**3D Scanner**

Graduation Project Design

presented by

Waad Hammoud

Bashir Hatoum

Karim Olabi

To

Issam Fares Faculty of Technology

The Department of Mechatronics Engineering Technology

TECH391: Graduation Project Implementation

Guided By:

Dr. Gilbert Tekli

University of Balamand

Beirut, Lebanon

June 2023

Acknowledgment

We would like to express our gratitude and appreciation to our supervisor Dr. Gilbert Tekli for all his guidance and support. We are very thankful for your time and effort. Your useful advice and suggestions were very helpful to us.

**ABSTRACT**

**The goal of this project is to create a 3D scanner. This report will explore different types of scanners such as laser, structured-light, contact, and photogrammetry scanners. In order to find the best approach between these, previous research and projects were studied and analyzed to create the most suitable and cost-effective design. From there, a possible design for the 3D scanner is proposed and implemented. This design is separated into three parts: mechanical, electrical, and software design. The mechanical design includes a turntable rotating on a stepper motor with a Pi Camera moving horizontally and vertically relative to it via stepper motors. The electrical design describes the connections of the electrical components like the motors and their drivers, as well as pin assignments on the Raspberry Pi controller used. The software design includes python scripts on both the computer and the Raspberry Pi ends, as well as multiple software such as Meshroom, to implement photogrammetry, Wamp Server and VNC Viewer, to establish connections and automate the scanning process. Finally, the project will display the created prototype and the results of the testing done using it.**

Table of Contents

[1. Introduction 7](#_Toc136989752)

[2. Literature Review: 3D Scanning 8](#_Toc136989753)

[2.1. Laser Based 3D Scanners 8](#_Toc136989754)

[2.1.1. Projected or Structured Light 3D Scanners 9](#_Toc136989755)

[2.1.2. Contact Scanners 10](#_Toc136989756)

[2.1.3. Photogrammetry 10](#_Toc136989757)

[2.1.4. Others: Kinect 13](#_Toc136989758)

[3. Risk Assessment and Conceptualization 14](#_Toc136989759)

[3.1. 3D Scanning Method Assessment and Conceptualization 14](#_Toc136989760)

[3.2. Software Conceptualization 15](#_Toc136989761)

[3.3. Design Conceptualization 15](#_Toc136989762)

[3.4. Risks 16](#_Toc136989763)

[4. Application 18](#_Toc136989764)

[4.1. Mechanical Design 18](#_Toc136989765)

[4.2. Electrical Design 19](#_Toc136989766)

[4.3. Software Design 20](#_Toc136989767)

[4.3.1. Configuration 20](#_Toc136989768)

[4.3.2. Organigram 22](#_Toc136989769)

[5. Prototyping and Experimentation 25](#_Toc136989770)

[5.1. Prototype 25](#_Toc136989771)

[5.1.1. Vertical Axis 25](#_Toc136989772)

[5.1.2. Horizontal Axis 25](#_Toc136989773)

[5.1.3. Turntable 26](#_Toc136989774)

[5.1.4. Base 26](#_Toc136989775)

[5.1.5. Scanner 26](#_Toc136989776)

[5.2. Budget 27](#_Toc136989777)

[5.3. Experimentation 27](#_Toc136989778)

[6. Conclusion 30](#_Toc136989779)

[7. References 31](#_Toc136989780)

[8. Appendix 33](#_Toc136989781)

Table of Figures

[Figure 1 Faro 3D Scanner 8](#_Toc136989782)

[Figure 2 Fabscan Pi A Figure 3 Fabscan Pi B 9](#_Toc136989783)

[Figure 4 Projected Light Pattern 9](#_Toc136989784)

[Figure 5 Hexagon 3D Scanner 10](#_Toc136989785)

[Figure 6 Contact Scanner 10](#_Toc136989786)

[Figure 7 Multi-camera Photogrammetry 12](#_Toc136989787)

[Figure 8 DIY Photogrammetry Scanner 12](#_Toc136989788)

[Figure 9 Kinect 13](#_Toc136989789)

[Figure 10 Fixed Object Scanner A Figure 11 Fixed Object Scanner B 16](#_Toc136989790)

[Figure 12 Movable Camera and Object Scanner A Figure 13 Movable Camera and Object Scanner B 16](#_Toc136989791)

[Figure 14 Previous Mechanical Designs Figure 15 Final Mechanical Design 18](#_Toc136989792)

[Figure 16 Labeled Mechanical Schema 19](#_Toc136989793)

[Figure 17 Electrical Schema 19](#_Toc136989794)

[Figure 18 Scanner Flow Chart 22](#_Toc136989795)

[Figure 19 Raspberry Pi Script Flow Chart 23](#_Toc136989796)

[Figure 20 Computer Script Flow Chart 24](#_Toc136989797)

[Figure 21 Vertical Axis A Figure 22 Vertical Axis B 25](#_Toc136989798)

[Figure 23 Horizontal Axis 25](#_Toc136989799)

[Figure 24 Small Turntable Figure 25 Big Turntable 26](#_Toc136989800)

[Figure 26 Base 26](#_Toc136989801)

[Figure 27 3D Scanner 26](#_Toc136989802)

[Figure 28 Scanning 28](#_Toc136989803)

[Figure 29 Wamp Server Upload Figure 30 Meshroom Execution 28](#_Toc136989804)

[Figure 31 Final 3D Model 29](#_Toc136989805)

[Figure 32 Cleaned 3D Model 29](#_Toc136989806)

Table of Tables

[Table 1 Scanner Assessment 14](#_Toc136979360)

[Table 2 Budget 28](#_Toc136979361)

# Introduction

The aim of this project is to fully automate 3D scanning. 3D scanning allows for the creation of high-precision 3D models of real-world objects. This technology can be very useful in multiple disciplines. Anyone in manufacturing, engineering, design, development, surveying or testing can benefit from 3D scanning as it can be applied in any typical manufacturing cycle, saving time, money and material.

The most common use for 3D scanning is reverse engineering and prototype modelling. It can also be used in areas like the generation of 3D holographs of real objects, in product development and presentation. More particular uses of 3D scanners include giving doctors a close look inside their patients in a way that no other technology has ever given, another can be scanning historical objects without touching and having a clear diagram of it to save for the future.

With the use of Artificial Intelligence and Machine learning we can also make 3D scanners recognize objects according to previously learnt information, as well as combine and create new models on top of the old ones. An example can be using 3D scanning for human recognition for security systems.

In fact, 3D scanning technology also has immense potential in the apparel industry. It can revolutionize the way the apparel industry operates on all ends, from designers, to sellers, to consumers. It can maximize customer satisfaction while minimizing inventory cost as well as provide designers with different tools to expand their designs. 3-D body scanning technology can be useful to develop custom fit clothing and predicting sizes, custom pattern development, as well as trying on clothing virtually. It canprovide the customer with a virtual image of how he or she will look in a particular piece of clothing by combining the 3D model of the customer with existing models of the clothes using artificial intelligence and machine learning.

Furthermore, 3D scanners can be used in combinations with other 3D modeling machines such as CNC machines and 3D printers. Just recently, scientists have been using a combination of MRI and ultrasound imaging along with 3D-printing technology to help doctors prepare for fatal surgeries.

# Literature Review: 3D Scanning

This section will discuss different types of 3D scanners such as laser, projected or structured light, contact, and photogrammetry scanners as well as others such as the Kinect.

## Laser Based 3D Scanners

As the name suggests, this type of scanner works by projecting one or more laser lines onto the object to be scanned. One, or more, sensors placed at a known distance from the laser source, are used to detect the reflection of these lasers. These sensors are often photodiodes or laser rangefinders; and can either be used to detect the angle of reflection of the laser (laser phase-shift scanners) or the time it took for the laser to bounce back (laser pulse-based scanners). If they detect the first, the scanner uses a process called trigonometric triangulation to calculate the distance between each point of the object and the laser source. If they detect the later, the scanner uses the time of flight to calculate those distances. Either way, the scanner calculates and accumulates thousands of points from all around the object. These points are then recorded and rendered as cartesian coordinates to form what is called a point cloud. This point cloud can be used to create a mesh or CAD file of the object.

An example of a laser 3D scanner is the Faro scanner (Figure 1) [1]. This scanner is mostly used to capture buildings, facilities and complex environments.



Figure 1 Faro 3D Scanner

Another is Fabscan 3D, an open-source Raspberry Pi based 3D scanner (Figures 2-3) [2-3]. Fabscan also offers a DIY kit called the Fabscan Pi scanner which uses a Pi camera and a turn-table to capture all sides of the object in high-quality resolution.

