adafruit learning system

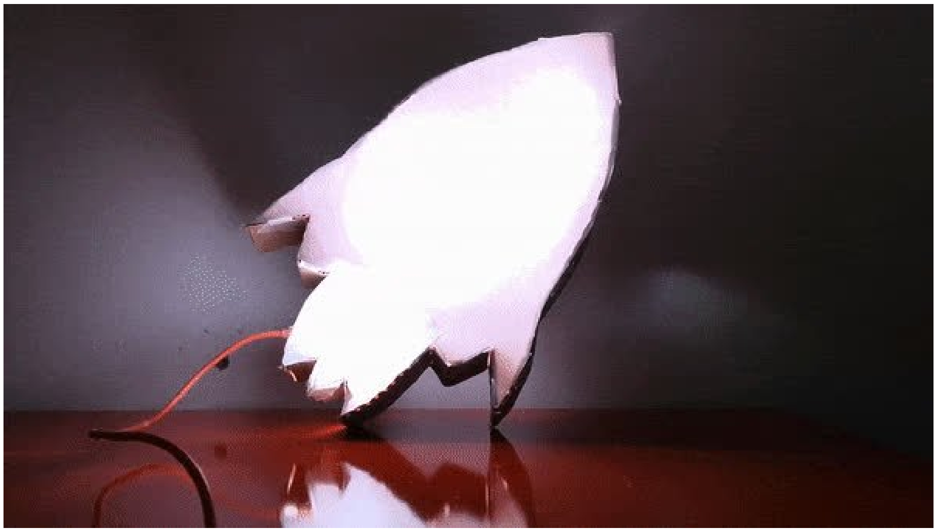
Circuit Playground Express Rocket Lamp

Created by Archie Roques

Last updated on 2019-03-06 07:00:49 PM UTC

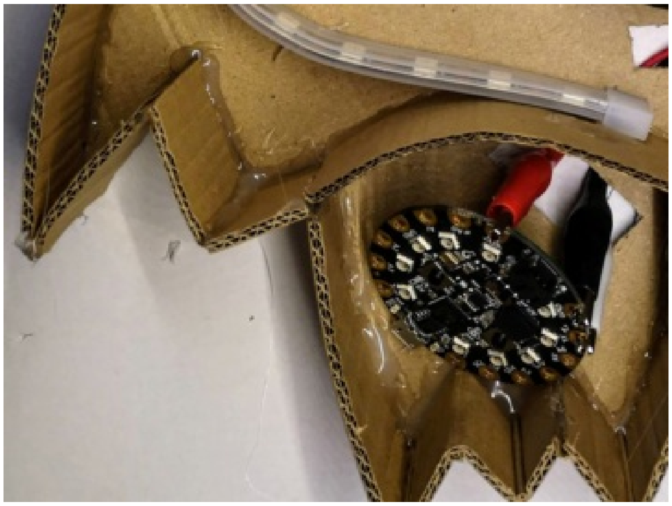
Overview

In this Project, we're building a rocket-themed NeoPixel lamp, using a Circuit Playground Express. The cardboard lamp will create a colourful ambient light most of the time, and when the button on the Circuit Playground Express is pressed, a light-and-sound countdown procedure will begin.



This would be a great project to do with kids - there's no soldering required, and the most dangerous tool you'll need is a craft knife or pair of sharp scissors. There are also lots of possible software or hardware extensions, so you can take the project wherever you like.

If you haven't worked with the Circuit Playground Express before, check out the introductory guide here (https://adafru.it/adafruit-cpx). There's also the NeoPixel Überguide (https://adafru.it/dhw) if you're new to NeoPixels.



The Circuit Playground Express runs CircuitPython code to run the lights and make the sound. This project also features a 50cm long NeoPixel LED strip, which comes pre-soldered with alligator clip leads so you can clip 'em right up to the Circuit Playground board. Handy!

Parts

1 x Adafruit Circuit Playground Express

The brains of the project

ADD TO CART

1 x Alligator Clip LED Strip - 60 LED/m 30cm

Lights up the inside of the rocket

ADD TO CART

1 x microUSB to USB cable

To download the code and (optionally) power the project

ADD TO CART

1 x Lithium Ion Polymer Battery - 3.7v 500mAh [optional]

Provides optional battery power to the project

ADD TO CART

1 x 3 x AA Battery Holder with On/Off Switch, JST, and Belt Clip [optional]

Provides optional battery power to the project

ADD TO CART

1 x AA batteries, 3 pack

Alkaline AA batteries (LR6) - 3 pack

ADD TO CART

1 x 5V 2.5A Switching Power Supply with 20AWG MicroUSB Cable

An alternate, more permanent wall power supply, if desired

ADD TO CART

1 x Flat sheets of fluted cardboard

Forms the body of the rocket

1 x Roll of masking tape

Holds it all together!

