Contents

Abstract……………………………………………………1

Introduction…………………………………………………1

Components Used…………………………………………2

Explanation of the parts used…………………………3-7

Method of installation……………………………….8-12

Potential Applications………………………………..13

Review and Development Potential………………..14-15

Conclusion…………………………………………………..16

References

1-Abstract

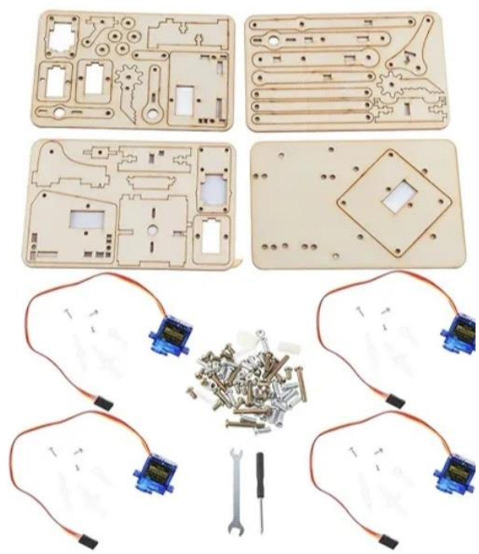
This project involves the design and implementation of an industrial robotic arm controlled by an Arduino microcontroller. The robotic arm comprises four SG90 servo motors, each manipulated through the adjustment of variable resistors. The goal is to create a flexible and easily controllable robotic arm for industrial applications.

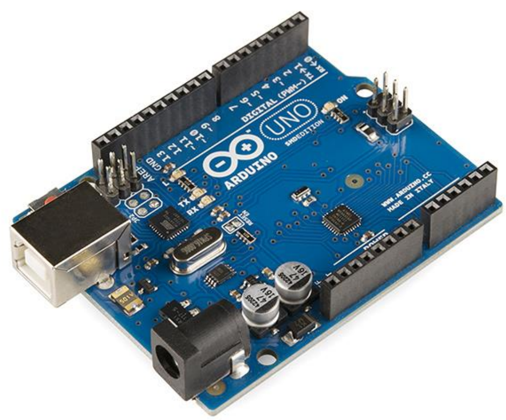
2- Introduction

In the realm of industrial automation, the integration of robotic arms has become an indispensable aspect, enhancing efficiency and precision in various manufacturing processes. The project at hand focuses on the installation of an industrial arm employing four SG90 servo motors, all meticulously controlled by an Arduino microcontroller. The innovation lies in the utilization of four variable resistors as the primary interface, allowing seamless and intuitive control over the robotic arm's movements. This fusion of servo motors, Arduino technology, and variable resistors not only showcases the integration of cutting-edge components but also promises to deliver a cost-effective and versatile solution for industrial applications. This report provides an in-depth analysis of the project, detailing the design, functionality, and potential applications of the SG90 servo motor-controlled industrial arm.

3-Components Used

- 4 x SG90 servo motors
- Arduino microcontroller
- 4 x variable resistors
- Breadboard and jumper wires
- Power supply
- Mechanical arm structure





4-Explanation of the parts used

What is a Servo motor?

A servo motor is a type of motor used in various applications to precisely control the position, speed, and angular rotation of mechanical systems. It's designed to provide accurate control of its output shaft, allowing for precise angular position control.

Servo motors typically consist of a small DC motor, a gear train, a control circuit, and a feedback system. The feedback system, often a potentiometer or an encoder, continuously sends information about the motor's current position back to the control circuit. This feedback allows the control circuit to adjust the motor's movement and ensure it reaches and maintains the desired position.

Types of Servo Motors?

Servo Motors are divided into two main types:

1-AC Servo Motor

2-DC Servo Motor

AC Servo motors use an AC power source, these motors use encoders for feedback. They are known for their high-speed and high-torque capabilities, making them suitable for industrial machinery and CNC equipment.

DC Servo motors use a DC power source and are often equipped with encoders for feedback. They offer precise control and are commonly used in robotics and small-scale automation.

In this tutorial, we are going to use a DC Servo Motor. If you want to learn more about different types of DC motors in general, You can do that and come back here again.

DC Servo Motors are classified into two types

Standard rotation Servo which moves from 0 to 180 degrees

Continuous rotation Servo which moves 360 degrees

We will be using a “Tower pro-Micro Servo sg90 motor” which is a type of DC standard rotation servo motor.

Specifications of Tower Pro SG90 Servo Motor?

Here are the specifications of a standard SG90 Servo we should know before getting using a servo motor for your project:

Operating Voltage	4.8 V - 6 V
Stall Torque	1.6 kg-cm
Stall Current	650 mA
Weight	9g
Operating temperature	- 30 to +60 degree Celsius

<u>Operating Voltage</u>: This servo motor operates within a voltage range of 4.8V to 6V. Ensure the power supply falls within this range to prevent damage or inefficient operation.

