MEng Final Project Report (2022/23) [1]**:**

Design and Development of an Open-Source Humanoid Robotic Head

Student Name	Nicholas Hedworth
Student Number	189099729
Degree Programme	Integrated Mechanical and Electrical Engineering MEng
Supervisor	Dr Uriel Martinez Hernandez
Assessor	Dr Min Pan

Submission Date	12 / 05 / 2023
Word Count	11,777

Department of Electrical Engineering

Faculty of Engineering and Design

UNIVERSITY OF BATH

Summary

This report presents the development process for the design of a humanoid robotic head which was created to be an accessible and affordable platform for research and teaching purposes. The head contributes to the BEATRIX project (**B**ath op**E**n hum**A**noid for **T**eaching and **R**esearch in **I**ntelligent roboti**X**), the first humanoid robot to be fully developed at the University of Bath. The system is comprised of several modules, including the neck mechanism, eye mechanism, head design, base structure, and control system.

The neck mechanism is based on a spherical parallel manipulator, which provides three degrees of freedom such that it can emulate the movement of a human head. The eye mechanism utilises a system of universal joints and servo motors, inspired by Will Cogley's "Advanced Eye Mechanism", to provide independent control of each eye in two axes. The head design is such that it is approachable, gender-neutral, and allows for easy access of the components it contains such as the eyes and electronic boards. The base structure supports the entire assembly and houses the power circuit. The control system includes actuator control, sensor integration, and communication with external devices, using various libraries and tools to achieve the desired functionality.

Throughout the design process cost, accessibility, and open sourcing have been considered. This ensures that the project remains affordable, easy to replicate, and adaptable for a wide range of users. Some limitations of the current design include slow stepper motors, weak swivelling arms, and heat dissipation concerns. Future work has been discussed for focus on enhancing the neck mechanism, incorporating more sensors for advanced control algorithms, and integrating the Robot Operating System (ROS) which would further contribute to the goals of the project.

Overall, the 3D-Printed humanoid robotic head presented in this report should provide a promising platform for the intended audience. I hope you enjoy reading my report!

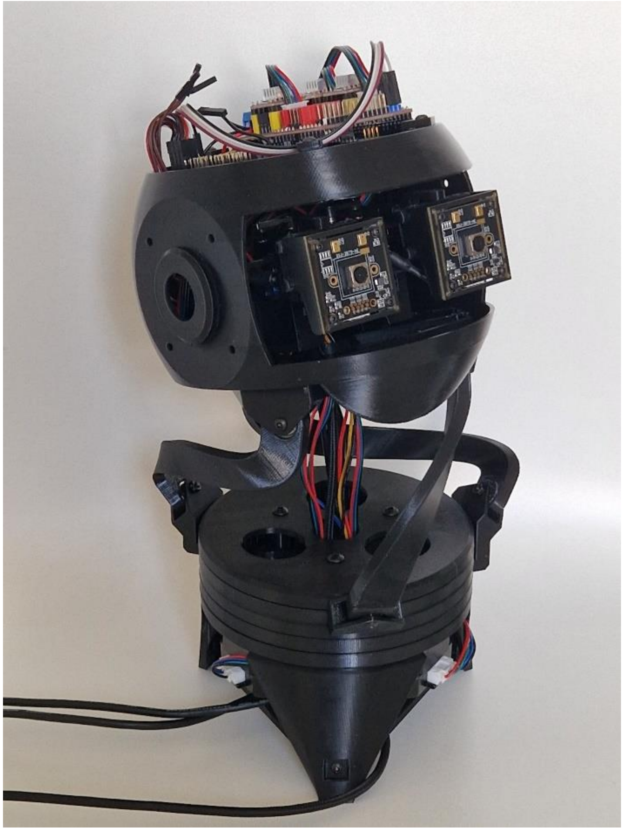


Figure 1 – Final Design Fully Assembled

Acknoledgements

There are many people that deserve thanks for their insight to accelerate the work provided in this document. I would firstly like to thank my supervisor Uriel Martinez Hernandez for expertly guiding me through how I was going to achieve the many ambitious visions I had, despite my lack of experience. My learning of robotic systems was greatly improved because of him.

I would like to thank Tareq Assaf for aiding me in the design of gears and pointing me in the right direction for resources in the subject.

Finally, thank you to Ishrat Fairoz for providing me with some insightful information on the dynamics of the human head to aid in the design of the project.

Figure 2 – Final Design Fully Assembled with Angled Mechanisms

Contents

Figures

1 Introduction

1.1 Humanoid Robots

A humanoid robot is a type of robot that is designed to look like humans and/or can perform similar physical tasks to a human [2]. These robots usually have a head, torso, arms, and legs, to perform tasks in a wide range of settings, much like a human.

One of the reasons for creating humanoid robots is to improve the interaction between machines and humans [2]. By designing robots that mimic humans, people are theoretically more likely to feel comfortable interacting with them, which can make them more effective than other types of robots in a variety of settings.

Humanoid robots can be used in applications such as manufacturing, healthcare, entertainment, education, and research [3]. In manufacturing, for example, humanoid robots can be used to perform tasks that are deemed too dangerous or difficult for humans to do, such as working with heavy machinery or hazardous chemicals.

The key challenges involved in designing a humanoid robot includes the mechanical design, power electronics, sensory capabilities, and digital control. Generally, humanoid robots are optimised for specific applications, resulting in low production volumes and a limited number of general-purpose robots in the industry. This restricts the affordability and accessibility of humanoid robots.

Figure 3 – Example of a Humanoid Robot, Honda Asimo [4]

1.2 The BEATRIX Project

BEATRIX (**B**ath op**E**n hum**A**noid for **T**eaching and **R**esearch in **I**ntelligent roboti**X**), the first humanoid robot to be fully developed at the University of Bath, is an exciting platform for researchers to explore the potential of humanoid robotics. A key aspect of the BEATRIX project is its open-source nature, providing the robotics community with free access to the system's design. This approach encourages education, research, and innovation in the field of humanoid robots.

The goal of the BEATRIX project is to design an accessible and affordable open-source humanoid robotics platform which retains as many of the key capabilities which a biological human has. This democratises humanoid robotics technology and encourage collaboration. The BEATRIX project allows researchers and students to experiment and develop their ideas through modular design. This open approach also addresses the challenges of limited production volumes and affordability, enabling a broader range of individuals and organisations to participate in the development and utilisation of humanoid robots.

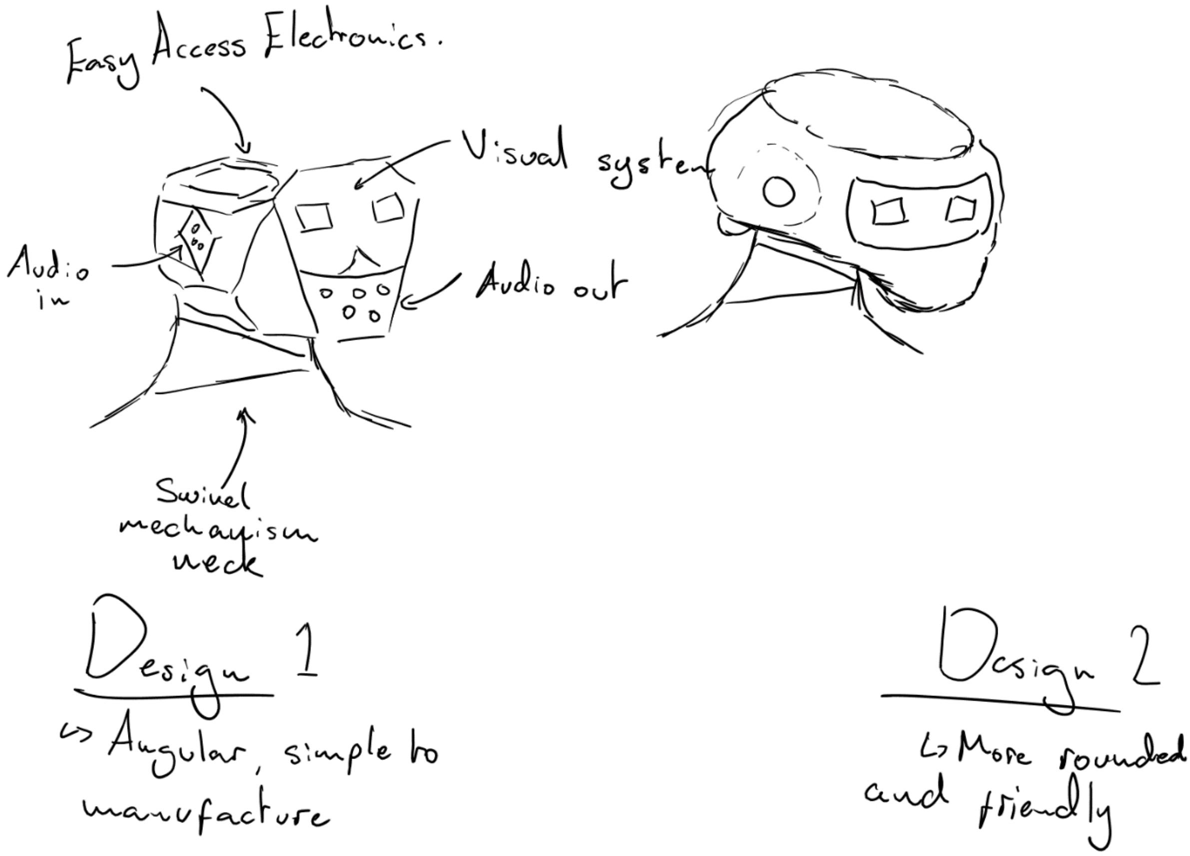


Figure 4 – Initial illustration of BEATRIX Head (sorry about my drawing skills)

1.3 Project Aims and Report Structure

The head of a humanoid robot is a crucial design element for sensory input, information processing, and communication with humans. As well as being in line with the goals of the BEATRIX platform, the project deliverables will be as follows:

- **Physical Prototype**: The primary goal of this project is to produce a physical model of a robotic head. This should be done with consideration of materials and manufacturing processes, whilst also designing it such that it is inclusive and friendly for people to interact with.
- **Electrical Component Specifications**: The prototype will include the necessary electrical component to facilitate all the integrated functions of the project. Specifications of these electrical components should be provided.
- **Digital Control and Sensor Transmission**: Control algorithms will be provided, and the transfer of data from sensors such as cameras and microphones to an external computer will be explored.

The aim of this report is to outline the process of designing the head portion of a humanoid robot and present a final design for the BEATRIX project to be built upon. The report is structured to first consider the mechanisms, which inform the requirements for actuators and electronics in the system. The sensors and digital architecture are then discussed, followed by the mechanical design process, which considers any difficulties or changes from the initial direction. Additional considerations, such as maintaining specific goals for the BEATRIX project, are also addressed. The report concludes with an examination of the completed model, reflects on the project, and provides suggestions for future work to improve upon the project's goals. The outcome is a fully functional model that moves and displays sensory data to an external computer.

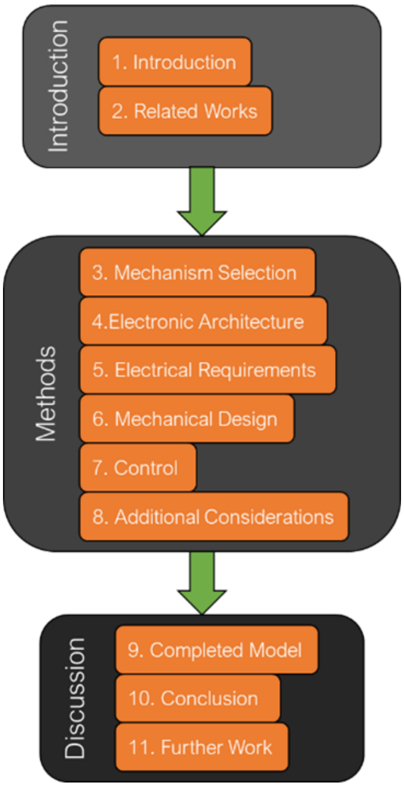


Figure 5 – Report flowchart

2 Related Work

Researching the work of existing humanoid robot designs plays a crucial role in developing a new platform, such as the BEATRIX project. Valuable insights can be gained from the successes and limitations of various approaches, as well as identifying potential areas for improvement or innovation. Moreover, the analysis of other designs can provide inspiration and guidance on best practices for mechanical design, electronic architecture, and control systems. This ultimately leads to a more efficient, cost-effective, and successful humanoid robot project.

In the context of the BEATRIX project, it is particularly important to focus on related works that align with the project's core objectives of accessibility, affordability, and openness. Therefore, the analysis should emphasise open-source projects, budget-friendly solutions, and those that have a strong educational or research component. By concentrating on these types of projects, the BEATRIX platform will be better positioned to benefit from the lessons learned and advances made by other teams working towards similar goals.

In this section, several notable humanoid robot projects will be reviewed, including InMoov, Poppy, Reachy, and iCub. Their design features, mechanisms, and control systems will be discussed, highlighting their strengths and weaknesses, as well as any innovative solutions that can be adapted or built upon for the BEATRIX project. Through a comprehensive analysis of these related works, a solid foundation can be established for the design and development of the BEATRIX humanoid robot head.

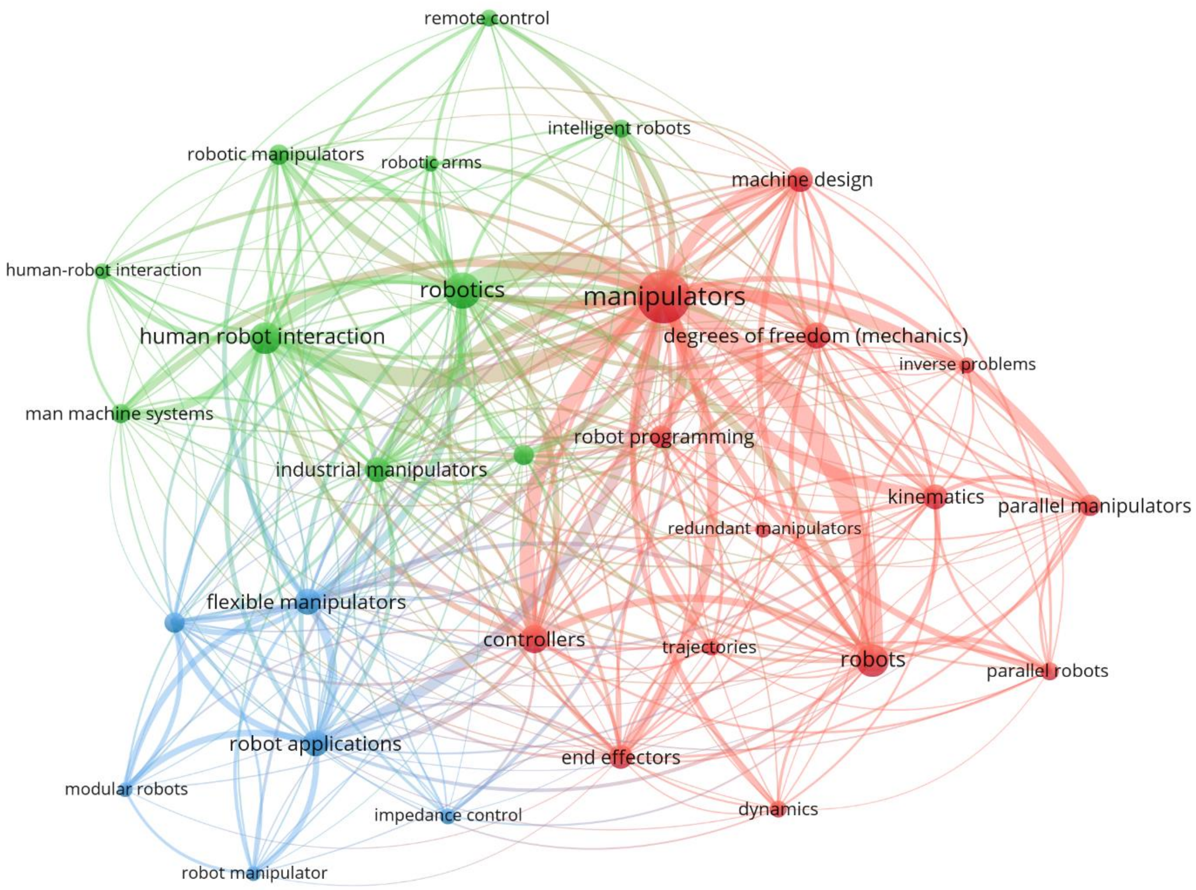


Figure 6 – Map of Terms used in Related Works (Made using VOSviewer)

2.1 InMoov

2.1.1 Summary and Description

InMoov is an open-source, 3D-printed, life-size humanoid robot project that was initiated by Gael Langevin, a French sculptor and designer [5]. The robot is designed with the aim of being a platform for development and robot learning, leveraging the growing availability of 3D printing technology. InMoov has gained significant popularity among the maker community, research institutions, and educational organisations worldwide, as it allows individuals to build their own humanoid robot using commonly available materials and components.

2.1.2 Head Functions and Implementation

- **Vision**: The InMoov head is usually equipped with two individually actuated cameras that serve as the robot's eyes, providing a wide field of view and stereoscopic vision [6].
- **Auditory Input/Output**: The head houses microphones and speakers for capturing and producing sound. In some projects, this enables the robot to respond to voice commands and engage in conversation with humans.
- **Head Movement**: The InMoov head is mounted on a neck mechanism consisting of three linear actuators and a ball joint which allows the robot to perform pan, tilt, and roll movements. This enables the head to track objects or people, maintain eye contact during conversation, and generally exhibit more lifelike behaviour.

These functions are achieved through a combination of 3D-printed mechanical components, servo motors, cameras, microphones, and an Arduino-based electronic control system. The modular and open-source nature of the InMoov project allows for continuous improvement and customisation of the head's features and capabilities by the community.

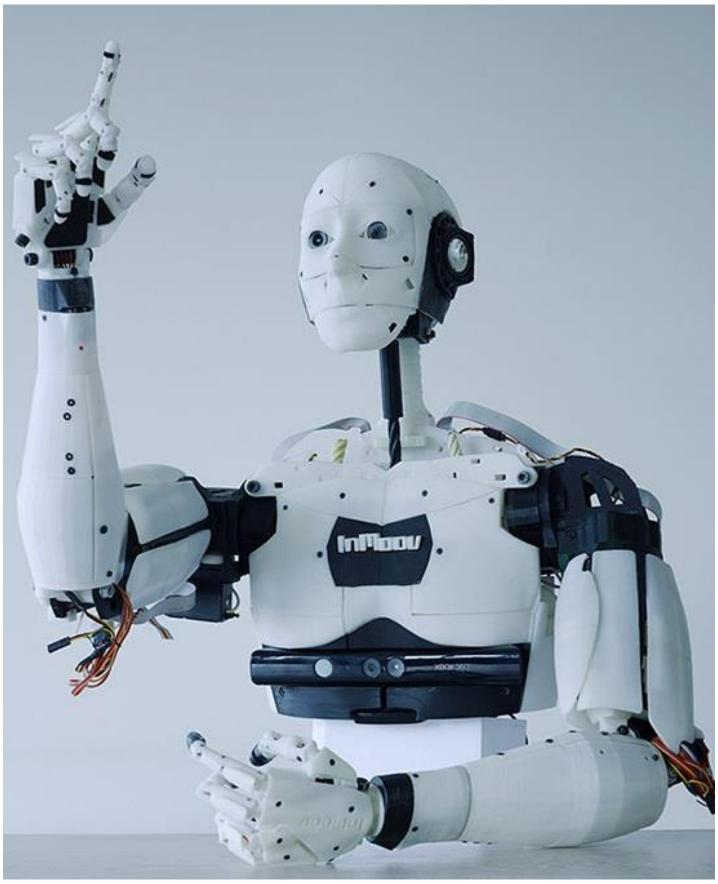


Figure 7 – Image of the InMoov Robot (by Fabrice Bouquet) [7]

2.2 Poppy

2.2.1 Summary and Description

Poppy Robot is an open-source, modular, and customisable robotic platform developed by the Flowers Lab at INRIA Bordeaux Sud-Ouest in France [8]. The project aims to facilitate research, education, and innovation in the fields of robotics, artificial intelligence, and computer science [9]. Poppy Robot's modular design enables users to create various robotic forms, including the popular Poppy Humanoid and Poppy Torso. The platform leverages 3D printing technology and the accessibility of common hardware components to encourage adoption of the platform and collaboration among researchers, educators, and enthusiasts.

2.2.2 Head Functions and Implementation

- **Vision**: Poppy's head features a single fixed camera that serves as the robot's primary visual sensor.
- **Auditory Input**: The head is equipped with a stereo microphone to capture sound, enabling the robot to perceive and analyse auditory information, such as voice commands or environmental sounds.
- **Facial Expressions**: Poppy can display video through the screen which acts as the robot's face. This method of expression is very cost effective and versatile for the purpose of exhibiting facial expressions.
- **Head Movement**: Poppy's head is mounted on a neck mechanism that provides two degrees of freedom (pan and tilt). This enables the head to perform basic movements, such as tracking objects.

These functions are achieved through a combination of 3D-printed mechanical components, Dynamixel servo motors, cameras, microphones, and a Raspberry Pi-based electronic control system. The Robot Operating System (ROS) is used for communication and control, while the Pypot library provides a high-level interface for programming and interacting with the robot [8]. The open-source and modular nature of the Poppy project allows users to easily customise and expand the head's features and capabilities to suit their specific requirements.

Figure 8 – Image of the Poppy Robot [8]

2.3 Reachy

2.3.1 Summary and Description

Reachy Robot is an open-source, versatile, and user-friendly robotic platform developed by Pollen Robotics in collaboration with INRIA Flowers Lab in France. Designed for research, education, and business applications, Reachy Robot is a highly adaptable robotic system that can be easily customised for various tasks. The platform primarily features a humanoid torso with two highly dexterous arms and a unique head. Reachy Robot's modularity and use of 3D printing technology, along with widely available hardware components, enable users to tailor the robot to their specific needs and strengthens collaboration within the community.

2.3.2 Head Functions and Implementation

- **Vision**: Reachy's head is equipped with two cameras, providing stereoscopic vision for the robot. They are individually motorised to enable zooming.
- **Auditory Input**: The head can be fitted with a microphone to capture sound, allowing the robot to perceive and process auditory information, such as voice commands or environmental sounds.
- **Head Movement**: Reachy's head is mounted on a neck mechanism called the Spherical Parallel Manipulator, which provides three degrees of freedom (pan, tilt, and roll). This allows the head to emulate the complex movements of a human neck.

Much like the Poppy robot, these functions are achieved using 3D-printed mechanical components, Dynamixel servo motors, cameras, microphones, and an electronic control system based on Raspberry Pi and Arduino.

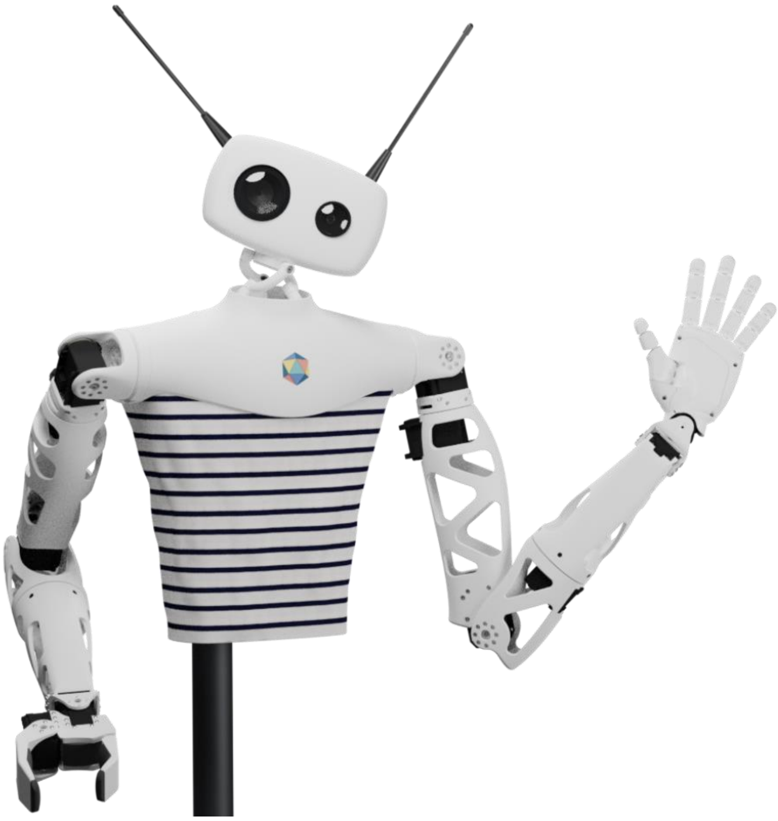


Figure 9 – Image of the Reachy Robot [10]

2.4 iCub

2.4.1 Summary and Description

The iCub Robot is an open-source, humanoid robot designed by the Italian Institute of Technology (IIT) as part of the EU-funded RobotCub project. The iCub is approximately the size of a 3.5-year-old child, and its design focuses on mimicking human sensorimotor and cognitive abilities [11]. The primary goal of the iCub project is to advance research in artificial intelligence, robotics, cognitive systems, and human-robot interaction. The robot has been adopted by numerous research institutions worldwide and has become a widely recognised platform for exploring developmental robotics, machine learning, and embodied cognition.

2.4.2 Head Functions and Implementation

- **Vision**: The iCub head is equipped with two high-resolution cameras located in its eye sockets, providing stereoscopic vision capabilities. This allows the robot to perceive depth, recognise objects, track movements, and interact with its environment more effectively. The eyes can move independently of one another, adding more possibilities for a range of applications [12].
- **Expression**: The iCub robot can express emotion using its eyes and eyelids, as well as lights at the back of its head.
- **Auditory Input/Output**: The iCub head contains two microphones and an artificial voice box, enabling the robot to capture and produce sounds. This facilitates speech recognition, voice command processing, and even engaging in conversations with humans.
- **Head Movement**: The iCub head is connected to a neck mechanism that uses a simple pan-tilt-roll mechanism to enable a wide range of movements. This allows it to better emulate the behaviour of a human to align with the goals of the project.

