Project – 4 DOF Robotic Gripper Arm

Team 6

Dheeraj Chilukuri

Kailash Nathan

Jonathan Reggie Ebenezer

Pranav Sai Muniganti

Srinivas Palaniraj

MAE 598
Spring 2025
Arizona State University

Table of Contents:

1. Introduction

This project aims to develop a 4 Degrees of Freedom (DOF) robotic arm with an end-effector gripper, built using low-cost, modular components such as an Arduino Uno, PCA9685 servo driver, and 3D-printed parts. The arm provides motion along four axes: base rotation, shoulder pitch, elbow pitch, and wrist orientation, with an additional gripper for object manipulation.

The system is designed to balance mechanical simplicity and functional versatility, serving as a suitable platform for educational and lightweight automation tasks.

Figure 1: 4-DOF robotic arm

2. Problem Statement

There is a rising demand in education, research, and prototyping sectors for robotic systems that are both cost-effective and customizable. While commercial robotic arms deliver high precision, they are often expensive, bulky, and closed-source, limiting their adaptability for teaching and innovation.

This project seeks to address this gap by proposing a robotic arm solution that is:
- Lightweight and 3D printable, making it ideal for desktop use
- Driven by open-source microcontrollers (Arduino) and I2C-based servo drivers
- Expandable to higher degrees of freedom
- Programmable for educational experimentation and basic automation

By creating a modular and open-source robotic arm, this project contributes to democratizing access to robotics.

3. Applications

The 4 DOF robotic arm has various practical and educational applications:

1. Pick-and-Place Operations: It can move objects between predefined points, aiding in repetitive tasks in production lines or warehouses.
2. Educational Platforms: Ideal for demonstrating robotic motion, control systems, and embedded programming to students.
3. Assembly Automation: Helps in screwing, stacking, and organizing small parts in lightweight assembly lines.
4. Delicate Material Handling: Its lightweight design allows handling of small items in electronics, food packaging, or laboratory environments.
5. Prototyping Tasks: Facilitates engineers and researchers in developing and testing automation sequences quickly.

4. Construction Overview

The robotic arm consists of modular 3D-printed segments and is driven by a total of five servo motors (MG996R and MG90S). The sub functional components/modules are described below:

4.1 Base Rotation Module:

- 3D printed structure providing stability to the entire robotic arm
- Houses the Arduino Uno controller board, base rotation servo motor and the HiLetgo PCA9685 Servo Driver
- Designed for clean wiring management and easy access to electronic components
- Provides a strong foundation for smooth and provides 360-degree horizontal rotation for the entire arm structure

Control: Controlled by a dedicated potentiometer that maps user input to base rotation angle

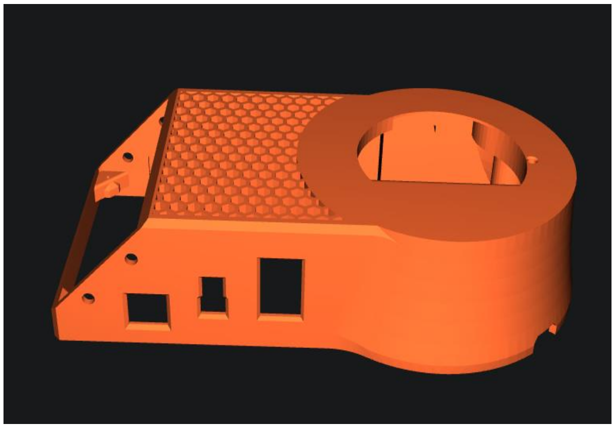

Figure 2: Base Rotation Module

4.2 Arm Segments

- **Shoulder Joint**: Features a high-torque servo motor for vertical rotation, controlling the up-and-down movement of the arm relative to the base
- **Elbow Joint**: Connects the upper arm to the forearm with 3D printed brackets that securely hold the servo and allow smooth rotation
- **Wrist Joint**: Enables orientation control of the gripper, adjusting its angle for optimal object gripping
- **Gripper Mount**: Connects the wrist to the actual gripper mechanism, supporting it for object manipulation

