

Physical Computing 101

Stanford d.school – October 2025

1. Introduction
2. Why prototype?
3. Workshop
4. Q&A

I'm Kay 🙋





ProtoPie

above



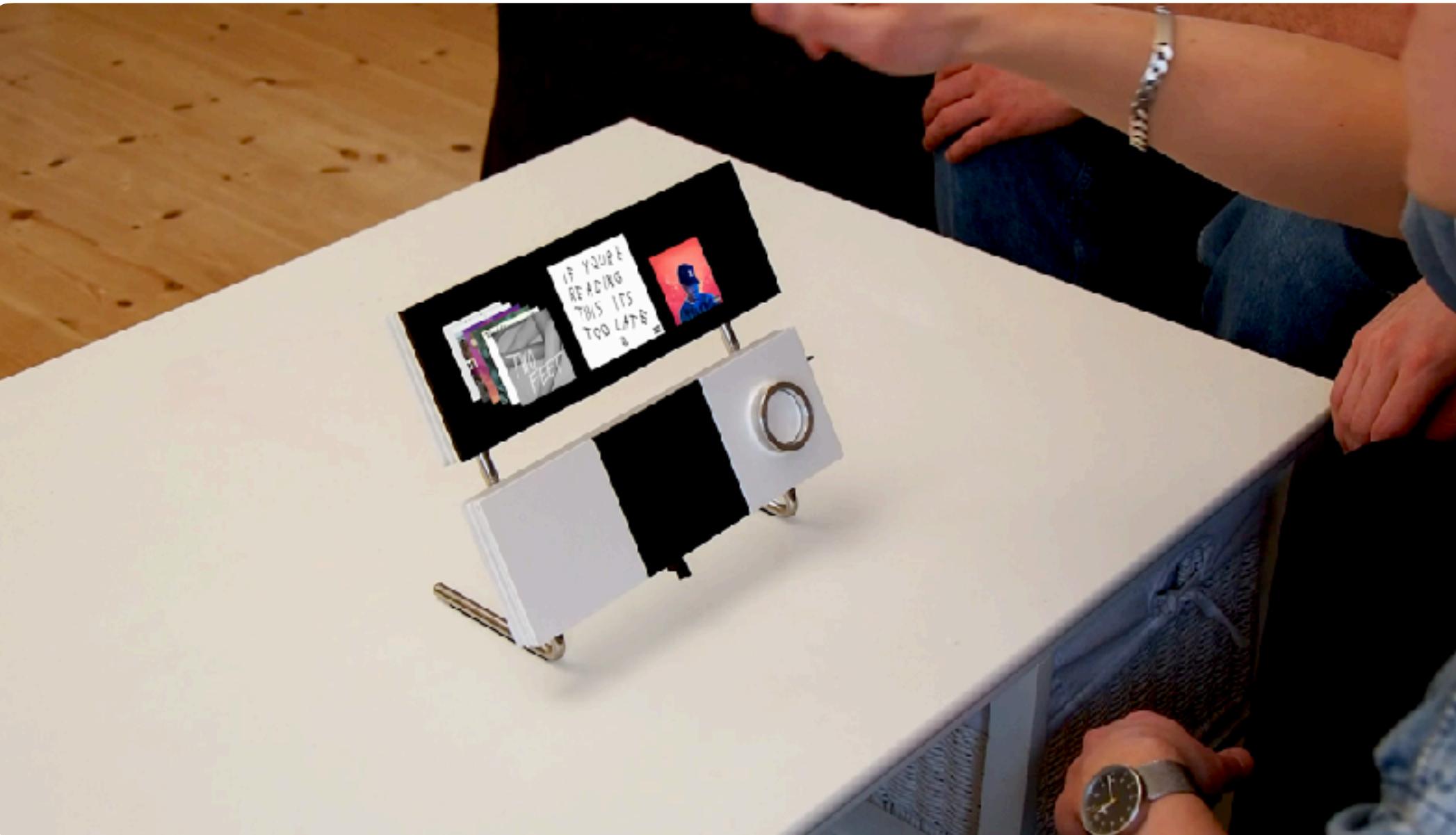
Interaction Designer & Prototyper



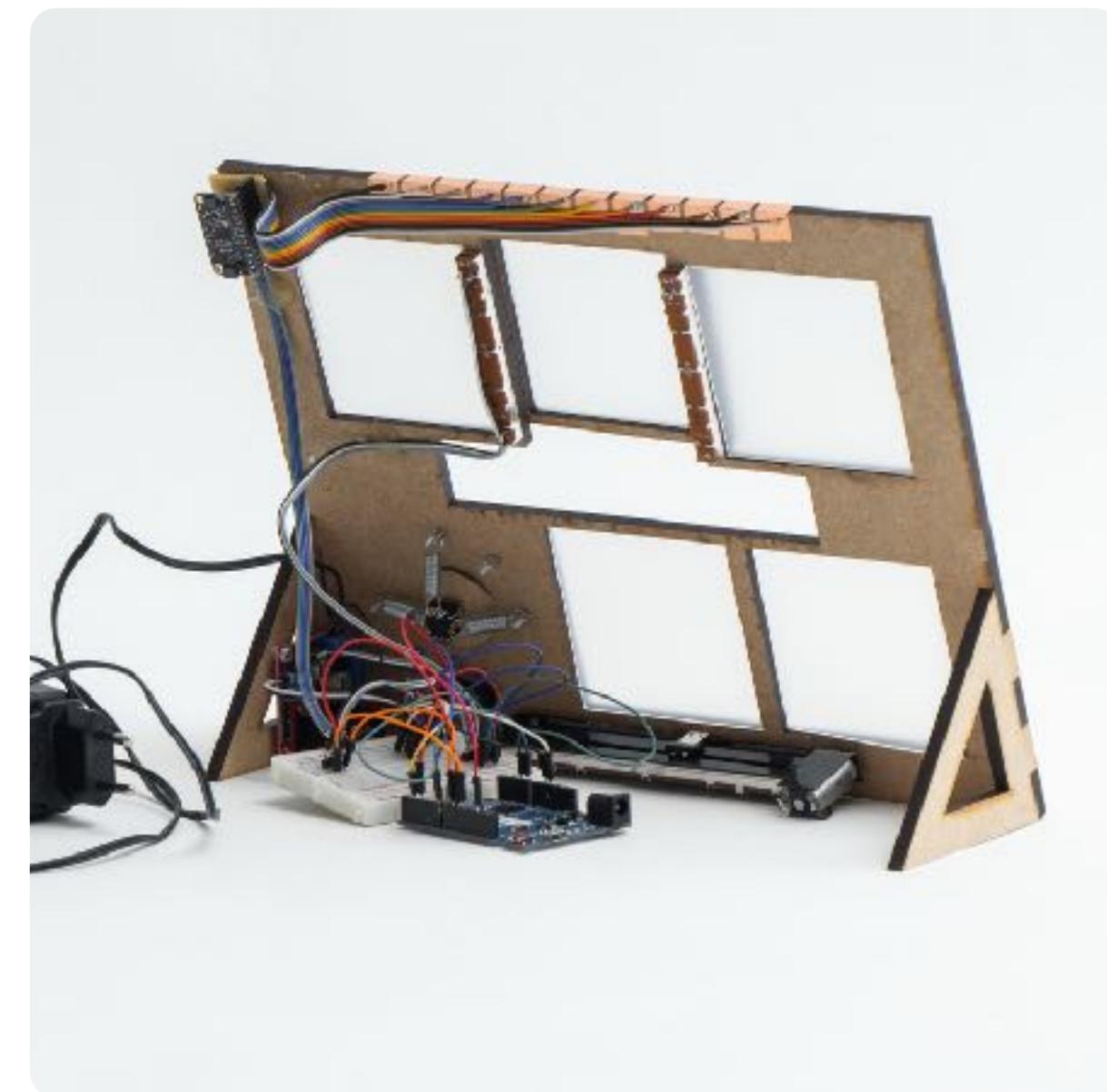
Archetype^(AI)

