

# Basics of Hardware with CircuitPython

---

# Hello!

## I am Kira Hartlage

---



- Mechanical engineering background
- Consumer appliance design
- Self-taught software engineer
- Embedded software development

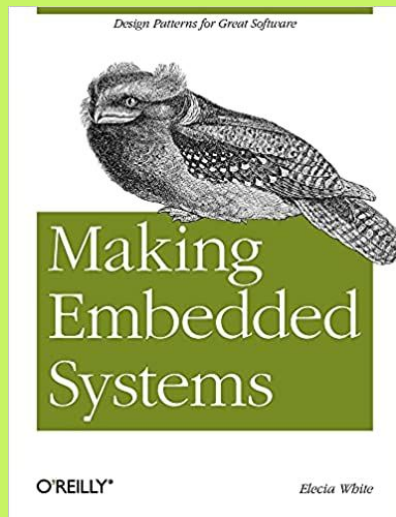
# Desired Outcomes

---

- Learn a bit about embedded software development
- Learn a bit about hardware
- Learn how you can get started using CircuitPython

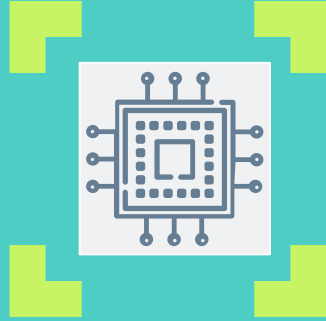
1.

# Embedded System



An embedded system is a computerized system that is purpose-built for its application.

- Elecia White



# Microcontroller

An integrated circuit with a processor, memory,  
and various input/output peripherals

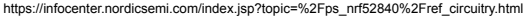
# What is a microcontroller?

---

1. Instructions - what the microcontroller is capable of doing
2. Registers - fast storage locations in memory that the microcontroller has access to that the instructions can use to do things
3. Memory - store code or data (usually slower than accessing registers)



\_\_\_\_\_





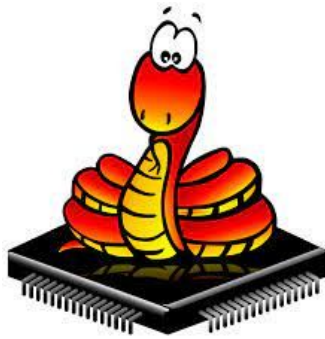
# 2.

# CircuitPython



# CircuitPython

---



MicroPython - full Python compiler and runtime environment that runs on bare-metal (i.e. no operating system)



CircuitPython - easy to learn derivative of MicroPython created to get users up and running as quickly as possible

# Writing code for hardware

---

## Challenges

- ▣ Limited memory (requiring C, assembly, or other low-level languages)
- ▣ No operating system (running on bare-metal)
- ▣ Startup code (lots of setup required just to get going and get an LED to blink)
- ▣ Really large (1000+ pages) manuals that are not user friendly
- ▣ Compilers, linkers, special equipment to flash the micro etc.

## CircuitPython

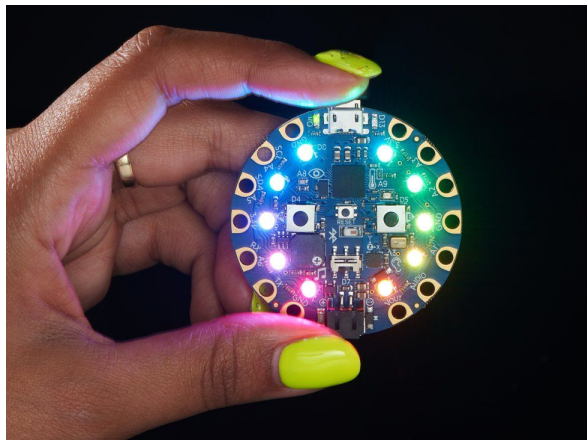
- ▣ Designed for beginners and people already familiar or new to Python
- ▣ Can purchase boards with microcontrollers already running CircuitPython (no setup required!)
  - ▣ Plug and play with USB
- ▣ No special desktop software needed
  - ▣ Can edit in your text editor of choice
  - ▣ Save file and it uploads the code for you!

# Adafruit

- A Minority and Woman-owned Business Enterprise
- “Adafruit was founded in 2005 by MIT engineer, Limor "Ladyada" Fried. Her goal was to create the best place online for learning electronics and making the best designed products for makers of all ages and skill levels.”

