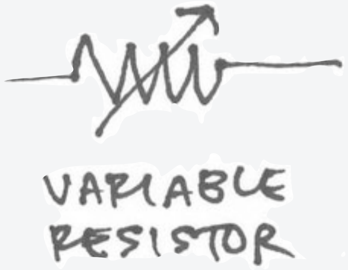
ANALOG INPUT

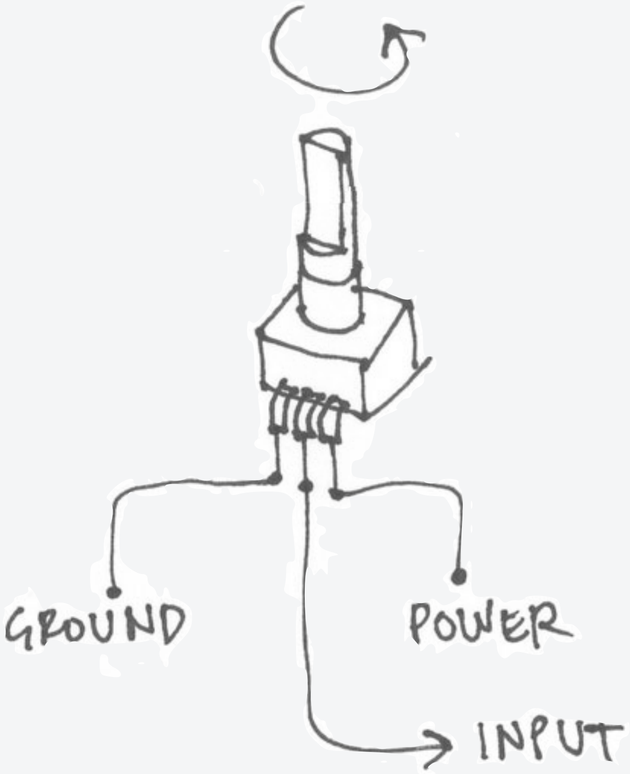
THE INPUTS ON THE ARDUINO READ VOLTAGE.

ALL INPUTS NEED TO BE THOUGHT OF IN TERMS OF VOLTAGE DIFFERENTIALS.

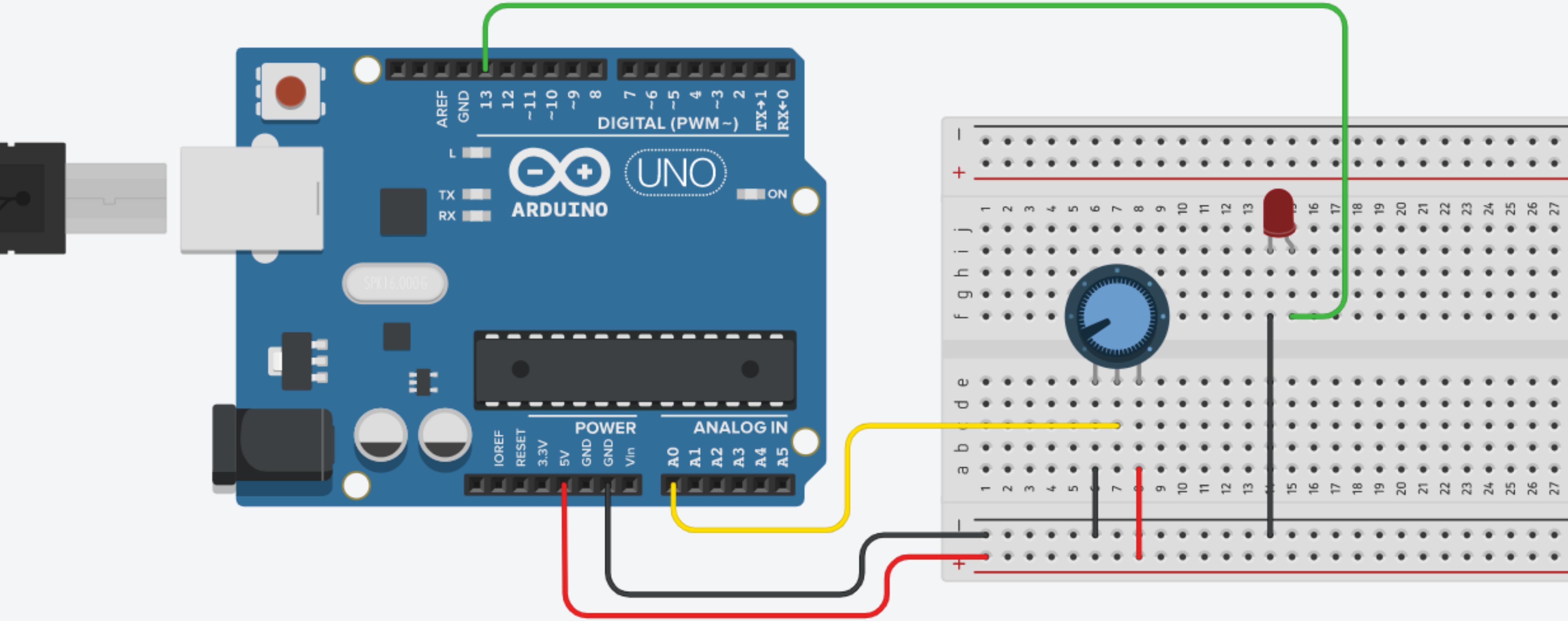
THE ANALOG INPUTS CONVERT VOLTAGE LEVELS TO A NUMERICAL VALUE.

