partir de différents angles, puis les traitent pour créer un modèle 3D. Les scanners de photogrammétrie sont des exemples de scanners utilisant des capteurs optiques.

➤ **Capteurs à lumière structurée**

Ces scanners projettent des modèles de lumière structurée sur l'objet, qui sont déformés par sa surface pour calculer sa géométrie 3D. Les scanners à lumière structurée utilisent des capteurs à lumière structurée.

➤ **Capteurs ultrasons**

Ces scanners émettent des ondes sonores qui rebondissent sur l'objet, puis les captent pour calculer sa géométrie 3D.

➤ **Capteurs infrarouges**

Ces scanners utilisent des rayonnements infrarouges pour capturer des données sur la géométrie et les caractéristiques physiques de l'objet. Ils peuvent fonctionner dans des conditions de faible éclairage et capturer des détails fins et des textures sur la surface de l'objet. Cependant, ils peuvent avoir des difficultés avec les surfaces transparentes ou très réfléchissantes, et ils peuvent être plus coûteux que d'autres types de scanners 3D.

1.4 Les domaines d'utilisations

Les modèles 3D peuvent être utilisés dans divers domaines, tel que : l'ingénierie inverse (analyse du matériel dans les domaines commercial et militaire) [5] , le design industriel (création de pièces et d'objets industriels) (figure 1.1).

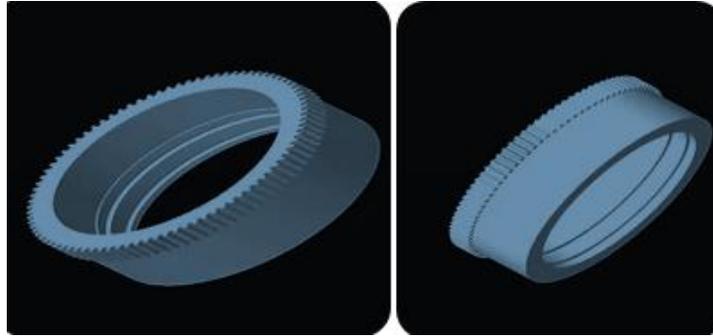


Figure 1.1. Numérisation 3D d'éléments mécaniques et de pièces industrielles [6]

L'archéologie où les scanners d'objet 3D peuvent être utilisés pour capturer des données de surface de sites archéologiques [3]. Le scanner 3D a aussi révolutionné la médecine permettant aux professionnels de la santé peuvent obtenir une représentation précise des structures anatomiques, ce qui facilite le diagnostic et le traitement des maladies, ils sont aussi utilisés pour la fabrication de prothèses sur mesure et la duplication d'organes (figure 1.2).

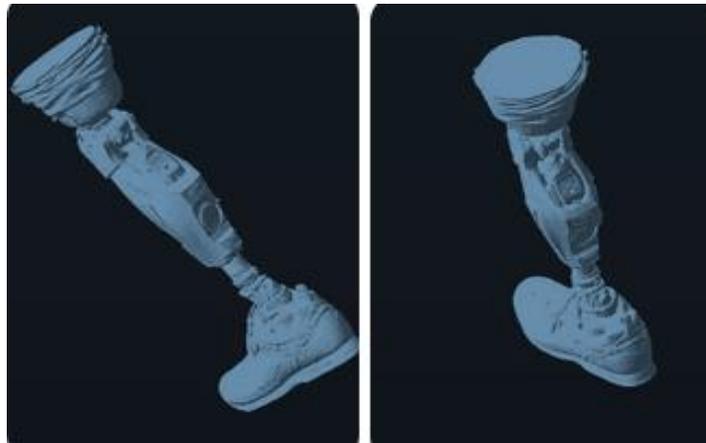


Figure 1.2. Numérisation 3D d'éléments orthopédiques [6]

Enfin, Les Scanner 3D sont aussi utilisés dans l'art et design et le divertissement.

1.5 Les technologies de numérisation 3D

Il existe différentes technologies de numérisation en 3D, elles sont classées en deux grandes catégories :

- **Les scanners à contact** : utilisant un capteur appelé "palpeur" ou "sonde" pour enregistrer chaque déformation de la surface de l'objet, (figure 1.3)

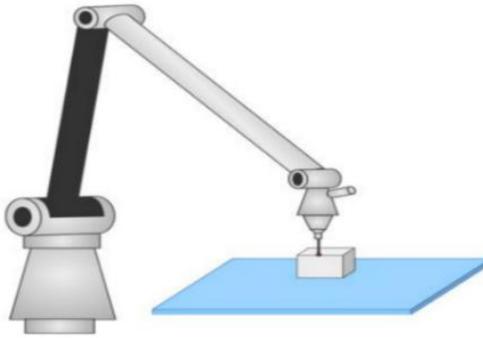


Figure 1. 3 . Scanneur de numérisation 3D avec contact par bras robotisé [7]

➤ **Les scanners sans contact :** Ces derniers sont divisés en deux :

Les Scanners sans contact actifs : projection d'une lumière et détection sa réflexion sur la surface de l'objet

Les scanners sans contact passifs : balayage de mesures à l'aide d'un laser, qui pour effectuer une télédétection laser par télémétrie [7]. En résumé ça dans la figure 1.4

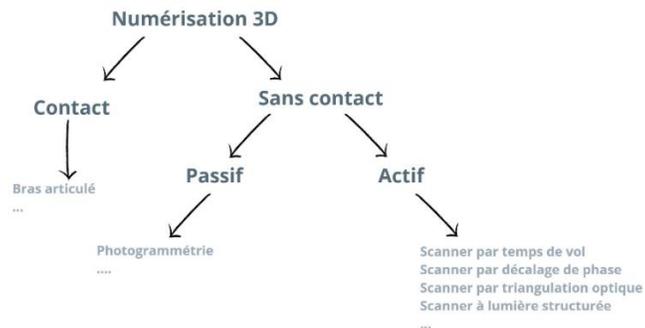


Figure 1. 4. les techniques du numérisation 3D [7]

1.5.1 Le scan laser par triangulation optique

Le scanner 3D projette un faisceau laser sur l'objet à numériser, la surface de ce dernier reflète le faisceau vers un capteur qui mesure la déviation du faisceau. En utilisant. L'appareil photo intégré au scanner, la position exacte du point de déviation du laser est enregistrée, permettant ainsi de reconstituer la forme en trois dimensions de l'objet.

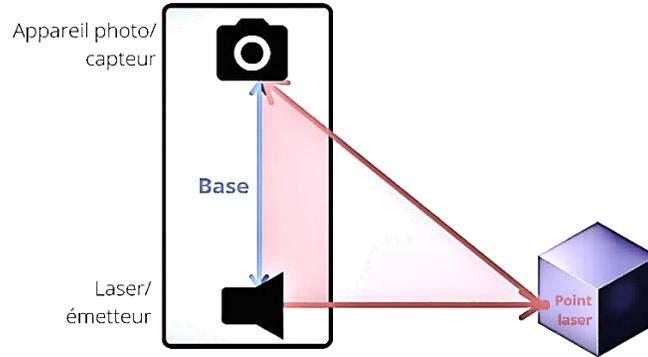


Figure 1. 5. Principe du scanner par triangulation optique [8]

Le scanner va calculer la distance scanner-objet grâce à la triangulation trigonométrique [7]:

- La distance entre l'appareil photo/capteur et le laser est connue
- L'angle entre le laser et l'appareil photo/capteur est également connu,
- Le scanner va déterminer l'angle auquel le faisceau laser réfléchi revient vers l'appareil photo/capteur

1.5.2 Le scan à la lumière structurée

Le scanner à lumière structurée est une alternative au balayage laser 3D, il utilise un projecteur numérique pour projeter un motif lumineux sur l'objet à numériser, tandis qu'une caméra enregistre les déviations de ce motif. Cela permet de calculer les distances et de reconstruire la géométrie en 3D [9]. Contrairement au balayage laser, cette technique permet de numériser une grande surface en une seule fois. Le scanner à lumière structurée offre une précision élevée et est utilisé dans diverses applications, comme l'ingénierie inverse et le contrôle qualité.

1.5.3 La photogrammétrie

La photogrammétrie est une technique qui consiste à créer un modèle 3D à partir de photos prises sous différents angles de vue [10]. Un logiciel analyse les similitudes entre les différentes images et crée un modèle 3D.

1.6 Type de données

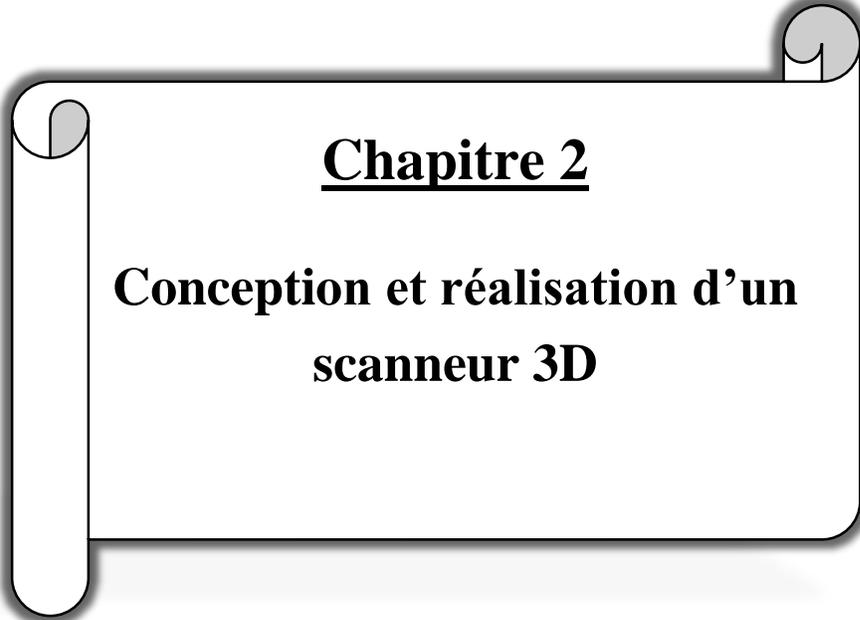
On a vu dans les paragraphes précédents que les scanners 3D utilisent différents types de capteurs, tels que des capteurs de distance, des lasers et des scanners à lumière structurée, pour collecter des données sur les objets en trois dimensions. Les données

collectées dépendent alors du type de capteur utilisé et de la technique de numérisation employée. Les principales données fournies par les scanners 3D sont les suivantes :

- **Nuage de points** : Les scanners 3D génèrent des nuages de points, qui sont des ensembles de coordonnées spatiales représentant la géométrie de la surface de l'objet. Ces nuages de points peuvent être utilisés pour créer des modèles 3D détaillés de l'objet, pour être visualisés ou utilisés pour des analyses ultérieures.
- **Maillage** : Le maillage est une représentation géométrique tridimensionnelle d'une surface, créée à partir d'un nuage de points. Les scanners 3D peuvent produire des maillages de haute qualité, souvent utilisés dans la conception de produits, l'animation, le cinéma, les jeux vidéo...etc.
- **Textures** : Les scanners 3D peuvent également capturer des informations de textures et de détails de surface. Ces données peuvent être utilisées pour créer des modèles 3D photo-réalistes.
- **Couleurs** : Certains scanners 3D sont équipés de caméras couleur qui permettent de capturer la couleur de la surface de l'objet. Ces données peuvent être utilisées pour créer des modèles 3D en couleur.
- **Mesures** : Certains scanners 3D sont équipés de capteurs de mesure précis, permettant de capturer des données de mesure telles que la distance, l'angle, la courbure, etc. Ces données peuvent être utilisées pour des analyses précises des propriétés géométriques de l'objet.
- **Données de réflectivité** : Certains scanners 3D peuvent capturer des informations sur la réflexion de la lumière sur la surface de l'objet, ce qui peut aider à identifier les matériaux, les textures et les caractéristiques de surface.
- **Déformation** : Les scanners 3D peuvent être utilisés pour mesurer les déformations et les déplacements d'un objet au fil du temps. Ces données peuvent être utiles dans la surveillance de structures, les tests de fatigue des matériaux, la vérification des défauts.
- **Thermographie** : Certains scanners 3D peuvent être équipés de caméras thermiques pour capturer des données de température de la surface de l'objet. Ces données peuvent être utilisées pour détecter les points chauds et froids, les fuites de chaleur, les défauts d'isolation.
- **Réfraction** : Certains scanners 3D peuvent capturer des informations sur la réfraction de la lumière à travers la surface de l'objet, ce qui peut aider à identifier les caractéristiques optiques telles que les lentilles, les prismes, les cristaux.
- **Densité** : Certains scanners 3D peuvent mesurer la densité des objets, qui est une mesure de la masse par unité de volume. Ces données peuvent être utiles pour la caractérisation des matériaux, la détection des défauts et la mesure des volumes.

