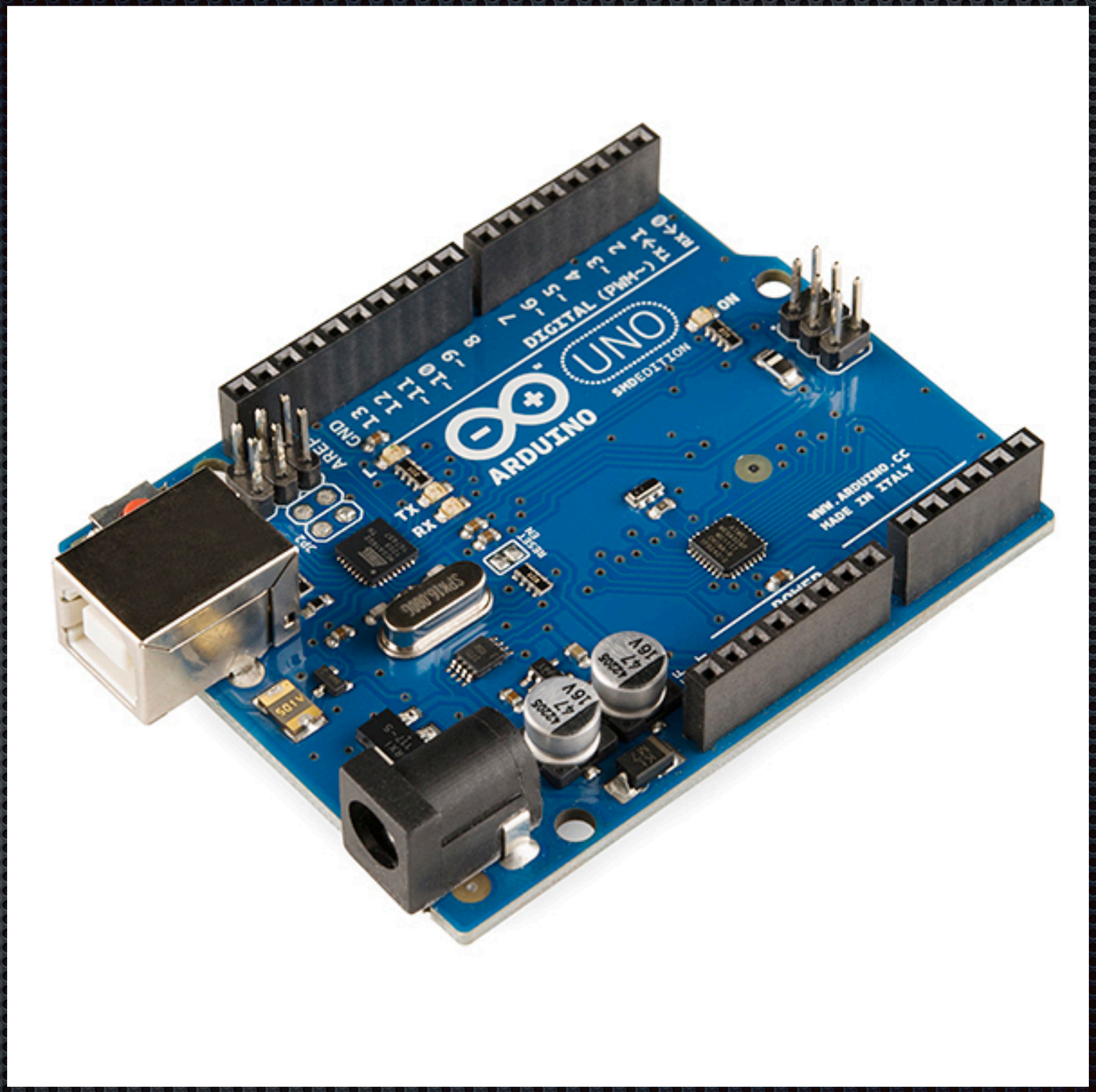
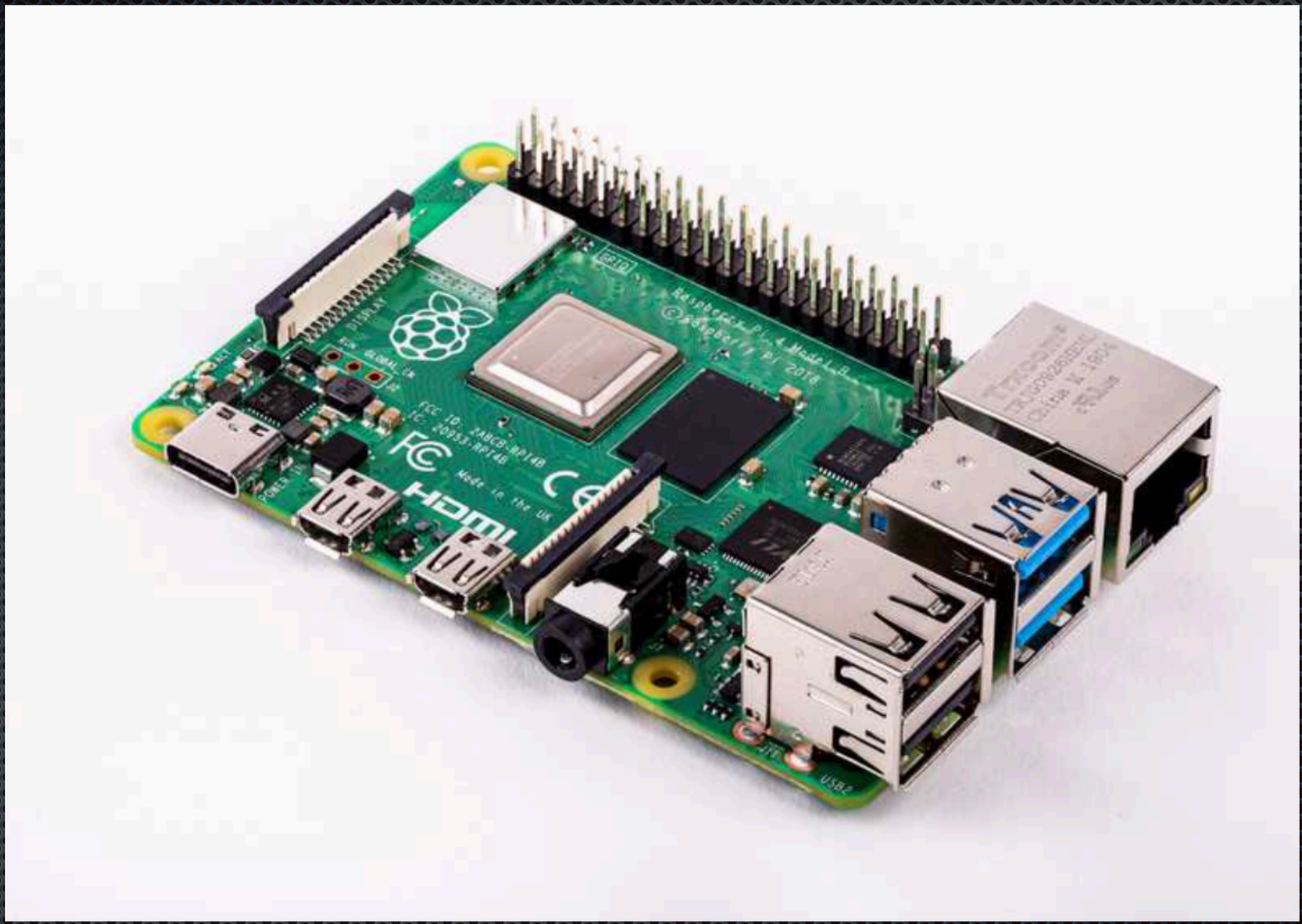
wearable python