POTENTIOMETER

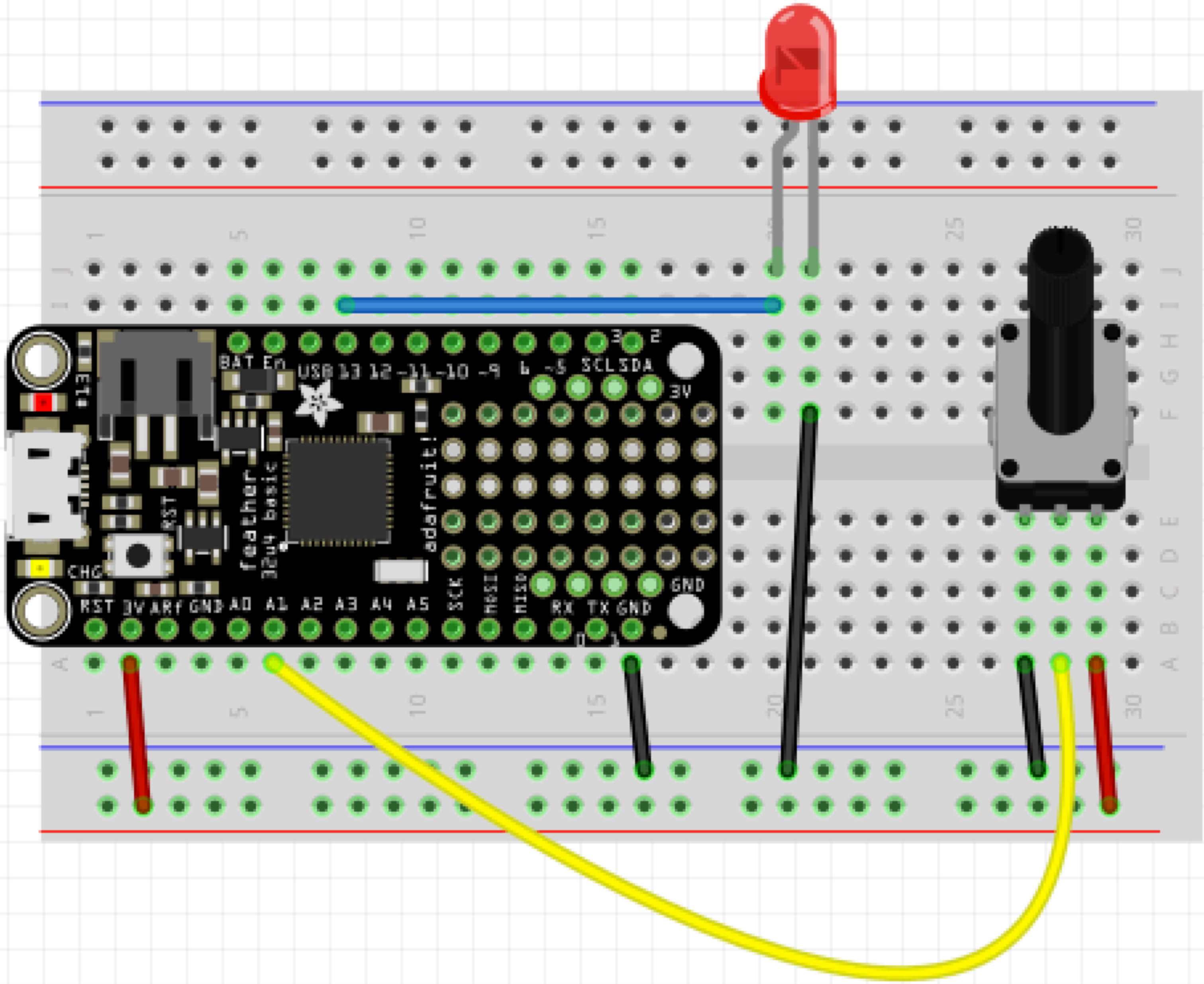




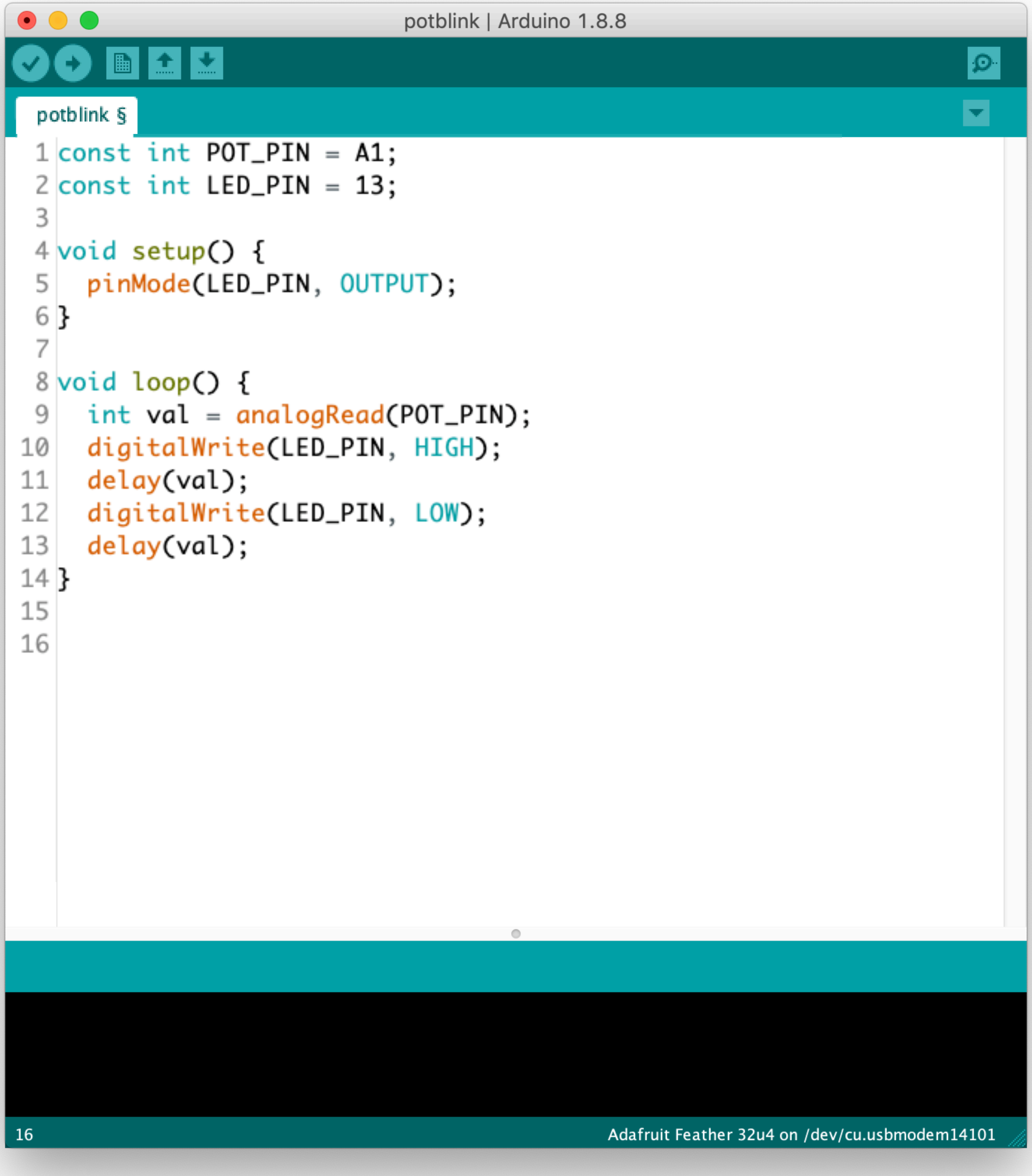
POTENTIOMETER WITH BLINKING LED



POTENTIOMETER WITH BLINKING LED


BAT En USB 13 12 11 10 9 6 5 SCL SDA
3 2
3V
#13
RST
CHG
feather 32u4 basic
adafruit!
RST 3V ARf GND A0 A1 A2 A3 A4 A5 SCK MOSI MISO RX TX GND
0 1
GND
1 5 10 15 20 25 30
J I H G F E D C B A

READ ANALOG VALUES AND BLINK LED

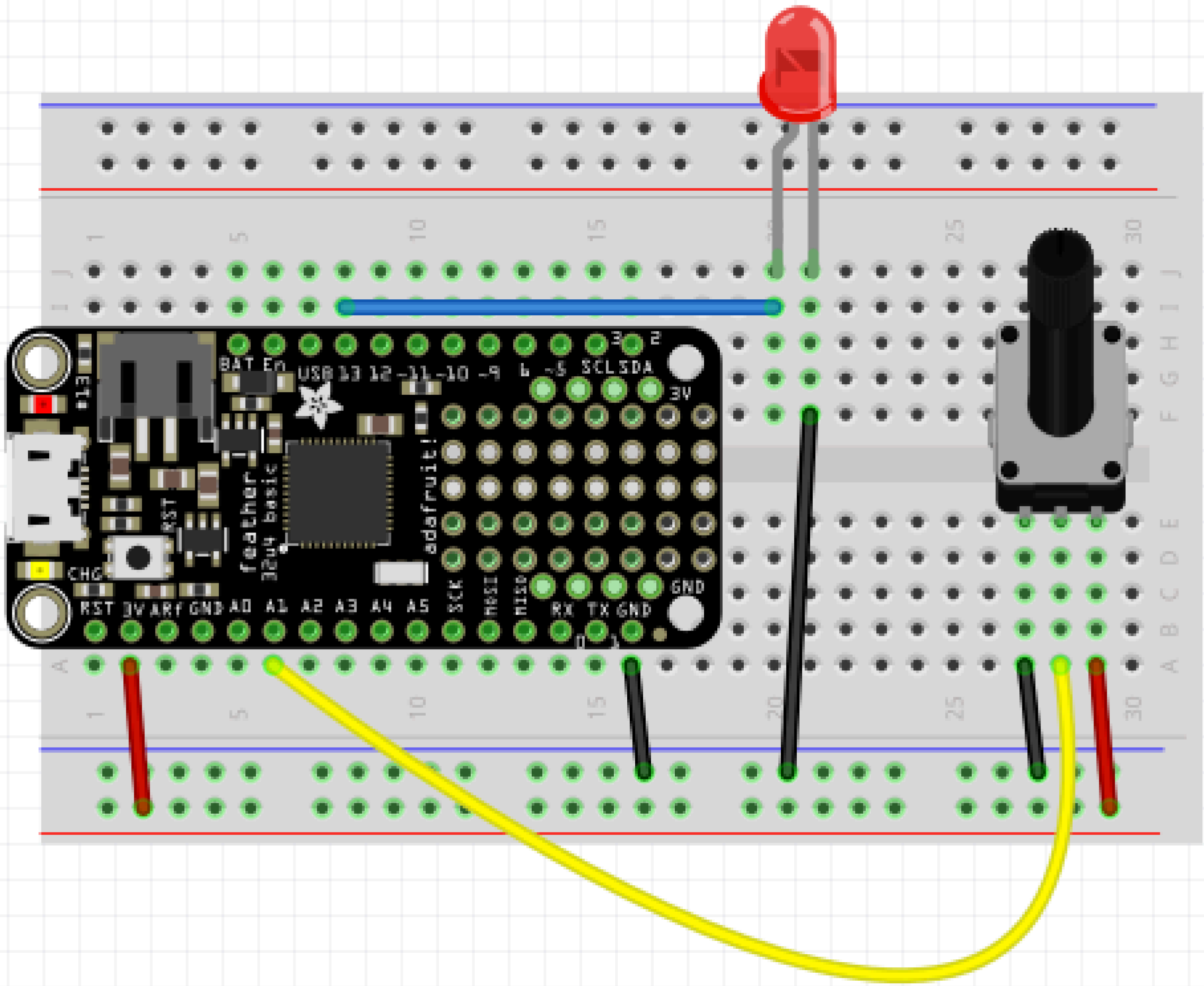


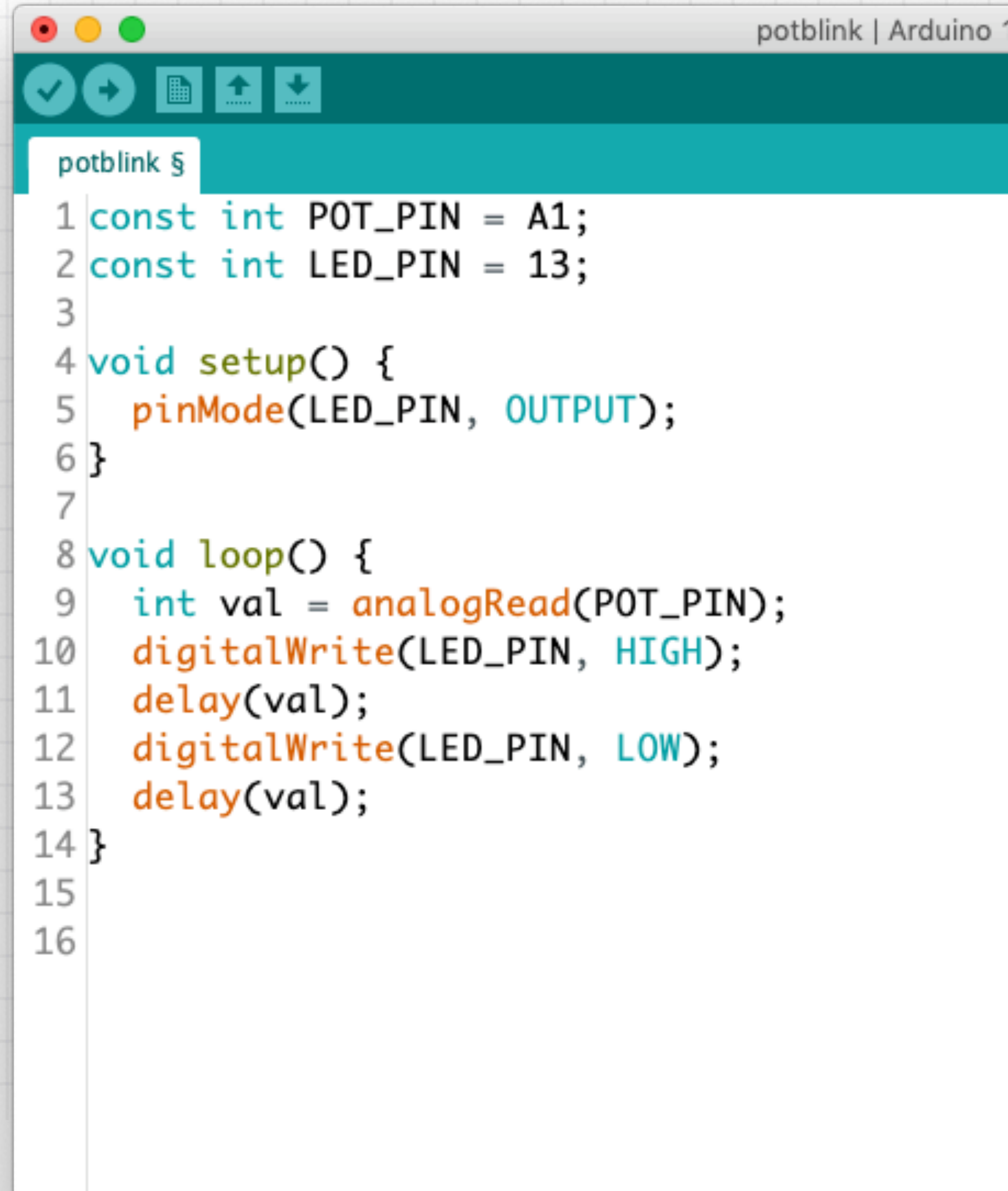
```
const int POT_PIN = A1;
const int LED_PIN = 13;

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

void loop() {
  int val = analogRead(POT_PIN);
  digitalWrite(LED_PIN, HIGH);
  delay(val);
  digitalWrite(LED_PIN, LOW);
  delay(val);
}
```

analogRead() gives values in a range from 0-1024.

