The Physics Teacher, 2016, **54**(4), 244-245 (doi: 10.1119/1.4944370)

A simple Experimental Setup for Teaching Additive Colors with Arduino

Paulo Simeão Carvalho[1,2,3], Marcelo Hahn[2]

[1] Faculdade de ciências da Universidade do Porto, UEC, Porto, Portugal
[2] Departamento de Física e Astronomia, FCUP, Porto, Portugal
[3] IFIMUP-IN, FCUP, Porto, Portugal

psimeao@fc.up.pt ; dumash@gmail.com

The result of additive colors is always fascinating to young students. When we teach this topic to 14-16 years old students, they do not usually notice we use maximum light quantities of red (R), green (G) and blue (B) to obtain yellow, magenta and cyan colors, in order to build the well-known additive color diagram of figure 1.
But how about using different light intensities for R, G and B? What colors do we get? This problem of color mixing has been intensively discussed for decades by several authors, as pointed out by Ruiz's "Color Additionand Subtraction Apps" work and the references included therein [1]. An early LED demonstrator for additive color mixing dates back to 1985 [2], and apps to illustrate color mixing are available online [1, 3, 4].

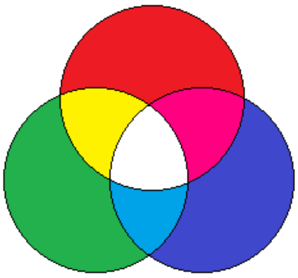

Figure 1: diagram for additive colors

In this work, we describe an experimental setup making use of a microcontroller device: the Arduino Uno. This setup is designed as a game, in order to improve students' understanding of color mixing.

Experimental setup

We have built a simple electric circuit where the light of an RGB LED, whose basic functioning is described in [2], is controlled by an Arduino Uno. Arduino is a microcontroller development board, an open-source electronics platform based on easy-to-use hardware and programming language [5].

Figure 2 presents the scheme of the full experimental setup, where all the components are identified. The complete list of components is: 1 Arduino UNO unit, 2 Breadboards, 1 resistor 10 kΩ, 10 resistors 220 Ω, 2 RBG LEDs, 1 Red LED, 1 Blue LED, 1 Green LED, 1 Yellow LED, 3 potentiometers 10 KΩ, 1 push button, wires.

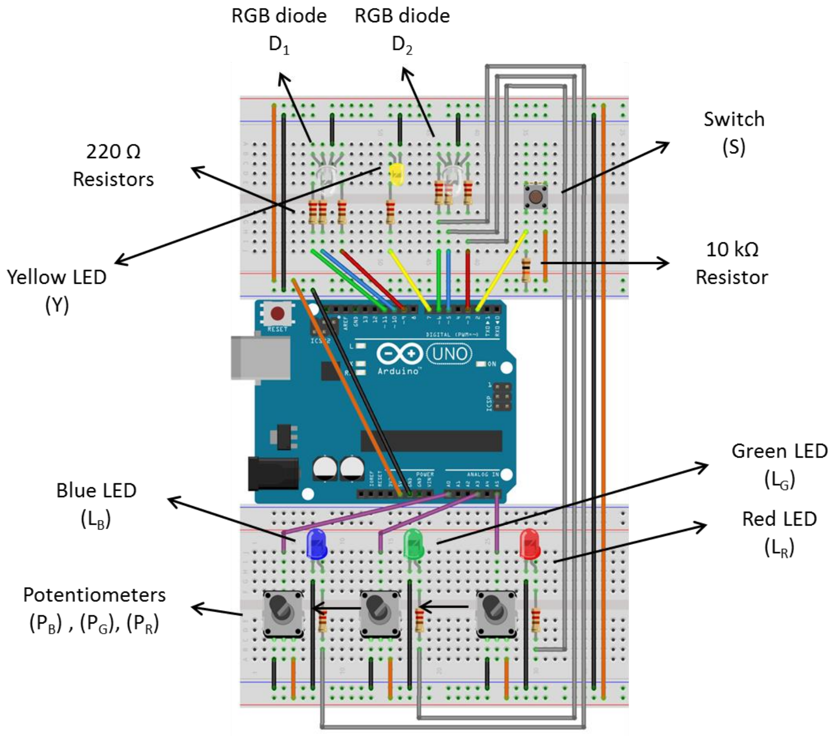

Figure 2: experimental setup. The scheme was created with *Fritzing* software [6].

D1 and D2 are RGB LEDs. In D1, the components of R, G and B are fixed and randomly chosen by the microcontroller, when the switch S is pressed; light in D2 LED is controlled by the students via knobs (which, indeed, are potentiometers) P_R, P_G and P_B – each knob controls light of a different color, respectively red, green and blue.

This control of the D2 LED is due to a feature in some Arduino's output pins called PWM (Pulse Width Modulation) [7], which allows the digital output to be controlled as if it was an analog output, hence, the students are able to increase the intensity of each desired primary color by the amount provided in the rotation of the potentiometer.

The students can actually see how much light intensity of each color is sending to D2 LED, by means of LEDs L_R, L_G and L_B.

All LEDs and RGB LED pins need to be protected by a current-limiting resistor of 200-300 ohms. For simplicity, we have used only 220 ohm resistors because these work quite well to that purpose for all those components.

The setup is used as a game in classroom, measuring the time interval students take to obtain the right combination of colors, or letting them compete for who's the first to complete the task. For practical reasons, we restrict the intensity of each color to lower values (typically in the range 0 to 200 in decimal notation [8]), because human eye can see better differences in colors with low intensity of light. The main goal

The program [9] establishes that when all values of R, G and B chosen by the students *(via the potentiometers)* are within about 10 % those fixed by the microcontroller, a yellow light (Y) turns ON.

Then they can conclude how much of each R, G and B color was necessary to obtain the resulting color of D1 LED.

Very often, the random color in D1 is a combination of all colors with quite different intensities, and this helps students to get an idea of how many combinations we can have from only the three primary colors, and to understand how much physics is important in technology to produce all the images in TVs, computers and cell phones.

We have tested this experimental setup with 43 students of low secondary level (14 to 16 years old) and it was a real success. They enjoyed playing against each other's, but most important is they have learned much more than with the traditional visualisation of additive colors in textbooks. Moreover, we have found they developed a huge interest in programing language. Programing is a very effective task to promote formal thinking development, creativity and critical reasoning. Teachers may find surprising how many creative ideas can arise when they let students interfere with the programming code, and even improve the "game" to a more sophisticated and challenging educational level.

Acknowledgements

This work is partially financed by the FCT – Fundação para a Ciência e a Tecnologia (Portuguese Foundation for Science and Technology) within project UID/NAN/50024/2013.

References

[1] F. Ruiz, M.J. Ruiz, "Color Addition and Subtraction Apps," *Phys. Teach.* **53**, 423-427 (2015).

[2] J.E. Kettler, "Tri-Color LED Demonstrator," Phys. Teach. **23**, 559-560 (1985).

[3] https://www.omsi.edu/tech/activities/colormix.swf, retrived on 05/11/2015.

[4] http://trycolors.com/, retrived on 05/11/2015.

[5] https://www.arduino.cc , retrieved on 05/11/2015.

[6] http://fritzing.org/home , retrieved on 05/11/2015.

[7] https://www.arduino.cc/en/Tutorial/PWM, retrieved on 05/11/2015.

[8] http://www.w3schools.com/cssref/css_colors.asp , retrieved on 05/11/2015.

[9] See EPAPS Document No. [**number will be inserted by AIP**] for the code. For more information on EPAPS, see http://www.aip.org/pubservs/epaps.html.