STM JOURNALS

Journal of
Electronic Design Technology

https://journals.stmjournals.com/joedt

ISSN: 2229-6980 (Online)
ISSN: 2321-4228 (Print)
Volume 16, Issue 1, 2025
January–April
DOI (Journal): 10.37591/JoEDT

Research JoEDT

Development of an Arduino-based Robotic Arm for Automated Tasks

Vinit Mahajan[1], Sudhanshu Kawale[1], Sourabh Koshti[1], Varsha Kshirsagar[2,*]

Abstract

A versatile robotic arm that can perform a variety of tasks in a variety of environments is the goal of this project, which will involve designing the arm's mechanical structure, choosing and integrating appropriate motors, and programming the control system, which will be based on a microcontroller that will receive input from the potentiometer and translate it into movement commands for the arm. In the current scenario, machines and robots play an important role in the automation industry. This study presents the process through which a robotic arm is made, using Arduino and a potentiometer to control and coordinate industrial processes. Numerous operations, such as manufacturing, assembly, and inspection, as well as medical and rehabilitation settings, can make use of the robotic arm. To increase productivity and efficiency across a range of industries, this project seeks study and dependable robotic arm that can carry out intricate operations with extreme accuracy and precision. In this study, we use potentiometers as sensors to accurately control a robotic arm's movements. These sensors are placed on each joint and change their resistance as the joints rotate. By measuring these changes, we can precisely track the arm's position and how much each joint moves. This ensures smooth and accurate control, making the robotic arm more reliable for various applications.

Keywords: Robotic arm, potentiometer, Arduino uno, automation, microcontroller

INTRODUCTION

Our lives are dependent on technology, which we rely on for nearly everything. It is becoming more difficult to complete every task accurately and on schedule as the amount of work we must perform keeps increasing. This is where robots are useful. They save us time, money, and resources, work with extreme precision, and lessen the effort that humans must put forth. Robots can operate continuously without complaining or growing weary, which speeds up and improves productivity. Robotics is a mix of mechanical, electrical, and computer engineering, which helps construct machines specialized for specific jobs.

***Author for Correspondence**
Varsha Kshirsagar
E-mail: entchod.rmdssoe@sinhgad.edu

[1]Student, Department of Electronics and Telecommunication Engineering, Rasiklal M. Dhariwal Sinhgad Technical Institutes Campus, Pune, Maharashtra, India
[2]Head, Department of Electronics and Telecommunication Engineering, Rasiklal M. Dhariwal Sinhgad Technical Institutes Campus, Pune, Maharashtra, India

Received Date: January 23, 2025
Accepted Date: February 07, 2025
Published Date: February 18, 2025

Citation: Vinit Mahajan, Sudhanshu Kawale, Sourabh Koshti, Varsha Kshirsagar. Development of an Arduino-based Robotic Arm for Automated Tasks. Journal of Electronic Design Technology. 2025; 16(1): 24–33p.

A programming language called Arduino, for instance, controls the movement of robotic arms. These robotic arms are employed in industries to move and lift objects, particularly in hazardous areas that are off-limits to humans. Robots essentially handle difficult, repetitive jobs so that humans can concentrate on more creative endeavors [1].

We employ potentiometers, which function as joint sensors, to precisely regulate the robotic arm's movements. A three-part potentiometer adjusts its resistance in response to shaft rotation. These sensors are placed on each joint of the robotic arm, allowing us to precisely determine the arm's location and quantify the amount of rotation in each joint.

LITERATURE SURVEY

Development of robotic arm using Arduino UNO [2]: This article describes how to construct a simple robotic arm out of an Arduino UNO and a few basic supplies. There are four joints on the arm, and a tiny motor that can rotate from 0 to 180° controls each joint. By rotating a potentiometer, signals are sent to the Arduino to control the arm's movement. The cardboard arm's purpose is to transfer and lift light objects. Arduino software is used for programming the arm. For minor jobs, the study demonstrates how to build a basic robotic arm out of inexpensive parts.

