A

Project Stage-II Report on

ROBOTIC ARM MODEL

Submitted in the partial fulfillment of the requirements of Semester-VIII

For the Award of the Degree of Bachelor of Technology (B.Tech) in

Electronics and Communication Engineering

Submitted by

Anshuman	PRN:	2114110433
Utkarsh Maurya	PRN:	2114110490
Mayank Mankar	PRN:	2114110488

Under the Guidance of

DR. ARUNDHATI A. SHINDE

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

BHARATI VIDYAPEETH DEEMED TO BE UNIVERSITY

COLLEGE OF ENGINEERING, PUNE - 411 043

SEMESTER-VIII, ACADEMIC YEAR: 2024-2025

Department of Electronics and Communication Engineering
BHARATI VIDYAPEETH (DEEMED TO BE UNIVERSITY)
COLLEGE OF ENGINEERING, PUNE

CERTIFICATE

This is to certify that the project phase-II report on **"Robotic Arm Model"** submitted by

Anshuman	**PRN:**	**2114110433**
Utkarsh Maurya	**PRN:**	**2114110490**
Mayank Mankar	**PRN:**	**2114110488**

in partial fulfillment of the requirements for the award of degree of Bachelor of Technology (B.Tech) in Electronics and Communication Engineering.

Dr. Arundhati A. Shinde
Guide, ECE Dept.,
BV(DU), COE, Pune

Dr. Dhiraj M. Dhane
Project Co-ordinator
BV(DU), COE, Pune

Prof.(Dr.) Arundhati A. Shinde
Head of the ECE Department
BV(DU), COE, Pune

Date:

Place: Pune

Acknowledgments

We would like to express our sincere gratitude to Prof. (Dr.) Arundhati A. Shinde, our Mentor and Head of the Department, for her invaluable guidance, continuous support, and encouragement throughout the project. We would also like to extend our thanks to our Project Coordinator Prof. (Dr.) Dhiraj M. Dhane for his support and coordination throughout the project. In addition, we thank our department and institution for giving us the opportunity to undertake this project.

Anshuman

Utkarsh Maurya

Mayank Mankar

Contents

List of Figures

List of Tables

Abstract

This project involves the design and development of a simple pick-and-place robotic arm for educational and small-scale applications. The arm is controlled using an Arduino microcontroller and powered by servo motors, enabling basic object movement tasks. The focus is on creating a cost-effective and easy-to-assemble system suitable for learning and demonstration purposes.

The project includes hardware assembly, circuit design, and basic programming to control the arm's movements. While the system is not designed for high precision, it effectively demonstrates the fundamental principles of robotics and automation. This project serves as a foundation for understanding robotic arm functionality and can be expanded in the future with additional features.

Chapter 1

Introduction

1.1 Overview

The advent of automation has significantly transformed the landscape of industrial manufacturing, leading to increased efficiency, precision, and safety in production processes. Among the most impactful innovations in this domain are industrial robotic arms, which have become indispensable tools in various sectors, including automotive, electronics, food processing, and pharmaceuticals. These robotic systems are designed to perform a wide range of tasks, from simple pick-and-place operations to complex assembly procedures, thus enabling manufacturers to streamline operations and improve output quality. Industrial robotic arms are characterized by their flexibility, repeatability, and ability to operate in environments that are challenging or hazardous for human workers. Equipped with advanced sensors and actuators, these robotic arms can perform intricate movements with high accuracy, reducing the risk of errors and enhancing production speeds. The integration of artificial intelligence (AI) and machine learning algorithms further augments their capabilities, allowing them to adapt to changing conditions and optimize their performance in real-time. As industries increasingly seek to enhance productivity while minimizing costs, the adoption of robotic arms is on the rise. However, the implementation of these technologies also presents challenges, such as the need for substantial initial investment, skilled workforce training, and effective programming. This paper aims to explore the various aspects of industrial robotic arms, including their design principles, operational mechanisms, and diverse applications across different sectors. By examining the current state of industrial robotics, we hope to provide insights into the future trajectory of robotic arm technology and its potential impact on manufacturing practices.

1.2 Literature review

Various studies and projects have explored the use of sensors and control systems to enhance the functionality of robotic arms. Here is an overview of relevant research and projects that contribute to the development of robotic arm technologies:

1.2.1 Design and Development of a Robotic Arm

B. K. Kumar, et al. [1] focused on developing a robotic arm tailored to assist elderly and specially challenged individuals with daily tasks, like feeding. This robotic arm has 5 degrees of freedom (DOF) and utilizes kinematic modeling in MATLAB for precise positioning, with control handled by an Arduino Mega2560. The arm is made from lightweight aluminum and uses stepper and DC motors for joint movements, supplemented by sensors to ensure accuracy. A user-friendly GUI, built in Processing3, allows intuitive control, enabling users with limited mobility to operate the device effectively. Future improvements aim to enhance real-time feedback and closed-loop control for expanded functionality.

1.2.2 Pick and Place Robotic Arm

Sharath Surati et al. [2] explored integrating AI into robotic arms, enhancing industrial applications in sectors like medicine and defense. The study categorized robotic arm types and their industrial advantages, emphasizing how AI, coupled with object recognition technologies, optimizes sorting, assembly, and quality control. Practical implementations using Arduino and servo motors were highlighted, showcasing tasks like welding and sorting. Innovations in control mechanisms, real-time feedback systems, and sensor technologies further enhanced precision, showcasing the adaptability of robotic arms in dynamic environments.

1.2.3 Robotic Arm Movement Grasping System Based on MYO

Ye Wei et al. [3] developed a robotic arm system leveraging surface electromyography (sEMG) and inertial measurement unit (IMU) signals for gesture recognition and control. The study proposed using MYO to gather sEMG and IMU signals for continuous motion estimation, processed through filtering techniques and machine learning models. The system aligns the robotic arm's movements with the user's gestures, enhancing applications in prosthetics and rehabilitation. This approach bridges human intentions with robotic execution, emphasizing real-time interaction and precise control through advanced kinematic modeling.

1.2.4 Design and Research of an Automatic Grasping System for a Robot Arm

Xiaofan Liu et al. [4] proposed an advanced grasping system utilizing Kinect and visual image capture technology. By integrating ResNet50 and ELU activation functions, the system achieved high gesture recognition accuracy. The use of a Kalman filter enhanced real-time motion tracking, enabling the arm to mimic human gestures with 96-98 accuracy. This innovation addresses the limitations of traditional robotic programming, offering a versatile and precise solution for dynamic environments, such as healthcare and assembly lines, through deep learning-enhanced robotic control.

1.2.5 Developing a Robotic Arm for Public Service and Industry

Florin Covaciu et al. [5] created an advanced software interface for controlling robotic arms in manufacturing and healthcare. By implementing sophisticated control algorithms, the system improved real-time monitoring and precision, simplifying programming for operators with varied expertise. The study emphasized the interface's role in increasing operational efficiency, reducing errors, and minimizing downtime. This accessible design fosters wider adoption of robotic technologies, highlighting their transformative potential across industries through enhanced accuracy and user-friendly features.

1.2.6 Robotic Arm Control System Based on AI Wearable Acceleration Sensor

Liang Chen et al. [6] designed a robotic arm control system using AI wearable acceleration sensors to enhance precision and adaptability. The system utilizes advanced filtering techniques and a Proportional-Integral (PI) algorithm for accurate speed control. A servo control system equipped with MEMS sensors facilitated responsive motion, while wireless communication enabled practical industrial and medical applications. The integration of temperature sensing at the gripper ensures safe handling, demonstrating the system's versatility and potential in diverse scenarios requiring delicate operations. The various methodologies used in the robotics arm are shown in Table 1.1.

Table 1.1: Applications using various technologies

Sources	**Technology Used**	**Key Findings**	**Challenges**	**Applications**
[1]	Arduino Mega 2560, Stepper motors, MATLAB Kinematic modeling	Developed a 5-DOF arm for elderly and disabled individuals	High cost and complexity	Assistive robotics for healthcare
[2]	Arduino, Servo motors, AI for object detection	Integrated AI for enhanced industrial applications	Requires advanced programming and hardware	Sorting, welding, quality control

Sources	Technology Used	Key Findings	Challenges	Applications
[3]	MYO armband, Machine Learning for gesture recognition	Gesture control using sEMG and IMU sensors	Limited to gesture-based control, not suitable for automation	Prosthetics and rehabilitation
[4]	Kinect, ResNet50, Kalman filter for real-time tracking	Achieved 96-98% accuracy in gesture recognition using deep learning	High computational requirements	Healthcare and industrial automation
[5]	Advanced control algorithms, AI-based monitoring	Developed a software interface for real-time control	Expensive and complex to implement	Smart automation in manufacturing
[6]	MEMS sensors, PI control algorithm, wireless communication	Real-time speed and movement adaptation using wearable sensors	Limited to specific industrial applications	Industrial robotics, delicate handling

Chapter 2

Need of the project

2.1 Objectives

The Pick and Place Robotic Arm project aims to provide a practical and educational approach to understanding automation and robotics concepts. The primary objectives are as follows:

- **Simplified Learning of Automation Concepts:** To demonstrate the fundamental principles of robotics and automation in a simplified and accessible manner, making it suitable for learners and enthusiasts.
- **Cost-Effective Prototyping:** To showcase how functional prototypes can be created using everyday materials like cardboard, emphasizing resourcefulness.
- **Motor Control and Integration:** To provide hands-on experience with controlling motors and integrating them with microcontrollers like Arduino for operational accuracy.
- **Designing Lightweight Mechanisms:** To explore effective techniques for handling lightweight objects and understanding the associated challenges.
- **Practical Application of Theory:** To bridge the gap between theoretical knowledge and real-world applications by incorporating concepts from mechanics, electronics, and programming.

