![Adafruit logo]

I Vote(d) Pin

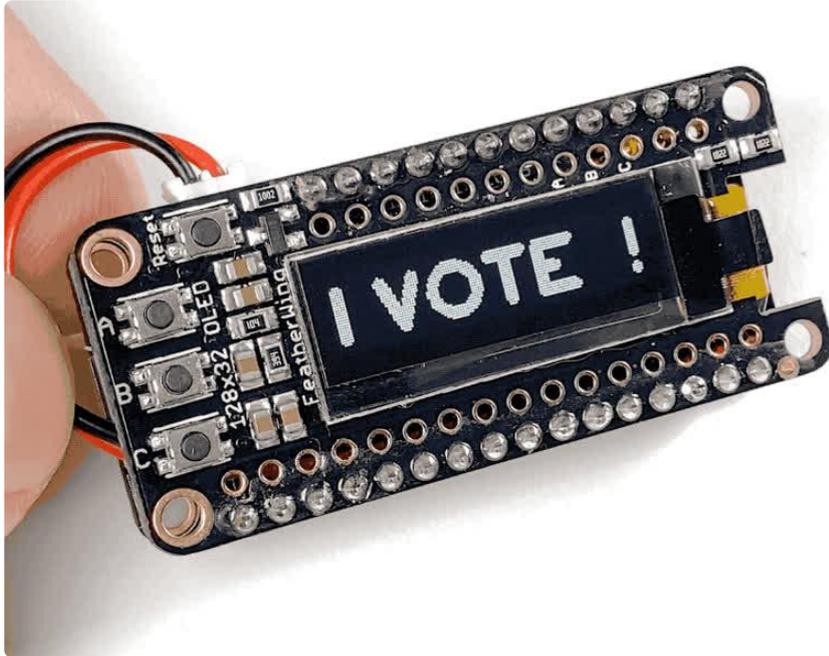Created by Collin Cunningham

https://learn.adafruit.com/i-vote-d-pin
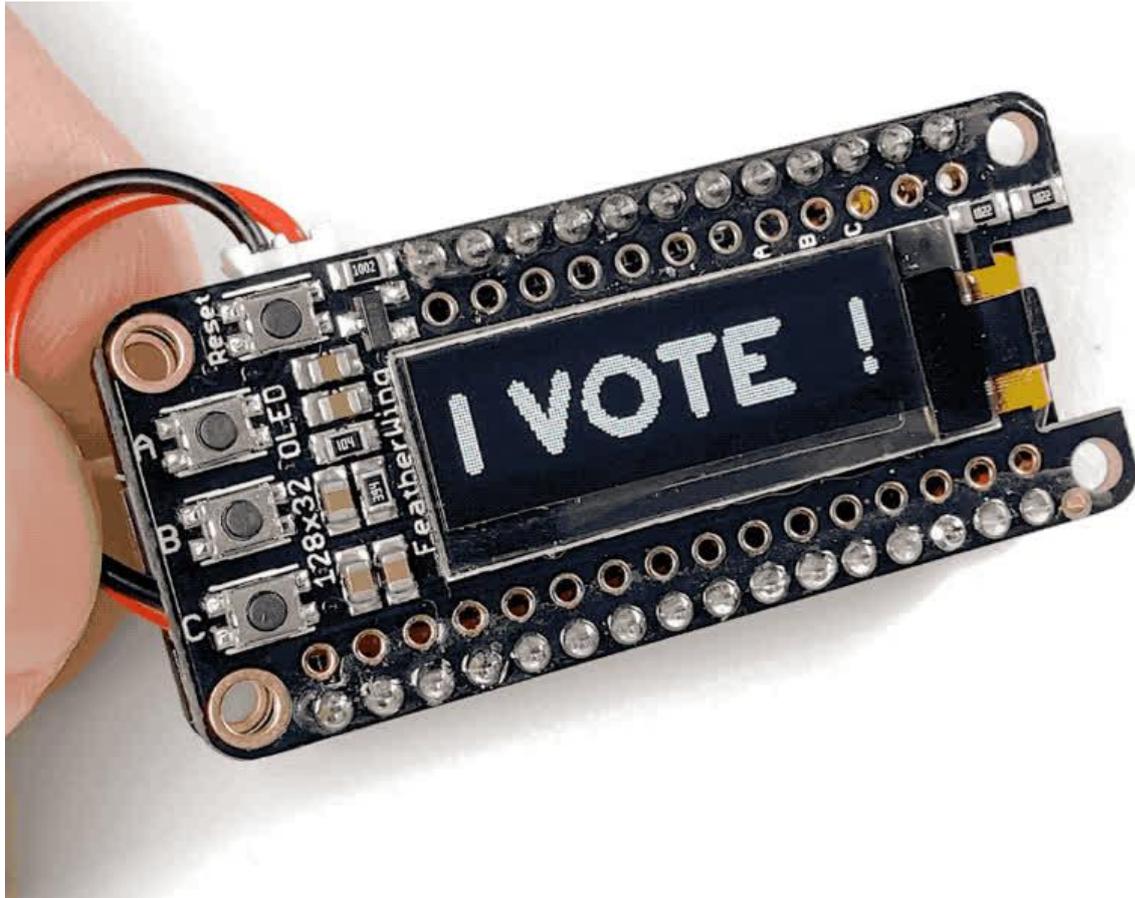
Last updated on 2022-12-01 03:56:07 PM EST

