Topic	RGB COLORS	
Class Description	**Student will be introduced to the concept of RGB LED and fading of LEDs using Pulse width Modulation and Resolution techniques**	
Class	**PRO C246**	
Class time	**50 mins**	
Goal	• Introduction to the RGB LED • Pulse width Modulation • Resolution • Fading of LED	
Resources Required	• Teacher Resources: ○ Laptop with internet connectivity ○ Earphones with mic ○ Notebook and pen ○ Smartphone • Student Resources: ○ Laptop with internet connectivity ○ Earphones with mic ○ Notebook and pen	
Class structure	**Warm-Up** **Student-Led Activity -1** **Student-Led Activity -2** **Wrap-Up**	**10 mins** **15 mins** **15 mins** **10 mins**
Credit & Permissions:	**Code samples used for Firebase-Google Authentication are licensed under the Apache 2.0 License.** **Expo documentation used from - https://expo.io** **Note: Keep this row section only if applicable**	

WARM-UP SESSION - 10 mins		
Teacher Action		**Student Action**
Hey <student's name>. How are you? It's great to see you! Are you excited to learn something new today? **Following are the WARM-UP session deliverables:** • Greet the student. • Revision of previous class activities. • Quizzes.		**ESR**: Hi, thanks! Yes, I am excited about it! Click on the slide show tab and present the slides
WARM-UP QUIZ Click on In-Class Quiz		
Activity Details **Following are the session deliverables:** • Appreciate the student. • Narrate the story by using hand gestures and voice modulation methods to bring in more interest in students.		
STUDENT-LED ACTIVITY-1 - 15mins		
Student Initiates Screen Share		
ACTIVITY • **Introduction to RGB** • **Pulse Width Modulation** • **Resolution**		
Teacher Action		**Student Action**

This class is a student-driven class, a teacher will guide the students to complete the activity	
Can you tell me what LED is?	**ESR:** Light Emitting diode
Great!	
What types of colors can we see in LEDs?	**ESR:** Red, green, yellow, white
Right!	
So can I say blue LED produces blue color, green LED produces green color, white LED produces white color	
What if I want to see all colors in one LED?	
Is it possible?	**ESR:** Varied!
Yes! That's possible.	
For that, we need to use a different type of LED called **RGB LED**	
RGB LED means red, blue, and green LEDs. RGB LED products combine these three colors to produce over 16 million hues of light except brown and pink.	
How is light produced in an LED?	
Light-emitting diodes produce light by the movement of electrons between the two terminals of a diode, which	

occur by a process called electroluminescence. All these materials are made from semiconductors.

RGB LED:

RGB LEDs have four pins—one for each LED(RED, Green, Blue) and another for the common anode or cathode.

- **R (red) pin:** is to control the red color element
- **G (green) pin:** is to control the green color element
- **B (blue) pin** is to control the blue color elements the,
- **Common Anode and Common Cathode RGB LEDs**
- In a common cathode RGB LED, all three pins share a single **negative connection (cathode).**
- In a common anode RGB LED, the three pins share a common **positive connection (anode)**