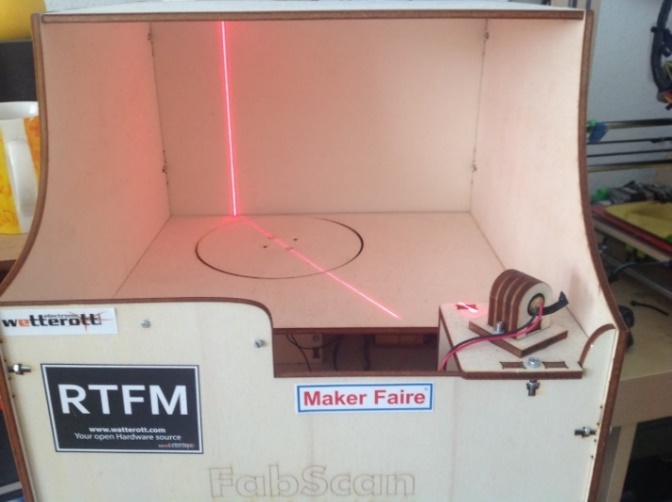
 

Figure 2 Fabscan Pi A Figure 3 Fabscan Pi B

### Projected or Structured Light 3D Scanners

These scanners use either white or blue light that is projected onto the object in a pattern that can be bars, blocks, stripes (Figure 4), dots, etc. Upon being projected onto the object, these patters become distorted. The 3D scanner has one or more sensors, that look at the edge of those patterns or structural shapes to determine the object's 3D shape. Typically, these sensors are cameras (such as a CCD or CMOS sensor).



Figure 4 Projected Light Pattern

Examples of structured light scanners are Hexagon Scanners (Figure 5) [4]:

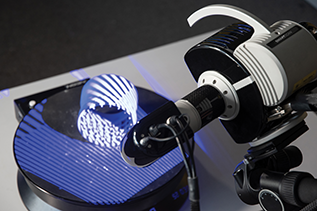


Figure 5 Hexagon 3D Scanner

These types of scanners are also simple enough to make using a light projector and one or two cameras [5].

### Contact Scanners

As the name suggests, this type of scanner requires physical contact with the object. For this type of scanner, the object is secured firmly and the scanner uses a probe or articulated arms with position encoders to touch the object from all sides (Figure 6). The scanner collects details about the object to create a 3D model [6].



Figure 6 Contact Scanner

### Photogrammetry

The term photogrammetry means “measuring from photos”. Several overlapping photos are captured from all around the object and at different angles. These photos can be captured using a regular camera or smartphone, or even multiple cameras setup in a booth to surround the object (Figure 7) [7]. These photos are then stitched together using special software. The software calculates the cartesian coordinates of each point on the object by mathematically reconstructing the rays representing the line of sight from the camera to the point. The software then identifies the pixels that correspond to the same physical point and combines the pictures together accordingly to produce a colored 3D model.

Photogrammetry determines the 3D model from photos by leveraging the principles of geometry and computer vision. The following summarizes the steps used in most photogrammetry software in reconstructing the 3D model:

* Image Acquisition: A series of overlapping photographs of the object or scene are captured from different viewpoints. These photographs can be taken using specialized photogrammetry equipment, such as calibrated cameras, or even with consumer-grade cameras.
* Feature Extraction: In this step, distinct features or points of interest are identified in each photograph. These features can be corners, edges, or other recognizable patterns that can be reliably identified across multiple images. Modern photogrammetry software often uses computer vision techniques to automatically extract these features.
* Correspondence Matching: The next step involves establishing correspondences between the features detected in different photographs. The software searches for matching features in multiple images and determines which features correspond to the same point in 3D space. By finding these correspondences, the software establishes the connections between different viewpoints of the object or scene.
* Camera Calibration: To accurately reconstruct the 3D geometry, it is essential to calibrate the cameras used to capture the images. Camera calibration involves determining the intrinsic parameters (e.g., focal length, lens distortion) and extrinsic parameters (e.g., camera position, orientation) for each camera. Calibration ensures that the software can accurately relate the 2D image points to their corresponding 3D positions.
* Triangulation: Once the correspondences are established and the cameras are calibrated, the software performs triangulation. It calculates the 3D position of each feature point by intersecting the corresponding rays projected from different camera viewpoints. This process creates a sparse point cloud representation of the object or scene, or in other words, determining the 3D coordinates of the identified features.
* Dense Point Cloud Generation: The sparse point cloud is then expanded into a dense point cloud by estimating the 3D positions of additional points within the surfaces of the object. This is done by analyzing the intensity values or colors in the images and applying interpolation techniques.
* Surface Reconstruction: The dense point cloud is used to generate a 3D surface mesh that represents the object’s shape. Various algorithms, such as Delaunay triangulation or Poisson surface reconstruction, can be employed to create a continuous surface from the point cloud data. The surface mesh provides a visual representation of the object’s geometry.
* Texture Mapping: To add visual details to the surface mesh, the original photographs are mapped onto the reconstructed geometry. This process involves projecting the 2D images onto the corresponding 3D surfaces, aligning the textures with the geometry. This step enhances the realism of the model by applying the original image details to the surface.
* Refinement and Optimization: The initial reconstruction may contain errors or inconsistencies due to noise, calibration inaccuracies, or other factors. Therefore, refinement techniques like bundle adjustment are employed to optimize the camera parameters, refine the 3D positions of points, and improve the overall accuracy of the model. These steps help to minimize errors and align the reconstructed 3D model with the original photos.
* Output: The final output of photogrammetry is a textured 3D model that can be visualized, analyzed, or further processed using appropriate software. The model can be exported in various formats compatible with 3D modeling, computer graphics, or virtual reality applications.

Some photogrammetry software includes: Reality capture, Blender, Meshroom, 3DF Zephyr, Agisoft Metashape, Autodesk Recap etc.



Figure 7 Multi-camera Photogrammetry

There are multiple existing DIY photogrammetry-based scanners.

One, in particular, (Figure 8) uses a DSLR camera, an Arduino, a stepper motor and driver, an LCD screen, and an IR LED. The goal of the hardware is to build a rotating platform that moves by set amounts so that the camera can photograph the object in a very detailed and controlled way. It also offers three examples of photogrammetry software that can be used to process the images obtained: Agisoft Photoscan, Autodesk Memento, and 123d Catch [9].

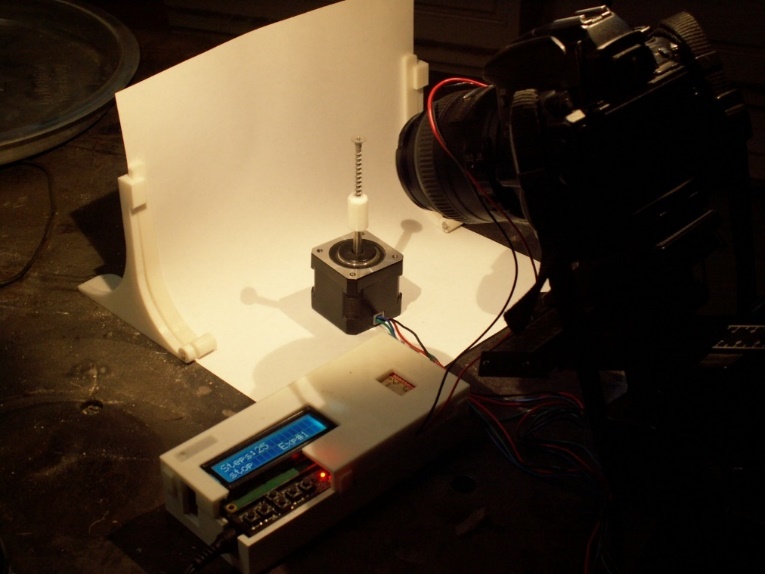


Figure 8 DIY Photogrammetry Scanner

### Others: Kinect

A well-known 3D scanner is the Kinect 3D scanner (Figure 9) which combines multiple technologies particularly photogrammetry and structured light scanning [8]. It consists of RGB cameras and infrared (IR) sensors. The process involves taking multiple pictures of the object from different angles using the cameras while the IR sensors are used to calculate the field of depth of the object and its surrounding to create a depth map. While Kinect V1 uses triangulation, Kinect V2 uses time of flight calculations. In fact, the Kinect is compatible with third party software such as Skanect and 3D Scan which allows the user to easily obtain 3D models of the scanned object.



Figure 9 Kinect

# Risk Assessment and Conceptualization

This section will discuss the different assessmnets made before implementing the design. This includes an assessment of the different scanning methods as well as an assessment of the risks involved with the software and hardware design.

## 3D Scanning Method Assessment and Conceptualization

Different characteristics of each type of scanner were assessed to find the best approach for the design. These characteristics include: cost, accuracy, time, safety, sensitivity and color detection (Table 1).

Table 1 Scanner Assessment

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Type** | **Cost** | **Accuracy** | **Time** | **Safety** | **Sensitivity/ Noise** | **Color** |
| Laser | Expensive. Can cost anywhere from 100$ to 300$ or even more | Generally, provides an acceptable level of accuracy. They are suitable for capturing fine details and textures and are commonly used for industrial applications. | Fast. | Biggest concern is eye-safety especially if intended for human scanning. | - Ambient light may blend with the laser and interfere with the scan’s accuracy. It is recommended that lighting be controlled.  - Struggles with scanning complex geometries and can be affected by surface reflectivity and color. | Without the use of a camera, color cannot be detected. |
| Structured Light | Expensive. Can cost anywhere from 100$ to 400$ or even more, depending in resolution and accuracy. | - High precision and accuracy.  - They can capture dense point clouds and their results are more accurate and detailed than Laser scanning. | Fast. | Safe. | - Easily affected by lighting conditions. Works best in pitch-black rooms.  - Glossy or transparent surfaces may cause issues, as they can distort the projected pattern. | Without the use of a camera, color cannot be detected. |
| Contact | Expensive. Can cost anywhere from 100$ to 300$ or even more | Accurate. | Very slow. | Safe. Although considered invasive due to its physical nature. | Unaffected by light. Transparent and reflective surfaces can still be accurately scanned. | Without the use of a camera, color cannot be detected. |
| Photogrammetry | Relatively cheap. Most components can be found in day-to-day use. | High accuracy and reliability. | Speed depends on the object to be scanned as well as processing power. Can take anywhere from 5 minutes to hours. | Safe. | - Affected by lighting conditions.  - Textured objects (hair, sand…) may be difficult to recognize.  - It is recommended that the object be placed with a featureless black or white background with no shadows. | Obtained 3D models are colored. |

Based on this, the chosen method for 3D scanning was photogrammetry.