1 x Sheet of tracing paper or printer paper

Forms the front of the rocket and diffuse the light

Tools

1 x Strong scissors

To cut the cardboard with

ADD TO CART

1 x Craft knife or box cutter

Good for cutting fiddly holes

ADD TO CART

1 x Cutting mat

Protect your table!

1 x Hot glue gun and glue sticks

To attach cardboard & components

1 x Printer or Laser Cutter

To make the template for cutting the rocket

1 x Makedo Cardboard Toolkit [optional]

This would be another, more kid-friendly way to build this project

ADD TO CART

Print and Cut Template

To make this project easier, we designed a large legal /A3 sized printable template with a nice rocket shape. If you don’t have a large legal / A3 printer, you should be able to tile it onto 2 sheets of A4 or letter size paper. You don’t have to have this, but I found I made a lot less mistakes when I used it!

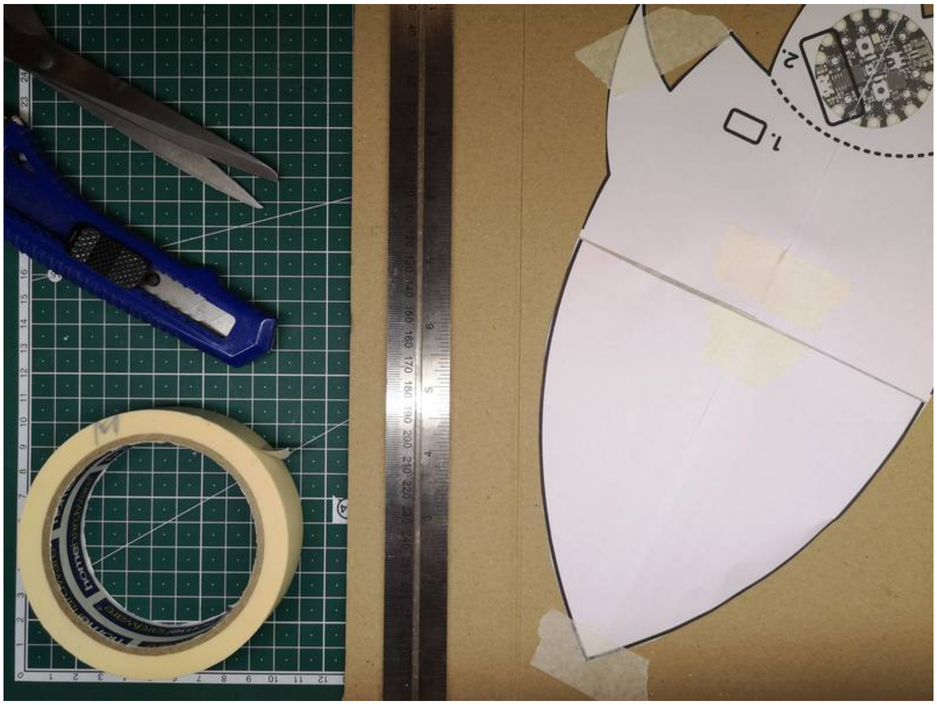
There’s also a file available for laser cutting the base, if you have access to a laser cutter.

