



# LED Lightbox

Created by Sam Clippinger



<https://learn.adafruit.com/led-lightbox>

Last updated on 2023-08-29 02:38:18 PM EDT

# Table of Contents

Overview	5
<ul style="list-style-type: none"><li>• TL;DR</li><li>• Skill Level</li></ul>	
Parts List	7
Knives and Drills	10
Wiring and Code	13
<ul style="list-style-type: none"><li>• Wiring</li><li>• Testing</li></ul>	
Final Assembly	16

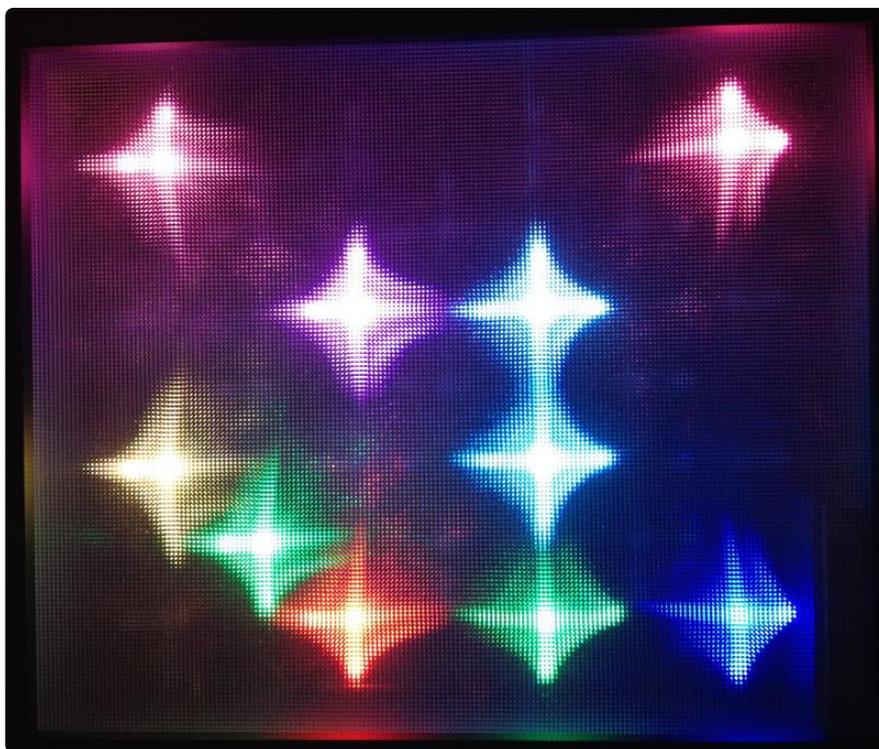


---

## Overview

When I was young, my family had a lightbox as part of our holiday decorations. No one could remember where it came from, probably a garage sale or something. It was extremely simple -- just six old-fashioned "blinker" Christmas tree lights behind a pane of textured acrylic that turned each light into a diamond pattern. The lights blinked on and off at random intervals. Each light was always the same color and there were only six of them, but something about it was... hypnotic.

One year it fell to the floor and broke. No big loss, everyone said, yet I'd never seen another one like it. Since then, I've always wanted to find another one. Years later, when I saw Adafruit's strings of 12mm LED pixels, I knew exactly what I wanted to make...



Of course, this lightbox does much more than the original ever could. The RGB LEDs can create any color. And because it's all controlled by an Arduino, they can create different coordinated patterns instead of just blinking. Read on for descriptions of what the code already does, or you can edit the code to add your own new patterns.