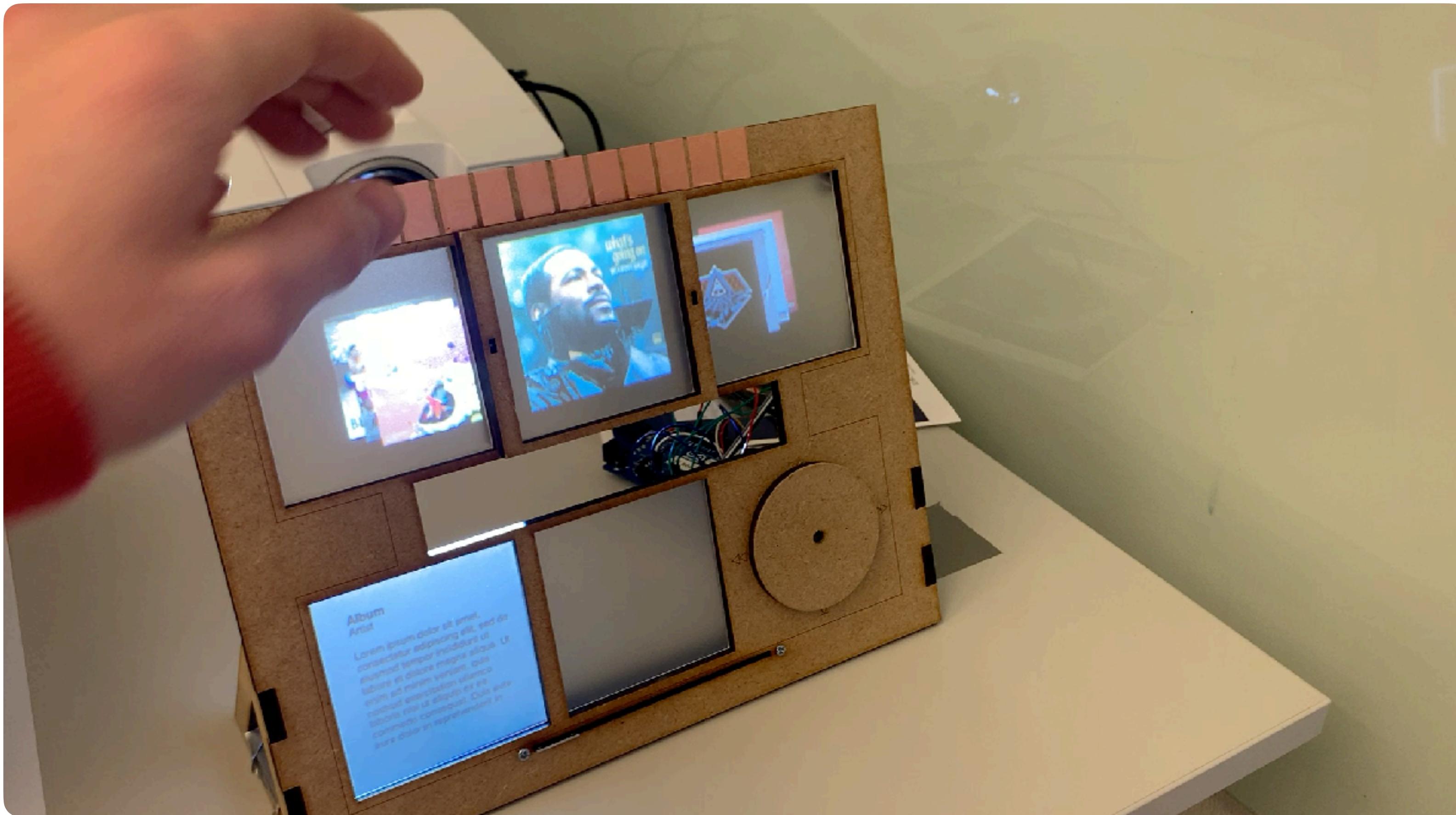


Prototyping hardware experiences

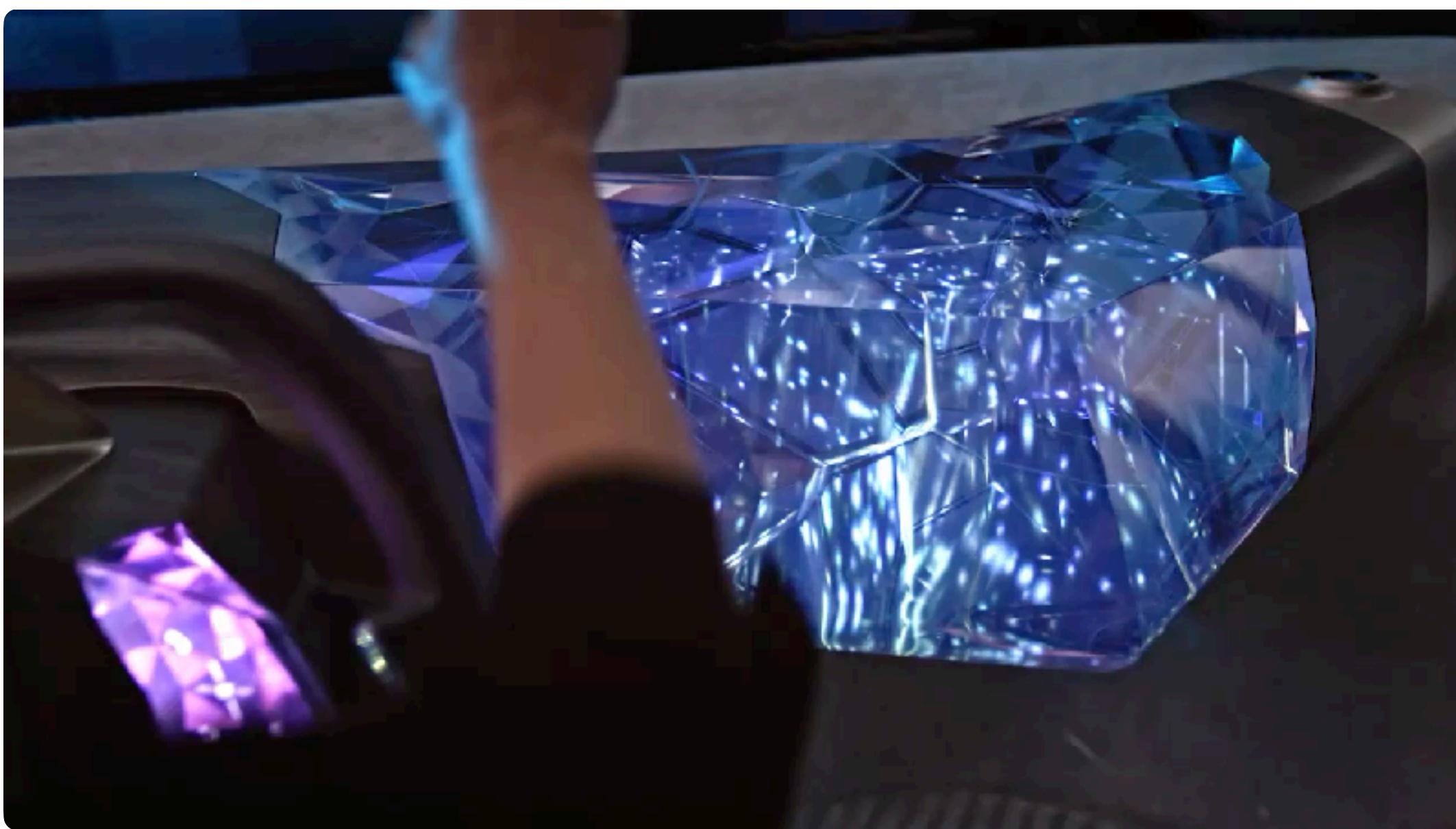


Using tech as an explorative material

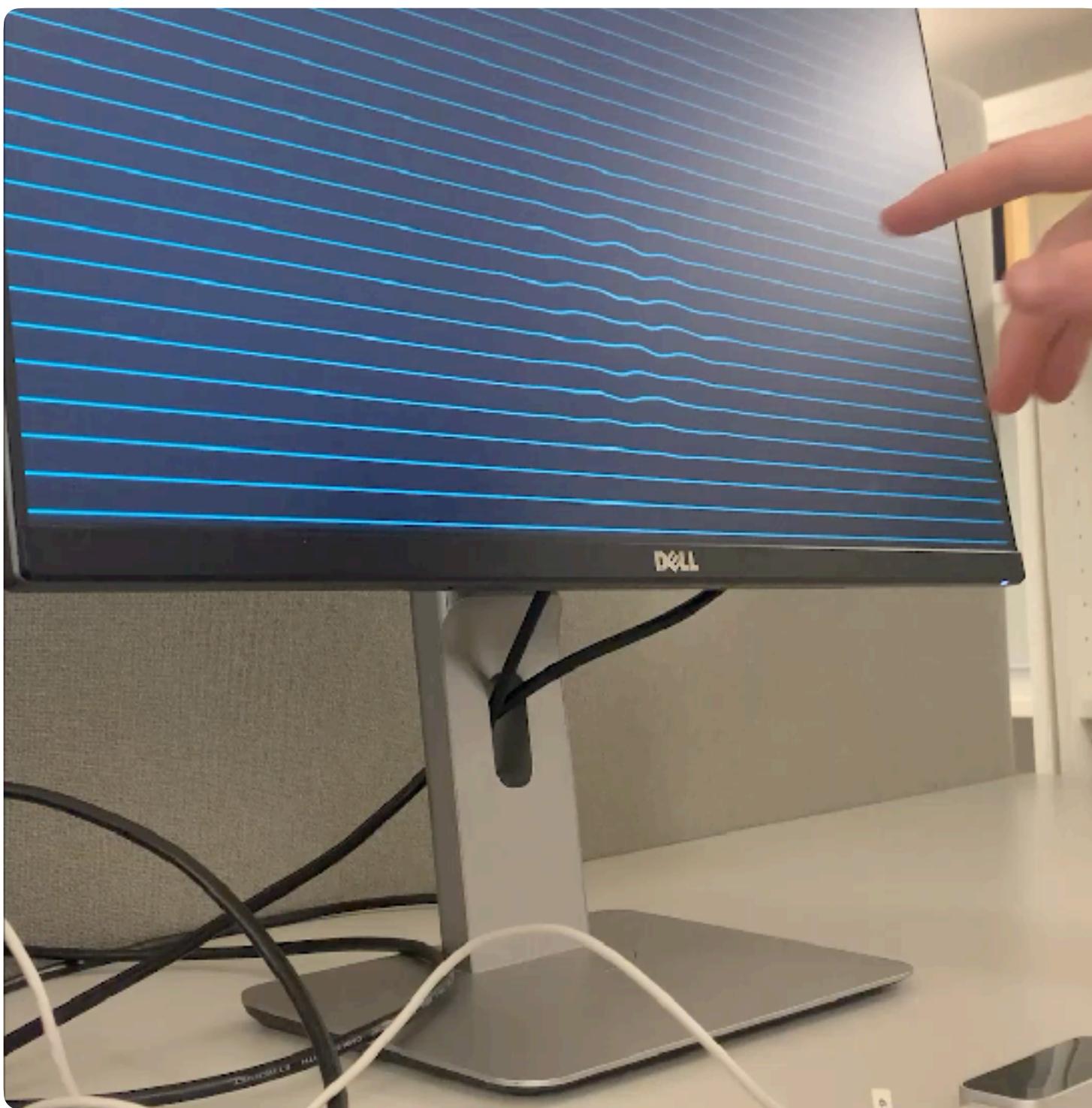


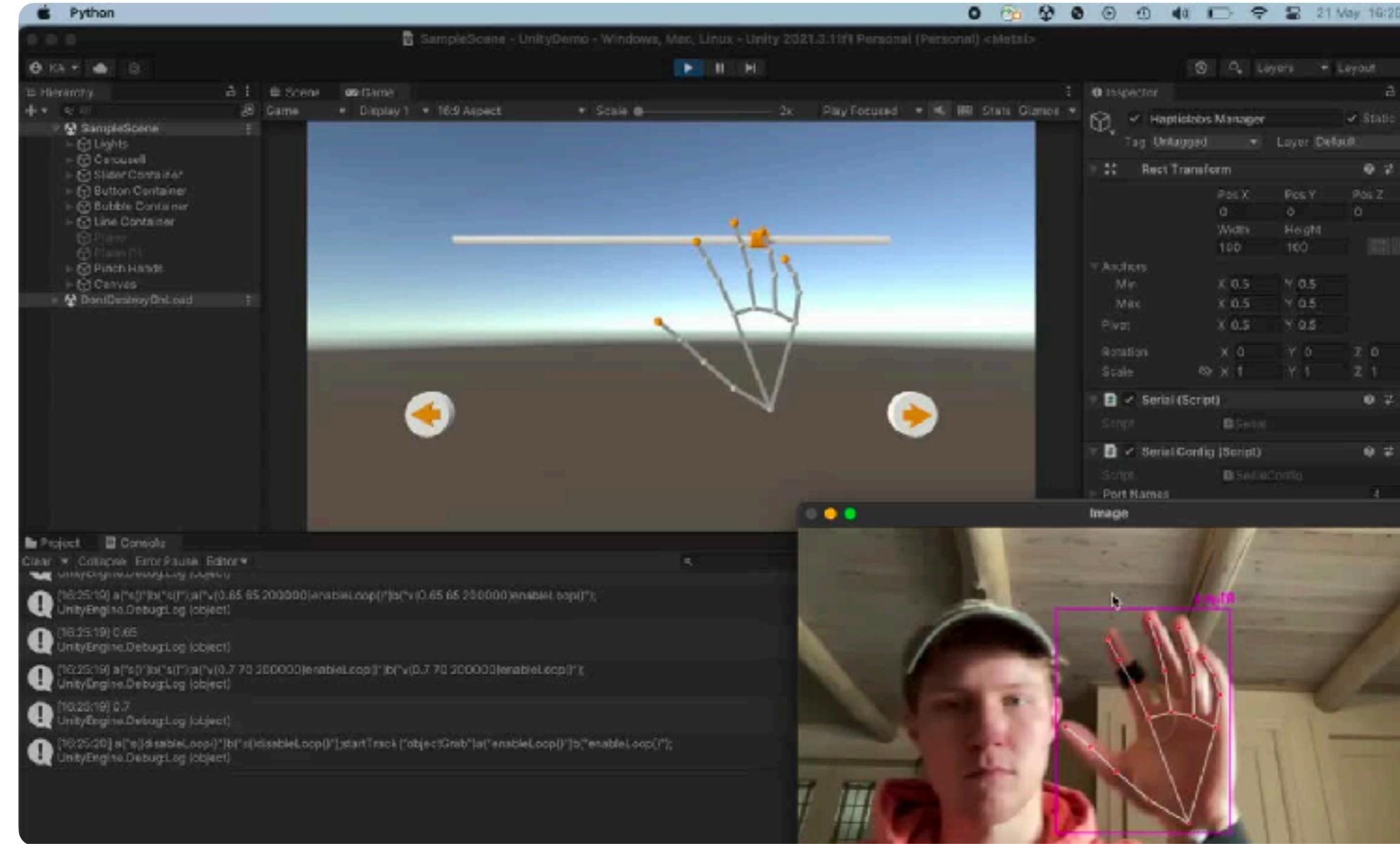


Experience prototyping



Fusing digital and physical

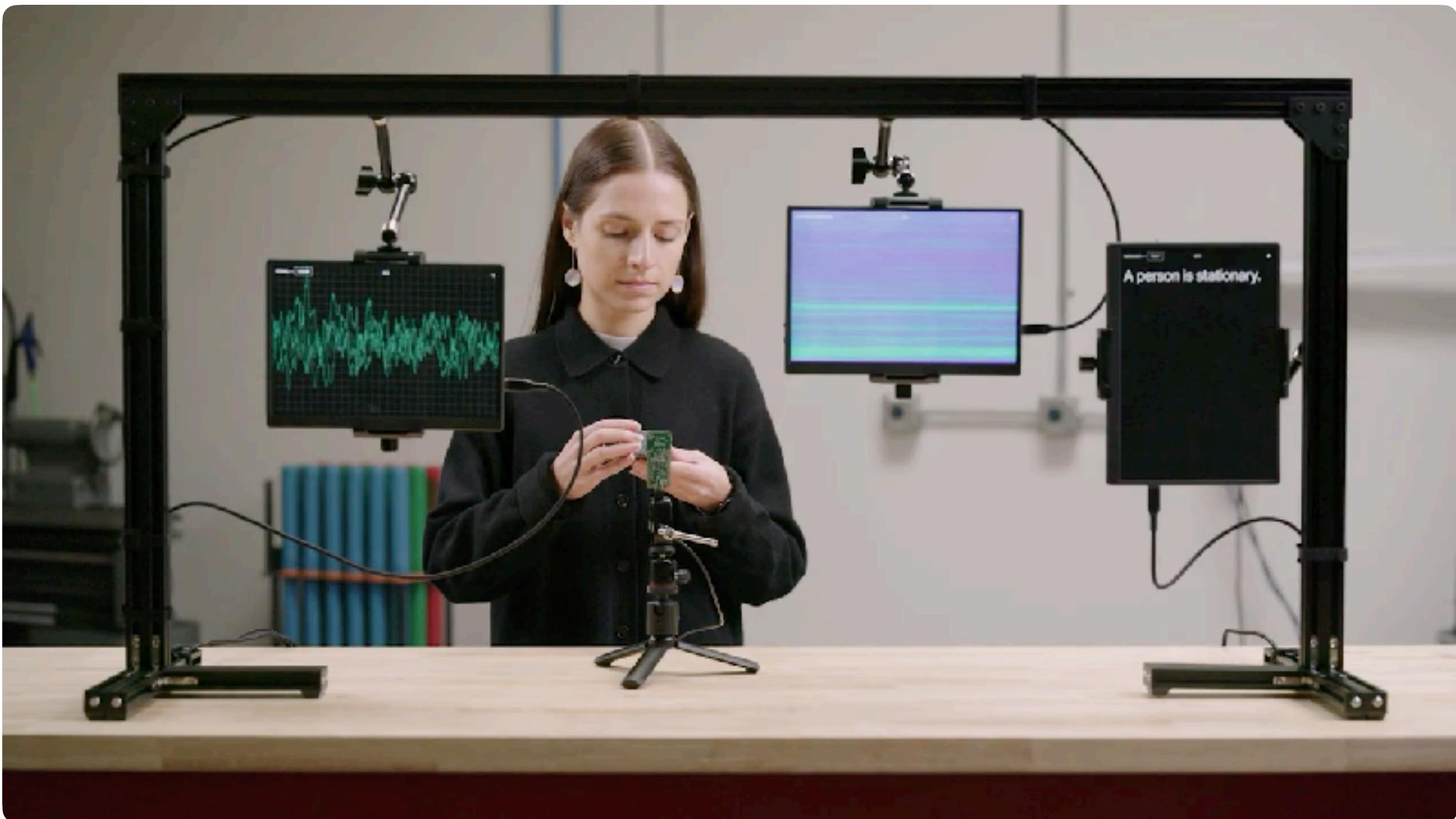




Bringing haptics into the digital world



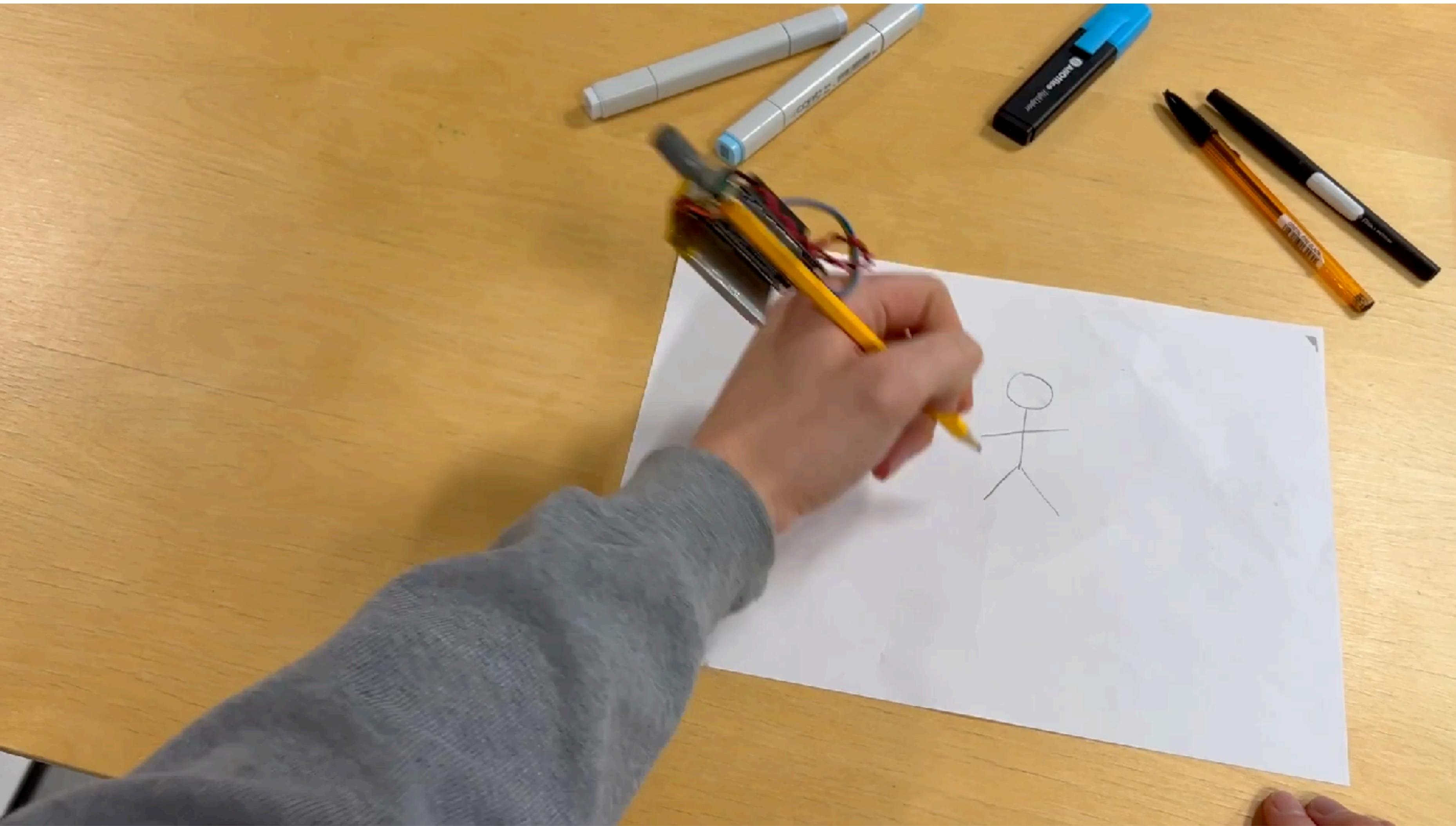
Humans and tech expressing intent towards each other



Enabling new experiences



Binoculars



Augmenting traditional tools with AI

Currently at the Google Envisioning Studio

Why prototype? 🤔

Explore and experience possible futures

A platform for discussion

Thinking through making

Prototype, try, iterate



Expose potential

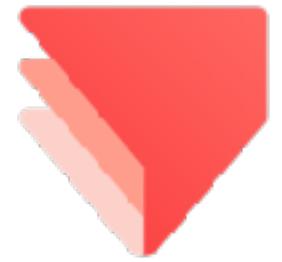
**Prototyping is about experiencing an idea,
not building the first version of a product.**

Also, it's just tons of fun to build stuff!!

Software Prototyping



Figma



ProtoPie



TouchDesigner



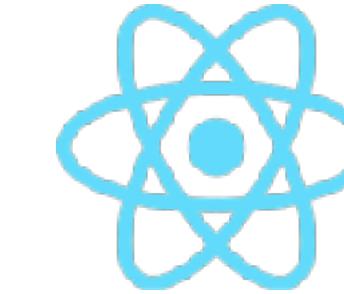
Lovable



Processing



Unity



React

Hardware Prototyping

Existing hardware

- Smartphones