Table of Contents

Overview

"I voted" stickers are so last election. This year, kick things up several notches by using a medium that commands real attention - electronics!

Spread the word by creating your own interactive election wearable using a Feather + OLED Featherwing. Once you've cast your ballot, tap a button to change the text from "I VOTE" to "I VOTED!" and show your democratic participation off to the world.

What you'll need
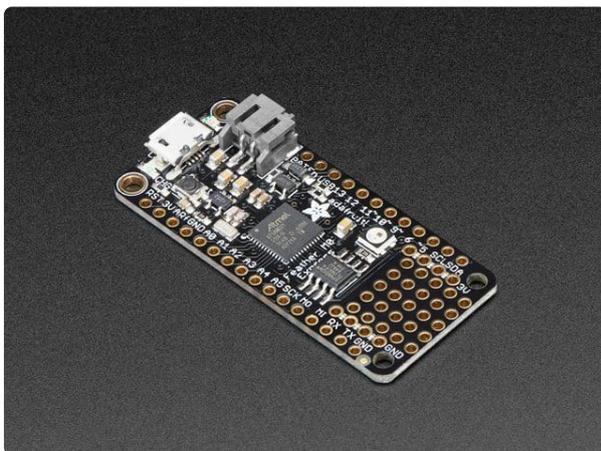
Adafruit Feather M0 Express
At the Feather M0's heart is an ATSAMD21G18 ARM Cortex M0+ processor, clocked at 48 MHz and at 3.3V logic, the same one used in the new
https://www.adafruit.com/product/3403
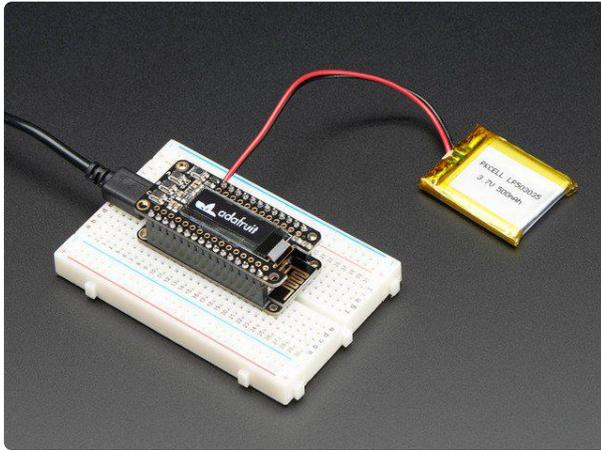
Adafruit FeatherWing OLED - 128x32 OLED Add-on For Feather

A Feather board without ambition is a Feather board without FeatherWings! This is the FeatherWing OLED: it adds a 128x32 monochrome OLED plus 3 user buttons to...

