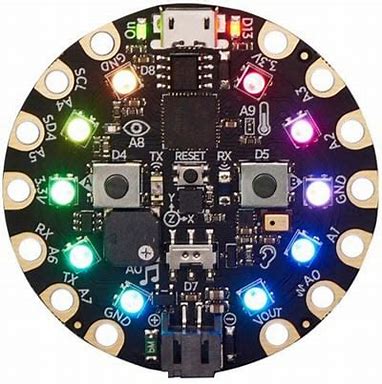
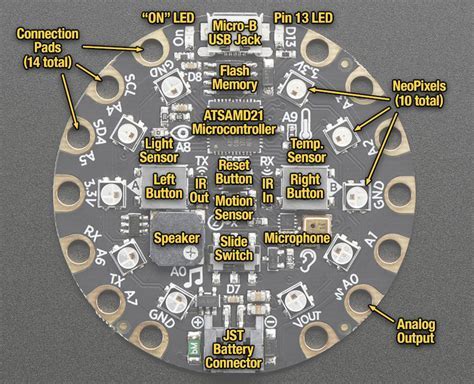
Embedded Programming with Microcontrollers

*Exploring the* ***Circuit Playground Express***

# Overview



The Circuit Playground Express is an example of microcontroller based electronics. It contains a number of input/output contacts, neopixels, and sensors that can be programmed with various programming environments: *Microsoft Makecode*, *Javascript*, *Python*, and the *Arduino* development environment.

# CircuitPython

***CircuitPython*** is the implementation of *Python* created by *Adafruit* for several products, including the *Circuit Playground Express*. It is a fork/derivative of *MicroPython*, a version of *Python* designed to run on microcontrollers. ***CircuitPython*** excels at the following:

* **Very fast development:** Write code, save the file, and it runs immediately. No compiling is required, since Python is an interpreted language – the hardware can interpret and act on each command you type, practically simultaneously.
* **REPL –** Read Evaluate Print Loop: You can start interactive programming with the REPL
* **Easy code changes:** Since your code lives on the flash drive, you can edit it whenever you like, and you can keep multiple files around for easy experimentation.
* **File Storage:** *CircuitPython’s* data storage ability makes it great for data logging, playing audio clips, and otherwise interacting with files.

# Using the Mu Editor

Install the *Mu* Editor from here: <https://codewith.mu/>

1. **Before starting *Mu***, please **plugin your Circuit Playground Express** and make sure it shows up as a **CIRCUITPY** drive in your computer’s file explorer/finder. When you are finished, you will eject this drive safely, just as you would any flash drive.
2. **Start *Mu*,** and select the **Adafruit *CircuitPython* mode.**
3. **Managing files:** The *CircuitPython* file you copy over to run on the Circuit Playground Express should always be called **code.py**. This is the file the device is looking to run. When you are finished creating, testing, and running this file, you should **create another copy** of this file **on your own machine** with a more appropriate filename; **e.g. blinkingLED.py**

# Creating our First Program

Our **first program** is made of the following code:

**import** time

**from** adafruit\_circuitplayground.express **import** cpx

**while** **True**:

cpx.red\_led = **True**

time.sleep(1)

cpx.red\_led = **False**

time.sleep(1)

## Running our First Program

Copy a file named **code.py** onto the **CIRCUITPY** drive. You can use the **Save** button to save the file directly to the Circuit Playground Express if you named the program **code.py**. If not, **double-click** the **filename tab**, and a dialog box will appear allowing you to name and save the code wherever you wish.

The program should automatically start blinking the **Pin 13 LED** *(beside the Micro USB jack)* repeatedly in **1 second intervals**. If the **LED** is not blinking, press the **Reset** button.

## Serial Output

**Insert the line:** print(“Hello CircuitPython!”) into the while loop. Copy the new version of the **code.py** file to the Circuit Playground Express and the **Serial Window** will open in the ***Mu* editor**. The **Serial window** acts a both a **command input/output** window and as an interactive **REPL** Python environment.

# Next Steps

## Using the NeoPixels

The ***NeoPixel*** LEDs are numbered **counterclockwise** from **cpx.pixels[0]** through **cpx.pixels[9]**.

All pixels can be lit to the same colour value specified by the **RGB colour** scheme. For example:

cpx.pixels.fill( 30, 0, 0)

will set all pixels to a red colour. The full ON value is 255 and the full OFF value is 0. Setting the (R, G, B) values in different ways produces different colours. The following program sets the ***NeoPixel*** LEDs to a series of random colours, every 5 seconds.

**import** time

**import** random

**from** adafruit\_circuitplayground.express **import** cpx

cpx.pixels.brightness = 0.1

**while** **True**:

**for** i **in** range(10):

r = random.randint(0,255)

g = random.randint(0,255)

b = random.randint(0,255)

cpx.pixels[i] = (r,g,b)

time.sleep(5)

## Reading Sensors

The following program shows how to get things started when it comes to reading sensors.