This project's code randomly displays many different patterns. The video below shows specially modified code to demonstrate all the patterns in the following order:

- Blinker bulbs, if old-fashioned blinkers could also fade to different colors, fade to new locations or flicker on/off. Random blinking is the default mode when no other effects are running. The video shows only a 2 second delay between each on/off effect; the unmodified code imposes a random delay.
- Tracers that chase across the field and bounce off the edges. The video first shows one tracer alone, then three simultaneously.
- Rapid scrolling of the entire field. The video shows this happening very suddenly, normally it doesn't follow so closely on the heels of a previous pattern.
- Conway's "Game of Life" (sorta), where each LED lights or fades based on the number of lit neighbors.
- A starfield, which lights all LEDs with a dim background color and random LEDs flare bright white.
- Blinkenleitz (for lack of a better name), where random LEDs flare brightly in random colors against a black background.
- A game of 2048, played randomly. Unfortunately for my ego, random play is nearly as successful as my own.
- Waves of random colors, spreading from a random point and consuming the entire field.
- Undulating/twinkling waves of colors, spreading across the entire field. The effect may remind you of looking up at the light through the surface of rippling water, or a flickering bed of coals in a dying fire, depending on how you like to spend your free time.

## TL;DR

Blinking lights are pretty!

## Skill Level

This project is very easy to assemble. Depending on what wire you use, you may not even need a soldering iron. In fact, the most important part is measuring carefully so everything looks straight -- more on that later!

---

# Parts List

I didn't take separate pictures of each item I used to build the lightbox, but I'll list everything out below.

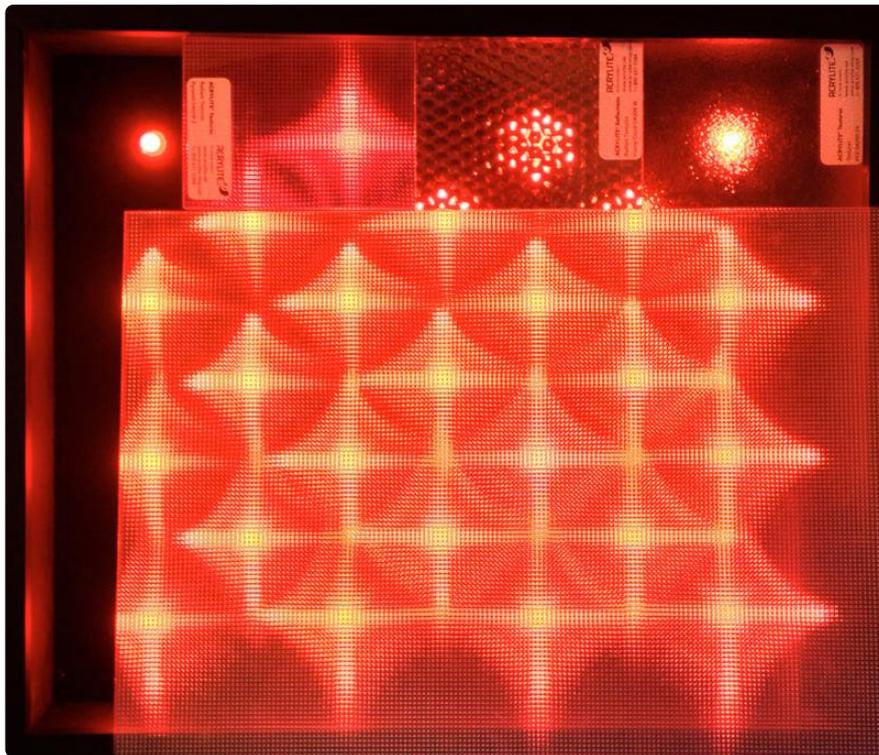


Yellow sticker reads: "Bad planning on your part does not constitute an automatic emergency on my part."

