# DIY Infinity Mirror Kaleidoscope with Arduino

## Setup

If you haven't already, please go to the Arduino website and download the "Arduino IDE" for either Windows or Mac: https://www.arduino.cc/en/Main/Software and then install the software on your computer.

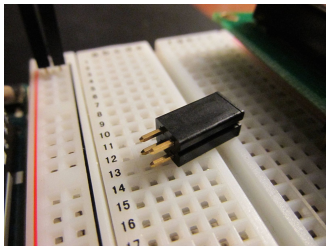
Under **Tools > Board**, select **Arduino/Genuino Uno**.
Under **Tools > Port**, select the **COM** port that has **(Arduino/Genuino Uno)** in it.
We also need the Adafruit **NeoPixel library**. Go to **Sketch > Include Library > Manage Libraries**. In the search box, type "Neopixel." The one we need is the third one on the list: **Adafruit Neopixel**. Click on it. There should now be an option in the bottom right of the box that says **Install**. Click it.

If you and your group have any questions, or get stuck as you work through this in-class exercise, please ask the instructor for assistance. Have fun!

## List of Materials

1 Arduino board, breadboard, and cable
1 pair of scissors (share if needed)
Scotch tape, double-sided tape

Capacitor



Tilt Sensor

In the bag:

- 1 petri dish
- 1 piece of reflective mirror film
- 1 Neoprene foam strip
- 1 LED strip
- 1 piece of white foam board

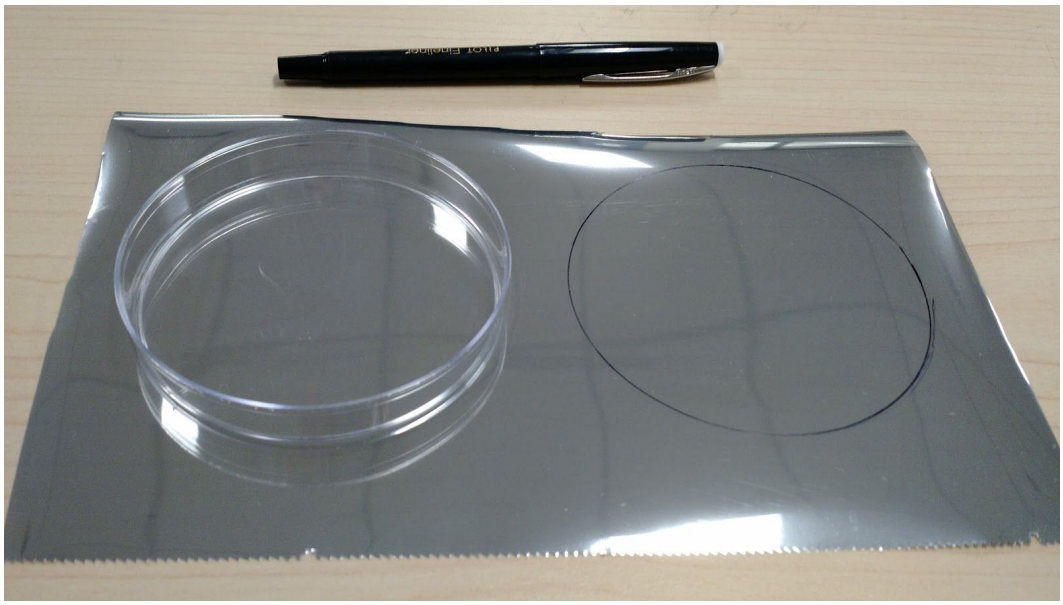
At the front:

- 1 10kΩ resistor (Brown-Black-Orange-Gold)
- 1 560Ω resistor (Green-Blue-Brown-Gold)
- 1 capacitor
- 11-12 wires (assorted length and colour)
- 1 tilt sensor
- Double-sided tape