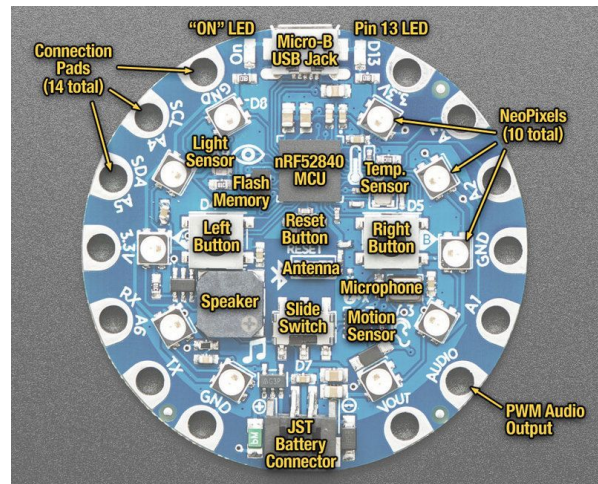


# CircuitPython w/ Circuit Playground Bluefruit



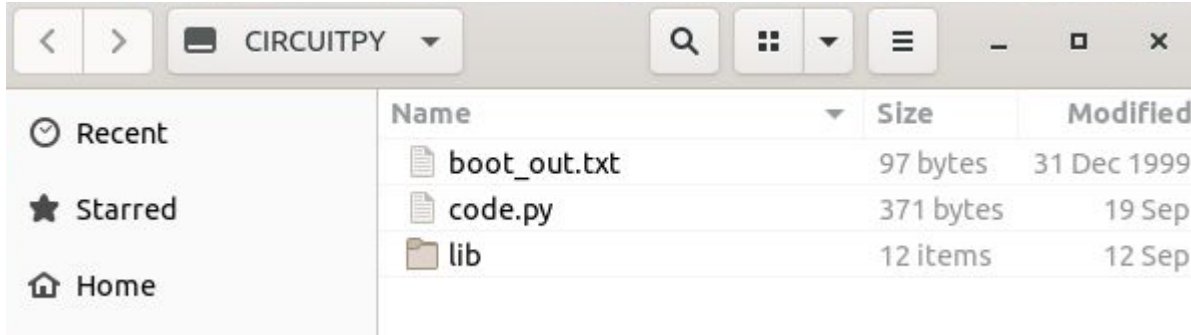
<https://cdn-shop.adafruit.com/970x728/4333-11.jpg>

- nRF52840 microcontroller
- Bluetooth Low Energy support for wireless connectivity
- 10 x mini NeoPixels, each one can display any color
- 1 x Motion sensor
- 1 x Temperature sensor (thermistor)
- 1 x Light sensor (phototransistor)
- 1 x Sound sensor (MEMS microphone)
- 1 x Mini speaker with class D amplifier
- 2 x Push buttons
- 1 x Slide switch
- And more!



<https://learn.adafruit.com/adafruit-circuit-playground-bluefruit/guided-tour>

# File Structure



A screenshot of a file explorer window titled "CIRCUITPY". The window has a sidebar on the left with navigation options: "Recent", "Starred", and "Home". The main area displays a table of files and folders. The table has three columns: "Name", "Size", and "Modified". The files listed are "boot\_out.txt" (97 bytes, 31 Dec 1999), "code.py" (371 bytes, 19 Sep), and a folder named "lib" (12 items, 12 Sep).

Name	Size	Modified
boot_out.txt	97 bytes	31 Dec 1999
code.py	371 bytes	19 Sep
lib	12 items	12 Sep

# GPIO - Output - LEDs blinking



```
import board
import digitalio
import time

led = digitalio.DigitalInOut(board.LED)
led.direction = digitalio.Direction.OUTPUT

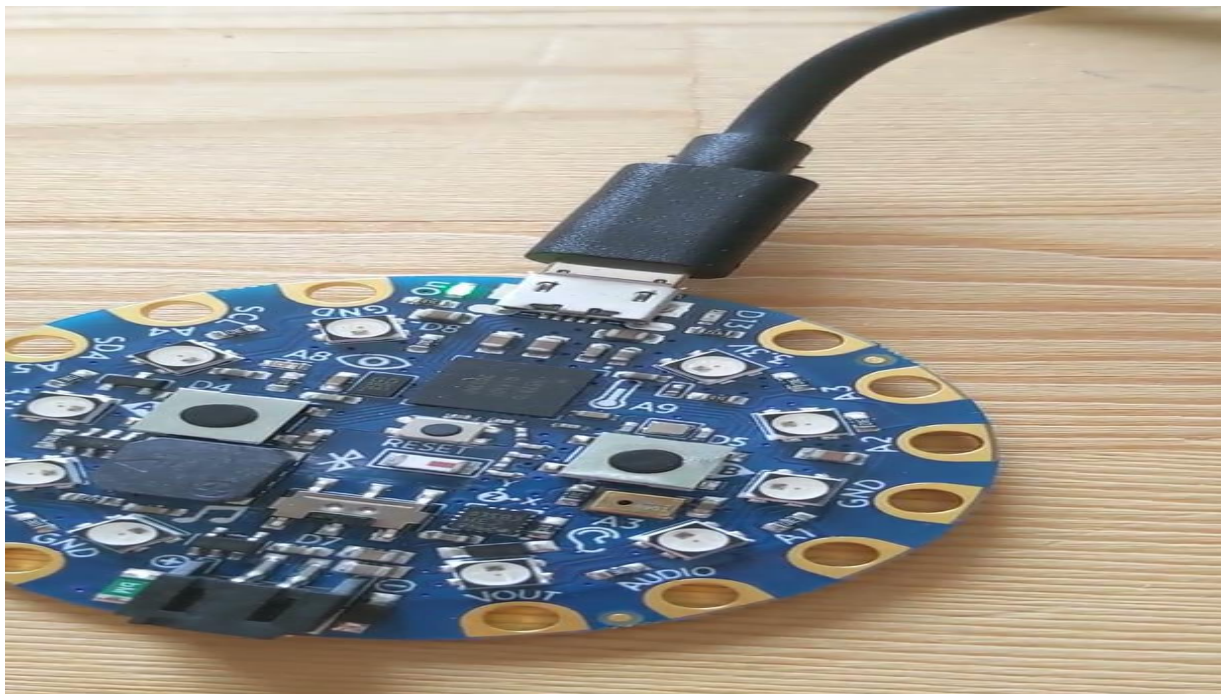
while True:
    led.value = True
    time.sleep(1)
    led.value = False
    time.sleep(1)
```

```
from adafruit_circuitplayground import cp
import time

while True:
    cp.red_led = True
    time.sleep(1)
    cp.red_led = False
    time.sleep(1)
```