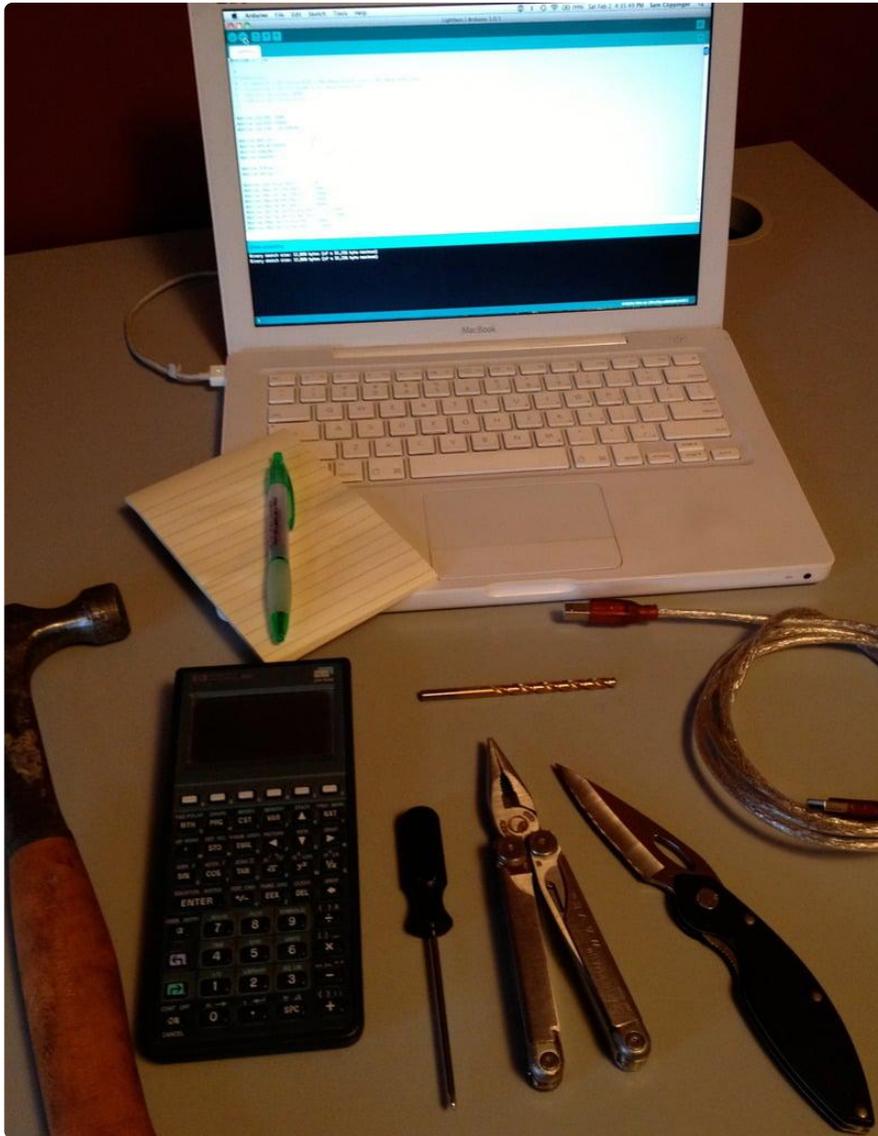
The most important piece is the case. It's what you'll see most, so it should look nice and you should be extra careful not to damage the exterior during construction. If you're handy enough, you can certainly build one yourself. I don't have a woodshop, so I bought a 12"x15" shadowbox from my local hobby store. Whatever you choose, be sure it puts several inches of distance between the glass and the back of the box. If the LEDs are too close the acrylic, their light won't be diffused enough.

Second most important is the patterned acrylic itself. I like the effect created by the diamond pattern, but don't let that stop you from finding something different! I purchased mine from [acrylite-shop.com](http://acrylite-shop.com) () because they laser cut it to the exact size needed. (As of this writing, they appear to offer a transparent honeycomb pattern that they didn't have before. It would probably look very nice.)

Be sure to remove and measure the glass from your shadowbox, don't just measure the visible portion!



Here you can see a few of the different patterns side-by-side. The big sheet at the bottom is the one I actually used. (Notice how the diamonds are quite a bit bigger in the bottom right compared to the top left? That's because the bottom right corner is about 0.3" further away from the LEDs than the top left.) Along the top row, left to right, the small samples are diamond patterned, honeycomb patterned and "stippled". (The small diamond and honeycomb samples also had an iridescent coating that doesn't show up in this photo; acrylite-shop.com calls it "Radiant"). It only looks good in white light; I wouldn't suggest it for this project.)



