Pico Four Keypad

Created by John Park

https://learn.adafruit.com/pico-four-key-macropad

Last updated on 2023-11-08 04:19:48 PM EST
# Table of Contents

## Overview
- Parts

## Installing CircuitPython
- CircuitPython Quickstart
- Flash Resetting UF2

## Installing Libraries
- Installing the Adafruit HID Library

## Build the Keypad
- Snap in Keys
- Wiring Switches
- Pico Wiring
- Insulate
- Close It Up
- Add Keycaps
- Deploy the Pico Four Keypad!

## Code the Four Keypad
- Text Editor
- CircuitPython Code
- Testing
- HID Keyboard Basics
- Customization
Overview

Select all. Cut. Copy. Paste!

Much like the now infamous Stack Overflow The Key copy/paste keypad, this small, beautiful keypad is designed to speed up your most important tasks!

You can build it with some hand wired mechanical keyswitches (no PCB or diode matrix required), and control it from a Raspberry Pi Pico RP2040 running CircuitPython.

The gorgeous silver aluminum enclosure houses it and you'll top it off with some delightfully gumdrop-y MA profile keycaps.
Parts

**Raspberry Pi Pico RP2040**
The Raspberry Pi foundation changed single-board computing when they released the Raspberry Pi computer, now they're ready to...
https://www.adafruit.com/product/4864

**Four Key Silver Aluminum Keypad Shell Enclosure**
For all the DIY keebfans in the house here is a super sleek and sturdy 4-Key Aluminum Keypad Shell in Silver, the perfect base to begin your custom...
https://www.adafruit.com/product/5073

**Kailh Mechanical Key Switches - 10 packs - Cherry MX Compatible**
For crafting your very own custom keyboard, these Kailh mechanical key switches are deeeeee-luxe! Come in a pack of 10 switches, plenty to make a...
https://www.adafruit.com/product/4996
**Orange MA Keycaps for MX Compatible Switches - 5 pack**
Dress up your mechanical keys in your favorite colors with a wide selection of gumdrop-like, retro, curvy, and stylish MA profile keycaps. Here is a 5 pack of Orange MA...
https://www.adafruit.com/product/5175

**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

**High Temperature Polyimide Tape - 1cm wide x 33 meter roll**
Polyimide Tape (sometimes referred to by the brand name Kapton Tape) is an interesting addition to your toolbox! Polyimide Tape remains stable across...
https://www.adafruit.com/product/3057

---

**Installing CircuitPython**

CircuitPython () is a derivative of MicroPython () 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.
CircuitPython Quickstart

Follow this step-by-step to quickly get CircuitPython working on your board.

Download the latest version of CircuitPython for the Raspberry Pi Pico from circuitpython.org

Click the link above and download the latest UF2 file.

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

Start with your Pico unplugged from USB. Hold down the BOOTSEL button, and while continuing to hold it (don't let go!), plug the Pico into USB. Continue to hold the BOOTSEL button until the RPI-RP2 drive appears!

If the drive does not appear, unplug your Pico and go through the above process again.

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.
You will see a new disk drive appear called RPI-RP2.

Drag the adafruit_circuitpython_etc.uf2 file to RPI-RP2.

The RPI-RP2 drive will disappear and a new disk drive called CIRCUITPY will appear.

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

Flash Resetting UF2

If your Pico ever gets into a really weird state and doesn't even show up as a disk drive when installing CircuitPython, try installing this 'nuke' UF2 which will do a 'deep clean' on your Flash Memory. You will lose all the files on the board, but at least you'll be able to revive it! After nuking, re-install CircuitPython

flash_nuke.uf2
Installing Libraries

Alongside the core CircuitPython libraries (which are baked into CircuitPython), you'll also add the Adafruit HID Library to add keyboard features.

Installing the Adafruit HID Library

Download the [library bundle](#) here.

Copy the adafruit_hid folder from the bundle to the lib folder on your CIRCUITPY drive.

![Folder Structure]

Before continuing make sure your board's lib folder has the adafruit_hid library folder copied over.

Build the Keypad

![Keypad Image]
Building the keypad involves snapping in your key switches, wiring/soldering them to the Pico, and closing it all up.

Snap in Keys
First, use a small screwdriver to remove the base plate.

