Embedded Systems

Traffic Light Prototype

Embedded Team, BFCAI

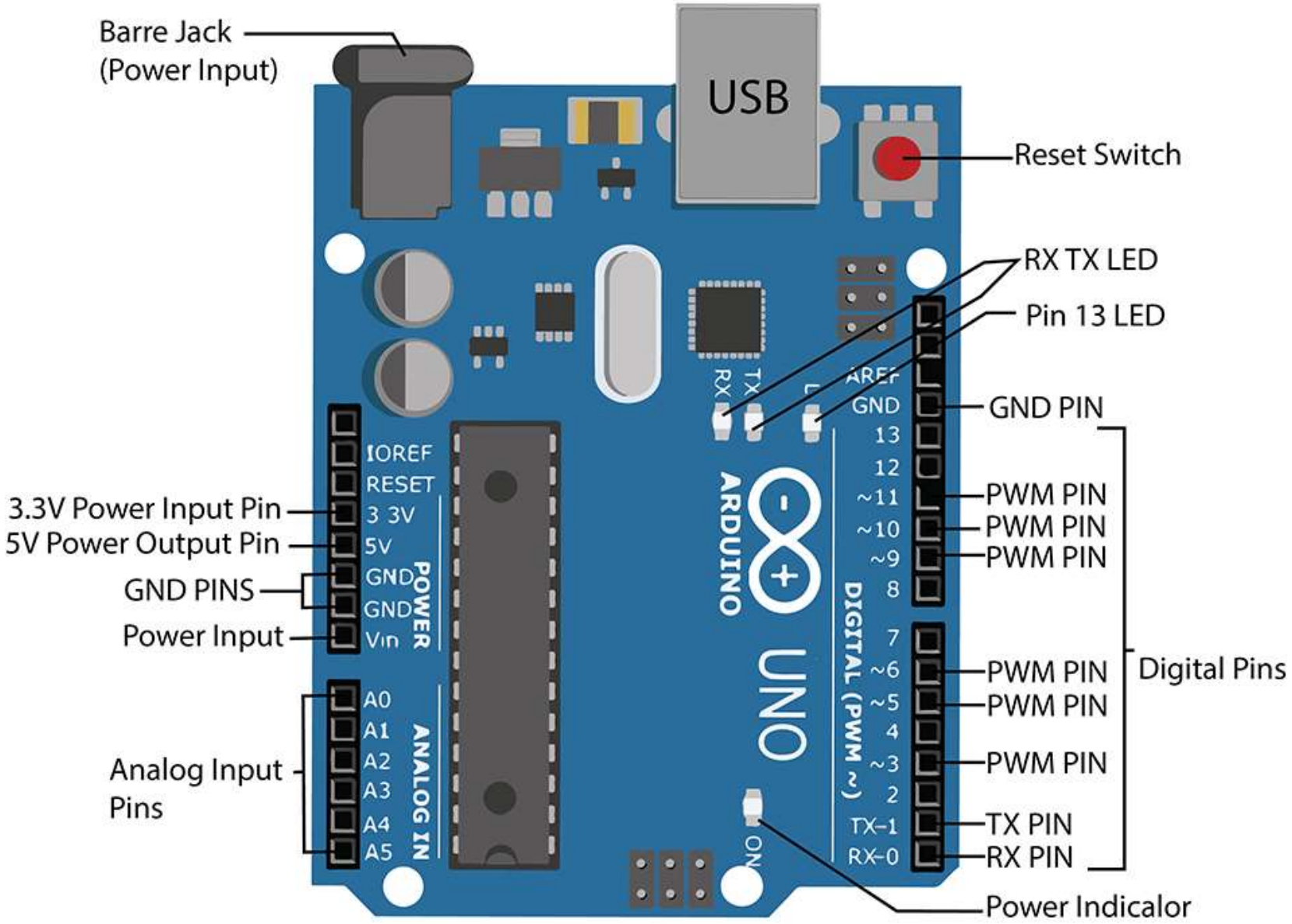
Arduino

- Arduino is open-source hardware that can be used to develop embedded systems with open-source software.
- Arduino has gained massive popularity among students for making a working model.
- The reasons behind the popularity of Arduino are its low cost, availability of software, and easy- to-interface possibility.
- The Arduino environment has been designed to be easy to use for beginners who have no software or electronics experience.

Arduino

- Arduino is used in many educational programs around the world, particularly by designers who want to easily create prototypes but do not need a deep understanding of the technical details.
- Because it is designed to be used by nontechnical people, the software includes plenty of example code to demonstrate how to use the Arduino board.
- People already working with microcontrollers are also attracted to Arduino because of its facility for quick implementation of ideas.

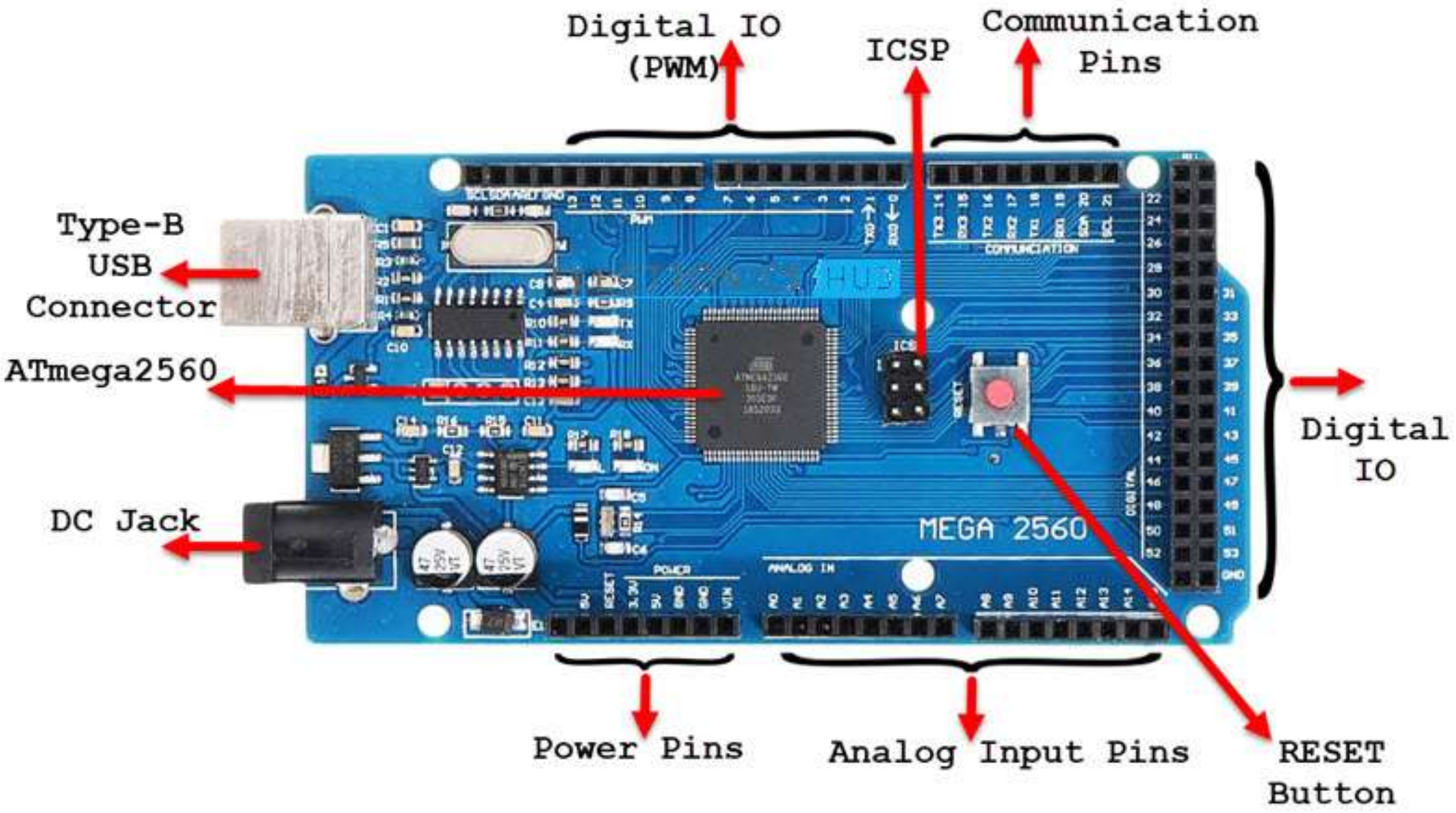
Arduino Uno Board



Arduino Uno: Tech Specs

Microcontroller	ATmega328P
Clock Speed	16 MHz
Flash Memory	32 KB
SRAM	2 KB
EEPROM	1 KB
Digital I/O Pins	14 (of which 6 PWM)
PWM Digital I/O Pins	6
Analog Input Pins	6
Operating Voltage	5V
DC Current per I/O Pin	20 mA

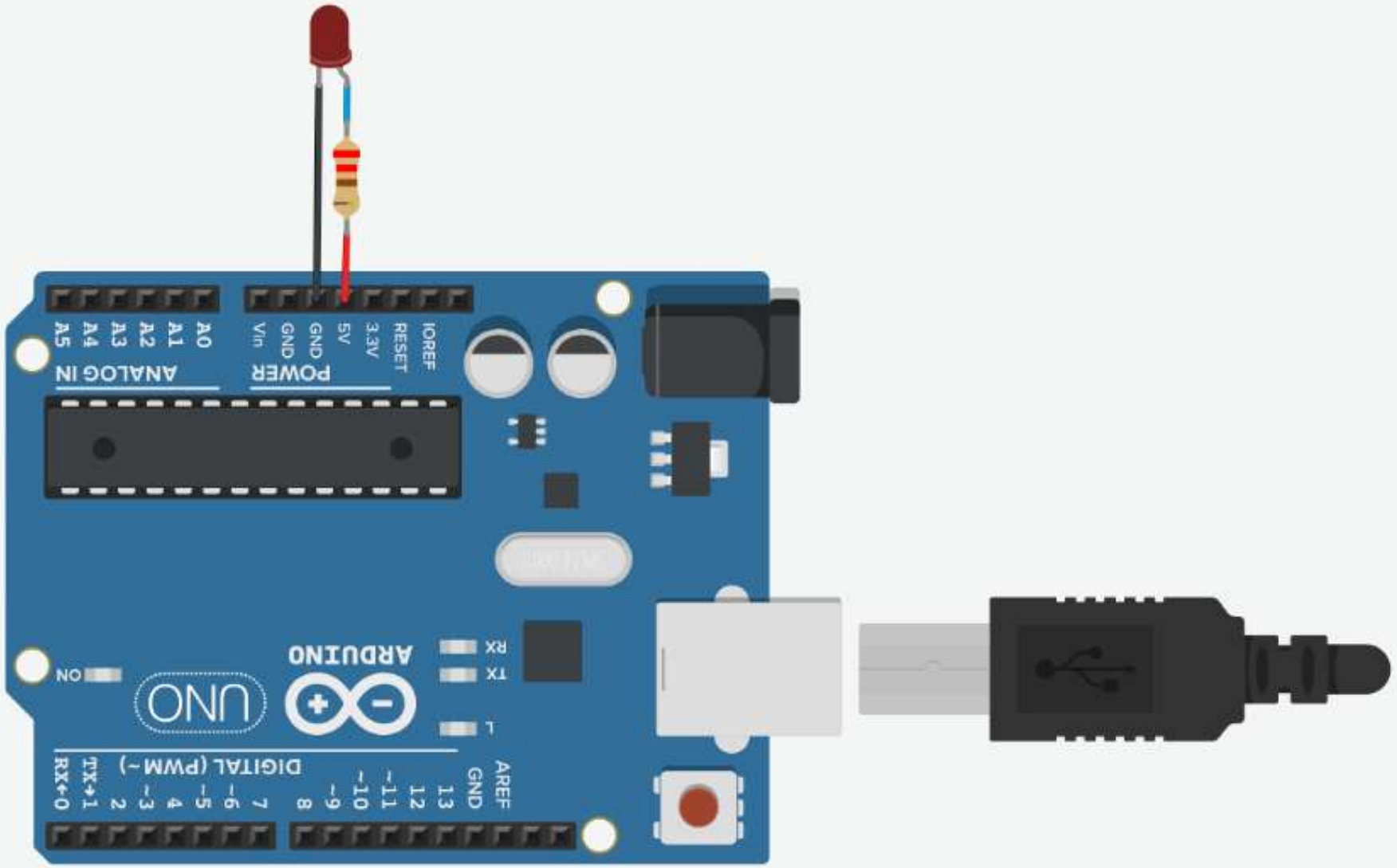
Arduino Mega Board



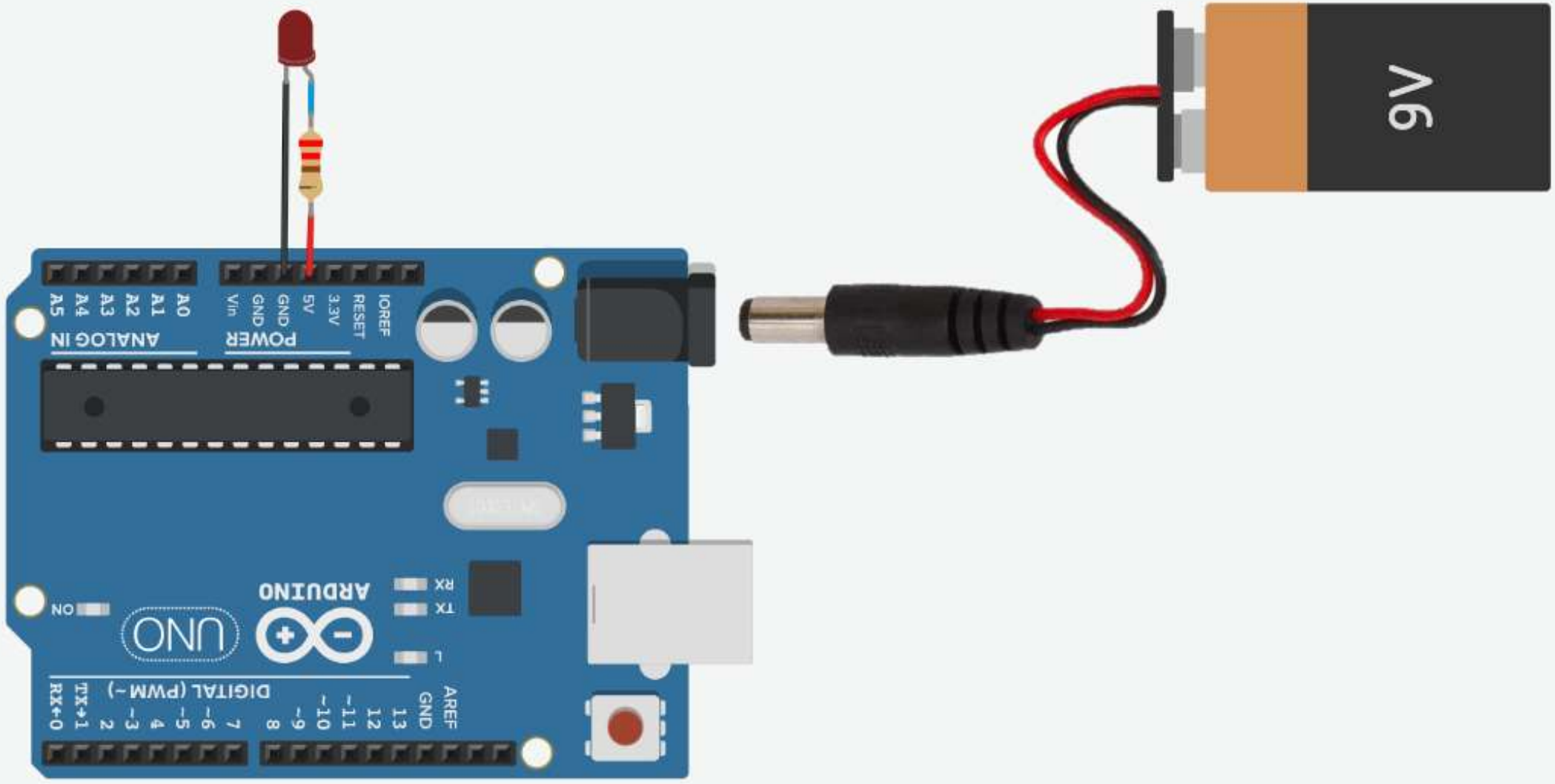
Arduino Mega: Tech Specs

Microcontroller	ATmega2560
Clock Speed	16 MHz
Flash Memory	256 KB
SRAM	8 KB
EEPROM	4 KB
Digital I/O Pins	54 (of which 15 PWM)
PWM Digital I/O Pins	15
Analog Input Pins	16
Operating Voltage	5V
DC Current per I/O Pin	20 mA

Connecting Arduino to Power

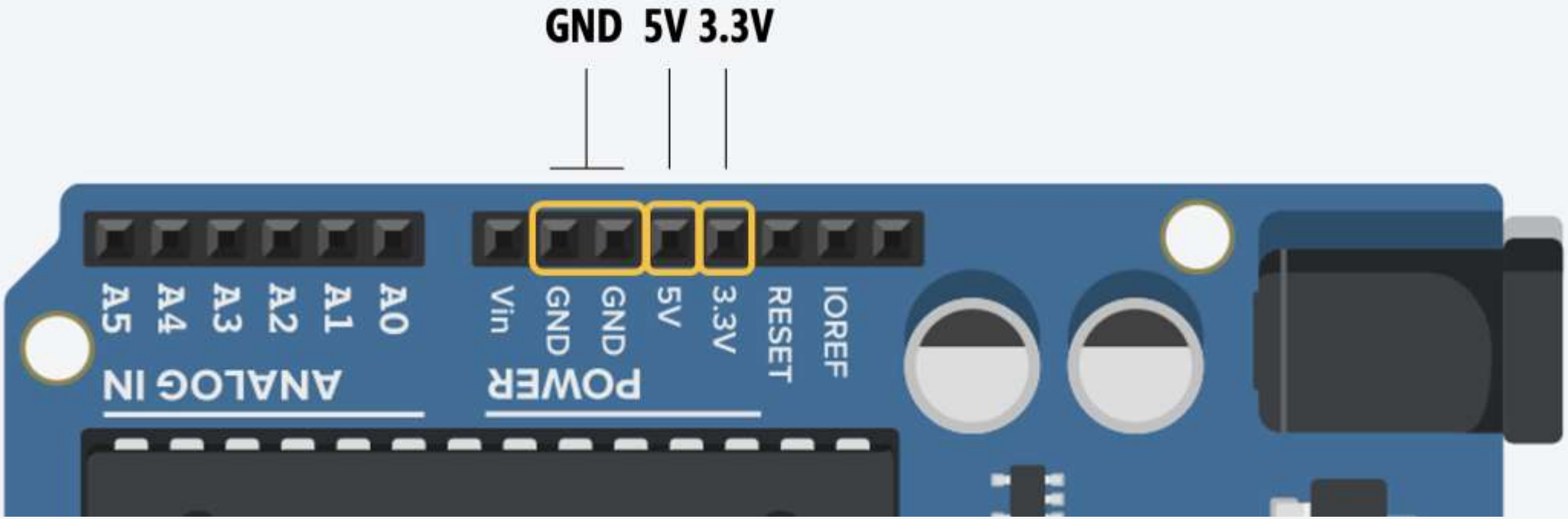


Connecting Arduino to Power



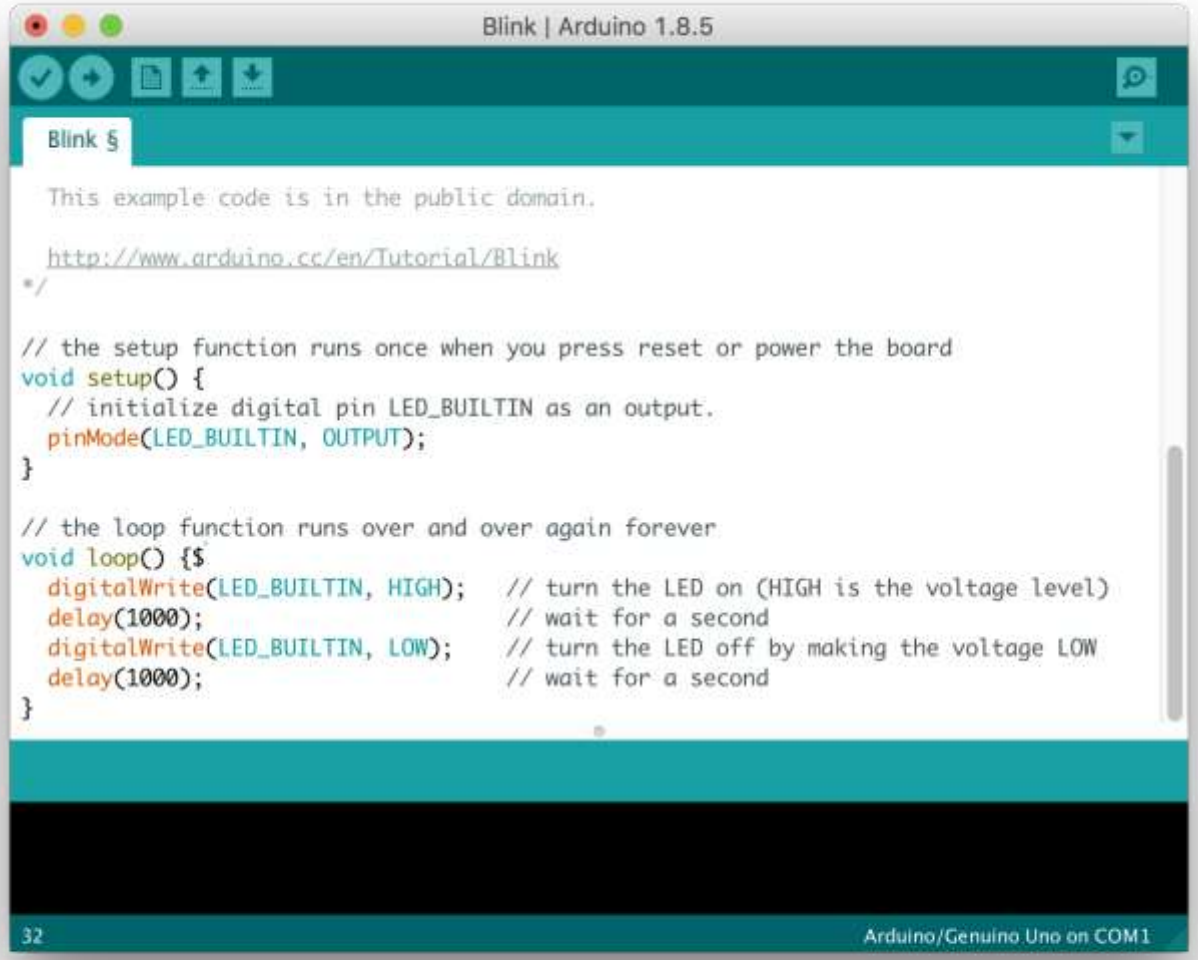
Power Supply Pins

- The Arduino Uno provides both a 5V, and a 3.3V power supply.



Arduino IDE

- The Arduino IDE enables you to write and edit code and convert this code into instructions that Arduino hardware understands.