Chapter 3

Project Requirements

3.1 Components

1. Servo Motor (x4)
2. Arduino Mega 2560 Microcontroller
3. Connecting Wires
4. Current Booster
5. Dual Axis XY Joystick Module
6. USB -USB cable
7. Robotic Arm Kit

3.2 Component Specifications

3.2.1 Servo Motor (SG90)

The SG90 servo motor is a commonly used micro servo in robotics and automation. Each servo provides 180° rotational freedom, enabling full-range articulation at each joint. Below are its specifications:

Specifications:

- **Model:** SG90
- **Operating Voltage:** Typically 4.8V to 6V (may vary depending on the servo model).
- **Torque:** Approximately 1.8 kg·cm at 4.8V.
- **Speed:** Around 0.1 sec/60° at 6V.
- **Control Signal:** PWM (Pulse Width Modulation) signal with a duty cycle that corresponds to the desired angle.
- **Rotation Range:** Standard SG90 servos rotate from 0° to 180°.
- **Applications:** Used in robotics, RC vehicles, and automated systems for precise angular control.
- The Servo Motor which will be used in robotics arm, as shown in figure 3.1.

Figure 3.1: A servo motor used in robotic applications (*Source:* `https://www.amazon.in/Robodo-Electronics-Tower-Micro-Servo/dp/B00MTFFAE0`)

3.2.2 Arduino Mega 2560

Specifications:

- **Microcontroller:** ATmega2560.

- **Operating Voltage:** 5V.
- **Input Voltage (recommended):** 7-12V.
- **Digital I/O Pins:** 54 (15 PWM outputs).
- **Analog Input Pins:** 16.
- **Flash Memory:** 256 KB (8 KB used by bootloader).
- **SRAM:** 8 KB.
- **EEPROM:** 4 KB.
- **Clock Speed:** 16 MHz.
- **USB Port:** Used for programming and power supply.
- The Arduino Mega 2560 which will be used in robotics arm, as shown in figure 3.2.

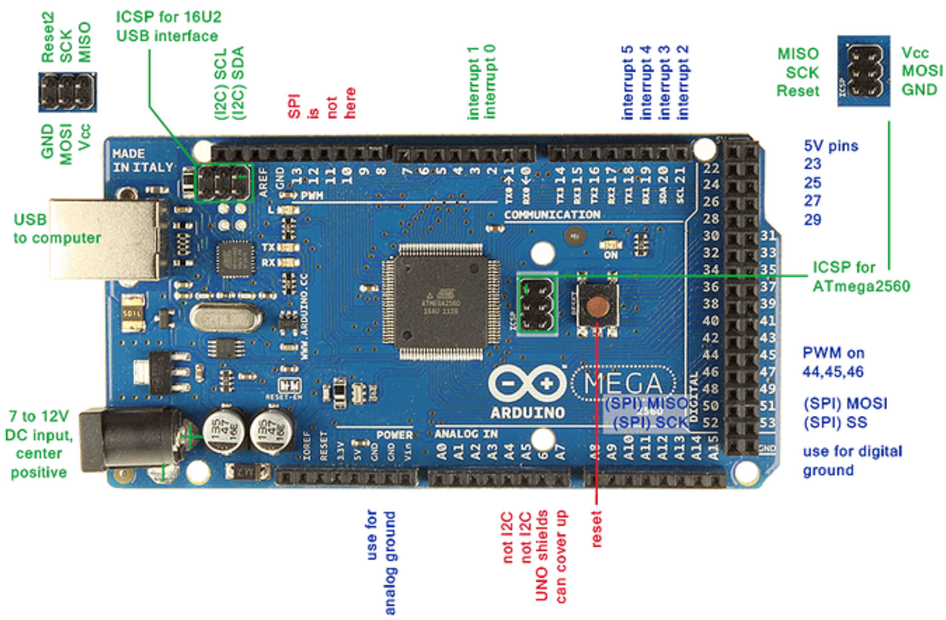


Figure 3.2: Arduino Mega 2560 board with pinouts (*Source:* *https://forum.arduino.cc/t/arduino-mega2560-r3-pinouts-photo/123330*)

Chapter 4

Design of Robotic Arm

4.1 Block Diagram of Robotic Arm

The block diagram of the robotic arm represents the key components and their interconnections essential for its functionality, as shown in Fig. 4.1.

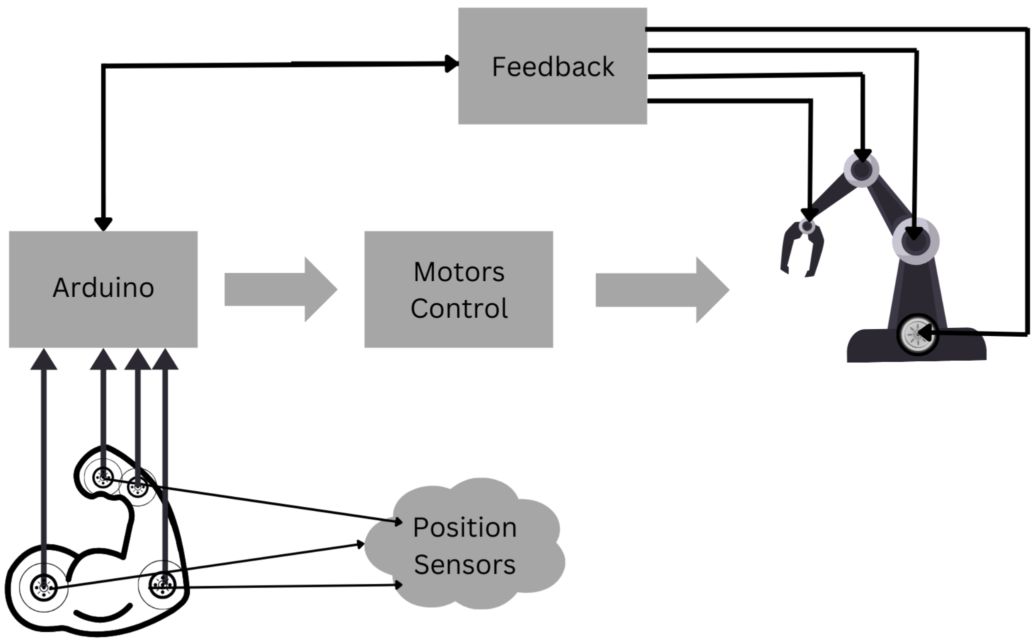


Figure 4.1: Block Diagram of a Robotic Arm

4.2 Circuit Diagram of Robotic Arm

The circuit diagram illustrates the electrical connections and components involved in controlling the robotic arm, as shown in Fig. 4.2.

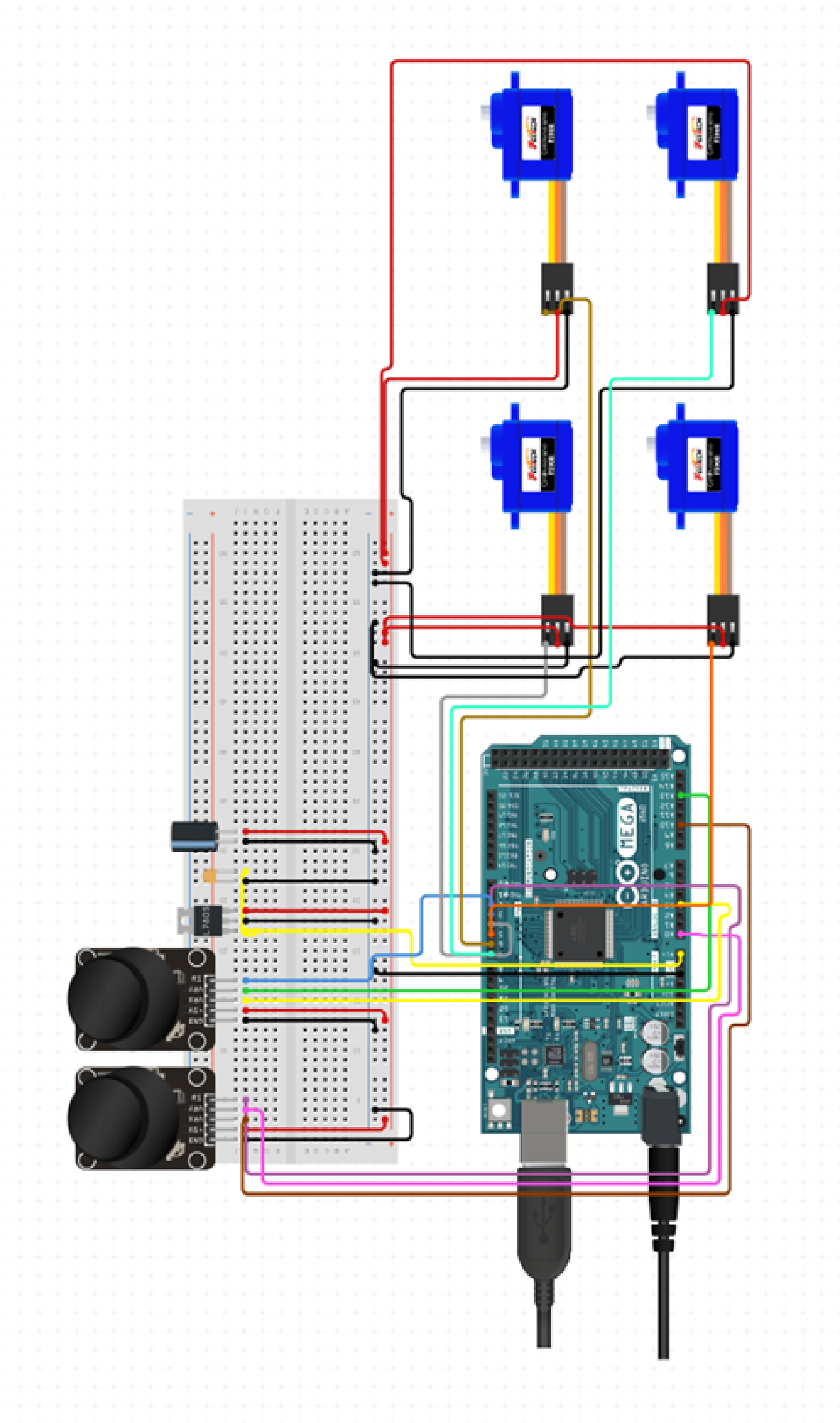


Figure 4.2: Circuit Diagram of Robotic Arm

Table 4.1: Joystick 1 to Arduino Connections, as shown in Fig. 4.2.

