

What is an Arduino?



The Arduino Revolution

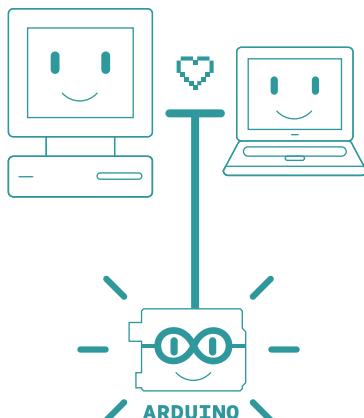
Arduino is an open-source physical computing platform designed to make experimenting with electronics more fun and intuitive. Arduino has its own unique, simplified programming language, a vast support network, and thousands of potential uses, making it the perfect platform for both beginner and advanced DIY enthusiasts.

arduino.cc

A Computer for the Physical World

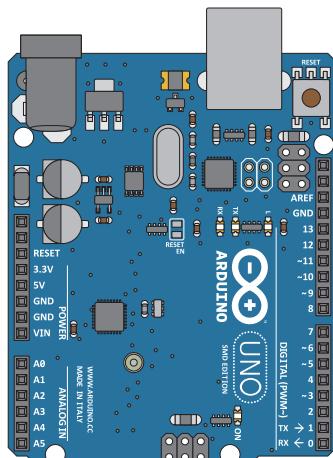
The friendly blue board in your hand (or on your desk) is the Arduino. In some ways you could think of Arduino as the child of traditional desktop and laptop computers. At its roots, the Arduino is essentially a small portable computer. It is capable of taking **inputs** (such as the push of a button or a reading from a light sensor) and interpreting that information to control various **outputs** (like a blinking LED light or an electric motor).

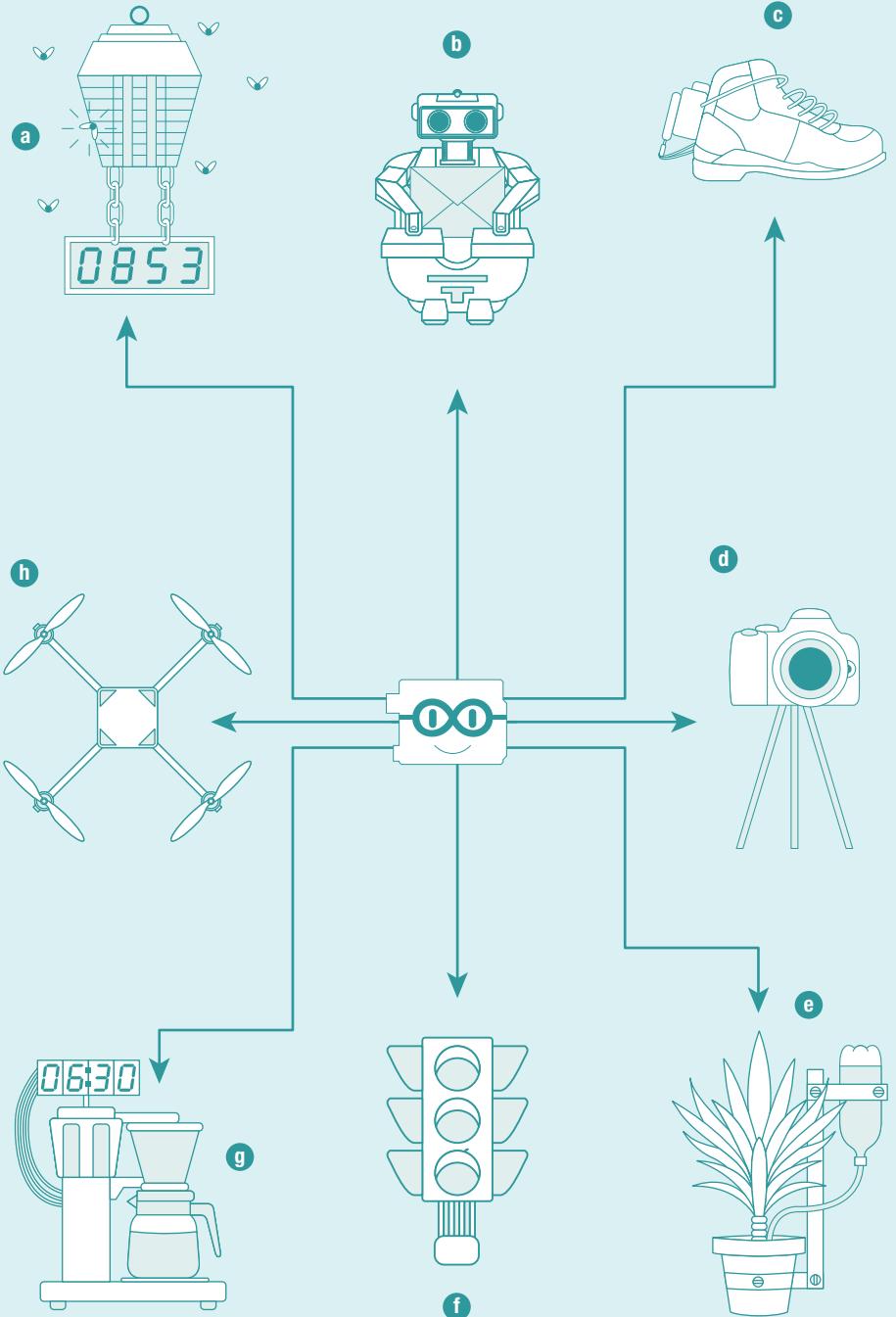
That's where the term "physical computing" is born - an Arduino is capable of taking the world of electronics and relating it to the physical world in a real and tangible way. Trust us - this will all make more sense soon.



