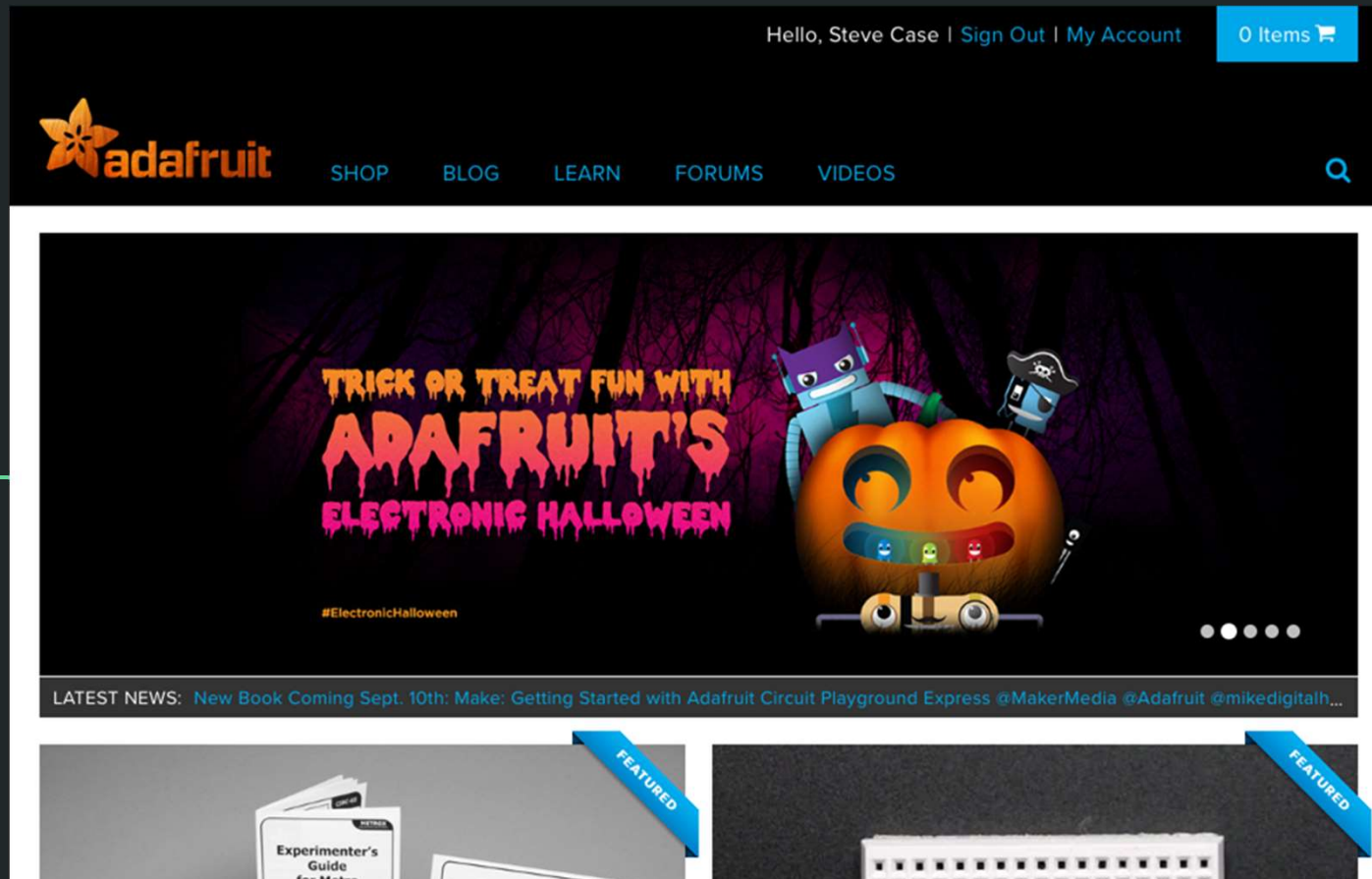


Gemma, Python and Wearables

Steve Case & Sana Sarfraz

Adafruit Industries



Limor Fried - Lady Ada




Adafruit Industries

← → ↺ <https://learn.adafruit.com/category/wearables> ☆ 🔊 📄 ⋮

Last-Minute Halloween Accoutrements with HalloWing


Quickly scare up a costume or accessory with items found in Adabox 009
by [Phillip Burgess](#)



Oh noes! An 11th-hour invite to a Halloween function and you don't have a costume ready! Or maybe you want something discreet but fun when the work dress code doesn't permit a full werewolf getup. HalloWing to the rescue! Our spooky little development board, plus a few extra tidbits from Adabox 009 (or your parts collection) can quickly

Glowing Mirror Mask


Halloween Masquerade Magic with NeoPixels
by [Erin St Blaine](#)



Create a holographic glowing mask with NeoPixels and a Hallowing. We've included three designs you can create with a vinyl cutter or cut by hand. Be the scariest skeleton on the street, the fiercest diving dragon, or the fanciest fairy at the faire.

Board

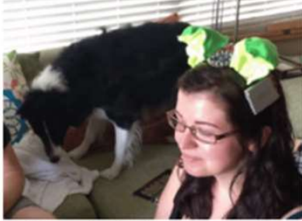
The not-so-mystifying oracle
by [Phillip Burgess](#)



Move around an invisible spirit board that can be seen only through the Hallowing's display...or touch one of the capacitive pads to have the oracle spell out a message on its own.

Express Perk-up Ears


Servo actuated ears that perk up when there's a loud noise.
by [Dave Astels](#)



Build a pair of animated ears that use servos and a Circuit Playground Express to perk up when there's a loud noise. Perfect for around friends, communing with your pet, or to dress up your next cosplay

PiGlass

Build a Raspberry Pi based wearable computer
by [M. Desmarais](#)

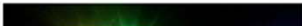


Cast a 3d Printed Necklace in Metal

Create a pewter pendant from a 3d printed design
by [Erin St Blaine](#)

NeoPixel Manicure

Light up your jazz hands with tiny NeoPixel LEDs
by [Sophy Wong](#)



Wearables Projects



Make a wearable cosplay unicorn horn that glows rainbow. When you touch the pretty copper decoration swirl, the horn will glow white at the tip. Purify your environment and create magic and giggles!