Joystick 1 Pin	Arduino Pin
5V	5V
GND	GND
VRX	A0
VRY	A1

Table 4.2: Joystick 2 to Arduino Connections, as shown in Fig. 4.2.

Joystick 2 Pin	Arduino Pin
5V	5V
GND	GND
VRX	A2
VRY	A3

Table 4.3: Servo Motor to Arduino Connections, as shown in Fig. 4.2.

Servo Component	Wire Color	Arduino Pin
Power (5V)	Red	5V
Ground (GND)	Black	GND
Servo 1 Signal	Orange	6
Servo 2 Signal	Orange	9
Servo 3 Signal	Orange	10
Servo 4 Signal	Orange	11

4.3 Robotic Arm Model

4.3.1 STEP 1: BOX STRUCTURE ASSEMBLY

Fix the servo motor with the help of M2.5 type screws and nuts. To fix all the parts mentioned above, we have used M3 type 10mm screws and nuts respectively, as shown in Fig. 4.11.

TOP VIEW:

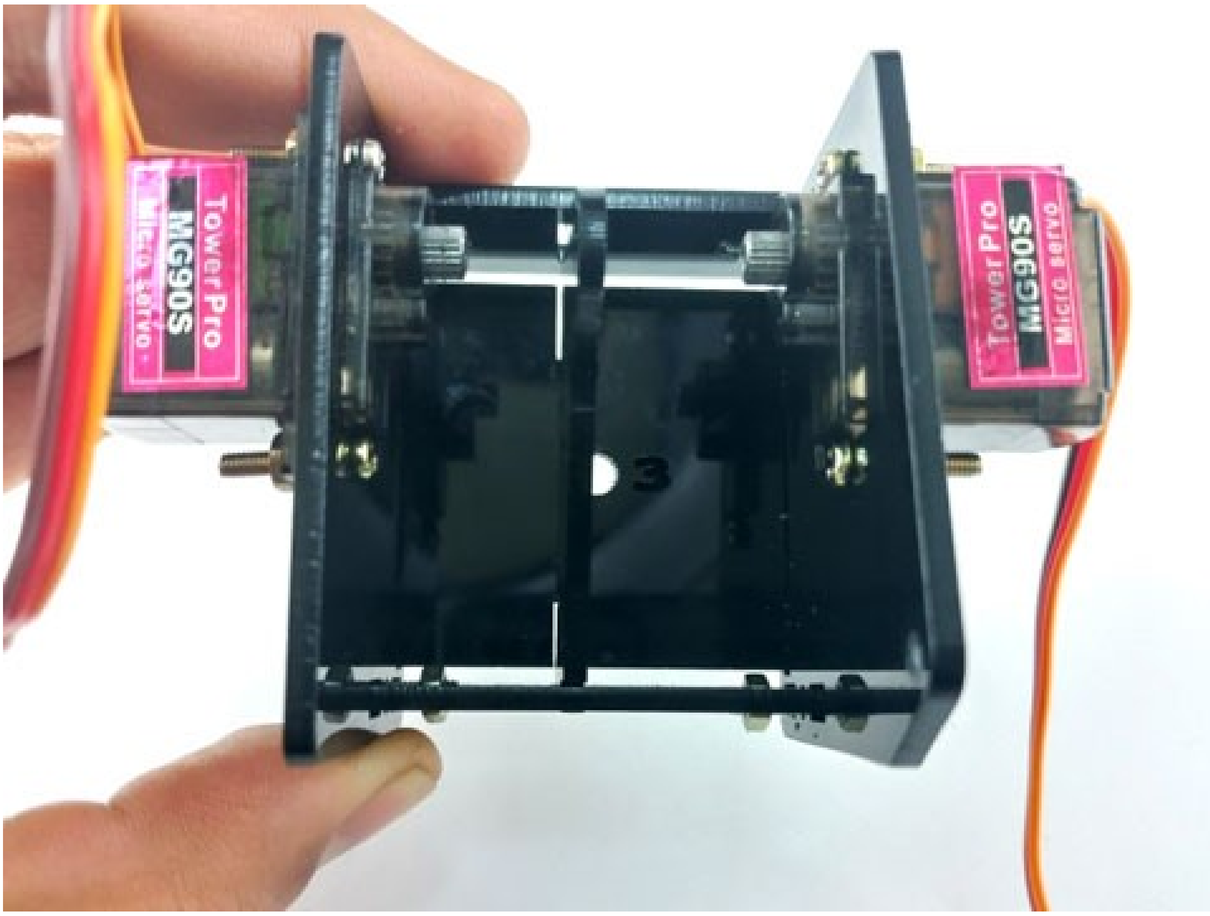


Figure 4.3: Box Structure Assembly – Top View

SIDE VIEW

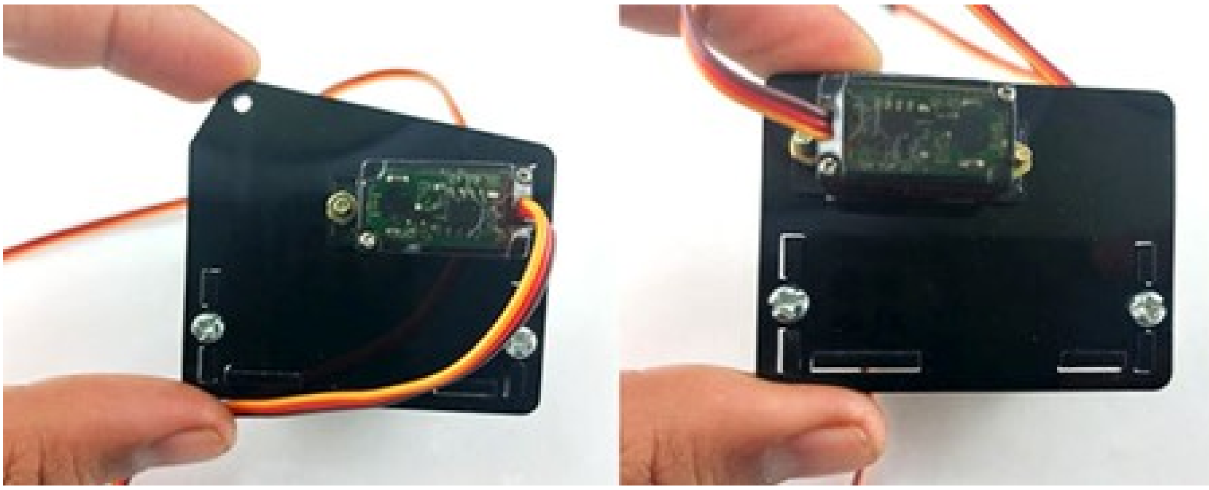


Figure 4.4: Box Structure Assembly – Side View

4.3.2 STEP 2: BRIDGE AND CLIPS ASSEMBLY