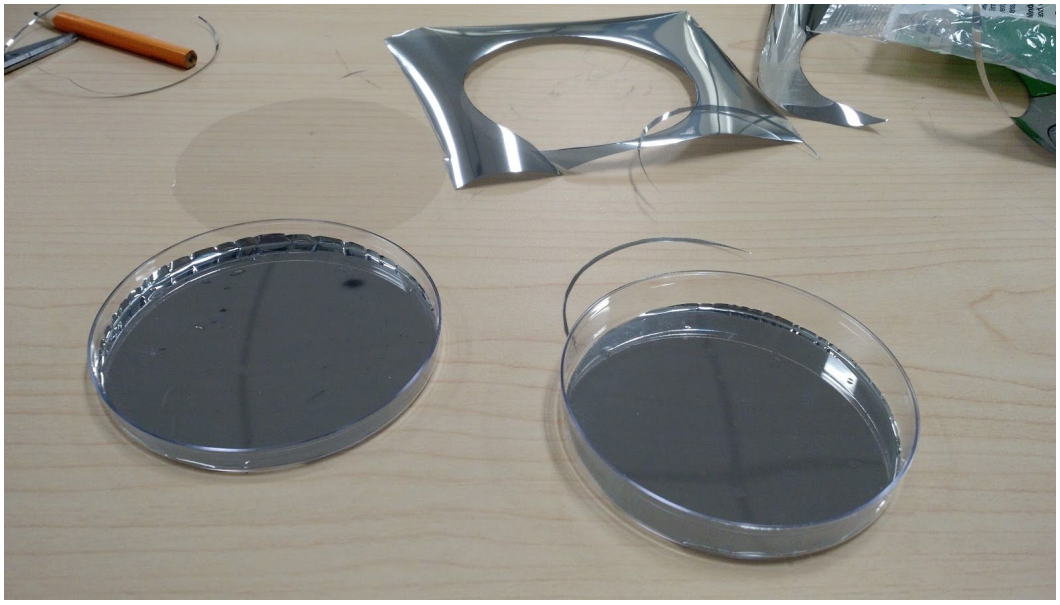
Resistor (560 ohm)

# Making the Kaleidoscope

1. Open your petri dish. Take half of it and use it to trace two circles onto the mirror film.



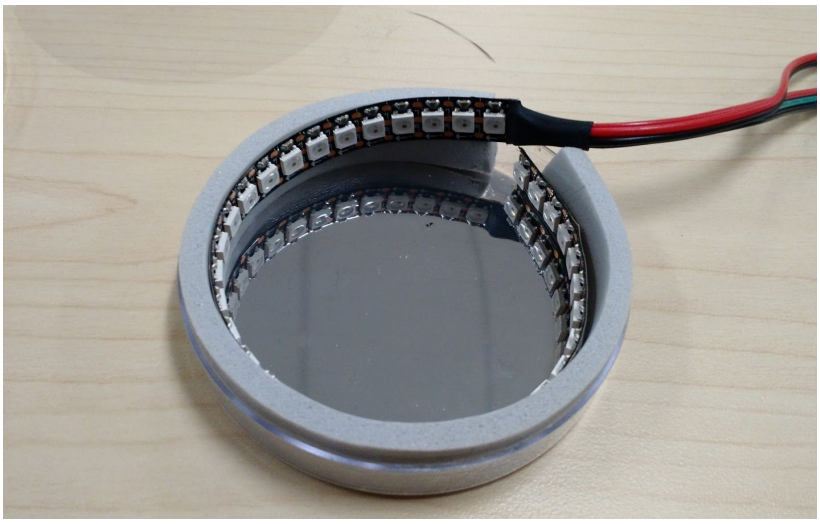
2. Cut out the circles and check the size by placing the circle inside either half of the petri. You'll need to shave a bit off the edges until the circle fits on the bottom of the petri dish. Don't worry if there are a few wrinkles or bubbles—it'll still look fine!
3. If you look closely at the 2 sides of the film, you'll notice that one is slightly more transparent than the other. We'll call this Side A and call the other, more reflective side, Side B. The film also has a backing that can be peeled off. When you're satisfied with the size of the circles, peel off this backing and attach them to each half of the petri dish with Side A on the bottom.



