

*The beauty and elegance of
programmable objects.*



What could be programmable objects?

What actually means programming?

What do objects need to get programmed?



Photo von Jon Fingas



Photo von Kārlis Dambrāns

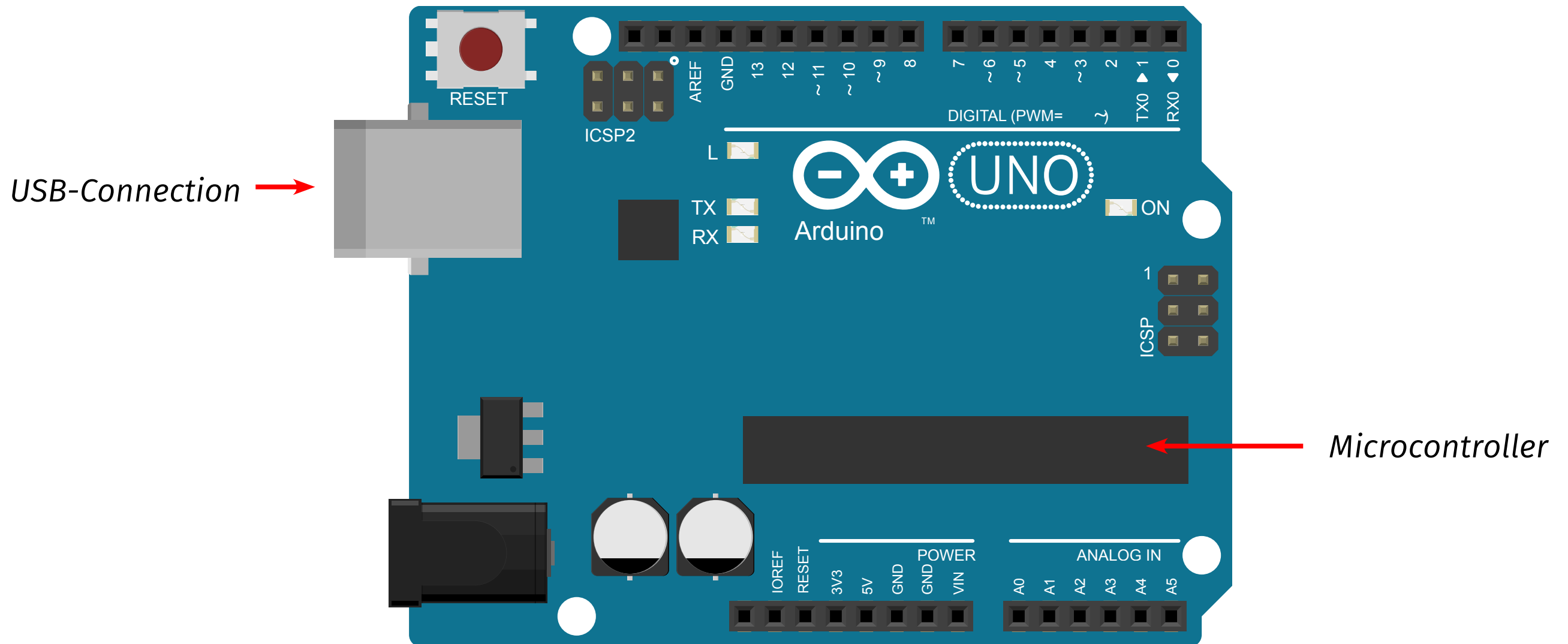




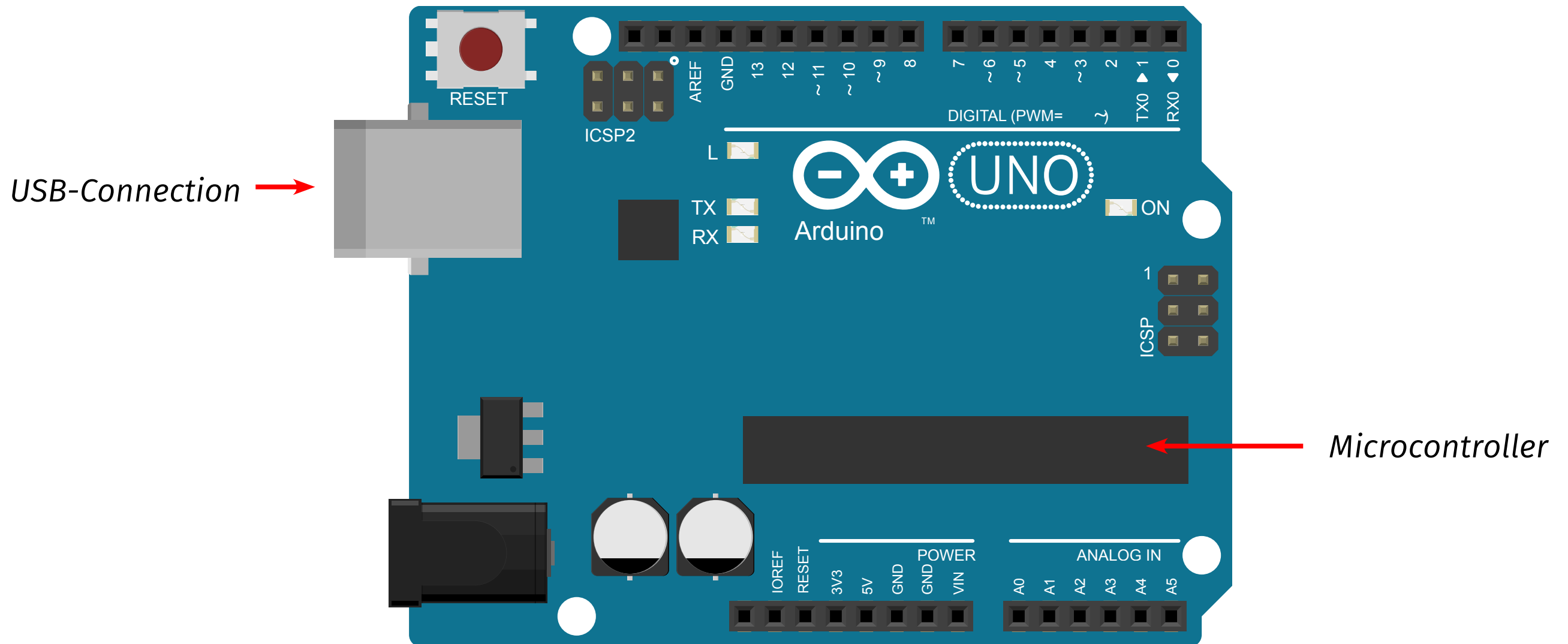
Photo von Cary Bass-Deschenes

CAUTION!
ZOMBIES!
AHEAD!!!

We are working with the Arduino!



A lot of different components can be attached and controlled by the Arduino

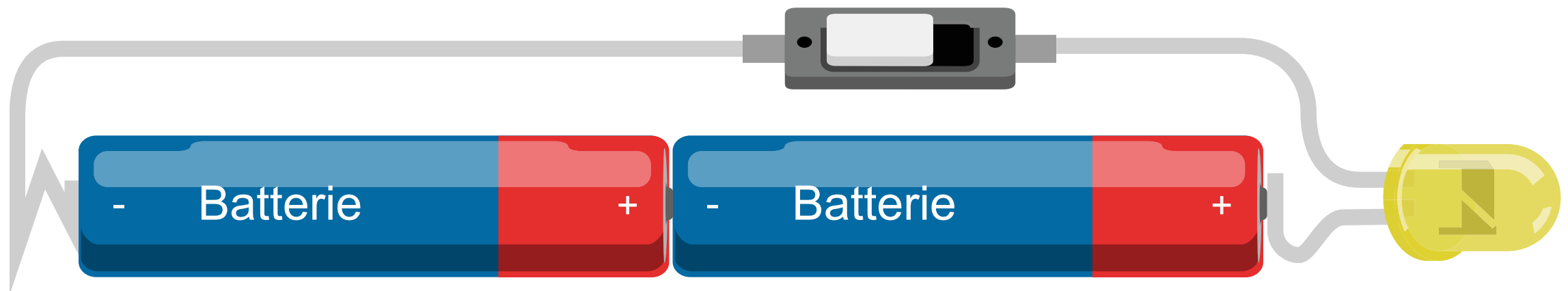


What is electronics?

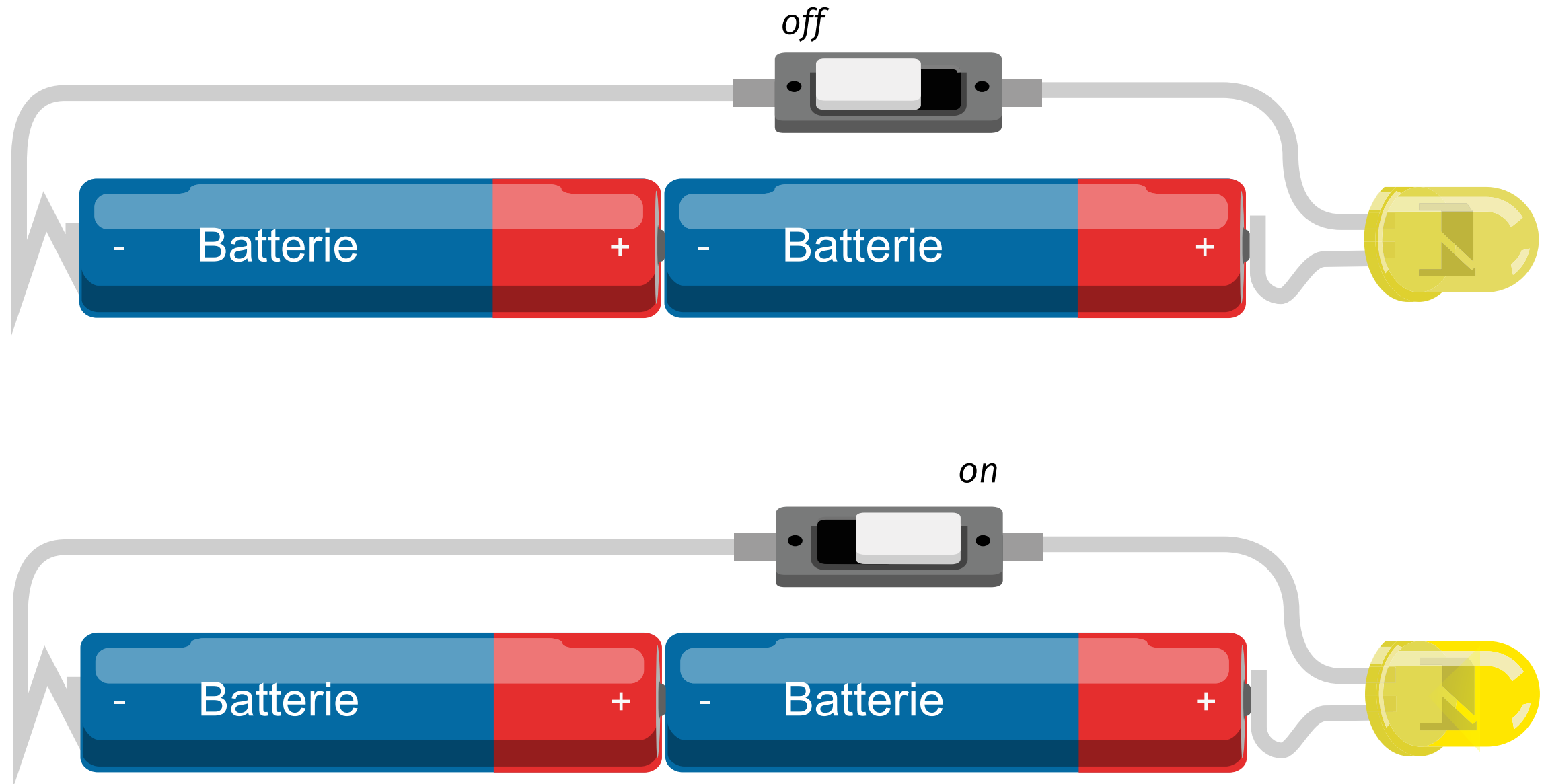
What is current?

What does a flash light consists of?

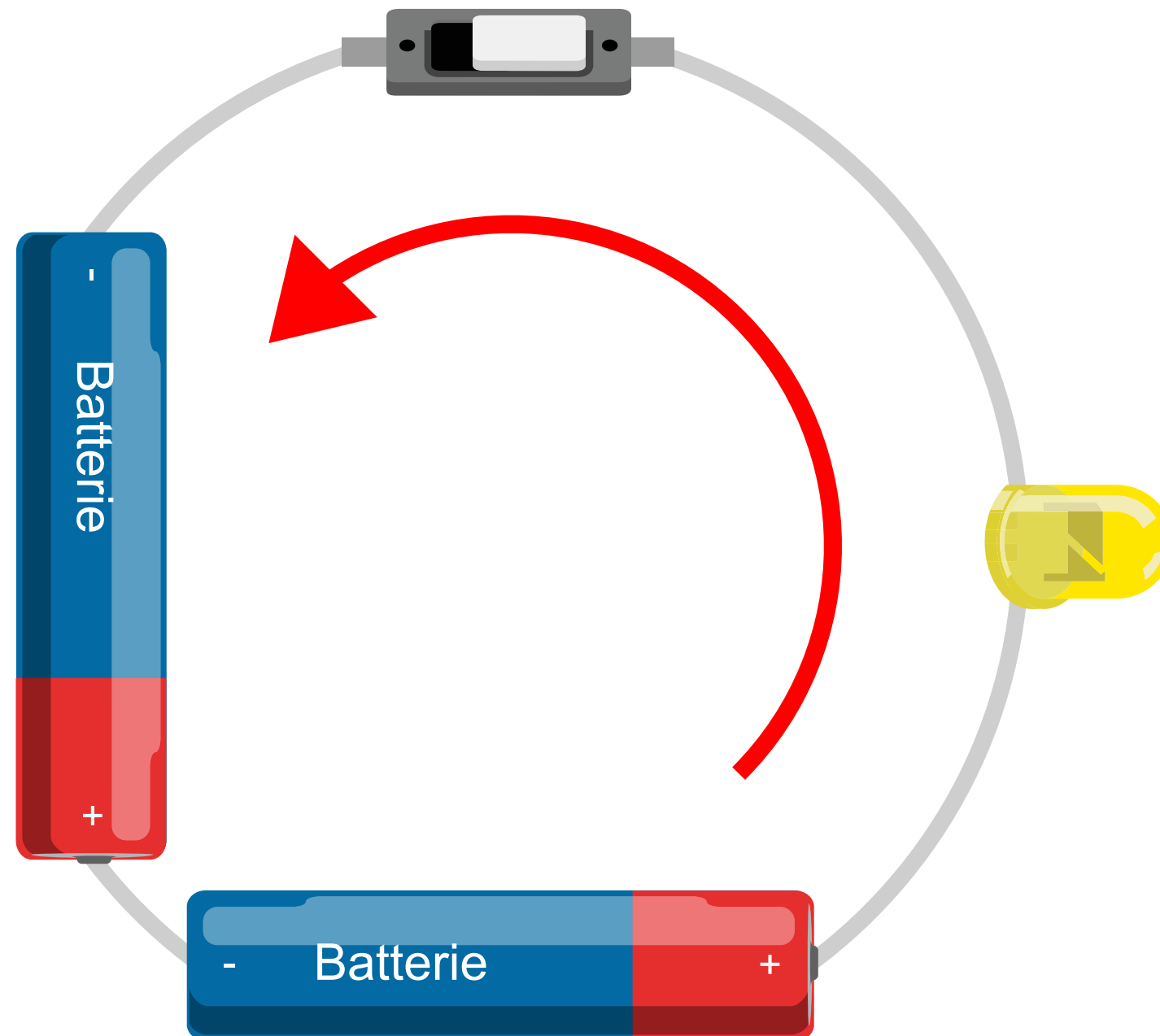
What does a flash light consists of?



Why does a flash light lights up when it is switched on?



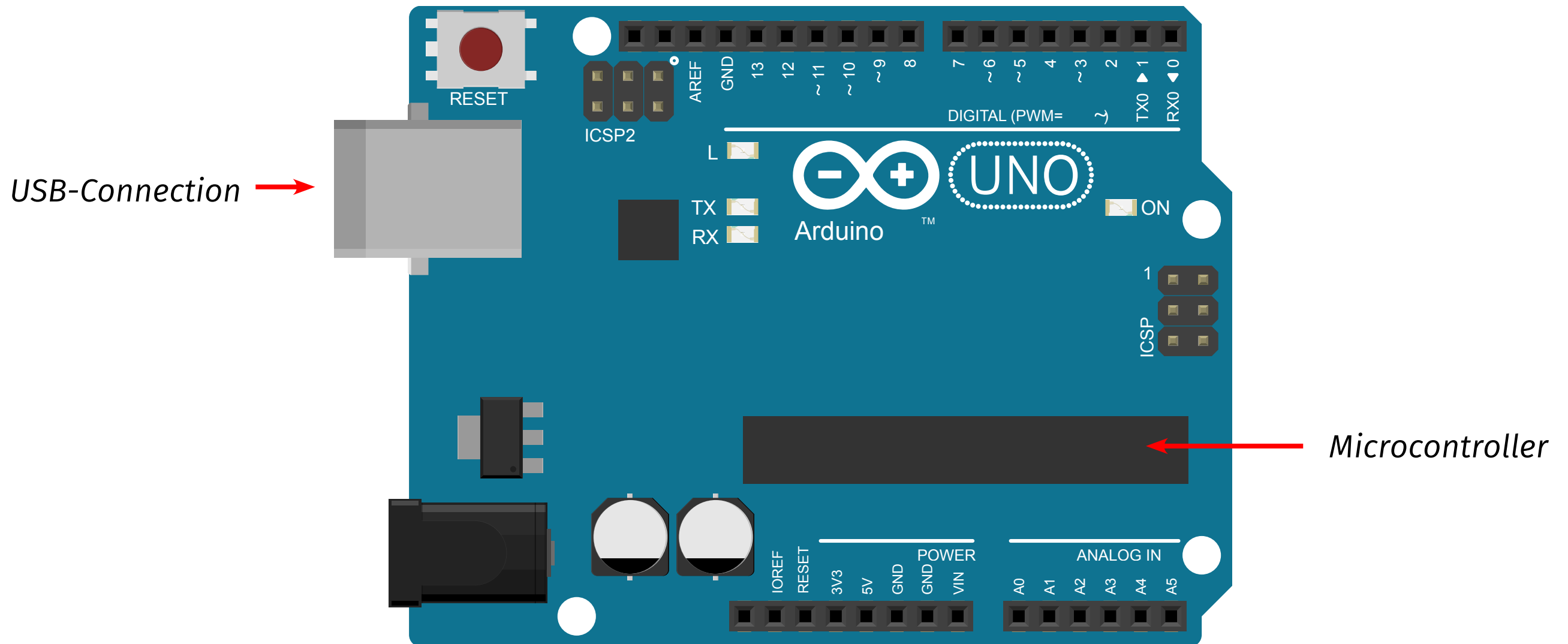
Current always runs from plus to minus!



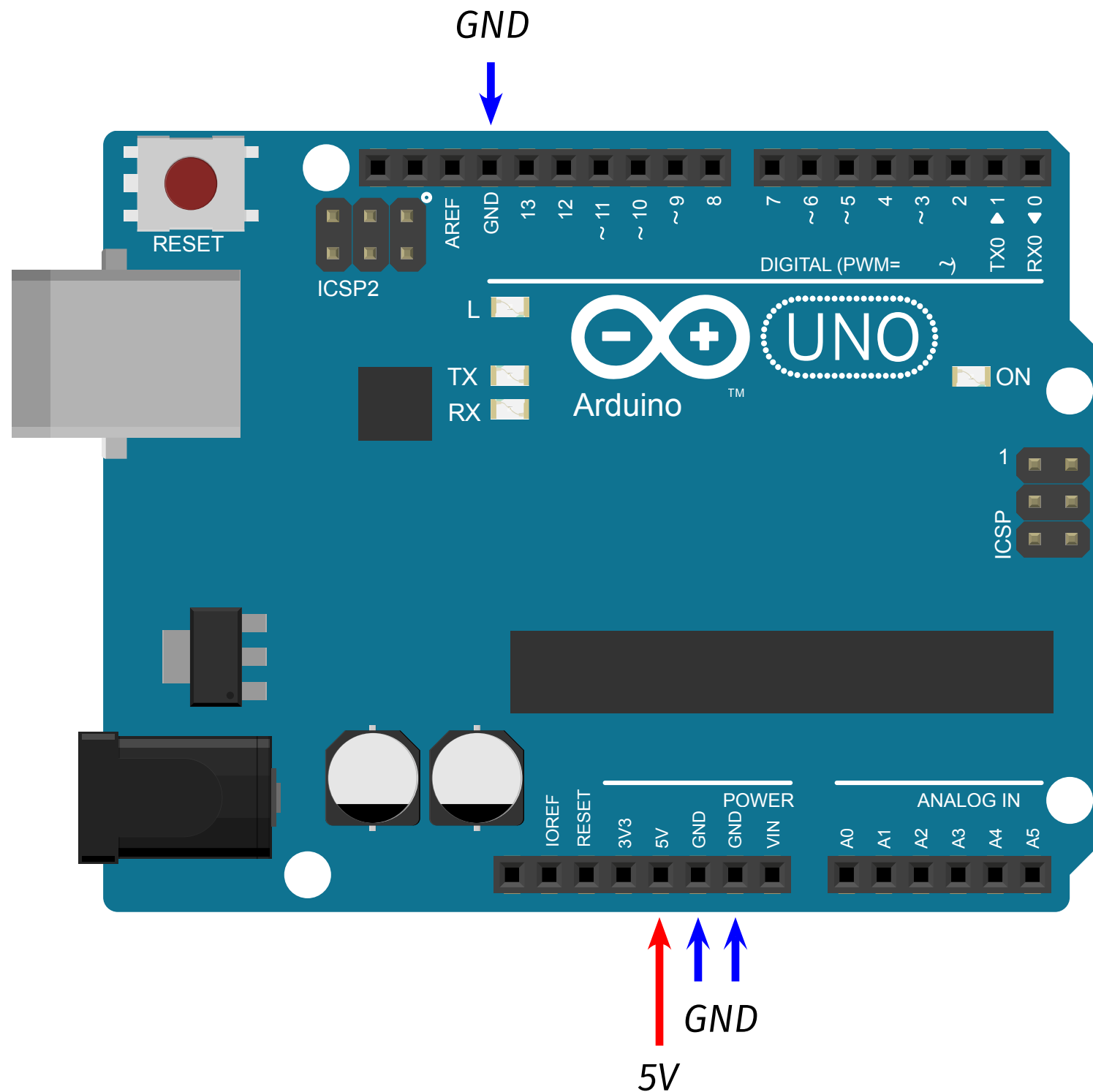
What happens, if you put the battery in the wrong direction?



Can you find the plus and minus on the Arduino?

