

MI&T

BS (EE)-2K19 SECTION: B (G-1)

Lab project Final Report



SUBMITTED BY:

NAME	REGISTRATION NUMBER
Muhammad Atif Riaz	19-EE-011
Muhammad Usman Ali	19-EE-092
Zulnorain Bashir	19-EE-006

Introduction:

A Robotic arm is a type of mechanical arm, usually programmable, with similar functions to a human arm; the arm may be the sum total of the mechanism or may be part of a more complex robot. The links of such a manipulator are connected by joints allowing either rotational motion (such as in an articulated robot) or translational (linear) displacement. They are mainly used for pick and place work, application of sealant, assembly operations, handling machine tools and arc welding. In this project we have used five Servo Motors to show the movement of a robotic arm.

Design Steps and Circuit Explanation:

Our project consists of main parts:

- 1) Servo Motors (Robotic Arms).

Procedure:

- Firstly, we started our working on the servo motor motion to explain the movement of arms in different directions . In this project I have used 5 positions using servo motors and potentiometers more can be used according to your project requirements.

Explanation.

Design:

- 1) Arduino Uno
- 2) Rotary Potentiometer.
- 3) Atmel 89C51.
- 4) Servo Motors.
- 5) ULN2003A .
- 6) Quartz Crystal, Logic Probe and Spark fun Push Button.
- 7) Resistors (100 and $8.2k\Omega$) and Capacitors (1nF).
- 8) Electrolytic Capacitor (10u, 16V, 30mA)

Working:

Our Robotic Arm model uses Atmel 89C51 Micro-controller for Servo Motors.

The basic principle of our project is that it takes input from the keypad and the number of distance units to be moved. Every key on Keypad defines a movement.

1 (UPWARD)

2 (DOWNWARD)

3 (LEFT)

4 (RIGHT)

5 (STOP)

Then the user inputs the number of distance units required for locomotion. Furthermore, for two axial motions (UPWARD & DOWNWARD) one servo motor is used. When the arm is moving upward the motor is rotating anti-clockwise. In reality, a robotic arm uses gears to do locomotion. So, we used this principle to show the movement of arms (by the rotation of servo motors).

Servo Motor 1. UPWARD: Anti-clockwise. DOWNWARD: Clockwise.

Servo Motor 2. FORWARD: Anti-clockwise. REVERSE: Clockwise.

Servo Motor 3. LEFT: Anti-clockwise. RIGHT: Clockwise. Servo Motor

The coding is done in arduino.

Code:

```
#include <Servo.h>
```

```
Servo servo1;
```

```
Servo servo2;
```

```
Servo servo3;
```

```
const int LED1 = 2;
```

```
const int LED2 = 3;
```

```
const int LED3 = 4;
```

```
const int LED4 = 7;
```

```
const int LED5 = 8;
```

```
const int button1 = 12;
```

```
const int button2 = 13;
```

```
int button1Presses = 0;
```

```
boolean button2Pressed = false;
```

```
const int pot1 = A0;
```

```
const int pot2 = A1;
```

```
const int pot3 = A2;
```

```
int pot1Val;
```

```
int pot2Val;
```

```
int pot3Val;
```

```
int pot1Angle;
```

```
int pot2Angle;
```

```
int pot3Angle;
```

```
int servo1PosSaves[] = {1,1,1,1,1};
```

```
int servo2PosSaves[] = {1,1,1,1,1};
```

```
int servo3PosSaves[] = {1,1,1,1,1};
```

```
void setup() {
```

```
    servo1.attach(5);
```

```
    servo2.attach(6);
```

```
    servo3.attach(9);
```

```
    pinMode(LED1, OUTPUT);
```

```
    pinMode(LED2, OUTPUT);
```

```
    pinMode(LED3, OUTPUT);
```

```
    pinMode(LED4, OUTPUT);
```

```
    pinMode(LED5, OUTPUT);
```

```
    pinMode(button1, INPUT);
```

```
    pinMode(button2, INPUT);
```

```
Serial.begin(9600);  
}  
  
void loop() {  
  
    pot1Val = analogRead(pot1);  
    pot1Angle = map(pot1Val, 0, 1023, 0, 179);  
    pot2Val = analogRead(pot2);  
    pot2Angle = map(pot2Val, 0, 1023, 0, 179);  
    pot3Val = analogRead(pot3);  
    pot3Angle = map(pot3Val, 0, 1023, 0, 179);  
  
    servo1.write(pot1Angle);  
    servo2.write(pot2Angle);  
    servo3.write(pot3Angle);  
  
    if(digitalRead(button1) == HIGH){  
        button1Presses++;  
        switch(button1Presses){  
            case 1:  
                servo1PosSaves[0] = pot1Angle;  
                servo2PosSaves[0] = pot2Angle;  
                servo3PosSaves[0] = pot3Angle;  
                digitalWrite(LED1, HIGH);  
                Serial.println("Pos 1 Saved");  
                break;  
            case 2:  
                servo1PosSaves[1] = pot1Angle;  
                servo2PosSaves[1] = pot2Angle;  
                servo3PosSaves[1] = pot3Angle;  
                digitalWrite(LED2, HIGH);  
                Serial.println("Pos 2 Saved");  
                break;
```