Adafruit Gemma M0

The Gemma M0 will supercharge your wearables! Small, light, and it's easy to use, so you can do more.

by [lady ada](#)



The Adafruit Gemma M0 may look small and cute: round, about the size of a quarter, with friendly alligator-clip sew pads. But do not be fooled! The Gemma M0 is incredibly powerful! We've taken the same form factor we used for the original ATtiny85-based

CircuitPython

by [John Park](#)



These LED rings add a nice touch to the tick-toc clockwork of your costume goggles. Perfect for Halloween and cosplay.

ISS Pin

You'll be star gazing when this NeoPixel pin lights up to notify you of the International Space Station flying overhead.

by [Leslie Birch](#)



Turn your own custom designs into beautiful pins with a milling machine, paint, and epoxy resin.

NeoPixie Dust Bag with Circuit Playground Express

Make a fancy, color changing Pixie Dust Bag with Circuit Playground Express!

by [John Park](#)



Build an electronic newt eye. Keep it in a jar to scare your halloween visitors, or wear it around your neck to complete your #ElectronicHalloween costume.

Glowing Viking Rune wayFinder

Create ancient magic with modern science


by [Erin St Blaine](#)




3d printing and laser-cut acrylic combine with neopixels and Arduino to create a movie-worthy Viking rune artifact. Wear it as a

Gemma M0 Development Board

iPad 8:32 AM 68%

SHOP BLOG LEARN FORUMS VIDEOS



Adafruit Gemma M0

The Gemma M0 will super-charge your wearables! Small, light, and it's easy to use, so you can do more.

Overview

[Guided Tour](#)

[Pinouts](#)

[Windows Driver Installation](#)

[What is CircuitPython?](#)

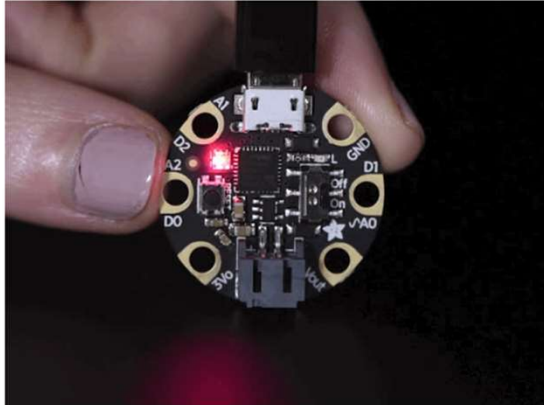
- ▶ [CircuitPython](#)
- ▶ [CircuitPython Essentials](#)
- ▶ [Arduino IDE Setup](#)

ARDUINO COMPATIBLES / ADAFRUIT GEMMA WEARABLES

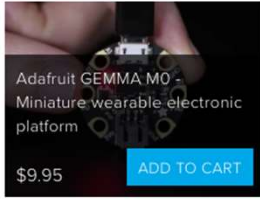
MICROCONTROLLERS / CIRCUITPYTHON

Overview

by lady ada

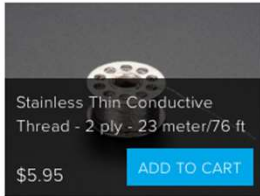


The Gemma M0 is a super small microcontroller board, with just enough to build many simple projects. It may look small and cute: round, about the size of a quarter, with friendly alligator-clip sew pads. But do not be fooled! The Gemma M0 is incredibly powerful! We've taken the same form factor we used for [the original ATtiny85-based Gemma](#) and gave it a power up. The Gemma M0 has swapped out the lightweight ATtiny85 for a [ATSAMD21E18](#) powerhouse.



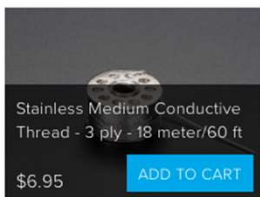
Adafruit GEMMA M0 - Miniature wearable electronic platform

\$9.95 [ADD TO CART](#)




Stainless Thin Conductive Thread - 2 ply - 23 meter/76 ft

\$5.95 [ADD TO CART](#)



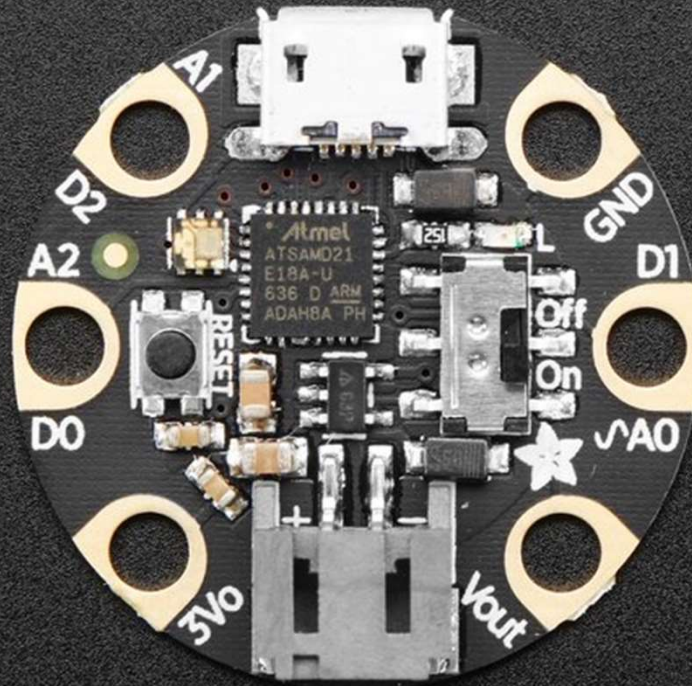
Stainless Medium Conductive Thread - 3 ply - 18 meter/60 ft

