GERMAN - MONGOLIAN INSTITUTE
FOR RESOURCES AND TECHNOLOGY

The present work was submitted to the Faculty of Engineering

Design of 4-DOF Robotic Arm For Pick-and-Place Application

Bachelor Thesis

by

Bayarjavkhlan Sayannyamba

Supervisor 1: **Prof Odbileg Norovrinchen**

Supervisor 2: **Mr Myagmarjav Bold**

Ulaanbaatar/Nalaikh, 11/14/2025

Statutory Declaration

Sayannyamba, Bayarjavkhlan (15622942110089)

I hereby affirm in lieu of an oath that I provided the submitted bachelor thesis xxx

I did not use any sources other than those stated. In case that the work is additionally submitted on a data medium, I declare that the written and the electronic form are completely identical. The work was not submitted in the same or similar form to any examination authority.

Place, Date Signature

С.Баяржавхлан

11/14/2025

Acknowledgement

I would like to express my sincere gratitude to my primary supervisor, Prof. Odbileg, for his invaluable guidance, support, and encouragement throughout the course of this research. His knowledge, guidance, and valuable feedback have greatly influenced the direction and success of this thesis.

I am also deeply thankful to Mr.Myagmarjav, my secondary supervisor, for his assistance, mentorship, and dedication to this project. His practical knowledge, attention to detail, and commitment to excellence have been indispensable in overcoming challenges and achieving milestones.

I would like to acknowledge the contributions of my colleagues, friends, and family members who have supported me throughout this journey. Their encouragement, understanding, and encouragement have been a source of motivation and inspiration.

Lastly, I am grateful to the academic staff, research participants, and institutions who have contributed to this thesis in various ways. Their collaboration, cooperation, and contributions have been essential to the successful completion of this research project.

Abstract

The increasing requirement for automation in various sectors has spurred the development of cost-efficient robotic solutions that ensure precision and consistency in repetitive operations. This thesis focuses on the design and development of a 4-Degree-of-Freedom (DOF) robotic arm for pick-and-place tasks. The system is created to be affordable, simple, and educational, making it ideal for small-scale industrial applications and academic projects.

The robotic arm has a lightweight structure powered by MG996 servo motors and is controlled by an ESP32 microcontroller. A central feature of the system is the integration of a PS4 controller, providing intuitive and immediate manual manipulation of the robotic arm. This control system improves user interaction by enabling smooth multi-axis movement through a single input device.

List of Figures

Figure 1. Free Body Diagram of the Robot Arm 12

Figure 2. Work region of the robotic arm. 30

Figure 3. Force diagram of the robot arm 31

Figure 4. Force diagram of the link CB 31

Figure 5. MG996R servo motor 33

Figure 6. MG90S servo motor 33

Figure 7. PS4 controller 34

Figure 8. Servo Driver 34

Figure 9. ESP32 microcontroller 35

Figure 10. Electronic circuit diagram 36

Figure 11. Design of the robot arm 47

List of Tables

Table 1. The values used for torque calculation 32

Table of contents

Acknowledgement **2**
Abstract **3**
List of Figures **4**
List of Tables **4**
Table of contents **5**
1. Introduction 6
    1.1 Background and Motivation 6
    1.2 Problem Statement 7
    1.3 Scope and Limitations 8
2. Literature Review 9
    2.1 Overview of Robotic Arms and Their Applications 9
    2.2 Degrees of Freedom in Robotic Systems 11
    2.3 Historical Development of 4-DOF Robotic Arms 15
    2.4 Joystick-Controlled Robotics in Practice 17
3. Methodology 19
    3.1 Overview of the Robotic Arm Architecture 19
    3.2 3D Modelling with Autodesk Inventor 20
    3.3 Electronic Hardware 21
    3.4 Control System Design 22
4. System Analysis and Movement Behavior 24
    4.1 Movement Capabilities and Joint Limits 24
    4.2 Response to Joystick Inputs 25
    4.3 Observed Limitations and Adjustments 26
5. Implementation and Assembly 28
    5.1 Mechanical Design 28
    5.2 Hardware Components 29
    5.3 Electronic Circuit Design and Wiring 35
6. Testing and Validation 36
7. Discussion 38
    7.1 Manual Control 38
    7.2 Challenges Faced 38
8. Conclusion 39
    8.1 Summary of Key Findings 39
    8.2 Future Research Directions 40
    8.3 Recommendations for Practical Applications 41
9. References 42
10. Appendices 44
    10.1 Source Code and Algorithms 44
    10.2 Additional Pictures 48

1. Introduction

1.1 Background and Motivation

In recent years, robotics has become a powerful part of modern engineering. The introduction of robotic arms has revolutionized many sectors, enhancing task efficiency, safety, and reliability through automation. They are widely employed in industrial, logistical, and medical settings due to their ability to execute repetitive operations with consistent accuracy. One prevalent application is the pick-and-place process, involving the transfer of items from one position to another.

However, most industrial robotic arms are very advanced and expensive. It's now possible to build simpler and cheaper robotic arms. These kinds of projects are great for learning, research, and small tasks. They give students and engineers a chance to learn about robotics without needing a large budget or expensive equipment.

The goal of this project is to design and build a low-cost, user-friendly robotic arm capable of performing pick-and-place tasks. Instead of relying on complex controls or automated programming, this robotic arm is manually controlled using a PS4 controller, making it easier and more intuitive to operate. By using a PS4 controller, the project also offers hands-on experience with electronics, mechanics, and programming, all essential elements of robotics. In the future, this project could be enhanced with features like automation, object recognition, or wireless control. For now, it serves as a strong foundation for anyone interested in learning about robotic arms and their functionality.

1.2 Problem Statement

Today, robotics is essential for addressing real-world challenges, including automating repetitive tasks, operating in hazardous conditions, enhancing productivity, and improving accuracy. Robotic arms are widely used in factories, warehouses, and even medical fields to handle tasks that would be difficult, tiring, or unsafe for humans.

However, most robotic arms available on the market are complex, expensive, and require advanced programming knowledge. This makes them difficult to use for students, small-scale users, and beginners who want to explore robotics or use it for simple tasks like picking and placing objects.

Although some systems rely on automation or potentiometers, these approaches can be difficult to configure, inflexible, or unintuitive. The key challenge is the absence of affordable, easy-to-use robotic arms that provide full range of motion without complex programming.

This project addresses that problem by designing a low-cost, 4-DOF robotic arm that is fully manually controlled using a joystick. Joystick control makes the system extremely easy and natural to use. The robotic arm operates as a natural extension of the human hand, capable of motion throughout its workspace. This characteristic makes it well-suited for educational purposes, prototyping, and basic pick-and-place operations with ease and flexibility.

1.3 Scope and Limitations

Scope of the Study:

This project focuses on the design and development of a 4-degree-of-freedom (4-DOF) robotic arm intended for simple pick-and-place operations. The robotic arm's mechanical framework was modeled using Autodesk Inventor, and its operation is managed by an ESP32-based control system. The robotic arm uses MG996 servo motors for movement and is manually controlled using a joystick module, allowing real-time control by the user.

Key features within the scope include:

- ☐ Manual joystick-based control for intuitive operation
- ☐ Design of robotic arm structure with 4 DOF
- ☐ Use of lightweight, low-cost components
- ☐ Basic pick-and-place functionality
- ☐ Educational and experimental use for understanding robotics

Limitations of the Study:

While the robotic arm achieves its basic functional goals, there are several limitations to be considered:

- ☐ The system is fully manual and does not support automation or pre-programmed sequences.
- ☐ The joystick provides basic control but lacks feedback or precision adjustments for fine positioning.
- ☐ The robotic arm does not include sensors, such as cameras or limit switches, which limits its adaptability to different environments.

