Department of Electronic & Telecommunication Engineering,
University of Moratuwa, Sri Lanka.

# 3D Mapping Device for Object Tracking Design Methodology Report - Group B

Submitted By:

| | |
|---|---|
| 210728C | Wijewickrama W.K.D.D.D. |
| 210031H | Amarasinghe A.A.W.L.R. |


Submitted in partial fulfillment of the requirements for the module

## EN 2160 : Electronic Design Realization

20th June 2024

# Contents

# 1 Introduction

This report covers the documentation with regard to the 3D mapping device for object tracking designed in fulfillment of the Electronic Design Realization module. This documentation covers the review process, design process, conceptual designs and final designs with regard to the product.

# 2 Review Progress

## 2.1 Companies in the 3D Mapping Field

1. **Microsoft:** Microsoft has developed the Microsoft Kinect sensor, which utilizes TOF technology for depth sensing. While the Kinect was initially popular in gaming, its applications have extended to various industries, including robotics, healthcare, and augmented reality.

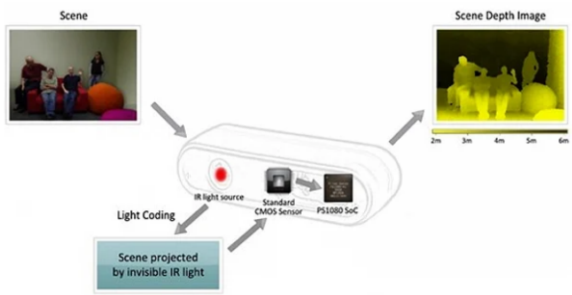


Figure 1: Kinect Sensor

2. **Intel:** Intel has been actively involved in developing TOF sensors for depth sensing applications. Their RealSense series includes TOF cameras suitable for various industrial and consumer applications, including robotics, augmented reality, and 3D scanning.

Figure 2: RealSense

3. **Velodyne:** While primarily known for their lidar sensors for autonomous vehicles, Velodyne has also been exploring TOF technology for short-range applications. They have developed sensors capable of accurate distance measurement and object detection.

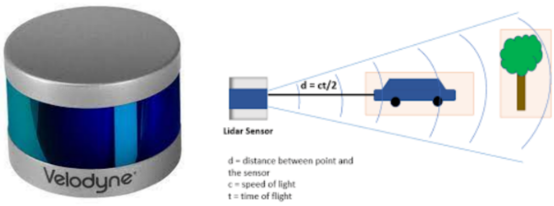


Figure 3: Velodyne

## 2.2 Researches Regarding 3D Mapping Technologies

1. **Research on 3D Digital Map System and Key Technology - Zhao Zhongyuan**
    - **Concepts and Characteristics:**
        - 3D digital maps represent a continuous 3D digital model of a city, serving as the core of 3D digital city construction.
        - They integrate various elements such as natural topography, environment, roads, buildings, and infrastructure into a seamless mosaic under a coordinate system.
        - These maps leverage technologies like GIS, RS, network, multimedia, and VR to create a multi-dimensional digital city information platform.
    - **Characteristics of 3D Digital Map Systems:**
        - They create a virtual space environment of the real city, enabling dynamic interactive browsing, spatial analysis, and simulation of large amounts of data.
        - They support geographic synergies in network environments, allowing seamless management of urban space and multi-user concurrent access.
        - They serve as a 3D space basic platform for city management, providing macro information, and detailed data for various city management activities.
    - **Key Technologies:**
        - Advanced technologies such as 3DGIS, VR, data compression, computer, and network technologies are employed.
2. **A Survey of 3D Object Detection Algorithms for Intelligent Vehicles Development**
    - **Background:**
        - Despite the significant advancements in 2D object detection, particularly in computer vision applications, there are limitations when it comes to intelligent driving.
        - The safety and reliability of self-driving cars require the ability to detect 3D models of objects in their surroundings to perceive real driving situations accurately.
    - **Development of 3D Object Detection:**
        - The paper systematically surveys the development of 3D object detection methods applied to intelligent driving technology.
        - It analyzes the shortcomings of existing 3D detection algorithms and discusses future development directions in this field, emphasizing the importance of accurate environment perception for reducing traffic accidents.
    - **Hardware Preparations for Intelligent Driving:**
        - The paper discusses the technologies demanded by intelligent driving, including sensors, high-precision maps, Internet of Vehicles (IoV), and high-performance chips.
        - It provides an overview of the development levels of autonomous driving, highlighting the progression from Level 0 (fully manual) to Level 5 (fully autonomous).

3. **2D and 3D Object Detection Algorithms from Images: A Survey**

- **Object Detection Overview:**
  - Object detection, a crucial branch of computer vision, locates and classifies objects in images, encompassing tasks such as object recognition and localization.
  - Traditional object detection algorithms relied on hand-designed methods for feature extraction and suffered from limitations such as slow speed, low accuracy, and high computational overhead.
- **Transition to Deep Learning:**
  - Deep convolutional neural networks (CNNs) revolutionized object detection by efficiently extracting features from images, leading to significant improvements in speed and accuracy.
  - The rise of CNN-based models, such as AlexNet, and Swin Transformer, replaced traditional algorithms and became the mainstream in object detection research.
- **Recent Advances:**
  - Recent advancements include the adoption of Transformer models, such as ViT, in object detection, offering dynamic parameter learning mechanisms and improved adaptability to large datasets.
  - These developments have sparked research into incorporating Transformers into object detection frameworks as backbone or neck components, further enhancing performance.
- **Transition to 3D Object Detection:**
  - While 2D object detection focuses on regressing 2D bounding boxes, the practical needs of real-world applications require 3D object detection to predict detailed information about an object's 3D size, coordinates, speed, and orientation.
  - However, transitioning from 2D to 3D object detection presents challenges such as the lack of geometric constraints, and handling multidimensional data.
- **Importance of 3D Object Detection:**
  - Despite these challenges, 3D object detection algorithms have gained importance in both industrial and academic communities due to their potential applications in unmanned vehicles, smart robots, and real-time traffic monitoring.

## 2.3 YouTube Videos Regarding 3D Mapping Technologies

1. **3D Time-of-Flight (ToF) camera for accurate 3D depth imaging — e-con Systems**
    - **Link:** https://www.youtube.com/watch?v=8JIBTZCXkHw
    - **Description:** DepthVista is a 3D Time-of-Flight (ToF) camera designed by e-con Systems for accurate 3D depth imaging to help Autonomous Mobile Robots (AMRs) and Automated Guided Vehicles (AGVs) with safer navigation.
    - **Features:**
        - Streams 3D depth data & RGB data in single frame: Enables both obstacle detection & object recognition/identification with one camera.
        - Uses VCSEL of 850nm: Safer for human eyes and can operate even in absolute darkness.
        - High speed & high resolution 640×480 @ 30 fps: Allows AMRs and AGVs to perceive their environment and plan their paths optimally, and safely complete their tasks.
        - Onboard depth image processing: Provides ready-to-use depth data from the camera itself thereby reducing the computational load on the host.
    - **Target Applications:**
        - Autonomous Mobile Robots (AMRs)
        - Autonomous Guided Vehicles (AGVs)
        - People counting in retail analytics
        - Patient care / Patient Monitoring
        - 3D Face recognition for anti-spoofing
        - Robotic arms
2. **Distance linear image sensor / Object detection [TOF]**
    - **Link:** `https://www.youtube.com/watch?v=Y0P-fXSTbzQ`
    - **Description:** The distance image sensors are designed to measure the distance to an object using the TOF method. This video demonstrates distance measurement to a target object ahead (human) using a unit equipped with a distance linear image sensor and light source. The output waveform, matched to the target object's position and movement, is shown.
3. **Visionary-T Mini: 3D ToF camera solving industrial applications**
    - **Link:** https://www.youtube.com/watch?v=n6u2HQ5R2w8
    - **Description:** The Visionary-T Mini ensures more efficiency in industrial settings wherever reliable 3D depth values for dynamic processes are needed.
    - **Fields of Applications:**
        - Object detection
        - Navigation
        - Palletizing and depalletizing
        - Measurement and volume detection
        - Positioning
        - Gesture control
        - Area monitoring

# 3 Identification of Stake Holders

- The stakeholder map diagram delineates various stakeholders involved in a project, categorized based on their levels of interest and influence. Through a quadrant-based approach, stakeholders are strategically positioned according to their degree of engagement and impact on project outcomes.
- This systematic classification facilitates stakeholder management by enabling the anticipation of stakeholders' needs, ensuring thorough engagement with key stakeholders, and maintaining regular communication channels with those deemed most critical.
- By aligning stakeholder engagement strategies with their respective levels of interest and influence, project stakeholders can be effectively managed to optimize project outcomes.

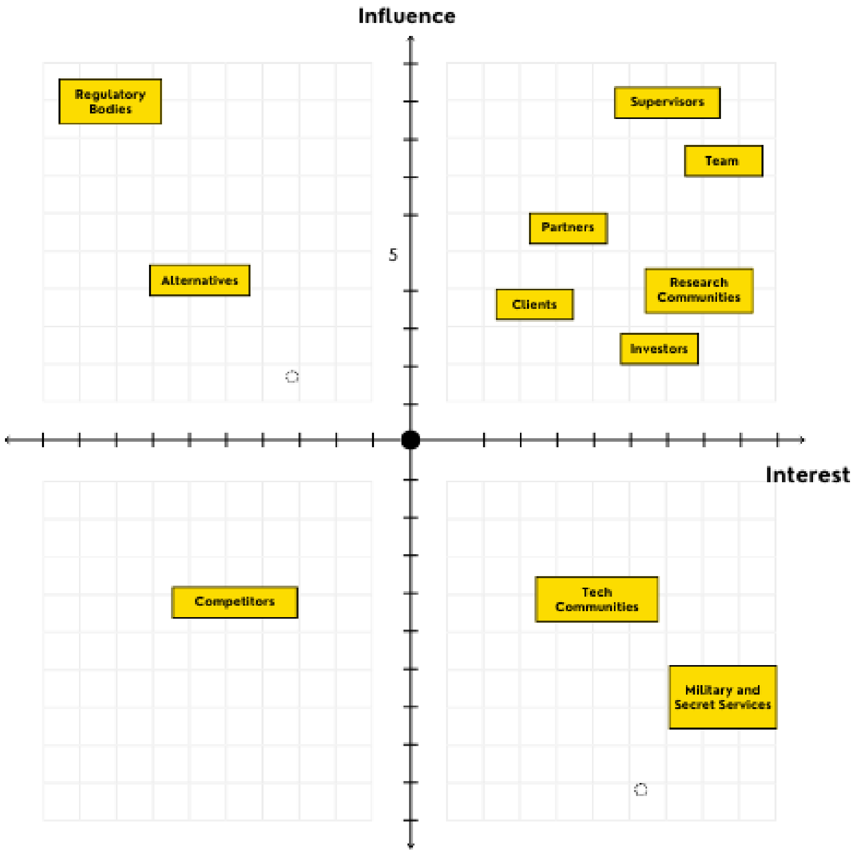


Figure 4: Stakeholder Map

# 4 Observe Users

## 4.1 Purpose

The observation of users is crucial for understanding how individuals interact with the 3D surround scanner prototype. By observing users during the scanning process, valuable insights can be gained regarding usability, effectiveness, and potential areas for improvement.

## 4.2 Methodology

1. **Participant Selection**: A diverse group of participants from various backgrounds and skill levels should be selected to ensure a comprehensive understanding of user interactions.

2. **Observation Setup**: Participants will be invited to interact with the 3D surround scanner prototype in a controlled environment. The setup will include clear instructions on how to operate the device.

3. **Data Collection**: Observations will be conducted in real-time as participants engage with the prototype. This may involve video recording, note-taking, or both to capture detailed interactions and reactions.

4. **Interviews (Optional)**: Following the observation session, participants may be interviewed to gather additional feedback and insights into their experience using the scanner.

## 4.3 Key Observations

1. **User Interaction**: Observations will focus on how users interact with the scanning interface, including their ability to understand and navigate the controls.

2. **Efficiency and Accuracy**: The efficiency and accuracy of the scanning process will be assessed, noting any challenges or errors encountered by users.

3. **User Feedback**: Participants' verbal and non-verbal feedback will be recorded, highlighting any likes, dislikes, or suggestions for improvement.

4. **Overall Experience**: Observations will aim to capture the overall user experience, including aspects such as ease of use, satisfaction, and perceived usefulness of the scanner.

## 4.4 Analysis

1. **Identify Patterns**: Data collected from observations will be analyzed to identify recurring patterns, trends, or issues observed across multiple participants.

2. **User Insights**: Insights gleaned from user observations will be used to inform iterative improvements to the scanner prototype, focusing on enhancing user experience and addressing usability challenges.

3. **Recommendations**: Based on the analysis, recommendations will be made for refinements or adjustments to the design, interface, or functionality of the 3D surround scanner to better meet user needs and preferences.

### 4.5 User Requirements

1. **Accuracy and Precision:** Users will likely expect the device to accurately map the 3D space with high precision, capturing details and dimensions reliably.
2. **Real-time Mapping:** Depending on the application, users may require real-time mapping capabilities to see the 3D environment being mapped as they move through it.
3. **Portability:** For users who need to map various locations or environments, portability may be important. A device that is lightweight, compact, and easy to transport would be desirable.
4. **Ease of Use:** The device should have a user-friendly interface that is intuitive to operate, allowing users to start mapping with minimal training or expertise.
5. **Compatibility:** Compatibility with different operating systems or platforms may be necessary to ensure the device can be used with various devices or integrated into existing systems.
6. **Range and Coverage:** Users may have specific requirements for the range and coverage of the mapping device, depending on the size and complexity of the environments they need to map.
7. **Data Output:** Users will likely need the device to provide output in a usable format, such as 3D models, point clouds, or maps that can be easily integrated into other software or analyzed further.
8. **Environmental Adaptability:** The device should be able to map different types of environments, including indoor and outdoor spaces, and adapt to various lighting conditions and surface types.
9. **Security and Privacy:** Depending on the application, users may have concerns about the security and privacy of the data collected by the mapping device, so measures to protect sensitive information may be required.

# 5 Need List

In order to successfully implement the 3D surround scanner using TOF sensors and stepper motors, the following components and resources are required:

## 5.1 Hardware

- Two Time-of-Flight (TOF) sensors with appropriate specifications for distance measurement and accuracy.
- Two stepper motors capable of rotating the TOF sensors independently.
- Microcontroller or development board to control the stepper motors and interface with the TOF sensors.
- Power supply unit to provide adequate power to the system components.
- Mounting hardware and mechanical components for assembling the scanner.
- Computer or embedded system for processing and visualization of the captured 3D point cloud data.

## 5.2 Software

- Firmware development environment for programming the microcontroller or development board.
- Software libraries or drivers for interfacing with the TOF sensors and stepper motors.
- Programming environment for developing the 3D point cloud generation algorithm.
- Visualization software or libraries for rendering and viewing the captured 3D point cloud data.

## 5.3 Documentation and Support

- Datasheets and technical documentation for the TOF sensors, stepper motors, and other hardware components.
- Tutorials or guides on using the chosen microcontroller or development board.
- Community forums or online resources for troubleshooting and seeking assistance during the project implementation.

# 6 Stimulate Ideas

In the development of our 3D mapping device, stimulating innovative ideas is critical to enhancing the system's functionality, usability, and adaptability. We used several approaches and considerations to foster creativity and innovation throughout the project.

