

## **Practical – 3**

**Aim:** What's PWM? Interfacing & Programming for fading LED & RGB LED & DC motor RPM control with Arduino UNO.

### **What's Pulse width modulation? How it works? It's use?**

Pulse Width Modulation(PWM) is a digital technology that uses the amount of power delivered to a device that can be changed. It generates analogue signals by using a digital source. A PWM signal is basically a square wave which is switched between on and off state. The duty cycle and frequency of a PWM signal determine its behaviour.

The duty cycle of the PWM signal refers to the ratio of the time that the signal is in a high(on) state over the total time it takes to complete one cycle. It is commonly expressed as a percentage or a ratio.

A 50% duty cycle means that the high state takes half of the time and the low state takes the other half of the time, this is the same as an ideal square wave. If this ratio is greater than 50%, the logic high signal takes a longer time than logic low, vice versa. Thus, a 100% duty cycle means the signal is always on(full-scale), and the 0% duty cycle means the signal is always off(grounding).

- Drive buzzer with different loudness
- Control speed of the motor
- Control the direction of a servo
- Provide an analog output
- Generate audio signal
- Telecommunication: Encode message
- Adjust Brightness of Screen using PWM

### **1: Fade LED using Arduino**

**Code:-**

```
int led = 9;

int brightness = 0;

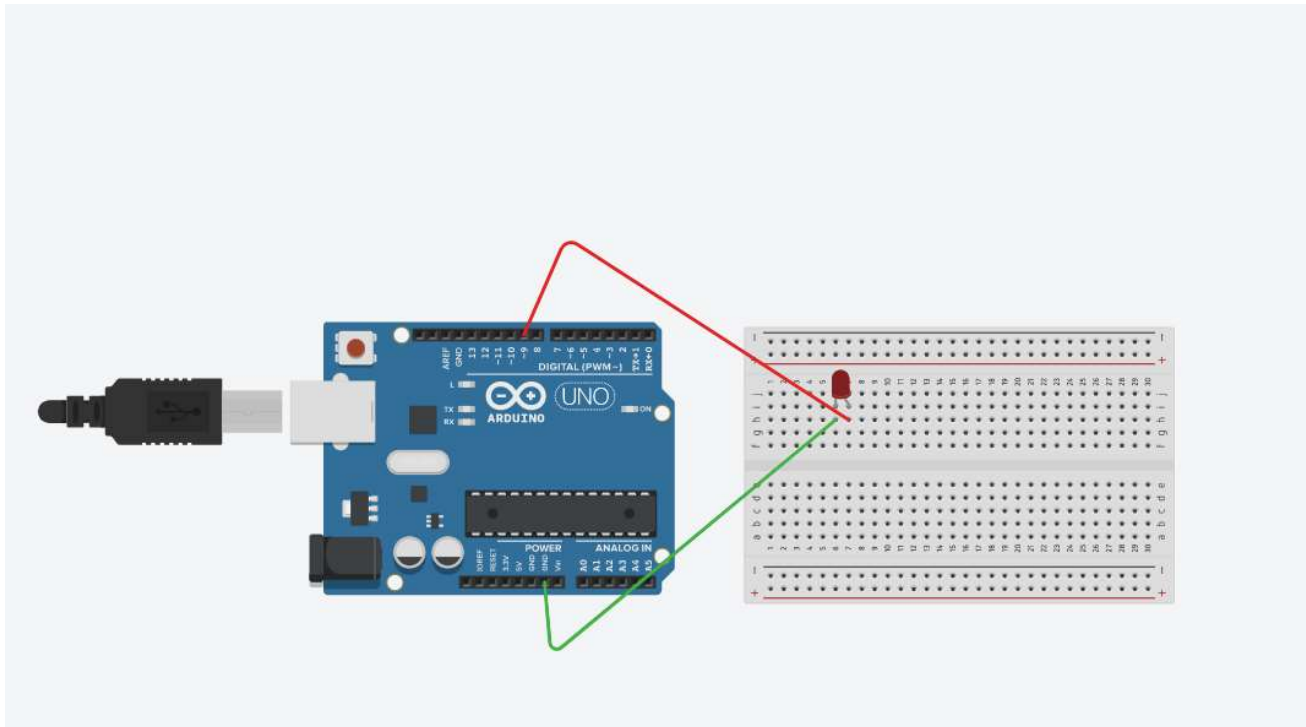
int fadeAmount = 5;


void setup()
{
    pinMode(led, OUTPUT);
}

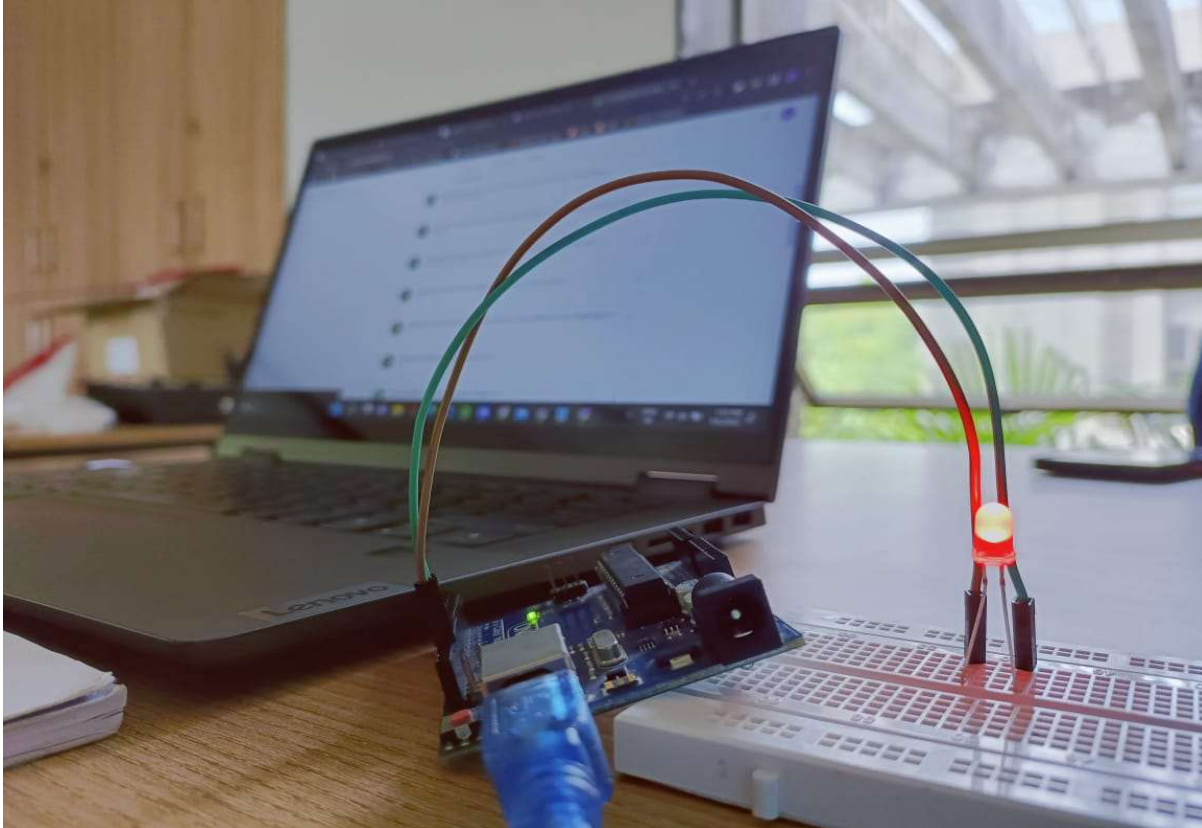
void loop()
```

```
{  
  analogWrite(led, brightness);  
  brightness = brightness + fadeAmount;  
  if (brightness == 0 || brightness == 255)  
  {  
    fadeAmount = -fadeAmount ;  
  }  
  delay(1000);  
}
```

### SET UP :-



### CIRCUIT OUTPUT:-



## **2: Fade RGB LED using Arduino UNO**

### **Code:-**

```
int rVal = 254;  
int gVal = 1;  
int bVal = 127;
```

```
int rDir = -1;  
int gDir = 1;  
int bDir = -1;
```

```
// constants to name the pins  
const int rPin = 13;  
const int gPin = 11;  
const int bPin = 12;
```

