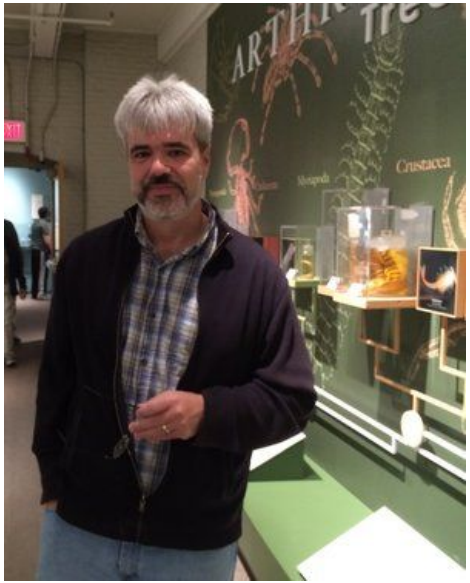
Programmable Instrumentation with Circuit Python

Head Neuroengineer:
Ed Soucy PhD



Neuroengineer:
Brett Graham PhD

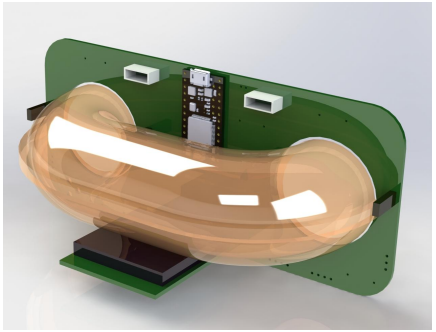
Harvard Center for Brain Science Neurotechnology Core

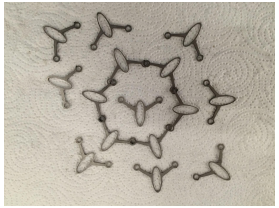
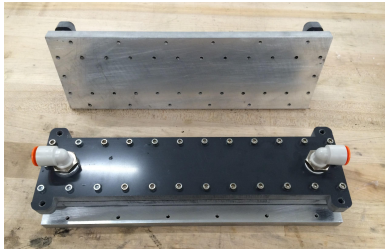
Technology can help advance science

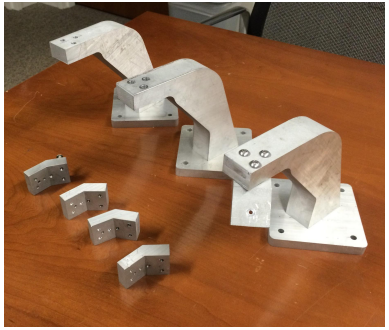
The Neurotechnology core offers:

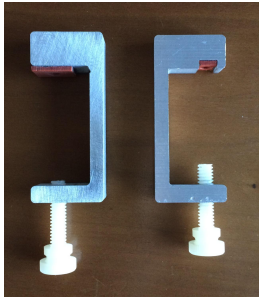
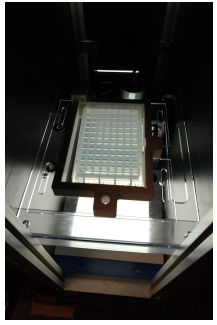
- A modern fabrication and prototyping shop (makerspace)
- Training on tools, instrument design, data analysis and general scientific consulting
- Design and fabrication of novel instrumentation

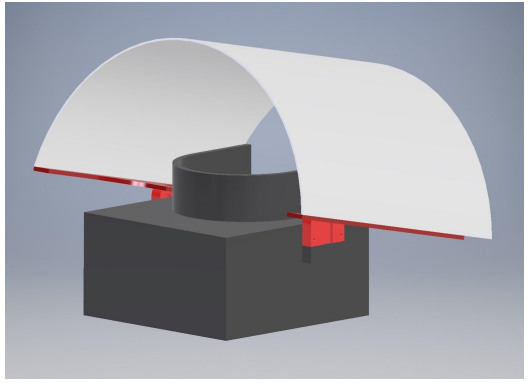
We are generalist, scientist, engineers that help will all technical aspects of science



Example: Measuring reaction time

How quickly can a person press a button when queued by a light flash or tone?