Design of robotic arm with gripper and end effector for spot welding [3]: A gripper will be incorporated into the two-degree-of-freedom robotic arm, which is being constructed for spot welding, according to the report. A combination of threaded shafts, spur gears, and an AC motor make up the end effector. The following goals were considered when creating the robotic arm:

1. To have a rigid framework.
2. Part movement to specified angles.
3. To achieve optimal power utilization.

The author of this study paper describes how they constructed a robotic arm that has four degrees of freedom, or the ability to move in four different directions [4]. They employed clever techniques like fuzzy logic (FL) and genetic algorithms (GAs) to plan the arm's course and regulate its movement. These techniques aid in ensuring that the arm moves smoothly and without delays or friction. They employed genetic optimization to determine the ideal angles for fluid movement in the arm's joints rather than speculating or testing a wide range of angles. To improve the arm's accuracy and dependability, fuzzy logic was also utilized to regulate joint movement, friction, and the arm's rate of self-settling.

Simply put, this paper describes how the author controlled a robotic arm that replicates the actions of a human arm using sensors [5]. To detect motion, flex and gyro sensors are among the sensors positioned close to the fingers. The robotic arm is moved by means of servo motors that are controlled by an Arduino Atmega328 controller that receives signals from the fingers when they move. Arduino is programmed using the embedded C programming language. Future developments are also discussed in the paper, such as the addition of more sensors to enable more precise and fluid arm movement.

Stated differently, the authors of this work describe how they developed a robotic arm with five distinct motions [6]. The arm can raise up to 100 g of things and is controlled by a microprocessor (Cortex M3 LPC1768). The arm employs ultrasonic sensors to detect items by emitting sound waves and timing how long it takes for the sound to return after striking an object. Following object detection, the microcontroller instructs the arm's motors to pick up and transport the object to a new location.

This study describes how scientists used two pieces (links) to model, simulate, and control a robotic arm that has two degrees of freedom that move in two different directions [7]. They calculated the movement of the arm and the connections between the pieces using a technique known as Denavit Hartenberg parameters. Additionally, they employed inverse kinematics to determine the precise location of the arm's end, the component that performs the work. A Permanent Magnet DC motor powers the robotic arm, and the movements and simulations were controlled by MATLAB software.

The paper describes the characteristics and operation of a smart robotic arm [8]. Since it can assist in resolving common issues, robotic arms are being developed, and there are numerous varieties of robotic arms on the market today. The application and control of robotic arms in several fields are examined in this paper. Voice commands, applications, and remote controls are some of the ways to control the arm. The graphical user interface (GUI) that the authors employed in this project made it simpler and more convenient for users to control the arm. The software allows users to select whether they want to cut fruits or vegetables, as well as what kind of fruit or vegetable they want to cut and how. The study also makes the argument that the robotic arm might become even more intelligent using AI and machine learning, enabling it to carry out jobs without assistance from humans. Elements used in robotic arm are:

- Arduino,
- Servo motor,
- Gear box,
- Side shaft geared motor, and
- L298n motor.

The project's basic goal is to create an accurate robotic arm that uses servos to operate its joints. CATIA is the software used for design [9]. The arm is designed to carry out functions in an Automated Storage and Retrieval System (ASRS). The project's main goal is to determine which servos are best for each joint and whether they have sufficient power to move the arm. They built the arm out of metal. Selecting the appropriate servo for each joint and determining the force (torque) required at each are also included. A specially created piece of software that was created using Microsoft's programming language controls the arm.

The paper describes how to build a robotic arm using an Arduino and a potentiometer [10]. Servo motors enable the robotic arm to move in four directions, and the Arduino UNO overseas transforming analog impulses into digital ones that the servo motors can comprehend. The project's technical details, difficulties, and potential applications for the robotic arm in automation-related businesses are also included in the article. The robotic arm might also be used as an artificial arm for those who have lost a hand in an accident.

PROPOSED WORK

In this project, a robotic arm with six degrees of freedom will be created. The robotic arm is intended to be used for small-scale object lifting and movement. Servo motors power the arm's components, enabling it to move in all directions. Every motor is manually operated by means of a potentiometer. An Arduino board, which is coupled to six servo motors, powers the device and regulates the arm's movement in various directions. Furthermore, the Arduino is linked to six potentiometers to assist in manually controlling the arm's movement.

System Block Diagram

An Arduino Uno and potentiometers are used in this schematic to drive a 6-joint robotic arm (Figure 1). Every potentiometer is linked to the Arduino and modifies the servo motors' angle, which regulates the arm's joints' motion. The Arduino Uno serves as the system's brain, translating the potentiometers' analog signals into digital signals that power the servo motors. The arm can move in many directions because each joint is powered by a different servo motor, giving it six degrees of freedom (6-DOF). This straightforward design is perfect for learning and practical experimentation, and it offers a basis for building and testing a robotic arm. More intricate systems may incorporate more sensors and motors.