# Heartbeat blinking

---





# Serial Console

- Can use print statements to help debug
- But where do the print statements go? There's no screen!
- On Linux, can you use something called "screen"

```
from adafruit_circuitplayground import cp
import time
```

```
while True:
    print("light on")
    cp.red_led = True
    time.sleep(1)
    print("light off")
    cp.red_led = False
    time.sleep(1)
```

Auto-reload is on. Simply save files over USB to run them or enter REPL to disable.

code.py output:

```
light on
light off
light on
light off
light on
light off
light on
light off
light on
light off
```

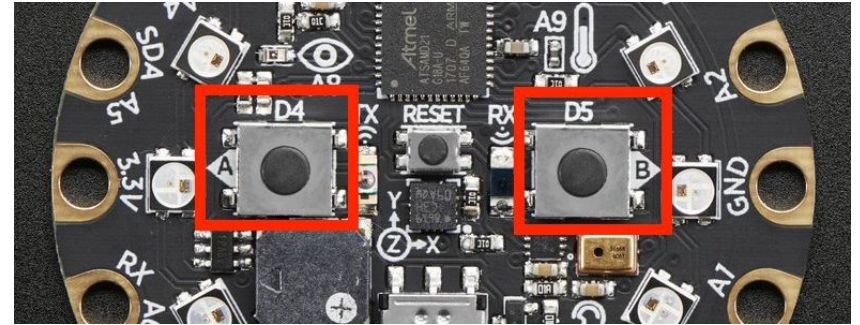
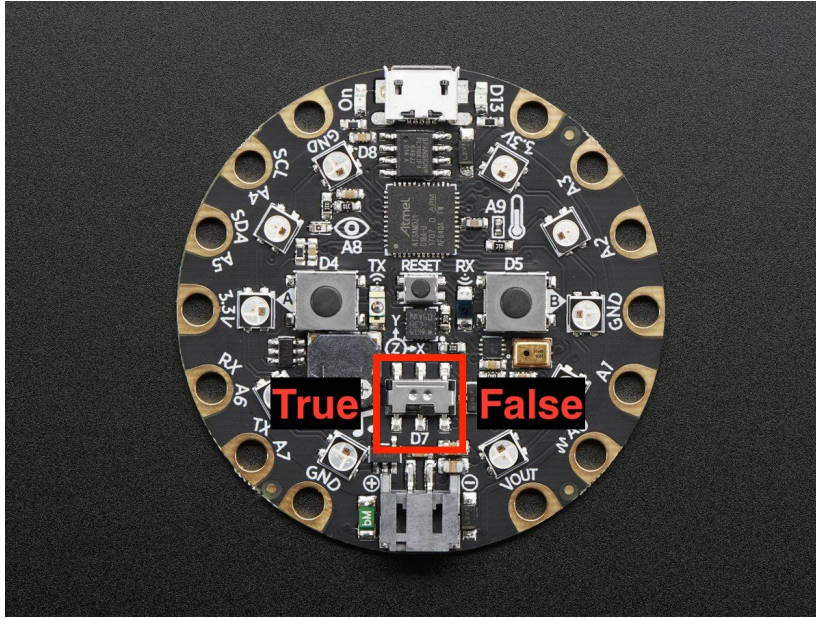
# REPL



Press any key to enter the REPL. Use CTRL-D to reload.

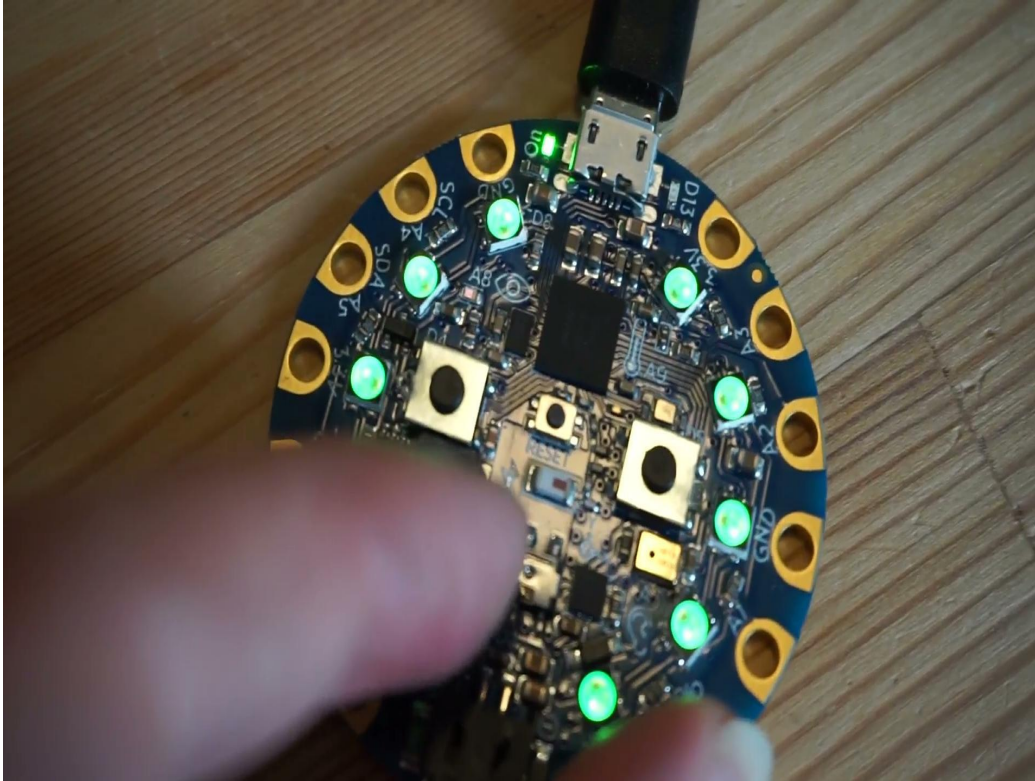
```
Adafruit CircuitPython 6.2.0 on 2021-04-05; Adafruit Circuit Playground Bluefruit with nRF52840
>>> import board
>>> import digitalio
>>> led = digitalio.DigitalInOut(board.LED)
>>> led.direction = digitalio.Direction.OUTPUT
>>> led.value = True
>>> █
```