1. **Collaborative Brainstorming Sessions**
   Organizing regular brainstorming sessions with team members of the large group who are from diverse backgrounds including electronics, software engineering, and mechanical design. This multidisciplinary approach leads to unique solutions and perspectives, fostering a more comprehensive and innovative design.

2. **User-Centric Design Approach**
   Engaging with potential users of the 3D mapping device to gather insights into their specific needs and challenges. This feedback can inspire features and improvements that make the device more user-friendly and effective in real-world applications. Conducting surveys, interviews, and usability testing sessions is also helpful.

3. **Exploration of Advanced Technologies**
   Investigating the integration of advanced technologies such as machine learning algorithms for more intelligent data processing, AI-driven error correction, or augmented reality for real-time visualization of the captured 3D point cloud data. Keeping abreast of the latest technological advancements can provide a competitive edge and open new avenues for innovation.

4. **Prototype Iteration and Testing**
   Developing multiple prototypes with varying designs and functionalities. Testing these prototypes under different conditions will help identify the strengths and weaknesses of each approach, leading to more refined and innovative final designs. Embracing a fail-fast mentality to quickly learn from mistakes and pivot to better solutions.

5. **Inspirational Research**
   Conducting comprehensive research on existing 3D mapping technologies and related fields. Analyzing case studies, white papers, and patents to understand how other solutions have addressed similar challenges. This research can spark new ideas and inspire improvements or novel approaches to our design.

6. **Creative Use of Materials and Components**
   Experimenting with different materials and components that could offer better performance, cost-effectiveness, or ease of assembly. For example, exploring lightweight yet durable materials for the mounting hardware, or alternative power supply options that enhance portability and efficiency.

7. **Hackathons and Innovation Challenges**
   Participating in hackathons and innovation challenges focused on 3D mapping and related technologies. These events can provide a platform for rapid ideation and prototyping, fostering a competitive yet collaborative environment that stimulates creative thinking.

8. **Cross-Industry Insights**
   Looking beyond the immediate field of 3D mapping and exploring how other industries tackle similar problems. For instance, techniques from robotics, aerospace, or even biomedical engineering could offer valuable insights and innovative solutions applicable to our device.

9. **Open Source and Community Engagement**
   Leveraging open-source projects and engaging with online communities focused on 3D mapping and sensor technologies. Contributing to and learning from these communities can provide new ideas, collaborative opportunities, and shared resources that enhance the project's development.

10. **Scenario Planning and Future Trends**
    Considering future trends and potential scenarios in which the 3D mapping device might be used. This forward-thinking approach can help identify long-term opportunities and guide the development of features that ensure the device remains relevant and innovative in the years to come.

By actively pursuing these strategies, we were able to stimulate a continuous flow of ideas and innovations, ultimately leading to a more robust, versatile, and advanced 3D mapping device. This proactive approach to idea generation and development is essential for staying ahead in the rapidly evolving field of 3D mapping and object tracking.

# 7 Conceptual Designs

## 7.1 Design 1

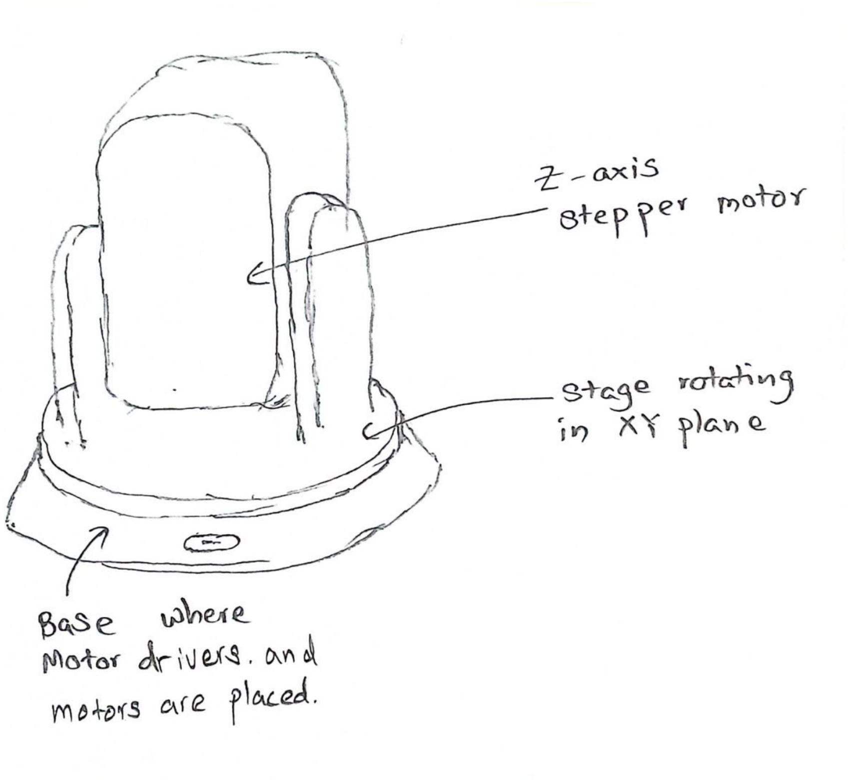


Figure 5: Design 1: Stage and 2 Steppers

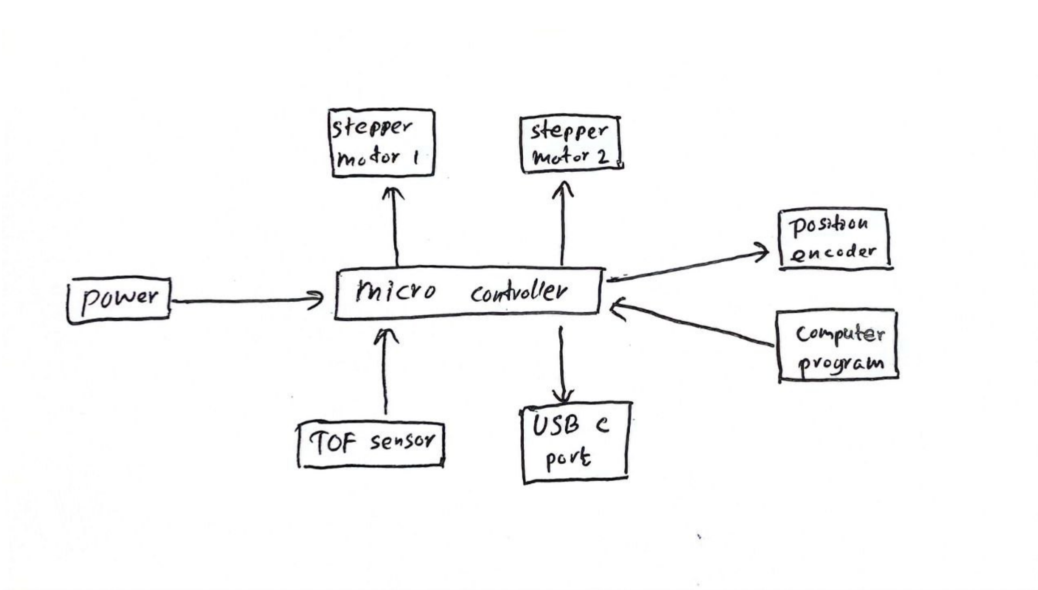


Figure 6: Block Diagram 1

#### 7.1.1 Overview

This innovative design merges the Time of Flight (ToF) sensor concept with the precision control of stepper motors to create a dynamic two Degrees of Freedom (2-DoF) rotational system. It incorporates two stepper motors, with one strategically positioned to govern movement along the xy plane within the base, while the other orchestrates stage rotation using z-axis angles obtained from the ToF sensor. This arrangement ensures meticulous measurement of $r$, $\theta$, and $\alpha$ values for each reading, thereby enhancing system accuracy.

#### 7.1.2 Operation

- Integration of ToF sensor technology with stepper motors for precise control.
- Two stepper motors utilized: one for xy plane movement and another for stage rotation.
- ToF sensor provides z-axis angles for enhanced control and accuracy.
- Enables meticulous measurement of $r$, $\theta$, and $\alpha$ values for each reading.
- Data transmission facilitated through USB Type-C port, ensuring seamless connectivity and efficient data transfer.
- Advanced processing techniques applied on the computer to convert data into a comprehensive point cloud representation.
- Sophisticated image detection algorithms employed for in-depth analysis and interpretation.

#### 7.1.3 Advantages

- This design offers precise control over rotational movements.
- Streamlines data processing and analysis, making it a versatile solution for diverse applications requiring precise spatial measurements and analysis.

#### 7.1.4 Challenges

However, there are some challenges associated with this design:

- Mechanical Complexity: Incorporating two stepper motors and a ToF sensor system adds mechanical complexity, requiring careful calibration and maintenance to ensure reliable operation.
- Computational Requirements: Advanced processing techniques and image detection algorithms may demand significant computational power, posing challenges in terms of hardware requirements and computational efficiency.

## 7.2 Design 2

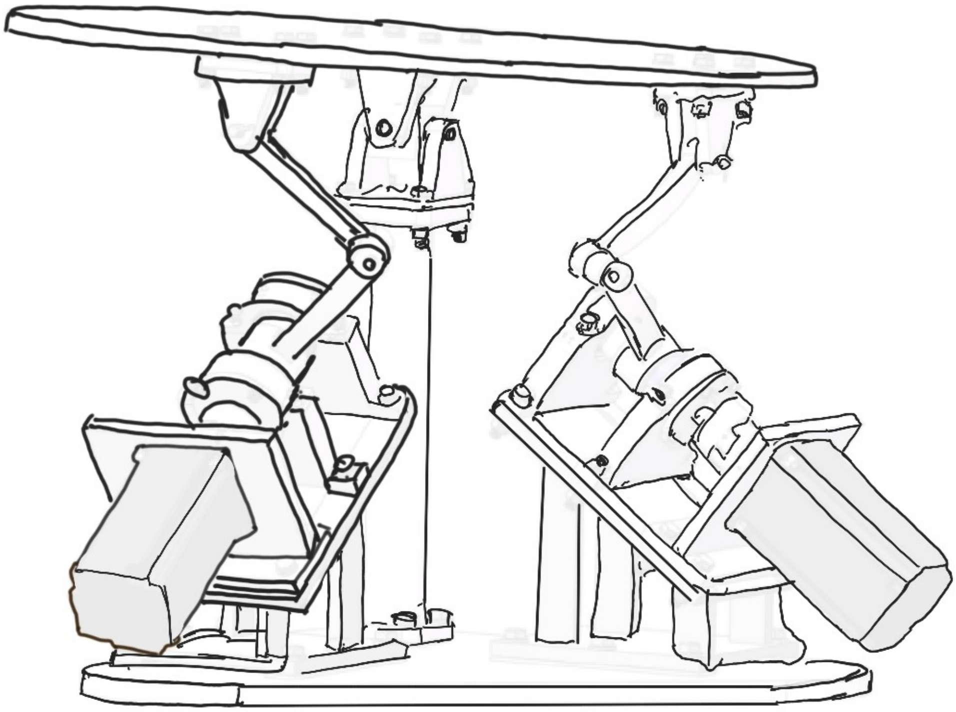

Figure 7: Design 2: Servo Motors and Liver System

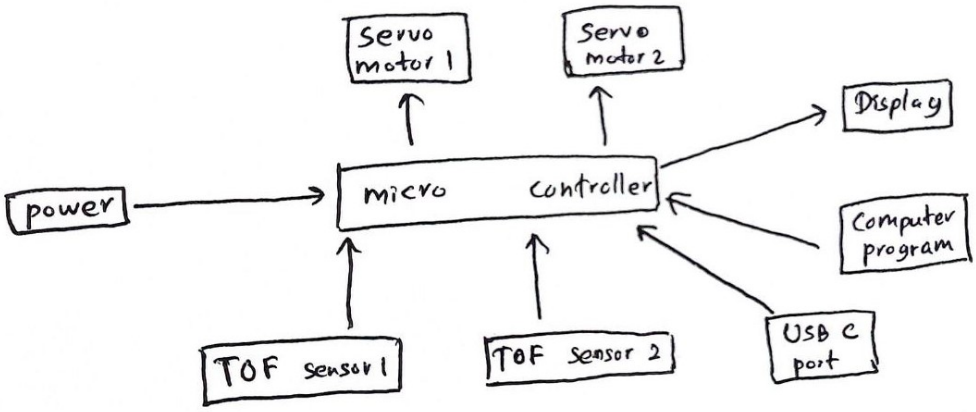


Figure 8: Block Diagram 2

#### 7.2.1 Overview:

Design 1 utilizes a basement housing two time-of-flight (ToF) sensors, each equipped with servo motors for precise adjustment. The servo motors enable the rotation and tilting of the basement to capture surrounding data from various angles.

**Components:**

1. **Basement Housing:** The basement serves as the platform for mounting the ToF sensors and servo motors. It is designed to be adjustable in both rotation and tilt angles.

2. **ToF Sensors:** Two ToF sensors are mounted on the basement to capture depth information. These sensors emit infrared light pulses and measure the time taken for the pulses to reflect back, allowing for accurate distance calculations.

3. **Servo Motors:** The servo motors are responsible for adjusting the orientation of the basement. They provide precise control over the rotation and tilt angles, allowing for comprehensive coverage of the surrounding environment.

#### 7.2.2 Operation:

- **Rotation Control:** One servo motor is dedicated to controlling the rotation of the basement. By rotating the basement, the sensors can scan the surroundings horizontally, providing a panoramic view of the scene.

- **Tilt Control:** The second servo motor adjusts the tilt angle of the basement. This enables the sensors to capture data from different elevations, facilitating a more detailed representation of the environment.

#### 7.2.3 Advantages:

- **Precise Control:** The servo motors offer precise adjustment capabilities, allowing for fine-tuning of the sensor positions to capture accurate data.

- **Comprehensive Coverage:** By rotating and tilting the basement, the design ensures thorough coverage of the surrounding environment, resulting in more comprehensive 3D scans.

#### 7.2.4 Challenges:

- **Mechanical Complexity:** The incorporation of servo motors adds mechanical complexity to the design, requiring careful calibration and maintenance to ensure reliable operation.

- **Power Consumption:** Servo motors may consume significant power, which could impact the overall energy efficiency of the system.

