



# Digital Development Workshop

Spring 2018 · Beeps and Bloops

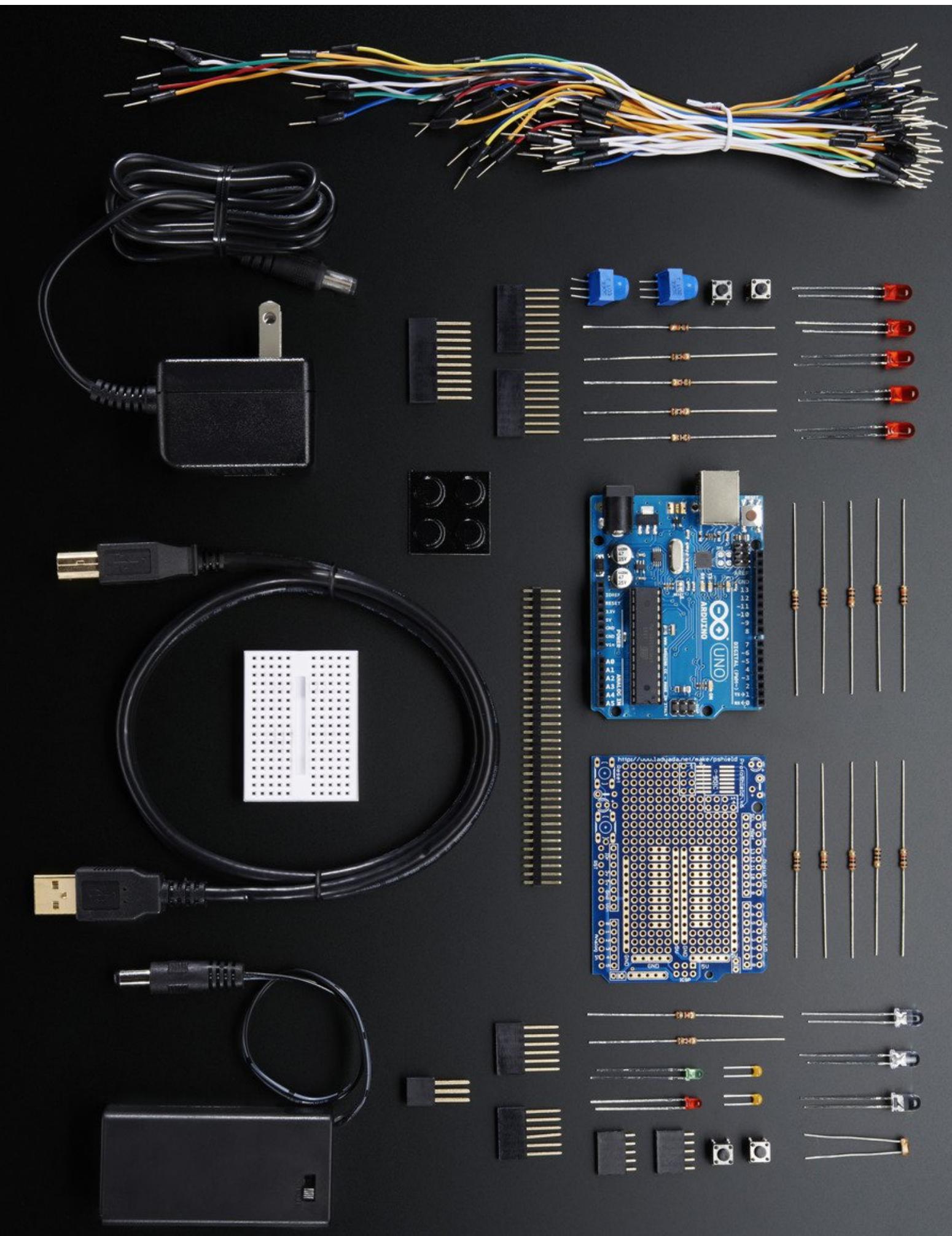
```

// Returns a point representing the exact center of the cell.
center: function() {
    var x_avg = 0.0;
    for (var i=0 ; i<this.walls.length ; i++) {
        x_avg += this.walls[i].points[0].x;
        x_avg += this.walls[i].points[1].x;
    }
    x_avg /= this.walls.length * 2;
    var y_avg = 0.0;
    for (var i=0 ; i<this.walls.length ; i++) {
        y_avg += this.walls[i].points[0].y;
        y_avg += this.walls[i].points[1].y;
    }
    y_avg /= this.walls.length * 2;
    return(new Point(x_avg,y_avg));
},

// Return a list of all vertices of a cell, in order
vertices: function() {
    var points = [];
    // I need to pick only a single point from each wall, and
    // will be the point that isn't in the next wall.
    for (var i=0 ; i<this.walls.length ; i++) {
        var this_wall = this.walls[i];
        var next_wall = this.walls[i+1];
        if (!next_wall) { next_wall = this.walls[0]; }
        if (this_wall.points[0] != next_wall.points[0] && this_wall.points[0] != next_wall.points[1])
            points.push(this_wall.points[0]);
        else
            points.push(this_wall.points[1]);
    }
    return points;
},

// Return a list of neighboring cells
neighbors: function() {
    var ret = new Array;
    for (var i=0 ; i < this.walls.length ; i++) {
        var this_wall = this.walls[i];

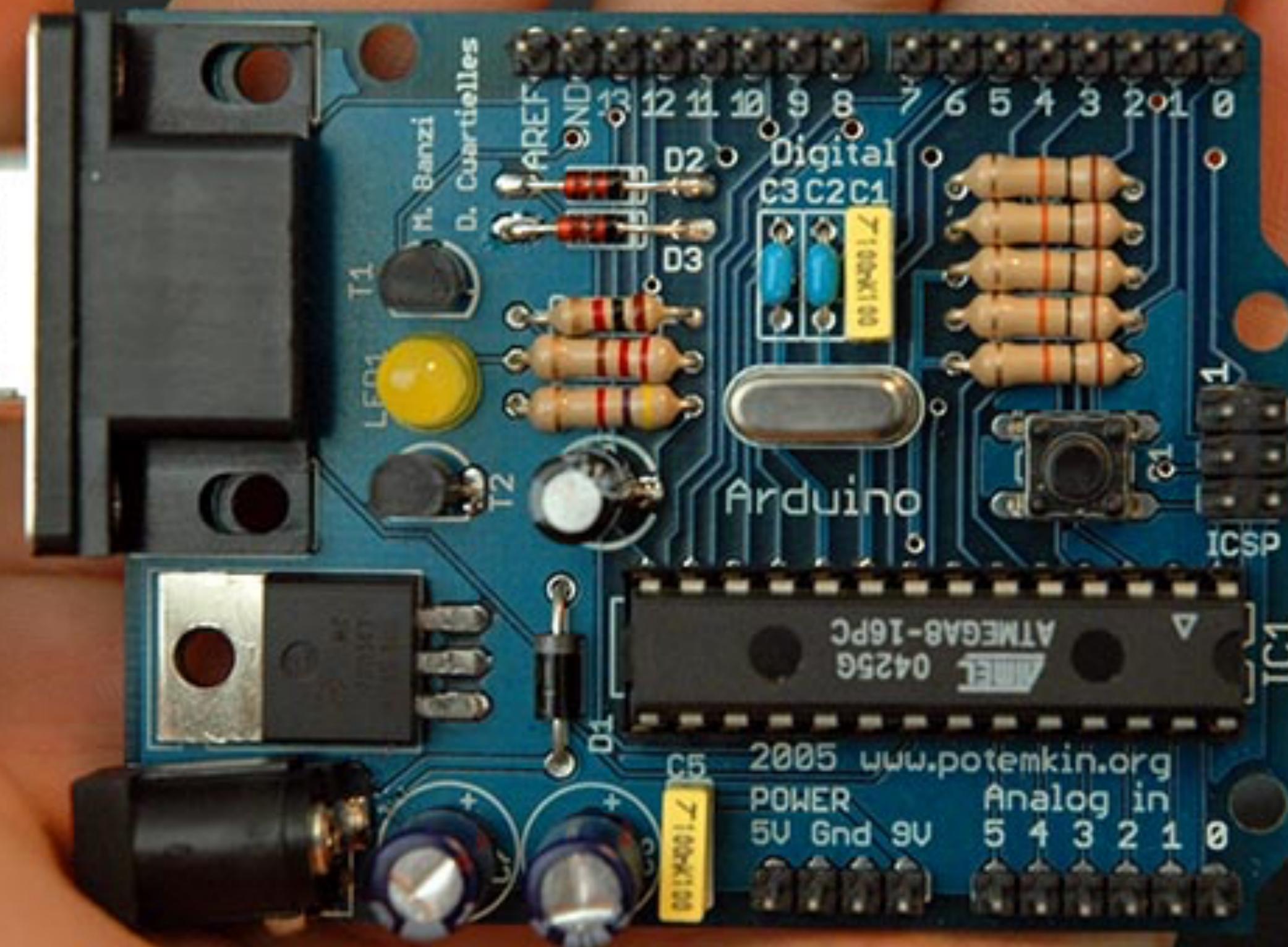
```



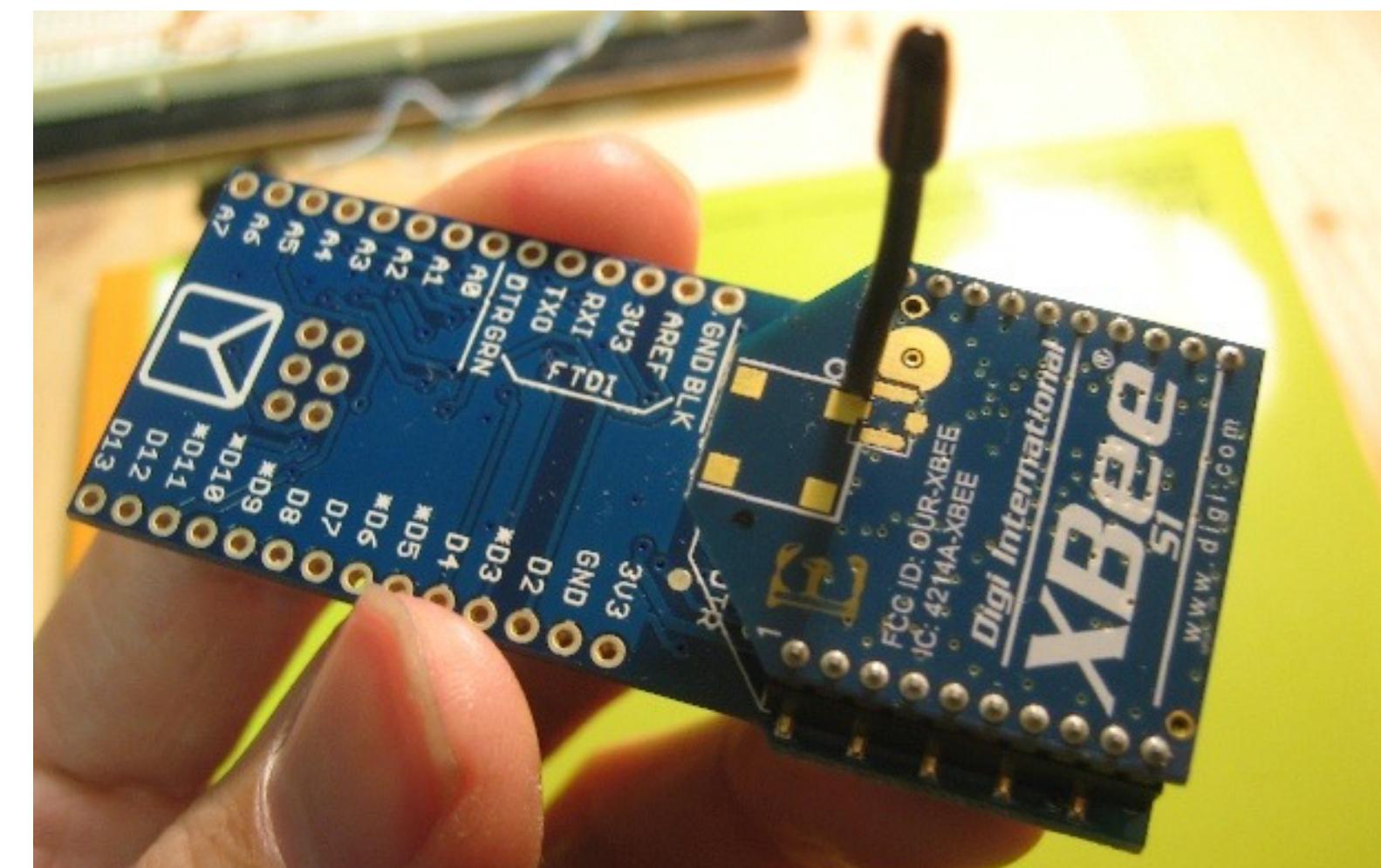
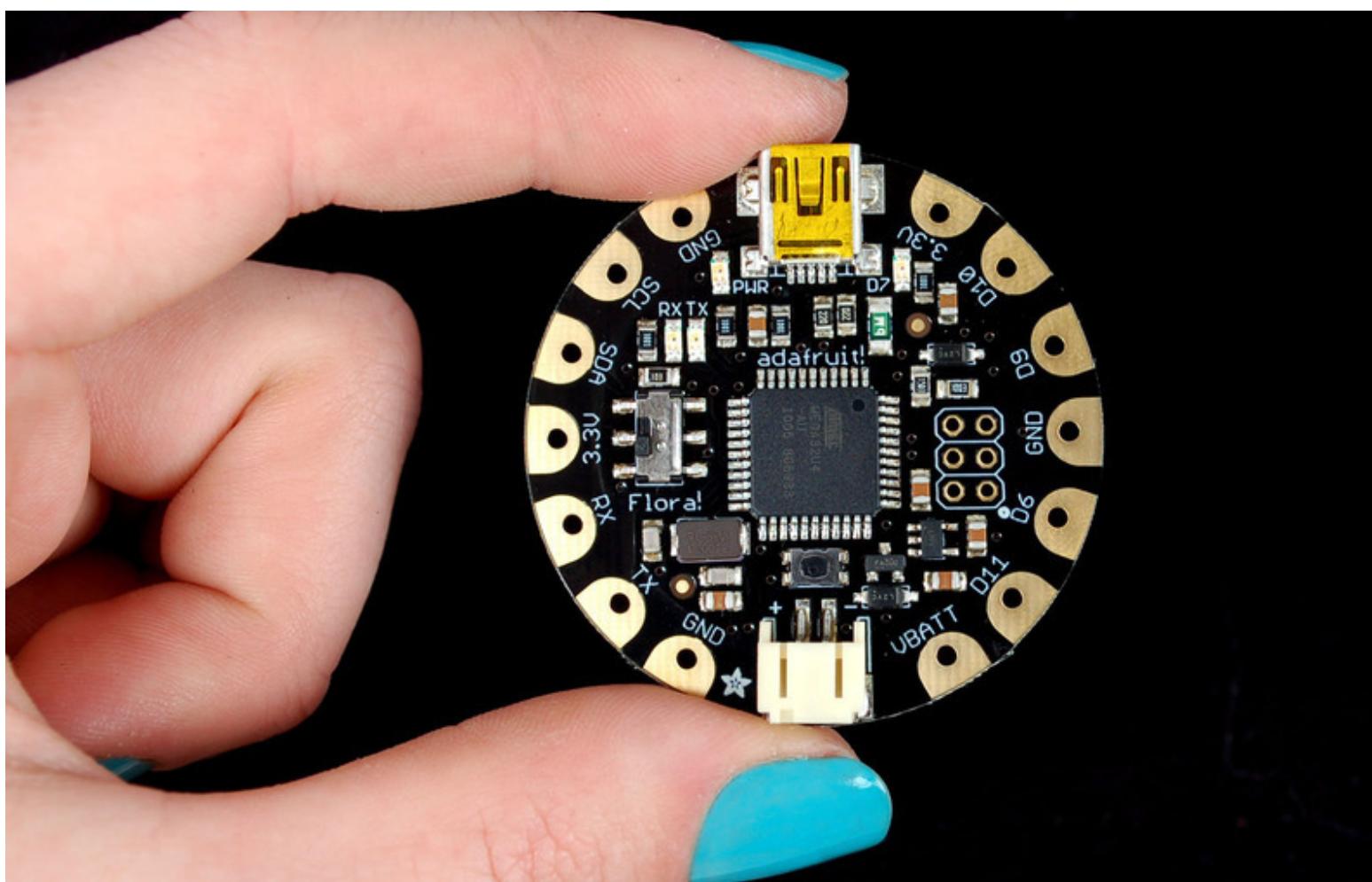
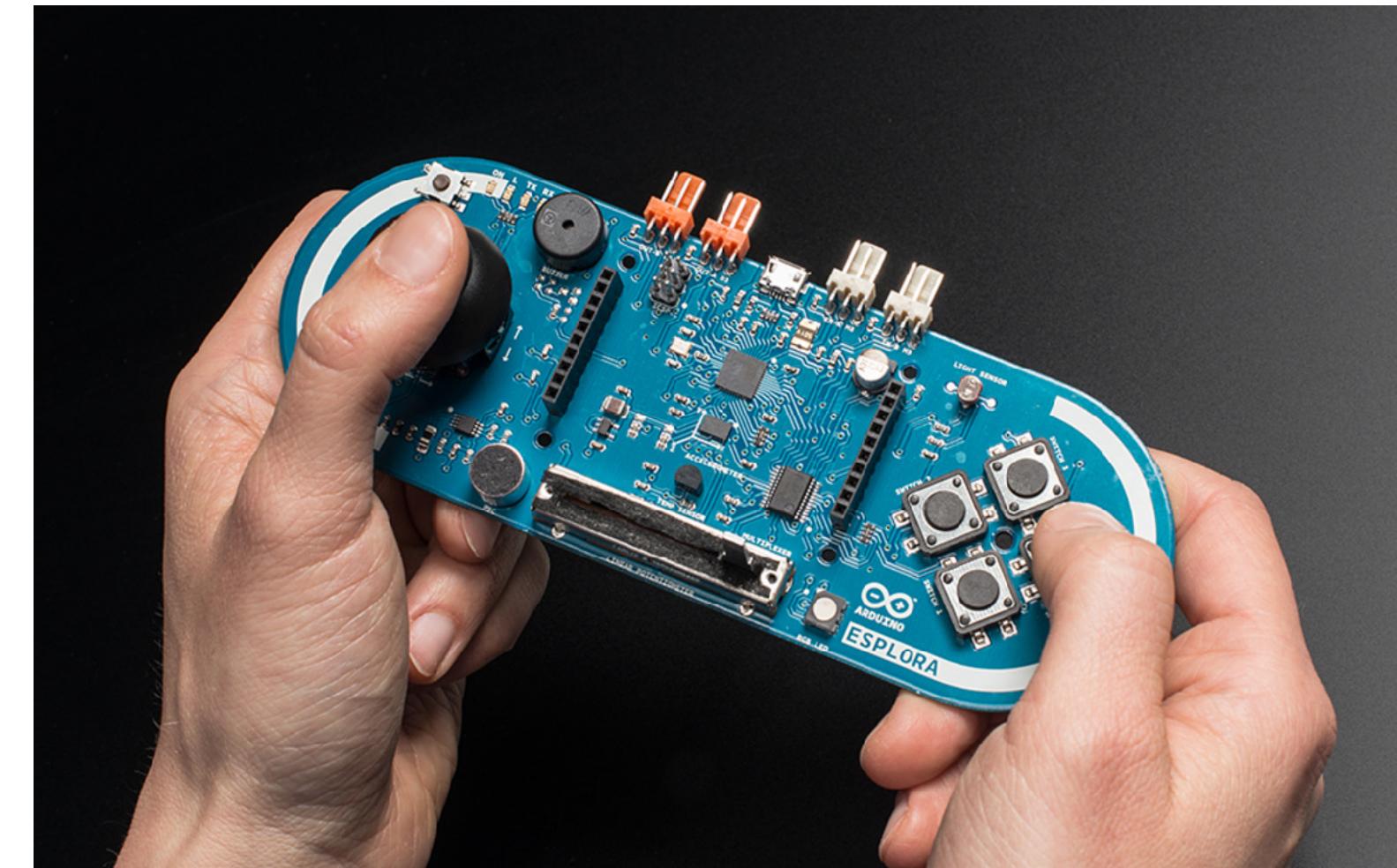
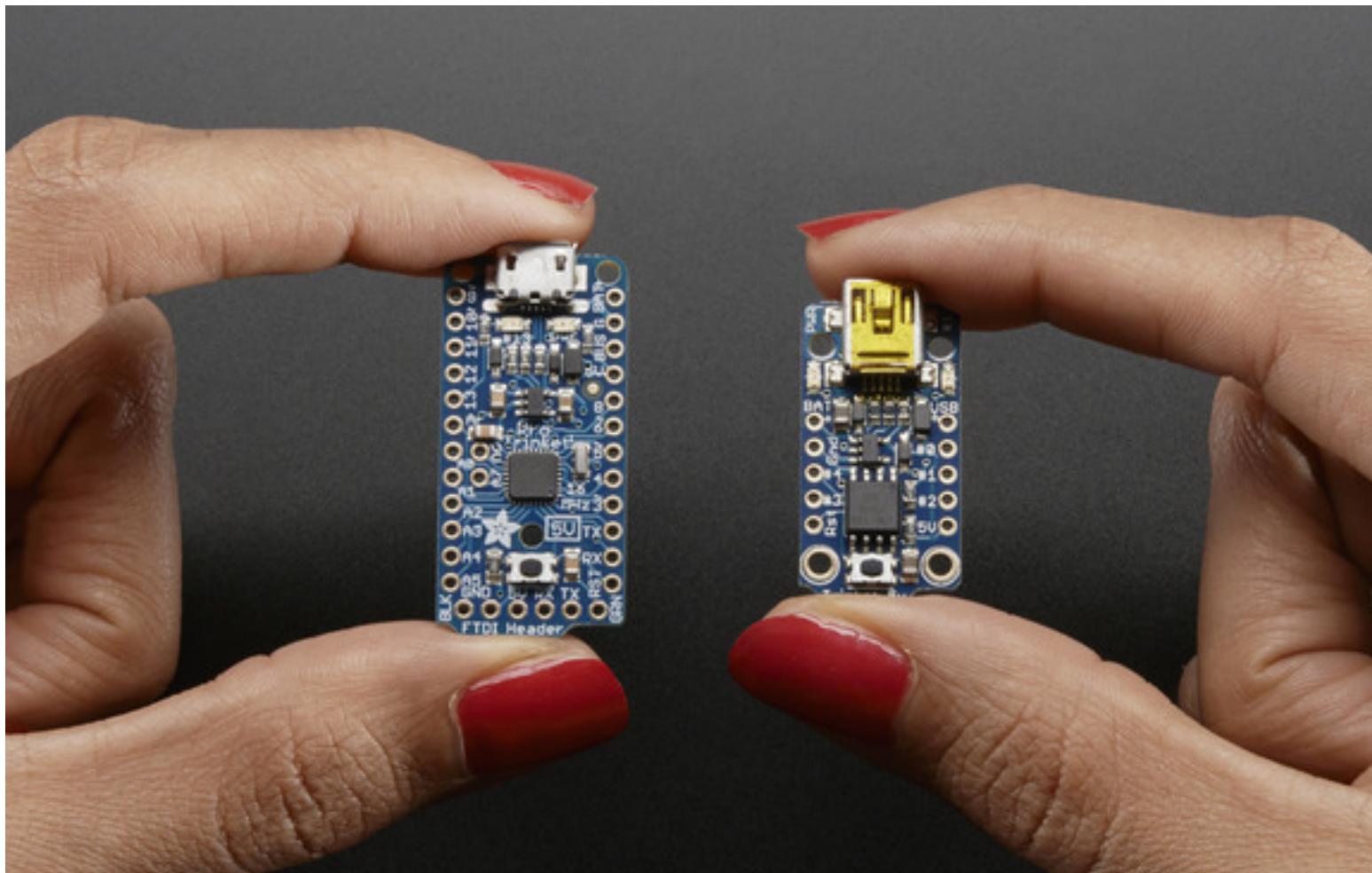


