Led fading with polutiometer

```
int ledPin = 9;
void setup()
{
  Serial.begin(9600);
    pinMode (ledPin, OUTPUT);
}
  void loop()
  {
     int analogValue = analogRead (A0);
     int brightness = map (analogValue, 0, 1023, 0, 255);
     analogWrite (ledPin, brightness);
     Serial.print(" Analog. ");
     Serial.print (analogValue);
     Serial.print (", Brightness:  ");
     Serial.println (brightness);
      delay (100);
   }
```

Trishala K Shetty
1BM18CS118

Fading LED

Aim/Theory

Code:

```
void setup()
{
    pinMode (3, OUTPUT);
}

void loop()
{
    for(int fade=0; fade<=255; fade+=5)
    {
        analogWrite (9, fade);
        delay(30);
    }

    for (int fade=255 >= 0; fade-=5)
    {
        analogWrite (9, fade);
        delay(30);
    }
}
```

Vibrator motor

```
int motorPin = 3;
int SensorPin = A1;
int threshold = 400;
void setup()
{
    pinmode(motorPin, OUTPUT);
    Serial.begin(9600);
}
void loop()
{
    int SensorValue = analogRead(SensorPin);
    Serial.println(SensorValue);
    if(Sensorvalue > threshold)
    {
        digitalwrite(motorPin, HIGH);
    }
    else
    {
        digitalwrite(motorPin, Low);
    }
}
```

Push button:

```
int buttonState = 0;
const int buttonPin = 2;
const int ledPin = 13;
void setup ()
{
    pinMode (13, OUTPUT);
    pinMode (2, INPUT);
}
void loop ()
{
    buttonState = digitalRead (buttonPin);
    if (buttonState == HIGH)
    {
        digitalWrite (ledPin, HIGH);
    }
    else
    {
        digitalWrite (ledPin, low);
    }
}
```

Trishala .K. Shetter
IBM18CS118

LDR

```
Const int ledPin=13;
Const int ldrPin=A0;
void Setup()
{
    Serial.begin(9600);
    pinmode(ledPin, OUTPUT);
    pinmode(ldrPin, INPUT);
}

void loop()
{
    int ldrStatus = analogRead(ldrPin);
    Serial.println(ldrStatus);
    if(ldrStatus <=10)
    {
        digitalwrite(ledPin, HIGH);
        Serial.println("LDR is Dark, LED is ON");
    }
    else
    {
        digitalwrite(ledPin, Low);
        Serial.println("LDR is BRIGHT, LFD is OFF");
        Serial.println("-----");
    }
}
```

Jushela R. Shette
13MMESUP

PIR:

```
int ledPin=13;
int inputPin=2;
int PIRState = LOW;
int val=0
void setup()
{
  pinmode (ledPin, OUTPUT);
  pinmode (inputPin, INPUT);
  Serial.begin (9600);
}
void loop()
{
  val= digitalRead (inputPin);
  if (val == HIGH)
    digitalwrite (ledPin, HIGH);
    if (PIRState == low)
    {
      Serial.println (" Sensor Activates");
      PIRState == HIGH;
    }
  }
  else
  {
```

Trishala K Shetter
1BM18CS118

```
digitalwrite (ledPin, LOW);
if (PIRState == HIGH)
{
    Serial.println("motion ended!");
    PIRState = LOW;
}
}
}
```

Alert System:

```
int temperaturePin = 0;
int buzzer = 12;
void setup()
{
    Serial.begin(9600);
    pinmode(buzzer, OUTPUT);
    pinmode(9, OUTPUT);
}
float get voltage (int pin)
{
    return (analogRead(Pin) + 0.0048828147);
}
void loop()
{
    float voltag, degrees c;
    voltag = get voltage (temperature Pin);
    degreesc = (voltag - 0.5)*100.0;
    digital-write (9, LOW);
    if (degreesc > 37)

    Serial.print (degreesc);
    Serial.println ("ALERT!");
    digitalwrite (buzzer, LOW);
```

```
digitalWrite (9, HIGH);
tone (12, 1000B, 100);
delay (200);
}
else {
    Serial.Print ("degree c);
    Serial.Print ("SAVE!");
    delay(200);
    }
}
```