## 7.3 Design 3

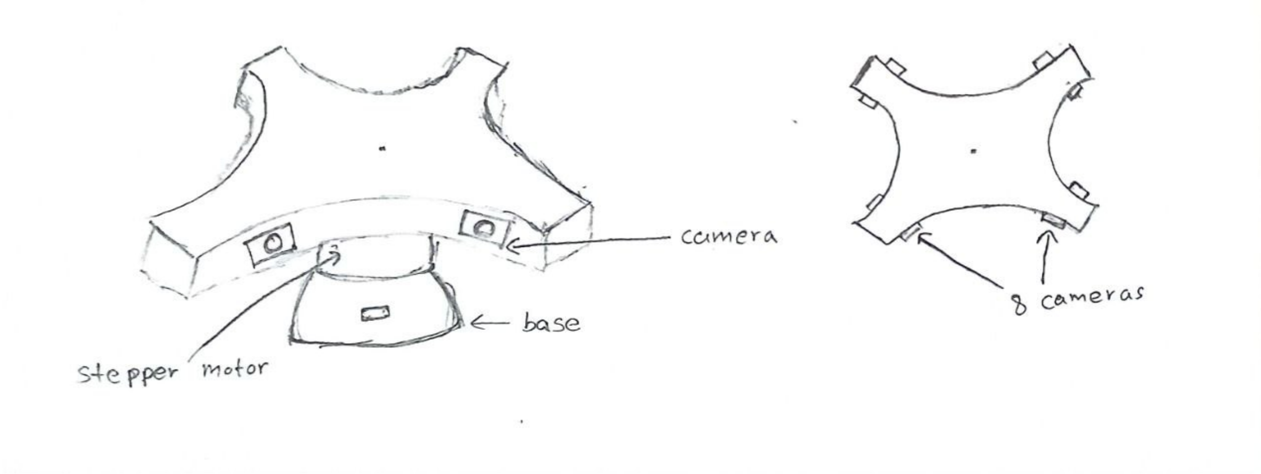


Figure 9: Design 3

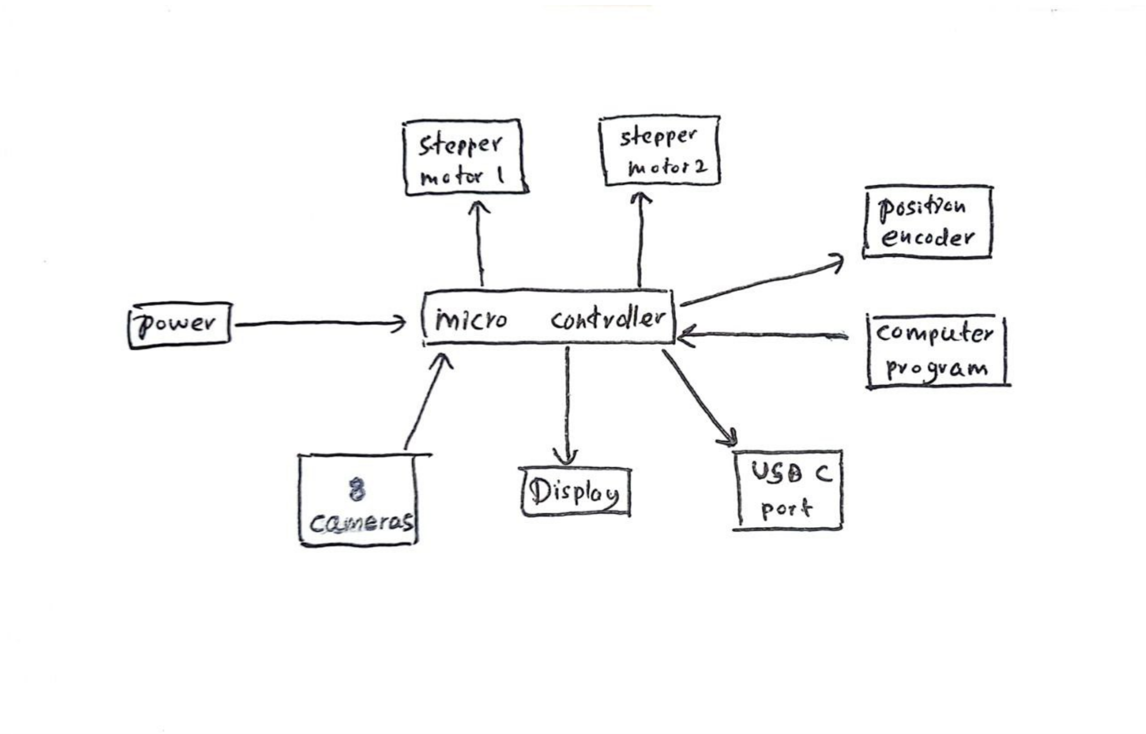


Figure 10: Block Diagram 3

#### 7.3.1 Overview

This cutting-edge design leverages image processing technology, employing a sophisticated array of eight cameras strategically positioned to capture comprehensive visual data. Among these cameras, a subset is dedicated to recording videos from one side, utilizing precise angles and calibration to facilitate the generation of detailed 3D objects through advanced image processing concepts. The collective data captured by the eight cameras is then meticulously processed through Convolutional Neural Networks (CNNs), enabling the generation of accurate 3D mappings and point clouds.

- Utilization of eight cameras for comprehensive visual data capture.
- Subset of cameras dedicated to video recording from one side.
- Angled and calibrated cameras facilitate detailed 3D object generation.
- Data from the camera array processed through Convolutional Neural Networks (CNNs).
- CNNs enable accurate generation of 3D mappings and point clouds.

#### 7.3.2 Challenges

However, a notable challenge lies in the associated costs and computational requirements. The complexity of processing data from multiple cameras in real-time demands significant computational power, which may pose challenges in terms of both hardware affordability and computational efficiency.

#### 7.3.3 Advantages

The advantages of this design include:

- High-Quality Data: The use of eight cameras allows for comprehensive visual data capture, resulting in high-quality 3D reconstructions.
- Detailed 3D Objects: The subset of cameras dedicated to video recording from one side, along with precise angles and calibration, facilitates the generation of detailed 3D objects.
- Accurate Processing: The utilization of Convolutional Neural Networks (CNNs) ensures accurate processing of the captured data, leading to precise 3D mappings and point clouds.

#### 7.3.4 Significance

This design represents a powerful fusion of cutting-edge imaging technology and advanced computational algorithms, promising high-fidelity 3D reconstructions and data analysis. Despite the challenges posed by cost and computational demands, the potential applications of such a system in various fields such as robotics, augmented reality, and medical imaging underscore its significance and potential impact.

# 8 Evaluation of Conceptual Designs

## 8.1 Comparison of the Designs

Table 1: Comparison of Designs

| Criteria | Design 1 | Design 2 | Design 3 |
|---|---|---|---|
| **Features** | | | |
| TOF Sensor | ✓ | ✓ | |
| Cameras | | | ✓ |
| Servo Motors | | ✓ | |
| Stepper Motors | ✓ | | ✓ |
| Display | | ✓ | ✓ |
| Position Encoder | ✓ | | ✓ |
| **Enclosure Design Criteria Comparison** | | | |
| Functionality | 8 | 7 | 8 |
| Aesthetics | 9 | 8 | 6 |
| Heat Dissipation | 9 | 8 | 9 |
| Assembly and Serviceability | 8 | 7 | 6 |
| Ergonomics | 9 | 9 | 7 |
| Simplicity | 9 | 5 | 7 |
| Durability | 9 | 7 | 6 |
| **Functional Block Design Criteria Comparison** | | | |
| Functionality | 8 | 7 | 8 |
| User Experience | 9 | 8 | 7 |
| Manufacturing Feasibility | 9 | 6 | 9 |
| Cost Efficiency | 8 | 6 | 9 |
| Performance | 8 | 7 | 8 |
| Future Proofing | 9 | 8 | 9 |
| Power | 8 | 7 | 6 |
| **Total** | 120 | 100 | 105 |

## 8.2 Evaluation Criteria

### 8.2.1 Enclosure Design Criteria:

1. **Functionality:** How well the design supports the main functionalities?
2. **Aesthetics:** How eye-catching is the design and what is the overall appeal to the user?
3. **Heat Dissipation:** How much heat is generated and how well has it been managed?
4. **Assembly and Serviceability:** How easily can the assembly and disassembly be done?
5. **Ergonomics:** How well does the design fit in the user's hand and allow easy interaction?
6. **Durability:** How well does the design withstand impacts and environmental conditions?
7. **Simplicity:** How simple is the design?

### 8.2.2 Functional Block Diagram Criteria:

1. **Functionality:** How well does the circuit design meet functional requirements?
2. **User Experience:** How intuitive and user-friendly is the interaction?
3. **Manufacturing Feasibility:** Evaluate the feasibility of manufacturing the design.
4. **Cost:** Evaluate the overall cost-effectiveness for the provided functionality.
5. **Performance:** Evaluate signal quality, resolution, and bandwidth range.
6. **Future Proofing:** To what extent does the design allow for easy replacement or upgrade of individual components?
7. **Power Efficiency:** How effectively does the device manage power consumption?

## 8.3 Selected Design

After careful consideration of different factors, we selected Design 1 as the best design.

### 8.3.1 Reasons for Selection

- Integration of Time of Flight (ToF) sensor technology with precision-controlled stepper motors.
- Strategic positioning of two stepper motors ensures meticulous measurement of spatial coordinates, enhancing accuracy.
- Real-time mapping capabilities enable capturing dynamic environments with precision as users move through them.
- Utilization of USB Type-C port for efficient data transmission.
- Implementation of advanced processing techniques for data conversion and analysis.
- Despite mechanical complexity, Design 1 offers comprehensive spatial mapping, ease of use, and potential for diverse applications.

# 9 Schematic Design

The following is the schematic of the product. The schematic was designed after a thorough analysis of existing designs. A hierarchical design has been used.