How might we measure this?

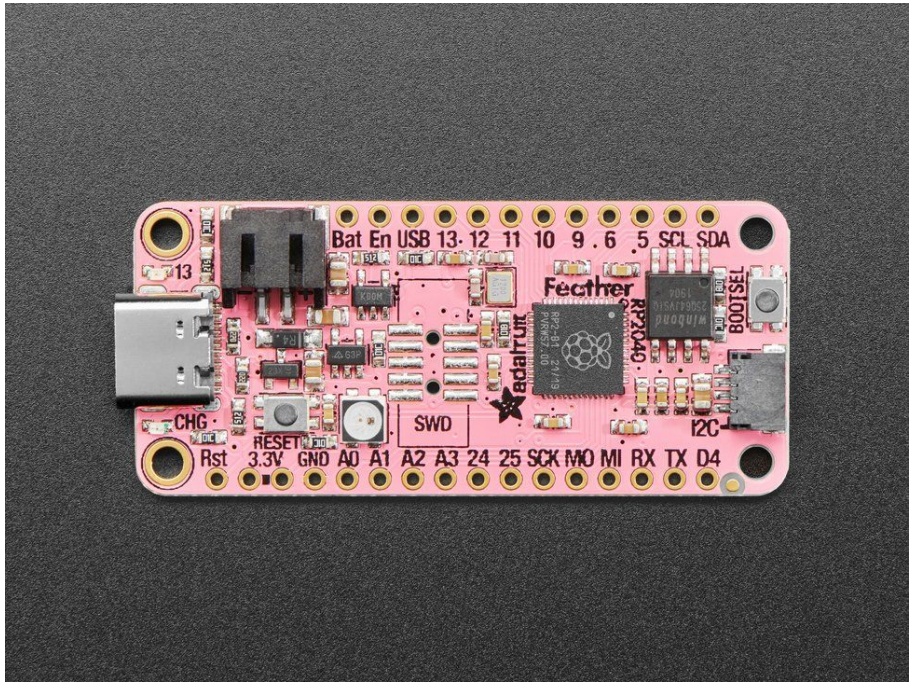
What is a microcontroller? (RP2040)

Small **programmable** computer

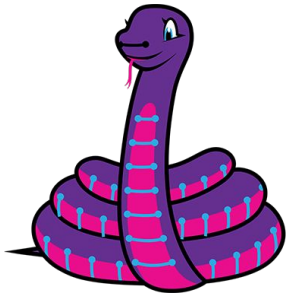
- Dual core ~125 MHz
- 264 kB RAM
- 8 MB storage (Flash)

No OS = more control

Lots of ‘low level’ IO



What is CircuitPython?



circuit python

Adaptation of Python language to microcontrollers

Officially supported by Adafruit electronics (adafruit.com)