Arduino Sketches

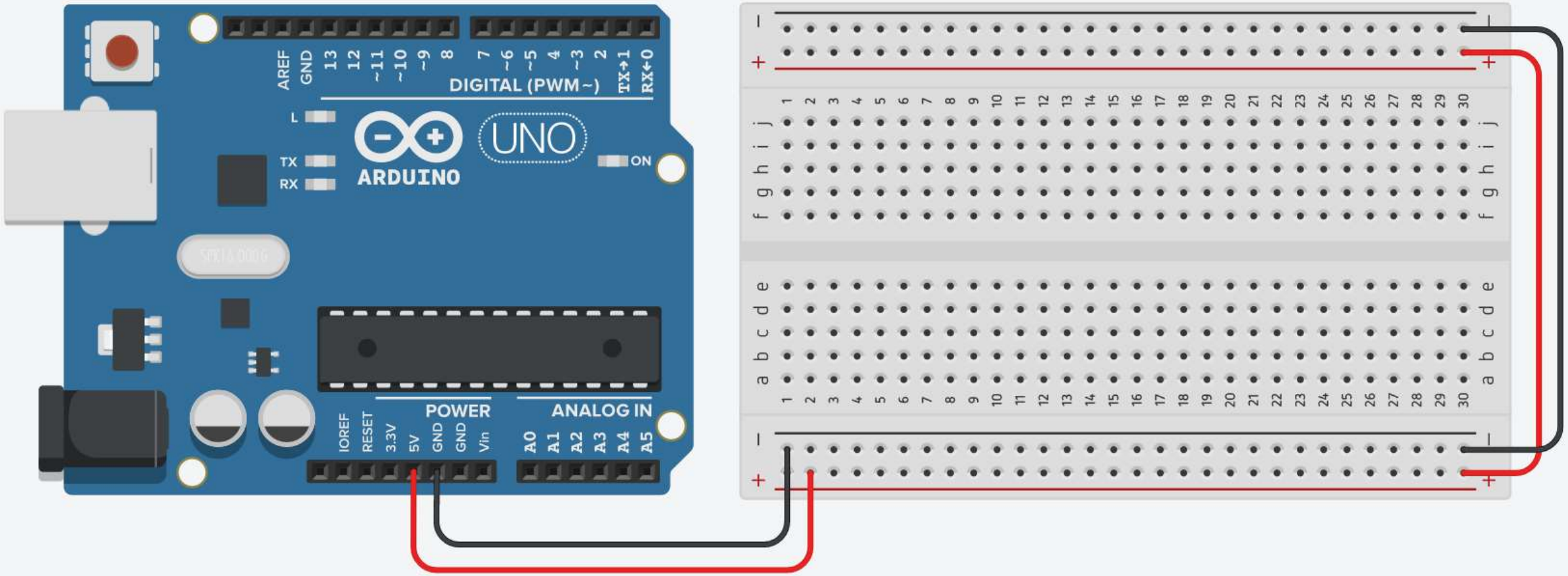
- A sketch is the name that Arduino uses for a program.

```
void setup() {
  // put your setup code here, to run once:

}

void loop() {
  // put your main code here, to run repeatedly:

}
```

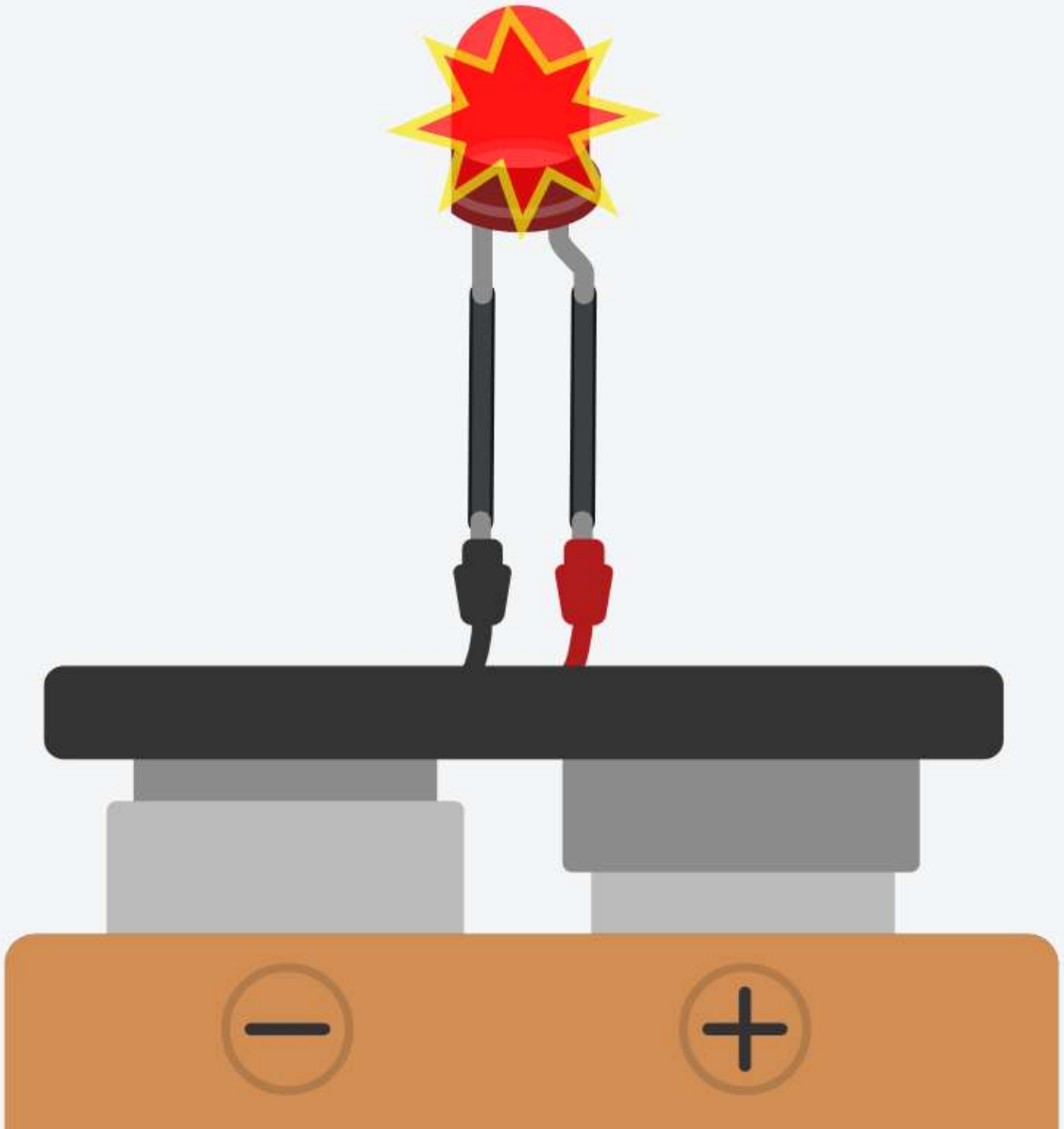
Arduino Sketches

- There are two special functions that are a part of every Arduino sketch: `setup()` and `loop()`.
- The `setup()` is called once, when the sketch starts.
- It's a good place to do setup tasks like setting pin modes.
- The `loop()` function is called over and over and is heart of most sketches.
- You need to include both functions in your sketch, even if you don't need them for anything.

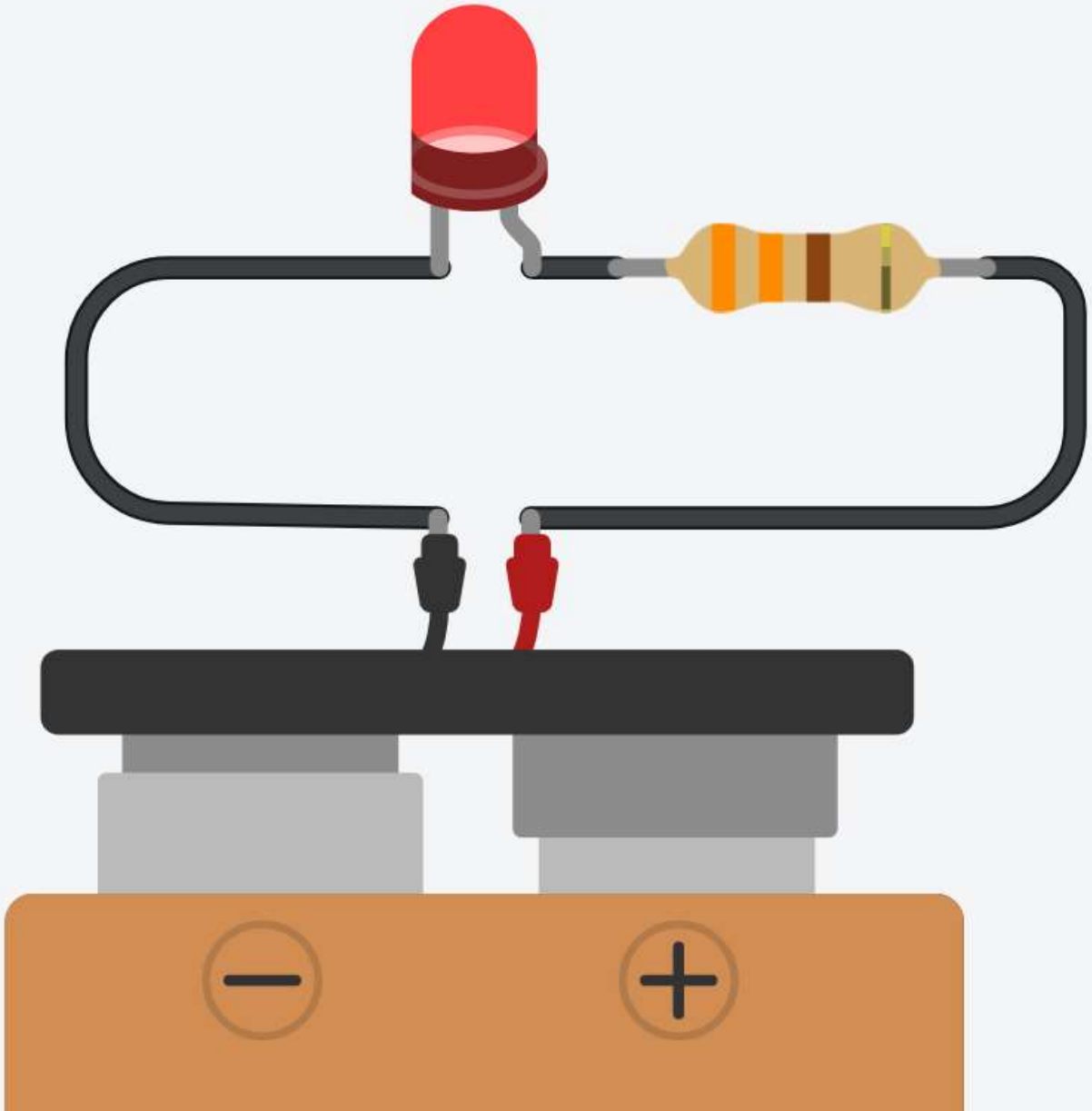
Breadboard


AREF
GND
13
12
~11
~10
~9
8
7
~6
~5
4
~3
2
TX→1
RX←0
DIGITAL (PWM~)
L
TX
RX
ARDUINO
UNO
ON
POWER
ANALOG IN
IOREF
RESET
3.3V
5V
GND
GND
Vin
A0
A1
A2
A3
A4
A5

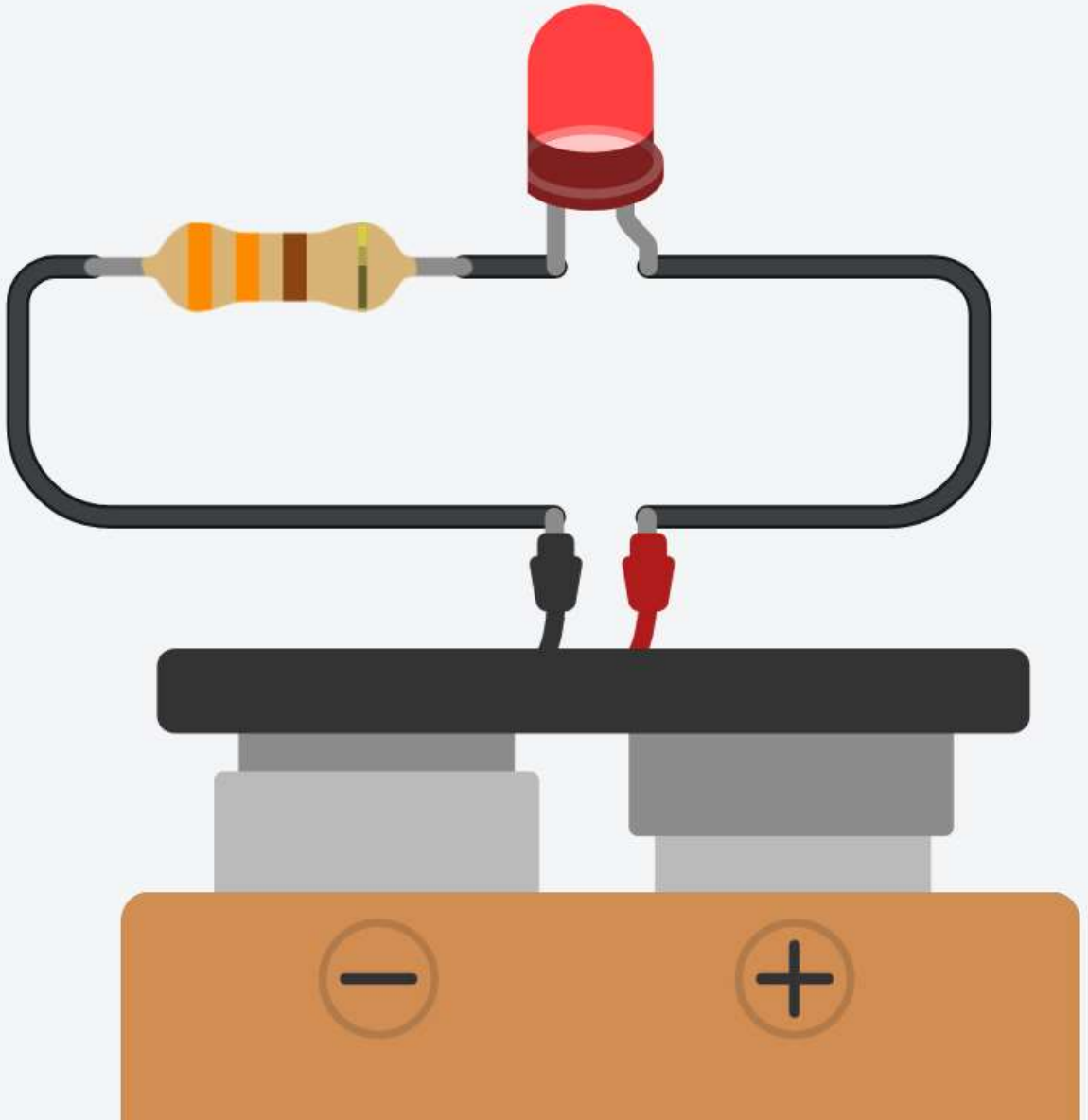
Turning on an LED



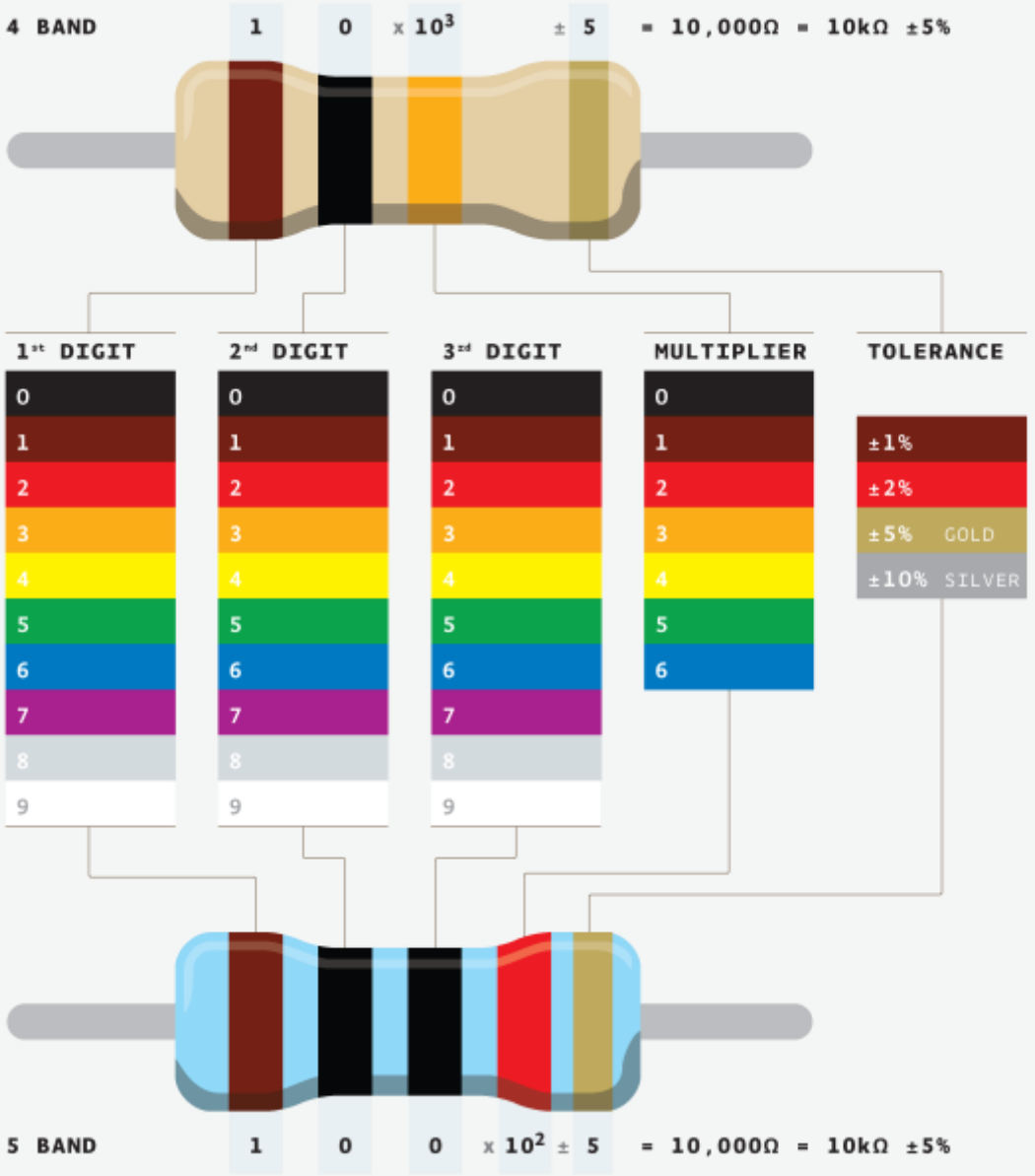
Turning on an LED



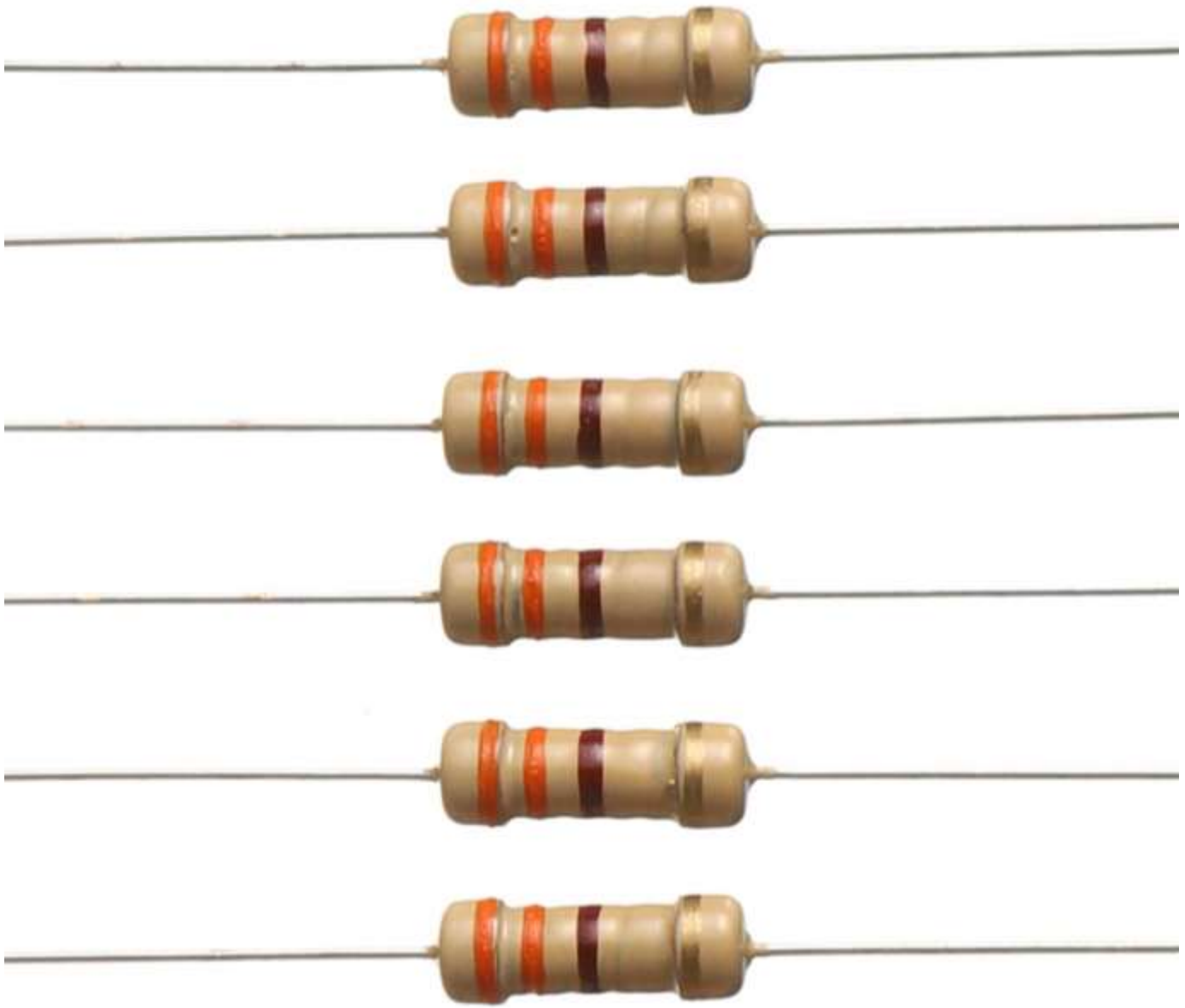
Turning on an LED



Turning on an LED

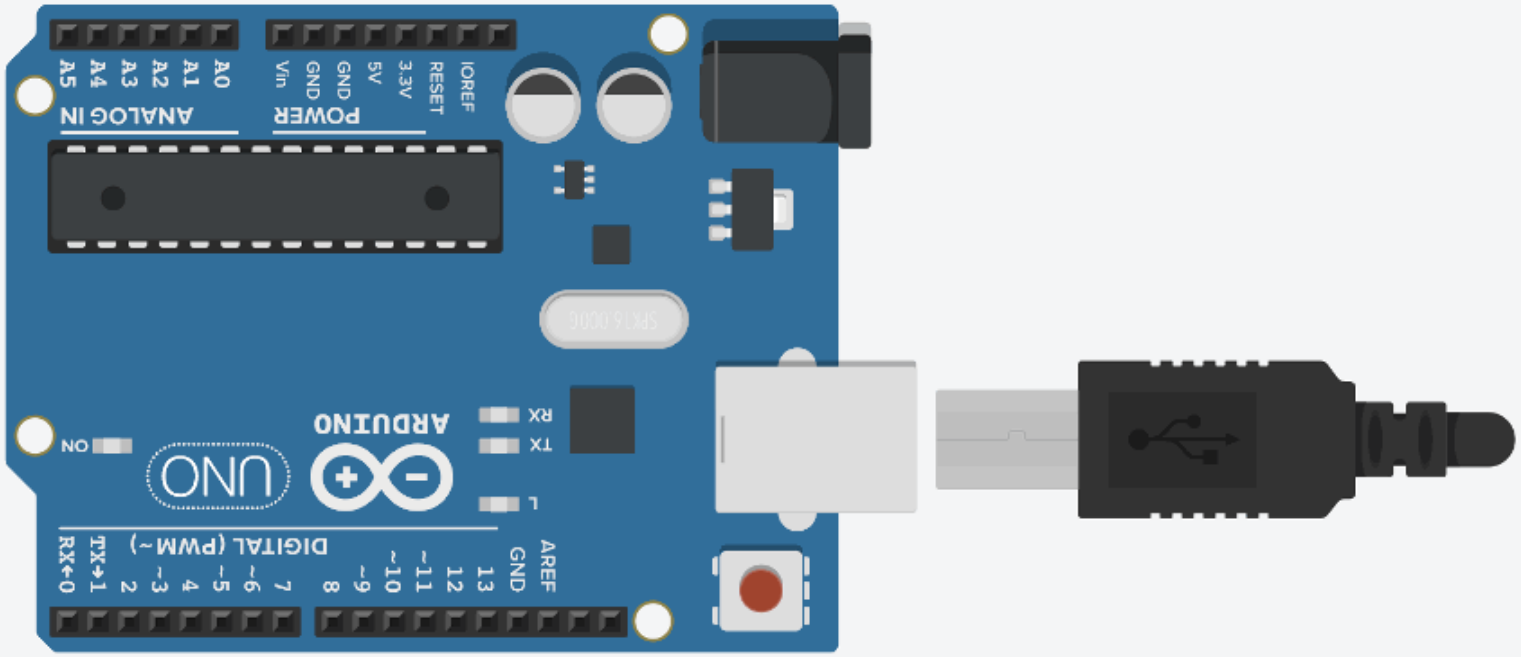


Turning on an LED

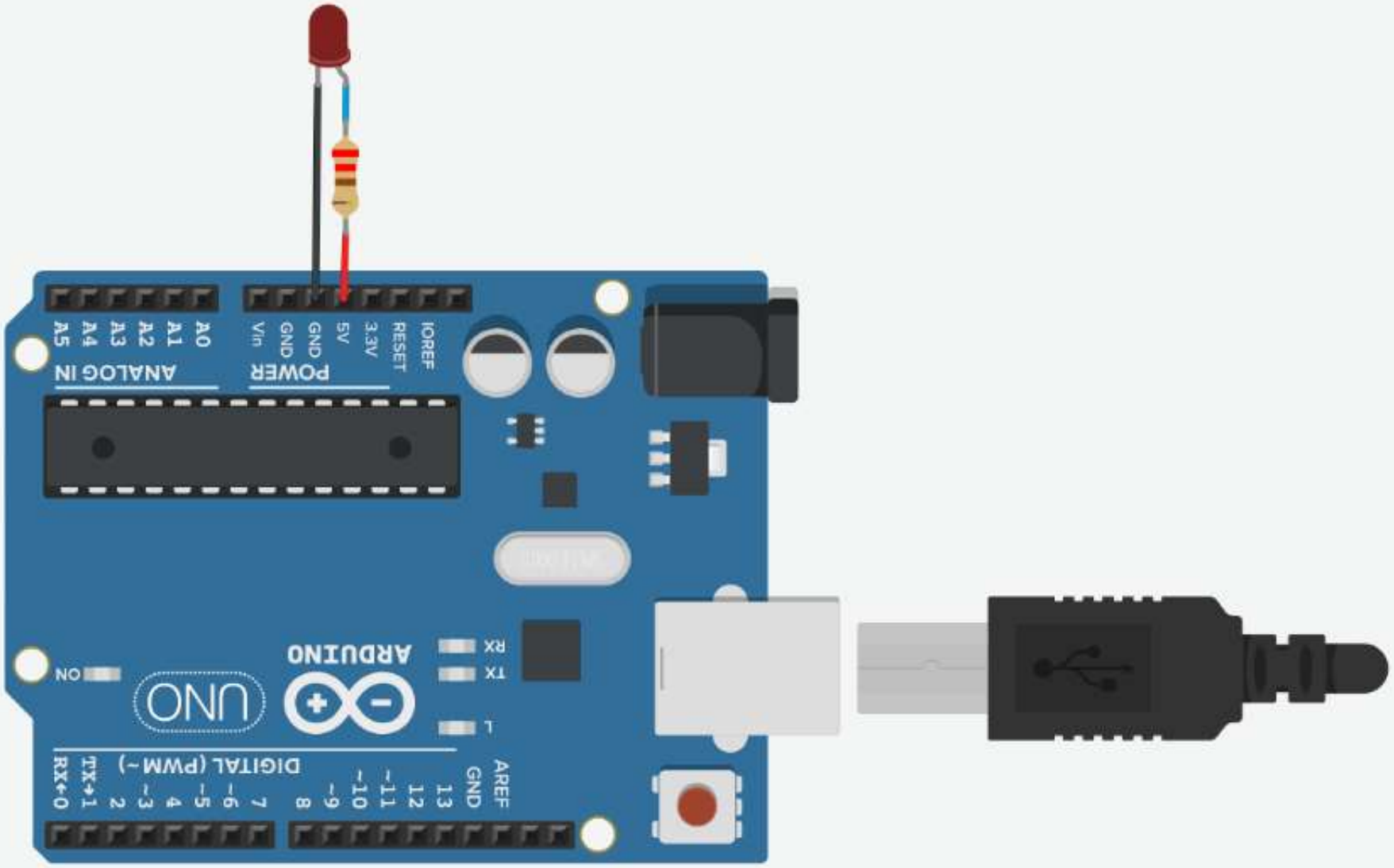


Turning on an LED

LED Cathode Towards GND
The LED cathode (short leg or look for flat side of LED casing) connects to GND