## 9.1 Main

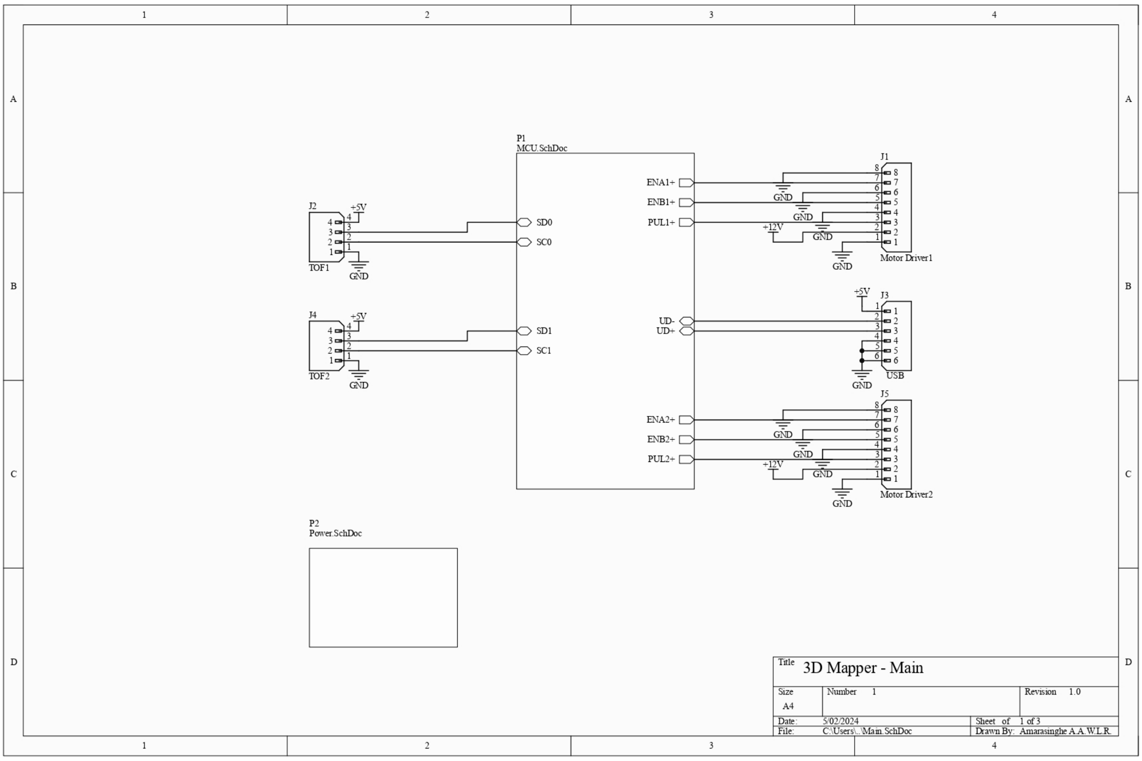


Figure 11: Main

## 9.2 MCU

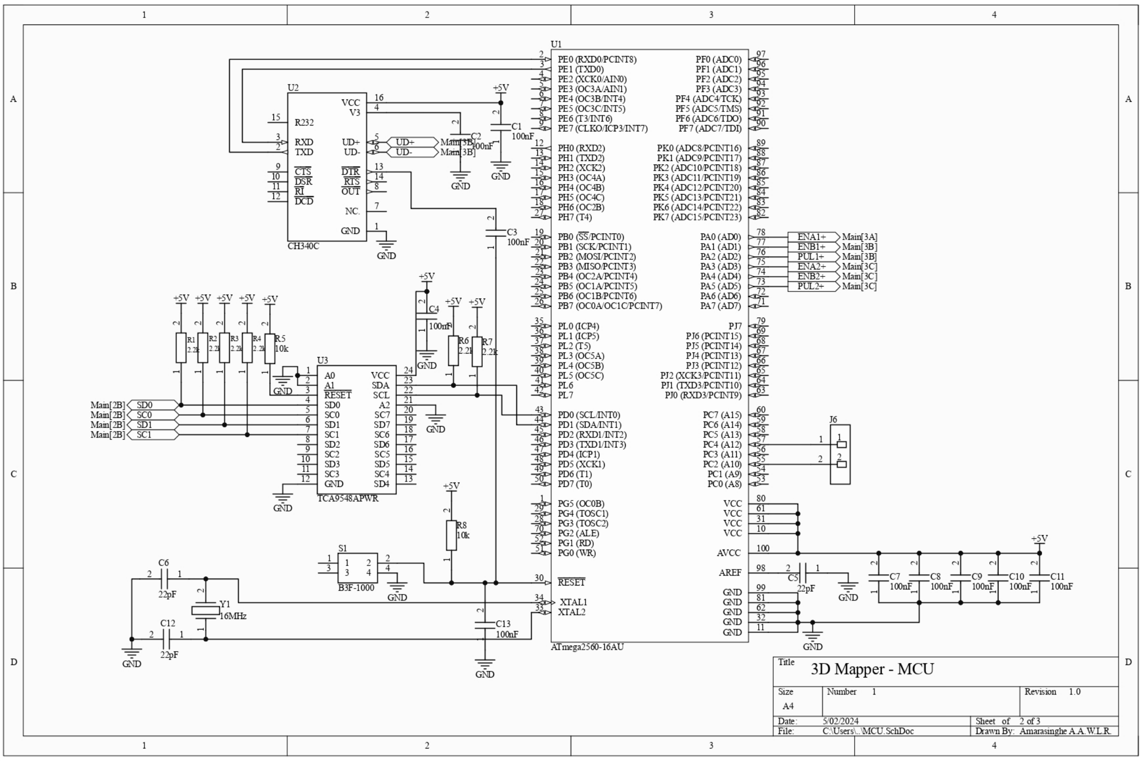


Figure 12: MCU

## 9.3 Power

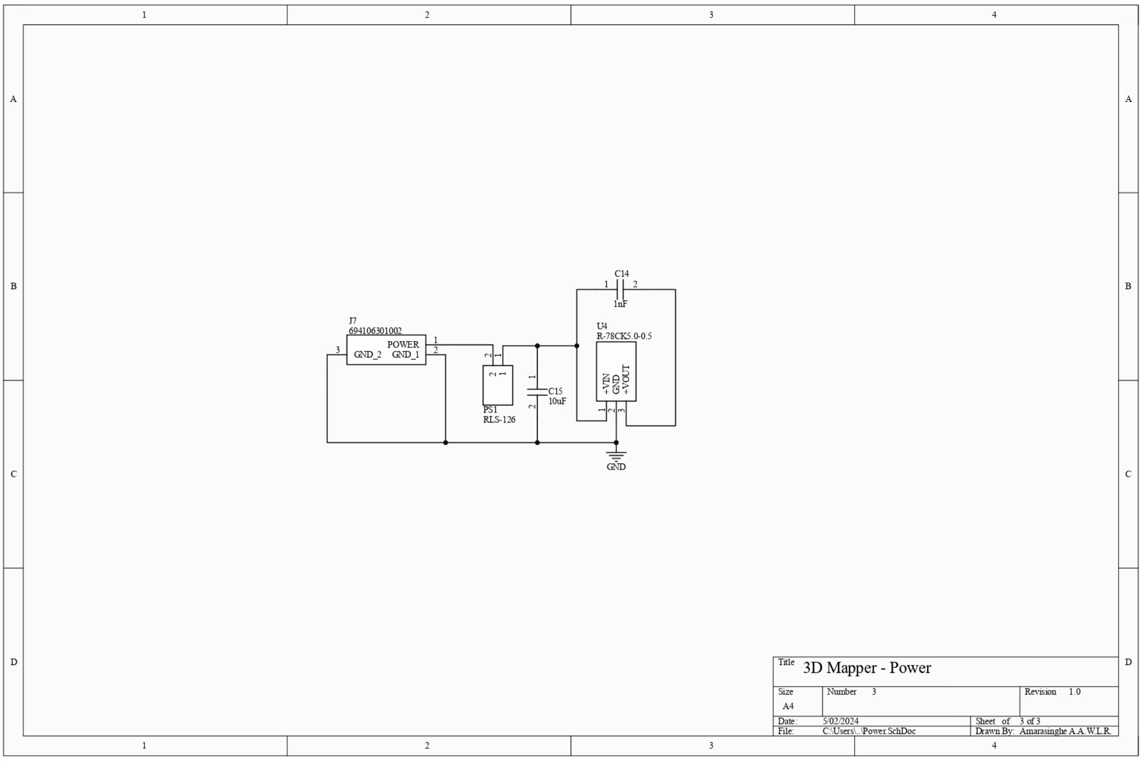


Figure 13: Power

## 9.4 Bill of Materials

| Comment | Description | Designator | Footprint | LibRef | Quantity |
|---|---|---|---|---|---|
| VJ0805Y104JXXPW1BC | Capacitor | C1, C2, C3, C4, C7, C8, C9, C10, C11, C13 | CAPC2012X90N | VJ0805Y104JXXPW1BC | 10 |
| 0805ZA220JAT2A | Capacitor | C5, C6, C12, C15 | CAPC2012X94N | 0805ZA220JAT2A | 4 |
| C1206C102GBGACTU | Capacitor | C14 | C1206 | C1206C102GBGACTU | 1 |
| B8B-XH-A | Connector Header Through Hole 8 position 0.098 (2.50mm) | J1, J5 | JST_B8B-XH-A | B8B-XH-A | 2 |
| B4B-EH-A | Connector Header Through Hole 4 position 0.098 (2.50mm) | J2, J4 | JST_B4B-EH-A | B4B-EH-A | 2 |
| B6B-XH-A(LF)(SN) | Connector Header Through Hole 6 position 0.098 (2.50mm) | J3 | JST_B6B-XH-A(LF)(SN) | B6B-XH-A(LF)(SN) | 1 |
| B2B-XH-AM(LF)(SN) | CONN HEADER VERT 2POS 2.5MM | J6 | FP-B2B-XH-AM_LF_SN-MFG | CMP-17439-000037-1 | 1 |
| 694106301002 | Connector | J7 | 694106301002_1 | 694106301002 | 1 |
| RLS-126 | Power Supply | PS1 | RLS126 | RLS-126 | 1 |
| ERA-6APB222V | Resistor | R1, R2, R3, R4, R6, R7 | ERA6AEB1020V | ERA-6APB222V | 6 |
| CMP0805AFX-1002ELF | Resistor | R5, R8 | RESC2012X60N | CMP0805AFX-1002ELF | 2 |
| B3F-1000 | Switch | S1 | B3F1002 | B3F-1000 | 1 |
| ATmega2560-16AU | 8-bit AVR Microcontroller, 4.5-5.5V, 16MHz, 256KB Flash, 4KB EEPROM, 8KB SRAM, 86 GPIO pins, 100-pin TQFP, Industrial Grade (-40°C to 85°C), Pb-Free | U1 | 100A_M | CMP-0095-00210-2 | 1 |
| CH340C | USB to serial chip CH340 | U2 | SOIC127P600X180-16N | CH340C | 1 |
| TCA9548APWR | Integrated Circuit | U3 | SOP65P640X120-24N | TCA9548APWR | 1 |
| R-78CK5.0-0.5 | Power Supply | U4 | R78CK5005 | R-78CK5.0-0.5 | 1 |
| LFXTAL027946Reel | Crystal or Oscillator | Y1 | LFXTAL027946Reel | LFXTAL027946Reel | 1 |

Figure 14: BOM

## 10 PCB Design

The following is the printed circuit board of the product. This was manufactured by JLC PCB in China. Component placement and routing were done by us.

### 10.1 PCB

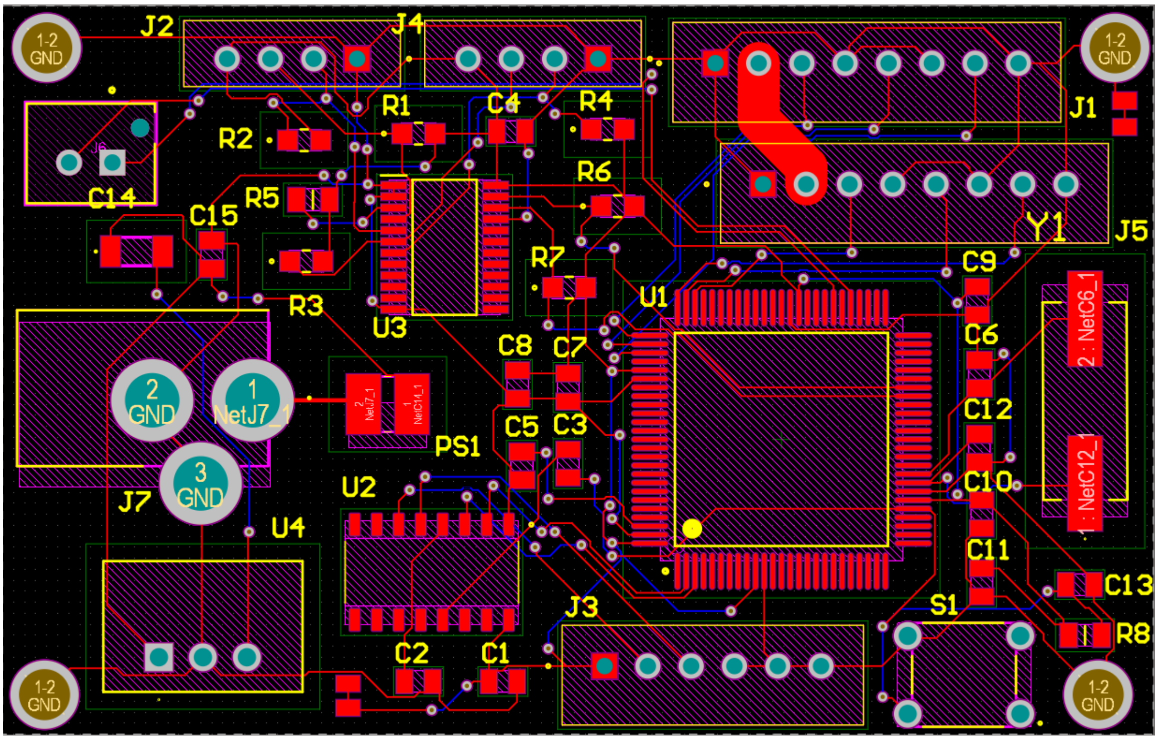


Figure 15: PCB

- PCB size= 1650mil * 2615mil ( 4.20cm * 6.64cm )