Full parts list:

- shadowbox
- patterned acrylic panel
- [Arduino Uno](http://adafru.it/50) (<http://adafru.it/50>)
- [12mm Diffused Flat RGB LED Pixels \(strand of 25\)](http://adafru.it/738) (<http://adafru.it/738>)
- [5V 2A switching power supply](http://adafru.it/276) (<http://adafru.it/276>)
- [2.1mm female DC power jack](#) ()
- posterboard or foamboard (black)
- razor knife
- wire cutters
- heat shrink tubing OR electrical tape
- yardstick OR meterstick (depending on where you live)
- pencil
- hot glue gun or epoxy
- wire

- drill with 0.5" and 0.25" bits
- small phillips screwdriver
- calculator and notepad
- computer and USB cable

Pictured but not used:

- hammer (if frustration levels rise too high)

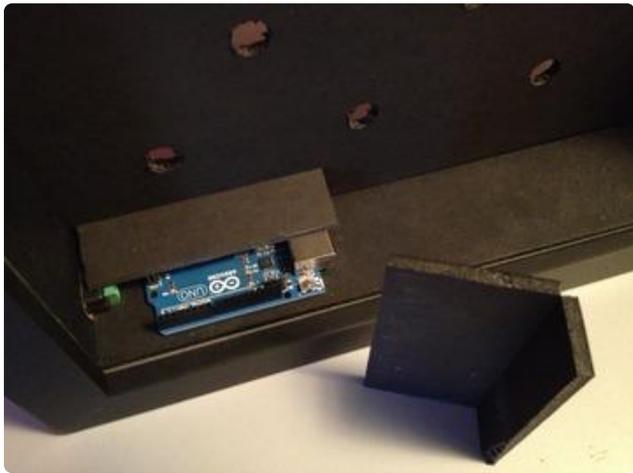
---

## Knives and Drills

The LEDs will be mounted in foamboard (or posterboard) just inside the back panel. If you're using a purchased shadowbox, it will come with a panel for mounting the display items (probably covered in cloth). Remove that panel and cut a piece of foamboard to replace it.



In my shadowbox, the Arduino is too big to stuff between the foamboard and the back panel, but it'll fit nicely along the side. So I cut a hole in the foamboard for it.



The Arduino has some onboard LEDs of the boring non-RGB variety, which don't really fit with the theme of the project. Worse, some of its components are shiny and will reflect the other LEDs. So to hide it from view, I created a little "tent" out of some extra foamboard and glued it in place.



Don't glue the Arduino down. It needs to remain removable so you can plug in the USB cable and upload code.



Use a calculator to determine the spacing you need and draw out the grid. For 25 pixels, you'll need 7 lines horizontally and vertically. The LEDs will go on every other intersection.



Mark the points you're going to cut, otherwise you'll cut the wrong place when you're almost done. Having to start over will reduce your enthusiasm. (Trust me.)



Cut the holes with your razor knife. I found it was easiest to cut square holes with the knife, then round them out by using a 0.5" drill bit as a hand tool (not in the drill).

Take extra special care to measure very carefully when cutting these holes! A crooked line or sloppy cut will put an LED (slightly) out of line and will be VERY noticeable in the end product.



Create a hole for the power jack by drilling from the inside. The 2.1mm jack is bigger near the posts than it is at the tip, so I used the drill to make the inside of the hole wider than the outside.

Don't glue it in place yet -- wait until you have all the wires connected!

---

## Wiring and Code

The Arduino code for this project can be downloaded here <https://github.com/samclippinger/lightbox> ()

Almost all of the tweakable values are set using `#define` commands near the top. For example, if you want to change the chance of a starfield effect appearing, look for `STARFIELD_CHANCE_PERCENT`. The comments should help explain what each value does.