- ☐ The design is not industrial-grade and is intended for learning, prototyping, and demonstration purposes only.

2. Literature Review

2.1 Overview of Robotic Arms and Their Applications

A robotic arm is a machine designed to mimic the movements of a human arm. It consists of multiple solid segments, known as links, which are joined by joints. These joints enable the links to move either through rotation or linear sliding. When arranged in sequence, the links and joints form a kinematic chain, which is essentially a series of connected parts that can move in a coordinated manner.

The robotic arm's motion can be controlled either through programming or manual input, depending on its design. In your project, for example, the movement is controlled manually using a joystick. This gives the user direct control over the arm's position and actions.

At the very end of the robotic arm is a part called the End Effector. This is the tool or attachment that actually interacts with the objects—like a gripper, claw, or suction cup—and it works just like a human hand. The end effector is used to grab, lift, or move objects, depending on the task the arm is performing.

Simply put, a robotic arm is a machine that can be programmed or manually operated to imitate the movements of a human arm, commonly performing tasks like picking, placing, or assembling objects.

Types of Robotic Arms

- ☐ Cartesian robot:- It consists of three prismatic joints aligned with the Cartesian coordinate axes. It is commonly used in applications such as arc welding, precision tool handling, and pick-and-place operations.

- ☐ Cylindrical robot:- It has axes arranged in a cylindrical coordinate system, allowing rotation around a vertical axis and linear movement up and down or outward. It is often used for assembly, handling tools, spot welding, and moving parts at die-casting machines.

- ☐ Spherical robot:- It has axes arranged in a polar coordinate system, allowing rotational and radial movement. It is commonly used for tasks such as handling machine tools, spot welding, die casting, fettling, gas welding, and arc welding.

- ☐ Scara Robot:- It consists of two parallel rotary joints that provide motion within a single plane. It is commonly used for pick-and-place tasks, applying sealants, assembly operations, and handling machine tools.

- ☐ Articulated robot: - It is a manipulator with an arm containing at least three rotary joints. It is commonly employed in die casting, assembly tasks, fettling machines, gas welding, arc welding, and spray painting.

- ☐ Parallel Robot:- It consists of an arm with prismatic or rotary joints arranged to act concurrently. One common application is in mobile platforms for handling cockpit flight simulators.

- ☐ Anthropomorphic robot:- It is a robotic arm designed to resemble a human hand, featuring independent fingers and a thumb.

2.2 Degrees of Freedom in Robotic Systems

The concept is central to robotics, representing the number of independent parameters needed to uniquely define the position and orientation of a rigid body in space. Grasping DOF is crucial for evaluating a robot's motion capabilities and is a key factor in the design and control of robotic manipulators.

DOF of a Rigid Body in Three-Dimensional Space

In three-dimensional Cartesian space, a rigid body has six degrees of freedom:

- **Three translational DOF**, corresponding to linear motion along the x-, y-, and z-axes;

- **Three rotational degrees of freedom** allow a body to rotate around each of the three axes. These rotations are often described as rotation around the front-back axis, side-to-side axis, and top-bottom axis, corresponding to roll, pitch, and yaw, respectively.

These six DOF collectively define the pose of a rigid body, encompassing both its spatial position and its orientation.

Degrees of Freedom in Robotic Manipulators

A robotic manipulator is made up of multiple links that are connected through joints. Typically, each joint adds one degree of freedom:

- A **revolute joint** permits rotational motion around a single axis;

- A **prismatic joint** allows linear translation along a single axis.

The total number of DOF in a robotic arm is equal to the sum of the DOF provided by its joints. To achieve arbitrary positioning and orientation of an end-effector in 3D space, a robot must have at least six degrees of freedom. This is known as a non-redundant manipulator, which has just enough freedom to reach any pose within its workspace, assuming no constraints or singularities.

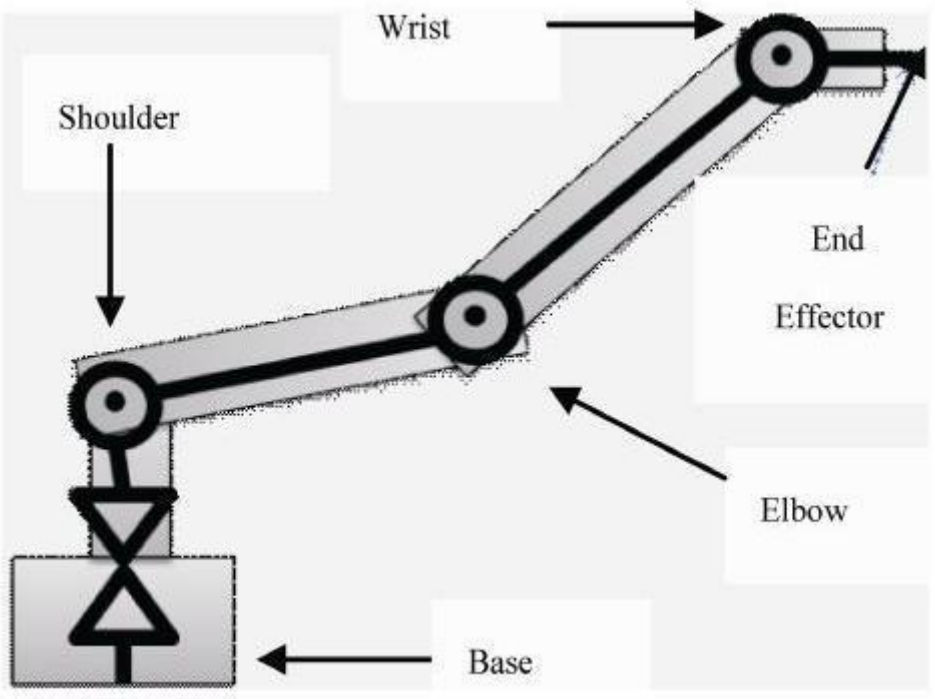


Figure 1**. Free Body Diagram of the Robot Arm**

Redundant Manipulators

Manipulators with more than six DOF are referred to as redundant. These extra degrees of freedom do not expand the task space itself but offer greater flexibility in planning and executing movements. Redundant robots can optimize secondary criteria such as obstacle avoidance, joint limit avoidance, or energy efficiency while performing a primary task.

Planar Motion and Reduced DOF

In two-dimensional (planar) environments, the number of DOF required to describe a rigid body reduces to three:

- ☐ Two translational (x and y),
- ☐ One rotational (about the axis perpendicular to the plane, typically the z-axis).

Accordingly, a planar robotic arm requires only three joints to fully control the pose of its end-effector within the plane.

Significance in Robotic Design

The robot's mechanical complexity, control approach, and operational capabilities are directly affected by its number of degrees of freedom. A robotic system with insufficient DOF may not be able to complete certain tasks or reach specific positions and orientations. Conversely, a system with excessive DOF may require more advanced control algorithms and can be more computationally demanding, but offers superior dexterity and flexibility in complex environments.

How degrees of freedom in robotics affect robot arm capabilities?

A robotic arm with a greater number of degrees of freedom can perform tasks more skillfully, adapt to a wider range of operations, and offer increased flexibility. Increasing the number of joints in a robotic arm provides greater flexibility and expands its range of motion. Think of your own arm — with joints at the shoulder, elbow, and wrist, you have a high level of manipulability.

1-DOF (One Degree of Freedom): This represents the simplest type of motion in a robot joint. It permits movement along a single linear direction, such as extending a robotic arm or opening and closing a gripper — similar to a light switch that can only move up or down.

2-DOF (Two Degrees of Freedom): Here, the joint can move in two independent ways. Imagine a robotic arm moving along a flat plane. It can typically extend and retract (linear movement) along one axis, and also rotate left and right (rotational

movement) around another axis. This combination allows for tasks like picking up objects on a flat surface.