## 10.2 Top Layer

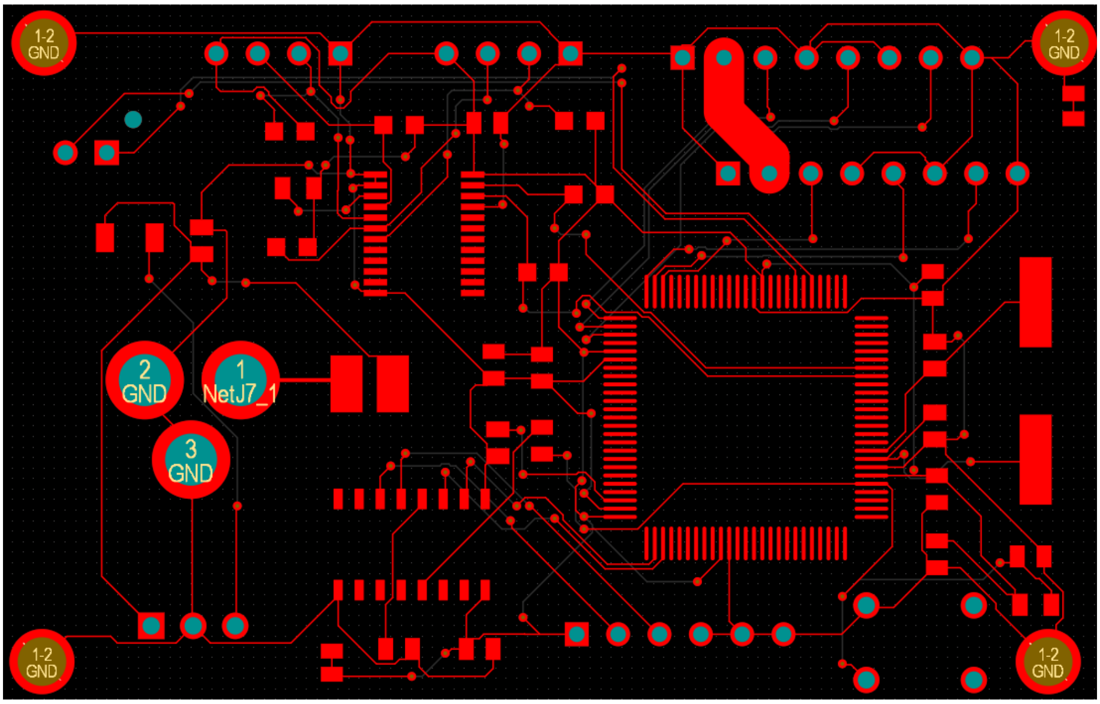


Figure 16: Top Layer

## 10.3 Bottom Layer

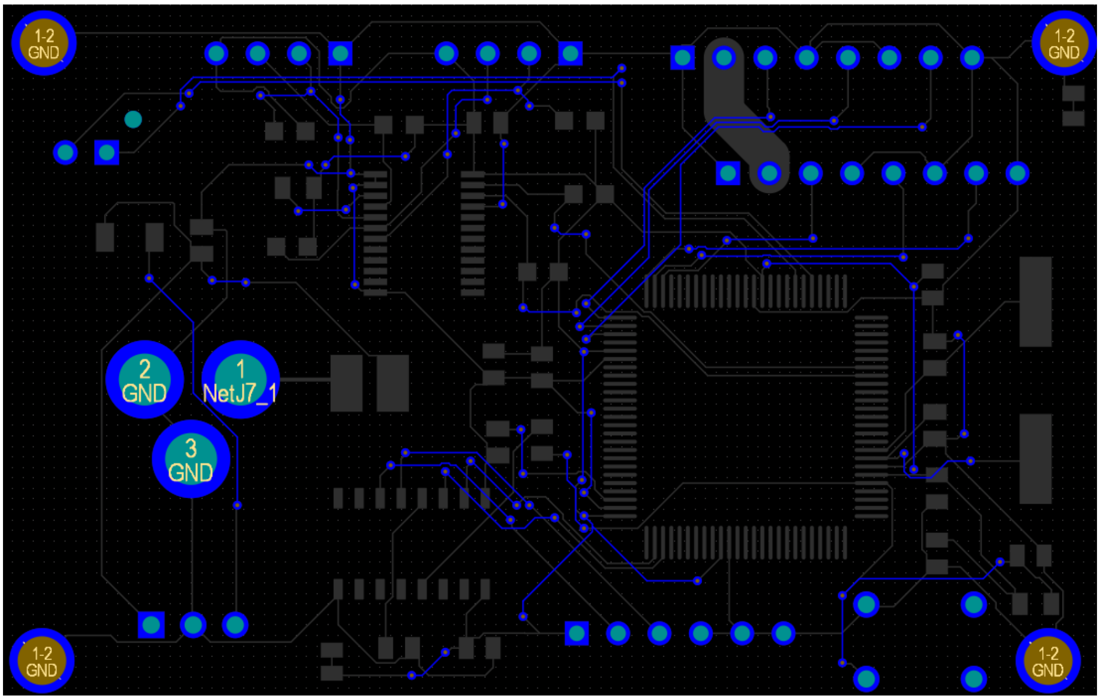


Figure 17: Bottom Layer

### 10.4 Overlay Layer

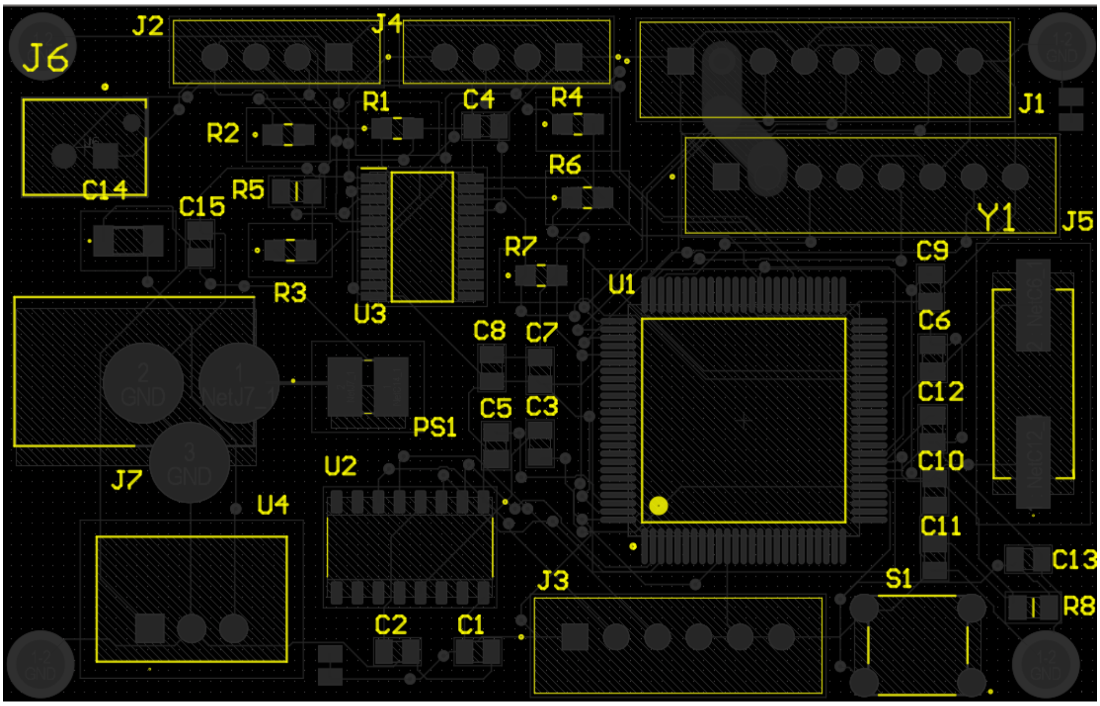


Figure 18: Overlay Layer

### 10.5 3D View

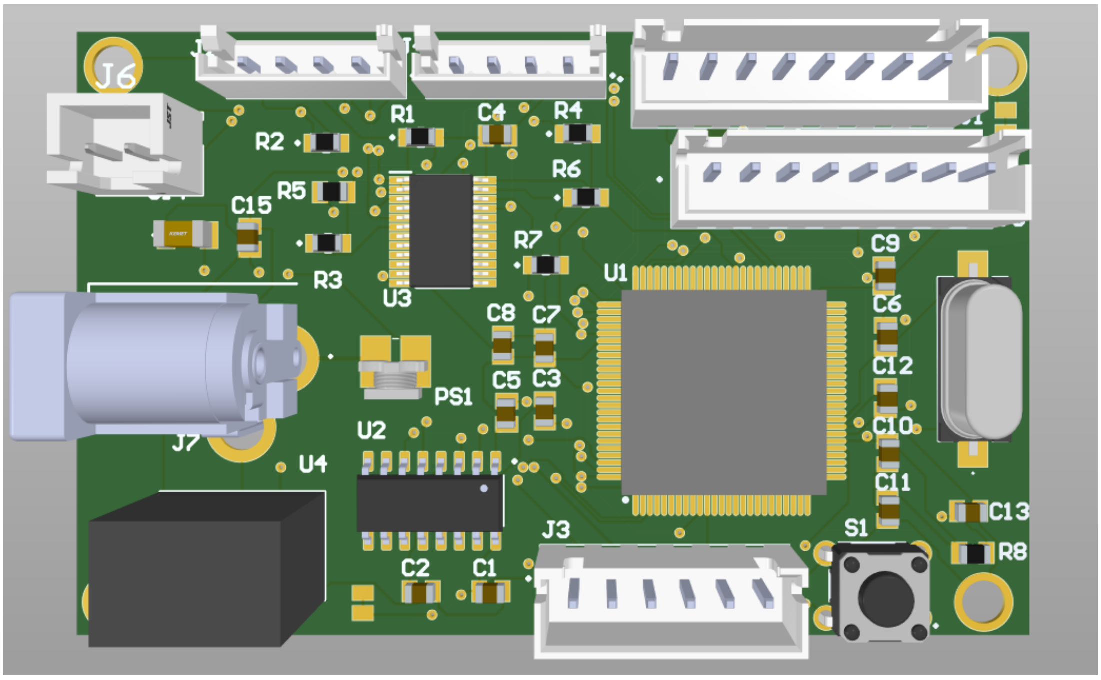


Figure 19: 3D View

## 10.6 Drill Drawings

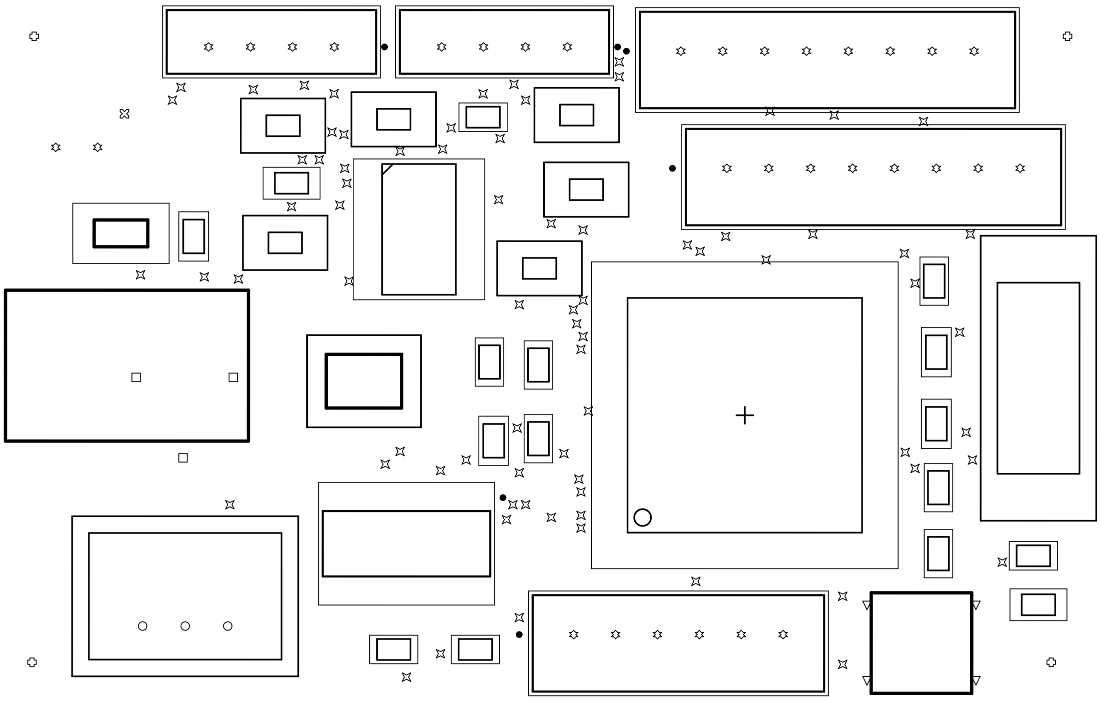

Figure 20: Drill Drawings

# 11 Solidworks Design

## 11.1 Bottom Part

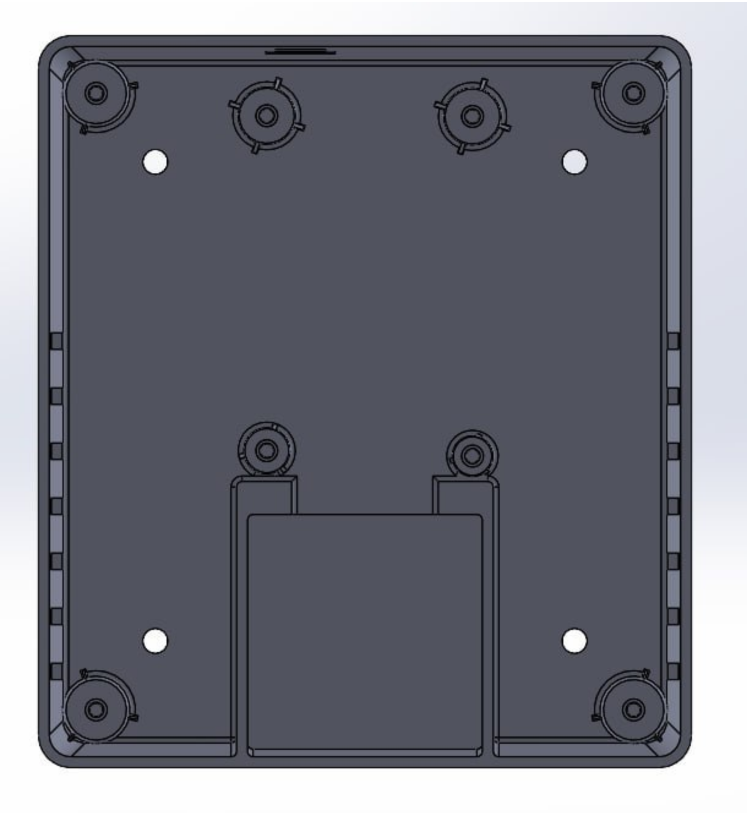

Figure 21: top view

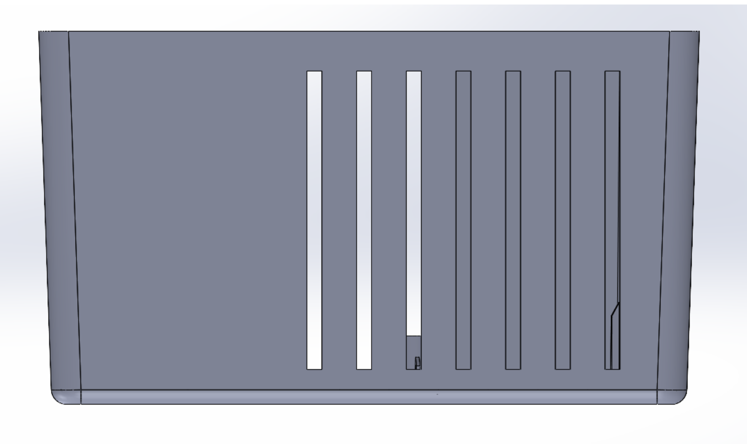

Figure 22: side view

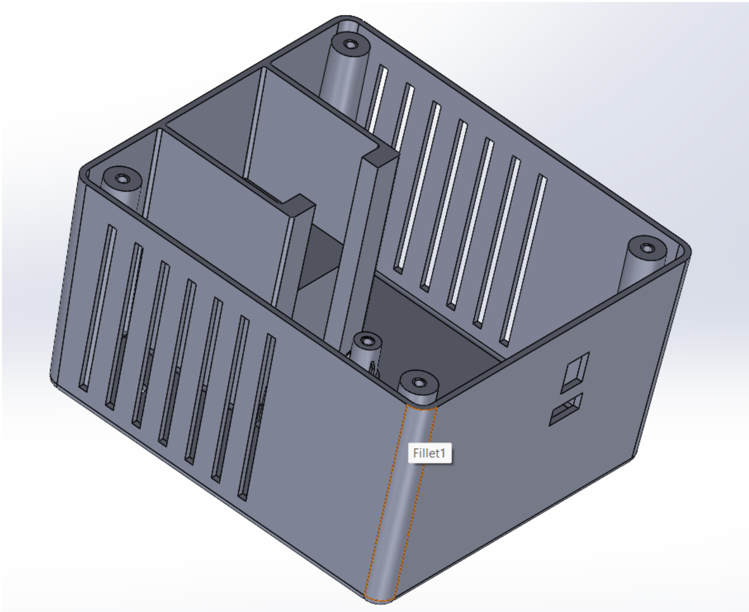


Figure 23: isometric view

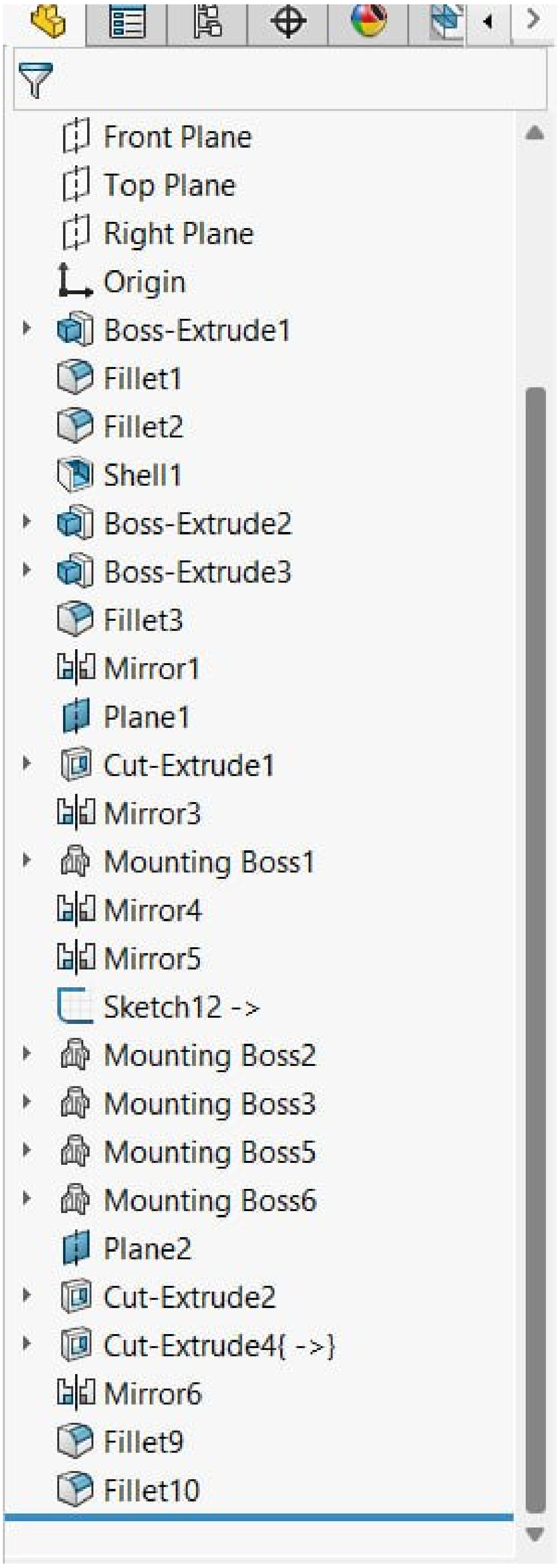


Figure 24: model tree

## 11.2 Top Part

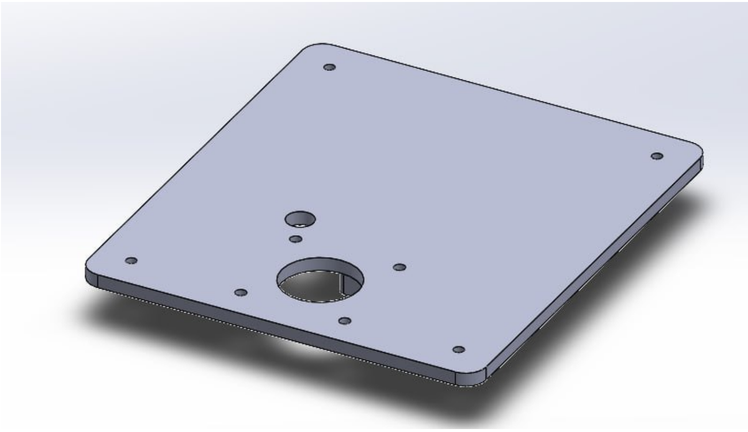

Figure 25: top view

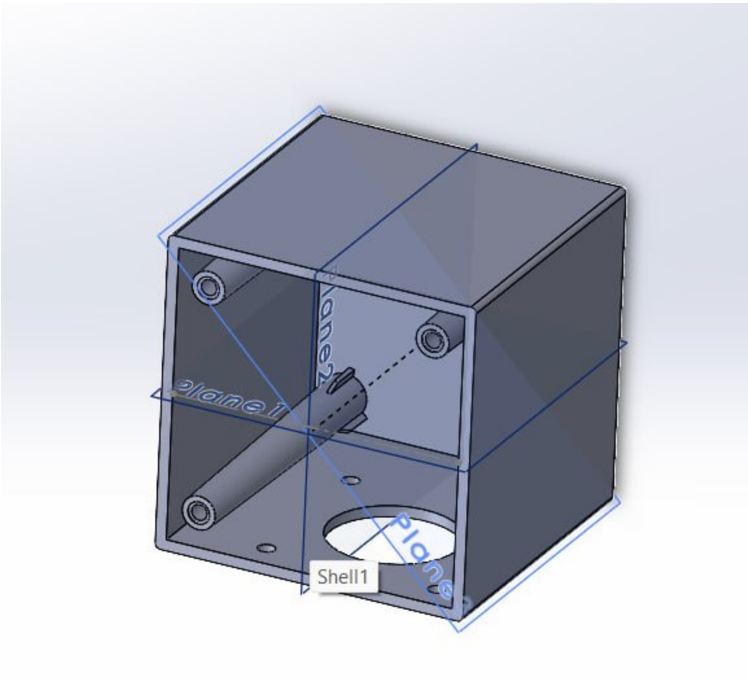


Figure 26: isometric view

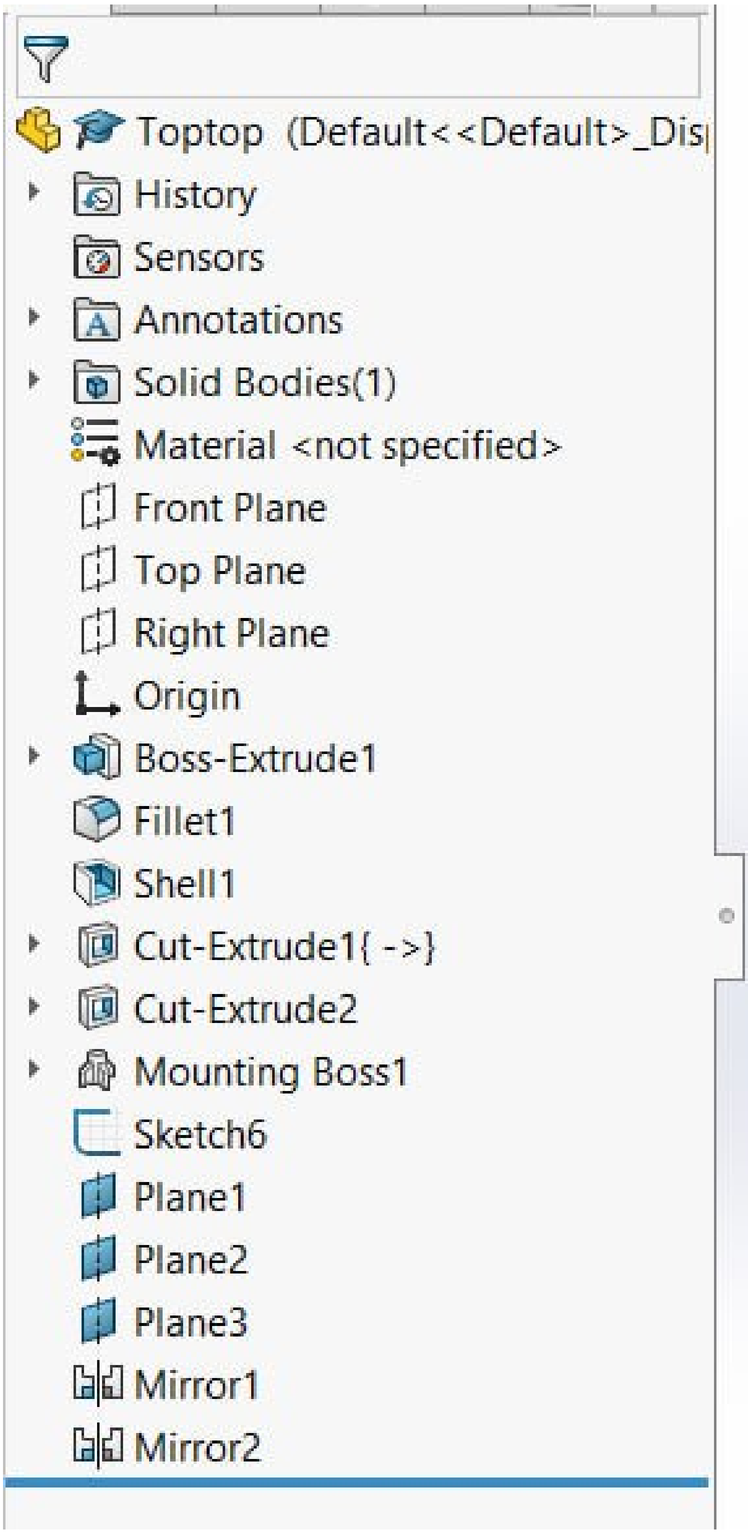


Figure 27: isometric view

## 11.3 Assembly

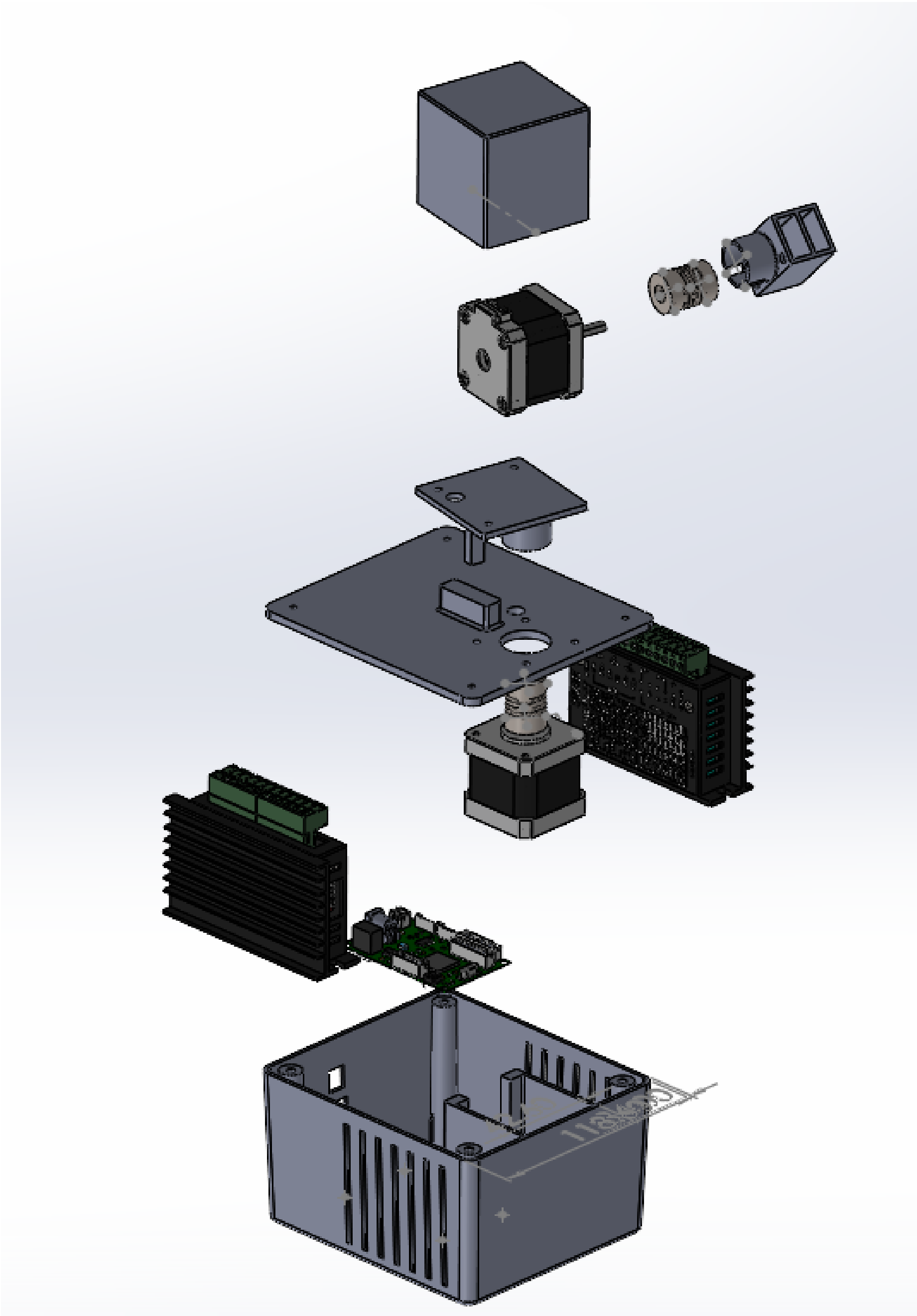

Figure 28: Exploded Isometric View

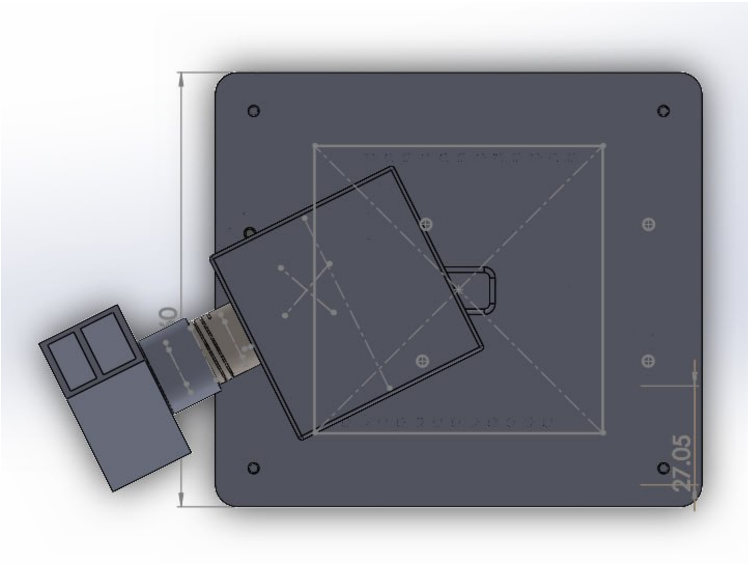


Figure 29: top view

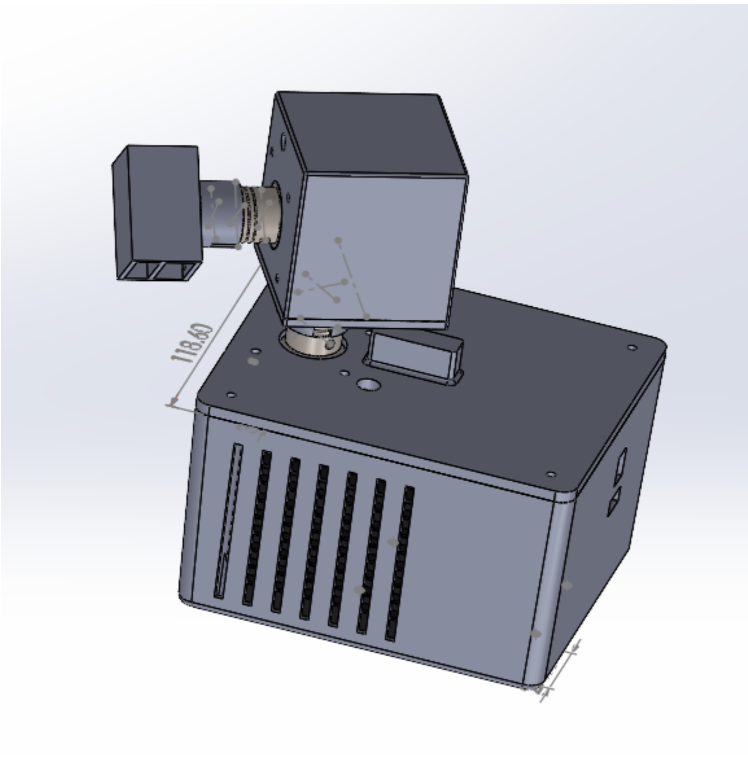


Figure 30: side view

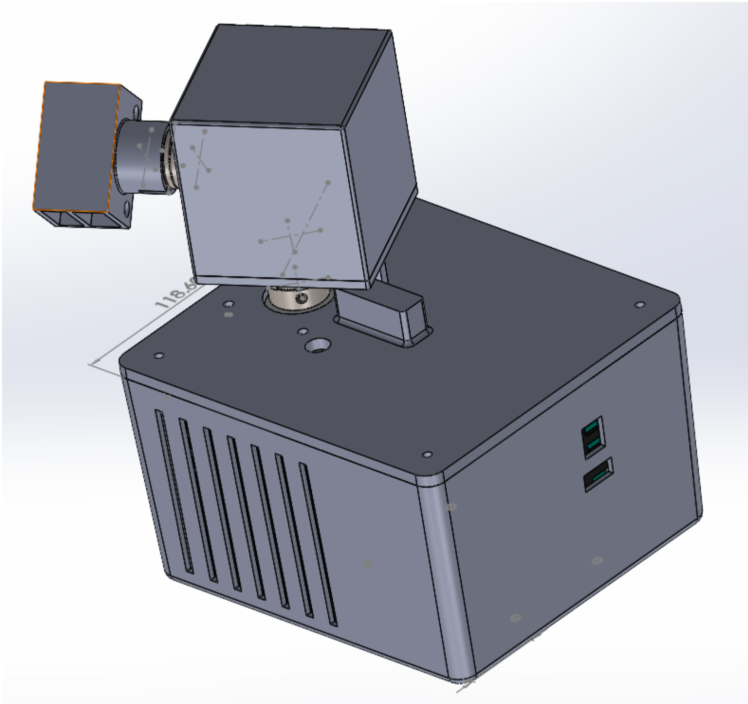

Figure 31: isometric view

### 11.4 TOF holder

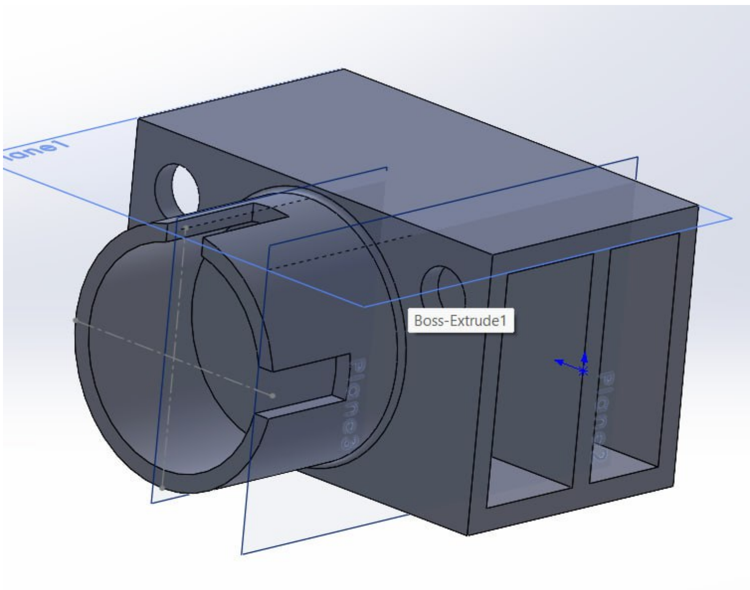


Figure 32: TOF holder

## 11.5 Standard Drawings

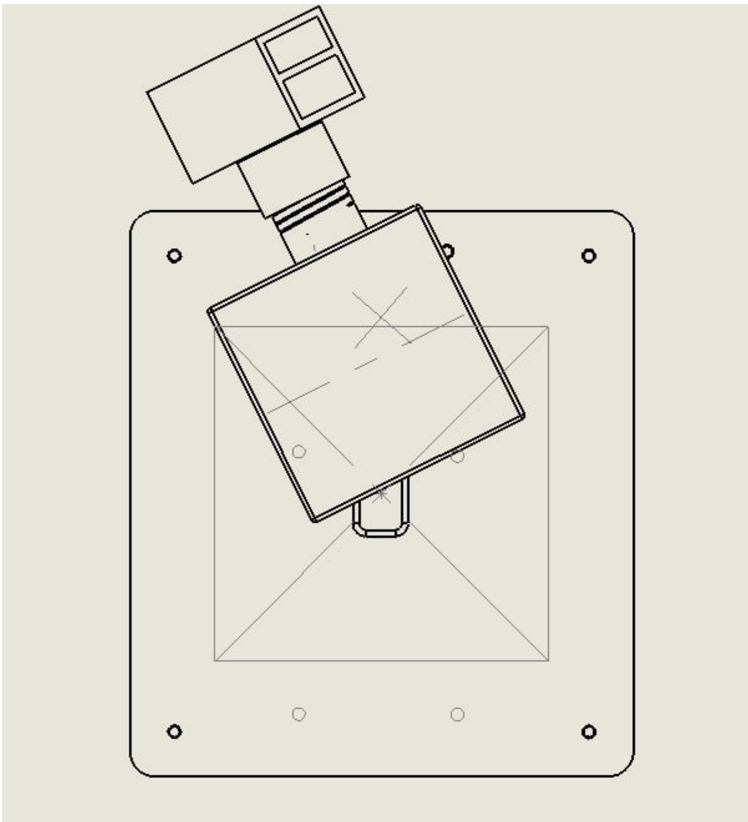

Figure 33: top view

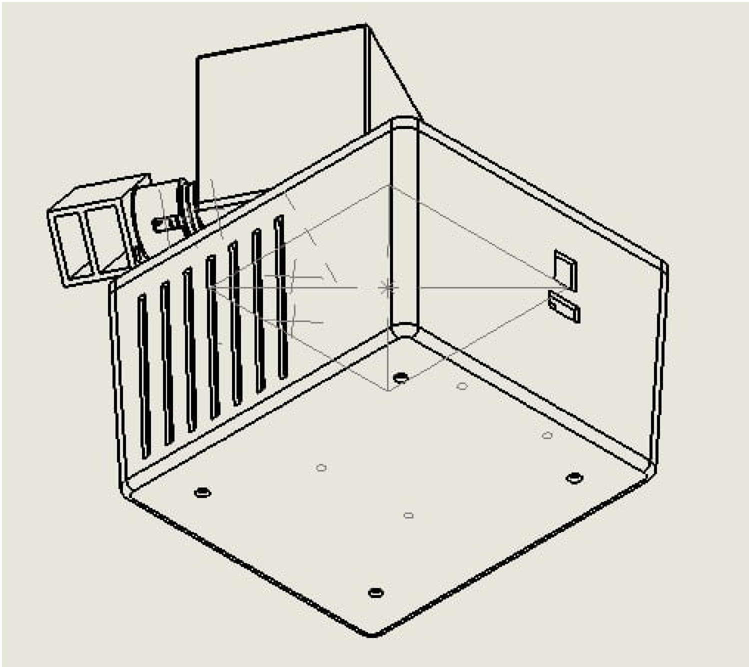

Figure 34: isometric bottom view

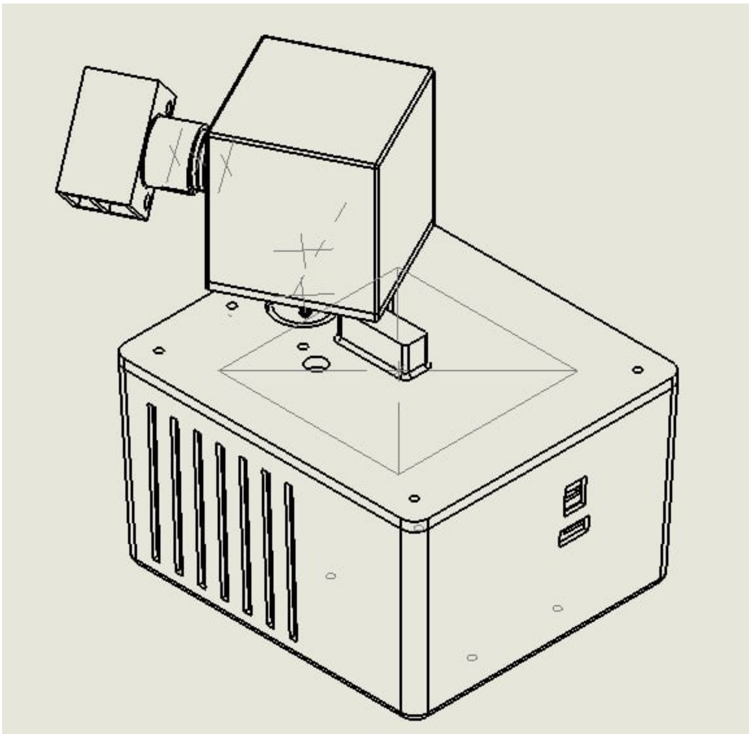

Figure 35: isometric top view

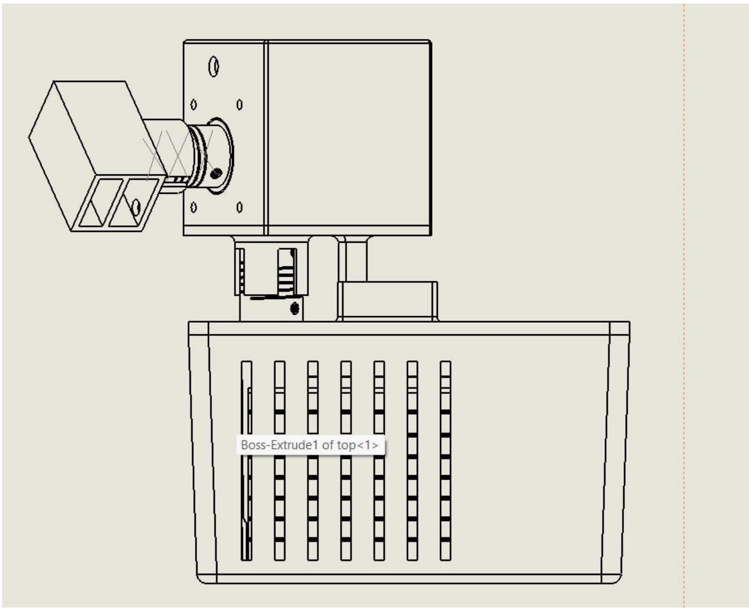


Figure 36: side view

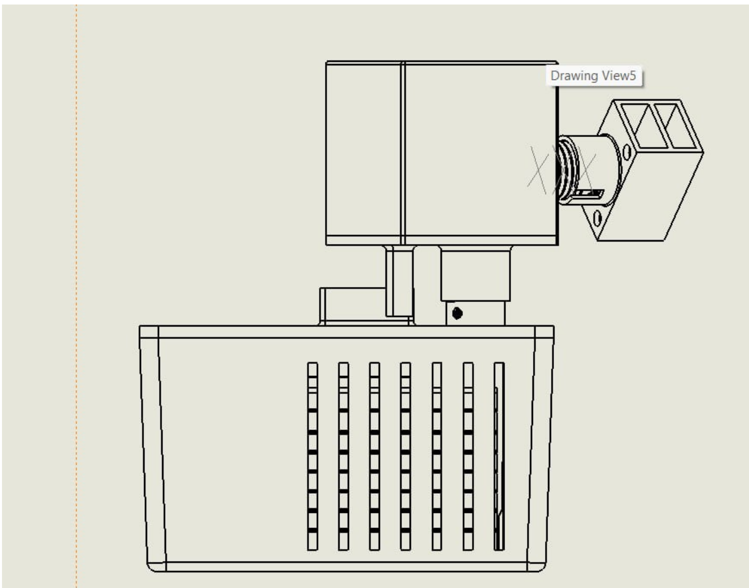


Figure 37: side view

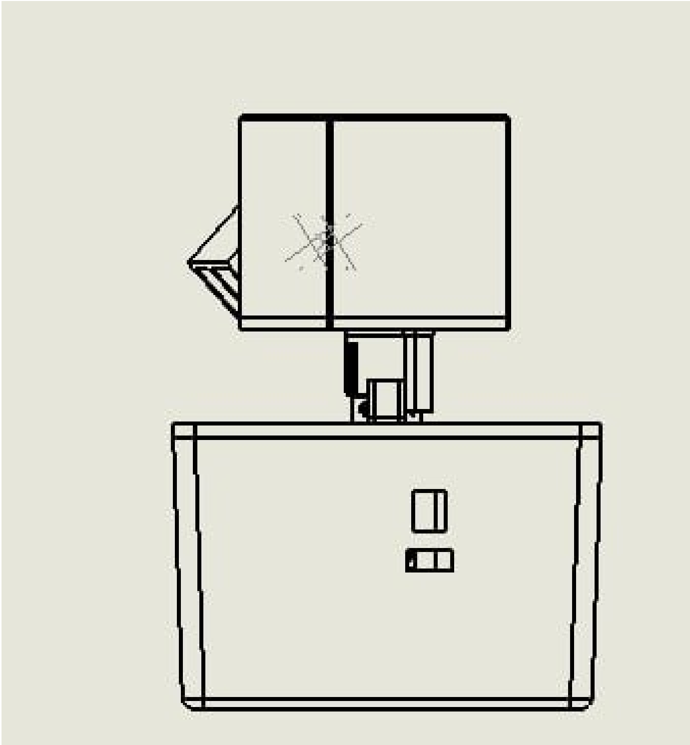

