Engineering Principles

Name___________________________________

Electrical Engineering

Color Wheel 4.9.1 Software: Timing

Date_____________________ Team __________

Problem statement

Develop a persistence-of-vision color wheel utilizing 8 RGB diodes and an Arduino UNO R3 board.

Programming: Getting Started

1. Find your Arduino code file. If it has been modified by another member of your team since you last worked on it you may need to download it from Canvas.
2. Download any updated image.c files that your group has made to the same directory as your code file.
3. Open your program file with Arduino. You should see tabs for your image files as well.

Programming: Timing

It is likely that your code took longer to draw than you expected. There are several ways to speed things up. Discuss with your project team what options you may want to consider. Some possibilities may be reducing the image size, optimizing the code, particularly in the inner-most loop, reducing the image resolution (instead of stepping by 1 through the intensity loop, step by a bigger number).

4. Modify your system so that it will display a picture in less than 0.25 seconds.

To display the image in the same place each time we need to have a reference point in the rotation. We will use a Hall-effect sensor that toggles between HIGH and LOW each time the handle turns. First wait for the sensor to be HIGH, and then wait for the sensor to go LOW again. Because of the way the part is designed, the LOW-going edge is more precise.

The sensor is connected to an analog pin which has a range from 0 to 1023 (10 bits). By setting a HIGH threshold of 600 and a LOW threshold of 400 we avoid false triggers as the input slowly moves through the LOW to HIGH transition. This is called hysteresis.

5. Create a function called `waitForTrigger()` that reads the analog I/O pin connected to the Hall-effect sensor. Use `analogRead()` to get the input value. Use a first `while ()` loop to wait until the pin is high (value > 600). Then use a second `while`() loop to wait until it goes low (<400).
6. Call the `waitForTrigger()` function at the beginning of your main `loop()` function.

Programming: Finish and Save

7. **In your lab notebook** write down what you expect the code to do.
8. Test the code using your group's hardware.
9. **In your lab notebook** describe what your code actually did. If different from your expectations, explain. Share the results with your team.
10. How long does it take to draw a complete picture?
11. Is the picture recognizable?
12. Save your code and upload it to your group's file space on Canvas. Use a new name so that prior versions of the code are not over-written.