3-DOF (Three Degrees of Freedom): Adding another DOF grants the robot movement in three independent directions. This is commonly achieved with a three-axis robotic arm. It is capable of moving along linear paths in three directions—forward/backward, left/right, and up/down—allowing for a greater range of motion. This allows for picking and placing objects at different heights or stacking items.

4-DOF (Four Degrees of Freedom): Less common than the previous examples, 4 DOF robots can usually move in the same three linear directions (forward/back, left/right, up/down) as a 3 DOF robot, but with the addition of one rotational freedom. This might be achieved by adding a wrist joint that allows the robot to twist its gripper, further enhancing its manipulation capabilities.

5-DOF (Five Degrees of Freedom): Building upon 4 DOF, a 5 DOF robot might have the same linear movements but with a more complex wrist joint. This could allow for not only twisting but also tilting the gripper, granting even greater control over object orientation.

6-DOF (Six Degrees of Freedom): This is considered full articulation for most industrial robots. They can achieve all three linear movements (forward/back, left/right, up/down) and all three rotational movements (twisting, tilting, and rotating the arm itself). This allows for incredibly precise and complex manipulation, mimicking the flexibility of a human arm.

7-DOF (Seven Degrees of Freedom and More): In specialized applications, robots might have even more DOF. These additional joints might offer specific functionalities tailored to unique tasks. For example, a snake robot designed for navigating tight spaces might have many DOF to achieve a serpentine movement.

2.3 Historical Development of 4-DOF Robotic Arms

The evolution of robotic manipulators has been shaped by advancements in automation, control systems, and mechanical engineering. Robotic arms have existed since the early 1900s, but their development accelerated in the mid-20th century alongside industrial automation. The 4-degree-of-freedom (4-DOF) arm, in particular, has proven to be a practical and economical choice for tasks that do not demand complete spatial maneuverability.

Early Developments in Robotic Manipulation

George Devol and Joseph Engelberger created the Unimate in the late 1950s, marking the introduction of the first industrial robot. It was first deployed on a General Motors assembly line in 1961. This robot had a relatively simple structure and was designed to perform repetitive pick-and-place operations. Although it had limited degrees of freedom, it laid the groundwork for programmable, electrically controlled robotic systems.

Robotics research and development grew substantially during the 1970s and 1980s, especially in the United States, Japan, and Europe. Robots during this period became more sophisticated, incorporating sensors, microprocessors, and advanced control strategies. The introduction of Denavit-Hartenberg parameters and kinematic modeling techniques enabled the precise analysis of robotic motion, including arms with fewer than six degrees of freedom.

Rise of 4-DOF Robotic Arms

The 4-DOF robotic arm configuration became particularly popular in applications where complete orientation control of the end-effector was not required. These include:

- ☐ Pick-and-place operations,
- ☐ Sorting and packaging,
- ☐ Simple assembly tasks,
- ☐ Educational and research tools.

By eliminating two degrees of freedom (typically the wrist pitch and yaw), the 4-DOF design simplifies the mechanical structure and control logic, resulting in a more affordable and lightweight system. This made it highly suitable for small-scale manufacturing environments and embedded robotic platforms.

Technological Advances and Educational Use

In recent decades, 4-DOF arms have benefited from the widespread availability of microcontrollers (such as Arduino and ESP32), servo motors, and low-cost sensors. As a result, these robotic arms are now commonly used in STEM education and university-level research. Open-source platforms and accessible simulation tools have further accelerated their adoption in academic settings, enabling students to experiment with robotic kinematics, control algorithms, and mechatronic integration.

Additionally, the integration of wireless controllers, such as PS4 or Bluetooth-based input devices, has enhanced the interactivity and usability of these systems in human-robot interaction studies.

Current Applications and Trends

Today, 4-DOF robotic arms are widely used in:

- ☐ Consumer robotics kits,
- ☐ Hobbyist projects,
- ☐ Prototyping in research labs,
- ☐ Automated inspection systems.

Their simplicity, combined with advances in embedded systems and real-time control, continues to make them a valuable platform for innovation and learning. Although they do not offer the full range of motion of 6-DOF industrial arms, they remain valuable for specialized tasks in both educational and industrial settings.

2.4 Joystick-Controlled Robotics in Practice

Controlling the robotic arm using a joystick — specifically, a PS4 controller — provided an intuitive and responsive way to operate the system. The joystick inputs

were wirelessly transmitted to the ESP32 microcontroller, which then converted these commands into movement instructions for the MG996R servo motors.

Each axis and button on the controller was mapped to specific joints or actions of the arm:

- ☐ One joystick controlled the base rotation and shoulder movement.
- ☐ The other joystick managed the elbow and wrist positioning.
- ☐ The buttons were assigned to operate the gripper, allowing it to open and close.

This setup allowed for smooth, real-time manual control, making it easy to position the arm precisely without the need for pre-programmed motions. The wireless communication, once properly paired, was generally stable within a reasonable operating distance.

A major benefit of joystick control is the ability to vary movement speed in real time—for instance, moving slowly when handling delicate items and faster during longer motions. However, it also required some practice to develop the coordination needed to move multiple joints smoothly together.

Overall, joystick-controlled operation made the robotic arm highly flexible and user-friendly, especially for tasks like pick-and-place where adaptability is important.

3. Methodology

3.1 Overview of the Robotic Arm Architecture

The 4 DOF robotic arm is designed to perform basic pick-and-place tasks using a combination of servo motors, a wireless controller, and a compact microcontroller. The system is built around four main parts: the mechanical structure, the actuation system, the control unit, and the user interface.

At the heart of the system is the robotic arm, designed with four degrees of freedom that allow rotation of the base and wrist, as well as movement at the shoulder and elbow joints. These allow the arm to reach, lift, and place objects within its operating range. The frame is constructed from durable yet lightweight materials, providing stability while keeping the load manageable for the motors.

Each joint is powered by an MG996R servo motor, which is well-suited for medium-duty tasks thanks to its high torque and simple control interface. These motors respond to PWM (Pulse Width Modulation) signals to set their position accurately, which is essential for precise and repeatable movements.

Controlling the entire system is an ESP32 microcontroller. It serves as the central processor, handling incoming commands and generating the appropriate signals for each motor. The ESP32 is chosen for its built-in Bluetooth capability and fast performance, which allows it to wirelessly communicate with a PlayStation 4 controller.

The PS4 controller serves as the user interface, providing an intuitive way to manually control the arm. With the joysticks and buttons, the user can move the arm’s joints in real time or trigger pre-programmed motions. The Bluetooth link between the controller and ESP32 provides both convenience and operational flexibility.

Power for the motors comes from an external power supply to ensure stable and reliable performance, while the ESP32 is powered separately, often through USB or a regulated 5V source. Care is taken to manage power and signal lines to avoid interference or instability.

Altogether, this architecture allows the robotic arm to operate smoothly and accurately in a variety of simple pick-and-place applications. It's also designed to be modular and upgradable, making it easy to add features like sensors or automation in the future.

3.2 3D Modelling with Autodesk Inventor

For the mechanical design of the 4 DOF robotic arm, I used Autodesk Inventor to plan, design, and visualize the layout of the system before starting the physical build. The entire structure was modeled with the intention of 3D printing custom parts that would fit the MG996R servo motors and meet the mechanical requirements of the arm.

The robotic arm is made up of four main joints—base, shoulder, elbow, and wrist—each driven by an MG996R servo. In Inventor, I designed custom mounts and connecting arms to securely hold the servos and provide a smooth range of motion for each joint. The design process involved carefully considering alignment, spacing, and mechanical strength to ensure reliable performance during pick-and-place tasks.