felice ho
october 2019

Deloitte.
Digital

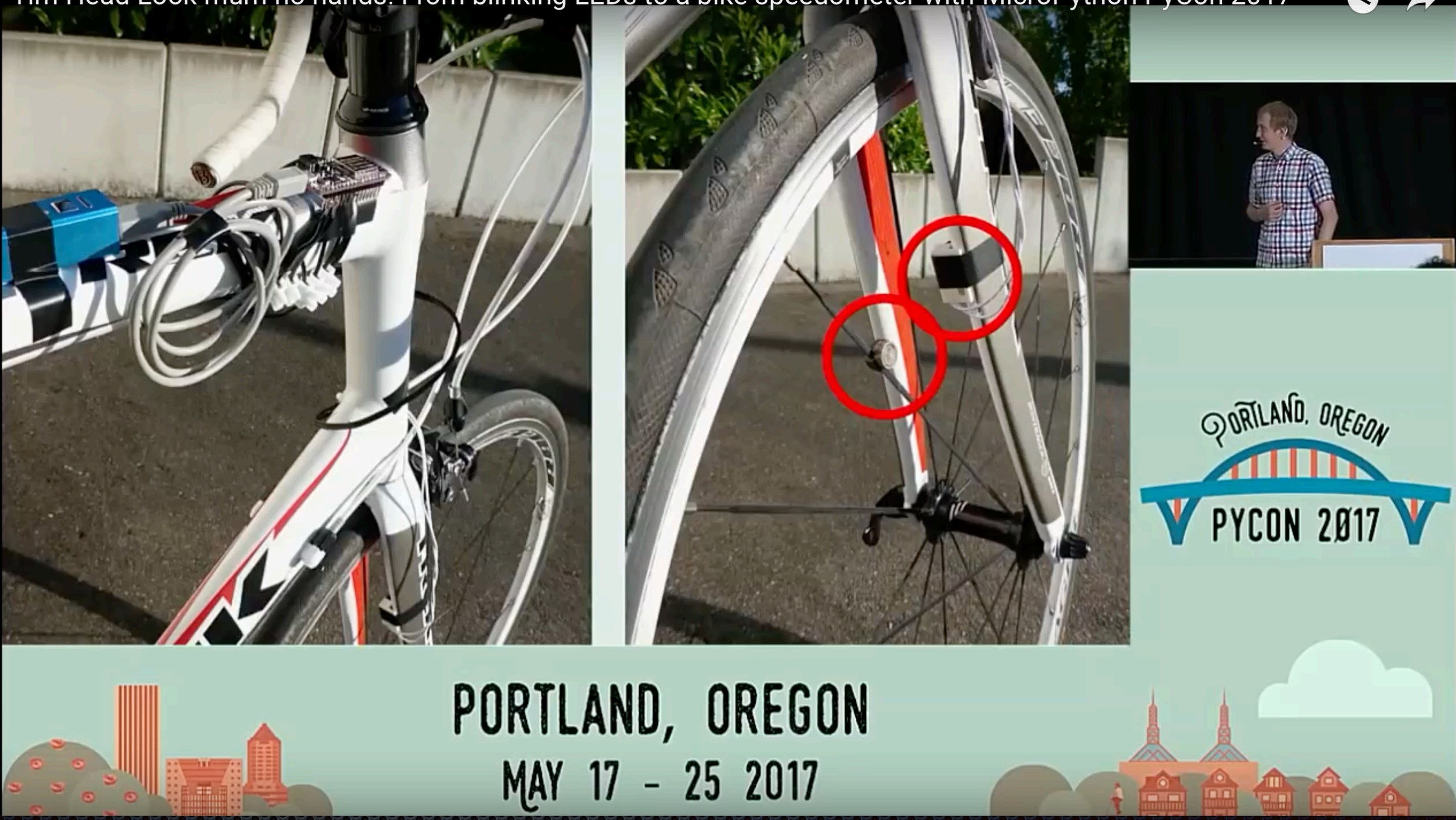


pictured: Arduino Uno

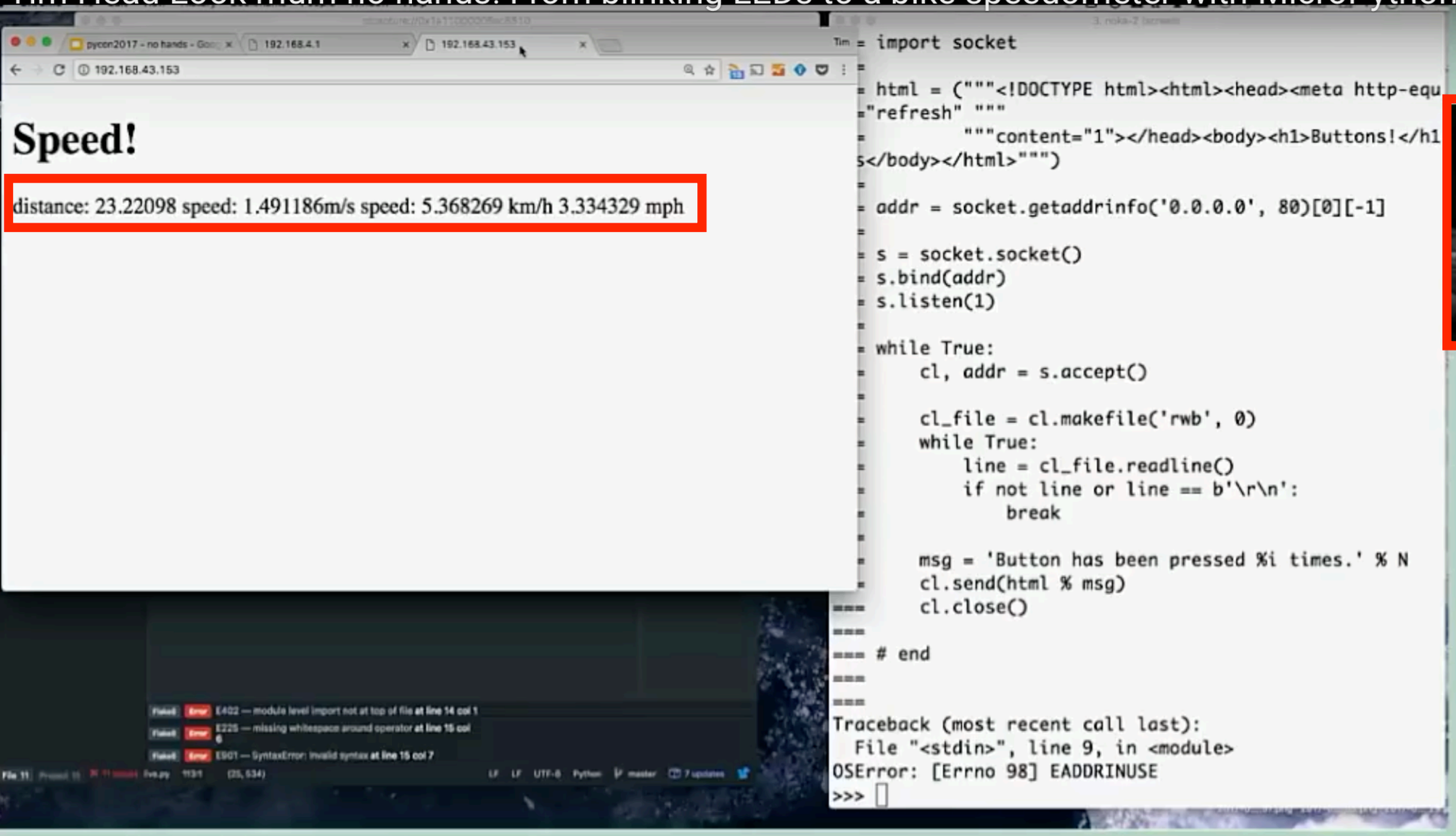


pictured: Raspberry Pi 4

Tim Head Look mum no hands! From blinking LEDs to a bike speedometer with MicroPython PyCon 2017

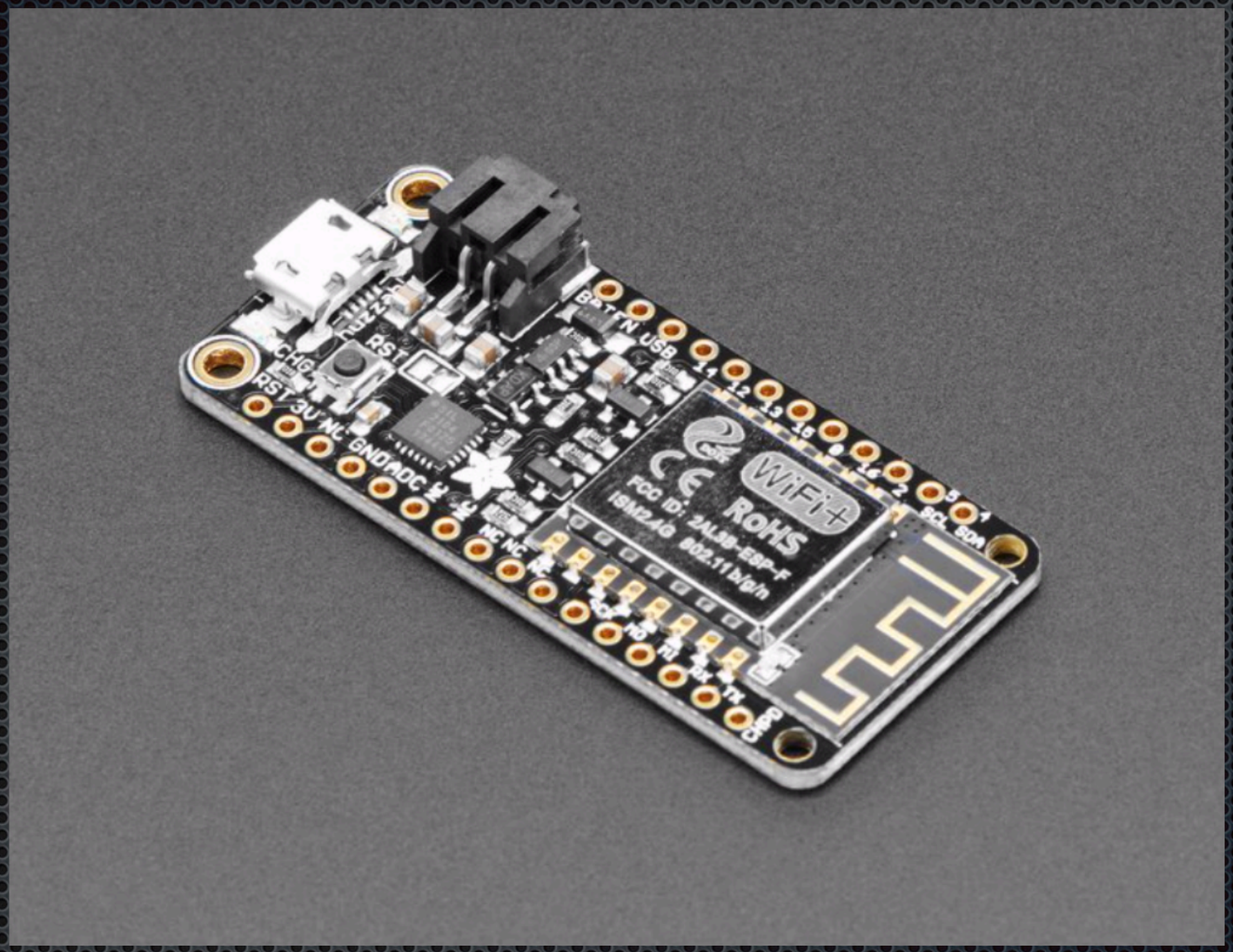


Tim Head Look mum no hands! From blinking LEDs to a bike speedometer with MicroPython PyCon 2017



