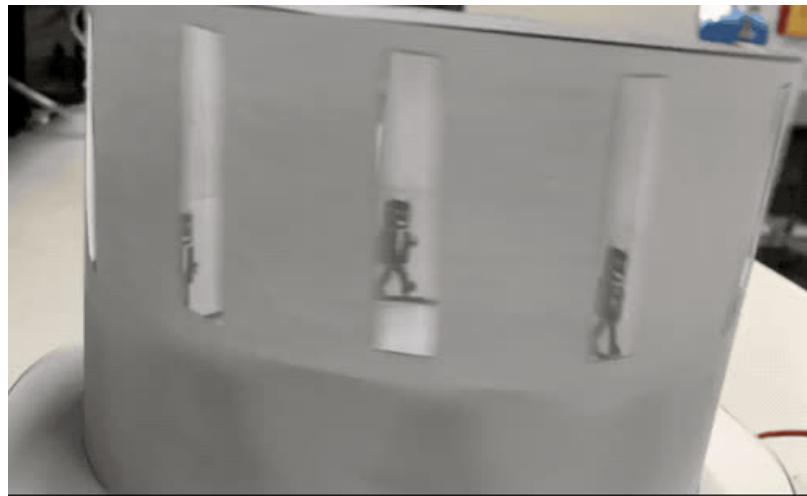


## Paper Craft Zoetrope with Crickit

Created by Dano Wall



Last updated on 2018-09-06 02:07:37 PM UTC

## Guide Contents

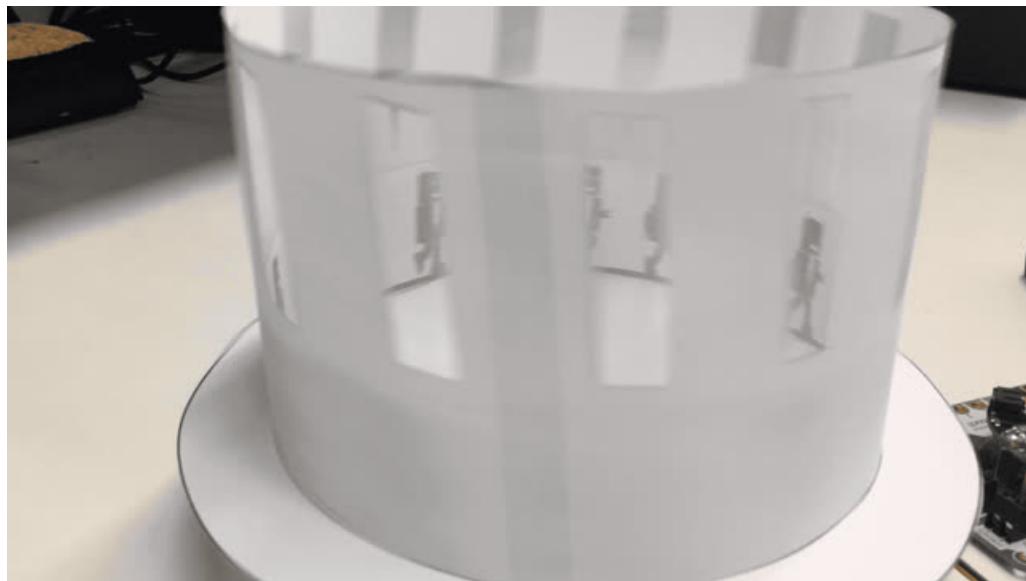
Guide Contents	2
Overview	3
Household Materials	3
The Electronics	4
Print & Prep	7
Print the parts	7
Zoetrope Assembly	13
Making Donuts	13
The Wall	14
Bring it all together	16
Connect to CRICKIT	19
Motor holder (optional)	20
Make the holes	20
Mount the motor	22
Add a washer	25
Coding with CircuitPython	28
Getting Familiar	28
Let's upload some code!	28
Sample Code	28
Making Changes	29
Experiment!	30
Custom Animations	31
Exploring further	33

## Overview

---

A [zoetrope](https://adafruit.it/BPZ) (<https://adafruit.it/BPZ>) is a device that makes a series of images to appear animated. It does this by taking advantage of a phenomenon called [Persistence of Vision](https://adafruit.it/xCS) (<https://adafruit.it/xCS>) (POV).

When you look through the slits in the wall of a zoetrope, your visual system retains the image it's presented with for a fraction of a second after the next image has moved into view, so as it spins your brain merges the previous image with the next one, creating the appearance of fluid motion. This is same principle upon which [flip books, CRT televisions, and film movies work](https://adafruit.it/BSu) (<https://adafruit.it/BSu>).



One magical aspect of zoetropes is how difficult it can be to capture the effect on camera. This is due to something called the rolling shutter effect ([this video by Destin Sandlin of Smarter Every Day](https://adafruit.it/BP-) (<https://adafruit.it/BP->) provides a great explanation of the phenomenon). Because of rolling shutter, the zoetrope is best experienced in real life, as the mystical, defiant object that it is.

In this guide we will build a low-cost zoetrope that you can use to make and view your own animated sequences. The tools and materials used will be kept as simple and inexpensive as possible, so as to make it accessible and quick to put together.

## Household Materials

---

In addition to the electronics you will need to gather some everyday materials from around the home.

- Printer & paper (3 sheets)
- Tape
- Scissors
- X-acto knife
- Piece of scrap cardboard (optional)
- Metal washer (optional)
- A hole punch (optional)



## The Electronics

---

### 1x [Circuit Playground Express](#)

Circuit Playground Express is the perfect introduction to electronics and programming

[ADD TO CART](#)

### 1x [Adafruit CRICKIT for Circuit Playground Express](#)

Creative Robotics and Interactive Construction Kit is an add-on to our popular Circuit Playground Express that lets you #MakeRobotFriend using CircuitPython

[OUT OF STOCK](#)

### 1x [USB cable - A/MicroB - 3ft](#)

Standard A to micro-B USB cable

[ADD TO CART](#)

### 1x [Alkaline AA batteries - 3 pack](#)

These batteries are good quality at a good price, and work fantastic with any of the kits or projects that use AAs.

[ADD TO CART](#)

### 1x [3 x AA Battery Holder](#)

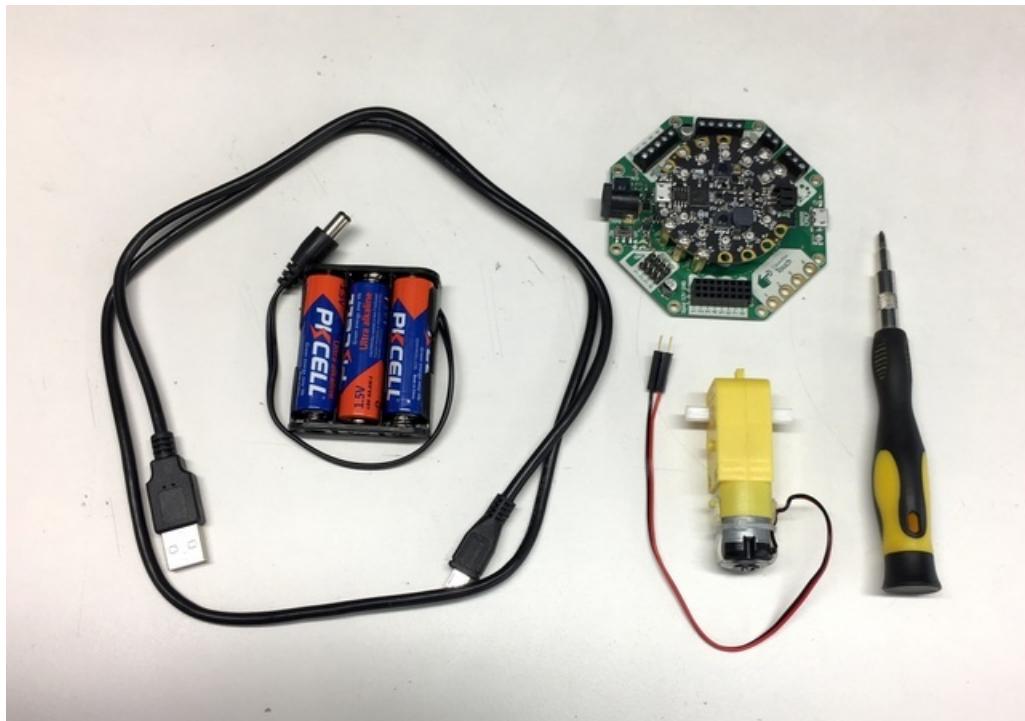
Battery holder with On/Off Switch, DC plug

[ADD TO CART](#)

### 1x [DC Gearbox Motor - "TT Motor" - 200RPM - 3 to 6VDC](#)

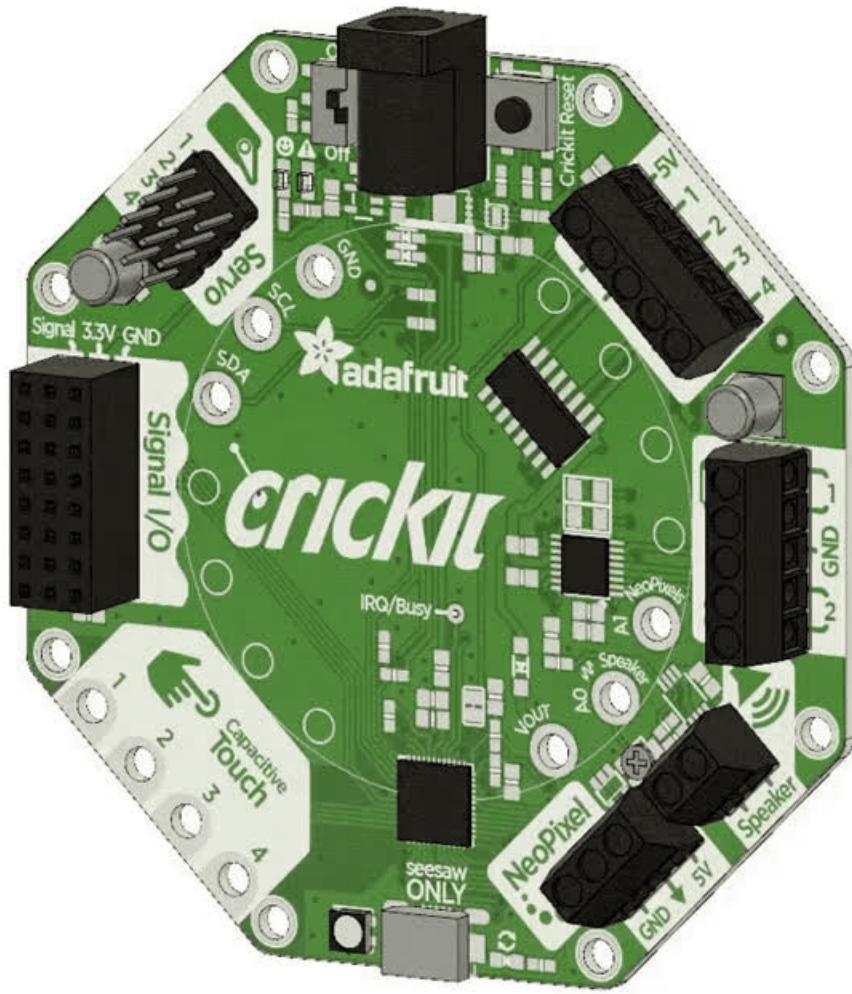
TT DC Gearbox Motor with a gear ratio of 1:48 and 2 x 200mm wires with breadboard-friendly 0.1" male connectors

[ADD TO CART](#)



If your Circuit Playground Express and CRICKIT aren't already connected, now is the time to do that.