READ ANALOG VALUES AND BLINK LED

analogRead() gives values in a range from 0-1024.

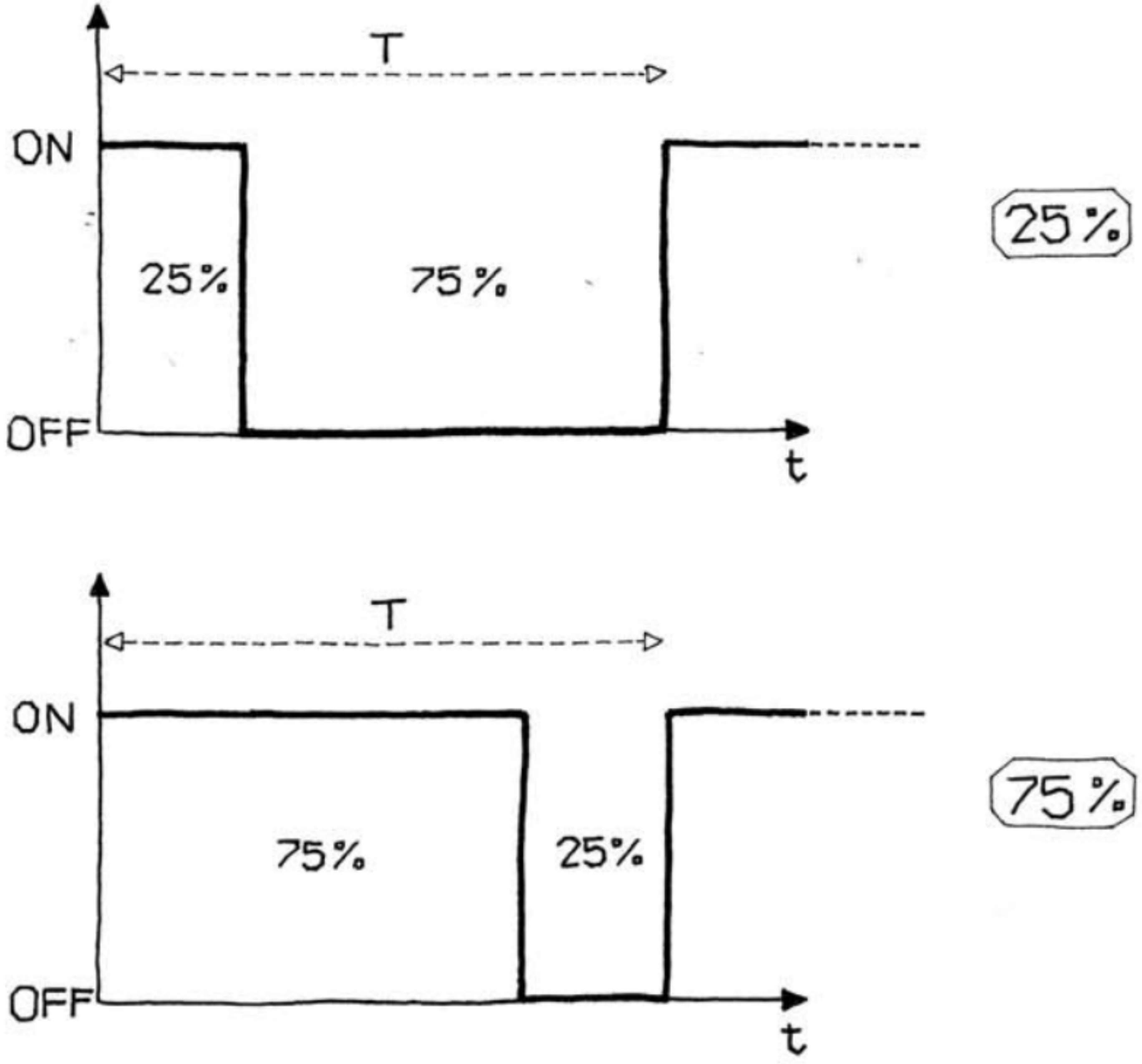

BAT En USB 13 12 11 10 9 6 5 SCL SDA
3V
GND
RST 3V ARf GND A0 A1 A2 A3 A4 A5 SCK MOSI MISO RX TX GND
feather 32u4 basic
adafruit!
RST
CHG
#13
1
5
10
15
20
25
30
A
B
C
D
E
F
G
H
I
J



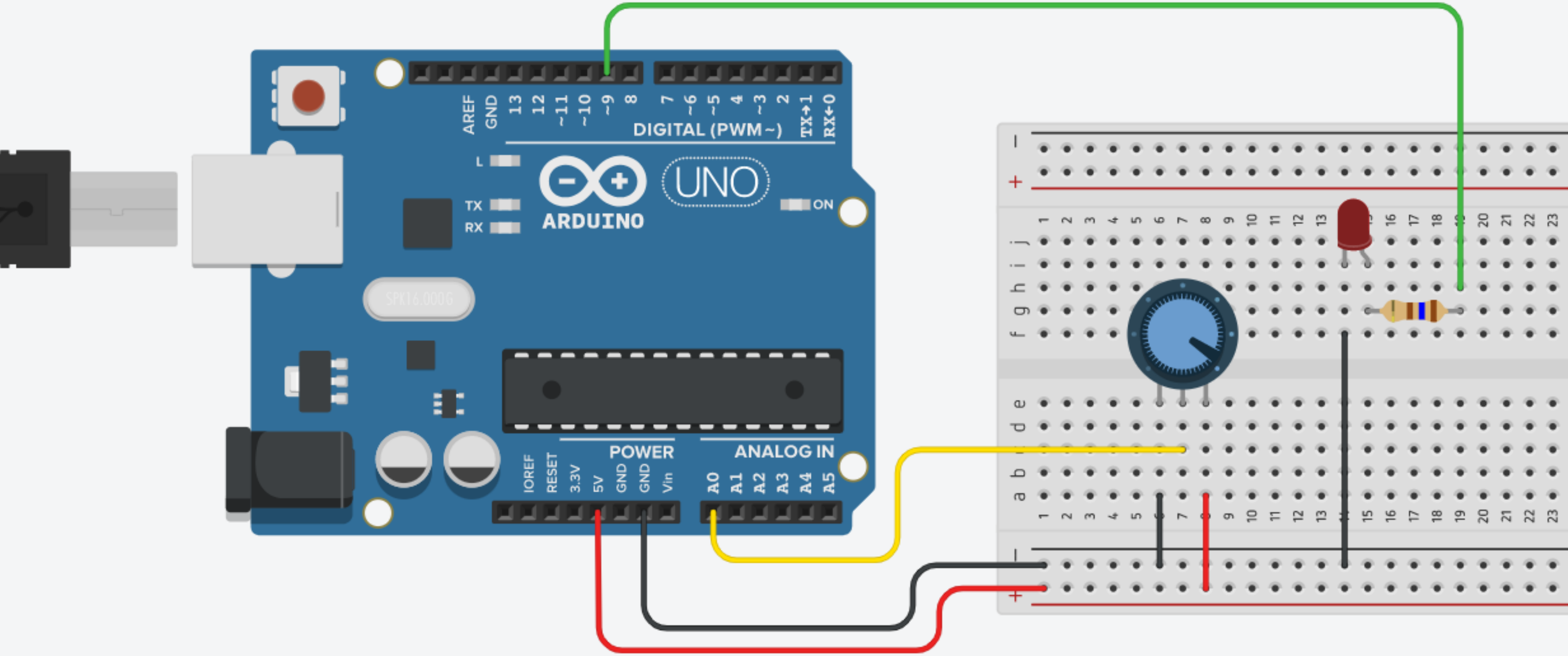
```
const int POT_PIN = A1;
const int LED_PIN = 13;

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

void loop() {
  int val = analogRead(POT_PIN);
  digitalWrite(LED_PIN, HIGH);
  delay(val);
  digitalWrite(LED_PIN, LOW);
  delay(val);
}
```