Figure 4.5: Bridge and Clips Assembly – Structural View

4.3.3 STEP 3: CLIPPER ASSEMBLY

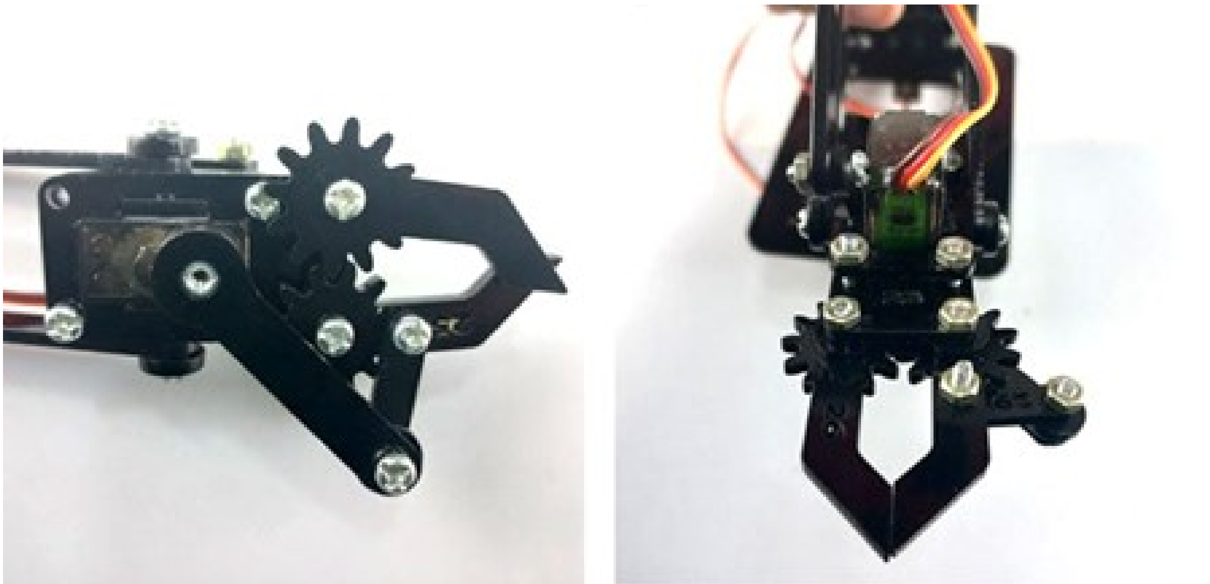


Figure 4.6: Clipper Assembly – Fixed Component View

4.3.4 STEP 4: BASE MOUNT ASSEMBLY

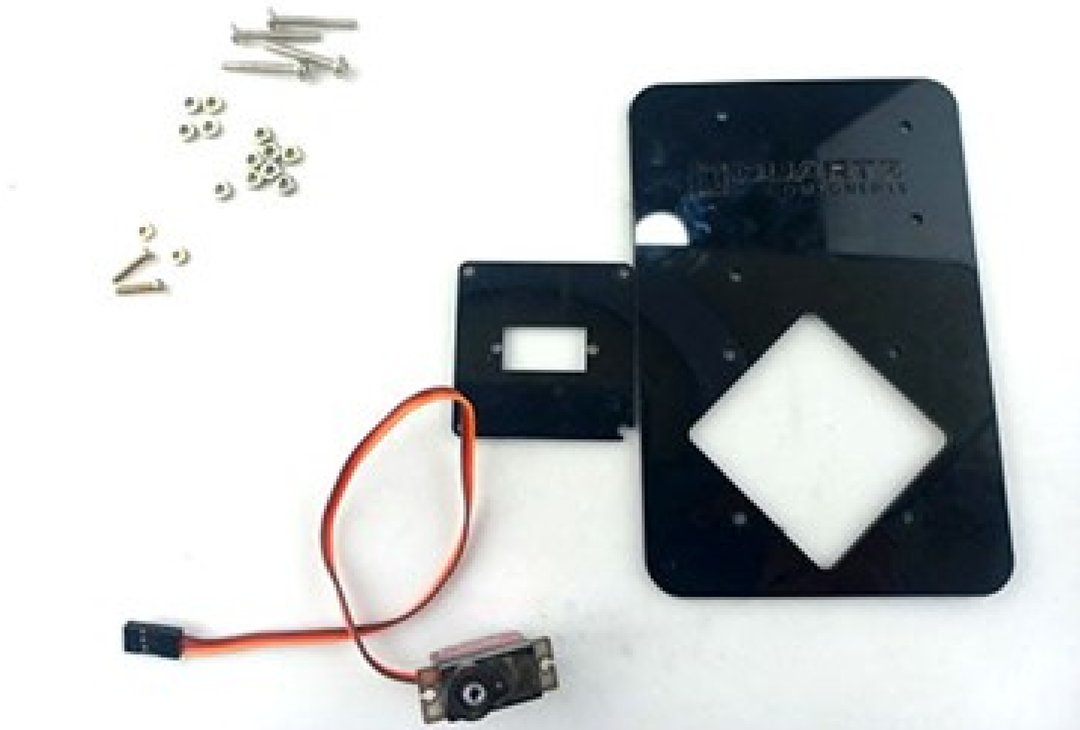


Figure 4.7: Base Mount Assembly – Side Perspective

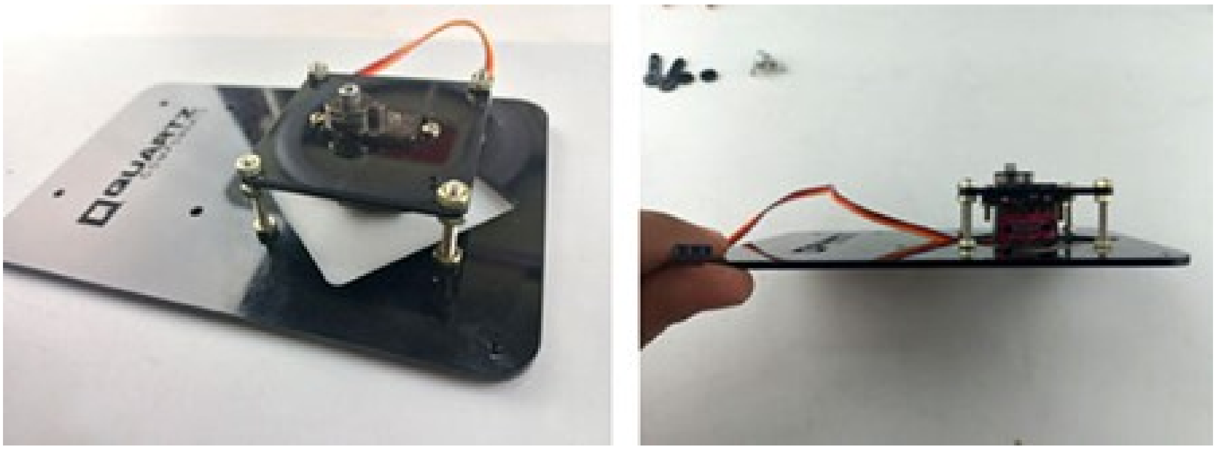


Figure 4.8: Base Mount Assembly – Top Perspective

4.3.5 STEP 5: BODY ATTACHMENT

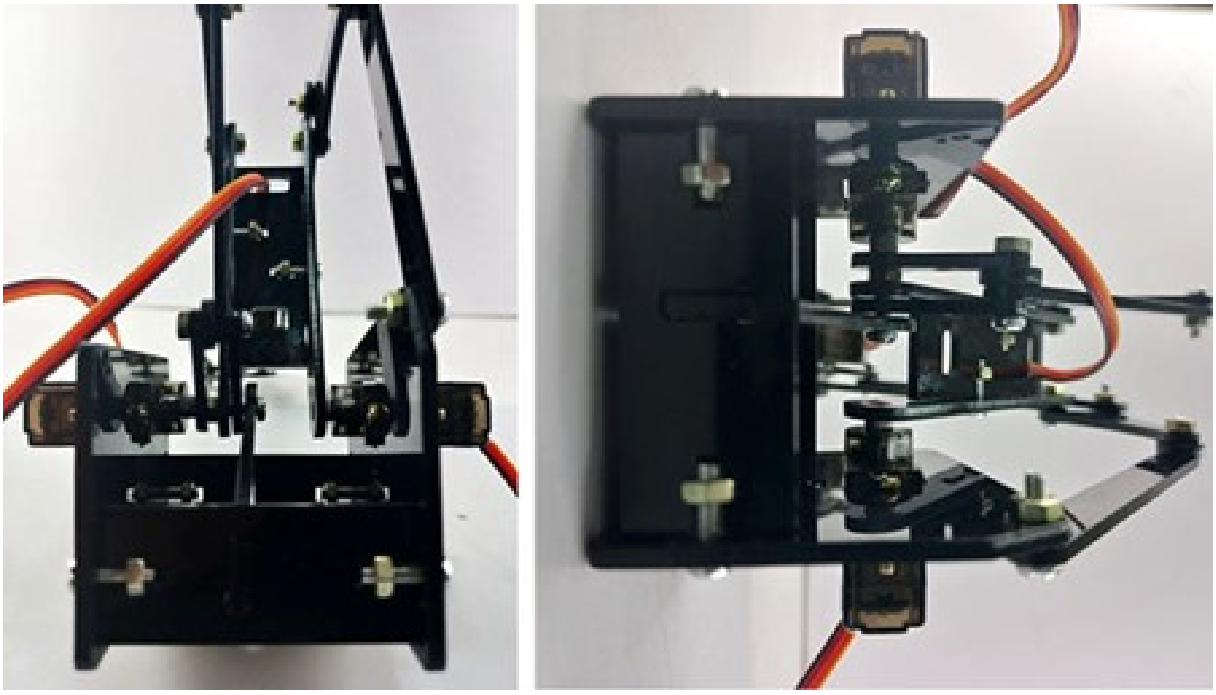


Figure 4.9: Body Attachment to Base and Clipper

4.3.6 STEP 6: CLIPPER AND BASE MOUNT ATTACHMENT TO THE BODY

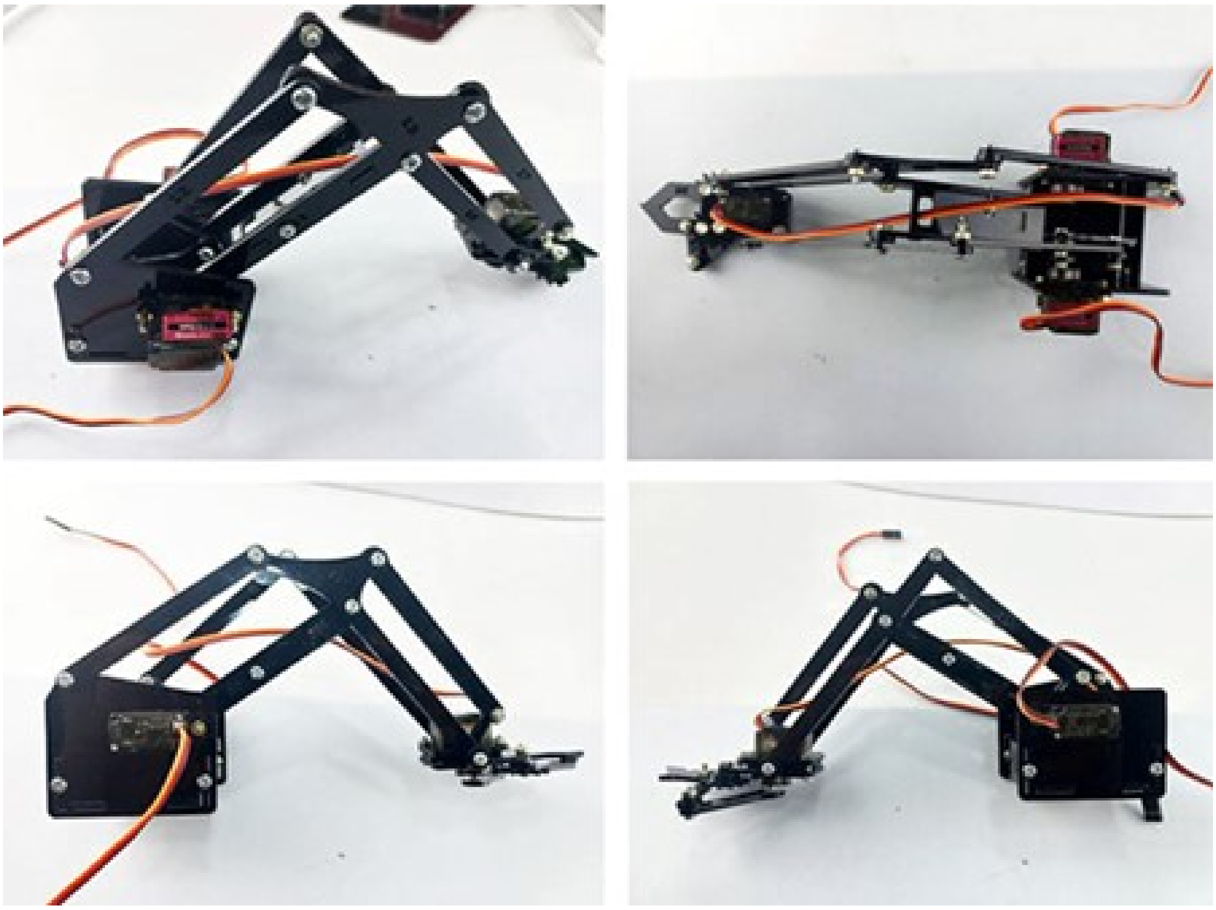


Figure 4.10: Clipper and Base Mount Final Assembly

4.3.7 Robotic Arm Base Assembly

The 4-DOF(Degrees of Freedom) design allows independent 180° movement at the base, elbow, wrist, and gripper joints, as shown in Fig. 4.11, providing a total working envelope of 23 cm radius.