Then, snap in four keyswitches. It helps to use a consistent orientation as shown here.
Wiring Switches

Each switch needs to have one leg wired to common ground, and the other to a GPIO pin on the Pico.

Start by soldering black wire to connect the ground legs of each switch together (either can be the ground, so just pick one side and be consistent for neatness sake). Solder a few inches of wire to the end of one ground leg to run to the Pico.

Solder a wire about 4-5" (10cm) in length to the free leg on each switch. These will run to their own GPIO pins respectively on the Pico.
Pico Wiring
Now, run the wiring to the Pico.
Black wire to GND
Blue to GP0
Green to GP1
Yellow to GP2
White to GP3
Be sure to skip the ground pin on the Pico that site between the GP1 and GP2 pins -- you can tell it's a ground because of the flat end on the copper pad.

Insulate
Add some Kapton tape to the exposed pads and side of the enclosure to avoid any shorts.
Close It Up

Use one screw to fasten the acrylic base to the enclosure. (The other screw hole is obscured by the Pico board.)

Add two rubber feet to the bottom.
Add Keycaps
Place your keycaps on the key switch stems and press down. You’re ready to use the Pico Four Keypad!

Deploy the Pico Four Keypad!
Plug it into your computer via USB cable, and now you can select all, cut, copy, and paste each with just a touch of the button!
Code the Four Keypad

Text Editor

Adafruit recommends using the Mu editor for using your CircuitPython code with the Pico. You can get more info in [this guide](#).

Alternatively, you can use any text editor that saves files.

CircuitPython Code

Copy the code below and paste it into Mu. Then, save it to your Pico as code.py.

```python
# SPDX-FileCopyrightText: 2021 john park for Adafruit Industries
# SPDX-License-Identifier: MIT
# Pico Four Key USB HID Keypad

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

kpd = Keyboard(usb_hid.devices)
# Choose the correct modifier key for Windows or Mac.
```

©Adafruit Industries
# Comment one line and uncomment the other.
# MODIFIER = Keycode.CONTROL  # For Windows
MODIFIER = Keycode.COMMAND  # For Mac

# define buttons
NUM_KEYS = 4
PINS = (board.GP0,
        board.GP1,
        board.GP2,
        board.GP3,
    )

KEYMAP = (
    ("Select all", [MODIFIER, Keycode.A]),
    ("Cut", [MODIFIER, Keycode.X]),
    ("Copy", [MODIFIER, Keycode.C]),
    ("Paste", [MODIFIER, Keycode.V]),
)

keys = []
for pin in PINS:
    dio = DigitalInOut(pin)
    dio.pull = Pull.UP
    keys.append(Debouncer(dio))

print("Welcome to keypad")
print("keymap:"
for k in range(NUM_KEYS):
    print("t", (KEYMAP[k][0]))

while True:
    for i in range(NUM_KEYS):
        keys[i].update()
        if keys[i].fell:
            print(KEYPAN[k][0])
            kpd.send(*KEYMAP[i][1])

Testing

Even before we've wired up the mechanical keyswitches, you can test the code by shorting each of the four GPIO pins used to ground. This is the same as pressing a key. Just be careful not to short power to ground!

Use a text editor with some text in it to watch as your keys select all, cut, copy, and paste!

USB keyboards and mice show up on your computer as 'HID' devices, which stands for 'Human Interface Device'

HID Keyboard Basics

This guide page has a great intro to CircuitPython HID Keyboard.
For even more details, check out the documentation at https://circuitpython.readthedocs.io/projects/hid/en/latest/ which includes all of the keycodes and media codes you can use.

By importing the adafruit_hid library into the program, you can make calls to send keyboard keys and media keys.

**Customization**

You can have your keypad type other keys or key combos if you like. This is the section of code to adjust:

```python
KEYMAP = (
    ("Select all", [MODIFIER, Keycode.A]),
    ("Cut", [MODIFIER, Keycode.X]),
    ("Copy", [MODIFIER, Keycode.C]),
    ("Paste", [MODIFIER, Keycode.V]),
)
```

The string is only there for debugging/testing purposes. The section inside the brackets is where you can put your own keys and combos. Just remember to add a comma between each key in a combo.