The animation below demonstrates how the two become one.



## Print & Prep

### Print the parts

Print out a copy of one of each of the PDFs below.

<https://adafru.it/BPm>

<https://adafru.it/BPm>

<https://adafru.it/BPn>

<https://adafru.it/BPn>

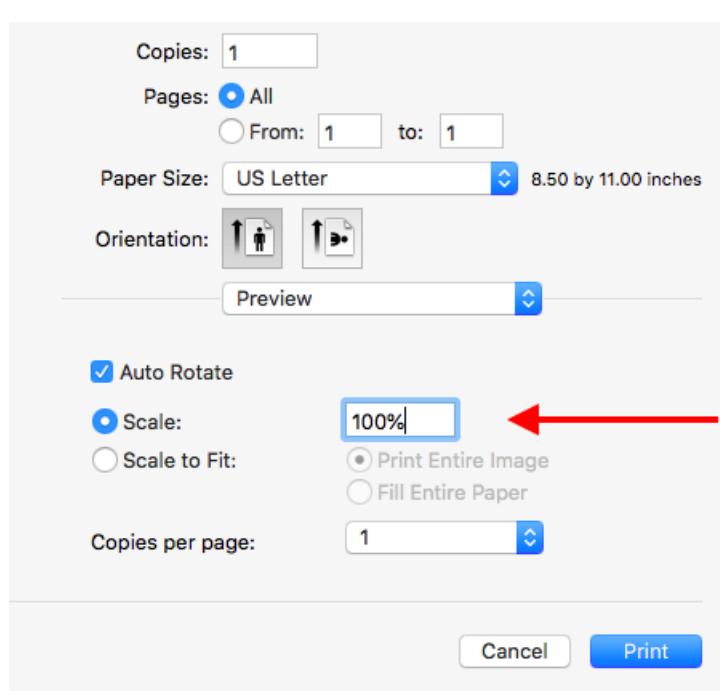
<https://adafru.it/BPo>

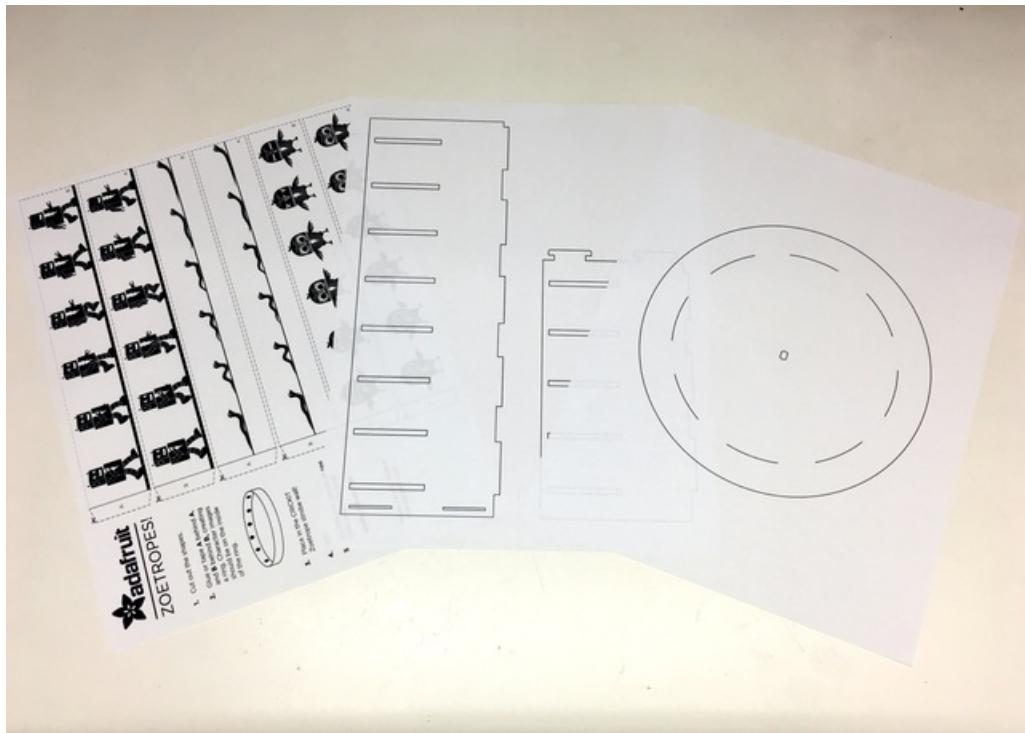
<https://adafru.it/BPo>

These are designed to be printed on 8.5" x 11" (letter) paper.

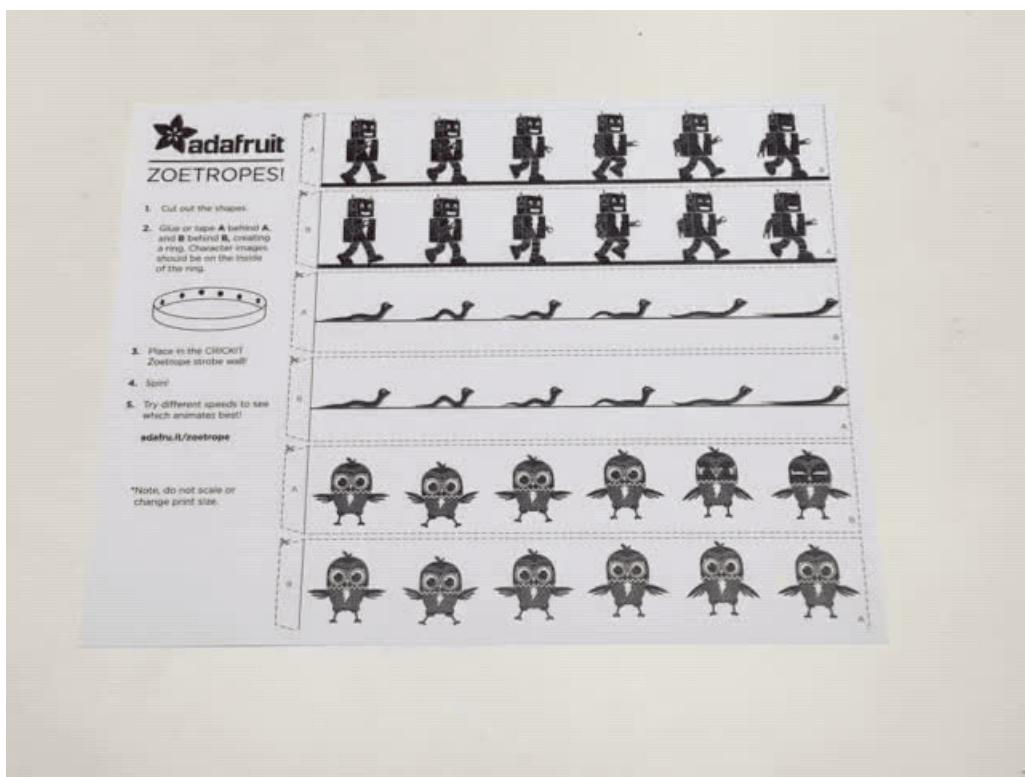
Because these parts are all meant to fit together, it's important that they're printed at the correct scale. Be careful when sending them to the printer that they are all scaled correctly.

Make sure each page is printed at 100% scale!



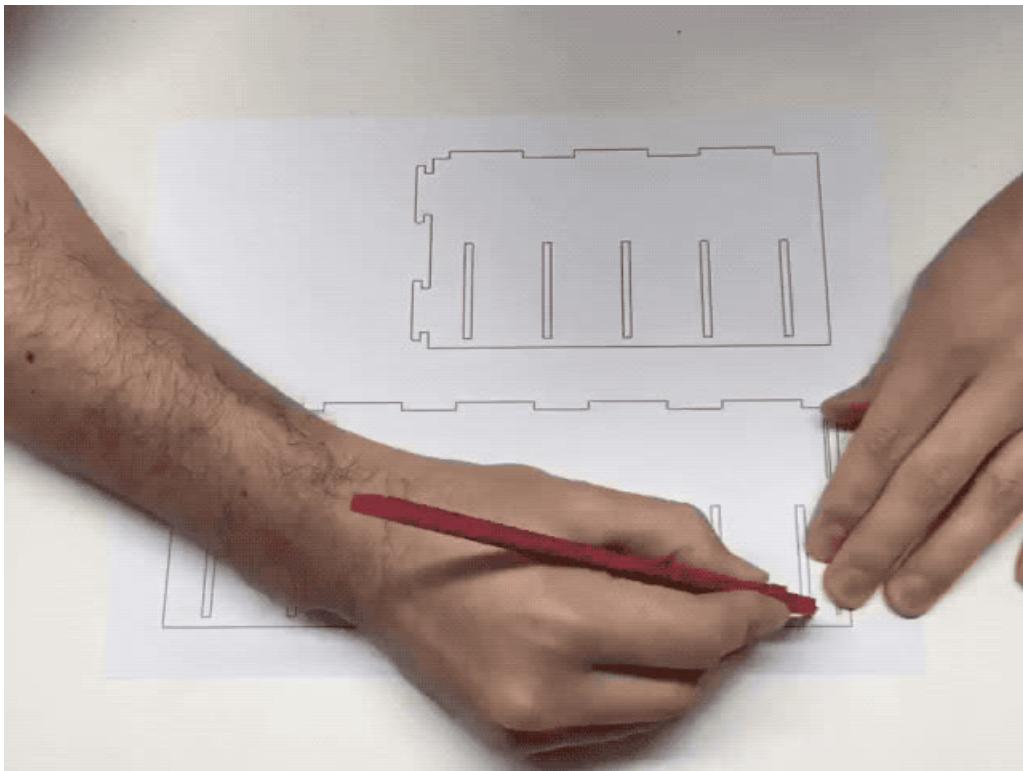


Cut out the three zoetrope designs.



The wall of the zoetrope will be formed from the two rectangular shapes.

