



# **PATUAKHALI SCIENCE AND TECHNOLOGY UNIVERSITY**

**COURSE CODE EEE-212**

**Electrical Technology Sessional**

---

## **Lab Report**

### **SUBMITTED TO:**

**Md. Naimur Rahman**

**Professor**

**Department of Electrical and Electronics Engineering**

**Faculty of Computer Science and Engineering**

---

### **SUBMITTED BY:**

**MD Mobin Haque**

**ID: 2102064,**

**Registration No: 10191**

**Faculty of Computer Science and Engineering**

---

**Date of submission: 29 October, 2024**

**Project title: Joystick/Button-Controlled Industrial  
Automation System for Multiple Servo Motor Control  
Using Arduino**

# Lab Report:

## Joystick-Controlled Dual Servo Motor

### 1. Project Overview

This project aims to control two servo motors using a joystick module and an Arduino Uno, allowing intuitive manipulation of each servo's position based on joystick movement. The system can be applied in various projects, such as mini robotic arms, pan-tilt camera platforms, and interactive art devices. This report outlines the components, wiring, coding, and functionality, providing a comprehensive guide to the project.

### 2. Objectives

- To use a joystick to control the position of two servo motors.
- To explore the practical applications of servo motor control via a joystick interface.
- To create a flexible system for use in applications such as robotic arms, camera stabilizers, or laser pointer controllers.

### 3. Components

The project uses the following components:

- **Arduino Uno (SMD version):** The central controller, processing the joystick input and controlling the servos.
- **Servo Motors (2 units):** Used to achieve precise angular control based on joystick movement.
- **Joystick Module:** Allows control in two axes (X and Y) through an analog interface.
- **9V Battery with Breadboard Connector:** Provides power to the servos and Arduino.
- **Breadboard and Jumper Wires:** For wiring the components together.

### 4. Circuit Diagram and Setup

The joystick provides analog outputs for X and Y positions, which are read by the Arduino. The servos respond to these values to align with joystick movements.

## Wiring Details

### 1. Joystick Module:

- **GND** to Arduino **GND**
- **VCC** to Arduino **5V**
- **VRx** to Arduino **A0**
- **VRy** to Arduino **A1**

### 2. Servo Motors:

- **Servo 1:**
  - **Signal (Yellow/White)** to Arduino **D9**
  - **Power (Red)** to breadboard + **rail** (connected to 9V battery)
  - **Ground (Brown/Black)** to breadboard **GND rail**
- **Servo 2:**
  - **Signal (Yellow/White)** to Arduino **D10**
  - **Power (Red)** to breadboard + **rail**
  - **Ground (Brown/Black)** to breadboard **GND rail**

### 3. 9V Battery:

- Connect **+9V** to breadboard + **rail** to power the servos.
- Connect **GND** to breadboard **GND rail**.
- Run a wire from the breadboard **GND rail** to the Arduino **GND** for a common ground.

*Note:* Servos typically run on 4.8V to 6V; using a 9V battery may require a voltage regulator for optimal and safe servo performance.

## 5. Code Implementation

The following Arduino code reads joystick values and maps them to servo angles, providing smooth and responsive servo control based on joystick input.

```
#include <Servo.h>

Servo servoX; // X-axis control servo
Servo servoY; // Y-axis control servo

int joyX = A0; // Joystick X-axis
int joyY = A1; // Joystick Y-axis

void setup() {
  servoX.attach(9); // Attach X-axis servo to pin 9
  servoY.attach(10); // Attach Y-axis servo to pin 10
  Serial.begin(9600);
}

void loop() {
  int xVal = analogRead(joyX); // Read joystick X-axis
  int yVal = analogRead(joyY); // Read joystick Y-axis

  int angleX = map(xVal, 0, 1023, 0, 180); // Map to servo angle
```

```
int angleY = map(yVal, 0, 1023, 0, 180);

servoX.write(angleX); // Move X-axis servo
servoY.write(angleY); // Move Y-axis servo

Serial.print("X-Axis: "); Serial.print(angleX);
Serial.print(" | Y-Axis: "); Serial.println(angleY);

delay(15);
}
```

## Explanation of Code

- **Mapping Joystick Values:** The joystick outputs values from 0 to 1023, which the code maps to servo angles from 0 to 180 degrees.
- **Serial Output:** Displays the current position of each servo in the Serial Monitor for easy debugging and feedback.
- **Servo Control:** Each axis of the joystick controls a corresponding servo motor, making it easy to control the servos intuitively.

## 6. Results and Applications

### Results

The project successfully demonstrated that a joystick module could control two servo motors independently. This functionality opens doors to several potential applications, where precise control is needed in two axes.

### Potential Applications

1. **Robotic Arm:** A basic robotic arm where the servos move joints to perform tasks.
2. **Pan-Tilt Camera Control:** Manually control a camera's orientation for photography or security purposes.
3. **Laser Pointer Control:** Use servos to direct a laser pointer's position for fun or interactive tasks.
4. **Art Creation:** Move a drawing tool across two axes to create simple, joystick-guided artwork.

## 7. Challenges and Considerations

- **Power Supply:** Using a 9V battery can provide unstable power to the servos. A voltage regulator or an alternative battery pack (like AA batteries) could provide a more suitable voltage for the servos.

- **Servo Noise:** Servos can be noisy when moving quickly in response to joystick input, which may require delay adjustments for smoother transitions.
- **Joystick Calibration:** Some joystick modules might not be centered at exact values, which can cause minor issues with centering the servos.

## 8. Future Improvements

To enhance the project further:

1. **Add a Button:** Use the joystick's integrated push-button to trigger additional actions, such as switching modes or activating automated movements.
2. **Automatic Stabilization:** Integrate an accelerometer to provide automatic stabilization, which could be useful for gimbal applications.
3. **Multi-Servo Control:** Expand the project to control more servos or add custom movements.

## 9. Conclusion

This project demonstrates how a joystick-controlled dual-servo system can be built easily with an Arduino, serving as a base for robotics, stabilization, and control projects. It allows for simple, precise control and provides insights into working with analog sensors and servos. The project is a valuable learning experience in interfacing with hardware and controlling actuators in real-time.