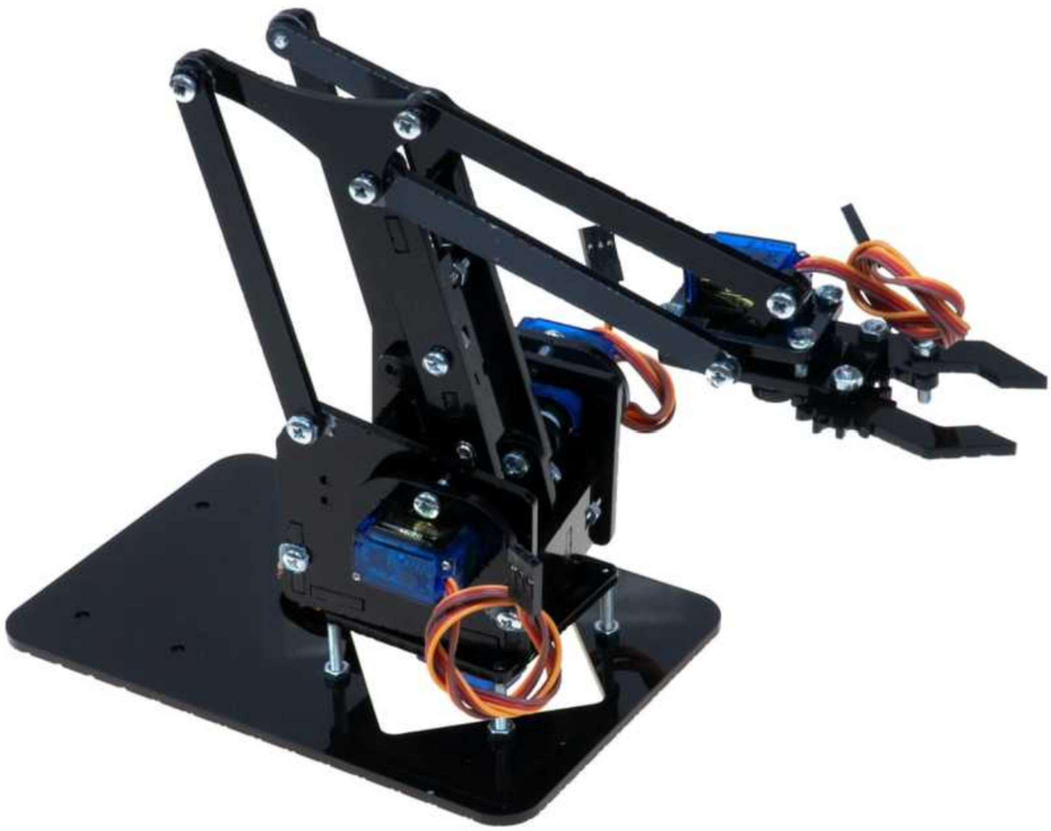


Figure 4.11: Robotic Arm Base Assembly

Kinematic Limitation: While each servo can move up to 180°, practical limits due to link interference or wiring tension may slightly reduce the usable range.