- Keyboards
- Camera
- Microphone
- Speaker
- Kinect
- Leap Motion
- TouchBar



Arduino



KiCad



Adafruit

Input



Code



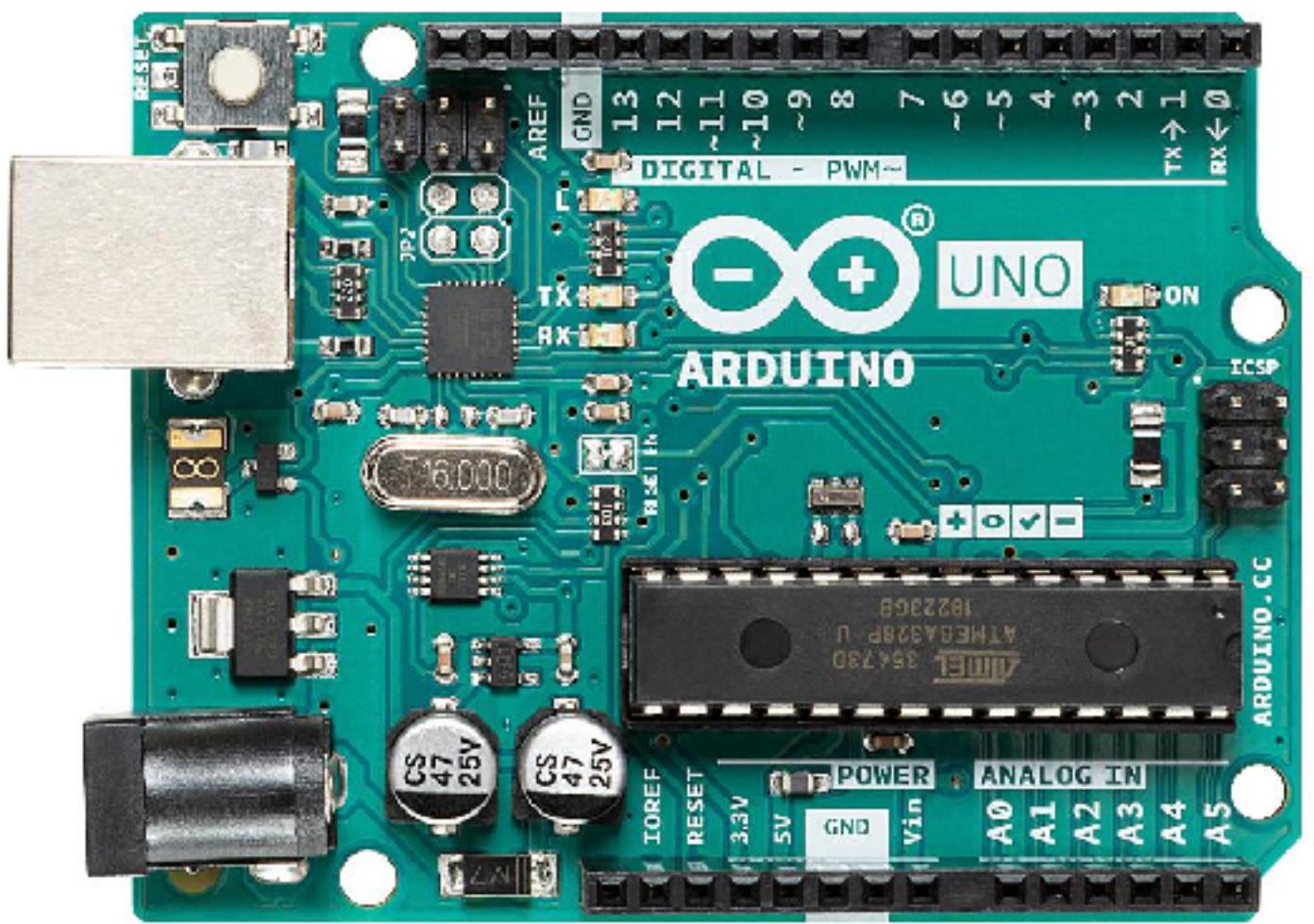
Output

Sensors sense somethings in the environment, and convert it into a signal.

Actuators receive a signal and use it to perform an action or motion.

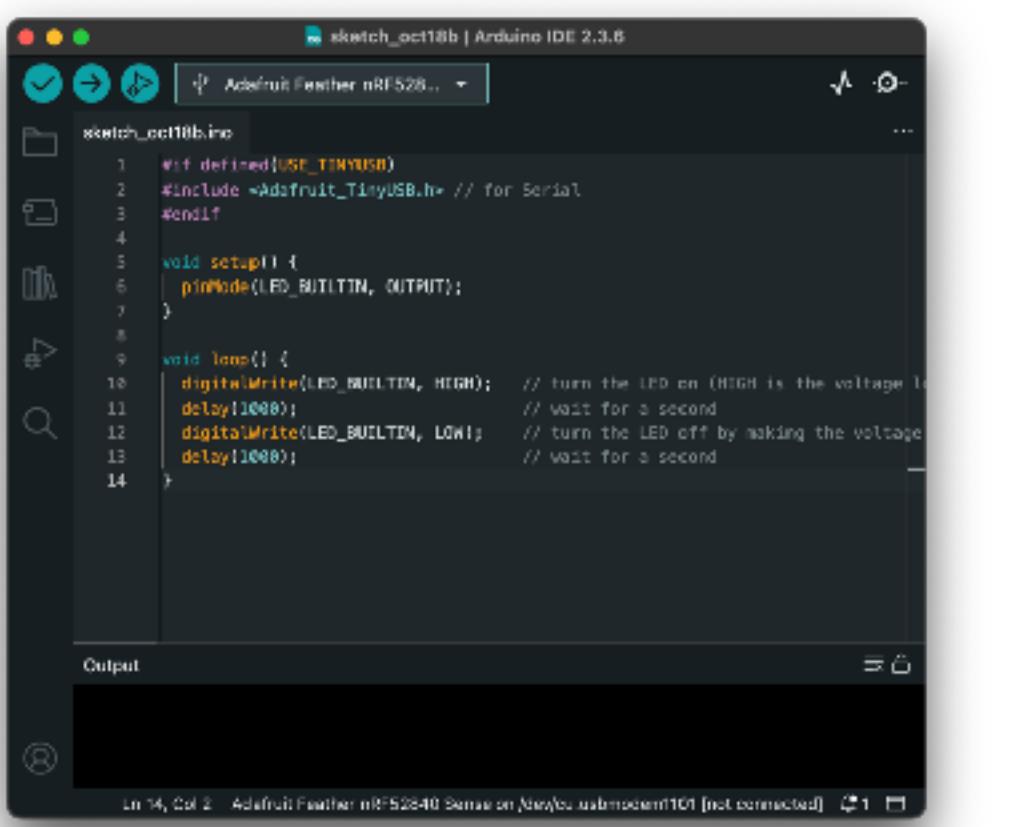


Custom Printed Circuit Board (PCB)



Programmable Circuit Board (microcontroller)

**Microcontrollers enable creators to reprogram
and reuse electronics for various purposes.**

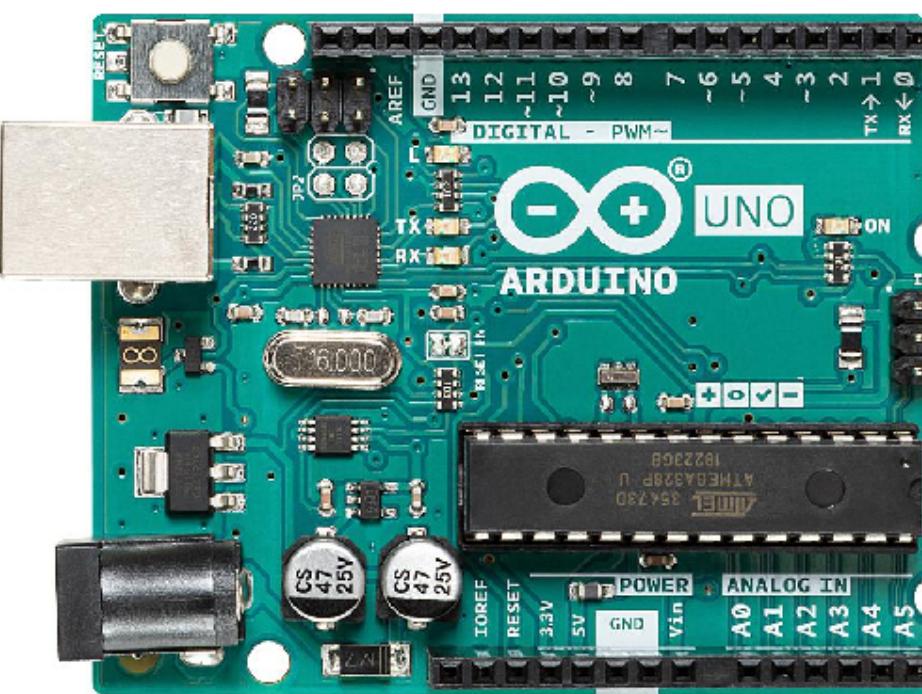


The screenshot shows the Arduino IDE interface with the following code:

```
sketch_oct18b | Arduino IDE 2.3.6
Adafruit Feather nRF52840

sketch_oct18b.ino
1 // If defined USE_TINYUSB
2 #include <Adafruit_TinyUSB.h> // for Serial
3 #endif
4
5 void setup() {
6   pinMode(LED_BUILTIN, OUTPUT);
7 }
8
9 void loop() {
10   digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)
11   delay(1000); // wait for a second
12   digitalWrite(LED_BUILTIN, LOW); // turn the LED off by making the voltage low
13   delay(1000); // wait for a second
14 }
```

