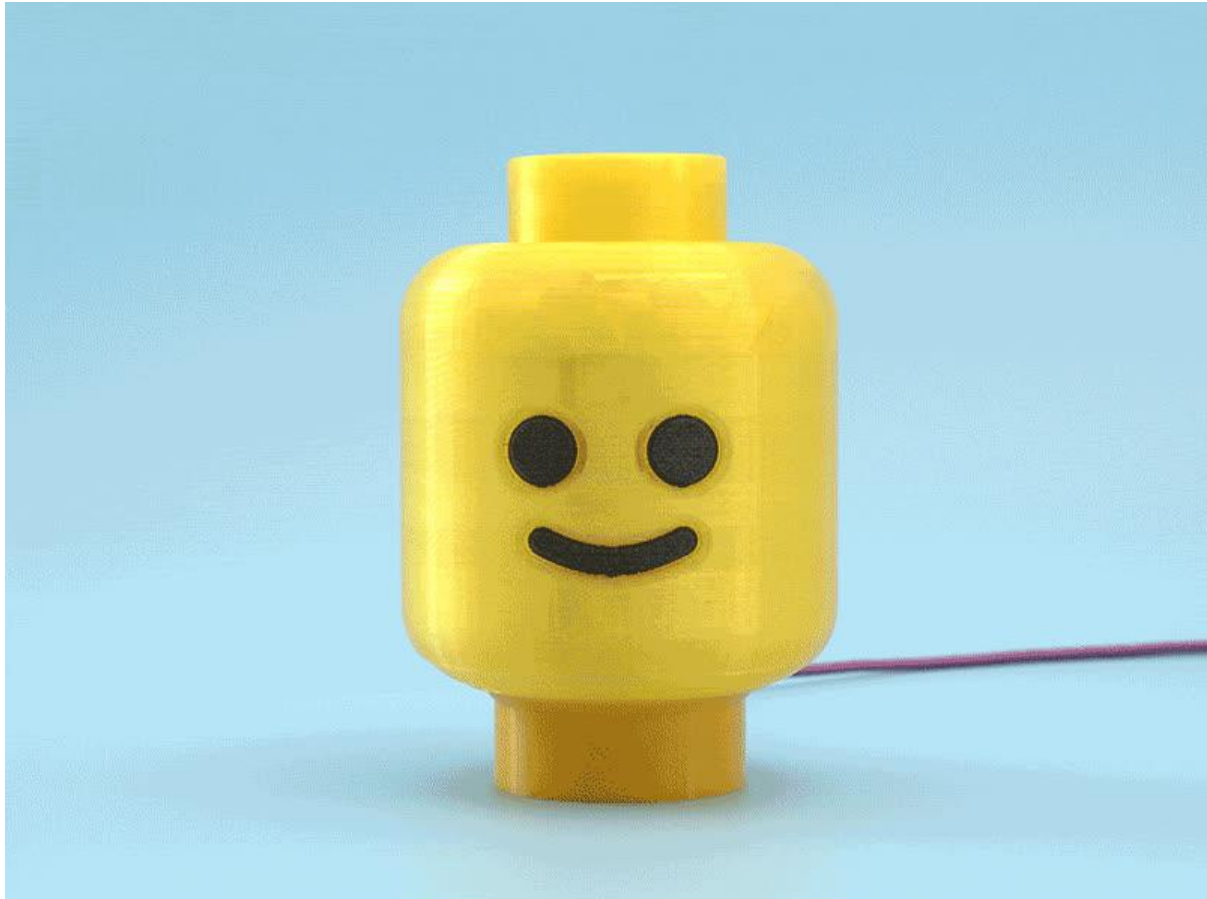




LEGO Head Lamp with Audio

Created by Ruiz Brothers



<https://learn.adafruit.com/lego-head-lamp-with-audio>

Last updated on 2022-12-01 03:44:24 PM EST

Table of Contents

Overview	3
<ul style="list-style-type: none">• Interactive Lamp Projects• CircuitPython• STEMMA Speaker for Circuit Playground• Parts	
Circuit Diagram	6
<ul style="list-style-type: none">• Circuit Diagram• Adafruit Library for Fritzing• Wired Connections• USB Power	
Code	7
<ul style="list-style-type: none">• Setup Circuit Playground Express with CircuitPython• Quick Start• Download Adafruit CircuitPython Library Bundle• Required Library• Upload Code• Audio Files• Mu Python Editor• Loop and Play• Customization Options	
3D Printing	11
<ul style="list-style-type: none">• 3D Parts• Slicing Parts• Snap Fit Mount• Threaded Parts• Design Source Files• CAD Assembly• Speaker Holder• Press Fit Speaker• Installed Speaker Holder• Speaker Placement• Install CPX to Mount• Connect the Alligator Clips to CPX• Install Case to Head• Connect STEMMA Speaker• Install Top to Head• USB Power	