https://adafru.it/E8T

https://adafru.it/E8T

https://adafru.it/E8U

https://adafru.it/E8U

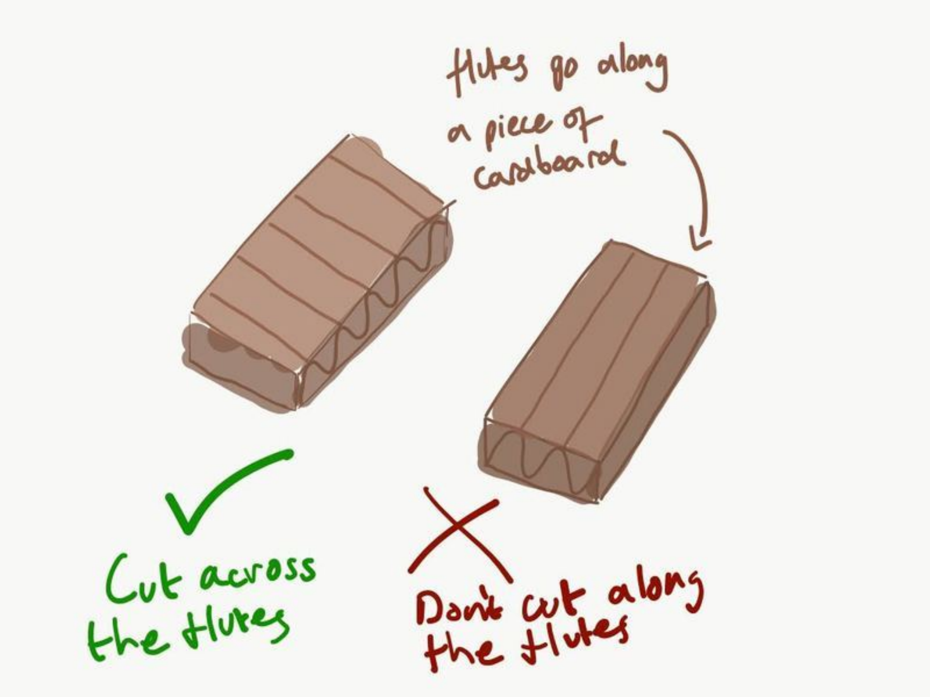


⚠ Scissors and knives are sharp! Make sure you closely supervise kids if you're doing this project with them. A specialist cardboard cutter (as mentioned on the Overview page) is a good idea if little fingers are involved.

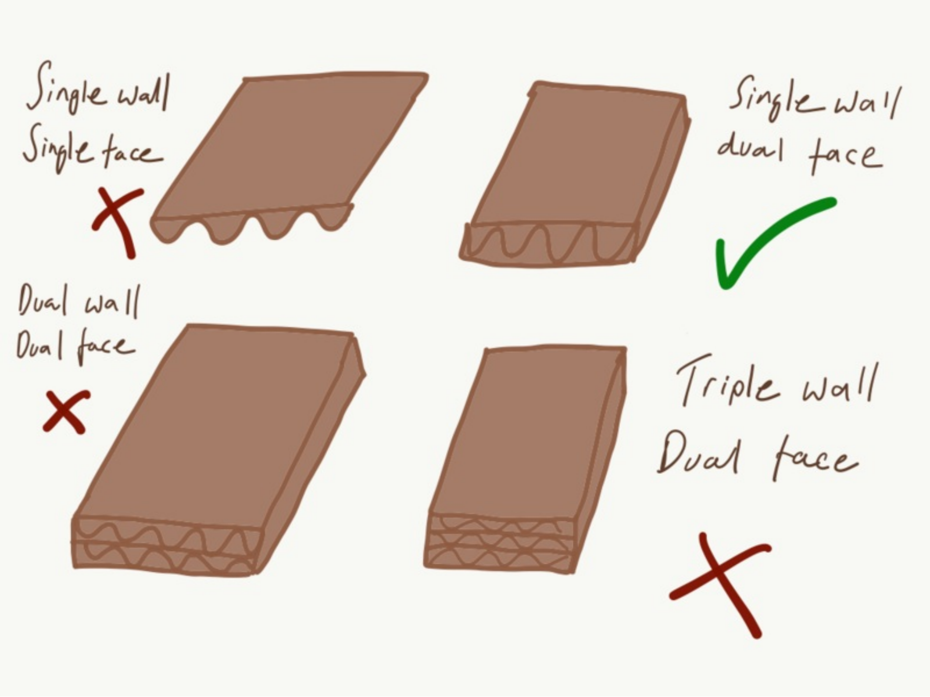
Cut the template out, and if you've tiled it over two pages, tape them together. Place and tape or glue it onto a piece of cardboard, but try not to go over joins or bends in the board.

Use either a craft knife or a pair of scissors to cut around it to make your base shape. You’ll also need to cut out two slots for cables as on the template for the alligator clips. Then remove the template from the cardboard.

You'll also need to cut some strips of card about a 1.5 inches wide to glue around the edge, but make sure you cut these strips across (perpendicular to) the flutes of the cardboard - see the diagram.



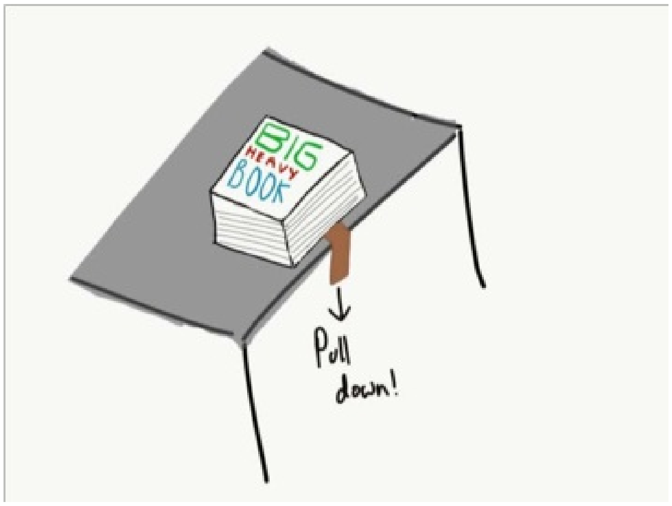
You'll need to use fluted cardboard, preferably with a single wall and dual face. The diagram below should help you figure out what your cardboard is...



You can also laser-cut the base from cardboard using the DXF file, but that's beyond the scope of this tutorial.

