PROJECT DOCUMENTATION

Title–***3D Printed Robotic Arm with 5 Degrees of Freedom***

Abstract-This project presents the design and development of a 3D-printed robotic arm with five degrees of freedom (5-DOF), controlled using an Arduino microcontroller and MG995 servo motors. The robotic arm is designed to replicate human arm movements with precision, flexibility, and affordability, making it suitable for automation, education, and research purposes. All structural components are fabricated using 3D printing technology, which ensures lightweight construction, customizable design, and cost efficiency. The MG995 servo motors provide sufficient torque and stability for each joint movement, including base rotation, shoulder, elbow, wrist, and gripper actuation. The Arduino acts as the central control unit, interfacing with the servos through PWM signals and allowing motion control through programmed sequences or manual input via potentiometers or serial commands. This system demonstrates the practical integration of mechanical design, embedded systems, and control engineering to build a functional robotic manipulator. The project aims to serve as a foundational platform for future applications such as object sorting, pick-and-place operations, and AI-based gesture or vision control.

Introduction-Automation and robotics are rapidly transforming modern industries, research, and daily life by improving precision, efficiency, and safety. Among the various robotic mechanisms, robotic arms play a crucial role in performing repetitive or complex tasks such as assembly, welding, painting, and material handling. However, most commercial robotic arms are expensive and inaccessible to students or small-scale developers. To bridge this gap, this project focuses on designing and building a cost-effective 3D-printed robotic arm that mimics the motion of a human arm using affordable electronic and mechanical components.

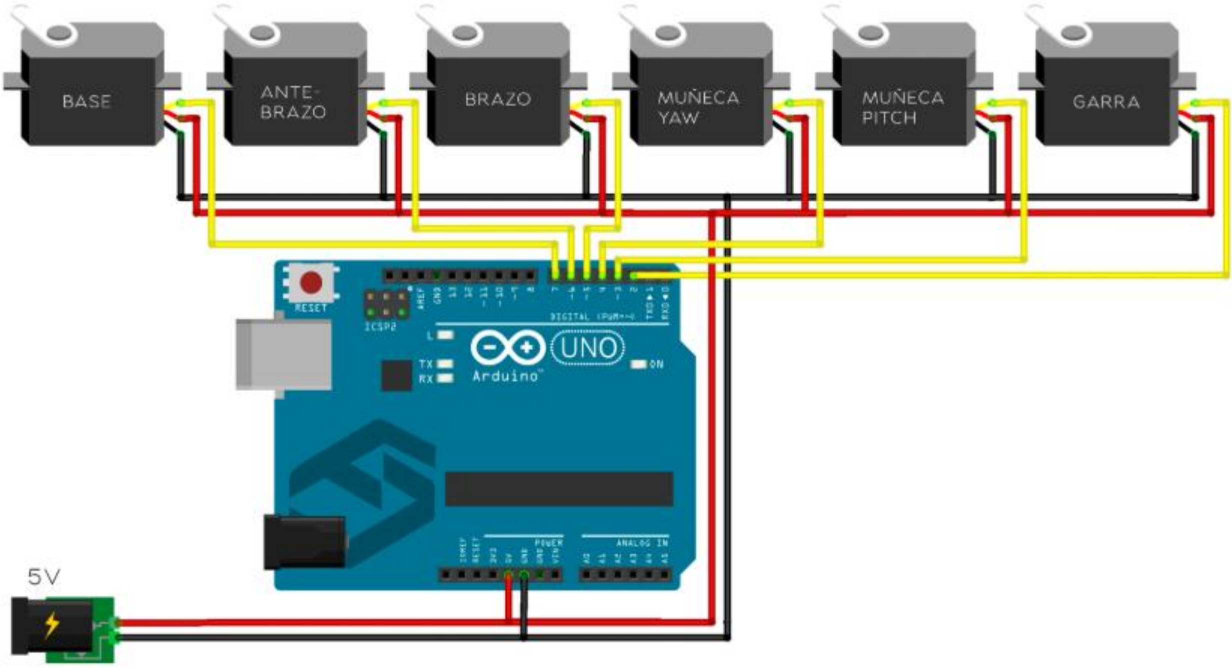
The developed robotic arm offers five degrees of freedom (5-DOF) — base rotation, shoulder, elbow, wrist, and gripper — allowing it to perform a wide range of motion and manipulation tasks. The structure of the arm is 3D printed, making it lightweight, customizable, and easy to reproduce. The MG995 servo motors are used at each joint for their high torque and precise angular control, while an Arduino microcontroller serves as the brain of the system, generating the required control signals to drive the servos.

