

## EXPERIMENT-12

**Aim:** To develop a Smart bulb using IOT.

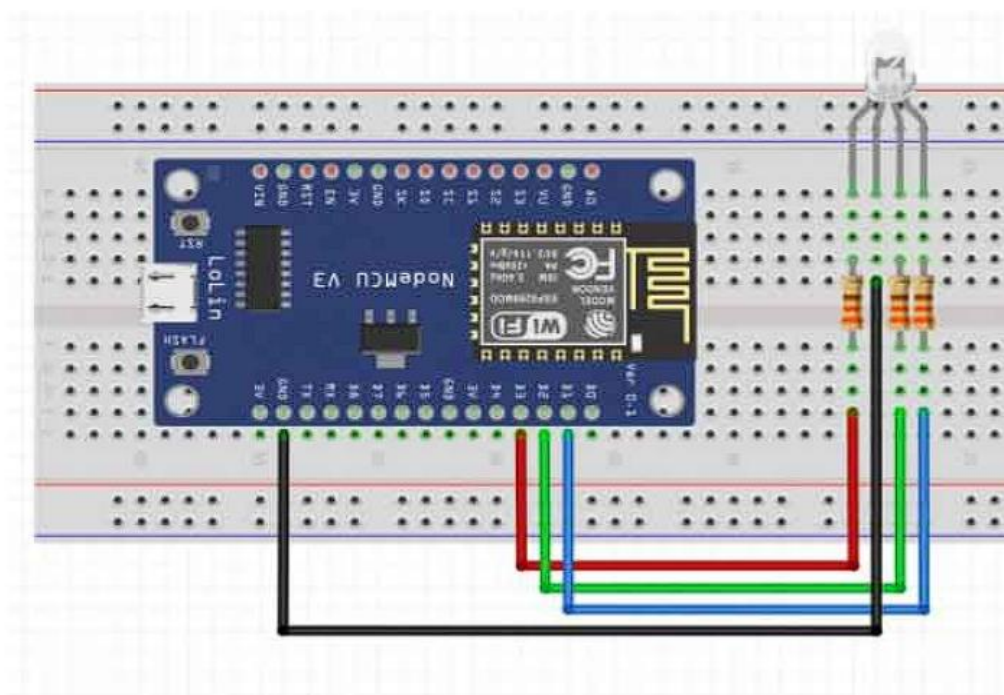
**Apparatus:**

- ESP8266 (NodeMCU) ESP12E Development Board
- Breadboard
- Micro USB Cable
- RGB LED
- Jumper wires

**Theory:**

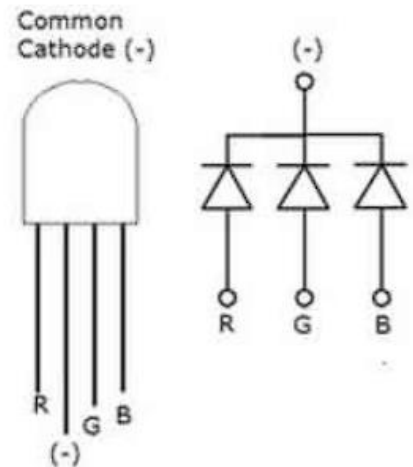
The Smart Bulb is a versatile and exciting project that combines the power of the ESP8266 microcontroller with the vibrant world of RGB LEDs. This project allows you to create a web-based interface to control RGB LED LIGHT, making it an ideal choice for lighting enthusiasts, DIY hobbyists, and home automation enthusiasts. We are sending a “GET” request from our custom webpage that will have a color picker to send **red**, **green**, and **blue** values. The **ESP8266 NodeMCU ESP-12E** receives data and sets the **color of the RGB LED** based on the color picker **values**.

**Circuit Diagram:**



## Wiring NodeMCU and RGB LED

ESP12E NodeMCU module	RGB LED
D3	R
GND	GND
D2	G
D1	B



- First, connect the **R Pin** to the **D3** pin on your ESP8266 via a  $220\Omega$  resistor.
- Second, connect the **G Pin** through the  $220\Omega$  resistor to the **D2** pin on your ESP8266 NodeMCU.
- Third, connect the **B Pin** through the  $220\Omega$  resistor to the **D1** pin on your ESP8266.
- Finally, (-) Connect directly (no resistors) to the **GND** pin on your ESP board.