1.7 Conclusion

Dans ce chapitre, nous avons examiné les différents types de scanners 3D, les techniques de mesure des distances utilisées ainsi que les méthodes d'acquisition de données 3D optiques. Certains systèmes nécessitent d'être fixés à un dispositif de positionnement pour balayer toute la surface de l'objet, tandis que d'autres peuvent être couplés à un système de positionnement pour faciliter et accélérer le processus d'acquisition. Chaque système d'acquisition présente ses avantages et ses inconvénients, qui dépendent du type de pièce à numériser et de l'application visée. Dans notre projet de fin de cycle, nous avons conçus et réaliser un scanner d'objet en 3D. Les étapes de conception et de réalisation seront détaillées dans le prochain chapitre.



Chapitre 2

Conception et réalisation d'un scanneur 3D

2.1 Introduction

Ce chapitre va être consacré à la présentation des différentes étapes de conception et de réalisation de notre scanner d'objet en 3D qui constitue la principale contribution de notre projet de fin d'étude. Dans ce qui suit, nous allons présenter une description du matériel utilisés, de la phase de conception ainsi que la phase de réalisations. Les différents tests et résultats obtenus seront aussi présenter.

2.2 Matérielle utilisée

2.2.1 Carte Arduino Uno

Arduino Uno est une carte électronique open-source basée sur un microcontrôleur ATmega328P. Elle possède 14 broches d'entrées/sorties numériques, 6 entrées analogiques, un oscillateur à quartz de 16 MHz, une connexion USB, un connecteur d'alimentation, une broche Reset et une interface ICSP pour la programmation (Figure 2.1). Elle est largement utilisée pour les projets électroniques de bricolage, l'automatisation, les objets connectés, etc. grâce à sa simplicité d'utilisation et son coût abordable. [11]

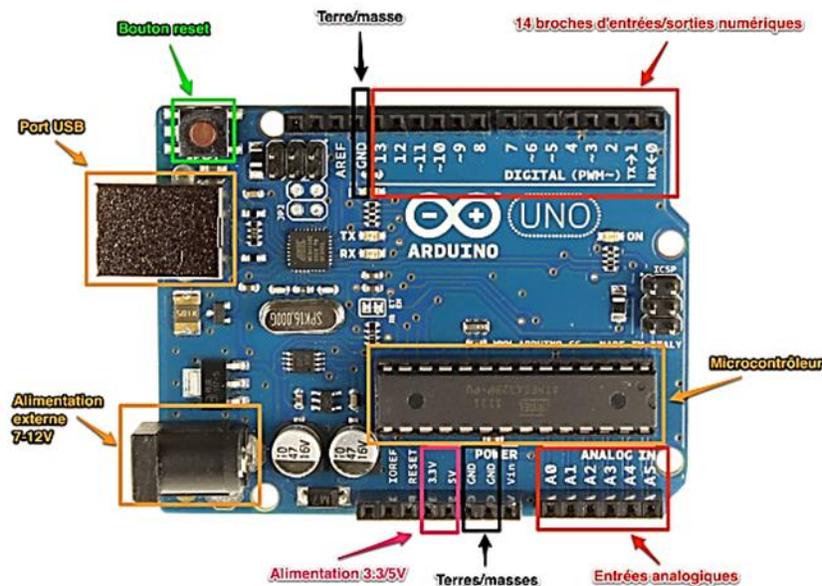


Figure 2.1. Tous les composant d'Arduino Uno

2.2.1.1 Les capacités d'Arduino

catégorie	valeur
microcontrôleur	ATmega 328
Fréquence d'horloge	16 MHz
Tension de service	5 V
Tension d'entrée (recommandé)	7-12 V
Tension d'entrée (limites)	6– 20 V
Ports numérique	14 entrées et sorties
Ports analogique	6 entrées analogique
Courant maxi par broche d'E/S	40 mA
Courant maxi par broche 3,3 V	50 mA
Mémoire	32 Ko Flash, 2 Ko SRAM, 1 Ko EEPROM
Chargeur d'amorçage	0,5 Ko (en mémoire Flash)
interface	USB
Prix (DA)	2800

Tableau 2. 1. Les spécificques de la carte Arduino Uno

2.2.2 Le module de pilote A4988

Le module A4988 (Figure 2. 2) est un dispositif électronique qui permet de contrôler et de piloter les moteurs pas à pas. Il est utilisé dans de nombreux projets électroniques et robotiques pour contrôler précisément les mouvements des moteurs pas à pas. [12]

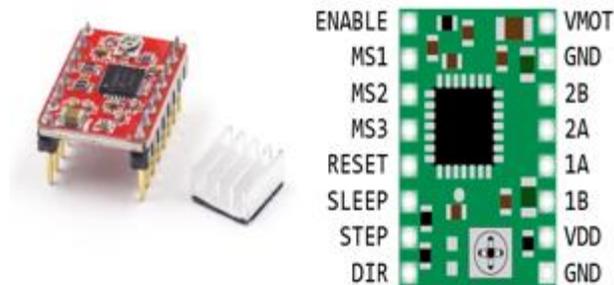


Figure 2. 2. Le module A4988

2.2.2.1 Les caractéristiques principales de ce module

- Tension d'alimentation: 8 V à 35 V
- Courant de sortie réglable jusqu'à 2 A par phase
- Micro-pas : jusqu'à 1/16 étapes
- Compatible avec les moteurs pas à pas bipolaires à 4 fils et à 6 fils
- Contrôle du sens de rotation et de la vitesse du moteur grâce à des signaux logiques

- Protection contre les surchauffes et les court-circuit
- Dimensions : 20 mm x 15 mm

Le câblage du moteur pas à pas avec le module A4988, doit prendre être réalisé selon la figure ci-dessous.

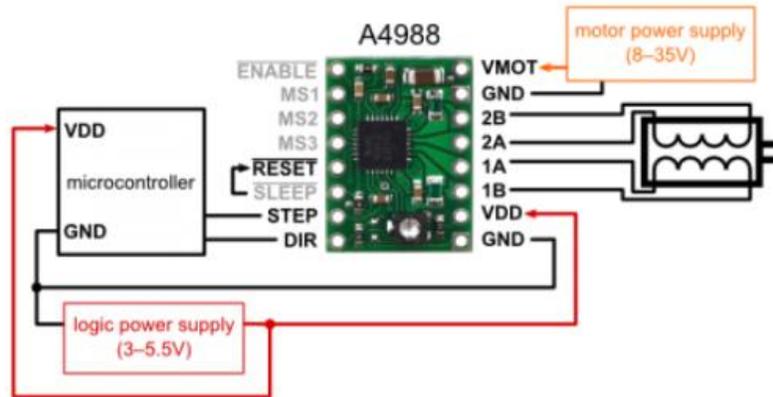


Figure 2. 3 . schéma de principe de montage A4988 [12]

2.2.3 Le moteur NEMA 17

Le moteur pas à pas NEMA17 (Figure 2.4) est un type de moteur électrique utilisé dans de nombreux projets électroniques et mécatroniques. Il tire son nom de la norme de fixation "NEMA" et de sa taille de boîtier de 1,7 pouce de côté (17 en référence au nombre 17 dans la désignation). Ce moteur fonctionne en effectuant des mouvements discrets ou "pas", d'où son nom "pas à pas". Le moteur pas à pas NEMA17 est un moteur bipolaire, ce qui signifie qu'il possède deux bobines qui permettent de contrôler la direction et la rotation du moteur. Il beaucoup plus utilisé dans des applications nécessitant un positionnement précis, telles que les imprimantes 3D, les machines CNC (Commande Numérique par Ordinateur), les robots et d'autres projets où une précision de mouvement est essentielle. [13]



Figure 2. 4. Moteur NEMA17

➤ **Les principales caractéristiques de ce moteur sont :**

➤ Taille de boîtier	42 mm
➤ Nombre de fils	4 ou 6
➤ Tension nominale	12 V ou 24 V
➤ Courant nominal	1,5 A
➤ Couple de maintien	0,4 Nm (environ)
➤ Pas d'angle	1,8 degrés par pas
➤ Inductance	Typiquement entre 2,3 et 3,6 MH
➤ Résistance	Typiquement entre 1,0 et 1,5 ohm

2.2.4 Le capteur de distance Sharp GP2Y0A41SK0F

Pour un scanner 3D, le système d'acquisition des mesures (données) joue un rôle essentiel dans la reconstruction de la forme 3D. Dans notre cas, nous avons choisi d'utiliser un capteur de distance à infrarouge (Figure 2. 5). Après avoir étudié la distance entre le capteur et le plateau tournant selon le modèle 3D de notre scanner, il est apparu nécessaire d'opter pour un capteur à petite portée. Nos recherches nous ont conduits à sélectionner le capteur de distance Sharp GP2Y0A41SK0F, qui offre une plage de mesure de 4 à 30 cm.



Figure 2. 5. Le capteur Sharp GP2Y0A41SK0F

➤ **Les principales caractéristiques de ce capteur sont :**

➤ Type de capteur	distance infrarouge
➤ Distance de mesure	4 cm à 30 cm
➤ Tension d'alimentation	4,5 V à 5,5 V
➤ Consommation de courant	33 mA typique, 50 mA max
➤ Sortie	analogique
➤ Précision	±2,5 cm à 15 cm, ±5 cm à 30 cm
➤ Angle de détection	58 degrés
➤ Temps de réponse	39 ms typique, 50 ms max
➤ Dimensions	29,5 mm x 13 mm x 13,5 mm

Ce capteur utilise un émetteur infrarouge pour émettre des faisceaux lumineux vers la cible, la lumière réfléchie est alors détectée et convertie en tension de sortie analogique proportionnelle à la distance entre le capteur et la cible selon la figure suivante [14] :

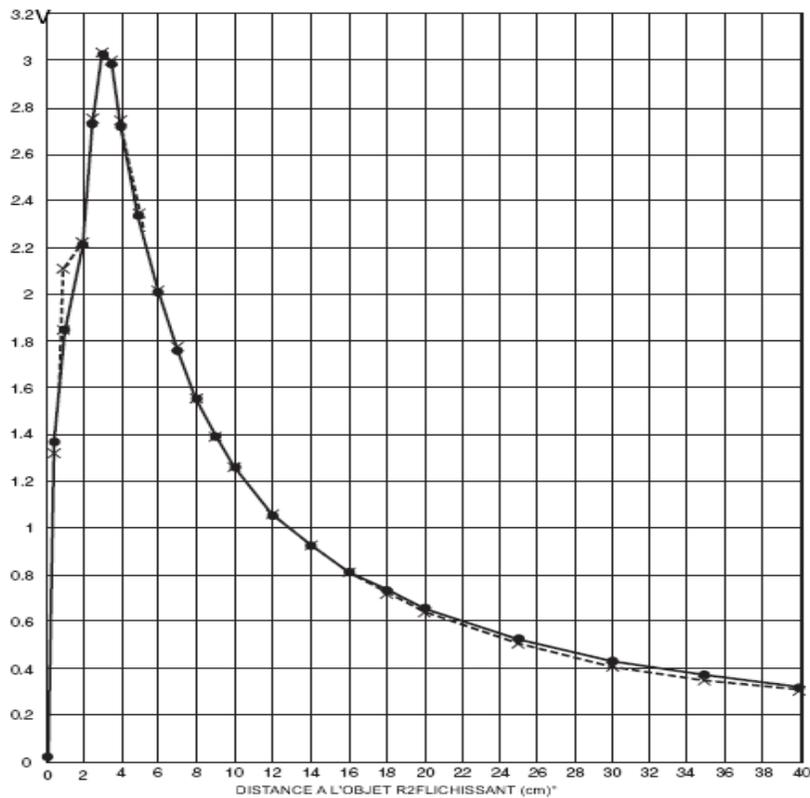


Figure 2.6. Graphe analogie distance/voltage du capteur Sharp GP2Y0A41SK0F

2.2.5 Le micro SD shield

Il s'agit d'un dispositif électronique qui permet à un appareil, tel qu'un Arduino, de lire et d'écrire des données sur une carte mémoire de type Micro SD. Ce module est utilisé pour étendre la capacité de stockage de l'appareil, en lui permettant de stocker et d'accéder à des fichiers tels que des images, des vidéos ou des données stockées sur la carte.

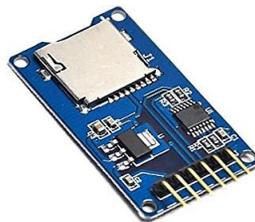


Figure 2.7. La carte de micro SD shield

Le module peut être alimenté avec une tension de 4,5V à 5,5V DC, il est compatible avec les cartes mémoire Micro SD qui utilisent une interface SPI pour communiquer avec le microcontrôleur par simple câblage (Figure 2. 8). Le module dispose aussi d'un circuit de protection contre les surtensions et les surintensités pour éviter tout dommage aux cartes mémoire.