Stall Torque: This indicates the maximum torque output of the motor when it's unable to rotate further. It's the force the motor can exert at a given voltage, crucial for applications that require the motor to overcome resistance or hold positions against external forces.

Stall Current: Stall current refers to the amount of current drawn by a motor when it's operating at stall or maximum load condition, and the shaft is prevented from rotating. It signifies the maximum current which the motor can draw.

Weight: The weight of the servo motor is essential in applications where size and weight constraints are critical, such as in drones, small robotic arms, or RC vehicles.

Operating Temperature: Operating temperature refers to the range of temperatures within which a device or component can safely and effectively operate without risking damage or performance degradation.

The TG9e boasts the same performance as other servo's 10x the price with a .10sec travel time and up to 1.5kg in torque and an ultra narrow dead bandwidth!

The TG9e performance is on par with the famous HXT900, however the TG9e isn't as resistant to crashes or over-loading.

Please always ensure your control surfaces are bind free.

Spec.

Dimension: 23x12.2x29mm

Torque: 1.5kg/cm (4.8V)

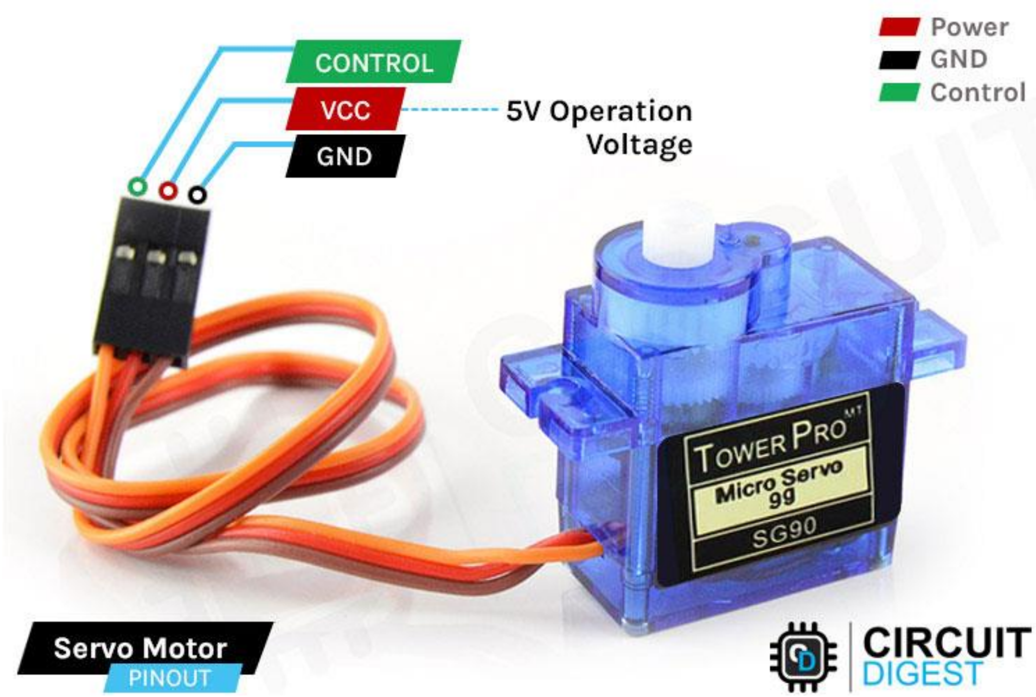
Operating speed: 0.10sec/60 degree 0.09sec/60 degree(6.0V)