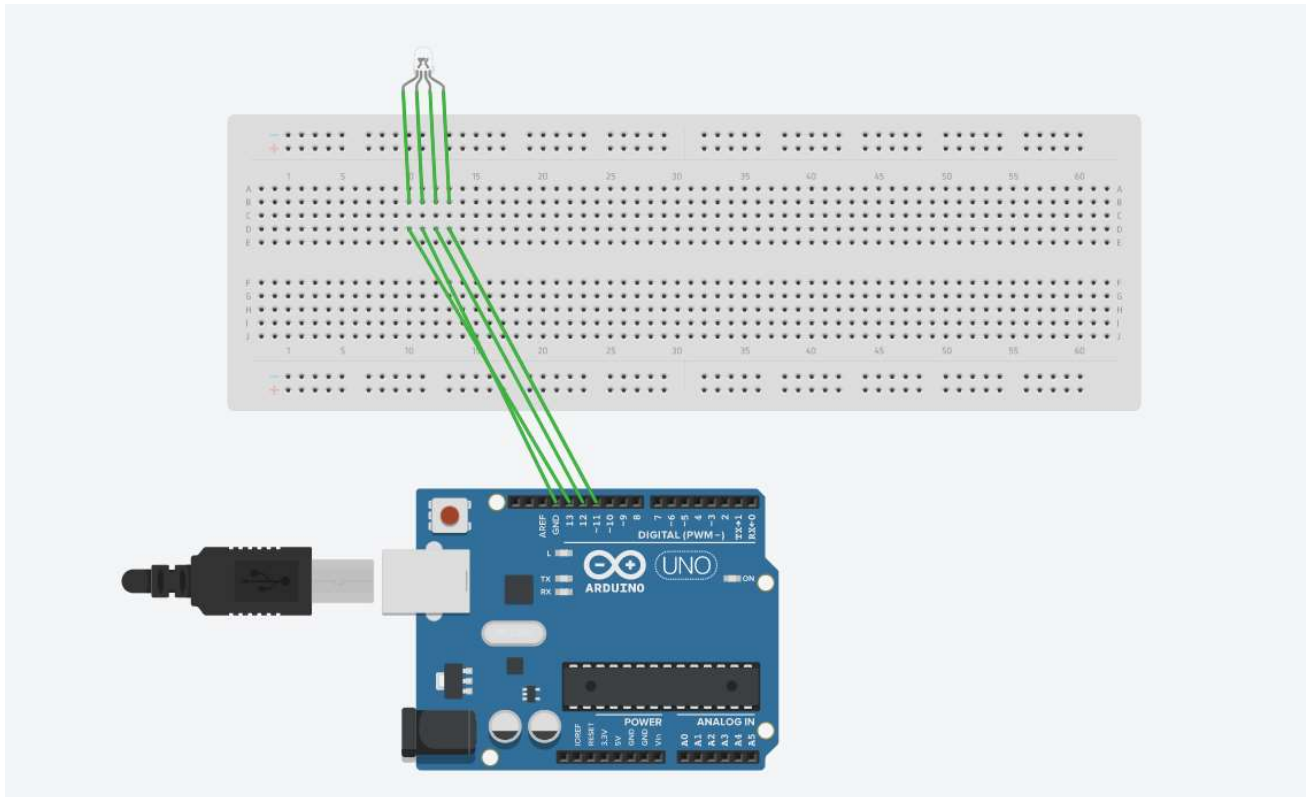
```
void setup() {  
    // declare the pinModes  
    pinMode(rPin, OUTPUT);  
    pinMode(gPin, OUTPUT);  
    pinMode(bPin, OUTPUT);  
}  
  
void loop()  
{  
    analogWrite(rPin, rVal);  
    analogWrite(gPin, gVal);  
    analogWrite(bPin, bVal);  
  
    // change the values of the LEDs  
    rVal = rVal + rDir;  
    gVal = gVal + gDir;  
    bVal = bVal + bDir;  
  
    // for each color, change direction if  
    // you reached 0 or 255  
    if (rVal >= 255 || rVal <= 0) {  
        rDir = rDir * -1;  
    }  
  
    if (gVal >= 255 || gVal <= 0) {  
        gDir = gDir * -1;  
    }  
  
    if (bVal >= 255 || bVal <= 0) {  
        bDir = bDir * -1;  
    }  
}
```

