Deco Two-Key Feather Macro Pad

Created by John Park

https://learn.adafruit.com/deco-two-key-keypad-macropad-circuitpython-feather

Last updated on 2022-03-02 12:08:53 PM EST
## Table of Contents

**Overview**
- Parts 4
- Feather 5

**CircuitPython for Feather nRF52840** 7
- Set up CircuitPython Quick Start!

**Code the Deco Keypad** 9
- Text Editor 9
- Download the Project Bundle 10
- Custom Key Mapping 11
- How It Works 11

**Build the Deco Keypad** 14
- Assembly 14
- Case and Base 16
Overview

Minimal mechanical macropads are fun and useful. Perfect for playing rhythm games (Osu!), media controls, mic/camera muting, and more. You can customize the Deco Keypad to press any key or key-combo, as well as pick your favorite colors for the under-lit NeoPixels.

Build your own using the NeoKey FeatherWing, two mechanical keyswitches with keycaps, and the CircuitPython-capable Feather of your choice.

The Art Deco-inspired, 3D printed case has ports for USB, STEMMA QT cable, and the reset button. Add rubber bumper feet for a nice, stable macropad experience!
Parts

NeoKey FeatherWing - Two Mechanical Key Switches with NeoPixels
The only thing better than a nice mechanical key, is two of them, and ones that also can glow any color of the rainbow - and that's what the Adafruit NeoKey FeatherWing...
https://www.adafruit.com/product/4979

Kailh Mechanical Key Switches - 10 packs - Cherry MX Compatible
For crafting your very own custom keyboard, these Kailh mechanical key switches are deeeee-luxe!Come in a pack of 10 switches, plenty to make a...
https://www.adafruit.com/product/4996
Translucent Keycaps for MX Compatible Switches - 10 pack
Get ready to clacky to your heart's content. Here is a 10 pack of translucent keycaps for your next mechanical keyboard or
https://www.adafruit.com/product/4956

Feather

You can use pretty much any CircuitPython-capable Feather for this project.

NOTE: The code for this project sends keystrokes over a USB cable, not Bluetooth Low Energy. That's entirely possible if you'd like to adjust the code accordingly, just not the goal of this particular project.

Adafruit Feather nRF52840 Express
The Adafruit Feather nRF52840 Express is the new Feather family member with Bluetooth Low Energy and native USB support featuring the nRF52840! It's...
https://www.adafruit.com/product/4062

Any Feather that can run CircuitPython will be totally fine for this project:
Adafruit Feather M4 Express - Featuring ATSAMD51
It's what you've been waiting for, the Feather M4 Express featuring ATSAMD51. This Feather is fast like a swift, smart like an owl, strong like a ox-bird (it's half ox,...
https://www.adafruit.com/product/3857

Adafruit Feather M0 Express
At the Feather M0’s heart is an ATSAMD21G18 ARM Cortex M0+ processor, clocked at 48 MHz and at 3.3V logic, the same one used in the new
https://www.adafruit.com/product/3403

Adafruit Feather RP2040
A new chip means a new Feather, and the Raspberry Pi RP2040 is no exception. When we saw this chip we thought "this chip is going to be awesome when we give it the Feather...
https://www.adafruit.com/product/4884

Header Kit for Feather - 12-pin and 16-pin Female Header Set
These two Female Headers alone are, well, lonely. But pair them with any of our
https://www.adafruit.com/product/2886
Pink and Purple Woven USB A to USB C Cable - 2 meters long
This cable is not only super-fashionable, with a woven pink and purple Blinka-like pattern, it’s also made for USB C for our modernized breakout boards, Feathers and more.
https://www.adafruit.com/product/5044

Little Rubber Bumper Feet - Pack of 4
Keep your electronics from going barefoot, give them little rubber feet! These small sticky bumpers are our favorite accessory for any electronic kit or device. They are sticky, but...
https://www.adafruit.com/product/550

CircuitPython for Feather nRF52840

CircuitPython (https://adafruit.it/tB7) is a derivative of MicroPython (https://adafruit.it/BeZ) designed to simplify experimentation and education on low-cost microcontrollers. It makes it easier than ever to get prototyping by requiring no upfront desktop software downloads. Simply copy and edit files on the CIRCUITPY drive to iterate.

The following instructions will show you how to install CircuitPython. If you’ve already installed CircuitPython but are looking to update it or reinstall it, the same steps work for that as well!

Set up CircuitPython Quick Start!

Follow this quick step-by-step for super-fast Python power :)
Click the link above to download the latest UF2 file.

Download and save it to your desktop (or wherever is handy).

