< foreach

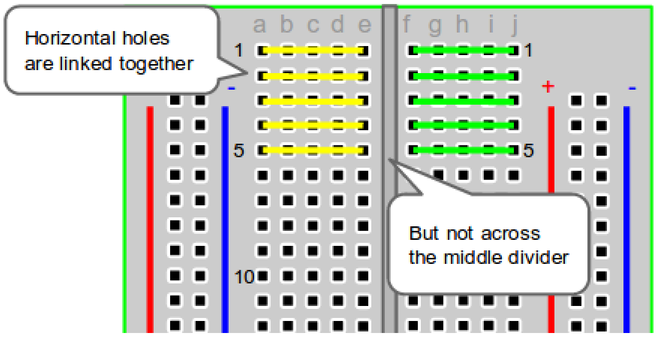
Breadboard and LED Blinking

Breadboard

A breadboard is a great tool for quickly testing out a prototype circuit or hooking up a quick experiment. Breadboarding is an important step in testing out ideas or learning how things work. In this lesson, the basics of how a breadboard is laid out will be explained.

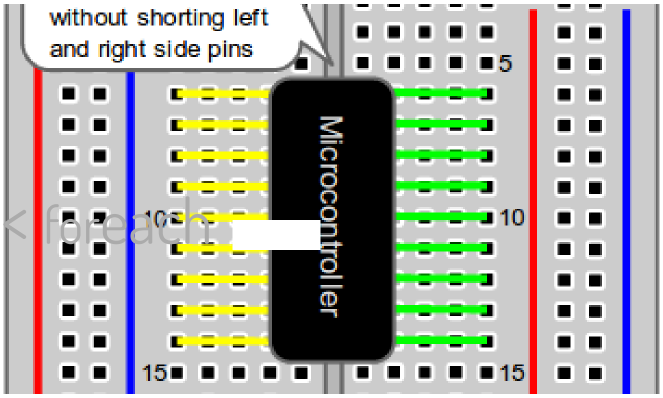
Horizontal Rows

Horizontal rows are connected together in the inside.

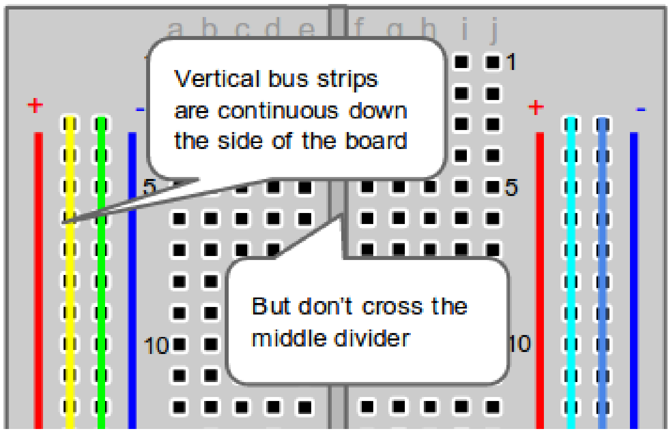

Horizontal holes are linked together
a b c d e
f g h i j
1
5
10
+
-
But not across the middle divider

When we put an IC chip on board it should hurdle the center divider.

We can see from diagram below that the pins of the IC are now accessible by inserting a component or jumper wire in the available horizontal pins and connecting to another row.



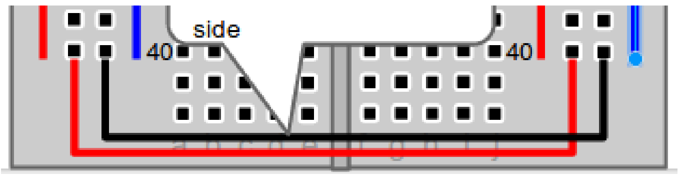
Vertical Columns


a b c d e f g h i j
+
-
1
5
10
Vertical bus strips are continuous down the side of the board
But don't cross the middle divider

Vertical columns on the side of the breadboard are for power and ground.

These power rails are also isolated to the right and left side of the breadboard. If you have to manage two different power supplies or voltages, they can be isolated by keeping them on either side of the board.

For ease of use many people link left and right side of the board so voltage and ground are handy on both sides of the center.



LED Blinking

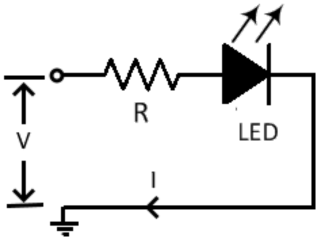
Let's apply the knowledge about breadboard to blink an LED. LED blinking can be considered as a Hello World Program compared to other languages!

