SIT 111 COMPUTER SYSTEMS

TASK 3.6 p : Servo Motor Control with Arduino

Summary:

In this task I worked with a software called Arduino uno. This task focused on writing a code and the objective was to understand the basics of servo motor operations and how to interface them with microcontrollers for precise rotational movement control. The materials I required were

- Arduino Uno (or similar Arduino board)
- Servo Motor (e.g., SG90 micro servo motor)
- Breadboard
- Jumper wires
- USB cable to connect the Arduino to a computer
- Arduino IDE installed on the computer

Personal summary :

I followed the sample circuit picture on the task sheet and formed the circuit in the same way. For this task the code was already provided so initially we worked on that as it aligned with my previous learning experiences as well so it was easy to understand the code. We included a new library called servo.h and simultaneously created a new servo object, defined the servo pins, attached the servo variable to a pin , defined the angles and so and so. I then verified and compiled and uploaded my code .

Module summary:

Motor control basics:

Motor control is nothing but the process of managing the operation and movement of motors in various applications. Motors are essential components in countless devices and machines, from simple household appliances like washing machines to advanced industrial robots.

DC motor control:

The DC motors convert electrical energy from a power source, like a battery or power supply, into mechanical motion. Since DC motors can rotate in both the directions they are quite versatile for many tasks. Using Arduino can control DC motors by adjusting the voltage supplied to them, control their movements. Some of the key features of it are :

Versatile actutators:

Bidirectional movement:

Speed control:

Servo motor control:

Servo motors unlike the DC motors have the ability to move to specific angles with great accuracy. They take continuous feedback about their position and adjust their rotation accordingly. Arduino can be used to control servo motors, making them a valuable component in many DIY projects and robotics applications. We can summarise the key features as follows :

Precision positioning

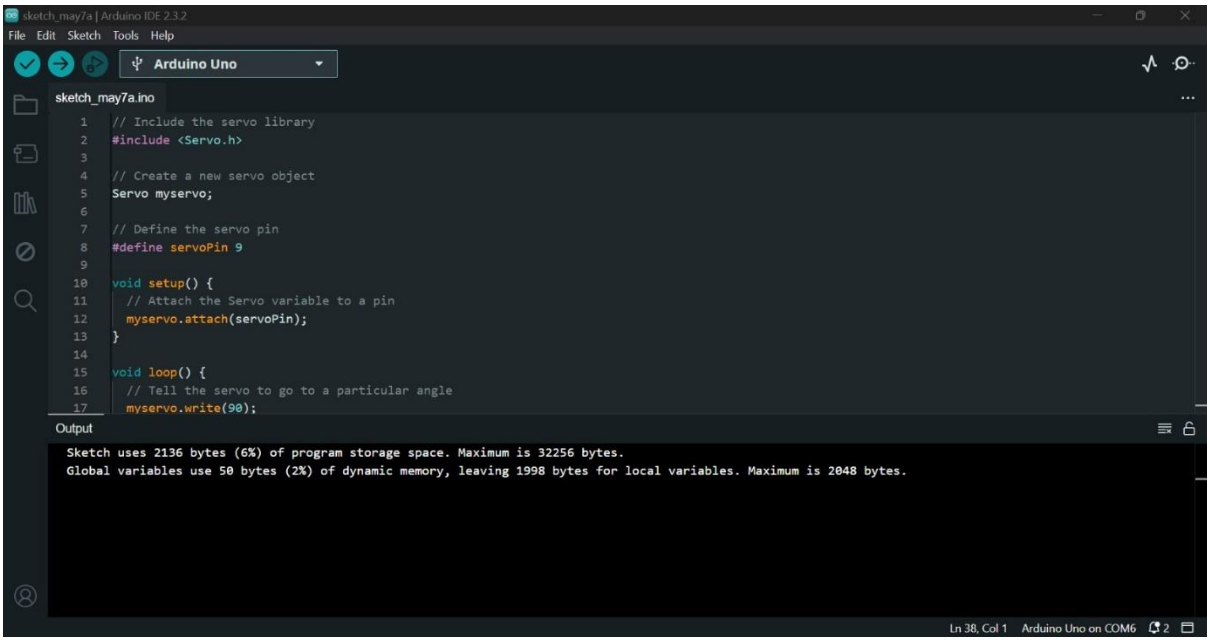
Angle feedback

Limited angular range

Servo control:

Servo motors are quite versatile and precise and they are controlled using a method known as pulse width modulation. In PWM the width or duration of the PWM signal is used to specify the position or angle at which the servo motor should rotate. We can control the servo’s movement by adjusting the width of the signal.

