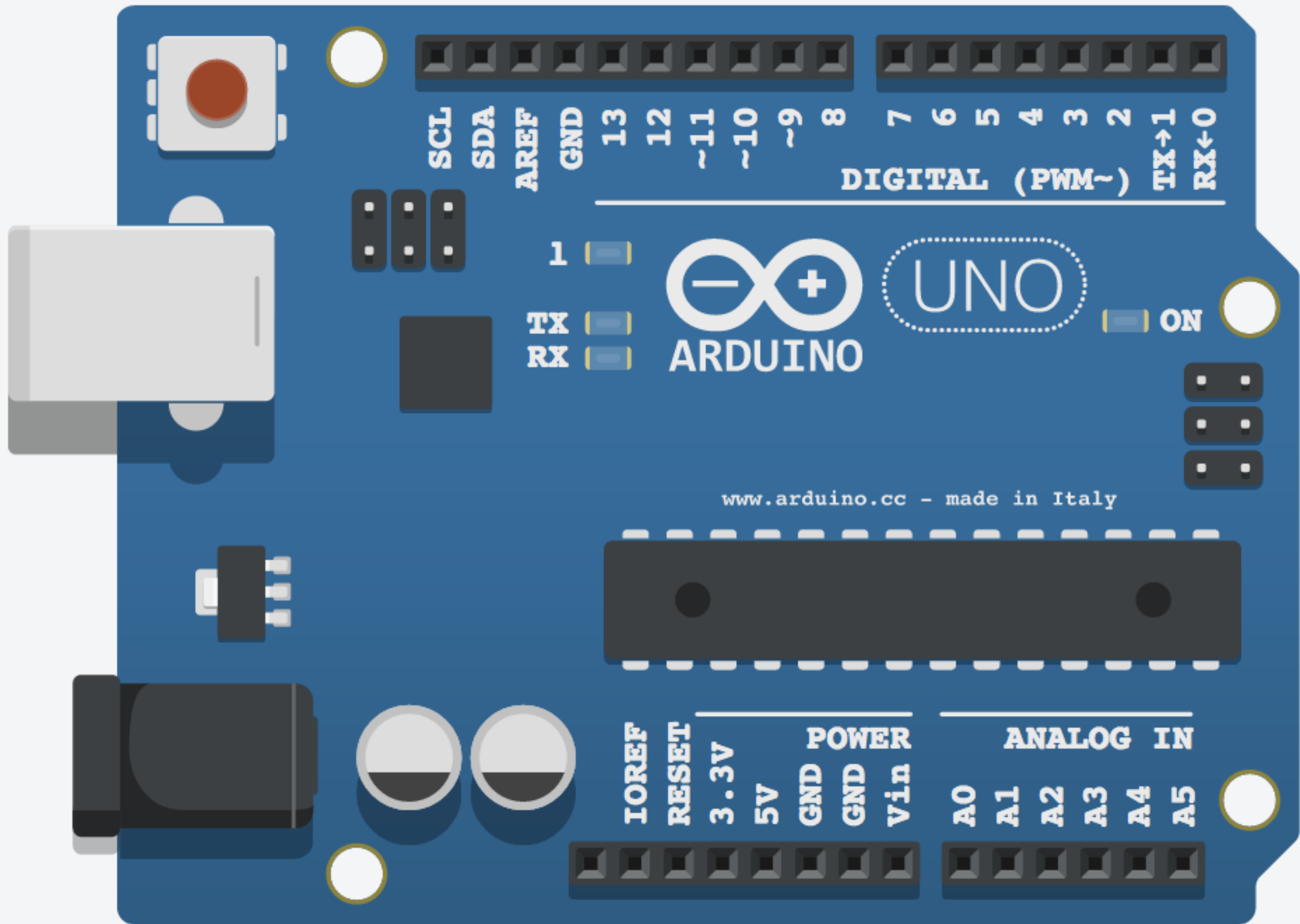
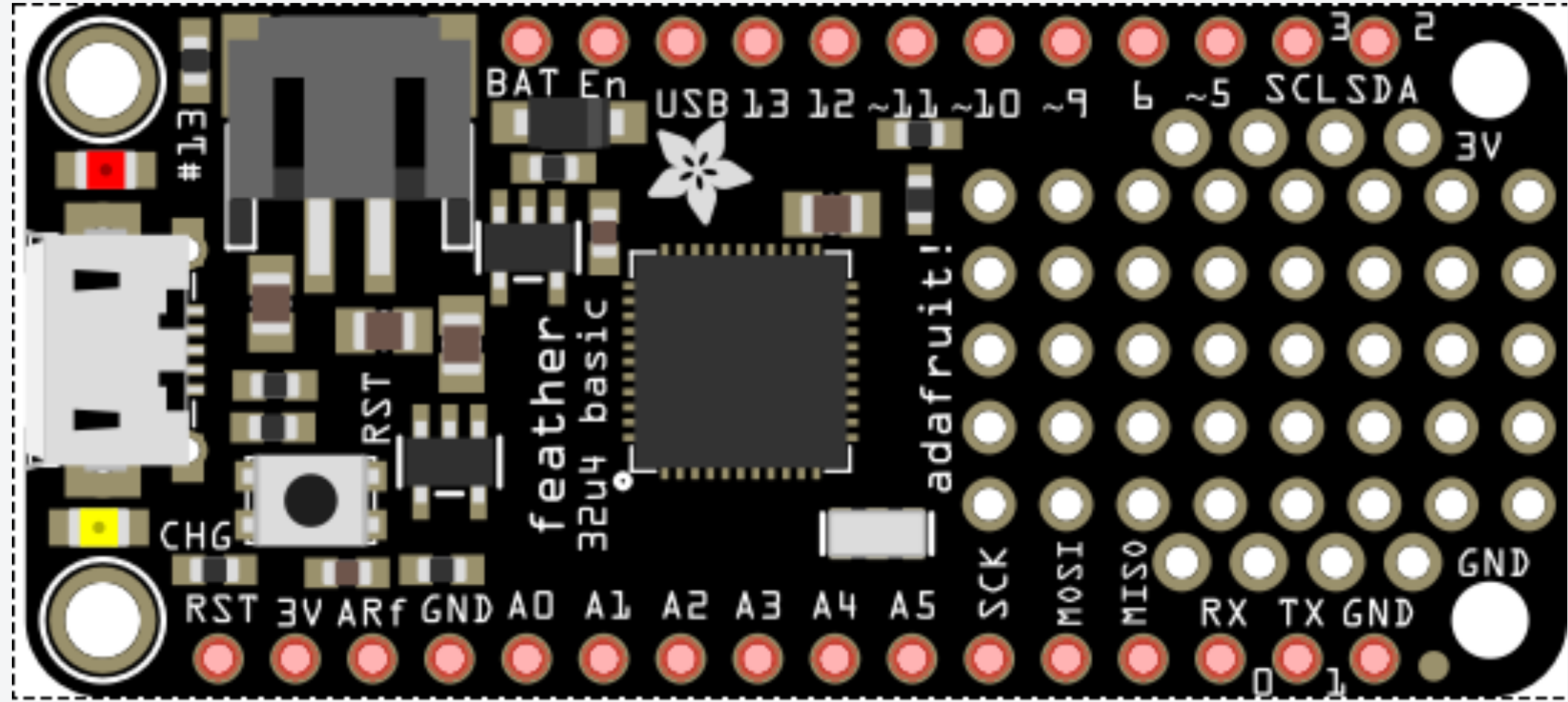