System Circuit Diagram

The robotic arm is primarily controlled by the Arduino Uno board. The six servo motors that move the arm's joints receive the proper signals from it after it has read the values from the six potentiometers that regulate joint movement. On the Arduino board, each servo motor is attached to a PWM output pin, and each potentiometer is connected to an analog input pin. A separate 5 V power supply is required for the servo motors, and it is connected to the Arduino as is shown in Figure 2. The Arduino board's ground pin receives all the ground connections from the potentiometers, servo motors, and power supply, guaranteeing that everything functions as it should.

Explanation of the Circuit Diagram and Block Diagram

Potentiometers: Attach each potentiometer's center pin to one of the Arduino Uno board's analog input pins (A0 to A5). The Arduino's ground and 5 V pins should be linked to the potentiometer's other two pins. Each servo motor's signal wire should be connected to one of the Arduino board's PWM output pins (9–11). The external power supply's 5 V and ground pins should be linked to the servo motors' power and ground wires.

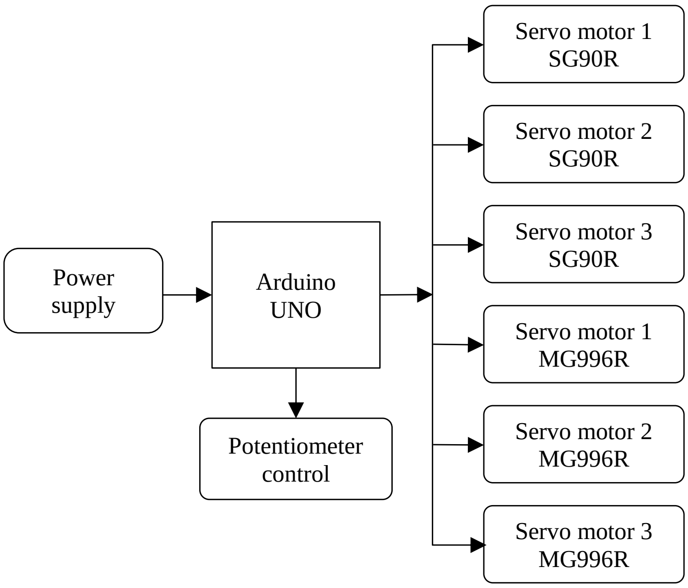

Figure 1. Block diagram of the proposed system.

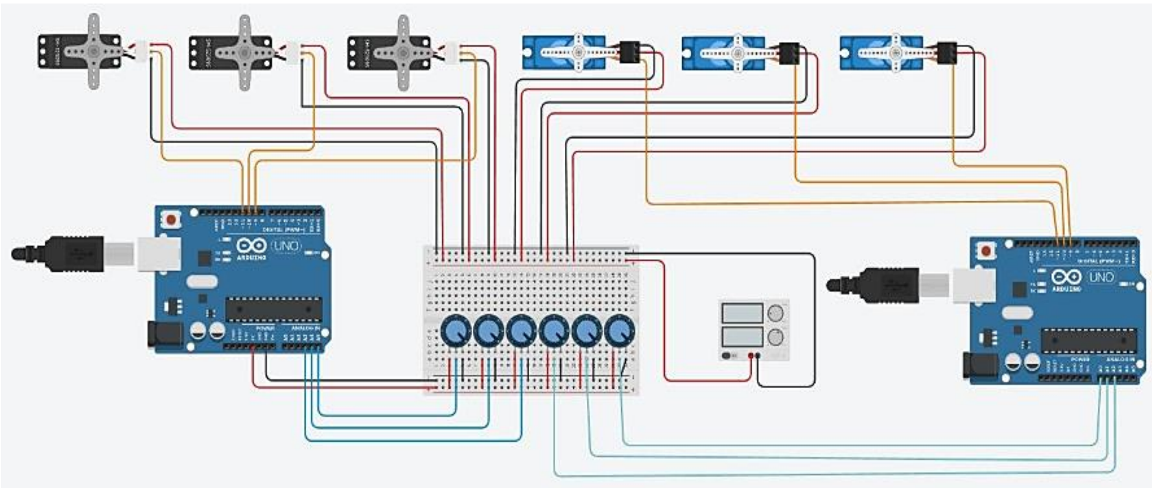

Figure 2. Circuit diagram.