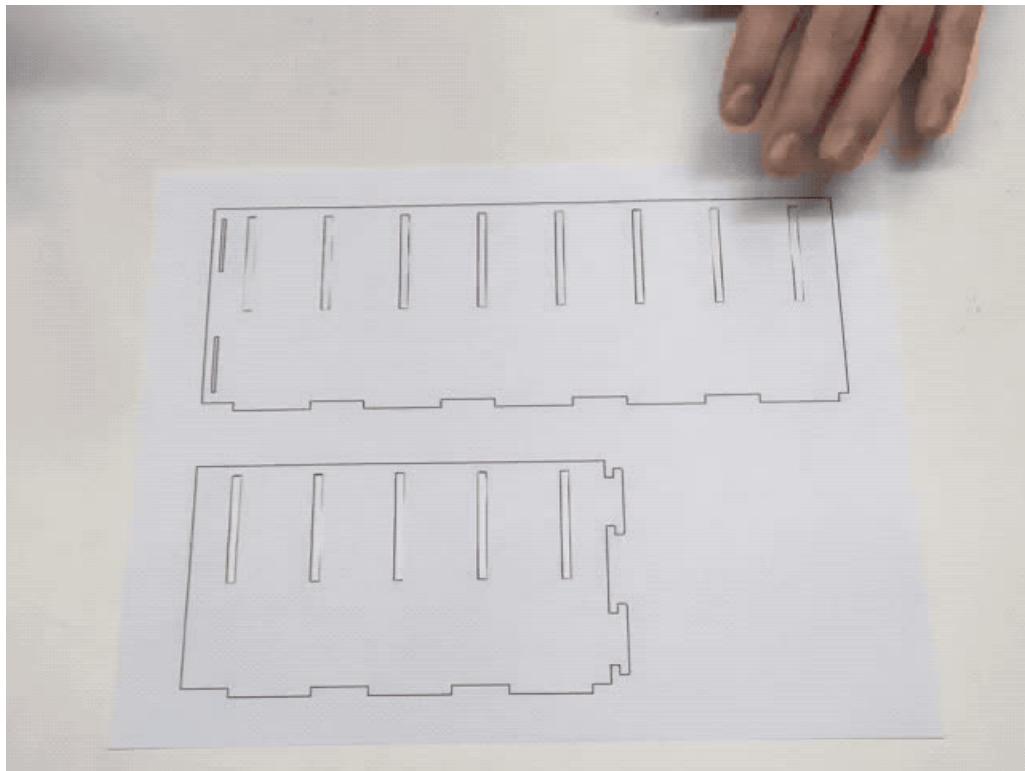
Start by cutting out the "windows" with a blade.



Be careful using a hobby knife, they are sharp! Have someone help you if you need help.

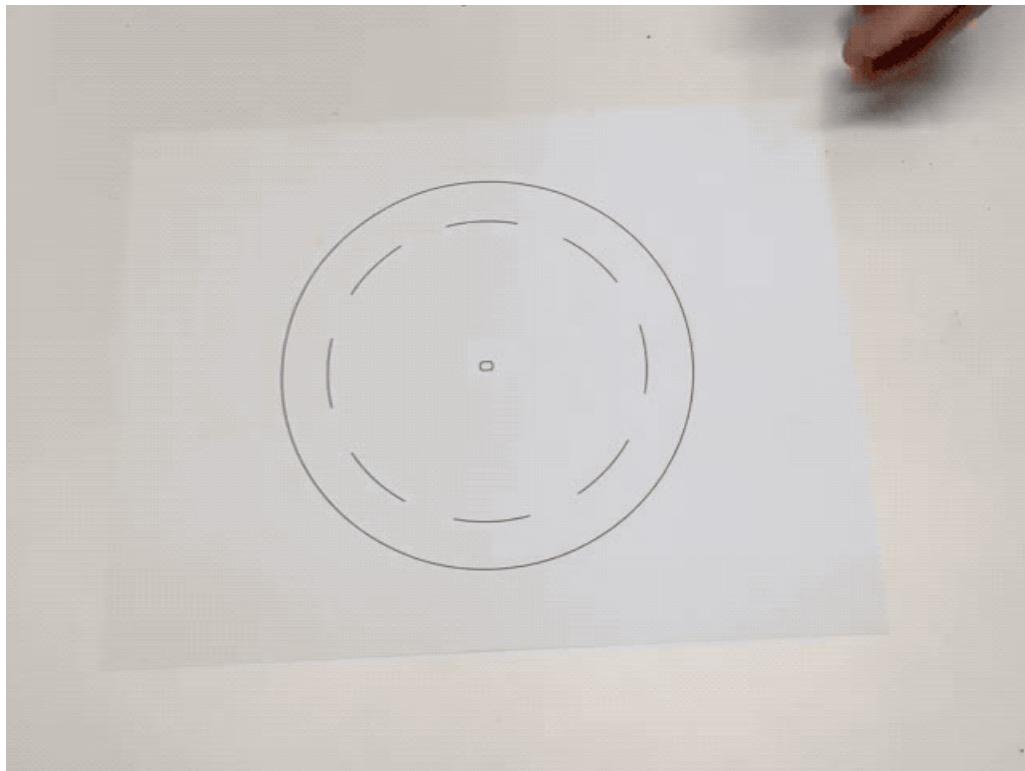
Cut around the outside edge using scissors. A knife is useful for cutting out the small details.

If you don't have an X-acto knife, the 3D printed handle used here can be found on [Thingiverse \(https://adafru.it/BPp\)](https://adafru.it/BPp) and pairs well with standard utility knife blades to make an excellent hobby knife.

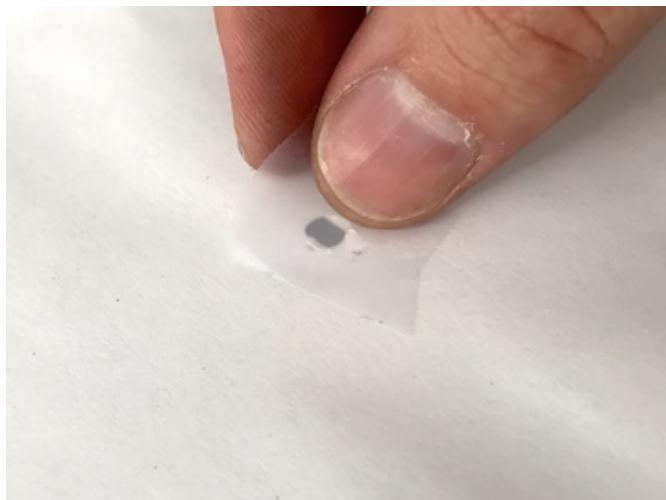
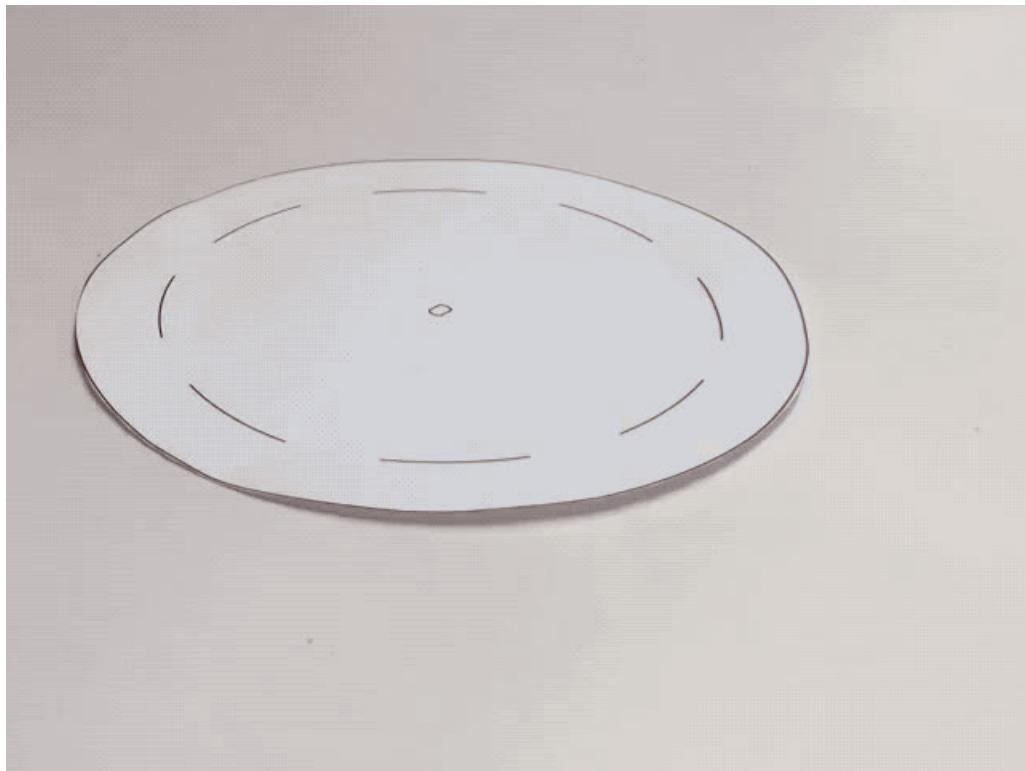


The circular part will become the base for the zoetrope.

Cut around its outline, then use a knife to slice along the inner dashed lines.



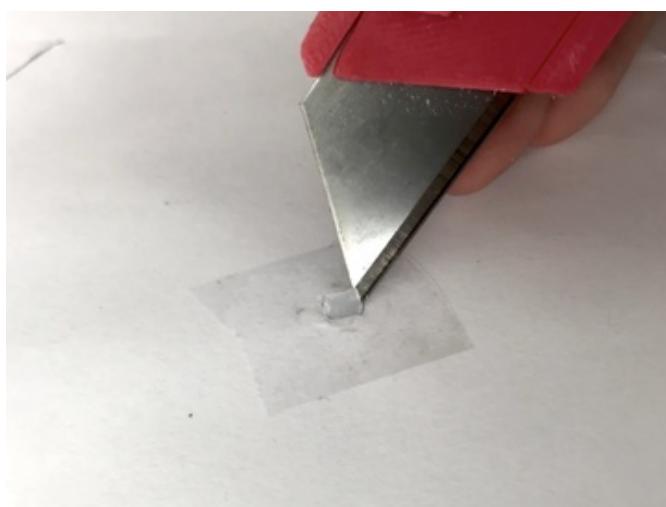
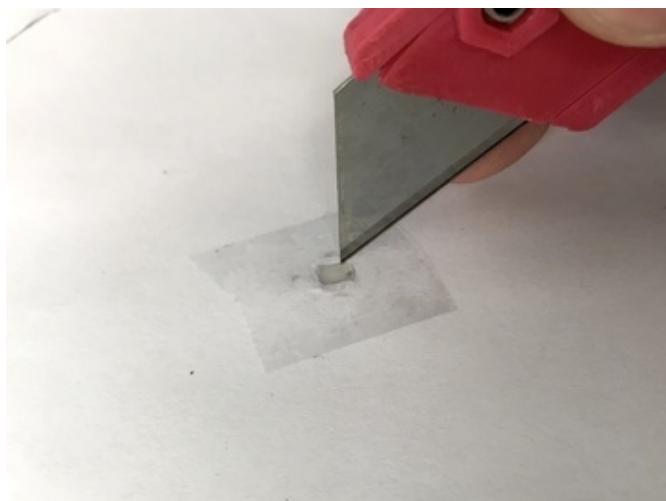
The oval in the center will fit snugly on the DC motor mount.