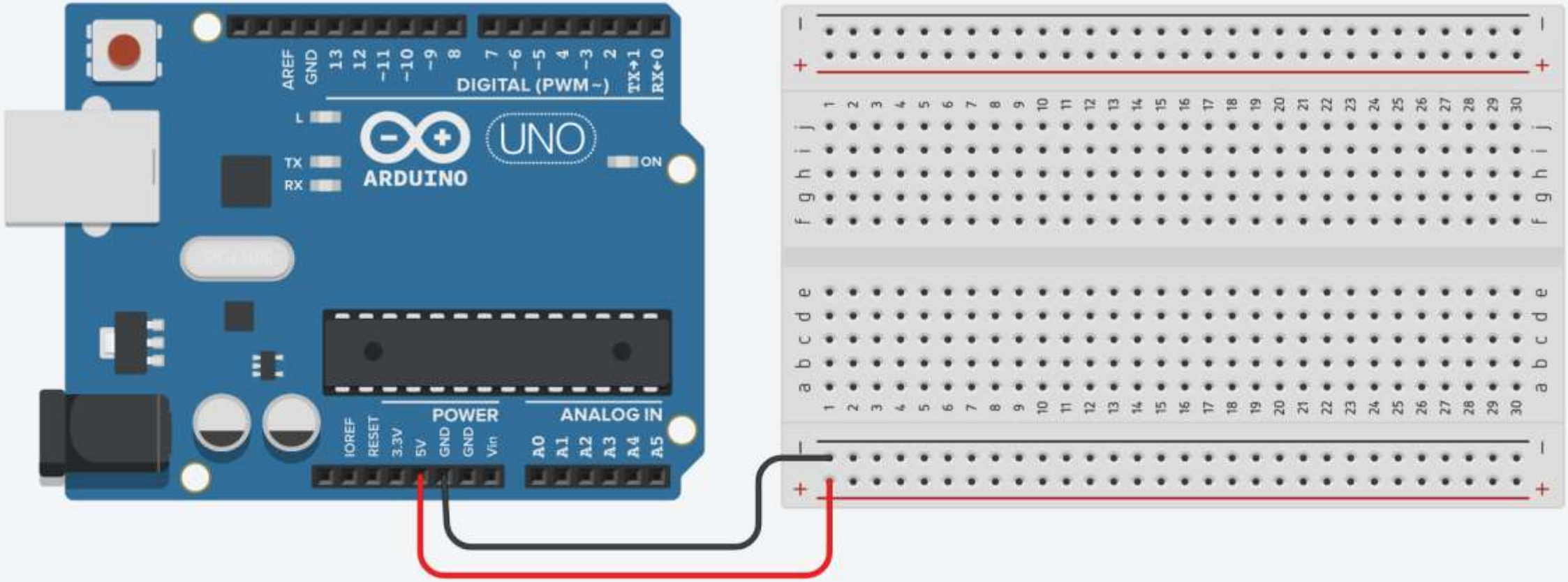


Turning on an LED



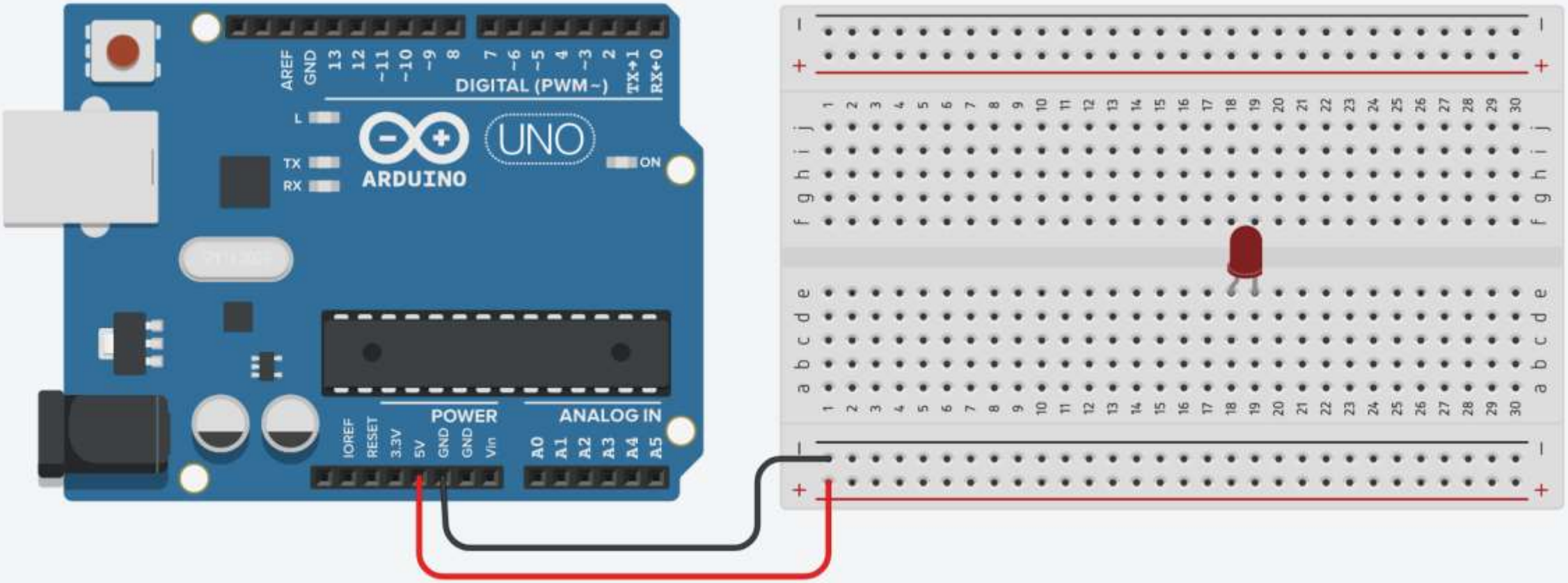
Turning on an LED: Steps

1. Connect breadboard power (+) and ground (-) rails to Arduino 5V and ground (GND), respectively.



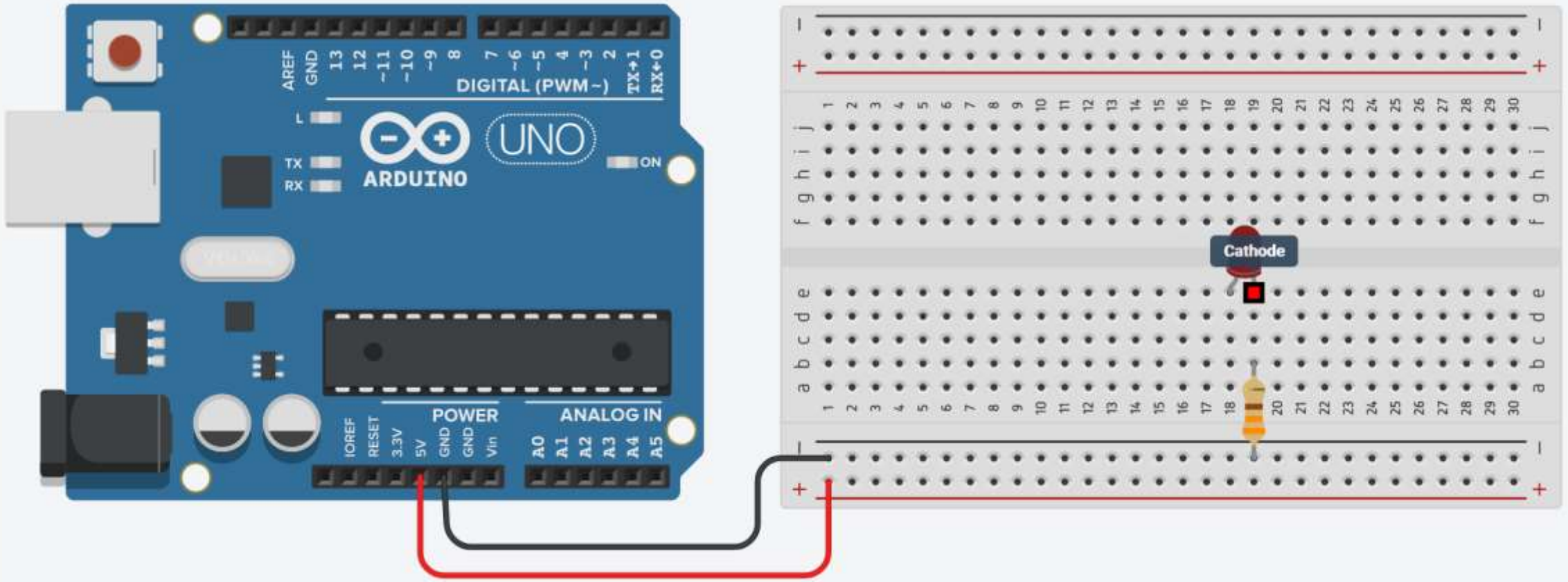
Turning on an LED: Steps

2. Plug the LED into two different breadboard rows.



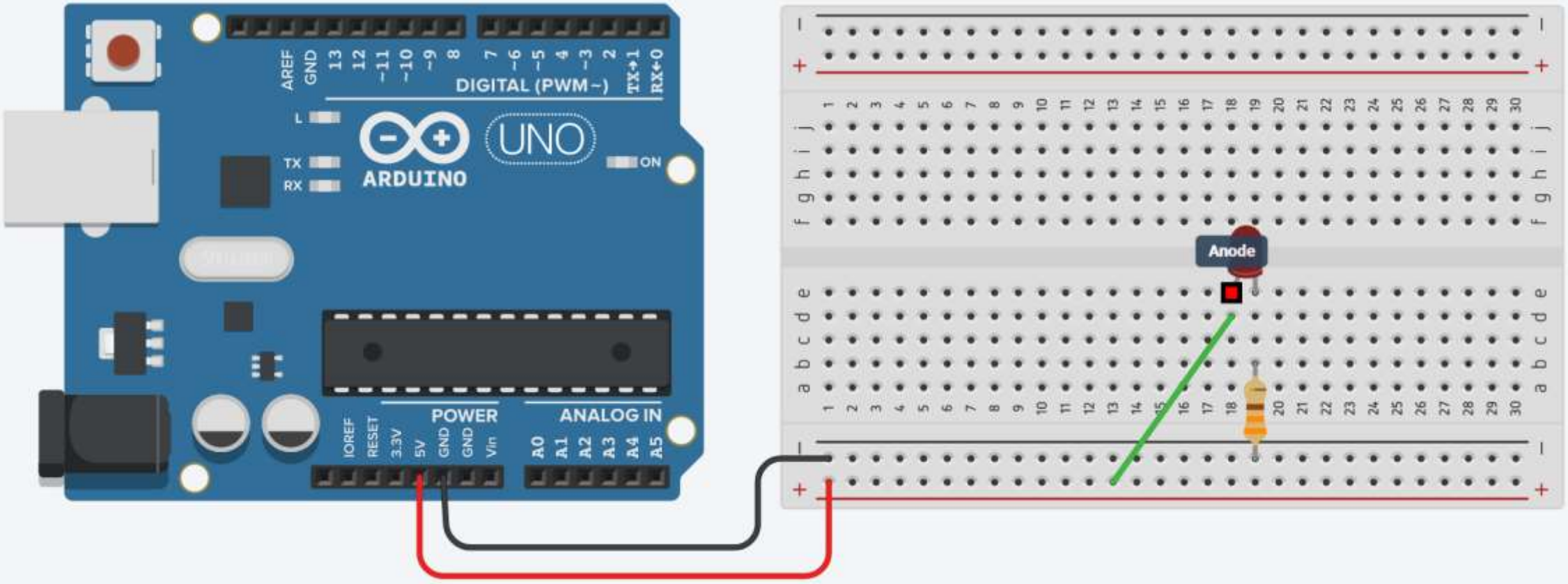
Turning on an LED: Steps

3. The cathode (shorter leg) connects to one leg of a resistor of 330Ω, and the other resistor leg to the ground.



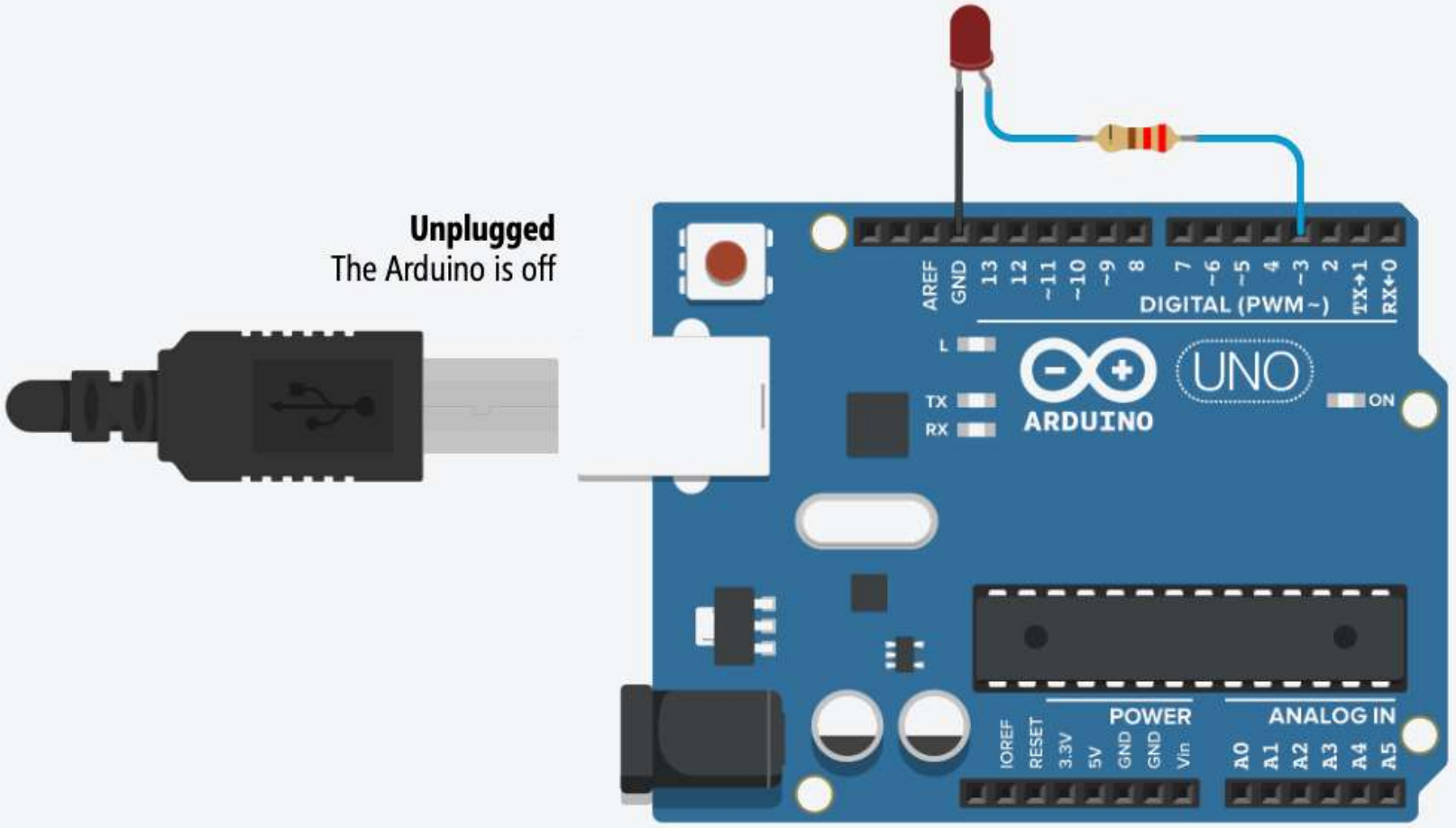
Turning on an LED: Steps

4. Wire up the LED anode (longer leg) to the power.



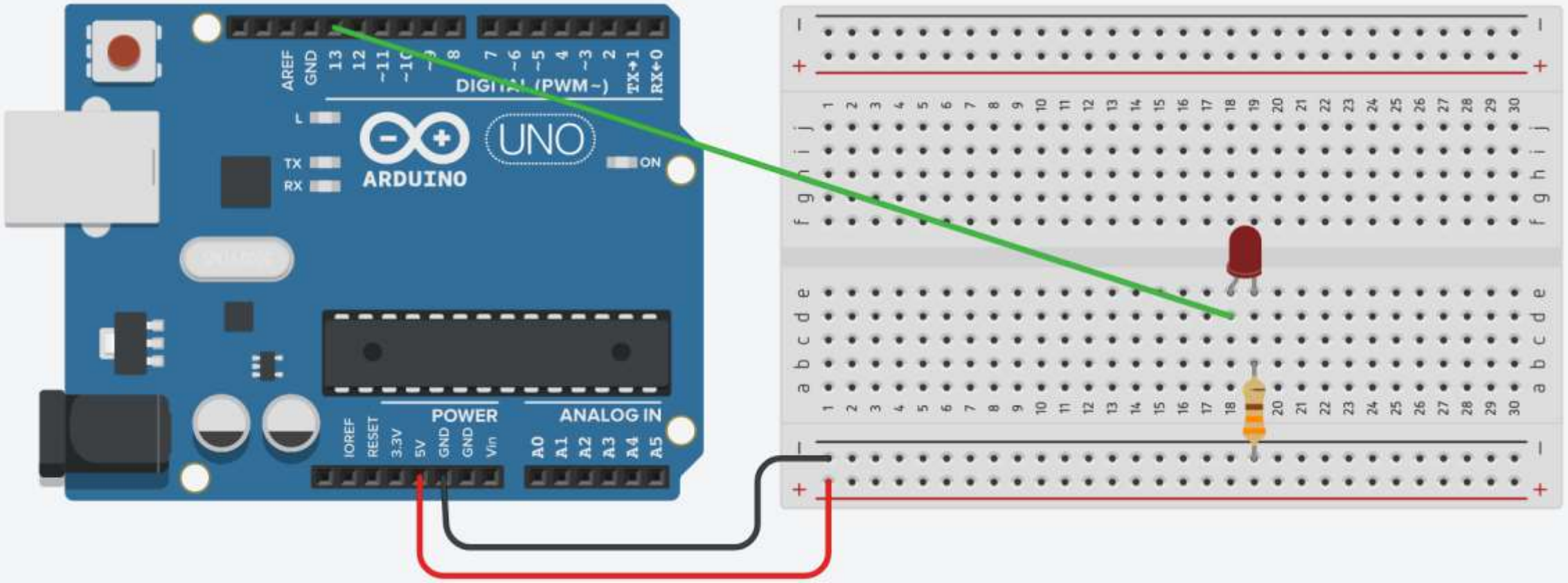
Your First Arduino Project: Blinking an LED

- Turn an LED on and off every second.



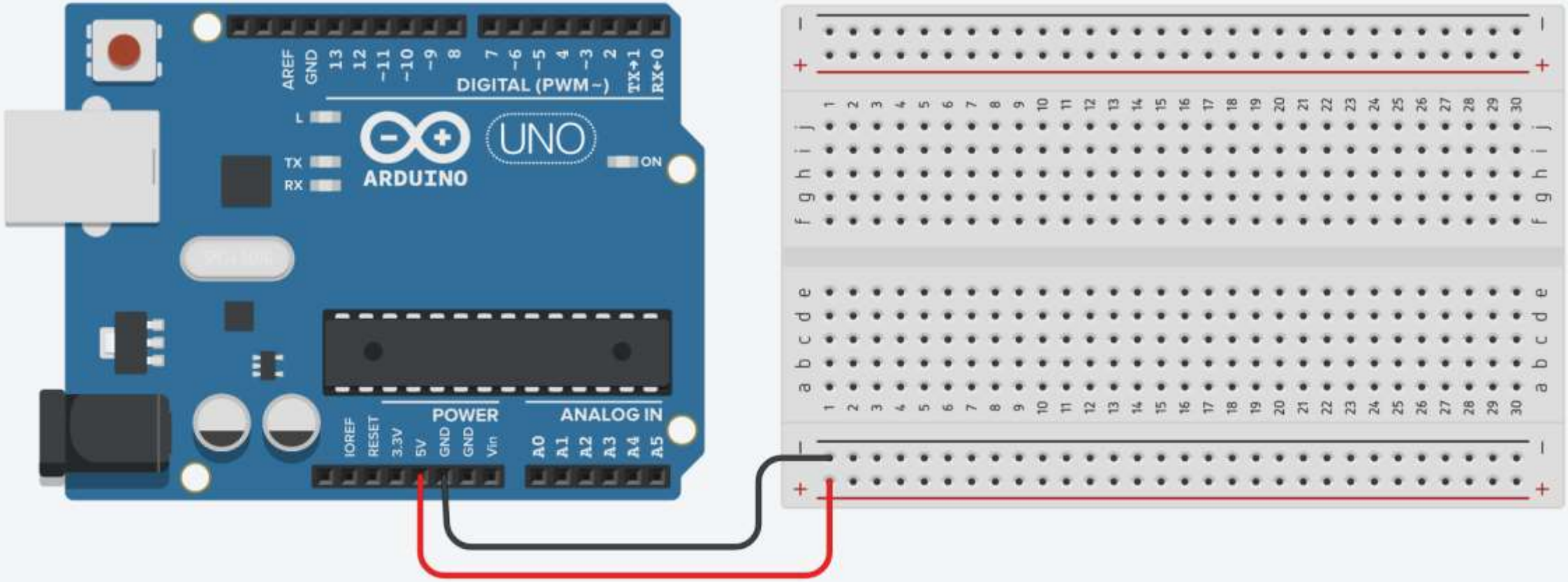
Your First Arduino Project: Circuit

- Turn an LED on and off every second.