Note: We are using common cathode RGB LED, so will connect with this GND

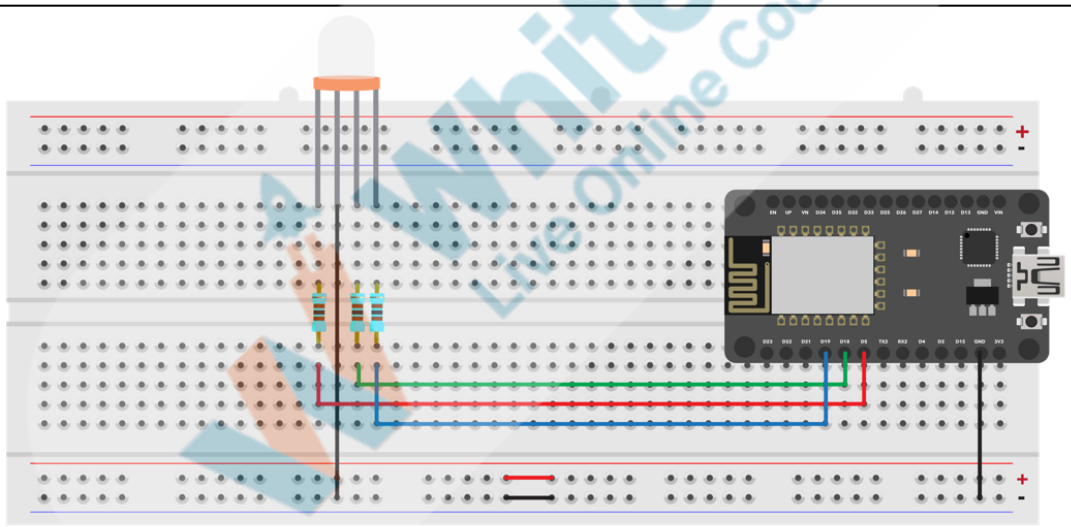
Step -1:Gather the material from the IoT kit:

- 1 x ESP32
- 1 x USB Cable
- 1 x Breadboard
- 4 x Jumper wires
- 1 x RGB LED

- 3 x 330 Ohm Resistors

Step -2: Let's do connections:

- Insert RGB LED into a breadboard, insert four leg's into breadboard
- The three positive leads of the LEDs (one red, one green, and one blue) are each connected to the 330-ohm resistors.
- The other end of the resistor will go to the ESP32 GPIO pins, **D5, D18, D19** respectively.
- The common negative connection of the RGB LEDwhich is the second pin from the flat side of the LED package. It is also the longest of the four leads. This lead will be connected to **ESP 32 GND** pin



Step-3 Let's write a code:
specify to which GPIO or GPIOs the signal will appear upon.

- define GPIO signal pins: **LEDR** pins **5, 18, 19**
- define **R_channel, G_channel, B_channel 0,1,2**
- define **pwm_Frequency 5000,** PWM stands for Pulse Width Modulation which is used to check portion of the time the signal spends on versus the time that the signal spends off. PWM on-off pattern can simulate voltages in between the full Vcc of the board (e.g., 5 V/3.3 V on a and off (0 Volts) The duration of "on time" is called the pulse width.
- For an LED, PWM frequency is 5000 Hz
- defined as **pwm_resolution 8,**8-bit resolution, which means we can control the LED brightness using a value from 0 to 255**.**

```
#define LEDR 5
#define LEDG 18
#define LEDB 19

#define R_channel 0
#define G_channel 1
#define B_channel 2

#define pwm_Frequency 5000 // pwm frequency
#define pwm_resolution 8 // 8 bit resolution
```

Initialize using **void setup()** function

- **ledcAttachPin** function accepts two arguments. The first is the GPIO that will output the signal, and the second is the channel that will generate the signal.

- Set **channel** for all three colors and their, **pwm frequency** along with **pwm_resolution.**

```
void setup() {

  ledcAttachPin(LEDR, R_channel);
  ledcAttachPin(LEDG, G_channel);
  ledcAttachPin(LEDB, B_channel);

  ledcSetup(R_channel, pwm_Frequency, pwm_resolution);
  ledcSetup(G_channel, pwm_Frequency, pwm_resolution);
  ledcSetup(B_channel, pwm_Frequency, pwm_resolution);
}
```

To execute the main process write the **void loop()**
RGB (Red, Green, Blue) all colors have 8 bits each. The range for each individual color is 0-255 The combination range is **256***256*256.To display different colours we need to show variation between o to 255

- RGB_Color for RED (255,0,0)
- RGB_Color for Green(0.255,0)
- RGB_Color for Blue (0,0,255)
- RGB_Color for Yellow (255,255,0)
- RGB_Color for Cyan (0, 255,255)
- RGB_Color for magenta(255,0,255)
- RGB_Color for pink(255,0,147)

```
void loop() {
  RGB_Color(255, 0, 0); // RED ccolor
  delay(500);
  RGB_Color(0, 255, 0); // green color
  delay(500);
  RGB_Color(0, 0, 255); // blue color
  delay(500);
  RGB_Color(255, 255, 0); // yellow color
  delay(500);
  RGB_Color(0, 255, 255); // cyan color
  delay(500);
  RGB_Color(255, 0, 255); // magenta color
  delay(500);
  RGB_Color(255, 20, 147); // deep pink color
}
```

Make a loop pattern for all described colors using loops.