Congrats, you've just made an infinity mirror! An infinity mirror is made of two mirrors that face each other and create a seemingly endless number of smaller and smaller reflections. For example, see Yayoi Kusama's artwork below.

Image of "Infinity Mirrored Room - The Souls of Millions of Light Years Away" from the Helsinki Museum and *The Broad*

5. Now we need to insert a LED light strip inside one half of the mirror/kaleidoscope so that when you look through, you'll see receding rings of lights.
6. First, we'll need some padding for the LED lights. Measure a piece of the neoprene weather stripping that will fit inside the petri dish, leaving a little gap about a centimetre wide (guesstimating is fine!). The strip should have one sticky side and another side covered with paper. Stick the neoprene along the edges and bottom of one half of the petri dish. It should be as close to the bottom as possible.
7. Place your LED strip along the neoprene with the lights facing inwards and the wires coming out through the gap in the foam strip. (You can remove the paper from the back of the LEDs and stick it to the neoprene if you want, but the fit is snug enough that it shouldn't be necessary.)

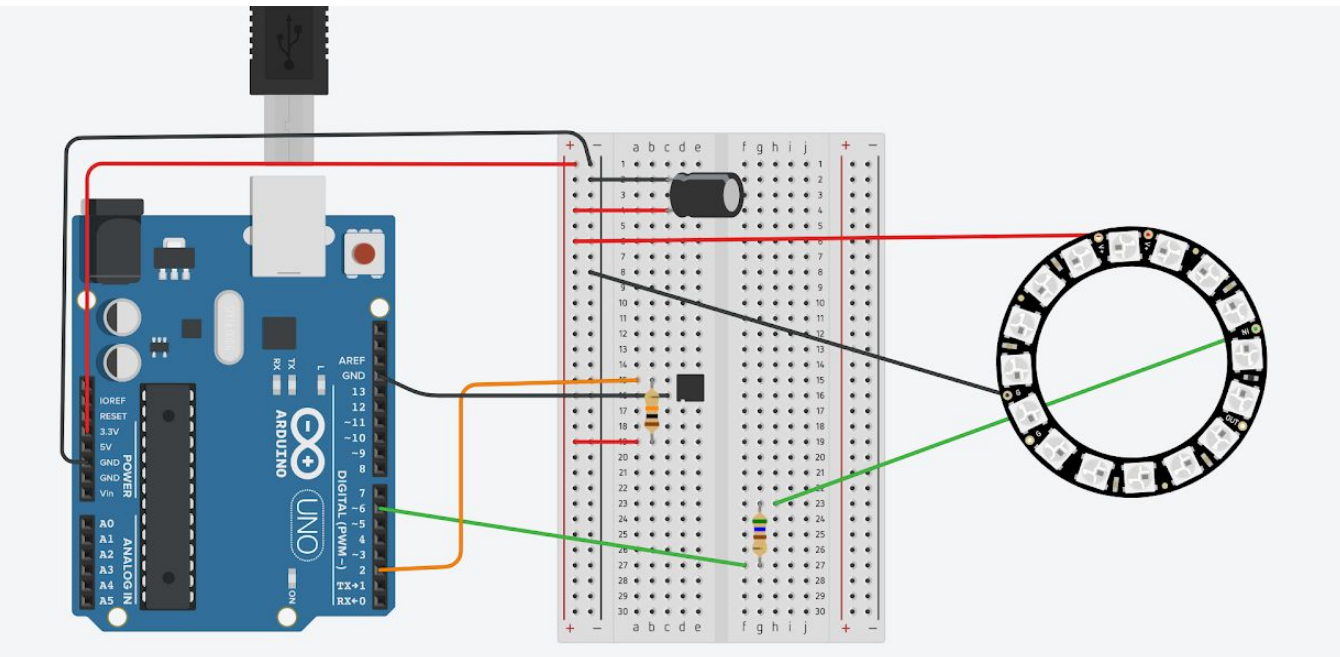


8. Close the two halves together. If you want, you can secure them with some tape.

## Making Connections

9. Now we have to connect the LEDs to the Arduino so we can power them and give them instructions. Let's start with the connections on the Arduino and breadboard. Connect the tilt sensor, the capacitor, the resistor, and the power supply as pictured on the next page. (Remember that the current runs up and down on the left-most 2 columns and the right-most 2 columns. For the remaining grid, current runs across each middle row so that 1a will be connected to 1b, 1c, and so on.)

