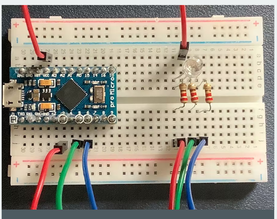


Common Anode RGB LED © GPL3+

A short, simple project to make a common anode RGB LED illuminate the colours of the visual light spectrum.

common anode | led

AUTHOR

lewiskell
2 PROJECTS 1 FOLLOWER

FOLLOW

PUBLISHED ON

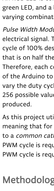
September 17, 2021

TABLE OF CONTENTS

COMPONENTS AND SUPPLIES

Arduino Leonardo
In this project I used the ProMicro, a small clone of the Leonardo.
× 1

Resistor 220 ohm
× 3



RGB Diffused Common Anode
× 1

Breadboard (generic)
× 1

Jumper wires (generic)
× 1

ABOUT THIS PROJECT

Introduction

This project uses a common anode RGB LED to produce colours of the visible light spectrum. This will allow you to explore the connection between digital (the Arduino) and analogue (the LED).

Theory

An RGB LED is comprised of 3 individual LEDs on a single package; a red LED, a green LED, and a blue LED. These three primary colours can be added together in varying combinations to produce a wide array of colours.

Pulse Width Modulation (PWM) is a method of reducing the average power of an electrical signal. The *duty cycle* is the proportion of time the signal is on. A duty cycle of 100% describes a full power signal. A duty cycle of 50% describes a signal that is on half the time and off the other half, which corresponds to half power. Therefore, each colour of LED can be varied in brightness by using the PWM pins of the Arduino to achieve various colour combinations. analogueWrite() is used to vary the duty cycle of the PWM pins. With an 8-bit PWM output, this gives $2^8 = 256$ possible values to vary the duty cycle. This is how different colours are produced.

As this project utilises a *common anode* LED, the ON state of the LED is active low meaning that for the LED to illuminate, the signal must be LOW. This is *opposite* to a *common cathode* LED. As a result, for a *common anode* LED to be ON a 0% PWM cycle is required using analogueWrite(0) and for a LED to be OFF, a 100% PWM cycle is required using analogueWrite(255).

Methodology

This project uses an Arduino Leonardo clone (ProMicro), however can be recreated using other microcontrollers.

Pulse Width Modulation (PWM) is used to control the brightness of the LEDs. The PWM pins are shown by Figure 1. Pins 3, 5, and 6 are used for the red, green, and blue LEDs respectively. For other microcontrollers, refer to the respective board pin diagram.

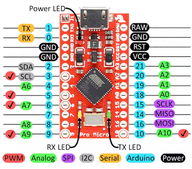


Figure 1: Arduino ProMicro Pin Diagram (Source: SparkFun)

The pins of a common anode RGB LED are shown by Figure 2. The positive pin of the RCB LED connects to the VCC pin of the Arduino to provide power. The red, green, and blue LEDs are then connected to pins 3, 5, and 6 using a 220Ω resistors in series to limit current, preventing the LEDs from being damaged.

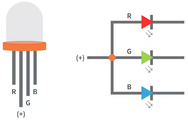


Figure 2: Common Anode RGB LED diagram (Source: CircuitBread)

CODE

CA_RGB_LED Arduino

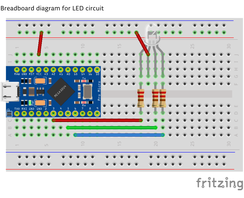
```
// LEDs must be connected to arduino PWM pins - see board pinout
#define RED  6  // pin that red led is connected to
#define GREEN  5 // pin that green led is connected to
#define BLUE  3 // pin that blue led is connected to
#define DELAY  20 // interval delay in ms

void setup() {
  // LED connection pins to be set as an output
  pinMode(RED, OUTPUT);
  pinMode(GREEN, OUTPUT);
  pinMode(BLUE, OUTPUT);
}

void loop() {
  // fade from red to green
  for(int i=0; i<255; i++) {
    analogWrite(RED, i);   // red initially ON
    analogWrite(GREEN, 255-i);  // green initially OFF
    analogWrite(BLUE, 255);  // blue OFF
    delay(DELAY);  // wait before next transition
  }
```

SCHEMATICS

CA_RGB_LED

DOWNLOAD

Breadboard diagram for LED circuit



COMMENTS

Please log in or sign up to comment.

Be the first to comment!