- ledCWrite function is used to control the LED brightness using PWM for **R_channel, G_channel, B_channel**

```
void RGB_Color(int i, int j, int k){
  ledcWrite(R_channel, i);
  ledcWrite(G_channel, j);
  ledcWrite(B_channel, k);
}
```

Output:
Compile and upload the program to ESP32 board using Arduino IDE

- Verify the program by clicking the Tick option
- Upload the program by clicking the arrow option

Note: If the port is not selected, insert the USB cable in Computer's port and select the port

- You will see different colors on the LED

Student Stops Screen Share

So its time to do the second Activity, now you are comfortable with LEDs
Please share your screen with me.

We have one more class challenge for you.
Can you solve it?

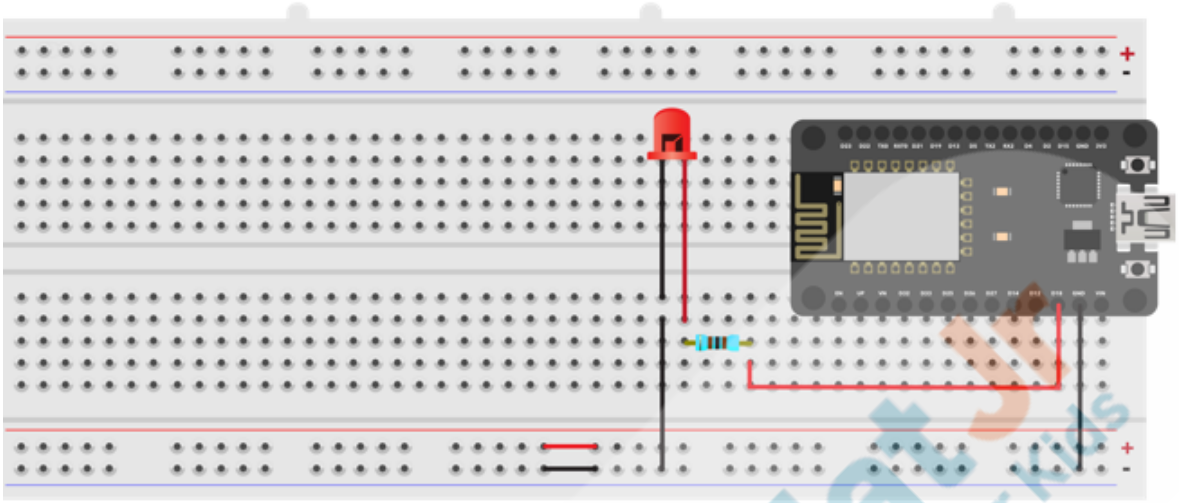
Let's try. I will guide you through it.

STUDENT-LED ACTIVITY-2 - 15 mins

- Ask the student to press the ESC key to come back to the panel.
- Guide the student to start Screen Share.
- The teacher gets into Full Screen.