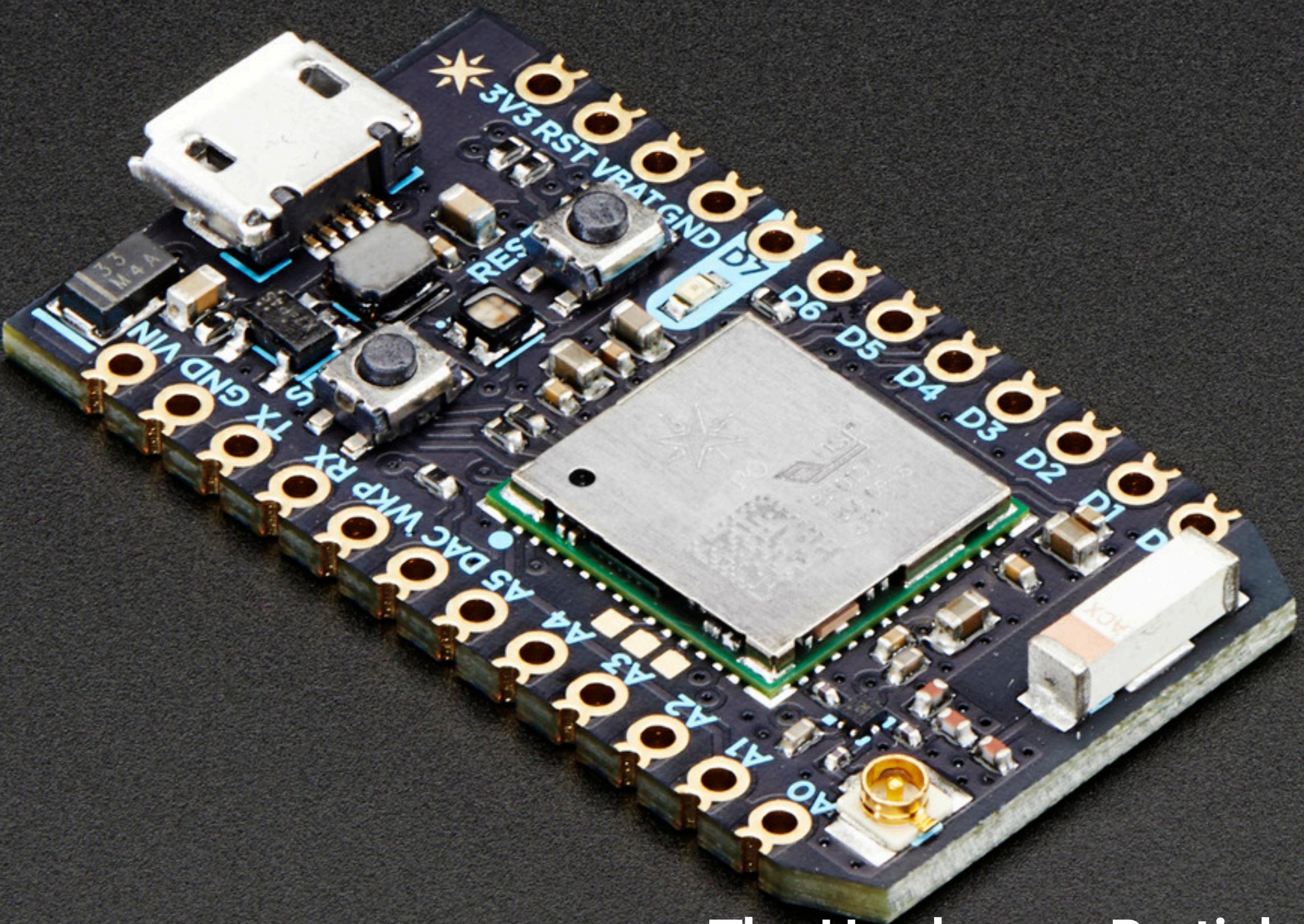
```
void loop()
{
    //MCU Task
    for (NUM_I
    {
        if (
            fr
```



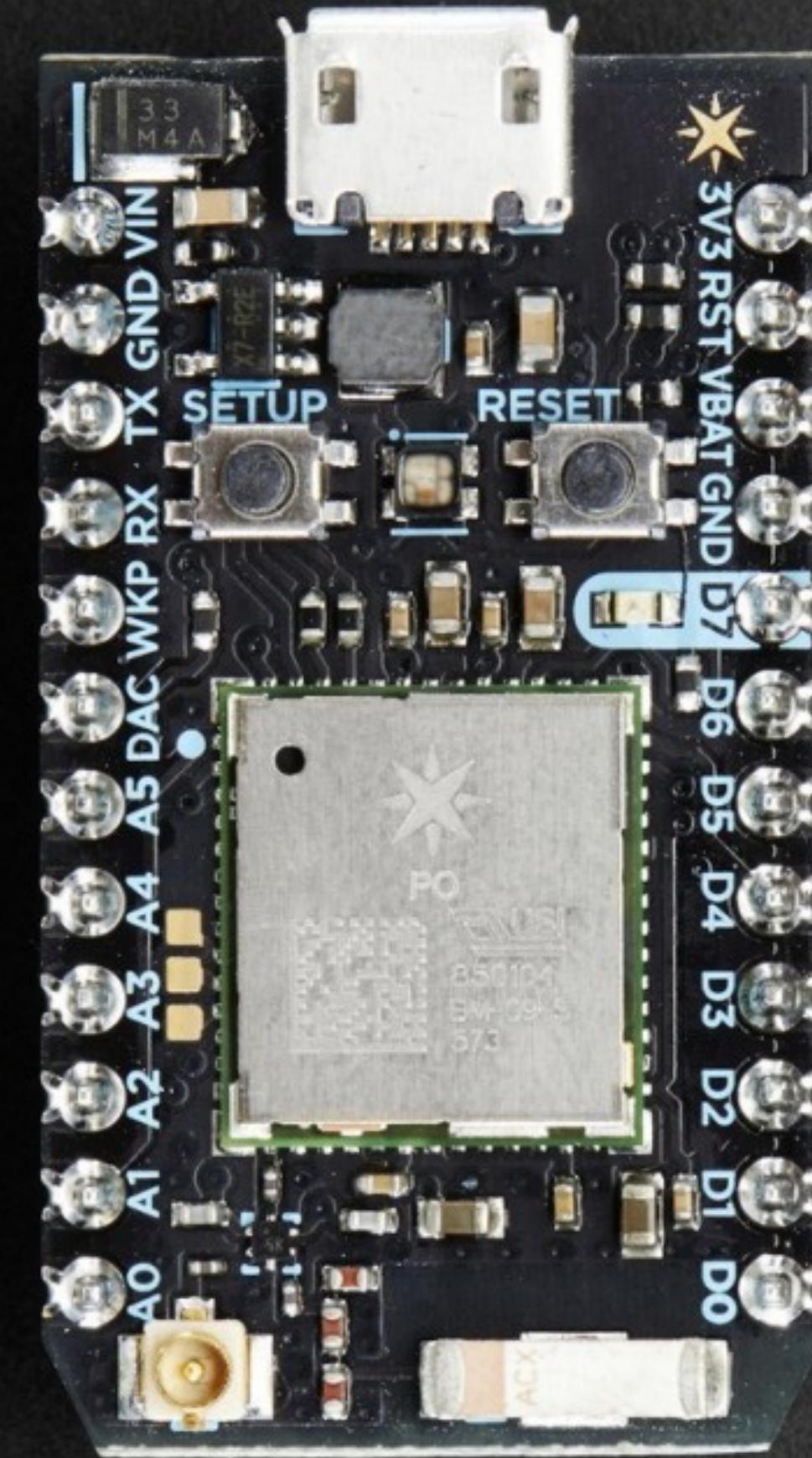
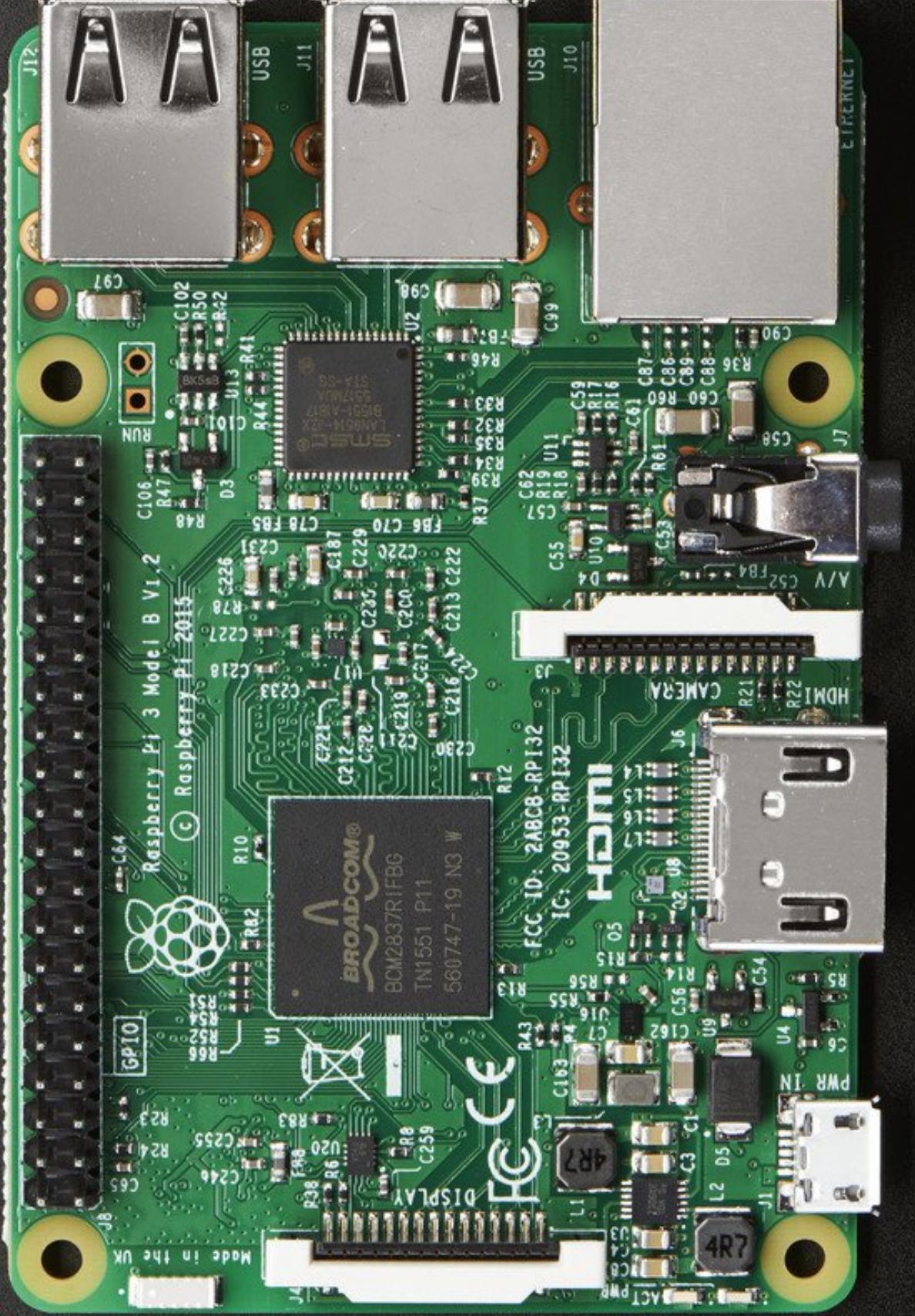