# GPIO - Input - Buttons and switch



<https://learn.adafruit.com/circuitpython-made-easy-on-circuit-playground-express/buttons>

# Buttons and switch



```
from adafruit_circuitplayground import cp
```

```
cp.pixels.brightness = 0.01
```

```
green = (0, 255, 0)
```

```
pink = (255, 100, 150)
```

```
red = (255, 0, 0)
```

```
yellow = (255, 255, 0)
```

```
while True:
```

```
    if cp.switch:
```

```
        cp.pixels.fill(green)
```

```
    else:
```

```
        if cp.button_a:
```

```
            cp.pixels.fill(pink)
```

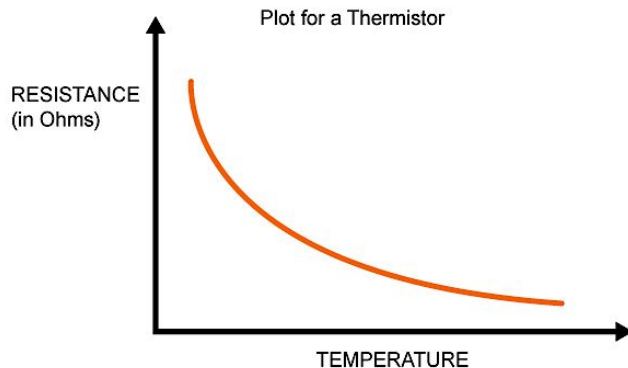
```
        elif cp.button_b:
```

```
            cp.pixels.fill(yellow)
```

```
        else:
```

```
            cp.pixels.fill(red)
```

# ADC - Temperature



```
from adafruit_circuitplayground import cp
import time

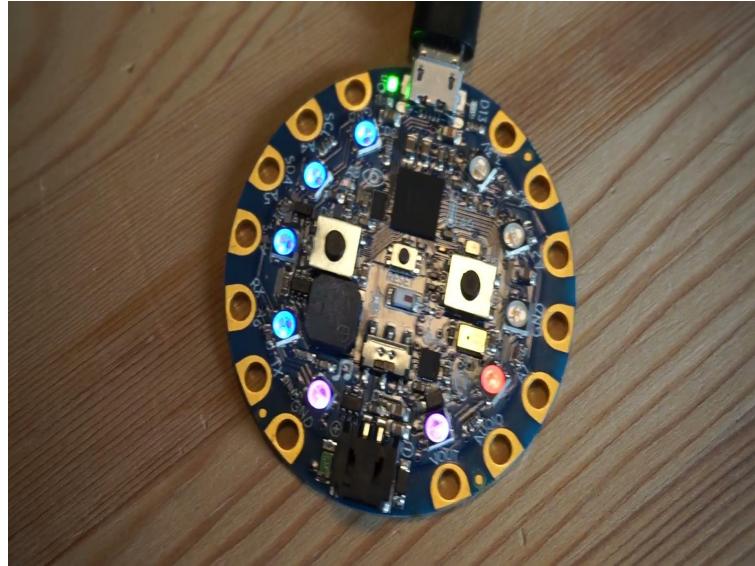
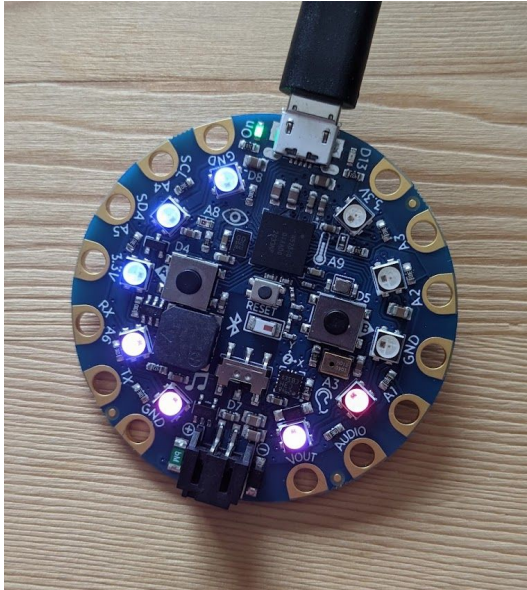
cp.pixels.brightness = 0.01
delay_time = 0.5
colors = [(0, 0, 200), (0, 0, 220), (0, 0, 255),
          (100, 0, 215), (200, 0, 150), (200, 0, 200),
          (215, 0, 40), (255, 0, 0), (255, 0, 0), (255, 0, 0)]

while True:
    temperature = cp.temperature
    for pixel in range(min(temperature / 4, 10)):
        cp.pixels[pixel] = colors[pixel]
    time.sleep(delay_time)
```



