Circuit Playground Bluefruit Pumpkin with Lights and Sounds
Created by Ruiz Brothers

https://learn.adafruit.com/pumpkin-with-circuit-playground-bluefruit

Last updated on 2023-08-29 04:19:20 PM EDT
# Table of Contents

## Overview
- Glowing Halloween Props with Sound Effects
- Illuminated Pumpkin
- Parts List
- Prerequisite Guides
- Parts
- Optional On/Off Switch

## Circuit Diagram
- Circuit Diagram
- Adafruit Library for Fritzing
- Wired Connections
- Battery Toggle Switch

## Software
- Setup Circuit Playground Bluefruit with CircuitPython
- Mu Python Editor
- Quick Start
- Download Adafruit CircuitPython Library Bundle
- Required Libraries
- Upload Code
- Upload Sounds
- Mapped Sounds
- Supported Audio Formats
- NeoPixel Colors

## Bluefruit LE Connect App
- Bluetooth App
- Connect to Circuit Playground Bluefruit
- Using Bluefruit LE Connect App

## 3D Printing
- 3D Parts
- Design Source Files
- Slicing Parts
- Glue Parts
- Attached Parts

## Toggle Switch
- Wiring Toggle Switch
- Wired Toggle Switch

## STEMMA Speaker
- Connect Speaker to Circuit Playground Bluefruit
- Prep Wires
- Wired STEMMA Speaker

## Assembly
- Test Circuit
- Pumpkin Screws
- Remove Bottom
- Test Fit 3D Printed Bottom
- Install STEMMA Speaker
- Installing Battery
- Battery Placement
- Installing Circuit Playground Bluefruit
- Installed Circuit Playground Bluefruit
- Wire Adjustments

Final Build
- Drill Hole for Switch
- Secure Switch
- Connect Battery to Switch
- Connect Switch to Circuit Playground Bluefruit
- Final Circuit Test
- Installing Bottom
- Secure Bottom
- Final Adjustments
Overview

Glowing Halloween Props with Sound Effects
Add NeoPixel LEDs and sound effects to your Halloween Props with Adafruit's Circuit Playground Bluefruit. Wireless trigger lights and sounds with the Bluefruit LE connect app for iOS and Android.

In this project, we'll show you how to add the Circuit Playground Bluefruit and STEMMA speaker to an inexpensive plastic pumpkin. These LED pumpkins from SEASONS is commonly available and features a single LED with a 2x AA battery pack.

Illuminated Pumpkin
These plastic pumpkins are available in many different retail stores. They come in different colors and styles and feature the same circuit. Check the bottom cover to see if your pumpkin is similar. The label reads "SEASONS - U.S. PATENT NO: 6629770". The patent document can be found here.
Parts List
Components used to build this project.

- Circuit Playground Bluefruit https://www.adafruit.com/product/4333
- STEMMA Speaker https://www.adafruit.com/product/3885
- 500mAh Battery https://www.adafruit.com/product/1578
- Mini Toggle Switch https://www.adafruit.com/product/3221
- JST Female Cable https://www.adafruit.com/product/261
- JST Male Cable https://www.adafruit.com/product/3814
- 4-Wire Ribbon Wire https://www.adafruit.com/product/3891

Prerequisite Guides
The following guide walks through wiring a slide switch adapter in great detail. Check it out if you're new to soldering or looking for more examples of wiring up a slide switch.

- DIY Slide Switch Adapter

Parts

Circuit Playground Bluefruit - Bluetooth Low Energy
Circuit Playground Bluefruit is our third board in the Circuit Playground series, another step towards a perfect introduction to electronics and programming. We've...
https://www.adafruit.com/product/4333
Lithium Ion Polymer Battery - 3.7v 500mAh
Lithium-ion polymer (also known as 'lipo' or 'lipoly') batteries are thin, light, and powerful. The output ranges from 4.2V when completely charged to 3.7V. This...
https://www.adafruit.com/product/1578

Mini Panel Mount SPDT Toggle Switch
This or that, one or the other, perhaps or perhaps not! So hard to make decisions these days without feeling like you're just going back and forth constantly. Deciding whether or...
https://www.adafruit.com/product/3221

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 2-Pin Cable - Female Connector 100mm
Red and black tinned wires with a 2-pin JST PH connector on the end. 4" / 100mm long. Matches up nicely with our Lipoly chargers!
https://www.adafruit.com/product/261
JST PH 2-Pin Cable – Male Header 200mm
For a really long time we assumed that the JST PH didn't have a free-hanging male header version. But then we found this JST-PH 2-pin Male Cable, and we were...
https://www.adafruit.com/product/3814

Silicone Cover Stranded-Core Ribbon Cable - 4 Wires 1 Meter Long
For those who are fans of our silicone-covered wires, but are always looking to up their wiring game. We now have Silicone Cover Ribbon cables! These may look...
https://www.adafruit.com/product/3891

Optional On/Off Switch
If you'd like to use an on/off switch that doesn't require any soldering, you can use the pre-wired JST on/off switch. It's a bit bigger and may not be as hidden.