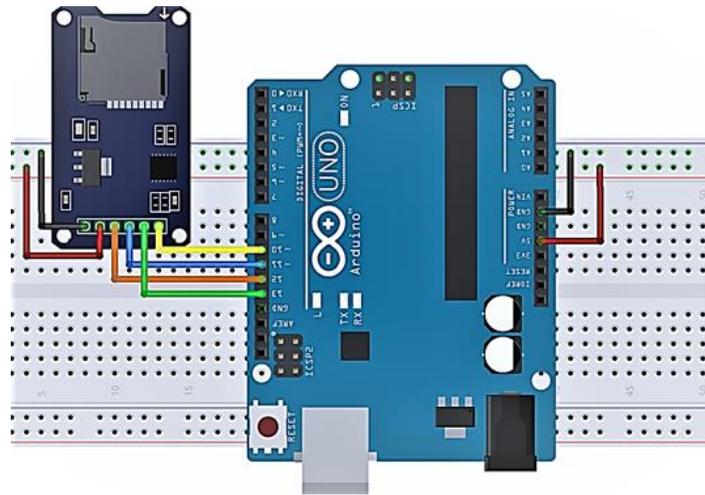


Figure 2. 8. Câblage plaque d'essai/Arduino/ micro SD shield

2.2.6 Module Bluetooth HC-06

On a utilisé le module HC-06 (figure 2.9) pour contrôler le scanner 3D à partir d'une application téléphonique, par une connexion Bluetooth. Cela nous permet de d'envoyer des commandes et des instructions au scanner 3D à partir d'un smartphone, cela nous offre ainsi une solution pratique et flexible pour le contrôle du scanner 3D à distance. [15]



Figure 2. 9 . Le module HC-06

2.2.7 Interrupteur (switch) et bouton poussoir

Dans notre projet, nous avons un moteur qui effectue la translation du fixateur du capteur vers une position précise sur l'objet qu'on souhaite numériser. Un capteur fin de course (interrupteur) est utilisé pour savoir si le fixateur a atteint la position de démarrage souhaitée (Figure 2.10). Ensuite, il faut appuyez sur le bouton poussoir (Figure 2.9) pour commencer le scan de l'objet après un délai de 3 secondes.



Figure 2. 11. Capteur fin de course



Figure 2. 10. Bouton poussoir

2.2.8 Schéma du câblage final du projet

Le schéma du câblage final de notre projet de conception d'un capteur d'objet en 3D est effectué par le logiciel Fritzing selon la figure ci-après

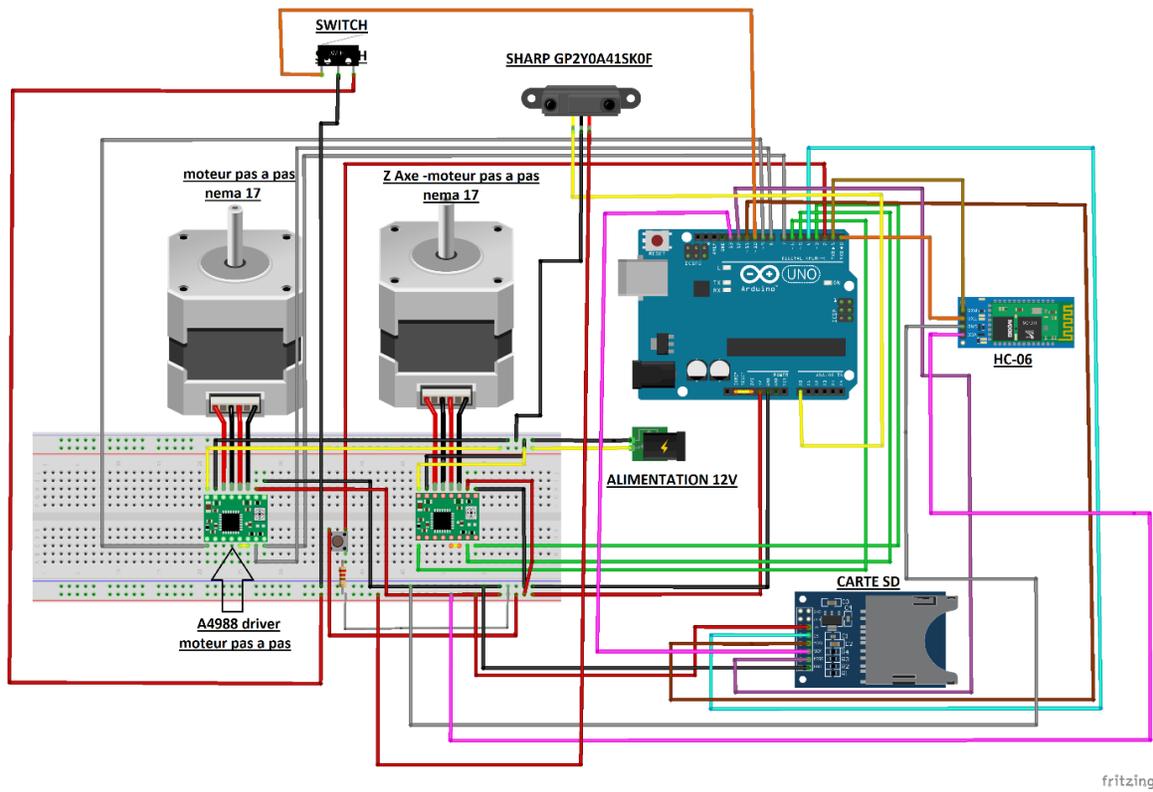


Figure 2. 12. Représente le câblage final de matérielle électrique

2.2.9 Circuit imprimé

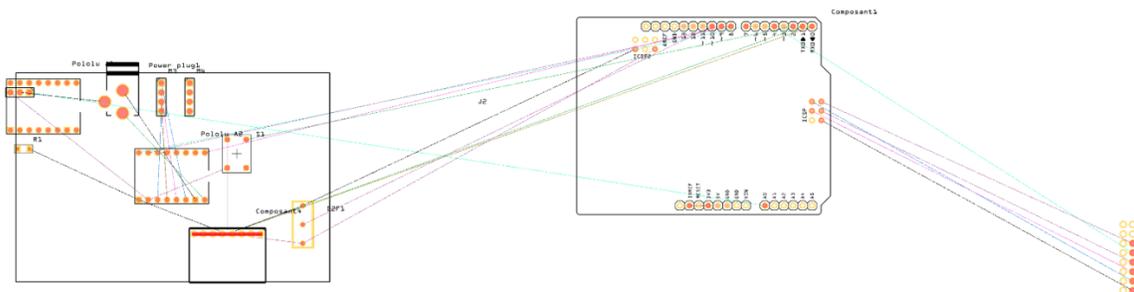


Figure 2. 13. Le circuit électronique du projet

2.2.10 Partie mécanique

2.2.10.1 Tige filetée ACME axe Z avec écrou antijeu

La tige filetée est un type de filetage couramment utilisé dans les systèmes de mouvement linéaire comme les imprimantes 3D et les machines CNC, elle est fabriquée avec un filetage trapézoïdal ACME qui offre une meilleure précision et résistance à l'usure avec de diamètre de 8 mm et 1.25 mm de pas. Lorsqu'elle est utilisée sur l'axe Z d'un système de coordonnées cartésiennes, elle permet de contrôler précisément le mouvement vertical de la plate-forme de travail par le moteur [16]. La figure 2.13 montre la tige que nous avons utilisée avec en conjonction une pièce antijeu pour réduire ou éliminer le jeu ou le jeu mécanique.



Figure 2. 14. Photo réel de la tige 8mm avec l'écrou anti jeu

2.2.10.2 Axe linéaire de 8 mm

L'arbre linéaire (figure 2.14) est généralement fabriqué en acier de haute qualité et possède une surface très lisse et usinée avec précision, qui permet à des roulements linéaires ou des paliers lisses de glisser facilement sur l'arbre. La longueur de 200 mm fait référence à la distance entre les extrémités de l'arbre.



Figure 2. 15. Axe linéaire de 8mm et 200mm

2.2.10.3 Les roulements de 8 mm

Les roulements à billes linéaires (Figure 2.15) sont utilisés dans une variété d'applications industrielles et de machines-outils, il s'agit d'éléments mécaniques utilisés pour guider et supporter des charges linéaires, notamment dans les systèmes de positionnement linéaire, les machines d'emballage, les machines CNC, les imprimantes 3D et les robots industriels. Ils peuvent supporter des charges importantes et garantissent une précision de mouvement élevée [17].



Figure 2. 16. Roulement de 8mm

2.3 Logiciel utilisée

2.3.1 Arduino IDE

L'IDE Arduino est un logiciel utilisé pour programmer et charger du code sur les cartes Arduino. Il offre une interface conviviale pour écrire, compiler et télécharger des programmes sur les microcontrôleurs Arduino. Il utilise le langage de programmation Arduino basé sur C/C++. L'IDE propose des bibliothèques et des exemples de code préprogrammés pour faciliter le développement de projets. Il prend également en charge la vérification syntaxique, la compilation du code source et le téléchargement du programme sur la carte Arduino connectée [18]. Arduino IDE est disponible gratuitement et est compatible avec différents systèmes d'exploitation tels que Windows, MacOS et Linux.

2.3.2 Fritzing

Fritzing est un logiciel de conception électronique utilisé par les professionnels pour créer des schémas électriques, des circuits imprimés et des prototypes virtuels. Il offre une interface conviviale et intuitive, une vaste bibliothèque de composants électroniques et des fonctionnalités avancées pour la création de PCB. Les professionnels de l'électronique l'utilisent pour développer des projets, effectuer des simulations, créer des prototypes et documenter leur travail. Fritzing facilite la collaboration, la

communication technique et répond aux besoins professionnels en conception électronique [19].

2.3.3 MeshLab

MeshLab est un logiciel open-source prisé par les professionnels pour la manipulation avancée de modèles 3D et de maillages. Il offre une interface conviviale et une gamme complète d'outils pour visualiser, éditer et réparer les maillages. MeshLab est régulièrement mis à jour par une communauté de développeurs actifs, garantissant ainsi l'ajout de nouvelles fonctionnalités et la résolution des problèmes. Sa compatibilité avec de nombreux formats de fichiers courants facilite l'échange de modèles avec d'autres logiciels. En somme, MeshLab est apprécié pour sa flexibilité, sa facilité d'utilisation et sa compatibilité avec les standards de l'industrie de la modélisation 3D [20]

2.3.4 SolidWorks

SolidWorks est un logiciel de conception assistée par ordinateur (CAO) 3D largement utilisé dans l'industrie. Il permet aux concepteurs et ingénieurs de créer des modèles 3D précis de pièces, d'assemblages et de systèmes complets. Avec une interface conviviale, SolidWorks offre une large gamme d'outils de modélisation et de simulation, permettant aux utilisateurs d'évaluer et d'optimiser leurs conceptions avant la production [21]. Le logiciel permet également de générer des plans de fabrication détaillés et des fichiers compatibles avec les machines-outils à commande numérique (CNC). SolidWorks est apprécié pour sa facilité d'utilisation, sa stabilité et son intégration avec d'autres logiciels et systèmes de gestion de données. Il est largement utilisé dans divers secteurs industriels, tels que l'ingénierie mécanique, l'aérospatiale et l'automobile. [22]

2.4 Schéma fonctionnel du scanner 3D

Le programme de commande envoie d'abord aux contrôleurs des moteurs les informations sur la direction et le nombre de pas à effectuer. Ensuite, il fournit une tension modulée et un courant régulé aux moteurs pour un contrôle précis du mouvement. En ce qui concerne le capteur, le programme de commande envoie des commandes pour activer et contrôler celui-ci. Une fois les données acquises, le programme les traite ou les manipule. Ensuite, les enregistrements obtenus sont stockés dans une carte mémoire SD pour une conservation ultérieure. Le schéma illustre le fonctionnement du scanner. Le moteur responsable de la rotation du plateau tournant et appelé "Moteur 2", tandis que le moteur qui assure la translation du capteur est appelé "Moteur 1".