19SE02IT058

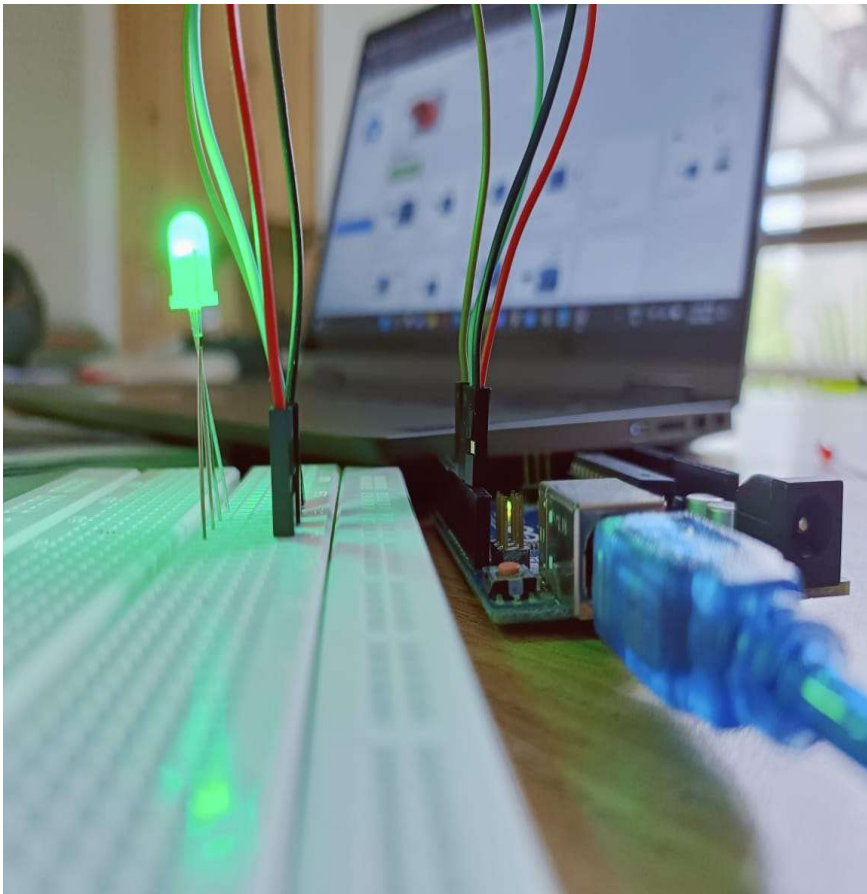
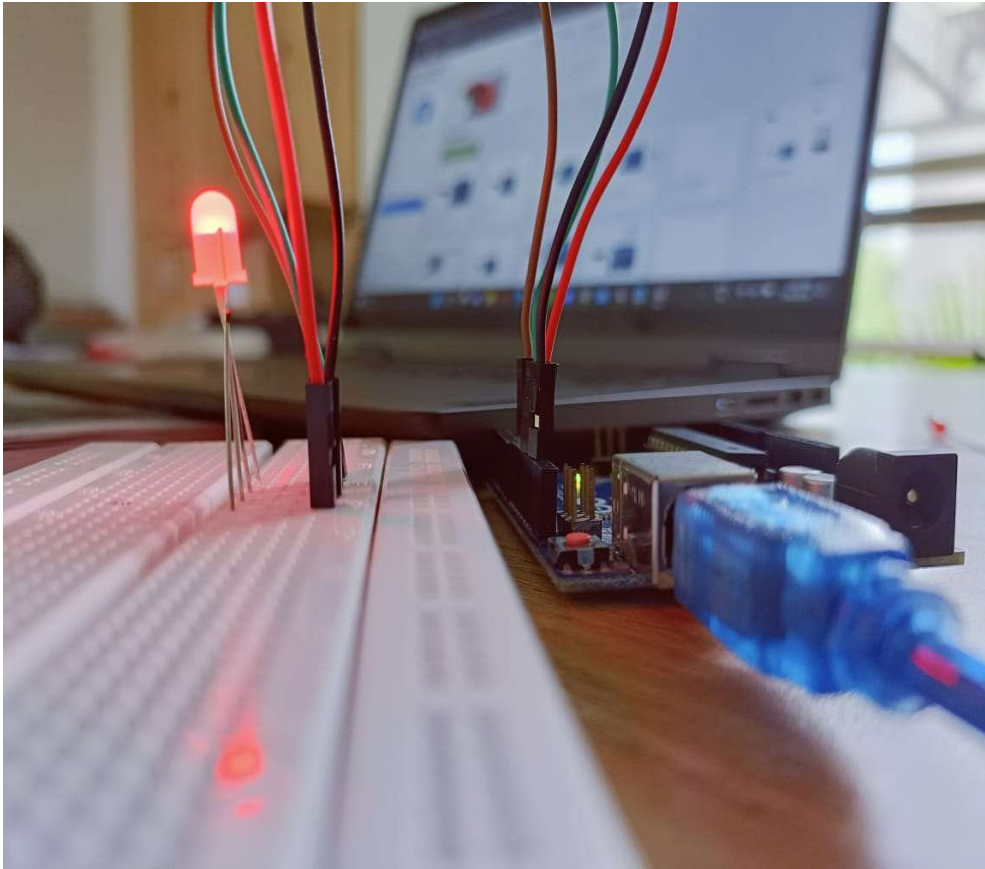
IOT

