

EXPERIMENT - 5

AIM → Write an Embedded C Program for Servo Motor interfaced with Arduino UNO.

SOFTWARE USED → Arduino IDE & Simulator.

PROGRAM →

// Embedded C Program for Servo motor interfaced with Arduino UNO.

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

```
Servo myservo;
```

```
int pos;
```

```
void setup()
```

```
{
```

```
  myservo.attach(10);
```

```
}
```

```
void loop()
```

```
{
```

```
  // loop for clockwise movement of Servo motor
```

```
  for (pos = 0; pos <= 180; pos++)
```

```
  {
```

```
    myservo.write(pos);
```

```
    delay(10);
```

```
  }
```

```
  // loop for anti-clockwise movement of Servo motor
```

```
  for (pos = 180; pos >= 0; pos--)
```

```
  {
```

```
    myservo.write(pos);
```

```
    delay(10);
```

```
  }
```

```
}
```


EXPERIMENT - 6

AIM → Write an Embedded C Program interfacing LCD with Pulse Width Modulation using Arduino UNO.

SOFTWARE USED → Arduino IDE & Simulator.

PROGRAM →

// Embedded C Program to interface LCD with Pulse Width Modulation Using Arduino UNO.

void Setup()

{

pinMode(10, OUTPUT);

}

void loop()

{

// duty cycle = 25%

analogWrite(10, 64);

delay(2000);

// duty cycle = 50%

analogWrite(10, 127);

delay(4000);

// duty cycle = 75%

analogWrite(10, 191);

delay(7000);

// duty cycle = 100%

analogWrite(10, 255);

delay(5000);

// duty cycle = 0%

analogWrite(10, 0);

delay(4000);

}

EXPERIMENT - 7

AIM → Write an Embedded C Program
Interfacing LCD with PIR Motion Sensor
Using Arduino UNO.

SOFTWARE USED → Arduino IDE & Simulator

PROGRAM →

// Embedded C Program to interface LCD
with PIR Motion Sensor Using Arduino
UNO.

```
int ledPin = 13; // choose the pin for LED
int inputPin = 2; // input pin for PIR sensor
int pinState = LOW; // assuming, no motion
                    // is detected
```

```
int Val = 0; // variable for reading the
              // pin status
```

void Setup ()

```
{
  pinMode (ledPin, OUTPUT); // declare LED
                             // as output
  pinMode (inputPin, INPUT); // declare
                              // sensor as input
  Serial.begin (9600);
}
```

void loop ()

```
{
  Val = digitalRead (inputPin); // Read
                                  // Input Value
}
```



```
// Check if input is HIGH  
if (val == HIGH)
```

```
{  
  digitalWrite (ledPin, HIGH);
```

```
  // turn LED ON  
  if (pinState == LOW)
```

```
{  
  // turned on LED
```

```
  Serial.println("motion detected!");
```

```
  // printing the output change  
  and not output state
```

```
  pinState = HIGH;
```

```
}
```

```
}
```

```
else
```

```
{
```

```
  // turn LED OFF
```

```
  digitalWrite (ledPin, LOW);
```

```
  if (pinState == HIGH)
```

```
{
```

```
  // turned OFF
```

```
  Serial.println("motion ended!");
```

```
  // printing the output change  
  and not output state
```

```
  pinState = LOW;
```

```
}
```

```
}
```

```
}
```

EXPERIMENT - 8

AIM →

Write an Embedded C Program interfacing Stepper motor Using Arduino UNO.

SOFTWARE USED →

Arduino IDE & Simulator.

PROGRAM →

// Embedded C Program to interface Stepper motor Using Arduino UNO

```
#include < stepper.h >
```

```
const int stepsPerRevolution = 200;  
// change this to fit the number of  
// steps per revolution
```

```
// initialize the stepper library  
// on pins 8 through 11.  
Stepper myStepper(stepsPerRevolution,  
8, 9, 10, 11);
```

```
void setup()
```

```
{  
  // set the speed at 60 rpm  
  myStepper.setSpeed(60);  
  // initialize the serial port  
  Serial.begin(9600);  
}
```


void loop ()

{

// step one Revolution in one direction
Serial.println ("Clockwise");
myStepper.step(400);
delay(500);

// step one Revolution in other
direction
Serial.println ("Counter Clockwise");
myStepper.step(400);
delay(500);

}

RESULT →

The Embedded C Program interfacing Stepper Motor Using Arduino Uno is written and simulated successfully.