https://docs.circuitpython.org/en/7.3.x/README.html

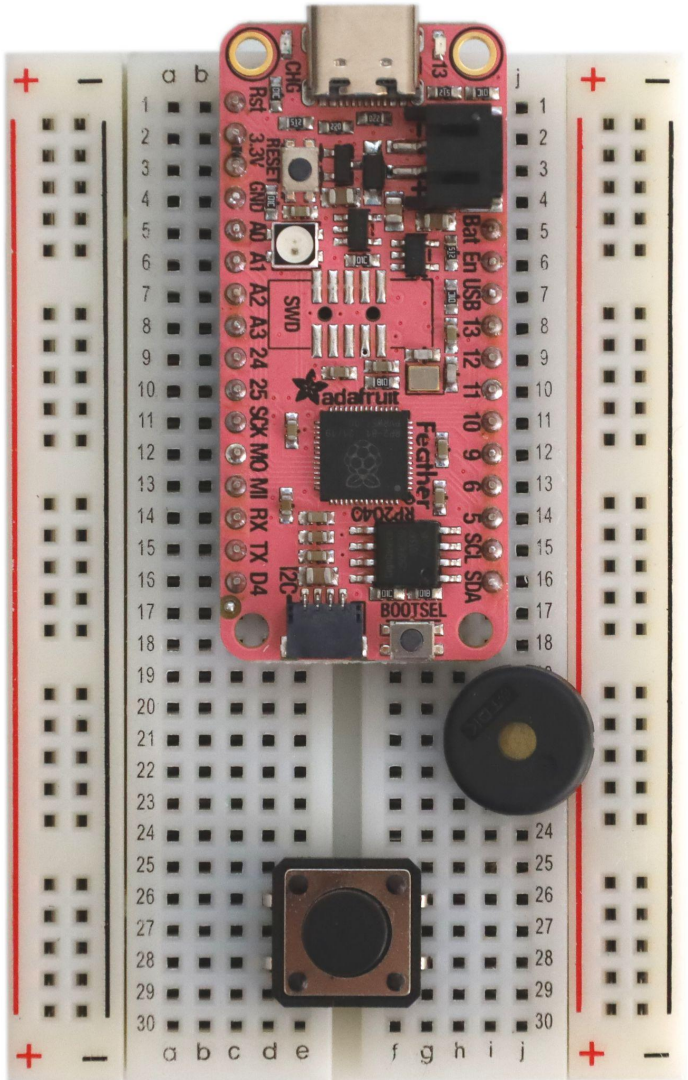
Alternative to Arduino (which uses C/C++) boards can often be used with either

Any text editor works, we'll use Mu: https://codewith.mu/

Building your instrument

Solderless Breadboard

Good for testing your circuit

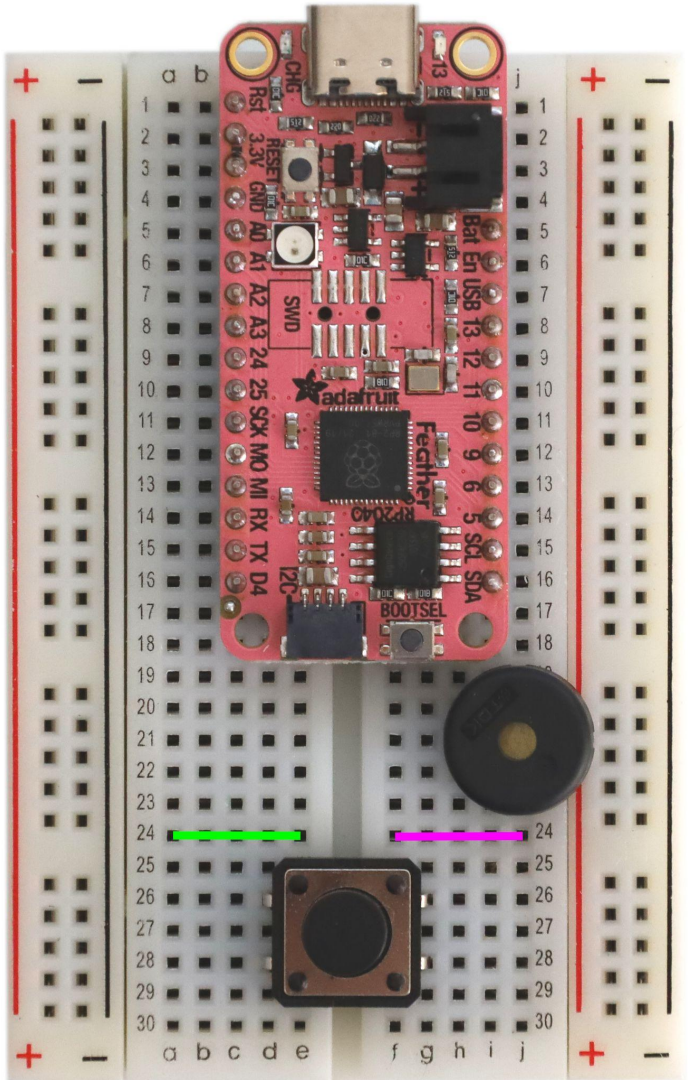


Breadboard connections

Rows of sockets are connected

24 a b c d e : all connected (green)