case 3:

```
servo1PosSaves[2] = pot1Angle;  
servo2PosSaves[2] = pot2Angle;  
servo3PosSaves[2] = pot3Angle;  
digitalWrite(LED3, HIGH);  
Serial.println("Pos 3 Saved");  
break;
```

case 4:

```
servo1PosSaves[3] = pot1Angle;  
servo2PosSaves[3] = pot2Angle;  
servo3PosSaves[3] = pot3Angle;  
digitalWrite(LED4, HIGH);  
Serial.println("Pos 4 Saved");  
break;
```

case 5:

```
servo1PosSaves[4] = pot1Angle;  
servo2PosSaves[4] = pot2Angle;  
servo3PosSaves[4] = pot3Angle;  
digitalWrite(LED5, HIGH);  
Serial.println("Pos 5 Saved");  
break;
```

```
}
```

```
}
```

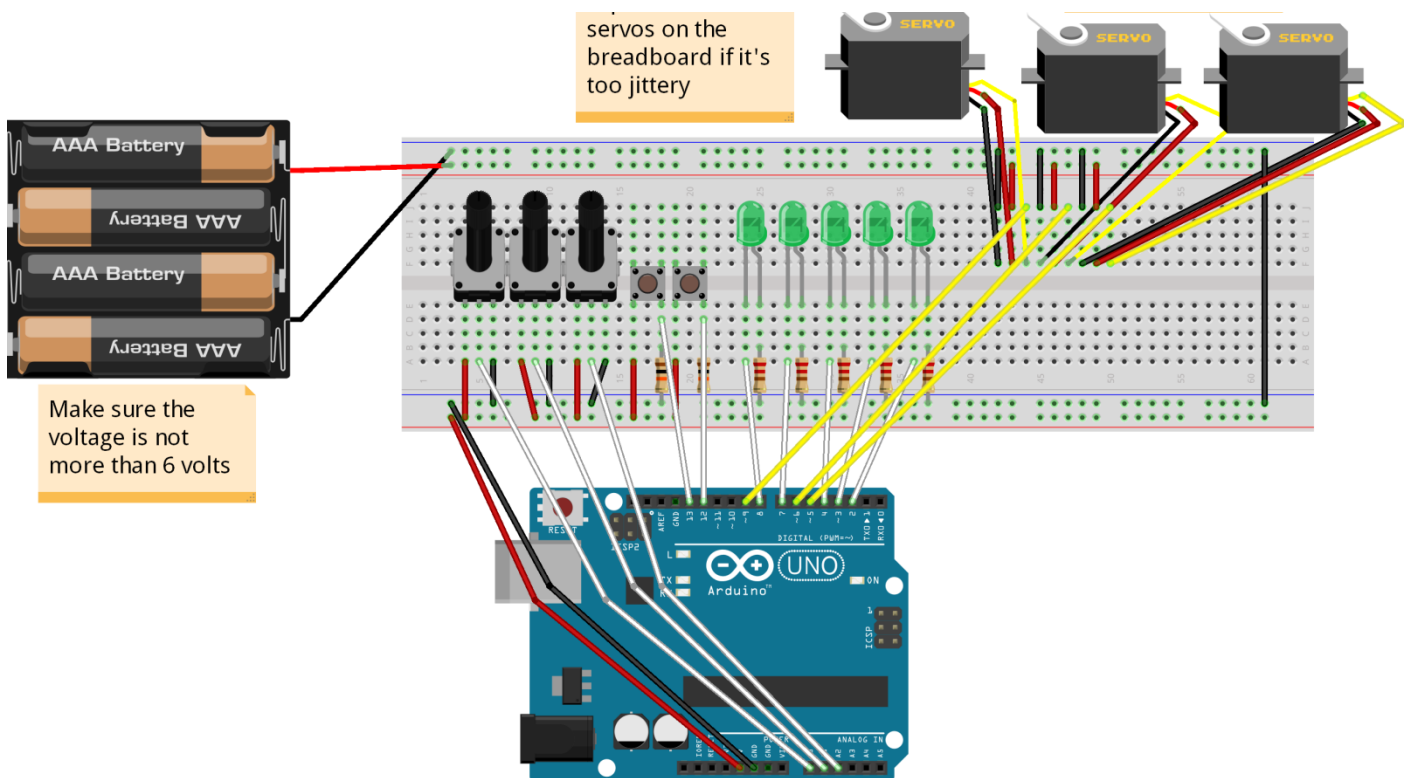
```
if(digitalRead(button2) == HIGH){  
    button2Pressed = true;  
}
```