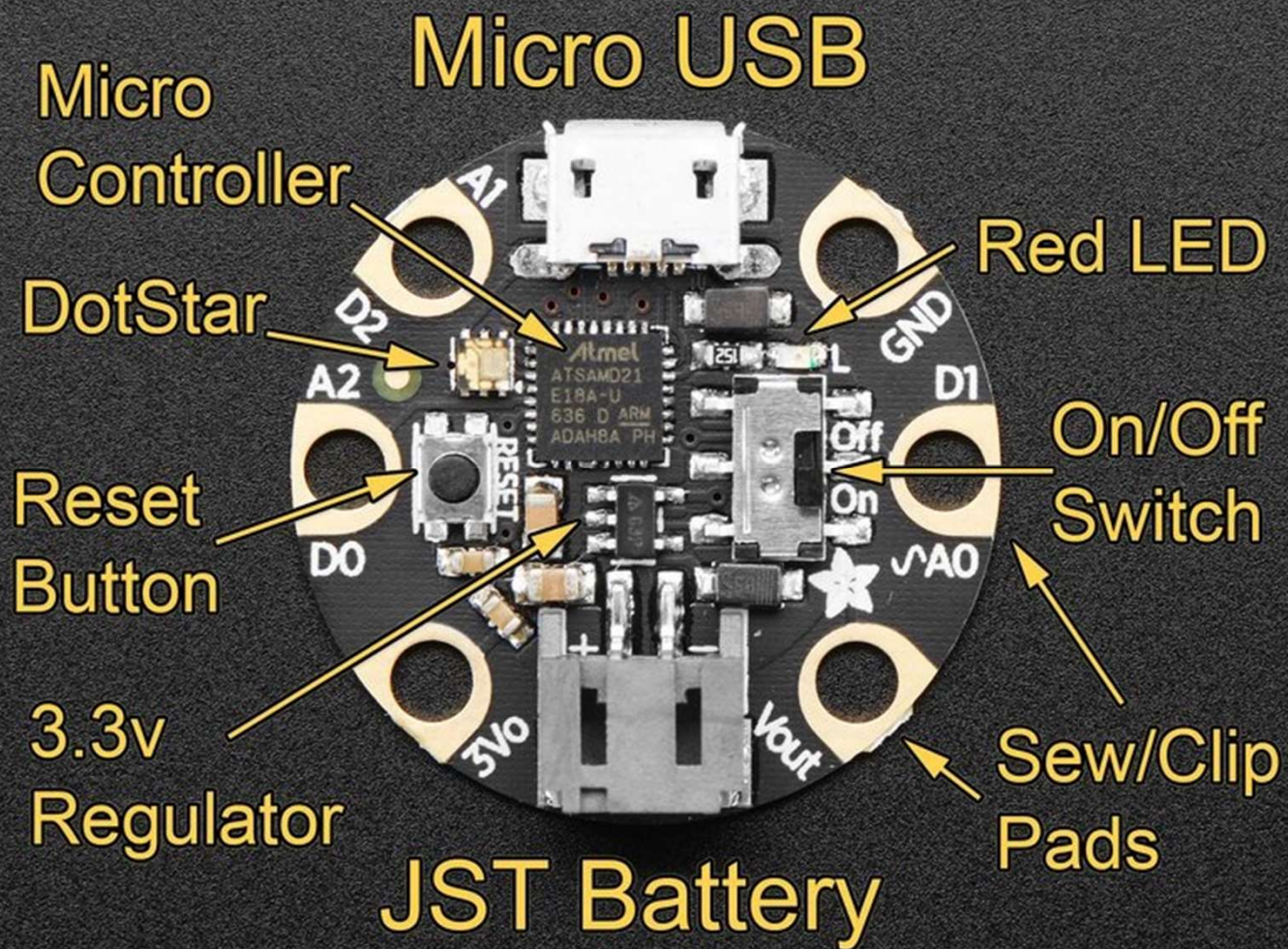
\$6.95 [ADD TO CART](#)



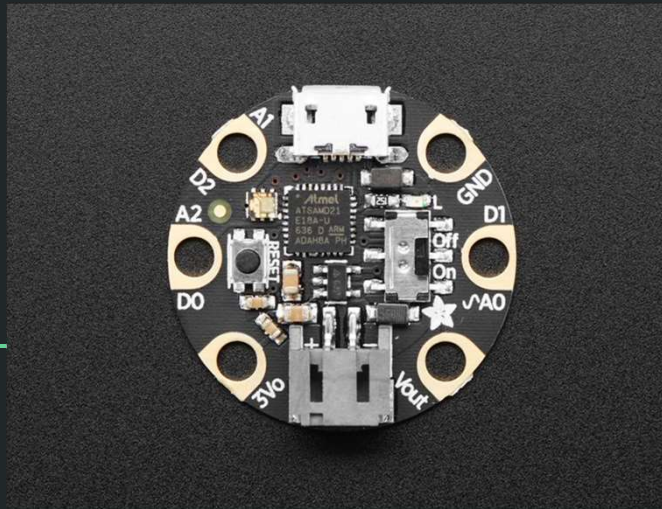
Copper Foil Tape with

Gemma M0 Development Board





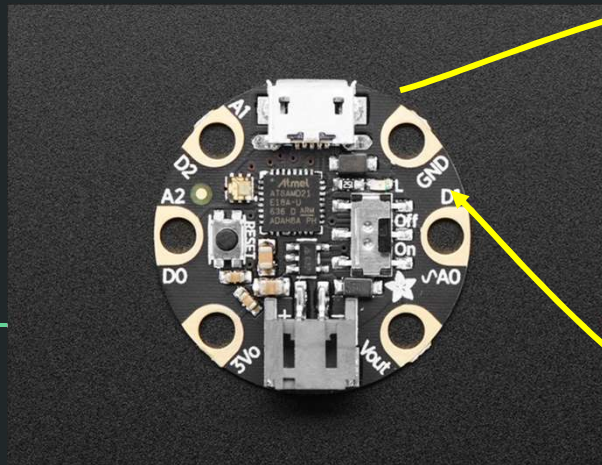
Gemma And CircuitPython



Gemma And REPL

The REPL acts as a monitor and gives feedback on program operation.