// Arduino UNO SMD R3

The Arduino Uno is one of several development boards based on the ATmega328. We like it mainly because of its extensive support network and its versatility. It has 14 digital input/output pins (6 of which can be PWM outputs), 6 analog inputs, a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. Don't worry, you'll learn about all these later.





a Bug Zapper Counter

b Old Toy Email Notifier

c Power-Lacing High Tops

d Camera Time-lapse operation

e Auto-Plant Watering

f Re-Programmed Traffic Light

g Auto-Coffee Maker

h Quadcopter

Download the Arduino IDE (Integrated Development Environment)



Access the Internet

In order to get your Arduino up and running, you'll need to download some software first from www.arduino.cc (it's free!). This software, known as the Arduino IDE, will allow you to program the Arduino to do exactly what you want. It's like a word processor for writing programs. With an internet-capable computer, open up your favorite browser and type in the following URL into the address bar:

arduino.cc/en/Main/Software < case sensitive >

1

Download

Click on the “+” sign next to your appropriate computer operating system.

- + Windows
- + Mac OS X
- + Linux: 32 bit, 64 bit
- + source

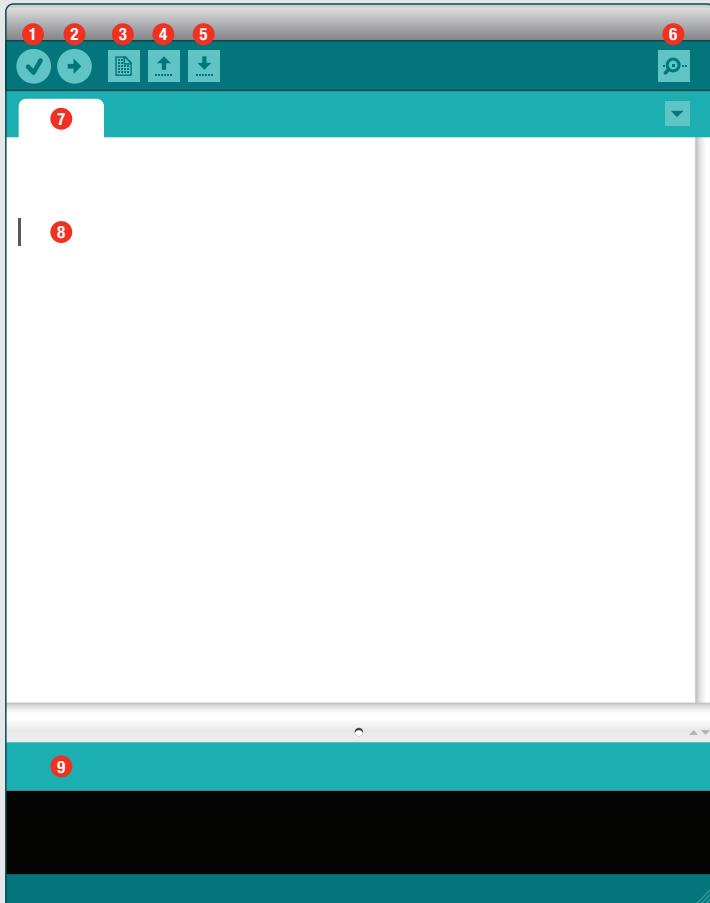


Choose the appropriate Operating System installation package for your computer.