It's a good idea at this point to reinforce the mounting hole for the motor.

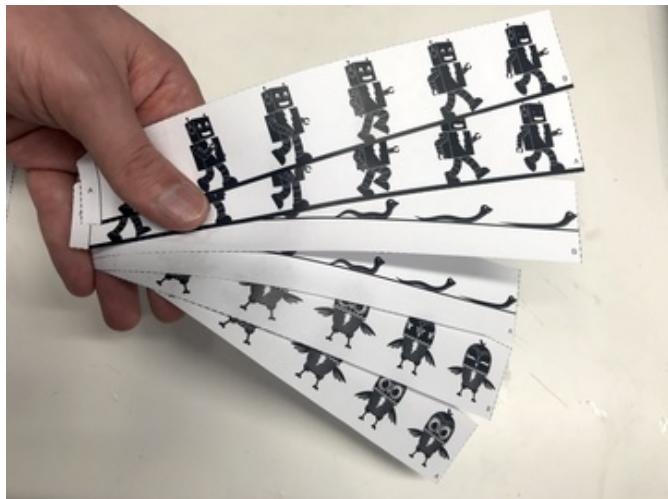
Do this by placing a piece of tape over the top and bottom of the hole before cutting it out.

This will help the zoetrope spin for a long time without wearing out.



## Zoetrope Assembly

Once you've printed and cut the parts for the Zoetrope, it's time to put it all together.

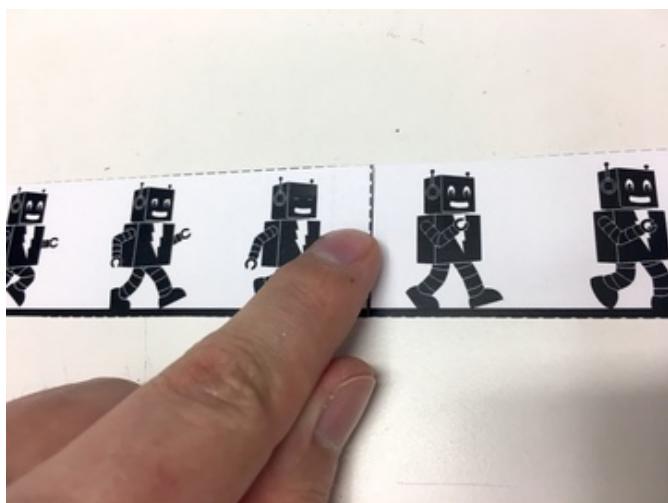
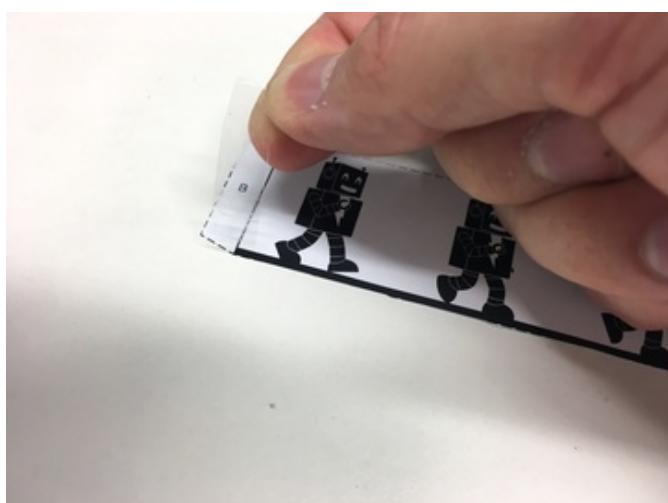


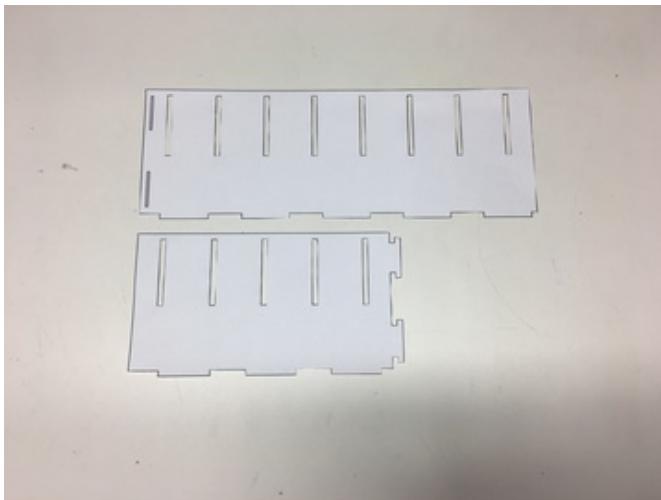
### Making Donuts

These strips of paper will be made into animation loops, which will sit inside our zoetrope.

Start by taping the tab marked "A" to the end of the strip marked "A".

Complete the loop by taping the "B" tab to the end of the strip marked "B".





## The Wall

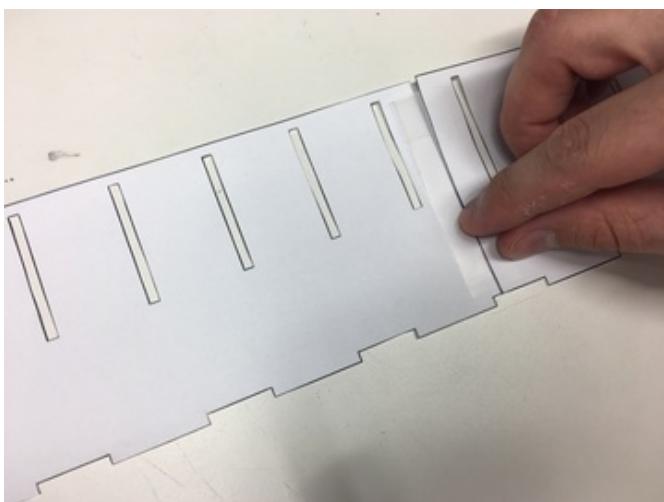
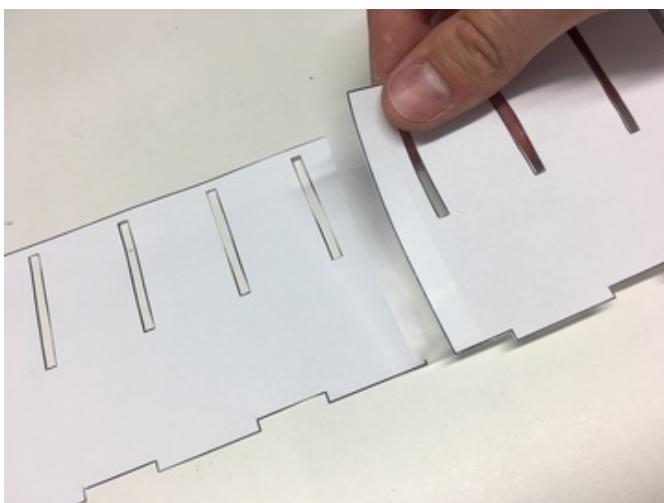
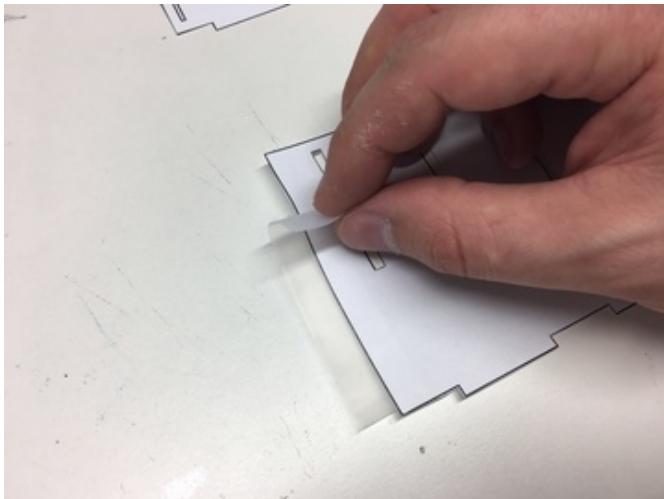
Find the two pieces that will form the wall of the zoetrope.

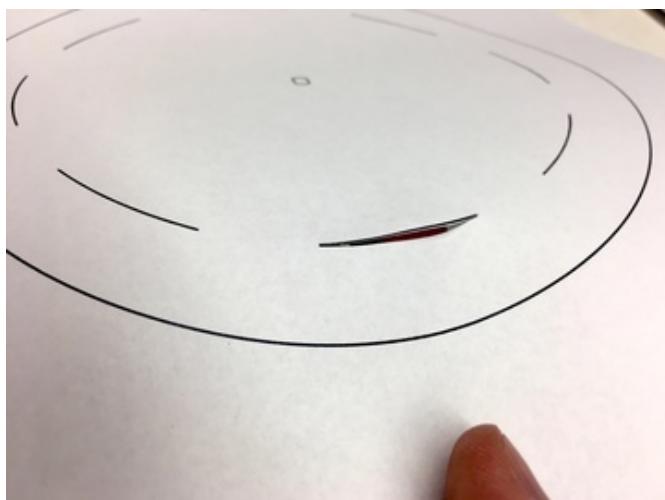
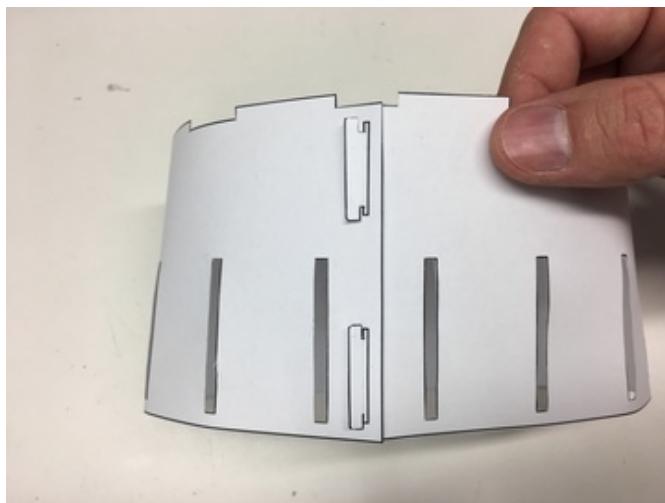
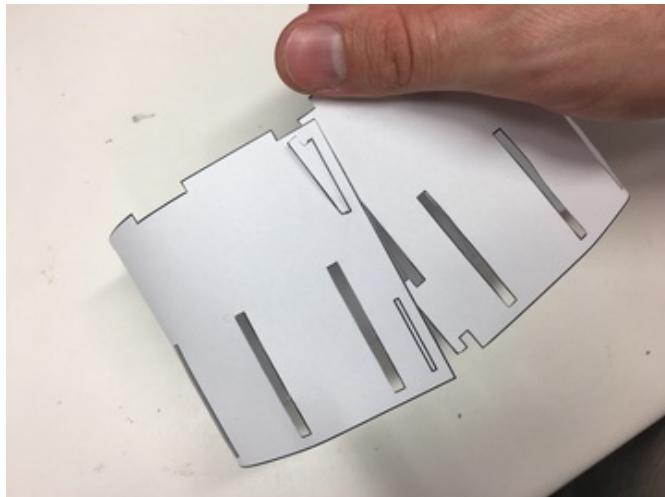
Attach the two straight ends with tape.