Operating voltage: 4.8V

Temperature range: 0-55C

Dead band-width: 7us

Lead Length: 260mm



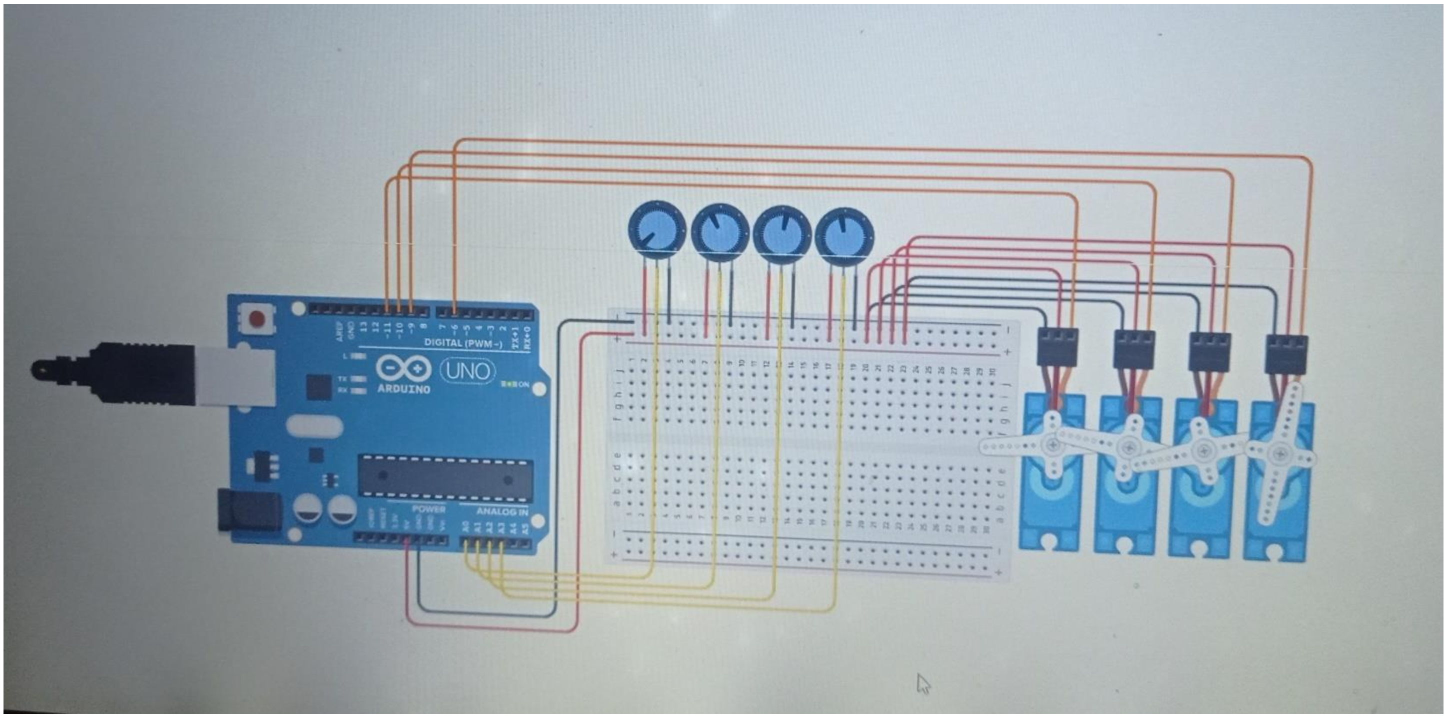
Servo Motor Working?

A servo motor receives PWM (Pulse Width Modulation) signals to determine its angle of rotation. The PWM signal is a square wave with a variable duty cycle, where the width or duration of the high signal (the "on" time) corresponds to a specific angle. This signal is usually generated by a microcontroller, such as an Arduino.

When the PWM signal is fed to the servo motor's control input (usually the orange or yellow wire), the motor's control circuit interprets it as a command to move to a particular angle. The control circuit measures the duration of the high part of the PWM signal, which corresponds to the desired angle. This duration is typically in the range of 1 to 2 milliseconds.

The control circuit then compares this measured duration with the center position (typically 1.5 milliseconds) and adjusts the motor's shaft to move towards the desired angle. If the duration of the high part of the PWM signal is shorter than the center position, the motor turns in one direction; if it's longer, the motor turns in the opposite direction.

5-Method of installation



We connected four variable resistors to the board. We connected their electrodes to the board, and their control terminals were connected to the Arduino with the analog inputs. As for the servo parts, we attached the electrodes to the board, and their control terminals were connected to the Arduino with the following digital inputs (-11,-10,-9, -6) As shown in the previous picture

****Arduino Code:***

```
#include <Servo.h>
Servo myservo,myservo1,myservo2,myservo3;
int sensor=0;
int angle=0;

int sensor1=0;
int angle1=0;

int sensor2=0;
int angle2=0;

int sensor3=0;
int angle3=0;

void setup() {
 myservo.attach(6);
 myservo1.attach(9);
 myservo2.attach(10);
 myservo3.attach(11);
```

```
}
void loop() {
  sensor=analogRead(A0);
  angle=map(sensor,0,1023,0,180);
  myservo.write(angle);
  delay(15);
  sensor1=analogRead(A1);
  angle1=map(sensor1,0,1023,0,180);
  myservo1.write(angle1);
  delay(15);
  sensor2=analogRead(A2);
  angle2=map(sensor2,0,1023,0,180);
  myservo2.write(angle2);
  delay(15);
  sensor3=analogRead(A3);
  angle3=map(sensor3,0,1023,0,180);
  myservo3.write(angle3);
  delay(15);
}
```

This Arduino code is for controlling four servo motors using an array of analog sensors. The code reads the analog values from the sensors (A0, A1, A2, A3) and maps them to an angle value between 0 and 180 degrees. The angle value is then used to control the corresponding servo motor.