ANALOG INPUT WITH PULSE WIDTH MODULATION

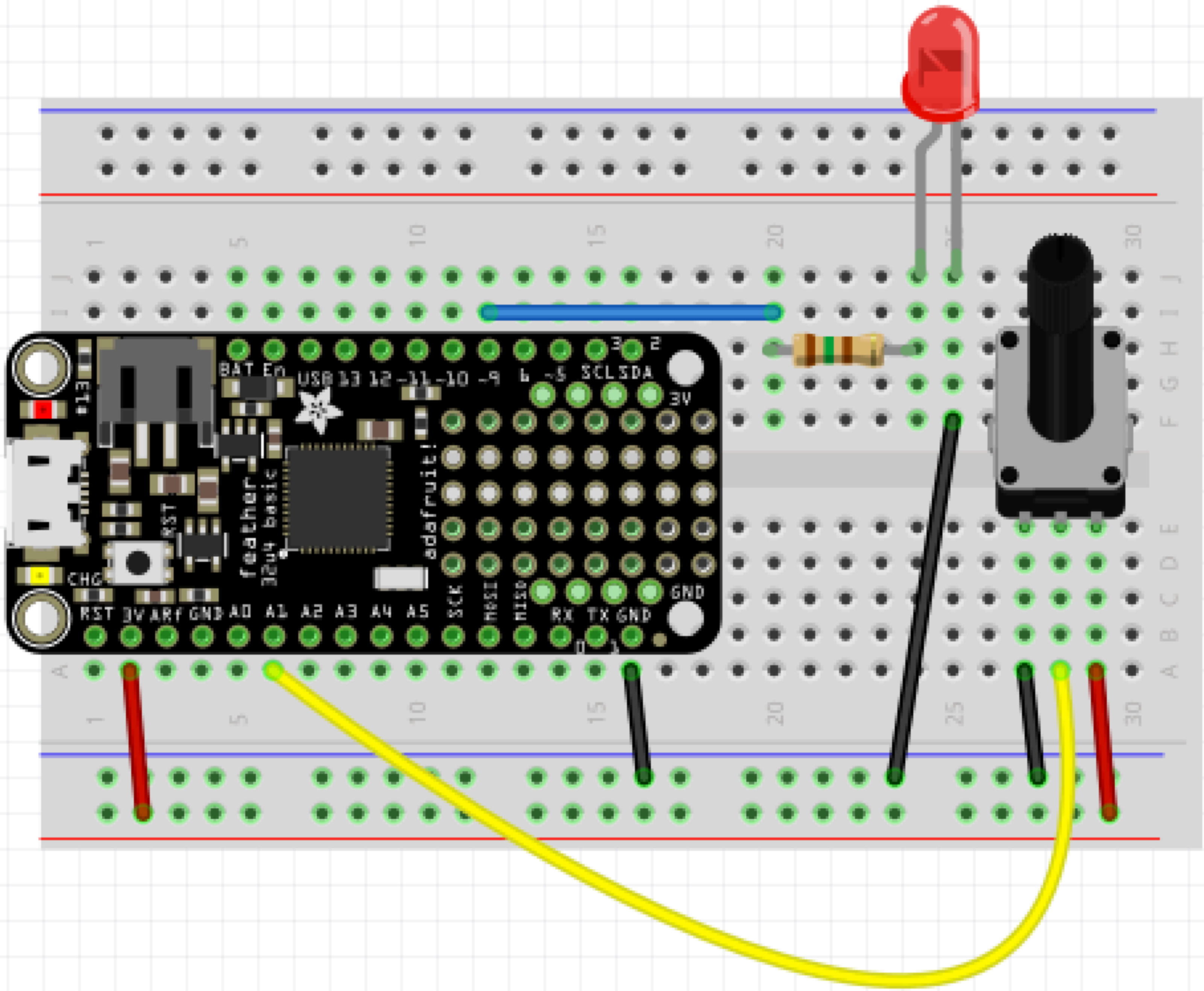
PULSE WIDTH MODULATION



POTENTIOMETER WITH PWM LED

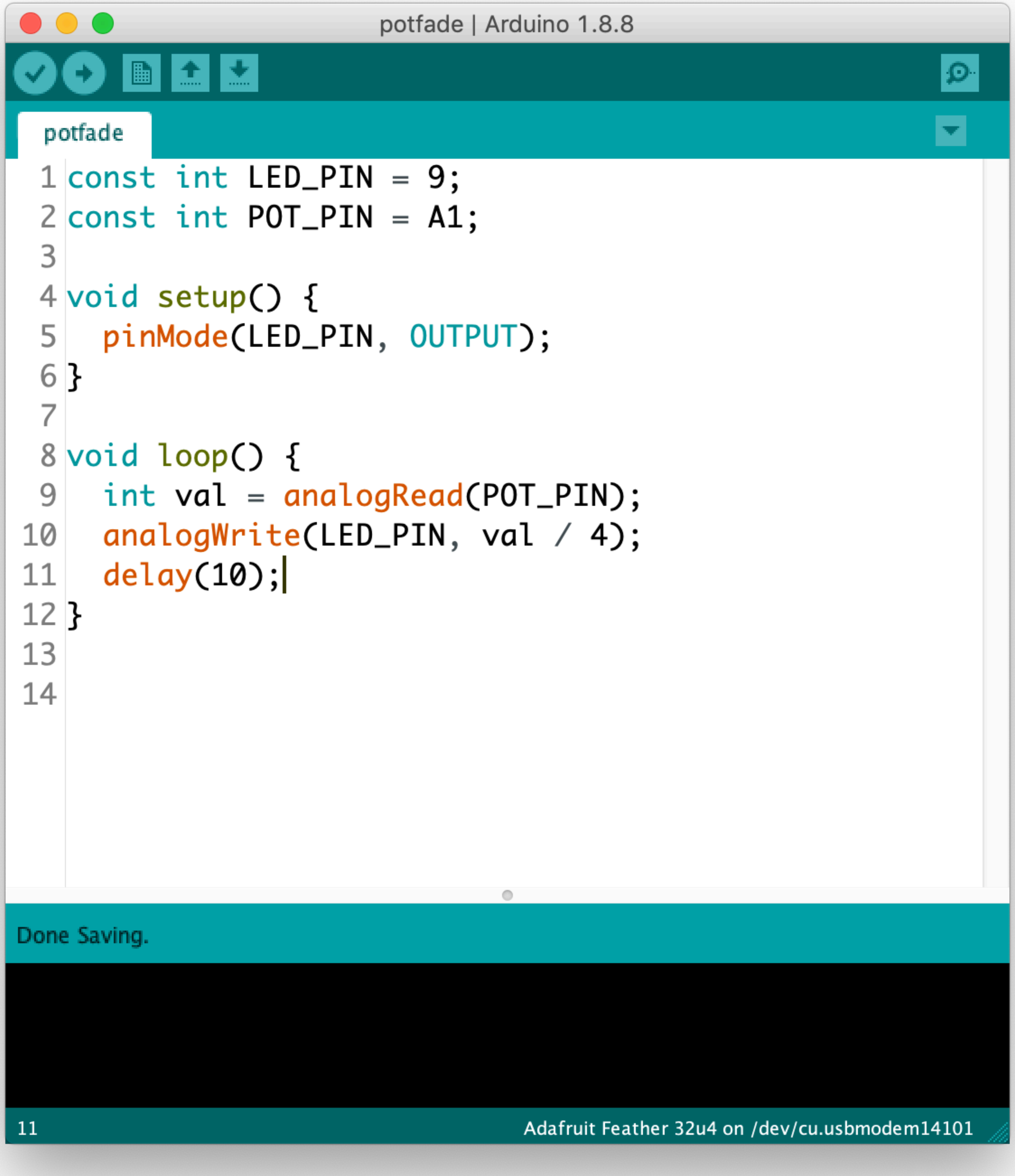


POTENTIOMETER WITH PWM LED



BAT En USB 13 12 ~11 ~10 ~9 6 ~5 SCL SDA
3V
#13
CHG
RST
feather
32u4 basic
adafruit!
RST 3V ARf GND A0 A1 A2 A3 A4 A5 SCK MOSI MISO RX TX GND
GND
1 5 10 15 20 25 30
J I H G F E D C B A

READ ANALOG VALUES AND FADE LED



```
const int LED_PIN = 9;
const int POT_PIN = A1;

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

void loop() {
  int val = analogRead(POT_PIN);
  analogWrite(LED_PIN, val / 4);
  delay(10);
}
```

Done Saving.

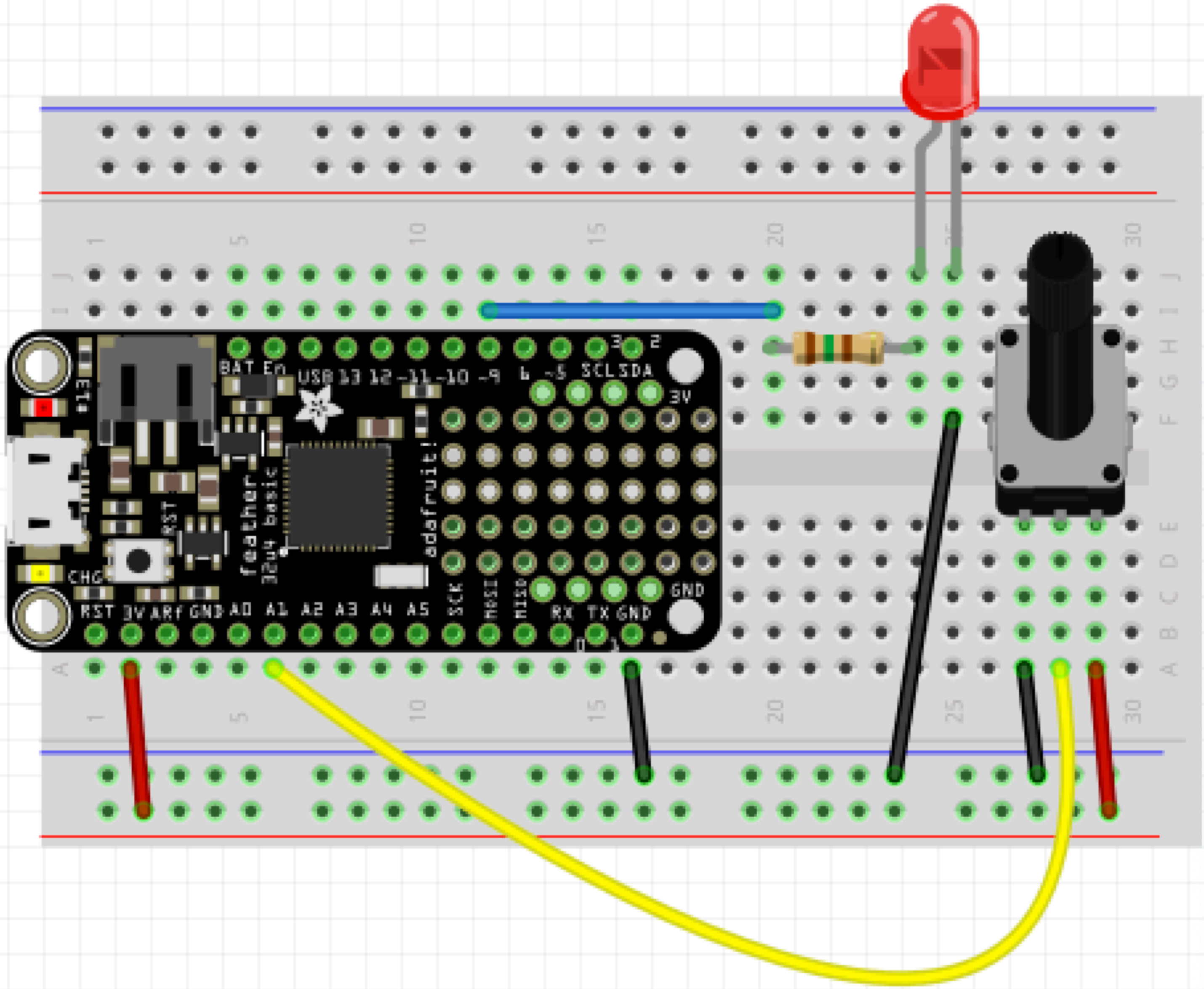
11   Adafruit Feather 32u4 on /dev/cu.usbmodem14101