Bending the board

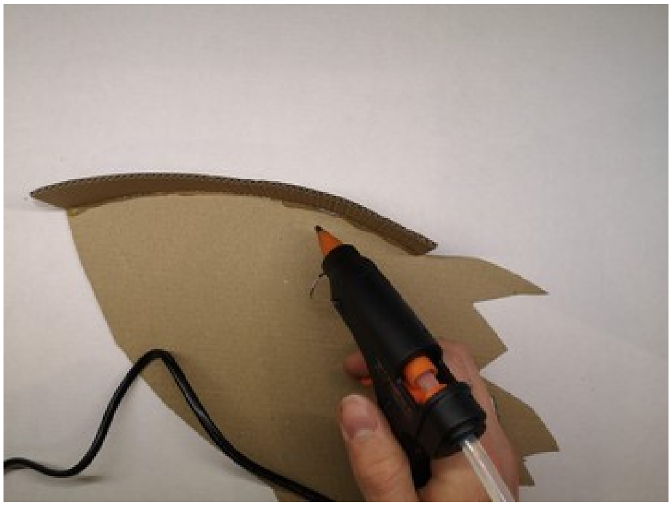
The first stage is to bend the cardboard strips you've cut so that they are flexible. If you don't do this, they'll just kink, but we want them to form nice sweeping arcs for the curve of the rocket's body.



Grab a heavy book, and put it near the edge of a table. Place the cardboard strip under the book, and then pull it out and downwards. The diagram and gif should help you work out what to do here. You'll need to leave one strip not bent, so that you can use it for the straight bits.

Gluing

The next step is to use a hot glue gun to attach the strips you cut to the edge of the rocket shape. This step is fiddly! If you're doing this project with kids it might be an idea to use a cold-melt glue gun, or make do with traditional paper building methods, tape or a Makedo cardboard toolkit (https://adafru.it/BFH).



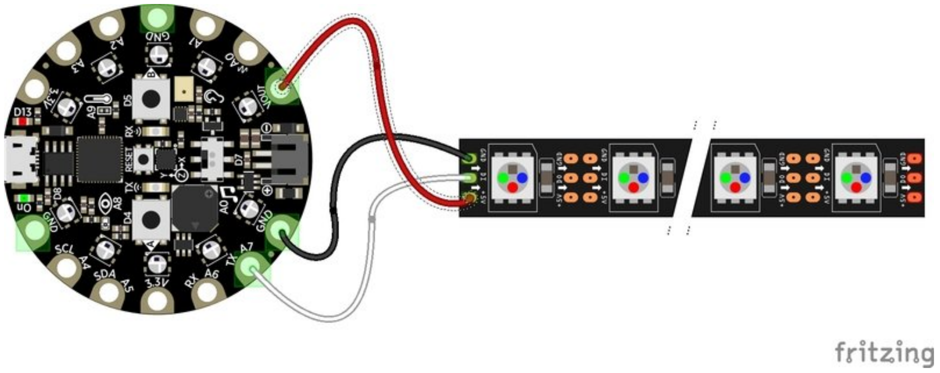
Position the card strip around the edge of the rocket shape and cut it to length. Then put some sticky tape or blue tack at each end to secure it in place. Run a bead of hot glue all along the bottom edge of the corner to secure it in place. It's also a good idea to glue up the joins of the pieces of cardboard so that the light doesn't leak out.

Do this for all the bits of cardboard. Then do another strip along the dotted line of the template, to separate the fire from the rest of the rocket (otherwise the different colours of light will merge into one).

If you're using a USB cable to power it (not a battery), you'll need to test-fit the Circuit Playground Express into the bottom of the rocket and cut a little slot for the USB cable to pass through.

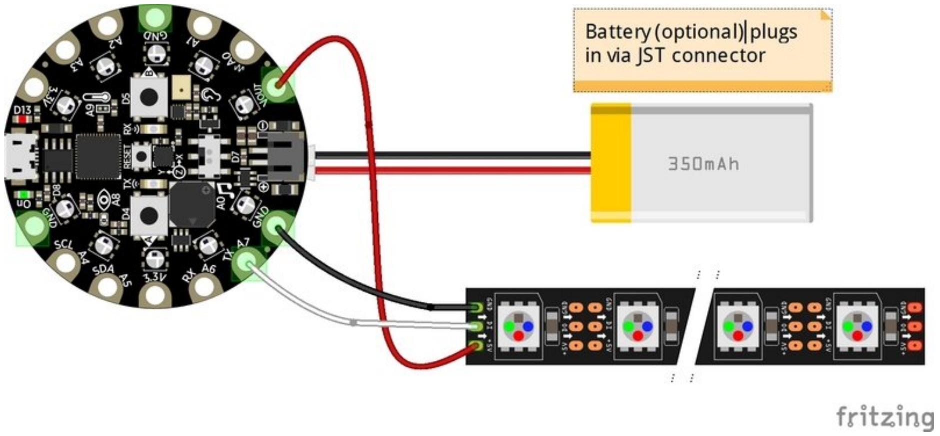
You can also paint or decorate the inside or outside rocket at this stage. Just bear in mind that the inside of the rocket will be covered with paper by the end.