Figure 38: side view

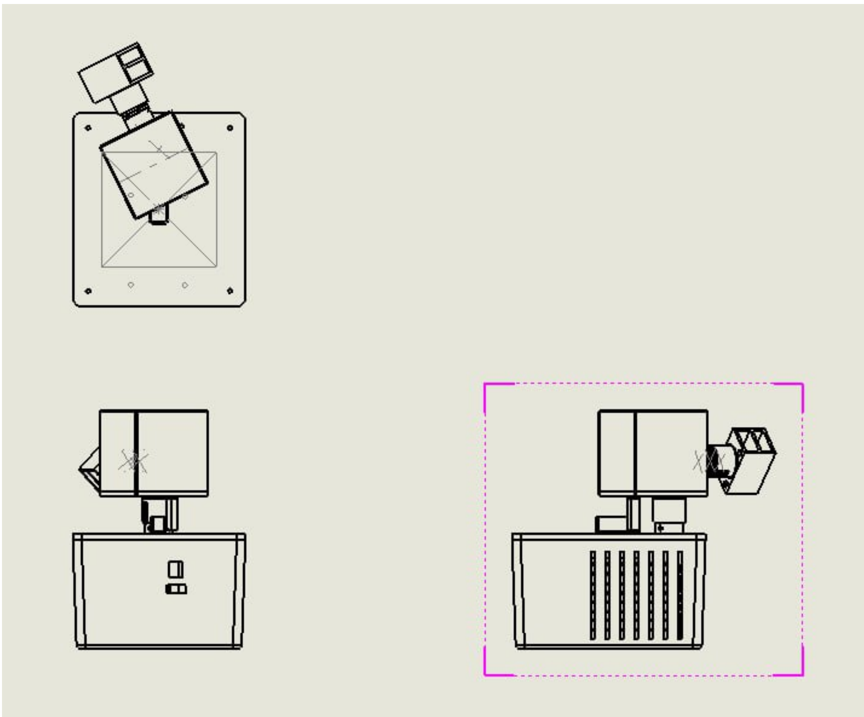

Figure 39: all views

# 12 Coding and Verification

## 12.1 Microcontroller Programming

The ATMEGA 2560 MCU was selected as the central processing unit for the 3D surround scanner. This microcontroller was chosen for its adequate memory, processing speed, and overall reliability. The key features that made the ATMEGA 2560 suitable for our project include:

- **Memory:** The microcontroller offers sufficient memory for handling the initial stages of data processing.
- **Processing Speed:** Its processing capabilities meet the demands of real-time data collection and processing.
- **Reliability:** The ATMEGA 2560 is known for its stable performance in various applications.

Despite its advantages, the memory capacity of the ATMEGA 2560 was insufficient to store the entire dataset generated during the scanning process. Therefore, the data readings had to be transmitted incrementally to a connected computer for further processing and storage.

## 12.2 Arduino Code

```
#include <Wire.h>
#include <Adafruit_VL53L0X.h>

#define STEP_PIN_1 8
#define DIR_PIN_1 9
#define ENA_PIN_1 10

#define STEP_PIN_2 4
#define DIR_PIN_2 3
#define ENA_PIN_2 2

#define XY_REV 200
#define XZ_REV 50

#define ANGLE_INCREMENT 1.8

Adafruit_VL53L0X lox = Adafruit_VL53L0X();

float measurements[XZ_REV][3];

void setup() {
  Serial.begin(9600);  // Initialize the serial monitor

  if (!lox.begin()) {
    Serial.println(F("Failed to boot VL53L0X"));
    while (1);
  }
  pinMode(4,OUTPUT);
  pinMode(3,OUTPUT);
  pinMode(2,OUTPUT);

  pinMode(7,OUTPUT);
  pinMode(8,OUTPUT);
  pinMode(9,OUTPUT);

  digitalWrite(ENA_PIN_1, LOW);
  digitalWrite(ENA_PIN_2, LOW);

  Serial.println(F("VL53L0X test with Stepper Motor"));