Ground connection: To ensure that everything has a common reference, connect the Arduino Uno board's ground pin to the ground pin of the external power source. With this configuration, the servo motors and potentiometers can be controlled by the Arduino, and the external power source makes sure the servos have the power they require.

The Arduino IDE's analog Read() method may be used to read the potentiometer values when the connections are established, and the Servo library can be used to provide the appropriate PWM signals to the servo motors. According to the potentiometer values, the code may be set to control the robotic arm's movement, giving it six degrees of freedom.

Algorithm/Flowchart

This Figure 3 shows the step-by-step process of controlling a robotic arm using potentiometers and servo motors. To find the arm's location, it first sets up the servos and sensors before reading the potentiometer values. The motors make the necessary adjustments after converting these numbers into servo angles. To maintain the arm moving smoothly, the process repeats after a little delay that enables the servo to reach the proper position. Upon the arm reaching the desired position, the program comes to an end.

ISSN: 2229-6980 (Online), ISSN: 2321-4228 (Print)

SOFTWARE USED IN THE PROJECT

Software Used: Arduino IDE

Arduino is an open-source platform that facilitates the use of microcontrollers in the construction of digital products. For making electronic projects, it offers software (Arduino IDE and libraries) (Figure 4) as well as hardware (microcontroller boards). Since it is licensed under a creative commons license, anyone can create or alter the hardware. Open licenses also allow for the usage, modification, and sharing of the program.

You use a condensed form of C/C++ when writing code for an Arduino. Technical elements, such as creating function prototypes, are handled by the Arduino software, which then forwards the code to a C/C++ compiler so that the microcontroller can execute it. In essence, it makes it simple to control the microcontroller by enabling you to write code in a language you are comfortable with.

DESIGN AND DEVELOPMENT

Step 1: Gather Materials and Tools

The first step is to gather all the necessary materials and tools required for the project. These include:

1. Arduino Uno board.
2. 3D printed Plastic arm parts.
3. 6 Servo motors.
4. Potentiometers.
5. Jumper wires.
6. Breadboard.
7. USB cable.
8. Power supply.
9. Plywood or acrylic base sheet.
10. Screws and nuts.
11. Screwdriver and pliers.

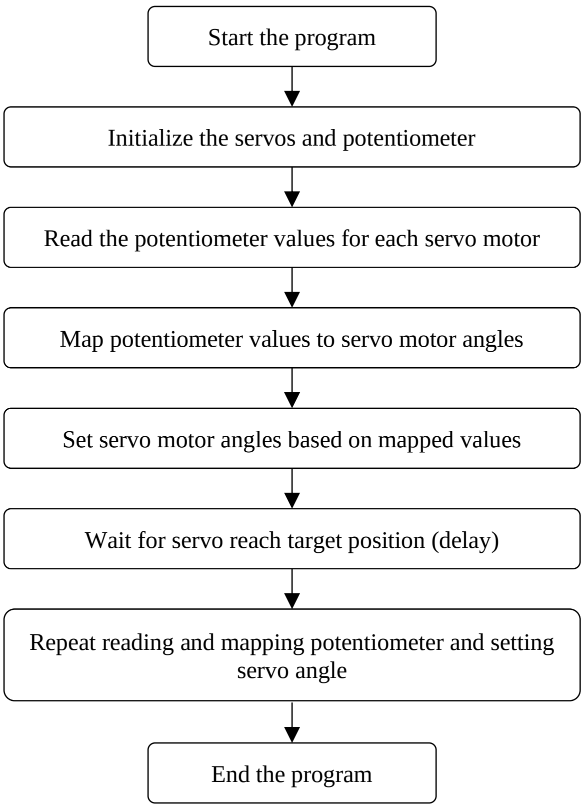

Figure 3. System's flowchart.