These functionalities are achieved through a combination of custom-made mechanical components, motors, cameras, microphones, and an advanced electronic control system. The open-source nature of the iCub project cultivates a collaborative environment where researchers and developers can contribute to the platform's continuous improvement and explore new applications and capabilities.

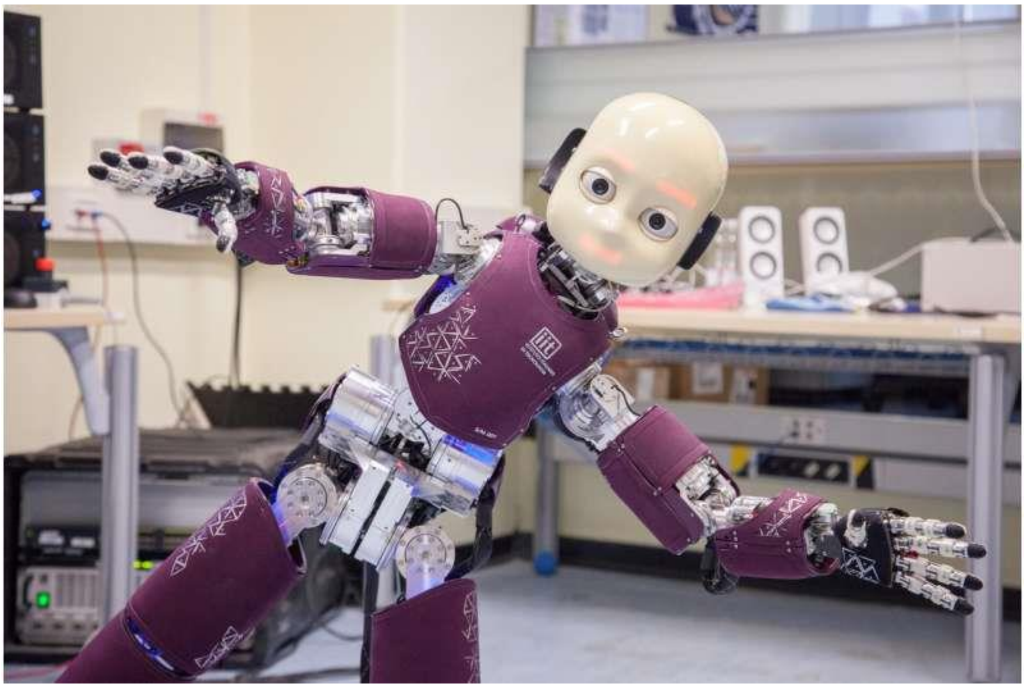


Figure 10 – Image of the iCub Robot [13]

2.5 Comparison and Summary

The key takeaways from this section are summarised:

- **Open-source and modular design**: All four projects embrace open-source principles, enabling continuous improvement, customisation, and community involvement. Modularity is a common theme among these robots, allowing users to adapt them to various applications and encouraging innovation.
- **3D printing and accessible hardware**: InMoov, Poppy, and Reachy all leverage 3D printing technology and widely available hardware components to achieve affordability and accessibility. This approach aligns well with the core objectives of the BEATRIX project.
- **Vision and auditory systems**: Each robot implement vision and auditory input/output capabilities, which are essential for effective human-robot interaction. Different approaches to these systems, such as monocular vs. stereoscopic vision or varying degrees of head movement, can provide valuable insights for the BEATRIX design.
- **Control systems and software**: The choice of electronic control systems, such as Arduino or Raspberry Pi, and the use of software frameworks like ROS or Python libraries, can impact the overall performance, ease of use, and expandability of the robot.

By analysing the InMoov, Poppy, Reachy, and iCub humanoid robot projects, important lessons can be drawn and inspiration for the BEATRIX platform. Open-source, modular design, and the use of 3D printing technology and accessible hardware are key factors in achieving the project's goals of accessibility, affordability, and openness. Examining the different approaches to vision, auditory, and head movement systems, as well as control systems and software, will guide the development of the BEATRIX humanoid robot head, ensuring a successful and impactful project outcome.

	Vision		Audio			
	Stereo	Movement	Input	Output	Expression	Head Movement
InMoov						
Poppy	Some Variants of the robot have stereo					Only a 2DOF mechanism is used
Reachy					Expression is done through the antennas	
iCub						

Table 1 - Comparative Analysis of Related Work

3 Mechanism Selection

3.1 Mechanism Identification

3.1.1 Overview of Human Head Mechanisms

The human head houses various mechanical mechanisms that contribute to functions like movement, facial expressions, and communication. Here, the key mechanical mechanisms which are present in a human head are described, as well as their functionalities, and their operating principles.

- **Facial Expressions**: Facial movements are enabled through muscles which are connected to the facial skin and bones. The muscles can contract or relax to make the skin and facial features move in various ways. The primary purpose of having facial expressions is to convey emotion and information.
- **Jaw Movement**: The jaw can move in a hinge-like motion (opening and closing) as well as moving from side to side. The joint that enables this is the temporomandibular joint along with various surrounding muscle tissue. The purpose of having jaw movement is for eating and speaking.
- **Ear Movement**: This operates in a similar manner to the facial expressions. Muscles around the ear contract or relax to move the ear over a limited range. These don't have much use for humans but are biologically relevant for animals like cats or horses which use this ability to better hear in the direction of interest.
- **Eye Movement**: Six extraocular muscles work together to move the eyeball in various directions. The operating principle is based on coordinated muscle contractions that rotate the eyeball within the eye socket. This enables the eyes to track objects and quickly and efficiently change the direction of focus.
- **Neck Movement**: The neck connects the head to the torso and consists of vertebrae, muscles, and ligaments that enable a range of head movements, such as rotation, tilting, and flexion. The operating principle involves the coordinated action of muscles and joints to move the head in various directions.

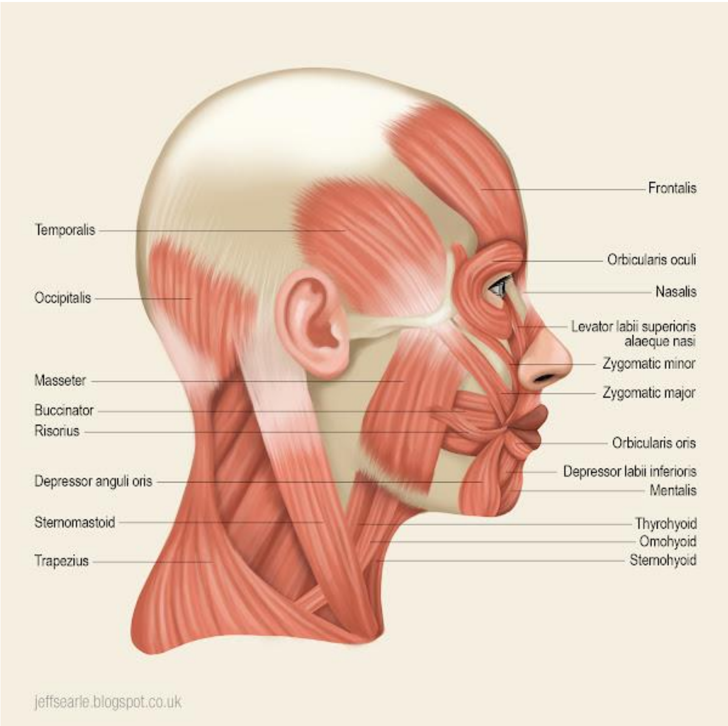


Figure 11 – Biological Diagram of the Head and Muscles [14]

3.1.2 Key Mechanisms for Humanoid

Considering the goals of the BEATRIX project and with respect to the related work already developed in the field, the focus will be placed on the neck and eye mechanisms for the following reasons:

- **Comparative analysis**: By examining the related work section, it can be observed that that the head portion of the InMoov, Poppy, and Reachy robots are mainly focused on neck movements to achieve their specific goals. The iCub robot, on the other hand, has a focus on neck and eye movement. Because of this, the ambitious goal of designing both mechanisms can be worked towards.
- **Budget constraints**: The BEATRIX project aims to be accessible and affordable, focusing on the neck and eye movements allow for the allocation of resources to be focused on these essential functions, thereby maximising the robot's performance within the project's budget constraints.
- **Core functionality**: Eye and neck movements are crucial for functions such as visual perception, object tracking, and establishing a field of view. These movements enhance the robot's ability to interact with its environment, making them more relevant to the project's goals.

The following reasons justify why facial expressions, jaw movements, and ear movements are not considered relevant for the BEATRIX project:

- **Complexity**: Implementing facial expressions, mouth movements, and ear movements would require very complex and intricate mechanisms, which would lead to higher production costs, thereby clashing with the project's goals on accessibility and affordability.
- **Limited benefits**: While facial expressions, mouth movements, and ear movements can enhance human-robot interactions, they may not significantly contribute to the robot's core functions, especially in the context of the BEATRIX project. The added benefits of implementing these mechanisms may not justify the additional costs and complexity associated with their development.
- **Focus on primary functions**: By concentrating on the neck and eye mechanisms, the project can prioritise the development of primary functions that are essential for a humanoid robot's performance, such as visual perception and head mobility. This focus aligns better with the goals and requirements of the BEATRIX project.

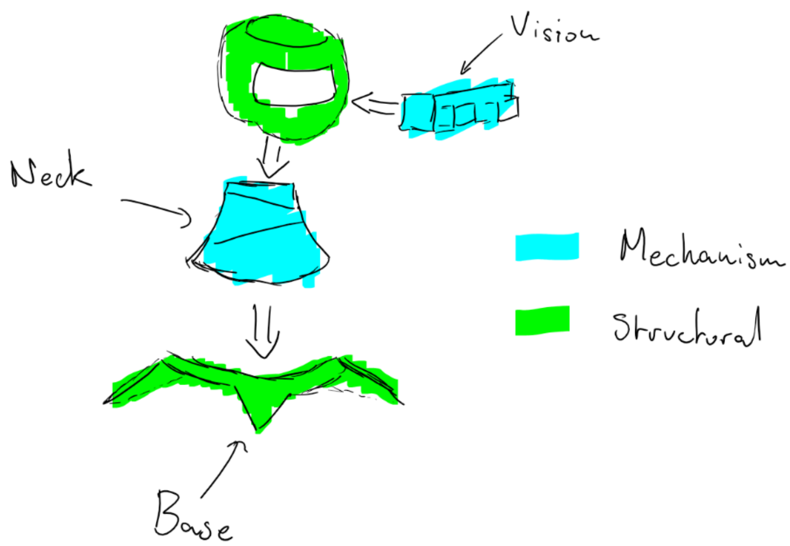


Figure 12 – Diagram of Complete Head Highlighting Mechanisms.

3.2 Neck

3.2.1 Analysis of Human Neck Movement

The human neck is a complex structure that provides support, flexibility, and a wide range of motion for the head. It enables people to perform various tasks, such as looking around, nodding, and tilting our head. The neck's movement is facilitated by a combination of bones, muscles, ligaments, and joints working together.

The neck, or cervical spine, consists of seven vertebrae (C1 to C7) stacked one on top of the other. These bones are separated by intervertebral discs that act as cushions and absorb shock [15]. The cervical vertebrae form joints with one another, allowing for a limited range of motion. Numerous muscles and ligaments in the neck region work together to control the movement.

The neck's range of motion is categorised into three primary movements:

- **Flexion and Extension**: Bending the head forward and backward.
- **Lateral flexion**: Bending the head to the side, bringing the ear toward the shoulder.
- **Rotation**: Turning the head from side to side, looking over the shoulder.

The neck's range of motion varies among individuals, but on average, the neck can rotate approximately 80-90 degrees in each direction, flex 45-50 degrees, extend 50-60 degrees, and laterally flex 45 degrees.

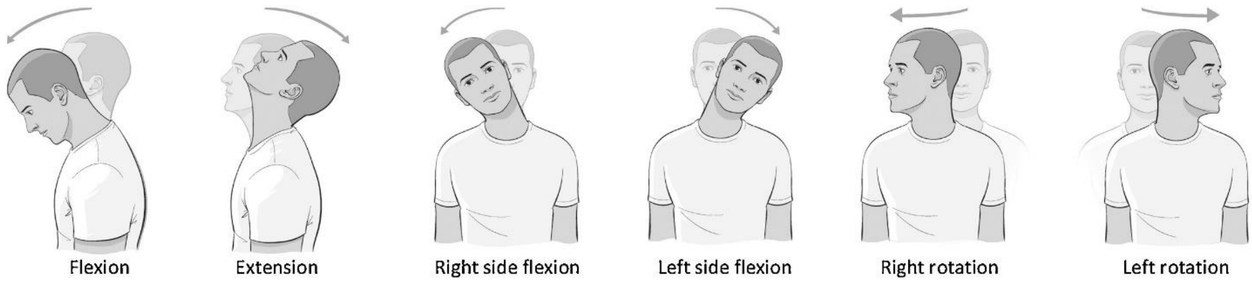


Figure 13 – Diagram of the Neck Movement [16]

3.2.2 Pan-Tilt-Roll Mechanism

The Pan-Tilt-Roll (PTR) mechanism is a general form of manipulator whereby three joints are placed at 90 degrees from each other to provide pan (yaw), tilt (pitch), and roll [17]. The structure usually consists of a base connected to the pan joint, which is then connected to the tilt joint, and finally, the roll joint is connected to the robot's head. This mechanism, as applied to a humanoid robot's neck, can provide a simple and effective solution for achieving the desired range of motion.

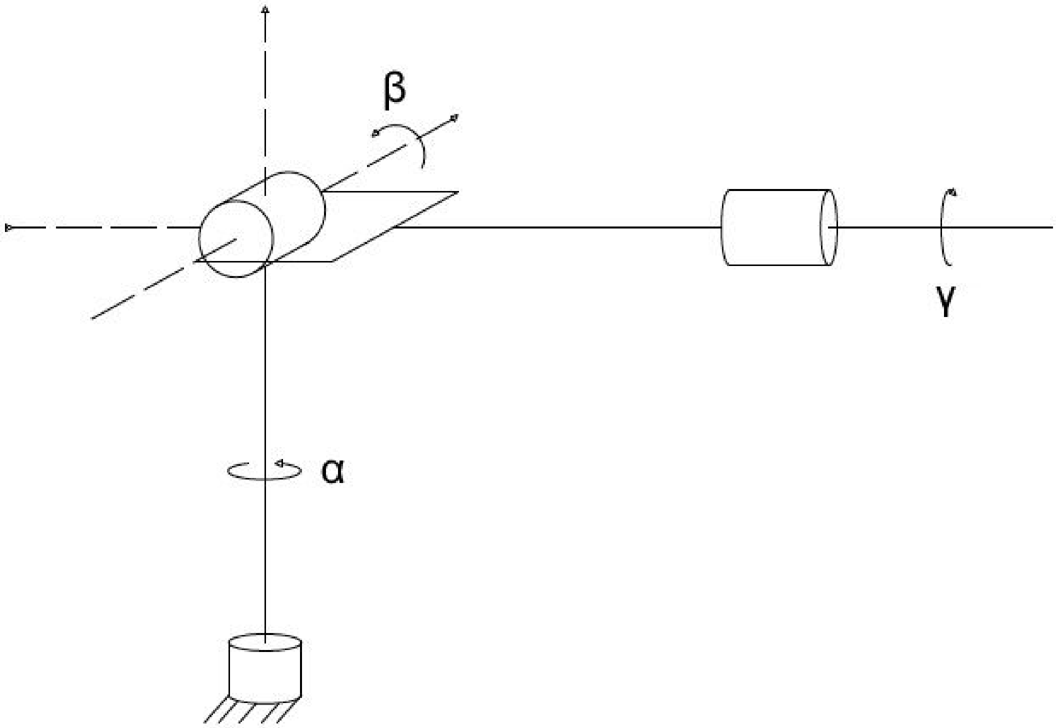


Figure 14 – Diagram of the PTR Mechanism [18]

Advantages:

- **Simplicity**: The PTR uses a combination of three rotary joints (or actuators), which simplifies the design, making it easier to construct, control, and maintain.
- **Compactness**: The use of rotary joints allows for a compact design, which helps to minimise the overall size and weight of the humanoid robot's head and neck structure.
- **Cost-effectiveness**: Due to its simplicity and the use of readily available components like servo motors or other rotary actuators, the Pan-Tilt-Roll mechanism can be a cost-effective solution for achieving the desired neck movement in a humanoid robot.

Downsides:

- **Speed**: The motors are heavy, and because this mechanism places them in series, each motor will have to cope with the increased inertia of the motors it is driving as well as the structure of the head.
- **Reduced load capacity**: The Pan-Tilt-Roll mechanism's compact design and use of rotary joints may limit the load capacity that the neck can handle, which could be a concern if the humanoid robot's head is heavy or contains additional components such as sensors or cameras.
- **Potential backlash**: Depending on the quality and type of actuators used, the Pan-Tilt-Roll mechanism may exhibit backlash or play, which can lead to less precise and less smooth movements.

3.2.3 Spherical Parallel Manipulator (3 DOF)

Coaxial Spherical Parallel Manipulators (CSPMs) are a special case of spherical parallel manipulators that provide three degrees-of-freedom (3-DOF) in pure rotational motion. These manipulators consist of a fixed base, a mobile platform, and parallel chains connecting the base and platform, with a collinear (coaxial) arrangement of their input axes [19]. This mechanism can be seen on the Reachy robot [20].

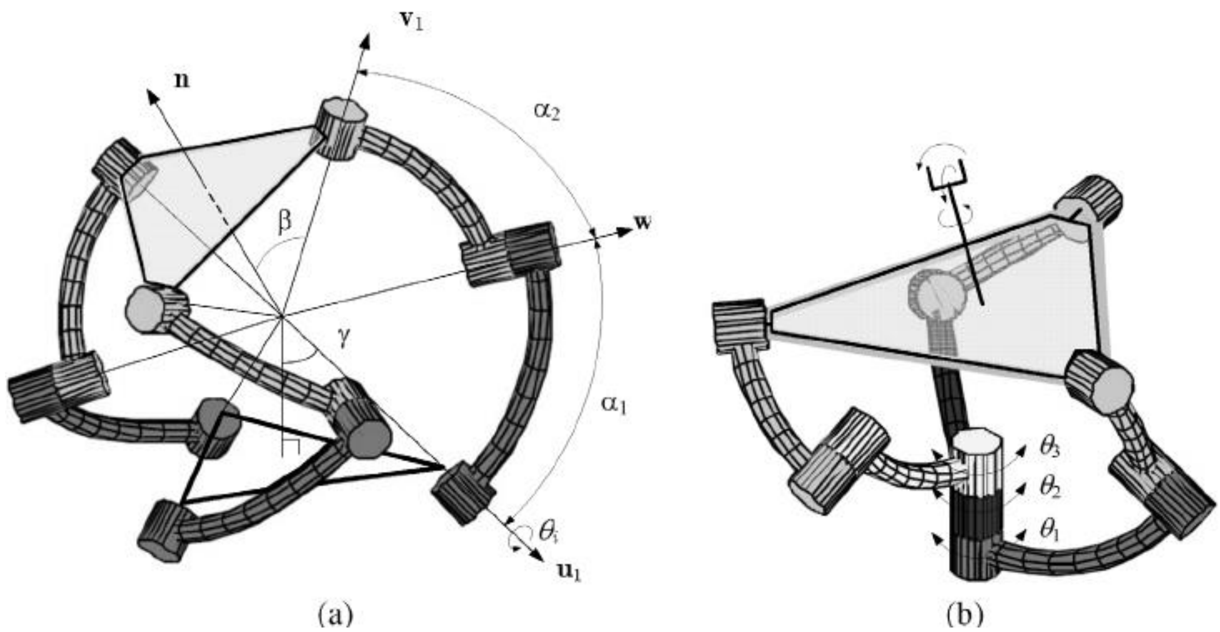


Figure 15 – Diagram of the CSPM Mechanism

Advantages:

- **Speed**: All the motors are mounted at the bottom of the mechanism, which would give the CSPM the advantage that the weight of the end effector would stay very low.
- **Design Simplicity**: The relative complexity of designing a system like this is very low.
- **Workspace optimisation**: Spherical Parallel Manipulators can be designed to optimise the workspace for specific applications, such as human-robot interactions or object manipulation, by adjusting the geometry and configuration of the mechanism.

Downsides:

- **Kinematic Complexity**: The parallel configuration of the mechanism, along with the multiple kinematic chains, can result in increased complexity in the control of the humanoid robot's head.
- **Strength**: The kinematic chains are relatively thin; therefore, the load capacity would be reduced.
- **Backlash**: The transmission of force must go through many links such as gears and bearings, this could give the system significant overall backlash if not correctly designed.

3.2.4 Oblique Swivel Joint

An Oblique Swivel Joint is a type of joint that allows two parts to rotate relative to each other around an axis that is neither parallel nor perpendicular to the primary axes of the connected parts [21]. In the context of a humanoid robot head, the Oblique Swivel Joint mechanism can be used to achieve three degrees of freedom for the neck, including pitch (flexion/extension), roll (lateral flexion), and yaw (rotation) movements. This is a unique mechanism that can be found on the F-35B VTOL swivelling nozzle [22].

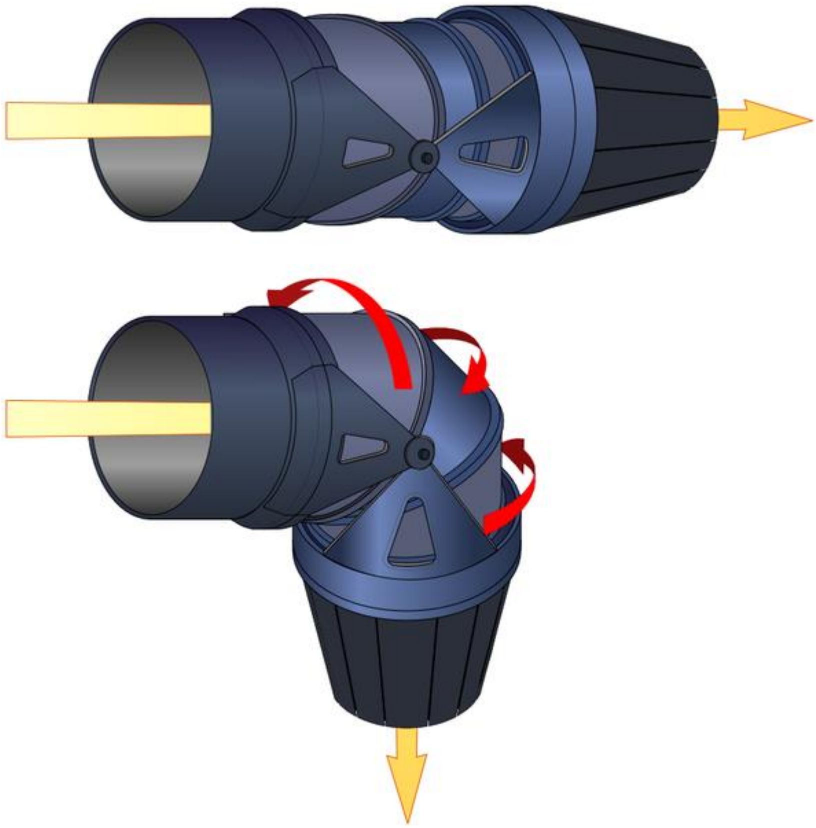


Figure 16 – Diagram of the OBJ Mechanism [23]

Advantages

- **Smooth Motion**: The Oblique Swivel Joint mechanism can provide a continuous and smooth range of motion, enabling more natural and fluid head movements.
- **Biomimicry & Uniqueness**: This is a unique design for a robot that can better mimic the human neck in the visual sense. The system is also quite unique, with only a limited number of similar example available.
- **Strength & Backlash Resistance**: The strength of this system would make it a very suitable candidate for the application of a neck.

Downsides

1. **Limited range of motion**: The Oblique Swivel Joint mechanism may not provide the same range of motion as other mechanisms, such as the Spherical Parallel Manipulator or the Pan-Tilt-Roll mechanism, potentially limiting the robot's ability to interact with its environment.
2. **Design Complexity**: There are very few working examples of this mechanism, and for good reason. The system would likely need a complicated arrangement of gears and joints to achieve the desired operation.

3.2.5 Neck Mechanism Comparison

After analysing the various neck mechanisms presented, and considering the requirements of the project, a ranking of the desired mechanisms is shown below. The mechanisms ranked below the top act as fallback mechanisms if the primary mechanism is deemed too difficult to pursue in the given time frame of the project:

1. **Oblique Swivel Joint**
2. **Coaxial Spherical Parallel Manipulator**
3. **Pan-Tilt-Roll Mechanism**

3.3 Eyes

3.3.1 Analysis of Eye Movement

The human eye is a highly complex and intricate organ that allows us to perceive the world around us through the detection of light and its conversion into electrical signals that are then processed by the brain. The human eye can move in 2 axes (each) of spherical motion to track objects and maintain visual stability.

There are six extraocular muscles, consisting of four rectus muscles (superior, inferior, lateral, and medial) and two oblique muscles (superior and inferior), control these movements. The rectus muscles move the eye up, down, left, and right, while the oblique muscles enable torsional rotation [24]. The three primary types of eye movements are saccades (rapid, jerky movements to bring an object into the fovea), smooth pursuit (steady tracking of a moving object), and vergence (simultaneous inward or outward rotation of both eyes to maintain focus on an object at different distances).