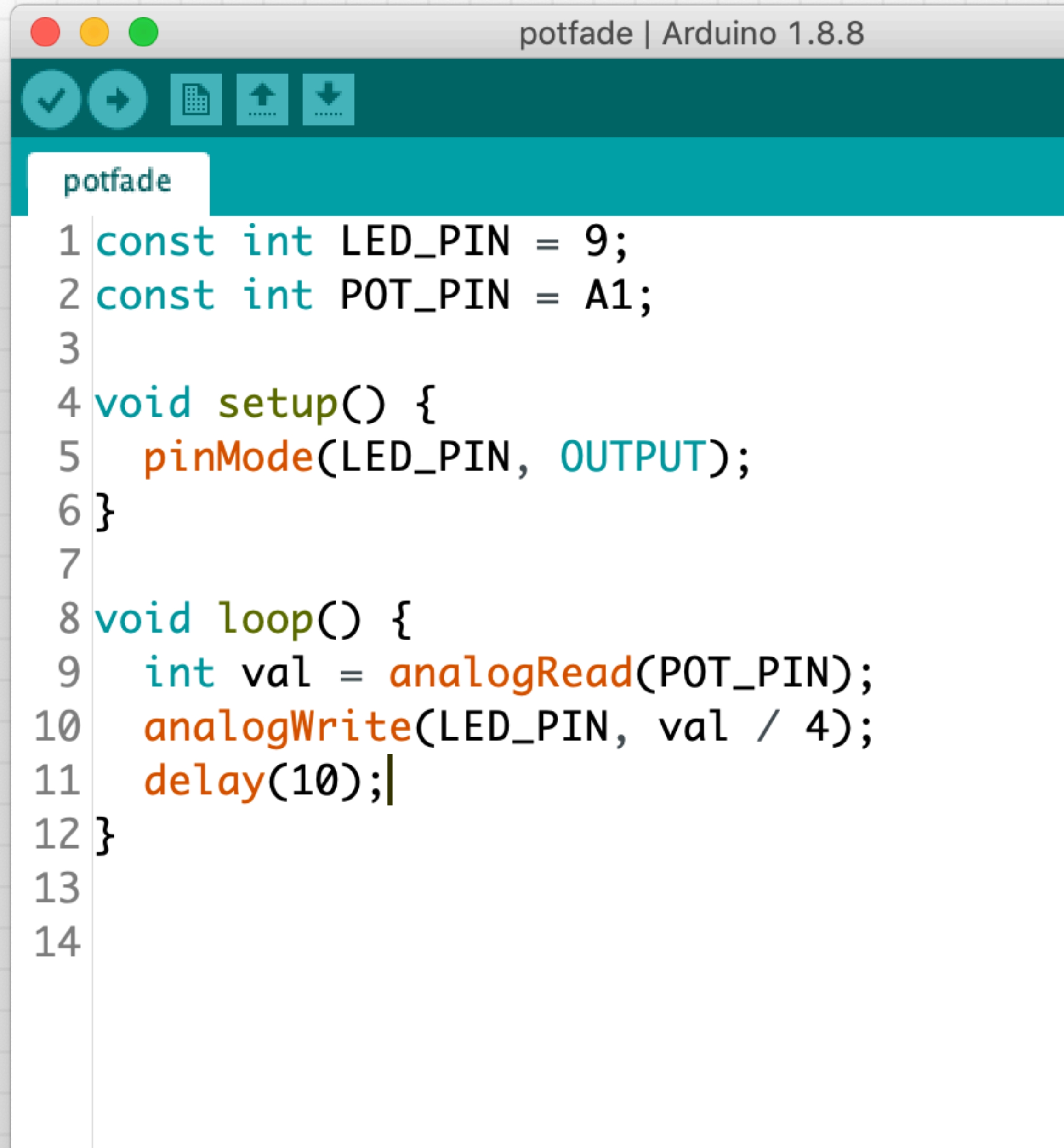
analogRead() gives values in a range from 0-1024.

analogWrite() can only write values from 0-255.

To account for this we divide the value received from analogRead() by 4.

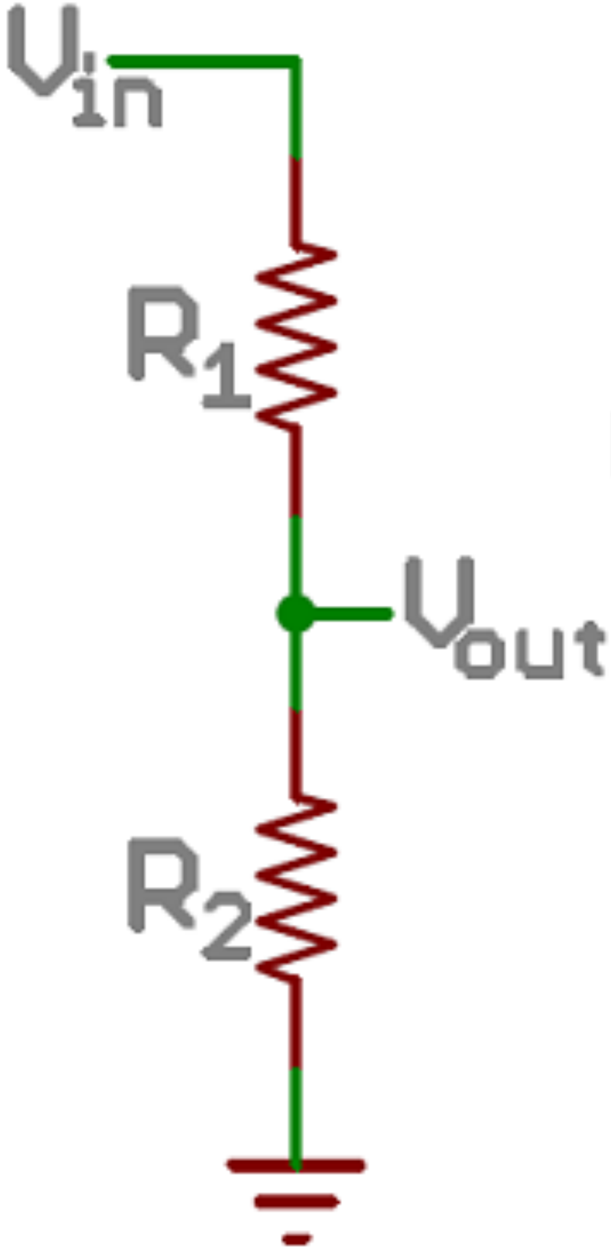
POTENTIOMETER WITH PWM LED


feather 32u4 basic
adafruit!
BAT En USB 13 12 11 10 9 6 5 SCL SDA
RST 3V ARf GND A0 A1 A2 A3 A4 A5 SCK MOSI MISO RX TX GND



```
const int LED_PIN = 9;
const int POT_PIN = A1;

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

void loop() {
  int val = analogRead(POT_PIN);
  analogWrite(LED_PIN, val / 4);
  delay(10);
}
```

VOLTAGE DIVIDER



VOLTAGE DIVIDER

The resistor closest to the input voltage (V_{in}) is called **R_1**, and the resistor closest to ground **R_2**.

The voltage drop across **R_2** is called **V_{out}**, that’s the divided voltage our circuit exists to make.