}

void loop() {
  // put your main code here, to run repeatedly:
  for (int i=0; i < XY_REV; i++) {
    if (i%2 == 0) {
      digitalWrite(DIR_PIN_2,HIGH);
    } else {
      digitalWrite(DIR_PIN_2,LOW);
    }
    for (int j=0; j < XZ_REV; j++) {
      VL53L0X_RangingMeasurementData_t measure;
      lox.rangingTest(&measure, false);

      if (i%2 == 0) {
        if (measure.RangeStatus != 4) {  // phase failures have incorrect data
          measurements[j][0] = j * ANGLE_INCREMENT - 45;
          measurements[j][1] = i * ANGLE_INCREMENT;
          measurements[j][2] = measure.RangeMilliMeter;
        } else {
          measurements[j][0] = j * ANGLE_INCREMENT - 45;
          measurements[j][1] = i * ANGLE_INCREMENT;
          measurements[j][2] = 10000;
        }
      } else {
        if (measure.RangeStatus != 4) {  // phase failures have incorrect data
          measurements[XZ_REV - j -1][0] = j * ANGLE_INCREMENT - 45;
```

```
        measurements[XZ_REV - j - 1][1] = i * ANGLE_INCREMENT;
        measurements[XZ_REV - j - 1][2] = measure.RangeMilliMeter;
      } else {
        measurements[XZ_REV - j - 1][0] = j * ANGLE_INCREMENT - 45;
        measurements[XZ_REV - j - 1][1] = i * ANGLE_INCREMENT;
        measurements[XZ_REV - j - 1][2] = 10000;
      }
      }
      digitalWrite(STEP_PIN_2, HIGH);
      digitalWrite(STEP_PIN_2, LOW);
    }
    Serial.print("aaa[");
  for (int k = 0; k < XZ_REV; k++) {
    Serial.print("[");
    Serial.print(measurements[k][0]);
    Serial.print(",");
    Serial.print(measurements[k][1]);
    Serial.print("],");
    Serial.print(measurements[k][2]);
    Serial.print("],");
    }
  Serial.println("]");
  digitalWrite(STEP_PIN_1, HIGH);
  digitalWrite(STEP_PIN_1, LOW);
  }

}
```