Using Autodesk Inventor allowed me to simulate the movement of the arm and check for potential issues like joint collisions or restricted motion. It also helped in planning cable management and optimizing the arm's structure to balance strength and weight.

Once the models were finalized, the parts were 3D printed using lightweight yet durable materials. Keeping the arm as light as possible was essential, given the torque

limitations of the MG996R servos. The printed parts were designed to be easily assembled with standard screws, making the construction process straightforward and allowing for adjustments if needed.

In summary, Autodesk Inventor helped me piece together the mechanical structure virtually before manufacturing. Custom 3D-printed parts gave the robotic arm a fully tailored structure, improving both its performance and adaptability for future improvements.

3.3 Electronic Hardware

The electronic hardware forms the core of the robotic arm's control system. It includes the microcontroller, servo motors, power supply, and communication interface. The configuration prioritizes simplicity, performance, and responsiveness, providing seamless control of all four DOF.

At the heart of the system is the ESP32 microcontroller, which handles all communication and motor control. It's a compact, dual-core board with built-in Bluetooth and Wi-Fi, making it ideal for wireless control applications like this one. In this system, the ESP32 communicates with a PS4 controller via Bluetooth, allowing the controller to function as the primary interface for manual operation of the robotic arm.

The arm is powered by MG996R servo motors, one for each joint. These are high-torque servo motors capable of lifting and moving lightweight objects with good speed and accuracy. Each servo is connected directly to the ESP32's PWM-capable

pins, which allows for precise angle control based on user input from the PS4 controller.

Servos can require significant current when moving under load, so an external 5–6 V power supply is provided to ensure reliable operation. This ensures stable operation and prevents the ESP32 from resetting due to voltage drops. A common ground is shared between the power supply and the ESP32 to maintain signal integrity.

The PS4 controller communicates wirelessly with the ESP32 via Bluetooth, sending joystick and button data in real time. After receiving the input, the ESP32 calculates the desired angles for each joint and sends the matching PWM signals to control the servos accordingly.

To ensure reliability and safety, basic wiring practices are followed: each servo has its own dedicated power and signal line, and all grounds are connected properly. Additional components like resistors, capacitors, or logic level converters may be added depending on the final wiring setup and stability requirements.

In summary, the electronic hardware setup consists of a Bluetooth-enabled ESP32, four MG996R servo motors, a PS4 controller for wireless input, and a separate power supply for the motors. This combination offers a compact, low-cost solution for real-time robotic arm control.

3.4 Control System Design

The control system converts commands from the PS4 controller into accurate motion of the robotic arm. It is built to remain straightforward, responsive, and easily adaptable for future enhancements or automation.

The ESP32 acts as the system's main controller, coordinating the Bluetooth connection and managing the logic that drives the robotic arm. Using Bluetooth, the PS4 controller communicates with the system wirelessly, allowing the user to operate the robotic arm instantly through its buttons and joysticks, all without any direct cable connection.

Once the ESP32 receives input from the controller, it processes the joystick and button values to determine how each joint of the arm should move. For example, one joystick might control the base rotation and shoulder movement, while the other could manage the elbow and wrist. If needed, the buttons can perform additional tasks—like opening or closing a gripper, returning the system to a default position, or switching among different control schemes.

Every joint in the robotic arm is actuated by an MG996R servo motor, which responds to position commands encoded as PWM signals. The ESP32 generates these signals to control the angle of each servo. The PWM signals are updated continuously in response to the user's inputs, ensuring smooth and responsive motion.

A key part of the design is mapping the analog joystick values (usually ranging from -128 to 127 or 0 to 255) to a usable servo angle range (typically 0° to 180°). The control code includes limits to prevent the servos from rotating beyond their safe angles, protecting both the motors and the structure of the arm.

The system also includes safety and calibration features, such as limiting the speed of motion to avoid sudden jerks, and ensuring that the servos start at a known default position on power-up. If needed, the control logic can also be extended to include position feedback, sensors, or automated sequences in the future.

Overall, the control system is designed for manual, real-time operation, allowing the user to intuitively move the robotic arm using a wireless game controller. It's easy to work with but still flexible enough for many tasks, so it's great for trying things out, learning, or showing how something works.

4. System Analysis and Movement Behavior

4.1 Movement Capabilities and Joint Limits

The 4 DOF robotic arm is designed to perform a range of pick-and-place tasks by combining controlled movements across four joints: the base, shoulder, elbow, and wrist. Each joint is powered by an MG996R servo motor and has its own specific range of motion, allowing the arm to reach, lift, and position objects within its working area.

- **Base Rotation:**

  The base can rotate left and right, giving the arm a full horizontal sweep. To protect the servo and ensure reliable operation, the rotation is limited to approximately **180°**.

- **Shoulder Movement:**

  The shoulder joint raises and lowers the main arm. It allows the arm to reach upward or downward, typically within a range of **0° to 135°**, depending on mechanical limits and servo strength.

- **Elbow Movement:**

  The elbow bends the arm inward and outward, giving it the ability to extend its reach or pull objects closer. This joint usually operates within a range of **0° to 135°** as well.

- **Wrist Movement:**

  The wrist joint adjusts the angle of the end effector (gripper or tool). It provides fine positioning control and is capable of rotating approximately **0° to 180°**.

These ranges were chosen to balance functionality with mechanical safety. Each joint's motion is slightly restricted in software to avoid overextending the servos or causing mechanical strain.

Overall, the arm’s movement capabilities allow it to pick up objects placed at different heights and angles within a semicircular working area around its base.

4.2 Response to Joystick Inputs

The robotic arm is manually operated using a PS4 controller, which communicates with the ESP32 via Bluetooth. The controller's joysticks and buttons are mapped to control each joint of the robotic arm, allowing real-time, intuitive movement.

When the user moves a joystick, the ESP32 receives the corresponding input values as numerical data. The values normally fit within a byte-sized range, which is usually 0 to 255 (in some libraries, spanning from -128 to 127 instead). The code running on the ESP32 interprets these values and maps them to servo angles in a range that is safe and functional—usually between 0° and 180°.

Each joystick axis is assigned to control a specific joint:

- ☐ **Left joystick (X-axis)** – rotates the base.
- ☐ **Left joystick (Y-axis)** – moves the shoulder up and down.
- ☐ **Right joystick (Y-axis)** – controls the elbow joint.
- ☐ **Right joystick (X-axis)** – adjusts the wrist angle.

As the joystick is pushed in a direction, the corresponding joint begins to move. When the joystick is released, the servo holds its last position. This real-time mapping makes the system feel responsive and natural, giving the user precise control over each movement.

To prevent abrupt or unsafe motion, the code includes:

- ☐ **Dead zones** around the joystick center to ignore small unintentional movements.
- ☐ **Speed limits** or gradual angle changes for smoother transitions.
- ☐ **Angle constraints** to ensure servos don't move beyond their physical or mechanical limits.

This responsive control system allows for accurate positioning during pick-and-place tasks and makes the arm easy to operate even for users with no robotics experience.

4.3 Observed Limitations and Adjustments

Throughout the development and testing of the robotic arm, a few limitations became apparent, which led to several improvements and adjustments to ensure better performance and reliability.

1. Limited Torque Performance

The MG996R servo motors provided good movement for most tasks, but when the arm was fully extended or lifting heavier objects, there were signs of strain, especially at the shoulder and elbow joints. To address this, the design was kept lightweight, and careful movement planning helped reduce unnecessary stress on the servos.

2. Power Supply Issues

When multiple servos operated at the same time, occasional voltage drops caused slight instability or reduced responsiveness. This was solved by using an

external, stable power supply dedicated to the servos, separate from the ESP32's power source, improving overall reliability during operation.

3. Bluetooth Connection Instability

The connection between the PS4 controller and the ESP32 sometimes dropped during early tests. Reconfiguring the controller using SixaxisPairTool, ensuring proper pairing, and maintaining a clear line of sight between the devices helped reduce connection issues significantly.

