**Understanding the working of Breadboard:**

**What is a breadboard?**

A breadboard is being used for building temporary circuits. You can use it for proof of concept circuits which can later be designed as a printed circuit board or soldered. This way you avoid having to solder and desolder components many times.

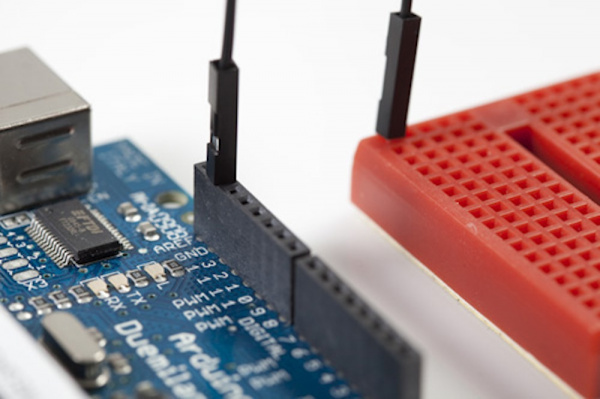
You can easily prick and remove all your components in the breadboard. The holes are pre-connected in a special way. This is useful because you don't need wires to create these connections

**Providing Power to a Breadboard:**

When it comes to providing power to you breadboard, there are numerous options.

Borrowing from Other Power Sources

If you are working with a development board such as an Arduino, then you can simply pull power from the Arduino’s female headers. The Arduino has multiple power and ground pins that you can connect to the power rails or other rows on a breadboard.

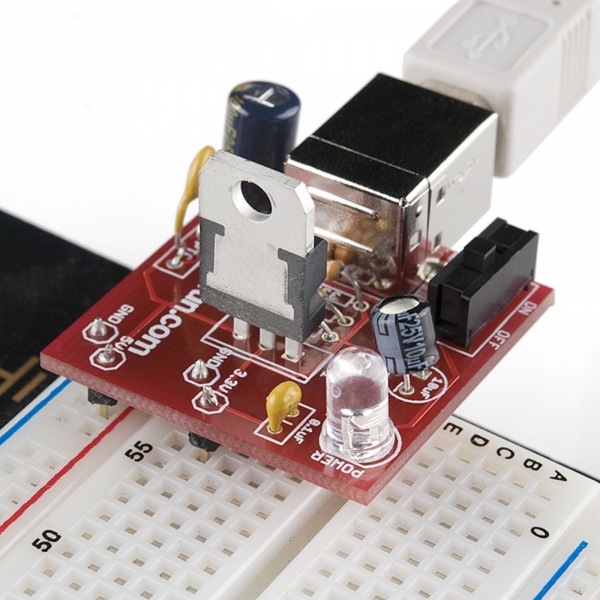
.

Connecting the Ground (GND) pin from an Arduino to a row on a mini breadboard. Now any leg or wire connected to that row will also be connected to Ground

The Arduino usually gets its power from the USB port on a computer or an external power supply such as a battery pack or a wall wart.

**Breadboard Power Supplies**

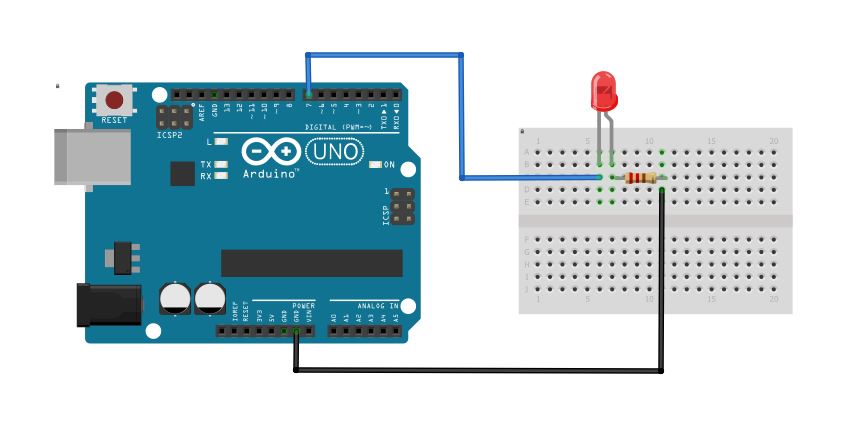
Yet another method for powering your breadboard is to use one of the many breadboard power supplies available. SparkFun carries a number of kits and boards that you can use to plug power directly into your breadboard. Some allow you to plug a wall wart directly into the breadboard. Others allow you to pull power directly from your computer via the USB connections. And, almost all of them have the capability to adjust the voltage, giving you a full range of the common voltages needed when building circuits.



**Building Your First Breadboard Circuit**

Now that we're familiar with the internals of a breadboard and how to provide power to them, what do we do with them? We are going to start with a simple circuit.

**Circuit Diagram**

****

**Arduino Code**

int ledPin = 7;

void setup() {

pinMode(ledPin, OUTPUT);

}

void loop() {

digitalWrite(ledPin, HIGH);

delay(1000);

digitalWrite(ledPin, LOW);

delay(1000);

}

**Explaining the Code**

Here we are going to write a program to blink an LED interval of 1000 ms (1 sec). The code will be the same as onboard LED Blinking. You only have to change the pin number accordingly. We are using pin no 7 (Digital Pin) so we will write int ledPin = 7; and other things will remain the same.

First, we initialize pin no 7 as the led pin and set it as an output pin. Then we create an infinite loop in which the LED pin is in a high state for 1000 ms and low for the next 1000 ms and continues..