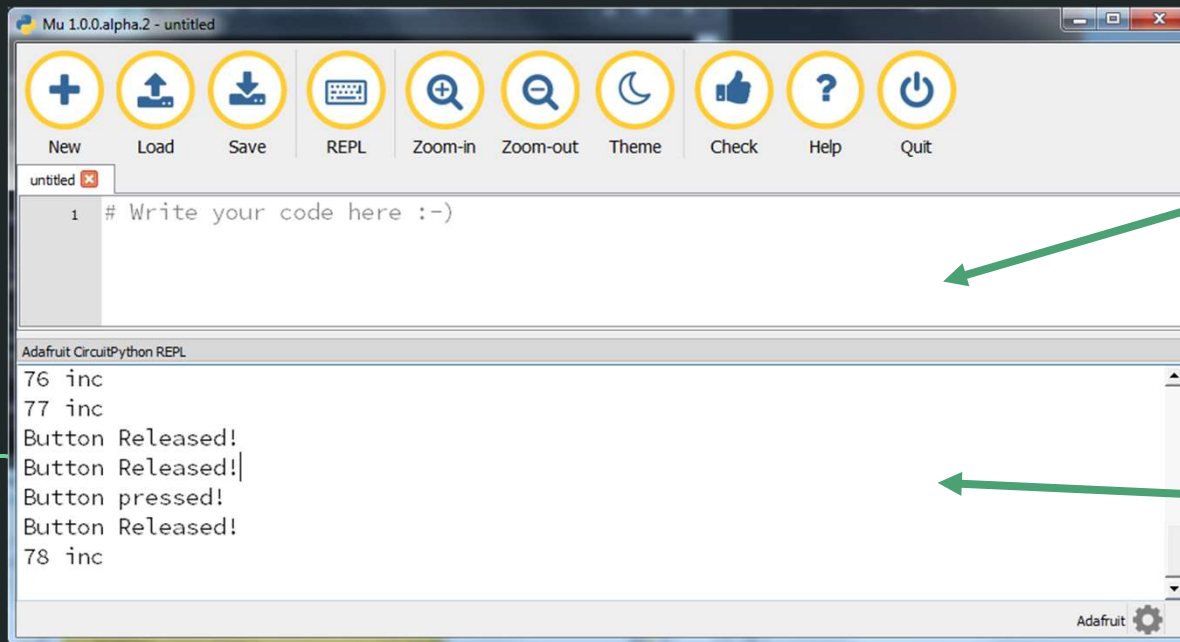
....010101010100111011101101



010101010100111011101101....

Print statements from your program also show up in the monitor screen. (REPL)

Mu Editor



Program area

REPL area

Copy & Paste Technique

Edit one program from the hard disk.