// Open the Arduino IDE:

Open the Arduino IDE software on your computer. Poke around and get to know the interface. We aren't going to code right away, this is just an introduction. The step is to set your IDE to identify your Arduino Uno.



GUI (Graphical User Interface)

- 1 **Verify:** Compiles and approves your code. It will catch errors in syntax (like missing semi-colons or parenthesis). // See Diagram Below
- 2 **Upload:** Sends your code to the Arduino board. When you click it, you should see the lights on your board blink rapidly. // See Diagram Below
- 3 **New:** This button opens up a new code window tab.
- 4 **Open:** This button will let you open up an existing sketch. // See Diagram Below
- 5 **Save:** This saves the currently active sketch.
- 6 **Serial Monitor:** This will open a window that displays any serial information your Arduino is transmitting. It is very useful for debugging.
- 7 **Sketch Name:** This shows the name of the sketch you are currently working on.
- 8 **Code Area:** This is the area where you compose the code for your sketch.
- 9 **Message Area:** This is where the IDE tells you if there were any errors in your code.

// The three most important commands for this guide are seen below:



Open



Verify

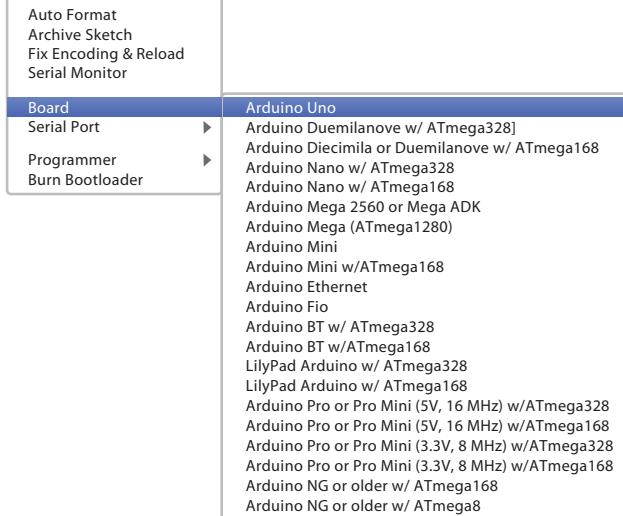


Upload

4

// Select your board: Arduino Uno

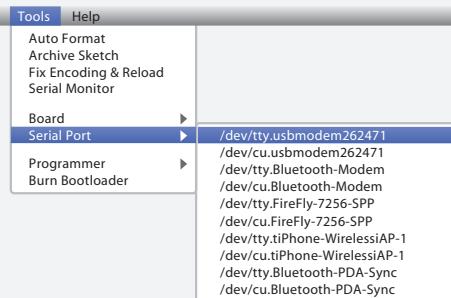
File Edit Sketch Tools Help



Select the serial device of the Arduino board from the Tools | Serial Port menu. This is likely to be **com3 or higher** (COM1 and COM2 are usually reserved for hardware serial ports). To find out, you can disconnect your Arduino board and re-open the menu; the entry that disappears should be the Arduino board. Reconnect the board and select that serial port.

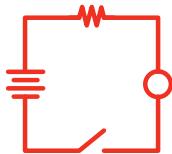


Select the serial device of the Arduino board from the Tools > Serial Port menu. On the Mac, this should be something with **/dev/tty.usbmodem** (for the Uno or Mega 2560) or **/dev/tty.usbserial** (for older boards) in it.



<http://www.arduino.cc/playground/Learning/Linux>

Getting Started with Circuits



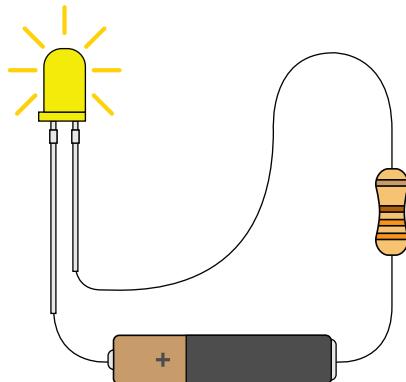
What is an Electrical Circuit?

A circuit is basically an electronics loop with a starting point and an ending point - with any number of components in between. Circuits can include resistors, diodes, inductors, sensors of all sizes and shapes, motors, and any other handful of hundreds of thousands of components.

Circuits are usually divided into three categories - analog circuits, digital circuits, or mixed-signal circuits. In this guide, you will explore all three sets of circuits.