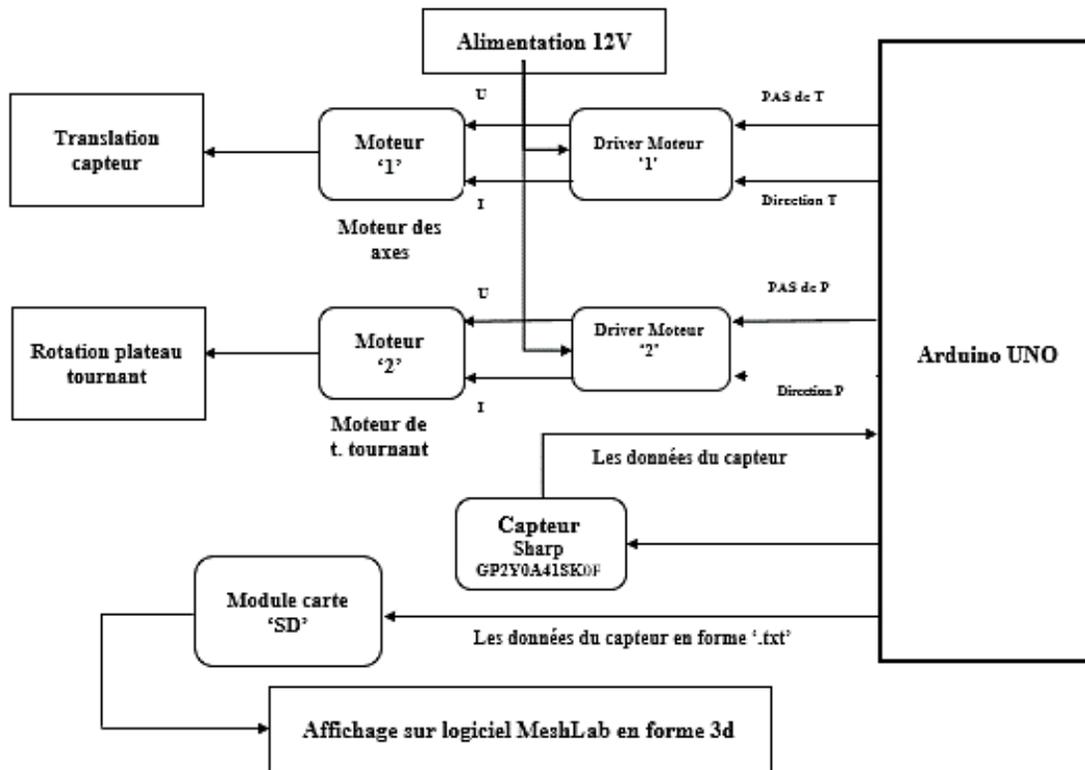


Figure 2. 17. Schéma fonctionnel du scanner 3d

2.5 Conception du scanner 3D

Le scanner à concevoir est composé de cinq éléments principaux : les deux moteurs pas à pas, le support des moteurs, le plateau tournant et le support de fixation du capteur et le coupleur d'arbre. Ces éléments sont essentiels pour assurer le bon fonctionnement du scanner et permettre la capture précise des données. Les autres éléments inclus dans le scanner sont des outils d'assemblage qui facilitent l'assemblage et la stabilité de l'ensemble. Afin de créer une représentation visuelle réaliste de ces pièces principales, nous avons utilisé le logiciel SolidWorks pour réaliser une modélisation 3D. Dans le processus de modélisation, nous avons pris en compte les dimensions réelles des pièces pour obtenir une représentation précise et fidèle du scanner. Les modèles SolidWorks des différentes pièces du projet sont présentés par les figures suivantes :

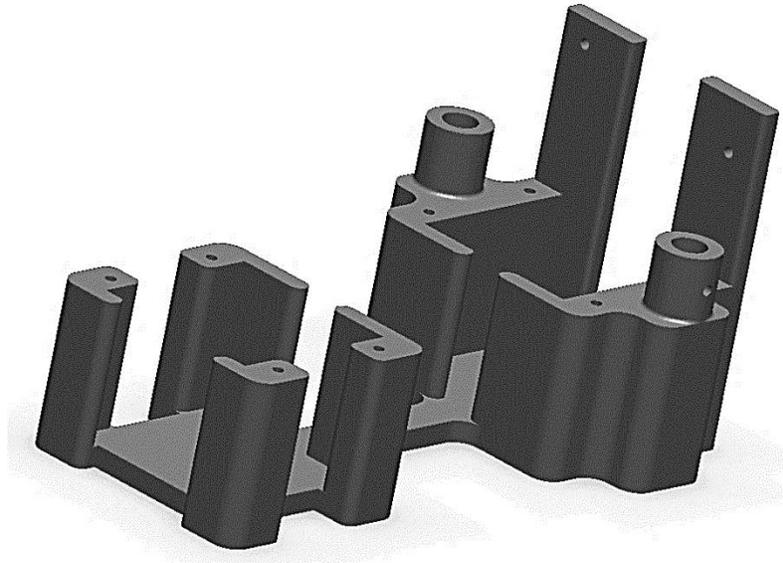


Figure 2. 18. La création du modèle 3D du support des moteurs

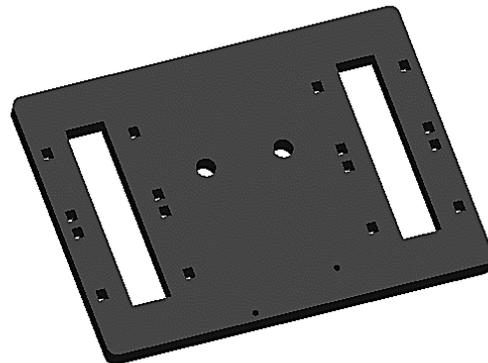


Figure 2. 19. Le modèle 3D de fixateur de capteur

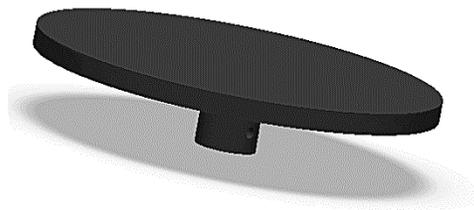


Figure 2. 21. Le modèle 3D de plateau tournant

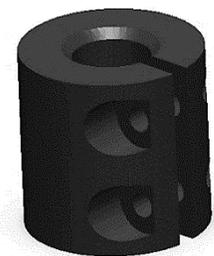


Figure 2. 20. Modèle 3D de coupleur d'arbre



Figure 2. 22. Modèle 3D de la plaque de fixation

Nous avons utilisé le coupleur d'arbre pour connecter l'arbre du moteur à la tige filetée. Le coupleur d'arbre permettra de transférer le mouvement de rotation du moteur à la tige filetée de manière précise et sans jeu excessif, ce qui permettra une transmission efficace, en minimisant les désalignements et les vibrations indésirables. Nous avons aussi utilisé une plaque de fixation sur les axes pour stabiliser les deux axes et éliminée le jeu et les vibrations entre le fixateur de capteur et les axes.

2.5.1 L'assemblage toutes les pièces

Nous avons assemblé les pièces conçues dans le logiciel SolidWorks, comme le montre la figure suivante :

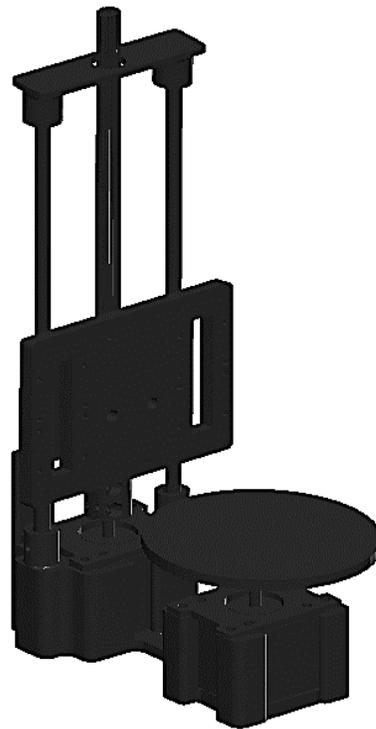


Figure 2. 23 . La conception finale du scanner 3d

La photo réelle du projet complet est montrée pour la figure suivante :



Figure 2. 24 . Photo réelle du scanner d'objet 3D réalisé

2.6 Test et résultat

- Objet de test réel : une forme cylindrique



Figure 2. 25 . Objet de test réel

Les données obtenues après la fin du scan sont représentées sur logiciel MeshLab comme suit :

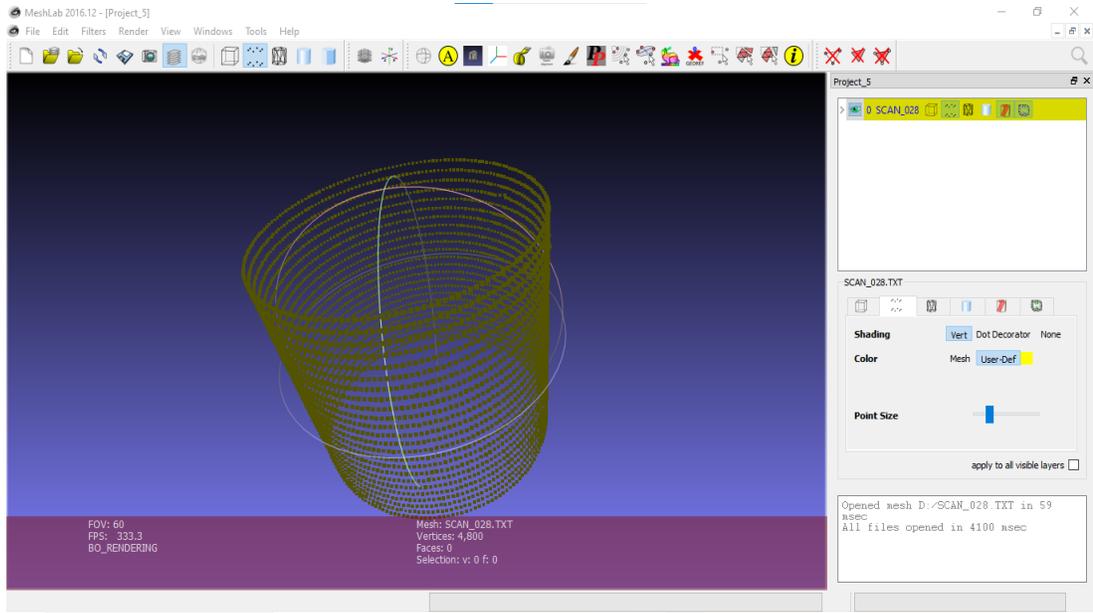


Figure 2. 26 . Données obtenues après le scan représentées par le logiciel MeshLab

La figure suivante présente le résultat du scan après le traitement des données par le logiciel MeshLab.

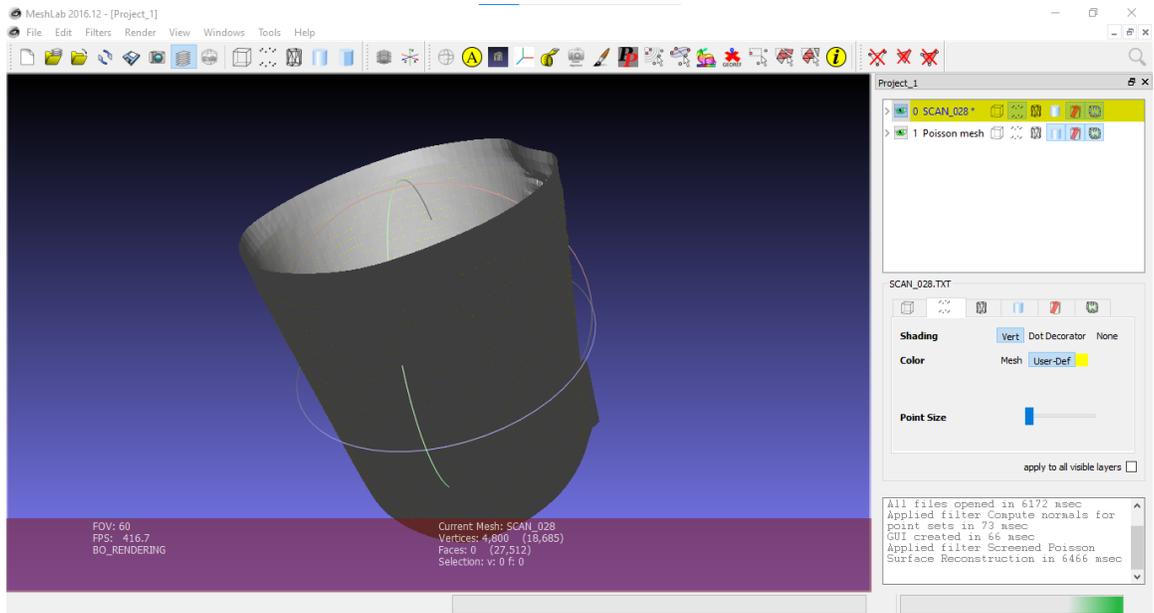
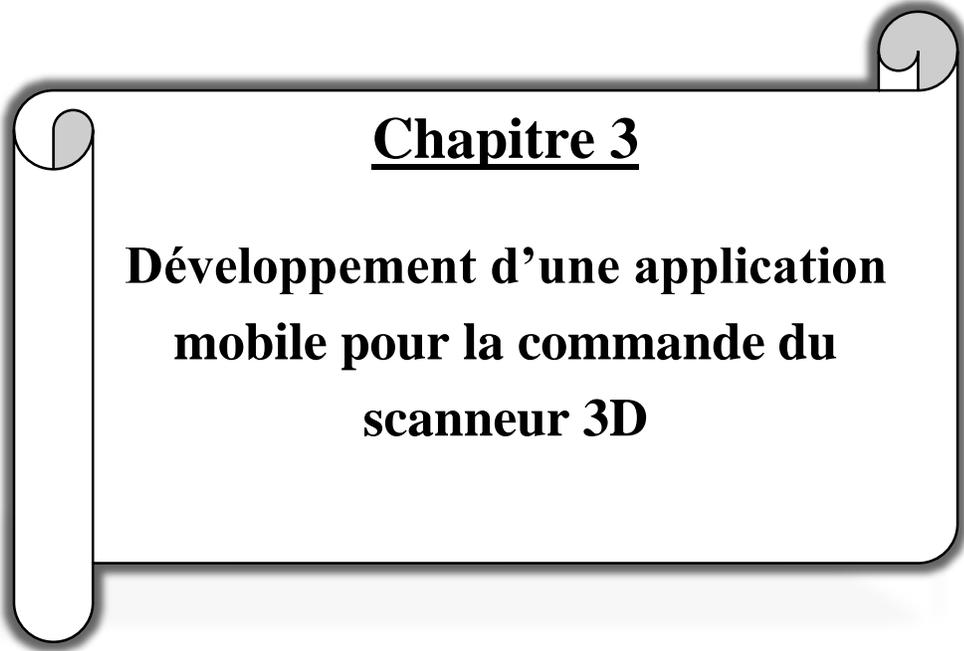