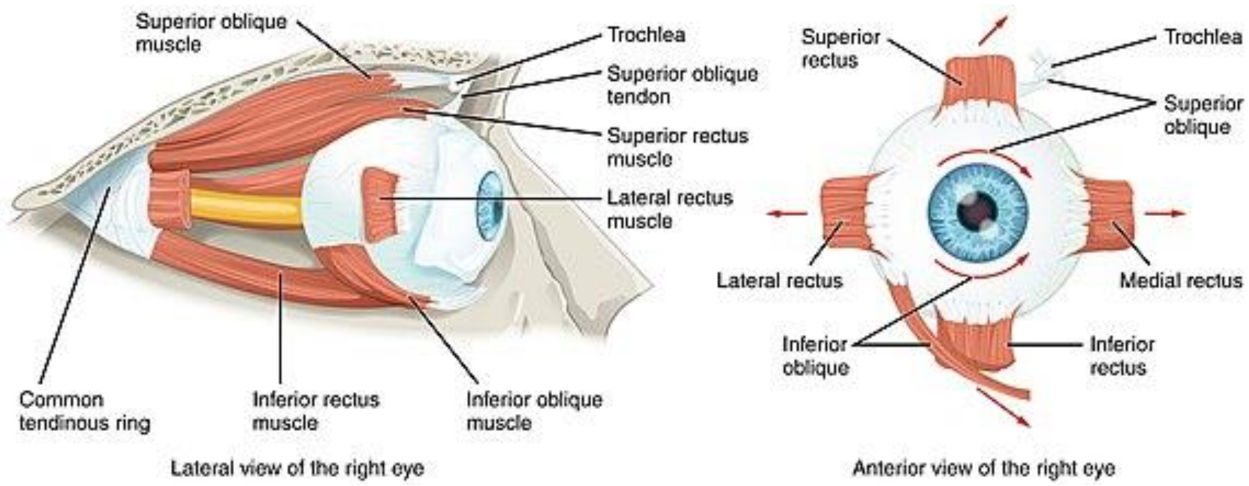


Figure 17 – Diagram of Eye Movement [25]

3.3.2 Simple Animatronic Eye (2 DOF)

This is a commonly used mechanism in animatronics to create eye movement, a variation can be seen for the iCub and InMoov robots. It operates by using a motor to deal with looking up and down and another motor connected in series to deal with the looking left to right. Overall, this is a very simple design with kinematics which are easy to understand.

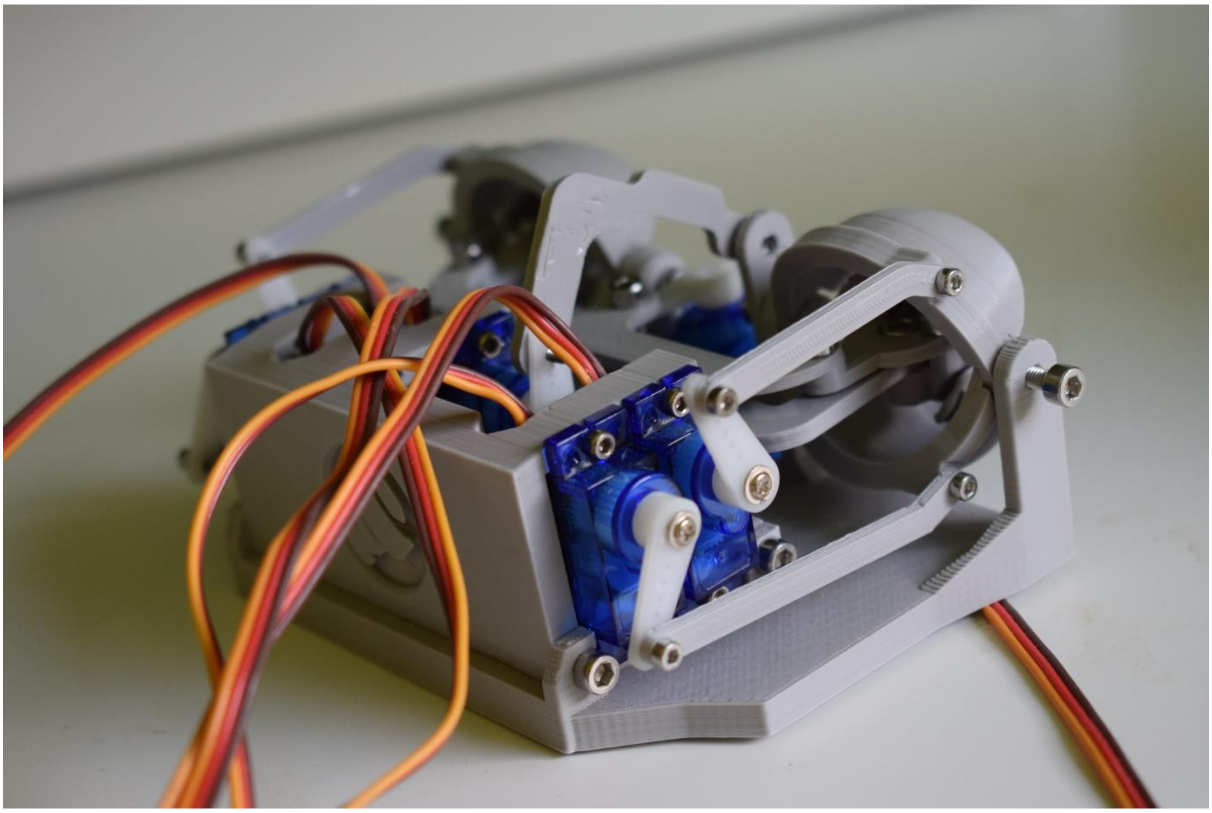


Figure 18 –Simple Animatronic Eye (Eyelids included) [26]

Advantages

- **Cheap**: This mechanism requires less motors and components overall.
- **Simple**: The reduces number of components, actuators and degrees of freedom make this mechanism extremely easy to develop.
- **Compact**: This mechanism generally doesn't take up much space, just watch out for the moving motor!

Downsides

- **Slow**: Because this is a series manipulator, the inertia from the driving motor will be slower than that of a parallel manipulator
- **Vision Obstruction**: There is potential for visual obstruction at extreme angles by various components in the design such as the motor.
- **Less Degrees of Freedom**: This mechanism cannot move the eyes independently of one another and therefore have less control of the positions of each eye with respect to one another.

3.3.3 Dual Spherical Parallel Manipulator (2x2 DOF)

This is a variation of the full 3DOF SPM mechanism, where there are now only 2 degrees of freedom for each eye. This mechanism has been developed in various ways such as the popular "Agile Eye" projects. This design has fixed motors to achieve the movement and is a very fast parallel manipulator mechanism.

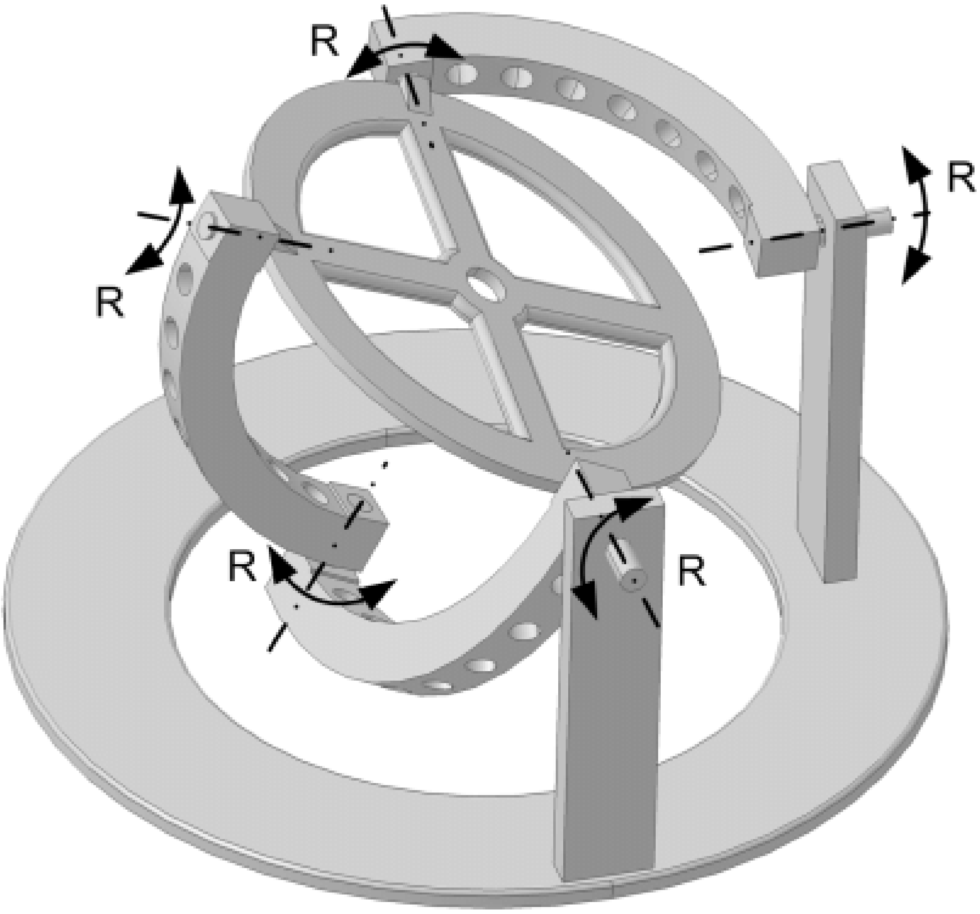


Figure 19 – Diagram of 2DOF Spherical Parallel Manipulator [27]

Advantages

- **Fast**: This mechanism allows the end effector to be mostly directly driven by the motors.
- **Full control**: Both eyes can move independently from each other.
- **Simple Design**: This mechanism is simple in a design perspective, and for the kinematics.

Downsides

- **Larger footprint**: The design of this mechanism inherently requires more space that the other mechanism described in this section.
- **Vision Obstruction**: There is a risk that at more extreme angles, the mechanism could obstruct the field of view of the camera.

3.3.4 Advanced Animatronic Eye (2x2 DOF)

This mechanism is based on the "Advanced Mechatronic Eye" mechanism developed by Will Cogley [28] for his animatronics project. Essentially the mechanism consists of three universal joints, all of which are constrained to the camera on one end, one is fixed on the other end by a supporting structure, and the other two are driven by a linear actuator, and positioned such that each linear actuator is responsible for its own axis of rotation.

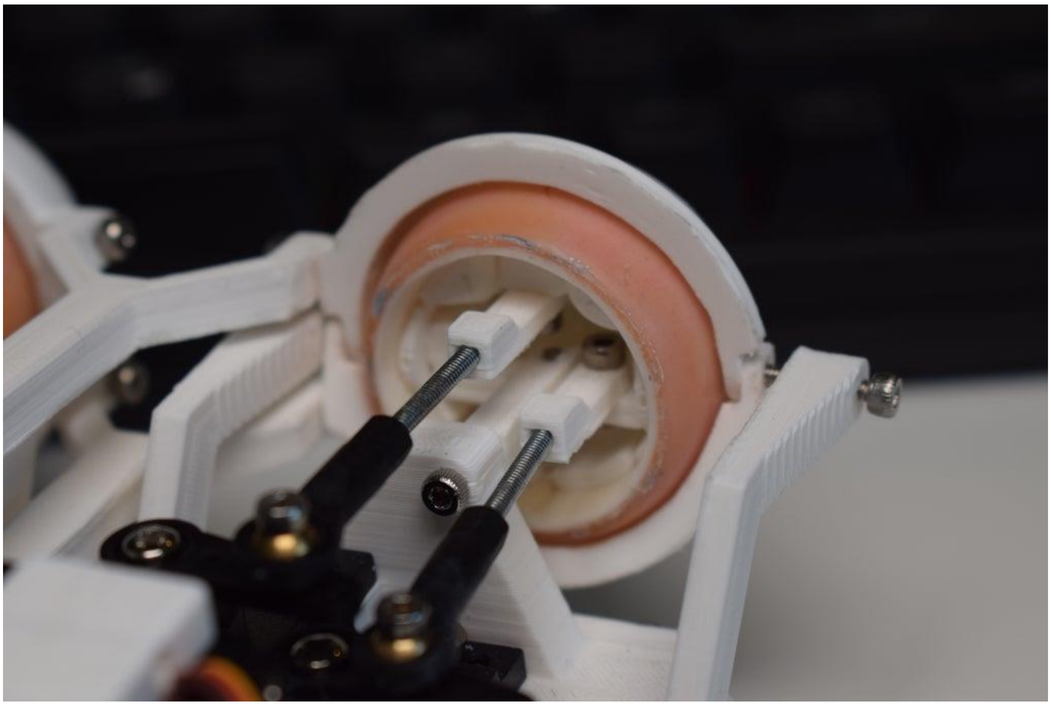


Figure 20 – Diagram of Advanced Mechatronic Eye Mechanism [28]

Advantages

- **Very Compact**: It doesn't get more compact than this, all motors in the same area to actuate the cameras with are positioned very closely.
- **Fast**: Parallel manipulators are always quick; this is no exception.
- **No risk of obstruction**: all the components are behind the cameras, therefore none of the components can obstruct the vision except for the other camera.
- **Full control**: full control of both vision systems allows for more possibilities.

Downsides

- **Complicated Kinematics**: The calculation for estimating the orientation of the end effector is difficult compared to the other designs.
- **Complex Design**: This design requires slightly more intricate design choices due to the compactness of the system.

3.3.5 Eye Mechanism Comparison

After analysing the various eye mechanisms presented, and considering the requirements of the project, a ranking of the desired mechanisms is shown below. The mechanisms ranked below the top act as fallback mechanisms if the primary mechanism is deemed too difficult to pursue in the given time frame of the project:

1. **Advanced Animatronic Eye**
2. **Simple Animatronic Eye**
3. **Spherical Parallel Manipulator**

4 Electronic Architecture

4.1 Sensor Implementation

4.1.1 Visual

Stereoscopic vision

To accurately mimic human vision, it is essential to have two cameras or "eyes" in the humanoid robot. This dual-camera setup allows the robot to perceive depth and distance, like human stereoscopic vision. By capturing images from two slightly different perspectives, the robot can use triangulation techniques to determine the distance of objects in its field of view, which is crucial for tasks such as object recognition, navigation, and manipulation.

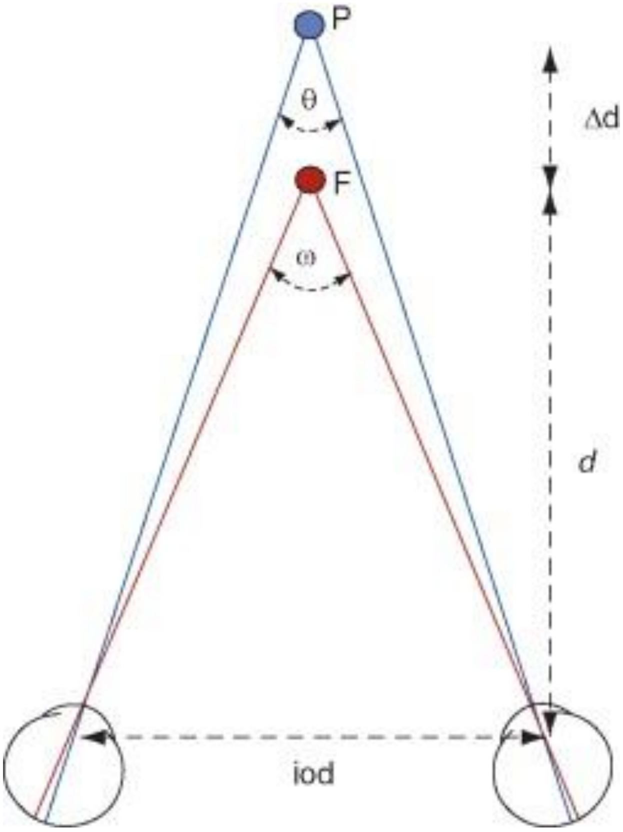


Figure 21 – Picture of Basic Principle of Stereoscopic Vision [29]

Form factor and interface for robotics applications

When selecting cameras for a humanoid robot, it is important to consider factors such as size, weight, and ease of integration. A small form factor is ideal to ensure the cameras can be easily incorporated into the robot's head without adding excessive bulk or hindering movement. Additionally, a simple and widely supported interface, such as USB, can facilitate the transfer of image data from the cameras to the robot's processing unit for real-time analysis.

Selected camera

The DFROBOT FIT0729 USB camera, featuring a high-quality Sony IMX179 sensor, is a good choice for this application. The camera's small form factor (38x38mm PCB size) makes it suitable for integration into the humanoid robot's eye mechanism, and its USB interface allows for straightforward connectivity to a separate processing unit. The Sony IMX179 sensor provides high-resolution images and excellent low-light performance, ensuring the robot can capture detailed and accurate visual information in various lighting conditions.

Figure 22 – Picture of Selected Camera (DFROBOT FIT0729) [30]

4.1.2 Audio

The inclusion of audio sensing in a humanoid robot offers various benefits, most notable for this project are the following:

- **Speech recognition**: By incorporating audio sensing, the robot can be equipped with speech recognition capabilities, allowing it to understand and respond to spoken commands or engage in conversations with humans. This feature significantly enhances the robot's user-friendliness and potential applications.
- **Audio localisation**: Audio localisation allows the robot to determine the direction and distance of a sound source, enabling it to identify and track objects or people based on the sounds they make. This functionality can be beneficial in various scenarios, such as locating a person in need of assistance or identifying the source of an alarm.

While equipping the robot with two microphones for binaural audio reception could potentially enhance its audio sensing capabilities, it was ultimately decided that this was not necessary for the project. The primary goal was to allow for basic speech recognition, and the DFROBOT FIT0729 camera already includes a built-in microphone that can be used for this purpose.

By using the integrated microphone in the DFROBOT FIT0729 camera, the humanoid robot can be equipped with basic speech recognition capabilities without the need for additional microphones. This simplifies the electronic architecture and conserves resources while still providing the desired audio sensing functionality.

4.2 Controller Selection

It is important to select a controller that aligns with the goals of the project, being op-source, accessible, and affordable [31]. For affordability and simplicity, the chosen controller should be a microcontroller capable of handling the 7 actuators as described in the Mechanism Selection section.

Various microcontrollers suitable for this application were explored, including Teensy boards, Raspberry Pi Pico, and Arduino. Each of these options offers different advantages, such as processing power, connectivity, and ease of use.

After a thorough analysis, the Arduino Uno emerged as the most appropriate choice for the project, primarily due to the following reasons:

- **Popularity**: Arduino Uno is a widely popular and commonly used microcontroller in various applications, making it easily accessible to users.
- **Affordability**: The Arduino Uno is a cost-effective option, making it an ideal choice for an affordable project.
- **Open-source nature**: Arduino Uno is an open-source platform with a vast community of users and resources, ensuring compatibility with other open-source components and easy integration into the project.
- **Capability**: The Arduino Uno can control the 7 actuators required for the project, ensuring smooth operation and functionality.

In conclusion, the Arduino Uno microcontroller aligns well with the project goals of being open-source, accessible, and affordable. Its popularity, cost-effectiveness, and capability make it the ideal choice for this humanoid robot application, ensuring a simple and efficient electronic architecture.

Figure 23 – Picture of Arduino Uno R3 Board [31]

4.3 Connectivity

While the head is often considered to contain the brain and should control the rest of the body, there is only a limited amount of space within the head to accommodate the eye mechanism and other controllers. Therefore, the microcontroller selected in the previous section must communicate with an external computer. Several methods can be used to send data to a microcontroller from an external computer, including USB, Wi-Fi, Bluetooth, and other wireless protocols.

Holding true to the project's goals of simplicity, accessibility, and affordability, the most appropriate way of communicating with the microcontroller is through a USB connection [32]. USB offers several advantages, making it the best choice for this project:

- **Simplicity**: USB connections are easy to set up and use, requiring minimal configuration and additional components.
- **Accessibility**: USB ports are available on almost every modern computer, ensuring that most users can easily connect their microcontroller to an external computer without the need for additional adapters or interfaces.
- **Reliability**: USB connections provide a stable and consistent means of communication between the microcontroller and the external computer, ensuring smooth data transfer and control.
- **Cost-effectiveness**: USB cables and connectors are widely available and affordable, further supporting the project's affordability goal.

5 Electrical Requirements

5.1 Actuators

5.1.1 Neck Mechanism

For the application of an oblique swivel joint, various types of actuators can be considered, such as servo motors, DC motors with encoders, and stepper motors. Each of these actuators offers distinct advantages and disadvantages, depending on the specific requirements of the project.

In summary, servo motors are usually easily controlled with a PWM driven and are typically very compact but offer limited range of motion. Stepper motors offer a holding torque and very precise positioning but are inefficient in power draw [33]. DC motors with an encoder attached can be made to go very fast but controlling them is complicated and very involved.

The final selection for this project is a NEMA 17 (42x42mm) stepper motor, chosen for its ability to accurately position itself regardless of the number of turns the motor has made (a requirement for the oblique swivel joint), its wide availability and affordability at a standard size, and the ease of control. Furthermore, stepper motors feature holding torque, allowing them to maintain their position even when not actively moving [34].

A CNC shield for the Arduino Uno can be used to connect the stepper motors, accommodating up to four stepper motor drivers. The A4988 stepper motor driver is an appropriate choice for this application due to its compatibility with the NEMA 17 stepper motor and Arduino Uno. The A4988 driver offers several advantages, including microstepping [35] capabilities for smoother and more precise motor control, adjustable current limiting, and over-temperature and over-current protection.

To tune the A4988 driver to the specifications of the stepper motor, a potentiometer is used. The potentiometer allows for fine adjustments to the motor's current limit, ensuring optimal performance and preventing damage due to excessive current. Tuning the A4988 driver is a critical step in setting up the stepper motor, as it ensures that the motor operates within its specified limits and achieves the desired accuracy and torque.

In summary, the NEMA 17 stepper motor, in conjunction with the Arduino Uno CNC shield and A4988 stepper motor driver, provides an ideal solution for the oblique swivel joint application. This combination of components meets the project's goals for accurate positioning, wide availability, affordability, and ease of control, making it a suitable choice for the humanoid robot's neck mechanism.

Figure 24 – Picture a NEMA 17 Stepper Motor with Flanged Coupler

5.1.2 Eye Mechanism

For the eye mechanism, the MG90 servo motor is an appropriate choice, considering the specific requirements of the application. The MG90 servo motor offers several advantages that make it suitable for this purpose:

- **Limited Rotation**: The eye mechanism does not require continuous rotation capabilities. The MG90 servo motor is designed to provide a controlled range of movement up to 180 degrees, which is sufficient for the desired eye movement in the humanoid robot.
- **Compact Size**: The MG90 servo motor is compact and lightweight, making it an ideal choice for the small form factor required by the eye mechanism. Its dimensions allow for easy integration into the robot's head without adding unnecessary bulk or weight.
- **Wide Availability**: The MG90 servo motor is widely available in various forms, making it an accessible option for hobbyists and professionals alike. Its popularity ensures that replacements and support are readily available, which contributes to the open-source nature of the project.
- **Cost-Effective**: The MG90 servo motor is extremely cost-effective, aligning with the project's goal of affordability. Its low price point enables the construction of a high-quality humanoid robot without breaking the budget.

To control the MG90 servo motors, a PCA9685 by Adafruit is used. This widely available servo motor driver allows for up to 16 servos to be connected at one time, making it a cost-effective and versatile solution for controlling multiple servo motors. The PCA9685 is compatible with the Arduino Uno, enabling seamless integration into the existing project architecture.

In conclusion, the MG90 servo motor is an excellent choice for the chosen eye mechanism due to its limited rotation capabilities, compact size, wide availability, and cost-effectiveness. The PCA9685 by Adafruit serves as a reliable and affordable solution for controlling the servo motors, ensuring smooth and precise eye movements in the humanoid robot.

Figure 25 – Picture of an MG90 Servo Motor [36]

5.2 Power Delivery

In the humanoid robot project, it is crucial to ensure that power is efficiently and safely delivered to the various systems, particularly the actuators. The Arduino Uno cannot handle high current or high-power applications directly, especially in a system with seven actuators, therefore, alternative methods of power delivery must be explored.

The stepper motor system requires a 12V DC power source, while the servo motor system with the PCA9685 requires a 5-6V power supply. To deliver power to both systems effectively and efficiently, a 12V 5A power supply, along with an LM2596 buck converter circuit, was chosen as the best solution.

The LM2596 buck converter is an appropriate choice for the following reasons:

- **Variable Output**: The LM2596 buck converter can efficiently step down the input voltage from the 12V power supply to the 5-6V required by the servo motor system. This variable output capability ensures that both the stepper and servo motor systems receive the appropriate voltage levels.
- **Efficiency**: Buck converters are generally more efficient than boost converters, as they can step down the input voltage without generating excessive heat. This increased efficiency results in lower power consumption, making the humanoid robot more energy-efficient and cost-effective [37].
- **Accessibility**: 12V power supplies are widely available, making them an affordable and accessible option for the project. The LM2596 buck converter is also readily available, ensuring compatibility with the project's open-source and accessibility goals.

In conclusion, the combination of a 12V 5A power supply and an LM2596 buck converter circuit is an effective, efficient, and accessible solution for delivering power to the humanoid robot's systems. This setup ensures that the stepper and servo motor systems receive the appropriate voltage levels, without overloading the Arduino Uno or compromising the project's goals of affordability and accessibility.

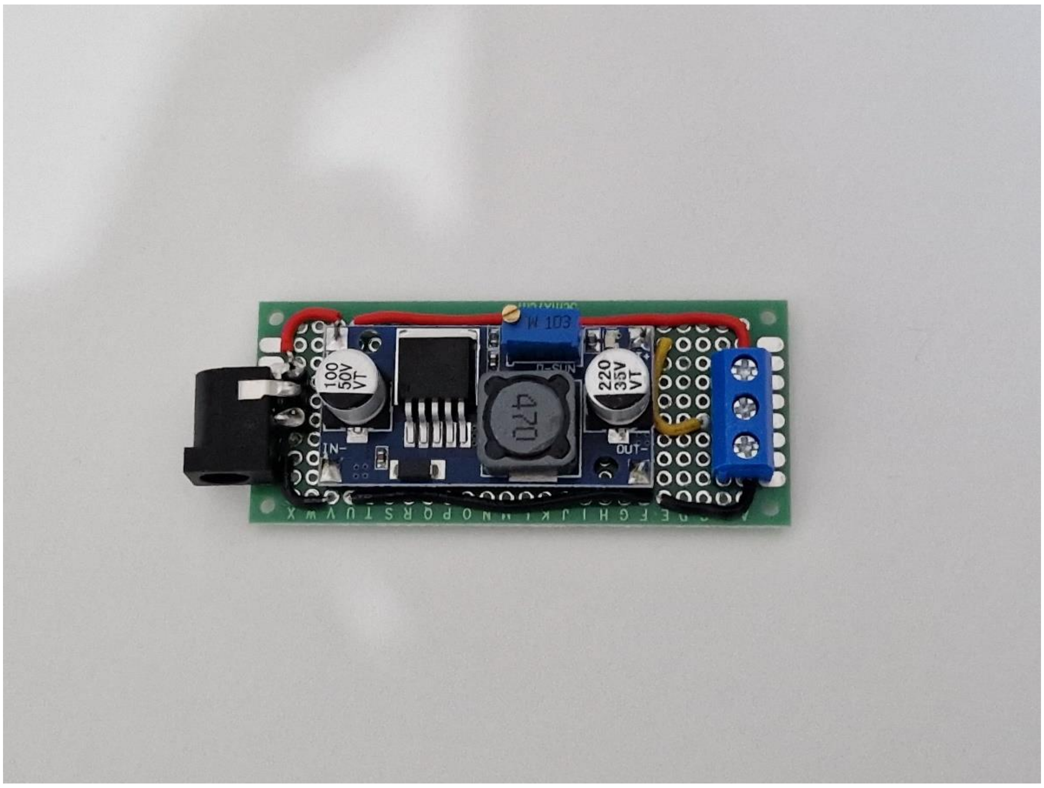


Figure 26 – Picture of the Power Supply Circuit

6 Mechanical Design

6.1 Suitable Manufacturing Methods

When designing a humanoid robot, it is essential to consider appropriate construction methods that are accessible to researchers and hobbyists. The goal is to ensure that the target audience can easily convert 3D CAD models into physical form using efficient and accurate methods.