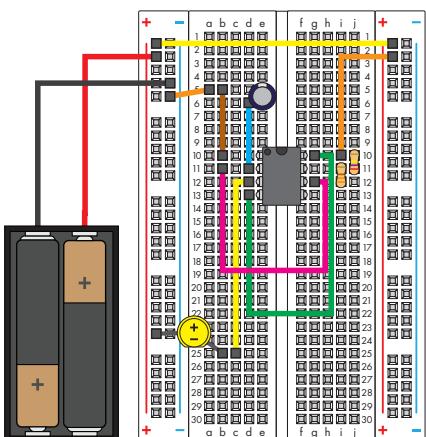
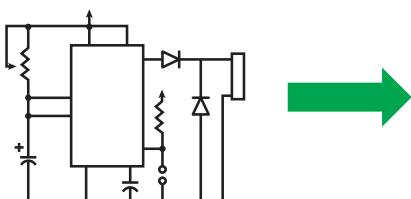
The World Runs on Circuits:

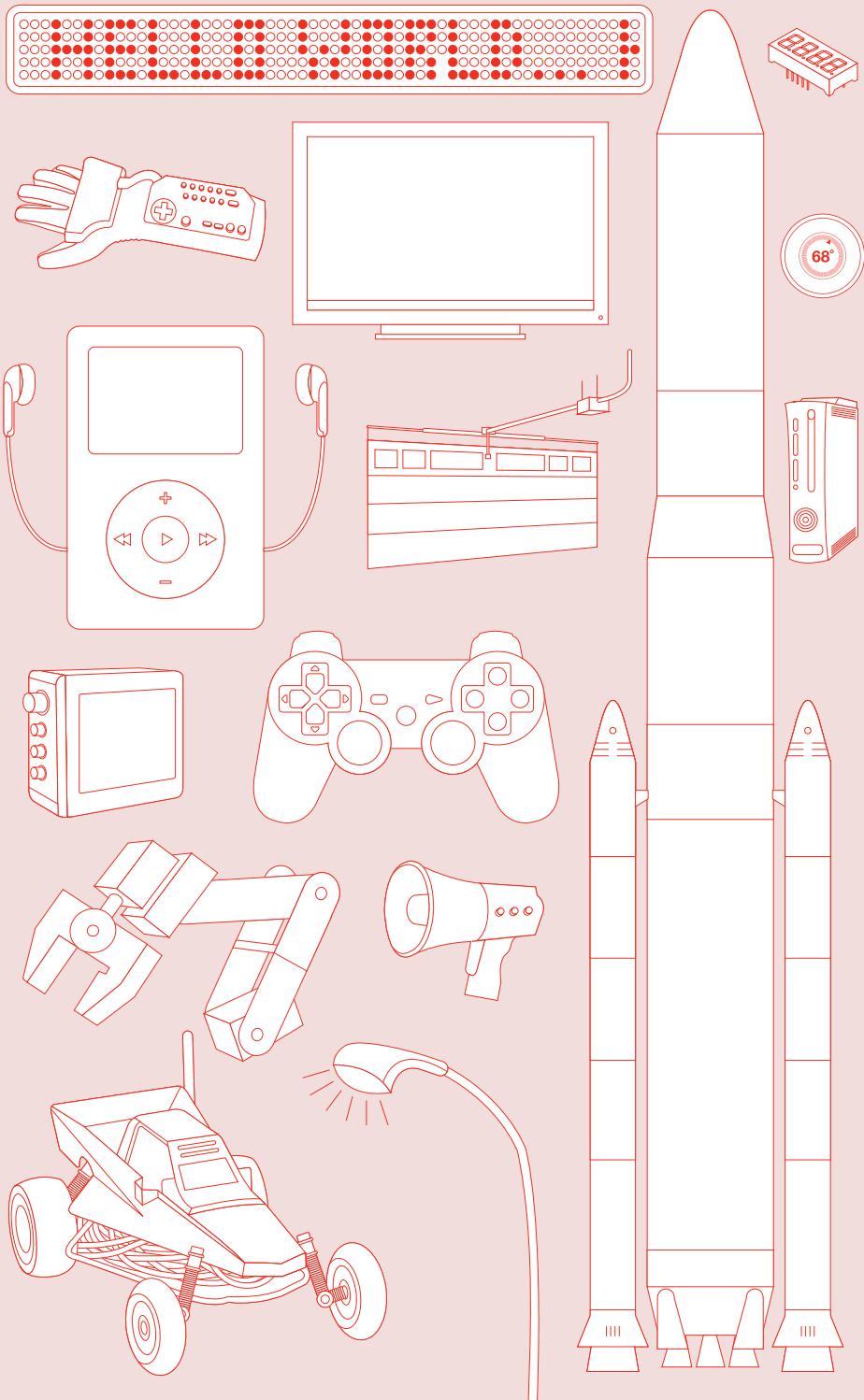
Everywhere you look, you'll find circuits. The cell phone in your pocket, the computer that controls your car's emissions system, your video game console - all these things are chock full of circuits. In this guide, you'll experiment with some simple circuits and learn the gist of the world of embedded electronics.



