

Colour Mixing with an RGB LED [Arduino] WIP

Keywords: 2017, Arduino, Beginners, Colour Mixing, Computer science, Computing, E&T, Interfacing, LED, Microcontroller, Physics, RGB, Sketch

Meta Description

Mix some colours using an Arduino Uno to control an RGB LED. Introduction to microcontroller programming. Sample code available. Great for beginners!

Learning Objectives

Understanding the operation of an RGB LED.

Implementing and debugging a microcontroller system which can control an LED.

Familiarization with the coding structure used to program a microcontroller.

Key Terms

Duty Cycle

The ratio of the 'high' time to a 'low' time in a PWM wave.

Forward Voltage

The voltage drop across the LED when it switches on.

Function

A sub-program which executes a particular function. It can be called by the main function or another function.

Lead

Leads are short metal wires extending out from electronic components which allow them to be plugged into a breadboard.

Light Emitting Diode (LED)

A polarized two-lead device which lights up.

Modular Programming

A programming method in which a program is divided into a number of functions, with each function performing a specific task.

Polarised Device

An electrical device which only allows current flow in a single direction.

Pulse Width Modulation (PWM)

A method of varying the output power from a microcontroller pin by outputting a square wave with a varying duty cycle.

Sketch

Code files used to program an Arduino.

Method

Step 1

Assemble the circuit according to the Fritzing diagram in Figure x. The RGB LED is a polarized device which can only be connected in one direction. If it is not connected as shown in the diagram the circuit will not work as expected and the LED may get damaged. The cathode terminal must be connected as described in the table. Consult the manufacturer's datasheet for the pin-out diagram.

Colour Mixing with an RGB LED [Arduino] (steamexperiments.com)

Image not found or type unknown

Step 2

Launch the Arduino IDE software on the computer. Follow the [link](#) for the initial setup for first-time users.

Step 3

Paste the code into a sketch within the Arduino IDE.

Step 4

Connect the Arduino Board to the computer via the USB cable. It is advisable to unplug the laptop from it's charger before starting the experiment – a connection to the mains power line can affect the stability of the circuit. Place the Arduino board on a non-conductive surface such as glass, wood or plastic to prevent current conduction through the surface which can damage the board.

Step 5

Click Upload in the IDE.

Step 6

Observe what happens.

Step 7

After dismantling, store the Arduino board in an antistatic bag to prevent damage due to electrostatic discharge.

Wiring Table:

Arduino Pin	RGB LED Pin
GND	Common Cathode
9	Red
10	Green
11	Blue

Precautions

1. Always safely eject the USB connector from the computer.
2. Never short the power pins (5V or 3V) of the microcontroller with its ground (GND) pins. This creates a short circuit which can damage the device and cause substantial heating which may lead to burns.
3. The safest way to power the Arduino board is from the laptop via the USB cable. In the unlikely case that the Arduino must be powered via a power supply and not from the laptop, set a current limit of around 100mA on the supply. Take good care to keep the positive and negative supply terminal outputs from touching to avoid a short circuit condition which can lead to current limiting (which could lead to impaired performance), heating, damage to the circuit or laptop and in rare cases sparking.
4. The Arduino board and the components usually have sharp edges which can prick fingers, particularly when wiring the breadboard, so care should be taken during assembly and disassembly of the circuit.
5. Keep liquids away from the work surface and do not place any of the equipment on a damp surface. Preferably, floors near the work surface should also be dry.

Narrative

Imagine three bulbs on a ceiling. One is red, one is green and the other is blue. Now, imagine that each light is controlled by a switch on the wall. By turning the switches on or off you can switch on or off any light at full brightness. This is what is happening in the `mainColors()` function: when the output pin associated with a certain colour outputs a logic 'high' that bulb is switched on, when the pin output a logic 'low', the bulb is switched off.