Copy it into main.py to run it

Use control-A to grab the code,

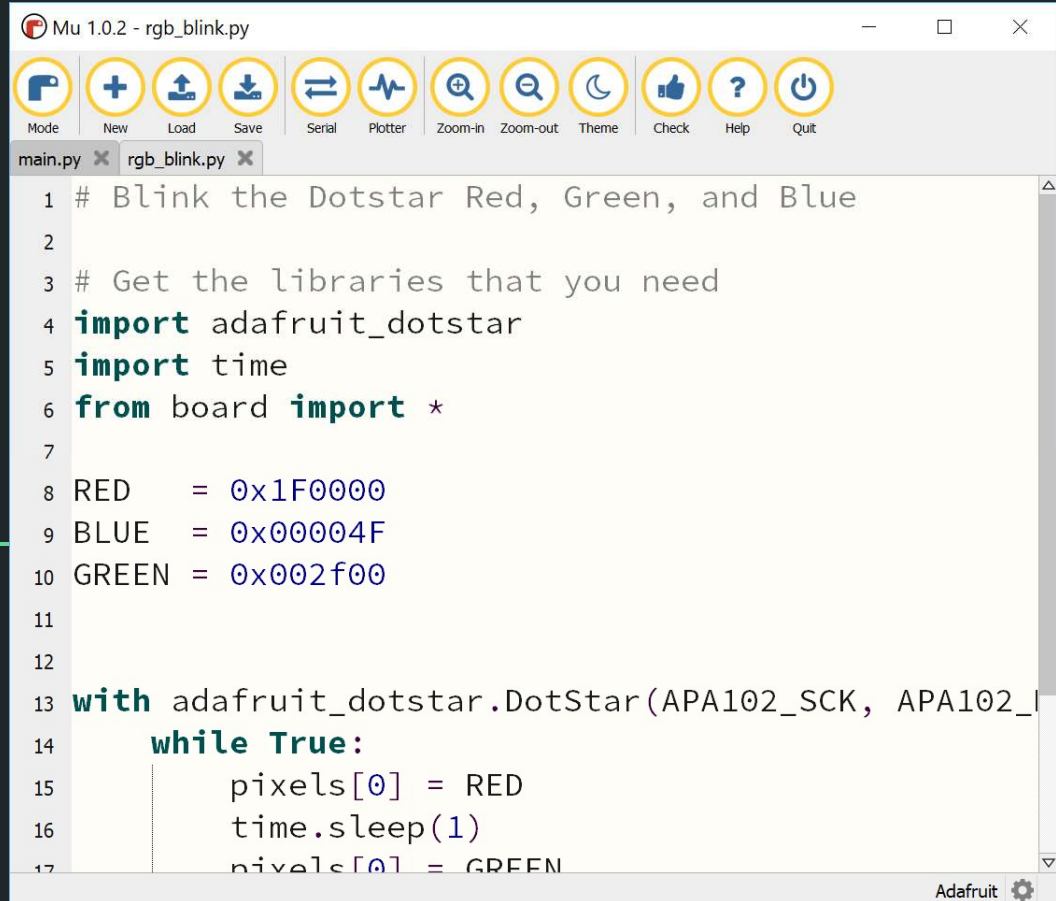
Use control-C to copy the code,

Move to main.py

Use control-A to grab the code,

Use control-V to paste the new code

Save main.py to load it to the Gemma



```
Mu 1.0.2 - rgb_blink.py
Mode New Load Save Serial Plotter Zoom-In Zoom-Out Theme Check Help Quit

main.py x rgb_blink.py x
1 # Blink the Dotstar Red, Green, and Blue
2
3 # Get the libraries that you need
4 import adafruit_dotstar
5 import time
6 from board import *
7
8 RED = 0x1F0000
9 BLUE = 0x00004F
10 GREEN = 0x002f00
11
12
13 with adafruit_dotstar.DotStar(APA102_SCK, APA102_
14     while True:
15         pixels[0] = RED
16         time.sleep(1)
17         pixels[0] = GREEN
```

Copy & Paste Technique

Edit one program from the hard disk.

Copy it into main.py to run it

Use control-A to grab the code,

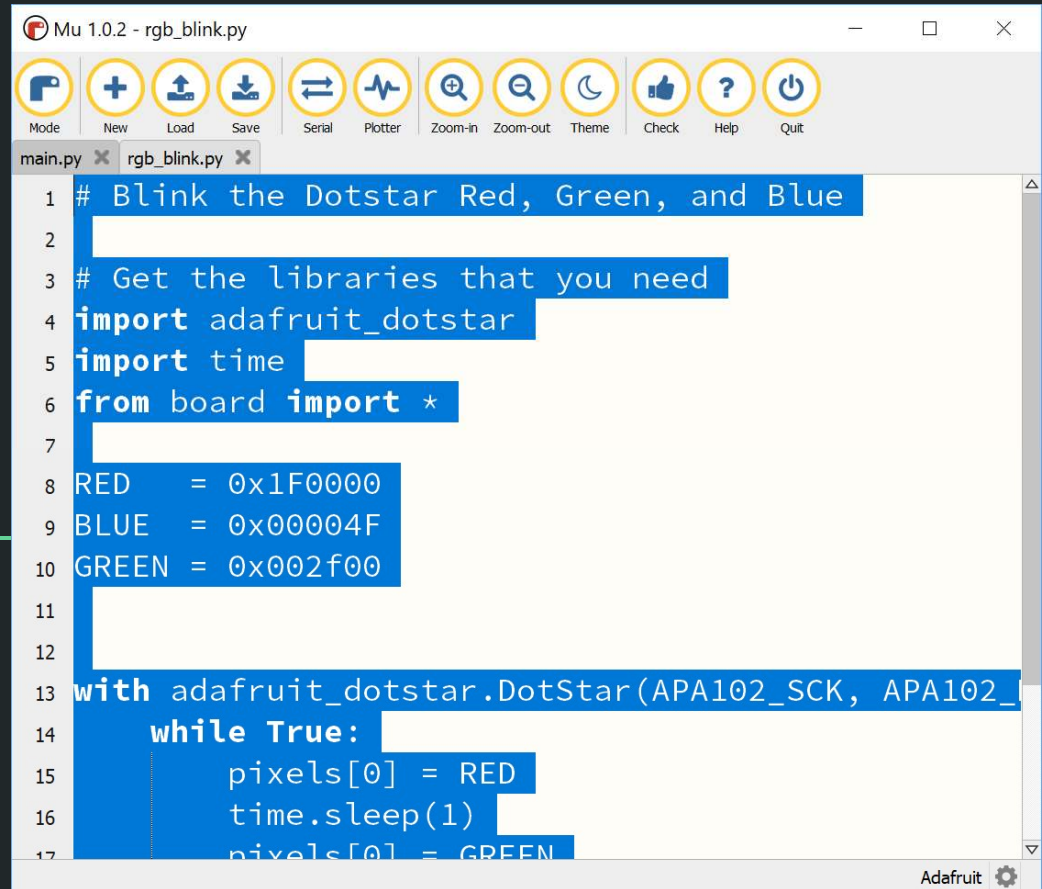
Use control-C to copy the code,

Move to main.py

Use control-A to grab the code,

Use control-V to paste the new code

Save main.py to load it to the Gemma



```
Mu 1.0.2 - rgb_blink.py
Mode New Load Save Serial Plotter Zoom-in Zoom-out Theme Check Help Quit
main.py x rgb_blink.py x
1 # Blink the Dotstar Red, Green, and Blue
2
3 # Get the libraries that you need
4 import adafruit_dotstar
5 import time
6 from board import *
7
8 RED = 0x1F0000
9 BLUE = 0x00004F
10 GREEN = 0x002f00
11
12
13 with adafruit_dotstar.DotStar(APA102_SCK, APA102_
14     while True:
15         pixels[0] = RED
16         time.sleep(1)
17         pixels[0] = GREEN
```

Copy & Paste Technique

Edit one program from the hard disk.