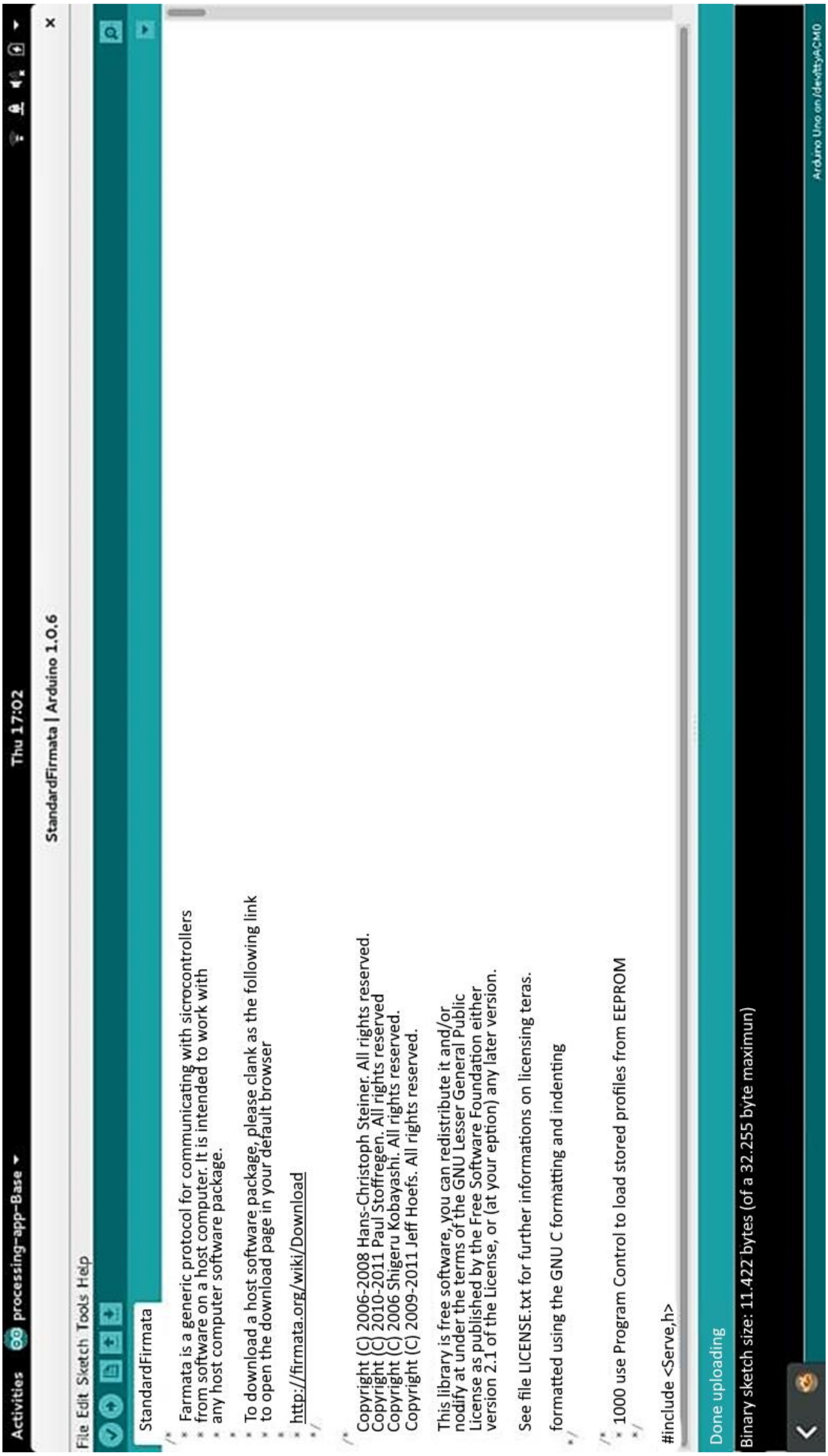

Figure 4. Arduino IDE interface.

Step 2: Design the Robotic Arm

The robotic arm must then be designed either on paper or with computer-aided design tools. Prior to designing the arm, it is crucial to comprehend its kinematics and dynamics. The design should specify the arm's measurements as well as the positions of the potentiometers, servo motors, and any other required parts.

Step 3: Build the Robotic Arm Structure

The robotic arm structure is constructed in this step utilizing an acrylic or plywood sheet as the foundation (Figure 5). Using screws and nuts, join the sheets after cutting them to the proper size. To sustain the weight of the servo motors and other parts, the construction must be strong enough.

Step 4: Mount the Servo Motors and Potentiometers

Mounting the potentiometers and servo motors on the robotic arm frame is step-4 (Figure 6). Use screws and nuts to secure the servo motors in place at the appropriate positions. In a similar manner, install the potentiometers at the arm joints.

Step 5: Connect the Servo Motors and Potentiometers

Using jumper wires, the servo motors and potentiometers are connected to the Arduino Uno board in this step. Connect the Arduino Uno board to the power source, then test each potentiometer and servo motor separately.