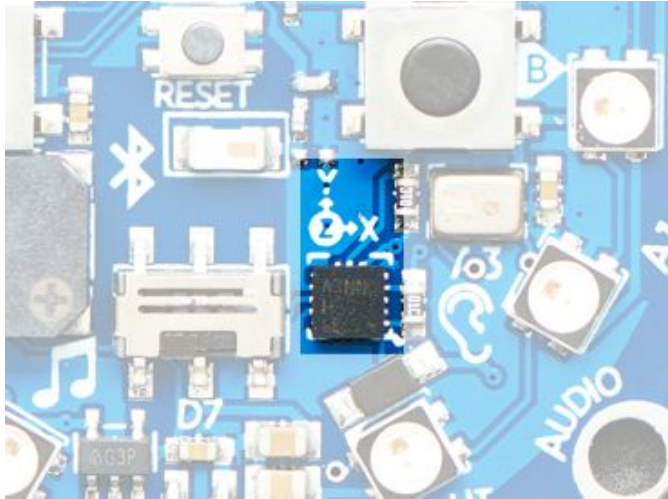
# Temperature w/ LEDs

---



# ADC - Accelerometer

Tippy Lights Example modified from  
<https://www.linkedin.com/learning/learning-circuitpython-with-circuit-playground-express/understanding-basic-circuits>



Can detect 3-axis acceleration

```
from adafruit_circuitplayground import cp
import time

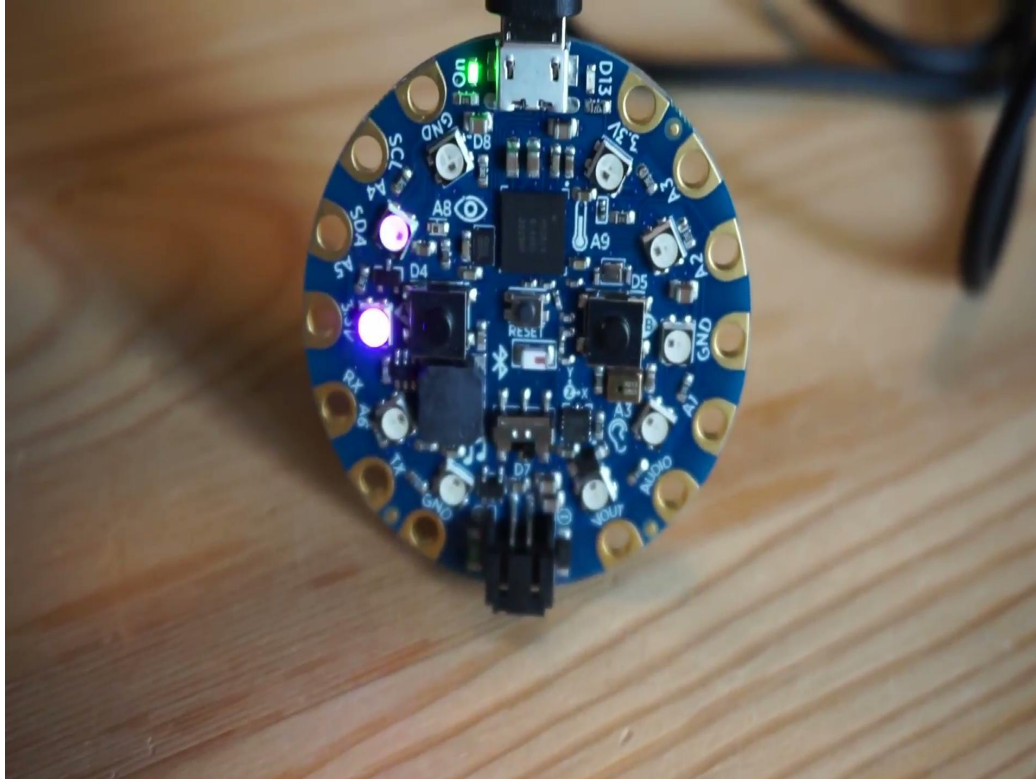
pink = (200, 0, 200)
clear = (0, 0, 0)
cp.pixels.brightness = 0.1
delay_time = 0.05

while True:
    for pixel in range(10):
        cp.pixels[pixel] = pink
        time.sleep(delay_time)

        cp.pixels[pixel] = clear
        delay_time = abs(cp.acceleration.y - 10)* 0.03
```