Overview



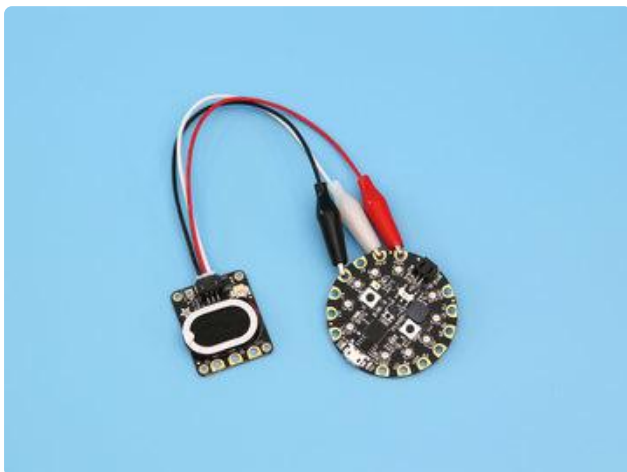
Interactive Lamp Projects

Build a light up LEGO head with Adafruit's Circuit Playground Express! 3D print the parts and make an interactive lamp that plays sound clips from the LEGO movie. It uses the built-in accelerometer to do shake detect to play audio wave files and animate NeoPixels.



CircuitPython

Use Adafruit's CircuitPython to program the NeoPixels and play audio files. You can use the code to create your own sketches with flash animations with shake-to-play audio wave files.



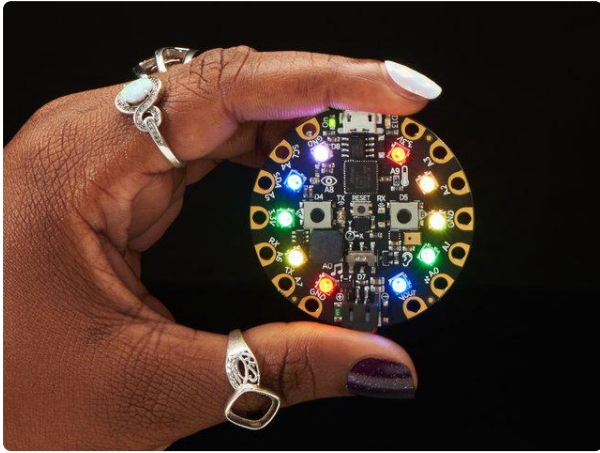
STEMMA Speaker for Circuit Playground

Use the STEMMA speaker to easily add audio to the Circuit Playground Express. With the 3-wire JST alligator cable, you can quickly connect and no soldering is necessary!

Parts



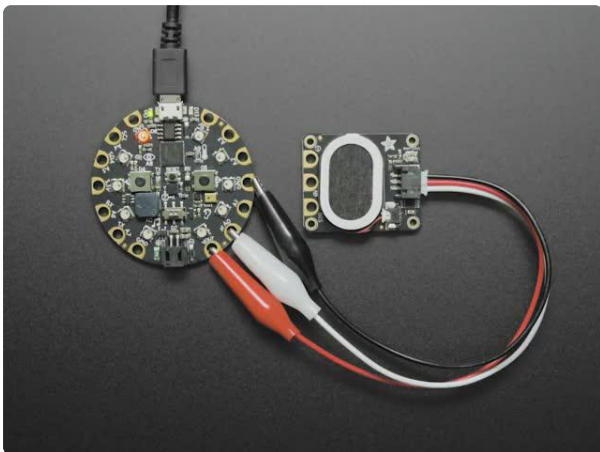
All you need to build this project is a Circuit Playground Express, STEMMA Speaker, 3-wire JST alligator cable and a microUSB cable. We think this makes a great kit for a classroom setting!



[Circuit Playground Express](#)

Circuit Playground Express is the next step towards a perfect introduction to electronics and programming. We've taken the original Circuit Playground Classic and...