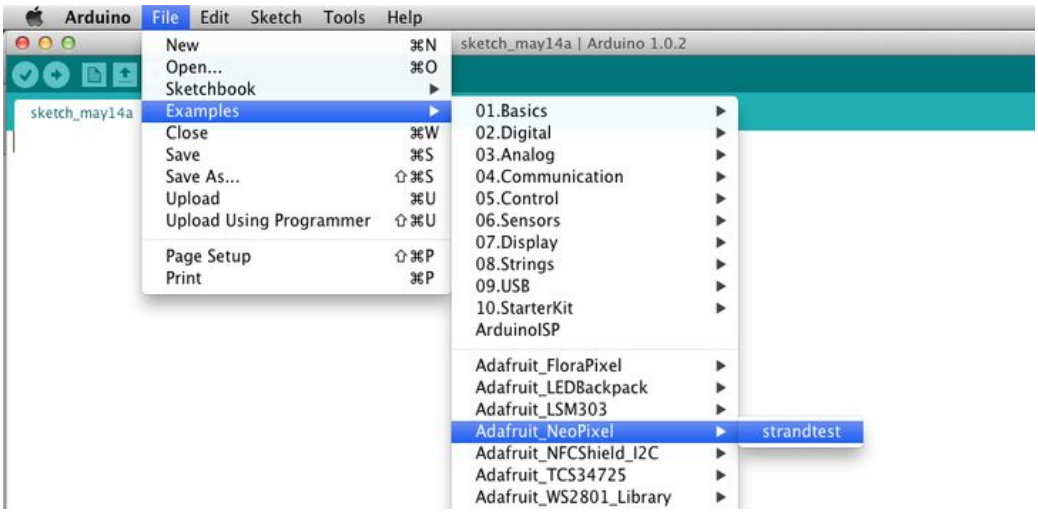
**Notes: The LED strip is represented by the ring below. The capacitor (black barrel) is polarized, which means that direction matters: the negative(-) terminal with the black stripe should be connected to GND or (-) on the breadboard, and the positive (+) terminal should be connected to (+). For the tilt sensor, it doesn't matter which pins you use.**