PORTLAND, OREGON
PYCON 2017

PORTLAND, OREGON
MAY 17 - 25 2017

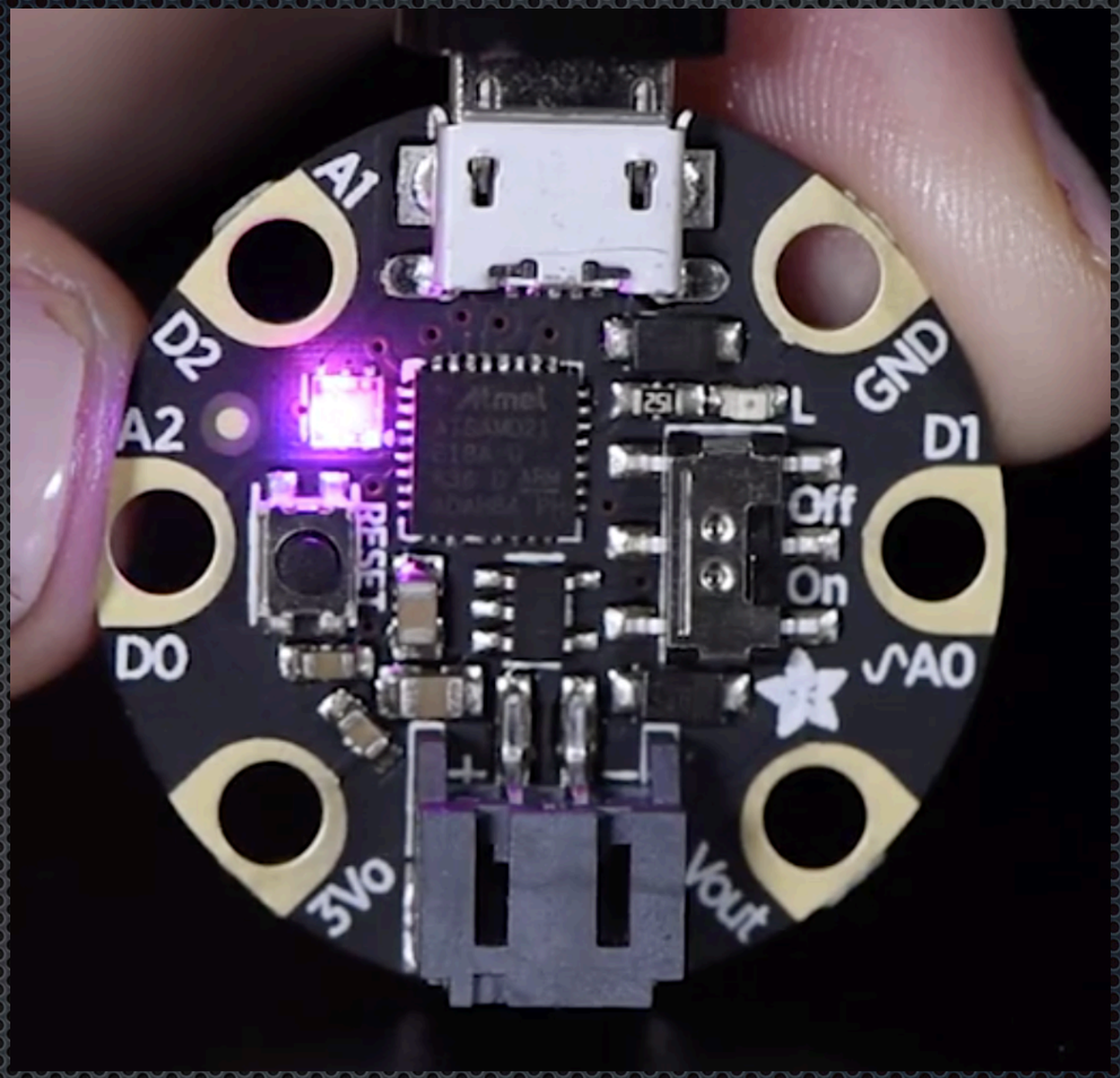


pictured: Adafruit Feather HUZZAH with ESP8266

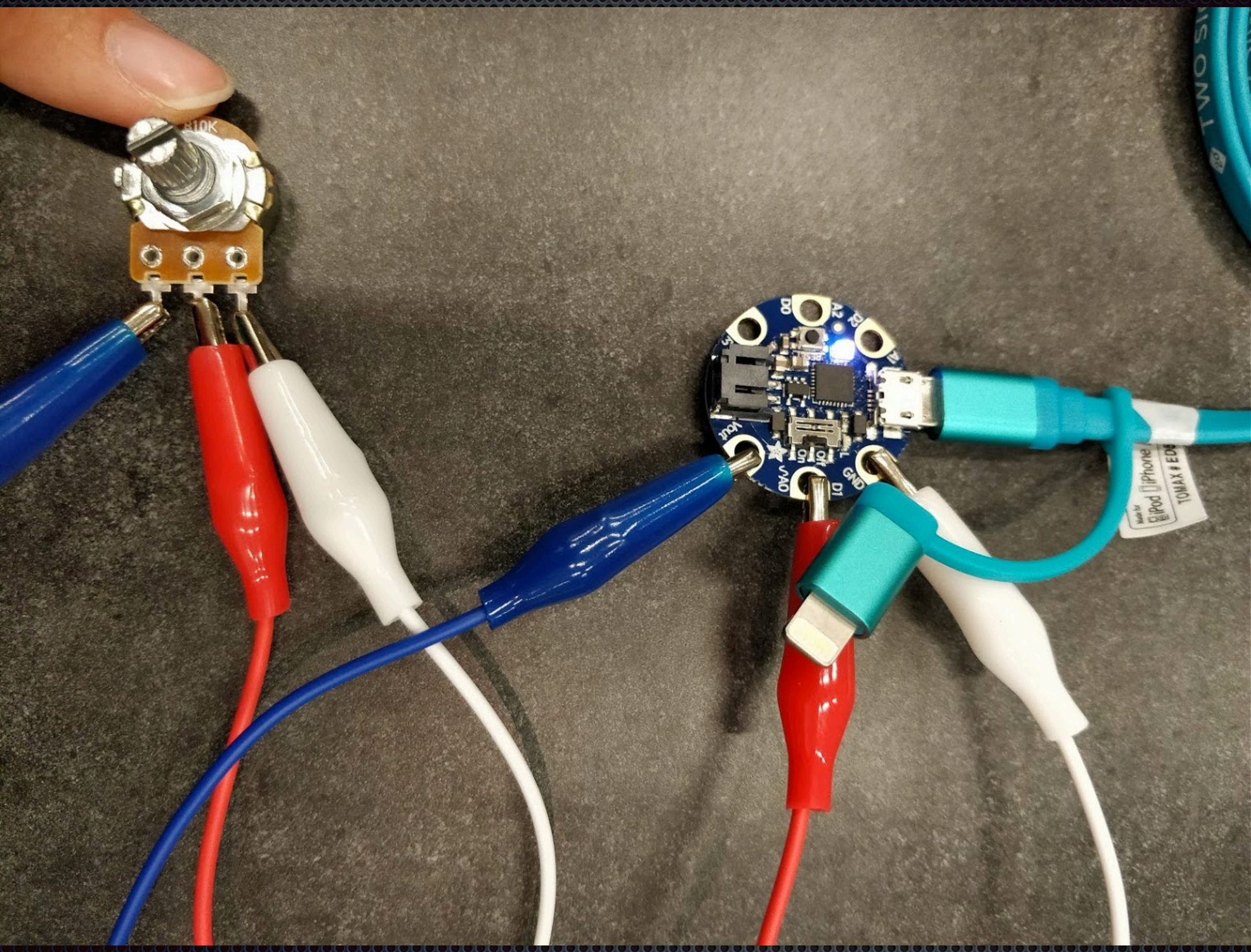
MicroPython

MicroPython is a lean and efficient implementation of the Python 3 programming language that includes a small subset of the Python standard library and is optimised to run on microcontrollers and in constrained environments.

source: http://micropython.org



pictured: Adafruit GEMMA M0


B10K
D0
A2
B
A1
D5
Vout
D2
D3
GND
D1
GND
TWO SI
iPod iPhone
TOMAX # ED

pictured: Limor "Ladyada" Fried

ABOUT US

Adafruit was founded in 2005 by MIT hacker & engineer, Limor "Ladyada" Fried. **Her goal was to create the best place online for learning electronics and making the best designed products for makers of all ages and skill levels.** Adafruit has grown to over 100+ employees in the heart of NYC with a 50,000+ sq ft. factory. Adafruit has expanded offerings to include tools, equipment, and electronics that Limor personally selects, tests, and approves before going in to the Adafruit store. Adafruit is a 100% woman owned company.

