DESIGN A ROBOTIC ARM WITH 6 DIGREES OF FREEDOM

Access Online

Article DOI:
10.5281/zenodo.21412472

Dr. A. Chaitanya Krishna

Asst. Prof., ECE, Military College of Electronics and Mechanical Engineering, Secunderabad

ABSTRACT

A robotic arm is one of the most popular papers that almost every electronics enthusiast tries to build at least once in their life. Robotic arms are fascinating papers because they can be built for many practical applications, such as pick-and-place operations, component assembly, repetitive tasks, and even creating intelligent systems like the robotic arm “Dummy” seen in the Iron Man movie. These robotic arms are mainly used in environments where hazardous chemicals must be handled safely or where tasks are complex but repetitive, and in the electronics industry, they are widely used for PCB assembly because they can work faster and with higher accuracy than humans. Okay, let’s make an Arduino robotic arm that is easier to operate and simpler to build. This comprehensive robotic arm paper using Arduino demonstrates how to create a fully functional 6 DOF robotic arm capable of performing pick and place robotic arm operations with precision and reliability. Our robotic arm Arduino paper includes a complete robotic arm circuit diagram, and step-by-step instructions for building a robotic arm with servo-motor actuation.

KEYWORDS: Arduino, Robotic Arm, Degrees of Freedom

How to Cite:
Dr. A. Chaitanya Krishna (2026), Design A Robotic Arm With 6 Digrees Of Freedom, International Educational Journal of Science & Engineering (IEJSE),Vol: 9, Special Issue, May 2026 296-299

INTRODUCTION

Imagine a robotic arm built with Arduino technology, designed to mimic the movements of a human arm. This mechanical marvel can perform precise and repeatable tasks, all controlled through programming. The arm is made up of several rigid segments linked by joints, powered by servo motors that allow it to move in multiple directions. The key to its flexibility lies in its degrees of freedom (DOF)—the more DOF it has, like 4 or 6, the more versatile and capable it becomes.

These robotic arms have become indispensable in many industries. In factories, they expertly assemble cars, weld metal, organize components, and apply paint with flawless precision. Meanwhile, in the medical world, they assist surgeons during complex procedures, providing the steadiness and accuracy that human hands alone might struggle to achieve.

From manufacturing to medicine, the Arduino robotic arm stands as a powerful tool, combining technology and engineering to extend human capabilities and improve the way we work and heal.

KEY COMPONENTS OF A 6 DOF ROBOTIC ARM

Table: DOF

Component	Function
Base Joint	360-degree rotation around vertical axis
Shoulder Joint	Vertical arm movement (up/down)
Elbow Joint	Arm extension and retraction
Wrist Pitch	End effector angle adjustment
Wrist Rotation	Gripper orientation control
Gripper	Object grasping mechanism

Joints in a robotic arm function like human joints—such as elbows and

Research Paper

wrists—enabling the arm to bend, rotate, and change direction. Each joint adds to the arm's overall range of motion. For instance, a 6-axis robotic arm has six independent joints that work together to achieve complex, smooth movements in three-dimensional space.

Degrees of freedom (DOF) indicate how many independent movements a robotic arm can perform. A 1-DOF arm moves in a single direction, whereas a 3-DOF arm can move up and down, side to side, and rotate. Increasing the DOF makes the robotic arm more flexible, capable, and human-like in its motion. Most beginner robotic arms use servo motors for precise control. Servos rotate to specific angles based on control signals, making them ideal for accurate positioning in Arduino robotic arm projects. Typically, hobby servos rotate within a limited range (usually 0 to 180 degrees) and provide good positional control. The torque rating of a servo determines how much weight the arm can lift. Understanding servo characteristics is vital since weak or overloaded servos can limit the arm's performance.

Every servo or joint has physical rotation limits. While most hobby servos rotate up to 180 degrees, some specialized servos allow full 360-degree rotation. These limits define the arm's working area and influence mechanical design decisions in DIY Arduino robotic arm projects.

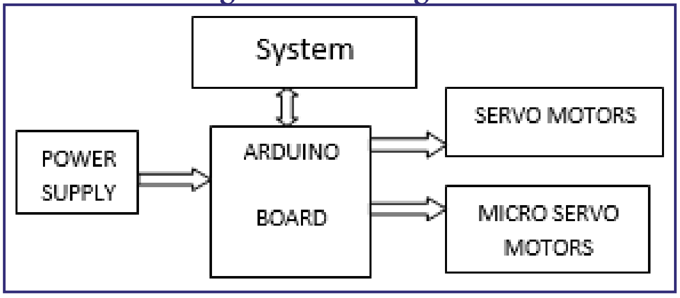
Robotic arms with multiple servos require a stable and sufficient power supply. USB power alone is often inadequate; a separate power source for servos is usually necessary. Both voltage and current ratings are important because insufficient current can cause servo shaking, overheating, or damage.

