CircuitPython/Micropython – How to get started

Agenda:

- What is CircuitPython/MicroPython?
- Why would I want to use CircuitPython/MicroPython?
- Development setup, configuration, and tools
- Jump into to lots of live examples

What is CircuitPython/MicroPython?

- A complete reimplementation of Python 3 for microcontrollers
- Has all the features of regular Python (sans standard library) and can run in as little as 8k of RAM
- CircuitPython (Adafruit) builds on MicroPython’s rock-solid foundation with education/beginner friendly libraries and APIs

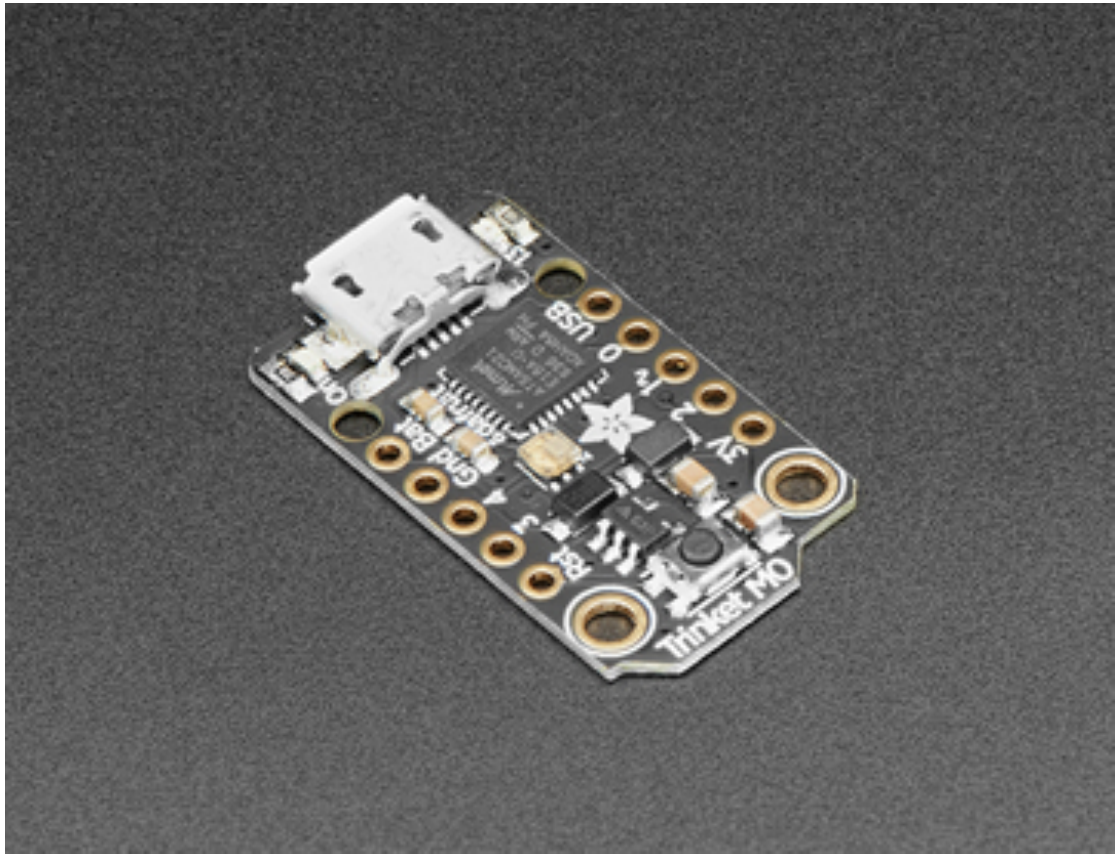
Why use CircuitPython/MicroPython?