source: https://www.adafruit.com/about

GEMMA Hoop Earrings

NeoPixel Ring Bling

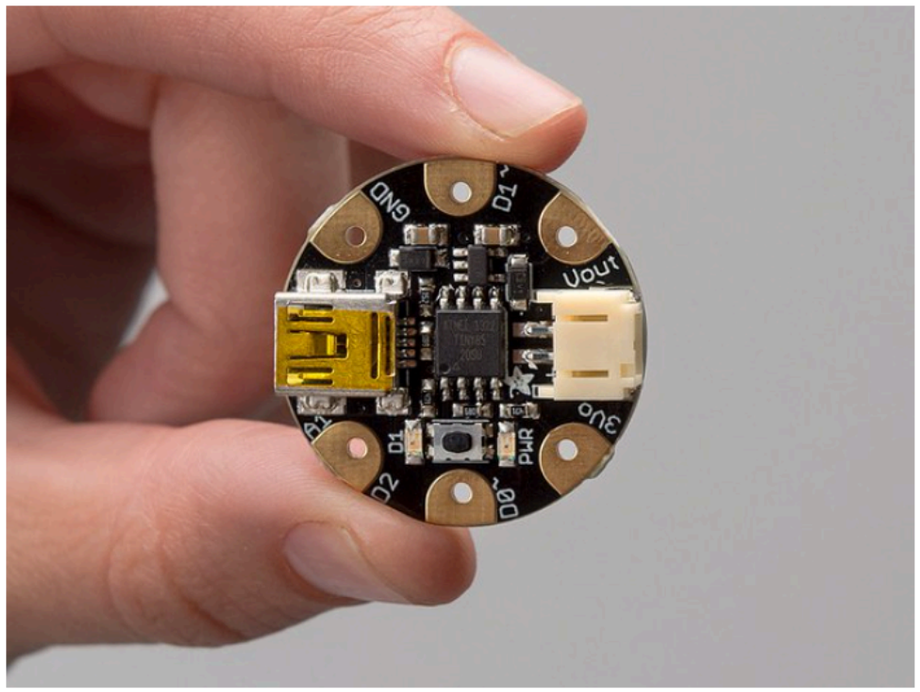
Overview

Tools & Supplies

LEDS / LED PIXELS | ARDUINO COMPATIBLES / ADAFRUIT GEMMA | WEARABLES

Tools & Supplies

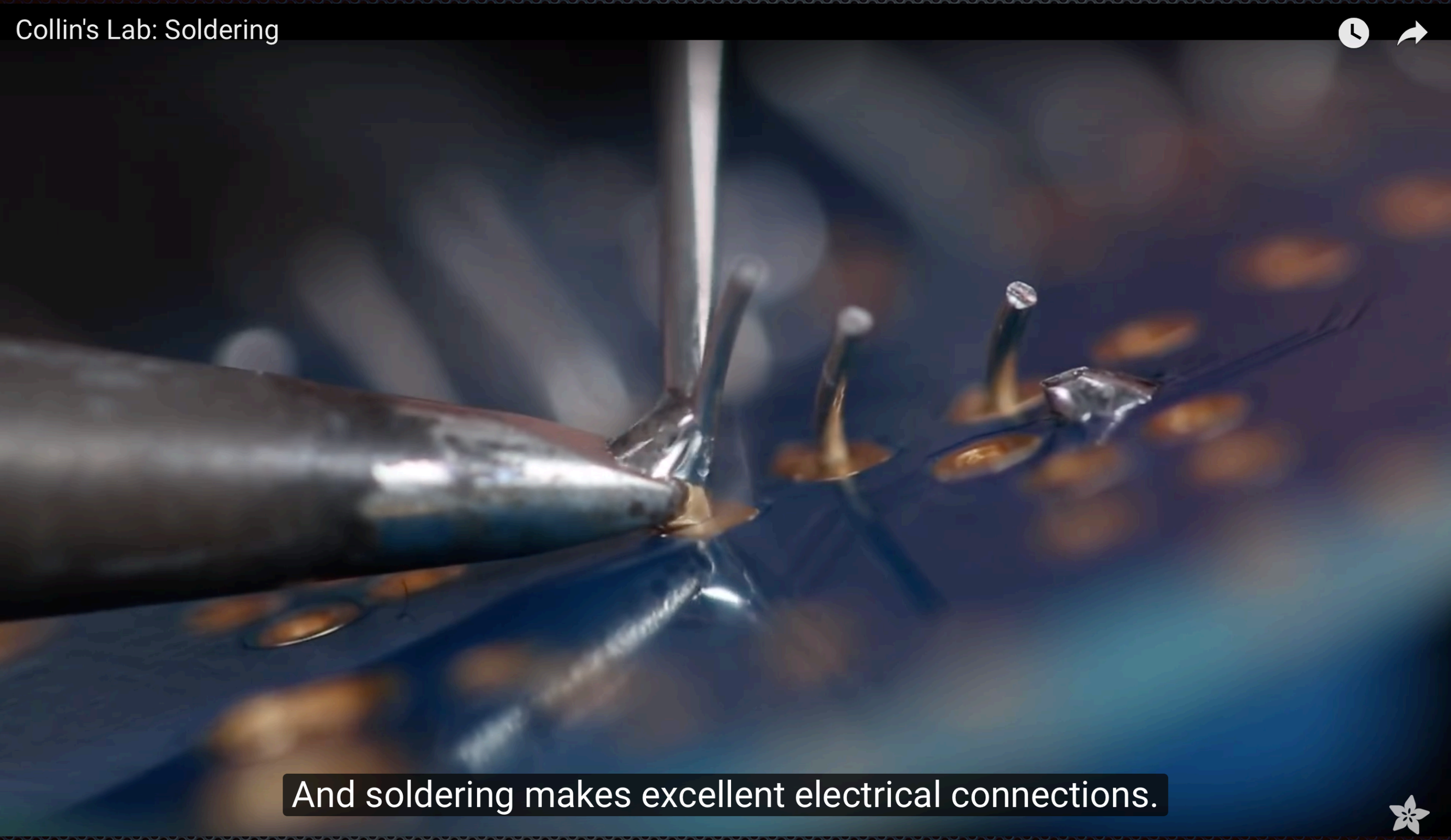
by Becky Stern



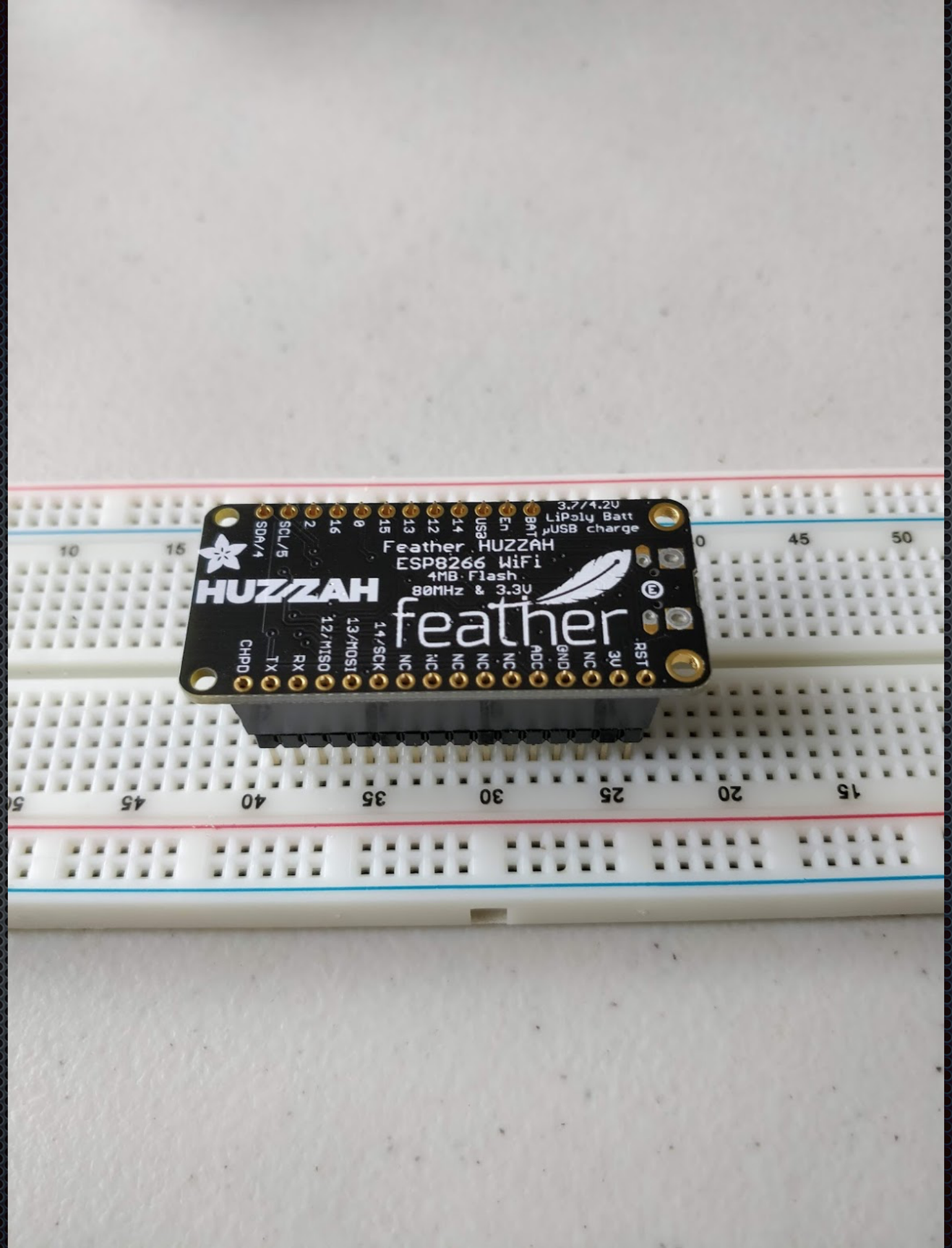
Bill of materials (for one pendant-- double this for a pair of earrings):