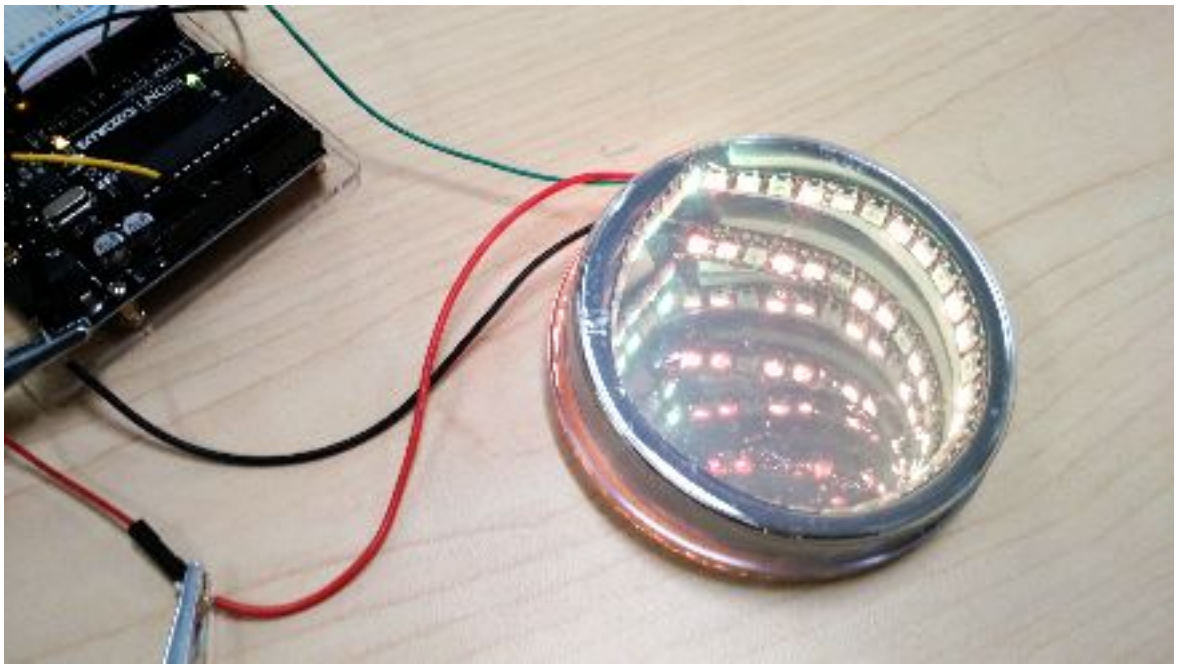

**If the Arduino is connected to your computer, unplug it before doing the next step.**
**It's also good practice to connect GND first.**
These steps protect the Arduino and the LED strips from getting damaged by excess current.

10. Please try to be gentle with the solder joints! The tilt sensor can be a little finnicky so feel free to tape it down to the breadboard if that helps.

11. Now that everything is connected, let's run a test program to make sure everything is set up correctly. Go to **File > Examples > Under "Examples from Custom Libraries," Adafruit Neopixel > strandtest.** A new window should pop up. In that window, click **Upload** and watch the magic begin. If nothing happens, double-check all your connections.




12. Now we'll give Arduino a set of different instructions for communicating with the tilt switch. Create a new sketch by hitting Ctrl + N (Windows) or Cmd + N (Mac) while in Arduino. Then go to tinyurl.com/tiltKal and paste the code into the window. **Make sure to replace *everything* that was in the window.**
13. Click **Upload.** If you get the error message "Problem uploading to board"), make sure that **Tools > Board and Tools > Port** are selected correctly (see "Setup" at the top of the first page).



**How does it work?**

Arduino tracks the orientation of the board & breadboard with the tilt sensor. The sensor has a little ball inside (you can hear it if you shake the sensor close to your ear). The tilt sensor acts like a switch: when you tilt it far enough, Arduino detects this and changes the lights' colour. In this sketch, Arduino changes from blue (default or not tilted) to red (tilted). For more detailed notes, see the "Optional: We Must Go Deeper" section.

## Putting It All Together

15. **Disconnect your computer from the Arduino** for now. For our final step, we have to connect the petri dish to the Arduino and breadboard. Turn the board over so you can see the back. Put the piece of foam onto the back, in the middle of all four pegs. Secure the foam with scotch tape.


16. Using the double-sided tape, stick one side of your petri dish, lights included, onto the foam. Double-check all your connections, then connect the Arduino to your computer to power it.

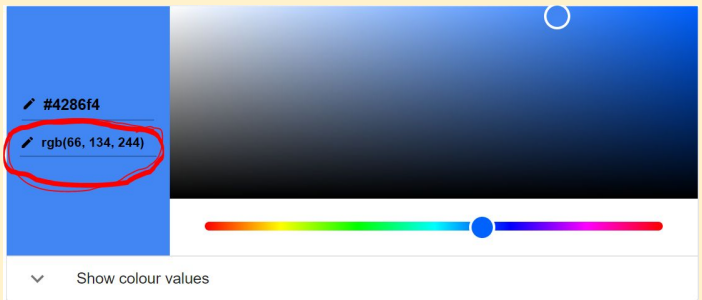
**Congrats, you've done it! Now tilt away to your heart's delight!**

**Bonus: How to Change Colour**

Scroll down to these lines in the code:

```
    if (switchstate == LOW)
    colorWipe(strip.Color(0, 0, 255), 50); // Blue
    else
    colorWipe(strip.Color(255, 0, 0), 50); // Red
  }
```

Inside the brackets after `strip.Color()` are 3 values that determine the strip's colours in this order: (1) red, (2) green, and (3) blue. Open a browser window or tab and google "color picker." This window appears at the top of your search results:




Click on the colour you want and copy/paste the rgb values (x, x, x) into `strip.Color()`. The line just below the "if" determines LED colour when powered on. The line just below "else" determines the colour when you tilt Arduino.

