



Overview:

In this TI-Nspire™ CX II lesson, students apply their knowledge of analog output to control the brightness of the red, green, and blue channels of an RGB (Red-Green-Blue) LED mounted in a diffuser box. A program is written to display a random set of red, green, and blue colors. Students experience how emitted light mixes to make secondary and tertiary colors. As an option, the mood lamp housing can be designed in CAD software and printed on a 3-D printer.

Goals:

1. Apply the principles of emitted light color mixing.
2. Design and build a lamp housing for RGB (Red-Green-Blue) LED.
3. Author a Python program that generates a random red, green, and blue analog output value and displays the mixture on the mood lamp for one second.
4. Create a 3-D printed housing for mood lamp (optional).

Background:

Light is very familiar to us all, but what is it? Light is a wave that moves through a stationary feature of space called the electromagnetic field. Light has many interesting features that our eyes can perceive. One feature is color while another is brightness.

The color of light is due to the frequency of the wave, that is, how often it vibrates up and down. Frequency of light is measured with the units of Hertz (Hz). The brightness of light is due to how much energy is within the wave and may be measured in Lux (Lx). Interestingly, our brain perceives different colors of light when three primary emitted colors are mixed with different brightnesses before entering our eyes.

For example, the brain will perceive yellow if a light wave with exactly the right single frequency enters the eye or when the two colors of red and green light are mixed evenly before entering the eye. Similarly, the eye will perceive purple when a light wave with exactly the right single frequency enters the eye or when bright red is mixed with dim blue. The colors red, green, and blue are referred to as primary colors when emitted light is mixed before entering the eye.

In this project, a special light emitting diode is used that actually contains three LEDs. The RGB (Red-Green-Blue) LED has a red, green, and blue diode in one plastic case. Since there are three LEDs in one case and the three share a common cathode (-) lead, the package has four leads. One is the red anode (+), another is the green anode (+), another is the blue anode (+) while the forth is the common cathode (-) for the three separate anodes.

When the RGB LED anodes are connected to three separate analog out pins on the TI-Innovator Hub and the common cathode is connected to a ground pin, the brightness of each color may be independently controlled using the skill of analog output (skill builder 3). In order to mix the light from the three separate internal LEDs, a white colored shade is helpful. A shade can be built from white construction paper or printed with a 3D printer.

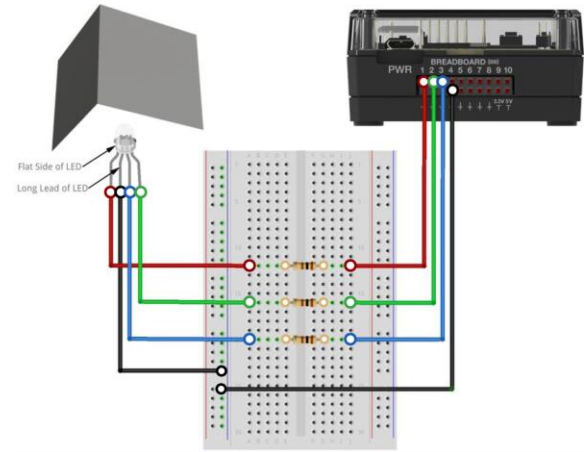
Materials and Tools:

- TI-Nspire CX II Technology
- TI-Innovator™ Hub with USB Cable
- White construction paper
- White 3D printer filament (optional)
- TI-Innovator Breadboard Pack:
 - Breadboard
 - Male to Male jumper Cables
 - Female to Male jumper Cables
 - RGB (Red-Green-Blue) LED
 - 100 Ohm Resistors



Build the Hardware:

1. Insert each leg of the RGB LED into a separate Male to Female Jumper Cable.
2. Insert three 100 Ohm resistors into separate columns of the breadboard. Be sure each resistor bridges the gap in the board so the two ends are electrically separated.
3. Connect each of the RGB anodes to one of the 100 Ohm Resistors using the Male to Female Jumper Cable.
4. Use a Male to Male Jumper Cable to connect the other end of each 100 Ohm resistor to BB1, BB2, and BB3.
5. Connect the common cathode leg of the RGB LED to the blue ground bus on the breadboard using the Male to Female Jumper Cable.
6. Use a Male to Male Jumper Cable to connect the blue ground bus of the breadboard to any ground on the TI-Innovator Hub.



Write the Software for the TI-Nspire CX II:

Example Code for the TI-Nspire CX II:

```
from ti_hub import *
from random import *
from time import *

red=analog_out("BB 1")
green=analog_out("BB 2")
blue=analog_out("BB 3")

t=int(input("How Many Seconds? "))
s=0
while s<t:
    r=randint(0,255)
    g=randint(0,255)
    b=randint(0,255)
    red.set(r)
    green.set(g)
    blue.set(b)
    text_at(6,"R = "+str(r),"left")
    text_at(7,"G = "+str(g),"left")
    text_at(8,"B = "+str(b),"left")
    sleep(1)
    s+=1
```

Task: Write a Python program that...

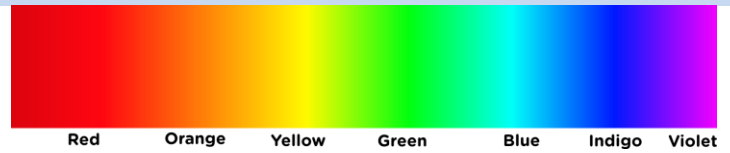
1. connects three analog_out objects 1 through 3 to the breadboard pins BB1, BB2, and BB3 respectively.
2. defines three variables: one each for the red, green, and blue output values.
3. defines a variable for duration of the program.
4. prompts the user for the duration and stores it in the appropriate variable.
5. utilizes a while statement to run a loop for the duration of the program.
6. tells each loop to create a new random red, green, and blue brightness value in the range of 0 to 255 and stores that value in the appropriate variable.
7. sets the analog output for each color channel to the appropriate random brightness value.
8. displays the random brightness value to the user.
9. waits for 1 second or longer.
10. increments a timer index variable.
11. exits the while loop when the timer index is greater than the duration.

Teacher Tip: This is the first time the Python code is not provided to the student.



Extra for Experts:

Write a program that slowly displays each of the colors in the rainbow from red to violet.



Teacher Tip: The RGB values in the program may need to be adjusted to match the colors in the rainbow. Numbers will vary according to user.

Sample Extra for Experts Program:

```
from ti_hub import *
```

```
red=analog_out("BB 1")
green=analog_out("BB 2")
blue=analog_out("BB 3")
```

```
text_at(6,"RED","left")
red.set(255)
green.set(0)
blue.set(0)
sleep(4)
```

```
text_at(6,"ORANGE","left")
red.set(255)
green.set(50)
blue.set(0)
sleep(4)
```

```
text_at(6,"YELLOW","left")
red.set(255)
green.set(200)
blue.set(0)
sleep(4)
```

```
text_at(6,"GREEN","left")
red.set(0)
green.set(255)
blue.set(0)
sleep(4)
```

```
text_at(6,"BLUE","left")
red.set(0)
green.set(0)
blue.set(255)
sleep(4)
```

```
text_at(6,"INDIGO","left")
red.set(75)
green.set(0)
blue.set(130)
sleep(4)
```

```
text_at(6,"VIOLET","left")
red.set(128)
green.set(0)
blue.set(128)
sleep(4)
```

```
text_at(6,"OFF","left")
red.set(0)
green.set(0)
blue.set(0)
```