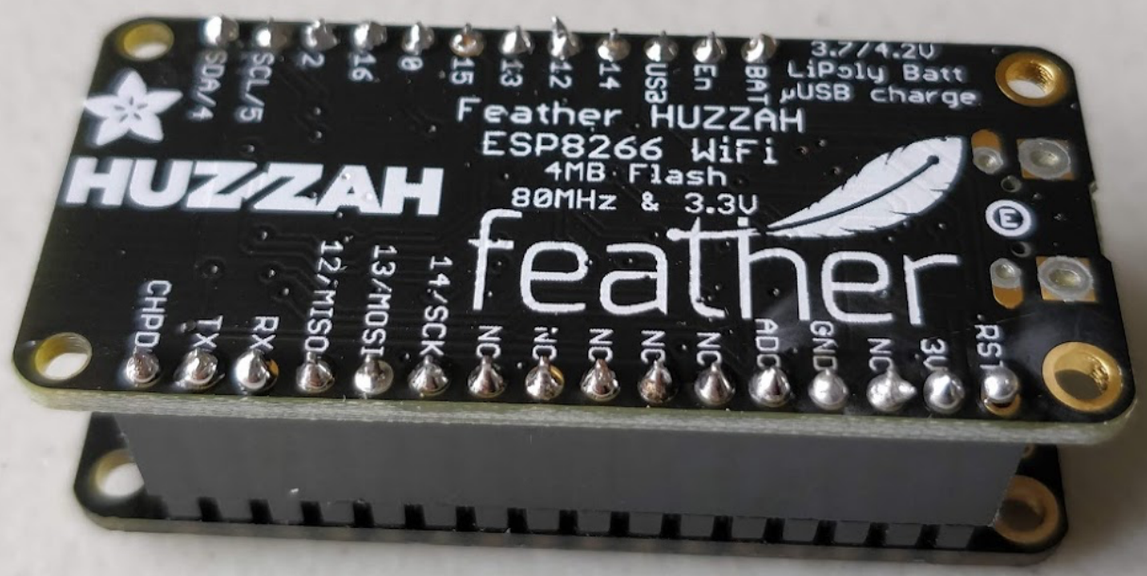
- Gemma M0 or Original Gemma (M0 type is recommended!)
- NeoPixel ring
- tiny lipoly battery with charger
- pendant hanger or ear wires/jump rings
- stranded wire
- clear thread
- double stick tape
- E6000 adhesive (if using glue-on pendant hanger)

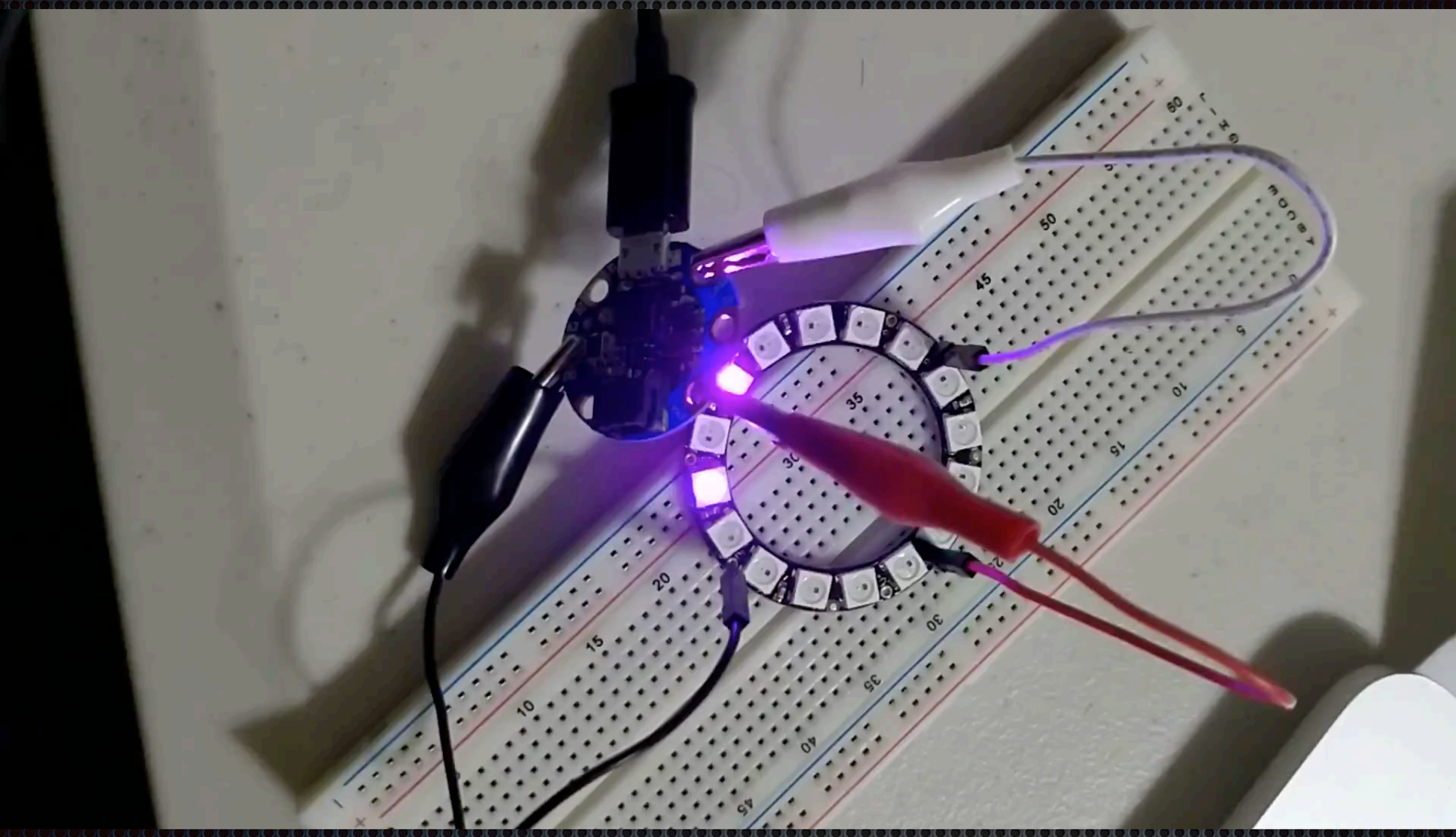
source: https://learn.adafruit.com/gemma-hoop-earrings