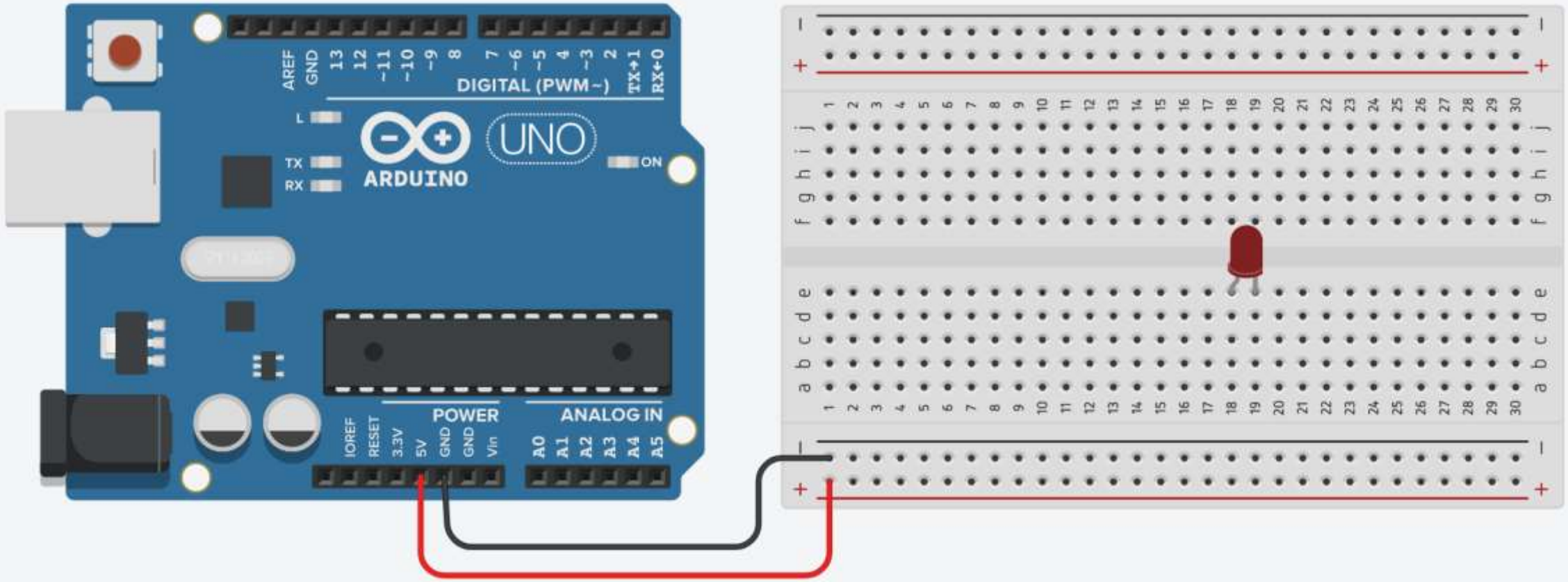
Your First Arduino Project: Steps

1. Connect breadboard power (+) and ground (-) rails to Arduino 5V and ground (GND), respectively.



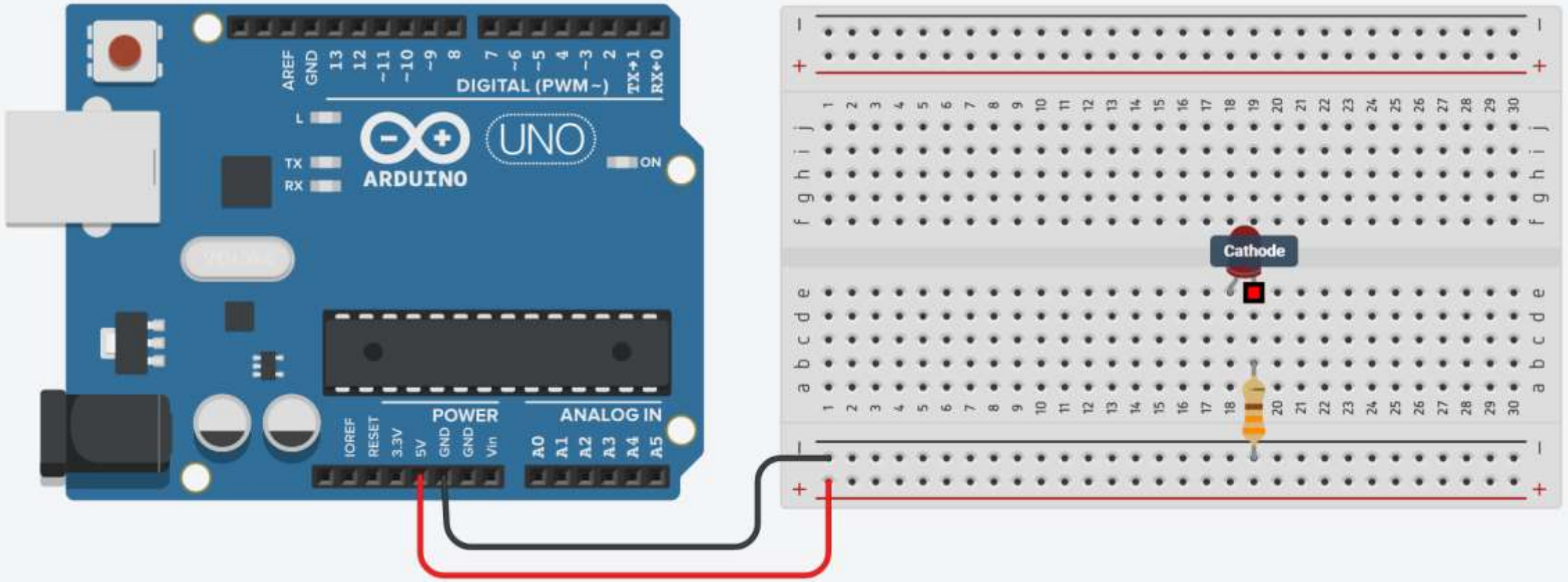
Your First Arduino Project: Steps

2. Plug the LED into two different breadboard rows.



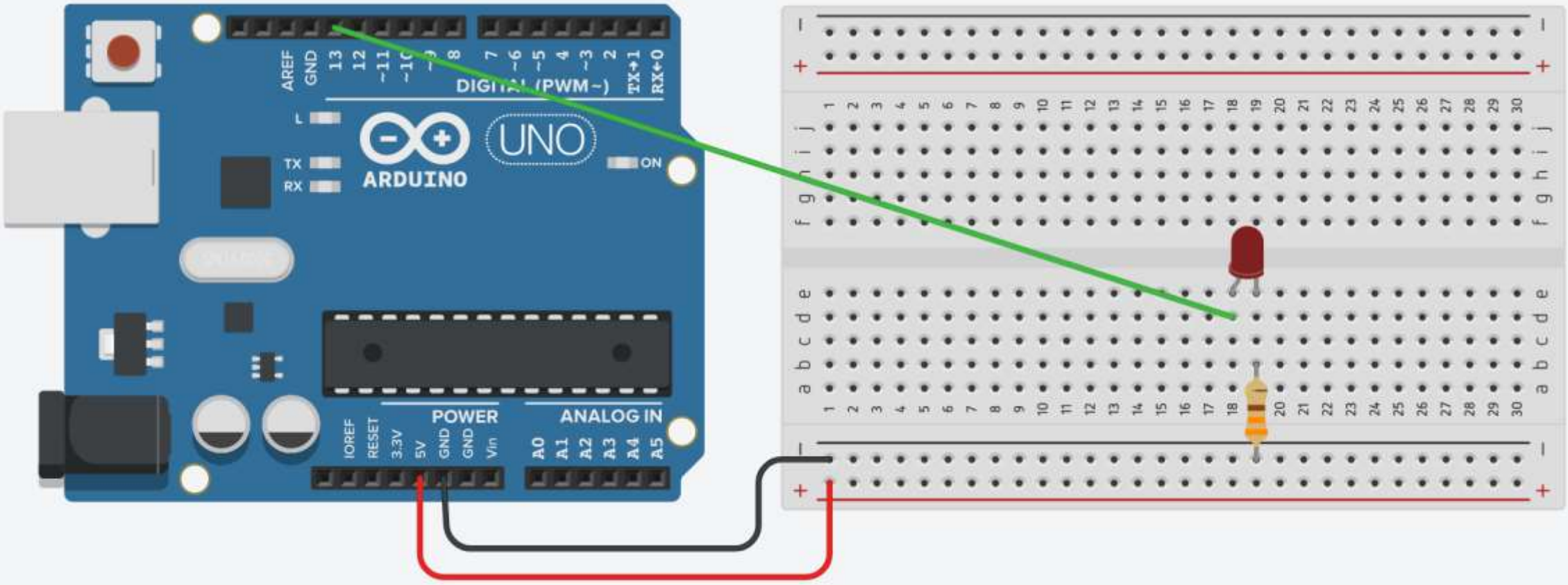
Your First Arduino Project: Steps

3. The cathode (shorter leg) connects to one leg of a resistor of 330Ω, and the other resistor leg to the ground.

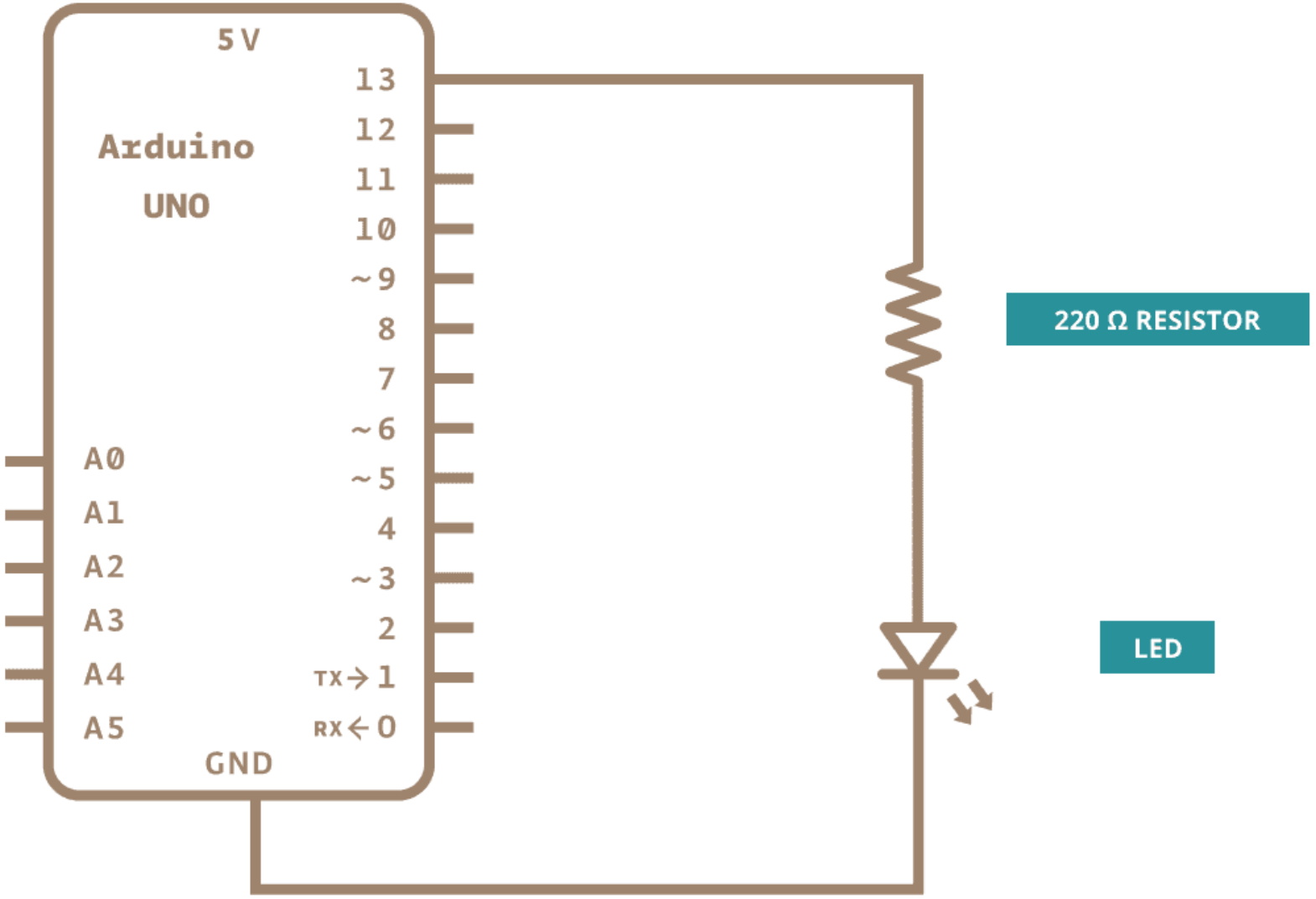


Your First Arduino Project: Steps

4. Wire up the LED anode (longer leg) to Arduino pin 13.

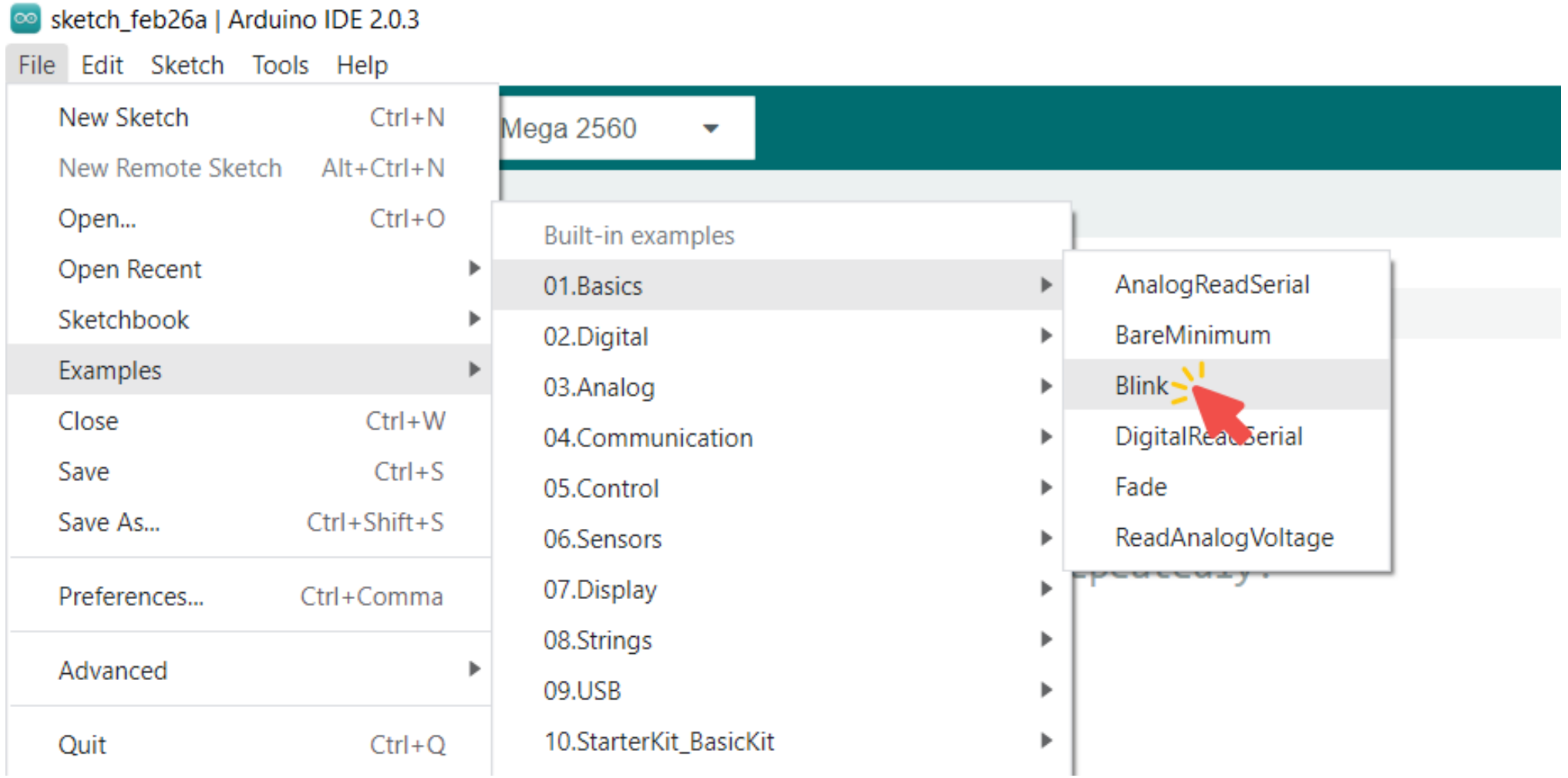


Your First Arduino Project: Schematic



Your First Arduino Project: Blink

You may also load it from File → Examples → 01.Basics → Blink

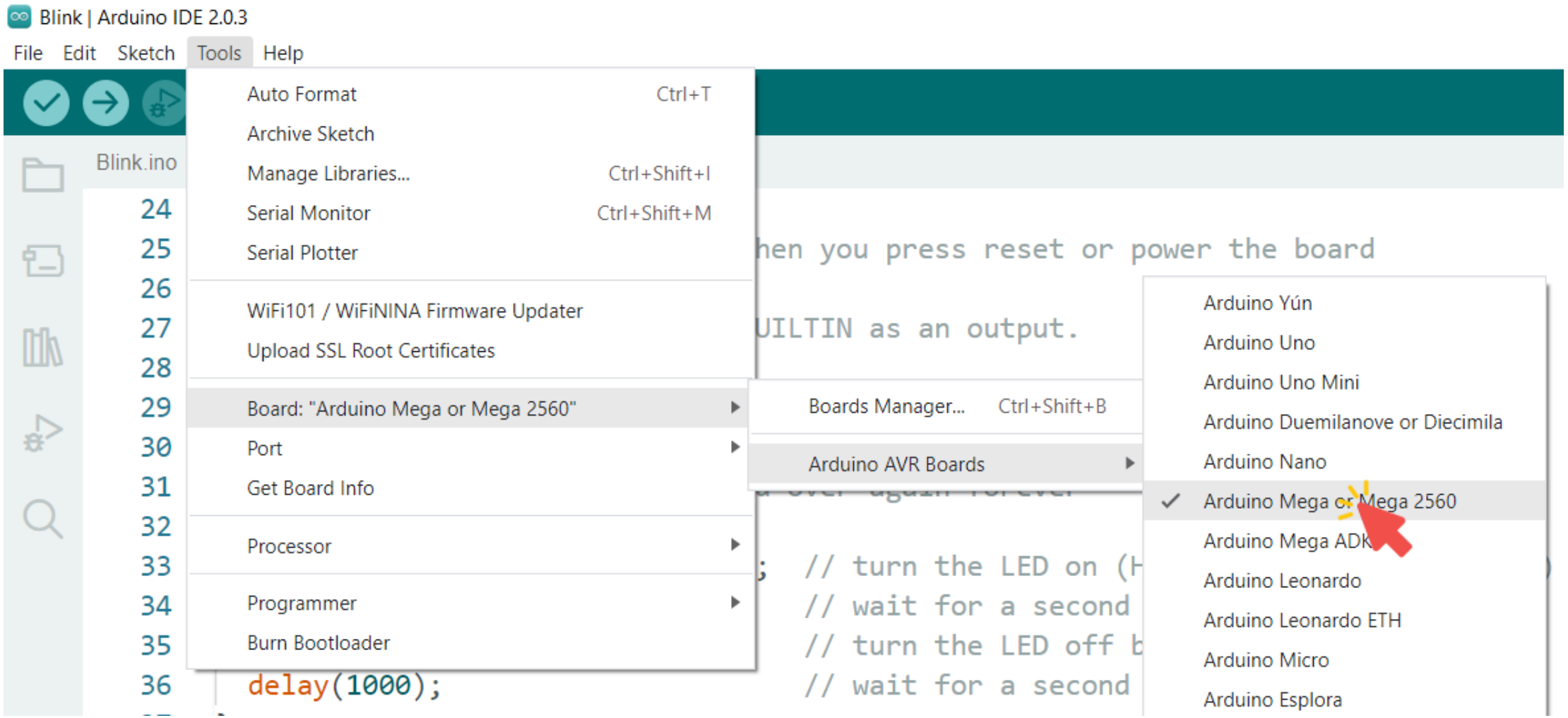


Your First Arduino Project: Code

```
// Turns an LED on for one second, then off for one second, repeatedly.

// The setup function runs once when you press reset or power the board
void setup() {
  // Initialize digital pin LED_BUILTIN (13) as an output.
  pinMode(LED_BUILTIN, OUTPUT);
}

// The loop function runs over and over again forever
void loop() {
  digitalWrite(LED_BUILTIN, HIGH);      // Turn the LED on
  delay(1000);                          // Wait for a second

  digitalWrite(LED_BUILTIN, LOW);       // Turn the LED off
  delay(1000);                          // Wait for a second
}
```

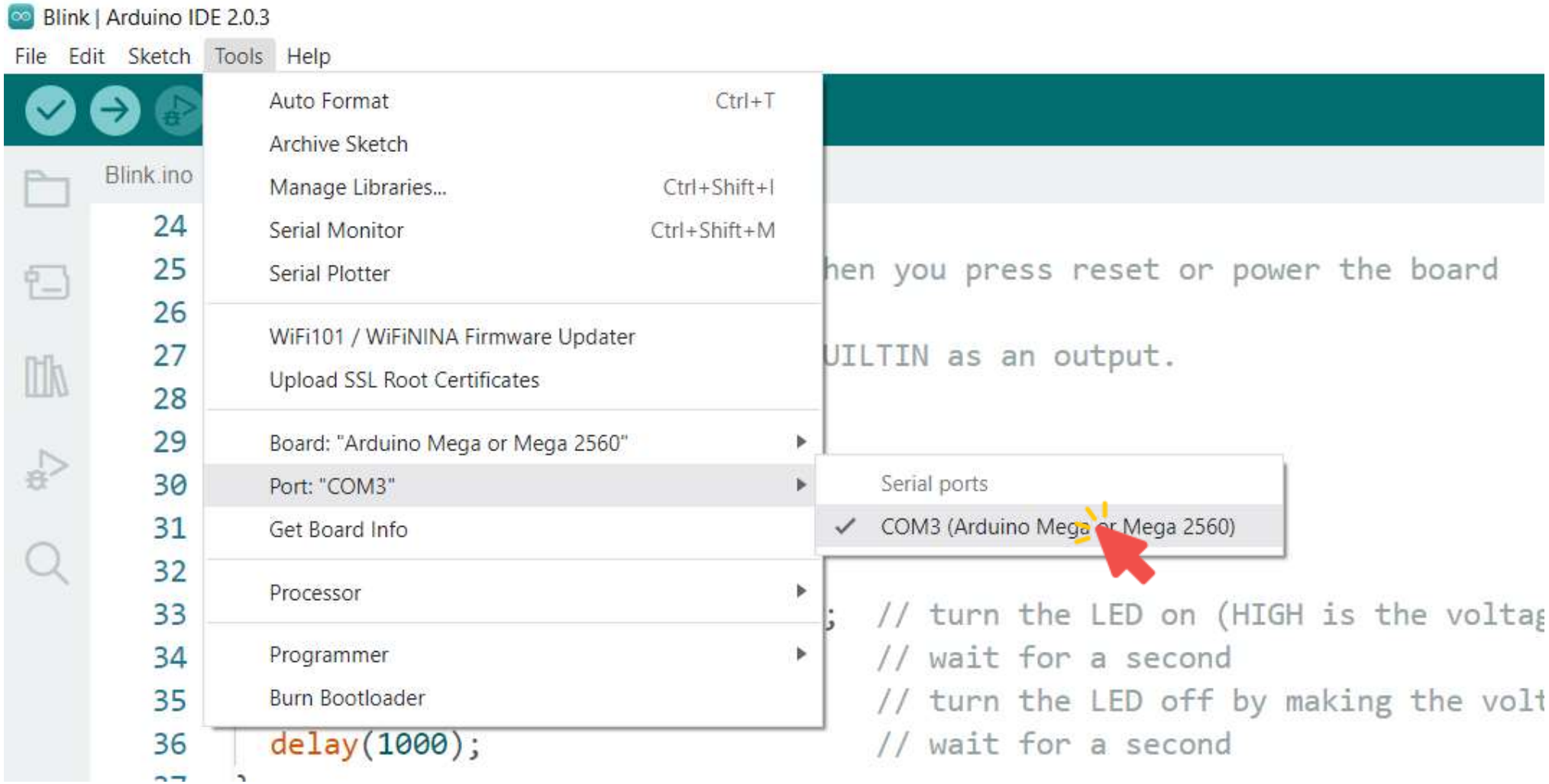
Your First Arduino Project: Arduino AVR Boards

Go to Tools → Board, and select your board.



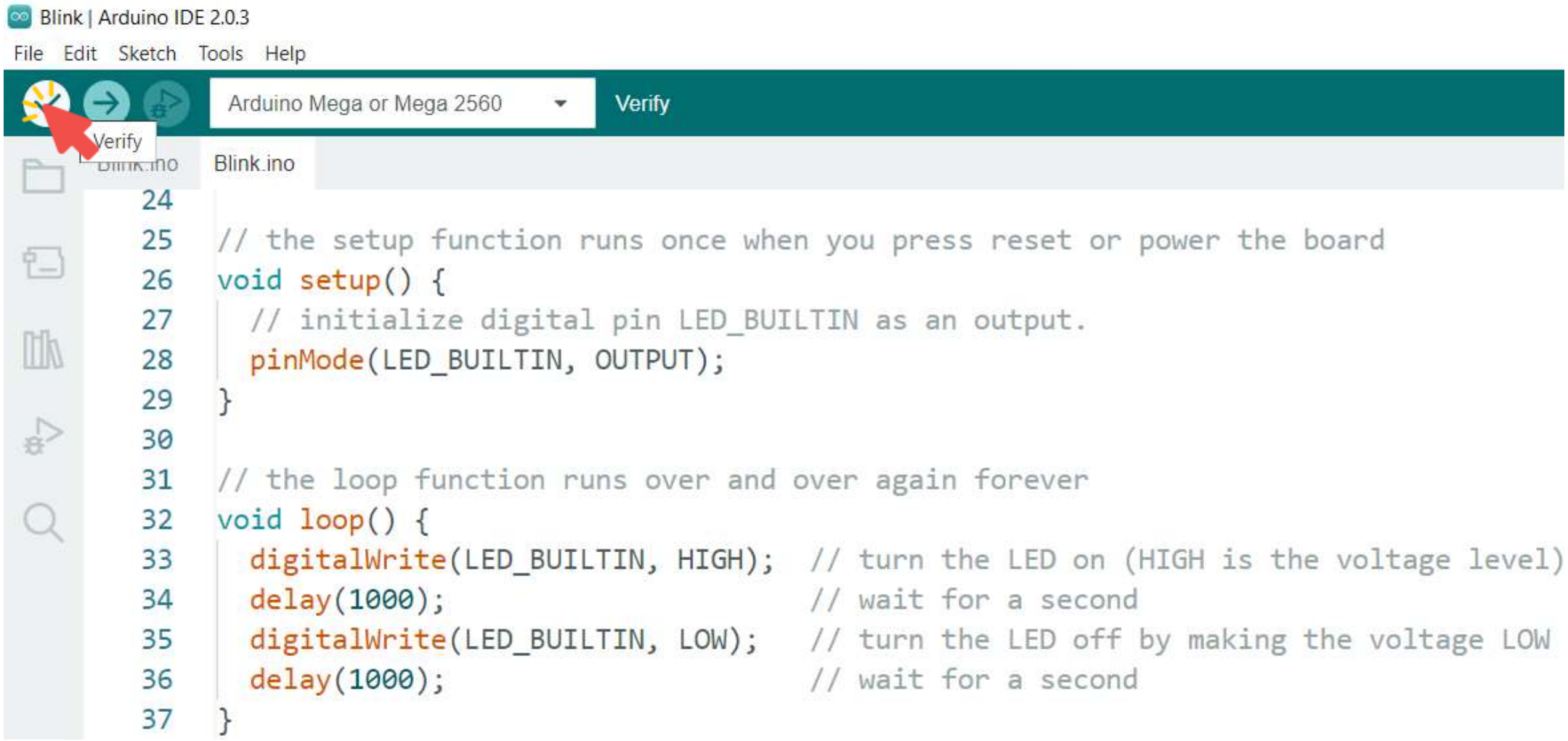
Your First Arduino Project: Port

Go to Tools → Port, and select the port of the Arduino board.