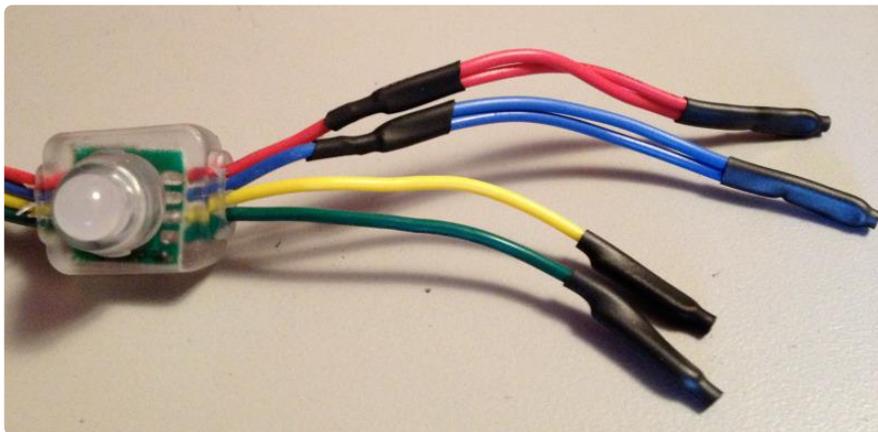
# Wiring

Very little wiring is needed to make this project work. To start, the plugs on the ends of the LED strand are useless here, so go ahead and cut them off.

With only one of strand of LEDs, the "output" end isn't needed. I just folded the wires over and covered them in heatshrink. (I didn't want them to accidentally touch and short out some day.) If you want to make this project bigger and use more LEDs, you'll need to connect the strings and (if you have more than 50 LEDs) provide additional power mid-strand.

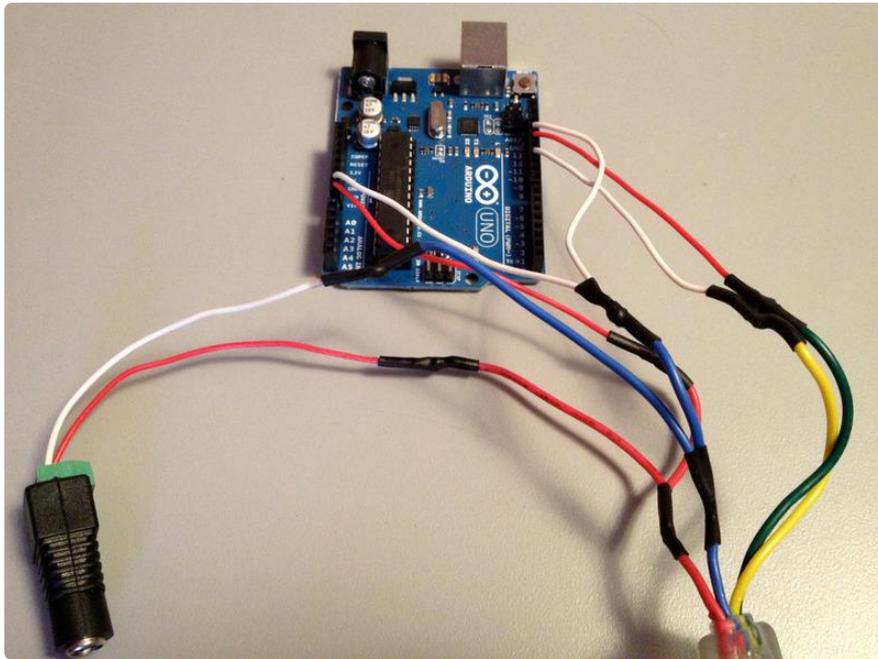
[For more details on wiring up these pixels, check out the main guide for them at https://learn.adafruit.com/12mm-led-pixels \(\)](https://learn.adafruit.com/12mm-led-pixels)

Before cutting any wires, carefully check your LED strand's markings to be very sure which end is "input" and which is "output".



On the "input" end of the strand, the "serial data" and "serial clock" lines will need to connect to the Arduino. The wires in the LED strand are multi-strand, which makes them nicely flexible but very difficult to mash into a breadboard (or an Arduino). I solved problem this by splicing them to a short piece of 20 gauge solid-core copper wire, which fits easily into an Arduino hole. After twisting the wires together, I dribbled a bit of solder on the connection to hold it. As a last step, I covered each junction with heatshrink so no one can see how bad I am at soldering...

The 5V and ground lines were already each split into two lines, but I decided I needed three: two to the Arduino and one to the power jack. Again, heatshrink makes it look very professional.