// Simple and Complex Circuits

In this guide, you will be primarily exploring simple circuits - but that doesn't mean you can't do amazing things with simple tools! When you've finished the SIK, your knowledge of circuits will enable you to explore amazing projects and unleash the power of your imagination.





Inventory of Parts

Jumper Wire

Various Colors

x30



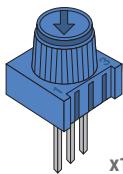
330Ω Resistor



x25

* ACTUAL SIZE

Potentiometer



x1

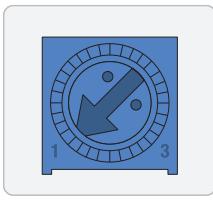
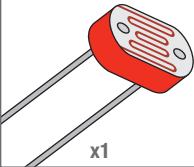
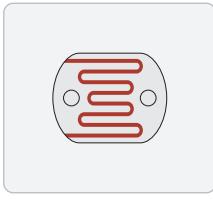


Photo Resistor

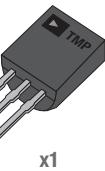


x1

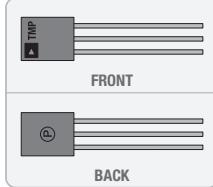


Temp. Sensor

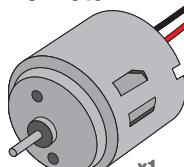
(TMP36)



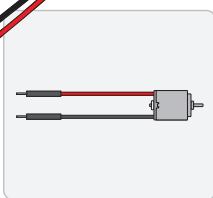
x1



DC Motor



x1



LED (5mm)

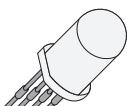
(Light Emitting Diode)



x10



x10



x1

10KΩ Resistor

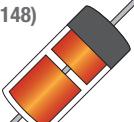


x25

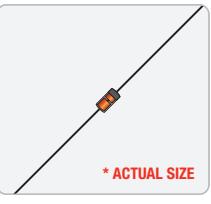
* ACTUAL SIZE

Diode

(1N4148)

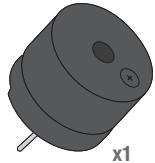


x2

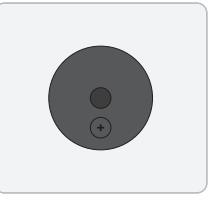


* ACTUAL SIZE

Piezo Element



x1

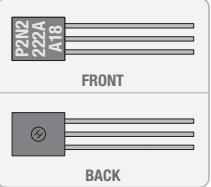


Transistor

(P2N2222AG)