Plug your Feather nRF52840 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 Reset button next to the USB connector on your board, and you will see the NeoPixel RGB LED turn green (identified by the arrow in the image). If it turns red, check the USB cable, try another USB port, etc. Note: The little red LED next to the USB connector will pulse red. That's ok!

If double-clicking doesn't work the first time, try again. Sometimes it can take a few tries to get the rhythm right!
You will see a new disk drive appear called FTHR840BOOT.

Drag the adafruit_circuitpython_etc.uf2 file to FTHR840BOOT.

The LED will flash. Then, the FTHR840BOOT drive will disappear and a new disk drive called CIRCUITPY will appear.

That's it, you're done! :)

---

**Code the Deco Keypad**

**Text Editor**

Adafruit recommends using the Mu editor for editing your CircuitPython code. You can get more info in [this guide](https://adafruit.it/ANO).

Alternatively, you can use any text editor that saves simple text files.
Download the Project Bundle

Your project will use a specific set of CircuitPython libraries and the code.py file. In order to get the libraries you need, click on the Download Project Bundle link below, and uncompress the .zip file.

Drag the contents of the uncompressed bundle directory onto your Feather board’s CI RCUITPY drive, replacing any existing files or directories with the same names, and adding any new ones that are necessary.

```
# SPDX-FileCopyrightText: Copyright (c) 2021 John Park for Adafruit
# SPDX-License-Identifier: MIT
# Deco Keypad

import time
import board
from digitalio import DigitalInOut, Pull
from adafruit_debouncer import Debouncer
import usb_hid
from adafruit_hid.keyboard import Keyboard
from adafruit_hid.keyboard_layout_us import KeyboardLayoutUS
from adafruit_hid.keycode import Keycode
import neopixel

print("- Deco Keypad -")
time.sleep(1)  # Sleep for a bit to avoid a race condition on some systems

# ----- Keymap ----- 
# change as needed, e.g. capital A (Keycode.SHIFT, Keycode.A)
switch_a_output = Keycode.Z
switch_b_output = Keycode.X

# ----- Keyboard setup -----  
keyboard = Keyboard(usb_hid.devices)
keyboard_layout = KeyboardLayoutUS(keyboard)  # We're in the US :) 

# ----- Key setup -----  
switch_a_in = DigitalInOut(board.D5)
switch_b_in = DigitalInOut(board.D6)
switch_a_in.pull = Pull.UP
switch_b_in.pull = Pull.UP
switch_a = Debouncer(switch_a_in)
switch_b = Debouncer(switch_b_in)

# ----- NeoPixel setup ----- 

MAGENTA = 0xFF00FF
CYAN = 0x0088DD
WHITE = 0xCCCCCC
BLACK = 0x000000

d pixel_pin = board.D9
pixels = neopixel.NeoPixel(pixel_pin, 2, brightness=1.0)
pixels.fill(BLACK)
time.sleep(0.3)
pixels.fill(WHITE)
time.sleep(0.3)
pixels.fill(BLACK)
time.sleep(0.3)
pixels[0] = MAGENTA
```
while True:
    switch_a.update()  # Debouncer checks for changes in switch state
    switch_b.update()

    if switch_a.fell:
        keyboard.press(switch_a_output)
        pixels[0] = WHITE
    if switch_a.rose:
        keyboard.release(switch_a_output)
        pixels[0] = MAGENTA

    if switch_b.fell:
        keyboard.press(switch_b_output)
        pixels[1] = WHITE
    if switch_b.rose:
        keyboard.release(switch_b_output)
        pixels[1] = CYAN

Custom Key Mapping

To adjust the key mapping of the keys, look for this section of the code:

```python
# ----- Keymap ----- #
# change as needed, e.g. capital A (Keycode.SHIFT, Keycode.A)
switch_a_output = Keycode.Z
switch_b_output = Keycode.X
```

`switch_a` is the keyswitch closer to the USB port. You can use single keystrokes, or multi-key combos as shown in the comment.

For more info on using USB HID keycodes, [check out this documentation](https://adafru.it/MuD).