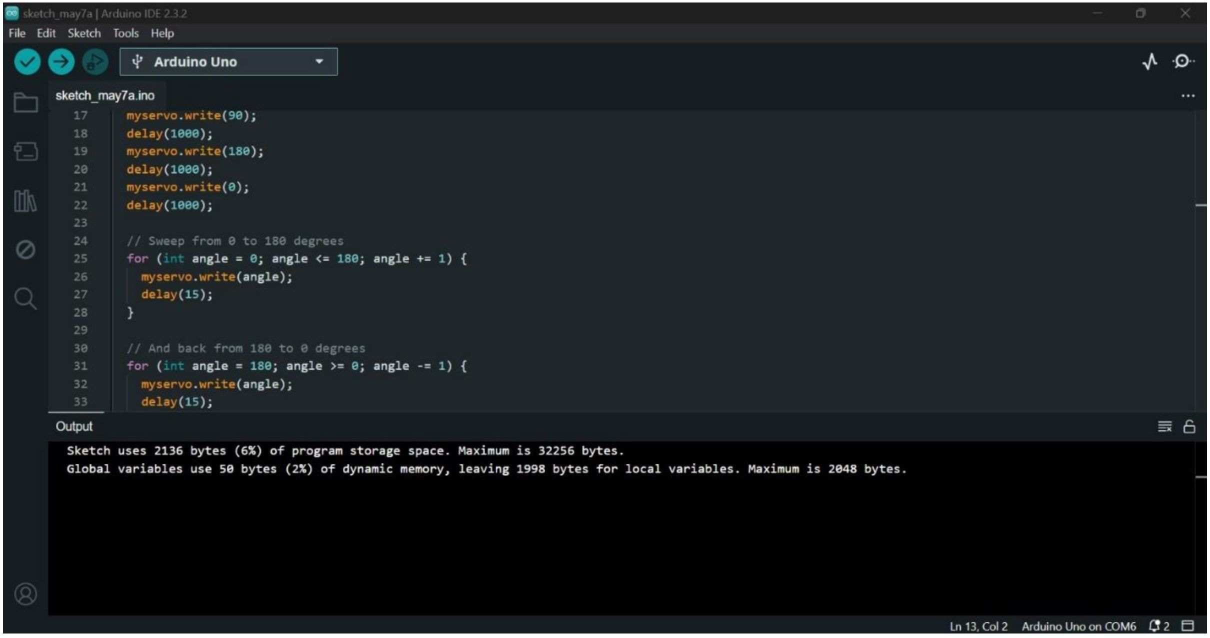
Learning journey :



sketch_may7a | Arduino IDE 2.3.2
File Edit Sketch Tools Help
Arduino Uno
sketch_may7a.ino
Output
Ln 38, Col 1 Arduino Uno on COM6 2

```
// Include the servo library
#include <Servo.h>

// Create a new servo object
Servo myservo;

// Define the servo pin
#define servoPin 9

void setup() {
  // Attach the Servo variable to a pin
  myservo.attach(servoPin);
}

void loop() {
  // Tell the servo to go to a particular angle
  myservo.write(90);
```