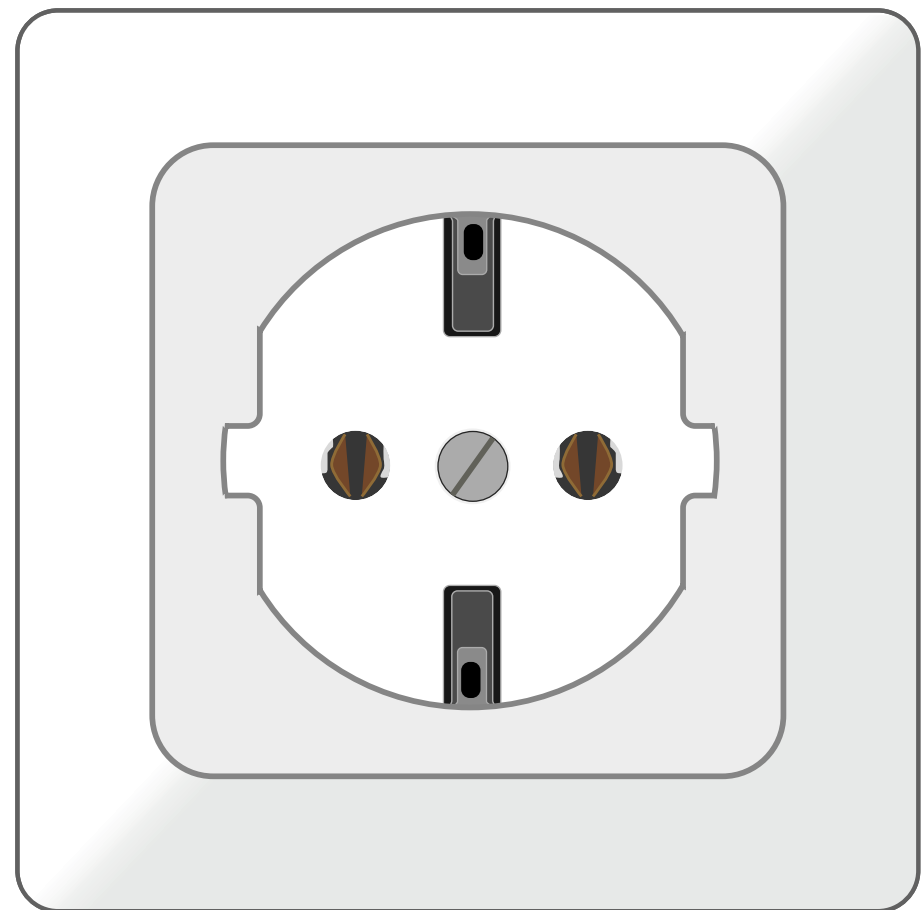
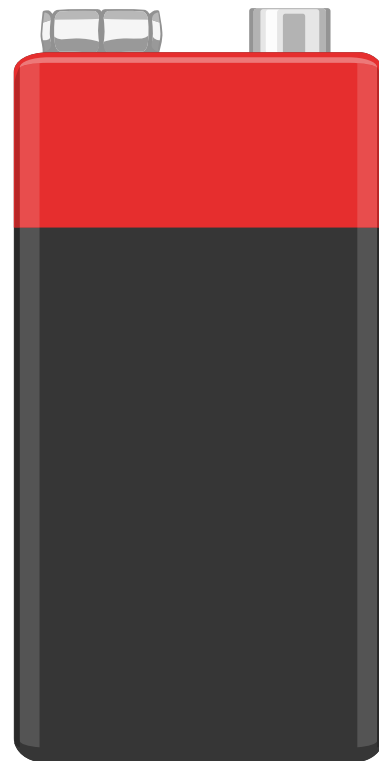


What could 5V mean?



- 5V = Plus
- GND = Minus

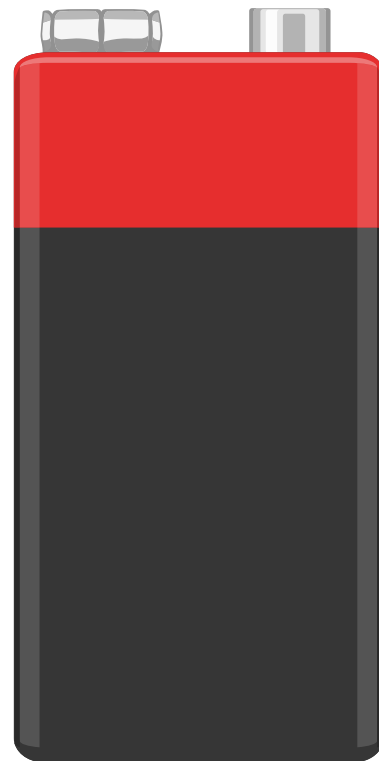
How much voltage is in here?



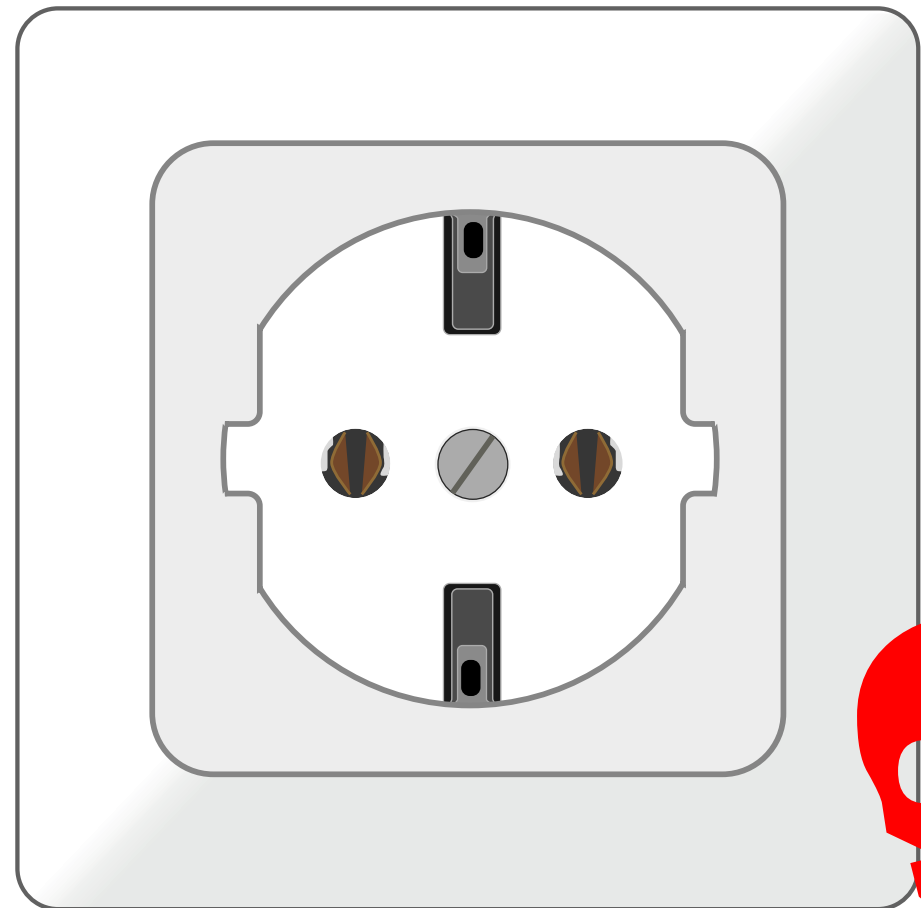
How much voltage is in here?



1,5 Volt



9 Volt



230 Volt



We build a flash light

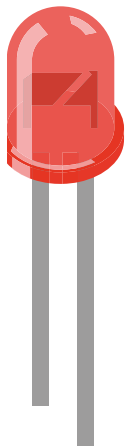
Parts



Resistor, 220 Ohm
(red-red-brown-gold)



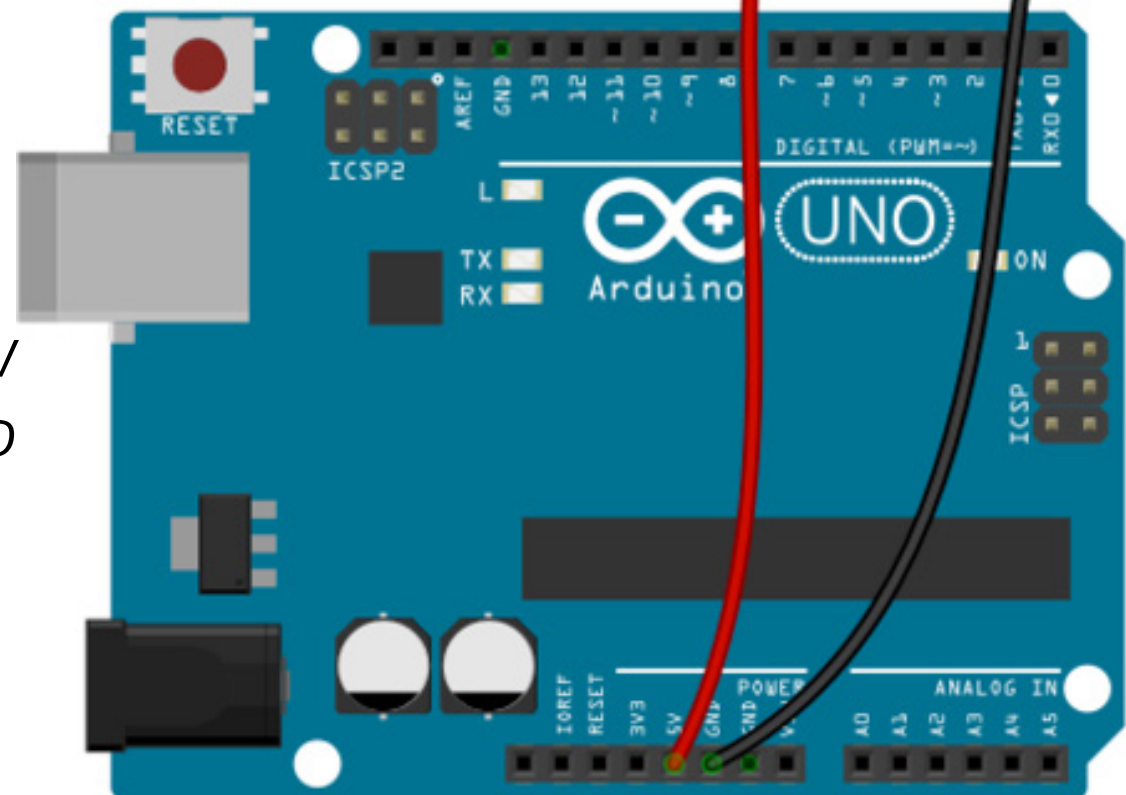
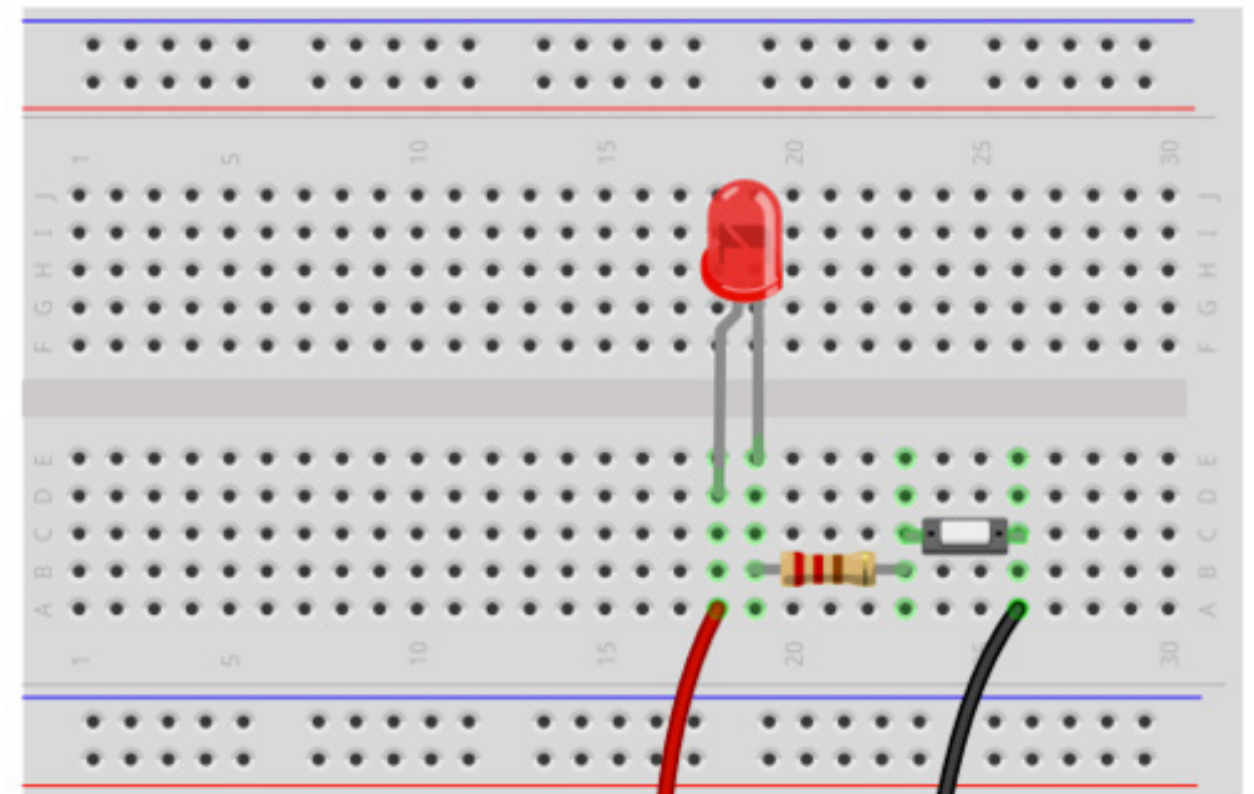
Pushbutton



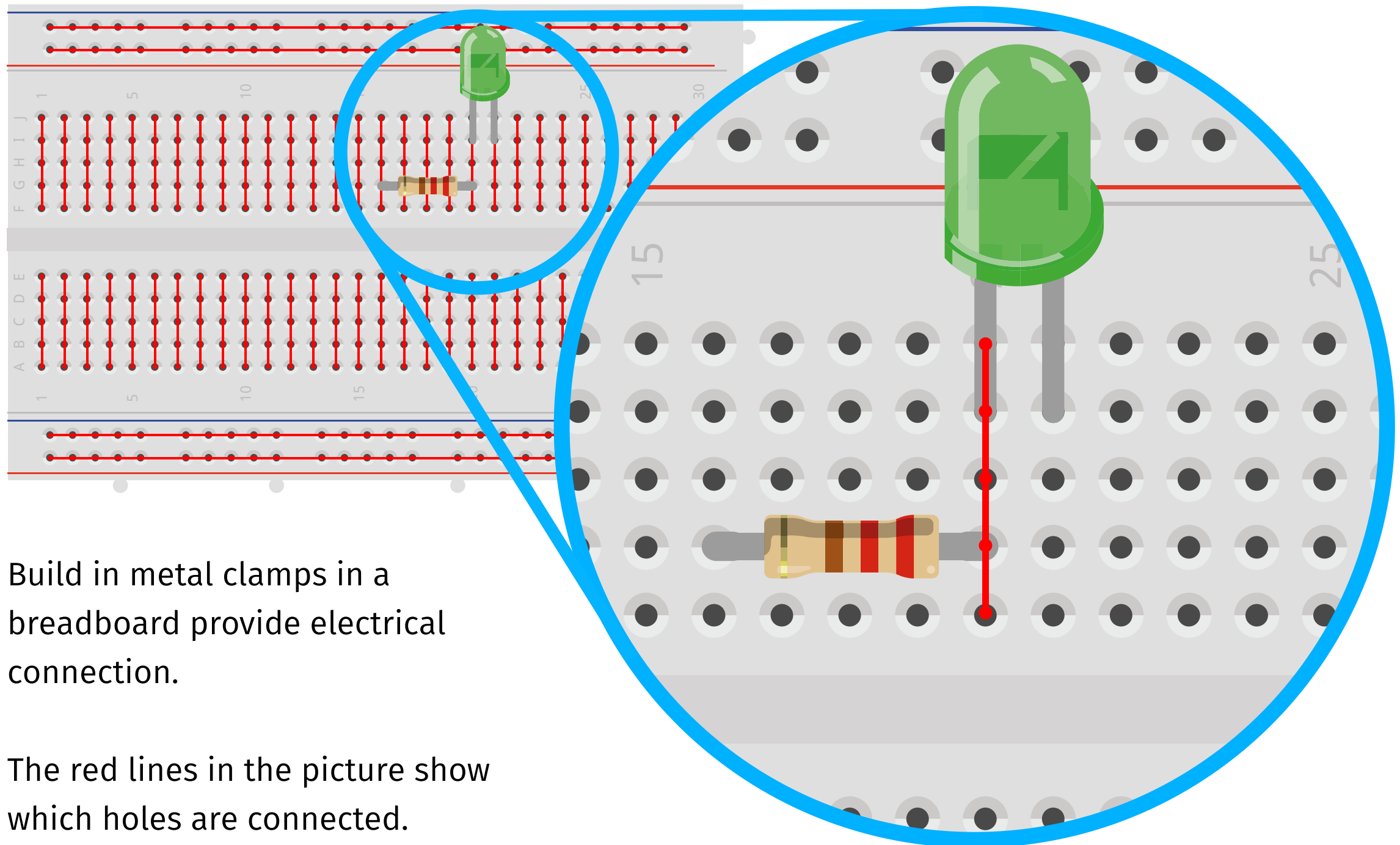
LED
(short leg to the GND)

red wire to 5V
black wire to GND

Circuit of a flash light



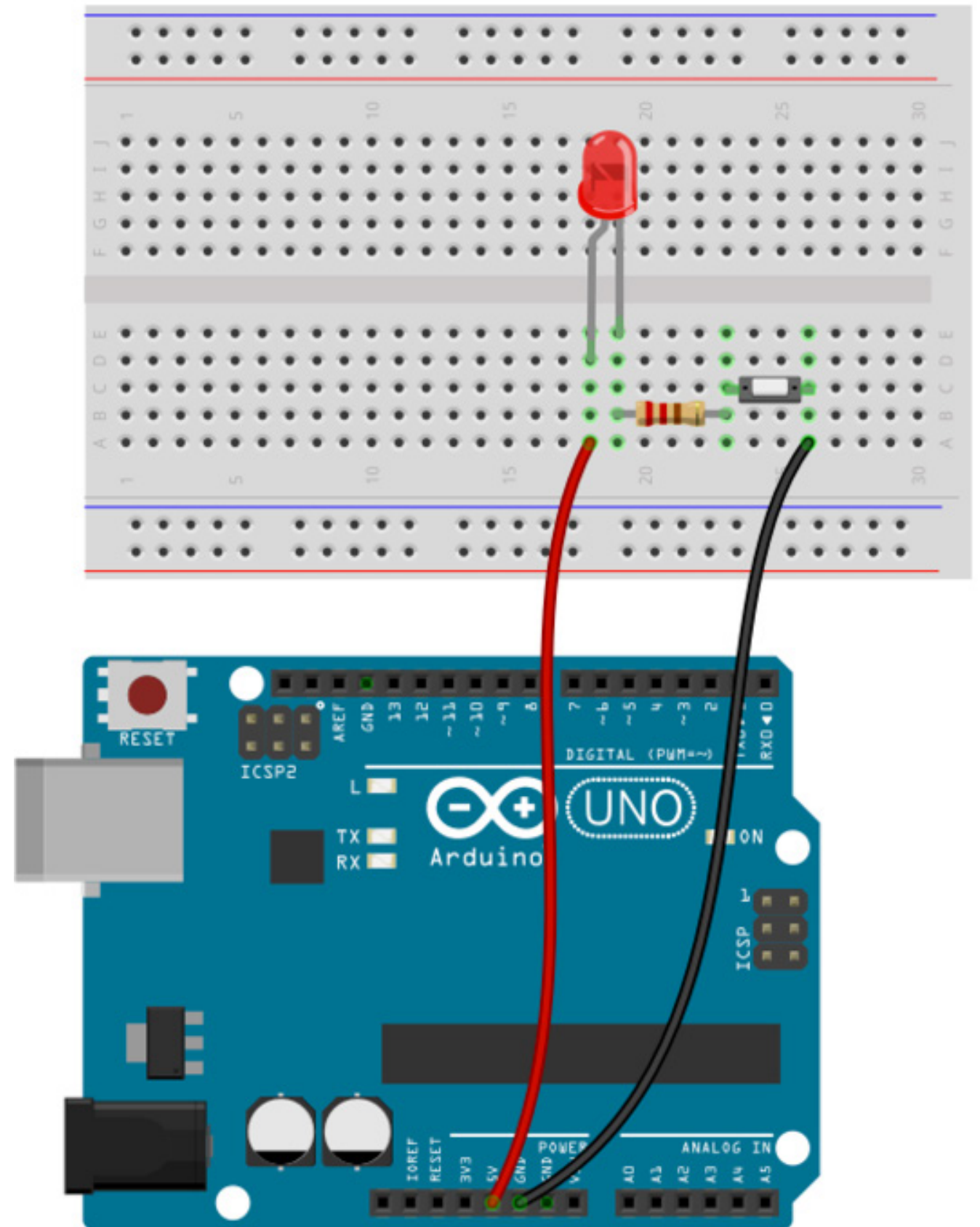
Inner connection of the holes of a breadboard



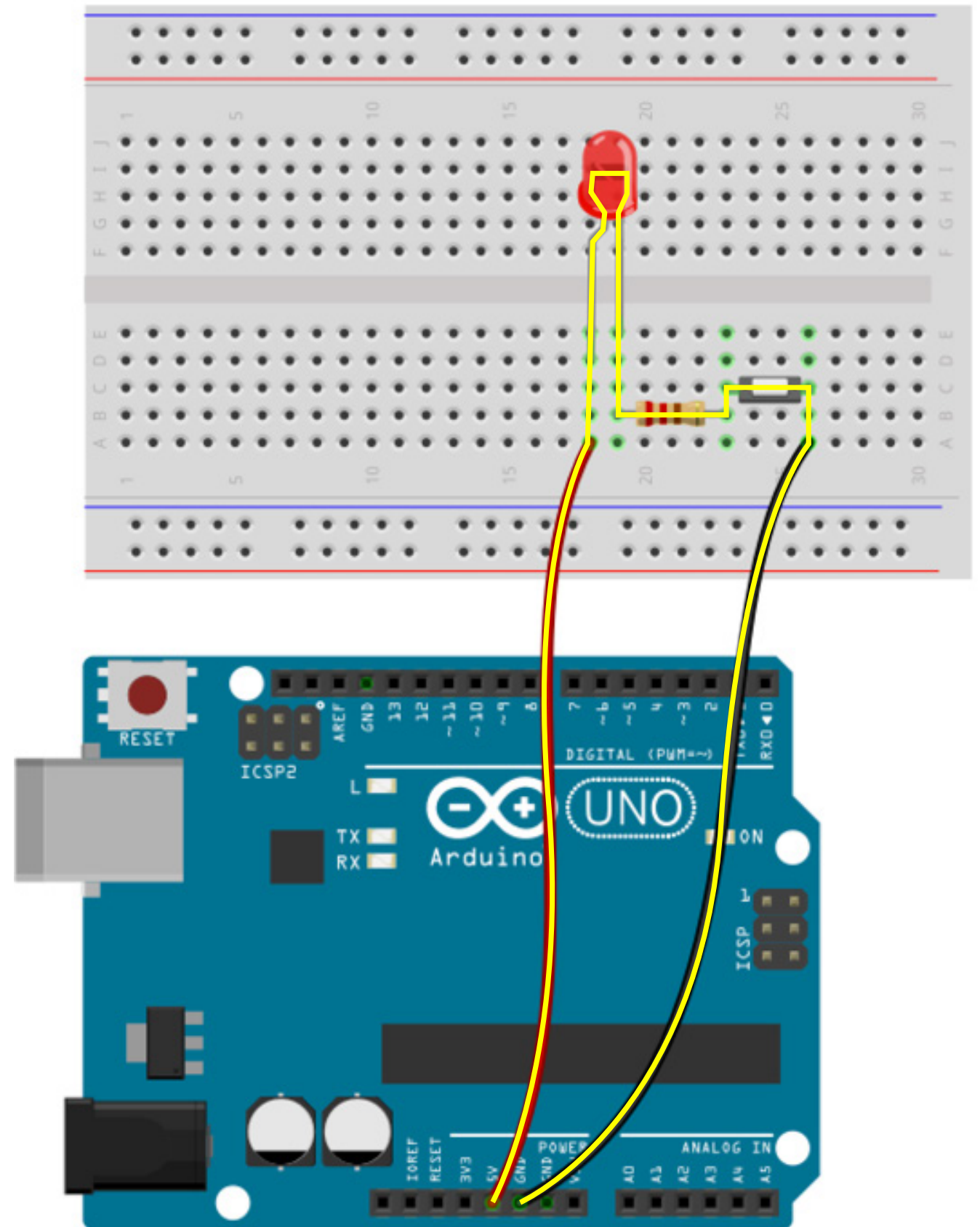
Build in metal clamps in a breadboard provide electrical connection.

The red lines in the picture show which holes are connected.

Where does the current flows?



Where does the current flows?