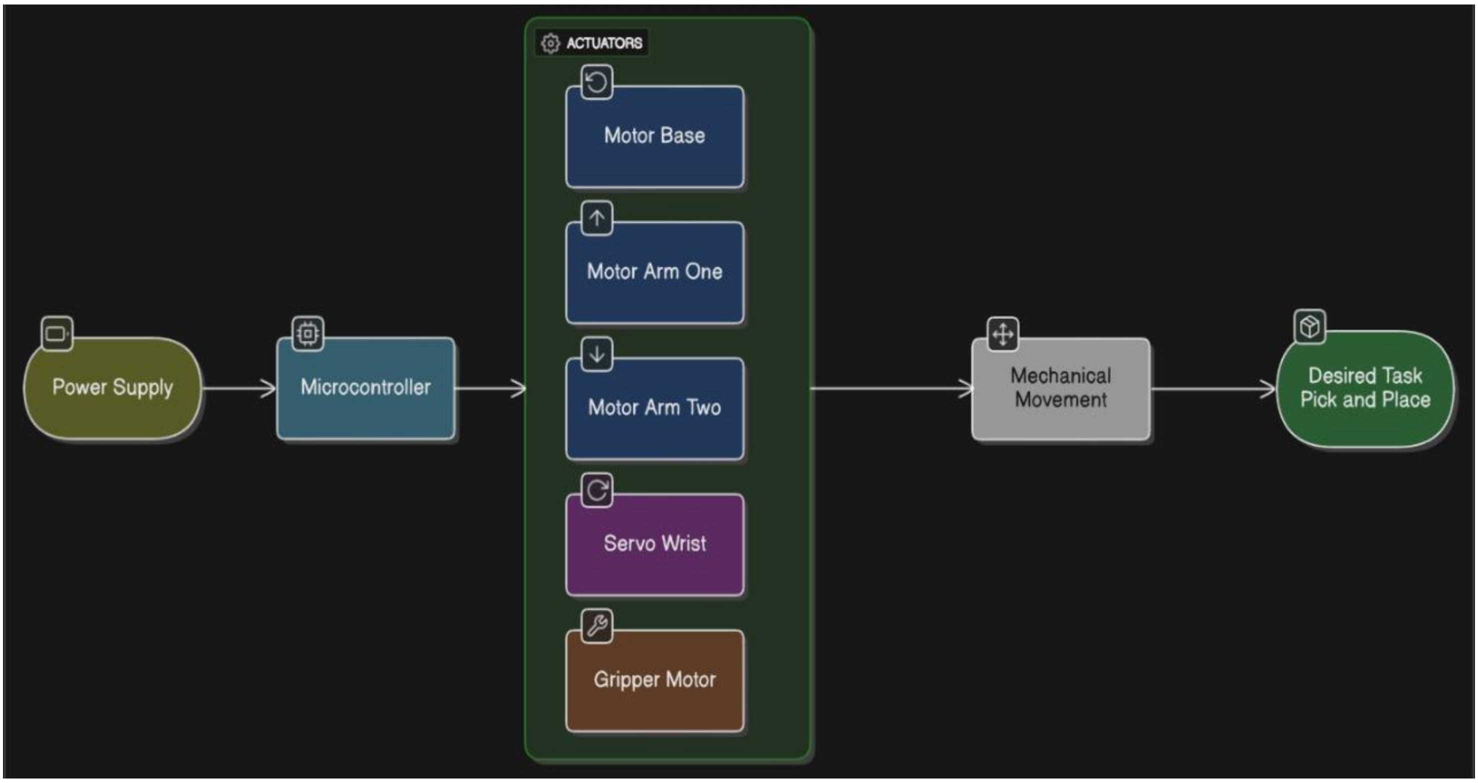
The project integrates mechanical design, electronics, and control programming to demonstrate how open-source hardware and additive manufacturing can be combined to create practical robotic systems. In addition to its educational value, the robotic arm can be expanded with features such as sensor feedback, wireless control, or computer vision, paving the way for advanced automation and research applications.