### Code:

```
#include <ESP8266WiFi.h>
#include <DNSServer.h>
#include <ESP8266WebServer.h>

const char *ssid = "S23 ultra";
const char *password = "zxcvbnm.";
const byte DNS_PORT = 53;
const int redLED = 0;    //D3
const int greenLED = 4;  //D2
const int blueLED = 5;   //D1
IPAddress apIP(192, 168, 1, 5);
DNSServer dnsServer;
ESP8266WebServer webServer(80);

String webpage = ""
"<!DOCTYPE html><html><head><title>RGB control</title><meta name='mobile-web-
app-capable' content='yes' />"
"<meta name='viewport' content='width=device-width' /></head><body
style='margin: 0px; padding: 0px;'>"
"<canvas id='colorspace'></canvas></body>"
```

```

"<script type='text/javascript'"
"(function () {"
" var canvas = document.getElementById('colorspace');"
" var ctx = canvas.getContext('2d');"
" function drawCanvas() {"
" var colours = ctx.createLinearGradient(0, 0, window.innerWidth, 0);"
" for(var i=0; i <= 360; i+=10) {"
" colours.addColorStop(i/360, 'hsl(' + i + ', 100%, 50%)');"
" }"
" ctx.fillStyle = colours;"
" ctx.fillRect(0, 0, window.innerWidth, window.innerHeight);"
" var luminance = ctx.createLinearGradient(0, 0, 0, ctx.canvas.height);"
" luminance.addColorStop(0, 'ffffff');"
" luminance.addColorStop(0.05, 'ffffff');"
" luminance.addColorStop(0.5, 'rgba(0,0,0,0)');"
" luminance.addColorStop(0.95, '#000000');"
" luminance.addColorStop(1, '#000000');"
" ctx.fillStyle = luminance;"
" ctx.fillRect(0, 0, ctx.canvas.width, ctx.canvas.height);"
" }"
" var eventLocked = false;"
" function handleEvent(clientX, clientY) {"
" if(eventLocked) {"
" return;"
" }"
" function colourCorrect(v) {"
" return Math.round(1023-(v*v)/64);"
" }"
" var data = ctx.getImageData(clientX, clientY, 1, 1).data;"
" var params = ["
" 'r=' + colourCorrect(data[0]),"
" 'g=' + colourCorrect(data[1]),"
" 'b=' + colourCorrect(data[2])"
" ].join('&');"
" var req = new XMLHttpRequest();"
" req.open('POST', '?' + params, true);"
" req.send();"
" eventLocked = true;"
" req.onreadystatechange = function() {"
" if(req.readyState == 4) {"
" eventLocked = false;"
" }"
" }"
" }"
" canvas.addEventListener('click', function(event) {"
" handleEvent(event.clientX, event.clientY, true);"
" }, false);"
" canvas.addEventListener('touchmove', function(event) {"
" handleEvent(event.touches[0].clientX, event.touches[0].clientY);"
" }, false);"
" function resizeCanvas() {"
" canvas.width = window.innerWidth;"
" canvas.height = window.innerHeight;"
" drawCanvas();"
" }"