JST 2-pin Extension Cable with On/Off Switch - JST PH2
By popular request - we now have a way you can turn on-and-off Lithium Polymer batteries without unplugging them. This PH2 Female/Male JST 2-pin Extension...
https://www.adafruit.com/product/3064
Circuit Diagram

The diagram below provides a visual reference for wiring of the components. They aren't true to scale, but are meant to be used as a reference. This diagrams was created using Fritzing software.

Adafruit Library for Fritzing

Use Adafruit's Fritzing parts library to create circuit diagrams for your projects. Download the library or just grab the individual parts. Get the library and parts from GitHub Adafruit Fritzing Parts.

Wired Connections

The STEMMA Speaker needs three wired connections. Reference the circuit diagram and follow the colored wired to properly connect the STEMMA speaker to the Circuit Playground Bluefruit.

- Black wire from STEMMA Speaker to GND on CPB
- White wire from STEMMA Speaker to A0 on CPB
- Red wire from STEMMA Speaker to VOUT on CPB
Battery Toggle Switch

The 500mah battery is connected to a toggle switch via a 2-pin JST cable. The voltage (red) wire is connected in-line with the toggle switch. Use the middle pin and either the left or right pins. This allows the power to be turned on and off.

Software

Setup Circuit Playground Bluefruit 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.

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!

Quick Start

- Download the CircuitPython UF2 for Circuit Playground Bluefruit
Connect Circuit Playground Bluefruit 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

This project needs version 5.0.0-beta.0 or higher.

Download Circuit Python for Circuit Playground Bluefruit

Download Circuit Python Library Bundle

Download Adafruit CircuitPython Library Bundle

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

Required Libraries
adafruit_ble
adafruit_bluefruit_connect
adafruit_bus_device
neopixel.mpy
simpleio.mpy