The controller acts as the robotic arm's brain, influencing its capabilities. Arduino boards are suitable for simple pick-and-place tasks; Raspberry Pi supports advanced processing and vision systems; and ESP32 enables wireless control and IoT integration.

DESIGN

The Arduino Robotic Arm Circuit Diagram below illustrates the connections between all components in the setup. In this configuration, the Arduino Uno microcontroller serves as the brain of the robotic arm, hosting the control program that directs its operation. Each of the six servo motors is connected to the Arduino Uno's PWM pins, allowing precise control over the movement of every joint in the arm.

Fig. 1: Block Diagram



A breadboard is used as a common platform to neatly interconnect all the components. An external power supply feeds the breadboard's power and ground rails, distributing the necessary power and ground to all servo motors and other parts uniformly. This hardware setup is critical for ensuring the robotic arm operates smoothly and reliably in real time. Careful connections help prevent loose wiring or unstable power delivery, which could disrupt the arm's function. The motors are securely mounted to the arm structure to maintain stability during movement, minimizing vibrations or jitter. Each motor's control signal is connected to the Arduino UNO's PWM pins, while power and ground are supplied externally to ensure proper operation.

3D PRINTING PROCESS FOR ROBOTIC ARM STRUCTURAL COMPONENTS

The robotic arm parts were fabricated using 3D printing, as shown in the image below. All major mechanical components—including the base, arm links, joints, and gripper—were printed separately and then assembled to create the complete robotic arm structure. After printing, each part was checked for proper fit and alignment, and the servo motors were mounted into their designated slots. The components were carefully assembled step by step using screws to ensure smooth movement at every joint.

Using 3D printing made the robotic arm lightweight, easy to assemble, and ideal for rapid prototyping. To enhance its appearance, two different filaments were used: yellow filament for the moving parts and black filament for the enclosure base parts. This color distinction not only improves aesthetics but also helps visually differentiate functional sections of the arm.

ARDUINO ROBOTIC ARM CODE EXPLANATION

This Arduino code controls a 6-DOF robotic arm by operating servo motors through serial commands. Each servo corresponds to a specific joint—base, shoulder, elbow, wrist rotation, wrist pitch, and gripper—and can be moved smoothly to a target angle within defined safety limits. The arm initializes in a safe starting position and supports real-time position feedback via serial communication.

To enhance usability, a dashboard with sliders was created, allowing intuitive control of the arm. The control data from the sliders is sent via the serial port to the Arduino.

Step 1: Servo Object Declaration and Initialization

- Six Servo objects are created, one for each joint.
- Initial angles are set to 90°, positioning the arm neutrally to reduce mechanical stress and ensure predictable motion.

Step 2: Setup Function: PWM Pin Assignment and Serial Communication

- Servo pins are assigned according to the circuit diagram, using Arduino PWM-capable digital pins.
- SERVO_STEP_DELAY is defined to control the smoothness of servo movement.
- Serial communication is initialized to receive control commands.

Step 3: Initial Position Configuration

- After startup, servos stabilize and move to their predefined initial angles to avoid sudden jerks or unsafe movement.
- A confirmation message is sent via serial to indicate the system is ready.

Step 4: Serial Command Processing and Safety Constraints

- Commands like B90 or S45 are parsed, where the first letter identifies the joint and the number specifies the target angle.
- Constrain () is used to keep servo angles within safe mechanical limits.
- Invalid commands or joint identifiers are rejected to prevent errors.

Step 5: Smooth Motion Control Function

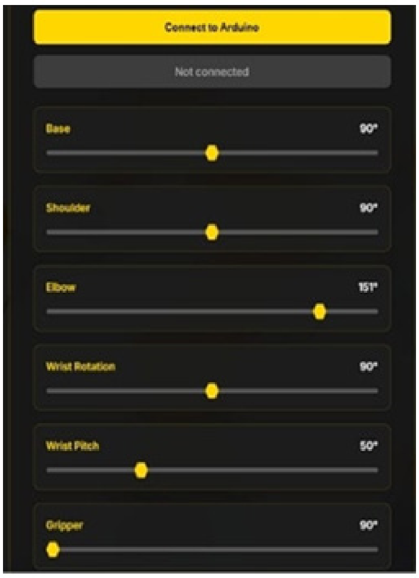
- Instead of moving instantly, servos incrementally move one degree at a time toward the target angle.
- Delays between steps create smooth, realistic motion, reducing vibrations, current spikes, and mechanical wear.