Integrated Development
Environment (IDE)

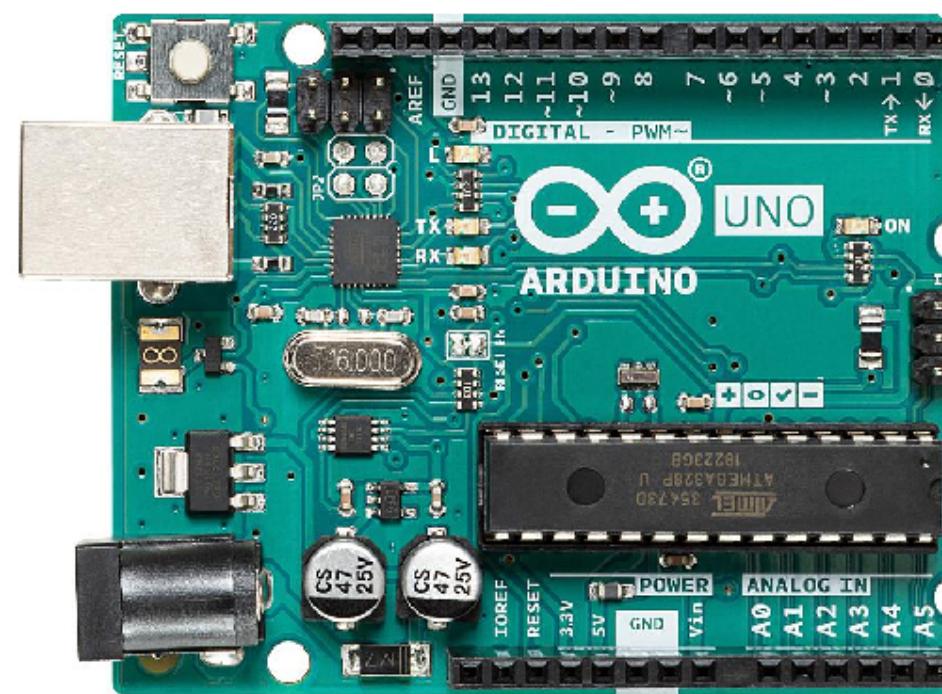


Microcontroller

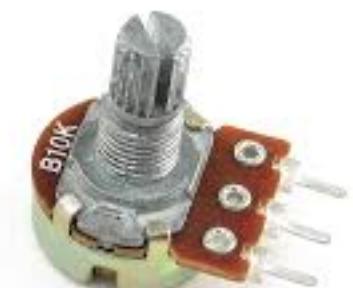
The screenshot shows the Arduino IDE interface with the following code:

```
sketch_neo18b | Arduino IDE 2.3.6
Arduino Feather nRF52840 - (USB)
sketch_neo18b.ino
1 // If defined USE_TINYUSB
2 #include <Adafruit_TinyUSB.h> // for Serial
3 #endif
4
5 void setup() {
6   pinMode(LED_BUILTIN, OUTPUT);
7 }
8
9 void loop() {
10   digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)
11   delay(1000); // wait for a second
12   digitalWrite(LED_BUILTIN, LOW); // turn the LED off by making the voltage low
13   delay(1000); // wait for a second
14 }
```

Integrated Development
Environment (IDE)

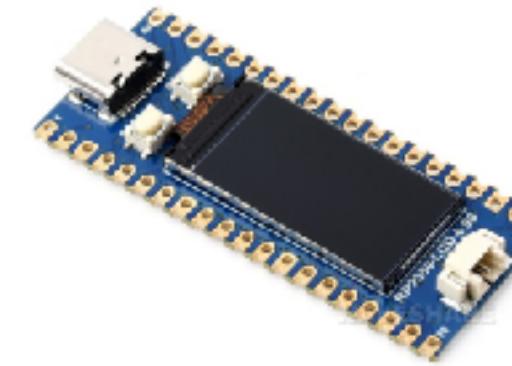
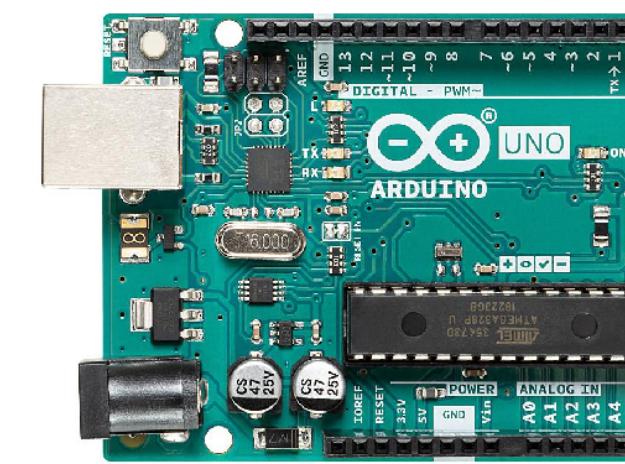
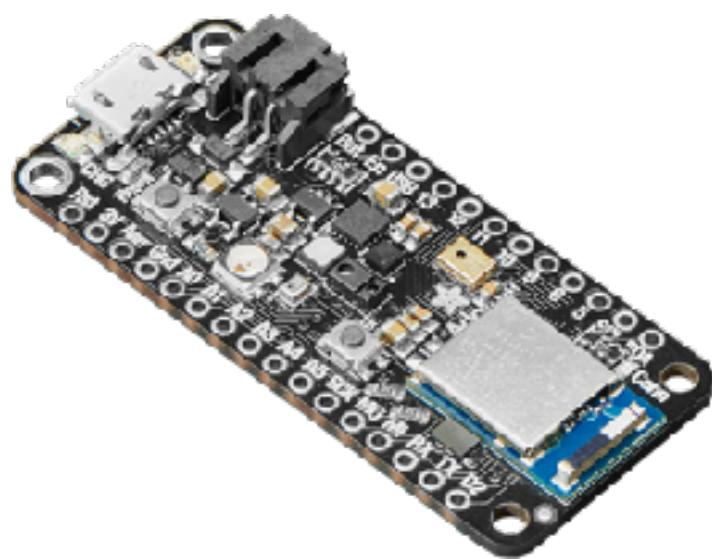
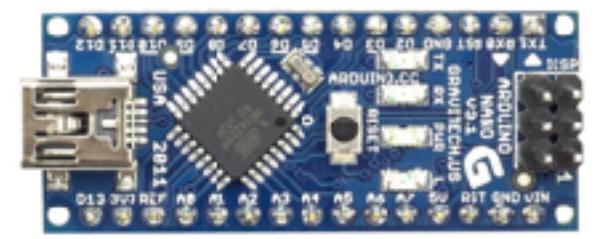


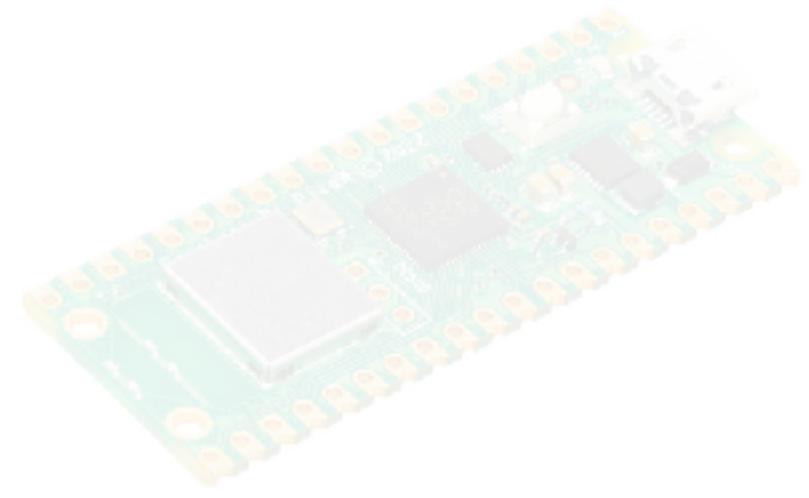
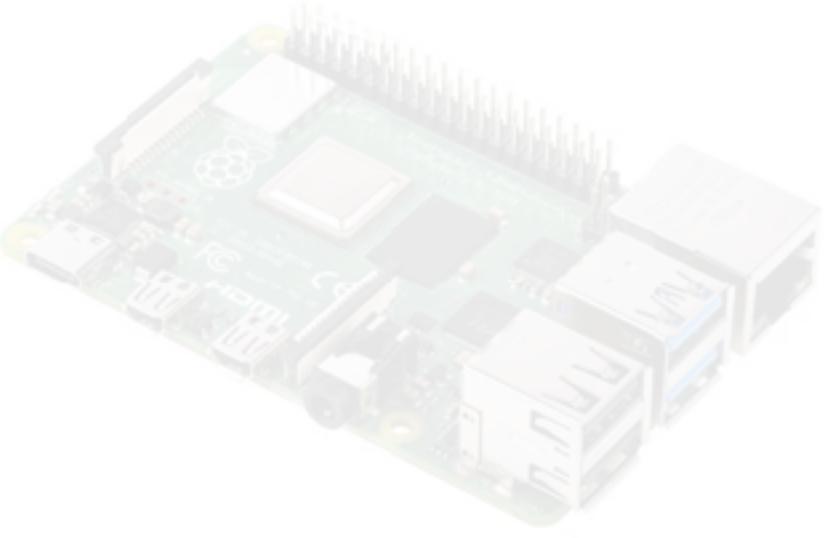
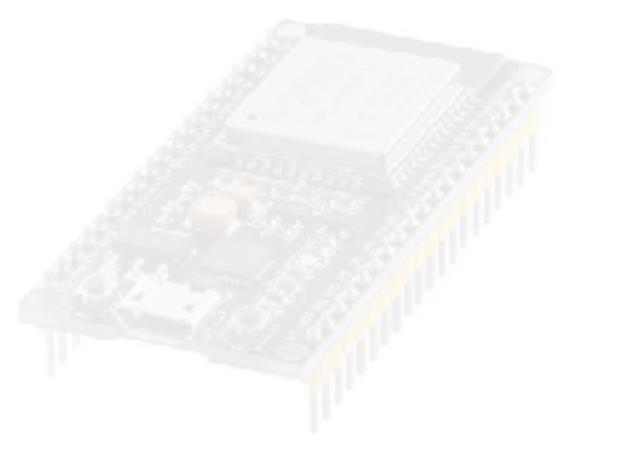
Microcontroller



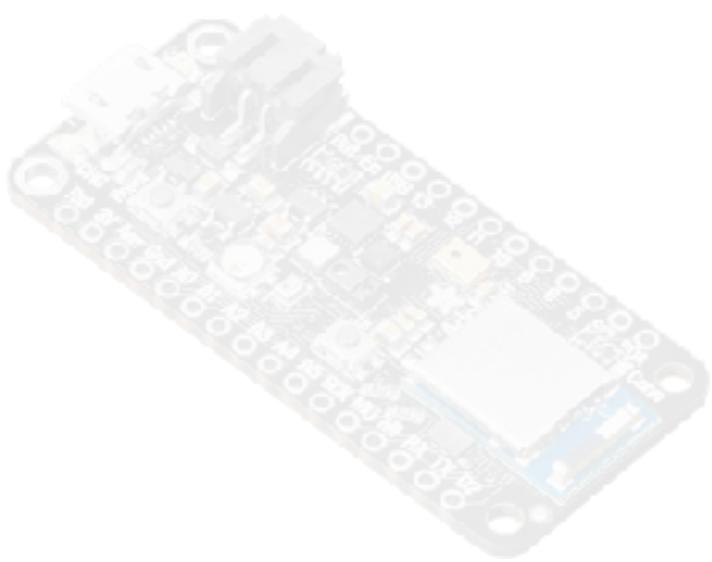
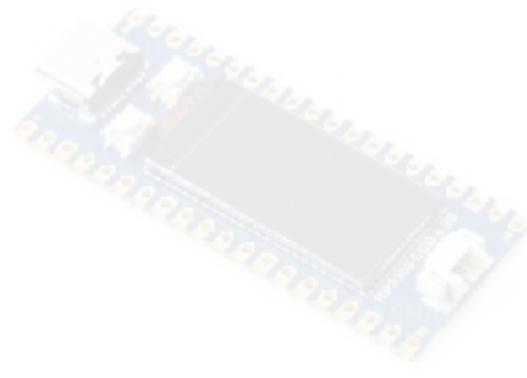
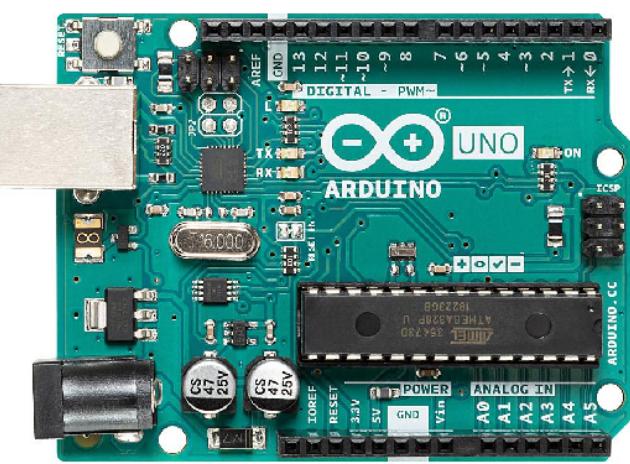


There are endless microcontroller options,
each with their own features and price.