3D Printing: One of the most accessible methods for creating physical components from 3D CAD models is 3D printing. By using a budget, readily available 3D printer, the barrier to entry is significantly reduced. PLA is chosen as the material for its strength, cost-effectiveness, and ease of use with 3D printers.

Fastening Methods: When assembling the humanoid robot, it is important to compare the use of threaded inserts and nuts as fastening methods. Embedded nuts are utilised for most of the project due to their lower profile, while threaded inserts are used where required for added strength and durability.

Fastener Sizes: In the project, different sizes of nuts and bolts are used depending on the specific application. M2 fasteners are employed for securing bearings, gears, and electronics in low-profile applications. M3 fasteners are utilised for higher strength applications, such as fastenings between fixed 3D printed parts. M4 bolts are used for threaded inserts, as they provide the best results when inserted into the plastic.

6.2 Neck Mechanism

6.2.1 1st Iteration: Oblique Swivel Joint Mechanism

6.2.1.1 Mechanism Architecture

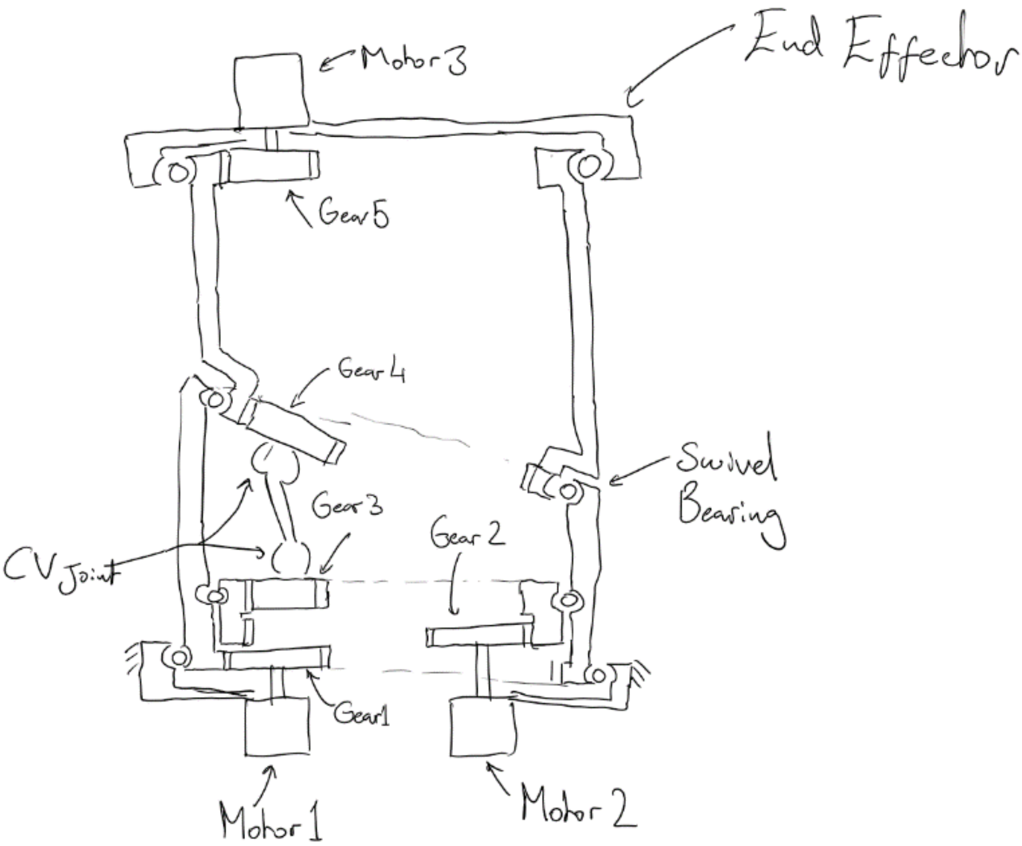


Figure 27 – Initial Oblique Swivel Joint Sketch

The Oblique swivel joint is designed to provide three degrees of freedom, mimicking the movement capabilities of a human neck. However, one issue that arises when performing movements like "Neck Circles" is the potential for wires connecting the motors to become twisted after multiple repetitions in the same direction.

To resolve this issue, the middle motor can be placed at the base of the mechanism, and a gearbox and shaft mechanism can be utilised to connect the motor and the joint without causing wire twisting. This arrangement results in two motors at the bottom of the mechanism and one at the top.

It is essential to ensure that there is sufficient space for wires (for power and communication) to be routed inside the neck mechanism, as this will enable a clean design and protect the wires from damage. To accomplish this, each motor is connected to its respective joint via an internal gear with a gear ratio of 2.5:1 and above. This design consideration allows for efficient wire routing while maintaining the desired movement capabilities of the Oblique swivel joint.

6.2.1.2 3D printed Bearing Design

The average diameter of a human neck is approximately 11.5 cm. To better replicate the appearance of a human neck and improve the structure's strength, the joints of the Oblique Swivel Joint should have a diameter approximating that of a human neck. Large diameter bearings are not commonly available, making them difficult to find or expensive. To address this issue, the development of a 3D printed bearing was conducted.

Plastic is weaker and more flexible compared to steel, which is typically used in bearings. To compensate for this, more plastic should be used to regain the stiffness of a traditional ball bearing. This can be done without using up more space by utilising the unused space within a ball bearing. This design takes advantage of the fact that a 3D printed raceway can be more complex than the simple raceway of a traditional ball bearing. Consequently, balls are placed one by one through a slot in one section of the bearing, as opposed to the traditional method of assembling ball bearings.

The balls used for the 3D printed bearing are 4.5mm steel BB balls, chosen for their affordability and the fact that they don't require extremely tight tolerances. This allows the design to be adjusted based on the size of the balls available. A thin 3D printed part, known as the separator, keeps the balls separated within the bearing.

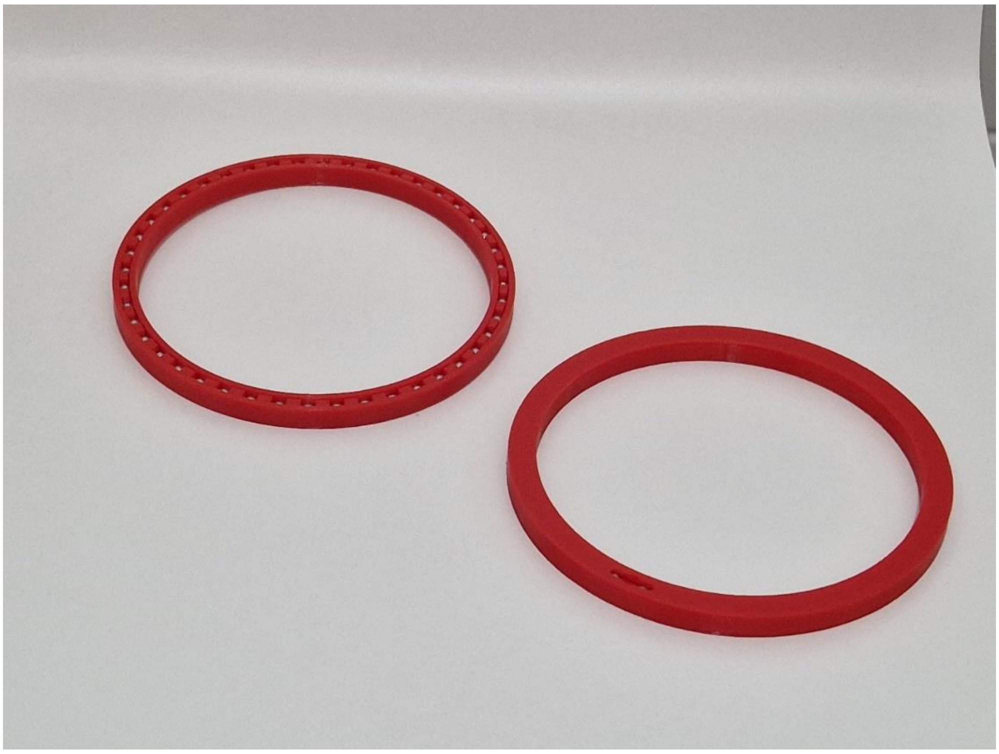


Figure 28 – Image of the 3D printed Bearings

6.2.1.3 Internal Gear Design

With the custom 3D printed bearings designed, it is possible to incorporate integrated gears to maintain a clean and aesthetically pleasing external appearance. To achieve this, an internal gear is added within the bearing. A space in the middle of the acting gears is necessary for wires to pass through, so a gear ratio of over 2.5:1 should be implemented.

There are several factors to consider when designing gears for this application. The modulus should be small to minimise backlash and because the system is not required to handle a large load. The goal is to design the outer border of the ring gear to fit the inner diameter of the bearing it is integrated into.

Autodesk Inventor's Design Accelerator was used to design these gears, as it includes a calculator for all the parameters of a gear system. This tool streamlines the design process and ensures the gears are accurately designed to meet the specific requirements of the Oblique Swivel Joint.

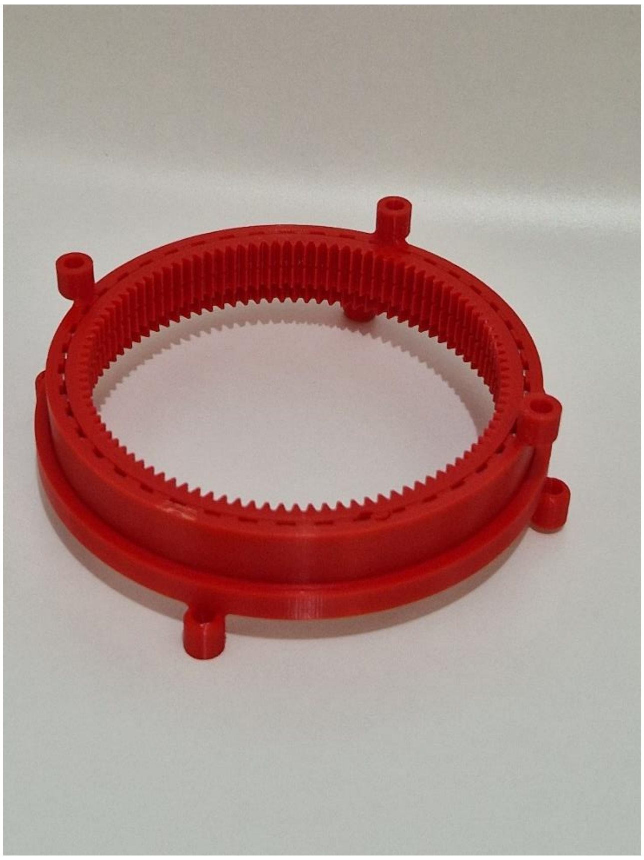


Figure 29 – Image of the 3D Printed Internal Gear Mechanism

6.2.1.4 Constant Velocity Joint

In addition to the bearings and gears, the new mechanism architecture described in previous sections necessitates the use of a constant velocity (CV) joint, also known as a Rzeppa joint, to connect the middle joint to the mechanism and the required motor. Similar methods of 3D printing were employed, and the same balls used in the 3D printed bearings were utilised for this joint.

The simpler universal joint could not be used in this application due to its effects on output rotation when at an angle. This would result in vibrations and less precise movements. In contrast, the CV joint maintains a constant rotational speed and smooth operation even at various angles, making it a more suitable choice for the Oblique Swivel Joint. This joint ensures accurate and smooth neck movements, contributing to the overall effectiveness and functionality of the humanoid robot head.

6.2.1.5 Conclusion

In conclusion, the first iteration of the Oblique Swivel Joint displayed significant progress in the design and implementation of various components, such as the bearings, gears, and constant velocity joint. However, despite overcoming several challenges, the middle joint posed issues with space constraints and complexity.

The system's intricacy and lack of space to address the problems with the middle joint limited its effectiveness and efficiency as a neck mechanism for the humanoid robot head. The application of this mechanism might be better suited for an arm, where more space is available within the link to generate torque in the joint.

To summarise, the first iteration of the Oblique Swivel Joint provided valuable insights and lessons for the design process, but its limitations and complexity ultimately called for a revaluation and further iterations to develop a more suitable neck mechanism for the humanoid robot head.

6.2.2 2nd Iteration: Spherical Parallel Manipulator

6.2.2.1 Design Architecture

In this iteration, the design will focus on the coaxial spherical parallel manipulator, a mechanism that positions all three motors at the base. A coaxial spherical parallel manipulator consists of three independent kinematic chains that connect the base to the end effector. Each chain has a swivelling arm driven by a motor at the base, and the orientation and position of the end effector are determined by the combination of the angles formed by the swivelling arms.

This type of mechanism offers several advantages, such as improved speed, compactness, and reduced design complexity compared to the previous iteration. Additionally, the coaxial arrangement of the motors simplifies the wiring and reduces the chances of wire entanglement.

The learnings from the previous iteration, particularly about bearings and gears, will not go to waste, as they can be applied to enhance and expedite the design process of the spherical parallel manipulator. For this design, a thruster bearing variety will be employed, as it allows all swivelling arms to be driven by an integrated internal gear, determining the end effector's position more efficiently and accurately.

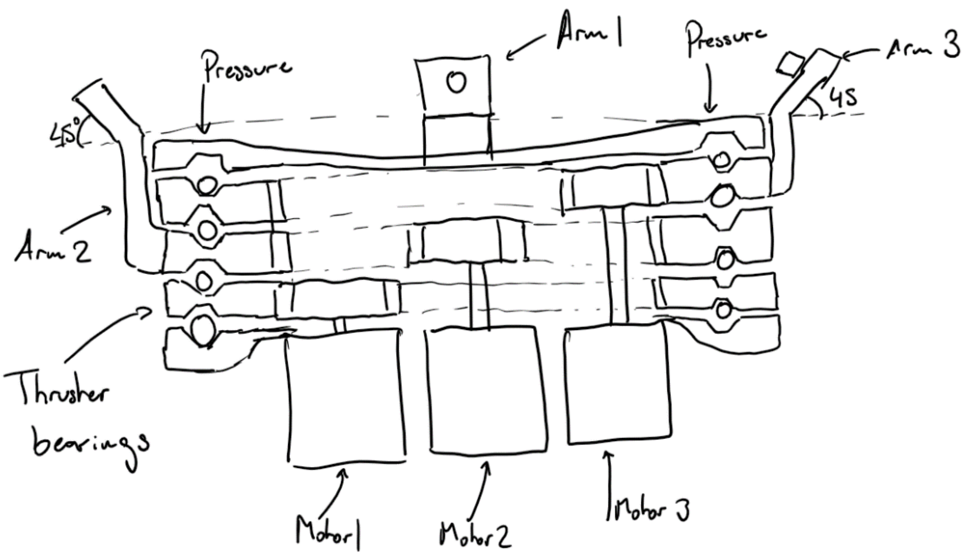


Figure 30 – Initial Coaxial Spherical Parallel Manipulator Sketch

6.2.2.2 Motor Mount

The motor mount is a crucial component in the spherical parallel manipulator design, serving as the foundation for securing the NEMA 17 motors in place. This part is carefully designed to ensure proper alignment and support for the motors while providing a versatile connection point for integrating the neck mechanism into the larger humanoid structure.

To accommodate various mounting configurations, M3 embedded nuts are strategically placed symmetrically around the motor mount, allowing for easy attachment to different parts of the humanoid. This flexibility is essential for adapting the neck mechanism to various project requirements and ensuring seamless integration with other components.

In addition to its functional role, the motor mount has been designed with structural considerations in mind. Ribs have been added at key locations to enhance the strength and rigidity of the part without adding unnecessary bulk. By maintaining a low profile, the motor mount contributes to the overall aesthetic appeal of the neck mechanism while ensuring reliable performance and durability.

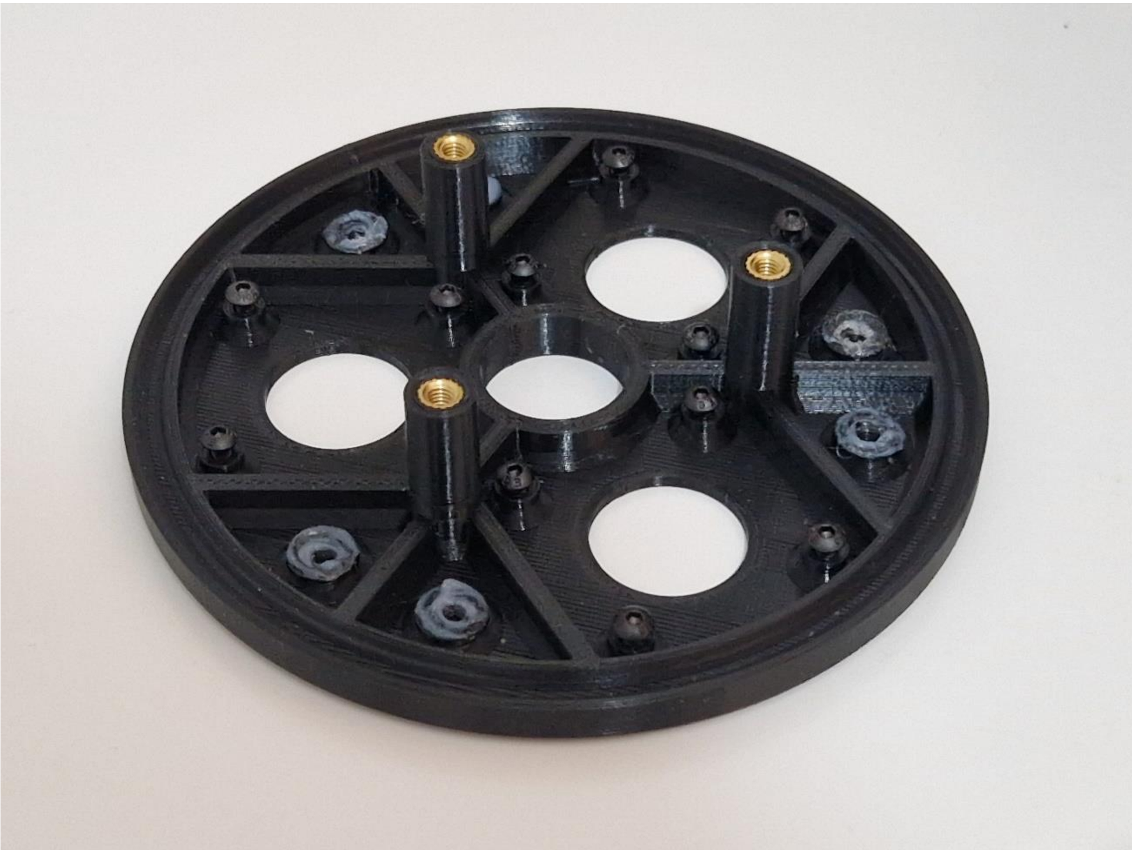


Figure 31 – Image of Motor Mount

6.2.2.3 Thruster Bearing Design

Thruster bearings, also known as axial or thrust bearings, are specifically designed to handle axial loads, which are loads applied parallel to the axis of the shaft. These bearings come in various types and configurations, such as ball, roller, or tapered designs, and are used in applications where high axial forces are present, including automotive, aerospace, and industrial machinery.

In the context of the spherical parallel manipulator, the 3D printed thruster bearings are designed to handle the forces exerted by the swivelling arms. The thruster bearing in this design features a 45-degree angular contact between the raceway and the bearing balls. This angle was chosen to address the potential overhang issue that could arise during 3D printing of the raceway. If the overhang were too steep, it would result in an undesirable contact surface for the ball bearings, leading to increased friction and compromised performance.

The 45-degree angle ensures a smooth contact surface for the ball bearings, reducing friction and allowing for efficient operation of the swivelling arms. The three driven thruster bearings are held in place by the motor mount base structure and a top bracket, which adds preload to the bearings. This preload increases the rigidity of the arms, ensuring stable and precise positioning of the end effector.

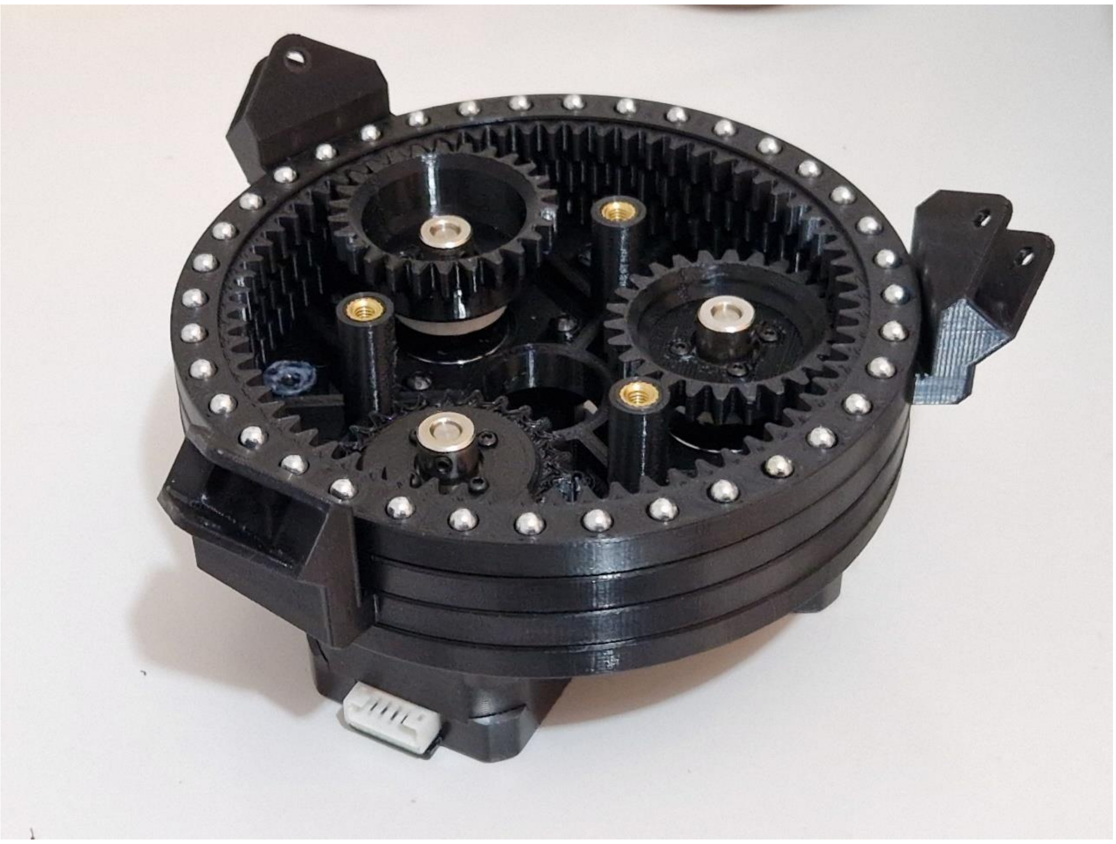


Figure 32 – Image of Thruster Bearing

6.2.2.4 Swivelling Arm Design

The swivelling arms play a crucial role in the functionality of the spherical parallel manipulator by connecting the motor mount base to the end effector, ultimately supporting, and manoeuvring the humanoid head. These arms must be designed to withstand the forces acting on the end effector, which include inertial forces resulting from head movement and external forces that may be encountered in various scenarios.

To optimise the performance of the neck mechanism, the swivelling arm design emphasises both strength and range of motion. The arm's topology is specifically engineered to maximise its movement capabilities, ensuring that the end effector can achieve a broad array of positions and orientations.

To achieve this, the arms are crafted from robust materials and feature a streamlined geometry that balances the need for rigidity and flexibility. The arms must maintain their structural integrity under various loads while allowing for smooth, precise movements in multiple directions. By optimising the swivelling arm design, the spherical parallel manipulator can effectively replicate the range of motion and responsiveness of a human neck, enhancing the overall realism and functionality of the humanoid.

Figure 33 – Image of Swivelling Arm

6.2.2.5 End Effector Mount Design

The end effector mount serves as the connection point between the swivelling arms and the humanoid head or skull. This component plays a vital role in ensuring that the head is securely attached to the neck mechanism while still allowing for a wide range of motion and precise movements.

To provide optimal support for the head, the end effector mount is designed to be highly structurally rigid. It features ribs on the bottom side, which increase the part's strength and rigidity without adding excessive weight to the overall system. By minimizing the end effector weight, the neck mechanism can operate more efficiently and respond more quickly to control inputs.

Embedded nuts are strategically placed symmetrically around the mount, allowing for easy installation of the head onto the neck mechanism. This design ensures that the head can be securely fastened to the end effector mount, providing a stable platform for the head while still enabling the spherical parallel manipulator to perform complex movements that closely mimic the motion of a human neck.