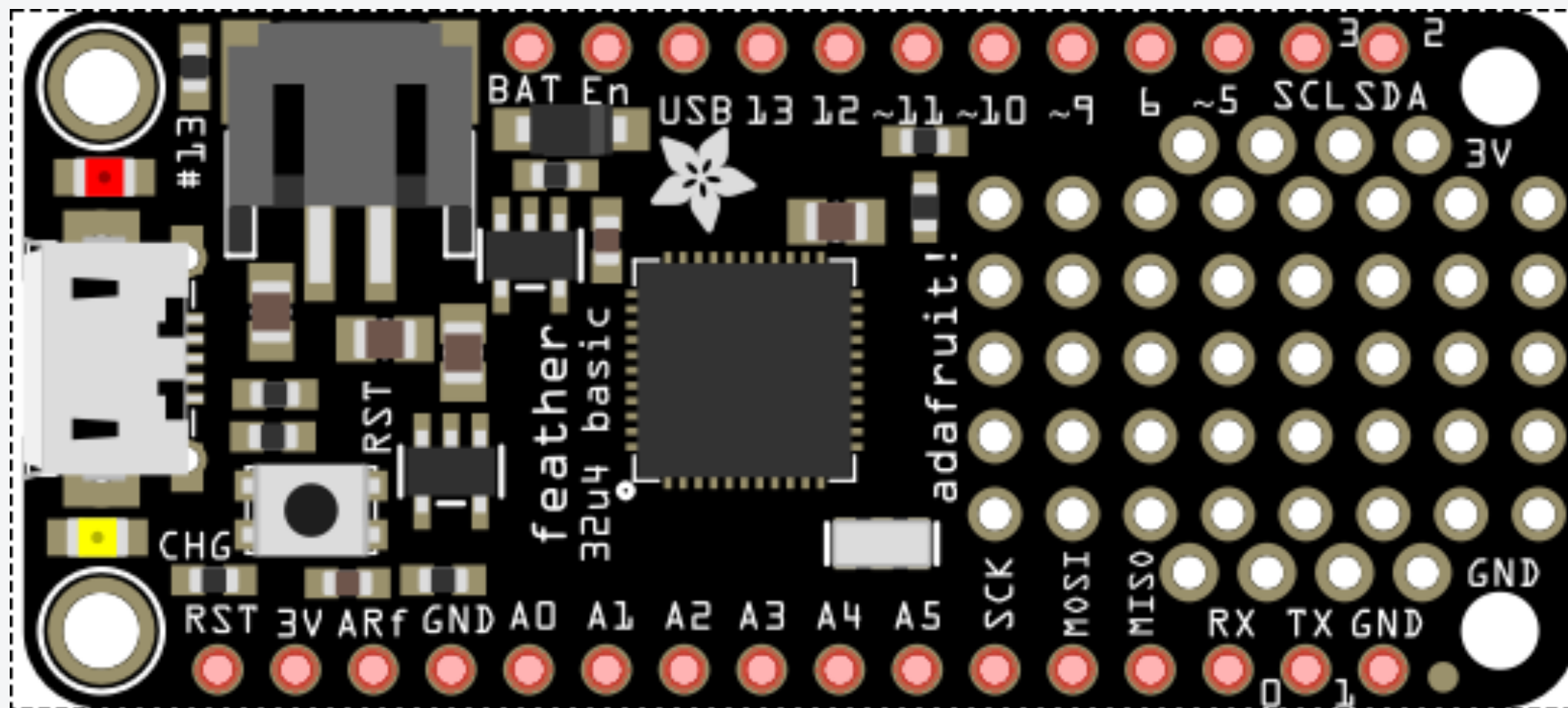
ARDUINO

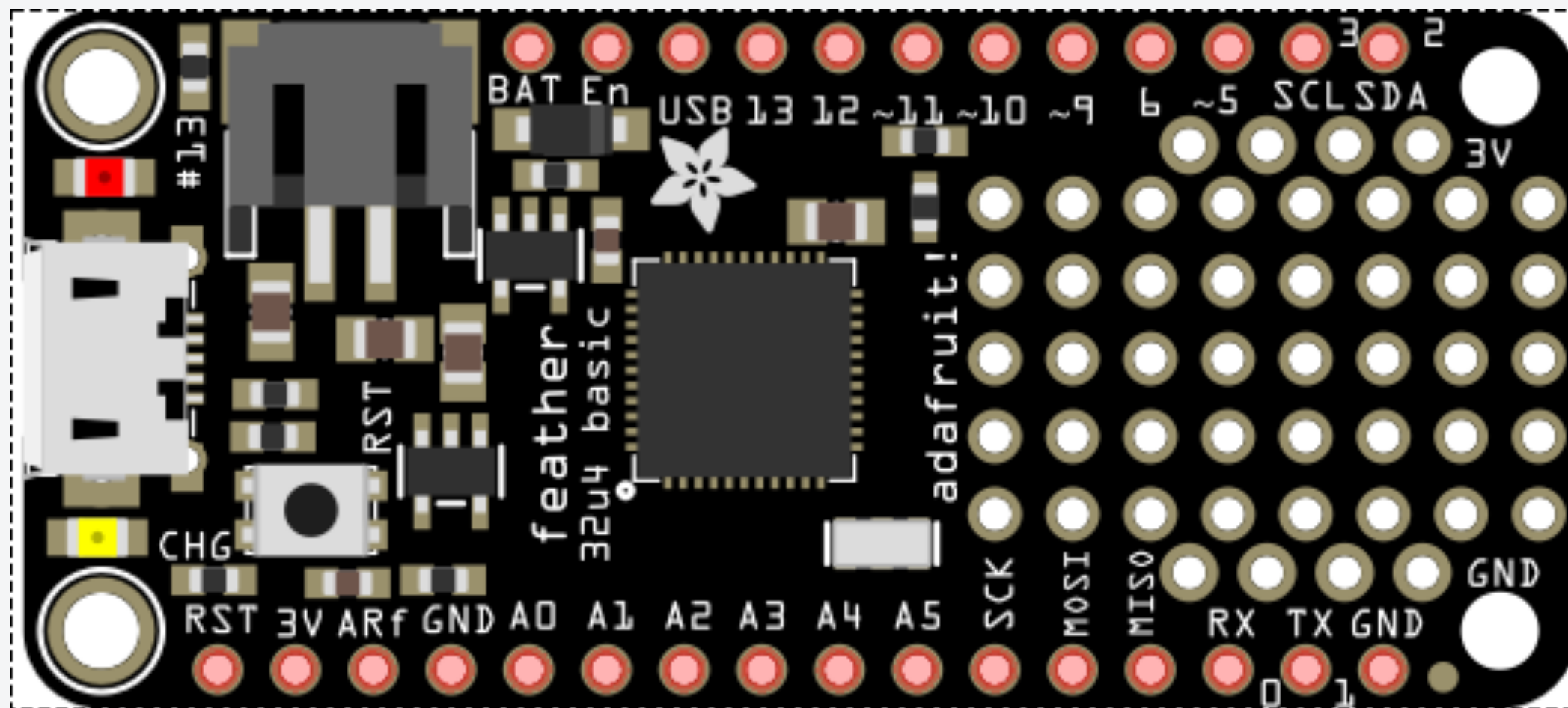
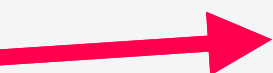
~~ARDUINO~~