## Software Conceptualization

Furthermore, out of the aforementioned photogrammetry software, Meshroom was chosen. Meshroom is a free, open-source 3D Reconstruction Software based on the AliceVision framework which is a Photogrammetric Computer Vision Framework that provides 3D Reconstruction and Camera Tracking algorithms. Meshroom was chosen because of its desirable results, ease of use as well as it being inexpensive.

## Design Conceptualization

As seen in the literature review, there are multiple possible designs for 3D scanners. The main difference in these designs is whether it is the camera or the object that moves and along which axis or axes. The axes of movement may differ but the designs can be categorized into the following:

* Fixed object and movable camera (Figures 10-11) [10].

Figure 10 Fixed Object Scanner A Figure 11 Fixed Object Scanner B

* Movable camera and movable object (Figures 12-13) [11-12].

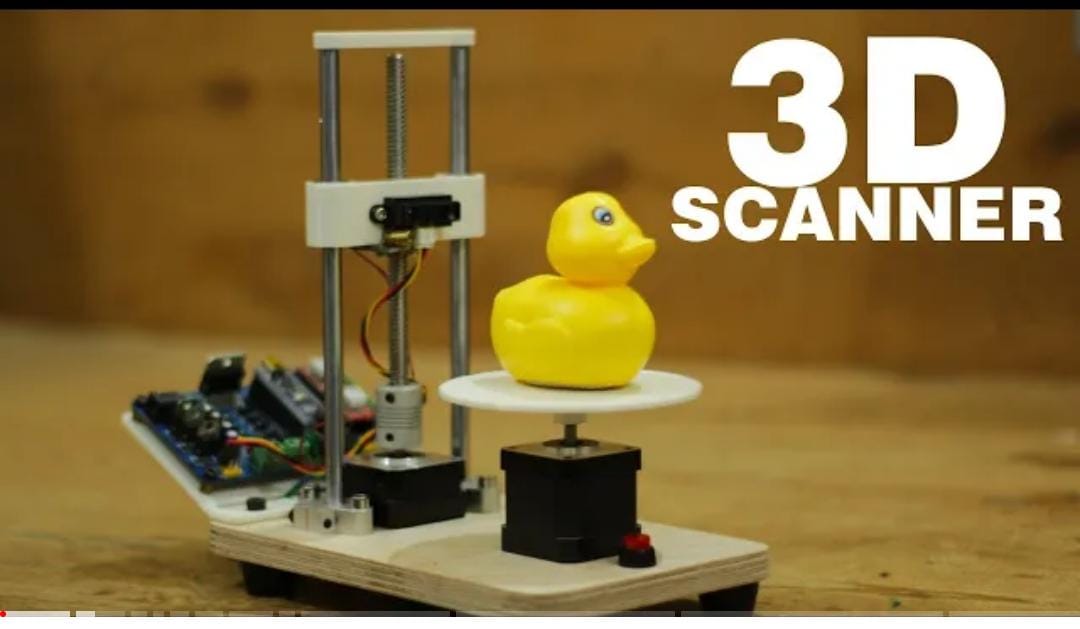
 

Figure 12 Movable Camera and Object Scanner A Figure 13 Movable Camera and Object Scanner B

* Fixed camera and movable object. It is uncommon for the camera to be fixed, with the only real case being the multi-camera example seen previously.

After reviewing these designs, for this project, the following was decided: The object would rotate and one camera translates along two axes, horizontal and vertical. This way the cost is reduced but the quality is not compromised.

Furthermore, a microcontroller was needed to control the motors and camera. For this project, the microcontroller chosen is the Raspberry Pi 3 Model B. The main reason for this choice is the Raspberry Pi’s compatibility with the Pi Camera. Furthermore, in comparison to other controller boards like the Arduino, the Raspberry Pi seems more suited for IoT applications which will be required in this project. It can collect data from several sensors, control actuators, access the Internet to pull or push data, connect to a smartphone and provide a complex output on a display.

## Risks

Different possible risks were assessed to ensure that the design could be implemented. These included 5 main risks on both the hardware and software ends.

**Risk 1: Can the software be automated?**

For the scanner to be automated, the photogrammetry software chosen needed to be able to run without any direct user input. This was easily resolved since Meshroom offers a CLI (Command Line Interface). These are command line features and tools that can be used to run and control Meshroom from the command line without the use of its GUI (Graphical User Interface).

**Risk 2: Will a Raspberry Pi Camera work for this project’s application?**

There are no particular requirements for the camera in Meshroom. With that said, the image quality is important and has a large impact on the quality of the final mesh. The Raspberry Pi Camera V2 is capable of 3280 x 2464-pixel static images, which is enough for this project’s application.

**Risk 3: Can the Raspberry Pi 3 handle controlling multiple motors?**

The Raspberry Pi 3 can control multiple stepper motors, but a separate power supply must be used for these motors.

**Risk 4: Can we send images from the Raspberry Pi to Computer?**

There are multiple ways to send images from the Raspberry Pi to the computer including: SSH, Bluetooth, email, FTP, or over WIFI. For our application, the chosen method was the last one, particularly, the Raspberry Pi uploads images to the computer via a server.

**Risk 5: Can the motors carry the weight required?**

To rotate the object a full 360 degrees, one motor is required. Also, the object's weight was limited to around 2 Kg or less. Therefore, based on its torque, a Nema 17 stepper motor was sufficient to turn this weight.

Similarly, the vertical axis needed a motor to move the camera up and down. Therefore, the weight that this motor needed to carry was estimated. This weight included the lead screw, wooden base, the raspberry pi and the breadboard. Therefore, once again a Nema 17 had sufficient torque to be used for the vertical axis.

Similarly, since the vertical axis was to be mounted on the horizontal one, the weight of the entire vertical axis itself was estimated. This weight included that of the 2 metal rods, lead screw, wood, the vertical axis motor, the raspberry pi and breadboard. As a result, another Nema 17 was once again sufficient to move the horizontal axis.

Therefore, all the risks were resolved and the project could be implemented.

# Application

With the concept confirmed, this section will move on to discuss the actual mechanical, electrical, and software design and application.

## Mechanical Design

Autodesk Inventor Professional was used to simulate the mechanical design. In order to generate a 3D mesh, photos needed to be taken from all around the object in 360° and at different heights, depending on the size of the object. Therefore, after multiple trial designs (Figure 10), the final design was obtained (Figures 11-12). The object to be scanned is placed on a turntable controlled by one motor. This turntable sits on 3 metal rods connected to free-wheels for support. A second motor is used to move the Pi Camera along a vertical axis, and a third motor is used to move it along a horizontal axis. Each of these axes consists of 2 metal rods and one lead screw connected to their motor. These lead screws each move a nut along their axis when their motor is turned. This allows the bases mounted on top of their nut to move as well.