THE EQUATION

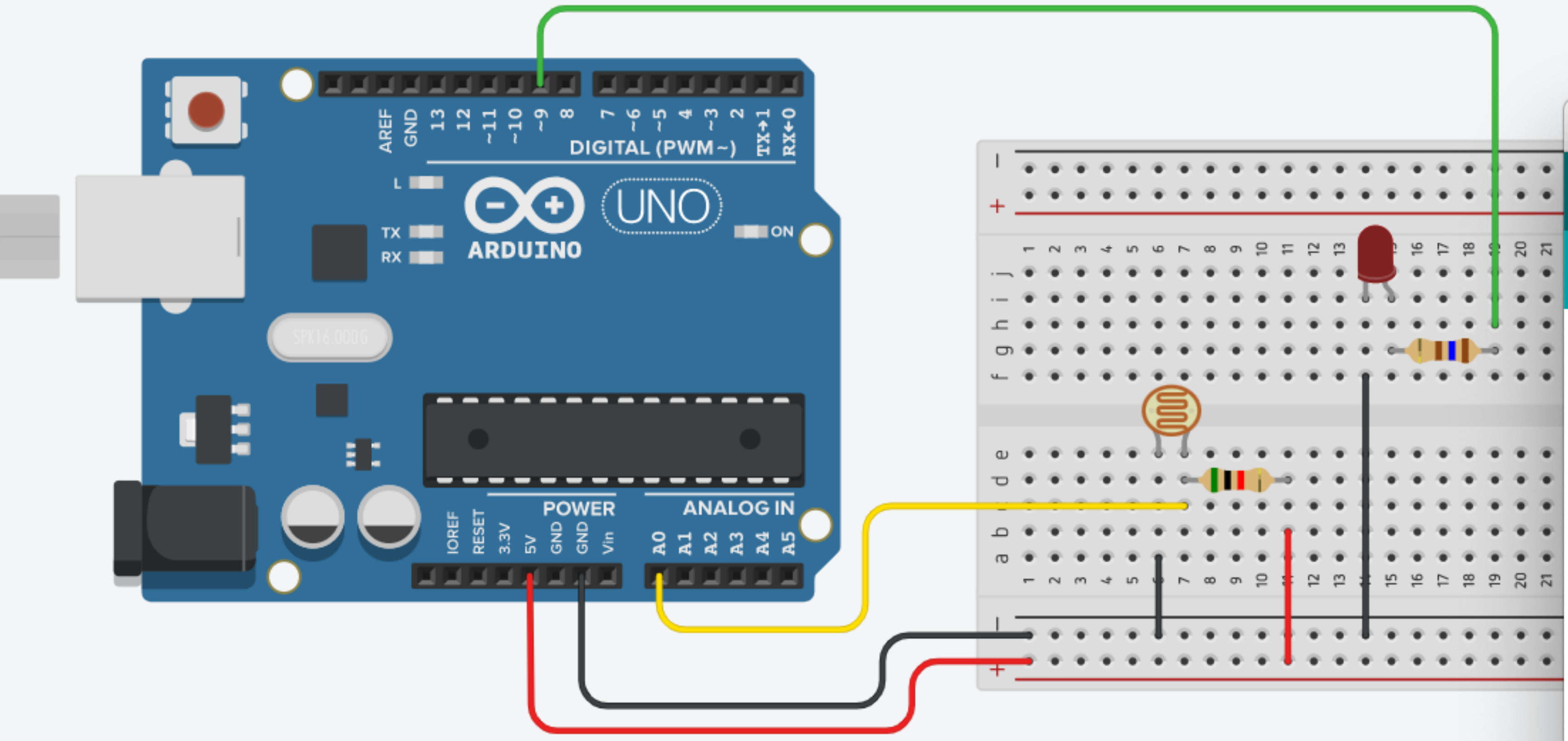
$$V_{out} = V_{in} \cdot \frac{R_2}{R_1 + R_2}$$

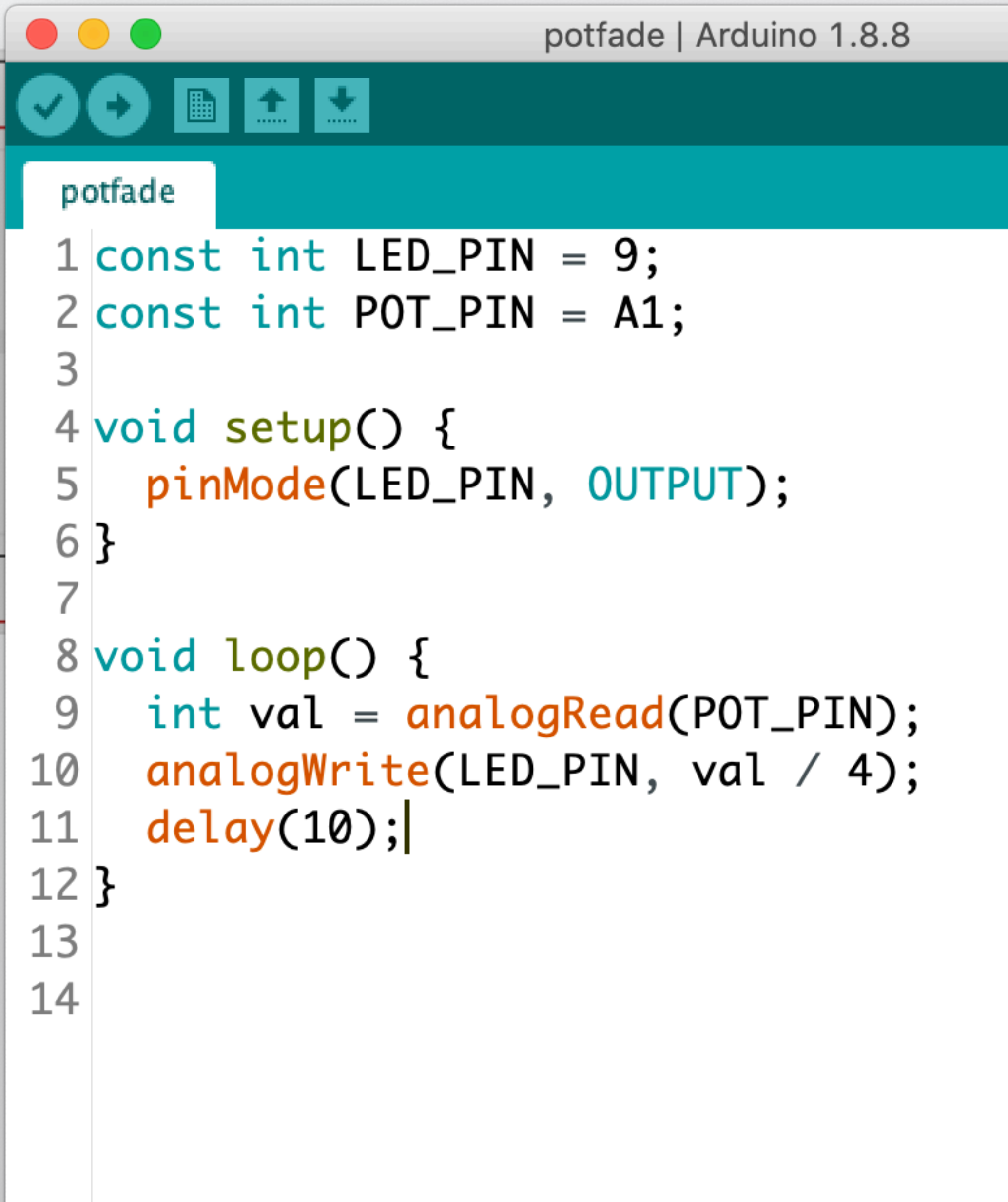
IF R_2 AND R_1 ARE EQUAL
THEN THE OUTPUT VOLTAGE IS HALF THAT OF THE INPUT.

IF R_2 IS MUCH LARGER THAN R_1
(AT LEAST AN ORDER OF MAGNITUDE)
THEN THE OUTPUT VOLTAGE WILL BE VERY CLOSE TO THE INPUT.

IF R_2 IS MUCH SMALLER THAN R_1
THEN OUTPUT VOLTAGE WILL BE TINY COMPARED TO THE INPUT.