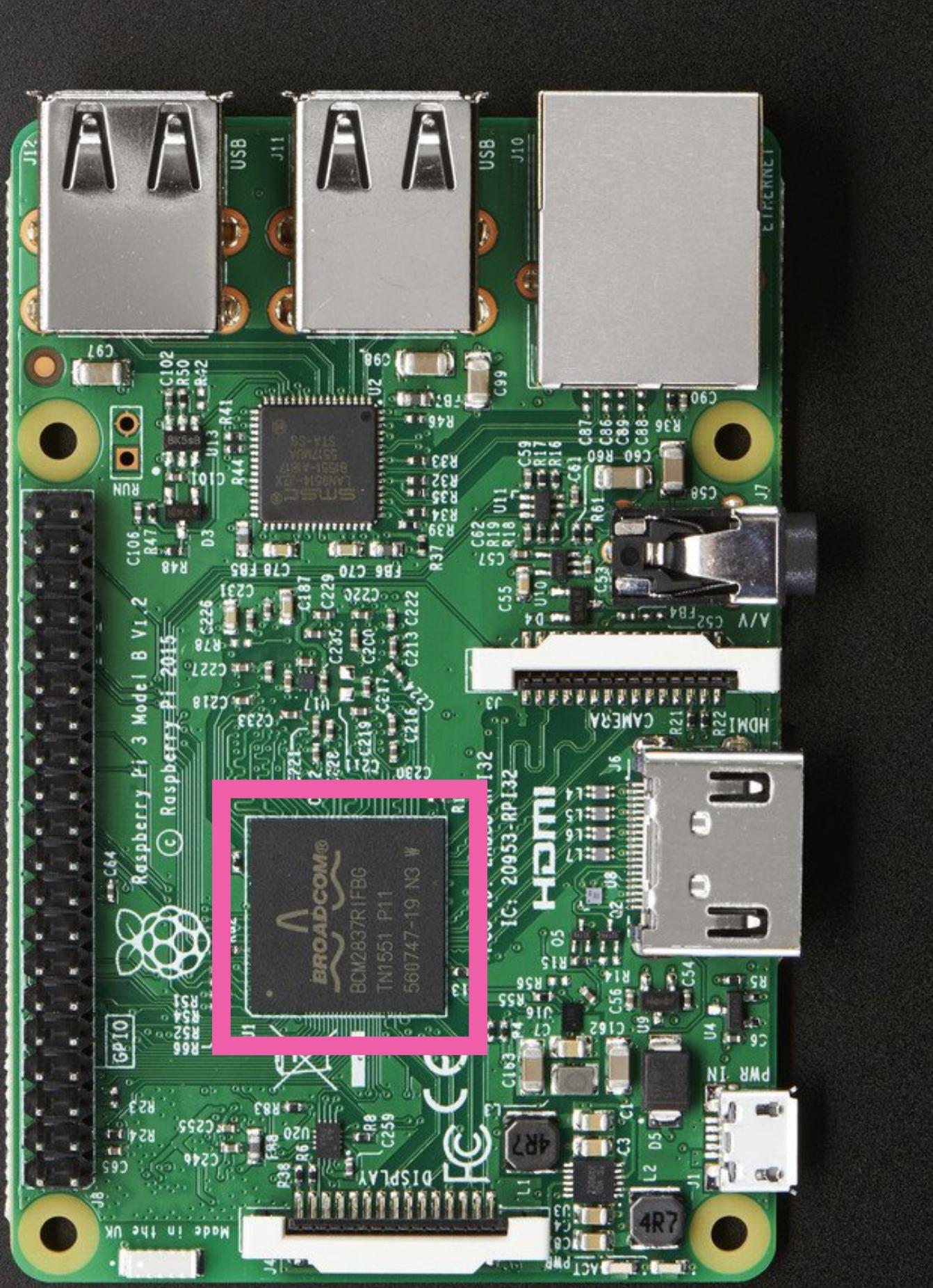
The Hardware · Arduino Platform  
\$10 Embeddable Brain



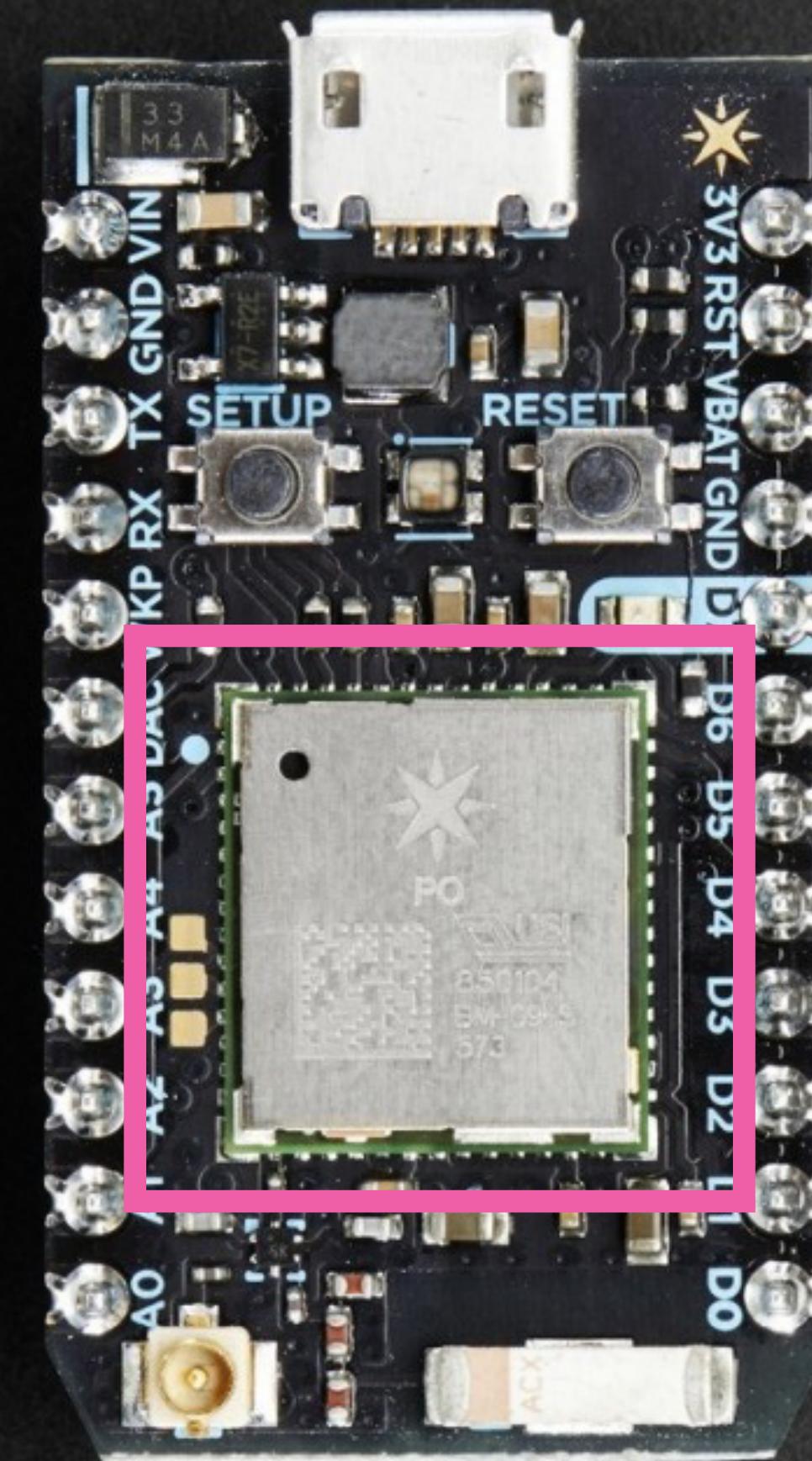


The Hardware · Particle Platform  
\$10 Embeddable Brain

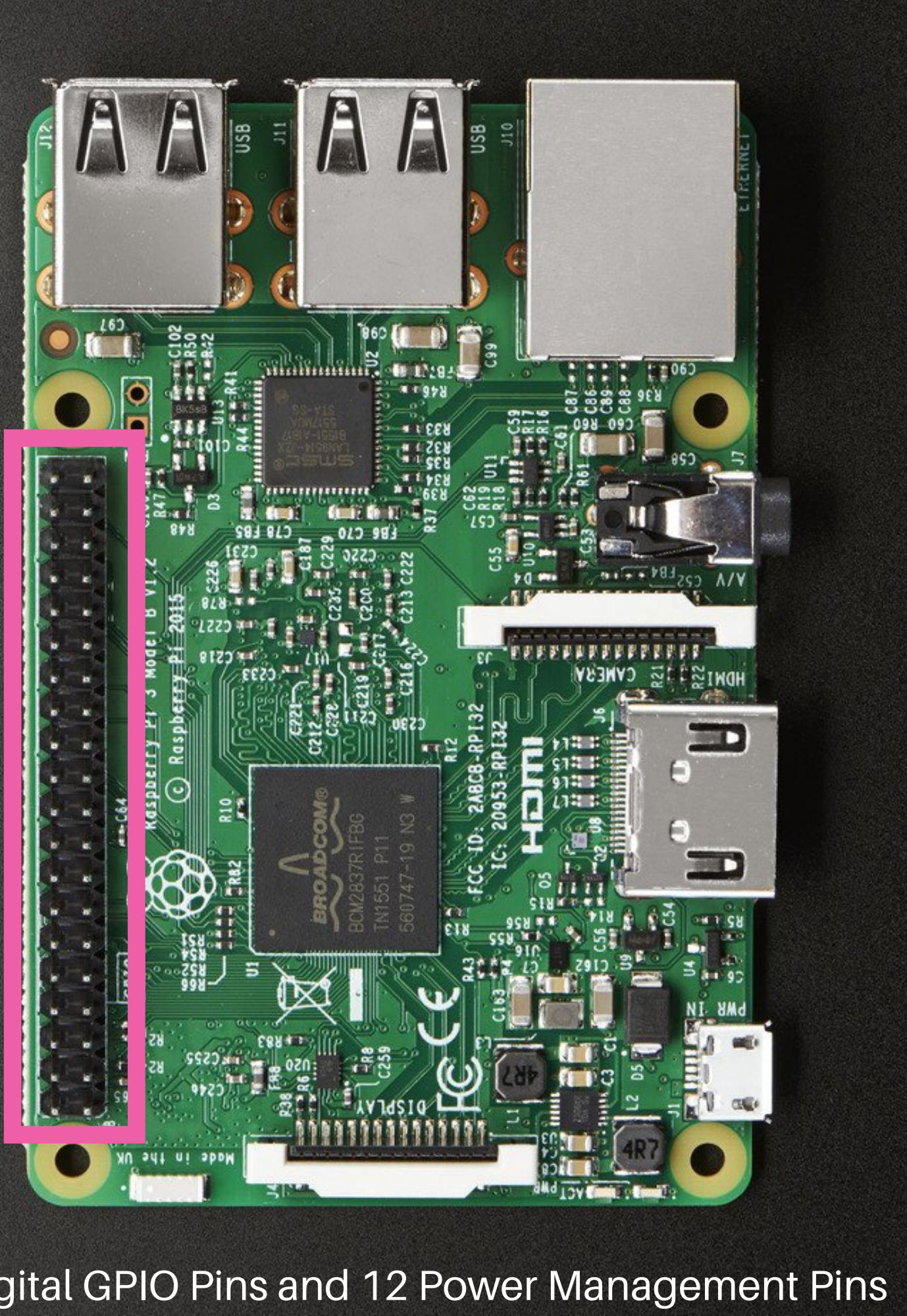




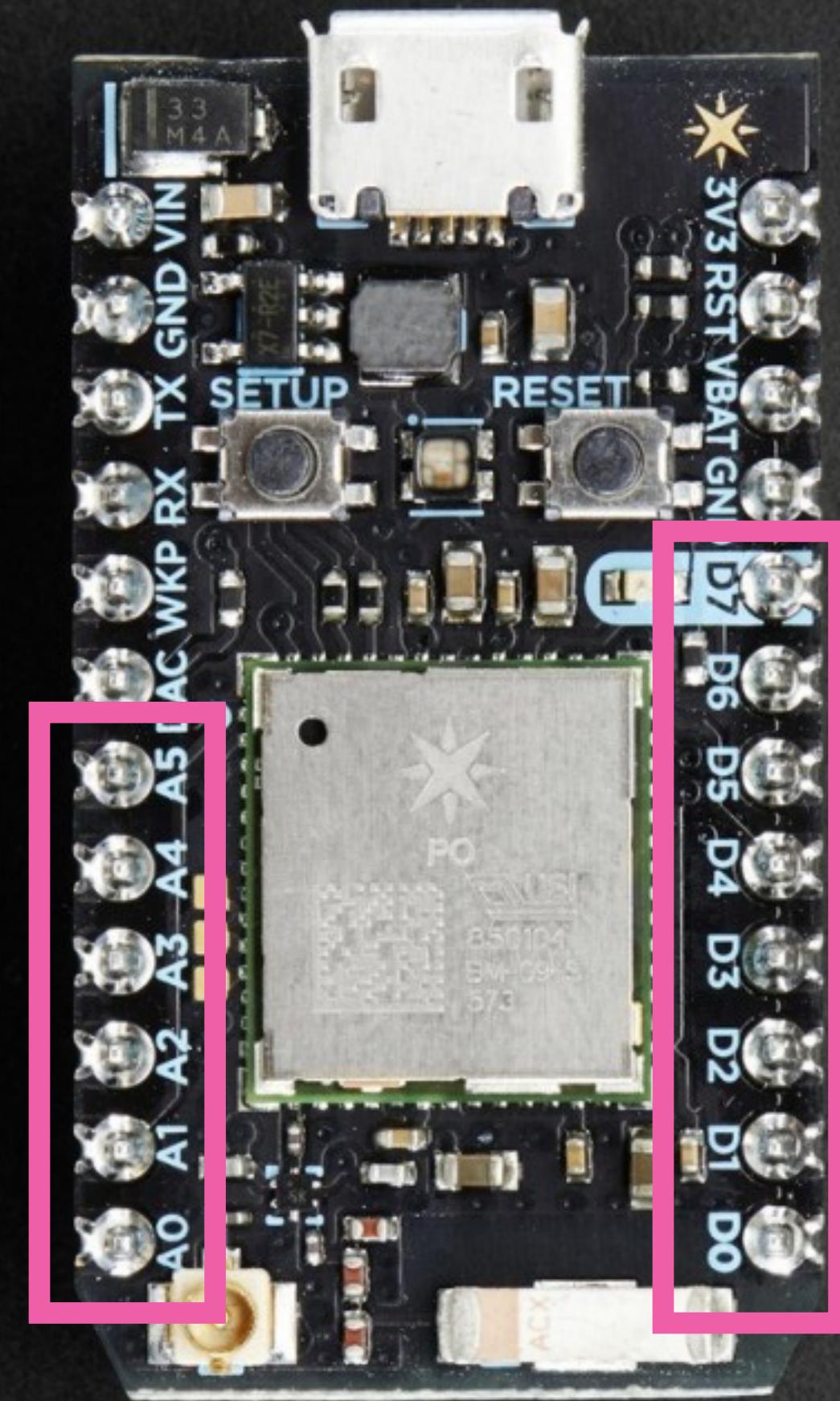
## 1GHz Processor with 512MB of RAM



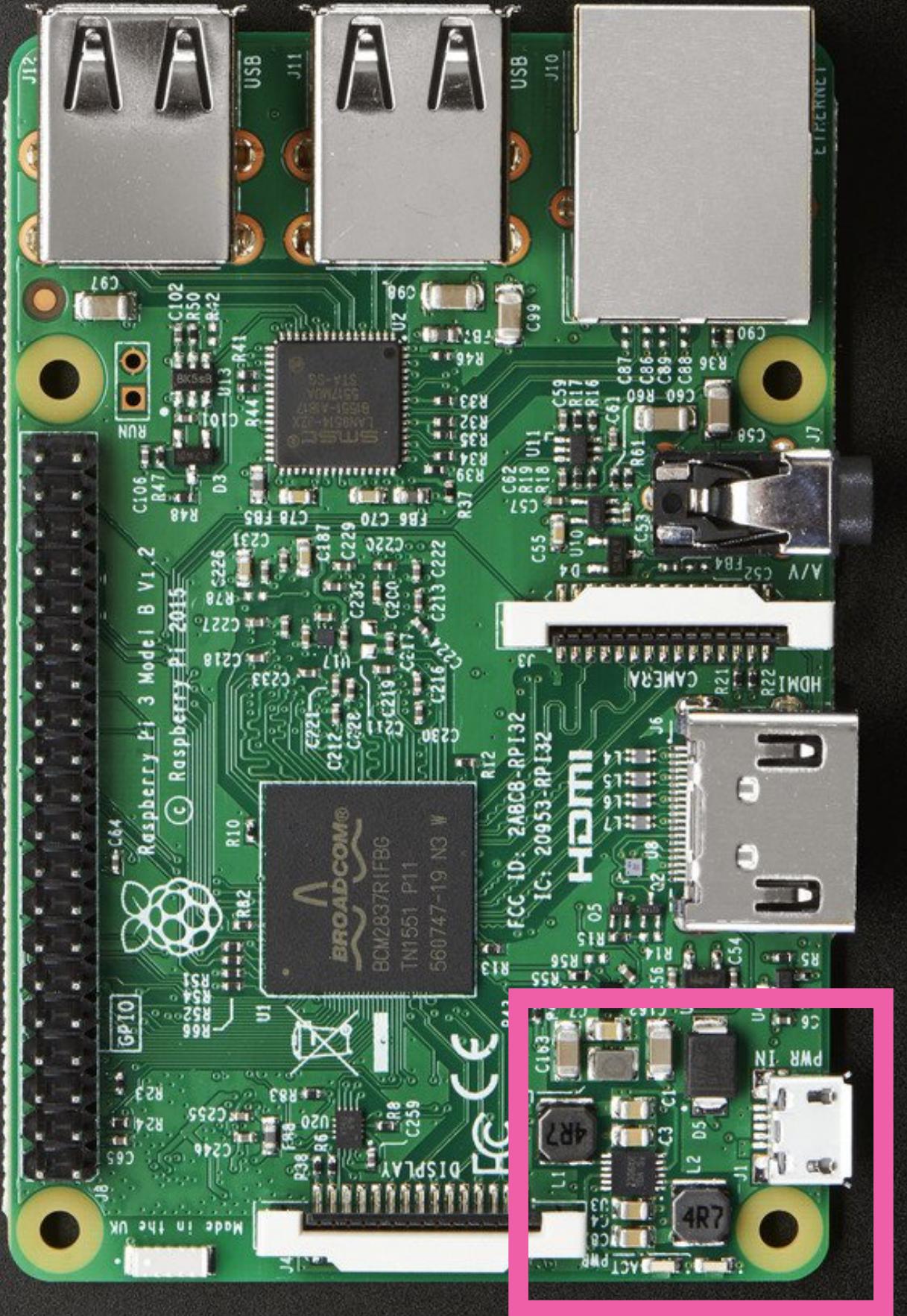
120 MHz ARM Processor with 128 MB of RAM