<https://www.adafruit.com/product/3333>



[Adafruit STEMMA Speaker - Plug and Play Audio Amplifier](#)

Hey, have you heard the good news? With Adafruit STEMMA boards you can easily and safely plug sensors and devices together, like this Adafruit STEMMA Speaker - Plug and Play...

<https://www.adafruit.com/product/3885>



JST PH 2mm 3-pin Plug to Color Coded Alligator Clips Cable

This cable will let you turn a JST PH 3-pin cable port into 3 individual wires with grippy mini alligator clips. We're carrying these to match up with any of our boards or...

<https://www.adafruit.com/product/4030>



Fully Reversible Pink/Purple USB A to micro B Cable - 1m long

This cable is not only super-fashionable, with a woven pink and purple Blinka-like pattern, it's also fully reversible! That's right, you will save seconds a day by...

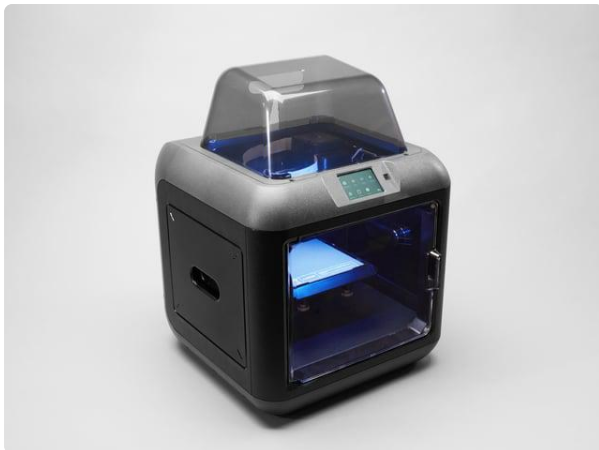
<https://www.adafruit.com/product/4111>



Filament for 3D Printers in Various Colors and Types

Having a 3D printer without filament is sort of like having a regular printer without paper or ink. And while a lot of printers come with some filament there's a good chance...

<https://www.adafruit.com/product/2080>



Monoprice Inventor II 3D Printer with Touchscreen and WiFi

The Monoprice Inventor II 3D Printer Touchscreen with WiFi is a perfect entry-level 3D printer with small footprint and reliable performance. It comes equipped with...

<https://www.adafruit.com/product/3897>

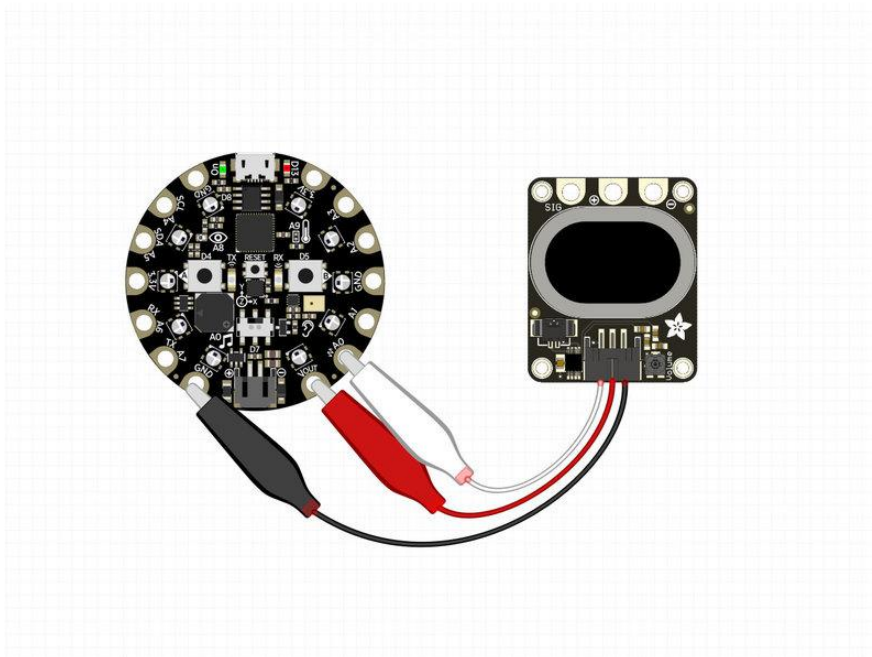
Circuit Diagram

Circuit Diagram

This provides a visual reference for wiring of the components. They aren't true to scale, just meant to be used as reference. This diagrams was created using [Fritzing software](#) ().