```

```
" window.addEventListener('resize', resizeCanvas, false);"
" resizeCanvas();"
" drawCanvas();"
" document.ontouchmove = function(e) {e.preventDefault()};"
" })();"
"</script></html>";

void handleRoot() {
// Serial.println("handle root..");
String red = webServer.arg(0); // read RGB arguments
String green = webServer.arg(1);
String blue = webServer.arg(2);

if((red != "") && (green != "") && (blue != ""))
{
    analogWrite(redLED, 1023 - red.toInt());
    analogWrite(greenLED, 1023 - green.toInt());
    analogWrite(blueLED, 1023 - blue.toInt());
}

Serial.print("Red: ");
Serial.println(red.toInt());
Serial.print("Green: ");
Serial.println(green.toInt());
Serial.print("Blue: ");
Serial.println(blue.toInt());
Serial.println();

webServer.send(200, "text/html", webpage);
}

void setup() {

pinMode(redLED, OUTPUT);
pinMode(greenLED, OUTPUT);
pinMode(blueLED, OUTPUT);

analogWrite(redLED, 0);
analogWrite(greenLED, 0);
analogWrite(blueLED, 0);

delay(1000);
Serial.begin(115200);
Serial.println();

WiFi.mode(WIFI_AP);
WiFi.softAPConfig(apIP, apIP, IPAddress(255, 255, 255, 0));
WiFi.softAP(ssid);

// if DNSServer is started with "*" for domain name, it will reply with
provided IP to all DNS request
dnsServer.start(DNS_PORT, "rgb", apIP);

webServer.on("/", handleRoot);
```

```
webServer.begin();

testRGB();
}

void loop() {

  dnsServer.processNextRequest();
  webServer.handleClient();

}

void testRGB() { // fade in and out of Red, Green, Blue

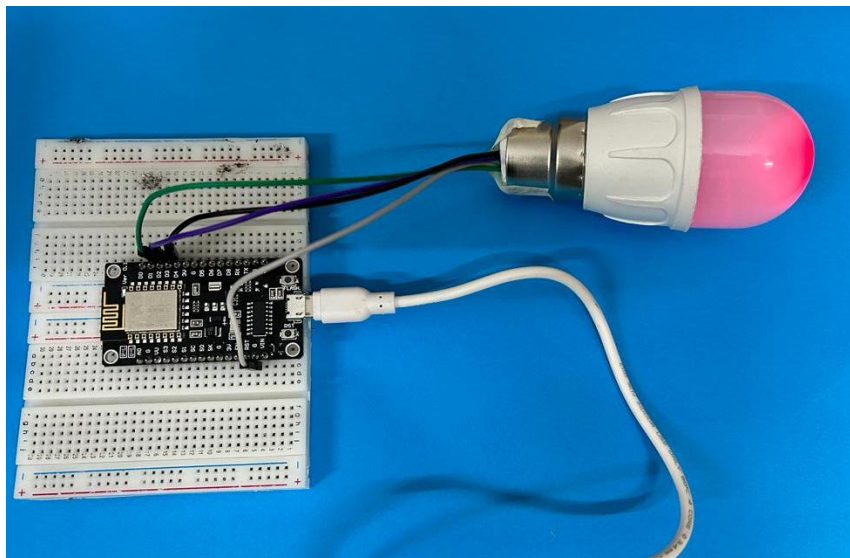
  analogWrite(redLED, 0); // R off
  analogWrite(greenLED, 0); // G off
  analogWrite(blueLED, 0); // B off

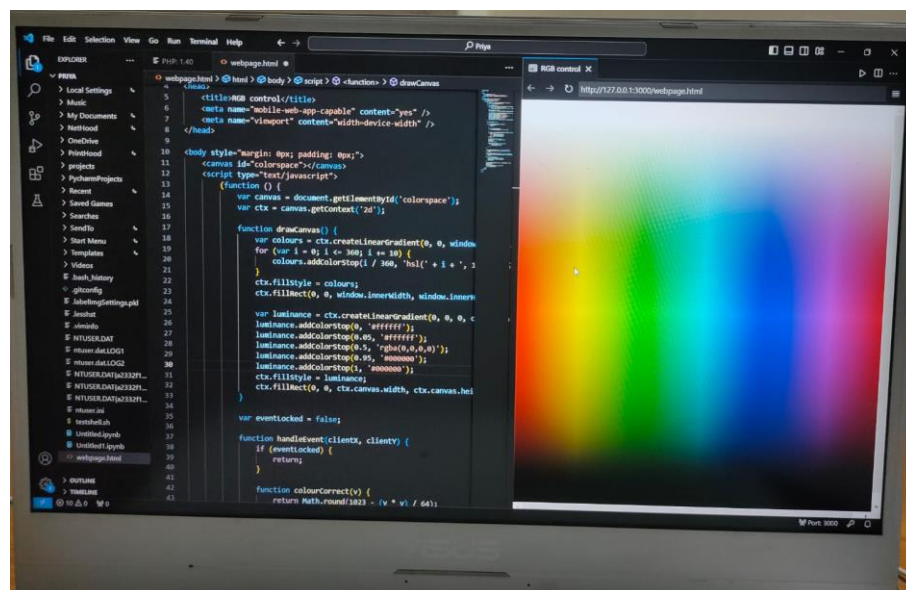
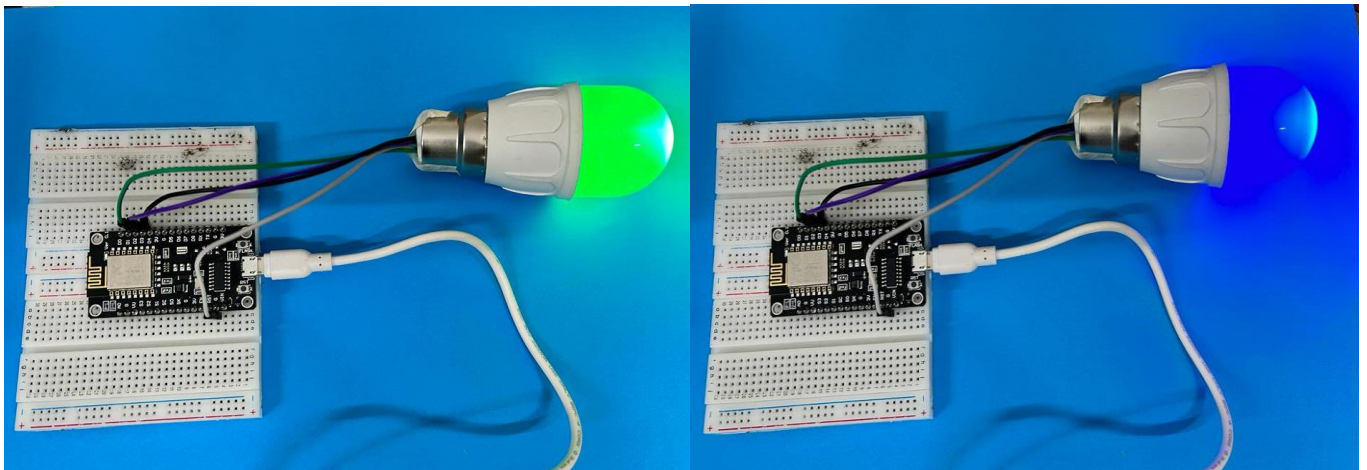
  fade(redLED); // R
  fade(greenLED); // G
  fade(blueLED); // B
}

void fade(int pin) {

  for (int u = 0; u < 1024; u++) {
    analogWrite(pin, u);
    delay(1);
  }
  for (int u = 0; u < 1024; u++) {
    analogWrite(pin, 1023 - u);
    delay(1);
  }
}
```

## Output:





**Conclusion:** In this experiment, we learned how to create Smart Bulb with the help of ESP8266 (NodeMCU) ESP12E Development board.