Circuit Diagram



The circuit is really simple, perfect for beginners. It’s a good idea to do a dry-run of the circuit assembly first to test everything works before assembly on the rocket. You only need to connect 3 alligator clips:

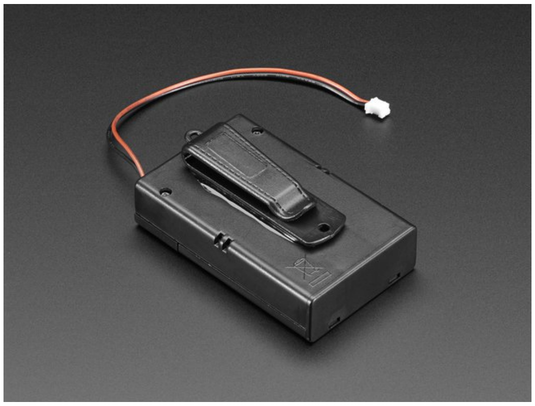
- **Red** goes to **Vout**
- **Black** goes to **Gnd**
- **White** goes to **A7**



For a permanent installation, a USB cable and 5V supply are fine.

If you’re using a battery pack or LiPo battery, plug it into the black JST connector of the Circuit Playground Express, opposite the microUSB connector. If you need more advice on what battery to choose, check out this guide (https://adafru.it/jNa).

For a kid-friendly and mobile option, the 3xAA battery pack would be great.



3 x AA Battery Holder with On/Off Switch, JST, and Belt Clip

$2.95
IN STOCK

ADD TO CART

 If you're using a Lithium Polymer battery, it's a good idea for an adult to do this step - they don't react kindly if short circuited!

If you'd like to permanently mount the project with wall power, this supply may work well.

5V 2.5A Switching Power Supply with 20AWG MicroUSB Cable

$7.50
IN STOCK

ADD TO CART

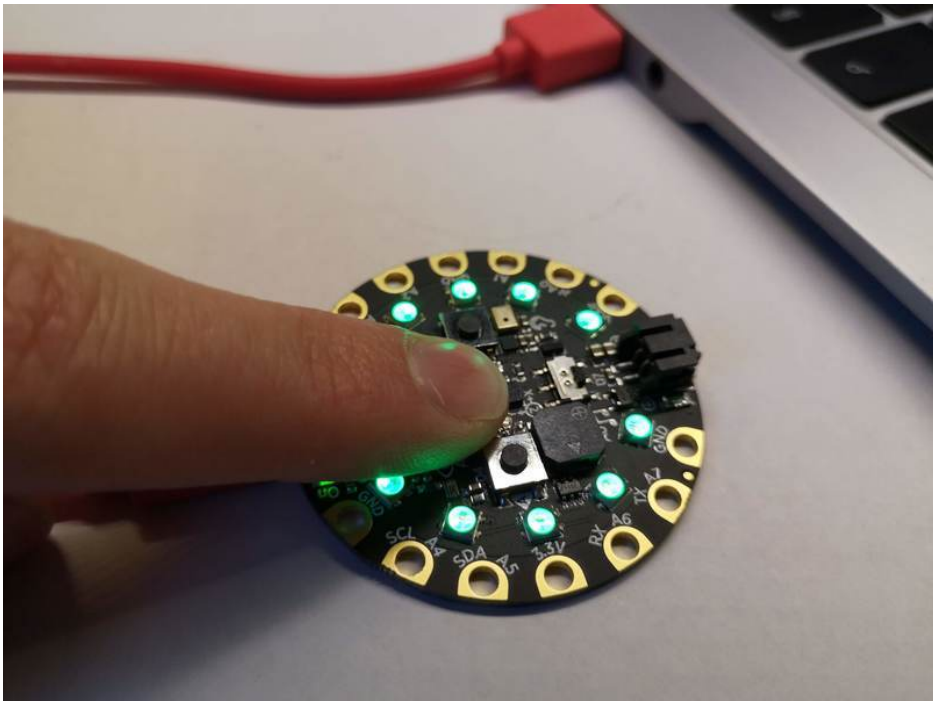
Code

Setup

If you've never used the Circuit Playground Express with CircuitPython before, you'll need to do some setup. Plug in the Circuit Playground Express to your computer via a known good data+power USB cable, and click the little reset button in the center to put it into bootloader mode.

If you have a new flash drive on your computer named **CIRCUITPY**, all is good.