Now, imagine the same three-bulb scenario, but now each light can be controlled by a dimmer switch. The brightness of each bulb can be varied from the off state to the fully bright state. Thus, you can control the effect of each bulb on the final, mixed colour. This is what happens in the `showSpectrum()` function. The PWM function on the output pin acts as the dimmer switch. The larger the duty cycle value, the brighter the colour shines until a value of 255 is reached, which corresponds to full brightness. Conversely, the smaller the duty cycle value, the dimmer the colour becomes, until a duty cycle of 0 is reached, which corresponds to the LED being supplied no power (fully off).

Questions

Is the white light emitted from the LED pure white light?

No. It does not contain all the wavelengths in white light.

How does PWM work?

It varies the power output at the pin.

How is the brightness varied?

Increasing current increases brightness.

Why do we need the resistors?

To limit the current in the LED.

How is a colour 'switched off'?

Voltage across the colour is lower than forward voltage.

Brief Explanation

The RGB LED has three tiny red, green and blue LEDs within it. Each LED has two leads. The cathode leads of the devices are connected at the common terminal. The remaining three leads are the anodes of each colour. The voltage across an LED of a given colour, V_c , is:

$$V_c = V_{\text{pin}} - V_{\text{cathode}} \quad [1]$$

Where V_{pin} is the voltage on the colour pin and V_{cathode} is the cathode voltage. In this case $V_{\text{cathode}} = 0V$. Therefore:

$$V_c = V_{\text{pin}} \quad [2]$$

Each colour has a special forward voltage. When V_c exceeds this voltage, the LED of that colour switches on.

The larger the current through the LED, the brighter it is. By increasing V_{pin} the current through the LED is increased and it shines brighter. The opposite is true is when the voltage is decreased. Thus, the contribution of a certain colour can be varied by controlling the voltage at the output pin. This is done through the `analogWrite` function, which uses a special microcontroller feature called pulse width modulation to change the voltage at the colour pin.

Modular programming should be used when dealing with microcontrollers. In this approach a number of sub- functions are called by the main function. This ensures that program flow remains simple and makes code easier to debug. The sketch (code) is split into two distinct parts. In the first part, a number of functions are called. In the second part, these functions are defined. This is a modular structure.

Detailed Explanation

The RGB common cathode has 4 leads: 3 anode leads, one for each colour component and one common cathode for all three LEDs. Each colour has a characteristic forward voltage. The forward voltage drop is the voltage across the LED when it conducts electricity. In this condition, the current passes through the LED and the diode is illuminated. The LED is a current controlled device- the brightness of a given colour is directly proportional to the current passing through the LED.

The voltage is altered to control the current through the LED. From the IV characteristic in the figure below, it is evident that a linear variation in the voltage across the LED does not correspond to a linear variation in current through the LED.

Colour Mixing with an RGB LED [Arduino] (steamexperiments.com)

Image not found or type unknown

The variation in the output brightness shares the same non-linear relationship with the voltage.

Thus, by controlling the voltage across the LED corresponding to a particular colour, the brightness of that colour is varied. The voltage across an LED of a given colour, V_c , is:

$$V_c = V_{pin} - V_{cathode} \quad [3]$$

Where V_{pin} is the voltage on the colour pin and $V_{cathode}$ is the cathode voltage. In this case $V_{cathode} = 0$ V. Therefore:

pin

$$V_c = V \quad [4]$$

When V_c is below the characteristic forward voltage of the colour, V_F , the colour will be turned off or be very dim. When $V_c = V_F$ operation surpasses the knee of the characteristic and the LED illuminates brightly. Beyond this point a small change in voltage corresponds to a large growth in current.

The 330 Ω resistors perform a current limiting function. They prevent the LED from draining too much current from the microcontroller and exceeding the current sourcing capabilities of the output pin. Exceeding the maximum rated current output can lead to possible burnout of the LED, pin and/or chip.