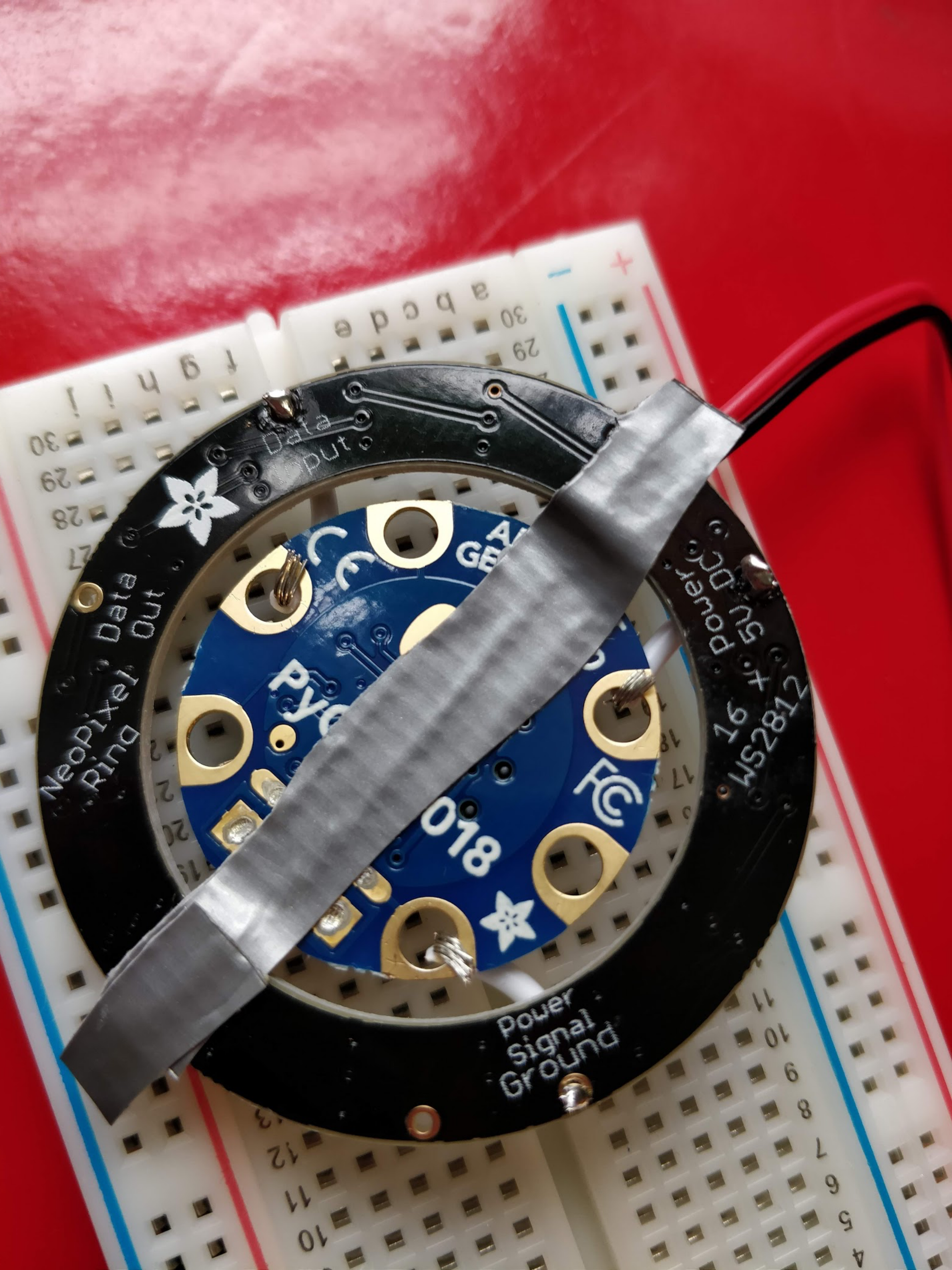


source: https://learn.adafruit.com/collins-lab-soldering

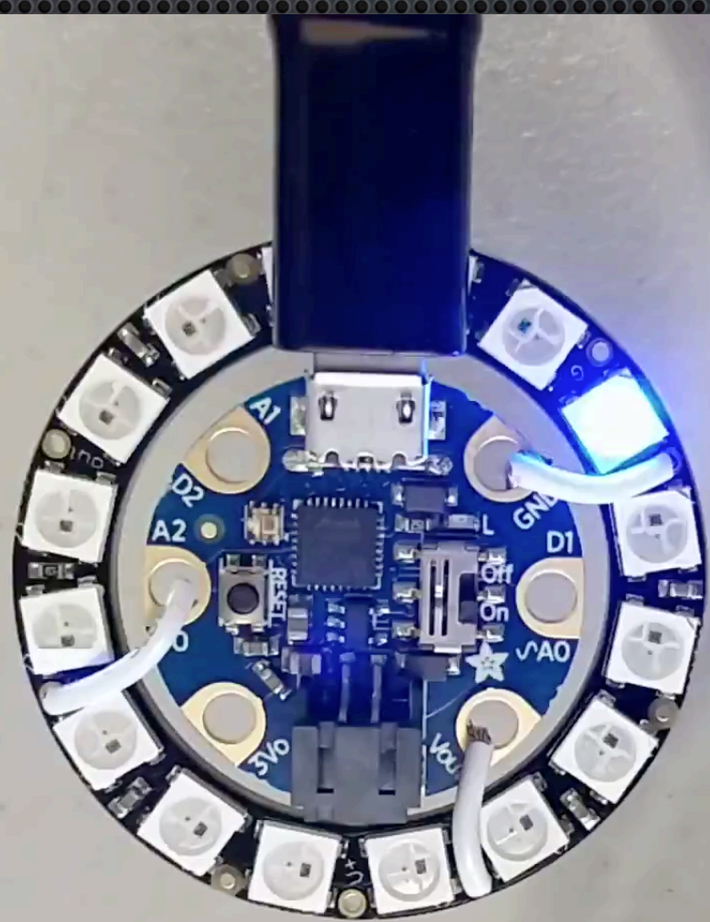

3.7/4.2V
LiPoly Batt
µUSB charge
SDA/4
SCL/5
2
16
0
15
13
12
14
USB
EN
BAT
Feather HUZZAH
ESP8266 WiFi
4MB Flash
80MHz & 3.3V
HUZZAH
feather
CHPD
TX
RX
12/MISO
13/MOSI
14/SCK
NC
NC
NC
NC
NC
ADC
GND
NC
3V
RST

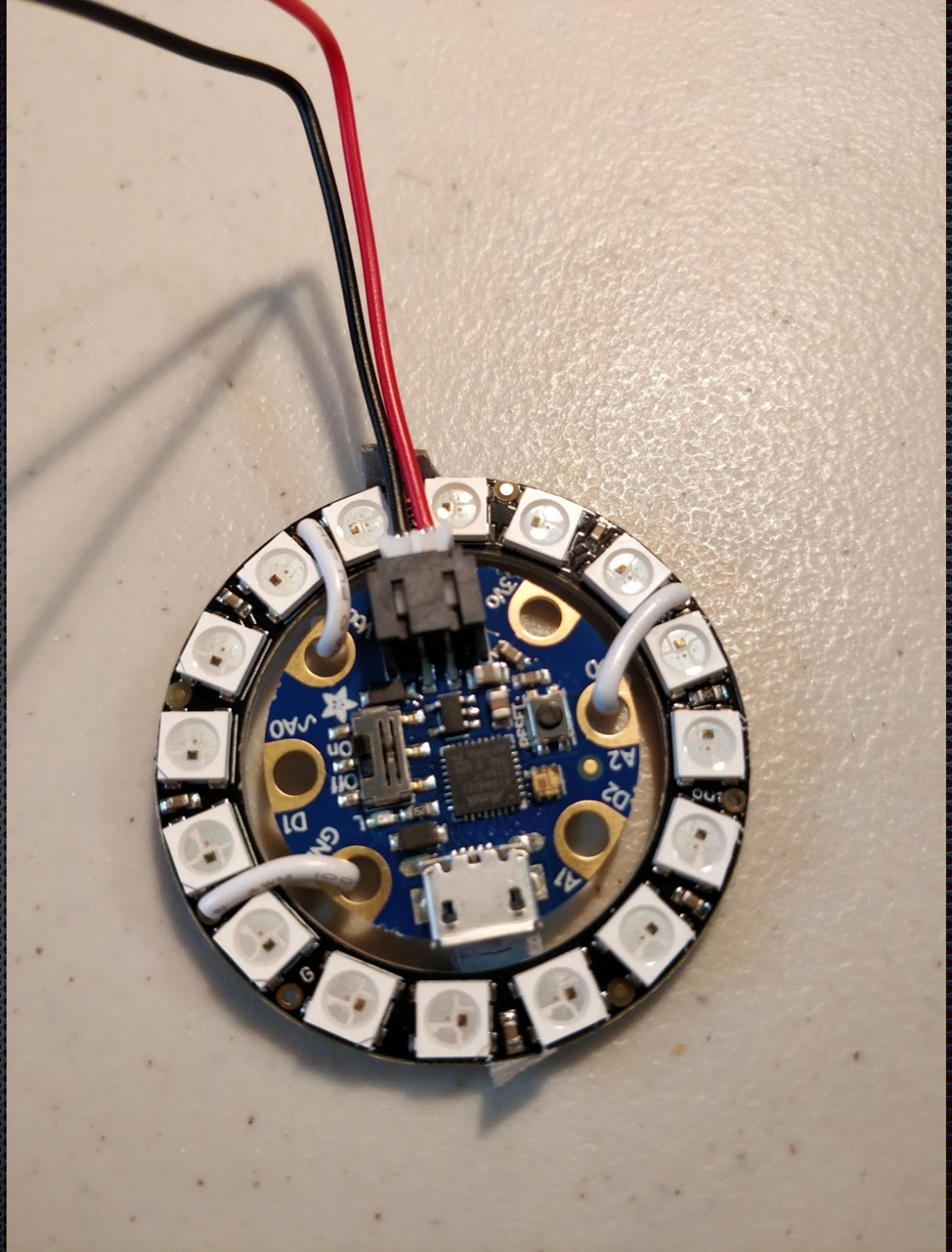

3.7/4.2V
LiPoly Batt
µUSB charge
Feather HUZZAH
ESP8266 WiFi
4MB Flash
80MHz & 3.3V
HUZZAH
feather
SDA/4
SCL/5
2
16
0
15
13
12
14
USB
En
BAT
CHPD
TX
RX
12/MISO
13/MOSI
14/SCK
NC
NC
NC
NC
NC
ADC
GND
NC
3V
RST




NeoPixel Ring
Data Out
Data Input
16 x 5V Power
WS2812
018
Power
Signal
Ground

ADAFRUIT
GEMMA M0
PyCon 2018
Data
Input
Data
Out
NeoPixel
Ring
Power
Signal
Ground
16 x Power
5V DC
WS2812


A1
D2
A2
GND
D1
L
Off
On
RESET
A0
3Vo


SAO
D1
A2
D2
A1

1. bash

```
Last login: Wed Oct  9 10:16:35 on ttys000
Felice's MacBook Pro:~ FeliceHo$ screen /dev/tty.usbmodem14201
```

Hardware requirements

Beside of a micropython board itself, a **MicroUSB cable** (USB A to USB micro B) is required.

Please be aware that not all MicroUSB cables on the market are suitable. Esp. cheap cables to "charge smartphones" might only carry out the power lines but not the data lines of the USB connection. Thus, those cables will nicely power the micropython board (e.g. from a battery pack) but there will be no data connection.

If there seems to be no data connection at all, try another cable! Most likely cables which came with smartphones or other devices which can be connected to computers will be ok.

1. bash

```
Last login: Wed Oct  9 10:16:35 on ttys000
Felice's MacBook Pro:~ FeliceHo$ screen /dev/tty.usbmodem14201
```

CIRCUITPY

Search

Applications

iCloud

iCloud Drive

Locations

Felice's MacBook Pro

CIRCUITPY

Remote Disc

Tags

Name	Date Modified
boot_out.txt	Sep 1, 2016 at 4:43 PM
lib	Mar 27, 2018 at 4:51 PM
main.py	Sep 5, 2019 at 11:09 PM
PYCON 2018 README.txt	Mar 27, 2018 at 4:52 PM
Windows 7 Driver	Mar 27, 2018 at 4:51 PM

CIRCUITPYTHON | PROGRAMMING / MICROPYTHON / CIRCUITPYTHON

What is CircuitPython?