Adafruit Library for Fritzing

Use our Fritzing parts library to create circuit diagrams for your projects. Download the library or just grab the individual parts. Get library and parts from [GitHub Adafruit Fritzing Parts](#) ().



Wired Connections

Plug in the 3-wire JST-PH cable to the port on the STEMMA Speaker. Reference the circuit diagram and follow the colored wired to properly connect the STEMMA speaker to the CPX.

- Black wire from STEMMA Speaker to GND on CPX
- White wire from STEMMA Speaker to A0 on CPX
- Red wire from STEMMA Speaker to VOUT on CPX

USB Power

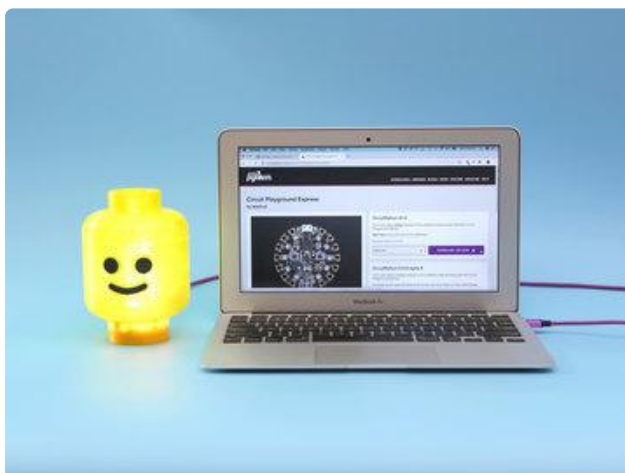
The Circuit Playground Express can stay connected via USB power from either a 5V wall adapter or computer's hub. This allows the project to stay on as long as you want (nightlight mode anyone?). The project has a cutaway to accommodate for a USB cable.

Code



Setup Circuit Playground Express with CircuitPython

We'll need to get our board setup so we can run CircuitPython code. Let's walk through these steps to get the latest version of CircuitPython onto your board.



Quick Start

Download the CircuitPython UF2 for Circuit Playground Express

Connect Circuit Playground Express to your computer over USB and press the Reset button

Drag-n-drop the CircuitPython UF2 onto the CPLAYBOOT drive - the drive will vanish and a new CIRCUITPY drive should appear.

Copy code and library files to the CIRCUITPY drive

[Download Latest Circuit Python for Circuit Playground Express](#)

Download Adafruit CircuitPython Library Bundle

In order to run the code, we'll need to download a library. The download linked below will contain all the libraries available for CircuitPython. To run the code for this project, we only need the Adafruit Circuit Playground library. Unzip the library bundle and search for the library. Drag and drop it onto a folder named lib on the CIRCUITPY drive (create the folder if it is not already on the Circuit Playground Express).

Required Library

- Adafruit Circuit Playground – `adafruit_circuitplayground`

Download Latest Circuit Python
Library Bundle

Upload Code

Click on the download link below to grab the main code directly from GitHub. Rename the file to `code.py` and drop it onto the CIRCUITPY main (root) directory. Create a new folder on the CIRCUITPY drive and name it sounds. Upload the audio files to that folder. The code will run properly when all of the files have been uploaded.

Use any text editor or favorite IDE to modify the code. We suggest using the Mu Python Editor. See below for more on Mu.

Audio Files

The sounds used in this project were sampled from the LEGO movie. This pack contains sounds that are already in the supported audio format.

Adafruit CircuitPython supports 16-bit, Mono, 22.050kHz .wav audio format. [See this guide \(\)](#) to help format any audio files you might want to use in this project besides the files provided.

Mu Python Editor

Mu is a simple Python editor that works with Adafruit CircuitPython hardware. It's written in Python and works on Windows, MacOS, Linux and Raspberry Pi. The serial console is built right in so you get immediate feedback from your board's serial output!