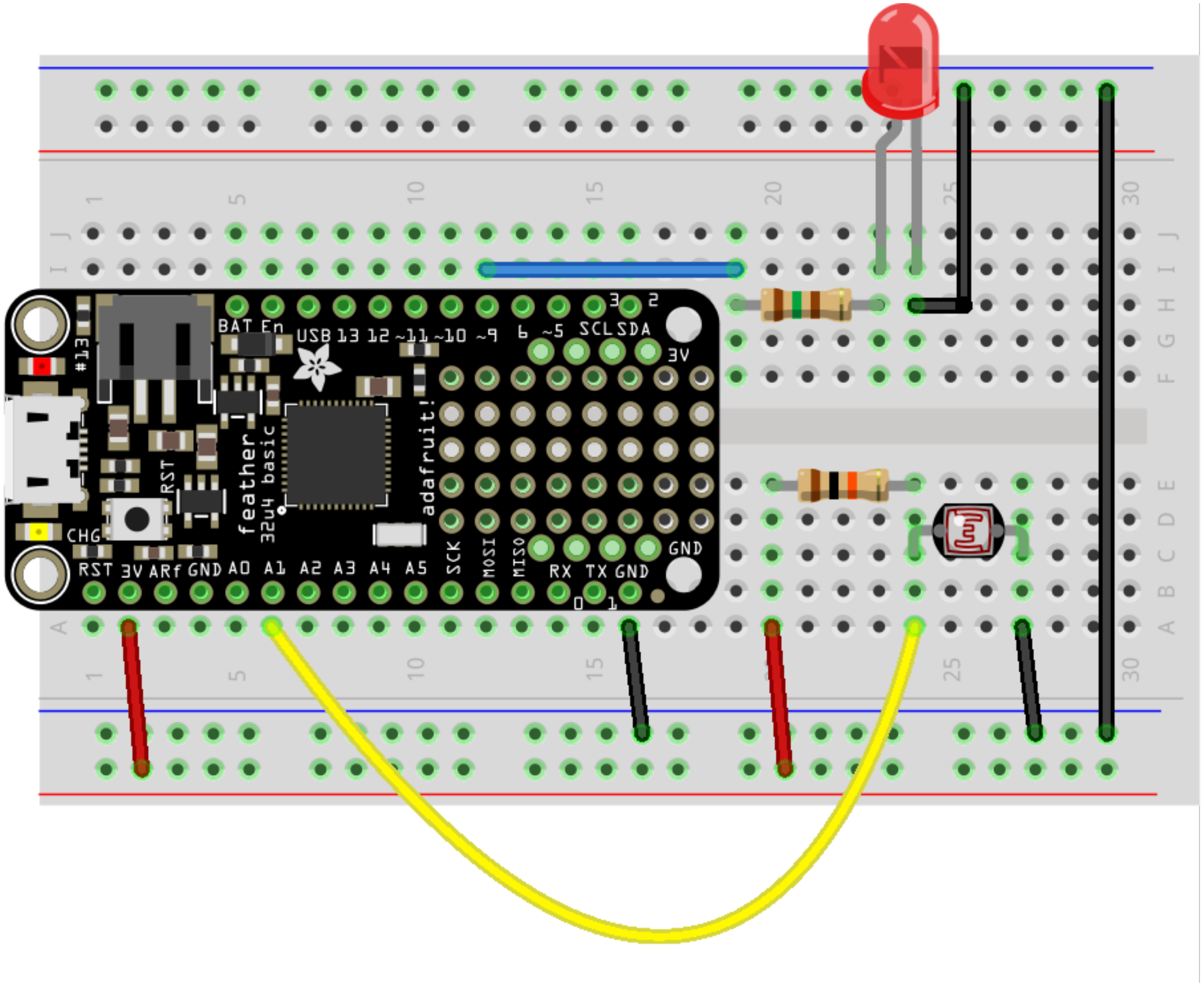
PHOTOCELL WITH VOLTAGE DIVIDER



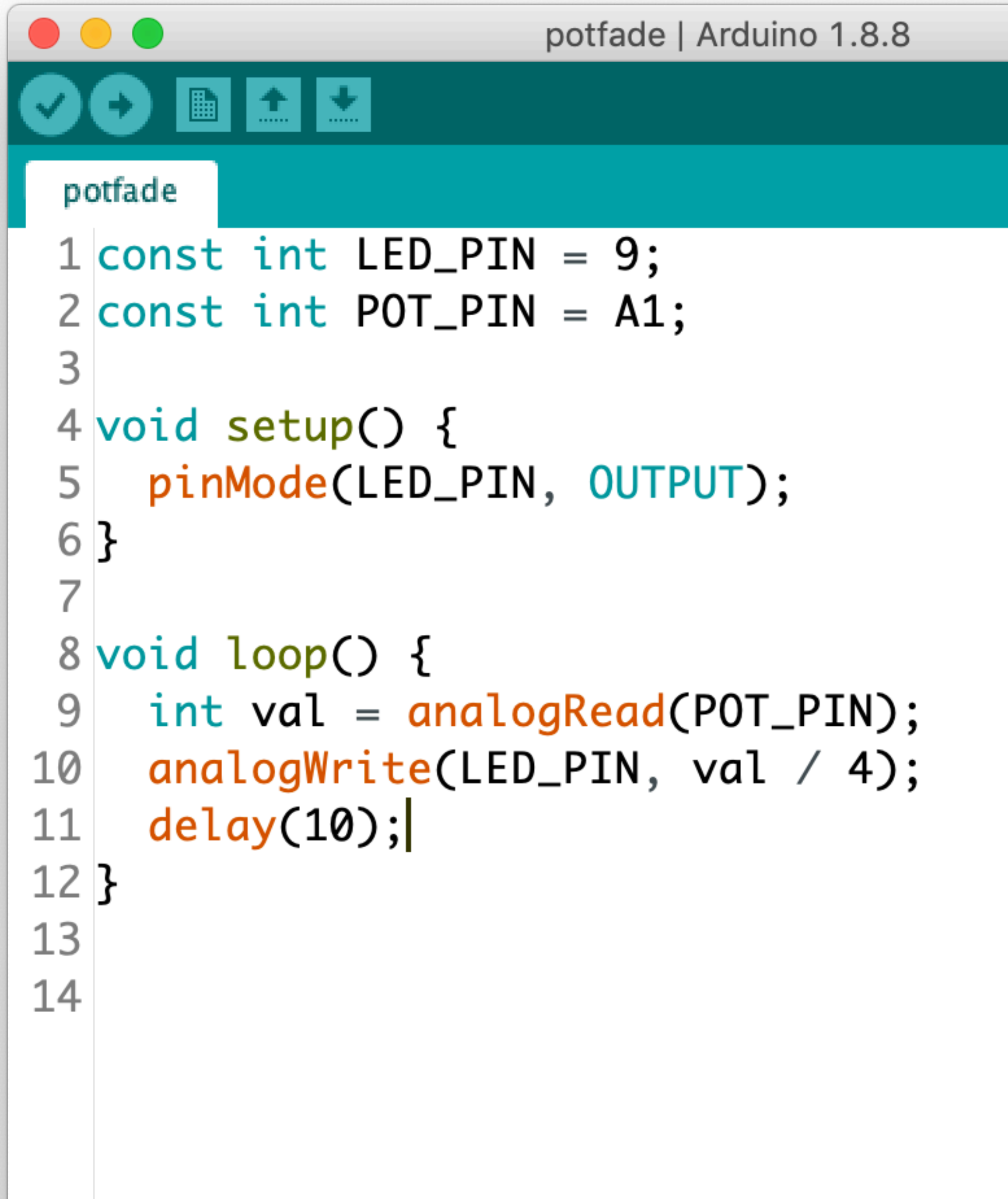


```
const int LED_PIN = 9;
const int POT_PIN = A1;

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

void loop() {
  int val = analogRead(POT_PIN);
  analogWrite(LED_PIN, val / 4);
  delay(10);
}
```

PHOTOCELL WITH VOLTAGE DIVIDER



BAT En USB 13 12 ~11 ~10 ~9 6 ~5 SCL SDA 3 2
3V
#13
feather 32u4 basic
adafruit!
CHG
RST
RST 3V ARf GND A0 A1 A2 A3 A4 A5 SCK MOSI MISO RX TX GND
0 1
GND
1 5 10 15 20 25 30
J I H G F E D C B A

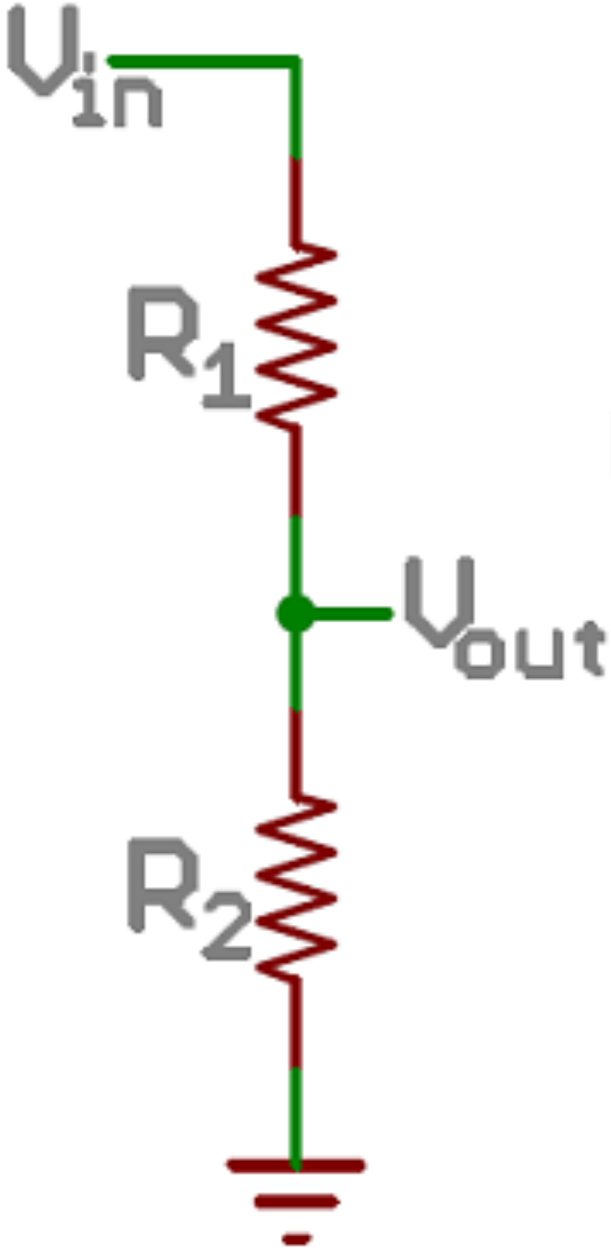


```
const int LED_PIN = 9;
const int POT_PIN = A1;

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

void loop() {
  int val = analogRead(POT_PIN);
  analogWrite(LED_PIN, val / 4);
  delay(10);
}
```

PHOTOCELL WITH VOLTAGE DIVIDER

Light Level	R_2 (Sensor)	R_1 (Fixed)	$R_2/(R_1+R_2)$	V_{out}	Value from analogRead
Bright	300Ω	10kΩ	0.029	0.096 V	~30
Dim	7kΩ	10kΩ	0.411	1.35 V	~420
Dark	40kΩ	10kΩ	0.800	2.64 V	~819

