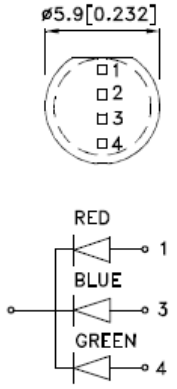
CSE 102 – Studio 4

Color Pixel

To explore the use of color display capabilities, we will use the tri-color LED. The physical pinout is illustrated below. The full datasheet is available here.



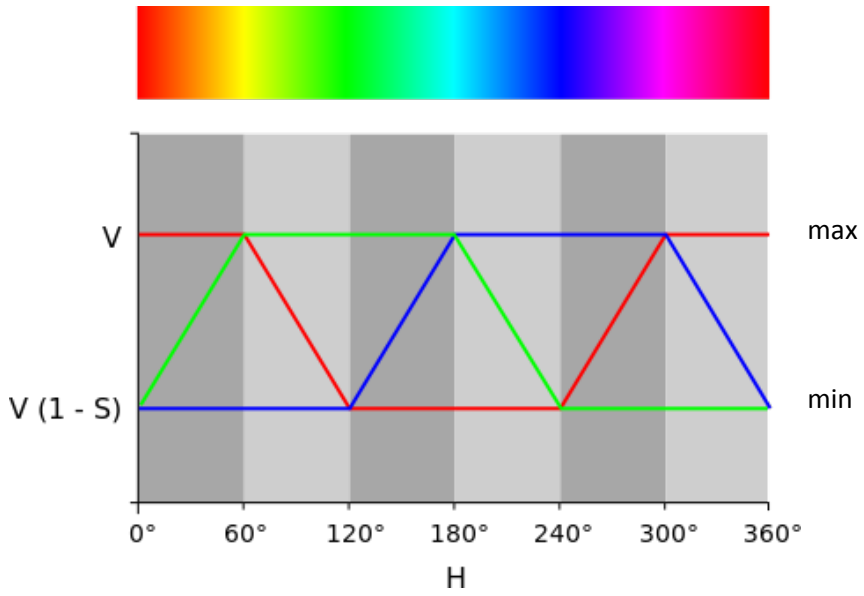
You should connect pin 2 to GND, pins 1, 3, and 4 should be connected to 3 digital output pins on the Arduino board that support PWM output (use three of pins 3, 5, 6, 9, 10, and 11) through a 330Ω resistor (i.e., wire each color just like the LEDs you have been using all semester). One difference to note is that the common cathode (pin 2) is the long lead, which is different than the discrete LEDs we've used so far.

PWM is described as quoted below from the Arduino PWM tutorial page:

> *Pulse Width Modulation, or PWM, is a technique for getting analog results with digital means. Digital control is used to create a square wave, a signal switched between on and off. This on-off pattern can simulate voltages in between full on (5 Volts) and off (0 Volts) by changing the portion of the time the signal spends on versus the time that the signal spends off. The duration of "on time" is called the pulse width. To get varying analog values, you change, or modulate, that pulse width. If you repeat this on-off pattern fast enough with an LED for example, the result is as if the signal is a steady voltage between 0 and 5v controlling the brightness of the LED.*

We will exploit this property for each color individually, allowing us to independently control the intensity of each color.

Write a program that cycles through the color sequence below:

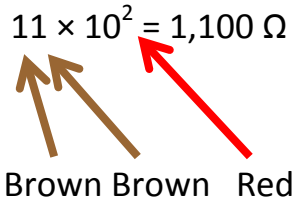


You can adjust the intensity of each color with the analogWrite() function (from Studio 1 – Analog Output). In the waveform above, the vertical axis position labeled V corresponds to analog value 255 (maximum intensity) and the position labeled V(1-S) corresponds to analog value 0 (minimum intensity). Use the delay() function to move across the x axis at some reasonable rate (i.e., slow enough to see, fast enough that it doesn't take forever).

Observe the color sequence visible out of the plastic housing (which serves as a diffuser) and show it to the instructor or TA. This is how one pixel works in larger color displays (that have lots of pixels).

5 by 7 Pixel Display

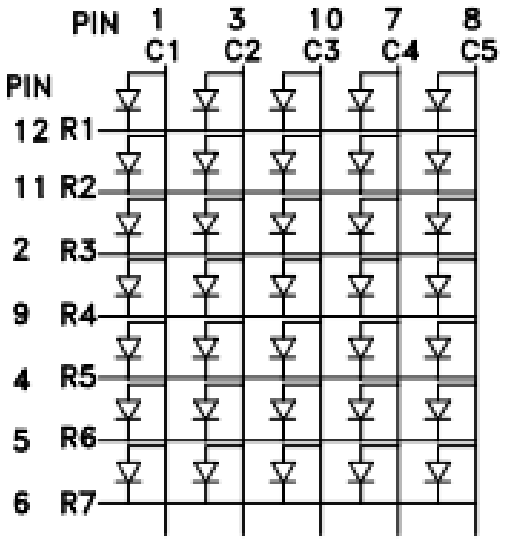
Wire one column of the 5 by 7 pixel display by connecting column 1 (pin 1) to a digital output port. Each row (pins 12, 11, 2, 9, 4, 5, 6) should be connected to a digital output port through a 1.1 kΩ resistor. A 1.1 kΩ resistor will have a color code as follows:



Recall that the table of values is as follows:

0	1	2	3	4	5	6	7	8	9
black	brown	Red	Orange	yellow	green	blue	violet	grey	white

The complete pinout for the 5 by 7 pixel display is shown below. The full datasheet is here.



Repeat the Alternate Encoding exercise of Studio 3, counting 0 to 7 and displaying the current count using a "one-hot" encoding scheme. Here, 0 is represented by all LEDs off, 1 is represented by the LED in row 1, column 1 being on, 2 is represented by the LED in row 2, column 1 being on, etc.

Recall that since the row is connected to the cathode of the LED, to turn the LED on the column digital output must be set HIGH and the row digital output must be set LOW.

Once you are successfully sequencing through all of the LEDs in column 1, extend the sequence to the remaining columns. Each column will need to be connected to a digital output port. Cycle through a "one-hot" encoded count of 1 to 5 (representing the columns), for each column cycle through the row count above. This should yield each of the LEDs turning on one at a time, with a 1 LED pause between columns (when the row count is 0).

Again, remember to use the correct polarity for controlling the columns. Since they are connected to the anode of the LED, on requires a HIGH output and off requires a LOW output.

Show your display sequencing through all pixels to an instructor or TA.

The above mechanism only supports a limited number of LEDs on at a time. To independently allow any pattern of pixels to be lit simultaneously we must perform faster multiplexed control. Using the column-based multiplexed control pseudo-code described in class, pick a fixed pattern to display on each column (you can choose whatever you wish, just make sure that at least some of the columns have more than one pixel on and there is a variety of patterns across the columns). Implement the multiplexed control to display your chosen image.

Show off your image to an instructor or TA. You will use this 5 by 7 pixel display, and the column-based multiplexed control, as component pieces of the next assignment.