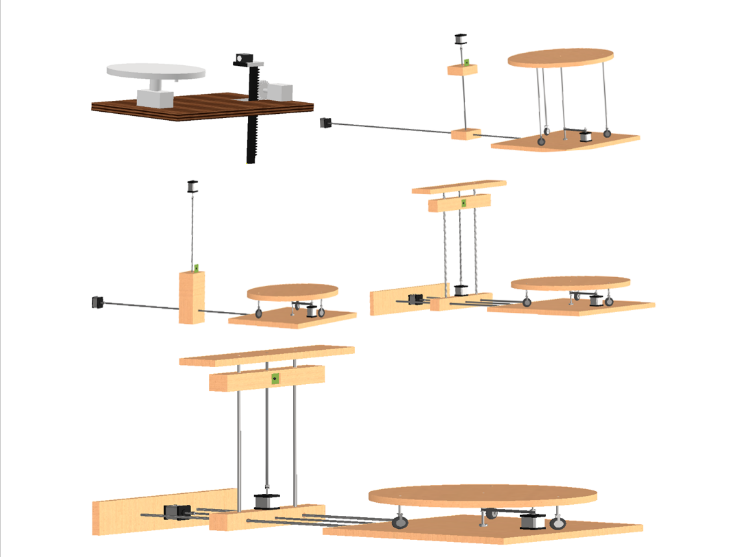
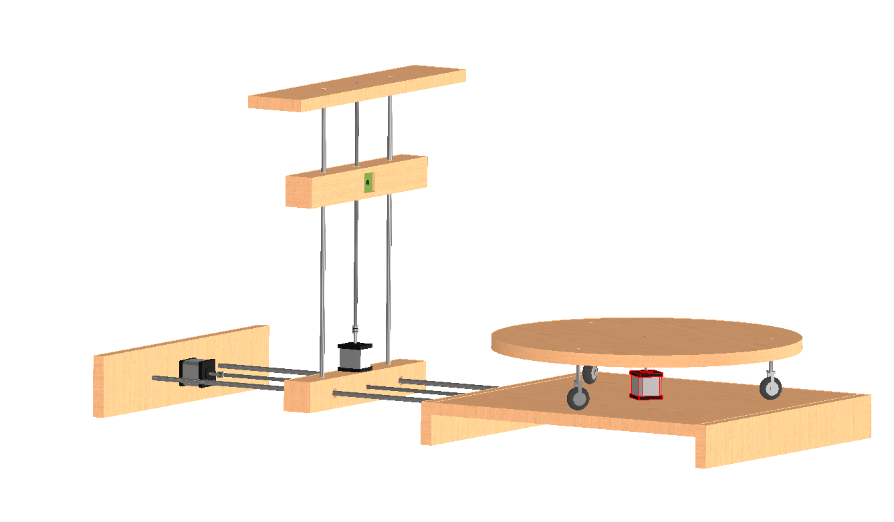
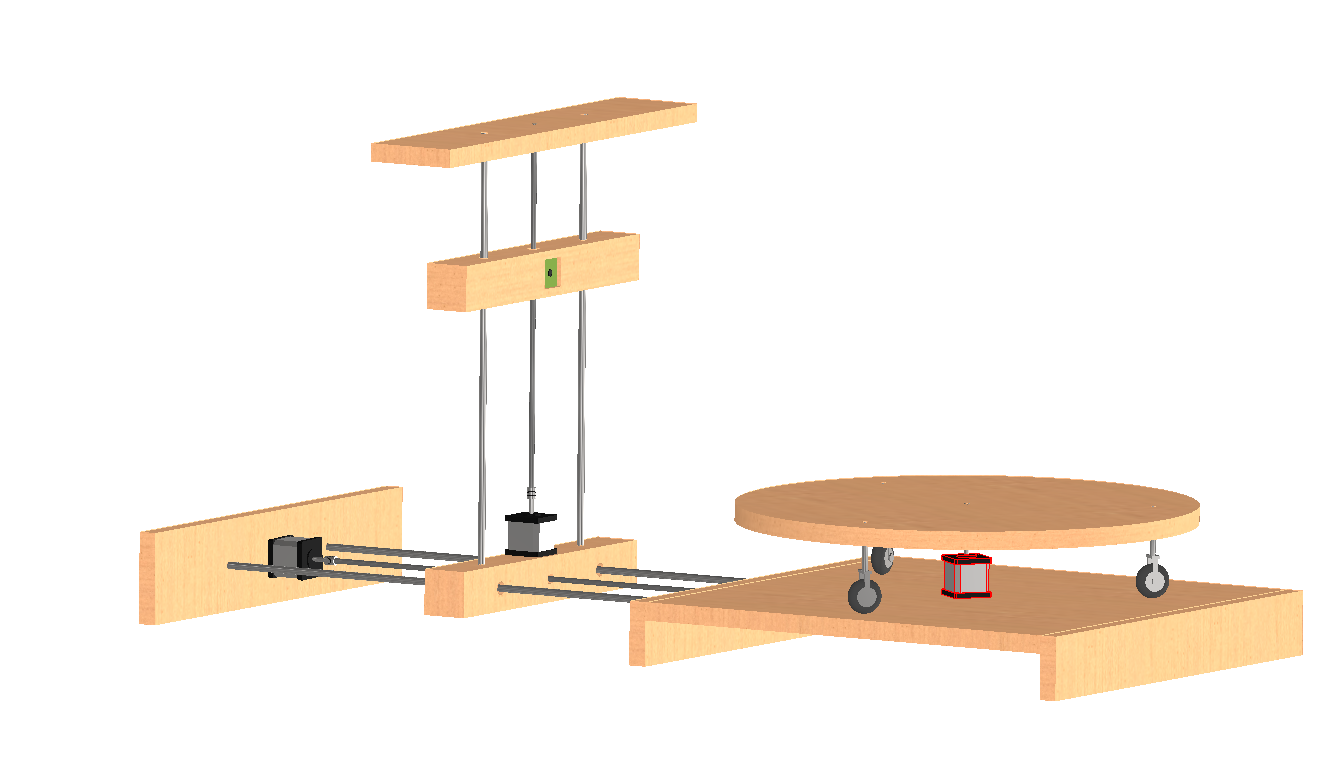
 

Figure 14 Previous Mechanical Designs Figure 15 Final Mechanical Design

The basic components needed for the mechanical design were identified (Figure 16).





T-Nuts

Rolling Bearings

Linear Bearings

Free Wheels

Metal Rods

Lead Screws

Pi Camera

Couplers

Motors

Figure 16 Labeled Mechanical Schema

## Electrical Design

This project uses the following electrical components to be connected together (Figure 17):

* Raspberry Pi
* 3 Motor Drivers
* 3 Nema 17 Stepper Motors
* 3 100 capacitors
* 12V, 15A charger

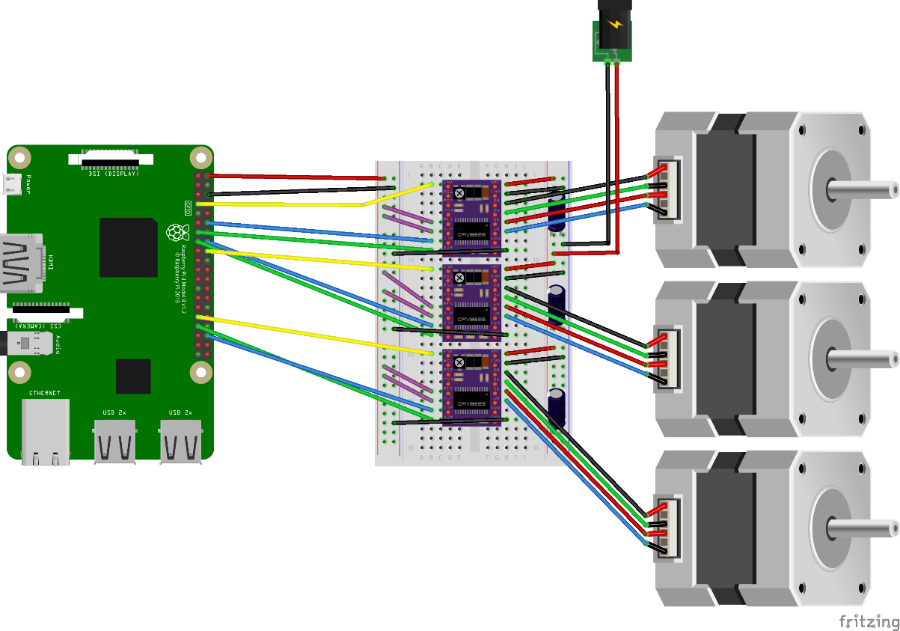


Figure 17 Electrical Schema

## Software Design

This project requires the user to download the following software:

* Python: The programming language used for this project is Python. Therefore, the user needs to download Python in order to run the Python scripts or they can download a Python IDE like Thonny.
* Meshroom: As aforementioned, the photogrammetry software used is Meshroom. Meshroom requires a CUDA-enabled NIVADA GPU. If this is not available, an alternative version of Meshroom, Meshroom CL, can be used, but it requires more time and yields less accurate results.
* MeshLab: MeshLab is a 3D mesh processing software system that provides a set of tools for editing, cleaning, healing, inspecting, rendering, and converting meshes. In this project, this software is used to view the 3D mesh once it is generated.
* VNC Viewer: VNC allows a client to connect to a server, and interact with the desktop of the remote machine. In this case, the client refers to the computer and the server refers to the Raspberry Pi. Therefore, this software is used to connect to the Raspberry Pi and will allow the user to access the live feed from the camera while scanning is taking place.
* Wamp Server: Wamp acts like a virtual server on your computer. In this project, it is used to receive images from the Raspberry Pi on the computer.

### Configuration

Before scanning, the user needs to configure a few things on both the Raspberry Pi and the computer end.

Raspberry Pi Configuration

The Raspberry Pi needs to be connected to the same network as the computer. To do that simply use an HDMI cable to view the Raspberry Pi Desktop and connect it to the network. Furthermore, the python script on the Raspberry Pi requires the following libraries:

* pynput: This library allows for the monitoring of input from the mouse and keyboard.
* RPi.GPIO: This is a module used to control Raspberry Pi GPIO channels.
* selenium: This library offers a range of tools for browser automation. In our application, the browser used on the Raspberry Pi end is the Chromium browser. Therefore, both Chromium and its designated Chromium browser should be installed on the Raspberry Pi.
* RpiMotorLib: This library allows the Raspberry Pi to drive motor controllers and servos.

These libraries can be installed by typing: “sudo pip install library\_name” in the Raspberry Pi terminal.

Wamp Server Configuration

* Open Wamp Server.
* Click on the Wamp Icon in the taskbar.
* Go to Apache->httpd-vhosts.conf
* Once the file is open, change “Require local” to “Require all granted”

Next,

* Open Windows Security -> Firewall & Network Security
* Click Allow app through firewall -> Change Settings -> Allow another app.
* Find and choose C:/wamp64/bin/apache/apache2/bin/bin/httpd.exe.
* Restart Wamp.

This will enable local network users to access this computer’s Wamp sites by simply typing this computer’s IP address in a browser.