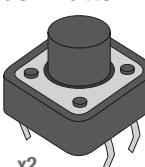
x2



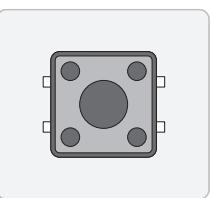
FRONT

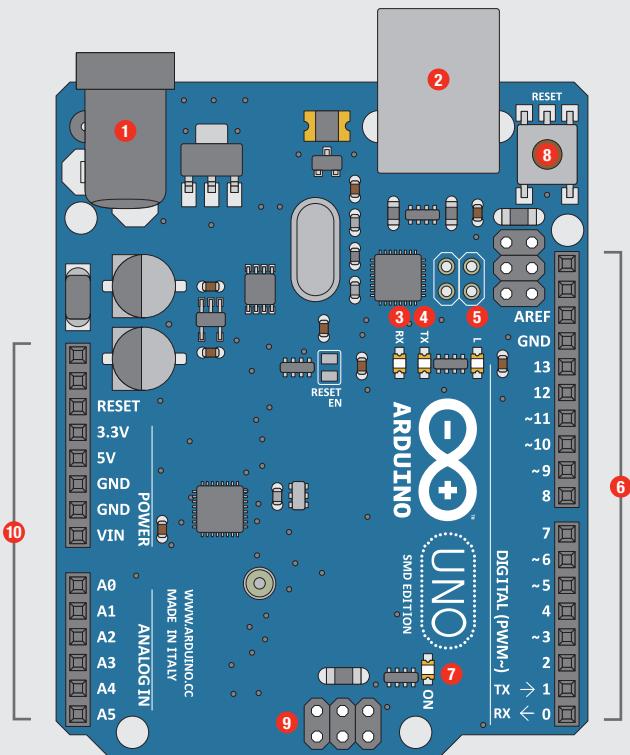
BACK

Push Button



x2

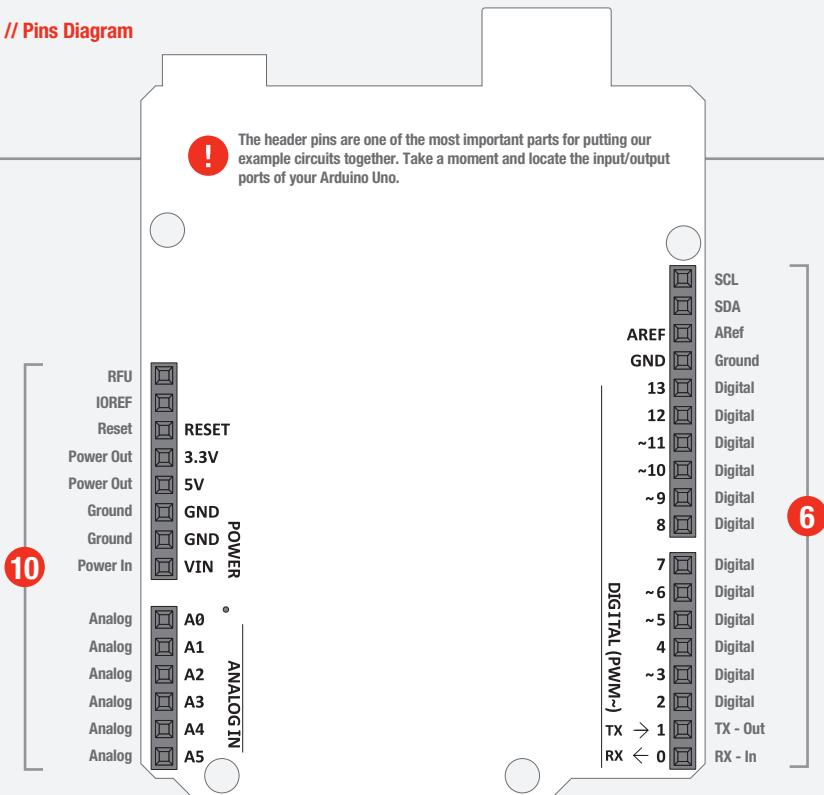




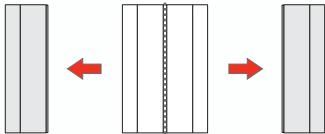
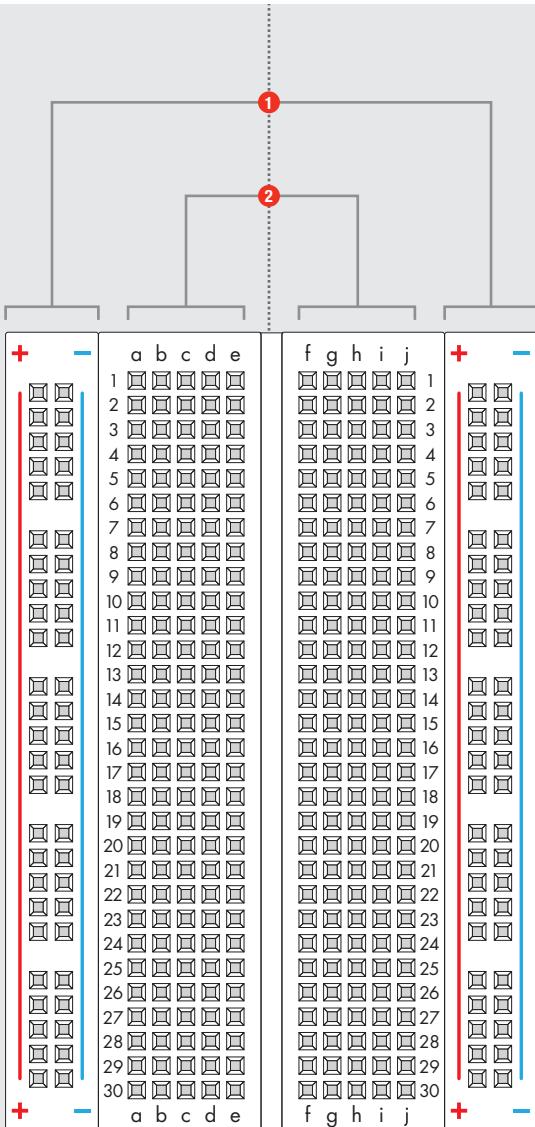
Arduino Uno