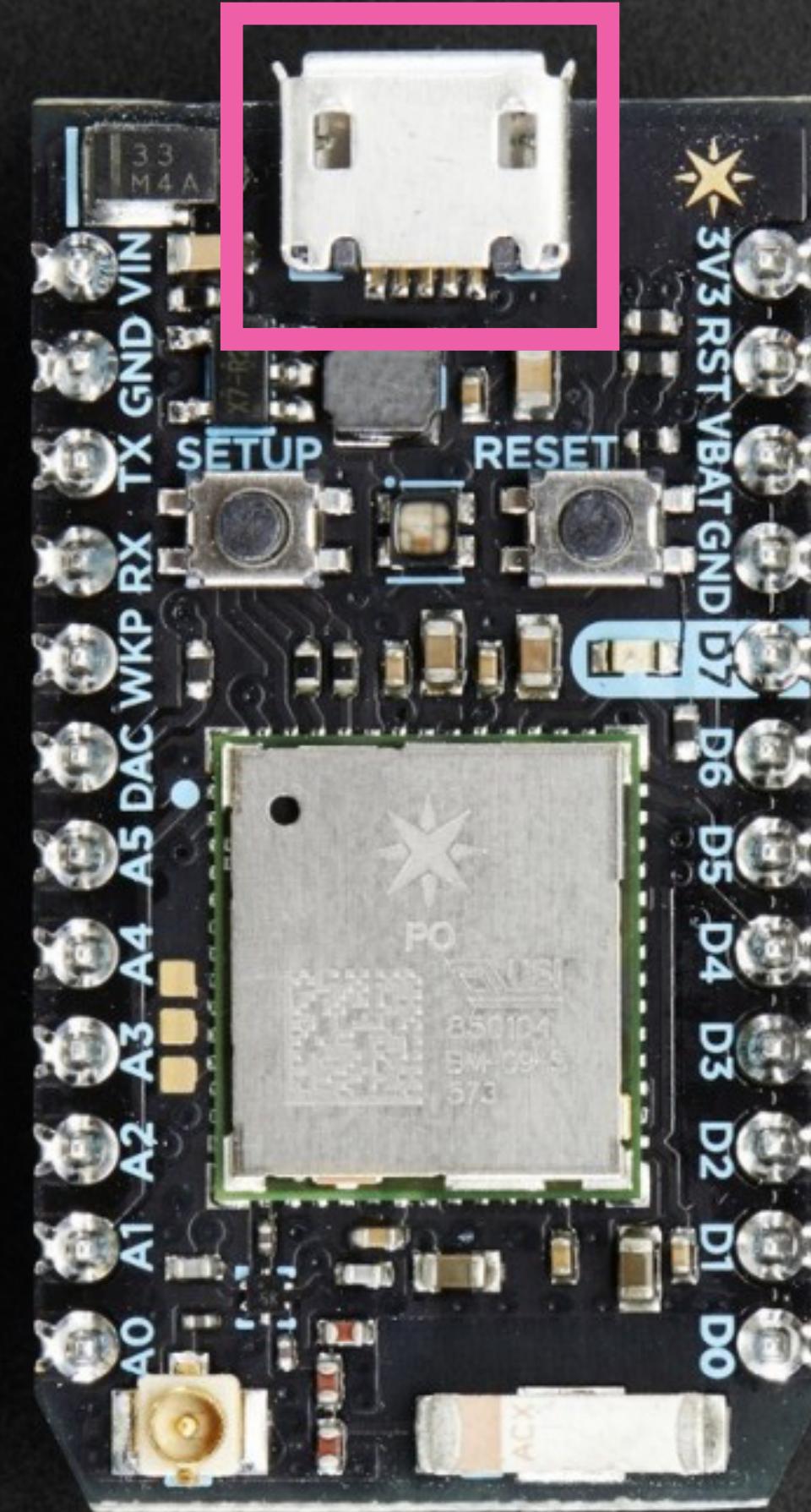
28 Digital GPIO Pins and 12 Power Management Pins



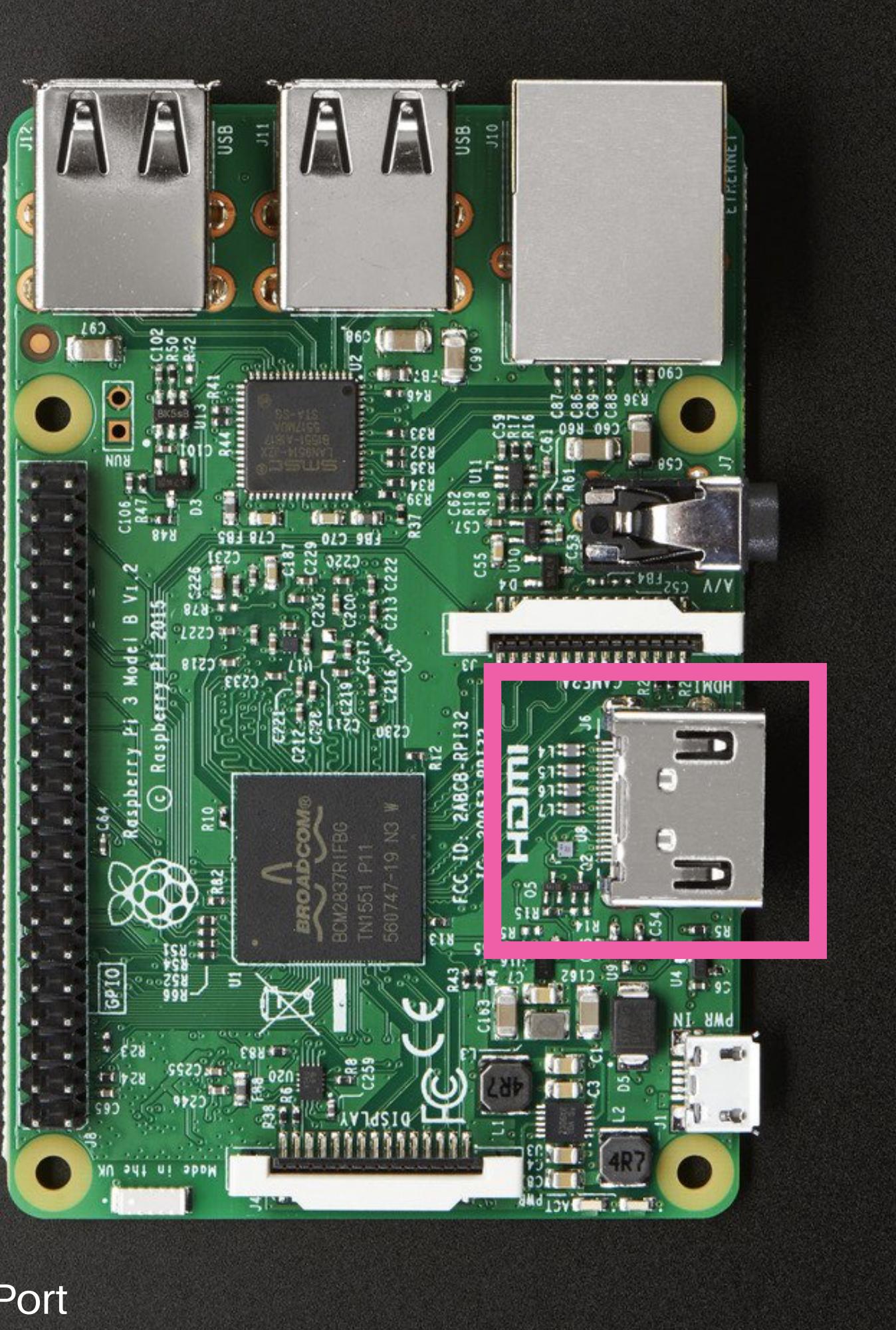
8 Digital IO Pins, 6 Analog IO Pins, and 10 Utility Pins



