Maker Space Session

Bluefruit Intro & Pre-Requisites

Session Goals:

MCU Fundamentals
Install Pre-Requisites
Make a blinky thing

Assumptions:

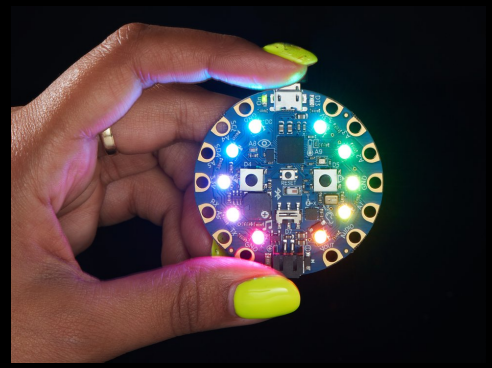
You have software development experience.

You may not have much hardware or microcontroller experience.

Maker Space Session

Fundamentals
& Intro
To Microcontrollers

Introduction



What is this thing?

Demonstration

A very full featured device.

A great intro to micro controllers and programming.

Key Features

Blinky Lights (Neopixels)

Jiggle Detection (LIS3DH)

Light, Sound, Temp Sensors

Beepy Thing (Speaker)

Pushy Things (button, Switches)

Shiny Metal (GPIO /Interface Pins)

What is a Micro Controller (MCU)?

A compact circuit designed for a specific single purpose, in an embedded system.

Similarities to a Computer

Has a CPU, Ram, and I/O*

* but a lot less (Bluefruit has 2MB of Flash storage, and 32K ram- enough for a few hundred lines program. Flash can also store libraries and data structures referenced from code.)

MCU vs. Computer

Doesn’t run an OS

Typically, only has basic interfaces (I2C, SPI, UART)

Lower Power Requirements

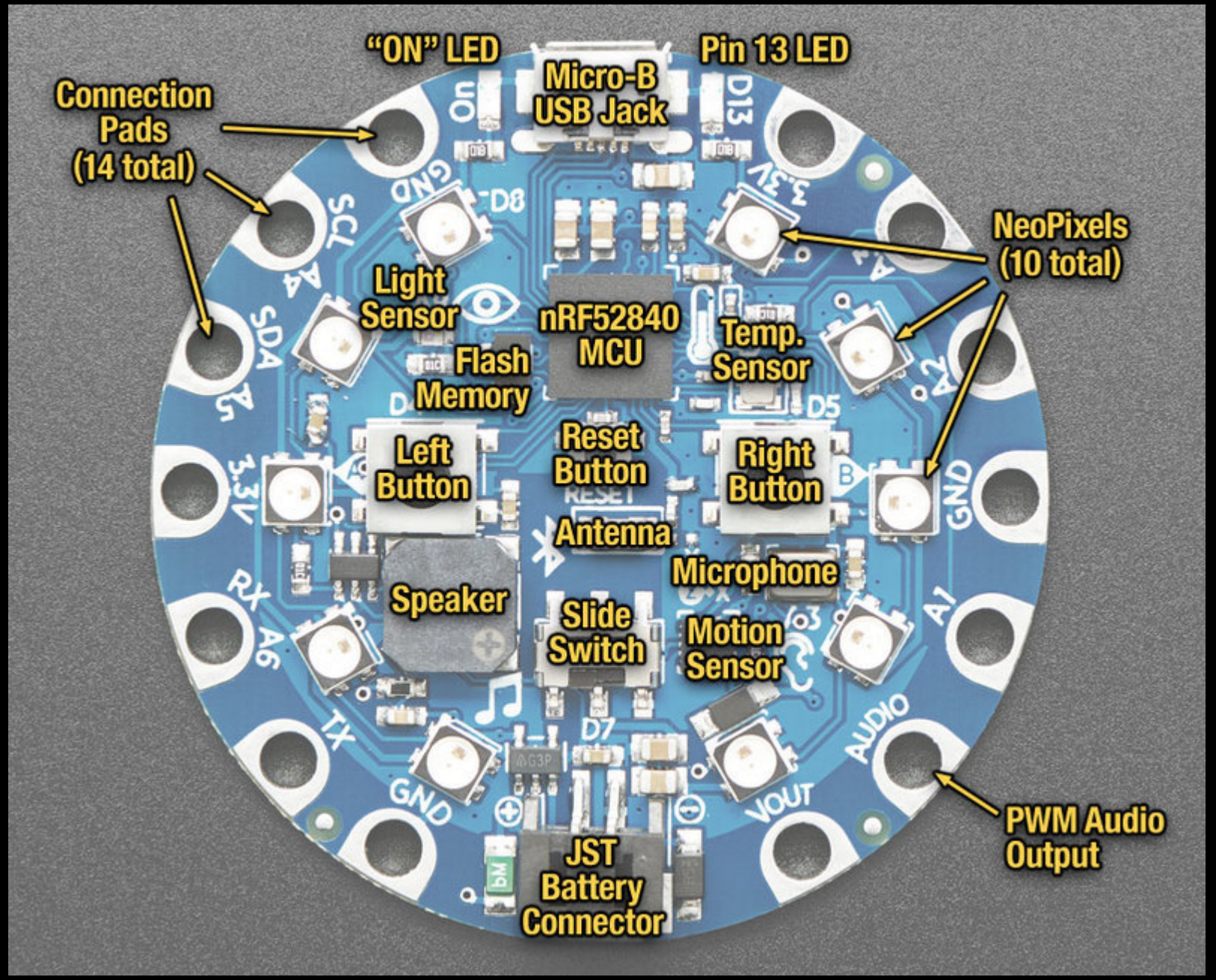
MCU's Commonly Used:

Robotics

3d Printing, Drones,

Automobiles, Cell Phones,

IoT Devices


"ON" LED
Micro-B USB Jack
Pin 13 LED
Connection Pads (14 total)
NeoPixels (10 total)
Light Sensor
nRF52840 MCU
Temp. Sensor
Flash Memory
Reset Button
Left Button
Right Button
Antenna
Microphone
Speaker
Slide Switch
Motion Sensor
JST Battery Connector
PWM Audio Output

How do you Program it?

Circuit Python or

Arduino/C++

Circuit Python:

Higher Level Language (like Java or .NET)

Arduino:

Lower Level Language (C++)

Circuit Python:

"Easier" to code

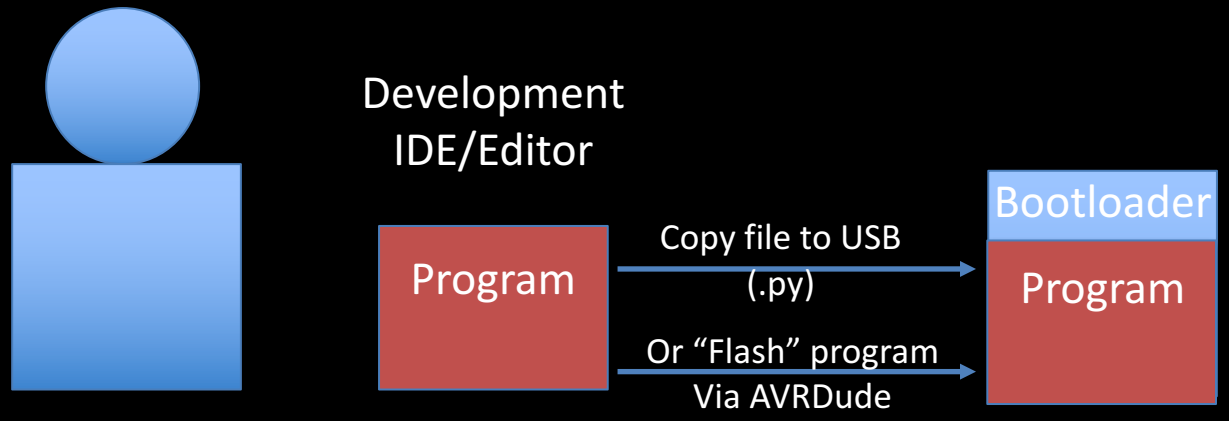
White Space Language

Arduino/C++

Lower Level, more control,

Best performance, *more familiar

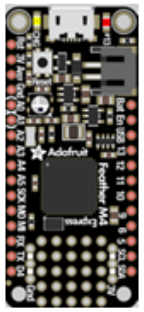
What is the Development Process?