If the on-board NeoPixels should all turn green and a USB storage drive called **CPLAYBOOT** shows up, we'll need to load a fresh copy of CircuitPython.



You’ll need to follow the below guide to set up CircuitPython and then come back to continue.

https://adafru.it/AFI

https://adafru.it/AFI

Upload the code

The next step is to put the code onto the Circuit Playground and give it a test! Open a new text file in your text editor of choice, and paste in the code below. Then save the file onto the USB drive under the filename **code.py**. Please don't rename it, it must be called **code.py**.

```
# Code for Circuit Playground Express Rocket Lamp

# A fun lighting project using Circuit Playground Express and NeoPixels

# Written by Archie Roques for Adafruit Industries
# For full instructions see learn.adafruit.com/cpx-rocket-lamp !
# MIT License, see LICENSE for more info.

# import the libraries needed for this project
import time
import random
import board
import neopixel
import digitalio
import audioio

# enables the speaker for audio output
spkrenable = digitalio.DigitalInOut(board.SPEAKER_ENABLE)
spkrenable.direction = digitalio.Direction.OUTPUT
spkrenable.value = True

# define the onboard NeoPixel strip, and the externally connected one
pixels = neopixel.NeoPixel(board.NEOPIXEL, 10, brightness=.2)
strip = neopixel.NeoPixel(board.A7, 30, brightness=.2)

# turn off the onboard pixels
pixels.fill((0, 0, 0))
pixels.show()

# set the rocket body LEDs to a pretty colour - we chose green!
# Colours are expressed in RGB format
# with each digit going up to 255.
# In this case we've used 0 red, 255 green and 150 blue.
strip.fill((0, 255, 150))
strip.show()

# set up the buttons to trigger the countdown
buttonA = digitalio.DigitalInOut(board.BUTTON_A)
buttonA.direction = digitalio.Direction.INPUT
buttonA.pull = digitalio.Pull.DOWN

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

#this function will play the audio file
def play_audio():
    #open the file
    wave_file = open("liftoff.wav", "rb")
    #play the file
    with audioio.WaveFile(wave_file) as wave:
        with audioio.AudioOut(board.A0) as audio:
            audio.play(wave)
            #wait until audio is done
            while audio.playing:
                pass

# this function lights the CPX NeoPixels up with a blue fire animation
# (really quite hot) for about 1 second
def blue_fire(repeats):
    # fills them with a blue colour to start
    pixels.fill((40, 0, 200))
    pixels.show()
    # each repeat roughly corresponds to a second of running including processing time
    for _ in range(repeats):
        for _ in range(20):
            # pick a random pixel to fill
            j = random.randint(0, 9)
            # makes the pixel either lighter or darker to simulate firey goodness
            if random.random() > 0.5:
                pixels[j] = (40, 80, 250)
            else:
                pixels[j] = (20, 0, 200)
            pixels.show()
            # waits a bit so you can see it
            time.sleep(0.02)
            # returns pixel to original colour
            pixels[j] = (40, 0, 200)
            pixels.show()

# this function lights the CPX NeoPixels up with a white fire animation
# (super very hot) for about 1 second
def white_fire(repeats):
    # fills them with a white colour to start
    pixels.fill((100, 100, 100))
    pixels.show()
    # each repeat roughly corresponds to a second of running including processing time
    for _ in range(repeats):
        for _ in range(20):
            # pick a random pixel to fill
            j = random.randint(0, 9)
            # makes the pixel either lighter or darker to simulate firey goodness
            if random.random() > 0.5:
                pixels[j] = (140, 140, 100)
            else:
                pixels[j] = (100, 100, 140)
            pixels.show()
            # waits a bit so you can see it
            time.sleep(0.01)
            pixels[j] = (100, 100, 100)
            # returns pixel to original colour
            pixels.show()

# this function lights the CPX NeoPixels up with an orange fire animation
# (nice and toasty) for about 1 second
def orange_fire(repeats):
    # fills them with an orangy colour to start
    pixels.fill((200, 50, 0))
    pixels.show()
    # each repeat roughly corresponds to a second of running including processing time
    for _ in range(repeats):
        for _ in range(10):
            # pick a random pixel to fill
            j = random.randint(0, 9)
            # makes the pixel either lighter or darker to simulate firey goodness
            if random.random() > 0.5:
                pixels[j] = (200, 10, 0)
            else:
                pixels[j] = (200, 200, 0)
            pixels.show()
            # waits a bit so you can see it
            time.sleep(0.057)
            # returns pixel to original colour
            pixels[j] = (200, 50, 0)
            pixels.show()
            pixels.show()

# this function makes the body of the rocket light up in a countdown animation
def countdown(seconds):
    # turns off all the lights
    strip.fill((0, 0, 0))
    strip.show()
    # we pass the amount of seconds into the function at the start so the countdown
    # runs for the right amount of time 15 seconds is the max since we only have 30
    # NeoPixels in our strip, and the countdown runs on both sides. 30/2 = 15 seconds
    for i in range(seconds):
        # lights the top pixels, then the next ones down, etc etc all the way to the bottom
        strip[seconds-(i+1)] = (200, 200, 200)
        strip[30-(seconds-i)] = (200, 200, 200)
        strip.show()
        # we use the white fire animation as a timer since it takes about a second to run
        white_fire(1)
    # when the countdown's done, flash all the pixels and play a sound to celebrate take off!
    play_audio()
    for i in range(3):
        strip.fill((200, 200, 200))
        strip.show()
        time.sleep(0.1)
        strip.fill((0, 0, 0))
        strip.show()
        time.sleep(0.1)
    # return the strip to the original colour
    strip.fill((0, 200, 100))
    strip.show()

# this loop of code runs all the time and controls when all the other functions happen
while True:
    # if button A is pressed
    if buttonA.value:
        #run the ten second countdown procedure
        countdown(10)
    elif buttonB.value:
        # run the blue fire for a bit and then trigger the final countdown
        # this sequence is about a minute long
        blue_fire(45)
        countdown(10)
    # if the button isn't pressed, loop the normal animation sequence of orange fire
    else:
        orange_fire(1)
```