Figure 2. 27 . Résultat obtenu après traitement des données par MeshLab

On constate qu'on a obtenu la forme originale de l'objet

2.7 Conclusion

Dans ce chapitre, nous avons présenté la réalisation du scanner 3D en mettant en place le matériel électrique et mécanique nécessaire à son fonctionnement. Chaque pièce a été soigneusement assemblée pour former un scanner opérationnel. Nous avons également abordé les logiciels utilisés pour la modélisation des pièces et la programmation du scanner. Un élément clé de la réalisation du scanner a été l'utilisation de l'impression 3D. Cette technologie nous a permis de créer des pièces sur mesure avec une grande précision et une résistance adaptée, contribuant ainsi à la qualité et à la fonctionnalité du scanner. Grâce à l'impression 3D, nous avons pu matérialiser nos conceptions et progresser dans la réalisation du scanner. Une fois le scanner assemblé, il est prêt à être utilisé pour l'acquisition de formes en 3D. Grâce à sa conception et à sa construction minutieuse, nous pouvons nous attendre à ce qu'il soit capable de capturer avec précision les objets en 3D. En conclusion, en combinant le matériel électrique et mécanique, les logiciels de modélisation et l'impression 3D, nous avons réussi à créer un scanner opérationnel. Les prochaines étapes consisteront à exploiter tout le potentiel du scanner pour la numérisation en 3D et à continuer d'améliorer ses performances. Grâce à notre travail rigoureux et à notre utilisation des technologies avancées, nous sommes confiants dans la qualité et les possibilités du scanner que nous avons réalisé.



Chapitre 3

**Développement d'une application
mobile pour la commande du
scanneur 3D**

3.1 Introduction

Dans ce chapitre, nous abordons le développement d'une l'application de contrôle et de gestion du scanner 3D, le but est de pouvoir le commander à distance par Bluetooth en utilisant un smartphone, à l'aide d'une application Android. Pour cela, nous avons utilisés MIT App Inventor. Cette application permet aux utilisateurs de mettre en marche ou d'éteindre le scanner et de gérer la connexion Bluetooth avec le module HC-06. Nous décrivons également en détail les fonctionnalités et le fonctionnement de l'application que nous avons créée pour le scanner 3D.

3.2 La plateforme MIT App inventor

App Inventor est une plateforme en ligne gratuite dédiée au développement de logiciels qui facilite la création d'applications Android de manière conviviale et accessible. Il est actuellement entretenu par le Massachusetts Institute of Technology.

L'utilisation d'App Inventor a la possibilité de choisir parmi divers composants préconfigurés tels que des boutons, du texte et des images, et de les assembler pour construire leur application. Pour ajouter des fonctionnalités à ces composants, ils peuvent utiliser des blocs de programmation similaires à ceux de Scratch.

Une fois que l'application est prête, les utilisateurs ont la possibilité de la tester sur leur propre appareil Android ou de la partager avec d'autres utilisateurs. App Inventor permet également aux utilisateurs de publier leur application sur le Google Play Store, rendant ainsi leur création disponible pour tous les utilisateurs Android. [23]

Le processus de développement avec App Inventor comprend trois fenêtres distinctes :

- **Une fenêtre dédiée à la création de l'interface utilisateur (IHM) :** elle vous permet de définir l'apparence de votre application.

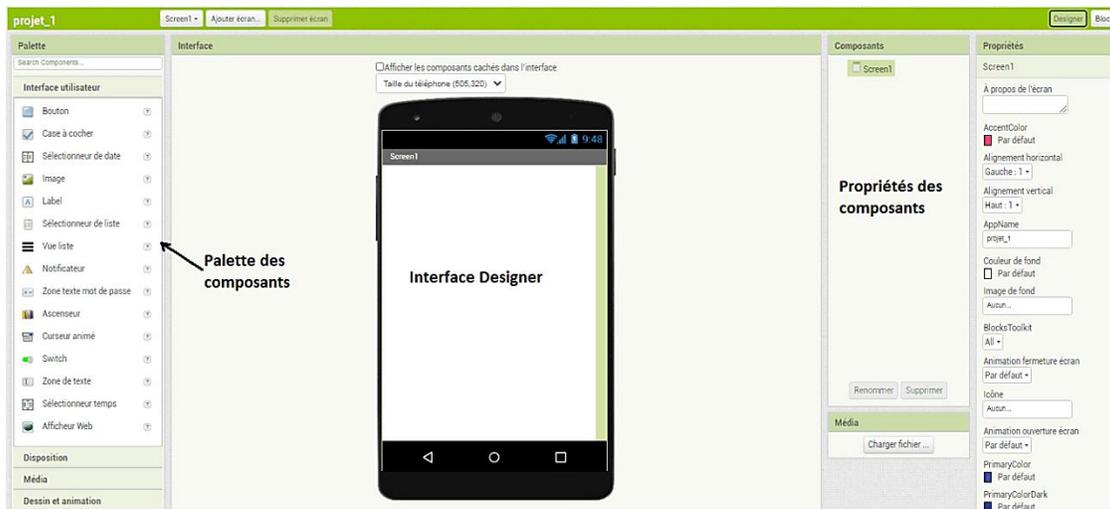


Figure 3. 1. L'espace designer

- **Une fenêtre consacrée à la programmation** : en assemblant des blocs, vous pouvez définir le comportement de votre application.

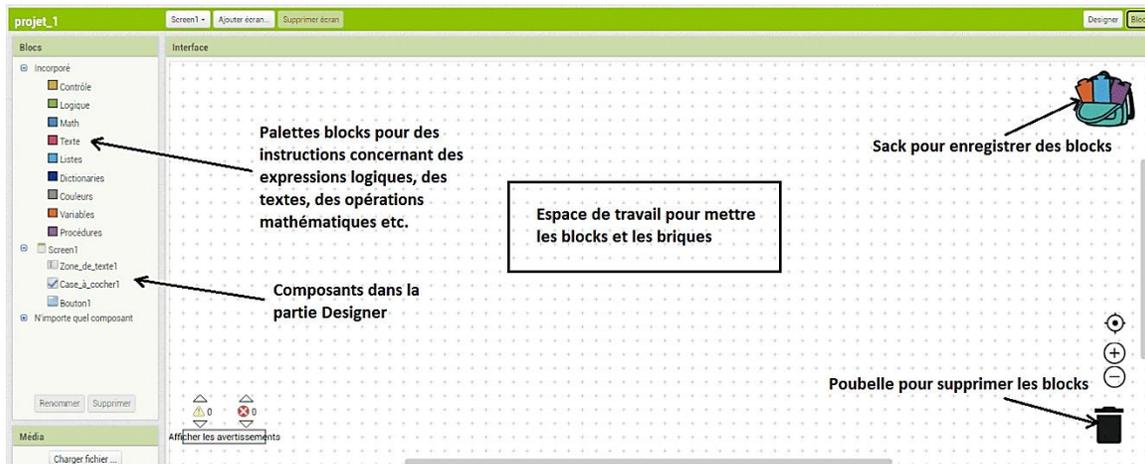


Figure 3. 2. L'espace bloc

- **Une fenêtre pour l'émulateur** : elle vous permet de tester votre application en simulant un terminal réel. L'émulateur vérifie le bon fonctionnement du programme.



Figure 3. 3. Emulateur de MIT App [24]

3.3 Utilisation dans notre projet

Dans ce qui suit, nous allons présenter les étapes de construction de l'application Android personnalisée pour contrôler notre scanner selon l'organigramme de la figure 3.1 à l'aide du MIT App Inventor via Bluetooth. On commence par :

3.3.1 L'espace designer

Nous avons d'abord ajouté quelques HorizontalArrangements de la palette de mise en page et défini leurs propriétés comme la hauteur, la largeur et l'alignement pour correspondre à l'apparence souhaitée par notre programme.

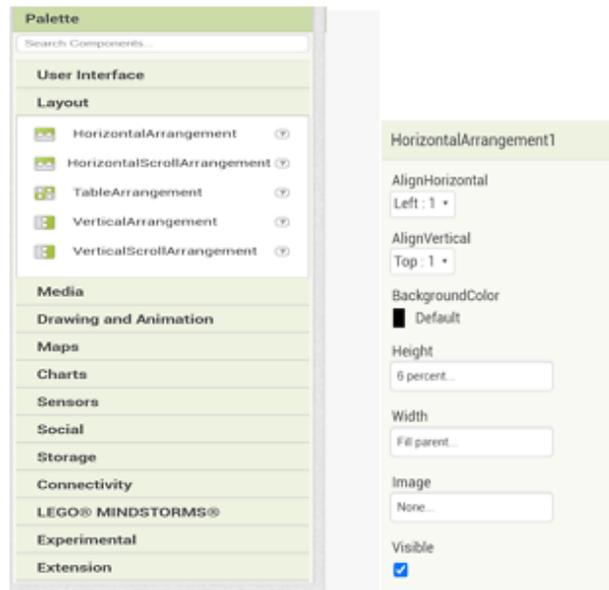


Figure 3. 4. HorizontalArrangements et leurs propriétés

Ensuite, à partir de la palette User Interface, nous avons ajouté un ListPicker et y avons attaché une image. Le ListPicker a été utilisé pour sélectionner le périphérique Bluetooth auquel notre smartphone s'est connecté.

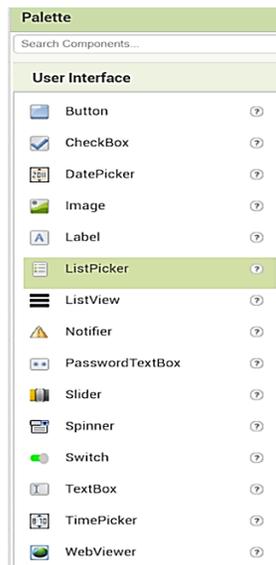


Figure 3. 5. ListPicker

Ensuite, nous avons ajouté un autre HorizontalArrangements dans lequel nous avons placé un Label. Cette étiquette indiquait si le smartphone était connecté ou non au module Bluetooth, et c'est pour cela que nous avons défini le texte initial de cette étiquette sur "Not connected".

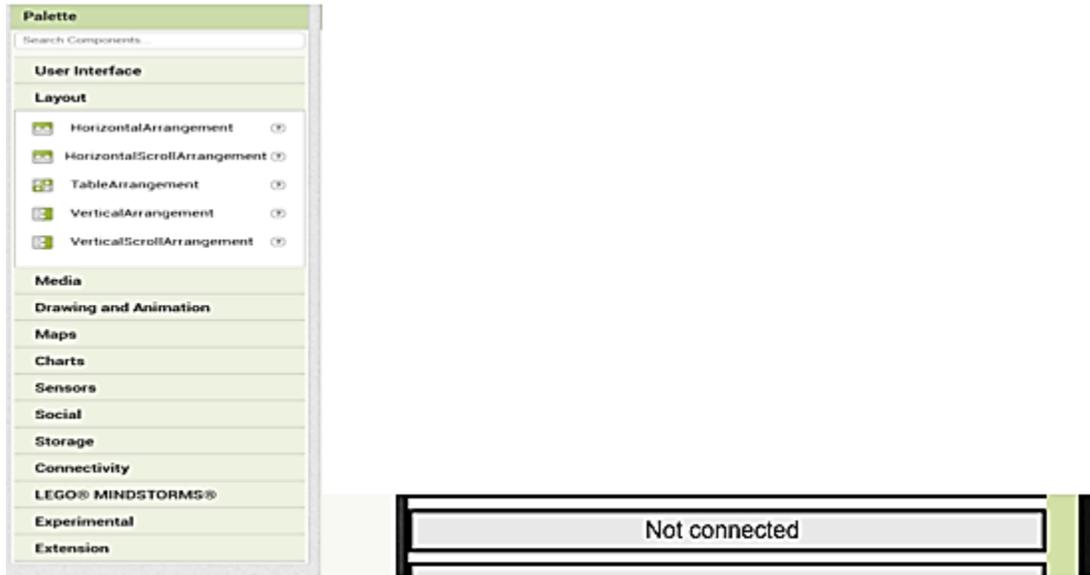


Figure 3. 6. HorizontalArrangements et l'étiquette "Not connected"

L'étiquette suivante sera utilisée pour afficher l'état des moteurs et du balayage, qu'il soit éteint ou allumé. L'état initial sera "MOTOR OFF"



Figure 3. 7. L'étiquette "MOTOR OFF"

Ensuite, nous avons ajouté les deux boutons, "TURN : ON" et "TURN : OFF", pour contrôler les moteurs et le scan.