For best results, make sure to add tape on **both** sides of the seam.

The two tabs at one end will fit into the slits cut in the other end.

Carefully notch these tabs into place one at a time.



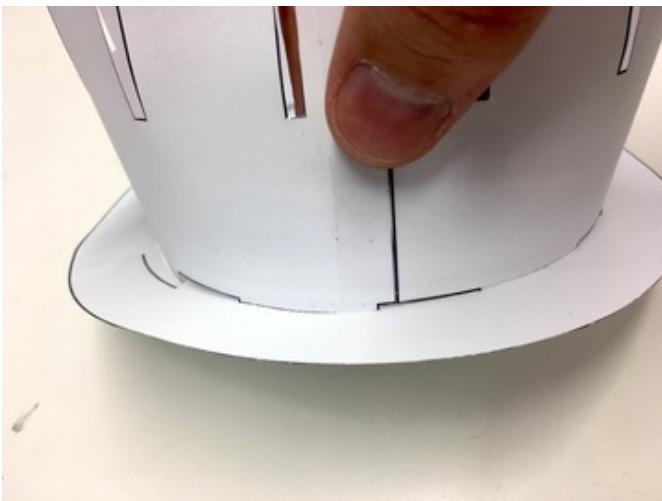


## Bring it all together

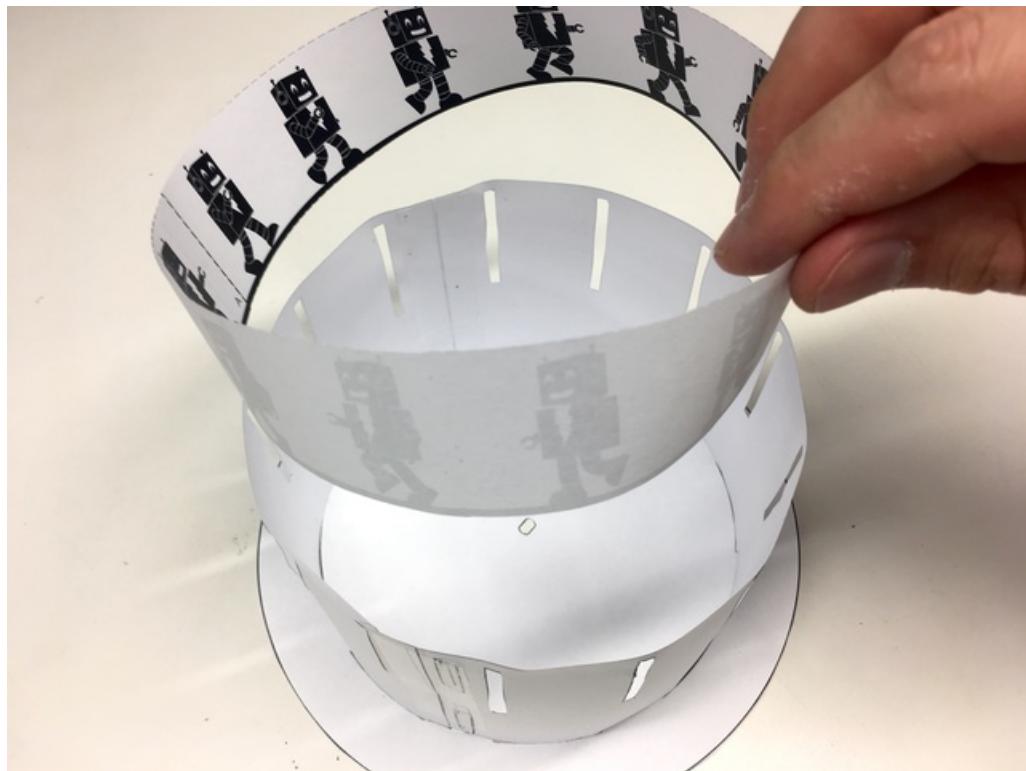
Align the tabs along the bottom of the wall with the slits in the zoetrope base.

Gently poke these through the base, tugging them into place one by one.

The fit should be snug enough that no tape or glue will be needed to hold them together (you can add either if you want to make the bond more permanent).



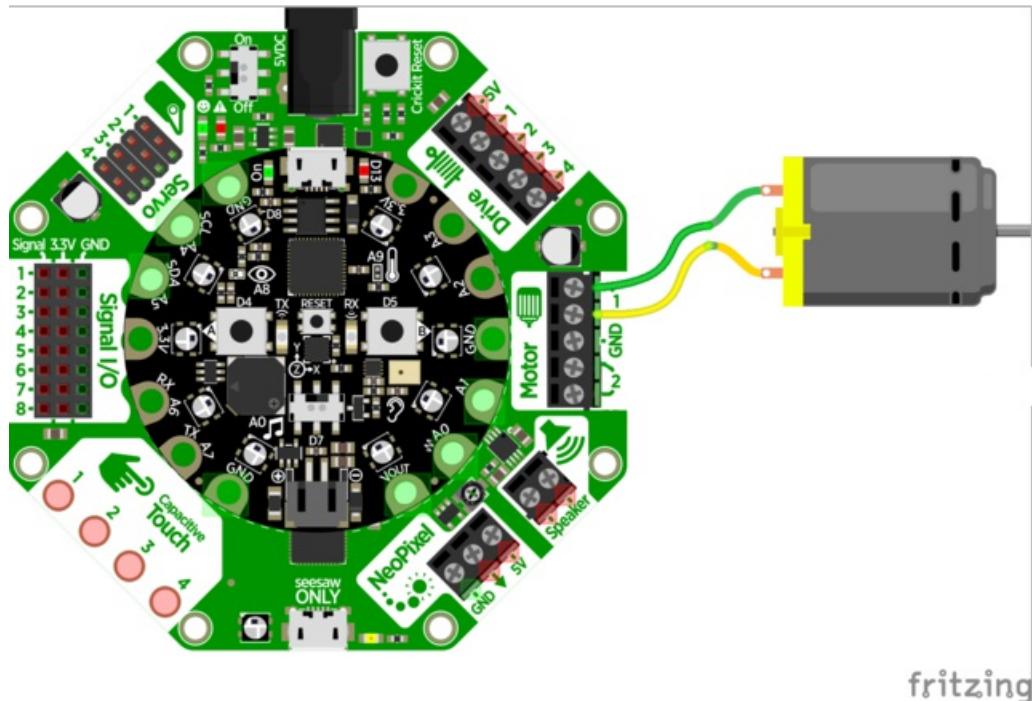
Now you can take your animation strip and drop it into the center of your zoetrope!

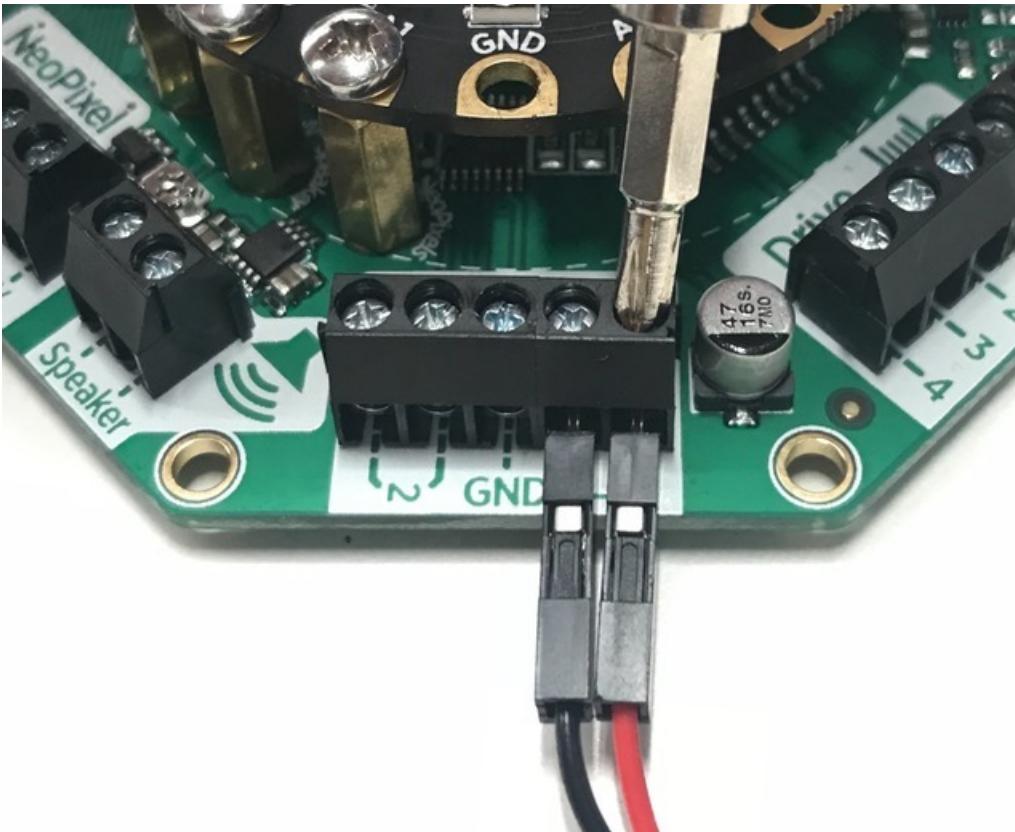


## Connect to CRICKIT

At this point you should be ready to add some electronics to the mix.

The first, only, and most essential step is to connect the motor to the screw terminals on the CRICKIT board.





## Motor holder (optional)

Holding the motor in your hand can quickly become tiresome, so optionally, you can create a small cardboard mount to hold the motor for you.

The PDF below contains the 2D geometry of the [TT motor](https://adafru.it/BzE) (<https://adafru.it/BzE>) used in this project. Print it out and use it to mark the placement of the motor hub and mounting holes on a piece of cardboard.