On my LED strand:

- Red is the 5V line
- Blue is the ground line
- Yellow is "serial data"
- Green is "serial clock"

[For more details on wiring up these pixels, check out the main guide for them at https://learn.adafruit.com/12mm-led-pixels \(\)](https://learn.adafruit.com/12mm-led-pixels)

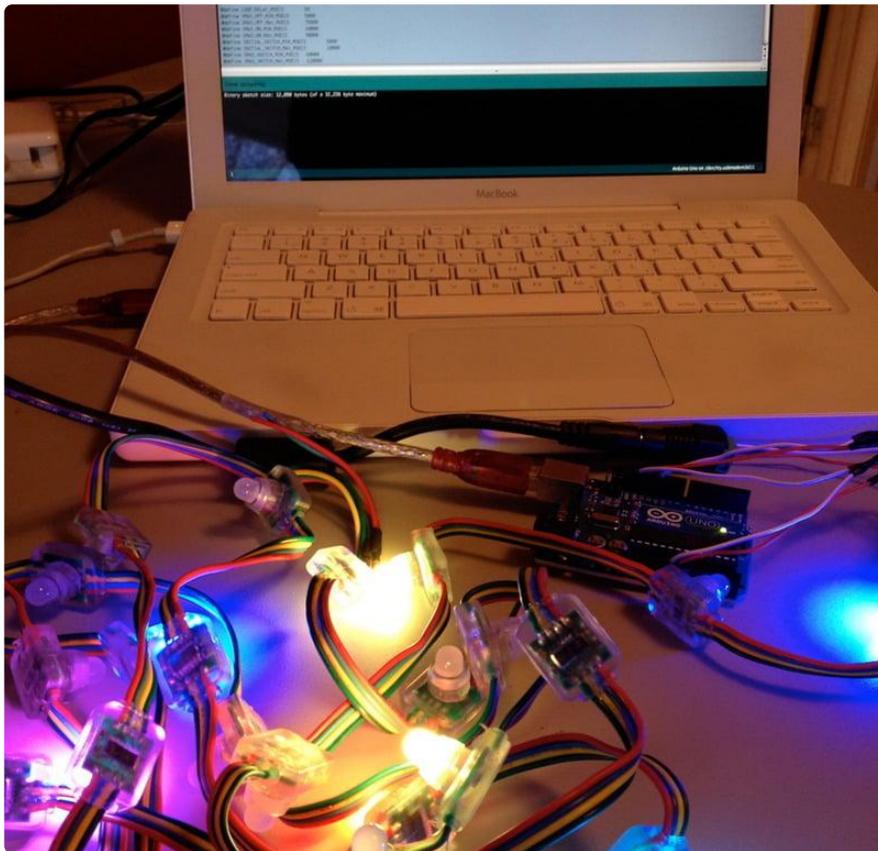
Check your LEDs carefully to figure out your own wire colors -- they may be different from mine. Connecting 5V to the wrong line could destroy your LED strand!

- Connect the 5V line to both the positive post on the power jack and the "Vin" hole on the Arduino.
- Connect the ground line to the negative post on the power jack and the two "GN D" holes on the Arduino.
- Connect the "serial data input" line to the Arduino's digital pin 11
- Connect the "serial clock input" line to the Arduino's digital pin 13

## Testing

Before you go any further, plug in the 5V power. Then connect the Arduino to your computer with a USB cable. (Don't plug in the USB first, or the Arduino will try to power the LEDs from the USB power, which could damage the Arduino.)

Use your computer to upload this project's code to the Arduino. If everything is wired correctly, you should see some of the LEDs light up and start to blink and/or fade between colors. The patterns won't be distinguishable while the LEDs are in a pile, but that's OK. If they light up, your wiring is correct.