4. Lack of Position Feedback

Because the MG996R servos operate without feedback, the arm cannot directly measure its joint positions. While this is acceptable for manual joystick control, it limits the ability to create precise automated movements. Adding position sensors or using feedback-capable servos would be a useful future upgrade.

5. Minor Mechanical Flexibility

Some minor flexibility was noticed in the 3D-printed parts under load. While it did not cause serious problems, it slightly affected the precision of the end effector's positioning. Reinforcing critical parts or adjusting print settings like infill density helped minimize this issue.

5. Implementation and Assembly

The implementation and assembly phase focused on putting together the mechanical structure and electronic control system to build a fully operational 4 DOF robotic arm. After completing the 3D-printed components and selecting core elements such as MG996R servo motors, the ESP32 microcontroller, and a PS4 controller, the project moved into the integration stage.

Each servo motor was securely mounted at its respective joint—base, shoulder, elbow, and wrist—ensuring the correct alignment and range of motion. To handle the power and signal distribution efficiently, a servo driver module (PWM controller) was used. This allowed multiple servos to be controlled simultaneously with stable pulse-width modulation signals from the ESP32, avoiding issues caused by limited onboard PWM pins.

A dedicated power supply was connected to the servo driver to ensure that all motors received consistent voltage without overloading the ESP32. The PS4 controller was paired via Bluetooth, allowing for wireless, real-time control of each joint.

This phase was important for testing how well the components worked together, identifying any mechanical or electrical issues, and making the final adjustments needed for smooth operation.

5.1 Mechanical Design

The mechanical design of the robotic arm is modeled after a manipulator with functions similar to a human arm. Its links are connected by joints that allow rotational motion, forming what is known as a kinematic chain. The terminal part of this chain, referred to as the end effector or end-of-arm tooling, serves a role analogous to the human hand. As shown in Figure 1, the end effector is not included in the custom design, as a commercially available gripper is used instead. This choice is made

because the end effector is one of the most complex components, and using a commercial solution is both simpler and more cost-effective. Figure 2 illustrates the work region of the robotic arm, which represents the typical workspace for a four-degree-of-freedom (4-DOF) manipulator. The mechanical design was limited to 4 DOF to balance functionality with cost and complexity, providing sufficient movement for most tasks. Joint rotations are configured such that the shoulder allows motion around two axes, while the elbow and wrist rotate around a single axis, as shown in Figure 1. The arm's joints are powered by electric motors, with servo motors selected because their built-in encoders provide automatic feedback for accurate position control. A limitation of these servos is that their rotation range is under 180°, which reduces the reachable workspace and possible arm positions. The specifications for the servo motors were chosen based on the maximum torque required to handle the structure and anticipated loads.

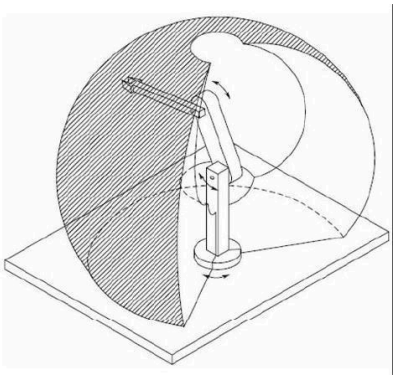


Figure 2**. Work region of the robotic arm.**

5.2 Hardware Components

Figure 3 presents the force diagram used for load analysis. Calculations were performed only for the joints experiencing the highest loads, as the remaining joints use the same motors, which are capable of moving their respective links without issue. The analysis accounted for the weight of the motors, approximately 50 grams each, except for the motor at joint B, since its weight is supported by link BA. Figure 4 illustrates the

force diagram for link CB, which includes joints B and C—the joints with the greatest load due to supporting links DC and ED. The load calculations for these joints were conducted as described below.

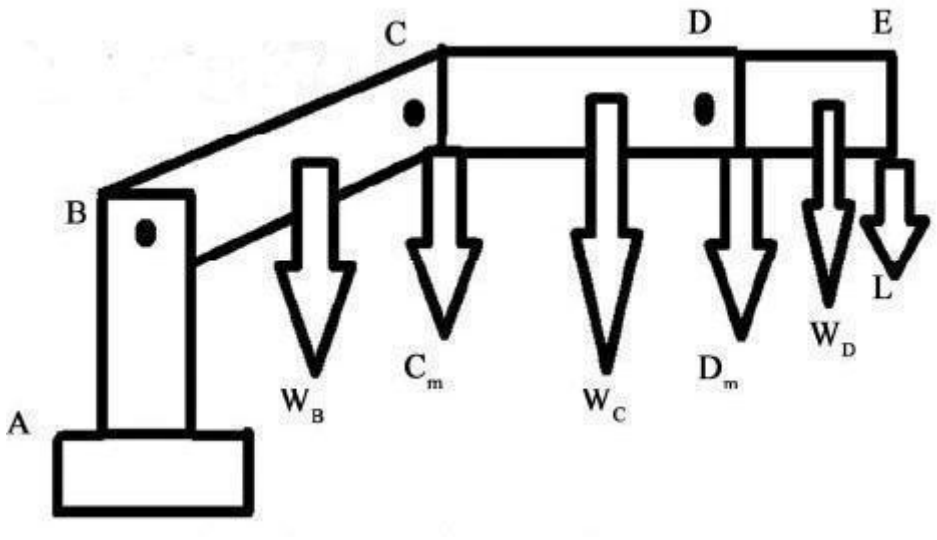


Figure 3**. Force diagram of the robot arm**

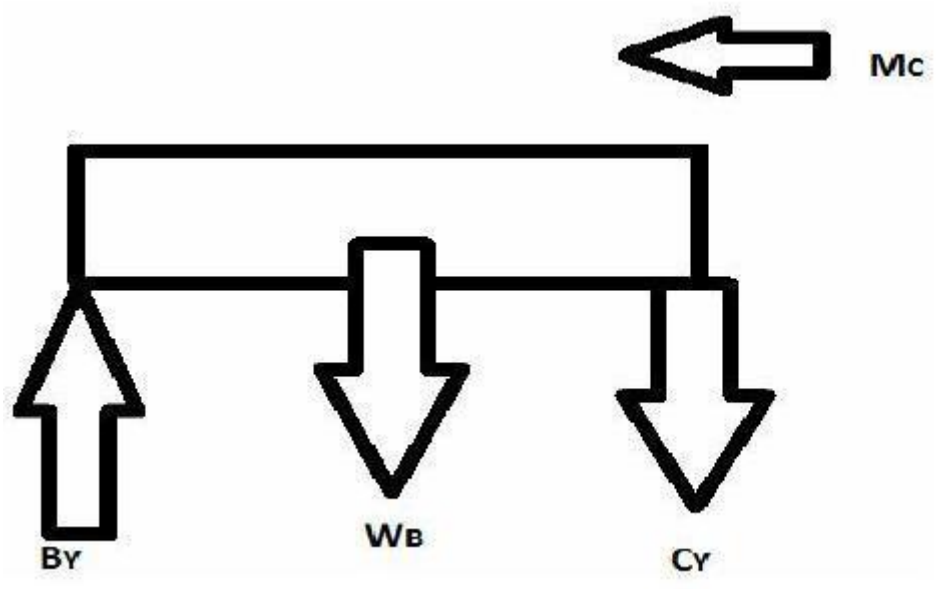


Figure 4**. Force diagram of the link CB**

Weight of link DE	W_d = 0.123 kg
Weight of link CD	W_c = 0.309 kg
Weight of link CB	W_b = 0.250 kg
Load	L = 0.05kg
Weight of motor	$C_m = D_m$ = 0.050 kg
Length of link BC	L_{BC} = 0.127 m
Length of link CD	L_{CD} = 0.203962 m
Length of link DE	L_{DE} = 0.0762 m