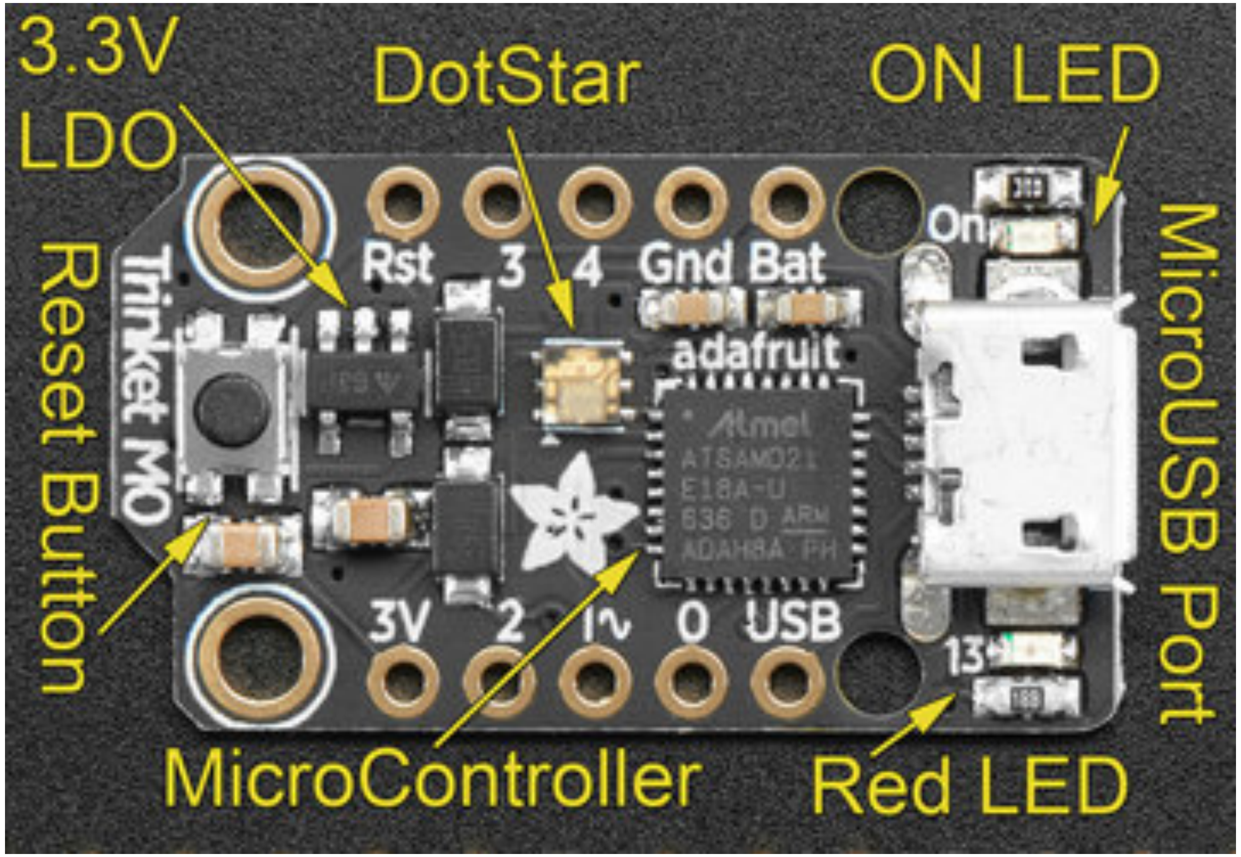
- New to programming
- Want to get up and running quickly
- Easily update you code in a live environment
- Serial Console and REPL
- Great Hardware support