Copy it into main.py to run it

Use control-A to grab the code,

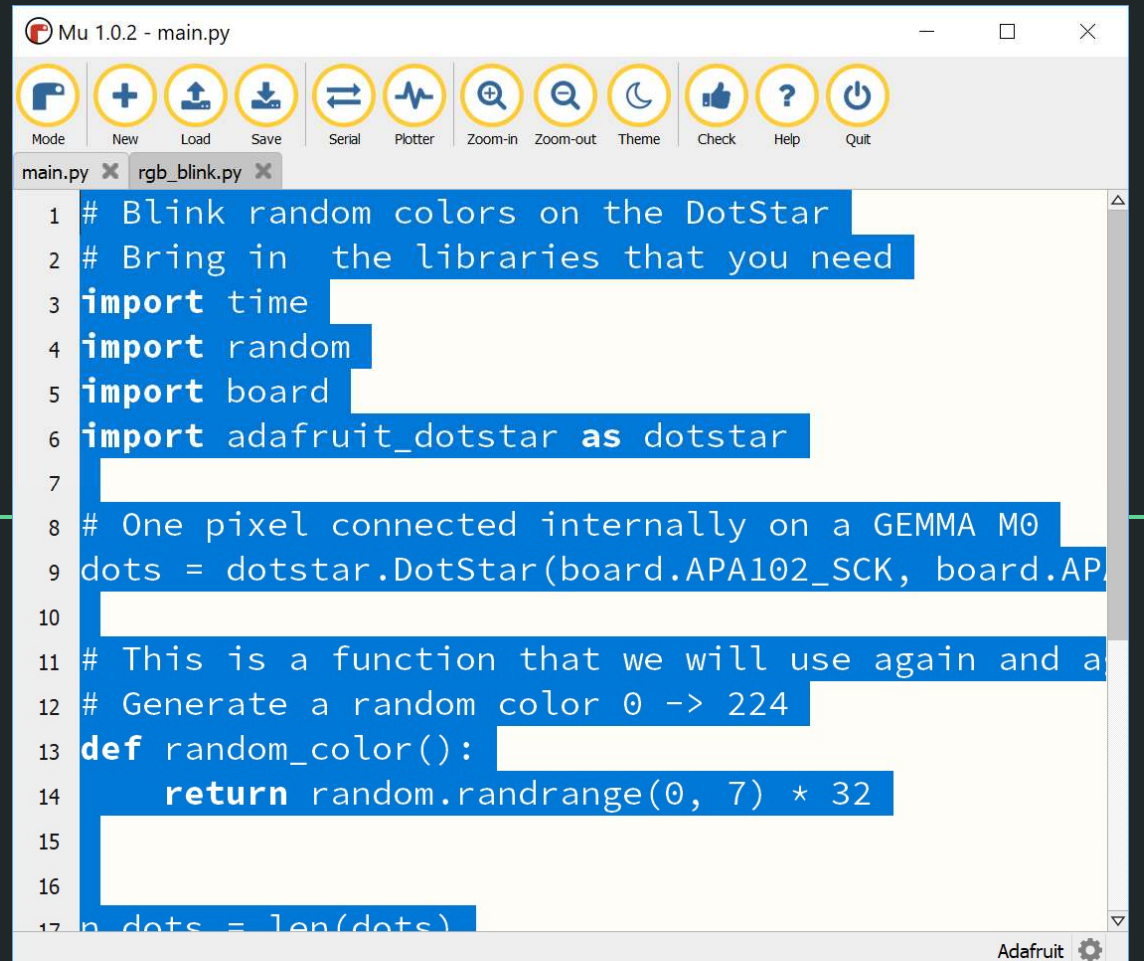
Use control-C to copy the code,

Move to main.py

Use control-A to grab the code,

Use control-V to paste the new code

Save main.py to load it to the Gemma



The screenshot shows the Mu 1.0.2 IDE interface. The title bar reads "Mu 1.0.2 - main.py". The menu bar includes icons for Mode, New, Load, Save, Serial, Plotter, Zoom-In, Zoom-Out, Theme, Check, Help, and Quit. The file explorer shows two tabs: "main.py" and "rgb_blink.py". The code editor displays the following Python code:

```
1 # Blink random colors on the DotStar
2 # Bring in the libraries that you need
3 import time
4 import random
5 import board
6 import adafruit_dotstar as dotstar
7
8 # One pixel connected internally on a GEMMA M0
9 dots = dotstar.DotStar(board.APA102_SCK, board.APA102_MOSI, 1)
10
11 # This is a function that we will use again and again
12 # Generate a random color 0 -> 224
13 def random_color():
14     return random.randrange(0, 7) * 32
15
16
17 n_dots = len(dots)
```

The Adafruit logo is visible in the bottom right corner of the IDE window.

Copy & Paste Technique

Edit one program from the hard disk.

