BEFORE WE BEGIN:

You got an email!

- **Install MU**
- **Open the In-Class Resource Doc**
- **Download the libraries**

micro motion

movement-responsive microcontrollers

Also, introduce yourself to your fellow students!

We'll be working with each other.

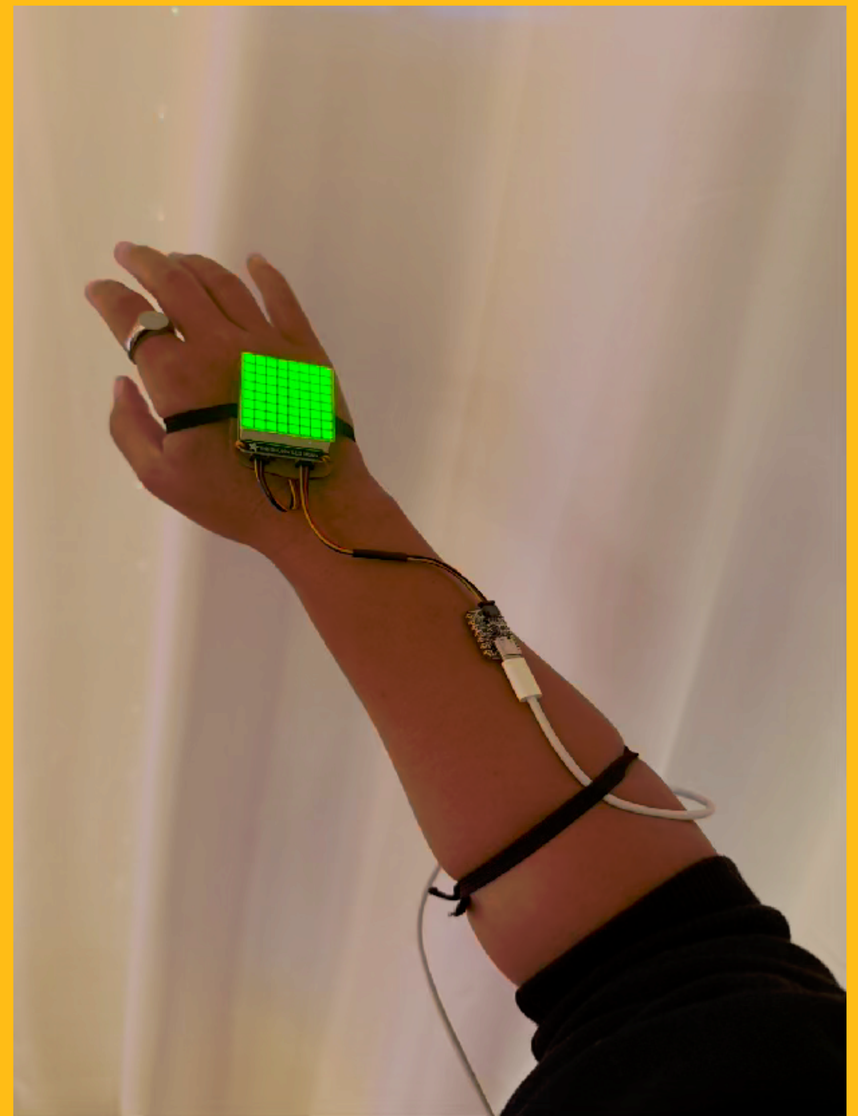
~ leia s chang

i’m Leia

they/them

@leia.make

what you'll make



class overview

- intro to microcontrollers
- talking to your board
- adding peripherals
- the accelerometer
- the led matrix
- time to play

Hardware[01]:

QTPY RP2040

LIS3DH accelerometer

8x8 bicolor LED matrix

The Don’ts

- Don’t put your powered circuits on metal!
- Don’t force a plug into a socket
- Don’t struggle alone; help each other out!

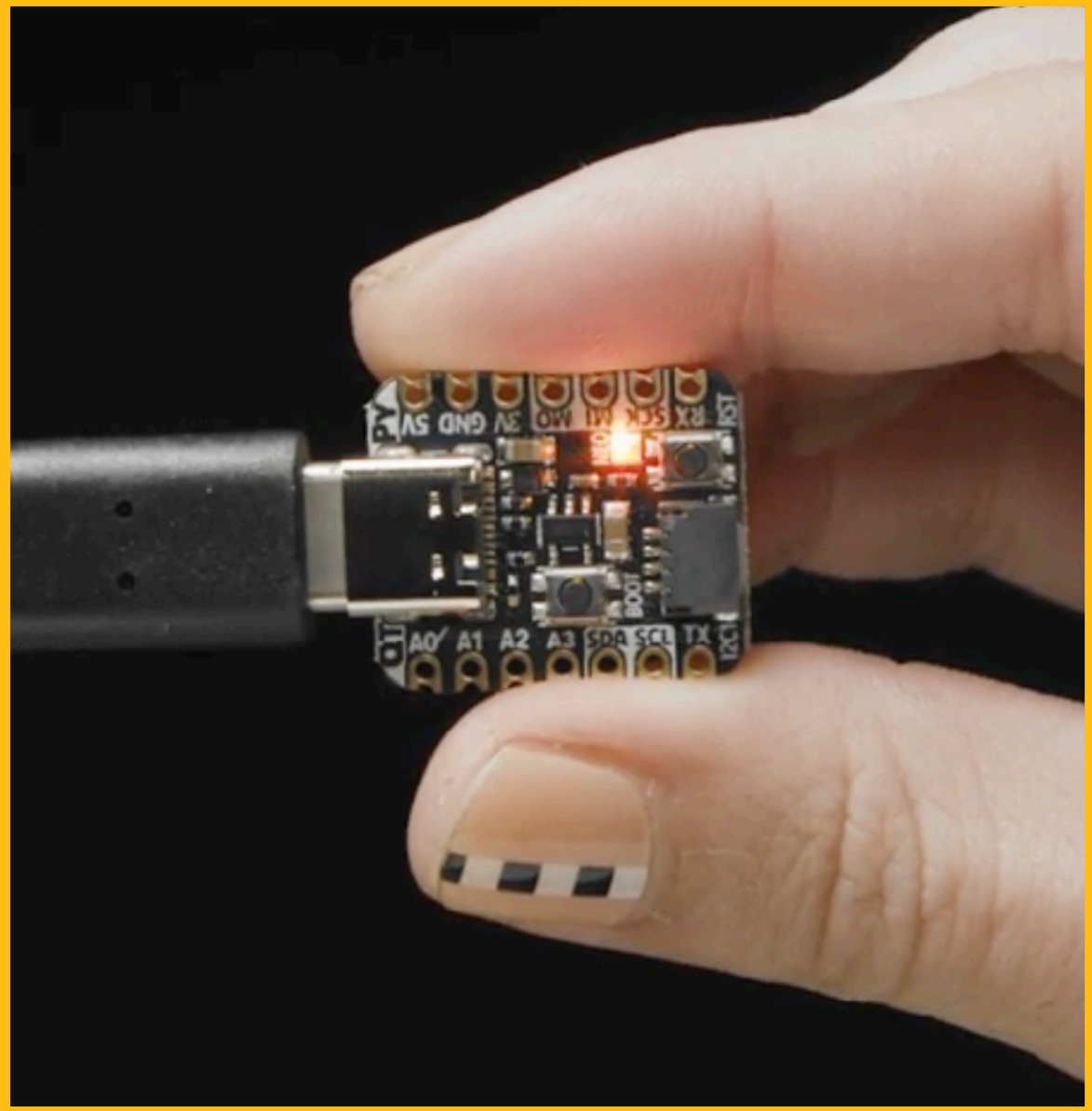
Shit happens … static electricity and shorted circuits can kill a board.