The CircuitPython Boot Sequence

Bootloader Mode

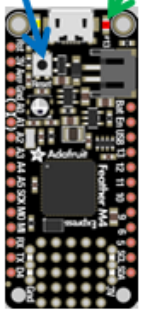
Board is powered on and bootloader code starts



The bootloader turns on the Red LED

Reset

LED solid red



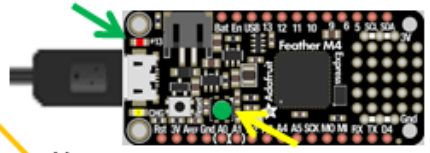
Wait 500 milliseconds to see if RESET pushed

RESET pushed

RESET not pushed: go to User Code

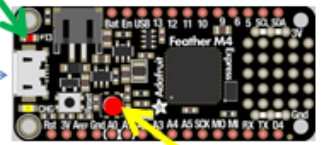
USB Connected ?

Yes



LED Pulses slowly, RGB LED Green

No



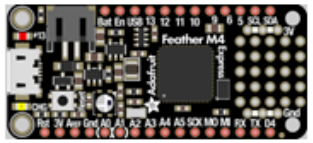
LED Pulses quickly, RGB LED red

User Code Mode

ARDUINO

Code Type ?

circuit python



Code starts immediately

Bootloader waits an additional 700ms for a RESET to Safe Mode

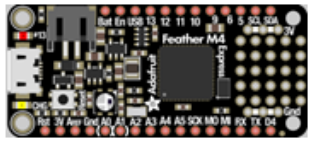
Reset

Red LED blinks 3 times

RGB LED is Yellow

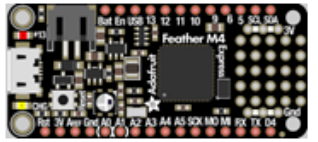
RESET pushed ?

Yes



Safe Mode: board is a USB drive, code.py and boot.py are not run

No



Run user code: Boot.py runs then code.py

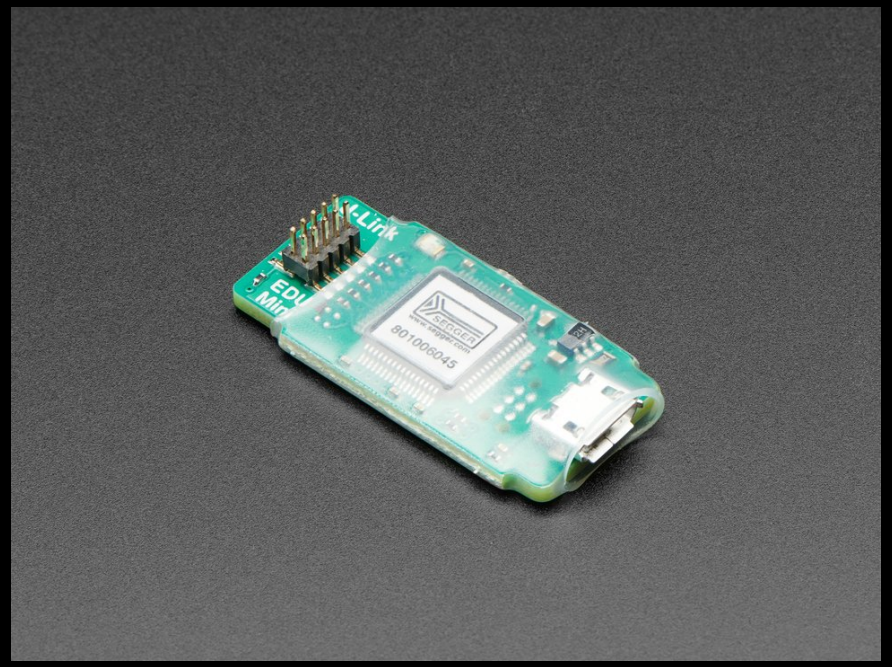
Version 1.00 @Adafruit

What about Debugging?

Serial.Print(“Here!”);

It’s a bit more challenging on Micro Controllers w/o specialized hardware.

Circuit Python recommended Debugger (for SWD or JDAG Interface): J-Link



($20 – Non Commercial)

($400 – Commercial)