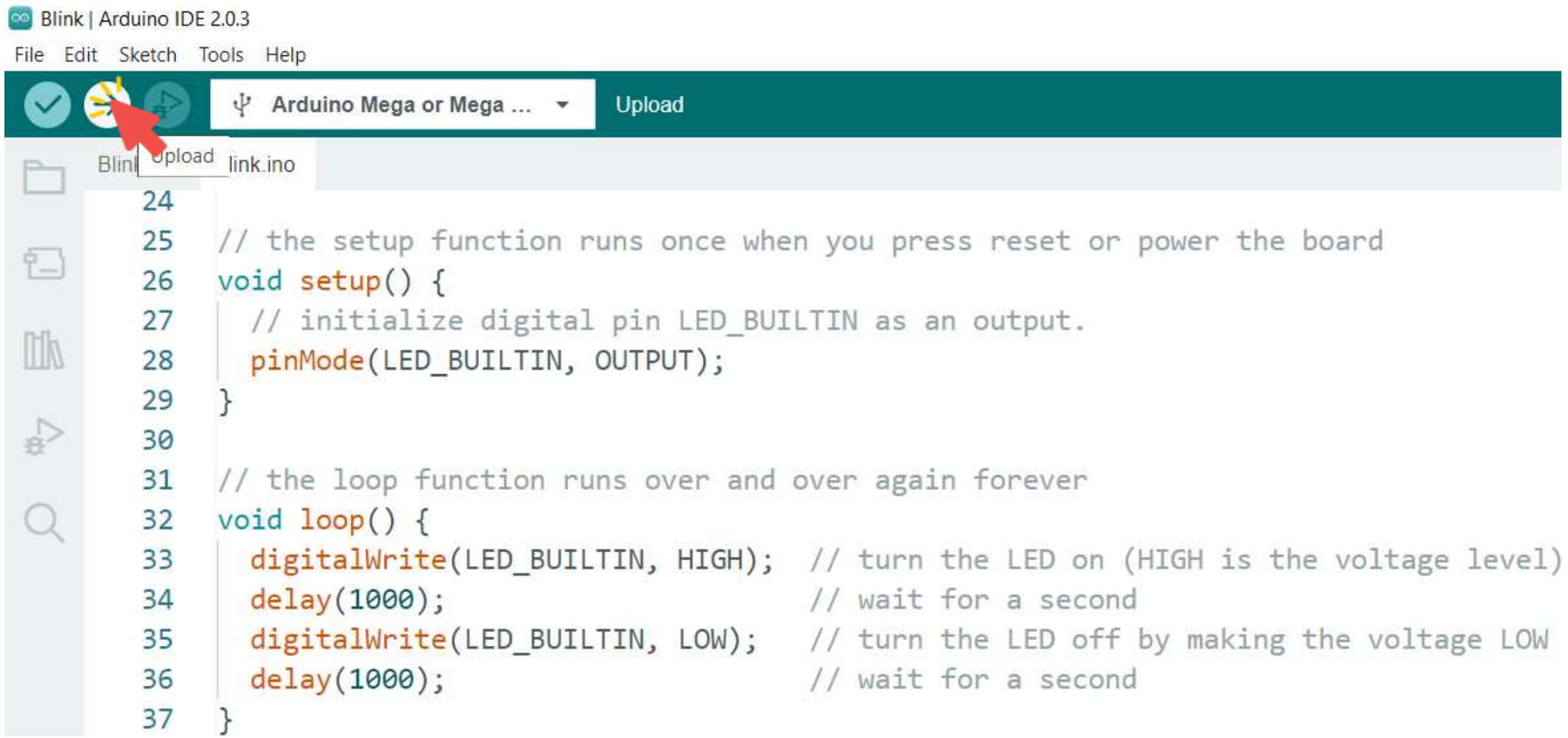
Your First Arduino Project: Verify a Sketch

Click the Verify button to try compiling the sketch and check for errors.



Your First Arduino Project: Upload a Sketch

Click the Upload button to program the board with the sketch.



Your First Arduino Project: Discussion

- The first thing you do is to initialize `LED_BUILTIN` pin as an output pin with the line:

```
pinMode(LED_BUILTIN, OUTPUT);
```

- In the main loop, you turn the LED on with the line:

```
digitalWrite(LED_BUILTIN, HIGH);
```

- Then you turn it off with the line:

```
digitalWrite(LED_BUILTIN, LOW);
```

Your First Arduino Project: Discussion

- The `delay()` causes the Arduino to wait for the specified number of milliseconds before continuing on to the next line.
- There are 1000 milliseconds in a second, so the following line creates a delay of one second.

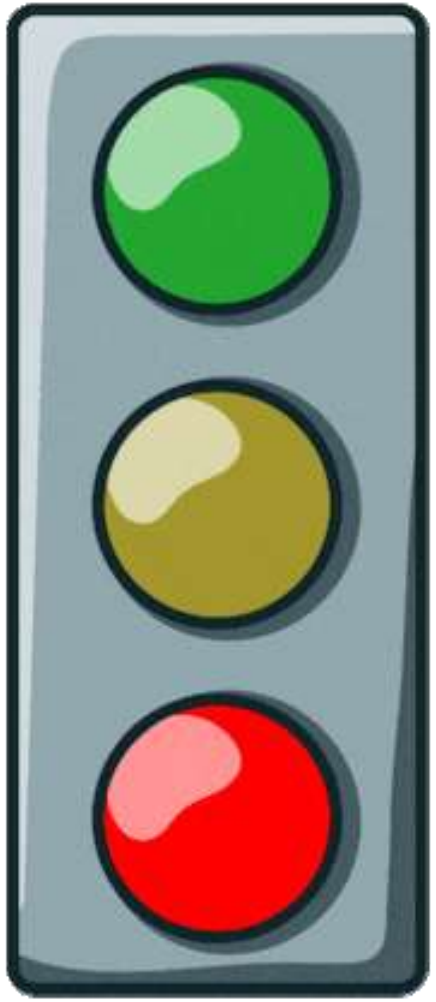
```
delay(1000);
```

- Constants are used to make the programs easier to read.
- The constant `LED_BUILTIN` is the number of the pin to which the on-board LED is connected.
- Most boards have this LED connected to digital pin 13.

Your First Arduino Project: Alternative Code

```
// Turns an LED on for one second, then off for one second, repeatedly.

// The setup function runs once when you press reset or power the board
void setup() {
  // Initialize digital pin 13 as an output.
  pinMode(13, OUTPUT);
}

// The loop function runs over and over again forever
void loop() {
  digitalWrite(13, HIGH);         // Turn the LED on
  delay(1000);                    // Wait for a second

  digitalWrite(13, LOW);          // Turn the LED off
  delay(1000);                    // Wait for a second
}
```

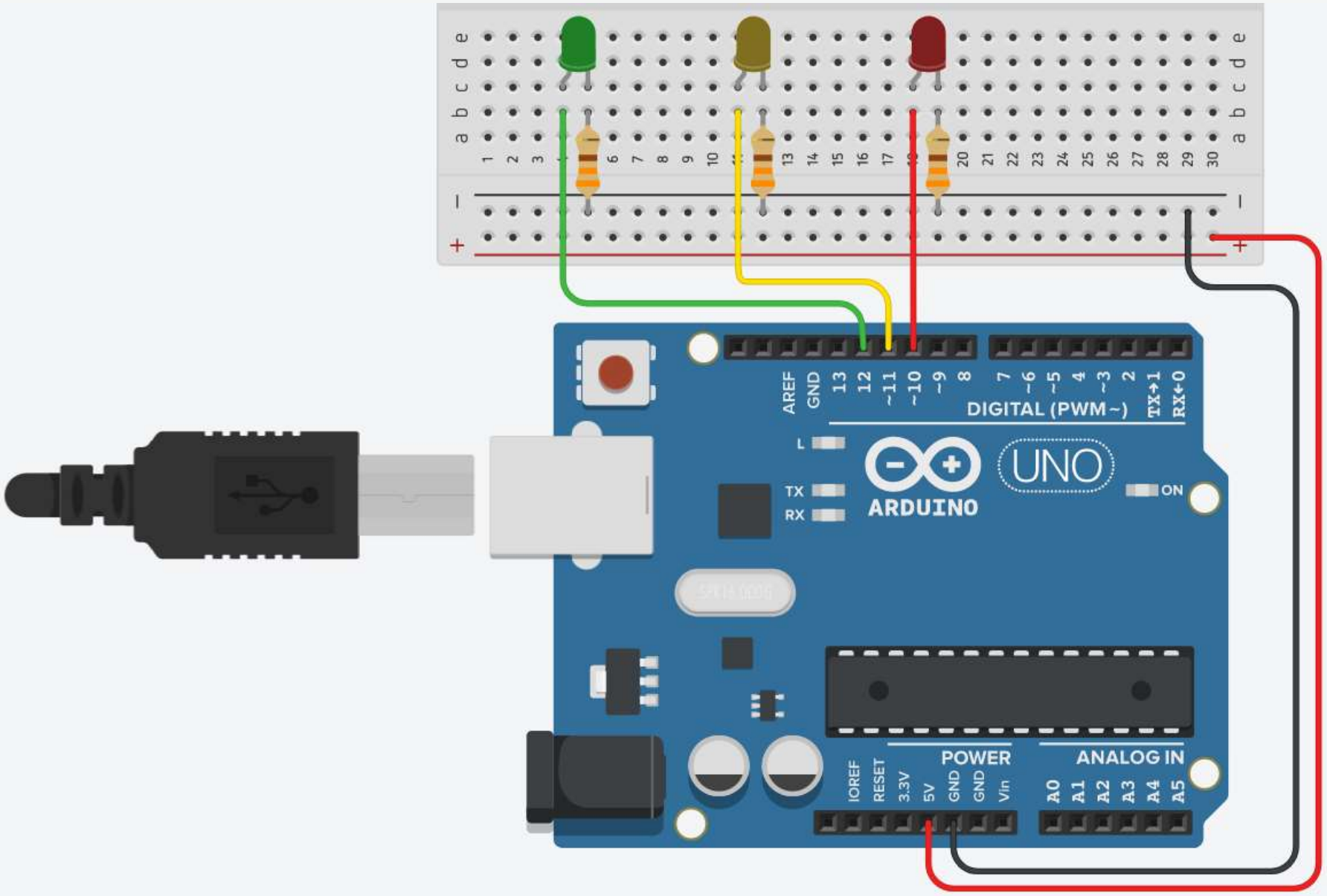
Traffic Light Prototype



Traffic Light Prototype: Components

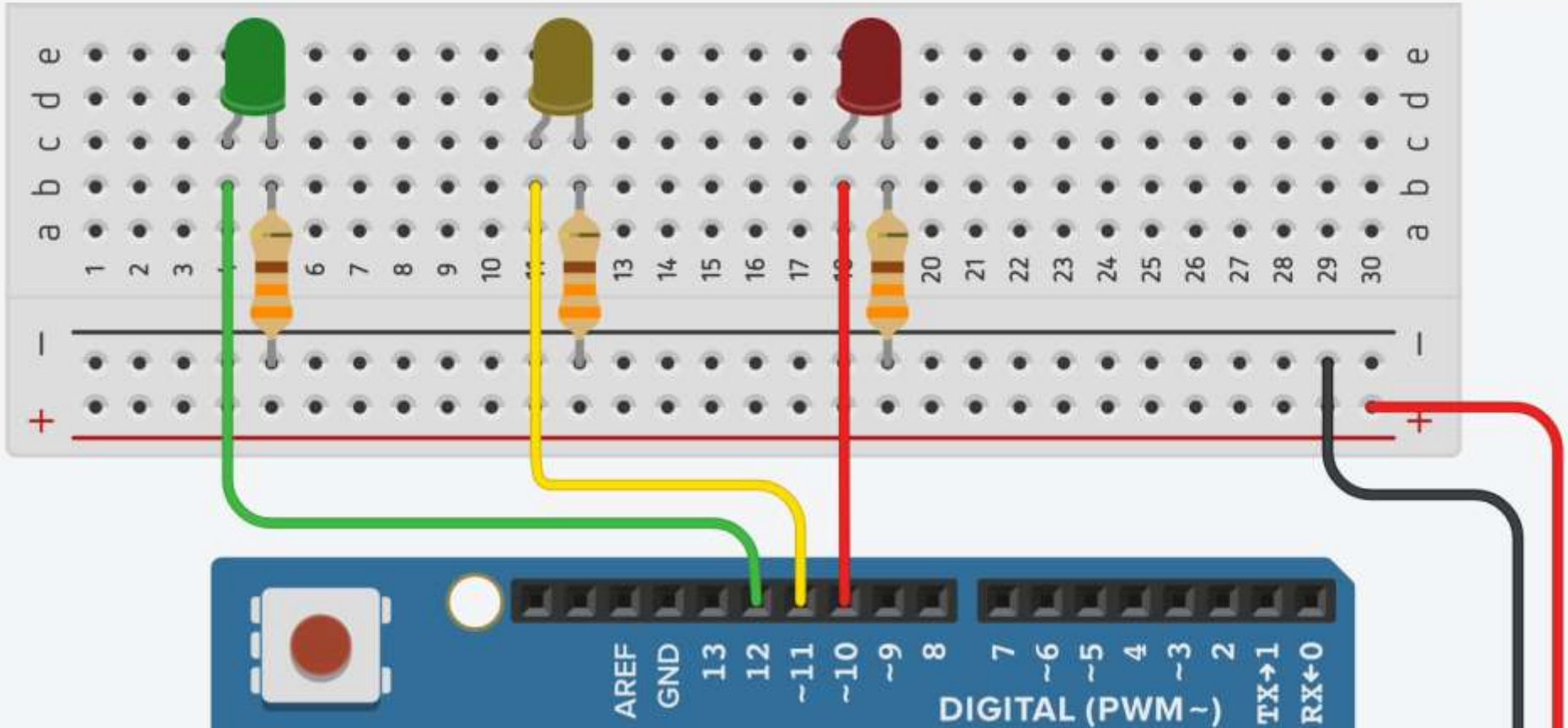
- Components
  - Arduino
  - Breadboard
  - Green LED
  - Yellow LED
  - Red LED
  - 330Ω Resistors
  - Jumpers

Traffic Light Prototype: Circuit



Traffic Light Prototype: Connections

- The Red LED connects to digital pin 10.
- The Yellow LED connects to digital pin 11.
- The Green LED connects to digital pin 12.



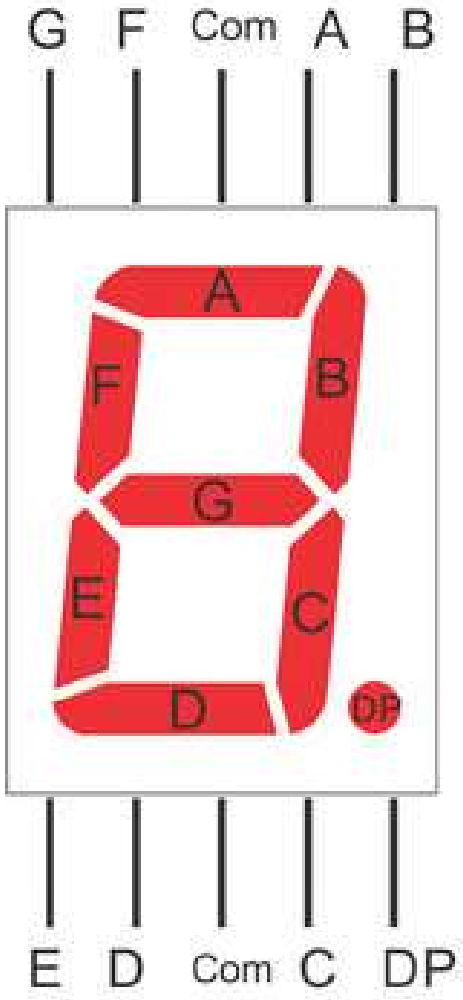
Traffic Light Prototype: Code

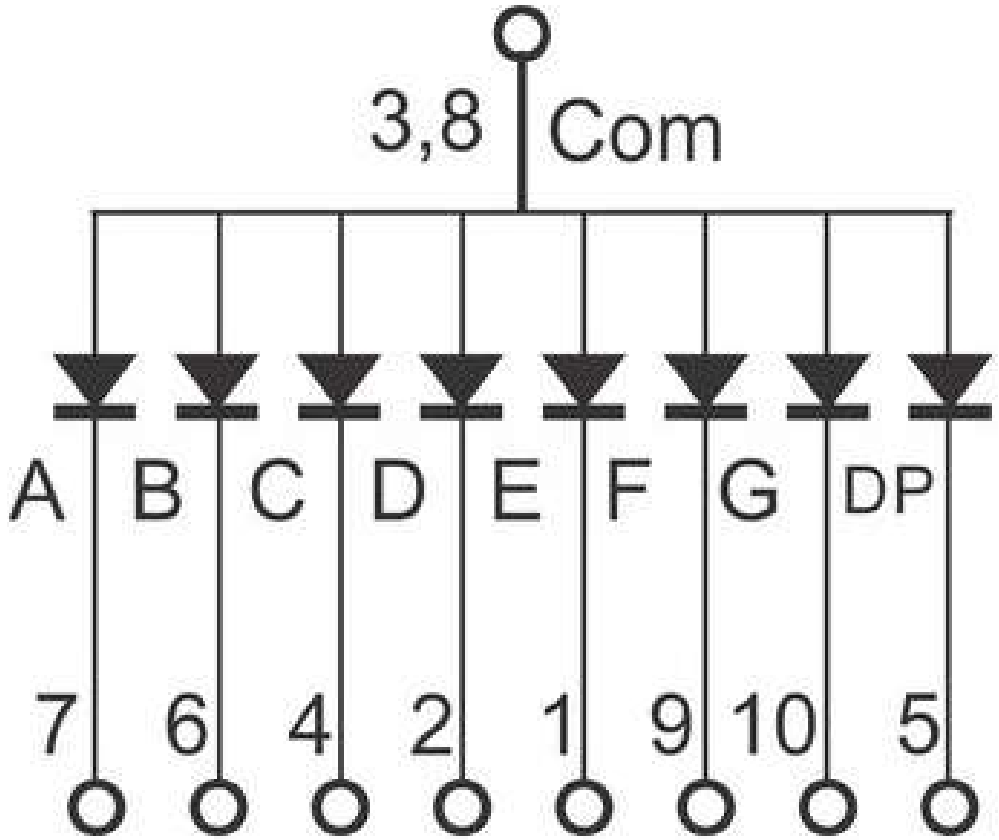
```
#define RED 10                    // The Red LED connects to digital pin 10
#define YELLOW 11                 // The Yellow LED connects to digital pin 11
#define GREEN 12                  // The Green LED connects to digital pin 12

void setup() {
  pinMode(RED, OUTPUT);           // Declare the digital pin 10 as output
  pinMode(YELLOW, OUTPUT);        // Declare the digital pin 11 as output
  pinMode(GREEN, OUTPUT);         // Declare the digital pin 12 as output
}

void loop() {
  digitalWrite(GREEN, HIGH);      // Turn the Green LED on
  digitalWrite(YELLOW, LOW);      // Turn the Yellow LED off
  digitalWrite(RED, LOW);         // Turn the Red LED off
  delay(5000);                    // Wait for 5 seconds

  digitalWrite(GREEN, LOW);       // Turn the Green LED off
  digitalWrite(YELLOW, HIGH);     // Turn the Yellow LED on
  digitalWrite(RED, LOW);         // Turn the Red LED
  delay(2000);                    // Wait for 2 seconds

  digitalWrite(GREEN, LOW);       // Turn the Green LED off
  digitalWrite(YELLOW, LOW);      // Turn the Yellow LED off
  digitalWrite(RED, HIGH);        // Turn the Red LED on
  delay(5000);                    // Wait for 5 seconds
}
```

Traffic Light Prototype: Dirty Code

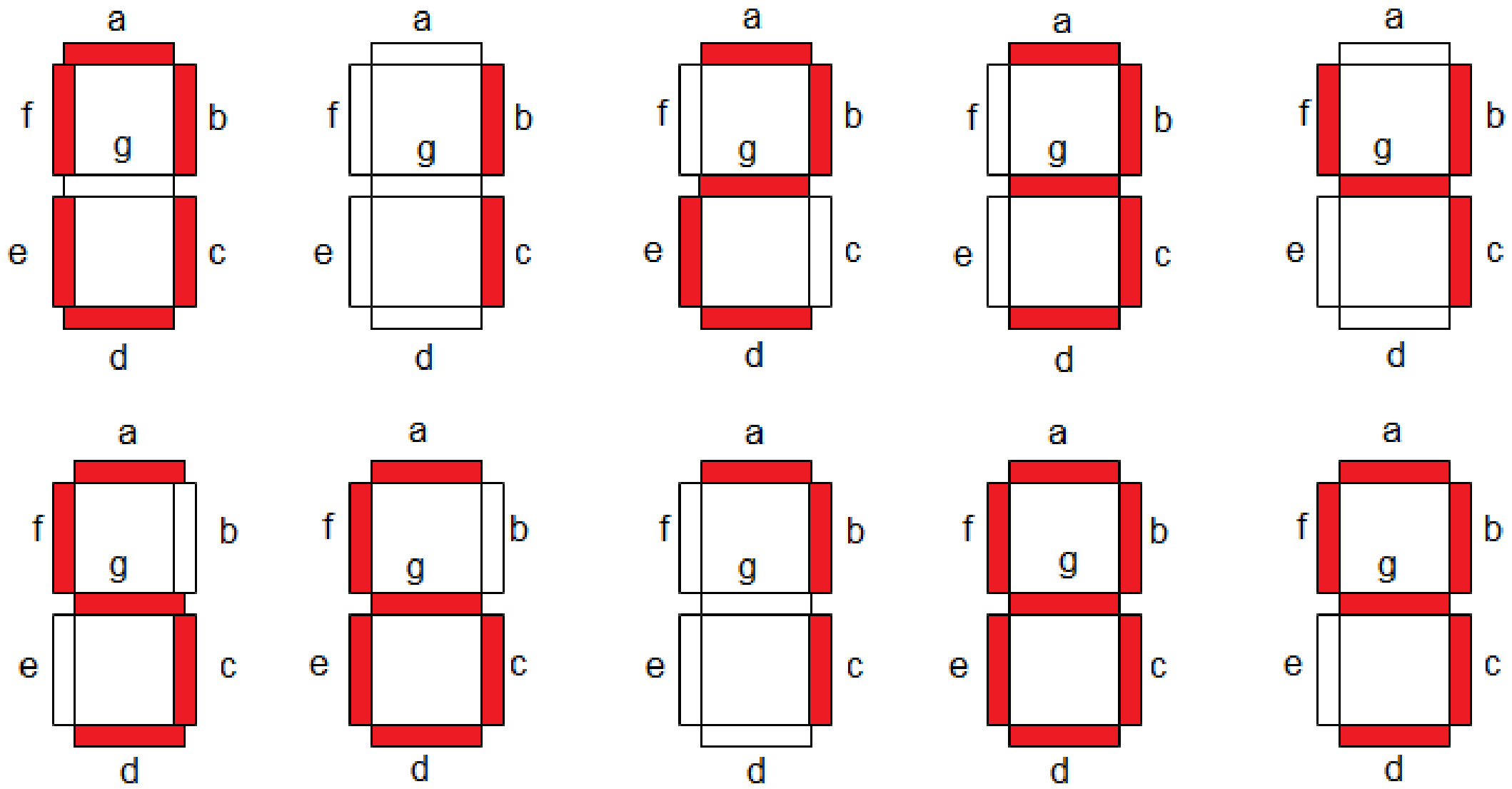
```
void setup() {
  pinMode(10, OUTPUT);
  pinMode(11, OUTPUT);
  pinMode(12, OUTPUT);
}

void loop() {
  digitalWrite(12, HIGH);
  digitalWrite(11, LOW);
  digitalWrite(10, LOW);
  delay(5000);

  digitalWrite(12, LOW);
  digitalWrite(11, HIGH);
  digitalWrite(10, LOW);
  delay(2000);

  digitalWrite(12, LOW);
  digitalWrite(11, LOW);
  digitalWrite(10, HIGH);
  delay(5000);
}
```

7-Segment Display





7-Segment Display



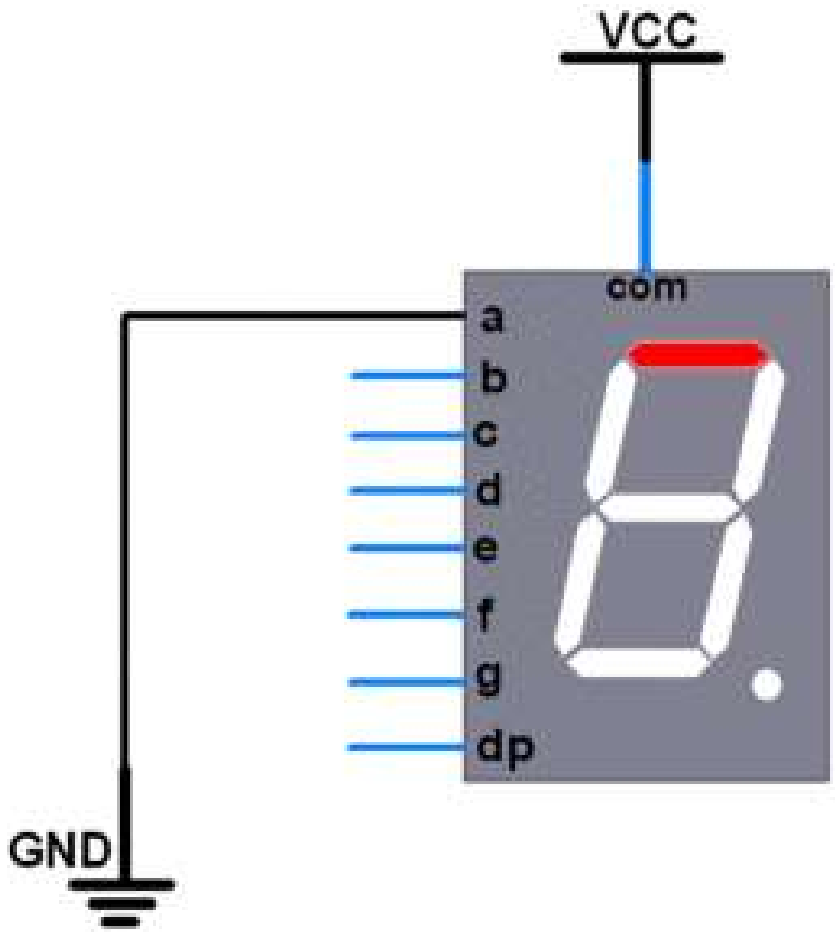
7-Segment Display

Digit	gfedcba	abcdefg	a	b	c	d	e	f	g
0	0×3F	0×7E	**on**	**on**	**on**	**on**	**on**	**on**	off
1	0×06	0×30	off	**on**	**on**	off	off	off	off
2	0×5B	0×6D	**on**	**on**	off	**on**	**on**	off	**on**
3	0×4F	0×79	**on**	**on**	**on**	**on**	off	off	**on**
4	0×66	0×33	off	**on**	**on**	off	off	**on**	**on**
5	0×6D	0×5B	**on**	off	**on**	**on**	off	**on**	**on**
6	0×7D	0×5F	**on**	off	**on**	**on**	**on**	**on**	**on**
7	0×07	0×70	**on**	**on**	**on**	off	off	off	off
8	0×7F	0×7F	**on**	**on**	**on**	**on**	**on**	**on**	**on**
9	0×6F	0×7B	**on**	**on**	**on**	**on**	off	**on**	**on**