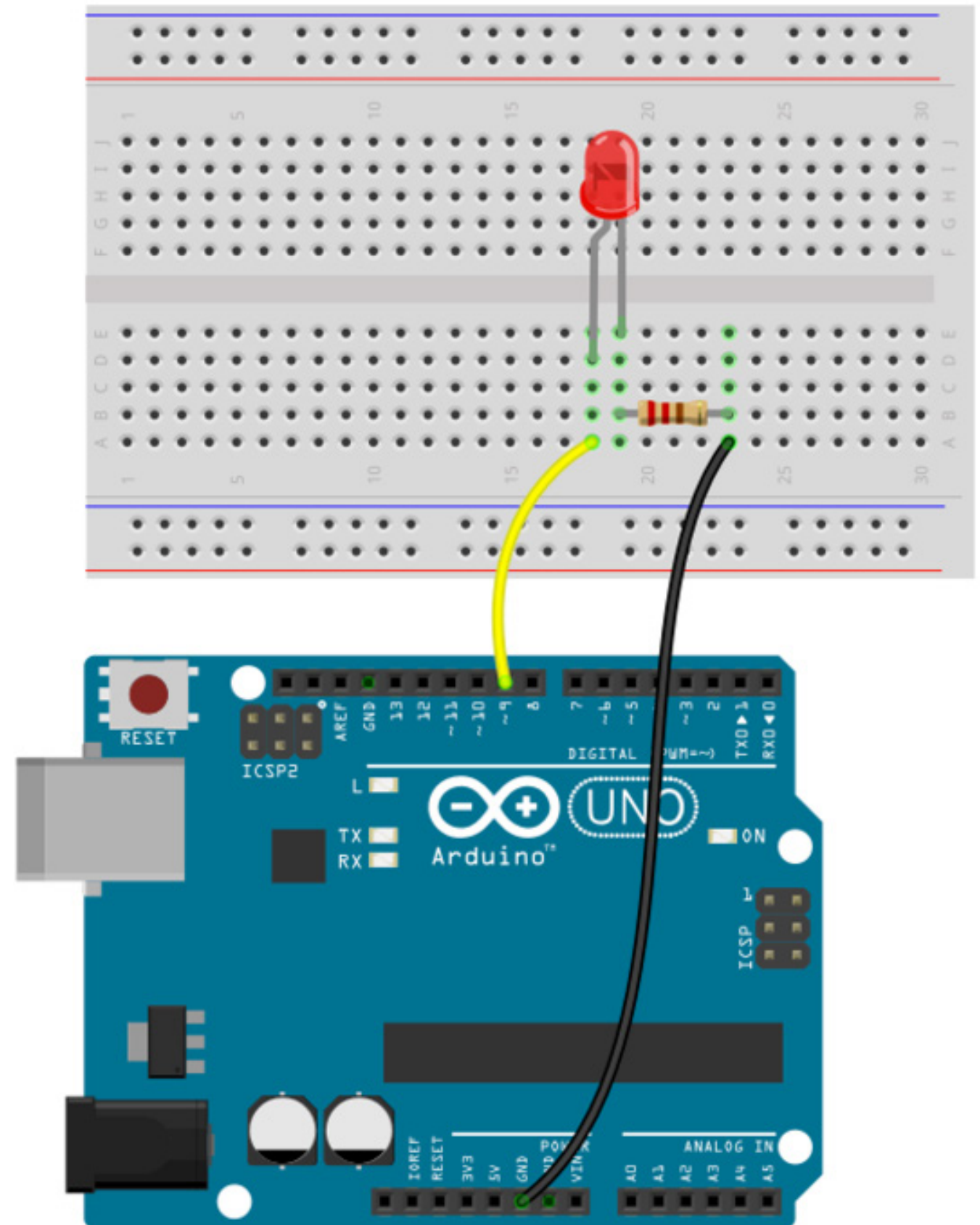


Now we wanna blink an LED

Build the circuit

What are the steps blinking a LED?

Circuit

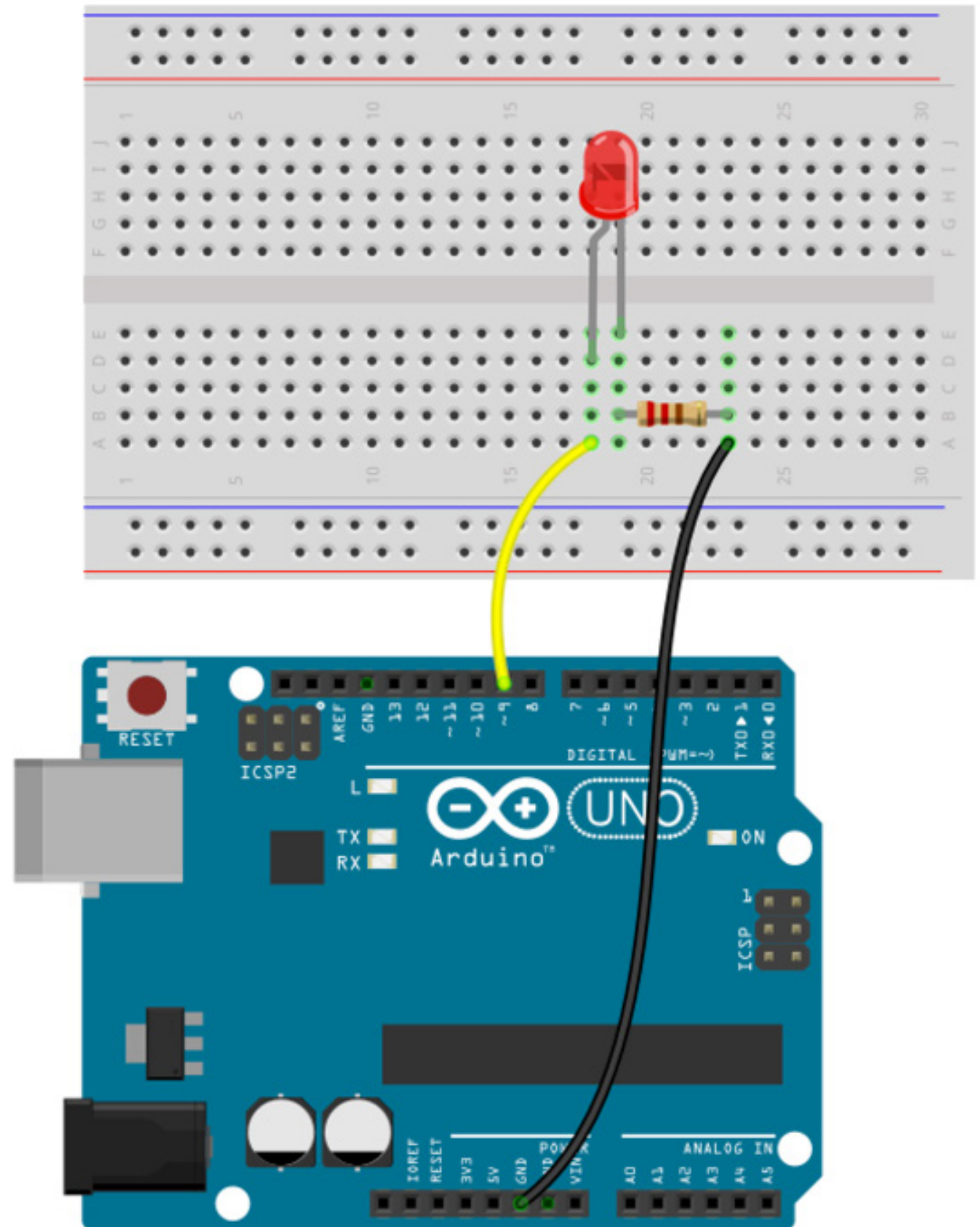


Code

```
void setup(){  
  pinMode(9,OUTPUT);  
}  
  
void loop(){  
  digitalWrite(9,HIGH);  
}
```

(The Arduino-Software needs to be installed. You can get it from www.arduino.cc.)

Circuit



Code



```
void setup(){
  pinMode(9,OUTPUT);
}

void loop(){
  digitalWrite(9,HIGH);
}
```

Caution! The program needs to be uploaded to the Arduino-Board. To do so you have to select the right communication port in the Arduino-Software. You can find the selection in the Tools menu:

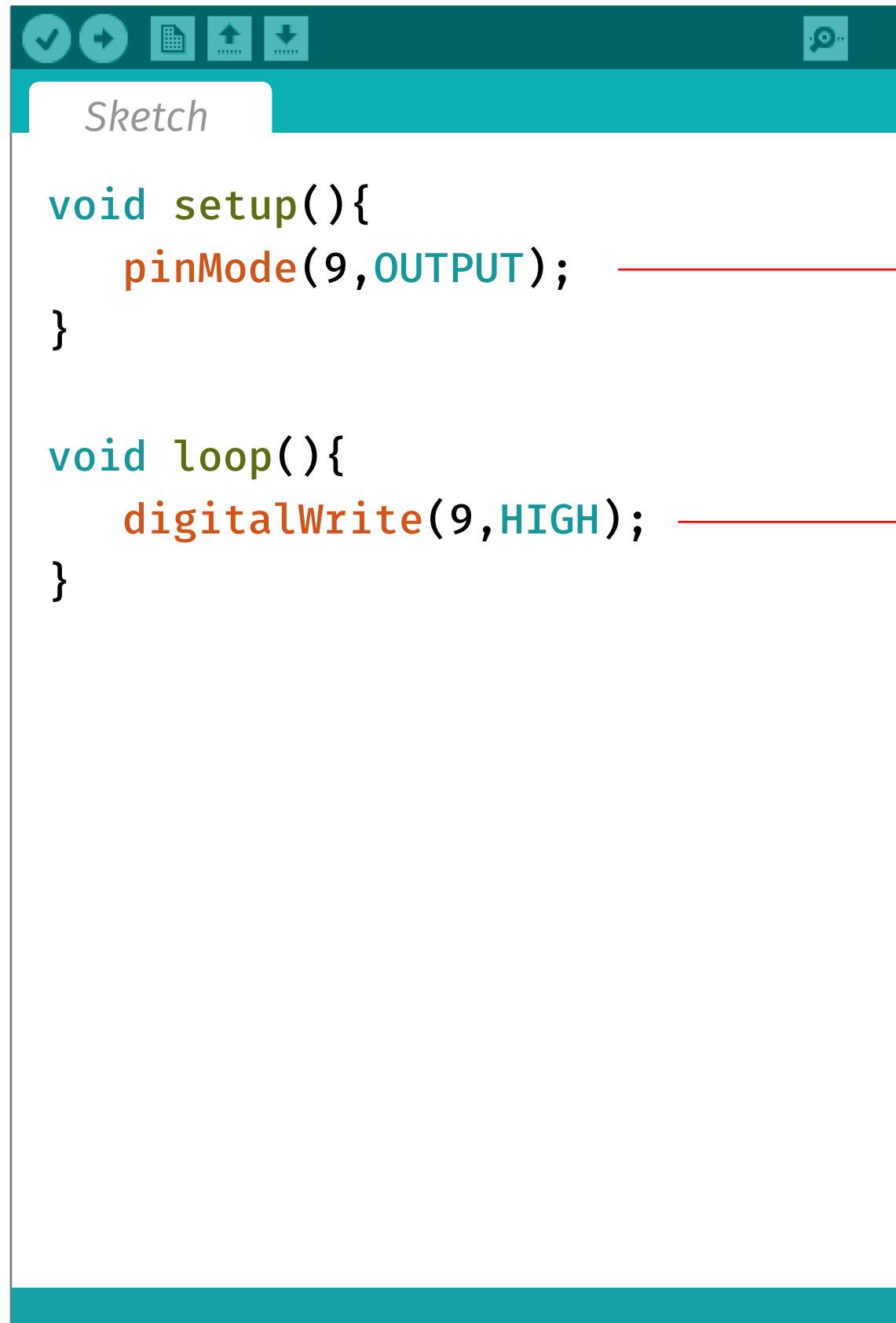
Tools > Port > xxxxx (Arduino UNO)

Upload the program by clicking

File > Upload

or use the Icon with the arrow right in the symbols bar.

Code



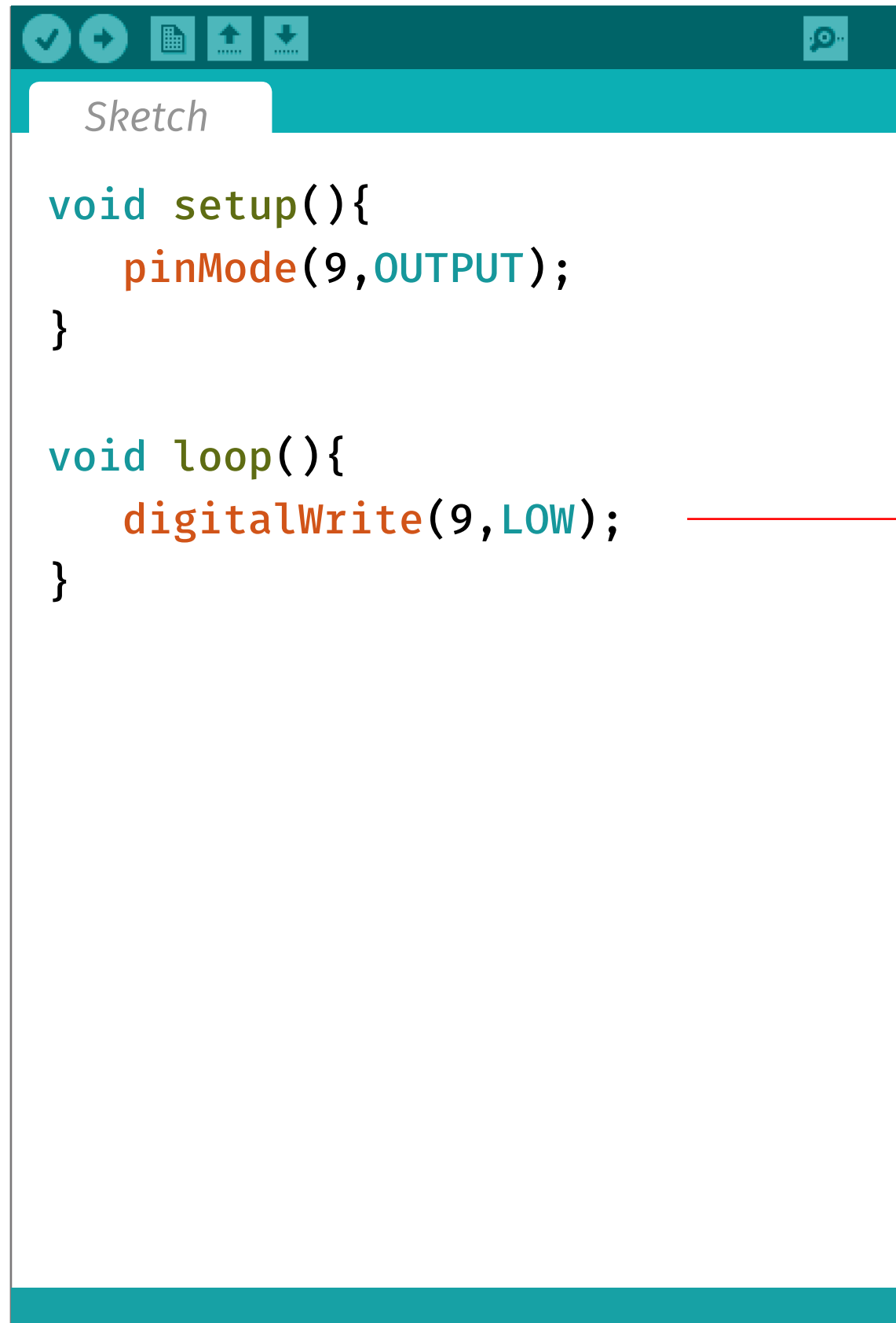
```
void setup(){  
  pinMode(9,OUTPUT);  
}  
  
void loop(){  
  digitalWrite(9,HIGH);  
}
```

This command tells the Arduino to use pin 9 as an Output so that we can switch it on or off.

This command switches pin 9 on (HIGH).

Do you have an idea, how to switch it off?

Code

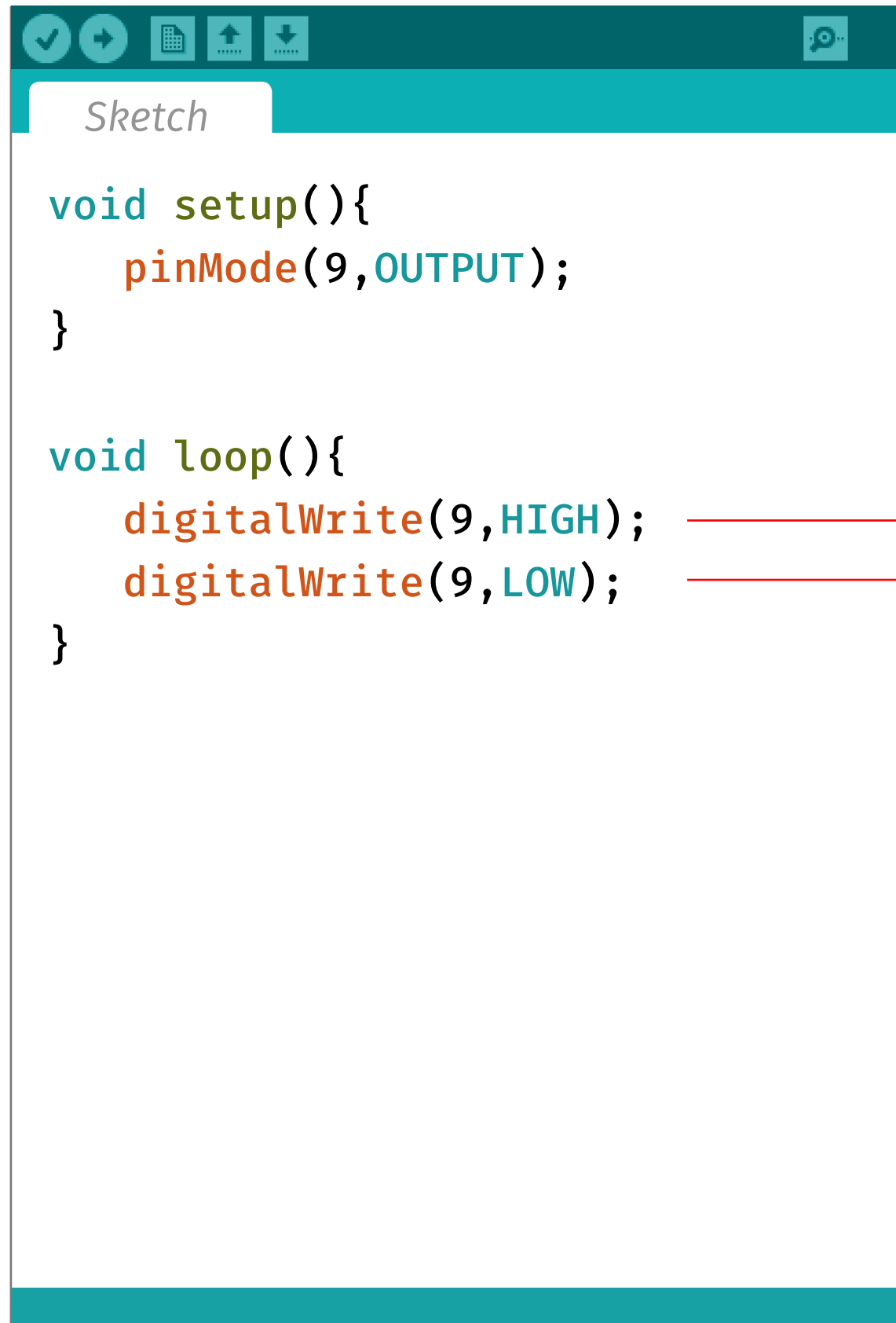


```
void setup(){  
  pinMode(9,OUTPUT);  
}  
  
void loop(){  
  digitalWrite(9,LOW);  
}
```

This command switches pin 9 off (LOW).

Now, make the LED blinking!

Code



The image shows a screenshot of the Arduino IDE interface. At the top, there is a toolbar with icons for checking, running, saving, and uploading. Below the toolbar is a tab labeled "Sketch". The main area contains the following code:

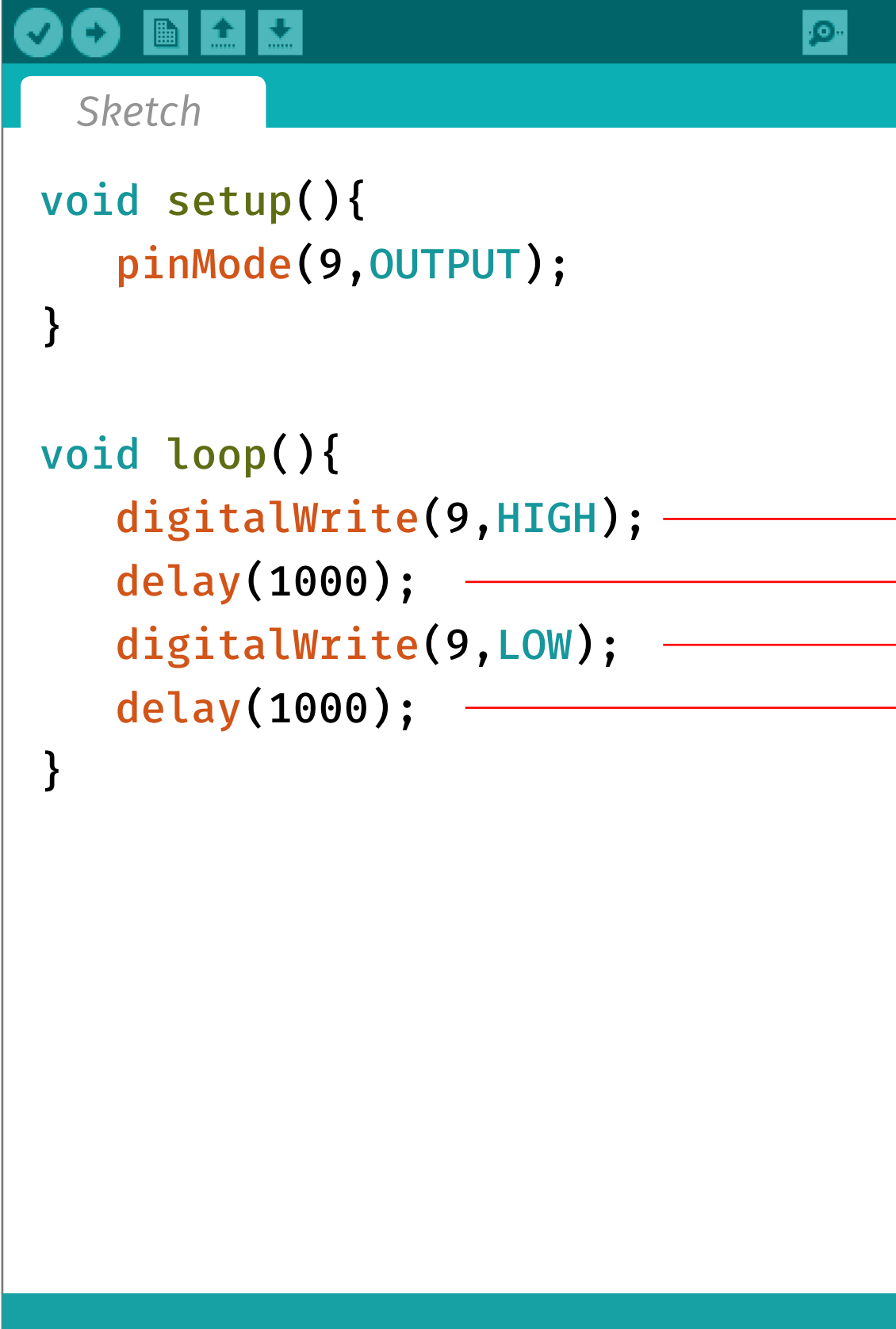
```
void setup(){  
  pinMode(9,OUTPUT);  
}  
  
void loop(){  
  digitalWrite(9,HIGH);  
  digitalWrite(9,LOW);  
}
```

switches LED on

switches LED off

Does the LED blinks now?