4.3.8 Real Time Implemented Robotic Arm Model

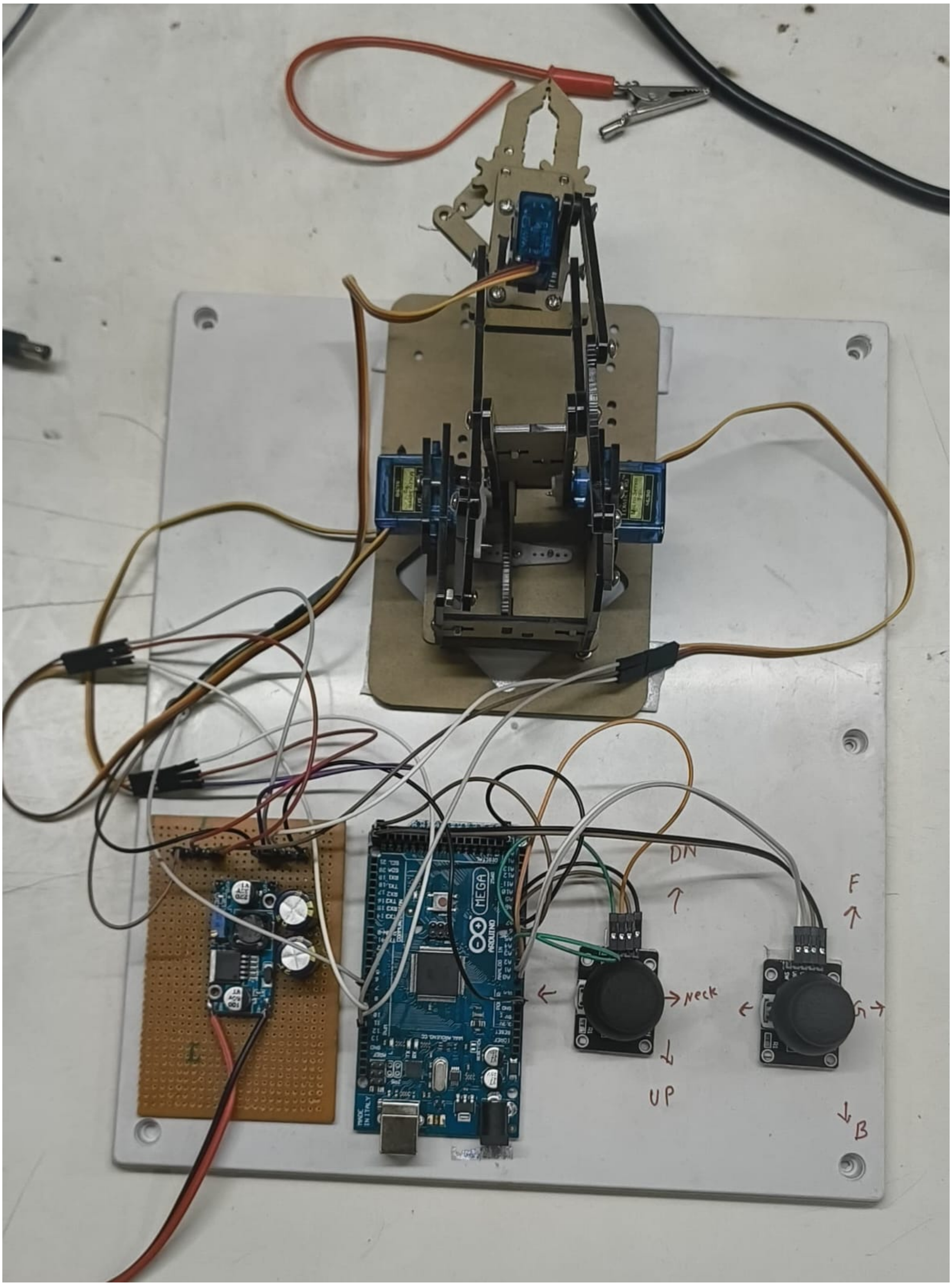


Figure 4.12: Top View Of Real Time Implemented Robotic Arm Model

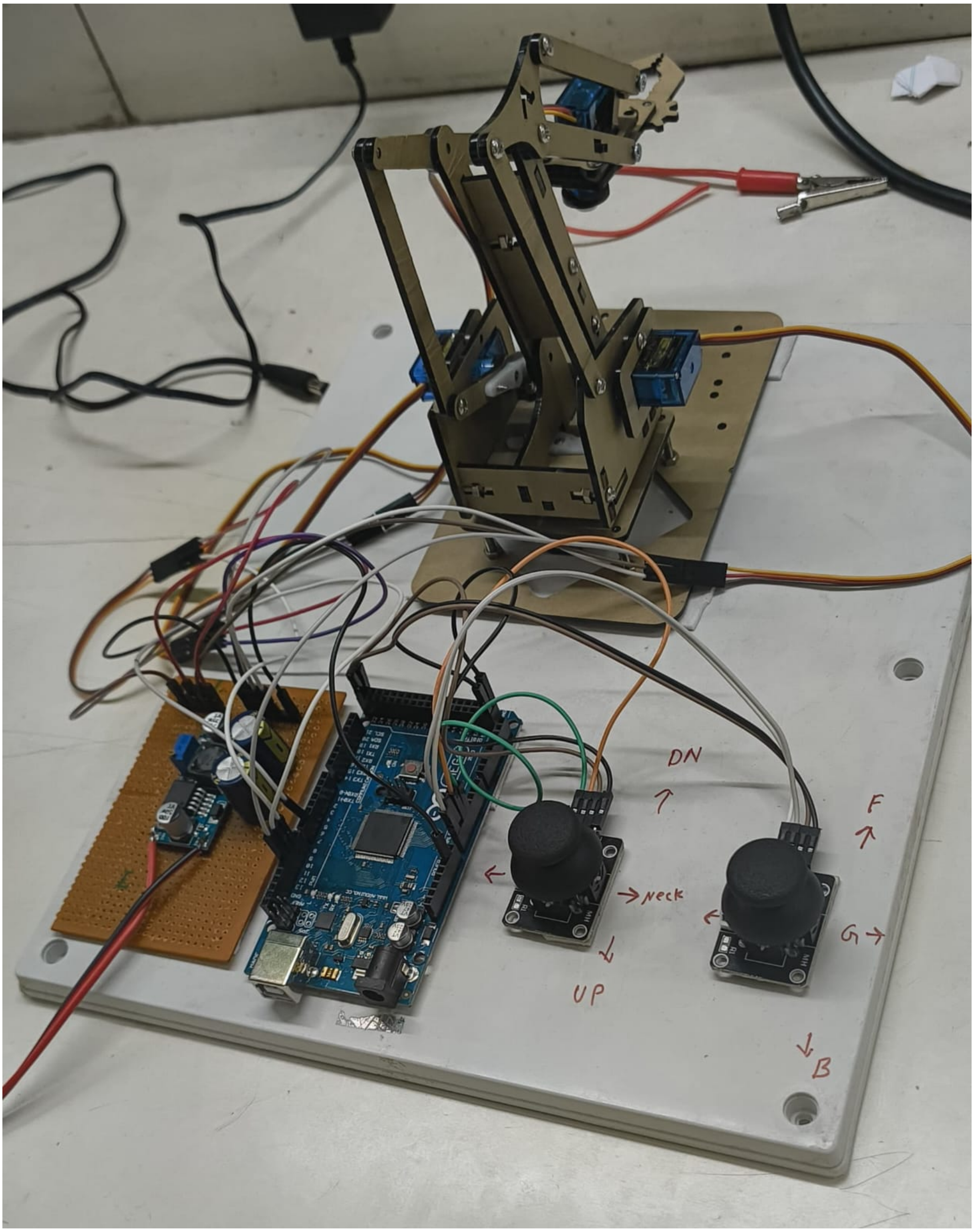


Figure 4.13: Side View Of Real Time Implemented Robotic Arm Model

4.4 Flow Chart

The flow chart represents the operational logic and sequence of actions taken by the robotic arm system, as shown in Fig. 4.14.

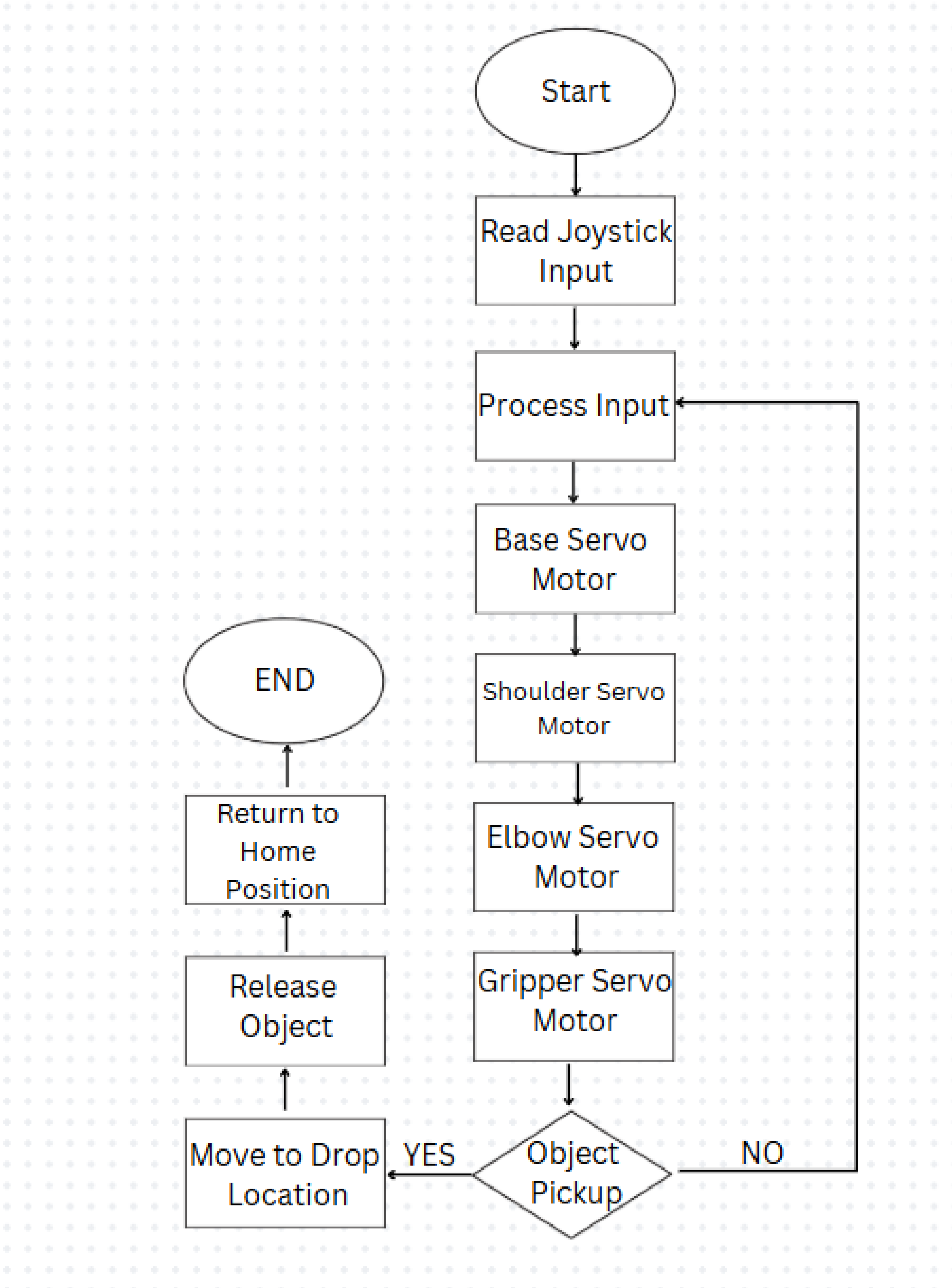


Figure 4.14: Flow Chart of Robotic Arm

Chapter 5

Bills of materials

5.1 Purchase List

The table below lists the materials required for building the robotic arm, including their unit prices and total costs, as shown in Table 5.1.

Sr. No.	Name	Unit	Total (in ₹)
1	Robotic Arm Kit	1	₹1000
2	Laser Cutting	1	₹2500
3	Arduino Mega Board	1	₹1400
4	Current Booster	1	₹100
5	USB - USB cable	1	₹110
6	12v Adapter	1	₹150
7	Servo Motor	4	₹436
8	Dual Axis XY Joystick Module	2	₹380
9	Connecting Wires	-	₹100
		Total (in ₹)	₹6176

Table 5.1: Purchase List for Robotic Arm

Chapter 6

Project timeline

6.1 Project Implementation Schedule

The project implementation schedule outlines the structured approach to designing, developing, and testing a pick-and-place robotic arm. The project is divided into two major stages, spanning one year: **Stage 1: Research and Documentation (Aug–Dec 2024)** and **Stage 2: Implementation (Jan–Apr 2025)**. Each phase ensures systematic progress and timely completion of the project.

Stage 1: Research and Documentation (Aug–Dec 2024)

- **Literature Review**: Conduct an in-depth review of robotic arm technologies, focusing on design principles, control mechanisms, and applications in automation and education.
  - **Duration**: 29 Aug 2024 – 15 Sep 2024
  - **Progress**: 100% Complete
- **Data Collection and Preprocessing**: Identify and acquire relevant datasets or specifications for robotic arm components, such as servo motors, microcontrollers, and sensors. Apply preprocessing techniques to ensure compatibility and optimal performance.
  - **Duration**: 16 Sep 2024 – 30 Sep 2024

  - **Progress**: 100% Complete

- **Theoretical Framework Development**: Define the conceptual architecture of the robotic arm, including kinematic analysis, control algorithms, and hardware-software integration.
  - **Duration**: 1 Oct 2024 – 15 Oct 2024
  - **Progress**: 100% Complete
- **Comparative Model Analysis**: Analyze existing robotic arm designs and compare their strengths, weaknesses, and suitability for pick-and-place tasks.
  - **Duration**: 16 Oct 2024 – 25 Oct 2024
  - **Progress**: 100% Complete
- **Initial Findings Compilation**: Summarize key insights from literature and dataset analysis, highlighting trends, challenges, and potential improvements in robotic arm design.
  - **Duration**: 25 Oct 2024 – 30 Oct 2024
  - **Progress**: 100% Complete
- **Research Paper Drafting**: Begin writing the research paper, including sections on introduction, methodology, literature review, findings, and conclusion.
  - **Duration**: 1 Nov 2024 – 5 Nov 2024
  - **Progress**: 100% Complete
- **Final Report Preparation and Submission**: Refine, format, and finalize the research paper for submission to an academic journal or conference.
  - **Duration**: 5 Nov 2024 – 11 Nov 2024
  - **Progress**: 100% Complete

Stage 2: Implementation (Jan–Apr 2025)

- **Implementation Setup**: Develop the robotic arm prototype, set up the development environment (e.g., Arduino IDE), and integrate hardware components such as servo motors, sensors, and microcontrollers.