Installing and Using the Mu Editor

Download Project Zip from GitHub

```
# SPDX-FileCopyrightText: 2019 Noe Ruiz for Adafruit Industries
#
# SPDX-License-Identifier: MIT

# Shake Audio Lamp
# for Adafruit Circuit Playground express
# with CircuitPython
import time
import audioio
import audiocore
import board

from adafruit_circuitplayground.express import cpx

# External Audio Stuff
audio = audioio.AudioOut(board.A0) # Speaker
wave_file = None

def play_wav(name, loop=False):
    """
    Play a WAV file in the 'sounds' directory.
    :param name: partial file name string, complete name will be built around
                 this, e.g. passing 'foo' will play file 'sounds/foo.wav'.
    :param loop: if True, sound will repeat indefinitely (until interrupted
                 by another sound).
    """
    global wave_file # pylint: disable=global-statement
    print("playing", name)
    if wave_file:
        wave_file.close()
    try:
        wave_file = open('sounds/' + name + '.wav', 'rb') # using wave files from
sounds folder
        wave = audiocore.WaveFile(wave_file)
        audio.play(wave, loop=loop)
    except OSError:
        pass # we'll just skip playing then

# flash neopixel effects
def party_flash(duration):
    cpx.pixels.fill((255, 255, 255))
    cpx.pixels.show()
    time.sleep(duration)
    cpx.pixels.fill((255, 0, 0))
    cpx.pixels.show()
    time.sleep(duration)

def led_flash(duration):
```

```

cpx.pixels.fill((255, 255, 255))
cpx.pixels.show()
time.sleep(duration)
cpx.pixels.fill((0, 0, 0))
cpx.pixels.show()
time.sleep(duration)

# make a counter variable
counter = 0

while True:
    # Listen for shakes
    if cpx.shake(shake_threshold=15): # adjust sensitivity - low number is more
sensitive
        print("Shake detected!") # Let us know there was a shake
        counter = counter + 1 # Start a counter
        if counter == 2: # On second shake
            play_wav("awe-a") # play audio
            for _ in range(3): # loop x times
                party_flash(0.4) # neopixel flash
        elif counter == 3: # On third shake
            play_wav("awe-b")
            for _ in range(3): # loop x times
                party_flash(0.4) # neopixel flash
        elif counter == 4: # On fourth shake
            play_wav("awe-c")
            for _ in range(3): # loop x times
                party_flash(0.4) # neopixel flash
        elif counter == 5: # on fifth shake
            counter = 0 # Reset the counter back to zero
            play_wav("untz") #play audio
            for _ in range(3): # loop x times
                led_flash(.18) # faster pixel flash
            cpx.pixels.fill((255,255,255)) # solid pixel
            time.sleep(1) # light it for one second
        else: # On first shake
            play_wav("haha") # play audio
            cpx.pixels.fill((255,255,255)) # white color
            time.sleep(1) # for one second
    else: # When there's no shakyness to be had
        cpx.pixels.fill((0, 0, 0)) # keep pixels off when not shaking

```

Loop and Play

The `play_wav` function handles the location of the wav files. In this project, the audio wav files reside in a folder named sounds. The `party_flash` and `led_flash` animations use `cpx.pixels.fill` to quickly flash between two colors.

In the loop, we set the shake sensitivity and print a statement to let us know when a shake is detected. A variable is setup and starts a counter which increments each time a shake has been detected. This way we can play different audio files each time we shake it. In this project, we'll play up to five different wave files.

- On the first shake, we play haha.wav and turn all NeoPixels white for one second.
- One second shake, we play awe-a.wav and run `party_flash` 3 times, 0.4 seconds each flash.

- On third shake, we play awe-b.wav and run `part_flash` 3 times, 0.4 seconds each flash.
- On fourth shake, we play awe-c.wav and run the `part_flash` 3 times, 0.4 seconds each flash.
- On fifth shake, we set the counter to 0, play untz.wav and run `led_flash` 3 times, 0.18 seconds each flash.

Customization Options

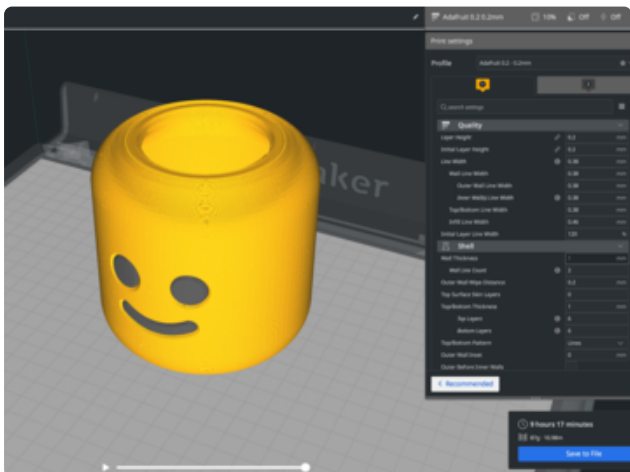
Minor adjustments can be made to customize the functionality. Change the colors of the NeoPixels by adding your own values, just look for the RGB numbers in the code. If you want the lights to always stay on, look for the last line that uses `cpx.pixels.fill` and adjust the RGB values. Rearrange the animations and adjust the timing to match your audio files.