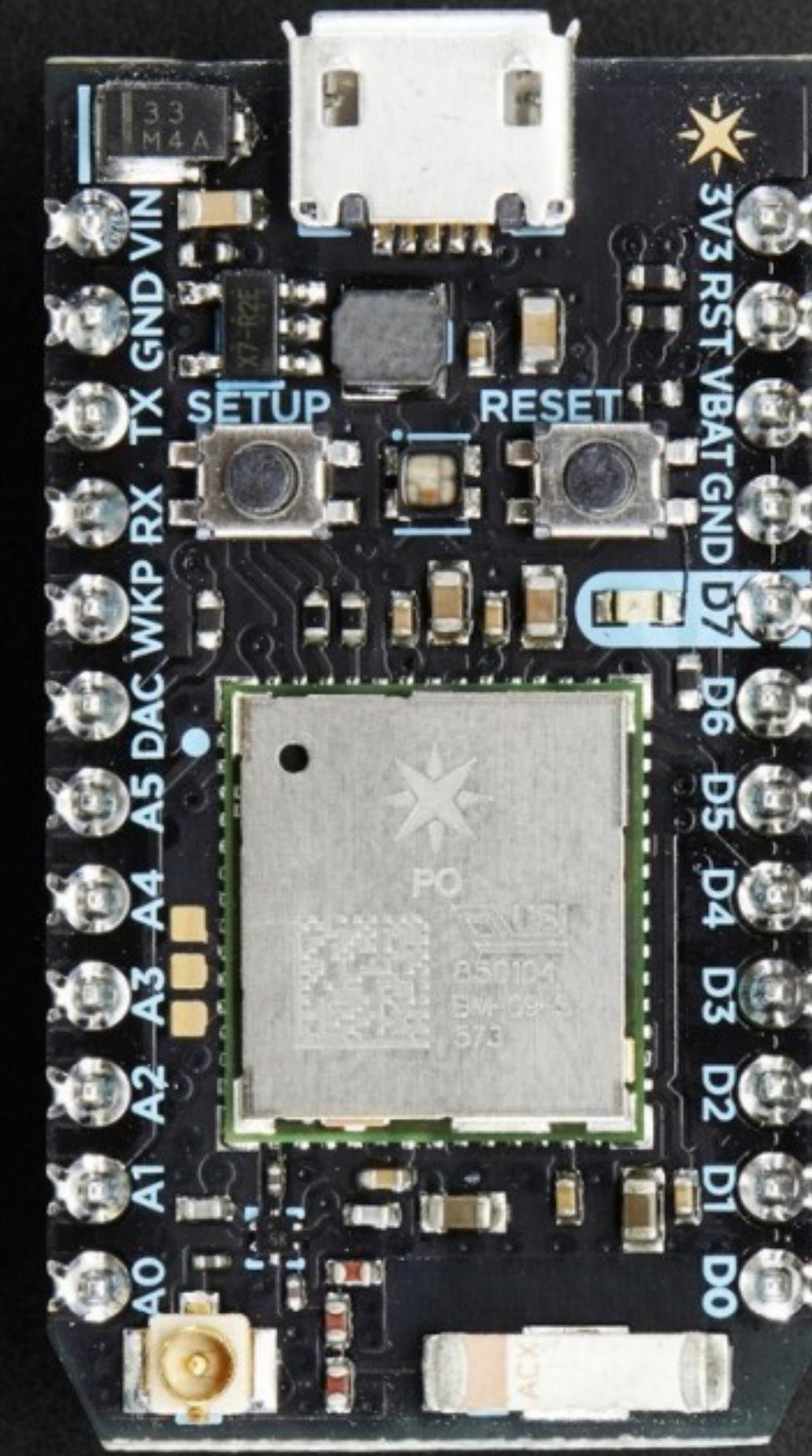
Micro USB Power Source

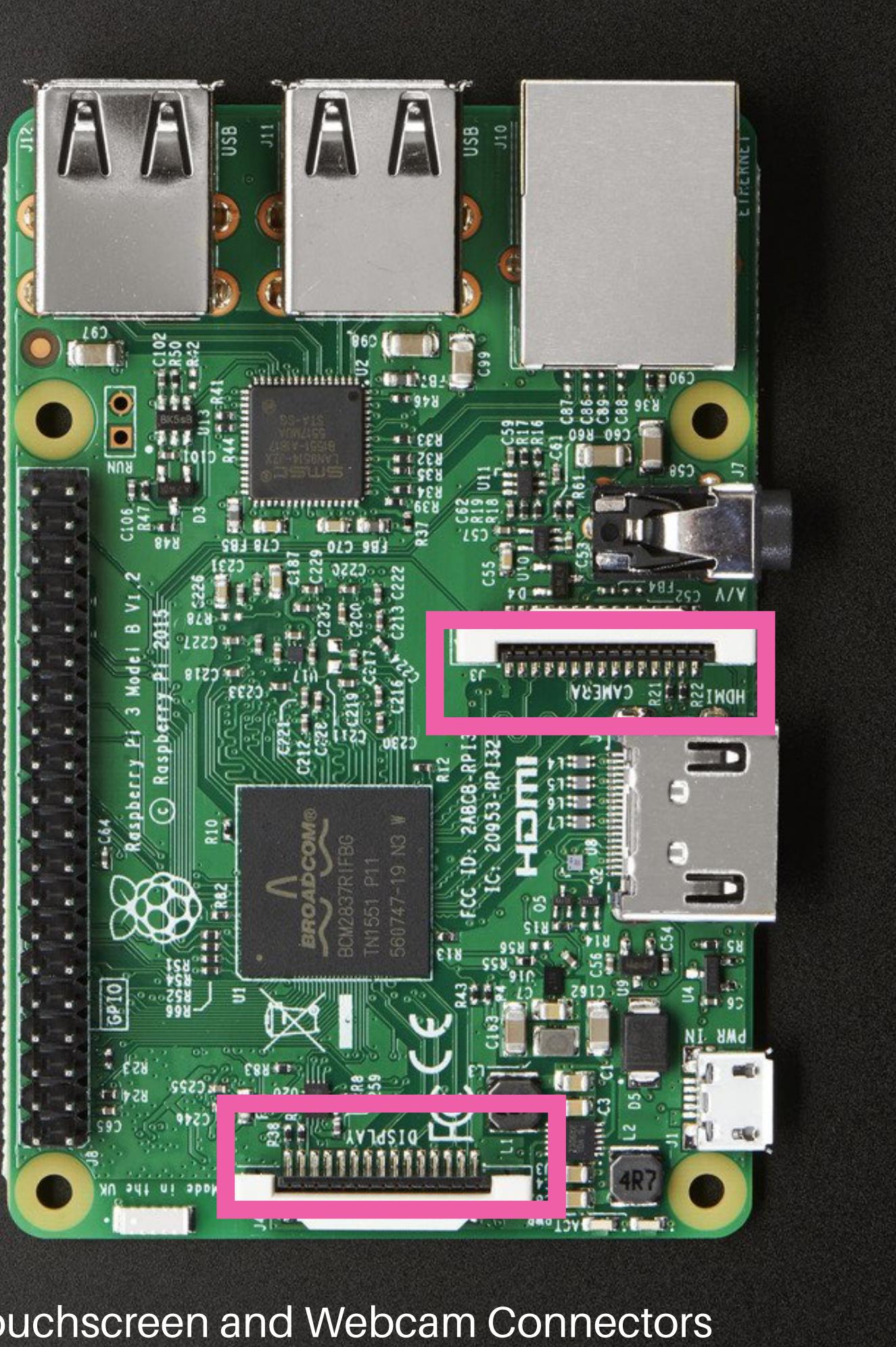


Micro USB Power Source

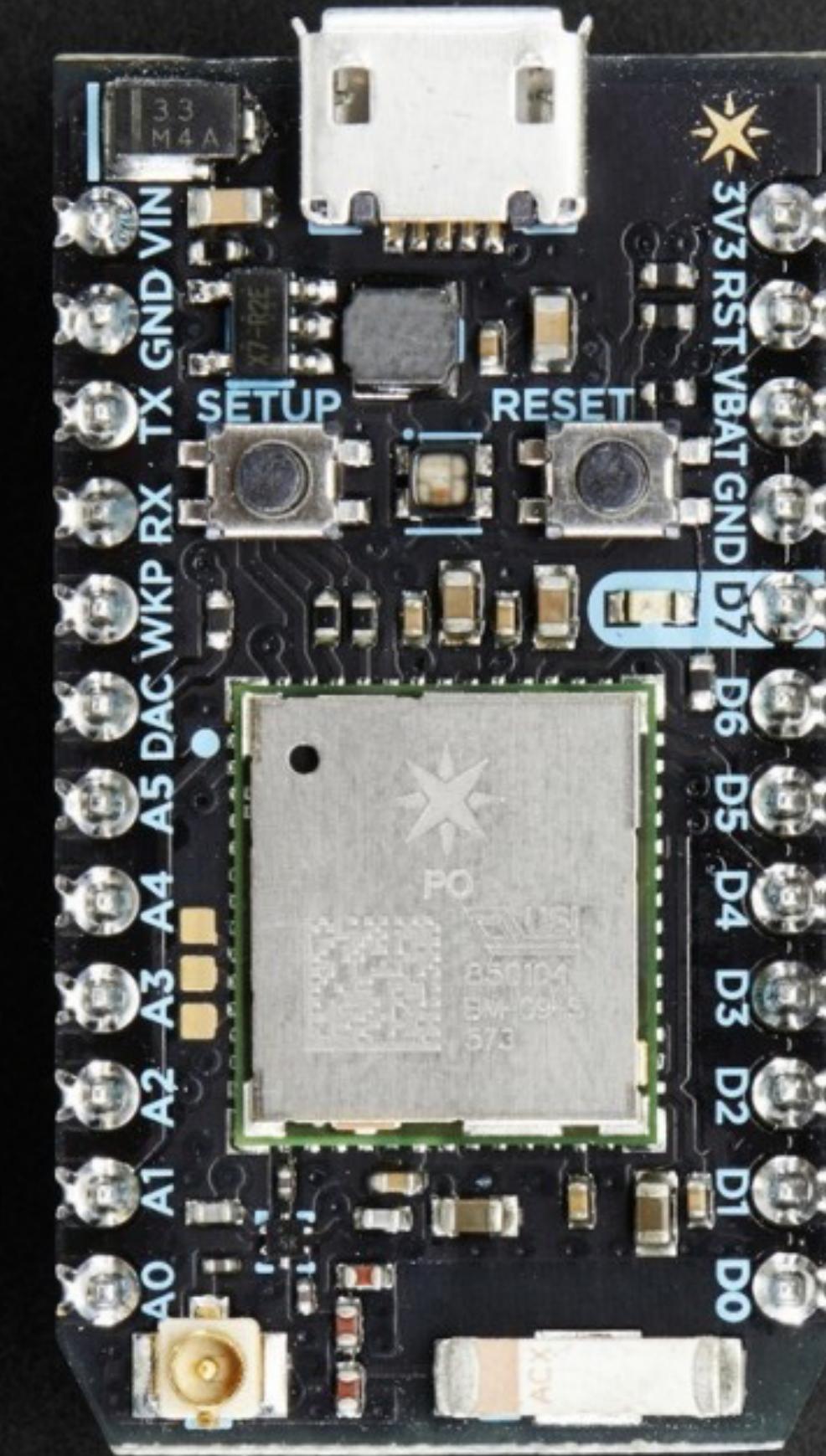


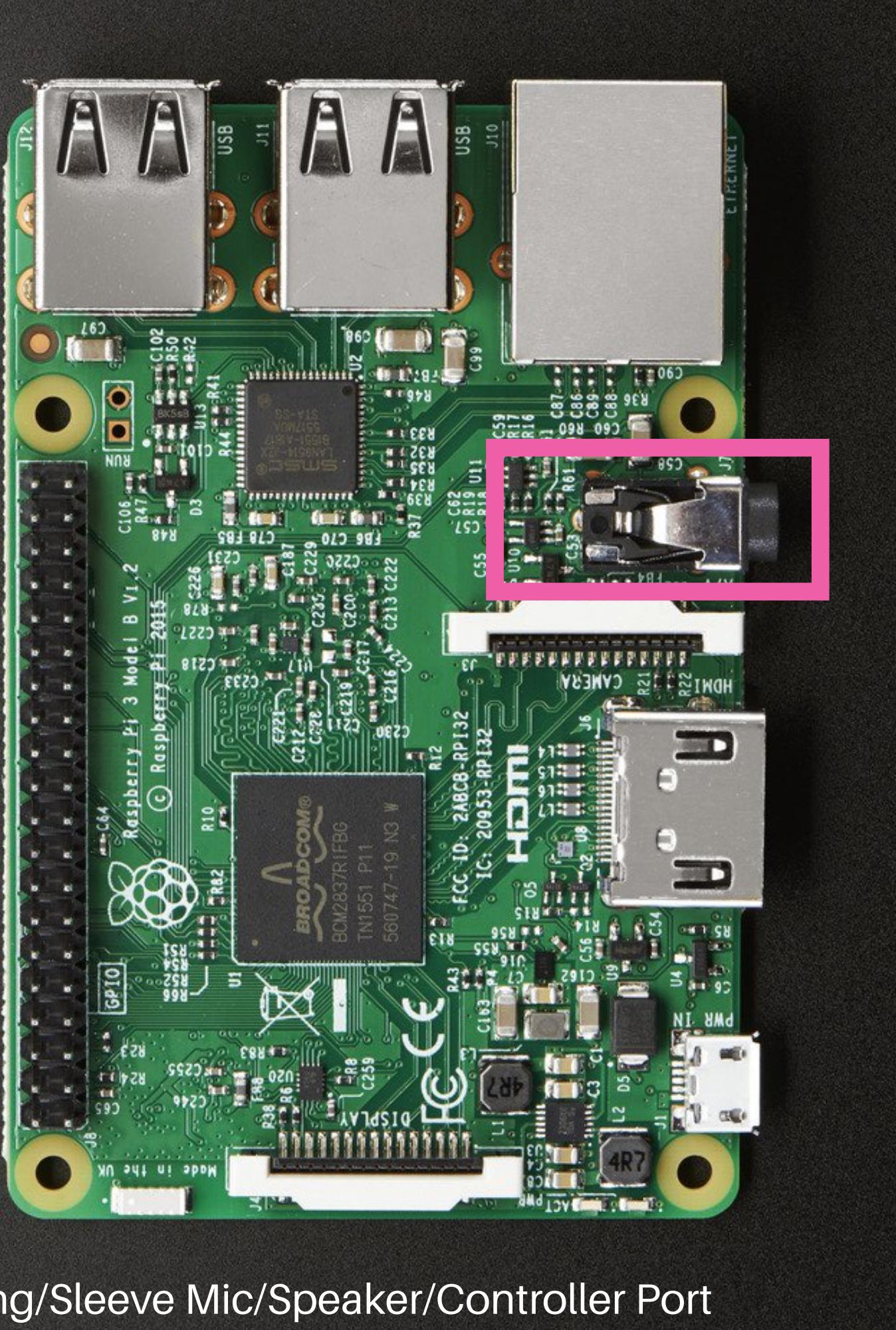
HDMI Port



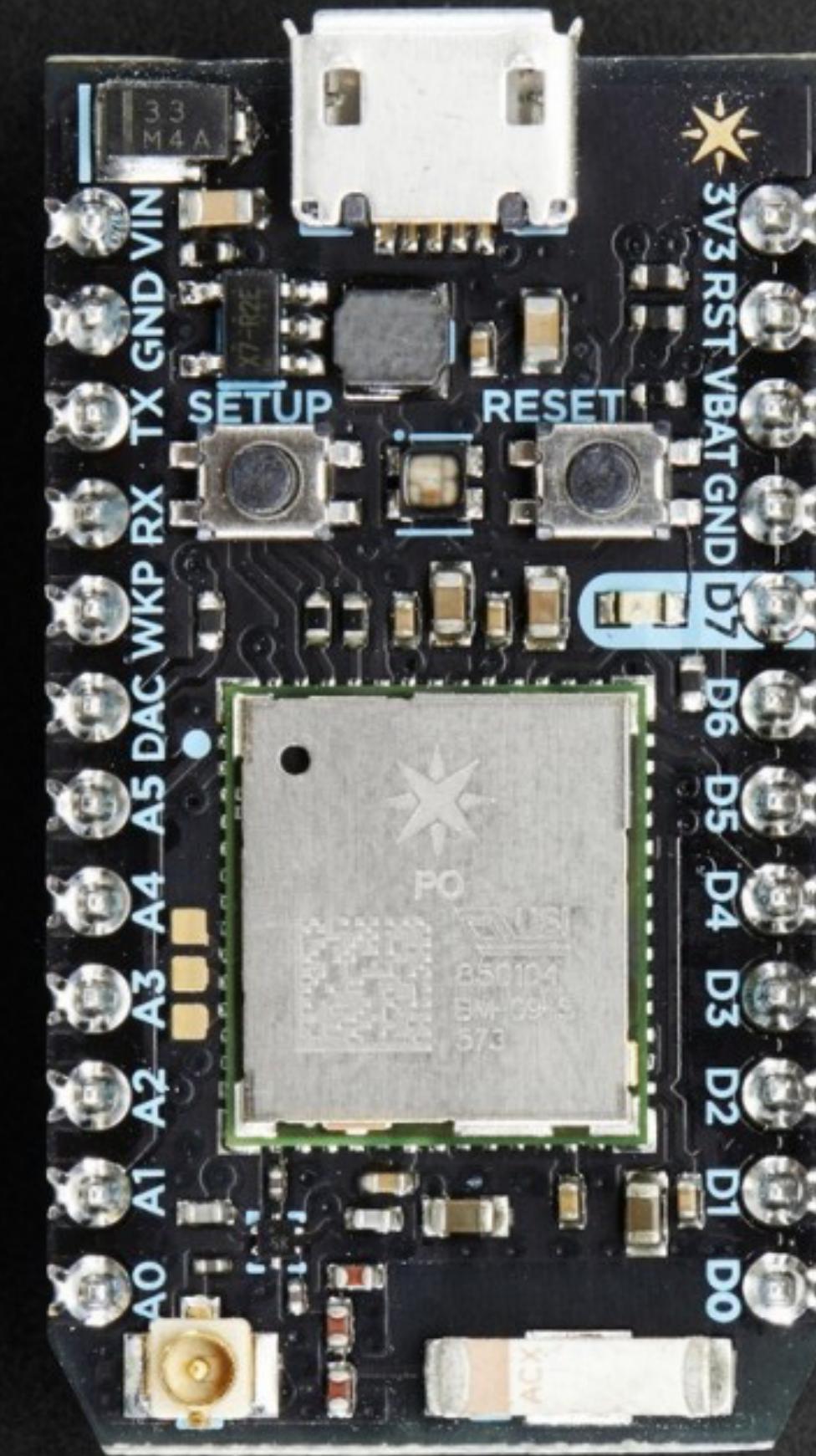


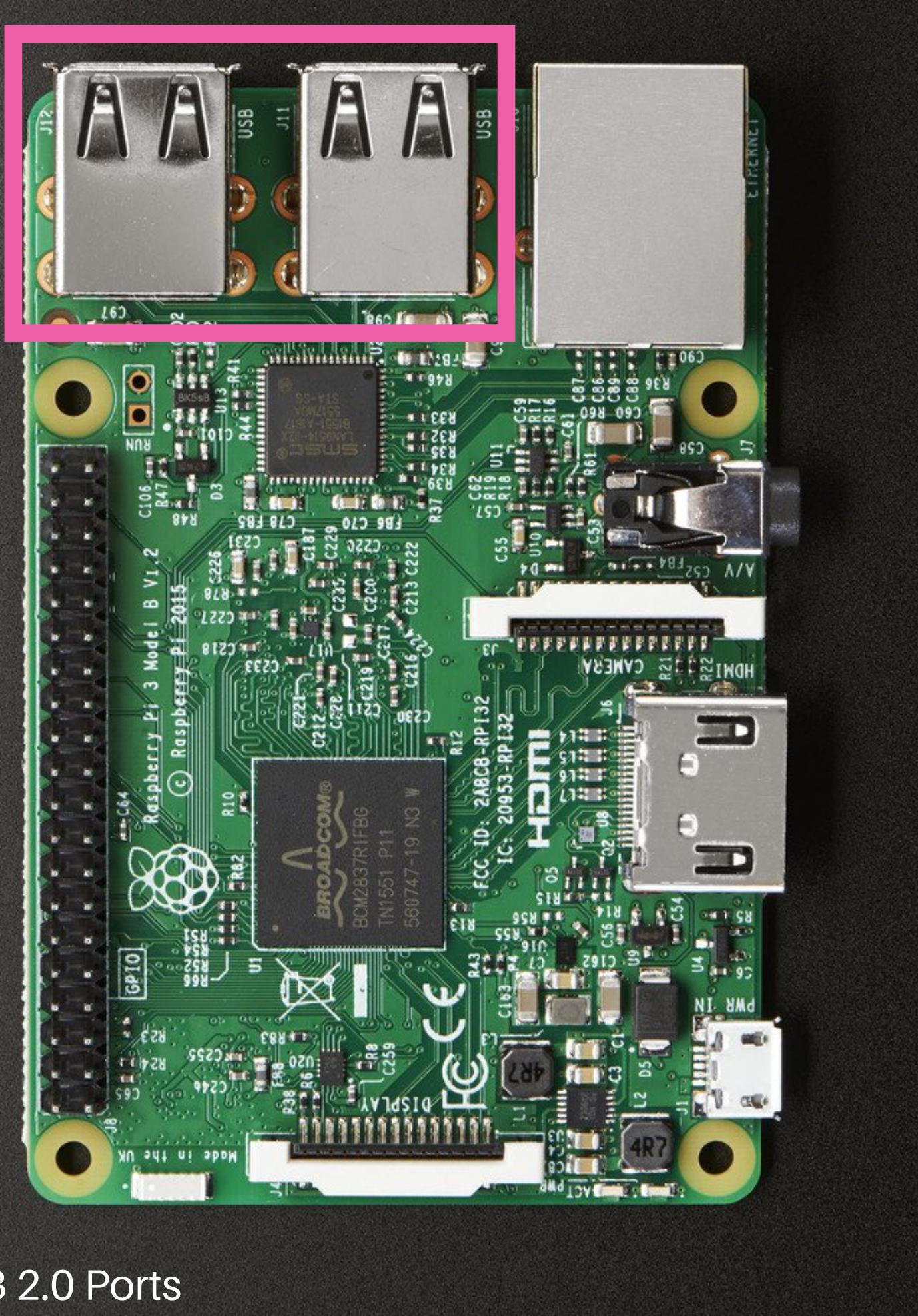
TFT Touchscreen and Webcam Connectors



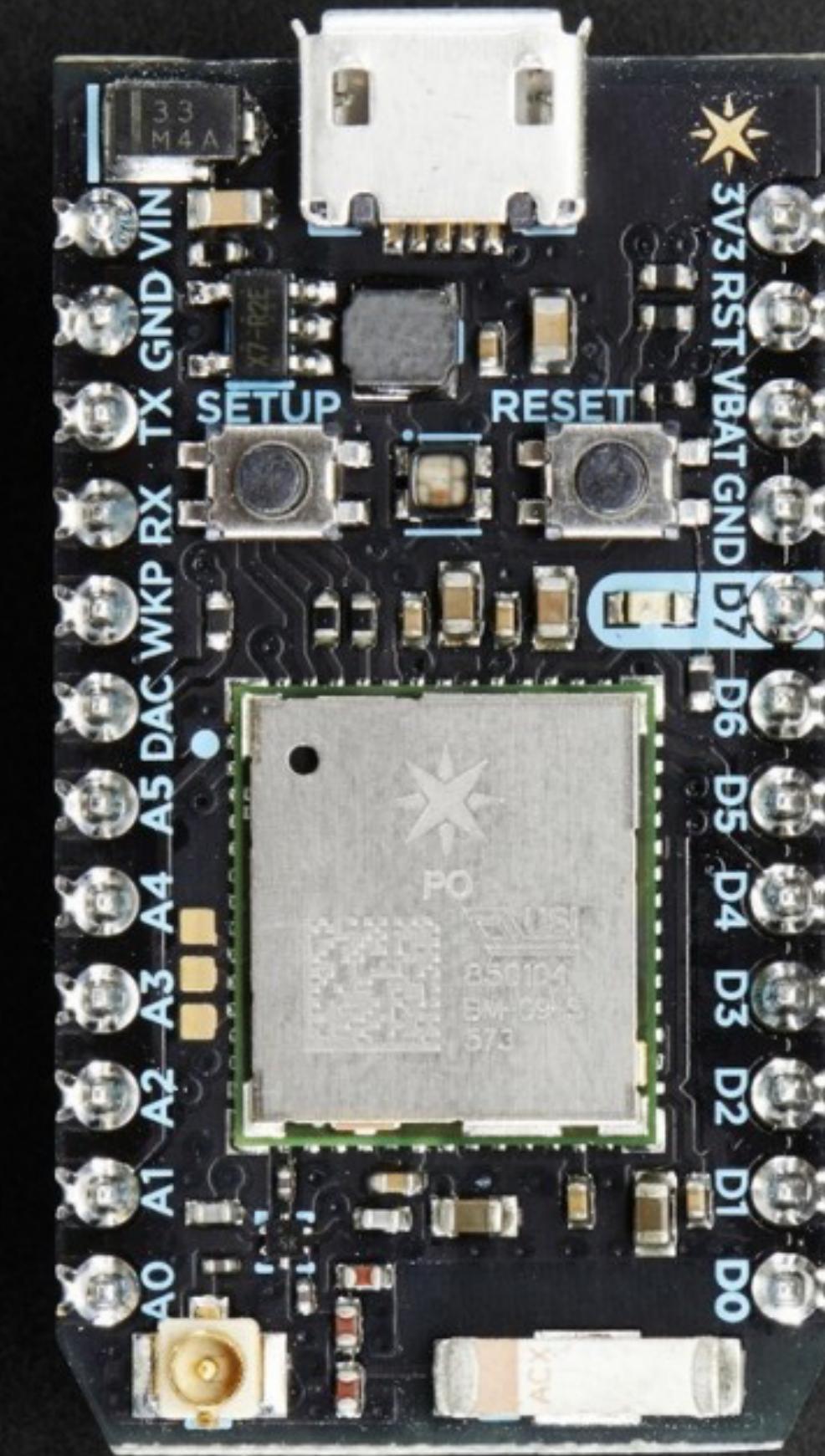


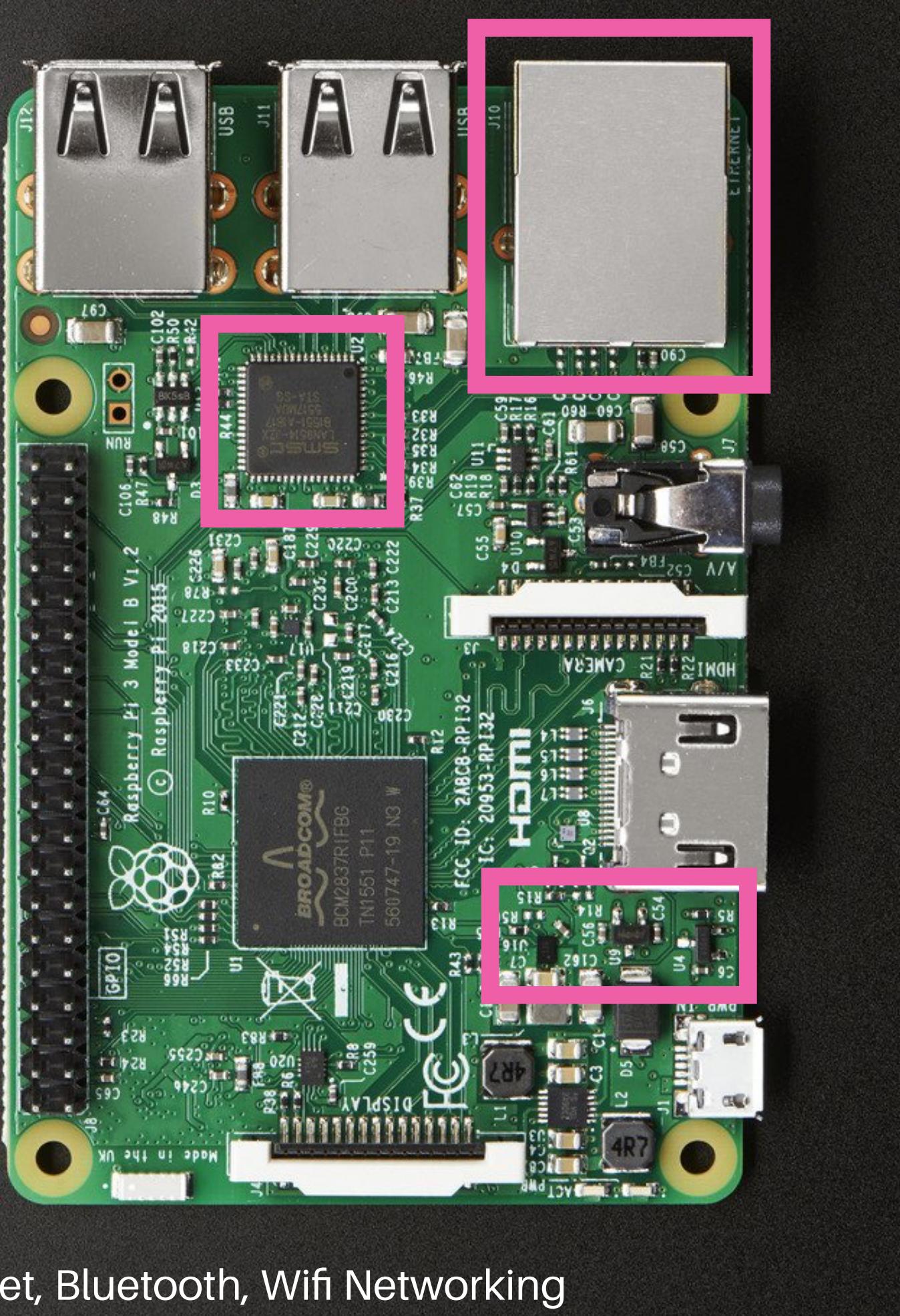
Tip/Ring/Sleeve Mic/Speaker/Controller Port