Control: Operated through dedicated potentiometers for precise angle control

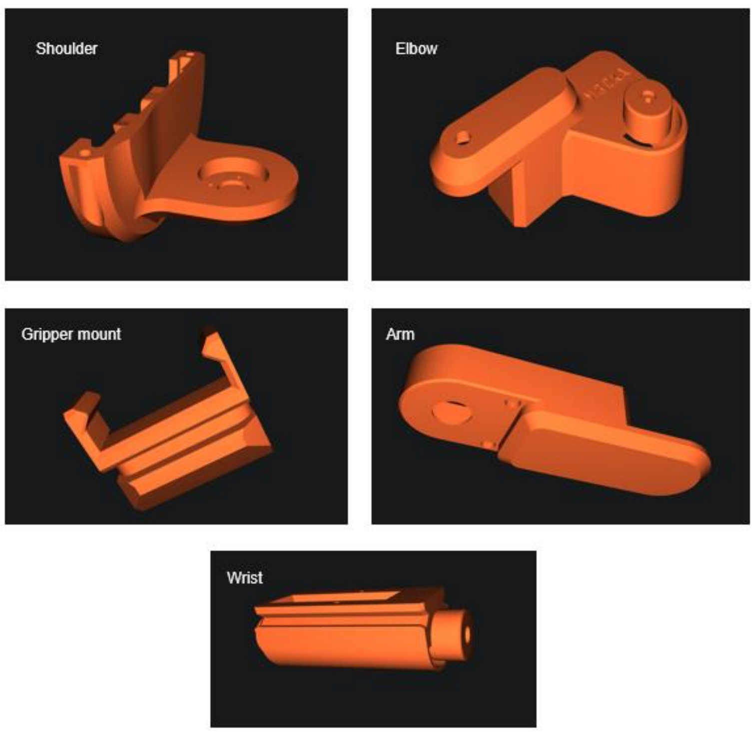

Figure 3: Arm Segments

4.3 End Effector

- Gripper mechanism for grasping and manipulating objects
- Operated by a dedicated servo motor
- Designed for lightweight material handling

Control: Operated through a push button for open/close functionality

All parts were designed to be printed using PLA plastic and assembled using nuts, bolts, and brackets for modularity.

5. Mechatronic Systems/Components Description

The project integrates various electrical and mechanical components to realize a functional robotic system:

5.1 Electronic Systems/Components

- **Arduino UNO**: Serves as the central controller of the system
  - Reads analog signals from potentiometers
  - Communicates with the servo driver via I2C protocol
  - Translates user inputs into servo commands
- **HiLetgo PCA9685 Servo Driver**:
  - 16-channel PWM (Pulse Width Modulation) module for precise multi-servo control
  - Generates accurate PWM signals for multiple servos
  - Enables smooth, synchronized control without overloading the Arduino
  - Connected to Arduino via I2C communication protocol
- **Power Supply**:
  - Provides appropriate voltage and current for the electronic components and servo motors

5.2 Control Systems/Components

- **Control Software**: Arduino programming environment used to develop control algorithms
- **Input Devices**:
  - **Potentiometers**: Manual input devices that vary resistance; Arduino reads these signals to control motor positions
  - **Push Button**: Simple digital input used to command gripper open/close actions during operation
- **Control Logic**: Converts analog inputs from potentiometers to appropriate servo positions using Arduino programming

5.3 Mechanical Systems/Components

- **Servo Motors**:
  - **MG996R**: High-torque servos used for base, shoulder, and elbow joints
  - **MG90S**: Lightweight servos used for wrist and gripper mechanisms
- **3D Printed Structural Components**:
  - Base, arm segments, and gripper components fabricated using 3D printing technology

- Designed for lightweight construction while maintaining structural integrity

- **Joints and Linkages**:
  - Mechanical connections between arm segments allowing rotational movement
  - Optimized for smooth operation and stability

Figure 4: Mechatronic Components

These components are connected on a breadboard, powered via external supply, and controlled via Arduino code using the I2C protocol.

6. Mechatronic System Integration

6.1 Hardware Integration

The system integrates mechanical components with electronic control systems through a modular design approach. The Arduino UNO serves as the central controller, connecting to the PCA9685 servo driver via I2C. The servo driver then manages the multiple servo motors distributed throughout the arm structure. User input is captured through potentiometers and buttons connected to the Arduino's input pins.