Development setup, configuration, and tools

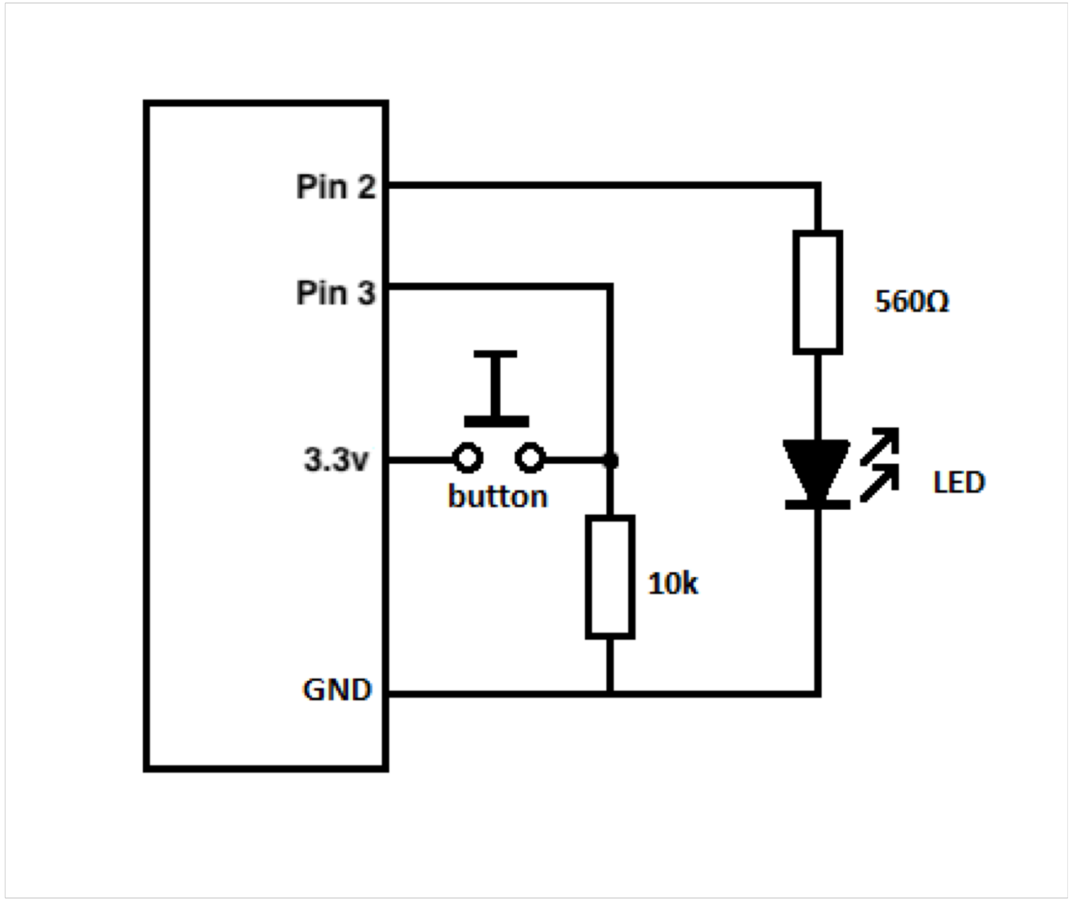
- Boards used in examples (Trinket M0, Circuit Playground Express)
- Installing CircuitPython
- Serial interface to boards
- REPL
- Mu Editor

Trinket M0




3.3V LDO
DotStar
ON LED
Reset Button
MicroUSB Port
MicroController
Red LED
Rst
3
4
Gnd
Bat
adafruit
Trinket M0
3V
2
1~
0
USB
On
13

Let’s look at this circuit with Serial Interface and REPL



Coding Conventions

```
# any library import
import board

#setup code
x = 12

# Run loop
    while True:
```

File naming

You can use any of the following:

- code.txt
- code.py
- main.txt
- main.py

Circuit Python looks for code to run, and looks for files named as above. It looks for them in that order.