You'll also need to download the WAV sound file from the link below, and copy that to the **CIRCUITPY** USB drive too.

https://adafru.it/E90

https://adafru.it/E90

Then eject the drive - the code should automatically run and you should get a nice light show! Press button A and watch the countdown run. Button B should provide a 1-minute long (ish!) timer sequence.

If you want to edit the code, Mu is a good choice for a code editor. Mu is written in Python, and works on all 3 major platforms: MacOS, Windows and Linux (including the Raspberry Pi). It has an awesome CircuitPython mode, and has a built-in serial console, so you can easily see text output from your code too. You can follow the below instructions to install Mu:

https://adafru.it/ANO

https://adafru.it/ANO

Extending the Code

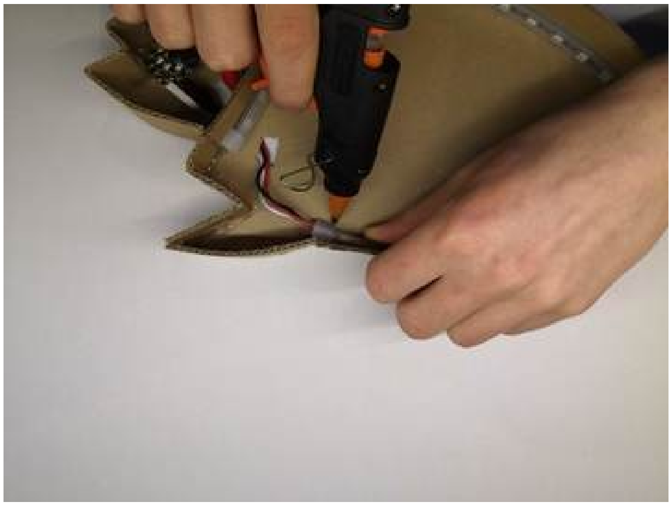
There are all sorts of ideas for extending the code - here are a few possible starting points:

- Change the delay on the timer
- Change the sound when the timer finishes
- Change the colour of the rocket or fire
- Change the speed of the fire animation
- Change the countdown animation

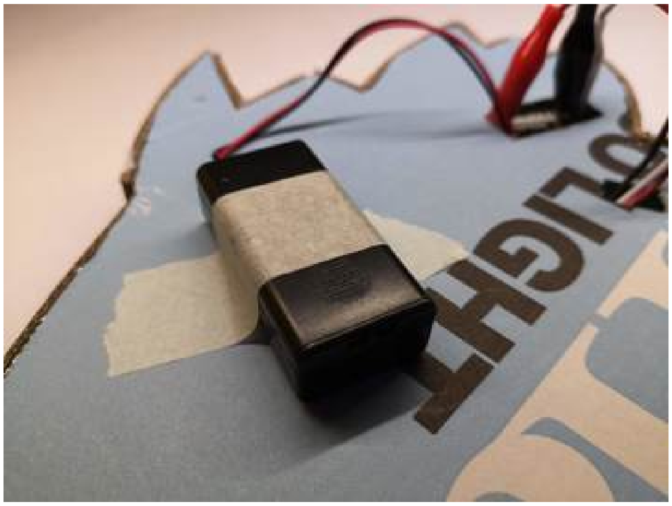
Have fun with it!

Final Assembly

The final step is to assemble everything into the lamp. Start by unclipping the NeoPixel strip from the Circuit Playground Express and thread the alligator clips down through slot 1, and up through slot 2 before clipping them back in (a reminder of the wiring is available here (https://adafru.it/E8s)).