6.2 Software Integration

The control software programmed into the Arduino reads the analog values from the potentiometers, maps these values to appropriate servo positions, and sends commands to the PCA9685 driver. This driver then generates precise PWM signals to control each servo motor (base, shoulder, elbow, wrist, gripper) independently, allowing for coordinated movement of the entire robotic arm. A push button is debounced to operate the gripper on digital read. This modular code structure allows scalability and intuitive control over robotic joints without manual angle tuning.

Executed code snippet:

Compact_Robot_Arm_Code.ino

```
#include <Wire.h>

#include <Adafruit_PWMServoDriver.h>

#define MIN_PULSE_WIDTH       650
#define MAX_PULSE_WIDTH       2350
#define FREQUENCY             50

Adafruit_PWMServoDriver pwm = Adafruit_PWMServoDriver();

int potWrist = A3;
int potElbow = A2;                      //Assign Potentiometers to pins on Arduino Uno
int potShoulder = A1;
int potBase = A0;

int hand = 11;
int wrist = 12;
int elbow = 13;                         //Assign Motors to pins on Servo Driver Board
int shoulder = 14;
int base = 15;

void setup()
{
  delay(5000);                          // <-- So I have time to get controller to starting position

  pwm.begin();
  pwm.setPWMFreq(FREQUENCY);
  pwm.setPWM(11, 0, 90);                //Set Gripper to 90 degrees (Close Gripper)

  pinMode(13,INPUT_PULLUP);

  Serial.begin(9600);
}


void moveMotor(int controlIn, int motorOut)
{
  int pulse_wide, pulse_width, potVal;

  potVal = analogRead(controlIn);                                                  //Read value of Potentiometer

  pulse_wide = map(potVal, 800, 240, MIN_PULSE_WIDTH, MAX_PULSE_WIDTH);
  pulse_width = int(float(pulse_wide) / 1000000 * FREQUENCY * 4096);               //Map Potentiometer position to Motor

  pwm.setPWM(motorOut, 0, pulse_width);

}

void loop()
{
  moveMotor(potWrist, wrist);

  moveMotor(potElbow, elbow);
                                                  //Assign Motors to corresponding Potentiometers
  moveMotor(potShoulder, shoulder);

  moveMotor(potBase, base);



  int pushButton = digitalRead(13);
  if(pushButton == LOW)
  {
    pwm.setPWM(hand, 0, 180);                             //Keep Gripper closed when button is not pressed
    Serial.println("Grab");
  }
  else
  {
    pwm.setPWM(hand, 0, 90);                              //Open Gripper when button is pressed
    Serial.println("Release");
  }
}
```

6.3 Block Diagram

The system follows a hierarchical control structure:

1. User Input Devices (Potentiometers, Push Button)
2. Arduino UNO Controller (Signal Processing)
3. PCA9685 Servo Driver (PWM Signal Generation)
4. Servo Motors (Mechanical Actuation)
5. Robotic Arm Structure (Physical Movement)