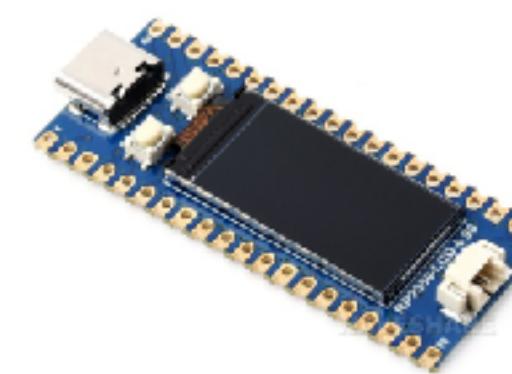
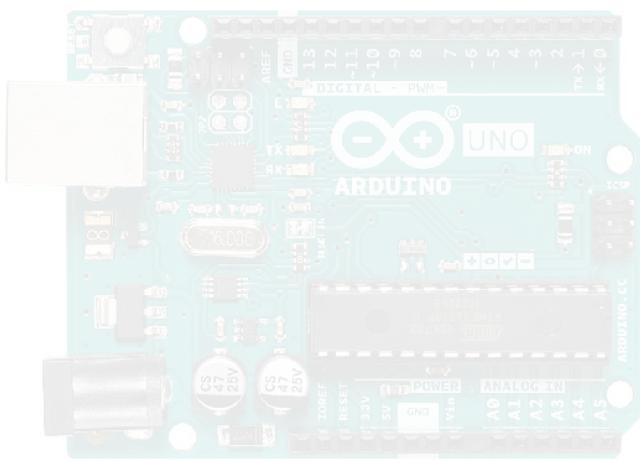
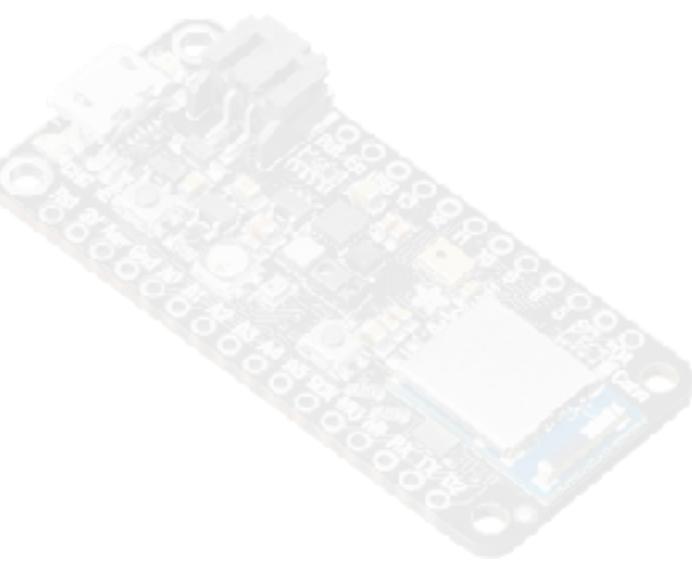
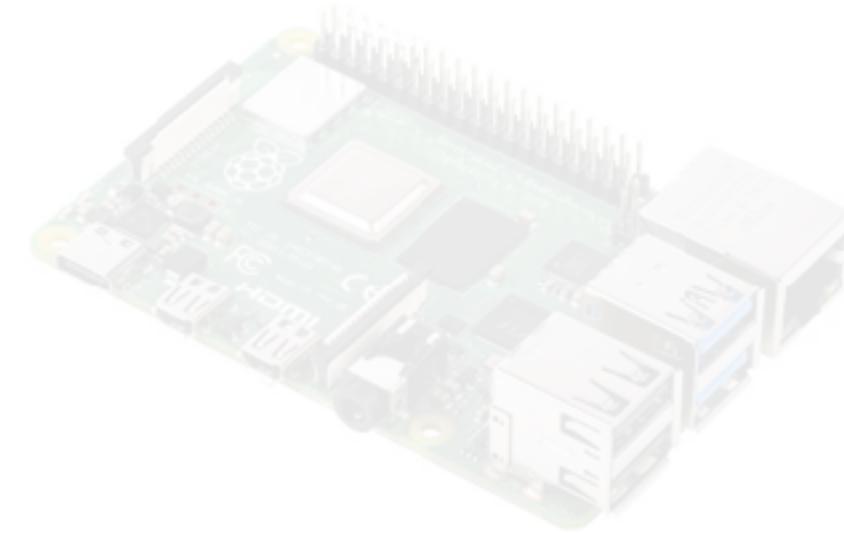
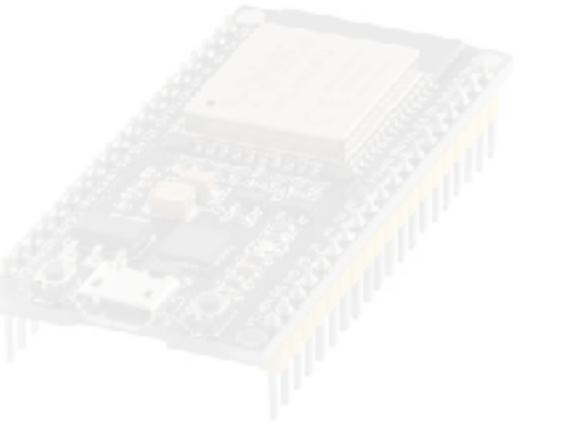
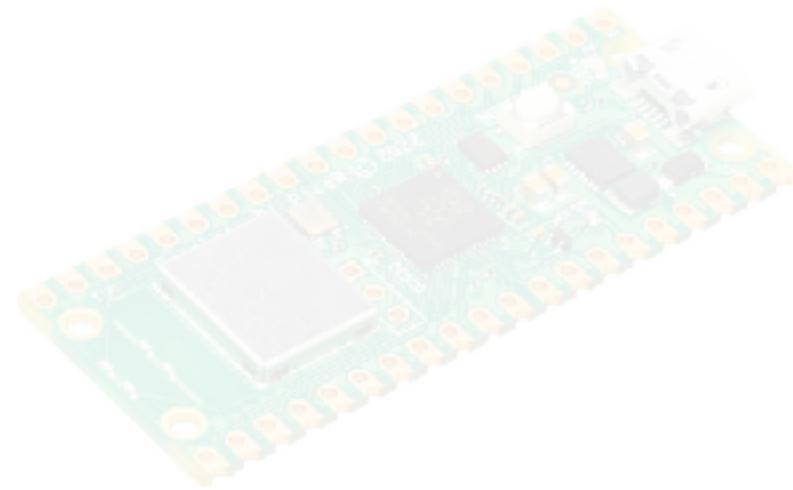
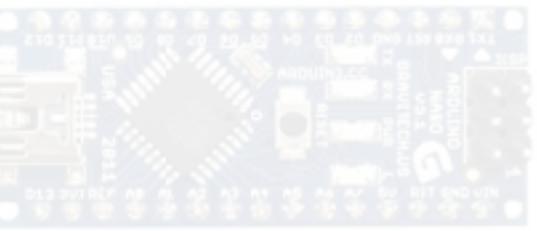


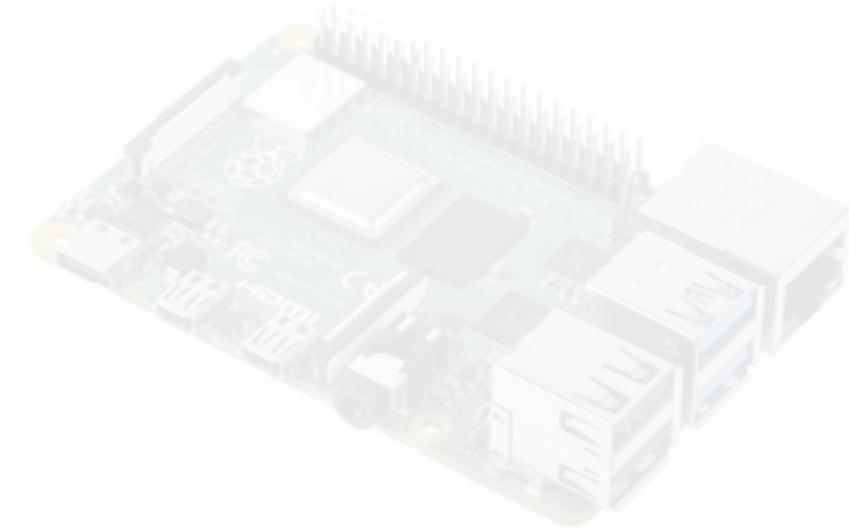


Basic ones ...

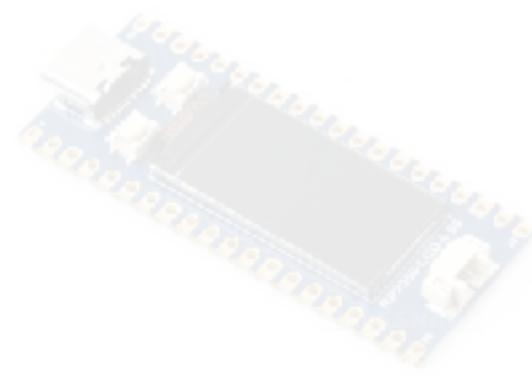
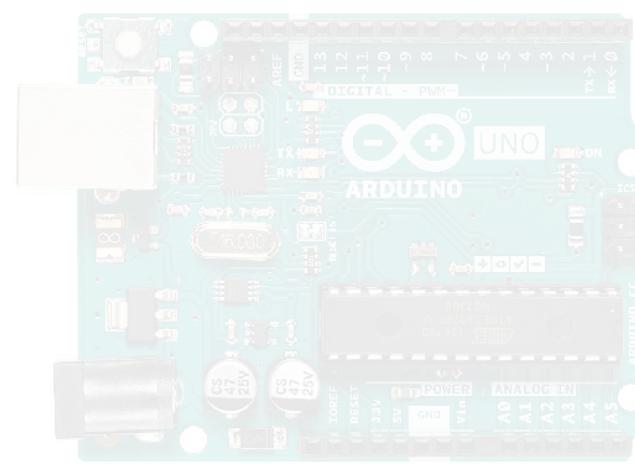
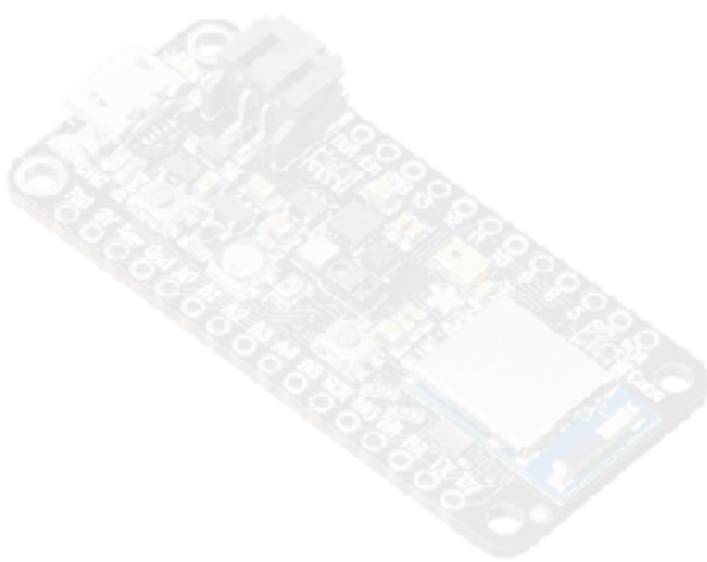
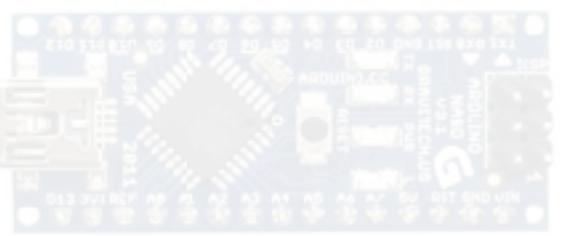


some have tiny displays ...



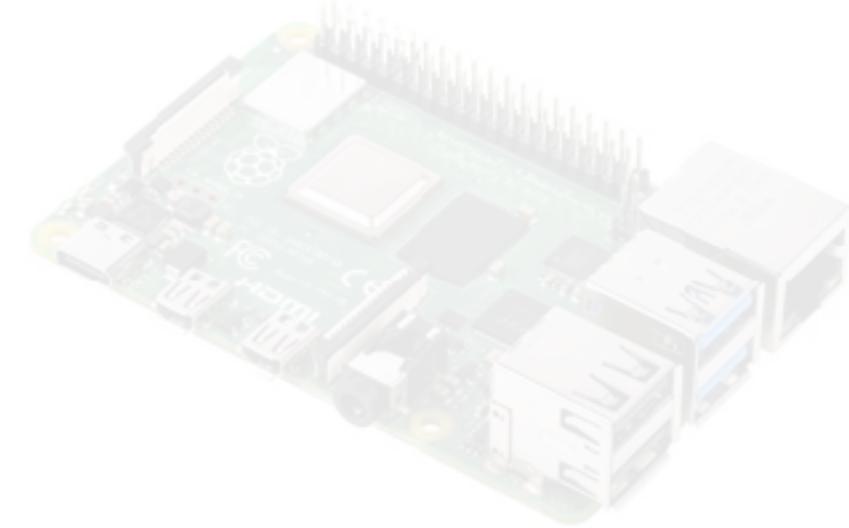
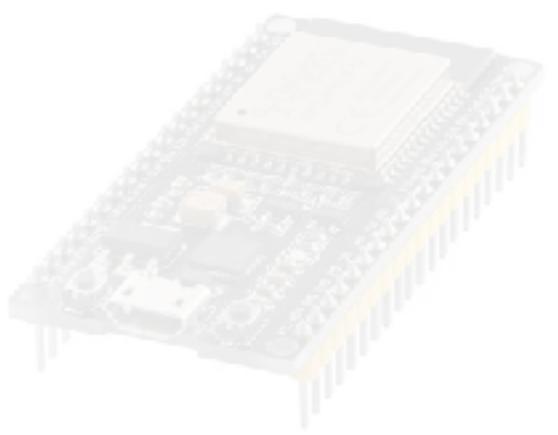
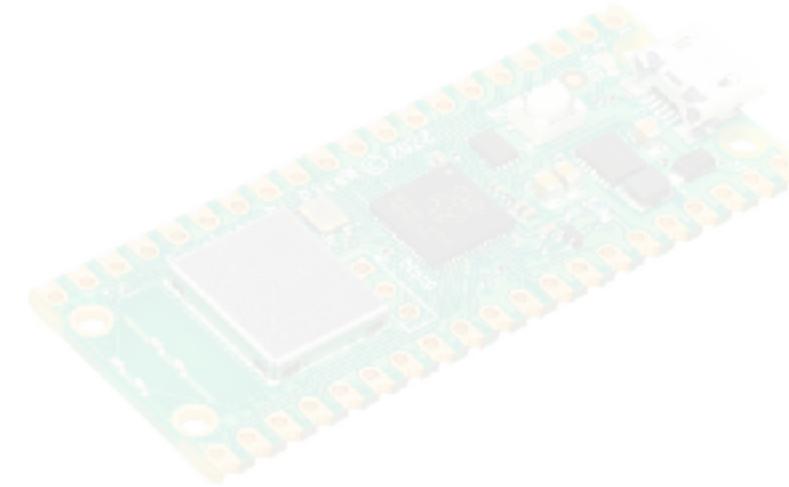


some have Bluetooth or WiFi ...