Table 1**. The values used for torque calculation**

$$\sum F_y = \left(L + W_d + D_m + W_c + C_m\right)\mathrm{g} - C_Y = 0$$

C_y = (0.582kg) 9.8m/s^2 = 5.7036 N

$$\sum F_y = \left(L + W_d + D_m + W_c + C_m + W_B\right)g - C_B = 0$$

C_B = (0.782kg) 9.8m/s^2 = 7.6636 N

$$\sum M_c = -\left(\frac{W_c L_{CD}}{2}\right) - W_D\left(L_{CD} + \frac{L_{DE}}{2}\right) - L\left(L_{CD} + L_{DE}\right) - D_m\left(L_{CD}\right) + M_c = 0$$

$$\sum M_B = -L\left(L_{BC} + L_{CD} + L_{DE}\right) - W_D\left(L_{BC} + L_{CD} + \frac{L_{DE}}{2}\right) - D_m\left(L_{BC} + L_{CD}\right) - W_c\left(L_{BC} + \frac{L_{CD}}{2}\right) - C_m\left(L_{BC}\right) - W_B\left(\frac{L_{BC}}{2}\right) + M_B = 0$$

M_c = 0.838 Nm

M_B = 1.337 Nm

Based on the load calculations, the MG996R servo motor was selected, offering a torque of 0.838 Nm. This motor was chosen not only for its suitable specifications but also because it is more affordable than other motors with comparable performance. Since I need more torque at B, I used MG90s, which has a maximum torque of 1.86Nm.

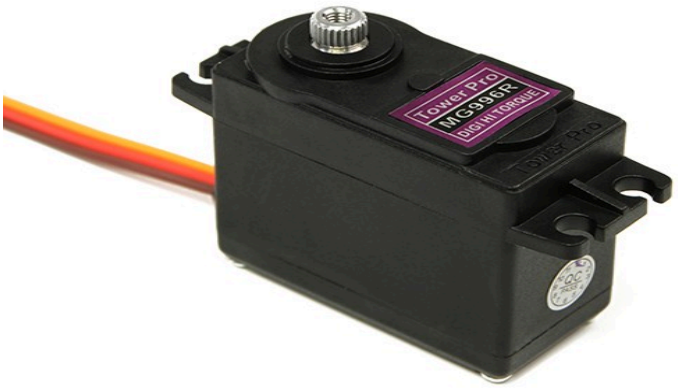


Figure 5**. MG996R servo motor**



Figure 6**. MG90S servo motor**



Figure 7**. PS4 controller**

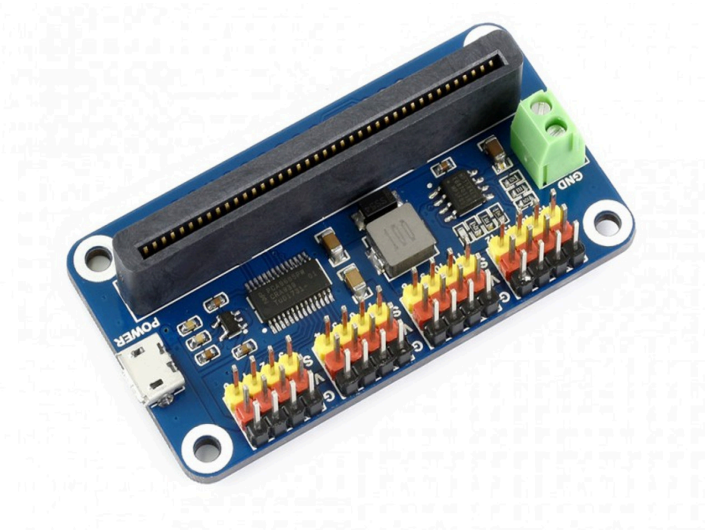


Figure 8**. Servo Driver**

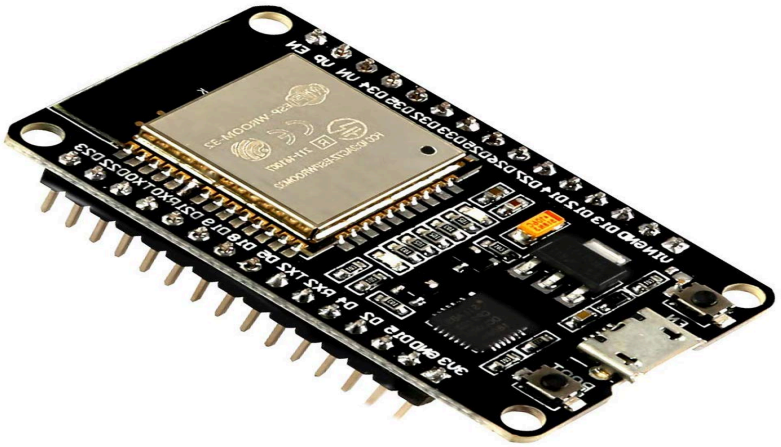


Figure 9**. ESP32 microcontroller**

5.3 Electronic Circuit Design and Wiring

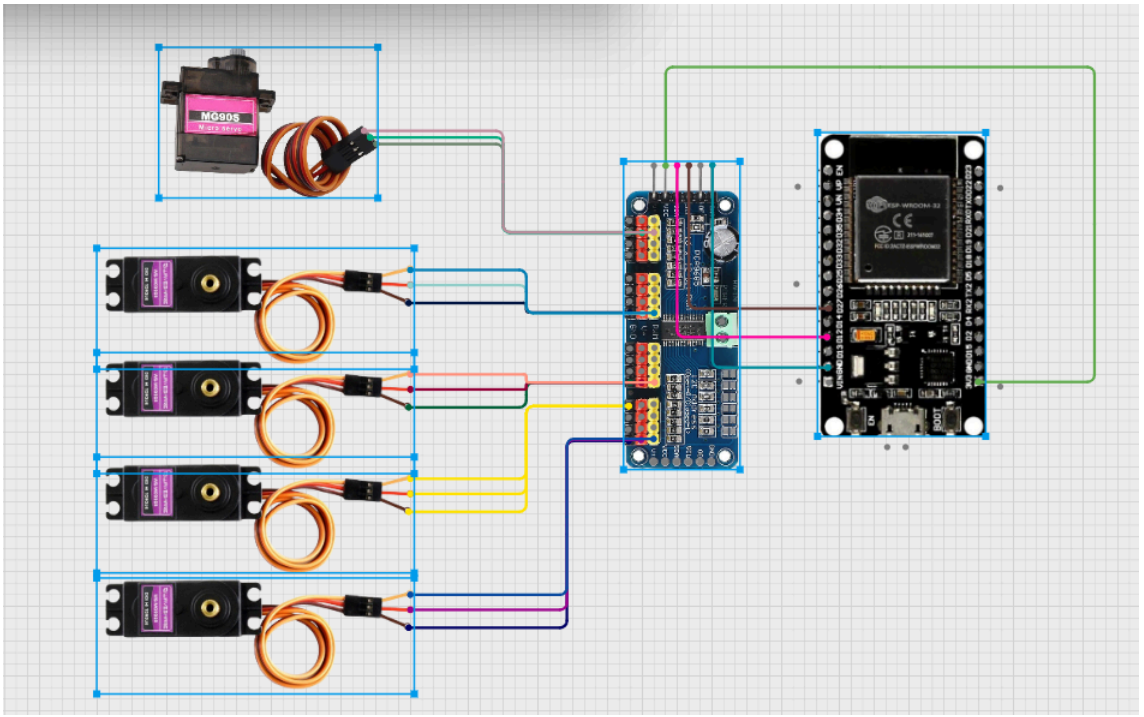


Figure 10. **Electronic circuit diagram**

6. Testing and Validation

Before integrating the full robotic arm system, all major components were tested individually to confirm their proper functionality and to minimize the risk of errors during assembly. The testing process began with the MG996R servo motors. Each servo was operated using a simple PWM control script uploaded to the ESP32. This allowed evaluation of the servo's full range of motion, its response time, and its stability during continuous operation. Special attention was given to identifying any jitter, irregular movement, or signs of overheating. All servo motors performed smoothly throughout these tests, demonstrating consistent responsiveness and confirming that each unit was fully operational.