# Accelerometer - Tippy Lights

---





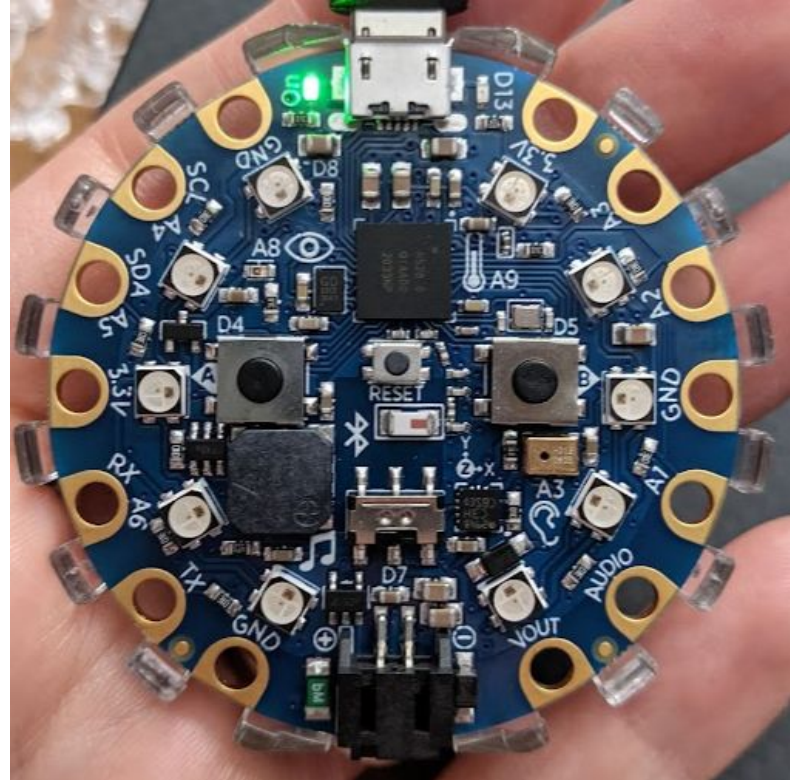
# Capacitive Touch and Sound

```
import time
from adafruit_circuitplayground import cp

scale = [523, 587, 659, 698, 784, 880, 988]
cp.adjust_touch_threshold(200)

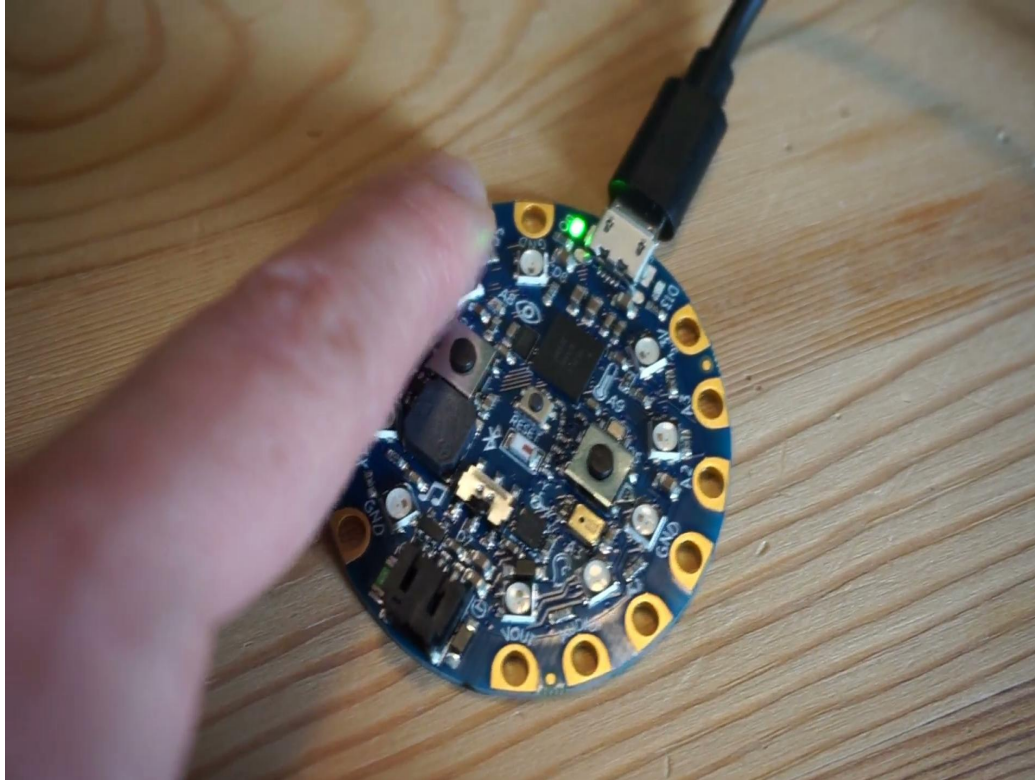
while True:
    if cp.touch_A1:
        cp.play_tone(scale[0], 0.25)
    if cp.touch_A2:
        cp.play_tone(scale[1], 0.25)
    if cp.touch_A3:
        cp.play_tone(scale[2], 0.25)
    if cp.touch_A4:
        cp.play_tone(scale[3], 0.25)
    if cp.touch_A5:
        cp.play_tone(scale[4], 0.25)
    if cp.touch_A6:
        cp.play_tone(scale[5], 0.25)
    if cp.touch_TX:
        cp.play_tone(scale[6], 0.25)

    time.sleep(0.05)
```