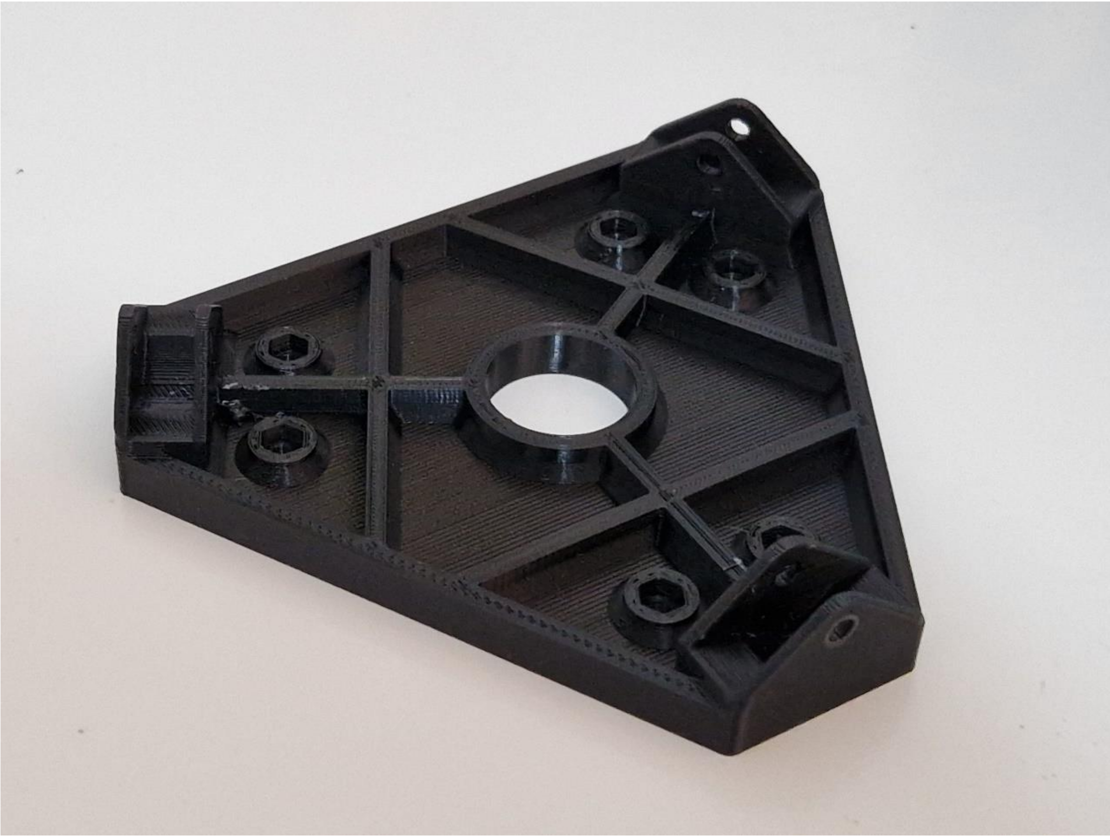


Figure 34 – Image of End Effector Mount

6.2.2.6 Conclusion

In conclusion, the design of the coaxial spherical parallel manipulator for the neck mechanism has successfully addressed the issues encountered in the previous iteration while achieving the desired range of motion and functionality. By incorporating 3D printed thruster bearings, a motor mount, swivelling arms, and an end effector mount, the mechanism is both robust and versatile.

The thruster bearings enable smooth and efficient motion, while the motor mount provides a secure platform for the NEMA 17 motors and allows for easy integration with the rest of the humanoid system. The swivelling arms are designed to be strong and lightweight, maximising the range of motion and minimising inertia. Finally, the end effector mount connects the head to the neck mechanism, ensuring a stable and secure attachment.

Overall, the coaxial spherical parallel manipulator demonstrates a well-thought-out and efficient design that meets the requirements of researchers and hobbyists. Its use of readily available components and 3D printed parts makes it accessible and affordable, making it a suitable solution for a humanoid neck mechanism.

Figure 35 – Image of Completed Neck Mechanism

6.3 Eye Mechanism

The chosen eye mechanism being explored can be effectively analysed as having three universal joints connected to each eye. One universal joint is fixed in space, while the other two are driven by their respective servo motors via a push rod. The two driven universal joints are positioned in such a way that one is vertical to the fixed universal joint, and the other is horizontal to the universal joint. This arrangement ensures that the driven universal joints do not affect each other in any way, and each is responsible for a separate axis of rotation.

In this application, a bush bearing between two plastic parts (3D printed PLA) provides sufficient tolerance and performance. These bush bearings can be designed to be held in place using a nut and bolt. The minimum number of supported bearings is 7.

The motor mount holds all the motors together and supports both eyes through two arms, each with a bush bearing on the end. This is also the location where the mounting holes connect to the head structure (described in the next section).

The eye mounts support two 'rotary rods', each supported by two bush bearings on either side. The rotary rods each have three holes, making them symmetrical and interchangeable.

The pushrod connectors are a bush bearing that mounts a bolt into a commonly found RC ball joint, driven by an arm from the servo motor. This design ensures smooth and precise control of the eye mechanism, allowing for realistic and responsive eye movements in the humanoid system.

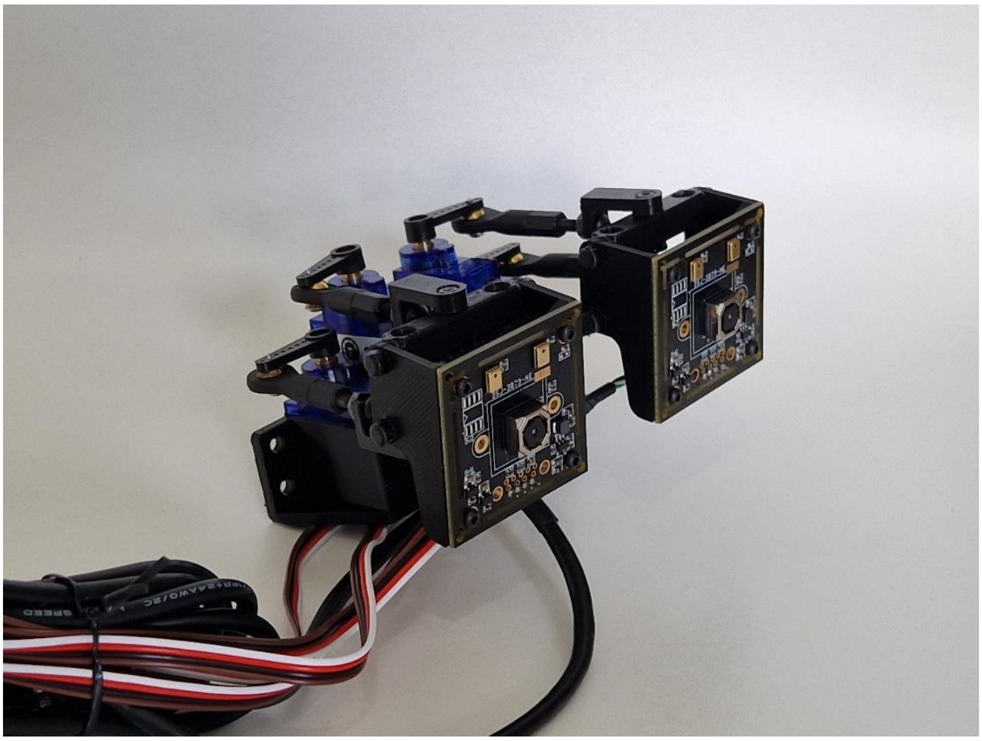


Figure 36 – Image of Completed Eye Mechanism

6.4 Head Design

The head design is a vital component in the overall appearance and functionality of the robot. Its primary functions include providing a visually appealing and approachable look, connecting to the neck end effector mount, securing the eye mechanism, and housing the control circuits.

Connection to the Neck

The head attaches to the neck through embedded nuts on the neck's end effector mount, utilising appropriately sized bolts. To ensure high strength and low weight, the base of the head features rib reinforcements.

Eye Mechanism Integration

The eyes mount onto the head using vertically installed embedded nuts. This design choice allows for easy installation and the flexibility to make design changes in the future. The positioning of the eyes closely mimics that of humans for a more natural appearance.

PCB Mounting and Accessibility

A separate 'PCB mount' component holds the PCBs and is secured to the tabs on the head's main skull part using nuts and bolts. This configuration situates the PCB on top of the eye mechanism, providing easy access for adjustments.

Aesthetics and Approachability

The robot's design should convey a gender-neutral and approachable appearance while ensuring that aesthetics do not limit the range of motion for any mechanism. The current design meets these criteria and even accommodates future expansions, such as dedicated microphones for ears. The head, sized to match an average human head, offers a simple, approachable, and accessible appearance that encourages positive interactions.

Expandability

The robotic head contains many additional mounts which enables other developers to mount their own hardware such as new stereo microphones or sensors.

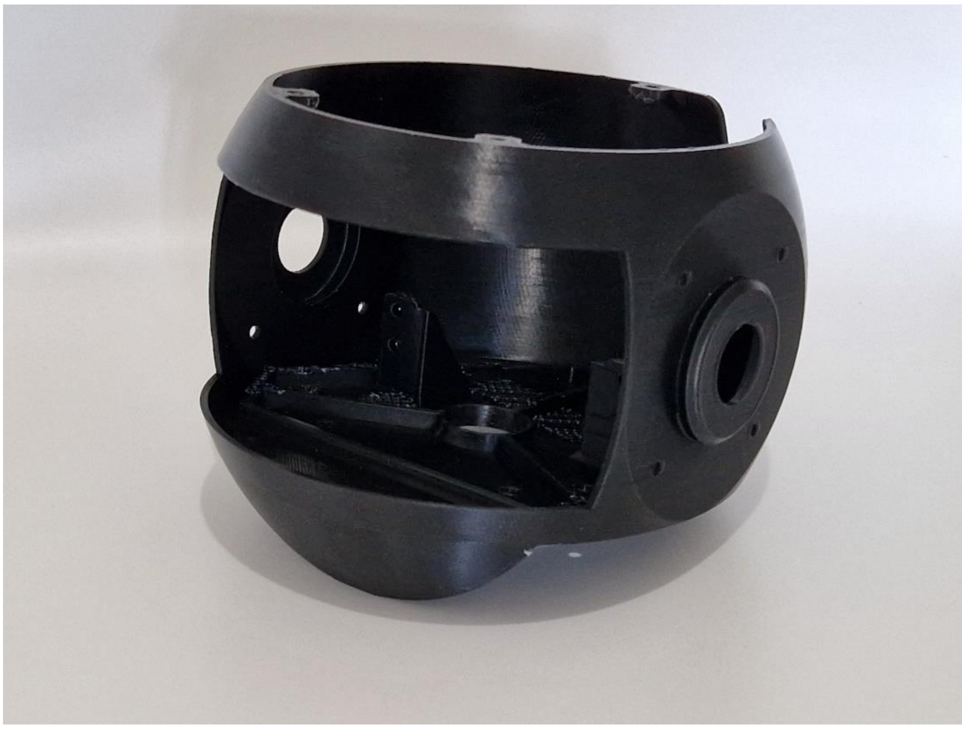


Figure 37 – Image of Completed Head Model

6.5 Base Design

The base is designed to support the head structure above the surface on which it is placed. With three points of contact, the base provides inherent stability. Since most of the head's weight is concentrated at the bottom due to the motors, the base's size does not need to significantly exceed the neck's diameter.

In addition to offering support, the base also houses the power circuit mentioned earlier. This design assumes that the power supply (battery) is in the robot's torso, which the head would ideally be mounted to. By integrating the power circuit within the base, the design offers a streamlined system.

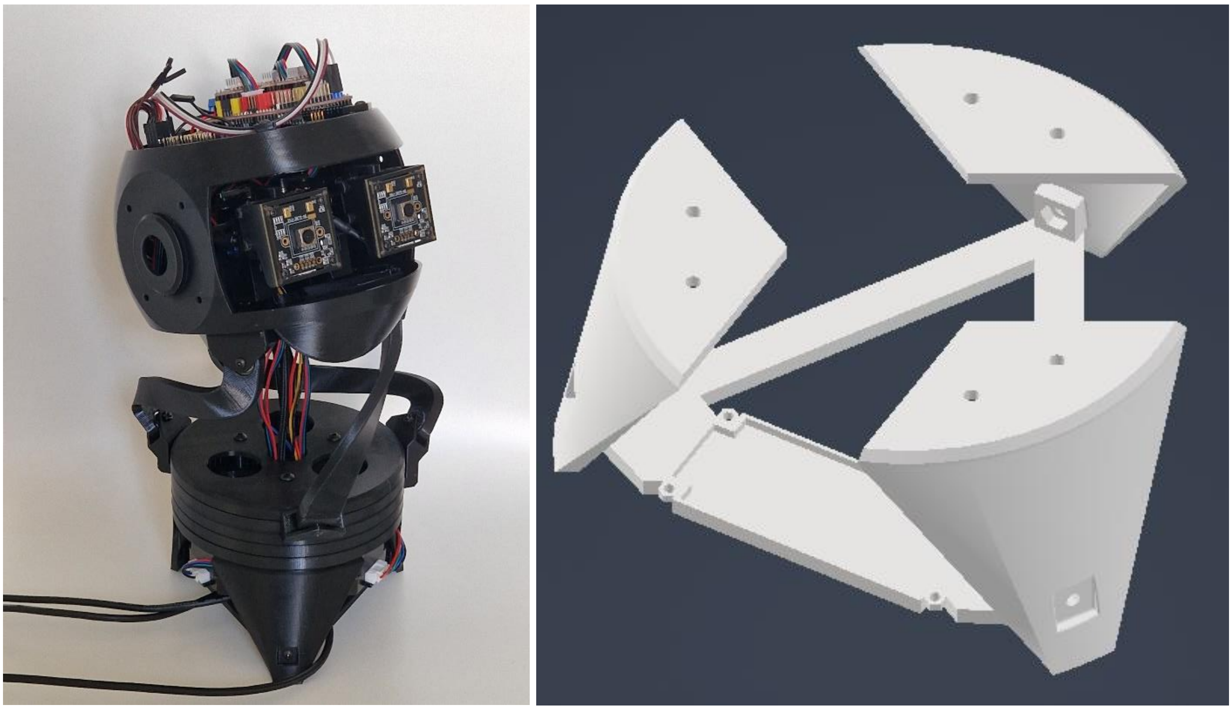


Figure 38 – Image of Completed Base

7 Control

7.1 Actuator Control

The stepper motor is controlled using the AccelStepper.h library, which contains various functions to facilitate the desired control of each stepper motor [38]. For the purposes of the project, the primary interest is in motor speed rather than position, as it is challenging to pinpoint the exact motor positions at startup without rotary encoders or other sensors. In Case 0, the stepper motor remains at a standstill; in Case 1, it moves forward (direction is arbitrary and depends on the stepper motor's connection to the driver); and in Case 2, it moves backward. This interaction allows educators to demonstrate how adjusting the rotational position of one or more stepper motors impacts the end effector's orientation.

The servo motors are controlled using the Adafruit_PWMServoDriver.h library, which is specific to the PCA 9685 servo driver board [39]. With this library, the desired position of each servo motor can be set. In Case 0, the servo motor assumes a neutral position (90 degrees); in Case 1, it moves backward to approximately 60 degrees; and in Case 2, it moves forward to around 120 degrees. This setup enables easy control of each eye's orientation in any desired direction.

7.2 Communication

Keyboard keys are read and encoded into a 16-bit integer. This integer is then decoded to provide the specified cases described in Section 7.1 Actuator Control.

To achieve this communication, several libraries are required:

- For the Python script, the `pyserial` library is needed to establish a serial communication between the computer and the Arduino. It allows sending and receiving data through the serial port.
- The `keyboard` library in Python is used to detect and process the keyboard input from the user. It enables the script to identify which keys are being pressed and subsequently encode the corresponding commands.

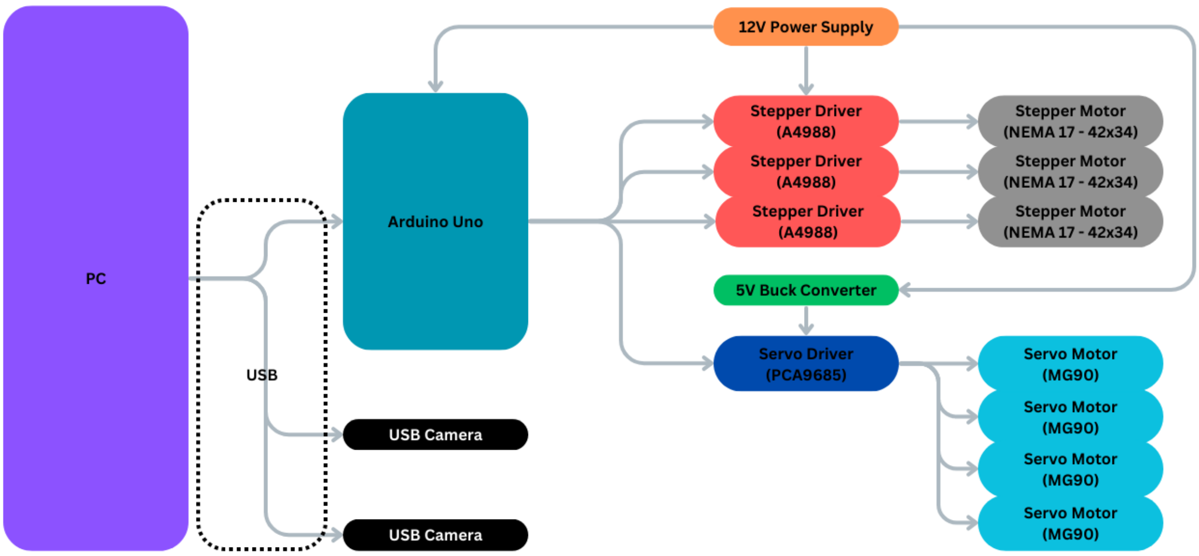


Figure 39 – Overall Architecture of the Systems and Electronics

8 Additional Considerations

8.1 Research Platform

In addition to the features and functionality already described, there are several other aspects of this project that can enhance its purpose as a research and educational platform:

- **Modularity**: The design of the robot's head, neck, and eye mechanisms are modular, allowing for easy customisation and modification. This enables educators and researchers to adapt the platform to specific needs, swap out components, or test new configurations without extensive redesign efforts.
- **Open-source**: By keeping the project open-source, the platform encourages collaboration and innovation. Students, educators, and researchers can access and contribute to the project, promoting a community-driven approach to advancement in robotics.
- **Scalability**: The platform is designed with scalability in mind, enabling users to extend the system with additional sensors, actuators, or other components. This flexibility allows the platform to grow and evolve with advancements in robotics and AI technology.
- **Documentation**: Providing comprehensive documentation and tutorials on the project can significantly contribute to its educational value. This will enable users to understand the underlying principles, design choices, and implementation details, helping them learn and experiment more effectively.

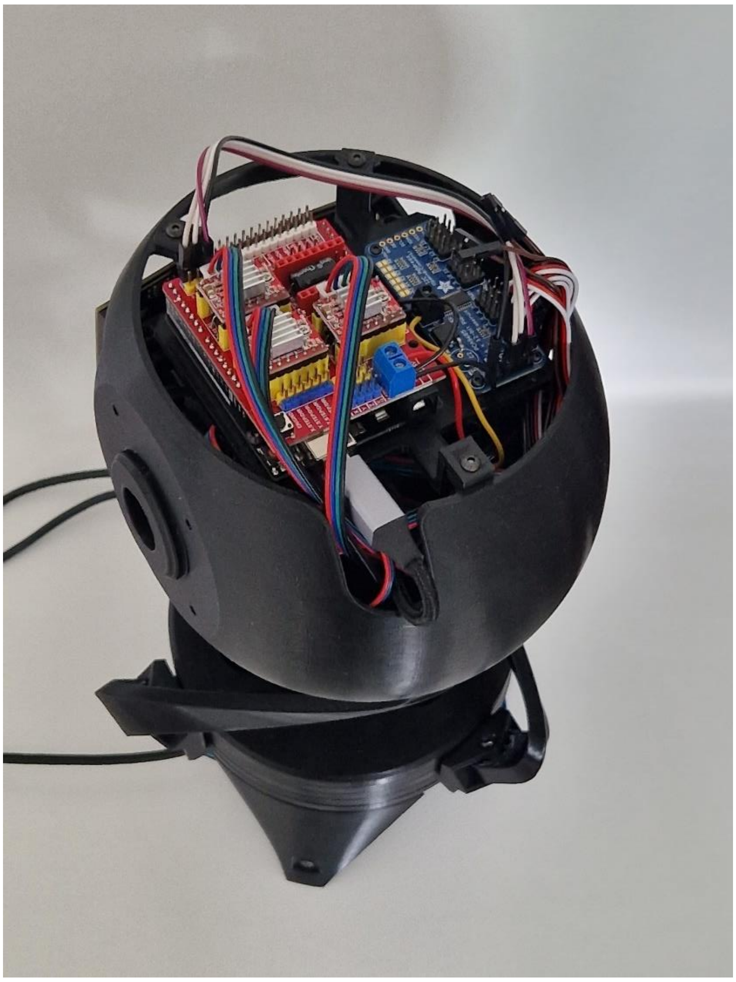


Figure 40 – Rear of Completed Model

8.2 Accessible and Affordable

One of the main objectives of this project was to create a robotic platform that is both accessible and affordable, aiming for a total cost of around £200-£250. The measures taken to achieve this goal include:

- **Cost-effective Actuators**: By utilising affordable actuators, the project significantly reduces overall costs. The chosen actuators still provide a good level of performance and reliability while maintaining a budget-friendly price point of approximately £80.
- **Electronics**: The electronic components, including the Arduino, servo driver board, and other essential components, were selected to minimise costs without compromising functionality. This resulted in a cost of around £60 for the electronic parts.
- **Sensors**: The sensors used in the project were chosen based on their cost-effectiveness, reliability, and availability. By selecting affordable yet reliable sensors, the project ensures the desired functionality while keeping the costs down to approximately £80.
- **3D Printed Parts**: Utilising 3D printing to create structural and functional components significantly contributed to the project's affordability. 3D printing enables rapid prototyping and manufacturing at a low cost; the 3D printed parts for this project costing around £20.

Part Name	Qty	Unit Price	Total Price
LM2596	1	£ 1.66	£ 1.66
NEMA 17 motor (28mm)	3	£ 13.50	£ 40.50
Flanged Shaft Coupler	3	£ 4.00	£ 12.00
696ZZ Ball bearing (6x15x5mm)	6	£ 1.37	£ 8.24
MG90 servo Motor	4	£ 3.91	£ 15.64
USB Camera	2	£ 38.62	£ 77.24
PCA9685	1	£ 7.69	£ 7.69
Arduino Uno	1	£ 23.48	£ 23.48
CNC Arduino Shield	1	£ 5.99	£ 5.99
A4988	3	£ 6.49	£ 19.47
3D printed Parts	0.8	£ 20.00	£ 16.00
		Total	£ 227.90

Table 2 - Bill of Materials

8.3 Open-Sourcing

Open-sourcing is a critical aspect of this project, as it aims to make the robotic platform available to a wider audience, encouraging innovation, collaboration, and learning. A GitHub repository was created to store all the necessary files for the project under nrwh/BEATRIX_Head (github.com). The following files have been uploaded to make this an effective open-source project:

- **Design Files**: All the design files for the 3D printed parts, electrical schematics, and other components are made available in accessible formats. This allows users to download, modify, and share these files freely, encouraging collaboration and customisation.
- **Source Code**: The source code for the Arduino microcontroller and external computer are released with proper commenting and documentation on how to use it.
- **Documentation**: Guides on how to assemble the system are detailed as well as reasoning for the chosen components.

9 Completed Model

The completed model, as described throughout this report, is an innovative and cost-effective robotic head designed for research and educational purposes. The design incorporates a neck mechanism based on a spherical parallel manipulator, eye mechanism driven by servo motors, an aesthetically pleasing and functional head design, and a stable base to support the entire structure. The project is open-source and accessible, promoting collaboration and knowledge sharing.

However, there are some limitations to the current design:

- **Slow Stepper Motors**: The Arduino Uno limits the speed at which a stepper motor can be run at to 4,000 steps per second. Due to the use of 1/16th micro-stepping for lower noise and smoother movement, this speed may not be sufficient for all applications. Upgrading to a faster microprocessor would enhance the performance of the stepper motors.

$$Revolutions\ per\ Second = \frac{\frac{Max\ Steps\ per\ Second}{Number\ of\ Motors} \times Microstepping}{Number\ of\ Steps}$$

Equation 1

$$Revolutions\ per\ Second = \frac{\frac{4000}{3} \times \frac{1}{16}}{300} \approx 0.3$$

- **Weak Spherical Parallel Manipulator Swivelling Arms**: The current design of the swivelling arms may not provide the optimal strength and rigidity needed for the neck mechanism. Exploring alternative materials or design topologies could lead to improvements in the performance and durability of the neck mechanism.
- **Hot Stepper Motors**: Stepper motors are known to generate heat during operation, and this heat weakens the plastic components onto which they are mounted. Investigating alternative mounting materials or incorporating heat dissipation solutions will help mitigate this issue and extend the lifespan of the robotic head.

Despite these limitations, the completed model represents a significant achievement in the development of an affordable and accessible robotic head platform. Future improvements and refinements to the design, driven by further research to enhance the capabilities of this robotic platform, making it an even more valuable resource for education and research in robotics.

10 Conclusion

In conclusion, this project successfully delivered a functional and cost-effective robotic head, designed to serve as a research and educational platform in the field of robotics. By focusing on affordability, accessibility, and open-sourcing, the project aims to promote collaboration, knowledge sharing, and inspire further innovation in robotics research and education.

The robotic head features a neck mechanism based on a spherical parallel manipulator actuated by stepper motors, a responsive and compact eye mechanism driven by servo motors, a visually appealing and functional head design, and a stable base to support the entire structure. The design process involved careful consideration of various factors, such as the choice of materials, manufacturing techniques, and control algorithms, to optimise performance and maintain the project's budget constraints.

Despite some limitations, such as slow stepper motors, weak swivelling arms, and heat dissipation concerns, the completed model serves as a solid foundation for future improvements and refinements. Ongoing research and user feedback will drive the development of even more advanced and capable robotic platforms.

Overall, this robotic head project represents a significant contribution to the field of robotics at the University of Bath, providing an affordable, accessible, and versatile platform for researchers, educators, and enthusiasts alike. By continuing to build on this foundation and sharing knowledge openly, the robotics community can work together to advance the understanding and application of robotics in a variety of settings and industries.