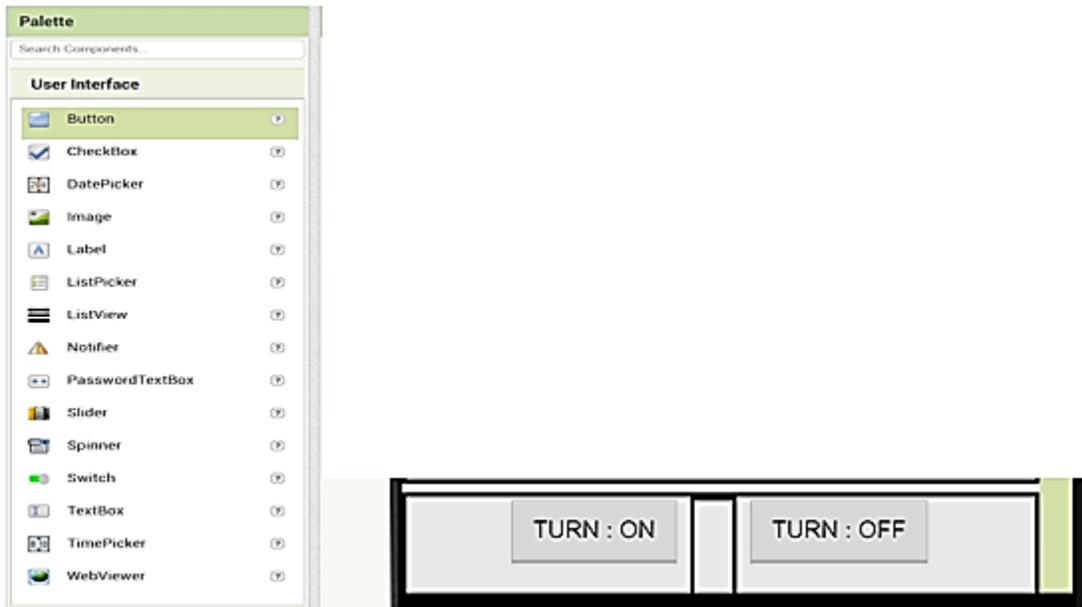


Figure 3. 8. Les boutons TURN : ON et TURN : OFF

Enfin, nous avons ajouté le Bluetooth Client qui était un composant non visible, ainsi qu'une horloge qui serait utilisée pour l'indication en temps réel de l'état de la connexion.

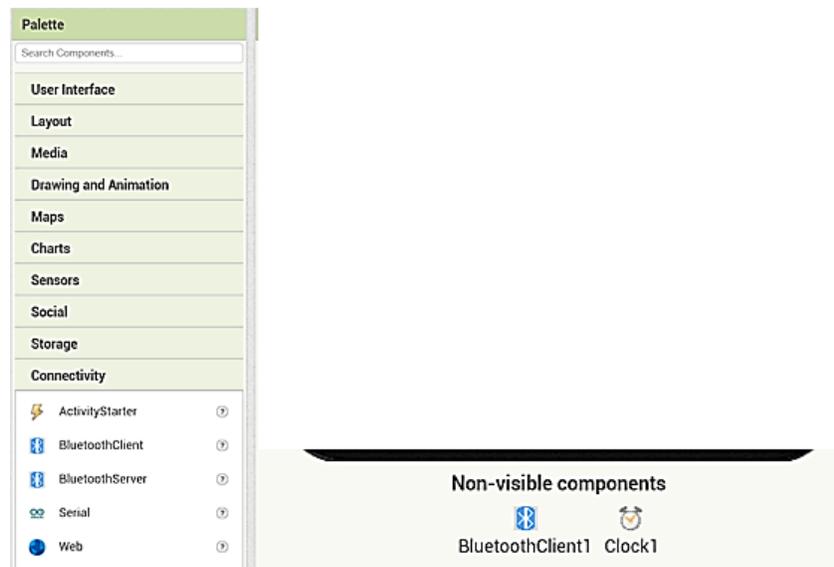


Figure 3. 9. Les composants non-visibles

3.3.2 L'espace Blocs

Nous avons commencé par ajouter le Bluetooth List ListPicker. Puis, nous avons ajouté le bloc "BeforePicking" et y avons attaché le bloc "set Bluetooth Eléments". Ensuite, à partir des blocs Bluetooth Client, nous avons ajouté le bloc "BluetoothClient1 AddressesAndNames". Ce groupe de blocs a permis de définir une liste d'appareils Bluetooth déjà associés à notre téléphone. Ainsi, lorsque nous cliquons sur le bouton "Connecter" du ListPicker, la liste de tous les appareils associés s'afficherait.



Figure 3. 10. Groupe de blocs pour les appareils Bluetooth

En plus, nous avons dû définir ce qui se passerait après avoir sélectionné notre module Bluetooth particulier. À partir du bloc BluetoothClient1, nous avons ajouté le bloc "call BluetoothClient1.Connect adresse" et y avons ajouté le bloc "BluetoothList Selection", ce qui signifiait que notre téléphone se connecterait à l'adresse Bluetooth que nous avons précédemment sélectionnée.

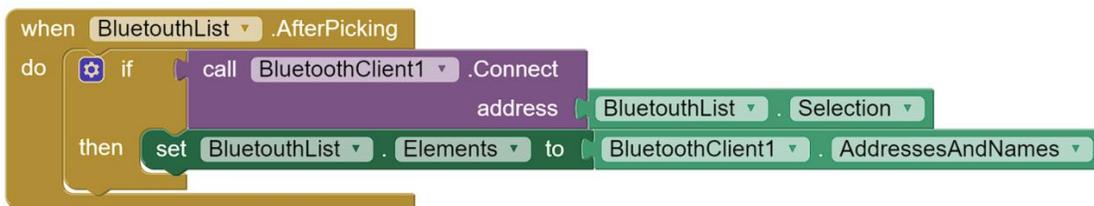


Figure 3. 11. Groupe de blocs pour sélectionner notre Bluetooth

Ensuite, à partir des blocs Clock, nous avons ajouté le bloc "Timer". À l'intérieur de ce bloc, nous avons indiqué en temps réel si le téléphone était connecté ou non au module Bluetooth en utilisant le bloc "set Text" de l'étiquette nommée "Connected".

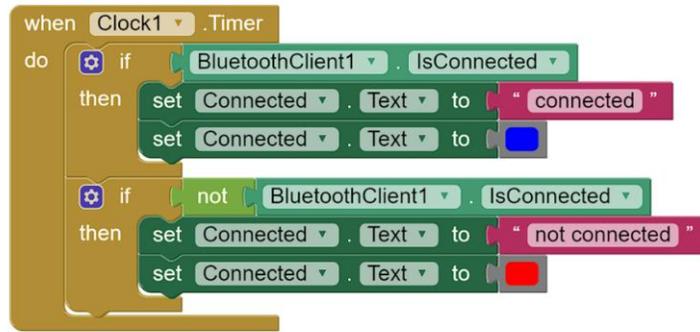


Figure 3. 12. Groupe de blocs de 'Timer'+ statut du Bluetooth

Après, nous avons donné vie aux deux boutons. Ainsi, lorsque le bouton "Turn_On_Button" était cliqué, nous utilisons la fonction "Send1ByteNumber" du client Bluetooth pour envoyer un nombre au module Bluetooth de l'Arduino. Dans notre cas, il s'agissait du nombre 49 qui correspondait au caractère '1' selon la table ASCII, et cela permettait à notre scanner de commencer. Immédiatement après, nous utilisons la fonction "ReceiveText" du Bluetooth Client pour recevoir la chaîne de caractères entrante renvoyée de l'Arduino vers le téléphone. Cette chaîne de caractères était ensuite définie dans l'étiquette "MOTORSStatus".



Figure 3. 13. Groupe de blocs de bouton TURN : ON

La même procédure s'appliquait au bouton "Turn_Off_Button", où le nombre envoyé devait être modifié en 48, ce qui correspondait au caractère '0'. Maintenant, Il ne reste plus qu'à télécharger et installer le programme sur notre smartphone : nous pouvons le faire à partir du menu "Build" en le sauvegardant sur notre ordinateur, puis en le transférant sur notre téléphone, ou en scannant un code QR pour le télécharger en ligne.



Figure 3. 14. Groupe de blocs de bouton TURN : OFF

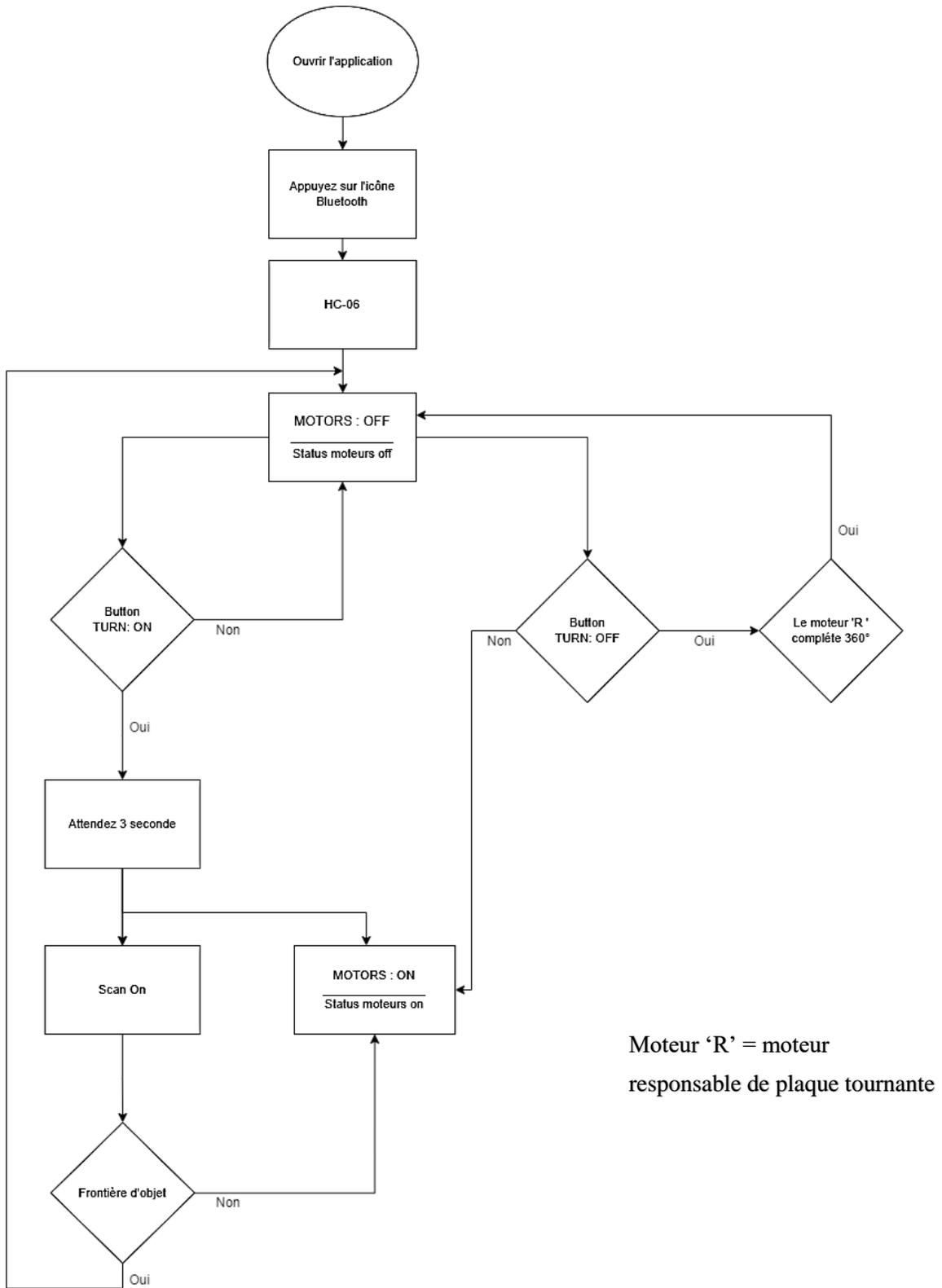


Figure 3. 15. Organigramme de l'application mobile

3.4 Test et résultat

Voici l'interface et les blocs de notre application 'Scanner 3D' :