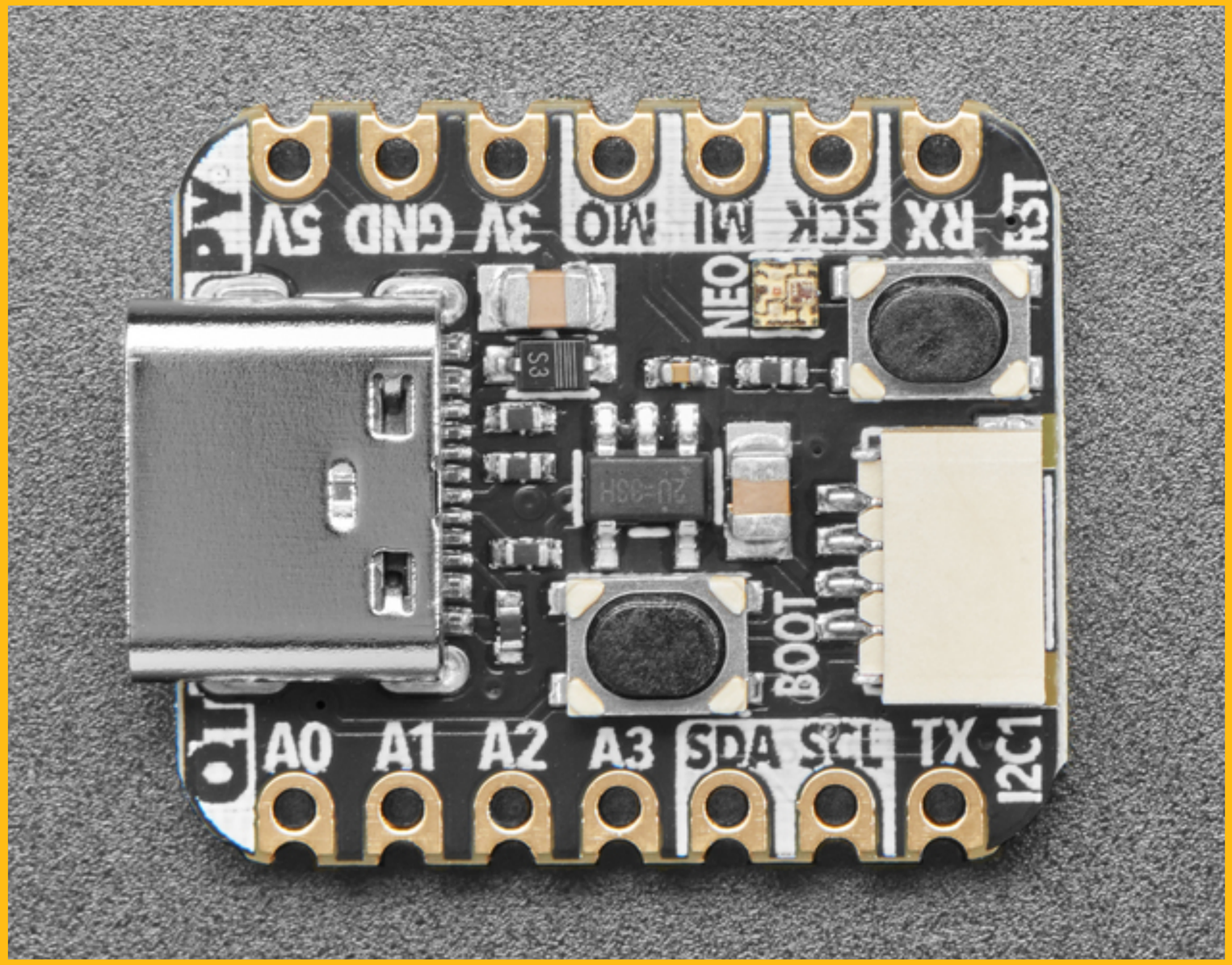
QTPy RP2040

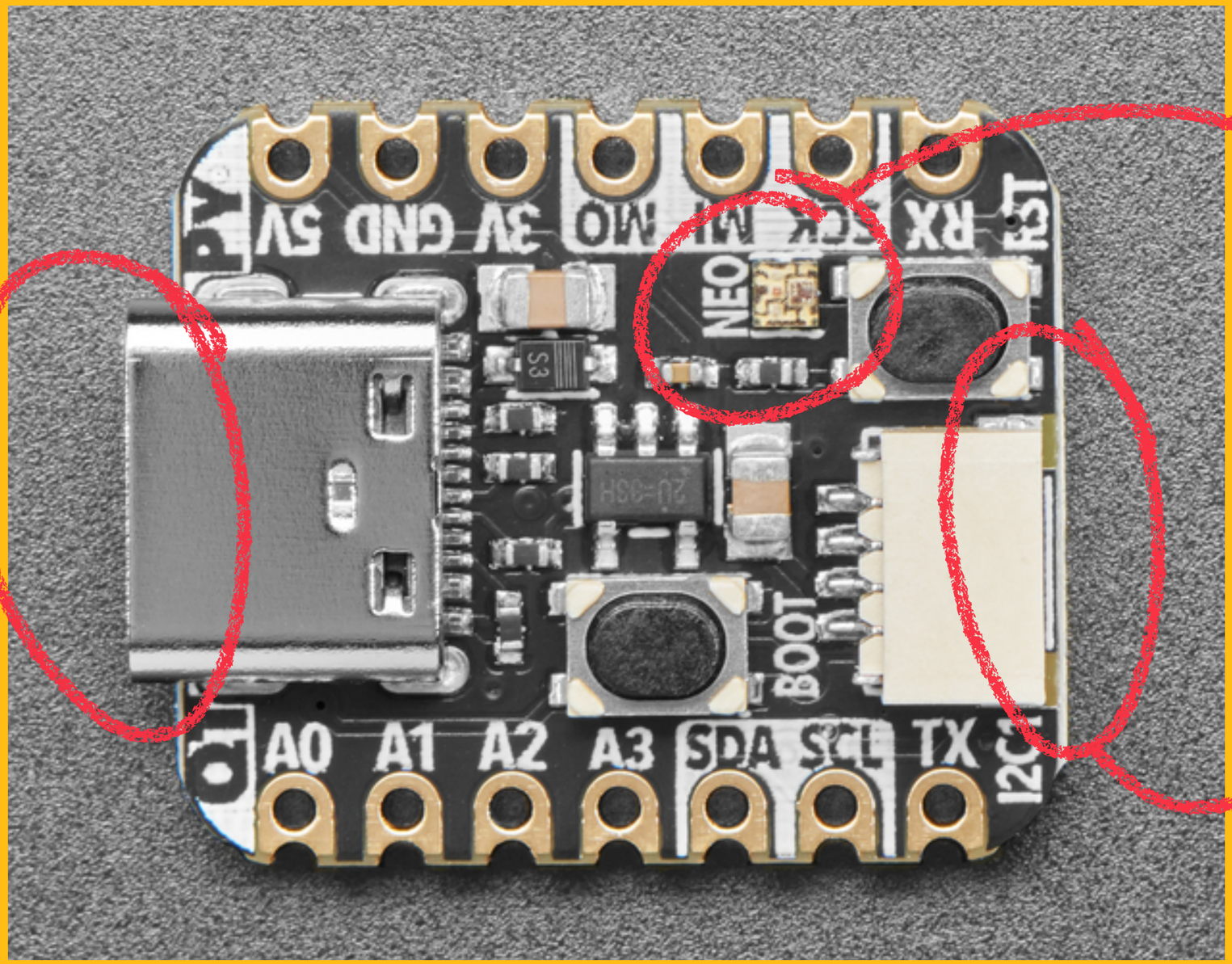
your microcontroller

- This is what’s gonna hold our code and talk to all the other stuff
- QTPy is the form factor
- RP2040 is the chip/brains

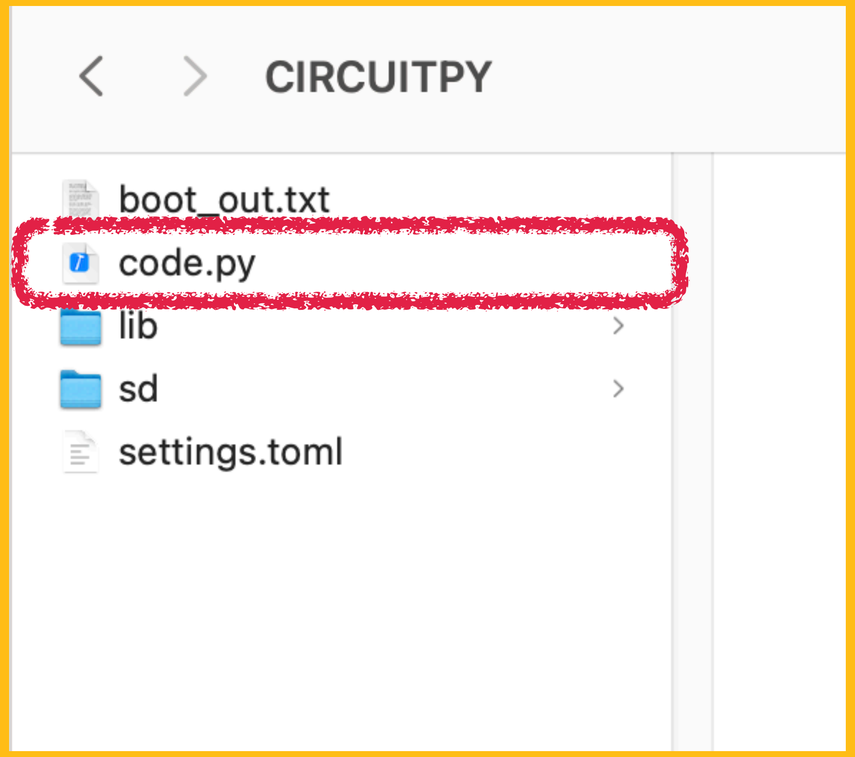
“The Board”