Here's a breakdown of the code:

1_Include

This line includes the Servo library, which provides functions to control servo motors.

2....Declare four Servo objects

(myservo, myservo1, myservo2, myservo3) and four integer variables (sensor, angle, sensor1, angle1, sensor2, angle2, sensor3, angle3) to store the sensor values and the corresponding angle values.

3. **Setup Function:

```
void setup() {
  myservo.attach(6);
  myservo1.attach(9);
```

```
  myservo2.attach(10);
  myservo3.attach(11);
}
```

In the `setup` function, each servo object is attached to a specific digital pin (6, 9, 10, 11). This initializes the servo motors to those pins.

4. **Loop Function:

```
void loop() {
  sensor = analogRead(A0);
  angle = map(sensor, 0, 1023, 0, 180);
  myservo.write(angle);
  delay(15);
  // Repeat the same block of code for the other three servo motors (servo1, servo2, servo3).
}
```

In the `loop` function, the code continuously reads analog values from sensors connected to pins A0, A1, A2, and A3. The analog values are then mapped from the range 0-1023 to the range 0-180 (suitable for servo control). The corresponding servo is then moved to the calculated angle. A small delay of 15 milliseconds is added to provide stability and avoid rapid changes in servo positions.

Overall, the code allows you to control four servo motors based on analog sensor readings, with each servo connected to a different analog pin. The position of each servo is determined by the analog sensor reading from its respective pin.

6-Potential Applications

The versatility of this industrial arm control system opens the door to a myriad of applications, including:

Material Handling: Precise control allows for safe and efficient material handling in manufacturing processes.

Assembly Line Operations: The arm can be programmed for intricate assembly tasks, improving production efficiency.

Quality Control: The system can be employed for precise inspection and quality control procedures.

Research and Development: Ideal for prototyping and testing in research and development environments.

7-Reviw and Development Potential

1-Pointe Strengths:

.

*Cost-Effective Solution: The use of SG90 servo motors and Arduino components provides a cost-effective alternative without compromising on performance. This makes the project accessible for a wide range of industrial applications.

*Intuitive Control Interface: The incorporation of variable resistors as control knobs enables an intuitive and user-friendly control mechanism. Operators can easily manipulate the arm's movements, facilitating efficient and adaptive operation.

*Versatility in Applications: The adaptability of the system for various industrial tasks, such as material handling, assembly line operations, and quality control, underscores its potential impact across diverse sectors.

*Real-Time Feedback: The inclusion of feedback mechanisms, such as LED indicators or display screens, enhances the system's usability by providing real-time information on the arm's position and status.

2. Development Possibilities:

*Enhanced Precision and Accuracy: Further refinement of the control algorithm and calibration routines can contribute to increased precision and accuracy in the arm's movements. This is particularly crucial for applications demanding intricate and high-precision tasks.

*Integration of Sensory Feedback: Incorporating sensors, such as force or proximity sensors, could enhance the system's ability to interact with the environment intelligently. This would add an extra layer of adaptability, especially in dynamic industrial settings.

*Wireless Control: Implementing wireless control mechanisms, such as Bluetooth or Wi-Fi connectivity, could offer greater flexibility in operation. This would enable remote control and monitoring, expanding the system's usability in scenarios where physical proximity is a limitation.

*Advanced Programming and Automation: The inclusion of advanced programming capabilities, perhaps through the integration of machine learning algorithms, could enable the arm to learn and adapt to different tasks autonomously, taking industrial automation to a higher level.

*Expandability and Modular Design: Designing the system with an emphasis on expandability and modularity would allow for easy integration of additional servo motors or functionalities, making it a scalable solution for evolving industrial needs.

8-Conclusion

This project successfully demonstrates the integration of SG90 servo motors and Arduino technology to create a robust and cost-effective industrial arm control system. The use of variable resistors as the control interface ensures user-friendly operation, making it a valuable asset in various industrial settings. The flexibility and adaptability of this system position it as a viable solution for tasks requiring precision, control, and versatility in automation.

FACULTY OF ENGINERING
ELELECTERCITY DEP
COMPUTER AND CONTROL
THIRD YEAR
SECOND SEMESTER

PUBLIC OF YEMEN
HIGH EDUCATION
SANA'A UNIVERSITY

ELECTRIC MACHINERY

MeArm-Robotic Arm Control

Done by:

Mohammed Arif (202270203)

Supervision by:

Dr. Mohammed Al-Yadumi

Eng. Ali-Al-Yaqahbi

Sana'a 2024