Notes on the code:

The `analogWrite` function was used for colour mixing in the `showRGB` function. The `analogWrite` function controls the pulse width modulation (PWM) output on the microcontroller pins. It passes two values: `analogWrite(pin, value)`, where `pin` is the pin number and `value` refers to duty cycle (d.c.). A value of 0 corresponds to the pin fully off (d.c. = 0) while a value of 255 corresponds to the pin being fully on (d.c. = 1).

The 1000ms delays enable the user to observe the colour changes. Without this delay, the changes would occur at the clock speed of the microcontroller, which is too fast for the human eye to perceive.

Applications and Research

Applications

Microcontrollers are used in a wide range of applications. They are used to control alternative energy systems to maximise energy generation and reduce waste. They are also used in automotive electronics and consumer electronics including televisions, microwaves and washing machines, particularly for automated processes.

The microcontroller model chosen depends on the application. When choosing a microcontroller its specifications need to be considered. These include accuracy, precision, dimensions, word size, memory, ADC/DAC capabilities, number of pins, EMI immunity, power consumption and cost.

Research

Microcontrollers are often used in sensor systems. A recent project focused on using a microcontroller in an air pollution monitoring system which can be used by local authorities, companies or individuals. In this system a microcontroller received inputs from 4 analogue chemical sensors and external sensors for temperature, humidity and pressure. The microcontroller processes the input signals, formats them and transmits them to an external control board.

Investigation

- At different points in the experiment, use a voltmeter to read the voltage across the different colour LEDs. To do this, attach the probe of the black lead (called the common lead) of the meter to ground (GND) and hold the probe of the red lead of the meter to one of the colour pins of the LED. What is the relationship between the voltage across the LED and the brightness of that colour?



Subjects

Computer Science
Engineering & Technology
Physics

Education

Post Secondary
University

Time Required

~30 minutes
Preparation: 10 minutes
Conducting: 30 minutes
Clean Up: 5 minutes

Cost

25 – 50 €

Recommended Age

13 – 16
>16

Number of People

1 participant

Supervision

Not Required

Location

Indoors
Laboratory

Materials

3 x 330 Ω Resistors

5 x Jumper Wires

Arduino Uno Board

Breadboard

Computer with Arduino IDE software installed. To [download the software](#)

LED – RGB, Common Cathode

USB cable to connect the Arduino Board to the computer

Contributors

Natasha Padfield

Author

Benji Fenech Salerno

Editor

Sources

Comprehensive Advanced Microcontrollers and Microprocessor; S.K Venkata Ram, Pg 209 (Book)

Microelectronic Circuits; Sedra and Smith, Pgs 156, 157, 204, 205 (Book)

[10 Steps to Selecting a Microcontroller](#)

[AirSensEUR: An Open Data/Software/Hardware Multi-sensor Platform for Air Quality monitoring. Part A: Sensor Shield.](#)

[AnalogWrite\(\)](#)

[Applications for Microcontrollers \(MCU\)](#)

[EMI Electromagnetic Interference Basics](#)

[Experiment 3: Driving an RGB LED](#)

[Delay Loops](#)

[LED Current Limiting Resistors](#)

[Light – Emitting Diodes \(LEDs\)](#)

[Additional Content](#)

[4x4x4 LED Cube \(Arduino Uno\)](#)

[10 Best Microcontroller Boards for Hobbyists and Engineers](#)

[10 Steps to Selecting a Microcontroller](#)

[25 Microchips That Shook the World](#)

[112 Lights Projects](#)

[Arduino Rising: 10 Amazing Projects People Are Doing With The Tiny Microcontroller](#)

[How Microcontrollers Work](#)

[Simple Arduino Traffic Lights](#)

[What is a Microcontroller?](#)

[Cite this Experiment](#)

Padfield, N., & Fenech Salerno, B. (2017, September 30). Colour Mixing with an RGB LED [Arduino]. Retrieved from <http://steamexperiments.com/experiment/colour-mixing-with-an-rgb-led-arduino/>

First published: **September 30, 2017**

Last modified: **April 27, 2020**