## Optional: Changing the Animation

To do this, we'll use Adafruit's NeoPixel library (a library is a collection of pre-written code that we can play with). You'll notice that we already call the library at the top of our code:

```
#include <Adafruit_NeoPixel.h>
```

For now, the code is set to the `colorWipe()` function, which lights the LEDs in order as if the colour was wiping across a screen.

```
    if (switchstate == LOW)
    colorWipe(strip.Color(0, 0, 255), 50); // Blue when not tilted
    else
    colorWipe(strip.Color(255, 0, 0), 50); // Red when tilted
  }
```

Below that are some other animations that we can call in NeoPixel:

```
//  theaterChase(strip.Color(127, 127, 127), 50); // White
//  theaterChase(strip.Color(127, 0, 0), 50); // Red
//  theaterChase(strip.Color(0, 0, 127), 50); // Blue
```

For example, in `theaterChase()`, we can change the 3 values in `strip.Color()` to change the colour, just like `colorWipe()`. For now, you'll notice that `theaterChase()` has two slashes (//) in front of it. This means that the command is inactive. In programming, this is called "commenting out" code: the two slashes tell Arduino to treat what comes after as comments or notes for the programmer instead of as commands for Arduino.

If we want to run `theaterChase()`, we can cut or copy and paste it into the `if/else` clause like so:

```
    if (switchstate == LOW)
    theaterChase(strip.Color(127, 127, 127), 50); // When not tilted
    else
    theaterChase(strip.Color(0, 0, 127), 50); // When tilted
  }
```

Bonus points: if we wanted to change the code to show colorWipe() when not tilted and theaterChase() when tilted, what could we do?

The other patterns are a bit different because they cycle through a set of colors in order (in this case, each color of the rainbow). Instead, we can set the amount of time it takes per cycle. The value inside the brackets is how long (in milliseconds) to take for each step. You can increase the number to make the animation longer or decrease it to make it shorter.

```
//  rainbow(20);
//  rainbowCycle(20);
//  theaterChaseRainbow(50);
```

# Further Resources and FAQs

**Tutorials**

- Learn.adafruit.com
- Learn.sparkfun.com
- Arduino Tutorials: www.arduino.cc/en/Tutorial/HomePage (good for beginner projects)
- Instructables.com

**Where can I find the components or the board?**

- You can borrow the Arduino, breadboard, and cable from the Music and Media Desk (1st floor MacPherson Library) for up to a week, with the option to renew.
- You can also borrow components from the DSC. Just ask at the desk (immediately to the right when you walk into the DSC).

**I want to buy my own parts or the DSC does not have the parts I need**

- If you know what component(s) you want, we may be able to order them, but you can only borrow the parts from us. Individual components are usually very cheap (less than $1 each).
- Buy locally or in person:
    - Queale Electronics: goo.gl/W46Fz7
    - Lee's Electronics (Vancouver): leeselectronic.com/en/
- Buy online:
    - You can try Amazon.ca. You might also be able to buy parts from Sparkfun or Adafruit (I don't know if they ship to Canada; I've never tried it before).

**Where can I get help for my own projects?**

- I have drop-in hours at the DSC, which are *usually* on Monday (12:30-4:30pm) and Tuesday (1:00-5:00pm) afternoons. I can answer questions by email, but it's typically better to look at the project in front of us. You can also email me (tjychan@uvic.ca) or the DSC (dscommons@uvic.ca) to confirm when I'll be there that week.