  - **Duration**: 1 Jan 2025 – 15 Jan 2025
  - **Progress**: 100% Complete

- **Requirement Analysis for Phase 2**: Define the hardware and software requirements, ensure compatibility with deployment platforms, and plan the next steps for system integration.
  - **Duration**: 16 Jan 2025 – 31 Jan 2025
  - **Progress**: 100% Complete

- **Model Training and Optimization**: Program the robotic arm for pick-and-place tasks, fine-tune control algorithms, and optimize performance to ensure accurate and efficient operation.
  - **Duration**: 1 Feb 2025 – 14 Feb 2025
  - **Progress**: 100% Complete

- **Evaluation and Fine-Tuning**: Evaluate the robotic arm using performance metrics (e.g., accuracy, speed, repeatability) and refine the system based on test results.
  - **Duration**: 16 Feb 2025 – 28 Feb 2025
  - **Progress**: 100% Complete

- **System Testing**: Perform extensive testing on the robotic arm, ensuring reliable operation and smooth interaction with objects of varying shapes and sizes.
  - **Duration**: 1 Mar 2025 – 15 Mar 2025
  - **Progress**: 100% Complete

- **Final Results Compilation**: Document key results, present performance metrics, and analyze the system's effectiveness in pick-and-place tasks.
  - **Duration**: 16 Mar 2025 – 22 Mar 2025
  - **Progress**: 100% Complete

- **Report Writing for Phase 2**: Prepare the final technical report, summarizing the implementation, results, challenges, and future improvements.

– **Duration**: 23 Mar 2025 – 31 Mar 2025

– **Progress**: 100% Complete

- **Final Presentation and Demonstration**: Develop a presentation to showcase the project, demonstrate the robotic arm's functionality, and explain its design and operation, as shown in Fig. 6.1.

  – **Duration**: 1 Apr 2025 – 6 Apr 2025

  – **Progress**: 100% Complete

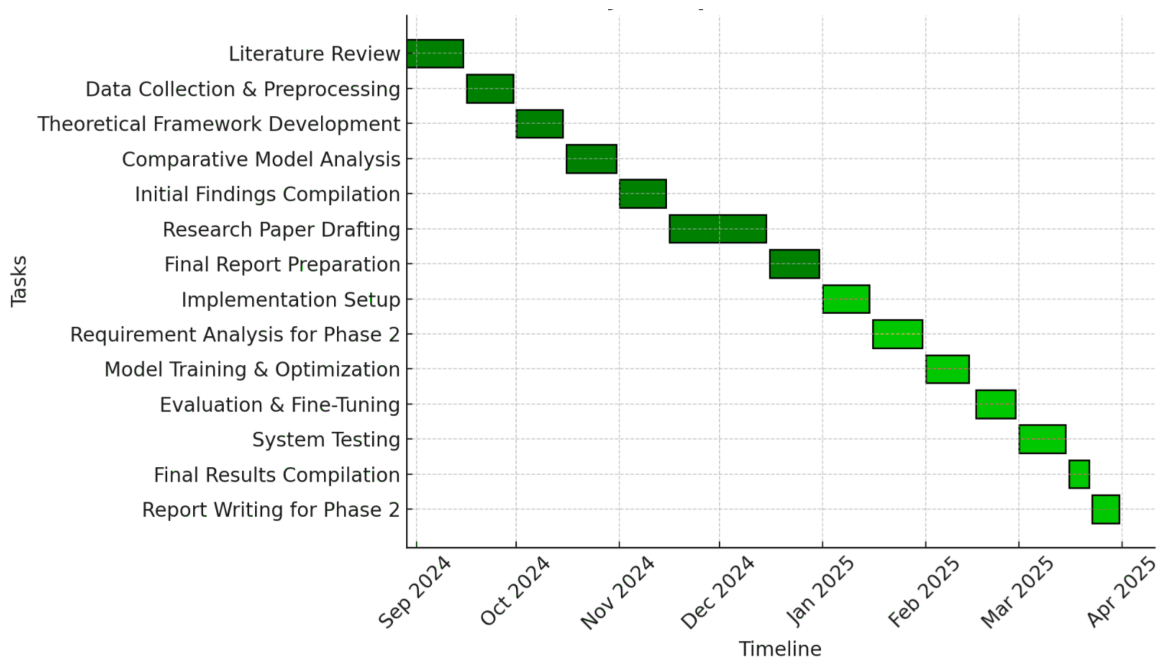


Figure 6.1: Project Implementation Schedule Gantt Chart

Chapter 7

Conclusion and Future Scope

7.1 Summary

This project focused on studying and analyzing the design, specifications, and applications of robotic arms by reviewing existing research papers and component specifications. The objective was to understand the working principles, potential use cases, and advancements in robotic arm technology.

Through an extensive literature review, we explored different robotic arm models, control mechanisms, and sensor-based automation techniques. We also studied key components such as **servo motors and Arduino Mega 2560**, understanding their roles in robotic arm development.

7.2 Key Learnings

- Understanding the mechanical structure and control system of robotic arms.
- Exploring different algorithms and techniques used for movement of object.
- Studying the integration of sensors for automation and precision control.

7.3 Conclusion

The Pick and Drop Robotic Arm project successfully demonstrates the integration of mechanical design and embedded systems to automate object handling tasks. By utilizing servo motors, joystick controls, and EEPROM memory, the system is capable of performing precise movements while retaining position data, ensuring reliability during repeated operations. This model not only highlights the potential of robotics in automation and manufacturing but also offers a scalable base for more advanced applications such as obstacle detection, AI-based path planning, or wireless control. The project deepened our understanding of servo control, memory management, and real-time system response, laying a strong foundation for further exploration in robotics and automation.

7.4 Future Scope

Building on our research and analysis, the next phase of our project involves the **practical implementation** of a **robotic arm system controlled using joysticks and servo motors**, as depicted in our circuit design. The following key areas will be focused on in future development:

7.4.1 Prototype Development

- Constructing the robotic arm using **SG90 servo motors** for movement.
- Implementing a **structural frame** to hold the servos and provide stability.

7.4.2 Joystick-Based Control System

- Using **two joystick modules** to control the robotic arm's movement.
- Mapping joystick inputs to control the **four servo motors (A, B, C, D)** for precise motion.

7.4.3 Arduino-Based Implementation

- Programming the **Arduino Uno** to process joystick inputs and send corresponding signals to servos.
- Fine-tuning the servo angles for smoother and more accurate object manipulation.

7.4.4 Object Pickup and Drop Mechanism

- Implementing an efficient gripping mechanism using servos.
- Automating object detection and pick-and-place tasks as shown in the flowchart.

7.4.5 Testing and Optimization

- Calibrating the servo movements for **better precision and stability**.
- Enhancing response time and minimizing errors in object handling.

Once successfully implemented, this model can serve as a **foundation for advanced robotic applications**, such as **automated sorting systems, assistive robotics, and industrial automation**.

References

[1] K. Kruthika, B. K. Kumar, and S. Lakshminarayanan, “Design and development of a robotic arm,” in *2016 International Conference on Circuits, Controls, Communications and Computing (I4C)*. IEEE, 2016, pp. 1–4.

[2] S. Surati, S. Hedaoo, T. Rotti, V. Ahuja, and N. Patel, “Pick and place robotic arm: a review paper,” *Int. Res. J. Eng. Technol*, vol. 8, no. 2, pp. 2121–2129, 2021.

[3] Y. Wei and D. Jia, “Research on robotic arm movement grasping system based on myo,” in *Journal of Physics: Conference Series*, vol. 1754, no. 1. IOP Publishing, 2021, p. 012173.

[4] X. Liu, S. Ren, G. Wang, L. Ma, and Y. Sun, “Design and research of an automatic grasping system for a robot arm based on visual image capture technology,” *Frontiers in Mechanical Engineering*, vol. 10, p. 1364394, 2024.

[5] F. Covaciu, P. Bec, and D.-L. Băldean, “Developing and researching a robotic arm for public service and industry to highlight and mitigate its inherent technical vulnerabilities,” in *Proceedings*, vol. 63, no. 1. MDPI, 2020, p. 25.

[6] L. Chen, H. Sun, W. Zhao, and T. Yu, “[retracted] robotic arm control system based on ai wearable acceleration sensor,” *Mathematical Problems in Engineering*, vol. 2021, no. 1, p. 5544375, 2021.

Appendix A: Source Code

The following is the source code used for the Robotic Arm Model implementation:

Listing 1: ARDUINO SOURCE CODE

```
//Quartz Mini Robotic ARM - Testing code
//Joystick 1 and 2 is connected to A0,A1, A2 and A3
//SERVO CONNECTIONS
//Neck sero - D10 - pos is saved at EEPROM 3
//Front and Back - D11 - pos is saved at EEPROM 2
//UP and Down - D9 - pos is saved at EEPROM 1
//Gripper - D6 - pos is saved at EEPROM 0
//created by: Aswinth Raj
//created for: quartzcomponents.com
#include <Servo.h>
#include <EEPROM.h>
//All the gripper positions will be saved and read from eeprom to
    resume same positon
int gripper_pos = EEPROM.read(0);
int updown_pos = EEPROM.read(1);
int frontback_pos = EEPROM.read(2);
int neck_pos = EEPROM.read(3);
// Create a servo object - one each for 4 servos
Servo Gripper_servo;
Servo UpDown_servo;
Servo FrontBack_servo;
Servo Neck_servo;
//Function the control the servo based on joystick position
void control_servo (Servo &current_servo, int current_pos, int
    EEPROM_addr)
{
    //positon should alwasy be between 0 to 180
```

```
    if (current_pos>=180)
    current_pos=175; //jitter at maximum limit
    if (current_pos<=0)
    current_pos=10; //jitter at minimum limit
    current_servo.write(current_pos); //update servo postion
    EEPROM.write(EEPROM_addr, current_pos); //save position in EEPROM
    Serial.print(EEPROM_addr); Serial.print(" = "); Serial.println(current_pos); //for debugging on serial monitor
}
void setup() {
  Serial.begin (9600);
  Gripper_servo.attach(6);
  Gripper_servo.write(gripper_pos);
  UpDown_servo.attach(9);
  UpDown_servo.write(gripper_pos);
  FrontBack_servo.attach(11);
  FrontBack_servo.write(gripper_pos);
  Neck_servo.attach(10);
  Neck_servo.write(gripper_pos);
}
void loop() {
  /*//USE FOR DEBUGGING
  Serial.println ("Gripper, UpDown, FrontBack, Neck");
  Serial.print(gripper_pos);Serial.print(",");
  Serial.print(updown_pos);Serial.print(",");
  Serial.print(frontback_pos);Serial.print(",");
  Serial.print(neck_pos);Serial.println(".");*/
  delay(50); //predefined  delay to make the servo move slower
//A0 to control Gripper Servo
  int Joy_value_X1 = analogRead (A0);
  if (Joy_value_X1 > 700){
```

```
    gripper_pos = gripper_pos + 1;
    control_servo (Gripper_servo, gripper_pos, 0);
  }
  if (Joy_value_X1 < 300){
    gripper_pos = gripper_pos - 1;
    control_servo (Gripper_servo, gripper_pos, 0);
  }
//A1 to control UpDown Servo
    int Joy_value_Y1 = analogRead (A1);
  if (Joy_value_Y1 > 700){
    updown_pos = updown_pos + 1;
    control_servo (UpDown_servo, updown_pos, 1);
  }
  if (Joy_value_Y1 < 300){
    updown_pos = updown_pos - 1;
    control_servo (UpDown_servo, updown_pos, 1);
  }
//A2 to control FrontBack Servo
      int Joy_value_X2 = analogRead (A2);
  if (Joy_value_X2 > 700){
    frontback_pos = frontback_pos + 1;
    control_servo (FrontBack_servo, frontback_pos, 2);
  }
  if (Joy_value_X2 < 300){
    frontback_pos = frontback_pos - 1;
    control_servo (FrontBack_servo, frontback_pos, 2);
  }
//A3 to control Neck Servo
        int Joy_value_Y2 = analogRead (A3);
  if (Joy_value_Y2 > 700){
    neck_pos = neck_pos + 1;
```

```
    control_servo (Neck_servo, neck_pos, 3);
  }
  if (Joy_value_Y2 < 300){
    neck_pos = neck_pos - 1;
    control_servo (Neck_servo, neck_pos, 3);
  }
}
```