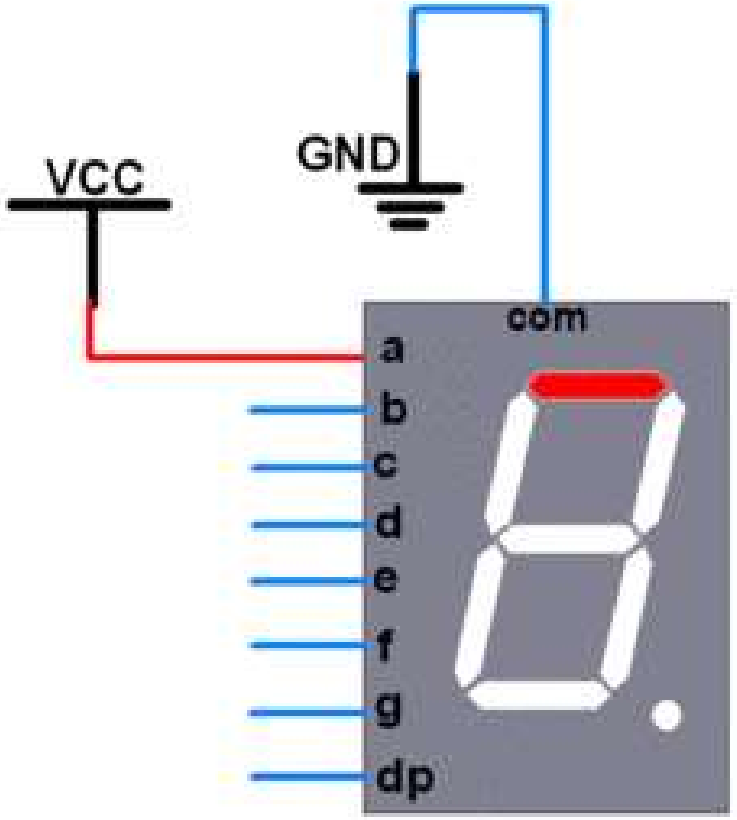
7-Segment Display

Digit	gfedcba	abcdefg	a	b	c	d	e	f	g
A	0×77	0×77	**on**	**on**	**on**	off	**on**	**on**	**on**
B	0×7C	0×1F	off	off	**on**	**on**	**on**	**on**	**on**
C	0×39	0×4E	**on**	off	off	**on**	**on**	**on**	off
D	0×5E	0×3D	off	**on**	**on**	**on**	**on**	off	**on**
E	0×79	0×4F	**on**	off	off	**on**	**on**	**on**	**on**
F	0×71	0×47	**on**	off	off	off	**on**	**on**	**on**

7-Segment Display: Common Anode vs. Common Cathode

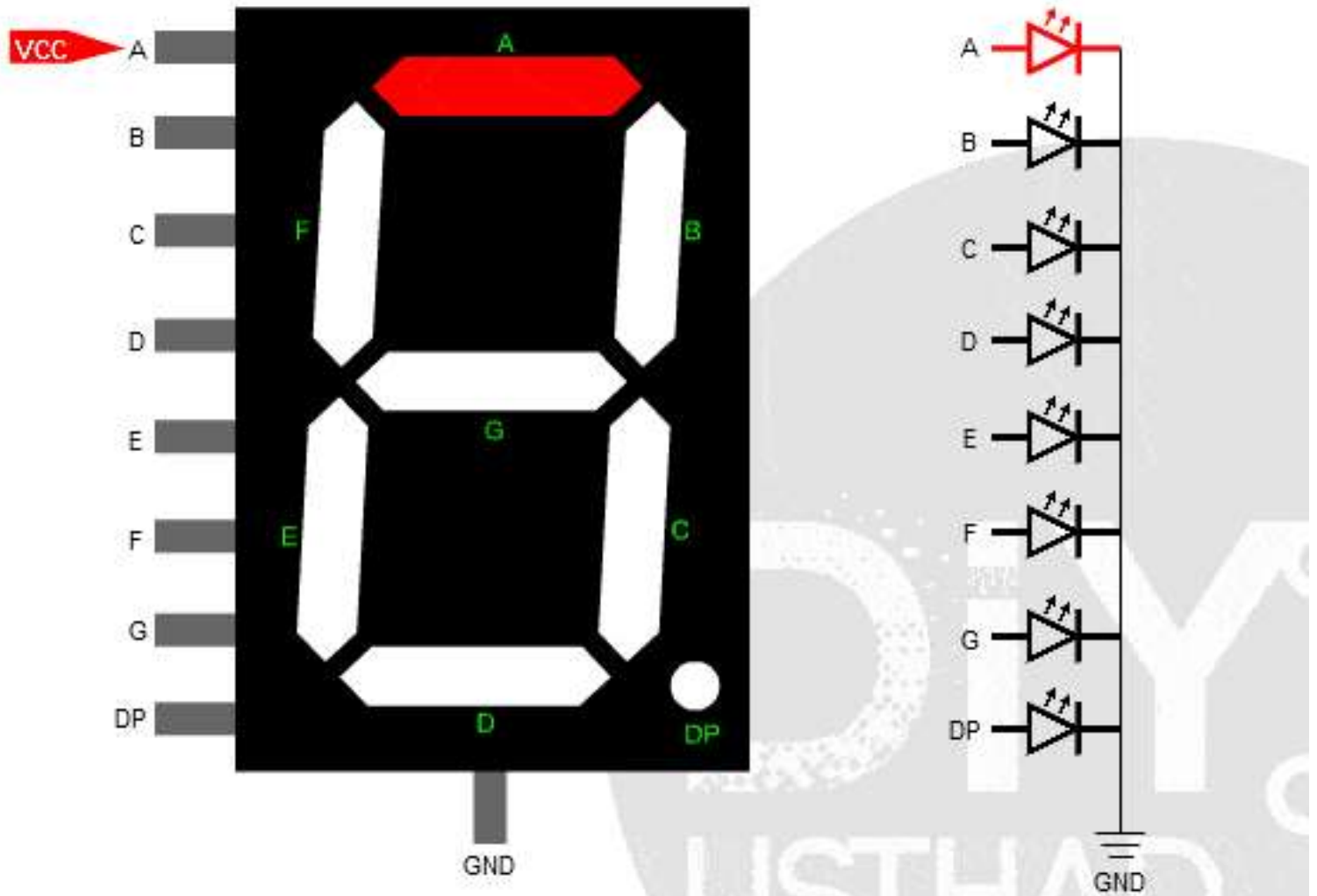


Common Anode

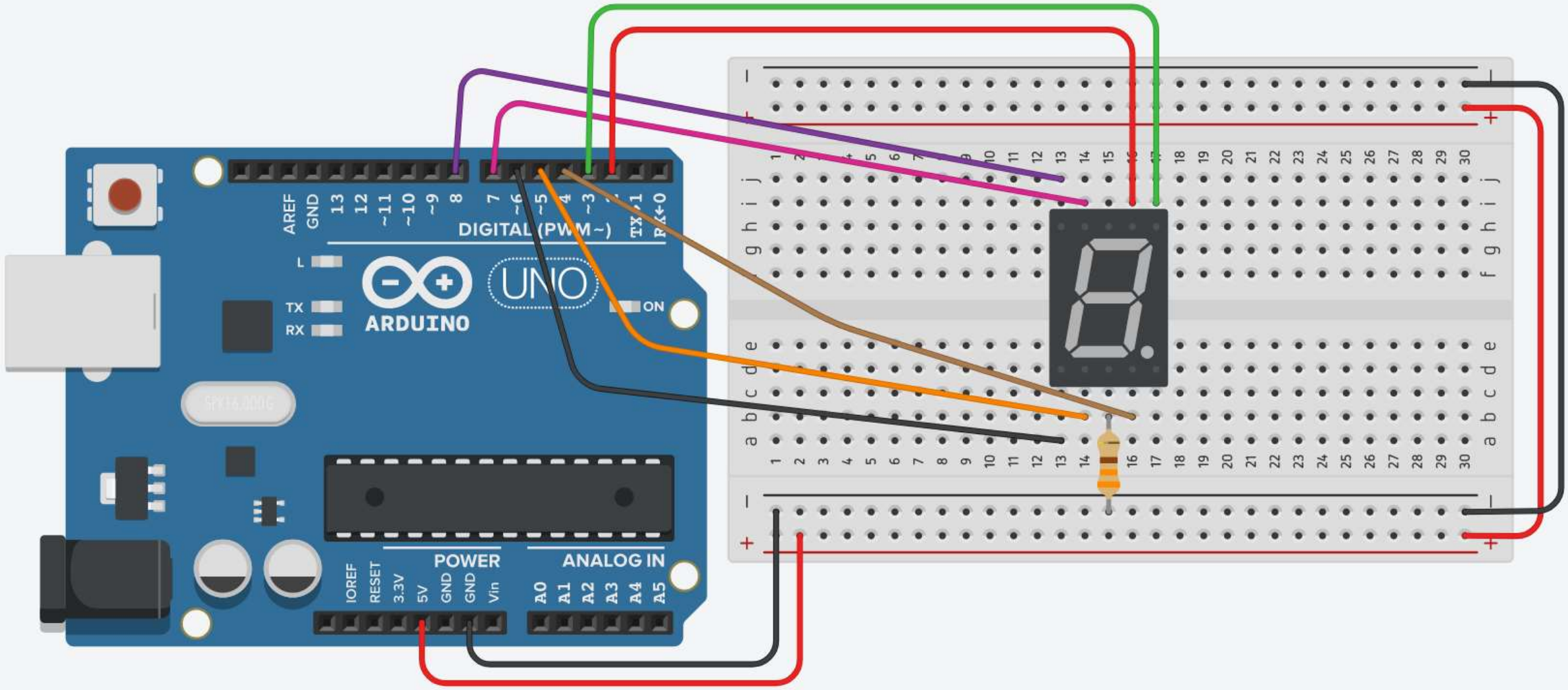


Common Cathode

Common Cathode 7-Segment Display



Common Cathode 7-Segment Display: Circuit

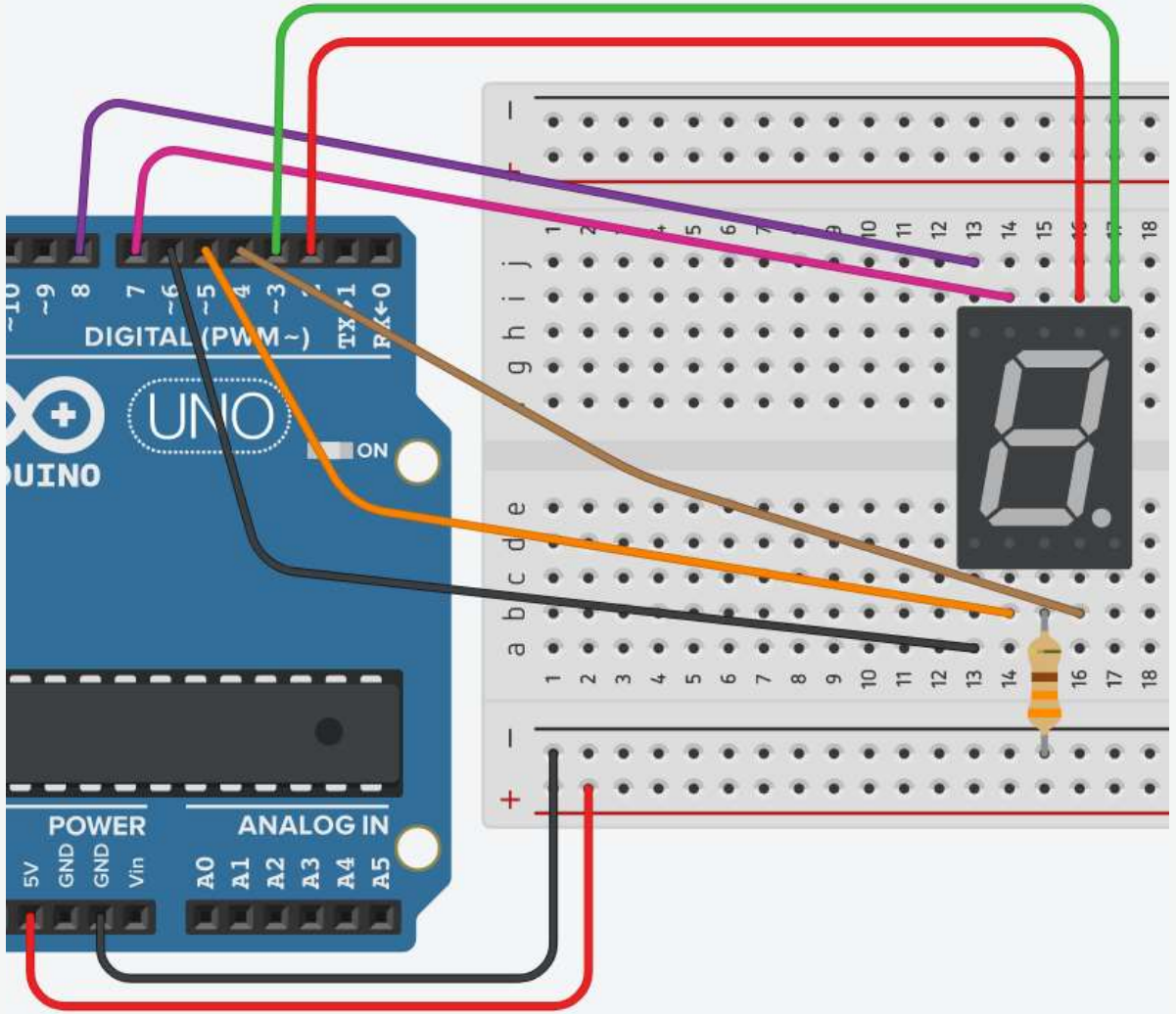


Common Cathode 7-Segment Display: Components

- Components
  - Arduino
  - Breadboard
  - 7-Segment Display (Common Cathode)
  - 330Ω Resistor
  - Jumpers

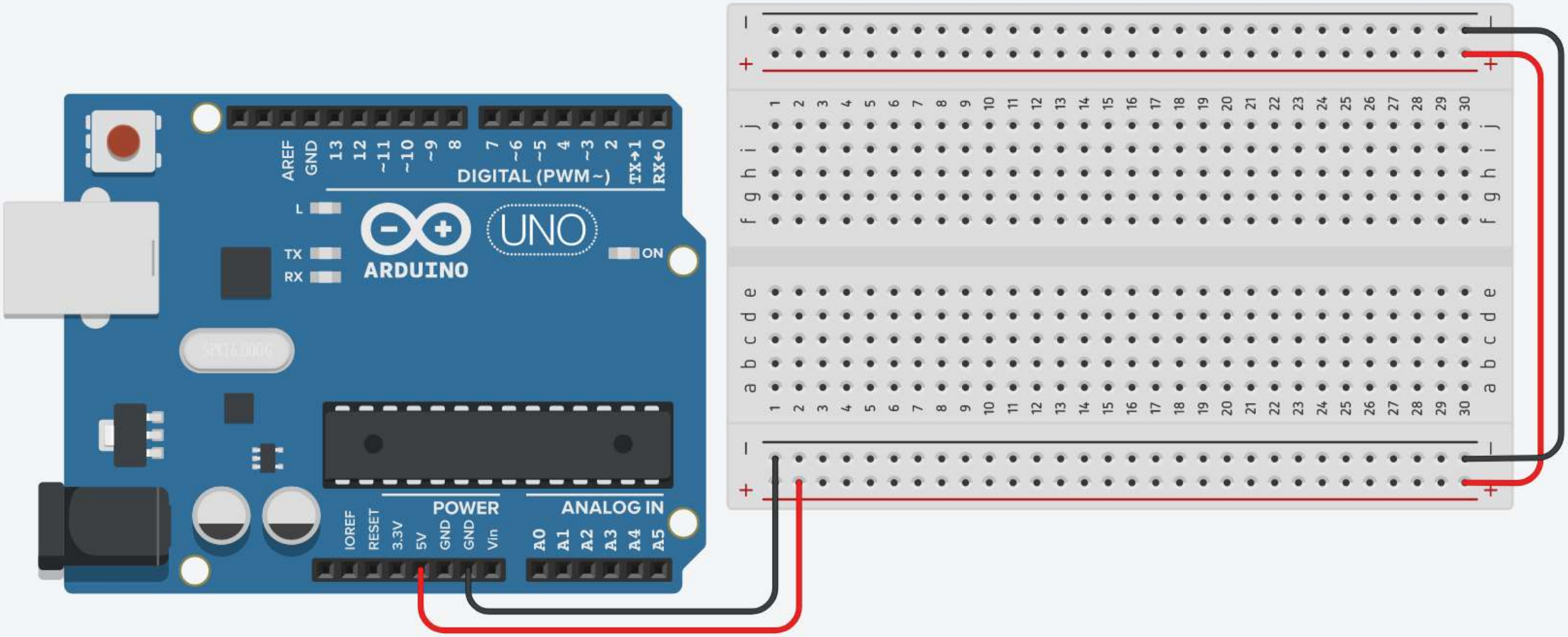
Common Cathode 7-Segment Display: Connections

- The pin a connects to digital pin 2.
- The pin b connects to digital pin 3.
- The pin c connects to digital pin 4.
- The pin d connects to digital pin 5.
- The pin e connects to digital pin 6.
- The pin f connects to digital pin 7.
- The pin g connects to digital pin 8.