Listing 1: Arduino Code for VL53L0X Sensor with Stepper Motors

## 12.3 Data Visualization and Reciever Implementation

## 12.4 Python Code

```
import serial
import json
from math import cos, sin, pi
import matplotlib.pyplot as plt
import numpy as np
from mpl_toolkits.mplot3d import Axes3D

from datetime import datetime

# datetime object containing current date and time
now = datetime.now()

print("now =", now)

# dd/mm/YY H:M:S
dt_string = now.strftime("%d/%m/%Y %H:%M:%S")

printed = False

def plot_lines(coordinates):
    x_values, y_values = zip(*coordinates)
    plt.plot(x_values, y_values, color='blue', linestyle='-', linewidth=2,
        label='Line')

    plt.title('Graph with Connected Line (No Dots)')
    plt.xlabel('X-axis')
    plt.ylabel('Y-axis')

    # Set axis limits to always be 0 to 300
    #plt.xlim(0, 300)
    #plt.ylim(0, 300)

    # Set the aspect ratio to be equal
    plt.gca().set_aspect('equal', adjustable='box')

    plt.grid(True)
    plt.legend()
    plt.show()

def plot_dots(coordinates):
    x_values, y_values = zip(*coordinates)
    plt.scatter(x_values, y_values, color='blue', marker='o')
    plt.title('Graph with Dots')
    plt.xlabel('X-axis')
    plt.ylabel('Y-axis')

    #plt.xlim(0, 300)
    #plt.ylim(0, 300)

    plt.gca().set_aspect('equal', adjustable='box')

    plt.grid(True)
    plt.show()


def plot_3d_surface(coordinates):
    x_coords = [coordinate[0] for coordinate in coordinates]
    y_coords = [coordinate[1] for coordinate in coordinates]
    z_coords = [coordinate[2] for coordinate in coordinates]

    # Convert coordinates to numpy arrays
    x = np.array(x_coords)
    y = np.array(y_coords)
```

```
    z = np.array(z_coords)

    # Create a 3D plot
    fig = plt.figure()
    ax = fig.add_subplot(111, projection='3d')

    # Plot the scatter plot
    ax.scatter(x, y, z, c='b', marker='o', s = 1)

    # Connect the points with lines
    for i in range(1, len(x)):
        ax.plot([x[i-1], x[i]], [y[i-1], y[i]], [z[i-1], z[i]], c='b')

    ax.set_xlabel('x')
    ax.set_ylabel('y')
    ax.set_zlabel('z')

    plt.show()

def plot_3d_wireframe(coordinates):
    x_coords = [coordinate[0] for coordinate in coordinates]
    y_coords = [coordinate[1] for coordinate in coordinates]
    z_coords = [coordinate[2] for coordinate in coordinates]

    # Convert coordinates to numpy arrays
    x = np.array(x_coords)
    y = np.array(y_coords)
    z = np.array(z_coords)

    # Create a 3D plot
    fig = plt.figure()
    ax = fig.add_subplot(111, projection='3d')

    # Plot the wireframe
    ax.plot_trisurf(x, y, z, linewidth=0, antialiased=False)

    ax.set_xlabel('x')
    ax.set_ylabel('y')
    ax.set_zlabel('z')

    plt.show()


def str2list(input_string):
    input_string = input_string.replace('\r', "")
    input_string = input_string.replace('\n', "")

    temp_buffer = []
    temp_text = ''
    selected_dots = []

    for stringData in input_string.split('\n'):
        temp_buffer.append(temp_text + stringData.split('\n')[0])
        temp_text = ''

        if temp_buffer[-1][:3] == "aaa":
            temp_text = ''
            build_temp = temp_buffer[-1][3:].rstrip('\n\r,')

            # Handle the case when the string ends with a trailing comma
            if build_temp[-2]:
                build_temp = build_temp[:-2]+build_temp[-1]

            try:
                nested_list = [list(map(float, innerList)) for innerList in
                    json.loads(build_temp)]
            except json.decoder.JSONDecodeError as e:
                print("Error decoding JSON:", e)
```

```
                print("Problematic data:", build_temp)
                return False

            for nested_item in nested_list:
                x_temp = ((nested_item[2] * cos(nested_item[0] * (pi / 180)) * cos(nested_item[1] * (pi / 180))) + 2000) /10
                y_temp = ((nested_item[2] * cos(nested_item[0] * (pi / 180)) * sin(nested_item[1] * (pi / 180))) + 2000) /10
                z_temp = ((nested_item[2] * sin(nested_item[0] * (pi / 180))) + 2000) /10

                if x_temp < 1000 and y_temp < 1000:
                    selected_dots.append([x_temp, y_temp,z_temp])
                    #print("Selected dots:", selected_dots)
            return selected_dots

    return False


class ReadLine:
    def init(self, s):
        self.buf = bytearray()
        self.s = s

    def readline(self):
        i = self.buf.find(b"\n")
        if i >= 0:
            r = self.buf[:i+1]
            self.buf = self.buf[i+1:]
            return r.decode("utf-8")

        while True:
            i = max(1, min(2048, self.s.in_waiting))
            data = self.s.read(i)
            i = data.find(b"\n")
            if i >= 0:
                r = self.buf + data[:i+1]
                self.buf[0:] = data[i+1:]
                return r.decode("utf-8")
            else:
                self.buf.extend(data)

ser = serial.Serial('COM3', 9600)
rl = ReadLine(ser)

can_update = False
list_for_3d = []

while not printed:
    line = rl.readline()
    #print("Received:", line)
    print_temp = str2list(line)
    if print_temp != False:
        #print(print_temp)
        if (int(print_temp[0][1]) == 0 and int(print_temp[0][0]) == 0):
            can_update = True
        if True:
            for i in print_temp:
                list_for_3d.append(i)
            print(len(list_for_3d)/50)
            if len(list_for_3d) >= 200*50:
                #plot_dots(print_temp)
                #plot_lines(print_temp)
                #print(len(list_for_3d))
                # Specify the file path
                file_path = "example" + dt_string + ".txt"

                # Open the file in write mode
```

```
                with open(file_path, 'w') as file:
                    # Write the text to the file
                    file.write(str(list_for_3d))

                print(f"Text saved to {file_path}")
                #print(list_for_3d)
                plot_3d_surface(list_for_3d)
                printed = True
```

Listing 2: Python Code for receiver

# 13 Photographs

## 13.1 Bare PCB

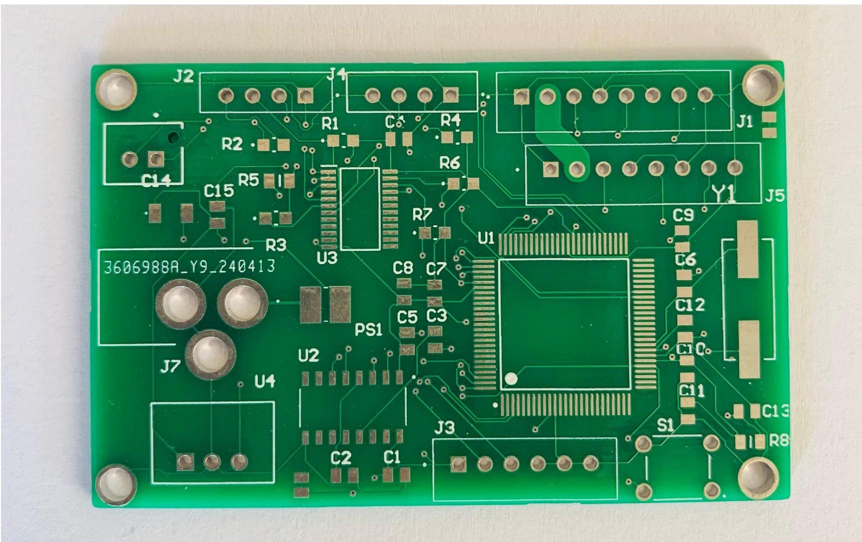


Figure 40: bare PCB

## 13.2 Soldered PCB

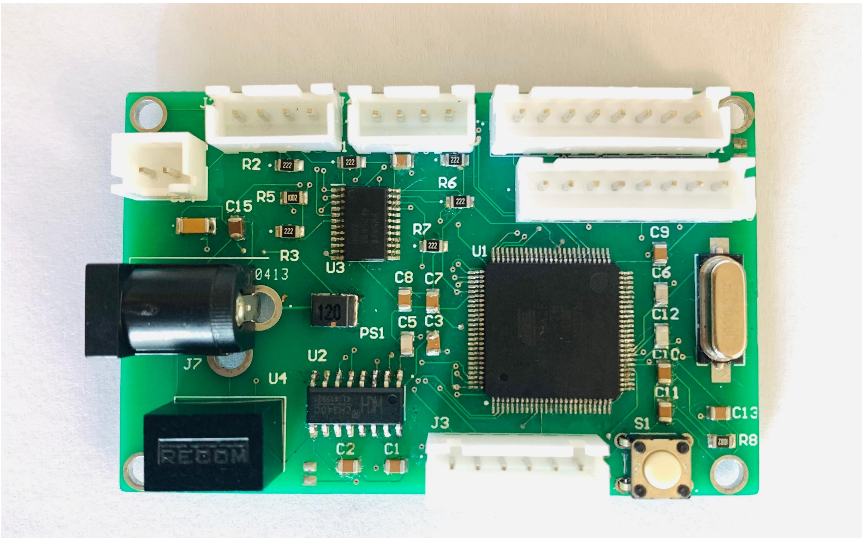


Figure 41: soldered PCB

### 13.3 Physically Built Enclosure

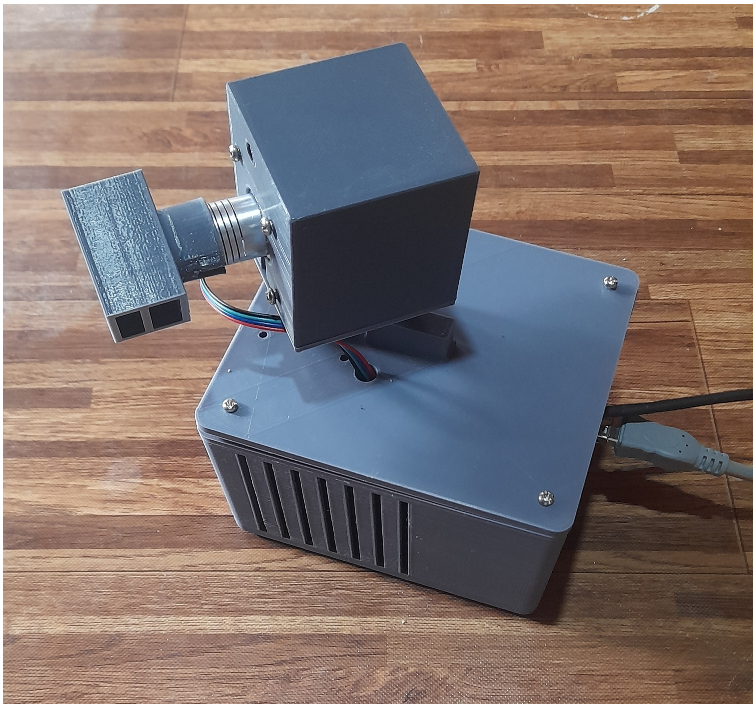

Figure 42: Enclosure

Figure 43: Enclosure

Figure 44: Enclosure

## 13.4 System Integration

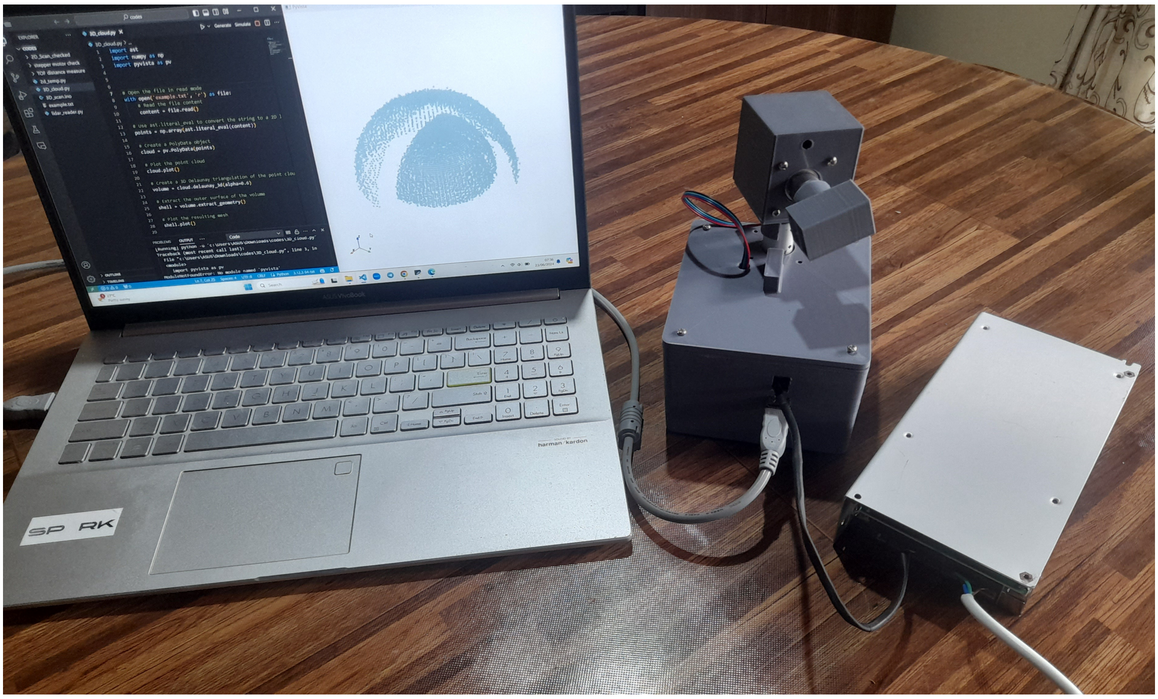

Figure 45: System Integration

Figure 46: System Integration


Figure 47: 3D scanner and power supply

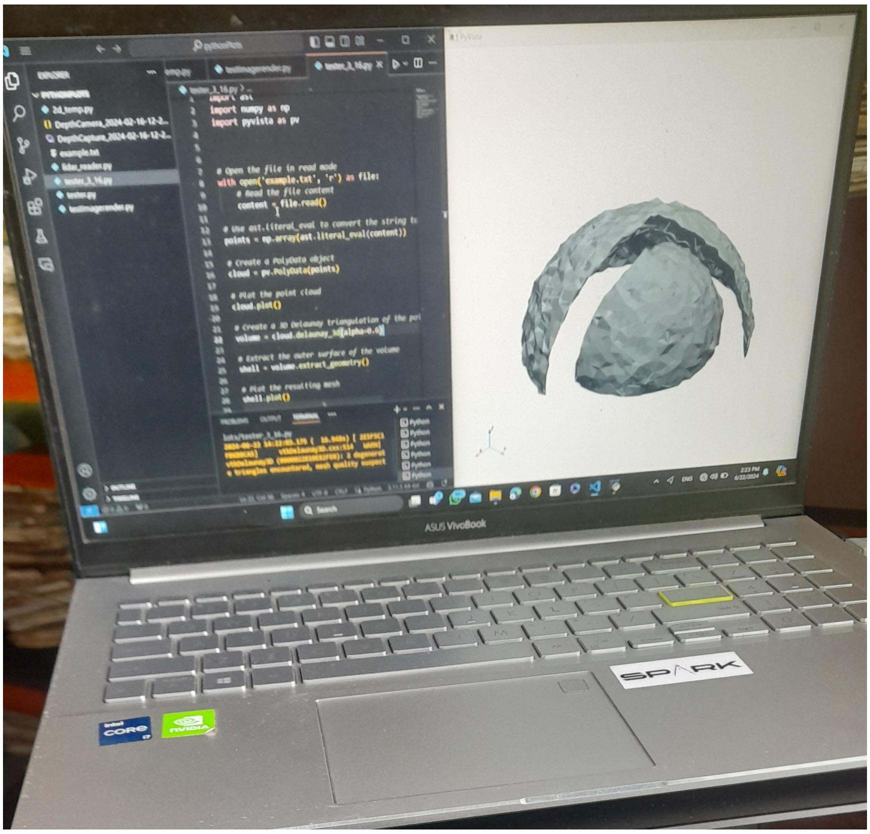


Figure 48: 3D map