3D Printing



3D Parts

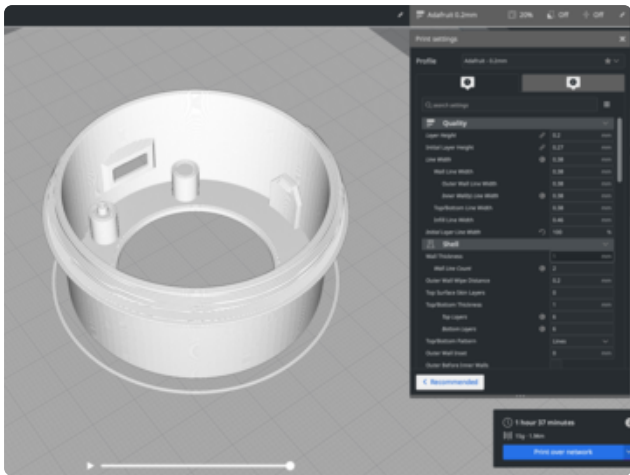
STL files for 3D printing are oriented to print "as-is" on FDM style machines. Original design source may be downloaded using the links below.



Slicing Parts

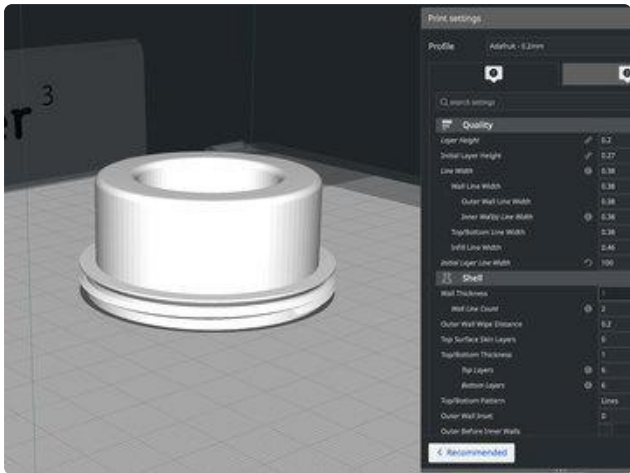
We 3D Printed the head with dual-extrusion so the eyes and mouth are dark when the lights are on. The model has a thin shell and it can be printed without any support material.

PLA filament 220c extruder
 0.2 layer height
 10% gyroid infill
 60mm/s print speed
 60c heated bed



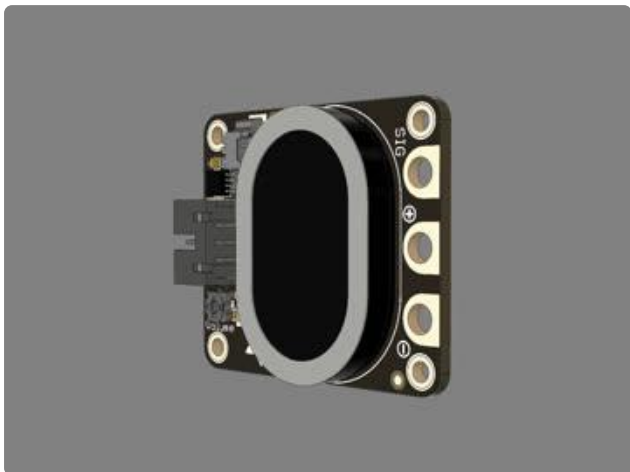
Snap Fit Mount

The Circuit Playground Express PCB snap fits into the case. Insert the PCB at an angle to fit under the snap. Line up the stand offs with the mounting holes and press down to snap fit.



Threaded Parts

The top and bottom parts are designed to screw into the LEGO head. The threads are mostly angled surfaces that 3D print without support material.



Design Source Files

The project assembly was designed in Fusion 360. This can be downloaded in different formats like STEP, SAT and more. Electronic components like Adafruit's board, displays, connectors and more can be downloaded from our [Adafruit CAD parts GitHub Repo](#) ().



CAD Assembly

The components were modeled in the cad assembly. This animation shows where the electronics are mounted and how the top and bottom screw into the base.