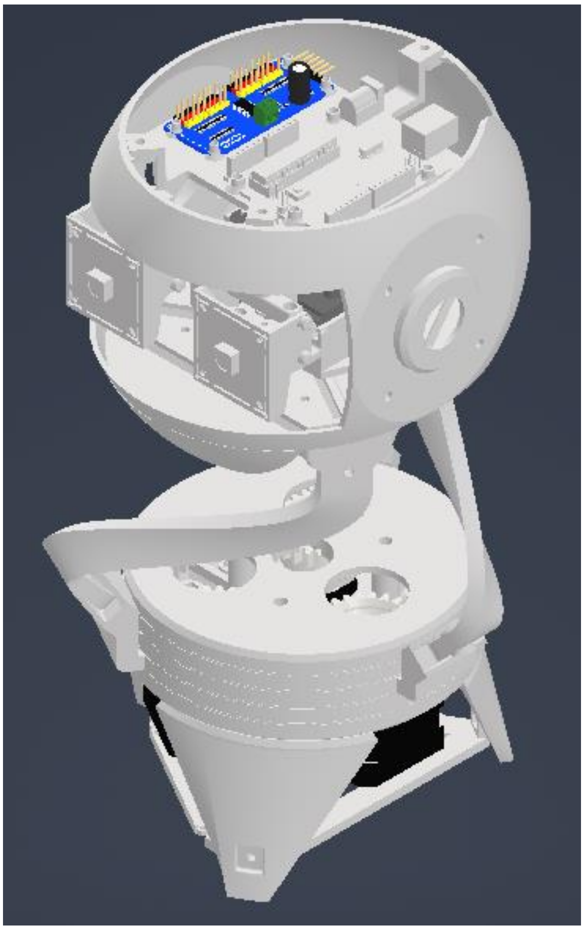


Figure 41 – Picture of the Final Head Design CAD

11 Further Work

11.1 Enhanced Neck Mechanism

As previously discussed in the limitations of the neck mechanism, here are the following ways in which the neck mechanism can be further enhanced:

- **Strengthened Arms**: The swivelling arms of the neck mechanism currently face challenges in terms of strength and durability. Exploring alternative materials, such as high-strength plastics, composites, or even lightweight metals, can significantly increase the strength and durability of the arms. Additionally, optimising the topology and design of the arms can lead to improved performance and reduced stress concentrations.
- **Heat Management**: The contact between the stepper motors to the plastic components of the neck mechanism poses a risk of heat-related damage or deformation over time. Implementing design modifications that allow for better heat dissipation, such as adding heat sinks or air vents, can help mitigate this risk. Alternatively, active cooling solutions, like fans, can be considered to maintain the temperature of the motors within safe limits.
- **Protected wires**: The wires are exposed to the harsh moving teeth of the internal gears in the gearbox. Creating shield to protect the wires would be a worthwhile upgrade to the system.

11.2 Sensors and Advanced Control

Incorporating additional sensors and implementing advanced control algorithms can significantly improve the positioning performance and overall capabilities of the robotic head. Some possible enhancements include:

- **Encoders**: Adding rotary encoders to the stepper motors can provide real-time feedback on their position and movement, enabling closed-loop control and more accurate positioning. This can also help in compensating for any backlash, slippage, or misalignment that may occur during operation.
- **Inertial Measurement Units** (IMUs): Integrating IMUs, such as accelerometers, gyroscopes, and magnetometers, can provide valuable information about the orientation and movement of the robotic head. This data can be used to enhance the control algorithms, enabling more precise and stable motion, as well as aiding in calibration and error correction.
- **Force/Torque Sensors**: Implementing force or torque sensors at the joints or on the end effector can enable the robotic head to sense and respond to external forces. This can be particularly useful for applications that involve interaction with objects or environments, such as robotic manipulation or haptic feedback.
- **Advanced Control Algorithms**: Leveraging the data collected from the various sensors, advanced control algorithms can be developed to optimize the performance of the robotic head. Techniques such as model predictive control, adaptive control, or machine learning-based approaches can be employed to enhance the accuracy, stability, and adaptability of the system.

11.3 ROS

The Robot Operating System (ROS) is a flexible, open-source framework that provides a wide range of tools, libraries, and software packages for developing and controlling robots [40]. Integrating ROS into the robotic head design can bring numerous benefits, including:

- **Modularity**: ROS follows a modular architecture, which enables the separation of the robot's various functionalities into independent nodes. This modularity simplifies development, maintenance, and upgrading of individual components, as well as allowing for easier integration of new sensors or actuators.
- **Extensibility**: The ROS ecosystem comprises a vast array of packages and tools developed by researchers and developers worldwide. By leveraging these resources, the robotic head can be easily extended with new capabilities, such as perception, planning, or navigation, without the need for developing these functionalities from scratch.

- **Standardisation**: ROS provides standardised interfaces for communication between nodes, simplifying the integration of various sensors, actuators, and control algorithms. This standardisation can also improve the reusability of code and facilitate collaboration between developers.
- **Scalability**: As ROS can run on multiple machines and support distributed computing, it allows for easy scaling of the robotic system. This means that the robotic head can be easily integrated into a larger robotic system or connected to external devices, such as other robots, computers, or cloud services.

References

[1] D. P. Shields, “Final Year Project Handbook,” University of Bath, 2023. [Online]. [Accessed 14 February 2023].

[2] Z. A. Barakeh, S. Alkork, A. S. Karar, S. Said and T. Beyrouthy, “Pepper Humanoid Robot as a Service Robot: a Customer Approach,” in *2019 3rd International Conference on Bio-engineering for Smart Technologies (BioSMART)*, 2019.

[3] Universal Robots, “HUMANOID ROBOTS - APPLICATIONS AND FUTURE SCOPE,” Universal Robots, 24 May 2019. [Online]. Available: https://www.universal-robots.com/in/blog/humanoid-robots-applications-and-future-scope/#:~:text=Most%20of%20the%20humanoid%20robots%20are%20used%20in,the%20medically%20ill%20elderly%20patients.%20Education%20and%20entertainment. [Accessed 11 May 2023].

[4] G. R, “Robot Asimo : Honda met fin à son développement,” objetconnecte.com, 29 June 2018. [Online]. Available: https://www.objetconnecte.com/robot-asimo-honda/. [Accessed 8 May 2023].

[5] G. Langevin, “Home Page,” InMoov, [Online]. Available: https://inmoov.fr/. [Accessed 11 May 2023].

[6] N. O. Valencia, L. F. V. Londoño, M. A. Jinete and R. Jiménez, “Movement detection for object tracking applied to the InMoov robot head,” in *2016 XXI Symposium on Signal Processing, Images and Artificial Vision (STSIVA)*, 2016, pp. 1-6.

[7] “Gael Langevin and Inmoov an interview,” Robots and Androids, [Online]. Available: http://www.robots-and-androids.com/Gael-Langevin-and-InMoov.html. [Accessed 11 May 2023].

[8] Poppy Station, “Poppy Robot Home Page,” [Online]. Available: https://www.poppy-project.org/en/. [Accessed 11 May 2023].

[9] M. Lapeyre, P. Rouanet, J. Grizou, S. N'Guyen, A. Le Falher, F. Depraetre and P.-Y. Oudeyer, “Poppy: Open source 3D printed robot for experiments in developmental robotics,” in *4th International Conference on Development and Learning and on Epigenetic Robotics*, 2014, pp. 173-174.

[10] Pollen Robotics, “Pollen Robotics Home Page,” Pollen Robotics, [Online]. Available: https://www.pollen-robotics.com/. [Accessed 11 May 2023].

[11] R. Beira, M. Lopes, M. Praca, A. Bernardino, G. Metta, F. Becchi and R. Saltaren, “Design of the robot-cub (iCub) head,” in *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.*, 2006, pp. 94-100.

[12] G. Gibaldi, A. Canessa, M. Chessa, S. P. Sabatini and F. Solari, “A neuromorphic control module for real-time vergence eye movements on the iCub robot head,” in *2011 11th IEEE-RAS International Conference on Humanoid Robots*, 2011, pp. 543-550.

[13] Istituto Italliano di Technologia iCub, “Home Page,” IIT, [Online]. Available: https://icub.iit.it/. [Accessed 11 May 2023].

[14] J. Searle, “Muscles of the Head and Neck,” 30 April 2015. [Online]. Available: https://jeffsearle.blogspot.com/2015/04/muscles-of-head-and-neck.html. [Accessed 11 May 2023].

[15] M. Jeung, G. Lee and Y. Oh, “Realization of human neck motion with novel robotic mechanism,” in *2016 13th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI)*, 2016, pp. 482-486.

[16] I. Lascurain-Aguirrebeña, D. J. Newham, X. Casado-Zumeta, A. Lertxundi and D. J. Critchley, "Immediate effects of cervical mobilisations on neck muscle activity during active neck movements in patients with non-specific neck pain. A double blind placebo controlled trial," *Physio Therapy,* vol. 110, no. 0031-9406, pp. P42-53, 2019.

[17] J. Wei, M. Jin and Y. Liu, "A General Method to Solve Inverse Kinematics of Spherical Wrist Manipulators," in *2019 IEEE International Conference on Mechatronics and Automation (ICMA)*, 2019, pp. 2132-2137.

[18] K. Tokarz and S. Kieltyka, "Geometric approach to inverse kinematics for arm manipulator," *International Conference on Systems - Proceedings,* vol. 1, 2010.

[19] Y. B. Li and Z. L. Jin, "Static analysis and design of a 3-DOF spherical parallel manipulator," in *2008 IEEE International Conference on Industrial Technology*, 2008, pp. 1-4.

[20] Reachy, "Controlling the head," [Online]. Available: https://docs.pollen-robotics.com/sdk/first-moves/head/#:~:text=The%20Orbita%20actuator%20is%20a%20unique%20technology%20develope d,the%20degrees%20of%20freedom%20of%20the%20human%20neck.. [Accessed 11 May 2023].

[21] S. Hirose, "Design of Hyper-Redundant Arm," in *Robotics Research*, London, Springer London, 1996, pp. 548-557.

[22] Wikipedia, "Rolls-Royce LiftSystem," [Online]. Available: https://en.wikipedia.org/wiki/Rolls-Royce_LiftSystem#:~:text=Instead%20of%20using%20separate%20lift%20engines%2C%20like%20th e,temperatures%20in%20conventional%20flight%20to%20achieve%20supersonic%20speeds.. [Accessed 11 May 2023].

[23] Tokasa, "Jet engine F135(STOVL variant)'s thrust vectoring nozzle," Wikimedia Commons, 2 March 2010. [Online]. Available: https://commons.wikimedia.org/w/index.php?curid=9623473. [Accessed 11 May 2023].

[24] Wikipedia, "Extraocular muscles," Wikipedia, [Online]. Available: https://en.wikipedia.org/wiki/Extraocular_muscles. [Accessed 11 May 2023].

[25] T. J. G. Betts, Anatomy and Physiology, Houston, Texas, USA: OpenStax, 2013.

[26] ikkalebob, "Simplified 3D Printed Animatronic Dual Eye Mechanism," Autodesk Instructables, [Online]. Available: https://www.instructables.com/Simplified-3D-Printed-Animatronic-Dual-Eye-Mechani/. [Accessed 11 May 2023].

[27] M. Callegari, L. Carbonari, G. Palmieri and M. Palpacelli, "Parallel Wrists for Enhancing Grasping Performance," 2013, pp. 189-219.

[28] Ikkalebob, "DIY Compact 3D Printed Animatronic Eye Mechanism," Autodesk Instructables, [Online]. Available: https://www.instructables.com/DIY-Compact-3D-Printed-Animatronic-Eye-Mechanism/. [Accessed 11 May 2023].

[29] L. Wilcox and J. Harris, "Fundamentals of Stereopsis," in *Encyclopedia of the Eye*, Oxford, Academic Press, 2010, pp. 164-171.

[30] DFROBOT, "8 Megapixels USB Camera with Microphone (Compatible with Raspberry Pi/ LattePanda/ Jetson Nano)," DFROBOT, [Online]. Available: https://www.dfrobot.com/product-2188.html. [Accessed 11 May 2023].

[31] Arduino, "UNO R3," Arduino, [Online]. Available: https://docs.arduino.cc/hardware/uno-rev3. [Accessed 11 May 2023].

[32] "USB," Wikipedia, [Online]. Available: https://en.wikipedia.org/wiki/USB. [Accessed 11 May 2023].

[33] robocraze, "Different Types of Motors Used in Robotics," robocraze, [Online]. Available: https://robocraze.com/blogs/post/types-of-motors-used-in-robotics. [Accessed 11 May 2023].

[34] Wikipedia, "Stepper motor," Wikipedia, [Online]. Available: https://en.wikipedia.org/wiki/Stepper_motor. [Accessed 11 May 2023].

[35] J. O'Connell, "What Is Microstepping? – Simply Explained," All3DP, [Online]. Available: https://all3dp.com/2/microstepping-simply-eplained/. [Accessed 11 May 2023].

[36] Onecall Farnell, "SER0046," Farnell, [Online]. Available: https://onecall.farnell.com/dfrobot/ser0046/servo-motor-270deg-1-47n-m-6v/dp/3769942. [Accessed 11 May 2023].

[37] bigjoncoop, "Which is more efficient - Buck or Boost," All About Circuits, 16 April 2020. [Online]. Available: https://forum.allaboutcircuits.com/threads/which-is-more-efficient-buck-or-boost.168873/#:~:text=Boost%20converters%20are%20typically%20less%20efficient%20than%20Buck,the%20load%2C%20like%20it%20does%20on%20Buck%20converters.. [Accessed 11 May 2023].

[38] M. McCauley, "AccelStepper library for Arduino," 31 October 2022. [Online]. Available: http://www.airspayce.com/mikem/arduino/AccelStepper/. [Accessed 11 May 2023].

[39] Adafruit, "Adafruit-PWM-Servo-Driver-Library," Github, 29 June 2022. [Online]. Available: https://github.com/adafruit/Adafruit-PWM-Servo-Driver-Library. [Accessed 11 May 2023].

[40] "ROS Home Page," Open Robotics, [Online]. Available: https://www.ros.org/. [Accessed 11 May 2023].

Appendix

A. Control Code

a. External Controller

```
import keyboard
import serial
import struct
import time

# Set your Arduino's COM port
arduino_port = "COM3" # Example for Windows
# arduino_port = "/dev/ttyACM0" # Example for Linux

# Open serial connection
ser = serial.Serial(arduino_port, 9600)

def send_uint16(value):
if 0 <= value <= 65535:
        packed_data = struct.pack("<H", value)
        ser.write(packed_data)
    else:
        raise ValueError("Value must be between 0 and 65535.")
```

```
def read_uint16():
    data = ser.read(2)
    if len(data) == 2:
        value = struct.unpack("<H", data)[0]
        return value
    else:
        return None

time.sleep(2)  # Wait for the Arduino to initial

while not keyboard.is_pressed("z"):
    time.sleep(0.1)
    stepperX_forward = keyboard.is_pressed("q")
    stepperX_backward = keyboard.is_pressed("a")
    stepperY_forward = keyboard.is_pressed("w")
    stepperY_backward = keyboard.is_pressed("s")
    stepperZ_forward = keyboard.is_pressed("e")
    stepperZ_backward = keyboard.is_pressed("d")

    servo0_forward = keyboard.is_pressed("r")
    servo0_backward = keyboard.is_pressed("f")
    servo1_forward = keyboard.is_pressed("t")
    servo1_backward = keyboard.is_pressed("g")
    servo2_forward = keyboard.is_pressed("y")
    servo2_backward = keyboard.is_pressed("h")
    servo3_forward = keyboard.is_pressed("u")
    servo3_backward = keyboard.is_pressed("j")

    StepperX = ((2 * stepperX_forward) + stepperX_backward) * (not
(stepperX_forward and stepperX_backward))
    StepperY = ((2 * stepperY_forward) + stepperY_backward) * (not
(stepperY_forward and stepperY_backward))
    StepperZ = ((2 * stepperZ_forward) + stepperZ_backward) * (not
(stepperZ_forward and stepperZ_backward))

    servo0 = (2 * servo0_forward + servo0_backward) * (not (servo0_forward and
servo0_backward))
    servo1 = (2 * servo1_forward + servo1_backward) * (not (servo1_forward and
servo1_backward))
    servo2 = (2 * servo2_forward + servo2_backward) * (not (servo2_forward and
servo2_backward))
    servo3 = (2 * servo3_forward + servo3_backward) * (not (servo3_forward and
servo3_backward))

    sendValue = (StepperX * (2**0)) + (StepperY * (2**2)) + (StepperZ * (2**4)) +
(servo0 * (2**6)) + (servo1 * (2**8)) + (servo2 * (2**10)) + (servo3 * (2**12)) +
(1 * 2**15)
    send_uint16(sendValue)
    print(sendValue)
```

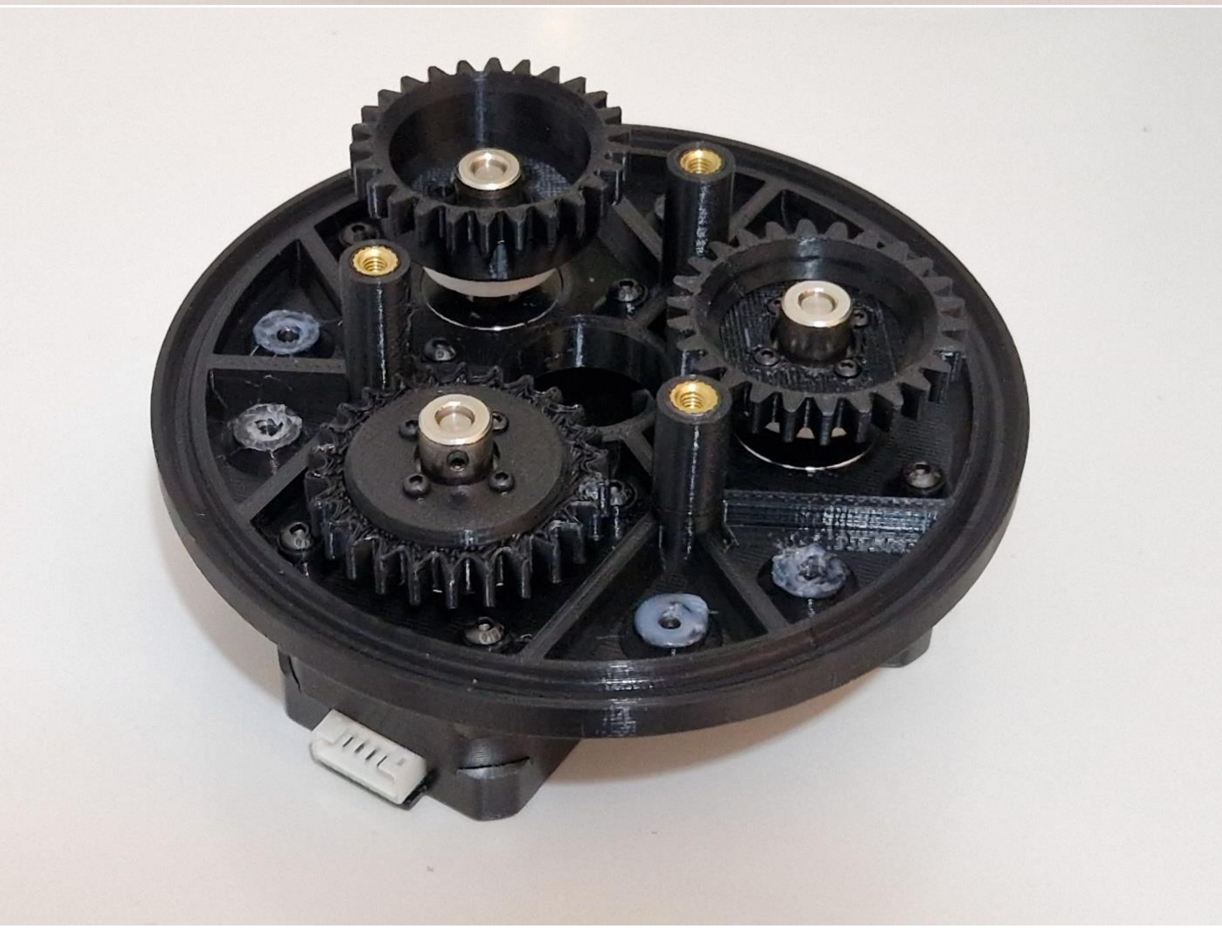
b. Arduino Code

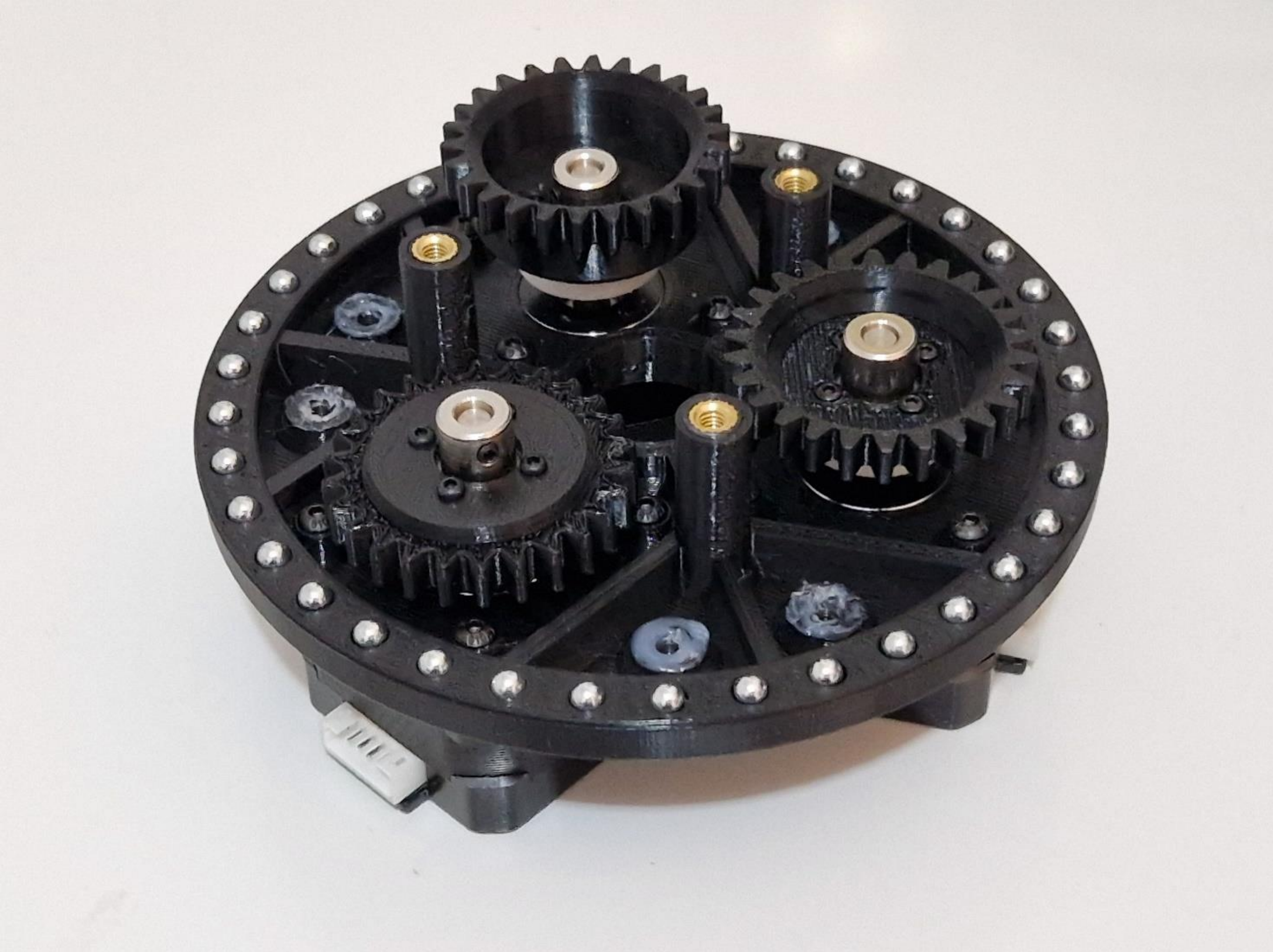
```
#include <Wire.h>
#include <Adafruit_PWMServoDriver.h>
#include <AccelStepper.h>

// Define stepper motor pins
#define X_STEP_PIN 2
#define X_DIR_PIN 5
#define Y_STEP_PIN 3
#define Y_DIR_PIN 6
#define Z_STEP_PIN 4
#define Z_DIR_PIN 7
#define ENABLE_PIN 8

// Create AccelStepper instances
AccelStepper stepperX(1, X_STEP_PIN, X_DIR_PIN);
AccelStepper stepperY(1, Y_STEP_PIN, Y_DIR_PIN);
AccelStepper stepperZ(1, Z_STEP_PIN, Z_DIR_PIN);

float maxSpeed = 1000;
float maxAccel = 100;
// Define target speeds (steps per second)
float speedX = maxSpeed;
float speedY = maxSpeed;
float speedZ = maxSpeed;

// Define acceleration (steps per second per second)
float accelerationX = maxAccel;
float accelerationY = maxAccel;
float accelerationZ = maxAccel;

// Create PCA9685 instance
Adafruit_PWMServoDriver pwm = Adafruit_PWMServoDriver();

// Servo settings
#define SERVOMIN 150 // This is the 'minimum' pulse length count (out of 4096)
#define SERVOMAX 600 // This is the 'maximum' pulse length count (out of 4096)

// Add global variables to store motor states
uint16_t motorCommands = 0;

void setup() {
  // Set initial speed and acceleration for each motor
  stepperX.setMaxSpeed(speedX);
  stepperX.setAcceleration(accelerationX);

  stepperY.setMaxSpeed(speedY);
  stepperY.setAcceleration(accelerationY);

  stepperZ.setMaxSpeed(speedZ);
```