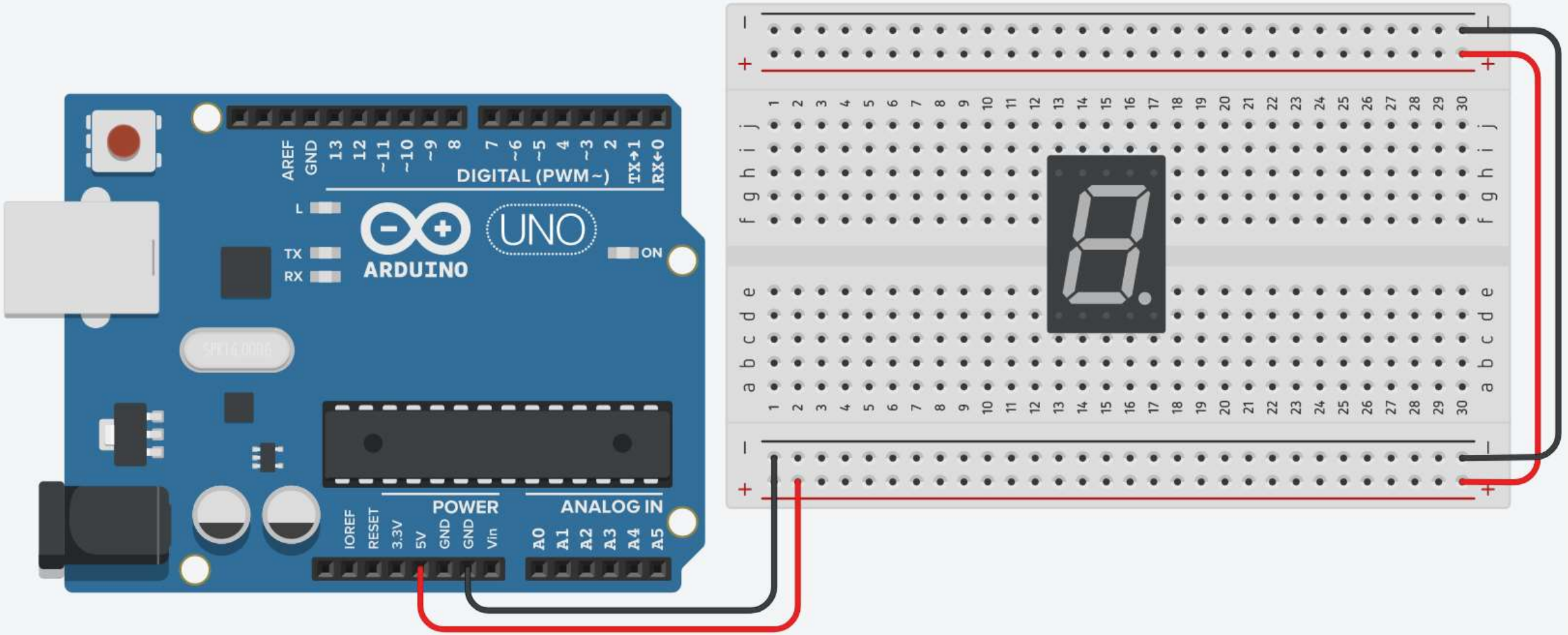
Common Cathode 7-Segment Display: Steps

1. Connect breadboard power (+) and ground (-) rails to Arduino 5V and ground (GND), respectively.



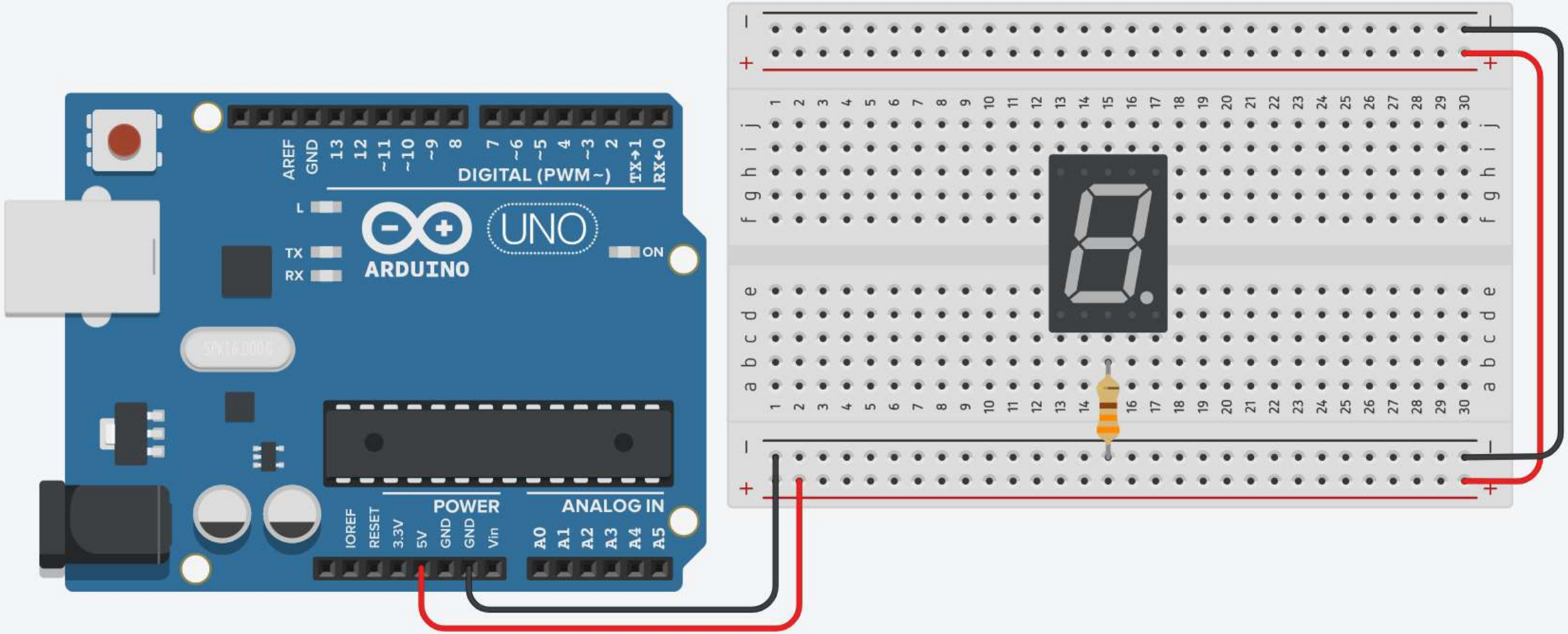
Common Cathode 7-Segment Display: Steps

2. Plug the 7-segment display into the breadboard.



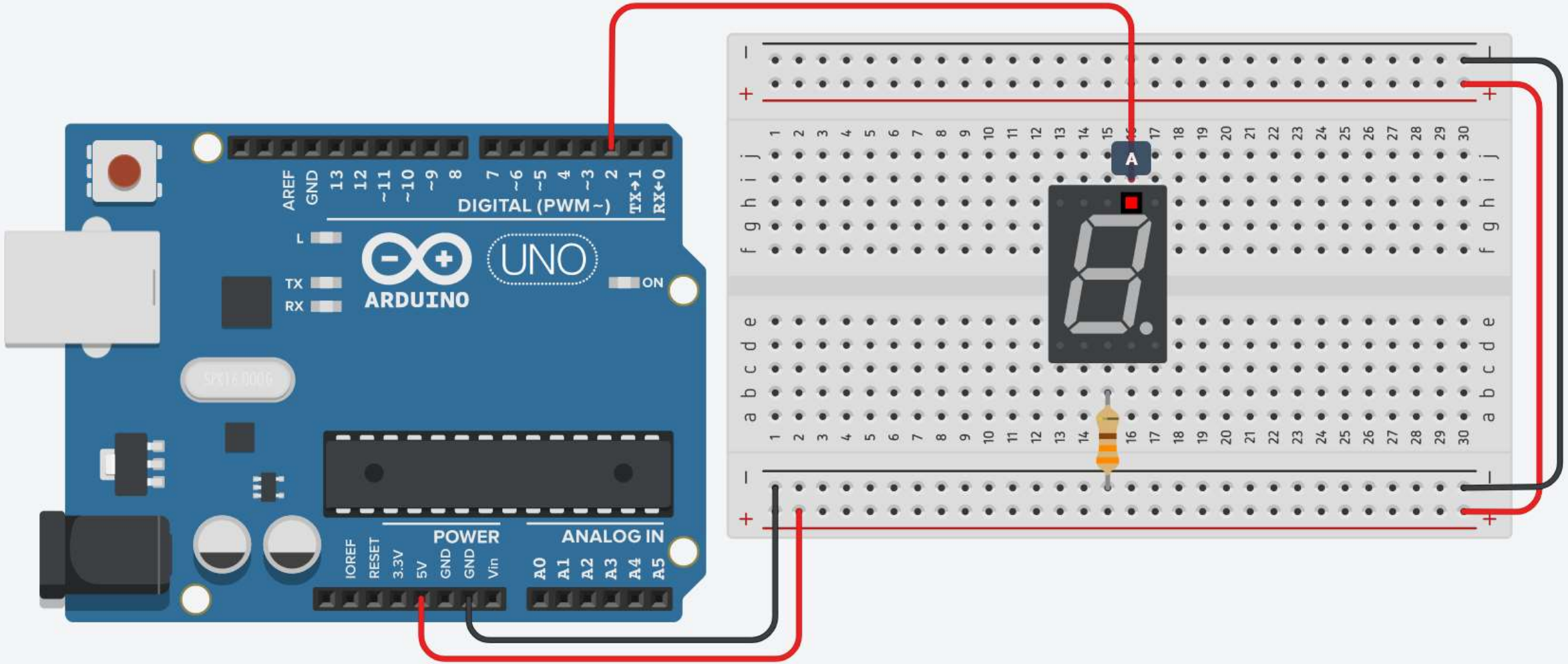
Common Cathode 7-Segment Display: Steps

3. Connect the common of the 7-segment to the ground with a 330Ω resistor.



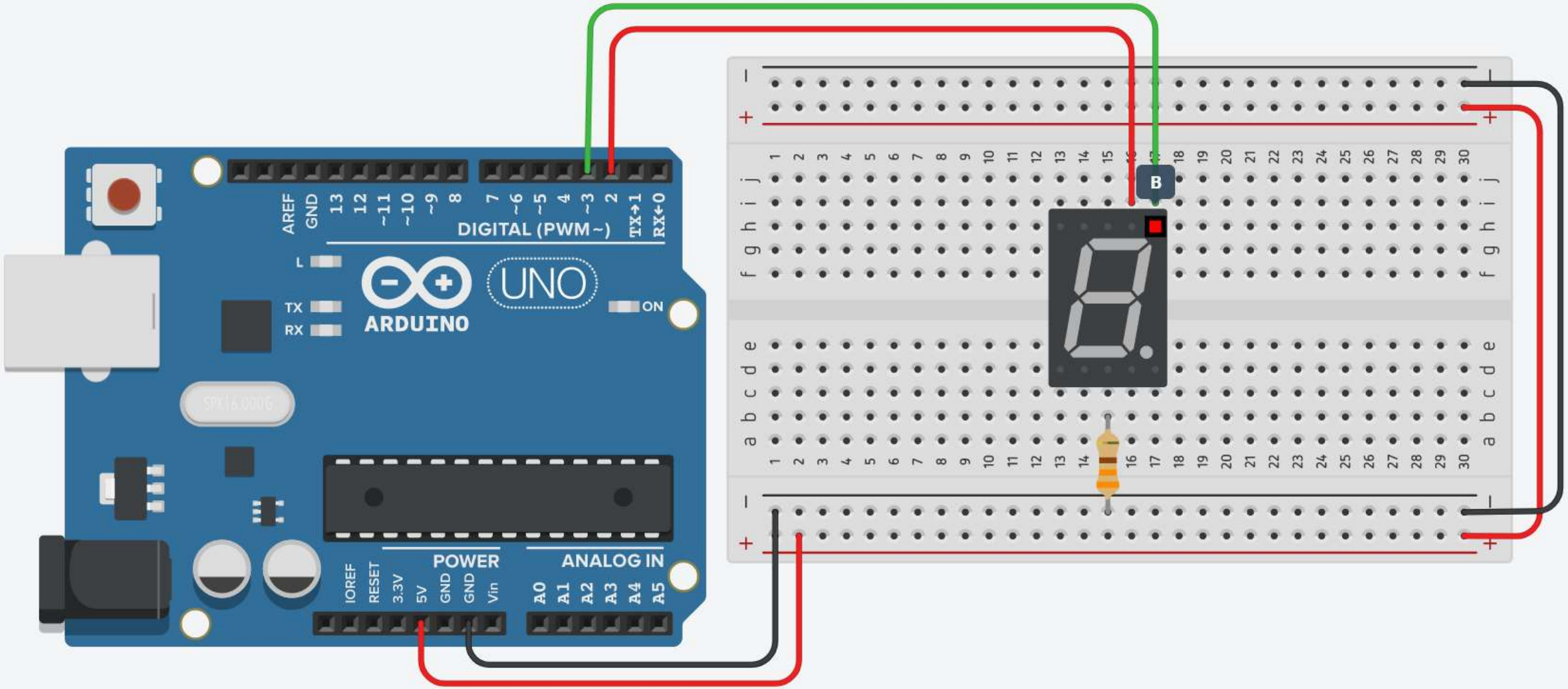
Common Cathode 7-Segment Display: Steps

4. Connect pin a to digital pin 2 on Arduino.



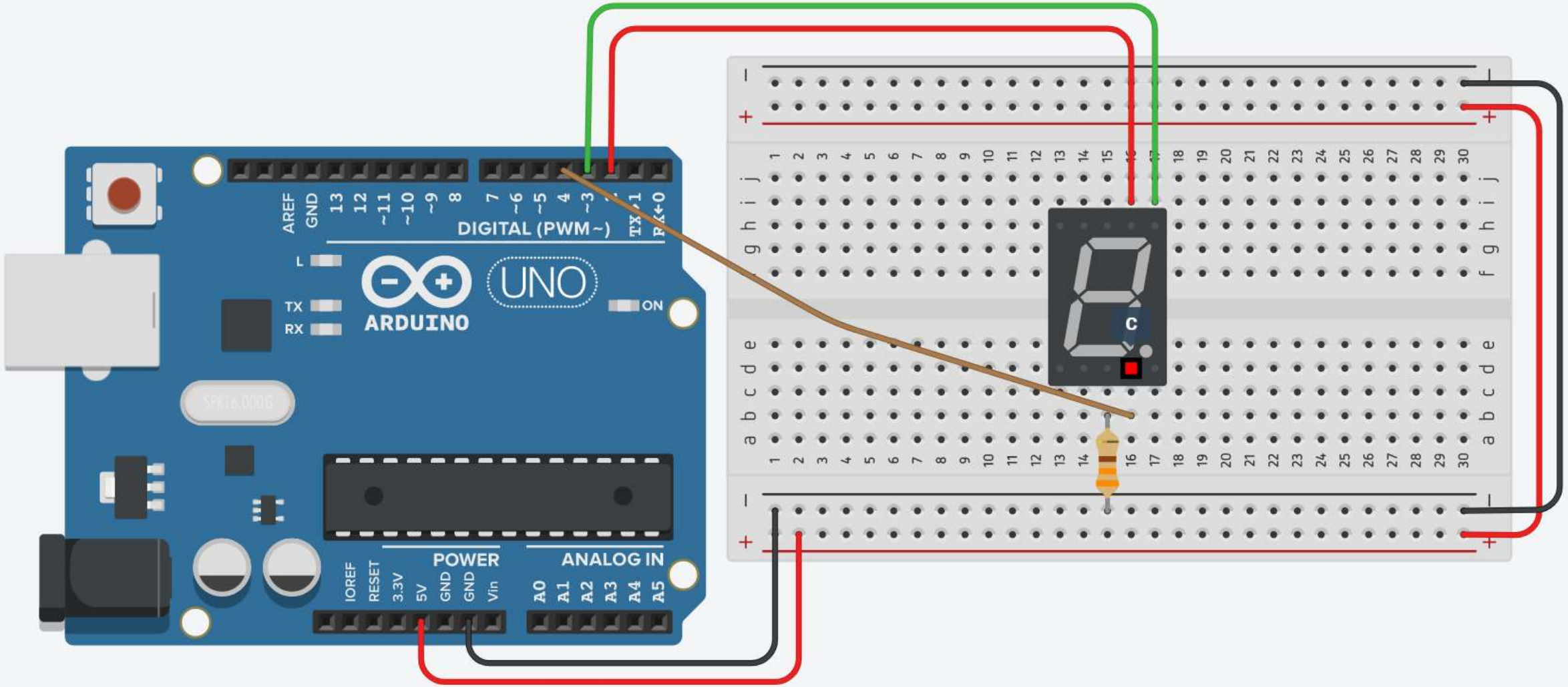
Common Cathode 7-Segment Display: Steps

5. Connect pin b to digital pin 3 on Arduino.



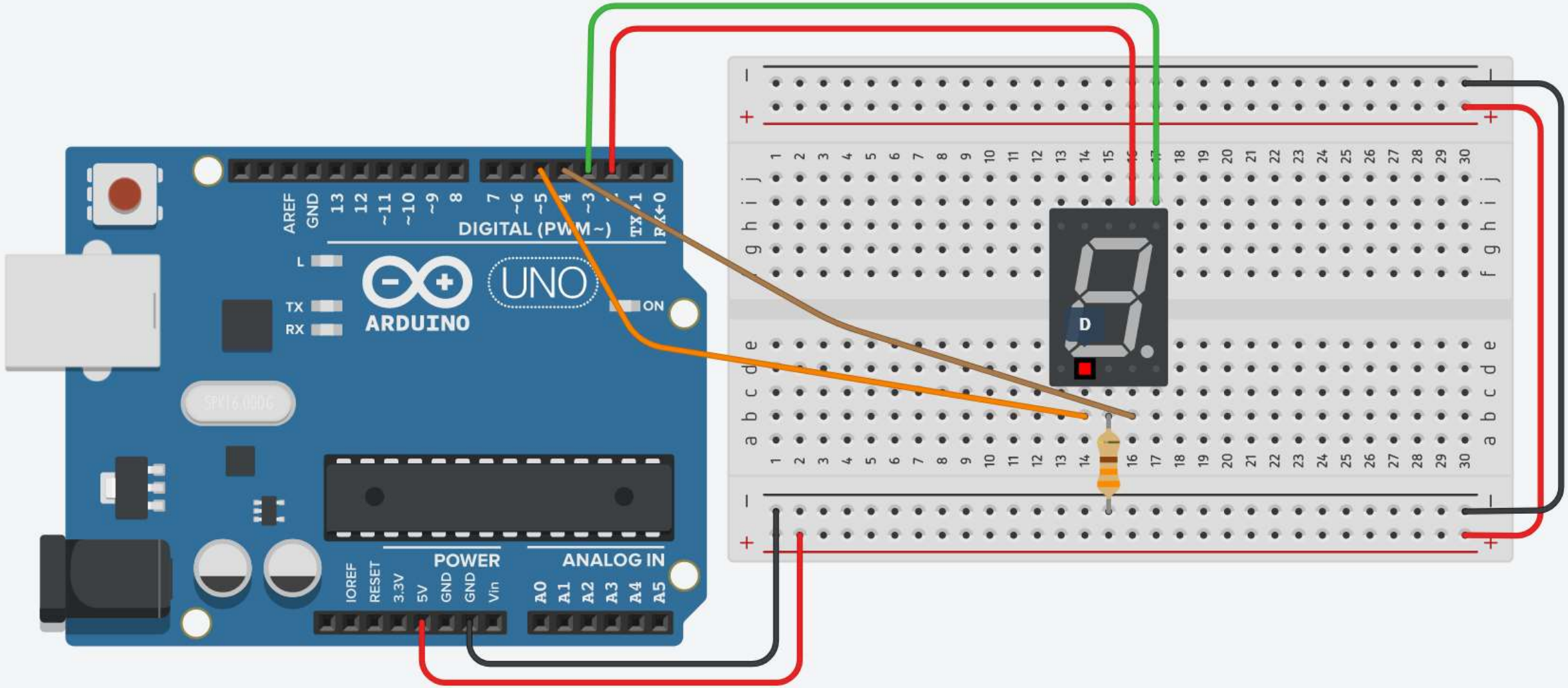
Common Cathode 7-Segment Display: Steps

6. Connect pin c to digital pin 4 on Arduino.



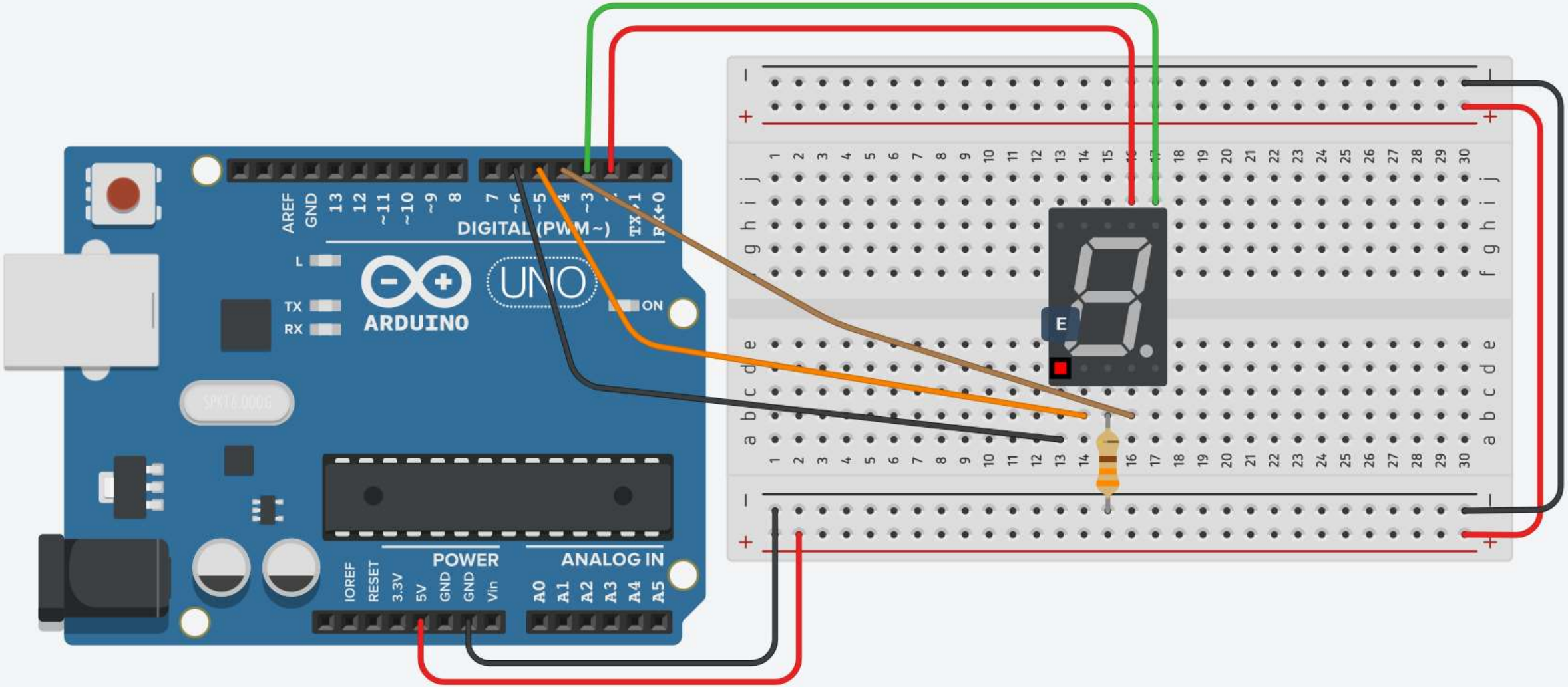
Common Cathode 7-Segment Display: Steps

7. Connect pin d to digital pin 5 on Arduino.



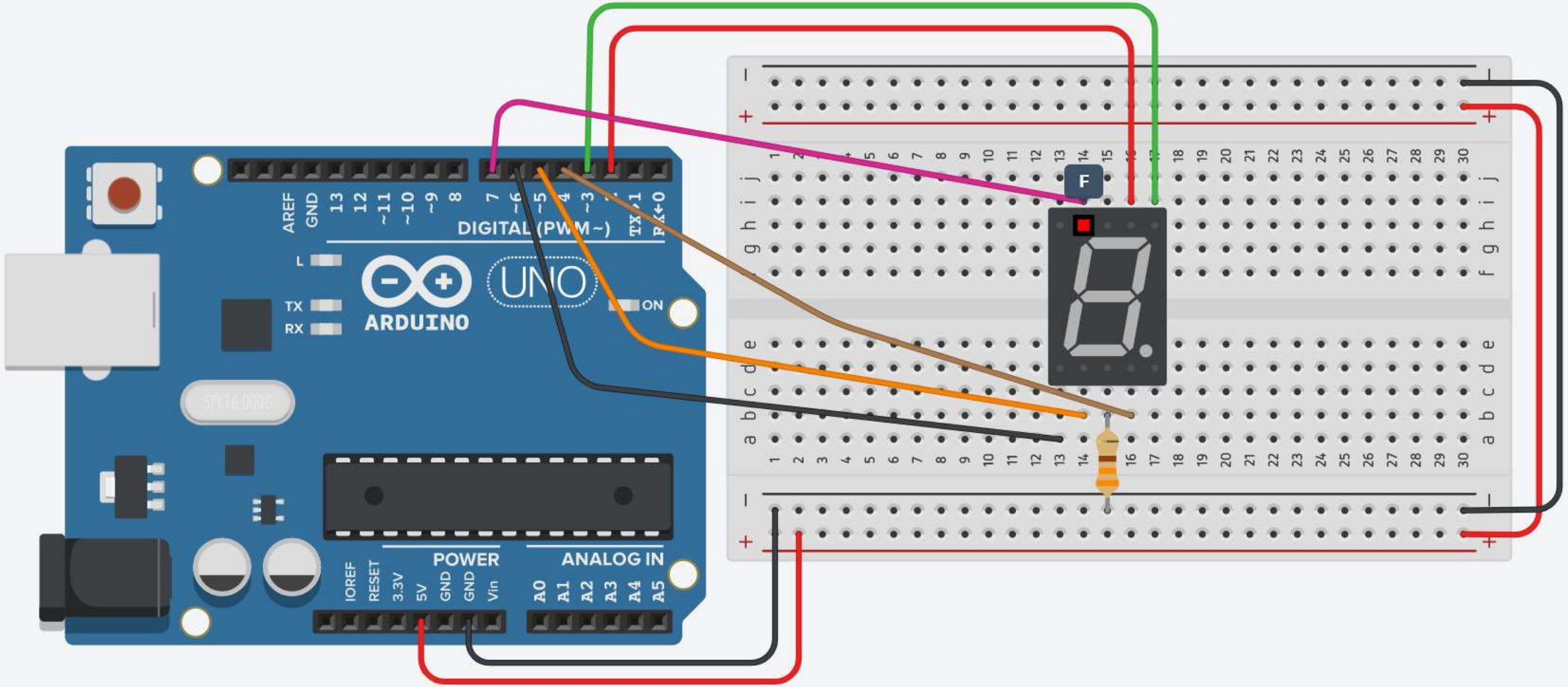
Common Cathode 7-Segment Display: Steps

8. Connect pin e to digital pin 6 on Arduino.



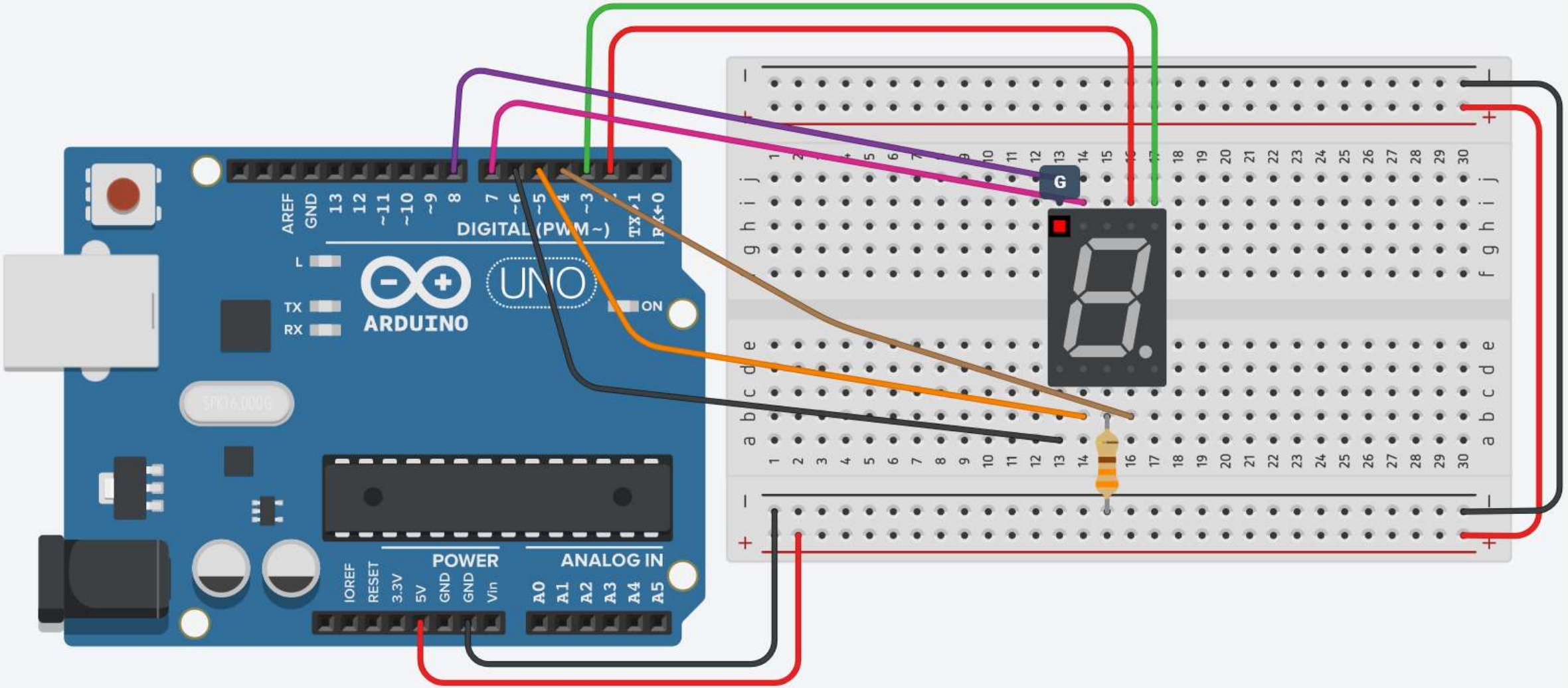
Common Cathode 7-Segment Display: Steps

9. Connect pin f to digital pin 7 on Arduino.



Common Cathode 7-Segment Display: Steps

10. Connect pin g to digital pin 8 on Arduino.



Common Cathode 7-Segment Display: Code

```
// Common cathode 7-segment display

#define a 2
#define b 3
#define c 4
#define d 5
#define e 6
#define f 7
#define g 8

void setup() {
  // Define pin modes
  pinMode(a, OUTPUT);
  pinMode(b, OUTPUT);
  pinMode(c, OUTPUT);
  pinMode(d, OUTPUT);
  pinMode(e, OUTPUT);
  pinMode(f, OUTPUT);
  pinMode(g, OUTPUT);

  // Clear the 7-segment display
  reset();
}
```

Common Cathode 7-Segment Display: Code

```
void loop() {
  // BCD Counter
  for(int digit = 0; digit <= 9; digit++){
    // Display the digit on the 7-segment display
    display(digit);

    // Wait for a second
    delay(1000);
  }
}

void reset(){
  // Clear the 7-segment display
  digitalWrite(a, LOW);
  digitalWrite(b, LOW);
  digitalWrite(c, LOW);
  digitalWrite(d, LOW);
  digitalWrite(e, LOW);
  digitalWrite(f, LOW);
  digitalWrite(g, LOW);
}
```

Common Cathode 7-Segment Display: Code

```
void display(int digit){
  if(digit == 0)
  {
    digitalWrite(a, HIGH);
    digitalWrite(b, HIGH);
    digitalWrite(c, HIGH);
    digitalWrite(d, HIGH);
    digitalWrite(e, HIGH);
    digitalWrite(f, HIGH);
    digitalWrite(g, LOW);
  }
  else if(digit == 1)
  {
    digitalWrite(a, LOW);
    digitalWrite(b, HIGH);
    digitalWrite(c, HIGH);
    digitalWrite(d, LOW);
    digitalWrite(e, LOW);
    digitalWrite(f, LOW);
    digitalWrite(g, LOW);
  }
```

Common Cathode 7-Segment Display: Code

```
else if(digit == 2)
{
  digitalWrite(a, HIGH);
  digitalWrite(b, HIGH);
  digitalWrite(c, LOW);
  digitalWrite(d, HIGH);
  digitalWrite(e, HIGH);
  digitalWrite(f, LOW);
  digitalWrite(g, HIGH);
}
else if(digit == 3)
{
  digitalWrite(a, HIGH);
  digitalWrite(b, HIGH);
  digitalWrite(c, HIGH);
  digitalWrite(d, HIGH);
  digitalWrite(e, LOW);
  digitalWrite(f, LOW);
  digitalWrite(g, HIGH);
}
```

Common Cathode 7-Segment Display: Code

```
else if(digit == 4)
{
  digitalWrite(a, LOW);
  digitalWrite(b, HIGH);
  digitalWrite(c, HIGH);
  digitalWrite(d, LOW);
  digitalWrite(e, LOW);
  digitalWrite(f, HIGH);
  digitalWrite(g, HIGH);
}
else if(digit == 5)
{
  digitalWrite(a, HIGH);
  digitalWrite(b, LOW);
  digitalWrite(c, HIGH);
  digitalWrite(d, HIGH);
  digitalWrite(e, LOW);
  digitalWrite(f, HIGH);
  digitalWrite(g, HIGH);
}
```