**import** time

**from** adafruit\_circuitplayground.express **import** cpx

**while** **True**:

**print**(“Temperature: “, cpx.temperature, “, Light Intensity: “, cpx.light)

time.sleep(1)

## Mu Plotting

***Mu*** has a built-in plotting capability that allows you to visualize what a sensor is doing as data is being collected. Outputting a **Tuple** using the *print()* will provide a real-time plot of each item in the **Tuple** relative to time. The **Plotter** icon must be clicked in order to see the plotted values. The following program shows how to plot the light intensity values read by the light sensor.

**import** time

**from** adafruit\_circuitplayground.express **import** cpx

**while** **True**:

intensity = cpx.light

**print**( (intensity, ) )

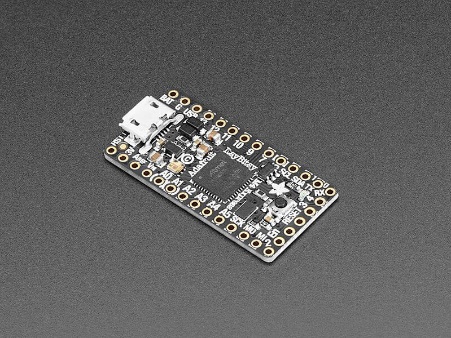
time.sleep(0.1)

## Challenges

1. Create a program that: turns all NeoPixels green, waits 5 seconds, turns all NeoPixels yellow, waits two seconds, then turns all NeoPixels red for 5 seconds. This sequence should repeat indefinitely.
2. Create a program that rotates a single red-light counter-clockwise from through NeoPixels 0 to 9. This sequence should repeat indefinitely.
3. Create a program that rotates a single red-light counter-clockwise from through NeoPixels 0 to 9 and then clock-wise through NeoPixels 9 to 0. This sequence should repeat indefinitely.

*Exploring the* ***ItsyBitsy M4 Express***

# Overview



The ***ItsyBitsy M4 Express*** is a small microcontroller board that can be programming using either the *Arduino IDE* or ***CircuitPython***. With ***CircuitPython*** you can use the ***Mu*** editor, if you’d like, or any text editor of your choosing. You actually don’t need to install any IDE onto your computer. Just edit the main.py file on the board *(it shows up like a flash drive)*, save it, and your *Python* code will run. ***Note:*** Loose headers come with the board and must be soldered on to be used with a breadboard.

## Examples Using a Breadboard:

### Built-in Blinking LED

**import** board

**import** digitalio

**import** time

led = digitalio.DigitalInOut(board.D13)

led.direction = digitalio.Direction.OUTPUT

**while** **True**:

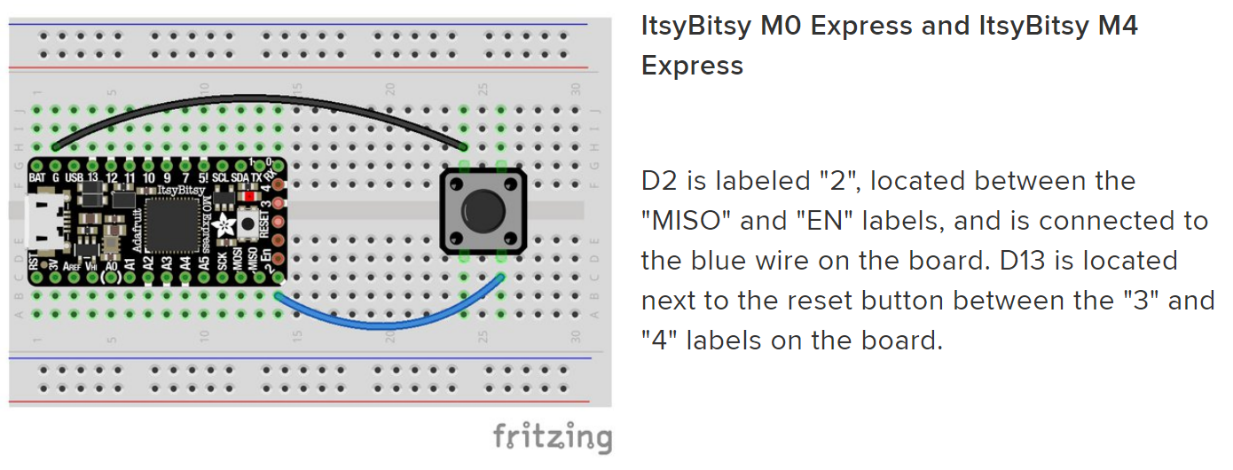
led.value = **True**

time.sleep(0.5)

led.value = **False**

time.sleep(0.5)