5V
GND
3V
MO
MI
SCK
RX
RST
NEO
S3
BOOT
A0
A1
A2
A3
SDA
SCL
TX
I2C1


USB-C
LED
STEMMA
QWIIC
CONNECTOR
5V
GND
3V
MO
SCK
RX
RST
NEO
BOOT
A0
A1
A2
A3
SDA
SCL
TX
I2C

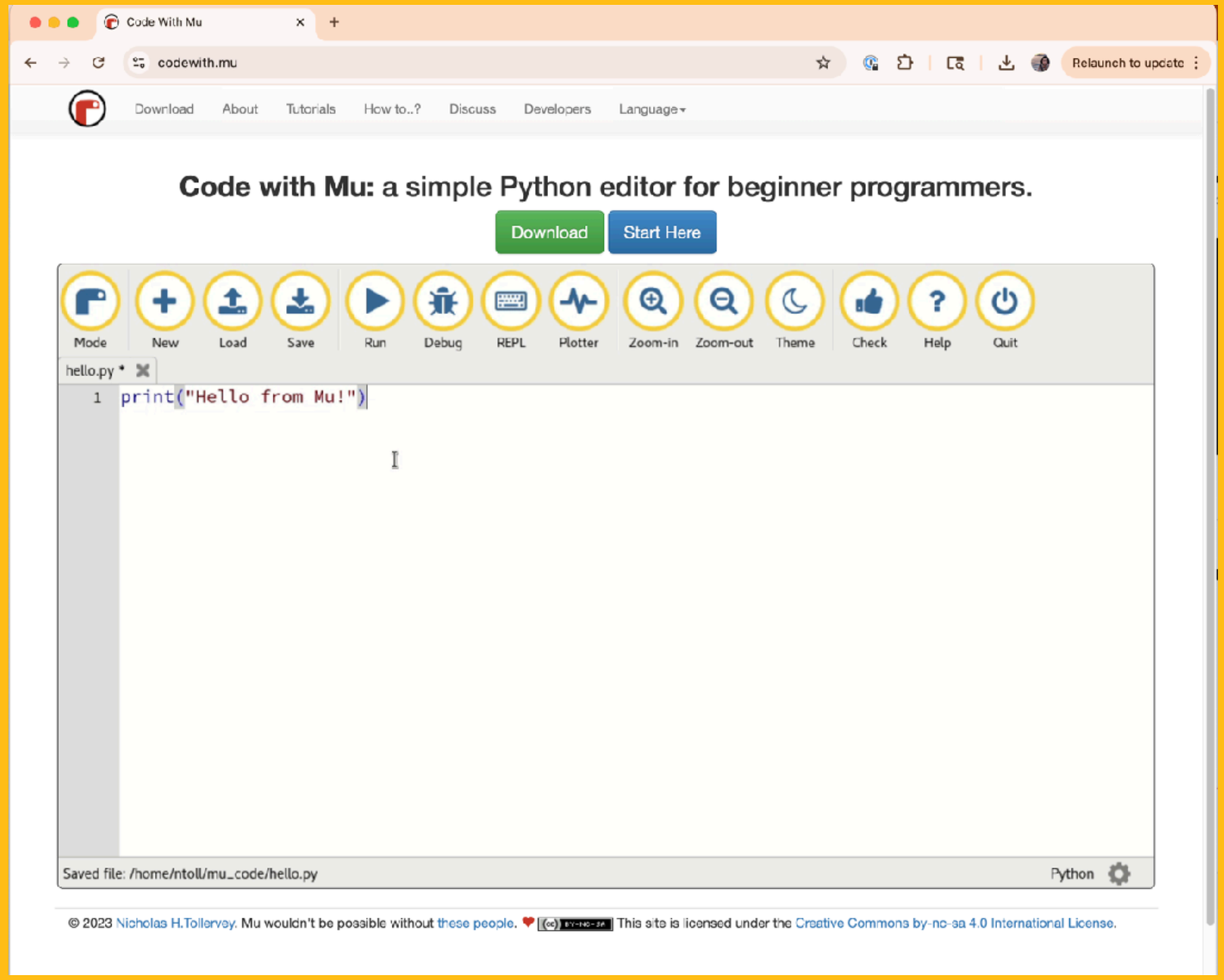
1. Plug in your microcontroller
2. Open it in your finder/ file explorer



Keep in mind …

- Code lives in **code.py**

Let's Open Mu[02]



https://codewith.mu/en/

Find a partner or a trio.

Get your board to say “Hello” to your partner by updating the text with their name.

a little bit about CircuitPython

- runs when you save
- code lives in code.py
- formatting matters

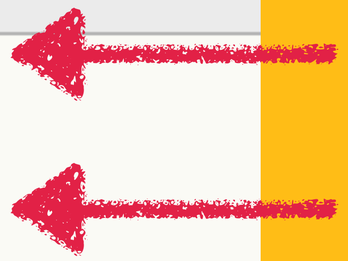
more code 01

Keep in mind …

- Code lives in **code.py**
- Open Serial to read from your microcontroller
- Formatting matters!

code.py

```
1  import time
2
3  while True:
4      print("Hello")
5      time.sleep(1)
6
7  print("I never print!")
```



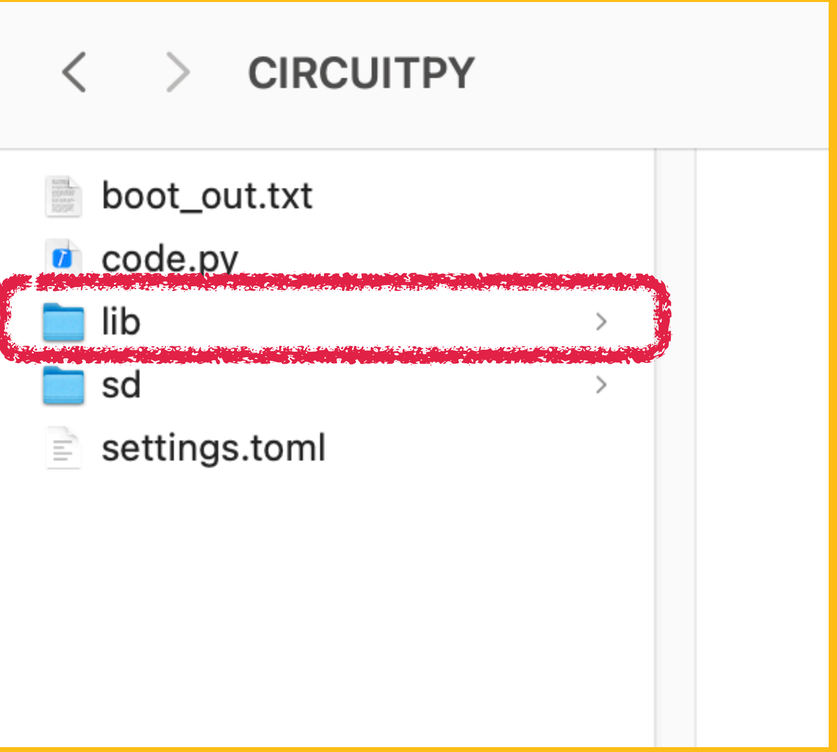
- copy the code
- paste into Mu, replacing everything in code.py
- save code.py to run
- try changing time.sleep()

https://github.com/leils/workshop-resources/blob/main/micromovement_ga2025/example%20code/01%20repeated%20hellos/code.py

more code 02

```
1  import time
2  import board
3  import neopixel
4
5  pixels = neopixel.NeoPixel(board.NEOPIXEL, 1)
6
7  while True:
8      pixels.fill((255, 0, 0))
9      time.sleep(0.5)
10     pixels.fill((0, 0, 0))
11     time.sleep(0.5)
```

https://github.com/leils/workshop-resources/blob/main/micromovement_ga2025/example%20code/02%20neopixel%20blink/code.py

