MIDI Stomping Pads
Created by Liz Clark

https://learn.adafruit.com/midi-stomping-pads

Last updated on 2022-02-09 11:06:59 AM EST
# Table of Contents

## Overview
- Parts

## Circuit Diagram
- Wiring Connections

## 3D Printing
- Housing the Circuit Playground Express

## CircuitPython
- Install or update CircuitPython!
- Further Information

## Coding the MIDI Stomping Pads
- Upload the Code and Libraries to the Circuit Playground Express
- How the CircuitPython Code Works

## Wiring

## Assembly

## Stomp!
- Additional Uses
Overview

Sometimes you just need to let out your emotions in a healthy way with music. This project lets you build 3D printed stomping pads to send MIDI notes with a Circuit Playground Express running CircuitPython.

The two 3D printed squares have pieces of copper foil sheet on each side. Wires are soldered to the copper foil to create a digital input circuit with the Circuit Playground Express. One side is connected to GND and the other is connected to an input pin.

The 3D printed squares connect to each other with springs with the help of some hot glue so that you can press down with purpose to send MIDI notes.
The NeoPixels on the Circuit Playground Express change color depending on which input is pressed.

This project was inspired by the Garageband and Makey Makey Stomping Pads (https://adafruit.it/XnE) project by Frazer Merrick.

Parts

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

Copper Foil Sheet with Conductive Adhesive - 12" x12" Sheet
Copper sheet can be an interesting addition to your toolbox. The sheet itself is made of thin pure copper so its extremely flexible and can take on nearly any shape. You can easily...
https://www.adafruit.com/product/4607
1 x Compression springs
Compression springs for the stomp pads

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

Silicone Cover Stranded-Core Wire - 30AWG in Various Colors
Silicone-sheathing wire is super-flexible and soft, and its also strong! Able to handle up to 200°C and up to 600V, it will do when PVC covered wire wimps out. We like this wire...
https://www.adafruit.com/product/2051

Circuit Diagram

Wiring Connections

Solder four copper pads to GND and four copper pads to input pins A1, A2, A3 and A4.

- Pad 1 to pin A4
• Pad 2 to GND
• Pad 3 to GND
• Pad 4 to pin A1
• Pad 5 to GND
• Pad 6 to pin A2
• Pad 7 to GND
• Pad 8 to pin A3

3D Printing

The MIDI Stomping Pads are assembled with 3D printed squares. They print with no supports. Two squares will be needed to assemble each stomping pad.

The cpxMidiStompingPad.stl can be downloaded directly here or from Thingiverse.

cpxMidiStompingPad.stl
https://adafru.it/XoC

Thingiverse download
https://adafru.it/XoD
The 3D printed square has a raised section in the middle to place the copper foil. There are also slots to place the springs in the corners and a slot to run the wire to the Circuit Playground Express.

Housing the Circuit Playground Express

The Circuit Playground Express can be housed in this 3D printed case by the Ruiz Brothers, found in this Learn Guide (https://adafru.it/CkV).

Alternatively, you can use the Adafruit Circuit Playground Enclosure (https://adafru.it/DoV) sold in the Adafruit store.

Case for Circuit Playground Learn Guide

https://adafru.it/CkV
Adafruit Circuit Playground Express or Bluefruit Enclosure
We've got nice cases for many of our beloved boards, but the Circuit Playground Express and https://www.adafruit.com/product/3915

CircuitPython

As we continue to develop CircuitPython and create new releases, we will stop supporting older releases. If you are running an older version of CircuitPython, you need to update. Click the button below to download the latest!

Install or update CircuitPython!

Follow this quick step-by-step for super-fast Python power :)

Download the latest version of CircuitPython for this board via CircuitPython.org

https://adafruit.it/cp-cpx

Click the link above and download the latest UF2 file

Download and save it to your Desktop (or wherever is handy)
Plug your Circuit Playground Express into your computer using a known-good USB cable

A lot of people end up using charge-only USB cables and it is very frustrating! So make sure you have a USB cable you know is good for data sync.

Double-click the small Reset button in the middle of the CPX, you will see all of the LEDs turn green. If they turn all red, check the USB cable, try another USB port, etc.

(If double-clicking doesn't do it, try a single-click!)

You will see a new disk drive appear called CPLAYBOOT.

Drag the adafruit-circuitpython-etc...uf2 file onto it.
The CPLAYBOOT drive will disappear and a new disk drive will appear called CIRCUITPY
That’s it! You’re done :)

Further Information

For more detailed info on installing CircuitPython, check out [Installing CircuitPython](https://adafruit.it/Amd).

Coding the MIDI Stomping Pads

Once you’ve finished setting up your Circuit Playground Express with CircuitPython, you can access the code and necessary libraries by downloading the Project Bundle.