Copy it into main.py to run it

Use control-A to grab the code,

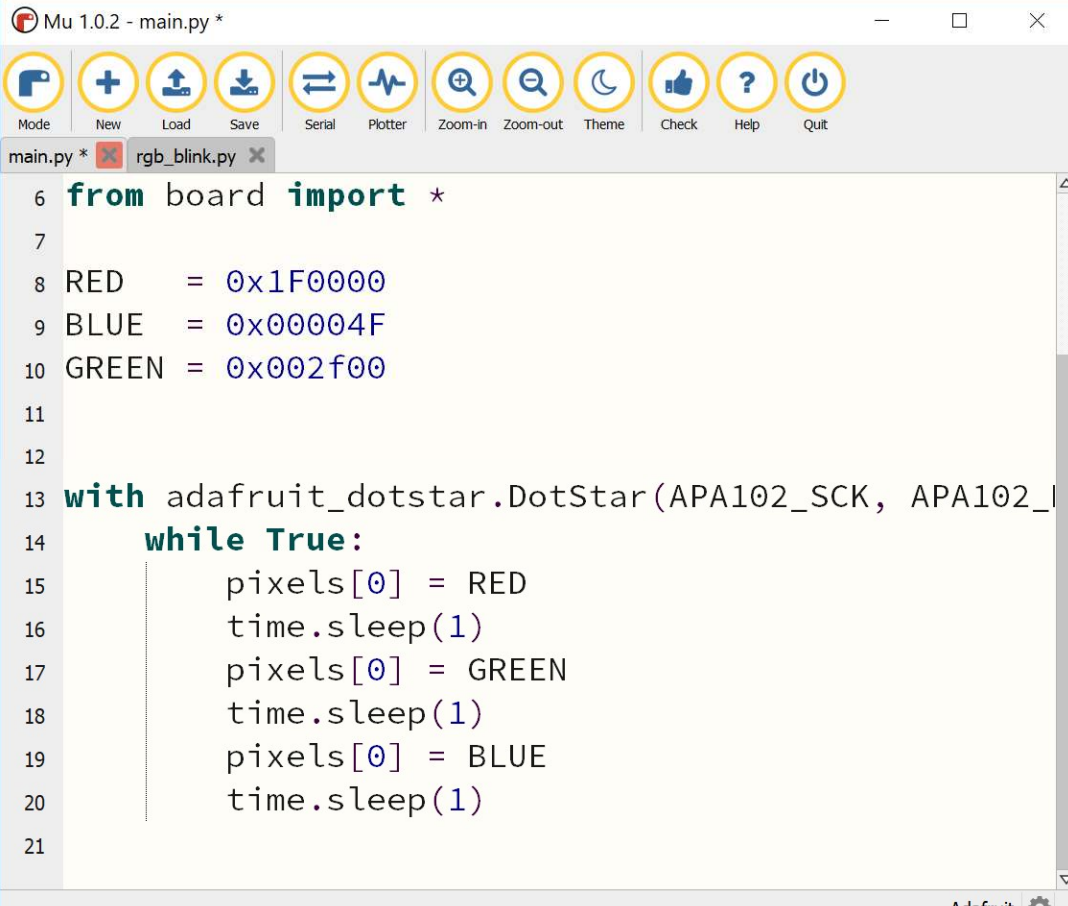
Use control-C to copy the code,

Move to main.py

Use control-A to grab the code,

Use control-V to paste the new code

Save main.py to load it to the Gemma



```
Mu 1.0.2 - main.py *  
Mode New Load Save Serial Plotter Zoom-in Zoom-out Theme Check Help Quit  
main.py * x rgb_blink.py x  
6 from board import *  
7  
8 RED = 0x1F0000  
9 BLUE = 0x00004F  
10 GREEN = 0x002f00  
11  
12  
13 with adafruit_dotstar.DotStar(APA102_SCK, APA102_...  
14 while True:  
15     pixels[0] = RED  
16     time.sleep(1)  
17     pixels[0] = GREEN  
18     time.sleep(1)  
19     pixels[0] = BLUE  
20     time.sleep(1)  
21  
Adafruit
```

while_loop.py

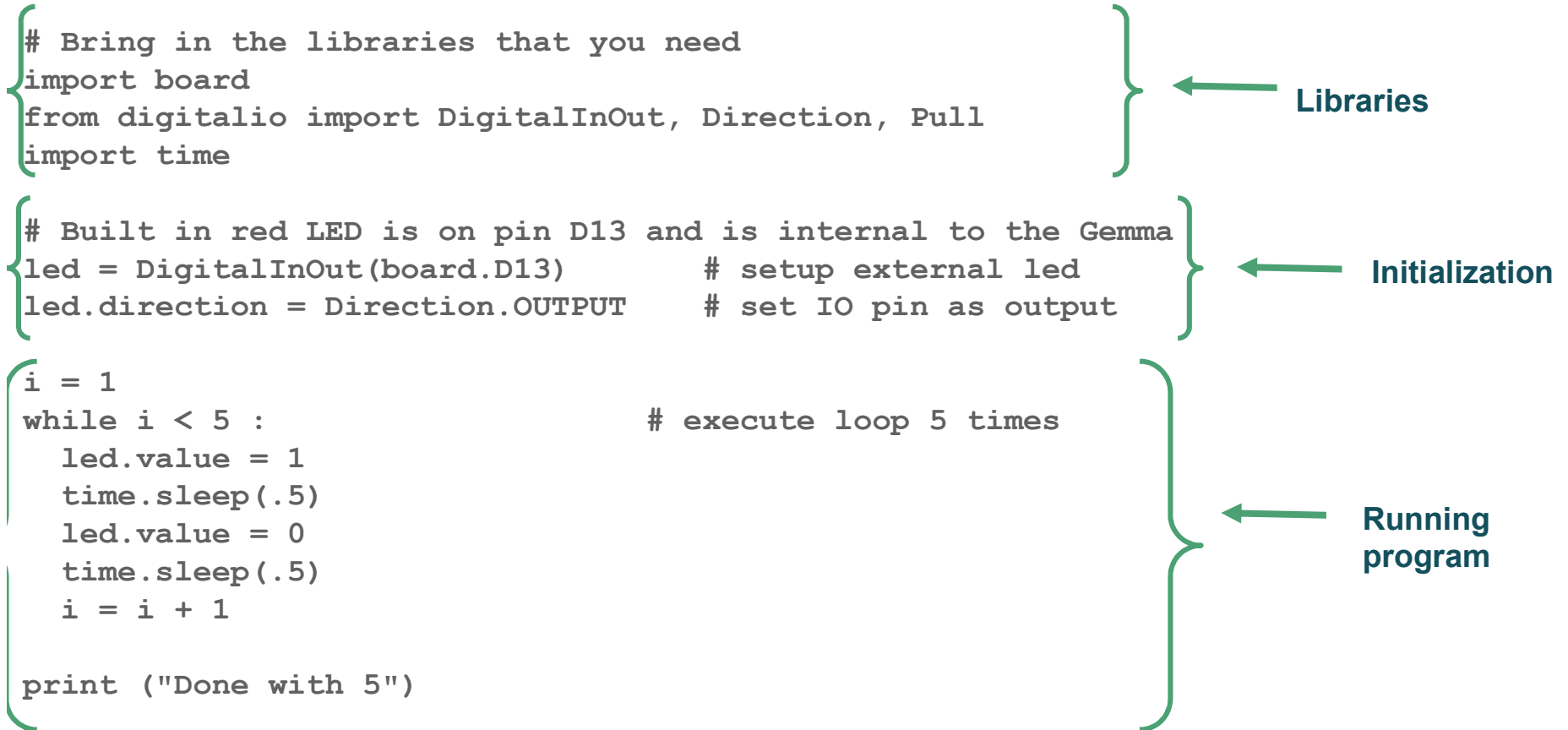
```
# Bring in the libraries that you need
import board
from digitalio import DigitalInOut, Direction, Pull
import time

# Built in red LED is on pin D13 and is internal to the Gemma
led = DigitalInOut(board.D13)      # setup external led
led.direction = Direction.OUTPUT  # set IO pin as output

i = 1
while i < 5 :                       # execute loop 5 times
    led.value = 1
    time.sleep(.5)
    led.value = 0
    time.sleep(.5)
    i = i + 1

print ("Done with 5")
```