The preferred naming is: code.py

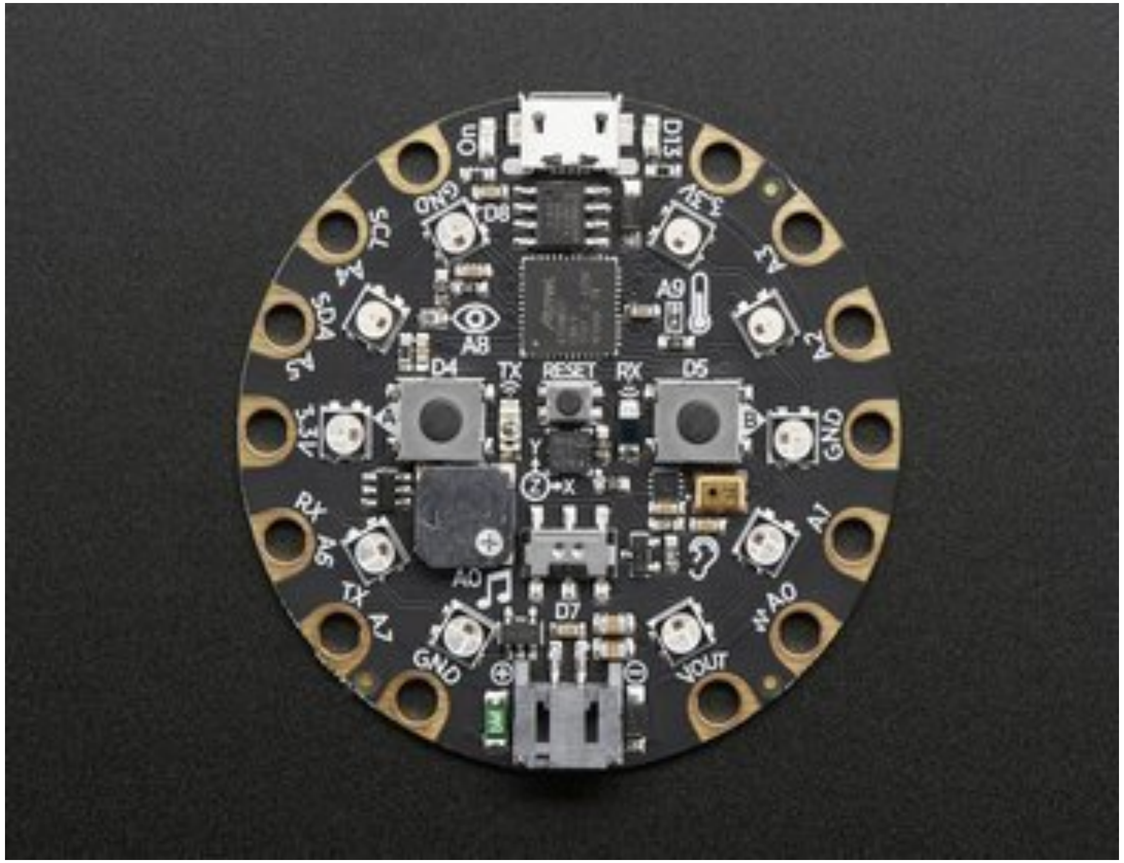
Hello World – Blink

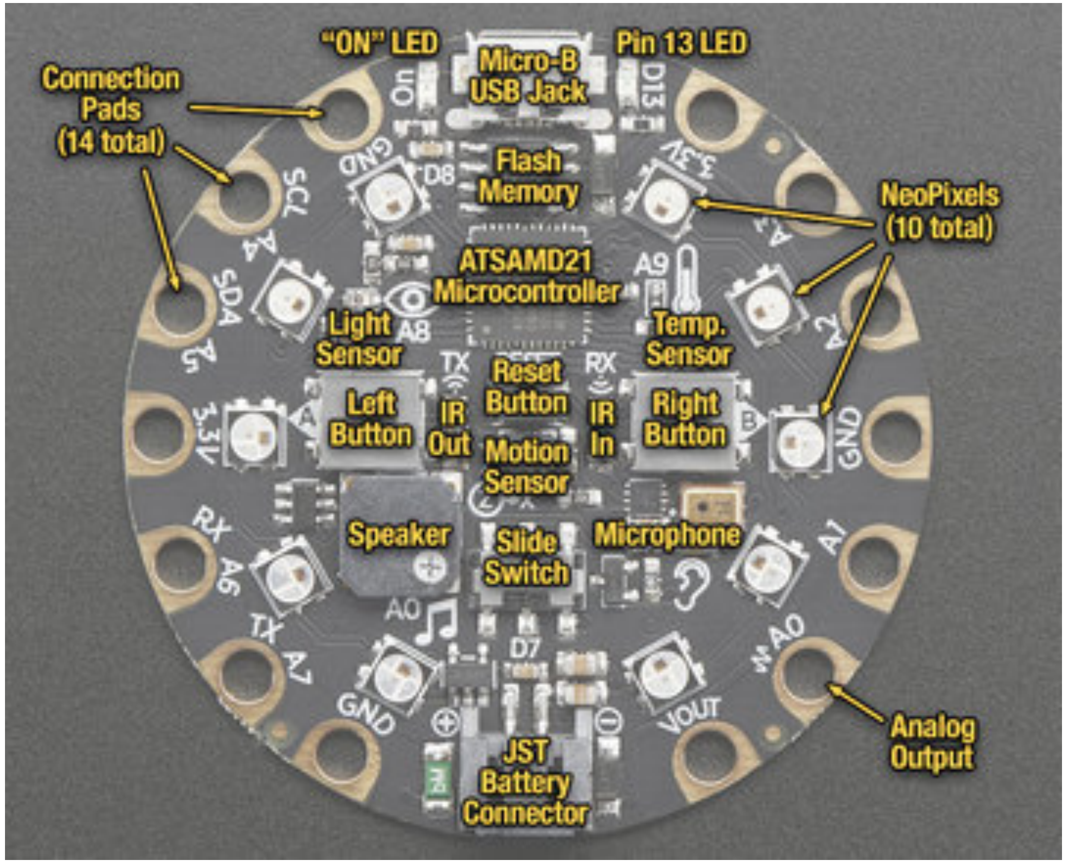
```
import board
import digitalio
import time

led = digitalio.DigitalInOut(board.D13)
led.direction = digitalio.Direction.OUTPUT

while True:
    led.value = True
    time.sleep(0.5)
    led.value = False
    time.sleep(0.5)
```