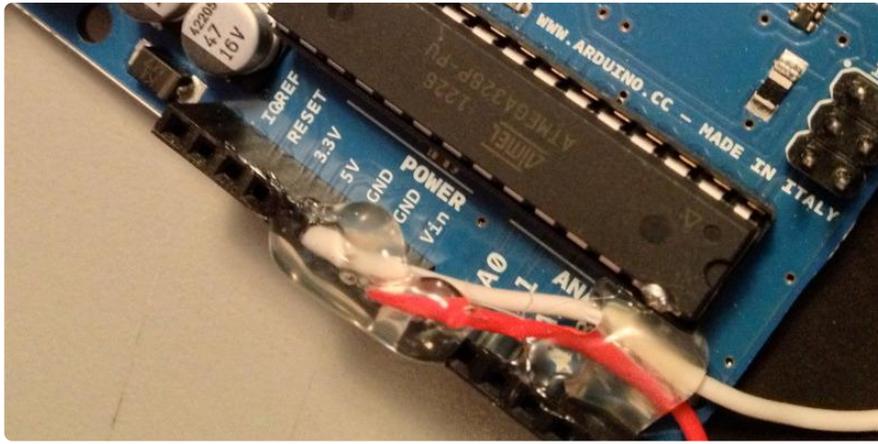


---

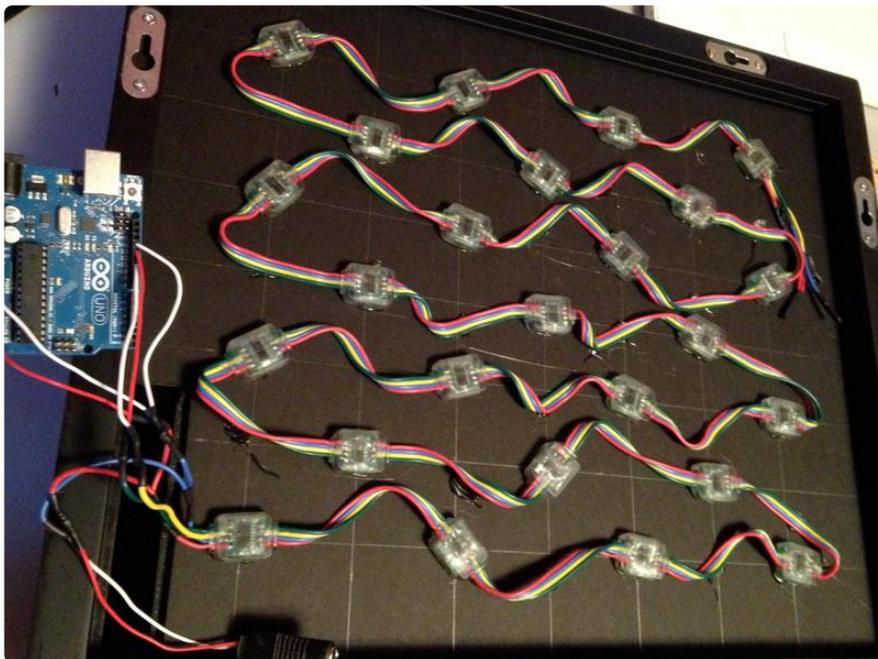
## Final Assembly

Nearly done!

The Arduino is going to remain unglued so it'll be easy to remove and upload new code. But since you don't want the connections to work free, it's not a bad idea to secure them with some hot glue.



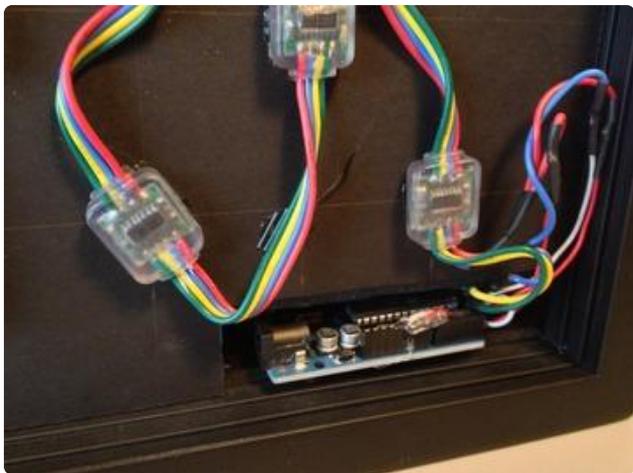
Next, glue the LEDs into the foamboard. Be sure to follow the same layout as in the picture below! If they're installed in a different order, the animations won't work correctly.





Install the acrylic panel, glue the power jack in place and tuck the Arduino into its little cave.

Lastly, close up the back panel, apply a sticker and you're done!



Hang it on the wall, put it on your desk or give it as a gift! Every time you have more than a minute to watch it, you'll end up hypnotized by it.