Next,

* Go to C:\wamp64\www
* Create a folder called “Scanner”
* Copy the following files into the “Scanner” folder:
  + connection.vnc
  + fileUpload.php
  + index.php

The two php folders are the website that the Raspberry Pi uploads images to. These images are directly saved locally on the computer in the “Scanner” folder. The connection.vnc file is the one that will allow the computer to make a VNC connection with the Raspberry Pi. For more about these files, view the [Appendix.](#_Appendix)

Next,

* Open Wamp.
* Click on the Wamp Icon in the Taskbar.
* Note the PHP version used next to the PHP option.
* Click on PHP->php.ini
* If there are multiple PHP versions, choose the one noted previously.
* Once the file is open, use Ctrl-F to search for and change the following:
  + post\_max\_size = 750M
  + upload\_max\_filesize = 750M
  + max\_execution\_time = 5000
  + max\_input\_time = 5000
  + memory\_limit = 1000M
* Restart Wamp

This will increase the file size that can be uploaded on Wamp Server. This is needed because the zipped file containing the images which the Raspberry Pi uploads to the server will be greater in size than the default limit.

It is also important to note that before starting the scanning process, the user must make sure Wamp server is running.

### Organigram

The connection between the Raspberry Pi and the computer is established using multiple methods. The first is by broadcasting and receiving messages on the network. This allows each of them to know the other's IP address on the local network. Once this is done, a connection is established using VNC. This will allow the user to view the live camera feed, first to calibrate and then to view the scanning procedure. Finally, the VNC connection is killed and the Raspberry Pi uses Wamp Server to upload images, which are then received by the computer. To enable this upload of files on Wamp Server, a small website is made that allows users to upload files into a specific folder on the computer that hosts Wamp Server. The communication between the computer and the controller can be represented using a flow chart (Figure 18).

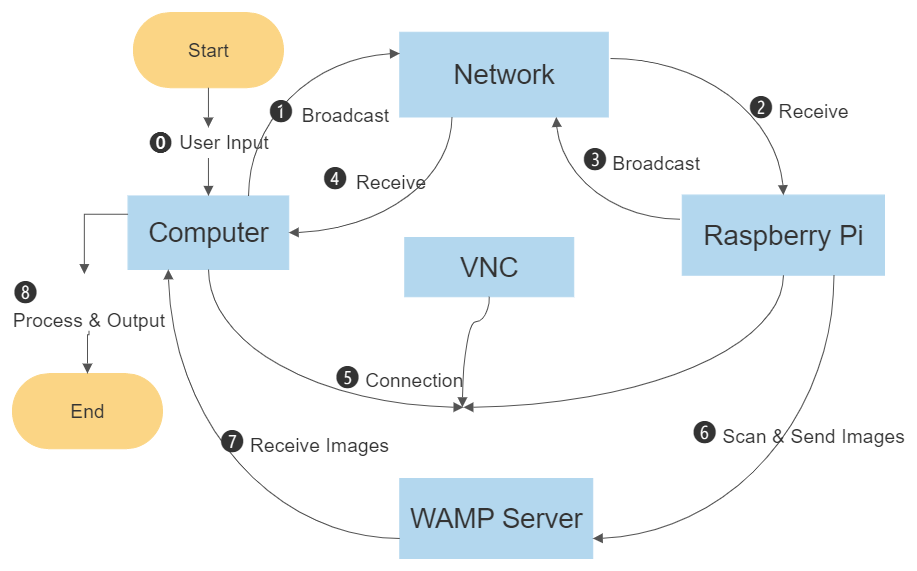


Figure 18 Scanner Flow Chart

The scanning process can be separated into two parts or scripts that work together: Computer end and Raspberry Pi end. These python scripts handle the communication between the computer and the Raspberry Pi, as well as their separate operation.

The raspberry pi script handles motor calibration as well as scanning, which includes moving the motors, taking pictures using the Pi Camera, saving them, zipping them and uploading them. Meanwhile, the computer script handles finding and calling the different programs, like Meshroom and MeshLab, unzipping the picture folder received, uploading it to Meshroom, executing the scanning, retrieving the results, and presenting them to the user via MeshLab.

Raspberry Pi Script

The Raspberry Pi script runs on start-up. This is done by creating a .desktop file and placing it in the autostart folder of the Raspberry Pi (“/home/pi/.config/autostart”). The Raspberry Pi end of the project can be represented using a flow chart (Figure 19).

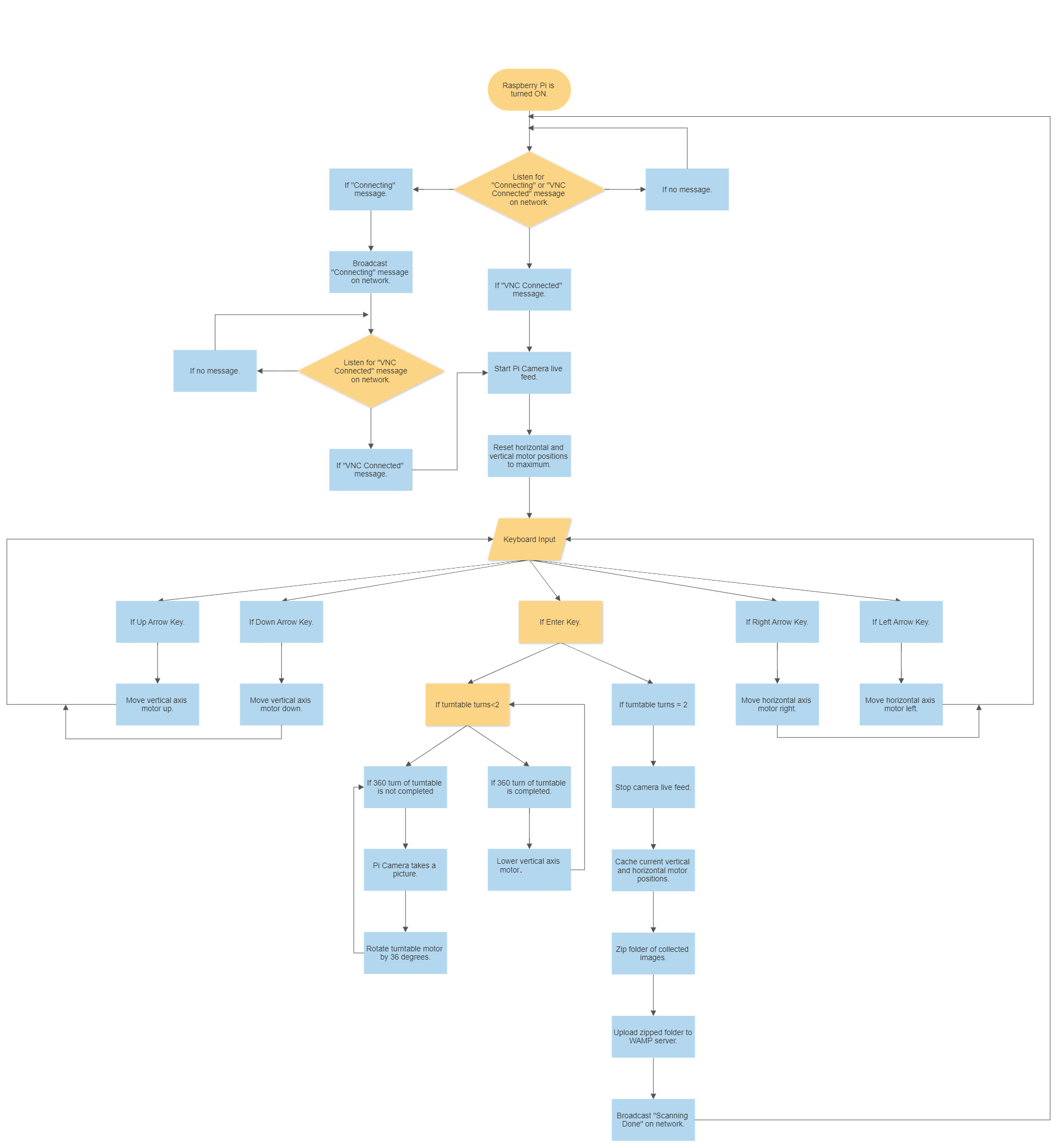


Figure 19 Raspberry Pi Script Flow Chart

Computer Script

The computer end of the project can also be represented using a flow chart (Figure 20).