Circuit Playground Express





Circuit Playground Express Features

- 10 x mini NeoPixels, each one can display any color
- 1 x Motion sensor (LIS3DH triple-axis accelerometer with tap detection, free-fall detection)
- 1 x Temperature sensor (thermistor)
- 1 x Light sensor (phototransistor). Can also act as a color sensor and pulse sensor.
- 1 x Sound sensor (MEMS microphone)
- 1 x Mini speaker with class D amplifier (7.5mm magnetic speaker/buzzer)
- 2 x Push buttons, labeled A and B
- 1 x Slide switch
- Infrared receiver and transmitter - can receive and transmit any remote control codes, as well as send messages between Circuit Playground Expresses. Can also act as a proximity sensor.
- 8 x alligator-clip friendly input/output pins
- Includes I2C, UART, 8 pins that can do analog inputs, multiple PWM output
- 7 pads can act as capacitive touch inputs and the 1 remaining is a true analog output
- Green "ON" LED so you know its powered
- Red "#13" LED for basic blinking
- Reset button
- ATSAMD21 ARM Cortex M0 Processor, running at 3.3V and 48MHz
- 2 MB of SPI Flash storage, used primarily with CircuitPython to store code and libraries.
- MicroUSB port for programming and debugging
- USB port can act like serial port, keyboard, mouse, joystick or MIDI!

Blink with button

```
import time
import board
import digitalio

led = digitalio.DigitalInOut(board.D13)
led.direction = digitalio.Direction.OUTPUT

button = digitalio.DigitalInOut(board.BUTTON_A)
button.direction = digitalio.Direction.INPUT
button.pull = digitalio.Pull.DOWN

while True:
    if button.value:  # button is pushed
        led.value = True
    else:
        led.value = False

    time.sleep(0.01)
```

Super Combination for Quick Prototyping

Circuit Python

+

Circuit Playground Express

+

Circuit Playground Express Library

Just import this library…

```
from adafruit_circuitplayground.express import cpx
```

01 Blink

```
import time
from adafruit_circuitplayground.express import cpx

while True:
    cpx.red_led = True
    time.sleep(0.5)
    cpx.red_led = False
    time.sleep(0.5)
```

02 Button A

```
from adafruit_circuitplayground.express import cpx

while True:
    if cpx.button_a:
        cpx.red_led = True
```

03 Button B Toggle

```
from adafruit_circuitplayground.express import cpx

while True:
    if cpx.button_b:
        cpx.red_led = True
    else:
        cpx.red_led = False
```