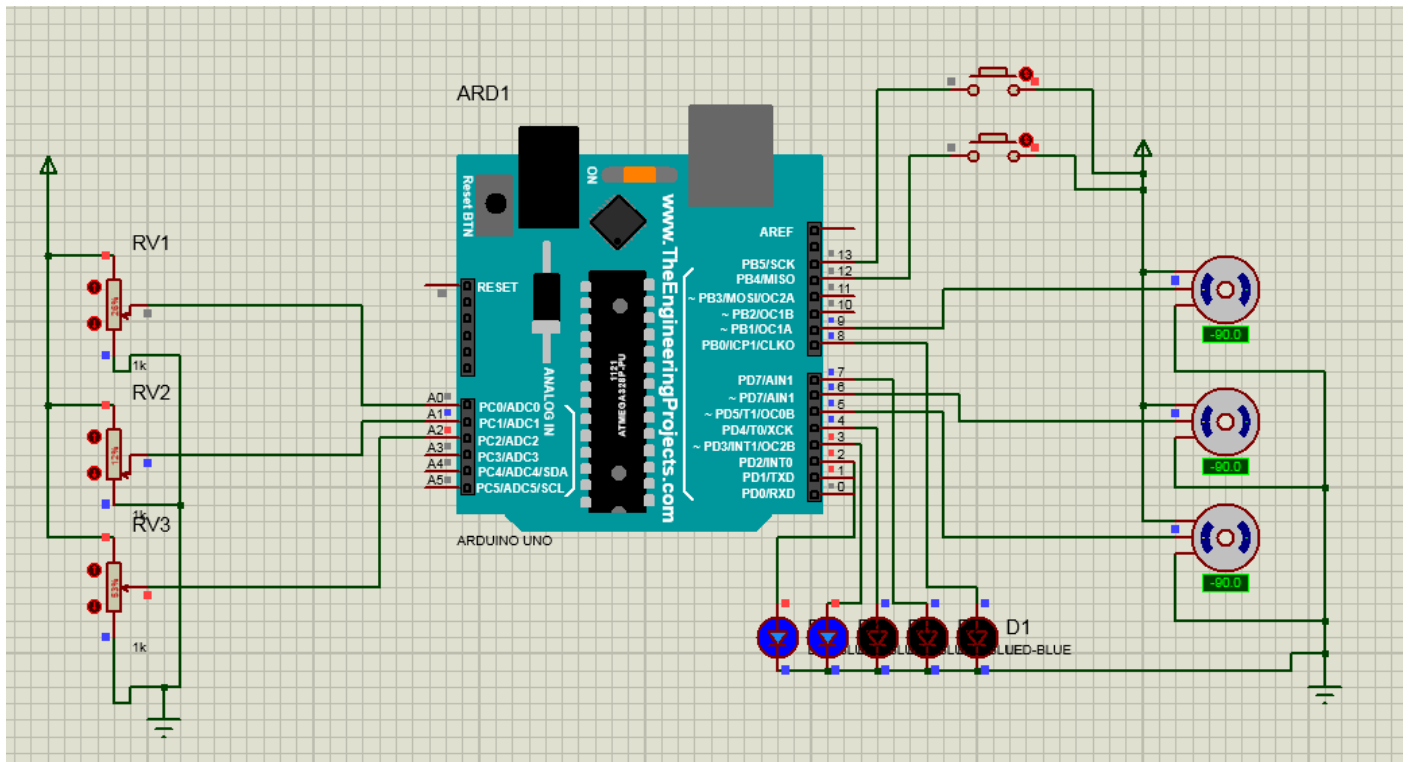
```
if(button2Pressed){  
    for(int i = 0; i < 5; i++){  
        servo1.write(servo1PosSaves[i]);  
        servo2.write(servo2PosSaves[i]);  
        servo3.write(servo3PosSaves[i]);  
        Serial.println(" potentiometer Angles: ");
```

```

Serial.println(servo1PosSaves[i]);
Serial.println(servo2PosSaves[i]);
Serial.println(servo3PosSaves[i]);
delay(1050);
}
}
delay(300);
}

```





Results and Achievements:

Results Achieved:

We have successfully created Robotic Arm using Atmel 89C51. The Servo Motors work properly and show the locomotion as displayed.

Recommended Future Work:

The things that could have been added were:

- Using the cartesian coordinates (x, y, z) input from user to define the locomotion of Robotic Arm.
- Saving the movement and the automatic locomotion.