Code



```
void setup(){
  pinMode(9,OUTPUT);
}

void loop(){
  digitalWrite(9,HIGH);
  delay(1000);
  digitalWrite(9,LOW);
  delay(1000);
}
```

The Arduino-Board works so fast — can't see the switching. We need a break in between to see the change.

switching the LED on

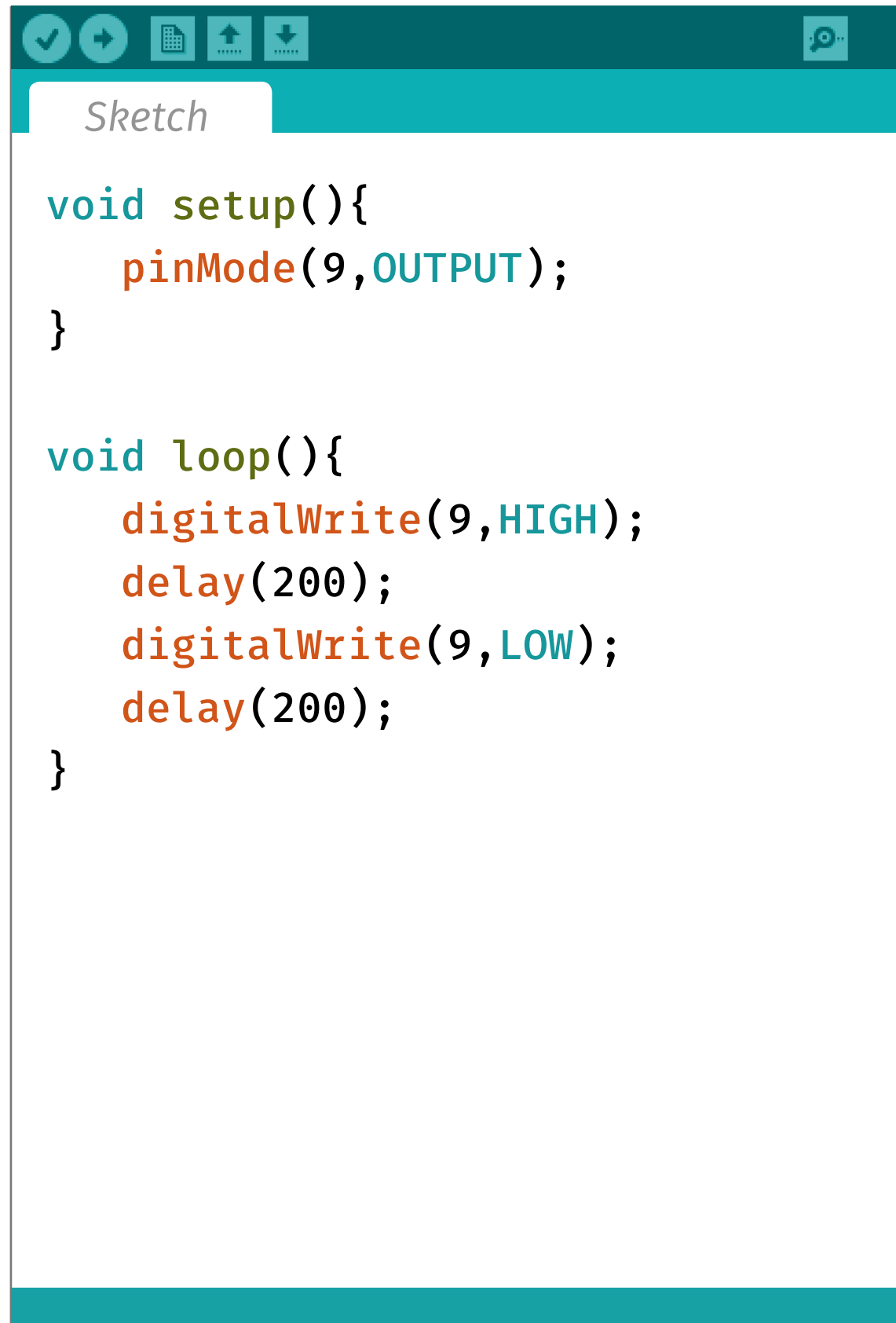
stopping the program for 1000 milliseconds

switching the LED off

stopping the program for 1000 milliseconds

Got it? Great. Make it blinking faster!


Code

The image shows a screenshot of the Arduino IDE interface. At the top, there is a toolbar with icons for checking, running, saving, uploading, and downloading. Below the toolbar is a tab labeled "Sketch". The main area contains the following C++ code:

```
void setup(){  
  pinMode(9,OUTPUT);  
}  
  
void loop(){  
  digitalWrite(9,HIGH);  
  delay(200);  
  digitalWrite(9,LOW);  
  delay(200);  
}
```

The smaller the delay gets, the faster the LED blinks.

Code



```

...
// kurzes Blinken
digitalWrite(9,HIGH);
delay(100);
digitalWrite(9,LOW);
delay(100);

// langes Blinken
digitalWrite(9,HIGH);
delay(400);
digitalWrite(9,LOW);
delay(400);
...

```

Now, we will make a little exercise. Let's the LED blink your name in Morse code.

A point means blink short, a dash means blink long.

A · -	H · · · ·	O - - -	V · · · -
B - · · ·	I · ·	P · - - ·	W · - -
C - · - ·	J · - - -	Q - - · -	X - · · -
D - · ·	K - · -	R · - ·	Y - · - -
E ·	L · - · ·	S · · ·	Z - - · ·
F · · - ·	M - -	T -	
G - - ·	N - ·	U · · -	

Don't forget to place a longer delay after each character. Make an even longer one at the end of your name.

StartHardware.org

*The beauty and elegance of
programmable objects.*



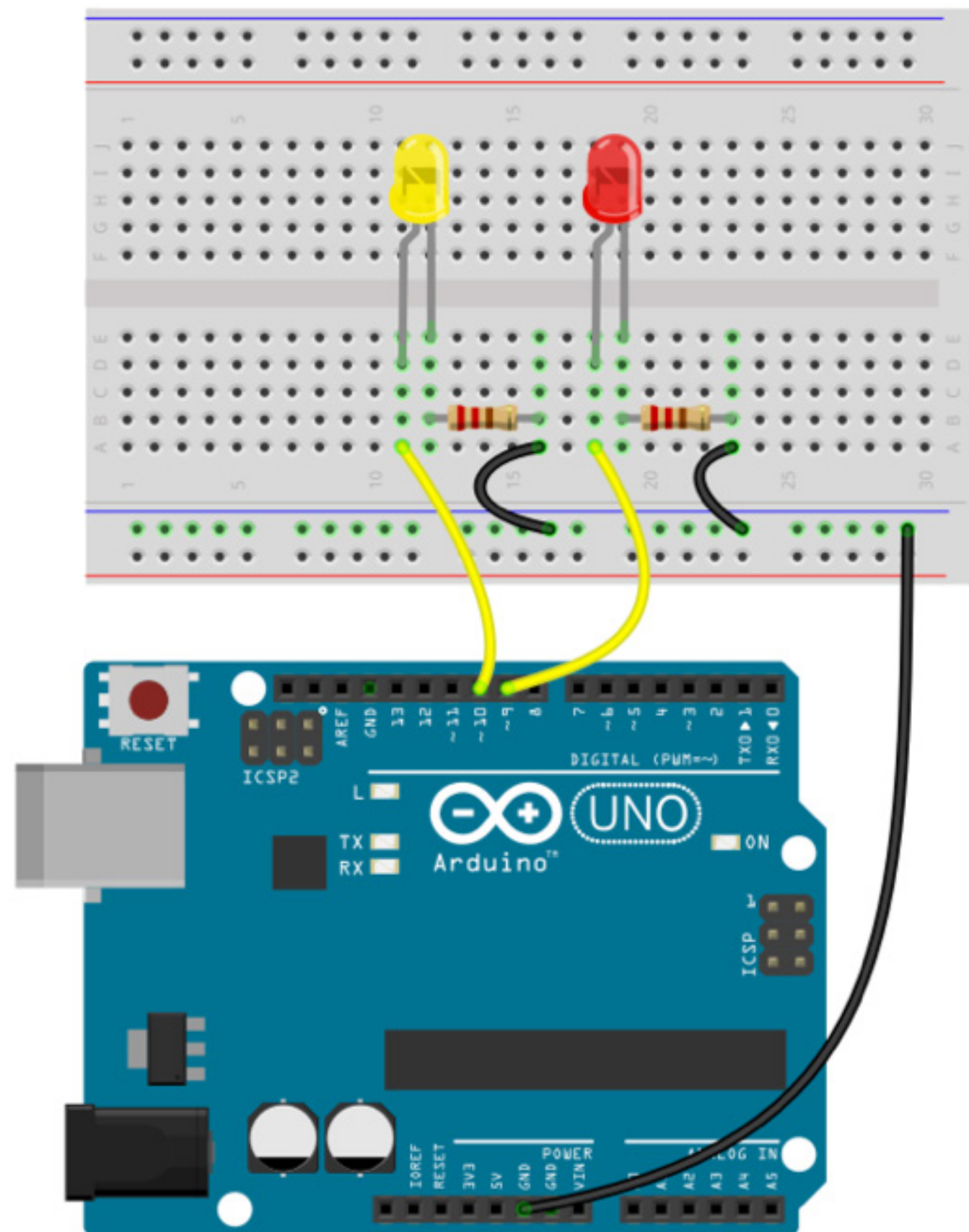
Code

Start with the Blink example code and try to extend it for two LEDs.

```
void setup(){
  pinMode(9,OUTPUT);
}

void loop(){
  digitalWrite(9,HIGH);
  delay(200);
  digitalWrite(9,LOW);
  delay(200);
}
```

Circuit with two LEDs

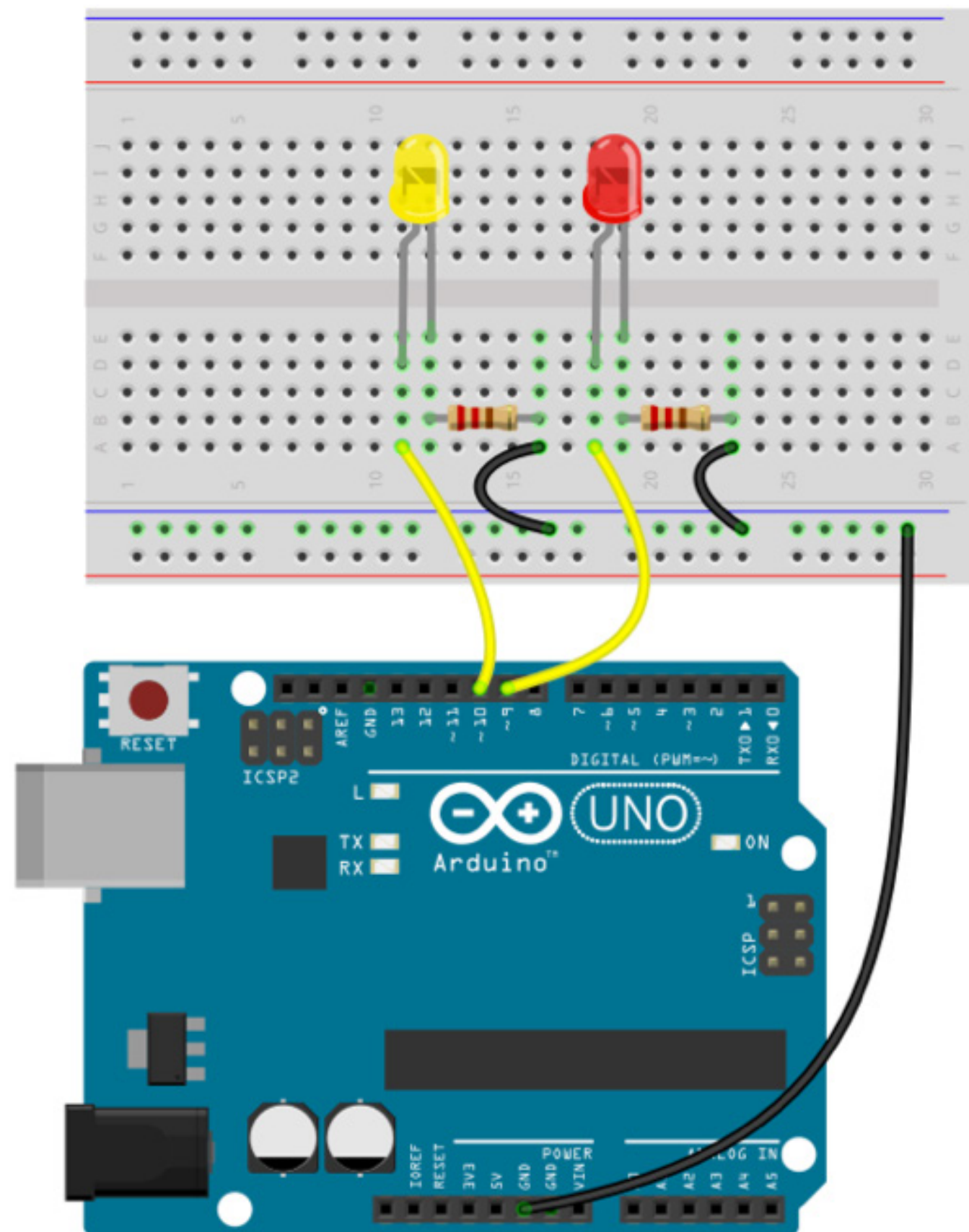


Code

```
void setup(){
  pinMode(9,OUTPUT);
  pinMode(10,OUTPUT);
}

void loop(){
  digitalWrite(9,HIGH);
  digitalWrite(10,LOW);
  delay(200);
  digitalWrite(9,LOW);
  digitalWrite(10,HIGH);
  delay(200);
}
```

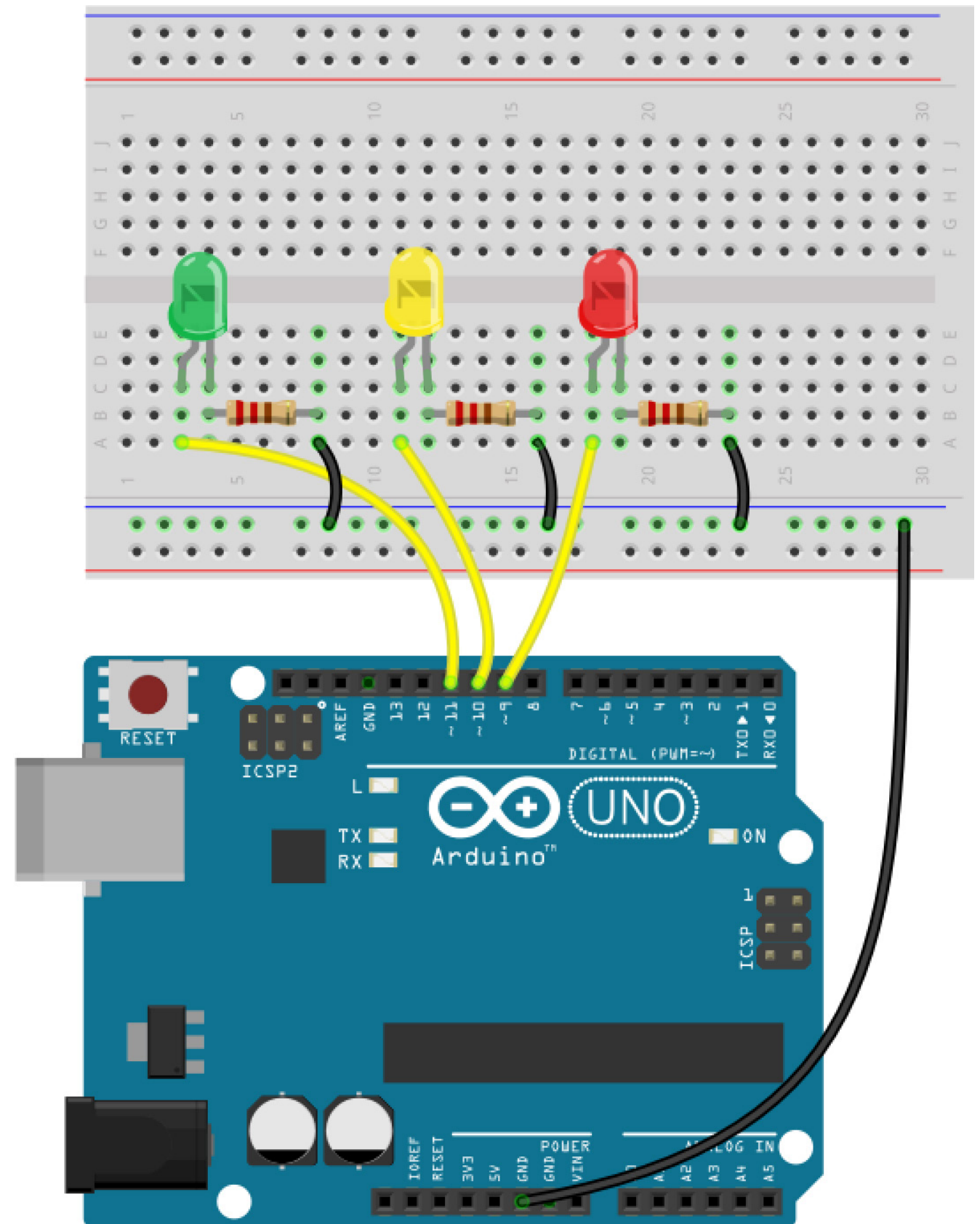
Circuit with two LEDs



Circuit with two LEDs

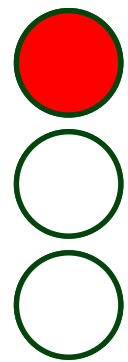
Now, program a traffic light.

What states can it show?

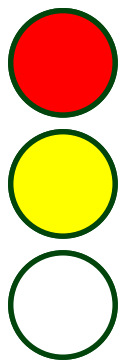


Now, program a traffic light.

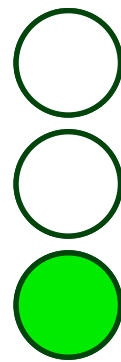
What states can it show?