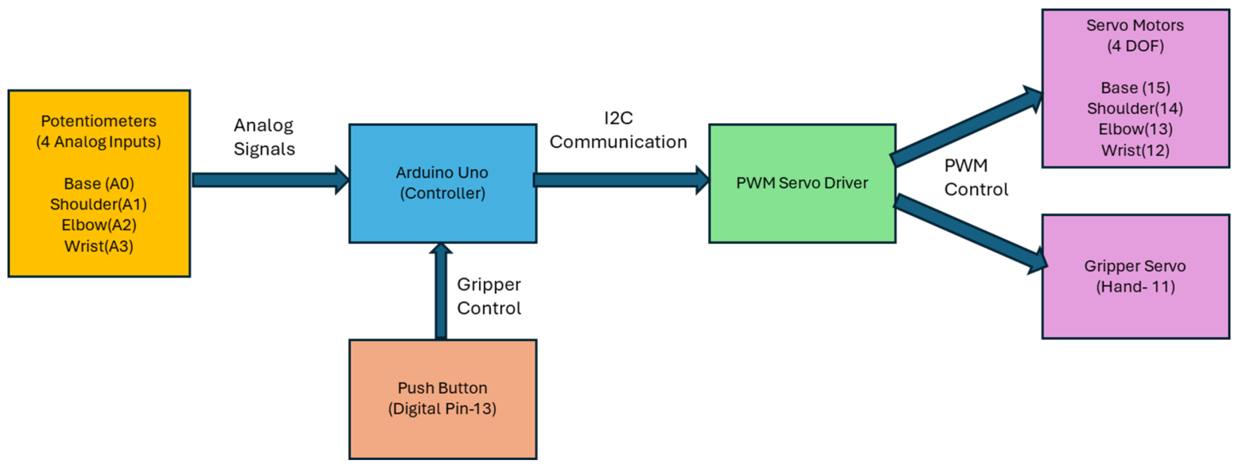

Figure 5: Block Diagram

7. Problems Encountered and Lessons Learned

7.1 Technical Challenges

- **Servo Motor Calibration and Alignment**: Achieving precise alignment between servo motors and mechanical components proved challenging, requiring careful calibration and adjustment
- **Control Interface Limitations**: The potentiometer-based control system was found to be difficult for users to operate intuitively. It lacked precision and repeatability.
- **Design Modifications**: Initial plans to implement tele-operation features like remote joystick control were abandoned due to hardware limitations.

7.2 Implementation Issues

- **Wiring Complexity**: Managing servo wiring and power supply was cumbersome in a compact structure.

- **Mechanical Tolerances**: Variations in 3D printing quality caused joint looseness and alignment issues.

7.3 Lessons Learned

- Importance of precise planning and measurement before fabrication
- Need for comprehensive testing of control interfaces for user-friendliness
- Value of modular design approaches that allow for iterative improvements

8. Conclusion and Future Improvements

8.1 Summary

The project successfully achieved its primary objective of designing and building a functional 4 DOF Robot Gripper Arm using low-cost, modular components. The integration of 3D printed mechanical parts with Arduino-based control systems demonstrated the practical application of mechatronics principles. The arm achieves smooth motion and reliable operation through PWM-based multi-servo control, making it suitable for educational purposes and lightweight material handling tasks.

Identified Limitations:
- Manual controls require improvement for real-time responsiveness.
- Mechanical backlash due to printed parts affects accuracy.

8.2 Future Improvements

- **Enhanced Control Interface**: Implementation of a more intuitive control system, possibly through a graphical user interface or joystick control
- **Wireless Operation**: Implementing wireless control via Bluetooth/ESP32 modules.
- **Vision-Based Systems**: Integration of cameras and computer vision for automated object recognition and handling
- **Increased Degrees of Freedom**: Expansion to 6 DOF for greater flexibility and range of motion
- **Precision Enhancements**: Introducing closed-loop feedback with encoders or sensors for better accuracy.
- **Programming Capabilities**: Development of a programming interface for storing and executing pre-defined movement sequences

8.3 Thought Evaluation

This project demonstrates the effective fusion of mechanical design, electronic control, and programming in a practical mechatronics application. The challenges encountered highlight the importance of careful planning, precise implementation, and iterative testing in robotics development. The modular approach adopted in this project provides a solid foundation for future enhancements and expanded functionality.

9. Teammates Contributions

Mechanical design and 3d printing	Jonathan Reggie Ebenezer, Dheeraj Chilukuri
Programming, circuit design, control system development	Kailash Nathan, Srinivas Palaniraj
Assembly	Kailash Nathan, Srinivas Palaniraj, Jonathan Reggie Ebenezer, Dheeraj Chilukuri, Pranav Sai Muniganti
Presentation Document	Kailash Nathan, Pranav Sai Muniganti
Presentation	Kailash Nathan, Srinivas Palaniraj, Jonathan Reggie Ebenezer, Dheeraj Chilukuri, Pranav Sai Muniganti
Final Report	Jonathan Reggie Ebenezer, Pranav Sai Muniganti

10. References

- Adafruit PWM Servo Driver Library: https://github.com/adafruit/Adafruit-PWM-Servo-Driver-Library
- Arduino Documentation: https://www.arduino.cc/en/Reference/HomePage
- MG996R Datasheet
- PCA9685 Specifications and Examples