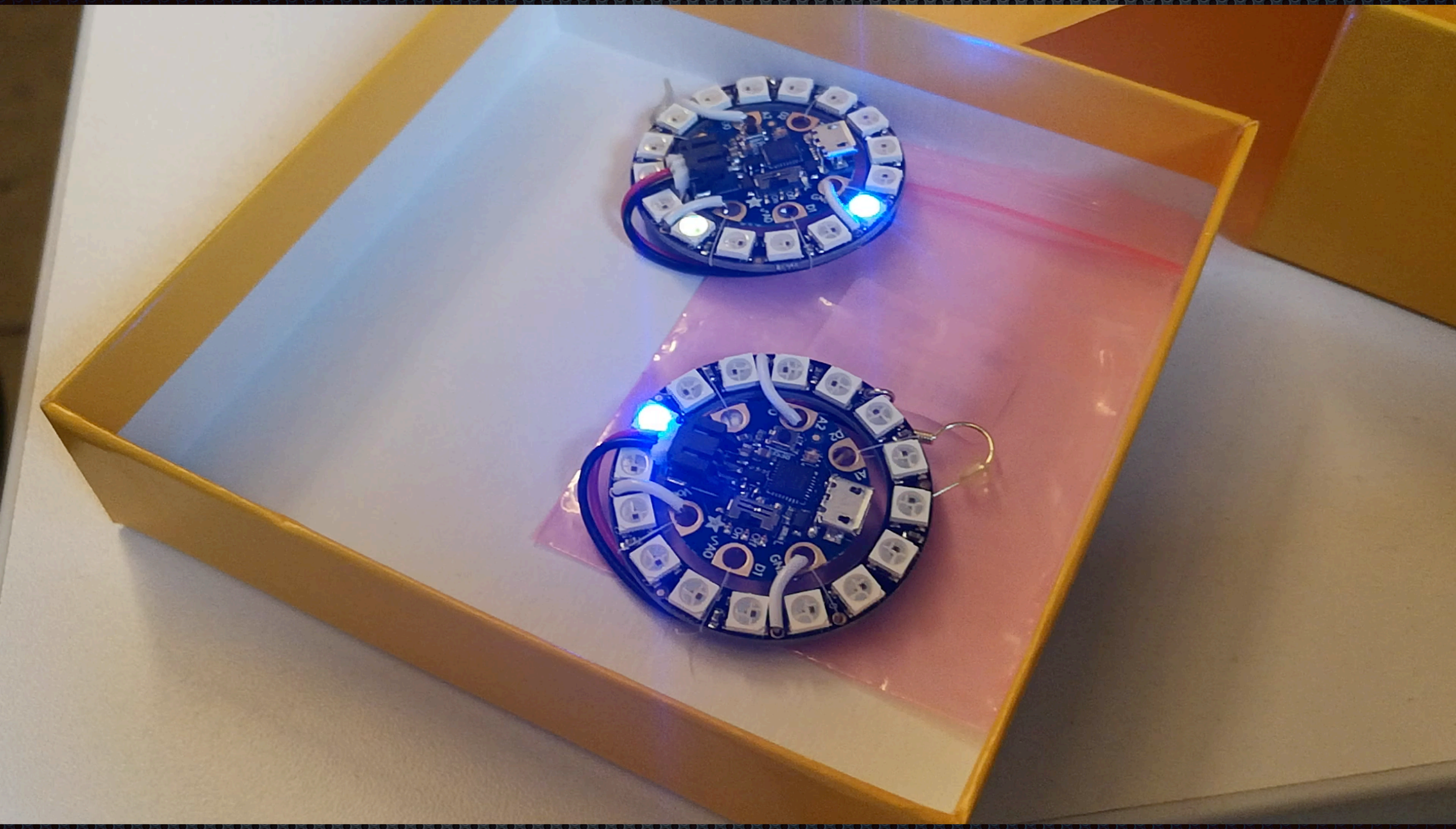
by Kattni Rembor

CircuitPython is a programming language designed to simplify experimenting and learning to program on low-cost microcontroller boards. It makes getting started easier than ever with no upfront desktop downloads needed. Once you get your board set up, open any text editor, and get started editing code. It's that simple.

source: https://learn.adafruit.com/welcome-to-circuitpython/what-is-circuitpython

```python

# NeoPixel earrings example.  Makes a nice blinky display with just a
# few LEDs on at any time...uses MUCH less juice than rainbow display!

import time

import board
import neopixel
import adafruit_dotstar
import microcontroller

try:
    import urandom as random  # for v1.0 API support
except ImportError:
    import random

dot = adafruit_dotstar.DotStar(board.APA102_SCK, board.APA102_MOSI, 1, bright
dot[0] = (0, 0, 0)

numpix = 16  # Number of NeoPixels (e.g. 16-pixel ring)
pixpin = board.D0  # Pin where NeoPixels are connected
# strip = neopixel.NeoPixel(pixpin, numpix, brightness=.3, auto_write=False)
strip = neopixel.NeoPixel(pixpin, numpix, brightness=.3, auto_write=False)

```

```
60  ##########################################
61  # for pycon
62  show_blue = True
63  blue_hue = 160
64  yellow_hue = 50
65  color_blue = wheel(blue_hue & 255)
66  color_yellow = wheel(yellow_hue & 255)
67  ##########################################
68
69  while True:  # Loop forever...
70      if mode in (0, 2, 4):  # Random sparkles - lights just one LED at a time
71          i = random.randint(0, numpix - 1)  # Choose random pixel
72          ##################
73          # for pycon
74          if show_blue:
75              color = color_blue
76              show_blue = False
77          else:
78              color = color_yellow
79              show_blue = True
80          ##################
81          strip[i] = color   # Set it to current color
82          strip.show()       # Refresh LED states
83          # Set same pixel to "off" color now but DON'T refresh...
84          # it stays on for now...bot this and the next random
85          # pixel will be refreshed on the next pass.
86          strip[i] = [0, 0, 0]
87          time.sleep(0.008)  # 8 millisecond delay
```

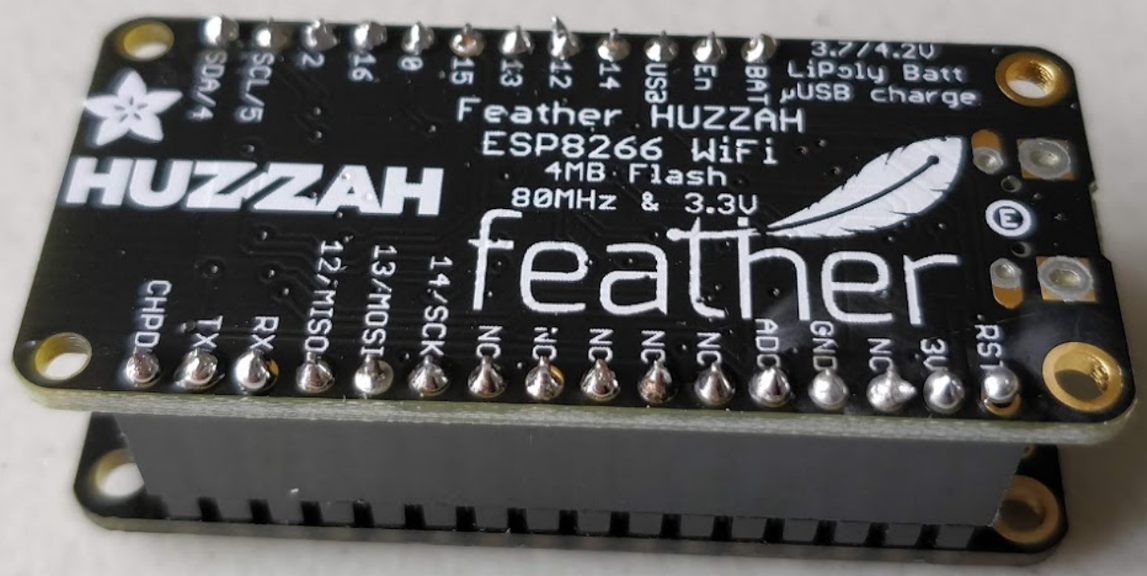


PYCON
CLEVELAND 2019
Guido van Rossum
Emeritus BDFL
Steering Council
membeer

CITADEL
CITADEL | Securities
SOFTWARE ENGINEERS
the world's mos

of Cleveland
and
Global Center
for Health Innovation
Global Center
8:22 pm
WELCOME WELCOME WELCO
22
ME

Deloitte
Digital


3.7/4.2V
LiPoly Batt
µUSB charge
Feather HUZZAH
ESP8266 WiFi
4MB Flash
80MHz & 3.3V
HUZZAH
feather
SDA/4
SCL/5
2
16
0
15
13
12
14
USB
En
BAT
CHPD
TX
RX
12/MISO
13/MOSI
14/SCK
NC
NC
NC
NC
NC
ADC
GND
NC
3V
RST

pictured: NeoPixel FeatherWing - 4x8 RGB LED

Re: Feather HUZZAH ESP8266 and NeoPixel FeatherWing

by adafruit_support_rick on Fri Feb 05, 2016 9:46 am

You need to cut the pin 16 jumper trace on the back of the wing and solder closed the pin 15 jumper.

source: https://forums.adafruit.com/viewtopic.php?f=47&t=87960

1. Getting started with MicroPython on the ESP8266

Using MicroPython is a great way to get the most of your ESP8266 board. And vice versa, the ESP8266 chip is a great platform for using MicroPython. This tutorial will guide you through setting up MicroPython, getting a prompt, using WebREPL, connecting to the network and communicating with the Internet, using the hardware peripherals, and controlling some external components.

Let's get started!

1.1. Requirements

The first thing you need is a board with an ESP8266 chip. The MicroPython software supports the ESP8266 chip itself and any board should work. The main characteristic of a board is how much flash it has, how the GPIO pins are connected to the outside world, and whether it includes a built-in USB-serial convertor to make the UART available to your PC.

source: https://docs.micropython.org/en/latest/esp8266/tutorial/intro.html

Firmware for ESP8266 boards

The following files are stable firmware for the ESP8266. Program your board using the esptool.py program as described in the tutorial.

- esp8266-20190529-v1.11.bin (elf, map) (latest)

```
pip install esptool
```

Versions starting with 1.3 support both Python 2.7 and Python 3.4 (or newer). An older version (at least 1.2.1 is needed) works fine but will require Python 2.7.

Any other flashing program should work, so feel free to try them out or refer to the documentation for your board to see its recommendations.