04 Neopixel

```
from adafruit_circuitplayground.express import cpx

while True:
    cpx.pixels[1] = (0, 50, 0)
```

05 Neopixel fill

```
from adafruit_circuitplayground.express import cpx

cpx.pixels.brightness = 0.3

while True:
    cpx.pixels.fill((255, 0, 0))
```

06 Buttons and Neopixels

```
from adafruit_circuitplayground.express import cpx

cpx.pixels.brightness = 0.3

while True:
    if cpx.button_a:
        cpx.pixels[0:5] = [(255, 0, 0)] * 5
    elif cpx.button_b:
        cpx.pixels[5:10] = [(0, 255, 0)] * 5
    else:
        cpx.pixels.fill((0, 0, 0))
```

07 Slide Switch

```
from adafruit_circuitplayground.express import cpx

while True:
    # Left returns True. Right returns False.
    cpx.red_led = cpx.switch
```

08 Capacitive Touch

```
import time
from adafruit_circuitplayground.express import cpx

while True:
    if cpx.touch_A1:
        print("Touched A1!")
    time.sleep(0.1)
```

09 Capacitive Touch - all

```
import time
from adafruit_circuitplayground.express import cpx

while True:
    if cpx.touch_A1:
        print("Touched A1!")
    if cpx.touch_A2:
        print("Touched A2!")
    if cpx.touch_A3:
        print("Touched A3!")
    if cpx.touch_A4:
        print("Touched A4!")
    if cpx.touch_A5:
        print("Touched A5!")
    if cpx.touch_A6:
        print("Touched A6!")
    if cpx.touch_A7:
        print("Touched A7!")
    time.sleep(0.1)
```

10 Light Sensor (Look at plotter)

```
import time
from adafruit_circuitplayground.express import cpx

while True:
    print("Light level:", cpx.light)
    print((cpx.light,))
    time.sleep(1)
```

11 Accelerometer

```
import time
from adafruit_circuitplayground.express import cpx

while True:
    x, y, z = cpx.acceleration
    print((x, y, z))
    time.sleep(0.5)
```

12 Shake (use of accelerometer)

```
from adafruit_circuitplayground.express import cpx

while True:
    if cpx.shake():
        print("Shake detected!")
        cpx.red_led = True
    else:
        cpx.red_led = False
```

13 Tap detection

```
from adafruit_circuitplayground.express import cpx

cpx.detect_taps = 2

while True:
    if cpx.tapped:
        print("Tap detected!")
```

14 Tap and Neopixel

```
from adafruit_circuitplayground.express import cpx

cpx.detect_taps = 2
pixel_number = 0

while True:
    if cpx.tapped:
        print("Tap detected!")
        cpx.pixels.fill((0, 0, 0))
        cpx.pixels[pixel_number] = (0, 0, 50)
        pixel_number += 1
        if pixel_number >= 10:
            pixel_number = 0
```

15 Play tone

```
from adafruit_circuitplayground.express import cpx

cpx.play_tone(262, 1)
cpx.play_tone(294, 1)
```

16 Start Tone – Stop Tone

```
from adafruit_circuitplayground.express import cpx

while True:
    if cpx.button_a:
        cpx.start_tone(262)
    elif cpx.button_b:
        cpx.start_tone(294)
    else:
        cpx.stop_tone()
```

17 Play wave file

```
from adafruit_circuitplayground.express import cpx

while True:
    if cpx.button_a:
        cpx.play_file("coin.wav")
    elif cpx.button_b:
        cpx.play_file("eep.wav")
```

Questions?

Thank you!

Find Me:
dan@kacenjar.com
Twitter @kacenjar
https://github.com/kacenjar

Circuit Python Resources:
https://learn.adafruit.com/welcome-to-circuitpython
https://learn.adafruit.com/welcome-to-circuitpython/circuitpython-essentials
https://learn.adafruit.com/adafruit-circuit-playground-express
https://learn.adafruit.com/circuitpython-made-easy-on-circuit-playground-express
https://circuitpython.readthedocs.io