<https://adafru.it/BQ4>

<https://adafru.it/BQ4>

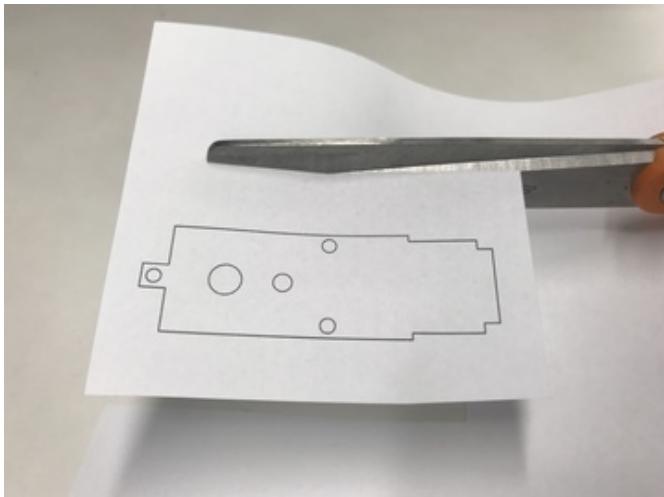


## Make the holes

Find a small piece of scrap cardboard, about 2" x 8" (a piece of a shipping box - like the ones from Adafruit - works great).

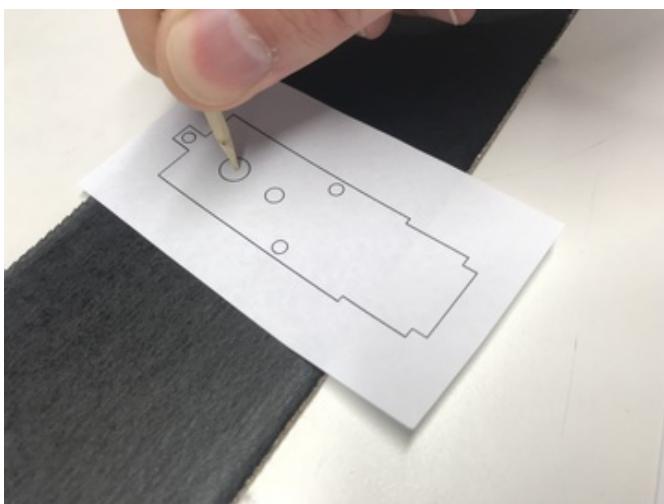
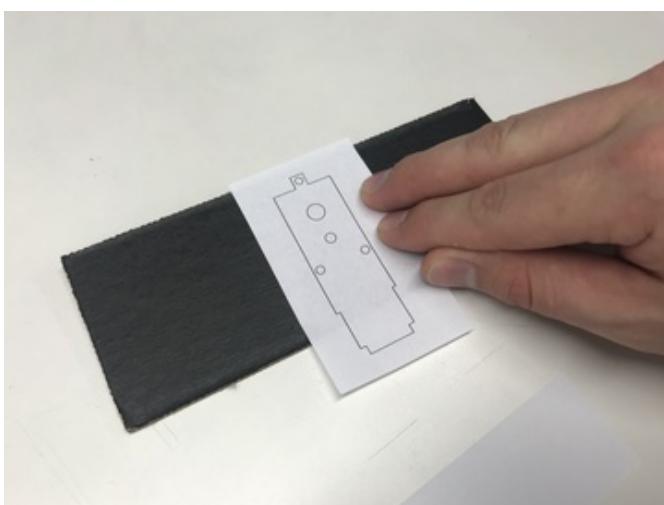
Print the motor outline at 100% scale on a normal 8.5"x11" piece of paper. Cut a rectangle around the TT motor outline.

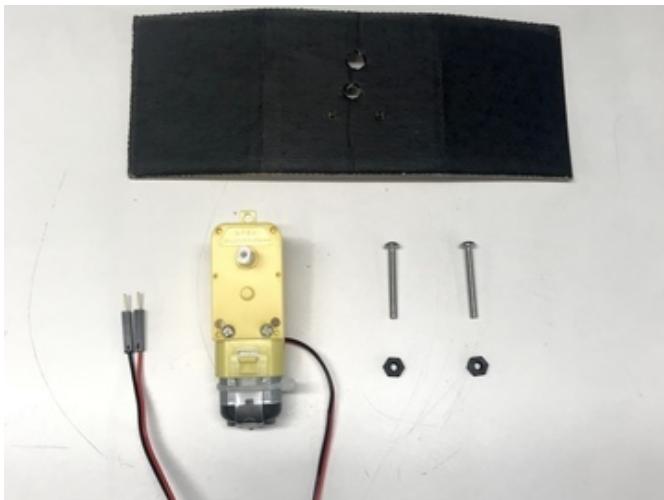
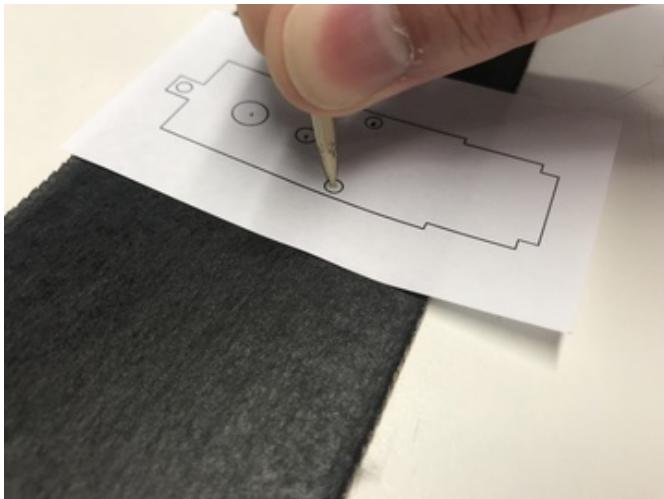
Use a pen or skewer to mark the locations of the holes on a piece of cardboard.



After marking their locations, widen these holes using a pen or pencil tip.

You are now ready to mount the motor!





## Mount the motor

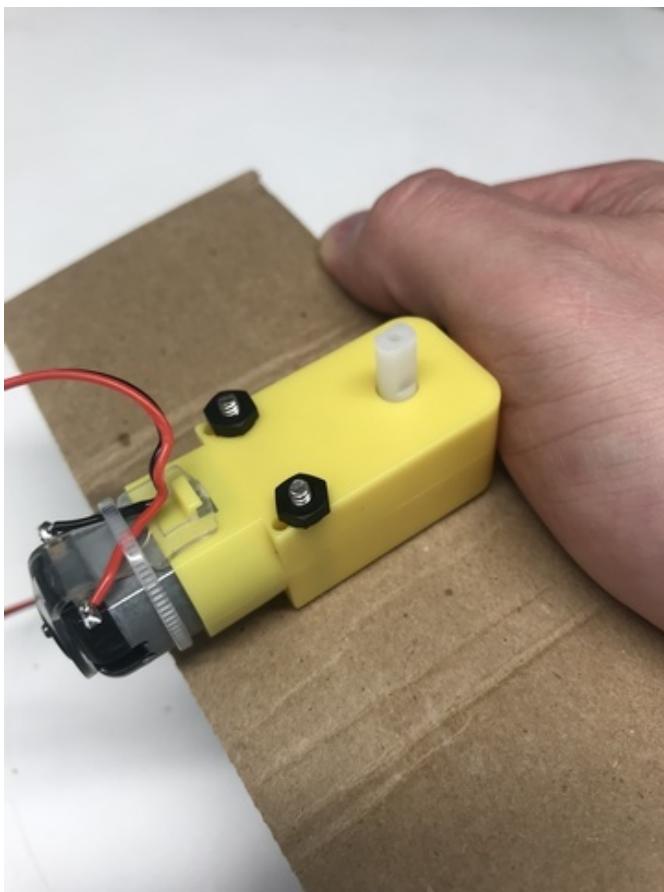
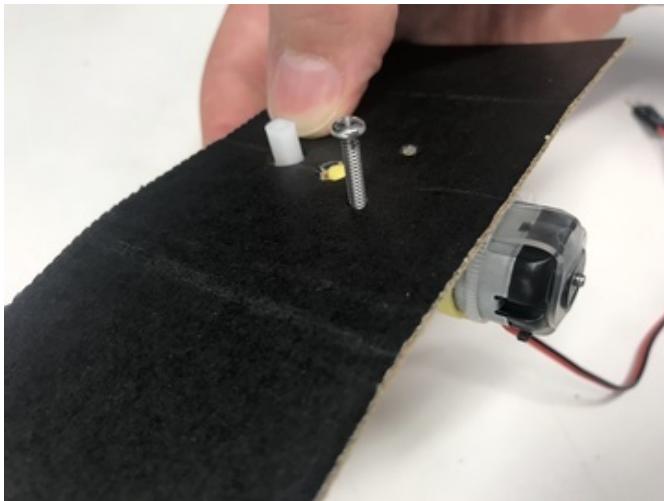
Push motor into cardboard.

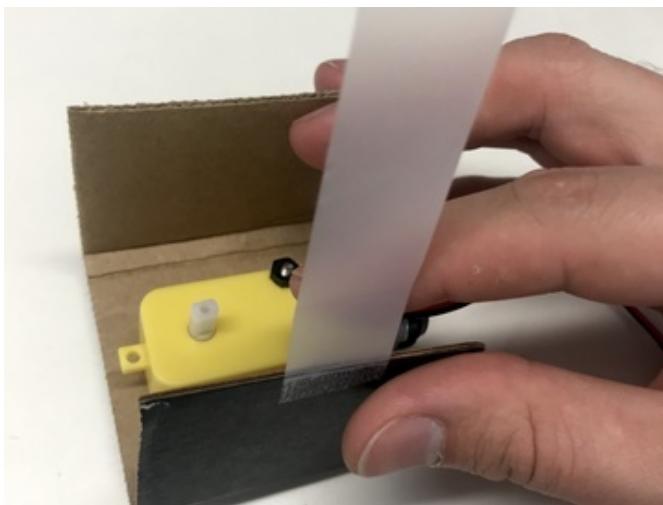
Mount motor to the cardboard using 1" long screws (double sided tape will also work).

Fold sides of cardboard over.

Hold these flaps in place with a strip of tape.

Now you have a handy stand for your zoetrope!