To Connect the Servo Motors, Follow These Steps:

1. Attach the servo motor's black wire to the Arduino Uno board's GND pin.
2. Attach the servo motor's red wire to the Arduino Uno board's 5 V pin.
3. Attach the servo motor's yellow wire to an Arduino Uno board PWM pin. Any digital PWM pin between 9 and 11 can be used.

To Connect the Potentiometers, Follow These Steps:

1. Attach the potentiometer's one end to the Arduino Uno board's 5 V pin.
2. Attach the other end of the potentiometer to the Arduino Uno board's GND pin.
3. Attach an analog input pin on the Arduino Uno board to the potentiometer's center pin. Analog input pins A0 through A5 can be used.

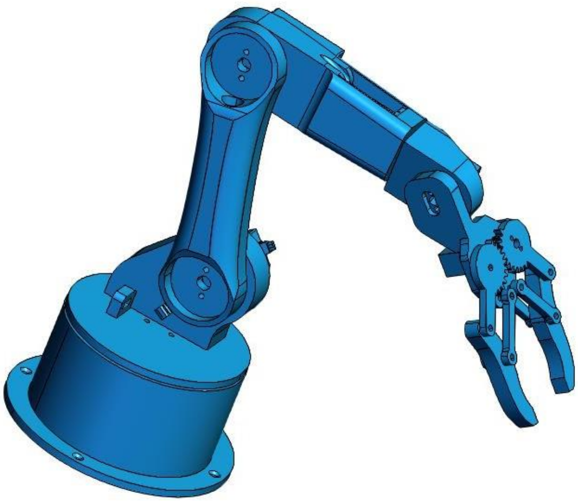

Figure 5. 3D Design of robotic arm structure.

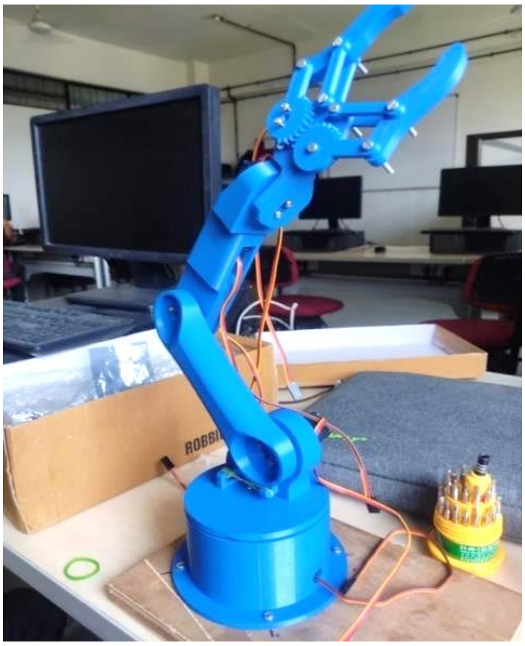

Figure 6. Robotic arm structure.

Step 6: Write the Arduino Code

Writing the Arduino code to operate the servo motors and reading the potentiometers is the sixth step. Each servo motor's control logic and the reading of the associated potentiometer value should be included in the code. To move the arm to the appropriate location, a control method needs also to be included in the code.

Step 7: Test the Robotic Arm

As shown in Figure 7, to test the robotic arm, use a computer or a mobile device to send commands to the Arduino board. The arm should glide smoothly and precisely into the appropriate places. Visual examination of the robotic arm must first be visually inspected to make sure all the parts are correctly put together and fastened. Examining the wiring, potentiometers, servos or stepper motors, and mechanical structure are all part of this. To make sure the potentiometers give the Arduino precise positional feedback, the next step is to calibrate them.

This entails calibrating the potentiometers to correspond with the arm's range of motion and setting up the Arduino code to accurately read and analyze the data. Testing the range of motion: the arm's range of motion must be verified after calibration to make sure it moves correctly in each of the six degrees of freedom. To do this, the Arduino must be programmed to move the arm through its whole range of motion and its accuracy and smoothness must be verified.

Tests of load: It is necessary to test the arm's load-bearing capability to make sure it can support the weight of the objects it is meant to operate. Weights are attached to the end of the arm, and the arm's ability to lift and move them smoothly is checked.

Rapid and accurate testing to make sure that the arm can carry out its intended functions accurately and efficiently, it is necessary to assess its speed and precision. Programming the Arduino to carry out tasks and verifying that the arm moves and arranges things as desired are part of this process.