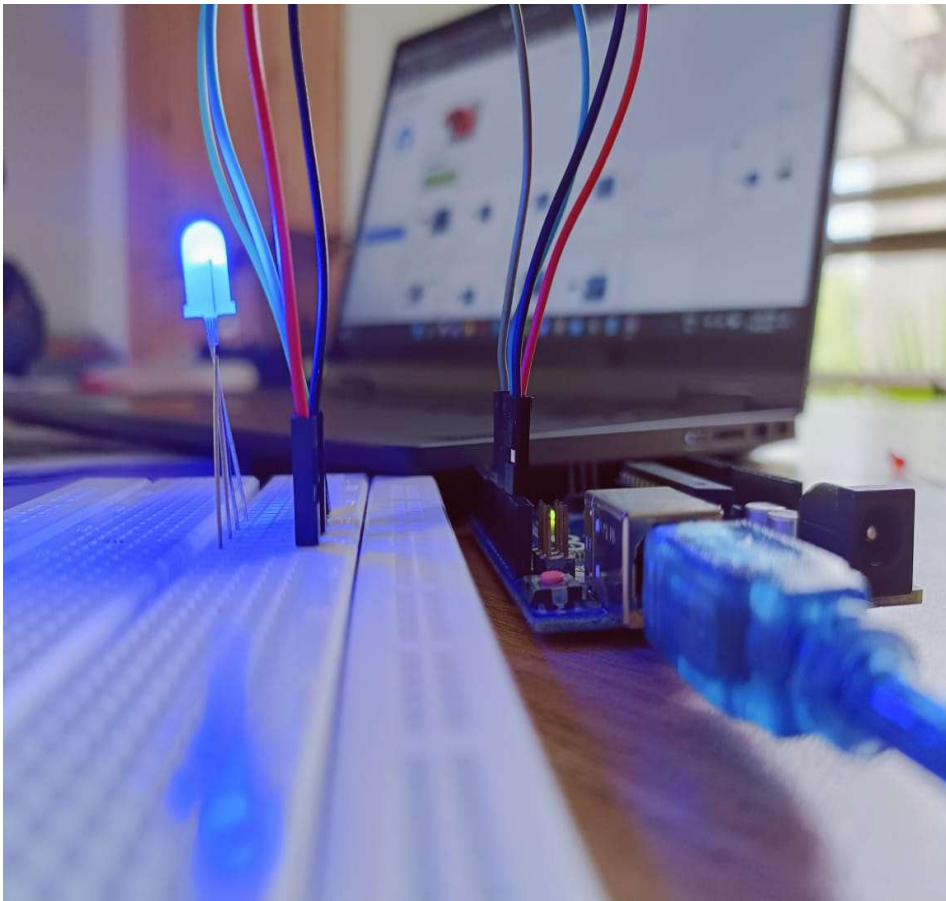
```
// slight delay so it doesn't rotate color too quickly  
delay(33);  
}
```

**SET UP :-**



**CIRCUIT OUTPUT :-**





### **3. DC motor RPM control**

Code:

```
int brightness = 0;

void setup()
{
  pinMode(9 , OUTPUT);
}

void loop()
{
  for (brightness = 0; brightness <= 255 ; brightness += 5 ){
    analogWrite ( 9 , brightness );
    delay(30);
  }
  for (brightness <= 255 ; brightness >= 0; brightness -= 5 ){
```



```
analogWrite ( 9 , brightness );  
delay(30);  
}  
}
```

**SET UP :-**