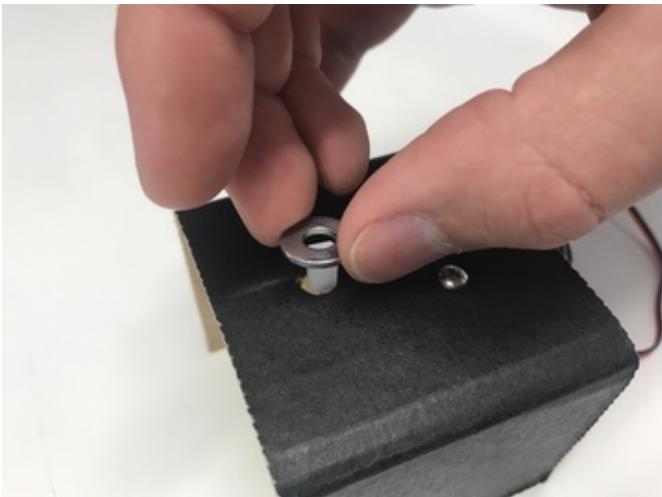


## Add a washer

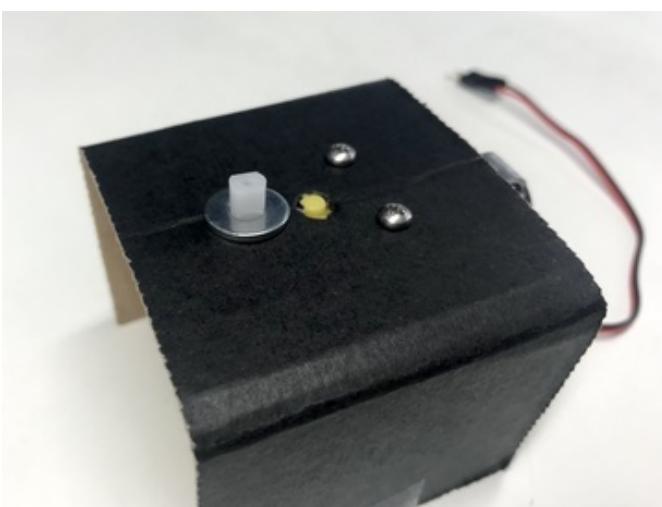
A washer is useful to decrease friction between the base of your zoetrope and the motor housing.

If you don't have a metal washer handy, a cardboard one will do just as well.

Cut out a circle of cardboard and use a hole punch to make a hole in the center.



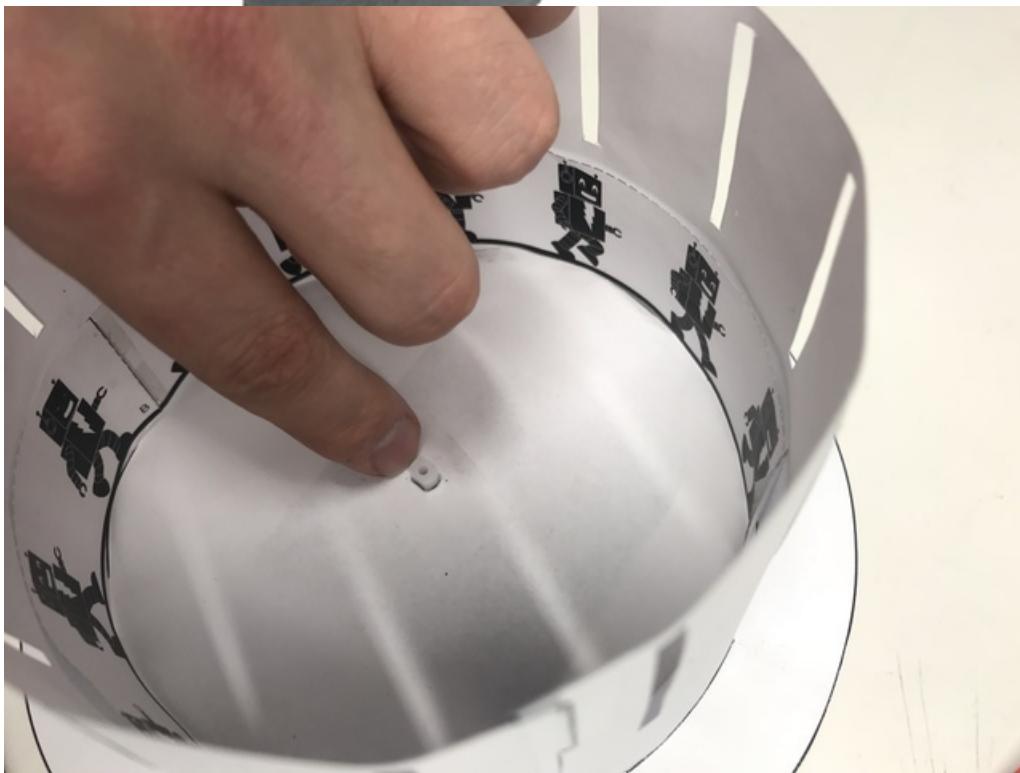
Drop your washer over the motor shaft.





---

Mount the zoetrope on the motor hub, and we're ready to get things moving!



# Coding with CircuitPython

## Getting Familiar

This project uses CircuitPython, which works best if using the Mu Editor. If you haven't previously used Mu, [this guide will get you started \(https://adafru.it/ANO\)](https://adafru.it/ANO).

If you haven't used Circuit Playground Express with CRICKIT before, make sure you've updated it with the latest special 'seesaw' version of the CPX firmware. [This guide will show you how \(https://adafru.it/Bfh\)](https://adafru.it/Bfh).

Once you have completed this step you're ready to move on.

## Let's upload some code!

The sample code for this project is taken from the [Make it Move with CRICKIT master guide \(https://adafru.it/BQ2\)](https://adafru.it/BQ2), which demonstrates all the ways that CRICKIT can be used.

This guide is an excellent resource for exploring CRICKIT further. Prefer to use a continuous rotation servo instead of a DC motor? No problem, the [Make it Move with CRICKIT guide \(https://adafru.it/BQ5\)](https://adafru.it/BQ5) can show you how.

## Sample Code

Make sure you've connected the Circuit Playground Express to your computer and have Mu open.

Copy and paste this code into your Mu editor window.

```
from digitalio import DigitalInOut, Pull, Direction
from adafruit_crickit import crickit
import board

# Two onboard CPX buttons for input (low level saves memory)
button_a = DigitalInOut(board.BUTTON_A)
button_a.direction = Direction.INPUT
button_a.pull = Pull.DOWN

button_b = DigitalInOut(board.BUTTON_B)
button_b.direction = Direction.INPUT
button_b.pull = Pull.DOWN

# Create one motor on seesaw motor port #1
motor = crickit.dc_motor_1

while True:
    if button_a.value:
        print("Button A pressed, go!")
        motor.throttle = 1.0 # full speed!

    if button_b.value:
        print("Button B pressed, stop!")
        motor.throttle = 0 # stop!
```

This snippet of code gives Circuit Playground Express two simple instructions:

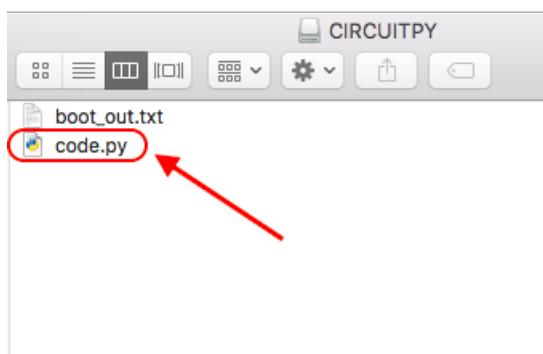
1. Press the A button to turn motor ON

## 2. Press B button to turn motor OFF

Once this code is copied into Mu, press the **Save** button - your code should be saved to the **CIRCUITPY** disk drive (which appears when the Circuit Playground Express is plugged into your computer) as **code.py**.



It's important to name your file **code.py**! If you name your program something else, like **zoetrope.py**, the Circuit Playground Express won't know to automatically run it.

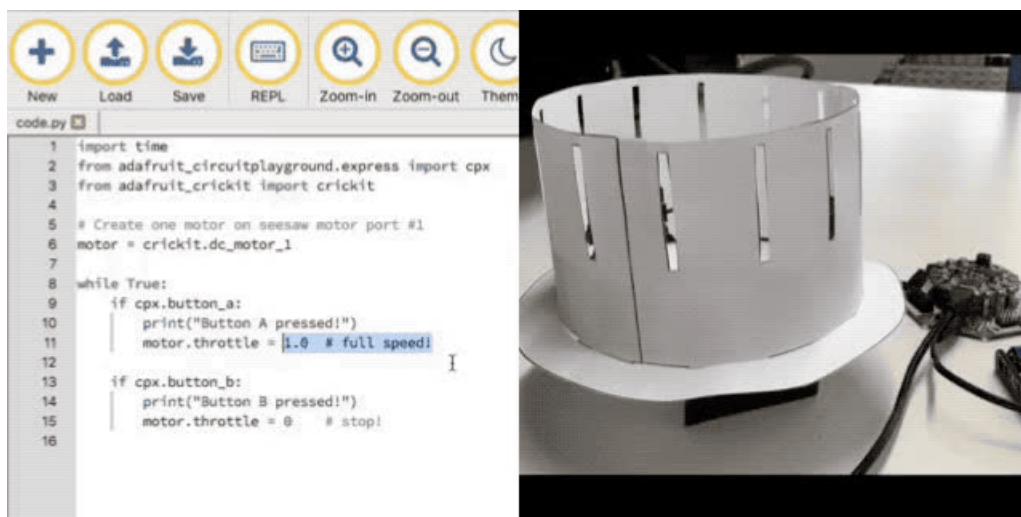


## Making Changes

You will likely notice that by default this sample code spins your motor *very fast*. Almost too fast to see the animation clearly.

Adabot's sautering stride becomes much more visible at a **slower** rotational speed. We can adjust this by changing the **motor.throttle** value in the code from **= 1.0** (full speed) to something like **0.5** (half speed), then clicking "Save" again.

You may have to adjust it several times down, up, and down. This is where Mu makes it super easy to "dial-in" on the perfect value for the speed.

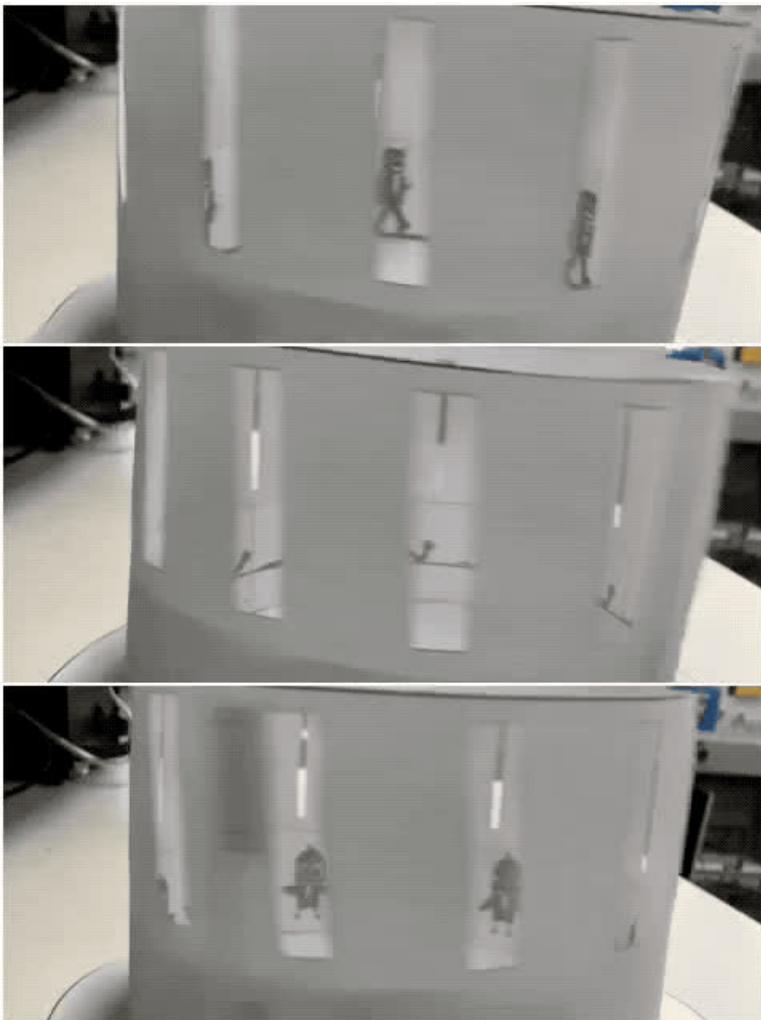


## Experiment!

Have fun making changes to the code and observing what effect it has. For example, you aren't stuck to only using positive numbers. A negative `motor.throttle` value will make Adabot moonwalk backwards rather than walking forwards!