Object distance:

```
int trigpin=4;
int echopin=2;
long distance, cm, inches;
void setup()
{
    Serial.begin(9600);
    pinmode(trigpin, OUTPUT);
    pinmode(echopin, INPUT);
}

void loop()
{
    digital write (trigpin, low);
    delay microseconds(5);
    digital write (trigpin, HIGH);
    delaymicroseconds(10);
    digitalwrite (trigpin, low);
    duration = pulse In (echopin, HIGH)
    cm = (duration /2)/29.1;
    inches = (duration/2)/74;
    Serial.print(" Distance: ");
    Serial.print(inches);
    Serial.print(cm);
    Serial.print(" : cms");
    delay(300);
}
```

Gas Leakage:

```
int led=13;
const int gas=0;
int gaspin=A0;
void setup()
{
  Serial. begin(9600);
}

void loop()
{
  float sensorvalue= analogRead(gaspin);
  if(sensorvalues >= 300)
  {
    digital write (led, HIGH);
    Serial.print (sensorvalue);
    Serial.println (" *** Smoke DETECTED*** !");
    delay(sensor value);
  }
  else
  {
    digitalwrite (led, LOW)
    Serial.println ("Serial value : ");
    Serial.println(sensorvalue);
  }
  delay(1000)
}
```

IR Servo motor

```
#include <Servo.h>
#include <IRremote.h>
  int RECV_PIN=11
    IRecv, irrecv (RECV_PIN);
    decode_results results;
   Servo myservo;
 void setup()
    {
   Serial.begin (9600);
    irrecv. enable IRIN();
    }
  void loop()
    {
    if (irrecv. decode (&result )){
    Switch (results.value)
    {
     Case OxFD00FF:
         myservo.attach (9)-
         Serial.println (" Start... ");
         break;
     Case 0xFD20DF:
         myservo. write (-360);
         Serial.println(" Counter Clockwise");
         break;
```

```
· default:
Serial.print(" unrecognized Code received = ox")
Serial.print(n(results.value, HEX);
   break;
}
irrecv-resume();
}
}
```

Smart Packaging System

```
int tilt = 2;
int led = 13;
void Serup()
{
    pin mode (tilt, INPUT);
    pin mode (led, OUTPUT);
}
void loop()
{
    int Reading;
    Reading = digital Read( tilt);
    if (Reading)
    digital write (led, Low);
    else
    digital write (led, HIGH);
}
```

RGB

```
int red = 10;
int green = 8;
int blue = 9;
void setup()
{
    pinmode(10, OUTPUT);
    pinmode(9, OUTPUT);
    pinmode(8, OUTPUT);
}
void loop()
{
    lcd.set cursor(0,0);
    lcd.print("RGB Color print!");
    delay(1000);
    lcd.clear();
    RGB color(255,0,0);
    lcd.print("Red");
    delay(1000);
    lcd.clear();

    RGB color(0,255,0);
    lcd.print("Green");
    delay(1000);
    lcd.clear();
```

```
    RGB color(0,0,255);
    lcd.print("BLUE");
    delay(1000);
    lcd.clear();

    RGB color(0,0,0);
    lcd.print("white");
    delay(1000);
    lcd.clear();
}
void RGB color(int red value, int green value, int
                                     blue value)
{
    analogwrite(red, red value);
    analogwrite(green, green value);
    analogwrite(blue, blue value);
```

Smart irrigation:

```
#include < servo.h>
servo myservo;

int pos=0;
int sensor Pin=A0;
int sensorvalue=0;
void setup()
{
   myservo. attach(3);
   Serial. begin(9600);
}
void loop()
{
   sensor value=analogread(sensorPin)
      serial.print(sensorvalue)
   if (sensor value>500)
   {
   for (pos=0; pos<=180; pos+=1)
   {
   myservo. write(pos);
      delay(15);
   }
   for (pos=180; pos>=0; pos-=1)
```

```
   }
   myservo. write(pos);
      delay(15);
   }
   }
   delay(1000);
}
```