How It Works

Libraries

The code first imports libraries to help out with a number of tasks, including `time`, `board` for pin definitions, `digitalio` and `adafruit_debouncer` to simplify key press and release detection, `usb_hid` and `adafruit_hid.keyboard`, plus `neopixel`.

```python
import time
import board
from digitalio import DigitalInOut, Pull
from adafruit_debouncer import Debouncer
import usb_hid
from adafruit_hid.keyboard import Keyboard
```
from adafruit_hid.keyboard_layout_us import KeyboardLayoutUS
from adafruit_hid.keycode import Keycode
import neopixel

Keyboard Setup

This section sets the keymapping, keyboard setup, and pin setup with digitalio and debouncer.

```
switch_a_output = Keycode.Z
switch_b_output = Keycode.X

# ----- Keyboard setup ----- #
keyboard = Keyboard(usb_hid.devices)
keyboard_layout = KeyboardLayoutUS(keyboard)  # We're in the US :)

# ----- Key setup ----- #
switch_a_in = DigitalInOut(board.D5)
switch_b_in = DigitalInOut(board.D6)
switch_a_in.pull = Pull.UP
switch_b_in.pull = Pull.UP
switch_a = Debouncer(switch_a_in)
switch_b = Debouncer(switch_b_in)
```

NeoPixels

You'll create some color definitions and set up the two on-board NeoPixels next. Once they're set up, you'll give them a little startup blink so you know when the board has been reset.

```
# ----- NeoPixel setup ----- #
MAGENTA = 0xFF00FF
CYAN = 0x0088DD
WHITE = 0xC0C0C0
BLACK = 0x000000

pixel_pin = board.D9
pixels = neopixel.NeoPixel(pixel_pin, 2, brightness=1.0)
pixels.fill(BLACK)
time.sleep(0.3)
pixels.fill(WHITE)
time.sleep(0.3)
pixels.fill(BLACK)
time.sleep(0.3)
pixels[0] = MAGENTA
pixels[1] = CYAN
```

Main Loop

The main code loop uses the debouncer to check both switches.
When either switch is pressed, the **debouncer** notices that it **fell** (voltage goes from high to low since we're using the built-in pull up resistor on each digital pin). Then, the `keyboard.press()` command sends the keyboard output for the associated key, and sets the NeoPixel to white.

When the **debouncer** registers that it **rose**, this means the key has been released, and the `keyboard.release()` command is sent for that keycode. The NeoPixel color returns to its default for that key.

```python
switch_a.update()  # Debouncer checks for changes in switch state
switch_b.update()
if switch_a.fell:
    keyboard.press(switch_a_output)
    pixels[0] = WHITE
if switch_a.rose:
    keyboard.release(switch_a_output)
    pixels[0] = MAGENTA
if switch_b.fell:
    keyboard.press(switch_b_output)
    pixels[1] = WHITE
if switch_b.rose:
    keyboard.release(switch_b_output)
    pixels[1] = CYAN
```
Build the Deco Keypad

Assembly

Feather Headers
First, you'll prepare your Feather by soldering on the header sockets. This guide has details (https://adafru.it/DRi) on soldering Feather headers.
FeatherWing Pins
Next, solder on the headers to the FeatherWing. *This guide section (https://adafru.it/DRi)* show how to solder plain headers.

Be sure to double-check the orientation before soldering. The FeatherWing will insert into the Feather with the flat side of the 'Wing exposed on top.
Press the 'Wing into the Feather as shown.

Case and Base

The case is designed to hold the keyswitches securely (just like the switch plate of a typical mechanical keyboard) while protecting the electronics and looking stylish.

There are openings in the case for USB port, STEMMA QT, if you have plans for expansion over I2C, and the reset button.

The base snaps onto the bottom and give you some surface area to stick on some rubber bumper feet to prevent sliding.

You can print the model files linked below on a 3D printer, or have them printed through a service.

Deco Keypad 3D model files
https://adafruit.it/Tc9
Fit Feather in Case
Align the Feather's USB port with the larger opening on the case, then push the stacked FeatherWing/Feather through the bottom of the case. It should be a fairly snug fit.
Keyswitches

Insert the keyswitches from the top of the case into the FeatherWing -- be careful to align the pins of the keyswitches with the sockets of the FeatherWing to avoid bending the delicate legs.

You should push the keyswitches in firmly to make a solid connection with the sockets by squeezing up on the Feather/FeatherWing and down on the keyswitches.

Once the keyswitches are in place, place the case on your work surface and push down on both keyswitches to click them into the case's integrated switchplate.
Base of Case

"All that she wants is another keyswitch..."
(Oh, wait, that's Ace of Base.)

Fit the base onto the bottom of the case.

To prevent sliding, put rubber bumper feet onto the base.
Keycaps
Press your keycaps onto the keyswitch stems. You can use any MX compatible keycaps you like -- both translucent and windowed keycaps work particularly well for seeing the NeoPixel under-lighting.

Now, you're ready to play some [Osu!](https://adafru.it/Tca) or other one or two-button rhythm game!

Or, you can use the Deco Keypad for an keyboard shortcut, such as copy/paste, volume -/+s, and more.