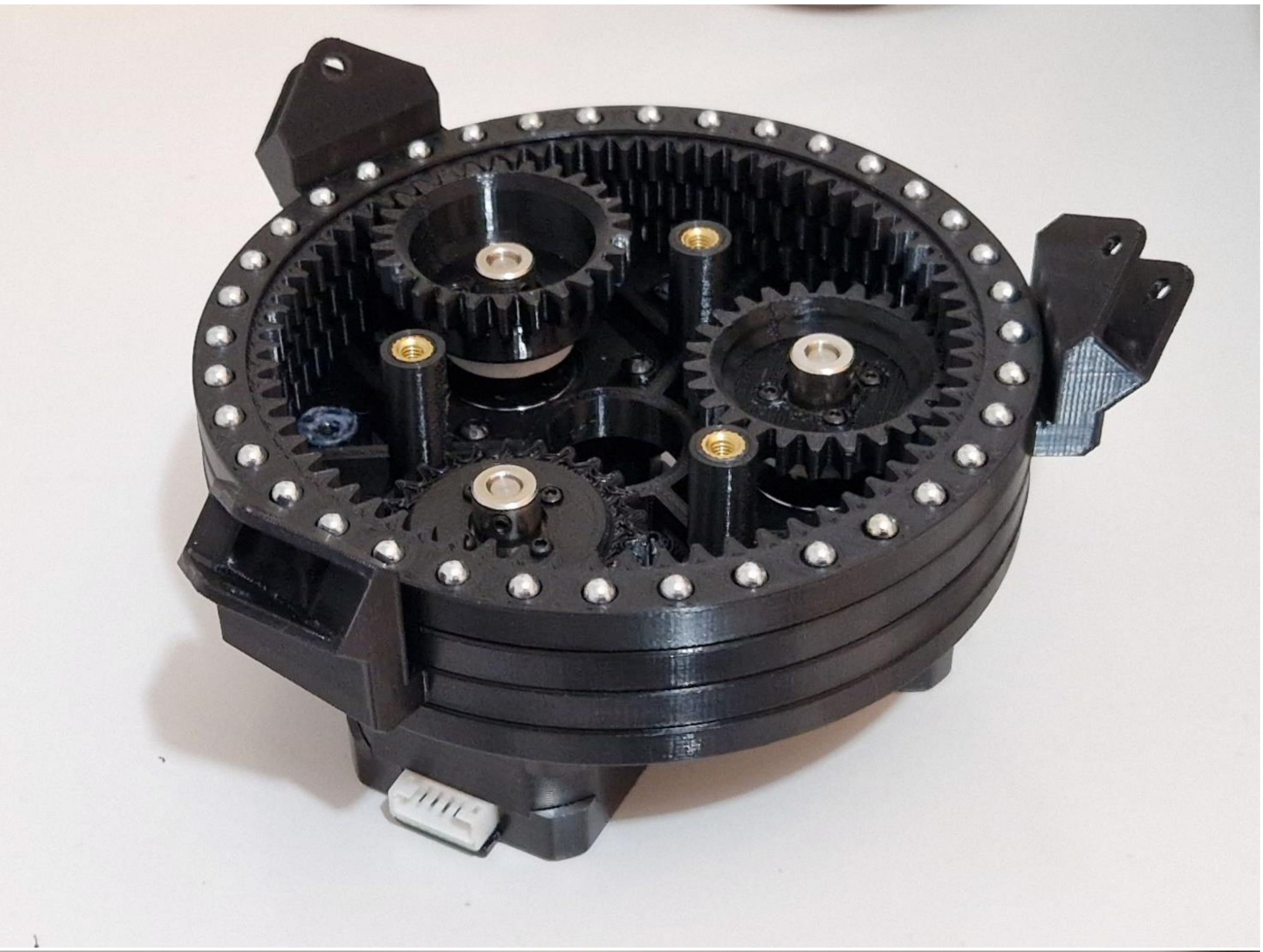
```
  stepperZ.setAcceleration(accelerationZ);

  // Initialize PCA9685
  pwm.begin();
  pwm.setPWMFreq(60);  // Analog servos run at ~60 Hz updates

  Serial.begin(9600);
  pinMode(ENABLE_PIN, OUTPUT);
  digitalWrite(ENABLE_PIN, LOW);
}

void loop() {
  // put your main code here, to run repeatedly:
  // Read incoming data from Python script
  if (Serial.available() >= 2) {
    uint16_t receivedValue = Serial.read() | (Serial.read() << 8);
    bool check = (receivedValue >> 15) & 1;
    if (check & (motorCommands != receivedValue & 0b0011111111111111)) {
      motorCommands = receivedValue & 0b0011111111111111;
    }
  }
  // Control stepper motors
  controlStepper(stepperX, (motorCommands >> 0) & 0b11);
  controlStepper(stepperY, (motorCommands >> 2) & 0b11);
  controlStepper(stepperZ, (motorCommands >> 4) & 0b11);


  // Control servo motors
  controlServo(0, (motorCommands >> 6) & 0b11);
  controlServo(1, (motorCommands >> 8) & 0b11);
  controlServo(2, (motorCommands >> 10) & 0b11);
  controlServo(3, (motorCommands >> 12) & 0b11);
}

void controlStepper(AccelStepper& stepper, uint8_t command) {
  if (command == 1) {
    stepper.moveTo(1000);
    stepper.setSpeed(maxSpeed);

    // stepper.move(10000);
    // stepper.run();
  } else if (command == 2) {
    stepper.move(-10000);
    stepper.run();
  } else {
    stepper.stop();
  }
}

void setServoPosition(uint8_t servoIndex, uint8_t angle) {
```

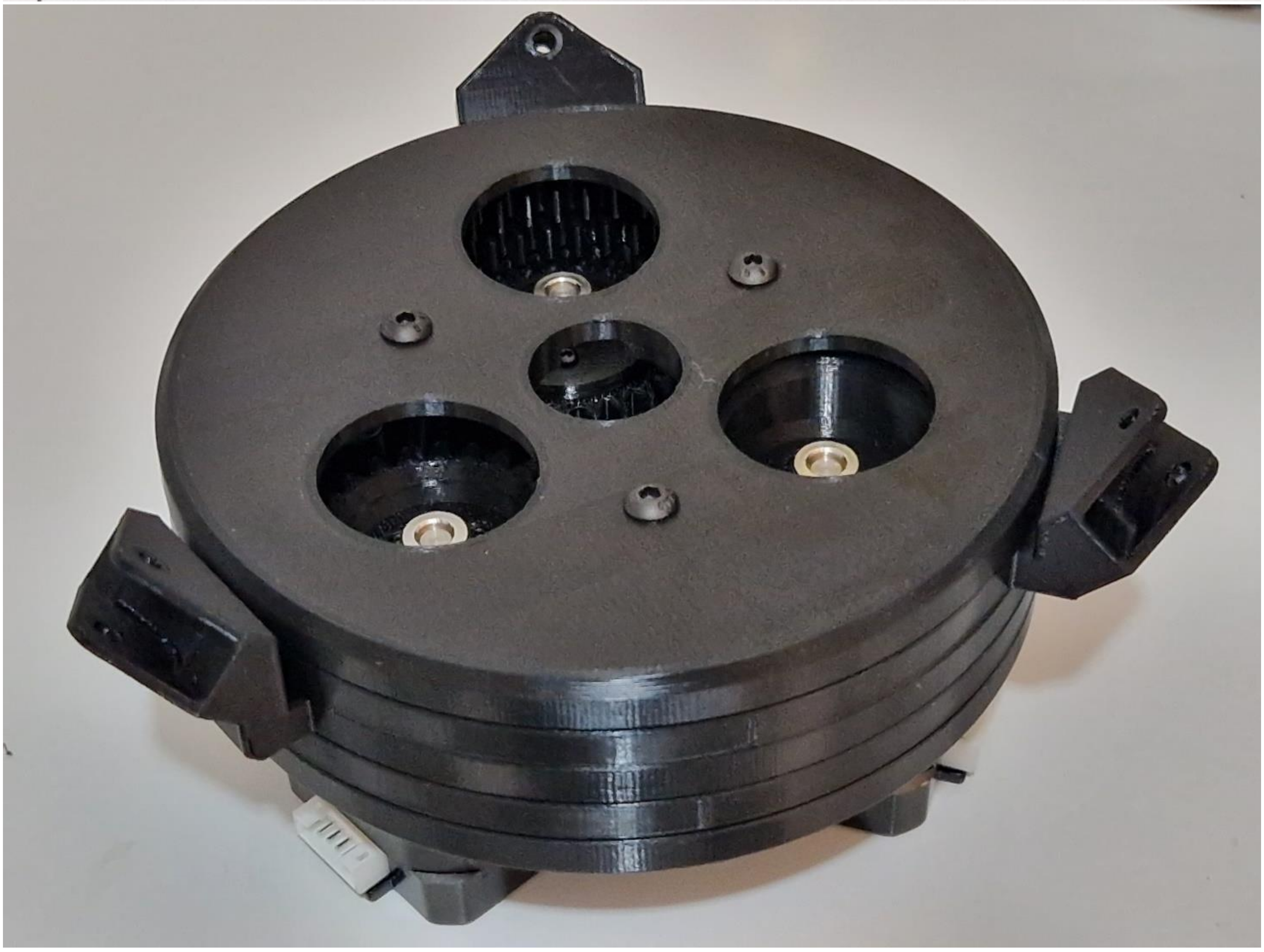
```
  if (angle > 180) {
    angle = 180;
  }
  uint16_t pulseLength = map(angle, 0, 180, SERVOMIN, SERVOMAX);
  pwm.setPWM(servoIndex, 0, pulseLength);
}

void controlServo(uint8_t servoIndex, uint8_t command) {
  if (command == 0) {
    setServoPosition(servoIndex, 90);
  }
  else if (command == 1) {
    setServoPosition(servoIndex, 110);
  }
  else if (command == 2) {
    setServoPosition(servoIndex, 70);
  }
}
```

