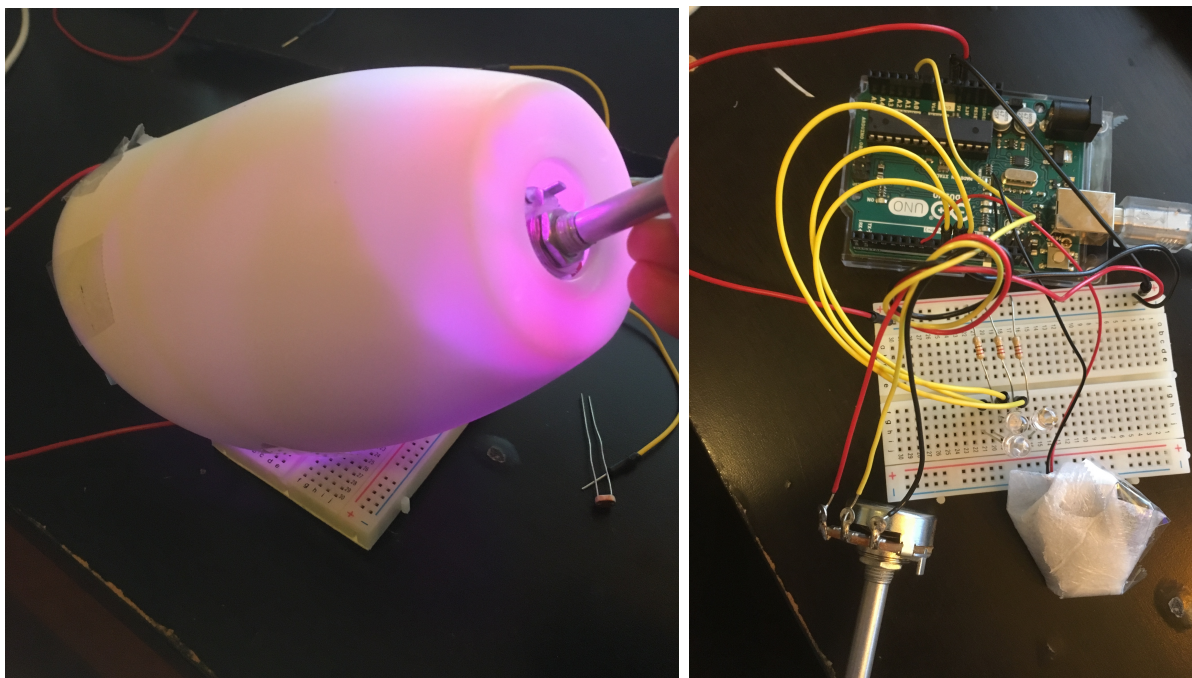


Jin Jeon
Info C262

Lab 6: Output Coincidence

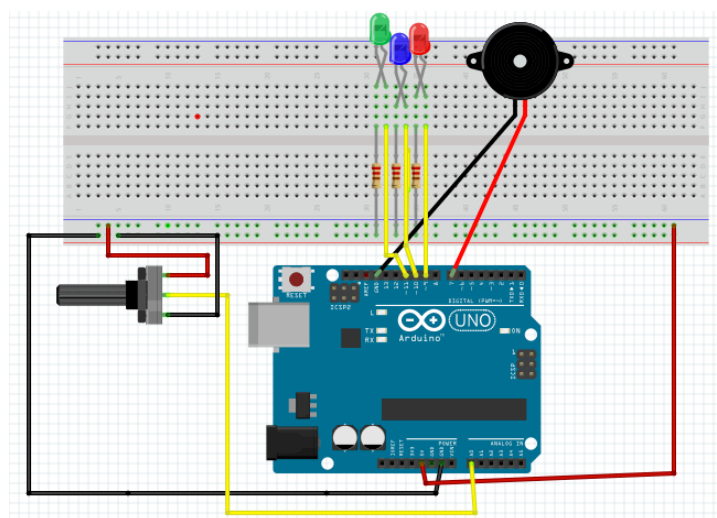
Description

In this lab, I created a calming mood lamp as a sleeping aid. The mood lamp changes color slowly over time to create a calm, ambient light environment while a classic song Twinkle Twinkle Little Star is played by the piezo speaker within the lamp. I attached a potentiometer at the hole of the lamp so that the user can control both the speed of color changes and the tempo of the song. The slower the tempo, the slower the color changes will show as output. I tried different materials to change the sound of the piezo speaker, but then I decided to place plastic foams over it to make the sound less abrupt. Also, placing the speaker on the 3 LED lights allowed light to be more diffused efficiently.



Components

- 1 Arduino
- 3 LED (red, green, blue)
- 3 Resistors
- 1 Breadboard
- 1 Potentiometer
- 1 Piezo Speaker
- 9 Jumper wires



Arduino Code

```
//Jin Jeon
//Info C262 Lab 6

// Output
int redPin    = 9;    // Red LED,    connected to digital pin 9
int greenPin  = 10;   // Green LED,  connected to digital pin 10
int bluePin   = 11;   // Blue LED,   connected to digital pin 11
int pot = A0;
int speakerPin = 7;
int toneVal;
int noteDuration = 10; //ms
int redVal    = 255; // Variables to store the values to send to the pins
int greenVal  = 1;   // Initial values are Red full, Green and Blue off
int blueVal   = 1;
int wait = 5;
int i = 0;      // Loop counter
int k = 0;      // sound counter
int length = 15; // # of notes

//twinkle twinkle little star
char notes[] = "ccggaag ffeeddc ggffeed ggffeed ccggaag ffeeddc "; // a
space represents a rest
int beats[] = { 1, 1, 1, 1, 1, 1, 2, 1, 1, 1, 1, 1, 2, 4 };
int tempo = 600;