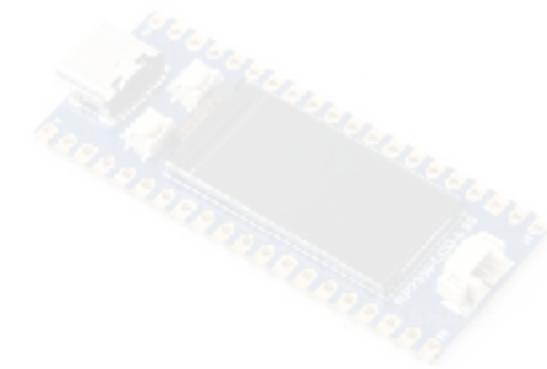
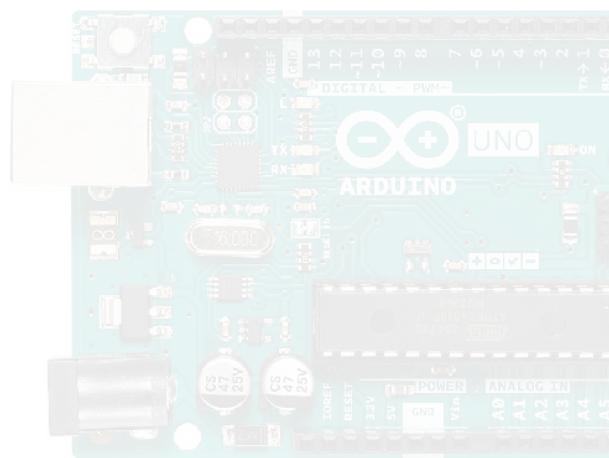
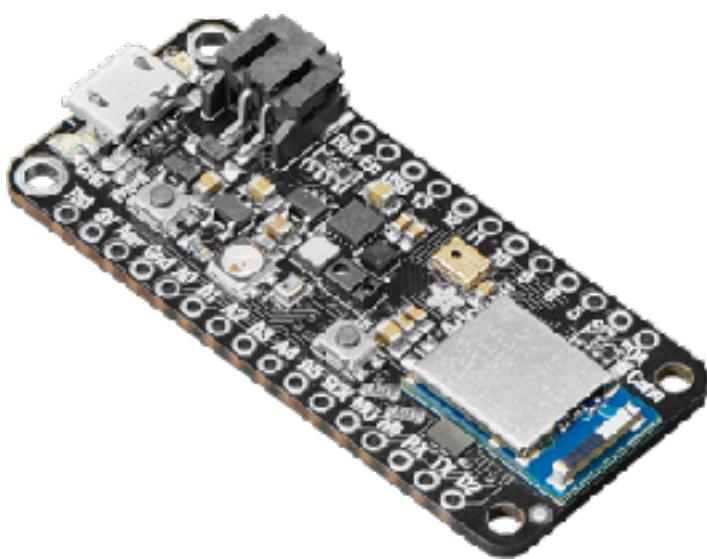
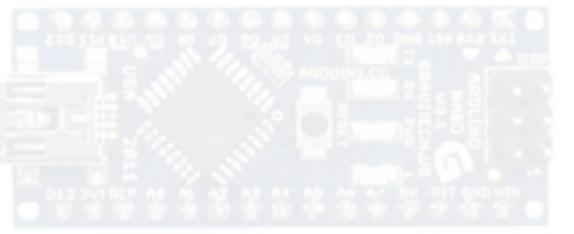


**some can even function like a PC, connecting
a keyboard, mouse, AUX, HDMI display, ...**





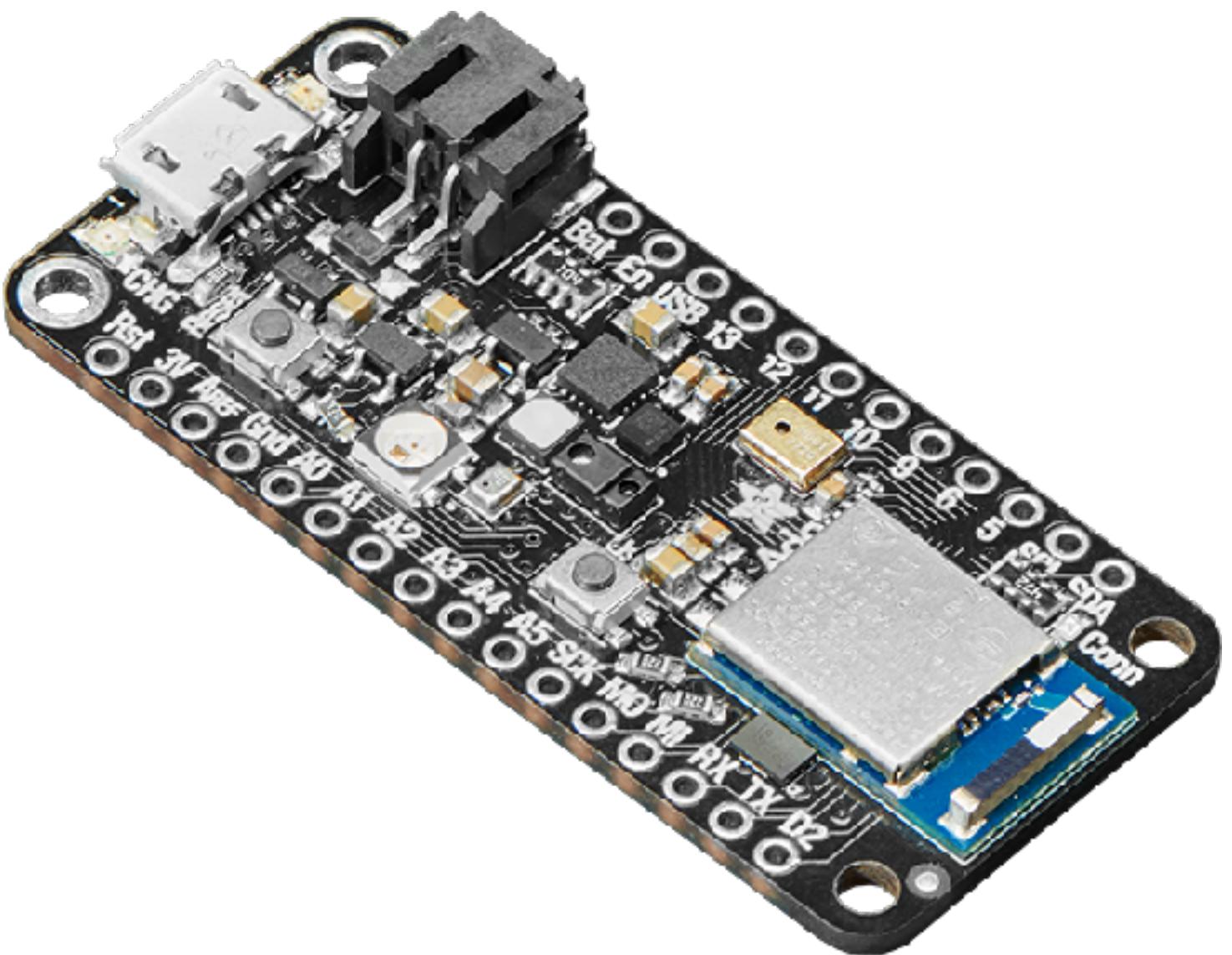
and some have a bunch of sensors built-in
which make them great for prototyping!



Adafruit Feather Sense

Built-in sensors

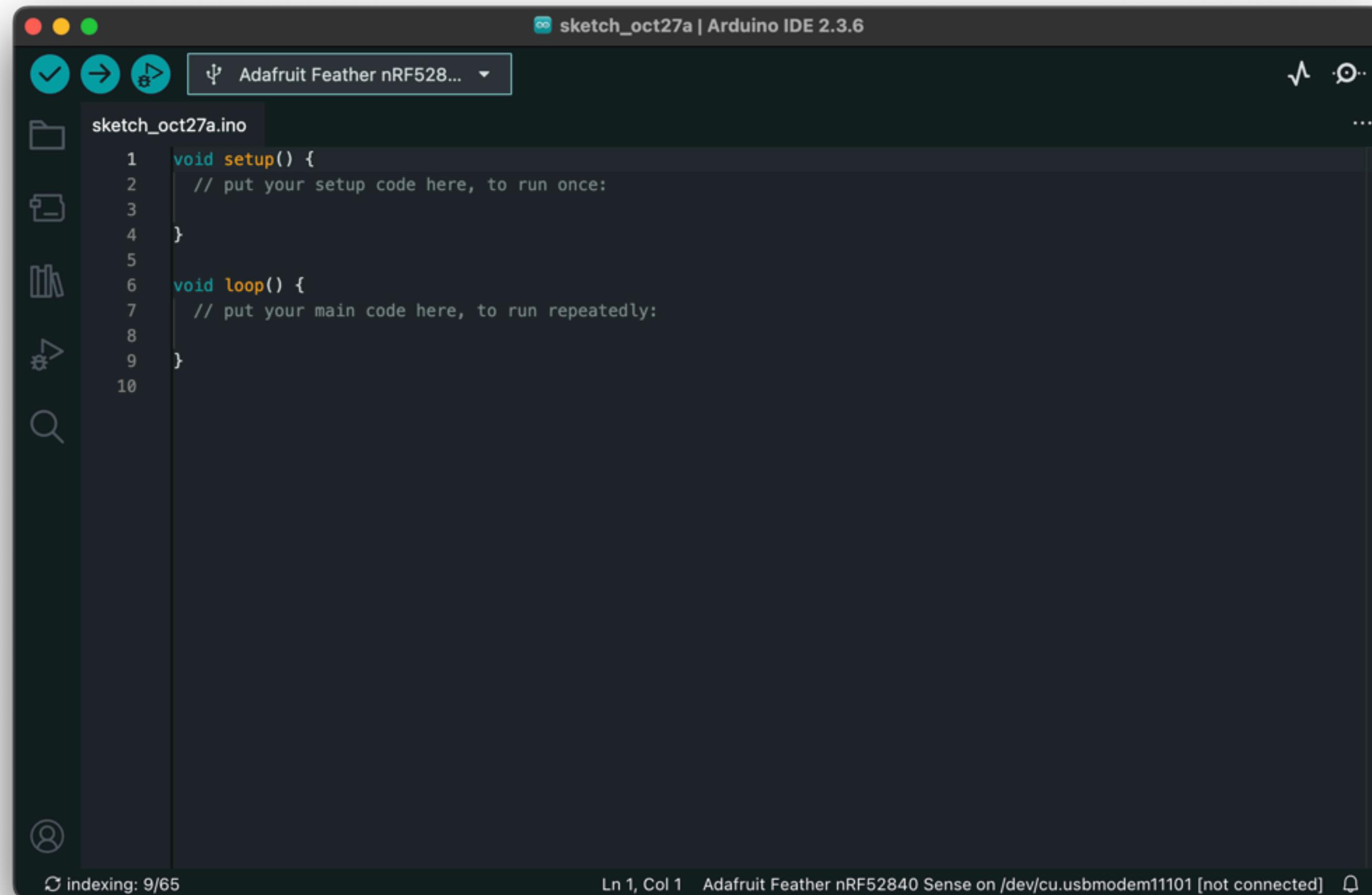
- Humidity
- Magnetometer
- Microphone
- Light
- Color
- Gesture
- Proximity
- Temperature
- Pressure
- Altitude
- Accelerometer
- Gyroscope
- Button