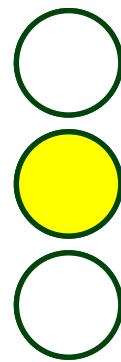
red



red-yellow

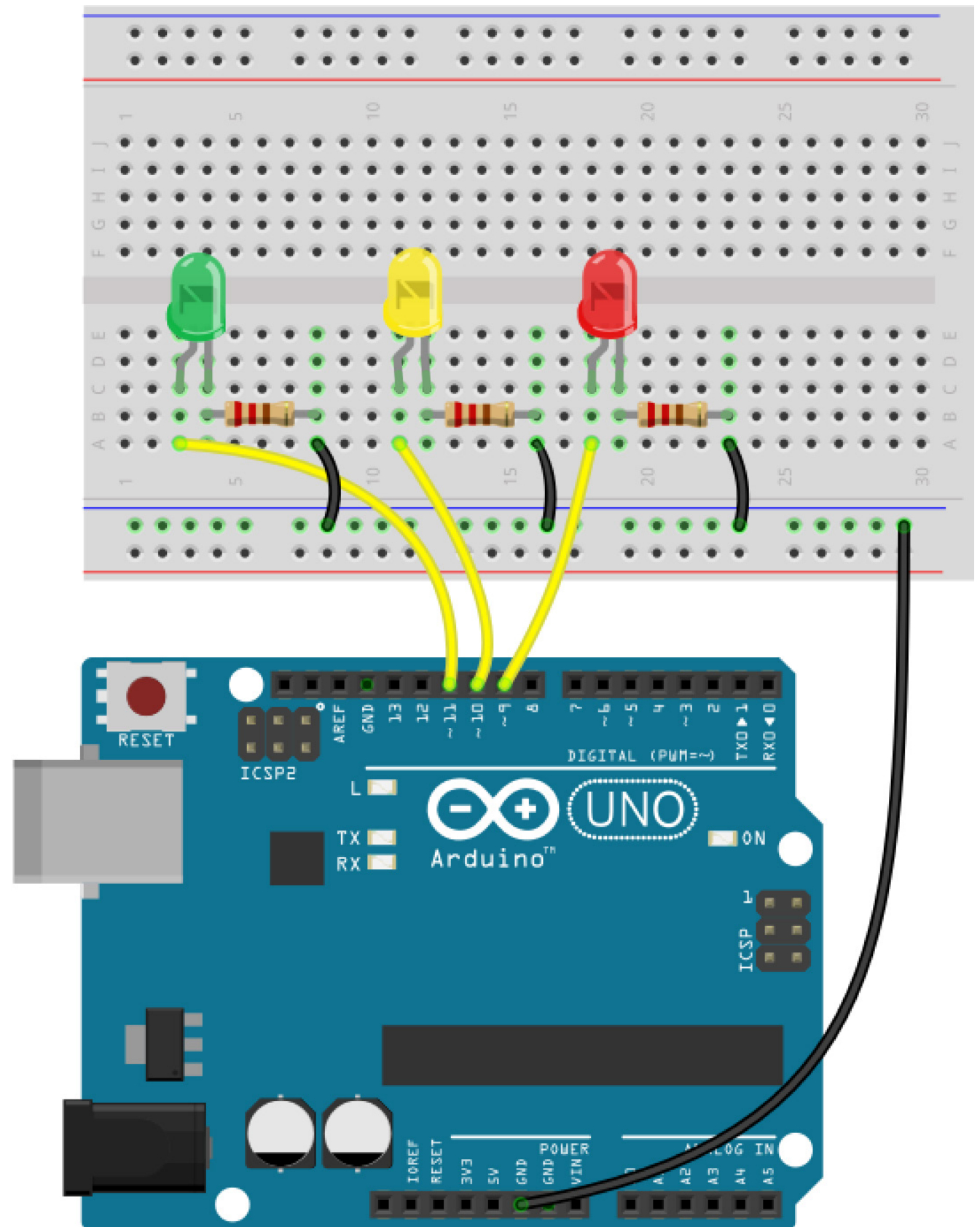


green

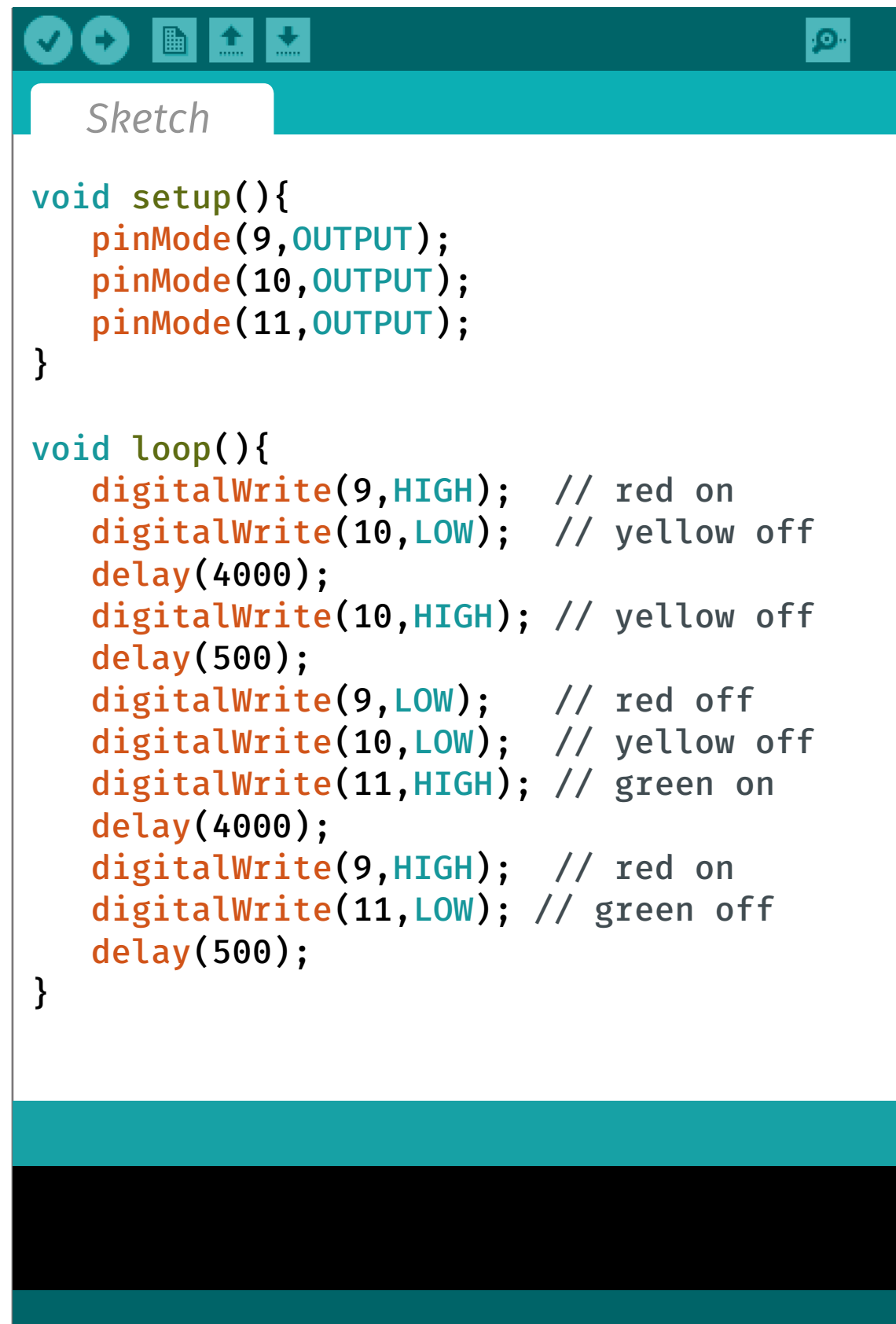


yellow

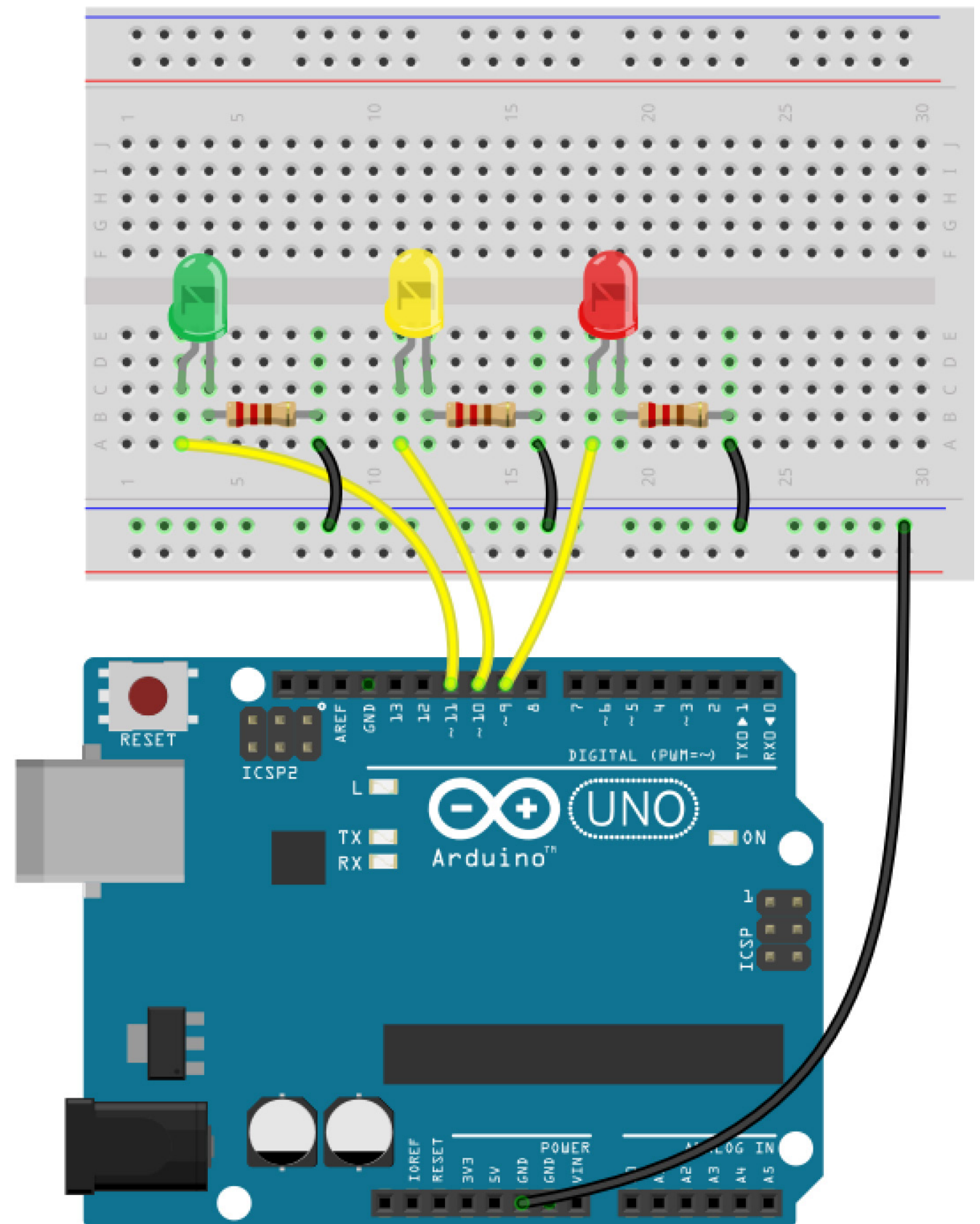
Circuit with three LEDs



Code of a traffic light

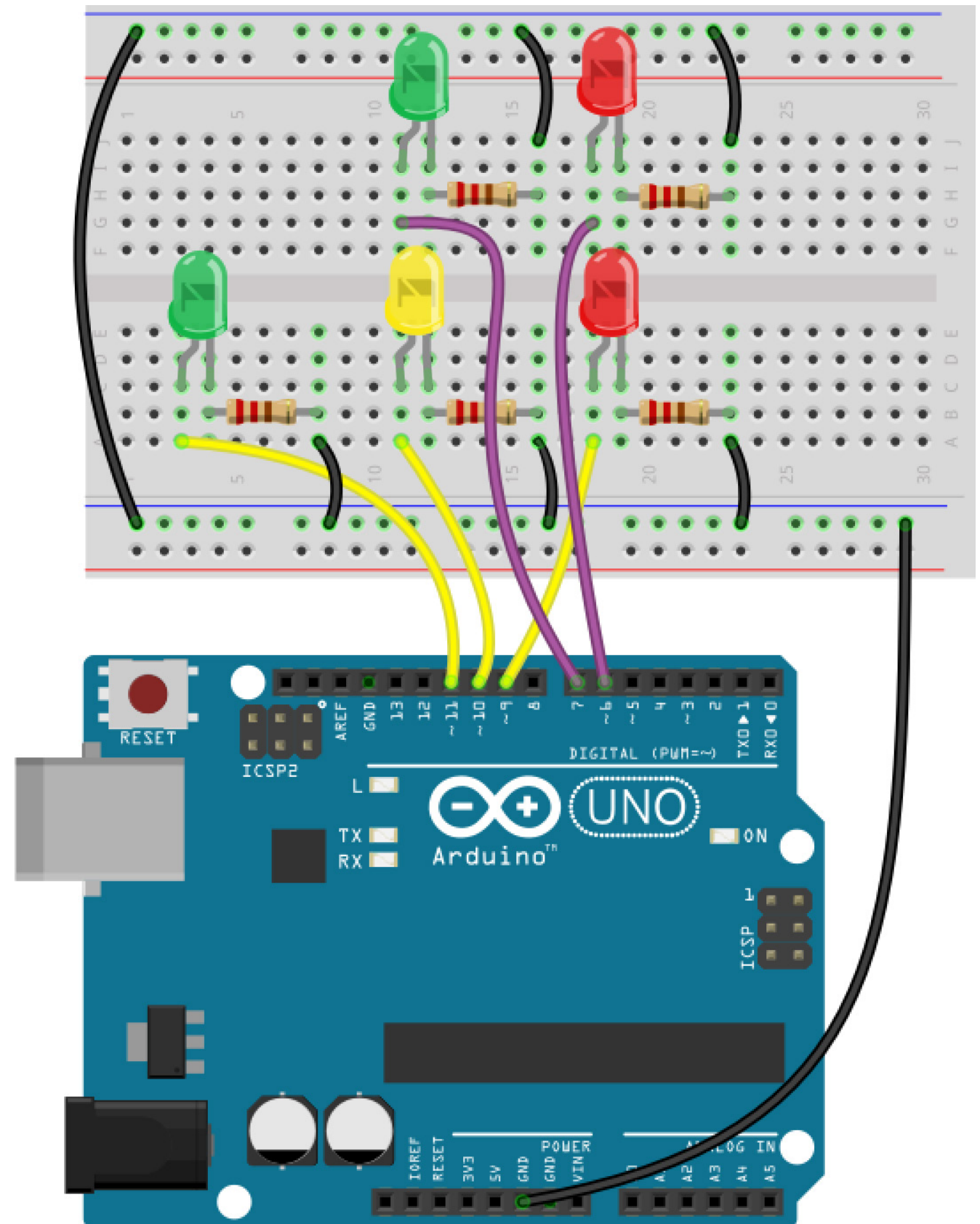


Circuit with three LEDs

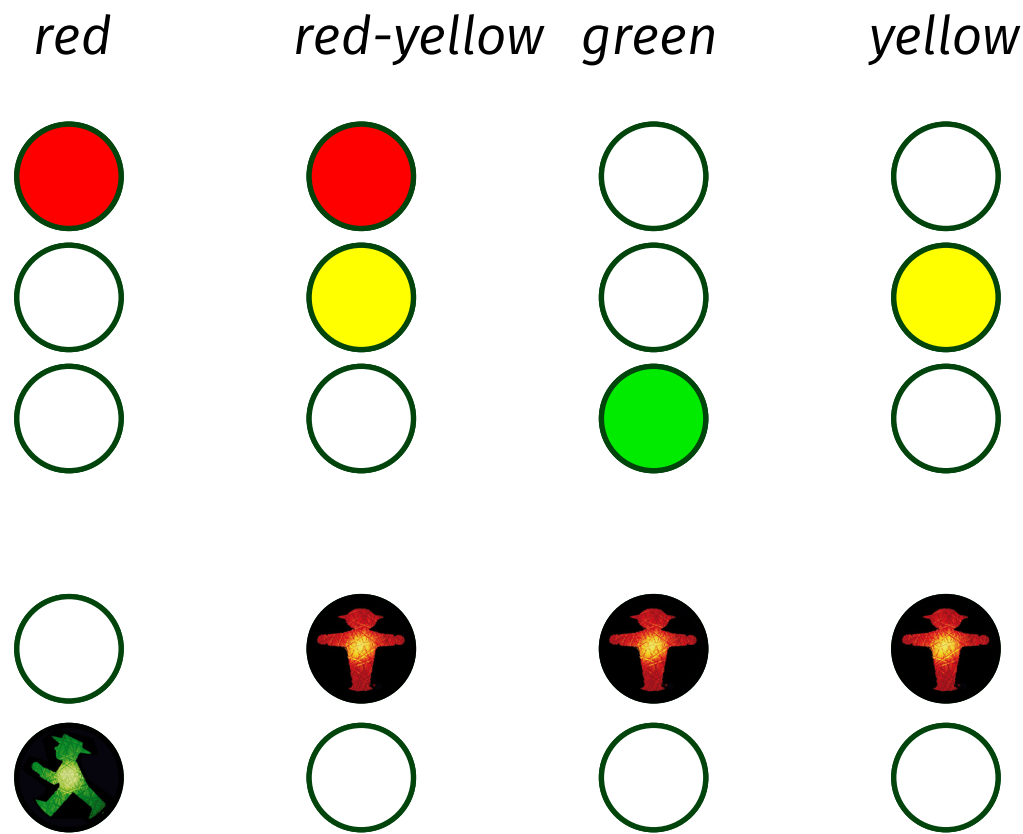


Now, let's try to build an extra pedestrian light.
What states can it show?

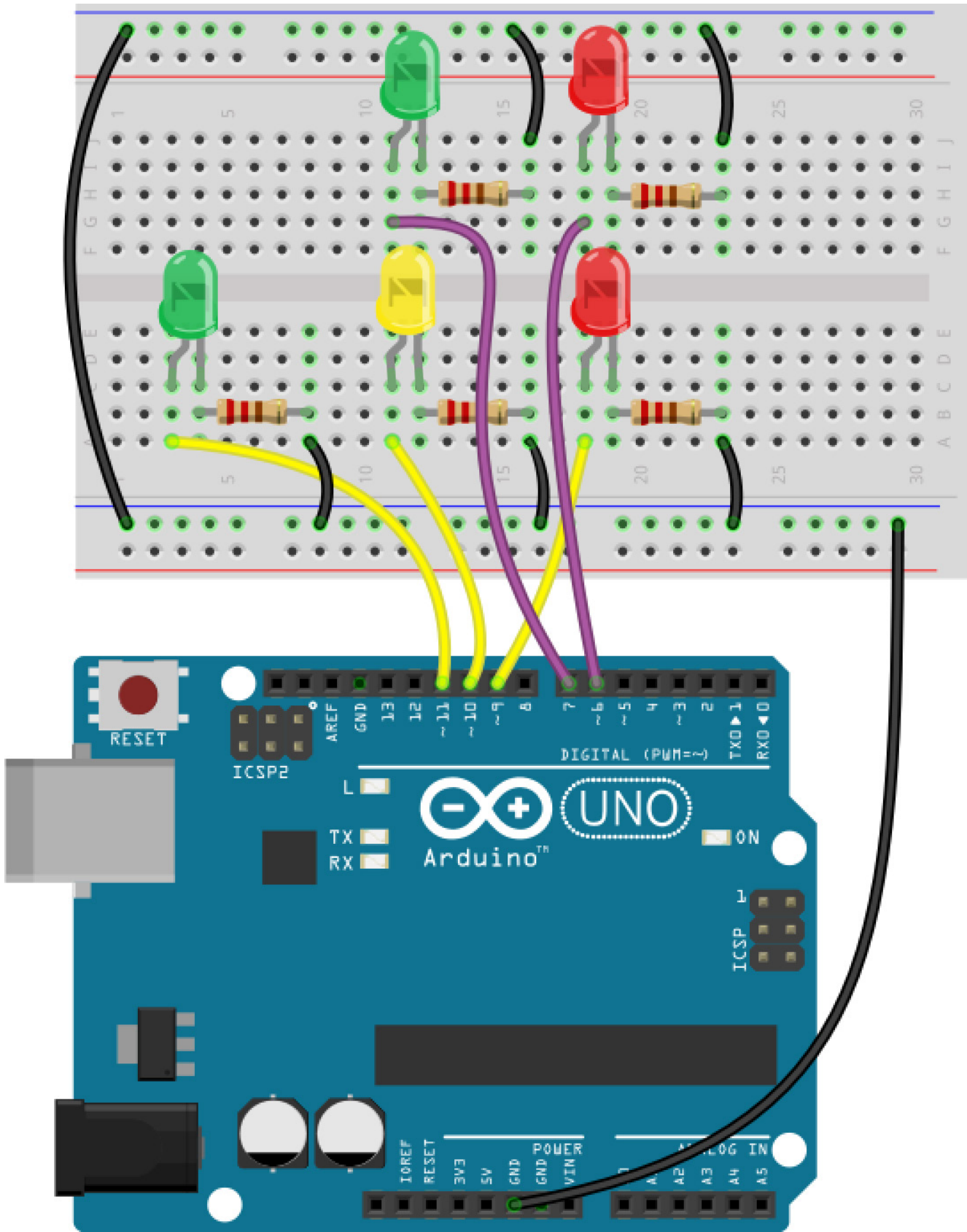
Circuit with five LEDs



Now, let's try to build an extra pedestrian light.
What states can it show?



Circuit with five LEDs



Code of a traffic light

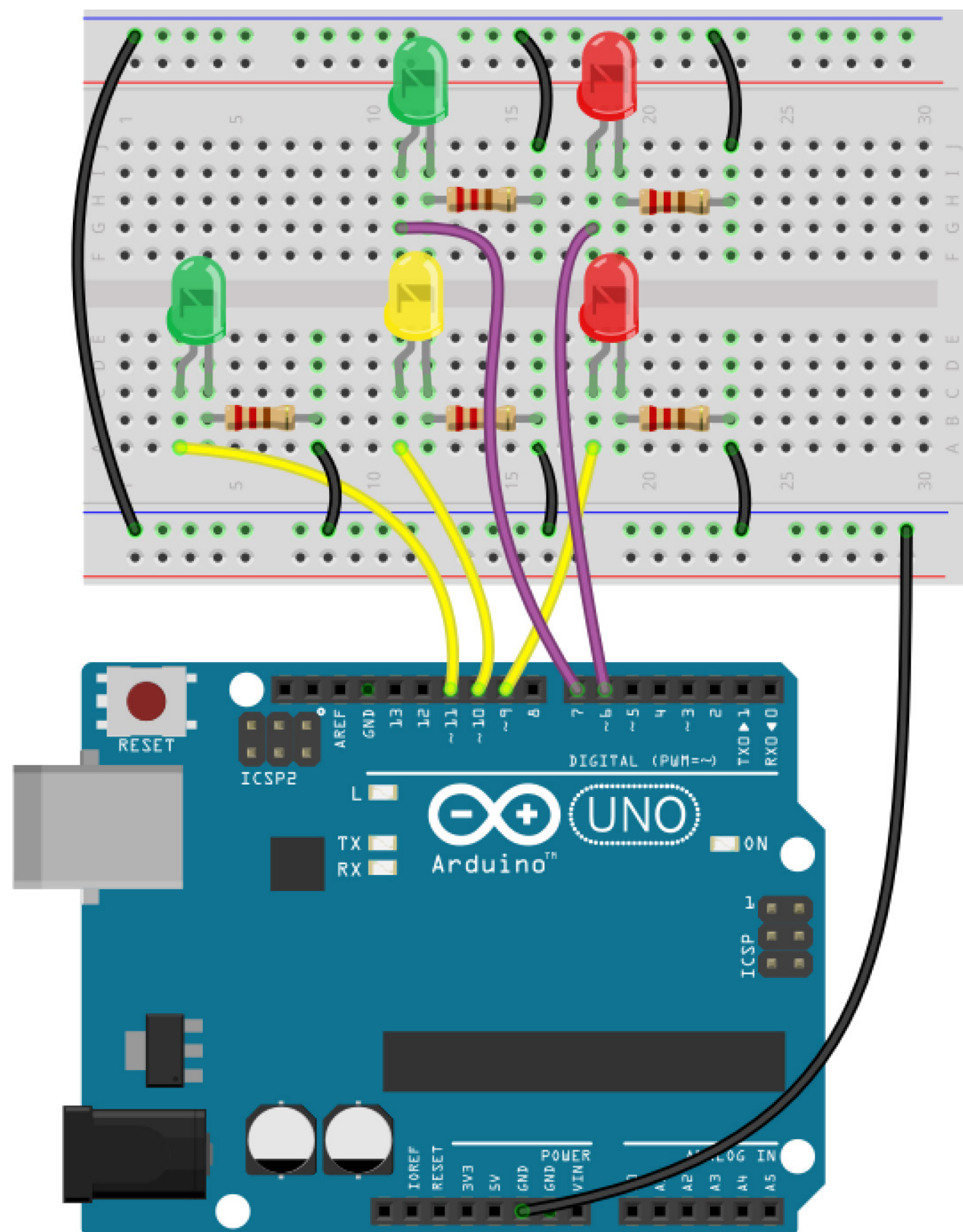
```

Sketch

void setup(){
  pinMode(6,OUTPUT); // red Ped
  pinMode(7,OUTPUT); // green Ped
  pinMode(9,OUTPUT); // red Car
  pinMode(10,OUTPUT); // yellow Car
  pinMode(11,OUTPUT); // green Car
}

void loop(){
  digitalWrite(6,LOW); // red Ped off
  digitalWrite(7,HIGH); // green Ped on
  digitalWrite(9,HIGH); // red on
  digitalWrite(10,LOW); // yellow off
  delay(4000);
  digitalWrite(6,HIGH); // red Ped on
  digitalWrite(7,LOW); // green Ped off
  digitalWrite(10,HIGH); // yellow off
  delay(500);
  digitalWrite(9,LOW); // red off
  digitalWrite(10,LOW); // yellow off
  digitalWrite(11,HIGH); // green on
  delay(4000);
  digitalWrite(9,HIGH); // red on
  digitalWrite(11,LOW); // green off
  delay(500);
}
    
```

Circuit with five LEDs

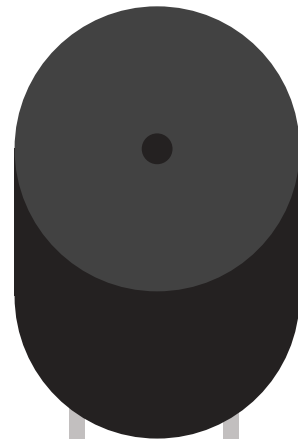


StartHardware.org

*The beauty and elegance of
programmable objects.*



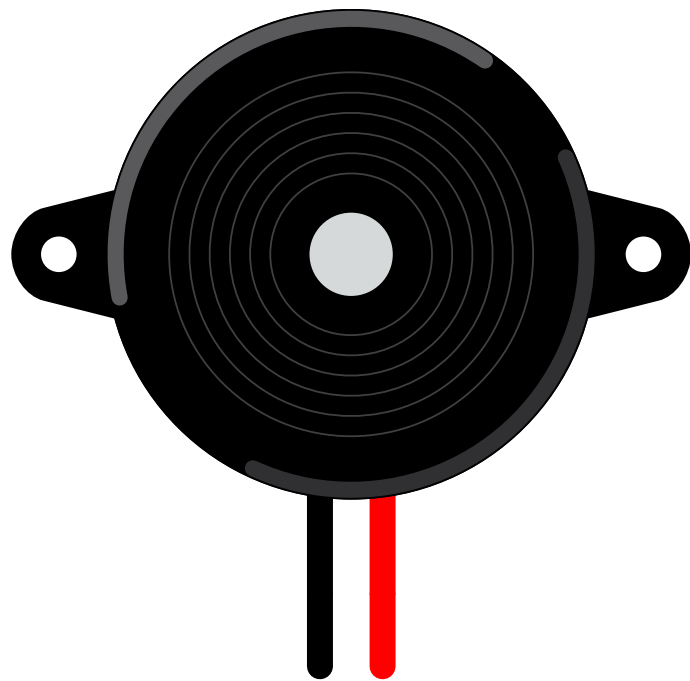
Music!



A piezo speaker transforms electricity into sound

There are a lot of different shapes.

Piezos are used in toys and electronic birthday and christmas cards.