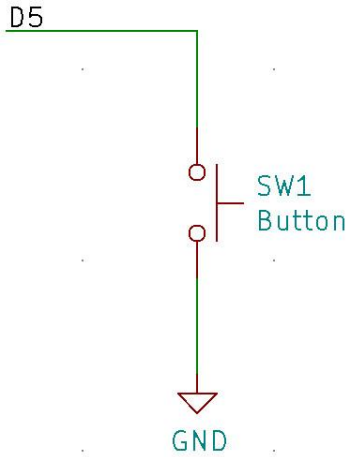
24 f g h i j : all connected (pink)

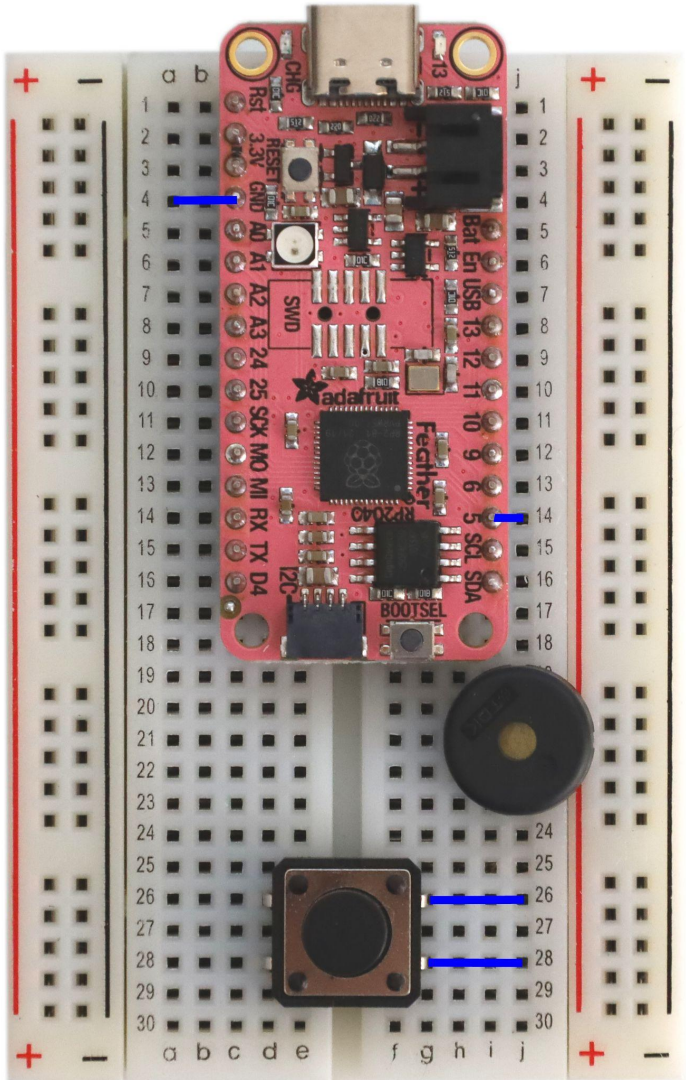


Button circuit

Requires 2 connections:

- Digital input (pin 5)
- Ground (GND)