Code Explanation

`#include <Servo.h>` and `#include <EEPROM.h>`

Here we have included the Servo and EEPROM libraries. All the positions will be saved in EEPROM and will resume from the last position.

```
int gripper_pos = EEPROM.read(0);
int updown_pos = EEPROM.read(1);
int frontback_pos = EEPROM.read(2);
int neck_pos = EEPROM.read(3);
```

We have allotted the EEPROM memory addresses to store the positions of respective robotic arm joints.

```
Servo Gripper_servo;
Servo UpDown_servo;
Servo FrontBack_servo;
Servo Neck_servo;
```

Servo objects are created for each of the four servos controlling different parts of the robot.

In the `control_servo()` function, servos are controlled with the help of joysticks. The current positions of all the servo motors get stored in the EEPROM.

```
void setup()
```

In this section, we define the respective pins of all servos.

```
void loop()
```

In this section, as the joysticks are operated, the values increase or decrease by 1. Maximum and minimum values are allotted as 700 and 300 respectively. The servos operate according to the values fed by the joystick.

ANSHUMAN

Pune, Maharashtra • anshuman140204@gmail.com • +91-9322823758 • LinkedIn • GitHub

SKILLS

- **Programming Languages:** JavaScript, Java, Python, SQL
- **Technologies/Frameworks:** HTML, CSS, Bootstrap, Tailwind, JavaScript, React.js, Node.js, Express.js,
- **Database:** MySQL, MongoDB
- **Tools:** Git/GitHub, VS Code, Postman

EDUCATION

Bharati Vidyapeeth College of Engineering, Pune *2021 - 2025*
B.Tech (Electronics and Communication) **CGPA 9.1/10**

Kendriya Vidyalaya Southern Command, Pune *2020 – 2021*
Class 12th - CBSE **Percentage – 83.6%**

Kendriya Vidyalaya Southern Command, Pune *2018 – 2019*
Class 10th - CBSE **Percentage – 86.4%**

EXPERIENCE

Web Dev Intern, Octanet Services Pvt. Ltd. *Dec 2024– Mar 2025*

- Designed and developed a responsive UI for the company’s portfolio, improving user engagement by 30%.
- Enhanced front-end performance, reducing page load times by 40% and improving user interaction.
- Optimized code integration between front-end and back-end, increasing operational efficiency by 25%.
- Collaborated with a team of 5+ developers, refining design concepts and implementing over 50+ user feedback improvements into the final product.

PROJECTS

Memories App

- Created a full-stack web application using the MERN stack, enabling users to manage their memories with ease.
- Engaged users by implementing commenting functionality, promoting interaction and community building.
- Implemented CRUD functionalities, allowing users to add, edit, and delete memories with full control.

E-commerce Platform

- Developed & designed a full-stack e-commerce platform using React.js to create a seamless shopping experience.
- Applied payment gateway integration with Razorpay for secure and efficient payment processing.
- Managed product inventory and order tracking, developed features for product management and order processing.
- Ensured secure user sessions with JWT for authentication, enabling registration, login, and profile management.

PLACEMENTS

- TATA Consultancy Services (TCS)
- Capgemini

ACHIEVEMENTS

- Third Rank in **National Level Intercollegiate General Knowledge Test 2024**
- 10th Rank in **KIMO's-Edge' 23 Tech Competition**
- Solved 350+ DSA problems on multiple platforms
- Runner-up of Inter College Cricket Championship

CERTIFICATIONS

- Alpha (DSA with Java) (Apna College)
- Certificate Course in AR VR Game Development (CDAC)
- A Brief Introduction of Micro Sensors (NPTEL)
- The Complete Full-Stack Web Development Bootcamp (Udemy)

Utkarsh Kumar Maurya

Pune, Maharashtra
+91-9359600456 mauryautkarshk@gmail.com Linkedin Github

EDUCATION

Bharati Vidyapeeth College of Engineering, Pune — **Dec 2021 – July 2025**
B.Tech -Electronics and Communication- ***CGPA - 8.6*** — *Pune, Maharashtra*

Kendriya Vidyalaya Southern Command, Pune — **2021**
XII - CBSE - ***Percentage - 84%*** — *Pune, Maharashtra*

Kendriya Vidyalaya Southern Command, Pune — **2019**
X - CBSE - ***Percentage - 76.2%*** — *Pune, Maharashtra*

TECHNICAL SKILLS

Programming Languages: Java, HTML, CSS, JavaScript, SQL
Developer Tools: VS Code, Postman
Technologies/Frameworks: Angular, Node.js, Bootstrap
Concepts: Data Structures and Algorithms, Object-Oriented Programming
Soft Skills: Problem Solving, Communication, Teamwork, Critical Thinking

EXPERIENCE

Techfino Capital Pvt. Ltd. — **July-August 2024**
Software Engineer Intern — *Bengaluru, Karnataka*

- Developed dynamic and responsive frontend using Angular, improving user experience.
- Implemented backend CRUD operations with Node.js and MySQL for seamless data management.
- Designed secure authentication and authorization mechanisms for login functionalities.
- Conducted API integration and debugging using Postman, ensuring smooth frontend-backend communication.

PROJECTS

Loan Management System | Angular, Node.js, MySQL

- Developed a full-stack web application to manage loan applications with role-based access control.
- Created an intuitive dashboard for customers to apply for loans and track application status.
- Built an admin panel for reviewing, verifying, and approving/rejecting loan applications.

Expense and Income Tracker | Angular

- Developed a full-stack web app for tracking expenses and income, featuring real-time updates, budget management, and data visualization using Chart.js.

Weather Application | HTML, CSS, JavaScript

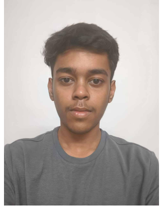
- Designed a real-time weather web app that fetches weather data using external APIs.
- Integrated geolocation services to auto-fetch the user's location for weather updates.

CERTIFICATIONS & EXTRACURRICULAR

- Awarded Elite Badge for Introduction to Internet of Things from NPTEL.
- Completed JAVA and Data Structures & Algorithms(DSA) Course from Apna College.
- Winner of Inter-School Badminton Championship.

Mayank Mankar

To gain knowledge in fields of teamwork,software designing,project management and event management Skilled in HTML, CSS, and JavaScript for front-end development, and adept in integrating secure database systems for scalable backend solutions.



Contact

MayankMankar1

Mayankmankar9876@gmail.com

7822062126

PROJECTS

News Aggregator with API

- Created a web page that fetches and displays news which is Centralized News Collection.
- Real-Time Updates: Integrated real-time updates with push notifications for breaking news, bookmarking, and sharing features. This project collects news and showcase on the platform with the help of API.

Secure Login System With Database Integration | Expresjs,Mongodb

- Integrated MongoDB/MySQL database for efficient form data storage and retrieval.
- Built user authentication with Node.js, Express js,for secure login and account creation.
- Secured authorization checks to restrict access to authenticated users, enhancing data protection.

HANGMAN GAME | HTML, CSS, JavaScript

- A game where players have to guess letters to uncover a hidden word
- The game displays a series of blank spaces representing the letters of the hidden word, and players try to guess each letter one at a time
- Used HTML, CSS, and JavaScript to create a responsive and engaging game experience.

Education

Bachelor In Technology(BTech)
Bharati Vidyapeeth College of Engineering
(Electronics and communication):: CGPA- 8.0
(2021 - 2025)

DR. ULHAS PATIL ENG MED SCHOOL RAVER JALGAON (2021)
HSC (12th): CGPA -7

KENDRIYA VIDYALAYA(O.F)BHUSAWAL(SSC)(2019)
SSC (10th): CGPA--7

Skills

Languages

- C++
- JS(JavaScript)
- HTML5
- CSS

Database

- MonogoDB
- MYsql

Devloper Tools

- GitHub,Postman, Visual Studio code

Frame work

- Express js
- Jquery

Certification/Short Courses

- A brief introduction of -Micro Sensors (NPTEL)
- The Complete Web Development(Udemy)
- C++ Programming (Udemy)

EXTRACURRICULAR

- Developed a strong interest in badminton, played at the state level in multiple tournaments, and won district championships