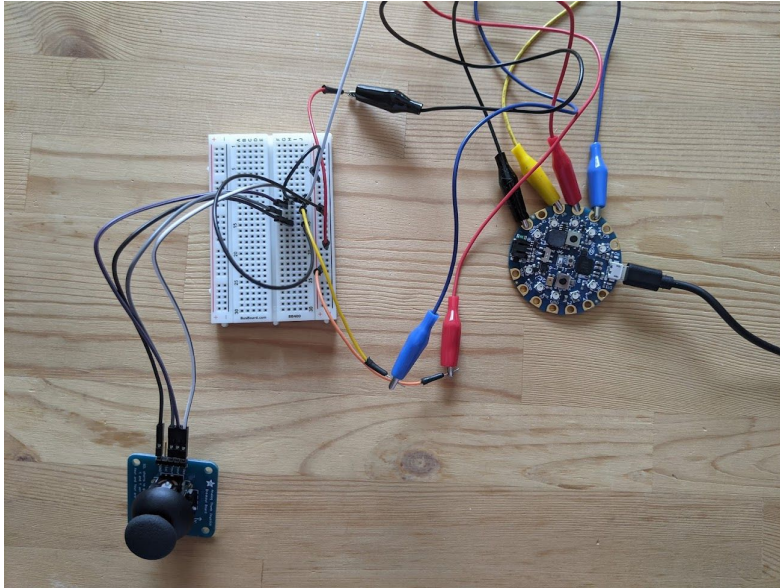


# Cap Touch and Play a Note

---



# Adding external components - potentiometer (ADC)



```
from adafruit_circuitplayground import cp
import time
import board
import analogio

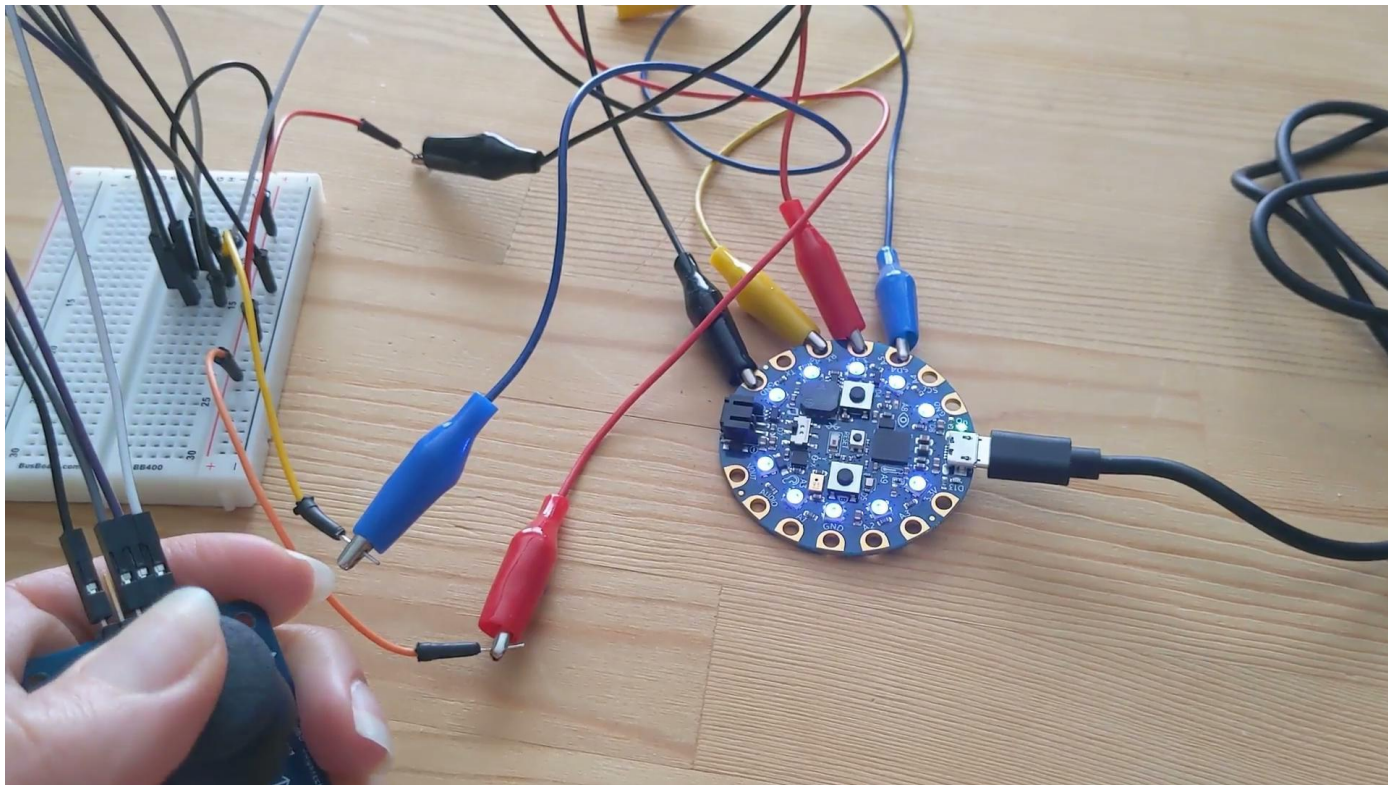
cp.pixels.brightness = 0.05
cp.pixels.auto_write = False

knob = analogio.AnalogIn(board.A6)

while True:
    cp.pixels.fill((255, 0, 0))
    pot_value = round(knob.value/65535 * 100)
    cp.pixels.brightness = pot_value/100
    cp.pixels.show()
    time.sleep(0.1)
```



# Joystick LED Dimmer and Color Changer



```
from adafruit_circuitplayground import cp
import time
import board
import analogio

cp.pixels.brightness = 0.05
cp.pixels.auto_write = False

pink = (250, 50, 174)
blue = (0, 0, 255)
green = (0, 255, 0)

knob_x = analogio.AnalogIn(board.A6)
knob_y = analogio.AnalogIn(board.A5)

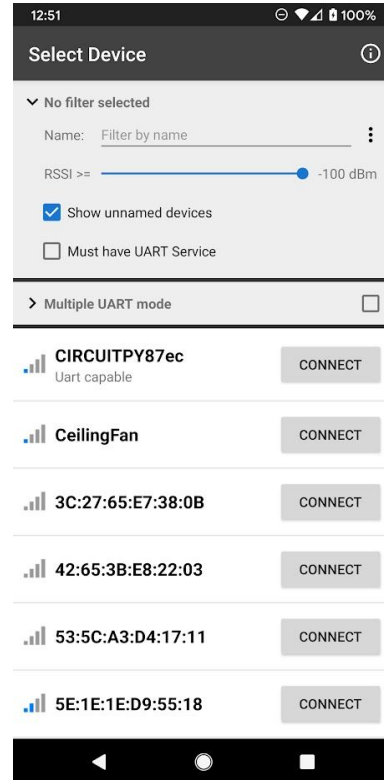
while True:
    pot_value_x = round(knob_x.value/65535 * 100)
    pot_value_y = round(knob_y.value/65535 * 100)

    brightness = (pot_value_x + pot_value_y)/2000
    cp.pixels.brightness = brightness

    if brightness < 0.03:
        cp.pixels.fill(blue)
    elif brightness < 0.05:
        cp.pixels.fill(green)
    else:
        cp.pixels.fill(pink)

    cp.pixels.show()
    time.sleep(0.1)
```