The LED strip is in a silicone casing, so it’s fine to use hot glue to fix it onto the cardboard. You'll need to make sure the glue goes all the way over the LED strip so it doesn't peel out.

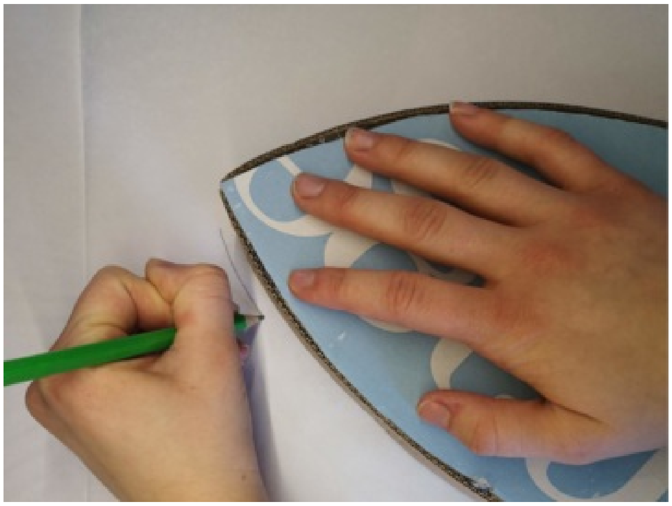


If you’re using a battery, thread the wire into slot number 2 and connect it to the JST connector. You can secure the battery to the back of the rocket with tape.

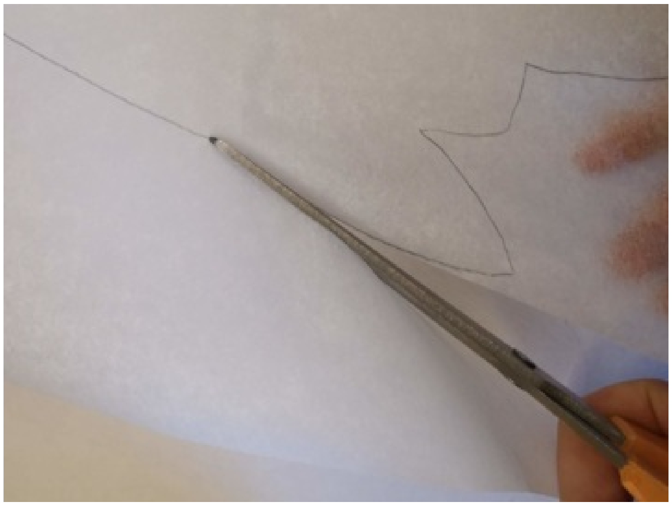
You should now have a fully assembled rocket lamp! There’s just one more step to finish the project off.

You'll need to make a paper cover for the main body of the rocket. This helps diffuse the light, and make it a bit more gentle, especially for the fire section.

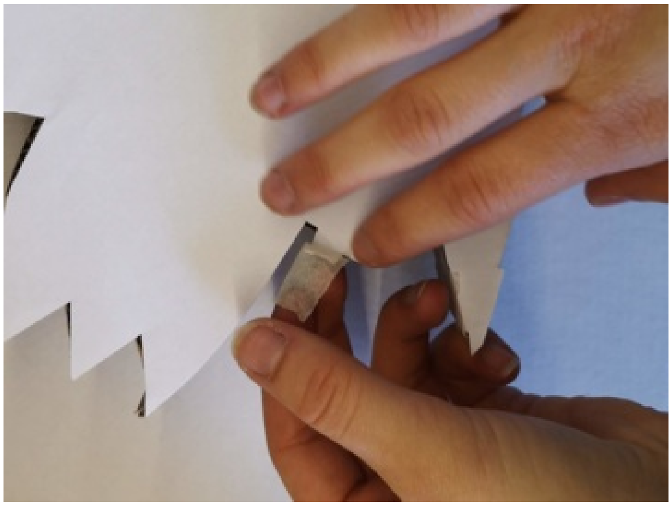
You can use either tracing paper or standard printer paper for this - or even kitchen greaseproof/baking paper. You could even try it with coloured paper, tissue paper or more - get creative!



Use the body of the rocket as a template to draw around. Leave a few tabs on each side to cut around, so that you can tape the paper onto the rocket.



Cut the template out with scissors. It's a good idea to cut a little outside the line so that the paper will definitely cover all of the rocket body - you don't want little gaps for the light to leak out of.



Use some masking tape to attach the tabs of paper to the rocket. Don't forget that you'll still need to access the buttons, so leave the area of paper covering the flame loose, so it forms a lift-up flap.

You could also modify the code to use one of the capacitive touch pads on the Circuit Playground Express (tutorial here (https://adafru.it/ANC)) and connect it to another alligator clip lead or some tin foil, but that's beyond the scope of this tutorial.

And you're done!