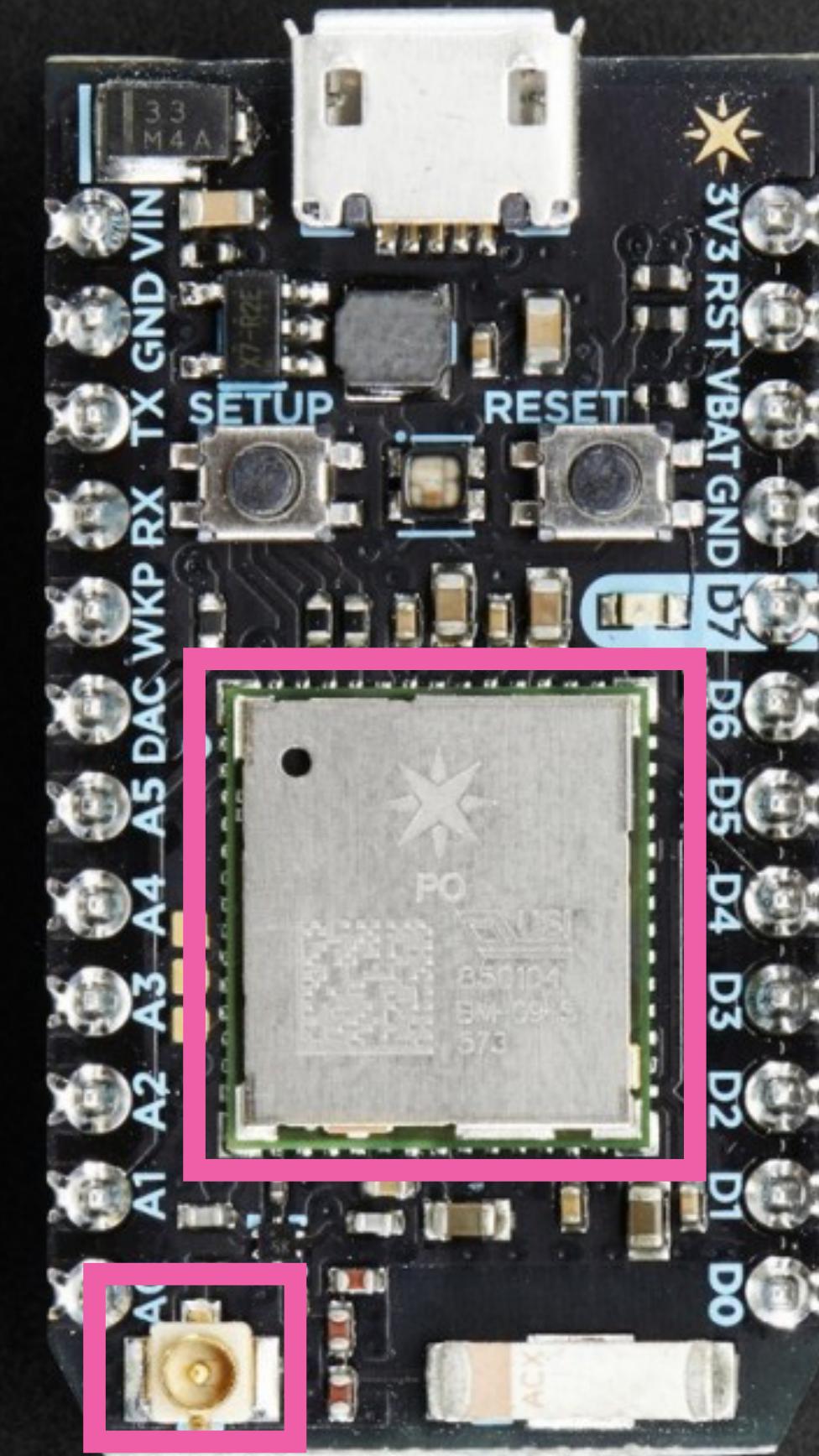


4 USB 2.0 Ports

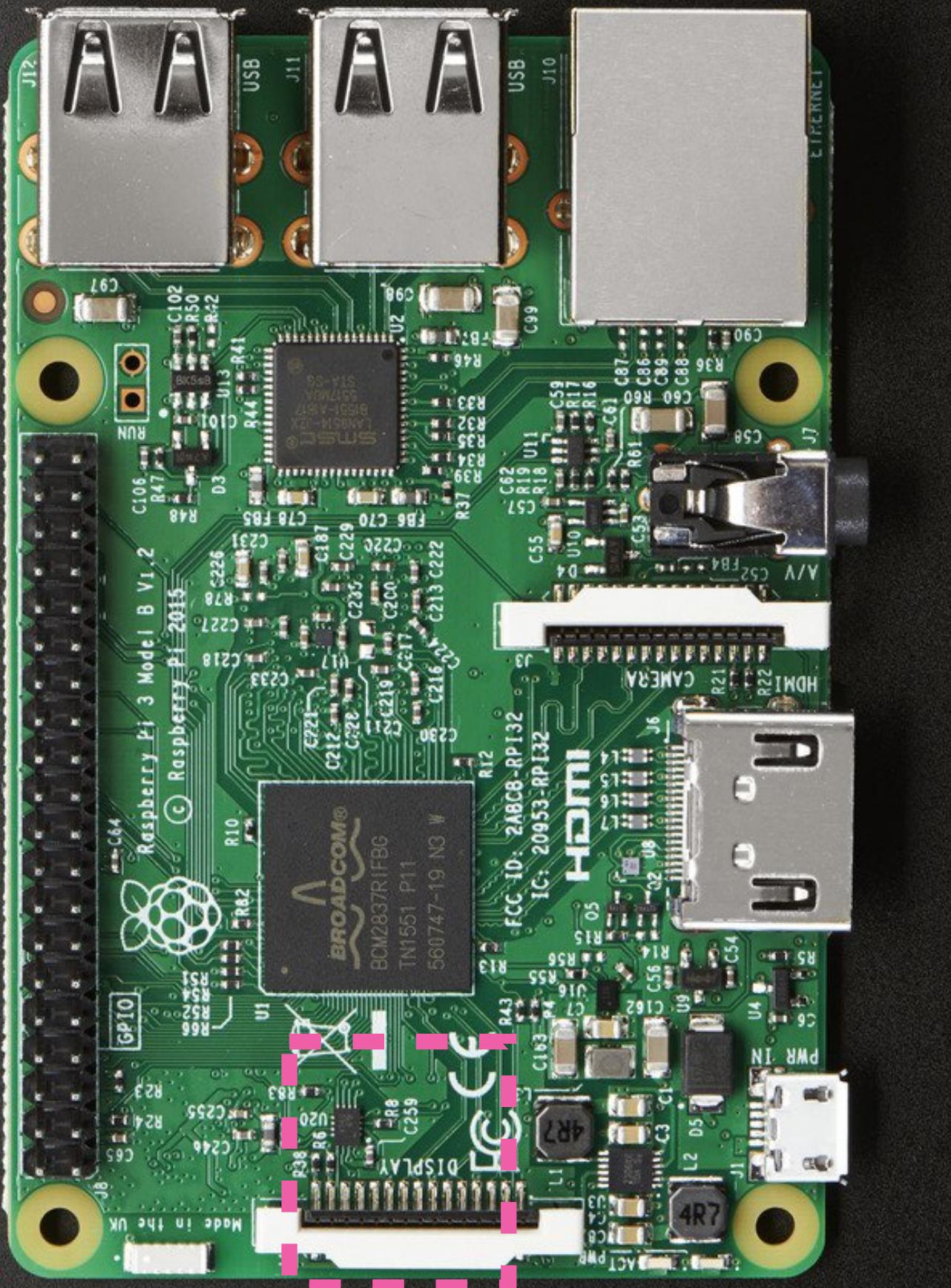




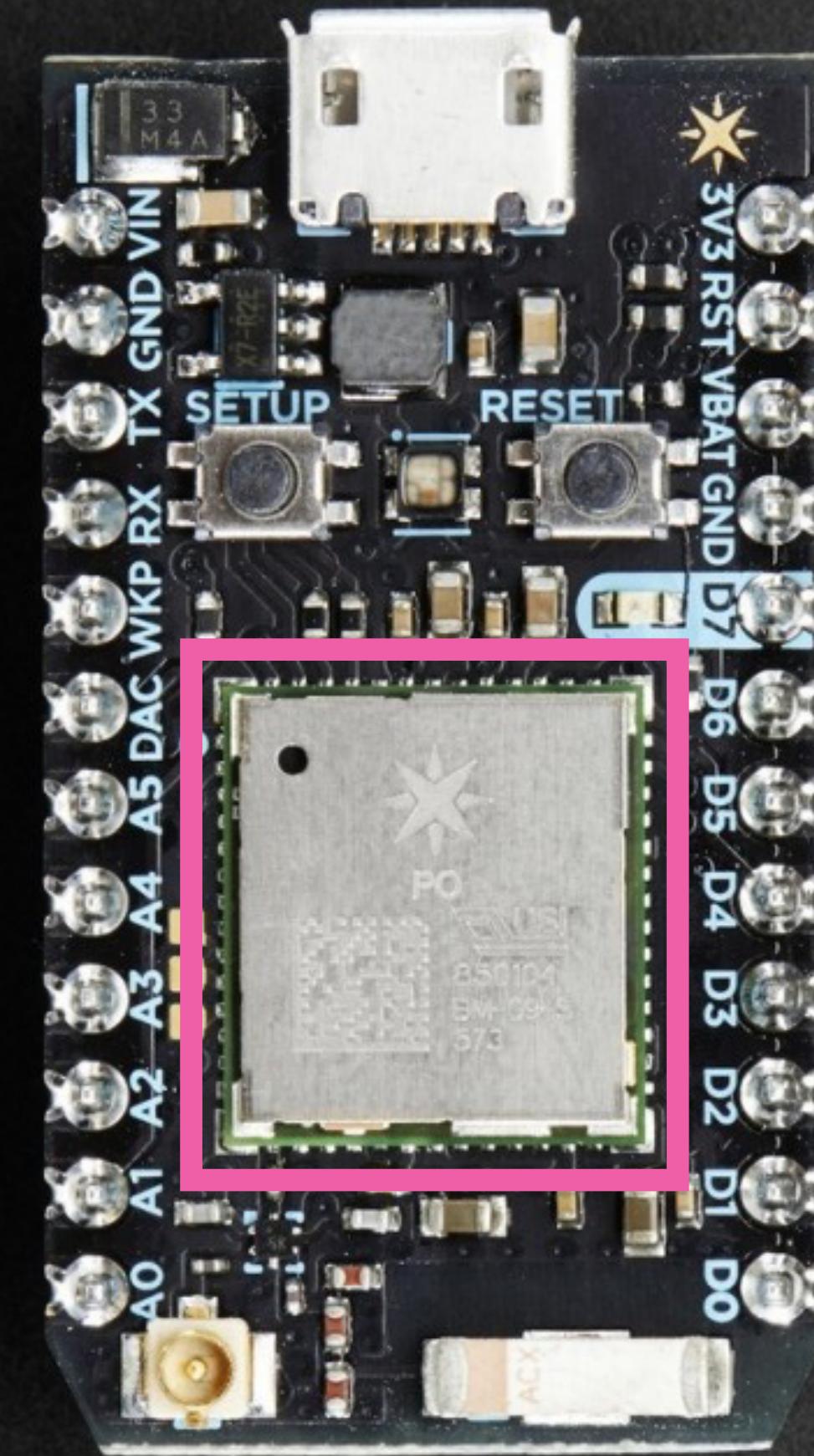
Ethernet, Bluetooth, Wifi Networking



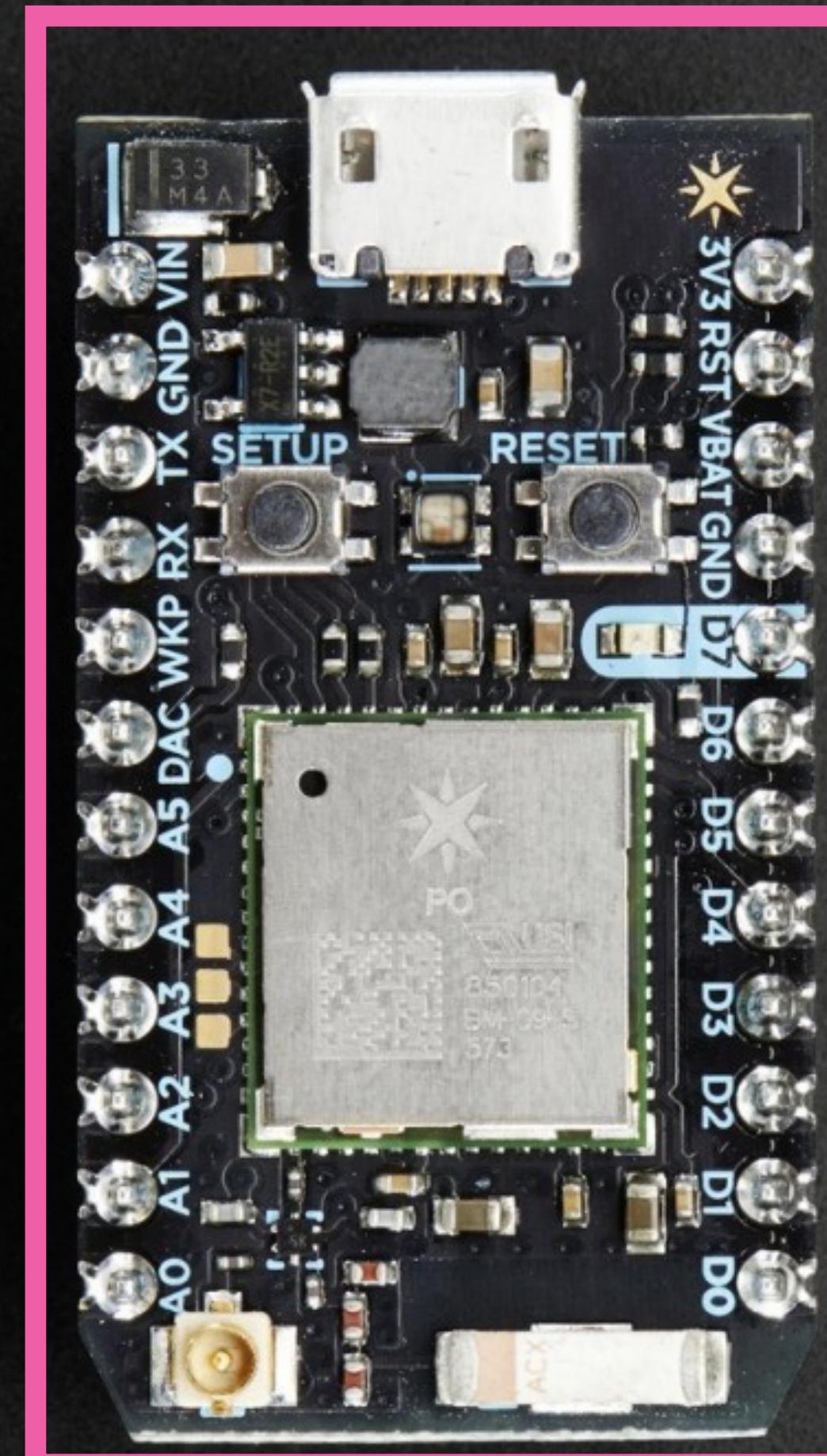
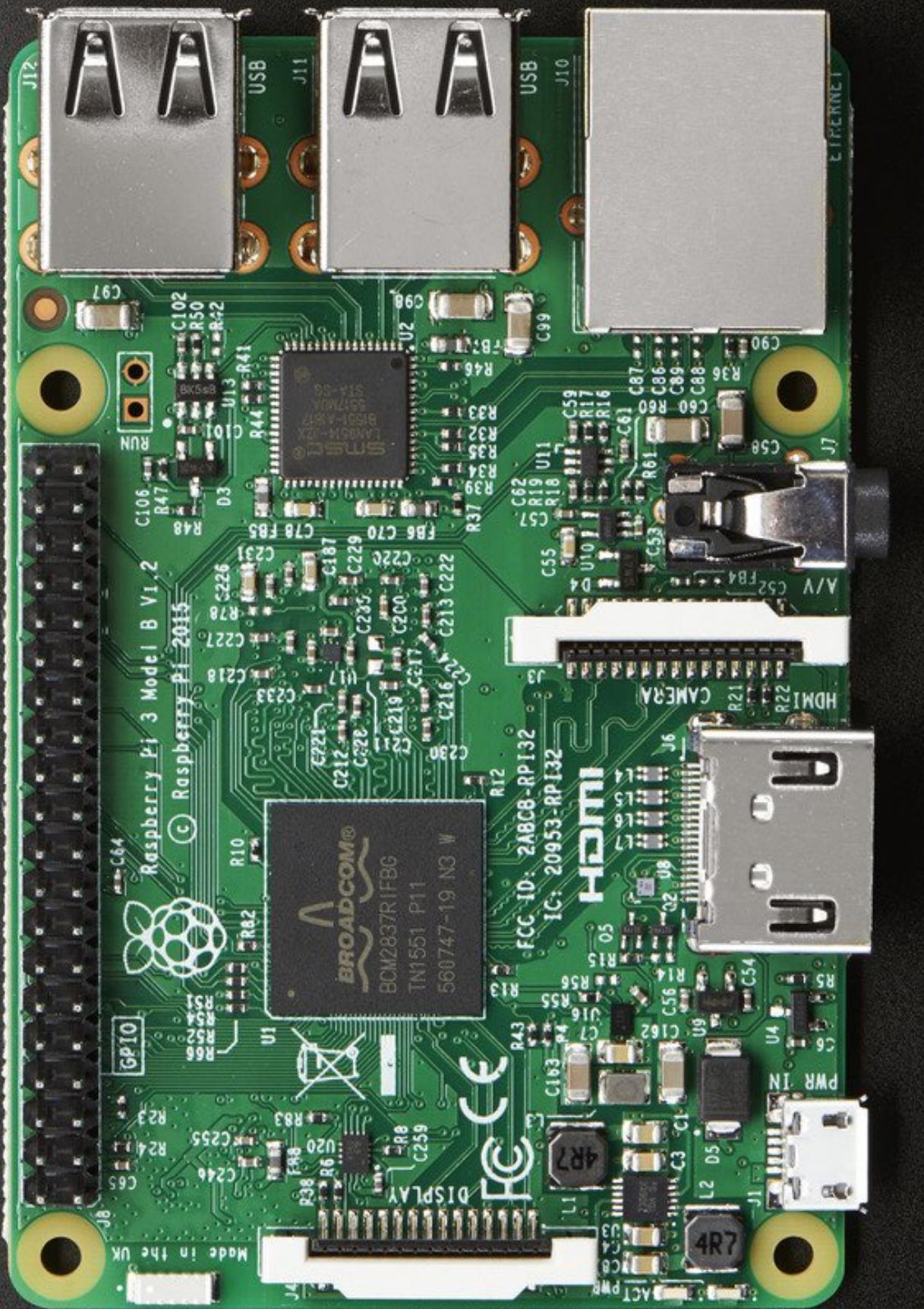
Low Power WiFi Module and Antenna Mount