- 1 Power In (Barrel Jack) - Can be used with either a 9V or 12V wall-wart or battery.
- 2 Power In (USB Port) - Provides power and communicates with your board when plugged into your computer via USB.
- 3 LED (RX: Receiving) - This shows when the Arduino is receiving data (such as when being programmed).
- 4 LED (TX: Transmitting) - This shows when your Arduino is transmitting data (such as when running a program).
- 5 LED (Pin 13: Troubleshooting) - This LED is incorporated into your sketch to show if your program is running properly.
- 6 Pins (ARef, Ground, Digital, Rx, Tx) - These various pins can be used for inputs, outputs, power, and ground. // See Diagram Below
- 7 LED (Indicates Arduino is ON) - This is a simple power indicator LED.
- 8 Reset Button - This is a way to manually reset your Arduino, which makes your code restart.
- 9 ICSP Pins (Uploading Code without Bootloader) - This is for "In-Circuit Serial Programming," used if you want to bypass the boot loader.
- 10 Pins (Analog In, Power In, Ground, Power Out, Reset) - These various pins can be used for inputs, outputs, power, and ground. // See Diagram Below

// Pins Diagram



~ = PWM/Analog out compatible (i.e. ~3)



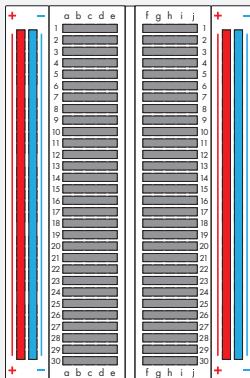
This line divides the board in half, restricting electricity to one half or the other.

Breadboard

1 Vertical Connection (+ Power and - Ground // See Diagram Below)

2 Horizontal Connection (a-e & f-j // See Diagram Below)

How's it all connected?



+ Power:

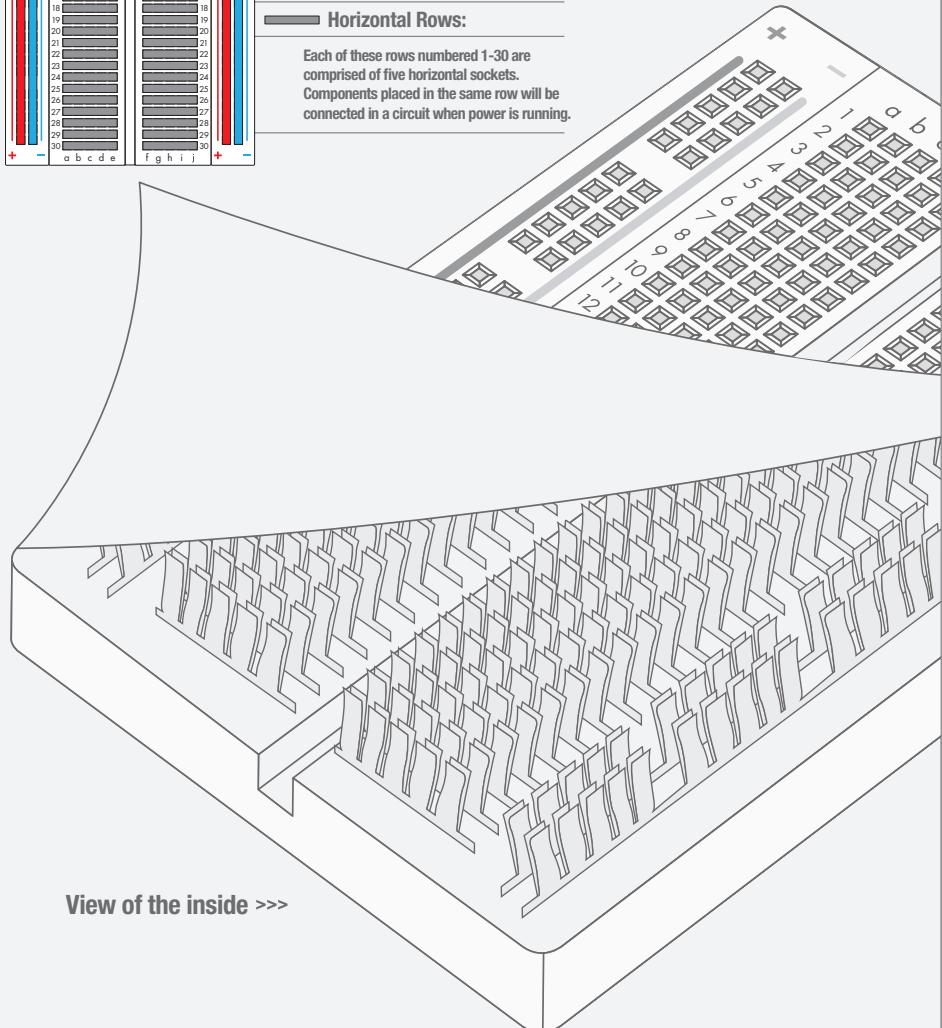
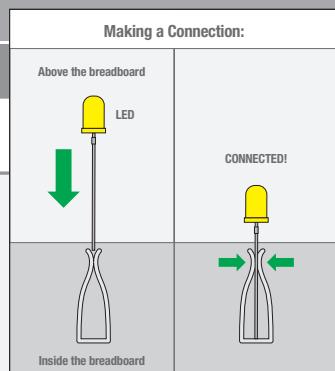
Each + sign runs power anywhere in the vertical column.