Required Materials:

- Arduino UNO as Development Board
- LED
- 220 ohm resistor (220R) but any other resistor with any value ranging from 100 to 330 ohms kilo ohm can do the job.
- Breadboard
- Jumper wires

***Soldering Iron, and Soldering Wire** are used to solder together a resistor and an LED. In this practice a resistor is soldered to a positive leg of the LED. But it can be soldered to a negative leg of the LED as well.

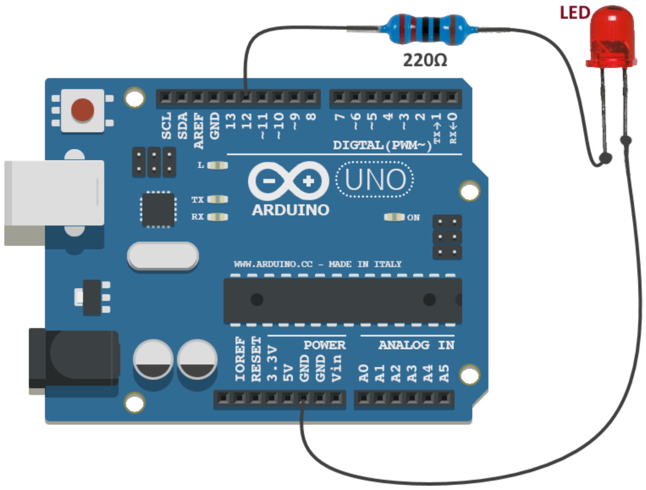
Theory behind this practice



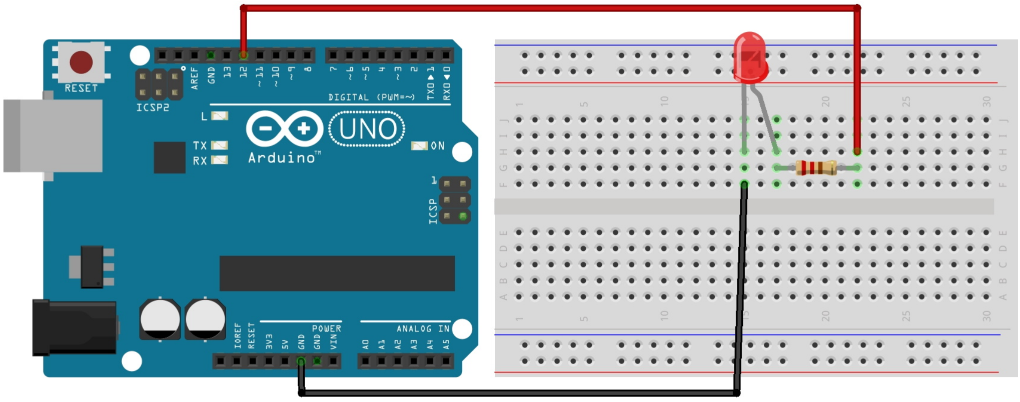
the LED according to Ohm's Law: I = V/R, the current; R, the resistance offered by the resistor; and V, the voltage supplied to the circuit.

If too much current is going through your LED, it will burn out too fast. If too little current is going through it, it might not be enough to lit the LED.

Circuit Diagram



The diagram below shows how to wire the circuit given above using breadboard.



we'll be able to take it down to the breadboard. Later at the end of the course, we'll learn how to deploy a circuit to the Printed Circuit Board (PCB).

Code

```
int LED = 12;
void setup(){
  pinMode(LED, OUTPUT);
}
void loop(){
  digitalWrite(LED, HIGH);
  delay(1000);
  digitalWrite(LED, LOW);
  delay(1000);
}
```

Code Explanation

`int LED = 12;` //We declare a variable LED of type integer and assign it 12 which is the pin number on the Arduino Board. The pin can be specified as either a variable or constant (0-13).

`void setup(){` //We declare a function setup() that returns nothing and that has no parameters. The setup() function is modes, begin serial, begin lcd and much more. It must be included in a program even if there are no statements to run.

`pinMode(LED, OUTPUT)` //pinMode() function is used to define the mode of a digital pin LED as OUTPUT.

`void loop(){` //We declare a function loop() that returns nothing and that has no parameters. The loop() function is called once when your program starts. After calling the setup() function, the loop() function does precisely what its name suggests, and loops consecutively. It must be included in a program even if there are no statements to run.

`digitalWrite(LED, HIGH)` //digitalWrite() function is used to set the digital pin LED to HIGH thus turning ON the LED connected to that pin.

`delay(1000)` // waits for one 1000 milliseconds. It pauses a program for the amount of time as specified in milliseconds, where 1000 equals 1 second.

`digitalWrite(LED, LOW)` //digitalWrite() function is used to set the digital pin LED to LOW thus turning OFF the LED connected to that pin.