Persistent Micro SD Storage



1MB Onboard Flash Memory

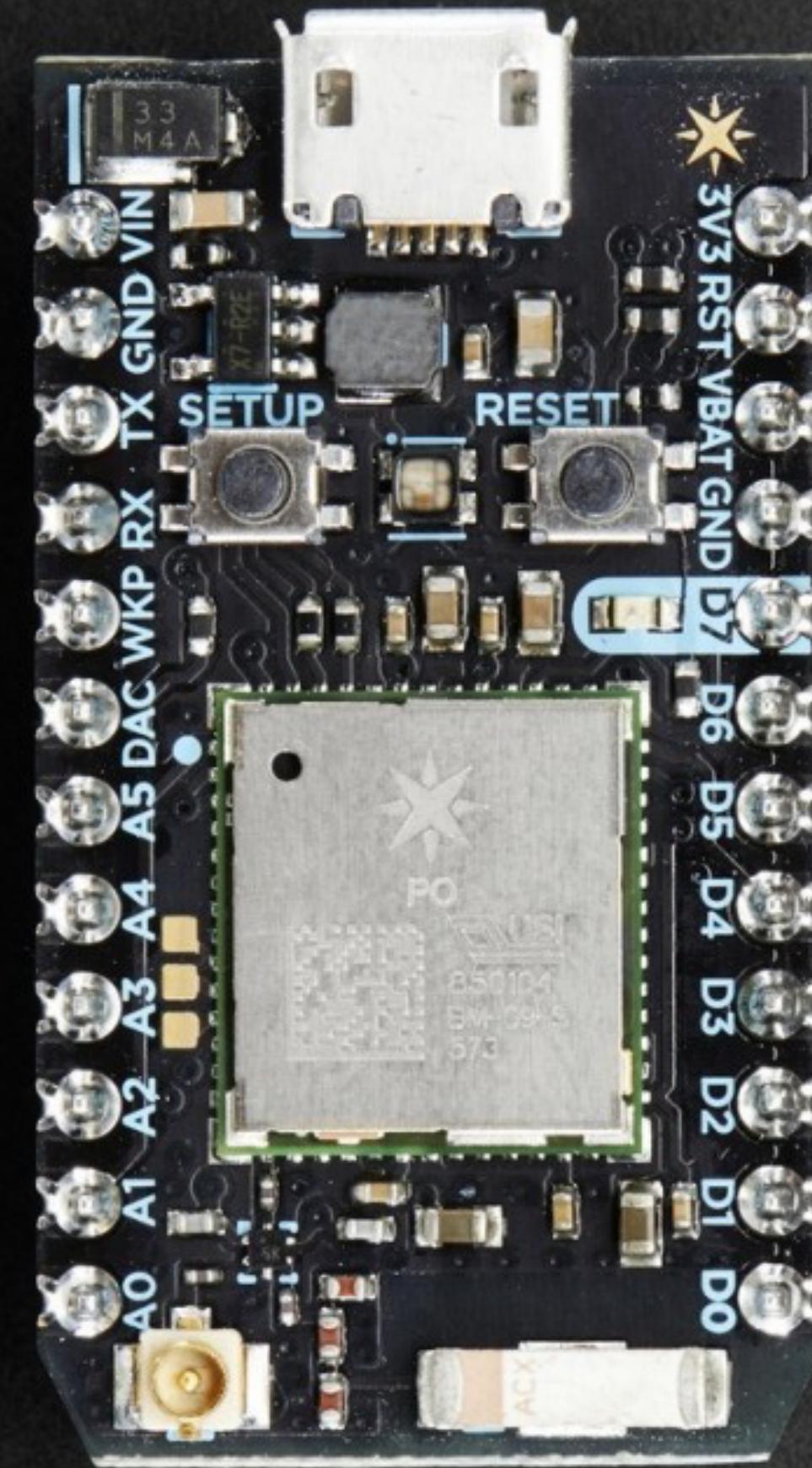


Supporting Cloud and Hardware Platform

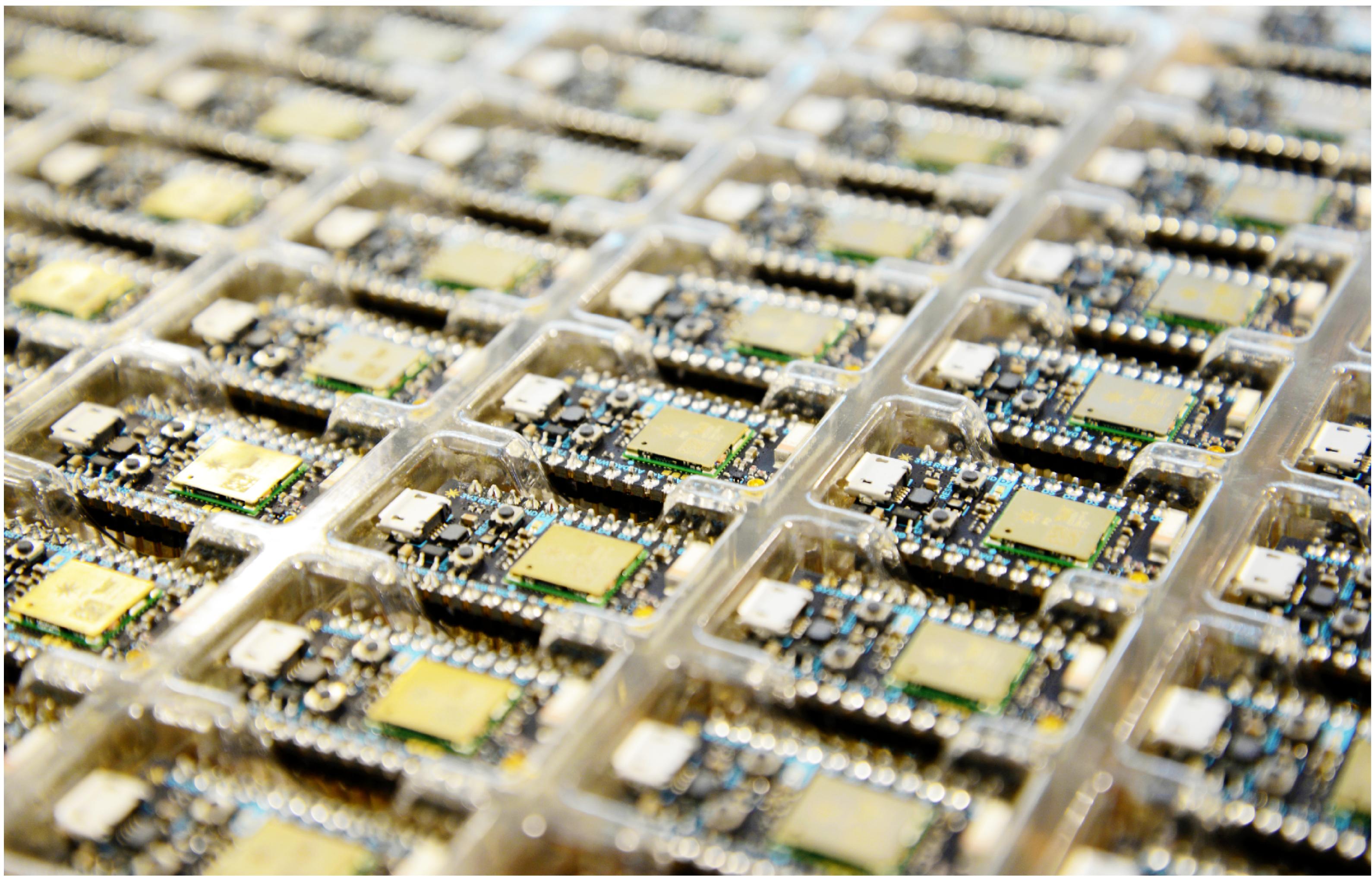




Personal Computer



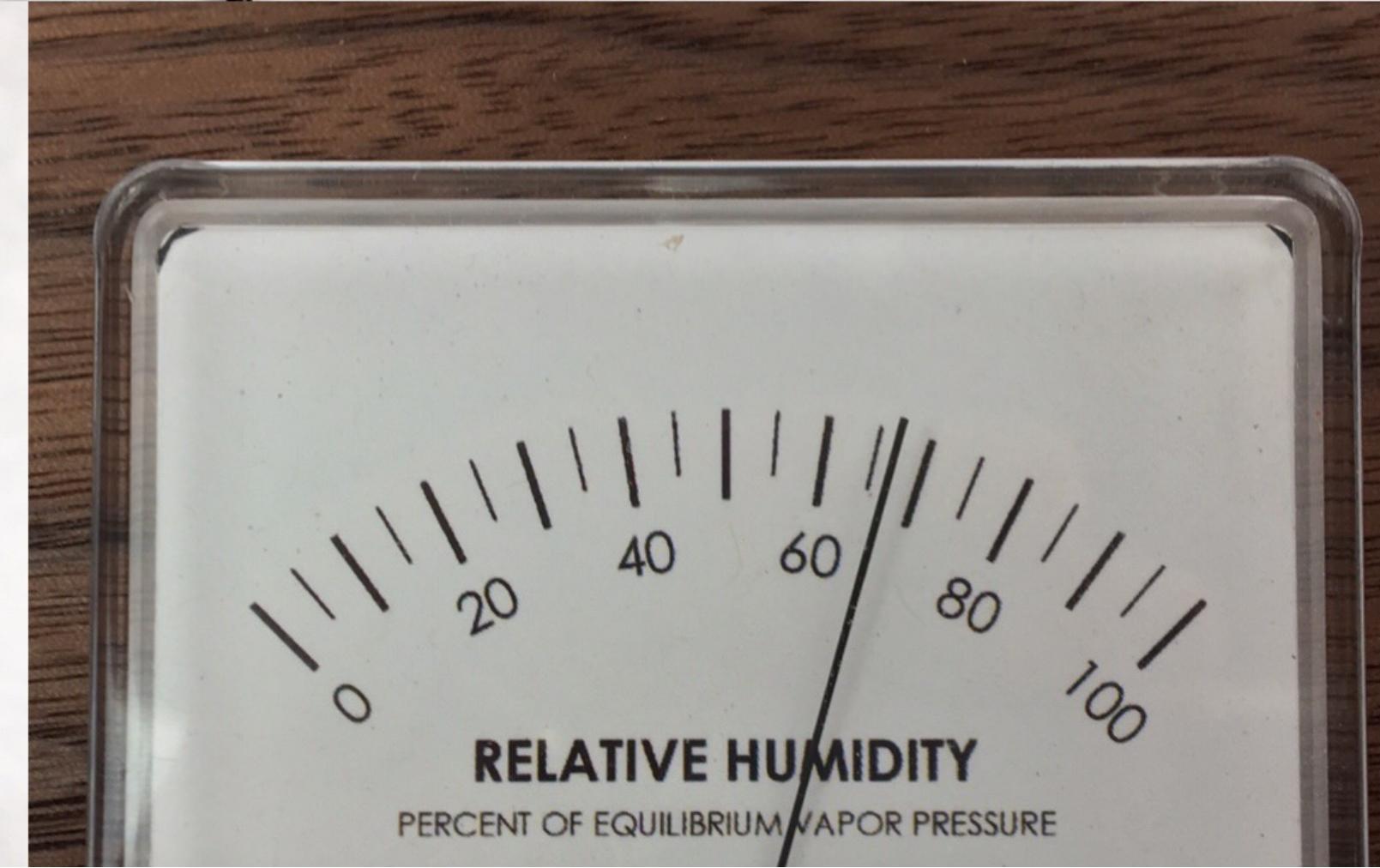
Embedded IoT Controller

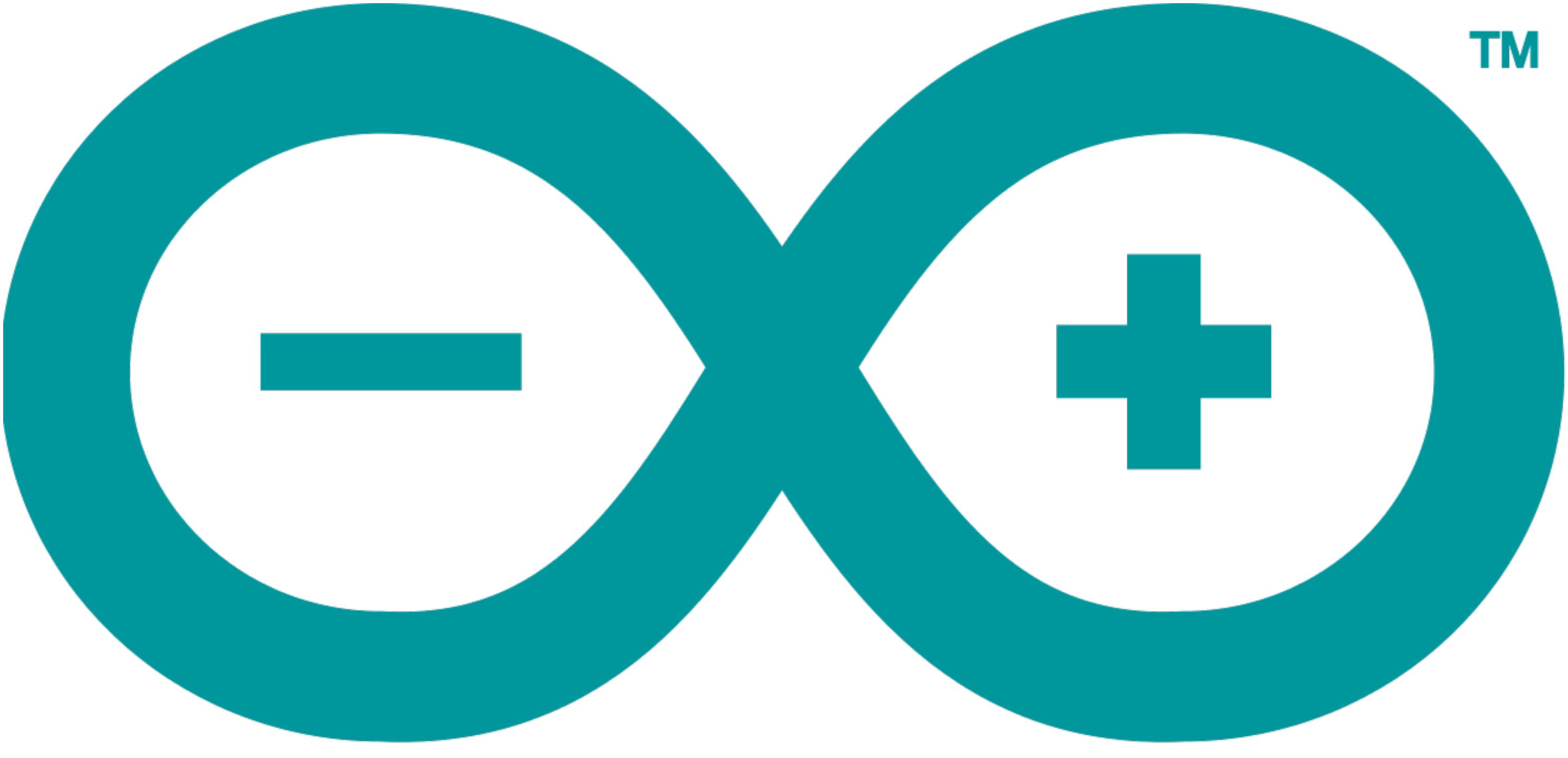










™

# ARDUTINO

The Software • Arduino-Flavored C  
Low-Level Assisted Programming



Massimo Banzi



David Cuartielles







## LANGUAGE

## FUNCTIONS

## VARIABLES

## STRUCTURE

## LIBRARIES

## GLOSSARY

The Arduino Reference text is licensed under a [Creative Commons Attribution-Share Alike 3.0 License](#).

Find anything that can be improved? [Suggest corrections and new documentation via GitHub](#).

Doubts on how to use Github? Learn everything you need to know in [this tutorial](#).

# Language Reference

Arduino programming language can be divided in three main parts: structure, values (variables and constants), and functions.

## FUNCTIONS

For controlling the Arduino board and performing computations.