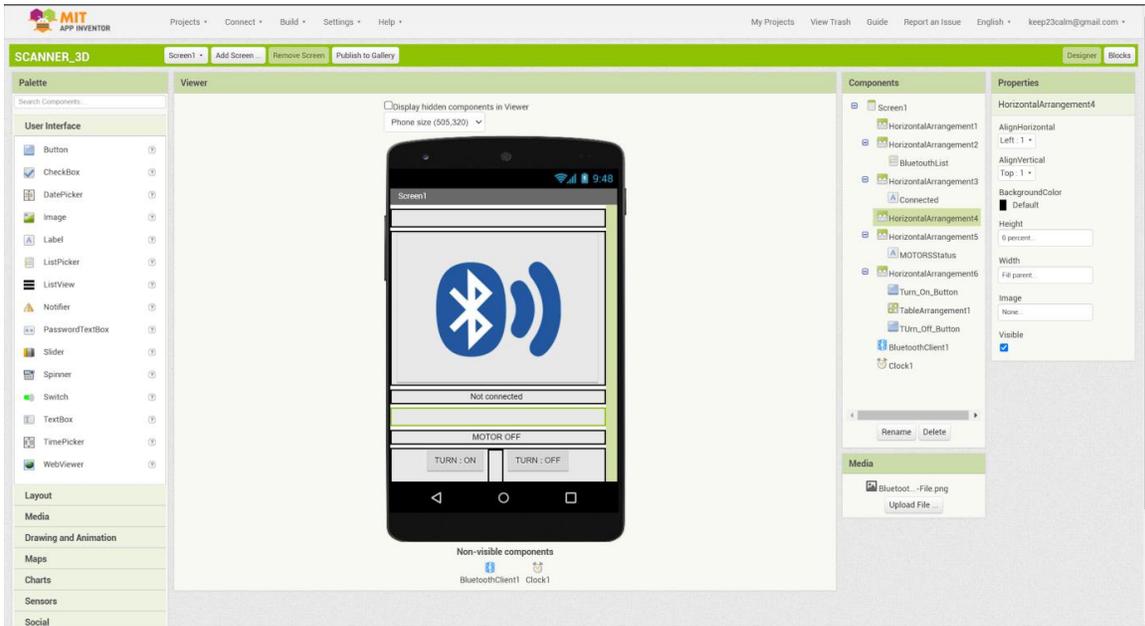


Figure 3. 16. Interface d'application scanner 3D

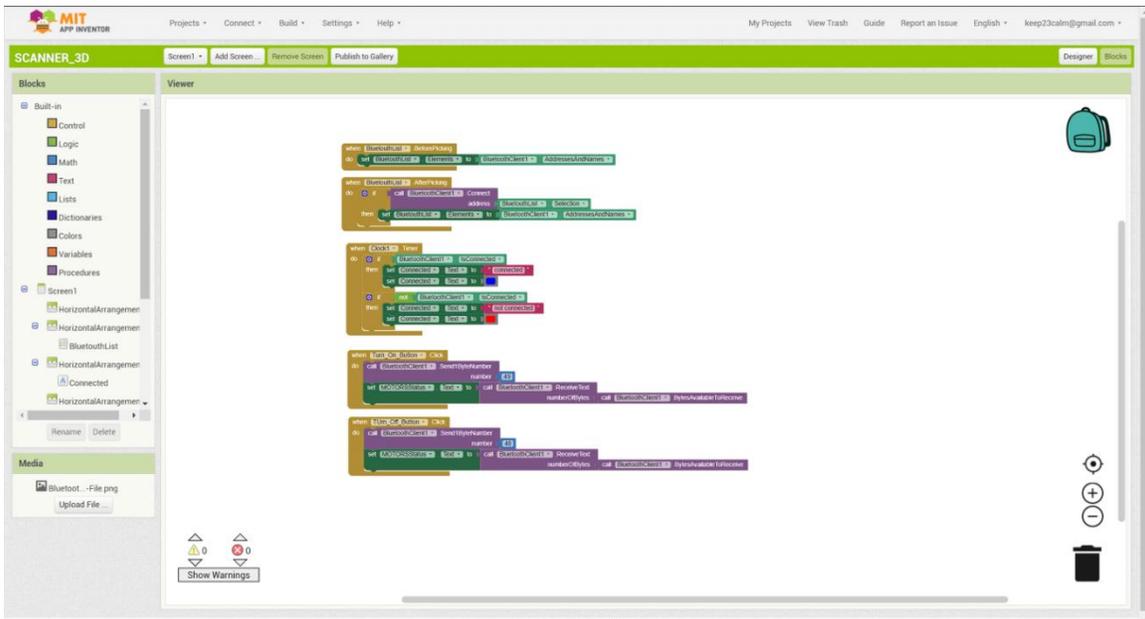


Figure 3. 17. Les blocs de l'application

Nous avons aussi conçu un code QR (Figure 3.18) pour faciliter le téléchargement de l'application, comme le montre la figure suivante.

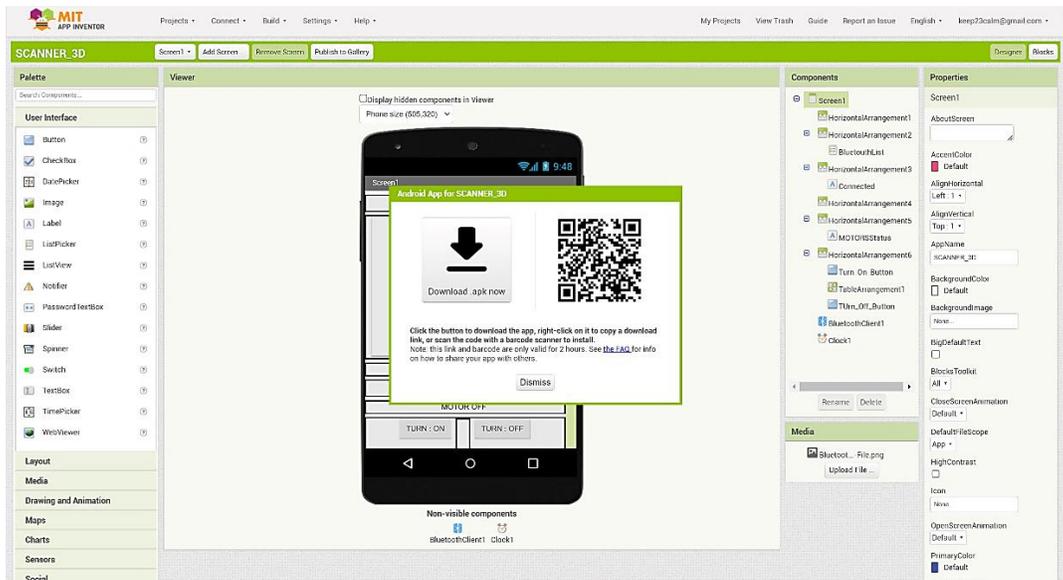


Figure 3. 18. Le code QR de l'application

Après le téléchargement de l'application, on est enfin prêt à la tester.

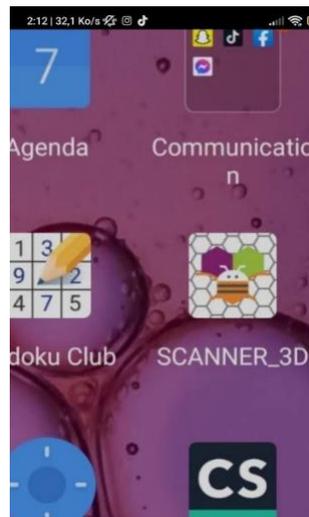


Figure 3. 19. L'icône de l'application SCANNER_3D sur le smartphone

3.4.1 Test réel de l'application

Les figures suivantes montrent le test en conditions réelles de notre application.



Figure 3. 20. Interface de l'application

- Dans l'interface de l'application, appuyer sur l'icône Bluetooth pour sélectionner le réseaux approprié (celui du HC06).



Figure 3. 21. Sélection du réseau Bluetooth

- Sélectionner marche ou arrêt des moteurs.

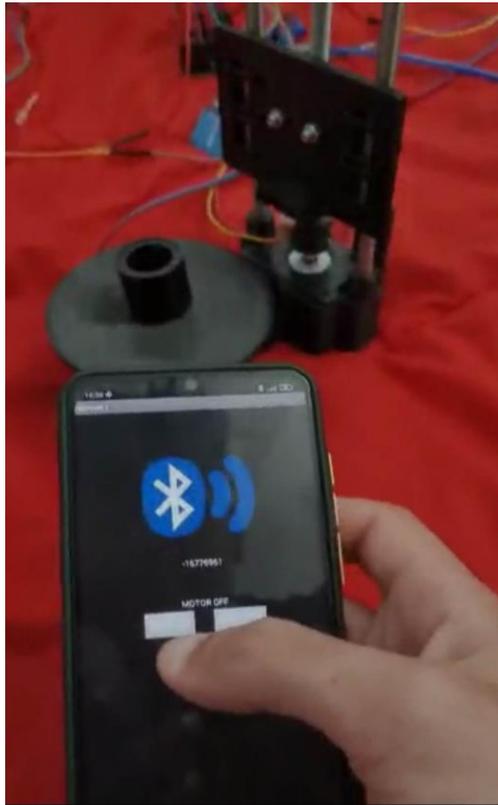


Figure 3. 22. Boutons de sélection marche/arrêt des moteurs

- Appuyer sur TURN : ON

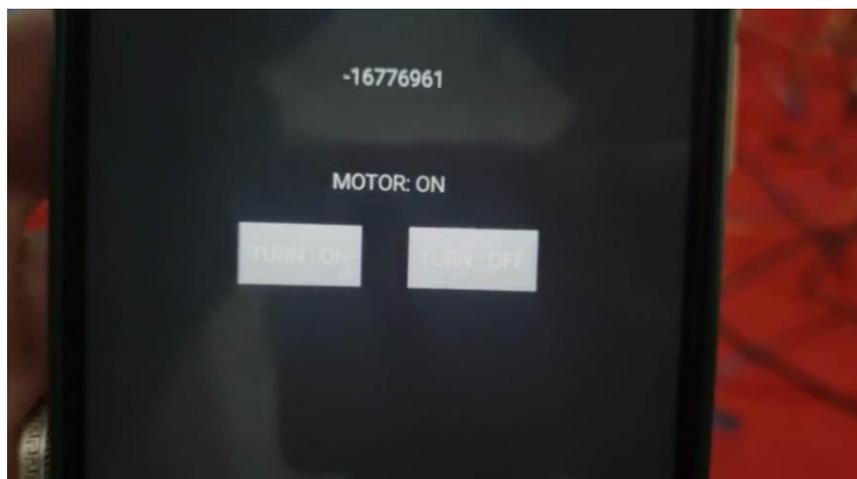


Figure 3. 23. Statuts des moteurs et du scan

- Le scan commence et les moteurs sont en marche, quand le scan se termine (frontière de l'objet à scanner atteinte), les moteurs s'arrêtent et leur statut devient off.

3.5 Conclusion

En conclusion, MIT App Inventor a été utilisé avec succès pour créer une application de contrôle du scanner 3D. Son interface conviviale et ses blocs de programmation visuelle ont facilité le développement de l'application. Grâce à cet outil, le démarrage et l'arrêt du scanner ont été réalisés simplement en appuyant sur un bouton.

Conclusion générale et perspective

La réalisation de ce mémoire de fin d'études nous a permis d'explorer en profondeur le domaine des scanners d'objets en 3D et de développer un scanner 3D fonctionnel ainsi qu'une application mobile de commande. À travers les différents chapitres, nous avons abordé les notions générales sur les scanners 3D, les types de capteurs utilisés, les technologies de numérisation, ainsi que les domaines d'utilisation.

Dans le chapitre 2, nous avons présenté en détail la conception et la réalisation du scanner 3D. Nous avons exposé les composants matériels utilisés, tels que la carte Arduino Uno, le module de pilote A4988, le moteur NEMA 17, le capteur de distance Sharp GP2Y0A41SK0F, entre autres. De plus, nous avons décrit les logiciels employés, notamment Arduino IDE, Fritzing, MeshLab et SolidWorks. Enfin, nous avons réalisé le schéma fonctionnel du scanner 3D, suivi de la conception et du montage des pièces. Les tests effectués ont permis de valider le fonctionnement du scanner 3D.

Dans le chapitre 3, nous avons développé une application mobile permettant de contrôler le scanner 3D. Nous avons utilisé la plateforme MIT App Inventor et présenté en détail les différentes étapes de conception dans l'espace Designer et l'espace Blocs. Des tests réels ont été réalisés pour évaluer l'efficacité de l'application.

Enfin, cette étude s'est conclue par une synthèse générale des résultats obtenus et des perspectives futures. La réalisation du scanner 3D et de l'application mobile constitue une avancée significative dans le domaine de la numérisation 3D. Cependant, des améliorations sont encore possibles, notamment en termes de précision et de vitesse de numérisation. Des travaux ultérieurs pourraient se concentrer sur l'optimisation de ces aspects.