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. The code will run properly when all of the files have been uploaded including libraries.
Use any text editor or favorite IDE to modify the code. We suggest using Mu as noted above.

```python
# SPDX-FileCopyrightText: Prof. John Gallaugher
# SPDX-FileCopyrightText: 2019 Noe Ruiz for Adafruit Industries
# SPDX-License-Identifier: MIT

# Code written by Prof. John Gallaugher, modified by Noe Ruiz for Adafruit Industries
# Adafruit Circuit Playground Express Bluefruit

import time
import board
import digitalio
import neopixel
from adafruit_ble import BLERadio
from adafruit_ble.advertising.standard import ProvideServicesAdvertisement
from adafruit_ble.services.nordic import UARTService
from audiopwmio import PWMAudioOut as AudioOut
from audiocore import WaveFile
from adafruit_bluefruit_connect.packet import Packet
from adafruit_bluefruit_connect.color_packet import ColorPacket
from adafruit_bluefruit_connect.button_packet import ButtonPacket

# setup pixels
pixels = neopixel.NeoPixel(board.NEOPIXEL, 10, brightness=1, auto_write=True)

# name colors so you don’t need to refer to numbers
RED = (255, 0, 0)
ORANGE = (255, 50, 0)
BLACK = (0, 0, 0)
GREEN = (0, 255, 0)
PURPLE = (100, 0, 255)
YELLOW = (255, 230, 0)
BLUE = (0, 0, 255)

# setup bluetooth
ble = BLERadio()
uart_service = UARTService()
advertisement = ProvideServicesAdvertisement(uart_service)

# External Audio Stuff
speaker_enable = digitalio.DigitalInOut(board.SPEAKER_ENABLE)
speaker_enable.direction = digitalio.Direction.OUTPUT
speaker_enable.value = True

audio = AudioOut(board.SPEAKER)  # Speaker
wave_file = None

def play_wav(name, loop=False):
    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 = WaveFile(wave_file)
audio.play(wave, loop=loop)
except OSError:
    pass # we'll just skip playing then

while True:
    # set CPXb up so that it can be discovered by the app
    ble.start_advertising(advertisement)
    while not ble.connected:
        pass

    # Now we're connected
    while ble.connected:
        if uart_service.in_waiting:
            try:
                packet = Packet.from_stream(uart_service)
            except ValueError:
                continue # or pass.

            if isinstance(packet, ColorPacket):  # check if a color was sent from
                pixels.fill(packet.color)
            if isinstance(packet, ButtonPacket):  # check if a button was pressed
                if packet.button == ButtonPacket.BUTTON_1:  # if button #1
                    pixels.fill(BLUE)
                    play_wav("bluefruit")
                    time.sleep(3)
                    pixels.fill(BLACK)
                if packet.button == ButtonPacket.BUTTON_2:  # if button #2
                    pixels.fill(ORANGE)
                    play_wav("halloween")
                    time.sleep(3)
                    pixels.fill(BLACK)
                if packet.button == ButtonPacket.BUTTON_3:  # if button #2
                    pixels.fill(PURPLE)
                    play_wav("muhaha")
                    time.sleep(2)
                    pixels.fill(BLACK)
                if packet.button == ButtonPacket.BUTTON_4:  # if button #2
                    pixels.fill(GREEN)
                    play_wav("neopixels")
                    time.sleep(3)
                    pixels.fill(BLACK)
                if packet.button == ButtonPacket.UP:  # if button #2
                    pixels.fill(YELLOW)
                    play_wav("organic")
                    time.sleep(2.6)
                    pixels.fill(BLACK)
                if packet.button == ButtonPacket.DOWN:  # if button #2
                    pixels.fill(PURPLE)
                    play_wav("python")
                    time.sleep(2)
                    pixels.fill(BLACK)
                if packet.button == ButtonPacket.LEFT:  # if button #2
                    pixels.fill(GREEN)
                    play_wav("smell")
                    time.sleep(2.5)
                    pixels.fill(BLACK)
                if packet.button == ButtonPacket.RIGHT:  # if button #2
                    pixels.fill(ORANGE)
                    play_wav("who")
                    time.sleep(2)
                    pixels.fill(BLACK)
Upload Sounds

Download the zip file containing the audio wave files using the link below. Create a new folder in the CIRCUITPY drive. Drag and drop the .wav files into the sounds directory.

Mapped Sounds

The audio files were voiced by Noe Ruiz and modified with a low-pitch effect. The audio files are assigned to the following buttons on the Bluefruit LE Connect App.

- UP – organic.wav
- DOWN – python.wav
- LEFT – smell.wav
- RIGHT – who.wav
- 1 – bluefruit.wav
- 2 – halloween.wav
- 3 – muahaha.wav
- 4 – neopixels.wav

Supported Audio Formats

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.

NeoPixel Colors

Supported colors are in the list below. Optionally create custom colors using RGB values. List of colors are located in the code on line 19.