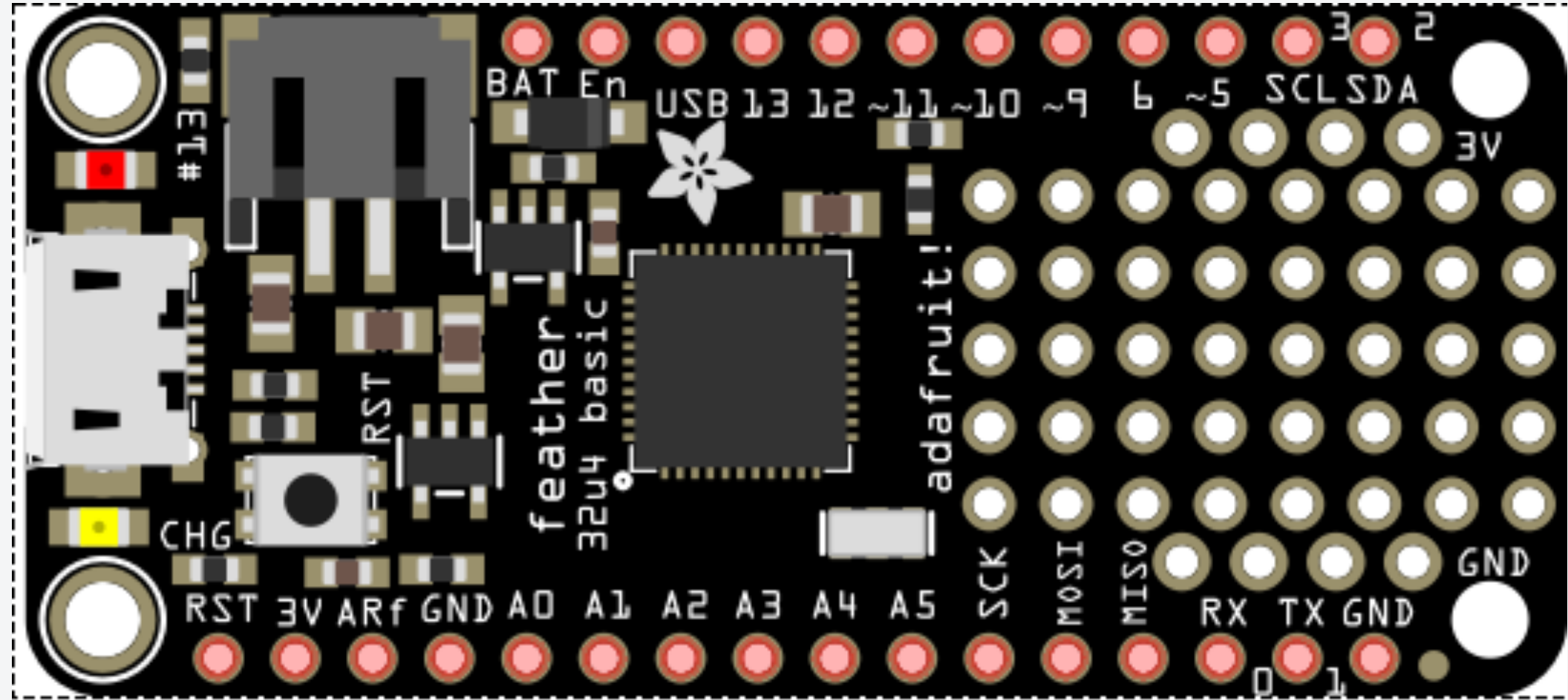
FEATHER 32U4

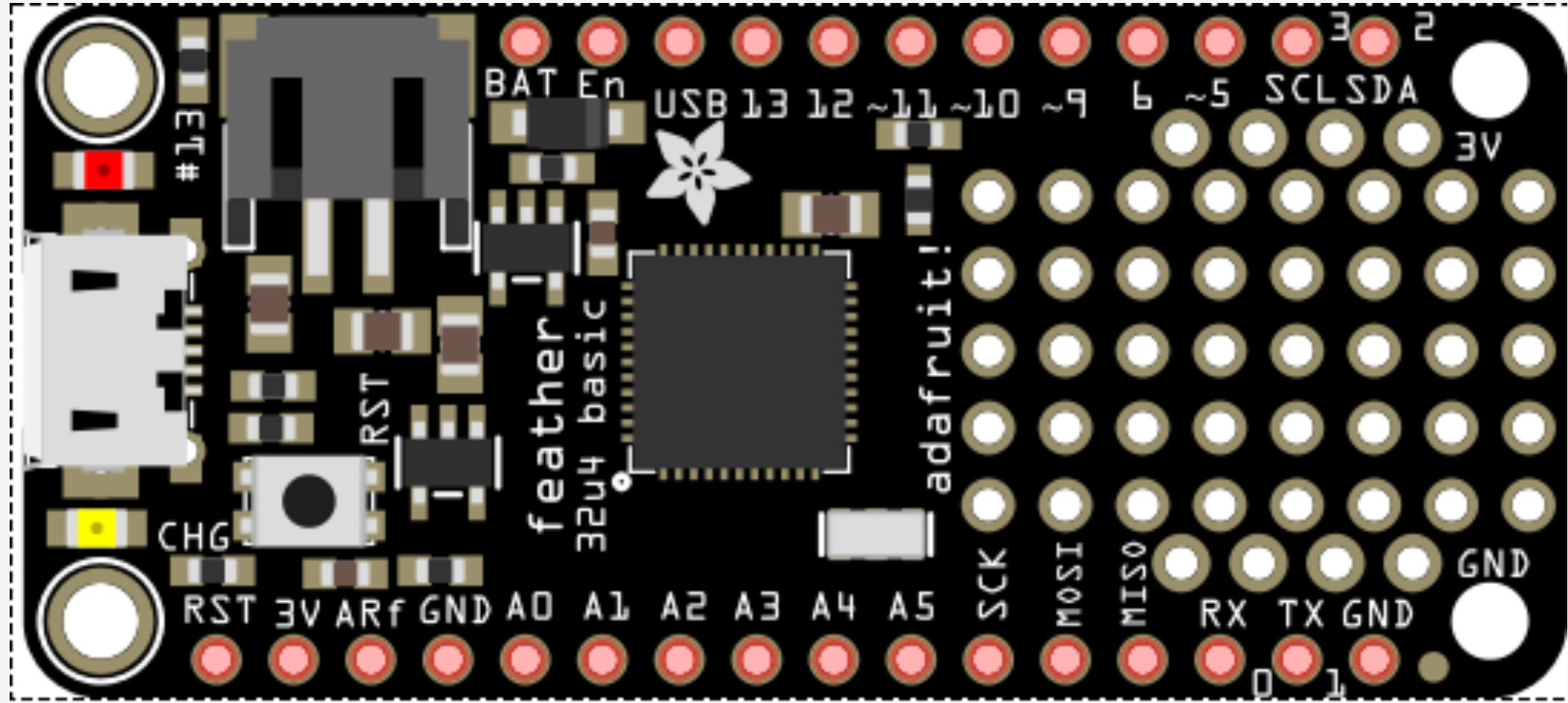


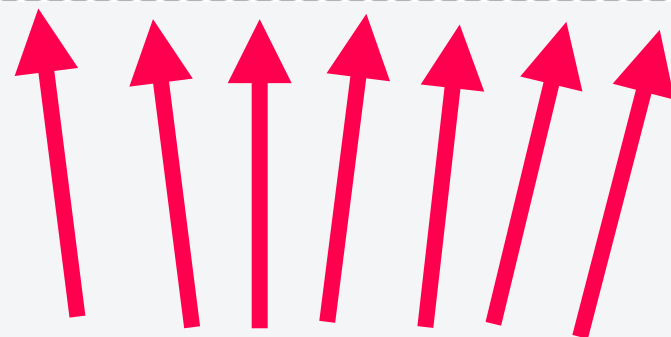
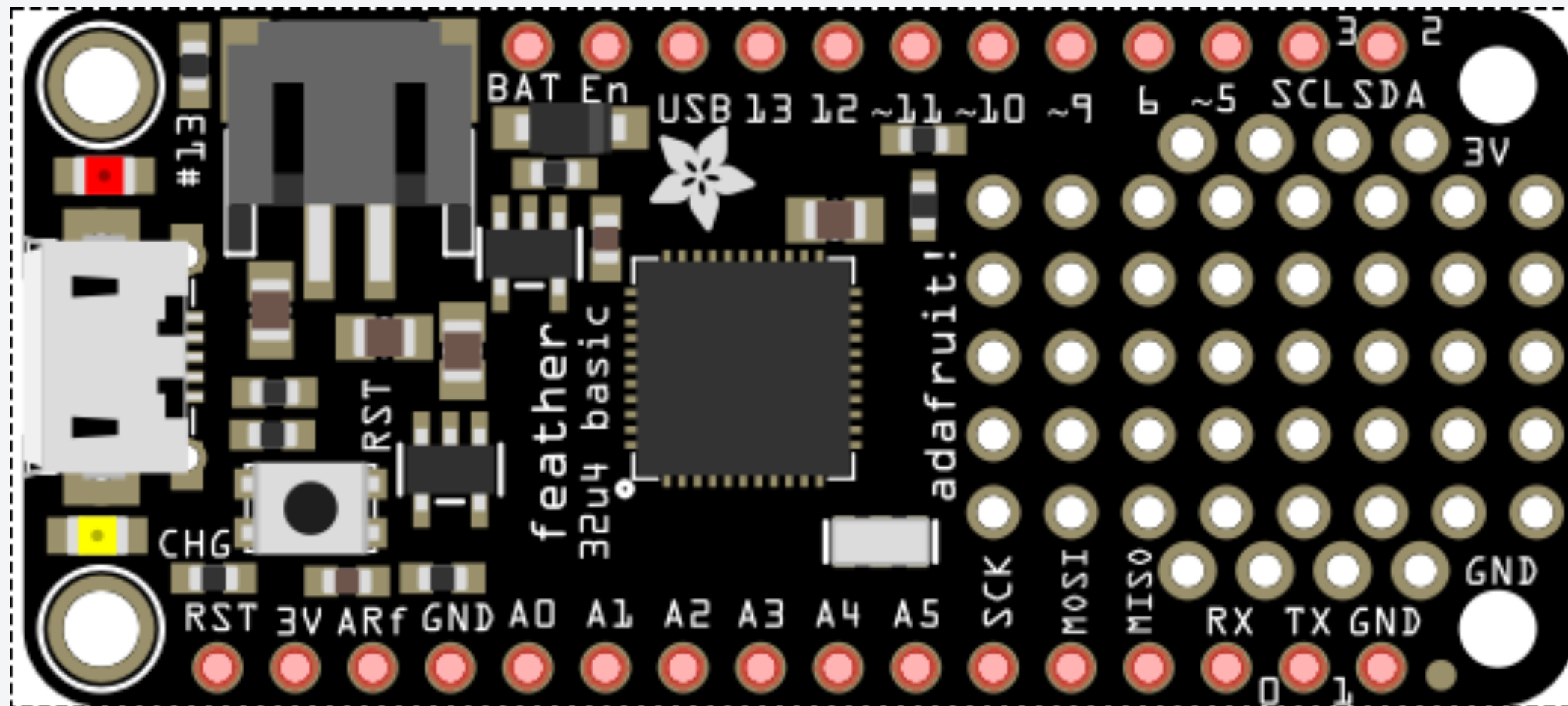