The ESP32 microcontroller was also tested independently to ensure its reliability as the system's central controller. This included verifying that programs could be uploaded successfully and that communication with the PS4 controller remained stable. The PWM outputs were checked to confirm that they generated clean and accurate control signals for operating the servo motors. The microcontroller met all expectations, exhibiting no functional issues during these standalone tests.

In addition to testing control components, the power supply system was evaluated carefully. Using a multimeter, the output voltage was measured to verify that it matched the required specifications for powering the servo motors and microcontroller. The current capability of the power supply was also assessed to ensure it could support multiple servos operating simultaneously. Measurements taken under load confirmed that the power supply delivered stable and sufficient power, making it suitable for the robotic arm system.

Once all individual components were validated, the full system was assembled and integration testing began. The PS4 controller was paired with the ESP32 to verify wireless communication and ensure accurate interpretation of user inputs. During this phase, joystick movements and button presses were monitored to confirm that all commands were transmitted reliably without delays or disruptions.

After confirming communication stability, each joint of the robotic arm was tested through the integrated control system. The mapping between controller input and joint movement was examined to ensure correct behavior, and the servos were observed for smooth, precise motion across their operating ranges. The mechanical structure of the arm was inspected to ensure that no physical interference occurred between components during motion. Additionally, torque performance was evaluated to confirm that each joint could lift and position the arm as required.

Overall, the robotic arm responded accurately and consistently to all control inputs during integration testing. No unexpected behavior or mechanical issues were observed, demonstrating that the system operated reliably as a complete unit. This successful testing process provided confidence in the functionality, stability, and readiness of the robotic arm for further experimentation and project evaluation.

7. Discussion

7.1 Manual Control

This type of control allows the user to move the arm in real-time by sending commands through the joystick buttons and analog sticks. The ESP32 microcontroller receives the signals from the controller and translates them into movements for each servo motor.

Manual control provides a simple and direct way to test the arm's movement and performance. It allows the user to practice precise positioning and adjust the joint angles freely without needing complex software or programming. This approach is particularly valuable in the initial stages of development and testing, as it allows for rapid identification of mechanical or electrical problems.

Although manual control is easy to implement and use, it can be limiting for more complex tasks or when precise and repeatable movements are needed. In future versions, manual control can be combined with automated features like inverse kinematics or pre-programmed motion sequences for better performance.

7.2 Challenges Faced

During the design process, I encountered challenges in creating strong joints between PLA parts. A mechanical connection using screws and nuts was implemented, and to facilitate this, a small feature was incorporated to allow the bolts to be secured with nuts without inserting screws directly into the thin PLA layers.

8. Conclusion

This thesis presents the design and development of a robotic arm intended for pick-and-place operations. The arm was made from PLA plastic and powered by servo motors, which moved the joints. However, the motors can only rotate less than 180°, which limits the arm's range and the positions it can reach. The system incorporates four degrees of freedom, providing necessary motion capabilities while ensuring the design remains straightforward and economical.

8.1 Summary of Key Findings

The development of a 4-DOF robotic arm using PLA material and MG996R servo motors successfully demonstrated the feasibility of low-cost, joystick-controlled pick-and-place automation. The use of a PS4 controller provided intuitive manual control, while the ESP32 microcontroller ensured responsive communication and motor handling. Despite the limited rotation range of the servo motors (less than 180°), the arm was able to achieve a functional workspace for basic tasks. The mechanical structure proved lightweight and modular, which simplified assembly and allowed for easy modifications. However, torque limitations of the MG996R servos constrained the payload capacity and movement speed. Overall, the design met its goals for functionality, simplicity, and affordability, but future improvements could include higher-torque motors and expanded motion range.

8.2 Future Research Directions

This robotic arm was designed with the possibility to be upgraded to a 6-degree-of-freedom (DOF) system. While the current 4-degree-of-freedom design is effective for fundamental pick-and-place tasks, multiple enhancements could be considered for future development:

1. Torque Improvement

The servo motors used in this project have limited torque, which affects how much weight the arm can carry. In future versions, using stronger motors or adding gear systems could help the arm lift heavier objects more smoothly and safely.

2. Adding More Degrees of Freedom

By increasing the number of joints to 6 DOF, the arm would be able to move more freely and reach more positions. This would allow it to perform more complex movements, such as rotating objects or working around obstacles.

3. Inverse Kinematics

Right now, the arm is controlled manually using a joystick. In the future, adding inverse kinematics software would let the arm move automatically to a chosen point in space by calculating the needed angles for each joint. This would make the arm more accurate and easier to control.

4. IoT (Internet of Things) Features

Since the ESP32 microcontroller supports Wi-Fi and Bluetooth, future versions of the project could include remote control and monitoring. Such an enhancement would enable operation of the robotic arm through a mobile or web-based application and facilitate the logging of its movements.

8.3 Recommendations for Practical Applications

The robotic arm designed in this project can be used in several real-world applications, especially in areas where simple and repetitive tasks are required. Based on its current capabilities, and with some improvements, the following practical uses are recommended:

1. Educational Robotics

The arm can be used as a learning tool in schools and universities to teach students about robotics, electronics, and programming. Its simple design and low cost make it ideal for hands-on training.

2. Light Pick-and-Place Operations

In environments like small workshops or packaging lines, the arm can be used to move small objects from one place to another. It can help automate basic tasks such as sorting, placing, or assembling lightweight parts.

3. Prototype for Advanced Systems

The existing design offers a platform that can be expanded into more complex and capable robotic systems. With further development—such as adding more degrees of freedom, stronger motors, or AI control—it can be adapted for more complex tasks like 3D printing, drawing, or machine operation.

9. References

[1] Pawar H, Darade D, Ingle M, Kale S, Deshmukh R. Pick and Place Robotic Arm: A Review. *International Research Journal of Engineering and Technology (IRJET)*. 2021;8(2):1970–1973. https://d1wqtxts1xzle7.cloudfront.net/68831557/IRJET_V8I2311-libre.pdf

[2] Lynch KM, Park FC. 2.2. Degrees of Freedom of a Robot [Internet]. Modern Robotics. 2017 Sep 1. https://modernrobotics.northwestern.edu/nu-gm-book-resource/2-2-degrees-of-freedom-of-a-robot/

[3] Nakamura K. Development of a 4-DOF Serial Robotic Manipulator for Pick-and-Place Tasks [Internet]. 2018. https://kainakamura.com/rbe3001/report.pdf

[4] Deokar M, Deshmukh V, Dhomane S. Design and Development of 4 DOF Robot Arm for Picking and Placing. In: *Proceedings of the 2023 7th International Conference on Computing Methodologies and Communication (ICCMC)*. 2023. https://dl.acm.org/doi/10.1145/3608367.3608373

[5] López-Mata J, Rodríguez-Rodríguez J, Cruz-Ledesma LF, Feregrino-Uribe C, Rodríguez-Reséndiz J. Physical Modeling of a 4-DOF Manipulator for Learning. *Int J Mech Eng Robot Res* [Internet]. 2024. https://www.ijmerr.com/2024/IJMERR-V13N3-392.pdf