Anatomy of Circuitpython programs



Python Basics - Variables

```
>>> i = 1  
>>> print(i)  
1
```

```
>>> j = 2.  
>>> print(j)  
2.0
```

```
>>> name = "Coco"  
>>> print(name)  
Coco
```

for_loop.py

```
# Bring in the libraries that you need
import board
from digitalio import DigitalInOut, Direction, Pull
import time

# Built in red LED is on pin D13 and is internal to the Gemma
led = DigitalInOut(board.D13)      # setup external led
led.direction = Direction.OUTPUT  # set IO pin as output

# Execute loop 5 times
for i in range (5):
    led.value = 1
    time.sleep(.5)
    led.value = 0
    time.sleep(.5)

print ("Done with 5")
```

ex_led_blink.py

```
# Bring in the libraries that you need
import board
from digitalio import DigitalInOut, Direction
import time

# Built in red LED is on pin D13 and is internal to the Gemma
led = DigitalInOut(board.D13)      # setup external led
led.direction = Direction.OUTPUT   # set IO pin as output
exled = DigitalInOut(board.D1)     # setup external led
exled.direction = Direction.OUTPUT # and the IO pin as output


while True:                        # execute loop forever
    led.value = 1
    exled.value = 0
    time.sleep(4)
    led.value = 0
    exled.value = 1
    time.sleep(4)
```

ex_led_blink2.py

```
# Bring in the libraries that you need
import board
from digitalio import DigitalInOut, Direction
import time

# Built in red LED is on pin D13 and is internal to the Gemma
led = DigitalInOut(board.D13)      # setup external led
led.direction = Direction.OUTPUT   # set IO pin as output
exled = DigitalInOut(board.D1)     # setup external led
exled.direction = Direction.OUTPUT # and the IO pin as output
ON  = 1
OFF = 0

while True:                        # execute loop forever
    led.value = OFF
    exled.value = OFF
    time.sleep(4)
    led.value = ON
    exled.value = ON
    time.sleep(4)
```



dot_star0.py

```
import adafruit_dotstar
import time
from board import *

RED = 0x100000

with adafruit_dotstar.DotStar(APA102_SCK, APA102_MOSI, 1) as pixels:
    pixels[0] = RED
    time.sleep(2)
```


dot_star1.py

```
# Blink random colors on the DotStar
# Bring in the libraries that you need
import time
import random
import board
import adafruit_dotstar as dotstar

# One pixel connected internally on a GEMMA M0
dots = dotstar.DotStar(board.APA102_SCK, board.APA102_MOSI, 1, brightness=0.2)

# This is a function that we will use again and again
# Generate a random color 0 -> 224
def random_color():
    return random.randrange(0, 7) * 32

n_dots = len(dots)
while True:
    #fill each dot with a random color
    for dot in range(n_dots):
        dots[dot] = (random_color(), random_color(), random_color())
    # show all dots in strip
    dots.show()
    time.sleep(.25)
```



wink.py


```
# Use both of the LED's on the Gemma to wink together

# Get the libraries that you need
import adafruit_dotstar
import time
from board import *
from digitalio import DigitalInOut, Direction

# Built in red LED
led = DigitalInOut(D13)
led.direction = Direction.OUTPUT      # set IO pin as output

RED = 0x1F0000
OFF = 0x000000

with adafruit_dotstar.DotStar(APA102_SCK, APA102_MOSI, 1) as pixels:
    while True:
        pixels[0] = RED
        led.value = 1
        time.sleep(2)
        pixels[0] = OFF
        led.value = 0
        time.sleep(2)
```



wink2.py

```
# Use both of the LED's on the Gemma to wink together
```

```
. . .
```

```
RED = 0x1F0000
```

```
OFF = 0x000000
```

```
with adafruit_dotstar.DotStar(APA102_SCK, APA102_MOSI, 1) as pixels:
```

```
    while True:
```

```
        for i in range (16):
```

```
            pixels[0] = RED
```

```
            time.sleep(.1)
```

```
            pixels[0] = OFF
```

```
            time.sleep(.1)
```

```
        for i in range (16):
```

```
            led.value = 1
```

```
            time.sleep(.1)
```

```
            led.value = 0
```

```
            time.sleep(.1)
```


Reference material

Adafruit Industries: www.adafruit.com

Gemma M0 learning: www.learn.adafruit.com/adafruit-gemma-m0?view=all

Sparkfun electronics: www.sparkfun.com

Mu-Editor Installation:

<https://codewith.mu/en/download>

Seminar Slides:

<https://docs.google.com/presentation/d/1-rhsECbdl-QQHVEifSHUANQlVRY3bAzd5QX7KLlMLc/edit?usp=sharing>

Python:

<https://www.w3schools.com/python/default.asp>

<https://www.python.org/> (advanced full reference for Python)

<https://wiki.python.org/moin/BeginnersGuide> (still pretty advanced)

Electronics basics videos

Ed10kms Channel