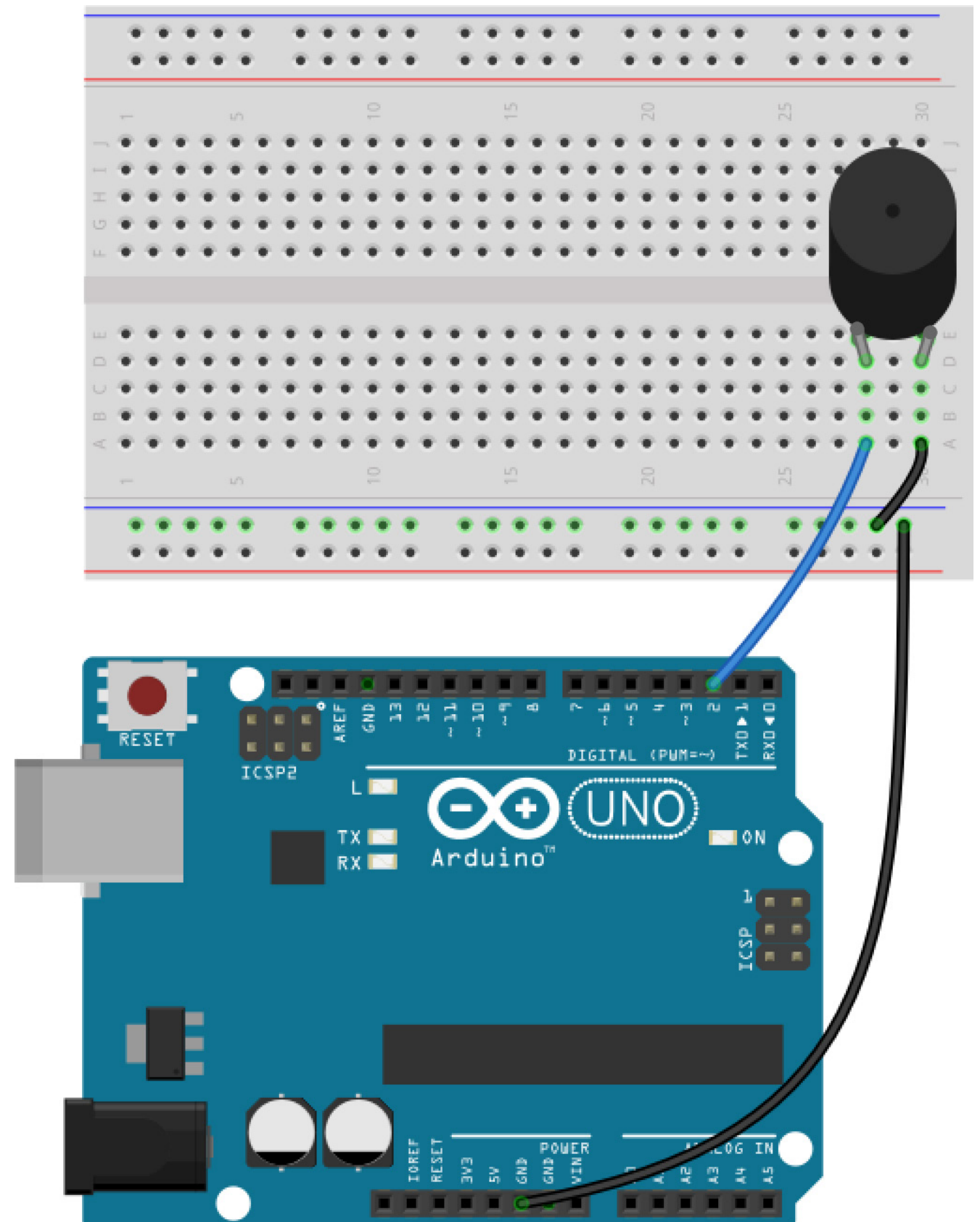


Code

```
void setup(){
}

void loop(){
  tone(2, 400);    // sound is played
  delay(200);      // pause
  noTone(2);       // sound is switched off
  delay(2000);     // pause
}
```

Circuit with a piezo speaker

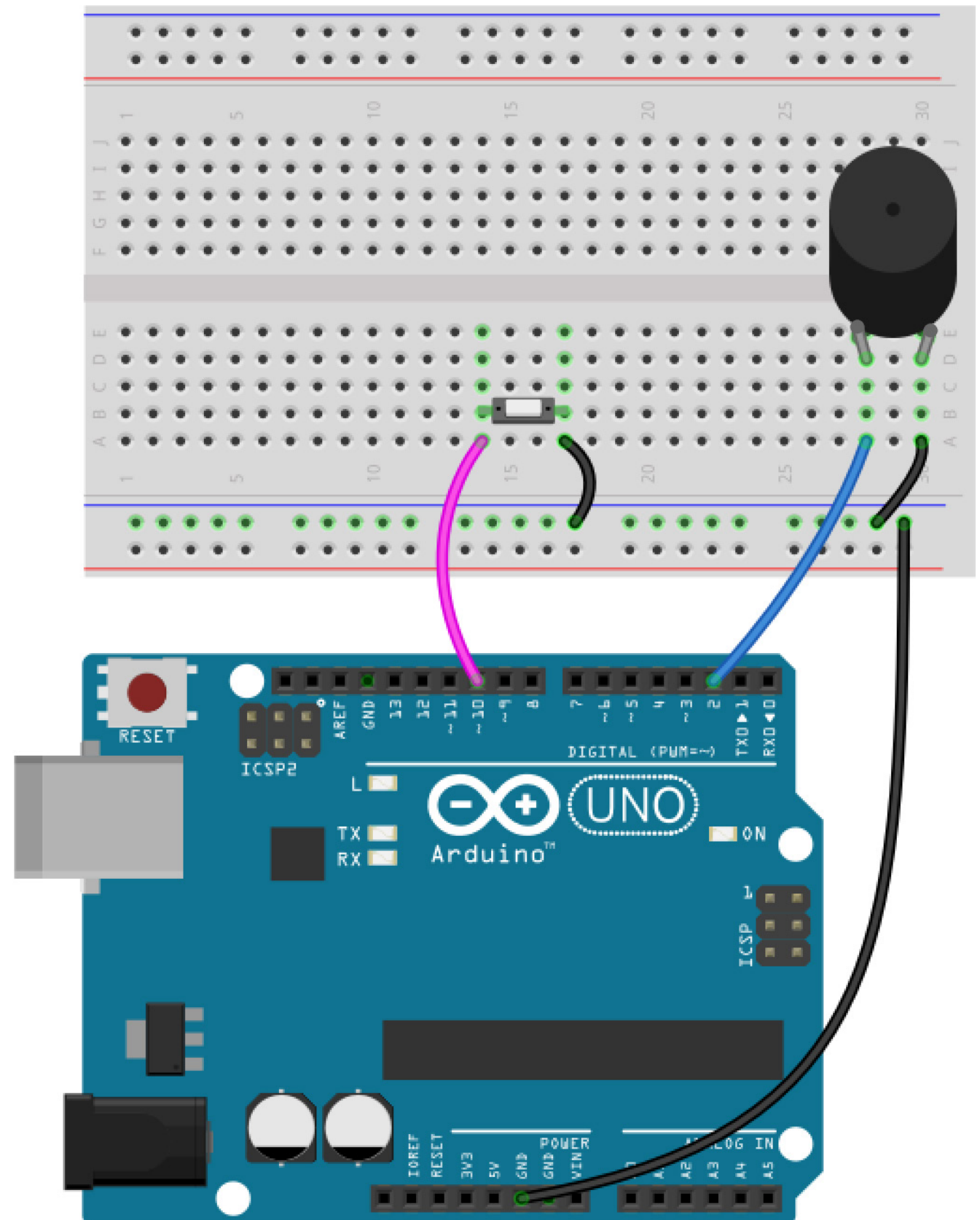


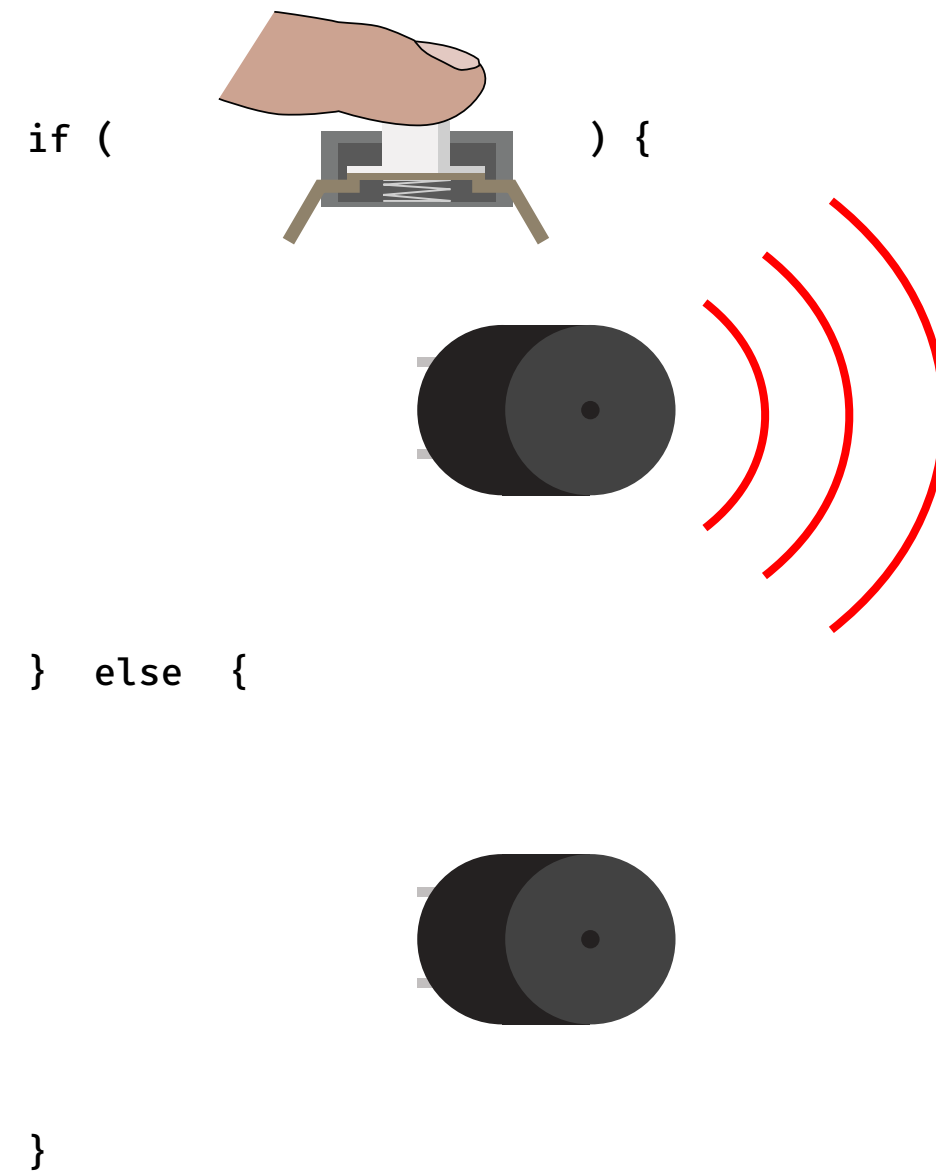
Build the circuit

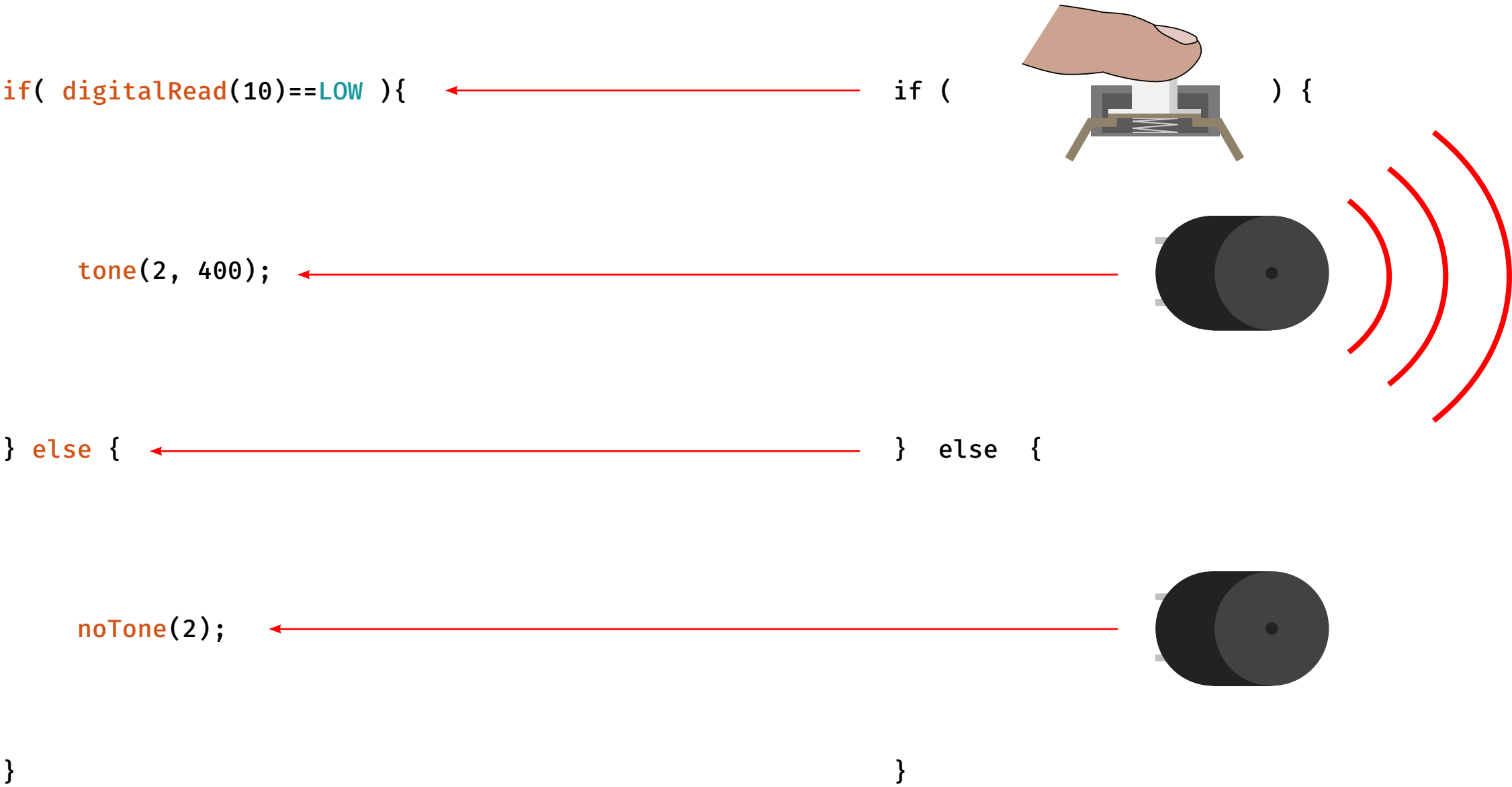
The sound should be played when the button is pushed.

How do we program it?

Circuit with a piezo and a pushbutton



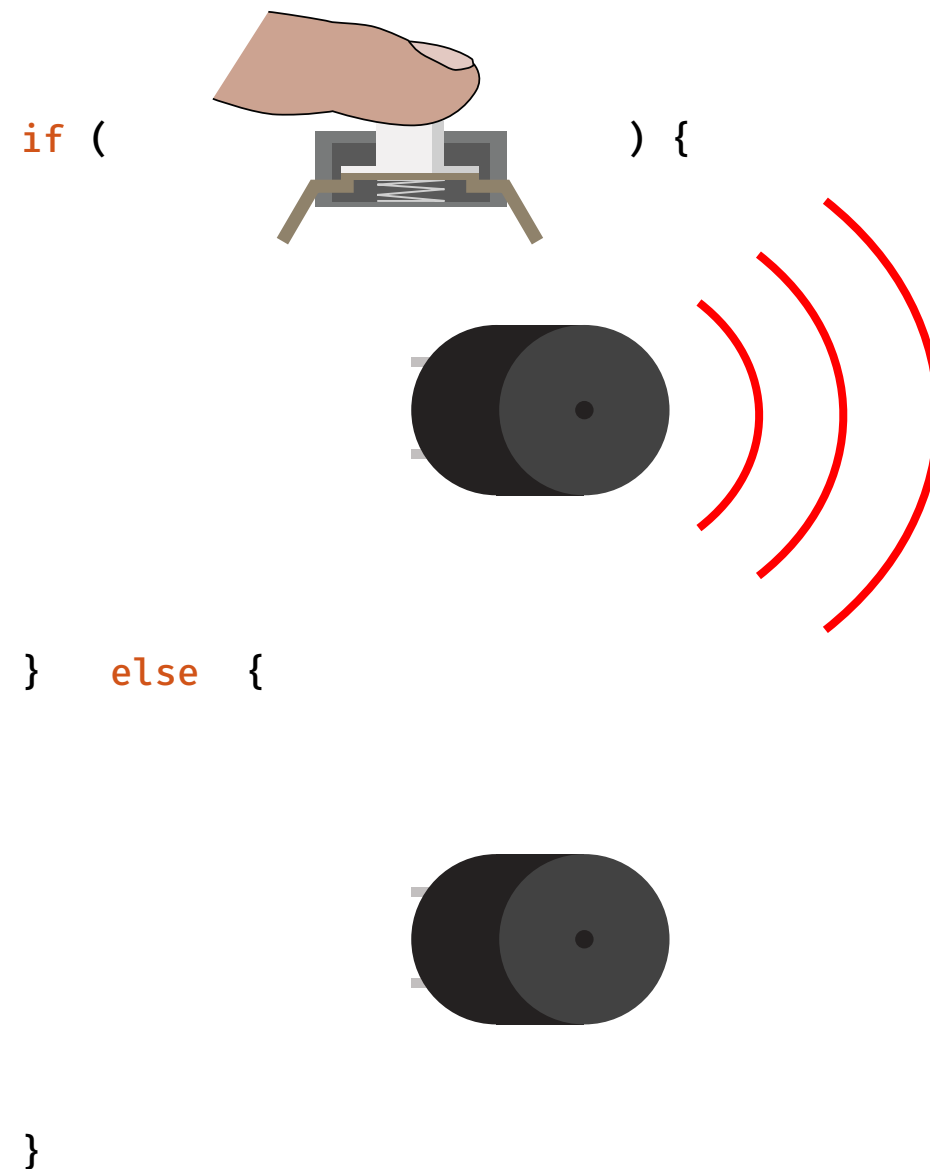




Code

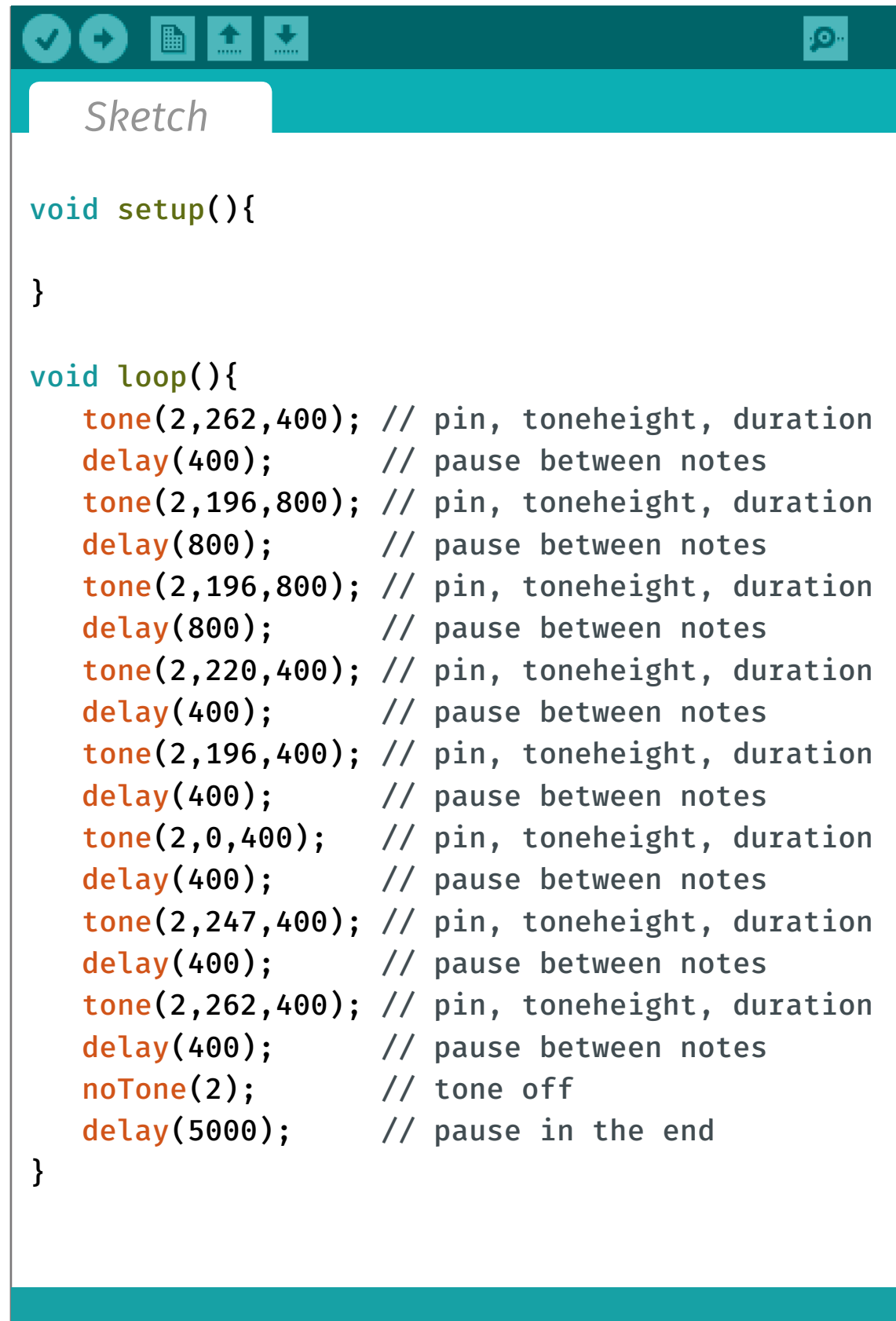
```
void setup(){
  pinMode(10,INPUT_PULLUP);    // pushbutton
}

void loop(){
  if(digitalRead(10)==LOW){
    tone(2, 400);
  } else {
    noTone(2);
  }
}
```



Melodie

Code

The image shows the Arduino IDE interface. At the top, there is a toolbar with icons for checking, running, uploading, and downloading. Below the toolbar, the 'Sketch' tab is active. The code editor contains the following C++ code:

```
void setup(){  
  
}  
  
void loop(){  
  tone(2,262,400); // pin, toneheight, duration  
  delay(400);      // pause between notes  
  tone(2,196,800); // pin, toneheight, duration  
  delay(800);      // pause between notes  
  tone(2,196,800); // pin, toneheight, duration  
  delay(800);      // pause between notes  
  tone(2,220,400); // pin, toneheight, duration  
  delay(400);      // pause between notes  
  tone(2,196,400); // pin, toneheight, duration  
  delay(400);      // pause between notes  
  tone(2,0,400);   // pin, toneheight, duration  
  delay(400);      // pause between notes  
  tone(2,247,400); // pin, toneheight, duration  
  delay(400);      // pause between notes  
  tone(2,262,400); // pin, toneheight, duration  
  delay(400);      // pause between notes  
  noTone(2);        // tone off  
  delay(5000);      // pause in the end  
}
```

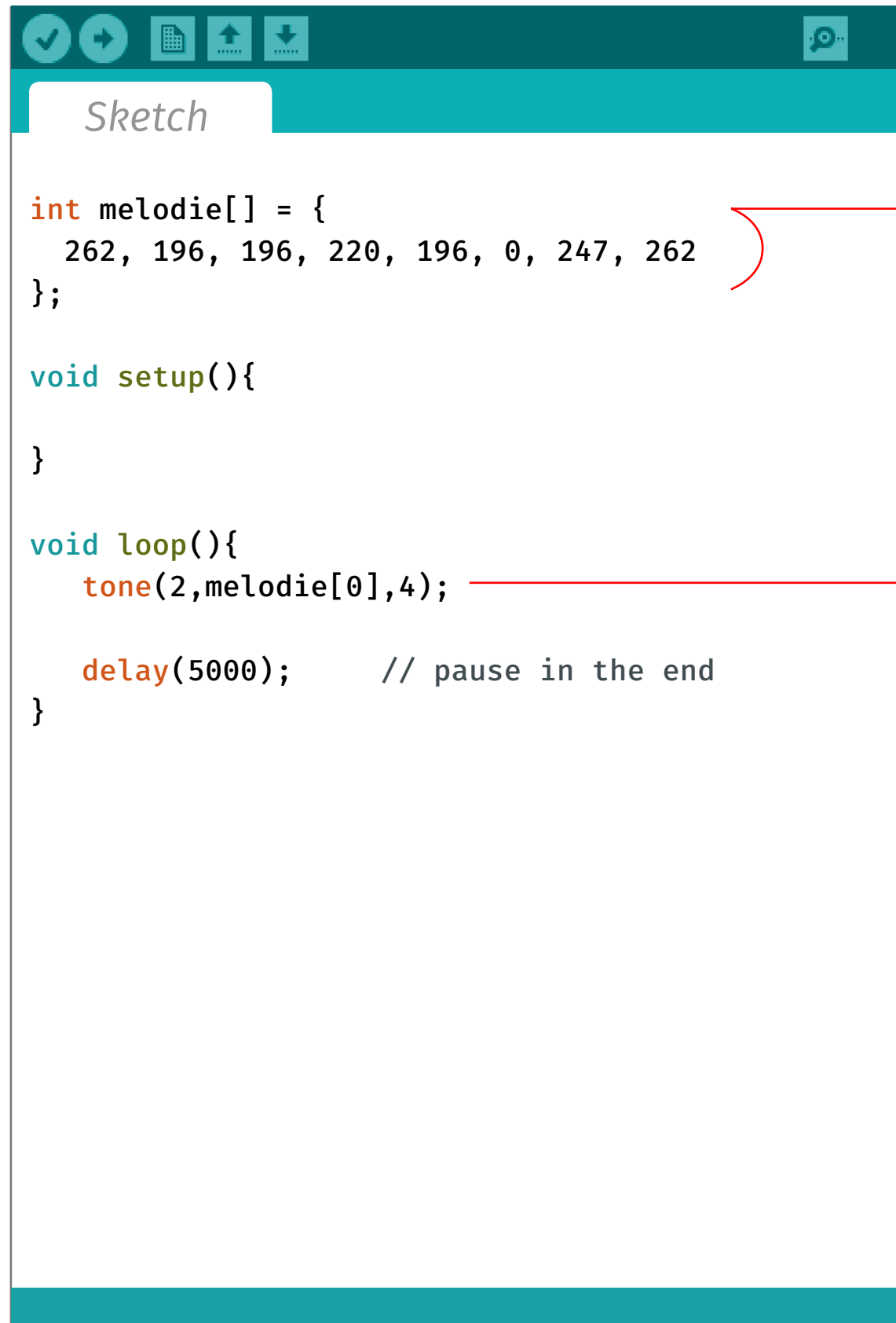
A melody consists of tones in different heights and durations:

262, 196, 196, 220, 196, 0, 247, 262
4, 8, 8, 4, 4, 4, 4, 4

Even the code of a short melody becomes very complex. It would be great, if we could store the duration and tone informations differently.

