

## 01 - Multi LEDs

Now that you've successfully gotten a single LED to blink with Arduino, let's look at adding more! In this project, we'll introduce you to a new tool called a solderless breadboard and look more in-depth into the code that we're using.



## SKILL REQUIREMENTS

ELECTRICAL PROTOTYPING	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
ROBOTICS	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
SOLDERING	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
PROGRAMMING	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
DIY	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

### MATERIALS LIST

- Computer
- Arduino UNO
- USB Cable
- LEDs
- Breadboard

### STEP 1: Introducing the Breadboard (Solderless Breadboard, that is!)

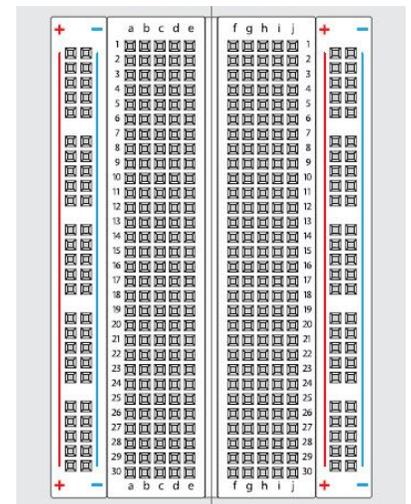
Find this nifty device in your kit. It's called a *solderless breadboard*. It allows us to prototype our circuits quickly without soldering or twisting wires together.

Take a look at your breadboard. Do you notice that there is a pattern or grouping of 5 holes across the entire board? Each horizontal row of 5 holes are connected internally by a little metal clip that looks like this. When you plug wires into the same row, they are connected together by this nifty metal clip.



Also, notice that there is a divide in the middle of the board. This is what we call the "ditch." The groupings of 5 holes on the left side are separate from the groupings on the right side.

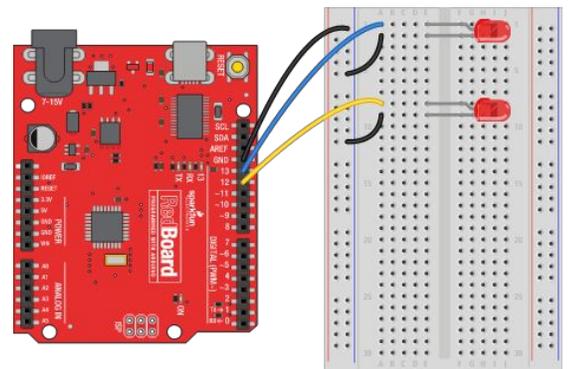
Finally, along the sides have groups of 2 holes that run vertically. These are called power rails. On these, the entire column is connected. We use this to provide power and GND to a variety of components. Let's wire up our first circuit!



### STEP 2: Wiring two LEDs<sup>1</sup>

Use the power rails on the side to connect GND to each of the LEDs. Notice that a wire goes from GND to the top of the column. It can go into any of the holes in this vertical column.

Next, connect up two LEDs. Notice that the legs on the LEDs are in two different rows. This allows us to connect pin 13 (or pin 12) to one of the legs of the LED and GND to the other.



Made with Fritzing.org

### STEP 3: Code

The full code example is available on codebender at this URL:  
<https://codebender.cc/sketch:136777>

While you can simply use the example code, let's take a look at how this code differs from our last project. Starting with the "Blink" example from our last project, we need to add a few lines of code to make this project work.

<sup>1</sup> Note: We purposefully do not use current limiting resistors in these circuits. It is

First, we need to add a `pinMode()` command inside the `setup()` function. Add this line:

```
pinMode(12, OUTPUT);
```

Note: When adding lines to the `setup()` make sure that it is in between the two curly braces `{ }`.

All of the pins on the Arduino board are generic or general purpose. Before we use a pin, we need to configure it. This command sets the pin up as an OUTPUT so that it can drive current to the LED.

Next, controlling the pin. We have a simple function to control the voltage on a pin.

```
digitalWrite(12, HIGH);
```

Now, to make one LED on and the other off and vice-versa, our `loop()` ends up looking like this:

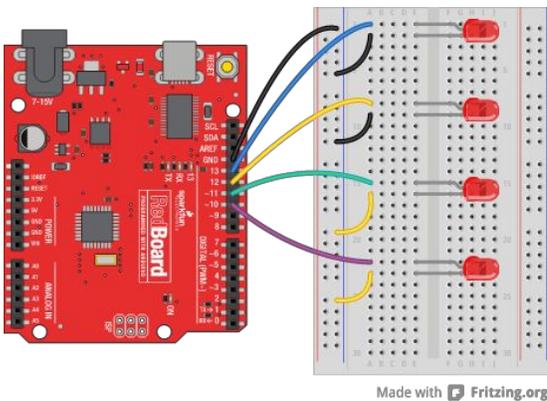
```
digitalWrite(12, HIGH);  
digitalWrite(13, LOW);  
delay(150);  
digitalWrite(12, LOW);  
digitalWrite(13, HIGH);  
delay(150);
```

When your code is ready, click on the Upload button.



This will *compile* your code - converting it into 1's and 0's - or machine language for the Arduino and then upload it to the board. If you get errors, double-check your code for spelling, capitalization, and missing semi-colons. The code must be exactly as it is in our example. These are all essential.

#### STEP 4: Mo' LEDs



Now that you have two LEDs working, let's add two more! Here's a diagram to help you out. Now, modify the code to control all four of these LEDs.

Open Challenges:

- **Outside-In / Inside-Out:** Modify the code so that we get alternating LEDs blinking -- alternating between the outer LEDs and then the inner ones. Once you get it, save this code.
- **Landing lights:** Now, modify the code so that you get the LEDs turning on one at a time and then all off. Again, save this code example (using a new filename).
- **Make your own light show:** Invent your own light show. Add more LEDs, if you want. Can you get it to speed up then slow down? Choose your own adventure!
- **Man Overboard:** Take your LED design and move the LEDs off the breadboard and onto a piece of cardboard, plastic, or other medium. Be imaginative and create your own light sculpture.

#### Need more help?

- An example of the Landing lights code is available here: <https://codebender.cc/sketch:136774>
- If your LEDs are dim, make sure that you have a `pinMode()` command in your `setup` for each pin that you're using.
- Compiling errors? Double check to make sure that you didn't leave out any semicolons. Make sure that everything is spelled correctly and uses the correct capitalizations.



Visit us at: [learn.sparkfun.com](https://learn.sparkfun.com)

## Need more help?

- An example of the Landing lights code is available here:  
<https://codebender.cc/sketch:136774>
- If your LEDs are dim, make sure that you have a `pinMode()` command in your setup for each pin that you're using.
- Compiling errors? Double check to make sure that you didn't leave out any semicolons. Make sure that everything is spelled correctly and uses the correct capitalizations.