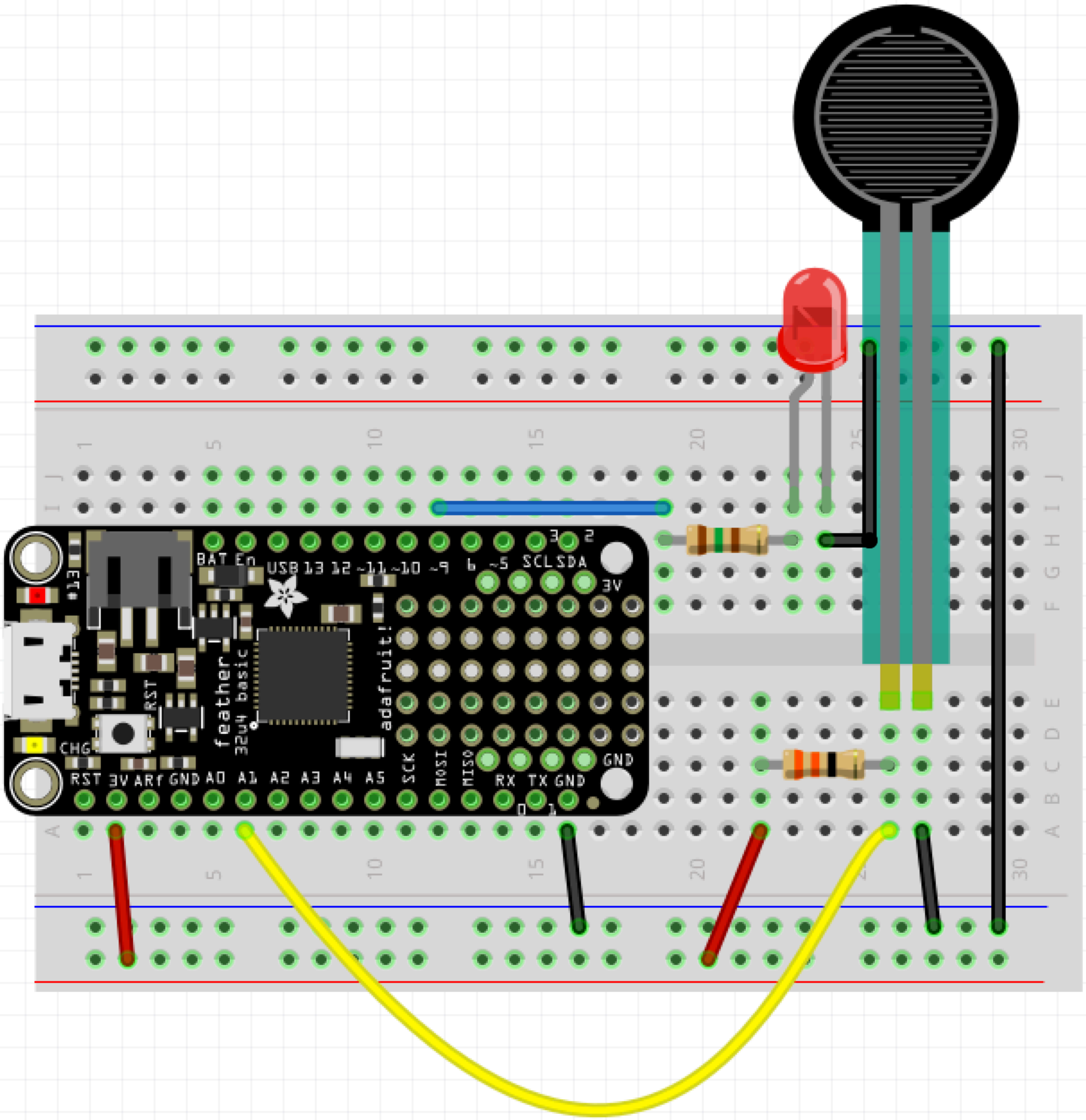
What the input pin gets



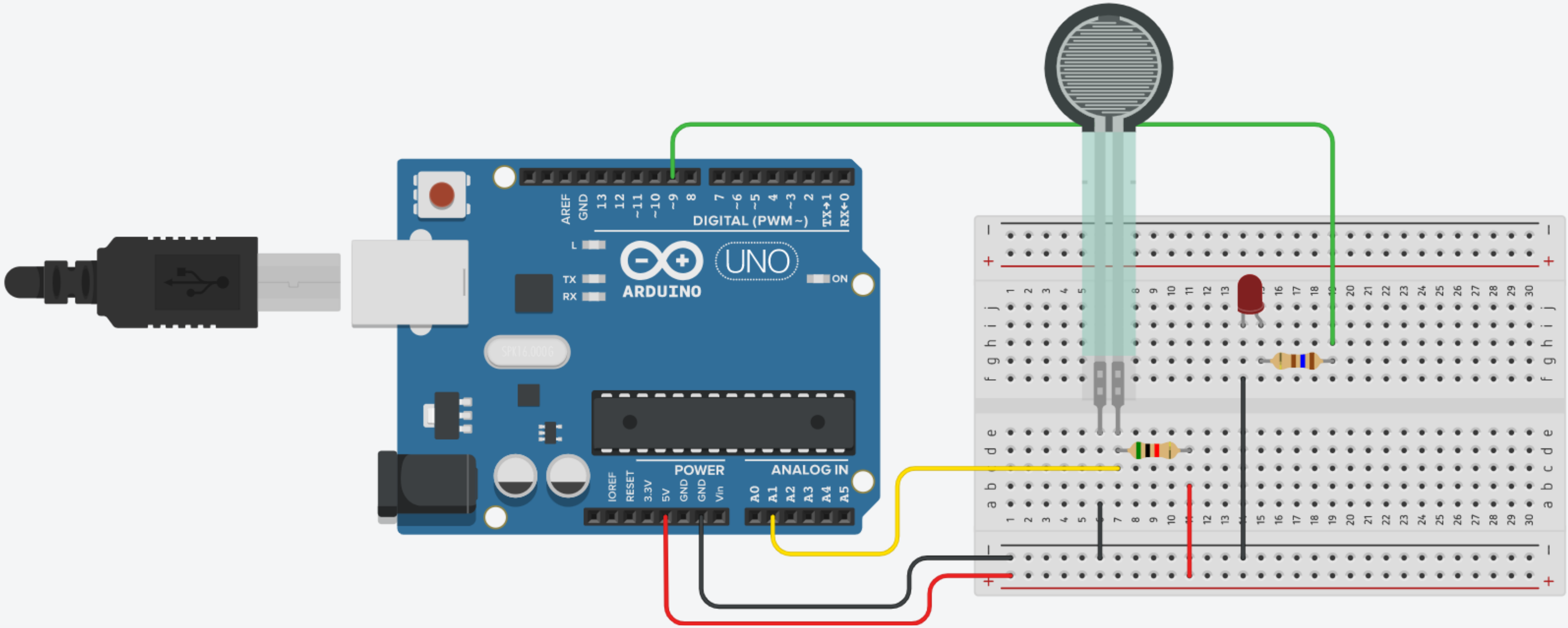
VOLTAGE DIVIDER

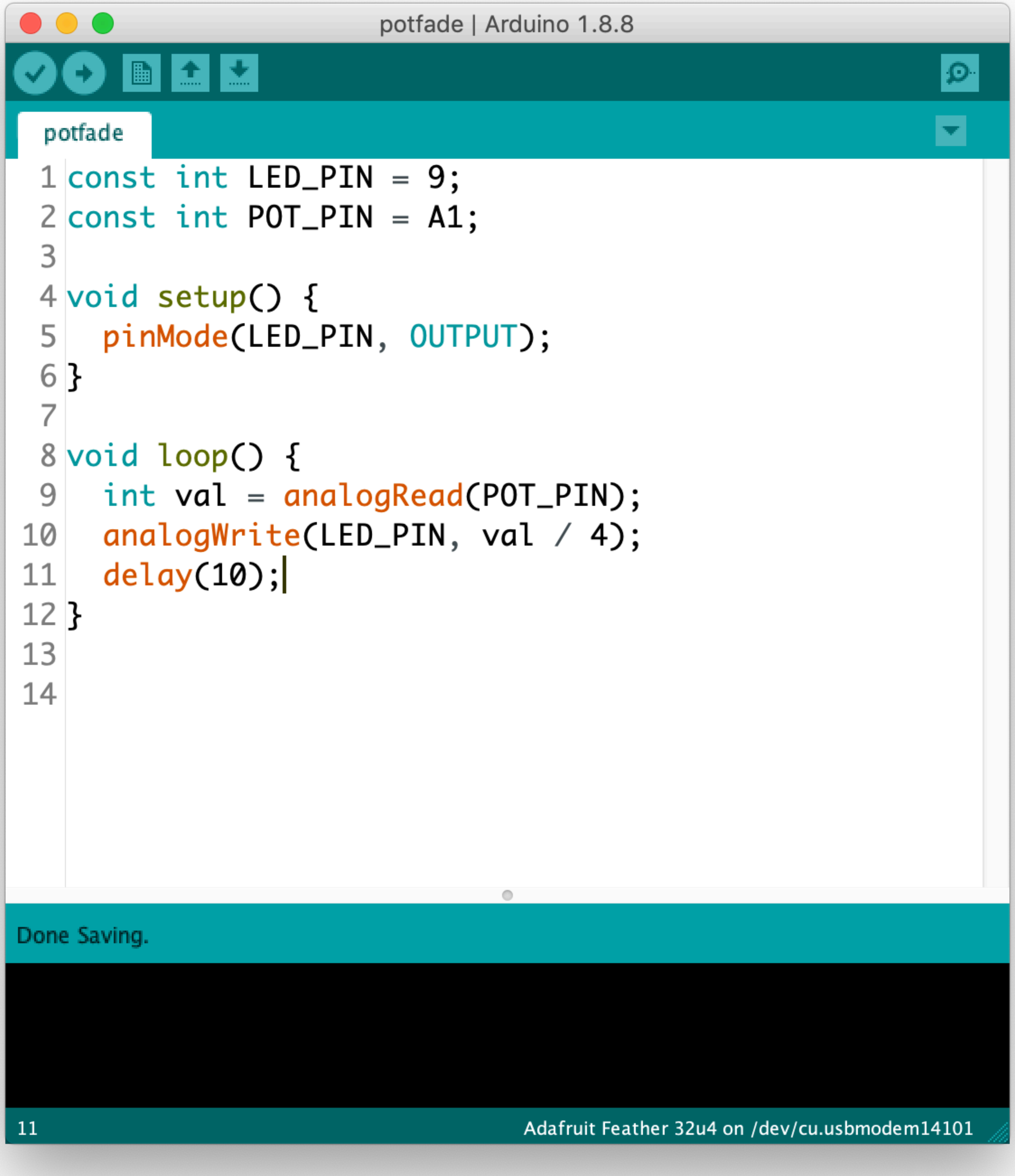
You can “tune” the output of the sensor by changing the value of **R_1**.

PRESSURE SENSOR WITH VOLTAGE DIVIDER


BAT En USB 13 12 ~11 ~10 ~9 6 ~5 SCL SDA 3 2
3V
GND
RST 3V ARf GND A0 A1 A2 A3 A4 A5 SCK MOSI MISO RX TX GND
0 1
#13
CHG
RST
feather
32u4 basic
adafruit!
1 5 10 15 20 25 30
J I H G F E D C B A

PRESSURE SENSOR WITH VOLTAGE DIVIDER





```
const int LED_PIN = 9;
const int POT_PIN = A1;

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

void loop() {
  int val = analogRead(POT_PIN);
  analogWrite(LED_PIN, val / 4);
  delay(10);
}
```

Done Saving.

11 Adafruit Feather 32u4 on /dev/cu.usbmodem14101

READ ANALOG VALUES AND FADE LED

analogRead() gives values in a range from 0-1024.

analogWrite() can only write values from 0-255.

To account for this we divide the value received from analogRead() by 4.