Student Initiates Screen Share

ACTIVITY • Fading LED	
Teacher Action	Student Action
The next task is to Fade the LED, Now the main task is to write the program and control the intensity and brightness of a LED Let's start	!
Step -1:Gather the material from the IoT kit: • 1 x ESP32 • 1 x USB Cable • 1 x Breadboard • 4 x Jumper wires • 1 x LED • 1 x Resistor	
Step -2: Let's do connections: • Supply positive(VCC (+ve)) from the ESP 32 to breadboard terminal • Supply negative(GND (-ve)) from the ESP 32 to breadboard negative terminal • Insert the Led into the breadboard • Connect the longer leg of the LED to one end of the resistor as shown below: • Connect another end of the resistor with the ESP32 PIN 32 of the Breadboard • Connect shorter Leg of the breadboard with the GND	



Step-3 Now let's write a program:

Define Pins

- define **GPIO pin for LED, led_gpio =32**
- define **brightness =0**
- **define fadeAmount =5**

```
const byte led_gpio = 32;
int brightness = 0;
int fadeAmount = 5;
```

Initialize the setup()

- **ledcAttachPin** function accepts two arguments. The first is the GPIO that will output the signal, and the second is the channel that will generate the signal.

- Set Channel i.e=0, frequency. i.e. 4000 and resolution i.e 8

```
void setup() {
  ledcAttachPin(led_gpio, 0); // assign a led pins to a channel


  ledcSetup(0, 4000, 8); // 12 kHz PWM, 8-bit resolution
}
```

To execute the main process write the **void loop()**
We need to control the LED brightness in this main function

- **ledCWrite** function is used to control the LED brightness, it will take two arguments channels and brightness
- Declare the variable brightness and change the brightness level using brightness and fade Amount
- Write the condition using if condition, if brightness is less than 0 and more than 255, then decrease the **fadeAmount** value.
- Set up **delay** for **30 ms**

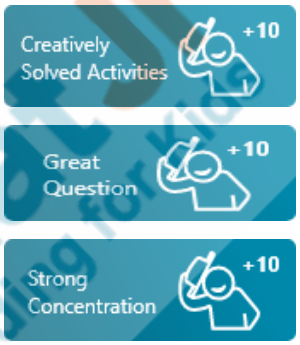
```
void loop() {
  ledcWrite(0, brightness); // set the brightness of the LED

  // change the brightness for next time through the loop:
  brightness = brightness + fadeAmount;

  // reverse the direction of the fading at the ends of the fade:
  if (brightness <= 0 || brightness >= 255) {
    fadeAmount = -fadeAmount;
  }
  // wait for 30 milliseconds to see the dimming effect
  delay(30);
}
```

Output:
Compile and upload the program to ESP32 board using Arduino IDE

- Verify the program by clicking the Tick option

• Upload the program by clicking the arrow option *Note: If the port is not selected, insert the USB cable in Computer's port and select the port*	
Go to Tools and select **Serial Monitor** • See the **brightness of the LED and Value**	
So, today we learned about RGB LED and how to fade an LED. That's fun!	
Teacher Guides Student to Stop Screen Share	
WRAP-UP SESSION - 05 mins	
Activity details **Following are the WRAP-UP session deliverables:** • Appreciate the student. • Revise the current class activities. • Discuss the quizzes.	
WRAP-UP QUIZ Click on In-Class Quiz	
Activity Details **Following are the session deliverables:** • Explain the facts and trivia • Next class challenge • Project for the day • Additional Activity (Optional)	
FEEDBACK • **Appreciate and compliment the student for trying to learn a difficult concept.**	

- **Get to know how they are feeling after the session.**
- **Review and check their understanding.**

Teacher Action	Student Action
You get “hats-off” for your excellent work! In the next class, we will learn about web servers	*Make sure you have given at least 2 hats-off during the class for:* 

PROJECT OVERVIEW DISCUSSION

Refer the document below in Activity Links Sections

Teacher Clicks ✖ End Class

ACTIVITY LINKS

Activity Name	Description	Links
Teacher Activity 1	Reference Code -RGB	https://github.com/procodingclass/PRO-C246-Reference-Code
Teacher Activity 2	Reference Code -Fade LED	https://github.com/procodingclass/PRO-C246-Reference-Code-SA

Teacher Reference 1	In-Class Quiz	https://s3-whjr-curriculum-uploads.whjr.online/5d88ef57-2b7c-45ef-929e-96931c525081.docx