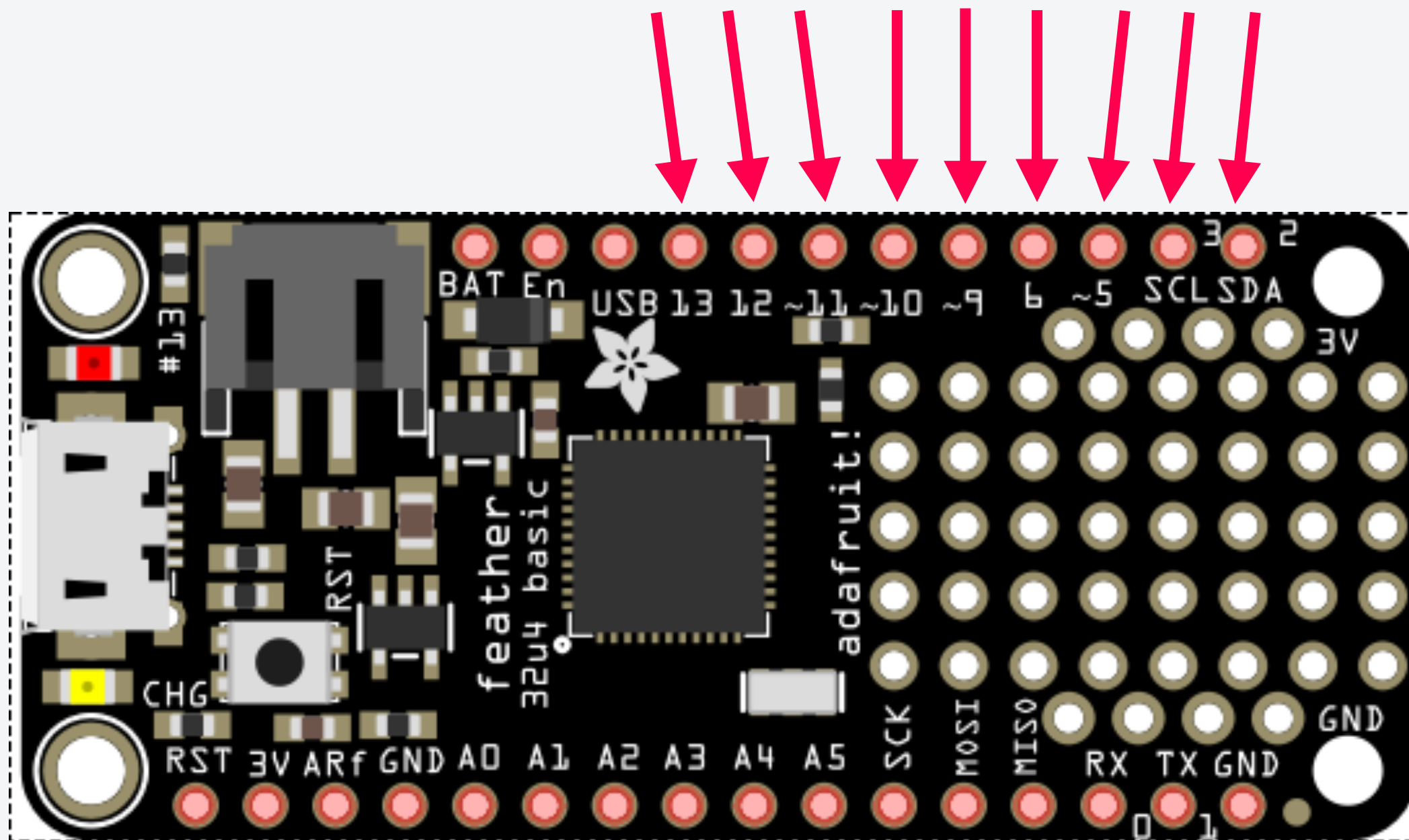


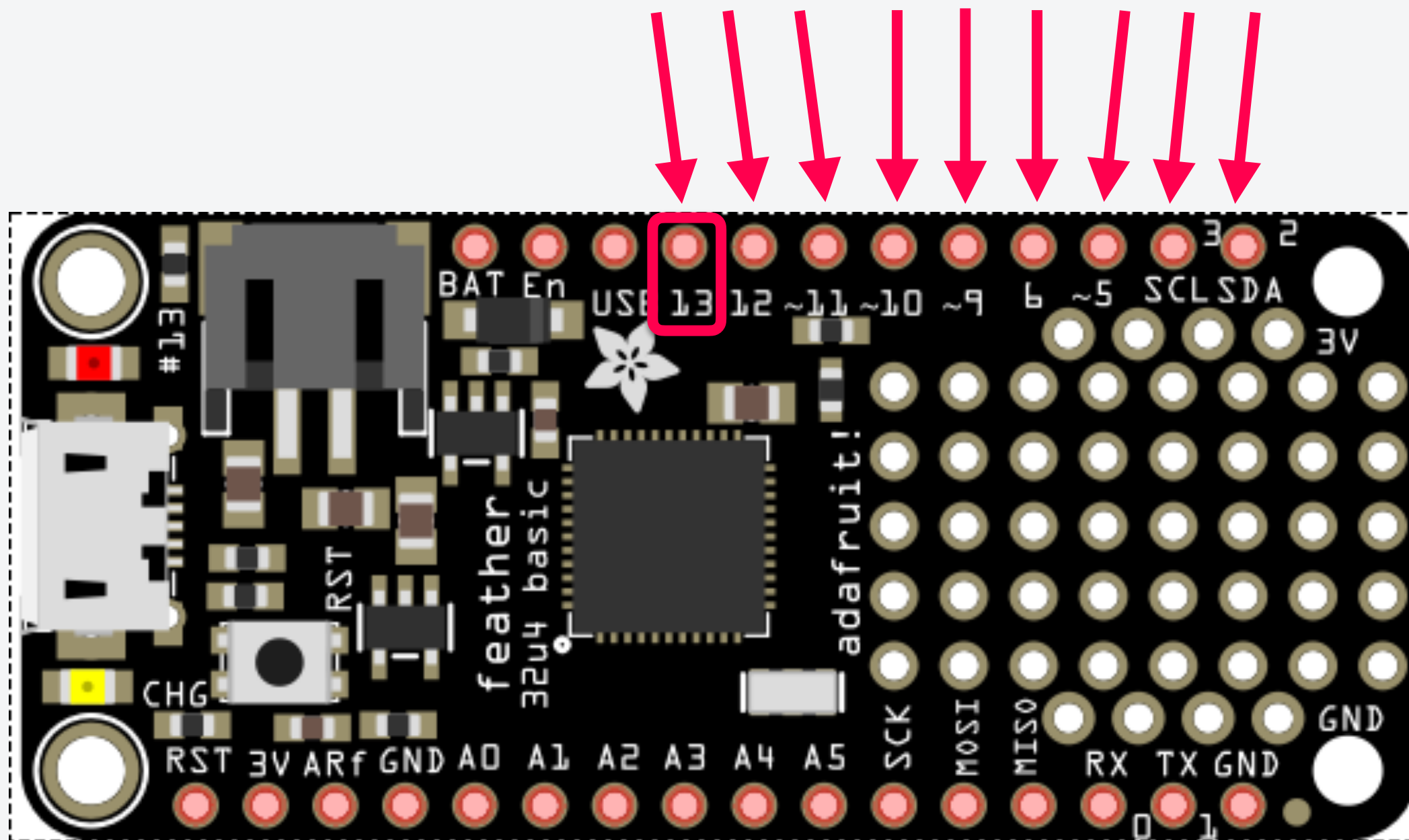






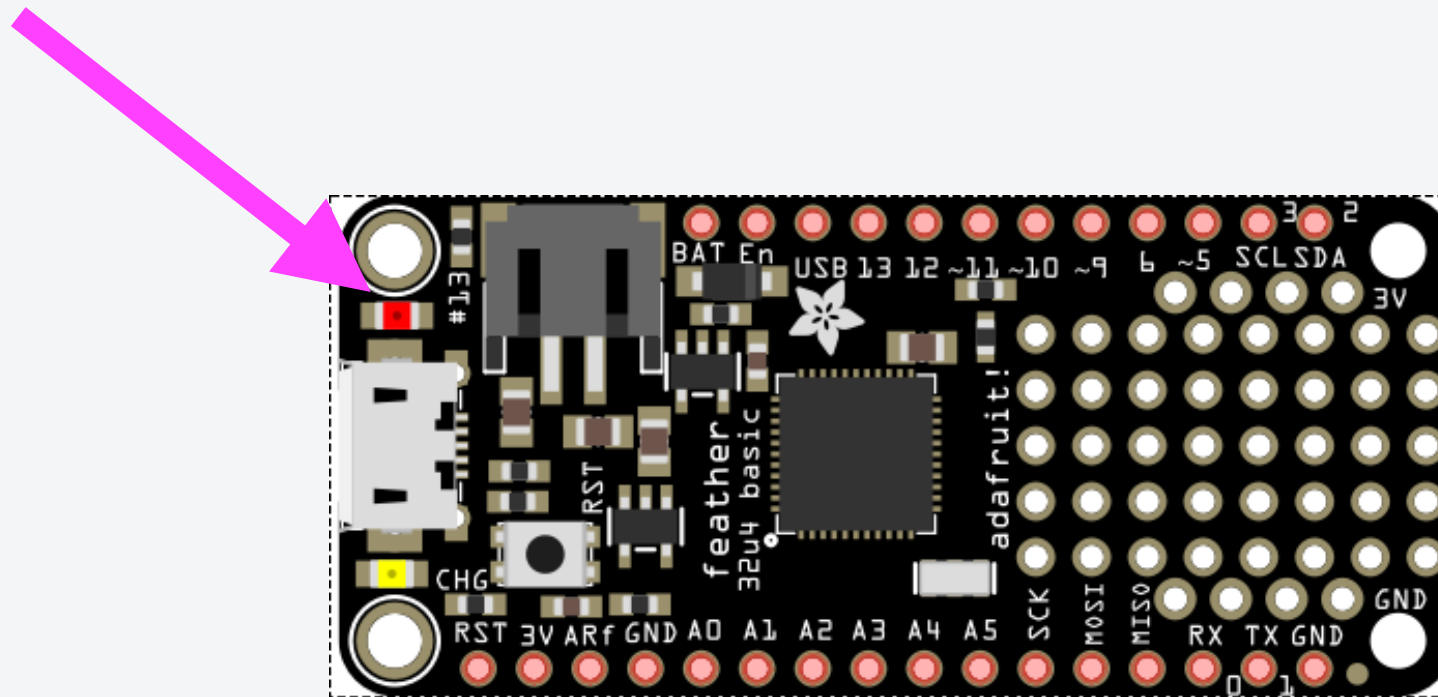




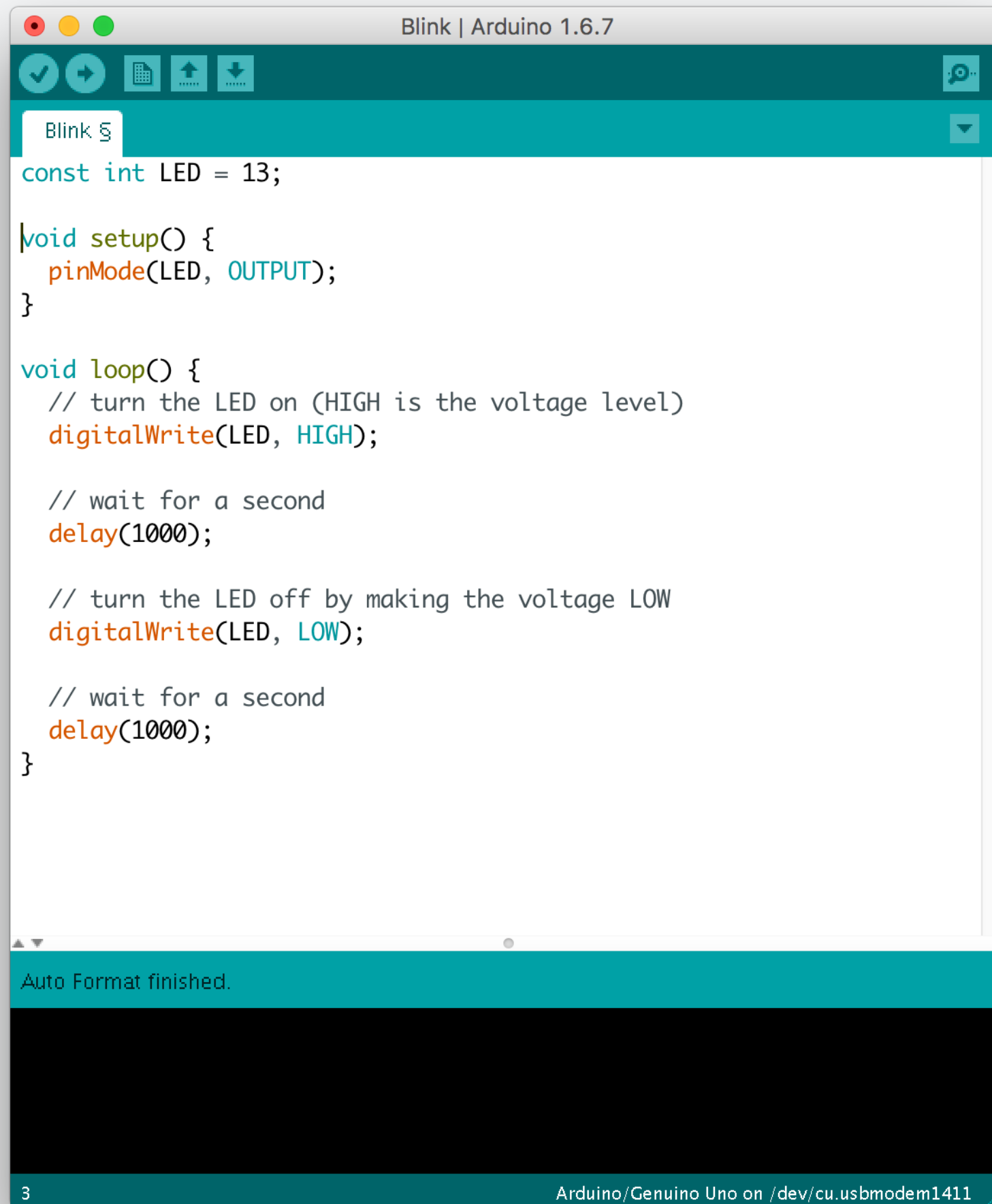


HELLO LED!

BLINK



BLINK



The image shows a screenshot of the Arduino IDE interface. The title bar at the top reads "Blink | Arduino 1.6.7". Below the title bar is a toolbar with icons for checking, running, saving, and uploading. The main text area contains the following C++ code for the Blink sketch:

```
const int LED = 13;

void setup() {
  pinMode(LED, OUTPUT);
}

void loop() {
  // turn the LED on (HIGH is the voltage level)
  digitalWrite(LED, HIGH);

  // wait for a second
  delay(1000);

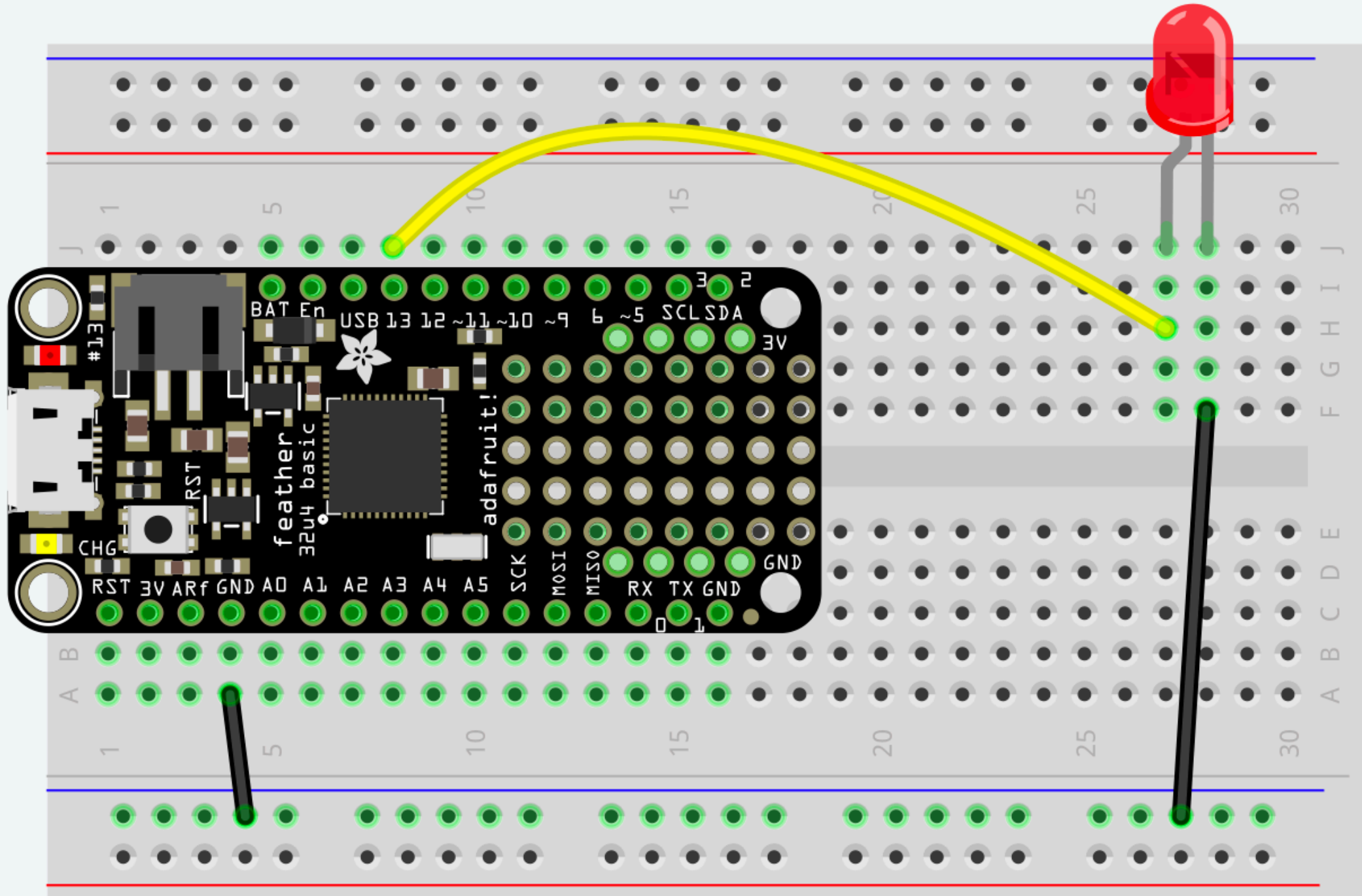
  // turn the LED off by making the voltage LOW
  digitalWrite(LED, LOW);

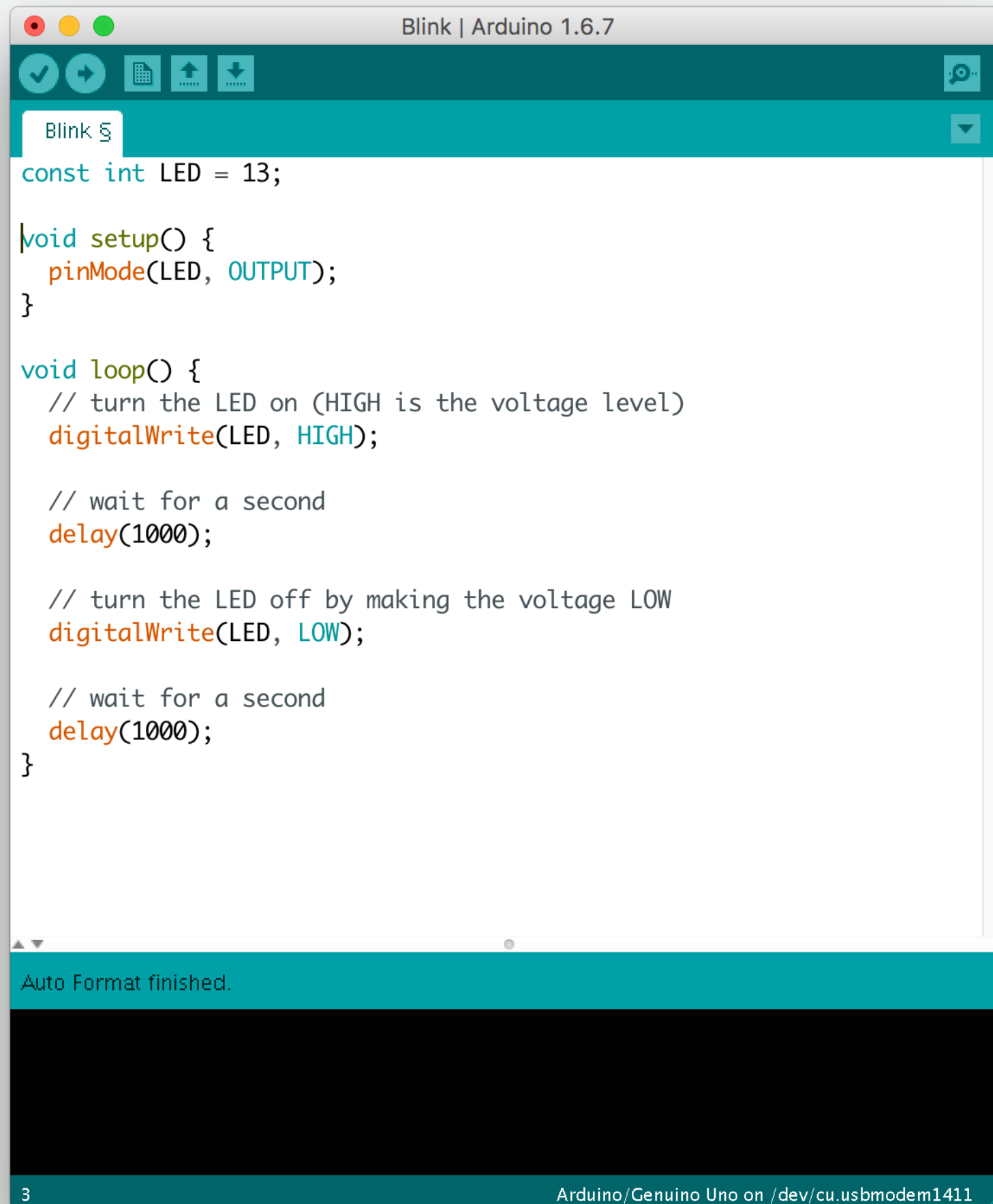
  // wait for a second
  delay(1000);
}
```