WEB-BASED CONTROL DASHBOARD FOR ARDUINO ROBOTIC ARM

The robotic arm is controlled through a web-based dashboard built with HTML, CSS, and JavaScript. This dashboard is designed as a plug-and-play interface, requiring no additional installation or configuration. Simply save the file with a .html extension and open it in a web browser.

Upon opening, the dashboard connects to the Arduino via Web Serial communication, allowing real-time control of all servo motors using sliders. Each slider corresponds to a specific joint of the robotic arm. The dashboard also offers advanced features such as resetting the arm to its default positions, recording motions, playing back recorded sequences, and saving/loading motion patterns. These capabilities enable smooth, precise, and repeatable operation of the robotic arm.

Fig. 2: Robot Dashboard



REAL-WORLD APPLICATIONS OF ARDUINO ROBOTIC ARMS

- Robotic arms excel in performing tasks in hazardous environments—such as handling chemicals, explosives, or radioactive materials, thereby protecting human workers from danger. Remote-controlled industrial robotic arms are especially valuable in operating safely where human presence is risky or difficult.
- Affordable robotic arms built with Arduino and 3D-printed components serve as excellent educational tools. They offer hands-on experience with robotics principles, automation basics, programming skills, and engineering problem-solving.
- In laboratories, robotic arms efficiently handle test tubes, samples, and repetitive tasks, reducing human error. Pick-and-place arms automate processes like pipetting, specimen organization, and executing standard lab protocols, improving consistency and productivity.
- In medicine, specialized 6-axis robotic arms assist surgeons in delicate procedures requiring high stability and precision. These robotic systems enable minimally invasive surgeries with enhanced accuracy, supporting better patient outcomes.

CONCLUSION

In this paper, we designed and built a DIY 6-DOF robotic arm using an Arduino Uno, servo motors, and custom 3D-printed components, all controlled through a web-based dashboard. The dashboard simplifies operation by enabling users to control each joint in real time via sliders, as well as to record, save, and replay motion sequences, effectively simulating the behavior of basic robotic arms. Through this project, users gain hands-on experience in mechanical assembly, servo calibration, and embedded programming, while also learning the importance of power management and precise control. Overall, this paper provides a valuable learning platform for students and beginners to explore robotics in a practical and engaging manner, laying a solid foundation for future enhancements such as wireless control, automation, AI integration, and IoT applications.

REFERENCES

1. Master-Slave Control of a Teleoperated Anthropomorphic Robotic Arm with Gripping Force Sensing Authors: G. Sen Gupta, S. C. Mukhopadhyay, C. H. Messom, S. Demidenko
2. Adaptive Multi-dimensional Compliance Control of a Humanoid Robotic Arm with Anti-Windup Compensation Authors: Said G Khan, Guido Herrmann, Tony Pipe and Chris Melhuish
3. Dynamic Control of Soft Robotic Arm: A Simulation Study Authors: Milad Azizkhani, Isuru S. Godage, Yue Chen
4. An Inverse Kinematics Method of a Soft Robotic Arm with Three-Dimensional Locomotion for Underwater Manipulation Authors: Zheyuan Gong, Jiahui Cheng, Kainan Hu, Tianmiao Wang
5. Modeling and Adaptive Neural Network Control for a Soft Robotic Arm with Prescribed Motion Constraints Authors: Yan Yang, Jiangtao Han, Zhijie Liu.
6. Teach Pendant at Fingertips: Intuitive Vision Based Gesture-Driven Control of Dexter ER2 Robotic Arm Authors: S. Abhishek, Yash S. Jogi, Umesh Kumar Sahu*, Santanu Kumar Dash, Umesh Kumar
7. Cartesian Space Control and Joint Tracking Control for a Robotic Arm System with Explicit time Proportional Convergence Authors: W. Yan, T. Zhao, B. Niu, Z. Shi, and E. Q. Wu
8. A Modular Robotic Arm Control Stack for Research: Franka-Interface and FrankaPy Authors: Kevin Zhang, Mohit Sharma, Jacky Liang, Oliver Kroemer.
9. Design And Control of a Robotic Arm for Industrial Applications Authors: Sathish Krishna Anumula, SVSV Prasad Sanaboina, Ravi Kumar Nagula, R. Nagaraju.
10. RoboFiSense: Attention-Based Robotic Arm Activity Recognition with WiFi Sensing Authors: Rojin Zandi, Kian Behzad, Elaheh Motamedi, Hojjat Salehinejad, Milad Siami