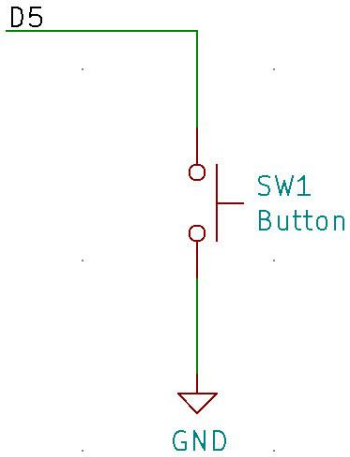


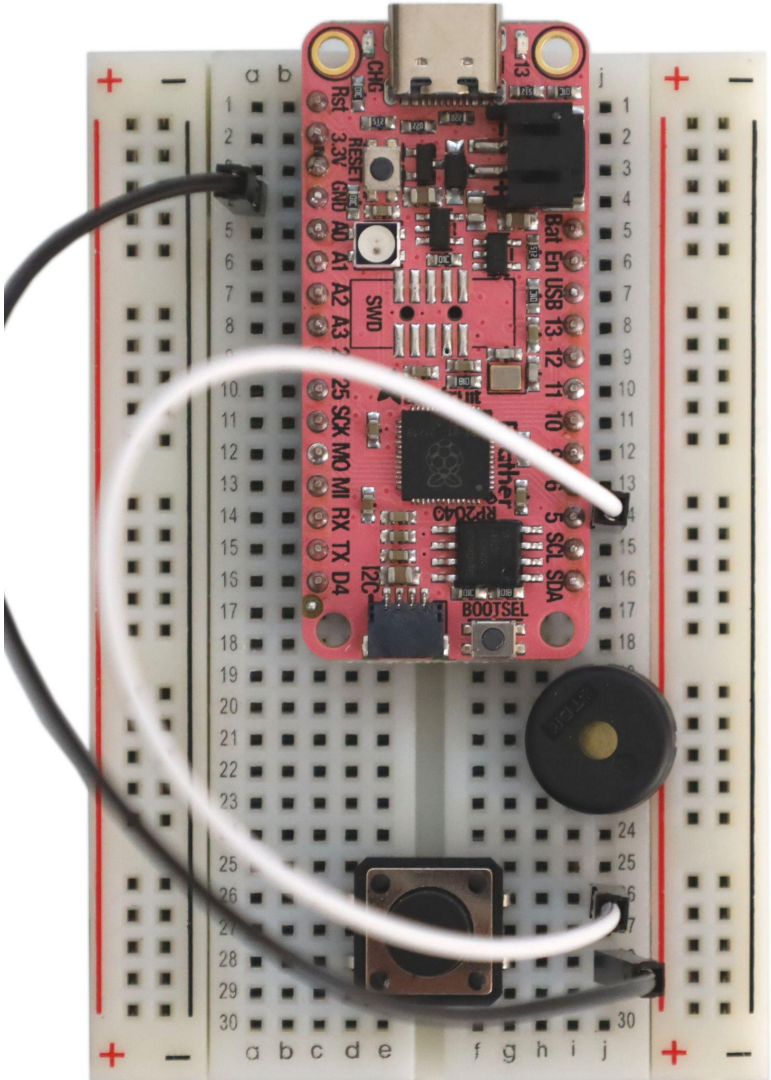


Wiring

Connect 2 wires

- 4 a to 28 j
- 14 j to 26 j





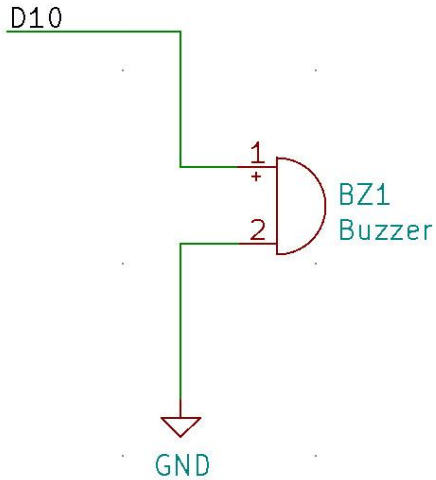
Buzzer circuit

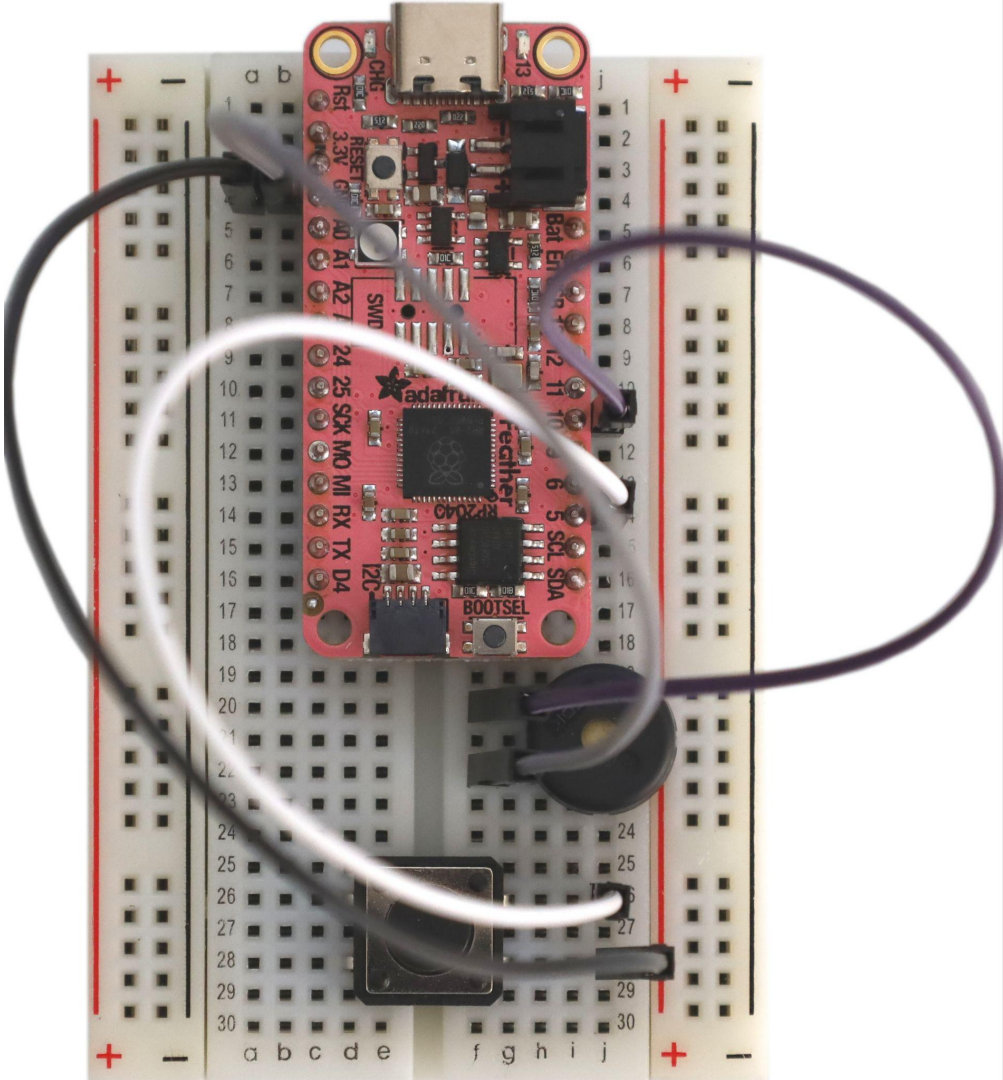
Requires 2 connections

- Digital output (pin 10)
- Ground (GND)

Run 2 wires

- 11 j to 20 f
- 4 b to 22 f





Verifying your wiring

Your boards are pre-programmed to light an LED when the button is pressed

Wire up your board, plug it into your computer and give it a try!