[Download CAD files from GitHub](#)

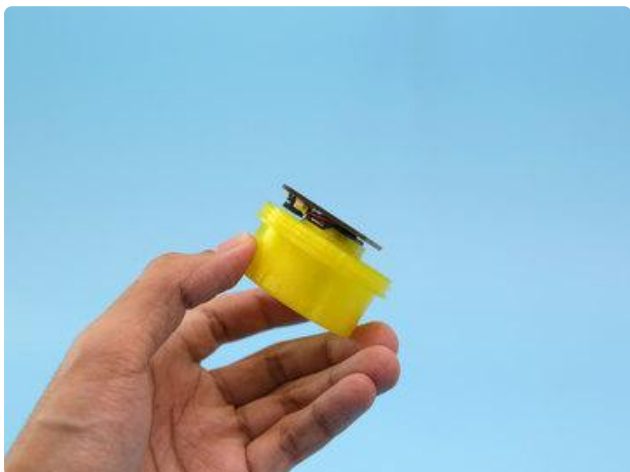
[Adafruit CAD Parts on GitHub](#)

[Fusion 360 Project Share Link](#)



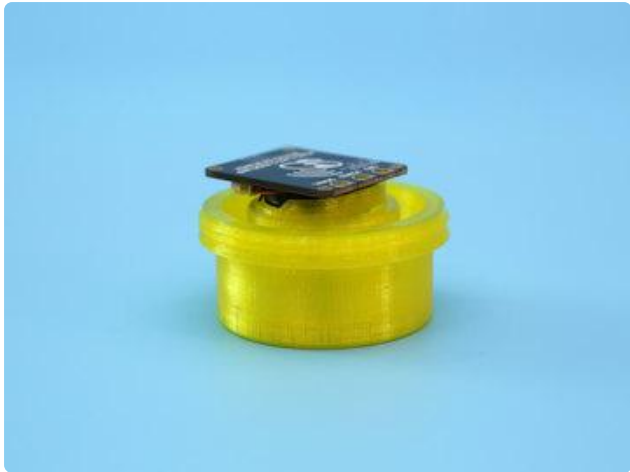
Speaker Holder

The speaker holder will need to be super glued to the top stud. The speaker holder should be centered with the speaker grill on the top stud.



Press Fit Speaker

The speaker is press fitted into the holder and has a snug fit. The sticky adhesive is not necessary so you can leave the protective backing on.



Installed Speaker Holder

A small notch in the speaker holders lip accommodates for clearances for the pico connector and wiring.



Speaker Placement

The placement of the speaker sounds best when it's facing up, pressed up against the speaker grill. When the top is installed the LEGO head sound is amplified and much louder.



Install CPX to Mount

Orient the USB port with the opening and line up the mounting holes. Insert the PCB at an angle to get underneath the snaps. Firmly press down to snap fit the PCB into place.



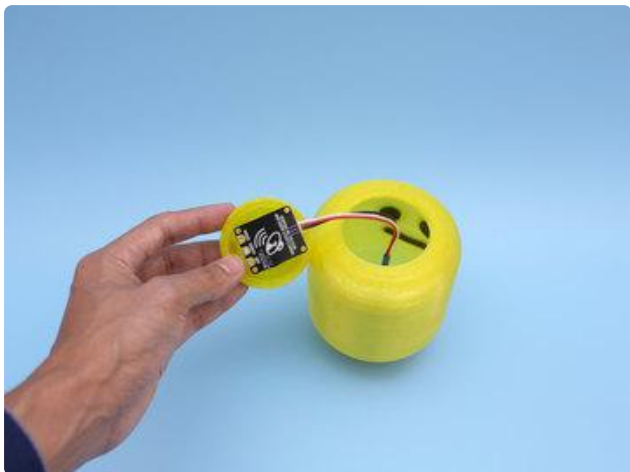
Connect the Alligator Clips to CPX

Grab the JST cable and connect the alligator clips to pads. Red to 3.3V pad, white to A0 pad and black to GND pad. Be sure the alligator clips have a sturdy grip on the Circuit Playground Express.



Install Case to Head

Insert the JST cable into the bottom of the LEGO head and screw in the case.



Connect STEMMA Speaker

Grab the JST cable from top of LEGO head and plug it into the JST port on the STEMMA Speaker.



Install Top to Head

Twist the top stud onto the LEGO head to install it.



USB Power

Use a microUSB cable with a data line. You can leave it plugged into a 5V USB wall adapter or computer's USB hub.