# Bluetooth with the Bluefruit



Add picture of options

<https://learn.adafruit.com/bluefruit-le-connect>

# UART - Acceleration



```
from adafruit_ble import BLERadio
from adafruit_ble.advertising.standard import ProvideServicesAdvertisement
from adafruit_ble.services.nordic import UARTService
from adafruit_circuitplayground import cp
import time

ble = BLERadio()
uart = UARTService()
advertisement = ProvideServicesAdvertisement(uart)

ble.start_advertising(advertisement)
print("Waiting to connect")

while True:
    while not ble.connected:
        pass

    result = (round(cp.acceleration.x, 2), round(cp.acceleration.y, 2), round(cp.acceleration.z, 2))

    if result:
        try:
            uart.write(str(result).encode("utf-8"))
        except Exception as e:
            print(repr(e))

    time.sleep(0.5)
```

# External Libraries

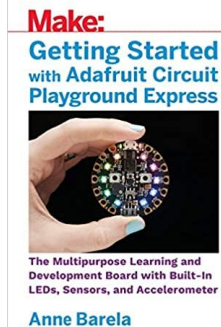
---

- External libraries available for various microcontrollers and peripherals
- Can download all of the latest library files in a bundle here:  
<https://circuitpython.org/libraries>
- Tool available to keep your libraries up to date on your board:  
<https://pypi.org/project/circup/>
- Due to memory constraints, you may not be able to include everything on your board. Only include what you need.

# More Resources

---

- <https://learn.adafruit.com/>
- <https://learn.adafruit.com/category/circuit-playground>
- <https://circuitpython.readthedocs.io>
- <https://www.linkedin.com/learning/learning-circuitpython-with-circuit-playground-express/understanding-basic-circuits>
- <https://www.amazon.com/Getting-Started-Adafruit-Circuit-Playground/dp/1680454889>
- <https://makecode.adafruit.com/>





# Thanks!

## Any questions?

Resources located at:

<https://github.com/kirakirakira/hardware-circuitpython>

Contact me at [kira.hartlage@gmail.com](mailto:kira.hartlage@gmail.com)

# Credits

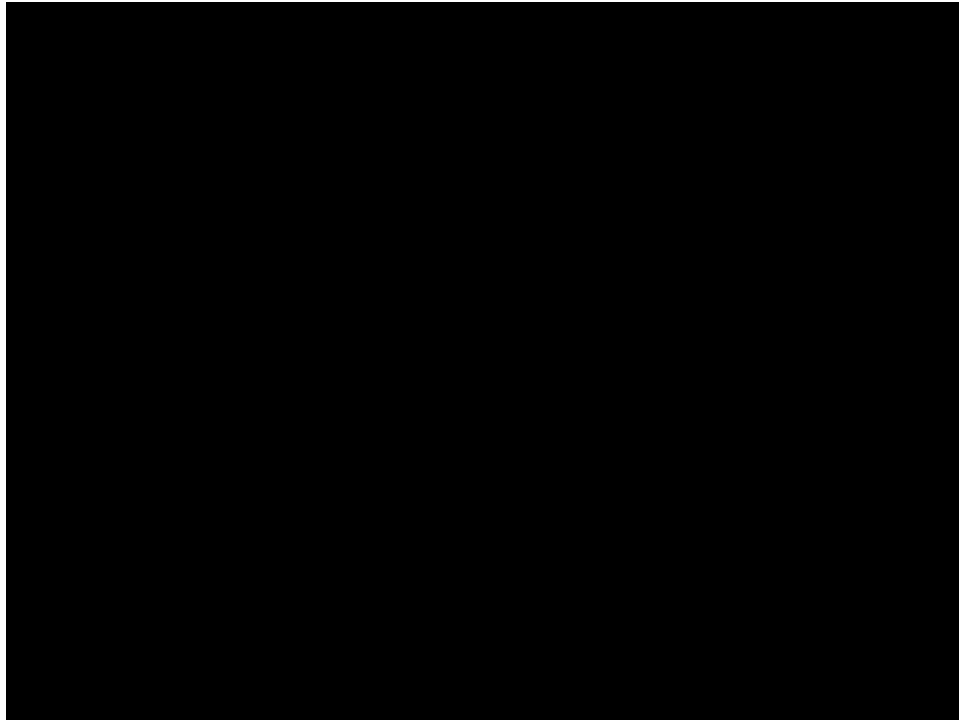
---

Special thanks to all the people who made and released these awesome resources for free:

- Presentation template by SlidesCarnival
- Photographs by Unsplash

# Light Dimmer Video #2

---



# Control from the app?

---