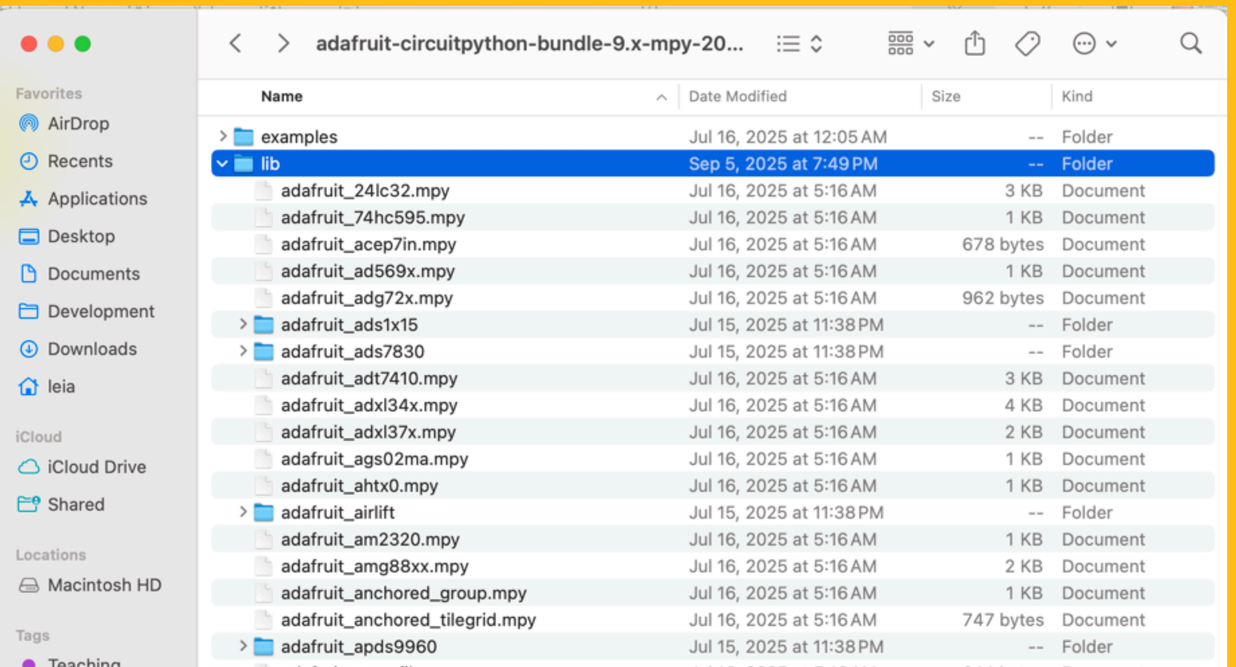


https://circuitpython.org/libraries

Keep in mind …

- Code lives in **code.py**
- Open Serial to read from your microcontroller
- Formatting matters!
- Make sure you have your libraries

libraries



adafruit_pixelbuf.mpy
neopixel.mpy

```
1  import time
2
3  while True:
4      print("Hello!")
5      time.sleep(0.1)
```

adafruit_pixelbuf.mpy
neopixel.mpy

```
1  import time
2  import board
3  import neopixel
4
5  pixels = neopixel.NeoPixel(board.NEOPIXEL, 1)
6
7  while True:
8      pixels.fill((255, 0, 0))
9      time.sleep(0.5)
10     pixels.fill((0, 0, 0))
11     time.sleep(0.5)
```

Keep in mind …

- Code lives in **code.py**
- Open Serial to read from your microcontroller
- Formatting matters!
- Make sure you have your libraries

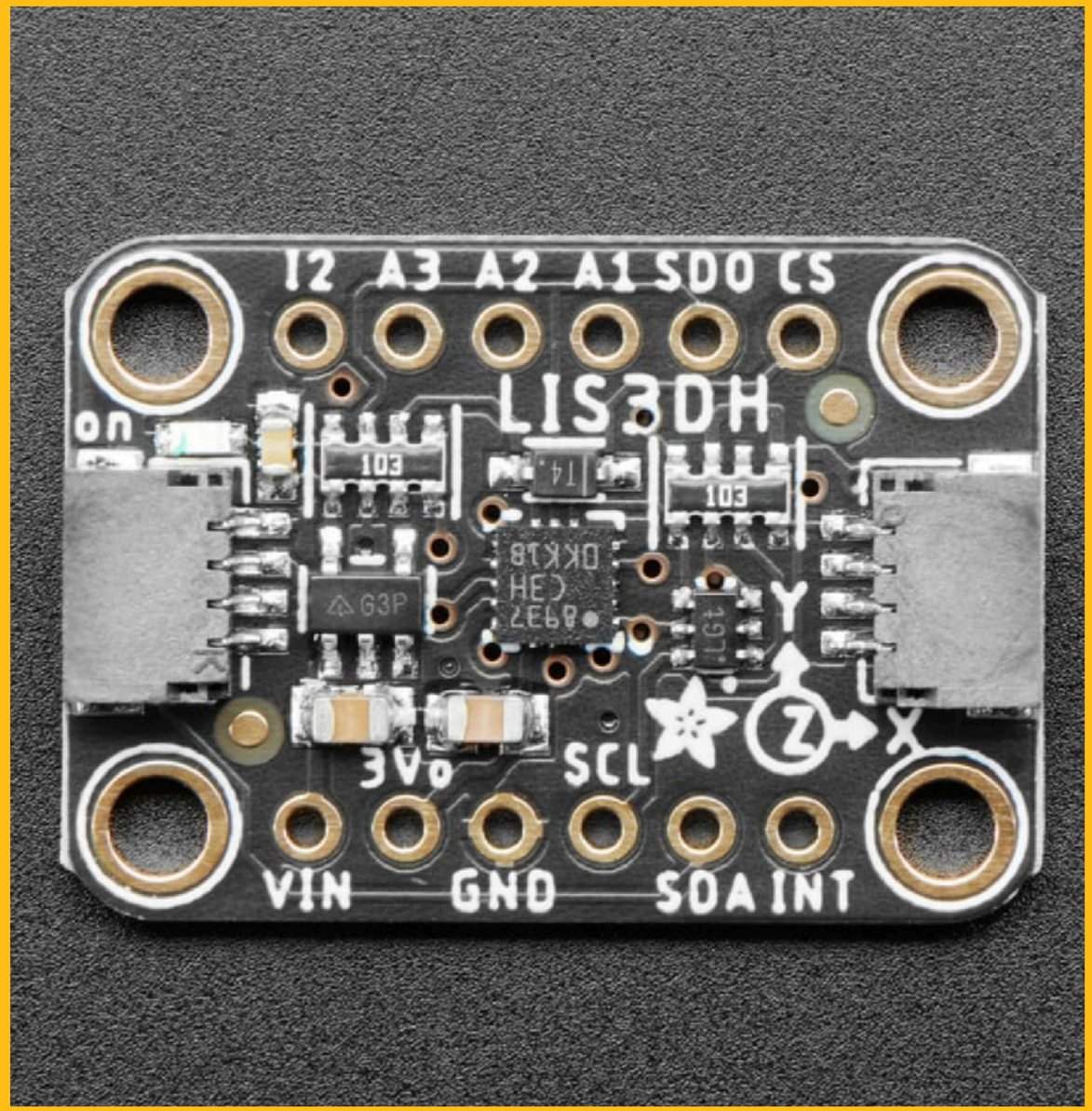
play ar e

Some challenges:

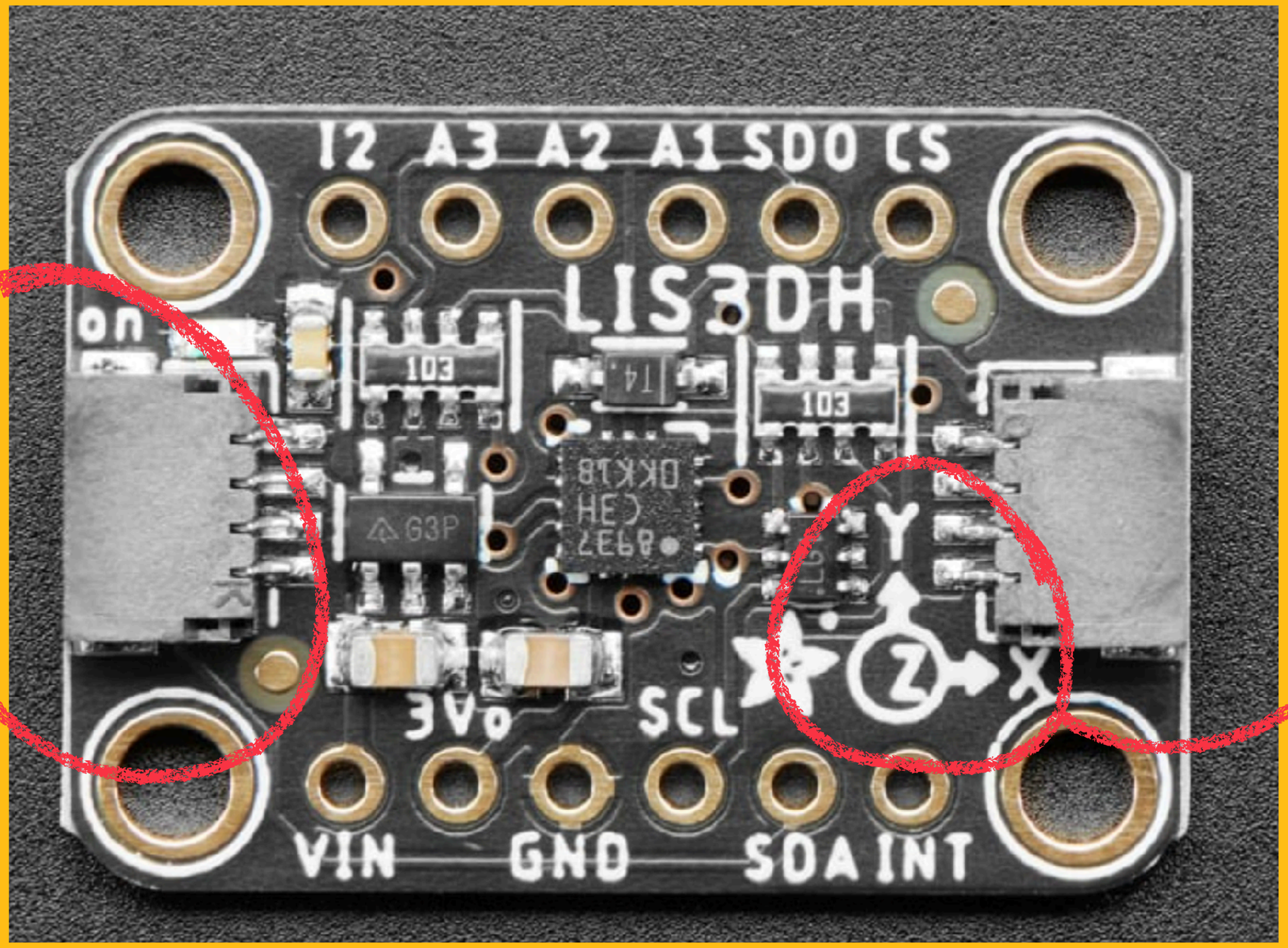
- Get your microcontroller to say your name or recite a poem
- Play around with the Neopixel; can you make it show your favorite color? Can you make it display a rainbow?
- Can you make it print and change colors at the same time?

LIS3DH Accelerometer

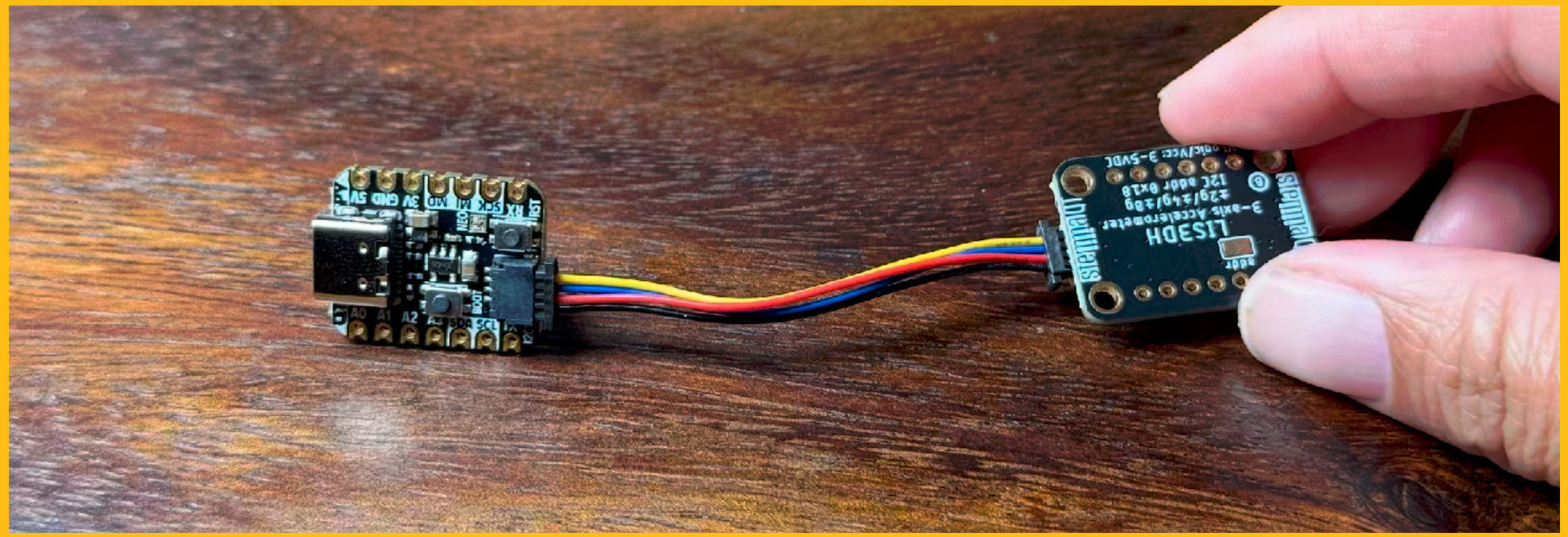
- We need the adafruit_lis3dh.mpy library



adafruit_lis3dh.mpy library



adafruit_lis3dh.mpy library



accelerometer code

```
1  import board
2  import adafruit_lis3dh
3  import time
4
5  i2c = board.STEMMA_I2C()  # For using the built-in STEMMA QT connector on a microcontroller
6  lis3dh = adafruit_lis3dh.LIS3DH_I2C(i2c)
7  lis3dh.range = adafruit_lis3dh.RANGE_2_G
8
9  # Loop forever printing accelerometer values
10 while True:
11     # Read accelerometer values (in m / s ^ 2).  Returns a 3-tuple of x, y,
12     # z axis values.  We divide them by 9.806 (STANDARD_GRAVITY) to convert to Gs.
13     x, y, z = (value / adafruit_lis3dh.STANDARD_GRAVITY for value in lis3dh.acceleration)
14     print(f"x = {x:.3f} G, y = {y:.3f} G, z = {z:.3f} G")
15
16     time.sleep(0.1)
```

adafruit_lis3dh.mpy library

https://github.com/leils/workshop-resources/blob/main/micromovement_ga2025/example%20code/03%20accelerometer/code.py

Mu 1.2.0

Mode | New | Load | Save | Serial | Plotter | Zoom-in | Zoom-out | T

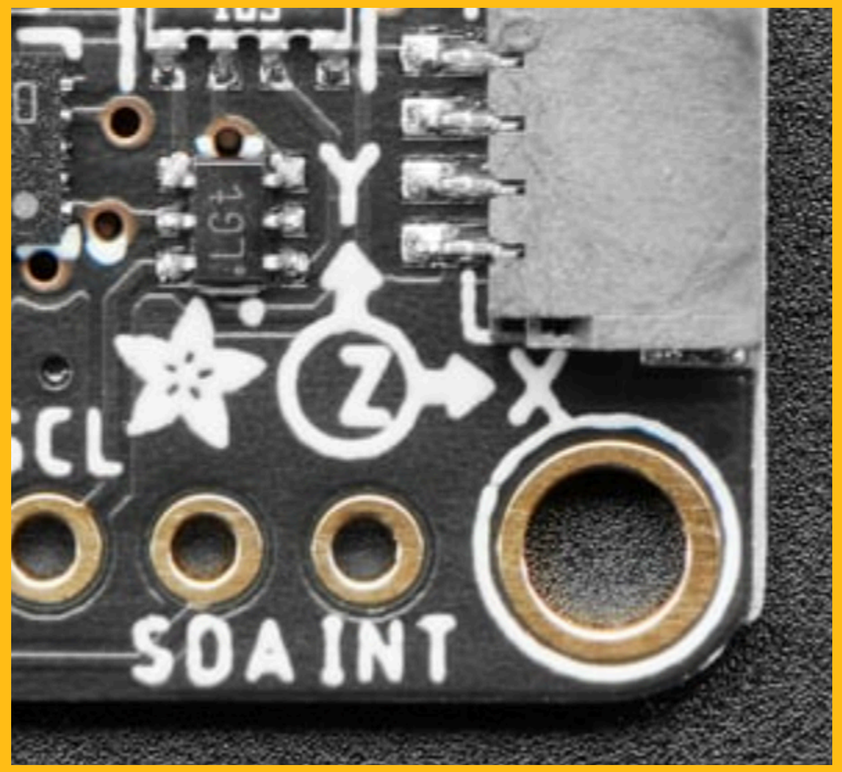
code.py

```
2  import adafruit_lis3dh
3  import time
4
5  from adafruit_ht16k33.matrix import Matrix8x8x2
6
7  i2c = board.STEMMA_I2C()
8  matrix = Matrix8x8x2(i2c)
9  lis3dh = adafruit_lis3dh.LIS3DH_I2C(i2c)
10
11 matrix[0,0] = matrix.LED_GREEN
```