Below the code editor, a status bar shows "Auto Format finished." and a black console area. At the very bottom, a teal bar displays the page number "3" and the connection information "Arduino/Genuino Uno on /dev/cu.usbmodem1411".

Try changing the blink frequency, then try creating a pattern.

BLINK 2



A screenshot of the Arduino IDE interface. The title bar at the top reads "Blink | Arduino 1.6.7". Below the title bar is a toolbar with icons for checking, running, uploading, and downloading. The main text area contains the following code:

```
const int LED = 13;

void setup() {
  pinMode(LED, OUTPUT);
}

void loop() {
  // turn the LED on (HIGH is the voltage level)
  digitalWrite(LED, HIGH);

  // wait for a second
  delay(1000);

  // turn the LED off by making the voltage LOW
  digitalWrite(LED, LOW);

  // wait for a second
  delay(1000);
}
```

A status bar at the bottom of the IDE shows "Auto Format finished." on the left and "3" on the right. The bottom-most status bar indicates "Arduino/Genuino Uno on /dev/cu.usbmodem1411".

```
Blink | Arduino 1.6.7
```

```
const int LED = 13;

void setup() {
  pinMode(LED, OUTPUT);
}

void loop() {
  // turn the LED on (HIGH is the voltage level)
  digitalWrite(LED, HIGH);

  // wait for a second
  delay(1000);

  // turn the LED off by making the voltage LOW
  digitalWrite(LED, LOW);

  // wait for a second
  delay(1000);
}
```

```
Auto Format finished.
```

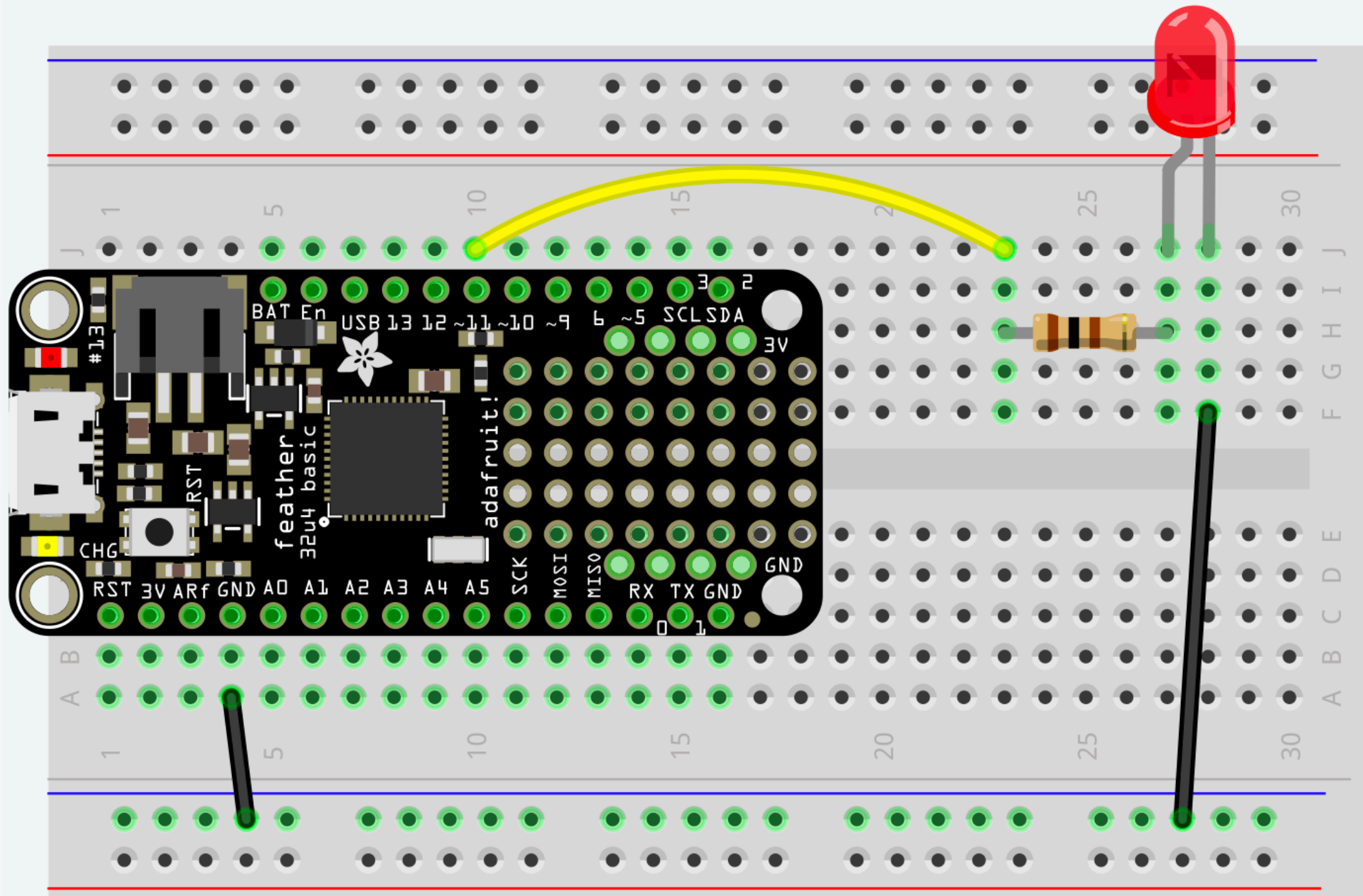
```
3
```

```
Arduino/Genuino Uno on /dev/cu.usbmodem1411
```