https://www.adafruit.com/product/2900

---

1 x Short Male Headers
Short Feather Male Headers - 12-pin and 16-pin Male Header Set

https://www.adafruit.com/product/3002

---

1 x Short Female Headers
Short Headers Kit for Feather - 12-pin + 16-pin Female Headers

https://www.adafruit.com/product/2940

---

1 x Lipoly Battery
Lithium Ion Polymer Battery

https://www.adafruit.com/product/3898

---

1 x Pin Back
Silver Pin Back with Foam Adhesive

https://www.adafruit.com/product/2838

---

1 x USB Micro Cable
For programming Feather

https://www.adafruit.com/product/2185

---

1 x Soldering Iron
Adjustable 30W 110V soldering iron

https://www.adafruit.com/product/180

---

1 x Solder
Mini Solder spool - 60/40 lead rosin-core solder 0.031" diameter

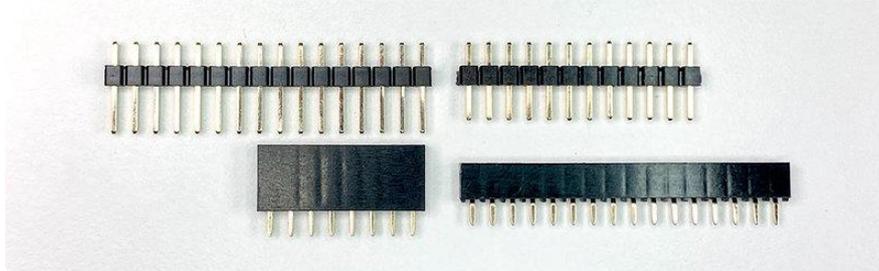https://www.adafruit.com/product/145

---

1 x Breadboard
Half-size breadboard

https://www.adafruit.com/product/64

Hardware Assembly

Headers

In order to keep a low profile on this wearable, we'll connect the Feather & OLED Featherwing using short headers. Using these will reduce the pin's height by about 3 mm vs normal headers, so the boards will lie flatter against a shirt or lapel. That being said, you can use normal headers instead – the pin will just be a bit more awkward to wear.
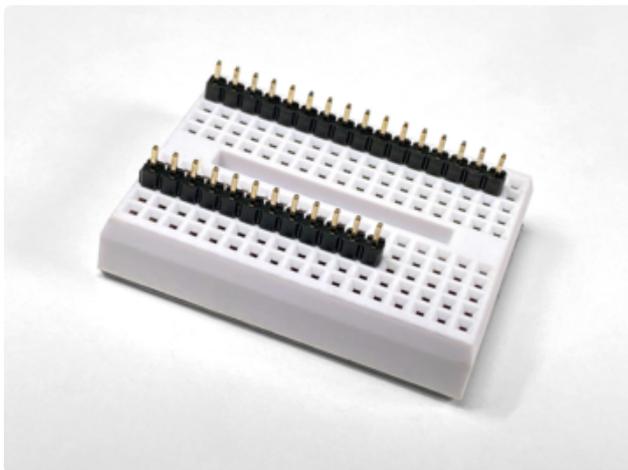
Insert the male header pins into a spare breadboard in order to keep them stable and straight while soldering.
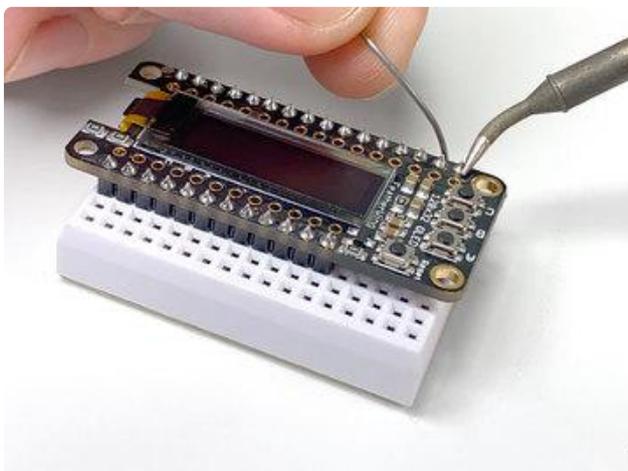
Place the OLED Featherwing onto the pins.

Solder each pin to it's surrounding pad on the top side of the Featherwing.

Remove the OLED Featherwing from the breadboard and flip the Featherwing upside down. Temporarily mount the female header pins on the Featherwing to keep them in place while soldering to the Feather.

Flip the Feather upside down and place it onto the short pins of the female header.
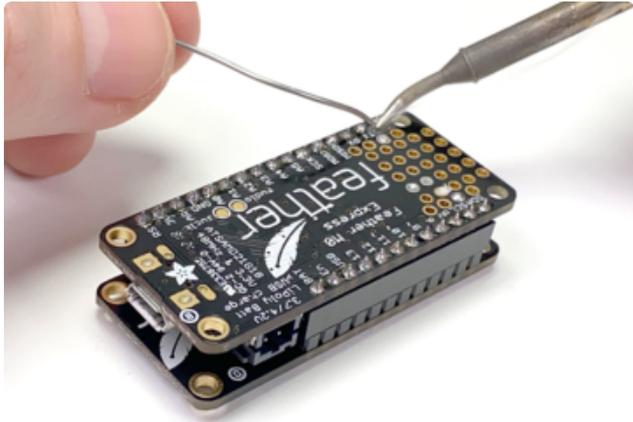
Solder the short pins of the female header to the pads on the Feather.