CircuitPython REPL

```
x = -0.170 G, y = 0.228 G, z = -0.916 G
x = -0.168 G, y = 0.232 G, z = -0.904 G
x = -0.169 G, y = 0.229 G, z = -0.910 G
x =  0.168 G, y = 0.234 G, z =  0.914 G
x = -0.151 G, y = 0.224 G, z = -0.907 G
x = -0.167 G, y = 0.223 G, z = -0.913 G
x =  0.170 G, y = 0.229 G, z =  0.904 G
x = -0.166 G, y = 0.232 G, z = -0.890 G
x = -0.167 G, y = 0.232 G, z = -0.881 G
x =  0.172 G, y = 0.229 G, z =  0.935 G
x = -0.171 G, y = 0.230 G, z = -0.906 G
x = -0.181 G, y = 0.226 G, z = -0.922 G
x = -0.170 G, y = 0.226 G, z = -0.857 G
x = -0.153 G, y = 0.225 G, z = -0.904 G
x = -0.172 G, y = 0.231 G, z = -0.905 G
x = -0.186 G, y = 0.229 G, z = -0.902 G
x = -0.184 G, y = 0.219 G, z = -0.954 G
x = -0.169 G, y = 0.227 G, z = -0.906 G
```

accelerometer numbers



Keep in mind …

- Code lives in **code.py**
- Open Serial to read from your microcontroller
- Formatting matters!
- Make sure you have your libraries

play around a little

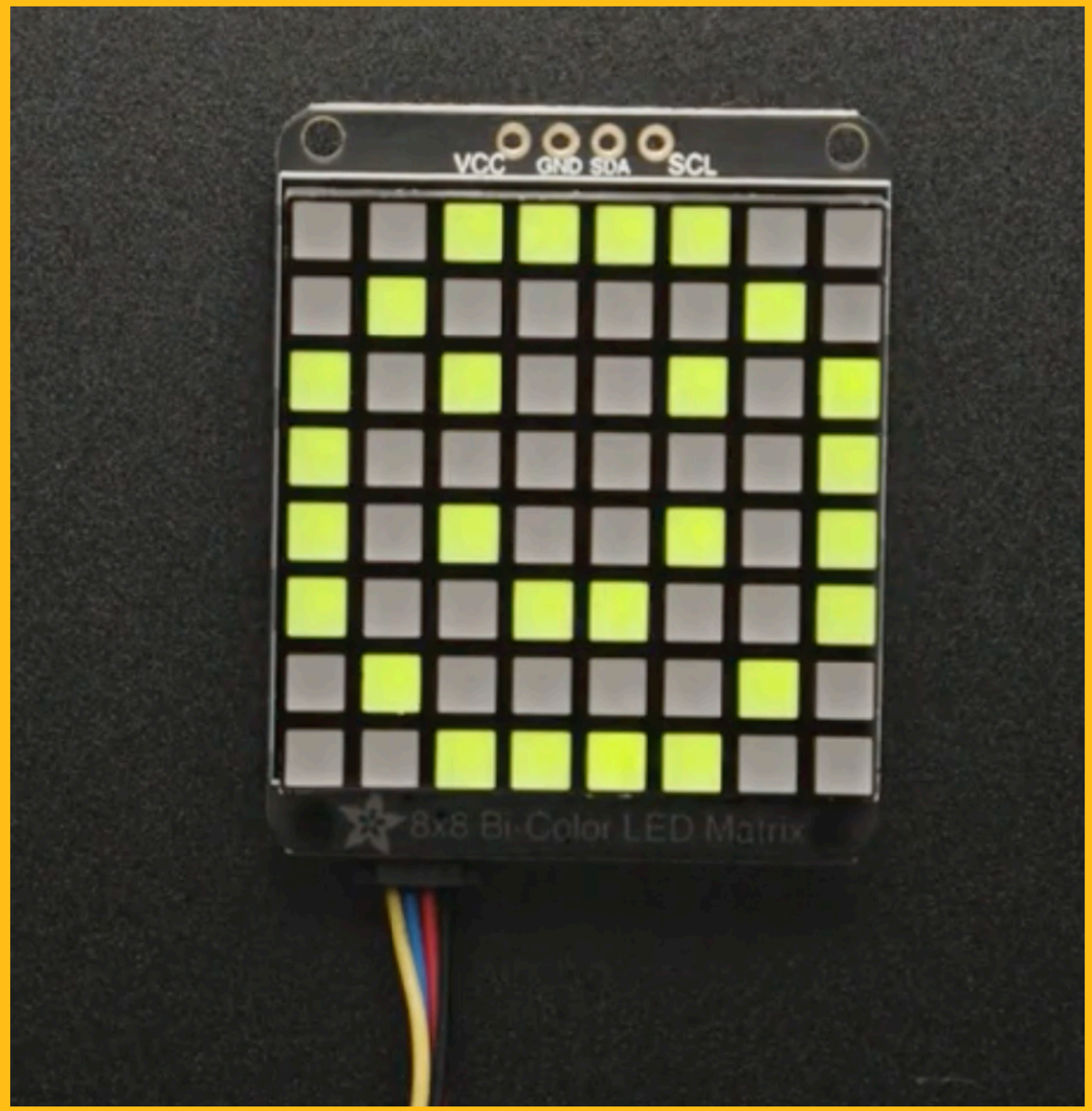
Play with the accelerometer, understand it more.

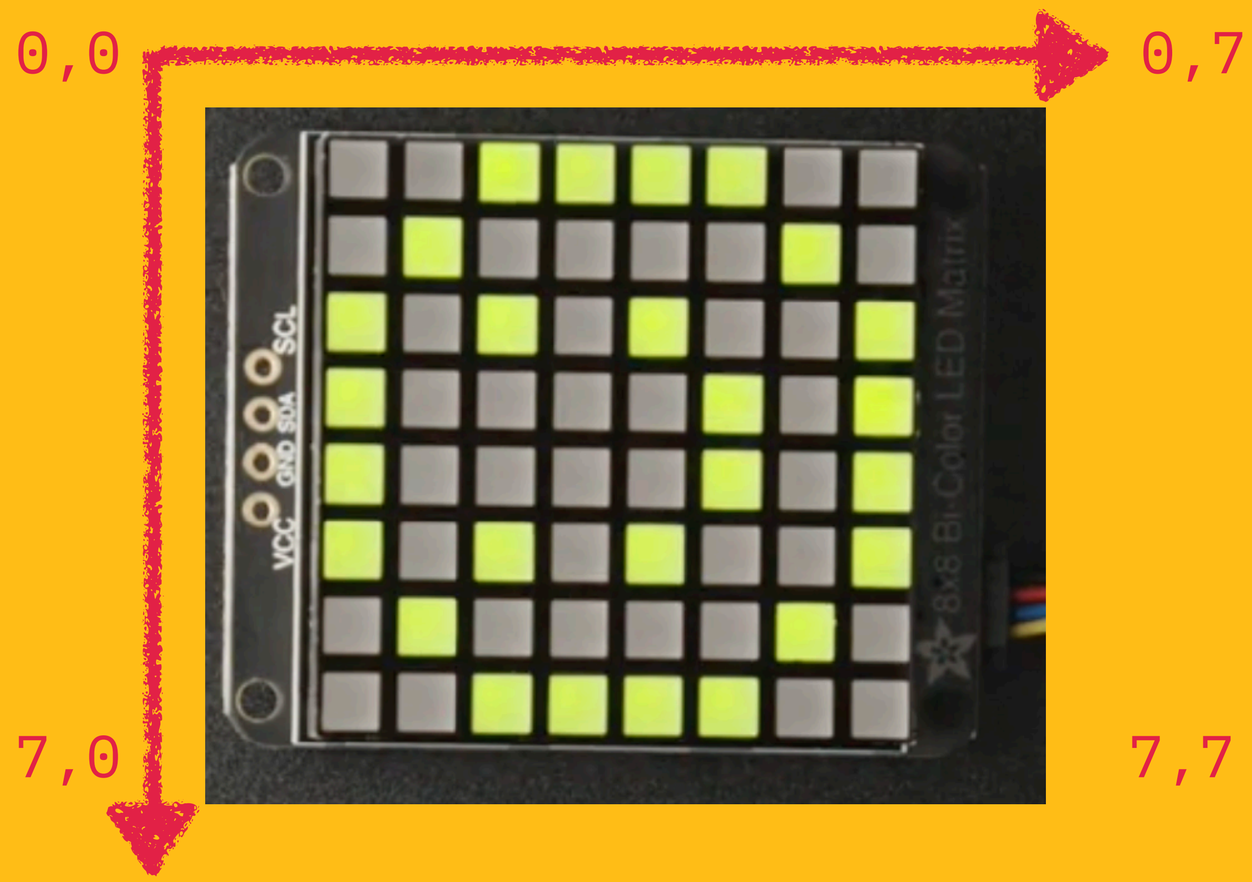
Some challenges:

- Can you control the Neopixel with the accelerometer?
- What about printing?

8x8 Bicolor LED Matrix

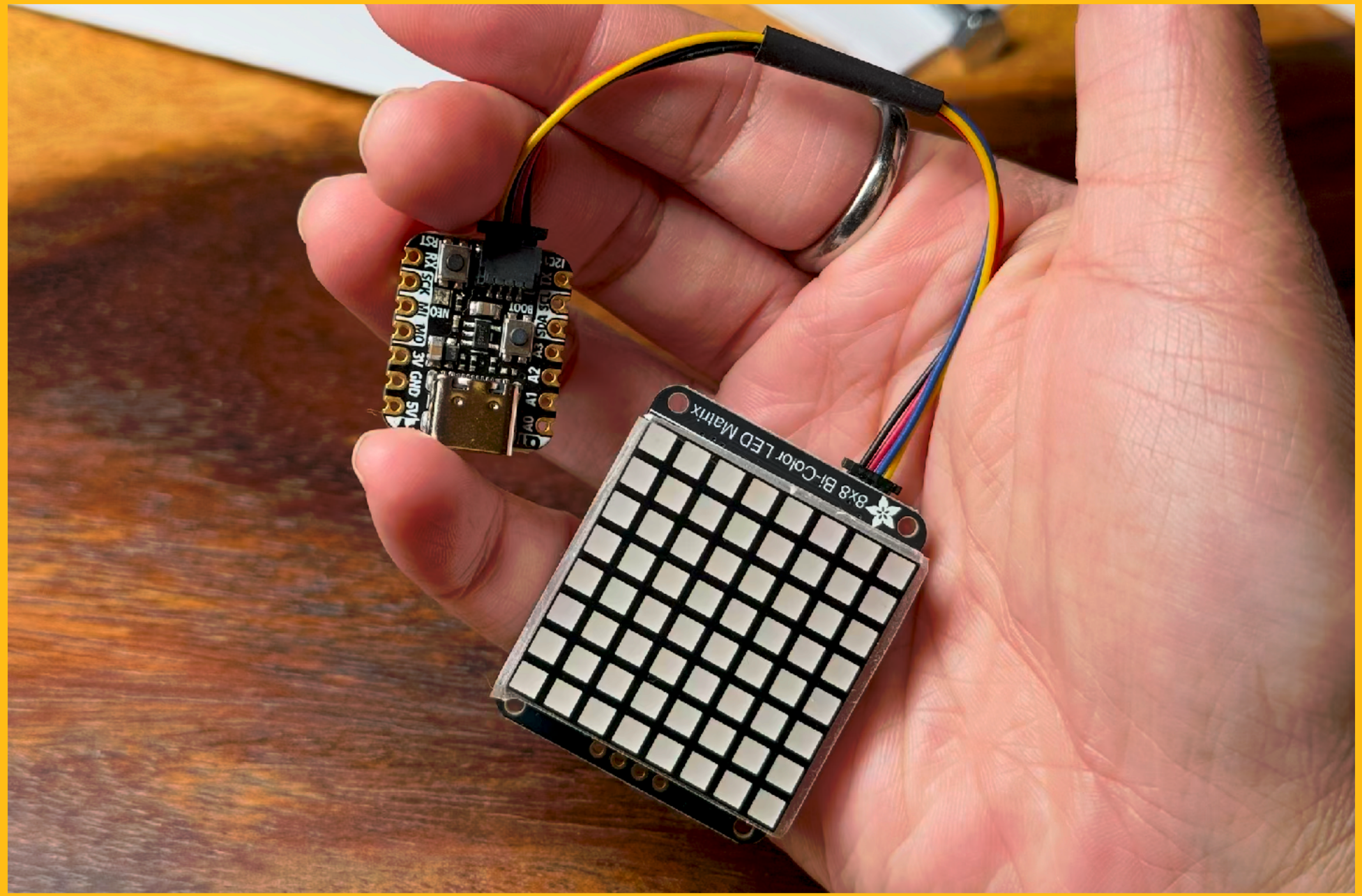
- 64 bi-color LEDs
- Can show red, green, and yellow
- Uses a coordinate system




0,0
0,7
7,0
7,7
VCC
GND
SDA
SCL
8x8 Bi-Color LED Matrix

Keep in mind …

- Code lives in **code.py**
- Open Serial to read from your microcontroller
- Formatting matters!
- Make sure you have your libraries
- Be careful with coordinates


8x8 Bi-Color LED Matrix

8x8 matrix code

adafruit_ht16k33 << full folder!

```
1  import board
2  import time
3
4  from adafruit_ht16k33.matrix import Matrix8x8x2
5
6  i2c = board.STEMMA_I2C()
7  matrix = Matrix8x8x2(i2c)
8
9  matrix[0,0] = matrix.LED_GREEN
10 matrix[7,7] = matrix.LED_RED
11 # matrix.fill(matrix.LED_YELLOW)
12
```

matrix.LED_RED

matrix.LED_GREEN

matrix.LED_YELLOW

matrix.LED_OFF

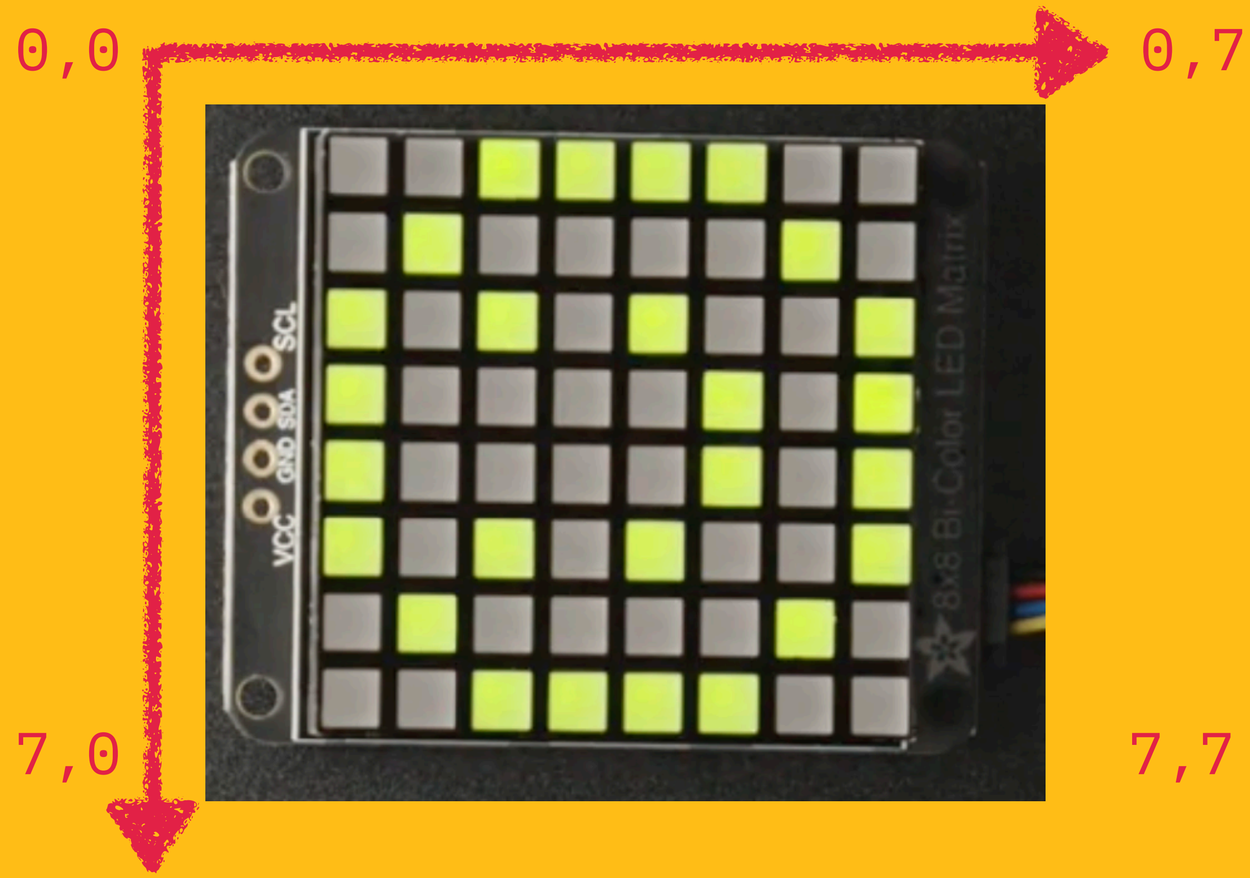
https://github.com/leils/workshop-resources/blob/main/micromovement_ga2025/example%20code/04%20matrix%20/code.py