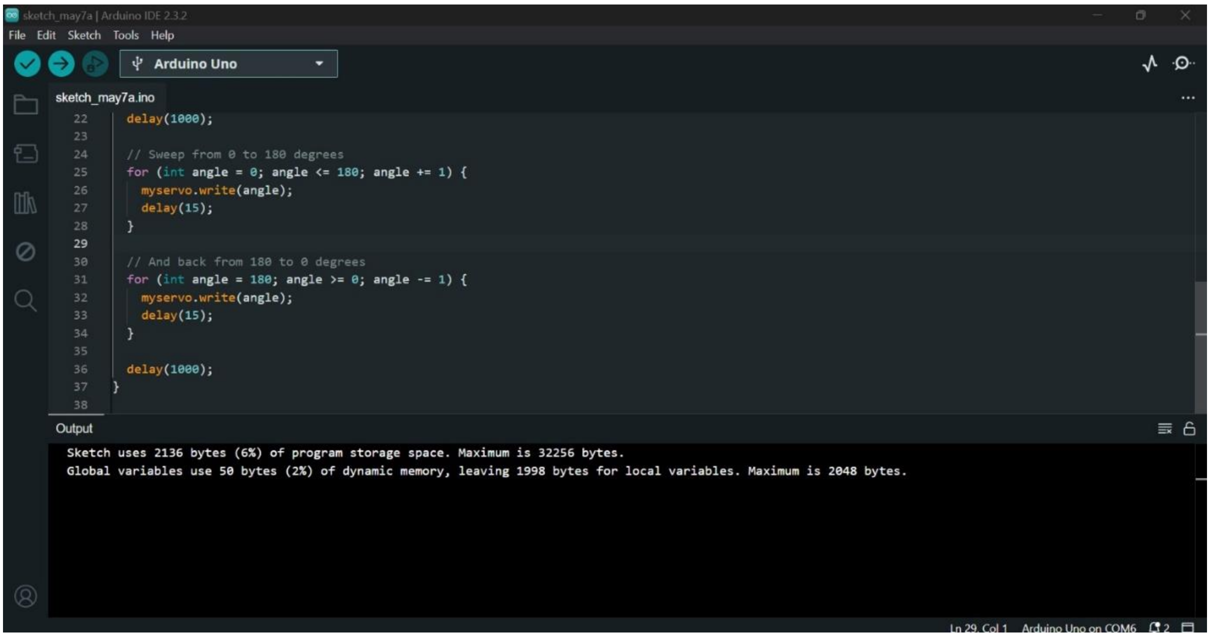
```
Sketch uses 2136 bytes (6%) of program storage space. Maximum is 32256 bytes.
Global variables use 50 bytes (2%) of dynamic memory, leaving 1998 bytes for local variables. Maximum is 2048 bytes.
```



sketch_may7a | Arduino IDE 2.3.2
File Edit Sketch Tools Help
Arduino Uno
sketch_may7a.ino
Output
Ln 13, Col 2 Arduino Uno on COM6 2

```
  myservo.write(90);
  delay(1000);
  myservo.write(180);
  delay(1000);
  myservo.write(0);
  delay(1000);

  // Sweep from 0 to 180 degrees
  for (int angle = 0; angle <= 180; angle += 1) {
    myservo.write(angle);
    delay(15);
  }

  // And back from 180 to 0 degrees
  for (int angle = 180; angle >= 0; angle -= 1) {
    myservo.write(angle);
    delay(15);
```

```
Sketch uses 2136 bytes (6%) of program storage space. Maximum is 32256 bytes.
Global variables use 50 bytes (2%) of dynamic memory, leaving 1998 bytes for local variables. Maximum is 2048 bytes.
```



sketch_may7a | Arduino IDE 2.3.2
File Edit Sketch Tools Help
Arduino Uno
sketch_may7a.ino
Output
Ln 29, Col 1 Arduino Uno on COM6 2

```
  delay(1000);

  // Sweep from 0 to 180 degrees
  for (int angle = 0; angle <= 180; angle += 1) {
    myservo.write(angle);
    delay(15);
  }

  // And back from 180 to 0 degrees
  for (int angle = 180; angle >= 0; angle -= 1) {
    myservo.write(angle);
    delay(15);
  }

  delay(1000);
}
```

```
Sketch uses 2136 bytes (6%) of program storage space. Maximum is 32256 bytes.
Global variables use 50 bytes (2%) of dynamic memory, leaving 1998 bytes for local variables. Maximum is 2048 bytes.
```

Reflections :

How do you know you have achieved the learning goals?

I learned about the basic commands and some new ways to use Arduino in this task and also got familiar with the usage of Arduino uno. Got my hands on compiling the circuit and knowing about the components. This was my first time working with a motor and I found it quite interesting .

- What is the most important thing you learned from this and why?

For me, the most important thing was formation of the circuit, as it was somewhat different from others and as mentioned above I worked with a motor for the first time, so learning about different types of motor in detail was something I found quite important.

- How does the content or skills learned here relate to things you already know?

I already knew about the coding for this task I.e the commands and inputs. It only deepens my understanding.

- Where or when do you think it will be useful?

Since it is quite practical and I could see the output working, it not only boosted my confidence but also helped me develop interest in hardware. And ofcourse as I read in the module how there are so many practical application of motors I feel it can be quite useful if I wish to choose a career pathway which deals with motors and similar hardware.

Youtube video link: https://youtu.be/4eIMIkqGDH8