- Ground:

Each - sign runs to ground anywhere in the vertical column.

Horizontal Rows:

Each of these rows numbered 1-30 are comprised of five horizontal sockets. Components placed in the same row will be connected in a circuit when power is running.



CIRCUIT #1 - Your First Circuit

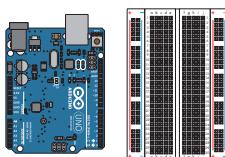
How It Works:

1 ASSEMBLE

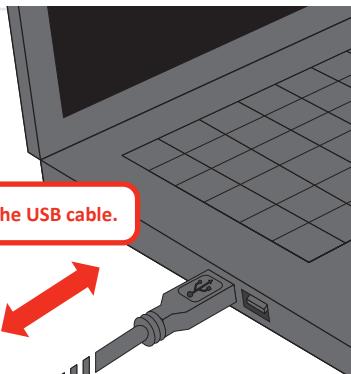
2 WRITE

3 UPLOAD

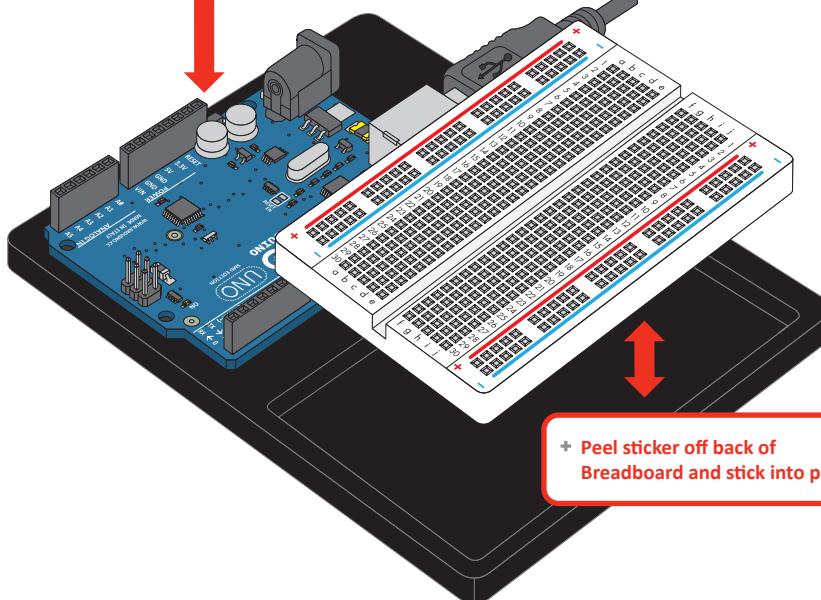
- + Make sure the text on the Arduino and Breadboard are facing up so you can read them.



- + Connect the USB cable.



- + Screw the Arduino board down and into place.



- + Peel sticker off back of Breadboard and stick into place.



5V Current Your Arduino runs on five volts. This is the power that will be supplied from your computer via USB and will be the driving force behind any components you use in your circuits. By plugging your Arduino board into your computer, you are supplying it with just the right voltage it needs to thrive! 5V can't hurt you, so don't be afraid to touch anything in your circuit.