BLINK 2

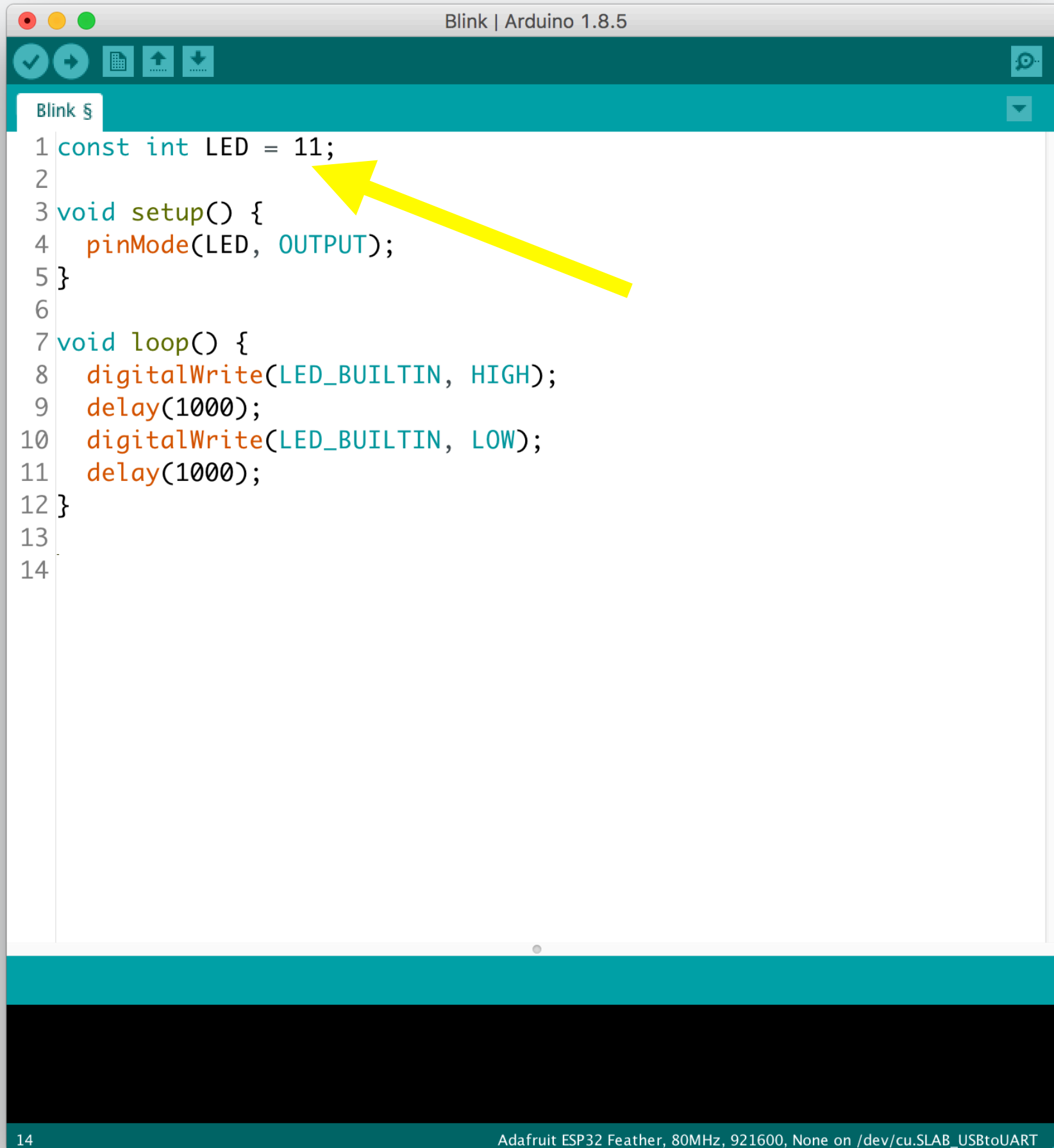
Notice
that the
code is
the same
as Blink 1

BLINK 3



* Resistor is 100Ω (Brown Black Brown)

BLINK 3



```
Blink | Arduino 1.8.5  
Blink §  
1 const int LED = 11;  
2  
3 void setup() {  
4   pinMode(LED, OUTPUT);  
5 }  
6  
7 void loop() {  
8   digitalWrite(LED_BUILTIN, HIGH);  
9   delay(1000);  
10  digitalWrite(LED_BUILTIN, LOW);  
11  delay(1000);  
12 }  
13  
14
```

14

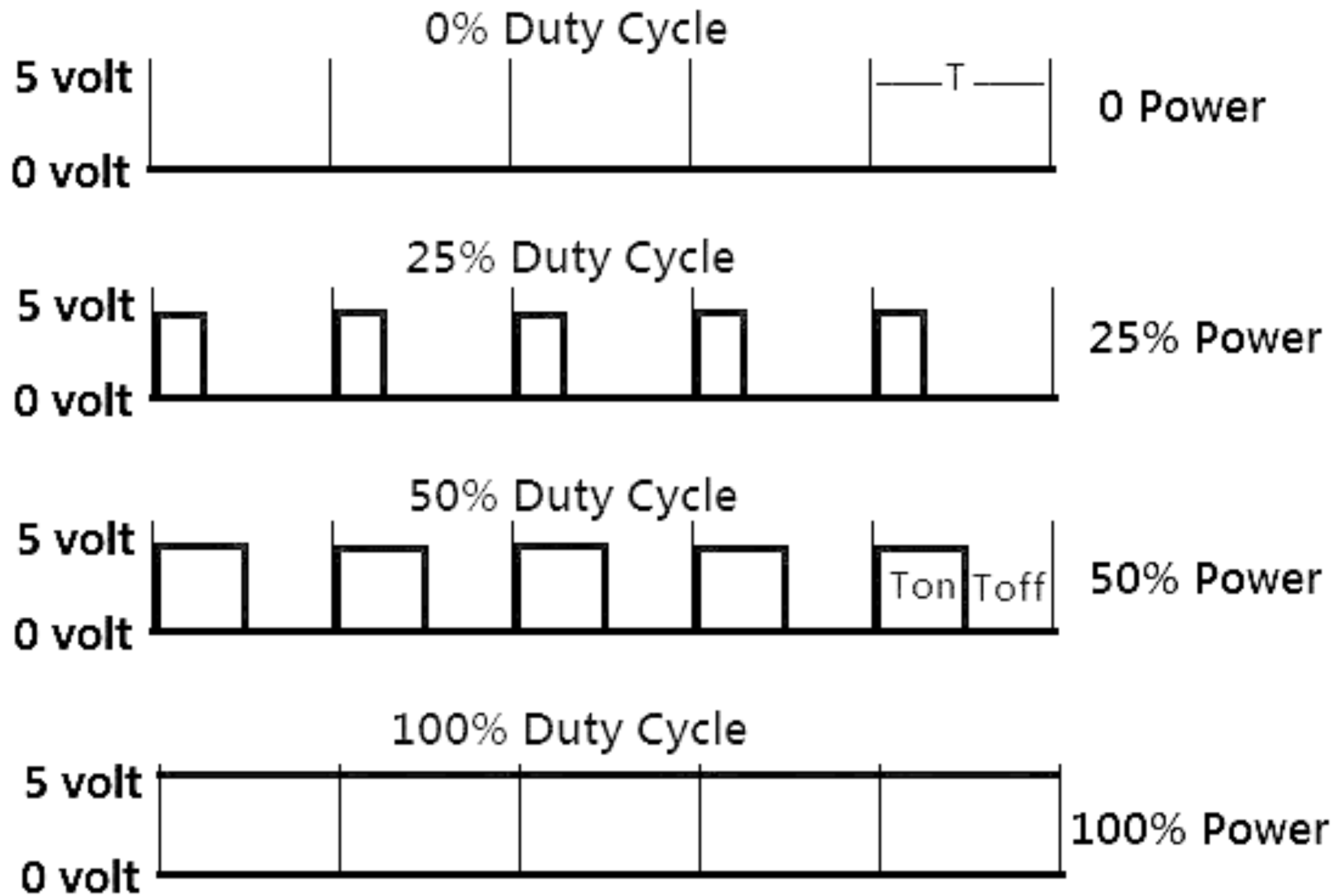
Adafruit ESP32 Feather, 80MHz, 921600, None on /dev/cu.SLAB_USBtoUART

Try connecting more LEDs to other pins. What patterns can you create? What limits/complications are caused by using the delay function?