8x8 matrix code

```
matrix.shift(1, 0, True)
```

(0, 0) -> (1, 0)

(7, 0) -> (0, 0)



adafruit_ht16k33 << full folder!

https://github.com/leils/workshop-resources/blob/a7381dc76700d1280ebf35cda3f6a18206a5cd68/micromovement_ga2025/example%20code/04%20matrix%20/code.py

8x8 matrix code

```
matrix.shift(1, 0, True)
```

```
(0, 0) -> (1, 0)
(7, 0) -> (0, 0)
```

Use with caution[03]:

```
matrix.shift_down(True)
matrix.shift_up(True)
matrix.shift_left(True)
matrix.shift_right(True)
```

adafruit_ht16k33 << full folder!

https://github.com/leils/workshop-resources/blob/a7381dc76700d1280ebf35cda3f6a18206a5cd68/micromovement_ga2025/example%20code/04%20matrix%20/code.py

play around a li

Just play with the Bicolor Matrix!

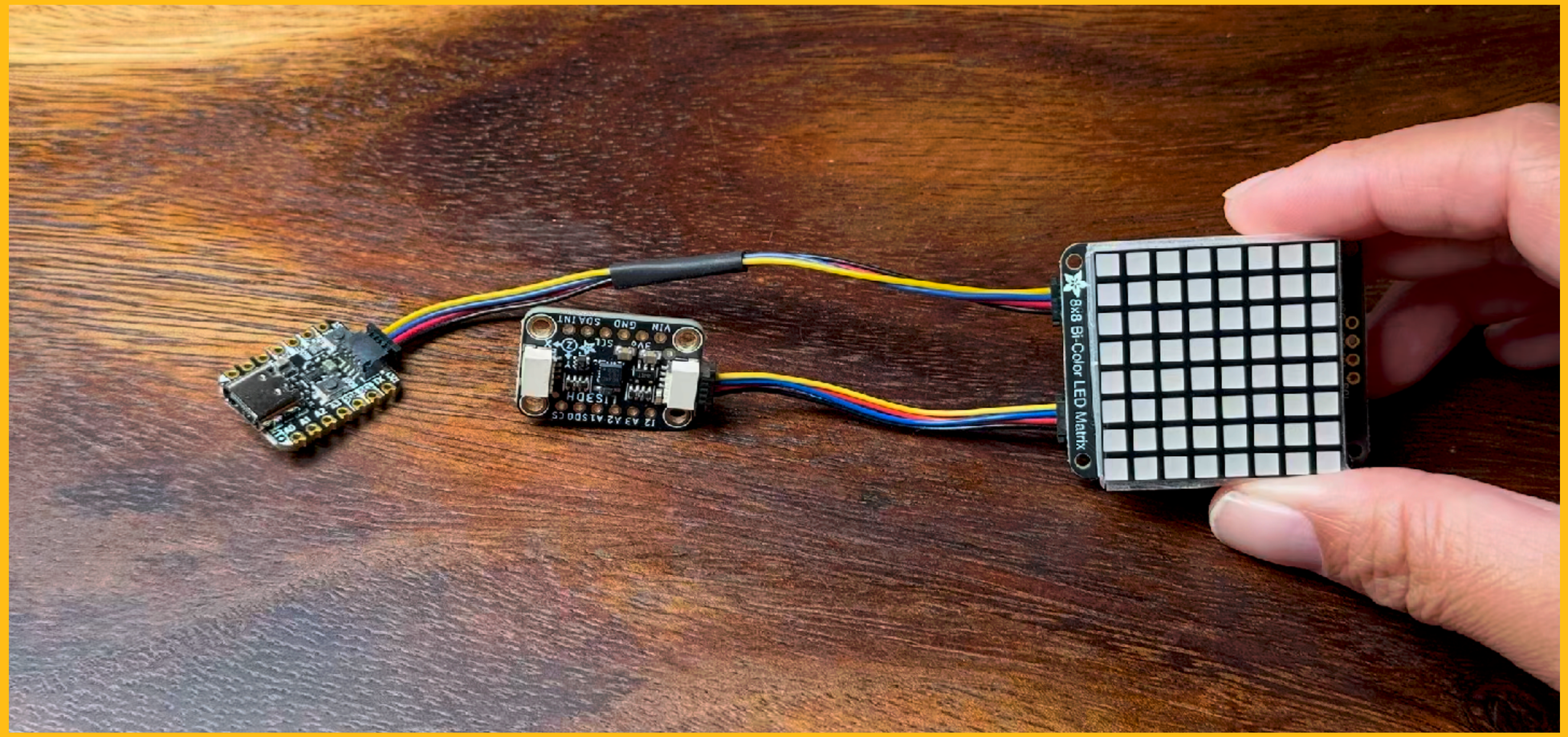
Can you:

- Make a line of color?
- A square? A rectangle?
- Move a single pixel around. What can you do with that?

Keep in mind …

- Code lives in **code.py**
- Open Serial to read from your microcontroller
- Formatting matters!
- Make sure you have your libraries
- Be careful with coordinates

let’s put it all together


8x8 Bi-Color LED Matrix

some of the code

```
import board
import adafruit_lis3dh
import time

from adafruit_ht16k33.matrix import Matrix8x8x2

i2c = board.STEMMA_I2C()
matrix = Matrix8x8x2(i2c)
lis3dh = adafruit_lis3dh.LIS3DH_I2C(i2c)

# Set range of accelerometer (can be RANGE_2_G, RANGE_4_G, RANGE_8_G or RANGE_16_G).
lis3dh.range = adafruit_lis3dh.RANGE_2_G

while True:
    x, y, z = (value / adafruit_lis3dh.STANDARD_GRAVITY for value in lis3dh.acceleration)

    # Put your code here!


    time.sleep(0.1)

```

https://github.com/leils/workshop-resources/blob/main/micromovement_ga2025/example%20code/05%20matrix%20and%20accelerometer%20exercise/code.py

play around a li

Make something with the accelerometer and the LED matrix!

Can you:

- Make an X appear when more than a certain force is sensed?
- Show a light only if the object is moving?
- Make a dot “jump” if you tap it?

Keep in mind …

- Code lives in **code.py**
- Open Serial to read from your microcontroller
- Formatting matters!
- Make sure you have your libraries
- Be careful with coordinates

have fun!
@leia.make

01: all the hardware for this class was sourced from Adafruit.

02: eagle eyed viewers may notice that mu has been archived and is no longer maintained. unfortunately for us, this happened two weeks before the workshop! it's still the suggested editor from Adafruit, and this will work for now, but know that you can also edit your code with any other text or code editor. code.circuitpython.org will also work.

03: i find using shift_down, shift_up, etc to be unintuitive and limiting. don't let the library's definition of "down" limit what you imagine!

You can find all of the code, and other resources here: https://github.com/leils/workshop-resources/tree/main/micromovement_ga2025