Battery
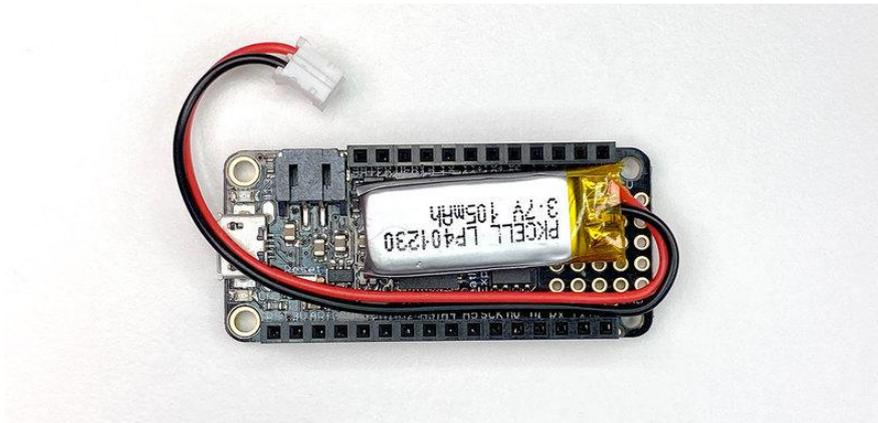
Disconnect the Feather from the Featherwing and place the Lipoly battery lengthwise between the Feather's female headers with the JST connector hanging over the Feather's USB jack.

Reconnect the OLED Featherwing to the Feather while being careful not to pinch any part of the battery in the process. Lipoly batteries are somewhat delicate and can be dangerous if punctured!

Connect the battery's JST connector to the Feather's power jack. You can now charge the battery by connecting the Feather to a USB power supply

Pin back

Peel the protective layer off of the pin back's adhesive side and attach it to the back of the Feather. Placing the pin back as seen above will help prevent the pin from being top heavy and tipping forward while in use. Press the pin back firmly in place to ensure it's secure.

Code

Feather M0 Express comes preloaded with CircuitPython. If you've since used it to run Arduino code, or you'd like to upgrade to the latest version, follow the instructions here to install the latest CircuitPython. ()

Connect to computer

Connect Feather M0 Express to your computer using a micro USB Cable. A drive named CIRCUITPY should appear on your computer.

Open the CIRCUITPY drive and create a folder named lib inside (if it doesn't already exist).

Libraries

The code for this project requires four software libraries. Click the link below to download the CircuitPython library bundle which matches the version of CircuitPython

you are running. You can check the boot_out.txt file on the CIRCUITPY drive to determine the major version of CircuitPython you are using.

<div align="center">

Library Bundle

</div>

Unzip the library bundle, and open the lib folder inside.

You'll need to copy four libraries from this folder to the CIRCUITPY drive's lib folder:

- Locate the folder named adafruit_bitmap_font and copy it to the CIRCUITPY drive's lib folder.
- Locate the file named adafruit_debouncer.mpy and copy it to the CIRCUITPY drive's lib folder.
- Locate the folder named adafruit_display_text and copy it to the CIRCUITPY drive's lib folder.
- Locate the file named adafruit_displayio_ssd1306.mpy and copy it to the CIRCUITPY drive's lib folder.