Materials used -

- 3D Printed Parts
- Arduino Uno
- MG995 servo motors (5 units)
- 5V–6V DC power source (2A–5A)
- Jumper wires
- Arduino IDE
- Breadboard

Design,Schematic and block diagrams -





Implementation-The implementation of the 3D Printed Robotic Arm involves several key stages — mechanical design, electronic circuit setup, programming, and system integration. Each stage was carefully executed to ensure smooth operation, precise movement, and reliable control of all five degrees of freedom.

---

1. Mechanical Design and Assembly

The robotic arm was designed using Fusion 360 to replicate human arm motion with joints representing the base, shoulder, elbow, wrist, and gripper. Each component was 3D printed using PLA filament for lightweight and structural rigidity. The arm parts were assembled using screws, nuts, and bearings to ensure stable rotation and minimal friction. The MG995 servo motors were mounted at each joint to provide controlled angular movement.

The base rotation allows the arm to rotate horizontally, while the shoulder and elbow joints control the vertical and forward motion. The wrist enables angular adjustment of the gripper, and the gripper servo allows objects to be picked and released.

---

2. Electronic Circuit Design

The control circuit was built around an Arduino Uno microcontroller, which generates PWM (Pulse Width Modulation) signals to control the position of each servo motor.

The servos were powered through an external 5V–6V DC power supply to provide sufficient current and avoid overloading the Arduino. The signal wires from all servos were connected to the Arduino’s digital PWM pins. A common ground was established between the Arduino and the power source to ensure synchronized control.

In some versions, potentiometers or joysticks were added as input devices, allowing manual control of servo positions by varying voltage levels.

---

3. Programming and Control Logic

The control program was written in Arduino IDE using the Servo.h library. Each servo was initialized and assigned to a digital PWM pin. The program defines angle limits for each servo and sequences their movements to simulate human-like arm motion.

The code allows two modes of operation:

- Manual Mode: Servo angles are adjusted using potentiometers or joystick inputs.
- Automatic Mode: The arm performs pre-programmed tasks such as pick-and-place movements.

Delays and angle transitions were fine-tuned to achieve smooth and coordinated motion of all joints. The system can also be expanded to integrate Bluetooth, Wi-Fi, or vision-based control for advanced applications.

---

4. Testing and Calibration

After assembly and coding, each joint was tested individually to verify correct range of motion and servo direction. Calibration was performed by adjusting servo limits and ensuring that all parts moved without mechanical interference.

Test runs were conducted for tasks such as picking and placing lightweight objects, confirming accurate control and repeatable motion. The arm demonstrated stable operation, minimal jitter, and satisfactory torque output for small object manipulation.

---

5. Final Integration

All mechanical, electrical, and software systems were integrated into a single functional prototype. Cable management and component mounting were finalized to enhance the arm's stability and appearance. The completed robotic arm successfully achieved coordinated motion across five degrees of freedom, demonstrating effective control through Arduino-based programming and low-cost 3D printed construction.

Code –

```
#include

Servo servo_0; // Declaration of object to control the first servo
Servo servo_1; // Declaration of object to control the second servo
Servo servo_2; // Declaration of object to control the third servo
Servo servo_3; // Declaration of object to control the fourth servo
Servo servo_4; // Declaration of object to control the fifth servo
Servo servo_5; // Declaration of object to control the sixth servo
Servo servo_6; // Declaration of object to control the seventh servo (not used in this project)

void setup() {
 Serial.begin(9600); // Initialize serial communication
 servo_0.attach(2); // Associate servo_0 to pin 2
 servo_1.attach(3); // Associate servo_1 to pin 3
 servo_2.attach(4); // Associate servo_2 to pin 4
 servo_3.attach(5); // Associate servo_3 to pin 5
 servo_4.attach(6); // Associate servo_4 to pin 6
 servo_5.attach(7); // Associate servo_5 to pin 7
 servo_6.attach(8); // Associate servo_6 to pin 8
}

void loop() {
 if (Serial.available() > 0) { // If there is data available to read
  String input = Serial.readStringUntil('\n'); // Read the data string until newline
  int servoIndex = input.substring(0, 1).toInt(); // Get the servo index
  int servoValue = input.substring(2).toInt(); // Get the servo value

  switch (servoIndex) {
   case 1:
    servo_0.write(servoValue);
    break;
   case 2:
    servo_1.write(servoValue);
    break;
   case 3:
    servo_2.write(servoValue);
    break;
   case 4:
    servo_3.write(servoValue);
    break;
   case 5:
    servo_4.write(servoValue);
    servo_6.write(180 - servoValue);
    break;
   case 6:
    servo_5.write(servoValue);
    break;
   default:
    // Invalid servo index
    break;
  }
 }
}
```