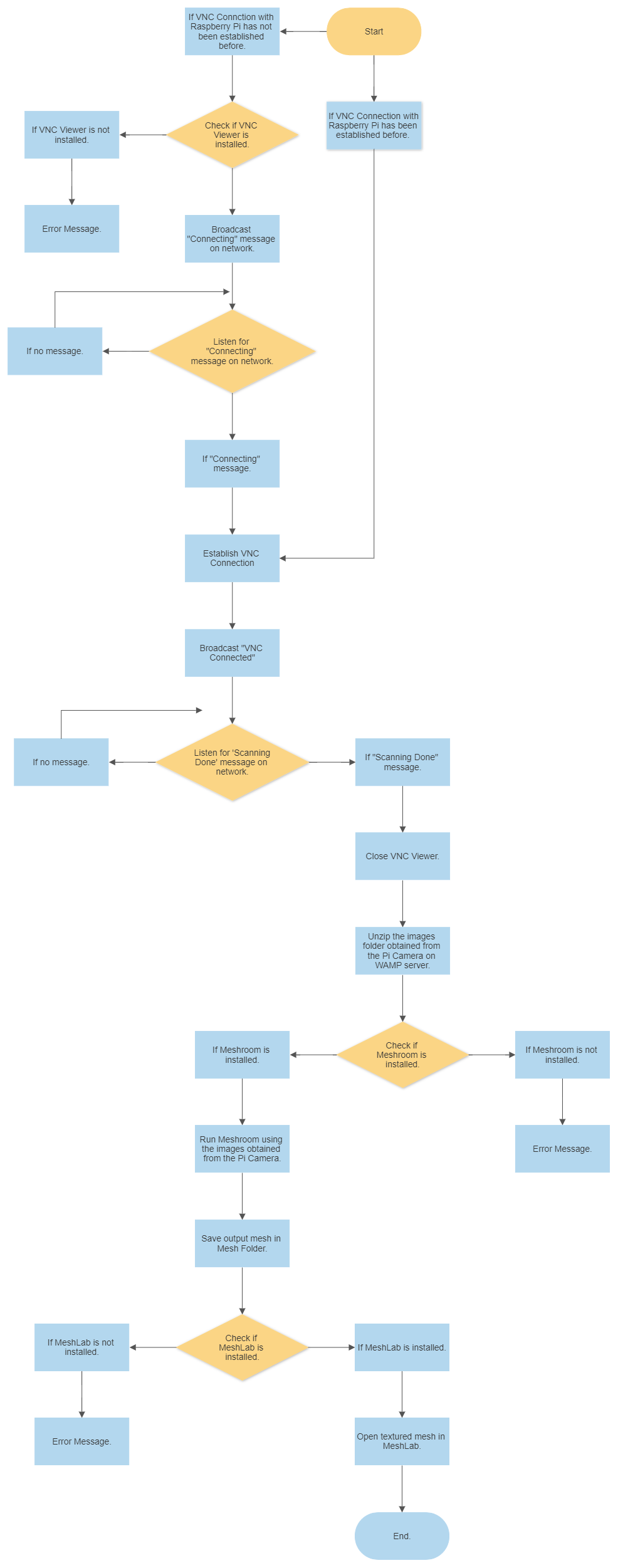


Figure 20 Computer Script Flow Chart

For more about these codes, view the [Appendix.](#_Appendix)

# Prototyping and Experimentation

This section presents an overview the prototype and its different features, the budget used, as well as the results of the experimentation.

## Prototype

The machine (mechanical schema) was built using thick cardboard. This section will show all its different parts, the vertical and horizontal axis, the turntable and the base, as well as how these parts were connected together to make the end product.

### Vertical Axis

The vertical axis looked as shown in Figures 21-22.

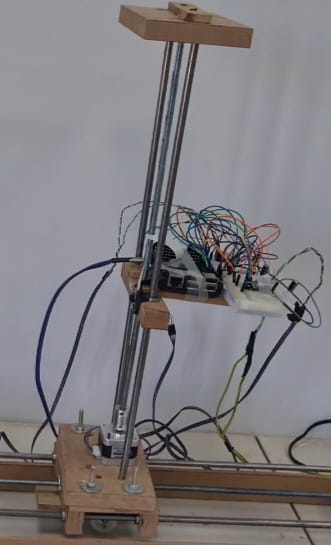
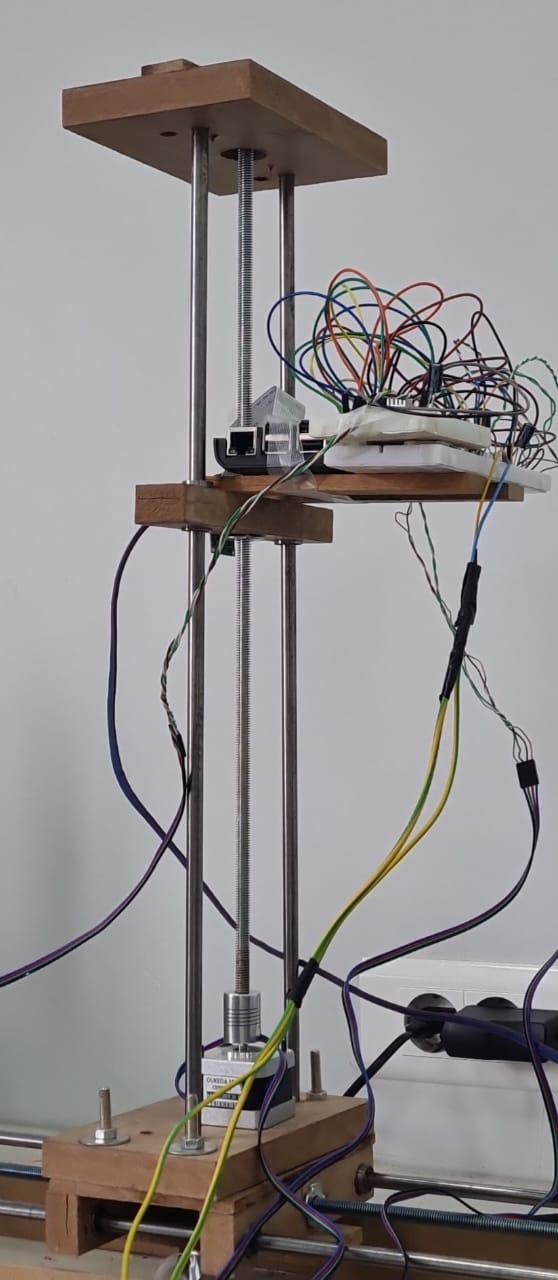
 

Figure 21 Vertical Axis A Figure 22 Vertical Axis B

### Horizontal Axis

The horizontal axis looked as shown in Figure 23.

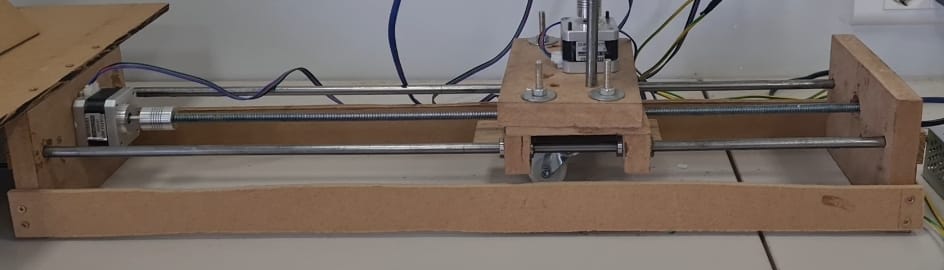


Figure 23 Horizontal Axis

### Turntable

Two versions of the turntable were made, a small one (Figure 24) and a big one (Figure 25). However, only the small one was used because the bigger one required a bigger stepper motor which could not be obtained currently.

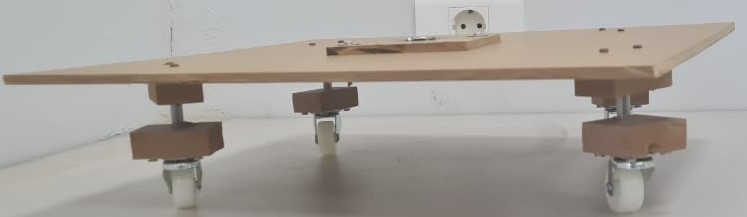
 

Figure 24 Small Turntable Figure 25 Big Turntable

### Base

The base looked as shown in Figure 26.



Figure 26 Base

### Scanner

These parts were connected together to make the final 3D Scanner (Figure 27).

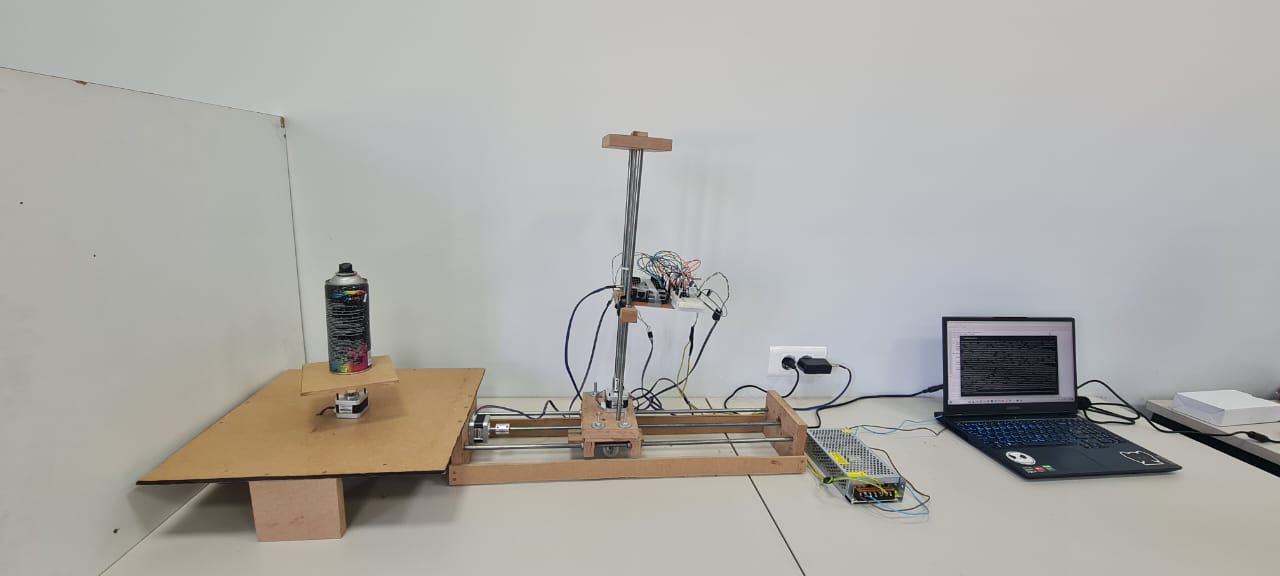


Figure 27 3D Scanner

## Budget

The cost of all the components were recorded and added to obtain the final budget for the project (Table 2).