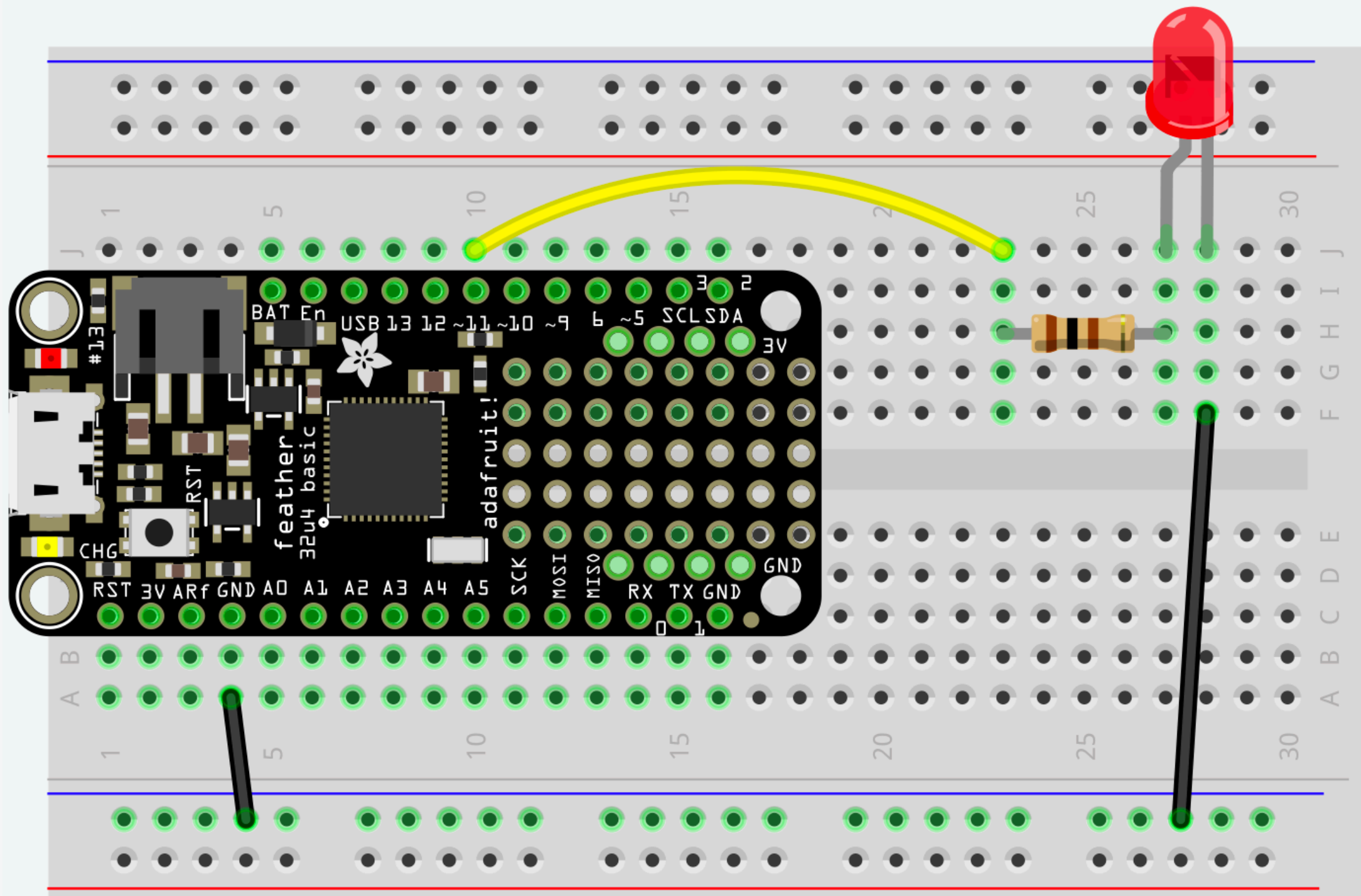
ANALOG OUT

(PULSE WIDTH MODULATION)

PULSE WIDTH MODULATION



PWM LED



* Resistor is 100Ω (Brown Black Brown)

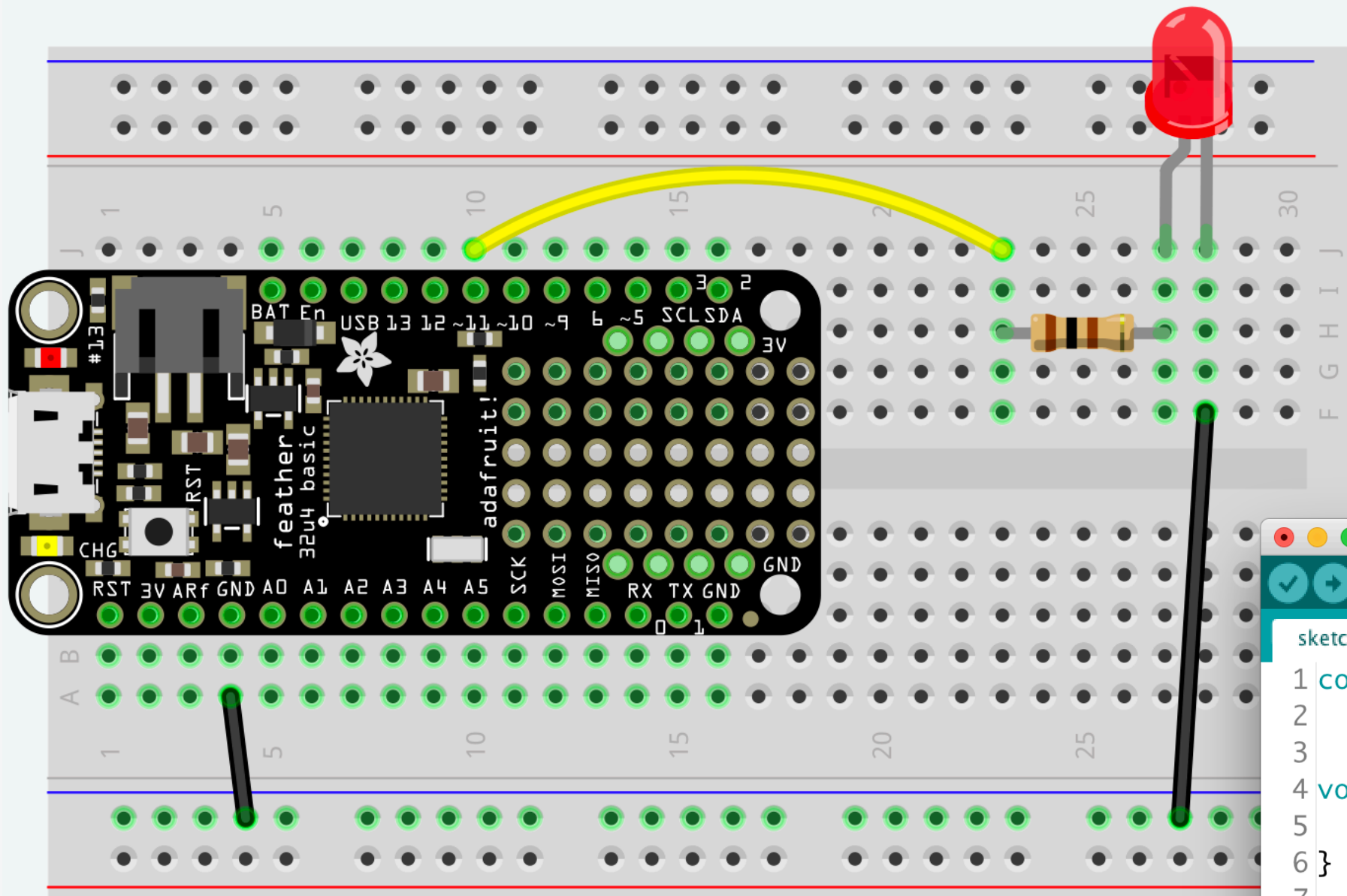
PWM LED

```
sketch_jan16a | Arduino 1.8.5
sketch_jan16a $
1 const int LED = 11;
2
3
4 void setup() {
5   pinMode(LED, OUTPUT);
6 }
7
8 void loop() {
9   for (int i = 0; i <= 255; i += 5) {
10    analogWrite(LED, i);
11    delay(30);
12  }
13
14  for (int i = 255; i >= 0; i -= 5) {
15    analogWrite(LED, i);
16    delay(30);
17  }
18 }
19
20
21
```

Auto Format finished.

21 Adafruit ESP32 Feather, 80MHz, 921600, None on /dev/cu.SLAB_USBtoUART

PWM LED



```
sketch_jan16a | Ar  
sketch_jan16a §  
1 const int LED = 11;  
2  
3  
4 void setup() {  
5   pinMode(LED, OUTPUT);  
6 }  
7  
8 void loop() {  
9   for (int i = 0; i <= 255; i += 5) {  
10     analogWrite(LED, i);  
11     delay(30);  
12   }  
13  
14   for (int i = 255; i >= 0; i -= 5) {  
15     analogWrite(LED, i);  
16     delay(30);  
17   }  
18 }  
19  
20  
21
```