Any ideas?

Code



```
int melodie[] = {
  262, 196, 196, 220, 196, 0, 247, 262
};

void setup(){

}

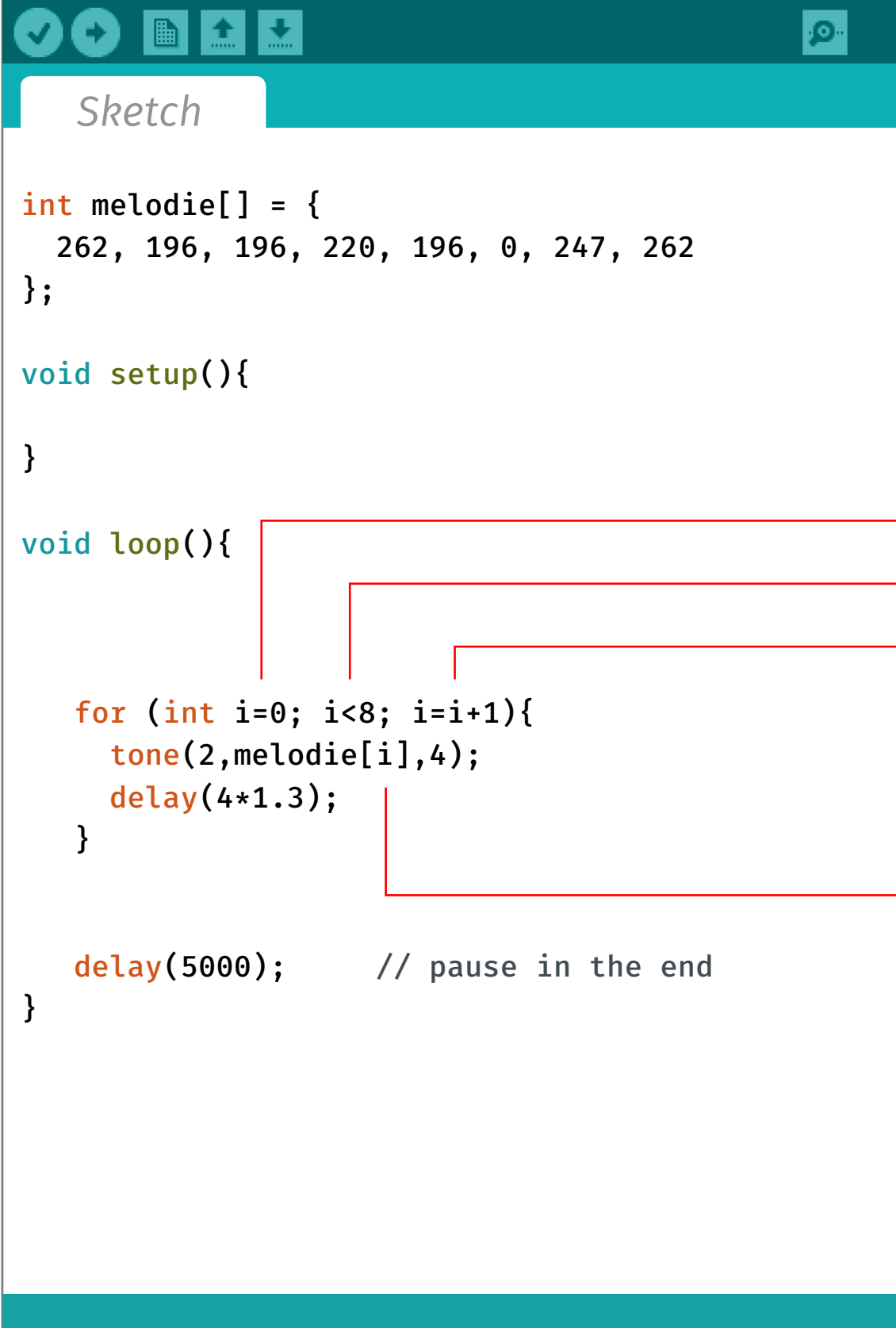
void loop(){
  tone(2,melodie[0],4);
  delay(5000);    // pause in the end
}
```

This construction is an «array». The word `int` tells it can store integer numbers.

If you want to read out a value, you just need to write the arrays name and ask for the position in rectangular brackets. 0 is the first position of an array. So it returns 262.

Now, we need something to repeat the tone but shifting the asked position.

Code



```
int melodie[] = {
  262, 196, 196, 220, 196, 0, 247, 262
};

void setup(){

}

void loop(){

  for (int i=0; i<8; i=i+1){
    tone(2,melodie[i],4);
    delay(4*1.3);
  }

  delay(5000);    // pause in the end
}
```

The for-loop repeats commands for a defined duration.

```
for (start value, condition, sequel) {
  commands
}
```

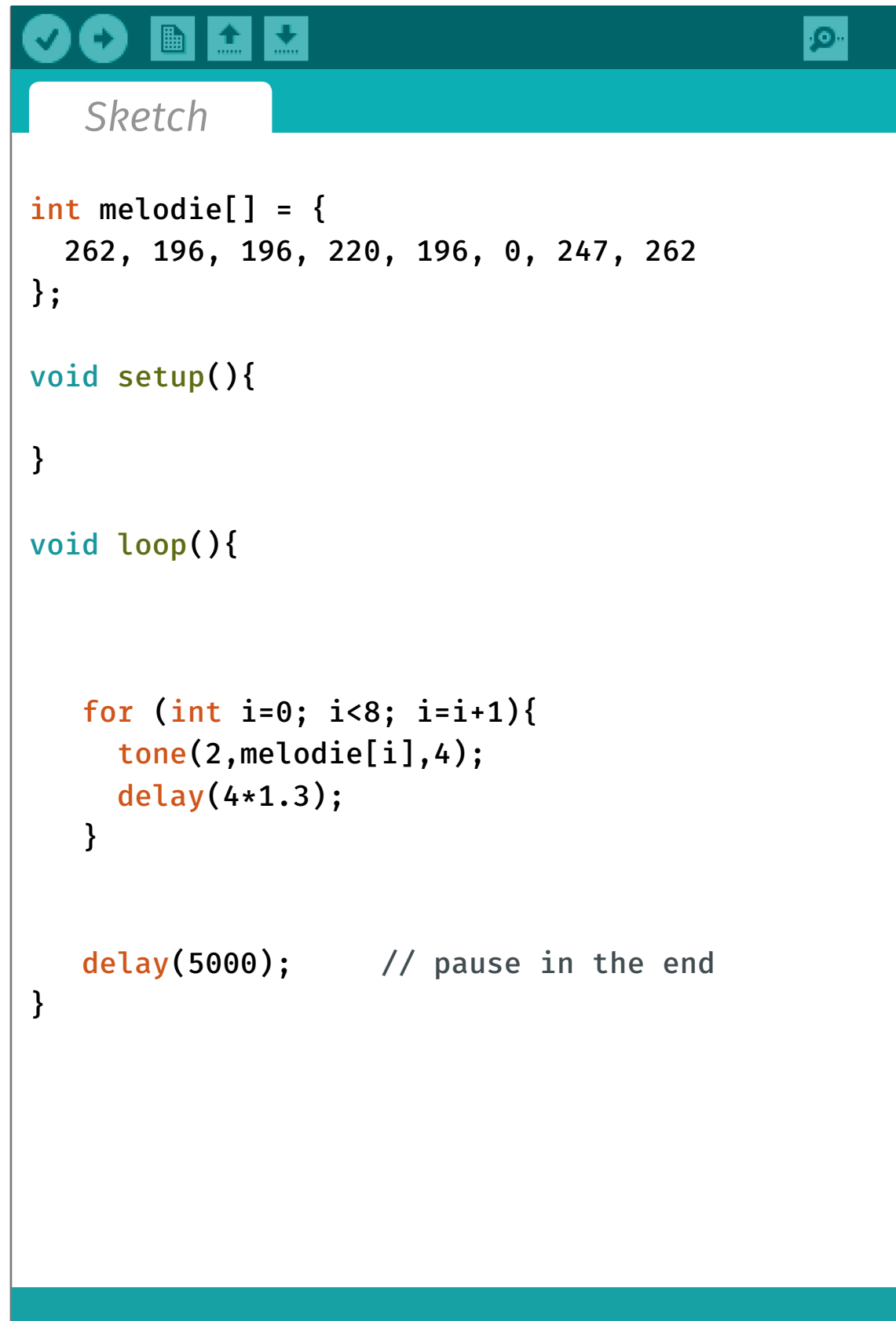
start value of a counting variable is 0

condition: as long as i is smaller than 8

sequel: i is counting up in steps of 1

The variable i is counting up by 1 per repetition of the for-loop, starting at 0 and ending at 7 (smaller than 8). Now, we just have to send it to our array.

Code

The image shows the Arduino IDE interface. At the top, there is a toolbar with icons for checking, running, saving, uploading, and downloading. Below the toolbar is a tab labeled "Sketch". The main area contains the following code:

```
int melodie[] = {
  262, 196, 196, 220, 196, 0, 247, 262
};

void setup(){

}

void loop(){

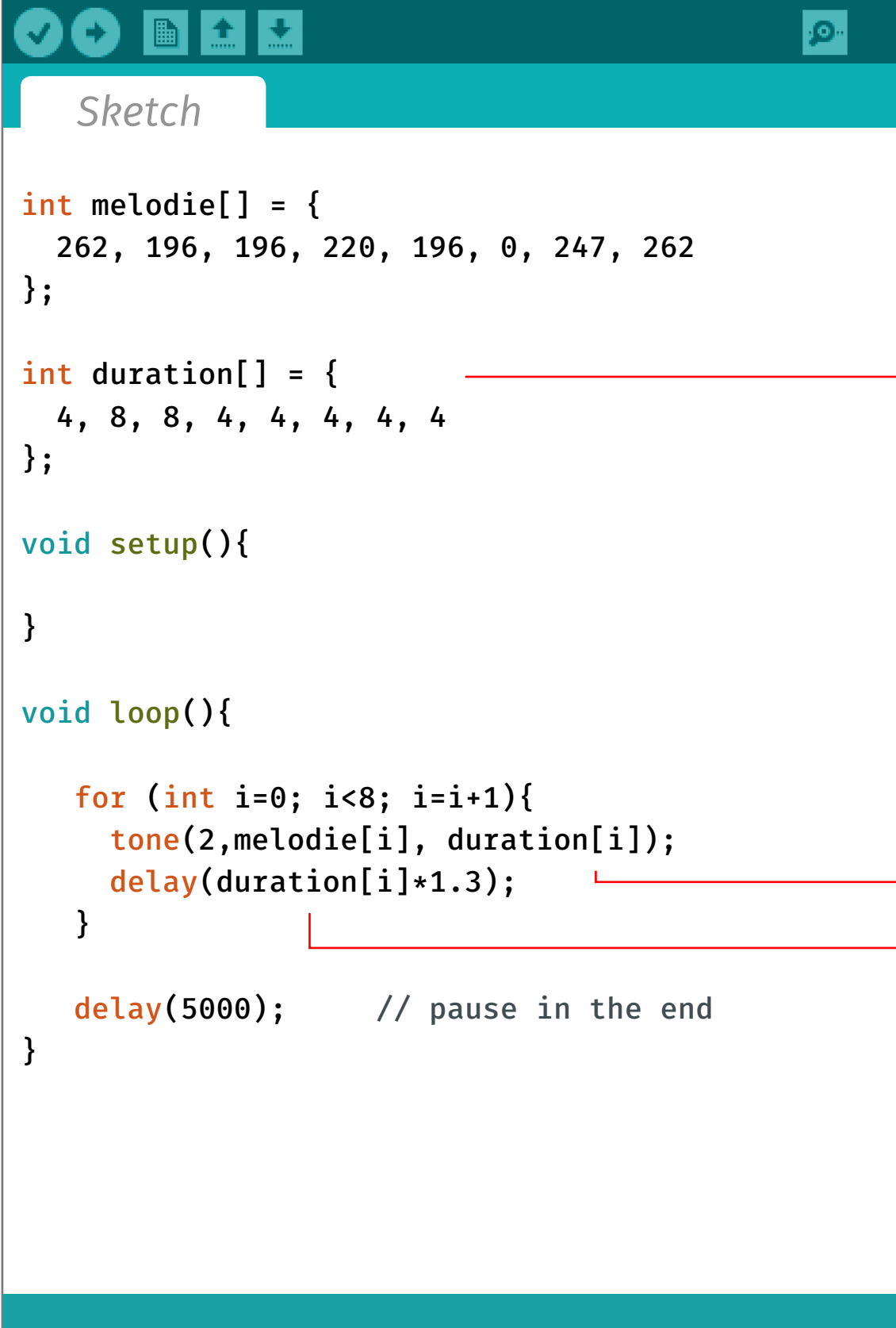
  for (int i=0; i<8; i=i+1){
    tone(2,melodie[i],4);
    delay(4*1.3);
  }

  delay(5000);    // pause in the end
}
```

Now, we will do the same for the tone duration.

4, 8, 8, 4, 4, 4, 4, 4

Code



```
int melodie[] = {
  262, 196, 196, 220, 196, 0, 247, 262
};

int duration[] = {
  4, 8, 8, 4, 4, 4, 4, 4
};

void setup(){

}

void loop(){

  for (int i=0; i<8; i=i+1){
    tone(2,melodie[i], duration[i]);
    delay(duration[i]*1.3);
  }

  delay(5000);    // pause in the end
}
```

This second array stores the tone duration.

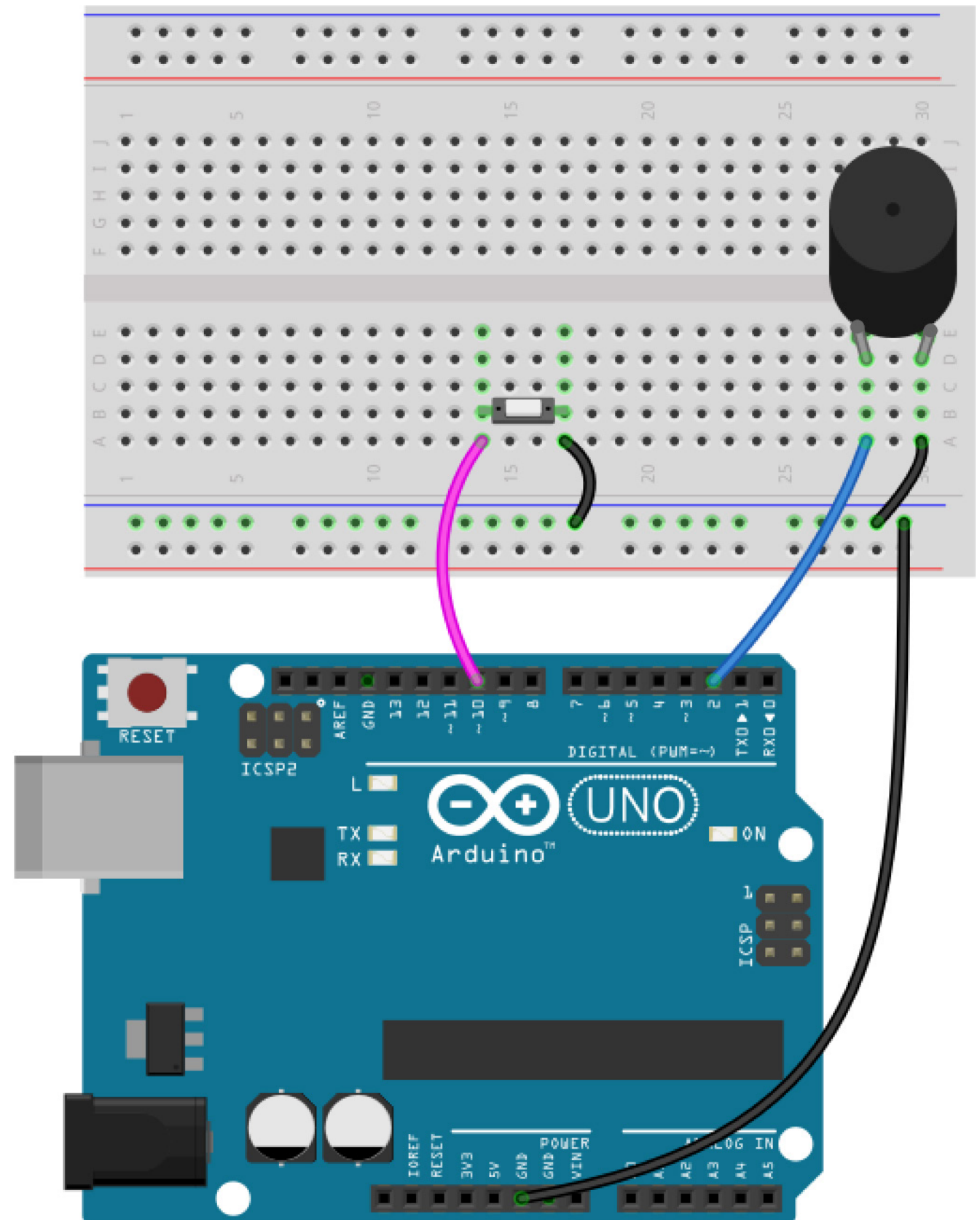
Here, we need to send the info for the tone height and for the duration.

One-Button-Piano

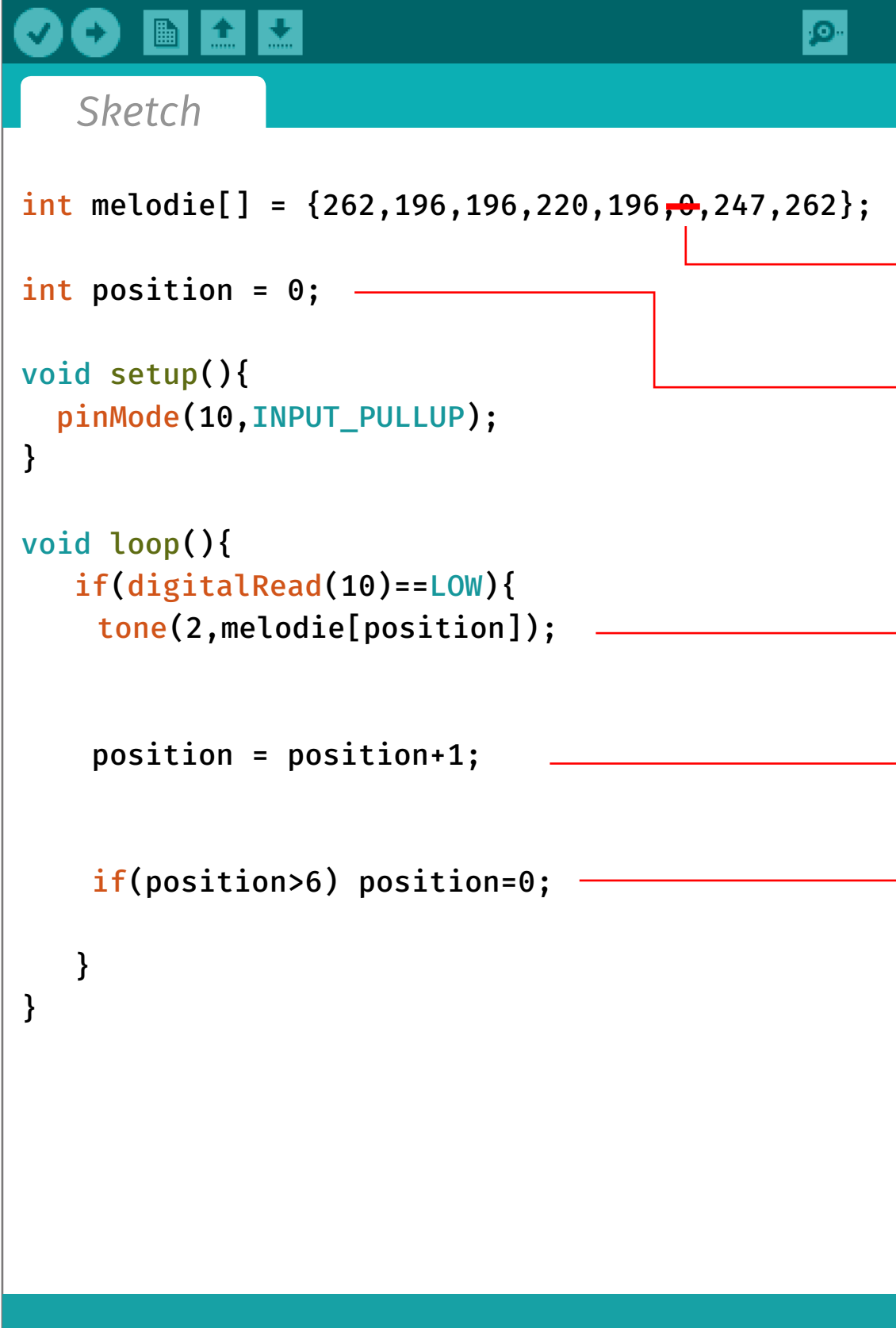
Build the circuit

Now, we are building a piano with only one button.

Circuit with a piezo and a button



Code



```
int melodie[] = {262,196,196,220,196,0,247,262};

int position = 0;

void setup(){
  pinMode(10,INPUT_PULLUP);
}

void loop(){
  if(digitalRead(10)==LOW){
    tone(2,melodie[position]);

    position = position+1;

    if(position>6) position=0;
  }
}
```

Each time the button is pressed, the next tone of the melody should play. (The tone 0 can be deleted)

The variable position should store the position of the melody which was played last.

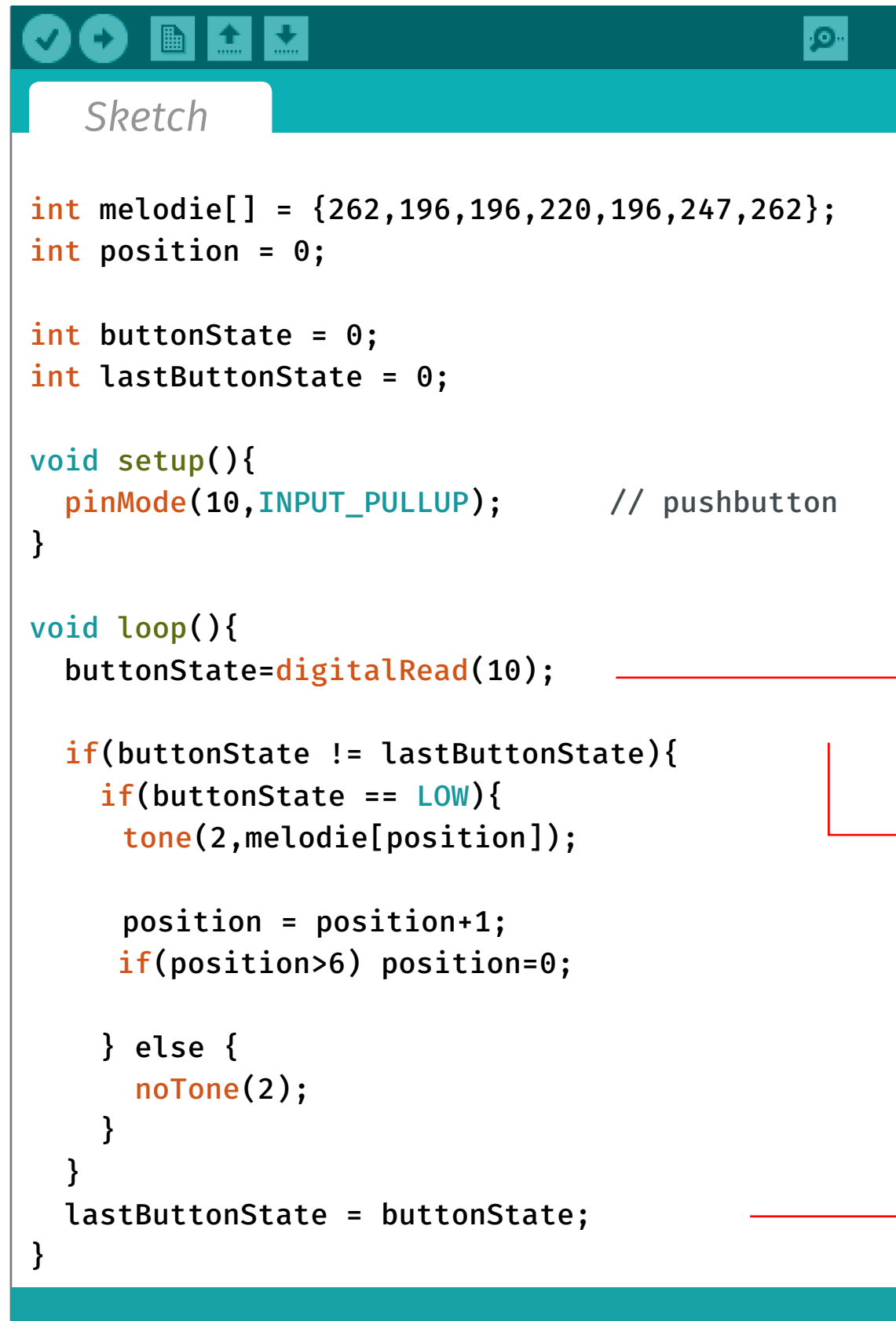
When the button is pressed, the tone of the melody array is played.