void playTone(int tone, int duration) {
  for (long x = 0; x < duration * 1000L; x += tone * 2) {
    digitalWrite(speakerPin, HIGH);
    delayMicroseconds(tone);
    digitalWrite(speakerPin, LOW);
    delayMicroseconds(tone);
  }
}

void playNote(char note, int duration) {
  char names[] = { 'c', 'd', 'e', 'f', 'g', 'a', 'b', 'C' };
  int tones[] = { 1915, 1700, 1519, 1432, 1275, 1136, 1014, 956 };

  // play the tone corresponding to the note name
  for (int x = 0; x < 8; x++) {
    if (names[x] == note) {
      playTone(tones[x], duration);
    }
  }
}

void setup()
{
  Serial.begin(9600);
  pinMode(redPin, OUTPUT);    // sets the pins as output
  pinMode(greenPin, OUTPUT);
  pinMode(bluePin, OUTPUT);
  pinMode(speakerPin, OUTPUT);
}

void loop() {
  wait = analogRead(pot);
  i += 1;      // Increment counter
  if (i < 255) // First phase of fades
  {
    redVal    -= 1; // Red down
```

```

    greenVal += 1; // Green up
    blueVal  = 1; // Blue low
}
else if (i < 509) // Second phase of fades
{
    redVal    = 1; // Red low
    greenVal -= 1; // Green down
    blueVal  += 1; // Blue up
}
else if (i < 763) // Third phase of fades
{
    redVal  += 1; // Red up
    greenVal = 1; // Green low
    blueVal -= 1; // Blue down
}
else // Re-set the counter, and start the fades again
{
    i = 1;
}

analogWrite(redPin,  redVal); // Write current values to LED pins
analogWrite(greenPin, greenVal);
analogWrite(bluePin,  blueVal);
delay(wait); //
if (k >= 15) {
    k = 0;
}
if (k < 15) { // (int x = 0; x < length; x++) {
    if (notes[k] == ' ') {
        delay(beats[k] * tempo); // rest
        k = k + 1;
    } else {
        playNote(notes[k], beats[k] * tempo);
        k = k + 1;
    }

    // pause between notes
    delay(wait * 3);
}
Serial.println(wait);
}

```