Department of Electronics and Computer
Science **Pillai College Of Engineering**

(Autonomous)

New Panvel – 410 206

S.E.2023-2024

A Project On

Servo motor Interfacing with 8051 Microcontroller

Submitted by

Member 1: 430 Om Khairnar

Member 2: 434 Ayush Dugade

Member 3: 435 Gopal K S

Supervisor

Guide: Dr Ravi Biradar

Introduction

Servo motors are sophisticated devices widely used in automation, robotics, and control systems due to their ability to provide precise angular positioning and controlled motion. Unlike standard motors, servo motors incorporate a feedback mechanism that allows them to maintain a specific position accurately, making them invaluable in applications where precise control is crucial.

The 8051 microcontroller, a popular choice in embedded systems, provides a versatile platform for interfacing with various external devices, including servo motors. By generating Pulse Width Modulation (PWM) signals, the 8051 microcontroller can precisely control the position and movement of servo motors, opening up a wide range of possibilities for automation, robotics, and other applications.

This project delves into the intricacies of interfacing a servo motor with an 8051 microcontroller using PWM signals. The objective is to demonstrate the principles behind servo motor control, the design process involved in generating PWM signals, and the advantages and limitations of this interfacing method. Through this project, we aim to showcase the capabilities of the 8051 microcontroller in controlling sophisticated electromechanical systems like servo motors, paving the way for innovative and efficient embedded systems development.

Principle Behind the Circuit

The principle behind interfacing the servo motor with the 8051 microcontroller lies in generating Pulse Width Modulation (PWM) signals. The code configures Timer 0 of the 8051 microcontroller in Mode 1 to generate PWM signals. The PWM duty cycle determines the position of the servo motor, with different duty cycles corresponding to various angular positions.

Components Required

• 8051 Microcontroller (AT89C51)

• Servo Motor

• Resistor 10K

• Capacitor 22pF (x2)

• Capacitor 10uF

• Crystal Oscillator (11.0592MHz)

• 5v DC Supply

How to Design

Hardware Setup:

- **Connect the control pin of the servo motor to a PWM-capable pin of the 8051 microcontroller (e.g., P2^0).**
- **Ensure proper power and ground connections for the servo motor.**

Software Design:

Timer Configuration: Set Timer 0 of the 8051 microcontroller in Mode 1 (`TMOD=0x01`) to generate PWM signals.

PWM Signal Generation: Use the `servo_delay` function to create delays in multiples of 50 microseconds, creating the PWM signal for servo control.
Main Loop: The main loop in the code varies the PWM duty cycle (`n` from 13 to 27 with a step of 2), controlling the servo motor's position.

Advantages

- Precision: Servo motors offer precise angular positioning, making them suitable for applications requiring accuracy.
- Ease of Control: The 8051 microcontroller's PWM capabilities simplify the control of servo motors.
- Versatility: Servo motors can be used in a wide range of applications, from robotics to camera systems.

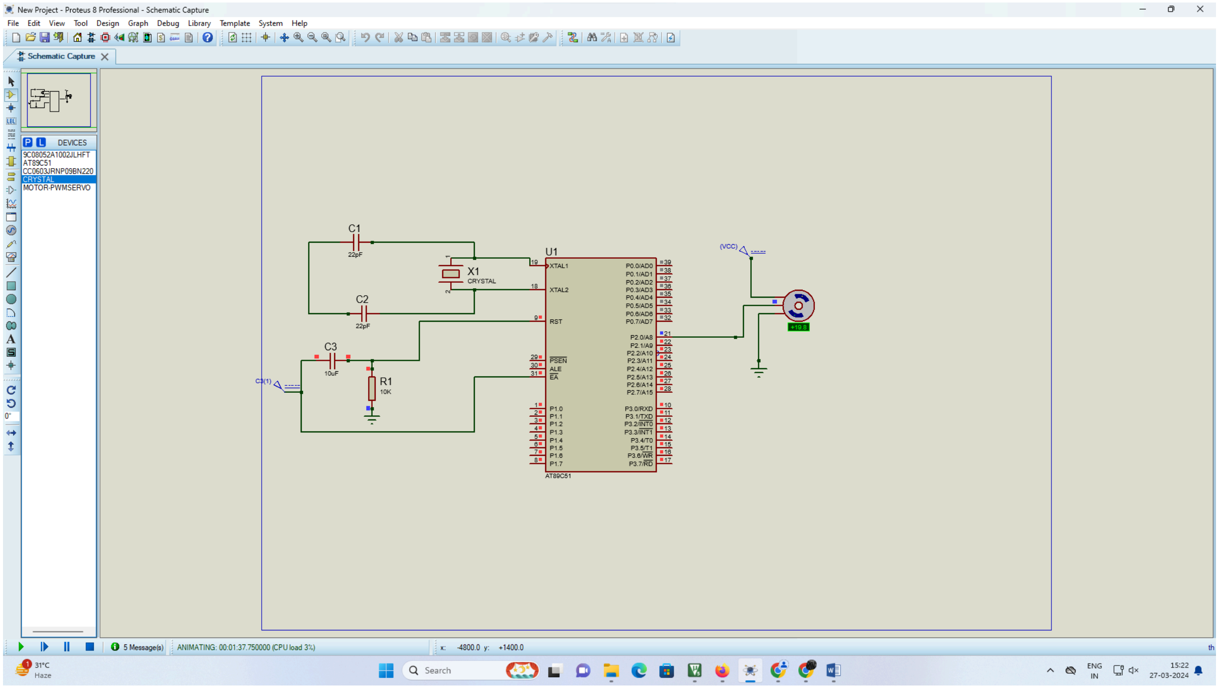
Applications

- Robotics: Used in robotic arms, grippers, and joints for precise movements.
- Automation: Control mechanisms and actuators in automated systems.
- Camera Systems: Pan-tilt mechanisms for surveillance and tracking.

Limitations

- Limited Range: Servo motors typically have a limited range of motion (e.g., 180 degrees), which can be a limitation in certain applications.
- Torque: Depending on the servo motor's specifications, torque may be limited compared to other motor types.
- Cost: High-quality servo motors can be expensive, impacting project budgets.

Output :



New Project - Proteus 8 Professional - Schematic Capture

DEVICES
9C08052A1002JLHFT
AT89C51
CC0603JRNP09BN220
CRYSTAL
MOTOR-PWMSERVO

5 Message(s)   ANIMATING: 00:01:37.750000 (CPU load 3%)   x: -4800.0 y: +1400.0

Code :

```
#include<reg51.h>
sbit output=P2^0;
void msdelay(unsigned int time)  // Function for creating delay in milliseconds.
{
   unsigned i,j ;
   for(i=0;i<time;i++)
   for(j=0;j<1275;j++);
}
void servo_delay(int times)     // Creating Delay in multilple of 50us using 8051 Timers
{
   int m;
   for(m=0;m<times;m++)
   {
      TH0=0xFF;
      TL0=0xD2;
      TR0=1;
      while(TF0==0);
      TF0=0;
      TR0=0;
   }
}
void main()
{
   int n;
   TMOD=0x01;          // Selecting Timer 0, Mode 1
   output=0;
   while(1)
   {
      for(n=13;n<28;n=n+2)
      {
         output=1;
         servo_delay(n);
         output=0;
         servo_delay(260);
         msdelay(200);
      }
   }
}
```