Using esptool.py you can erase the flash with the command:

```
esptool.py --port /dev/ttyUSB0 erase_flash
```

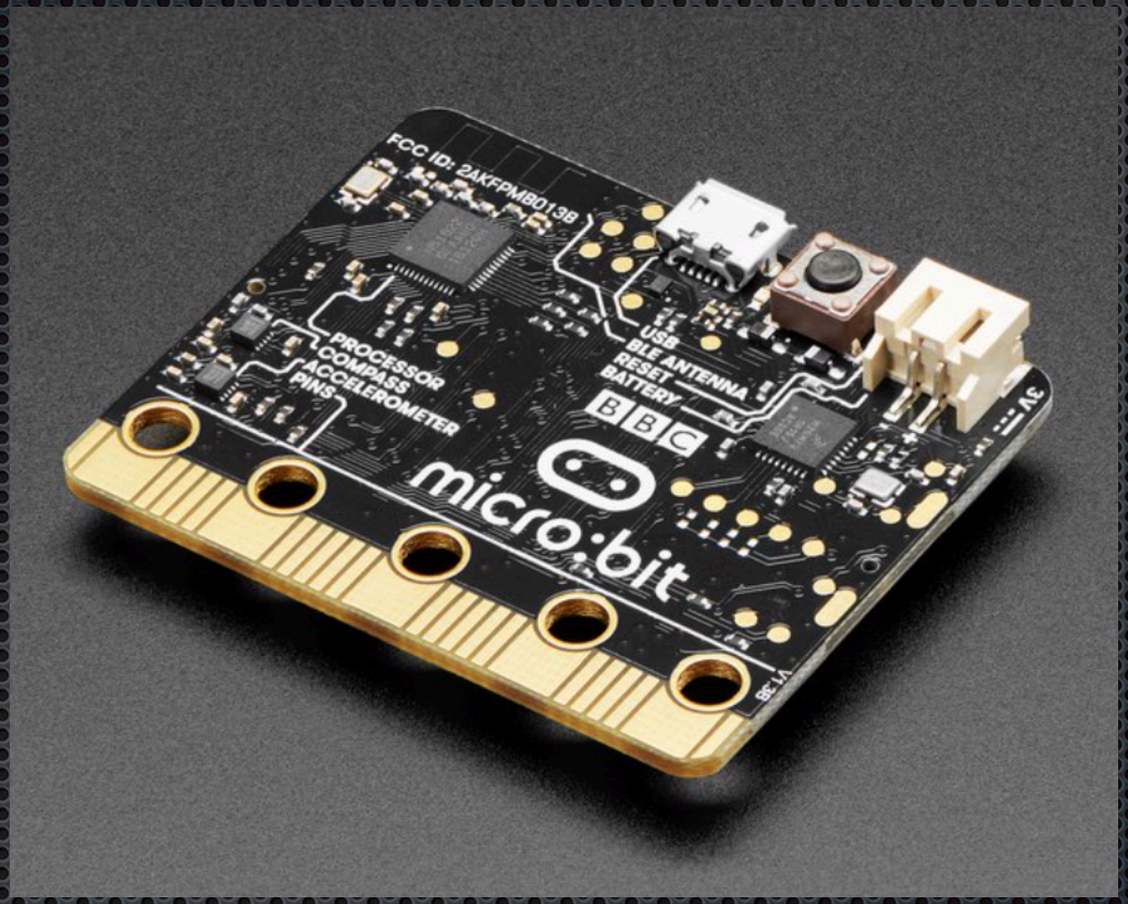
And then deploy the new firmware using:

```
esptool.py --port /dev/ttyUSB0 --baud 460800 write_flash --flash_size=detect 0 esp8266-20170108
```

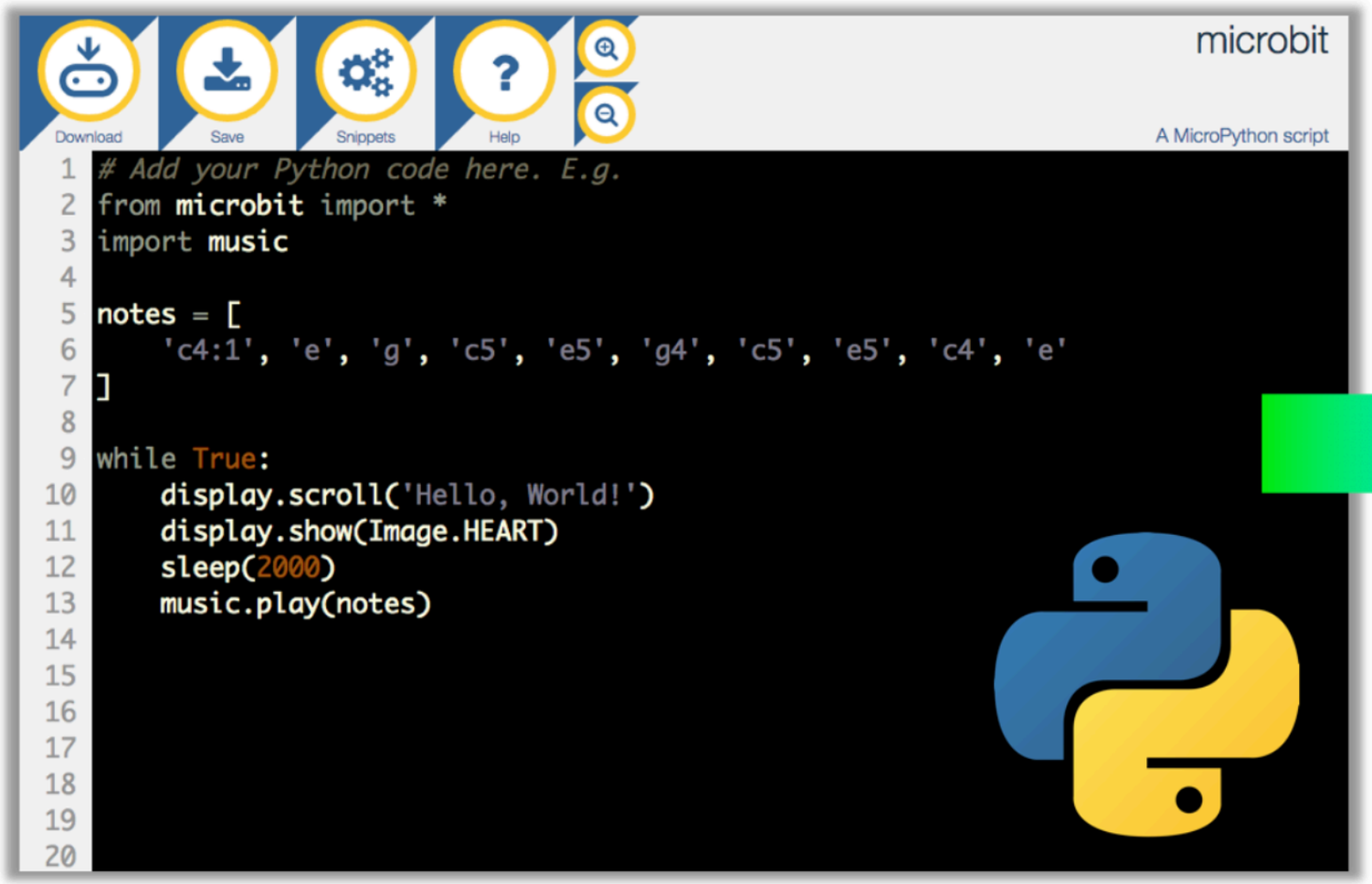
```
>>>
>>>
>>> print('hi')
hi
```

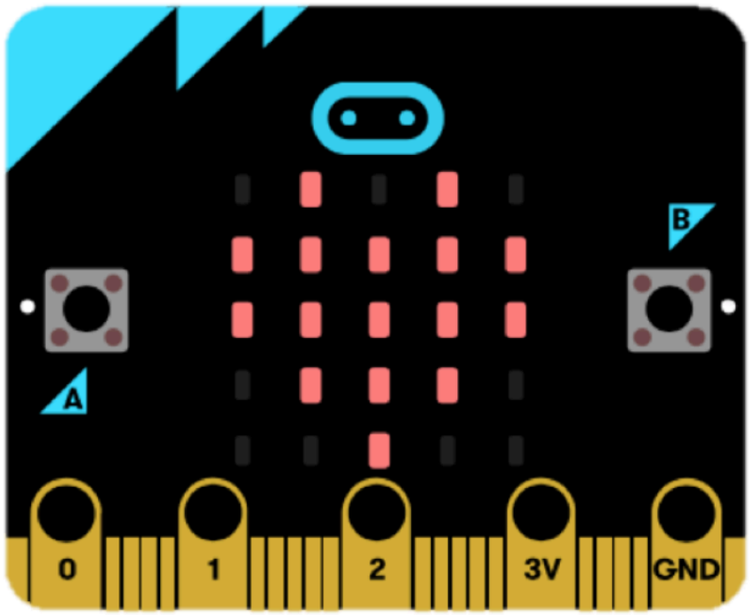
```
>>>
>>>
>>> print('hi')
hi
```

```
>>> import machine, neopixel
>>> np = neopixel.NeoPixel(machine.Pin(15), 8)
>>> np[2] = (0, 0, 64)  # set to blue, quarter brightness
>>> np.write()
```



pictured: BBC micro:bit





The MicroPython guide to BBC micro:bit

thank you!

- extras

Tim Head Look mum no hands! From blinking LEDs to a bike speedometer with MicroPython PyCon 2017

MicroPython

Python that runs on microcontrollers!

- REPL!
- Reimplemented CPython
- Open-source
- Contains a subset of the standard library(!!)

micropython.org

CircuitPython for ESP8266

by lady ada

We are no longer supporting CircuitPython on ESP8266. This page is for historical purposes only. There is no guarantee that the instructions will continue to work.

Why are we dropping support for ESP8266?

CircuitPython on ESP8266 has not been a great experience for users. Its hard to send files to the device because it has no native USB, and you can quickly run out of RAM (there's less than you think once the networking stack is added). We've decided to only use ESP as a co-processor. Specifically, the ESP32 because it has good TLS/SSL support which is now essential for even the most basic interactions.

If you'd like to use ESP8266, please keep to version 3.x with the knowledge that it isn't supported. You can also check out MicroPython for ESP8266 which is still supported!

source: https://learn.adafruit.com/welcome-to-circuitpython/circuitpython-for-esp8266