The variable position is increased.

If the position gets larger than 6, position is set back to 0.

After the upload, we see that this is still not a very satisfying result. What could be the reason?

Code



```
int melodie[] = {262,196,196,220,196,247,262};
int position = 0;

int buttonState = 0;
int lastButtonState = 0;

void setup(){
  pinMode(10,INPUT_PULLUP);      // pushbutton
}

void loop(){
  buttonState=digitalRead(10);

  if(buttonState != lastButtonState){
    if(buttonState == LOW){
      tone(2,melodie[position]);

      position = position+1;
      if(position>6) position=0;

    } else {
      noTone(2);
    }
  }
  lastButtonState = buttonState;
}
```

The variable position changes as long as the button is pressed and hold. That means all the time!

Actually, we only want to increase position when the button is pressed (not hold).

The state of the button is stored in the variable buttonState.

If the state of the button changed, the program is executed.

In the end, we put the value of the button into the variable lastButtonState.

Light plays music

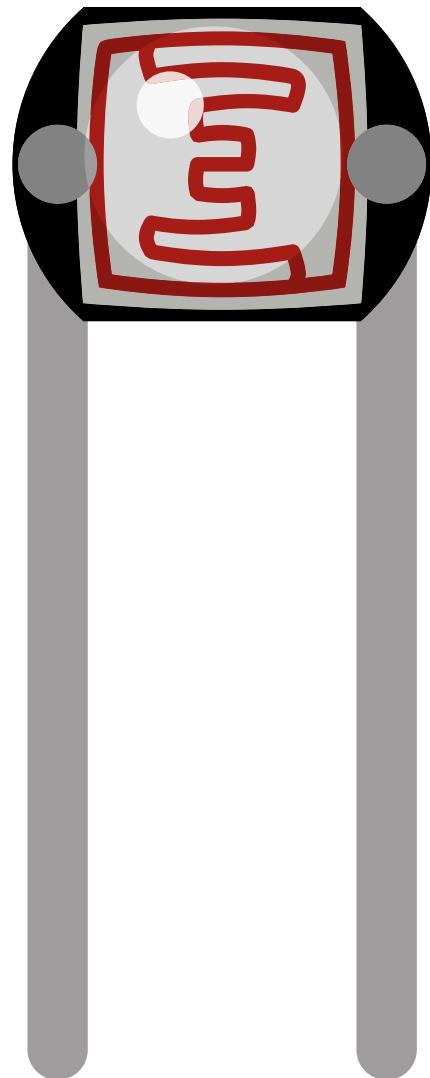


photo resistor

A photo resistor reacts on light

The more light hits the photo resistor, the smaller it's inner resistance gets. The darker it gets, the higher the inner resistance gets.

The Arduino board is capable of detect that change. Therefore, we will need an additional reference resistor fitting to our «sensor» (e.g. 100 k Ω).



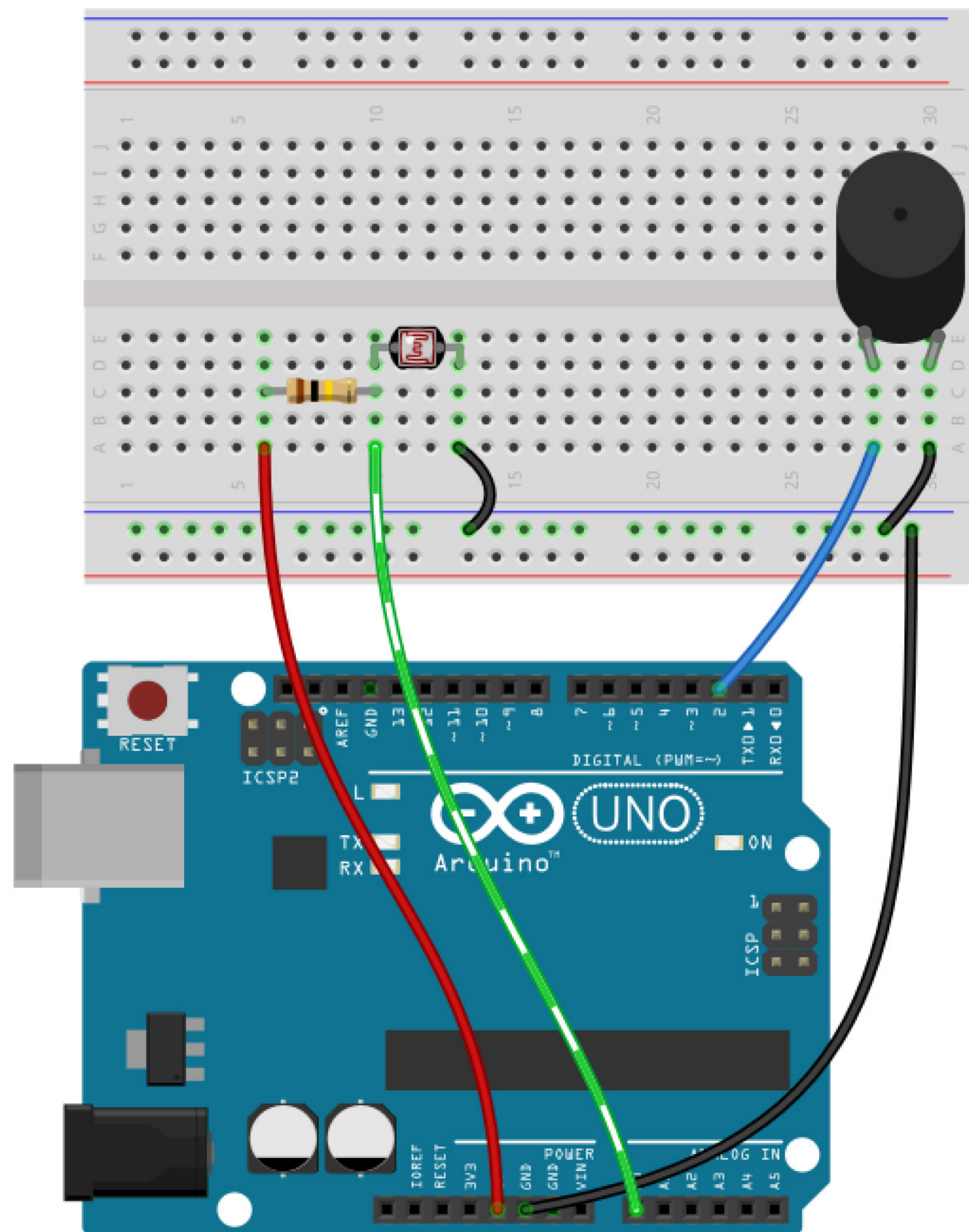
reference resistor 100 k Ω

Build the circuit

Photo resistor and reference resistor are connected in series. The one side is connected to the 5V (red wire) and the other side to the GND (black wire).

The junction between both is connected to an analog input pin of the Arduino-Board.


Circuit with a photo resistor





Circuit with a photo resistor


✓

→



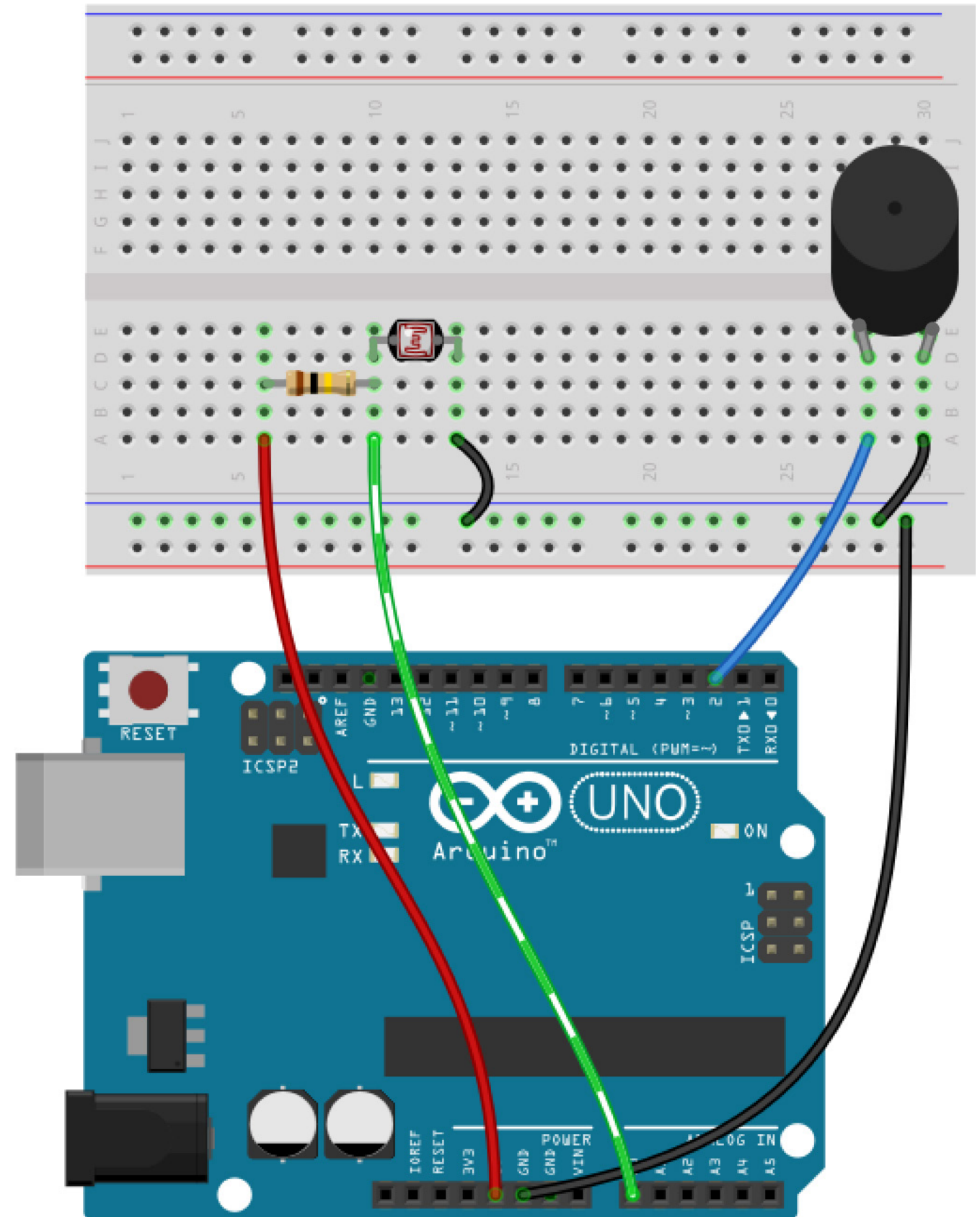




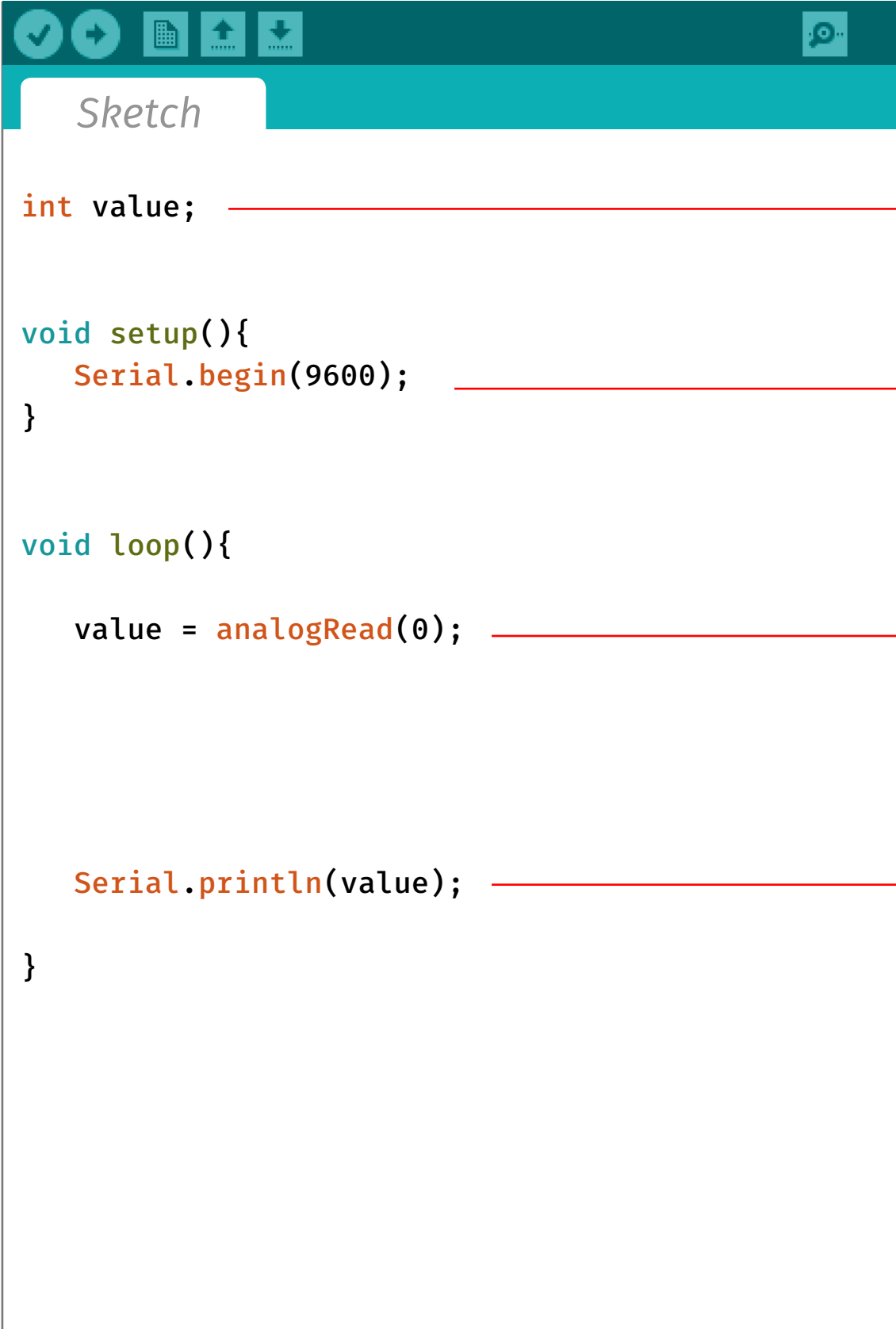


Sketch

```
void setup(){  
  
}  
  
void loop(){  
  value = analogRead(0);  
}
```



Code



```
int value;

void setup(){
  Serial.begin(9600);
}

void loop(){
  value = analogRead(0);

  Serial.println(value);
}
```

This variable is to store the value of the photo resistor.

Starts the serial communication. It makes the exchange of data between the Arduino and the computer possible so we can see it.

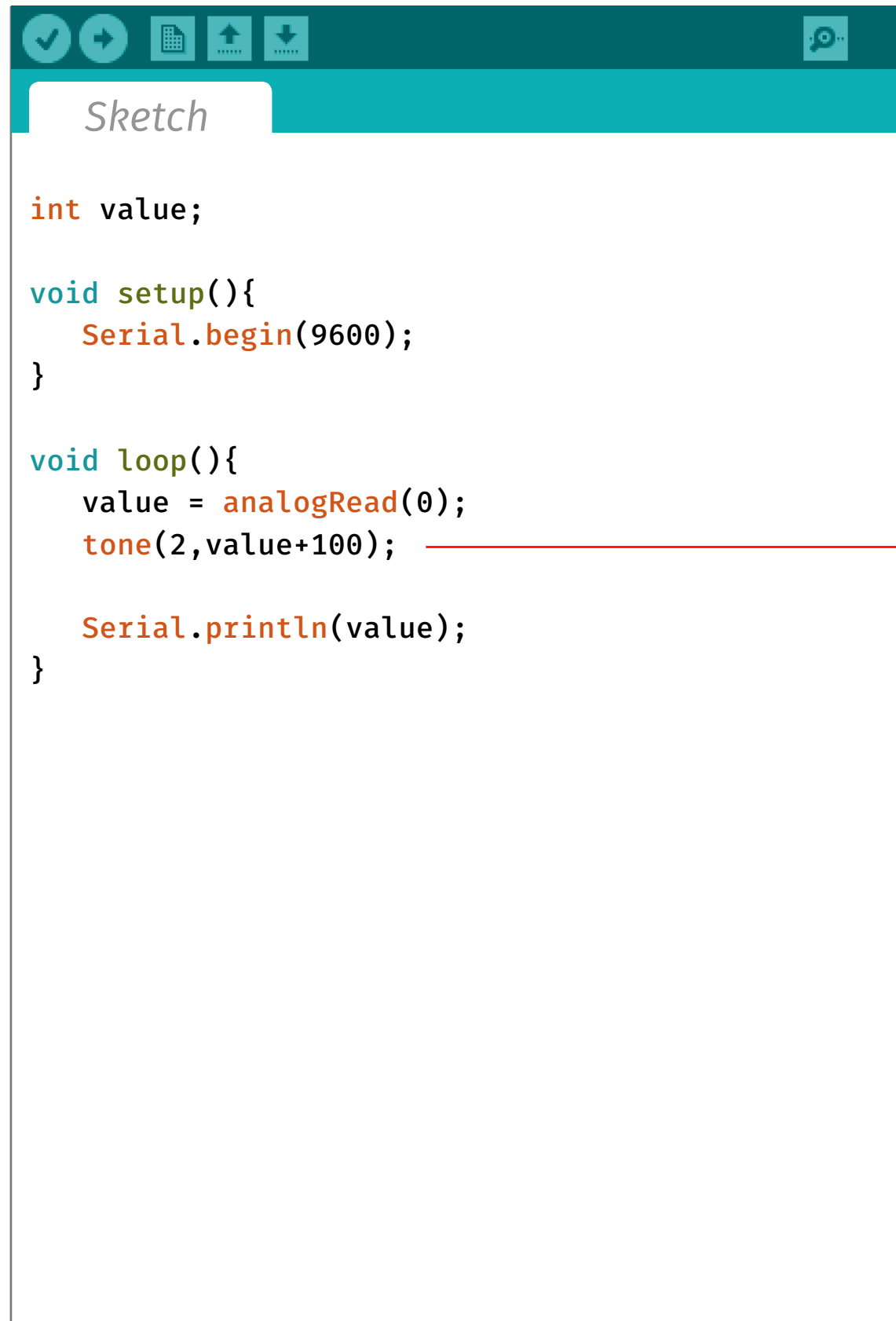
The command `analogRead(Pin)` delivers values between 0 and 1023 — in our case dependent on the incoming light.

Here, the value is send to the serial port.



Click on the magnifying glass icon in the menu bar. It opens the serial monitor. This is the place where the data is displayed. What do you get?

Code



```
int value;

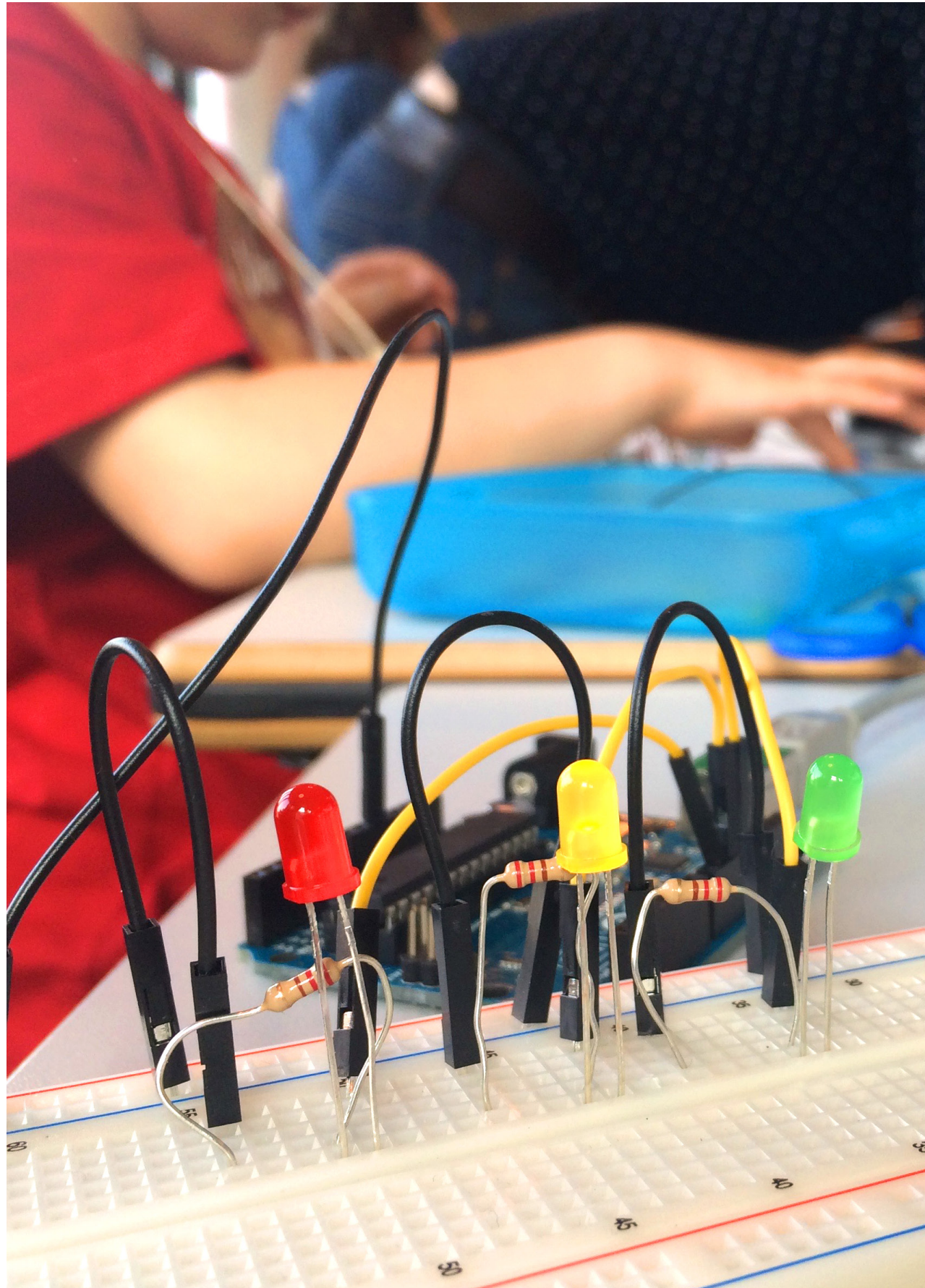
void setup(){
  Serial.begin(9600);
}

void loop(){
  value = analogRead(0);
  tone(2,value+100);

  Serial.println(value);
}
```

Now, we create a tone out of this value using a piezo speaker. We simply put the value into the `tone();` command but add 100 to it.

comming to an end



Conclusion

- Circuits (flash light)
- The Arduino
- The Breadboard
- *Commands*
- Morse code
- Traffic light for cars
- Traffic light for cars and pedestrians
- Piezo speaker
- *if statement*
- One-button-piano
- *Variables*
- analog input
- *serial communication*

The End

The beauty and elegance of programmable objects.

Presentation, PDF



Stefan Hermann, 2015
Gülser Weg 23
12681 Berlin

hallo@starthardware.org



This document is licensed under a Creative Commons Lizenz BY – SA 4.0

You are free to

Share — copy and redistribute the material in any medium or format

Adapt — remix, transform, and build upon the material
for any purpose, **even commercially**.

The licensor cannot revoke these freedoms as long as you follow the license terms.

Under the following terms

Attribution — You must give appropriate credit, provide a link to the license, and indicate if changes were made.
You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use.

ShareAlike — If you remix, transform, or build upon the material, you must distribute your contributions under the same license as the original.

No additional restrictions — You may not apply legal terms or technological measures that legally restrict others from doing anything the license permits.