Your CIRCUITPY drive's file structure should now look like this:
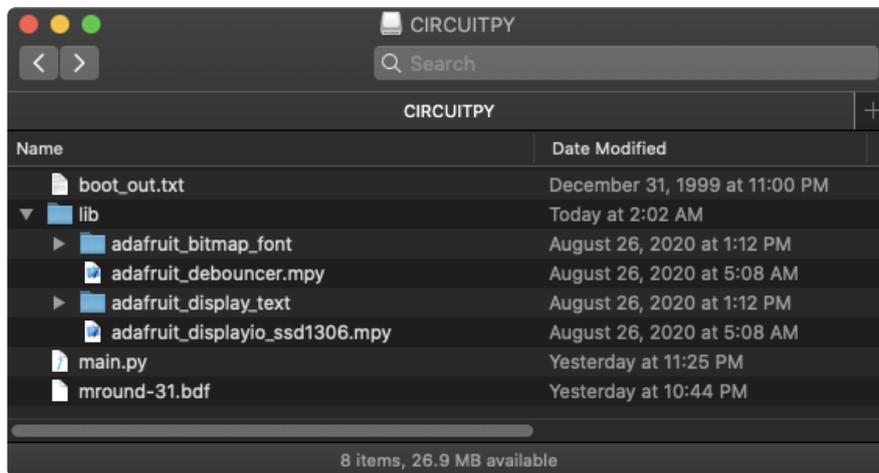
Copy the code below and paste it into a new text file.

Save the text file as code.py to the root of the CIRCUITPY drive, overwriting any preexisting file.

```
# SPDX-FileCopyrightText: 2020 Collin Cunningham for Adafruit Industries
#
# SPDX-License-Identifier: MIT

import board
from adafruit_debouncer import Debouncer
import digitalio
import displayio
```

```python
from adafruit_display_text import label
import adafruit_displayio_ssd1306
from adafruit_bitmap_font import bitmap_font

displayio.release_displays()

# Set up button pins
pin_a = digitalio.DigitalInOut(board.D9)
pin_a.direction = digitalio.Direction.INPUT
pin_a.pull = digitalio.Pull.UP

pin_b = digitalio.DigitalInOut(board.D6)
pin_b.direction = digitalio.Direction.INPUT
pin_b.pull = digitalio.Pull.UP

pin_c = digitalio.DigitalInOut(board.D5)
pin_c.direction = digitalio.Direction.INPUT
pin_c.pull = digitalio.Pull.UP

button_a = Debouncer(pin_a) #9
button_b = Debouncer(pin_b) #6
button_c = Debouncer(pin_c) #5

# Load font
font = bitmap_font.load_font('/mround-31.bdf')

# Set up display & add group
i2c = board.I2C()
display_bus = displayio.I2CDisplay(i2c, device_address=0x3C)
display = adafruit_displayio_ssd1306.SSD1306(display_bus, width=128, height=32)
group = displayio.Group()
display.show(group)

# Add content to group
default_text = "I VOTE  !"
text_area = label.Label(font, text=default_text, color=0xFFFFFF, x=0, y=17)
group.append(text_area)

while True:

    # Debounce buttons
    button_a.update()
    button_b.update()
    button_c.update()

    # Check for button presses & set text
    if button_a.fell:
        text_area.text = default_text
        text_area.x = 0
    elif button_b.fell:
        text_area.text = "I VOTED!"
        text_area.x = 0
    elif button_c.fell:
        text_area.text = "DID U?"
        text_area.x = 18

    display.show(group)
```

Usage & customization

Once you've saved code.py to your Feather M0 Express, you should see some white text flash across the screen before the larger I VOTE ! text appears. Use the OLED Featherwing's A, B, & C buttons to control what text appears on the display:

- Button A: I VOTE !
- Button B: I VOTED!
- Button C: DID U?

It's easy to customize these strings if you want the display to say something different. Open the code.py file and find line 43:

```
default_text = "I VOTE  !"

text_area = label.Label(font, text=default_text, color=0xFFFFFF, x=0, y=17)
group.append(text_area)
```

This is where the initial I VOTE ! text is defined as `default_text`. You can change the default text by editing the string within the quotation marks on line 43. This initial text is defined separately from the others because it's used in two different places – when defining the `text_area` you see first, and when button A is pressed by the user.

Speaking of buttons, go to line 54 to see the code which executes when each of the buttons are pressed:

```
if button_a.fell:
    text_area.text = default_text
    text_area.x = 0
elif button_b.fell:
    text_area.text = "I VOTED!"
    text_area.x = 0
elif button_c.fell:
    text_area.text = "DID U?"
    text_area.x = 18
```

Whenever a button press or `fell` is detected, the `text_area.text` variable is changed.

The text for button B can be changed on line 58 & the text for button C can be changed on line 61.

Learn More

Of course, this is just the beginning of what you can do with Feather and the OLED Featherwing. If you're interested in learning more about CircuitPython, check out the full guide here (). Or to jump right into learning about how to control the OLED Featherwing (), head over to the Displayio library guide () for all the details.