Built-in actuators

- RGB LED
- Red LED

setup() runs once, loop() runs repeatedly

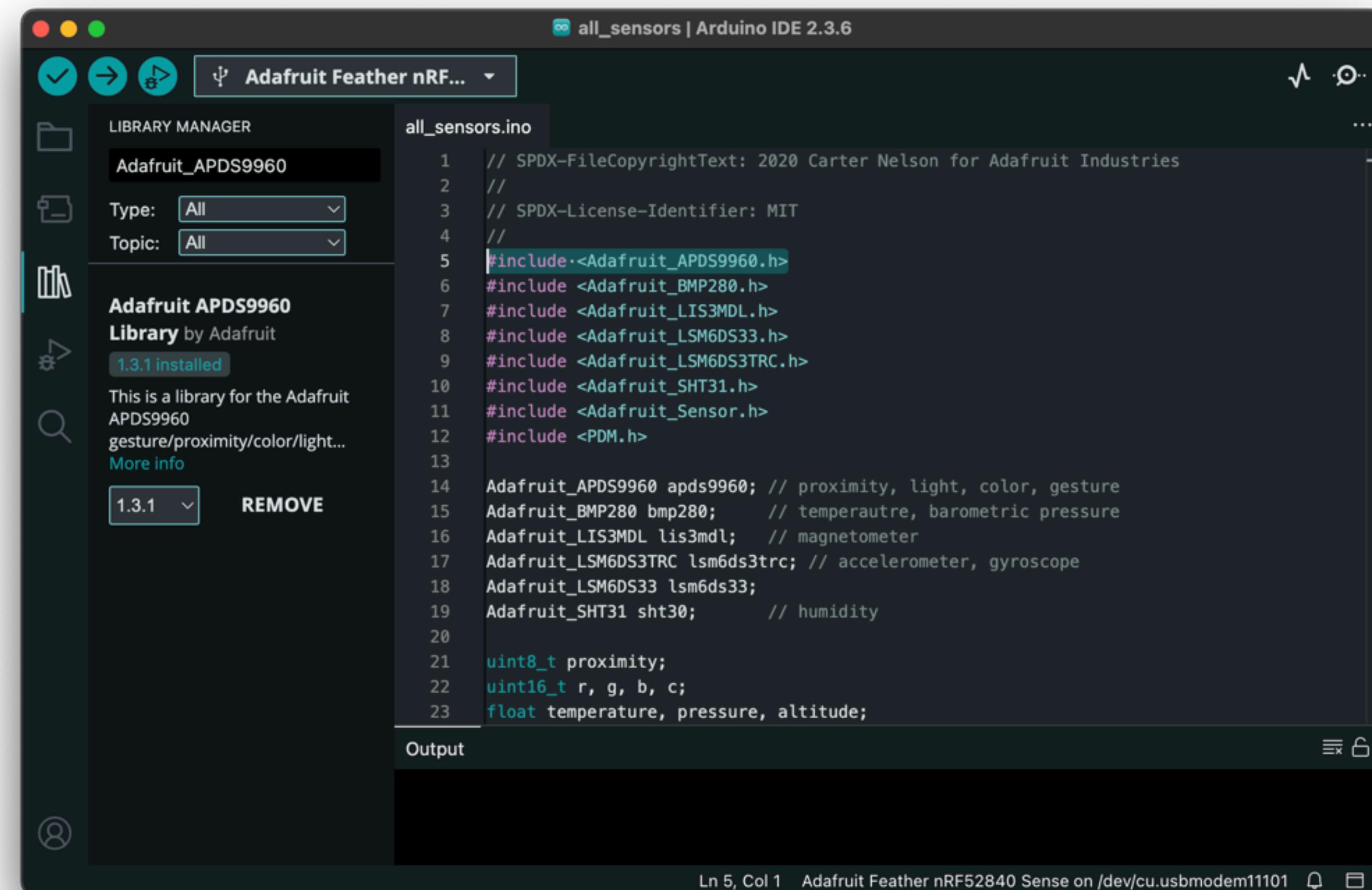


The screenshot shows the Arduino IDE interface with a dark theme. The title bar reads "sketch_oct27a | Arduino IDE 2.3.6". The central code editor window displays the following sketch code:

```
sketch_oct27a.ino
1 void setup() {
2     // put your setup code here, to run once:
3
4 }
5
6 void loop() {
7     // put your main code here, to run repeatedly:
8 }
9
10
```

The code consists of two functions: `setup()` and `loop()`. The `setup()` function is intended to run once at the start of the program, while the `loop()` function is intended to run repeatedly. The Arduino IDE interface includes a toolbar with icons for file operations, a sidebar with navigation and search tools, and a status bar at the bottom.

Install the libraries for the sensors



Try building something with the built-in sensors and actuators

Built-in sensors

- Humidity
- Magnetometer
- Microphone
- Light
- Color
- Gesture
- Proximity
- Temperature
- Pressure
- Altitude
- Accelerometer
- Gyroscope
- Button

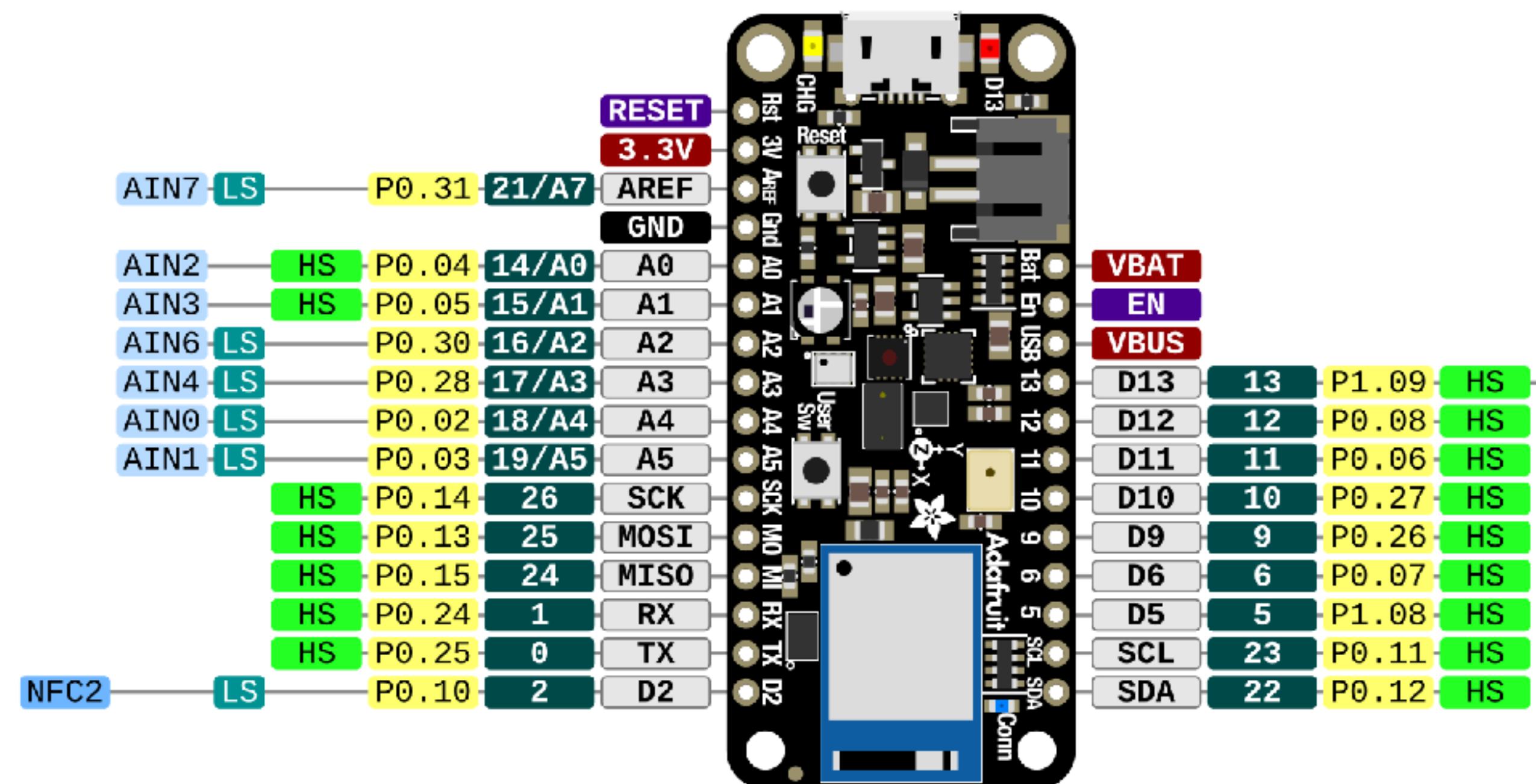
Built-in actuators

- RGB LED
- Red LED

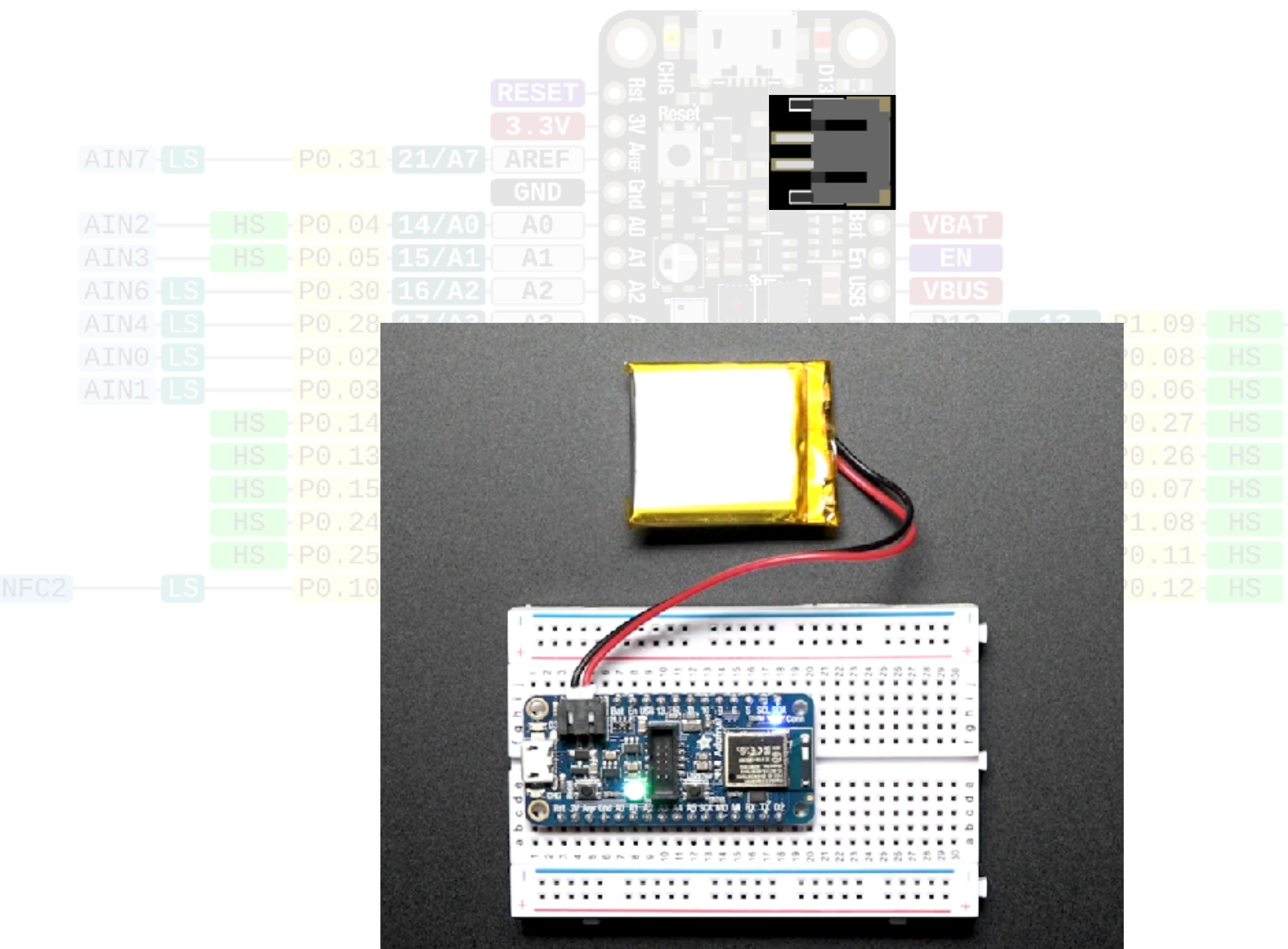
*Click the button, blink the LED?
Tilt it, increase brightness?
Clap changes color?*

github.com/kayvandenaker/workshop

To hook up other electronics, check the pinout for the wiring.



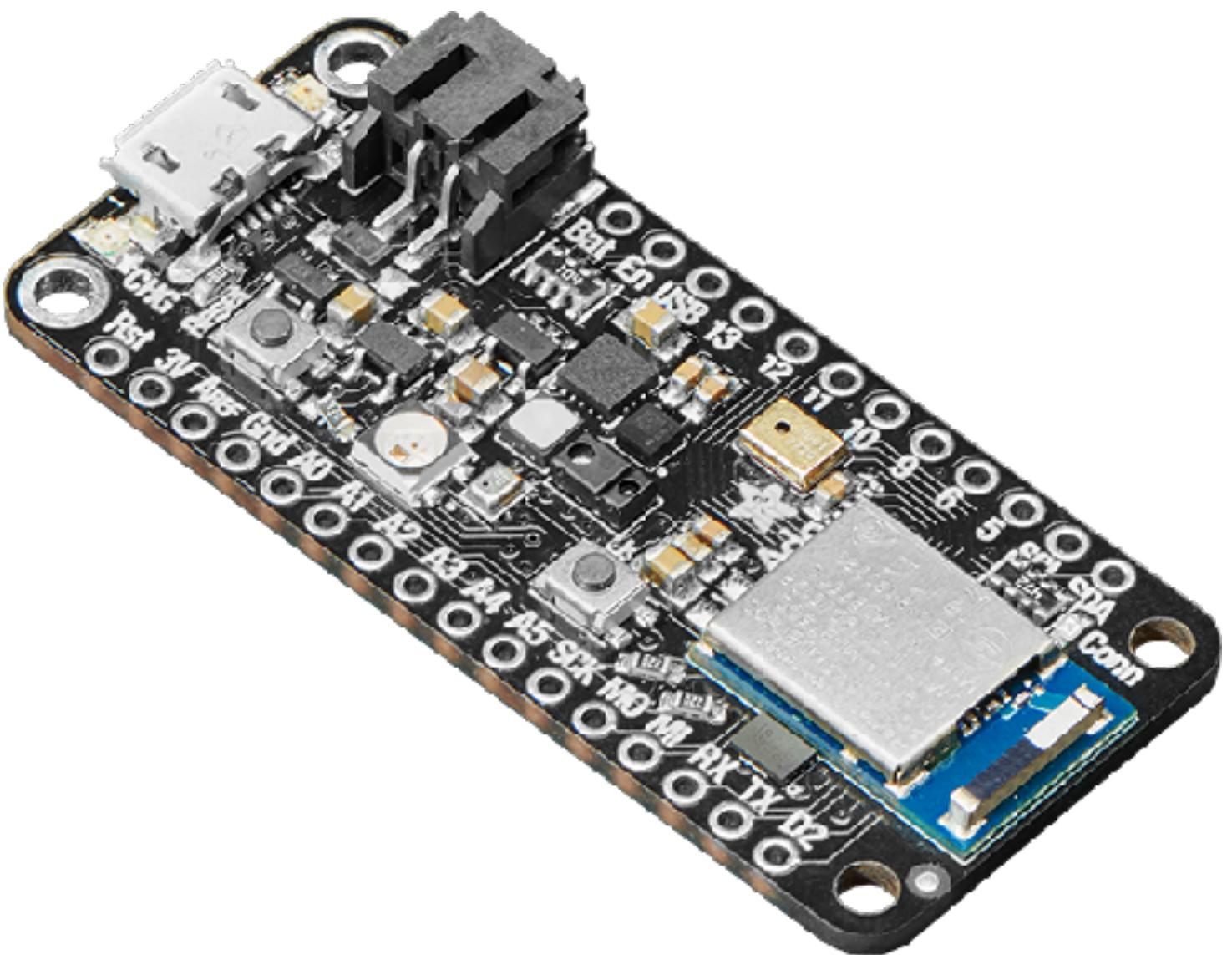
If you plug in the battery, it will power the Feather. If you connect the USB cable, it will charge the battery and power the Feather over USB