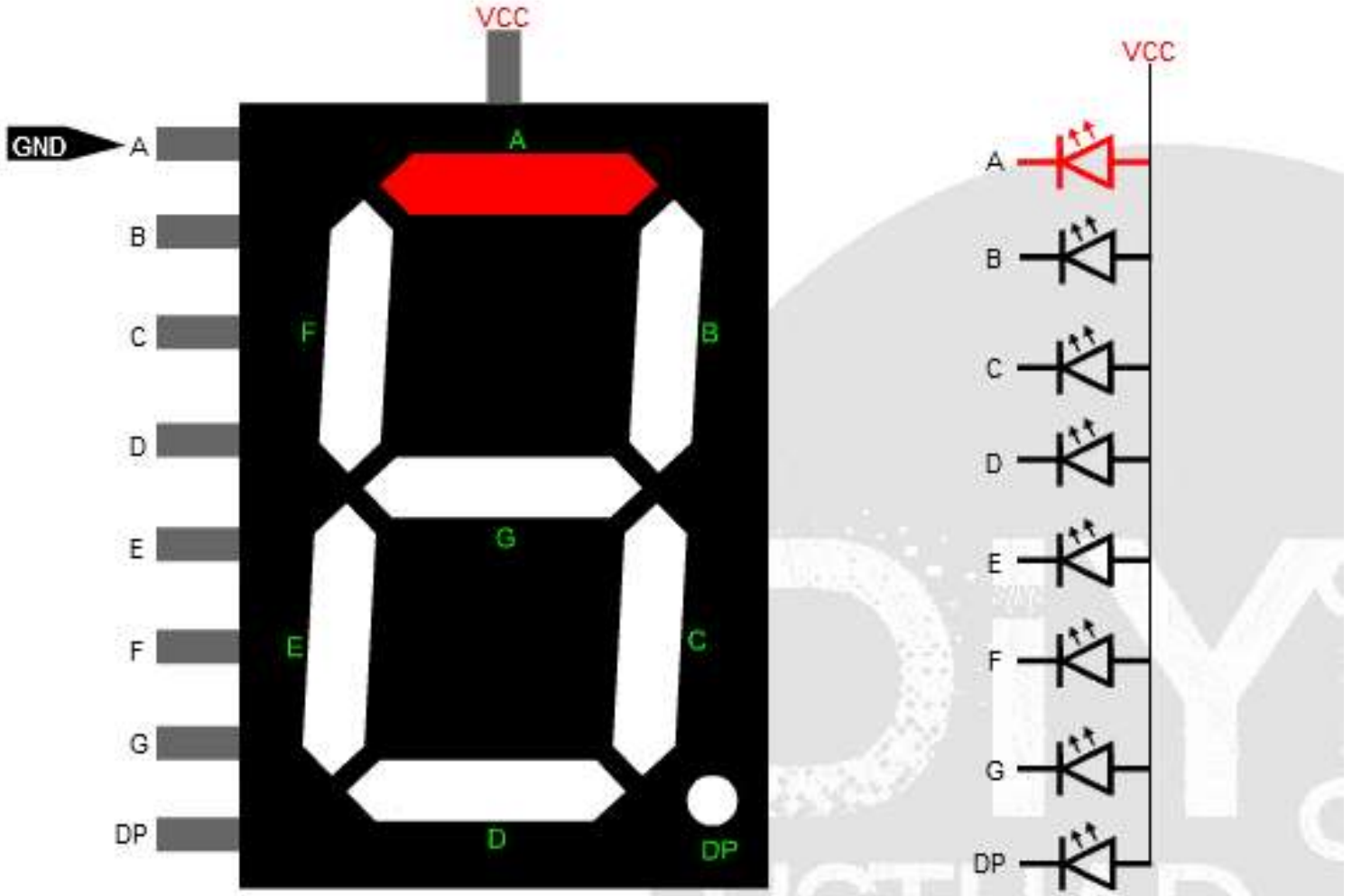
Common Cathode 7-Segment Display: Code

```
else if(digit == 6)
{
  digitalWrite(a, HIGH);
  digitalWrite(b, LOW);
  digitalWrite(c, HIGH);
  digitalWrite(d, HIGH);
  digitalWrite(e, HIGH);
  digitalWrite(f, HIGH);
  digitalWrite(g, HIGH);
}
else if(digit == 7)
{
  digitalWrite(a, HIGH);
  digitalWrite(b, HIGH);
  digitalWrite(c, HIGH);
  digitalWrite(d, LOW);
  digitalWrite(e, LOW);
  digitalWrite(f, LOW);
  digitalWrite(g, LOW);
}
```

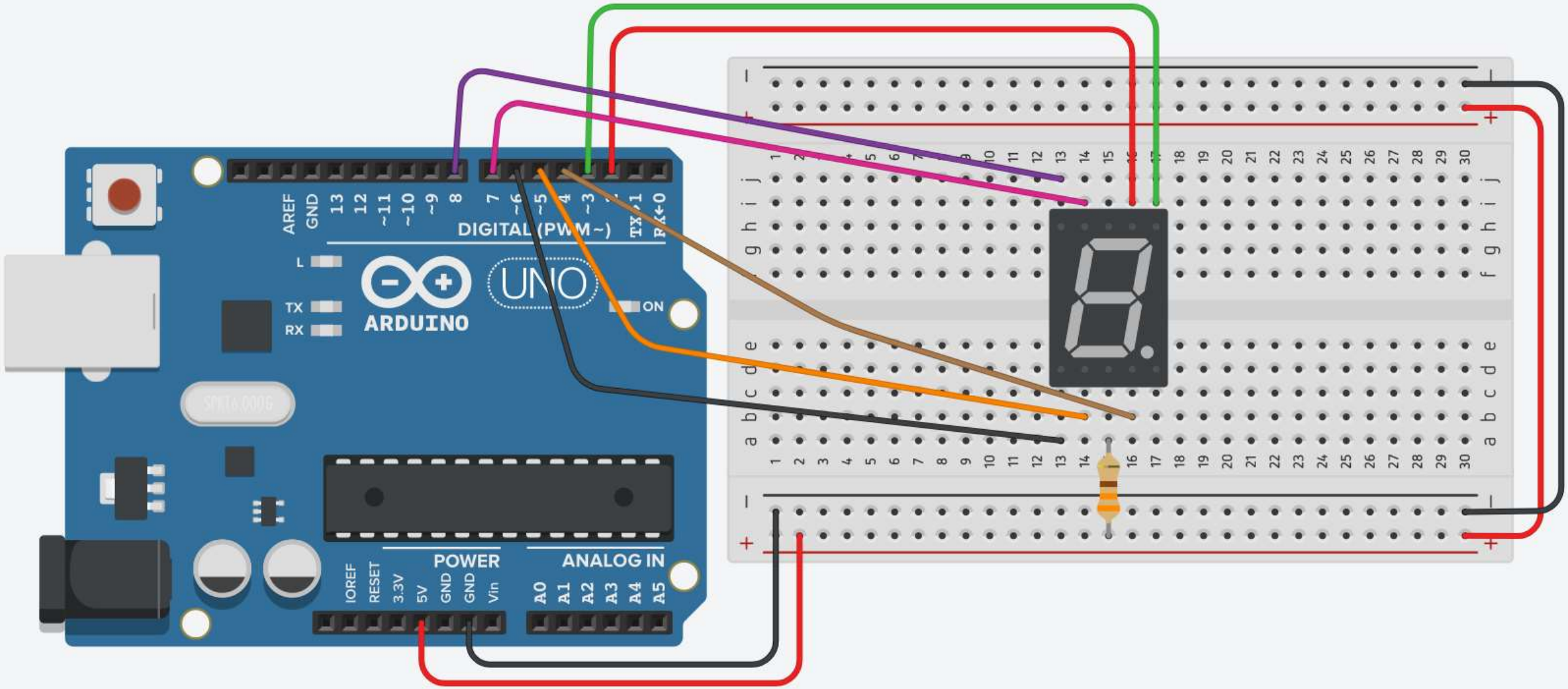
Common Cathode 7-Segment Display: Code

```
  else if(digit == 8)
  {
    digitalWrite(a, HIGH);
    digitalWrite(b, HIGH);
    digitalWrite(c, HIGH);
    digitalWrite(d, HIGH);
    digitalWrite(e, HIGH);
    digitalWrite(f, HIGH);
    digitalWrite(g, HIGH);
  }
  else if(digit == 9)
  {
    digitalWrite(a, HIGH);
    digitalWrite(b, HIGH);
    digitalWrite(c, HIGH);
    digitalWrite(d, HIGH);
    digitalWrite(e, LOW);
    digitalWrite(f, HIGH);
    digitalWrite(g, HIGH);
  }
}
```

Common Anode 7-Segment Display



Common Anode 7-Segment Display: Circuit



Common Anode 7-Segment Display: Components

- Components
  - Arduino
  - Breadboard
  - 7-Segment Display (Common Anode)
  - 330Ω Resistor
  - Jumpers

Common Anode 7-Segment Display: Code

```
// Common anode 7-segment display

#define a 2
#define b 3
#define c 4
#define d 5
#define e 6
#define f 7
#define g 8

void setup() {
  // Define pin modes
  pinMode(a, OUTPUT);
  pinMode(b, OUTPUT);
  pinMode(c, OUTPUT);
  pinMode(d, OUTPUT);
  pinMode(e, OUTPUT);
  pinMode(f, OUTPUT);
  pinMode(g, OUTPUT);

  // Clear the 7-segment display
  reset();
}
```

Common Anode 7-Segment Display: Code

```
void loop() {
  // BCD Counter
  for(int digit = 0; digit <= 9; digit++){
    // Display the digit on the 7-segment display
    display(digit);

    // Wait for a second
    delay(1000);
  }
}

void reset(){
  // Clear the 7-segment display
  digitalWrite(a, HIGH);
  digitalWrite(b, HIGH);
  digitalWrite(c, HIGH);
  digitalWrite(d, HIGH);
  digitalWrite(e, HIGH);
  digitalWrite(f, HIGH);
  digitalWrite(g, HIGH);
}
```

Common Anode 7-Segment Display: Code

```
void display(int digit){
  if(digit == 0)
  {
    digitalWrite(a, LOW);
    digitalWrite(b, LOW);
    digitalWrite(c, LOW);
    digitalWrite(d, LOW);
    digitalWrite(e, LOW);
    digitalWrite(f, LOW);
    digitalWrite(g, HIGH);
  }
  else if(digit == 1)
  {
    digitalWrite(a, HIGH);
    digitalWrite(b, LOW);
    digitalWrite(c, LOW);
    digitalWrite(d, HIGH);
    digitalWrite(e, HIGH);
    digitalWrite(f, HIGH);
    digitalWrite(g, HIGH);
  }
```

Common Anode 7-Segment Display: Code

```
else if(digit == 2)
{
  digitalWrite(a, LOW);
  digitalWrite(b, LOW);
  digitalWrite(c, HIGH);
  digitalWrite(d, LOW);
  digitalWrite(e, LOW);
  digitalWrite(f, HIGH);
  digitalWrite(g, LOW);
}
else if(digit == 3)
{
  digitalWrite(a, LOW);
  digitalWrite(b, LOW);
  digitalWrite(c, LOW);
  digitalWrite(d, LOW);
  digitalWrite(e, HIGH);
  digitalWrite(f, HIGH);
  digitalWrite(g, LOW);
}
```

Common Anode 7-Segment Display: Code

```
else if(digit == 4)
{
  digitalWrite(a, HIGH);
  digitalWrite(b, LOW);
  digitalWrite(c, LOW);
  digitalWrite(d, HIGH);
  digitalWrite(e, HIGH);
  digitalWrite(f, LOW);
  digitalWrite(g, LOW);
}
else if(digit == 5)
{
  digitalWrite(a, LOW);
  digitalWrite(b, HIGH);
  digitalWrite(c, LOW);
  digitalWrite(d, LOW);
  digitalWrite(e, HIGH);
  digitalWrite(f, LOW);
  digitalWrite(g, LOW);
}
```

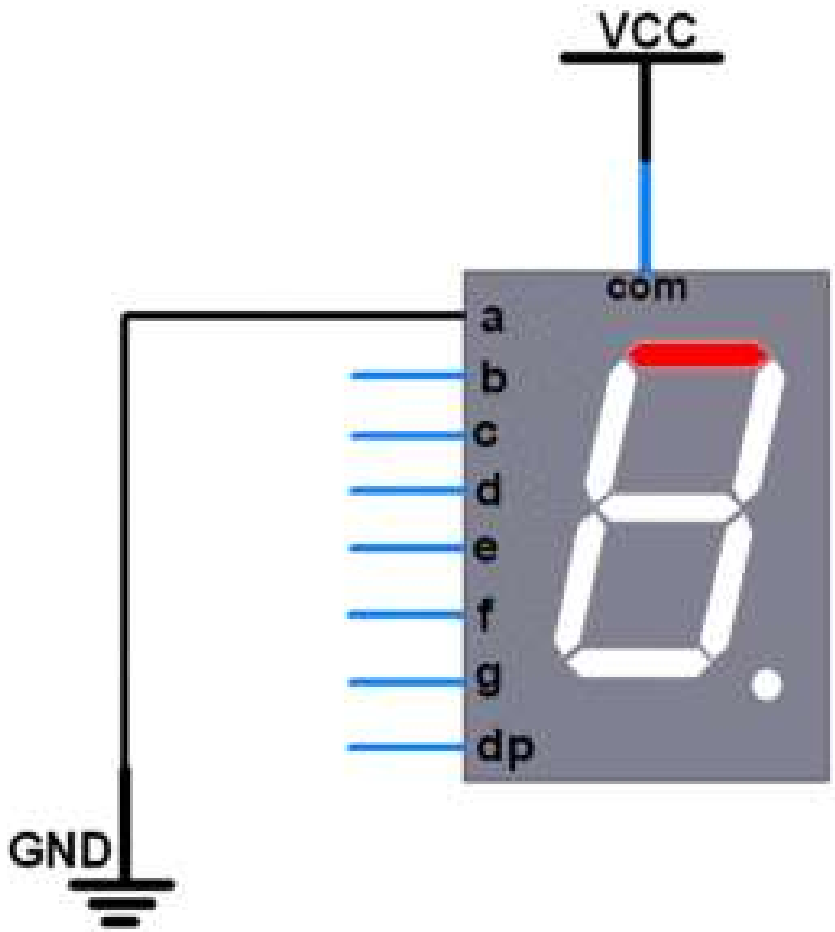
Common Anode 7-Segment Display: Code

```
else if(digit == 6)
{
  digitalWrite(a, LOW);
  digitalWrite(b, HIGH);
  digitalWrite(c, LOW);
  digitalWrite(d, LOW);
  digitalWrite(e, LOW);
  digitalWrite(f, LOW);
  digitalWrite(g, LOW);
}
else if(digit == 7)
{
  digitalWrite(a, LOW);
  digitalWrite(b, LOW);
  digitalWrite(c, LOW);
  digitalWrite(d, HIGH);
  digitalWrite(e, HIGH);
  digitalWrite(f, HIGH);
  digitalWrite(g, HIGH);
}
```

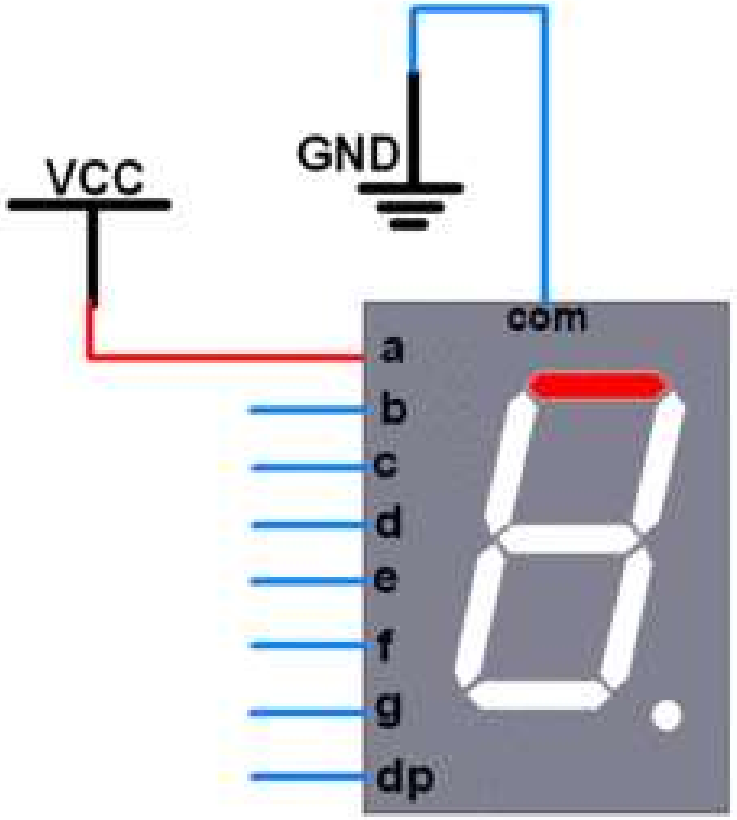
Common Anode 7-Segment Display: Code

```
else if(digit == 8)
  {
    digitalWrite(a, LOW);
    digitalWrite(b, LOW);
    digitalWrite(c, LOW);
    digitalWrite(d, LOW);
    digitalWrite(e, LOW);
    digitalWrite(f, LOW);
    digitalWrite(g, LOW);
  }
  else if(digit == 9)
  {
    digitalWrite(a, LOW);
    digitalWrite(b, LOW);
    digitalWrite(c, LOW);
    digitalWrite(d, LOW);
    digitalWrite(e, HIGH);
    digitalWrite(f, LOW);
    digitalWrite(g, LOW);
  }
}
```

7-Segment Display



Common Anode



Common Cathode

7-Segment Display: Better Code

```
//                a, b, c, d, e, f, g
int pins[7] = {2, 3, 4, 5, 6, 7, 8};

bool common = 0;

//                          a, b, c, d, e, f, g
int segments[10][7] = {{1, 1, 1, 1, 1, 1, 0},     // 0
                       {0, 1, 1, 0, 0, 0, 0},     // 1
                       {1, 1, 0, 1, 1, 0, 1},     // 2
                       {1, 1, 1, 1, 0, 0, 1},     // 3
                       {0, 1, 1, 0, 0, 1, 1},     // 4
                       {1, 0, 1, 1, 0, 1, 1},     // 5
                       {1, 0, 1, 1, 1, 1, 1},     // 6
                       {1, 1, 1, 0, 0, 0, 0},     // 7
                       {1, 1, 1, 1, 1, 1, 1},     // 8
                       {1, 1, 1, 0, 0, 1, 1}};    // 9
```

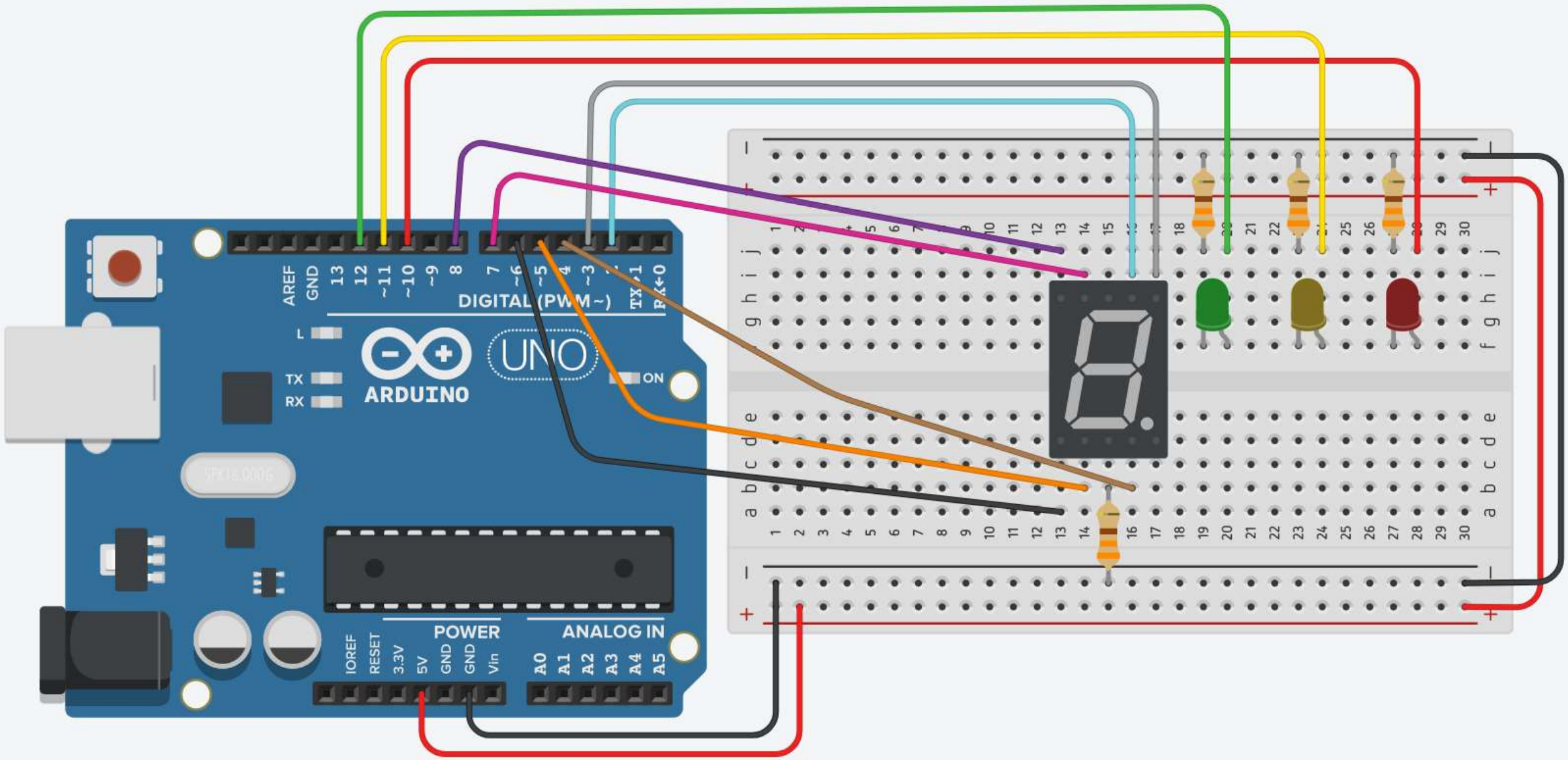
7-Segment Display: Better Code

```
void setup() {
  // Define pin modes
  for(int i = 0; i < 7; i++)
    pinMode(pins[i], OUTPUT);

  // Clear the 7-segment display
  reset();
}

void loop() {
  // BCD Counter
  for(int digit = 0; digit <= 9; digit++){
    // Display the digit on the 7-segment display
    display(digit);

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

7-Segment Display: Better Code

```
void reset(){
  // Clear the 7-segment display
  for(int i = 0; i < 7; i++)
    digitalWrite(pins[i], common);
}

void display(int digit){
  for(int i = 0; i < 7; i++)
  {
    if(common == 0)
      digitalWrite(pins[i], segments[digit][i]);
    else
       digitalWrite(pins[i], !segments[digit][i]);
  }
}
```

Integrating LEDs with 7-Segment Display



Integrating LEDs with 7-Segment Display

```
#define RED 10                          // The Red LED connects to digital pin 10
#define YELLOW 11                       // The Yellow LED connects to digital pin 11
#define GREEN 12                        // The Green LED connects to digital pin 12

//              a, b, c, d, e, f, g
int pins[7] = {2, 3, 4, 5, 6, 7, 8};

bool common = 0;

//                        a, b, c, d, e, f, g
int segments[10][7] = {{1, 1, 1, 1, 1, 1, 0},    // 0
                       {0, 1, 1, 0, 0, 0, 0},    // 1
                       {1, 1, 0, 1, 1, 0, 1},    // 2
                       {1, 1, 1, 1, 0, 0, 1},    // 3
                       {0, 1, 1, 0, 0, 1, 1},    // 4
                       {1, 0, 1, 1, 0, 1, 1},    // 5
                       {1, 0, 1, 1, 1, 1, 1},    // 6
                       {1, 1, 1, 0, 0, 0, 0},    // 7
                       {1, 1, 1, 1, 1, 1, 1},    // 8
                       {1, 1, 1, 0, 0, 1, 1}};   // 9
```

Integrating LEDs with 7-Segment Display

```
void setup() {
  pinMode(RED, OUTPUT);         // Declare the digital pin 10 as output
  pinMode(YELLOW, OUTPUT);      // Declare the digital pin 11 as output
  pinMode(GREEN, OUTPUT);       // Declare the digital pin 12 as output

  // Define the 7-segment display pins as outputs
  for(int i = 0; i < 7; i++)
    pinMode(pins[i], OUTPUT);

  // Clear the 7-segment display
  reset();
}
```

Integrating LEDs with 7-Segment Display

```
void loop() {
  digitalWrite(GREEN, HIGH);          // Turn the Green LED on
  digitalWrite(YELLOW, LOW);          // Turn the Yellow LED off
  digitalWrite(RED, LOW);             // Turn the Red LED off

  // Wait 5 seconds
  for(int i = 1; i <= 5; i++){
    display(i);
    delay(1000);
  }

  digitalWrite(GREEN, LOW);           // Turn the Green LED off
  digitalWrite(YELLOW, HIGH);         // Turn the Yellow LED on
  digitalWrite(RED, LOW);             // Turn the Red LED

  // Wait 2 seconds
  for(int i = 1; i <= 2; i++){
    display(i);
    delay(1000);
  }

  digitalWrite(GREEN, LOW);           // Turn the Green LED off
  digitalWrite(YELLOW, LOW);          // Turn the Yellow LED off
  digitalWrite(RED, HIGH);            // Turn the Red LED on

  // Wait 5 seconds
  for(int i = 1; i <= 5; i++){
    display(i);
    delay(1000);
  }
}
```

Integrating LEDs with 7-Segment Display

```
void reset(){
  // Clear the 7-segment display
  for(int i = 0; i < 7; i++)
    digitalWrite(pins[i], common);
}

void display(int digit){
  for(int i = 0; i < 7; i++)
  {
    if(common == 0)
      digitalWrite(pins[i], segments[digit][i]);
    else
       digitalWrite(pins[i], !segments[digit][i]);
  }
}
```