### Digital I/O

[digitalRead\(\)](#)  
[digitalWrite\(\)](#)  
[pinMode\(\)](#)

### Analog I/O

[analogRead\(\)](#)  
[analogReference\(\)](#)  
[analogWrite\(\)](#)

### Zero, Due & MKR Family

[analogReadResolution\(\)](#)  
[analogWriteResolution\(\)](#)

### Advanced I/O

[noTone\(\)](#)  
[pulseIn\(\)](#)  
[pulseInLong\(\)](#)  
[shiftIn\(\)](#)  
[shiftOut\(\)](#)  
[tone\(\)](#)

### Time

[delay\(\)](#)  
[delayMicroseconds\(\)](#)  
[micros\(\)](#)  
[millis\(\)](#)

### Math

[abs\(\)](#)  
[constrain\(\)](#)  
[map\(\)](#)  
[max\(\)](#)  
[min\(\)](#)  
[pow\(\)](#)  
[sq\(\)](#)  
[sqrt\(\)](#)

### Trigonometry

[cos\(\)](#)  
[sin\(\)](#)  
[tan\(\)](#)

### Characters

[isAlpha\(\)](#)  
[isAlphaNumeric\(\)](#)  
[isAscii\(\)](#)  
[isControl\(\)](#)  
[isDigit\(\)](#)  
[isGraph\(\)](#)  
[isHexadecimalDigit\(\)](#)  
[isLowerCase\(\)](#)  
[isPrintable\(\)](#)  
[isPunct\(\)](#)  
[isSpace\(\)](#)  
[isUpperCase\(\)](#)  
[isWhitespace\(\)](#)

### Random Numbers

[random\(\)](#)  
[randomSeed\(\)](#)

### Bits and Bytes

[bit\(\)](#)  
[bitClear\(\)](#)  
[bitRead\(\)](#)  
[bitSet\(\)](#)  
[bitWrite\(\)](#)

### External Interrupts

[attachInterrupt\(\)](#)  
[detachInterrupt\(\)](#)

### Interrupts

[interrupts\(\)](#)  
[noInterrupts\(\)](#)

### Communication

[serial](#)  
[stream](#)

### USB

[Keyboard](#)  
[Mouse](#)

```
// print the integers from 1 to 9  
  
for (int i = 1; i < 10; i++)  
{    console.log(i);}
```

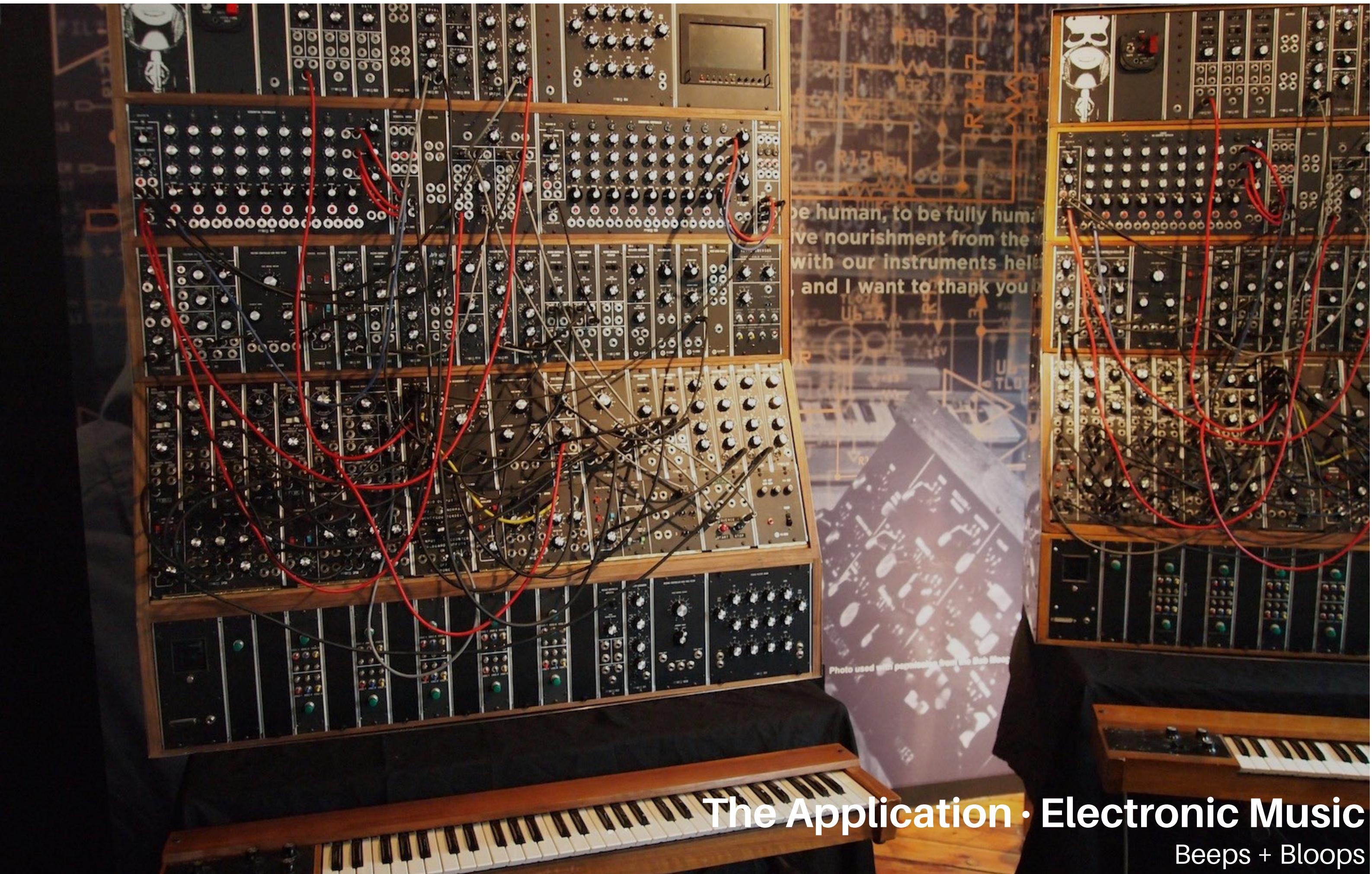
```
//value dependent behavior  
  
if (n === 0) {  
    console.log('no hits');}  
  
else if (n === 1) {  
    console.log('1 hit');}  
  
else {  
    console.log(n + ' hits');
```

```
var stringA = "abcd";  
var stringB = "ab";  
var stringC = "yz";  
var strt;  
var ends;  
  
//starts is True  
strt = stringA.substr(0, stringB.length) == stringB  
  
//ends is False  
ends = stringA.substr(-stringC.length) == stringC
```

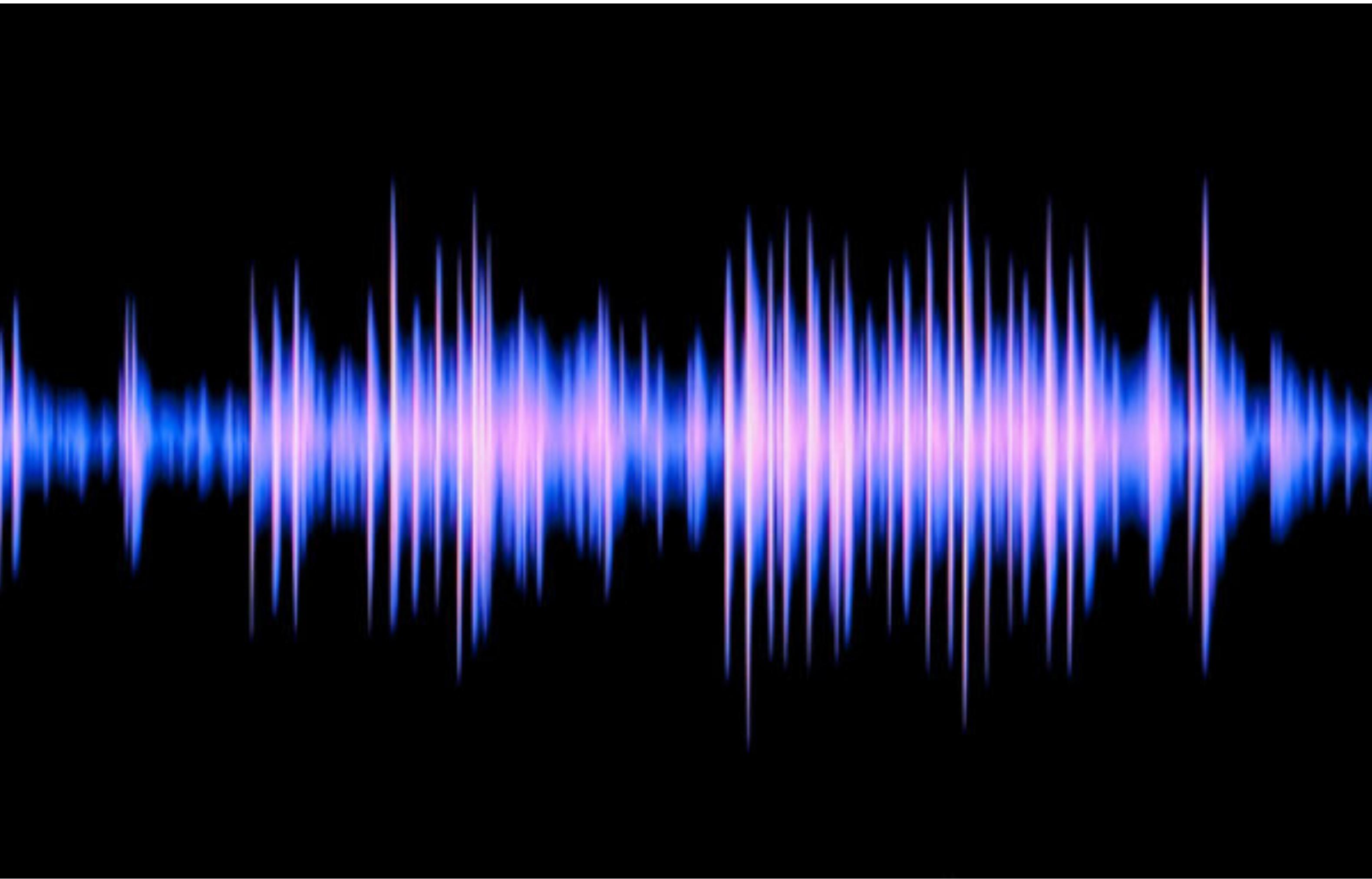
```
//print the integers from 1 to 9  
  
for i in range(1,10):  
    print i
```

```
//value dependent behavior  
  
if 0 == n:  
    print('no hits')  
elif 1 == n:  
    print('one hit')  
else:  
    print(str(n) + ' hits')
```

```
//returns True  
"abcd".startswith("ab")  
  
//returns False  
"abcd".endswith("yz")
```



The Application • Electronic Music  
Beeps + Bloops





**Theremin**

Alexandra Stepanoff playing the instrument invented by Leon Teremin in 1920



**Moog Modular Synthesizer**  
First Analog Synthesizer / Sequencer



**MicroKorg**  
Polyphony!



**Servo Harmonics**  
0s and 1s



**Chiptunes**  
8-Bit



**Reactable**  
Tangible Music

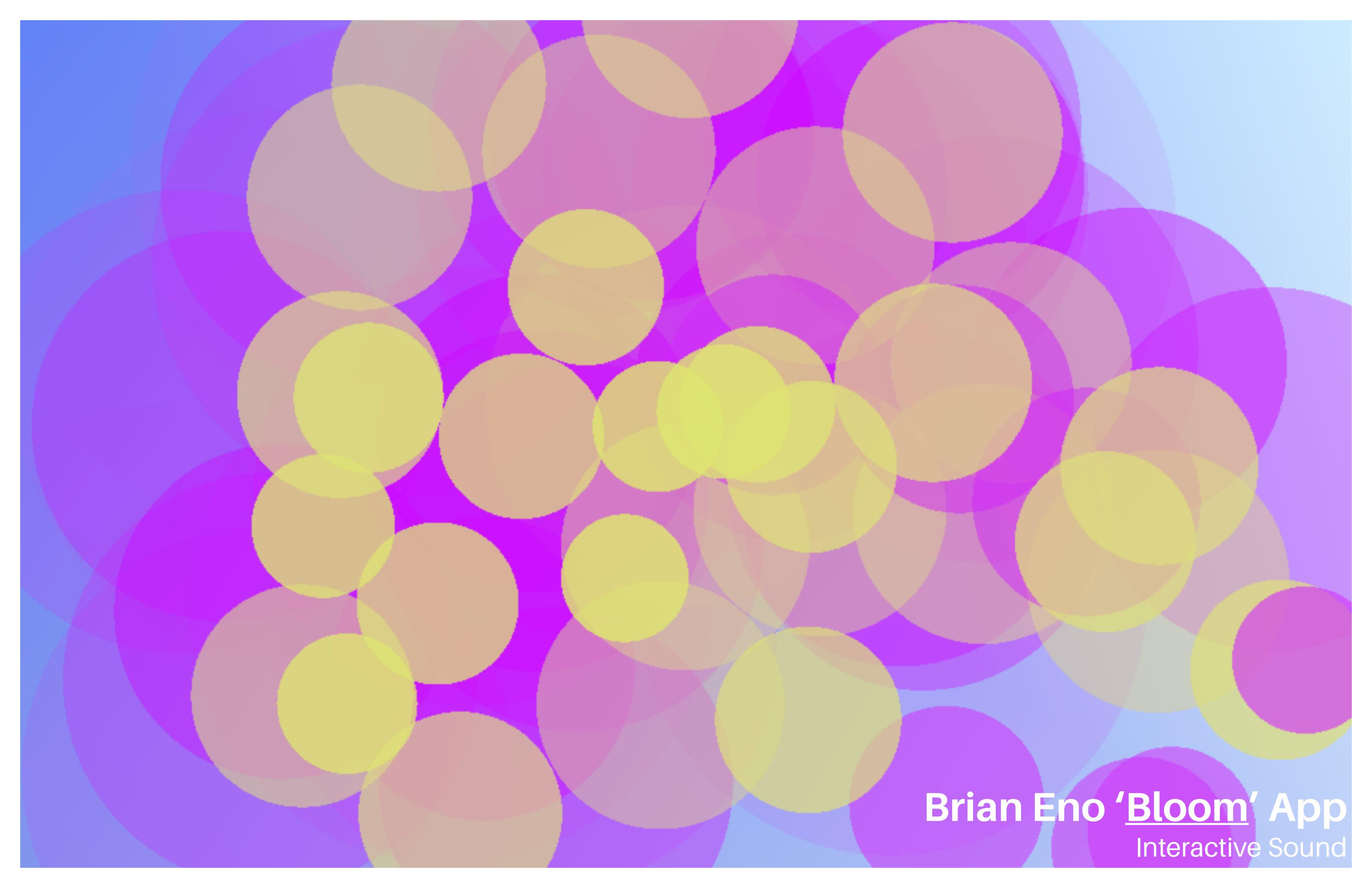


**Vocoder**

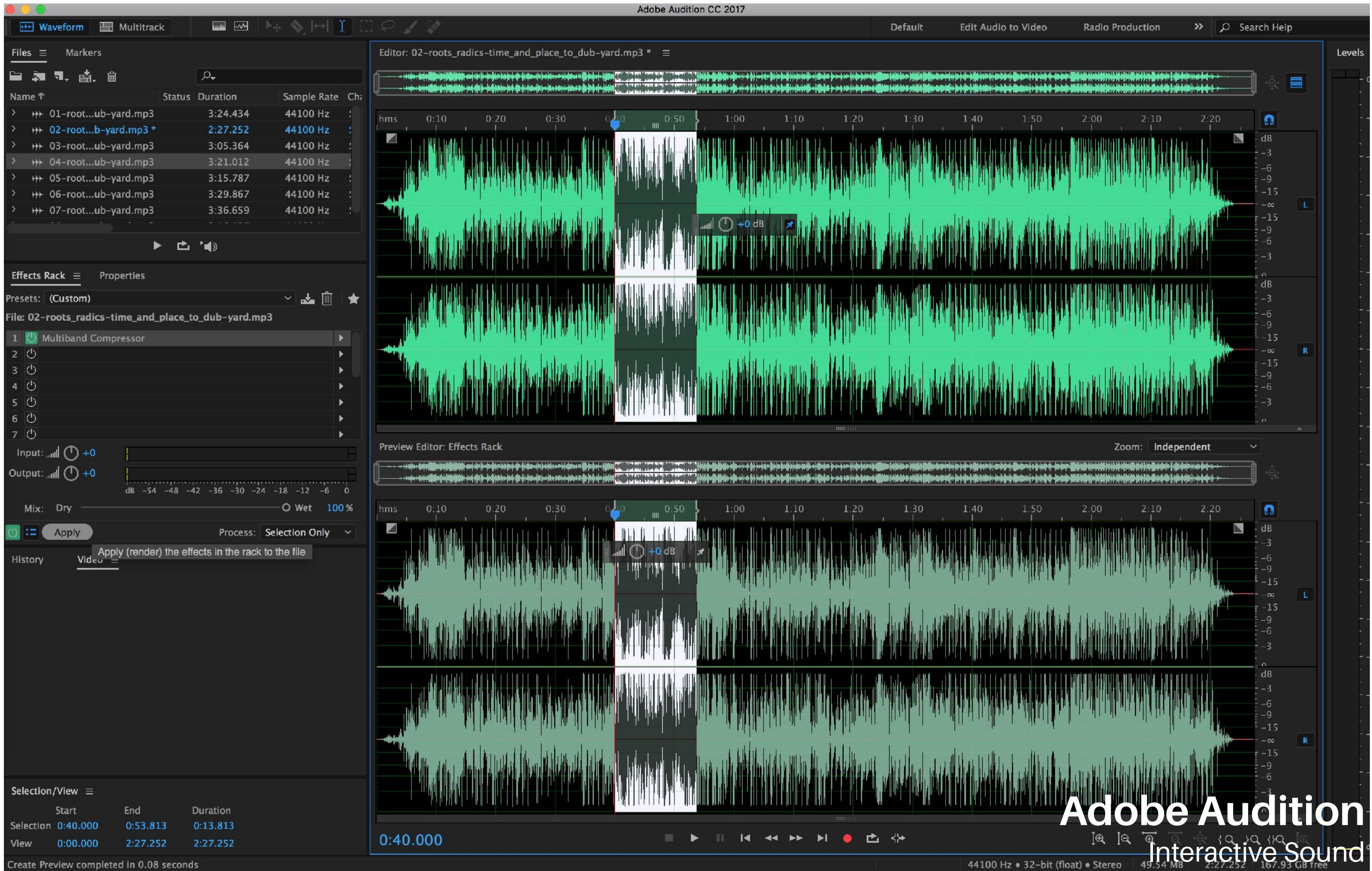
Autotu-u-u-u-une!



**Daft Punk, DeadMau5, Digitalism, Marshmello**  
Electronic Music and Wearable Technology

The background of the image is a dense, abstract pattern of overlapping circles. The circles are primarily in shades of yellow, orange, and pink, with some blue and purple ones interspersed. They overlap in various ways, creating a complex, organic texture across the entire frame.

Brian Eno 'Bloom' App  
Interactive Sound



# Adobe Audition

## Interactive Sound



**Seaboard**  
Analog Instrument Renewed



Teenage Engineering  
FX Boards



**Enhancia**

Analog Instrument Renewed



# Digital Development Workshop

Spring 2018 · Beeps and Bloops