If you'd like to change the direction permanently without changing the code, unscrew the terminals where the motor connects to Crickit and switch the position of the red and black wires.

Try dropping in each of the strips and watching them animate. It can be fun to adjust the `motor.throttle` for each to find the optimal viewing speed for each.



## Custom Animations

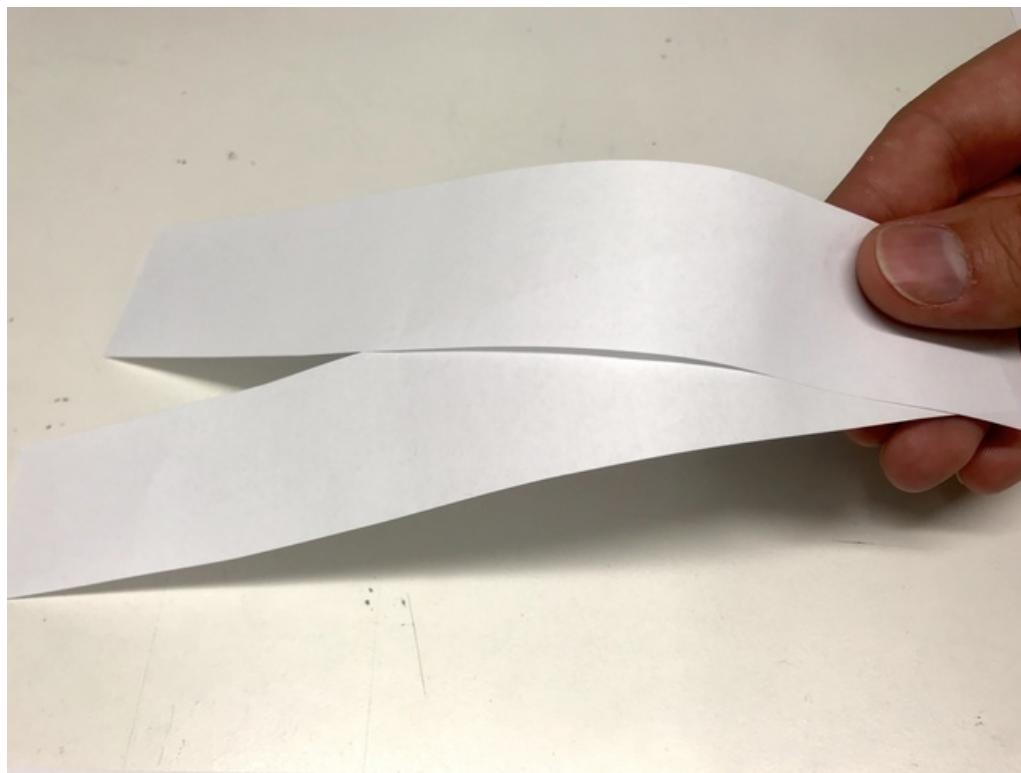
Different animations like the one of a flying pigeon below can be fun to play with.

<https://adafruit.it/BRI>

<https://adafruit.it/BRI>

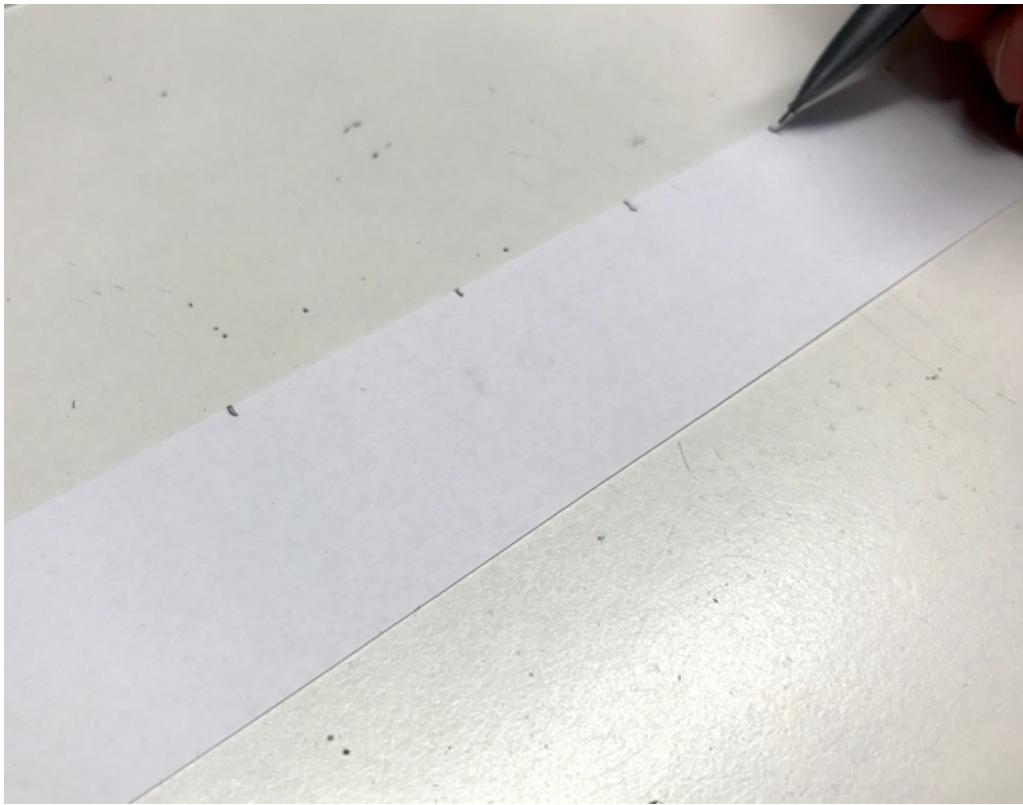
If you find yourself becoming bored by the three animation strips provided in this guide, it might be time to make some of your own!

To do this, cut out two strips of paper, approximately the same length and width as the ones used in the other animations. Tape these two strips together so you have one long strip.

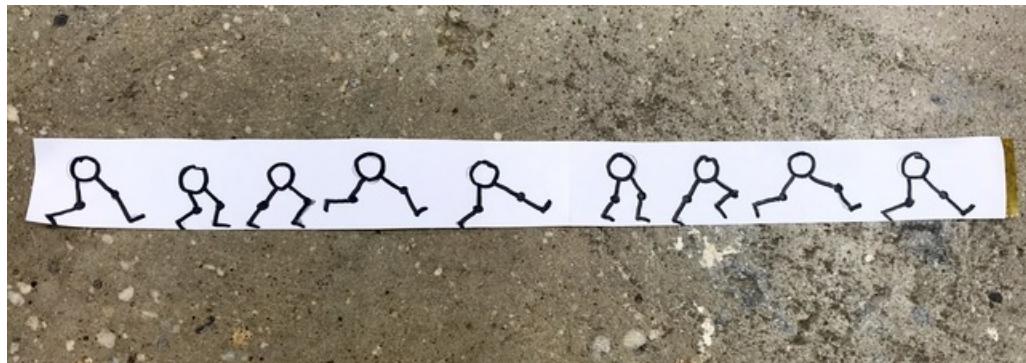


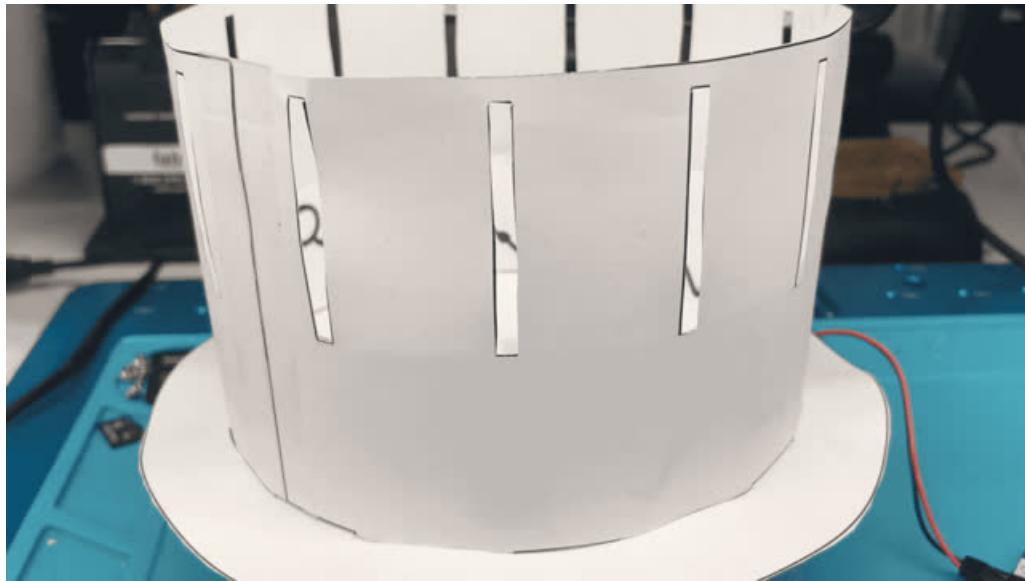
For the best results, give yourself enough space to draw between 10-12 different images, with about equal distance between them.

It can be helpful in this case to make small markings at equal distances along the strip to help space out your drawings evenly.



This is an example of a simple running animation (ref: [Netta Canfi's excellent resource for aspiring animators \(<https://adafru.it/BQ6>\)\).](https://adafru.it/BQ6)





## Exploring further

---

If you enjoy CircuitPython and want to continue learning you can find [lots more CircuitPython projects on the Adafruit Learn System](https://adafru.it/BQP) (<https://adafru.it/BQP>).