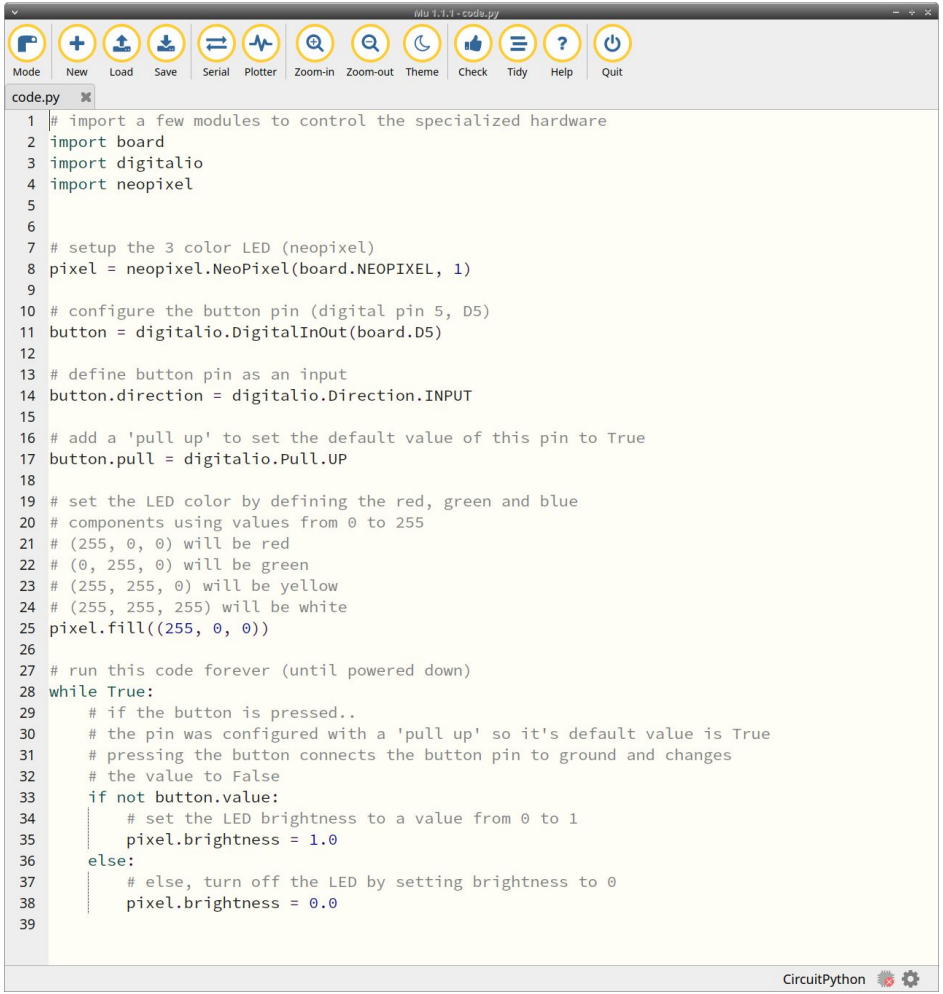
Programming

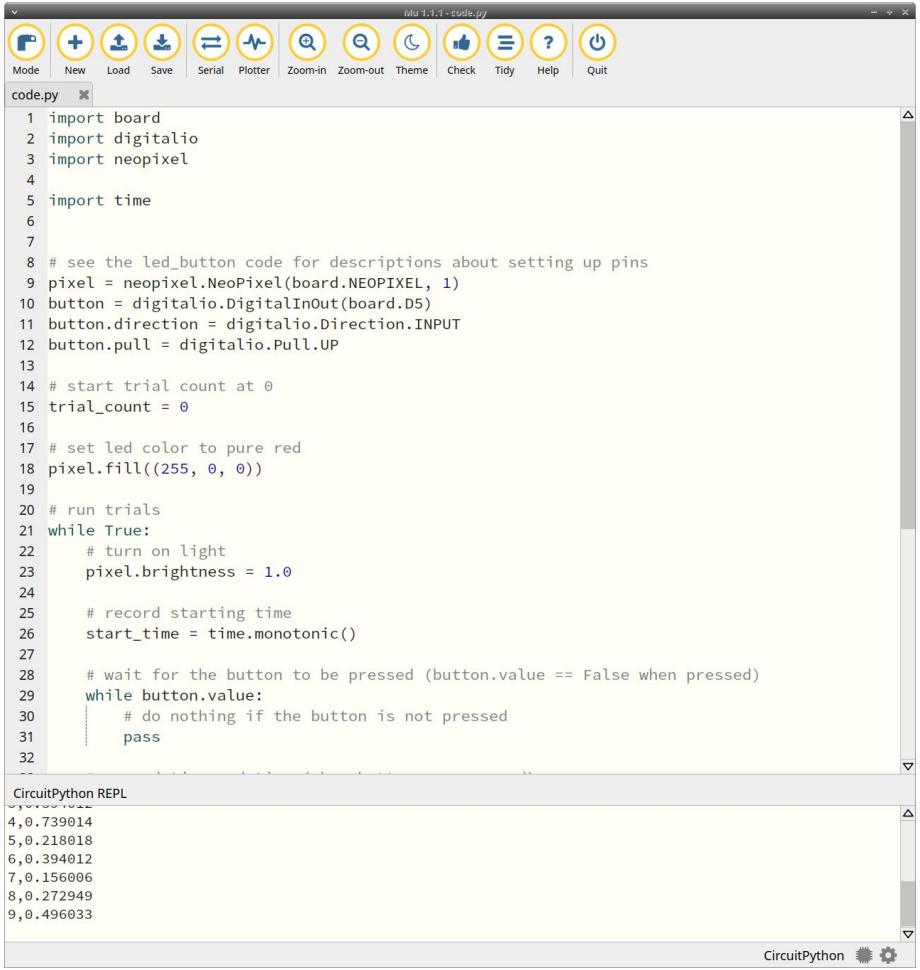
Appears as USB drive

‘code.py’ is plain text

Editing this text automatically resets the device (if all goes well)

Also appears as a ‘serial’ device which will report errors and other text output (like your measured reaction times)




Mode
New
Load
Save
Serial
Plotter
Zoom-in
Zoom-out
Theme
Check
Tidy
Help
Quit
code.py
1 import board
2 import digitalio
3 import neopixel
4
5 import time
6
7
8 # see the led_button code for descriptions about setting up pins
9 pixel = neopixel.NeoPixel(board.NEOPIXEL, 1)
10 button = digitalio.DigitalInOut(board.D5)
11 button.direction = digitalio.Direction.INPUT
12 button.pull = digitalio.Pull.UP
13
14 # start trial count at 0
15 trial_count = 0
16
17 # set led color to pure red
18 pixel.fill((255, 0, 0))
19
20 # run trials
21 while True:
22 # turn on light
23 pixel.brightness = 1.0
24
25 # record starting time
26 start_time = time.monotonic()
27
28 # wait for the button to be pressed (button.value == False when pressed)
29 while button.value:
30 # do nothing if the button is not pressed
31 pass
32
CircuitPython REPL
4,0.739014
5,0.218018
6,0.394012
7,0.156006
8,0.272949
9,0.496033
CircuitPython

Modifications

Measure average reaction time

Does your reaction time change across trials?

Use tone instead of light

Change light color and/or brightness

Change tone frequency

What strategies might a participant use to ‘cheat’ the experiment? If you use these strategies can you change your measured reaction time?

Where to learn more about electronics?

https://docs.circuitpython.org/en/7.3.x/README.html

https://learn.adafruit.com/

https://learn.sparkfun.com/

https://www.arduino.cc/

https://github.com/HMS-RIC/ArduinoNanocourse

The Art of Electronics

Us:

brett_graham@fas.harvard.edu

soucy@mcb.harvard.edu