En conclusion, ce mémoire de fin d'études nous a permis d'acquérir une compréhension approfondie des scanners d'objets en 3D, ainsi que des compétences pratiques en conception et réalisation d'un scanner 3D fonctionnel. Nous espérons que cette étude contribuera à l'avancement de la numérisation 3D et suscitera de nouvelles recherches dans ce domaine passionnant.

Bibliographie

- [1] S. JAILLET, B. SADIER, G. PERAZIO et J. D. Jean.france Brevet Une brève histoire de la 3D en grotte." *Karstologia* 63 : 3-20., 2014.
- [2] M. GREGOR, S. MEDVECKÝ et A. ŠTEFÁNIK , *3D laserové skenovanie*, CEIT, Žilina, 2016.
- [3] «Archéologie : Imagerie numérique et 3D,» chez *Séminaire scientifique et technique organisé par le Inrap et Inria/IRISA*, Rennes, 2018 .
- [4] M. Daneshmand, A. Helmi, E. Avots, F. Noroozi, F. Alisinanoglu, H. Arslan et G. Anbarjafari, *3d scanning: A comprehensive survey*, arXiv preprint arXiv:1801.08863., 2018.
- [5] «Ingénierie inverse | Balayage laser 3D | Scan 3D | Nuage de points,» scanphase, 22 mai 2023. [En ligne]. Available: <https://www.scanphase.com/ingenierie-inverse>.
- [6] S. e. 3. |. M. 3. |. n. i. |. B. |. E. (. d. scanphase.. [En ligne]. Available: <https://www.scanphase.com/scan-de-pieces>. [Accès le 23 mai 2023].
- [7] «Les scanners 3D | Trousses à outils,» [En ligne]. Available: <https://toolbox.hub-charleroi.be/fiche/les-scanners-3d>. [Accès le 20 mai 2023].
- [8] B. LORIOT, *Automatisation de l'acquisition et des post-traitements en numérisation 3D*, thèse de Université de Bourgogne., 2009.
- [9] L.-S. BIERI, Y. L. DE MENESES et J. et JACOT, *Laboratoire de Production Microtechnique, Institut de Production et Robotiqu*, suisse: Ecole Polytechnique F ederale de Lausanne, 2003.
- [10] «La modélisation 3D par photogrammétrie,» EXID DIAGNOSTIC, 20 avril 2021. [En ligne]. Available: <https://www.exid-diagnostic.fr/la-modelisation-3d-par-photogrammetrie/>. [Accès le 6 juin 2023].
- [11] e. bartman, *le grand livre d'arduino*, 2 éd., eyrolles, 2018, pp. 28-30.
- [12] «Comprendre et bien utiliser le driver A4988 pour moteurs pas à pas. -,» [En ligne]. Available: <https://devenez-pro-en-electronique.com/comprendre-et-bien-utiliser-le>

- driver-a4988-pour-moteurs-pas-a-pas/. [Accès le 28 5 2023].
- [13] M. Correvon, *LES MOTEURS PAS À PAS*, Haute Ecole d'Ingénierie et de Gestion Du Canton du Vaud, pp. 1-4.
- [14] «The outline and the characteristics of the analog ouput distance measuring sensor,» Sharp, [En ligne]. Available: <https://html.alldatasheet.com/html-pdf/506283/SHARP/GP2Y0A41SK0F/1443/9/GP2Y0A41SK0F.html>. [Accès le 31 05 2023].
- [15] Xukyō, «Votre Arduino communique avec le module HC-05 • AranaCorp,» AranaCorp, 31 juillet 2018. [En ligne]. Available: <https://www.aranacorp.com/fr/votre-arduino-communique-avec-le-module-hc-05/> . [Accès le 2 juin 2023].
- [16] «Les vis trapézoïdales,» WIIP, [En ligne]. Available: <http://wiip.fr/node/8519> . [Accès le 5 juin 2023].
- [17] Douilles, «Roulements À Billes Linéaires,» RS, [En ligne]. Available: <https://fr.rs-online.com/web/c/roulements-et-joints/mouvements-lineaires/roulements-a-billes-lineaires/> . [Accès le 5 juin 2023].
- [18] e. bartman, *le grand livre d'arduino*, 2 éd., paris: eyrolles, 2022, pp. 34-66.
- [19] e. bartman, *le grand livre d'arduino*, 2 éd., paris: eyrolles, 2022, pp. 132-133.
- [20] P. Cignoni, G. Ranzuglia, M. Callieri, M. Corsini, F. Ganovelli, N. Pietroni et M. Tarini, «Meshlab,» IRIS, 18 February 2019. [En ligne]. Available: <https://air.unimi.it/handle/2434/625490>.
- [21] «SolidWorks : Logiciel de modélisation 3D par conception paramétrique,» Sculpteo, [En ligne]. Available: <https://www.sculpteo.com/fr/glossaire/solidworks-definition-fr/>. [Accès le 31 mai 2023].
- [22] M. Lombard, J. Wiley et Sons, *Solidworks bible*, willy, 2013.
- [23] «« App inventor »,» [En ligne]. Available: <https://info789.jimdofree.com/cours/9ème/app-inventor/>. [Accès le 7 juin 2023].
- [24] «Debuter app inventor,» sig.fgranotier.info , [En ligne]. Available: http://sig.fgranotier.info/IMG/pdf/debuter_app_inventor.pdf . [Accès le 7 juin 2023].

ANNEXE

Programme Arduino IDE

```
#include <SPI.h>
#include <SD.h>
int state = 0;
//Variables éditables
int scan_amount = 80 ; //Quantité de scans pour chaque point
String file="scan_001.txt"; //Nom du fichier enregistré sur la carte
SD
int z_axis_height = 12; //in cm //Hauteur maximale du fichier numérisé
int step_delay = 3000; //in us //Retard pour chaque étape pour le moteur
pas à pas en microsecondes
float z_layer_height = 1 ; //in mm //Hauteur de couche. La quantité de mm
pour chaque couche.
int lead_screw_rotations_per_cm = 12; //Combien de rotations la vis mère doit-
elle faire pour faire 1 cm.
int steps_per_rotation_for_motor = 200 ; //Étapes dont le moteur a besoin pour
une rotation complète.
int distance_to_center = 7.5; //Distance du capteur au centre du plateau
tournant en cm
//E/S
int button = 2;
int limit_switch = 10;
//Turntable driver pins
int dir_r = 7;
int step_r = 8;
int enable_r = 9;
//Z-axis driver pins
int dir_z = 3;
int step_z = 5;
int enable_z = 6;
//Variables
File file_values; //Utilisé pour le module de carte SD
int scan = 0; //Activer/désactiver la numérisation
int scan_changed = 0; //Le processus de numérisation a été modifié
float distance = 0; //Distance mesurée
float angle = 90; //Angle de rotation pour chaque boucle (0°-360°)
float x = 0; //Coordonnées X, Y et Z
float y = 0;
float z = 0;
```

```
int z_loop = 0;           //variable utilisée pour la rotation du moteur sur
                           l'axe z
int r_loop = 0;           //variable utilisée pour la rotation du moteur du
                           plateau tournant
float measured_analog = 0; //Lecture analogique du capteur de distance
float analog = 0;         //MOYENNE analogique
float RADIANS = 0.0;      //Angle en radians. Nous calculons cette valeur plus
                           tard dans la boucle de configuration
int steps_z_height = 0;   //Variable utilisée pour le nombre de pas sur l'axe z
int homed = 0;
void setup() {
  Serial.begin(9600);
  pinMode(A0, INPUT);
  pinMode(limit_switch, INPUT);
  pinMode(button, INPUT);
  analogReference(INTERNAL);
  SD.begin(4);
  pinMode(dir_z, OUTPUT);
  pinMode(step_z, OUTPUT);
  pinMode(enable_z, OUTPUT);
  pinMode(dir_r, OUTPUT);
  pinMode(step_r, OUTPUT);
  pinMode(enable_r, OUTPUT);
  pinMode(button, OUTPUT);
  pinMode(A0, OUTPUT);
  digitalWrite(enable_z,HIGH); //désactiver le pilote z_azis
  digitalWrite(enable_r,HIGH); //désactiver le pilote z_azis
  //Calculer les variables
  RADIANS = (3.141592 / 150.0) * (360/steps_per_rotation_for_motor);
  steps_z_height = (z_layer_height * steps_per_rotation_for_motor *
lead_screw_rotations_per_cm)/10;
  while(digitalRead(limit_switch) && homed == 0)
  {
    digitalWrite(enable_z,LOW); //activer le pilote z_azis
    digitalWrite(dir_z,HIGH); //z_axis tourne vers la gauche(abaissement de l'axe
z)
    digitalWrite(step_z,HIGH); //les z_axis font un pas
    delayMicroseconds(step_delay);
    digitalWrite(step_z,LOW);
    delayMicroseconds(step_delay);
  }
  homed = 1;
}
```

```
}
void loop() {
  if(Serial.available() > 0){ // Vérifie si les données proviennent du port série
    state = Serial.read(); // Lit les données du port série
  }
  if (state == '1') {
    digitalWrite(button, HIGH);
    Serial.println("MOTOR: ON"); // Renvoyez au téléphone la chaîne "MOTOR : ON"
    state = 0;
  }
  else if (state == '0') {
    digitalWrite(enable_z,HIGH);
    digitalWrite(enable_r,HIGH);
    delay(500000);
    Serial.println("MOTOR: OFF");
    state = 0;
  }
  //Attendre que le bouton poussoir soit appuyé
  if(digitalRead(button))
  {
    if(scan == 1 && scan_changed == 0)
    {
      scan=0;
      delay(3000);
      scan_changed=1;
    }
    if(scan == 0 && scan_changed == 0)
    {
      scan=1;
      delay(3000);
      scan_changed=1;
    }
    scan_changed = 0;
  }
  //Si le processus de numérisation est activé
  if(scan == 1)
  {
    //On s'arrête quand on atteint la hauteur maximale
    if(z < z_axis_height)
    {
```

```
for(int loop_cont = 0 ; loop_cont < steps_per_rotation_for_motor; loop_cont++)
{
    getDistance();
    digitalWrite(enable_r,LOW);    // active le pilote de la platine
    digitalWrite(dir_r,LOW);      //rotation du plateau tournant vers la droite
    digitalWrite(step_r,HIGH);    //faire un pas
    delayMicroseconds(step_delay);
    digitalWrite(step_r,LOW);
    delayMicroseconds(step_delay);
    angle = angle + RADIANS;      //Augmenter l'angle d'une unité
    supplémentaire
    write_to_SD(x,y,z);          //Écrire des fichiers x, y, z sur la fonction de
    carte SD
}
angle = 0; //Réinitialiser la valeur d'angle pour la prochaine rotation

while(z_loop < steps_z_height)
{
    digitalWrite(enable_z,LOW);   //activer le pilote z_axis
    digitalWrite(dir_z,LOW);      //z_axis tourne à droite
    digitalWrite(step_z,HIGH);    //z_axis faire un pas
    delayMicroseconds(step_delay);
    digitalWrite(step_z,LOW);
    delayMicroseconds(step_delay);
    z_loop = z_loop+1;           //Augmenter la boucle de 1
}
z = z + z_layer_height;        //Augmentez la hauteur z faite de 1 unité
z_loop = 0;                    //Réinitialiser la variable de rotation de
l'axe z

} //fin de si hauteur z
//On a fini le scan, on arrête les pilotes
else
{
    digitalWrite(enable_z,HIGH);
    digitalWrite(enable_r,HIGH);
}
} //si numérisation
} //Fin de "void loop"
```

```
//Fonction qui obtient la distance du capteur
double getDistance()
{
  for (int aa=0; aa < scan_amount; aa++)
  {
    measured_analog= analogRead(A0);
    delay(2);
    analog = analog + measured_analog;
  }
  distance = analog/scan_amount;          //Obtenez la moyenne. Divisez le scan par
le nombre de scans.
  analog=0;//réinitialiser la valeur totale de lecture andlog
  measured_analog = 0; //réinitialiser la valeur de lecture andlog
  distance = mapFloat(distance,0.0,1023.0,0.0,3.3); //Convertir la lecture de
broche analogique en tension
  distance = -1.9492 * pow(distance, 3) + 8.5459 * pow(distance, 2) - 10.311 *
distance + 4.0975; //De la fiche technique;
  distance = distance_to_center - distance; //la distance d = distance du
capteur au centre - distance mesurée
  y = (cos(angle) * distance);
  x = (sin(angle) * distance);
}
//Fonction qui mappe la valeur dans un format flottant
float mapFloat(float fval, float val_in_min, float val_in_max, float val_out_min,
float val_out_max)
{
  return (fval - val_in_min) * (val_out_max - val_out_min) / (val_in_max -
val_in_min) + val_out_min;
}
//Fonction qui écrit la valeur sur la carte SD
void write_to_SD(float SDx, float SDy, float SDz)
{
  file_values = SD.open(file, FILE_WRITE);
  if (file_values)
  {
    file_values.print(x); file_values.print(",");
    file_values.print(y); file_values.print(",");
    file_values.println(z);
    file_values.close();
  }
}
```