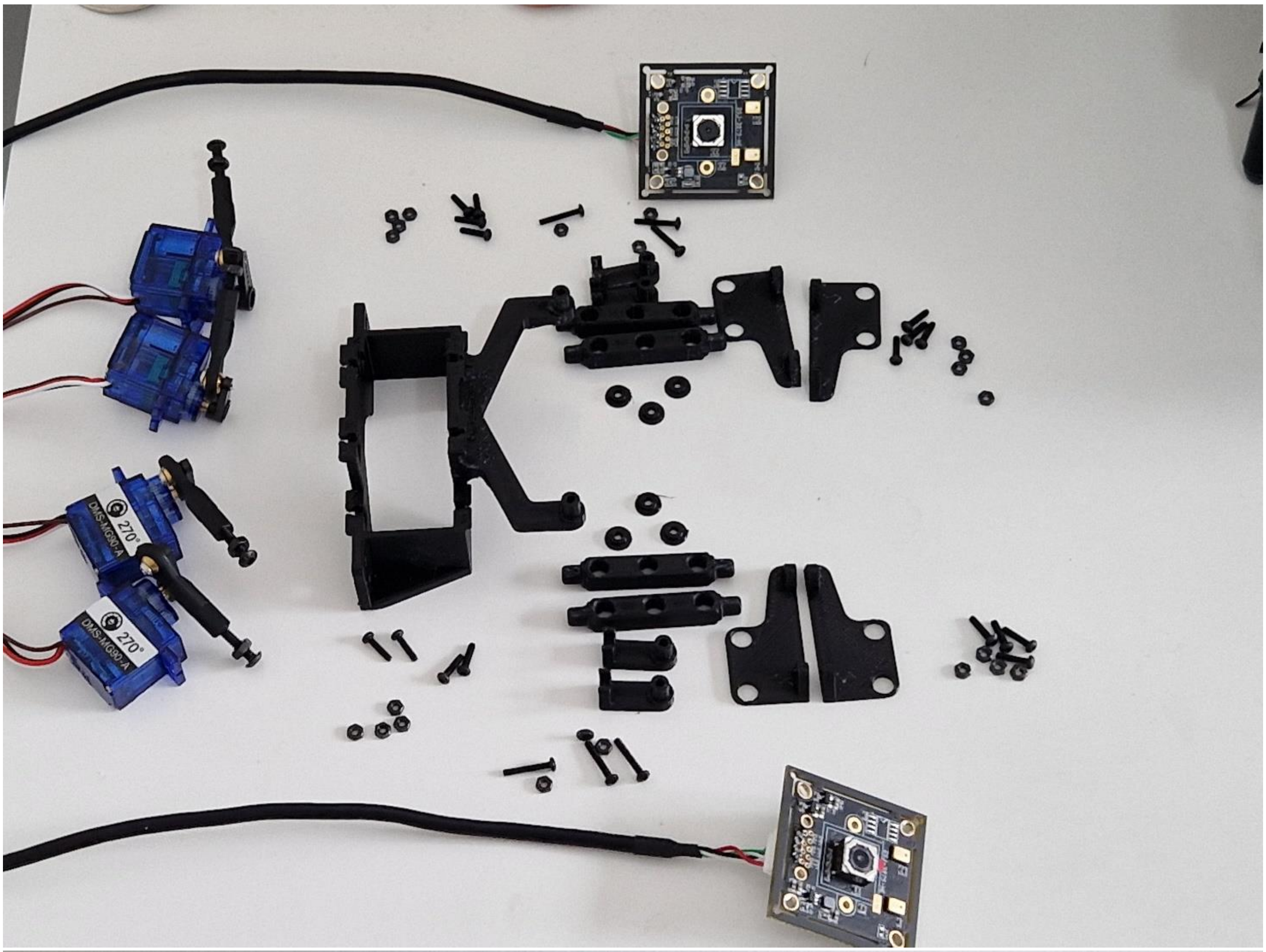
B. Assembly Process

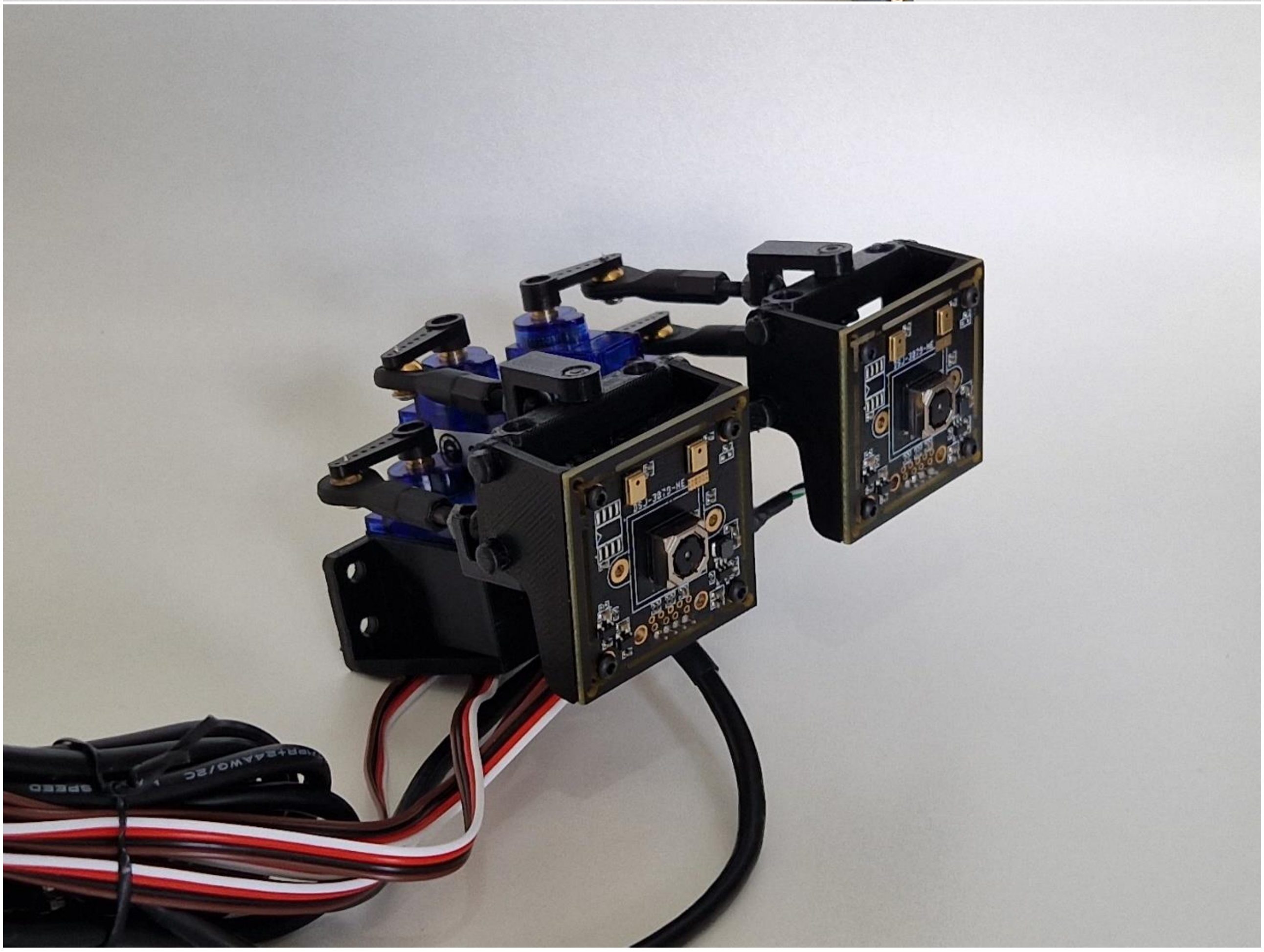


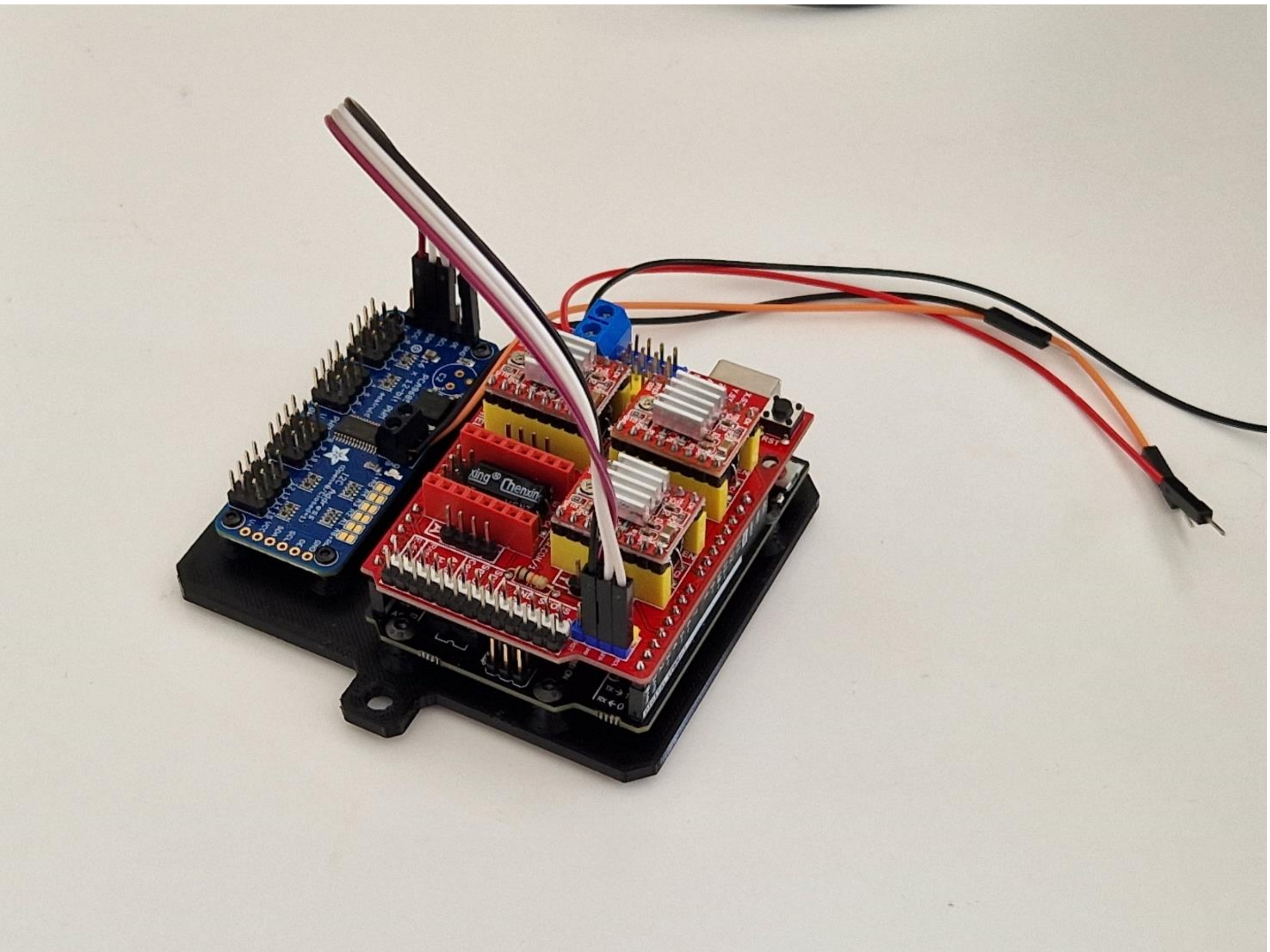


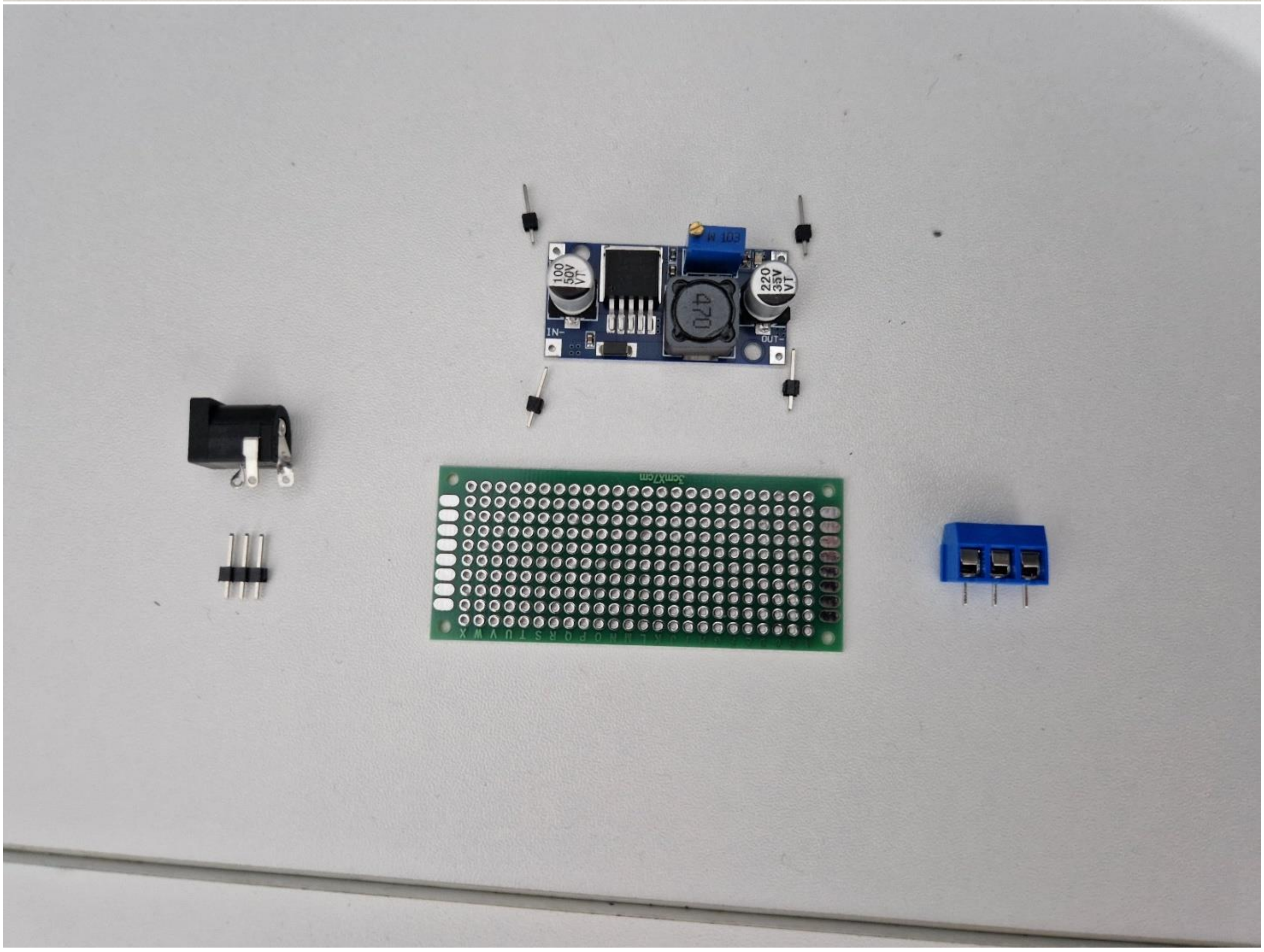


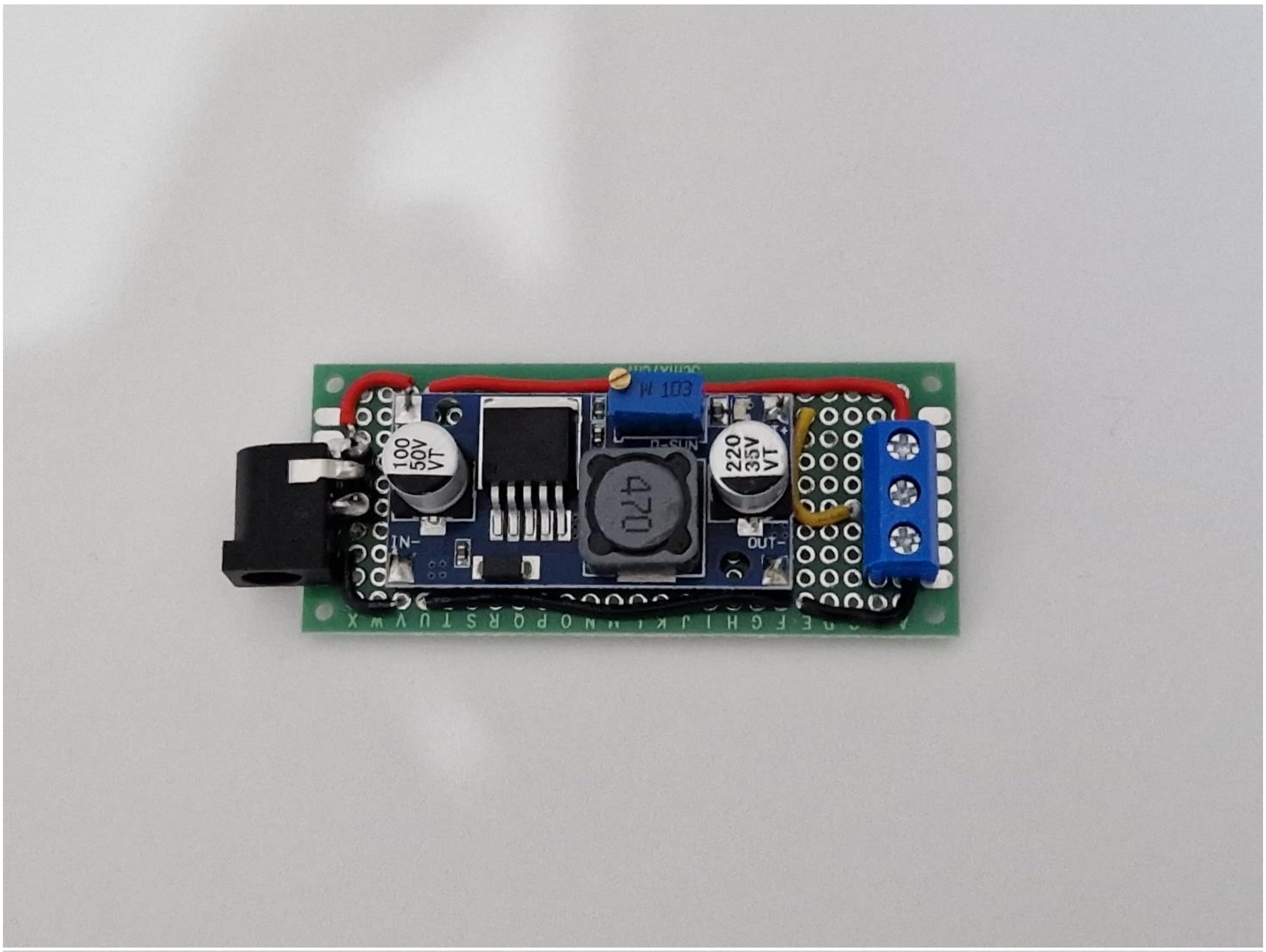


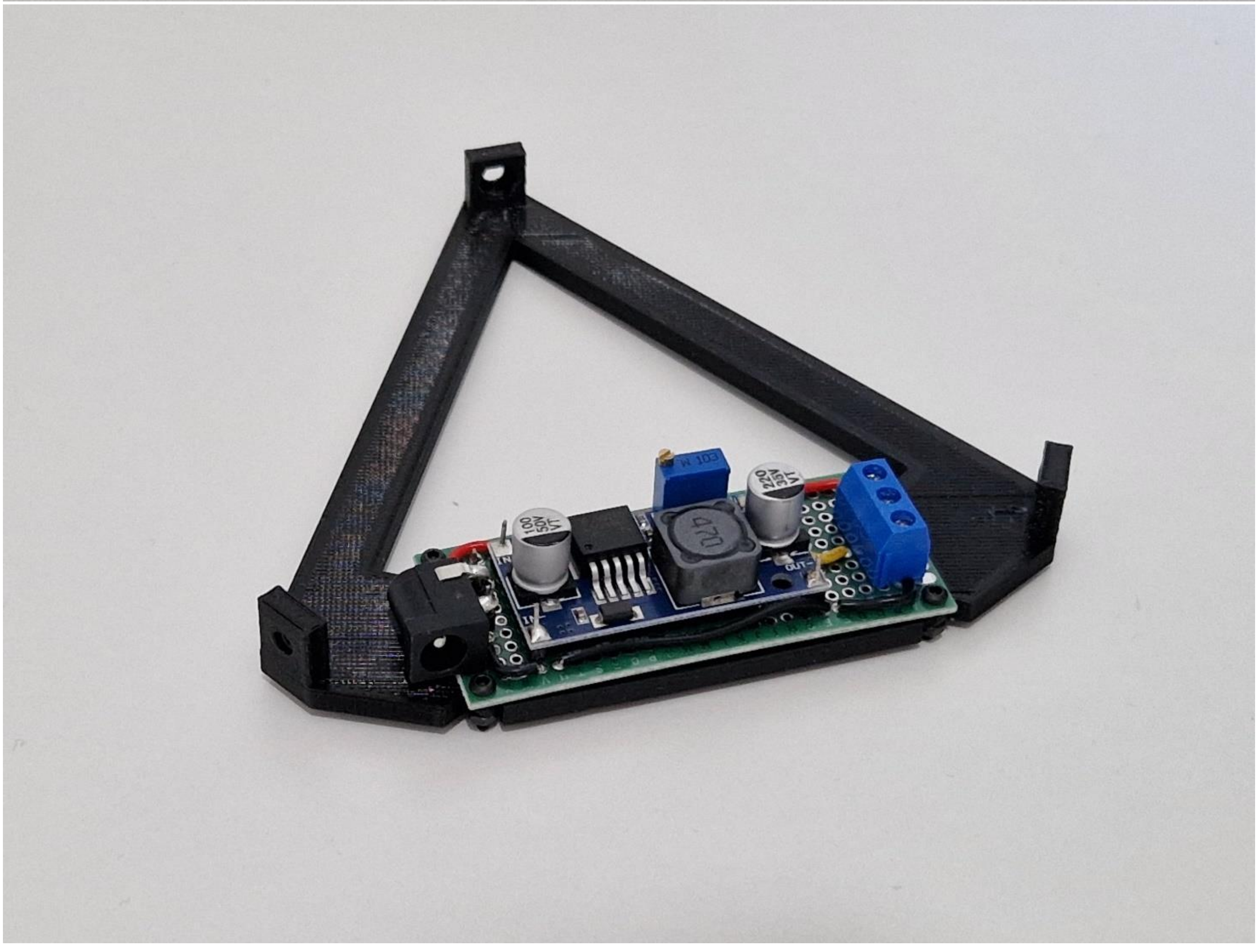

270°
DMS-MG90-A
270°
DMS-MG90-A

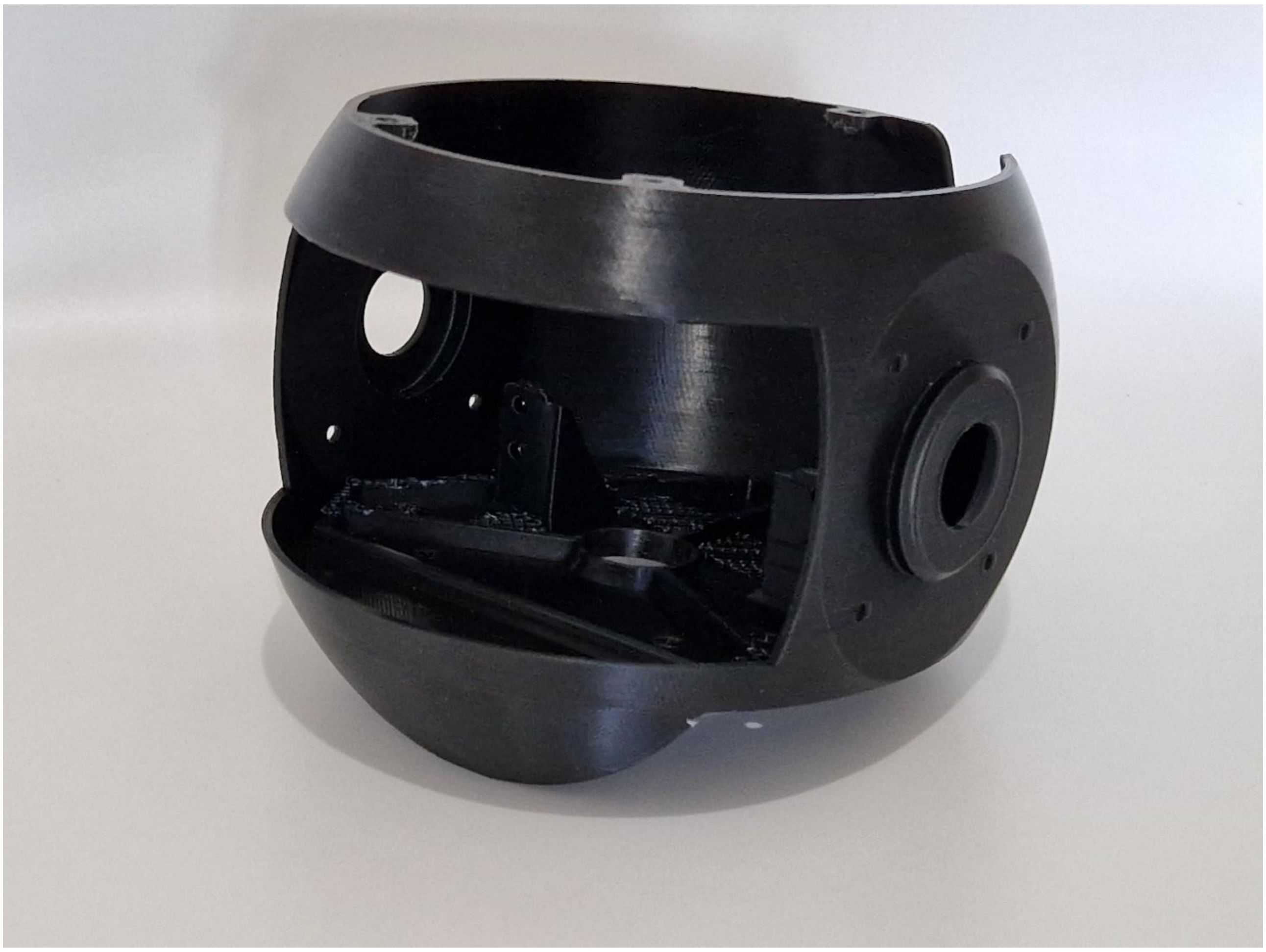




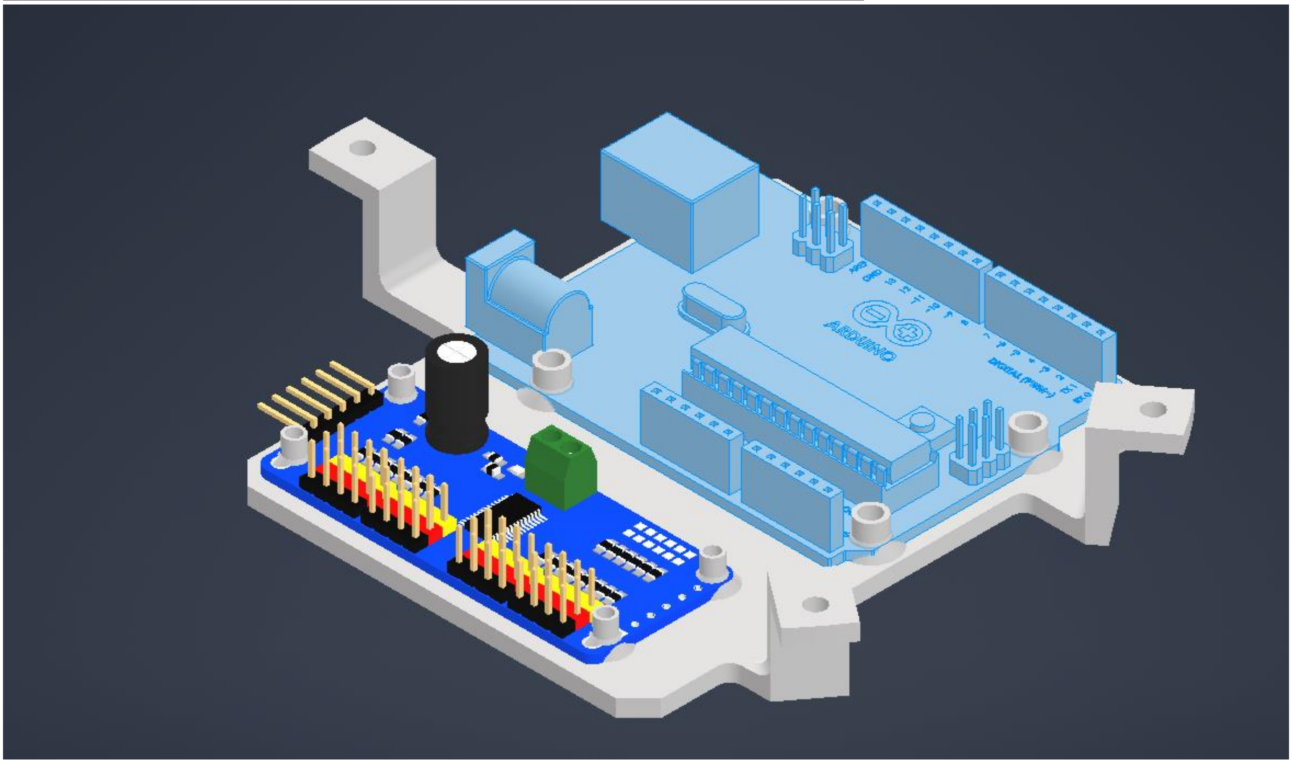


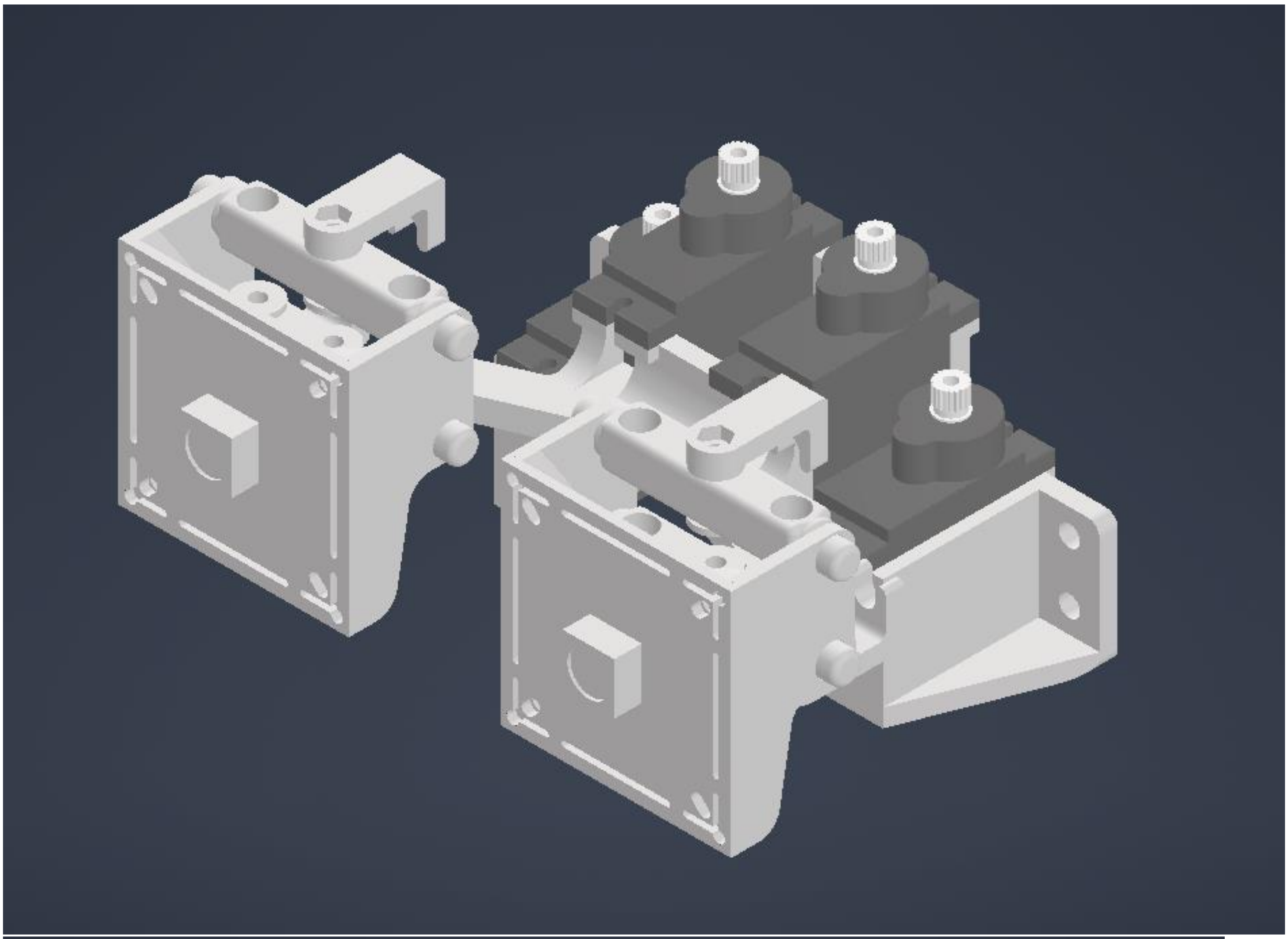

W 103
100
50V
VT
220
35V
VT
470
IN-
OUT-

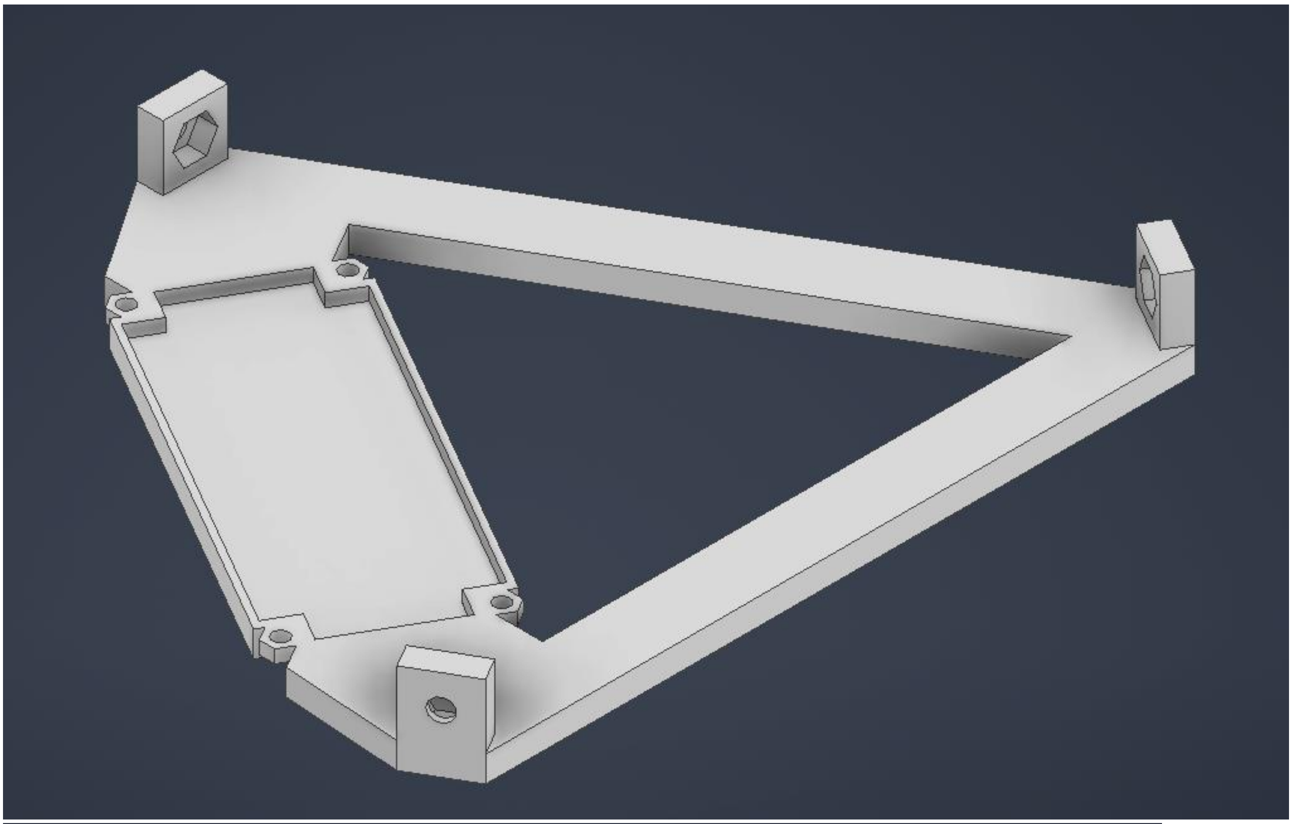


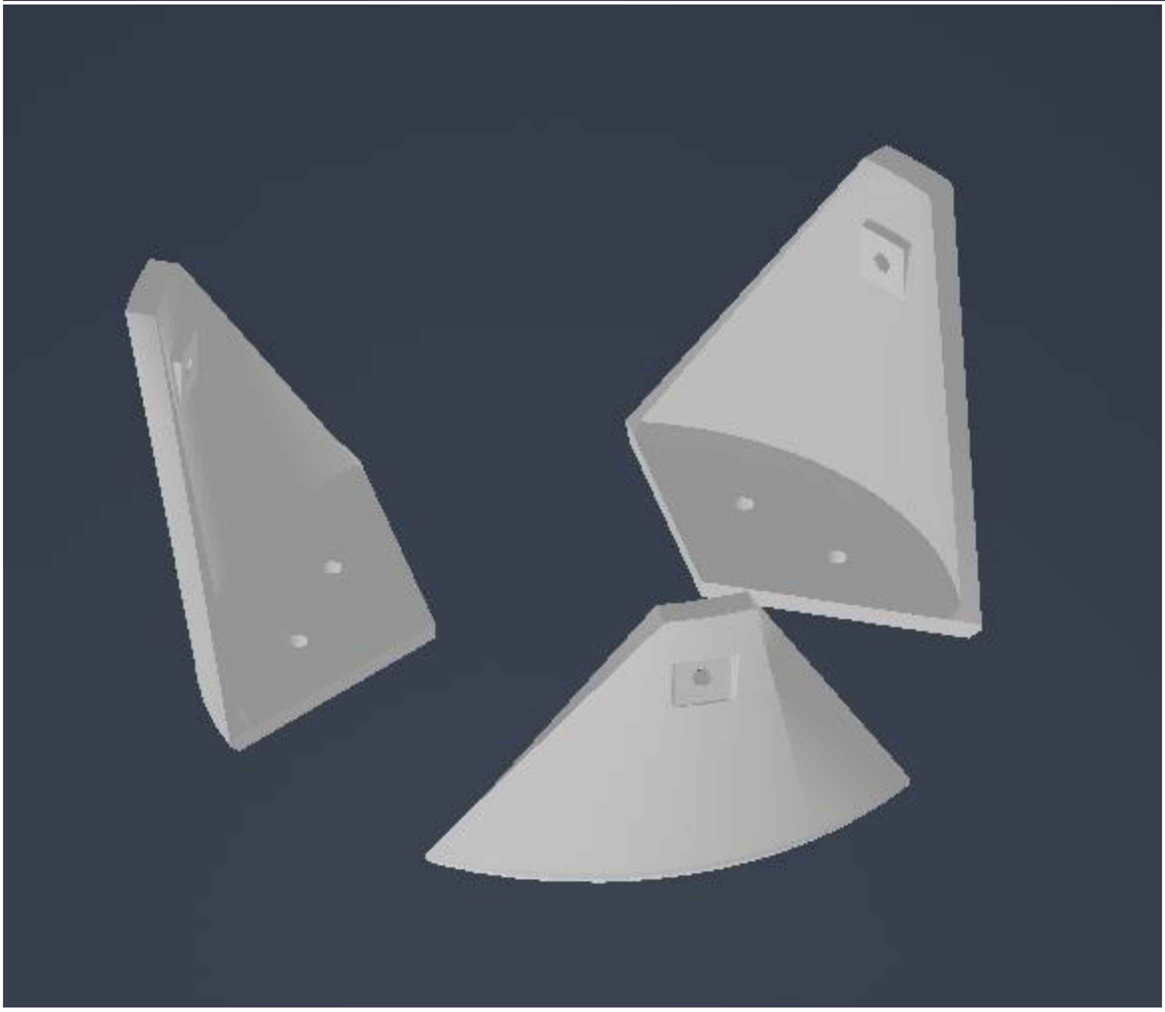


C. CAD / Physical Part Pictures









D. Cost incurred Directly by the Project

Total = 165.05 (ex VAT)

ORDER FORM

DEPT OF ELECTRONIC & ELECTRICAL ENGINEERING

Name:	Nicholas Hedworth
Email Address:	nrwh20@bath.ac.uk
Budget Code:	**BA-EE2FGN**
Supplier:	**Farnell (https://onecall.farnell.com)**

**** PLEASE USE A SEPARATE FORM FOR EACH SUPPLIER ****

Qty.	SUPPLIER order code	Brief description of item	Unit price (Ex. VAT)
1	CN23185	CON70 - CCTV 2.1mm DC Plug 2x Screw Terminal Ada	£0.58
1	CN23186	ON71 - CCTV 2.1mm DC Socket 2x Screw Terminal Ada	£0.58
2	3769901	Module, Raspberry Pi and Jetson Nano Mainboard, Imx	£32.18
3	2507562	per Motor, High Torque, Single Shaft, 42 mm, Bipolar,	£12.29
1	SC13001	A000073 - Arduino UNO SMD Rev3	£16.64
4	3769942	SER0046 - Servo Motor, 6 V, 1.47 N-m	£5.57
1	PW04032	056-FR - 12V, 2.5A, Desktop Power Supply (Level VI), I	£15.05
1	PL13305	059 - UK Plug to IEC C13 Socket Mains Lead, 5A, 2m,	£2.26
1	PW02844	T6164ST - 5V, 2.5A, Plug In Power Supply, 2.1mm Plug	£6.43

The preferred suppliers are Farnell or CPC at website: http://onecall.farnell.com
Other suppliers in order of preference are: RS, Rapid, Robot Shop and Robosavvy.
Please check that items are in stock and that they are not listed as 'U.S. stock'.
(Note: Surface mount components are usually marked SMD or SOIC, and
non-surface mount 'through hole' or DIP).

When completed, please email to: jh2476@bath.ac.uk

Once delivered, you'll receive an email and items will be left on the bench
near the door in 2E 2.17.