[6] Cacacho E, Reina G, Milella A, Mazzia V. Simplified 4-DOF Manipulator for Rapid Robotic Apple Harvesting. *Comput Electron Agric* [Internet]. 2023; 205:107621. https://www.sciencedirect.com/science/article/abs/pii/S016816992200494X

[7] Nallusamy R, Karthikeyan K. Design of a 4-DOF Robot Manipulator with Optimized Algorithm for Inverse Kinematics. *Indian J Sci Technol* [Internet]. 2015.https://www.researchgate.net/publication/288669075

[8] Aspelund S, Patel P, Lee M-H, Kagerer FA, Ranganathan R, Mukherjee R. Controlling a robotic arm for functional tasks using a wireless head-joystick: A case study of a child with congenital absence of upper and lower limbs. PLoS One. 2020. https://doi.org/10.1371/journal.pone.0226052

[9] Elfasakhany A, Yanez E, Baylon K, Salgado R. Design and development of a competitive low-cost robot arm with four degrees of freedom. *Mod Mech Eng*. 2011;1(2):47–55.
https://www.scirp.org/html/8833.html

[10] Mustaffa MF, Ahmad Shauri RL, Roslan MI, Idris MA. Development of a 4-DOF robotic arm: prototype design. Int J Eng Technol. 2018 Nov 27;7(4.18):429–32.
https://www.researchgate.net/publication/382628174_Development_of_a_4-DOF_Robotic_Arm_Prototype_Design

10. Appendices

10.1 Source Code and Algorithms

```
#include <PS4Controller.h>
#include <ESP32Servo.h>

#define SERVO_FORWARD_STEP_ANGLE 1
#define SERVO_BACKWARD_STEP_ANGLE -1

struct ServoPins
{
  Servo servo;
  int servoPin;
  String servoName;
  int initialPosition;
};
std::vector<ServoPins> servoPins =
{
  { Servo(), 27 , "Base", 90},
  { Servo(), 26 , "Shoulder", 90},
  { Servo(), 25 , "Elbow", 90},
  { Servo(), 33 , "Gripper", 90},
};

bool gripperSwitch = false;


void writeServoValues(int servoIndex, int servoMoveStepSize, bool
servoStepSizeIsActualServoPosition = false)
{
  int servoPosition;
  if (servoStepSizeIsActualServoPosition)
  {
    servoPosition = servoMoveStepSize;
```

```
  }
  else
  {
    servoPosition = servoPins[servoIndex].servo.read();
    servoPosition = servoPosition + servoMoveStepSize;
  }
  if (servoPosition > 180 || servoPosition < 0)
  {
    return;
  }

  servoPins[servoIndex].servo.write(servoPosition);
}


void notify()
{
  int rx =(PS4.data.analog.stick.rx);  //Base       =>  Right stick
- x axis
  int ry =(PS4.data.analog.stick.ry);  //Shoulder   =>  Right stick
- y axis
  int ly =(PS4.data.analog.stick.ly);  //Elbow      =>  Left stick
- y axis
  int lx =(PS4.data.analog.stick.lx);  //Gripper    =>  Left stick
- x axis

  if (rx > 50)
  {
    writeServoValues(0, SERVO_BACKWARD_STEP_ANGLE);
  }
  else if (rx < -50)
  {
    writeServoValues(0, SERVO_FORWARD_STEP_ANGLE);
  }

  if (ry > 50)
  {
    writeServoValues(1, SERVO_BACKWARD_STEP_ANGLE);
  }
  else if (ry < -50)
```

```
  {
    writeServoValues(1, SERVO_FORWARD_STEP_ANGLE);
  }

  if (ly > 50)
  {
    writeServoValues(2, SERVO_FORWARD_STEP_ANGLE);
  }
  else if (ly < -50)
  {
    writeServoValues(2, SERVO_BACKWARD_STEP_ANGLE);
  }

  if (lx > 50)
  {
    writeServoValues(3, SERVO_BACKWARD_STEP_ANGLE);
  }
  else if (lx < -50)
  {
    writeServoValues(3, SERVO_FORWARD_STEP_ANGLE);
  }

  if (PS4.event.button_down.r2)
  {
    gripperSwitch = !gripperSwitch;  //Toggle gripper close / open
    gripperSwitch ? writeServoValues(3, 170, true) :
writeServoValues(3, 100, true) ;
  }



  delay(10);

}

void onConnect()
{
  Serial.println("Connected!.");
}
```

```
void onDisConnect()
{
  Serial.println("Disconnected!.");
}




void setup()
{
  setUpPinModes();
  Serial.begin(115200);
  PS4.attach(notify);
  PS4.attachOnConnect(onConnect);
  PS4.attachOnDisconnect(onDisConnect);
  PS4.begin();
  Serial.println("Ready.");
}

void loop()
{
}
```

10.2 Additional Pictures

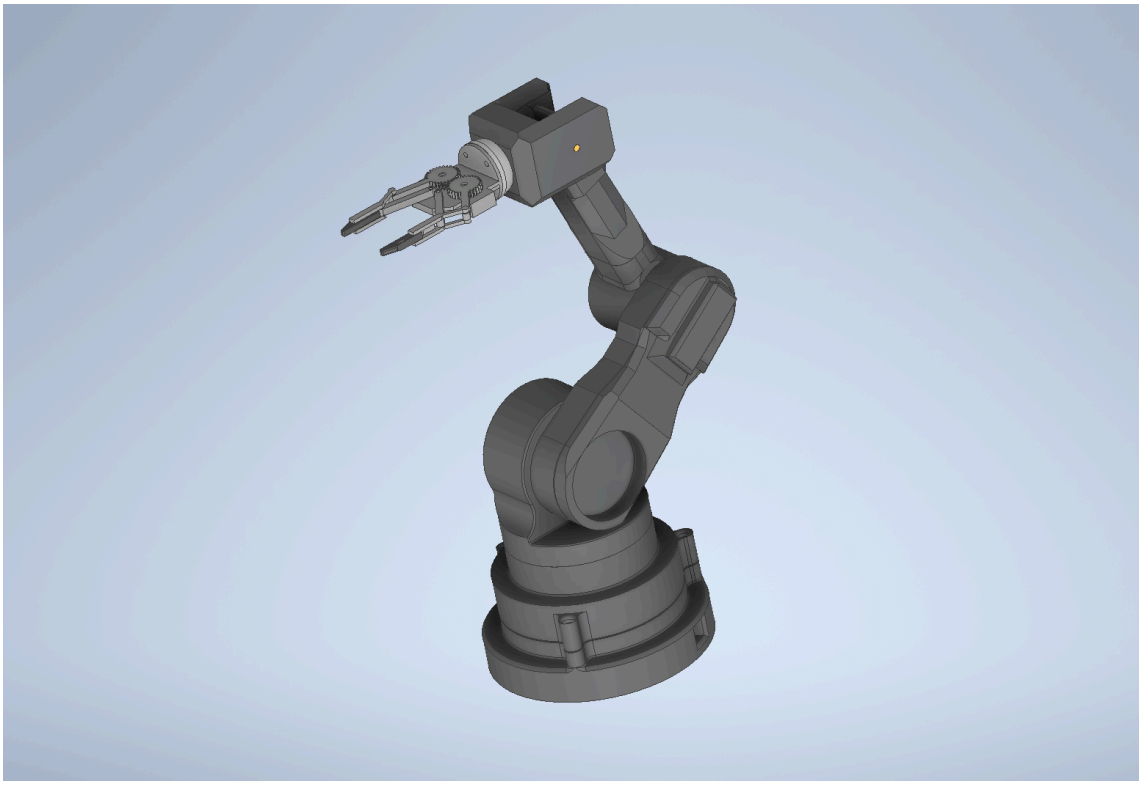


Figure 11**. Design of the robot arm**