Table 2 Budget

|  |  |  |  |
| --- | --- | --- | --- |
| **Component** | **Quantity** | **Unit Price** | **Total Price** |
| Raspberry Pi 3 Model B | 1 | $85 | $85 |
| Nema 17 Stepper Motor | 3 | $16 | $48 |
| 16GB Micro SD Card | 1 | $10 | $10 |
| Pi Camera | 1 | $30 | $30 |
| 12V, 15A Power Supply | 1 | $9 | $9 |
| Breadboard | 1 | $1 | $1 |
| Wood | 1 | $50 | $50 |
| 2.5 m Metal Rod of 8mm diameter | 1 | $4 | $4 |
| 1m Lead Screw of 8mm diameter | 1 | 0.4$ | 0.4$ |
| CNC Linear Bearing LM 8UU | 4 | $2 | $8 |
| Caster Wheel [25 mm diameter] [35 mm height] | 4 | $3 | $12 |
| T-Nuts (pack) | 1 | 0.6$ | 0.6$ |
| 9mm Rolling Bearing | 3 | $1 | $3 |
| Flexible Coupler D19L25 (5\*8mm Hole) | 2 | $3 | $6 |
| 3D Printer GT2 Pulley 8mm 20T | 1 | $5 | $5 |
| 3D Printer GT2 Pulley 5mm 20T | 1 | $5 | $5 |
| 3D Printer GT2 Belt 288mm | 1 | $4 | $4 |
| - | - | - | $281 |

## Experimentation

For the experimentation, a spray can was used as the object to be scanned. The scanner was turned on, calibrated, and the object was scanned (Figure 28).

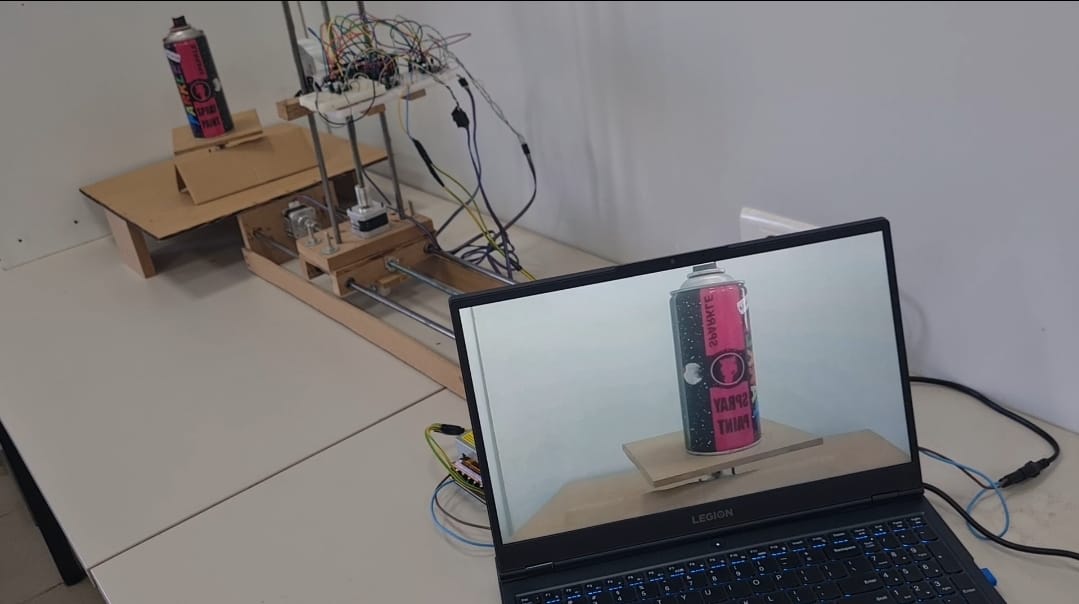


Figure 28 Scanning

Next, the photos were uploaded unto Wamp Server automatically (Figure 29) and Meshroom was run automatically as well (Figure 30).

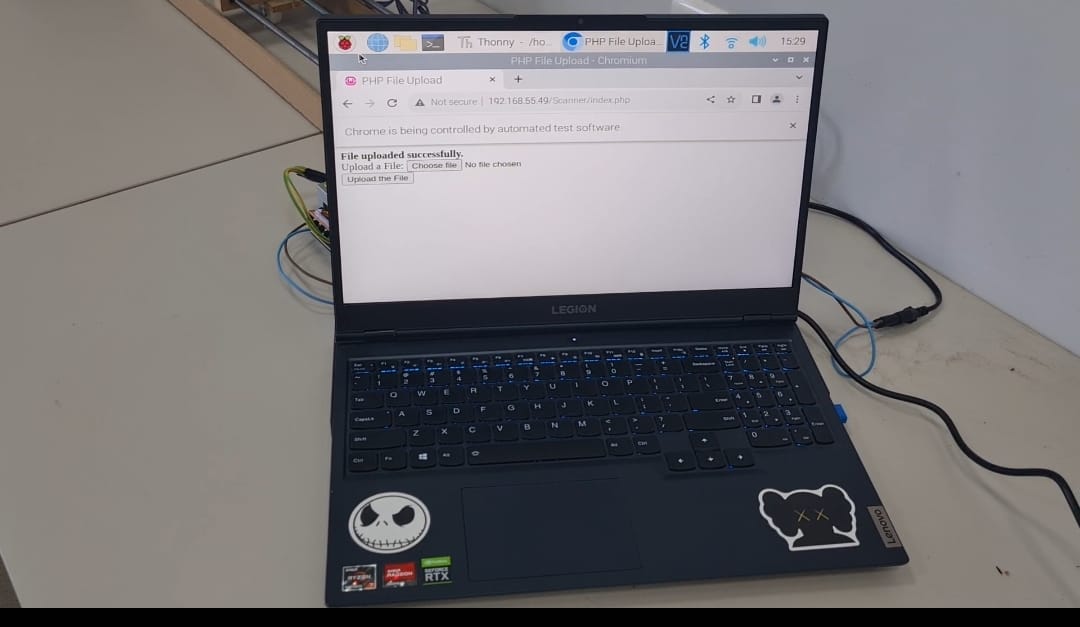
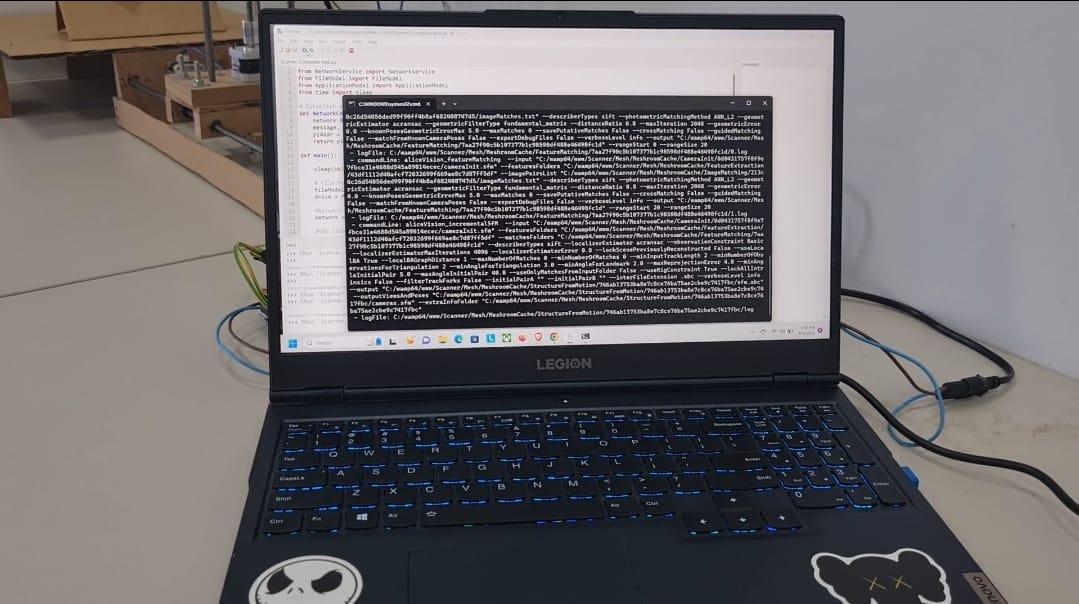
 

Figure 29 Wamp Server Upload Figure 30 Meshroom Execution

Finally, the program opened the final 3D model in MeshLab (Figure 31).