RESULTS AND DISCUSSION

A robotic arm with six degrees of freedom (DOF) can move in various directions and perform complex tasks. To control it, an Arduino microcontroller is used because it is easy to program, affordable, and versatile. Potentiometers, which are sensors that measure the angle of each joint, are attached to the arm to track its position. The Arduino reads the data from the potentiometers and sends signals to the servo motors to move the arm accordingly.

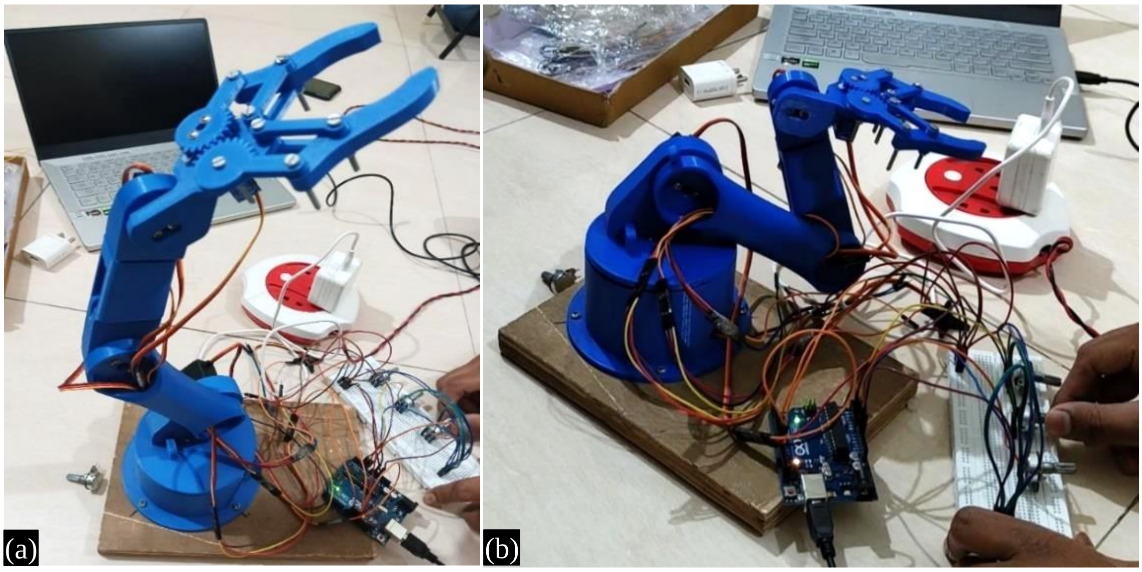

Figure 7. Testing a robotic arm.