```python
# name colors so you don't need to refer to numbers
RED = (255, 0, 0)
```
Bluefruit LE Connect App

Bluetooth App
This project uses the Adafruit Bluefruit LE connect app (available free for Android and iOS) to trigger the lights and sounds. It uses the control pad to trigger up to 8 different colors and sound effects. If you haven't downloaded the app yet, use the button below to install it on your mobile device.

[Download BLE Connect App iOS]
[Download BLE Connect App for Android]

Connect to Circuit Playground Bluefruit

Turn on the Circuit Playground Bluefruit by either connecting it via USB to your computer or with the 500mAh battery.

Using Bluefruit LE Connect App

Open the Bluefruit LE connect app and locate the device named CIRCUITPY and tap the connect button. Locate and tap on Controller. Under module, tab on Control Pad.

Use the arrow buttons or 1-4 buttons to trigger the NeoPixels. The audio files will play through the built-in speaker. The volume will be much louder when the STEMMA speaker is connected.
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.

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.

Slicing Parts
The parts were sliced using CURA using the slice settings below. The parts were 3D printed using PLA filament.

- PLA filament 220c extruder
- 0.2 layer height
- 10% gyroid infill
- 60mm/s print speed
- 60c heated bed
Glue Parts
The PCB mount and bottom cover need to be glued together. Join the flat surfaces together using super glue. Line up the mounting holes and notch before applying glue.

Attached Parts
The PCB mount and bottom cover are designed as separate parts to avoid using support material.

Toggle Switch

Wiring Toggle Switch
The slide switch will be wired to the male and female 2-pin JST cables. Using wire cutters, trim the wires short so they're the following length.

80mm (3.15inch)
Wired Toggle Switch
The voltage (red) wires are connected to the middle pin and either left or right pin on the switch. The ground (black) wires are connected to each other. Use piece of heat shrink tubing to insulate the exposed connection.

STEMMA Speaker

Connect Speaker to Circuit Playground Bluefruit
The STEMMA speaker is connected to the Circuit Playground Bluefruit using a 3-wire ribbon cable. Using wire cutters, trim the cable to the following length.

104mm (4.09 inches)

Prep Wires
Using wire strippers, remove a bit of insulation from the tips of each wire. Add a bit of solder to the tips to tin them. This will prevent the wires from fraying and make soldering easier.
Wired STEMMA Speaker
Solder the 3-wire cable to the pads on the back of the STEMMA speaker PCB. Reference the photo for best wire placement.

Solder the other end of the cable to the pads on the top of the Circuit Playground Bluefruit.

Assembly

Test Circuit
With the components wired up, it's time to connect them together and test out the circuit. Plug in the female JST connector from the switch to the Circuit Playground Bluefruit. Plug in the battery to the male JST connector to the switch. Flip the toggle switch to power on the Circuit Playground Bluefruit.

Pumpkin Screws
Locate the four screws on the bottom of the pumpkin. These screws hold the bottom cover to the pumpkin.
Remove Bottom
Using a Phillips screw driver, unfasten the four screws from the bottom cover. Keep the screws as you'll need them to secure the 3D printed mount to the pumpkin.

Test Fit 3D Printed Bottom
Now is a good time to test fit the 3D printed bottom cover. Orient the bottom cover so the notch and mounting holes are lined up. Press fit the mount into the recessed area on the bottom of the pumpkin. Remove the bottom once it has been test fitted.

Install STEMMA Speaker
The STEMMA speaker PCB snap fits into the center of the 3d printed mount. Insert the PCB at an angle and fit one side underneath one of the tabs. Press the PCB to snap fit it into place.
Installing Battery
The 500mAh battery will be placed in between the STEMMA speaker PCB and Circuit Playground Bluefruit PCB.

Battery Placement
Place the battery on top of the STEMMA speaker PCB. Adjust the cable so its position on top of the battery. Orientation of the battery can be adjusted later for better cable management.

Installing Circuit Playground Bluefruit
Place the CPB on top of the battery. Orient the CPB so the 3.3V and GND pads are lined up with the tabs. The mount is symmetrical so it can be position in either orientation. Insert the PCB at an angle and fit it underneath the tab with the protruding post. Place the CPB so one of the post fits into one of the pads.
Installed Circuit Playground Bluefruit
Adjust the wiring so it is not being pinched or kinked. Press the CPB down to snap fit it into the second tab.

The PCB mount is designed to be symmetrical so the Circuit Playground Bluefruit can fit in either orientation.

Wire Adjustments
Thoroughly inspect the components and wiring. There's just enough clearance for the components and wiring to be sandwiched in between the two PCBs.
Final Build

Drill Hole for Switch
In order to panel mount the toggle switch, a small hole will need to be drilled on the back of the pumpkin. I suggest using a rotary tool to create the hole.

Use a 1/4in (6.35mm) sized drill bit.

Secure Switch
Insert the toggle switch into the pumpkin and push it through the hole. Hold the switch in place while installing the washer. Tightly fasten the hex nut onto the threaded stem on the toggle switch. Pull the two JST connector out of the pumpkin so the battery and CPB can be connected.
Connect Battery to Switch
Grab the male connector from the switch and plug it into the JST connector on the 500mAh battery.

Connect Switch to Circuit Playground Bluefruit
Plug in the female JST connector from the switch to the JST port on the Circuit Playground Bluefruit. The length of the cables should be sufficient for the CPB to rest outside of the pumpkin.

Final Circuit Test
Before installing the 3D printed bottom onto the pumpkin, it's a good idea to test the circuit. Flip the switch to power on the Circuit Playground Bluefruit. Turn the circuit off once test is working and complete.
Installing Bottom
Fit the mount into the pumpkin with the notch and mounting holes lined up. Carefully fit the wiring into the pumpkin and try to avoid pinching or kinking the cables.
Secure Bottom

Grab the four screws and begin to fasten them into the mounting holes. Hold the pumpkin in place while fastening the screws.

Final Adjustments

Turn the circuit on and test out the lights and sounds. The wires from the battery and switch might be visible inside the pumpkin. You can adjust them if you’d like but it may become tedious. I found it a bit difficult to hide all of the wiring so I just left it as it. Feel free to make any adjustments.