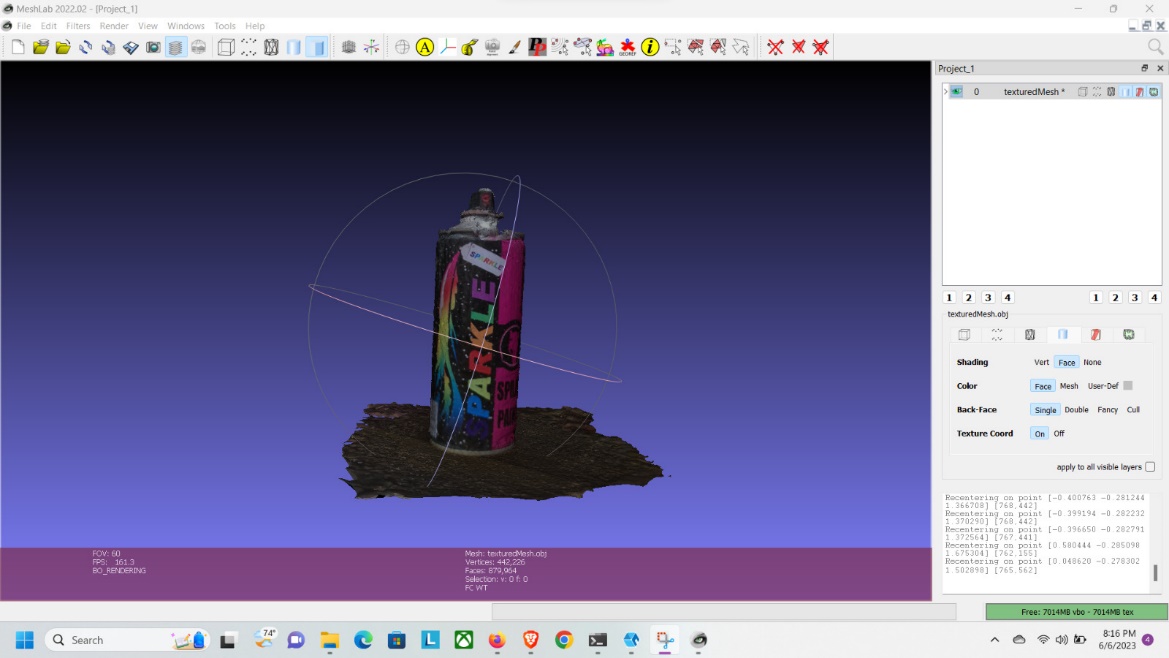


Figure 31 Final 3D Model

This model was cleaned up manually to remove any excess background (Figure 32).

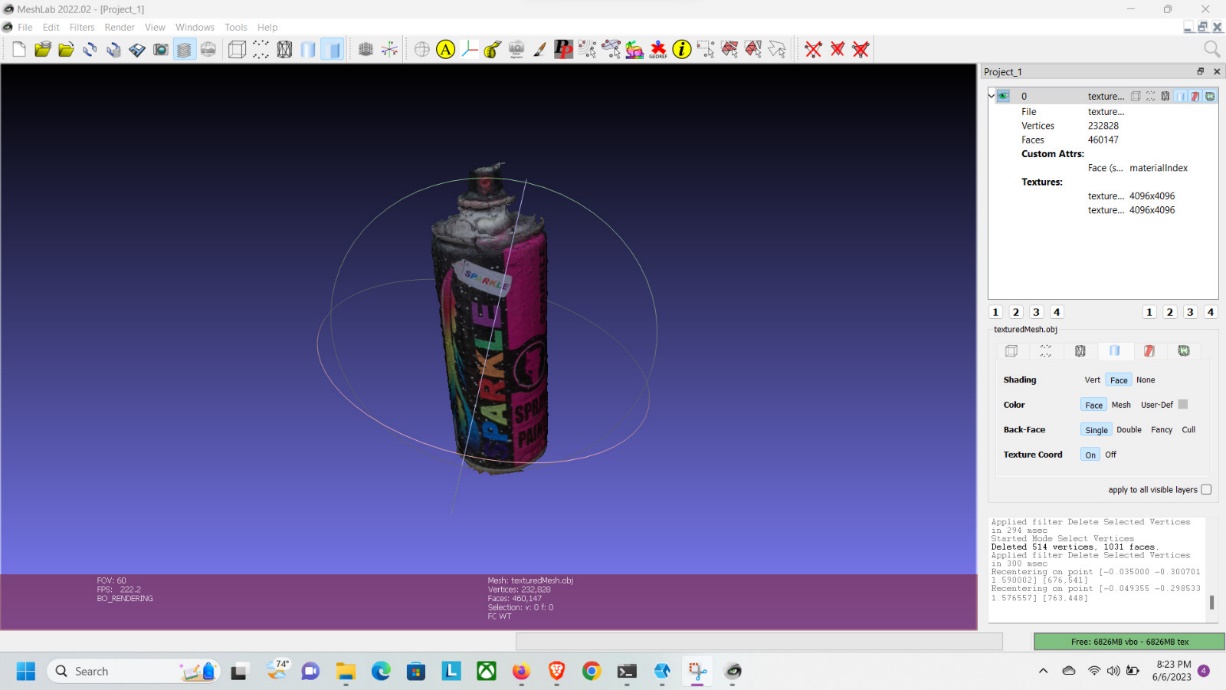


Figure 32 Cleaned 3D Model

This final output was an .obj file. It is later exported as a .stl file to be in other projects like with a 3D printer for example.

Due to time limitations, only a few objects were experimented on. More objects will need to be tested to make sure the scanner is working at full capacity.

# Conclusion

Finally, this project aimed to automate 3D scanning and it was successful in doing so. Our research provided a comprehensive review of different types of scanners, how they work, as well as their advantages and disadvantages which allowed us to design and create our machine in the most suitable way. In addition, this project offers a new design for 3D scanners that improves upon the ones seen before, both mechanically and in software. However, this is simply a prototype. Therefore, we hope to improve on our 3D scanner, especially if given the right tools to do so. Some of these improvements include: getting a bigger motor to turn a bigger turntable, make the turntable more stable by adding circular railways for the wheels, and adding stoppers for the motors to not exceed the vertical and horizontal axes. Not only that, but we hope to expand this project to one day include 3D modeling and display in the form of a 3D printer and 3D holograms respectively. We also wish to implement artificial intelligence and machine learning techniques that will work with more complex models, in areas such as recognizing and identifying models, combining different models etc.

# References

[1] (Retrieved 2023, January 12) “Faro Scanner” <https://www.google.com/search?q=Faro+laser+scanner&tbm=isch&sa=X&ved=2ahUKEwjbzprToMP8AhVE57sIHZTuDFAQ0pQJegQIChAB&biw=992&bih=617&dpr=1.5#imgrc=dAZX1lsr-0mCsM>

[2] (Retrieved 2023, January 12) “Fabscan Pi”

<https://www.google.com/search?q=Fabscan+Pi&tbm=isch&sa=X&ved=2ahUKEwilge32qMP8AhVIgP0HHcELCSoQ0pQJegQIDBAB&biw=992&bih=617&dpr=1.5#imgrc=3NKtJfC_amTqLM>

[3] (Retrieved 2023, January 12) “Fabscan Pi Project”

<https://www.google.com/search?q=Fabscan+Pi&tbm=isch&sa=X&ved=2ahUKEwilge32qMP8AhVIgP0HHcELCSoQ0pQJegQIDBAB&biw=992&bih=617&dpr=1.5#imgrc=vfbTLQ06SyMpwM>

[4] (Retrieved 2023, January 12) “Structured Light Scanners | Hexagon”

<https://www.google.com/search?q=structured+light+scanners&tbm=isch&sa=X&ved=2ahUKEwjLrcnUq8P8AhUJnf0HHYb1BIEQ0pQJegQIChAB&biw=992&bih=560&dpr=1.5#imgrc=YsrO5dS_VB8tjM>

[5] (Retrieved 2023, January 12) “DIY 3D Scanner Based on Structured Light and Stereo Vision in Python Language”

<https://www.instructables.com/DIY-3D-scanner-based-on-structured-light-and-stere/>

[6] (Retrieved 2023, January 12) “Types of 3D Scanners”

<https://www.3d-prototype.co.uk/3d-scanner/types-of-3d-scanners/>

[7] (Retrieved 2023, January 12) “Multiple Raspberry Pi 3D Scanner”

<https://www.instructables.com/Multiple-Raspberry-PI-3D-Scanner/>

[8] (Retrieved 2023, January 12) “Microsoft Kinect Sensor”

<https://www.google.com/search?q=kinect&tbm=isch&sa=X&ved=2ahUKEwi3kPT2_cb8AhVoTaQEHZq8DvQQ0pQJegQIDBAB#imgrc=1daD6wC5zG6b6M>

[9] (Retrieved 2023, April 30) “Build A DIY Desktop 3D Scanner With Infinite Resolution”

<https://www.instructables.com/Build-a-DIY-Desktop-3d-Scanner-With-Infinite-Resol/>

[10] (Retrieved 2023, June 4) “Professional Video Studio Equipment”

<https://glidegear.net/collections/studio-equipment>

[11] (Retrieved 2023, June 4) “DIY 3D Scanner – Fully 3d printed photogrammetry rig”

<https://youtu.be/CBpZtnu1Mig>

[12] (Retrieved 2023, June 4) “DIY Arduino based 3D scanning machine| IR 3D Scanner”

<https://youtu.be/diuOMDRtIMI>

# Appendix

This project is also available on github at <https://github.com/Wh20031/3D-Scanner-Repository.git>