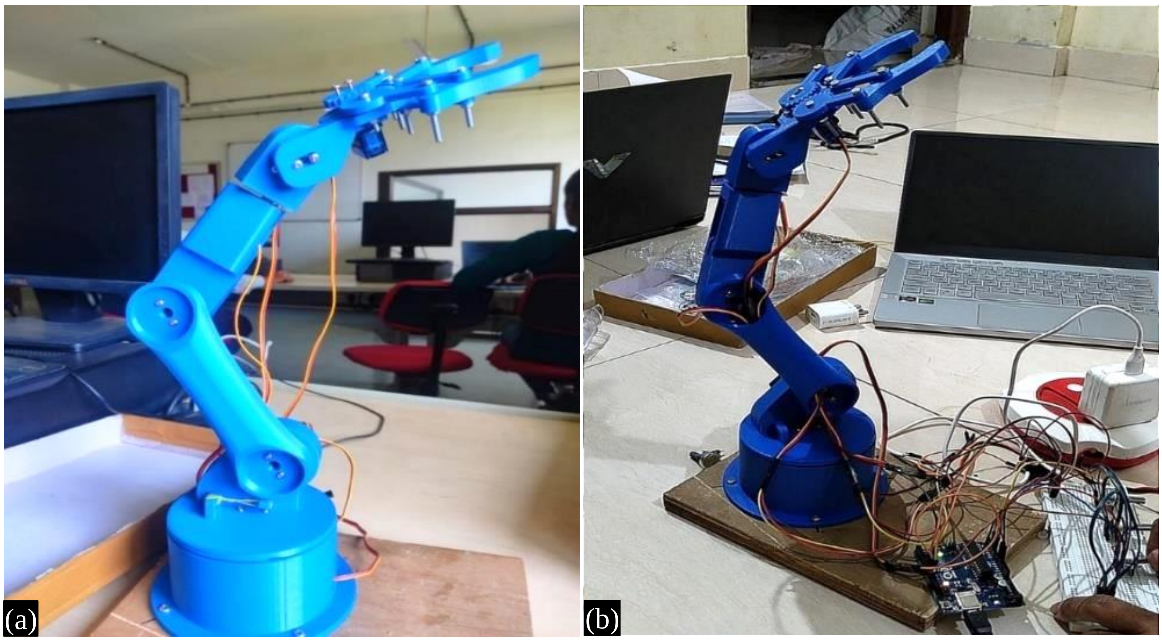

Figure 8. Assembled structure of the robotic arm.

To build the arm, you need to assemble the mechanical parts, attach the potentiometers to the joints, and connect the servo motors that control the movement. These servo motors relate to screws to 3D-printed plastic parts, allowing the arm to move smoothly. Each servo controls one part of the arm and is securely fastened to ensure stable movement. Once all servos are connected and the arm is assembled, it can be used to pick up and move objects effectively. The final output of the system is shown in Figure 8.

Output Pictures

Figure 8 shows the complete and assembled robotic arm, highlighting how its different parts fit and work together. This image helps to clearly understand the arm's design, structure, and how it functions.

CONCLUSION

It is a difficult but worthwhile effort to build a 6-DOF robotic arm using Arduino and potentiometers. A mechanical arm that can move accurately in six distinct directions must be designed and put together. An Arduino microcontroller and potentiometers are used to regulate movement. You must be proficient in programming, mechanical design, and problem-solving techniques to finish the project.

Depending on the user's needs, this robotic arm can be employed in a variety of capacities, including automation, manufacturing, and even as an assistant. For example, a four-degree-of-freedom robotic arm can be made to spin 180 or even 360°, which makes it adaptable to a variety of uses.

Future Scope

- *Integration of AI and machine learning:* Machine learning can be used to program the robotic arm to learn and get better over time. Because of this, it is more effective and able to manage challenging jobs.
- *Gesture control:* The robotic arm may be operated with hand gestures rather than buttons or a remote, which makes it easier to use and more natural.
- *IoT connectivity:* Users can remotely monitor and control the robotic arm by connecting it to the internet. Tasks that must be mechanized or completed under dangerous conditions benefit greatly from this.
- *Sensor integration:* By monitoring its environment, the robotic arm may function more precisely and safely with the addition of sensors such as force, temperature, and proximity sensors.

REFERENCES

1. Smys S, Ranganathan G. Robot assisted sensing control and manufacture in automobile industry. Journal of ISMAC. 2019 Dec; 1(03): 180–7.
2. Bhargava A, Kumar A. Arduino controlled robotic arm. In 2017 IEEE International conference of electronics, communication and aerospace technology (ICECA). 2017 Apr 20; 2: 376–380.
3. Singh P, Kumar A, Vashisth M. Design of a robotic arm with gripper & end effector for spot welding. Univers J Mech Eng. 2013 May; 1(3): 92–7.
4. Anughna N, Ranjitha V, Tanuja G. Design and Implementation of Wireless Robotic Arm Model using Flex and Gyro Sensor. Int J Recent Technol Eng. 2020; 8(5): 2978–83.
5. Jadeja Y, Pandya B. Design and development of 5-DOF robotic arm manipulators. Int J Sci Technol Res. 2019 Nov; 8(11): 2158–67.
6. Ghaleb NM, Aly AA. Modeling and control of 2-DOF robot arm. International Journal of Emerging Engineering Research and Technology (IJEERT). 2018; 6(11): 24–31.
7. Khosla D, Sharma M, Khanna SK, Khanna P, Kaur G. Smart robotic arm. J Artif Intell Res Adv. 2019; 6(2): 58–62.
8. Gao G, Ye M, Zhang M. Synchronous robust sliding mode control of a parallel robot for automobile electro-coating conveying. IEEE Access. 2019 Jun 26; 7: 85838–47.
9. Gong L, Zou B, Kan Z. Modeling and optimization for automobile mixed assembly line in industry 4.0. J Control Sci Eng. 2019; 2019(1): 3105267.
10. Li J, Liu J, Wang X, Ge W. Structure Design and Analysis of Reconfigurable Fixture Robot Based on the Auto-body Panels. In 2018 IEEE International Conference on Mechatronics and Automation (ICMA). 2018 Aug 5; 1771–1776.