Circuit Python: the REPL

The REPL

Save Subscribe

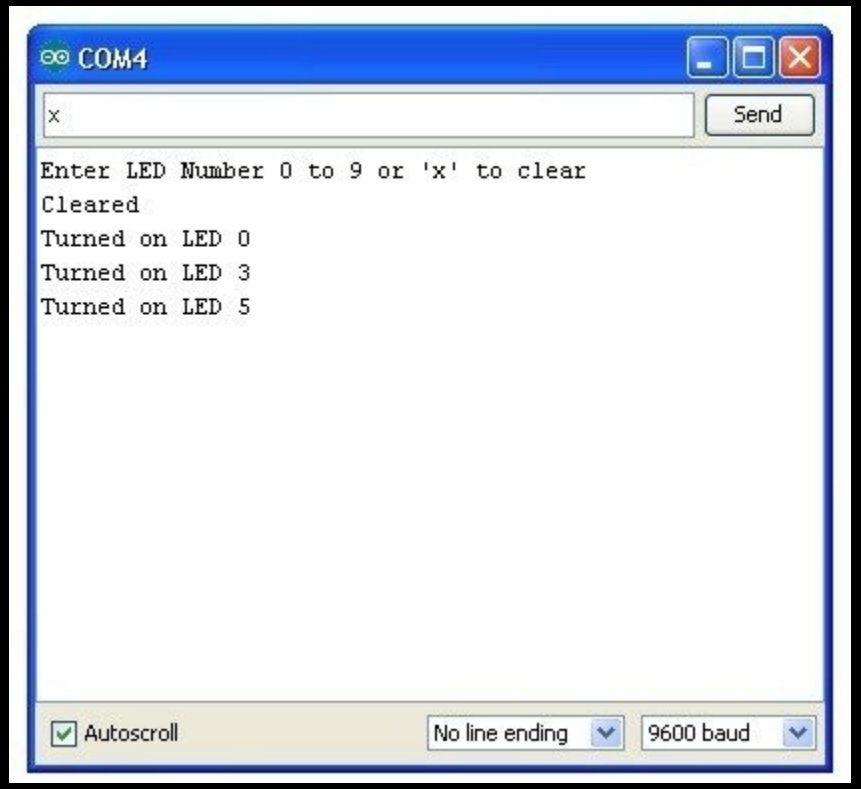
The other feature of the serial connection is the **R**ead-**E**valuate-**P**rint-**L**oop, or REPL. The REPL allows you to enter individual lines of code and have them run immediately. It's really handy if you're running into trouble with a particular program and can't figure out why. It's interactive so it's great for testing new ideas.

Entering the REPL

To use the REPL, you first need to be connected to the serial console. Once that connection has been established, you'll want to press **CTRL+C**.

If there is code running, in this case code measuring distance, it will stop and you'll see `Press any key to enter the REPL. Use CTRL-D to reload.` Follow those instructions, and press any key on your keyboard.

Arduino: Serial Monitor



** There is a relatively new “Arduino Extension for VS Code” Available (In Preview), allows more robust debugging of Arduino without specialized hardware.

Prepping BlueFruit & Installing Mu

Bluefruit Device Pre-Requisites

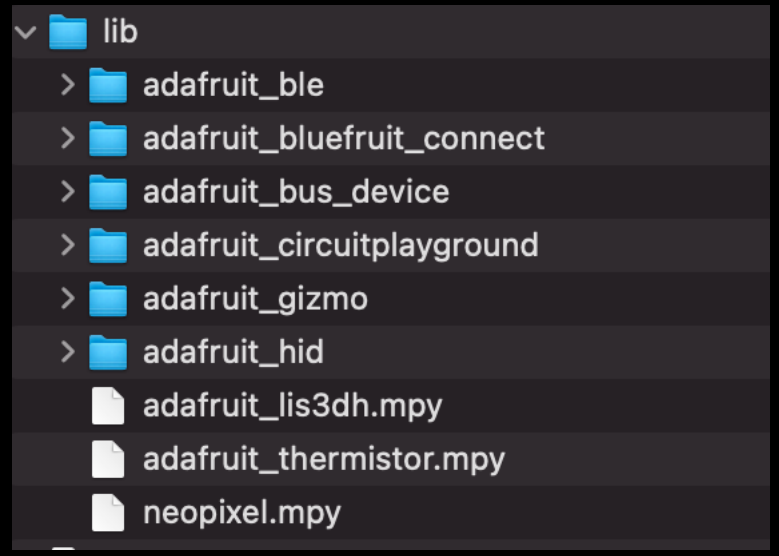
- Install latest stable release of CircuitPython Runtime
- https://circuitpython.org/board/circuitplayground_bluefruit/

- Steps:
  - Connect Bluefruit with a USB cable you know works with Data
  - Double click middle button of Bluefruit to enter DFU mode (Device Firmware Update mode)
  - Browse to the USB drive (CPLAYBTBOOT)
  - Copy the .UF2 file you just downloaded to the CPLAYBTBOOT root directory (Wait for light sequence to change)
  - After updating, your drive will be renamed to CIRCUITPY and you will have a few files in there (code.py, which is a "Hello World" sketch).