Results and Analysis–

Upon completion of the assembly and programming of the 5-DOF robotic arm, several tests were conducted to evaluate its performance:

- Range of Motion: Each joint demonstrated a full range of movement, with the MG995 servos providing sufficient torque to achieve the desired angles without stalling.

- Precision: The arm successfully performed tasks such as picking and placing lightweight objects, indicating accurate control over the servos.

- Repeatability: Pre-programmed sequences were executed multiple times with consistent results, showcasing the arm's ability to perform repetitive tasks reliably.

- Control Interface: Manual control via potentiometers allowed for real-time adjustments, while automated sequences were executed through Arduino programming, demonstrating versatility in control methods.

Component datasheet–

- **SG-90 SERVO MOTOR –** *https://www.friendlywire.com/projects/ne555-servo-safe/SG90-datasheet.pdf*
- **MG995 SERVO MOTOR–** *https://www.electronicoscaldas.com/datasheet/MG995_Tower-Pro.pdf?srsltid=AfmBOorrl9rxFe6Sf9ueJeN5aQju1YoREgRAT6IiBtEig8zC6EQRpvh4*
- **ARUDINO UNO–** *https://docs.arduino.cc/resources/datasheets/A000066-datasheet.pdf*

Discussion and Limitations–

While the project achieved its primary objectives, several limitations were identified:

- Torque Limitations: The MG995 servos, though adequate for lightweight tasks, may struggle with heavier payloads, limiting the arm's application in industrial settings.

- Power Supply Constraints: The servos require a stable power supply; fluctuations can lead to erratic movements or servo damage, as noted in user experiences with similar setups Arduino Forum.

- Mechanical Wear: Repeated use can lead to wear and tear on the 3D printed components, potentially affecting the arm's longevity and precision.

- Limited Feedback Mechanisms: The absence of advanced feedback systems (e.g., encoders) restricts the arm's ability to perform complex tasks requiring high precision.

- Control Interface Limitations: While manual control is feasible, integrating advanced control systems (e.g., inverse kinematics, machine learning) could enhance the arm's capabilities.

Conclusion–

The development of a 3D printed robotic arm with 5 degrees of freedom using MG995 servos and Arduino has proven successful in demonstrating the feasibility of low-cost automation solutions. The arm's ability to perform basic tasks such as object manipulation and sequence execution highlights its potential as an educational tool and a foundation for more advanced robotic systems. The integration of 3D printing technology allows for customizable designs, making it accessible for prototyping and experimentation.

Future Scope–

Future enhancements to the robotic arm could focus on:

- Increased Payload Capacity: Upgrading to higher-torque servos or incorporating additional support structures to handle heavier objects.

- Advanced Control Systems: Implementing inverse kinematics algorithms and incorporating sensors for feedback to improve precision and adaptability.

- Wireless Control: Integrating Bluetooth or Wi-Fi modules for remote operation, expanding the arm's usability in various environments.

- AI Integration: Employing machine learning techniques to enable the arm to learn and adapt to new tasks autonomously.

- Durability Improvements: Using more robust materials for 3D printing to enhance the arm's longevity and performance under continuous use.

References–

1. https://www.youtube.com/watch?v=ZEir102PxJ8

2. https://fabricreator.com/en/blogs/tutorials/robotic-arm-with-arduino-save-play-export-import-positions