Adafruit Feather Sense – Connectivity

Built-in sensors

- Humidity
- Magnetometer
- Microphone
- Light
- Color
- Gesture
- Proximity
- Temperature
- Pressure
- Altitude
- Accelerometer
- Gyroscope
- Button



Built-in actuators

- RGB LED
- Red LED

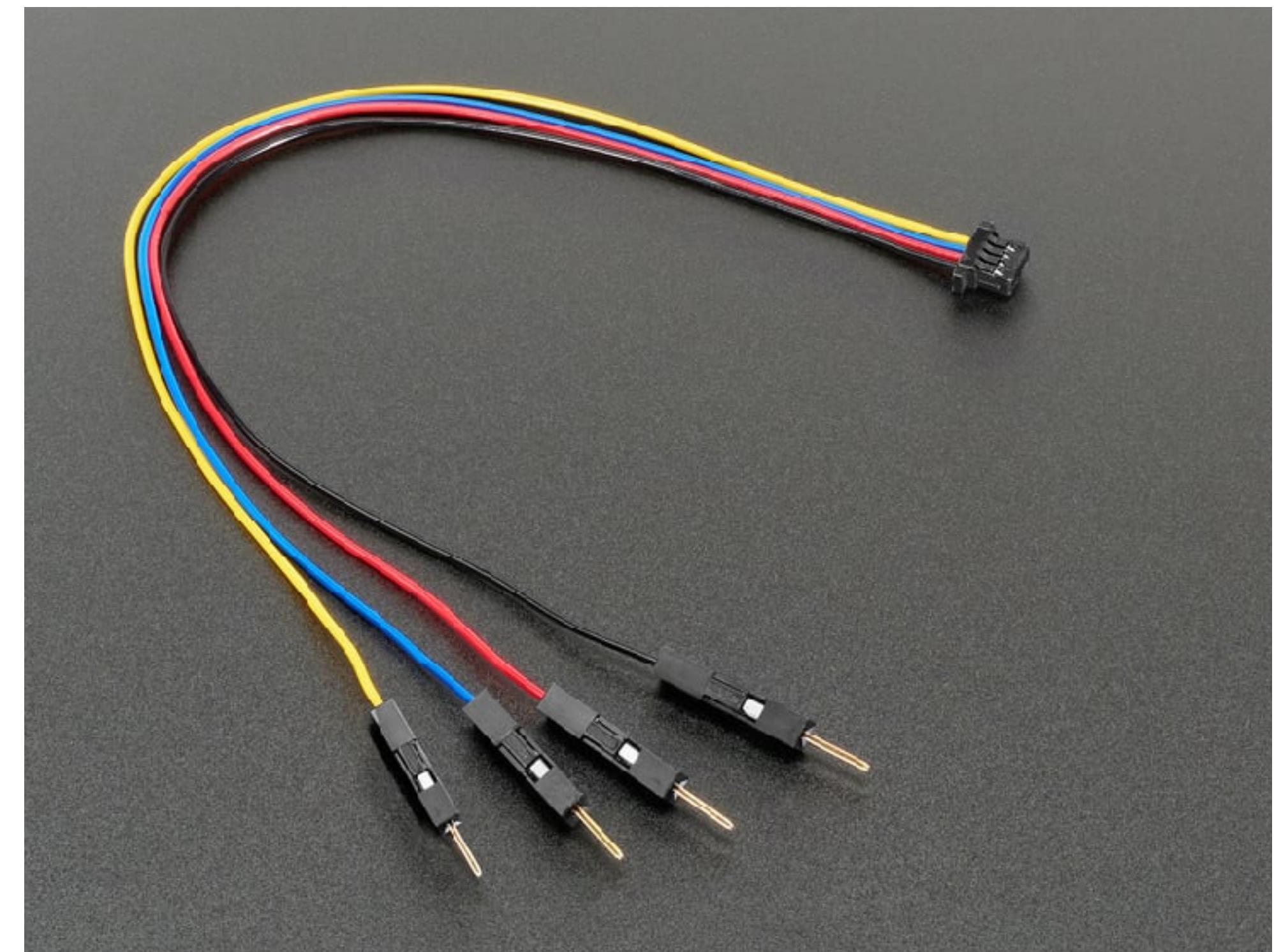
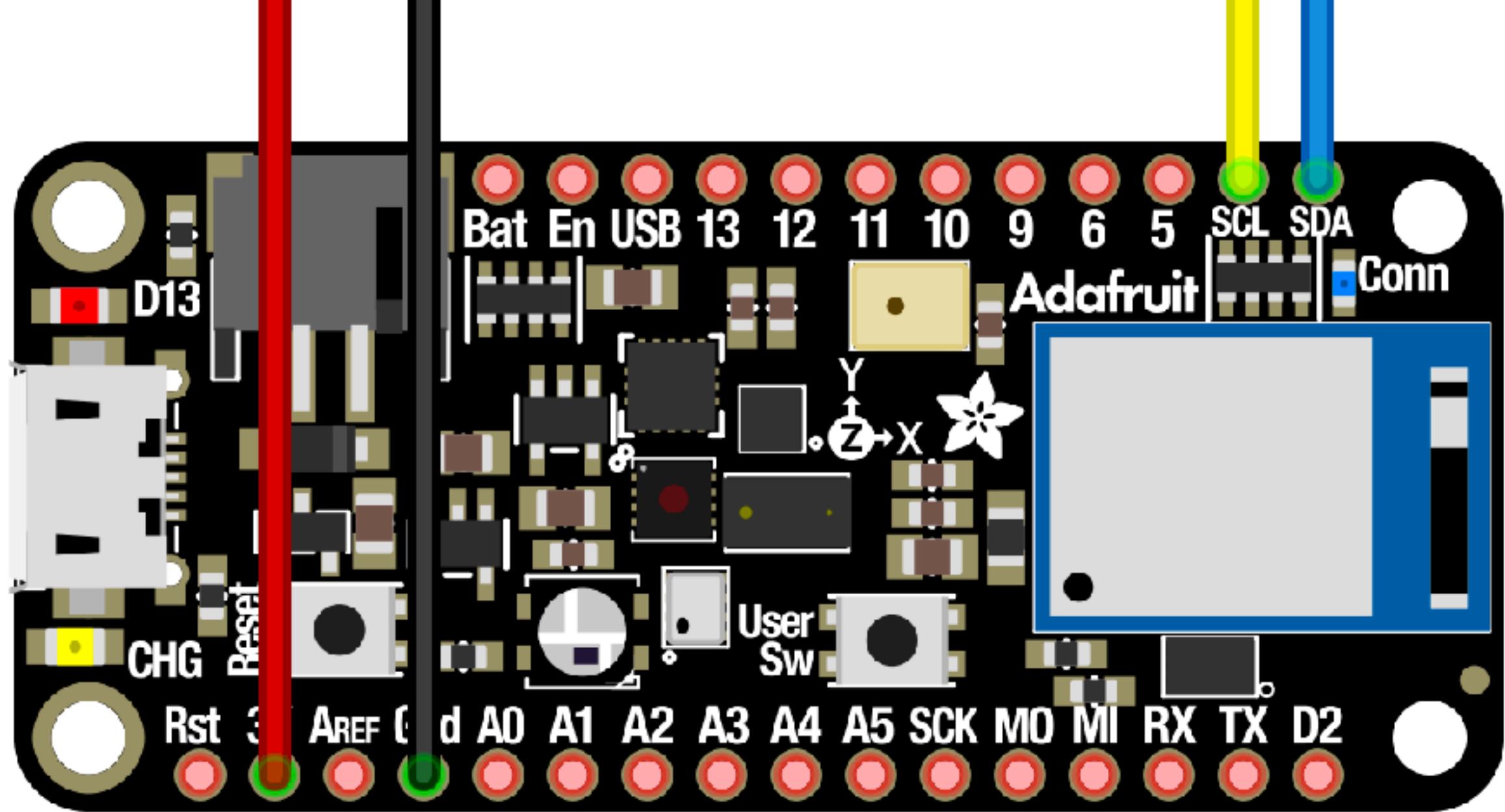
Built-in connectivity

- Bluetooth
- I2C

**I2C allows microcontrollers to talk to many
sensors and actuators at the same time.**

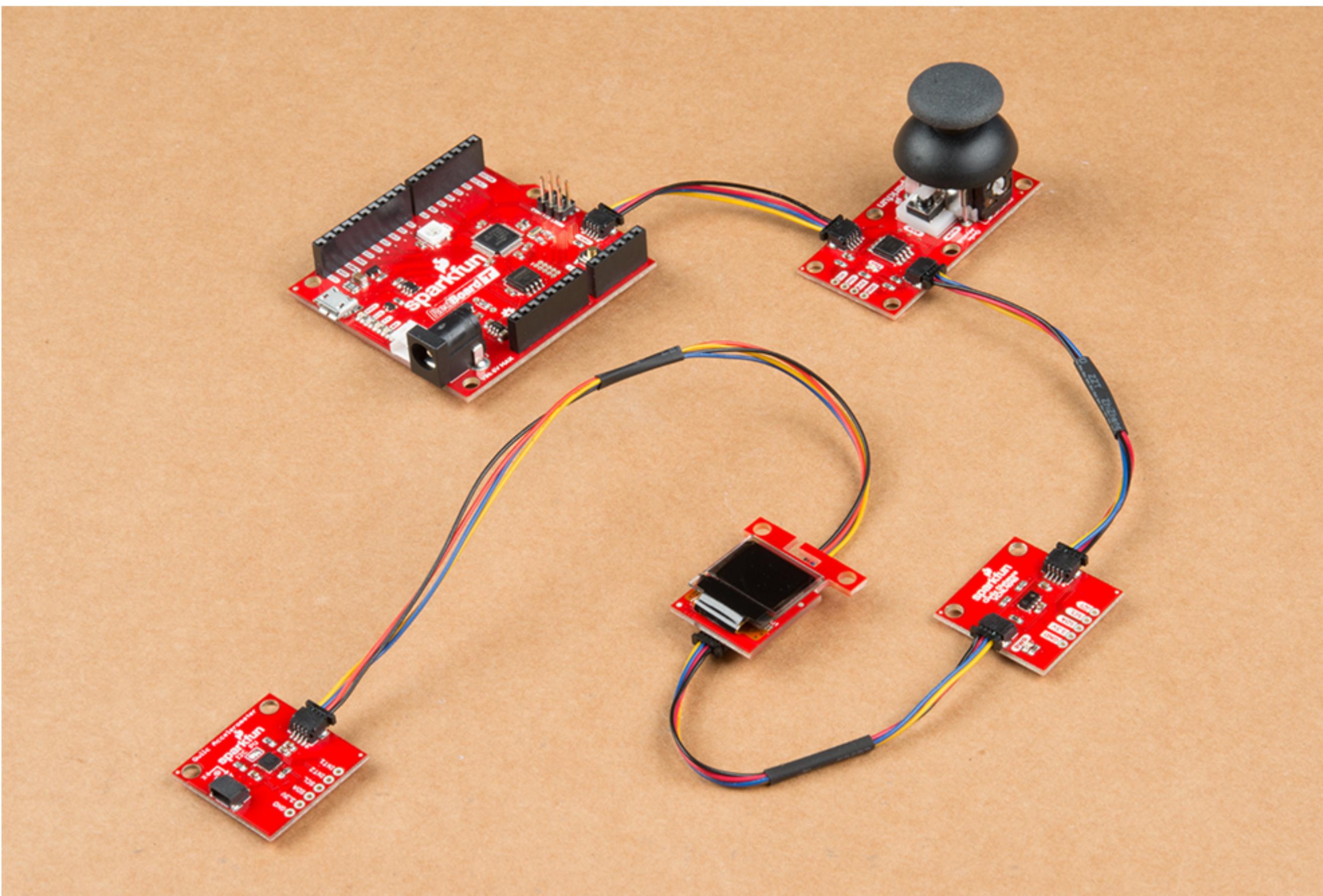
Connect to Feather Pins

- Red - Power
- Black - Ground
- Blue - SDA
- Yellow - SCL



```
module.begin(0x70);
```

Daisy-chain I2C modules using STEMMA connectors.



**To connect your microcontroller to a prototype on
your computer, there are a few options ...**

Set it up as a keyboard! (easiest)

Just let it mimic keyboard inputs on your Mac

Connect it using Serial

Web Serial, Processing, Unity, ProtoPie etc.

Connect it using Bluetooth

Good luck, happy building!

Feel free to reach out 

That's it! Questions or thoughts? 🤔