Circuit Python Libraries (V7)

- Optional- Windows Only (Driver): https://github.com/adafruit/Adafruit_Windows_Drivers/releases/tag/2.5.0.0

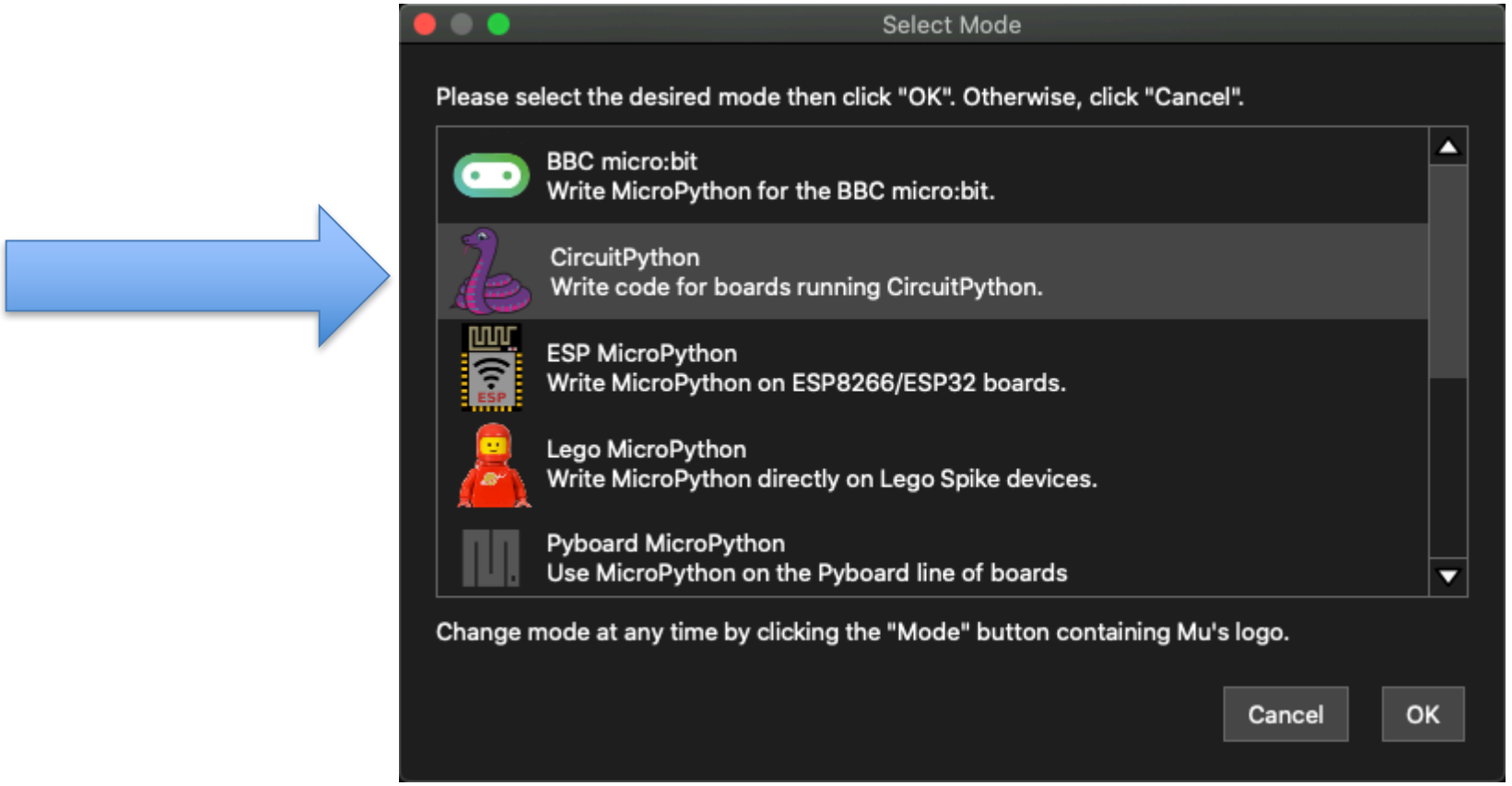
- Download Circuit Python Libraries (v7): https://circuitpython.org/libraries
- Copy over these to a folder called “lib” on your Bluefruit:



Download & Install Mu

https://codewith.mu

Open Mu, and select “CircuitPython”



Creating your first program

code.py

```
1  import time
2  from adafruit_circuitplayground import cp
3  cp.pixels.brightness = 0.3
4
5  while True:
6      cp.pixels[0] = (255, 0, 0)
7      time.sleep(0.5)
8      cp.pixels[0] = (0,0,0)
9      time.sleep(0.5)
10
```

Save as “code.py” into root folder of CIRCUITPY USB Drive and watch the magic happen.

Got it working?

Exercise 2: Modify the code to make **all** of the neopixels blink.
Bonus points for efficient code (12 lines or less).

(Difficulty Level: Easy)

Exercise #2

code.py

```
1  import time
2  from adafruit_circuitplayground import cp
3  cp.pixels.brightness = 0.3
4
5  while True:
6      for x in range(0, 10, 1):
7          cp.pixels[x] = (255,0,0)
8      time.sleep(0.5)
9
10     for x in range(0, 10, 1):
11         cp.pixels[x] = (0,0,0)
12     time.sleep(0.5)
```

Save as “code.py” into root folder of CIRCUITPY USB Drive and watch the magic happen.

Got it working?

Exercise 3: Use the switch to toggle the LED's from blinking on and off (hint: check the status of cp.switch, which is a boolean)

(Difficulty Level: Super Easy)

Exercise #3

code.py

```
1  import time
2  from adafruit_circuitplayground import cp
3  cp.pixels.brightness = 0.3
4
5  while True:
6      if cp.switch == True:
7          for x in range(0, 10, 1):
8              cp.pixels[x] = (255,0,0)
9          time.sleep(0.5)
10
11         for x in range(0, 10, 1):
12             cp.pixels[x] = (0,0,0)
13         time.sleep(0.5)
14
15
```

Save as “code.py” into root folder of CIRCUITPY USB Drive and watch the magic happen.

Got it working?

Exercise 4: Press the a button to cycle colors forward (red, yellow, green, cyan, purple, white) for all Neopixels, and the b button to cycle back to the previous set of color.

Hint: Buttons are referenced by cp.button_a and cp.button_b and have a boolean value)

Hint: Utilize an Array to define and iterate colors

Hint: Don’t forget to use REPL for helpful debugging!

(Difficulty Level: Medium)

code.py

```
import time
from adafruit_circuitplayground import cp
cp.pixels.brightness = 0.3

colors = [[255,0,0],   # red
          [255,150,0], #yellow
          [0,255,0],   #green
          [0,255,255], #cyan
          [0,0,255],   #blue
          [180,0,255], # purple
          [255,255,255]] #white

numcolors = 7
x = 1

while True:
    for y in range(0, 10, 1):
        cp.pixels[y] = colors[x-1]

    if cp.button_a is True:
        print(x)
        if x < numcolors:
            x = x+ 1
        time.sleep(0.5)


    if cp.button_b is True:
        print(x)
        if x > 1:
            x = x - 1
        time.sleep(0.5)

```

Exercise #4

Save as “code.py” into root folder of CIRCUITPY USB Drive and watch the magic happen.

So now what?

- Like the Hardware? Purchase a BlueFruit Ticket on Eventbrite (supplies limited) to buy a kit ($38). Otherwise, please return hardware to a volunteer.
- Want to do more? Hang out in the Maker Space, and visit the self paced examples on the Adafruit Web Site.
- **https://learn.adafruit.com/adafruit-circuit-playground-bluefruit/circuitpython-neopixel**

Bluefruit Challenge

- Got some spare time? Work in the Maker Space to make something awesome. Consider incorporating 3d printing and other electronics into your design.
- Provide a demonstration by 1pm Friday in the Maker Space.
- Awards will be given for the following (original designs only):
  - Most Impressive (individual)
  - Most Creative (individual)
  - Most Collaborative / Best Teamwork (up to 3 people)

Advanced Projects

- Get a buddy and play Quick Draw- with 2 Bluefruits! https://learn.adafruit.com/circuit-playground-quick-draw/circuitpython

- Pick a color with an app on your phone, and change the color of the Bluefruit (Arduino): https://learn.adafruit.com/adafruit-circuit-playground-bluefruit/ble-uart-controller