To do this, click on the Download Project Bundle button in the window below. It will download as a zipped folder.

```python
# SPDX-FileCopyrightText: 2021 Liz Clark for Adafruit Industries
# SPDX-License-Identifier: MIT

import board
from digitalio import DigitalInOut, Direction, Pull
import neopixel
import usb_midi
import adafruit_midi
from adafruit_midi.note_on import NoteOn
from adafruit_midi.note_off import NoteOff

# USB MIDI setup
midi = adafruit_midi.MIDI(midi_out=usb_midi.ports[1], out_channel=0)

# NeoPixel setup
pixels = neopixel.NeoPixel(board.NEOPIXEL, 10, brightness=0.02, auto_write=False)

# setup for digital inputs
inputs = []

for pin in cpx_pins:
    cpx_pin = DigitalInOut(pin)
    cpx_pin.direction = Direction.INPUT
    cpx_pin.pull = Pull.UP
    inputs.append(cpx_pin)
```

©Adafruit Industries
# NeoPixel colors
GREEN = (0, 255, 0)
CYAN = (0, 255, 255)
BLUE = (0, 0, 255)
PURPLE = (180, 0, 255)

# debounce states for inputs
state_A1 = False
state_A2 = False
state_A3 = False
state_A4 = False

# array of NeoPixel colors
colors = [GREEN, CYAN, BLUE, PURPLE]

# array of MIDI notes
midi_notes = [60, 64, 67, 72]

# array of input states
input_states = [state_A1, state_A2, state_A3, state_A4]

while True:
    # iterate through colors, inputs and MIDI notes
    for i in range(4):
        # reset the state of the input and send NoteOff msg
        # after the input is released
        if inputs[i].value and input_states[i]:
            input_states[i] = False
            midi.send(NoteOff(midi_notes[i], 120))

        # if an input is pressed...
        if not inputs[i].value and not input_states[i]:
            # change the colors of the NeoPixels
            pixels.fill(colors[i])
            pixels.show()
            # send the NoteOn msg
            midi.send(NoteOn(midi_notes[i], 120))
            # update input state
            input_states[i] = True

Upload the Code and Libraries to the Circuit Playground Express

After downloading the Project Bundle, plug your Circuit Playground Express into the computer's USB port. You should see a new flash drive appear in the computer's File Explorer or Finder (depending on your operating system) called CIRCUITPY. Unzip the folder and copy the following items to the Circuit Playground Express's CIRCUITPY drive.

- lib folder
- code.py

Your Circuit Playground Express CIRCUITPY drive should look like this after copying the lib folder and the code.py file.
How the CircuitPython Code Works

There are arrays of digital input pins, pin states, MIDI note numbers and NeoPixel colors setup in the code before the loop. Each of these arrays contain four indexes that correspond with each other. This will allow you to iterate through these arrays in the loop.

To customize your setup, you can change the defined NeoPixel colors in the `colors` array and the MIDI note numbers in the `midi_notes` array.
# setup for digital inputs
inputs = []

for pin in cpx_pins:
    cpx_pin = DigitalInOut(pin)
    cpx_pin.direction = Direction.INPUT
    cpx_pin.pull = Pull.UP
    inputs.append(cpx_pin)

# NeoPixel colors
GREEN = (0, 255, 0)
CYAN = (0, 255, 255)
BLUE = (0, 0, 255)
PURPLE = (180, 0, 255)

# debounce states for inputs
state_A1 = False
state_A2 = False
state_A3 = False
state_A4 = False

# array of NeoPixel colors
colors = [GREEN, CYAN, BLUE, PURPLE]
# array of MIDI notes
midi_notes = [60, 64, 67, 72]
# array of input states
input_states = [state_A1, state_A2, state_A3, state_A4]

In the loop, \( i \) represents the selected index in each of the arrays. The `for` statement lets you setup the code so that you can check for the state of the different digital inputs. When a state change is detected, aka when a stomp pad is stomped, the NeoPixel color will change and a MIDI note message will be sent accordingly.

```python
# iterate through colors, inputs and MIDI notes
for i in range(4):
    # reset the state of the input and send NoteOff msg
    # after the input is released
    if inputs[i].value and input_states[i]:
        input_states[i] = False
        midi.send(NoteOff(midi_notes[i], 120))

    # if an input is pressed...
    if not inputs[i].value and not input_states[i]:
        # change the colors of the NeoPixels
        pixels.fill(colors[i])
        pixels.show()
        # send the NoteOn msg
        midi.send(NoteOn(midi_notes[i], 120))
        # update input state
        input_states[i] = True
```
Cut two squares of the copper foil sheet that are each 5.5 cm wide by 5.5 cm long. You can use a pencil to mark your measurement on the back and then use scissors to cut them out.
Cut two lengths of wire that are at least 12 inches long each. Splice and tin both ends of the wires with solder.

Solder one wire to each of the copper foil squares.
Solder one wire to pin A1 on the Circuit Playground Express. Solder the second wire to GND on the Circuit Playground Express.

Repeat this process for the remaining stomping pads, soldering three copper squares to GND and three copper squares to pins A2, A3 and A4.

Alternatively, you could have this be a solder-free project by using alligator clips and copper foil tape.
Assembly
Remove the protective backing from the copper foil squares and place them on the 3D printed square pads. The 3D printed squares have slots for the wires to sit in.
Dab hot glue onto the top of a spring and attach it to the indented corner of the 3D printed square. Repeat this for the three other springs.

Dab hot glue onto the tops of the secured springs. Place the springs into the corners of the second 3D printed square.

Repeat this process for the remaining stomping pads.
Stomp!

Attach your Circuit Playground Express via USB to your preferred MIDI instrument. You can use a digital audio workstation (DAW) on your computer or use a USB MIDI passthrough to connect to a hardware instrument (like a robot xylophone (https://adafruit.it/XoE)).

Stomp, slap, bang or pound on the stomping pads to send beautiful music out into the world. You can customize the code with your preferred MIDI notes. Think about experimenting with different instruments too, such as drums or weird sound effects.

Additional Uses

The stomping pad mechanism can also be used with your feet, really opening up the musical possibilities. Since this project is setup as a series of digital inputs, you could also adjust the code to be an HID device or any other project that utilizes inputs. There’s also potential for accessibility uses with the stomping pads.