### Digital Input/Output



**import** time

**import** board

**from** digitalio **import** DigitalInOut, Direction, Pull

led = DigitalInOut(board.D13)

led.direction = Direction.OUTPUT

switch = DigitalInOut(board.D2)

switch.direction = Direction.INPUT

switch.pull = Pull.UP

**while** **True**:

**if** switch.value:

led.value = **False**

**else**:

led.value = **True**

time.sleep(0.01) **#** debounce delay

### Pulse Width Modulation

**import** time

**import** board

**import** pulseio

piezo = pulseio.PWMOut(board.A1, duty\_cycle=0, frequency=440, variable\_frequency=True)

**while** **True**:

**for** f **in** (262, 294, 330, 349, 392, 440, 494, 523):

piezo.frequency = f

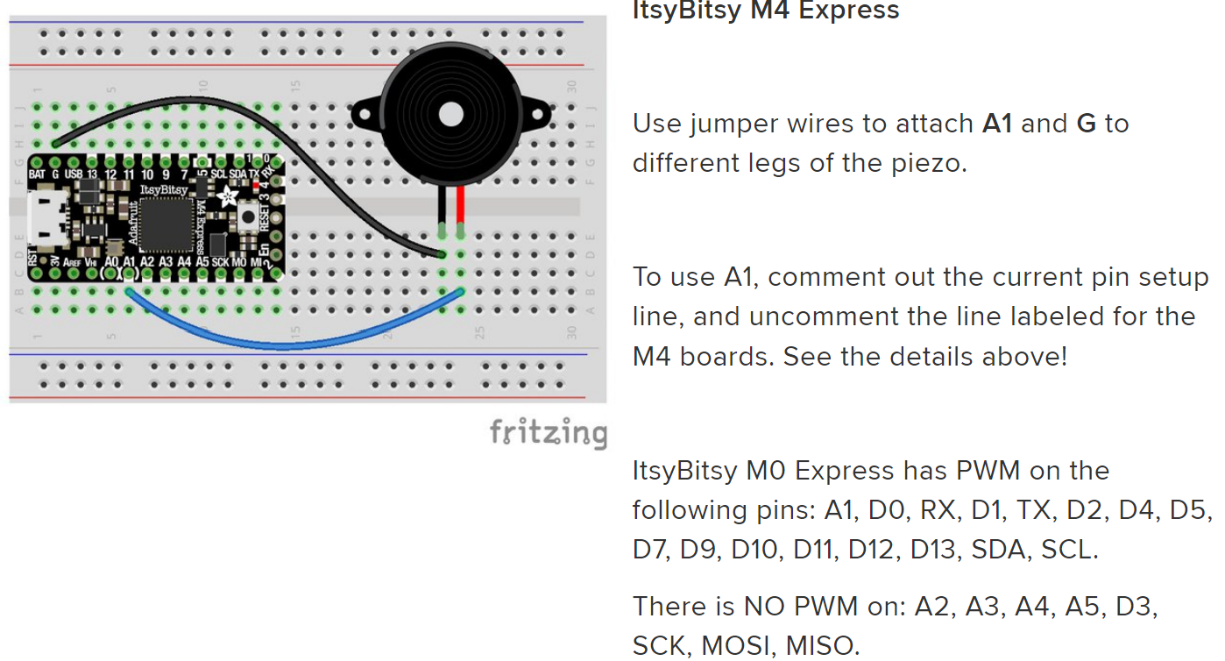
piezo.duty\_cycle = 65536 // 2 **#** On 50%

time.sleep(0.25) **#** On for 1/4 second

piezo.duty\_cycle = 0 **#** Off

time.sleep(0.05) **#** Pause between notes

time.sleep(0.5)



## Challenge

1. Given: a red LED, a 220 Ohm resistor, and 2 jumper wires. Create a circuit and program that: blinks the red LED once every second. **Note:** The circuit can be created between the ground **(G)** pin and pin **7**. Remember: The long lead on the red LED is the positive one.

## Resources

What is Circuit Python? - <https://learn.adafruit.com/welcome-to-circuitpython/what-is-circuitpython>

Circuit Python Libraries - <https://learn.adafruit.com/welcome-to-circuitpython/circuitpython-libraries>

Circuit Python Essentials - <https://learn.adafruit.com/welcome-to-circuitpython/circuitpython-essentials>

Circuit Python - <https://circuitpython.org/>

Circuit Playground Express Library - <https://circuitpython.readthedocs.io/projects/circuitplayground/en/latest/api.html>

Introducing Adafruit ItsyBitsy M4 Express - <https://learn.adafruit.com/introducing-adafruit-itsybitsy-m4>

## Where to Buy?

Mouser Canada - <https://www.mouser.ca/>

Elmwood Electronics - <https://elmwoodelectronics.ca/>